

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.

Visit our website - *Click HERE*

Sifam Instruments Limited
Woodland Road
Torquay
Devon, TQ2 7AY
United Kingdom

Tel: +44 1803 407 700
Fax: +44 1803 407 699
E-Mail: queensgate@sifam.com
Website: www.nanopositioning.com



NPS3110, NPS3220, NPS3330

Operating Manual

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Warranty

Extent

Queensgate Limited ("Queensgate") warrants that the System shall for a period of twelve months from the date of delivery be free from defects in design, workmanship and materials (other than defects attributable to ordinary wear and tear) and, where applicable, shall meet the specifications referred to in the Special Conditions. If the System does not conform to such warranty Queensgate Instruments shall at its option:

replace the System or any part of it found by Queensgate in its sole judgment not to conform to the warranty (all parts replaced by Queensgate Instruments becoming the property of Queensgate Instruments); or
take such steps as Queensgate deems necessary to bring the System into a state where it is free from such defects or meets such specifications, PROVIDED THAT if there is a manufacturer's guarantee in force in respect of the System or any part thereof, the period of twelve months shall be substituted by the period left to expire of such manufacturer's guarantee.

Limitation

Subject as herein provided the aggregate liability of Queensgate in contract, for negligence or otherwise shall in no event exceed the price payable or paid by the BUYER for the System and performance of either one of the options under the above warranty shall constitute an entire discharge of Queensgate's liability under the above warranty.

Conditions

The above warranty is conditional upon:

the BUYER providing Queensgate with adequate written notice of the alleged defect within the above warranty period;
the BUYER affording Queensgate reasonable opportunity to inspect the System on site;
the BUYER using and maintaining the System in accordance with any instructions or recommendations of Queensgate and in particular not subjecting the System to misuse, abuse, neglect, accident, improper alteration or modification or negligence in use, storage, transportation or handling;
as regards defects in design, the design in question not having been made, furnished or supplied by the BUYER

NPS3110, NPS3220, NPS3330

Operating Manual

©2000, Queensgate Ltd.
All rights reserved.
Bracknell, Berkshire, United Kingdom
Third Edition, August 2000.
Document number NPS-3022-M

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Safety Precautions

WARNINGS

HAZARDOUS VOLTAGES

The NPS3110, NPS3220 and NPS3330 Digital Controllers are scientific Instruments, which rely on the provision of a protective earth (ground) conductor to prevent user accessible components developing a hazardous potential in the event of an insulation failure. A protective earth (ground) connection MUST be made to the unit.

DO NOT remove the equipment's cover. There are no user serviceable parts within the equipment and removal of the cover will invalidate the Queensgate Warranty.

Hazardous voltages are present at the unit's front panel mounted NanoMechanism connectors. Unused NanoMechanism connectors should be fitted with the protective covers supplied.

CAUTIONS

ELECTROSTATIC SENSITIVE DEVICES (ESD)

The unit contains components that are susceptible to damage through electrostatic discharge at the NanoMechanism and interface connectors.



Removal of the protective connector covers and connection of cables should be performed in a static safe environment using approved static safety handling procedures (e.g. procedures to BS5783).

Protective covers should be left in place on unused connectors.

ENVIRONMENT

The unit is designed for use in an office or laboratory environment.

Extremes of temperature, humidity, dust or acoustic/mechanical vibration may cause faulty operation or damage to components.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Electromagnetic Compatibility & Low Voltage Directive

CE

The Queensgate® NS2000 modules conforms to the CE Marking Directive 93/68/EEC.

Relevant standards:

1 EMC

Emissions

BS EN 50081-1 1992 EMC Emissions – Residential, commercial and light industrial (class B level).

Immunity

BS EN 50082-1 1992 EMC Immunity – Residential, commercial and light industrial (Class B level). The AX101/AX301 system relies for its operation on the detection of very small signals from its capacitance bridge. As such, exposure to interference fields as defined in BS EN 50082-1 1992 Immunity Standard, Residential, commercial and light industrial may cause spurious voltage fluctuations to the piezo translator causing undesired motion.

2 Safety

BS EN 61010-1, Safety requirements for electrical equipment for measurement control and laboratory use.

Thomas R Hicks

8 May 1997

Dr. Thomas Hicks
Chief Scientist
Queensgate Ltd

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Damage in Transit

The contents of the package should be thoroughly inspected immediately upon receipt.

All material in the container should be checked against the packing list. The manufacturer will not be responsible for shortages against this list unless notified immediately.

If the instrument is damaged in any way, a claim should be made against the carrier. A full report of the damage should be made, including the type and serial number of the instrument, and forwarded to Queensgate Ltd.

Upon receipt of this report, you will be advised of the disposition of the instrument for repair or replacement.

Manual Print History

The print history shown below lists the printing dates of all revisions created for this manual.

Original, NPS-3022-M Issue 1	1997
Revision, NPS-3022-M Issue 2	1999
Revision, NPS-3022-M Issue 3	2000

NanoSensor® and Queensgate® are trademarks of Sifam Instruments Ltd.
All other brand and product names are trademarks or registered trademarks of their respective companies.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Table of Contents

1 General Information 15

1.1 Introduction 15
1.2 Features 15
1.3 Warranty 16
1.4 Safety Symbols and Terms 16
1.5 Specifications 17
1.6 Inspection 17
1.7 Options and Accessories 17

2 Starting Up 19

2.1 Introduction 19
2.2 Front Panel Presentation 19
2.3 Back Panel Presentation 20
2.4 Unpacking and Handling 20
2.5 Mounting the NanoMechanism 21
2.6 Mounting the Controller 22
2.7 Cabling 22

3 NanoControl Panel Software – An Introduction 24

3.1 Introduction 24
3.2 Overview 24
3.3 Installing the NanoControl Panel 25
3.4 Switch On 27
3.5 Switch Off 27
3.6 How do I begin to use my NanoMechanism 28
3.7 The Channel Display 28
3.8 Snapshot Mode 30
3.8.1 To Perform a Snapshot 30
3.9 Monitor Mode 34

4 NanoControl Panel 35

4.1 Introduction 35
4.2 The NanoControl Panel Window 35
4.3 Interface 37
4.4 Channel Parameters Menu 38
4.4.1 PID Parameters 38
4.4.2 Linearisation 39
4.4.3 Calibration 40
4.4.4 Yaw 40
4.4.5 Units 41

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

4.5 Controller Menu 44
4.5.1 Store Channel Parameters 44
4.5.2. Retrieve Channel Parameters 45
4.5.3. Reset Controller 45
4.5.4. Edit Parameters within Controller 45
4.5.5 System and EEPROM Parameters 46
4.5.6 Stage Step 46
4.5.7 Read All 47
4.5.8 Function Generator 47
4.6 Mode 49
4.6.1 Closed Loop 49
4.6.2 Open Loop 49
4.6.3 Invert 49
4.6.4 Freeze 49
4.6.5 Analogue 49
4.7 Configure Menu 50
4.7.1 Store Dynamic Set-up 50
4.7.2 Recall Dynamic Set-up 50
4.7.3 Save System 50
4.7.4 Save Stage 50
4.8 Locking Menu 50
4.8.1 Lock Controller 51
4.8.2 Unlock Controller 51
4.8.3 Set Lock Code 51
4.9 Help Menu 52
4.9.1 Controller ID 52
4.9.2 Stage ID 52
4.9.3 About SDL Queensgate Limited 53
4.10 NanoControl Panel Menu Summary 53

5 Guide to PID Optimisation 57

5.1 Introduction 57
5.2 Altering PID Parameters 58
5.3 PID Parameters 60
5.4 Integrator Time Constant (τ_{int}) 60
5.5 Example Snapshot Responses for Different Values of τ_{int} 60
5.6 Stage Response 61
5.7 Integrator Time Constant Summary 62
5.8 Differential Feedback (τ_{diff} and G_{diff}) 64
5.9 Example Damping of Small Amplitude Resonant Oscillation 64
5.9.1 Differential Feedback Summary 64
5.9.2 Example: Differential Feedback 64
5.9.3 Proportional Feedback (G_{prop} and G_{sp}) 65
5.9.4 Example: Negative Proportional Gain 65
5.9.5 Proportional Feedback Summary 67
5.9.6 Integrator Limit (e_{max} and e_{min}) 67
5.9.7 Integrator Limit Summary 67
5.10 Dynamic Optimisation Summary 68

6 RS232 Parallel Interface 71

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

7 DSP Interface (QI bus) 72

8 NPS-ANA-A 73

8.1 NPS-ANA-A Device Description 73
8.2 Input Connectors 73
8.3 NPS-ANA-A Operation 73
8.3.1 Enabling/Disabling the Analogue Interface using the NPS3000-series Command Language 74
8.3.1.1 *set_mode()* Command 67 decimal 74
8.3.1.2 *read_mode()* Command 66 decimal 74
8.3.2 Enabling/Disabling the Analogue Interface using the NanoControl Panel Software 74
8.3.3 Saving the Configuration to the EEPROM using the NanoControl Panel Software 74
8.3.4 Scanning the System using the NPS-ANA-A 75
8.3.5 Maximum Command Bandwidth 75
8.4 NPS-ANA-A System Calibration 76
8.4.1 Setting the Analogue Input Scale Factor and Offset using the NPS3000-series Command Language 76
8.4.1.1 *read/set_analogue_input_scale_factor()* command code (144 / 145) decimal Format=Floating-point 76
8.4.1.2 *read/set_analogue_input_scale_offset()* command code (146 / 147) decimal Format = Floating-point 76
8.4.2 Setting the Analogue Input Scale Factor and Offset using the NanoControl Panel 77
8.4.2.1 Setting the Analogue Input Scale Factor or Analogue Input Scale Offset 77
8.4.2.2 Storing the Analogue Input Scale Factor or Analogue Input Scale Offset into the System EEPROMs 77

9 NPS-PAR-A Fast Parallel Interface Card

9.1 Description 78
9.2 Input/Output Connectors 78
9.3 Parallel Interface Operation 78
9.3.1 Command String 78
9.3.1.1 Write Cycle 79
9.3.1.2 Read Cycle 80
9.4 Sample Software 82
9.5 PAR-A Interface Performance 86

10 NPS-PAR-B Fast Parallel Interface Card 87

10.1 NPS-PAR-B Device Description 87
10.2 Input/Output Connectors 88
10.3 NPS-PAR-B Operation/Interface Inputs 89
10.3.1 Setting-up the desired Waveform 89
10.3.2 Starting and Stopping the Signal 89
10.3.2.1 Read/set Function Generator Mode 90
10.3.2.2 Differential Line Receiver SCAN Input 90

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

10.3.3 Differential Line Driver SYNC Output 90
10.3.4 Maximum Cable Length 91
10.4 Function Generator Mode 91
10.4.1 Function Generator Command Language Commands 91
10.4.1.1 read/set Function Generator Mode command code 208 / 209 decimal 91
10.4.1.2 read/set Waveform Type 210 / 211 92
10.4.1.3 read/set Waveform Period 212 / 213 93
10.4.1.4 read/set Waveform Scale Factor 214 / 215 93
10.4.1.5 read/set Waveform Scale Offset 216 / 217 93
10.4.1.6 read/set Waveform Data 218 / 219 94
10.4.1.7 store Waveform Data 221 94
10.5 Function Generator Mode Operation 94
10.5.1 Operation Sequence using the SCAN input 94
10.5.2 Operation Sequence using the Set Function Generator Mode Command 95
10.5.3 Changing Waveform Parameters 95
10.6 NPS-PAR-B Parallel Interface Operation 95
10.6.1 Command String 95
10.6.1.1 Write Cycle 96
10.6.1.2 Read Cycle 97
10.7 Sample Software 98
10.8 NPS-PAR-B Interface Performance 102

11 NPS-PAR-C 103

11.1 NPS-PAR-C Device Description 103
11.2 Input/Output Connectors 103
11.3 Operation of the NPS-PAR-C 103
11.3.1 User Interface Connector pin out (NPS3000 series rear panel) 103
11.3.2 Command Word 104
11.3.3 Channel address (A0, A1) decode 104
11.3.4 Command Write Cycle 105
11.3.5 Command Sequence for writing to all 3 Channels (CH1, CH2, CH3) 106
11.3.6 Command Sequence for writing to less than 3 Channels 106
11.3.7 Status signals from NPS-PAR-C 106

12 NPS-SER-A 107

12.1 NPS-SER-A Device Description 107
12.2 Input/Output Connectors 107
12.3 Operation of the NPS-SER-A 107
12.3.1 Interface Connector pin out (NPS3000 series rear panel) 107
12.3.2 NPS-SER-A Timing Diagram 108
12.4 NPS-SER-A Specification 109

13. Custom Interfaces 110

14. Command Language 111

14.1 Introduction 111

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

14.2 Overview 111
14.3 Controller Commands 111
14.3.1 Command Sequence 111
14.3.2 RS232 Serial Interface 111
14.3.3 DSP Port Serial Interface 112
14.3.4 PAR Parallel Interface 112
14.3.5 Custom User Interface 112
14.3.6 Command Word 112
14.3.7 Command Code 113
14.3.8 Channel Specifier 113
14.3.9 Controller Specifier 114
14.3.10 Checksum 114
14.3.11 Data Word 114
14.3.12 Terminator 115
14.4 Controller Responses 115
14.5 Error Handling 115

15 Dynamic Link Library (DLL) 124

15.1 Introduction 124
15.2 Overview 124
15.3 Functions contained within NGCMOD32.h 124
15.4 NGCMOD32.h Code Print out 124

LabVIEW® Introduction 127

16 NanoScan 127

16.1 Starting NanoScan 127
16.1.1 Connecting Controller to PC 127
16.1.2 Starting the Programme 127
16.2 Configuring a NanoScan 128
16.2.1 Description 128
16.2.2 Positive Fly back Example 130
16.2.3 Negative Raster Example 132
16.2.4 Custom Example 135
16.3 Running a Scan 139
16.4 Changing Default Settings 142
16.4.1 Scan Data File Path 142
16.4.2 Graph Settings 143
16.4.3 Hardware 144
16.4.4 INI Path 145
16.4.5 Chan Set-up 146
16.5 Quitting Programme 146

17 Hardware Drivers 147

17.1 Hardware Driver Details 147

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

17.1.1 NPS3XXX Interface 147
17.1.2 NPS3XXX Calibration 148
17.1.3 NPS3XXX Controller ID 150
17.1.4 NPS3XXX Controller Snapshot.vi 152
17.1.5 NPS3XXX Controller Status 153
17.1.6 NPS3XXX Dynamic Set-up 154
17.1.7 NPS3XXX Stage Control 156
17.1.8 NPS3XXX Stage ID 157
17.1.9 NPS3XXX Controller Function Generator 158
17.1.10 NPS3XXX Error 159
17.2 Usage Examples 160

18 Software Examples 161

18.1 Introduction 161
18.2 Example 1: Using the command dll NGCMOD32.dll with MS Windows® SDK 161
18.3 Example 2: Using the command dll NGCMOD32.dll with Visual Basic C++ 4.0 and MFC 162
18.4 Example 3: Using the OLE control NGC_OCX.ocx. with Visual Basic 4.0 164

19 Application Note 1: Setting Up Custom NanoMechanism 166

19.1 Introduction 166
19.2 Procedure 166

20 Application Note 2: Measurement of Resonant Frequency 170

20.1 Introduction 170
20.2 Calculation of RF and BW 170
20.3 Measurement 170
20.4 Example 171
20.5 Conclusion 173

21 Troubleshooting 174

21.1 Introduction 174

22 Maintenance and Configuration 176

22.1 Introduction 176
22.2 Routine Maintenance 176
22.3 Configuration 176
22.3.1 Controllers 176

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

22.3.2 NanoMechanism 176

23 Queensgate After-Sales Office 178

Appendix A: Specifications 179

Appendix B: Glossary 180

Appendix C: Customer Return Report: QCD6115f 182

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

1 General Information

1.1 Introduction

This chapter sums up the essential information one needs to be familiar with before using a system composed of an Queensgate Digital Controller and a NanoMechanism.

It looks at:

- Features
- Warranty
- Safety
- Specifications
- Inspection
- Options & Accessories

1.2 Features

The NPS3110, NPS3220 and NPS3330 Digital Controllers use advanced Digital Signal Processor (DSP) based techniques with 21-bit resolution to provide positioning noise to better than 1 part in $2x10^6$ (equivalent to 0.05 nm in 100µm).

The Controller includes EEPROM storage of Electronic Data Sheet configuration data and built-in algorithms to minimize settling times and the affects of mechanical resonance.

The Controller is supplied with an RS232C serial PC interface and a DSP Port serial interface as standard. Additional alternative interfaces can be supplied to order

Control, monitoring and re-configuration of the NanoPositioning system is performed remotely by using a PC and one of the following:

- The NanoControl Panel software supplied.
- Your own application software.
- Calls to the Dynamic Link Library.
- LabVIEW®.

The only user control fitted to the Controller is its mains On/Off switch.

The Digital Controller features are:

- Digital signal processing.
- 21-bit resolution providing resolution of 0.05 nm in 100µm.
- Proportional, Integral and Differential (PID) feedback.
- Better than 0.02% linearity error.
- Abbe error correction.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

- Closed or open loop operation.
- Software re-configurable loop parameters for system optimisation.
- EEPROM storage of:
    - Controller electronic data sheet.
    - NanoMechanism stage electronic data sheet.
    - 5-user defined dynamic set-ups.
- Choice of computer interface:
- Programmable RS232C serial interface is fitted as standard.
- DSP Port is fitted as standard.
- Alternative Analogue, High Speed Parallel and Custom interfaces are available as options.
- Up to 6 NanoMechanism channels can be controlled from one computer by means of a Master/Slave connection of 2 Controllers.
- Free standing unit with universal input power supply (90-260V rms, 45-60 Hz).
- NanoControl Panel software providing Windows control, monitoring and re-configuration facilities.
- Dynamic Link Libraries, supporting the entire NPS3000 command language, for simple integration of control, monitoring and re-configuration of the system into application software.
- LabVIEW® drivers enabling control to be exercised in LabVIEW®'s 'G' graphical programming environment.
- Visual Basic 4 OCX for access to commands in Microsoft Visual Basic V4.

1.3 Warranty Information

Warranty information is located at the front of this User's Manual. Should your Digital Controller require warranty service, contact Queensgate Ltd (see Section 23) directly or one of its representative in your area for further information. When returning the Digital Controller for repair, be sure to fill in and include the service form (Appendix C) with an RA number obtained from Queensgate's Sales Administrator to provide the repair department with the necessary information.

1.4 Safety Symbols and Terms

The  symbol on the Digital Controller highlights the fact that the Digital Controller is an electro-static sensitive device and should be handled appropriately.

The **WARNING** heading used in this User's Manual explains dangers that might result in personal injury of death.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

1.5 Specifications

Full Digital Controller specifications are included in Appendix A.

1.6 Inspection

Your Digital Controller was methodically inspected electrically and mechanically before shipment. After unpacking all items from Queensgate specially designed shipping carton, check for obvious signs of physical damage that may have occurred during transit.

Report any damage to the shipping agent immediately.

Save Queensgate original packing carton and foam for possible reshipment.

The following items are included with every Digital Controller:

- A Digital Controller NPS3110 or NPS3220 or NPS3330;
- A mains power cable;
- RS232 Interface Cable;
- A CD-ROM, ref. NGC-3037-S containing sample software and .pdf formats of this Operating Manual and the Command Language Reference Manual;
- Possibly a hard copy of this Operating Manual, ref. NPS-3022-M Issue 3 and one of the Command Language Reference Manual, ref. NPS-3023-M, Issue 2.

If you require an additional Operating Manual, order it by contacting Queensgate (see Section 23).

1.7 Options and Accessories

The following options are available to order:

- Low Noise option: noise levels 3 times lower than standard. (The maximum NanoMechanism cable length for this option is 2m.)
- Low Drift option having drift levels of 70ppm • K^{-1} (3 times lower than standard).
- 1, 2 or 3 Channel options (4, 5 or 6 using 2 Controllers):
  - o NPS3110 1 Channel option.
  - o NPS3220 2 Channel option.
  - o NPS3330 3 Channel option.
- NanoMechanism channel extension cables of 3m and 5m (these are not compatible with the Low Noise option).

Note: The maximum cable length (NanoMechanism cable plus extension cable) must not exceed 6m.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

When driving cables over 2m in length the noise increases by approximately 20% per meter of cable.

- Alternative Controller interface options are available to order:
  - **Range of high-speed parallel interfaces**. These allow command rates of up to 4000 commands per second. These may be used in conjunction with the RS232C interface.
  - **Analogue interface (NPS-ANA-A)**. This provides an analogue command input for each NanoMechanism channel. This may be used in conjunction with the RS232C interface. The RS232C interface can be used to modify the NanoMechanism sensitivity to analogue commands.
  - **Custom interface**. Custom interfaces can be designed to your requirements. These can allow command rates of up to 4000 commands per second. Please contact Queensgate Instruments for further information, see page 4 for details.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

2 Starting Up

2.1 Introduction

This Section will take the user through each starting step to get familiar with the System. It is highly recommended to read this Section before going any further by directly plugging the system.

- Presenting the Front Panel;
- Presenting the Back Panel;
- Unpacking and Handling;
- Mounting the NanoMechanism;
- Mounting the Controller;
- Cabling.

2.2 Front Panel Presentation

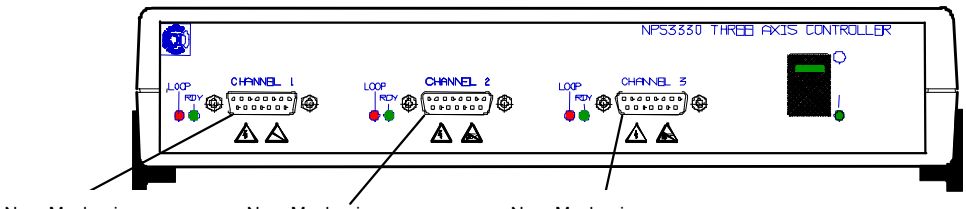
This is fitted as standard. The Controller may be fitted with 1, 2 or 3 NanoMechanism Channels (NPS3110, NPS3220 and NPS3330 respectively).

The NanoMechanism Channel provides closed-loop or open-loop control of Queensgate's range of:

- NanoMechanisms or
- NanoSensors® and/or Piezo Drivers (via adapter cables).

Connection is via the 15-way D-Type connectors on the Controller's front panel figure 2.1.

Figure 2.1: NanoMechanism Connectors



NanoMechanism Channel 1 15 way D-type (NPS3110, NPS3220 and NPS 3330)

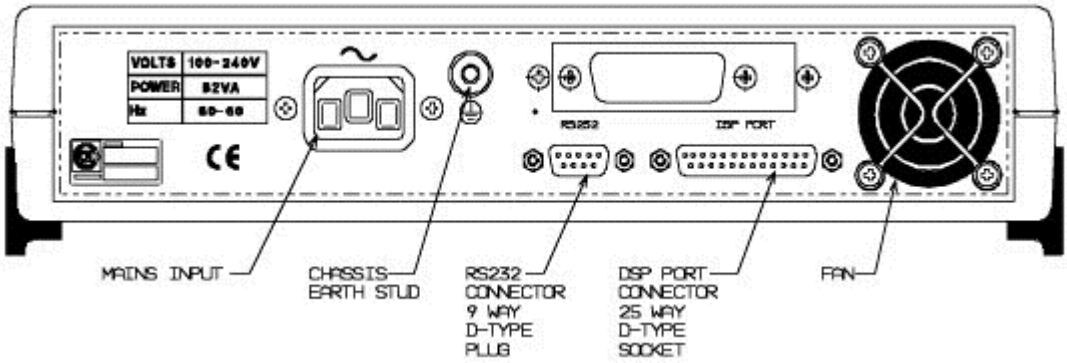
NanoMechanism Channel 2 15 way D-type (NPS3220 and NPS 3330)

NanoMechanism Channel 3 15 way D-type (NPS 3330)

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

2.3 Back Panel Presentation

The back panel is standard for the three Digital Controller models.



2.4 Unpacking and Handling

Ensure you are working in a static safe environment and are using static handling procedures.

The units are robust and require no special handling precautions other than those normally used for electronic equipment.



Carefully remove the products' transit packaging, DO NOT remove the connector protective covers until connections are to be made. Retain the transit packaging for possible future use.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Visually inspect the system for transit damage and check for missing items. If any damage or missing items are apparent do not attempt to install the system:



- If the system is damaged in any way, report the damage to Queensgate and return the product (see Section 23 for contact details).
- If there are missing items, report them immediately to Queensgate (see Section 23 for contact details).

2.5 Mounting the NanoMechanism

NanoMechanisms are NanoPositioning devices with nanometre resolution and stability. They require care in use and MUST NOT be subjected to large torques.

You should refer to the NanoMechanism User Guide and the NanoPositioning Book for detailed installation information for the range of NanoMechanisms.

Briefly, the following points should be noted:

- The mounting surfaces for the NanoMechanisms should be ground flat (N5 or better).
- NanoMechanisms must be installed using only the specified threaded mounting points provided. Spring washers are to be used. The recommended tightening torque is 0.5 Nm (for M3 threads).
- NanoMechanisms should not be over constrained. The NanoMechanism should be preloaded into the system to avoid over constraint.
- NanoMechanisms are internally preloaded (see the NanoMechanism Specification Sheet) and can exert a pulling force up to the preload value.
- Ensure the routing of the NanoMechanism cable is such that:
  - The cable does not exert a force on the NanoMechanism.
  - The route does not pose a safety hazard.
  - The cable is not twisted or kinked and is not unduly stressed.
  - There is no chaffing or risk of future mechanical damage.

For further assistance on installing NanoMechanisms please contact Queensgate Instruments, see page 4 for details.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

2.6 Mounting the Controller



Do not remove the connector protective covers until ready to connect the cables.

The Digital Controller is free standing and may be located on any stable flat surface (e.g. a workstation or a desk) providing protection against accidental damage.

Do not obstruct the Digital Controller's cooling fan airflow of side air vents.

2.7 Cabling



Ensure the Controller is switched off before installing or removing cables.

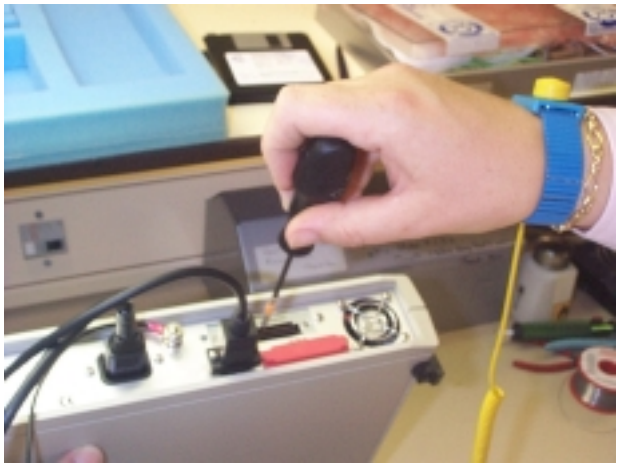
When routing cables ensure:

- The route does not pose a safety hazard.
- The cable is not twisted or kinked and is not unduly stressed.
- There is no chaffing or risk of future mechanical damage.
- The cable is properly strain relieved.

Queensgate recommends the cables are connected in the order listed below and the connectors secured using their locking screws to prevent accidental disconnection:

1. Connect the NanoMechanism(s) to the appropriate channels on the Controller. Please refer to Chapter Four for details of mounting NanoMechanism. Remove the Controller front panel D-type Channel 1 connector protective cover and connect the required NanoMechanism cable connector. Repeat, as necessary, for the remaining channels. Leave the connector protective covers in place on unused channels.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com



2. Connect the supplied Null Modem cable between the computer's 9-pin D-type COM port and the Controller's rear panel RS232C computer interface connector. Secure the connectors in place with the locking screws. If your computer has a 25-way serial port connector, use the Null Modem cable with a 9-way to 25-way adapter.

Connect the ac mains supply cable to the Controller rear panel mains input connector



For further information on RS232C Interface, see Section 6. If your system is fitted with additional user interfaces, refer to Sections 8 onwards.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

3. NanoControl Panel Software

3.1 Introduction

This Chapter presents the accompanying software and its set-up. The overview gives a summary of the possibilities of the software. Then there is a series of points explaining in detail the installation of the software:

- Installing NanoControl Panel Software
- Switch On
- Switch Off
- How do I begin to use my NanoMechanism
- Confidence Check
- Monitor
- Snapshot

3.2 Overview

This software has been specifically written to allow the user a simple interface to Queensgate Digital Controller and NanoMechanism. From this software the user can:

- Manually command positions to the installed channel or channels. This may be used in non-automated applications and for confidence check of the operation of the NanoMechanism on unpacking and testing.
- Read-back the NanoSensor® output from the NanoMechanism. This will give some indication of the position and position noise of the NanoMechanism. (Note that this reading does not give the true position noise of the system.)
- Selected pre-set PID electronic values for fast, medium and slow speed settings. These settings adjust the closed-loop electronics to correspond to the controller closed-loop motion of different applied masses.
- Manually adjust PID electronic values. This feature allows the user to adjust PID parameters for specific configurations and performance optimisation. Please refer to Chapter Five for a complete description of this facility.
- Apply and monitor step-response stimuli (Snapshot mode). This feature allows the user to check the step response and settle times for the mechanism *in situ*, with applied load. This feature is particularly powerful when used in conjunction with the PID adjustment. Simple, iterative adjustment allows optimisation of dynamic response.
- Monitor stage position as a function of time.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

3.3 Installing the NanoControl Panel, Sample Software and Documentation

To install the NanoControl Panel and DLL software:

1. Switch on the PC and start Windows 95, 98 (or Windows NT).
2. Insert CD ROM in the D:\ disk drive.
3. Click Start
4. Click Run.
5. At the Run command line Type a:\setup [Return].
6. Installation commences. Follow the series of prompts that guide you through the process until set-up is complete. This takes around three minutes, depending on the speed of your computer.
7. Click Finish.
8. Remove the disks and start the Queensgate® NanoControl Panel program.

Note: you can abort the installation process at any time by using the Cancel button.

The software screens involved in this procedure are shown graphically in Figure 3.1

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Figure 3.1: Software Dialogue Screens seen during NanoControl Panel Software Installation

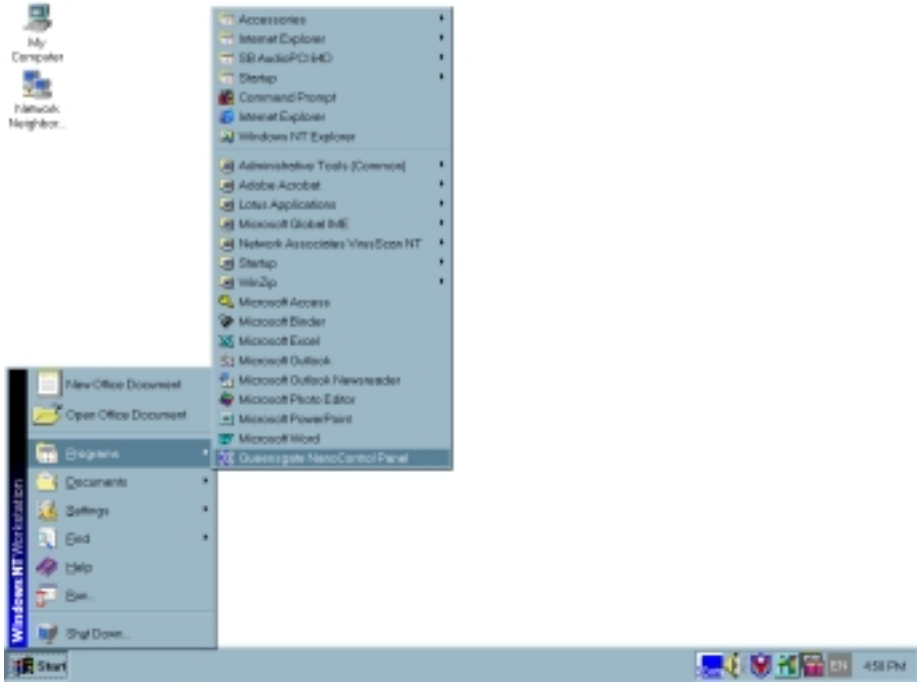


3.4 Switch On

Check that the system has been correctly installed as per the instructions in the preceding Chapters. Follow this procedure for switching on the system.

1. Switch On the ac mains supply.
2. Set the Controller On/Off switch to On (|). The green Supply On LED lights come on.
3. At the system computer:
   - o Switch On.
   - o Start Windows 95, 98 (or Windows NT).
   - o Start the Queensgate NanoControl Panel program.

Figure 3.2:



4. The Queensgate NanoControl Panel is displayed on the computer screen.

3.5 Switch Off

1. The software may be left on during switch off, but it is good practice to shut down NanoControl Panel first.
2. The order of the switch off procedure should be followed closely to prevent any damage to your NanoMechanism. Set the Controller On/Off switch to Off (0). The green Supply On LED goes out.
3. Switch Off the AC mains supply or disconnect the mains cord.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

3.6 How do I begin to use my NanoMechanism

Operation of your NanoMechanism is very simple via the NanoControl Panel software. When your Queensgate NanoPositioning system is delivered to you, all the following are ready to use:

- The system is linearised. The commanded position equals the measured position to within the specification for your NanoMechanism given in your test report for the calibration data. The linearisation coefficients are pre-programmed into the controller, this feature is invisible to you.
- The system is calibrated. The calibration factor(s) are pre-programmed into the controller, this feature is invisible to you. You simply command the position to the stage. The default units of measurement are microns (µm, unless expressly documented) you can change the units via the «Channel Parameters» and then the «Units» menu.
- The factory default (unless expressly documented) is that each axis of your NanoMechanism will be working CLOSED LOOP at switch-on. The 'LOOP' LED beside the NanoMechanism connector (both on the controller and on the software screen will be coloured green. This means the servo-loop is in operation. If OPEN LOOP is selected for a channel then the LED for that channel will become red.
- Queensgate has set up your mechanism is as follows: The travel range of you NanoMechanism is calibrated to move to ±50% of a zero position. The factory default position (unless expressly documented) is 0.000µm. Hence with a 15µm travel range mechanism, you will find you can command any position from -7.500µm to +7.500µm. Check you test reports, you may well have much more range available than we specify in our datasheets!
- The factory default interface (unless expressly documented) is the serial RS232. Queensgate has already configured the PID loop settings for you. The PID settings refer to the different terms within the closed-loop electronics used by Queensgate to control the dynamic performance of the NanoMechanism. From the «configure» menu, you can select «Recall Dynamic Set-up». Three PID system configurations have been pre-configured for your use: fast, medium and slow. Fast means the fastest the stage can stably move with a light load defined in the test report. Medium is an intermediate setting. Slow means the speed at which the servo loop is stable for all masses up to the maximum allowed mass, an equivalently low noise setting.
- In most applications, you will find a pre-configured PID setting, which closely corresponds to your applied load. The factory default PID setting (unless expressly documented) is FAST. In some instances, you will want to manually adjust the PID settings to optimise stepping and scanning response of your NanoMechanism. Please refer to Section 5 for a description of PID adjustment.
- Before beginning to use your NanoMechanism, it is always useful to perform some simple confidence checks.

