

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *Aerotech U100M* at our website: ***Click HERE***

THE UNIDEX® 100/U100i MOTION CONTROLLER

OPERATION & TECHNICAL MANUAL

P/N: EDU128 (V1.6)



AEROTECH, Inc. • 101 Zeta Drive • Pittsburgh, PA. 15238-2897 • USA
Phone (412) 963-7470 • Fax (412) 963-7459
Product Service: (412) 967-6440; (412) 967-6870 (Fax)

www.aerotechinc.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

If you should have any questions about the UNIDEX 100/UNIDEX 100i controllers or comments regarding the documentation, please refer to Aerotech online at:
http://www.aerotechinc.com.

For your convenience, a product registration form is available at our web site.

Our web site is continually updated with new product information, free downloadable software and special pricing on selected products.

The UNIDEX 100 PC-based motion controller and UNIDEX 100i are products of Aerotech, Inc., LabVIEW software package is a product of Natioinal Instruments.
Borland C is a product of Borland International, Inc.
IBM PC/AT bus is a registered trademark of International Business Machines, Inc.
Inductosyn is a registered trademark of Farrand Industries, Inc.
MS-DOS, QuickBASIC and Windows are products of Microsoft Corporation.
Windows NT and Windows 95 are products of Microsoft.

The UNIDEX 100/U100i Motion Controller
Operations and Technical Manual Revision History:

Rev 1.0	July 1, 1993
Rev 1.1	December 10, 1993
Rev 1.2	June 14, 1994
Rev 1.3	November 13, 1995
Rev 1.4	September 20, 1996
Rev 1.5	August 31, 1998
Rev 1.6	July 24, 2000

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION 1-1

1.1. Overview of the UNIDEX 100/U100i System 1-1
1.2. Ordering Information 1-2
1.3. Options and Accessories 1-5

CHAPTER 2: GETTING STARTED 2-1

2.1. Introduction 2-1
2.2. Unpacking the UNIDEX 100/U100i System 2-1
2.3. Minimum Hardware Requirements and Recommended System Configurations 2-2
2.4. Inspection of the UNIDEX 100/U100i 2-3
2.5. Inspection of Interface Control Cards 2-3
2.6. Physical Setup of the Motion Controller 2-5
2.6.1 Standard Installation 2-8
2.7. Simple Program Applications 2-11
2.7.1. Sample Program Number One 2-12
2.7.2. Sample Program Number 2 2-15
2.7.3. Multitasking Programs 2-17

CHAPTER 3: HARDWARE CONFIGURATION & DESCRIPTION 3-1

3.1. Introduction 3-1
3.2. UNIDEX 100/U100i Motion Controllers 3-2
3.2.1. COM Port (U100/U100i) 3-3
3.2.2. Motor and Power Connections (U100 only) 3-3
3.2.3. LED Status Indicators (U100 only) 3-3
3.2.3.1. Shunt Active LED 3-4
3.2.3.2. Drive Fault LED 3-4
3.2.3.3. Marker LED 3-4
3.2.3.4. Reset LED 3-4
3.2.3.5. Control Fault LED 3-4
3.2.3.6. Remote LED 3-5
3.2.3.7. In Position LED 3-5
3.2.3.8. Overload LED 3-5
3.2.3.9. Limits (+ and -) LEDs 3-5
3.2.4. I/O Port (U100/U100i) 3-6
3.2.5. Encoder/Limits/Hall Effect Port (U100/U100i) 3-6
3.2.6. Auxiliary Encoder Port (U100/U100i) 3-6
3.2.7. Option Ports 3-7
3.2.8. Amplifier Feedback (U100i only) 3-7
3.3. UNIDEX 100/U100i Control Board Jumper Configurations 3-8
3.3.1. Current/Velocity Command Jumper (JP8) 3-9
3.3.2. Current Output Jumpers (JP13, JP18 & JP19) 3-10
3.3.3. Motor Type Configuration Jumpers (JP16 & JP17) 3-11
3.3.4. Firmware Boot Jumper (JP34) 3-11
3.4. U100 Power Configurations - U100 Controller 3-12
3.4.1. U100 Transformer 3-12
3.4.2. U100 Fuses 3-13
3.5. Installing Other Aerotech Components 3-14

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.5.1. The Hand Held Terminal (HT) 3-14
3.5.2. Joystick 3-16
3.5.3. Thumbwheel 3-17
3.5.3.1. Thumbwheel Sign 3-19
3.5.3.2. Thumbwheel Execute Button 3-19
3.5.4. Display Option 3-20
3.5.5. Resolver-to-Digital Converter 3-22
3.5.6. The INT Option 3-24
3.5.6.1. INT Jumper Settings 3-27
3.5.7. IE-488 Option 3-29
3.6. After Factory Installation of Option Boards 3-33

CHAPTER 4: SOFTWARE INSTALLATION AND INTERFACE 4-1

4.1. Introduction 4-1
4.2. Installing the Software Package 4-1
4.2.1. The Installation Disk 4-1
4.2.2. Standard Installation 4-3
4.3. Overview of the Main Menu Window 4-5
4.3.1. The File Edit Menu 4-6
4.3.2. Run Selection Menu 4-9
4.3.2.1. Auto Mode 4-10
4.3.2.2. Block Mode 4-10
4.3.3. MDI Selection Menu 4-10
4.3.4. Status Selection Menu 4-11
4.3.4.1. Motion Status Window 4-12
4.3.4.2. General Status Window 4-12
4.3.4.3. Memory Status and Version Windows 4-13
4.3.5. Control Selection Menu 4-13
4.3.6. MISC Selection Menu 4-15
4.3.6.1. Erase Files Menu 4-15
4.3.6.2. Copy Files Menu 4-16
4.3.6.3. Setup Data Types Menu 4-17
4.3.6.4. Archive Files Menu 4-24
4.4. Controller Setup Process U100/U100i 4-26
4.5. Using the Hand held Terminal (HT) 4-27

CHAPTER 5: PROGRAMMING COMMANDS 5-1

5.1. Introduction 5-1
5.2. Multitasking Operating System 5-3
5.2.1. Multitasking Time Slots 5-3
5.2.2. Multiple Programs 5-7
5.3. Interrupts 5-8
5.4. Exception Processing 5-9
5.4.1. Invert Mask 5-10
5.4.2. Reset Status Register 5-11
5.4.3. Latch Mask 5-11
5.4.4. Error Status Register 5-11
5.4.5. Disable Mask 5-12
5.4.6. SRQ Mask 5-12
5.4.7. Auxiliary Mask 5-12

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.4.8. Freeze Mask 5-12
5.4.9. Task 1 Mask 5-12
5.4.10. Task 2 Mask 5-12
5.4.11. Positive Direction Mask 5-13
5.4.12. Negative Direction Mask 5-13
5.4.13. Status Active Register 5-13
5.5. Service Request 5-14
5.6. U100/U100i Supported File Types 5-17
5.6.1. Program Files (PGM) 5-17
5.6.2. Macro Files 5-19
5.6.3. List files (LST) 5-21
5.6.4. Definition Files (DEF) 5-22
5.6.5. CAM Files 5-23
5.7. Programming Commands 5-24
5.7.1. A Command (Accel/Decel Ramp Rate in User units/sec$^{2)}$ 5-27
5.7.2. ABS Absolute Value 5-28
5.7.3. ABSL Absolute Positioning Mode 5-29
5.7.4. ADC Analog to Digital Conversion 5-30
5.7.5. + Add 5-31
5.7.6. AND (Logical And) 5-32
5.7.7. ARCHIVE (ARCH ()) Command 5-32
5.7.8. Begin Command 5-33
5.7.9. CAM Command 5-33
5.7.10. CBI: Command 5-36
5.7.11. CIB Command 5-37
5.7.12. CLN Command 5-38
5.7.13. CLS Command 5-39
5.7.14. COS Command (Cosine) 5-40
5.7.15. CUR Command 5-41
5.7.16. D Command (Distance in User Units to move/position) 5-42
5.7.17. DAC Command (Digital to Analog Conversion) 5-45
5.7.18. DD Command (Direct Drive) 5-46
5.7.19. DEC Command (Decrement) 5-47
5.7.20. DEF File 5-48
5.7.21. DI Command (Disable Interrupt) 5-50
5.7.22. / Divide Command 5-51
5.7.23. DW Command (Dwell Time) 5-52
5.7.24. EI Command (Enable Interrupt) 5-52
5.7.25. ELSE Command 5-53
5.7.26. ELSEIF Command 5-54
5.7.27. END Command 5-55
5.7.28. ENDIF Command 5-56
5.7.29. ENDMAC Command 5-57
5.7.30. ENDSUB Command 5-58
5.7.31. ENDWHL Command 5-59
5.7.32. = Assign a value to 5-60
5.7.33. EXIT Command 5-60
5.7.34. GC Command (Get Character) 5-62
5.7.35. GEAR Command 5-62

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.36. GM Command (Get Message) 5-63
5.7.37. GO Command (Begin Move) 5-64
5.7.38. GOSUB Command 5-65
5.7.39. GOTO Command 5-66
5.7.40. HM Command (Hardware Home) 5-67
5.7.41. IF Command 5-69
5.7.42. INC Command (Increment) 5-70
5.7.43. INCR Command (Incremental Position Mode) 5-71
5.7.44. LB Command (Label) 5-72
5.7.45. LOCK Command (Lock Task Execution) 5-73
5.7.46. MAC Command 5-74
5.7.46.1. MAC Definition 5-75
5.7.47. MDX Command Modulo Index 5-76
5.7.48. * Multiply 5-77
5.7.49. OR Command (Logical OR) 5-78
5.7.50. PC Command (Print Character) 5-78
5.7.51. PM Command (Print Message) 5-79
5.7.51.1. PM Command (Print Formatted Messages) 5-80
5.7.52. PVI Command 5-81
5.7.53. RI Command (Reset Interrupt Latch) 5-82
5.7.54. RUN Command (Run Pre-compiled PGM Program) 5-83
5.7.54.1. RUN Command - Extended format 5-83
5.7.55. SIN Command (Sine) 5-84
5.7.56. SRQ Command (Service Request) 5-85
5.7.57. SQRT Command (Square root) 5-86
5.7.58. SUB Command 5-87
5.7.59. - Subtract 5-88
5.7.60. SYNC Command 5-89
5.7.61. T Command (Ramp Time) 5-90
5.7.62. TAN Tangent 5-91
5.7.63. TITLE Command 5-92
5.7.64. TUNE Command (Autotune the system) 5-93
5.7.65. V Command (Velocity) 5-94
5.7.66. WHL Command 5-95
5.7.67. XOR Exclusive OR 5-96

CHAPTER 6: PARAMETERS 6-1

6.1. Introduction 6-1
6.2. Communication Parameters 6-2
6.2.1. RS-232-C General Control Register PRM:001 6-4
6.2.2. RS-232-C Clock Control Register PRM:002 6-5
6.2.3. RS-232-C Loop Control Character PRM:003 6-6
6.2.4. RS-232-C Loop Address PRM:004 6-6
6.2.5. RS-232-C Daisy Chain Enable/Disable PRM:005 6-6
6.2.5.1. Daisy Chain Operation 6-7
6.2.6. RS-232-C Loop Group Trigger PRM:006 6-10
6.2.7. IEEE-488-Address Mode Register PRM:007 6-10
6.2.8. IEEE-488 Address Register "0" PRM:008 6-10
6.2.9. IEEE-488 Address Register "1" PRM:009 6-11
6.2.10. IEEE-488 Auxiliary Register "A" PRM:010 6-11
6.2.11. IEEE-488 Auxiliary Register "B" PRM:011 6-11

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.2.12. IEEE-488 Command Register PRM:012 6-12
6.2.13. IEEE-488 Auxiliary Register "E" PRM:013 6-12
6.2.14. IEEE-488 Parallel Poll Register PRM:014 6-13
6.2.15. IEEE-488 Timer Control Register PRM:015 6-13
6.2.16. IEEE-488 End of Sequence Register PRM:016 6-13
6.2.17. Communications Mode PRM:019 6-14
6.2.18. RS-232-C Group Trigger Enable/Disable PRM:020 6-14
6.2.19. Cursor Control for "CUR( )" Command PRM:021 6-15
6.2.20. Communications SRQ On/Off PRM:022 6-15
6.2.21. Send SRQ At End of Task 1 PRM:023 6-16
6.2.22. Send SRQ At End of Task 2 PRM:024 6-16
6.2.23. LST1 File Generation On/Off PRM:025 6-17
6.2.24. Auto DEF100/MAC100 Lookup in MDI Mode PRM:026 6-17
6.2.25. End Of File Character Code PRM:027 6-17
6.2.26. Service Request Character Code PRM:028 6-17
6.2.27. Service Acknowledge Character Code PRM:029 6-18
6.2.28. Task 1 Boot Program PRM:030 6-18
6.2.29. Task 2 Boot Program PRM:031 6-18
6.2.30. RS-232-C Xon Sent From Host PRM:032 6-18
6.2.31. RS-232-C Xoff Sent From Host PRM:033 6-18
6.2.32. Software Reset (HT Mode) PRM:034 6-19
6.2.33. Software Reset (Host Mode) PRM:035 6-19
6.2.34. Motion Status Total Digits, Line 1 PRM:036 6-19
6.2.35. Motion Status Decimal Digits, Line 1 PRM:037 6-20
6.2.36. Motion Status Total Digits, Line 2 PRM:038 6-20
6.2.37. Motion Status Decimal Digits, Line 2 PRM:039 6-20
6.2.38. Motion Status Scan Type, Line 2 PRM:040 6-21
6.2.39. Motion Status Scan Type, Line 1 PRM:041 6-21
6.3. Trajectory Generation 6-22
6.3.1. General Trajectory Generation 6-22
6.3.2. Scaled Trajectory Modifier 6-23
6.3.3. Direct Drive Trajectory Generation 6-23
6.3.4. CAM Trajectory Generation 6-23
6.3.5. First Order Filter 6-23
6.3.6. GEAR Function 6-23
6.3.7. PID Control Algorithm 6-23
6.3.8. Motor 6-23
6.4. Motion Parameters 6-24
6.4.1. User Units Scale Factor PRM:100 6-26
6.4.2. Command Segment Filter Coefficient PRM:101 6-26
6.4.3. Command Segment Filter On/Off PRM:102 6-26
6.4.4. Command Linear Interpolation PRM:103 6-26
6.4.5. Default Ramp Time PRM:104 6-27
6.4.6. Default Distance PRM:105 6-27
6.4.7. Default Acceleration/Deceleration PRM:106 6-27
6.4.8. Default Velocity PRM:107 6-27
6.4.9. "S" Curve Generation PRM:108 6-27
6.4.10. In Position “Wait” PRM:109 6-28
6.4.11. Lower "Modulo" Index Limit PRM:110 6-28
6.4.12. Upper "Modulo" Index Limit PRM:111 6-28

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.4.13. DD Command Scale Factor PRM:112 6-28
6.4.14. Home Switch Fast Reference PRM:115 6-29
6.4.15. Home Switch Direction PRM:116 6-29
6.4.16. Home Switch Selection PRM:117 6-29
6.4.17. Home Marker Selection PRM:118 6-30
6.4.18. Home Velocity After Power Up PRM:119 6-30
6.4.19. Home Limit Offset PRM:120 6-30
6.4.20. Home Ending Offset PRM:121 6-31
6.4.21. Home Marker Velocity PRM:122 6-31
6.4.22. Task 1 Exception Processing PRM:124 6-31
6.4.23. Task 2 Exception Processing PRM:125 6-31
6.4.24. +/- Limit Reset Distance PRM:128 6-32
6.4.25. + Software Limit PRM:129 6-32
6.4.26. - Software Limit PRM:130 6-32
6.4.27. Software Limit Reference PRM:131 6-32
6.4.28. External Interrupt Latch Source PRM:132 6-33
6.4.29. External Interrupt Mode PRM:133 6-34
6.4.30. External Interrupt Type PRM:134 6-35
6.4.31. External Interrupt Source PRM:135 6-35
6.4.32. R/D Resolution Mode PRM:137 6-36
6.4.33. Velocity Feedback Type PRM:138 6-36
6.4.34. Position Feedback Type PRM:139 6-37
6.4.35. Analog-to-Digital Scale Factor PRM:140 6-37
6.4.36. Analog-to-Digital Deadband PRM:141 6-37
6.4.37. Digital-to-Analog Scale Factor PRM:142 6-37
6.4.38. Scaled Trajectory Task 1 Variable Type PRM:144 6-38
6.4.39. Scaled Trajectory Task 1 Gain Factor PRM:145 6-39
6.4.40. Scaled Trajectory Task 1 Index (PV/BV/LV/FV) PRM:146 6-39
6.4.41. Scaled Trajectory Task 1 Ramp Time PRM:147 6-39
6.4.42. Scaled Trajectory Task 2 Variable Type PRM:148 6-40
6.4.43. Scaled Trajectory Task 2 Gain Factor PRM:149 6-40
6.4.44. Scaled Trajectory Task 2 Index (PV/BV/LV/FV) PRM:150 6-40
6.4.45. Scaled Trajectory Task 2 Ramp Time PRM:151 6-40
6.4.46. Spline Interpolation Mode PRM:152 6-41
6.4.47. Spline Point Number PRM:153 6-41
6.4.48. Spline Scaling Mode PRM:154 6-42
6.4.49. Spline Reference Mode PRM:155 6-42
6.4.50. Spline Reference Time PRM:156 6-43
6.4.51. Spline Reference Scale PRM:157 6-43
6.4.52. Spline Reference -Limit PRM:158 6-43
6.4.53. Spline Reference Starting Point PRM:159 6-43
6.4.54. Spline Reference (+Limit) PRM:160 6-44
6.4.55. Spline Modulo Mode PRM:161 6-44
6.4.56. Spline Output Mode PRM:162 6-45
6.4.57. Spline Output PV Index PRM:163 6-45
6.5. Drive Parameters 6-46
6.5.1. Stepper Power Stage Mode PRM:200 6-47
6.5.2. Stepper Running Current PRM:201 6-47
6.5.3. Stepper Holding Current PRM:202 6-47

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.5.4. Stepper Resolution PRM:203 6-48
6.5.5. Stepper Verification Enable/Disable PRM:204 6-48
6.5.6. Stepper Slip Tolerance PRM:205 6-48
6.5.7. Stepper Correction Velocity PRM:206 6-48
6.5.8. Stepper Damping Enable/Disable PRM:207 6-49
6.5.9. Stepper “Extended” Slip Tolerance PRM:208 6-49
6.5.10. Stepper “Extended” Correction Velocity PRM:209 6-49
6.5.11. Servo Current Limit Trap Time PRM:213 6-50
6.5.12. Servo "Velocity Error" Integration Control PRM:214 6-50
6.5.13. Servo "In Position" Tolerance PRM:215 6-50
6.5.14. Servo Peak Current Limit PRM:216 6-51
6.5.15. Servo Current Limit Trap Level PRM:217 6-51
6.5.16. Kp Servo Loop Gain PRM:218 6-51
6.5.17. Ki Servo Loop Gain PRM:219 6-51
6.5.18. Kpos Servo Loop Gain PRM:220 6-51
6.5.19. Kf Servo Loop Gain PRM:221 6-51
6.5.20. Servo Feedforward On/Off PRM:222 6-51
6.5.21. Servo Position Error Trap PRM:223 6-52
6.5.22. Servo Velocity Trap PRM:224 6-52
6.5.23. Servo Acceleration/Deceleration Trap PRM:225 6-52
6.5.24. Tune Minimum Frequency PRM:227 6-53
6.5.25. Tune Time PRM:228 6-53
6.5.26. Tune Damping Factor PRM:229 6-53
6.5.27. Tune Corner Frequency PRM:230 6-53
6.5.28. Gear Mode PRM:231 6-54
6.5.29. Gear Source PRM:232 6-54
6.5.30. Gear Scale Factor PRM:233 6-55
6.5.31. Axis Calibration Mode PRM:234 6-55
6.5.32. Axis Calibration Table Size PRM:235 6-55
6.5.33. Axis Calibration Increment Size PRM:236 6-56
6.5.34. Axis Calibration Backlash PRM:237 6-56
6.5.35. Brushless Commutation Type PRM:239 6-57
6.5.36. Brushless Commutation Step/Cycle PRM:240 6-57
6.5.37. Brushless Commutation Table Step Shift PRM:241 6-58
6.6. System Parameters 6-59
6.6.1. Port Variable (PV) Wait States PRM:300 6-60
6.6.2. Task 1 Background Time PRM:301 6-60
6.6.3. Task 2 Background Time PRM:302 6-60
6.6.4. Scaled Trajectory External Port Fetch On/Off PRM:303 6-61
6.6.5. Servo Loop Update Time PRM:304 6-61
6.6.6. Commutation Loop Update Time PRM:305 6-62
6.6.7. Stepper Loop Update Time PRM:306 6-62
6.6.8. Encoder 1 Update Time PRM:307 6-63
6.6.9. Encoder 2 Update Time PRM:308 6-64
6.6.10. ENC Update Time PRM:309 6-65
6.6.11. Resolver-to-Digital Update Time PRM:310 6-66
6.6.12. Fault Mask Update Time PRM:311 6-66
6.6.13. Control Loop Checks PRM:312 6-67
6.6.14. Combine Task 1 and Task 2 Time Slots PRM:313 6-67
6.6.15. Library Ending Offset PRM:314 6-67

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.6.16. Library Program Call Mode PRM:315 6-67
6.6.17. Library Memory Limit PRM:316 6-68
6.6.18. Error Status Invert Mask PRM:317 6-68
6.6.19. Error Status Latch Mask PRM:318 6-71
6.6.20. Error Status Disable Mask PRM:319 6-72
6.6.21. Error Status Service Request Mask PRM:321 6-73
6.6.22. Error Status Auxiliary Mask PRM:322 6-74
6.6.23. Error Status Freeze Mask PRM:323 6-75
6.6.24. Error Status Task 1 Mask PRM:324 6-76
6.6.25. Error Status Task 2 Mask PRM:325 6-77
6.6.26. Error Status "+" Direction Mask PRM:326 6-78
6.6.27. Error Status "-" Direction Mask PRM:327 6-79
6.6.28. Command Buffer Partition PRM:330 6-80
6.6.29. Extension Buffer Partition PRM:331 6-80
6.6.30. String Buffer Partition PRM:332 6-80
6.6.31. Reset Delay on Motion PRM:334 6-80

CHAPTER 7: REGISTERS AND VARIABLES 7-1

7.1. Introduction 7-1
7.2. Communication Registers 7-2
7.2.1. Bit Input (Read, integer) REG:001 7-4
7.2.2. Bit Output (Read/Write, integer) REG:002 7-4
7.2.3. Port B Status (Read Only, integer) REG:003 7-5
7.2.4. Port C Status (Read Only, integer) REG:004 7-6
7.2.5. Timer Control/Status (Read/Write, Integer) REG:005 7-7
7.2.6. Timer Count (Read/Write, Integer) REG:006 7-8
7.2.7. External Input (Read Only, Integer) REG:007 7-8
7.2.8. RS-232-C Queue In Pointer (Read Only, integer) REG:010 7-8
7.2.9. RS-232 Queue Out Pointer (Read Only, integer) REG:011 7-9
7.2.10. RS-232-C Status (Read Only, integer) REG:012 7-9
7.2.11. Archive Boot Status (Read Only, Integer) REG:013 7-9
7.2.12. Compiler Error Line Number (Read Only, integer) REG:014 7-10
7.2.13. Compiler Error (Read Only, integer) REG:015 7-10
7.2.14. Task 1 Run Time Error (Read Only, integer) REG:016 7-10
7.2.15. Task 2 Run Time Error (Read Only, integer) REG:017 7-10
7.2.16. System Library Access Error (Read Only, integer) REG:018 7-10
7.2.17. Encoder Counters 7-10
7.2.17.1. Output Status Registers (Read Only, Integer) REG:020 / REG:030 7-12
7.2.17.2. Output Latch Registers (Read Only, Integer) REG:021 / REG:031 7-13
7.2.17.3. Preset Registers (Write Only, Integer) REG:022 / REG:032 7-13
7.2.17.4. Master Control Regs (Write Only, Int.) REG:023 / REG:033 7-14

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.2.17.5. Input Control Regs (Write Only, Int.) REG:024 / REG:034 7-15
7.2.17.6. Output Control Regs (Write Only, Int) REG:025 / REG:035 7-16
7.2.17.7. Quadrature Regs (Write Only, Int) REG:026 / REG:036 7-17
7.2.18. R/D Counter Register (Read Only, integer) REG:038 7-17
7.2.19. IEEE-488 Cmd Pass Thru (Read Only, integer) REG:040 7-17
7.2.20. IEEE-488 Status Register 1 (Read Only, integer) REG:041 7-18
7.2.21. IEEE-488 Status Register 2 (Read Only, integer) REG:042 7-18
7.2.22. IEEE-488 Serial Poll Status (Read Only, integer) REG:043 7-18
7.2.23. IEEE-488 Address Status (Read Only, integer) REG:044 7-19
7.2.24. IEEE-488 Data In (Read Only, integer) REG:045 7-19
7.2.25. IEEE-488 Data Out (Write Only, integer) REG:046..... 7-19
7.3. Motion Registers 7-20
7.3.1. Encoder 1 Position (Read/Write, long) REG:102 7-20
7.3.2. Encoder 2 Position (Read/Write, long) REG:103 7-20
7.3.3. R/D Position (Read/Write, long) REG:104..... 7-20
7.3.4. ENC Position (Read/Write, long) REG:105..... 7-21
7.3.5. Encoder 1 Velocity (Read Only, integer) REG:108 7-21
7.3.6. Encoder 2 Velocity (Read Only, integer) REG:109 7-21
7.3.7. R/D Velocity (Read Only, integer) REG:110..... 7-21
7.3.8. ENC Velocity (Read Only, integer) REG:111 7-21
7.3.9. External Interrupt Position Latch (Read Only, long) REG:113 7-21
7.4. Drive Registers 7-22
7.4.1. Position Reset (Read/Write, long) REG:200..... 7-23
7.4.2. Position Reset Trigger (Write Only, integer) REG:201 7-23
7.4.3. Position Command (Read/Write, long) REG:202 7-23
7.4.4. Position Feedback (Read/Write, long) REG:203 7-23
7.4.5. Current Command (Read/Write, long) REG:204 7-23
7.4.6. Instantaneous Velocity Feedback (Read Only, long) REG:205 7-23
7.4.7. Averaged Velocity Command (Read Only, long) REG:206 7-23
7.4.8. Kernel Timeslice Mask (Read/Write, Integer) REG:207 7-23
7.4.9. Position Error (Read Only, long) REG:208 7-24
7.4.10. Averaged Velocity Feedback (Read Only, long) REG:209 7-24
7.4.11. Accel/Decel Feedback (Read Only, long) REG:210 7-24
7.4.12. Axis Calibrated Position Cmd (Read Only, long) REG:211 7-24
7.4.13. Axis Calibrated Position Feedback (Read Only, long) REG:212 7-24

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.4.14. Axis Calibration Compensation (Read Only, long) REG:213 ... 7-24
7.5. System Registers ... 7-25
7.5.1. Motion Status (Read Only, integer) REG:301 ... 7-25
7.5.2. Error Status (Read Only, integer) REG:302 ... 7-26
7.5.3. Axis Status (Read Only, integer) REG:303 ... 7-27
7.5.4. System Status (Read Only, integer) REG:304 ... 7-28
7.5.5. Error Status "Active Mask" (Read Only, integer) REG:307 ... 7-30
7.5.6. Error Status "Reset Mask"(Write Only, integer) REG:308 ... 7-30
7.5.7. Error Status "Overlay"(Write Only, integer) REG:309 ... 7-30
7.6. Variables ... 7-31
7.6.1. Indexing the Variables ... 7-31
7.6.2. Numerical Data Types ... 7-32
7.6.3. Integers (24 bit or 48 bit) ... 7-32
7.6.4. Floating Point ... 7-32

CHAPTER 8: SAMPLE PROGRAMS ... 8-1

8.1. Introduction ... 8-1
8.2. Trapezoidal Move ... 8-2
8.3. Trapezoidal Move with Output-on-the-Fly ... 8-2
8.4. Thumbwheel Input for Distance ... 8-3
8.5. Display Readout ... 8-4
8.6. Operator Input from HT ... 8-5
8.7. Multitasking ... 8-6
8.8. Data Logging On-the-Fly ... 8-8
8.9. Master/Slave Move ... 8-9
8.10. Proportional Joystick ... 8-10
8.11. Synchronized Trajectory Table Execution ... 8-11
8.12. Using External Interrupt to Change a Motion Trajectory ... 8-12
8.13. Initializing Counter Chip/Data Transfer from Preset Register ... 8-14
8.14. Transferring Data from the Counter to Output Latch ... 8-15
8.15. Using Comparator Outputs of the Encoder Counter Chip ... 8-16
8.16. Borrow and Carry Control Outputs of the Counter Chip ... 8-18
8.17. IEEE-488 Program Examples ... 8-20
8.17.1. HP85 Immediate Mode Command ... 8-20
8.17.2. HP85 Write and Read BV Variable ... 8-20
8.17.3. HP85 Retrieve Program from UNIDEX 100 ... 8-21
8.17.4. QB Immediate Mode Command ... 8-22
8.17.5. QB Write and Read BV Variable ... 8-23
8.17.6. Retrieve Program from UNIDEX 100 ... 8-24

CHAPTER 9: TECHNICAL DETAILS ... 9-1

9.1. U100 Control Board and Power Board ... 9-1
9.2. Motor Output Power (U100 Only) ... 9-4
9.2.1. DC Brush Motor Wiring ... 9-4
9.2.2. Stepping Motor Wiring ... 9-5
9.2.3. AC Brushless Motor Wiring ... 9-7
9.3. Limits and Primary Encoder Port (U100/U100i) ... 9-8

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9.3.1. Encoder Interface 9-9
9.3.2. Limits Interface 9-10
9.3.3. Hall Effect Interface 9-12
9.4. Communications Port (U100/U100i) 9-14
9.5. I/O Port 9-16
9.5.1. Digital Inputs Specifications 9-17
9.5.2. Digital Outputs Specifications 9-19
9.5.3. Analog Input and Output 9-20
9.6. Amplifier Connector (U100i) 9-20

CHAPTER 10: TROUBLESHOOTING 10-1

10.1. Installation, Startup and Communication Problems 10-2
10.2. Motor and Related Problems 10-3
10.3. Fault Conditions and Other Related Problems 10-5

CHAPTER 11: PID LOOP TUNING 11-1

11.1. Introduction 11-1
11.2. PID Gain Parameters 11-2
11.3. Tuning the PID Loop 11-3
11.4. Parameters and the Effect of Servo Loop Performance 11-7

CHAPTER 12: HOST MODE OPERATION 12-1

12.1. Introduction 12-1
12.2. Service Request 12-2
12.3. Host Mode Command Set 12-4
12.3.1. Running a "PGM:xx" File 12-5
12.3.2. Sending an Immediate Command 12-5
12.3.3. Making a Copy of a File 12-6
12.3.4. Erasing a File 12-7
12.3.5. Generating a File Directory 12-7
12.3.6. Transfer a File from the Host to the Controller 12-8
12.3.7. Transfer a File from the Controller to the Host 12-8
12.3.8. Modify the Value of a Parameter 12-9
12.3.9. Retrieve the Value of a Parameter 12-10
12.3.10. Modify the Value of a "Write Only" Register 12-11
12.3.11. Retrieve the Value of a "Read Only" Register 12-12
12.3.12. Modify the Value of a "Read/Write" Register 12-12
12.3.13. Retrieve the Value of a "Read/Write" Register 12-13
12.3.14. Modify the Value of a Variable 12-15
12.3.15. Retrieve the Value of a Variable 12-15
12.3.16. Modify the Value of a String Storage Buffer 12-16
12.3.17. Retrieve the Value of a String Storage Buffer 12-16
12.3.18. Move Out of a Limit 12-17
12.3.19. Task Control Functions 12-17

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

APPENDIX A: GLOSSARY OF TERMS A-1

APPENDIX B: WARRANTY AND FIELD SERVICE POLICY B-1

APPENDIX C: SPLINE GENERATION C-1

APPENDIX D: PROCESSING ERROR CODES D-1

D.1. Pre-Compiler Errors D-2
D.2. In-Process Compiler Errors D-2
D.3. Post-Compiler Errors D-3
D.4. Host Errors D-4
D.5. Run Time Errors D-6

INDEX

∇ ∇ ∇

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

LIST OF FIGURES

Figure 1-1. The UNIDEX 100 System Diagram 1-1
Figure 1-2. UNIDEX 100i System Diagram 1-2

Figure 2-1. Typical Interface Board 2-4
Figure 2-2. UNIDEX 100 System Configuration with Aerotech's Rotary Stage 2-5
Figure 2-3. UNIDEX 100 with Aerotech's Brushless Motor 2-6
Figure 2-4. UNIDEX 100i System Configuration 2-6
Figure 2-5. UNIDEX 100i & BA Amplifier with Aerotech's Brushless Motor 2-7
Figure 2-6. UNIDEX 100/U100I & BA Amplifier with Aerotech's BLM Linear Motor 2-7
Figure 2-7. RS-232-C Connection Diagram 2-8
Figure 2-8. Software Display Window 2-9
Figure 2-9. Main Menu Window (power applied to U100/U100i) 2-10
Figure 2-10. Main Menu Window 2-11
Figure 2-11. Main Menu Window 2-12
Figure 2-12. File Edit Window 2-12
Figure 2-13. Edit File Type Window 2-12
Figure 2-14. Enter File Number Window 2-13
Figure 2-15. Edit File Window 2-13
Figure 2-16. Finished Editing File Window 2-14
Figure 2-17. Select Task Window 2-14
Figure 2-18. Select Running Mode Window 2-14
Figure 2-19. Enter File Number Window 2-15
Figure 2-20. Status Selection Window 2-16
Figure 2-21. Motion Status Window 2-16
Figure 2-22. Control Selection Window 2-16
Figure 2-23. Task Window 2-17
Figure 2-24. Multitasking (Displayed on Screen while the Motor is Rotating) 2-18

Figure 3-1. UNIDEX 100 Controller 3-2
Figure 3-2. UNIDEX 100i Controller 3-3
Figure 3-3. Example System Using Velocity and Position Feedback 3-7
Figure 3-4. UNIDEX 100/U100i Control Board 3-8
Figure 3-5. UNIDEX 100 Power Board 3-13
Figure 3-6. Hand Held Terminal (HT) 3-14
Figure 3-7. HT Configuration (U100) 3-14
Figure 3-8. HT Configuration (U100i) 3-15
Figure 3-9. Optional UNIDEX 100/U100i Joystick 3-16
Figure 3-10. Optional Thumbwheel 3-17
Figure 3-11. Thumbwheel Interface 3-18
Figure 3-12. Optional LED Display 3-20
Figure 3-13. Display Configuration 3-20
Figure 3-14. Programmable Locations on LED Display 3-21
Figure 3-15. R/D Interface Board Connector Pinouts 3-23
Figure 3-16. Sample Uses of the PB# Boards 3-25
Figure 3-17. INT Board Jumper and Switch SW1 Locations 3-29
Figure 3-18. IEEE-488 Interface Card 3-30
Figure 3-19. Typical Installation of Optional Interface Card 3-33

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Figure 4-1. RS-232-C Connection Diagram 4-3
Figure 4-2. Software Display Window 4-4
Figure 4-3. Main Menu Window (power applied to U100/U100i) 4-5
Figure 4-4. Main Menu Window 4-5
Figure 4-5. File Edit Window 4-6
Figure 4-6. Edit File Type Window 4-6
Figure 4-7. Enter File Number Window 4-7
Figure 4-8. Edit File Window 4-7
Figure 4-9. File Directory Window 4-8
Figure 4-10. Select Task Window 4-9
Figure 4-11. Select Running Mode window 4-9
Figure 4-12. Select Task Window (MDI Mode) 4-10
Figure 4-13. Enter Command Window (MDI Mode) 4-10
Figure 4-14. Executing Command Window 4-11
Figure 4-15. Status Selection Window 4-11
Figure 4-16. Motion Status Window 4-12
Figure 4-17. General Status Window 4-12
Figure 4-18. Memory Available Window 4-13
Figure 4-19. Software Version Window 4-13
Figure 4-20. Control Select Window 4-13
Figure 4-21. Limit Reset Window 4-14
Figure 4-22. Task window 4-14
Figure 4-23. MISC Selection Window 4-15
Figure 4-24. Erase File(s) Window 4-15
Figure 4-25. File Type to Erase Selection Window 4-16
Figure 4-26. Erase All Files Selection Window 4-16
Figure 4-27. Copy File Type Selection Window 4-16
Figure 4-28. Setup Selection Window 4-17
Figure 4-29. Parameter Type Setup Window 4-17
Figure 4-30. Change Parameter Window (1 of 2) 4-18
Figure 4-31. Change Parameter Window (2 of 2) 4-18
Figure 4-32. Change Parameter Setting Window 4-18
Figure 4-33. Read/Write Selection Window 4-19
Figure 4-34. Write Only Register Change Window 4-19
Figure 4-35. Read Only Register Change Window 4-19
Figure 4-36. Load Register Window 4-20
Figure 4-37. Variable Type Selection Window 4-20
Figure 4-38. Enter Variable Window 4-20
Figure 4-39. Modify/Scan Select Window 4-21
Figure 4-40. Scan Variable Window 4-21
Figure 4-41. Enter String Window 4-21
Figure 4-42. Modify String Window 4-22
Figure 4-43. Enter New String Window 4-22
Figure 4-44. Transfer Select Window 4-23
Figure 4-45. Transfer File to U100 Window 4-23
Figure 4-46. Transfer File to PC Window 4-23
Figure 4-47. Read and Write from Archive Selection 4-24
Figure 4-48. Machine and Library Selection Menu 4-24
Figure 4-49. Enter Parameter (Machine and Library Image) 4-25
Figure 4-50. HT to U100/U100i Configuration 4-27

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Figure 4-51. HT Menu 4-27

Figure 5-1. Exception Processing Flow Chart 5-10
Figure 5-2. Example of Maximum Lines in PGM Files 5-17
Figure 5-3. Velocity Profiling in 3 Segments 5-43

Figure 6-1. 4-Axis Daisy Chain Configuration 6-8
Figure 6-2. 4-Axis Daisy Chain Configuration (U100i) 6-9
Figure 6-3. Trajectory Generation Overview 6-22
Figure 6-4. U100/U100i PID Loop 6-52
Figure 6-5. Decimal to Binary to Hexadecimal Conversion Chart 6-70

Figure 7-1. Connector P2 Bit to Input/Output for Registers 7-4
Figure 7-2. Encoder Counter Circuits (Primary and Auxiliary) 7-11
Figure 7-3. Block Diagram of Internal Counter Functions 7-12

Figure 9-1. Control Board 9-2
Figure 9-2. Power Board (U100 Controller Only) 9-3
Figure 9-3. DC Brush Motor System 9-4
Figure 9-4. DC Brush Motor Wiring 9-5
Figure 9-5. Stepping Motor System 9-6
Figure 9-6. Stepping Motor Wiring 9-6
Figure 9-7. AC Brushless Motor System 9-7
Figure 9-8. AC Brushless Motor Wiring 9-8
Figure 9-9. U100/U100i Servo Loop 9-8
Figure 9-10. Limits/Primary Encoder Port (P3) 9-9
Figure 9-11. Auxiliary Encoder Port (P4) 9-9
Figure 9-12. CW Motor Rotation (Viewed from Mounting Flange End) 9-10
Figure 9-13. Limit Switch Input Circuit 9-11
Figure 9-14. Motor Rotation 9-11
Figure 9-15. Hall Effect Input Circuit 9-12
Figure 9-16. Hall Effect and Motor Phasing 9-13
Figure 9-17. Motor Phase Voltage Observation Scheme 9-14
Figure 9-18. Encoder Phase Voltage Observation Scheme 9-14
Figure 9-19. Com Port (P1) 9-15
Figure 9-20. Standard RS-232-C Interface 9-15
Figure 9-21. Optional RS-422-A Interface 9-15
Figure 9-22. I/O Port (P2) 9-16
Figure 9-23. Electrical Characteristics of Opto-Isolated Input 9-18
Figure 9-24. Electrical Characteristics of Opto-Isolated Input (w/External Power Supply) 9-19
Figure 9-25. Electrical Characteristics of Opto-Isolated Output 9-19
Figure 9-26. Amplifier Connector Pinouts 9-20

Figure 11-1. U100/U100i PID Loop 11-1
Figure 11-2. Velocity Feedback Signals 11-6

Figure C-1. Diagram of a Typical Spline Profile C-2
Figure C-2. Trajectory Generation Overview C-4

∇ ∇ ∇

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

LIST OF TABLES

Table 1-1. Available Motion Controllers in the UNIDEX 100 Series 1-2
Table 1-2. Available Stepping Motor Drivers for UNIDEX 100 Series 1-3
Table 1-3. Available DC Servo Motor Drivers for UNIDEX 100 Series 1-3
Table 1-4. Available Brushless Servo Motor Drives for UNIDEX 100 Series 1-4
Table 1-5. Options and Accessories Available for the UNIDEX 100 Series 1-5
Table 2-1. Minimum Hardware Requirements and Recommendations 2-3
Table 3-1. Current/Velocity Command Jumper Setting 3-9
Table 3-2. Current Output Jumper Settings 3-10
Table 3-3. Motor Type Jumper Selections 3-11
Table 3-4. Firmware Boot Jumper Selections (JP34) 3-11
Table 3-5. Thumbwheel Bus Addresses 3-18
Table 3-6. Display Bus Addresses 3-21
Table 3-7. R/D Connector Pinout Descriptions 3-23
Table 3-8. R/D Standard Parameter Settings 3-23
Table 3-9. INT Bus Address Selections 3-26
Table 3-10. INT Board Jumper Selections 3-28
Table 3-11. IEEE-488 Connector P2 Pin Descriptions 3-30
Table 3-12. IE-488 Standard Parameter Settings 3-31
Table 4-1. Summary of Files on the Software Installation Diskette 4-2
Table 4-2. Minimum Hardware Requirements and Recommendations 4-3
Table 4-3. Keyboard Function Keys and their Function 4-8

Table 5-1. Programming Conventions Used in This Manual 5-2
Table 5-2. Kernel Interrupt Time Slot Description 5-4
Table 5-3. Bit Definitions for REG:302 5-11
Table 5-4. Bit Definitions for REG:307 5-13
Table 5-5. Task 0 Service Request Status Codes 5-14
Table 5-6. Task 1 Service Request Status Codes 5-15
Table 5-7. Task 2 Service Request Status Codes 5-15
Table 5-8. Kernel Service Request Status Codes 5-16
Table 5-9. PGM File Template 5-18
Table 5-10. UNIDEX 100/U100i Programming Commands 5-24

Table 6-1. Communication Parameters 6-3
Table 6-2. RS-232-C Bit Data 6-4
Table 6-3. Parameter Settings for Clock Control Register (PRM:002) 6-5
Table 6-4. Settings for Daisy Chain Enable/Disable (PRM:005) 6-6
Table 6-5. Settings for Communications Mode (PRM:019) 6-14
Table 6-6. Settings for Group Trigger Enable/Disable (PRM:020) 6-14
Table 6-7. Settings for Cursor Control (PRM:021) 6-15
Table 6-8. Settings for Communications SRQ On/Off (PRM:022) 6-15
Table 6-9. Settings for Send SRQ At End of Task 1 (PRM:023) 6-16
Table 6-10. Settings for Send SRQ At End of Task 2 (PRM:024) 6-16
Table 6-11. Settings for LST1 File Generation On/Off (PRM:025) 6-17
Table 6-12. Settings for Motion Status Scan Type (PRM:040) 6-21
Table 6-13. Settings for Motion Status Scan Type (PRM:041) 6-21
Table 6-14. Motion Parameters Summary 6-24

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 6-15. Settings for Command Segment Filter On/Off (PRM:102) 6-26
Table 6-16. Settings for "S" Curve Generation (PRM:108) 6-28
Table 6-17. Settings for In Position “Wait” (PRM:109) 6-28
Table 6-18. Settings for Home Switch Fast Reference (PRM:115) 6-29
Table 6-19. Settings for Home Switch Direction (PRM:116) 6-29
Table 6-20. Settings for Home switch Selection (PRM:117) 6-29
Table 6-21. Settings for Home Marker Selection 6-30
Table 6-22. Settings for Task1 Exception Processing (PRM:124) 6-31
Table 6-23. Settings for Task2 Exception Processing (PRM:125) 6-31
Table 6-24. Settings for Software Limit Reference (PRM:131) 6-32
Table 6-25. Settings for External Interrupt Latch Source (PRM:132) 6-33
Table 6-26. Settings for External Interrupt Mode (PRM:133) 6-34
Table 6-27. Settings for External Interrupt Type (PRM:134) 6-35
Table 6-28. Settings for External Interrupt Source (PRM:135) 6-35
Table 6-29. Settings for R/D Resolution Mode (PRM:137) 6-36
Table 6-30. Settings for Velocity Feedback Type (PRM:138) 6-36
Table 6-31. Settings for Position Feedback Type (PRM:139) 6-37
Table 6-32. Scaled Trajectory Task 1 Variable Type settings (PRM:144) 6-38
Table 6-33. Scaled Trajectory Task 2 Variable Type Settings (PRM:148) 6-40
Table 6-34. Spline Interpolation Mode Setting s PRM:152 6-41
Table 6-35. Spline Scaling Mode Settings (PRM:154) 6-42
Table 6-36. Spline Reference Mode Settings (PRM:155) 6-42
Table 6-37. Spline Modulo Mode Settings (PRM:161) 6-44
Table 6-38. Spline Output Mode Settings (PRM:162) 6-45
Table 6-39. Drive Parameters Summary 6-46
Table 6-40. Stepper Power Stage Mode Settings (PRM:200) 6-47
Table 6-41. Stepper Verification Enable/Disable Settings (PRM:204) 6-48
Table 6-42. Stepper Damping Enable/Disable Settings (PRM:207) 6-49
Table 6-43. Servo "Velocity Error" Integration Control (PRM:214) 6-50
Table 6-44. Servo Feedforward On/Off Settings (PRM:222) 6-52
Table 6-45. Gear Mode Selections (PRM:231) 6-54
Table 6-46. Gear Source Selections (PRM:232) 6-54
Table 6-47. Axis Calibration Mode Settings (PRM:234) 6-55
Table 6-48. Brushless Commutation Type Settings (PRM:239) 6-57
Table 6-49. Values for Parameter PRM:240 6-57
Table 6-50. System Parameters Summary 6-59
Table 6-51. Settings for Scaled Trajectory External Port Fetch (PRM:303) 6-61
Table 6-52. Settings for Servo Loop Update Time (PRM:304) 6-61
Table 6-53. Settings for Communication Loop Update Time (PRM:305) 6-62
Table 6-54. Settings for Stepper Loop Update Time (PRM:306) 6-62
Table 6-55. Settings for Encoder 1 Update Time (PRM:307) 6-63
Table 6-56. Settings for Encoder 2 Update Time (PRM:308) 6-64
Table 6-57. Settings for ENC Update Time (PRM:309) 6-65
Table 6-58. Settings for Resolver to Digital Update Time (PRM:310) 6-66
Table 6-59. Settings for Fault Mask Update Time (PRM:311) 6-66
Table 6-60. Settings for Control Loop Checks (PRM:312) 6-67
Table 6-61. Settings for Combine Task 1 and Task 2 Time Slots (PRM:313) 6-67
Table 6-62. Error Status Bit Definitions 6-69
Table 6-63. Error Status Latch Mask Bit Definitions 6-71
Table 6-64. Error Status Disable Mask Bit Definitions 6-72

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 6-65. Error Status Service Request Mask Bit Definitions 6-73
Table 6-66. Error Status Auxiliary Mask Bit Definitions 6-74
Table 6-67. Error Status Freeze Mask Bit Definitions 6-75
Table 6-68. Error Status Task 1 Mask Bit Definitions 6-76
Table 6-69. Error Status Latch Mask Bit Definitions 6-77
Table 6-70. Error Status "+" Direction Mask Bit Definitions 6-78
Table 6-71. Error Status "–" Direction Mask Bit Definitions 6-79

Table 7-1. Communication Registers Summary 7-3
Table 7-2. Bit Input for Register:001 7-4
Table 7-3. Bit Output for Register:002 7-4
Table 7-4. Bit Definitions for REG:003 7-5
Table 7-5. Bit Definitions for REG:004 7-6
Table 7-6. Bit Definitions for REG:005 7-7
Table 7-7. Bit Definitions for REG:007 7-8
Table 7-8. RS-232-C Status Register Bit Definitions 7-9
Table 7-9. Bit Definitions for REG:013 7-9
Table 7-10. Bit Definitions for Output Status Registers 7-13
Table 7-11. Bit Definitions for Master Control Registers 7-14
Table 7-12. Bit Definitions for Input Control Registers 7-15
Table 7-13. Bit Definitions for Output Control Registers 7-16
Table 7-14. Bit Definitions for Quadrature Registers 7-17
Table 7-15. Bit Definitions for REG:041 7-18
Table 7-16. Bit Definitions for REG:042 7-18
Table 7-17. Bit Definitions for REG:044 7-19
Table 7-18. Motion Registers Summary 7-20
Table 7-19. Drive Registers Summary 7-22
Table 7-20. Bit Definitions for REG:207 7-24
Table 7-21. System Registers Summary 7-25
Table 7-22. Bit Definitions for REG:301 7-25
Table 7-23. Bit Definitions for REG:302 7-26
Table 7-24. Bit Definitions for REG:303 7-27
Table 7-25. Bit Definitions for REG:304 7-28
Table 7-26. Bit Definitions for REG:307 7-30

Table 9-1. Available I/O Port Functions 9-16

Table 10-1. Troubleshooting for Common Installation, Startup, and Communication Problems 10-2
Table 10-2. Troubleshooting for Motors, and Related Problems 10-3
Table 10-3. Troubleshooting for Fault Conditions and Other Related Problems 10-5

Table 11-1. PID Loop Individual Element Descriptions 11-2

Table 12-1. Task 0 Service Request Status Codes 12-2
Table 12-2. Task 1 Service Request Status Codes 12-3
Table 12-3. Task 2 Service Request Status Codes 12-3
Table 12-4. Kernel Service Request Status Codes 12-4

Table D-1. Pre-Compiler Error Codes D-2

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table D-2. In-Process Compiler Error Codes D-2
Table D-3. Post-Compiler Error Codes D-3
Table D-4. Host Errors Codes D-4
Table D-5. Host Errors Codes [Library Utility Mode Only] D-5
Table D-6. Run Time Error Codes D-6
Table D-7. Run Time Error Codes [RUN ( ) Command Only] D-8

∇ ∇ ∇

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

DECLARATION OF CONFORMITY

Manufacturer's Name and Address

Aerotech, Inc.
101 Zeta Drive
Pittsburgh, PA 15238-2897

Declares that the product:

Product Name: UNIDEX 100 Single Axis Controller

Conforms to the following product specifications :

EMC: EN 55011: 1991 Class B Emissions
EN 50082-1: 1992 Immunity
IEC 801-2: 1984
IEC 801-3: 1984
IEC 801-4: 1988
LVD: IEC 204-1

and complies with EMC directive 89/336/EEC.

Pittsburgh, PA
August, 1998

David F. Kincel
Quality Assurance Manager

Robert Novotnak
Engineer Verifying Compliance

General notes concerning the test setup.

This product was tested at Washington Laboratories, LTD. in Gaithersburgh, MD on October 19, 1995. The report number is WLL 2939F.

To ensure that the UNIDEX 100 conforms to the conducted emissions standards, a Schaffner FN 2070-10-06 line filter must be connected to the AC mains. To ensure that the UNIDEX 100 conforms to the radiated emissions standards, ferrite must be added (in common mode) to the motor leads and the shield of the motor cable must be tied to earth ground. Failure to follow the described configuration may cause the controller to exceed emission limits.

∇ ∇ ∇

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PREFACE

This section provides an overview of topics covered and conventions used in each section of this manual. This manual contains information on the following topics:

CHAPTER 1: OVERVIEW

Chapter 1 contains an overview of the UNIDEX 100/U100i motion controller as well as sample system diagrams. Included are listings of the available motors that the U100/U100i can drive and the available options and accessories. This chapter also contains precautionary notes about installing and using the UNIDEX 100/U100i motion controllers.

CHAPTER 2: GETTING STARTED

Chapter 2 contains information about the components of the UNIDEX 100/U100i system, unpacking and inspecting the equipment, and minimum hardware and software requirements for proper operation. In addition, Chapter 2 walks the user through physically setting up the motion controller(s), connecting the signal cables, installing the software, and using the software for a simple motion applications.

CHAPTER 3: HARDWARE CONFIGURATION & DESCRIPTION

Sections in this chapter provide the user with information about the hardware of the UNIDEX 100/U100i system. This includes a discussion of the hardware components and individual optional configurations.

CHAPTER 4: SOFTWARE INSTALLATION AND INTERFACE

This chapter contains information about the UT100 Interface software that is supplied with the UNIDEX 100/U100i system. Included is an explanation of how to install the Interface software for proper operation, starting the Interface software and configuring the software for proper operation. Also, a discussion covering the complete list of menu items and options provided through the software interface. Sample screens are illustrated for all options. Provided is an explanation of the operational interface between the Hand held Terminal (HT) and the controller.

CHAPTER 5: PROGRAMMING COMMANDS

Chapter 5 supplies information required to understand the UNIDEX 100/U100i programming environment. Some covered topics are multi-tasking, interrupts, and exception processing. This chapter also includes an in-depth discussion of individual programming commands.

CHAPTER 6: PARAMETERS

This chapter provides information that helps the operator to understand and configure the parameters within the UNIDEX 100/U100i system. Appropriate parameter configuration optimizes the UNIDEX 100/U100i for an application. This chapter includes detailed discussions of all software parameters, motor and feedback configurations, and other topics related to the operation and configuration of the UNIDEX 100/U100i system.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CHAPTER 7: REGISTERS AND VARIABLES

Information presented in Chapter 7 helps the operator to understand the use of controller registers and variables. Provided is a detailed explanation of each register and its use.

CHAPTER 8: SAMPLE PROGRAMS

Chapter 8 contains sample applications, highlighting UNIDEX 100/U100i features, parameter settings and sample programs.

CHAPTER 9: TECHNICAL DETAILS

This chapter supplies a variety of technical specifications for the UNIDEX 100/U100i. These specifications include encoder signal specifications, pinouts, outputs, bus specifications, motor power connections and configurations, and others.

CHAPTER 10: TROUBLESHOOTING

Chapter 10 provides a reference tool if problems with the UNIDEX 100/U100i arise.

CHAPTER 11: PID LOOP TUNING

Information in Chapter 11 explains the proper technique for tuning the controller PID loop with the parameters and registers that affect this process.

CHAPTER 12: HOST MODE OPERATION

This chapter introduces the C-based functions that allow the operator to create a customized operator interface, rather than using the UT00 software.

APPENDIX A: GLOSSARY OF TERMS

Appendix A contains a list of definitions of terms used in this manual.

APPENDIX B: WARRANTY AND FIELD SERVICE

Appendix B contains the warranty and field service policy for Aerotech products.

APPENDIX C: SPLINE GENERATION

Appendix C contains a detailed explanation of the spline generation mode initiated by the CAM() command.

APPENDIX D: PROCESSING ERROR CODES

Appendix D contains a reference of all UNIDEX 100/U100i processing error codes.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

INDEX

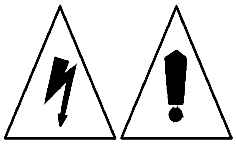
The index contains a page number reference of topics discussed in this manual. Locator page references in the index contain the chapter number (or appendix letter) followed by the page number of the reference. Locator page numbers may appear in one of four possible styles: standard serif font (e.g., 3-1) for normal references, boldface font (e.g., **3-1**) for references to illustrations, italic font (e.g., *3-1*) for references to tables, or boldface italic font (e.g., ***3-1***) for references to illustrations within tables.

CUSTOMER SURVEY FORM

A customer survey form is included at the end of this manual for the reader's comments and suggestions about this manual. Reader's are encouraged to critique the manual and offer their feedback by completing the form and either mailing or faxing it to Aerotech.

Throughout this manual the following conventions are used:

- Capitalized letters within a command indicate the command must use all capital letters (e.g., BEGIN)
- The terms UNIDEX 100, UNIDEX 100i, and U100/U100i are used interchangeably throughout this manual
- *Italic font* is used to illustrate syntax and arguments for programming commands
- Keys such as Shift, Ctrl, Alt and Enter are enclosed in brackets (e.g., <Shift>, <Ctrl>, <Alt> and <Enter>) to distinguish them from individual keystrokes
- The text <Enter> and the symbol ↵ are used interchangeably throughout this manual to indicate that the Enter/Return key on the keyboard is to be pressed
- Hexadecimal numbers are listed using a preceding "0x" (for example, 0x300, 0x12F, 0x01EA, etc.,) to distinguish them from decimal numbers
- The terms <Enter> key and <Return> key are used interchangeably throughout this document when referring to the keyboard
- Graphic icons or keywords may appear in the outer margins to provide visual references of key features, components, operations or notes.
- Danger and/or Warning symbols (see right) appear in the outer margins next to important precautions. Failure to observe these precautions could result in serious injury and/or damage to the equipment
- The following statements apply wherever a Warning or Danger symbol appears within this manual. Failure to observe these precautions could result in serious injury to those performing the procedures and/or damage to the equipment



To minimize the possibility of electrical shock and bodily injury, make certain that all of the electrical power switches are in the off position prior to making any electrical connections.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

To minimize the possibility of electrical shock and bodily injury when any electrical circuit is in use, ensure that no person comes in contact with the circuitry.

When this controller is installed within a system, mechanical motion will occur. Care must be exercised that all personnel remain clear of any moving parts.

To minimize the possibility of bodily injury, make certain that all electrical power switches are in the off position prior to making any mechanical adjustments.

- This manual uses the symbol "∇ ∇ ∇" to indicate the end of a chapter

Although every effort has been made to ensure consistency, subtle differences may exist between the illustrations in this manual and the component and/or software screens that they represent.

∇ ∇ ∇

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CHAPTER 1: INTRODUCTION

In This Section:

- Overview of the UNIDEX 100/U100i System 1-1
- Ordering Information..... 1-2
- Options and Accessories..... 1-5
- Safety Procedures and Warnings 1-6

1.1. Overview of the UNIDEX 100/U100i System

The UNIDEX 100 and UNIDEX 100i systems are a combination of the motion controller, the UT100 interface software, and any number of optional accessories. The UNIDEX 100 controller package includes control board, driver, and power supply. The UNIDEX 100i controller package consists only of the control board. The controller's versatility provides features such as variables, math and multitasking that make it ideal for tough automation projects. The U100i integrates with Aerotech's BA Series amplifiers or third party amplifiers providing the user more flexibility for motion control applications. Both controllers integrate with stepping, DC servo, and brushless motors, positioning stages, and these accessories to form a complete, programmable, customized control system that is suitable for a wide range of motion control applications. A typical system using the U100 is shown in Figure 1-1. Figure 1-2 shows a typical U100i system configuration.

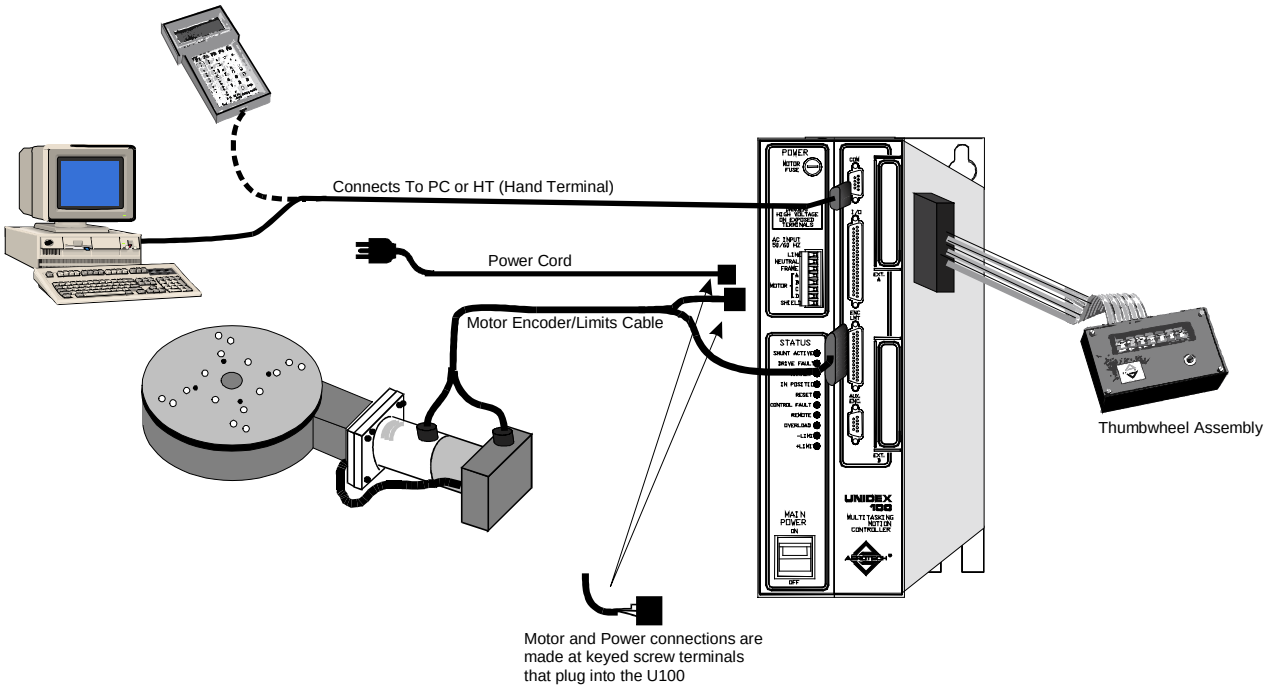


Figure 1-1. The UNIDEX 100 System Diagram

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

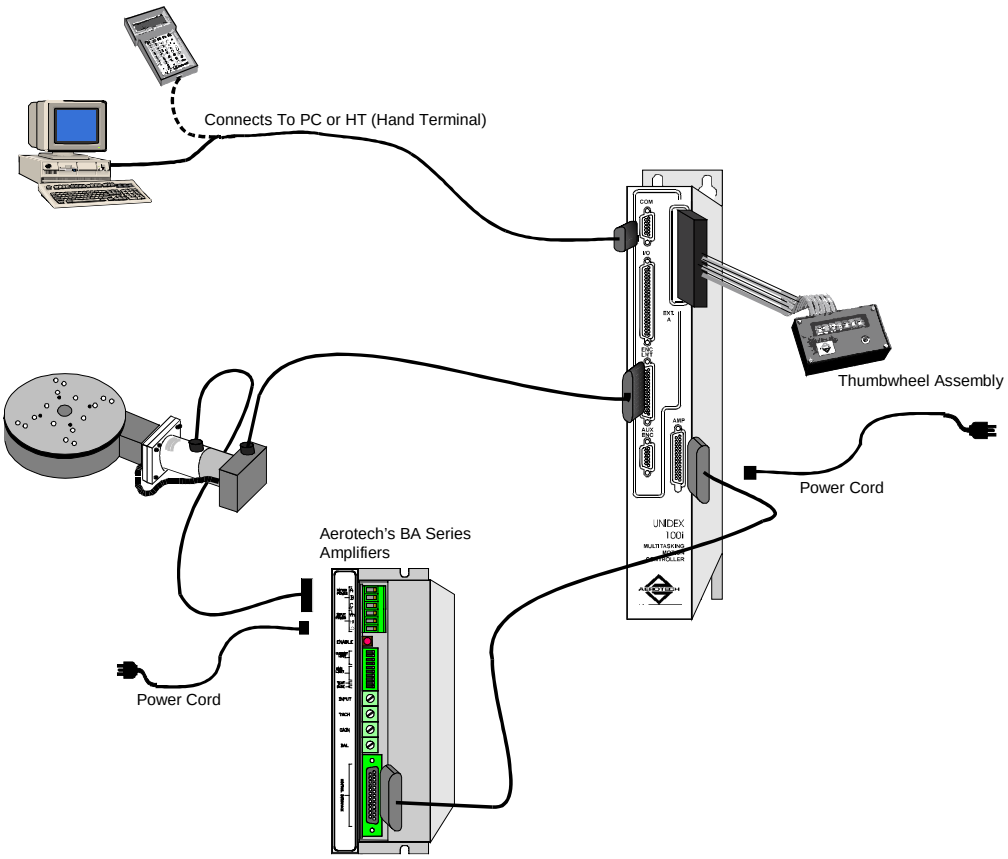


Figure 1-2. UNIDEX 100i System Diagram

1.2. Ordering Information

Table 1-1 lists the motion control models in the 100 series that are available from Aerotech, Inc. For complete ordering information, refer to the *Aerotech Motion Control Catalog*.

Table 1-1. Available Motion Controllers in the UNIDEX 100 Series

Controller	Description
U100M/	Sub-panel mount package with microstepping drive
U101M/	Open frame package with microstepping drive
U100S/	Sub-panel mount package with DC servo drive
U101S/	Open frame package with DC servo drive
U100Z/	Sub-panel mount package with brushless servo drive
U101Z/	Open frame package with brushless servo drive

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 1-2 lists the available Aerotech stepping motor drivers that can be used with the UNIDEX 100 series motion controllers. The available DC servo and AC brushless drivers are listed in Tables 1-3 and 1-4, respectively.

Table 1-2. Available Stepping Motor Drivers for UNIDEX 100 Series

Motor Model	Continuous Torque oz-in (N-m)	Peak Torque oz-in (N-m)	Speed RPM	Motor Output Power Watts	Inertia oz-in-sec^2 (kg-m^2)	Motor Weight lbs (kg)	Bus Voltage VDC(2) (nom.)	Fuse Rating Amps
ATS50, ATS100, ART50, and ART100 Stages	N/A	N/A	N/A	N/A	N/A	N/A	40	0.5
50SMB2	38 (0.3)	N/A	900	12	$1.7x10^{-3}(11,8x10^{-6})$	1.8 (0.8)	40	1
101SMB2	55(0.4)	N/A	1500	40	$1.42x10^{-3}(10x10^{-6})$	1.5 (0.7)	160(1)	0.8
55SMP	90 (0.7)	N/A	2200	53	$5x10^{-3}(35x10^{-6})$	3.3 (1.5)	40	5
140SMP	140 (1.0)	N/A	2200	90	$5x10^{-3}(35x10^{-6})$	3.1 (1.4)	160(1)	1.4
310SMB3	370 (2.6)	N/A	2200	250	$27x10^{-3}(187x10^{-6})$	8.3 (3.8)	80	6
450SMP	450 (3.2)	N/A	2200	240	$27x10^{-3}(187x10^{-6})$	8 (3.6)	160(1)	3.5
1010SMB4	1050 (7.4)	N/A	1500	380	$114x10^{-3}(805x10^{-6})$	20.5 (9.3)	160(1)	8

Table 1-3. Available DC Servo Motor Drivers for UNIDEX 100 Series

Motor Model	Continuous Torque oz-in (N-m)	Peak Torque oz-in (N-m)	Speed RPM	Motor Output Power Watts	Inertia oz-in-sec^2 (kg-m2)	Motor Weight lbs (kg)	Bus Voltage VDC(2) (nom.)	Fuse Rating Amps
1035LT-MSO1/ E500LD/	35 (0.3)	170 (1.2)	5000	90	$5x10^{-3}(38x10^{-6})$	3.3 (1.5)	40	4
1050LT-MSO1/ E500LD/	50 (0.4)	186 (1.3)	4200	115	$8x10^{-3}(57x10^{-6})$	3.8 (1.7)	40	5
1075LT-MSO1/ E500LD	75 (0.5)	280 (2.0)	5000	140	$23x10^{-3}(160x10^{-6})$	5.8 (2.6)	80	5
1135LT-MSO1/ E500LD	135 (1.0)	490 (3.5)	3600	200	$52x10^{-3}(360x10^{-6})$	10.3 (4.7)	80	5
1210LT-MSO1/ E500LD/	210 (1.5)	700 (4.9)	2500	220	$130x10^{-3}(920x10^{-6})$	10.0 (4.5)	80	6
1960LT-MSO1/ E500LD/	800 (5.6)	1600 (11.3)	2400	780	$370x10^{-3}(2600x10^{-6})$	38 (17.3)	160(1)	10

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 1-4. Available Brushless Servo Motor Drives for UNIDEX 100 Series

Motor Model	Continuous Torque oz-in (N-M\m)	Peak Torque oz-in (N-m)	Speed RPM (max)	Motor Output Power Watts	Inertia oz-in-sec^2 (kg-m2)	Motor Weight lbs (kg)	Bus Voltage VDC(2) (nom.)
BM75	75 (0.53)	200 (1.141)	10,000	210	0.0007 ($0.52x10^{-5}$)	2.5 (1.1)	160
BM130	140 (1.0)	350 (2.5)	10,000	290	0.0013 ($0.92x10^{-5}$)	3.3 (1.5)	160
BM200	200 (1.41)	500 (3.5)	8,000	450	0.0019 ($1.3x10^{-5}$)	4.3 (2.0)	160
BM250	260 (1.85)	650 (4.6)	8,000	560	0.011 ($7.8x10^{-5}$)	8 (3.6)	160
BM500	510 (3.6)	1275 (9.0)	8,000	1100	0.020 ($13.9x10^{-5}$)	11 (5.0)	160
BM800	780 (5.5)	2000 (14)	6,000	1400	0.042 ($30x10^{-5}$)	14.5 (6.6)	320
BM1400	1365 (9.6)	3400 (24)	6,000	2330	0.080 ($56x10^{-5}$)	23.5 (10.7)	320

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

1.3. Options and Accessories

A variety of options and accessories may be purchased with the UNIDEX 100/U100i to enhance its standard operation. Table 1-5 lists the Aerotech options and accessories that can be used with the UNIDEX 100 Series motion controllers. Brief descriptions of each option follow. These items are explained in Chapter 3: Hardware Configurations and Descriptions.

Table 1-5. Options and Accessories Available for the UNIDEX 100 Series

Options/Accessories	Description
HT/	Hand held terminal
JOY/	Joystick with digitizing capability
THM/	6 digit plus sign thumbwheel assembly with 10 ft. cable and interface card
DISP/	6 digit plus sign LED display with 10 ft. cable and interface card
TS/	37 point screw terminal strip with 3 ft. cable
GDS100/	Graphical development software with PC cable and user's manual
IE488/	IEEE 488 interface card
INT/	Interface card to one Opto 22 PB8 with 3 ft. cable
RD/	Resolver/Inductosyn to digital converter card
OAC5A/	AC output module, 24 to 280 VAC, 2A
IAC5/	AC input module, 90 to 140 VAC
ODC5/	DC output module, 5 to 60 VDC, 2A
IDC5/	DC input module, 10 to 32 VDC
IDC5B/	DC input module, 4 to 6 VDC
AF5/	EMI filter module, in-line enclosure for DC servo models with 1-5A continuous current
AF6/	EMI filter module, in-line enclosure for DC servo models with 6-10A continuous current
TV0.3/	0.3 KVA auto transformer for U101, 115VAC
TV0.4/	0.4 KVA auto transformer for U101, 230 VAC

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

1.4. Safety Procedures and Warnings

The following statements apply wherever the Warning or Danger symbol appears within this manual. Failure to observe these precautions could result in serious injury to those performing the procedures and/or damage to the equipment.



To minimize the possibility of electrical shock and bodily injury, make certain that all of the electrical power switches are in the off position prior to making any electrical connections.



To minimize the possibility of electrical shock and bodily injury when any electrical circuit is in use, ensure that no person comes in contact with the circuitry.



When this controller is installed within a system, mechanical motion will occur. Care must be exercised that all personnel remain clear of any moving parts.



To minimize the possibility of bodily injury, make certain that all electrical power switches are in the off position prior to making any mechanical adjustments.

∇ ∇ ∇

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CHAPTER 2: GETTING STARTED

In This Section:

- Introduction 2-1
- Unpacking the UNIDEX 100/U100i System 2-1
- Minimum Hardware Requirements 2-2
- Recommended System Configurations 2-2
- Inspection of the UNIDEX 100/U100i 2-3
- Physical Setup of the Motion Controller 2-5
- Standard Installation 2-8
- Simple Program Applications 2-11

2.1. Introduction

Chapter 2 contains information about the components of the UNIDEX 100/U100i system, unpacking and inspecting the equipment, and minimum hardware and software requirements for proper operation. In addition, this chapter walks the user through physically setting up the motion controller(s), connecting the signal cables, installing the software, and using the software for a simple motion applications.

This chapter is designed for technically oriented individuals familiar with computers and software. Those individuals that need detailed explanations about other features, techniques, and procedures should review Chapter 3 through Chapter 12.
Also, the information in this chapter is just one example to help the user to become familiar with the U100/U100i Motion Control System. It should not be regarded as the only configuration or use of the system.



2.2. Unpacking the UNIDEX 100/U100i System

Before unpacking any components, visually inspect the containers of the U100/U100i system for any evidence of shipping damage. If any such damage exists, notify the shipping carrier immediately.

All electronic equipment is wrapped in antistatic material and packaged with desiccant (a drying agent used to reduce moisture). Make certain that the antistatic material is not damaged during unpacking.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

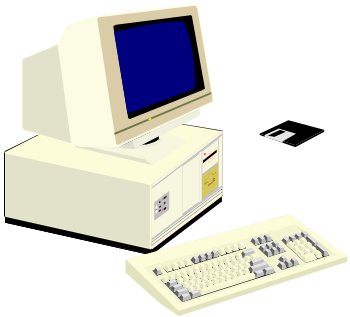
Remove the packing list from the controller container. Make certain that the items listed on the packing slip are contained within the package. The following items should be found in every UNIDEX 100 series system:

- The Motion Controller
- UNIDEX 100/U100i Motion Controller Operation & Technical Manual
- UT100 software (on one double-sided, high-density floppy diskette)
- Controller packing slip (listing products shipped with the order)
- Controller screwdriver (Phillips and blade)

The following list of additional items may be included with the UNIDEX 100 series system, depending on the options and accessories that have been specified:

- Hand held terminal (HT)
- Thumbwheel assembly with cable
- LED display with cable
- Joystick and cable (with digitizing capability)
- Terminal strip with cable
- GDS100 Graphical Development Software
- IEEE 488 interface card (installed)
- Resolver/Inductosyn converter card (installed)

2.3. Minimum Hardware Requirements and Recommended System Configurations



The PC into which the software is installed must meet certain minimum requirements for proper operation of the system. These requirements include the type of computer (i.e., the type of microprocessor it uses), the amount of memory it contains, the graphics display, available hard disk space, floppy disk drives and an operating system. The minimum PC requirements are listed in Table 2-1.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 2-1. Minimum Hardware Requirements and Recommendations

Equipment	Minimum	Recommended
Computer (microprocessor)	IBM PC AT or PS/2 80486 (or higher) or 100% compatible	80486 or higher
Computer Memory	4 MB of memory (conventional & extended)	4 MB memory (conventional & extended)
Graphics Display	EGA/VGA	EGA/VGA
Hard Disk Space	4 MB	5 MB or more
Mouse	Any mouse supported by the computer	Any mouse supported by the computer
Floppy Disk Drives	3 1/2" DSHD	3 1/2" DSHD
Windows (for GDS100 software)	Windows 3.1 or higher	Windows 3.1 or higher
DOS	5.0 or higher	5.0 or higher

2.4. Inspection of the UNIDEX 100/U100i

All products undergo a total quality inspection before they are shipped from Aerotech. Even with a stringent quality assurance program, however, it is still possible that a product may arrive in less than perfect condition due to improper handling during shipment. After unpacking the controller, check to ensure that there are no visible signs of damage.

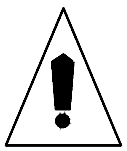
2.5. Inspection of Interface Control Cards

Before touching any of the controller interface cards, be sure to observe the electrostatic discharge precautions that are listed below.

Inspection of interface cards is only necessary if the user desires installing additional options instead of returning the unit for factory installation.



The U100 series interface cards are sensitive to static electricity. To greatly reduce the possibility of board damage due to electrostatic discharge, adhere to the following precautions.



WARNING

1. Do not remove an interface card from the antistatic bag until it is ready to be installed. When removing a card from a system, immediately place the card in an antistatic bag.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2. Make certain that anyone who is handling the board (or any associated components) is wearing a properly grounded static strap.
3. When handling an interface card, hold the card by its edges and the mounting brackets. Avoid touching board components and the edge connectors that plug into the expansion slots.
4. Do not slide the interface card over any surface.
5. Avoid plastic, Styrofoam or vinyl in the work area.
6. Static charge buildup may be removed from an object by touching the object to a properly grounded piece of metal.

The U100 series interface cards were tested and inspected before being shipped from Aerotech, Inc. Vibration during shipment, however, may have loosened certain board components.

Immediately prior to installing an interface card into the controller, visually inspect the card. Make certain that all socketed ICs are firmly seated in their sockets. If a chip has become loose, carefully reinstall it into its socket. Be sure to observe the proper antistatic precautions mentioned above. A typical interface board is illustrated in Figure 2-1.

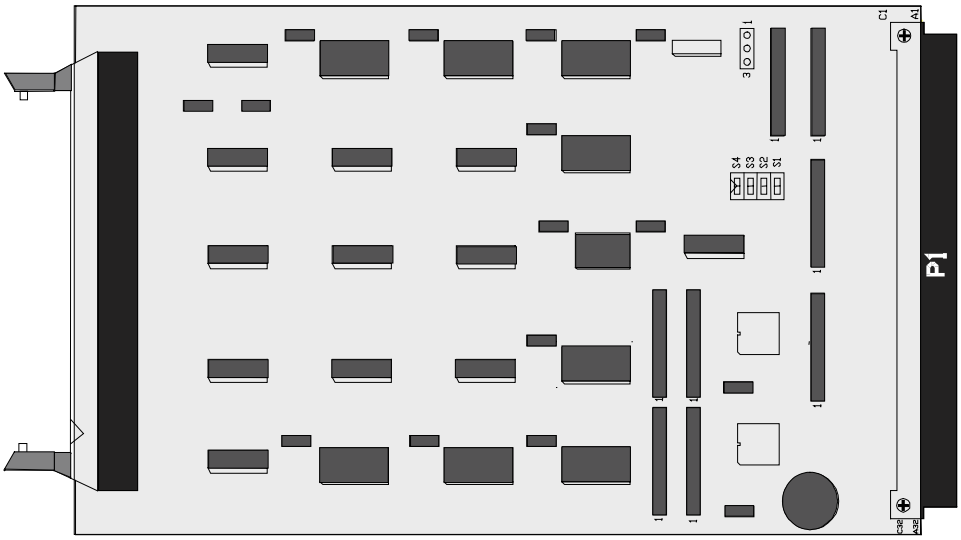


Figure 2-1. Typical Interface Board

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.6. Physical Setup of the Motion Controller

This section explains how to setup the motion controller to a PC or Hand held Terminal (HT) , stage or applicable amplifier (if using the U100i) and install the system software. Also, a simple application is provided so the user can become familiar with using the motion control system.

The electrical signal connections between the U100/U100i and Aerotech's rotary stages, brushless amplifiers, brushless motors, and linear motors are shown in Figure 2-2, Figure 2-3, Figure 2-4, Figure 2-5, and Figure 2-6.

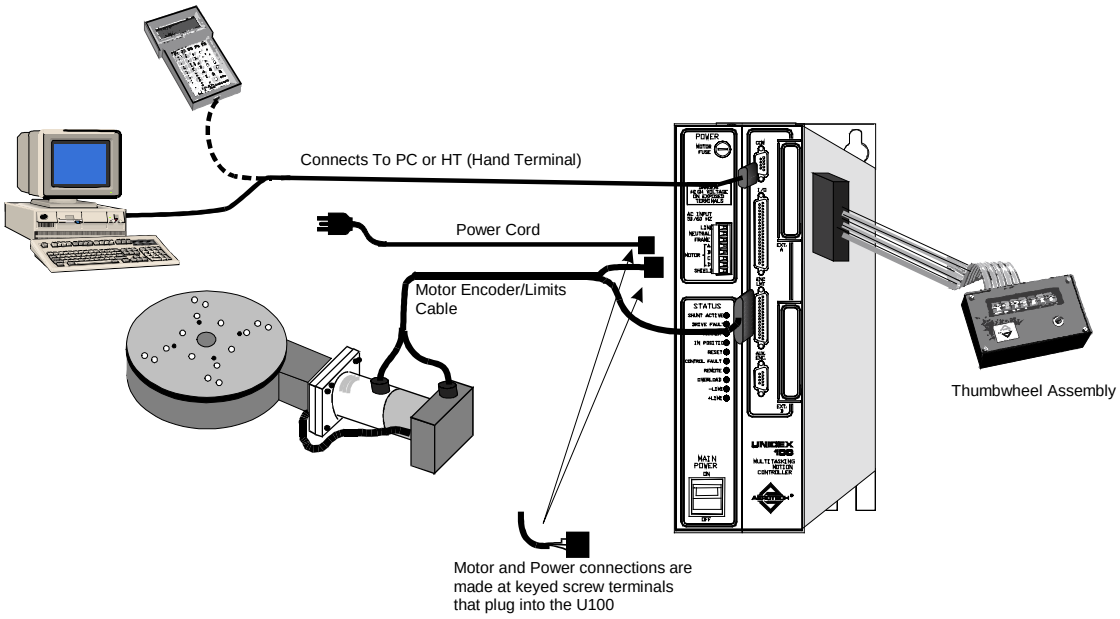


Figure 2-2. UNIDEX 100 System Configuration with Aerotech's Rotary Stage

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

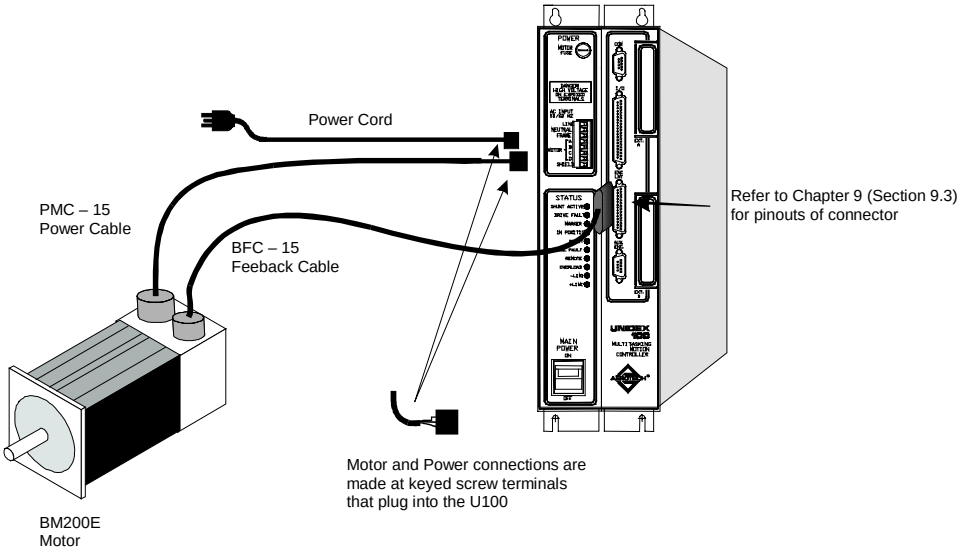


Figure 2-3. UNIDEX 100 with Aerotech's Brushless Motor

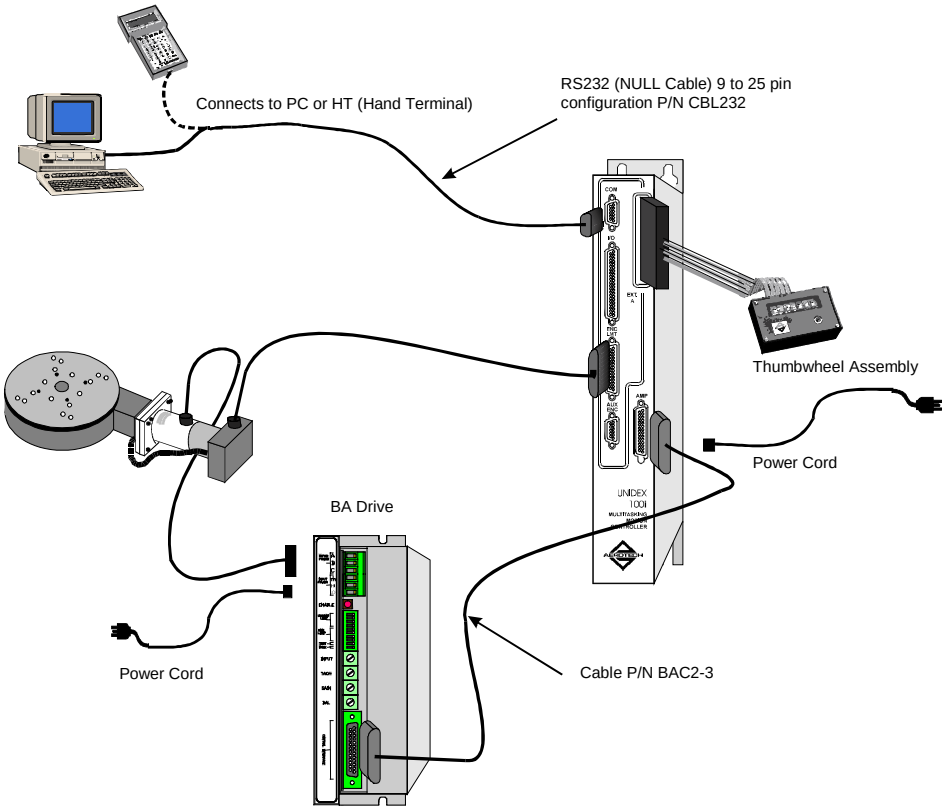


Figure 2-4. UNIDEX 100i System Configuration

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

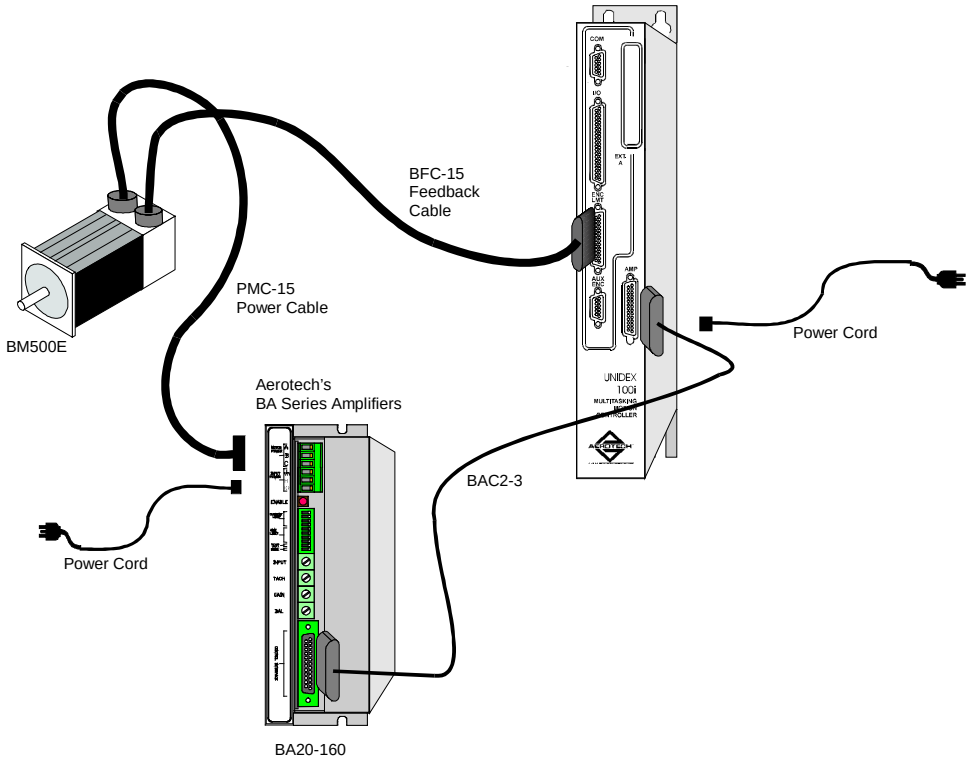


Figure 2-5. UNIDEX 100i & BA Amplifier with Aerotech's Brushless Motor

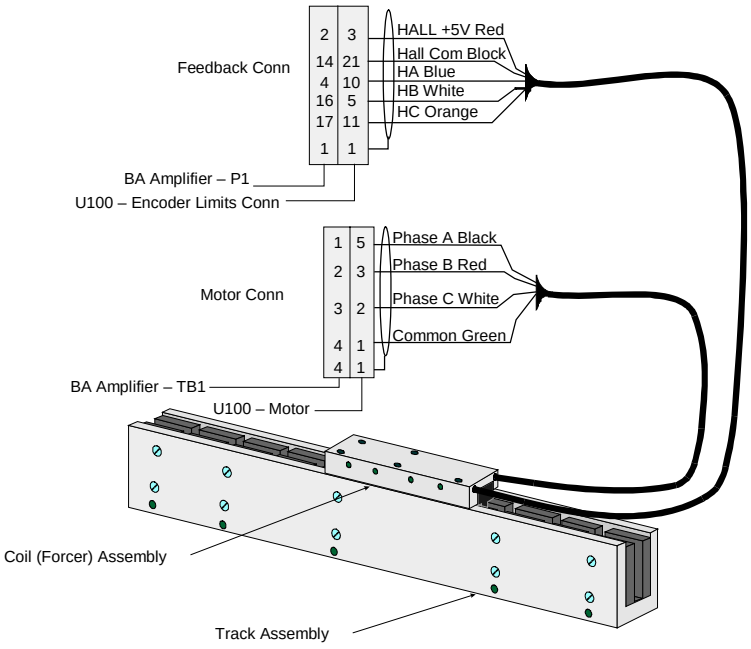


Figure 2-6. UNIDEX 100/U100I & BA Amplifier with Aerotech's BLM Linear Motor

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.6.1. Standard Installation

To install the software, the operator must follow the steps listed below.

1. Connect the RS-232-C cable between COM1 or COM2 on the PC and the COM port (P1) on the controller, Refer to Figure 2-7.

Since the UNIDEX 100/U100i are Data Terminal Equipment (DTE) devices, the user must connect to the com ports of controller and PC with a NULL modem cable or adapter when using the 9 pin configuration. This will ensure proper connection of the transmit (TX) and receive (RX) signals. Refer to Figure 2-7.

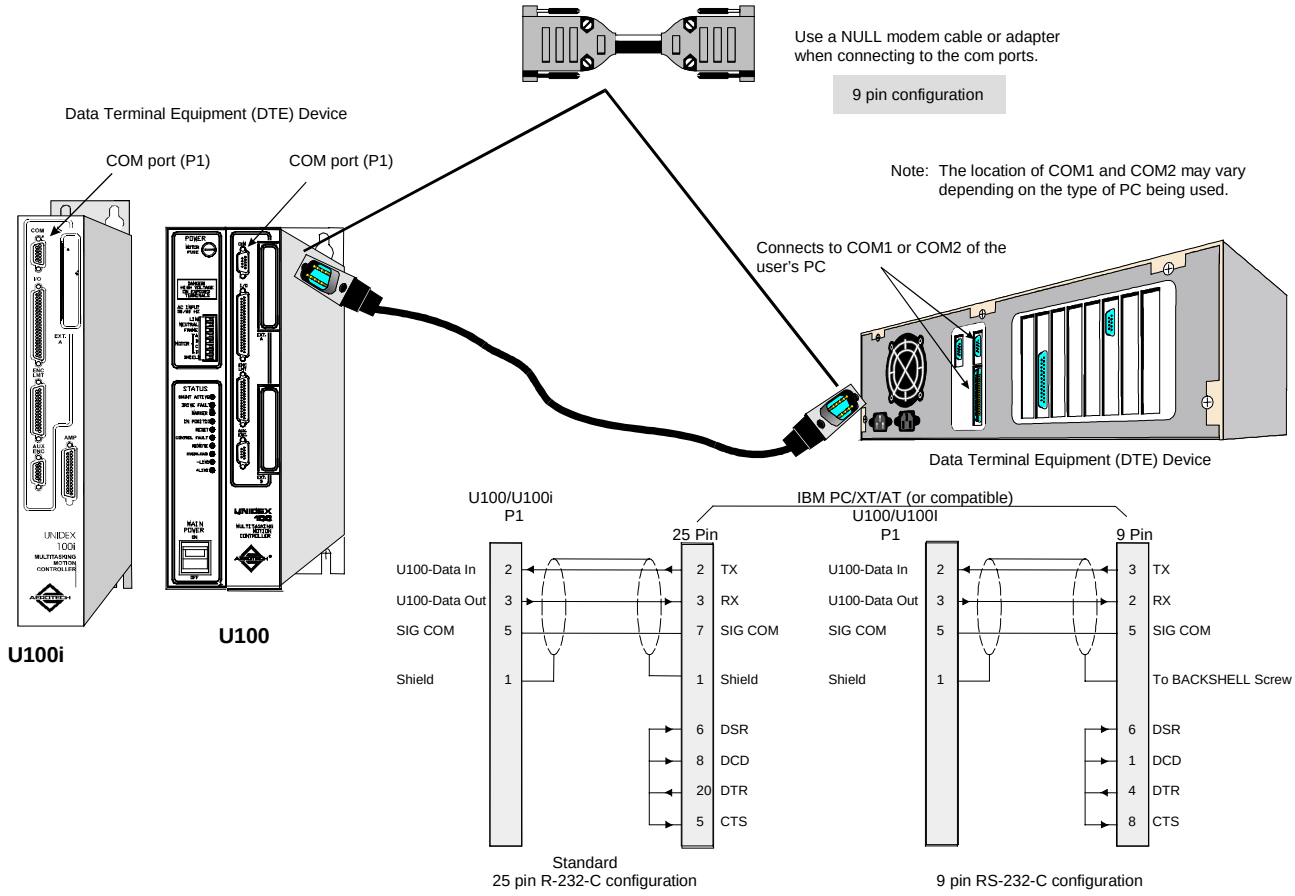


Figure 2-7. RS-232-C Connection Diagram

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2. Using an editor, add the following command to the CONFIG.SYS file.

   For MSDOS operation, add the following line to the DOS CONFIG.SYS configuration file.

   DEVICE=<directory>\ANSI.SYS

   Where <directory> denotes the location of the ANSI.SYS file.

   For Win95 operation, add this line to the Win95 CONFIG.SYS configuration file.

   DEVICE=C:\WIN95\COMMAND\ANSI.SYS

   For WinNT operation, add this line to the WinNT CONFIG.NT configuration file.

   DEVICE=<directory>\system\ansi.sys

   Where <directory> denotes the location of the ANSI.SYS file.

3. Insert the UT100 disk into the 3.5” floppy disk (e.g., drive A:\).

4. Copy all files from the floppy disk drive to the hard drive. For example:

   COPY A*.* to C:\U100*.*

   If a drive other than A:\ is used, substitute the appropriate letter.

5. Run the U100 communications "com_100.exe" from the PC by typing “com_100.exe 1" (if connected to COM1 port). If running from COM2 port, type “com_100.exe 2”.

   The screen shown in Figure 2-8 appears on the PC.

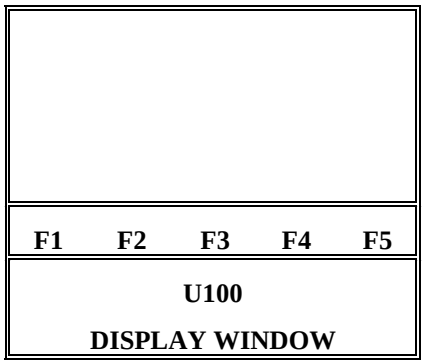


Figure 2-8. Software Display Window

6. Turn on the U100 power switch and the Main Menu window in Figure 2-9 appears. If using the U100i, power is applied after plugging in the power cord. Hit CTRL-D if the screen does not appear.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



Figure 2-9. Main Menu Window (power applied to U100/U100i)

The “In Position” and “Marker” LEDs are the only LEDs that should be lit (U100 only). If these LEDs are not lit or other LEDs besides these two are lit, refer to Chapter 10: Troubleshooting to determine the problem.