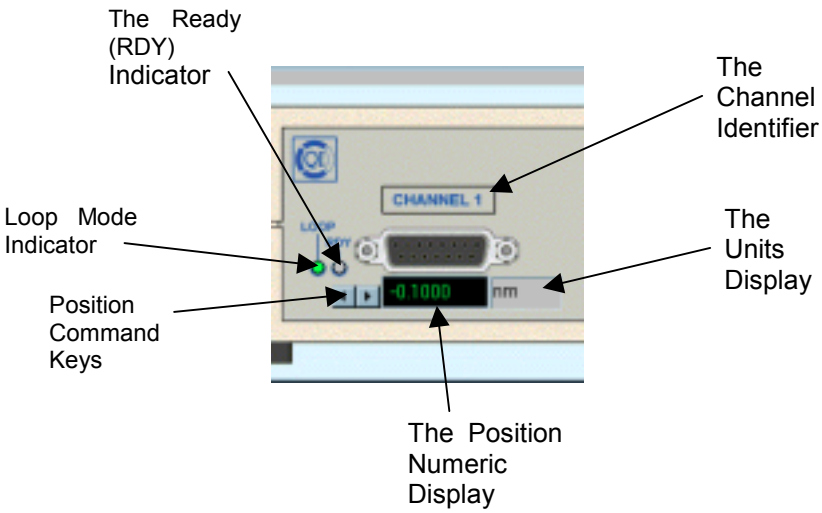
3.7 The Channel Display

The Queensgate NanoControl Panel software allows you to command NanoMechanism position changes and monitor the resultant movement in near real time on the Graphic Display. This can be used as a system confidence check as follows:

The Channel Display has the following features.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Figure 3.3: Channel Display



- The Channel Identifier. Although the positions for all 3 channels are updated continuously, this indicates which channel is selected. Only the selected channel can be adjusted. Channel can be selected by two methods:
  - A pull down menu in the tool bar (directly below the <<controller>> menu).
  - Channel selector above the PID sliders.
- The LOOP Mode Indicator: This is always (green = closed loop and red = open loop). All the Queensgate NanoMechanisms (unless expressly documented), will be configured for closed-loop operation. The factory default setting should be closed-loop hence the LED should be coloured green. If the LED is coloured red, use the <<mode>> menu and check the <<closed-loop>> option. The LED should turn from red to green.

  Note: When in closed-loop and the position is commanded to 0.000µm, this is what your read-out will display 0.000µm. If you are open loop mode, drift and creep will mean that the read-out position will not correspond to the commanded position.
- The Units Display: This indicates the selected operating units for the Position Numeric Display (Section 4.4.5). If you wish to change the measurement units used by the Controller, you may do so from the option buttons on the units dialog box (accessed from the «Channel Parameters» menu):

  Micrometers: µm. The Units display shows um and the position readout will be in micrometers as shown in the left-hand picture below:
  Nanometres: nm. The Units Display shows nm and the position readout will be in nanometres as shown in the right-hand picture below.

Figure 3.4: Unit Displays





- The Position Numeric Display. Measurement noise ($\delta x_{m\cdot n}$) may cause the display to vary a little. This measurement noise will be due to vibration of the stage. The high speed of the closed loop means that the mechanism will vary by around ±0.010µm whilst sat on a (non-isolated) bench. To command a zero position, type 0.000 in the command box and check that you see 0.000 ±0.010µm in the position numeric display.
- The Ready (RDY) Indicator. This is lit on the Controller front panel and is **flashing** in the display on NanoControl Panel (closed loop mode only). The ready limit is set in the «channel parameters» > «units» menu. The ready limit is lit when the measured position is within ± (ready limit) of the commanded position. Queensgate pre-set the ready limit based on the particular NanoMechanism and the noise levels seen in the system.
- Commanding a position:
  - Position Command Keys: you can sequentially step the NanoMechanism using the ◂ and ▸ keys beneath the position numeric display. It will move the NanoMechanism in steps of 1nm or 1um depending on the units selected.
  - Selecting the Position Numeric Display: Click on the display and enter the required position in the units selected. (I.e. it is desired to command the NanoMechanism to a position of 4.4µm. If units are in micrometers enter 4.4. If they are nanometres then enter 4400).
  - Position command box: This is directly below the Mode Menu in NanoControl panel. Enter the required position.

3.8 Snapshot Mode

Snapshot mode allows you to command a channel to a particular position (i.e. a step of defined magnitude and duration). It provides a graphic display of a defined time slice of that (or another) channel's response. Hence you can measure the dynamic performance of the NanoMechanism in real time. As a confidence check, you can re-create the Queensgate tests for Fast. Medium and Slow PID loop settings.

This is especially useful when you are optimising your system and wish to observe the effect that changes to a channel's PID parameters has on that or another channel, please refer to Chapter Five for a detailed description.

As an aid to analysis, two cursors are available on the Graphic Display to mark positions of interest on the graph.

Snapshot data may be saved to disk and retrieved from your computer for analysis in other programs (e.g. spectral analysis). You may also (by agreement) return the snapshot data file to Queensgate for interpretation and advice.

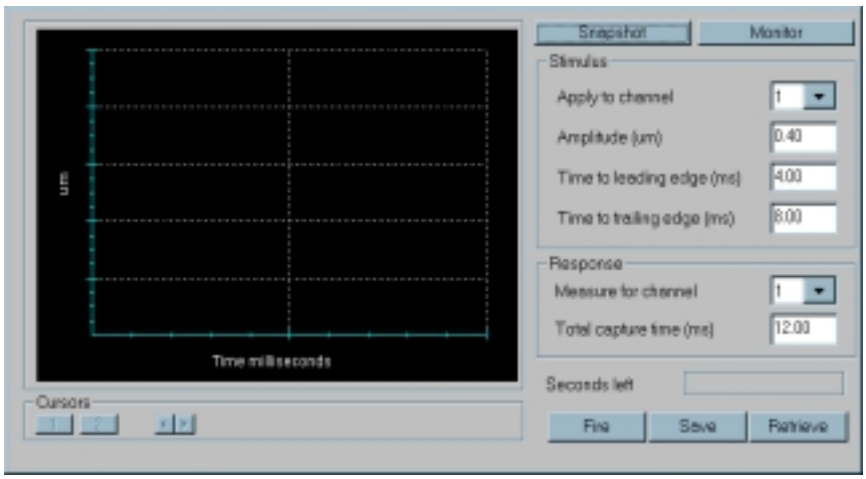
3.8.1 To Perform a Snapshot

1. Step the NanoMechanism (on the channel of interest) to the numeric position required prior to the snapshot (typically 0.000µm).

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

2. Click on the <<Snapshot>> button. Part of the screen looks like this:

Figure 3.5: Screen



3. Select the stimulus parameters:

   The parameters will have the default values in as shown above. These of course can be changed. Brief explanation of them and some values to enter to get you going with a step response;

   - «Apply to channel»

   The channel to be stepped 1, 2 or 3.

   - «Amplitude»

   The step 'Amplitude in the current units selected (Section 4.4.5). Try 0.5µm as the amplitude.

   - «Time to leading edge»

   As required in milliseconds. Try 0.04ms. *This is the time delay between the start of monitoring and the leading edge of the snapshot step pulse.*

   - «Time to trailing edge»

   As required in milliseconds. Try 10ms. *This is the duration of the snapshot step pulse.*

4. Select the Response parameters:

   - «Measure for channel»

   The channel to be read as required (1, 2 or 3).

   - «Total capture time»

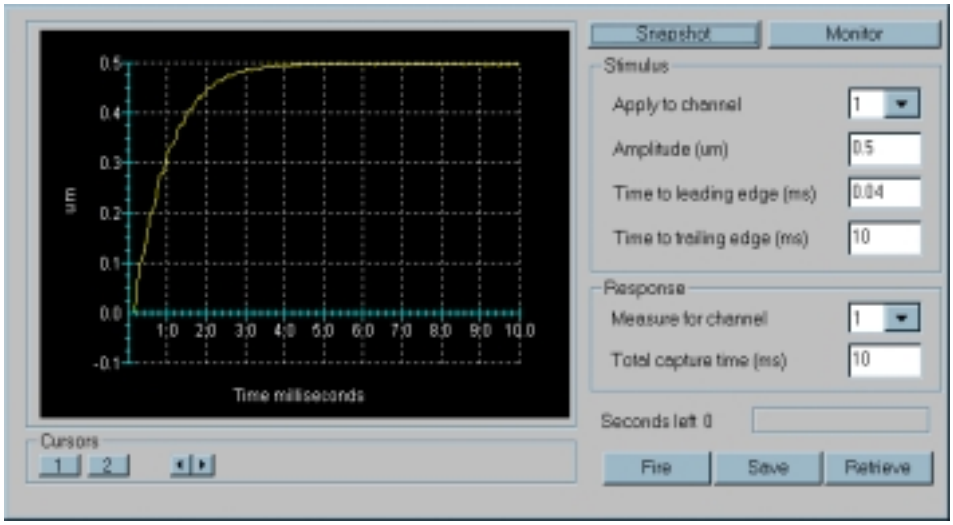
   As required in milliseconds. Try 10ms. *To ensure the snapshot time slice captures the entire action make Total Capture Time = (Time to leading edge) + (Time to trailing edge)+ (Estimated post step settling time).*

5. Click on the «Fire» button and wait for the display to appear.

   - The 'Seconds left' should count down and then a snapshot should appear as shown in the example below.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Figure 3.6: Example of Snapshot

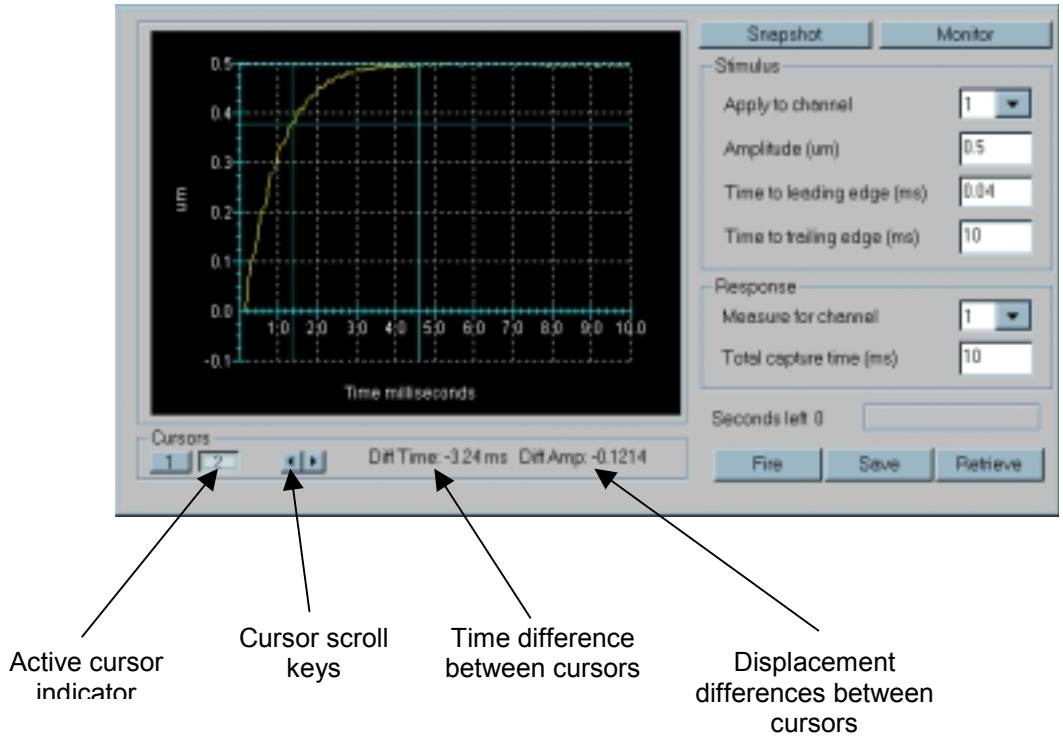


6. To aid PID analysis, cursors are available to give time and displacement information.

- Click Cursor «On».
- Click «1» On and use the Cursor buttons (◂ ▸) to move Cursor 1 to the desired position.
- Click «2» On and use the Cursor buttons (◂ ▸) to move Cursor 2 to the desired position.

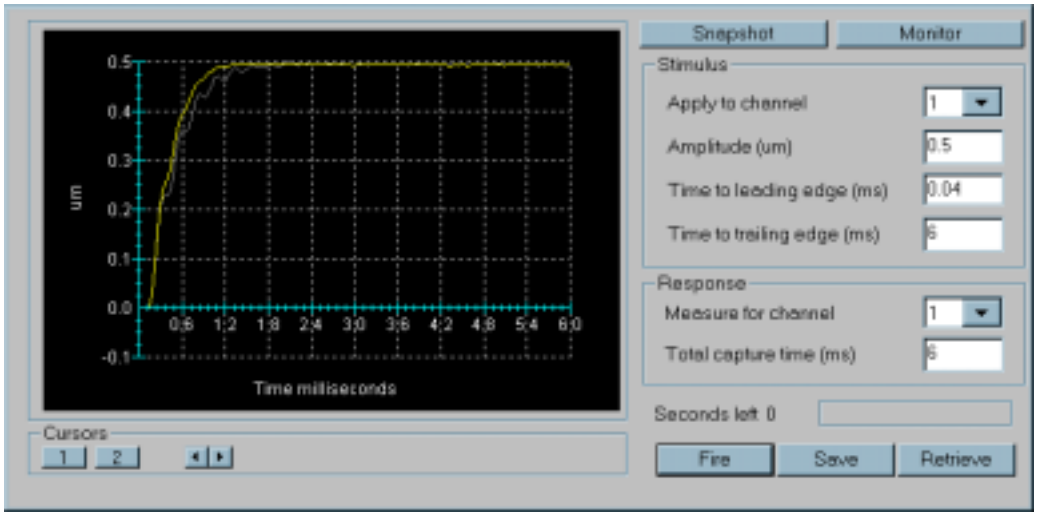
Below is the snapshot with the cursors.

Figure 3.7: Snapshot showing Cursor



7. As a further aid to PID analysis, there is a previous snapshot ghost effect. This enables the user to directly compare a snapshot with the previous one.

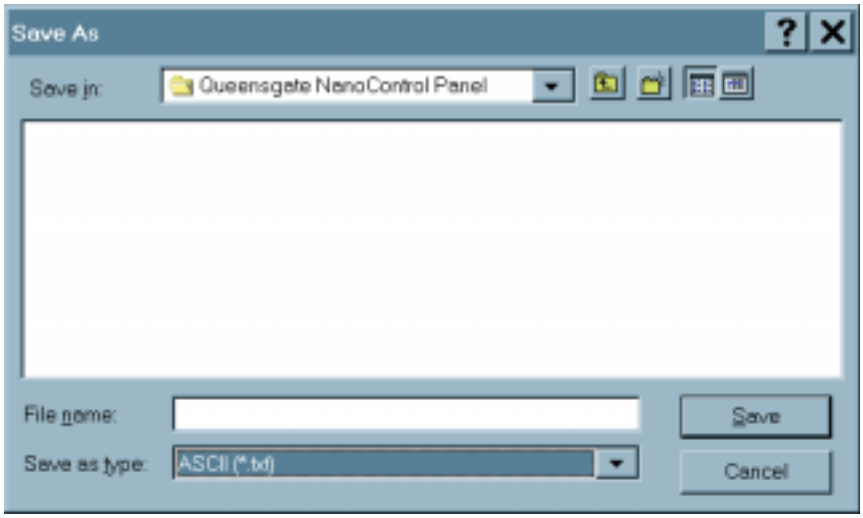
Figure 3.8: Second Snapshot highlighting the Ghost Effect



8. If you wish to save the snapshot data, click the «Save» button to get the Snapshot Save dialog box.

Note: The user can select the folder into which to save the data file. The default is the NanoControl Panel folder. Data can be saved in *.csv (comma separated variable), *.dat (Queensgate format) or *.txt (ASCII text) formats. Remember to type the full filename and the extension, i.e. testdata001.txt

Figure 3.9: Save As Box



9. If you wish to retrieve previously saved snapshot data, click the Retrieve button to get the Snapshot Retrieve dialog box.

10. It is possible to perform cross talk, or cross coupling, measurements on multiple axes systems. Simply select different channel numbers for the «apply to channel» and «measure for channel» boxes.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

3.9 Monitor Mode

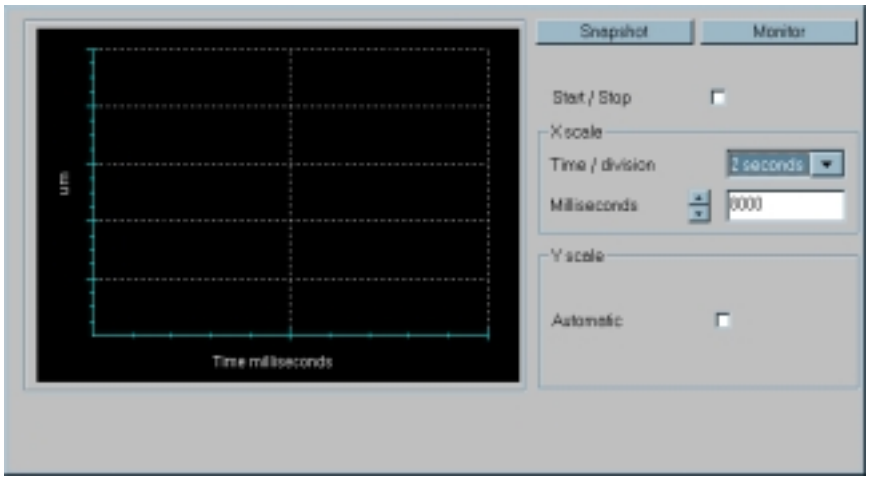
Use the monitor mode to see the movement and the stability of the NanoMechanism as a function of time.

- Click on the «Monitor» button.
- Set the monitoring parameters:

  «Time/division» 2 seconds

  «Y Scale Automatic»

Figure 3.10:



- Click on Monitor «Start/Stop», a tick appears in the check box and the graphical display begins. You will see a scrolling graphical display of the measured position. Again you will see some noise.
- Double click on Channel 1 position numeric display, Type '0.1' and press ENTER. Observe the response on the Graphic display.
- Change the Time / division to 1 second. The time base should now change.
- Click on Monitor «Start/Stop» to stop it.

Note:

- Use «Time/division» to change the X-axis scale on the graphical display.
- Use «Automatic» to allow the software to chose the Y scale on the graphic display.

4 The NanoControl Panel – An Introduction

4.1 Introduction

This Section presents the NanoControl Panel software. It will give a fundamental overview of the use of this software.

4.2 The NanoControl Panel Window

The NanoControl Panel enables the user to command position changes, optimise PID parameters, and monitor system performance via an easy to use MS Windows® graphical user interface. The window will differ if the Digital Controller is 1-, 2- or 3-channel by displaying the appropriate Front Panel.

Figure 4.1: Screen for a 1-Channel Digital Controller (NPS3110)

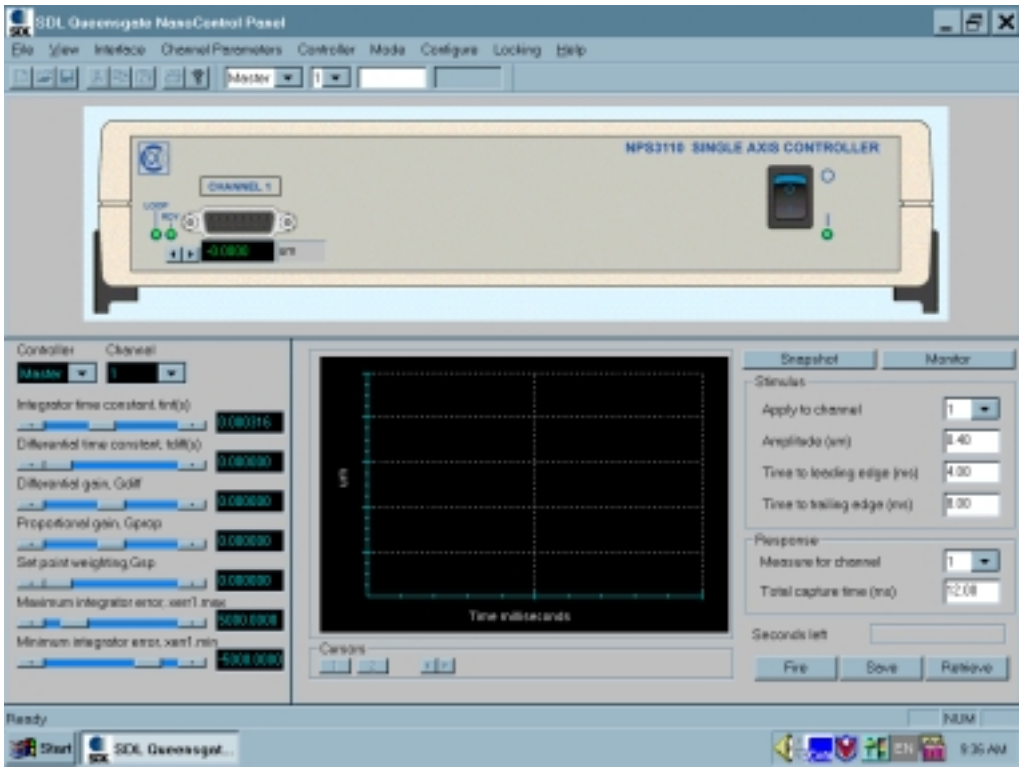


Figure 4.2: Screen for a 3-Channel Digital Controller (NPS3330)



The NanoControl Panel consists of:

1. **Title bar**: standard MS Windows® title bar.
2. **Menu bar**: accesses the pull-down menus for all commands. The pull-down menus are detailed in Table 4.1. Similar to other MS Windows applications, some commands may not be activated and are therefore greyed out, and many commands have shortcuts displayed directly on the screen.
3. **Tool bar**: standard MS Windows® tool bar (may be hidden using Menu bar View option).
4. **Virtual Control Panel**: For each NanoMechanism channel this menu provides:
   - Facility to step the NanoMechanism to a new position.
   - NanoMechanism sensor position numeric display.
   - Ready limit indicator.
   - Indication of the channel's operating mode, Closed Loop (LED light is coloured green) or Open Loop (LED light is coloured red).
   - Indication of which is the active channel (i.e. the channel's PID parameters are selected for modification). A box drawn around the Channel label indicates the active channel.
5. **Channel Selector**: selects the active channel.
6. **PID Sliders**: these enable modification of the active channel's PID parameters using the sliders, slider buttons or numeric display.
7. **Graphical Display**: provides a graphical representation of the motion of a selected channel's NanoMechanism. The display mode is selected at the Snapshot/Monitor Controls:

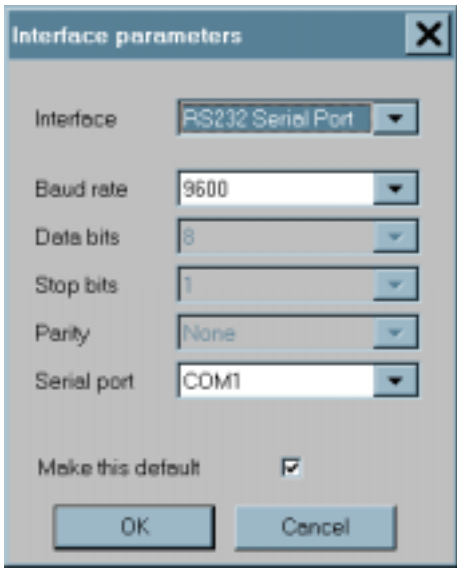
- *Snapshot*. A snapshot of the NanoMechanism's motion on the channel of interest in response to a snapshot stimulus on any channel (not necessarily the same channel).
- *Monitor*. A continuous display of the selected channel's NanoMechanism motion in near real time.

8. **Snapshot/Monitor controls**:
   - *Snapshot controls*. Set up and trigger the snapshot parameters.
   - *Monitor controls*. Set up the monitoring parameters.
9. **Status bar**: Standard MS Windows® status bar (may be hidden using Menu bar, View option).

4.3 Interface Menu

The «interface» menu allows the user to set up the interface between the computer and the NPS3xxx Digital Controller. Selecting the interface menu displays the window shown in figure 4.3.

Figure 4.3: Interface Menu Window



The «Interface» box shows the selected interface.

The NanoControl Panel default is to default to the RS232 interface, which every digital controller has. This is depicted by the tick in the «make this default» box. The «Baud rate» defaults to 9600 and should not be changed.

The «Serial port» defaults to COM1. If required, the RS232 cable may be plugged into the COM2 port and set the «serial port» to COM2

If the digital controller is fitted with the NPS-PAR-A or NPS-PAR-B then the parallel interface option can be selected. The «make this default» box can be ticked while the parallel interface option is selected so that the parallel interface becomes default the next time the system is started.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

4.4 Channel Parameters Menu

The «channel parameters» menu allows access to the linearisation parameters, calibration parameters, yaw and units used by the software. The ability to change these parameters is governed by a software lock (password) to prevent accidental loss of data or system performance.

Unlocking and Locking:

The Locking menu bar then «Unlock Controller» are selected. The Lock Code is to be entered. The Lock Code is set by Queensgate to be the serial number of the Controller (See section 4.8). In this example the Lock Code is 699000.

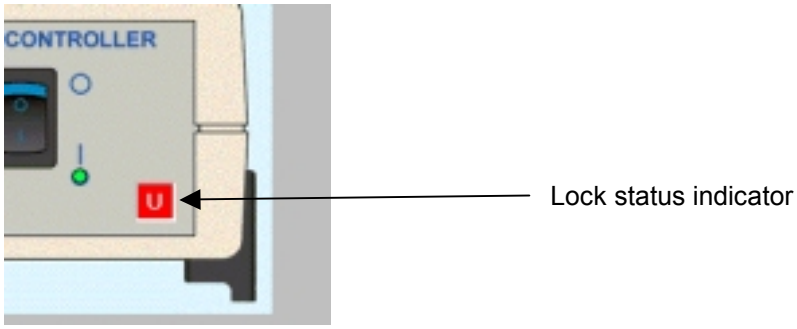
Figure 4.4: Lock Code Window



Unlocking:

Click ‘OK’. The controller is now ‘User Unlocked’. The controller status indicator will now show ‘U’.

Figure 4.5: Unlocking Mark in Digital Controller Screen



Locking:

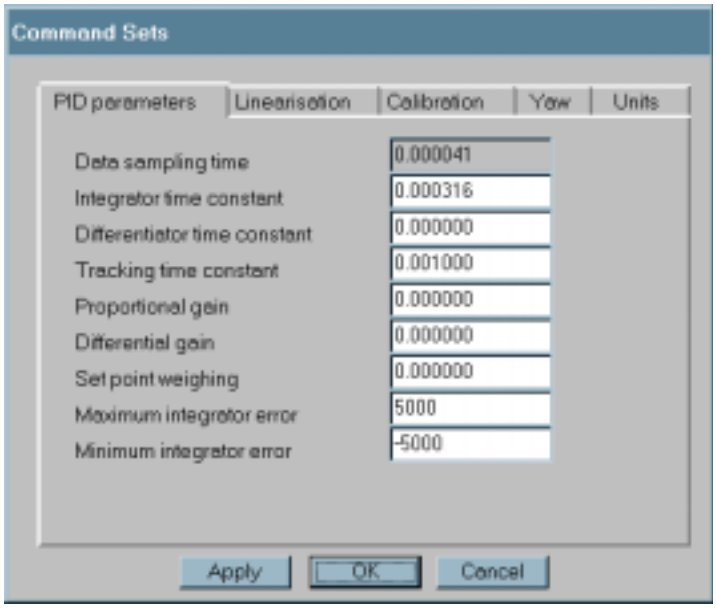
The «Lock Controller» is selected from the Locking menu and the lock status indicator will not display. This means that the controller is now locked.

4.4.1 « PID Parameters »

This menu gives an alternative means of changing the PID parameters. The use of the PID parameters is covered in section 5.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Figure 4.6:



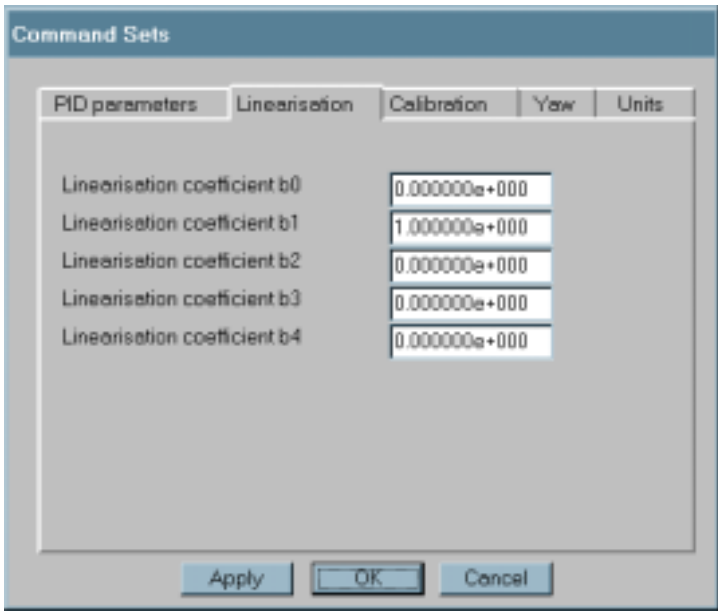
4.4.2 « Linearisation »

The linearisation parameters are the b_0, b_1, b_2, b_3 and b_4 coefficients of the fourth order linearisation polynomial. See the NanoPositioning Book Chapters 2 and 7 for detailed information on these coefficients.

The active channel's linearisation parameters dialog box is accessed from the «Channel Parameters» menu. The linearisation parameters are user password protected to prevent accidental or unauthorized changes being made.

Contact Queensgate (see Section 23) for information on changing these parameters.

Figure 4.7:



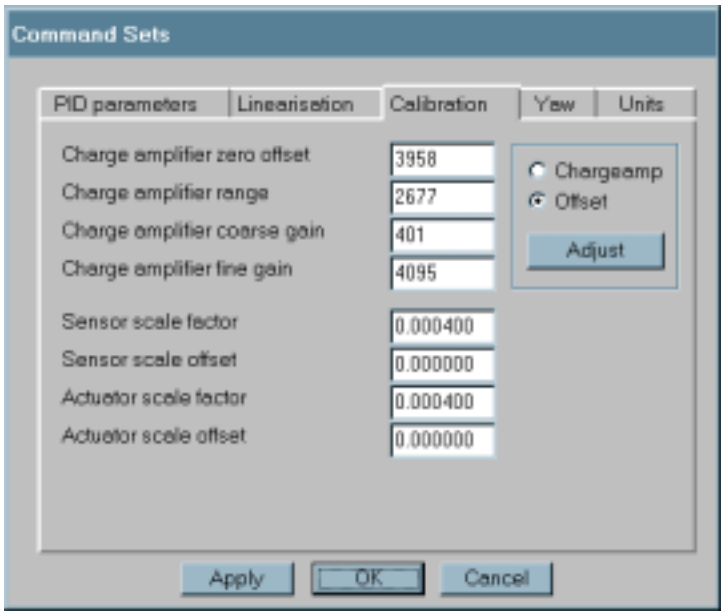
4.4.3 « Calibration »

The active channel's calibration parameters dialog box is accessed from the «Channel Parameters» menu. The calibration parameters are user password protected to prevent accidental or unauthorized changes being made.

Contact Queensgate Ltd (see Section 23) for information on changing these parameters.

The calibration parameters convert system units to micrometers (µm). These parameters are measured at Queensgate Ltd and, in general do not need to be changed by the user. To work in units other than micrometers, use the Units tab in the Channel Parameters dialog box (refer to Table 4.1 and to section 4.4.5).

Figure 4.8: Calibration Tab



4.4.4 « Yaw »

The active channel's yaw parameters dialog box is accessed from the Channel Parameters menu. This facility is used to correct Abbe error if you are measuring or working away from the axis of motion.

Yaw rate is the rotational error of the stage in radians per meter of stage motion (equivalent to microradians per micrometer) in the relevant axis.

Lever arm length is the perpendicular distance off axis.

Contact Queensgate Ltd (see Section 23) for information on changing these parameters.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Figure 4.9: Yaw Tab

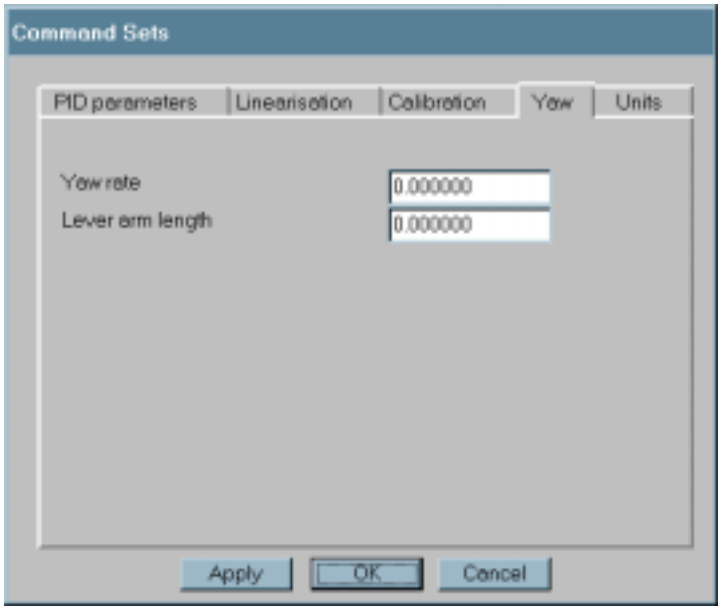
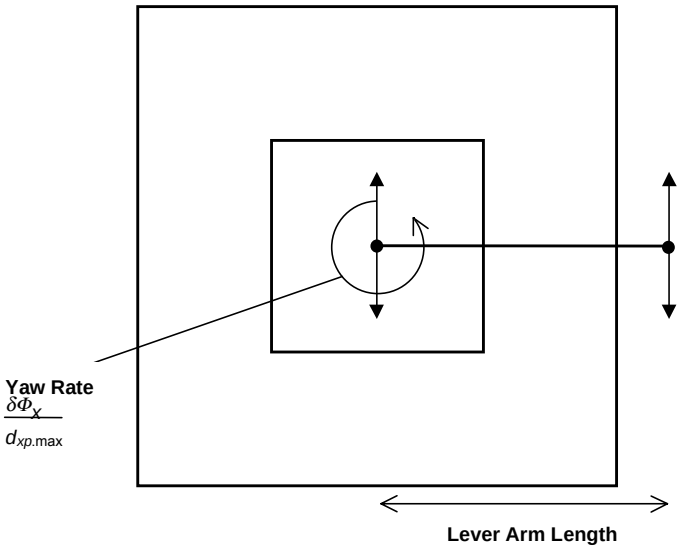


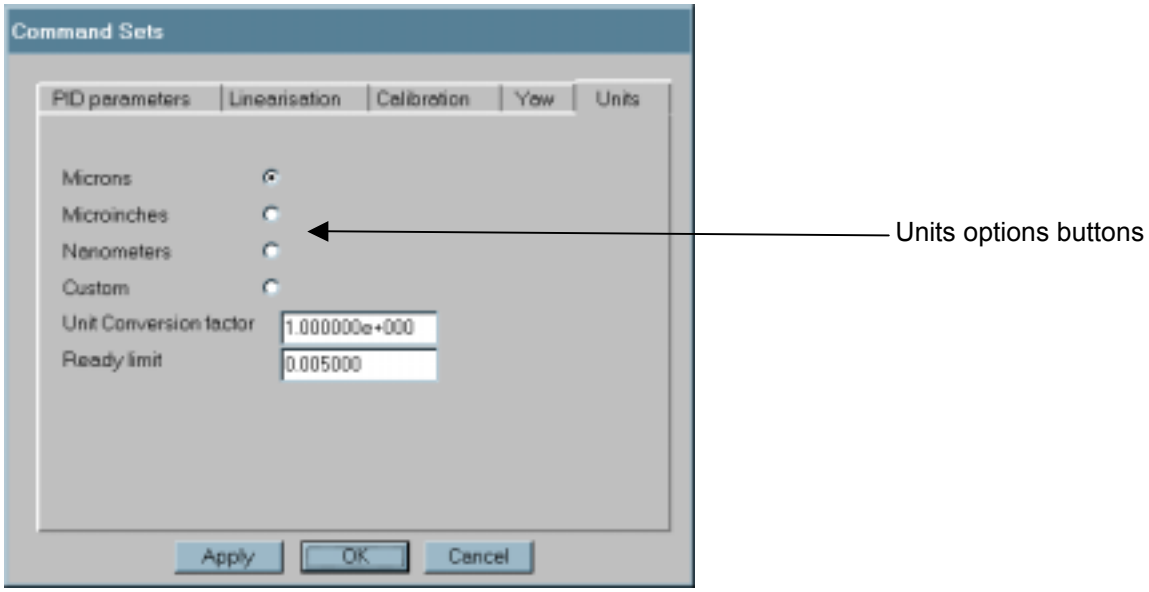
Figure 4.10: Yaw Rate and Lever Arm Length



4.4.5 « Units »

If you wish to change the measurement units used by the Controller, you may do so from the units options buttons on the units tab.