7. Type "D" to get into the Status Menu and another "D" to display the Version Menu. The Version Menu returns the software version number installed in the controller.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.7. Simple Program Applications

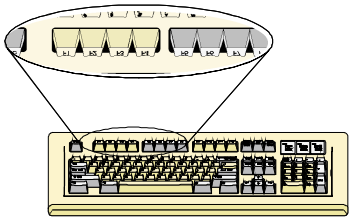
This section provides some simple program applications that briefly walk the user through using the U100/U100i software interface. After the software package is installed and initiated, the Main Menu window is shown on the monitor. The Main Menu window illustrated in Figure 2-10 provides six selections to choose from for the user.



Figure 2-10. Main Menu Window

The Main Menu window contains the following selections:

- File
- Run
- MDI
- Status
- Control
- Misc



In the middle of the Main Menu window is a row of function keys, F1 through F5. These function keys correspond with the function keys on the user's keyboard and the Hand held Terminal (HT). If the user is in another window and wants to return to the Main Menu window, pressing F1 on the keyboard returns the user to the Main Menu. Pressing F5 returns the user to the previous menu.

The function keys F2, F3, and F4 have no function when moving between menus, but do have functions when editing a file. Their functions are discussed in Chapter 4: Software Installation and Interface.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.7.1. Sample Program Number One

To create a simple program, start from the Main Menu window, and perform the following steps.

1. Press “F1” on the keyboard (using PC or Hand held Terminal [HT]) to return to the Main Menu, refer to Figure 2-11.



Figure 2-11. Main Menu Window

2. Press “A” in the Main Menu, the File Edit window appears. Refer to Figure 2-12.



Figure 2-12. File Edit Window

3. Press “A” again in the File Edit window to display the Edit File Type window in Figure 2-13.



Figure 2-13. Edit File Type Window

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4. Press “A” a third time to edit a PGM file.
5. The Enter File Number window will come up, refer to Figure 2-14. When prompted to enter a file number, enter “1”.



Figure 2-14. Enter File Number Window

Since the file does not exist, the software will ask for confirmation to create it.

5. Hit “Y” or yes to create the file and the following screen appears in Figure 2-15.



Figure 2-15. Edit File Window

6. Type in the following program:

```
BEGIN
T(.05)
V(10000)
D(10000)
GO
END
```

Type in the program the same way if using any other editor, hit “Enter” to move to the next line or hit F2 on the keyboard. The use of the function keys for file editing can be found in Table 4-3 of Chapter 4: Software Installation and Interface. After completely typing in the program, perform the following sequences to exit the editor and run the program.

7. Hit F5 twice, then press “Y” to exit the editor. Refer to Figure 2-16.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



Figure 2-16. Finished Editing File Window

8. Return to the Main Menu by hitting F1.
9. Press “B” to run a program and the Select Task window appears in Figure 2-17.



Figure 2-17. Select Task Window

10. Press “A” for Task 1 and the Select Running Mode window appears, refer to Figure 2-18.



Figure 2-18. Select Running Mode Window

11. Press “A” for Auto mode. The Enter File # window appears, enter “1” to run the file just created, refer to Figure 2-19. Hit “Enter” or “Return”.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



Figure 2-19. Enter File Number Window

After hitting "Enter", the motor will rotate 10,000 machine steps. The T(.05) statement in program 1 defines the ramp time. The V(10000) is the velocity of the motor in machine steps/second and the D(10000) is the distance the motor rotates in machine steps.

2.7.2. Sample Program Number 2

The U100/U100i software interface allows monitoring of the position and velocity of the motor. This is done through the Motion Status window. For an example of this, type in the following program in the Edit File window. Perform the same steps as in the previous section, except name this file number 2.

```
BEGIN
LB:10
T(.05)
V(10000)
D(10000)
GO
DW(1)
T(.05)
V(10000)
D(-10000)
GO
DW(1)
GOTO:10
END
```

Exit the Edit File window and perform the same steps as in the previous section to run the program. When prompted to enter the file number, enter number 2 this time. This program rotates the motor 10,000 machine steps, waits 1 second, then reverses the motor rotation by -10,000 machine steps, pauses a second and repeats the process. To monitor the position and velocity, perform the following steps.

1. After the motor starts to rotate, press F1 to return to the Main Menu.
2. Press "D" (STATUS) to bring up the Status Selection window in Figure 2-20.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



Figure 2-20. Status Selection Window

3. Press “A” to bring up the Motion Status window shown in Figure 2-21.



Figure 2-21. Motion Status Window



Using stepping motors will not display position or velocity feedback.

4. To stop or quit the motion, return to the Main Menu and select “E” (CONTROL) and the Control Select window in Figure 2-22 appears.



Figure 2-22. Control Selection Window

5. Press “B” (TASK WINDOW) and the Task window will appear, refer to Figure 2-23.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

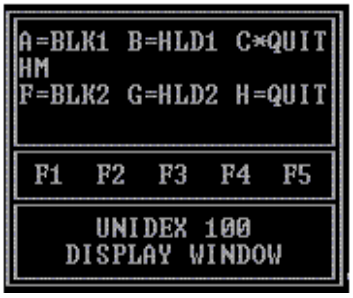


Figure 2-23. Task Window

6. Press "C" to quit.

The ramp time, velocity and distance in the two sample programs were all specified as exact quantities. All these numbers can be replaced with variables. The following example program illustrates the use of variables to replace the exact quantities.

```
BEGIN
FV:1=0.05
BV:1=10000
BV:2=10000
T(FV:1)
V(BV:1)
D(BV:2)
GO
END
```

2.7.3. Multitasking Programs

The two previous example programs were run in Task 1. However, the U100/U100i provide multitasking capabilities where a second program can be executed in Task 2. This multitasking allows the programs in Task 1 and Task 2 to execute simultaneously. For an example of multitasking, type in the following program in the Edit File window. Perform the same steps as in the previous sections and name this program number 3.

```
BEGIN
LB:10
BV:22=0
IF BV:22=100
BV:22=0
CLS
ENDIF
BV:22=BV:22+1
CUR(1,1)
PM("BV:22=")
CUR(1,7)
PM(BV:22)
GOTO:10
END
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Execute PGM2 in Task 1, then hit F1 to return to the Main Menu and run PGM3 in Task 2. While Task 1 is executing (rotating the motor), Task 2 displays the contents of a variable on the screen. The screen will look similar to Figure 2-24.



Figure 2-24. Multitasking (Displayed on Screen while the Motor is Rotating)

If either task uses a command to display text on a screen (i.e., “PM”), then Task 0 is not available. This means that pressing F1 will not return the user to the Main Menu.

To end the programs, hit CTRL -D.

∇ ∇ ∇

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CHAPTER 3: HARDWARE CONFIGURATION & DESCRIPTION

In This Section:

- Introduction 3-1
- UNIDEX 100/U100i Motion Controllers 3-2
- UNIDEX 100/U100i Control Board Jumper Configurations 3-8
- U100 Power Configurations 3-12
- Installing Other Aerotech Components..... 3-14
- After Factory Installation of Option Boards 3-33

3.1. Introduction

This chapter explains how to configure and install the UNIDEX 100 and U100i as well as optional hardware accessories. Provided is a description of the U100/U100i and optional accessories. Configuration of the U100 control board and power board includes jumper settings and wiring references for the power switch, transformer, and fan. The description portion of the optional accessories discusses proper setup procedures for each interface board as well as appropriate installation techniques.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.2. UNIDEX 100/U100i Motion Controllers

The UNIDEX 100/U100i are single axis, programmable, precise positioning motion controllers that can control stepping, DC brush, and AC brushless motors. The standard form of communication for the U100/U100i is a built in RS-232-C port that is labeled "COM" on the front panel. If desired, this COM port can be configured as an RS-422-A interface that requires factory modifications and must be specified when ordering the unit. The front panel contains all the necessary connections for configuring a control system. The U100/U100i can send and receive messages from a PC, terminal, or thumbwheel and can be customized accordingly. The ability to interface with a digitizing joystick is available, so the user can teach the U100/U100i positions. Also, on the front panel (the U100 only) is a column of LED status indicators that indicate the status of the system. Physically, there are two circuit boards that make up the U100: a control board and a power board. Figure 3-1 illustrates the location of all interface connections, power switch, motor fuse, and LEDs. As for the U100i, the physical make up consists only of one control board. Figure 3-2 illustrates the interface connections for U100i. The sections that follow will describe each of them.

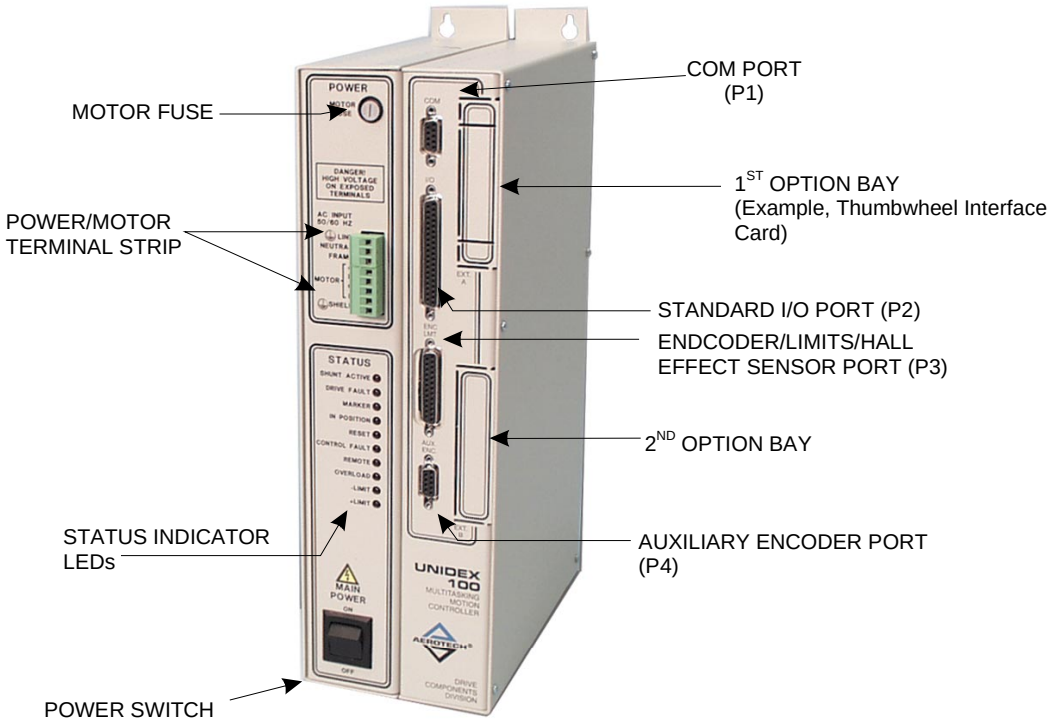


Figure 3-1. UNIDEX 100 Controller

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

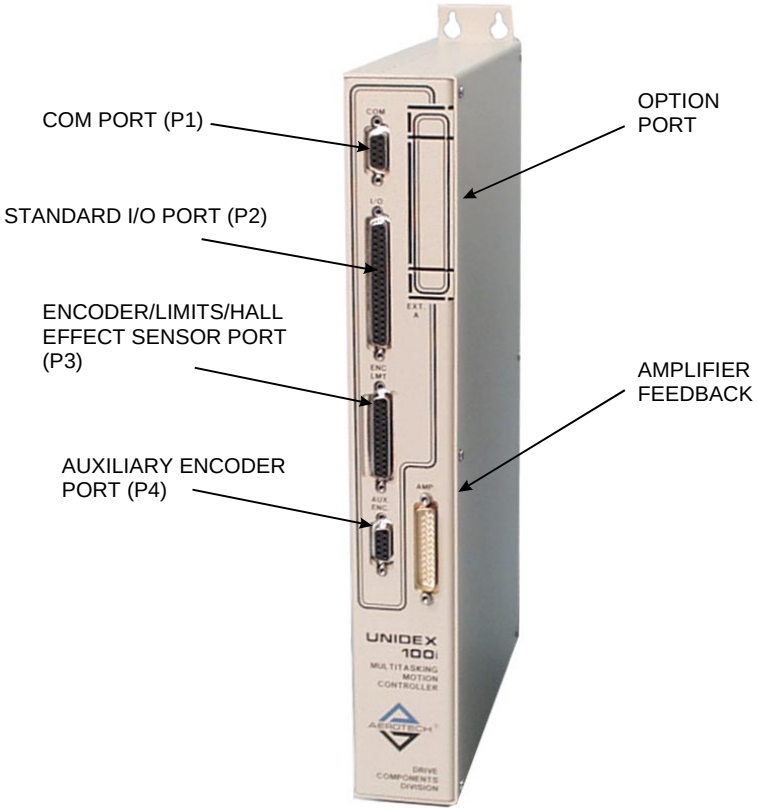
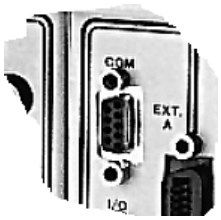


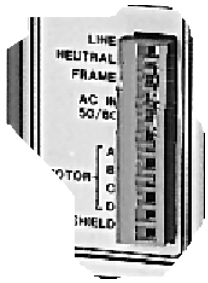
Figure 3-2. UNIDEX 100i Controller

3.2.1. COM Port (U100/U100i)



The COM port is a standard 9 pin "D" connector located on the front of the U100. It consists of two signal lines; transmit (TXD) and receive (RXD), a ground, shield, and a 5V power supply line used to power the Aerotech Hand held Terminal (HT). The COM port can be configured for standard RS-232-C communication or RS-422-A. The differences between the two standards are: the RS-422 interface adds two complimentary signals; receive-n (RXD-N) and transmit-n (TXD-N) and the RS-422 interface uses 5 volt logic levels as opposed to the ±12V signals used by the RS-232 interface. The advantage of the RS-422 interface is it offers higher noise immunity.

3.2.2. Motor and Power Connections (U100 only)



Located on the front panel are removable, keyed screw terminal strips. These strips provide the power and motor connections to the U100. The top terminal strip provides three connections for AC input power. The bottom terminal strip is for motor output power and shielding.

3.2.3. LED Status Indicators (U100 only)

The U100 contains ten LEDs located on the front of the unit that indicate the current status of the system. The LEDs operate independently of each other and represent the

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

presence of several conditions. They provide information on system faults as well as verify proper operation. The following sections explain the function of each LED.

3.2.3.1. Shunt Active LED



While the U100 shunt circuit is on, the "Shunt Active" LED activates only for a brief time (less than 1 sec.). The shunt circuit discharges excess energy from the bus power supply that is pumped into it by back EMF produced by the motor during deceleration. The possibility exists for the motor to produce enough regeneration to blow the shunt circuit fuse. If this happens, the "Shunt Active" LED will stay on longer (possibly continuously), thus activating the "Drive Fault" LED and immediately disabling the power stage.

3.2.3.2. Drive Fault LED

The "Drive Fault" LED is hardware controlled and indicates the status of the power stage. Linked to the drive circuitry within the power stage, this LED will activate anytime there is an over current fault, shunt failure, or a short circuit condition. Any of these conditions will disable the power stage and activate the "Drive Fault" LED. Refer to Chapter 10: Troubleshooting, for symptoms, probable causes, and solutions.

3.2.3.3. Marker LED

The "Marker" LED indicates the status of the encoder marker or index pulse. The primary use of the marker in the home cycle is to find a reference point from which to begin all motion. However, due to the nature of an incremental rotary encoder, the "Marker" LED activates once per revolution of the motor; regardless of whether the motor starts from the home position. The marker activates every time the system returns to the home point with motion command. This verifies that the system is still referenced to the same point it referenced after the most recent home cycle.

3.2.3.4. Reset LED

The "Reset" LED is a hardware activated LED. This LED activates under only two conditions. One, during initial power up of the U100, and two, during activation of the external reset line of the P2 connector. The "Reset " LED activates for approximately 0.5 seconds, then deactivates indicating the microprocessor is running and the unit is active.

3.2.3.5. Control Fault LED



The "Control Fault" LED is capable of indicating a variety of fault conditions. These conditions are divided into two groups: feedback errors and run time errors. The feedback errors consist of either a faulty encoder or no encoder on the primary and/or auxiliary encoder ports.

The run time errors that activate the "Control Fault" LED are programming errors. These errors cannot be detected by the U100 compiler and are not capable of being executed (e.g., changing motor direction within one motion block).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.2.3.6. Remote LED

The "Remote" LED indicates the communication status of the U100. If the U100 is being operated with a hand held terminal or the UT100 Terminal Emulation software, the "Remote" LED is deactivated. However, if communications are established through the optional IEEE 488 bus interface or RS-232 Host Mode, the "Remote" LED is activated.

3.2.3.7. In Position LED

The "In Position" LED indicates that the motor has reached the point of the most recent command. During power up, the "In Position" LED should be activated. When the motor is moving, the indicator is deactivated. Once the motor comes to rest, the indicator activates indicating that there is no commanded motion and the motor is at its commanded position.

3.2.3.8. Overload LED

The "Overload" LED activates each time the U100 detects an exceeded trap condition. The trap conditions that will activate the "Overload" LED are:

- Position Error Trap
- Velocity Feedback Trap
- Acceleration/Deceleration Trap
- RMS Current Limit Setting

3.2.3.9. Limits (+ and -) LEDs

There are two limits LEDs: "+Limit" and "-Limit". The "+Limit" indicates a limit condition responding to Clockwise (CW) rotation of the motor shaft. The "-Limit" indicates a limit condition responding to Counterclockwise (CCW) rotation of the motor shaft.

The limits are divided into sub groups: software limits and hardware limits. The U100 activates software limits when the motor travels out of the boundaries set by parameters "PRM:129" (+software limit) and "PRM:130" (-software limit). Hardware limits activate each time the user trips the limit switches connected to P3-12 (+limit) and P3-24 (-limit). These limit switches can be active high or active low, but it is necessary to set the fault mask accordingly (see Chapter 5 regarding these fault masks). Activation of any of the aforementioned limit conditions, illuminates the appropriate LED and prohibits further movement in that direction.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



3.2.4. I/O Port (U100/U100i)

The I/O port is a 37 pin “D” style connector that contains all the general purpose I/O functions. These functions include:

- 16 non-dedicated digital I/O signals
- 12 dedicated digital I/O signals
- 2 non-dedicated analog I/O signals

There are eight digital opto-isolated inputs and outputs. These digital signals are all user programmable and accessed through registers (see Chapter 7: Registers and Variables for more information on using registers to access these signals).

The analog I/O port of this connector is standard on the U100/U100i. It is in the form of one input and one output, both having a range of ±10V. They are 8 bit ports, thus having a 256 step resolution and both are user programmable.



3.2.5. Encoder/Limits/Hall Effect Port (U100/U100i)

The 25 pin "D" style connector contains all the necessary feedback inputs to complete a servo loop. This port has inputs for a 3 channel encoder, 3 limit switches, and 3 Hall effect devices. Each of these inputs provides feedback for the microprocessor controlled position and velocity loops. The three encoder signals consists of the following: sine (SIN), cosine (COS), and marker (MKR) as well as their complementary signals: sine-n (SIN-N), cosine-n (COS-N), and marker-n (MKR-N). Two of the three limit inputs are end of travel sensing (+ Limit and - Limit) while the third is a reference limit (Home Limit). The Hall effect switch inputs are necessary for AC brushless motor commutation in an AC brushless system that uses an encoder as a feedback device.



3.2.6. Auxiliary Encoder Port (U100/U100i)

The U100/U100i provide an auxiliary port for a second encoder. This is a 9 pin “D” style connector located below the primary encoder port. A common use for the auxiliary port is in a system that uses the primary encoder for velocity feedback and the auxiliary encoder for position feedback. Figure 3-3 shows a simple system using this application.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

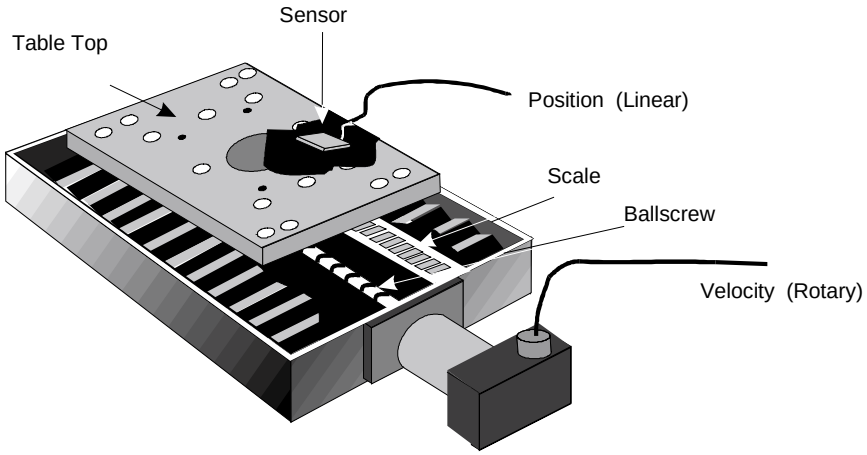
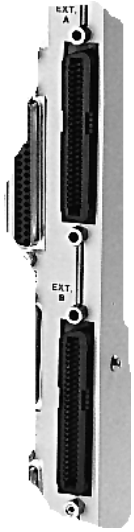


Figure 3-3. Example System Using Velocity and Position Feedback

3.2.7. Option Ports



The two option ports on the U100 (one on the U100i) allow the user to install extension bus option cards. These options include: a thumbwheel assembly, LED display assembly, an IEEE 488 interface card, and a resolver/Inductosyn to digital converter card. A detailed explanation of these options is provided later in this chapter.

3.2.8. Amplifier Feedback (U100i only)

The amplifier feedback connector is the interface between the U100i and Aerotech's BA Series amplifiers or similar.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.3. UNIDEX 100/U100i Control Board Jumper Configurations

This section outlines the jumper configurations of the UNIDEX 100/U100i control board. Descriptions are based on nine functional groups of jumpers:

- Current/Velocity command jumper
- Watchdog enable/disable jumper
- Current output jumpers
- Motor type configuration jumpers
- Battery backed RAM selection jumper

The control board jumpers of the UNIDEX 100/U100i board are configured at the factory according to the application specifications. If no specifications are available, the default jumper settings are used. The locations of the UNIDEX 100/U100i control board jumpers and their default settings are shown in Figure 3-4.

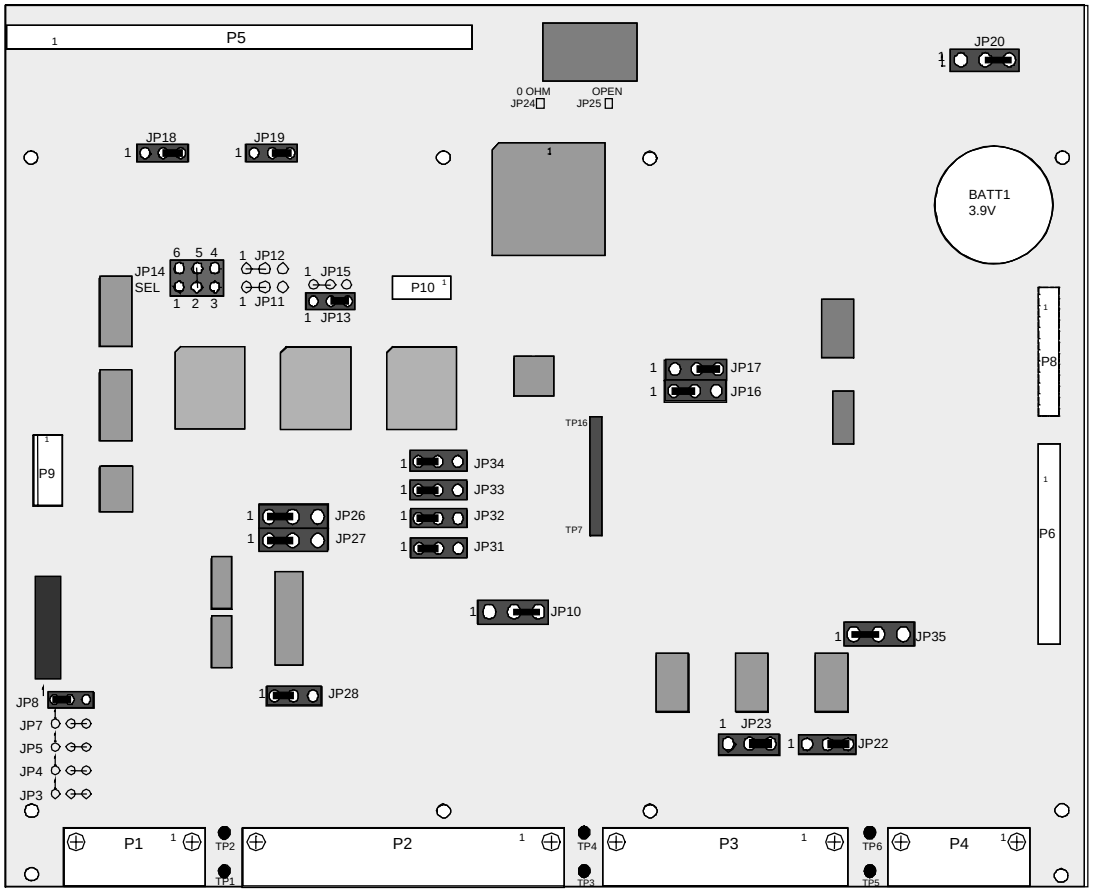


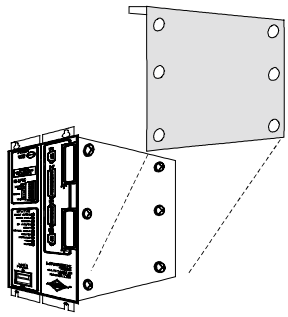
Figure 3-4. UNIDEX 100/U100i Control Board

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

To access the jumpers on the control board, remove 6 screws from the right side of the unit.

To minimize the possibility of electrical shock and bodily injury, make certain that all of the electrical power switches are in the off position and all electrical cables are disconnected prior to making any electrical connections.



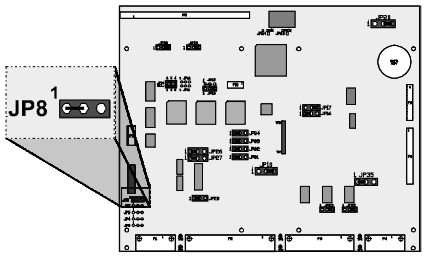


3.3.1. Current/Velocity Command Jumper (JP8)

If the application is a DC brush servo system, JP8 permits the routing of the current command input through the on-board pre-amplifier. Summed with a tachometer feedback signal, this current command signal acts as a velocity command signal. In this configuration, the amplifier controls the speed of the motor and adjusts the velocity depending upon the current command input. Table 3-1 shows the settings for JP8 and Figure 3-4 shows the location of the control board jumper.

Table 3-1. Current/Velocity Command Jumper Setting

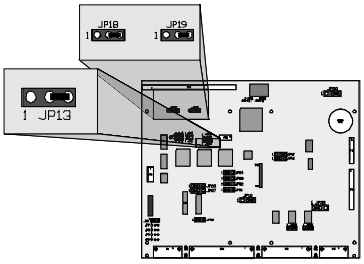
Configuration	JP8
Current Command (default)	
Velocity Command	



The Resistor and Capacitor Network (RCN1) must be installed if configured for Velocity command.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



3.3.2. Current Output Jumpers (JP13, JP18 & JP19)

The UNIDEX 100/U100i provide the capability to drive 3 types of motors, each with different current ratings. From the control board, the user can select the appropriate output current by reconfiguring the jumpers. The setting of these jumpers will depend on the motor and current required to drive it. Table 3-2 shows the current ranges for all jumper settings. Figure 3-4 shows the control board jumper locations.

Table 3-2. Current Output Jumper Settings

DC or AC Servo Current	Stepper	JP13	JP18	JP19
0 Amps	0 Amps			
3 Amps	1.5 Amps			
5.5 Amps	2.75 Amps			
8.5 Amps	4.25 Amps			
11 Amps	5.5 Amps			
14 Amps	7 Amps			
17 Amps (default)	8. 5 Amps			
20 Amps	10 Amps			



The current range is the absolute maximum current that the U100/U100i outputs for a given setting.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.3.3. Motor Type Configuration Jumpers (JP16 & JP17)

The UNIDEX 100/U100i is capable of driving three types of motors: DC Brush, AC brushless, and steppers. There are two DC brush motor configurations, velocity command and current command. To run in the velocity command mode, JP8 must be configured properly and RCN1 installed. Table 3-3 shows the proper configuration of these jumpers depending upon the type of motor driven by the controller. Figure 3-4 shows the jumper locations on the control board.

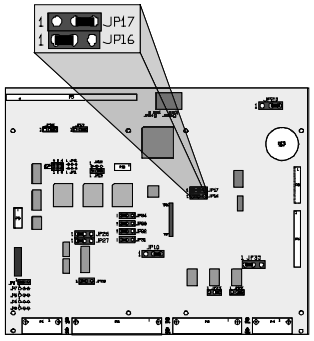


Table 3-3. Motor Type Jumper Selections

Mode	JP16	JP17
DC Brush (Velocity command)	1 3	1 3
DC Brush (Current command)	1 3	1 3
Stepping	1 3	1 3
AC Brushless	1 3	1 3

3.3.4. Firmware Boot Jumper (JP34)

The firmware boot jumper allows the U100 firmware to be downloaded from the PC to the U100 through the RS-232 connection. Refer to Table 3-4 for jumper settings. Figure 3-4 shows jumper location.

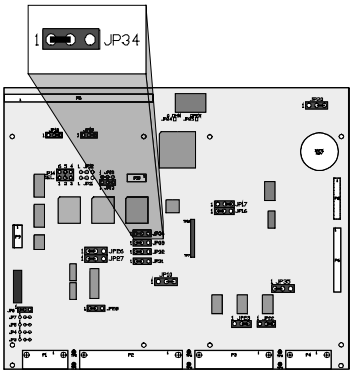
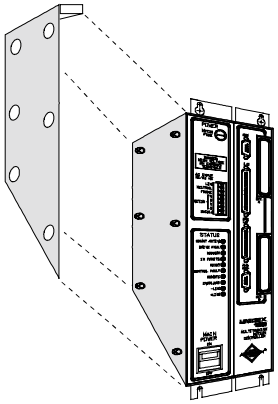


Table 3-4. Firmware Boot Jumper Selections (JP34)

Mode	JP34
Normal	1
Boot loaded	1

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



3.4. U100 Power Configurations - U100 Controller

The UNIDEX 100 can be configured to operate from 100, 115, and 230 VAC input. The AC input select jumpers (JP5 and JP6) located on the U100 power board provides the ability to do this. These jumpers are factory set according to the specifications provided by the user, so the user does not need to change these settings. Additional connections and configurations made to the power board are for the power switch, transformer, fan, and fuses (refer to Figure 3-5). The fuses are the only items the user should be concerned with when encountering a problem.

3.4.1. U100 Transformer

The standard configuration of the U100 operates with a DC bus voltage of 160 VDC. This is 115 VAC line rectified. However, there are cases when this is not desirable. For example, running a DC servo motor requires a 40 VDC bus. These and other possible configurations require modification of the bus voltage. This is accomplished with the installation of an auto transformer. Depending on the specification provided by the user, the auto transformer is factory installed. Figure 3-5 shows the location of the auto transformer and the power connections on the power board when installed in the U100.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

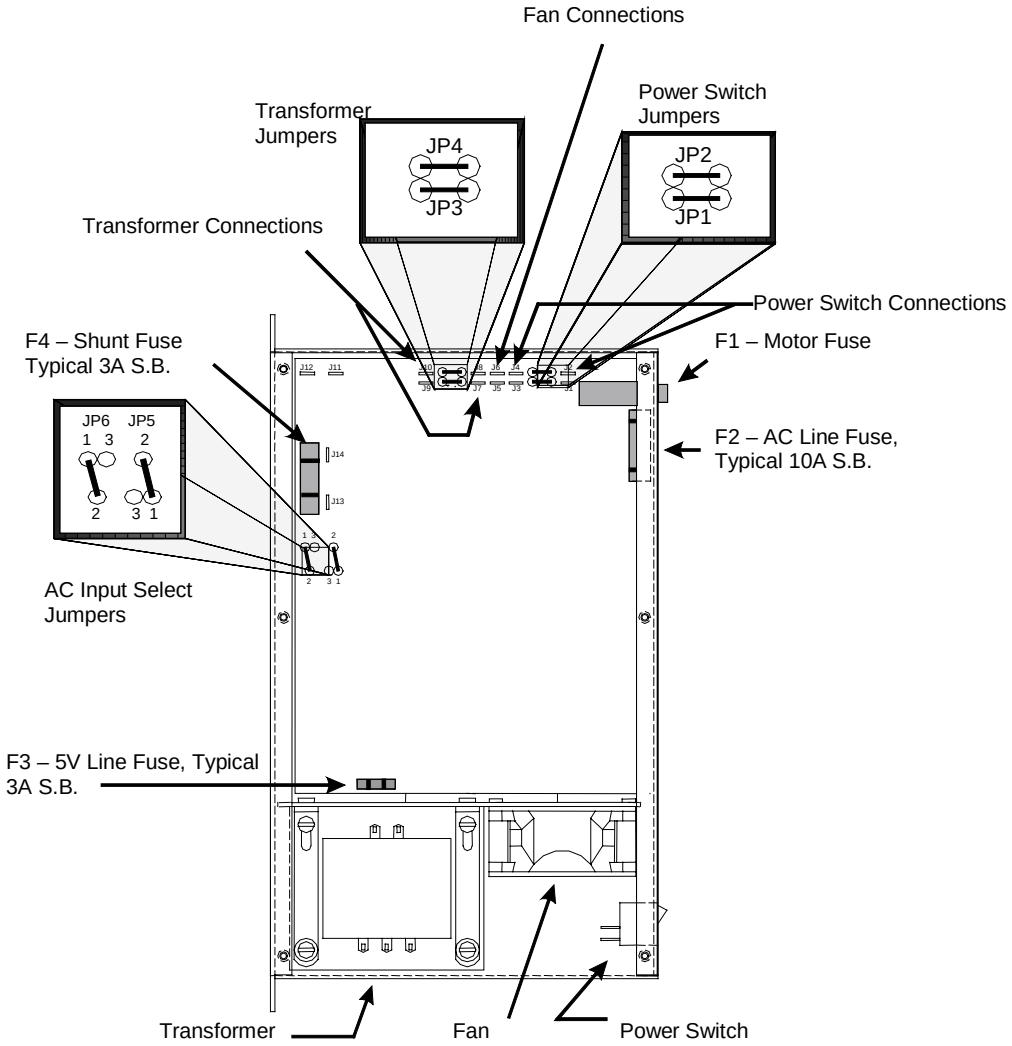


Figure 3-5. UNIDEX 100 Power Board

3.4.2. U100 Fuses

The U100 fuses are located on the power board except for the motor fuse (F1), this fuse is accessible from the outside of the unit (refer to Figure 3-5). In most cases under normal operation, it is not necessary to change the fuses on the power board. Fuse F1 will require replacement most of the time, since programming errors occasionally cause the fuse to blow. To access the other fuses on the power board, remove six screws that attach the left side cover to the frame of the U100.

The value of the motor fuse (F1) depends upon the type of motor being driven by the U100.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.5. Installing Other Aerotech Components

System installation varies with the number and types of components that have been purchased from Aerotech, Inc. to complement the UNIDEX 100/U100i controllers. The following descriptions may not be applicable to all systems.

3.5.1. The Hand Held Terminal (HT)

The Hand Held Terminal or (HT), shown in Figure 3-6, is an optional operator interface that provides the operator local control of the controller through menu assisted screens. The HT plugs into the COM port (P1) of the controller that is configured for standard RS-232-C communication. The controller provides a 5 V power supply line through the COM port that powers the HT. The HT can be used to edit programs, run programs, run commands, and check controller status information. Figure 3-7 shows a typical configuration using the HT. Figure 3-8 shows the HT configuration using the U100i. Refer to Chapter 4: Software Installation and Interface, for a detailed explanation of the setup and interface of the HT.



Figure 3-6. Hand Held Terminal (HT)

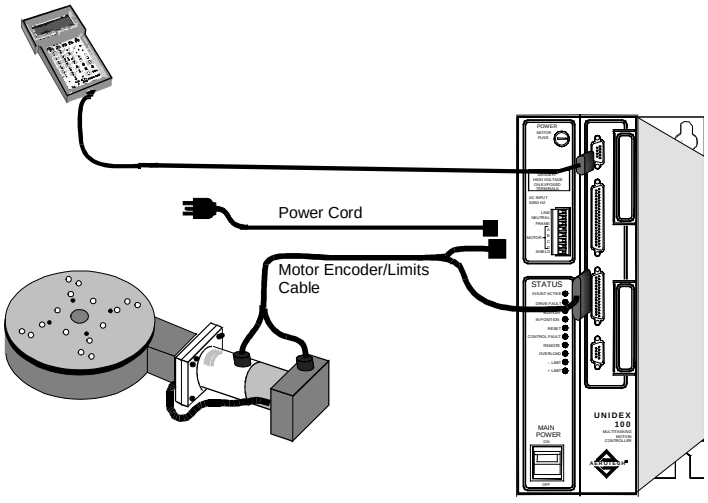


Figure 3-7. HT Configuration (U100)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

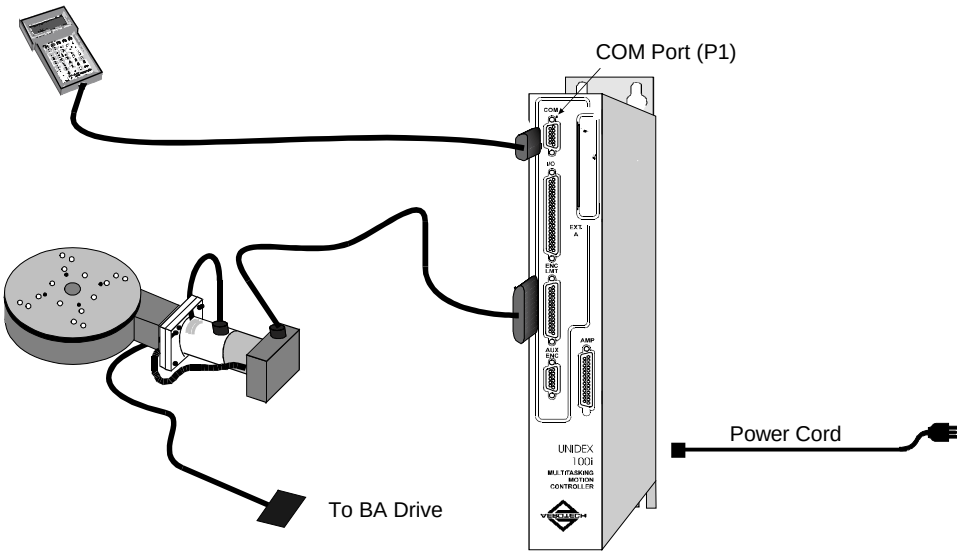


Figure 3-8. HT Configuration (U100i)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.5.2. Joystick

The joystick is an optional interface accessory that provides the user the capability to manually slew an axis with proportional speed control. Power for the joystick is provided by the controller. The digitizing feature allows the user to teach the U100/U100i positions. The joystick comes with a 5 ft flying leads cable for connection to the I/O port. An illustration of the controller joystick is shown in Figure 3-9.



An example program on how to use the analog input from the joystick can be found in Chapter 8: Sample Programs.

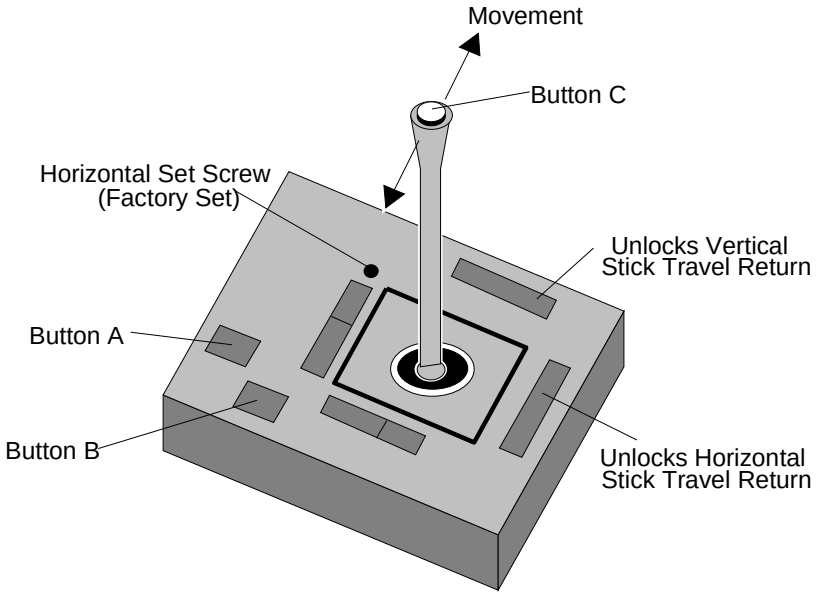


Figure 3-9. Optional UNIDEX 100/U100i Joystick

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.5.3. Thumbwheel

The thumbwheel is a 6 digit, plus sign assembly with a 10 ft cable. Included with the thumbwheel is an interface card that is factory installed and plugs into the controller extension bus (P5 of the controller control board). Power for the thumbwheel is provided by the controller. The thumbwheel allows the user to enter data such as distance, speed, or number of cycles into parts program variables. The thumbwheel is shown in Figure 3-10.



Figure 3-10. Optional Thumbwheel

There are two connectors available on the thumbwheel interface card that is installed in the controller. If desired, two thumbwheels can be used per card in a system, each independently addressed (refer to Figure 3-11). The thumbwheel is a read only device that is read with port variables (PV:). The default address is "0xE000" and can be changed using the DIP switch (SW1) located on the interface board inside the controller. Refer to Table 3-5 for base address selections.

There are four THM locations that are accessible by the user, these locations are:

- THM 1 Data (P2 of Thumbwheel interface board) - 0xE0*0
- THM 1 Sign and execute (P2 of Thumbwheel interface board) - 0xE0*1
- THM 2 Data (P3 of Thumbwheel interface board) - 0xE0*2
- THM 2 Sign and execute (P3 of Thumbwheel board) - 0xE0*3

The "*" indicates the assignment of a base address to this location from Table 3-5.

The designation of the thumbwheel as a read only device means the controller is the reference point where it can only read data from the thumbwheel and not return or exchange data.

The interface is the same for the U100i using its single option port.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

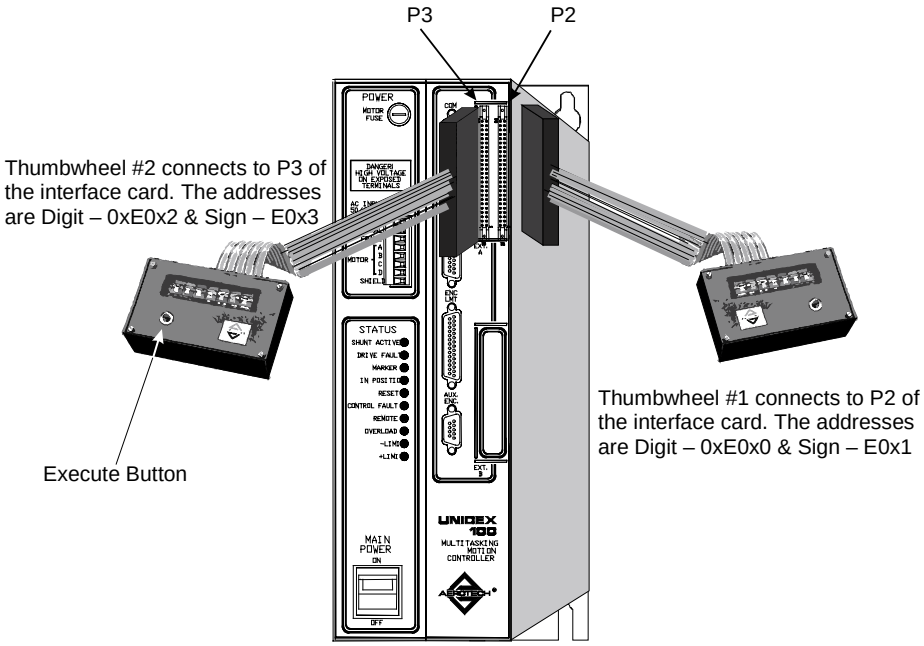


Figure 3-11. Thumbwheel Interface

Table 3-5. Thumbwheel Bus Addresses

Base Address	SW1-1	SW1-2	SW1-3	SW1-4
0xE000	on	on	on	on
0xE010	off	on	on	on
0xE020	on	off	on	on
0xE030	off	off	on	on
0xE040	on	on	off	on
0xE050	off	on	off	on
0xE060	on	off	off	on
0xE070	off	off	off	on
0xE080	on	on	on	off
0xE090	off	on	on	off
0xE0A0	on	off	on	off
0xE0B0	off	off	on	off
0xE0C0	on	on	off	off
0xE0D0	off	on	off	off
0xE0E0	on	off	off	off
0xE0F0	off	off	off	off

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Address selections provide the capability to use multiple boards or more than one Thumbwheel interface board.



The controller reads the thumbwheel data with a “PV:” variable addressed to the thumbwheel data port. Thumbwheel #1 data port reads thumbwheel connected to P2 of the interface board and uses an address of “0xE0*0”. Thumbwheel #2 data port reads thumbwheel connected to P3 of the interface board and uses an address of “0xE0*2”.

The “*” is determined by the base address selected with DIP switch (See Table 3-5).



Before the thumbwheel data is read, it is necessary to latch it using the thumbwheel execute button (refer to Figure 3-11). The execute latches the data and sign into the thumbwheel interface card. A program written by the operator will input the value set on the thumbwheel to a variable of a parts program; whether it is for distance, speed, or number of cycles.

An example program that illustrates how this works is in Chapter 8: Sample Programs.



3.5.3.1. Thumbwheel Sign

A separate data address reads the Thumbwheel sign bit. Like the data there are two sign addresses for Thumbwheel #1 and Thumbwheel #2. To obtain Thumbwheel #1 (P2 of the interface board) the address is 0xE0*1 (* determined by the base address, see Table 3-5). Thumbwheel #2 (P3 of the interface board) requires that address 0xE0*3 be read to obtain its sign.

Bit #23 holds the sign status and changes its setting to a one if the sign is negative. When the user requires a negative data value, the sign can be determined and the data negated by testing this bit. For example: BV:1=PV:0xE001 AND 0x800000, IF(BV:1 EQ 0x800000) which reads the Thumbwheel #1 sign and tests the sign for a negative. The status of the sign gets sent to the BV:1 variable.

3.5.3.2. Thumbwheel Execute Button

The execute latches the data and sign into the Thumbwheel interface board upon pressing the execute button. The execute status is bit #16 and uses the same address as the sign. Bit #16 is high upon pressing the execute button. For example IF(PV=0xE001 AND 0x010000) which checks for a Thumbwheel #1 execute switch closure.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.5.4. Display Option

Another optional operator interface is the 6 digit, plus sign LED display with a 10 ft cable and interface card. The interface card is factory installed and plugs into the controller extension bus (P5 of the controller control board). A power cable that is also factory installed, connects to the interface card and is fed through the same option port as the interface connector. This accessory allows the user to display position, velocity, cycle number or any other parts program variable. Figure 3-12 shows the 6 digit LED display.



Figure 3-12. Optional LED Display

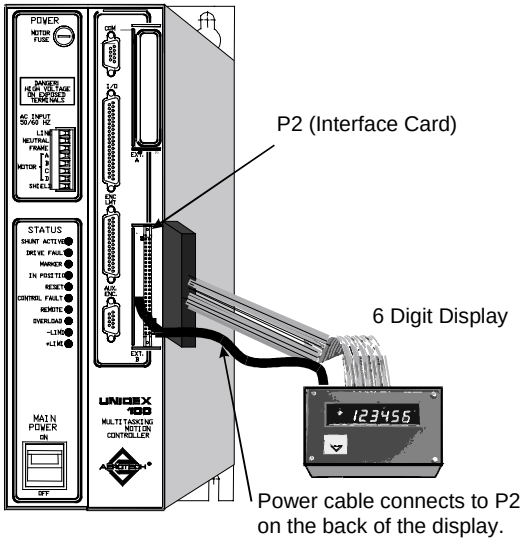


Figure 3-13. Display Configuration

The display connects to P2 of the display interface card installed in the controller. A power cable and connector extend out from the same port and connects to P2 on the display box (refer to Figure 3-13). The interface is the same for the U100i using its single option port.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 3-6. Display Bus Addresses

Base Address	SW1-1	SW1-2	SW1-3	SW1-4
0xE000	on	on	on	on
0xE010	off	on	on	on
0xE020	on	off	on	on
0xE030	off	off	on	on
0xE040	on	on	off	on
0xE050	off	on	off	on
0xE060	on	off	off	on
0xE070	off	off	off	on
0xE080	on	on	on	off
0xE090	off	on	on	off
0xE0A0	on	off	on	off
0xE0B0	off	off	on	off
0xE0C0	on	on	off	off
0xE0D0	off	on	off	off
0xE0E0	on	off	off	off
0xE0F0	off	off	off	off

The display is a write only device that is programmed with port variables (PV:). The default address is "0xE000" and can be changed using the DIP switch (SW1) located on the interface board. Refer to Table 3-6 for base address selections.

There are three display locations that are programmable by the user, refer to Figure 3-14, these locations are:

- Data display (0xE0*0)
- ± Sign data (0xE0*1)
- and number of places after the decimal point (0xE0*2).

The "*" indicates the assignment of a base address to this location from Table 3-6.

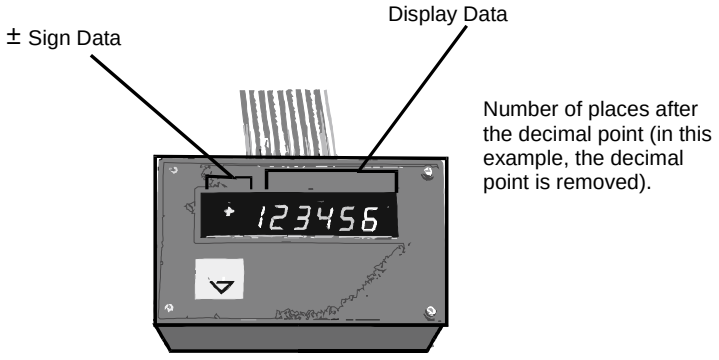


Figure 3-14. Programmable Locations on LED Display

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

To display information, the appropriate port variable “PV:” must be set to the desired data and a base address assigned to the display data location. In the example below, the base address is “0xE000” and all six digits are displayed with no places to the right of the decimal point.

PV:0xE000 = 123456

To program the ± sign, the appropriate port variable “PV:” must be set.

To program the decimal point, enter the desired number of places to the right of the decimal point to the port variable “0xE0*2. Follow the examples below.

PV:0xE002=3	Displays 3 places to the right of the decimal point.
PV:0xE002=0	Removes the decimal point.
PV:0xE002=6	Displays all 6 digits to the right of the decimal point.



Example programs that illustrate how this works is in Chapter 8: Sample Programs.

3.5.5. Resolver-to-Digital Converter

The optional Resolver-to-Digital (R/D) converter board enables the U100/U100i to use resolver or inductosyn feedback as position and/or velocity information. It allows for high resolution positioning performance by converting analog feedback signals into 12, 14, or 16 bit digital data. Thus, one electrical cycle of a resolver or inductosyn can be interpreted as 4,096 counts (12 bit), 16,384 counts (14 Bit), or 65,536 counts (16 bit).

The controller R/D board incorporates a mode for automatic dynamic resolution switching on the fly. By switching to the 14 bit mode after exceeding a pre-determined speed, the system can maintain high speed capability. Switching back to the 16 bit mode below the speed threshold allows 16 bit positioning accuracy.

The R/D board plugs into the extension bus (P5 of the control board) and mounts to the control board. The controller provides the power for the R/D board. The interface to a resolver or Inductosyn is through a 15 pin “D” style connector (P2 on the interface card) that protrudes from one of the option ports, refer to Figure 3-15. Table 3-7 lists the pinout descriptions.



The interface is the same for the U100i using its single option port.

To use the R/D option, the controller must be configured by setting certain parameters in the controller’s battery backed memory. Table 3-8 shows the parameters and their settings. For additional settings for each of these parameters, turn to Chapter 6: Parameters.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

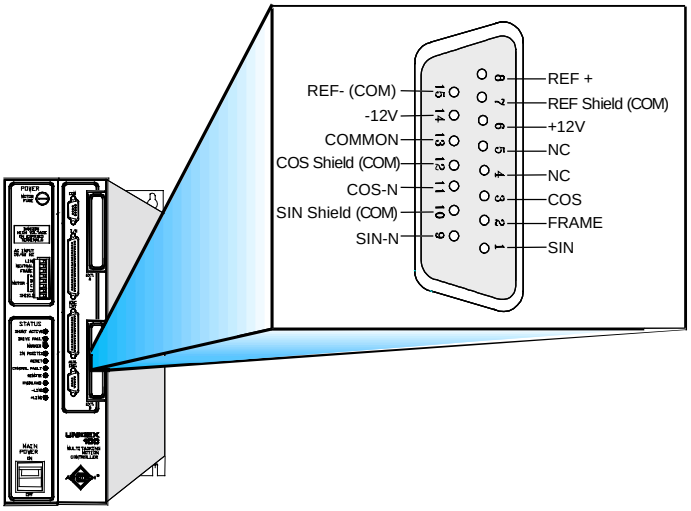


Figure 3-15. R/D Interface Board Connector Pinouts

Table 3-7. R/D Connector Pinout Descriptions

P2	Description	
Pin 1	**SIN**	Sine Feedback Signal
Pin 9	**SIN-N**	Compliment Sine Feedback Signal
Pin 2	**FRAME**	Earth Ground
Pin 10	**SIN SHIELD (Com)**	Shield Connection for SIN/SIN-N (twisted pair), connects to common
Pin 3	**COS**	Cosine Feedback Signal
Pin 11	**COS-N**	Compliment Cosine Feedback Signal
Pin 4	**NC**	
Pin 12	**COS SHIELD (Com)**	Shield Connection for COS/COS-N (twisted pair), connects to common
Pin 5	**NC**	
Pin 13	**SIGNAL COMMON**	
Pin 6	**+12V P.S.**	For supplying Inductosyn pre-amp board
Pin 14	**-12V P.S.**	For supplying Inductosyn pre-amp board
Pin 7	**REF SHIELD (Com)**	Shield Connection for REF/REF-N (twisted pair), connects to common
Pin 15	**REF-N (Com)**	Signal Common
Pin 8	**REF**	R/D Output Signal for Driving Resolver or Inductosyn

The maximum R/D feedback data rate is 1,000,000 machine steps/second. If you are using a rotary resolver, this translates to 14,648 rpm in the 12 bit mode, 3,662 rpm in the 14 bit mode (or 16/14 bit mode), and 915 rpm in the 16 bit mode.

Table 3-8. R/D Standard Parameter Settings

Parameter	Setting	Function
PRM:118	4	Home on zero crossing of R/D
PRM:137	4	Enables 16/14 bit auto-select mode
PRM:138	4	Enables velocity feedback from R/D
PRM:139	4	Enables position feedback from R/D
PRM:310	4	Sets the R/D update rate at .8 ms

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

When PRM:137 is set to "4", the controller positions in 16 bit mode. However, above a preset speed, the controller switches into 14 bit mode to allow high speed operation.

Setting parameter PRM:310 to "4" allows proper tuning and operation for most systems. This value, however, can be set from 1 to 8 depending of the construction of the system. The guideline to be observed in this setting is that the higher the inertia of the mechanical system or the lower the resolution of the feedback device (resolver or Inductosyn), the higher this value must be set. For lower inertia loads or a higher resolution feedback device, a lower value must be used.

In most cases, the R/D is the source of both position and velocity feedback for the UNIDEX 100/U100i. However, it is possible to use the R/D as only a position or velocity feedback source, while an incremental encoder is the other feedback source (position or velocity). Generally, this would only be done in a highly compliant mechanical system.

Parameters PRM:307 and PRM:308 (refer to the U100 manual for more information on these parameters) should be set to "0" if no incremental encoder is being used to supply position and/or velocity feedback to the controller.



Parameter PRM:307 applies to the primary encoder and parameter PRM:308 corresponds to the auxiliary encoder. Setting these parameters to zero provides better performance of the UNIDEX 100/U100i operating system. For additional parameter settings, turn to Chapter 6: Parameters (Sections 6.6.8 and 6.6.9).



It is only necessary to set the parameters mentioned above once. However, if the battery backed memory becomes corrupted for any reason, clear the memory and perform a setup function. Refer to Chapter 4: Software Installation and Interface for a description of the setup function.

3.5.6. The INT Option

The INT option provides the user the ability to interface with one or two OPTO 22 PB8, PB16, or PB24 mounting rack. The PB8, PB16 and PB24 options are boards that provide additional controller inputs and outputs. A PB# mounting rack connects to P2 or P3 of the INT interface card installed in the controller using a 50-pin ribbon cable (provided), refer to Figure 3-16. The interface card plugs into the controller extension bus (P5 of the control board) and mounts to control board. The PB8 provides 8 I/O points, the PB16 provides 16 I/O points, and the PB24 provides 24 I/O points. PB# mounting racks accept a wide variety of I/O modules for different levels of AC or DC voltage. I/O points can be selected to be inputs or outputs in groups of 8.

EXAMPLE:

PB24

INPUT	OUTPUT
0	24
8	16
16	8
24	0

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

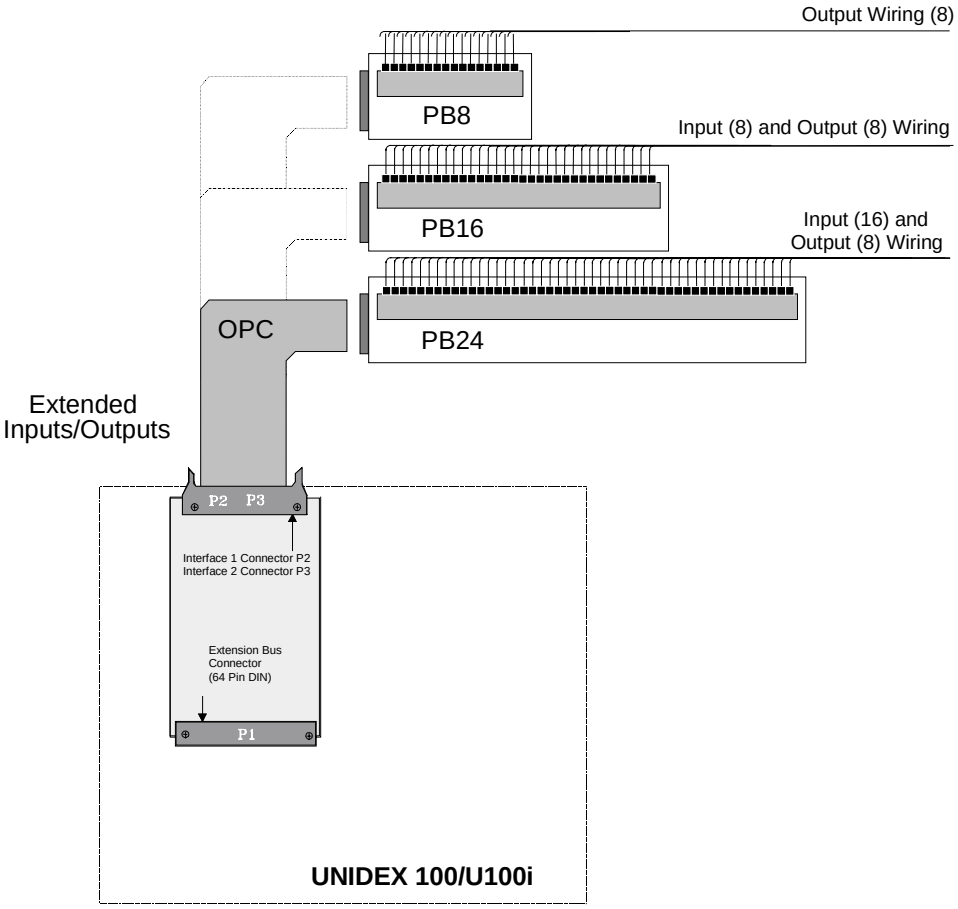


Figure 3-16. Sample Uses of the PB# Boards

The INT option board is a read/write device that is programmed with port variables (PV:). The default address is "0xE000" and can be changed using the DIP switch (SW1) located on the INT option board. Refer to Table 3-9 for base address selections.

Address selections provide the capability to use multiple boards or more than one INT option board.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 3-9. INT Bus Address Selections

Address	SW1-1	SW1-2	SW1-3	SW1-4
0xE000	on	on	on	on
0xE010	off	on	on	on
0xE020	on	off	on	on
0xE030	off	off	on	on
0xE040	on	on	off	on
0xE050	off	on	off	on
0xE060	on	off	off	on
0xE070	off	off	off	on
0xE080	on	on	on	off
0xE090	off	on	on	off
0xE0A0	on	off	on	off
0xE0B0	off	off	on	off
0xE0C0	on	on	off	off
0xE0D0	off	on	off	off
0xE0E0	on	off	off	off
0xE0F0	off	off	off	off

There are three INT locations that are programmable by the user, these locations are:

- 0xE0*0 - 1st PB board
- 0xE0*1 - 2nd PB board
- 0xE0*2 - reset latch (special location)



"0xE0*2" is a special location. The controller INT interface board incorporates a latch circuit that guarantees all outputs will go to an inactive state upon a system reset or power up condition. Before data can be output to either mounting rack, a write function, using any data must be performed to "0xE0*2" in order to bring the outputs out of this latched inactive state.

Inputs can be read without having to clear the reset latch. It is also possible to write data to the output latches before having cleared the reset latch. However, this does not appear at the PB board until the write is performed to 0xE0*2.

Addresses 0xE0*0 and 0xE0*1 allow communication with the modules installed on the Opto 22 PB board(s) connected to the INT interface. By adjusting jumper selections, the user can configure each group of eight I/O points as inputs or outputs. Refer to the examples provided below.

On a PB24, modules 0-7 could be configured as inputs with modules 8-23 configured as outputs.

On a PB24, modules 0-7 and 16-23 could be configured as outputs with modules 8-15 configured as inputs.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

To write data to outputs do the following.

```
PV:0xE000 = 0x555555          ;turns on every other output starting with module 0.
```

When writing data to the INT option, ensure the bits configured as inputs are not affected and do not need to be masked out of the data being written.



To read data from inputs perform the following.

```
BV:1=PV:0xE000          ; this reads data from modules 0-23.
```

The above statement can be used to read inputs and the programmed state of the outputs. If all modules were configured to be inputs, this statement would reflect the 24 bit input data. If all modules were configured as outputs this statement would reflect the data that was most recently written to the outputs through a port variable. If the 24 modules were divided as inputs and outputs the above statement would reflect the appropriate combination of input data and programmed output data.



Also, it is important to consider that mathematical and logical functions can be performed on output data since it can be read back. The following example illustrates this feature.

EXAMPLE:

```
PV:0xE001=PV:0xE001 AND 0x0000FF    ;The above statement turns off the upper
                                    ;sixteen bits of the PB24 interface not
                                    ;disturbing any of the lower eight bits.
```

3.5.6.1. INT Jumper Settings

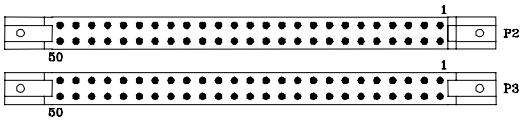
There are three groups of jumpers on the INT option board that permit the user to configure each group of I/O points as inputs or outputs. One group of jumpers can be used as interrupts for the controller extension bus. Refer to Table 3-10 for jumper selections on INT board. Figure 3-17 shows the location of these jumpers.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 3-10. INT Board Jumper Selections

PB# Interface 2 Input/Output Jumper Selections						
	JP1	**JP2**	**JP3**	**JP4**	**JP5**	**JP6**
	Modules 0-7		**Modules 8-15**		**Modules 16-23**	
Input						
Output						
PB# Interface 1 Input/Output Jumper Selections						
	JP7	**JP8**	**JP9**	**JP10**	**JP11**	**JP12**
	Modules 0-7		**Modules 8-15**		**Modules 16-23**	
Input						
Output						
Interrupt Polarity Select Jumpers						
	JP13		**JP14**		**JP15**	
Interrupt Polarity Active Low			**Note: JP13 must always be set to 1-2. 2-3 is invalid.**			
Interrupt 1 Enable						
Interrupt 1 Disable						
Interrupt 2 Enable						
Interrupt 2 Disable						

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



MODULE 23	01 – 01	02 – 26	SIGNAL COMMON
MODULE 22	02 – 03	04 – 27	SIGNAL COMMON
MODULE 21	03 – 05	06 – 28	SIGNAL COMMON
MODULE 20	04 – 07	08 – 29	SIGNAL COMMON
MODULE 19	05 – 09	10 – 30	SIGNAL COMMON
MODULE 18	06 – 11	12 – 31	SIGNAL COMMON
MODULE 17	07 – 13	14 – 32	SIGNAL COMMON
MODULE 16	08 – 15	16 – 33	SIGNAL COMMON
MODULE 15	09 – 17	18 – 34	SIGNAL COMMON
MODULE 14	10 – 19	20 – 35	SIGNAL COMMON
MODULE 13	11 – 21	22 – 36	SIGNAL COMMON
MODULE 12	12 – 23	24 – 37	SIGNAL COMMON
MODULE 11	13 – 25	26 – 38	SIGNAL COMMON
MODULE 10	14 – 27	28 – 39	SIGNAL COMMON
MODULE 09	15 – 29	30 – 40	SIGNAL COMMON
MODULE 08	16 – 31	32 – 41	SIGNAL COMMON
MODULE 07	17 – 33	34 – 42	SIGNAL COMMON
MODULE 06	18 – 35	36 – 43	SIGNAL COMMON
MODULE 05	19 – 37	38 – 44	SIGNAL COMMON
MODULE 04	20 – 39	40 – 45	SIGNAL COMMON
MODULE 03	21 – 41	42 – 46	SIGNAL COMMON
MODULE 02	22 – 43	44 – 47	SIGNAL COMMON
MODULE 01	23 – 45	46 – 48	SIGNAL COMMON
MODULE 00	24 – 47	48 – 49	SIGNAL COMMON
+ 5 Volts	25 – 49	50 – 50	SIGNAL COMMON

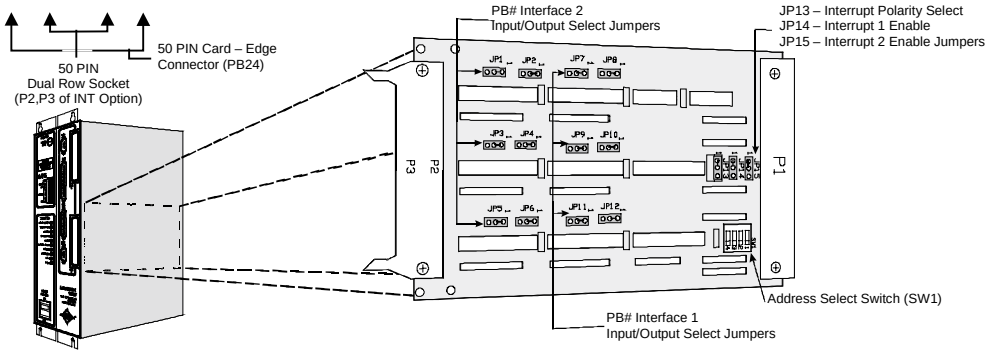


Figure 3-17. INT Board Jumper and Switch SW1 Locations

3.5.7. IE-488 Option

The IE-488 option permits the user to control the UNIDEX 100/U100i through the IEEE-488 bus. Using the HOST command set, the IE-488 offers many control capabilities. These include: executing programs and commands, transferring files, and reading or modifying data.

The IEEE-488 interface is an 8 bit parallel data bus with eight control lines. It can accommodate up to fifteen devices on the bus. This bus also provides a method of requesting attention from the Host controller, a method referred to as the service request (see Chapter 5: Programming Commands). The IEEE-488 interface card plugs into the controller extension bus (P5 of the control board) and mounts to the control board. Connections to the interface card must be made with approved IEEE-488 cables with a cable length that does not exceed twenty meters.

The IE-488 connector is a 24 pin CHAMP connector. Figure 3-18 shows the IEEE-488 interface card and connector P2 pinouts. Refer to Table 3-11 for pin descriptions.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

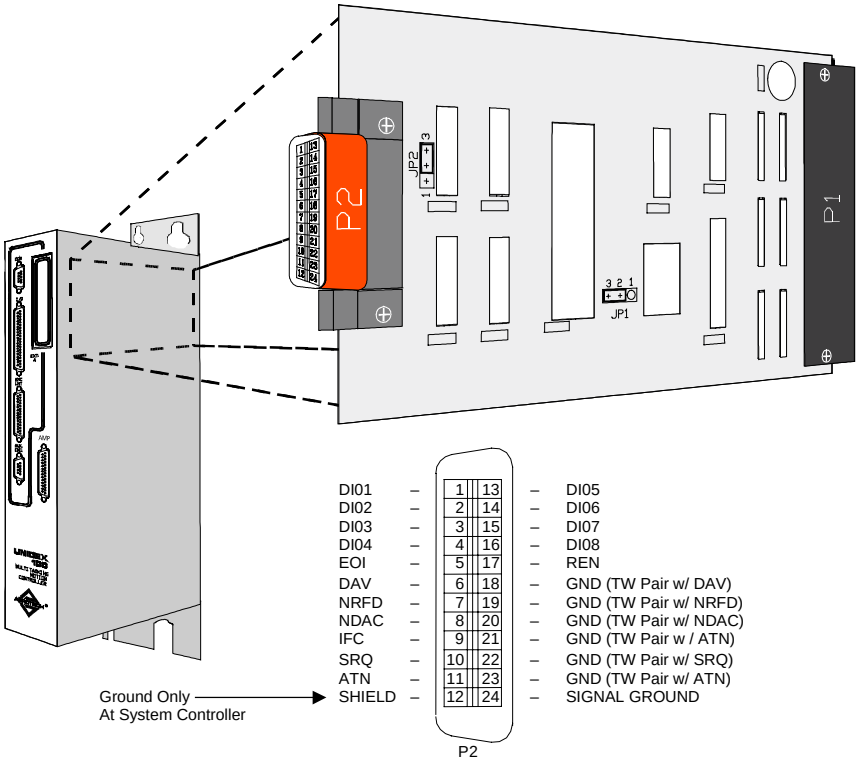


Figure 3-18. IEEE-488 Interface Card

Table 3-11. IEEE-488 Connector P2 Pin Descriptions

P2	Description
Pin 1	DIO1 - Data line #1.
Pin 2	DIO2 - Data line #2.
Pin 3	DIO3 - Data line #3.
Pin 4	DIO4 - Data line #4.
Pin 5	EOI (End of Identify) - This control line indicates the last data byte and will parallel poll devices using the ATN line.
Pin 6	DAV (Data Valid) - This control line, used by the talker device, will indicate to the listener devices that the data is valid.
Pin 7	NRFD (Not Ready for Data) - This control line indicates that one or more devices are not ready for data.
Pin 8	NDAC (Not Data Accepted) - This control line indicates that one or more devices have not accepted the data.
Pin 9	IFC (Interface Clear) - The system controller activates this control line and place all devices in the unaddressed state. This line will also cause the system controller to be the active device.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 3-11. IEEE-488 Connector P2 Pin Descriptions Cont'd

P2	Description
Pin 10	SRQ (Service Request) - This control line indicates that one or more devices require attention. Following activation of this line the controller performs a poll of the devices to determine which one is requesting service. A Serial Poll will clear the SRQ.
Pin 11	ATN (Attention) - This control line when asserted true sends bus interface messages on the data bus. This line, asserted with EOI, will do a parallel poll. When ATN is false, it is possible to send data over the bus by a designated talker.
Pin 12	Shield - This pin is the cable shield that normally connects to ground at the system controller. The U100 IE-488 board contains a shield jumper JP2 (1-2 is grounded and 2-3 is unconnected).
Pin 13	DIO5 - Data line #5.
Pin 14	DIO6 - Data line #6.
Pin 15	DIO7 - Data line #7.
Pin 16	DIO8 - Data line #8.
Pin 17	REN (Remote Enable) - This control line will place any addressed listening device into the remote mode, upon asserting this line.
Pin 18	GND (Ground) - Typically twisted wire pair with DAV line.
Pin 19	GND (Ground) - Typically twisted wire pair with NRFD line.
Pin 20	GND (Ground) - Typically twisted wire pair with NDAC line.
Pin 21	GND (Ground) - Typically twisted wire pair with IFC line.
Pin 22	GND (Ground) - Typically twisted wire pair with SRQ line.
Pin 23	GND (Ground) - Typically twisted wire pair with ATN line.
Pin 24	Signal Ground - This is signal common.

To use the IE-488 option, the controller must be configured by setting certain parameters in the controller's battery backed memory. Table 3-12 shows the parameters and there settings. For additional settings and explanations for each of these parameters, turn to Chapter 6: Parameters.

Table 3-12. IE-488 Standard Parameter Settings

Parameter	Setting	Function
PRM:008	0x24(default)	Sets the mode zero address
PRM:009	0x44(default)	sets the mode one address
PRM:019	2	Selects IEEE-488 mode of Communication

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



Parameters PRM:008 and PRM:009 should usually use the same address (lower 5 bits). Normally, the upper bits do not change, nor should they change. For example, if the user desires to change the IEEE-488 device to #7, parameter PRM:008 should equal “0x27” and parameter PRM:009 should equal “0x47”.



To retrieve all the default parameter values, the user must perform a setup. Refer to Chapter 4: Software Interface and Installation to perform a setup.

After configuring the controller for IEEE-488 communication mode, host mode commands must be used in order to control the U100/U100i. These commands are special code formats translated by the controller. *For more information regarding the host mode commands, refer to Chapter 12: Host Mode Operation.*

The controller uses the service request line to request attention from the host controller while operating in the IEEE-488 communication mode. *For additional information on the service request line, refer to Chapter 5: Programming Commands.*

To verify that the IEEE-488 communication mode is operating, perform the following.

1. Before disconnecting the RS-232 device from the controller, verify that the RS-232 mode is not operational.



If RS-232 mode is operational, parameter PRM:019 may contain an incorrect setting.

2. Verify that parameter PRM:019 is set to 2.
3. Send a host mode command to the controller (preferably an immediate motion command).



If the LEDs on the U100 are blinking, this may indicate that the U100 IEEE-488 interface is not operational.

Descriptions of the error codes are in Appendix D of this manual. The status registers used by the controller are in Chapter 7: Registers and Variables.



IEEE-488 basic programming examples are in Chapter 8: Sample Programs

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.6. After Factory Installation of Option Boards

In most cases the desired option(s) will already be installed in the controller according to the specifications provided by the user. However, each optional interface card can be installed in the field. Mounting of an interface card is typical regardless of the option desired. Each board mounts to four #3-40 x 3/4" standoffs. The standoffs screw into controller metalwork through mounting holes in the control board. Two #3-40 x 1/2" screws secure the rear of the board to the standoffs through mounting holes in the DIN connector of the board. Two #3-40 x 1/4" secure the front of the board to the standoffs through mounting holes in the option board. Figure 3-19 shows the typical installation of an optional interface card.

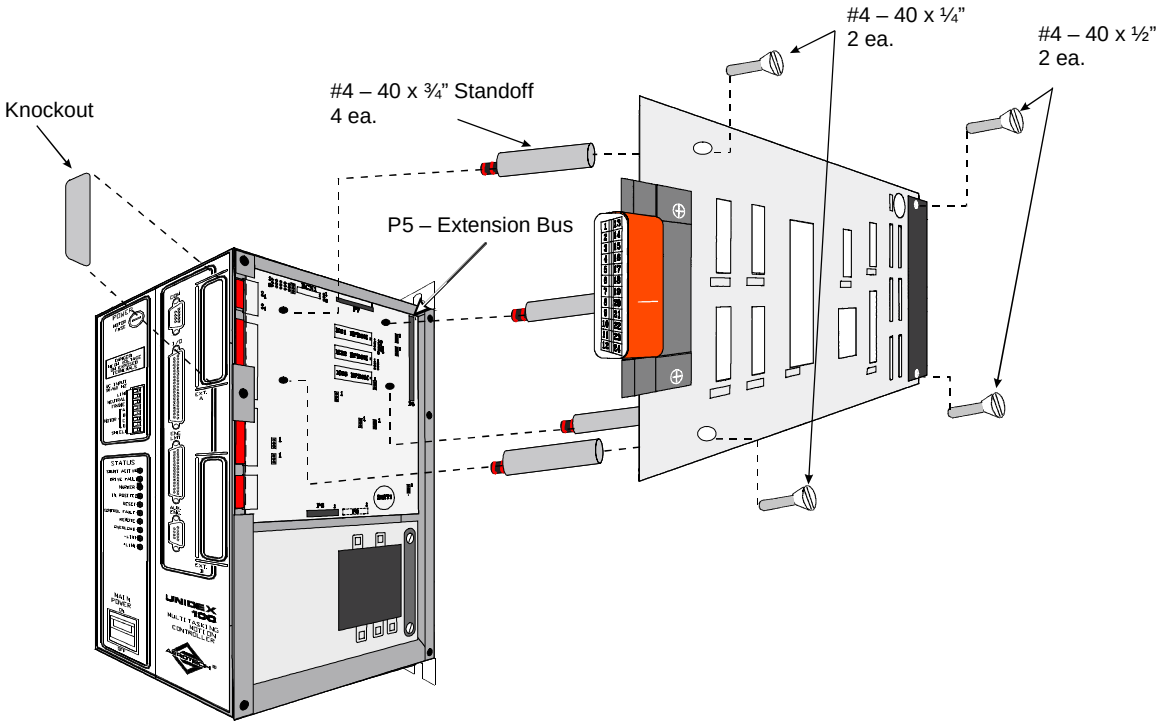


Figure 3-19. Typical Installation of Optional Interface Card

A 64 pin ribbon cable (supplied with option) connects the interface card to the controller extension bus (P5).

A maximum of two option boards can be mounted to the U100 control board. The second board mounts below the first (not shown in Figure 3-19).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



If two boards are required, consult Aerotech, Inc. for wiring specifications.



The U100i has only one option board capability.



630B1541 is the 64 pin ribbon cable shipped standard with all options. For installing 2 options, consult factory for wiring.

∇ ∇ ∇

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CHAPTER 4: SOFTWARE INSTALLATION AND INTERFACE

In This Section:

- Introduction 4-1
- Installing the Software Package..... 4-1
- Overview of the Main Menu Window 4-5
- Controller Setup Process U100/U100i 4-26
- Using the Hand held Terminal (HT)..... 4-27

4.1. Introduction

This chapter explains how to install and run the UT100 software that is part of the UNIDEX 100/U100i system. This chapter also discusses basic features such as pull-down menus, and text windows. Sample screens are illustrated to provide an understanding of the menu structure and software interface. Provided is an explanation of the operational interface between the Hand held Terminal (HT) and the controller. Important software configuration information is also discussed in this chapter.

4.2. Installing the Software Package

The operator uses the software package as the interface to the UNIDEX 100/U100i motion controllers. This software program (which is stored on one floppy diskette) needs to be copied onto the hard disk drive. This software installation process requires less than 5 minutes to complete.

4.2.1. The Installation Disk

The controller software package is stored on one double-sided, double-density floppy disk that is labeled "UNIDEX 100" or "UNIDEX 100i". The software includes an HT terminal emulator (com_100.exe), a parts program simulator, and a C language function library. This section describes how a user can communicate with the controller using the com_100.exe terminal emulator program. The purpose of this program is to execute programs or individual commands, and check the controller status information through a menu-driven interface.

Also included in the UT100 Utility Software package is a general information text file called "READ.ME" and a "PGM100" file.

PGM100 is a controller executable file that the user can download through the "com_100.exe" utility program. The file contains one or more parameter statements (e.g., PRM:133=2). This file includes a list of all the parameters that Aerotech changed to match the configuration found on the customer order form. This file lists all parameters that deviate from the default settings.

In later versions of the U100/U100i, a file labeled PRM100 (not to be confused with the parameter PRM:100) may be included in the UT100 Utility Software disk. This file (if present), performs the same function as the file PGM100 except it uses the utility "host_100.exe" as its downloading mechanism to the controller. The replacement of the PGM100 file with the PRM100 file, to hold factory setup parameters, was generated because of a problem running the PGM100 file in the controller to set the parameters

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

when a fault condition was present. If a fault on the controller was present while running the PGM100 program, the parameters would not be registered because of the way the exception processing masks the controller are set in the default operating mode. The "host_100.exe" is unaffected by any fault conditions on the controller.

Fore more information on this file view the "READ.ME" text file stored on the UT100 Utility Software disk.

A summary of the file types contained on the installation disk is given in Table 4-1.

Table 4-1. Summary of Files on the Software Installation Diskette

File Name	Description/Contents
com_100.EXE	PC terminal emulation program. Also allows selection of the 19200 and 38400 baud rate operation.
READ.ME	This is a general information text file included with UT100 Utility software package.
PGM100	Lists all parameters that deviate from the default setting to match the configuration provide by the customer.
BOOT_100.EXE	Allows the entire U100 operating system to be transferred from the PC into flash ROM. Allowing firmware changes to be in the field.
MCOM_100.EXE	PC terminal emulation program for multiple controllers configured in RS-232 daisy-chain mode. Also allows selection of the 19200 and 38400 baud rate operation.
HOST_100.EXE	Utility for communicating with the controller in “HOST” mode (C language source code included). Permits user data stored in flash ROM to be uploaded or downloaded to the PC. Also allows selection of the 19200 and 38400 baud rate operation.
TUNE_100.EXE	Windows based executable file used for tuning the PID control algorithm. Also allows selection of the 19200 and 38400 baud rate operation.
QBIE_100.EXE	Utility for communicating with the U100 in “HOST” mode using Quick Basic (Basic language source code included).
PRM100	Performs the same function as the file “PGM100” except it uses the utility "host_100.exe" as its downloading mechanism to the controller.

Before the software is installed, the system PC must meet or exceed certain minimum requirements for proper operation. These requirements are listed in Table 4-2.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 4-2. Minimum Hardware Requirements and Recommendations

Equipment	Minimum	Recommended
Computer	IBM PC AT, PS/2, or 100% compatible	IBM PC AT, PS/2, or 100% compatible
Microprocessor	80486	80486 or higher
RAM	4 MB	4 MB or more
Graphics Display	EGA or VGA	EGA or VGA
Hard Disk Space	4 MB	5 MB or more
Mouse	Any supported mouse	Any supported mouse
Floppy Disk Drive	3.5” DSHD	3.5” DSHD
Windows Software	Version 3.1 or newer	Version 3.1 or newer
Operating System	DOS 5.0	DOS 5.0 or higher

4.2.2. Standard Installation

To install the software, the operator must follow the steps listed below.

1. Connect the RS-232-C cable between COM1 or COM2 on the PC and the COM port (P1) on the controller, Refer to Figure 4-1.

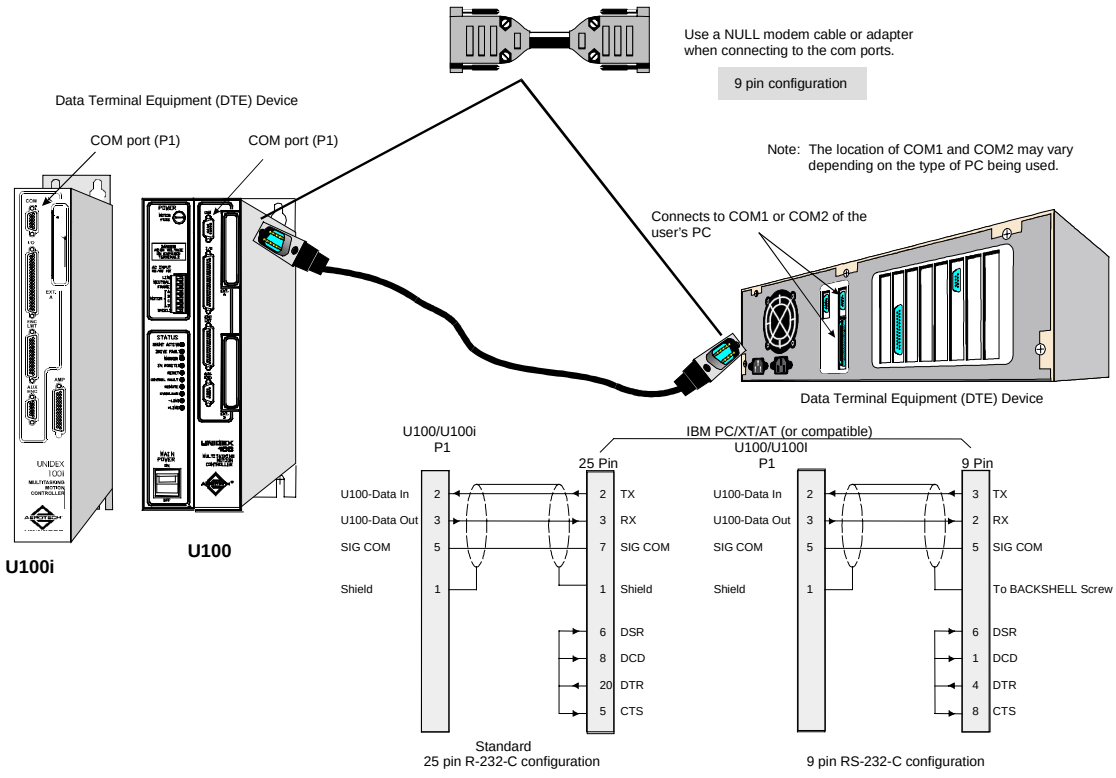


Figure 4-1. RS-232-C Connection Diagram

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Since the UNIDEX 100/U100i are Data Terminal Equipment (DTE) devices, the user must connect to the com ports of controller and PC with a NULL modem cable or adapter when using the 9 pin configuration. This will ensure proper connection of the transmit (TX) and receive (RX) signals. Refer to Figure 4-1.

2. Using an editor, add the following command to the CONFIG.SYS file.

   DEVICE=C:\DOS\ANSI.SYS

where the "C" represents the boot drive and DOS is the directory where the ANSI.SYS resides.

3. Insert the UT100 disk into the 3.5” floppy disk (e.g., drive A:\).
4. Copy all files from the floppy disk drive to the hard drive. For example:

   COPY A*.* to C:\U100*.*

   If a drive other than A:\ is used, substitute the appropriate letter.
5. Run program (UT100 software) from the PC by typing “com_100.exe 1 (if connected to COM1 port). If running from COM2 port, type “com_100.exe 2”.

   The screen shown in Figure 4-2 appears on the PC.

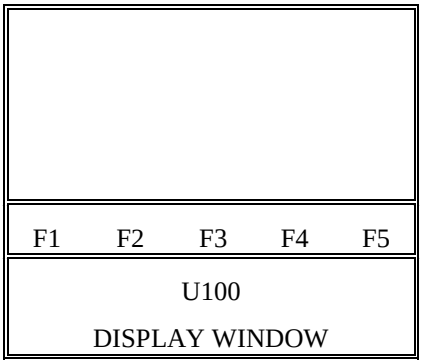


Figure 4-2. Software Display Window

6. Turn on the U100 power switch and the Main Menu window in Figure 4-3 appears.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

PRESS FOR MODE TYPE:	
A=FILE	B=RUN
C=MDI	D=STATUS
E=CONTROL	F=MISC
F1 F2 F3 F4 F5	
U100 DISPLAY WINDOW	

Figure 4-3. Main Menu Window (power applied to U100/U100i)

1. Type "D" to get into the Status Menu and another "D" to display the Version Menu. The Version Menu returns the software version number installed in the controller.

4.3. Overview of the Main Menu Window

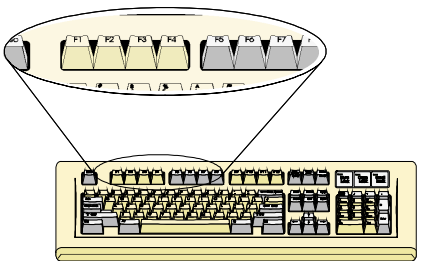
After the software package is installed and initiated, the Main Menu window is shown on the monitor. The Main Menu window illustrated in Figure 4-4 provides six selections to choose from for the user.



Figure 4-4. Main Menu Window

The Main Menu window contains the following selections:

- File
- Run
- MDI
- Status
- Control
- Misc.



Each of these is discussed in the following sections.

In the middle of the Main Menu window is a row of function keys, F1 through F5. These function keys correspond with the function keys on the user's keyboard. If the user is in another window and wants to return to the Main Menu window, pressing F1 on the keyboard returns the user to the Main Menu. Pressing F5 returns the user to the previous menu.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



The function keys F2, F3, and F4 have no function when moving between menus, but do have functions when editing a file. Their functions are discussed in another section later in this chapter.

4.3.1. The File Edit Menu

The File Edit menu is accessed by selecting “A” (FILE) from the Main Menu window, refer to Figure 4-5. The file edit function permits the user to create new files as well as edit existing ones. The data entered into the file is the same data displayed on the screen. The screen can display up to three rows by twenty columns of file data without horizontal or vertical scrolling.



Figure 4-5. File Edit Window

To edit existing files stored in memory, select “A” (EDIT FILE) from the File Edit window and the Edit File Type window appears, shown in Figure 4-6.



Figure 4-6. Edit File Type Window

The user is then prompted to enter the desired file number in the Enter File Number window shown in Figure 4-7.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

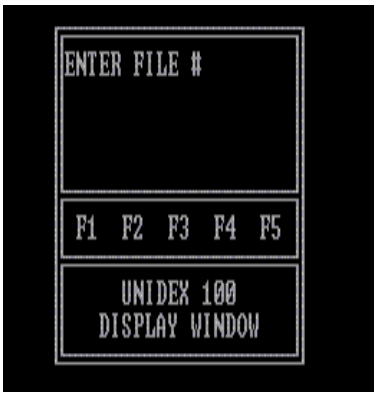


Figure 4-7. Enter File Number Window

In order to edit a selected file, the function keys F1 through F5 on the user's keyboard must be used. Table 4-3 lists the keys and their function while in the file edit mode. Figure 4-8 is an example of the edit window displayed on the user's monitor.



Figure 4-8. Edit File Window

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 4-3. Keyboard Function Keys and their Function

Function Key	Function	Description
F1	LF	Selecting F1 moves the cursor one position to the left.
F2	DN	Selecting F2 scrolls down one line at a time.
F3	UP	Selecting F3 scrolls up one line at a time.
F4	RT	Selecting F4 moves the cursor one position to the right.
F5	-more-	Selecting F5 provides another level of functions for function keys F1 through F5.
Level 2		
F1	IL	F1 in level 2 allows the user to insert a new line. The line to be inserted appears after the line that contains the cursor. (The insertion of a line is accomplished by placing the cursor above the line where the new will go and then pressing <Return> or <Enter>)
F2	DL	Selecting F2 will delete a line at the cursor's present location. After deleting a line, all subsequent lines scroll up one line in the file.
F3	P#	Selecting F3 allows the user to switch between pages in a file. After pressing F3, the user will be prompted to enter the desired page number to edit. **One page is equivalent to 100 lines of text. The user must switch between pages to edit information on a specific page.**
F4	L#	F4 allows the user to enter a line number and jump to it in an existing file. **However, if the user is working on page 1 of a file and desires to edit line 50 on page 2, the user must switch to page 2 first before using this function. For every new page in a file the line numbers do not carry over, but start over with number one.**
F5	-done-	F5 permits the user to exit and save all editing performed in the file. The user will be prompted by the U100 to either exit (Y) or continue (N). **If the user chooses "N" the function keys will return to their level one functions.**

The file directory permits the user to view a list of all programs in memory. The format used to list programs includes the type and number, refer to Figure 4-9. The directories also display titles for programs that have them. To access the proper menu for displaying directories, the user must select "B" (FILE DIRECTORY) from the File Edit Menu, refer back to Figure 4-5.

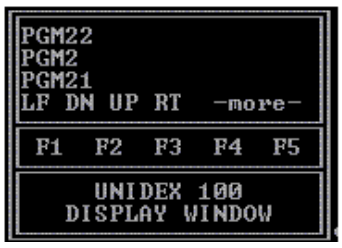


Figure 4-9. File Directory Window

If there are no files present in the controller directory, the controller prompts the viewer with a message "No Programs Found".

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

If the “TITLE” command was inserted into a PGM file, the ASCII text following the command will appear next to the PGM file name in that directory.



4.3.2. Run Selection Menu

To execute a program select “B” (RUN) from the Main Menu.

The controller only executes programs from PGM files.



The controller prompts the user to enter the task to be run from the Select Task window shown in Figure 4-10. The user must select “A” for Task 1 or “B” for Task 2. This is followed by the Select Running Mode window, refer to Figure 4-11. From this menu the user must select the running mode type. The running mode types available are Auto and Block mode. Auto mode executes the entire program at one time. Block mode walks the user through the entire program.



Figure 4-10. Select Task Window



Figure 4-11. Select Running Mode window

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3.2.1. Auto Mode

The selection of the Auto mode executes the selected program at once. After starting a program, the user may exit the present screen and perform other operations. One of these operations may be to perform a feedhold of the program in execution. To perform a feedhold of the program, the user must enter the Control mode. While in this window the user has the option to abort the initiated task. For an understanding of how this process works, refer to the Control Menu (Section 4.3.5).

4.3.2.2. Block Mode

The Block mode causes the controller to walk the user through the selected program. After keying in the program number, the controller returns to the Main Menu even though the program is executing in the Block mode. Due to the requirements of the Block mode, the user must select the Control Menu to step through the program entered. See Section 4.3.5. on the Control mode.

4.3.3. MDI Selection Menu

The MDI mode allows the user to enter a single command or change parameters in the Immediate mode. To access this mode, select "C" (MDI) from the Main Menu and the Select Task window appears, refer to Figure 4-12.



Figure 4-12. Select Task Window (MDI Mode)

After selecting the task, the user may enter either a command or parameter change into the Enter Command window, refer to Figure 4-13.



Figure 4-13. Enter Command Window (MDI Mode)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

If the user must edit the command while in this mode, the left, right, up, and down arrow keys do not function. Therefore, it is necessary to use the backspace function key to edit the command.

To execute the command entered in the window, press the <ENTER> key. While the command is executing, the controller displays the following message shown in Figure 4-14.



Figure 4-14. Executing Command Window

To exit the MDI mode at any time, press the F1 function key once or the F5 function key twice. This will return the user to the Main Menu.

Any time during the motion execution, the user may select to feedhold or abort the task. In order to do this, return to the Main Menu and select “E” (Control). Refer to Section 4.3.5. for more information on the Control Menu.

4.3.4. Status Selection Menu

The controller permits four types of status inquiries: Motion Status, General Status, Memory Status, and Version Status. To access the Status Selection window, select “D” (STATUS) from the Main Menu window and the Status Selection window shown in Figure 4-15 will appear.



Figure 4-15. Status Selection Window

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3.4.1. Motion Status Window

The Motion Status defaults to display the position and velocity. However, the user may change these words (position and velocity) by changing the contents of the variable SV:16 (for position) and SV:17 (for velocity). Parameters PRM:040 and PRM:041 select the data source that is displayed in the Motion Status window. Parameters PRM:036 through PRM:039 permit the user to adjust the number of digits and decimal location of these numbers. Seven digits is the default (refer to Figure 4-16).



Figure 4-16. Motion Status Window

4.3.4.2. General Status Window

The General Status window, shown in Figure 4-17, provides an overview of the U100's status. There are eight categories in this window. The status of the eight categories is indicated with a "Y" or "N". A "Y" or yes indicates that the status is true and "N" or no indicates the status is false. The +LIMIT is true when the axis is in the positive limit. The -LIMIT is true when the axis is in the negative limit. The MARKER status is true when the motor is on the marker position. ZERO status is true when the motor is in position. The ABSL is true when the U100 is in the Absolute mode. The OVLD status is true if the U100 has encountered a PID trap condition. The FAULT status is true if a fault condition has occurred. The final status is the MASK status and is true if any of the bits of the Active Mask register are true (see Exception Processing in Chapter 5: Programming Commands).



Figure 4-17. General Status Window

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3.4.3. Memory Status and Version Windows

The Memory Status window, shown in Figure 4-18, shows the amount of available words in memory. The Version Status window in Figure 4-19 displays the version of the software.



Figure 4-18. Memory Available Window



Figure 4-19. Software Version Window

4.3.5. Control Selection Menu

The Control Selection Menu permits the user to perform a limit reset or select the task window. To access the Control Select Menu, press "E" (CONTROL) from the Main Menu and the window in Figure 4-20 appears.



Figure 4-20. Control Select Window

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Accessing the Limit Reset Menu in Figure 4-21 causes the axis to move out of a limit.



Figure 4-21. Limit Reset Window

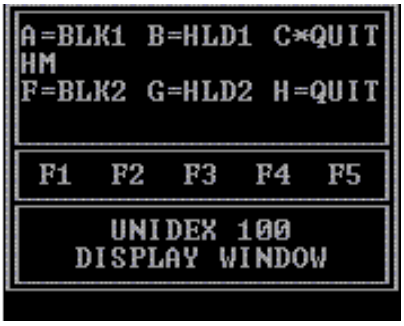


Figure 4-22. Task window

Accessing the Task Window Menu in Figure 4-22 allows the user to control Task 1 or Task 2.

Task 1 control provides block execution using BLK1, feedhold using HLD1, and quit capabilities using the associated QUIT. Block execute steps the user through a block run program each time the user presses "A". Select "B" to initiate a feedhold, then press “B” again to release the feedhold for Task 1. To quit Task 1, press "C".

Task 2 control provides block execution using BLK2, feedhold using HLD2, and quit capabilities using the associated QUIT. Block execute steps the user through a block run program each time the user presses "F". Select "G" to initiate a feedhold, then press “G” again to release the feedhold for Task 2. To quit Task 2, press "H".

The task window also displays the program lines of the task being executed. An asterisk before QUIT indicates a command or program is being executed in that task. The asterisk indicates a feedhold when it appears before HLD1 or HLD2. To indicate the Block mode the asterisk appears before BLK1 or BLK2.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3.6. MISC Selection Menu

The Miscellaneous Selection Menu allows the user to erase or copy files, assign new values to any of the four data types, transfer files between the controller and the user's PC, and archive parameters, variables, and/or user programs (PGM, DEF files, exe) to flash ROM. To access the MISC Selection Window, press "F" (MISC) from the Main Menu and the window in Figure 4-23 will appear.



Figure 4-23. MISC Selection Window

There is an additional option that the user can select (not shown in Figure 4-23), the option is "E=LIBRARY". This option will only appear on the MISC Selection window if the MEM option is installed.



4.3.6.1. Erase Files Menu

The U100 permits the user to erase certain files or all file types. The erase file(s) operation is accessed by pressing "A" from the Misc Selection window. The Erase File window in Figure 4-24 appears.



Figure 4-24. Erase File(s) Window

To erase a single file, select "A" or to erase all files, select "B". If erasing a single file, a file type must be selected from the File to Erase Selection window shown in Figure 4-25.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



Figure 4-25. File Type to Erase Selection Window



Figure 4-26. Erase All Files Selection Window

If a selection to erase all files is made, the user will be prompted to select “Y” (Yes) or “N” (No) shown in Figure 4-26 before erasing all files.

4.3.6.2. Copy Files Menu

The Copy Files Menu permits the user to copy information between files of the same type. The Copy File window is accessed by pressing “B” (COPY) from the MISC Selection window. The user must select the file type to copy from the Copy File Type Selection window, refer to Figure 4-27.



Figure 4-27. Copy File Type Selection Window

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

After selecting the file type an Enter from File and to File Window appears on the monitor. The user must enter the file number of the file to be copied. Likewise, for the "To File", the user must enter the file number.

Verify that the copy was successful by editing the new file. If the copy was unsuccessful, check for a wrong file number or repeat process.



4.3.6.3. Setup Data Types Menu

The Setup Menu permits the user to assign new values to any of the four data types. The new values override the default settings for each type specified. The data types that appear in the Setup Menu include parameters, registers, variables, and strings. Select "C" (SETUP) from the MISC Selection window and the Setup Menu window in Figure 4-28 appears.



Figure 4-28. Setup Selection Window

To change a parameter or group of parameters, select "A" (PARAMETERS) from the Setup Selection window. The window in Figure 4-29 appears.



Figure 4-29. Parameter Type Setup Window

The "xx" indicates that the user only has to enter the last two digits of the parameter to be changed after selecting the parameter type. Once the parameter type is selected, the Change Parameter window appears in one of two ways, refer to Figure 4-30 and Figure 4-31.

The parameter selected will determine which window appears.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

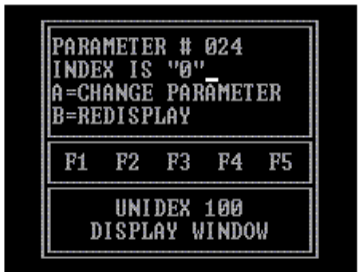


Figure 4-30. Change Parameter Window (1 of 2)

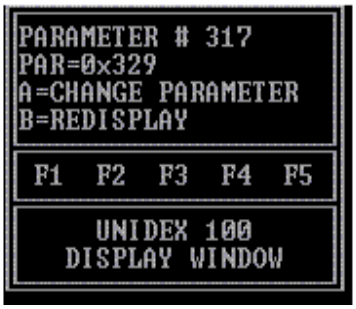


Figure 4-31. Change Parameter Window (2 of 2)



Figure 4-32. Change Parameter Setting Window

In Figure 4-32, selecting “U” increases the parameter value. To decrease the parameter value, select “D”. Selecting “B” redisplays the parameter value.



INDEX is “X” corresponds to the setting for a particular parameter (e.g., PRM:311 index is “2” [default setting]; hitting “U” will change setting to 3 or 4. Hitting “D” will change setting to 0 or 1).

To change a Read, Write, or a Read/Write register, select “B” from the Setup Selection window. The window that appears is the same as the Parameter Type Setup window in Figure 4-29.

If a selected register is a read/write register, the window in Figure 4-33 appears.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



Figure 4-33. Read/Write Selection Window



Figure 4-34. Write Only Register Change Window

Figure 4-34 applies to the write access of a register. The “A=Base” changes the initial hex value of register to the decimal equivalent. The “B=Load” loads the number entered on the screen into the register, refer to Figure 4-36.



Figure 4-35. Read Only Register Change Window

Figure 4-35 applies to the read access a selected register. Like the write access registers, the “A=Base” changes the initial hex value to the decimal equivalent. The “B=Redisplay” updates the value on the screen (this value could be changing internally).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



Figure 4-36. Load Register Window

To change the value of variables, select “C” (VARIABLES) from the Setup Selection window, in Figure 4-28. The window that appears is shown in Figure 4-37.



Figure 4-37. Variable Type Selection Window

The Enter Variable window in Figure 4-38 appears once the user selects the variable type. From this window the user enters the number of the variable to be changed.



Figure 4-38. Enter Variable Window

The Modify/Scan Select window shown in Figure 4-39 prompts the user to either modify or scan the variable entered. Scanning a variable means it is constantly being updated. If the user selects “B” (or Scan) the Scan Variable window in Figure 4-40 appears.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



Figure 4-39. Modify/Scan Select Window



Figure 4-40. Scan Variable Window

The controller denotes strings as “SV:xx” variables where “xx” is an integer between 1 and 19. To modify strings press “D” (STRINGS) on the Setup window and the Enter String window appears, refer to Figure 4-41.



Figure 4-41. Enter String Window

Once the user enters the string number to be modified, the Modify String window in Figure 4-42 appears.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



Figure 4-42. Modify String Window

To modify the string entered, select “A” (MODIFY) and the Enter New String window will appear, refer to Figure 4-43.



Figure 4-43. Enter New String Window



Strings SV:16, SV:17, SV:18, and SV:19 have special functions besides storing string data. Depending on the settings of PRM:040 and PRM:041, all or part of strings SV:16 and SV:17 is displayed on lines 1 and 2 of the Motion Status window (Section 4.3.4.1). Strings SV:18 and SV:19 always appear in lines 3 and 4 of the Motion Status window. This permits visual interaction with Task 1 and Task 2 without having to use PM () statement.

Transferring files is a term used for uploading and downloading files between the controller and the PC. The types of files that the controller permits the user to transfer include: programs, definitions, macros, list files, and directories.



List and directory files do not normally need transferred.

The controller transfers these files in two ways. First, is to take an existing file directly from the Host PC and down load that file to the controller. The second is to take an existing file from the controller's memory and upload that file to the PC. To perform one

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

of these two transfers, select “D” (TRANSFER) from the Setup window and the Transfer Select window in Figure 4-44 appears.



Figure 4-44. Transfer Select Window

Depending on the selection made in the Transfer Select window, either the Transfer File to controller window or the Transfer File to PC window in Figure 4-45 and Figure 4-46 respectively will appear on the user’s monitor.



Figure 4-45. Transfer File to U100 Window



Figure 4-46. Transfer File to PC Window

If the file exist, the controller overwrites it during the transfer process regardless of the direction of the transfer.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3.6.4. Archive Files Menu

The Archive Files operation is selected from the Miscellaneous Selection Menu by pressing "F", refer to Figure 4-23. Selecting "F" produces the menu in Figure 4- 47.

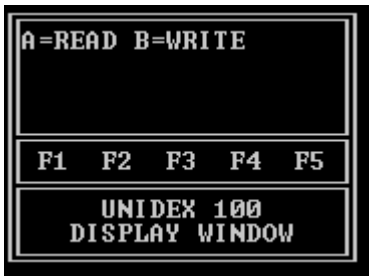


Figure 4-47. Read and Write from Archive Selection

Selecting "A" allows the user to manually copy data from an appropriate sector of flash ROM to a RAM area. Selecting "B" allows the user to write data from the RAM area to the appropriate sector in flash ROM. Whether the user selects "A" or "B", the menu selection in Figure 4-48 appears.



Figure 4-48. Machine and Library Selection Menu

Selecting "MACHINE IMAGE" (pressing "A" on the keyboard) allows the user access to the storage area for parameters, user programs, and variables. Selecting "LIBRARY IMAGE" (pressing "B" on the keyboard) allows the user access to one of four storage areas for backing up the contents of the MEM board.

Selecting either menu item produces the menu in Figure 4-49.



The "Library Image" section is only valid when using the "MEM" board option. Its use is not discussed in this manual.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



Figure 4-49. Enter Parameter (Machine and Library Image)

For the "Machine Image" section, the user types in a parameter to specify the type of read or write operation to be performed. They are as follows:

1. From the main menu, press "F" for the MISC screen, "F" for ARCHIVE, "B" for WRITE, and "A" for MACHINE IMAGE.
2. At the prompt for PAR=, type the following hexadecimal number for the type of backup required.

"0"	backup all variables, parameters, and user programs, but not rebooting them to RAM on power up.
"0x1"	backup all variables, parameters, and user programs only allowing **BV:1** through **BV:1000** to reboot to RAM on power up.
"0x2"	backup all variables, parameters, and user programs only allowing **LV:1** through **LV:200** to reboot to RAM on power up.
"0x4"	backup all variables, parameters, and user programs only allowing **FV:1** through **LV:1000** to reboot to RAM on power up.
"0x8"	backup all variables, parameters, and user programs only allowing user programs (**PGM**x, **CAM**x, etc.) to reboot to RAM on power up.
"0x10"	backup all variables, parameters, and user programs only allowing **SV:1** through **SV:19** and parameters PRM:0xx, PRM:1xx, PRM:2xx, PRM:3xx to reboot to RAM on power up.
"0x20"	backup all variables, parameters, and user programs only allowing variables on the MEM board (optional) to reboot to the MEM on power up (i.e., PV:xx variables and "LIBRARY" programs.

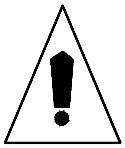
Any or all combinations of the numbers listed above can be used. For example, 0x11 allows both **BV** and **SV** variables to reboot on power up.

Writing to "flash" ROM takes about five to eight seconds per sector. During this period, the UNIDEX 100 communications are suspended. However, the operating kernel (i.e., PID loop, fault masks, etc.) are still operational.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.4. Controller Setup Process U100/U100i

The user may initialize the controller with the hardware setup input. This setup loads all the parameters with the default values. It is necessary to do a setup for each controller firmware update, corrupt parameters, or when having difficulty establishing communications. Also, each time the user updates the firmware or has had trouble with memory, it is good practice to erase all files within memory.



Changing certain key parameters, such as Brushless Commutation (PRM:239 through PRM:241), after the last setup requires that the user reconfigure them before applying motor power. Downloading and running the PGM100 file restores the controller parameters to their original factory settings.



To minimize the possibility of electrical shock and bodily injury, make certain that all of the electrical power switches are in the off position prior to making any electrical connections.

To perform a setup, do the following.

1. Turn off power switch (U100 only).
2. Place a jumper between pins 19 (setup-n) and 16 (common) of the I/O connector (P2) on the U100/U100i.
3. Place a jumper between pins 20 (input common) and 30 (5V) of the I/O connector (P2) on the U100.



Only if pin 20 is not connected to an external power supply.

4. Power up the U100/U100i for approximately 5 seconds.
5. Remove power.
6. Remove jumpers from the I/O connector (P2).

The U100/U100i is setup with the parameters set at the default settings.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.5. Using the Hand held Terminal (HT)

The Hand held Terminal (HT) is an optional operator interface that allows the user to control the U100/U100i through menu assisted screens. This terminal plugs into the communications COM port (P1) on the controller, refer to Figure 4-50. The HT works over the RS-232-C interface and receives its power from the controller.

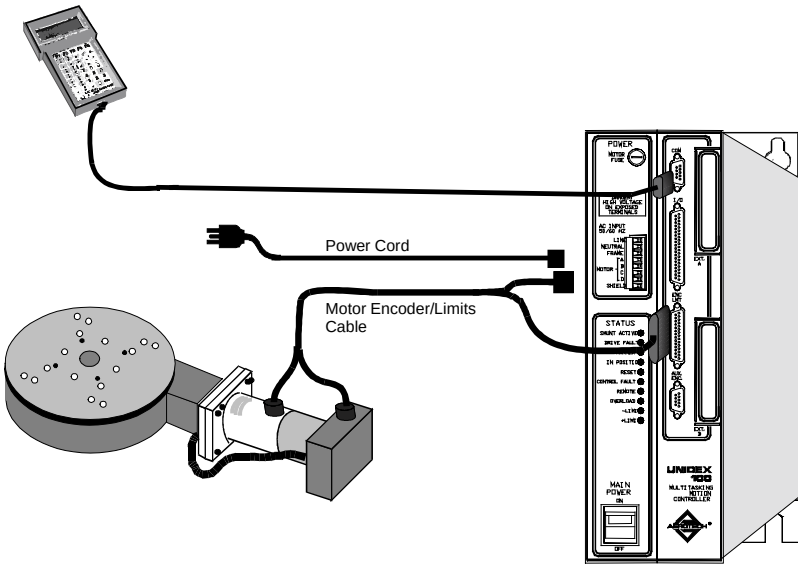


Figure 4-50. HT to U100/U100i Configuration

The HT can be used to edit programs, run programs, run commands, and check controller status information.

To operate the HT, plug the HT into the U100 COM (P1) and turn on the U100 power switch. The menu in Figure 4-51 appears on the HT.

PRESS FOR MODE TYPE:	
A=FILE	B=RUN
C=MDI	D=STATUS
E=CONTROL	F=MISC

Figure 4-51. HT Menu

The menu selections and functions on the HT are the same as those discussed in the previous sections when interfacing the controller with a PC.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

There are some features about HT the user should know before operating the controller and they are:

- The shift key must be held to change from lower case to upper case or to access secondary keys such as the "=" operator
- The F1 function key like on the keyboard returns the user to the Main Menu
- The F5 function key returns the user to the previous menu
- Pressing <CTRL> +<D> resets the controller

∇ ∇ ∇

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CHAPTER 5: PROGRAMMING COMMANDS

In This Section:

- Introduction 5-1
- Multitasking Operating System 5-3
- Interrupts 5-8
- Exception Processing 5-9
- Service Request 5-14
- U100/U100i Supported File Types 5-17
- Programming Commands 5-24

5.1. Introduction

The UNIDEX 100/U100i decodes the programming commands using the entire characters of the command. The commands are not case sensitive, except for the compiler directive commands, they must be entered with capital letters. Throughout this chapter the commands appear in uppercase letters for easy recognition.

This chapter uses the typographical conventions listed in Table 5-1.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 5-1. Programming Conventions Used in This Manual

Example	Description
BV:xx LV:xx FV:xx PV:xx REG:xx PRM:xx BV:BV:xx LV:BV:xx FV:BV:xx PV:BV:xx	Terms <var1>, <var2> variables can be any one of these forms unless a given command indicates otherwise. “xx” is the specific index number.
integer constant fixed point constant float point constant	The term <constant> can be any one of these forms unless a given command indicates otherwise.
SV:xx	The term <string> can take the following forms. Note, that “xx” is a number from 1 to 19.
“this is a string”	Maximum of 20 characters, must have beginning and ending quotes.
<description>	Can be any set of characters up 80 in length. Quotation marks (“”) are not necessary.
<number>	Refers to the linking of an integer number to a command (e.g., MAC<number> same as MAC22).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.2. Multitasking Operating System

The controller incorporates real-time multitasking capabilities that allow execution of more than one operation at a time. The user can perform tasks such as running multiple programs, user interaction, and exception processing.

5.2.1. Multitasking Time Slots

To achieve real-time Multitasking requires the use of dedicated and selectable time slots. The controller time slot contains a minimum of 0.1 millisec. To complete an entire multitasking cycle takes 12.8 milliseconds (128 times 0.1 millisec. time slots). The Kernel Interrupt Time Slot Chart (Table 5-3) shows the dedication of each time slot.

The first column in the time slot chart is the time slot number. This number begins at 0 msec. and ends with 12.7 msec. (12.8 begins the repeat of 0). Column 2 is the task type (PRM:301 and PRM:302) and shows which of the three tasks (T-0, T-1, or T-2) are currently being worked on. The main purpose behind Task 0 is to carry out communications and background operations. Task 1 and Task 2 do program and command execution and are adjustable with PRM:301 and PRM:302. The system dedicates the remaining time for Task 0 use.

Column 3, the Servo Update Time (PRM:304), shows the time intervals in which the servo loop gets updated. The servo loop gets updated every 0.8 millisec by default. The Commutation Loop Update Time (PRM:305) is the next column and updates every 0.4 millisec. The Stepper Loop Update Time column (PRM:306) reads Encoder 1 and gets updated every 0.1 millisec. The next three columns are the Encoder 2 Update Time (PRM:308), the ENC Update Time (PRM:309), and the R/D Update Time (PRM:310). These three parameters define no time slots by default. The Fault Mask Update Time (PRM:311) is the next column and is responsible for Exception Processing that executes every 6.4 millisec. The Control Loop Checks (PRM:312) column performs over every 12.8 millisec and is responsible for PID Loop Trap and General Error Processing. The Scaled Trajectory External Port Fetch (PRM:303) gets updated every 6.4 millisec.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 5-2. Kernel Interrupt Time Slot Description

	Controlling Parameters											
KERNEL INTERRUPT TIME SLOT	**301 302**	**304**	**305**	**306**	**307**	**308**	**309**	**310**	**311**	**312**	**303**	
0 msec	T-0											
0.1 msec	**T-1**											
0.2 msec	**301**											
0.3 msec	" "											
0.4 msec	" "											
0.5 msec	" "											
0.6 msec	" "											
0.7 msec	" "											
0.8 msec	" "											
0.9 msec	" "											
1.0 msec	" "											
1.1 msec	T-0											
1.2 msec	" "											
1.3 msec	" "											
1.4 msec	" "											
1.5 msec	" "											
1.6 msec	" "											
1.7 msec	" "											
1.8 msec	" "											
1.9 msec	" "											
2.0 msec	" "											
2.1 msec	" "											
2.2 msec	" "											
2.3 msec	" "											
2.4 msec	" "											
2.5 msec	" "											
2.6 msec	" "											
2.7 msec	" "											
2.8 msec	" "											
2.9 msec	" "											
3.0 msec	" "											
3.1 msec	" "											

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 5-2. Kernel Interrupt Time Slot Description

	Controlling Parameters											
KERNEL INTERRUPT TIME SLOT	**301 302**	**304**	**305**	**306**	**307**	**308**	**309**	**310**	**311**	**312**	**303**	
3.2 msec	" "											
3.3 msec	**(1)**											
3.4 msec	" "											
3.5 msec	" "											
3.6 msec	" "											
3.7 msec	" "											
3.8 msec	" "											
3.9 msec	" "											
4.0 msec	" "											
4.1 msec	" "											
4.2 msec	" "											
4.3 msec	T-0											
4.4 msec	" "											
4.5 msec	" "											
4.6 msec	" "											
4.7 msec	" "											
4.8 msec	" "											
4.9 msec	" "											
5.0 msec	" "											
5.1 msec	" "											
5.2 msec	" "											
5.3 msec	" "											
5.4 msec	" "											
5.5 msec	" "											
5.6 msec	" "											
5.7 msec	" "											
5.8 msec	" "											
5.9 msec	" "											
6.0 msec	" "											
6.1 msec	" "											
6.2 msec	" "											
6.3 msec	" "											
6.4 msec	" "											
6.5 msec	**T-1**											
6.6 msec	**301**											
6.7 msec	" "											

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 5-2. Kernel Interrupt Time Slot Description (Cont'd)

KERNEL INTERRUPT TIME SLOT	301 302	304	305	306	307	308	309	310	311	312	303	
6.8 msec	" "											
6.9 msec	" "											
7.0 msec	" "											
7.1 msec	" "											
7.2 msec	" "											
7.3 msec	" "											
7.4 msec	" "											
7.5 msec	T-0											
7.6 msec	" "											
7.7 msec	" "											
7.8 msec	" "											
7.9 msec	" "											
8.0 msec	" "											
8.1 msec	" "											
8.2 msec	" "											
8.3 msec	" "											
8.4 msec	" "											
8.5 msec	" "											
8.6 msec	" "											
8.7 msec	" "											
8.8 msec	" "											
8.9 msec	" "											
9.0 msec	" "											
9.1 msec	" "											
9.2 msec	" "											
9.3 msec	" "											
9.4 msec	" "											
9.5 msec	" "											
9.6 msec	" "											
9.7 msec	**(2)**											
9.8 msec	" "											
9.9 msec	" "											
10.0 msec	" "											
10.1 msec	" "											
10.2 msec	" "											
10.3 msec	" "											
10.4 msec	" "											

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 5-2. Kernel Interrupt Time Slot Description (Cont'd)

	Controlling Parameters											
KERNEL INTERRUPT TIME SLOT	**301 302**	**304**	**305**	**306**	**307**	**308**	**309**	**310**	**311**	**312**	**303**	
10.5 msec	" "											
10.6 msec	" "											
10.7 msec	T-0											
10.8 msec	" "											
10.9 msec	" "											
11.0 msec	" "											
11.1 msec	" "											
11.2 msec	" "											
11.3 msec	" "											
11.4 msec	" "											
11.5 msec	" "											
11.6 msec	" "											
11.7 msec	" "											
11.8 msec	" "											
11.9 msec	" "											
12.0 msec	" "											
12.1 msec	" "											
12.2 msec	" "											
12.3 msec	" "											
12.4 msec	" "											
12.5 msec	" "											
12.6 msec	" "											
12.7 msec	" "											

5.2.2. Multiple Programs

The controller permits execution of two programs simultaneously allowing the user to perform more than one operation at the same time. Applications include interrupts and interactive programs.

The procedure for executing two programs at once is the same as the one used for a single program. Select and execute the first program, then select and run the second program in the remaining task. Programs are initiated immediately after loading them and do not start together.

Interrupts must use Task 2.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

To synchronize the start of the first program with the second, use a variable as a control flag between the two programs.



Because of interaction between programs, some commands may not perform as intended.

Multiple programs should not execute motion commands at the same time.

Programs should not use the same variables unless they are transferring information between programs.

5.3. Interrupts

Programming interrupts allow the user to initiate a program when a user defined interrupt takes place. The interrupt is selected, defined, and controlled through a combination of parameters, registers, and program commands. Parameters PRM:132 through PRM:135 (see Chapter 6: Parameters) are needed to configure the interrupts.

Most interrupt applications require the use of an interrupt program, which is a PGM file that is loaded into Task 2 and executed upon receipt of the interrupt. An example interrupt program is shown below with the interrupt parameter values used for this program.

EXAMPLE:

```
TITLE**Task 1**          ; Attach the title to the program number.
BEGIN                    ; Start the program and wait for interrupts
EI                       ; Enable the external interrupt.
BV:1=1                   ; Set the integer, variable #1 equal to one.
D(5)                     ; Set the distance equal to five.
V(2)                     ; Set the velocity equal to two.
GO                       ; Start the move.
END                      ; End the program.
```

```
TITLE**Task 2**          ; Attach the title to the program number.
BEGIN                    ; Start the program
LV:BV:1=REG:113          ; Evaluate the value found in register #113,
                         ; then equate that value to integer, variable #1
                         ; and put final result in the LV variable.
BV:1=BV:1+1              ; Set integer variable equal to integer,
                         ; variable 1 plus one.
EI                       ; Enable the external interrupt.
END                      ; End the program.
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Parameter values for the previous programs.

PRM:133=1 "Task 2 interrupt"

PRM:134=4 "latched, edge low"

PRM:135=2 "External Interrupt"

The controller contains three interrupt related program commands that allow the user to enable (EI), disable (DI), and reset (RI) interrupt operation. The enable interrupt command (EI) enables (and resets a latched interrupt) for the program running in Task 2. The disable interrupt command (DI) disables execution of the interrupt operation even though an interrupt occurred. The reset interrupt command (RI) resets a latched interrupt and enables the interrupt.

The user may choose to configure interrupts for latching the value of a user selectable source into REG:113. This mode is useful for obtaining a position on the fly, immediately after the occurrence of the interrupt. Parameter PRM:132 selects the source to latch.

5.4. Exception Processing

Exception processing is the process that configures the controller to perform certain operations dependent on the settings of certain Error Status register bits. This operation allows the user to select the response for a given fault that best suits the application. This process occurs at least once every 12.8 milliseconds and depends on the setting of PRM:311 (Fault Mask Update Time). The following sections describe each step of the Exception Process, refer to Figure 5-1.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

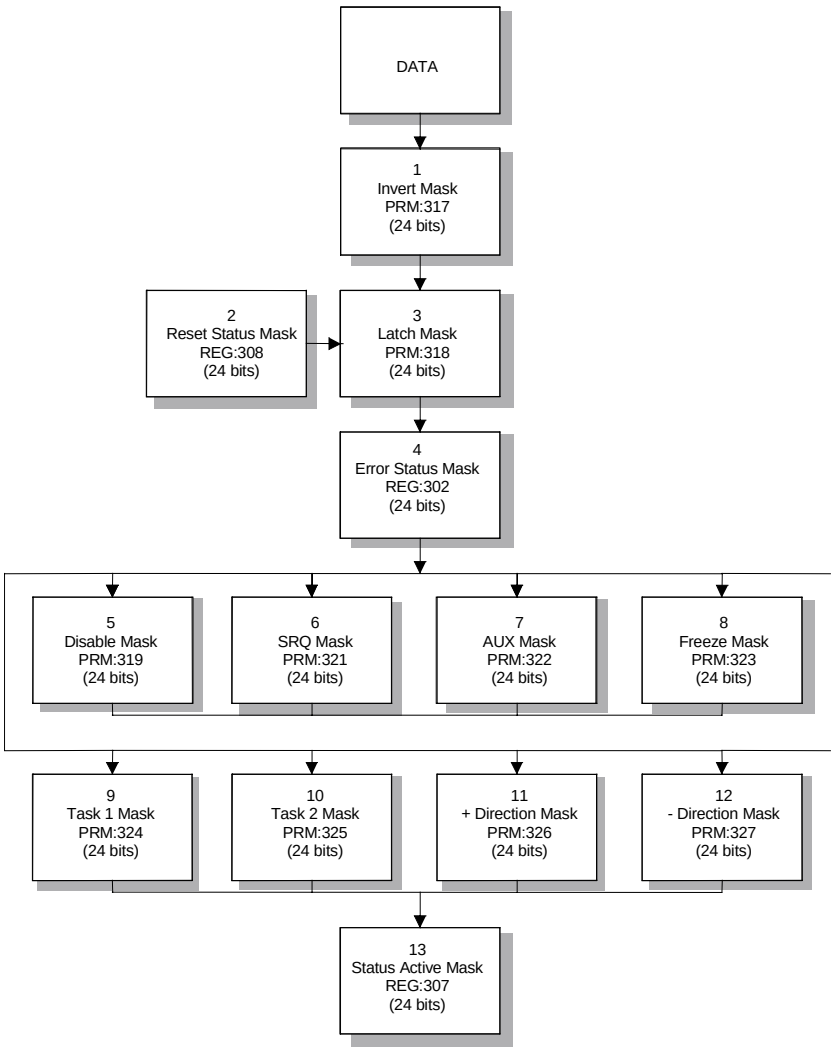


Figure 5-1. Exception Processing Flow Chart

5.4.1. Invert Mask

The invert mask parameter (PRM:317) inverts the polarity of selected data bits. The controller inverts the data bit if the associated invert mask bit setting contains a one. Do not change PRM:317 unless a prior change occurred regarding the polarity of the hardware related faults (e.g., change limit switch polarity). The invert mask bit description is the same as the error status register.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.4.2. Reset Status Register

The reset status register (REG:308) selectively resets the latch mask bits. The latch mask bits get reset when the associated reset status register bits are set to one. The reset status register bit description is the same as the error status register.

5.4.3. Latch Mask

The latch mask parameter (PRM:318) selects what error status bits to latch. The processed error status bit latches when the associated latch mask parameter bit is a one. The latch mask parameter bit description is the same as the error status register.

5.4.4. Error Status Register

The error status register (REG:302) is a 24-bit register containing error and fault status data for the controller. The following bit data applies.

Table 5-3. Bit Definitions for REG:302

BIT #	Definitions
BIT 0	Encoder 1 Feedback Error
BIT 1	Encoder 2 Feedback Error
BIT 2	Drive Fault
BIT 3	User Defined Input 2 (default executes a "fast feedhold")
BIT 4	ENC Feedback Error
BIT 5	User Defined Input 1 (default disables power amplifier)
BIT 6	+ Soft Limit Encountered
BIT 7	- Soft Limit Encountered
BIT 8	+ Hard Limit Encountered
BIT 9	- Hard Limit Encountered
BIT 10	Reserved
BIT 11	R/D (Resolver/Digital) Feedback Error
BIT 12	Position Error Trap
BIT 13	Velocity Feedback Trap
BIT 14	Accel/Decel Feedback Trap
BIT 15	Current Limit Time-out Trap
BIT 16	User Defined Software Trap 1 (default stops Task 1)
BIT 17	User Defined Software Trap 2 (default stops Task 2)
BIT 18	User Defined Software Trap 3 (default disables the power amplifier)
BIT 19	Run Time Error occurs in Task 1
BIT 20	Run Time Error occurs in Task 2
BIT 21	Axis Calibration Range Error
BIT 22	Reserved
BIT 23	Reserved

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.4.5. Disable Mask

The disable mask parameter (PRM:319) selects the conditions that should disable the drive. The disable mask parameter bits get "ANDed" with the error status bits. If any of the resultant bits test true, the controller disables the drive (turns off the amplifier power stage). The disable mask bit description is the same as the error status register.

5.4.6. SRQ Mask

The SRQ mask parameter (PRM:321) selects the conditions for sending a Service Request (SRQ). The SRQ mask parameter bits get "ANDed" with the error status bits. If any of the resultant bits test true, the controller sends a Service Request . The SRQ mask bit description is the same as the error status register.

5.4.7. Auxiliary Mask

The auxiliary mask parameter (PRM:322) selects the conditions for setting the auxiliary output. The auxiliary mask parameter bits get "ANDed" with the error status bits. If any of the resultant bits test true, the controller sets the auxiliary output. The auxiliary mask bit description is the same as the error status register.

5.4.8. Freeze Mask

The freeze mask parameter (PRM:323) selects the conditions for executing a "fast" feedhold. The freeze mask parameter bits get "ANDed" with the error status bits. If any of the resultant bits test true, the controller executes a feedhold. The freeze mask bit description is the same as the error status register.

5.4.9. Task 1 Mask

The Task 1 mask parameter (PRM:324) selects the conditions for performing a Task 1 Exception Operation. The Task 1 mask parameter bits get "ANDed" with the error status bits. If any of the resultant bits test true, the controller initiates a Task 1 Exception Operation. The value set in PRM:124 determines the process for the Task 1 Exception Operation. The Task 1 Mask bit description is the same as the error status register.

5.4.10. Task 2 Mask

The Task 2 Mask parameter (PRM:325) selects the conditions for performing a Task 2 Exception Operation. The Task 2 mask parameter bits get "ANDed" with the error status bits. If any of the resultant bits test true, the controller initiates a Task 2 Exception Operation. The value set in PRM:125 determines the Task 2 Exception Operation. The Task 2 Mask bit description is the same as the error status register.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.4.11. Positive Direction Mask

The positive direction Mask parameter (PRM:326) selects the conditions for inhibiting positive direction motion. The controller "ANDs" the positive direction parameter bits with the error status bits. If any of the resultant bits test true, the controller prevents motion in the positive direction. A common use of this mask replaces the limit mask in joystick applications where a limit occurrence must not terminate the program. The positive direction mask parameter bit description is the same as the error status register.

5.4.12. Negative Direction Mask

The negative direction mask parameter (PRM:327) selects the conditions for inhibiting negative direction motion. The controller "ANDs" the negative direction parameter bits with the error status bits. If any of the resultant bits test true, the controller prevents motion in the negative direction. A common use of this mask replaces the limit mask in joystick applications where a limit occurrence must not terminate the program. The negative direction mask parameter bit description is the same as the error status register.

5.4.13. Status Active Register

The status active register (REG:307) bits to the true state if the controller initiates any of the exception operations. This register contains the status of blocks five through twelve (1 is active). The following status active register bit data applies.

Table 5-4. Bit Definitions for REG:307

BIT #	Description
BIT 0	Disable Mask
BIT 1	Not Used
BIT 2	Service Request Mask
BIT 3	Auxiliary Mask
BIT 4	Freeze Mask
BIT 5	Task 1 Mask
BIT 6	Task 2 Mask
BIT 7	+ Direction Mask
BIT 8	- Direction Mask

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.5. Service Request

The controller uses the service request to ask for attention from the Host Controller when operating in RS-232 or IEEE-488 Host communication modes. Upon sending out the service request, it is necessary for the Host controller to acknowledge it with the correct reply. Failure to acknowledge the service request results in the appearance of a lock up of the controller.

The service request consists of the service request character followed by a SRQ status code and finally the complement of the SRQ status code. The SRQ character is user definable where it can be adjusted through PRM:028. For IEEE-488 operation, the service request mechanism is built into the interface through a mode known as "Serial Poll". However, for both RS-232 and IEEE-488, the status codes are the same. The controller sends multiple SRQ's on an individual basis with each one requiring a service request acknowledge. The SRQ status code provides information concerning the service request.

For RS-232 mode, the sequence is as follows:

<SRQ character> <SRQ status> <compliment of SRQ status>

For IEEE-488, the SRQ status loads directly into the SRQ register.

There are specific functions that determine the SRQ status (see the following).



The prefix "0x" denotes Hexadecimal (base 16) integers.

Table 5-5. Task 0 Service Request Status Codes

Codes	Definition
0x00	Reserved
0x01	Program Uploading in Progress
0x02	Program Downloading in Progress
0x03	Host Mode Command Processing Error
0x04	Memory Allocation Error (out of user memory)
0x05	"Line #" Update for "Task 1 Window" Running in the RS-232 Host Mode only
0x06	"Line #" Update for "Task 2 Window" Running in the RS-232 Host Mode only
0x07	Ready for Commands in Host Mode (RS-232 only)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 5-6. Task 1 Service Request Status Codes

Codes	Definition
0x10	Reserved
0x11	User Defined #1
0x12	User Defined #2
0x13	User Defined #3
0x14	User Defined #4
0x15	User Defined #5
0x16	User Defined #6
0x17	User Defined #7
0x18	User Defined #8
0x19	User Defined #9
0x1A	User Defined #10
0x1B	User Defined #11
0x1C	User Defined #12
0x1D	User Defined #13
0x1E	User Defined #14
0x1F	User Defined #15

The controller command "SRQ(<number>)" generates the codes in Table 5-7. The <number> is an integer 1 through 15 for status codes 0x11 through 0x1F respectively.

Table 5-7. Task 2 Service Request Status Codes

Codes	Definition
0x20	Reserved
0x21	User Defined #1
0x22	User Defined #2
0x23	User Defined #3
0x24	User Defined #4
0x25	User Defined #5
0x26	User Defined #6
0x27	User Defined #7
0x28	User Defined #8
0x29	User Defined #9
0x2A	User Defined #10
0x2B	User Defined #11
0x2C	User Defined #12
0x2D	User Defined #13
0x2E	User Defined #14
0x2F	User Defined #15

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The controller generates the codes in Table 5-8 in the same manner as that used in Task 1 for status codes 0x20 through 0x2F respectively.

Table 5-8. Kernel Service Request Status Codes

Codes	Definition
0x30	Reserved
0x31	Initiate "Xon" (allow RS-232 Host to transmit)
0x32	Initiate "Xoff" (stop RS-232 Host from transmitting)
0x33	SRQ Exception Mask encountered a "true" condition



IEEE-488 Service Request Status Codes may contain Bit 6 set. (e.g., Status Code 0x01 may be received as 0x41).

In the RS-232 mode, it is necessary for the Host Controller to send the Service Request Acknowledge (SA) to the controller after receiving a Service Request (SRQ). For the IEEE-488 mode the "Serial Poll" accomplishes the same task. Parameter PRM:029 permits the user to select the SA character for the RS-232 mode. Failure to send the Service Request Acknowledge causes the controller to get hung up and not respond to commands. A Service Request Acknowledge sent to the controller without a previous SRQ does not cause a problem.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.6. U100/U100i Supported File Types

The controller supports five different file types and they are as follows:

- PGM files
- DEF files
- MAC files
- LST files
- and CAM files

All file types require that their name contain a controller fixed prefix and a supporting constant (e.g., PGM22). Without the assigned prefix, the controller does not recognize the file being use. Each of these files perform different functions, however, the PGM files are the most important, since these files contain the instructions executed by the controller. The other file types support the PGM files.

5.6.1. Program Files (PGM)

The PGM file is a list of instructions based on the controller programming commands. Each program used with the controller must contain a "PGM" prefix followed by a constant ranging from 1 to 9999 (e.g., PGM<number>, PGM123 and do not include ending zeros in names [i. e., use **PGM1** instead of **PGM01**]). The PGM files are referred to as the main program. Multiple programs permit a maximum of 300 lines total between Task 1 and Task 2. This includes the "MAC" files, refer to Figure 5-2.

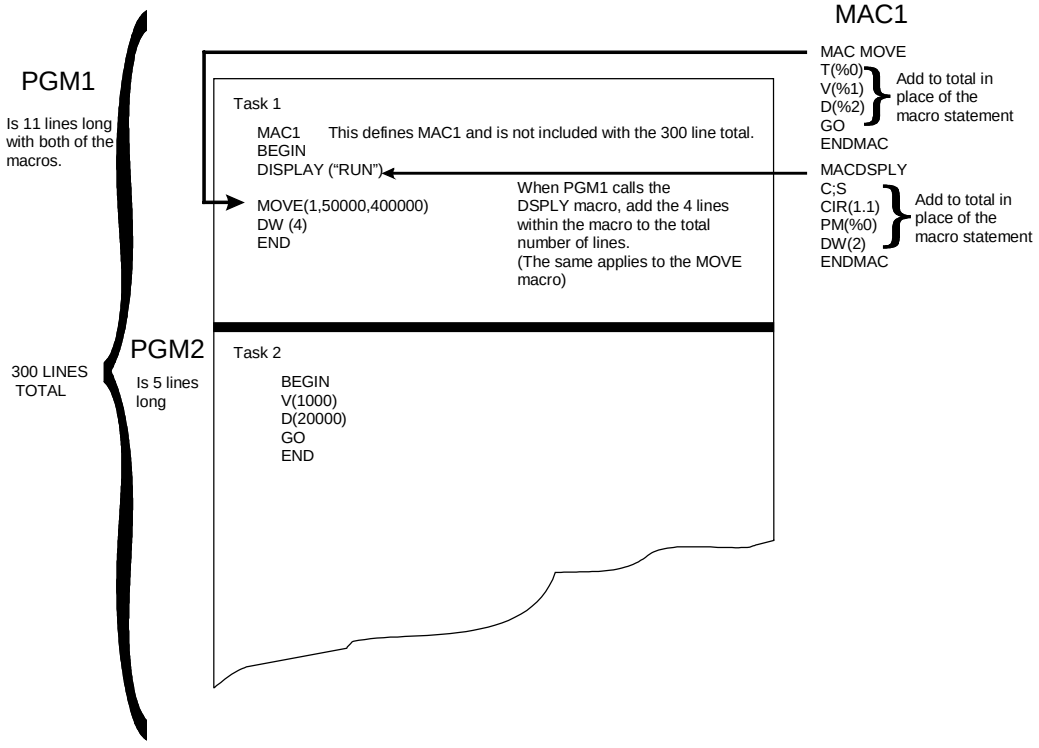


Figure 5-2. Example of Maximum Lines in PGM Files

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

It is necessary to enclose the list of commands in between a BEGIN and an END statement. The execution of these commands starts precisely after the BEGIN statement.



Programs generated directly through the controller do not need the “PGM” prefix added to them.

All programs generated without the controller must have the “PGM” prefix added to the name.

PGM files must follow certain guidelines regarding program structure. Table 5-7 is a template that illustrates this structure.

Table 5-9. PGM File Template

Statement	Description
TITLE <description>	*This statement must be the first statement in the PGM file. The “TITLE” statement is optional and used only to identify a program use when viewing the file directory. It must be capitalized.*
MAC<number> DEF<number>	;optional comment field ;optional comment field *The statements above instruct the compiler to search the specified MAC and/or DEF files to resolve macros and definitions that may be contained in the program body. Up to four MAC files and DEF files may be specified. Both of these statements must be capitalized.*
BEGIN	*This statement indicates the beginning of an executable program. It is required by all PGM programs and must be capitalized.*
BV:1=BV:2+BV:3	;optional comment field *The body of the controller command start after the BEGIN statement. These statements can be upper or lower case.*
EXIT	;optional command field *This statement is optional and is required only if subroutine(s) are included in the PGM program. If no subroutines are present, the “END” statement acts as the end of program execution terminator.*
SUB:<constant> ENDSUB	;optional comment field *The statement above provides delimiters for subroutine calls where “constant” is the number of the subroutine. They are required only if subroutine calls are made from the statement body.*
END	*This is the last statement in a PGM program and provides two purposes. First, as an end of compile delimiter when the program is being compiled for execution. Secondly, as an end of execution for simple programs that do not incorporate subroutine calls. It must be capitalized.*

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.6.2. Macro Files

Macro files or “MACS” are modules made up of programming commands that are called by the main program using a user defined macro. This macro is defined before the execution of the main program. After completing the compiling process, the macro program replaces the statement used to call the macro with the macro commands.

The user cannot use the assigned macro name for anything else in the program.

Once compiled, the “LST1” file allows the user to view the expanded code.



Macro statements permit the user to pass variables, constants, and strings to a program that uses arguments. When identifying an argument in a macro, include a “%” sign followed by an argument number (0 through 9).

The *controller* allows a maximum of ten arguments in MACRO defining.



Each macro contained within a MAC file must start with a MAC name statement. A MAC file may contain more than one macro section. However, because of memory restrictions the *controller* may not be able to handle large MAC files. The macro is a code replacement module and not a subroutine even though, in some sense, the function is similar. Since there are limitations to the number of lines a program may contain, a complete understanding of macros and their use must be fully considered. This understanding will prevent exceeding the program capabilities of the *controller*. Refer to the following examples, the first is the main program that calls two macros from the second example which is a MAC file.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

EXAMPLE:

```
TITLE**This is PGM 25**      ; Attach the program description to the
                             ; program number.
MAC1                         ; Define MAC file #1 so that the program may
                             ; use the macro programs contained therein.
BEGIN                        ; Start the program.
DSPLY("RUN")                 ; Call the display macro from the macro file.
                             ; The argument contains the text string.
MOVE(1,50000,400000)         ; Call the motion macro from the macro file.
                             ; Where argument 1 is time, 2 is velocity, and 3
                             ; is distance.
DW(4)                        ; Set the dwell time equal to four seconds.
END                          ; End the program.
```

```
MAC MOVE                     ; The macro program name is MOVE.
T(%0)                        ; Time is specified by the first argument in the
                             ; Main Program.
V(%1)                        ; Velocity is specified by the second argument
                             ; in the Main Program.
D(%2)                        ; Distance is specified by the third argument in
                             ; the Main Program.
GO                           ; Start the move.
ENDMAC                       ; End the macro called MOVE.

MAC DSPLY                    ; The macro program name is DSPLY.
CLS                          ; Clear the terminal screen.
CUR(1,1)                     ; Position the cursor at row 1, column 1.
PM(%0)                       ; Display the message denoted by the first
                             ; argument in the Main Program.
DW(2)                        ; Set the dwell time equal to 2 seconds.
ENDMAC                       ; End the macro DSPLY.
```



Comments (;) are not allowed in MAC files. The example above is for the readability of the syntax only.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.6.3. List files (LST)

List files are generated each time a program gets compiled, provided the setting of PRM:25 is set to "1". If PRM:25 is set to zero (0), then the list file generation is suppressed. The file name given to list file is always LST1 and is overwritten each time the controller compiles a program. The information contained in this file consists of the compiler code and the user may save this file under a new name for later referral.

One of the advantages provided by the list file is it allows the user to view the expanded code brought into the program from the MAC and DEF files. An example of a list file generated from a program that called in macros is shown below. The main program is shown first.

EXAMPLE:

```
TITLE**This is PGM 25**          ; Attach the program description to the
                                 ; program number.
MAC1                             ; Define MAC file #1 so that the program may
                                 ; use the macro programs contained therein.
BEGIN                            ; Start the program.
DSPLY("RUN")                     ; Call the display macro from the macro file.
                                 ; The argument contains the text string.
MOVE(1,50000,400000)             ; Call the motion macro from the macro file.
                                 ; Where argument 1 is time, 2 is velocity, and 3
                                 ; is distance.
DW(4)                            ; Set the dwell time equal to four seconds.
END                              ; End the program.
```

```
CLS
CUR(1,1)
PM("RUN")
DW(2)

T(1)
V(50000)
D(400000)
GO

DW(4)
END
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.6.4. Definition Files (DEF)

Definition files assign names to variables or common strings. These files must be declared in the main program before the "BEGIN" statement. After compiling a program that uses a DEF file, the user defined variables in the program are replaced with the variables in the DEF file.

The format of a definition file does not require a "BEGIN" and "END" statement. Refer to Section 5.7.20 for a detailed description of the DEF command used to call definition files within a program. The following is an example of a DEF file called "DEF99". The program below it is an example that illustrates the use of the file "DEF99".

EXAMPLE:

```
TRAVEL=FV:1          ; Define TRAVEL as a floating point variable.
APART=FV:2           ; Define APART as a floating point variable.
OUTPUT=FV:3          ; Define OUTPUT as a floating point variable.
```



Comments (;) are not allowed in DEF files. The example above is for the readability of the syntax only.

```
DEF99                ; Open Definition File DEF99.
BEGIN                ; Start the program.
CLS                  ; Clear the terminal screen.
CUR(1,1)             ; Position the cursor at row 1, column 1.
PM("Enter length ")  ; Print the message to the terminal screen.
TRAVEL=GM(20)        ; Get the input from the terminal and place the
                     ; result in the floating point variable TRAVEL.
CUR(2,1)             ; Position the cursor at row 2, column 1.
PM("How far apart? ") ; Print the message to the terminal screen.
APART=GM(20)         ; Get the input from the terminal and place the
                     ; result in the floating point variable APART.
CUR(3,1)             ; Position the cursor at row 3, column 1.
OUTPUT=TRAVEL/APART  ; Set output equal to the quotient of floating
                     ; point variable TRAVEL divided by floating
                     ; point variable APART.
PM("Total outputs ") ; Print the message to the terminal screen.
PM(OUTPUT,"%.2f")    ; Print the result of floating point variable
                     ; OUTPUT, but suppress the zero's to two
                     ; places following the decimal.
END                  ; End the program.
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Defined names should not be substrings of other names
(e.g., X=BV:1 and X1=BV:10) these would cause an error.



5.6.5. CAM Files

CAM files store position data for cubic spline trajectory generation. These files contain up to two columns of numerical data. The first column contains points for connecting the internally generated cubics spline curve. The second column contains optional output data that is sent to a specified port to the positions specified in the first column. The following is an example of a CAM file.

EXAMPLE:

3.345e-3	0x10	Floating point format for the first column (position data).
6.34565	0x20	Integer format for the second column (output data).
10.423	0x50	
12.345	0x50	Comments can be replaced any time after the first (or
15.598	0x40	second) column of data. “White” spaces (e.g., “tab” or
11	0x30	“space”) are used to separate the first and second data
8.5698	12	columns as well as the comment field. A “newline” character
3.8901	10	must terminate each line.
1.234e-1	2	
1.44e-2	1	
0	10	

There are no blank lines separating the individual rows.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7. Programming Commands

The UNIDEX 100/U100i support many programming commands. These commands are listed in Table 5-10 and explained in detail in the sections that follow. There are essentially five types of commands used in a PGM program file. These include: compiler directives, communication commands, math and logical operations, general motion commands, and program control commands.

Table 5-10. UNIDEX 100/U100i Programming Commands

Command	Page	Description
Compiler Directives		Provide the compiler with processing instructions such as directives for searching "DEF" and "MAC" files for resolving definitions and expanding macros.
BEGIN	5-33	Identifies the start of a program
DEF	5-48	Assigns names to variables or commands
END	5-55	Identifies the end of a program
ENDMAC	5-57	Denotes the end of a macro
MAC MAC<number> MAC<space><macro name>	5-74	Denotes a macro file
TITLE<space><description>	5-92	Used to attach a description of the program
Communication Commands		Communication commands perform operations such as: transferring messages, clearing screens, and entering data.
CLN	5-38	Clears the screen from cursor to column
CLS	5-39	Clears entire screen
CUR	5-41	Moves the cursor to a specific area on the screen
GC (Get Character)	5-62	Takes inputted character and converts it to its numerical value
GM (Get Message)	5-63	Gets messages entered thru the terminal and displays them on the screen
PC (Print Character)	5-78	Works in conjunction with PM to allow non-printable characters to be displayed
PM (Print Message)	5-79	Prints messages, outputs strings to the display
General Motion Commands		General motion commands are used by the controller to initiate motion. These commands allow the user to home and index the axis. Quit often, execution of these commands occurs while in the MDI mode of operation.
A	5-27	Sets the acceleration/deceleration ramp speed
ABSL	5-29	Causes all moves following this command to be interpreted as position moves
ADC	5-30	Reads analog input and converts it to digital equivalent
CAM	5-33	Allows a sequence of data points to be connected together in smooth curve
D	5-42	Represents the distance in user units to move/position
DAC	5-45	Digital to analog conversion
DD	5-46	Allows the user to enter a trajectory command directly to the drive

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 5-10. UNIDEX 100/U100i Programming Commands (Continued)

Command	Page	Description
DW (Dwell Time)	5-52	Acts as a programmable wait
GEAR	5-62	Accepts feedback information from either of the two encoder ports or the optional R/D board and executes motion
GO	5-64	This command initiates motion, either incremental or absolute
HM	5-67	Sends the axis to the hardware home position
INCR	5-71	Causes all moves following this command to be interpreted as distance
T	5-90	Sets the time required for the acceleration and deceleration ramp rate
TUNE	5-93	Allows the controller to automatically choose the controller gains
V	5-94	Sets the velocity for the associated commands in a motion block
Program Control Commands		Provide the ability to make decisions, interrupt, loop, branch, and execute subroutines. These commands provide a large selection of logical decisions.
DI	5-50	Disables the external interrupt that had been previously enabled
EI	5-52	Enables the external interrupt
ELSE	5-53	Provides an alternative to executing an IF statement
ELSEIF	5-54	Used in combination with IF and ELSE to make another test
ENDIF	5-56	Ends the IF statement when the statement is false
ENDSUB	5-58	This command ends a subroutine task
ENDWHL	5-59	Indicates the end of a “While” loop
EXIT	5-60	Allows the operator to quit a program
GOSUB	5-65	Instructs the program to perform a specific subroutine
GOTO	5-66	Allows the user to jump to a label in parts program
IF	5-69	Test for a specific condition and executes if statement is true
LB	5-72	Labels a position within a parts program
RI	5-82	Resets the controller interrupt latch
RUN	5-83	Allows a program in Task 1 to automatically load and run a specified library program in Task 2
SRQ	5-85	Allows the controller to request the attention of the master controller
SUB	5-87	Identifies the beginning of a subroutine
SYNC	5-89	Used to synchronize a command to its task
WHL	5-95	Performs a logical test of two operands and if true, executes the subsequent commands
Math and Logical Operators		Math and logical operations are numerical operators used by the controller to perform math functions using variables and constants. **The operation of compound statements is not possible. Therefore, operator precedence does not apply.**
ABS (Absolute Value)	5-28	Calculates the absolute value of integers, long integers, floating point, and hexadecimal numbers

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 5-10. UNIDEX 100/U100i Programming Commands (Continued)

Command	Page	Description
+ (Add)	5-31	Permits addition of integers, long integers, floating point, and hexadecimal numbers
AND (Logical And)	5-32	Permits the user to set a variable equal to the value of two other variables
= (Assign a value to)	5-60	Assigns the value from the right of the "=" to the left of the "="
CBI	5-36	Converts a binary coded decimal number to an integer or float
CIB	5-37	Converts an integer to a binary coded decimal
COS (Cosine)	5-40	Converts a floating point number to the cosine value of an angle in radians
DEC (Decrement)	5-47	Subtracts one from a specified variable
/ (Divide)	5-51	Allows division of integers, long integers, floating point, and hexadecimal numbers.
INC (Increment)	5-70	Adds one to a specified variable
MDX (Modulo Index)	5-76	Allows the user to retrieve data from an array
* (Multiply)	5-77	Allows multiplication of integers, long integers, floating point, and hexadecimal numbers
OR (Logical OR)	5-78	Allows the user set the value of a variable to one of two other variables
SIN (Sine)	5-84	Converts a floating point number to the sine value of an angle in radians
SQRT	5-86	Returns the square root value based on some constant or variable
- (Subtract)	5-88	Permits subtraction of integers, long integers, floating point, and hexadecimal numbers
TAN	5-91	Converts a floating point number to the tangent value of an angle in radians
XOR (Exclusive OR)	5-96	Permits the user to "exclusive or" two binary numbers
Miscellaneous		This a list of miscellaneous commands used with U100/U100i
ARCH (Archive)	5-32	Allows parameters, variables, and/or user programs to be stored in or retrieved from flash ROM
LOCK	5-73	Enables and disables multitasking on the controller. When issued, freezes task switching
PVI	5-81	Allows retrieval of data stored in one of the four MEM board image sectors of the flash ROM, accessible by a user program

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.1. A Command (Accel/Decel Ramp Rate in User units/sec$^{2)}$

The **A** command allows the user to set the acceleration/deceleration ramp rate. This command is an alternative to the ramp time "T" command. The user must enter the ramp rate in units/sec/sec. The basis for this command is to execute velocity changes at a constant rate. This command is very useful for obtaining motion optimization as well as to minimize the motor's overshoot.

The controller accepts variables as well as parameters with this command. The value defined with this command remains in effect until entry of a new ac/de command. Use of this command does not overwrite the associated default parameter PRM:106.

It is possible to change the parameters default from within the program, but this change is global and affects all programs using the ac/de command.

Normal use of this command is in conjunction with the D, V, and GO commands to perform motion.

SYNTAX: *A(<var or constant>)*

EXAMPLE:

```
BEGIN           ; Start the program.
HM              ; Initialize the home position to zero.
A(20000)        ; Set the ramp rate equal to 20000 units per sec2.
D(1000000)      ; Move a distance of 1000000 positive units.
V(50000)        ; Set the velocity equal to 50000 units/sec.
GO              ; Start the move from the home position.
END             ; End of the program.
```

Related Commands

D, T, V, GO

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.2. ABS Absolute Value

The **ABS** command calculates the absolute value of integers, long integers, floating point, and hexadecimal numbers.

The controller truncates the decimal portion of an expression that contains a fraction if it is being equated to an integer variable (e.g., BV:1 or LV:1=ABS(-3.25)).

SYNTAX: *<var>=ABS(<var or constant>)*

EXAMPLES:

```
BEGIN                   ; Start the program.
BV:1=-10                ; Set integer, variable 1 equal to a negative ten.
BV:2=ABS(-50)           ; Set integer, variable 2 equal to the absolute
                        ; value of a negative fifty.  The result is fifty.
BV:3=ABS(BV:1)          ; Set integer, variable 3 equal to the absolute
                        ; value of integer, variable 1.  The result is ten.
END                     ; End the program.
```

```
BEGIN                   ; Start the program.
LV:1=-20                ; Set long integer, variable 1 equal to a negative
                        ; twenty.
LV:1=ABS(LV:1)          ; Set long integer, variable 1 equal to the
                        ; absolute value of the pre-defined long integer,
                        ; variable 1.  The result is twenty.
BV:1=ABS(-5.75)         ; Set integer, variable 1 equal to the absolute
                        ; value of a negative five and seventy-five
                        ; hundreds.  The result is five.
END                     ; End the program.
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.3. ABSL Absolute Positioning Mode

The **ABSL** command causes all moves following this command to be interpreted as absolute position moves. These moves always start at the last established software position. The controller calculates the move, by taking the difference between the present position and the commanded position. If the present position is larger than the command position the result is a negative counterclockwise (CCW) move. Since absolute mode calculates a move based on position, consecutive moves with the same distance (D) values do not result in any motion (e.g., 2000-2000=0). An alternative to this command would be the "INCR" (incremental) command positioning mode.

SYNTAX: *ABSL*

It is not possible to change the absolute mode while executing motion.



EXAMPLE:

```
BEGIN           ; Start the program.
HM              ; Initialize the home position to zero.
ABSL            ; Initialize the absolute (position) mode.
V(20000)        ; Set the velocity equal to 20000 units per
                ; second.
D(100000)       ; Move to position 100000 in the positive CW
                ; direction.
GO              ; Start the move from the home position.
DW(5)           ; Set the dwell time to 5 seconds.
V(30000)        ; Set the velocity equal to 30000 units per
                ; second.
D(250000)       ; Move to position 250000 in the positive CW
                ; direction.
GO              ; Start the move form the previous position
                ; of 100000. The move is equal to the
                ; commanded position of 250000 minus the
                ; present position of 100000 and therefore only
                ; moves 150000 steps.
END             ; End of the program.
```

Related Commands

INCR

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.4. ADC Analog to Digital Conversion

The **ADC** command reads the analog input port and places a value representative of that input into a user selected variable. The value loaded into the variable is determined by the input voltage, the analog to digital scale factor parameter (PRM:140), and the analog to digital deadband parameter (PRM:141). The analog to digital scale factor parameter (PRM:140) permits the user to select the maximum value that is equated to an input of 10 volts. The Deadband parameter (PRM:141) allows the user to select a portion of the 10 volts for use as a zero volt input. The variable value is the ratio of the input voltage to 10 volts (minus any deadband) multiplied by the A/D scale factor.

SYNTAX: *<var>=ADC*

EXAMPLE:

```
BEGIN                   ; Start the program.
WHL(REG:1 EQ 0Xff)      ; While register 1 is equal to 0Xff loop around.
                        ; Otherwise, break the loop.
FV:1=ADC                ; Obtain the scaled number for the analog input.
  DD(FV:1)              ; Output the trajectory command directly to the
                        ; drive.
ENDWHL                  ; End the while loop.
END                     ; End the program.
```

Related Commands

DAC

To perform the necessary calculations assume that parameter PRM:140 (ADC 10 Volt reference) is equal to 1.0, parameter PRM:141 (Deadband Fraction) equals 0.0, and the analog voltage input is -5.0 volts.

Calculate the Deadband Voltage using the following method

Dead Band Voltage = 10.0V x (PRM:141)

Dead Band Voltage = 10.0V x 0.0

Dead Band Voltage = 0

Calculate the ADC Value using the following method.

ADC# = {(Analog Volts in - Dead Band Volts) / (10.0V - Deadband Volts)} x PRM:140

ADC# = {(-5.0 Volts - 0 Volts) / (10.0 - 0 Volts)} x 1.0

ADC# = -.5

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.5. + Add

Choosing to use the + (add) function allows for addition of integers, long integers, floating point, and hexadecimal numbers. Expressions may also contain variables and therefore it is possible to add both numbers and variables in any combination.

The controller truncates the decimal portion of an expression that contains a fraction if it is being equated to an integer variable (e.g., BV:1 or LV:1=3.25+5.5).

SYNTAX: *<var or constant>+<var or constant>*

EXAMPLES:

```
BEGIN                   ; Start the program.
LV:1=5                  ; Set long integer, variable 1 equal to five.
LV:2=10                 ; Set long integer, variable 2 equal to ten.
LV:3=LV:1+25            ; Set long integer, variable 3 equal to the sum of
                        ; long integer, variable 1 and twenty five.  The
                        ; result is thirty.
END                     ; End the program.
```

```
BEGIN                   ; Start the program.
BV:1=5                  ; Set integer, variable 1 equal to five.
PV:1=10.5               ; Set port, variable 1 equal to ten and a half.
BV:2=BV:1+PV:1          ; Set integer, variable 2 equal to the sum of
                        ; integer, variable 1 and port, variable 1.  The
                        ; result is fifteen.
END                     ; End the program.
```

Related Commands

-, *, /, =

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.6. AND (Logical And)

The **AND** function permits the user to AND two binary numbers.

The expression may contain a number, a variable, or a combination of both. The numbers contained in the expression may be of type integer or long integer.

SYNTAX: *(<var or constant>AND<var or constant>)*

EXAMPLE:

```
BEGIN                   ; Start the program.
BV:1=2                  ; Set integer, variable 1 equal to two.
BV:2=2                  ; Set integer, variable 2 equal to two.
BV:3=BV:1 AND BV:2      ; Set integer, variable 3 equal to the value of
                        ; variable 1 AND variable 2.. The result is two.
                        ; For example: binary 2 AND binary 2 = binary 2
END                     ; End the program.
```

Related Commands

OR, XOR

5.7.7. ARCHIVE (ARCH ()) Command

The **ARCH** command allows parameters, variables, and/or user programs (PGM, DEF files, etc.) to be stored in or retrieved from flash ROM.

It is also possible to copy up to four separate images of the optional MEM board memory to *flash* ROM.

SYNTAX: *ARCH (<var or constant>,<BV: or constant>)*

Where:

Parameter 1 specifies “1” for READ or “2” for WRITE.

Parameter 2 specifies the “boot status” when performing a WRITE or “transfer type” when performing a READ. (Note: No matter what “boot status” is specified for a WRITE operation, all variables, parameters, and programs are copied to flash ROM).

“boot status” and “transfer type” are defined as follows and can contain any or all of the selections shown below:

0x0 – no data to be “booted” on power up.
0x1 – BV:1 through BV:1000 booted or transferred.
0x2 – LV:1 through LV:200 booted or transferred.
0x4 – FV: 1 through FV:1000 booted or transferred.
0x8 - PGM, CAM, DEF, et. booted or transferred.
0x10 – SV:1 through SV:19 and parameters PRM:0xx, PRM:1xx, PRM:2xx, PRM:3xx booted or transferred.
0x20 – PV:0x200 through PV:0xdfff or Library files booted or transferred (requires MEM board option).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.8. Begin Command

The **BEGIN** command identifies the start of a program. The first command following this command is the starting line of the main body.

This command is case sensitive, use capital letters when entering this command.



Failure to start a program with a BEGIN statement results in a pre-compiler error #101.



SYNTAX: *BEGIN*

EXAMPLE:

```
BEGIN          ; Start the program.
V(1000)        ; Set the velocity equal to one thousand.
D(20000)       ; Set the distance equal to twenty thousand.
GO             ; Start the move.
END            ; End the program.
```

Related Commands

END

5.7.9. CAM Command

The **CAM** command allows a sequence of data points to be connected together in a smooth curve and executed as position relative to some pre-defined reference source. The reference for this position data can be made to represent time or, can be synchronized to an external source such as an encoder.

To fully understand the operation of the **CAM** command, parameters PRM:153 through PRM:163 should be reviewed. Additionally, Appendix C: Spline Generation should be reviewed. This section provides information relative to the theory of operation of the **CAM** command along with additional information on how parameters PRM:153 through PRM:163 can be used for tailoring the **CAM** command to a specific applications.

The **CAM** command can be used in one of two ways depending on the size of the position data point array. First, if specified in the form **CAM(**<file number>), where "file number" corresponds to the ASCII file **CAM**<file number> stored in user memory; position data in this file is automatically translated and stored in the "FV" variable array.

Second, if PRM:162 is set to a value other than zero, output data is also scanned in this file and stored in the "BV" variable array.

The following is an example of a **CAM(**<file number>**)** file. The first column of data represents the spline position data loaded into the "FV" variable array starting with FV:1. The second column represents the output data loaded in the "BV" variable array starting with BV:1.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

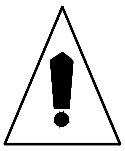


If no output data is contained in the **CAM** file, parameter PRM:162 must be set to zero, or a processing error will occur. However, if output data is specified in the second column and PRM:162 is set to zero, no processing error will occur.

SYNTAX: *CAM(<var or constant>)*

EXAMPLE:

3.345e-3	0x10	Floating point format for the first column (position data).
6.34565	0x20	Integer format for the second column (output data).
10.423	0x50	
12.345	0x50	Comments can be replaced any time after the first (or
15.598	0x40	second) column of data. “White” spaces (e.g., “tab” or
11	0x30	“space”) are used to separate the first and second data
8.5698	12	columns as well as the comment field. A “newline”
3.8901	10	character must terminate each line.
1.234e-1	2	
1.44e-2	1	
0	10	



There are no blank lines separating the individual rows.

After the **CAM** file is processed, the number of rows of data are recorded and automatically loaded into parameter PRM:153. Thus, the user does not need to be concerned with changing the spline point number in this parameter when executing **CAM** files in succession with different numbers of spline points.

Finally, assuming no errors in interpreting the **CAM** file, execution of the spline begins.



Before continuing, certain ,“run time” errors (REG:16 for task 1 and REG:17 for task 2) can occur during processing of a **CAM** file. These errors, ranging in numbers from 115 through 322 are explained in Appendix D: Processing Error Codes.



When Task 1 or Task 2 is executing a cam command with a file number other than “0”, user memory is being accessed. If at the same time task 0 (the communications task) is editing , deleting , download , or compiling a file, file shifting in memory may be taking place. The task executing the cam command does not know of this and may scan the wrong data.

The problem of compiling and running a file for another task in this situation can be eliminated if parameter PRM:025 is set to zero (“LST1” file generation is disabled).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

For profiles that require a large amount of spline data points, storage of these points in ASCII form using the **CAM** file may not be practical.

It is in this situation that the second form of the cam command, **CAM(**0**)** should be used in which the dummy file number "0" is specified. This form instructs the interpreter *not* to search for a **CAM** file, but use the data already loaded in the "FV" (and if PRM:162 not equal to zero, the "BV") variable arrays for spline execution.

It is assumed that the current data in the "FV" array (and "BV" array) has been previously loaded by way of the "HOST_100.EXE" DOS utility or similar process.

Since a **CAM** file did not load the data, it is the user 's responsibility to set parameter PRM:153 to the appropriate spline point number.



The CAM() command is computationally time intensive. So, depending on the task that CAM() command is issued, the value of PRM:301 (Task 1) or PRM:302 (Task 2) must be increased to "20" time slots or a "Runtime Error", 121 will be issued.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.10. CBI: Command

Choosing to use the **CBI** converts a BCD (binary coded decimal) number to an integer or float. The controller can not distinguish between BCD and Binary (Hexadecimal) numbers. Therefore, the BCD numbers get treated as if they were binary numbers. All mathematical expressions that contain a BCD number require the use of this conversion. This conversion function is usually necessary when reading the Thumbwheel Option Board (THM option).

SYNTAX: *<var>=CBI(<var or constant>)*

For example:

BV:1=CBI(0x1234) returns an integer 1234 to variable BV:1. The result of this conversion is as follows:

BCD = |0000| |0000| |0001| |0010| |0011| |0100|

BV:1 = |0000| |0000| |0000| |0100| |1101| |0010|

EXAMPLE:

```
BEGIN                   ; Start the program.
PV:1=0x10               ; Set integer, variable 1 equal to 0x10.
BV:2=CBI(0x20)          ; Set integer, variable 2 equal to 0x20.
                        ; Internal bit representation appears as:
                        ; |0000| |0000| |0000| |0000| |0001| |0100|
BV:3=CBI(PV:1)          ; Convert 0x10 to the integer value of port
                        ; variable 1. The result is ten. Then, store this
                        ; result in integer, variable 3.
END                     ; Internal bit representation appears as:
                        ; |0000| |0000| |0000| |0000| |0000| |1010|
                        ; End the program.
```

Related Commands

CIB

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.11. CIB Command

Choosing to use the **CIB** converts an integer to a (BCD) binary coded decimal. The controller cannot distinguish BCD numbers from Binary (Hexadecimal) numbers. Therefore, in a mathematical expression, it is not possible to combine these numbers with any other type of number. Doing so can result in some erroneous result. This conversion function is usually necessary when writing data to the Display Option Board (DISP option).

SYNTAX: *<var>=CIB(<var or constant>)*

For example:

PV:1=CIB(4756) returns a hexadecimal 0x4756 to the external port variable PV:1. The system interprets the result as a binary representation for a hexadecimal number as shown below.

integer = |0000| |0000| |0000| |0100| |1101| |0010|

PV:1 = |0000| |0000| |0001| |0010| |0011| |0100|

EXAMPLE:

```
BEGIN                   ; Start the program.
BV:1=1234               ; Set integer, variable 1 equal to 1234.
PV:1=CIB(BV:1)          ; Set BCD value of port, variable 1 equal to the
                        ; BCD value of integer, variable 1. The result is
                        ; 0x1234.
                        ; Internal bit representation appears as:
                        ; |0000| |0000| |0001| |0010| |0011| |0100|
END                     ; End the program.
```

Related Commands

CBI

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.12. CLN Command

The **CLN** command clears the screen from the present cursor position to the column containing a constant or a variable. It is good practice to set the cursor's initial position before using the CLN command. This guarantees the exact position of the cursor.



The CLN column number must be larger than the current cursor position and not exceed the number twenty.

SYNTAX: *CLN(<constant or var>)*

EXAMPLE:

```
BEGIN                    ; Start the program.
CUR(1,1)                 ; Position the cursor at row 1, column 1.
PM("This is a CLN demo") ; Print the message to the terminal.
CUR(1,11)                ; Position the cursor at row 1, column 11.
CLN(14)                  ; Erase any text in row 1 from column 11 up to
                         ; and including any text in column 15.  This
                         ; prints only the message "This is a     demo".
DW(5)                    ; Set the dwell time for 5 seconds.
END                      ; End of the program.
```

Related Commands

CLS, CUR

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.13. CLS Command

The **CLS** command clears the entire screen. After clearing the screen the cursor normally returns to a position at row one, column one. However, to guarantee the cursor's exact location, use the "CUR" command following use of the CLS command.

The user may wish to display the screen for a specified period of time before using this command. To do this refer to the DW dwell duration command, section 5.7.23.

An important practice to follow is placing the CLS command after the "BEGIN" command in a program. This guarantees a clear screen for the user to work from.



SYNTAX: *CLS*

EXAMPLE:

```
BEGIN                   ; Start the program.
CLS                     ; Clear any previous data from the screen.
CUR(1,1)                ; Position the cursor at row 1, column 1.
PM("This is a CLS ")    ; Print the message to the terminal.
CUR(2,1)                ; Position the cursor at row 2, column 1.
PM("Watch it go!")      ; Print the message to the terminal.
DW(5)                   ; Set the dwell time to 5 seconds.
CLS                     ; Display the above messages for 2 seconds and
                        ; clear the entire screen.
CUR(2,1)                ; Position the cursor again at row 2, column 1.
PM("Did you understand?") ; Print the message to the terminal.
DW(5)                   ; Set the dwell time to 5 seconds.
END                     ; End of the program.
```

Related Commands

CLN, CUR

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.14. COS Command (Cosine)

The **COS** function converts a floating point number to the cosine value of an angle in radians. It is important the user understands not to equate this function to variables of the type integer, long integer, since the result of this conversion is always a number between ±1.0.

SYNTAX: *<var>=COS(<var or constant>)*

EXAMPLE:

```
BEGIN                ; Start the program.
FV:1=COS(.5236)      ; Set floating point, variable 1 equal to the cosine
                     ; of .5236 radians or 30°.  The FV:1 variable
                     ; stores the result of .866025.
FV:2=COS(.785)       ; Set floating point, variable 2 equal to the cosine
                     ; of .7854 radians or 45°.  The FV:2 variable
                     ; stores the result of .707388.
FV:3=COS(1.0472)     ; Set floating point, variable 3 equal to the cosine
                     ; of 1.0472 radians or 60°.  The FV:3 variable
                     ; stores the result of .499998.
FV:3=0               ; Set floating point, variable 3 equal to 0 radians
                     ; or 0°.
FV:4=COS(FV:3)       ; Set floating point, variable 4 equal to the cosine
                     ; of floating point, variable 3.  The FV:4 variable
                     ; stores the result of 1.000000.
END                  ; End the program.
```

Related Commands

SIN, TAN

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.15. CUR Command

The **CUR** command moves the cursor to the row and column containing a constant or variable. This command is normally used in conjunction with the "PM" (Print Message) and "GM" (Get Message) commands. This command allows the user to position the text on the screen. Otherwise, printing multiple messages to the screen simultaneously starts each message at the end of the last message recognized.

SYNTAX: *CUR(<constant>,<constant>)*

or

CUR(<BV:number>,<BV:number>)

EXAMPLE:

```
BEGIN                   ; Start the program.
CLS                     ; Clear any previous data from the screen.
CUR(1,1)                ; Position the cursor at row 1, column 1.
PM("First Message")     ; Print the first message to the terminal screen.
CUR(2,3)                ; Position the cursor at row 2, column 3.
PM("Second Message")    ; Print the second message to the terminal
                        ; screen directly under the first message but
                        ; indented 3 character spaces.
DW(5)                   ; Set the dwell time to 5 seconds.
END                     ; End of the program.
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.16. D Command (Distance in User Units to move/position)

The **D** command represents the distance or position that the motor must move after executing a motion command. If the previous mode set was the "INCR" (incremental mode), the D command is the distance to move in units. For the "ABSL" mode, the D command represents a position that is relative to the present position. The constant or variable used with this command may be either a positive or a negative number. A negative number moves in a counterclockwise (CCW) direction where a positive number moves in the clockwise (CW) direction. (Assuming the incremental mode.)

The value defined with this command remains in effect until entry of a new distance command. The associated default parameter is PRM:105, and does not get overwritten when using this command.



It is possible to change the parameters default from within the program, but this change is global and affects all programs using the Distance command.

The distance scale factor is programmed in user units through PRM:100. The scale factor default setting for this parameter is "1.0".

By specifying multiple "D( )" commands (with accompanying "V( )", "T( )", and/or "A()" commands if desired), into a series, and then following these commands with a "GO", velocity profiling for the specified motion can be obtained.

SYNTAX: *D(<var or constant>)*

The example shown below produces the profile shown in Figure 5-3.

```
D(100000)       Segment 1
V(5000)         Segment 1
T(1)            Segment 1
D(200000)       Segment 2
V(10000)        Segment 2
D(100000)       Segment 3
V(2000)         Segment 3
GO
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

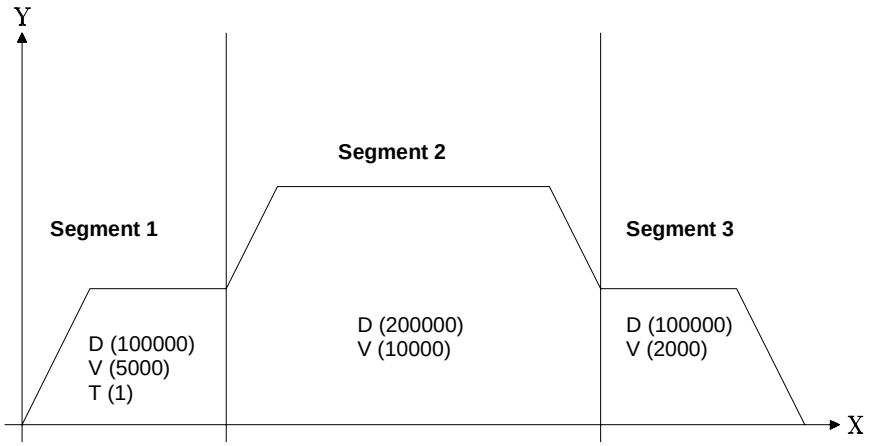


Figure 5-3. Velocity Profiling in 3 Segments

If a "V( )", "T( ), or an "A( )" command is not specified in subsequent segments, the last specified parameter is used for the given segment. This carry over is true even when a new motion command series with the terminating "GO" is specified in the program.

If a motion command series is encountered without any previous reference to a motion parameter, the respective parameter defaults are used (PRM:104, PRM:105, PRM:106, and PRM:107).

Input and output "on the fly" can also be provided when executing motion profiles.

For example, the same motion command series shown above can provide input and/or output at the segment transition points, as shown below.

```
D(100000)          Segment 1
V(5000)            Segment 1
T(1)               Segment 1
BV:1=REG:1         Segment 2
D(200000)          Segment 2
V(10000)           Segment 2
REG:2=0x33         Segment 3
FV:1=REG:202       Segment 3
D(100000)          Segment 3
V(2000)            Segment 3
GO
```

The input statement "BV:1=REG:1" executes between the ending of segment 1 and the beginning of segment 2.

The output statement "REG:2=0x33" and input statement "FV:1=REG:202" execute between the ending of segment 2 and the beginning of segment 3.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

As indicated, more than one input/output command can be inserted between segments. As a rule, limit the amount of inserted statements to two. If more statements are needed, Task 1 or Task 2 background times may have to be increased (see PRM:301 and PRM:302) or the motion profile will exhibit a bump at the segment transition points.

The type of statement that can be inserted between segments is listed as follows.

<var>=<var or constant> statement

RI command

EI command

DI command

EXAMPLE:

```
BEGIN          ;Start the program.
HM             ;Initialize the home position to zero.
ABSL           ;Initialize the absolute (position) mode.
V(20000)       ;Set the velocity to 20000 units per second.
D(100000)      ;Move to position 100000 in the CW direction.
GO             ;Start the move from the home position.
INCR           ;Initialize the incremental (distance) mode.
V(30000)       ;Set the velocity to 30000 units per second.
D(250000)      ;Move to position 250000 in the CW direction.
GO             ;Start the move from the previous position
               ;of 100000.  Move an additional 250,000 steps
               ;and stop at 350,000.
END            ;End the program.
```

Related Commands

A, GO, T, V

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.17. DAC Command (Digital to Analog Conversion)

The **DAC** command provides a scaled analog output voltage of +10 volts to -10 volts to the analog output port. This DAC value uses parameter PRM:142 as a 100% reference (10 volt reference). The negative DAC values provide negative output voltages. The following procedure explains how you can calculate the voltage output.

Vout = (DAC value / PRM:142 value) * 10 volts

If parameter PRM:142 is equal to 1 and the DAC value is at .2, the formula would appear as follows:

Vout = (.2/1) * 10 volts = 2 volts

The result of this would be a 2 volt output.

SYNTAX: *DAC(<var or constant>)*

EXAMPLE:

```
BEGIN                ; Start the program.
FV:1=.2              ; Set floating point variable #1 equal to .2.
DAC(FV:1)            ; Scale and output floating point variable #1 as
                     ; analog voltage based on parameter #142 (100%)
END                  ; End the program.
```

Related Commands

ADC

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.18. DD Command (Direct Drive)

The **DD** command allows the user to enter a trajectory command directly to the drive. The value loaded into the DD( ) command is linear interpolated and executed over a 6.4 msec time period. Scaling for the DD command is selected by PRM:112 (default is 1.0). The user may generate trajectory commands using the information available (e.g., encoder input, internal calculations, etc.) to the controller. Typical applications for this command include: joystick, slave, and freerun. The controller offers a first order filter for use with this command. To use this filter the user must enable this filter with parameter PRM:102 and set the filter coefficient with parameter PRM:101.

The controller synchronizes this command to the task internally. This insures an accurate transfer of the DD data every 6.4 msec. If using the DD command inside loops, the user may adjust the task background time (PRM:301 and PRM:302) to optimize the performance.

The user may choose to output the DD commands at a rate of once every 6.4 millisec. Parameter PRM:112 sets the User Units for the DD command.

SYNTAX: *DD(<var or constant>)*

EXAMPLE:

```
BEGIN                   ; Start the program.
FV:1=100                ; Initialize the variable to 100.
WHL(REG:1 EQ 0Xff)      ; While register 1 is equal to 0Xff loop around.
                        ; Otherwise, break the loop.
  DD(FV:1)              ; Output the trajectory command directly to the
                        ; drive.
ENDWHL                  ; End the while loop.
END                     ; End the program.
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.19. DEC Command (Decrement)

The **DEC** or decrement command subtracts one from a specified variable. This command is very useful when the user wishes to count down a variable used for loop counting. The number of times that the loop executes is contingent upon the initial count.

SYNTAX: *DEC(<var>)*

EXAMPLE:

```
BEGIN                ; Start the program.
BV:1=99              ; Set integer, variable 1 equal to ninety-nine.
WHL(BV:1 NE 0)       ; While integer, variable 1 is not equal to 0
                     ; continue to repeat the loop.  Otherwise stop.
  V(10000)           ; Set the velocity equal to ten thousand.
  D(50000)           ; Set the distance equal to fifty thousand.
  GO                 ; Start the move.
  DEC(BV:1)          ; Decrement integer, variable 1 by one only until
                     ; integer, variable 1 is not equal to zero.
ENDWHL               ; End the while loop when integer, variable 1 is
                     ; equal to one.
END                  ; End of the program.
```

Related Commands

INC

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.20. DEF File

A **DEF** file is a definition file that is external to the actual program. This file allows the user to assign names to variables or commands used in the program. A DEF file also permits the user to change the type of variable without having to search through an entire program.



This command is case sensitive, use capital letters when entering this command.

The format for creating a definition file is almost the same as that of a program. The name of the file can only be a constant from 1 to 9999 and defined as DEF1 - DEF9999. Enter the data just as you would a program but omit the use of the BEGIN and END statements.



Comment fields are not allowed in DEF files.

SYNTAX: *DEF<number>*

The following is an example of a definition file and the file name is “DEF99”.

```
TRAVEL=FV:1
APART=FV:2
OUTPUT=FV:3
```

To open a DEF file in a program, insert it before the BEGIN statement in a program. The DEF file and program file are combined together during the run operation. See example on following page.

Once compiled, the definition is replaced by the controller statement. To verify replacement, view the LST1 list file.



Defined names should not be substrings of other names
(e.g., X=BV:1 or X1=BV:10).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

EXAMPLE:

```
DEF99                       ;Open Definition File DEF99.
BEGIN                       ;Start the program.
CLS                         ;Clear the terminal screen.
CUR(1,1)                    ;Position the cursor at row 1, column 1.
PM("Enter length ")         ;Print the message to the terminal screen.
TRAVEL=GM(20)               ;Get the input from the terminal and place the
                            ;result in the floating point variable TRAVEL.
CUR(2,1)                    ;Position the cursor at row 2, column 1.
PM("How far apart? ")       ;Print the message to the terminal screen.
APART=GM(20)                ;Get the input from the terminal and place the
                            ;result in the floating point variable APART.
CUR(3,1)                    ;Position the cursor at row 3, column 1.
OUTPUT=TRAVEL/APAR          ;Set output equal to the quotient of floating
T                           ;point variable TRAVEL divided by floating
                            ;point variable APART.
                            ;Print the message to the terminal screen.
PM("Total outputs ")        ;Print the result of floating point variable
PM(OUTPUT,"%.2f")           ;OUTPUT, but suppress the zero's to two places
                            ;following the decimal.
                            ;End the program.
END
```

Related Commands

PGM, MAC

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.21. DI Command (Disable Interrupt)

The **DI** command disables the external interrupt that was previously enabled. The system ignores all interrupts that occur after this command. The controller enables interrupts using the EI command and resets them with the RI command.

The associated parameters for defining the interrupt are PRM:132 through PRM:135. These parameters select the source of interrupt as well as configure it.

SYNTAX: *DI*

EXAMPLE:

```
BEGIN          ; Start the program.
EI             ; Enable the external interrupt.
V(1000)        ; Set the velocity equal to 1000.
D(20000)       ; Set the distance equal to 20000.
GO             ; Start the move.
DI             ; Disable the external interrupt.
END            ; End the program.
```

Related Commands

EI, RI

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.22. / Divide Command

The / (divide) function permits the division of integers, long integers, floating point, and hexadecimal numbers. Expressions may also contain variables and therefore it is possible to divide both numbers and variables in any combination.

The controller truncates the decimal portion of an expression that contains a fraction if it is being equated to an integer variable (e.g., BV:1 or LV:1=3/1.25).

SYNTAX: *<var or constant>/<var or constant>*

EXAMPLES:

```
BEGIN                   ; Start the program.
BV:1=10                 ; Set integer, variable 1 equal to ten.
BV:2=5                  ; Set integer, variable 2 equal to five.
BV:3=BV:1/BV:2          ; Set integer, variable 3 equal to the quotient of
                        ; integer, variable 1 and integer, variable 2. The
                        ; result is two.
BV:4=BV:1/10            ; Set integer, variable 4 equal to the quotient of
                        ; integer, variable 1 and ten. The result is one.
END                     ; End the program.
```

```
BEGIN                   ; Start the program.
BV:1=5                  ; Set integer, variable 1 equal to five.
LV:2=2.5                ; Set long integer, variable 2 equal to two.
FV:3=BV:1/LV:2          ; Set floating point, variable 3 equal to the
                        ; quotient of integer, variable 1 and long integer,
                        ; variable 2. The result is two.
END                     ; End the program.
```

Related Commands

+, -, *, =

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.23. DW Command (Dwell Time)

Using the **DW** command acts as a programmable wait with a resolution of 6.4 msec. The value specified by the variable or constant must be a positive number only. Data entered to the command is in seconds.

SYNTAX: *DW(<var or constant>)*

EXAMPLE:

```
BEGIN           ; Start the program.
HM              ; Initialize the home position to zero.
INCR            ; Initialize the incremental (distance) mode.
V(20000)        ; Set the velocity to 20000 units per second.
D(100000)       ; Move a distance of 100000 steps in the CW
                ; direction.
GO              ; Start the move from the home position.
DW(2.0)         ; Set the dwell time to 2 seconds.
V(30000)        ; Set the velocity to 30000 units per second.
D(250000)       ; Move a distance of 250000 in the CW direction.
GO              ; Start the move from 100000 and move 250000
                ; steps until a distance of 350000 is reached.
END             ; End the program.
```

5.7.24. EI Command (Enable Interrupt)

The **EI** command permits the user to enable the external interrupt. This command will also abort a latched interrupt by automatically issuing an "RI" command. The conditions specified by the interrupt parameters determine how to enable the interrupt. The user may configure interrupts to initiate a Task 2 program or latch a program (a user selected position). The DI command disables the interrupt and the RI command resets a latch interrupt.

The response to an interrupt depends on the selectable interrupt parameters PRM:132 through PRM:135.

SYNTAX: *EI*

EXAMPLE:

```
BEGIN           ; Start the program.
EI              ; Enable the external interrupt.
V(1000)         ; Set the velocity equal to 1000.
D(20000)        ; Set the distance equal to 20000.
GO              ; Start the move.
DI              ; Disable the external interrupt.
END             ; End the program.
```

Related Commands

DI, RI

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.25. ELSE Command

The **ELSE** command provides an alternative to executing an IF statement should the IF statement be false. The first expression is the one following the IF statement and the second expression follows the ELSE statement. When the IF condition tested is true, the first expression executes, otherwise the controller executes the second statement that follows the ELSE command.

SYNTAX: *ELSE*

If the controller encounters an ELSE command in a program without the accompanying IF command, an in-process compiler error #210 results.



EXAMPLE:

```
BEGIN                   ; Start the program.
CLS                     ; Clear the terminal screen.
CUR(1,1)                ; Position the cursor at row 1, column 1.
FV:1=25                 ; Set floating point, variable 1 to twenty-five.
FV:2=25                 ; Set floating point, variable 2 to twenty-five.
IF(FV:1 EQ FV:2)        ; Evaluate the IF condition, if it is true carry out
  FV:1=FV:1+100         ; the statement.  Otherwise, perform the ELSE
                        ; function.  The result is true and returns 125.
ELSE                    ; Perform the ELSE statement if the IF condition
  FV:1=0                ; was not met.
ENDIF                   ; End the IF statement.
PM(FV:1,"%.2f")         ; Print floating point, variable 1 to the terminal
                        ; screen, but only include two decimal places to
                        ; the right of the decimal.
END                     ; End the program.
```

Related Commands

IF, ELSEIF, ENDIF

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.26. ELSEIF Command

The **ELSEIF** command is a combination of both the ELSE and the IF commands. This command follows an IF command to make another test. This manual refers to these tests as conditions. If the controller encounters an ELSEIF statement during program execution, the system stops and evaluates that condition. If the condition tested is true, the expression following that condition executes. Otherwise, the program resumes at the next ENDIF, ELSE, or ELSEIF statement.

The operator "OP" used with this statement can be one of the following.

EQ	equal to
GE	greater than or equal to
LE	less than or equal to
GT	greater than
LT	less than
NE	not equal to

SYNTAX: *ELSEIF(<var or constant> op <var or constant>)*

The operator "OP" can also be one of the following.

AND	binary AND, if result is "0", statement is false, otherwise statement is true.
OR	binary OR, if result is "0", statement is false, otherwise statement is true.

EXAMPLE:

```
BEGIN                    ; Start the program.
CUR(1,1)                 ; Position the cursor at row 1, column 1.
BV:1=1                   ; Set integer variable 1 equal to one.
BV:2=2                   ; Set integer variable 2 equal to two.
IF(BV:1 GT BV:2)         ; Evaluate the IF condition, if true carry out the
  BV:1=BV:1-1            ; statement; if false continue on. The result is
                         ; false.
ELSEIF(BV:1 EQ BV:2)     ; Evaluate the ELSEIF condition, if true carry
  BV:1=BV:1+1            ; out the statement; if false continue on. Note
                         ; that neither condition was met and therefore
                         ; no action was taken.
ENDIF                    ; End the IF statement.
PM(BV:1)                 ; Print the value of integer variable 1 to the
                         ; terminal screen.
END                      ; End the program.
```

Related Commands

IF, ELSE, ENDIF

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.27. END Command

The **END** command identifies the end of a program and/or end of compile. This command must appear after the last line of the main body of a program. If subroutines are present in the PGM file, the END statement must be placed after the ENDSUB statement of the last subroutine.

This command is case sensitive, use capital letters when entering this command.

Failure to indicate the end of a program results in a post-compiler error #307.



EXAMPLE:

```
BEGIN                ; Start the program.
V(1000)              ; Set the velocity equal to one thousand.
D(20000)             ; Set the distance equal to twenty thousand.
GO                   ; Start the program.
END                  ; End compiler and program execution.
```

Related commands

BEGIN, EXIT

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.28. ENDIF Command

The **ENDIF** statement must always appear at the end of the IF/ELSE or IF/ELSEIF block. This statement serves as an alternative point at which a program should continue, when the preceding IF or ELSEIF statement is false.



Failure to conclude an IF statement with its corresponding ENDIF statement results in a post-compiler error #302.

SYNTAX: *ENDIF*

EXAMPLE:

```
BEGIN                   ; Start the program.
BV:1=10                 ; Set integer, variable 1 equal to ten.
IF(BV:1=5)              ; If integer, variable 1 is equal to five, execute
                        ; the following statement.  The statement is
                        ; false.
  DW(1.0)               ; Set the dwell time equal to one second if the
                        ; corresponding IF statement is true.
ENDIF                   ; End the IF statement.
END                     ; End the program.
```

Related Commands

IF, ELSE, ELSEIF

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.29. ENDMAC Command

The **ENDMAC** command appears only in a MAC file (Macro file). This command denotes the end of the macro contained in the MAC file. This command can not be used without a corresponding "MAC<macro name>" command.

This command is case sensitive, use capital letters when entering this command.



SYNTAX: *ENDMAC*

EXAMPLE:

```
MAC MOVE          ; The macro program name is MOVE.
T(%0)             ; Time is specified by the first argument in the
                  ; main program.
V(%1)             ; Velocity is specified by the second argument in
                  ; the main program.
D(%2)             ; Distance is specified by the third argument in
                  ; the main program.
GO                ; Start the move.
ENDMAC            ; End the macro called MOVE.

MAC DSPLY         ; The macro program name is DSPLY.
CLS               ; Clear the terminal screen.
CUR(1,1)          ; Position the cursor at row 1, column 1.
PM(%0)            ; Display the message denoted by the first
                  ; argument in the main program.
DW(2)             ; Set the dwell time equal to 2 seconds.
ENDMAC            ; End the macro DSPLY.
```

Related Commands

MAC<macro number>

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.30. ENDSUB Command

The **ENDSUB** command is the last command of a subroutine task. The ENDSUB does not function alone, it also requires a corresponding "SUB" command. After completion of this command, program execution picks up at the command following the "GOSUB" that called the subroutine.



In the event that the user fails to indicate the end of a subroutine using the "ENDSUB" command, the controller returns an in-process compiler error #202.

SYNTAX: *ENDSUB*

EXAMPLE:

```
BEGIN           ; Start the program.
REG:2=0Xff      ; Set the bit output register #2 equal to 0Xff.
D(100)          ; Set the distance equal to one hundred.
V(100)          ; Set the velocity equal to one hundred.
T(.1)           ; Set the time equal to .1 second.
GO              ; Start the move.
GOSUB:10        ; Go to subroutine 10.
EXIT            ; Exit the program and carry out the subroutine.
SUB:10          ; Define the following as subroutine 10.
REG:2=0Xfd      ; Set the output bit register 2 equal to 0Xfd.
DW(1.0)         ; Set the dwell time to one second.
REG:2=0Xff      ; Set the output bit register 2 equal to 0Xff.
ENDSUB          ; End the subroutine.
END             ; End the program.
```

Related Commands

SUB, GOSUB

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.31. ENDWHL Command

The **ENDWHL** command indicates the end of a while loop. For this command to be used in a program, an associated WHL command must also exist. Failure to include both of these commands causes a compiler error to occur. The ENDWHL command reinstates a while loop for as long as the previous while test is true. If the previous while test is false, then the program execution continues with the command following the ENDWHL command.