Figure 4.11: Units Tab



- Micrometers, µm. The Units Indicator is reading “um” for micrometers and the Position Numeric Display will be in micrometers as shown in figure 4.12.

Figure 4.12: Position Numeric Display showing the Micrometer Unit



- Nanometres, nm. The Units Indicator is reading “nm” for nanometres and the Position Numeric Display will be in nanometres as shown below;

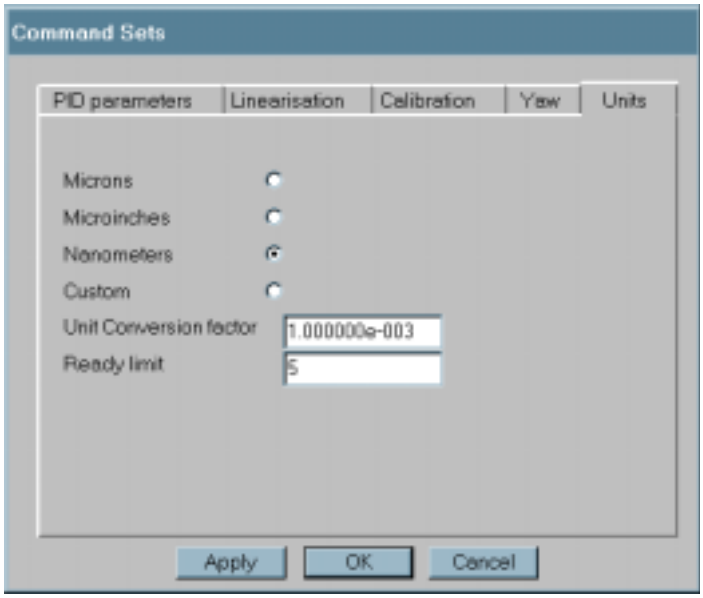
Figure 4.13: Position Numeric Display showing the Nanometre Unit



- Custom: when custom is selected, enter the required conversion factor in the text box, e.g. to work in micro inches enter 0.03937 µinch per µm. The conversion factor is lost when you next change units.

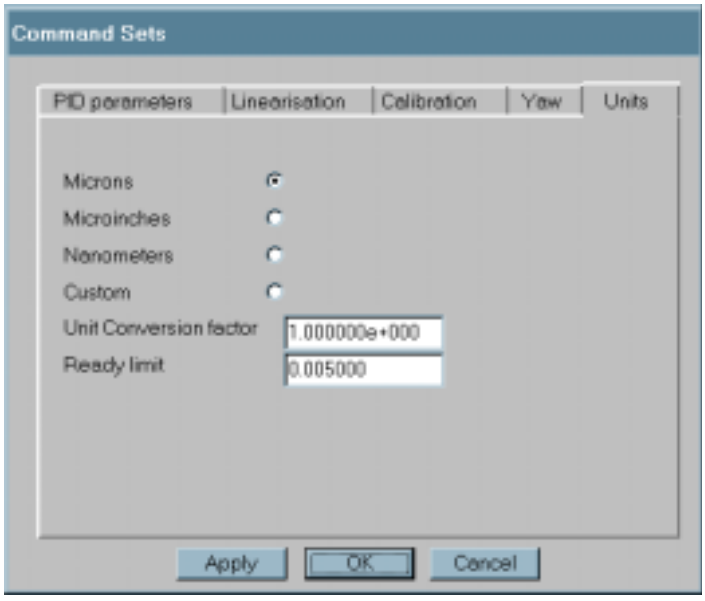
PLEASE NOTE THAT THE UNITS INDICATOR WILL NOT SHOW WHEN EITHER MICROINCHES OR CUSTOM UNITS ARE SELECTED. IT WILL DISPLAY WHICHEVER OF EITHER MICROMETERS OR NANOMETERS WAS LAST DISPLAYED.

Figure 4.14: Ready Limit in Nanometre



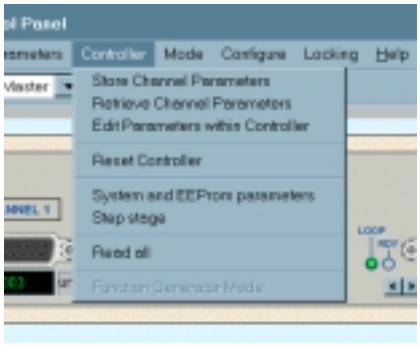
Ready limit: Described in section 3.7. If the units are selected as nanometres as shown above, then the ready limits will be in nanometres. In above the ready limit is set to 5nm (or 0.005µm). Below the units are selected as micrometers and the ready limit is still set to 0.005nm (5 microns).

Figure 4.15: Ready Limit in Micrometer



Note: After a change has been made in the «channel parameters» menu then the «apply» button must be clicked to effect the changes.

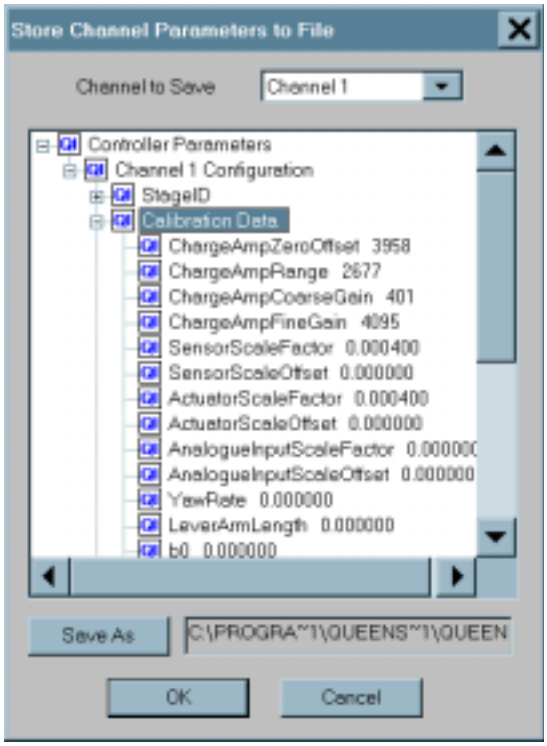
4.5 Controller Menu

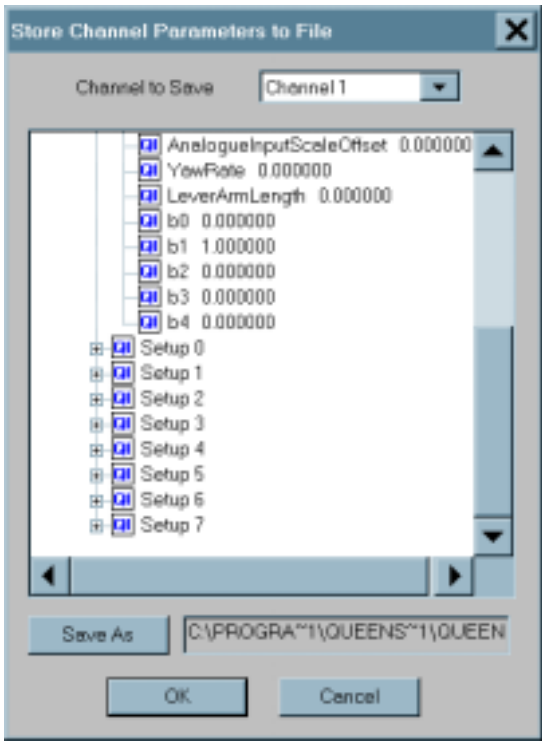


4.5.1 « Store Channel Parameters »

This enables the user to store the complete set of parameters to disk.
Select the channel to be saved in the «Channel to Save» box. Click on 'Calibration Data' to view the parameters. Scroll down to see the linearisation coefficients and the dynamic set-ups (see section 4.7 'Configure Menu'). Please note that this display does not use exponential mathematical notation and is limited to six decimal places. Therefore some of the small numbers may appear to be zero (i.e. b2, b3 and b4).

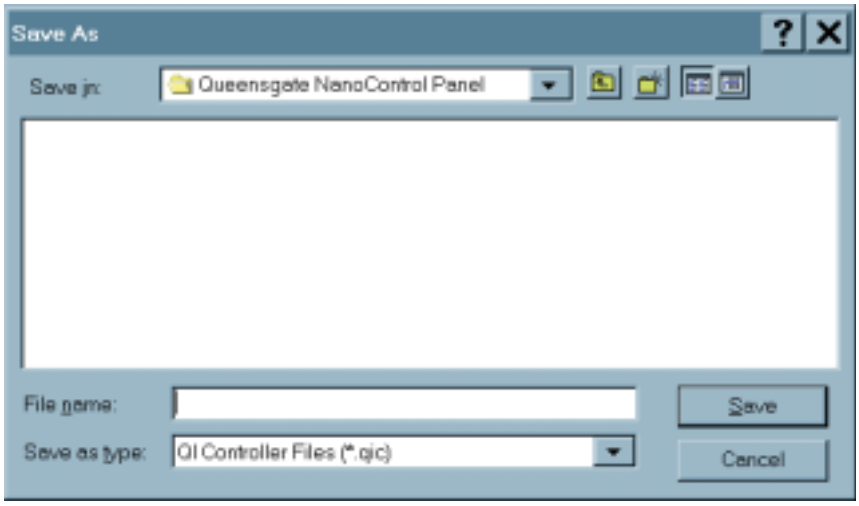
Figure 4.17: Store Channel Parameters Window (in 2 parts)





Click the 'Save As' button to get the box shown in figure 4.18.

Figure 4.18: Save As Window



Select the destination for the file in the 'Save In' box.

Type in a filename with the file extension (.qic). and save.

4.5.2 « Retrieve Channel Parameters »

This enables the user to load from disk a complete set of parameters.

4.5.3 « Reset Controller »

This menu is a software switch off for the controller. The NanoControl Panel will then display the parameters saved in the controller or stage EEPROMS. The saving parameters procedure is covered in Section 4.7.

This will result in unsaved changes to the Controller RAM being lost, and the controller being locked.

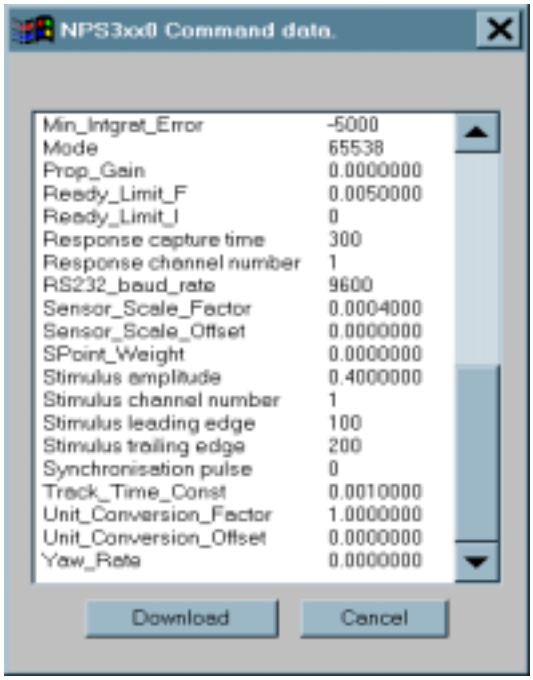
4.5.4 « Edit Parameters within Controller »

Displays all the programmed parameters within the controller except the controller and stage serial numbers.

This provides an alternative method for editing the system parameters.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Figure 4.19: System Parameters Editing Box



4.5.5 «System and EEPROM parameters»

This enables the user to view and save parameters to disk.

Figure 4.20:



4.5.6 «Stage Step»

This performs a square wave step function at the user defined amplitude and frequency. The dialog box below shows how this is achieved.

The ‘Start position’ controls the DC position about which the system will be stepped.

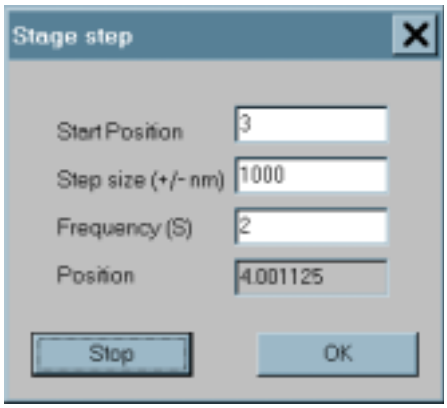
The Step size is specified in nanometres ONLY. The units do not matter.

The frequency defines the period for the step in seconds.

The dialog box below gives an example. The stage will move to a position of 3 microns, then will start stepping ±1000nm from that position. The ‘position’ read back will read back the position to confirm the stepping.

The stage will continue to step until the stop button is pressed. (When finished with the « Stage Step » the stage will remain at the ‘Start position’.)

Figure 4.21: Stage Step Window

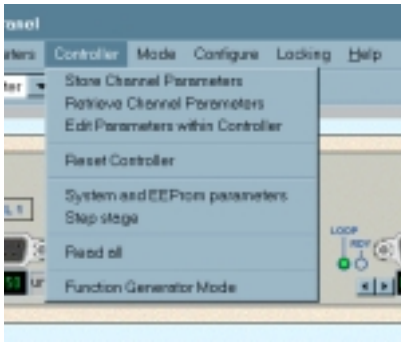


4.5.7 «Read All»

Updates the screen to display the parameters programmed into the controller EEPROM. This is useful when a digital controller is switched on with the NanoControl Panel already opened. If the NanoControl Panel is started after the Digital Controller is switched on, the NanoControl Panel will automatically update. NanoControl Panel does not automatically read the controller but only reads the information when the software is started.

4.5.8 «Function Generator»

Figure 4.22: Controller Menu



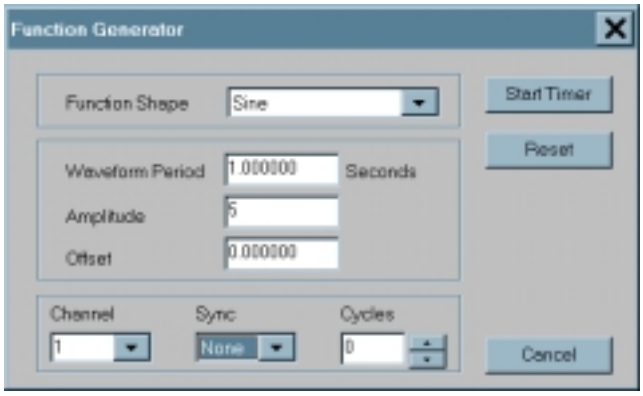
Refer to Section 10 for full details of the NPS-PAR-B parallel interface.

The NPS-PAR-B Interface Function Generator takes amplitude, offset, period and cycle information from the user and then automatically drives the mechanism with the selected waveform without additional user input.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

The Function Generator will operate using the RS232 communications. This option will be only be active if an NPS-PAR-B in present. The Function Generator allows the User to scan the stage with a variety of waveforms. Selecting 'Function Generator' in the Controller Menu brings up the following window.

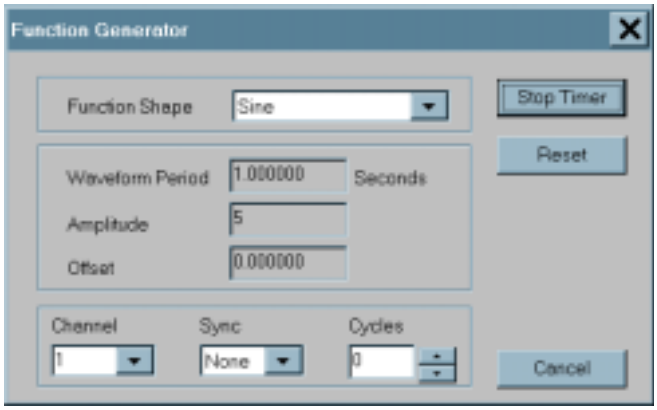
Figure 4.23: Function Generator Window



- Function Shape: The waveform type is selected from a menu under 'Function Shape'.
- Waveform Period, Amplitude and Offset: The waveform details are entered in 'Waveform Period' in seconds, Amplitude and 'Offset' in the units displayed in the Units Display in the NanoControl Panel described in Section 3.7.
- Channel: This indicates to which channel the waveform is applied.
- Cycles: This can be set for the generator to count a number of cycles before stopping.

When the 'Start timer' button is pressed, the window appears like below

Figure 4.24: Function Generator after the 'Start Timer' Button is pressed

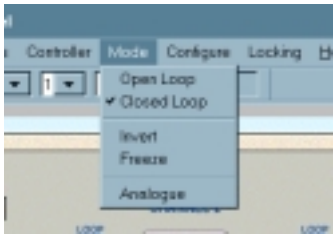


The Waveform type and details cannot be altered during its operation. The 'Stop Timer' button must be pressed first.

To abort a scan; send a *Set Function Generator Mode (Stop)* command. If the *Set Function Generator Mode (Start)* command is sent to the controller then the interface will resume scanning from the same point that it stopped scanning. To reset the scan to the start, press the reset button.

4.6 Mode Menu

Figure 4.25: Mode Menu



The mode menu controls the status of the servo loop. The selection of any of the following is indicated by a tick next to the item.

4.6.1 «Closed Loop»

Closes the servo loop for closed loop operation of the NanoMechanism.

4.6.2 «Open Loop»

Opens the servo loop for open loop operation of the NanoMechanism. The stage position readout described in Section 3.7 will show piezo drift in the stage position

4.6.3 «Invert»

Inverts the phase of the servo loop. The invert setting is set for correct NanoMechanism operation during factory calibration.

4.6.4 «Freeze»

It freezes the output of the Digital to Analogue Converter. This has the effect of opening the servo loop.

4.6.5 «Analogue»

It enables the Queensgate NPS-ANA-A analogue interface (see Section 8) card to be enabled on the selected channel.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

4.7 Configure Menu

Figure 4.26: Configure Menu



This menu is for saving parameters to the Controller and Mechanism EEPROMS.

4.7.1 «Store Dynamic Set-up»

Saves the PID (see Section 5) values displayed in the sliders at the bottom left of the NanoControl Panel screen to the dynamic set-up selected already. If 'all' is selected then all 8 dynamic set-up will be programmed with the current PID values.

4.7.2 «Recall Dynamic Set-up»

Recalls from the EEPROM and activate the PID values programmed into the dynamic set-up selected. The PID sliders will be updated.

4.7.3 «Save system»

Saves all the stage and controller parameters to the digital controller EEPROM. This allows dynamic settings to be saved to the controller.

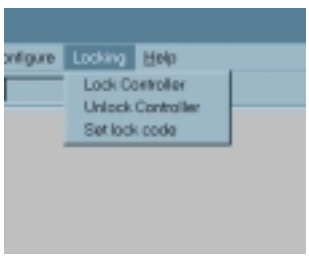
4.7.4 «Save stage»

The customer cannot save parameters to the stage EEPROM.

IF THE DIGITAL CONTROLLER IS EITHER RESET OR SWITHED OFF, THE RAM INFORMATION IS LOST, HENCE CHANGES NOT SAVED TO EEPROM WILL BE LOST.

4.8 Locking Menu

Figure 4.27: Locking Menu



Many of the functions and features of the NanoControl Panel are protected by a password or locking facility. The «Locking» menu gives access to «unlock», «lock» or «change the lock code» of the controller. The reasons for this are obvious. The linearisation and calibration of your NanoMechanism are crucial to the performance of the system. Any change to these values will seriously change the linearity and calibration (and hence the accuracy) of your NanoMechanism. There are three main levels of software unlocking:

- **Operator mode:** This is the normal starting mode of the software, access to change linearisation, calibration, yaw and units factors is denied. THE ABILITY TO SAVE SYSTEM CONFIGURATION IS DENIED.
- **USER unlock:** By entering the unlock code, the user can access all the above, modify the values and save the new system values.

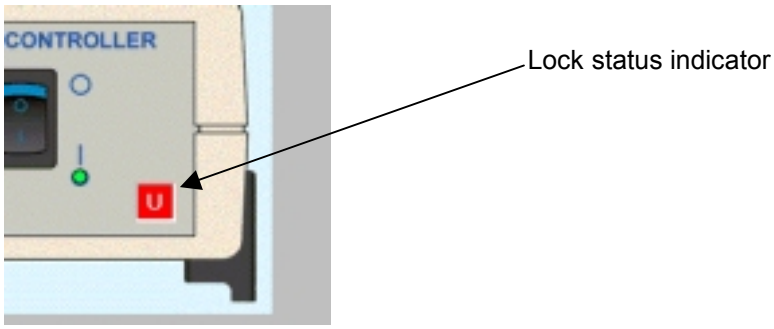
4.8.1 «Lock Controller»

Selecting this will lock the controller. The Lock Status indicator will disappear. Remember that the User cannot save parameters to the controller EEPROM using the « Save system » described in Section 4.7.3 while the controller is locked. They will of course not be able to set the Lock Code.

4.8.2 «Unlock Controller»

Enter the User Lock Code to unlock controller. The Lock Status Indicator will show a 'U'.

Figure 4.28:



This allows the Lock Code to altered if the User wishes. This will allow the User to Save parameters to the controller EEPROM by using the «Save system» described in 4.7.3.

Figure 4.29: Lock Code Window with Serial Number Code

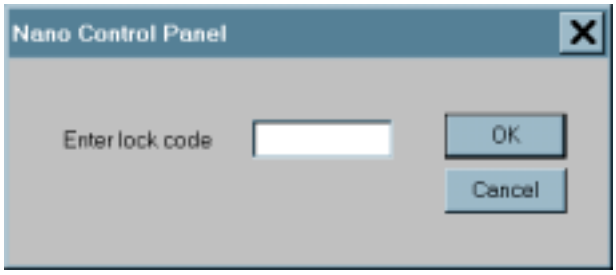


4.8.3 «Set Lock Code»

The Lock Code is factory set as the serial number of your NPS3xxx Digital Controller. The box below allows the user to change the Lock Code if desired. The controller must be unlocked for the User to change the Lock Code.

The user must make a note of changes to the lock code for obvious reasons.

Figure 4.30: Lock Code Window

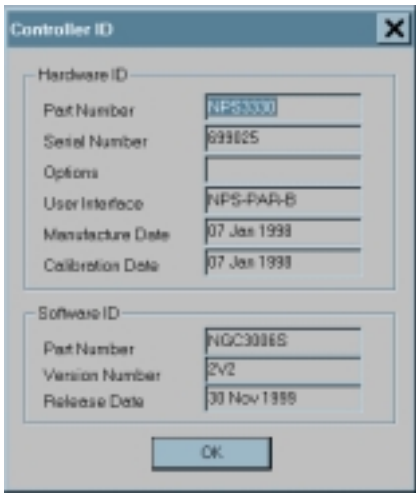


4.9 Help Menu

4.9.1 Controller ID

This gives a window with the details of your digital Controller. The User cannot edit this window. The Part Number, Serial Number, Manufacture and Calibration Dates are programmed into the controller at Queensgate. The Digital Controller acknowledges the presence of any User Interface such as an NPS-PAR-B as shown in figure 4.31. If there is no interface then 'None' appears in the User Interface box.

Figure 4.31: Controller ID Window

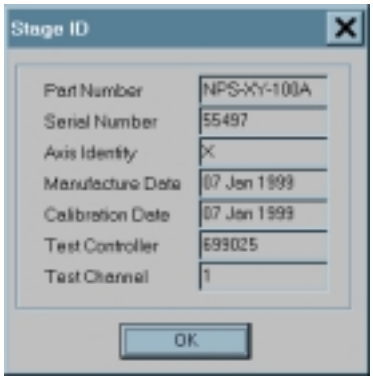


The software ID confirms the type and issue of the controller Firmware.

4.9.2 Stage ID

This display gives the NanoMechanism details. The User cannot edit this window the information is programmed at Queensgate.

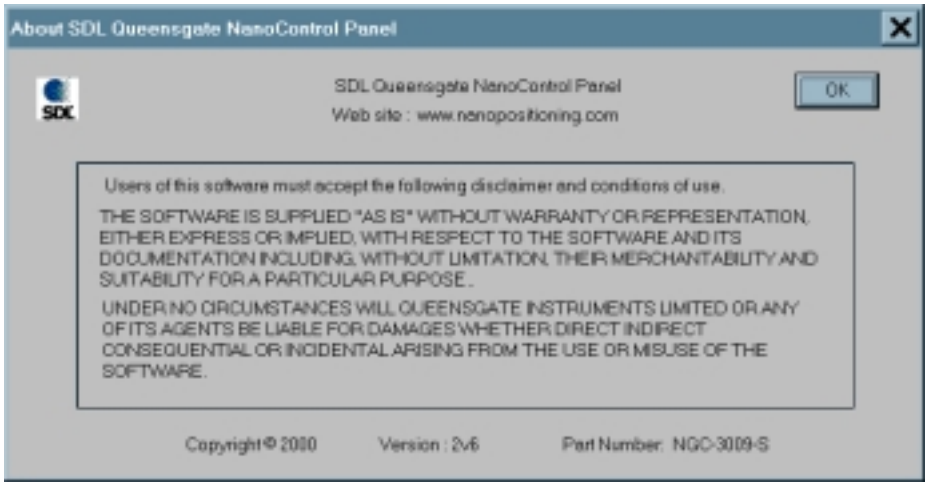
Figure 4.32: Stage ID Window



4.9.3 About Queensgate Limited

This gives software details and disclaimers.

Figure 4.33: “About SDLQ” Window



4.10 NanoControl Panel Menu Summary

Table 4.34: Queensgate NanoControl Panel Menu Summary

Queensgate NanoControl Panel Menu	**Menu Item**	**Select**	**Description**
File	Print Set-up	MS Windows® Print Set-up dialog box	Select your printer options
	Exit		Exit the NanoControl Panel program
Edit	-		-
View	Toolbar	Tick / No tick	Display / Hide toolbar
	Status bar	Tick / No tick	Display / Hide Status bar
Interface	Interface Parameters	Interface Parameters dialog box	Baud rate Select the RS232C Interface baud rate (computer and Controller): 2400, 4800, 9600, 19200 or 38400 baud. Serial Port Select the computer’s serial port: COM1, COM2 or COM3.

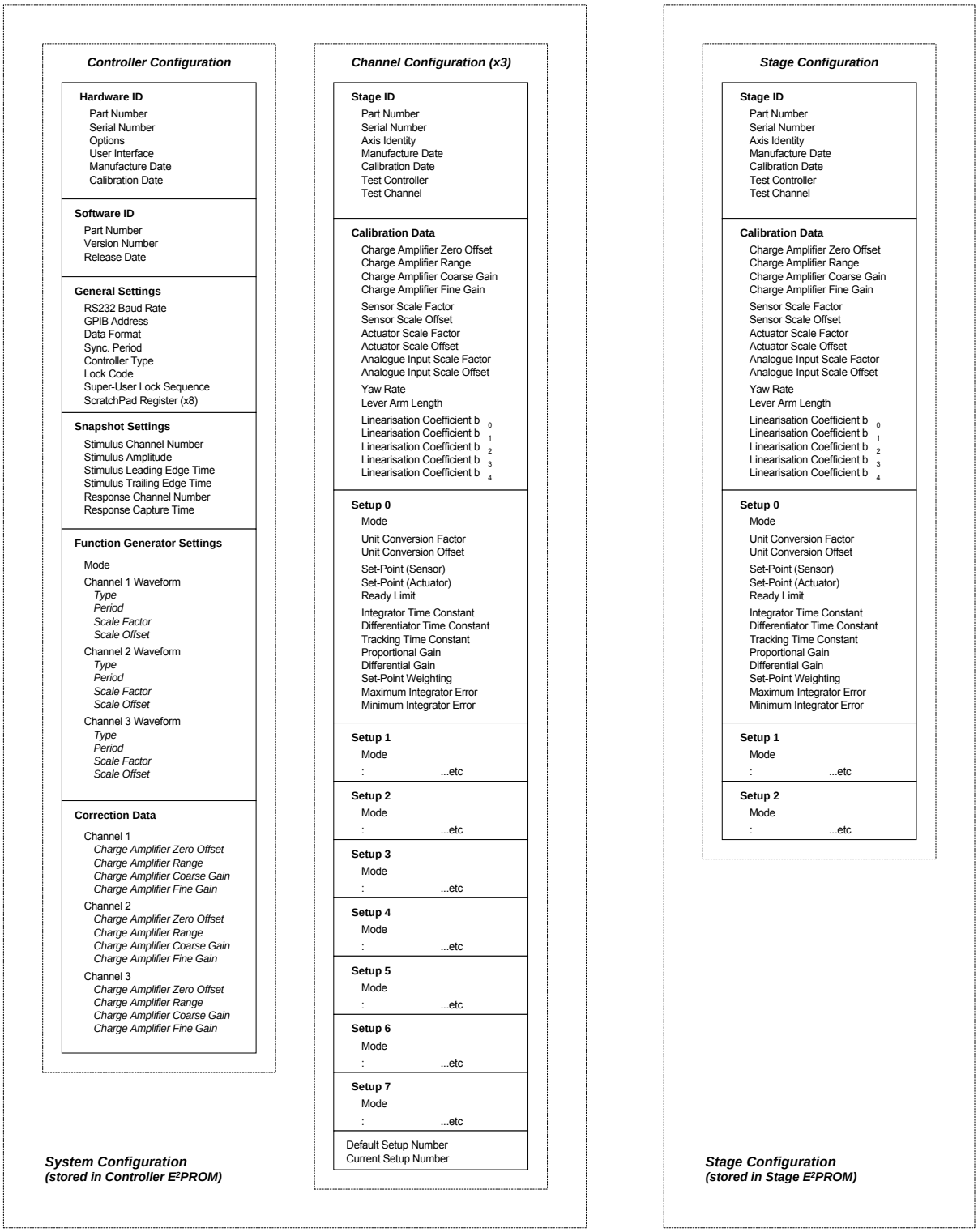
Queensgate NanoControl Panel Menu	Menu Item	Select	Description
			Parallel Interface allows the user to select between the serial or parallel interface (if present). Note the serial interface is the default setting. **Notes** Controller default baud rate at power-up or reset is 9600 baud. Some computer serial ports may not be able to run at the higher baud rates.
Channel Parameters	Channel Parameters dialog box	PID tab	Select and modify the active channel's PID parameters: • Data Sampling Time (t_s) (not variable) • Integrator Time Constant (t_{int}) * • Differentiator Time Constant (t_{diff}) * • Tracking Time Constant (t_t) • Proportional Gain (G_{prop}) * • Differential Gain (G_{diff}) * • Set Point Weighting (G_{sp}) * • Max Integrator Error (e_{max}) * • Min Integrator Error (e_{min}) * **Note** Parameters marked * may also be modified using the PID Sliders.
		Linearisation tab[1]	User Lock dialog box. User password required to enable modification of the active channel's: • Linearisation Coefficient b_0 • Linearisation Coefficient b_1 • Linearisation Coefficient b_2 • Linearisation Coefficient b_3 • Linearisation Coefficient b_4
		Calibration tab[1]	User Lock dialog box. User password required to enable modification of the active channel's: • Charge Amplifier Zero Offset • Charge Amplifier Range • Charge Amplifier Coarse Gain • Charge Amplifier Fine Gain • Sensor Scale Factor • Sensor Scale Offset • Actuator Scale Factor • Actuator Scale Offset
		Yaw tab[1]	Select and modify the active channel's: • Yaw Rate • Lever Arm Length
		Units tab[1]	Select option button for required working units: • Micrometers, µm • Nanometres, nm • Custom (key in conversion factor in text box) • Ready limit. **Note** Custom conversion factor is lost when either Micrometers or Nanometres is selected
Controller	Retrieve Parameters from File	Retrieve Parameters from File dialog box	Enables you to load the Controller with your customized parameters previously saved as a file on your computer
	Store Parameters to File	Store Parameters to File dialog box	Enables you to save your customized Controller parameters as a file on your computer
	Reset Controller	Reset Controller	Software turn off/ on for digital controller
	System and EEPROM parameters		View and save parameters to disk
	Step Stage		Enables the stage to be stepped

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Queensgate NanoControl Panel Menu	Menu Item	Select	Description
	Read All		Updates the NanoControl Panel display
	Edit Parameters within Controller		Password locked facility (not normally available to users), which allows all the Controller's parameters to be edited from one dialog box. Contact Queensgate Instruments if you wish to access this facility.
Mode	Open Loop	Tick / No tick	Tick. To select open loop mode for selected channel. Closed loop then auto deselects
	Closed Loop	Tick / No tick	Tick. To select closed loop mode for selected channel. Open loop then auto deselects
	Invert	Tick / No tick	Tick. To invert the phase of the sensor feedback loop. Used when sensor is mounted 180° out of phase with translator. No tick. Used with NanoMechanism or sensor in phase with translator.
	Freeze	Tick / No tick	Turns On/Off freeze mode (low position noise, fixed position).
Configure	Store Dynamic Set-up	Select from 0 - 7 List[1]	Stores current NanoControl Panel active channel PID parameters in Controller RAM as selected Set-up. **Note:** The data is not stored in Controller EEPROM and will be lost at power-down or reset. To store Set-up data in Controller EEPROM, the Save System Configuration menu option must subsequently be selected.
	Recall Dynamic Set-up	Select from 0 - 7 List	Reloads selected set-up as the active channel's PID parameters in Controller RAM.
	Set Default	Select from 0 - 7 List	Defines the selected set-up as the default set-up on power-up or reset.
	Save System Configuration	User Lock dialog box	User password required to permanently store to Controller EEPROM all the Set-ups that have been stored in Controller RAM by the Store Dynamic Set-up menu option.
Window	-		-
Help	About Queensgate Digital Controller ID and Stage ID		Software notices

[1] Indicates that the software should be **unlocked** to User level or above for this feature to work, or for the data to be stared permanently on the system. See description of Lock/Unlock facility for more details.