SYNTAX: *ENDWHL*

EXAMPLE:

```
BEGIN                       ; Start the program.
BV:1=0                      ; Set integer, variable 1 equal to zero.
CLS                         ; Clear the terminal screen.
WHL(BV:1 NE 10)             ; While integer, variable 1 is not equal to ten,
                            ; continue with the loop. Otherwise don't
                            ; execute.
CUR(1,1)                    ; Position the cursor at row 1, column 1.
PM(BV:1, "BV:1 equals %d")  ; Print the message to the terminal screen.
DW(1)                       ; Set the dwell time equal to one second.
INC(BV:1)                   ; Increment the integer, variable 1 by one.
ENDWHL                      ; End the while loop.
END                         ; End the program.
```

Related Commands

WHL

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.32. = Assign a value to

The = (equal) function assigns a value from the right of the "=" to a variable to the left of the "=". The expression to the right of the equal sign can contain integers, long integers, floating point, and hexadecimal numbers. Expressions may also contain variables and therefore it is possible to add, subtract, multiply, and divide both numbers and variables in any combination.

The controller truncates the decimal portion of an expression that contains a fraction if it is being equated to an integer variable (e.g., BV:1 or LV:1=3.25).

SYNTAX: *<var>=<var or constant>*

EXAMPLES:

```
BEGIN           ; Start the program.
BV:1=10         ; Set integer, variable 1 equal to ten.
BV:2=5          ; Set integer, variable 2 equal to five.
BV:3=BV:1       ; Set integer, variable 3 equal to the value of
                ; integer, variable 1.
END             ; End the program.
```

```
BEGIN           ; Start the program.
BV:1=10         ; Set integer, variable 1 equal to ten.
PV:2=5          ; Set port, variable 2 equal to five.
PV:3=BV:1+32    ; Set port, variable 3 equal to the sum of integer,
                ; variable 1 and thirty-two. The result is 0X2A.
END             ; End the program.
```

5.7.33. EXIT Command

The **EXIT** command permits the user to quit a program. However, the "END" command is still a requirement for ending an entire program. It is necessary that this command always be used to end a program that does not end with an "END" statement (e.g., when subroutines are listed at the bottom of the program).



Failure to place an EXIT command before a subroutine results in an in-process compiler error #202. This error may indicate that the user failed to end a subroutine using the "ENDSUB" command.

SYNTAX: *EXIT*

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

EXAMPLE:

```
BEGIN                ; Start the program.
REG:2=0Xff           ; Set the bit output register #2 equal to 0Xff.
D(100)               ; Set the distance equal to one hundred.
V(100)               ; Set the velocity equal to one hundred.
T(.1)                ; Set the time equal to .1 second.
GO                   ; Start the move.
GOSUB:10             ; Go to subroutine ten.
EXIT                 ; Exit the program and carry out the subroutine.
SUB:10               ; Define the following as subroutine 10.
REG:2=0Xfd           ; Set the output bit register 2 equal to 0Xfd.
DW(1.0)              ; Set the dwell time to one second.
REG:2=0Xff           ; Set the output bit register 2 equal to 0Xff.
ENDSUB               ; End the subroutine.
END                  ; End the program.
```

Related Commands

SUB, ENDSUB

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.34. GC Command (Get Character)

The **GC** command takes a character entered by the user and converts that character to its numerical value and stores the result in the specified variable. Unlike string variables, this command permits the user to compare variables and include them in calculations.

SYNTAX: *<var>=GC*

EXAMPLE:

```
BEGIN                   ; Start the program.
CLS                     ; Clear the terminal screen.
CUR(1,1)                ; Position the cursor at row 1, column 1.
PM("Move ? Y/N ")       ; Print the message to the terminal screen.
BV:1=GC                 ; Get a character equivalent to integer variable
                        ; BV:1.
IF(BV:1 EQ 89)          ; Evaluate the IF condition. If true execute the
                        ; move; if false bypass the move.
D(10000)                ; Move a distance of 10000 steps in the positive
                        ; direction.
GO                      ; Start the move.
ENDIF                   ; End the IF statement.
END                     ; End of the program.
```

Related Commands

GM

5.7.35. GEAR Command

The **GEAR** command is designed to accept feedback information from either of the two encoder ports or the optional R/D board. When issuing this command, the U100/U100i controller will be slaved to this feedback. The gearing function operates at the servo update time setting (PRM:304). Thus, the function can be made to operate at update times as low as .1 mSec.

Since the gearing function works at the servo loop level, motion command statements (D(), V(), T ()… GO, CAM(), and/or DD()) can be issued at the program level while the gearing function is active.

The functionality of the **GEAR** command can be altered through parameters PRM:231, PRM:232, and PRM:233. The accel/decel rate of the motor when starting and stopping the gear function is controlled by PRM:104 (default ramp time parameter).

SYNTAX: *GEAR(<var or constant>)*

Where:

Parameter “1” specifies **1** for ON or **0** for OFF

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

EXAMPLE:

```
BEGIN                   ; Start the program.
GEAR (1)                ; Start Gear.
WHL (1 EQ1)             ; Always.
CLS                     ; Clear the terminal screen.
CUR(1,1)                ; Position the cursor at row 1, column 1.
PM("Adjust ? Y/N ")     ; Print the message to the terminal screen.
BV:1=GC                 ; Get a character equivalent to integer variable
                        ; BV:1.
IF(BV:1 EQ 89)          ; Evaluate the IF condition.  If true execute the
                        ; move; if  false bypass the move.
D(10000)                ; Move a distance of 10000 steps in the positive
                        ; direction on top of the gear motion.
GO                      ; Start the move.
ENDIF                   ; End the IF statement.
ENDWHL                  ; End of While loop.
END                     ; End of the program.
```

Related Commands

D(), V(), T ()… GO, CAM(), and/or DD()

5.7.36. GM Command (Get Message)

The **GM** command gets messages or data entered through the terminal and displays them on the screen. Messages start at the present cursor position and end at the column that contains the specified variable or constant. The cursor's present position is either the most recently recognized position or some pre-defined position specified by the "CUR" command. The length of the message depends on the specified constant or variable of the column containing the present cursor position.

The message is displayed on the screen and stored in the variable equated by the user. This variable must match the data the user desires to enter as well as the number of characters.

If the variable equated to a statement is an integer and the user entered data other than numbers, the system prompts the user for re-entry. However, if the user should accidentally enter a return at this prompt the message displays a zero.

SYNTAX: *<var>=GM(<constant or var>)*

or

SV:<number>=GM(<constant or var>)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

EXAMPLE:

```
BEGIN                   ; Start the program.
CUR(1,1)                ; Position the cursor at row 1, column 1.
PM("Enter Length ")     ; Print the message to the terminal screen.
CUR(1,16)               ; Position the cursor at row 1, column 16.
BV:1=GM(17)             ; Get the input from the terminal, display the
                        ; result, and store it in the integer, variable 1.
CUR(3,1)                ; Position the cursor at row 3, column 1.
PM("You entered ")      ; Print the message to the terminal screen.
PM(BV:1)                ; Print the message to the terminal screen on the
                        ; same line as the most recent message.
DW(5)                   ; Set the dwell time to 5 seconds.
END                     ; End of the program.
```

Related Commands

GC, PM

5.7.37. GO Command (Begin Move)

The **GO** command initiates motion. The motion depends on the type of mode (Incremental or Absolute) and the corresponding V, D, T or A commands.

SYNTAX: *GO*

EXAMPLE:

```
BEGIN                   ; Start the program.
HM                      ; Initialize the home position to zero.
INCR                    ; Initialize the incremental (distance) mode.
V(20000)                ; Set the velocity to 20000 units per second.
D(100000)               ; Move a distance of 100000 steps in the CW
                        ; direction.
GO                      ; Start the move from the home position.
END                     ; End the program.
```

Related Commands

A, ABSL, D, GO, INCR, T, V

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.38. GOSUB Command

A **GOSUB** command is nothing more than an instruction to delegate a task. The task to be performed is the subroutine. When the controller encounters a GOSUB command in a program it transfers control to the subroutine. Upon completion of the subroutine task the program returns to the command line following the GOSUB and proceeds as normal. An advantage of using this command is that the user may repeat the same task multiple times in a program without having to duplicate the commands.

SYNTAX: *GOSUB:<number>*

EXAMPLE:

```
BEGIN                ; Start the program.
REG:2=0Xff           ; Set the bit output register #2 equal to 0Xff.
D(100)               ; Set the distance equal to one hundred.
V(100)               ; Set the velocity equal to one hundred.
T(.1)                ; Set the time equal to .1 second.
GO                   ; Start the move.
GOSUB:10             ; Go to subroutine 10.
EXIT                 ; Exit the program and carry out the subroutine.
SUB:10               ; Define the following as subroutine 10.
REG:2=0Xfd           ; Set the output bit register 2 equal to 0Xfd.
DW(1.0)              ; Set the dwell time to one second.
REG:2=0Xff           ; Set the output bit register 2 equal to 0Xff.
ENDSUB               ; End the subroutine.
END                  ; End the program.
```

Related Commands

ENDSUB, SUB

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.39. GOTO Command

The **GOTO** command in a program permits the user to jump to a label, LB:<number>.



"GOTO:" statements cannot be inserted between the IF( )/ELSEIF( ), IF( )/ELSE, or ELSEIF( )/ENDIF blocks unless their relative label ("LB.xx") statements are within that same block.

It is allowable to have "GOTO:xx" statements jump out of an IF( )/ENDIF block.

SYNTAX: *GOTO:<number>*

EXAMPLE:

```
BEGIN                   ; Start the program.
BV:1=0                  ; Set integer, variable 1 equal to zero.
LB:10                   ; Define the following program block as label 10.
BV:1=BV:1+1             ; Set integer, variable 1 equal to the sum of
                        ; integer, variable 1 plus one.
IF(BV:1 NE 10)          ; Determine if integer, variable 1 is not equal to
                        ; ten.  If it is true carry out the following
                        ; statement.  The result is false.
  GOTO:10               ; Jump to the block labeled 10 when integer,
                        ; variable 1 is not equal to ten.
ENDIF                   ; End the IF statement.
END                     ; End the program.
```

Related Commands

LB, GOSUB

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.40. HM Command (Hardware Home)

The **HM** command sends the axis to the Hardware Home position. The axis first moves to a home limit switch, then optionally moves out until it locates the home marker. If the user requires a home position different from the home marker position, a home offset position may be established using parameters PRM:120 and PRM:121. The system utilizes two different home feedrates for homing: Home Velocity on power up (PRM:119) or the Default Velocity (PRM:107). This provides for rapid system homing after the controller has established the initial home position.

The Position Command register will be cleared to zero automatically after execution of a Home command.

SYNTAX: *HM*

EXAMPLE:

```
BEGIN    ; Start the program.
HM       ; Send the axis home.
END      ; End the program.
```

Parameters PRM:116 through PRM:122 allow special setup conditions for the HM command.

The setting of PRM:116 dictates the initial homing direction. If the "+" limit switch is selected for home reference, set PRM:116 to a "1". If the "-" limit switch is selected, set this parameter to a "2". If the home limit switch is selected, the setting of this parameter dictates the side of the switch to be used as the home reference.

The setting of PRM:117 dictates the selected home switch reference. In addition to this selection, the polarity of the selected switch can be set.

The setting of PRM:118 selects the marker (or index pulse) of the selected encoder interface to be used as the actual home position. If an encoder is not being used (e.g., a stepping motor with no encoder), the user can set PRM:118 to a "3" indicating that the home switch reference selected by PRM:117 provides the actual home position.

If an encoder marker is used as the home position, PRM:122 selects the rate at which to search for the home marker.

The value set for parameter PRM:122 is critical for accurate homing to the marker. This value must be set to no more than 1/2 of a user unit per servo update time (PRM:304 selects the servo update time). If set greater than this value, the system may never recognize the marker pulse.

For example, the default servo update time is .8 msec and the user units scale factor is "1" (PRM:100). Thus (1/.8e-3)*(1/2) yields 625 which is the default value for PRM:122.

After powering up the controller and executing the first HM command, any motion command(s) will always be referenced to the initial home position. Thus, if another HM command is issued, the controller has knowledge of the home position. Thus on subsequent HM commands, the selected home reference switch (PRM:117) can be sought at a much greater speed. PRM:107 provides the setting for this speed which is ramped via setting of the default ramp time, parameter PRM:104. If the controller is reset (or turned off), this knowledge of the home position is lost. In this case, the system uses the power up velocity (PRM:119) to find the switch and no ramping is executed.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

There are two other parameters provided for homing. These parameters are PRM:120 (Home Limit Offset) and PRM:121 (Home Ending Offset), both have a default setting of "0".

The distance set in PRM:120 will be automatically executed after the activation of the home reference switch and before searching for the home marker (if one exists). After executing this distance, the system searches for the home marker. This feature allows marker pulses to be skipped before searching the next marker pulse.

The distance set in PRM:121 executes automatically after finding the home marker (if one exists). This feature provides home offset control. After executing this distance, the system sets the Position Command register to a "0".

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.41. IF Command

The **IF** command permits the user to test for the condition defined in the parentheses. The statement following the IF command line executes, if the statement is true. If the statement is false, the controller proceeds to the next ELSE, ELSEIF, or ENDIF statement in the program.

The operator "OP" used with this statement can be one of the following.

EQ	equal to
GE	greater than or equal to
LE	less than or equal to
GT	greater than
LT	less than
NE	not equal to

The operator "OP" can also be one of the following.

AND	binary AND, if result is "0", statement is false, otherwise statement is true.
OR	binary OR, if result is "0", statement is false, otherwise statement is true.

SYNTAX: *IF(<var or constant> op <var or constant>)*

EXAMPLE:

```
BEGIN                   ; Start the program.
BV:1=0                  ; Set integer, variable 1 equal to zero.
BV:2=100                ; Set integer, variable 2 equal to one hundred.
IF(BV:2 NE 101)         ; If integer, variable 2 is not equal to 101 then
                        ; move to the next statement.  Otherwise end the
                        ; IF statement.  The result is true.
  BV:1=1                ; Store a one in integer, variable 1 if the IF
                        ; statement is true.
ENDIF                   ; End the IF statement.
END                     ; End the program.
```

Related Commands

ELSE, ELSEIF, ENDIF

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.42. INC Command (Increment)

The **INC** command adds one to a specified variable. This command is very useful for counting the number of times something has occurred.

SYNTAX: *INC(<var>)*

EXAMPLE:

```
BEGIN                   ; Start the program.
BV:1=0                  ; Set integer, variable 1 equal to zero.
WHL(REG:1 EQ 0Xff)      ; While register 1 is equal to 0Xff repeat the
                        ; following move continuously.  Otherwise stop.
  V(100)                ; Set the velocity equal to one hundred.
  D(1)                  ; Set the distance equal to one.
  GO                    ; Start the move.
  INC(BV:1)             ; Increment integer, variable 1 by one only until
                        ; register 1 is equal to 0Xff.
ENDWHL                  ; End the while loop when register 1 is 0Xff.
END                     ; End of the program.
```

Related Commands

DEC

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.43. INCR Command (Incremental Position Mode)

The **INCR** mode command causes all moves following this command to be interpreted as distances. These moves reference the last established distance and add to it the new commanded distance. The controller calculates the move by adding or subtracting the commanded distance to the present position. Based on the distances value, the move follows in the negative CCW or positive CW direction. An alternative to this command would be the "ABSL" absolute positioning mode.

SYNTAX: *INCR*

It is not possible to change the incremental mode while executing motion.



EXAMPLE:

```
BEGIN          ; Start the program.
HM             ; Initialize the home position to zero.
INCR           ; Initialize the incremental (distance) mode.
V(20000)       ; Set the velocity equal to 20000 units per
               ; second.
D(100000)      ; Move a distance of 100000 units in the positive
               ; CW direction.
GO             ; Start the move from the home position.
D(-75000)      ; Move a distance of 75000 units in the negative
               ; CCW direction.
GO             ; Start the move from the previously established
               ; position of 100000.  The resulting position is
               ; 25000.
END            ; End the program.
```

Related Commands

ABSL

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.44. LB Command (Label)

The **LB** command permits the user to label a position in a program for later recall. The GOTO command is the command used to direct the program flow to a label (see the GOTO command, section 5.7.39). When the controller encounters a GOTO command in a program, it looks at the label attached to the command, jumps to that program block and continues execution. The constant used to define the labels used in a program may be any number.

SYNTAX: *LB:<number>*

EXAMPLE:

```
BEGIN                ; Start the program.
BV:1=0               ; Set integer, variable 1 equal to zero.
LB:10                ; Define the following program block as label 10.
BV:1=BV:1+1          ; Set integer, variable 1 equal to the sum of
                     ; integer, variable 1 plus one.
IF(BV:1 EQ 10)       ; Determine if integer, variable 1 is not equal to
                     ; ten.  If it is true carry out the following
                     ; statement.  The result is false.
  GOTO:10            ; Jump to the block labeled 10 when integer,
                     ; variable 1 is not equal to ten.
ENDIF                ; End the IF statement.
END                  ; End the program.
```

Related Commands

GOTO

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.45. LOCK Command (Lock Task Execution)

The **LOCK** command allows a program running in Task 1 or Task 2 to override the multitasking kernel of the UNIDEX 100/100i and execute 100% of processor speed. Meaning, all other tasks are suspended while the operating task is locked.

If this command is issued in Task 1, Task 0 (communications task) and Task 2 (program task) are suspended in their current state. Likewise, if issued in Task 2, Task 1 and Task 0 are suspended.

Care must be taken when using this command if a suspended program task (Task 1 or 2) is executing a motion command. Motion will stop abruptly. When the lock is removed, motion will resume in an abrupt manner.

In addition, if Task 0 is used for communications (via RS-232 or IEEE-488) when issuing a **LOCK** command, communications are suspended. However, the communications interrupt is still valid, allowing up to 256 characters to be received in the communication buffer. The processing of these characters will not continue until the lock is removed.

Communications through **GM** and **PM** commands are still valid if contained within the **LOCK** commands.



High level communication commands, such as SRQs (Service Requests), as well as the TFX and HOST reset commands are always active regardless of the active or inactive state of the lock.

SYNTAX: *LOCK(<var or constant>)*

Where:

var or *constant* must resolve one of two possible values.

EXAMPLE:

```
BEGIN                ; Start the program.
BV:1=1               ; Set integer, variable 1 equal to one.
.                    ;
.                    ; additional program statements
                     ;
LOCK(1)              ; this task (1 or 2) now has 100% processor time allocated
.                    ; to it
.                    ; all statements between LOCK(1) and LOCK(0) execute
.                    ; with no chance of being interrupted by the multitasking
                     ; kernel
LOCK(0)              ; normal multitasking operation resumes
END                  ; End the program.
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.46. MAC Command

The **MAC** command denotes a macro file. The MAC file contains one or more modules called macros that consist of program commands. To call a MAC file from a program, the MAC file must first be declared. To do this, specify the MAC command followed by the MAC file number. Once defined, specify the name of the macro contained in the macro file followed by the macro argument(s) enclosed in parentheses at the desired position in the program body. Once compiled, the macro program replaces the statement(s) that called the macro(s).



This command is case sensitive, use capital letters when entering this command.

Macro files also use a MAC command to define the individual macro module.

SYNTAX: *MAC<number>*

EXAMPLE:

```
TITLE**This is PGM 25**      ; Attach the program description to the program
                             ; number.
MAC1                         ; Define MAC file #1 so that the program may
                             ; use the macro programs contained therein.
BEGIN                        ; Start the program.
DSPLY("RUN")                 ; Call the display macro from the macro file.
                             ; The argument contains the text string.
MOVE(1,50000,400000)         ; Call the motion macro from the macro file.
                             ; Where argument 1 is time, 2 is velocity, and 3
                             ; is distance.
DW(4)                        ; Set the dwell time equal to four seconds.
END                          ; End the program.
```

The following is an example of the macro file used with the above “PGM” program example. This macro file contains 2 macros called “MOVE” and “DSPLY”. The file name is “MAC1”.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

EXAMPLE:

```
MAC MOVE          ; The macro program name is MOVE.
T(%0)             ; Time is specified by the first argument in the
                  ; main program.
V(%1)             ; Velocity is specified by the second argument in
                  ; the main program.
D(%2)             ; Distance is specified by the third argument in
                  ; the main program.
GO                ; Start the move.
ENDMAC            ; End the definition of the macro called MOVE.

MAC DSPLY         ; The macro program name is DSPLY.
CLS               ; Clear the terminal screen.
CUR(1,1)          ; Position the cursor at row 1, column 1.
PM(%0)            ; Display the message denoted by the first
                  ; argument in the main program.
DW(2)             ; Set the dwell time equal to 2 seconds.
ENDMAC            ; End the definition of the macro DSPLY.
```

Comments are not allowed in MAC files. The example above is for readability of the syntax only.



Related Commands

ENDMAC

5.7.46.1. MAC Definition

This MAC command appears inside the MAC file and defines the macro program. Use this command at the start of each macro program in the form of a MAC followed by the user defined macro name. The user defined macro name is the name given to the program command in the PGM file. This name identifies what macro to insert at that point. Following the macro name the PGM file requires that the user specify any argument, in parenthesis, for the macro to use. Refer to first example.

SYNTAX: *MAC<space><macro name>*

Related Commands

ENDMAC

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.47. MDX Command Modulo Index

The MDX command is a modulo index command used to obtain an offset pointer value. A typical use of this command allows the user to retrieve data from an array. This command works in conjunction with a lower and upper modulo index limit. Parameter PRM:110 controls the lower modulo index limit while parameter PRM:111 controls the upper modulo index limit. The lower modulo index limit is the minimum value the MDX command obtains. This limit is normally equal to the lowest variable number in the array. The upper modulo index limit is the maximum value that the MDX command obtains. Normally, this limit is equal to the largest variable number in the array.

Upon executing the MDX command an index value within the range of the lower and upper modulo indexes is derived. If the chosen variable exceeds the upper or lower limit, the residual is added to the lower bound or subtracted from the upper bounds depending on which side of the bounds the number exceeded. This provides the user with a circular pointer with several applications. After deriving this pointer, the controller places the pointer into the user selected variable. This pointer may then do indexing with variables.

SYNTAX: *<var>=MDX(<var>)*

EXAMPLE:

```
BEGIN                   ; Start the program.
BV:1=0                  ; Initialize the loop count variable to zero.
BV:2=100                ; Initialize the array pointer to 100.
WHL(BV:1 LT 500)        ; Loop until the array is loaded.
  FV:3=BV:1 * .01256637 ; Calculate the array angle in radians.
  FV:4=SIN(FV:3)        ; Get the sine value of an angle.
  FV:BV:2=FV:4&2000     ; Scale for maximum motion.
  INC(BV:1)             ; Add 1 more to the count.
  INC(BV:2)             ; The next array location to load.
ENDWHL                  ; End of the WHILE loop.
ABSL                    ; Initiate Absolute mode.
BV:22=0                 ; Initialize the array  pointer.
BV:21=1                 ; Array increment value.
LB:1                    ; Start of trajectory generator loop.
BV:22=BV:22+BV:21       ; Get the array offset value.
BV:1=MDX(BV:22)         ; Get the array location.
DD(FV:BV:1)             ; Output the trajectory from the array.
GOTO:1                  ; Repeat the trajectory loop.
END                     ; End the program.
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.48. * Multiply

The * (multiplication) function permits the multiplication of integers, long integers, floating point, and hexadecimal numbers. Expressions may also contain variables and therefore it is possible to multiply both numbers and variables in any combination.

The controller truncates the decimal portion of an expression that contains a fraction if it is being equated to an integer variable (e.g., BV:1 or LV:1=5.25*1).

SYNTAX: *<var or constant>*<var or constant>*

EXAMPLE:

```
BEGIN                   ; Start the program.
BV:1=5                  ; Set integer, variable 1 equal to five.
BV:2=10                 ; Set integer, variable 2 equal to ten.
BV:3=BV:1*BV:2          ; Set integer, variable 3 equal to the product of
                        ; integer, variable 1 and integer, variable 2. The
                        ; result is fifty.
BV:4=BV:1*2.5           ; Set integer, variable 4 equal to the product of
                        ; integer, variable 1 and two and a half. The
                        ; result is stored in integer, variable 4 and is
                        ; equal to twelve.
FV:1=BV:1*2.5           ; Set floating point, variable 1 equal to the
                        ; product of integer, variable 1 and two and a
                        ; half. The result is stored in floating point,
                        ; variable 1 and is equal to ten and a half.
END                     ; End the program.
```

Related Commands

+, -, /, =

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.49. OR Command (Logical OR)

The **OR** function permits the user to OR two binary numbers.

The expression may be a number, a variable, or a combination of both. See the examples below:

SYNTAX: *(<var or constant>OR<var or constant>)*

EXAMPLES:

```
BEGIN                   ; Start the program.
BV:1=1                  ; Set integer, variable 1 equal to one.
BV:2=2                  ; Set integer, variable 2 equal to two.
BV:3=BV:1 OR BV:2       ; Set integer, variable 3 equal to the value of
                        ; integer variable 1 OR integer, variable 2. The
                        ; result is three.
                        ; For example: binary 1 OR binary 2 = binary 3
END                     ; End of the program.
```

Related Commands

AND, XOR

5.7.50. PC Command (Print Character)

The **PC** command works in conduction with the **PM** command allowing non-printable characters to be displayed on the terminal screen or HT (Handheld Terminal).

SYNTAX: *PC(<var or constant>)*

Where:

Parameter “1” specifies the number 0 through 255 (0 through 0xff) representing the ASCII character to be sent.

EXAMPLES:

```
BEGIN                   ; Start the program.
CLS                     ; Clear any previous data from the screen.
CUR(1,1)                ; Position the cursor at row 1, column 1.
PC(0xA)                 ; Send "Carriage Return" character.
BV:1=50                 ; Set integer, variable 1 equal to fifty.
DW(5)                   ; Set the dwell time to five seconds.
END                     ; End of the program.
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.51. PM Command (Print Message)

The **PM** command outputs variables or strings (quotes) to the display. When displaying strings the system requires a quotation mark before and after the message. However, to display variables, place the variable name inside the parenthesis.

The maximum number of characters printed on one line can not exceed twenty.



SYNTAX: *PM(<var or string>)*

EXAMPLES:

```
BEGIN                   ; Start the program.
CLS                     ; Clear any previous data from the screen.
CUR(1,1)                ; Position the cursor at row 1, column 1.
PM("This is a PM ")     ; Print the message to the terminal screen.
BV:1=50                 ; Set integer, variable 1 equal to fifty.
PM(BV:1)                ; Print the value of integer, variable 1 beside the
                        ; most recently recognized message on the
                        ; terminal screen.
DW(5)                   ; Set the dwell time to five seconds.
END                     ; End of the program.
```

```
BEGIN                   ; Start the program.
CLS                     ; Clear any previous data from the screen.
CUR(1,1)                ; Position the cursor at row 1, column 1.
PM("HI")                ; Print the message to the terminal screen.
CUR(1,4)                ; Position the cursor at row 1, column 4.
PM("THERE!")            ; Print the message two spaces after HI on the
                        ; same line of the terminal screen.
DW(5)                   ; Set the dwell time to 5 seconds.
END                     ; End of the program.
```

Related Commands

PM(<var or quote>, <quote>)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.51.1. PM Command (Print Formatted Messages)

The **PM** command provides the user with the ability to insert variables or strings inside messages using only one command. The user may format variables so that decimal points (in the case of the FV: variable type) or "right justification" (in the case of all variable types) can be controlled.

If only simple variable output to the display is required, the "PM(<var or string>)" command form can be used.

SYNTAX: *PM(<var or string>, <string>)*

The format identifier for each of the controller numeric variable types is as follows:

FV:	%f	(fixed point format)
	%e	(float point format)
BV: and PV:	%d	(24 bit integer format)
	%c	(ASCII character format)
LV:	%ld	(48 bit integer format)

For the controller string variable:

SV:	%s	(string output)

Along with the format identifier, the fields of each of the numeric variable types can be controlled. For the BV:, PV:, and LV: variable types, right justification of the data output can be controlled by specifying a <number> in the format identifier. This number signifies the number of digit places to the right of the present cursor position to display the integer (or long integer) data.

%10d indicates that ten digit places to the right of the current cursor position should be maintained for displaying the integer data (24 bit in this case).



Specifying "fields" for the floating point variable FV: is also possible. With this variable an additional field is provided for specifying the "number of digits" to the right of the decimal point.

%10.3f indicates that ten digit places to the right of the current cursor position should be maintained for displaying the "fixed decimal point" data. Out of the ten digit places, three are allocated for digits to the "right" of the decimal point.

%10.3e indicates that ten digit places to the right of the current cursor position should be maintained for displaying the "floating decimal point with exponential notation" data. Three digit positions out of the ten are allocated for digits to the "right" of the decimal point.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Specifying "fields" for the numerical data types provides the additional benefit of automatically "clearing" the digit positions specified by the primary field specifier (e.g., in the above examples, the number "10"). This is useful if the program is constantly updating a given variable (termed, "scanning" the variable).



When scanning the variable, it is usually desirable to inhibit the viewing of the display cursor while the scanning loop is in progress. The cursor can be turned off and on through parameter PRM:021.

EXAMPLE:

```
BEGIN                       ; Start the program.
CLS                         ; Clear any previous data from the screen.
CUR(1,1)                    ; Position the cursor at row 1, column 1.
BV:1=5                      ; Set integer, variable 1 equal to five.
SV:1="string"               ; Set string, variable 1 equal to string.
PM(BV:1,"Five equals %d")   ; Print the message "Five equals 5" to the
                            ; terminal screen.  Note that the %d was
                            ; replaced with the value of integer, variable 1.
CUR(2,1)                    ; Position the cursor at row 2, column 1.
PM(SV:1,"Print %s")         ; Print the message "Print string" to the
                            ; terminal screen.  Note that again the %s was
                            ; replaced with the value of string, variable 1.
DW(5)                       ; Set the dwell time to 5 seconds.
END                         ; End of the program.
```

Related Commands

PM(<var or string>)

5.7.52. PVI Command

The **PVI** command allows the retrieval of data stored in one of the four MEM board image sectors of the flash ROM to be accessed by a user program. The usage of this command is similar to that of the **PV** command, accessing memory directly from the MEM board.

SYNTAX: *<var> = PVI(<var or constant>, <BV: or constant>)*

Where:

Parameter “1” specifies the port variable number 0x200 through 0xdfff.

Parameter “2” specifies the image (1 through 4) in flash ROM for which the copy of the variable resides.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.53. RI Command (Reset Interrupt Latch)

The **RI** command resets the controller interrupt latch. Interrupts may be setup for a latched or unlatched mode. Parameter PRM:134 permits the user to select between the latched and the unlatched mode.

SYNTAX: *RI*



Latched interrupts are not self clearing and require a RI command to clear them.

EXAMPLE:

```
BEGIN                ; Start the program.
EI                   ; Enable the interrupt.
BV:1=0               ; Set integer, variable 1 equal to zero.
WHL(BV:1 NE 10)      ; While integer, variable 1 is not equal to ten
                     ; perform the loop.  Otherwise stop.
  RI                 ; Reset the interrupt.
  V(1000)            ; Set the velocity equal to one thousand.
  D(5000)            ; Set the distance equal to five thousand.
  T(.1)              ; Set the time equal to .1 second.
  GO                 ; Start the move.
ENDWHL               ; End the while loop.
END                  ; End the program.
```

Related Commands

DI, EI

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.54. RUN Command (Run Pre-compiled PGM Program)

The **RUN(<file number>)** allows a program running in Task 1 to automatically load and run a specified library program (PGM) in Task 2.

The program (PGM library file) must have been previously compiled and loaded to the external memory board using the LIBRARY utility's "INSERT" function.

To use this command, the external memory board (MEM option) must be installed.

For more information on this command refer to the *UNIDEX 100 Memory Expansion Board Option Manual P/N EDU 136l.*

SYNTAX: *RUN(<var or constant>)*

EXAMPLE:

For a programming example on the RUN command refer to the *UNIDEX 100 Memory Expansion Board Option Manual P/N EDU 136.*

5.7.54.1. RUN Command - Extended format

The **RUN** command has been extended to allow an optional second parameter to be specified when running library programs from flash ROM directly. Library programs must first be developed using the MEM board. Afterwards, the contents of the MEM board can be transferred to one of four storage sectors on the flash ROM either by using the ARCHIVE menu screen or by using the HOST_100.EXE utility to directly upload the data to flash ROM.

SYNTAX: *RUN(<var or constant>, <BV: or constant>)*

Where:

Parameter "1" specifies the program number (1 through 9999).

Parameter "2" specifies the image (1 through 4) in flash ROM for which the copy of the program resides.

Like the original **RUN** command, this form of the command can only be executed from Task 1.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.55. SIN Command (Sine)

The **SIN** function converts a floating point number to the sine value of an angle in radians. It is important the user understands not to equate this function to variables of the type integer or long integer, since the result of the result of this conversion is always a number between ±1.0.

SYNTAX: *<var>=SIN(<var or constant>)*

EXAMPLE:

```
BEGIN                   ; Start the program.
FV:1=SIN(.5236)         ; Set floating point, variable 1 equal to the sine
                        ; of .5236 radians or 30°.  The FV:1 variable
                        ; stores the result of .500001.
FV:2=SIN(.785)          ; Set floating point, variable 2 equal to the sine
                        ; of .785 radians or 45°.  The FV:2 variable
                        ; stores the result of .706825.
FV:3=SIN(1.0472)        ; Set floating point, variable 3 equal to the sine
                        ; of 1.0472 radians or 60°.  The FV:3 variable
                        ; stores the result of .866027.
FV:3=0                  ; Set floating point, variable 3 equal to zero.
FV:4=SIN(FV:3)          ; Set floating point, variable 4 equal to the sine
                        ; of floating point, variable 3.  The FV:4 variable
                        ; stores the result of .000000.
END                     ; End of the program.
```

Related Commands

COS, TAN

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.56. SRQ Command (Service Request)

The **SRQ** command allows the controller to request the attention of the master controller. When executing this command the controller sends a service request. After sending the service request to the Host, the controller (master device) waits for a reply acknowledging receipt of the request. Task 1 and Task 2 each have one service request reserved and up to fifteen user defined status codes. A variable may contain the SRQ code, however it must be an integer from 1 to 15.

The communications link used to transfer the service request may be through an RS-232, RS-422, or an IEEE-488.

SYNTAX: *SRQ(<var or constant>)*

EXAMPLE:

```
BEGIN                ; Start the program.
BV:1=5               ; Set integer, variable 1 equal to five.
SRQ(BV:1)            ; Send the Service Request (status code #5) to the
                     ; Host and wait for an acknowledge signal.
V(10000)             ; Set the velocity equal to 10000.
D(50000)             ; Set the distance equal to 50000.
GO                   ; Start the move.
END                  ; End the program.
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.57. SQRT Command (Square root)

the **SQRT** command returns the square root value based on some constant or variable. This constant or variable must be positive and in the form of an integer, long integer OR floating point number.

To take the square root of a number that is not a perfect square it is necessary that this number be equated to a floating point variable.

SYNTAX: *<var>=SQRT(<var or constant>)*

EXAMPLE:

```
BEGIN                   ; Start the program.
BV:1=4                  ; Set integer, variable 1 equal to four.
BV:2=SQRT(64)           ; Set integer, variable 2 equal to the square root
                        ; of sixty-four.  The result is eight.
BV:3=SQRT(BV:1)         ; Set integer, variable 3 equal to the square root
                        ; of integer, variable 1.  The result is two.
FV:1=SQRT(10.0)         ; Set floating point, variable 1 equal to the
                        ; square root of ten.  The result is 3.162277.
BV:4=SQRT(10.0)         ; Set integer, variable 4 equal to the square root
                        ; of ten.  The result is three.
END                     ; End of the program.
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.58. SUB Command

A **SUB** command identifies the beginning of a subroutine. This command is paired with the ENDSUB statement to define the body of a subroutine.

SYNTAX: *SUB:<var or constant>*

EXAMPLE:

```
BEGIN                ; Start the program.
REG:2=0Xff           ; Set the bit output register #2 equal to 0Xff.
D(100)               ; Set the distance equal to one hundred.
V(100)               ; Set the velocity equal to one hundred.
T(.1)                ; Set the time equal to .1 second.
GO                   ; Start the move.
GOSUB:10             ; Go to subroutine 10.
EXIT                 ; Exit the program.
                     ;
SUB:10               ; Define the following as subroutine 10.
REG:2=0Xfd           ; Set the output bit register 2 equal to 0Xfd.
DW(1.0)              ; Set the dwell time to one second.
REG:2=0Xff           ; Set the output bit register 2 equal to 0Xff.
ENDSUB               ; End the subroutine.
END                  ; End the program.
```

Related Commands

ENDSUB, EXIT, GOSUB

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.59. - Subtract

The - (subtraction) function allows for subtraction of integers, long integers, floating point, and hexadecimal numbers. Expressions may also contain variables and therefore it is possible to subtract both numbers and variables in any combination.

The controller truncates the decimal portion of an expression that contains a fraction if it is being equated to an integer variable (e.g., BV:1 or LV:1=5.5-3.25).

SYNTAX: *<var or constant>-<var or constant>*

EXAMPLES:

```
BEGIN                   ; Start the program.
BV:1=5                  ; Set integer, variable 1 equal to five.
BV:2=10                 ; Set integer, variable 2 equal to ten.
BV:3=BV:1-BV:2          ; Set integer, variable 3 equal to the difference
                        ; between integer, variable 1 and integer,
                        ; variable 2.  The result is negative five.
BV:4=BV:1-5             ; Set integer, variable 4 equal to the difference
                        ; between integer, variable 1 and five.  The
                        ; result is zero.
END                     ; End the program.
```

```
BEGIN                   ; Start the program.
BV:1=5                  ; Set integer, variable 1 equal to five.
PV:2=10.5               ; Set port, variable 2 equal to ten and a half.
BV:3=PV:2-BV:1          ; Set integer, variable 3 equal to the difference of
                        ; port, variable 2 and integer, variable 1.  The
                        ; result is five.
END                     ; End the program.
```

Related Commands

+, *, /, =

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.60. SYNC Command

The **SYNC** command is used to synchronize a command to its task. Due to controller's multitasking operation, certain commands may never fully execute before changing the task. Commands that use parameters, variables (excluding the BV: or PV:) or registers, and appear in many other tasks are vulnerable to data corruption. To reduce the chances of having this problem, a SYNC command should precede any command that requires full execution before switching the task. The SYNC command causes the command that follows to delay execution until re-entry of the task. This provides the command with the most time possible to complete, but may also cause a slow down in program operation.

SYNTAX: *SYNC*

EXAMPLE:

```
BEGIN                ; Start the program.
FV:2=1.25            ; Set floating point, variable one equal to 1.25
FV:3=2.5             ; Set floating point, variable equal to 2.5.
SYNC                 ; Synchronize the following command.
FV:1=FV:2+FV:3       ; Set floating point, variable 1 equal to the sum
                     ; of floating, point variable 2 and floating point,
                     ; variable 3.
FV:2=1.3             ; Set floating point variable equal to 1.3
END                  ; End the program.
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.61. T Command (Ramp Time)

The **T** command is the ramp time command. It sets the time required for the acceleration and deceleration ramp rate. This command is an alternative to the Ac/De "A( )" ramp rate command. When entering the "T( )" command, the user must enter the ramp time in seconds.

The value defined with this command remains in effect until entry of a new T command. The associated default parameter is PRM:104, and does not get overwritten when using this command.



It is possible to change the parameters default from within the program, but this change is global and affects all programs using the T command.

SYNTAX: *T(<var or constant>)*

EXAMPLE:

```
BEGIN          ; Start the program.
HM             ; Initialize the home position to zero.
INCR           ; Initialize the incremental (distance) mode.
T(.2)          ; Set the ramp time to .2 seconds.
V(20000)       ; Set the velocity equal to 20000 units per
               ; second.
D(100000)      ; Move a distance of 100000 units in the positive
               ; CW direction.
GO             ; Start the move from the home position.
END            ; End the program.
```

Related Commands

A, D, GO, V

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.62. TAN Tangent

The **TAN** function converts a floating point number to the tangent value of an angle in radians.

SYNTAX: *<var>=TAN(<var or constant>)*

EXAMPLE:

```
BEGIN                   ; Start the program.
FV:1=TAN(.5236)         ; Set floating point, variable 1 equal to the
                        ; tangent of .5236 radians or 30°.  The FV:1
                        ; variable stores the result of .577352.
FV:2=TAN(.785)          ; Set floating point, variable 2 equal to the
                        ; tangent of .785 radians or 45°.  The FV:2
                        ; variable stores the result of .999204.
FV:3=TAN(1.0472)        ; Set floating point, variable 3 equal to the
                        ; tangent of 1.0472 radians or 60°.  The FV:3
                        ; variable stores the result of 1.732060.
FV:3=0                  ; Set floating point, variable 3 equal to zero.
FV:4=TAN(FV:3)          ; Set floating point, variable 4 equal to the
                        ; tangent of floating point, variable 3.  The FV:4
                        ; variable stores the result of .000000.
END                     ; End of the program.
```

Related Commands

COS, SIN

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.63. TITLE Command

The **TITLE** command permits the user to attach a description to a program. The title command is optional and must be the first statement in the PGM file. Programs that contain titles show the title along with the program number when displaying a directory.



This command is case sensitive, use capital letters when entering this command.

The description attached to a title may consist of up to seventy-three printable characters and preceded by a space. It is best to limit this description to twenty characters or less.

SYNTAX: *TITLE<space><description>*

EXAMPLE:

```
TITLE motion        ; Attach the title motion to the program number.
BEGIN               ; Start the program.
D(100)              ; Set the distance equal to one hundred.
V(100)              ; Set the velocity equal to one hundred.
T(.1)               ; Set the time equal to .1 second.
GO                  ; Start the move.
DW(1)               ; Set the dwell time equal to one second.
END                 ; End the program.
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.64. TUNE Command (Autotune the system)

The **TUNE** command provides the capability for the controller to automatically choose the controller gains. The user specifies the damping factor (PRM:229) and the corner frequency (PRM:230) (in Hertz) and the algorithm will choose the controller gains **Kpos** (PRM:220), **Kp** (PRM:218) and **Ki** (PRM:219). Other parameters associated with this command are PRM:227, PRM:228 and PRM:105. PRM:227 is the starting frequency of the tuning algorithm. PRM:228 represents the time given to the tuning algorithm to determine the gains and PRM:105 is the distance (in machine steps) the motor will travel from its starting point during the tuning process.

The damping factor (PRM:229) and the corner frequency (PRM:230) determine the approximate bandwidth of the velocity loop. The position loop gain, **Kpos** (PRM:220) is determined such that the bandwidth of the position loop is 20% of the velocity loop bandwidth. After executing the TUNE command, the closed loop velocity transfer function is given by

$$V_{cl} = \frac{(2\zeta\omega_n S + \omega_n^2)}{(S^2 + 2\zeta\omega_n + \omega_n^2)}$$

where ζ is the damping factor (PRM:229) and ωn (PRM:230) is the corner frequency. As the damping factor approaches zero, the response of the system becomes more oscillatory. For values of PRM:229 between 0 and 1, the system is under-damped and the closed loop poles are complex. For PRM:229>1, the system is over-damped and the closed loop poles are real. For PRM:229=1, the system is critically damped and the closed loop poles are real. The default value for this parameter is 0.5. Note that including velocity feedforward (PRM:222=1) substantially increases the bandwidth of the position loop.

When tuning systems with linear positioning stages, the user must make sure the stage does not enter a hardware or software limit. To avoid this potential problem, PRM:105 should be set to a value such that the distance from the starting point to PRM:105 is less than the distance from the starting point to the limit. For example, if the nearest limit to the starting point is 100000 machine steps, then PRM:105 < 100000. A typical value for PRM:105 is 32000 machine steps. With a 1000 line encoder (4000 counts/rev in x4 mode), this would cause the motor to turn 8 revolutions.

The PID current limit parameter (PRM:216) should be set to the amplifiers maximum value. If this value is too low, erroneous results may be achieved.

Should the user change the resolution of the position measurement, i.e., use a different resolution encoder, or change the servo loop update time, then the system must be retuned.

SYNTAX: *TUNE*

If a fault occurs while the system is autotuning or if the tuning algorithm produces gains that are out of range, Kp and Ki will revert back to their default values.



Assign a small value to PRM:105 (e.g., 32000), the reason is, the larger the value of PRM:105, the faster the motor will move.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



If the tuned gains cause the system to oscillate or the motor is noisy, then either the damping factor (PRM:229) should be changed or the corner frequency (PRM:230) should be lowered.

The **TUNE** command can be used only when the UNIDEX 100 is configured for DC brush or AC brushless motor operation.

5.7.65. V Command (Velocity)

The **V** command is the velocity command. This command sets the velocity for the associated commands in a motion block. The motion block may consist of a T, an A command, a D command, and a corresponding GO command to complete the block. When entering this command, the user must enter the velocity in units per second. Often an insufficient distance prevents the velocity from being attained, therefore the controller permits the user to alter these velocities. Velocities are in user units per second and determined by the setting of parameter PRM:100.

The value defined with this command remains in effect until entry of a new V command. The associated default parameter is PRM:107, and does not get overwritten when using this command.

SYNTAX: *V(<var or constant>)*



It is possible to change the parameters default from within the program, but this change is global and affects all programs using the **V** command.

EXAMPLE:

```
BEGIN          ; Start the program.
HM             ; Initialize the home position to zero.
INCR           ; Initialize the incremental (distance) mode.
T(.2)          ; Set the ramp time to .2 seconds.
V(50000.)      ; Set the velocity equal to 50000. units per
               ; second.
D(1000000)     ; Move a distance of 1000000 units in the
               ; positive CW direction.
GO             ; Start the move from the home position.
END            ; End the program.
```

Related Commands

A, D, GO, T

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.66. WHL Command

The **WHL** command performs a logical test of two operands and if true, executes the following commands until encountering an "ENDWHL" command. If the condition tested is true, the "ENDWHL" command directs operation back to the while condition and retests the condition. If the condition is false, the loop never executes and operation continues at the first command following the end of the while loop.

SYNTAX: *WHL(<var or constant> op <var or constant>)*

The operator "OP" used with this statement can be one of the following.

EQ	equal to
GE	greater than or equal to
LE	less than or equal to
GT	greater than
LT	less than
NE	not equal to

The operator "OP" can also be one of the following.

AND	binary AND, if result is "0", statement is false, otherwise statement is true.
OR	binary OR, if result is "0", statement is false, otherwise statement is true.

EXAMPLE:

```
BEGIN                   ; Start the program.
BV:1=0                  ; Set integer, variable 1 equal to zero.
CLS                     ; Clear the terminal screen.
WHL(BV:1 NE 10)         ; While integer, variable 1 is not equal to ten,
                        ; continue with the loop.  Otherwise don't
                        ; execute.
  CUR(1,1)              ; Position the cursor at row 1, column 1.
  PM(BV:1, equals %d")  ; Print the message to the terminal screen.
  DW(1)                 ; Set the dwell time equal to one second.
  INC(BV:1)             ; Increment the integer, variable 1 by one.
ENDWHL                  ; End the while loop.
END                     ; End the program.
```

Related Commands

ENDWHL

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.7.67. XOR Exclusive OR

The **XOR** function permits the user to EXCLUSIVE OR two binary numbers. The expression may contain a number, a variable, or a combination of both.

SYNTAX: *(<var or constant>XOR<var or constant>)*

EXAMPLE:

```
BEGIN                   ; Start the program.
BV:1=2                  ; Set integer, variable 1 equal to two.
BV:2=BV:1 XOR 2         ; Set integer variable 2 equal to integer variable 1
                        ; XOR 2.  The result is false and returns zero.
                        ; For example: binary 2 XOR binary 2 = binary 0
                        ; │00000010 XOR ½00000010= ½00000000
BV:3=2 XOR 3            ; Set integer, variable 3 equal to 2 XOR 3.  The
                        ; result is true and returns a one.
                        ; For example: binary 2 XOR binary 3 = binary 1
                        ; │00000010 XOR ½00000011 = ½00000001
END                     ; End of the program.
```

Related Commands

AND, OR

∇ ∇ ∇

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CHAPTER 6: PARAMETERS

In This Section:

- Introduction 6-1
- Communication Parameters 6-2
- Trajectory Generation 6-22
- Motion Parameters 6-24
- Drive Parameters 6-46
- System Parameters 6-59

6.1. Introduction

This chapter describes all of the parameters used by UNIDEX 100/U100i. Parameters are divided into 4 sections based on the type of parameters available. Those parameters are:

- Communication
- Motion
- Drive
- and System

Parameters are one of three data types supported by the controller. The other two types, registers and variables are discussed in the following chapter. The main purpose of data types is to exchange information, perform calculations, and control. Parameters have an assigned prefix (PRM) and require an integer index constant to access a particular element. This index constant must be proceeded by a colon ":" (e.g., PRM:312).

Parameters can be used in mathematical statements with integer constants. For example:

PRM:012=14

PRM:013=REG:100*2

Additional sections are included to explain topics that are closely related to the parameters. Parameters are listed in tables in the respective sections. Default parameters are shown in bold for easy reference.

These parameters hold the values necessary to configure the controller and customize it to the user's requirements. These requirements depend on the application being implemented.

With integer constants, including those used to index variable elements it is necessary to observe certain rules when expressing them in program statements (e.g., statements in a PGM file or an immediate command statement used in the MDI mode). These rules apply to base interpretation. The controller recognizes three base systems; they are:

- Octal (base 8)
- Decimal (base 10)
- and Hexadecimal (base 16)

The default base is "decimal". The following are examples of each.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Base 8 - “Octal” uses a leading “0” in front of the constant. For example:

PRM:310=021 is equivalent to PRM:301=17 in base 10.

Base 10 - “Decimal” is the default base and requires no leading qualifier.

Base 16 - “Hexadecimal” uses a leading “0x” in front of the constant. For example:

PV:0x100=1234 is equivalent to PV:256=1234 in base 10.



“Hard coded” default settings are initialized when performing a setup function and are stored in the EPROM. These values are listed in the summary charts for each group within this chapter.



The other set of parameter defaults are “customized”. They are factory selected to meet a particular application and are not to be changed by the user. These parameters are initialized with the PGM100 file provided with the system's software disks. The values chosen for these parameters appear on a hard copy titled "UNIDEX 100 Parameter Specifications" also referred to as ES12498. Parameters can also be saved in PRM files using the HOST_100 software.

6.2. Communication Parameters

The communication parameters are used to define RS-232 and IEEE-488 communications. There are 41 parameters in the communication parameters group. These parameters are listed Table 6-1 and explained in detail in the sections that follow.



The communication parameters have there indexes expressed in “decimal” even though their indexes are shown with a leading “0”. Be careful accessing these parameters in a PGM file or MDI command mode with a leading zero. Unless it is intended that the index be expressed in “octal” (e.g., PRM:009 would be entered as PRM:9 if it is intended that the “9” be interpreted as a decimal number).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 6-1. Communication Parameters

PRM	Description	Value Type	Default (Hard coded)	Min Value	Max Value
000	This Parameter is not used	-	-	-	-
001	RS-232-C General Control Register	integer	0x302	0	0x7fff
002	RS-232-C Clock Control Register	integer	0x20	0	0x7fff
003	RS-232-C Loop Control Character	integer	0x3	0	0x1f
004	RS-232-C Loop Address	integer	0x41	0x41	0x5A
005	RS-232-C Loop Enable/Disable	integer	0	0	1
006	RS232-C Loop Group Trigger	integer	0x2	0	0x1f
007	IEEE-488 Address Mode Register	integer	0x21	0	0xff
008	IEEE-488 Address Register "0"	integer	0x24	0	0xff
009	IEEE-488 Address Register "1"	integer	0x44	0	0xff
010	IEEE-488 Auxiliary Register "A"	integer	0x4	0	0x1f
011	IEEE-488 Auxiliary Register "B"	integer	0x2	0	0x1f
012	IEEE-488 Command Register	integer	0x0	0	0x1f
013	IEEE-488 Auxiliary Register "E"	integer	0x0	0	0x3
014	IEEE-488 Parallel Poll Register	integer	0x0	0	0x1f
015	IEEE-488 Timer Control Register	integer	0x5	0	0xf
016	IEEE-488 End of Sequence Register	integer	0xA	0	0xff
017	Reserved	-	-	-	-
018	Reserved	-	-	-	-
019	Communications Mode	integer	1	1	2
020	RS232-C Group Trigger Enable/Disable	integer	0	0	1
021	Cursor Control for "CUR( )" Command	integer	1	0	1
022	Communications SRQ On/Off	integer	1	0	1
023	Send SRQ At End of Task 1	integer	0	0	1
024	Send SRQ At End of Task 2	integer	0	0	1
025	LST1 File Generation On/Off	integer	0	0	1
026	Auto DEF100/MAC100 Lookup in MDI mode	integer	-	-	-
027	End Of File Character Code	integer	0x7E	0	0xff
028	Service Request Character Code	integer	0x16	0	0x1f
029	Service Acknowledge Character Code	integer	0x17	0	0x1f
030	Task 1 Boot Program	integer	0	0	9999
031	Task 2 Boot Program	integer	0	0	9999
032	RS-232-C Xon Sent From Host	integer	0x1C	0	0x1f
033	RS-232-C Xoff Sent From Host	integer	0x1D	0	0x1f
034	Software Reset (HT Mode)	integer	0x4	0	0x1f
035	Software Reset (Host Mode)	integer	0x1E	0	0x1f
036	Motion Status Total Digits (min.), Line 1	integer	10	1	12
037	Motion Status Decimal Digits, Line 1	integer	2	0	12
038	Motion Status Total Digits (min.), Line 2	integer	10	1	12
039	Motion Status Decimal Digits, Line 2	integer	2	0	12
040	Motion Status Scan Type, Line 2	integer	1	0	3
041	Motion Status Scan Type, Line 1	integer	1	0	3

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.2.1. RS-232-C General Control Register PRM:001

PRM:001 sets the RS-232 operating mode. It controls word formats, enables transmit and receive, and RS-232 interrupts. The lower 3 bits of this parameter set the word format (refer to Table 6-2). The default word format code is 010 for the lower three bits. The upper thirteen bits must be the same as that specified by the default.

The default value for this parameter is 0x302. Change this value by entering an integer from 0x300 to 0x307.

Table 6-2. RS-232-C Bit Data

BIT 2	BIT 1	BIT 0	WORD FORMAT
0	0	0	8-bit synchronous data (shift register mode)
0	0	1	Reserved
0	1	0	10-bit Asynchronous (1 start, 8 data, 1 stop)
0	1	1	Reserved
1	0	0	11-bit asynchronous (1 start, 8 data, 1 even parity, and 1 odd)
1	0	1	11-bit asynchronous (1 start, 8 data, 1 odd parity, 1 stop)
1	1	0	11-bit multidrop (1 start, 8 data, 1 type, 1 stop)
1	1	1	Reserved



To activate this parameter after its been changed, the controller must be reset. To initialize a reset hit **<CTRL>** + **<D>** .

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.2.2. RS-232-C Clock Control Register PRM:002

This parameter selects the RS-232 clock source and baud rate. The baud rate is set using the lower twelve bits (refer to Table 6-3).

The default value for this parameter is 0x20 (baud rate = 9600). Change this value by entering an integer from 0 to 0x7fff.

Table 6-3. Parameter Settings for Clock Control Register (PRM:002)

Baud Rate (BPS)	SCP Bit (13)	Divider Bits (CD0-CD11)	Baud Rate Error, Percent	PRM:002
320.0K	0	0x000	0	0x0000
56.0K	0	0x005	4.762	0x0005
38.4K	0	0x007	4.167	0x0007
19.2K	0	0x010	1.961	0x0010
9600	0	0x020	1.010	0x0020
8000	0	0x027	0	0x0027
4800	0	0x042	0.498	0x0042
2400	0	0x084	0.251	0x0084
1200	1	0x020	1.010	0x2020
600	1	0x042	0.498	0x2042
300	1	0x084	0.251	0x2084

To activate this parameter after its been changed, the controller must be reset. To initialize a reset hit **<CTRL>** + **<D>** .



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.2.3. RS-232-C Loop Control Character PRM:003

This character (set by a Host Controller - Hand held Terminal or PC) acts as a control flag to signal the closing or opening of the controller, when in the RS-232 communications loop mode. An address character always follows the character (see PRM:004).

The default value for this control character is "**CTRL-C**" (**0x3**). Change this value by entering an integer from 0 to 0x1f.

6.2.4. RS-232-C Loop Address PRM:004

The Loop Address character (always follows the Loop Control character, see PRM:003) selects the specific controller that the "loop" is to address. The user can set this address as the ASCII character "A" (0x41) through "Z" (0x5A). Each controller on the loop must have a unique address character. To close the RS-232 loop, send the space character (0x20) after the control character (PRM:003). After closing the loop, the user can address any controller that is on the loop.

The default value for this parameter is **0x41** (or "**A**"). To change the default character, enter a number between 0x41 through 0x5A.



"0x41" corresponds to the primary controller.

6.2.5. RS-232-C Daisy Chain Enable/Disable PRM:005

Parameter PRM:005 enables and disables the "Daisy Chain loop" depending on the value assigned to it. When set to (1), the "Daisy Chain Loop" is enabled. Setting the parameter to zero (0) disables the "Daisy Chain Loop", allowing the controller to communicate directly through the Host's RS-232 port.

The default value for this parameter is zero (0). Refer to Table 6-4 for parameter settings.

Table 6-4. Settings for Daisy Chain Enable/Disable (PRM:005)

Mode	PRM:005
Daisy chain enabled	1
Daisy chain disabled	**0** (default)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.2.5.1. Daisy Chain Operation

Up to 26 controller's may be daisy chained together when more than 1 axis is needed. Before it is possible to place the controller in a daisy chain loop, each controller must be set to its individual address character and the daisy chain mode enabled. To set the controller addresses, set PRM:004 to a letter from "A" to "Z". The default for this parameter is "A" (decimal value = 65 or Hex value = 0x41). The address character is entered as a numerical value. After the address character is set, the daisy chain must be enabled by setting PRM:005 to 1.

The following procedures describe how to set the controller for daisy chain operation at address "C".

1. Set address character to "C" by setting PRM:004 equal to 67 decimal or 0x43 Hex.
2. Enable daisy chain by setting PRM:005 to a 1.
3. Power down or reset the controller to ensure daisy chain operation.

Refer to Figure 6-1 and Figure 6-2 for examples of a 4 axis controller daisy chain configuration.

In order to address a controller, the user must send a sequence of deactivation commands (CTL "C" character (03 Hex) and a space character). Repeat this sequence until the command sequence is received by the sending PC. Following the deactivation sequence, address the controller by sending the "CTL C" character (03 Hex) followed by the address character (defined by parameter "PRM:004"). Wait approximately .1 second for the controller to enable the address, then send a F1 character (Hex value 0x11) to initiate the COM port. Be prepared to receive and acknowledge a service request.

Using the trigger command in the daisy chain configuration allows the user to initiate programs of two or more controllers at the same time. Setting parameter "PRM:020" to (1) enables the trigger command. In order to perform a trigger operation, the user must initiate each command or program individually in all the controllers to be triggered. Next, the must deactivate (close loop) all controllers using the CTL C character and SPACE character sequence used to deactivate a new address. After deactivating (loop is closed) the controller, the trigger character (the character set in PRM:006) is sent and every controller setup for a trigger operation will begin execution of the command or program.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

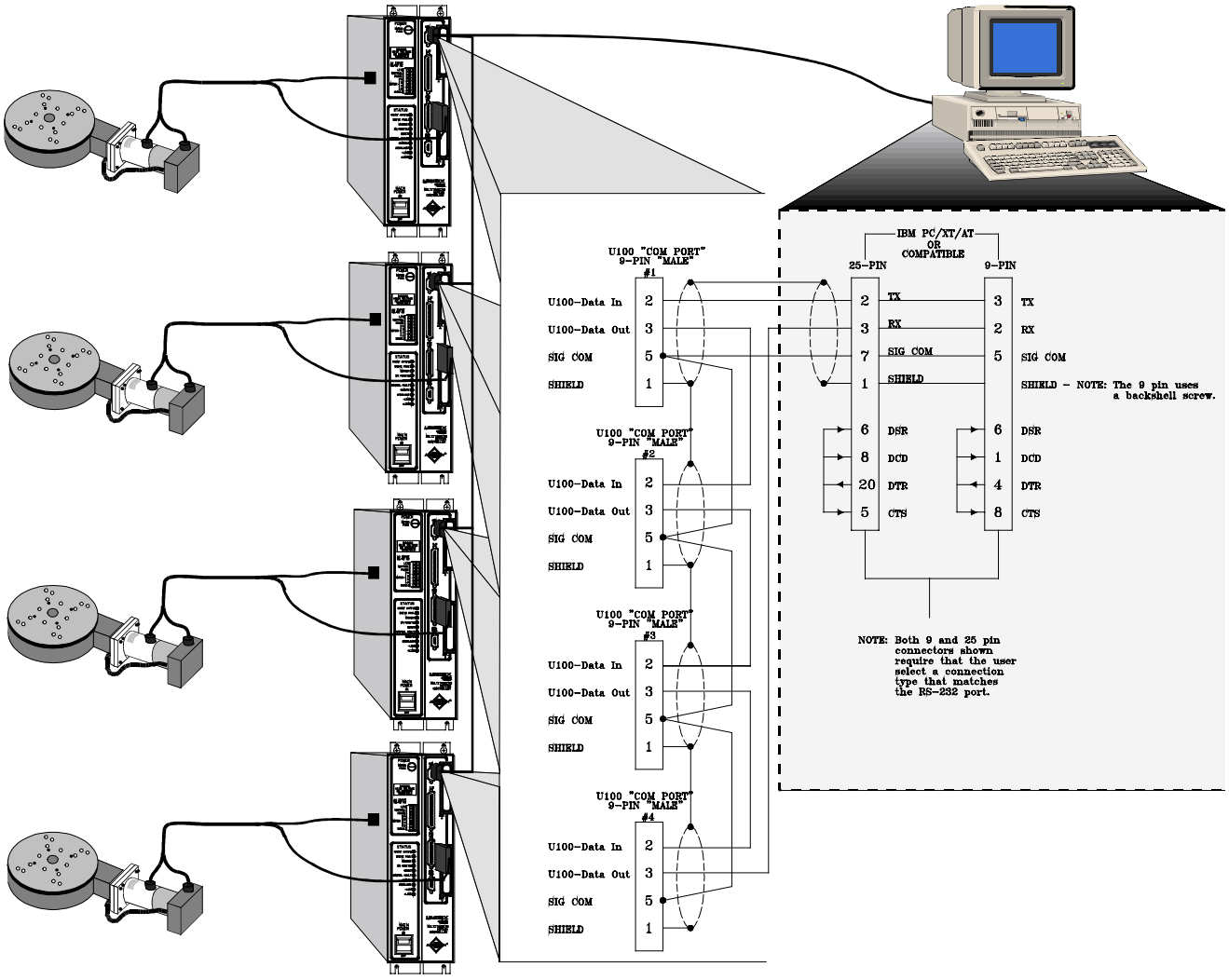


Figure 6-1. 4-Axis Daisy Chain Configuration

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

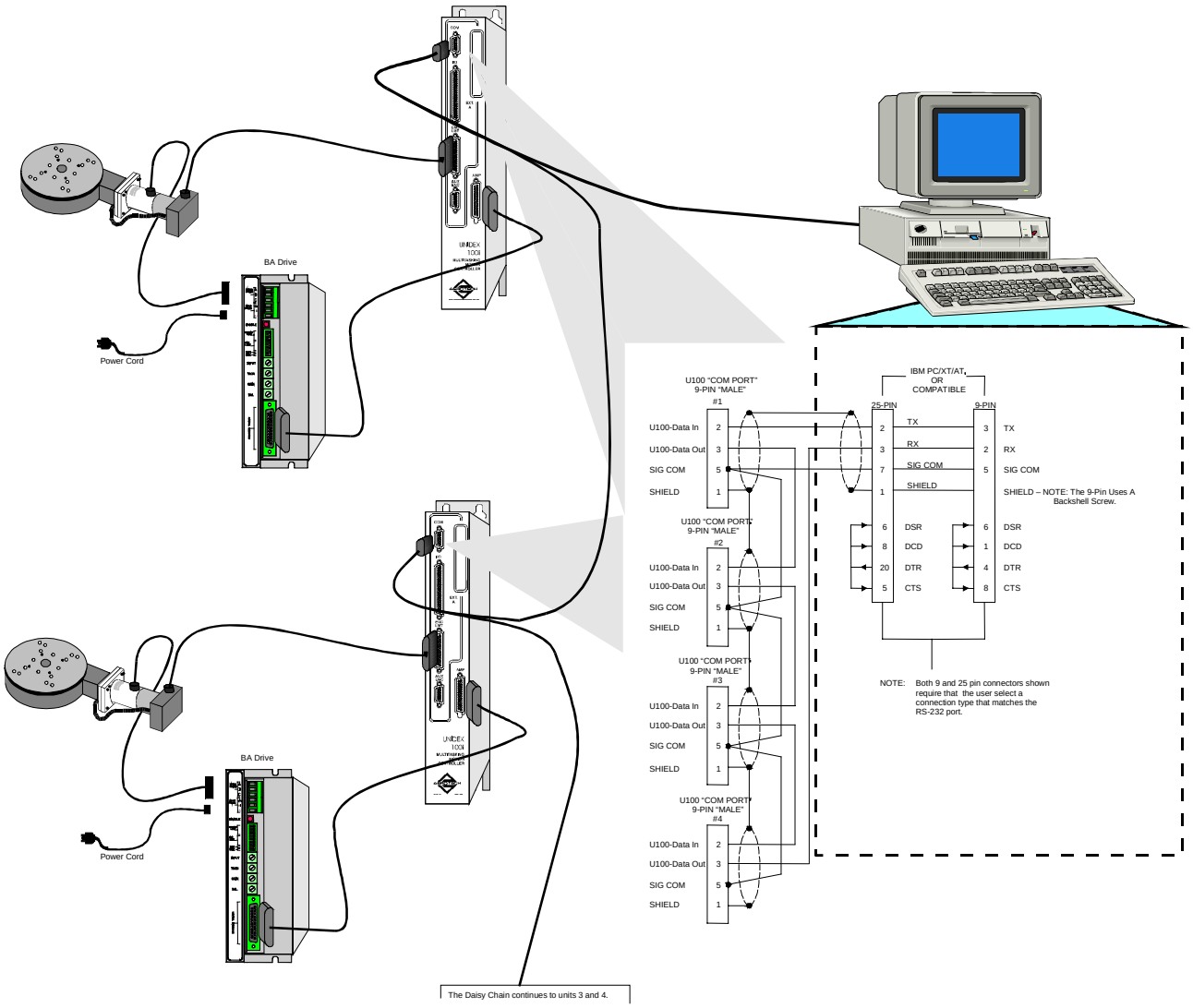


Figure 6-2. 4-Axis Daisy Chain Configuration (U100i)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.2.6. RS-232-C Loop Group Trigger PRM:006

This parameter selects the "group trigger" character. If the Group Trigger Enable/Disable parameter (PRM:020) contains a one (1), any command sent through the Immediate Mode (MDI) to Task 1 will wait for the trigger. Any program selected to run in Task 1 (Run Mode) is inhibited from execution until this character gets sent through the RS-232 Loop. The user must close the loop before this character gets sent (see PRM:003 and PRM:004).

The default value for this parameter is **0x2**. Change this value by entering a number between 0x0 and 0x1f.

6.2.7. IEEE-488-Address Mode Register PRM:007

Parameter PRM:007 selects the address mode of the IEEE-488 device and sets the mode for the transceiver control lines.

The default value for this parameter is **0x21**. Change this value by entering an integer from 0 to 0xff.



To activate this parameter after it has been changed, the controller must be reset. To initialize a reset hit **<CTRL>+<D>.**

6.2.8. IEEE-488 Address Register "0" PRM:008

PRM:008 sets the "zero" address mode, talker enable, and the listener enable status . The lower five bits are the device address bits (default is address #4). Bit #6 is the listener bit (default is enabled). Bit #7 is the talker bit (default is disabled).

The default value for this parameter is **0x24**. Change this value by entering an integer from 0 to 0xff.



To activate this parameter after it has been changed, the controller must be reset. To initialize a reset hit **<CTRL>+<D>.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.2.9. IEEE-488 Address Register "1" PRM:009

Parameter PRM:009 sets the "one" address mode, talker enable, and the listener enable status . The lower five bits are the device address bits (default is address #4). Bit #6 is the listener bit (default is disabled). Bit #7 is the talker bit (default is enabled).

The default value for this parameter is **0x44**. Change this value by entering a value from 0 to 0xff.

To activate this parameter after it has been changed, the controller must be reset. To initialize a reset hit **<CTRL>+<D>.**



6.2.10. IEEE-488 Auxiliary Register "A" PRM:010

This parameter specifies special operating features of the IEEE-488 device.

The default value for this parameter is **0x4**. Change this value by entering an integer from 0 to 0x1f.

To activate this parameter after it has been changed, the controller must be reset. To initialize a reset hit **<CTRL>+<D>.**



6.2.11. IEEE-488 Auxiliary Register "B" PRM:011

This parameter specifies special operating features of the IEEE-488 device.

The default value for this parameter is **0x2**. Change this value by entering an integer from 0 to 0x1f.

To activate this parameter after it has been changed, the controller must be reset. To initialize a reset hit **<CTRL>+<D>.**



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.2.12. IEEE-488 Command Register PRM:012

PRM:012 provides commands for the IEEE-488 device using the lower five bits.

The default value established for this parameter is **0x0**. Change this value by entering an integer from 0 to 0x1f.



To activate this parameter after it has been changed, the controller must be reset. To initialize a reset hit **<CTRL>+<D>.**

6.2.13. IEEE-488 Auxiliary Register "E" PRM:013

Parameter PRM:013 controls the data acceptance modes of the IEEE-488 device.

The default value for this parameter is **0x0**. Change this value by entering an integer from 0 to 0x3.



To activate this parameter after it has been changed, the controller must be reset. To initialize a reset hit **<CTRL>+<D>.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.2.14. IEEE-488 Parallel Poll Register PRM:014

This parameter defines the parallel poll response of the IEEE-488 device.

The default value for this parameter is **0x0**. Change this value by entering an integer from 0 to 0x1f.

To activate this parameter after it has been changed, the controller must be reset. To initialize a reset hit **<CTRL>+<D>.**



6.2.15. IEEE-488 Timer Control Register PRM:015

This parameter controls the IEEE-488 internal counter.

The default value for this parameter is **0x5**. Change this value by entering an integer from 0 to 0xf.

To activate this parameter after it has been changed, the controller must be reset. To initialize a reset hit **<CTRL>+<D>.**



6.2.16. IEEE-488 End of Sequence Register PRM:016

This parameter holds a seven or eight bit End of Sequence (EOS) character byte used to detect the end of a data block.

The default value for this parameter is **0xA**. Change this value by entering an integer from 0 to 0xff.

To activate this parameter after it has been changed, the controller must be reset. To initialize a reset hit **<CTRL>+<D>.**



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.2.17. Communications Mode PRM:019

PRM:019 selects the type of Communications Mode to use. There are two choices: RS-232 or IEEE-488.

The default value for this parameter is one (**1**). Refer to Table 6-5 for alternate setting.



To activate this parameter after it has been changed, the controller must be reset. To initialize a reset hit **<CTRL>+<D>.**



To activate the IEEE-488 Communications Mode, it is also necessary to set PRM:005 to zero (0).

Table 6-5. Settings for Communications Mode (PRM:019)

Mode	PRM:019
input and output are through the RS-232 port	**1** (default)
input and output are through IEEE-488 port	2

6.2.18. RS-232-C Group Trigger Enable/Disable PRM:020

Parameter PRM:020 enables and disables the Group Trigger Mode. MDI commands and programs in Task 1 require a trigger command character (defined by PRM:006) before the command will execute. The default value established for this parameter is zero (**0**). Refer to Table 6-6 for alternate setting.



PRM:005 must be set for Daisy Chain operation.

Table 6-6. Settings for Group Trigger Enable/Disable (PRM:020)

Function	PRM:020
disable the RS-232 Trigger Mode	**0** (default)
enable the RS-232 Trigger Mode	1

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.2.19. Cursor Control for "CUR( )" Command PRM:021

This parameter allows the blinking display cursor to be turned on or off when running a program with the Communications commands. This is particularly useful when "scanning" a variable or register and displaying it with the "PM( )" command.

Entering a one (1) for this parameter turns on the cursor while a zero (0) entry turns the cursor off. The cursor state change takes effect only after a "CUR( )" command is issued. If disabled, the cursor is automatically enabled when the program (PGM file) exits a task.

The default value for this parameter is one (**1**). Refer to Table 6-7 for alternate setting.

Table 6-7. Settings for Cursor Control (PRM:021)

Function	PRM:021
disable the cursor control	0
enable the cursor control	**1** (default)

6.2.20. Communications SRQ On/Off PRM:022

PRM:022 enables or disables the following service requests when operating in the Host Mode (RS-232 or IEEE-488).

0x1 Program Uploading in Progress.
0x2 Program Downloading in Progress.
0x5 "Line #" update for Task 1 Window.
0x6 "Line #" update for Task 2 Window.
0x7 Ready for commands in Host Mode.

The default value for this parameter is **1**. Refer to Table 6-8 for alternate setting.

Table 6-8. Settings for Communications SRQ On/Off (PRM:022)

Function	PRM:022
disable the Service Request	0
enable the Service Request	**1** (default)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.2.21. Send SRQ At End of Task 1 PRM:023

This parameter when set to a one (1) automatically sends a service request at the end of execution of a command entered in the MDI Mode or a PGM program running in the Run Mode for Task 1. The service request number sent is 0x11. Entering a zero (0) for this parameter inhibits sending a service request.

The default value for this parameter is zero (**0**). Refer Table 6-9 for alternate setting.

Table 6-9. Settings for Send SRQ At End of Task 1 (PRM:023)

Function	PRM:023
disable the sending of Service Request 0x11 for Task 1	**0** (default)
enable the sending of Service Request 0x11 for Task 1	1

6.2.22. Send SRQ At End of Task 2 PRM:024

This parameter when set to a one (1) automatically sends a service request at the end of execution of a command entered in the MDI Mode or a PGM program running in the Run Mode for Task 2. The service request number sent is 0x21. Entering a zero (0) for this parameter inhibits sending a service request.

The default value for this parameter is zero (**0**). Refer to Table 6-10 for alternate setting.

Table 6-10. Settings for Send SRQ At End of Task 2 (PRM:024)

Function	PRM:024
disable the sending of Service Request 0x21 for Task 2	**0** (default)
enable the sending of Service Request 0x21 for Task 2	1

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.2.23. LST1 File Generation On/Off PRM:025

This parameter when set to one (1) causes the controller to generate a LST1 (list) file when a program is compiled. The user may examine this file (called LST1) to help determine compiler errors. Setting PRM:025 to zero (0) suppresses the LST1 file generation.

The default value for this parameter is a zero (**0**). Refer to Table 6-11 for alternate setting.

To guarantee that the controller does not overwrite an existing list file with a newly generated LST1 file, copy the existing LST1 file to another number.

For compiling large program files it may be necessary to turn Off LST1 file generation to conserve memory.



Table 6-11. Settings for LST1 File Generation On/Off (PRM:025)

Function	PRM:025
turn Off controllers ability to create LST1 files	**0** (default)
turn On controllers ability to create LST1 files	1

6.2.24. Auto DEF100/MAC100 Lookup in MDI Mode PRM:026

Parameter PRM:026 allows any command entered in the MDI mode to be optionally processed through the files DEF100 and MAC100. These files must be resident in U100/U100i memory when this parameter is enabled and issuing a MDI command , or an error will occur. The MAC100 and DEF100 are special files provided by Aerotech to enhance (extend) the U100/U100i command language through the manipulation of the parameter and register sets.

6.2.25. End Of File Character Code PRM:027

PRM:027 contains the character used to denote the end of a file. The user must not include this character inside a file they wish to download to the controller since the download utility (com_100.exe) automatically attaches this character.

The default value for this parameter is **0x7E**. Change this value by entering an integer from 0 to 0xff.

6.2.26. Service Request Character Code PRM:028

This parameter contains the character that the controller uses to request immediate attention from the Host.

The default value for this parameter is **0x16**. Change this value by entering an integer from 0 to 0x1f.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.2.27. Service Acknowledge Character Code PRM:029

Parameter PRM:029 contains the character that the Host sends to the controller to acknowledge a previous service request (PRM:028).

The default value for this parameter is **0x17**. Change this value by entering an integer from 0 to 0x1f.

6.2.28. Task 1 Boot Program PRM:030

This parameter allows the user to select a “PGM:xx” file for automatic execution in Task 1 upon power up.

The default value for this parameter is **0** (no boot program). Change this value by entering a program number (1-999) that is currently in memory.



Do not include ending zeros in names (e.g., use PGM1 instead of PGM01).

6.2.29. Task 2 Boot Program PRM:031

This parameter allows the user to select a “PGM:xx” file for automatic execution in Task 2 upon power up.

The default value established for this parameter is **0** (no boot program). Change this value by entering a program number (1-999) that is currently in memory.



Do not include ending zeros in names (e.g., use PGM1 instead of PGM01).

6.2.30. RS-232-C Xon Sent From Host PRM:032

This parameter sets the character that denotes the Xon value that is sent by the Host as a result of re-enabling the communications.

The default value for this parameter is **0x1C**. Change this value by entering an integer from 0 to 0x1f.

6.2.31. RS-232-C Xoff Sent From Host PRM:033

This parameter sets the character that denotes the Xoff value that is sent by the Host as a result of a "Queue" overload. (See also PRM:032.)

The default value for this parameter is **0x1D**. Change this value by entering an integer from 0 to 0x1f.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.2.32. Software Reset (HT Mode) PRM:034

This parameter relates to the HT Mode or com_100.exe emulation mode of operation. This character acts as a system reset while operating in the HT Mode (e.g., **<CTRL> + <D>**).

The default value for this parameter is **0x4**. Change this value by entering an integer from 0 to 0x1f.

When in the RS-232 loop mode (see PRM:003 and PRM:004) it is necessary to have a closed loop before the Host can send this character.



6.2.33. Software Reset (Host Mode) PRM:035

PRM:035 relates to the Host Mode of operation. This character serves as a system reset while operating in the Host Mode.

The default value for this parameter is **0x1E** (Ctrl-⇑Shift). Change this value by entering an integer from 0 to 0x1f.

When in the RS-232 loop mode (see PRM:003 and PRM:004) it is necessary to have a closed loop before sending the character.



6.2.34. Motion Status Total Digits, Line 1 PRM:036

This parameter indicates the minimum number of digits that the user wishes to have on line 1 (default is position feedback) of the Motion Status Screen. This includes numbers to the left and right of the decimal point, the decimal point itself, and the minus (-) sign.

The default value for this parameter is set for ten (**10**) and includes the decimal point. Change this value by entering an integer from 1 to 12.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.2.35. Motion Status Decimal Digits, Line 1 PRM:037

This parameter indicates the number of places to the right of the decimal point on line 1 (default is position feedback) of the Motion Status Screen. This parameter requires that the user also set PRM:036. To determine the number of digits available on a line for position information, take the number established in PRM:036 subtract 1 for the decimal point and 1 for the (-) sign. The value of PRM:037 determines how many decimal places to put to the right of the decimal point.

The default value for this parameter is two (**2**). Change this value by entering an integer from 0 to 12.

6.2.36. Motion Status Total Digits, Line 2 PRM:038

Parameter PRM:038 indicates the minimum number of characters that the user wishes to have on line 2 (default is velocity feedback) of the Motion Status Screen. This includes numbers to the left and right of the decimal point, the decimal point itself, and the minus (-) sign.

The default value for this parameter is set for ten (**10**) and includes the decimal point. Change this value by entering an integer from 1 to 12.

6.2.37. Motion Status Decimal Digits, Line 2 PRM:039

This parameter indicates the number of places to the right of the decimal point on line 2 (default is velocity feedback) of the Motion Status Screen. This parameter requires that the user also set PRM:038. To determine the number of places available on a line for velocity information, take the number established in PRM:038 subtract 1 for the decimal point and 1 for the (-) sign. The value of PRM:039 determines how many decimal places to put to the right of the decimal point.

The default value for this parameter is two (**2**). Change this value by entering an integer from 0 to 12.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.2.38. Motion Status Scan Type, Line 2 PRM:040

This parameter for line 2 selects what data to display on line 2 of the Motion Status Screen.

The default value for this parameter is one (**1**). Refer to Table 6-12 for alternate settings.

The first eight characters of SV:17 appear on the display, followed by the data.



Table 6-12. Settings for Motion Status Scan Type (PRM:040)

Function	PRM:040
disable (the screen displays all 20 characters of SV:17)	0
velocity feedback	**1** (default)
velocity command	2
current command	3

6.2.39. Motion Status Scan Type, Line 1 PRM:041

This parameter for line 1 selects what signal to display on line 1 of the Motion Status Screen.

The default value for this parameter is one (**1**). Refer to Table 6-13 for alternate settings.

The first eight characters of SV:16 appear on the display, followed by the data.



Table 6-13. Settings for Motion Status Scan Type (PRM:041)

Function	PRM:041
disable (the screen displays all 20 characters of SV:16)	0
position feedback	**1** (default)
position command	2
position error	3

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.3. Trajectory Generation

There are three ways that the UNIDEX 100 can generate motion: general trajectory generation, direct drive trajectory generation, and cam trajectory generation. Figure 6-3 provides an overview of the various types of trajectory methods used by the UNIDEX 100. Also illustrated are related items such as the Scaled Trajectory Filter, the Trajectory First Order Filter, and the GEAR function. The following sections describe each of these in detail.

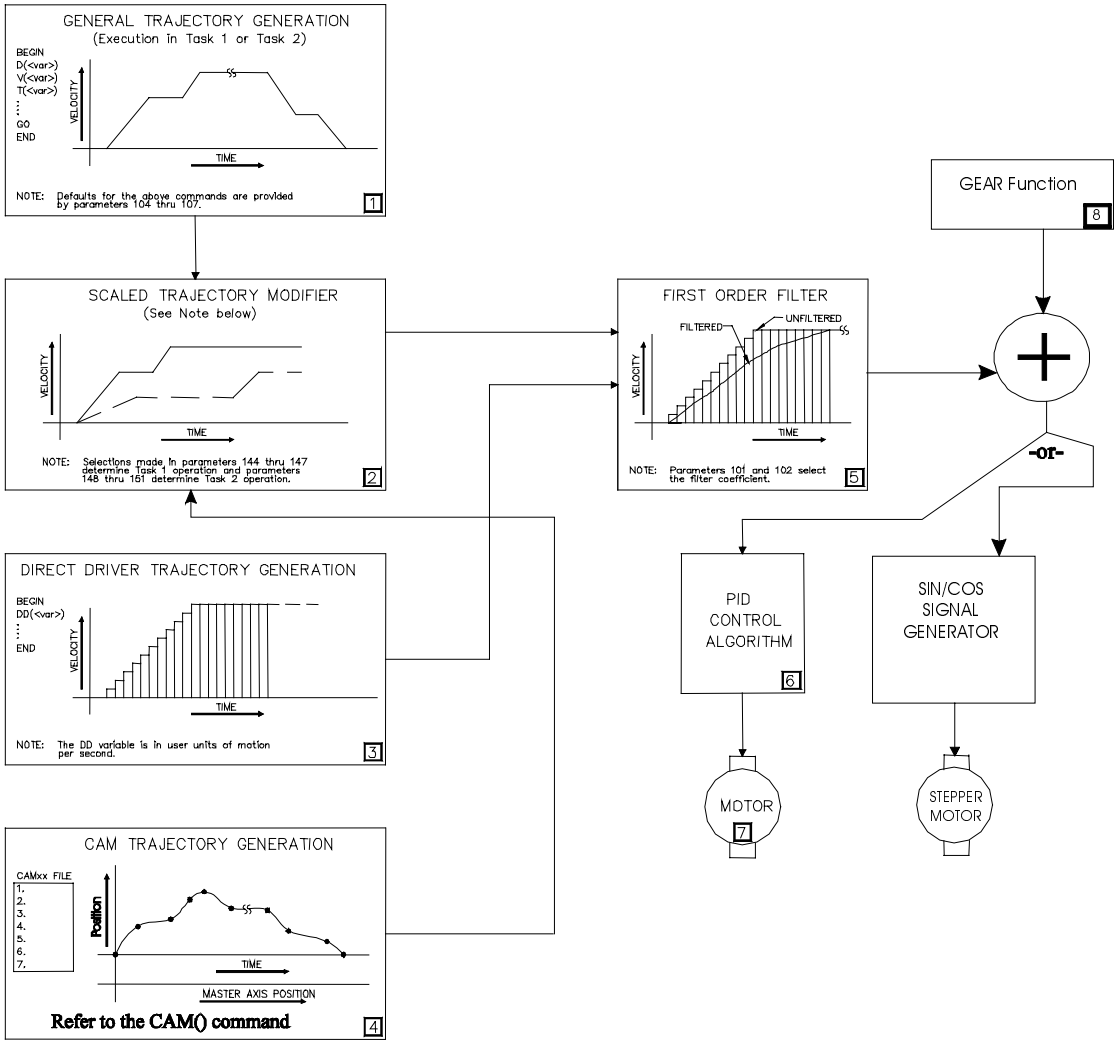


Figure 6-3. Trajectory Generation Overview

6.3.1. General Trajectory Generation

Block 1 in Figure 6-3 shows how to generate motion trajectories using the “D( )”and “GO( )” sequence of commands. Typically, General Trajectory profiles that involve this type of motion are trapezoidal and contain the following three segments: an accel, a run

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

constant, and a decel. There are four factors that control the trajectory move and these include distance (D), velocity (V), and the acceleration/deceleration (A) or time (T). Parameter “PRM:100” is the General Trajectory command scale factor.

6.3.2. Scaled Trajectory Modifier

Referring to block 2 in Figure 6-3, this block scales the General Trajectory General motion commands. The scale factor parameters (PRM:144 through PRM:151) determine how to scale the “D( )” and “GO( )” commands or “CAM( )”. Manual Feedrate Override (MFO) is a typical application that uses the Scaled Trajectory Modifier. By default, the Scaled Trajectory Modifier is set to unity scale (PRM:144 and PRM:148 set to “8”).

6.3.3. Direct Drive Trajectory Generation

Block 3 in Figure 6-3 generates motion trajectories from the Direct Drive command. The Direct Drive (DD) command outputs the motion as though it were a single position to be executed every 6.4 msec. An example of an application that uses the Direct Drive Trajectory Generator would be the joystick. This application requires an additional input (e.g., Analog to Digital) to perform motion. Parameter PRM:112 is the DD command scale factor.

6.3.4. CAM Trajectory Generation

Block 4 in Figure 6-3 permits the user to generate motion trajectories via the “CAM( )” command. Parameter “PRM:154” can be used to select scaling for either the “master axes” or “position axes”. For a complete description of the “CAM( )” command (refer to Chapter 5: Programming Commands).

6.3.5. First Order Filter

Block 5 in Figure 6-3 is a First Order Filter that filters the scaled trajectory, direct drive trajectory, and CAM trajectory generated commands. This filter smoothes out the trajectory transitions and helps to generate a type of accel/decel ramping for unramped direct drive trajectories. Parameters “PRM:101” and “PRM:102” enable and select the amount of filtering.

6.3.6. GEAR Function

The GEAR function provides electronic gearing, refer to the "GEAR()" command in Chapter 5. Any motion from the three trajectory generators (blocks 1, 3, or 4) is additive, refer to Figure 6-3.

6.3.7. PID Control Algorithm

Block 6 in Figure 6-3 is the PID loop. The three reasons for having a PID control algorithm are: to maintain the motors position, stability, and execute motion trajectories. Parameters “PRM:218” through “PRM:222” set the PID gains. Through a drive controller, the output of the PID loop controls the motor.

6.3.8. Motor

Block 7 in Figure 6-3 is the motor, which can be DC, brushless, or stepper. To close the position loop, all motors except the stepper require a position feedback device.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.4. Motion Parameters

This section discusses the parameters that pertain to programs and commands. The majority of these parameters relate directly to motion. There are 63 motion parameters, 9 that are reserved, listed in Table 6-14 and the following sections explain each in detail.

Table 6-14. Motion Parameters Summary

PRM	Description	Value Type	Default	Min Value	Max Value
100	User Units Scale Factor	float	1.0	>0	-
101	Command Segment Filter Coefficient	float	0.7	0.05	0.9999
102	Command Segment Filter On/Off	integer	0	0	1
103	Reserved	-	-	-	-
104	Default Ramp Time	float	0.5	>0	-
105	Default Distance	float	32000	-	-
106	Default Acceleration/Deceleration	float	500000.0	-	-
107	Default Velocity	float	10000.0	>0	-
108	"S" Curve Generation	integer	0	0	1
109	In Position “Wait”	integer	0	0	1
110	Lower "Modulo" Index Limit	integer	0	0	-
111	Upper "Modulo" Index Limit	integer	999	0	-
112	DD Command Scale Factor	float	1.0	>0	-
113	Reserved	-	-	-	-
114	Reserved	-	-	-	-
115	Home Switch Fast Reference	integer	0	0	1
116	Home Switch Direction	integer	2	1	2
117	Home Switch Selection	integer	6	1	6
118	Home Marker Selection	integer	1	1	4
119	Home Velocity After Power Up	float	10000.0	0.0	-
120	Home Limit Offset	float	0.0	0.0	-
121	Home Ending Offset	float	0.0	0.0	-
122	Home Marker Velocity	float	625.0	0.0	-
123	Reserved	-	-	-	-
124	Task 1 Exception Processing	integer	2	0	3

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 6-14. Motion Parameters Summary Cont'd

PRM	Description	Value Type	Default	Min. Value	Max. Value
125	Task 2 Exception Processing	integer	2	0	3
126	Reserved	-	-	-	-
127	Reserved	-	-	-	-
128	± Limit Reset Distance	float	4000.0	0.0	-
129	+ Software Limit	float	1E13	0.0	-
130	- Software Limit	float	-1E13	-	-
131	Software Limit Reference	integer	1	1	2
132	*External Interrupt Latch Source	integer	1	1	4
133	External Interrupt Mode	integer	0	0	3
134	*External Interrupt Type	integer	4	1	4
135	*External Interrupt Source	integer	2	1	2
136	Reserved	-	-	-	-
137	R/D Resolution Mode	integer	4	1	4
138	Velocity Feedback Type	integer	1	0	4
139	Position Feedback Type	integer	1	0	4
140	Analog to Digital Scale Factor	float	1.0	0.0	-
141	Analog to Digital Deadband	float	0.05	0.0	0.25
142	Digital to Analog Scale Factor	float	1.0	0.0	-
143	Reserved	-	-	-	-
144	Scaled Trajectory Task 1 Variable Type	integer	8	0	8
145	Scaled Trajectory Task 1 Gain Factor	float	0.01	>0	-
146	Scaled Trajectory Task 1 Index	integer	1000	0	-
147	Scaled Trajectory Task 1 Ramp Time	float	1.0	>0	-
148	Scaled Trajectory Task 2 Variable Type	integer	8	0	8
149	Scaled Trajectory Task 2 Gain Factor	float	0.01	>0	-
150	Scaled Trajectory Task 2 Index	integer	1000	0	-
151	Scaled Trajectory Task 2 Ramp Time	float	1.0	>0	-
152	Spline Interpolation Mode	Integer	1	1	2
153	Spline point number	integer	12	4	720
154	Spline scaling mode	integer	1	0	3
155	Spline reference mode	integer	1	1	5
156	Spline reference time	float	1.0	-	-
157	Spline reference scale	float	4000.0	-	-
158	Spline reference - limit	float	-12.0	-	-
159	Spline starting point	float	0	-	-
160	Spline reference + limit	float	+12.0	-	-
161	Spline module mode	integer	0	0	1
162	Spline output mode	integer	0	0	4
163	Spline output PV index	integer	0xE000	0x200	0xE100

The range for the maximum value is dependent on the size of the available storage.

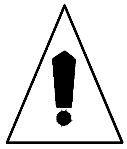


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.4.1. User Units Scale Factor PRM:100

This parameter acts as a multiplier for programmed distance. If set to one (1), the programmed distance equals the number of steps the motor moves. For example, 1000 x 1 = 1000. An application example using a stepper motor set to 8000 steps per revolution and driving a stage with a ball screw pitch of 4mm/revolution. The desired programming resolution is equal to 1mm. For example: 8000 steps/rev / by 4mm/rev is equal to 2000 steps/mm. Setting PRM:100 to 2000 with a programming distance of one D(1) results in the motor moving to 2000 steps and therefore the stage traveling 1mm.

The default value is 1.0. Change this value by entering a floating point number greater than zero.


IMPORTANT

This parameter determines the scale factor for the trajectory control. All parameters that specify distance, velocity, or acceleration (e.g., PRM:107, PRM:119, PRM:121, etc.), and A( ) use this parameter to determine the scale factor. Thus, the user must change all parameters listed above in accordance with any change in this parameter.

6.4.2. Command Segment Filter Coefficient PRM:101

This is the filter coefficient for the "First Order Filter" (refer to Figure 6-3 on page 6-22). As the coefficient decreases, the amount of filtering increases.

The default value is **0.7**. To change this, enter a floating pointing number from 0.05 (max. filtering) to 0.9999 (min. filtering).

6.4.3. Command Segment Filter On/Off PRM:102

This parameter enables or disables the Command Segment Filter. See also PRM:101.

The default value is zero (**0**). Refer to Table 6-15 for alternate selection.

Table 6-15. Settings for Command Segment Filter On/Off (PRM:102)

PRM:102	Function
0 (default)	turn off the filter.
1	turn on the filter.

6.4.4. Command Linear Interpolation PRM:103

Setting this parameter to “1” activates a Second Order linear interpolation algorithm that operates on the Trajectory Generator. The Trajectory Generator, initiated by the commands D()..G0, CAM(), or DD()) is updated every 6.4 mSec independent of the Servo Loop Update rate setting specified by PRM:304.

By setting PRM:103 to “1”, the 6.4 mSec samples of the command trajectory are “sub-interpolated” linearly (in-position) at the servo lop update rate. This provides for smoother accel/decel of the motor. Also, the effects of velocity feedforward compensation (PRM:222) and acceleration feedforward compensation (PRM:221) are enhanced.

When interpolation is enabled, the Command Segment Filter mode (parameters PRM:101 and PRM:102) are disabled.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The default setting for this parameter is “0” (interpolation disabled). To activate this parameter after it has been changed, the controller must be reset. To initialize a reset, press <CTRL>+<D>.

6.4.5. Default Ramp Time PRM:104

PRM:104 sets the ramp time for a “D( )...GO” type command if the user failed to previously specify the ramp time (e.g., T(<var>) command).

The default value is **0.5** (sec). To change this, enter a floating point number greater than zero.

6.4.6. Default Distance PRM:105

This is the distance that a move makes if the user failed to specify a distance (e.g., executing a V(<var>) and a GO without specifying a D(<var>) command).

This parameter also selects the distance the motor will travel from its starting point during autotuning. Systems with linear positioning stages, this value should be set such that the stage does not enter a hardware or software limit. See Chapter 5: Programming Commands for the TUNE command.

The default value is **32000.0** user units. To change this, enter a floating point number greater than zero.

6.4.7. Default Acceleration/Deceleration PRM:106

This is the acceleration/deceleration value for a move that is used when the user fails to specify a value.

The default value is **50000.0** user units/sec^2. To change this, enter a floating point number greater than zero.

6.4.8. Default Velocity PRM:107

This is the velocity that is used when the user fails to specify a new value when doing motion (e.g., specifying a D(<var>) and a GO without specifying a V(<var>) command).

The default value is **10000.0** user units/sec. To change this, enter a floating point number greater than zero.

6.4.9. "S" Curve Generation PRM:108

PRM:108 enables or disables the "S" curve trajectory generation. It applies only to motion generated by the "D( ), V( ), T( ), ...GO" command sequence.

The "S" curve profiling of the velocity changes is preferred when high inertia loads are present.

The default value is zero (**0**). Refer to Table 6-16 for alternate value setting.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 6-16. Settings for "S" Curve Generation (PRM:108)

PRM:108	Function
0 (default)	trajectory change is linear.
1	trajectory change is "S" curve.

6.4.10. In Position "Wait" PRM:109

This parameter when set to a "1" forces the motion trajectory generated by the D( ), V( ), .. GO commands to wait for the "in-position" indicator before proceeding to the next command statement.

For DC or AC servo motor control, the "in-position" tolerance is set by PRM:215. For stepping motors with an associated encoder operating in "Encoder Verification Mode" (e.g., PRM:204 set to "1" or "2"), the "in-position" tolerance is set by PRM:205.

The default setting for this parameter is "**0**" (in-position wait disabled). Refer to Table 6-17 for value settings.

Table 6-17. Settings for In Position "Wait" (PRM:109)

PRM:109	Function
0 (default)	In position "wait" disabled
1	In position "wait" enabled

6.4.11. Lower "Modulo" Index Limit PRM:110

This parameter is used with the MDX command. See Chapter 5: Programming Commands for more information on the MDX command.

The default value is zero (**0**). To change this, enter an integer greater than or equal to zero.

Setting PRM:110 equal to ten and PRM:111 equal to twenty and executing the BV:1=MDX(BV:1) with BV:1 set to eight results in the BV:1 variable changing to eighteen.

6.4.12. Upper "Modulo" Index Limit PRM:111

This parameter is used with the MDX command. See Chapter 5: Programming Commands for more information on the MDX command.

The default value is **999**. To change this, enter an integer greater than or equal to zero.

Setting PRM:110 equal to ten and PRM:111 equal to twenty and executing the BV:1=MDX(BV:1) with BV:1 set to twenty-five results in the BV:1 variable changing to thirteen.

6.4.13. DD Command Scale Factor PRM:112

This parameter is the User Units Scale Factor for the DD Trajectory command (e.g., DD(<var>) command).

The default value is **1.0**. To change this, enter a floating point number greater than or equal to zero.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.4.14. Home Switch Fast Reference PRM:115

This parameter selects fast homing operation. If set to "1" any "HM( )" commands issued after an initial "HM( )" command following power up will use the default velocity of PRM:107 to slew to the home reference switch (selected by PRM:117). Otherwise, PRM:119 will be used as the velocity.

If this parameter is set to "0", parameter PRM:119 will always be used as the home velocity to the selected limit switch.

Refer to parameters PRM:116 thru PRM:122 and the description of the "HM( )" command for additional information. The default value is "**1**", fast homing enabled. Table 6-18 shows the value settings for "PRM:115".

Table 6-18. Settings for Home Switch Fast Reference (PRM:115)

PRM:115	Function
1 (default)	Fast home enabled
0	Fast home disabled

6.4.15. Home Switch Direction PRM:116

This is the initial direction that the axis takes to seek the Home Limit Switch.

The default value is two (2). Table 6-19 shows the value settings for parameter "PRM:116".

Table 6-19. Settings for Home Switch Direction (PRM:116)

PRM:116	Function
1	approach the selected Limit Switch in the + direction.
2 (default)	approach the selected Limit Switch in the - direction.

6.4.16. Home Switch Selection PRM:117

This parameter selects which switch to search for as the Home Limit Switch and also the polarity of that switch.

The default value is six (**6**). Table 6-20 shows the value settings for parameter "PRM:117".

Table 6-20. Settings for Home switch Selection (PRM:117)

PRM:117	Function
1	"+" Limit Switch going high.
2	"+" Limit Switch going low.
3	"-" Limit Switch going high.
4	"-" Limit Switch going low.
5	Home Limit Switch going high.
6 (default)	Home Limit Switch going low.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.4.17. Home Marker Selection PRM:118

This parameter selects what signal to use as the Home Marker (after finding the selected Home Limit Switch specified in PRM:117).

The default value is one (**1**). To change this, enter one of the optional four selections shown in Table 6-21.

Table 6-21. Settings for Home Marker Selection

PRM:118	Function
1(default)	home on Marker #1.
2	home on Marker #2.
3	home on the selected Limit Switch indicated by PRM:117.
4	home on zero crossing of R/D converter counter (applicable only with the R/D option board).



The direction in which these signals are selected is the opposite of that selected by PRM:116.

6.4.18. Home Velocity After Power Up PRM:119

PRM:119 is the velocity at which the axis moves when seeking the home limit following a power up or software reset. Once the axis is homed after being powered up, the axis uses the default velocity from PRM:107 along with the default ramp time set in PRM:104.



Following the first "Home" after power up, the controller remembers the home limit switch position and executes a "fast" index to this switch on subsequent "Home" commands.

The default value is **10000.0** user units/sec. To change this, enter a floating point number greater than or equal to zero.

6.4.19. Home Limit Offset PRM:120

This is the distance at which the axis moves after finding the home limit switch but before searching for the home marker (not applicable when PRM:118 equals 3).

The default value is **0.0** user units. To change this, enter a floating point number greater than or equal to zero.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.4.20. Home Ending Offset PRM:121

This is the distance at which the axis moves after the axis has reached the home marker position (via PRM:118 setting). The "Home" position is then defined at the end of this distance.

The default value is **0.0** user units. To change this, enter a floating point number greater than or equal to zero.

6.4.21. Home Marker Velocity PRM:122

Parameter PRM:122 selects the rate at which the controller searches for the marker (index) pulse, after detection of the home limit switch.

The default value for this parameter is **625** user units/sec. To change this, enter a floating point number greater than or equal to zero.

6.4.22. Task 1 Exception Processing PRM:124

This parameter selects the response of Task 1 to an active "Task 1 Mask".

The default value is two (**2**). Table 6-22 shows the value settings for parameter "PRM:124".

Table 6-22. Settings for Task 1 Exception Processing (PRM:124)

PRM:124	Function
0	no response.
1	hold command execution (and/or rate controlled feedhold of motion via PRM:147).
2 (default)	abort Task 1 operation.
3	abort motion only, and continue with next program statement.

6.4.23. Task 2 Exception Processing PRM:125

This parameter selects the response of Task 2 to an active "Task 2 Mask".

The default value is two (**2**). Table 6-23 shows the value settings for parameter "PRM:125".

Table 6-23. Settings for Task 2 Exception Processing (PRM:125)

PRM:125	Function
0	no response.
1	hold command execution (and/or rate controlled feedhold of motion via PRM:151).
2	abort Task 2 operation.
3	abort motion only, and continue with next program statement.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.4.24. +/- Limit Reset Distance PRM:128

This parameter defines the distance that is executed by the “Limit Reset" menu command to clear a software or hardware limit.

The default is **4000**. To change this, enter a floating point number greater than or equal to zero.

6.4.25. + Software Limit PRM:129

This parameter sets the Positive Software Limit threshold.

The default value is $\mathbf{1x10^{13}}$. To change this, enter a floating point number greater than or equal to zero.

6.4.26. - Software Limit PRM:130

This parameter sets the Negative Software Limit threshold.

The default value is $\mathbf{-1x10^{13}}$. To change this, enter a floating point number.

6.4.27. Software Limit Reference PRM:131

Parameter PRM:131 allows the selection of the position reference for the +/- Software Limit settings (parameters PRM:129 and PRM:130). The position reference can be either the command position or the feedback position. Refer to Table 6-24 for value settings.

Table 6-24. Settings for Software Limit Reference (PRM:131)

PRM:131	Function
1 (default)	Feedback position
2	Command position

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.4.28. External Interrupt Latch Source PRM:132

This parameter selects which Position Feedback interface to latch into REG:113 when an interrupt occurs.

The default value is one (**1**). Table 6-25 shows the value settings for parameter "PRM:132".

Table 6-25. Settings for External Interrupt Latch Source (PRM:132)

PRM:132	Function
1 (default)	Encoder 1 is latched
2	Encoder 2 is latched.
3	ENC is latched (optional).
4	R/D is latched (optional).

To activate this parameter after it has been changed, the controller must be reset. To initialize a reset hit **<CTRL>+<D>.**



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.4.29. External Interrupt Mode PRM:133

This parameter enables interrupt operation and determines how the interrupt affects the program operation. The default value is (**0**). Table 6-26 shows the value settings for parameter "PRM:133".

Table 6-26. Settings for External Interrupt Mode (PRM:133)

PRM:133	Function
0 (default)	External interrupt Off. If Task 1 and/or Task 2 are loaded with a program, execution starts immediately.
1	Load Task 2 program first, followed by Task 1 program. Task 2 is "held" until interrupt occurs. Task 1 operates normally. When interrupt occurs, "hold" Task 1 until Task 2 finishes. This mode possesses special operational properties if the first command in Task 2 is a "D( )" (or multiple "D( )", "V( )", etc.). When an interrupt occurs, Task 2 "intercepts" Task 1 motion at the point of interrupt and continues with its prescribed motion profile (followed by any other motion or non-motion commands in Task 2). After finishing Task 2, it's "re-held" until another motion interrupt occurs. At the same time, Task 1 resumes operation at the point of interruption. Task 2 Program is automatically released, when Task 1 Program ends. Register REG:113 updates with the current position upon receiving the interrupt.
2	Task 1 operates normally. If Task 2 is loaded with a program, execution is "held" until receiving an interrupt. After receiving the interrupt, Task 2 begins execution but does not affect Task 1 operation. Upon finishing Task 2 execution, Task 2 returns to its beginning and is "held" until receiving the next interrupt. Register REG:113 updates with the current position when the interrupt is received.
3	Operation identical to mode 2 (above) with the addition that the kernel is reset to the first time slot (0 mSec) on the next kernel interrupt. This mode is useful when synchronizing the UNIDEX 100 command execution to an external event. For example, forcing a motion command programmed in task 2 to start in a repeatable fashion each time there is an initiation of an external interrupt.
4	REG:113 updated as described above. No other operations are performed. Task 1/Task 2 operate normally.
5	Motion termination on interrupt using the last program segment specified in the D(), V().. GO sequence. Motion can be in any task. The motion sequence should contain two or more D() commands (with optional T(), A(), and/or V() commands in the first segment only) followed by the GO command at the end of the sequence. When interrupt occurs, the last D() command is executed to finish the command. If the interrupt occurs during the execution of the last D() command, the interrupt is ignored and the sequence finishes as if no interrupt has occurred. Restrictions: Only the first segment should contain the V() command with optional A(), and T() commands. The last segment can contain an optional A(), or T() command but not a V() command. Example: D(100000) V(8000) T(.1) D(100000) EI D(12000) . . . D(8000) T(.05) GO

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.4.30. External Interrupt Type PRM:134

This parameter chooses the conditions for which the interrupt must respond. Interrupts can be selected to respond to either a low going edge or a low level. They may also be latched or unlatched (latched interrupt requires that the interrupt be reset to enable again).

The default value for this parameter is four (**4**). Table 6-27 shows the value settings for parameter "PRM:134".

Table 6-27. Settings for External Interrupt Type (PRM:134)

PRM:134	
1	unlatching level low interrupt source.
2	unlatching edge low interrupt source.
3	latching level low interrupt source.
4 (default)	latching edge low interrupt source.

To activate this parameter after it has been changed, the controller must be reset. To initialize a reset hit **<CTRL>+<D>.**



6.4.31. External Interrupt Source PRM:135

This parameter permits the user to select the source of the interrupt. The interrupt source can be the Extension Bus Interrupt Line or the External Input Interrupt Line.

The default value is two (**2**). Table 6-28 shows the value settings for parameter "PRM:135".

Table 6-28. Settings for External Interrupt Source (PRM:135)

PRM:135	Function
"1"	the Extension Bus Interrupt Line.
"2"	the External Input Interrupt Line.

To activate this parameter after it has been changed, the controller must be reset. To initialize a reset hit **<CTRL>+<D>.**



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.4.32. R/D Resolution Mode PRM:137

This parameter allows the selection of "tracking" resolution for the R/D option board as follows. The default value is (**4**). Table 6-29 shows the value settings for parameter "PRM:137".

Table 6-29. Settings for R/D Resolution Mode (PRM:137)

PRM:137	Function
1	fixed 16 bit tracking mode (65,536 bits).
2	fixed 14 bit tracking mode (16,384 bits).
3	fixed 12 bit tracking mode (4096 bits).
4	16/14 bit dynamic tracking mode (65,536 bits, default setting).

The bit resolution shown above is relative to the Resolver or Inductosyn electrical cycle.

For operating mode "4", tracking is based on 16 bit resolution starting from 0 to approximately 15 electrical cycles per second tracking rate. When this rate is exceeded, 14 bit mode is automatically entered and its result is then automatically scaled by a factor of 4 to maintain the effect of 16 bit operation. When the tracking speed falls below 13 electrical cycles per second, the 16 bit mode is again automatically engaged. The default value is "4", 16/14 bit tracking mode.



If operating modes 1 through 3 are selected, the tracking loop of the converter must be altered in order for proper operation to be obtained.

6.4.33. Velocity Feedback Type PRM:138

This parameter selects what input to use as the velocity feedback for the PID Control Loop.

The default value for this parameter is one (**1**). To change this, enter one of the optional four selections in Table 6-30.

Table 6-30. Settings for Velocity Feedback Type (PRM:138)

PRM:138	Function
0	disable (necessary when using the external tach options).
1 (default)	velocity feedback is Encoder 1.
2	velocity feedback is Encoder 2.
3	velocity feedback is ENC (optional interface).
4	velocity feedback is R/D (optional interface).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.4.34. Position Feedback Type PRM:139

This parameter selects what input to use as the position feedback for the PID Control Loop.

The default value for this parameter is one (**1**). To change this, enter one of the remaining four selections in Table 6-31.

Table 6-31. Settings for Position Feedback Type (PRM:139)

PRM:139	Function
0	disable.
1 (default)	position feedback is Encoder 1.
2	position feedback is Encoder 2.
3	position feedback is ENC (optional interface).
4	position feedback is R/D (optional interface).

6.4.35. Analog-to-Digital Scale Factor PRM:140

This parameter provides the numerical range that the controller is to interpret as the ± full signal input to the A/D. In other words, this value represents the maximum value that the ADC command provides for a voltage of ±10 volts.

The default value for this parameter is **1.0** (e.g., ±10 volts equals ±1 when read into a variable). To change this, enter a floating point number greater than or equal to zero.

6.4.36. Analog-to-Digital Deadband PRM:141

This parameter is the percentage of the 10 volts that is considered as a 0 volt input for the ADC command. This setting is useful for interfacing a joystick or other driver when a physical detent position does not provide exactly 0 volts.

The default value for this parameter is **0.05** (5%). To change this, enter a floating point number greater than or equal to zero and less than 0.25.

6.4.37. Digital-to-Analog Scale Factor PRM:142

This parameter provides the numerical range that the controller is to interpret as the ± full signal output to the D/A. In other words, the value represents the DAC value for a 10 volt output.

The default value for this parameter is **1.0**. To change this, enter a floating point number greater than or equal to zero.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.4.38. Scaled Trajectory Task 1 Variable Type PRM:144

This parameter allows the user to select the type of variable or register to use when scaling the trajectory in Task 1. (Trajectory generated by the D( ), V( ), A( ), T( ), and GO commands only.)

The default value for this parameter is eight (**8**). To change this, enter one of the optional eight selections in Table 6-32.

Table 6-32. Scaled Trajectory Task 1 Variable Type settings (PRM:144)

PRM:144	Function
0	turn off the scaled trajectory mode.
1	use a "PV" variable to scale the trajectory.
2	use a "BV" variable to scale the trajectory.
3	use a "LV" variable to scale the trajectory.
4	use a "FV" variable to scale the trajectory.
5	use the Velocity Feedback from REG:108.
6	use the Velocity Feedback from REG:109.
7	use the A/D to scale the trajectory (set PRM:145 to .00000011921 for this mode).
8 (default)	use the "PU" mode (trajectory responds only to the "HLD" command from the "Task Window" menu) to ramp down the trajectory.

Upon selecting a variable or register for trajectory scaling (modes 1 through 7), the system obtains an internal scale factor by multiplying the selected variable or register by the value of PRM:145.

The internal scale factor is applied to the currently executing trajectory every 6.4 msec to increase or decrease its execution rate. (Refer to page 5-21)

The "PU" mode feeds the scaling to 100% so that only the "HLD" control of the "Task Window" controls the scaling. Modes 1 through 7, the "HLD" overrides the selected variable or register.

Mode zero turns off the scaled trajectory algorithm.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.4.39. Scaled Trajectory Task 1 Gain Factor PRM:145

This parameter is the scale factor to use with PRM:144 to scale the trajectory.

The scale factor for a given variable or register selected by PRM:144 and PRM:146 must be selected such that the maximum expected value of the variable or register times PRM:145 equals one (1). For example, if it is expected that 100% programmed trajectory rate should be achieved when BV:1 = 100, PRM:145 should be set to .01. To have the selection of PRM:144 equal to 7 (A/D), enter a value of .00000011921.

The default value for this parameter is **0.01**. To change this, enter a floating point number.

6.4.40. Scaled Trajectory Task 1 Index (PV/BV/LV/FV) PRM:146

This parameter selects the variable number for the variable type designated in PRM:144. This variable may be a PV, BV, LV, or an FV.

The default value for this parameter is **1000**. To change this, enter an integer (value = variable number) greater than or equal to zero.

6.4.41. Scaled Trajectory Task 1 Ramp Time PRM:147

This parameter determines the time that the axis takes to fully react to a change in the value of the selected trajectory variable or register. For example, if this parameter were set to 1 sec and BV:1 was the selected trajectory variable (PRM:144=2, PRM:146=1, and PRM:145=0.01), changing BV:1 from 100 to 0 would ramp the trajectory rate from 100% to 0% in one second. Ramp times are entered in units of seconds and are used for Task 1.

The default value for this parameter is **1.0** (sec). To change this, enter a floating point number.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.4.42. Scaled Trajectory Task 2 Variable Type PRM:148

This parameter functions like PRM:144 except that the trajectory scaling applies to motion executing in Task 2. For the options available when using this parameter refer to PRM:144.

The default value for this parameter is eight (**8**). To change this, enter one of the remaining eight selections in Table 6-33.

Table 6-33. Scaled Trajectory Task 2 Variable Type Settings (PRM:148)

PRM:148	Function
0	turn off the scaled trajectory type.
1	use a "PV" variable to scale the trajectory.
2	use a "BV" variable to scale the trajectory.
3	use a "LV" variable to scale the trajectory.
4	use a "FV" variable to scale the trajectory.
5	use the Velocity Feedback from REG:108.
6	use the Velocity Feedback from REG:109.
7	use the A/D to scale the trajectory.
8 (default)	use the "PU" mode (trajectory responds only to the "HLD" command from the "Task Window" menu) to ramp down the trajectory.

6.4.43. Scaled Trajectory Task 2 Gain Factor PRM:149

This parameter functions like PRM:145 except that the trajectory scaling applies to motion executing in Task 2.

The default value for this parameter is **0.01**. To change this, enter a floating point number.

6.4.44. Scaled Trajectory Task 2 Index (PV/BV/LV/FV) PRM:150

This parameter functions like PRM:146 except that the trajectory scaling applies to motion executing in Task 2.

The default value for this parameter is one thousand (**1000**). To change this, enter an integer (value = variable number) greater than or equal to zero.

6.4.45. Scaled Trajectory Task 2 Ramp Time PRM:151

This parameter functions like PRM:147 except that the trajectory scaling applies to motion executing in Task 2.

The default value for this parameter is **1.0** (sec). To change this, enter a floating point number.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.4.46. Spline Interpolation Mode PRM:152

This parameter allows the selection of cubic (third order) or linear (first order) for the spline mode. Table 6-34 shows the value settings.

Table 6-34. Spline Interpolation Mode Setting s PRM:152

PRM:152	Function
1 (default)	Cubic interpolation
2	Linear interpolation

6.4.47. Spline Point Number PRM:153

This parameter specifies the number of data points or number of subintervals to be processed by the cubic spline algorithm.

If the “FV:1” through “FV:<number>“ are downloaded through the Host interface using the “CAM(0)” command to initiate the spline. Then, this parameter must be manually set to the number of spline data points.

If “FV:” through “FV:<number>“ are automatically loaded with the “CAM(<file number>) command. Then, this parameter is automatically set to the number of spline data entries in the file “CAM<file number>“.

The default for this parameter is “**12**”.

This command is used during setup for the cubic spline trajectory generation command, CAM().

A detailed description on the use of cubic spline trajectory generation can be found in Appendix C.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.4.48. Spline Scaling Mode PRM:154

This parameter scales the spline's reference axis (usually denoted by the variable "x"), or the spline's position axis, the variable "f(x)".



Reference axis scaling is only applicable when PRM:155 is set to "1" (reference is time).

The default for this parameter is "**1**", refer to Table 6-35 for other available settings.

Table 6-35. Spline Scaling Mode Settings (PRM:154)

PRM:154	Function
0	Disabled. Scale is 1 PU for both reference and position axis.
1 (default)	Scale reference X axis (time) (PRM:155 equal to 1 only).
2	Scale position "f(x)" axis (PRM:155 equal to 1 through 5).
3	Scale reference X axis (PRM:155 equal to 3 through 5).

6.4.49. Spline Reference Mode PRM:155

PRM:155 selects how the reference axis will be generated. The default setting for this parameter is "**1**". Table 6-36 shows the other available settings.

Table 6-36. Spline Reference Mode Settings (PRM:155)

PRM:155	Function
1 (default)	Reference is generated by time. Use parameter PRM:156 to select length of time between beginning and ending spline data points.
2	Reference is generated by integration of the signal at the A/D input of the UNIDEX 100. Use parameters PRM:140 and PRM:141 for deadband and gain increment settings.
3	Reference is generated by the position feedback from encoder 1. Use parameter PRM:157 to select the number of machine steps between spline data points.
4	Same as mode 3 except position feedback is from encoder 2.
5	Position feedback is from the resolver or inductosyn feedback (R/D option board required).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.4.50. Spline Reference Time PRM:156

This selects the execution time in seconds between the first and last spline data point. Parameter PRM:156 is applicable only when parameter PRM:155 is set to "1".

The polarity of the number selected dictates the direction of the generated reference. The particular direction dictates whether the spline point data will be interpolated forward (positive polarity) or backward (negative polarity).

The default for this parameter is **1.0** seconds.

6.4.51. Spline Reference Scale PRM:157

This parameter selects the scale factor between spline data point subintervals. The value selected represents the number of machine steps necessary to advance the reference one (**1**) spline interval.

This parameter is applicable only when parameter PRM:155 is set to 3, 4, or 5.



Like parameter PRM:156, the polarity of the number selected dictates the direction of the generated reference.

The default for this parameter is **4000.0** steps/interval.

6.4.52. Spline Reference -Limit PRM:158

This parameter is applicable for all spline reference operating modes (see PRM:155) including the modulo operating mode (PRM:161).

The value set in this parameter specifies the negative reference limit that, if attained, would end the current spline. Meaning, the number "-10.333" indicates that the current spline would cease execution at 66% of the subinterval defined by the spline data points at reference values "-11.0" and "-10.0".

The default value for this parameter is **-12**.

6.4.53. Spline Reference Starting Point PRM:159

The value specified in this parameter determines the initial starting reference point. For example, if the value is **5.0**, then the spline will begin execution exactly at the fifth spline data point.

Like parameter PRM:158, this parameter is applicable for all modes of operation specified in PRM:155 and PRM:161.

The default value for this parameter is **0.0**.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.4.54. Spline Reference (+Limit) PRM:160

This parameter is applicable for all modes of operation specified in PRM:155 and PRM:161.

The value set in this parameter specifies the positive reference limit that, if attained, would end the current spline. Meaning, the number "+10.333" indicates that the current spline would cease execution at 33% of the subinterval defined by the spline data points at reference values +10.0 and +11.0.

The default value for this parameter is **+12.0**.

6.4.55. Spline Modulo Mode PRM:161

This mode is enabled if it is intended that the interpolation of the spline data points be repeated once the ending spline data point (or beginning spline data point if the reference is increasing negatively) is reached. The default value for this parameter is **0**. Table 6-37 shows the alternative value setting.

Table 6-37. Spline Modulo Mode Settings (PRM:161)

PRM:161	Function
0 (default)	Modulo mode disabled. The spline ends when generated reference reaches the limit setting of parameters PRM:158 or PRM:160.
1	Modulo mode enabled. The spline repeats when the generated reference reaches the magnitude of the point number set by PRM:153.

This mode is usually enabled for camming circular operations where the last spline data point is interpolated as the imaginary point before the first spline point.

The modulo mode is applicable for all scaling and reference modes (see parameters PRM:154 and PRM:155).



-Limit, Starting, and +Limit reference settings (parameters PRM:158, PRM:159, and PRM:160) are still valid when the modulo mode is enabled.

To guarantee a continuous spline, the settings for PRM:158 and PRM:160 should be kept greater than the magnitude of the point number set by parameter PRM:153. Also, parameter PRM:159 should be set to a value less than the magnitude of PRM:153.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.4.56. Spline Output Mode PRM:162

This allows output on the fly to be executed at each spline data point. Depending on the setting of this parameter, data stored in the "BV:<number>" data array can be made to appear on one of three output ports of the UNIDEX 100.

The data should be set in the "BV" array so that "<number>" corresponds to the given spline data point stored in the "FV:<number>" array. The matching "BV" data is sent to the specified output port.

The default value for this parameter is **0**. Table 6-38 shows the four other settings for PRM:162.

Table 6-38. Spline Output Mode Settings (PRM:162)

PRM:162	Function
0 (default)	Data output disabled.
1	Output data to register REG:2. Only the 8 least significant bits of the given "BV" variable are sent to the register.
2	Output data to the D/A port. Only the 8 most significant bits of the given "BV" variable are sent to this port. These 8 bits form a 2's compliment number range so that 0x80xxxx (hexadecimal) represents -10 volts and 0x7fxxxx represents +10 volts.
3	Output data to the PV:<index number> port. Where "index number" is set by parameter PRM:163. All 24 bits of the given "BV" variable are sent to the specified "PV" port.
4	Output data to variable BV:1000. The use of this mode is explained in detail in Appendix C.

6.4.57. Spline Output PV Index PRM:163

This parameter specifies the "index number" of ports "PV:<index number>" when mode 3 of parameter PRM:162 is selected.

The index number can range between 0x200 and 0xe0ff hexadecimal. The default for this parameter is **0xE000**.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.5. Drive Parameters

The drive parameters configure and tune the driver and motor used with controller. Table 6-39 list the drive parameters and the sections that follow explain each in detail.

Table 6-39. Drive Parameters Summary

PRM	Description	Value Type	Default	Min. Value	Max. Value
200	Stepper Power Stage Mode	integer	0	0	2
201	Stepper Running Current	float	1.0	0.0	20.0
202	Stepper Holding Current	float	1.0	0.0	20.0
203	Stepper Resolution	integer	80	8	1024
204	Stepper Verification Enable/Disable	integer	0	0	2
205	Stepper Slip Tolerance	float	50.0	>0	-
206	Stepper Correction Velocity	float	4000.0	>0	-
207	Stepper Damping Enable/Disable	integer	0	0	1
208	Stepper “Extended” Slip Tolerance	float	50.0	0	2
209	Stepper “Extended” Correction Velocity	float	4000.0	>0	?
210	Reserved	-	-	-	-
211	Reserved	-	-	-	-
212	Reserved	-	-	-	-
213	Servo Current Limit Trap Time	float	10.0	0.0	1000.0
214	Servo "Velocity Error" Integration Control	integer	0	0	1
215	Servo "In Position" Tolerance	integer	1	0	-
216	Servo Peak Current Limit	float	5.0	0.0	20.0
217	Servo Trap Level Current Limit	float	2.0	0.0	20.0
218	Kp Servo Loop Gain	integer	10000	0	-
219	Ki Servo Loop Gain	integer	20	0	-
220	Kpos Servo Loop Gain	float	0.9999	0.0	.99999
221	Kf Servo Loop Gain	integer	0	0	-
222	Servo Feedforward On/Off	integer	0	0	1
223	Servo Position Error Trap	float	8000.0	0.0	-
224	Servo Velocity Trap	float	400000.0	0.0	-
225	Servo Acceleration/Deceleration Trap	float	5000000.0	0.0	-
226	Reserved	-	-	-	-
227	Tune minimum frequency	float	1.0	.16	4.0
228	Tune time	float	20.0	10.0	100.0
229	Tune damping factor	float	.7	.01	1000.0
230	Tune corner frequency	float	20.0	1.0	100.0
231	Gear Mode	Integer	-	1	5
232	Gear Source	Integer	-	1	4
233	Gear Scale Factor	Float	-	-	-
234	Axis Calibration Mode	integer	0	0	2
235	Axis Calibration Table Size	integer	500	100	1000
236	Axis Calibration Increment Size	integer	1000	10	10000
237	Axis Calibration Backlash	integer	0	-100	0
238	Reserved	-	-	-	-
239	*Brushless Commutation Type	integer	2	1	4
240	*Brushless Commutation Step/Cycle	integer	1500	0	-
241	*Brushless Commutation Table Step Shift	integer	-42	-127	127

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.5.1. Stepper Power Stage Mode PRM:200

Setting this parameter to zero (0) causes the power stage to run in a conventional "PWM" 4-quadrant operating mode. Setting this parameter to one (1) causes the stepper power stage to run in a recirculating "PWM" operating mode when in position. Setting this parameter to two (2) forces recirculation at all times. The purpose of having a recirculating mode is to sustain a lower motor case temperature at motor rest while still providing the motor holding current.

The default value for this parameter is zero (**0**). To change this, enter one of the optional selections shown in Table 6-40.

Table 6-40. Stepper Power Stage Mode Settings (PRM:200)

PRM:200	Function
0 (default)	no recirculating.
1	recirculating only when in position
2	always recirculating

6.5.2. Stepper Running Current PRM:201

This parameter sets the current (torque) output to the stepper motor when motion is commanded. The maximum value set for this parameter varies depending on the jumper settings of JP13, JP18, and JP19 on the control board. These jumpers configure the hardware for a maximum current output. The microprocessor also reads these jumpers so that the controller does not try to command more current than what the hardware configuration can deliver.

The default value for this parameter is **1.0** Amps. To change this, enter a floating point number from 0 to 20.0 Amps.

6.5.3. Stepper Holding Current PRM:202

Sets the current (torque) output to the stepper motor when the motor is stationary (in position). Jumpers JP13, JP18, and JP19 set the maximum value for this parameter, and is equal to the maximum value for PRM:201. To reduce motor heating when the motor is still, set the holding current to at least one half of the running current (set in PRM:201).

The default value for this parameter is **1.0** Amps. To change this, enter a floating point number from 0 to 20.0 Amps.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.5.4. Stepper Resolution PRM:203

Determines the micro-stepping resolution of the stepper motor. Stepping motors include four windings where each winding has 50 poles, thus giving the motor and inherent minimum resolution of 200 steps per revolution. Through software, microstepping resolution is programmable from 400 to 51,250 steps per revolution. Programming of this parameter is in steps per pole and not steps per revolution. Therefore, if a user desires 8000 steps per revolution, then the number programmed into this parameter should be: 8000/50 = 160 (steps/pole).

The default value for this parameter is **80** steps/pole (or 4,000 steps/rev based on 1.8° stepper motor). To change this, enter an integer from 8 to 1,024.

6.5.5. Stepper Verification Enable/Disable PRM:204

Permits the user to enable or disable encoder feedback verifications. When verification is enabled, position command and encoder feedback are constantly compared when the motor is at rest. If the difference in feedback and command exceed a specific tolerance (set by PRM:205), the motor is automatically jogged back to within the tolerance band via a rate specified by PRM:206.

The default value for this parameter is zero (**0**). Alternate selections are shown in Table 6-41.

Table 6-41. Stepper Verification Enable/Disable Settings (PRM:204)

PRM:204	Function
0 (default)	disable stepper verification.
1	enable stepper verification.
2	enable extended mode stepper verification.

6.5.6. Stepper Slip Tolerance PRM:205

This parameter sets the maximum amount of error allowed with encoder verification enabled (see PRM:204).

The default value for this parameter is **50.0** user units. To change this, enter a floating point number.

6.5.7. Stepper Correction Velocity PRM:206

This parameter determines the velocity that a closed loop stepper uses to correct a position difference with the encoder verification enabled (see PRM:204).

The default value for this parameter is **4000.0** user units/sec. To change this, enter a floating point number.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.5.8. Stepper Damping Enable/Disable PRM:207

This parameter allows the user to enable or disable the stepper damping (reduce resonance effects). This option requires an optional Resonance Damping Board.

The default value for this parameter is zero (**0**). Table 6-42 shows the value settings for this parameter.

Table 6-42. Stepper Damping Enable/Disable Settings (PRM:207)

PRM:207	Function
0 (default)	disable the stepper damping.
1	enable the stepper damping.

To activate this parameter after it has been changed, the controller must be reset. To initialize a reset hit **<CTRL>+<D>**.



6.5.9. Stepper "Extended" Slip Tolerance PRM:208

Sets the outer slip tolerance when PRM:204 is set to a "2". This parameter has no effect on slip correction when PRM:204 is set to a "1" or "0".

When PRM:204 is set to a "2", this parameter in conjunction with PRM:205 sets up a hysteresis band where PRM:208 is the outer slip band and PRM:205 is the inner slip band.

For proper operation, PRM:208 should be set greater than or equal to PRM:205.

See PRM:209 for a more detailed explanation on the effect of this parameter in the "extended" verification mode. The default value for this parameter is **50.0** user units.

6.5.10. Stepper "Extended" Correction Velocity PRM:209

This parameter sets a fast correction velocity that is executed after the completion of a trajectory generated move (e.g., D( )... GO commands). This parameter is engaged only if PRM:204 is equal to "2".

The fast correction velocity allows for rapid convergence of the stepping motor position to the inner slip tolerance threshold after a motion command has executed (assuming error was generated during the motion command).

Once the inner threshold is attained, the normal correction velocity (PRM:206) is automatically engaged to the hold position between the inner and outer slip tolerances (PRM:205 and PRM:208, respectively).

If PRM:205 and PRM:208 are set to low values, a slower correction velocity is necessary to keep the stepping motor from oscillating.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Oscillations can occur if there is a high electrical time constant of the motor relative to the correction velocity, or if there exists a mechanical compliance between the motor and the encoder.

For proper operation PRM:209 should be set greater than or equal to PRM:206. The default value for this parameter is **4,000** user units/sec.

6.5.11. Servo Current Limit Trap Time PRM:213

This parameter in conjunction with PRM:217 (Current Limit Trap Level on Threshold) is used to generate a form of "I^2T" limit protection for the servo motor. This parameter selects the maximum allowable time in which the command current can equal or exceed the threshold set by PRM:217.

Exceeding the threshold by the given time sets BIT #15 of REG:302.

The default value for this parameter is **10.0** seconds. To change this, enter a value between 0 and 1,000.0 Amps.

6.5.12. Servo "Velocity Error" Integration Control PRM:214

This parameter controls velocity error integration in the PID Control Loop. When enabled, the system temporarily sets PRM:219 (Ki Gain) to zero when motion is being executed. This form of control helps to minimize overshoot in the presence of high inertial loads.

The default value for this parameter is zero (**0**). Table 6-43 shows value settings for this parameter.

Table 6-43. Servo "Velocity Error" Integration Control (PRM:214)

PRM:214	Function
0 (default)	disable the velocity error control.
1	enable the velocity error control.

6.5.13. Servo "In Position" Tolerance PRM:215

This parameter controls the amount of position error to recognize as the "in position" indication.

The default value for this parameter is **1** (±1 machine steps). To change this, enter an integer greater than or equal to zero.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.5.14. Servo Peak Current Limit PRM:216

This parameter controls the maximum allowable current that a PID Loop can output. This overrides the hardware current limit set up by jumpers JP13, JP18, and JP19.

The default value for this parameter is **5.0** Amps. To change this, enter a floating point number between 0 and 20.0 Amps.

6.5.15. Servo Current Limit Trap Level PRM:217

This parameter sets the threshold for PRM:213 (Time Out).

The default value for this parameter is **2.0** Amps. To change this, enter a floating point number from 0 to 20.0 Amps.

6.5.16. Kp Servo Loop Gain PRM:218

This parameter provides proportional gain adjustment to the velocity error mode of the PID Control Loop, refer to Figure 6-3.

The default value for this parameter is **10000**. To change this, enter an integer greater than or equal to zero.

6.5.17. Ki Servo Loop Gain PRM:219

This parameter provides integral gain adjustment to the velocity error mode of the PID Control Loop, refer to Figure 6-3.

The default value for this parameter is **20**. To change this, enter an integer greater than or equal to zero.

6.5.18. Kpos Servo Loop Gain PRM:220

This parameter provides proportional gain adjustment to the position error mode of the PID Control Loop, refer to Figure 6-3.

The default value for this parameter is **0.9999**. To change this, enter a floating point number from 0 to .99999.

6.5.19. Kf Servo Loop Gain PRM:221

This parameter provides proportional gain adjustment for acceleration feedforward compensation to the PID Control Loop, refer to Figure 6-3. However, the acceleration feedforward gain can only be engaged if velocity feedforward compensation is On (see PRM:222).

The default value for this parameter is zero (**0**). To change this, enter an integer greater than or equal to zero.

6.5.20. Servo Feedforward On/Off PRM:222

This parameter enables or disables the velocity feedforward compensation to the PID Control Loop, refer to Figure 6-4.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The default value for this parameter is zero (**0**). Table 6-44 shows the value settings for this parameter.

Table 6-44. Servo Feedforward On/Off Settings (PRM:222)

PRM:222	Function
0 (default)	disable the feedforward.
1	enable the feedforward.

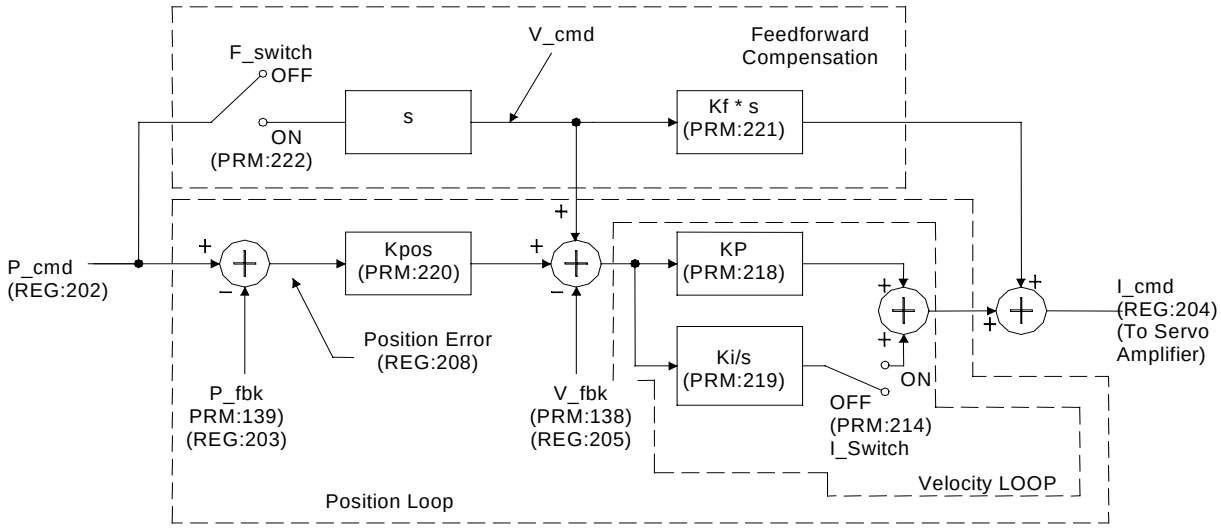


Figure 6-4. U100/U100i PID Loop

6.5.21. Servo Position Error Trap PRM:223

This parameter sets the maximum position error that a system can attain before causing a trap condition. To monitor the position error look at REG:208. If a trap occurred, the system sets BIT#12 of REG:302.

The default value for this parameter is **8000** user units. To change this, enter a floating point number greater than or equal to zero.

6.5.22. Servo Velocity Trap PRM:224

This parameter sets the maximum feedback velocity that a system can attain before causing a trap condition. The user can view the feedback velocity through REG:205. If a trap occurs, the system sets BIT#13 of REG:302.

The default value for this parameter is **400000.0** user units/sec. To change this, enter a floating point number greater than or equal to zero.

6.5.23. Servo Acceleration/Deceleration Trap PRM:225

This parameter sets the maximum amount of feedback acceleration or deceleration allowed before causing a trap. REG:210 permits the user to view the feedback acceleration, refer to Figure 6-3.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The default value for this parameter is **500000.0** user units/sec^2. To change this, enter a floating point number greater than or equal to zero.

6.5.24. Tune Minimum Frequency PRM:227

This parameter sets the starting frequency for the autotuning algorithm. During the tuning process this frequency is doubled and then quadrupled. For example, if the starting frequency is 1.0 Hz, the motor will be excited with frequencies of 1.0 Hz, 2.0 Hz, and 4.0 Hz.

The default value for this parameter is **1.0** Hz.

6.5.25. Tune Time PRM:228

This parameter sets the time for the autotuning algorithm to determine the controller gains. The default value for this parameter **20.0** seconds.

6.5.26. Tune Damping Factor PRM:229

This parameter sets the damping factor for the velocity loop. The value of this parameter must be greater than zero. As the damping factor approaches zero, the response of the system becomes more oscillatory. If the value of PRM:229 is less than 1, the system is under-damped and closed loop poles are complex. If PRM:229 >1, the system is over-damped and the closed loop poles are real. If PRM:229 = 1, the system is critically damped and the closed loop poles are real.

The default for this parameter is **1000.0**.

6.5.27. Tune Corner Frequency PRM:230

This parameter sets the corner frequency (in Hertz) of the velocity loop. Increasing or decreasing this parameter raises and lowers the bandwidth of the velocity loop.

The default value for this parameter is **20**.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.5.28. Gear Mode PRM:231

Allows the selection of one of three operating modes from a selection of five provided combinations. Refer to Table 6-45.

Table 6-45. Gear Mode Selections (PRM:231)

PRM:231	Function
1	SYNC
2	SYNC/INTERRUPT
3	SYNC/LIMIT
4	SYNC/INTERRUPT/LIMIT
5	ASYNC

SYNC provides ramping of the slave drive up to the master drive velocity without loss in position. ASYNC provides ramping without synchronization.

The INTERRUPT mode provides the ability to start the slave axis in synchronization with the master axis the instance an external interrupt is issued. Once up to speed, the slave and master axis should be no more that a few counts apart relative to the point of interrupt.

The LIMIT mode allows the setting of +/- position limits on the slave axis motion relative to the master axis.

6.5.29. Gear Source PRM:232

Allows the selection of the source for the master axis position command. Refer to Table 6-46 for selections.

Table 6-46. Gear Source Selections (PRM:232)

PRM:232	Function
1	Encoder 1
2	Encoder 2
3	Virtual (Freerun)
4	R/D



Number 3 selection permits freerun capability.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.5.30. Gear Scale Factor PRM:233

This parameter provides scaling of the master /slave axis between the range of real numbers +/-100.0 to 0. The polarity of the scale factor selects the direction of rotation of the slave axis relative to the master axis.

6.5.31. Axis Calibration Mode PRM:234

This parameter enables the Axis Calibration Mode of operation for the controller. Axis calibration is a special mode of operation. It allows position error data stored in a table to be evenly distributed along a path of motion, starting from a home position reference of zero through a maximum of +16777215 machine steps. If the commanded position exceeds this range or goes minus, bit #21 of REG:302 will become active.

Usually, an error table generated at the factory is provided for this mode (the DOS file name is ACAL100). This file contains position error data in tabular form starting from the home reference position.

Table 6-47 shows the value settings for this parameter.

Table 6-47. Axis Calibration Mode Settings (PRM:234)

PRM:234	Function
0	axis calibration is disabled.
1	axis calibration is enabled. The data file for this mode must reside in variable array BV:1 through BV:xx where "xx" is the number of the top of the data table set by parameter PRM:235.
2	axis calibration is enabled. The data file for this mode must reside in variable array PV:0xd000 through PV:xx where again, "xx" is the number of the top of the data table set by parameter PRM:235.

Both data array tables mentioned above can be easily loaded by using the controller utility "HOST_100.exe" provided in the UT100 software disk.

When axis calibration is enabled, scanning of the error table data occurs every 3.2 msec.

6.5.32. Axis Calibration Table Size PRM:235

This parameter sets the upper limit of the axis calibration table array (refer to PRM:234).

The default value for this parameter is **500**. To change this, enter an integer between 100 and 1,000.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.5.33. Axis Calibration Increment Size PRM:236

The position error table mentioned in the description of PRM:234 provides position error points relative to the home reference (e.g., Position command equals zero). These points are determined by the "command position - actual position" (usually measured with an interferometer) at predetermined increments. This parameter sets the size of the predetermined increment.

Thus the total length of calibrated travel equals PRM:236 * PRM:235 (in machine steps). This number cannot exceed +16777215 machine steps.

6.5.34. Axis Calibration Backlash PRM:237

This parameter sets the backlash compensation increment. A number between zero and 1000 denotes the amount of backlash applied. When the controller is sent home (via the "HM" command) the applied backlash at the zero position command is zero. As the controller is indexed in the positive direction, the applied backlash remains zero.

Only when the direction change is minus, is the backlash specified in PRM:237 added to the present command. When the indexing direction resumes in a positive direction, zero backlash is again applied.



PRM:234 must be greater than zero (e.g., Axis Calibration enabled), for backlash to be in effect.

The current calibration error compensation determined by parameters PRM:234, PRM:235 and PRM:236 as well as the current backlash compensation set by PRM:237 can be determined at any given moment by viewing REG:213.

REG:211 contains the current uncompensated position command. REG:213 contains the current uncompensated position feedback.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.5.35. Brushless Commutation Type PRM:239

This parameter sets the type of commutation for use with a brushless motor.

The default value for this parameter is two (**2**). Table 6-48 shows the value settings for this parameter.

Table 6-48. Brushless Commutation Type Settings (PRM:239)

PRM:239	Function
1	R/D has commutation.
2 (default)	hall effect has commutation. Position feedback performs "sine" wave interpolation in between the hall effect state transitions.
3	hall effect has commutation. The commutation is "six step", and does not permit interpolation of "sine" waves through the position feedback device.
4	Position feedback has commutation. Position feedback performs sine wave interpolation. Hall effect state transition performs only sine wave initialization. Use this when running a BLMUC, BLMC, or BLMH linear motor with a linear encoder.

To activate this parameter after it has been changed, the controller must be reset. To initialize a reset, hit **<CTRL>+<D>**.



6.5.36. Brushless Commutation Step/Cycle PRM:240

This parameter sets the value for the number of sine interpolated commutation steps per electrical cycle of a brushless motor.

For encoder/Hall effect commutation mode (set PRM:239 set to “4”), the number loaded into PRM:240 determines the number of encoder (X4 interpolation) steps required to make the motor traverse one full electrical cycle (e.g., 360 degrees for rotary motors or one magnetic cycle for linear motors).

Table 6-49. Values for Parameter PRM:240

Motor	Encoder Resolution (After Quadrature)						MAG Cycle
	0.1µm	0.5µm	1.0µm	5.0µm	10.0µm	20.0µm	
BLMUC	320000	64000	32000	6400	3200	1600	32mm
BLMC	500000	100000	50000	10000	5000	2500	50mm
BLM	609600	121920	60960	12192	6096	3048	60.96mm (2.4 in)
BLMH	600000	120000	60000	12000	6000	3000	60mm
BLMF	600000	120000	60000	12000	6000	3000	60mm

* PRM:239 = 4 with the above values for PRM:240.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Below is the equation for calculating PRM:240 for any motor.

$$\text{PRM}:240=\frac{(\text{MAG Cycle})}{(\text{Encoder Resolution per MAG Cycle})}$$

Example for BLM and 5.0µm encoder.

$$\text{PRM}:240=\frac{(2.4)(25.4\,\frac{\text{mm}}{\text{in}})}{0.005\mu m}=\frac{60.96}{0.005}=12192$$

For the R/D commutation mode (requires the R/D board option), the converter resolution is fixed at 65536 steps (16 bits) regardless of the setting of PRM:137. In order to determine the value to be inserted into this parameter, divide 65536 by the number of electrical cycles per motor revolution (e.g., for an 8 pole motor PRM:240 = 65536/4, or 16384).

The default value for this parameter is **1000** machine steps/electric cycle (for BM series rotary motors). To change this, enter an integer, refer to Table 6-49.



To activate this parameter after it has been changed, the controller must be reset. To initialize a reset, hit **<CTRL>+<D>**.

6.5.37. Brushless Commutation Table Step Shift PRM:241

This parameter provides a method of shifting the sine interpolation cycle, set up in PRM:240, with respect to the state transitions of the hall effect inputs.

The default value for this parameter is **-42**. To change this, enter an integer from -127 to 127. The number entered through the parameter can be correlated as “degrees of shift" by using the following formula:

DEG = PRM:241*1.40625



To activate this parameter after it has been changed, the controller must be reset. To initialize a reset, hit **<CTRL>+<D>**.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.6. System Parameters

The parameters in this section allow the user to optimize the performance of the controller by selecting the features required for a particular application. These features include: exception processing configuration, update time, setup, and program buffer allocation. For a Summary of these parameters, refer to Table 6-50.

Table 6-50. System Parameters Summary

PRM	Description	Value Type	Default	Min. Value	Max. Value
300	*Port Variable (PV) Wait States	integer	15	0	15
301	*Task 1 Background Time	integer	10	1	24
302	*Task 2 Background Time	integer	10	1	24
303	*Scaled Trajectory External Port Fetch On/Off	integer	1	0	1
304	*Servo Loop Update Time	integer	4	0	6
305	*Commutation Loop Update Time	integer	2	0	4
306	*Stepper Loop Update Time	integer	1	0	4
307	*Encoder 1 Update Time	integer	3	0	8
308	*Encoder 2 Update Time	integer	0	0	8
309	*ENC Update Time	integer	0	0	8
310	*Resolver to Digital Update Time	integer	0	0	8
311	Fault Mask Update Time	integer	2	0	5
312	Control Loop Checks	integer	1	0	1
313	Combine Task 1 & Task 2 Time Slots	integer	0	0	1
314	Library Ending Offset	integer	0x200	0x200	0xDfff
315	Library Program Call Mode	integer	1	1	2
316	Library Memory Limit	integer	0x7fff	0x200	0xDfff
317	Error Status Invert Mask	integer	0x329	-	-
318	Error Status Latch Mask	integer	0x1F9B01	-	-
319	Error Status Disable Mask	integer	0x049801	-	-
320	Reserved	-	-	-	-
321	Error Status Service Request Mask	integer	0x0	-	-
322	Error Status Auxiliary Mask	integer	0x0	-	-
323	Error Status Freeze Mask	integer	0x8	-	-
324	Error Status Task 1 Mask	integer	0x019B01	-	-
325	Error Status Task 2 Mask	integer	0x029B01	-	-
326	Error Status + Direction Mask	integer	0x100	-	-
327	Error Status - Direction Mask	integer	0x100	-	-
328	Reserved	-	-	-	-
329	Reserved	-	-	-	-
330	Command Buffer Partition	integer	250	1	300
331	Extension Buffer Partition	integer	40	1	49
332	String Buffer Partition	integer	40	1	49
333	Reserved	-	-	-	-
334	Reset Delay on Motion	integer	2000	1	10000

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.6.1. Port Variable (PV) Wait States PRM:300

This parameter selects the port variable (PV:xx) read/write time on the controller Extension Bus. This parameter value is multiplied by 47 nanoseconds to determine the read/write time.

The default value for this parameter is **15**. To change this, enter an integer from 0 to 15.



To activate this parameter after it has been changed, the controller must be reset. To initialize a reset, hit **<CTRL>+<D>**.

6.6.2. Task 1 Background Time PRM:301

This parameter sets the time allotted for Task 1. This time is equal to the parameter value multiplied by 100 usec .

The default value for this parameter is **10** (1.0 msec). To change this, enter an integer from 1 to 24.



Entering a 24 gives the maximum background time of 2.4 msec for Task 1.



To activate this parameter after it has been changed, the controller must be reset. To initialize a reset, hit **<CTRL>+<D>**.

6.6.3. Task 2 Background Time PRM:302

This parameter allows the user to change the background time allotted (msec) for Task 2. This time is equal to the parameter value multiplied by 100 usec.

The default value for this parameter is **10** (1.0 msec). To change this, enter an integer from 1 to 24.



Entering a 24 provides the maximum background time of 2.4 msec for Task 2.



To activate this parameter after it has been changed, the controller must be reset. To initialize a reset, hit **<CTRL>+<D>**.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.6.4. Scaled Trajectory External Port Fetch On/Off PRM:303

This parameter enables and disables automatic updating of the buffer used to control data transmission to or from the general purpose D/A and A/D ports (e.g., the controller "DAC( ) and "ADC" commands).

The default value for this parameter is one (**1**). Table 6-51 shows the alternate setting for parameter PRM:303.

Table 6-51. Settings for Scaled Trajectory External Port Fetch (PRM:303)

PRM:303	Function
0	turn off the external port fetch update time.
1 (default)	turn on the external port fetch update time.

To activate this parameter after it has been changed, the controller must be reset. To initialize a reset, hit **<CTRL>+<D>**.



6.6.5. Servo Loop Update Time PRM:304

This parameter determines how often to update the Servo Loop.

The default value for this parameter is four (**4**). Table 6-52 shows the optional settings for parameter PRM:304.

Table 6-52. Settings for Servo Loop Update Time (PRM:304)

PRM:304	Function
0	turn Off the Servo Loop update time.
1	set the Servo Loop update time at 0.1 msec.
2	set the Servo Loop update time at 0.2 msec.
3	set the Servo Loop update time at 0.4 msec.
4 (default)	set the Servo Loop update time at 0.8 msec.
5	set the Servo Loop update time at 1.6 msec.
6	set the Servo Loop update time at 3.2 msec.

To activate this parameter after it has been changed, the controller must be reset. To initialize a reset, hit **<CTRL>+<D>**.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.6.6. Commutation Loop Update Time PRM:305

This parameter selects the update rate for the Brushless Commutation Loop.

The default value for this parameter is two (**2**). Table 6-53 shows the optional settings for parameter PRM:305.

Table 6-53. Settings for Communication Loop Update Time (PRM:305)

PRM:305	Function
0	turn OFF the commutation loop update time.
1	set the commutation loop update time at 0.2 msec.
2 (default)	set the commutation loop update time at 0.4 msec.
3	set the commutation loop update time at 0.8 msec.
4	set the commutation loop update time at 1.6 msec.



The above time settings should perfect the time setting of PRM:307 or PRM:308 if the interface used for commutation is Encoder 1 or Encoder 2.



To activate this parameter after it has been changed, the controller must be reset. To initialize a reset, hit **<CTRL>+<D>**.

6.6.7. Stepper Loop Update Time PRM:306

This parameter selects the update rate for open loop stepping motor commutation.

The default value for this parameter is one (**1**). Table 6-54 shows the optional settings for parameter PRM:306.

Table 6-54. Settings for Stepper Loop Update Time (PRM:306)

PRM:306	Function
0	turn Off the stepper loop update time.
1 (default)	set the stepper loop update time at 0.1 msec.
2	set the stepper loop update time at 0.2 msec.
3	set the stepper loop update time at 0.4 msec.
4	set the stepper loop update time at 0.8 msec.



To activate this parameter after it has been changed, the controller must be reset. To initialize a reset, hit **<CTRL>+<D>**.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.6.8. Encoder 1 Update Time PRM:307

This parameter selects the update rate for the Encoder 1 interface.

The default value for this parameter is three (**3**). Table 6-55 shows the optional settings for parameter PRM:307.

Table 6-55. Settings for Encoder 1 Update Time (PRM:307)

PRM:307	Function
0	turn Off the Encoder 1 update time.
1	set the update time of Encoder 1 to 0.1 msec.
2	set the update time of Encoder 1 to 0.2 msec.
3 (default)	set the update time of Encoder 1 to 0.4 msec.
4	set the update time of Encoder 1 to 0.8 msec.
5	set the update time of Encoder 1 to 1.6 msec.
6	set the update time of Encoder 1 to 3.2 msec.
7	set the update time of Encoder 1 to 6.4 msec.
8	set the update time of Encoder 1 to 12.8 msec.

If the interface used as feedback to the PID Loop (velocity or position) is Encoder 1, it is necessary to set PRM:307 equal to or greater than the servo update time (set in PRM:304).



To activate this parameter after it has been changed, the controller must be reset. To initialize a reset, hit **<CTRL>+<D>**.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.6.9. Encoder 2 Update Time PRM:308

This parameter selects the update rate for the Encoder 2 interface.

The default value for this parameter is zero (**0**). Table 6-56 shows the optional settings for parameter PRM:308.

Table 6-56. Settings for Encoder 2 Update Time (PRM:308)

PRM:308	Function
0 (default)	turn Off the Encoder 2 update time.
1	set the update time of Encoder 2 to 0.1 msec.
2	set the update time of Encoder 2 to 0.2 msec.
3	set the update time of Encoder 2 to 0.4 msec.
4	set the update time of Encoder 2 to 0.8 msec.
5	set the update time of Encoder 2 to 1.6 msec.
6	set the update time of Encoder 2 to 3.2 msec.
7	set the update time of Encoder 2 to 6.4 msec.
8	set the update time of Encoder 2 to 12.8 msec.



If the interface used as feedback to the PID Loop (velocity or position) is Encoder 2, it is necessary to set PRM:308 equal to or greater than the servo update time (set in PRM:304).



To activate this parameter after it has been changed, the controller must be reset. To initialize a reset, hit **<CTRL>+<D>**.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.6.10. ENC Update Time PRM:309

This parameter selects the update rate for the ENC option. (This option requires the use of a Resolution Multiplication Interface Board.)

The default value for this parameter is zero (**0**). Table 6-57 shows the optional settings for parameter PRM:309.

Table 6-57. Settings for ENC Update Time (PRM:309)

PRM:309	Function
0 (default)	turn Off the ENC update time.
1	set the ENC update time at 0.1 msec.
2	set the ENC update time at 0.2 msec.
3	set the ENC update time at 0.4 msec.
4	set the ENC update time at 0.8 msec.
5	set the ENC update time at 1.6 msec.
6	set the ENC update time at 3.2 msec.
7	set the ENC update time at 6.4 msec.
8	set the ENC update time at 12.8 msec.

To activate this parameter after it has been changed, the controller must be reset. To initialize a reset, hit **<CTRL>+<D>**.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.6.11. Resolver-to-Digital Update Time PRM:310

This parameter selects the update rate for the R/D option. (This option requires the use of a Resolver to Digital/Inductosyn to Digital Interface Board.)

The default value for this parameter is zero (**0**). Table 6-58 shows the optional settings for parameter PRM:310.

Table 6-58. Settings for Resolver to Digital Update Time (PRM:310)

PRM:310	Function
0 (default)	turn Off the R/D update time.
1	set the R/D update time at 0.1 msec.
2	set the R/D update time at 0.2 msec.
3	set the R/D update time at 0.4 msec.
4	set the R/D update time at 0.8 msec.
5	set the R/D update time at 1.6 msec.
6	set the R/D update time at 3.2 msec.
7	set the R/D update time at 6.4 msec.
8	set the R/D update time at 12.8 msec.



To activate this parameter after it has been changed, the controller must be reset. To initialize a reset, hit **<CTRL>+<D>**.

6.6.12. Fault Mask Update Time PRM:311

This sets the rate at which exception processing occurs.

The default value for this parameter is two (**2**). Table 6-59 shows the optional settings for parameter PRM:311.

Table 6-59. Settings for Fault Mask Update Time (PRM:311)

PRM:311	Function
0	turn OFF the fault mask update time.
1	set the fault mask update time at 1.6 msec.
2 (default)	set the fault mask update time at 3.2 msec.
3	set the fault mask update time at 6.4 msec.
4	set the fault mask update time at 12.8 msec.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.6.13. Control Loop Checks PRM:312

This parameter enables or disables checking of: position error, velocity, accel/decel, and current limit time-out measurements. These measurements are needed for traps in the Error Status register to set Bits 12-15.

The default value for this parameter is one (**1**). Table 6-60 shows the optional settings for parameter PRM:312.

Table 6-60. Settings for Control Loop Checks (PRM:312)

PRM:312	Function
0	disable the control loop checks.
1 (default)	enable the control loop checks.

6.6.14. Combine Task 1 and Task 2 Time Slots PRM:313

This combines the Task 2 time slots with Task 1 time slots when the setting of this parameter contains a one (1). In other words, "T-2" time slots become "T-1" time slots. Thus, it disables Task 2.

The default value for this parameter is "**0**". Table 6-61 shows the optional settings for parameter PRM:313.

Table 6-61. Settings for Combine Task 1 and Task 2 Time Slots (PRM:313)

PRM:313	Function
0 (default)	Task 1 and Task 2 time slots enabled.
1	allocates Task 2 time slots to Task 1.

6.6.15. Library Ending Offset PRM:314

This parameter maintains the pointer for the end of library used with the Library utility functions.

Library file generation and usage can only be accomplished with the MEM option (memory board option). Refer to the *UNIDEX 100 Memory Expansion Board Option Manual P/N EDU 136* for more information concerning this parameter.

6.6.16. Library Program Call Mode PRM:315

This parameter controls the way in which library programs are executed using the "RUN( )" command.

Library file generation and usage can only be accomplished with the MEM option (memory board option). See the *UNIDEX 100 Memory Expansion Board Option Manual P/N EDU 136* for more information concerning this parameter.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.6.17. Library Memory Limit PRM:316

This parameter controls the amount of memory allocated to the external memory board for library file storage.

Library file generation and usage can only be accomplished with the MEM option (memory board option). See the *UNIDEX 100 Memory Expansion Board Option Manual P/N EDU 136* for more information concerning this parameter.

6.6.18. Error Status Invert Mask PRM:317

This parameter along with PRM:318, PRM:319, and PRM:321 through PRM:327 that follows control Exception Processing for the controller .

The "Error Status Invert Mask" inverts the raw bit data entering the Exception Processing routine. This bit data is a mixture of hardware status indicators (e.g., ± hardware limit switch inputs) or software control status indicators (e.g., ± software limit and PID Control Traps such as the Position Error Trap, etc.,).

This mask along with the other masks defined by PRM:318 through PRM:327 provide bit definitions that correspond to the bit definitions defined by the Error Status register REG:302.

The Invert Mask parameter provides the ability to invert all the bits defined by REG:302. However, it should be used only to change the polarity of a "+" and "-" hardware limit switch(s), and user defined inputs 1 and 2. All other bits defined by the default settings should remain the same. For the user defined input and ± hardware limit inputs, a setting of one (1) indicates the given input, when driven to a logic low (0) will provide a logic high (1) in REG:302.

The default value for this parameter is **0x329**. To change this, enter a hexadecimal number that is an integer.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The definition of all the Error Status bits appear below. As was mentioned above, REG:302 along with PRM:317, PRM:318, PRM:319, PRM:321, and PRM:322 through PRM:327 use this bit definition. Data is entered and displayed in hexadecimal notation ("0x" prefix). Each hexadecimal digit has a range of between 0 and F. For additional information on hexadecimal digits and their binary equivalents, refer to Figure 6-5.

Table 6-62. Error Status Bit Definitions

BIT #	Definition
BIT 0	Encoder 1 Feedback Error
BIT 1	Encoder 2 Feedback Error
BIT 2	Drive Fault
BIT 3	User Defined Input 2 (default executes "fast feedhold")
BIT 4	ENC Feedback Error
BIT 5	User Defined Input 1
BIT 6	Positive Software Limit Encountered
BIT 7	Negative Software Limit Encountered
BIT 8	Positive Hardware Limit Encountered
BIT 9	Negative Hardware Limit Encountered
BIT 10	Reserved
BIT 11	R/D (Resolver/Digital) Feedback Error
BIT 12	Position Error Trap
BIT 13	Velocity Feedback Trap
BIT 14	Accel/Decel Feedback Trap
BIT 15	Current Limit Time-out Trap
BIT 16	User Defined Software Trap 1 (default stops at Task 1)
BIT 17	User Defined Software Trap 2 (default stops at Task 2)
BIT 18	User Defined Software Trap 3 (default disables power amplifier)
BIT 19	Runtime Error in Task 1
BIT 20	Run Time Error in Task 2
BIT 21	Axis Calibration Range Error
BIT 22	Reserved
BIT 23	Reserved

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

DECIMAL	BINARY	HEX	DECIMAL	BINARY	HEX	DECIMAL	BINARY	HEX	DECIMAL	BINARY	HEX	DECIMAL	BINARY	HEX
0	00000000	00	52	00110100	34	104	01101000	68	156	10011100	9C	208	11010000	D0
1	00000001	01	53	00110101	35	105	01101001	69	157	10011101	9D	209	11010001	D1
2	00000010	02	54	00110110	36	106	01101010	6A	158	10011110	9E	210	11010010	D2
3	00000011	03	55	00110111	37	107	01101011	6B	159	10011111	9F	211	11010011	D3
4	00000100	04	56	00111000	38	108	01101100	6C	160	10100000	A0	212	11010100	D4
5	00000101	05	57	00111001	39	109	01101101	6D	161	10100001	A1	213	11010101	D5
6	00000110	06	58	00111010	3A	110	01101110	6E	162	10100010	A2	214	11010110	D6
7	00000111	07	59	00111011	3B	111	01101111	6F	163	10100011	A3	215	11010111	D7
8	00001000	08	60	00111100	3C	112	01110000	70	164	10100100	A4	216	11011000	D8
9	00001001	09	61	00111101	3D	113	01110001	71	165	10100101	A5	217	11011001	D9
10	00001010	0A	62	00111110	3E	114	01110010	72	166	10100110	A6	218	11011010	DA
11	00001011	0B	63	00111111	3F	115	01110011	73	167	10100111	A7	219	11011011	DB
12	00001100	0C	64	01000000	40	116	01110100	74	168	10101000	A8	220	11011100	DC
13	00001101	0D	65	01000001	41	117	01110101	75	169	10101001	A9	221	11011101	DD
14	00001110	0E	66	01000010	42	118	01110110	76	170	10101010	AA	222	11011110	DE
15	00001111	0F	67	01000011	43	119	01110111	77	171	10101011	AB	223	11011111	DF
16	00010000	10	68	01000100	44	120	01111000	78	172	10101100	AC	224	11100000	E0
17	00010001	11	69	01000101	45	121	01111001	79	173	10101101	AD	225	11100001	E1
18	00010010	12	70	01000110	46	122	01111010	7A	174	10101110	AE	226	11100010	E2
19	00010011	13	71	01000111	47	123	01111011	7B	175	10101111	AF	227	11100011	E3
20	00010100	14	72	01001000	48	124	01111100	7C	176	10110000	B0	228	11100100	E4
21	00010101	15	73	01001001	49	125	01111101	7D	177	10110001	B1	229	11100101	E5
22	00010110	16	74	01001010	4A	126	01111110	7E	178	10110010	B2	230	11100110	E6
23	00010111	17	75	01001011	4B	127	01111111	7F	179	10110011	B3	231	11100111	E7
24	00011000	18	76	01001100	4C	128	10000000	80	180	10110100	B4	232	11101000	E8
25	00011001	19	77	01001101	4D	129	10000001	81	181	10110101	B5	233	11101001	E9
26	00011010	1A	78	01001110	4E	130	10000010	82	182	10110110	B6	234	11101010	EA
27	00011011	1B	79	01001111	4F	131	10000011	83	183	10110111	B7	235	11101011	EB
28	00011100	1C	80	01010000	50	132	10000100	84	184	10111000	B8	236	11101100	EC
29	00011101	1D	81	01010001	51	133	10000101	85	185	10111001	B9	237	11101101	ED
30	00011110	1E	82	01010010	52	134	10000110	86	186	10111010	BA	238	11101110	EE
31	00011111	1F	83	01010011	53	135	10000111	87	187	10111011	BB	239	11101111	EF
32	00100000	20	84	01010100	54	136	10001000	88	188	10111100	BC	240	11110000	F0
33	00100001	21	85	01010101	55	137	10001001	89	189	10111101	BD	241	11110001	F1
34	00100010	22	86	01010110	56	138	10001010	8A	190	10111110	BE	242	11110010	F2
35	00100011	23	87	01010111	57	139	10001011	8B	191	10111111	BF	243	11110011	F3
36	00100100	24	88	01011000	58	140	10001100	8C	192	11000000	C0	244	11110100	F4
37	00100101	25	89	01011001	59	141	10001101	8D	193	11000001	C1	245	11110101	F5
38	00100110	26	90	01011010	5A	142	10001110	8E	194	11000010	C2	246	11110110	F6
39	00100111	27	91	01011011	5B	143	10001111	8F	195	11000011	C3	247	11110111	F7
40	00101000	28	92	01011100	5C	144	10010000	90	196	11000100	C4	248	11111000	F8
41	00101001	29	93	01011101	5D	145	10010001	91	197	11000101	C5	249	11111001	F9
42	00101010	2A	94	01011110	5E	146	10010010	92	198	11000110	C6	250	11111010	FA
43	00101011	2B	95	01011111	5F	147	10010011	93	199	11000111	C7	251	11111011	FB
44	00101100	2C	96	01100000	60	148	10010100	94	200	11001000	C8	252	11111100	FC
45	00101101	2D	97	01100001	61	149	10010101	95	201	11001001	C9	253	11111101	FD
46	00101110	2E	98	01100010	62	150	10010110	96	202	11001010	CA	254	11111110	FE
47	00101111	2F	99	01100011	63	151	10010111	97	203	11001011	CB	255	11111111	FF
48	00110000	30	100	01100100	64	152	10011000	98	204	11001100	CC			
49	00110001	31	101	01100101	65	153	10011001	99	205	11001101	CD			
50	00110010	32	102	01100110	66	154	10011010	9A	206	11001110	CE			
51	00110011	33	103	01100111	67	155	10011011	9B	207	11001111	CF			

Figure 6-5. Decimal to Binary to Hexadecimal Conversion Chart

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.6.19. Error Status Latch Mask PRM:318

This parameter selects the bits to be latched to the Error Status register (REG:302) .

For example, if it is desired that user defined Input 2 (Bit #3, which has a default setting for fast feedhold) be latched when this input is activated, Bit #3 of this parameter should be set to a one (1). A setting of zero inhibits latching.

The default value for this parameter is **0x1F9B01**. To change this, enter a hexadecimal number that is an integer. The following bit data applies.

Table 6-63. Error Status Latch Mask Bit Definitions

BIT #	Description
BIT 0	Encoder 1 Feedback Error
BIT 1	Encoder 2 Feedback Error
BIT 2	Drive Fault
BIT 3	User Defined Input 2 (default executes "fast feedhold")
BIT 4	ENC Feedback Error
BIT 5	User Defined Input 1
BIT 6	Positive Software Limit Encountered
BIT 7	Negative Software Limit Encountered
BIT 8	Positive Hardware Limit Encountered
BIT 9	Negative Hardware Limit Encountered
BIT 10	Reserved
BIT 11	R/D (Resolver/Digital) Feedback Error
BIT 12	Position Error Trap
BIT 13	Velocity Feedback Trap
BIT 14	Accel/Decel Feedback Trap
BIT 15	Current Limit Time-out Trap
BIT 16	User Defined Software Trap 1 (default stops at Task 1)
BIT 17	User Defined Software Trap 2 (default stops at Task 2)
BIT 18	User Defined Software Trap 3 (default disables power amplifier)
BIT 19	Runtime Error in Task 1
BIT 20	Run Time Error in Task 2
BIT 21	Axis Calibration Range Error
BIT 22	Reserved
BIT 23	Reserved

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.6.20. Error Status Disable Mask PRM:319

This parameter permits the user to disable the power amplifier for selected status register (REG:302) conditions.

For example, if it is desired that a "+" hardware limit input is to turn Off the power stage (e.g., turn off the power amplifier or motor power), set Bit #8 of this parameter to one (1).

The default value for this parameter is **0x049801**. To change this, enter a hexadecimal number that is an integer. The following bit data applies.

Table 6-64. Error Status Disable Mask Bit Definitions

BIT #	Description
BIT 0	Encoder 1 Feedback Error
BIT 1	Encoder 2 Feedback Error
BIT 2	Drive Fault
BIT 3	User Defined Input 2 (default executes "fast feedhold")
BIT 4	ENC Feedback Error
BIT 5	User Defined Input 1
BIT 6	Positive Software Limit Encountered
BIT 7	Negative Software Limit Encountered
BIT 8	Positive Hardware Limit Encountered
BIT 9	Negative Hardware Limit Encountered
BIT 10	Reserved
BIT 11	R/D (Resolver/Digital) Feedback Error
BIT 12	Position Error Trap
BIT 13	Velocity Feedback Trap
BIT 14	Accel/Decel Feedback Trap
BIT 15	Current Limit Time-out Trap
BIT 16	User Defined Software Trap 1 (default stops at Task 1)
BIT 17	User Defined Software Trap 2 (default stops at Task 2)
BIT 18	User Defined Software Trap 3 (default disables power amplifier)
BIT 19	Runtime Error in Task 1
BIT 20	Run Time Error in Task 2
BIT 21	Axis Calibration Range Error
BIT 22	Reserved
BIT 23	Reserved

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.6.21. Error Status Service Request Mask PRM:321

This parameter allows the user to send a Service Request to the Host Controller for selected status register (REG:302) conditions .

The Service Request Code sent to the Host is 0x33 and is a kernel generated request .

For example, upon detection of the "-" Software Limit, the user may wish to trigger the Host Controller to perform some auxiliary function unrelated to the controller. To do this, he may set Bit #7 of this parameter to one (1).

The default value for this parameter is **0x0**. To change this, enter a hexadecimal number that is an integer. The following bit data applies.

Table 6-65. Error Status Service Request Mask Bit Definitions

BIT #	Description
BIT 0	Encoder 1 Feedback Error
BIT 1	Encoder 2 Feedback Error
BIT 2	Drive Fault
BIT 3	User Defined Input 2 (default executes "fast feedhold")
BIT 4	ENC Feedback Error
BIT 5	User Defined Input 1
BIT 6	Positive Software Limit Encountered
BIT 7	Negative Software Limit Encountered
BIT 8	Positive Hardware Limit Encountered
BIT 9	Negative Hardware Limit Encountered
BIT 10	Reserved
BIT 11	R/D (Resolver/Digital) Feedback Error
BIT 12	Position Error Trap
BIT 13	Velocity Feedback Trap
BIT 14	Accel/Decel Feedback Trap
BIT 15	Current Limit Time-out Trap
BIT 16	User Defined Software Trap 1 (default stops at Task 1)
BIT 17	User Defined Software Trap 2 (default stops at Task 2)
BIT 18	User Defined Software Trap 3 (default disables power amplifier)
BIT 19	Runtime Error in Task 1
BIT 20	Run Time Error in Task 2
BIT 21	Axis Calibration Range Error
BIT 22	Reserved
BIT 23	Reserved

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.6.22. Error Status Auxiliary Mask PRM:322

This parameter allows the user to set the auxiliary output for selected status register (REG:302) conditions.

For example, if the user desires to have a "brake" controlled by the auxiliary output pin, and the brake is to be applied upon activation of any software or hardware travel limits, he may set bits #6 through #9 of this parameter to a one (1).

The default value for this parameter is **0x0**. To change this, enter a hexadecimal number that is an integer. The following bit data applies.

Table 6-66. Error Status Auxiliary Mask Bit Definitions

BIT #	Description
BIT 0	Encoder 1 Feedback Error
BIT 1	Encoder 2 Feedback Error
BIT 2	Drive Fault
BIT 3	User Defined Input 2 (default executes "fast feedhold")
BIT 4	ENC Feedback Error
BIT 5	User Defined Input 1
BIT 6	Positive Software Limit Encountered
BIT 7	Negative Software Limit Encountered
BIT 8	Positive Hardware Limit Encountered
BIT 9	Negative Hardware Limit Encountered
BIT 10	Reserved
BIT 11	R/D (Resolver/Digital) Feedback Error
BIT 12	Position Error Trap
BIT 13	Velocity Feedback Trap
BIT 14	Accel/Decel Feedback Trap
BIT 15	Current Limit Time-out Trap
BIT 16	User Defined Software Trap 1 (default stops at Task 1)
BIT 17	User Defined Software Trap 2 (default stops at Task 2)
BIT 18	User Defined Software Trap 3 (default disables power amplifier)
BIT 19	Runtime Error in Task 1
BIT 20	Run Time Error in Task 2
BIT 21	Axis Calibration Range Error
BIT 22	Reserved
BIT 23	Reserved

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.6.23. Error Status Freeze Mask PRM:323

This parameter allows the user to execute a fast feedhold for selected status register (REG:302) conditions.

A fast feedhold (freeze) when activated, immediately stops position command data (REG:202) from being updated by the Trajectory Generator. This effectively freezes the position command to the PID Servo Loop (or Open Loop Commutation Controller, if a stepping motor is being used).

For example, if it is desired that user defined Input 2 freeze motion, the user must set Bit #3 of this parameter equal to a one (1).

The default value for this parameter is **0x8**. To change this, enter a hexadecimal number that is an integer. The following bit data applies.

Table 6-67. Error Status Freeze Mask Bit Definitions

BIT #	Description
BIT 0	Encoder 1 Feedback Error
BIT 1	Encoder 2 Feedback Error
BIT 2	Drive Fault
BIT 3	User Defined Input 2 (default executes "fast feedhold")
BIT 4	ENC Feedback Error
BIT 5	User Defined Input 1
BIT 6	Positive Software Limit Encountered
BIT 7	Negative Software Limit Encountered
BIT 8	Positive Hardware Limit Encountered
BIT 9	Negative Hardware Limit Encountered
BIT 10	Reserved
BIT 11	R/D (Resolver/Digital) Feedback Error
BIT 12	Position Error Trap
BIT 13	Velocity Feedback Trap
BIT 14	Accel/Decel Feedback Trap
BIT 15	Current Limit Time-out Trap
BIT 16	User Defined Software Trap 1 (default stops at Task 1)
BIT 17	User Defined Software Trap 2 (default stops at Task 2)
BIT 18	User Defined Software Trap 3 (default disables power amplifier)
BIT 19	Runtime Error in Task 1
BIT 20	Run Time Error in Task 2
BIT 21	Axis Calibration Range Error
BIT 22	Reserved
BIT 23	Reserved

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.6.24. Error Status Task 1 Mask PRM:324

This parameter allows the user to initiate a Task 1 exception operation (based on PRM:124) for selected status register (REG:302) conditions.

When PRM:124 is set to two (2), this exception mask causes the program running in Task 1 to stop execution and exit. If PRM:124 is set to a one (1), this exception mask causes a feedhold (rate of feedhold programmable through PRM:147. Setting PRM:124 to zero (0) inhibits this exception.

The default value for this parameter is **0x019B01**. To change this, enter a hexadecimal number that is an integer. The following bit data applies.

Table 6-68. Error Status Task 1 Mask Bit Definitions

BIT #	Description
BIT 0	Encoder 1 Feedback Error
BIT 1	Encoder 2 Feedback Error
BIT 2	Drive Fault
BIT 3	User Defined Input 2 (default executes "fast feedhold")
BIT 4	ENC Feedback Error
BIT 5	User Defined Input 1
BIT 6	Positive Software Limit Encountered
BIT 7	Negative Software Limit Encountered
BIT 8	Positive Hardware Limit Encountered
BIT 9	Negative Hardware Limit Encountered
BIT 10	Reserved
BIT 11	R/D (Resolver/Digital) Feedback Error
BIT 12	Position Error Trap
BIT 13	Velocity Feedback Trap
BIT 14	Accel/Decel Feedback Trap
BIT 15	Current Limit Time-out Trap
BIT 16	User Defined Software Trap 1 (default stops at Task 1)
BIT 17	User Defined Software Trap 2 (default stops at Task 2)
BIT 18	User Defined Software Trap 3 (default disables power amplifier)
BIT 19	Runtime Error in Task 1
BIT 20	Run Time Error in Task 2
BIT 21	Axis Calibration Range Error
BIT 22	Reserved
BIT 23	Reserved

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.6.25. Error Status Task 2 Mask PRM:325

This parameter allows the user to initiate a Task 2 exception operation (based on PRM:125) for selected status register (REG:302) conditions.

When PRM:125 is set to two (2) this exception mask causes the program running in Task 2 to stop execution and exit. If PRM:125 is set to a one (1), this exception mask causes a feedhold (rate of feedhold programmable through PRM:151). When PRM:125 is set to zero (0), this exception becomes inhibited.

The default value for this parameter is **0x029B01**. To change this, enter a hexadecimal number that is an integer. The following bit data applies.

Table 6-69. Error Status Latch Mask Bit Definitions

BIT #	Description
BIT 0	Encoder 1 Feedback Error
BIT 1	Encoder 2 Feedback Error
BIT 2	Drive Fault
BIT 3	User Defined Input 2 (default executes "fast feedhold")
BIT 4	ENC Feedback Error
BIT 5	User Defined Input 1
BIT 6	Positive Software Limit Encountered
BIT 7	Negative Software Limit Encountered
BIT 8	Positive Hardware Limit Encountered
BIT 9	Negative Hardware Limit Encountered
BIT 10	Reserved
BIT 11	R/D (Resolver/Digital) Feedback Error
BIT 12	Position Error Trap
BIT 13	Velocity Feedback Trap
BIT 14	Accel/Decel Feedback Trap
BIT 15	Current Limit Time-out Trap
BIT 16	User Defined Software Trap 1 (default stops at Task 1)
BIT 17	User Defined Software Trap 2 (default stops at Task 2)
BIT 18	User Defined Software Trap 3 (default disables power amplifier)
BIT 19	Runtime Error in Task 1
BIT 20	Run Time Error in Task 2
BIT 21	Axis Calibration Range Error
BIT 22	Reserved
BIT 23	Reserved

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.6.26. Error Status "+" Direction Mask PRM:326

This parameter allows the user to inhibit positive ("+") direction motion for selected status register (REG:302) conditions.

When the positive direction exception is activated any motion being generated in Task 1 either by the D( ) or DD( ) commands, or CAM profiling is immediately inhibited in the "+" direction. This is different from the "freeze" exception (PRM:323) in that the inhibited motion is only in the "+" direction.

This exception when linked with a "+" Software Limit (unlatched) condition and used with the "ADC( )" and "DD( )" commands in a "WHL( )" command loop, serves as an effective way to stop motion in the "+" direction when emulating joystick control. When the limit is reached, the user simply changes the direction of the joystick (or other similar device) to move in the other direction.

The default value for this parameter is **0x100**. To change this, enter a hexadecimal number that is an integer. The following bit data applies.

Table 6-70. Error Status "+" Direction Mask Bit Definitions

BIT #	Description
BIT 0	Encoder 1 Feedback Error
BIT 1	Encoder 2 Feedback Error
BIT 2	Drive Fault
BIT 3	User Defined Input 2 (default executes "fast feedhold")
BIT 4	ENC Feedback Error
BIT 5	User Defined Input 1
BIT 6	Positive Software Limit Encountered
BIT 7	Negative Software Limit Encountered
BIT 8	Positive Hardware Limit Encountered
BIT 9	Negative Hardware Limit Encountered
BIT 10	Reserved
BIT 11	R/D (Resolver/Digital) Feedback Error
BIT 12	Position Error Trap
BIT 13	Velocity Feedback Trap
BIT 14	Accel/Decel Feedback Trap
BIT 15	Current Limit Time-out Trap
BIT 16	User Defined Software Trap 1 (default stops at Task 1)
BIT 17	User Defined Software Trap 2 (default stops at Task 2)
BIT 18	User Defined Software Trap 3 (default disables power amplifier)
BIT 19	Runtime Error in Task 1
BIT 20	Run Time Error in Task 2
BIT 21	Axis Calibration Range Error
BIT 22	Reserved
BIT 23	Reserved

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.6.27. Error Status "-" Direction Mask PRM:327

This parameter allows the user to inhibit negative ("-") direction motion for selected status register (REG:302) conditions.

The negative direction exception control works identically to the positive direction exception (PRM:326) except that motion is stopped in the "-" direction.

The default value for this parameter is **0x100**. To change this, enter a hexadecimal number that is an integer. The following bit data applies.

Table 6-71. Error Status "–" Direction Mask Bit Definitions

BIT #	Description
BIT 0	Encoder 1 Feedback Error
BIT 1	Encoder 2 Feedback Error
BIT 2	Drive Fault
BIT 3	User Defined Input 2 (default executes "fast feedhold")
BIT 4	ENC Feedback Error
BIT 5	User Defined Input 1 (default disables power amplifier)
BIT 6	Positive Software Limit Encountered
BIT 7	Negative Software Limit Encountered
BIT 8	Positive Hardware Limit Encountered
BIT 9	Negative Hardware Limit Encountered
BIT 10	Reserved
BIT 11	R/D (Resolver/Digital) Feedback Error
BIT 12	Position Error Trap
BIT 13	Velocity Feedback Trap
BIT 14	Accel/Decel Feedback Trap
BIT 15	Current Limit Time-out Trap
BIT 16	User Defined Software Trap 1 (default stops at Task 1)
BIT 17	User Defined Software Trap 2 (default stops at Task 2)
BIT 18	User Defined Software Trap 3 (default disables power amplifier)
BIT 19	Runtime Error in Task 1
BIT 20	Run Time Error in Task 2
BIT 21	Axis Calibration Range Error
BIT 22	Reserved
BIT 23	Reserved

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.6.28. Command Buffer Partition PRM:330

This parameter selects how many of the available command lines to dedicate to Task 1 use. The default value for this parameter is **250**. To change this, enter an integer from 1 to 299.

A total of 300 command statements can be compiled and run on the controller using its internal memory. With the addition of the external memory board option (MEM option), thousands of command lines can be compiled and run.

Using internal memory, the number of command line buffers allocated to Task 1 by this parameter automatically dictates the amount of buffers available for Task 2. If this parameter is set to 250, a maximum of 50 lines in a PGM file for Task 2 is allowed.

6.6.29. Extension Buffer Partition PRM:331

This parameter selects how many of the available parameter extensions to dedicate for Task 1 use.

Task 1 and Task 2 uses extension buffers whenever one or more register (REG:xx) or parameter (PRM:xx) variable is present in a single command line of a PGM file. Each command line uses one extension buffer containing a register and/or parameter variable(s). There is a maximum of 50 extension buffers.

This parameter functions similar to PRM:330 in that the number of extension buffers allocated for Task 1, less 50, leaves the number of extension buffers available for Task 2.

The default value for this parameter is **40**. To change this, enter an integer from 1 to 49.

6.6.30. String Buffer Partition PRM:332

This parameter selects how many of the available string buffers to dedicate for Task 1 use. Task 1 and Task 2 uses string buffers when string quotations ("...") are detected. For example, SV:1="this is string 1" or PM("this is string"). There is a total of 50 string buffers.

This parameter is similar to PRM:330 in that the number of string buffers allocated for Task 1, less 50, leaves the number of extension buffers available for Task 2.

The default value for this parameter is **40**. To change this, enter an integer from 1 to 49.

6.6.31. Reset Delay on Motion PRM:334

This parameter sets the amount of time that the PID Loop is to delay in resetting after the issuing of a system reset (<Ctrl> - <D> or <Ctrl> - <Shift> for HT or Host modes).

The time is specified in msec. A delay in resetting of the PID loop when controlling servo motors is necessary so that the position loop is not reset while the motor is spinning to a stop after the issue of a system reset.

The default value for this parameter is **2000**. To change this, enter an integer from 1 to 10000.

∇ ∇ ∇

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CHAPTER 7: REGISTERS AND VARIABLES

In This Section:

- Introduction 7-1
- Communication Registers 7-2
- Motion Registers 7-20
- Drive Registers 7-22
- System Registers 7-25
- Variables 7-31

7.1. Introduction

This chapter explains the use of the other two data types supported by the U100/U100i, registers and variables. Registers allow the user to read or write to a device (e.g., I/O) and supply information regarding the U100/U100i during operation (e.g., status). Some registers require initialization of certain parameters. There are four groups of registers:

- Communication
- Motion
- Drive
- and System

The main purpose of data types is to exchange information, perform calculations, and control. Registers and variables have assigned prefixes and require an integer index constant to access a particular element. Registers are designated as REG and variables are BV, LV, FV, and PV depending upon the type. The index constant must be preceded by a colon ":" (e.g., REG:100, BV:1).

Registers and variables can be used in mathematical statements with integer constants. For example:

BV:1=34

FV:1=REG:100*100

It is permissible to use fixed point or floating point constants in mathematical expressions. For example:

FV:300=7.412*REG:302

FV:3=3.4e^{-23}

With integer constants, including those used to index variable elements it is necessary to observe certain rules when expressing them in program statements (e.g., statements in a PGM file or an immediate command statement used in the MDI mode). These rules apply to base interpretation. The controller recognizes three base systems; they are:

- Octal (base 8)
- Decimal (base 10)
- and Hexadecimal (base 16)

The default base is "decimal". The following are examples of each.

Base 8 - "Octal" uses a leading "0" in front of the constant. For example:
PRM:310=021 is equivalent to PRM:301=17 in base 10.
Base 10 - "Decimal" is the default base and requires no leading qualifier.
Base 16 - "Hexadecimal" uses a leading "0x" in front of the constant. For example:
PV:0x100=1234 is equivalent to PV:256=1234 in base 10.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.2. Communication Registers

Communication registers pertain to communications such as RS-232 and IEEE, I/O, encoder control, and errors. The communication registers most often used are the bit input and bit output registers (REG:001 and REG:002). Refer to Table 7-1 for a summary of the communication registers.

Each register is specifically designated as one of three possible access types of either read only, write only, or read/write.

Read/write registers are the only type that can be included in program statements of "PGM" files and be used on both the left and right side of equate operators. For example, WHL(REG:2 NE BV:2) or WHL(BV:1 NE REG:2) are legal statements since REG:002 is a Read/Write register.

Each register has a basic access size of 24 bits (integer) or 48 bits (long). Some of the 24 bit registers have limited bit definitions (such as the 7-bit output register, REG:002).



The communication registers have there indexes expressed in "decimal" even though their indexes are shown with a leading "0". Be careful accessing these registers in a PGM file or MDI command mode with a leading zero. Unless it is intended that the index be expressed in "octal" (e.g., REG:020 would be entered as REG:20 if it is intended that the "20" be interpreted as a decimal number).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 7-1. Communication Registers Summary

REGISTER	DESCRIPTION	VALUE TYPE	ACCESS CAPABILITY
000	Not Used	-	-
001	Bit Input	integer	Read Only
002	Bit Output	integer	Read/Write
003	Port B Data	integer	Read Only
004	Port C Data	integer	Read Only
005	Timer Control/Status	integer	Read/Write
006	Timer Count Register	integer	Read/Write
007	External Input	integer	Read Only
008	Reserved	-	-
009	Reserved	-	-
010	RS-232-C Queue In Pointer	integer	Read Only
011	RS-232-C Queue Out Pointer	integer	Read Only
012	RS-232-C Status	integer	Read Only
013	Archive Boot Status	integer	Read Only
014	Compiler Error Line Number	integer	Read Only
015	Compiler Error	integer	Read Only
016	Task 1 Run Time Error	integer	Read Only
017	Task 2 Run Time Error	integer	Read Only
018	System Library Access Error	integer	Read Only
019	Reserved	-	-
020	Encoder 1 Output Status	integer	Read Only
021	Encoder 1 Output Latch	integer	Read Only
022	Encoder 1 Preset	integer	Write Only
023	Encoder 1 Master Control	integer	Write Only
024	Encoder 1 Input Control	integer	Write Only
025	Encoder 1 Output Control	integer	Write Only
026	Encoder 1 Quadrature Control	integer	Write Only
027	Reserved	-	-
028	Reserved	-	-
029	Reserved	-	-
030	Encoder 2 Output Status	integer	Read Only
031	Encoder 2 Output Latch	integer	Read Only
032	Encoder 2 Preset	integer	Write Only
033	Encoder 2 Master Control	integer	Write Only
034	Encoder 2 Input Control	integer	Write Only
035	Encoder 2 Output Control	integer	Write Only
036	Encoder 2 Quadrature Control	integer	Write Only
037	Reserved	-	-
038	R/D Counter Register	integer	Read Only
039	Reserved	-	-
040	IEEE-488 Command Pass Thru	integer	Read Only
041	IEEE-488 Status Register 1	integer	Read Only
042	IEEE-488 Status Register 2	integer	Read Only
043	IEEE-488 Serial Poll Status	integer	Read Only
044	IEEE-488 Address Status	integer	Read Only
045	IEEE-488 Data In	integer	Read Only
046	IEEE-488 Data Out	integer	Write Only

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.2.1. Bit Input (Read, integer)

This register is an 8 bit register that reads the 8 opto-isolated inputs on the control board (refer to Figure 7-1). A bit set to zero in this register indicates that an input is being pulled low. A bit set to one indicates a non active input (it is OFF). The "Bit to Input Pin" correspondence for register REG:001 is shown in Table 7-2.

Table 7-2. Bit Input for Register:001

Bit #	7	6	5	4	3	2	1	0
Input #	8	7	6	5	4	3	2	1

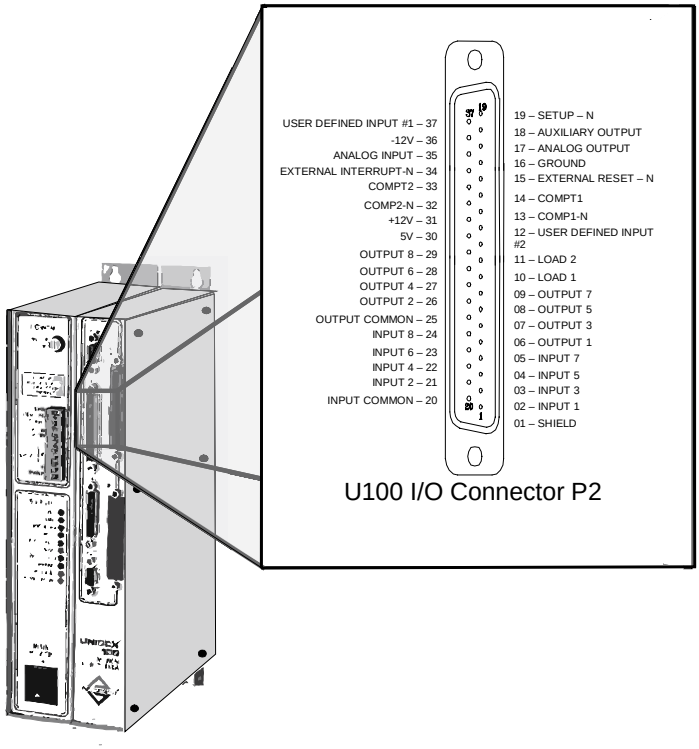


Figure 7-1. Connector P2 Bit to Input/Output for Registers

7.2.2. Bit Output (Read/Write, integer)

This register controls the 8 opto-isolated outputs on the control board (refer to Figure 7-1). Writing a zero to a bit sets the output low (active on) and writing a one to a bit sets the output high (in-active off). This is an 8 bit register with each bit assigned to one of the eight outputs. The "Bit to Output Pin" correspondence for register REG:002 is shown in Table 7-3.

Table 7-3. Bit Output for Register:002

Bit #	7	6	5	4	3	2	1	0
Output #	8	7	6	5	4	3	2	1

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.2.3. Port B Status (Read Only, integer) REG:003

This register allows the monitoring of miscellaneous status conditions present in the controller. This register does not normally need to be accessed for status since status conditions are monitored automatically through the exception processing structure. The bit definitions in Table 7-4 are shown as a servicing aid.

Table 7-4. Bit Definitions for REG:003

BIT	Definition
BIT 0	Encoder 1 fault indicator. Where: 0=fault or disconnect 1=encoder OK
BIT 1	Encoder 2 fault indicator. Where: 0=fault 1=encoder OK
BIT 2	Shutdown the amplifier output. Where: 0=enable power stage and opto-isolated outputs. 1=disable power stage and opto-isolated outputs.
BIT 3	Recirculating control pin. Where: 0=enable stepper motor recirculating mode. 1=stepper power stage conventional operation.
BIT 4	Not Used
BIT 5	User defined Input 2.
BIT 6	Setup input. Where: 0=setup mode. 1=regular operation mode. The setup pin is read only on power up, and if low, causes the controller to load the default parameters.
BIT 7	User defined Output
BIT 8	Interrupt reset. This indicates the status of the pin used by the microprocessor to reset a latched interrupt condition. Where: 0=interrupt reset, inactive. 1=interrupt reset, active.
BIT 9	Interrupt B select. Where: 0=interrupt B is being driven by the Extension Bus interrupt. 1=interrupt B is being driven by the opto-isolated external interrupt.
BIT 10	Not Used
BIT 11	Interrupt A select. Where: 0=interrupt A is coming from step motor damping module (option). 1=interrupt A is coming from a clock driven 10 KHz source.
BIT 12	ROM bank select. Where: 0=lower portion of program memory. 1=upper portion of program memory.
BIT 13	Not Used
BIT 14	Not Used

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.2.4. Port C Status (Read Only, integer) REG:004

This register monitors miscellaneous status conditions present in the controller. Like REG:003, this register does not normally need to be accessed. The applicable bit definitions are shown in Table 7-5.

Table 7-5. Bit Definitions for REG:004

BIT	Definition
BIT 0	RS-232 receive pin.
BIT 1	RS-232 transmit pin.
BIT 2	Drive fault input. Where: 0=power stage active. 1=power stage shutdown (current trip or shunt fuse opened).
BIT 3	User defined Input 1.
BIT 4	Current output range bit 0.
BIT 5	Current output range bit 1.
BIT 6	Current output range bit 2.
BIT 7	Mode type bit 0.
BIT 8	Mode type bit 1.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.2.5. Timer Control/Status (Read/Write, Integer) REG:005

Register REG:005 controls an auxiliary counter/timer unit in the U100 hardware. The applicable bit definitions are shown in Table 7-6.

Table 7-6. Bit Definitions for REG:005

BIT	Definition
BIT 0	Timer enabled.
BIT 1	Not used.
BIT 2	Invert input/output signal .
BIT 3	Mode select 0.
BIT 4	Mode select 1.
BIT 5	Mode select 2.
BIT 6	General purpose input/output.
BIT 7	Timer status.
BIT 8	Direction (input or output signal).
BIT 9	Data input.
BIT 10	Data output.
BIT 11 - 23	Not used.

Where Mode 1, 2 and 3 are defined as follows:

	Mode 2	Mode 1	Mode 0
General Purpose I/O	0	0	0
Timer Pulse	0	0	1
Timer Toggle	0	1	0
(not used)	0	1	1
Input Width	1	0	0
Input Period	1	0	1
Timer Counter	1	1	0
Event Counter	1	1	1

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.2.6. Timer Count (Read/Write, Integer) REG:006

Register REG:006 stores event information, timer rate, or input pulse/cycle width depending on the mode selected using REG:005. The counter/timer unit has the ability to count external events, as well as measure the width (time) between external events.

7.2.7. External Input (Read Only, Integer) REG:007

This register is like REG:003 and REG:004 and monitors the status conditions of the controller. With this register, however, the user can choose to monitor additional indicators such as hall effect inputs and markers.



The eight general purpose inputs can also be monitored exclusively by REG:001.

If stepping or DC Brush motors are not being controlled, the unused Hall effect inputs can be used as general purpose inputs. The pin input locations for the Hall effect inputs can be found in Chapter 9: Technical Details. These inputs are NOT opto-isolated.

The following bit data applies.

Table 7-7. Bit Definitions for REG:007

BIT #	Definition
BIT 0	Input #1
BIT 1	Input #2
BIT 2	Input #3
BIT 3	Input #4
BIT 4	Input #5
BIT 5	Input #6
BIT 6	Input #7
BIT 7	Input #8
BIT 8	Positive Limit Switch
BIT 9	Negative Limit Switch
BIT 10	Home Limit Switch
BIT 11	Marker 2 (Encoder 2 Interface)
BIT 12	Hall Effect "A"
BIT 13	Hall Effect "B"
BIT 14	Hall Effect "C"
BIT 15	Marker 1 (Encoder 1 Interface)

7.2.8. RS-232-C Queue In Pointer (Read Only, integer) REG:010

This register holds the current internal input queue pointer for the RS-232 character buffer. By comparing this pointer with the Queue Out pointer (REG:011), a determination as to the amount of unprocessed characters in the queue can be made (e.g., a difference of zero denotes no unprocessed character in the queue).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.2.9. RS-232 Queue Out Pointer (Read Only, integer) REG:011

This register holds the current internal output queue pointer for the RS-232 buffer (refer to REG:010).

7.2.10. RS-232-C Status (Read Only, integer) REG:012

This register provides RS-232 status information. The definition for each RS-232 status condition appears in Table 7-8. The user can use this register to determine if intermittent noise conditions are affecting RS-232 communications on the receiver. The three error bits (#4, #5, and #6) are "sticky". This means that a receiver error can be detected after the fault clears.

Table 7-8. RS-232-C Status Register Bit Definitions

BIT	Definition
BIT 0	Transmitter is empty.
BIT 1	Transmitter data register is empty.
BIT 2	Receive data register is full.
BIT 3	Idle Line Flag.
BIT 4	Overrun error flag (receiver).
BIT 5	Parity Error Flag (receiver).
BIT 6	Framing Error Flag (receiver).
BIT 7	Received bit #8.

7.2.11. Archive Boot Status (Read Only, Integer) REG:013

Register REG:013 allows monitoring of the boot status when backing up programs, parameters, and variables to flash ROM. The applicable bit definitions are shown in Table 7-9.

Table 7-9. Bit Definitions for REG:013

BIT	Definition
BIT 0	Set if BV:1 through BV:1000 are to be updated during power-up.
BIT 1	Set if LV:1 through LV:200 are to be updated during power-up.
BIT 2	Set if FV:1 through FV:1000 are to be updated during power-up.
BIT 3	Set if user programs (PGM, DEF, etc.) are to be updated during power-up.
BIT 4	Set if SV:1 through SV:19 are to be updated during power-up.
BIT 5	Set if PV:0x200 through PV:0xdfff are to be updated during power-up (requires MEM board option).
BIT 6 - 23	Not used.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.2.12. Compiler Error Line Number (Read Only, integer) REG:014

This register holds the line number of a LST file for a PGM that was compiled with an error. This register along with REG:015 provides interrogation of compiler errors when operating in the Host Mode.

7.2.13. Compiler Error (Read Only, integer) REG:015

This register contains the compiler error codes. Refer to Appendix D for the errors that apply to this register.

7.2.14. Task 1 Run Time Error (Read Only, integer) REG:016

This register contains the run time error codes for Task 1. Refer to Appendix D for a list of the errors that apply to this register.

7.2.15. Task 2 Run Time Error (Read Only, integer) REG:017

This register contains the run time error codes for Task 2. Refer to Appendix D for a list of the errors that apply to this register.

7.2.16. System Library Access Error (Read Only, integer) REG:018

This register contains error codes pertaining to the controller System Firmware. See Appendix D. The following errors apply.

0	No error
1	Domain Error
2	Out of Range Value (usually associated w/math function)
3	Reserved
4	Format Error (usually occurs on formatted PM commands)

7.2.17. Encoder Counters

The UNIDEX 100 has two encoder input ports. Most configurations obtain both position and velocity feedback from the primary encoder port when not using the auxiliary encoder port. However, there are many ways to use both ports. Some configurations might use one encoder port for position feedback and the other port for velocity feedback. The position feedback, in this case, comes from an encoder located elsewhere in a highly compliant mechanical system.

If one encoder can close (provide feedback for) both the position and velocity loops, the other encoder can do a wide variety of miscellaneous functions. Since each encoder port has one LOAD input and a pair of COMPARE outputs (1 mono-stable, 1 bi-stable), it is possible to address many applications with these functions. These functions include position capture, fixed distance output firing, and variable distance output firing. To implement these functions it is necessary to know the workings of the registers internal to the counter chips. This section contains information on the counters and references the register addresses for each of the two counters.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

When using the encoder to provide position and velocity feedback, the user may connect the other encoder port to either a separate encoder, or to the same one.

When using an encoder port for position and/or velocity feedback there is full dedication to this function. It is not available to the user for providing the above mentioned functions.

Do not pull the LOAD input of an encoder port used for position and velocity feedback to a logic low. Doing this destroys the integrity of the servo loop and causes erratic and possibly violent motor movement.

WARNING

Figure 7-2 is an illustration of the two-encoder counter circuits with references to the input and output connections. Input and output references for encoder number 2 (auxiliary encoder) are shown in parenthesis.

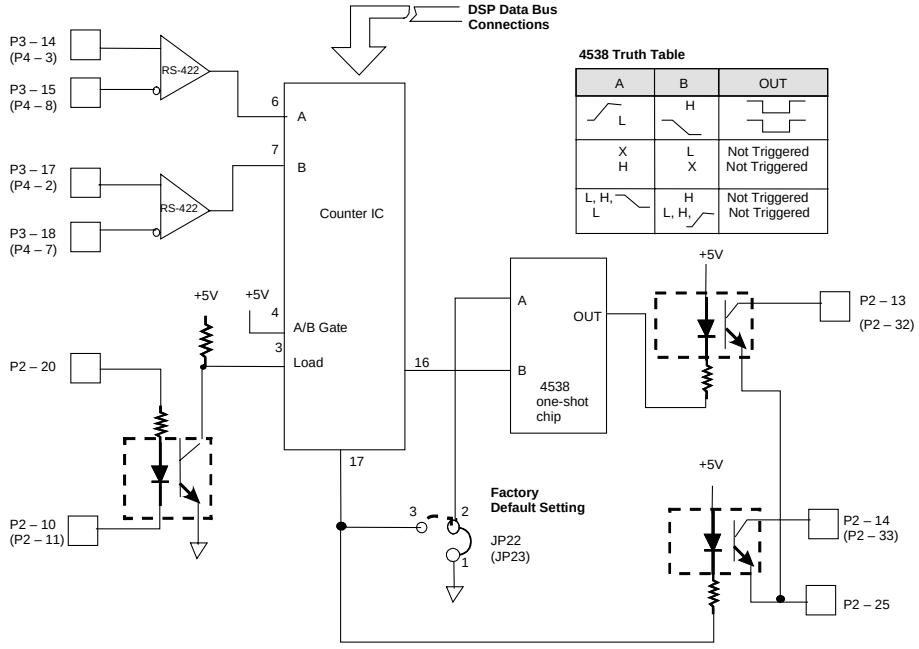


Figure 7-2. Encoder Counter Circuits (Primary and Auxiliary)

The jumper settings for JP22 (encoder #1) and JP23 (encoder #2) are factory set to utilize the “COMP” (comparator pulse) and “COMPT” (comparator toggle) functions.

If the jumpers are set to positions 2-3, the BW (borrow) and CY (carry) functions can be used to trigger the monostable outputs.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Figure 7-3 is a block diagram illustrating the internal counter functions. Like Figure 7-2, references for the auxiliary encoder are in parenthesis.

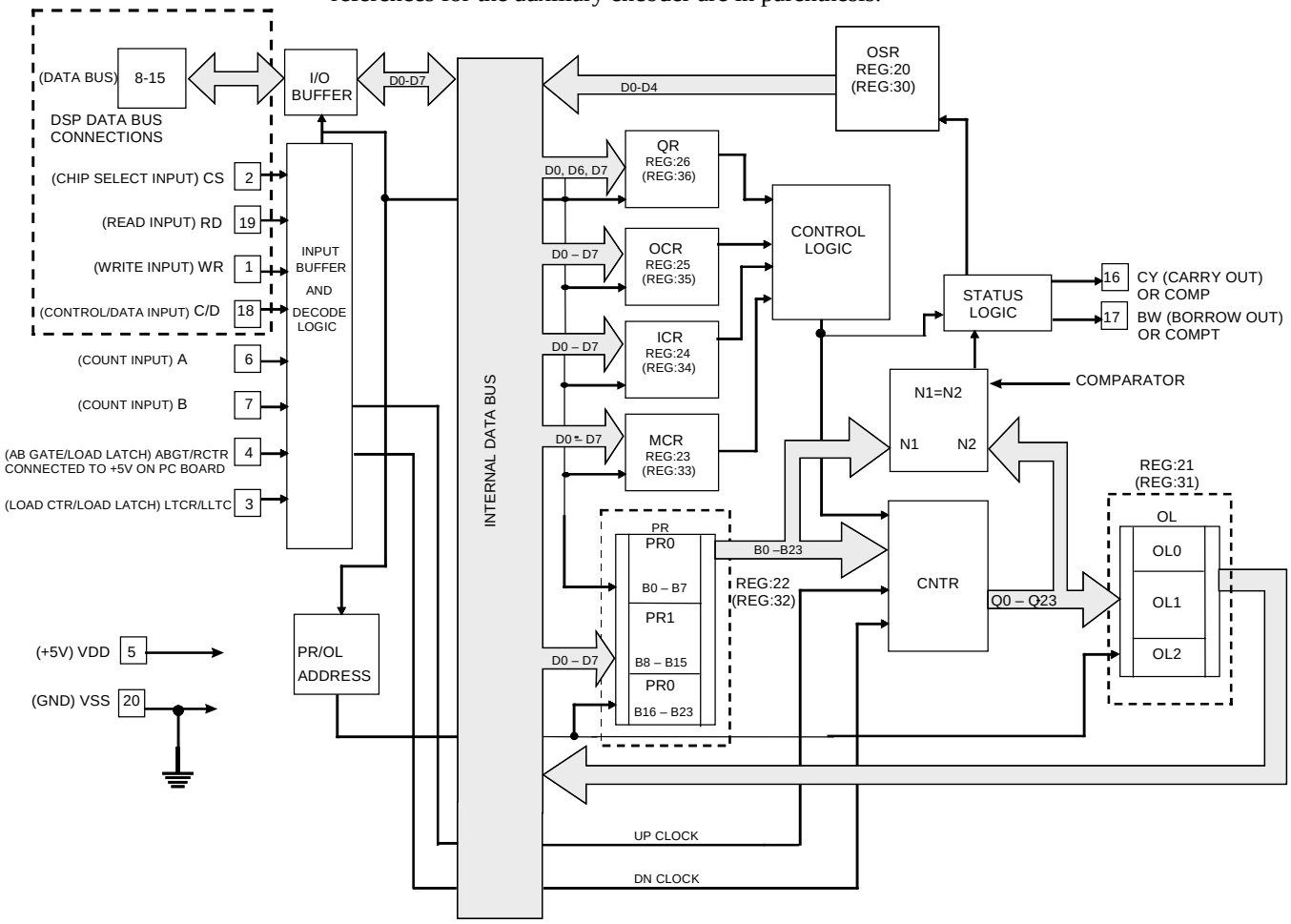


Figure 7-3. Block Diagram of Internal Counter Functions

7.2.17.1. Output Status Registers (Read Only, Integer)REG:020 / REG:030

The Output Status Register (OSR) indicates the counter (CNTR) status.

Primary Encoder - REG:020

Auxiliary Encoder - REG:030

The standard procedure for reading OSR into a variable is:

BV:1=REG:20 or BV:1=REG:30

Table 7-10 contains the bit definitions for registers REG:020 and REG:030.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 7-10. Bit Definitions for Output Status Registers

Bit	Definition
Bit 0	Borrow Toggle Flip-flop (BWT). Toggles every time CNTR under flows generating a borrow.
Bit 1	Carry Toggle Flip-flop (CYT). Toggles every time CNTR over flows generating a carry.
Bit 2	Compare Toggle Flip-flop (COMPT). Toggles every time CNTR equals PR.
Bit 3	Sign bit (SIGN) 0 = Reset when CNTR underflows 1 = Set when CNTR overflows
Bit 4	Count direction indicator in quadrature mode (UP/DOWN). 0 = Reset when counting down 1 = Set when counting up. Forced to 1 in non-quadrature mode.
Bit 5	Set to 1.
Bit 6	Set to 1.
Bit 7	Set to 1.

7.2.17.2. Output Latch Registers (Read Only, Integer) REG:021 / REG:031

The Output Latch (OL) is the output port for the (CNTR) counter.

Primary Encoder - REG:021

Auxiliary Encoder - REG:031

Bit 0 - 23 is the count stored in CNTR the last time a transfer CNTR to OL function took place.

The standard procedure for loading and reading OL is:

REG:23=0x02 or REG:33=0x02 (this transfers CNTR to OL).

BV:1=REG:21 or BV:1=REG:31 (reads 24 bit word from OL to variable).

7.2.17.3. Preset Registers (Write Only, Integer) REG:022 / REG:032

The Preset Register (PR) is the input port for the counter (CNTR). The counter is loaded with a 24 bit data word from the preset register.

Primary Encoder - REG:022

Auxiliary Encoder - REG:032

Bit 0 - 23 is the 24 bit word loaded into the preset register.

The standard procedure for writing either a variable value or a constant into the preset register is:

REG:22=BV:1 or REG:32=BV:1

or

REG:22=<constant> or REG:32=<constant>

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.2.17.4. Master Control Regs (Write Only, Int.) REG:023 / REG:033

The Master Control Register (MCR) performs register reset and load operations. Writing a "non-zero" word to the master control register does not require a follow-up write of an "all-zero" word to terminate a designated operation.

Primary Encoder - REG:023

Auxiliary Encoder - REG:033

Table 7-11. Bit Definitions for Master Control Registers

Bit	Definition
Bit 0	1 = Reset PR/OL address pointer - Transparent operation. Controlled by UNIDEX 100 operating system, automatically.
Bit 1	1 = Transfer CNTR to OL.
Bit 2	1 = Reset CNTR, BWT, and CYT. Set SIGN bit. (CNTR=0, BWT=0, CYT=0, SIGN=1).
Bit 3	1 = Transfer PR to CNTR.
Bit 4	1 = Reset COMPT (COMPT=0).
Bit 5	1 = Master reset. Reset CNTR, ICR, OCR, QR, BWT, CYT, COMPT, PR/OL address pointer. Set PR (PR=0xFFFFFF) and SIGN.
Bit 6	0
Bit 7	0

The standard procedure for writing either a variable value or a constant into the master control register is:

REG:23=BV:1 or REG:33=BV:1
or
REG:23=<constant> or REG:33 = <constant>



The control functions can be combined.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.2.17.5. Input Control Regs (Write Only, Int.) REG:024 / REG:034

The Input Control Register (ICR) initializes counter input operating modes.

Primary Encoder - REG:024

Auxiliary Encoder - REG:034

Table 7-12. Bit Definitions for Input Control Registers

Bit	Definition
Bit 0	**Encoder 1:** 0 = P3-14, P3-15 up count input. P3-17, P3-18 down count input. 1 = P3-14, P3-15 count input. P3-17, P3-18 count direction input (overridden in quadrature mode) where P3-17 equals 0, P3-18 equals 1 selects up count mode and P3-17 equals 1, P3-18 equals zero selects down count mode. **Encoder 2:** 0 = P4-3, P4-8 up count input. P4-2, P4-7 down count input. 1 = P4-3, P4-8 count input. P4-2, P4-7 count direction input (overridden in quadrature mode) where P4-2 equals 0, P4-7 equals 1 selects up count mode and P4-2 equals 1, P4-7 equals 0 selects down count mode.
Bit 1	0 = NOP 1 = Increment CNTR once
Bit 2	0 = NOP 1 = Decrement counter once.
Bit 3	0 = Disable inputs 1 = Enable inputs.
Bit 4	N/A
Bit 5	0 = Initialize LOAD1 or LOAD2 to transfer PR to CNTR when pulled to a logic low. 1 = Initialize LOAD1 or LOAD2 to transfer CNTR to OL when pulled to a logic low.
Bit 6	1
Bit 7	0

The standard procedure for writing either a variable value or a constant into the input control register is:

REG:24=BV:1 or REG:34=BV:1

or

REG:24=<constant> or REG:34=<constant>

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.2.17.6. Output Control Regs (Write Only, Int) REG:025 / REG:035

The Output Control Register (OCR) initializes the counter (CNTR) and output operating modes.

Primary Encoder - REG:025

Auxiliary Encoder - REG:035

Table 7-13. Bit Definitions for Output Control Registers

Bit	Definitions
Bit 0	0 = Binary count mode (Overridden by Bit 3=1). 1 = BCD count mode (Overridden by Bit 3=1).
Bit 1	0 = Normal count mode. 1 = Non-recycle count mode. CNTR enabled with a load or reset CNTR and disabled with the generation of a carry or borrow. In this mode no external carry or borrow is generated. CYT and/or BWT should be used as cycle completion indicators.
Bit 2	0 = Normal count mode. 1 = Divide by N count mode. CNTR is re-loaded with value in PR upon generation of a carry or borrow.
Bit 3	0 = Binary or BCD count mode (see Bit 0). 1 = 24 hour clock mode. Low byte = Sec. Middle byte = Min. High byte = Hr. (Overrides BCD/Binary modes.)
Bit 5 Bit 4 0 0 0 1 1 0 1 1	= Primary encoder - P2-13=CY-N, P2-14=BW-N Auxiliary encoder - P2-32=CY-N, P2-33=BW-N = Primary encoder - P2-13=CYT, P2-14=BWT Auxiliary encoder - P2-32=CYT, P2-33=BWT = Primary encoder - P2-13=CY, P2-14=BW Auxiliary encoder - P2-32=CY, P2-33=BW = Primary encoder - P2-13=COMP-N, P2-14=COMPT Auxiliary encoder - P2-32=COMP-N, P2-33=COMPT
Bit 6	0
Bit 7	1

The following is a list of meanings for each of the abbreviations in Table 7-13.

CY-N	= Complemented carry out (active low).
CY	= True carry out (active high).
CYT	= Carry toggle flip-flop out.
COMP-N	= Comparator out (active low).
BW-N	= Complemented borrow out (active low).
BW	= True borrow out (active high).
BWT	= Borrow toggle flip-flop out.
COMPT	= Comparator out (active high).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.2.17.7. Quadrature Regs (Write Only, Int) REG:026 / REG:036

The Quadrature Register (QR) selects the quadrature count mode.

Primary Encoder -REG:026

Auxiliary Encoder -REG:036

Table 7-14. Bit Definitions for Quadrature Registers

Bit	Definitions
Bit 1 Bit 0 0 0 0 1 1 0 1 1	 = Disable quadrature mode. = Enable x1 quadrature mode. = Enable x2 quadrature mode. = Enable x4 quadrature mode.
Bit 2	Don't care -or- x
Bit 3	Don't care -or- x
Bit 4	Don't care -or- x
Bit 5	Don't care -or- x
Bit 6	1
Bit 7	1

For programming examples that show the fundamentals necessary to use encoder counter chips, refer to Chapter 8: Sample Programs.



7.2.18. R/D Counter Register (Read Only, integer) REG:038

This register allows direct access to the R/D (Resolver to Digital) counter, when the R/D option board is installed.

The counter is aligned to the upper 24 bits of the data register. For example, if PRM:137 is set to 4 or 1 (16 bit tracking), the valid range of this register is 000000 through ffff00. If 12 bit tracking mode is selected (e.g., PRM:137=3), the valid range is 00000 through fff000.

7.2.19. IEEE-488 Cmd Pass Thru (Read Only, integer) REG:040

This register allows the controller to read the DIO lines in cases of undefined commands, secondary addresses, or parallel poll responses. It describes register access for the IEEE-488 Option Board. It is unnecessary for the user to access this register to perform normal IEEE-488 communications (e.g., PRM:19=2).

The controller provides this register for instances in which it is desired that an executing PGM file uses these registers to "interact" to another controller (running a similar PGM file) on another IEEE-488 device.

For a more detailed description of this register refer to the data sheet for the NEC PD7210 interface chip.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.2.20. IEEE-488 Status Register 1 (Read Only, integer) REG:041

This register is the IEEE interrupt status #1. It describes register access for the IEEE-488 Option Board. It is unnecessary for the user to access this register to perform normal IEEE-488 communications (e.g., PRM:19=2).

The controller provides this register for instances in which it is desired that an executing PGM file uses these registers to "interact" to another controller (running a similar PGM file) on another IEEE-488 device.

For a more detailed description of this register refer to the data sheet for the NEC PD7210 interface chip. The following bit data applies.

Table 7-15. Bit Definitions for REG:041

BIT #	Definition
BIT 0	Data In
BIT 1	Data Out
BIT 2	Error
BIT 3	Device Clear
BIT 4	End (END or EOS message received)
BIT 5	Device Trigger
BIT 6	Address Pass Through
BIT 7	Command Pass Through

7.2.21. IEEE-488 Status Register 2 (Read Only, integer) REG:042

This register 2 is the IEEE interrupt status #2 described below. It describes register access for the IEEE-488 Option Board. It is unnecessary for the user to access this register to perform normal IEEE-488 communications (e.g., PRM:19=2).

The controller provides this register for instances where it is desired that an executing PGM file uses these registers to interact to another controller (running a similar PGM file) on another IEEE-488 device.

For a more detailed description of this register refer to the data sheet for the NEC PD7210 interface chip. The following bit data applies.

Table 7-16. Bit Definitions for REG:042

BIT #	Definition
BIT 0	Address Status Change
BIT 1	Remote Change
BIT 2	Lockout Change
BIT 3	Command Output
BIT 4	Remote/Local
BIT 5	Lockout
BIT 6	Service Request Input
BIT 7	OR of all Unmasked Interrupt Status Bits

7.2.22. IEEE-488 Serial Poll Status (Read Only, integer) REG:043

This register contains the serial poll mode status byte. It describes register access for the IEEE-488 Option Board. It is unnecessary for the user to access this register to perform normal IEEE-488 communications (e.g., PRM:19=2).
The controller provides this register for instances in which it is desired that an executing PGM file uses these registers to "interact" to another controller (running a similar PGM file) on another IEEE-488 device.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

For a more detailed description of this register refer to the data sheet for the NEC PD7210 interface chip.

7.2.23. IEEE-488 Address Status (Read Only, integer) REG:044

This register contains the address status byte information. It describes register access for the IEEE-488 Option Board. It is unnecessary for the user to access this register to perform normal IEEE-488 communications (e.g., PRM:19=2).

The controller provides this register for instances in which it is desired that an executing PGM file uses these registers to "interact" to another controller (running a similar PGM file) on another IEEE-488 device.

For a more detailed description of this register refer to the data sheet for the NEC PD7210 interface chip. The following bit data applies.

Table 7-17. Bit Definitions for REG:044

BIT #	Definition
BIT 0	Sets minor T/L address, reset equals major T/L address
BIT 1	Talker Addressed
BIT 2	Listener Addressed
BIT 3	Talker Primary Addressed State
BIT 4	Listener Primary Addressed State
BIT 5	Serial Poll Mode State
BIT 6	Data Transfer Cycle (device in CSBS)
BIT 7	Controller Active

7.2.24. IEEE-488 Data In (Read Only, integer) REG:045

This register holds the 8 bit data from the GPIB interface. It describes register access for the IEEE-488 Option Board. It is unnecessary for the user to access this register to perform normal IEEE-488 communications (e.g., PRM:19=2).

The controller provides this register for instances in which it is desired that an executing PGM file uses these registers to interact to another controller (running a similar PGM file) on another IEEE-488 device.

For a more detailed description of this register refer to the data sheet for the NEC PD7210 interface chip.

7.2.25. IEEE-488 Data Out (Write Only, integer) REG:046

This register contains the 8 bit data that is transferred to the GPIB interface. It describes register access for the IEEE-488 Option Board. It is unnecessary for the user to access this register to perform normal IEEE-488 communications (e.g., PRM:19=2).

The controller provides this register for instances in which it is desired that an executing PGM file uses these registers to interact to another controller (running a similar PGM file) on another IEEE-488 device.

For a more detailed description of this register refer to the data sheet for the NEC PD7210 interface chip.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.3. Motion Registers

The Motion registers pertain to motion (position and velocity) for each of the controller feedback devices. Included is the external interrupt latch that allows the user to read the latch position of a selected device for an external interrupt.

Registers REG:102 through REG:105 and registers REG:108 through REG:111 update automatically at the rate indicated by PRM:307 through PRM:310 respectively. In the case of the velocity registers REG:108 through REG:111, the rate in counts per sample is the time increment set by these parameters.

The interrupt register REG:113 operates in accordance with settings provided by parameters PRM:132 through PRM:135.

Refer to Table 7-18 for a summary of the motion registers.

Table 7-18. Motion Registers Summary

REGISTER	DESCRIPTION	VALUE TYPE	ACCESS CAPABILITY
100	Reserved	-	-
101	Reserved	-	-
102	Encoder 1 Position	long	Read/Write
103	Encoder 2 Position	long	Read/Write
104	R/D Position	long	Read/Write
105	ENC Position	long	Read/Write
106	Reserved	-	-
107	Reserved	-	-
108	Encoder 1 Velocity	integer	Read Only
109	Encoder 2 Velocity	integer	Read Only
110	R/D Velocity	integer	Read Only
111	ENC Velocity	integer	Read Only
112	Reserved	-	-
113	External Interrupt Position Latch	long	Read Only



All registers are in units of "steps" or "steps per sample" accordingly.

7.3.1. Encoder 1 Position (Read/Write, long) REG:102

This register is the position, in steps, of encoder #1.

7.3.2. Encoder 2 Position (Read/Write, long) REG:103

This register is the position, in steps, of encoder #2.

7.3.3. R/D Position (Read/Write, long) REG:104

This register is the position, in steps, of the R/D option.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.3.4. ENC Position (Read/Write, long) REG:105

This register is the position, in steps, of the ENC option.

7.3.5. Encoder 1 Velocity (Read Only, integer) REG:108

This register is the present velocity of encoder #1 (steps/sample).

7.3.6. Encoder 2 Velocity (Read Only, integer) REG:109

This register is the present velocity of encoder #2 (steps/sample).

7.3.7. R/D Velocity (Read Only, integer) REG:110

This register is the present R/D velocity (steps/sample).

7.3.8. ENC Velocity (Read Only, integer) REG:111

This register is the present ENC velocity (steps/sample).

7.3.9. External Interrupt Position Latch (Read Only, long) REG:113

This register latches the position (in steps) for an external interrupt.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.4. Drive Registers

These registers provide general information on position, velocity, accel/decel, and current command relevant to the control of the motor (DC brush, AC brushless, or stepper).

REG:203 and REG:205 reflects the position and velocity feedback source as selected by parameters PRM:138 and PRM:139 respectively. The controller updates these registers every servo cycle (see PRM:304).



Position and velocity feedback are usually relevant only to DC brush and AC brushless motor operation.

The current command register, REG:204 reflects the magnitude of the output current where 0x7fffff equals "+" maximum and 0x80000 equals "-" maximum (2's compliment). The jumper arrangement dictates the actual maximum current setting.

The position error, velocity feedback, and accel/decel feedback registers REG:208 through REG:210 are updated every 12.8 msec regardless of the servo update rate setting of PRM:304. The Exception Processing algorithm automatically scans these registers to check for traps.

A pictorial representation of these registers relative to the PID Control Loop appears in Chapter 11: PID Loop Tuning.

It is important to know that these registers are valid for any motor type being controlled (DC brush, AC brushless, and stepper). However, feedback registers for stepper motor operation are valid only if the motor has incorporated a feedback device such as an encoder.

All registers (except REG:204) have values relative to "steps", "steps per sample" or "steps per sample (squared)" where appropriate.

Table 7-19. Drive Registers Summary

REGISTER	DESCRIPTION	VALUE TYPE	ACCESS CAPABILITY
200	Position Reset	long	Read/Write
201	Position Reset Trigger	integer	Write Only
202	Position Command	long	Read/Write
203	Position Feedback	long	Read/Write
204	Current Command	long	Read/Write
205	Instantaneous Velocity Feedback	long	Read Only
206	Averaged Velocity Command	long	Read Only
207	Kernal Timeslice Mask	Integer	Read/Write
208	Position Error	long	Read Only
209	Velocity Feedback	long	Read Only
210	Acceleration/Deceleration Feedback	long	Read Only
211	Axis Calibrated Position Command	long	Read Only
212	Axis Calibrated Position Feedback	long	Read Only
213	Axis Calibration Compensation	long	Read Only

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.4.1. Position Reset (Read/Write, long) REG:200

This register in conjunction with REG:201 provides the ability to "preset" the Position Command register (REG:202) to a new position without disturbing the PID Loop. The procedure for doing this is as follows:

1. Load REG:200 with the desired position (in machine steps).
2. After loading the position, load REG:201 with a "1". This effectively transfers the value in REG:200 to the internal command register (REG:202).

7.4.2. Position Reset Trigger (Write Only, integer) REG:201

Refer to REG:200, above, for an explanation of the use of this register.

7.4.3. Position Command (Read/Write, long) REG:202

This register contains the accumulated PID Loop Position Command. This register also allows the user to enter a direct position command to the PID Loop. This results in instantaneous motion (not recommended). REG:202 reflects the total accumulated position (in steps) generated by the D( )...GO, DD( ), or CAM( ) commands.

7.4.4. Position Feedback (Read/Write, long) REG:203

This register contains the accumulated motor position and is the instantaneous motor position.

7.4.5. Current Command (Read/Write, long) REG:204

This register contains the instantaneous ratioed current command. This command is a signed binary number with "+" 100% represented by 0x7fffff.

7.4.6. Instantaneous Velocity Feedback (Read Only, long) REG:205

This register contains the instantaneous velocity of the motor. The update rate of this register depends on the setting of PRM:307, PRM:308, PRM:309 or PRM:310.

7.4.7. Averaged Velocity Command (Read Only, long) REG:206

This register contains the average commanded velocity over 12.8 msec

7.4.8. Kernel Timeslice Mask (Read/Write, Integer) REG:207

Register REG:207 allows masking of control functions generated by the U100 operating kernel. When the appropriate bit is set to zero, the given function is disabled or the condition is made false.

Serious operational side-effects can be generated by the improper use of this register. This register is used by predefined macro command libraries supplied with U100 controller (e.g., MAC100 and DEF100 files).



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 7-20. Bit Definitions for REG:207

BIT #	Definition
BIT 0	Not used.
BIT 1	Task 1 currently processing.
BIT 2	Task 2 currently processing.
BIT 3	Task switch.
BIT 4	Load position command.
BIT 5	Fetch Encoder 1 feedback.
BIT 6	Execute PID loop.
BIT 7	Execute commutation loop.
BIT 8	Output torque command.
BIT 9	Calculate position command.
BIT 10	Fetch encoder 2 feedback.
BIT 11	Not used.
BIT 12	Fetch R/D feedback.
BIT 13	Fetch A/D data and update D/A data.
BIT 14	Execute stepping motor indexing loop.
BIT 15	Execute stepping motor feedback verification.
BIT 16	Check limits, faults, and execute trapping.
BIT 17	Check control loop maximums.
BIT 18	Execute axis calibration compensation.
BIT 19	Not used.
BIT 20	Not used.
BIT 21	Not used.
BIT 22	Set synchronization flags and execute service request.
BIT 23	Reset feedhold.

7.4.9. Position Error (Read Only, long) REG:208

This register is the average position error over 12.8 msec.

7.4.10. Averaged Velocity Feedback (Read Only, long) REG:209

This register is the average motor velocity over 12.8 msec.

7.4.11. Accel/Decel Feedback (Read Only, long) REG:210

This register is the average acceleration/deceleration over 12.8 msec.

7.4.12. Axis Calibrated Position Cmd (Read Only, long) REG:211

This register provides the present position command ("uncompensated") when the Axis Calibration Mode is enabled (e.g., PRM:234 equals 1 or 2). The compensated position command can be read by accessing REG:202.

7.4.13. Axis Calibrated Position Feedback (Read Only, long) REG:212

This register provides the present position feedback (uncompensated) when the Axis Calibration Mode is enabled (e.g., PRM:234 equals a 1 or 2). The compensated position command can be read by accessing REG:203.

7.4.14. Axis Calibration Compensation (Read Only, long) REG:213

This register provides the present axis calibration error. This error is automatically subtracted from the position command when the Axis Calibration Mode is enabled (e.g., PRM:234 equals a 1 or a 2). Calibration error is automatically determined by linearly interpolating the error correction table (see PRM:234, PRM:235, PRM:236, and PRM:237) and subtracting the result from the present position command. This process occurs every 3.2 msec.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.5. System Registers

The System Registers contain the system status, axis status, and error conditions of the controller. REG:302, REG:307, REG:308, and REG:309 relate to Exception Processing, described in Chapter 5: Programming Commands.

Table 7-21. System Registers Summary

REGISTER	DESCRIPTION	VALUE TYPE	ACCESS CAPABILITY
300	Reserved	-	-
301	Motion Status	integer	Read Only
302	Error Status	integer	Read Only
303	Axis Status	integer	Read Only
304	System Status	integer	Read Only
305	Reserved	-	-
306	Reserved	-	-
307	Error Status "Active Mask"	integer	Read Only
308	Error Status "Reset Mask"	integer	Write Only
309	Error Status "Overlay"	integer	Write Only

7.5.1. Motion Status (Read Only, integer) REG:301

This register provides information on the general status of the controller. Note that the bits of this register directly reflect the condition of the LED indicators on the face of the controller. The bits that apply to this register are as follows.

Table 7-22. Bit Definitions for REG:301

BIT #	Definition
BIT 0	Positive "+" Limit This represents both the "+" Hardware Limit and the "+" Software Limit status.
BIT 1	Negative "-" Limit This represents both the "-" Hardware Limit and the "-" Software Limit status.
BIT 2	Overload If this bit is set, the system exceeded the following conditions: Position Error Threshold Velocity Feedback Threshold Accel/Decel Feedback Threshold Current Limit Time Out Threshold These conditions affect this bit even if the Exception Processing masks for these exceptions are disabled.
BIT 3	Remote This bit is set when either the RS-232 or IEEE-488 Host Mode is active.
BIT 4	Fault With the presence of any of the following conditions, this bit is set. – Encoder 1 or Encoder 2, ENC option, or R/D option error – Drive Fault – Run Time Error(s) in Task 1 or Task 2
BIT 5	Axis Calibration Range Error
BIT 6	In Position The system sets this bit once the motor is at its commanded position.
BIT 7	Marker This bit is set when the index pulse for Encoder 1 or Encoder 2 is true. PRM:118 selects the encoder to be monitored.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.5.2. Error Status (Read Only, integer) REG:302

This register contains 24 error and exception status bits. The bit definition for this register lines up with the bit descriptions for the Exception Processing. The following bit data applies. Note that a "1" represents a true for all bit descriptions below.

Table 7-23. Bit Definitions for REG:302

BIT #	Definition
BIT 0	Encoder 1 Feedback Error This indicates that an invalid sin/cos differential input exists or Encoder 1 is disconnected.
BIT 1	Encoder 2 Feedback Error This indicates that an invalid sin/cos differential input exists or Encoder 2 is disconnected.
BIT 2	Drive Fault The amplifier has a current tripped, there is no DC bus voltage to the amplifier, or the shunt fuse is open.
BIT 3	User Defined Input 2 This input has the default setting for fast feedhold (e.g., "FREEZE").
BIT 4	ENC Feedback Error This requires the ENC option and indicates the Resolution Multiplier Board is disconnected or malfunctioning.
BIT 5	User Defined Input 1
BIT 6	+ Soft Limit Encountered This implies that the "+" Software Limit Threshold (set by PRM:129) is detected.
BIT 7	- Soft Limit Encountered This states that the "-" Software Limit Threshold (set by PRM:130) is detected.
BIT 8	+ Hard Limit Encountered This implies that a "+" Hardware Limit is detected.
BIT 9	- Hard Limit Encountered This implies that a "-" Hardware Limit is detected.
BIT 10	Reserved
BIT 11	R/D Feedback Error Requires the R/D opt. and is set if the R/D tracking regulator is out of tolerance or the resolver is disconnected.
BIT 12	Position Error Trap PRM:223 sets this threshold. This bit monitors the difference between the commanded position and selected position feedback source.
BIT 13	Velocity Feedback Trap PRM:224 sets this threshold. This bit monitors the selected velocity feedback source.
BIT 14	Accel/Decel Feedback Trap PRM:225 sets this threshold. This bit monitors the accel/decel derived from the selected position feedback source.
BIT 15	Current Limit Time Out Trap PRM:213 and PRM:217 sets this threshold. This bit monitors the current command signal.
BIT 16	User Defined Software Trap 1 Setting bit #16 of REG:309 sets this bit. (Factory default stops Task 1.)
BIT 17	User Defined Software Trap 2 Setting bit #17 of REG:309 sets this bit. (Factory default stops Task 2).
BIT 18	User Defined Software Trap 3 Setting bit #18 of REG:309 sets this bit. (Factory default disables the power amplifier.)
BIT 19	Run Time Error in Task 1 When set, this bit indicates a run time error has occurred while running a program in Task 1.
BIT 20	Run Time Error in Task 2 When set, this bit indicates a run time error has occurred while running a program in Task 2.
BIT 21	Axis Calibration Range Error This bit indicates the following conditions (assuming the Axis Calibration Mode is enabled through PRM:234) Present position is less than zero Present position is greater than 16,000,000 steps Calibration Table size (PRM:235) is exceeded
BIT 22-23	Reserved

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.5.3. Axis Status (Read Only, integer) REG:303

This register provides miscellaneous information relative to current controller operations. The information in this register acts as a troubleshooting aid and does not normally need accessed. The following bit data applies. Note that a "1" represents true for all bit descriptions below.

Table 7-24. Bit Definitions for REG:303

BIT #	Definition
BIT 0	Feedforward
BIT 1	PID Position Error integration disabled during "motion"
BIT 2	Reserved
BIT 3	Servo Loop current command is negated
BIT 4	Position Feedback from Encoder 1
BIT 5	Position Feedback from Encoder 2
BIT 6	Position Feedback from RMX
BIT 7	Position Feedback from R/D
BIT 8	Marker referenced from Encoder 1
BIT 9	Marker referenced from Encoder 2
BIT 10	Commutation for DC brush
BIT 11	Commutation from R/D
BIT 12	Commutation from Hall and Sine interpolation
BIT 13	Commutation from Hall, six step
BIT 14	Velocity Feedback from Encoder 1
BIT 15	Velocity Feedback from Encoder 2
BIT 16	Velocity Feedback from ENC option
BIT 17	Velocity Feedback from R/D option
BIT 18	Hold stepper current "high"
BIT 19	Commutate in six step mode 0
BIT 20	Commutate in six step mode 1
BIT 21	Motion command is present
BIT 22	"DISABLE" fault mask processing in progress
BIT 23	Home initiated since power up or software reset

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.5.4. System Status (Read Only, integer) REG:304

This register provides information concerning Task 1 and Task 2 operating status, communication status, and other operations that are linked with the controller operating system (Kernel). A description of each bit appears below. A "1" represents true for all bit descriptions below.

Table 7-25. Bit Definitions for REG:304

BIT #	Definition
BIT 0	Absolute Positioning Mode This indicates that the U100 is operating in Absolute Mode or Incremental Mode as set by the ABSL or INCR commands.
BIT 1	Kernal synchronization flag. Activates only when PRM:133 = 3.
BIT 2	Running in Task 1 This indicates that Task 1 is running a program.
BIT 3	Running in Task 2 This indicates that Task 2 is running a program.
BIT 4	Block Run Mode in Task 1 This indicates that Task 1 is configured to run in Block Mode (see Chapter 4: Software Installation and Interface). If "0" the Auto Mode is active.
BIT 5	Block Run Mode in Task 2 This bit indicates that Task 2 is configured to run in Block Mode (see Chapter 4: Software Installation and Interface). If "0" the Auto Mode is active.
BIT 6	Running Block in Task 1 This bit implies that a block (e.g., U100 command statement) is currently being executed in the Block Mode for Task 1.
BIT 7	Running Block in Task 2 This bit implies that a block (e.g., U100 command statement) is currently being executed in the Block Mode for Task 2.
BIT 8	Holding Execution in Task 1 This bit indicates that a feedhold for Task 1 has been initiated with the "HLD1" command, see Chapter 4: Software Installation and Interface.
BIT 9	Holding Execution in Task 2 This bit indicates that a feedhold for Task 2 has been initiated with the "HLD2" command, see Chapter 4: Software Installation and Interface.
BIT 10	Software command uses position command as reference.
BIT 11	R/D board Operating in 14 bit Dynamic Resolution Mode This bit indicates the R/D option board has automatically switched from 16 bit to 14 bit tracking mode. This “switch” occurs at high data rates automatically to prevent the R/D tracking loop from saturating. This mode is only applicable if PRM:137 is set to 4 (default setting).
BIT 12	Recirculating Mode Enabled This bit indicated that the power stage recirculation mode is enabled. It is set only if PRM:200 I set to a “1” or a “2”. Note that power stage recirculation is applicable only for stepping motor operation.
BIT 13	Recirculation Mode Remains on During Motion This bit is set if PRM:200 is set to a “2”. In this mode, recirculation remains on at all times, even during motion execution.
BIT 14	Gear mode active.
BIT 15	Axis Calibration Enabled

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 7-25. Bit Definitions for REG:304 Continued

BIT #	Definition
BIT 16	Axis Calibration using the MEM option (Memory Expansion Board) Table.
BIT 17	Holding on "Group Trigger" (IEEE-488 and RS-232) This bit indicates that the Host Trigger has been set. See PRM:003 through PRM:006 for more information.
BIT 18	RS-232 Loop Mode Enabled This bit indicates that the RS-232 Host Mode Loop or Daisy Chain Mode is active. See PRM:003 through PRM:006 for more information.
BIT 19	Homing Mode In Progress This bit indicates a homing sequence is in process as initiated by the U100 HM command.
BIT 20	Processing Trajectory Generator through First Order Filter This bit indicates that a First Order Filter has been activated to filter the Trajectory Generator command to the PID Loop (or Commutation Loop if stepper motor). See PRM:101 and PRM:102 for more information.
BIT 21	RS-232 Loop Connected at This Unit This bit indicates that RS-232 Host Mode Loop has been broken at this unit. See PRM:003 through PRM:006 for more information.
BIT 22	"Xoff" Mode Active (RS-232 Mode) This bit indicates that the Xoff character has been sent by the U100 to the controlling Host in the form of Service Request 0x32. See Chapter 5: Programming Commands for additional information.
BIT 23	Service Request Processing in Progress This bit indicates that a Service Request has been sent to the Host. See Chapter 5: Programming Commands for more information.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.5.5. Error Status "Active Mask" (Read Only, integer) REG:307

This register provides a quick reference for evaluating the status of the Exception Processing masks.

The bit definitions for this register appear below. Any bit set in this register denotes that the corresponding mask(s) have been activated.

Table 7-26. Bit Definitions for REG:307

BIT #	Definition
BIT 0	Disable Mask
BIT 1	Reserved
BIT 2	Service Request Mask
BIT 3	Auxiliary Mask
BIT 4	Freeze Mask
BIT 5	Task 1 Mask
BIT 6	Task 2 Mask
BIT 7	Positive Direction Mask
BIT 8	Negative Direction Mask
BIT 9	Reserved
	. . .
BIT 23	Reserved

7.5.6. Error Status "Reset Mask"(Write Only, integer) REG:308

This register provides the means in which to reset an active exception defined in REG:302 that has been preset by PRM:318 to be latched or remain "sticky", when detected by the Exception Processing algorithm.

Each bit in this register corresponds to the bit definitions in REG:302.

To execute a reset of a latched condition, load this register with bits set to "1" for the corresponding exception condition to be reset. The bits in this register that are set to "0" do not effect the present status of the given exception conditions.

7.5.7. Error Status "Overlay"(Write Only, integer) REG:309

This register provides the ability for a program running in Task 1 or Task 2 to implement a software user defined exception. In essence, this register provides the interface in which to virtually set bit numbers 16, 17, and 18 of REG:302.

Loading bits 16, 17, and/or 18 with "1" effectively sets bits 16, 17, and/or 18 of REG:302, respectively. Setting any other bits in this register other than 16, 17, and 18 have no effect on REG:302.

The factory default setting of the mask parameters PRM:319 through PRM:327 define bits 16, 17, and 18 for the following functions:

Bit 16 stops and exits Task 1 if Task 1 is running.
Bit 17 stops and exits Task 2 if Task 2 is running.
Bit 18 disables the power amplifier (turning off the power amplifier).



For all three definitions above, the default condition is to latch these conditions.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7.6. Variables

Variables are RAM locations used to hold data or strings for the user. There are four types of variables that include integers, long integers, strings, and floating point. Each one of these variables are described in detail below.

Integer Types (24 bit)

BV:1 through BV:1000

PV:512 through PV:61439 (requires MEM/, DISP/, THM/, etc., option boards)

Long Type (48 bit)

LV:1 through LV:200

Float Type (signed 23 bit mantissa, ±E38 exponent range)

FV:1 through FV:1000

String Type (20 characters)

SV:1 through SV:19

All variable types with the exception of the PV:xx type are internal to the controller and battery backed. Thus, all values are retained in memory after power down.

Both Task 1 and Task 2, when running PGM programs can access any of these variables. In addition, Task 0 (the communication task for IEEE-488 or RS-232) can access these variables by accessing the Setup window.

Exercise caution when accessing or changing the same variable in multiple tasks using the FV:xx, LV:xx, or SV:xx variable types. Since these variables use two or more memory spaces to store values, there exists a chance that a variable being modified in one task may be "partially" changed just as that task is ending its time slot allocation. Thus, when the next task starts and begins to read the same variable, the value is corrupt or invalid. Use the SYNC command to solve this problem.

The PV:xx variables (port variables) listed above are actually references to the controller Extension Bus that has an address range of 0x200 (PV:512) to 0xefff (PV:61439). Thus, the type of option board connected to the Extension Bus determines the meaning and use of the specific port variable address.

For example, if the MEM option board (memory board) is connected to the extension bus, the user has access to 56,832, 24 bit, battery backed memory locations starting at 0x200 to 0xdfff.

7.6.1. Indexing the Variables

The controller provides a method to index the PV:, BV:, LV:, and FV: variables. The indexing variable can be otherwise referred to as a "pointer" or "index variable".

The pointer or index variable can be allocated to any of the one thousand BV: variables. An example of an equate statement to a floating point variable using BV:10 as the "index variable" is as follows.

FV:BV:10 = 1.34

The statement above can be interpreted as follows:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Using the integer variable in BV:10 as the index variable, extract the value in BV:10 and use it as the index. Thus, with BV:10 set for example, to the value of 210, the statement above is interpreted as follows.

FV:210 = 1.34

Remember, only BV: variables can be used as the index operator.

In addition to the equate operation shown above any mathematical operation or conditional operation (e.g., the IF(...), WHL(...) statements, etc.,) can use "index variables". For example, the statement below is valid.

FV:BV:1 = LV:BV:21 * REG:102

Note that the dynamic range of the chosen BV: variable is checked during run time. For example, if the statement above is executing in Task 1, and at the time of execution, BV:21 has the value of say 20001, a Task 1 Run Time Error is flagged (BIT 19 of REG:302) and the task is exited.

Run Time Error codes can be viewed through REG:016 for Task 1 and REG:017 for Task 2. See Appendix D for a list of Run Time Error codes.

7.6.2. Numerical Data Types

The numerical data formats for the variables listed in Section 7.6 are reviewed below. These formats relate to constant expressions only.

7.6.3. Integers (24 bit or 48 bit)

Base 8 Notation:	leading "0" character
Example Statement:	BV:30 = 077 (which is BV:30 = 63 in base 10)
Base 10 Notation:	none
Example Statement:	LV:1 = 1234567
Base 16 Notation:	leading "0x" characters
Example Statement:	PV:0x200 = 100 (which is PV:512 = 100 in base 10)

7.6.4. Floating Point

Fractional Notation:	fixed decimal point
Example Statement:	FV:1 = 1.234
Floating Point Notation:	signed exponential, base 10
Example Statement:	FV:10 = 3.456789e-3

∇ ∇ ∇

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CHAPTER 8: SAMPLE PROGRAMS

In This Section:

- Introduction 8-1
- Trapezoidal Move 8-2
- Trapezoidal Move with Output-on-the-Fly 8-2
- Thumbwheel Input for Distance 8-3
- Display Readout 8-4
- Operator Input from HT 8-5
- Multitasking 8-6
- Data Logging On-the-Fly 8-8
- Master/Slave Move 8-9
- Proportional Joystick 8-10
- Synchronized Trajectory Table Execution 8-11
- Using External Interrupt to Change a Motion Trajectory 8-12
- Initializing Counter Chip/Data Transfer from Preset Register 8-14
- Transferring Data from the Counter to Output Latch 8-15
- Using Comparator Outputs of the Encoder Counter Chip 8-16
- Borrow and Carry Control Outputs of the Counter Chip 8-18
- IEEE-488 Program Examples 8-20

8.1. Introduction

This chapter is intended to provide an overview of several UNIDEX 100/U100i applications. It is assumed that the user has unpacked and checked the U100/U100i system, configured the hardware and software, installed the necessary components, and is otherwise ready to begin using the UNIDEX 100/U100i system in a *real* application.

The application examples and associated programs in this section are intended to give the reader only a general overview of just some of the capabilities of the UNIDEX 100/U100i system. These samples provide some basic fundamentals on which more advanced applications may be realized.

The program examples shown in this chapter use comments (descriptive text that follows a semicolon character ";") to explain the program statements. While thorough (and relevant) commenting in a program is very useful (and is encouraged), it may be necessary to limit the amount of comments in your program if the program size (including comments, commands, etc.) becomes too large



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.2. Trapezoidal Move

The trapezoidal move is a point-to-point move where distance, velocity, and acceleration are programmed in user units. Assuming that the user requires a 5 (.2"/rev.) pitch ball screw and a resolution of 2000 counts per revolution to accomplish a move. The user units parameter is 10,000 counts per inch (PRM:100=10,000).

```
BEGIN          ; Start the program.
D(5)           ; Set the distance equal to five inches.
V(2)           ; Set the velocity equal to two inches per second.
A(8)           ; Set the acceleration/deceleration equal to 8
               ; inches per second square.
GO             ; Start the move.
END            ; End the program.
```

8.3. Trapezoidal Move with Output-on-the-Fly

This move is the same as the trapezoidal move, but allows the user to add output-on-the-fly. Assuming that the user wants to place a uniform bead of glue on a surface this program uses output bit #1 to turn the glue on and off. The following program illustrates how to place a three-inch long bead of glue at a constant velocity.

```
BEGIN          ; Start the program.
REG:2=0x00     ; Reset output bit #1.
D(1)           ; Set the distance equal to one inch.
V(2)           ; Set the velocity equal to two inches per second.
REG:2=0x01     ; Set the output bit #1 high to turn on the glue.
D(3)           ; Set the distance equal to three inches.
REG:2=0x00     ; Reset output bit #1 to turn off the glue.
D(1)           ; Set the distance equal to one inch. Turn off the
               ; glue and move another inch.
GO             ; Start the move.
END            ; End the program.
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.4. Thumbwheel Input for Distance

In some cases the user may wish to enter the length necessary for cut-to-length applications. To do this, the length must be an input variable. The variable value can come from a Host controller, a set of thumbwheels, or an operator keypad/display unit. The optional THM is a 6-digit thumbwheel addressable on the controller expansion bus. The location for the thumbwheel is E000 (hex). The thumbwheel includes a push-button that latches data. The user may read this as a momentary switch located at E001 (hex). In this case, the user wants the thumbwheel range/resolution to be ±99.999 inches. The user units are 1000 counts per inch. For an example of how this works refer to the following program.

```
TITLE**Read the move distance from thumbwheel option**
DEF304                                  ; Include Definition File #304 in this program.
BEGIN                                   ; Start the program.
BUTTON=BUTADR AND 0x10000               ; Button will equal 0x10000 when depressed.
WHL(BUTTON NE 0x10000)                  ; Wait until the button is depressed.
BUTTON=BUTADR AND 0x10000               ; Set the BUTTON variable equal to BUTADR
                                        ; anded with 0x10000.
ENDWHL                                  ; End the while loop.
DIST=CBI(THUMB)                         ; Set distance equal to the CBI of variable
                                        ; THUMB.
IF(SIGN AND 0x800000)                   ; Convert distance from BCD to integer. If sign is
                                        ; negative, DIST is negated.
DIST=-1*DIST                            ; Set the variable DIST equal to a negative one
                                        ; times the variable DIST.
ENDIF                                   ; End the IF statement.
DIST=DIST/1000                          ; Set the DIST variable equal to the value in
                                        ; DIST divided by one thousand.
D(DIST)                                 ; Set the distance equal to the value of the DIST
                                        ; variable.
V(6000)                                 ; Set the velocity equal to 6000.
GO                                      ; Start the move.
END                                     ; End the program.
```

The DEF file below, referred to as DEF304, corresponds to the above program.

```
BUTADR=PV:0xE001
THUMB=PV:0xE000
SIGN=PV:0xE001
BUTTON=BV:1
DIST=FV:1
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.5. Display Readout

The following example is a remote position readout being updated every 0.1 second. The user units for this program are 1000 counts per inch.

```
BEGIN                       ; Start the program.
PV:0xE002=4                 ; Set the port variable equal to four (four
LB:10                       ; decimal places.
BV:1=REG:203                ; Define the following program blocks as label 10
                            ; Set integer variable #1 equal to the value of
PV:0xE000=CIB(BV:1)         ; register #203 (feedback position).
                            ; Set the port variable equal to the BCD value of
                            ; integer variable #1 and send the BCD position
IF(REG:203 LT 0)            ; data to the display.
                            ; Determine if register #203 is less than zero. If
                            ; it is negative, carry out the following
  PV:0xE001=-1              ; statement.
                            ; If the above statement was true, send the
ELSE                        ; minus sign to the display.
                            ; If register #203 is not less than zero, carry out
  PV:0xE021=0               ; the following statement.
                            ; If the above statement was not true, send plus
                            ; sign to the display.
ENDIF                       ; End the IF statement.

DW(.1)                      ; Set the dwell time equal to .1 second.
GOTO:10                     ; Jump to the block labeled 10.
END                         ; End the program.
```

This program would normally execute in one task while motion is performed in the other. This program only monitors position feedback and displays it.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.6. Operator Input from HT

Variable input and output through the hand held terminal is an option to the controller system. This option is the HT hand held terminal that includes an 80 character display and keypad. Using the HT, the user may send messages to an operator as well as transmit values to selected variables. The program below illustrates how to do this. Assume that the user units are 1000 counts per second, and the program name is PGM2.

```
DEF2                        ; Include Definition File #2 in this program.
BEGIN                       ; Start the program.
CLS                         ; Clear the terminal screen.
PM("Enter length of cut")   ; Print the message to the terminal screen.
CUR(2,1)                    ; Position the cursor at row 2, column 1.
DIST=GM(3)                  ; Get the input from the terminal, display the
                            ; result, and store it in the variable DIST.
CUR(3,1)                    ; Position the cursor at row 3, column 1.
PM("Enter # of moves")      ; Print the message to the terminal screen.
CUR(4,1)                    ; Position the cursor at row 4, column 1.
CYCLE=GM(3)                 ; Get the input from the terminal, display the
                            ; result, and store it in the variable CYCLE.
V(2)                        ; Set the velocity equal to 2 inches per second.
WHL(CYCLE GT 0)             ; While CYCLE is greater than zero perform the
                            ; following move.
D(DIST)                     ; Set the distance equal to the value in the DIST
                            ; variable.
GO                          ; Start the move.
CYCLE=CYCLE-1               ; Set CYCLE equal to the value in the CYCLE
                            ; variable minus one.
ENDWHL                      ; End the while loop.
END                         ; End the program.
```

The following file is the “DEF2” file that corresponds to PGM2 shown above.

```
DIST=FV:1
CYCLE=BV:2
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.7. Multitasking

It is necessary in many applications to monitor inputs or display data asynchronously from a motion program. The controller has multitasking ability to allow a second task to run while running a motion program. Task interaction may occur if both tasks are executing motion or using the same variables.

The following is an example of a remote position readout being updated every 0.1 second. The readout is the DISP option, that is a 6 digit LED display on the expansion bus at E020 hex. The user units for this program are 1000 counts per inch. See the definition file below followed by the actual program that uses it.

```
DISP=PV:0xE020      ; Equate the PV value to the display location.
SGN=PV:0xE021       ; Equate the PV value to the sign location.
                    ; Equate the PV value to the decimal point
DP=PV:0xE022        ; location.
```



Comments are not allowed in DEF files. The example above is for readability of the syntax only.

```
DEF:3               ; Include Definition File #3 in this program.
BEGIN               ; Start the program.
LB:10               ; Define the following program blocks as label 10
BV:1=ABS(REG:203)   ; Set integer variable #1 equal to the absolute
                    ; value of register #203 (feedback position).
DISP=CIB(BV:1)      ; Set the display variable equal to the CIB of
                    ; integer variable 1 and send the BCD position
                    ; data to the display.
IF(REG:203 LT 0)    ; Determine if register #203 is less than zero. If
                    ; it is negative, carry out the following
                    ; statement.
  SGN=-1            ; If the above statement was true, send the
                    ; minus sign to the display.
ELSE                ; If register #203 is not less than zero, carry out
                    ; the following statement.
  SGN=0             ; If the above statement was not true, send zero
                    ; to the display.
ENDIF               ; End the IF statement.
DP=4                ; Set the DP variable equal to four.
DW(.1)              ; Set the dwell time equal to .1 second.
GOTO:10             ; Jump to the block labeled 10.
END                 ; End the program.
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The following program is a motion program that can be run simultaneously with previous example program to demonstrate controller multitasking.

```
BEGIN
LB:20                   ;Define program block
D(10)                   ;Go 10 inches
V(1)                    ;Velocity of 1”/sec
T(.1)                   ;Ramp time of 100 mSec
GO                      ;make move
DW(1)                   ;dwell 1 second
D(-10)                  ;move back 10 inches
GO                      ;Make move
DW(1)                   ;Dwell 1 second
GOTO :20                ;Repeat program block
END
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.8. Data Logging On-the-Fly

Most often the defect locations for an inspection application are data logged. This allows the controller to capture position while moving within 5 micro seconds. Upon locating a defect, the user pulls the interrupt line. The function of the interrupt is to transfer the position data to a variable. The user may later transfer the variable back to the Host.

```
TITLE**Task 1**          ; Attach the title to the program number.
BEGIN                    ; Start the program.
EI                       ; Enable the external interrupt.
BV:1=1                   ; Set the integer, variable #1 equal to one.
D(5)                     ; Set the distance equal to five.
V(2)                     ; Set the velocity equal to two.
GO                       ; Start the move.
END                      ; End the program.
```

```
TITLE**Task 2**          ; Attach the title to the program number.
BEGIN                    ; Start the program
LV:BV:1=REG:113          ; Evaluate the value found in register #113,
                         ; then equate that value to integer, variable #1
                         ; and put final result in the LV variable.
BV:1=BV:1+1              ; Set integer variable equal to integer,
                         ; variable 1 plus one.
EI                       ; Enable the external interrupt.
END                      ; End the program.
```



For the program above, the following parameter values apply.
PRM:133=1 “Task2 interrupt”
PRM:134=2 “Unlatched, edge low”
PRM:135=2 “External Interrupt”

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.9. Master/Slave Move

To achieve a flying cutoff application it is necessary to synchronize both the start of the operation and the speed. To initiate the start the user must first detect the registration mark. To initiate the start of the operation requires that the registration mark be detected. The speed of the cutoff move (slave axis) is proportional to the master axis speed. An encoder attached to the master axis connects to the controller auxiliary encoder input. Assume that for the following program the user units are 1000 counts per second. After the synchronized move, the axis returns home in a non-slaved fashion.

```
BEGIN                ; Start the program.
HM                   ; Initialize the home position to zero.
PRM:145=.05          ; Set parameter #145 (scaled velocity) equal to
                     ; twice the auxiliary encoder rate.
PRM:147=1            ; Set parameter #147 (accel time) equal to 0.1
                     ; second during the velocity scaling.
LB:10                ; Define the following blocks as label 10
WHL(REG:1 EQ 0)      ; While register #1 is equal to zero, continue
                     ; with the loop. Otherwise don't execute.
ENDWHL               ; End the while loop.
PRM:144=6            ; Set parameter #144 (reset velocity scaling
                     ; mode) equal to six.
V(10)                ; Set the velocity equal to ten.
D(5)                 ; Set the distance equal to five.
GO                   ; Start the move.
PRM:144=0            ; Set parameter #144 (disable velocity scaling)
                     ; equal to zero.
V(4)                 ; Set the velocity equal to four.
D(-5)                ; Set the distance equal to negative five.
GO                   ; Start the move.
GOTO:10              ; Jump to the block labeled 10.
END                  ; End the program.
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.10. Proportional Joystick

This program uses the analog input as a joystick input, parameter PRM:140 is the scale factor, PRM:141 is the dead band adjust for zero, register REG:001 is the joystick buttons. Motion is accomplished by use of the DD (Direct Drive) command.

```
BEGIN                          ; Start the program.

BV:1=1                         ; Set integer variable #1 equal to 1 for
                               ; controlling program duration.
BV:3=1                         ; Initialize BV:3 to 1 for use as an array pointer.
PRM:140=100                    ; Set the scale factor to 100 (±10V=100 user
                               ; units/6.4msec).
PRM:141=.05                    ; Set the dead band equal to 5%
WHL(BV:1 EQ 1)                 ; While loop for controlling program duration.
  FV:1=ADC                     ; Read analog input (joystick) into FV:1.
  DD(FV:1)                     ; Write FV:1 data to trajectory regulator with
                               ; the direct drive (DD) command.
  IF(REG:001 EQ 0xFC)          ; Test for top button of joystick being pressed.
                               ; (button A wired to input 1, button B wired to
                               ; input 2, button C activates both inputs).
    LV:BV:3=REG:102            ; Read encoder 1 position into variable array.
    BV:3=BV:3+1                ; Increment array pointer.
    CLS                        ; Clear the terminal screen.
    CUR(1,1)                   ; Position the cursor at row 1, column 1.
    PM("LV:")                  ; Output the variable and value to the screen.
    CUR(1,4)                   ; Position the cursor at row 1, column 4.
    PM(BV:3, "%3.D")           ; Print the message to the terminal screen.
    CUR(1,7)                   ; Position the cursor at row 1, column 7.
    PM("=")                    ; Print the message to the terminal screen.
    CUR(1,10)                  ; Position the cursor at row 1, column 10.
    PM(REG:102)                ; Print the encoder position to the terminal screen.
    ENDIF                      ; End the IF statement.
ENDWHL                         ; End the while loop.
END                            ; End the program.
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.11. Synchronized Trajectory Table Execution

The controller can execute arbitrary trajectories that the user inputs in a table form. This mode of operation is similar to CAM table execution except for the way this mode inputs the tables.

Trajectory tables are positions that get stored in sequential variable locations. The controller can index through these tables in a continuous, circular fashion. The following example first generates a 500 point sine function table then executes it as a function of the auxiliary encoder position (REG:103). Position, speed, and direction of the motor get slaved to the auxiliary encoder input.

The MDX (Modulo) command works in conjunction with PRM:110 (first table entry) and PRM:111 (last table entry). This establishes the trajectory table pointer location. The user may smooth out the execution of the table by using the trajectory filter set with PRM:101 and PRM:102.

```
BEGIN                    ; Start the program.
BV:1=0                   ; Initialize the loop count to 0.
BV:2=100                 ; First table location.
WHL(BV:1 LT 500)         ; Start of sine table generator loop.
  FV:3=BV:1*.01256637    ; Get the angle in radians.
  FV:4=SIN(FV:3)         ; Calculate sine of angle.
  FV:BV:2=FV:4*2000      ; Scale and put into table.
  INC(BV:1)              ; Next loop count.
  INC(BV:2)              ; Next table location to fill.
ENDWHL                   ; End of table generator loop.
ABSL                     ; Initiate the Absolute Mode.
LB:1                     ; Start of trajectory generator loop.
FV:1=REG:103             ; Get auxiliary encoder position.
BV:1=MDX(FV:1)           ; Get table location.
DD(FV:BV:1)              ; Output the trajectory command.
GOTO:1                   ; Get the next trajectory.
END                      ; End the program.
```

The user must enable the Encoder 2 update by adjusting parameter PRM:308.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.12. Using External Interrupt to Change a Motion Trajectory

This example shows a method in which a motion trajectory can be altered dynamically, by using the external interrupt input. Typical applications that may use this mode are cut to length machines that use an index pulse to synchronize the conveyer to the cutting mechanism.

A detailed description of the parameters setting required to set up this mode of operations is explained in Chapter 6: Parameters. The specific parameters are PRM:132, PRM:133, PRM:134, and PRM:135.

For the programs listed below, set PRM:133 to a 1.

The program below is PGM1. Load this program into Task 1 after loading PGM2 into Task 2.

```
BEGIN                   ; Start the program.
BV:1=1                  ; Initialize the loop count to 1.
WHL(BV:1 EQ 1)          ; Execute the loop until BV:1 is not one.
  REG:2=0xFF            ; Set all 8 bit outputs high.

  WHL(REG:1 EQ 0xFF)    ; Hold execution until any bit input is set low.
  ENDWHL

  D(100000)             ; Begin execution of typical trajectory
  V(10000)              ; until 100000 user units is reached.
  T(.05)

  EI                    ; Then, enable the interrupt.

  D(100000)             ; Trajectory will continue on an additional 100000
                        ; user units unless an interrupt occurs.
  GO
                        ; Start the motion listed above.
  DI
                        ; Disable the interrupt at end (note that the
ENDWHL                  ; interrupt may or may not have occurred).
END                     ; End the outer loop.
                        ; End the program.
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The program below is PGM2. Load this program into Task 2 before loading PGM1 to Task 1.

```
BEGIN                   ; Start the program.
D(2500)                 ; Motion is intercepted with this trajectory when
V(8000)                 ; interrupt occurs.
T(.08)
GO

REG:2=0xFD              ; Turn bit output #2 on and off for .5 seconds.
DW(.5)
REG:2=0xFF
                        ; At this point, the program command pointer for
                        ; this program (PGM2) is reset to "D(2500)" (e.g.,
                        ; program loops around and waits for the next
                        ; interrupt). The program trajectory is aborted
                        ; and execution in PGM1 continues with the "DI"
                        ; command.
END                     ; End the program.
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.13. Initializing Counter Chip/Data Transfer from Preset Register

This is a very simple program that demonstrates two things; the basics of initializing a counter chip to perform a given operation, and how to transfer data from the preset register to the counter by way of activating the LOAD input.

The first step in programming a counter chip should always be to provide it with a master reset. Then, the rest of the initialization process can take place. There are two alternatives to transferring data. The first takes the data from the preset register and transfers it to the counter. The other takes the data from the counter and sends it to the output latch. To perform one of these two functions, it is necessary to use software commands or to configure the LOAD input.

With the LOAD input pulled to a logic low state, the following program uses the LOAD input to transfer the data in the preset register to the counter. Data gets transferred continuously from the counter into the output latch by way of a software command. The program reads the output latch after each transfer. Upon activation of the LOAD input, the number 1500 (decimal value) gets transferred from the preset register to the counter. Then, the software command forwards this value into the output latch. It then gets read into variable BV:1 and displayed on the screen. This example initializes the counter chip and transfers data from the preset register to the counter.

```
BEGIN           ;  Start the program.
REG:33=0x20     ;  Write to MCR (Master Reset)
REG:32=0x5DC    ;  Set the PR to decimal 1500.
REG:34=0x48     ;  Write to ICR (enable P4-3, 8, 2, 7 inputs and
                ;  arm LOAD 2 input to transfer PR to CNTR
                ;  when pulled to a logic low).
REG:36=0xC1     ;  Write to QR (enable x1 quadrature mode
                ;  counting).
CLS             ;  Clear the terminal screen.
CUR(1,1)        ;  Position the cursor at row 1, column 1.
PM("BV:1=")     ;  Print the message to the terminal screen.
LB:1            ;  Define the following program block as label 1.
REG:33=0x02     ;  Write to MCR (transfer CNTR to OL).
BV:1=REG:31     ;  Set BV:1 equal to the OL value in REG:031.
CUR(1,9)        ;  Position the cursor at row 1, column 9.
PM(BV:1)        ;  Print the message to the terminal screen.
GOTO:1          ;  Go to the label #1 and repeat the function
                ;  indefinitely.
END             ;  End of the program.
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.14. Transferring Data from the Counter to Output Latch

This example shows a somewhat more real world application of using the LOAD input. This program uses the LOAD input to transfer data from the counter to the output latch upon activation. This is essentially a position capture program that obtains the position of the system instantaneously. The program runs in one of the UNIDEX 100 tasks while a motion program is running in the other.

First, it is necessary to reset the master control register. Then, the inputs are enabled along with configuring LOAD2 to transfer the counter value to the output register upon activation. The counting occurs in times 4 quadrature mode. The preset register, counter, and output latch all get initialized to zero. The display constantly updates the value of BV:1 by reading the contents of the output latch into BV:1.

There is no direct command to transfer the counter value to the output latch. To do this it is necessary to pull LOAD 2 to a logic low. Thus the value of BV:1 only changes while LOAD 2 is at a logic low.



```
BEGIN                   ; Start the program.

REG:33=0x20             ; Write to MCR (master reset)
REG:34=0x68             ; Write to ICR (enable P4-3, 8, 2, 7 inputs and
                        ; arm LOAD 2 input to transfer CNTR to OL
                        ; when pulled to a logic low).
REG:36=0xC3             ; Write to QR (enable X4 quadrature mode
                        ; counting).
REG:32=0x0              ; Write to PR (initialize to 0).
REG:33=0x08             ; Write to MCR (transfer PR to CNTR).
REG:33=0x02             ; Write to MCR (transfer CNTR to OL).

CLS                     ; Clear the terminal screen.
CUR(1,1)                ; Position the cursor at row 1, column 1.
PM("POS=")              ; Print the message to the terminal screen.

LB:1                    ; Define the following program block as label 1.
BV:1=REG:31             ; Set BV:1 equal to the OL value in REG:031.
CUR(1,8)                ; Position the cursor at row 1, column 8.
PM(BV:1, "%10.d")       ; Print the message to the terminal screen.
GOTO:1                  ; Go to the label #1 and repeat the function
                        ; indefinitely.

END                     ; End of the program.
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.15. Using Comparator Outputs of the Encoder Counter Chip

This program (shown on next page) shows how to use the comparator outputs of the encoder counter chip. First, it is necessary to perform a master reset. Then, after enabling the inputs, by default, the system configures the LOAD 2 input to transfer the preset register to the counter. The quadrature register counts in times 4 quadrature mode. Once initialized to zero, the preset register transfers its value into the counter through use of a software command. The output control register provides a mono-stable, active low COMP2-N output that gets triggered each time it generates a carry or borrow. The COMPT2 output provides a bi-stable flip-flop that changes state each time it generates a carry or borrow. Set in the output control register, the divide-by-N count mode makes sure that the counter gets re-loaded with the preset register data each time it generates a carry or borrow.

The counter's setting is for 100 counts below the carry point so that while making a plus move it generates a carry every 100 steps, thus causing the outputs to fire. For a negative move, the counter's setting is at 100 (100 counts above the borrow point) so that while making a negative move, it generates a borrow for every 100 counts causing the outputs to fire.



When using comparator outputs of the encoder counter chip, jumpers JP22 and JP23 on controller control board must be placed in positions 1-2. Refer to Chapter 3: Hardware Configuration and Description, Figure 3-4.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

This program shows how to use the comparator outputs of the encoder counter chip.

```
BEGIN                   ; Start the program.
REG:33=0x20             ; Write to MCR (master reset)
REG:34=0x48             ; Write to ICR (enable P4-3, 8, 2, 7 inputs and
                        ; arm LOAD 2 input to transfer PR to CNTR).
REG:36=0xC3             ; Write to QR (enable x4 quadrature mode
                        ; counting).
REG:32=0x0              ; Write to PR (initialize to 0).
REG:33=0x08             ; Write to MCR (transfer PR to CNTR).
REG:35=0xB4             ; Write to OCR (set divide by N count mode and
                        ; set COMP2-N to function as COMP-N and
                        ; COMPT2 to function as COMPT).

BV:1=1                  ; Set integer variable #1 equal to 1.
WHL(BV:1=1)             ; While integer variable equals one loop until
                        ; indication to terminate the program.
  REG:32=0xFFFF9B       ; Write to PR(set to 16777115, which is 100
                        ; counts below a carry).
  REG:33=0x08           ; Transfer the PR to CNTR.
  D(10000)              ; Set the distance equal to 10000.
  V(10000)              ; Set the velocity equal to 10000.
  T(.1)                 ; Set the accel/decel time to .1 second.
  GO                    ; Start the move.
  DW(.5)                ; Dwell for a period of 0.5 seconds.
  REG:32=0x64           ; Write to PR (set to 100, which is 100 counts
                        ; above a borrow).
  REG:33=0x08           ; Transfer the PR to CNTR.
  D(-10000)             ; Set the distance equal to -10000 steps.
  GO                    ; Start move (use velocity and time constant
                        ; from the previous motion command).
  DW(.5)                ; Dwell for a period of 0.5 seconds.
ENDWHL                  ; End the WHILE loop.
END                     ; End of the program.
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.16. Borrow and Carry Control Outputs of the Counter Chip

This program is an example that shows how to use the BW (borrow) and CY (carry) control outputs of the counter chip.

The use of the BW and CY outputs are similar in operation to the “COMP-N” and “COMPT” output described in the previous section.

The difference is the way the counter (CNTR) is automatically loaded by the Preset Register (PR) during the terminal count. With the “COMP-N” and “COMPT” control outputs, the output state changes when the preset register equals the counter. This is while the counter is being loaded with the preset register at the terminal count (“carry” if passing the 24 bit modulus in the “+” direction. “Borrow”, if passing the 24 bit modulus in the “-” direction).

With the CY and BW control outputs, the output state changes at the terminal count (carry or borrow).

Therefore, with the “COMP-N” and “COMPT” outputs, there is a chance of a jitter (e.g., multiple output pulse) if the comparison between the preset register and the counter does not translate cleanly before the terminal count is reached.



Jumpers JP22 and JP23 must be placed in positions 2-3, refer to Figure 3-4 in Chapter 3: Hardware Configuration and Description.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

This program illustrates how to use the BW (borrow) and CY (carry) control outputs of the counter chip.

```
BEGIN
REG:33 = 0x20              ;Write to MCR (master reset)
REG:34 = 0x48              ;Write to ICR (enable P4 -3, 8, 2, 7 inputs)

REG:36 = 0xC3              ;Write to QR (enable x4 quadrature mode counting

REG:32 =0x0                ;Write to PR (initialize to 0)
REG:33 = 0x08              ;Write to MCR (transfer PR to CNTR)
REG:35 = 0xA4              ;Write to OCR (set divide by N count mode and set for CY
                           ;and BW active high outputs)
BV:1 = 0                   ;Execute 10 times....
WHL(BV:1 LT 10)
  REG:32 = -100            ;Write to PR register -100 counts (0xffff9c)
  REG:33 =0x8              ;Transfer PR to CNTR
  V(40000)                 ;Move motor 50000 counts
  D(50000)
  GO
  DW(2)                    ;Wait for 2 seconds
  INC(BV:1)
ENDWHL

DW(10)                     ;Dwell 10 seconds

REG:35 = 0x84              ;Write to OCR (set divide by N count mode and set for
                           ;CY-N and BW-N active low inputs)

BV:2 = 0                   ;Execute 10 times...
WHL(BV:2 LT 10)
  REG:32 = 99              ;Write to PR register 99 counts (0x63)
  REG:33 = 0x8             ;Transfer PR to CNTR
  D(-50000)
  GO
  DW(2)                    ;Wait for 2 seconds
  INC(BV:2)
ENDWHL

REG:34 = 0x40              ;Disable P4 -3, 8, 2, 7 inputs

END
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.17. IEEE-488 Program Examples

The programming examples in the following sections are very basic and serve as an aide to understand the IEEE-488 interface.



Example programs in 9.17.1., 9.17.2., and 9.17.3. must be run on a Hewlett-Packard (HP-85) computer. The controller must have an IEEE-488 interface.

8.17.1. HP85 Immediate Mode Command

The following program is an HP85 program used to issue an immediate mode motion command to the UNIDEX 100.

EXAMPLE:

```
10 !                                  ; HP85 Immediate Mode command to move
                                      ; axis.
20 ! Move Axis
30 !
40 IMAGE #, "#CBD(1000)", "↑"         ; Host Mode Immediate Mode command
50 OUTPUT 704 USING 40 ;              ; Send command to U100
60 END                                ; End of Program
```

8.17.2. HP85 Write and Read BV Variable

The following program is an HP85 program used to load BV:28 with a value of 10 and also read the value of BV:28 back.

EXAMPLE:

```
10 !
20 ! Write & Read BV:28               ; HP85 Write & Read BV:28
30 !
40 IMAGE #, "#FCCA", 2Z, "↑B"         ; Host Mode command for BV:28 Write
50 IMAGE #, 2Z, "↑"                   ; Specify # value digits and LF
60 OUTPUT 704 USING 40 ; 28           ; Send Host Mode command for BV:28
                                      ; Write
70 OUTPUT 704 USING 50 ; 10           ; Send new value for BV:28
80 IMAGE #, "#FCCA", 2Z, "↑A"         ; Host Mode command for BV:28 Read
90 IMAGE #%, #%K                      ; Input Specifications
100 OUTPUT 704 USING 80 ; 28          ; Send Host Mode command for BV:28
                                      ; Read
110 ENTER 704 USING 90 ; X$           ; Input BV:28 value
120 DISP X$                           ; Display BV:28 value
130 END                               ; End of Program
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.17.3. HP85 Retrieve Program from UNIDEX 100

This program is an HP85 program used to acquire a program from the UNIDEX 100. This program also contains a serial poll example.

EXAMPLE:

```
10 !
20 !                                    ; HP85 get program #2
30 !
40 DIM Z$[1000]                         ; Provide program buffer
50 IOBUFFER Z$                          ; Input/Output buffer
55 B=0                                  ; Serial poll flag
60 ON INTR 7 GOSUB 1000                 ; Setup interrupt conditions
70 ENABLE INTR 7 ; 8                    ; Enable interrupt
80 REMOTE 704                           ; Device #4 remote
90 IMAGE #, #FDBA", 1Z, "↑"             ; Host Mode command to request
                                        ; program
100 OUTPUT 704 USING 90 ; 2             ; Request program #2 for controller
110 IF B=0 THEN GOTO 110                ; Wait until serial poll is performed
120 ON EOT 7 GOTO 200                   ; Set up end of transmission entry
                                        ; line
130 TRANSFER 704 to Z$ INTR ; DELIM 126 ; EOF terminates
140 RESUME 7                            ; Continue
150 ! Loop Till Data Terminated         ; Loop again
160 GOTO 150                            ; Display program #2
200 DISP Z$                             ; End of Main Program
210 END

1000 !
1010 ! Service Request
1020 !
1030 S=SPOLL (704)                      ; Do serial poll
1040 STATUS 7, 1 ; C0                   ; IEEE status
1050 B=1                                ; Set serial poll flag
1060 ENABLE INTR 7 ; 8 @ RETURN         ; Enable serial poll interrupt
1100 END                                ; End of Program
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.17.4. QB Immediate Mode Command

The following program is a Quick Basic program used to issue an immediate mode motion command to the UNIDEX 100. Initially, Aerotech wrote this program for the National Instruments GPIB board using the Universal language interface "HP-Style Calls".

EXAMPLE:

```
/IEEE BASIC PROGRAM #2, IMMEDIATE MOVE COMMAND
/INITIALIZE THE INPUT & OUTPUT CHANNELS
OPEN "gpib0" FOR OUTPUT AS #1                ; IEEE output channel
OPEN "gpib0" FOR INPUT AS #2                 ; IEEE input channel
/INITIALIZE THE BUS & RESET TO DEFAULT PARAMETERS
PRINT #1, "ABORT"                            ; Initialize IEEE port
PRINT #1, "RESET"                            ; Reset IEEE port
PRINT #1, "GPIBEOS OUT CR"                   ; IEEE output string terminator
PRINT #1, "TIMEOUT 0"                        ; Timeout

/PLACE THE DEVICE IN THE REMOTE STATE
PRINT #1, "REMOTE 4"                         ; IEEE device #4 REMOTE

PRINT #1 "CLEAR 4"                           ; Clear device #4

/SEND THE COMMANDS TO THE controller         ; Immediate Mode Host Command
PRINT #1, "OUTPUT 4 ; #CBD(1000)"            ; "MOTION"

END                                          ; End of Program
```



Example programs in 8.17.4., 8.17.5., and 8.17.6. are Microsoft Quick Basic programs that run on an IBM PC. The PC and the controller must each be equipped with an IEEE-488 interface.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.17.5. QB Write and Read BV Variable

The following program is a Quick Basic program used to load BV:28 with a value of 10 and also read the value of BV:28 back. Aerotech wrote this program for the National Instruments GPIB board using the Universal Language. "HP-style Calls".