Table 4.35: System and Stage Configuration Data Sets



5 Guide to PID Optimisation

5.1 Introduction

Every axis on your NanoMechanism is supplied ready with 3 sets of dynamic set-ups. Each axis is unique and has it's own set of dynamic set-ups pre-configured at Queensgate Instruments. The use of these three parameter sets allows the NanoMechanism to move with different dynamic performances. Usually, these three settings are suitable for most of the different scanning and stepping configurations encountered. There will be configurations where optimisation of the stage response will be required to optimise the settle time or noise of the channel. In these cases, Queensgate have provided the tools to allow the user to adjust the PID settings.

Each dynamic set-up contains **P**roportional, **I**ntegral and **D**ifferential (PID) closed loop parameters. These PID Set-ups are stored in EEPROM on the NanoMechanism and copied into Controller RAM on power-up or reset, a hard copy of these values may also be found in your NanoMechanism test report. The three set-ups are configured as follows:

1. Set-up 0	**Fast**	**Or normal noise**	**Or light mass**
2. Set-up 1	**Medium**	**Or medium noise**	**Or normal mass**
3. Set-up 2	**Slow**	**Or low noise**	**Or large mass**

If you want to optimise the PID parameters to your specific system requirements, you can save another 5 sets of PID parameters for each channel (Set-ups 3 - 7 in the «configure» menu) using the NanoControl Panel. These PID set-ups must be are stored in EEPROMs on the Controller and the Stage, and preserved at power down.

Note: PID Set-ups 0-2 are stored in the Controller EEPROM and are loaded into the Controller RAM at power-up or reset. This may be of use when the channel is being used with closed loop actuator systems that do not have EEPROM PID Set-ups. In this situation the PID Set-ups will be read from the Controller EEPROM on power up / Reset Controller. If the channel is being used with another NanoMechanism with an EEPROM, the PID Set-ups 0 - 2 associated to this NanoMechanism immediately overwrite the Controller PID Set-ups 0 - 2 in the RAM.

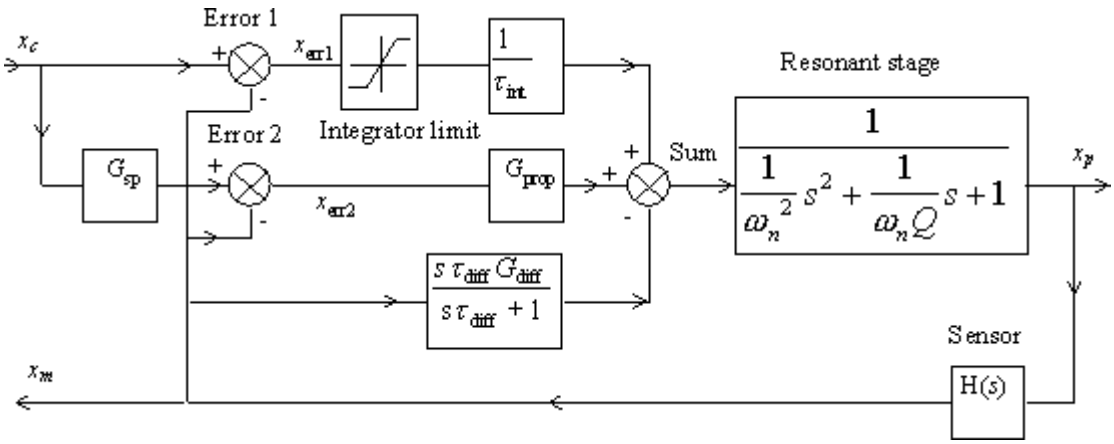
Switching between the different PID Set-up parameters (Set-up 0 - 7) can be accomplished at any time by accessing the «configuration» menu and selecting the «recall settings» sub-menu.

This Chapter outlines how changing PID parameters affects the operation of the system. When optimising the PID parameters you should bear the following in mind:

- Optimising the PID parameters is not easy. You are exploring seven-dimensional parameter space looking for a global minimum among many local minima and poles (unstable regions). The following chart is an image taken from The NanoPositioning Book and outlines the PID loop used in the NPS3000 series of digital controllers.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Figure 5.1: PID loop used in NPS3000 series



Source: The NanoPositioning Book, Hicks and Atherton, 1997.

- The situation is further complicated when different step sizes are used because each step size has its own fastest setting. You can program several different PID parameter sets in the 5 user set up areas (Set-ups 3 - 7 in the Controller EEPROM) and recall the optimum parameters for each step size. For example the fly back step of a raster scan is larger than the individual scan steps.
- A further complication arises in multi axis systems. If you are trying to optimise more than one axis, we recommend adjusting one parameter at a time across the system and ensuring each axis is stable with the dynamic condition of the other axes before adjusting the next parameter. To help explain this, consider having two slow axes and optimising a third axis for very fast response. When the next axis is optimised for very fast response the whole system becomes less stable and can become completely unstable. Remember you can store different PID parameters so you can arrange one axis for slow steps and one for fast steps (xy raster scan for example).

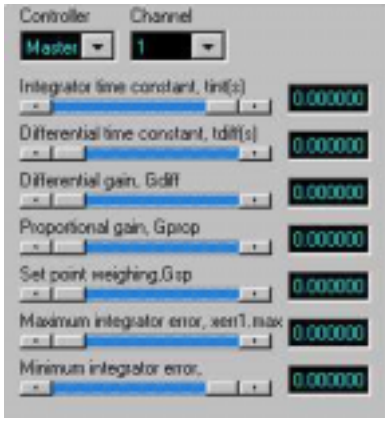
For a detailed description of the PID parameters and servo loop theory, refer to The NanoPositioning Book, Hicks & Atherton, 1997 (www.kogan-page.co.uk/penton/nano.htm).

5.2 Altering PID Parameters

The PID parameters of the active channel may be altered using:

- The PID sliders within the NanoControl Panel software:
  - Sliders, by click and drag,
  - Slider buttons by clicking the buttons (◂ ▸) or,
  - Entering the new values directly in the PID slider parameter numeric display. The 'Enter Key' must be used and the slider will move to reflect the change.

Figure 5.2: PID Parameters Dialog Box



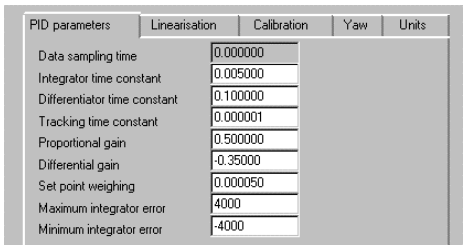
- The PID dialog box from the «Channel Parameters» menu (See section 4.4.1).

Note: to save the PID parameter values as a Set-up of a channel you must use the «Configure menu» selections «Store Dynamic Set-up» and «Save System Configuration». This is outlined in section 4.7. To complete this correctly, you must be USER unlocked.

In general the optimisation procedure is as follows:

1. Unlock your controller to the «user» level.
2. Set the Channel Selector to the channel of interest.
3. If the channel mode is currently Open Loop, change to Closed Loop at the «Mode menu». THE PID PARAMETERS ARE NOT EFFECTIVE IN OPEN LOOP MODE.
4. From the Configure menu select «Recall Dynamic Setup» 0 (nominally, this is the fast setting).

Figure 5.3: PID Parameters Window



5. Perform a few snapshots (This is described in section 3.8.1) with the load and step with magnitudes of interest and observe the results.
6. Repeat (3) and (4) for Set-ups 1 and 2 (nominally medium and slow).
7. Decide which of the Set-ups offers the best performance.
8. Recall that Set-up and use it as a starting point for your PID optimisation for the given load and step magnitude.
9. After each modification to a PID parameter perform a snapshot to view the results of the change.
10. When an optimisation is complete save («Store Dynamic Set-up») the Set-up to Controller RAM using Set-up numbers 3 - 7.
11. Repeat (1) - (9) for the remaining loads, step sizes and channels.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

When all optimisation is complete save («Save System Configuration») the PID parameters to Controller EEPROM.

5.3 PID Parameters

The following examples describe the use of the NanoControl Panel Snapshot facility to optimise a NanoMechanism. This is done via iterative use of the PID parameters and performing snapshots to monitor the system response. Except for the integrator limit parameters, the examples are for low amplitude step position changes (≤1µm).

Note: The numbers and parameter values shown in this Section are general and for guidance only. Unless explicitly stated otherwise, they are not absolute values or limits.

The user adjustable PID parameters are:

- Integrator Time Constant τ_{int}
- Differentiator Time Constant τ_{diff}
- Differential Gain G_{diff}
- Proportional Gain G_{prop}
- Set Point Weighting G_{sp}
- Integrator Limit (Maximum) e_{max}
- Integrator Limit (Minimum) e_{min}

5.4 Integrator Time Constant (τ_{int})

The lower the value of the integrator time constant, the faster the stage responds to a command:

- Always adjust the integrator time constant before the other PID parameters.
- Low δ_{int} values will make the system unstable.

5.5 Snapshot Examples of Small Signal Responses For Different Values of τ_{int}

To see the effects of Integrator Time Constant (τ_{int}), in the following the other parameters were set to zero except the Integrator Limit (Maximum) and Integrator Limit (Minimum). The Integrator Time Constant (τ_{int}) was changed to monitor the effects.

The Small Signal step response is the step response when the step size is small compared to the full movement range of the NanoMechanism. The step size used was 0.5µm.

This is used as an example to see how the PID affects the dynamic performance of a NanoMecham. An Queensgate NPS-X-15A NanoMechanism was used for the Snapshots.

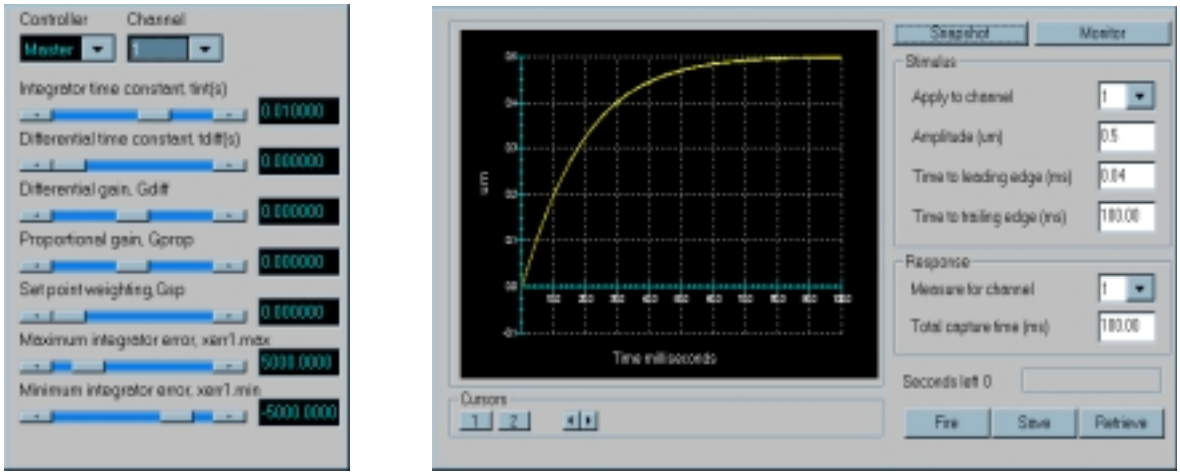
Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

5.6 Stage response

Typical response curves and their interpretation are listed below:

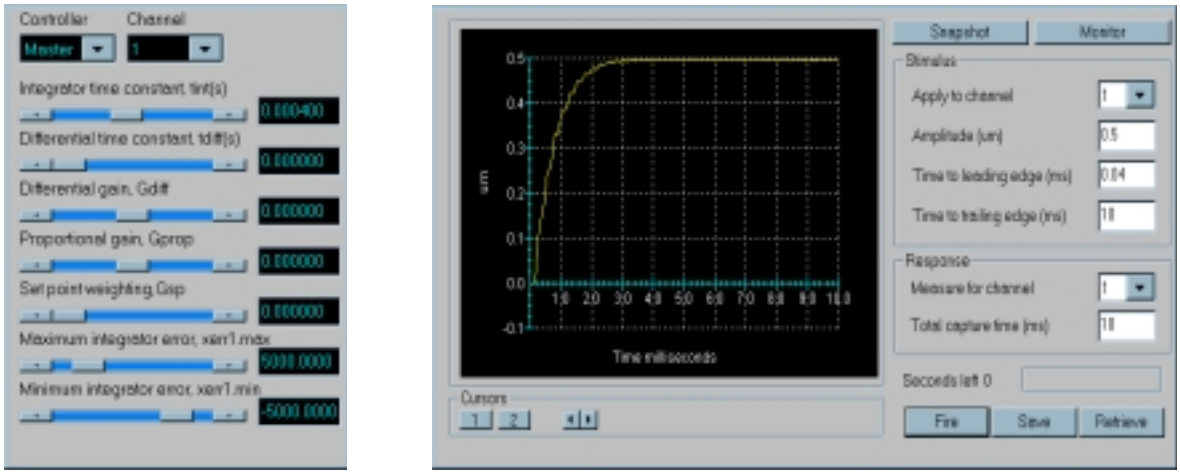
1. Initially, set τ_{int} to a very large value, say 0.01s. This gave a comparatively slow step response. This setting was ideal for low noise, but generally not very practical in dynamic configurations. Note the time base of 100ms.

Figure 5.4



2. Reducing the Integrator Time Constant (τ_{int}) to 0.0004s and performing the snapshot again shows the response to be much faster . For example, the following is a perfectly acceptable response, with good stability and acceptable position noise τ_{int}. This still may be too slow for some applications. The time base is now 10ms.

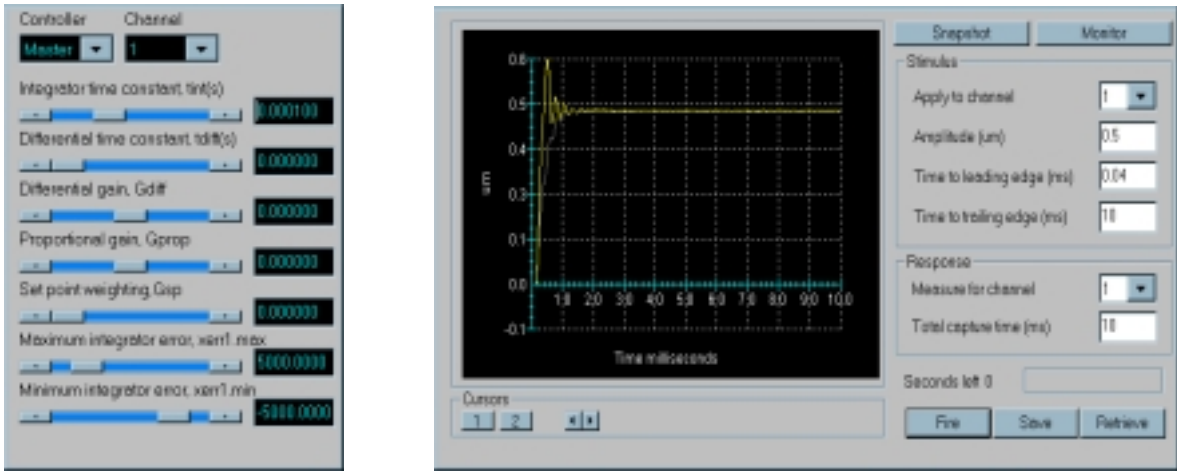
Figure 5.5



3. Reducing the Integrator Time Constant (τ_{int}) further again increased the response time, but note that the servo loop is getting close to instability. Note that the stage took a finite amount of time to damp down to the settle position. In many cases, you will see the stage did not settle. This is would be a good place to start adding in some damping with differential gain and proportional gain to help stabilise the loop.

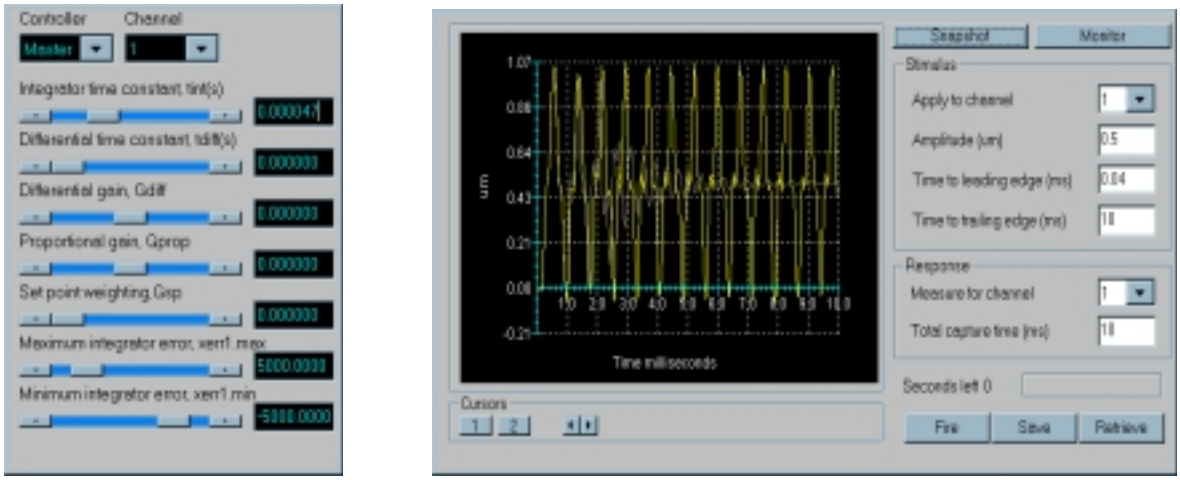
Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Figure 5.6



4. Decreasing τ_{int} to 0.00047s further resulted in an unstable servo loop, so the stage oscillated. This may have even been felt as a vibration or heard as a high-pitched noise. You will have certainly observed this as a huge increase in position noise. The Integrator Time Constant (τ_{int}) would need increasing to stabilise the system.

Figure 5.7



5. For an integrator time constant only PID set-up, the Snapshot in 2. Above is the best response with a setting of τ_{int} = 0.005s. To get a faster settle time while maintaining stability, we would need to reduce the Integrator Time Constant further and add some PID damping.

5.7 Integrator Time Constant Summary

Generally (for most NanoMechanisms):

- Move quickly, or with light load — τ_{int} = low, 0.001 s to 0.0001s.
- Move slowly, or with high load — τ_{int} = high, 0.05s or higher.
- Move with very low noise — τ_{int} = high, 0.1s or higher.

- For most masses <100g:

  τ_{int} = 0.0001s is fast or approaching unstable.

  τ_{int} = 0.01s is slow and stable but not too slow to be impractical.

- For large masses (>100g) τ_{int} = 0.1s should stabilize the loop.

5.8 Differential Feedback (τ_{diff} and G_{diff})

Differential feedback is a damping term. Resonances or oscillations, can be excited by step commands (for example, when τ_{diff} is set to a low value). Increasing the differential feedback term has the effect of reducing these effects. Differential feedback is not easy to set-up.

Note: A small amount of negative proportional gain G_{prop} can also damp resonant oscillations.

Differential feedback adjustment is provided by using the following 2 PID parameters:

- **Differentiator Time Constant (τ_{diff})** should initially be set low (τ_{diff} values of 0.00001 s to 0.0001 s for example) and then increased until the resonances are damped.
- **Differential Gain (G_{diff})** is the amount of damping. The more dominant the oscillation you are trying to damp, the more gain you will need. Generally, G_{diff} values above 3 do not make much difference and values below 0.01 have little effect. Negative values of G_{diff} result in loop instability.

Note: If the differential gain is set to zero (0), there is no differential feedback and adjusting τ_{diff} will have no effect.

It is a feature of the differential gain that it is a high pass filter; that means large G_{diff} values pass high frequency signals round the servo loop. When G_{diff} is 3 (say) the high frequency noise can be heard on the piezo stack.

Note: Although the noise is audible it does not represent large amplitude position noise, $\delta x_{p\cdot n}$, (particularly at high frequencies, for example greater than 3 kHz), however the position noise is slightly higher than if no (or small) differential gain is used. The measurement noise, $\delta x_{m\cdot n}$, is higher than the position noise and can be reduced by filtering (averaging).

Sometimes no differential term gives the best (fastest, most stable) step response.
For low measurement noise, $\delta x_{m\cdot n}$, and low position noise, $\delta x_{p\cdot n}$, use zero differential gain.

5.9 Example Damping of Small Amplitude Resonant Oscillation

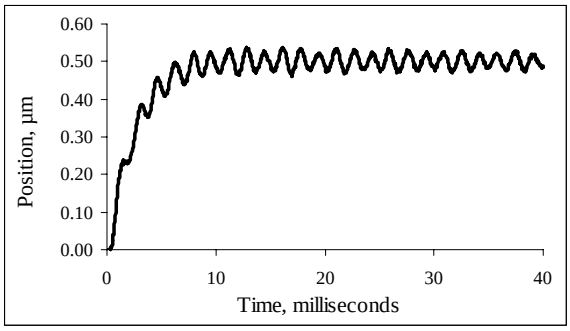
5.9.1 Differential Feedback Summary

The differential terms, τ_{diff} and G_{diff}, are damping terms. Below is an example of a stage that is unstable as a result of a low integrator time constant. In this case applying Differential Gain and Time Constant damps out the oscillations resulting in a much better response.

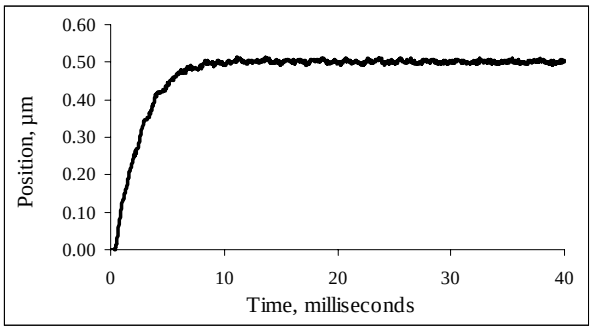
This is not always the case, and as shown in section 5.9.2, applying Differential Gain and Time Constant does not always work, and can make the loop even less stable. This case may need the application of Proportional Feedback discussed in the next section.

Figure 5.8

Small amplitude resonant oscillation excited due to low τ_{int} value, no differential feedback applied.



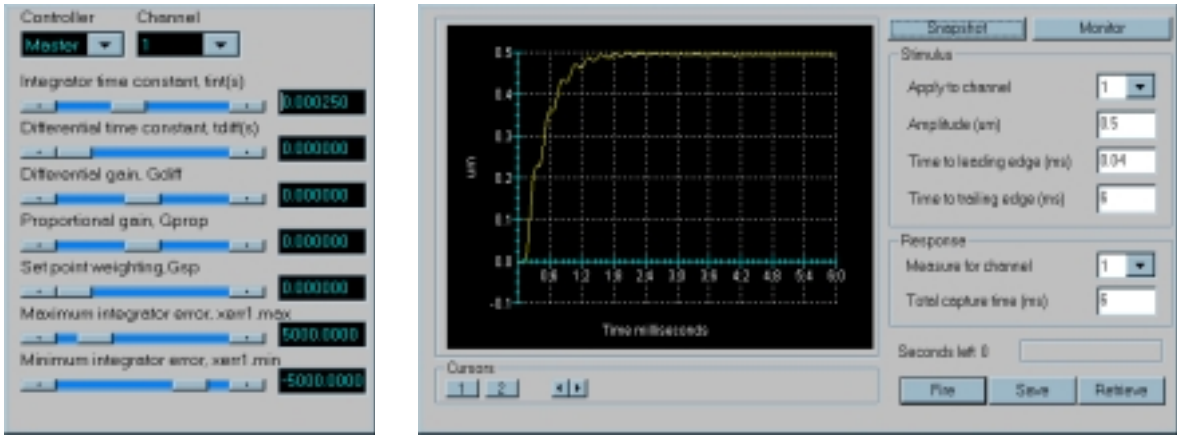
Oscillation damped by the differential term in the closed loop e.g. G_{diff} = 1, τ_{diff} = 0.0001 s



5.9.2 Example: Differential Feedback

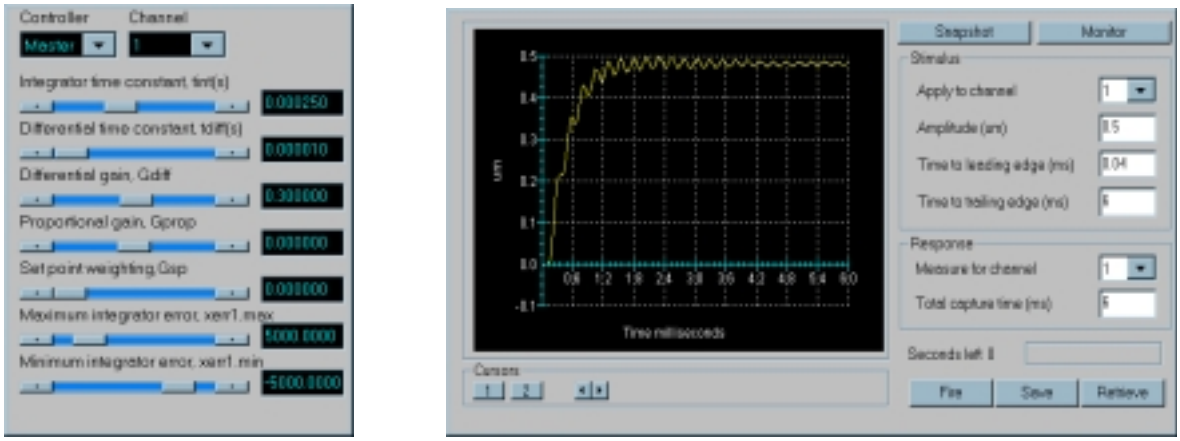
Below is a snapshot for the NPS-Z-15A with the Integrator Time Constant set to 0.00025s. There is some loop instability on the rising edge of the step.

Figure 5.9



The addition of a small amount of differential Feedback gives loop instability as shown below.

Figure 5.10



This shows that the addition of Differential Feedback is not always the solution to loop instability.

Table 5.11

G_{diff} maximum	10
G_{diff} minimum	-10
G_{diff} recommended	0.1 to 2, never < 0
G_{diff} for low noise	0

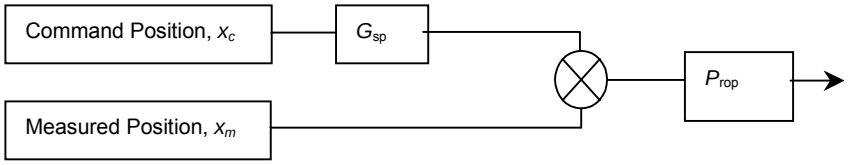
τ_{diff} maximum	0.001 s
τ_{diff} minimum	0 s
τ_{diff} recommended	0.00001 s to 0.001 s

5.9.3 Proportional Feedback (G_{prop} and G_{sp})

Proportional feedback adjustment is provided by using the following 2 PID parameters:

- **Proportional gain (G_{prop})** is the amount of signal fed back which is proportional to (rather than the integral of or differential of) the measured position and the command signal (x_c).
- **Set point weighting (G_{sp})** determines the amount of the command signal fed back.

Figure 5.12

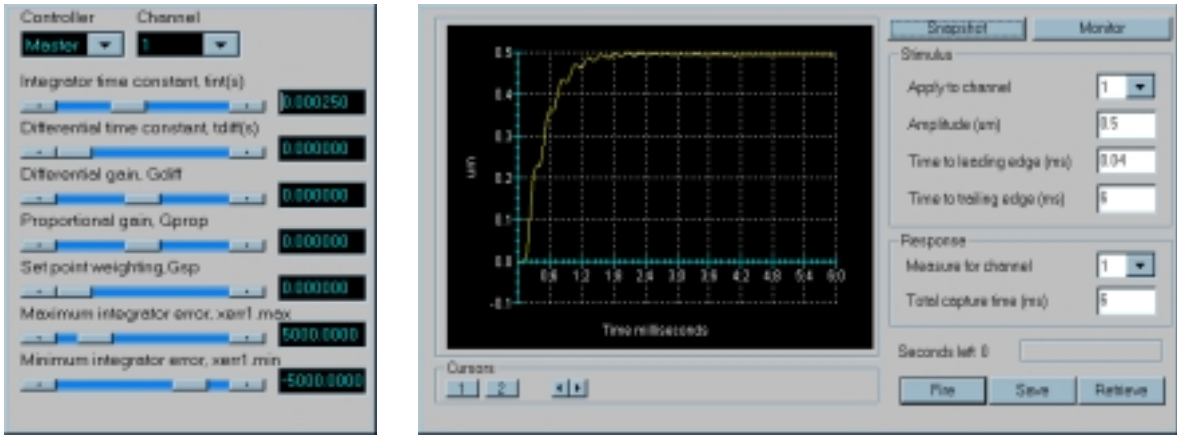


In practical terms a small amount of negative proportional gain (for example –0.1 to -1.0) can remove resonances and sharpen the rise of the waveform.

5.9.3 Example: Negative Proportional Gain

Below is a snapshot for the NPS-Z-15A with the Integrator Time Constant set to 0.00025s.
There is some loop instability on the rising edge of the step.

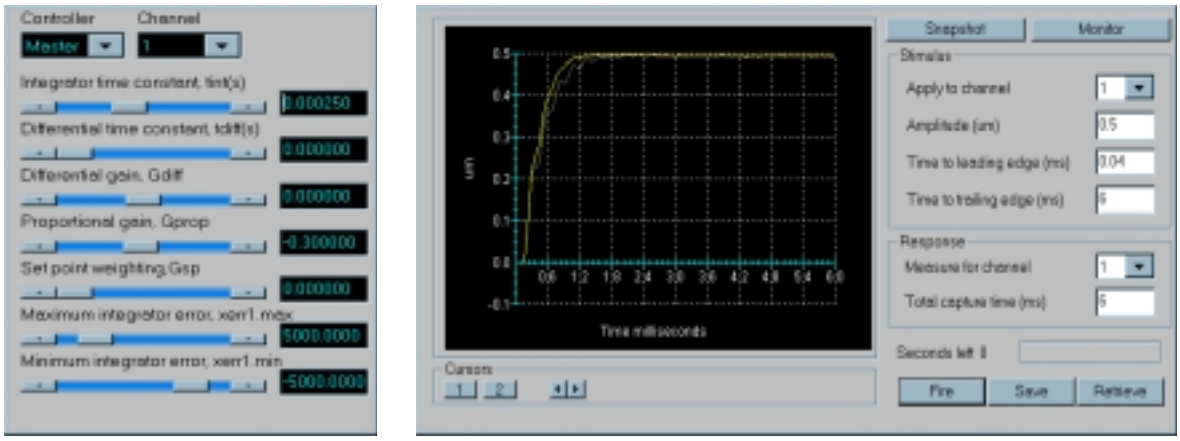
Figure 5.13



The addition of some negative Proportional Gain adds some damping and sharpens the rising edge.

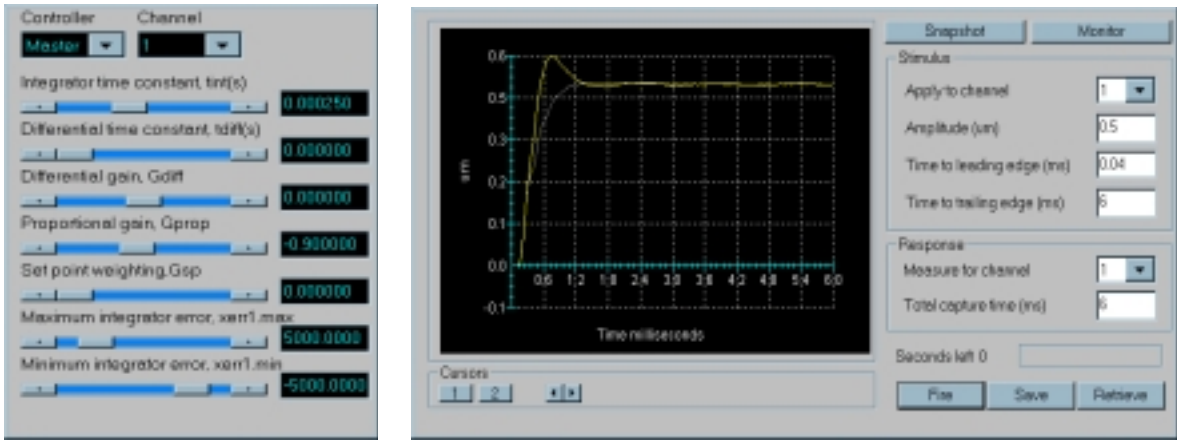
The Proportional Gain was set to – 0.3 to get the response below. The ghost image is the previous snapshot without Proportional Gain shown for comparison.

Figure 5.14



If too much Proportional Gain is applied, then overshoot results as shown below.

Figure 5.15



5.9.5 Proportional Feedback Summary

Proportional feedback affects the rise (the attack) of the response to a step command.

G_{sp} is the set point weighting, it controls the amount of command signal that the proportional gain acts upon. This is normally left set to zero.

For low noise operation, use zero proportional gain, or a small amount of proportional gain (e.g. 0 to 0.5) and set point weighting of 1.

Table 5.16

G_{prop} maximum	10
G_{prop} minimum	-10
G_{prop} recommended	-0.5 is a good place to start, other than that no specific recommendation except 0 for low noise
G_{prop} low noise	0

G_{sp} maximum	1
G_{sp} minimum	0
G_{sp} recommended	0 if G_{prop} is negative

5.9.6 Integrator Limit (e_{max} and e_{min})

Note: In The NanoPositioning Book, e_{max} and e_{min} are equivalent to $x_{err1 \cdot max}$; the relationship is $|e_{max}| = |e_{min}| = x_{err1.max}$.

These parameters affect the large signal step response. They determine the slew rate. If a large signal step response suffers from overshoot, reduce the e_{max} and e_{min} values until there is no overshoot.

e_{max} affects positive step commands, e_{min} affects negative step commands; therefore normally both numbers should be kept equal and opposite (for example e_{max} = 4000 and e_{min} = -4000). An exception to this is a preloaded NanoMechanism where the expansion and contraction have different rates of motion. In this case try experimenting with forward and backward steps to optimise these values.

5.9.7 Integrator Limit Summary

If a large step response suffers overshoot, reduce the integrator error.

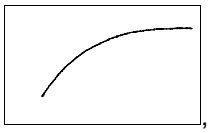
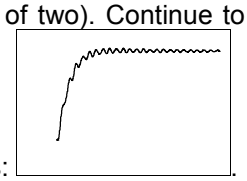
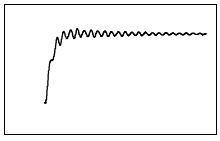
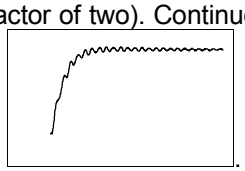
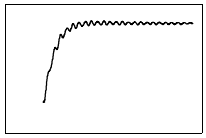
Table 5.17

e_{max} maximum	32767
e_{max} minimum	0 **Note:** Do not set e_{max} = 0
e_{max} recommended	4000 is a general purpose value, other than that no specific recommendation

e_{min} minimum	-32767
e_{min} maximum	0 **Note:** Do not set e_{min} = 0
e_{min} recommended	-4000 is a general purpose value, other than that no specific recommendation

5.10 Dynamic Optimisation Summary

The general method for optimising step response is:

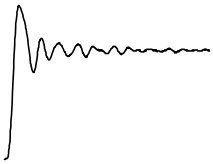
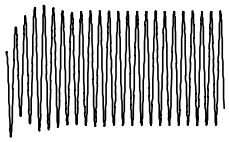
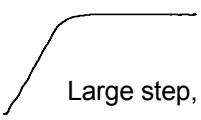
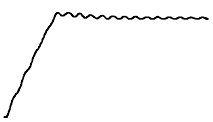
1. Set the differential and proportional gain to zero, G_{diff} = G_{prop} = 0.
2. Set the integrator limit to a general value, e_{max} = 4000 e_{min} = -4000
3. Set the integrator time constant to some value which you know will be stable, for example δ_{int} = 0.005 s
4. Fire a snapshot for a small signal step equal in amplitude to the step size you want to work with, for example 500 nm.
5. If the step response is , decrease the integrator time constant, δ_{int} (e.g. by a factor of two). Continue to decrease the integrator time constant, δ_{int} until the step response is: .
6. If the step response is  then increase the integrator time constant, δ_{int} (e.g. by a factor of two). Continue to increase the integrator time constant, δ_{int} until the step response is .
7. If (or when) the step response is  then start to add proportional and differential feed back.
8. For the proportional feedback try set point weighting, G_{sp} = 0 and gain, G_{prop} = -0.5. Fire another snapshot; if the response overshoots, reduce the gain (G_{prop}). If the response undershoots, increase the gain (G_{prop}).
9. For the differential feedback try time constant, τ_{diff} = 0.00001 s and gain, G_{diff} = 0.5. Fire another snapshot; if the transients are more pronounced, reduce τ_{diff} and / or G_{diff}. If the transients are the same or only slightly less, increase τ_{diff}. If the transients are still pronounced, increase G_{diff} also.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

10. Once you have added proportional and / or differential feedback the loop will be much more stable and you can reduce the integrator time constant, τ_{int}, further (maybe by a factor of two or more). That will help move much quicker, but note that after reducing τ_{int}, the proportional and differential terms often need altered a little also to achieve the fastest response possible.

11. Set the ready limit for the desired settle level.

Table 5.18:

Desired Performance	Actual Performance	Cause	Suggested Action
		τ_{int}, is too high	Decrease τ_{int}, by a factor of two and adjust G_{diff}, τ_{diff}, G_{sp}, and G_{prop} to improve the speed of response
		τ_{int}, is too low	Increase τ_{int}, by a factor of two and / or try increasing τ_{diff} and G_{diff} to damp the ringing
		G_{prop} is too -ve	Make G_{prop} a little closer to zero
		Not enough damping	Make G_{diff} and / or τ_{diff} larger. And / or make G_{prop} ~ -1
		G_{prop}<0 and G_{sp}>0	Make G_{sp} = 0
		The system is unstable	Increase τ_{int}, significantly and / or make G_{prop} = G_{diff} = 0
 Large step, >10µm for example		e_{max} is too large	Reduce e_{max}

Desired Performance	**Actual Performance**	**Cause**	**Suggested Action**
 Large step <-10µm for example		e_{min} is too large and negative	Adjust e_{min} closer to zero
Low noise	-	-	Make τ_{int}, = 0.05 to 0.1 and G_{prop} = 0 and G_{diff} = 0 and G_{sp} = 1
		The servo loop is completely unstable. For example G_{diff} < 0, or large positive G_{prop}	Make τ_{int}, = 0.02, G_{diff} = 0, G_{prop} = 0. Or recall one of the default setups, or switch the controller off and then on again
		The servo loop phase is incorrect	Invert the loop phase

6 RS232 Parallel Interface

The RS232C interface is fitted as standard and enables control and monitoring of the Controller from a computer's standard serial COM port. This interface allows a command rate of up to 100 commands per second, depending on the baud rate setting. Connection is via the Queensgate Instruments supplied Null Modem cable to the 9-way D-Type connector on the Controller's rear panel. If your computer uses a 25-way serial port connection, you will need a 9-way to 25-way adapter.

The Digital Controller is configured as an RS232C DTE (*Data Terminal Equipment)*.

The Digital Controller's factory default power-up configuration is:

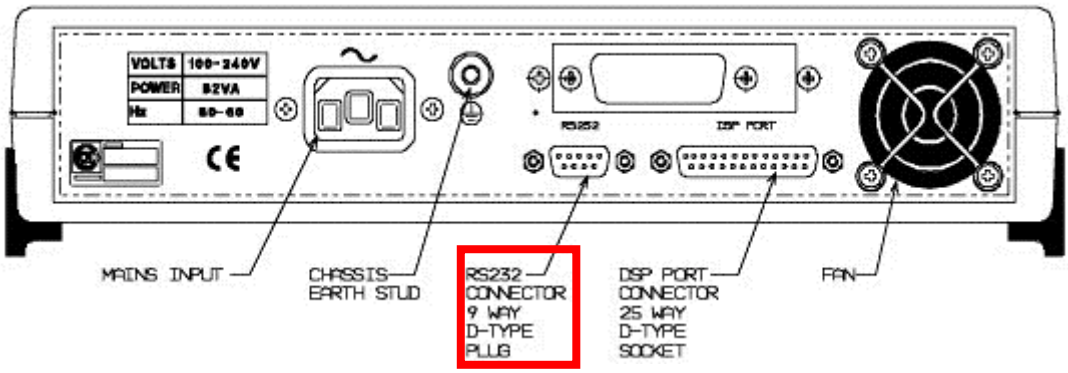
- Baud Rate: 9600 baud
- Data bits: 8
- Parity: None
- Start bits: 1
- Stop bits: 1

The baud rate can subsequently be re-configured by software command (e.g. from the NanoControl Panel's Interface menu) to be: 2400; 4800; 9600; 19200; or 38400 baud.

Table 6.1: RS232C Interface D-Type Connections

Pin	Signal		Pin	Signal	
1	DCD	*(Data Carrier Detect)*	6	DSR	(*Data Set Ready*)
2	RX	*(Receive data)*	7	RTS	*(Ready to Send)*
3	TX	*(Send data)*	8	CTS	*(Clear to Send)*
4	DT *(Data Terminal ready)*			RI	*(Ring Indicator)*
5	GND	*(Ground)*			

Figure 6.2: Position of RS232C Connector



Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

7 DSP Interface (QI bus)

This high-speed interface is fitted as standard. It has a command rate capacity of up to 4000 commands per second for either:

- Master/Slave interconnection of 2 Controllers to enable up to 6 channels to be controlled from a single computer interface,

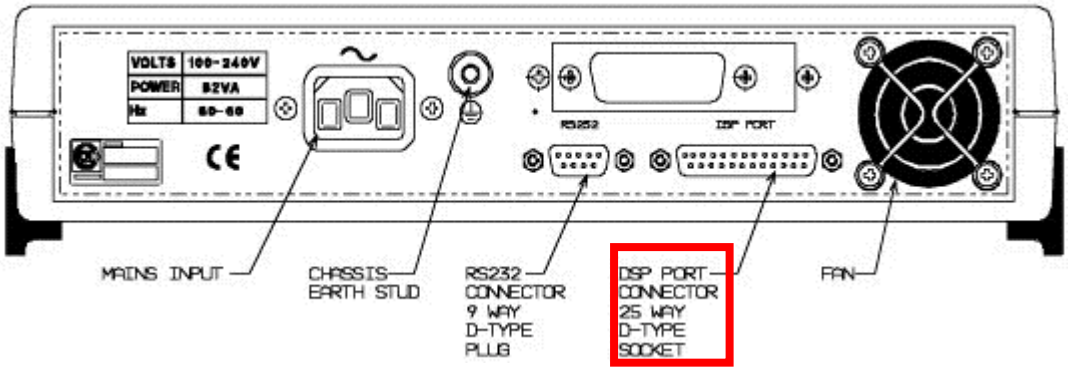
or

- As a serial interface for sending commands from another TMS320C32 DSP Board.

The Controller is fitted with a Texas Instruments TMS320C32 DSP board, which provides one 25-way D-type connector for the DSP Interface cable (Figure 7.1).

For further information on sending commands via the DSP interface please refer to Section 14 and the complementary Manual containing the Command Reference Library.

Figure 7.1: DSP Port Connector



8 NPS-ANA-A Analogue Interface Card

8.1 NPS-ANA-A Device Description.

The NPS-ANA-A interface is the optional analogue interface for the NPS3000 series of Digital Controller. The interface has three male BNC connectors, which allow the NPS3000 series to be driven with an input voltage range of ±10V.

Warning: Do not exceed ±20V analogue input range as this may damage the interface.

The NPS-ANA-A is designed so that position input commands can be given to the system using either the analogue or digital interfaces, either alone or combined as a summing input.

The NPS-ANA-A allows the system to be used as a stand-alone module without the need for a host PC, once the system has been optimised dynamically for the users load.

Each system will be set up at Queensgate Ltd with the NPS-ANA-A enabled as default and ready for operation, refer to the quick start guide for further details.

NPS-ANA-A Quick Start Guide.

- Set up the actuator and NPS3330 control electronics in the required position
- Connect the actuator to the control electronics
- Connect the NPS-ANA-A to the analogue input source
- Command the analogue input between ±10V for full range actuator motion

8.2 Input Connectors

Interfacing to the NPS-ANA-A can be achieved using the three Male BNC connectors fitted to the back panel of the NPS-ANA-A interface. There is one BNC connector for each channel of the NPS3000 series.

If using an NPS3220 the analogue input for channel 3 is ignored by the controller. If using an NPS3110 the analogue inputs for channels 2 and 3 are ignored by the controller.

The input impedance of each BNC input is 10kΩ.

8.3 NPS-ANA-A Operation

When a system with an NPS-ANA-A is delivered the system will be set up with the analogue interface enabled as the default interface.

This option must be enabled for each channel of the system if the system to be driven using the analogue interface.

The interface is enabled using one of the following procedures (8.3.1 or 8.3.2). These commands can be sent via the RS232 interface or the DSP serial port interface.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

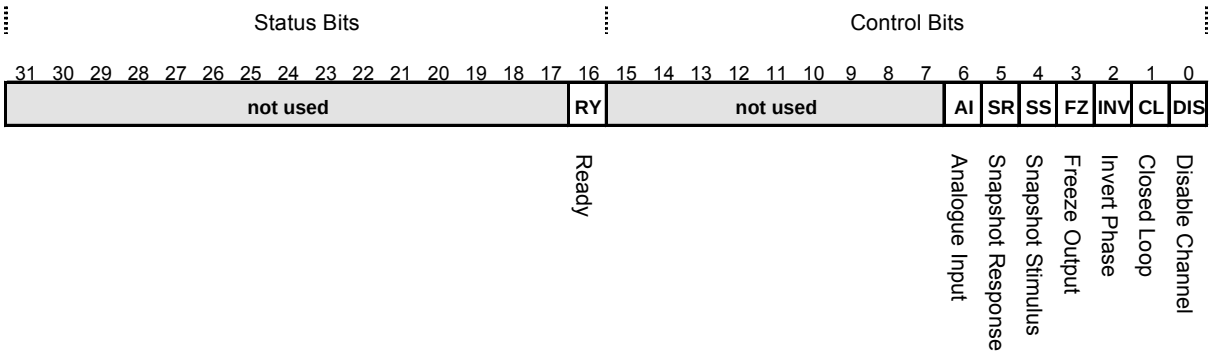
8.3.1 Enabling/Disabling the analogue interface using the NPS3000 series Command Language

To enable or disable the analogue interface using the NPS3000-series Command language complete action 3.1.1

8.3.1.1 ***set_mode()*** Command code 67 decimal

The interface can be enabled in user software by sending a *set_mode()* command[1]. To enable the analogue interface the Analogue Enable Bit (Bit 6) of the mode/status word should be set to 1. To disable the Analogue interface this bit should be set to 0.

Figure 8.1



8.3.1.2 ***read_mode()*** Command code 66 decimal

If unsure whether the Analogue interface is enabled or disabled the mode word can be checked by sending a *get_mode()*[1] command to the controller.

[1]Further information about the *set_mode()* and *read_mode()* commands can be found on page 93 of the Command Language Reference Manual.

8.3.2 Enabling / Disabling the analogue interface using NanoControl Panel Software

The interface can be enabled or disabled using the NanoControl Panel Software (version 1V4 onwards). To enable/disable the interface or check the mode status perform the following procedure.

- Start NanoControl Panel software
- Select the *Mode* menu
- Select the Analogue option by ticking the *analogue* menu item. To disable the analogue interface the menu item should be un-ticked.

8.3.3 Saving the Configuration to the EEPROM using NanoControl Panel Software

To save the analogue interface status (enabled / disabled) the following procedure should be carried out. Saving will allow the system to be powered down and restarted without the need for a host computer.

- Run the NanoControl Panel software

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

- Unlock the system to user level by selecting *unlock controller* in the *Locking menu* and entering the user lock code.
- Select the required channel (the mode status is channel specific so this procedure will need to be carried out from this point onwards for all system channels)
- Select the *Mode* menu
- Select the Analogue option by ticking the *analogue* menu item. To disable the analogue interface the menu item should be un-ticked.
- Select the *Configure* menu
- Select *Save System Configuration.*
- Select *Save Stage Configuration*

The *Save System Configuration and Save Stage Configuration* commands store the current programmed parameters into the system and stage EEPROMs. Before saving the configurations check that the correct parameters are programmed into the system e.g. calibration values, units, mode, commands.

8.3.4 Scanning the system using the NPS-ANA-A

The system can now be scanned using the analogue interface.

Each NPS-ANA-A channel has an analogue input operating rage of ±10 Volts. The Analogue_Input_Scale_Factor specified in the system test report will allow the analogue input (volts) to be converted to commanded stage motion (micrometers).

*Analogue Input(V) * Analogue_Input_Scale_Factor (µm·V^{-1}) + digital command (user units) = Position (µm)*

Details of how to change the *Analogue_Input_Scale_Factor* can be found in Point 11.4.1. Digital commands are commands set using either the RS232 or DSP interface ports.

Warning: Do not exceed ±20V analogue input range as this may damage the interface.

If a digital command is present this will sum with the analogue command to give the total command input for the actuator. The measured position display in the NanoControl Panel software will always show the actual position of the actuator (sum of analogue and digital commands).

To use the analogue input only and remove any position offset caused by digital position commands, the digital command should be set to zero.

8.3.5 Maximum Command Bandwidth

The NPS3000 series Digital Controller reads the analogue input voltage from all three NPS-ANA-A interface channels every 120µs. The NPS-ANA-A therefore has a maximum command bandwidth of 4kHz.

The overall system bandwidth will be lower than this as there are factors other than the command bandwidth which must be taken into account. These include the stage resonant frequency, the closed loop system bandwidth and the scan amplitude.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

8.4 NPS-ANA-A System Calibration

Each NPS3000 series system fitted with an NPS-ANA-A is calibrated at Queensgate Ltd so that the analogue input voltage range corresponds to the typical closed loop actuator range for the particular mechanism. For example:

A 15µm actuator will be calibrated so that ±9.0V analogue input corresponds to ±8.1µm motion. A 100µm actuator will be calibrated so that ±9.0V analogue input corresponds to ±54µm actuator motion.

This results in a typical analogue input scale factor of $0.9\mu m.V^{-1}$ for a 15µm actuator and $6.0\mu m.V^{-1}$ for a 100µm actuator.

This calibration of the system is achieved by setting the *Analogue_Input_Scale_Factor* and the *Analogue_Input_Scale_Offset* to the desired values. This can be achieved by using the NPS3330 command language or the NanoControl Panel software supplied, refer to Points 11.4.1 and 11.4.2 for further information.

8.4.1 Setting the Analogue Input Scale Factor and Offset using the NPS3000-series Command Language

The *Analogue_Input_Scale_Factor* and *Analogue_Input_Scale_Offset* can be set using the NPS3000-series command language using the commands detailed in Points 8.4.1.1 and 8.4.1.2**.**

8.4.1.1 read / set_analogue_input_scale_factor() command code (144/145) decimal
Format = Floating-point

The *Analogue_Input_Scale_Factor* is used to scale the analogue command input in volts to give the desired motion at the actuator. It is only a scaling value[3] and can be used to change the input sensitivity of the analogue input.

If greater analogue input sensitivity is required then the *Analogue_Input_Scale_Factor* should be reduced so that a smaller total scan range is observed for the same voltage input.

[3]As the *Analogue_Input_Scale_Factor is only a scaling value changing its value will not effect the overall calibration of the system.*

The *Analogue_Input_Scale_Factor* can be changed by sending a *set_analogue_input_scale_factor()* command. Further information about the *set_analogue_input_scale_factor()* command can be found in Section 14 and on page 16 of the Command Language Reference Manual.

The *Analogue_Input_Scale_Factor can be interrogated using the read_analogue_input_scale_factor()* command detailed on page 15 of the Command Language Reference Manual.

Note: If the *Analogue_Input_Scale_Factor* is changed the *Analogue_Input_Scale_Offset* will also need to be adjusted to maintain 0 microns position for 0 Volts analogue input.

8.4.1.2 read / set_analogue_input_scale_offset() command code (146/147) decimal
Format = Floating-point

The *Analogue_Input_Scale_Offset* is used to give an overall offset to the system so that when there is no command input[4] the system is set to a particular position, e.g. +50µm, depending on the desired scanning operation of the system.

When the systems are calibrated the *Analogue_Input_Scale_Offset* is set so that 0nm position is obtained for 0v analogue input.

If a larger position offset is required then this can be set using the *read / set_analogue_scale_offset()* command. Details of the *read / set_analogue_scale_offset()* command can be found on pages 13 and 14 of the Command Language Reference Manual.

8.4.2 Setting the Analogue Input Scale Factor and Offset using NanoControl Panel.

The Analogue Input scale factor and Analogue Input Scale Offset can be set using NanoControl Panel software using the procedure detailed in 8.4.2.1 and 8.4.2.2.

8.4.2.1 Setting the Analogue Input Scale Factor or Analogue Input Scale Offset

To change the current value in the controller, perform the following actions.

- Start NanoControl Panel software
- Select the *Controller* menu
- Select the *Edit Parameters within Controller* menu
- Select the Analogue_Input_Factor or Analogue_Input_Offset value by double clicking the required line in the table
- Enter the new value and press ‘okay’
- Click on Download

8.4.2.2 Storing the Analogue Input Scale Factor or Analogue Input Scale Offset into the system EEPROMs.

To store the values into the controller and stage EEPROMs perform the following actions.

- Run the NanoControl Panel software
- Unlock the system to user level by selecting *unlock controller* in the ‘Locking menu’ and entering the user lock code.
- Select the *Controller* menu
- Select the *Edit Parameters within Controller* menu
- Select the Analogue_Input_Factor or Analogue_Input_Offset value by double clicking the required line in the table
- Enter the new value and press ‘okay’
- Click on Download
- Select the *Configure* menu
- Select *Save System Configuration.*
- Select *Save Stage Configuration*

The *Save System Configuration and Save Stage Configuration* commands store the current programmed parameters into the system and stage EEPROMs. Before saving the configurations check that the correct parameters are programmed into the system e.g. calibration values, units, mode, commands.

Further information on programming software for this interface can be found in the Section 14, Command Language.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

9 NPS-PAR-A Fast Parallel Interface Card

9.1 Description

The NPS-PAR-A interface provides a high-speed parallel link between the NPS3000-series Digital Controller and the host computer system. The NanoControl Panel software is supplied which has the NPS-PAR-A communication control built in.

It is specifically designed to interface with the National Instruments™ PC-DIO-24 24-line digital input/output card. Queensgate Instruments strongly recommends that anyone that wishes to use this interface purchases this card from National Instruments™. The NPS-PAR-A is supplied with cables and software designed specifically for communication between the NPS-PAR-A and the PC-DIO-24 card. Technical details concerning the operation of the NPS-PAR-A parallel interface are given below.

9.2 Input / Output Connectors

Interfacing to the NPS-PAR-A can be achieved using the 50-way high-density connector provided, for parallel interface connection to the PC-DIO-24 PC based card.

9.3 Parallel Interface Operation

9.3.1 Command String

1. The parallel interface uses a command string of three 16-bit words to communicate. The string consists of a 32-bit data word (made from 2 x 16-bit words) and a 16-bit command word. Refer to Section 9.4 for further information on how to program for the NPS-PAR-A interface.
2. The string is sent in that order LSWord, MSWord, command word. The data and command words are sent using the digital lines (D15 – D0) in the NPS-PAR-A interface connections table, table 1.
3. Four control lines and two Status bits, BUSY-READ & BUSY-WRITE control handshaking between the computer and PAR-A interface. The status bits are multiplexed onto the bottom two bits of the data bus.
4. The NPS3000 series Digital Controller Command Language Reference, reference NPS-3023-M, contains a full list of the command words for the NPS3000.
5. The CRLF termination is not necessary.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Figure 9.1

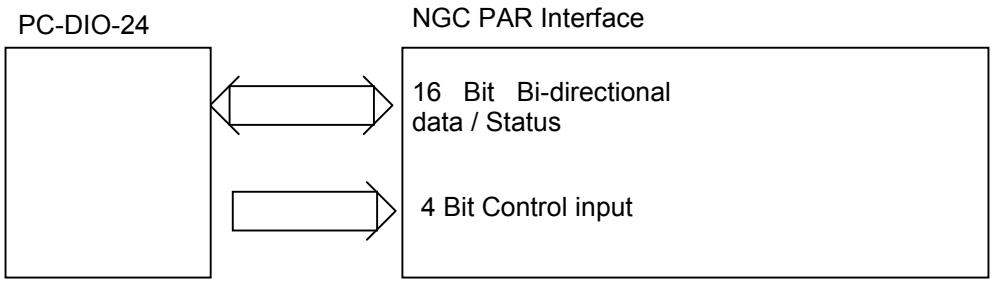


Table 9.2: PC-DIO-24 Pin Connections

Pin Number	Function	Direction
Port A (PA0-PA7)	**Lower byte for 16-bit word / Status**	Bi-Directional
Port B (PB0-PB7)	**Upper byte for 16-bit word**	Bi-Directional
PC0	Not Used	PC-DIO-24 output to NPS-PAR-A
PC1	Not Used	PC-DIO-24 output to NPS-PAR-A
PC2	Not Used	PC-DIO-24 output to NPS-PAR-A
PC3	Not Used	PC-DIO-24 output to NPS-PAR-A
PC4	**A0 register address line**	PC-DIO-24 output to NPS-PAR-A
PC5	**A1 register address line**	PC-DIO-24 output to NPS-PAR-A
PC6	**R/WBAR Control bit**	PC-DIO-24 output to NPS-PAR-A
PC7	**STROBE Control bit**	PC-DIO-24 output to NPS-PAR-A

9.3.1.1 Write Cycle

A transfer from the computer to the NGC PAR interface is illustrated in figure 9.3 below. Initially the R/WBAR control line is high. In this state the PAR interface drives the 16-bit data bus and the PC-DIO-24 must have these lines (Port A & Port B) configured as inputs.

With R/WBAR high the STROBE Control line selects between Data and Status. With STROBE high Data is driven onto the 16 Bit bus as selected by the A1, A0 address lines. With STROBE low the bottom two bits of the 16 Bit data bus are driven by two status bits, BUSY-READ (D0) and BUSY-WRITE (D1).

A. At point A in figure 9.3 the STROBE line goes low with R/WBAR high. The status bits are driven onto the data bus. The computer polls the BUSY-READ status line (D0) until it detects that it is low. A low level on BUSY-Read indicates that the PAR interface is ready to receive the next command from the computer.

B. At point B the computer takes the R/W Bar line Low to start the write cycle from computer to PAR interface. At this point the 16-bit data bus becomes tri-state, as it is not being driven from the computer or the PAR interface. The computer drives the Address lines to address 2 prior to the 1st data transfer, the least significant word of the 32-Bit data word (LSW). Note that the CRLF terminator at address 3 is optional.

C. The computer re-configures the 16-bit data bus as outputs and writes the 16-bit data to the ports.

D. The STROBE Line is driven high then low. This Latches the Data into the PAR interface.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

D to E – The computer sets up the remaining address and data and strobe most significant word (MSW) and command string into the PAR interface. The command string with address A1, A0 = 0 must be the last word written. When A1, A0 = 0 the rising edge of STROBE causes BUSY-READ (and BUSY-WRITE) to be set high.

E. The computer re-configures the 16 Bit data bus as inputs.

F. The computer drives the R/WBAR line high. With STROBE low the status Bits, BUSY-READ and BUSY-WRITE, are driven onto the bottom two bits of the data bus for the computer to poll.

Figure 9.3: Computer write cycle to PAR Interface

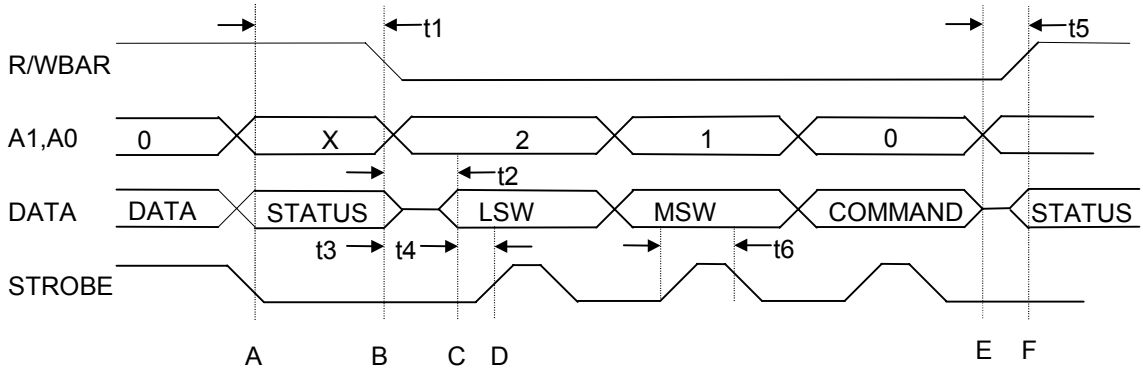


Table 9.4: Time t_x Description of Figure 9.3

	Description		**Max**	
t_1	Delay time, BUSY-READ low to R/WBAR low	0		ns
t_2	Set up time, R/WBAR low before DATA lines driven	40		ns
t_3	Set up time, Address stable before STROBE high	55		ns
t_4	Set up Time, DATA stable before STROBE high	55		ns
t_5	Delay time, R/WBAR high after DATA High Z	20		ns
t_6	Cycle time, STROBE high	160		ns

9.3.1.2 Read Cycle

A transfer from the NGC PAR interface to the computer is illustrated in figure 9.5 below. The R/WBAR control line is high. In this state the PAR interface drives the 16-bit data bus and the PC-DIO-24 must have these lines configured as inputs.

With R/WBAR high the STROBE Control line selects between Data and Status. With STROBE high Data is driven onto the 16-bit bus as selected by the A1, A0 address lines. With STROBE low the bottom two bits of the 16-bit data bus are driven by two status bits, BUSY-READ (D0) and BUSY-WRITE (D1).

A. At point A in figure 9.5 the STROBE line is low with R/WBAR high. The status bits are driven onto the data bus. The computer polls the BUSY-WRITE (D1) status line until it detects that it is low, indicating that the NPS controller has finished writing data to the PAR interface, and that the computer is allowed to read it.

B. At point B the computer takes the STROBE line high. Data rather than status is driven onto the 16 Bit bus. The data is selected by the A1, A0 address lines.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

C. The computer changes the address and the corresponding data appears on the data bus.

D. The computer changes the address and the corresponding data appears on the data bus. The data string can be read in any order. Reading the CRLF terminator is optional.

Figure 9.5: Computer Read cycle from PAR interface

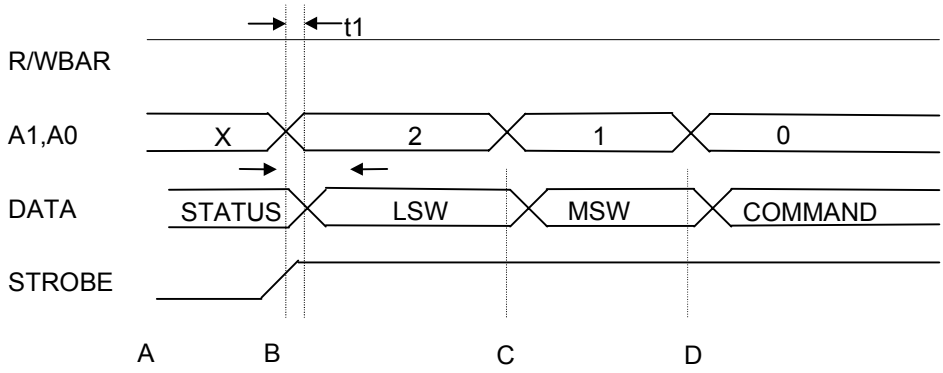


Table 9.6: Time t_x Description of Figure 9.5

No	Description	Min	Max	Unit
t_1	Delay time, DATA valid after Address stable		55	ns

Notes: 1/ The falling edge of BUSY-WRITE (D1 status bit) indicates that the NPS controller has written data in response to a command, and that the data is available to the PC.
2/ The CRLF terminator is not necessary.
3/ The address value indicates the value of the control lines A1and A0. Where A0 is the LSB.
4/ The data string can be read in any order.
5/ BUSY-WRITE is set high after a computer command string write cycle.

Table 9.7: NPS-PAR-A Parallel Interface Connections

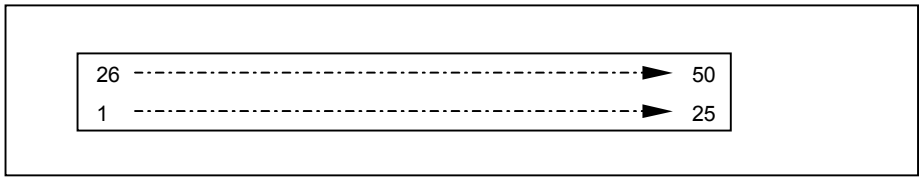
Pin Number	Function	Pin Number	Function
1	**Strobe**	2	Not Used
3	**R/W Bar Control Bit**	4	Not Used
5	**A1**	6	Not Used
7	**A0**	8	Not Used
9	Not Used	10	Not Used
11	Not Used	12	Not Used
13	Not Used	14	Not Used
15	Not Used	16	Not Used
17	**Data Line D15**	18	Not Used
19	**Data Line D14**	20	Not Used
21	**Data Line D13**	22	Not Used
23	**Data Line D12**	24	Not Used
25	**Data Line D11**	26	Not Used
27	**Data Line D10**	28	Not Used

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Pin Number	Function	Pin Number	Function
29	**Data Line D9**	30	Not Used
31	**Data Line D8**	32	Not Used
33	**Data Line D7**	34	Not Used
35	**Data Line D6**	36	Not Used
37	**Data Line D5**	38	Not Used
39	**Data Line D4**	40	Not Used
41	**Data Line D3**	42	Not Used
43	**Data Line D2**	44	Not Used
45	**Data Line D1 / BUSY-WRITE**	46	Not Used
47	**Data Line D0 / BUSY-READ**	48	**Digital Ground**
49	Not Used	50	**Digital Ground**

Viewing the controller from the back the pins are numbered as shown in figure 9.8.

Figure 9.8: NPS-PAR-A parallel interface viewed from behind



9.4 Sample Software

Please Note: This source code requires NI-DAQ Diver Software installed on the target machine for operation under MS Windows® 95 and NT.

```
/*     *** CODE FRAGMENT ***
/
/      PURPOSE: TO DESCRIBE A METHOD TO TRANSFER DATA AND COMMANDS
/      WITH THE WIN NT COMPATIBLE PAR INTERFACE (QI) AND AN NPS3***
/      CONTROLLER (QI) IN VISUAL C++ USING NI-DAQ DRIVER SOFTWARE
*/

/*
/      NGC Parallel Data Transfer (Write and Read Methods)
/      to support New Style PAR Interface for operation
/      using NI-DAQ Driver Software on Windows 95 and Windows NT
*/

#include <windows.h>
#include <stdio.h>
#include "dio24.h"
#include "nidaqcns.h"        /* National instruments header */
#include "utils.h"

#define LOW                  0
#define HIGH                 1
#define NOHANDSHAKE  0
#define HANDSHAKE            1
#define INPUT                0
#define OUTPUT               1
#define BIDIRECTIONAL2
```

```
#define BUSY_READ           0
#define BUSY_WRITE          1
#define SYNC                2
#define A0_REGISTER         4
#define A1_REGISTER         5
#define RWBAR               6
#define STROBE              7

static FARPROC Get_DAQ_Device_Info;
static FARPROC DIG_Prt_Config;
static FARPROC DIG_In_Port;
static FARPROC DIG_Out_Port;
static FARPROC DIG_In_Line;
static FARPROC DIG_Out_Line;

static BOOL NIDAQ_Init = FALSE;     // True when pointers to NI-DAQ functions  initialised

/*
/     Write data and commands to a NPS3*** series controller
/     via PC-DIO-24 interface using NI_DAQ functions.
*/
int NIDAQ_DIO_Transfer_Write(int device, long data, int commandword, long *result)
{
      int status, t;
      int linestate;
      int patternA = 0;
      int patternB = 0;
      int hiword;
      int loword;
      char lobyte;
      char hibyte;
      char CurrentState;

      // If first time round initialise function pointers
      if (NIDAQ_Init == FALSE)
      {
            Init_NIDAQ_Functions();
            NIDAQ_Init = TRUE;
      }

      // Configure Ports A & B as inputs and Port C as output
      status = DIG_Prt_Config(device, PORT_A, NOHANDSHAKE, INPUT);
      status = DIG_Prt_Config(device, PORT_B, NOHANDSHAKE, INPUT);
      status = DIG_Prt_Config(device, PORT_C, NOHANDSHAKE, OUTPUT);

      // Take the RWBAR line High
      //status = DIG_Out_Line(device, PORT_C, RWBAR, HIGH);
      CurrentState = 0x40;
      status = DIG_Out_Port(device, PORT_C, CurrentState);

      // If BUSY-READ Status is high wait for 10 microseconds
      linestate = 0;
      for(t = 0; t < 50; t++)
      {
            status = DIG_In_Port(device, PORT_A, &linestate);

            if((linestate & 0x01) == 0x01)
                  ExactMicroDelay(10);
            else
                  t = 50;
      }


      // Reconfigure the PC-DIO-24 data lines
      status = DIG_Prt_Config(device, PORT_A, NOHANDSHAKE, OUTPUT);
      status = DIG_Prt_Config(device, PORT_B, NOHANDSHAKE, OUTPUT);

      // Set A0 Low and A1 High
      CurrentState = 0x20;
      status = DIG_Out_Port(device, PORT_C, CurrentState);


      // Write LSB of NGC command string to Data Bus
      loword = HIWORD(data);
      hibyte = HIBYTE(loword);
      lobyte = LOBYTE(loword);
```

```
	status = DIG_Out_Port(device, PORT_A, (int)lobyte);
	status = DIG_Out_Port(device, PORT_B, (int)hibyte);

	// Take the STROBE line high to LATCH in the LSB data
	CurrentState = CurrentState | 0x80;
	status = DIG_Out_Port(device, PORT_C, CurrentState);

	// Take the STROBE line low and set A0 High and A1 Low
	CurrentState = 0x10;
	status = DIG_Out_Port(device, PORT_C, CurrentState);

	// Write MSB of NGC command string to Data Bus
	hiword = LOWORD(data);
	hibyte = HIBYTE(hiword);
	lobyte = LOBYTE(hiword);
	status = DIG_Out_Port(device, PORT_A, (int)lobyte);
	status = DIG_Out_Port(device, PORT_B, (int)hibyte);

	// Take the STROBE line high to LATCH in the MSB data
	CurrentState = CurrentState | 0x80;
	status = DIG_Out_Port(device, PORT_C, CurrentState);

	// Take the STROBE line low and set A0 Low and A1 High
	CurrentState = 0x00;
	status = DIG_Out_Port(device, PORT_C, CurrentState);

	// Write Control word of NGC command string to data bus
	hibyte = HIBYTE(commandword);
	lobyte = LOBYTE(commandword);
	status = DIG_Out_Port(device, PORT_A, (int)lobyte);
	status = DIG_Out_Port(device, PORT_B, (int)hibyte);

	// Take the STROBE line high to LATCH in the data
	CurrentState = CurrentState | 0x80;
	status = DIG_Out_Port(device, PORT_C, CurrentState);

	// Configure Ports A & B as inputs
	status = DIG_Prt_Config(device, PORT_A, NOHANDSHAKE, INPUT);
	status = DIG_Prt_Config(device, PORT_B, NOHANDSHAKE, INPUT);

	return 0;
}
/* End NIDAQ_DIO_Transfer_Write */


/*
/     Read data and commands from a NPS3*** series controller
/     via PC-DIO-24 interface using NI_DAQ functions
*/
int NIDAQ_DIO_Transfer_Read(int device, long *data, int *commandword)
{
	int status, t;
	int linestate;
	int patternA = 0;
	int patternB = 0;
	int hiword;
	int loword;
	char lobyte;
	char hibyte;
	char CurrentState;

	// Check that function pointers have been initialised
	if(NIDAQ_Init == FALSE)
	 { return -1; }

	// Take the RWBAR line High and set A1 High, A0 Low and Strobe Low
	CurrentState = 0x60;
	status = DIG_Out_Port(device, PORT_C, CurrentState);

	// Check the BUSY-WRITE status and wait until it is low
	// If busy write status is high wait for 10 microseconds
	linestate = 2;
	for(t = 0; t < 50; t++)
	 {
		status = DIG_In_Port(device, PORT_A, &linestate);
```

```
		if((linestate & 0x02) == 0x02)
			ExactMicroDelay(10);
		else
			t = 50;
	 }

	// Take the STROBE line high
	CurrentState = CurrentState | 0x80;
	status = DIG_Out_Port(device, PORT_C, CurrentState);

	status = DIG_In_Port(device, PORT_A, (int)&lobyte);
	status = DIG_In_Port(device, PORT_B, (int)&hibyte);
	loword = MAKEWORD(lobyte,hibyte);

	// Read the MSB from Address 1
	// Set A0 to high and A1 to low ...... keep RWbar high
	CurrentState = 0xd0;
	status = DIG_Out_Port(device, PORT_C, CurrentState);

	status = DIG_In_Port(device, PORT_A, (int)&lobyte);
	status = DIG_In_Port(device, PORT_B, (int)&hibyte);
	hiword = MAKEWORD(lobyte,hibyte);

	*data = MAKELONG(loword,hiword);

	// Read the Command word from Address 0
	// Set A0 to low  and A1 to low ..... kepp RWbar high
	CurrentState = 0xc0;
	status = DIG_Out_Port(device, PORT_C, CurrentState);

	status = DIG_In_Port(device, PORT_A, (int)&lobyte);
	status = DIG_In_Port(device, PORT_B, (int)&hibyte);
	*commandword = MAKEWORD(lobyte,hibyte);

	return 0;
}
/* End NIDAQ_DIO_Transfer_Read */

/*
/     Fetch the address of NI_DAQ functions from National Instruments
/     dll nidaq32.dll
*/
int Init_NIDAQ_Functions(void)
{
	#define NIDAQDLL	"nidaq32.dll"
	#define DEVICEINFO	"Get_DAQ_Device_Info"
	#define PORTCONFIG	"DIG_Prt_Config"
	#define INPORT		"DIG_In_Port"
	#define OUTPORT			"DIG_Out_Port"
	#define INLINE		"DIG_In_Line"
	#define OUTLINE			"DIG_Out_Line"

	HINSTANCE hLib;

	hLib = LoadLibrary(NIDAQDLL);
	if(hLib != NULL)
	{

		(FARPROC)(DIG_Prt_Config) = (FARPROC) GetProcAddress(hLib,PORTCONFIG);
		if(DIG_Prt_Config == NULL) return -1;

		(FARPROC)(DIG_In_Port) = (FARPROC) GetProcAddress(hLib,INPORT);
		if(DIG_In_Port == NULL) return -1;

		(FARPROC)(DIG_Out_Port) = (FARPROC) GetProcAddress(hLib,OUTPORT);
		if(DIG_In_Port == NULL) return -1;

		(FARPROC)(DIG_In_Line) = (FARPROC) GetProcAddress(hLib,INLINE);
		if(DIG_In_Port == NULL) return -1;

		(FARPROC)(DIG_Out_Line) = (FARPROC) GetProcAddress(hLib,OUTLINE);
		if(DIG_In_Port == NULL) return -1;
	}
}
```

9.5 PAR-A Interface Performance

The performance of the NPS-PAR-A and B interfaces has been benchmarked using the NanoControl Panel software provided by Queensgate using a 400MHz Pentium II computer. The number of write / read cycles per second was measured during the “snapshot” mode of operation which downloads large quantities of Data from the NPS3000 series Digital Controller.

Table 9.9: NPS-PAR-A Interface Write/Read Performance

Operating system	**Write / Read cycles per second**
Windows 95 (PC-DIO24-JSM & PC-DIO24-PnP)	4000
Windows 95 (PCI6503)	1150
Windows NT	1150

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

10 NPS-PAR-B Fast Parallel Interface Card

10.1 NPS-PAR-B Device Description

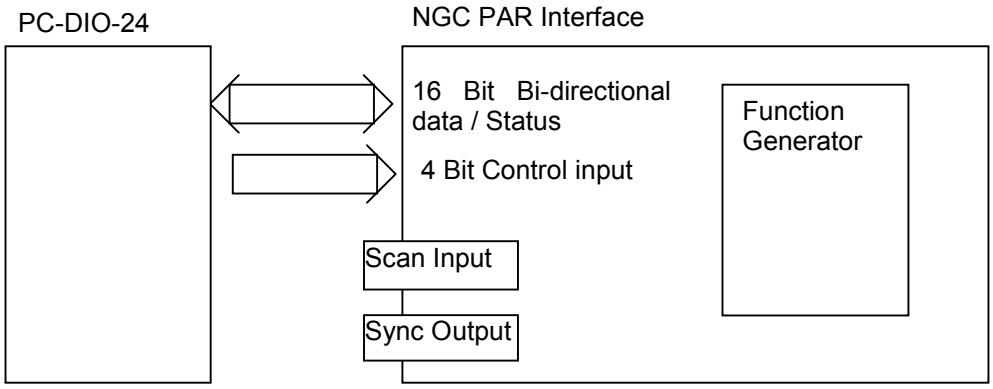
The NPS-PAR-B interface provides a high-speed parallel link between the controller and the host computer system, with the additional functionality of signal generator mode operation. The NPS-PAR-B is an extension of the NPS-PAR-A standard parallel interface. The NPS-PAR-B interface takes amplitude, offset, period and cycle information from the user and then automatically drives the mechanism with the selected waveform without additional user input. The NPS-PAR-B is supplied with NanoControl Panel software which has the function generator capability built in.

The NPS-PAR-B is specifically designed to interface with the National Instruments™ PC-DIO-24 card, 24-line digital input output card. Queensgate strongly recommend that anyone that wishes to use this interface purchases this card from National Instruments™. The NPS-PAR-B is supplied with cables and software designed specifically for communication between the NPS-PAR-B and PC-DIO-24 card. Technical details concerning the operation of the NPS-PAR-B parallel interface and function generator mode are detailed below.

If it is desired to change the scan parameters then the ramp signal must be stopped first. Any other command sent during the scan will be processed by the controller in the normal way.

A SYNC output signal is provided which provides two 1µs pulses per waveform cycle.
A SCAN input signal provides a facility for starting and stopping the function generator waveforms from an external control source.

Figure 10.1: NPS-PAR-B Block Diagram



The PC-DIO-24 card pin allocation is as follows.

Table 10.2: PC-DIO-24 Pin Connections

Pin Number	Function	Direction
Port A (PA0-PA7)	**Lower byte for 16-bit word / Status**	Bi-Directional
Port B (PB0-PB7)	**Upper byte for 16-bit word**	Bi-Directional
PC0	Not Used	PC-DIO-24 output to NPS-PAR-B
PC1	Not Used	PC-DIO-24 output to NPS-PAR-B
PC2	Not Used	PC-DIO-24 output to NPS-PAR-B

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

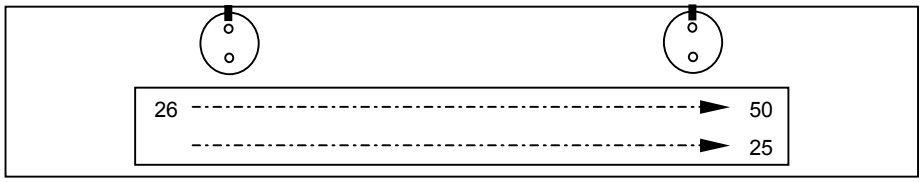
Pin Number	Function	Direction
PC3	Not Used	PC-DIO-24 output to NPS-PAR-B
PC4	**A0 register address line**	PC-DIO-24 output to NPS-PAR-B
PC5	**A1 register address line**	PC-DIO-24 output to NPS-PAR-B
PC6	**R/WBAR Control bit**	PC-DIO-24 output to NPS-PAR-B
PC7	**STROBE Control bit**	PC-DIO-24 output to NPS-PAR-B

10.2 Input / Output Connectors

Three connectors form the external interfacing to the NPS-PAR-B:

1. 50-way high-density connector for parallel interface connection to the PC-DIO-24 PC based card. The interface cable is provided.
2. A 2-way Lemo (00) socket (LEMO Part Number EGG-00-302-CNL) for the SCAN input. The input is an ANSI EIA/TIA 422 compatible differential line receiver.
3. A 2-way Lemo (00) socket (LEMO Part Number EGG-00-302-CNL) for the SYNC output. The output is an ANSI EIA/TIA 422 compatible differential line driver.

Figure 9.2: NPS-PAR-B Interface connections



Note that for the SYNC and Scan connectors the +ve connection is uppermost.

Table 10.3: NPS-PAR-B Parallel Interface Connections

Pin Number	Function	Pin Number	Function
1	**Strobe**	2	Not Used
3	**R/W Bar Control Bit**	4	Not Used
5	**A1**	6	Not Used
7	**A0**	8	Not Used
9	Not Used	10	Not Used
11	Not Used	12	Not Used
13	Not Used	14	Not Used
15	Not Used	16	Not Used
17	**Data Line D15**	18	Not Used
19	**Data Line D14**	20	Not Used
21	**Data Line D13**	22	Not Used
23	**Data Line D12**	24	Not Used

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Pin Number	Function	Pin Number	Function
25	**Data Line D11**	26	Not Used
27	**Data Line D10**	28	Not Used
29	**Data Line D9**	30	Not Used
31	**Data Line D8**	32	Not Used
33	**Data Line D7**	34	Not Used
35	**Data Line D6**	36	Not Used
37	**Data Line D5**	38	Not Used
39	**Data Line D4**	40	Not Used
41	**Data Line D3**	42	Not Used
43	**Data Line D2**	44	Not Used
45	**Data Line D1 / BUSY-WRITE**	46	Not Used
47	**Data Line D0 / BUSY-READ**	48	**Digital Ground**
49	Not Used	50	**Digital Ground**

10.3 NPS-PAR-B Operation / Interface Inputs.

10.3.1 Setting up the desired Waveform.

To operate the constant speed interface the user must input the following information (This can be achieved via the RS232 interface, the NPS-PAR-B parallel interface or the DSP port interface).

Refer to Point 10.4 for additional information about the input types and maximum input values for the different function generator commands.

1. Waveform Scan Type (to be applied to each channel, if any)
2. Waveform Scan Amplitude (to be applied to each channel)
3. Waveform Scan Speed (to be applied to each channel)
4. Number of scan cycles

All of these parameters can be accessed using The NanoControl Panel Software (version 1V4 onwards) or by using the software commands contained in Point 9.4 of this document.

Further information on programming software for this interface can be found in Section 14, ‘Command Language’.

10.3.2 Starting and stopping the Signal.

Two methods of starting and stopping the ramp signal are available. These are the *Read/Set Function Generator Mode* command and the TTL SCAN input signal.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

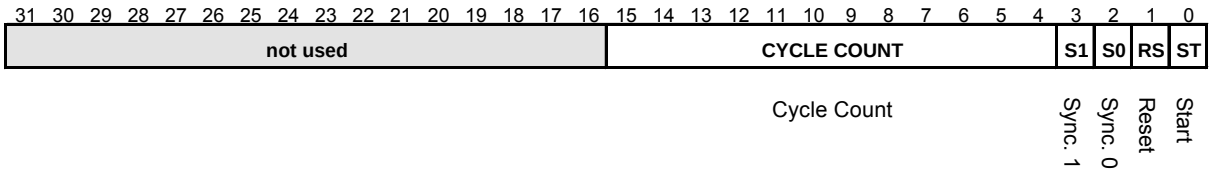
10.3.2.1 Read/Set Function Generator Mode

The *Set Function Generator Mode* has three different functions and is split into three different bit fields.

1. Starting and stopping the scanning
2. Setting the SYNC channel
3. Setting the number of scan cycles

The 32-bit data word is:

Figure 10.4: 32-bit Data Word



The *Read/Set Function Generator Mode* operation is the same as any other system command. Further details of this can be found in the NPS3330 Digital Controller User Manual on pages 34-38. These commands can be sent via the RS232, DSP port or parallel interfaces.

If the interface is started scanning using the *Set Function Generator Mode (Start)* command the system will continue scanning until the number of scan cycles is completed or a *Set Function Generator Mode (Stop)* command is sent.

10.3.2.2 Differential Line receiver SCAN Input

The ANSI EIA / TIA-422-B compatible SCAN input allows the user to start and stop the interface scanning without sending digital commands to the system, this allows the operation of the system without a host computer once the waveform is initially set up.

When the +ve SCAN input signal is greater than 0.2V above the –ve SCAN input signal the interface will start (I) sending waveform signal commands to the system. When the +ve SCAN input signal is greater than 0.2V below the –ve SCAN input signal the interface will stop (II) sending the waveform commands. If it is intended to use this method to start and stop the system scanning the number of scan cycles must be set to 0 (continuous scan).

If a system is stopped mid scan using the SCAN signal then when the scan is restarted it will begin from the point where it was stopped.

I. If the SCAN input is set to start the scanning, sending a *Set Function Generator Mode (Stop)* command will not stop the scan. The only way to stop the scan is to switch the SCAN signal to the STOP state.
II. If the *Set Function Generator Mode (Start)* command is used to start the scanning then setting the SCAN input to the STOP state will not stop the scanning.

10.3.3 Differential Line Driver SYNC Output

The NPS-PAR-B provides an ANSI EIA / TIA-422-B compatible differential line driver output signal of approximately 1µs width at the turning points of the periodic waveform. The position of the SYNC signal on the user defined waveform is at the points where the address is zero and 1024

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Decimal. There will be a small delay ~160µs after the SYNC signal is sent before the stage changes direction.

The channel, which generates the SYNC signal, can be selected using the *Set Function Generator Mode* command, see Section 10.4.1.1 below. Bits 2 & 3 of the command register select the SYNC channel.

10.3.4 Maximum Cable Length

The NPS-PAR-B SCAN and SYNC signals have line drivers and receivers that allow cables length of up to 10m to be used. The maximum cable length for the parallel interface connection is 2m (the parallel interface cable is supplied free of charge with the system).

10.4 Function Generator Mode.

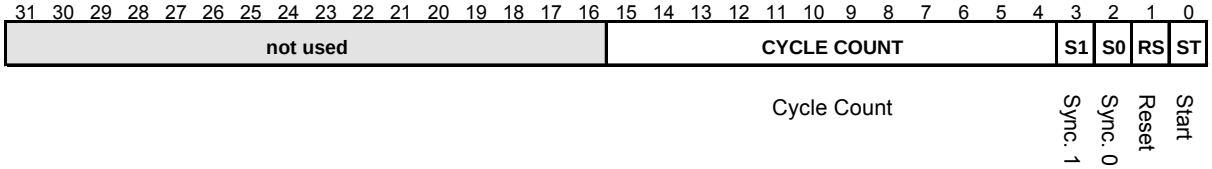
The operation of the NPS-PAR-B function generator mode is outlined below.

Different waveform types, scale factors, offsets and periods can be applied to each channel of the controller. When Set Waveform Type, Set Waveform Period, Set Waveform Scale Factor and Set Waveform Scale Factor Offset commands are issued they all need to be sent with a channel specifier as part of the command word. Further information on this is contained in the NPS3330 Digital Controller User Manual, Chapter 6 (Command language).

10.4.1 Function Generator Command Language Commands.

10.4.1.1 *Read / Set Function Generator Mode* (Command code 208 / 209 Decimal)

Figure 10.5: 32-bit Data Word



This command is used to start and stop scanning, set the number of cycles to scan and set the SYNC channel.

The number of scan cycles can be set to any integer value between 1 and 4095. If a continuous scan is required then the number of scan cycles should be set to 0. Any other value will return an error.

To use the cycle count facility set the function generator mode with the cycle count as shown. Taking the Start Bit low latches the value of cycle count into the interface. Taking the Start bit high will start the waveform. The cycle count applies to the channel selected by the SYNC bits, 1, 2 or 3.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Figure 10.6

S1	S0	SYNC Channel
0	0	No SYNC output
0	1	1
1	0	2
1	1	3

If the Reset Bit is commanded high the NPS controller automatically turns the signal into a High-Low-High pulse, which resets all of the waveforms on all channels to their start position. The Reset Bit will not latch high. The waveforms will automatically re-start unless the start bit has been set low.

10.4.1.2 Read / Set Waveform Type (210 / 211)

The NPS-PAR-B has four pre-programmed waveforms and one user* defined waveform which can be selected by the user for scanning SQueensgate NanoMechanisms.

The four pre-programmed waveforms are:

Figure 10.7: The 4 pre-programmed Waveform

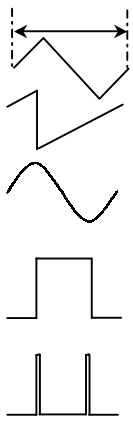
1 Ramp

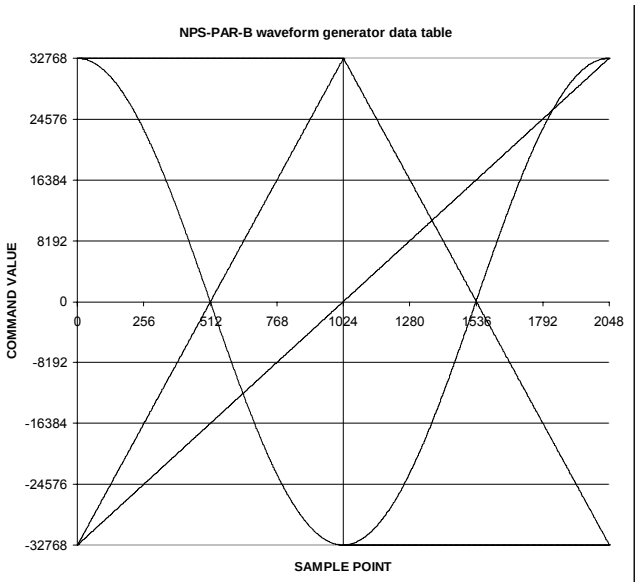
2 Saw tooth

3 Sine wave

4 Square wave

SYNC pulse





The waveforms are stored as 2048 16-Bit twos complement values as shown in the Waveform Generator table. The Sync Pulses are generated at sample point 0 and sample point 1024. When the waveforms are reset using the *Set Function Generator Mode command* the sample point is reset to zero. When the start signal is sent the waveforms will start from the value at sample point zero.

*This can be configured into the desired wave shape by the customer and then saved to memory and used as a fifth waveform.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

The valid waveform settings are listed below. Any other waveform setting will generate a *Waveform Type Invalid* error.

Figure 10.8: Waveform Types

Waveform Type		Description
(Dec)	(Hex)	
0	00	<no waveform>
1	01	Ramp
2	02	Saw-Tooth
3	03	Sine
4	04	Square
16	10	User-defined

10.4.1.3 *Read / Set Waveform Period* (212 / 213)

This is a floating-point number in the range $81.93e^{-6}$ to 5.36 representing the waveform period in seconds. Any other waveform period setting will generate a *Waveform Period Invalid* error.

For a given scan period the maximum scan amplitude that can be achieved will be limited by the range of the actuator and the maximum speed of the actuator. Furthermore the maximum scan amplitude will be limited by the stiffness of the mechanical system and the dynamic configuration of the system (servo loop setting).

10.4.1.4 *Read / Set Waveform Scale Factor* (214 / 215)

The default value for the Waveform Scale Factor is 1.000. The Waveform Scale Factor is a dimensionless ratio. Adjusting the waveform scale factor adjusts the amplitude of the waveform motion. For a desired pk-pk waveform motion set the Waveform Scale Factor as follows:

Motion in µm pk-pk = 2^{16} x Sensor Scale Factor x Waveform Scale Factor
Motion in units pk-pk = Motion in µm pk-pk / Unit Conversion Factor

The units can be selected as Microns, Nanometres or user defined units. To set the waveform scan peak-to-peak amplitude to the required user amplitude the Waveform Scale Factor should be calculated.

$$\text{Waveform Scale Factor} = \frac{\text{(Unit Conversion Factor x Desired Motion)}}{(2^{16} \text{ x Sensor Scale Factor})}$$

It is the Waveform_Scale_Factor value that must be sent to the interface and not the required position amplitude.

The NanoControl Panel software supplied with the controller has the function generator capability built into the 'Controller' menu. The NanoControl panel software has been programmed to take the amplitude value in user units (micrometers, nanometres or custom units) and convert to the Waveform Scale Factor, which is then commanded to the interface.

10.4.1.5 *Read / Set Waveform Scale Offset* (216 / 217)

The default value for the Waveform Scale Offset is zero. The valid range is from –32768 to +32767. To apply a position offset to the system the Waveform Scale Offset must be calculated.

Offset in µm = Sensor Scale Factor x Waveform Scale Offset
Offset in units = Offset in µm / Unit Conversion Factor

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

To set the waveform offset to the required user Offset the Waveform Scale Offset should be calculated.

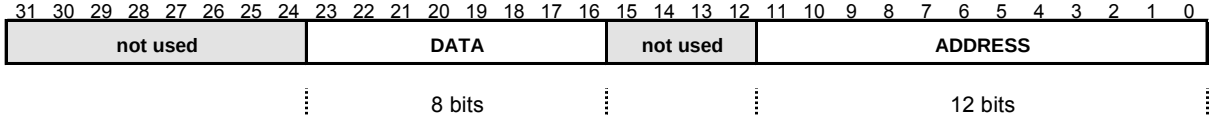
$$\text{Waveform Scale Offset} = \frac{\text{(Unit Conversion Factor x Required Offset)}}{\text{(Sensor Scale Factor)}}$$

In a similar way to the Waveform Scale Factor value, the Waveform Scale Offset is the value that must be commanded to the system, rather than the required position offset in micrometers or custom units. The Waveform Scale Offset is in System Units.
The NanoControl panel software has been programmed to take the offset value in user units (micrometers, nanometres or custom units) and convert to the Waveform_Scale_Offset, which is then commanded to the interface.

10.4.1.6 *Read / Set Waveform Data* (218 /219)

Read: This command is used to check the user-defined waveform data stored in the NPS-PAR-B interface. Waveforms are defined in the form of a look-up table, requiring an address field and a data field for each point. The two fields are coded into the data word as illustrated below.

Figure 10.9: 32-bit Data Word



The user waveform must have 2048 points per cycle. The data must be coded in 16Bit two's complement format. To program the waveform into the interface the 2048 16-Bit values must be split into 4096 8-Bit values. The even addresses must contain the lower byte of the 16-Bit word and the odd addresses must contain the upper byte.

Note: The Data field is not used when sending the Read Waveform Data command to the controller.

Set: This command is used to set the user defined waveform data stored in the NPS-PAR-B interface. The NanoControl Panel software has a feature, which allows a user to set this information using an Microsoft Excel file. Contact Queensgate Instruments Ltd for details.

10.4.1.7 Store Waveform Data (221)

This command is used to store the user-defined waveform data into the NPS-PAR-B non-volatile RAM. Once the command to store the waveform has been given the waveform will be re-stored if the unit is switched off and on again. If the command is not given the waveform will be lost when the unit is switched off. Contact Queensgate Instruments Ltd for details.

10.5 Function Generator Mode Operation.

The following section is intended to provide guidance on the operation of the NPS-PAR-B with and without SCAN input.

10.5.1 Operation sequence using the SCAN input

1. Select the Waveform Type to be applied to each channel, via RS232, Parallel or DSP Port interface.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

2. Set the Waveform Period to be applied to each channel.
3. Set the Waveform_Scale_Factor and Waveform_Scale_Offset for each selected channel.
4. Set the "Number of scan cycles" to zero.
5. Switch the SCAN input to the high level (+ve input > 200mV above the –ve input).
6. Monitor the output SYNC pulses until the desired number of scan cycles is completed.
7. Switch the SCAN input to the low level (+ve input > 200mV below the –ve input) to stop the scanning.

To abort a scan; switch the SCAN input to the low level.

If the SCAN level is set high again then the interface will resume scanning from the same point that it stopped scanning.

To reset the scan to the start of the waveform, send a *Set Function Generator Mode (Reset)* command via RS232, Parallel interface or the DSP port interface.

10.5.2 Operation sequence using the *Set Function generator Mode* command

1. Select the Waveform Type to be applied to be applied to each channel, via RS232 or Parallel or DSP Port interface.
2. Set the Waveform Period to be applied to each channel.
3. Set the Waveform_Scale_Factor and Waveform_Scale_Offset for each selected channel.
4. Set the "Number of scan cycles" to zero or the desired number of cycles.
5. Send a *Set Function Generator Mode (Start)* command to start the scanning.
6. Monitor the output TTL pulses until the desired number scan cycles is completed.
7. Send a *Set Function Generator Mode (Stop)* command to stop the scanning.

To abort a scan; send a *Set Function Generator Mode (Stop)* command. If the *Set Function Generator Mode (Start)* command is sent to the controller then the interface will resume scanning from the same point that it stopped scanning. To reset the scan to the start, send a *Set Function Generator Mode (Reset)* command via RS232 interface, the parallel interface, or the DSP port interface.

10.5.3 Changing Waveform Parameters

If it is desired to change the scan parameters then the waveform signal must be stopped first.

10.6 NPS-PAR-B Parallel Interface Operation

10.6.1 Command String

The parallel interface uses a command string of three 16-bit words to communicate. The string consists of a 32-bit data word (made from 2 x 16-bit words) and a 16-bit command word. Refer to Section 10.7 for further information on how to program for the NPS-PAR-A/B interface.

The string is sent in that order LSWord, MSWord, command word. The data and command words are sent using the digital lines (D15 – D0) in the NPS-PAR-B interface connections table, table 10.2.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Four control lines and two Status bits, BUSY-READ & BUSY-WRITE control handshaking between the computer and PAR-A interface. The status bits are multiplexed onto the bottom two bits of the data bus.

The NPS3330 Command Language Reference, NPS-3023-M, contains a full list of the command words for the NPS3330. The CRLF termination is optional.

10.6.1.1 Write Cycle

A transfer from the computer to the NGC PAR interface is illustrated in fig 3.1 below. Initially the R/WBAR control line is high. In this state the PAR interface drives the 16-bit data bus and the PC-DIO-24 must have these lines (Port A & Port B) configured as inputs.

With R/WBAR high the STROBE Control line selects between Data and Status. With STROBE high Data is driven onto the 16 Bit bus as selected by the A1, A0 address lines. With STROBE low the bottom two bits of the 16 Bit data bus are driven by two status bits, BUSY-READ (D0) and BUSY-WRITE (D1).

A. At point A in fig 3-1 the STROBE line goes low with R/WBAR high. The status bits are driven onto the data bus. The computer polls the BUSY-READ status line (D0) until it detects that it is low. A low level on BUSY-Read indicates that the PAR interface is ready to receive the next command from the computer.

B. At point B the computer takes the R/W Bar line Low to start the write cycle from computer to PAR interface. At this point the 16-bit data bus becomes tri-state, as it is not being driven from the computer or the PAR interface. The computer drives the Address lines to address 2 prior to the 1st data transfer, the least significant word of the 32-Bit data word (LSW). Note that the CRLF terminator at address 3 is optional.

C. The computer re-configures the 16-bit data bus as outputs and writes the 16 bit data to the ports.

D. The STROBE Line is driven high then low. This Latches the Data into the PAR interface.

D to E – The computer sets up the remaining address and data and strobe most significant word (MSW) and command string into the PAR interface. The command string with address A1, A0 = 0 must be the last word written. When A1, A0 = 0 the rising edge of STROBE causes BUSY-READ (and BUSY-WRITE) to be set high.

E. The computer re-configures the 16 Bit data bus as inputs.

F. The computer drives the R/WBAR line high. With STROBE low the status Bits, BUSY-READ and BUSY-WRITE, are driven onto the bottom two bits of the data bus for the computer to poll.

Figure 10.10: Computer Write Cycle to PAR Interface

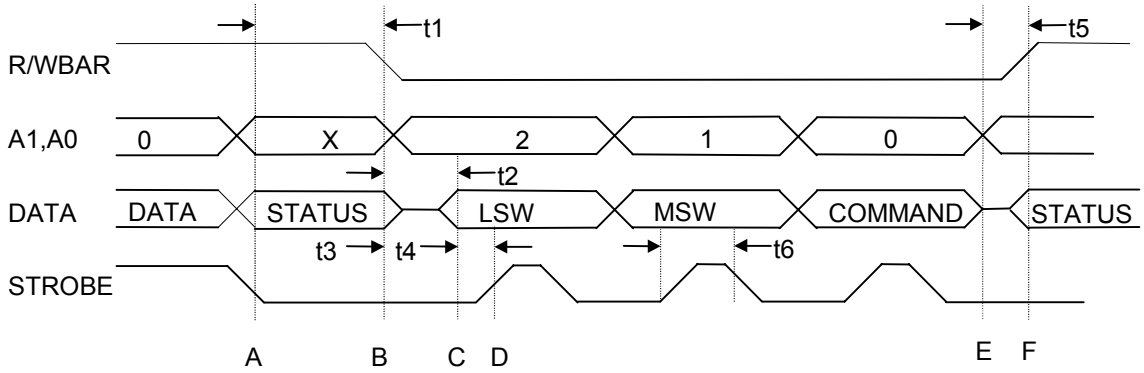


Table 10.11: Time t_x Description of Figure 10.10

No	Description	Min	Max	Unit
t_1	Delay time, BUSY-READ low to R/WBAR low	0		ns
t_2	Set up time, R/WBAR low before DATA lines driven	40		ns
t_3	Set up time, Address stable before STROBE high	55		ns
t_4	Set up Time, DATA stable before STROBE high	55		ns
t_5	Delay time, R/WBAR high after DATA High Z	20		ns
t_6	Cycle time, STROBE high	160		ns

10.6.1.2 Read Cycle

A transfer from the NGC PAR interface to the computer is illustrated in figure 9.4 below. The R/WBAR control line is high. In this state the PAR interface drives the 16-bit data bus and the PC-DIO-24 must have these lines configured as inputs.

With R/WBAR high the STROBE Control line selects between Data and Status. With STROBE high Data is driven onto the 16 Bit bus as selected by the A1, A0 address lines. With STROBE low the bottom two bits of the 16 Bit data bus are driven by two status bits, BUSY-READ (D0) and BUSY-WRITE (D1).

A. At point A in figure 3-2 the STROBE line is low with R/WBAR high. The status bits are driven onto the data bus. The computer polls the BUSY-WRITE (D1) status line until it detects that it is low, indicating that the NPS controller has finished writing data to the PAR interface, and that the computer is allowed to read it.

B. At point B the computer takes the STROBE line high. Data rather than status is driven onto the 16 Bit bus. The data is selected by the A1, A0 address lines.

C. The computer changes the address and the corresponding data appears on the data bus.

D. The computer changes the address and the corresponding data appears on the data bus. The data string can be read in any order. Reading the CRLF terminator is optional.

Figure 10.12: Computer Read Cycle from PAR interface

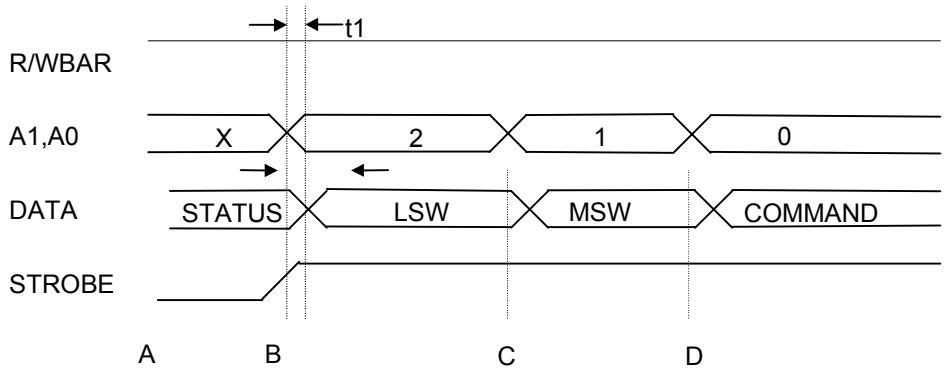


Table 10.13: Time t_x Description of Figure 10.12

No	Description	Min	Max	Unit
t1	Delay time, DATA valid after Address stable		55	ns

Notes: 1/ The falling edge of BUSY-WRITE (D1 status bit) indicates that the NPS controller has written data in response to a command, and that the data is available to the PC.
2/ The CRLF terminator is not necessary.
3/ The address value indicates the value of the control lines A1and A0. Where A0 is the LSB.
4/ The data string can be read in any order.
5/ BUSY-WRITE is set high after a computer command string write cycle.

10.7 Sample Software

Please Note: This source code requires NI-DAQ Diver Software installed on the target machine for operation under Windows 95 and NT.

```
/*          *** CODE FRAGMENT ***
/
/           PURPOSE: TO DESCRIBE A METHOD TO TRANSFER DATA AND COMMANDS
/           WITH THE WIN NT COMPATIBLE PAR INTERFACE (QI) AND AN NPS3***
/           CONTROLLER (QI) IN VISUAL C++ USING NI-DAQ DRIVER SOFTWARE
*/

/*
/           NGC Parallel Data Transfer (Write and Read Methods)
/           to support New Style PAR Interface for operation
/           using NI-DAQ Driver Software on Windows 95 and Windows NT
*/

#include <windows.h>
#include <stdio.h>
#include "dio24.h"
#include "nidaqcns.h"       /* National instruments header */
#include "utils.h"

#define LOW          0
#define HIGH                1
#define NOHANDSHAKE  0
#define HANDSHAKE           1
#define INPUT               0
#define OUTPUT                      1
#define BIDIRECTIONAL2

#define BUSY_READ           0
#define BUSY_WRITE          1
#define SYNC                2
#define A0_REGISTER         4
#define A1_REGISTER         5
#define RWBAR               6
#define STROBE                      7

static FARPROC Get_DAQ_Device_Info;
static FARPROC DIG_Prt_Config;
static FARPROC DIG_In_Port;
static FARPROC DIG_Out_Port;
static FARPROC DIG_In_Line;
static FARPROC DIG_Out_Line;

static BOOL NIDAQ_Init = FALSE;     // True when pointers to NI-DAQ functions  initialised
```

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
/*
/           Write data and commands to a NPS3*** series controller
/           via PC-DIO-24 interface using NI_DAQ functions.
*/
int NIDAQ_DIO_Transfer_Write(int device, long data, int commandword, long *result)
{
            int status, t;
            int linestate;
            int patternA = 0;
            int patternB = 0;
            int hiword;
            int loword;
            char lobyte;
            char hibyte;
            char CurrentState;

            // If first time round initialise function pointers
            if (NIDAQ_Init == FALSE)
            {
              Init_NIDAQ_Functions();
              NIDAQ_Init = TRUE;
            }

            // Configure Ports A & B as inputs and Port C as output
            status = DIG_Prt_Config(device, PORT_A, NOHANDSHAKE, INPUT);
            status = DIG_Prt_Config(device, PORT_B, NOHANDSHAKE, INPUT);
            status = DIG_Prt_Config(device, PORT_C, NOHANDSHAKE, OUTPUT);

            // Take the RWBAR line High
            //status = DIG_Out_Line(device, PORT_C, RWBAR, HIGH);
            CurrentState = 0x40;
            status = DIG_Out_Port(device, PORT_C, CurrentState);

            // If BUSY-READ Status is high wait for 10 microseconds
            linestate = 0;
            for(t = 0; t < 50; t++)
            {
              status = DIG_In_Port(device, PORT_A, &linestate);

              if((linestate & 0x01) == 0x01)
                      ExactMicroDelay(10);
              else
                      t = 50;
            }


            // Reconfigure the PC-DIO-24 data lines
            status = DIG_Prt_Config(device, PORT_A, NOHANDSHAKE, OUTPUT);
            status = DIG_Prt_Config(device, PORT_B, NOHANDSHAKE, OUTPUT);

            // Set A0 Low and A1 High
            CurrentState = 0x20;
            status = DIG_Out_Port(device, PORT_C, CurrentState);


            // Write LSB of NGC command string to Data Bus
            loword = HIWORD(data);
            hibyte = HIBYTE(loword);
            lobyte = LOBYTE(loword);
            status = DIG_Out_Port(device, PORT_A, (int)lobyte);
            status = DIG_Out_Port(device, PORT_B, (int)hibyte);

            // Take the STROBE line high to LATCH in the LSB data
            CurrentState = CurrentState | 0x80;
            status = DIG_Out_Port(device, PORT_C, CurrentState);

            // Take the STROBE line low and set A0 High and A1 Low
            CurrentState = 0x10;
            status = DIG_Out_Port(device, PORT_C, CurrentState);

            // Write MSB of NGC command string to Data Bus
            hiword = LOWORD(data);
            hibyte = HIBYTE(hiword);
            lobyte = LOBYTE(hiword);
            status = DIG_Out_Port(device, PORT_A, (int)lobyte);
            status = DIG_Out_Port(device, PORT_B, (int)hibyte);
```

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
            // Take the STROBE line high to LATCH in the MSB data
            CurrentState = CurrentState | 0x80;
            status = DIG_Out_Port(device, PORT_C, CurrentState);

            // Take the STROBE line low and set A0 Low and A1 High
            CurrentState = 0x00;
            status = DIG_Out_Port(device, PORT_C, CurrentState);

            // Write Control word of NGC command string to data bus
            hibyte = HIBYTE(commandword);
            lobyte = LOBYTE(commandword);
            status = DIG_Out_Port(device, PORT_A, (int)lobyte);
            status = DIG_Out_Port(device, PORT_B, (int)hibyte);

            // Take the STROBE line high to LATCH in the data
            CurrentState = CurrentState | 0x80;
            status = DIG_Out_Port(device, PORT_C, CurrentState);

            // Configure Ports A & B as inputs
            status = DIG_Prt_Config(device, PORT_A, NOHANDSHAKE, INPUT);
            status = DIG_Prt_Config(device, PORT_B, NOHANDSHAKE, INPUT);

            return 0;
}
/* End NIDAQ_DIO_Transfer_Write */
```

```
/*
/           Read data and commands from a NPS3*** series controller
/           via PC-DIO-24 interface using NI_DAQ functions
*/
int NIDAQ_DIO_Transfer_Read(int device, long *data, int *commandword)
{
            int status, t;
            int linestate;
            int patternA = 0;
            int patternB = 0;
            int hiword;
            int loword;
            char lobyte;
            char hibyte;
            char CurrentState;

            // Check that function pointers have been initialised
            if(NIDAQ_Init == FALSE)
             { return -1; }

            // Take the RWBAR line High and set A1 High, A0 Low and Strobe Low
            CurrentState = 0x60;
            status = DIG_Out_Port(device, PORT_C, CurrentState);

            // Check the BUSY-WRITE status and wait until it is low
            // If busy write status is high wait for 10 microseconds
            linestate = 2;
            for(t = 0; t < 50; t++)
             {
              status = DIG_In_Port(device, PORT_A, &linestate);

              if((linestate & 0x02) == 0x02)
                     ExactMicroDelay(10);
              else
                     t = 50;
             }

            // Take the STROBE line high
            CurrentState = CurrentState | 0x80;
            status = DIG_Out_Port(device, PORT_C, CurrentState);

            status = DIG_In_Port(device, PORT_A, (int)&lobyte);
            status = DIG_In_Port(device, PORT_B, (int)&hibyte);
            loword = MAKEWORD(lobyte,hibyte);

            // Read the MSB from Address 1
            // Set A0 to high and A1 to low ...... keep RWbar high
            CurrentState = 0xd0;
            status = DIG_Out_Port(device, PORT_C, CurrentState);

            status = DIG_In_Port(device, PORT_A, (int)&lobyte);
            status = DIG_In_Port(device, PORT_B, (int)&hibyte);
            hiword = MAKEWORD(lobyte,hibyte);

            *data = MAKELONG(loword,hiword);

            // Read the Command word from Address 0
            // Set A0 to low  and A1 to low ..... kepp RWbar high
            CurrentState = 0xc0;
            status = DIG_Out_Port(device, PORT_C, CurrentState);

            status = DIG_In_Port(device, PORT_A, (int)&lobyte);
            status = DIG_In_Port(device, PORT_B, (int)&hibyte);
            *commandword = MAKEWORD(lobyte,hibyte);

            return 0;
}
/* End NIDAQ_DIO_Transfer_Read */
```

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
/*
/           Fetch the address of NI_DAQ functions from National Instruments
/           dll nidaq32.dll
*/
int Init_NIDAQ_Functions(void)
{
            #define NIDAQDLL "nidaq32.dll"
            #define DEVICEINFO      "Get_DAQ_Device_Info"
            #define PORTCONFIG      "DIG_Prt_Config"
            #define INPORT          "DIG_In_Port"
            #define OUTPORT         "DIG_Out_Port"
            #define INLINE          "DIG_In_Line"
            #define OUTLINE         "DIG_Out_Line"

            HINSTANCE hLib;

            hLib = LoadLibrary(NIDAQDLL);
            if(hLib != NULL)
            {
              (FARPROC)(DIG_Prt_Config) = (FARPROC) GetProcAddress(hLib,PORTCONFIG);
              if(DIG_Prt_Config == NULL) return -1;

              (FARPROC)(DIG_In_Port) = (FARPROC) GetProcAddress(hLib,INPORT);
              if(DIG_In_Port == NULL) return -1;

              (FARPROC)(DIG_Out_Port) = (FARPROC) GetProcAddress(hLib,OUTPORT);
              if(DIG_In_Port == NULL) return -1;

              (FARPROC)(DIG_In_Line) = (FARPROC) GetProcAddress(hLib,INLINE);
              if(DIG_In_Port == NULL) return -1;

              (FARPROC)(DIG_Out_Line) = (FARPROC) GetProcAddress(hLib,OUTLINE);
              if(DIG_In_Port == NULL) return -1;
            }
}
```

10.8 PAR-B Interface Performance

The performance of the NPS-PAR-B interface has been benchmarked using the NanoControl Panel software provided by Queensgate using a 400MHz Pentium II computer. The number of write / read cycles per second was measured during the “snapshot” mode of operation which downloads large quantities of Data from the NPS3000-series Digital Controller.

Table 10.14

Operating System	**Write / Read cycles per second**
Windows 95 (PC-DIO24-JSM & PC-DIO24-PnP)	4000
Windows 95 (PCI6503)	1150
Windows NT	1150

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

11 NPS-PAR-C Fast Parallel Interface Card (ready limit output)

11.1 NPS-PAR-C Device Description.

The NPS-PAR-C interface provides a high-speed parallel command link between the controller and the host system with the added functionality of high-speed READY status update.

The NPS-PAR-C allows 16-bit parallel data input for ALL three channels of an NPS3000-series controller with ready status update for all three channels. NPS-PAR-C commands are summed with commands from the RS232 port AND the DSP Port. If there is a command from the RS232 or the DSP port interfaces, they will act as an offset to commands from the NPS-PAR-C.

11.2 Input / Output Connectors.

The NPS-PAR-C has a 25-way D-type socket connector for interfacing to the User. Each of the NPS-PAR-C input and output lines are buffered using a 74LS245 TTL buffer.

11.3 Operation of the NPS-PAR-C

11.3.1 User Interface connector pin out (NPS3000-series rear panel)

Table 11.1: 25-way D-type connector pin out

25 way D-type pin No.	Signal Name	INPUT / OUTPUT
1	0V (digital)	----------
2	Ready Channel 3	OUTPUT
3	Ready Channel 1	OUTPUT
4	A1 (address bit 1)	INPUT
5	WR (Data valid)	INPUT
6	D14	INPUT
7	D12	INPUT
8	D10	INPUT
9	D8	INPUT
10	D6	INPUT
11	D4	INPUT
12	D2	INPUT
13	D0	INPUT
14	0V (digital)	----------
15	Ready Channel 2	OUTPUT
16	Busy Read	OUTPUT

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

25 way D-type pin No.	Signal Name	INPUT / OUTPUT
17	A0 (address bit 0)	INPUT
18	D15	INPUT
19	D13	INPUT
20	D11	INPUT
21	D9	INPUT
22	D7	INPUT
23	D5	INPUT
24	D3	INPUT
25	D1	INPUT

11.3.2 Command Word

The command word is a 16-bit TTL parallel input bus D0-D15. The data is coded in Binary Two's Complement format giving a command range of 8000 hex (-32768 decimal) to 7FFF hex (32767 decimal).

The command can be converted into micrometers using the Digital Input Scale Factor given in the system test results and displayed in the NanoControl Panel software under Analogue_Input_Factor in the "Edit parameters within controller" menu.

The command input bus D0-D15 is common for all three channels and the 2 Address lines A0 and A1 select which channel the command is applied to.

Table 11.2: Stage Position vs. Digital input Command

COMMAND VALUE	STAGE POSITION
7FFF hex	+(Scan range)/2
0000 hex	Nominal position
8000 hex	-(Scan range)/2

11.3.3 Channel Address (A0, A1) Decode

Address lines A0 and A1 routes the command to the correct channel as shown in table below.

Table 11.3: Channel Address decode

A1	A0	
0	0	Channel 1 command data
	1	Channel 2 command data
1	0	Channel 3 command data
1	1	End transmission

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

11.3.4 Command Write Cycle

Figure 11.4 shows the timing requirements for a command write sequence from the User interface to the NPS-PAR-C interface.

Figure 11.4: Timing Diagram for a Command write cycle from Interface to NPS-PAR-C

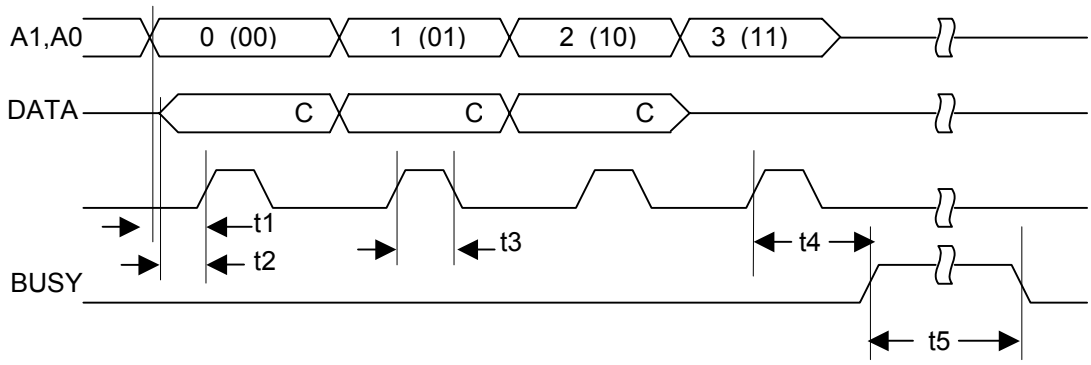


Table 11.5: Time t_x Description for Figure 11.4

Ref.	Description	Min	Max	Unit
t_1	Set up time, Address stable before WR high	55		ns
t_2	Set up Time, DATA stable before WR high	55		ns
t_3	Cycle time, WR high	160		ns
t_4	Delay time, BUSY high after WR high with A1=”1”,A0=”1”		100	ns
t_5	BUSY	120	240	µs

Notes:

1. The rising edge of WR “latches” the command data (D0-D15) into the registers indicated by the address bus (A1, A0). WR should remain low at the end of the write cycle.
2. The address A1=”1”, A0 = “1” should be the last address written. When A1=”1”, A0=“1” the rising edge of WR causes BUSY to be set high. When A1=”1”, A0=“1”, DATA bus state is DON’T CARE.
3. The falling edge of Busy indicates that the NPS3xx0 has read the command string data buffers and is ready for the next command string to be written by the User Interface. NEW commands can only be written when the BUSY line is LOW.
4. BUSY can be high for up to 240µs (t5). This is because the commands have to be synchronised with the DSP read cycle. If the “End Transmission” command is sent when the DSP read cycle starts, then the new commands have to “wait” a maximum of 120µs before the next DSP read cycle. The DSP read cycle takes 120µs. This synchronisation is done within the NPS-PAR-C.
5. If no data is written to the NPS-PAR-C when BUSY line is LOW, the command does NOT change.

NPS-PAR-C commands are summed with the commands via the RS232 AND the DSP Port. If there is a command from the RS232 or the DSP port interfaces, they will act as an offset to commands from the NPS-PAR-C.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

11.3.5 Command Sequence for writing to all 3 Channels (CH1, CH2 and CH3)

1. Set up address (A1="0", A0="0") and data for channel 1.
2. Take WR line HIGH and LOW.
3. Set up address (A1="0", A0="1") and data for channel 2.
4. Take WR line HIGH and LOW.
5. Set up address (A1="1", A0="0") and data for channel 3.
6. Take WR line HIGH and LOW.
7. Set up address for "End transmission" (A1="1" and A0="1").
8. Take WR line HIGH and LOW.
9. Wait for BUSY to go LOW, then repeat from (1) as necessary.

Data does NOT have to be written to all three channels.

The only requirement is that the WR must be taken HIGH when A1="1" and A0="1". This indicates to the interface that there are no more commands to be received. For example, if data is to be written to Channel 2 only, then set address and data for Channel 2, toggle WR. Then set A1 and A0 high and toggle WR. This will update Channel 2 data only.

11.3.6 Command sequence for writing to less than 3 channels

1. Set up address and data for the first channel.
2. Take WR line HIGH and LOW.
3. Set up address and data for next channel – otherwise go to (5).
4. Take WR line HIGH and LOW.
5. Set up address 3 (A1="1" and A0="1").
6. Take WR line HIGH and LOW.
7. Wait for BUSY to go LOW, then repeat from (1) as necessary.

11.3.7 Status signals from NPS-PAR-C

The NPS-PAR-C Status signals are:

- 1 off BUSY - when HIGH, indicates that the data written by the interface has NOT been read by the DSP.
- 3 off READY lines, one per channel – when HIGH, indicates that the stage has reached within the Commanded position ± READY limit. READY limit is set via the RS232 or the DSP PORT interfaces using the NPS3xx0 command language or the NanoControl Panel software in the Controller Edit Parameters within the Controller Menu.

These signals can be read at any time by the user Interface.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

12 NPS-SER-A Serial Interface

12.1 NPS-SER-A Device Description.

The NPS-SER-A interface provides a high-speed serial position monitor signal between the controller and the host system.

The NPS-SER-A outputs a 32-bit serial position monitor for CH1 (for NPS3110 controllers), CH1 & CH2 (for NPS3220 controllers), CH1, CH2 and CH3 (for NPS3330 controllers).

No user commands are required for the operation of this interface.

12.2 Input / Output Connectors

The NPS-SER-A has a 9-way D-type socket connector for interfacing to the User. Each of the NPS-SER-A output lines is buffered using a 75174 differential line driver (with a 100 ohm termination resistor).

12.3 Operation of the NPS-SER-A

12.3.1 Interface connector pin out (NPS3xx0 Rear panel).

Table 12.1: 9-way D-type connector pin out.

9 way D-type pin No.	Signal Name	INPUT / OUTPUT
1	Data CLOCK +	OUTPUT
2	Serial DATA +	OUTPUT
3	Sync./Enable signal +	OUTPUT
4	0V(D)	----------
5	0V(D)	----------
6	Data CLOCK -	OUTPUT
7	Serial DATA -	OUTPUT
8	Sync./Enable signal -	OUTPUT
9	0V(D)	----------

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

12.3.2 NPS-SER-A Timing Diagram

Figure 12.2: NPS-SER-A Timing Diagram

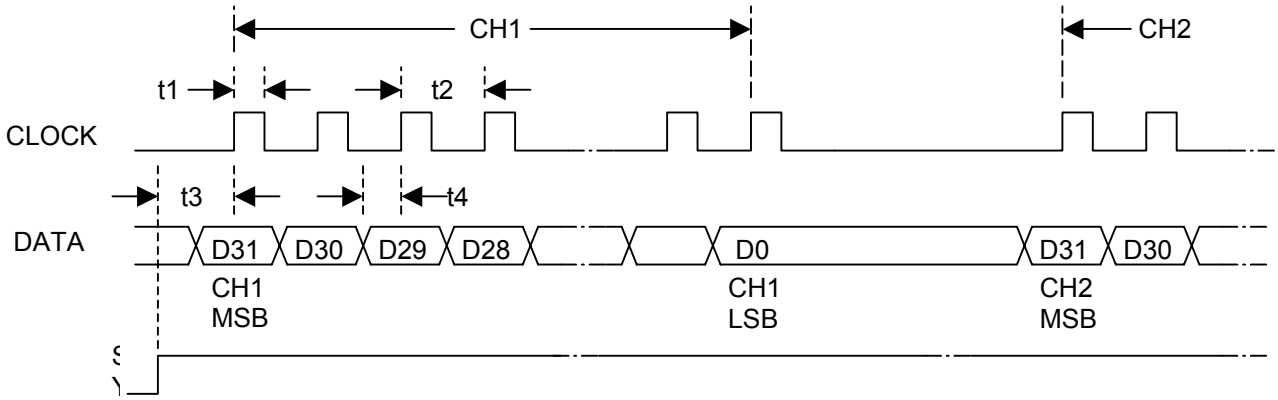


Figure 12.3: NPS-SER-A Timing Diagram

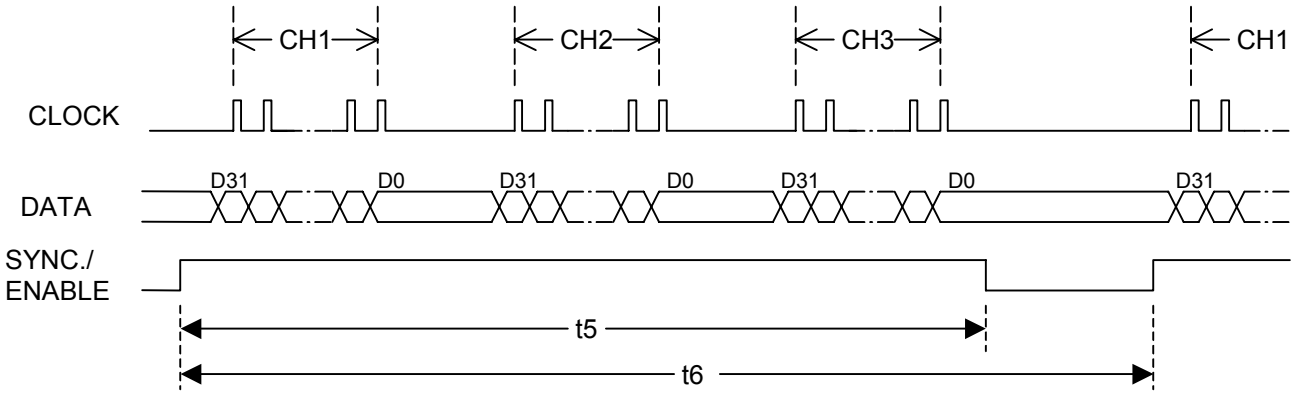


Table 12.4: Time t_x Description

Ref.	Description	Min	Max	Unit
t_1	Clock High time.	39	41	ns
t_2	Clock period.	273	287	ns
t_3	Set up time, Sync./Enable high before rising edge of first clock.	117	223	ns
t_4	Set up time, DATA stable before rising edge of clock.	187	197	ns
t_5	Sync./Enable pulse width.	29	31	µs
t_6	Sync./Enable period.	39	41	µs

Notes:

1. Data is valid on the rising edge (Low to high transition) of the clock.
2. First 32 bits of data after Sync./Enable going high is for Channel 1 (MSB, D31 first, then D30, etc to D0, LSB) followed by Channel 2 data then channel 3 data.
3. If an NPS3110 Digital Controller is used, then there will be 32 clock (and data bits), there will be no clock and data for channels 2 & 3. For an NPS3220, there will be 64 clock and data bits (32 for channel 1 and 32 for channel 2). For an NPS3330, there will be 96 clock and data bits (32 each for channels 1, 2 & 3).

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

12.4 NPS-SER-A Specification

The position data will be transmitted as a 32 bits, TI TMS320C3x format, floating point number for each channel.

Table 12.5

PARAMETER	VALUE	COMMENTS
NPS-SER-A INTERFACE TO USER		
Position Data Update time	40µs	Each channel position data will be updated at 40µs intervals.
Data Length	32, 64 or 96 bits	See Note 1.
Format	Texas Instruments TMS320C3x Floating Point	
Data Units	Queensgate System Units	See Note 2.
Interface type	Serial	
Data transfer rate	3.6MHz typical	See Note 3.
Number of lines	9	3 differential signals (Clock, Data, Sync./Enable) and 3 ground, 0V(D) lines.
Line driver	SN75174	Differential output with 100Ω terminator.
Interface connector	9-way D-type socket	Mounted on NPS3xx0 rear panel.

Notes:

1. Data length, i.e. number of bits transmitted per 40µs cycle time is 32 bits for an NPS3110, 64 bits for an NPS3220 and 96 bits for an NPS3330.
2. Position data will be in linearised System Units. The user will have to convert it from system units to micrometers using the Sensor Scale Factor.
3. Within the DSP firmware, channel data is processed in 10µs slots. This means that the position data has to be transferred within 10µs per channel giving a minimum transfer rate of 3.2MHz (32 bits in 10µs).

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

13 Custom Interfaces

For any other interfaces, please contact Queensgate Ltd (see Section 23).

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

14 Command Language

14.1 Introduction

This Chapter describes the command language used to control and monitor the NPS3110, NPS3220 and NPS3330 Digital Controllers. The command language has been designed to be independent of the communications interface used, though there are some minor differences in the way the commands are transmitted. Thus it is possible to send any command via the standard interfaces (RS232 and DSP Port) or via an optional plug-in interface (NPS-PAR parallel interface or custom user interface).

14.2 Overview

All communications between the Controller and computer are initiated by the computer transmitting a command sequence to the Controller. The command sequence may be either:

- A read command sequence in which the computer is requesting data from the Controller.

or

- A set command sequence in which the computer is writing data values to the Controller.

The Controller responds to a read command sequence by transmitting a response sequence to the computer containing the requested data.

14.3 Controller Commands

14.3.1 Command Sequence

Every command sequence comprises a:

- 16-bit command word
- 32-bit data word
- 16-bit terminator.

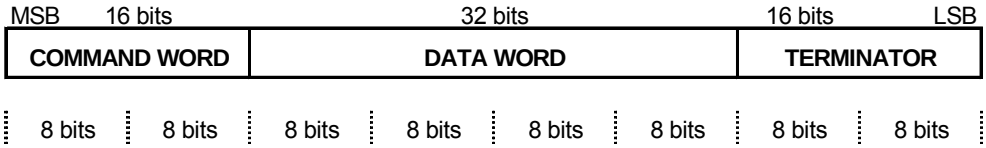
The manner in which these are transmitted depends on the interface used, as follows:

14.3.2 RS232 Serial Interface

The command sequence is transmitted beginning with the command word, followed by the data word and then the terminator. Since the RS232 interface is configured for 8-bit data transfers, the sequence is actually transmitted as eight bytes (1 byte = 8 bits), most significant byte first, as illustrated below.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

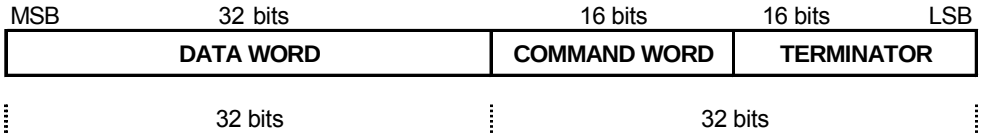
Figure 14.1: RS232 Command Sequence



14.3.3 DSP Port Serial Interface

Since the DSP Port interface uses 32-bit wide data transfers, the command sequence is transmitted in a slightly different order to maximize efficiency. Here the command word is transmitted after the data word. This allows the complete command sequence to be transmitted using two 32-bit words, as illustrated below.

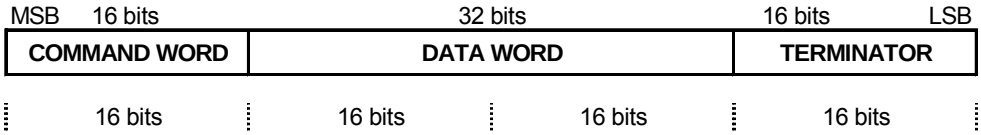
Figure 14.2: DSP Port Command Sequence



14.3.4 PAR Parallel Interface

In this case the command sequence is transmitted in the same order as for the RS232 interface, but split into four 16-bit words, as illustrated below.

Figure 14.3: PAR Command Sequence



14.3.5 Custom User Interfaces

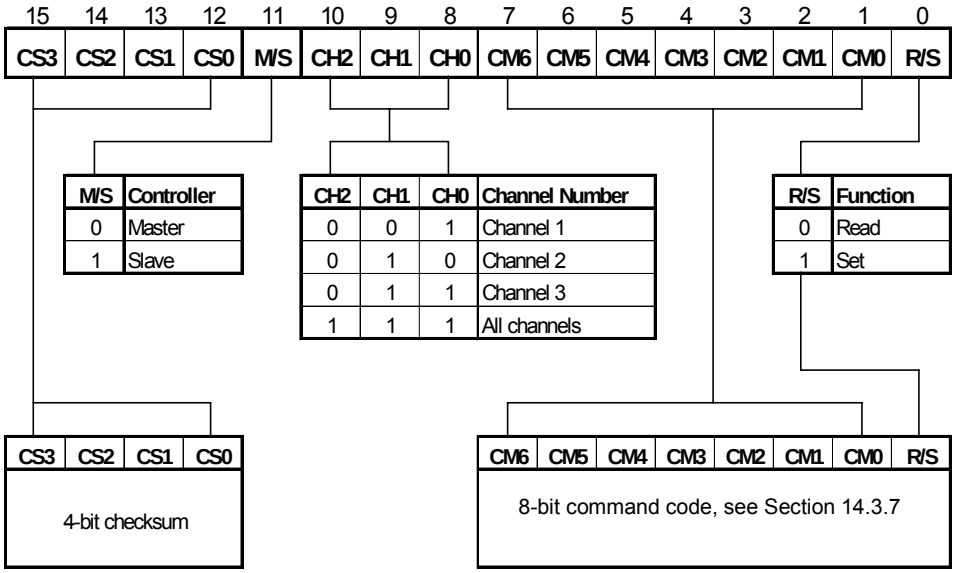
Depending on the data transfer method, the command sequence may be transmitted slightly differently for custom user interfaces.
Please refer to your custom user documentation for further details.

14.3.6 Command Word

The 16-bit command word, illustrated below, comprises:

- An 8-bit command code (bits 0-7)
- A 3-bit channel specifier (bits 8-10)
- A single-bit Controller specifier (bit 11)
- A 4-bit checksum. (bits 12-15)

Figure 14.4: Command Word



14.3.7 Command Code

The command code is formed from command bits CM0-CM6 and read/set bit R/S, allowing a total of 256 possible commands. This arrangement reflects the fact that the majority of commands relate to parameters that can be both set into the Controller or read back from it. Hence, the command structure really provides for 128 pairs of commands, the R/S bit specifying whether a parameter is being read from the Controller or set into it.

Note: There are a few commands that are not associated with specific parameters. These include the commands for resetting the Controller, storing the system configuration, and locking and unlocking the Controller. These commands are classified as set commands because they affect the operation of the Controller.

- All commands which request information from the Controller are Read commands (R/S = 0).
- All commands which affect the operation of the Controller are Set commands (R/S = 1).

The complete list of command codes and their corresponding functions are summarized in Table 14.9.

14.3.8 Channel Specifier

The channel specifier allows commands to be directed to the appropriate channel. All commands associated with channel parameters can be sent to any channel by setting the corresponding code into bits CH0-CH2. In addition, all commands that set the channel parameters can be addressed to all channels simultaneously by setting all three bits to 1. This provides a faster means of control in situations where the same value is to be set into all channels (e.g. initialising the command positions to zero).

Note: Some commands refer to the Controller as a whole and do not relate to any particular channel. These include the commands associated with RS232 baud rate, GPIB address and saving the system configuration. For these commands, the channel specifier is ignored.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

The scope of every command, in terms of its applicability to a single channel, to all channels, or to the Controller as a whole, is indicated in the Command Function Summary (Table 14.2).

14.3.9 Controller Specifier

The controller specifier, bit M/S, allows commands to be directed either to the Master Controller or to a Slave Controller, connected to the master via the DSP Port.

14.3.10 Checksum

The checksum enables the Controller to verify the integrity of the received command sequence. As illustrated, the checksum is calculated by splitting the command and data words into 4-bit wide segments, adding the segments together, and loading the lowest 4 bits of the result into bits CS0-CS3.

Table 14.5: Checksum

	M/S	CH2	CH1	CH0
+	CM6	CM5	CM4	CM3
+	CM2	CM1	CM0	R/W
+	DW31	DW30	DW29	DW28
+	DW27	DW26	DW25	DW24
+	DW23	DW22	DW21	DW20
+	DW19	DW18	DW17	DW16
+	DW15	DW14	DW13	DW12
+	DW11	DW10	DW09	DW08
+	DW07	DW06	DW05	DW04
+	DW03	DW02	DW01	DW00
=	CS3	CS2	CS1	CS0

If the checksum calculated by the Controller does not match that in the received command sequence the Controller ignores the command.

Note: Depending on the data transfer method, the checksum field may not be used on certain custom interfaces. Please refer to the appropriate documentation for further details.

14.3.11 Data Word

The format of the 32-bit data word depends on the type of value being sent, as follows:

- Commands which refer to floating-point parameters use either the IEEE single-precision floating-point format or the TMS320C32 single-precision floating point format, depending on the Controller's data format setting. This setting may be changed using the appropriate command (see Table 14.2).
- Commands which refer to integer parameters use the TMS320C32 single-precision integer format, which is equivalent to the standard signed long integer format.
- Commands which have no associated value transmit a null data word (all bits set to zero).

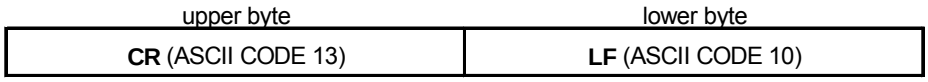
A complete list of the data formats associated with each command is included in the Command Function Summary (Table 14.9).

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

14.3.12 Terminator

The terminator consists of a standard Carriage Return - Line Feed sequence, as illustrated below.

Figure 14.6: Terminator

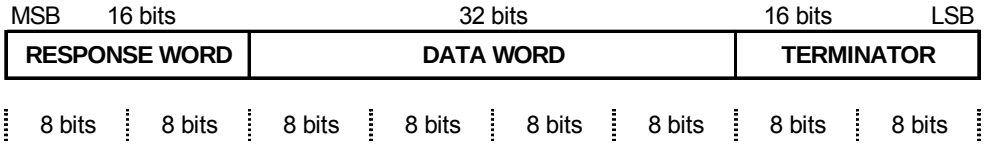


If the command sequence does not have a valid terminator the command is ignored by the Controller.

14.4 Controller Responses

Normally, the Controller will only transmit data in response to one of the Read command sequences (R/S bit set to zero). Response sequences follow a very similar format to the command sequences. The response sequence across the RS232 serial interface is illustrated below, transmitted as eight bytes (most significant byte first).

Figure 14.7: RS232 Response Sequence



The response sequence comprises:

- A 16-bit response word which may be:
  - A valid response. The response word contains a copy of the original command word but with the checksum field updated to take account of the contents of the data word. The response code is therefore identical to the original command code and, together with controller and channel specifiers, is used to identify the type and source of the information contained in the data word.
  - An error response. The response code returned in an error response is always Code 255 (FF Hex), in which case the associated data word will contain the relevant error code, see Error Handling.
- A 32-bit data word containing the requested information. The format of the data word again depends on whether a floating point or integer value is being sent. The complete list of response formats is included in the Command Function Summary (Table 14.9).
- 16-bit CR-LF terminator.

The response sequences are transmitted in exactly the same way as for the command sequences.

14.5 Error Handling

If any command sequence gives rise to an error condition, the Controller returns error response code 255 (FF Hex).

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

In this case, the 32-bit data word contains an error code, reporting the type of error, which occurred. The error codes and their meanings are listed in Table 14.10.

Command handling is illustrated in the following flowchart.

Figure 14.8: Controller Command Handling

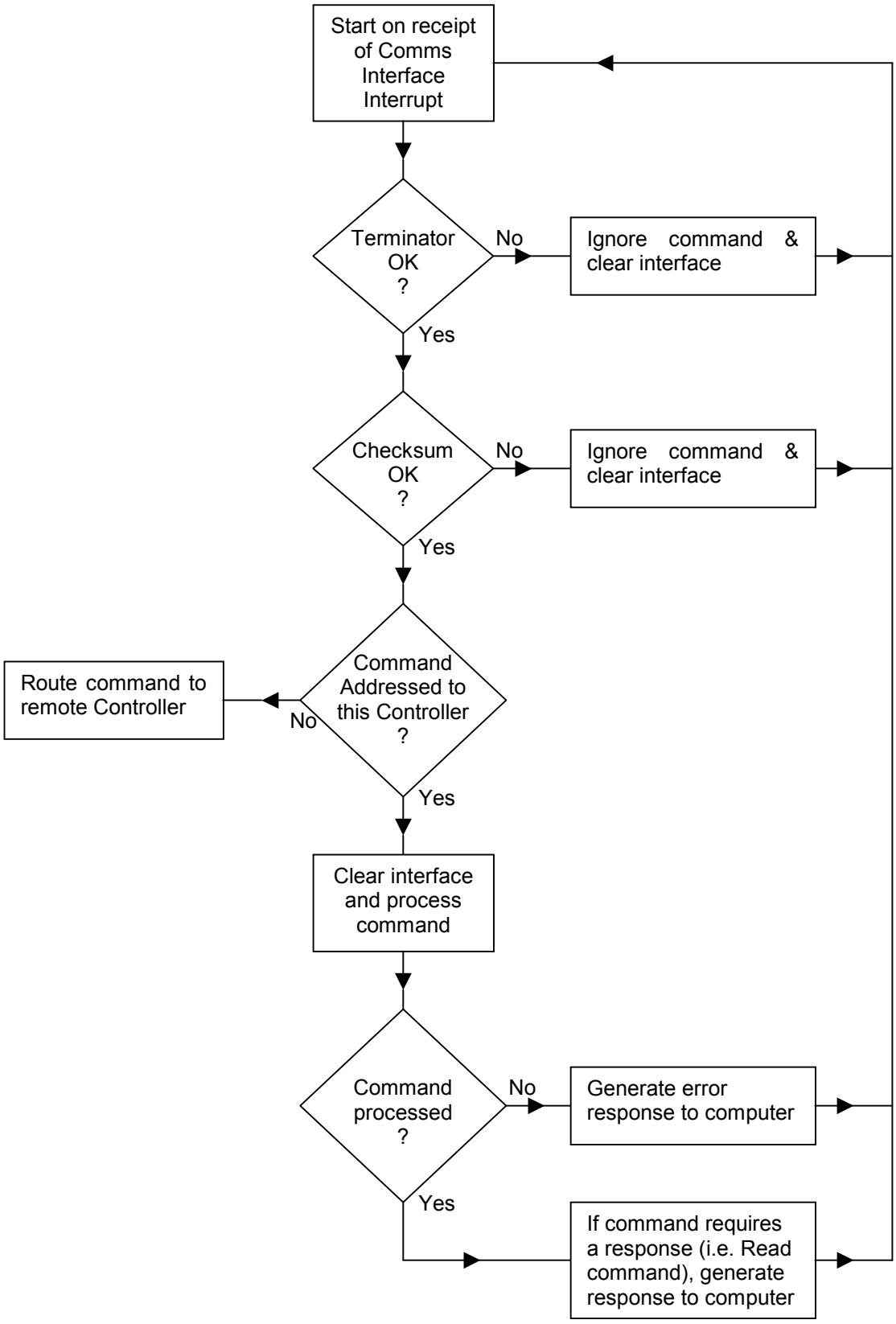


Table 14.9: Command Language Summary

Command Code		Function		Applies To			Security				Data Word Format		Valid Settings	
(Dec)	(Hex)			Controller	Single Channel	All Channels	Always Available	User Locked	Super-User Only	SDLC Only	Command	Response	Minimum	Maximum
0	00	Read	controller hardware ID	✓			✓				integer	integer		
1	01	Set		✓						✓	integer	-	See Text	
2	02	Read	controller software ID	✓			✓				integer	integer		
3	03													
4	04	Read	RS232 baud rate	✓			✓				null	integer		
5	05	Set		✓				✓			integer	-	See Note 1	
6	06	Read	GPIB address	✓			✓				null	integer		
7	07	Set		✓				✓			integer	-		
8	08	Read	data format	✓			✓				null	integer		
9	09	Set		✓				✓			integer	-	See Note 2	
10	0A	Read	synchronisation pulse (TACK) period	✓			✓				null	integer		
11	0B	Set		✓				✓			integer	-	0	32767
12	0C													
13	0D													
14	0E													
15	0F	Set	controller type	✓			✓				integer	-	See Note 3	
16	10													
17	11	Reset	controller	✓			✓				null	-		
18	12													
19	13	Store	system configuration	✓				✓			null	-		
20	14	Read	system status	✓			✓				null	integer		
21	15													
22	16													
23	17	Recall	standard controller configuration	✓						✓	null			
24	18													
25	19	Set	controller lock code	✓				✓			integer	-	-2.15E+09	2.15E+09
26	1A													
27	1B	Lock	controller	✓			✓				null	-		
28	1C													
29	1D	Unlock	controller	✓			✓				integer	-	Controller Lock Code Only	
30	1E													
31	1F													
32	20													
33	21													
34	22													
35	23													
36	24													
37	25													
38	26													
39	27													
40	28													
41	29													
42	2A													
43	2B													
44	2C													
45	2D													
46	2E													
47	2F													
48	30	Read	scratchpad register, SR0	✓			✓				null	integer		
49	31	Set		✓				✓			integer	-	-2.15E+09	2.15E+09
50	32	Read	scratchpad register, SR1	✓			✓				null	integer		
51	33	Set		✓				✓			integer	-	-2.15E+09	2.15E+09
52	34	Read	scratchpad register, SR2	✓			✓				null	integer		
53	35	Set		✓				✓			integer	-	-2.15E+09	2.15E+09
54	36	Read	scratchpad register, SR3	✓			✓				null	integer		
55	37	Set		✓				✓			integer	-	-2.15E+09	2.15E+09
56	38	Read	scratchpad register, SR4	✓			✓				null	integer		
57	39	Set		✓				✓			integer	-	-2.15E+09	2.15E+09
58	3A	Read	scratchpad register, SR5	✓			✓				null	integer		
59	3B	Set		✓				✓			integer	-	-2.15E+09	2.15E+09
60	3C	Read	scratchpad register, SR6	✓			✓				null	integer		
61	3D	Set		✓				✓			integer	-	-2.15E+09	2.15E+09
62	3E	Read	scratchpad register, SR7	✓			✓				null	integer		
63	3F	Set		✓				✓			integer	-	-2.15E+09	2.15E+09

Notes

1. Valid RS232 baud rate values are 1200, 2400, 4800, 9600, 19200 and 384
2. Valid data formats are 0 (IEEE) and 1 (TMS320C32).
3. Valid controller types are 0 (MASTER) and 1 (SLAVE).
4. Setups 0-2 are reserved for factory defaults. Setups 3-7 are for general us
5. Shaded commands are not yet implemented.

Command Code		Function		Applies To			Security				Data Word Format		Valid Settings	
(Dec)	(Hex)			Controller	Single Channel	All Channels	Always Available	User Locked	Super-User Only	SDLQ only	Command	Response	Minimum	Maximum
64	40	Read	stage ID		✓		✓				integer	integer		
65	41	Set							✓				See Text	
66	42	Read	mode/status		✓		✓				null	integer		
67	43	Set	mode		✓	✓	✓				integer	-	See Text	
68	44	Read	unit conversion factor		✓		✓				null	floating-point		
69	45	Set			✓	✓		✓			floating-point	-	Any Value	
70	46	Read	unit conversion offset		✓		✓				null	floating-point		
71	47	Set			✓	✓		✓			floating-point	-	Any Value	
72	48	Read	command position		✓		✓				null	floating-point		
73	49	Set			✓	✓	✓				floating-point	-	Depends on Stage	
74	4A	Read	command position		✓		✓				null	integer		
75	4B	Set			✓	✓	✓				integer	-	Depends on Stage	
76	4C	Read	measured position		✓		✓				null	floating-point		
77	4D													
78	4E	Read	measured position		✓		✓				null	integer		
79	4F													
80	50	Read	ready limit		✓		✓				null	floating-point		
81	51	Set			✓	✓		✓			floating-point	-	Any Value	
82	52	Read	ready limit		✓		✓				null	integer		
83	53	Set			✓	✓		✓			integer	-	Any Value	
84	54	Read	yaw rate		✓		✓				null	floating-point		
85	55	Set			✓	✓	✓				floating-point	-	Any Value	
86	56	Read	lever arm length		✓		✓				null	floating-point		
87	57	Set			✓	✓	✓				floating-point	-	Any Value	
88	58	Read	linearisation coefficient, b_0		✓		✓				null	floating-point		
89	59	Set			✓	✓		✓			floating-point	-	-10	+10
90	5A	Read	linearisation coefficient, b_1		✓		✓				null	floating-point		
91	5B	Set			✓	✓		✓			floating-point	-	-10	+10
92	5C	Read	linearisation coefficient, b_2		✓		✓				null	floating-point		
93	5D	Set			✓	✓		✓			floating-point	-	-10	+10
94	5E	Read	linearisation coefficient, b_3		✓		✓				null	floating-point		
95	5F	Set			✓	✓		✓			floating-point	-	-10	+10
96	60	Read	linearisation coefficient, b_4		✓		✓				null	floating-point		
97	61	Set			✓	✓		✓			floating-point	-	-10	+10
98	62	Read	data sampling time, T_s		✓		✓				null	floating-point		
99	63													
100	64	Read	integrator time constant, T_i		✓		✓				null	floating-point		
101	65	Set			✓	✓	✓				floating-point	-	0.000001	0.01
102	66	Read	differentiator time constant, T_d		✓		✓				null	floating-point		
103	67	Set			✓	✓	✓				floating-point	-	0	0.001
104	68	Read	tracking time constant, T_t		✓		✓				null	floating-point		
105	69	Set			✓	✓	✓				floating-point	-	0.000001	0.01
106	6A	Read	proportional gain, G_p		✓		✓				null	floating-point		
107	6B	Set			✓	✓	✓				floating-point	-	-10	+10
108	6C	Read	differential gain, G_d		✓		✓				null	floating-point		
109	6D	Set			✓	✓	✓				floating-point	-	-10	+10
110	6E	Read	set point weighting, G_{sp}		✓		✓				null	floating-point		
111	6F	Set			✓	✓	✓				floating-point	-	0.0	1.0
112	70	Read	maximum integrator error, e_{max}		✓		✓				null	integer		
113	71	Set			✓	✓	✓				integer	-	-32768	+32767
114	72	Read	minimum integrator error, e_{min}		✓		✓				null	integer		
115	73	Set			✓	✓	✓				integer	-	-32768	+32767
116	74													
117	75													
118	76													
119	77													
120	78													
121	79													
122	7A													
123	7B													
124	7C													
125	7D													
126	7E													
127	7F													

Command Code		Function		Applies To			Security				Data Word Format		Valid Settings	
(Dec)	(Hex)			Controller	Single Channel	All Channels	Always Available	User Locked	Super-User Only	SDLG Only	Command	Response	Minimum	Maximum
128	80	Read	charge amplifier zero offset		✓		✓				null	integer		
129	81	Set			✓	✓		✓			integer	-	0	4095
130	82	Read	charge amplifier range		✓		✓				null	integer		
131	83	Set			✓	✓		✓			integer	-	0	4095
132	84	Read	charge amplifier coarse gain		✓		✓				null	integer		
133	85	Set			✓	✓		✓			integer	-	0	4095
134	86	Read	charge amplifier fine gain		✓		✓				null	integer		
135	87	Set			✓	✓		✓			integer	-	0	4095
136	88	Read	sensor scale factor		✓		✓				null	floating-point		
137	89	Set			✓	✓		✓			floating-point	-	Any Value	
138	8A	Read	sensor scale offset		✓		✓				null	floating-point		
139	8B	Set			✓	✓		✓			floating-point	-	Any Value	
140	8C	Read	actuator scale factor		✓		✓				null	floating-point		
141	8D	Set			✓	✓		✓			floating-point	-	Any Value	
142	8E	Read	actuator scale offset		✓		✓				null	floating-point		
143	8F	Set			✓	✓		✓			floating-point	-	Any Value	
144	90	Read	analogue input scale factor		✓		✓				null	floating-point		
145	91	Set			✓	✓		✓			floating-point	-	Any Value	
146	92	Read	analogue input scale offset		✓		✓				null	floating-point		
147	93	Set			✓	✓		✓			floating-point	-	Any Value	
148	94													
149	95													
150	96													
151	97													
152	98													
153	99													
154	9A													
155	9B													
156	9C													
157	9D													
158	9E													
159	9F													
160	A0													
161	A1													
162	A2													
163	A3													
164	A4													
165	A5													
166	A6													
167	A7													
168	A8													
169	A9													
170	AA													
171	AB													
172	AC													
173	AD													
174	AE													
175	AF													
176	B0													
177	B1	Store	dynamic setup		✓	✓	✓				integer	-	See Note 4	7
178	B2													
179	B3	Recall	dynamic setup		✓	✓	✓				integer	-	0	7
180	B4	Read	default setup number		✓		✓				null	integer		
181	B5	Set			✓	✓		✓			integer	-	0	7
182	B6	Read	current setup number		✓		✓				null	integer		
183	B7													
184	B8													
185	B9													
186	BA													
187	BB													
188	BC													
189	BD													
190	BE													
191	BF	Store	stage configuration		✓				✓		null	-		

Command Code		Function		Applies To			Security				Data Word Format		Valid Settings	
(Dec)	(Hex)			Controller	Single Channel	All Channels	Always Available	User Locked	Super-User Only	SDLQ Only	Command	Response	Minimum	Maximum
192	C0	Read	stimulus channel number	✓			✓				null	integer		
193	C1	Set		✓			✓				integer	-	1	See Note 6
194	C2	Read	stimulus amplitude	✓			✓				null	floating-point		
195	C3	Set		✓			✓				floating-point	-	Any Value	
196	C4	Read	stimulus leading edge	✓			✓				null	integer		
197	C5	Set		✓			✓				integer	-	1	16383
198	C6	Read	stimulus trailing edge	✓			✓				null	integer		
199	C7	Set		✓			✓				integer	-	1	16383
200	C8	Read	response channel number	✓			✓				null	integer		
201	C9	Set		✓			✓				integer	-	1	See Note 6
202	CA	Read	response capture time	✓			✓				null	integer		
203	CB	Set		✓			✓				integer	-	1	16383
204	CC	Read	response data	✓			✓				integer	floating-point	0	See Note 7
205	CD													
206	CE													
207	CF	Fire	snapshot	✓			✓				null	null		
208	D0	Read	function generator mode	✓			✓				null	integer		
209	D1	Set		✓			✓				integer	-		See Text
210	D2	Read	waveform type		✓		✓				null	integer		
211	D3	Set			✓	✓	✓				integer	-		See Note 8
212	D4	Read	waveform period		✓		✓				null	floating-point		
213	D5	Set			✓	✓	✓				floating-point	-	0.00008192	5.368
214	D6	Read	waveform scale factor		✓		✓				null	floating-point		
215	D7	Set			✓	✓	✓				floating-point	-	Any Value	Any Value
216	D8	Read	waveform scale offset		✓		✓				null	floating-point		
217	D9				✓	✓	✓				floating-point	-	Any Value	
218	DA	Read	waveform data	✓			✓				null	integer		
219	DB	Set		✓				✓			integer	-		See Text
220	DC													
221	DD	Store	waveform data	✓				✓			null	null		
222	DE													
223	DF													
224	E0	Read	charge amplifier zero offset correction factor		✓		✓				null	floating-point		
225	E1	Set			✓	✓		✓			floating-point		0.75	1.25
226	E2	Read	charge amplifier range correction factor		✓		✓				null	floating-point		
227	E3	Set			✓	✓		✓			floating-point		0.75	1.25
228	E4	Read	charge amplifier coarse gain correction factor		✓		✓				null	floating-point		
229	E5	Set			✓	✓		✓			floating-point		0.75	1.25
230	E6	Read	charge amplifier fine gain correction factor		✓		✓				null	floating-point		
231	E7	Set			✓	✓		✓			floating-point		0.75	1.25
232	E8													
233	E9													
234	EA													
235	EB													
236	EC													
237	ED													
238	EE													
239	EF													
240	F0													
241	F1													
242	F2													
243	F3													
244	F4													
245	F5													
246	F6													
247	F7													
248	F8													
249	F9													
250	FA													
251	FB													
252	FC	Read	data field index	✓					✓		null	integer		
253	FD	Set		✓					✓		integer	-	See Text	
254	FE	Null response									-	null		
255	FF	Error condition									-	integer		

Table 14.10: Error Codes

Error Group	Error Code (Decimal)	Error Condition
General Command Errors	100	Terminator not found
	101	Checksum invalid
	102	Command invalid
	103	Command not available
	104	Channel specifier invalid
	105	Channel not available
	110	Value out of range
	111	Integer overflow
ID Command Errors	200	Hardware ID index invalid
	201	Software ID index invalid
	202	Stage ID index invalid
	210	Data field index invalid
Controller Command Errors	300	RS232 baud rate invalid
	301	GPIB address invalid
	302	Data format specifier invalid
	303	Synchronization period invalid
	304	Controller type invalid
	310	Lock code invalid
	311	No lock code defined
	312	Controller already locked
	313	Controller already unlocked
EEPROM Errors	400	Controller EEPROM write error
	410	Stage EEPROM write error
	411	Stage EEPROM read error
Position Command Errors	1000	Command position invalid
	1001	Ready limit invalid
PID Command Errors	1100	Integrator time constant invalid
	1101	Differentiator time constant invalid
	1102	Tracking time constant invalid
	1103	Proportional gain invalid
	1104	Differential gain invalid
	1105	Set point weighting invalid
	1106	Maximum error invalid
	1107	Minimum error invalid
Linearisation Command Errors	1200	Coefficient b_0 invalid
	1201	Coefficient b_1 invalid
	1202	Coefficient b_2 invalid
	1203	Coefficient b_3 invalid
	1204	Coefficient b_4 invalid
Calibration Command Errors	1300	Charge amp coarse gain invalid
	1301	Charge amp fine gain invalid
	1302	Charge amp offset invalid
	1303	Charge amp range invalid
	1310	Sensor scale factor invalid
	1311	Sensor scale offset invalid
	1312	Actuator scale factor invalid
	1313	Actuator scale offset invalid
Set-up/Save Command Errors	1900	Set-up number invalid
	1901	Default set-up number invalid
Snapshot Command Errors	2000	Stimulus channel number invalid
	2001	Stimulus amplitude invalid
	2002	Stimulus leading edge invalid
	2003	Stimulus trailing edge invalid
	2010	Response channel number invalid
	2011	Response capture time invalid
	2012	Response data subscript out of range

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Error Group	Error Code (Decimal)	Error Condition
Function Generator Command Errors	2100	Function generator busy
	2101	Waveform type invalid
	2102	Waveform period invalid
Correction Factor Command Errors	2300	Charge amp coarse gain factor invalid
	2301	Charge amp fine gain correction faction invalid
	2302	Charge amp offset correction factor invalid
	2303	Charge amp range correction factor invalid
Communications Errors	10010	DSP port transmit timeout
	10011	DSP port receive timeout
	10012	DSP port busy

Table 14.11: Mode/Status Bit Assignments

Bit	Name	Function
0	Enable	Enables channel
1	Closed	Closes servo loop
2	Invert	Inverts phase of sensor signal
3	Freeze	Freezes position output
4	Stimulus	Enables snapshot stimulus *(internal use only)*
5	Response	Enables snapshot response *(internal use only)*
6	-	
7	-	
8	-	
9	-	
10	-	
11	-	
12	-	
13	-	
14	-	
15	-	
16	Ready	Servo loop position within ready limit
17	-	
18	-	
19	-	
20	-	
21	-	
22	-	
23	-	
24	-	
25	-	
26	-	
27	-	

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Bit	Name	Function
28	-	
29	-	
30	-	
31	-	

Notes: 1. All bits active high.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

15 Dynamic Link Library (DLL)

15.1 Overview

This Chapter details the Dynamic Link Library (DLL) for NPS3110, NPS3220 and NPS3330 Digital Controllers. The DLL provides a simple and efficient way to include control and monitoring of the Controller in your Windows based application software. Both the serial RS232C Interface and the high speed NPS-PAR Parallel Interface are supported.

The DLL supports the entire command set of the Command Language (see Section 14) as a series of easy-to-use function calls from your application.

15.2 Introduction

NGCMOD32.dll exports all the functions you need to communicate with a Digital Controller to set and interrogate controller parameters.

The Dynamic Link Library ***NGCMOD32.dll*** is designed to work within Windows 95 and Window NT. Two additional files (***ngcmod32.h*** and ***ngcmod32.lib***) are supplied which are needed if you intend developing your own software to interrogate the controller.

Refer to Section 14 'Command Language' for details of specific controller commands.

15.3 Functions contained within NGCMOD32.h

NGCMOD32.h contains all of the exported function declarations from ngcmod32.dll.

InitComm, RestartCommPort and RestartCommRate are concerned with communication between the host computer and controller. All the other functions are concerned with setting and reading controller parameters.

The contents of ngcmod32.h are printed below, the code is available on the Software Sample Disk supplied.

15.4 NGCMOD32.h Code Print-out

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
#define DllExport       __declspec( dllexport )

extern "C"  {

int  DllExport  InitComm(LPSTR  port,LPSTR  baud,LPSTR  parity,LPSTR  databits,  LPSTR  stopbits,int  far
*error);
int DllExport RestartCommPort(LPSTR port);
int DllExport RestartCommRate(int ibaudrate);
int DllExport Set_ParallelDeviceNo(int far *device, int far *baseaddress);

int DllExport Read_Controller_Hardware_ID(int far *controller,int far *channel, long far *dest);
int DllExport Set_Controller_Hardware_ID(int far *controller,int far *channel, long far *source);
int DllExport Read_Controller_Software_ID(int far *controller,int far *channel, long far *dest);

int DllExport Read_RS232_baud_rate(int far *controller,long far* destination);
int DllExport Set_RS232_baud_rate(int far *controller, long far *source);
int DllExport Read_Data_Format(int far *controller,int far *channel,long far *dest);
int DllExport Set_Data_Format(int far *controller,int far *channel,long far *dest);

int DllExport Read_Sync_Pulse(int far *controller,int far *channel,long far *dest);
int DllExport Set_Sync_Pulse(int far *controller,int far *channel,long far *dest);
int DllExport Set_Controller_Type(int far *controller,int far *channel,long far *dest);
int DllExport Reset_Controller(int far *controller,int far *channel,long far *dest);
int DllExport Read_System_Status(int far *controller,int far *channel,long far *dest);
int DllExport Set_Controller_LockCode(int far *controller,int far *channel,long far *dest);
int DllExport Lock_Controller(int far *controller,int far *channel,long far *dest);
int DllExport Unlock_Controller(int far *controller,int far *channel,long far *dest);
int DllExport Read_Stage_ID(int far *controller,int far *channel,long far *dest);
int DllExport Set_Stage_ID(int far *controller,int far *channel,long far *dest);
int DllExport Read_Mode(int far *controller,int far *channel, long far *dest);
int DllExport Mode(int far *controller, int far *channel,long far* source);
int DllExport Read_Unit_Conversion_Factor(int far *controller,int far *channel, float far *dest);
int DllExport Set_Unit_Conversion_Factor(int far *controller,int far *channel,float far *source);
int DllExport Read_Unit_Conversion_Offset(int far *controller,int far *channel, float far *dest);
int DllExport Set_Unit_Conversion_Offset(int far *controller,int far *channel, float far *source);
int DllExport Read_Command_Position_F(int far *controller,int far *channel, float far *dest);
int DllExport Set_Command_Position_F(int far *controller,int far *channel, float far *source);
int DllExport Read_Command_Position_I(int far *controller,int far *channel, long far *dest);
int DllExport Set_Command_Position_I(int far *controller,int far *channel, long far *source);
int DllExport Read_Measured_Position_F(int far *controller,int far *channel, float far *dest);
int DllExport Read_Measured_Position_I(int far *controller,int far *channel, long far *dest);
int DllExport Read_Ready_Limit_F(int far *controller,int far *channel, float far *dest);
int DllExport Set_Ready_Limit_F(int far *controller,int far *channel, float far *source);
int DllExport Read_Ready_Limit_I(int far *controller,int far *channel, long far *dest);
int DllExport Set_Ready_Limit_I(int far *controller,int far *channel, long far *source);
int DllExport Read_Yaw_Rate(int far *controller,int far *channel, float far *dest);
int DllExport Set_Yaw_Rate(int far *controller,int far *channel, float far *source);
int DllExport Read_Lever_Arm_Length(int far *controller,int far *channel, float far *dest);
int DllExport Set_Lever_Arm_Length(int far *controller,int far *channel, float far *source);
int DllExport Read_Linco_B0(int far *controller,int far *channel, float far *dest);
int DllExport Set_Linco_B0(int far *controller,int far *channel, float far *source);
int DllExport Read_Linco_B1(int far *controller,int far *channel, float far *dest);
int DllExport Set_Linco_B1(int far *controller,int far *channel, float far *source);
int DllExport Read_Linco_B2(int far *controller,int far *channel, float far *dest);
int DllExport Set_Linco_B2(int far *controller,int far *channel, float far *source);
int DllExport Read_Linco_B3(int far *controller,int far *channel, float far *dest);
int DllExport Set_Linco_B3(int far *controller,int far *channel, float far *source);
int DllExport Read_Linco_B4(int far *controller,int far *channel, float far *dest);
int DllExport Set_Linco_B4(int far *controller,int far *channel, float far *source);
int DllExport Read_Data_Sample_Time(int far *controller,int far *channel, float far *dest);
int DllExport Read_Intgrat_Const(int far *controller,int far *channel, float far *dest);
int DllExport Set_Intgrat_Const(int far *controller,int far *channel, float far *source);
int DllExport Read_Diff_Const(int far *controller,int far *channel, float far *dest);
int DllExport Set_Diff_Const(int far *controller,int far *channel, float far *source);
int DllExport Read_Track_Time_Const(int far *controller,int far *channel, float far *dest);
int DllExport Set_Track_Time_Const(int far *controller,int far *channel, float far *source);
int DllExport Read_Prop_Gain(int far *controller,int far *channel, float far *dest);
int DllExport Set_Prop_Gain(int far *controller,int far *channel, float far *source);
int DllExport Read_Diff_Gain(int far *controller,int far *channel, float far *dest);
int DllExport Set_Diff_Gain(int far *controller,int far *channel, float far *source);
int DllExport Read_SPoint_Weight(int far *controller,int far *channel, float far *dest);
int DllExport Set_SPoint_Weight(int far *controller,int far *channel, float far *source);
int DllExport Read_Max_Intgrat_Error(int far *controller,int far *channel, long far *dest);
int DllExport Set_Max_Intgrat_Error(int far *controller,int far *channel, long far *source);
int DllExport Read_Min_Intgrat_Error(int far *controller,int far *channel, long far *dest);
int DllExport Set_Min_Intgrat_Error(int far *controller,int far *channel, long far *source);
int DllExport Read_Chargeamp_Zoffset(int far *controller,int far *channel, long far *dest);
int DllExport Set_Chargeamp_Zoffset(int far *controller,int far *channel, long far *source);
int DllExport Read_Chargeamp_Range(int far *controller,int far *channel, long far *dest);
int DllExport Set_Chargeamp_Range(int far *controller,int far *channel, long far *source);
int DllExport Read_Chargeamp_Coarse(int far *controller,int far *channel, long far *dest);
int DllExport Set_Chargeamp_Coarse(int far *controller,int far *channel, long far *source);
int DllExport Read_Chargeamp_Fine(int far *controller,int far *channel, long far *dest);
int DllExport Set_Chargeamp_Fine(int far *controller,int far *channel, long far *source);
int DllExport Read_Scale_Factor(int far *controller,int far *channel, float far *dest);
int DllExport Set_Scale_Factor(int far *controller,int far *channel, float far *source);
int DllExport Read_Offset(int far *controller,int far *channel, float far *dest);
int DllExport Set_Offset(int far *controller,int far *channel, float far *source);
```

```
int DllExport Read_Analogue_Input_Scale_Factor(int far *controller,int far *channel, float far *dest);
int DllExport Set_Analogue_Input_Scale_Factor(int far *controller,int far *channel, float far *dest);
int DllExport Read_Analogue_Input_Scale_Offset(int far *controller,int far *channel, float far *dest);
int DllExport Set_Analogue_Input_Scale_Offset(int far *controller,int far *channel, float far *dest);

int DllExport Read_Actuator_Offset(int far *controller,int far *channel, float far *dest);
int DllExport Set_Actuator_Offset(int far *controller,int far *channel, float far *dest);
int DllExport Read_Actuator_Factor(int far *controller,int far *channel, float far *dest);
int DllExport Set_Actuator_Factor(int far *controller,int far *channel, float far *dest);
int DllExport Read_Stimulus_Channel_No(int far *controller,int far *channel, long far *dest);
int DllExport Set_Stimulus_Channel_No(int far *controller,int far *channel, long far *dest);
int DllExport Read_Stimulus_Amplitude(int far *controller,int far *channel, float far *dest);
int DllExport Set_Stimulus_Amplitude(int far *controller,int far *channel, float far *dest);
int DllExport Read_Stimulus_Leading_Edge(int far *controller,int far *channel, long far *dest);
int DllExport Set_Stimulus_Leading_Edge(int far *controller,int far *channel, long far *dest);
int DllExport Read_Stimulus_Trailing_Edge(int far *controller,int far *channel, long far *dest);
int DllExport Set_Stimulus_Trailing_Edge(int far *controller,int far *channel, long far *dest);
int DllExport Read_Response_Channel_No(int far *controller,int far *channel, long far *dest);
int DllExport Set_Response_Channel_No(int far *controller,int far *channel, long far *dest);
int DllExport Read_Response_Capture_Time(int far *controller,int far *channel, long far *dest);
int DllExport Set_Response_Capture_Time(int far *controller,int far *channel, long far *dest);
int DllExport Read_Response_Data(int far *controller,int far *channel, float far *dest);
int DllExport Fire_Snapshot(int far *controller,int far *channel, float far *dest);

int  DllExport Store_Channel_Setup(int far *controller,int far *channel, long far *dest);
int  DllExport Recall_Channel_Setup(int far *controller,int far *channel, long far *dest);
int  DllExport Read_Default_Setup_Number(int far *controller,int far *channel, long far *dest);
int  DllExport Set_Default_Setup_Number(int far *controller,int far *channel, long far *dest);
int  DllExport Read_Current_Setup_Number(int far *controller,int far *channel, long far *dest);
int  DllExport Store_Stage_Configuration(int far *controller,int far *channel, long far *dest);
int  DllExport Store_System_Configuration(int far *controller,int far *channel, long far *dest);
int  DllExport Read_Data_Field_Index(int far *controller,int far *channel, long far *dest);
int  DllExport Set_Data_Field_Index(int far *controller,int far *channel, long far *dest);

int  DllExport Read_Function_Generator_Mode(int far *controller,int far *channel, long far *dest);
int  DllExport Set_Function_Generator_Mode(int far *controller,int far *channel, long far *source);
int  DllExport Read_Waveform_Type(int far *controller,int far *channel, long far *dest);
int  DllExport Set_Waveform_Type(int far *controller,int far *channel, long far *source);
int  DllExport Read_Waveform_Period(int far *controller,int far *channel, float far *dest);
int  DllExport Set_Waveform_Period(int far *controller,int far *channel, float far *source);
int  DllExport Read_Waveform_Scale_Factor(int far *controller,int far *channel, float far *dest);
int  DllExport Set_Waveform_Scale_Factor(int far *controller,int far *channel, float far *source);
int  DllExport Read_Waveform_Scale_Offset(int far *controller,int far *channel, float far *dest);
int  DllExport Set_Waveform_Scale_Offset(int far *controller,int far *channel, float far *source);
int  DllExport Read_Waveform_Data(int far *controller,int far *channel, long far *dest);
int  DllExport Set_Waveform_Data(int far *controller,int far *channel, long far *source);
int  DllExport Store_Waveform_Data(int far *controller,int far *channel, long far *source);

}
```

LabVIEW® - Introduction

Sections 16 and 17 of this Manual are designed to help the user load and operate the NanoScan software and the Lab VIEW Driver functions for the NPS3110, 3220 and 3330 digital controllers. There are 4 options that can be downloaded, the NanoScan executable, the LabVIEW® Runtime Engine, the LabVIEW® source code and the Hardware Driver libraries.

If the target PC doesn't have LabVIEW® 5.1.1 on it, the LabVIEW® Runtime Engine will need to be installed to allow the NanoScan executable to work.

16 NanoScan

NanoScan is a 3 axis scan control package for use with the NPS3330 family of Digital Controllers. It will measure the axis position and control the movement of the stages according to the scan settings. It is written in LabVIEW® 5.1.1. and is supplied in executable format and as source code for users who wish to integrate it into there test and measurement system.

16.1 Starting NanoScan