EXAMPLE:

```
/ IEEE BASIC PROG, WRITE, READ BV:28
/INITIALIZE THE INPUT & OUTPUT CHANNELS
OPEN "gpib0" FOR OUTPUT AS #1          ; IEEE Output Channel
OPEN "gpib0" FOR INPUT AS #2           ; IEEE Input Channel
/INITIALIZE THE BUS & RESET TO DEFAULT PARAMETERS
PRINT #1, "ABORT"                      ; Initialize IEEE Port
PRINT #1, "RESET"                      ; Reset IEEE Port
PRINT #1, "GPIBEOS OUT CR"             ; IEEE Output String Terminator
PRINT #1, TIMEOUT 0"                   ; Timeout
/PLACE THE DEVICE IN THE REMOVE STATE
PRINT #1, "REMOTE 4"                   ; IEEE Device #4 Remote

PRINT #1, "CLEAR 4"                    ; Clear IEEE Device #4

/SEND THE COMMANDS TO THE controller
PRINT #1, "OUTPUT 4 ; #FCCA28"         ; Send Host Command for BV:28
PRINT #1, "OUTPUT 4 ; A"               ; Read
PRINT #1, "ENTER 4"                    ; IEEE for Input
INPUT #2, A$                           ; Get BV:28 value
PRINT A$                               ; Display BV:28 value
END                                    ; End of Program
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8.17.6. Retrieve Program from UNIDEX 100

The following program is a Quick Basic program used to get a program from the UNIDEX 100. Aerotech wrote this program for the National Instruments GPIB board using the Universal language Interface "HP-Style Calls". This program will acquire program No. 2 from the UNIDEX 100.

EXAMPLE:

```
/IEEE QBASIC PROG, GET PROGRAM #2
/INTERRUPT SERVICE SETUP & ENABLE
ON PEN GOSUB INTSRVC                    ; Set up serial poll interrupt
PEN ON                                  ; Enable Interrupt
/INITIALIZE THE INPUT & OUTPUT CHANNELS
OPEN "gpib0" FOR OUTPUT AS #1           ; IEEE output channel
OPEN "gpib0" FOR INPUT AS #2            ; IEEE input channel
/INITIALIZE THE BUS & RESET TO DEFAULT PARAMETERS
PRINT #1, "ABORT"                       ; Initialize IEEE
PRINT #1, "RESET"                       ; Reset IEEE
PRINT #1, "GPIBEOS IN CHR(/x7E)"        ; IEEE input string terminator EOF
PRINT #1, "GPIBEOS OUT CR"              ; IEEE output string terminator
/PLACE THE DEVICE IN THE REMOTE STATE
PRINT #1, "REMOTE 4"                    ; Remote device #4
PRINT #1, "CLEAR 4"                     ; Reset device 4
/GET PROG #2 FROM controller
A = 0                                   ; Clear serial poll interrupt flag
CLS                                     ; Clear screen
PRINT #1, "OUTPUT 4 ; #FDBA2"           ; Host command to get program #2
PRINT "WAIT FOR INTERRUPT"              ; Display wait for interrupt
WHILE A = 0                             ; Loop till serial poll performed
WEND                                    ; Loop back to previous line
PRINT #1, "GPIEBOS IN CHR(/x07E) CHR(/x07E)"  ; Change terminator to 2 EOF's
PRINT #1, "ENTER 4"                     ; Enable input
INPUT #2, P$                            ; Read in program #2
PRINT P$                                ; Display program
END                                     ; End of Main Program
INTSRVC:                                ; Serial poll interrupt subroutine
PRINT #1, "SPOLL 4"                     ; Do serial poll
INPUT #2, SP%                           ; Get serial poll
A = 1                                   ; Set serial interrupt flag
PRINT #1, "STATUS"                      ; Request IEEE status
INPUT #2, IBSTA%, IBERR%, IBCNT%        ; Get status
PRINT SP%                               ; Print status
RETURN                                  ; Return to Main Program
END                                     ; End of Entire Program.
```

∇ ∇ ∇

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CHAPTER 9: TECHNICAL DETAILS

In This Section:

- U100 Control Board and Power Board 9-1
- Motor Output Power (U100 Only) 9-4
- Limits and Primary Encoder Port (U100/U100i)..... 9-8
- Communications Port (U100/U100i)..... 9-14
- I/O Port..... 9-16
- Amplifier Connector (U100i) 9-20

9.1. U100 Control Board and Power Board

The engine of the U100 is the DSP56002 Digital Signal Processor (DSP), refer to Figure 9-1. The processor has a 16 bit address bus and a 24 bit external data bus. The DSP runs off a 20.48 MHz clock. If the DSP is coupled with single cycle instructions and parallel processing capabilities it becomes a very fast and powerful processor. The DSP has a built-in serial port that allows communication with external hardware according to the RS-232-C standard.

The DSP communicates with a dual 12 bit Digital-to-Analog converter (DAC). The DAC outputs the analog current commands used for motor control. From the DAC, the signals are sent to a Pulse Width Modulated (PWM) current regulator. The current regulator sends four PWM digital drive signals (labeled A, B, C, and D) to a power amplifier on the U100 power board (refer to Figures 9-1 and 9-2).

A shunt regulator on the power board prevents the bus power supply from charging up to a higher voltage caused by regenerative energy fed into the bus power supply during motor deceleration.

The U100 uses three 128 Kbyte EPROMs for permanent storage of its multi-tasking operating system. The DSP has 64 Kbytes of RAM available that is divided into two banks of 32 Kbytes each. One bank is battery backed (refer to Figure 9-1) and is used to store user programs, parameters, and variables. The other bank is used for system operations and is available for stack storage and storing the compiled version of an executing program.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

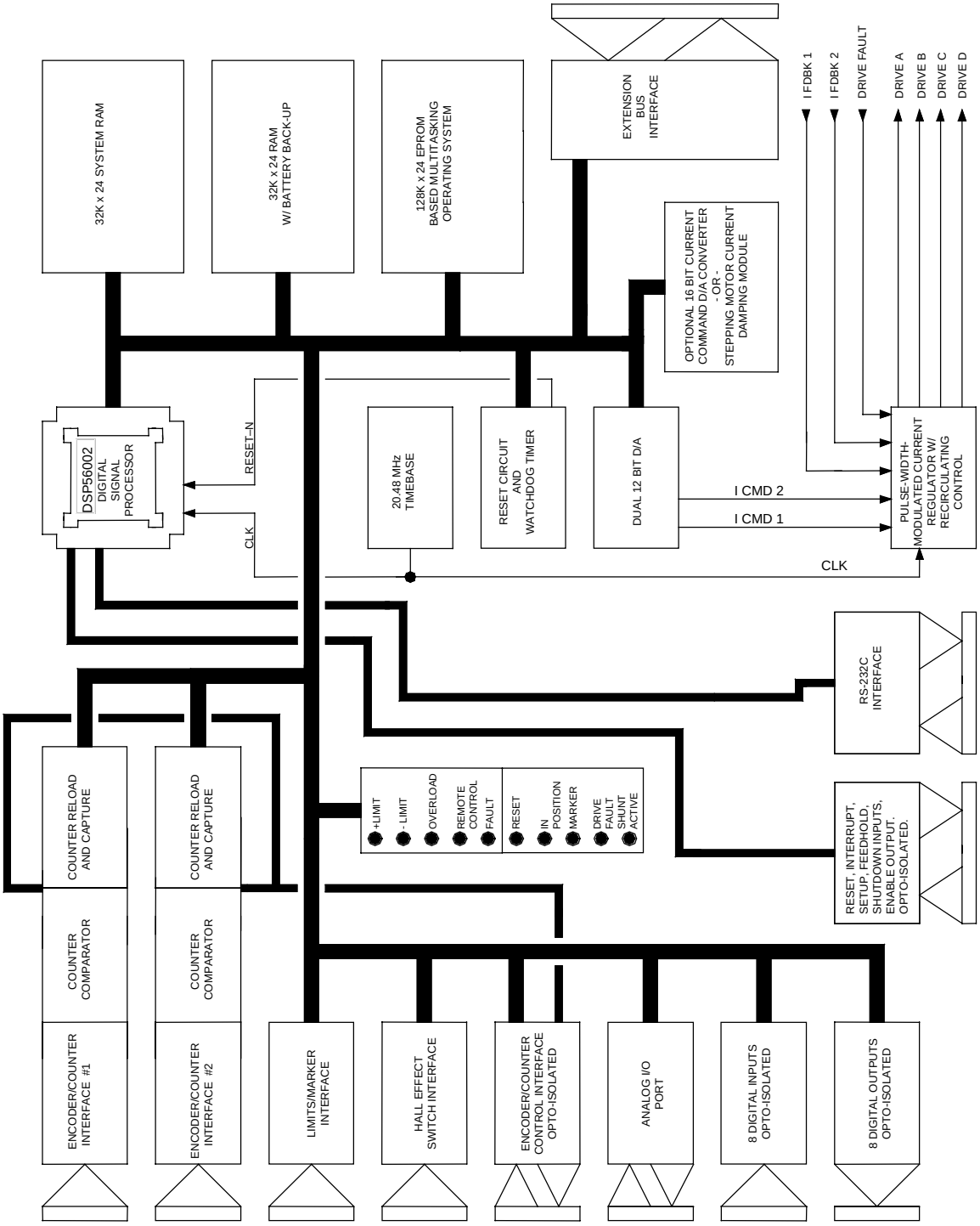


Figure 9-1. Control Board

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

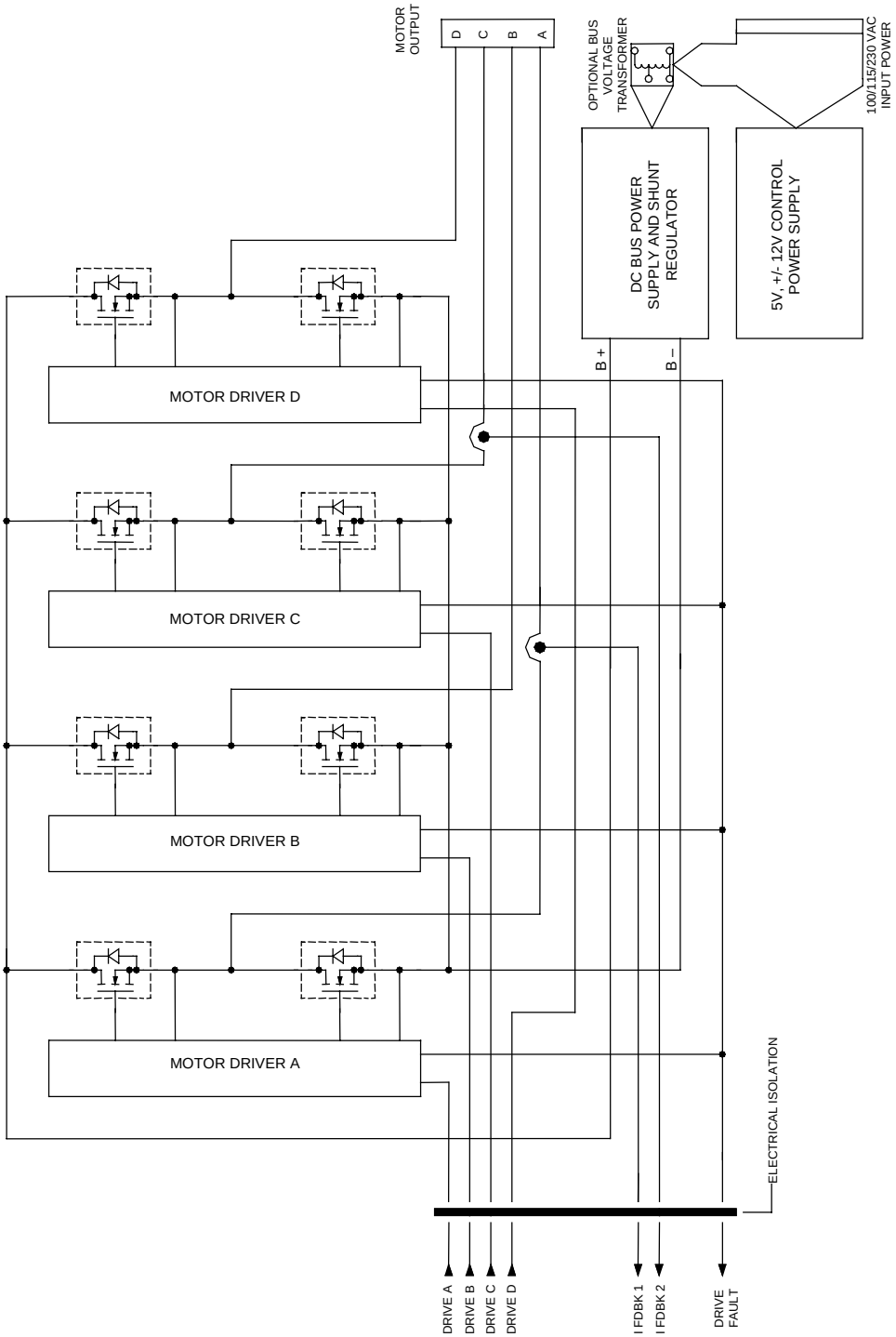


Figure 9-2. Power Board (U100 Controller Only)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9.2. Motor Output Power (U100 Only)

Output power to the motor comes from TB1 located on the front panel of the U100. Depending on the user's application, the U100 can power a DC brush motor, a stepping motor, and an AC brushless motor. Each of these power connections has its own wiring configuration.

The U100 is factory built to be a DC brush or stepper or brushless drive. Any one U100 cannot drive more than one type of motor.

9.2.1. DC Brush Motor Wiring

Two power connections are used for a DC brush motor configuration. Two additional connections are used for frame ground and motor shield. Figure 9-4 shows the output connections required for this application. Figure 9-3 shows an overview of a DC Brush motor configuration.

It is recommended that a shielded cable be used for the power connections. The purpose of a shielded cable is for safety reasons and the reduction of noise. General practice is to not connect the shield at the motor end of the cable, refer to Figure 9-4.

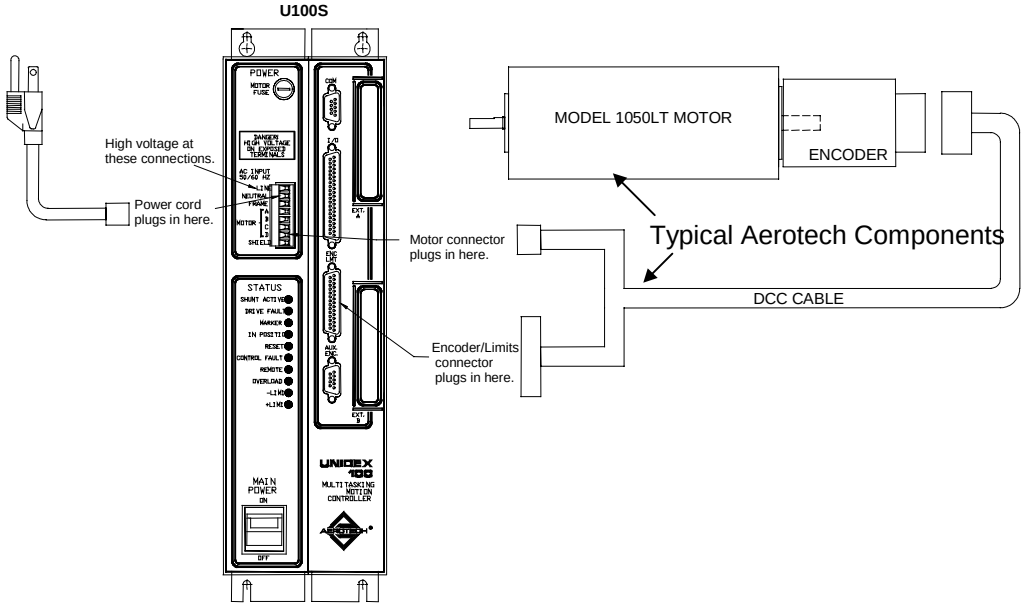
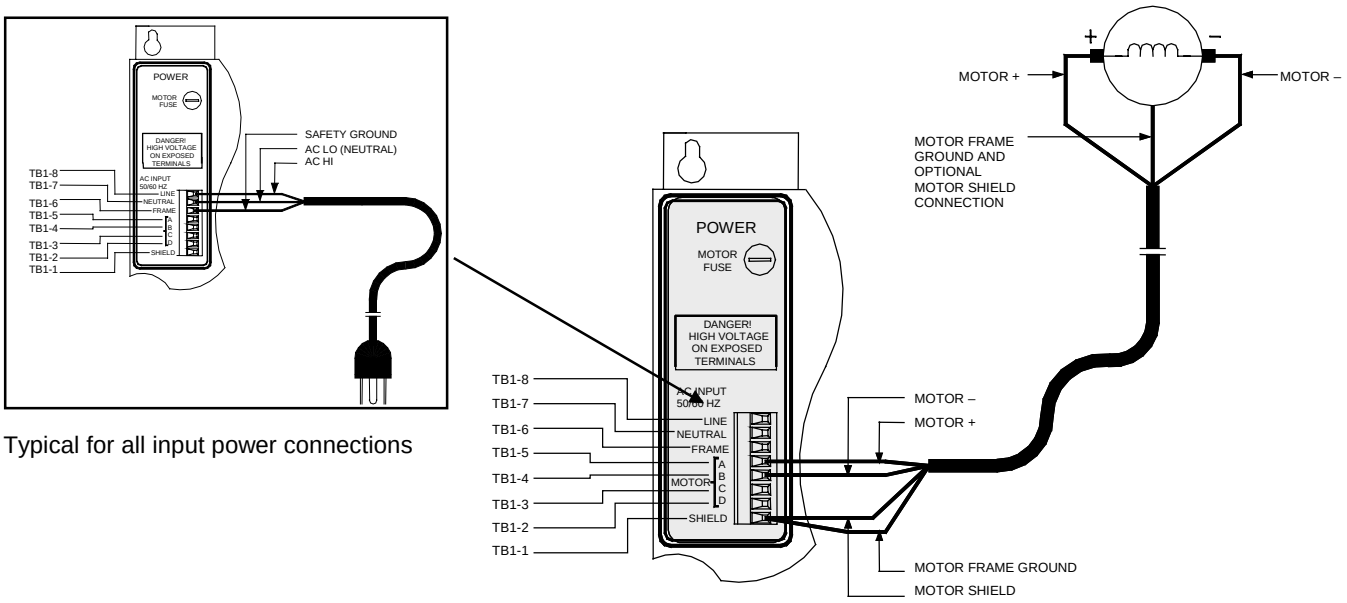


Figure 9-3. DC Brush Motor System

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



Typical for all input power connections

Figure 9-4. DC Brush Motor Wiring

9.2.2. Stepping Motor Wiring

A stepping motor contains two separate windings, therefore, this configuration uses all four motor connections. Like the DC brush motor configuration, the frame ground and motor shield are used. Figure 9-6 shows the output connections required for this application. Figure 9-5 shows an overview of a stepping motor configuration.

It is recommended that a shielded cable be used for the power connections. The purpose of a shielded cable is for safety reasons and the reduction of noise.



General practice is to not connect the shield at the motor end of the cable, refer to Figure 9-6.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

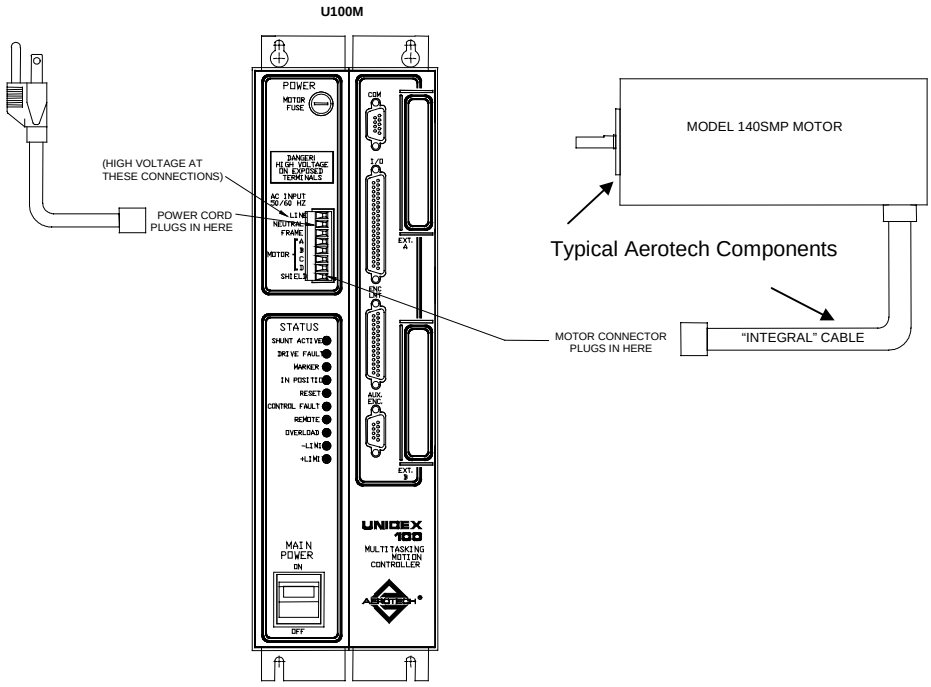


Figure 9-5. Stepping Motor System

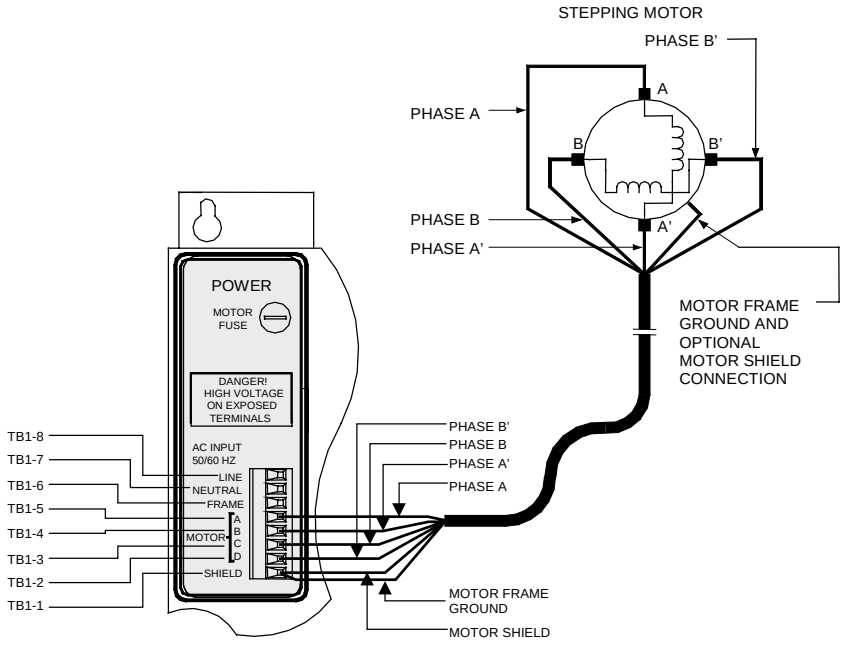


Figure 9-6. Stepping Motor Wiring

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9.2.3. AC Brushless Motor Wiring

AC brushless motors are driven with sinusoidal current commands that provide a smoother operation of the motor. Like the other two configurations the frame ground and motor shield are used. Figure 9-8 shows the output connections required for this application. Figure 9-7 shows an overview of an AC brushless motor configuration.

It is recommended that a shielded cable be used for the power connections. The purpose of a shielded cable is for safety reasons and the reduction of noise.

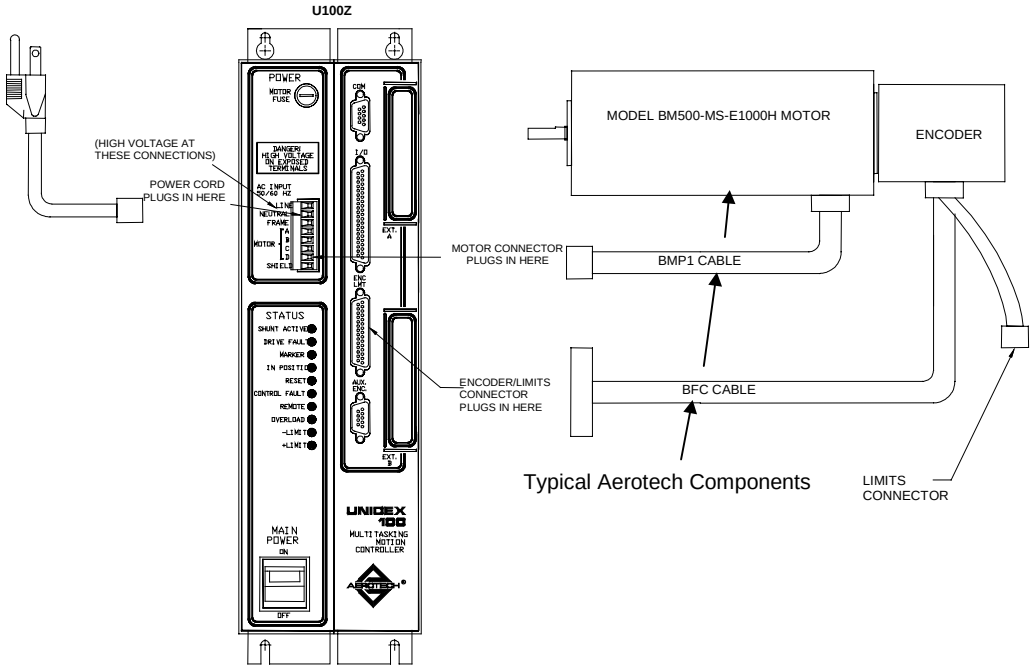


Figure 9-7. AC Brushless Motor System

General practice is to not connect the shield at the motor end of the cable, refer to Figure 9-8.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

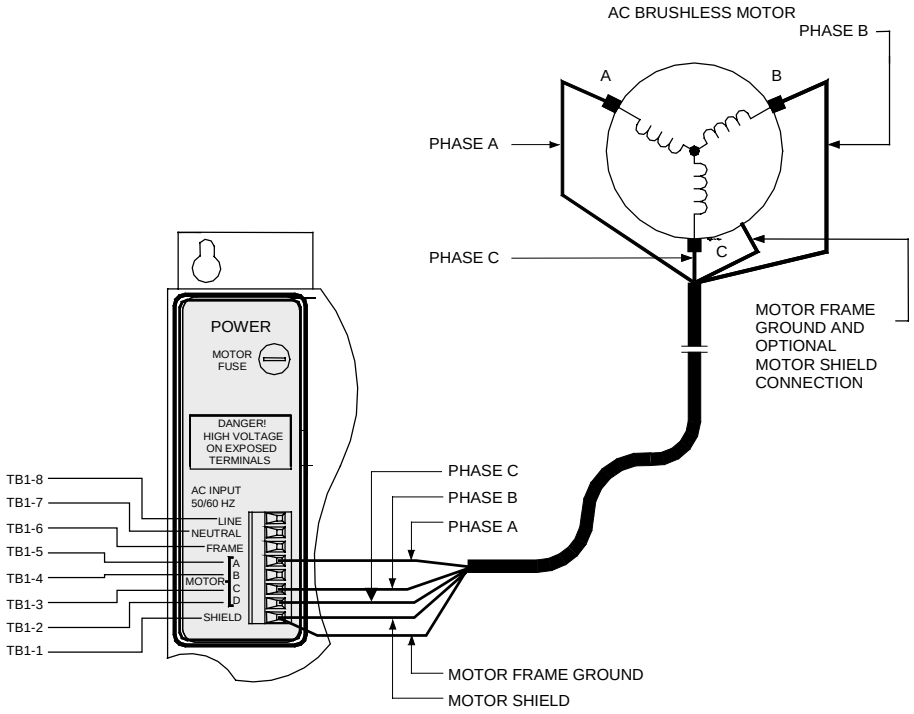


Figure 9-8. AC Brushless Motor Wiring

9.3. Limits and Primary Encoder Port (U100/U100i)

The limits and primary encoder port is a 25 pin "D" style connector that contains all the feedback inputs to complete a servo loop (refer to Figures 9-9 and 9-10). This port contains inputs for a 3 channel encoder, 3 limit switches, and 3 Hall effect devices that provide feedback for the microprocessor controlled position and velocity loops.

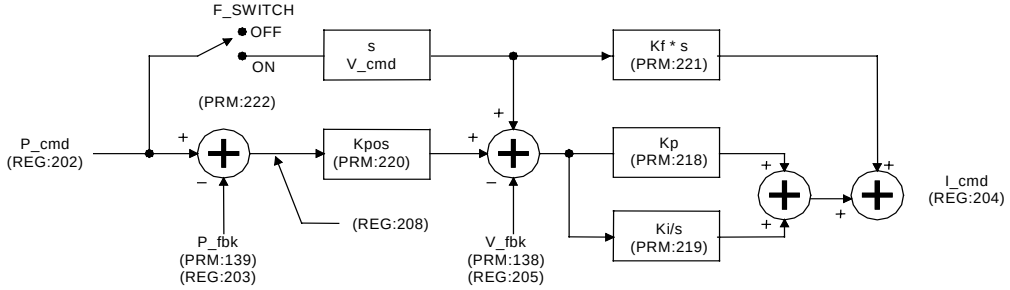


Figure 9-9. U100/U100i Servo Loop

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

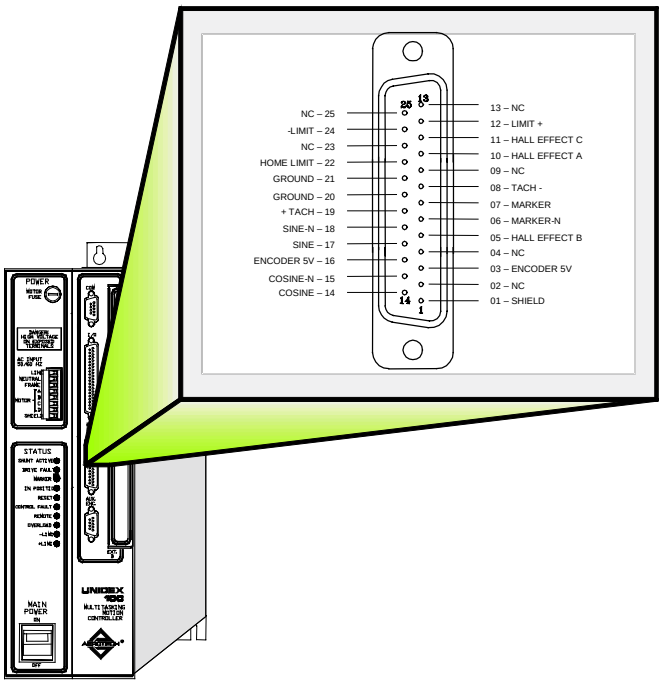


Figure 9-10. Limits/Primary Encoder Port (P3)

9.3.1. Encoder Interface

The controller incorporates two encoder interfaces, the primary encoder interface being P3 (the limits/primary encoder port). The second encoder interface plugs into P4 (the auxiliary encoder port), refer to Figure 9-11. The controller will use the primary encoder inputs to close both position and velocity loops in the servo loop. However, the controller offers versatility by providing an auxiliary encoder port. In a highly compliant system, the primary encoder can be used for position feedback and the auxiliary encoder for velocity feedback.

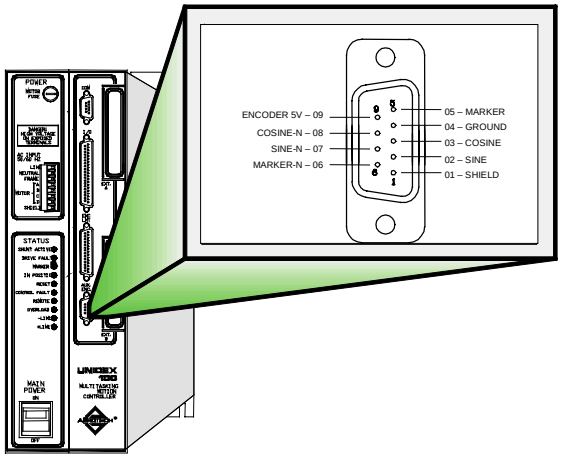


Figure 9-11. Auxiliary Encoder Port (P4)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Depending upon the setting of two parameters (PRM:138 and PRM:139) the functions of the encoder ports can be swapped. See Chapter 6: Parameters, for more information on these parameters.

The hearts of these two interfaces are two 24 bit multi-mode counters (see Figure 9-1). These counters accept square wave signals from a line driver differential output encoder. Figure 9-12 illustrates encoder phasing with relation to motor rotation.

Additionally, as part of each encoder interface, there are inputs for marker signals that are received by a differential input comparator circuit. These signals are standard on incremental encoders and provide a reference point for the mechanical system driven by the controller.

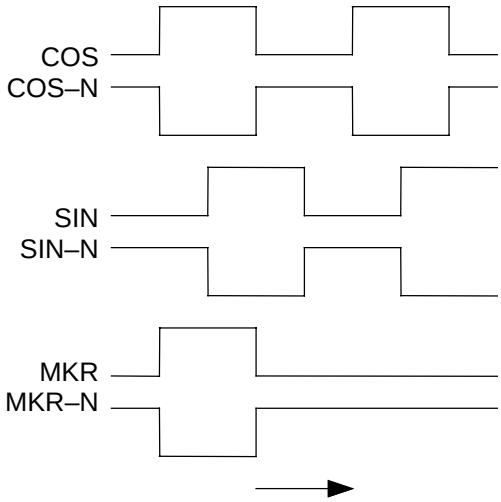


Figure 9-12. CW Motor Rotation (Viewed from Mounting Flange End)

The critical aspect of encoder phasing is COS is already at a logic high state when SIN is making a low to high transition during CW motor rotation.

9.3.2. Limits Interface

Three limit inputs are incorporated into the controller, the first two are end of travel limits and the third is the home limit, refer to Figure 9-13. The end of travel limits are designated as “+limit” and “-limit”. Clockwise (CW) motor rotation is inhibited by the “+limit”, while counterclockwise (CCW) rotation is inhibited by the “-limit”.

Viewing the motor from the mounting flange end of the motor determines motor rotation. Refer to Figure 9-14.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

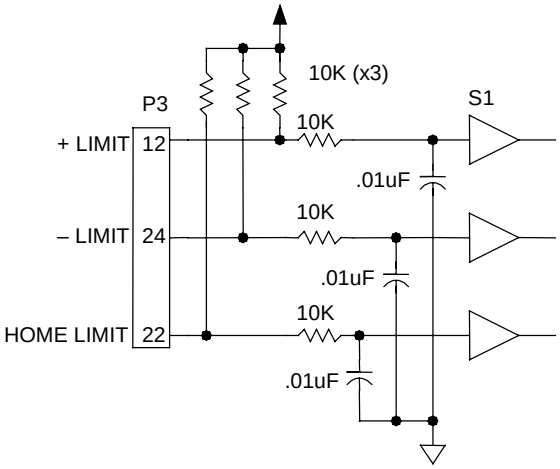


Figure 9-13. Limit Switch Input Circuit

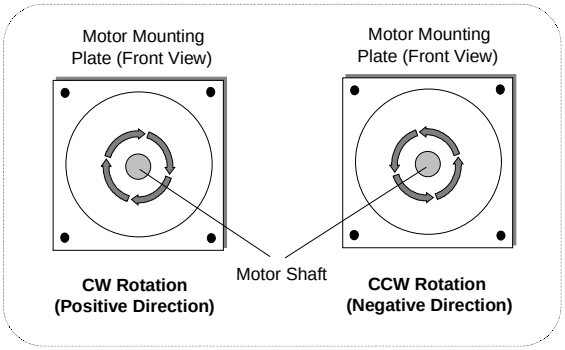


Figure 9-14. Motor Rotation

The home limit acts as a reference point so that motion may always begin from the same physical point in a system. When the controller receives a "Home" command, the system moves the motor in a preset direction in search of a home limit activation. When the controller finds the home limit, the motor reverses direction and stops on a preset occurrence, generally the first occurrence of the encoder's marker signal after the home limit has deactivated.

There are two choices to consider when wiring the three limits: active high (normally closed limit switch) or active low (normally open limit switch). To establish limit polarity with the controller, follow procedures for setting parameters in Chapter 6: Parameters.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9.3.3. Hall Effect Interface

AC brushless systems are the only systems that use the Hall effect inputs on P3 of the controller. The Hall effect switches form a digital representation of the motor phasing that the microprocessor uses to commutate the motor. This digital representation comprises three 5 volt signals that form a digital six step pattern. The controller receives these signals with a circuit that is electrically identical to the limit input circuit, refer to Figure 9-15.

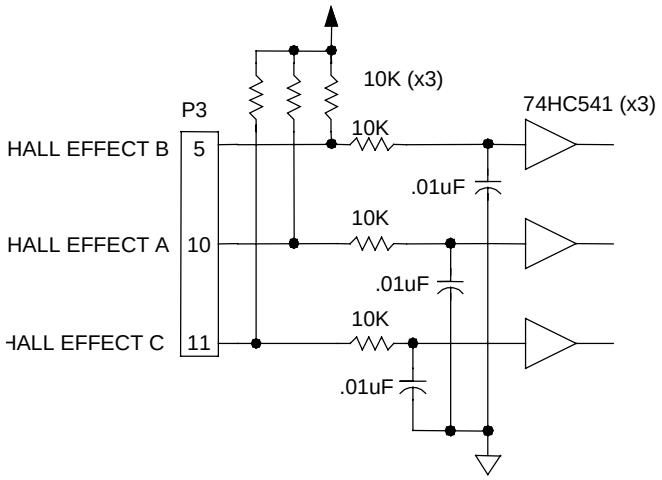


Figure 9-15. Hall Effect Input Circuit

Aerotech brushless rotary motors are shipped from the factory with the correct motor phase to Hall effect device relationship established. This is essential for proper motor commutation. The parameter that affects this relationship is PRM:241 and is set at "-42", refer to Figure 9-16. Figure 9-16 illustrates the proper Hall effect and motor phasing for both CW and CCW motor rotation (viewed looking into the mounting flange of the motor).

During CW motor rotation each Hall effect signal is at a logic low state when its corresponding motor phase is at a positive voltage. During CCW rotation each Hall effect signal is at a logic high state when its corresponding motor phase is at a positive voltage.

The waveforms created by the motor phasing can be observed using a wye resistor network, a dual trace oscilloscope, and performing the following steps.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

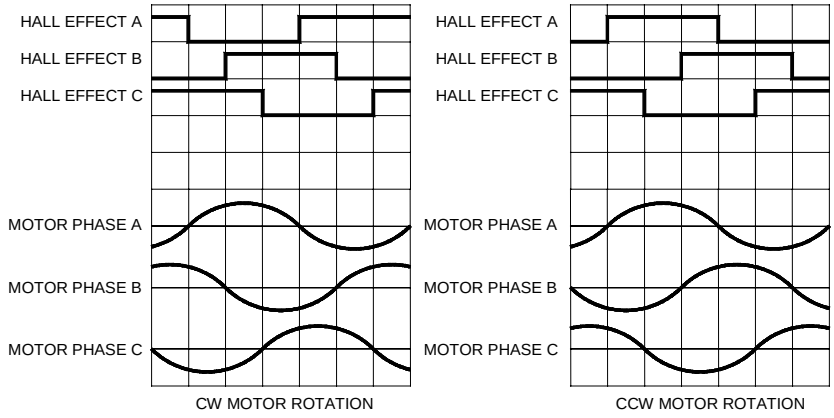


Figure 9-16. Hall Effect and Motor Phasing

The following procedure is only necessary if the user has to replace a defective encoder, or are using their own motors.

Motor voltage is monitored without power applied to the motor.

Before performing the following steps, remove all connections to the motor except the wye resistor network. Remove all mechanical connections to the motor shaft.

WARNING

1. Connect the ends of three resistors to motor terminals A, B, C. Refer to Figure 9-17.
2. Use one channel of the oscilloscope to monitor motor terminal A with respect to the “Wye” neutral (the point where all resistors are connected together), refer to Figure 9-17.
3. Connect a 5V power supply to the power pins of the encoder, refer to Figure 9-18.
4. Connect the second channel of the oscilloscope to Hall effect A on the encoder.
1. Once the oscilloscope is connected with one probe monitoring a motor phase and the other monitoring a Hall effect signal, rotate the motor by hand using the motor shaft. The motor will generate a voltage upon rotation.
2. Observe the phase relationship of the motor phase to the Hall effect signal.

It is necessary for the voltage generated by the motor phase A to be in phase with signal used as Hall effect A, refer to Figure 9-16.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

7. Move the probe on the Hall effect line to the other two Hall effect lines, observing their phase relationship to the motor voltage.
8. Repeat this process for the other two motor phases

Using this procedure, connect the Hall effect signals with the appropriate motor phases as shown in Figure 9-16.

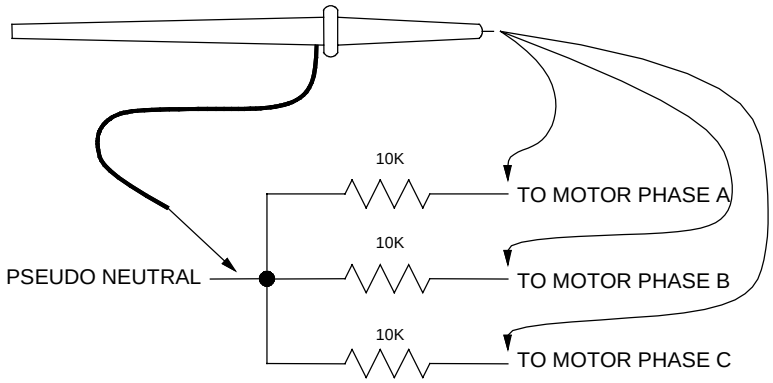


Figure 9-17. Motor Phase Voltage Observation Scheme

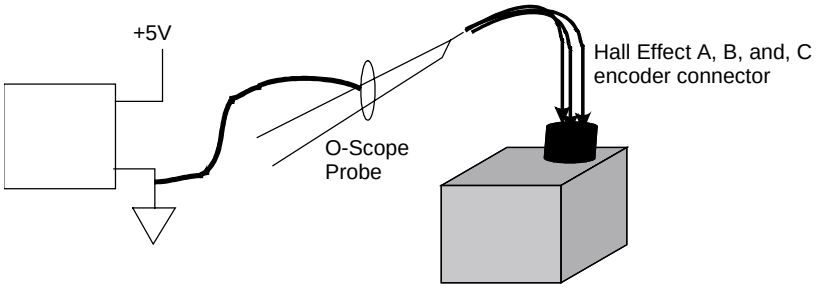


Figure 9-18. Encoder Phase Voltage Observation Scheme

9.4. Communications Port (U100/U100i)

The communications port is a 9 pin "D" type connector that contains the signal lines allowing the controller to interface with an external host. The COM port is located on the front panel of the controller and is labeled as such, refer to Figure 9-19. The controller communicates with on-board and external hardware according to the RS-232-C standard. However, the controller can be configured to serve as a RS-422-A interface (must be factory configured).

The standard RS-232-C interface operates with ±12V signals and Figure 9-20 illustrates the RS-232-C circuit. The optional RS-422-A interface uses differential 5 volt logic level signals. Figure 9-21 represents the circuit used for this interface.

The advantage of the RS-422-A interface over the RS-232 is it provides higher noise immunity, eliminating oscillations and ensuring a cleaner signal.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

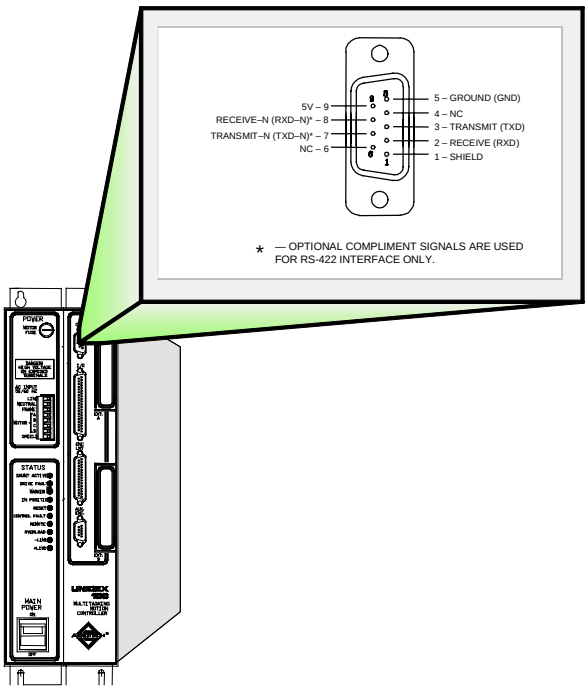


Figure 9-19. Com Port (P1)

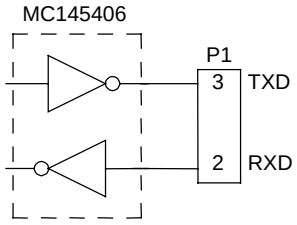


Figure 9-20. Standard RS-232-C Interface

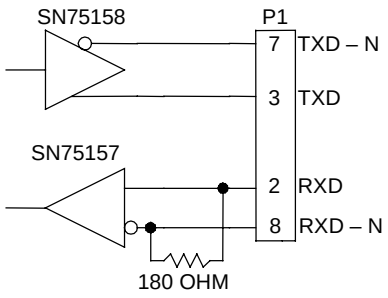


Figure 9-21. Optional RS-422-A Interface

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

9.5. I/O Port

The I/O port (P2) is a 37 pin "D" style connector containing all the general purpose I/O functions, refer to Figure 9-22. Table 9-1 list all functions available in the I/O port.

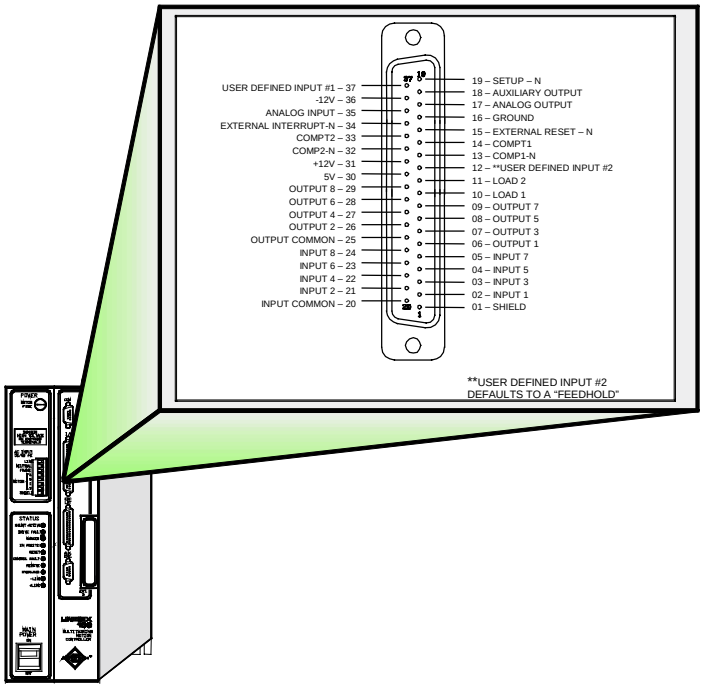


Figure 9-22. I/O Port (P2)

Table 9-1. Available I/O Port Functions

Location	Designation	Function
Pin 2	Input 1	User programmable input
Pin 21	Input 2	User programmable input
Pin 3	Input 3	User programmable input
Pin 22	Input 4	User programmable input
Pin 4	Input 5	User programmable input
Pin 23	Input 6	User programmable input
Pin 5	Input 7	User programmable input
Pin 24	Input 8	User programmable input
Pin 6	Output 1	User programmable output
Pin 26	Output 2	User programmable output
Pin 7	Output 3	User programmable output
Pin 27	Output 4	User programmable output
Pin 8	Output 5	User programmable output

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 9-1. Available I/O Port Functions Cont'd

Location	Designation	Function
Pin 28	Output 6	User programmable output
Pin 9	Output 7	User programmable output
Pin 29	Output 8	User programmable output
Pin 35	Analog Input	±10V Input; 8 bit resolution
Pin 17	Analog Output	±10V output; 8 bit resolution
Pin 10	Load 1	Position latch input for the primary encoder
Pin 11	Load 2	Position latch input for the auxiliary encoder
Pin 34	External interrupt	Active low input for interrupt operations (5µsec acknowledge time)
Pin 19	Setup input	Active low input for resetting the default parameters
Pin 15	External reset	Active low input for resetting the system
Pin 18	Auxiliary output	Controlled by fault mask
Pin 12	User defined input #2	Active low input for suspending motion
Pin 37	User defined input #1	Active low input for shutting down system operation
Pin 14	Compt1	Under flow or bi-stable toggle output for the primary encoder
Pin 33	Compt2	Under flow or bi-stable toggle output for the auxiliary encoder
Pin 13	Comp1-N	Overflow or mono-stable toggle output for the primary encoder
Pin 32	Comp2-N	Overflow or mono-stable toggle output for the auxiliary encoder
Pin 30	5V	U100 power supply
Pin 20	Input common	Common reference for all digital inputs
Pin 25	Output common	Common reference for all digital outputs
Pin 31	+12V	
Pin 36	-12V	
Pin 16	Ground	
Pin 1	Shield	

9.5.1. Digital Inputs Specifications

The U100/U100i have 8 digital inputs that are accessed through register "REG:001" (see Chapter 7: Registers and Variables for more information on REG:001). The design of the input circuit uses a 5V power supply. Refer to Figure 9-23 for electrical characteristics of the input circuit. If opto-isolation is not necessary, 5 volts can be supplied from the U100 on-board supply.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

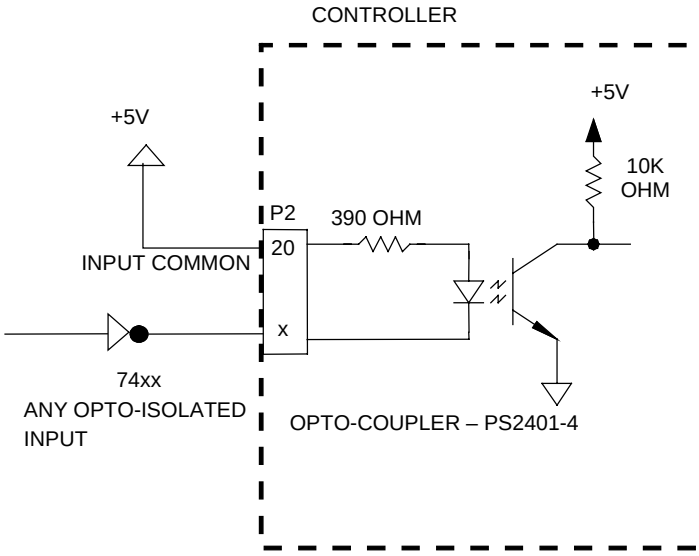


Figure 9-23. Electrical Characteristics of Opto-Isolated Input

All digital inputs must operate from the same power supply, since they all reference the same INPUT COMMON line (P2 - 20).

To add an external power supply that is greater than 9 volts, a resistor must be added to the input feed. Place the resistor in series with each input to limit the current to a safe 20 mA, refer to Figure 9-24. To calculate the resistor value refer to the following equations.

$$R_{total} = \frac{External\ Power\ Supply\ Voltage - 1.1}{.02}$$

$$R_{ext} \Rightarrow R_{total} - 390$$

$$R_{ext}\ Power\ Rating \Rightarrow .02^2 \times R_{ext} \times 2$$

Resistance (R) expressed in ohms.
Power rating expressed in watts.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

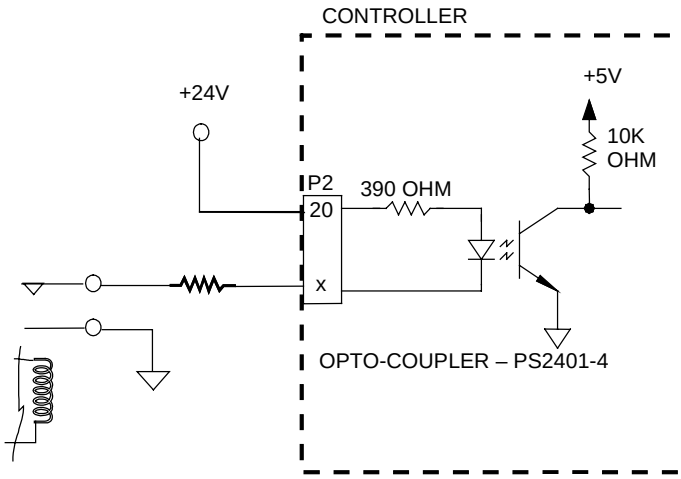


Figure 9-24. Electrical Characteristics of Opto-Isolated Input (w/External Power Supply)

9.5.2. Digital Outputs Specifications

The U100 has 8 digital outputs. Refer to Figure 9-25 for electrical characteristics of the opto-isolated output.

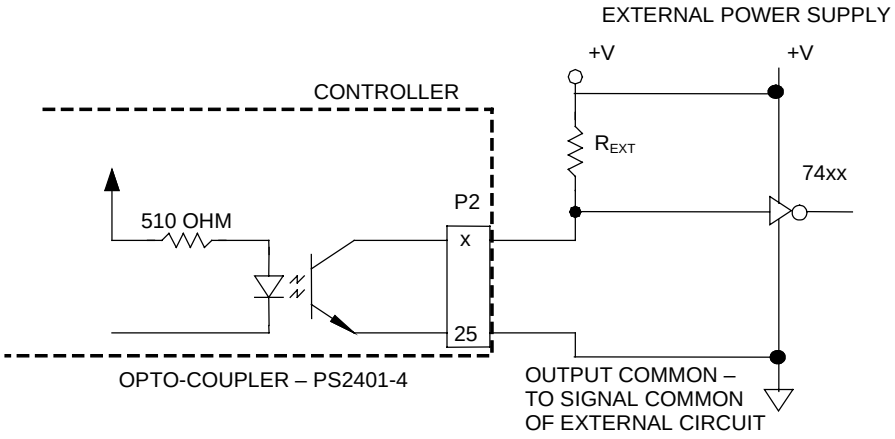


Figure 9-25. Electrical Characteristics of Opto-Isolated Output

All digital outputs must operate from the same power supply, since they all reference the same OUTPUT COMMON line (P2 - 25).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The following formula calculates the added pull-up resistor value for the external circuitry.

$$R_{e\ xt} \geq \frac{EXTERNAL\ POWER\ SUPPLY\ VOLTAGE - (^{V}OUTPUT\ COMMON + .4)}{.02}$$

$$R_{e\ xt}\ POWER\ RATING = \rangle .02^2 \times R_{e\ xt} x 2$$

Resistance (R) expressed in ohms.

Power rating expressed in watts.

Maximum external power supply voltage = 80 volts.

9.5.3. Analog Input and Output

The analog input and output are user programmable and accept signals within a voltage range of ±10 volts. The input signal contains 256 discrete increments (8 bit resolution) where each increment represents .078V. Since the analog output comes from an 8 bit device, the output signal also is in increments of .078V. The output has the capability to drive loads as low as 2K ohms input impedance.

9.6. Amplifier Connector (U100i)

The amplifier connector is only available on the U100i and provides the interface for input and output control connections between the U100i and Aerotech's BA Series amplifiers or similar. Refer to Figure 9-26 for the amplifier connector pinout.

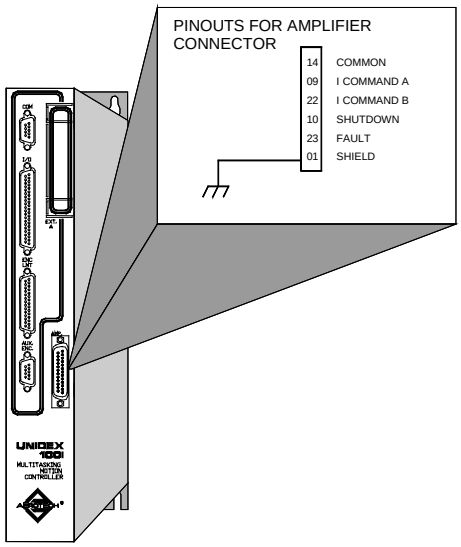


Figure 9-26. Amplifier Connector Pinouts

∇ ∇ ∇

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CHAPTER 10: TROUBLESHOOTING

In This Section:

- Installation, Startup and Communication Problems..... 10-2
- Motor and Related Problems..... 10-3
- Fault Conditions and Other Related Problems..... 10-5

If there are any technical support questions, please have the following information available before calling:

1. The customer order number.
2. We may also need to know the type of PC being used (brand name, CPU, available memory), the current version of DOS, and the contents of the AUTOEXEC.BAT and CONFIG.SYS files.
3. If developing own application, we will need to know what compiler and version number being used (e.g., Borland C v3.1, Microsoft Visual C, etc.).
4. If at all possible, try to be in front of the system where the problems are occurring.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

10.1. Installation, Startup and Communication Problems

Some common problems that relate to installation, startup and communications are listed and diagnosed in Table 10-1.

Table 10-1. Troubleshooting for Common Installation, Startup and Communication Problems

Problem	Possible Causes / Solutions	See Section
System does not power up at all.	System contains a blown AC line fuse (U100 only). The AC line fuse blows only under a severe fault condition. Before replacing the blown fuse with the correct value, verify all electrical connections to the system and disconnect all power from the unit. Refer to Chapter 3 for fuse locations and values. For the U100i, *refer to the applicable amplifier manual.*	3.4.2.
System does not power up. If system has fan, fan runs OK.	System contains a blown 5V supply fuse. Remove all external connections, if any, to the 5V terminal provided on P2-30. An external circuit may be overloading the 5V power supply. Replace the fuse. Refer to Chapter 3 for fuse locations and values. If the external circuitry is being connected to P2-30, it may be faulty, wrongfully connected, or too much of a load for the internal power supply. For the U100i, *refer to the applicable amplifier manual.*	3.4.2.
System does not come up.	This may result from installation of new software or corrupted memory. Perform the SETUP function. Jumper P2-19 (setup-n) to P2-16 (common). Jumper P2-20 (input common) to P2-30 (5V) ONLY if there is no connection from P2-20 to an external power supply. Once the jumpers are in their correct locations, power up the unit. Allow unit to remain powered up for at least 5 seconds. Power down and remove jumpers.	4.4.
System does not come up and/or appears to be operating abnormally.	JP20 on control board is missing or is not in 2-3. Install JP20 on control board 2-3.	3.3.4
Cannot store programs.	This may result from installation of new software or corrupted memory. Perform SETUP function as described above and/or clear memory.	4.4.
Cannot download programs.	This may result from installation of new software or corrupted memory. Perform SETUP function as described above and/or clear memory.	4.4.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 10-1. Troubleshooting for Common Installation, Startup and Communication Problems (Cont.)

Problem	Possible Causes / Solutions	See Section
System resets at approximately 0.5 second intervals.	5V power supply has too much of a load on it. Remove external circuits being powered by P2-30. It is possible that these circuits need powered by an external supply.	
There is no evidence of RS-232-C communication.	The setup calls for RS-422-A communication. Set jumpers on control board for RS-232-C communication.	
There is no evidence of RS-422-A communication.	The setup calls for RS-232-C communication. A standard configuration includes RS-232-C communication and NOT the RS-422-A. The control board may not have the necessary chips installed for RS-422-A operation. This must be specified when the unit is ordered.	
There is no evidence of RS-232-C communication.	The settings for the device used with the system do not match the settings of the RS-232-C communication parameters. Change the RS-232-C parameters to match the U100 communicating device OR perform the SETUP function described (see Chapter 4). Doing this resets the U100 default parameter values.	4.4.

10.2. Motor and Related Problems

Some common problems that relate to the use of the motors are listed and diagnosed in Table 10-2.

Table 10-2. Troubleshooting for Motors (and Related) Problems

Problem	Possible Causes / Solutions	See also...
The motor has no torque, trips out when commanded to move, or shows irregular motor movement	Drive mode jumpers JP16 and JP17 on control board contain improper settings for the type of motor being used. Set jumpers according to motor being used	Chapter 3 3.3.3
No torque on motor	Motor fuse (located on the front panel) is blown (U100 only) Replace motor fuse with appropriate sized fuse for motor being driven. For the U100i, *refer to the applicable amplifier manual*.	3.4.2.
An AC brushless motor has no torque	Wrong setting for the brushless commutation factor based on the motor type. Change value of PRM:239 to correct value (1, 2, or 3) for the given motor.	Chapter 6
An AC brushless motor has no torque	Hall effect switches not connected to P3-5, P3-10, and P3-11. It is necessary to use Hall effect switches to commutate a brushless motor that uses an encoder for providing position feedback. If the position feedback encoder connects to P4, it is still necessary to connect the Hall effect switches to P3.	

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 10-2. Troubleshooting for Motors (and Related) Problems Cont'd

Problem	Possible Causes/Solutions	See Also
Cannot set the stepper running current (PRM:201) to the desired value. Cannot set the stepper holding current (PRM:202) to the desired value. Cannot set the servo peak current limit (PRM:216) to the desired value. Cannot set the servo RMS current limit (PRM:217) to the desired value.	Settings for jumpers JP13, JP18, and JP19 on the Control Board provide a maximum current less than the desired current. Verify the current ratings of the motor in use and set jumpers JP13, JP18, and JP19 on control board to the appropriate value (up to 20 Amps). Then, set the desired parameter(s) to the desired value. Refer to Chapter 3 for jumper information. Do not exceed motor ratings.	3.3.2
Using a DC brush motor in a velocity loop causes the motor to run away.	Wrong configuration for JP8 on the control board. Install jumper JP8 on control board 2-3.	3.3.1
Using a DC brush motor in a velocity loop causes the motor to run away.	Wrong tachometer polarity for the type of motor in use. Reverse the tachometer connections to P3. For the U100i, *refer to the applicable amplifier manual.*	Chapter 9
The motor has no torque.	Improper connection of the encoder or no connection at all. Verify the connection to the encoder and the phasing. For the U100i, *refer to the applicable amplifier manual.*	Chapter 9
The motor has no torque.	Configuration of the system parameters are for the wrong encoder port. Correct the parameter configuration based on the encoder port being used.	Chapter 6
The motor is without torque when using an ENC board or an R/D board for position and/or velocity feedback.	Wrong configuration for the systems ENC or R/D. Set up the parameters so that the ENC and/or R/D provide for position and/or velocity feedback. Refer to the parameter section	Chapter 6
The motor contains no torque or the drive fault LED is on.	Short circuit across the motor leads. Verify all electrical connections to the motor. For the U100i, *refer to the applicable amplifier manual.*	Chapter 9
The motor has no torque or the drive fault LED is on.	System contains a blown shunt fuse F4 on power board caused by excessive regeneration from the motor. Replace the shunt fuse with the appropriate value. Refer to Chapter 3 for fuse information. If necessary, adjust potentiometer R2, located near F4, so that shunt turns on when bus regeneration is 15% higher than the bus voltage. For the U100i, *refer to the applicable amplifier manual.*	Section 3.4.2.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

10.3. Fault Conditions and Other Related Problems

Some common problems relating to fault conditions and other related problems are listed and diagnosed in Table 10-3.

Table 10-3. Troubleshooting for Fault Conditions and Other Related Problems

Problem	Possible Causes / Solutions	See Also
System resets at approximately 0.5 second intervals.	5V power supply has too much of a load on it. Remove external circuits being powered by P2-30. It is possible that these circuits need powered by an external supply.	
System does not retain programs after being powered down.	Battery is defective (U100 only). Replace the battery.	
No function being performed when using encoder counter to capture data and/or fire outputs.	The encoder counter in use is already being used by the system software to provide position and/or velocity feedback. Use the encoder counter NOT used for position or velocity feedback to perform miscellaneous functions.	
No function being performed when using encoder counter to capture data and/or fire outputs.	No encoder connected to the encoder port for the desired counter. Connect the encoder to the encoder port to establish miscellaneous functions. Use the same encoder that provides position and/or velocity feedback, or a separate encoder.	
No function being performed when using encoder counter to capture data and/or fire outputs.	Improper programming of encoder counter. Consult Chapter 8 for encoder counter programming instructions and examples.	Chapter 8 Section 8.2.14.
System faults when motion is commanded. (Drive LED illuminates)	Bus voltage is dropping to less than 25VDC. This is most typical when using a 40V bus. Verify that maximum speed of the motor is not being exceeded.	
Overload LED comes on.	The RMS current limit set in PRM:217 is being exceeded when the motor is running. This may have a variety of effects on the system depending on the setups of the fault masks. Raise the value of PRM:217 to match the RMS current limit of the motor in use. If PRM:217 matches the RMS current limit of the motor in use, the system exceeds the capabilities of the motor.	Chapter 6

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 10-3. Troubleshooting for Fault Conditions and Other Related Problems

Problem	Possible Causes / Solutions	See Also
Control fault LED comes on.	Violation of a condition(s) set in the fault mask. The system may present a wide variety of additional symptoms depending on the specified actions in the fault masks for different fault conditions. Closely examine the system configuration (e.g., limit polarity, feedback source(s), RMS current limit) and make certain that what represents an error in the fault masks also corresponds to what is a true error in the system configuration. Either the fault masks settings are incorrect for the system or there is an actual error being detected by the fault masks.	
System cannot read programmable inputs 1-8.	No power supply connected to opto-isolators (U100 only). Connect the P2-20 (input common) to a 5V power source whose common connects to P2-16 (ground). This powers the opto-isolators on the control board. Refer to Chapter 9 for alternative input common connections.	
Programmable outputs 1-8 cannot be activated.	There is no connection between the power supply and the output opto-isolators (U100 only). Connect the common from the external power supply to P2-25 (output common). Add a pull-up resistor to each used output to the supply line of the external supply. Refer to Chapter 9 for specifications on the external power supply and pull-up resistors.	
System LEDs indicate a limit condition although system is not in limit.	Polarity of hardware and/or software limits due to incorrect setting of PRM:317 (invert mask). In PRM:317, active low hardware limits must have bits 8 and 9 set to a 1. Active high hardware limits must have bits 8 and 9 set to a 0.	Chapter 5 Section 5.5.1. Chapter 6 Section 6.6.18.

∇ ∇ ∇

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CHAPTER 11: PID LOOP TUNING

In This Section:

- Introduction 11-1
- PID Gain Parameters 11-2
- Tuning the PID Loop 11-3
- Parameters and the Effect of Servo Loop Performance 11-7

11.1. Introduction

This chapter explains the adjustment of the motor position loop to obtain optimum performance. The PID Loop diagram shown in Figure 11-1 indicates the location of the parameters necessary to adjust the loop. The following sections discuss how to tune the PID Loop.

Only review this section if the controller is controlling a servo motor (e.g., Brush DC or Brushless AC).

TUNE_100.EXE, a UT100 utility program for Windows can be used a as an aid in tuning the PID loop.

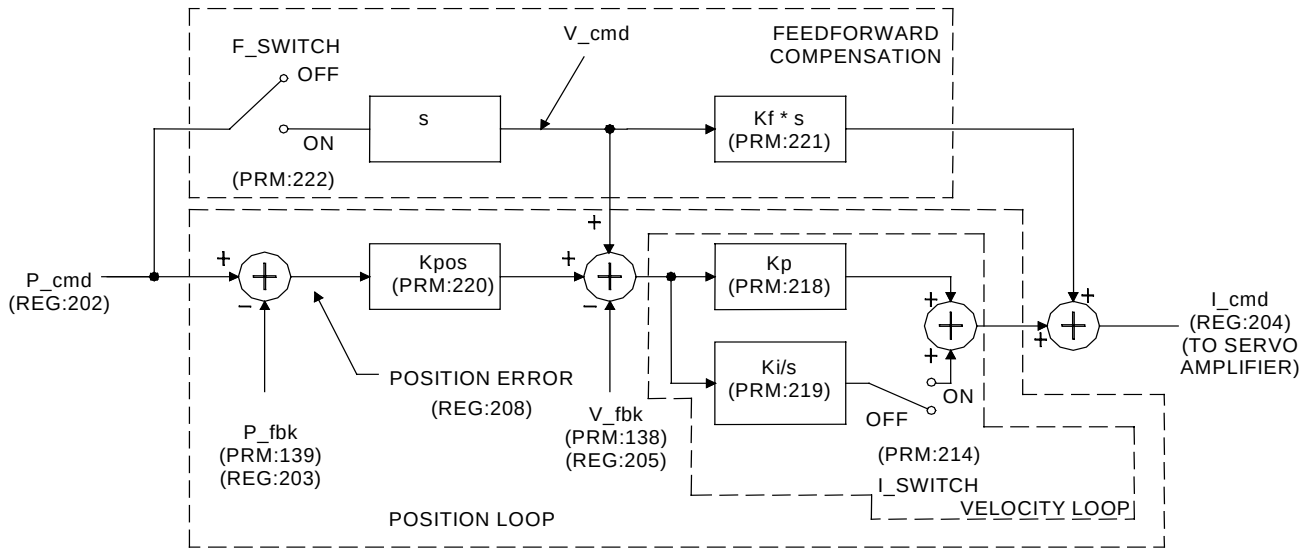


Figure 11-1. U100/U100i PID Loop

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

11.2. PID Gain Parameters

The Servo Control Loop block diagram shown in Figure 11-1 is a special form of classical PID (Proportion, Integral, Derivative) control. This control is the dual loop control (nested loop control) whose form is one of the most stable and easy to configure of all the PID control configurations.

Its inter-loop is composed of elements Kp and Ki/S and are mainly responsible for velocity stability. The outer-loop contains the Kpos element and is responsible for position control.

In addition to the inter/outer loops, there are also elements for velocity and acceleration feedforward compensation.

Table 11-1 provides a brief explanation of the individual elements of the PID loop shown in Figure 11-1.

Table 11-1. PID Loop Individual Element Descriptions

Element	Description
F_switch	The “F_switch” is a hypothetical switch controlled by PRM:222. When PRM:222 is set to “1”, the switch is closed and the “P_cmd” is processed through “S” to produce a “V_cmd”.
S	Represents differentiation. Meaning that any signal multiplied by this element produces an output signal that represents change of the input signal relative to time.
$\frac{1}{S}$	Term used to represent integration. Meaning that any signal multiplied by this element produces an output signal that “sums” the input signal over time.
I_switch	Controlled by parameter PRM:214. When PRM:214 is set to "0", this switch is closed at all times, and data processed through the gain block "Ki/S" is summed with other elements to produce the current command output signal "I_cmd".
Kpos	Denoted as the position loop gain and set by PRM:220. This parameter ranges between 0 and .999.
Kp	Denoted as the proportional velocity loop gain and set by PRM:218. The range of this parameter is between 0 and 8388607. This gain is ratiometric. The output of the gain block that contains this parameter is denoted in "amperes". This gain applies to proportional velocity error gain. To calculate the output, use the following gain calculation. **Kp * (Maximum Motor Current Setting) / 8388607**
Ki	Denoted as the integral velocity loop gain and is set by PRM:219. This parameter has the same range and gain characteristics as Kp. Note however, that this gain applies to the integration of velocity error instead of the proportional gain of velocity error.
Kf	Denoted as the acceleration feedforward gain. This gain is applied to the velocity command "V_cmd" to produce a "reference" current command (note that the V_cmd is zero if the F_switch is open).



In the calculation under Kp, the “Maximum Motor Current Setting” is in amps and is set by jumpers JP13, JP18, and JP19.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

11.3. Tuning the PID Loop

Tuning the PID loop is a relatively straight forward operation. The factory default settings of Kp, Ki, Kpos are set to provide adequate position response for most applications (e.g., they are usually adequate for ballscrew or leadscrew drive applications). However, for a direct drive application, adjustments may be necessary to stabilize performance.

The gains mentioned above revert to the factory default settings if a setup is performed on the controller.



The best method to manually adjust the Kp, Ki, Kpos gains is to perform the following steps.

1. Set "F_switch" to open (PRM:222 = 0) and the "I_switch" to closed (PRM:214 = 0).
2. Set PRM:138 (velocity feedback) and PRM:139 (position feedback) to the appropriate transducer feedback source(s).

When using transducers other than encoder 1 for position or velocity feedback, PRM:308, PRM:309, and PRM:310 are defaulted to zero (0). These parameters set the background scan time for encoder 2, ENC (Encoder Multiplication option), and R/D (Resolver-to-Digital converter option) respectively,

If applicable, set one of these three parameters to "3" (scan at .4 msec time period).



Before proceeding with the following adjustments, be aware of the hazards of the mechanical system being tuned.



Never grasp the motor shaft when applying power.



3. Set Kpos (PRM:220 = 0) and Ki (PRM:219 = 0). Keep Kp (PRM:218) at the default setting of "10000".

With Kpos set to zero (0), the position of the motor will drift. If this can not be tolerated, set Kpos to a small value of ".01".



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4. Turn on the U100 power switch, but be prepared to shut off power if the motor begins to spin.



If the motor begins to spin, its possible that the transducer providing the velocity feedback is not phased properly. (For most systems, the position and velocity feedback are derived from the same transducer).

5. Increase the value of Kp (PRM:218) until the load stiffens.



Setting the value of PRM:218 too high can cause oscillations to occur.

Also, high values of PRM:218 may cause a raspy sound to be emitted from the motor if it turns or is forced out of position. This sound is caused by digital quantization of the transducer supplying velocity feedback. It is necessary to set the value of PRM:218 to minimize the raspy sound, but still provide adequate stiffness to the motor load.

6. Slowly increase Kpos (PRM:220) and visually check position response each time this parameter is increased by commanding short indexed moves (through MDI mode).



With Ki (PRM:219) set to zero (0), it will take a long time for the “In Position” LED to energize. It may not energize at all if the load exhibits sufficient resistance.

7. In order to compensate for load friction, adjust Ki (PRM:219). Start with a small value, such as 10, then increase this value until the “In-Position” LED starts to activate (no more than a quarter of a second after the commanded move ends).



Too large a value will cause the motor to oscillate.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The dynamic performance of the motor can be checked by loading and executing two small programs (PGM files) shown below. Since the controller has a standard D/A output, the velocity feedback (REG:205) can be viewed in an analog form by feeding this register through the "DAC()" command.

8. Run the following program in Task 2.

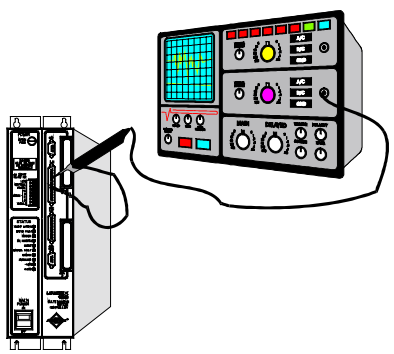
```
BEGIN          ; Start the program.
PRM:142=100    ; This parameter sets the scale factor for the D/A
               ; output.  This number may have to be adjusted
               ; relative to the resolution of the transducer
               ; feedback for velocity and/or the velocity in
               ; which the motor is running.
WHL(1EQ1)      ; Run the statements below, indefinitely.
SYNC           ; Insure that the D/A is updated every 6.4 msec.
DAC(REG:205)   ; Transfer the contents of the velocity feedback
               ; register to the D/A.  With PRM:142 set to 100
               ; above, ± 100 counts per servo cycle will produce
ENDWHL         ; ±10 volts on the D/A output via PRM:304.
END            ; End of the program.
```

a) Run the following program in Task 1.

```
BEGIN          ; Start the program.
WHL(1EQ1)      ; Run the statements below, indefinitely.
D(40000)       ; Cycle the motor back and forth.  Note that the
V(33333)       ; velocity and distance command may be altered
T(.2)          ; based on the resolution of the feedback devices
GO             ; and/or limitations of the connected load.
DW(1.0)
D(-40000)
GO
DW(1.0)
ENDWHL
END            ; End of the program.
```

The program running in Task 2 provides a representation of motor velocity at the D/A output (pin 17 of the I/O connector P2).

9. With the use of an O-scope and the probe connected to P2 - pin 17 and the reference connected to P2 - pin 16 view the velocity feedback signal.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Figure 11-2 shows the representation of acceptable and unacceptable velocity feedback signals. The unacceptable side illustrates velocity overshoot. Usually, the reduction of Kpos (PRM:220) causes a reduction in overshoot. However, this is only possible if Kp (PRM:218) and Ki (PRM:219) were adjusted properly as just described.

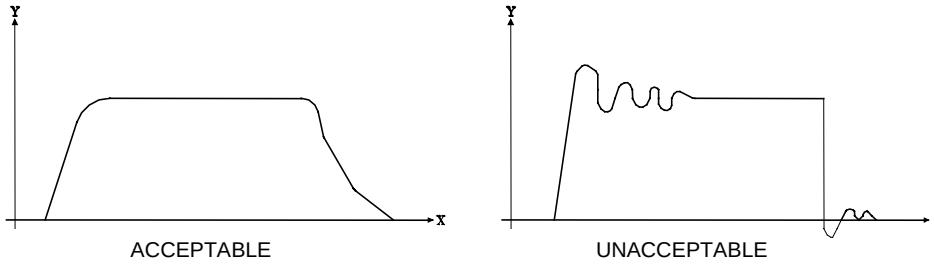


Figure 11-2. Velocity Feedback Signals

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

11.4. Parameters and the Effect of Servo Loop Performance

It is important to understand that the setting of other parameters can affect Servo Loop performance.

When performing a setup on the controller, the setting for Peak Current Limit (PRM:216) and Current Limit Trap (PRM:217) default to relatively low values. If PRM:216 is set to low, limit cycling can occur. Depending upon the magnitude of the programmed velocity and distance, this effect can sometimes resemble a poorly tuned Servo Loop.

For large inertia loads the default servo update rate of .8 msec (PRM:304) may be too fast. This can be a problem if the resolution of the velocity feedback transducer is relatively low. It has been shown in some cases that for large inertia loads, turning the servo update time up to 3.2 msec actually improves servo loop performance. Usually, .8 msec suffices for most applications. Also, for high inertia loads, it is desirable to enable the "I_switch" to stop integration of the velocity error during motion. This can be done by setting PRM:214 to “1”. This minimizes overshoot when reaching the position endpoint.

The setting of feedforward compensation can affect servo loop tracking of the position command. To enable velocity feedforward compensation, set PRM:222 to “1”. Velocity feedforward has the effect of canceling position error at a constant velocity (position error is the difference between the command position and the feedback position). Usually, for point-to-point applications, position error is not detrimental.

It should be noted that it is easier to minimize position overshoot with velocity feedforward disabled.



The only time the user can adjust acceleration feedforward (PRM:221) is if the “F_switch” is closed (e.g., PRM:222=1). Adjusting acceleration feedforward is effective only if the resolution of the position and velocity feedback transducer(s) is high. Typically, 500 line encoders in "X4" mode (e.g., 2000 step/rev. resolution) have insufficient resolution to allow introduction of acceleration feedforward. Sixteen bit resolver systems (65525 step/rev. resolution) should provide sufficient resolution for introduction of acceleration feedforward compensation.

Both acceleration and velocity feedforward compensation have the desired effect of minimizing position error. To see the affects of the adjustment of feedforward compensation, the scanning program ran in Task 2 for viewing the feedback velocity can be modified for viewing position error. Simply replace REG:205 with REG:208 (Position Error) in the program listed. Since position error magnitude is typically larger than velocity, it may be necessary to rescale PRM:142 (Digital to Analog Scale Factor).

∇ ∇ ∇

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CHAPTER 12: HOST MODE OPERATION

In This Section:

- Introduction 12-1
- Service Request 12-2
- Host Mode Command Set..... 12-4

12.1. Introduction

The user has the option of operating the UNIDEX 100/U100i under a Host Mode. Instead of operating the U100/U100i with the UT100 software, the operator has the option to create a customized operator interface (that the operator runs instead of running the software program) using the “HOST” command set. The intent of this chapter is to list the specifications for the "HOST" command set.

The Host Mode operation consists of commanding and querying the controller from a computer using character strings. These character strings that make up the Host Mode command set duplicate the keystroke sequences manually typed on a terminal (e.g., the HT) to achieve the same effect.

RS-232-C requires a software reset to enter the host mode. The controller will respond with a service request when ready (must be acknowledged with a SRQ acknowledge code). The host mode reset is a single character defined by PRM:035. The Hex value represents the character stored in this parameter.

The Host Mode operates in both RS-232-C and IEEE-488 communications mode. IEEE-488 operation automatically selects the host mode. When operating in the RS-232-C communication mode, the user must first send the "HOST" reset character (PRM:035) to the controller after power up. Failure to do this prevents the U100 from recognizing this command set.

The keyword <LF> is a line feed character.



All SRQ's must be acknowledged by sending the Service request acknowledge code or the controller will hang up. Service requests may be disabled by PRM:022. However, the system continues to send error service requests. PRM:028 and PRM:029 set the service request character and service request acknowledge character. The Hex value representing these are stored in the parameters (see Chapter 5: Programming Commands for more detailed information concerning service requests).

Possible errors may occur when executing these commands. If this happens, a service request number “0x03” gets sent to the Host (applies to IEEE-488 and RS-232-C interface modes). The controller sends a host error code following the SRQ code.

To determine the specific error code, refer Appendix D.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



IEEE operation does not perform service requests as does the RS-232. This operation is performed by serial polling and its only use is for errors and file transfers.

12.2. Service Request

The controller uses the service request to ask for attention from the Host Controller when operating in RS-232 or IEEE-488 Host communication modes. Upon sending out the service request, it is necessary for the Host controller to acknowledge it with the correct reply. Failure to acknowledge the service request results in the appearance of a lock up of the controller.

The service request consists of the service request character followed by a SRQ status code and finally the complement of the SRQ status code. The SRQ character is user definable where it can be adjusted through PRM:028. For IEEE-488 operation, the service request mechanism is built into the interface through a mode known as "Serial Poll". However, for both RS-232 and IEEE-488, the status codes are the same. The controller sends multiple SRQ's on an individual basis with each one requiring a service request acknowledge. The SRQ status code provides information concerning the service request.

For RS-232 mode, the sequence is as follows:

<SRQ character> <SRQ status> <compliment of SRQ status>

For IEEE-488, the SRQ status loads directly into the SRQ register.

There are specific functions that determine the SRQ status (see the following).



The prefix "0x" denotes Hexadecimal (base 16) integers.

Table 12-1. Task 0 Service Request Status Codes

Codes	Definition
0x00	Reserved
0x01	Program Uploading in Progress
0x02	Program Downloading in Progress
0x03	Host Mode Command Processing Error
0x04	Memory Allocation Error (out of user memory)
0x05	"Line #" Update for "Task 1 Window" Running in the RS-232 Host Mode only
0x06	"Line #" Update for "Task 2 Window" Running in the RS-232 Host Mode only
0x07	Ready for Commands in Host Mode (RS-232 only)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 12-2. Task 1 Service Request Status Codes

Codes	Definition
0x10	Reserved
0x11	User Defined #1
0x12	User Defined #2
0x13	User Defined #3
0x14	User Defined #4
0x15	User Defined #5
0x16	User Defined #6
0x17	User Defined #7
0x18	User Defined #8
0x19	User Defined #9
0x1A	User Defined #10
0x1B	User Defined #11
0x1C	User Defined #12
0x1D	User Defined #13
0x1E	User Defined #14
0x1F	User Defined #15

The controller command "SRQ(<number>)" generates the codes in Table 12-3. The <number> is an integer 1 through 15 for status codes 0x11 through 0x1F respectively.

Table 12-3. Task 2 Service Request Status Codes

Codes	Definition
0x20	Reserved
0x21	User Defined #1
0x22	User Defined #2
0x23	User Defined #3
0x24	User Defined #4
0x25	User Defined #5
0x26	User Defined #6
0x27	User Defined #7
0x28	User Defined #8
0x29	User Defined #9
0x2A	User Defined #10
0x2B	User Defined #11
0x2C	User Defined #12
0x2D	User Defined #13
0x2E	User Defined #14
0x2F	User Defined #15

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The controller generates the codes in Table 12-4 in the same manner as that used in Task 1 for status codes 0x20 through 0x2F respectively.

Table 12-4. Kernel Service Request Status Codes

Codes	Definition
0x30	Reserved
0x31	Initiate "Xon" (allow RS-232 Host to transmit)
0x32	Initiate "Xoff" (stop RS-232 Host from transmitting)
0x33	SRQ Exception Mask encountered a "true" condition



IEEE-488 Service Request Status Codes may contain Bit 6 set. (e.g., Status Code 0x01 may be received as 0x41).

In the RS-232 mode, it is necessary for the Host Controller to send the Service Request Acknowledge (SA) to the controller after receiving a Service Request (SRQ). For the IEEE-488 mode the "Serial Poll" accomplishes the same task. Parameter PRM:029 permits the user to select the SA character for the RS-232 mode. Failure to send the Service Request Acknowledge causes the controller to get hung up and not respond to commands. A Service Request Acknowledge sent to the controller without a previous SRQ does not cause a problem.

12.3. Host Mode Command Set

The HOST_100.exe file is an executable program that operates the controller in Host Mode. HOST_100.C is the source file for this program written in “C”. This file contains C functions for all of the Host Mode commands. These functions can easily be copied into another source file to embed controller operation in a particular application. The HOST_100.exe utility allows all user data stored in flash ROM to be uploaded or downloaded to the PC. This is useful for OEMs requiring all units to run with the same user-defined default parameters and programs.



The HOST_100 files are on the UT100 disk that accompanies each controller.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

12.3.1. Running a "PGM:xx" File

Use the following syntax for running programs while in the host mode.

Syntax: *#B<task number><run mode><program number><LF>*

where:

<task number>	= "A" for Task 1 ="B" for Task 2
<run mode>	= "A" for Auto Mode ="B" for Block Mode

The Block Mode does not implement for host interface in Revision 1.0 Software. Use the immediate command mode.



<program number> = number of the desired program (in memory).

EXAMPLE:

```
#BAA1<LF>          ; Run program PGM1 in Task 1 while in the
                   ; Auto Mode.
```

A service request occurs after the system receives each command. To allow the service request to immediately follow execution, set PRM:023 and PRM:024.



12.3.2. Sending an Immediate Command

Use the syntax below to send an immediate command while in the host mode.

Syntax: *#C<task number><command string><LF>*

where:

<task number>	= "A" for Task 1 = "B" for Task 2
<command string>	= ASCII string containing the specified command. Chapter 6 lists the available controller command set. However, certain commands are not applicable for immediate commands (e.g., GOTO:1).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

EXAMPLE:

```
#CBD(100)<LF>          ; Execute the command D(100) in Task 2.  Note
                       ; that when executing an immediate command
                       ; for motion (e.g., D< >, V< >, A< >, or T< >)
                       ; that the proceeding GO statement is inferred.

#CAFV:1=1.5<LF>        ; Execute the command FV:1=1.5 in Task 1.
```



After the system receives each command a service request occurs. To allow the Service Request to immediately follow execution, set PRM:023 and PRM:024.

12.3.3. Making a Copy of a File

Use the following syntax to make a copy of a file while in the host mode.

Syntax: *#FB<file type><file number 1><LF><file number 2><LF>*

where:

<file type> = "A" for "PGM" file
= "B" for "DEF" file
= "C" for "MAC" file
= "D" for "CAM" file
= "E" for "LST" file
= "F" for "DIR" file

<file number 1> = number of selected file to copy.

<file number 2> = number of file to be copied to.

EXAMPLE:

```
#FBA1<LF>2<LF>          ; Copy file PGM1 to PGM2.
```



Service request after copy.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

12.3.4. Erasing a File

Use the following syntax to erase a file while in the host mode.

Syntax: *#FA<erase selection><file type><file number><LF>*

where:

<erase selection>	= "A" for individual file = "BY" for all files
<file type>	= "A" for "PGM" file = "B" for "DEF" file = "C" for "MAC" file = "D" for "CAM" file = "E" for "LST" file = "F" for "DIR" file
<file number>	= number of selected file types to erase.

EXAMPLE:

#FABY<LF>	; Erase all files from program memory.
#FAAB3<LF>	; Erase the DEF3 file from program memory.

Service request after erase.



12.3.5. Generating a File Directory

While in the host mode, use the following syntax to generate a directory of files that are currently in program memory (e.g., PGM:xx, DEF:xx, etc.).

Syntax: *#AB<LF>*

EXAMPLE:

#AB<LF>	; Generate a file directory.

After generating the directory the Service Request is sent.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

12.3.6. Transfer a File from the Host to the Controller

The following syntax is required to transfer a file from the Host to the controller while in the host mode.

Syntax: *#FDA<file type><file number><LF><SRQ Acknowledge><ASCII DATA><EOF>*

where:

<file type> = "A" for "PGM" file
= "B" for "DEF" file
= "C" for "MAC" file
= "D" for "CAM" file
= "E" for "LST" file
= "F" for "DIR" file

<file number> = number of selected file to transfer.

EXAMPLE:

```
#FDAC4<LF><DATA><EOF>      ; Transfer the MAC4 file from the Host to the
                           ; UNIDEX 100's program memory.
```



SRQ #2 is sent to the host after reception of <LF> only if PRM:022 is set to "1".

EOF (value specified by PRM:027) must be sent to terminate download.

12.3.7. Transfer a File from the Controller to the Host

Use the following syntax to transfer a file from the controller to the Host while in the host mode.

Syntax: *#FDB<file type><file number><LF><SRQ Acknowledge>< wait for ASII Data to be sent followed by an EOF>*

where:

<file type> = "A" for "PGM" file
= "B" for "DEF" file
= "C" for "MAC" file
= "D" for "CAM" file
= "E" for "LST" file
= "F" for "DIR" file

<file number> = number of selected file type to transfer.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

EXAMPLE:

#FDBA2<LF><uploaded Data terminated by EOF>	; Transfer the program 2 file from the controller's ; program memory to the Host.

SRQ #1 is sent to the host after reception of <LF> only if PRM:022 is set to "1".

End of data upload is signaled by the reception of the EOF character specified by PRM:027.



12.3.8. Modify the Value of a Parameter

Use the syntax below to modify the value of a specific parameter while in the host mode.

Syntax: *#FCA<parm type><parm number><LF>B<parm value><LF>*

where:

<parm type> = "A" for Communication parameters (0xx)
= "B" for Motion parameters (1xx)
= "C" for Drive parameters (2xx)
= "D" for System parameters (3xx)

The parameter type is determined by MSD of parameter number (e.g., PRM:202 is drive parameter "C").



<parm number> = number of PRM:xx selected for retrieval. Parm number is the 2 LSD's of the parameter number (e.g., the parameter number for PRM:202 is 2).

<parm value> = value to be passed to the specified parameter. This value can be either a floating point or an integer type depending on the specific parameter selected.

EXAMPLE:

#FCAB39<LF>B3<LF>	; Modify parameter #139 to the value 3.

After modifying the parameter value, the service request gets sent.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

12.3.9. Retrieve the Value of a Parameter

Use the following syntax to retrieve the value of a specific parameter while in the host mode. In this example, the ending return is optional.

Syntax: *#FCA<parm type><parm number><LF>A*

where:

<parm type>
= "A" for Communication parameters (0xx)
= "B" for Motion parameters (1xx)
= "C" for Drive parameters (2xx)
= "D" for System parameters (3xx)



Parameter type is determined by MSD of parameter number (e.g., PRM:033 is communication parameter "A").

<parm number> = number of PRM:xx selected for retrieval. Parameter number is the 2 LSD's of the parameter number (e.g., the parameter number for PRM:033 is 33).

EXAMPLE:

#FCAA33<LF>A	; Retrieve the value of parameter PRM:033.



After retrieving the parameter a service request gets sent.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

12.3.10. Modify the Value of a "Write Only" Register

Use the syntax below to modify the value of a "write only" register while in the host mode.

Syntax: *#FCB<reg type><reg number><LF><reg value><LF>*

where:

<reg type> = "A" for Communication registers (0xx)
= "B" for Motion registers (1xx)
= "C" for Drive registers (2xx)
= "D" for System registers (3xx)

Register type is determined by MSD of the register number (e.g., REG:308 is the system register "D").



<reg number> = number of REG:xx selected for modification. The register number is the 2 LSD's of the register number (e.g., the reg number for REG:308 is 8).

<reg value> = value to be passed to the specified write only register. This value can be either an integer or long type depending on the register selected.

EXAMPLE:

```
#FCBD8<LF>0x300<LF>        ; Modify the value of the write only register
                           ; REG:008 to the hexadecimal value of 0x300.
```

After modifying the register the service request gets sent.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

12.3.11. Retrieve the Value of a "Read Only" Register

Use the syntax below to retrieve the value of a read only register while in the host mode.

Syntax: *#FCB<reg type><reg number><LF>*

where:

<reg type> = "A" for Communication registers (0xx)
= "B" for Motion registers (1xx)
= "C" for Drive registers (2xx)
= "D" for System registers (3xx)



The register type is determined by MSD of register number (e.g., REG:015 is communication register "A").

<reg number> = number of REG:xx selected for modification. The register number is the 2 LSD's of the register number (e.g., the register number for REG:015 is 15).

EXAMPLE:

```
#FCBA15<LF>          ; Retrieve the value of the read only register
                     ; REG:015.
```



After reading the register the service request gets sent.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

12.3.12. Modify the Value of a "Read/Write" Register

Use the following syntax to modify the value of a read/write register while in the host mode.

Syntax: *#FCB<reg type><reg number><LF>B<reg value><LF>*

where:

<reg type> = "A" for Communication registers (0xx)
= "B" for Motion registers (1xx)
= "C" for Drive registers (2xx)
= "D" for System registers (3xx)

The register type is determined by MSD of the register number (e.g., REG:202 is Drive Register "C").



<reg number> = number of REG:xx selected for modification. The register number is the 2 LSD's of the register number (e.g., the register number for REG:202 is 2).

<reg value> = value to be passed to the specified read/write register. This value can be either an integer or long type depending on the register selected.

EXAMPLE:

```
#FCBC2<LF>B0<LF>          ; Modify the value of a read/write register
                          ; REG:202 to the value 0.
```

After modifying the register the service request gets sent.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

12.3.13. Retrieve the Value of a "Read/Write" Register

Use the syntax below to retrieve the value of a read/write register while in the host mode. In this example, the ending return is optional.

Syntax: *#FCB<reg type><reg number><LF>A*

where:

<reg type> = "A" for Communication registers (0xx)
= "B" for Motion registers (1xx)
= "C" for Drive registers (2xx)
= "D" for System registers (3xx)



The register type is determined by the MSD of the register number (e.g., REG:202 is Drive Register "C").

<reg number> = number of REG:xx selected for retrieval. The register number is the 2 LSD's of the register number (e.g., the register number for REG:202 is 02).

EXAMPLE:

```
#FCBC2<LF>A          ; Retrieve the value of a read/write register
                     ; REG:202.
```



After reading the register the service request gets sent.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

12.3.14. Modify the Value of a Variable

Use the following syntax to modify the value of a variable while in the host mode.

Syntax: *#FCC<var type><var number><LF>B<var value><LF>*

where:

<var type> = "A" for Integer variables
= "B" for Long variables
= "C" for Float variables
= "D" for Port variables

<var number> = number of VAR:xx selected for modification.

<var value> = value to be passed to the specified variable.

EXAMPLE:

```
#FCCC1000<LF>B34.8<LF>          ; Modify the variable FV:1000 to the value
                                ; of 34.8.
```

After modifying the variable the service request gets sent.



12.3.15. Retrieve the Value of a Variable

Use the syntax below to retrieve the value of a variable while in the host mode. In this example, the ending return is optional.

Syntax: *#FCC<var type><var number><LF>A*

where:

<var type> = "A" for Integer variables
= "B" for Long variables
= "C" for Float variables
= "D" for Port variables

<var number> = number of VAR:xx selected for retrieval.

EXAMPLE:

```
#FCCA33<LF>A                    ; Retrieve the value of the variable BV:33.
```

After reading the variable the service request gets sent.



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

12.3.16. Modify the Value of a String Storage Buffer

Use the following syntax to modify the value of a string buffer while in the host mode.

Syntax: *#FCD<string number><LF>B<string data><LF>*

where:

<string number> = number of SV:xx selected for modification.

<string data> = any sequence of ASCII printable characters up to and including 20 per string buffer.

EXAMPLE:

#FCD5<LF>Bthis is string 5<LF>	Modify the string buffer SV:5 to the following string: "this is string 5"



After modifying the string the service request gets sent.

12.3.17. Retrieve the Value of a String Storage Buffer

Use the following syntax to retrieve the value of a string buffer while in the host mode. In this example, the ending return is optional.

Syntax: *#FCD<string number><LF>A*

where:

<string number> = number of SV:xx to be retrieved.

<string data> = any sequence of ASCII printable characters up to and including 20 per string buffer.

EXAMPLE:

#FCD1<LF>A	; Retrieve the character string that is stored in ; the string variable SV:1.



After reading the string the service request gets sent.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

12.3.18. Move Out of a Limit

Use the following syntax to reset a limit condition while in the host mode. Upon executing this command, the axis attempts to move out of the limit (PRM:128 determines the distance to move).

Syntax: *#EA*

After clearing the limit (service request is sent), the user must send a service request acknowledge code).



EXAMPLE:

```
#EA                    ; Reset the limit.
```

The user must acknowledge the service request



12.3.19. Task Control Functions

These functions allow the user to perform block executes, feedholds, and task quit operations. Use the following syntax to perform these functions.

Syntax: *#EB<function>*

where:

<function>

"A" = Task A Block Execute (for programs running in the Block Mode).
"B" = Task A Feedhold
"C" = Task A Quit
"F" = Task B Block Execute (for programs running in the Block Mode)
"G" = Task B Feedhold
"H" = Task B Quit

For RS-232, the operation line numbers get sent when running a program in the Block Run Mode.

The user must be prepared to handle multiple service requests (all service requests must be acknowledged).

The user must send the F1 code to return to the host mode. (The F1 code is equal to 11 Hex or 17 Dec.)



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

EXAMPLE:

```
#EBH                    ; Quit the Task B operation.
```



The user must handle the service request.

EXAMPLE:

```
<F1>                    ; Send the F1 code to return to the host mode.
```



The user must handle the Service Request.

The user must acknowledge the service request after the F1 code is sent to return to the host mode. (F1 is equal to 11 Hex or 17 Dec.)

∇ ∇ ∇

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

APPENDIX A: GLOSSARY OF TERMS

Appendix A

In This Section:
- Terms Used In This Manual
- Definitions

This appendix contains definitions of terms that are used throughout this manual.

absolute positioning - Absolute positioning is positioning that is done with respect to an initial starting position (typically referred to as the home position) and typically uses a standard coordinate system (using [X,Y] coordinates is an example of absolute positioning). In contrast, incremental (or relative) positioning is done using a series of relative moves. These moves are relative to the previous location rather than a single reference point (for example, relative changes in position [ΔX,ΔY] are examples of incremental positioning).

acceleration feed forward - Acceleration feed forward is a control strategy (represented as a dimensionless gain value that is sometimes used during the motor tuning process) in which acceleration commands are sent directly to the amplifier.

accuracy - Accuracy is the difference between an expected value and an actual value expressed as a percentage.

amplifier - An amplifier is a hardware device having an output that is a function of the input signal.

axis - An axis is a direction along which movement occurs.

axis calibration - Axis calibration is the process by which the current position of an axis is adjusted to match the actual position (as determined by a laser for example) of the axis.

backlash - Backlash is a movement that occurs between two or more interacting mechanical parts as a result of looseness.

ballscrew - A ballscrew is a precision motion component of mechanical stages and consists of a precisely threaded shaft (or channel) and a housing that rides along the shaft as the shaft is rotated. The housing of a ballscrew contains ball bearings that ride in the channel of the shaft as the shaft rotates. A small tube on the housing recycles the bearings as the shaft rotates. The conversion factor parameter calculation is different for ballscrew systems compared to other systems. (Compare with leadscrew.)

base address - A base address is a number that represents the memory location in the computer where input/output (I/O) information can be stored. All devices (e.g., the U100 card, network cards, tape backup cards, etc.) within a computer must have unique I/O base addresses.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

batch file - A batch file is a file that contains a series of commands (e.g., the AUTOEXEC.BAT file is a batch file).

bit - The term bit is an acronym for "Binary digIT" and represents a single binary number (i.e., a "1" or a "0"). In digital computers, a bit's two states can represent an off state and an on state, a high voltage and a low voltage, the numbers 0 and 1, etc.

brushless motor - Aerotech brushless motors are three-phase, rare earth permanent magnet servo motors which generate a sinusoidal back EMF voltage and are usually referred to as AC brushless motors. Another type, usually referred to as the DC brushless motors, generate a trapezoidal back EMF and produce more torque ripple.

byte - A byte is a common unit of information storage made up of eight binary digits (bits). A byte can be used to represent a single ASCII character (e.g., "A"= 10000001 [binary]) or binary numbers from 00000000 to 11111111 (from 0 to 255 decimal), depending on how it is used.

C - C is a high-level programming language (developed at Bell laboratories) that is able to manipulate a computer at a low level like assembly language. A C program can be used to write a customized software interface for the UNIDEX 100, rather than using the software program that is supplied with the system. If a customized interface program is created using C, the appropriate C library must be included when the program is compiled. Refer to Chapter 12: Host Mode Operation for more information.

cam motion - Digital cam motion refers to the electronic emulation of a mechanical cam. Digital cam motion is accomplished using a cam table - a list of positions that represent the profile of the cam. Typically, an axis will move through the cam table by indexing to the next position in a specified time interval.

circular interpolation - Circular interpolation refers to the UNIDEX 100's ability to coordinate two axes to produce accurate circular motion using minimal reference information (e.g., the center point and a radius).

closed loop system - A closed loop system is a drive system that uses sensors for direct feedback of position and/or velocity. Contrast with open loop system

commutation - Commutation refers to the process by which every other cycle of an alternating current is reversed so that a single unidirectional current is supplied. In the case of motors, commutation refers to the switching of current to motor windings which causes the motor to rotate. In a DC servo motor, this is done mechanically using brushes and a commutator. A brushless motor is electronically commutated using a position feedback device such as an encoder that is mounted to the rotor. Stepping motors are electronically commutated without feedback (in an open loop fashion).

constant velocity motion - Constant velocity motion refers to the controller's ability to perform motion while maintaining a constant velocity during the motion. For an application example, consider an irregularly shaped pattern that requires a series of *perforations* (made using a series of on/off laser pulses, for example). Constant velocity motion ensures that the length of each perforation is the same.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

cubic spline interpolation - Cubic spline interpolation is a mathematical process used by the controller in which a smooth curve (path) is based on two sets of coordinates ([X1,Y1] and [X2,Y2]) on the curve. Unlike linear interpolation, however, a previous coordinate ([X0,Y0]) and following coordinate ([X3,Y3]) are used to determine the curve's slope entering the path and exiting the path.

derivative gain - Derivative gain is a dimensionless motor tuning parameter that serves to dampen the system response by producing a dampening force as long as the system is progressing toward error reduction.

DOS - DOS is an acronym for Disk Operating System--a master control program that runs a computer and acts as a scheduler, providing job, task, data and device management. DOS is a generic term that refers to an operating system. MS-DOS is a single-user operating system for PCs that was designed by Microsoft Corporation. Since the DOS and MS-DOS programs are very, the terms DOS and MS-DOS are frequently interchanged.

double word - A word is a number of bytes that are processed as a single unit by a computer. In the controller, a word consists of two bytes or 16 bits. A double word is twice that amount (i.e., four bytes or 32 bits).

DSP - DSP is an acronym for Digital Signal Processor - an advanced, high-speed processor technology that analyzes input signals by first converting them into digital data and then uses various algorithms (such as fast Fourier transforms) for analysis. Once a signal has been converted to digital data, the individual components of the signal can be isolated and interpreted more readily than in analog form.

electronic gearing - Electronic gearing is the process of moving one or more slave axes in coordination with a master axis without continuously sending commands from the host program. By establishing a series of relationships between axes, the servo processor will continuously update the positions and velocities of the slave axes based on the commanded motion of the master axis. The master can be a physical axis in the system or a virtual axis used for synchronization purposes only.

encoder - An encoder is a rotary device that transmits a pulsed signal based on the number of revolutions of the device.

faults - A fault is an error condition that occurs when a component of the controller system operates outside certain parameters. Fault masks are used to allow the controller system to detect and act on any fault condition of the system. Examples of major fault conditions include position faults, velocity faults, integral faults, RMS over current faults, amplifier faults, and feedback faults.

feedrate error - A feedrate error is a type of fault that is generated by the controller if the current speed of an axis exceeds a programmable maximum speed (called the Top Feedrate parameter [x17]). Feedrate errors are necessary because certain stages or motors have a maximum operating speed above which components may be damaged.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

floating point number format - Floating point number format is a method of representing numbers without defining a fixed number of decimal places. Two common forms of floating point number format are fixed-style format (e.g., 12.345, 0.000001, -2, etc.) and scientific notation (e.g., 12.3E4, -1.2E-3, etc.). The controller uses fixed-style format for floating point numbers.

Hall effect switch - A Hall effect switch is a solid state switch that is activated by a magnetic field. Some AC brushless motors use Hall effect switches.

hexadecimal number format - Hexadecimal number format is a method of representing large numbers using base 16 rather than the standard base 10. In base 16 or hexadecimal number format (often abbreviated "hex"), the number positions represent powers of 16 (rather than powers of 10 in decimal). The decimal number positions (1's, 10's, 100's, 1,000's, 10,000's, etc.) are replaced with hexadecimal number positions (1's, 16's, 256's, 4096's, etc.). Also, while the individual numerals for the decimal system are 0-9, the numerals for the hexadecimal number system (which requires 16 unique "numerals") are 0-9 then A-F (where $A_{16}=10_{10}$, $B_{16}=11_{10}$, $C_{16}=12_{10}$, $D_{16}=13_{10}$, $E_{16}=14_{10}$, and $F_{16}=15_{10}$). For simplicity in this manual, hexadecimal numbers are written with a preceding "0x" rather than using the subscript 16. For example, the hexadecimal number 12A5 is written 0x12A5. Numbers without the preceding "0x" are assumed to be decimal unless otherwise indicated.

home cycle - The home cycle is series of motions that are used to move the specified axes to a hardware referenced position. There are two feedrates (in the form of parameters) associated with the home cycle: the Power On Feedrate and the Normal Home Feedrate. The power on home cycle (the first commanded home cycle following a power up) uses the Power On Feedrate parameter and the normal home cycle (all subsequent home cycles) uses the Normal Home Feedrate parameter.

home marker option - The home marker option is a type of encoder that can be used with stepper motors. This option provides an inexpensive way of establishing a very accurate home reference (usually within 0.1 microns, in most Aerotech equipment). The home marker is protected in a rugged housing that also provides terminal connections for the encoder, the motor and the limit switch.

in-position integrator - An in-position integrator is a motor tuning adjustment that can be used to help remove steady-state position errors as well as reduce the effects of tachometer loop drift. In-position integration is accomplished at a rate that is directly proportional to the integral gain value (Ki).

incremental positioning - Incremental (or relative) positioning is done using a series of relative moves. These moves are relative to the previous location rather than a single reference point (for example, relative changes in position [ΔX,ΔY] are examples of incremental positioning). In contrast, absolute positioning is positioning that is done with respect to an initial starting position (typically referred to as the home position) and typically uses a standard coordinate system (using [X,Y] coordinates is an example of absolute positioning).

Inductosyn - An Inductosyn is a rugged, very accurate, multi-pole electromagnetic transducer with an operating principle similar to that of a resolver.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

initialization - Initialization is the process in which the current configuration (.CFG) file and the current parameter (.PRM) file are sent down into the memory of the controller board from the PC. In addition, the actual control program (the *firmware* file U100.JWP) is also sent down to the controller board. The last step of the initialization process is the restarting of the controller firmware program. Initialization can be accomplished in any one of four ways: (1) selecting the Reset option of File menu, selecting the F9 Reset soft key at the bottom of the main screen, selecting the Reset option from the Functions menu, or pressing the F9 function key on the keyboard.

integral error - Integral error is the summation of position errors over time. An integral error fault is generated if the integral error for an axis exceeds a programmable maximum integral error value (parameter x20). Integral error is reset every time the axis is reset.

integral gain - Integral gain is a dimensionless motor tuning parameter that serves to help remove steady-state position errors as well as reduce the effects of tachometer loop drift.

jog move - A jog move is a momentary movement of an servo drive to provide manual control of axis motion.

joystick - A joystick is manual input control device that digitizes a path using two axes. A joystick offers direct motion control for easy machine setup and testing.

jumpers - Jumpers are hardware *ties* that you manually position into different sockets to configure the hardware platform. Jumpers on the controller board are used to configure the motor type, current output, communication mode, and other features.

leadscrew - A leadscrew is a motion component of stages and consists of a threaded shaft and a housing that rides along the shaft as the shaft is rotated. The housing of a leadscrew contains a similar thread, which rides along the shaft thread as the shaft rotates. Leadscrews are more economical, but less accurate than ballscrews.

LED - LED is an acronym for light-emitting diode. An LED is a semiconductor diode that converts electrical energy into visible electromagnetic radiation. The controller has LEDs located on the front panel that are used for diagnostic purposes.

linear interpolation - Linear interpolation is a mathematical process used by the U100 in which a straight line (path) is based on two sets of coordinates ([X1,Y1] and [X2,Y2]) on the line. Unlike cubic spline interpolation (which uses two additional coordinates to determine the slope of the smoothed curve), linear interpolation only uses two sets of coordinates and generates a straight (not smoothed) path..

linear motor driver - A linear motor driver is a non-switching type of DC servo amplifier that drives the motor with direct current. Linear amplifiers do not generate electrical noise or switching losses in the motor. They do, however, have a higher rate of power dissipation than a pulse-width modulated (PWM) amplifier.

machine step - A machine step is the smallest feedback device step that can be taken. This is the smallest possible increment of movement as measured by the feedback device.

microstepping - Microstepping is a technique for driving stepping motors more smoothly and with higher resolution than full step control. Current is divided in a sine-cosine

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

fashion between motor phases to provide intermediate positions between full step positions.

multitasking - Multitasking is software technique that gives several functions (or tasks) the appearance of individually having sole access to the resources of the system (for example, the microprocessor). In its simplest form, a multitasking system assigns a small time slice to each task in a round robin fashion. Only one task at a time has access to the multitasking system's resources. When each successive task has had the opportunity to use the system resources (for a brief period), the cycle repeats.

notch filter - A notch filter is a software filter that is used to remove a section of frequencies in order to stabilize a system with a known mechanical resonance.

open loop system - An open loop system is a drive system that does not employ feedback sensors to monitor position or velocity. Most stepper motor applications are open loop (that is, they have no feedback). The commanded position is the assumed motor position.
operator - (1) An operator is one who uses the controller system.

operator - (2) An operator is a programming element that is used to link terms in an expression. Programming operators include the standard arithmetic operators (e.g., +, -, * and /), comparison operators (e.g., < and >) and Boolean operators (e.g., AND, OR and NOT) and others.

orthogonality - Orthogonality is a state of two axes in which one is perpendicular to the other. The controller provides orthogonality correction capabilities that allow an axis to be corrected (using absolute machine step correction data) based on a position-dependent axis. Orthogonality correction, if used, is incorporated into the axis calibration (.CAL) file.

plane - A plane is an axis or group of axes that can be coordinated (for example, a particular action of one plane can trigger an action on another plane) or independent (for example, one plane can be milling a part while another plane is etching circles). Planes can also be virtual panes, which are not linked to any particular axis, but act as queues or buffers.

point-to-point motion - Point-to-point motion simply involves specifying a target position. After the target position is commanded, the controller strives to attain that position with no time or path constraints.

popup window - A popup window is a box that appears "on top of" the existing screen or display. Popup windows typically contain fields (for additional information to be supplied), buttons (to acknowledge or cancel a particular operation, for example) or a combination of both.

position error - Position error is the difference between the commanded position of an axis and the feedback position (i.e., the difference between the desired position and the actual position). A position error fault occurs if the current position error exceeds a programmable maximum position error (parameter x19). Position error is measured in machine steps.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

power-up home cycle - The power-up home cycle refers to the positioning of an axis to a known, hardware-referenced position when the axis is sent “home” for the first time after a power up. Subsequent home commands use the *runtime home cycle*.

program - A program is a set of instructions that are carried out in some predefined logical order. A controller program is a sequential list of controller programming commands (see Chapter 5) which tell the control board how to perform specific motions for a particular application. Controller programs may be created/edited on-line (from the UT100 software) or off-line (using any standard ASCII text editor).

program step - A program step is the smallest programmable increment of motion that can be commanded. A program step equals the programming units * 10^{ndec} where "ndec" is the number of decimal digits set by parameters 029, 030, 047 and 048 for Metric mode, and parameters 065, 066, 083 and 084 for English mode.

program unit - A program unit is a user-defined measurement unit such as inches, millimeters, degrees, etc. Program units are used within the application program and provide the operator with flexibility and ease of use. For example, it is more meaningful for an operator to command a “100 mm” move than it is to command a “752 machine step” move.

proportional gain - Proportional gain is a dimensionless motor tuning parameter that produces an output that is related to the Velocity Error in the servo loop.

pull-down menu - A pull-down menu is a vertical list of commands. When the menu is closed (or “rolled up”), only the menu name is visible on the menu bar. When the menu name is selected, the menu “unrolls” and the list of commands is displayed. Some options in pull-down menus are the names of other menus (called *cascading menus*).

qms - QMS is an abbreviation for quarter millisecond - a unit of time that is used when determining values such as velocity error, for example, which is measured in machine units per quarter millisecond (i.e., machine units/qms). One qms is equivalent to 0.25 milliseconds.

quadrature - Quadrature is the state of two signals that are displaced 90 degrees with respect to each other. In most rotary incremental optical encoders, light (from an LED, for example) is measured after it is passed through slits in a grating disk (which is attached to the axis being measured). Typically, two tracks on the disk have their gratings displaced 90 degrees with respect to each other (that is, the tracks are said to be in quadrature).

QuickBASIC - QuickBASIC is a popular compiler program from Microsoft, Inc. which provides an advanced number of features to the BASIC programming language.

registers -

reset - *see initialization.*

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

resolver - A resolver is a two-phase, rotary, electromagnetic transducer in which inductive coupling (between the rotor and stator windings) and trigonometric principles are employed to provide absolute position information over one electrical cycle (which is one revolution for "single-step" resolvers)

resolver-to-digital card (RDP-PC) - The RDP-PC card is an optional PC-based R/D card that is used to receive resolver or Inductosyn feedback. Resolution is selectable among 10-bit, 12-bit, 14-bit or 16-bit.

RMS current trap - RMS current trap is an error that occurs if the current being commanded to a motor exceeds a programmable limit (see parameters x48 and x49). RMS current trap is analogous to a software “fuse”. Essentially, this fault functions the same as a physical fuse, but is done through software. One obvious advantage is that a “software fuse” does not have to be replaced like a physical fuse.

runtime home cycle - The runtime home cycle refers to the positioning of an axis to a known, hardware-referenced position. The controller uses this sequence for all home cycles except when the axis is sent “home” for the first time after a power up. In this latter case, the *power-up home cycle* is used instead.

servo control system - A servo control system (servo loop) is a motion control system which continuously compares desired position/velocity to actual position/velocity and produces an error correction command. Servo systems use sensors to feedback actual position/velocity.

shaft runout - Shaft runout is an expression of the total indicated reading of wobble or nonconcentricity as measured at the end of a motor shaft when the shaft is rotated one complete revolution.

software - The term software refers to a computer program. Contrast software with hardware, the physical machinery, components and support peripherals through which the software runs.

task - A controller task is one of four sets of instructions that are executed sequentially at such a high speed that each task has the impression that it alone has full access to all of the microprocessor’s time.

time-based motion - A time-based motion is a motion that arrives at a specified location in a desired amount of time. After the target position of a move is programmed, the controller chooses any speed to achieve that position on time.

traps -*See* faults.

tuning - Tuning is the process of optimizing the operation of a servo system.

variables - Variables are programming terms that are used as temporary storage locations for calculations.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

velocity error - Velocity error is the difference between the commanded velocity and the velocity derived from the feedback position (i.e., the difference between the desired velocity and the actual velocity). Velocity error is measured in machine steps per quarter millisecond (machine steps/qms). A velocity error fault occurs if, at any time, the velocity error of the system exceeds a programmable velocity error (specified by parameter x18).

velocity feed forward - Velocity feed forward is a control strategy (represented as a dimensionless gain value that is sometimes used during the motor tuning process) in which current velocity disturbances are converted into corrective actions now in order to minimize the future effects of the disturbances.

velocity profiled motion - Velocity profiled motion is a move of a programmed distance and speed from the current position. Velocity profiled motions are executed only after the previous motion has reached its deceleration point.

word - A word is a number of bytes that are processed as a single unit by a computer. In the U100, a word consists of two bytes or 16 bits.

∇ ∇ ∇

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

APPENDIX B: WARRANTY AND FIELD SERVICE

Appendix B

In This Section:

- Laser Product Warranty
- Return Products Procedure
- Returned Product Warranty Determination
- Returned Product Non-warranty Determination
- Rush Service
- On-site Warranty Repair
- On-site Non-warranty Repair

Aerotech, Inc. warrants its products to be free from defects caused by faulty materials or poor workmanship for a minimum period of one year from date of shipment from Aerotech. Aerotech's liability is limited to replacing, repairing or issuing credit, at its option, for any products which are returned by the original purchaser during the warranty period. Aerotech makes no warranty that its products are fit for the use or purpose to which they may be put by the buyer, whether or not such use or purpose has been disclosed to Aerotech in specifications or drawings previously or subsequently provided, or whether or not Aerotech's products are specifically designed and/or manufactured for buyer's use or purpose. Aerotech's liability or any claim for loss or damage arising out of the sale, resale or use of any of its products shall in no event exceed the selling price of the unit.

Laser Products

Aerotech, Inc. warrants its laser products to the original purchaser for a minimum period of one year from date of shipment. This warranty covers defects in workmanship and material and is voided for all laser power supplies, plasma tubes and laser systems subject to electrical or physical abuse, tampering (such as opening the housing or removal of the serial tag) or improper operation as determined by Aerotech. This warranty is also voided for failure to comply with Aerotech's return procedures.

Return Procedure

Claims for shipment damage (evident or concealed) must be filed with the carrier by the buyer. Aerotech must be notified within (30) days of shipment of incorrect materials. No product may be returned, whether in warranty or out of warranty, without first obtaining approval from Aerotech. No credit will be given nor repairs made for products returned without such approval. Any returned product(s) must be accompanied by a return authorization number. The return authorization number may be obtained by calling an Aerotech service center. Products must be returned, prepaid, to an Aerotech service center (no C.O.D. or Collect Freight accepted). The status of any product returned later than (30) days after the issuance of a return authorization number will be subject to review.

Returned Product Warranty Determination

After Aerotech's examination, warranty or out-of-warranty status will be determined. If upon Aerotech's examination a warranted defect exists, then the product(s) will be repaired at no charge and shipped, prepaid, back to the buyer. If the buyer desires an air freight return, the product(s) will be shipped collect. Warranty repairs do not extend the original warranty period.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Returned Product Non-warranty Determination

After Aerotech's examination, the buyer shall be notified of the repair cost. At such time the buyer must issue a valid purchase order to cover the cost of the repair and freight, or authorize the product(s) to be shipped back as is, at the buyer's expense. Failure to obtain a purchase order number or approval within (30) days of notification will result in the product(s) being returned as is, at the buyer's expense. Repair work is warranted for (90) days from date of shipment. Replacement components are warranted for one year from date of shipment.

Rush Service

At times, the buyer may desire to expedite a repair. Regardless of warranty or out-of-warranty status, the buyer must issue a valid purchase order to cover the added rush service cost. Rush service is subject to Aerotech's approval.

On-site Warranty Repair

If an Aerotech product cannot be made functional by telephone assistance or by sending and having the customer install replacement parts, and cannot be returned to the Aerotech service center for repair, and if Aerotech determines the problem could be warranty-related, then the following policy applies:

Aerotech will provide an on-site field service representative in a reasonable amount of time, provided that the customer issues a valid purchase order to Aerotech covering all transportation and subsistence costs. For warranty field repairs, the customer will not be charged for the cost of labor and material. If service is rendered at times other than normal work periods, then special service rates apply.

If during the on-site repair it is determined the problem is not warranty related, then the terms and conditions stated in the following "On-Site Non-Warranty Repair" section apply.

On-site Non-warranty Repair

If any Aerotech product cannot be made functional by telephone assistance or purchased replacement parts, and cannot be returned to the Aerotech service center for repair, then the following field service policy applies:

Aerotech will provide an on-site field service representative in a reasonable amount of time, provided that the customer issues a valid purchase order to Aerotech covering all transportation and subsistence costs and the prevailing labor cost, including travel time, necessary to complete the repair.

Company Address

Aerotech, Inc.
101 Zeta Drive
Pittsburgh, PA 15238-2897
USA

Phone: (412) 963-7470
Fax: (412) 963-7459
TWX: (710) 795-3125

∇ ∇ ∇

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

APPENDIX C: SPLINE GENERATION

Appendix C

In This Section:
- U100 Spline Generation

The following section provides information concerning the operational characteristics of the spline generation mode that is initiated by the CAM() command.

In order to understand some of the issues presented in this section, parameters descriptions for PRM:152 through PRM:163 (Chapter 6: Parameters) and the description for the CAM( ) command (Chapter 5: Programming) should be reviewed.

Trajectory generation using the CAM( ) command provides a higher level of trajectory control than the familiar D( )... GO( ) command set, and the DD( ) command set. Unlike the latter commands, the CAM() command uses a cubic spline or linear interpolation algorithm to connect a series of pre-defined points into a smooth position profile that can be driven (geared) to a variety of reference sources.

Unlike the DD() command, the spline command CAM() can generate a smooth curve with as little as four data points, if the cubic spline mode is enabled. Multi-level velocity trajectories can be defined to have the ability to change direction without stopping at zero. This is unlike the D()…GO() command set.

The cubic spline algorithm generates a smooth curve to connect the desired spline points relative to the reference axis. For a given sub-interval connecting two adjacent points, this algorithm has the following form.

$$f(x_i) = A_i\, frac(x_i)^3 + B_i\, frac(x_i)^2 + C_i\, frac(x_i) + D_i$$

Where "*i*" denotes the given sub-interval for which coefficients *A,B,C,* and *D* are calculated.

The simple linear interpolation algorithm generates a piece-wise continuous profile to connect the desired points relative to the reference. For a given sub-interval connecting two adjacent points, this algorithm has the following form.

$$f(x_i) = C_i frac(x_i) + D_i$$

Where "*i*" denotes the given Sub-interval for which C and D calculated.

For the cubic spline, the position term (denoted by the term "*f(x)")* is determined by the addition of three coefficients *A*, *B*, and *C* multiplied by the appropriate powers of the fractional part of the reference variable "*x*" and added to a constant coefficient *D*. A

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

similar process is used to obtain the position term for linear interpolation, except only coefficients *C* and *D* are used.

At the points where "*x*" is an integer value, the equation above simplifies to the following.

$$f[i] = D_i$$

It is the coefficient *D* that is supplied by user as the desired position to be attained when the reference variable "*x*" approaches an integer value.

The selection of cubic spline or linear interpolation mode is provided by setting PRM:152 to 1 (default) for cubic spline, or 2 for linear interpolation.

An illustration of a typical spline profile is shown in Figure C-1. This illustration will serve as reference for demonstrating some key points of the controller spline control algorithm, which is explained in the remainder of this section. For simplicity, only the cubic spline mode will be discussed in the following paragraphs.

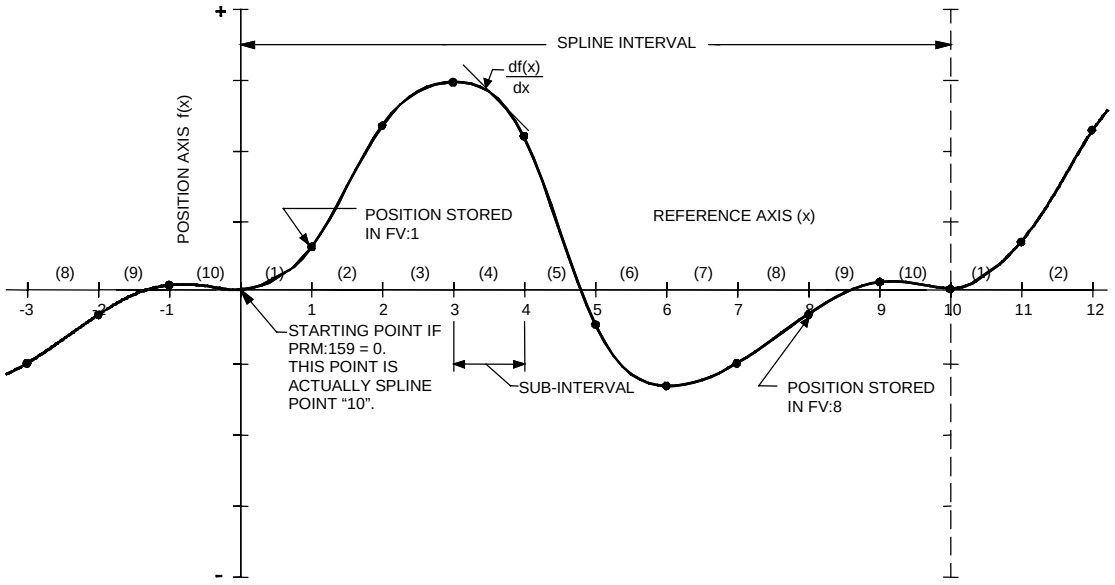


Figure C-1. Diagram of a Typical Spline Profile

Referring to C-1, a cubic spline has been drawn through ten (10) arbitrary points referenced to the "*x*" axis. For this example, the reference axis "*x*" has been broken into 10 segments defining the "*Spline Interval*". In reality, this interval can be specified as an integer number between 4 and 720 and is defined by parameter PRM:153.

The numbers in parenthesis on the "*x*" axis are defined as the "*Sub-Intervals*" of the spline and are always equal to the number of spline points set by PRM:153.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The internally generated reference, whose source is generated by the selection of one of five modes defined by parameter PRM:155, is setup in such a way that the integer portion of this reference always defines a given "*Sub-Interval"*.



For example, if parameter PRM:155 is set to mode 3 (position from encoder 1) with parameter PRM:157 set to 1000.0 , then every 1000 machine steps of encoder 1 will generate one sub-interval of axis "*x"*. Likewise, if parameter PRM:155 is set to mode 1 (time base reference) with parameter PRM:156 set to 10.0, then it will take exactly 10 seconds to execute the "*Spline Interval"* , refer to Figure C-1.

As shown in Figure C-1, the desired spline position points (shown as the heavy "dots" on the curve), defines the boundaries of each "*Sub-Interval"* and was defined earlier to be:

$$f[i] = D_i$$

The "*D"* coefficients for each sub-interval are entered into the UNIDEX 100 through the "FV:" variable array such that:

$$D_i = FV{:}<i>$$

The data in the "FV:" variables can be loaded through the host (with subsequent execution by the command CAM(0)) or automatically loaded and executed via the command "CAM(<file number>)". The data entered into the "FV" variables is always interpreted as absolute position relative to the imaginary variable "FV:0". In actuality, variable "FV:0" is FV:<PRM:153>.

The imaginary variable "FV:0" exists because of the nature in which the cubic spline is calculated. The calculations of coefficients A_i, B_i, C_i depend of the states of the following variables.

$$FV{:}<i-1>, FV{:}<i>, FV{:}<i+1>, FV{:}<i+2>$$

As can be seen from Figure C-1, to obtain the coefficients A, B, C for Sub-Interval 1 and 10, the curve must wrap around on itself. This is why the curve is depicted as repeating itself in the positive and negative direction as the reference goes towards + and - infinity.

Execution of the spline always starts at the reference value set by parameter PRM:159. The spline can be forced to end at any negative or positive reference value set by parameters PRM:158 or PRM:160, respectively.

The starting point for the spline (set by PRM:159) is always relative to the current position of the controller, regardless of the controller being set for incremental or absolute mode (INCR or ABSL commands).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

If parameter PRM:161 is set to a "1" (denoting modulo mode), the origin of the reference axis is reset to zero when the reference exceeds the magnitude of the value set in parameter PRM:153. In other words , the reference value of "11" would actually be the value "1" and the reference value of "-1" is actually the value "9", and so forth as shown in Figure C-1.

The modulo mode is useful in applications where the reference is generated by a rotating axis. This axis ,if allowed to revolve indefinitely could generate a reference value that would exceed the internal numerical range of the controller.

The velocity of the spline can be analytically determined by calculation of the rate of position change relative to the reference change (denoted *df(x)/dx*) in Figure C-1. Of course, the term "*dx"* is always dependent on the type of source that is generating the reference (see PRM:155) along with the setting of parameters PRM:156 and PRM:157, as well as PRM:154.
The average velocity of a given sub-interval can be determined by finding the difference in position of the two adjacent spline points and dividing that difference by length of the sub-interval.

Parameter PRM:154 defines the scaling mode of the spline interpolation algorithm. The scaling mode can be set to scale either the position axis *"f(x)"* (mode 2), the reference axis "*x*" (mode 1), or neither axis (mode 0, the default mode).

For the reference scaling mode (PRM:154 set to 3), *"x"* is pre-processed through the Scaled Trajectory Modifier algorithm (refer to Figure C-2) while *"f(x)"* is unaffected. This reference scaling mode is applicable for reference generating modes (PRM:155 equal to 3 through 5).

For the position scaling mode (PRM:154 set to 2), *"f(x)"* is pre-processed through the "Scaled Trajectory Modifier" algorithm while "*x*" is unaffected. The position scaling mode is applicable for all reference generating modes (PRM:155 equal to 1 through 5).

For the reference scaling mode (PRM:154 set to 1), "*x*" is pre-processed through the "Scaled Trajectory Modifier" algorithm (refer to Figure C-2) while "*f(x)*" is unaffected. Since the "Scaled Trajectory Modifier" algorithm operates with it's reference variable based on "time", this reference scaling mode can only be utilized when parameter PRM:155 is set to 1 (time reference spline generation).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

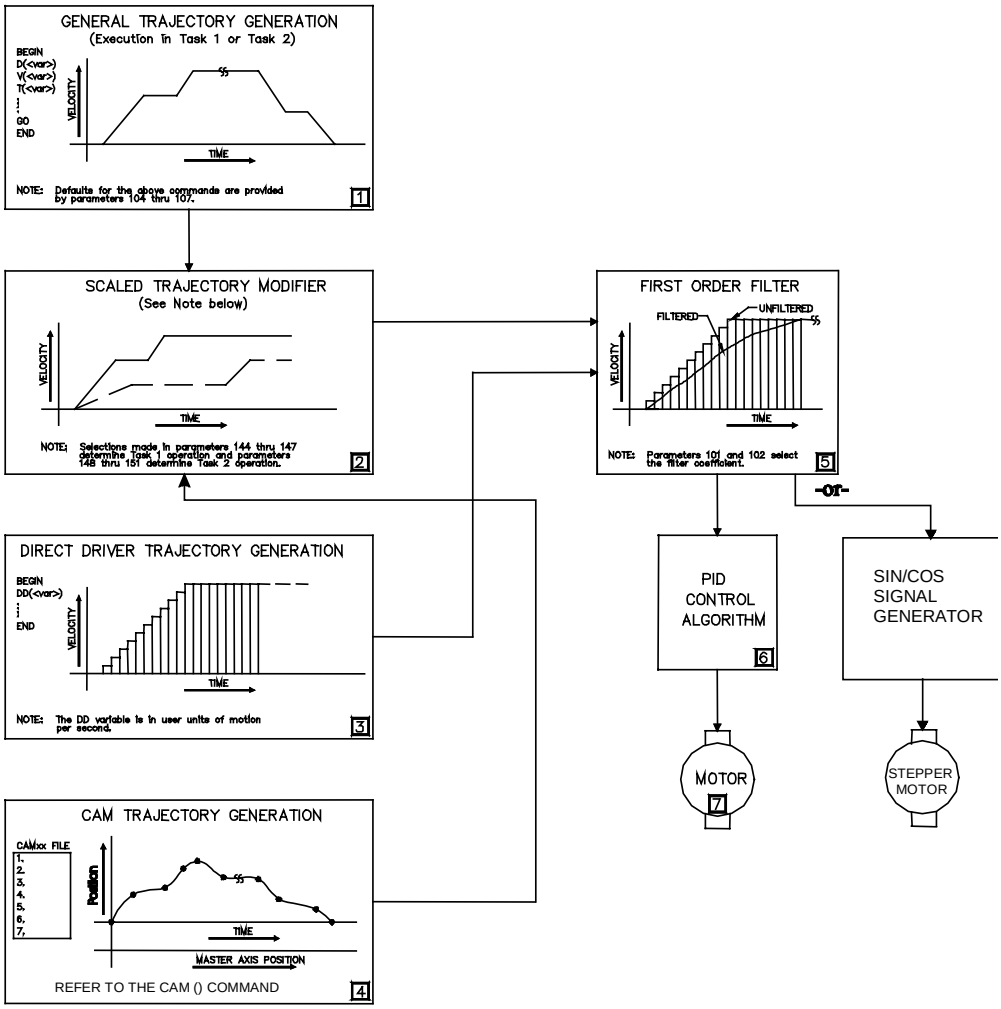


Figure C-2. Trajectory Generation Overview

The scaled "Scaled Trajectory Modifier" algorithm is controlled by parameters PRM:144 through PRM:147 if the CAM() command is running in task 1, or is controlled by parameters PRM:148 through PRM:151 if the CAM() command is running in task 2.

Finally, parameters PRM:162 and PRM:163 control data output "on the fly". If parameter PRM:162 is set to modes "1" through "4" data stored in the variable BV:<number> (where "number" represents the given "sub-interval"), will be sent to the specified output location. The data is sent at each sub-interval transition.

For example, if the value of the reference changes form 3.9999 to 4.0001, the data in BV:4 will be sent to the specified output location. Likewise, if the value of the reference changes from 4.0001 to 3.9999, then the data in BV:3 will be sent. The data sent to the output location remains in a latched state until the next sub-interval transition.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The given "BV" variable can be sent to one of four destinations as defined by the setting of parameter PRM:162. The four possible modes are as follows.

0 - Data output disabled.

1 - Send the data in the given "BV" variable to the external output register REG:2. Only the 8 least significant bits of the given "BV" variable are sent to this register.

2 - Send the data in the given "BV" variable to the external D/A output port. Only the 8 most significant bits of the given "BV" variable are sent to this port. These 8 bits form a 2's compliment number such that "0x80xxxx " (hexadecimal) represents -10 output and the "0x7fxxxx" represents a +10 volt output.

3 - Send the data in the given "BV" variable to the external port PV:<number> where "number" is set by parameter PRM:163. Unlike modes 1 and 2 described above, all 24 bits of the given "BV" variable are sent to the specified "PV" port.

4 - Send the data in the given "BV" variable to the variable BV:1000. This special output mode can be used in a variety of ways. One use can be as way of controlling the execution rate of scale for each individual "sub-interval" of the spline. The data in the "BV" array is sent to BV:1000 at each sub-interval transition and can be made to effect the "Scaled Trajectory Modifier" algorithm if variable BV:1000 is set by parameters PRM:144/PRM:146 (task 1) or PRM:148/PRM:150 (task 2) as the scaling modifier. This function can be useful in changing the time base reference rate at each sub-interval for the spline if parameter PRM:154 and PRM:155 are set to "1". If parameter PRM:154 is set to "2" (parameter PRM:155 can be set to any mode), the position axis scale factor can be changed at every sub-interval.

This mode is also useful for passing status flags to a "HOST" program (communicating through the RS-232 or IEEE-488 interface), or monitor program running in the unused task.

∇ ∇ ∇

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

APPENDIX D: PROCESSING ERROR CODES

Appendix D

In This Section:

- Pre-Compiler Errors D-2
- In-Process Compiler Errors D-2
- Post-Compiler Errors D-3
- Host Errors D-4
- Run Time Errors D-6

The U100/U100i check for three types of processing errors depending on current operating status. These errors can originate from the following sources.

Host	These types of errors occur in two forms. First, if in normal RS-232 communications mode where the "HT" terminal option or COM_100.EXE utility is running, these errors are visual prompts. For Host RS-232 or IEEE-488 communications, these errors are in the form of numbers read after the host error service request "3" is sent by the U100.
Compiler	These types of errors may occur when a "RUN" command is issued. These errors are displayed as numbers through prompts when in normal communications mode, or read as numbers in registers REG:14 and REG:15 in Host communications mode. Note that only compile errors 200 through 299 use the line register (REG:14). This register denotes the line error relative to the LST1 file.
Run Time	These errors are valid only when Task 1 and/or Task 2 are executing a successfully compiled PGM program. These errors, if they occur, are read in registers REG:16 for Task 1 and REG:17 for Task 2. In normal communications mode, (e.g., RS-232 running on the HT or COM_100.EXE utility), runtime errors are only evident by monitoring REG:302 or viewing the GENERAL window of the STATUS menu for a FAULT condition. This limited form of user runtime error detection is required because of the controller's multi-tasking environment. In Host mode however, an SRQ mask can be set to generate a service request to the Host upon detection of a Run Time error.

In addition, the compiler errors are broken into three sections, one for each phase of compilation. The detailed description of these errors along with the transfer and Run Time errors are described in the sections that follow.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

D.1. Pre-Compiler Errors

Pre-compiler errors are errors that the controller may encounter before compiling a PGM program. These errors are normally a result of failure to locate specified items or insufficient work area. For a summary of these error codes, see Table D-1. For these errors, the line number of register (REG:14) only denotes the last line of the generated LST1 file, and this cannot be used to locate the error.

Table D-1. Pre-Compiler Error Codes

Code	Description
101	Encountered an illegal string before the "BEGIN" or could not find the "BEGIN".
102	Failed attempt to locate the specified "DEFx" file.
103	Failed attempt to locate the specified "MACx" file.
104	Encountered a "MACx" file that does not contain an "ENDMAC" ending statement.
105	Failed attempt to process the "DEF" file due to insufficient work space. (Max. 500 characters)
106	Failed attempt to process the "MAC" file due to insufficient work space. (Max. 500 characters)
107	Failed attempt to process the "PGM" file due to insufficient work space. (Max. 500 characters)
108	More than ten “DEF” files specified in PGM file.
109	More than ten “MAC” files specified in PGM file.

D.2. In-Process Compiler Errors

An In-process Compiler error is an error encountered while compiling a PGM program. These errors are usually due to command format errors, out-of-range values, and missing commands. For this error, the line number (REG:14) of the statement containing the error is found in the LST1 file. For a summary of these error codes, see Table D-2.

Table D-2. In-Process Compiler Error Codes

Code	Description
201	General Format Error. This error is usually associated with improper statement syntax or an unrecognizable statement.
202	Missing an "EXIT" command before the first specified "SUB:x" or missing an "ENDSUB" command before the subsequent "SUB:x" commands.
203	Encountered an "ENDSUB" without accompanying a "SUB:x" command.
204	Encountered an "ENDIF" command without accompanying an "IF" command.
205	Reserved.
206	Encountered an "ENDWHL" command without accompanying a "WHL" command.
207	Reserved.
208	Encountered an "ELSEIF" command without accompanying an "IF" command.
209	Reserved.
210	Encountered an "ELSE" command without accompanying an "IF" command.
211	Encountered an "ENDIF" command within an un-terminated "WHL" command.
212	Encountered an "ENDWHL" command within an un-terminated "IF" command.
213	Parameter index number is "reserved" or out of range.
214	Register index number is "reserved" or out of range.
215	Register access error (Due to an invalid "read" or "write" access). This error occurs if a Read Only register is placed at the left of the "=" sign or a Write Only register is placed in the variable (or variables) position to the right of the "=" sign.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table D-2. In-Process Compiler Error Codes Cont'd

Code	Description
216	A declared string exceeded the 20 characters or maximum allowable string declarations exceeded. PRM:332 sets the maximum number of string declaration for Task 1 and Task 2.
217	Exceeded the maximum number of "GOTO:xx" statements.
218	Exceeded the maximum number of "LB:xx" statements.
219	Exceeded the maximum number of "GOSUB:xx" statements.
220	Exceeded the maximum number of "SUB:xx" statements.
221	Exceeded the maximum number of "IF" statements.
222	Exceeded the maximum number of "WHL" statements.
223	Exceeded the maximum number of "ELSEIF" statements within current "IF" progression.
224	Exceeded the maximum number of nested "IF" statements.
225	Exceeded the maximum number of nested "WHL" statements.
226	Out of range index value for "BV", "LV", "FV", "PV" or "SV" variables.
227	Exceeded the maximum "command structures" allocated for this task. The maximum number of command structures (e.g., PGM command statements) for Task 1 and Task 2 is set by PRM:330.
228	Exceeded maximum "extension structures" allocated for this task. The maximum number of "extension structures" (e.g., PGM command statements containing one or more REG:xx and/or PRM:xx references) for Task 1 and Task 2 is set by PRM:331.

D.3. Post-Compiler Errors

Post-compiler errors are errors detected after completing the in-process compile phase. These errors normally result from missing labels, invalid command duplications, or invalid writing of IF/WHL statements. For these errors, the line number register REG:14 only denotes that the last line of the LST1 file and cannot be used to locate the error. For a summary of these error codes, see Table D-3.

Table D-3. Post-Compiler Error Codes

Code	Description
301	Encountered a specified "WHL" statement that is missing the "ENDWHL" ending statement. (Usually this is associated with the last specified "WHL" command or the most "outer" "WHL" command of a nested "WHL" statement.)
302	Encountered a specified "IF" statement that is missing the "ENDIF" ending statement. (Usually this is associated with the last specified "IF" command or the most "outer" "IF" command of a nested "IF" statement.)
303	Encountered duplicate "LB:x" statements.
304	Encountered duplicate "SUB:x" statements.
305	Reserved.
306	Encountered a "GOSUB:x" command without an associated "SUB:x" statement.
307	Failed attempt to locate an "END" statement.
308	Encountered a "GOTO:x" statement without an associated "LB:x" statement.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

D.4. Host Errors

The Host errors are available for the Host Mode operation and used with SRQ #3 to provide error information about the controller to a host controller. For a summary of these error codes, see Table D-4.

Table D-4. Host Errors Codes

Code	Description
0	No Error
1	General command sequence error.
10	Attempted to erase a file not found in user memory.
11	Attempted to make a copy of a file not found in user memory.
12	Cannot "reset" an active limit condition. This error occurs if a "limit reset" command is issued but the associated hardware or software limit is still active after the execution of this command.
20	Exceeded the minimum limit of parameter input. The parameter was loaded with this minimum value.
21	Exceeded the maximum limit of parameter input. The parameter was loaded with this maximum value.
22	Invalid data input was detected for parameter or variable. Current parameter or variable value remains unchanged.
40	Detected a "BV:xx" variable number "xx" that is out of range or has an error in format.
41	Detected a "LV:xx" variable number "xx" that is out of range or has an error in format.
42	Detected a "FV:xxx" variable number "xx" that is out of range or has an error in format.
43	Detected a "PV:xx" variable number "xx" that is out of range or has an error in format.
50	Detected an invalid program number format for "file to copy from".
51	Detected an invalid program number format for "file to copy to".
60	Attempted to "upload" a file to the host that does not exist.
61	Failed attempt to "download" a file (checksum mismatch).
80	Attempted to access a parameter reserved for future use.
90	Attempted to access a register reserved for future use.
91	Invalid data entered to a "write only" register.
100	Detected a string character buffer number that is out of range.
120	Attempted to run a "PGM" file or "MDI" command in a task that is already active.
121	Attempted to run a "PGM" file that does not exist.
122	Detected an error when compiling the selected "PGM" file or "MDI" command. Check the "compiler error" REG:015 for the specific error.
123	Insufficient user memory for file copying, LST1 generation, or DIR1 generation.
124	Axis scope processing error.
312	Attempted to read from or write to a library image with no memory board installed.
313	Attempted to write to a library image or machine image while Task 1 and/or Task 2 is a running a program.
314	Attempted to download to the memory board an unprogrammed or corrupted library image.
315	Error encountered in erasing a machine or library image.
316	Error encountered in writing to a library image.
317	Attempted to read a machine image that has not yet been programmed.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The following error codes apply only to the LIBRARY utility mode. Refer also to the *UNIDEX 100 Memory Expansion Board Option Manual P/N EDU136.*

Table D-5. Host Errors Codes [Library Utility Mode Only]

Code	Description
202	Corrupted file discovered during library file search. If this error occurs, the area on the memory board where the library files exist has been corrupted. This is a non-recoverable error and requires that all library files be erased (by setting PRM:314 to 512 or executing a SETUP). This error can occur if a PV:xx variable below the current "top of library" specified by PRM:314 has been changed through the "SETUP" menu.
204	Insufficient memory to store specified library file. This error occurs if an "INSERT" of a library file at the top of memory was detected. This is a recoverable error. Increase PRM:316 or use the "DELETE" item of the LIBRARY utility to free up unused programs in order to make more memory available.
205	Attempted to enter the LIBRARY utility while Task 2 is running a program. Access to the LIBRARY utility is inhibited if Task 2 is running a program.
312 *	Attempted to read from or write to a library image with no memory board installed.
313 *	Attempted to write to a library image or machine image while Task 1 and/or Task 2 is a running a program.
314 *	Attempted to download to the memory board an unprogrammed or corrupted library image.
315 *	Error encountered in erasing a machine or library image.
316 *	Error encountered in writing to a library image.
317 *	Attempted to read a machine image that has not yet been programmed.

* May appear when in HT mode (menu mode via com_100.exe) when using the ARCHIVE utility.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

D.5. Run Time Errors

The Run Time errors are errors detected while executing a PGM program. These errors are normally the result of program conditions that may not be present during the compile process. Run-time error message codes can be view in REG:16 (for programs running is Task 1) or REG:17 (for programs running in Task 2). As one of the improvements to the 2.00 firmware, the *CONTROL FAULT* LED will be energized if a run-time error is detected. For a summary of these error codes, see Table D-6 below.

Table D-6. Run Time Error Codes

Code	Description
100	Detected a change in direction within a velocity profiling move (e.g., multiple "D( )" commands within the same "GO" command had different polarities).
101	The ending segment of a velocity profiling move contains insufficient distance to perform a proper decel to the ending position (e.g., last "D( )" command before the "GO" command has insufficient distance to decel motion..
102	Exceeded the programmed maximum trajectory size while interpreting a motion command string (e.g., processes 20 motion segments before the "GO" command was specified.
103	Attempted to send an invalid Service Request Number (e.g., "SRQ( )" commands can only have 1 through 15 specified).
104	Dynamic BV:xx index range error (e.g., BV:BV:xx resolved to "Out of Range" BV variable).
105	Dynamic LV:xx index range error (e.g., LV:BV:xx resolved to "Out of Range" LV variable).
106	Dynamic FV:xx index range error (e.g., FV:BV:xx resolved to "Out of Range" FV variable).
107	Dynamic SV:xx index range error (e.g., SV:BV:xx resolved to "Out of Range" SV string).
108	Dynamic PV:xx index range error (e.g., PV:BV:xx resolved to "Out of Range" PV variable).
109	Reserved
110	Error detected on home execution. During the "HM" command execution, a limit switch other than the specified Home Limit became active.
111	Detected a value less or equal to zero for a "V( )" command during trajectory building.
112	Attempted to execute the “TUNE()” command with an incompatible motor.
113	The execution of “TUNE()” resulted in the identification of incorrect parameter values for PRM:218, PRM:219, and PRM:220. Present values for these parameters remain unchanged.
114	Reserved
115	The file number specified in the command CAM(<number>) could not be found in user program memory
116	Failed to find a spline point and/or spline output variable on a given line in a CAM file. (This error usually occurs if 80 or more characters are detected in a CAM file before the detection of a “line-feed” character.
117	Detected an invalid numerical constant for a spline point in a CAM file. (Spline point data must be decimal integers [24 bit maximum] or fixed/floating point representations and must be in the second column of data [left side of the spline output data, if enabled by parameter PRM:162]. This error can also be caused by two successive “line-feed” characters [e.g., blank lines of data in the CAM file].
118	Detected an invalid numerical constant for a spline output in a CAM file. Spline output data can only be octal, decimal, or hexadecimal integer (24 bit maximum) and must be in the second column of data (right side of the spline point data). This error can also be caused by missing data in the spline output column of the CAM file when PRM:162 is enabled.
119	Less than four spline point data entries were detected in a CAM file. Data entries must be greater than or equal to four in order to compute a cubic spline.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table D-6. Run Time Error Codes Cont'd

Code	Description
120	The number specified in the command CAM(<number>) can not be less than zero or greater than 9999.
121	Attempted to execute the "CAM()" command with two low a task time slot number. (e.g., For task 1, PRM:301 must be set to at least 20. Similarly for task 2, PRM:302 must be set to at least 20).
122	Attempted to execute the "GEAR()" with PRM:231 set to "2" or "4" (e.g., interrupt mode), with the interrupt latch source (PRM:132) not equal to the "gear" source (PRM:232).
212	The parameter 2 of "RUN(<var or constant>,<BV: or constant>) can only be a BV: variable or integer constant.
213	The value specified for parameter 2 of "RUN()" command can only have a range from "0" through "4".
214	Library image specified by "RUN()" command has not been programmed.
215	The parameter 2 of "PVI(<var or constant>,<BV: or constant>) can only be a BV: variable or integer constant.
216	The value specified for parameter 2 of "PVI()" command can only have a range from "1" through "4".
217	The value specified for parameter 1 of "PVI()" command can only have a range between PRM:316 (image) and "0xdfff".
218	The library image specified by parameter 2 of "PVI()" command has not been programmed.
312	Attempted to read from or write to a library image with no memory board installed.
314	Attempted to download to the memory board an unprogrammed or corrupted library image.
315	Error encountered in erasing a machine or library image.
316	Error encountered in writing to a library image.
317	Attempted to read a machine image that has not yet been programmed.
319	The value specified for parameter 1 of "ARCH()" command can only be "1" for READ or "2" for "WRITE".
320	Cannot execute "ARCH()" "WRITE" command if "D()..GO", "DD()", "HM", or "CAM()" commands are being executed in other task.
321	Parameter 2 of "ARCH(<var or constant>,<BV: or constant>) can only be a BV: variable or integer constant.
322	The value specified for parameter 2 of "ARCH()" command can only have a range from "0" through "0x3f" (decimal 63).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The following error codes apply only to the RUN( ) command. To use the RUN command, the MEM option board must be installed. See the *UNIDEX 100 Memory Expansion Board Option Manual P/N EDU136* for additional information.

Table D-7. Run Time Error Codes [RUN ( ) Command Only]

Code	Description
201	The specified library file is not found. This error usually occurs if the PGM program number specified in the RUN( ) command was not compiled and loaded into the memory board using the "INSERT" function of the LIBRARY utility.
202	The specified library file that was found is corrupted. This error is an unrecoverable fault requiring the entire library storage area be erased by setting PRM:314 to 512 or executing a "Setup". Corruption can be caused by writing to a PV:xx variable using the SETUP menu whose address resides within the library storage area.
206	The command buffer was exceeded during loading. If this error occurs, PRM:330 needs to be decreased to increase the "command" buffer area for Task 2.
207	The extension buffer was exceeded during loading. If this error occurs, PRM:331 needs to be decreased to increase the "extension" buffer area for Task 2.
208	The string buffer was exceeded during loading. If this error occurs, PRM:332 needs to be decreased to increase the "string" buffer area for Task 2.
209	Detected file corruption during the loading process. This error is similar to error number 202 above except corruption was detected during the loading process instead of the search process. Like errors 202, this fault is unrecoverable and requires that the entire library be erased.
210	The LIBRARY utility was in use when a run command encountered or attempted to execute a "RUN( )" command in Task 2 or attempted to execute a "RUN( )" command without the MEM option board installed.
211	Attempted to execute a "RUN( )" command while processing a previous "RUN( )" command. This error can only occur if PRM:315 is set to a "2" indicating concurrent program call execution. This error occurs if a RUN( ) command attempts to execute a program while Task 2 is busy (e.g., still running a program from a previous "RUN( )" command).

∇ ∇ ∇

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

SYMBOLS

- command, 5-88
- Software Limit parameter, 6-32
"S" Curve Generation parameter, 6-27
"S" Curve Profiling, 6-27
* command, 5-77
/ command, 5-51
; Character, 8-1
|Velocity Loop, 10-4
+ command, 5-31
+ Software Limit parameter, 6-32
+/- Limit Reset Distance parameter, 6-32
= command, 5-60
0x, 5-14, 12-2
100, 1-2
25-pin "D" Style Connector, 3-6
80386 microprocessor, 4-3
80486 microprocessor, 4-3

A

ABS command, 5-28
Absolute Positioning Mode, 5-29
Absolute Value, 5-28
AC brushless motor wiring, 9-7, 9-8
AC Line Fuse, 10-2
AC Servo Motor Control, 6-28
Accel/Decel Ramp Rate in User units/sec^2, 5-27
Acceleration feedforward
  Kf, 11-2
Acceleration Feedforward, 11-7
Acceleration Feedforward Compensation, 11-7
Acceleration/Deceleration Feedback register, 7-24
Acceleration/Deceleration Trap, 3-5
Accessing Thumbwheel #2 Address, 3-19
Accumulated Motor Position, 7-23
Acknowledge of Service Request, 6-18
Activate LST1 File Generation, 6-17
ADC command, 5-30, 6-37
Add, 5-31
Address Character, 6-6
Address Status Byte Information, 7-19
Adjusting the SRQ Character, 5-14, 12-2
Analog input, 9-20
Analog output, 9-20
Analog to Digital Conversion, 5-30
Analog to Digital Deadband parameter, 6-37
Analog to Digital Scale Factor parameter, 6-37
AND command, 5-32
Antistatic Bag, 2-3
Archive Boot Status, 7-9
Archive command, 5-32
Assign a value to, 5-60
Assign Names to Strings, 5-22
Assign Names to Variables, 5-22
Auto DEF100/MAC100 Lookup in MDI Mode, 6-17
Auto mode, 4-10
Autotune, 5-93
Auxiliary encoder, 9-9
Auxiliary Encoder, 3-24
Auxiliary Encoder Input, 8-9
Auxiliary encoder port, 9-9
Auxiliary Encoder Port, 3-6, 7-10
Auxiliary Mask parameter, 5-12
Auxiliary Output Setting, 6-74
Averaged Velocity Command register, 7-23
Averaged Velocity Feedback register, 7-24
Axis Calibrated Position Command register, 7-24
Axis Calibrated Position Feedback register, 7-24
Axis Calibration Backlash parameter, 6-56
Axis Calibration Compensation register, 7-24
Axis Calibration Increment Size parameter, 6-56
Axis Calibration Mode, 7-24
Axis Calibration Mode parameter, 6-55
Axis Calibration Table Size parameter, 6-55
Axis Status register, 7-27

B

Background Operations, 5-3
Backlash Compensation, 6-56
Ballscrew Drive Application, 11-3
Base 10 Notation, 7-32
Base 16 Notation, 7-32
Base 8 Notation, 7-32
Begin command, 5-33
BEGIN command, 5-22
Begin Move, 5-64
Bi-stable Output, 7-10
Bit Input register, 7-4
Bit Output register, 7-4
Bits, 3-19
Block Mode, 4-9, 4-10
Boot prgram
  Task 2, 6-18
Boot program
  Task 1, 6-18
Borrow and carry control outputs of the counter chip, 8-18
Borrow Toggle Flip-flop bit, 7-13
Borrow Underflow, 7-13
Brushless Commutation Loop, 6-62
Brushless Commutation Step/Cycle parameter, 6-57
Brushless Commutation Table Step Shift parameter, 6-58
Brushless Commutation Type parameter, 6-57
Bus addresses
  Display, 3-21
  INT option, 3-26

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Thumbwhell, 3-18
BV variables, 7-31
BWT bit, 7-13

C

C Language Library, 4-1
Calibration Error Compensation, 6-56
CAM command, 5-33
CAM files, 5-23
CAM Table Execution, 8-11
Cam trajectory generation, 6-22
CAM trajectory generation, 6-23
Carry Toggle Flip-flop bit, 7-13
CBI command, 5-36
Change Existing File, 4-6
Changing a Motion Trajectory, 8-12
Changing Jumper Selections, 3-26
Changing Parameters, 4-10
Character code
End of file, 6-17
CIB command, 5-37
Circuit, Latch, 3-26
Clearing the Reset Latch, 3-26
CLN command, 5-38
Closed Loop Stepper Velocity, 6-48
CLS command, 5-39
CNTR Counter, 7-13
CNTR Over flow, 7-13
CNTR Status, 7-12
COM Communications Port, 4-27
COM port, 3-3
COM_100.EXE, 4-1
Combine Task 1 and Task 2 Time Slots parameter, 6-67
Command
Begin, 5-33
CLS, 5-39
Command Buffer Partition parameter, 6-80
Command Entry, 4-10
Command Segment Filter Coefficient parameter, 6-26
Command Segment Filter On/Off parameter, 6-26
Command Summary, C-1
Commands, 5-24
*, 5-77
/, 5-51
+, 5-31
=, 5-60
A, 5-27
ABS, 5-28
ABSL, 5-29
ADC, 5-30
AND, 5-32
BEGIN, 5-22
CBI, 5-36
CIB, 5-37
COS, 5-40
CUR, 5-41
D, 5-42
DAC, 5-45
DD, 5-46
DEC, 5-47
DEF. See
DI, 5-9, 5-50
DW, 5-52
EI, 5-9, 5-52
ELSE, 5-53
ELSEIF, 5-54
END, 5-22, 5-55
ENDIF. See
ENDMAC, 5-57
ENDSUB, 5-58
ENDWHL, 5-59
EXIT, 5-60
GC - Get Character, 5-62
Gear, 5-62
GM, 5-63
GO, 5-64
GOSUB, 5-65
GOTO, 5-66
HM, 5-67
IF, 5-69
INC, 5-70
INCR, 5-71
LB, 5-72
LOCK, 5-73
MAC, 5-74
MDX, 5-76
OR, 5-78
PC (Print Character), 5-78
PM, 5-79, 5-80
PVI, 5-81
RI, 5-9, 5-82
RUN, 5-83
SIN, 5-84
SQRT, 5-86
SRQ, 5-15, 5-85, 12-3
SUB, 5-87
SYNC, 5-89
T, 5-90
TAN, 5-91
TITLE, 5-92
Tune, 5-93
V, 5-94
WHL, 5-95
XOR, 5-96
Comment Statements, 8-1
Comments
limiting the number of, 8-1
Communicating with Opto 22 Modules, 3-26
Communication Difficulty, 4-26
Communication parameters, 6-2
Communication port (P1), 9-15
Communication Registers, 7-2

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Communications port, 9-14
Communications Problems, 10-2
Commutation, 6-57
Commutation Loop Update Time, 5-3
Commutation Loop Update Time parameter, 6-62
Comparator Outputs, 8-16
COMPARE Outputs, 7-10
Compare Toggle Flip-flop bit, 7-13
Compiler Error Line Number register, 7-10
Compiler Error register, 7-10
Compiler Errors, D-1
COMPT bit, 7-13
CONFIG.SYS file, 2-9, 4-4
CONFIG.SYS File
  customizing, 2-8, 4-3
Configuring I/O Points as
    Inputs. See
    Outputs. See
Configuring PB24 Modules, 3-26
Control board, 9-1.
Control Fault LED, 3-4
Control Loop, 5-3
Control Loop Checks parameter, 6-67
Control Menu, 4-10
Control Mode, 4-10
Coprocessor, Recommended, 2-3
Copy files, 4-16
Corrupt Parameters, 4-26
Corrupted Battery Backed Memory, 3-24
COS command, 5-40
COS Encoder Signal, 3-6
Cosine, 5-40
COS-N Encoder Signal, 3-6
Counter Status, 7-12
Create New File, 4-6
CUR command, 5-41
Current Command register, 7-23
Current Limit Trap, 11-7
Current Output Jumper Settings, 3-10
Current/Velocity Command Jumper, 3-9
Cursor Enable/Disable, 6-15
Customer Order Number, 10-1
CYT bit, 7-13

D

D command, 5-42
DAC command, 5-45
DAC Value, 6-37
Daisy chain enable/disable, 6-6
Daisy chain operation, 6-7
Data Latching, 3-19
Data Logging On-the-fly, 8-8
Data Output Procedures, 3-26
Data Types
  Parameters, 4-17
  Registers, 4-17
  Strings, 4-17
  Variables, 4-17
DC Brush Motor Wiring, 9-4, 9-5
DC Servo Motor Control, 6-28
DD command, 5-46
DD Command Scale Factor parameter, 6-28
DD Trajectory command, 6-28
DEC command, 5-47
Decrement, 5-47
DEF command, 5-48
DEF file, 5-22
Default Acceleration, 6-27
Default Acceleration/Deceleration parameter, 6-27
Default Deceleration, 6-27
Default Distance parameter, 6-27
Default Ramp Time parameter, 6-27
Default Velocity parameter, 6-27
Definition File, 5-22
Definitions of Terms, A-1
Descriptions
  Hardware, 3-1
DI command, 5-9, 5-50
Digital input specifications, 9-17
Digital output specifications, 9-19
Digital Signal Processor, 9-1
Digital to Analog Conversion, 5-45
Digital to Analog Scale Factor parameter, 6-37
Direct Drive, 5-46
Direct drive generation, 6-22, 6-23
Direction of Home Switch, 6-29
Directory of Files, 4-8
Disable "S" Curve Trajectory Generation, 6-27
Disable Accel/Decel Check, 6-67
Disable Buffer Updating, 6-61
Disable Command Segment Filter, 6-26
Disable Current Limit Time-Out Check, 6-67
Disable Encoder Feedback Verification, 6-48
Disable Group Trigger Mode, 6-14
Disable Interrupt, 5-50
Disable Interrupt Operation, 5-9
Disable LST1 File Generation, 6-17
Disable Mask parameter, 5-12
Disable Position Error Check, 6-67
Disable Stepper Damping, 6-49
Disable Velocity Check, 6-67
Disable Velocity Feedforward to PID Control Loop, 6-51
Disk Space, Recommended, 2-3
DISP option, 8-6
Display, 1-5
Display configuration, 3-20
Display option, 3-20
Display, Recommended, 2-3
Distance in User Units to move/position, 5-42
Divide, 5-51
DOS Version, Recommended, 2-3
Downloading Files, 4-22

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Drive Fault LED, 10-4
Drive Fault" LED, 3-4
Drive parameters, 6-46
Drive Registers, 7-22
Dual Loop Control, 11-2
DW command, 5-52
Dwell Time, 5-52

E

EI command, 5-9, 5-52
Electrical characteristics
  Opto-isolated input, 9-18
  Opto-isolated output, 9-19
Else, 5-53
ELSEIF command, 5-54
Enable "S" Curve Trajectory Generation, 6-27
Enable 16/14 Auto Select Mode, 3-23
Enable Accel/Decel Check, 6-67
Enable Buffer Updating, 6-61
Enable Command Segment Filter, 6-26
Enable Current Limit Time-Out Check, 6-67
Enable Encoder Feedback Verification, 6-48
Enable Group Trigger Mode, 6-14
Enable Interrupt, 5-52
Enable Interrupt Operation, 5-9, 6-34
Enable Position Error Check, 6-67
Enable Position Feedback from R/D, 3-23, 3-31
Enable Stepper Damping, 6-49
Enable Velocity Check, 6-67
Enable Velocity Feedback from R/D, 3-23
Enable Velocity Feedforward to PID Control Loop, 6-51
Enable/disable cursor, 6-15
ENC Board, 10-4
ENC option, 6-65
ENC Position register, 7-21
ENC System Configuration, 10-4
ENC Update Time, 5-3
ENC Update Time parameter, 6-65
ENC Velocity register, 7-21
Encoder
  Auxiliary, 9-9
  primary, 9-8
Encoder 1 Position register, 7-20
Encoder 1 Update Time parameter, 6-63
Encoder 1 Velocity register, 7-21
Encoder 2 Position register, 7-20
Encoder 2 Update Time, 5-3
Encoder 2 Update Time parameter, 6-64
Encoder 2 Velocity register, 7-21
Encoder Connection, 10-4
Encoder Counter, 10-5
Encoder Counter Chips, 8-16
Encoder counter circuits, 7-11
Encoder Counter Programming, 10-5
Encoder Input Ports, 7-10
Encoder interface, 9-9
Encoder marker signal, 9-11
Encoder phasing, 9-10
Encoder Port, 10-5
Encoder Port Configuration, 10-4
Encoder Sampling Frequency Jumper, 3-8
Encoder/Limits/Hall Effect Port, 3-6
END command, 5-22, 5-55
End of Data Block Detection, 6-13
End of file, 6-17
End Subroutine, 5-58
ENDIF command, 5-56
ENDMAC command, 5-57
ENDSUB command, 5-58
ENDWHL command, 5-59
Entering Commands, 4-10
Entire Program Execution, 4-9
Erasing a File, 12-7
Error Line Number Locating, 7-10
Error of Axis Calibration, 7-24
Error Processing, 5-3
Error Status "-" Direction Mask parameter, 6-79
Error Status "+" Direction Mask parameter, 6-78
Error Status "Active Mask" register, 7-30
Error Status "Overlay" register, 7-30
Error Status "Reset Mask" register, 7-30
Error Status Auxiliary Mask parameter, 6-74
Error Status bits, 6-69
Error Status Disable Mask parameter, 6-72
Error Status Freeze Mask parameter, 6-75
Error Status Invert Mask parameter, 6-68
Error Status Latch Mask parameter, 6-71
Error Status register, 5-11, 7-26
Error Status Service Request Mask parameter, 6-73
Error Status Task 1 Mask parameter, 6-76
Error Status Task 2 Mask parameter, 6-77
Errors, D-1
Example Program, 8-4
Exceeded Motor Speed, 10-5
Exception Processing, 5-3, 5-9, 6-66, 6-68
Exception Processing - Task 1, 6-31
Exception Processing - Task 2, 6-31
Exception Processing Algorithm, 7-30
Exclusive OR, 5-96
Execute a Program, 4-9
Execute Address, 3-19
Execute Switch Closure, 3-19
EXIT command, 5-60
Exit Program Operation, 5-60
Extension Buffer Partition parameter, 6-80
Extension Bus Interrupt Line, 6-35
External Input register, 7-8
External Interrupt Input, 8-12
External Interrupt Latch Source parameter, 6-33
External Interrupt Mode parameter, 6-34
External Interrupt Position Latch register, 7-21
External Interrupt Source parameter, 6-35

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

External Interrupt Type, 6-35

F

F_switch, 11-2, 11-7
Factory Set Parameters, 4-1
Fast Correction Velocity, 6-49
Fast Feedhold Execution, 6-75
Fault Condiitions, 4-1
Fault Mask Update Time, 5-3
Fault Mask Update Time parameter, 6-66
Feedback Device, 3-24
Feedback errors, 3-4
Feedhold, Program, 4-10
Field Service Information, B-1
Field Service Policy, B-1
File Changes, 4-6
File copying, 4-16
File Directory, 4-8
File Transfers, 4-22
File types, 5-17
Filter Coefficient for First Order Filter, 6-26
First order filter, 6-23
Floating Point, 7-32
Floppy disk
  UT100 software, 2-9
Floppy Disk Drive, Recommended, 2-3
Freeze Mask parameter, 5-12
Function keys
  Keyboard, 2-11, 4-5
Fuses
  Location, 3-13
FV variables, 7-31

G

GDS100 software, 1-5
Gear command, 5-62
Gear Mode, 6-54
Gear Scale Factor, 6-55
Gear Source, 6-54
General Status, 4-11
General Status Screens, 4-12
General trajectory, 6-22
General trajectory generation, 6-22
Generating a File Directory, 12-7
Get Character command, 5-62
Get Message for Variable, 5-63
Glossary of Terms, A-1
GM command, 5-63
GO command, 5-64
Go To a specified Subroutine, 5-65
Go To Specified Label, 5-66
GOSUB command, 5-65
GPIB interface, 7-19
Graphics Display, Recommended, 2-3

H

Hall effect devices, 9-8
Hall effect input circuit, 9-12
Hall effect interface, 9-12
Hall effect motor phasing, **9-13**
Hall effect Switches, 10-3
Hand held terminal, 1-5
Hand held Terminal, 2-5, 3-14
  HT, 2-12
Handheld terminal, 4-27
Handling the U100 interface cards, 2-4
Hard Disk Space, Recommended, 2-3
Hardware configurations, 3-1
Hardware descriptions, 3-1
Hardware Home, 5-67
Hardware requirements, 4-3
Hardware Status Indicators, 6-68
High Inertia Loads, 6-27
High Inertial Loads, 6-50
High Value Parameter Settings, 3-24
Highly Compliant Mechanical System, 3-24
Home cycle, 3-4
Home Ending Offset parameter, 6-31
Home limit, 9-11
Home Limit Offset parameter, 6-30
Home Marker Search Rate, 6-31
Home Marker Selection parameter, 6-30
Home Marker Velocity parameter, 6-31
Home Switch Direction parameter, 6-29
Home Switch Fast Reference parameter, 6-29
Home Switch Selection parameter, 6-29
Home Velocity After Power Up parameter, 6-30
Horizontal Screen Scrolling, 4-6
HOST Command Set, 12-1
Host Error Code, 12-1
Host Errors, D-1, D-4
Host Mode Block Executes, 12-17
Host Mode Feedhold, 12-17
Host Mode File Directory, 12-7
Host Mode Move Out of a Limit, 12-17
Host Mode Parameter Value Modification, 12-9
Host Mode Parameter Value Retrieval, 12-10
Host Mode Read Only Register Retrieval, 12-12
Host Mode Read/Write Register Modification, 12-13
Host Mode Read/Write Register Value Retrieval, 12-14
Host Mode String Storage Buffer Value Modification, 12-16
Host Mode String Storage Buffer Value Retrieval, 12-16
Host Mode Task Control Functions, 12-17
Host Mode Task Quit, 12-17
Host Mode Transfer Files, 12-8
Host Mode Variable Value Modification, 12-15
Host Mode Variable Value Retrieval, 12-15
Host Mode Write Only Register Modification, 12-11

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

HOST_100 Software, 6-2
HOST_100.EXE, 4-1, 4-2, 5-35
HT, 4-27
  Hand held terminal, 2-12
HT Handheld Terminal, 8-5
HT Terminal Emulator, 4-1

I

I/O interface, 9-16
I/O Port, 3-6
I/O Port functions, 9-16
I_switch, 11-2, 11-7
IBM PC, 4-3
ICR, 7-15
IEEE-488 Address Mode, 6-10
IEEE-488 Address Status register, 7-19
IEEE-488 Command Pass Thru register, 7-17
IEEE-488 Communications, 6-2
IEEE-488 Communications Mode, 6-14, 12-1
IEEE-488 Data In register, 7-19
IEEE-488 Data Out register, 7-19
IEEE-488 Host Communication Mode, 5-14, 12-2
IEEE-488 interface card, 1-5
IEEE-488 Internal Counter, 6-13
IEEE-488 option, 3-29, 7-17, 7-18, 7-19
IEEE-488 pinout description, 3-30
IEEE-488 program examples, 8-20
IEEE-488 Serial Poll Status register, 7-18
IEEE-488 Status Codes, 5-14, 12-2
IEEE-488 Status Register 1 register, 7-18
IEEE-488 Status Register 2 register, 7-18
IF command, 5-69
If/Then, 5-69
In Position “Wait” parameter, 6-28
In Position LED, 3-5
Inactive State of Outputs, 3-26
INC command, 5-70
Increment, 5-70
Incremental Motion, 8-1, 8-2
Incremental Position Mode, 5-71
Indexing Variables, 7-31
Inductosyn, 3-24
Inductosyn Electrical Cycle, 6-36
Inertia, 3-24
Inertia Load, 11-7
Inner Slip Tolerance Threshold, 6-49
In-position Position Error, 6-50
In-process Compiler Errors, D-2
Input Bits, 3-27
Input Control Register, 7-15
Input Queue Pointer, 7-8
Input Reading, 3-26
Inputs, 3-27
Inspecting the UNIDEX 100 Controller, 2-3
Installation of option boards, 3-33
Installation Problems, 10-2
Installing other components, 3-14
Installing the UT100 Software, 2-9, 4-4
Instantaneous Motor Position, 7-23
Instantaneous Ratioed Current Command, 7-23
Instantaneous Velocity Feedback register, 7-23
INT jumper and switch location, 3-29
INT jumper settings, 3-27
INT option
  Opto 22 boards, 3-24
Integer to BCD Conversion, 5-37
Integer Variables, 7-31
Integers, 7-32
Integral Gain Adjustment, 6-51
Internal counter functions, 7-12
Interrupt Programs, 5-8
Interrupt Response Conditions, 6-35
Invert Mask parameter, 5-10, 6-68

J

Joystick, 1-5, 3-2, 3-16
JP20 of Control Board, 10-2
JP8 of Control Board, 10-4
Jumper selections
  INT option board, 3-28
Jumpers, 3-8
  encoder sampling frequency, 3-8
  marker pulse qualification, 3-8

K

Kernel, 7-28
Kernel Interrupt Time Slot Chart, 5-3
Keyboard function keys, 2-11, 4-5
Kf Servo Loop Gain parameter, 6-51
Ki integral gain, 11-2
Ki Servo Loop Gain parameter, 6-51
Kp Servo Loop Gain parameter, 6-51
Kp velocity error, 11-2
Kpos, 11-2
Kpos Servo Loop Gain parameter, 6-51

L

Label, 5-72
Latch Circuit, 3-26
Latch Mask parameter, 5-11
Latch Position of External Interrupt, 7-21
Latch Register on Interrupt, 5-9
Latch Reset, 3-26
Latching a Position Feedback Interface, 6-33
Latching the Data, 3-19
Latching the Sign, 3-19
LB command, 5-72
Leadscrew Drive Application, 11-3
LED Status Indicators, 3-3
LEDs, 3-2

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

LF Keyword, 12-1
Library Ending Offset parameter, 6-67
Library File Generation, 6-67
Library File Storage, 6-68
Library Memory Limit parameter, 6-68
Library Program Call Mode parameter, 6-67
LIBRARY Utility Mode Errors, D-5
Limit Cycling, 11-7
Limit Reset Distance, 6-32
Limit Reset Menu, 4-14
Limits (+ and -) LEDs, 3-5
limits interface, 9-10
Limits port, 9-8
List files, 5-21
List of Files, 4-8
LOAD Input, 7-10, 8-14
Loading Parameters, 4-26
Location
  Fuses, 3-13
LOCK command, 5-73
Logical And, 5-32
Logical Functions, 3-27
Logical OR, 5-78
Long Variables, 7-31
Loop Address character, 6-6
Loop Control character, 6-6
Loop group trigger, 6-10
Loop Tuning, 11-1
Low Going Edge Condition, 6-35
Low Level Edge Condition, 6-35
Low Value Parameter Settings, 3-24
Lower "Modulo" Index Limit parameter, 6-28
LST1 file, 6-17, D-2
LV variables, 7-31

M

MAC command, 5-74
Macro files, 5-19
Main Menu window, 2-11, 4-5
Making a Copy of a File, 12-6
Marker Index Search Rate, 6-31
Marker LED, 3-4
Marker Pulse Qualification Jumper, 3-8
Masking Bits, 3-27
Master Control Register, 7-14
Master/Slave Move, 8-9
Mathematical Functions, 3-27
MCR, 7-14
MDI Mode, 4-10
MDX command, 5-76, 6-28, 8-11
MEM option, 6-67, 6-68
MEM option board, D-8
Memory
  conventional (recommended), 2-3
  extended, 2-3
Memory Status, 4-11, 4-13
Microprocessor, Recommended, 2-3
Micro-stepping Resolution of Stepper Motor, 6-48
Minimum Characters, Line 2, 6-20
Minimum Digits, Line 1, 6-19
Minimum hardware requirements, 4-3
Minimum Hardware requirements, 2-2
Miscellaneous Status Monitoring, 7-6
MKR Encoder Signal, 3-6
MKR-N Encoder Signal, 3-6
Modify the Value of a "Read/Write" Register, 12-13
Modify the Value of a "Write Only" Register, 12-11
Modify the Value of a Parameter, 12-9
Modify the Value of a String Storage Buffer, 12-16
Modify the Value of a Variable, 12-15
Modulo Index, 5-76
Mono-stable Output, 7-10
Motion controller, 2-5
Motion parameters, 6-24
Motion Registers, 7-20
Motion Status, 4-11
Motion Status register, 7-25
Motor configurations
  Type, 3-11
Motor connections, 3-3
Motor Drivers, 1-3
Motor Drivers, List of, 1-3
Motor Holding Current, 6-47
Motor rotation, 9-11
Motor Type Jumper Selections, 3-11
Motor wiring, 9-4
Motors
  AC brushless, 3-2
  DC brush, 3-2
  Stepping, 3-2
Mounting Rack #1, 3-26
Mounting Rack #2, 3-26
Mouse, Recommended, 2-3
Move Distance from Home Marker Position, 6-31
Move Out of a Limit, 12-17
Moving Out of a Limit, 4-14
Multiple programs, 5-7
Multiply, 5-77
Multitasking, 8-6
Multitasking Time Slots, 5-3

N

Negative Direction Mask parameter, 5-13
Negative Software Limit Threshold, 6-32
New File Creation, 4-6
No Feedback (Open Loops), A-6
Numerical Data Types, 7-32

O

OCR, 7-16
Offset from the Home Limit, 6-30

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

OL Output Port, 7-13
Open Loops, A-6
Operator Input from HT, 8-5
Option Ports, 3-7
Opto 22 Modules, 3-26
Opto-22, 1-5
Opto-isolated Inputs, 7-4
Opto-isolated Outputs, 7-4
Opto-isolators, 10-6
OR command, 5-78
Order information, 1-2
OSR register, 7-12
Outer Slip Tolerance Setting, 6-49
Output Control Register, 7-16, 8-16
Output Current to Stepping Motor, 6-47
Output Data, 3-27
Output Latch, 7-13
Output power, 9-4
Output Queue Pointer, 7-9
Output Status Register, 7-12
Output-on-the-fly, 8-2
Outputs Inactive State, 3-26
Overload LED, 3-5, 10-5
Overshoot, 6-50
Overview of the UNIDEX 100 System, 1-1

P

P2, 3-19
P3, 3-19
P3 Connector, 3-6
Parallel Poll Response, 6-13
Parameter
  Servo Velocity Trap, 6-52
Parameter Loading, 4-26
Parameter Settings, 4-1
Parameter Summary, A-1
Parameters, 6-1
  - Software Limit, 6-32
  "S" Curve Generation, 6-27
  + Software Limit, 6-32
  +/- Limit Reset Distance, 6-32
  Analog to Digital Deadband, 6-37
  Analog to Digital Scale Factor, 6-37
  Auxiliary Mask, 5-12
  Axis Calibration Backlash, 6-56
  Axis Calibration Increment Size, 6-56
  Axis Calibration Mode, 6-55
  Axis Calibration Table Size, 6-55
  Brushless Commutation Step/Cycle, 6-57
  Brushless Commutation Table Step Shift, 6-58
  Brushless Commutation Type, 6-57
  Combine Task 1 and Task 2 Time Slots, 6-67
  Command Buffer Partition, 6-80
  Command Segment Filter Coefficient, 6-26
  Command Segment Filter On/Off, 6-26
  Commutation Loop Update Time, 6-62
  Control Loop Checks, 6-67
  DD Command Scale Factor, 6-28
  Default Acceleration/Deceleration, 6-27
  Default Distance, 6-27
  Default Ramp Time, 6-27
  Default Velocity, 6-27
  Digital to Analog Scale Factor, 6-37
  Disable Mask, 5-12
  ENC Update Time, 6-65
  Encoder 1 Update Time, 6-63
  Encoder 2 Update Time, 6-64
  Error Status "-" Direction Mask, 6-79
  Error Status "+" Direction Mask, 6-78
  Error Status Auxiliary Mask, 6-74
  Error Status Disable Mask, 6-72
  Error Status Freeze Mask, 6-75
  Error Status Invert Mask, 6-68
  Error Status Latch Mask, 6-71
  Error Status Service Request Mask, 6-73
  Error Status Task 1 Mask, 6-76
  Error Status Task 2 Mask, 6-77
  Extension Buffer Partition, 6-80
  External Interrupt Latch Source, 6-33
  External Interrupt Mode, 6-34
  External Interrupt Source, 6-35
  External Interrupt Type, 6-35
  Fault Mask Update Time, 6-66
  Freeze Mask, 5-12
  Home Ending Offset, 6-31
  Home Limit Offset, 6-30
  Home Marker Selection, 6-30
  Home Marker Velocity, 6-31
  Home Switch Directioin, 6-29
  Home Switch Fast Reference, 6-29
  Home Switch Selection, 6-29
  Home Velocity After Power Up, 6-30
  In Position "Wait", 6-28
  Invert Mask, 5-10
  Ki Servo Loop Gain, 6-51
  Kp Servo Loop Gain, 6-51
  Kpos Servo Loop Gain, 6-51
  Latch Mask, 5-11
  Library Ending Offset, 6-67
  Library Memory Limit, 6-68
  Library Program Call Mode, 6-67
  Lower "Modulo" Index Limit, 6-28
  Negative Direction Mask, 5-13
  Port Variable (PV) Wait States, 6-60
  Position Feedback Type, 6-37
  Positive Direction Mask, 5-13
  R/D Resolution Mode, 6-36
  Reset Delay on Motion, 6-80
  Resolver to Digital Update Time, 6-66
  Scaled Trajectory External Port Fetch On/Off, 6-61
  Scaled Trajectory Task 1 Gain Factor, 6-39
  Scaled Trajectory Task 1 Index (PV/BV/LV/FV), 6-39

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Scaled Trajectory Task 1 Ramp Time, 6-39
Scaled Trajectory Task 1 Variable Type, 6-38
Scaled Trajectory Task 2 Gain Factor, 6-40
Scaled Trajectory Task 2 Index (PVBV/LV/FV), 6-40
Scaled Trajectory Task 2 Ramp Time, 6-40
Scaled Trajectory Task 2 Variable Type, 6-40
Servo "In Position" Tolerance, 6-50
Servo "Velocity Error" Integration Control, 6-50
Servo Acceleration/Deceleration Trap, 6-52
Servo Current Limit Trap Level, 6-51
Servo Current Limit Trap Time, 6-50
Servo Feedforward On/Off, 6-51
Servo Loop Update Time, 6-61
Servo Peak Current Limit, 6-51
Servo Position Error Trap, 6-52
SRQ Mask, 5-12
Stepper "Extended" Correction Velocity, 6-49
Stepper "Extended" Slip Tolerance, 6-49
Stepper Correction Velocity, 6-48
Stepper Damping Enable/Disable, 6-49
Stepper Loop Update Time, 6-62
Stepper Power Stage Mode, 6-47
Stepper Resolution, 6-48
Stepper Running Current, 6-47
Stepper Slip Tolerance, 6-48
Stepper Verification Enable/Disable, 6-48
String Buffer Partition, 6-80
Task 1 Background Time, 6-60
Task 1 Exception Processing, 6-31
Task 1 Mask, 5-12
Task 2 Background Time, 6-60
Task 2 Exception Processing, 6-31
Task 2 Mask, 5-12
Upper "Module" Index Limit, 6-28
Velocity Feedback Type, 6-36
PB24 Module Configuring, 3-26
PB24 Modules, 3-26
PC, 2-5
PC command, 5-78
Peak Current Limit, 11-7
Performing Logical Functions, 3-27
Performing Math Functions, 3-27
PGM file template, 5-18
PGM files, 4-9, 5-17
PGM program execution, D-6
PGM100 file, 4-1, 6-2
PGM100 program, 4-2
PID (Proportion, Integral, Derivative), 11-2
PID control algorithm, 6-23
PID Control Loop, 6-36, 6-37
Velocity Error in, 6-50
PID Loop, 11-1
PID Loop Disturbance, 7-23
PID Loop Position Command, 7-23
PID Loop Trap, 5-3
PID Servo Loop, 6-75
Pinouts
R/D interface board, 3-23
Places Right of Decimal Point, Line 1, 6-20
Places Right of Decimal Point, Line 2, 6-20
PM command, 5-79, 5-80
Point-to-point Move, 8-2
Port B Data register, 7-5
Port C Data register, 7-6
Port Variable (PV) Wait States parameter, 6-60
Port Variables, 3-27
Position Capture Program, 8-15
Position Command register, 7-23
Position error
Kpos, 11-2
Position Error, 11-7
Minimizing, 11-7
Position Error register, 7-24
Position Error Trap, 3-5
Position Feedback, 7-10, 10-4, 10-5
Position Feedback register, 7-23
Position Feedback Source, 3-24
Position Feedback Type parameter, 6-37
Position loop, 9-8
Position of ENC option, 7-21
Position of Encoder 1, 7-20
Position of Encoder 2, 7-20
Position Reset register, 7-23
Position Reset Trigger register, 7-23
Position Status, 4-12
Position-on-the-fly, 5-9
Positive Direction Mask parameter, 5-13
Positive Software Limit Threshold, 6-32
Post-compiler Errors, D-3
Power board, 9-3
Power configurations, 3-12
Power connections, 3-3
Power Stage Operating Mode, 6-47
Power Supply, 10-2, 10-3, 10-5, 10-6
Power Up, 3-26
PR Input Port, 7-13
Precautions, 1-6
Pre-compiler Errors, D-2
Prefix 0x, 5-14, 12-2
Primary Encoder, 3-24
primary encoder port, 9-8
Primary Encoder Port, 7-10
Print Formatted Messages, 5-80
Print Message, 5-79
PRM
026, 6-17
PRM100 file, 4-1
PRM307, 3-24
PRM308, 3-24
PRM310, 3-24
Procedures for Corrupted Battery Backed Memory, 3-24
Procedures for Data Output, 3-26

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Program editing
Function keys, 4-8
Program Execution, 4-9
Program Execution Upon Power-up, 6-18
Program Feedhold, 4-10
Program Number, 4-8
Program Size, Limiting, 8-1
Program Title, 4-8
Program Type, 4-8
Program Walking, 4-9
Programmable locations - display, 3-21
Programmed State of Outputs, 3-27
Programming
commands, 5-24
using comments, 8-1
Programming a Counter Chip, 8-14
Programming conventions, 5-2
Programming Example, 8-4
Programming of Encoder Counter, 10-5
Proper Operation, 3-24
Proper Tuning, 3-24
Proportional Gain Adjust for Accel Feedforward, 6-51
Proportional Gain Adjust to Position Error, 6-51
Proportional Gain Adjustment, 6-51
Proportional Joystick, 8-10
PV variables, 7-31
PVI command, 5-81
PWM 4-quadrant Operating Mode, 6-47

Q

Quadrature Register, 7-17, 8-16
Queue Overload, 6-18

R

R/D Board, 10-4
R/D Counter Register, 7-17
R/D interface board pin descriptions, 3-23
R/D interface board pinouts, 3-23
R/D option, 6-66, 7-17
R/D Option Board, 6-36
R/D Position register, 7-20
R/D Resolution Mode parameter, 6-36
R/D System Configuration, 10-4
R/D Update Time, 5-3
R/D Velocity register, 7-21
RAM, 4-3
Ramp Time, 5-90
Ramp Time Setting, 6-27
READ.ME text file, 4-1
Reading Inputs, 3-27
Reading Data From Inputs, 3-27
Reading Inputs, 3-26
Reading Programmed State of Outputs, 3-27
Reading the Thumbwheel #1 Sign, 3-19
Register Load, 7-14
Register Reset, 7-14
Registers, 7-1
Acceleration/Deceleration Feedback, 7-24
Averaged Velocity Command, 7-23
Averaged Velocity Feedback, 7-24
Axis Calibrated Position Command, 7-24
Axis Calibrated Position Feedback, 7-24
Axis Calibration Compensation, 7-24
Axis Status, 7-27
Bit Input, 7-4
Bit Output, 7-4
Compiler Error, 7-10
Compiler Error Line Number, 7-10
Current Command, 7-23
ENC Position, 7-21
ENC Velocity, 7-21
Encoder 1 Position, 7-20
Encoder 1 Velocity, 7-21
Encoder 2 Position, 7-20
Encoder 2 Velocity, 7-21
Error Status, 5-11, 7-26
Error Status "Active Mask", 7-30
Error Status "Overlay", 7-30
Error Status "Reset Mask", 7-30
External Input, 7-8
External Interrupt Position Latch, 7-21
IEEE-488 Address Status, 7-19
IEEE-488 Command Pass Thru, 7-17
IEEE-488 Data In, 7-19
IEEE-488 Data Out, 7-19
IEEE-488 Serial Poll Status, 7-18
IEEE-488 Status Register 1, 7-18
IEEE-488 Status Register 2, 7-18
Instantaneous Velocity Feedback, 7-23
Motion Status, 7-25
Port B Data, 7-5
Port C Data, 7-6
Position Command, 7-23
Position Error, 7-24
Position Feedback, 7-23
Position Reset, 7-23
Position Reset Trigger, 7-23
R/D Counter Register, 7-17
R/D Position, 7-20
R/D Velocity, 7-21
Reset Status, 5-11
RS-232 Queue Out Pointer, 7-9
RS-232-C Queue In Pointer, 7-8
RS-232-C Status, 7-9
Status Active, 5-13
System Library Access Error, 7-10
System Status, 7-28
Task 1 Run Time Error, 7-10
Task 2 Run Time Error, 7-10
Timer count, 7-8
Relative Motion, 8-1, 8-2
Remote LED, 3-5

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Renaming Position Status, 4-12
Renaming Velocity Status, 4-12
Reset Delay on Motion parameter, 6-80
Reset Interrupt Latch, 5-82
Reset Interrupt Operation, 5-9
Reset Latch, 3-26
Reset LED, 3-4
Reset Status register, 5-11
Reset System, 3-26
Resolution, 3-24
Resolution Tracking, 6-36
Resolver, 3-24
Resolver Electrical Cycle, 6-36
Resolver to Digital Update Time parameter, 6-66
Resolver-to-Digital converter, 3-22
Resolver-to-Digital interface card, 1-5
Retrieve the Value of a "Read Only" Register, 12-12
Retrieve the Value of a "Read/Write" Register, 12-14
Retrieve the Value of a Parameter, 12-10
Retrieve the Value of a String Storage Buffer, 12-16
Retrieve the Value of a Variable, 12-15
RI command, 5-9, 5-82
RMS Current Limit parameter, 10-4, 10-5
RMS Current Limit Setting, 3-5
RS-232 Communications, 6-2
RS-232 Communications "loop" Mode, 6-6
RS-232 Communications Mode, 6-14
RS-232 Operating Mode, 5-14, 12-2
RS-232 Queue Out Pointer register, 7-9
RS-232 Status Codes, 5-14, 12-2
RS-232-C, 3-3
RS-232-C Communication, 10-3
RS-232-C Communication parameter, 10-3
RS-232-C Communications Mode, 12-1
RS-232-C interface, 9-15
RS-232-C Jumper Setup, 10-3
RS-232-C Queue In Pointer register, 7-8
RS-232-C standard, 9-14
RS-232-C Status register, 7-9
RS-422-A, 3-3
RS-422-A Communication, 10-3
RS-422-A interface, 9-15
RS-422-A standard, 9-14
RUN command, 5-83
RUN command extended format, 5-83
Run Pre-compiled PGM Program, 5-83
RUN Selection Menu, 4-9
Run Time Error Code - Task 1, 7-10
Run Time Error Code - Task 2, 7-10
Run time errors, 3-4
Run Time Errors, D-1, D-6
RUN( ) command Errors, D-8
Running a "PGM:xx" File, 12-5

S

Safety Procedures, 1-6
Sample Programs, 8-1
Scale Factor of User Units, 6-26
Scale Factor Trajectory Control, 6-26
Scaled Trajectory Algorithm, 6-38
Scaled Trajectory External Port Fetch On/Off parameter, 6-61
Scaled trajectory generation, 6-23
Scaled Trajectory Task 1 Gain Factor parameter, 6-39
Scaled Trajectory Task 1 Index (PV/BV/LV/FV) parameter, 6-39
Scaled Trajectory Task 1 Ramp Time parameter, 6-39
Scaled Trajectory Task 1 Variable Type parameter, 6-38
Scaled Trajectory Task 2 Gain Factor parameter, 6-40
Scaled Trajectory Task 2 Index (PV/BV/LV/FV) parameter, 6-40
Scaled Trajectory Task 2 Ramp Time parameter, 6-40
Scaled Trajectory Task 2 Variable Type parameter, 6-40
Selecting the Home Marker, 6-30
Selecting to Latch a Source, 5-9
Selection of Home Switch, 6-29
Sending an Immediate Command, 12-5
Serial Poll, 5-16, 12-4
Serial Poll Mode, 5-14, 12-2
Serial Poll Mode Status Byte, 7-18
Service Request, 5-14, 5-85, 6-73, 12-2
Service Request Acknowledge, 5-14, 12-1, 12-2
Service Request Character, 5-14, 12-2
Servo "In Position" Tolerance parameter, 6-50
Servo "Velocity Error" Integration Control parameter, 6-50
Servo Acceleration/Deceleration Trap parameter, 6-52
Servo Current Limit Trap Level parameter, 6-51
Servo Current Limit Trap Time parameter, 6-50
Servo Feedforward On/Off parameter, 6-51
Servo loop, 9-8
Servo Loop, 3-6
Servo loop gains, 11-2
Servo Loop performance, 11-7
Servo Loop Performance, 11-7
Servo Loop tracking, 11-7
Servo Loop Update Time parameter, 6-61
Servo Motor Limit Protection, 6-50
Servo Peak Current Limit, 10-4
Servo Peak Current Limit parameter, 6-51
Servo Position Error Trap parameter, 6-52
Servo Problems, 10-5
Servo Update Time, 5-3
Servo Velocity Trap parameter, 6-52
Set the R/D Update Rate at .8 ms, 3-23
Setting In-position Tolerance, 6-28
SETUP, 10-2
Setup Function, 3-24
Setup procedure, 4-26
SETUP.EXE file, 4-2
SETUP.LST file, 4-2

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

SETUP1.EX_file, 4-2
Shunt Active LED, 3-4
Shunt Fuse Failure on Power Board, 10-4
Sign Addresses, 3-19
Sign bit, 7-13
SIGN bit, 7-13
Sign Latching, 3-19
Sign Status Bits, 3-19
Sign Status Storage, 3-19
SIN command, 5-84
SIN Encoder Signal, 3-6
Sine, 5-84
Sine Interpolated Commutation, 6-57
Sine Interpolation Cycle, 6-58
SIN-N Encoder Signal, 3-6
Software
  installation, 4-1
Software Control Status Indicators, 6-68
Software Diskette, 4-1
Software installation, 2-8, 4-3
Software Limit Reference, 6-32
Source of Interrupt, 6-35
Spline generation, 5-23, 5-33
Spline Interpolation Mode, 6-41
Spline modulo mode, 6-44
Spline output mode, 6-45
Spline output PV index, 6-45
Spline point number, 6-41
Spline reference +Limit, 6-44
Spline reference limit, 6-43
Spline reference mode, 6-42
Spline reference scale, 6-43
Spline reference starting point, 6-43
Spline reference time, 6-43
Spline scaling mode, 6-42
SQRT command, 5-86
Square root, 5-86
SRQ Character Modifications, 5-14, 12-2
SRQ command, 5-15, 5-85, 12-3
SRQ Enable/Disable, 6-15
SRQ Mask parameter, 5-12
SRQ Status Code, 5-14, 12-2
Standard RS-232-C interface, 9-15
Startup Problems, 10-2
Startup Software Diskette, 4-1
Static Charge Buildup, Removing, 2-4
Status Active register, 5-13
Status indicators, 3-2
Stepper “Extended” Correction Velocity parameter, 6-49
Stepper “Extended” Slip Tolerance parameter, 6-49
Stepper Correction Velocity parameter, 6-48
Stepper Damping Enable/Disable parameter, 6-49
Stepper Holding Current parameter, 6-47, 10-4
Stepper Loop Update Time, 5-3
Stepper Loop Update Time parameter, 6-62
Stepper Motor Commutation, 6-62
Stepper Motor Problems, 10-3
Stepper Power Stage Mode parameter, 6-47
Stepper Resolution parameter, 6-48
Stepper Running Current parameter, 6-47, 10-4
Stepper Slip Tolerance parameter, 6-48
Stepper Verification Enable/Disable parameter, 6-48
Stepping motor wiring, 9-5, 9-6
String Buffer Partition parameter, 6-80
Strings
  Changing, 4-21
SUB command, 5-87
Subroutine, 5-87
Subtract, 5-88
Summary of Files, 4-2
Supplying Feedback to UNIDEX 100, 3-24
Supplying Position Feedback to U100, 3-24
Supplying Velocity Feedback to U100, 3-24
Suppress LST1 File Generation, 6-17
Switch Closure, 3-19
SYNC command, 5-89
Synchronize, 5-89
Synchronized Trajectory Table Execution, 8-11
System LEDs, 10-6
System Library Access Error register, 7-10
System Operation, 3-24
System Parameters, 6-59
System Registers, 7-25
System Reset, 3-26
System Status register, 7-28
System Tuning, 3-24

T

T command, 5-90
TAN command, 5-91
Tangent, 5-91
Task 0, 5-3
Task 1, 5-3
Task 1 Background Time parameter, 6-60
Task 1 Command Line Dedication, 6-80
Task 1 Exception Operation, 6-76
Task 1 Exception Processing parameter, 6-31
Task 1 Mask parameter, 5-12
Task 1 Parameter Extensions, 6-80
Task 1 Run Time Error register, 7-10
Task 1 String Buffers, 6-80
Task 1/2 Communication Status, 7-28
Task 1/2 Operating Status, 7-28
Task 2, 5-3
Task 2 Background Time parameter, 6-60
Task 2 Exception Operation, 6-77
Task 2 Exception Processing parameter, 6-31
Task 2 Mask parameter, 5-12
Task 2 Run Time Error register, 7-10
Task 2 Trajectory Scaling, 6-40
Task Control Functions, 12-17
Task Interaction, 8-6

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Task Type, 5-3
Task Window Menu, 4-14
Technical Support Questions, 10-1
Template
  PGM file, 5-18
Testing Sign Status Bits, 3-19
THM option, 8-3
Thumbwheel, 1-5
  Description, 3-17
  setup, 3-17
Thumbwheel #1 Execute Switch Closure, 3-19
Thumbwheel Input for Distance, 8-3
Thumbwheel interface, 3-18
Thumbwheel Sign, 3-19
Time Slot Number, 5-3
Time Slots, 5-3
Timer Control/Status, 7-7
Timer Count, 7-8
TITLE command, 5-92
Tolerance for In-position, 6-28
Torque Output to Stepping Motor, 6-47
Tracking Resolution of R/D Option Board, 6-36
Trajectory Control Scale Factor, 6-26
Trajectory generation, 6-22
Trajectory Generator, 6-75
Trajectory Table, 8-11
Trajectory Table Pointer Location, 8-11
Transfer a File from the Host to the U100, 12-8
Transfer a File from the U100 to the Host, 12-8
Transferring Files, 4-22
Transformer, 3-12
Trapezoidal
  Motion, 6-22
Trapezoidal Move, 8-2
Trapezoidal Move with Output-on-the-fly, 8-2
Trigger command, 6-7
Trigger Command Character, 6-14
Tune command, 5-93
Tune corner frequency, 6-53
Tune damping factor, 6-53
Tune minimum frequency, 6-53
Tune time, 6-53
Tuning the System, 3-24
Turning Off PB24 Bits, 3-27

U

U100, 3-2
U100 Auxiliary Encoder Input, 8-9
U100 control board, 9-2
U100 Expansion Bus, 8-3
U100 Fault Condiitions, 4-1
U100 file types, 5-17
U100 Firmware Update, 4-26
U100 Lock Up, 5-14, 12-2
U100 Main Menu Display, 2-9, 4-4
U100 power board, 9-3
U100i system configuration, 2-6
Uncompensated Position Command, 6-56
Uncompensated Position Feedback, 6-56
UNIDEX 100
  enhancing operation with accessories, 1-5
  programming commands, 5-24
UNIDEX 100 Controller
  options, 1-5
UNIDEX 100 Operating Performance, 3-24
Unlatching Inactive State Outputs, 3-26
Unpacking the UNIDEX 100, 2-1
UP/DOWN Count bit, 7-13
Uploading Files, 4-22
Upper "Modulo" Index Limit parameter, 6-28
User Units Scale Factor, 6-28
Using an External Interrupt to Change a Motion Trajectory, 8-12
UT100 Software Installation, 2-9, 4-4

V

Variable Indexing, 7-31
Variables, 7-1, 7-31
  Changing, 4-20
  Float, 7-31
  Integers, 7-31
  Long, 7-31
  String, 7-31
Velocity, 5-94
Velocity command jumper, 3-9
Velocity Error, 11-7
Velocity Feedback, 7-10, 10-4, 10-5
Velocity feedback signals
  PID loop, 11-6
Velocity Feedback Source, 3-24
Velocity Feedback Transducer, 11-7
Velocity Feedback Trap, 3-5
Velocity Feedback Type parameter, 6-36
Velocity Feedforward, 11-7
Velocity for Seeking a Home Limit, 6-30
Velocity loop, 9-8
Velocity Loop, 10-4
Velocity of ENC, 7-21
Velocity of Encoder 1, 7-21
Velocity of Encoder 2, 7-21
Velocity of R/D, 7-21
Velocity Profiling, 8-1, 8-2
Velocity Stability, 11-2
Velocity Status, 4-12
Version Status, 4-11
Version Status Display, 4-13
Vertical Screen Scrolling, 4-6

W

Warnings, 1-6
Warranty Information, B-1

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Warranty Policy, B-1
While, 5-95
WHL command, 5-95
Windows, 2-3
Write Function, 3-26

- command, 5-88
- Software Limit parameter, 6-32
"S" Curve Generation parameter, 6-27
"S" Curve Profiling, 6-27
* command, 5-77
/ command, 5-51
; Character, 8-1
|Velocity Loop, 10-4
+ command, 5-31
+ Software Limit parameter, 6-32
+/- Limit Reset Distance parameter, 6-32
= command, 5-60
0x, 5-14, 12-2
100, 1-2
25-pin "D" Style Connector, 3-6
80386 microprocessor, 4-3
80486 microprocessor, 4-3
Writing Data to Output Latches, 3-26

X

XOR command, 5-96

∇ ∇ ∇

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

READER'S COMMENTS

UNIDEX 100/U100i Motion Controller Operation & Technical Manual
P/N EDU 128, July, 2000

Please answer the questions below and add any suggestions for improving this document. Is the information:

	Yes	No
Adequate to the subject?	____	____
Well organized?	____	____
Clearly presented?	____	____
Well illustrated?	____	____
Would you like to see more illustrations?	____	____
Would you like to see more text?	____	____

How do you use this document in your job? Does it meet your needs?
What improvements, if any, would you like to see? Please be specific or cite examples.

_____

_____

_____

_____

_____

_____

Your name _____
Your title _____
Company name _____
Address _____
_____

Remove this page from the document and fax or mail your comments to the technical writing department of Aerotech.

AEROTECH, INC.
Technical Writing Department
101 Zeta Drive
Pittsburgh, PA. 15238-2897 U.S.A.
Fax number (412) 963-7009

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP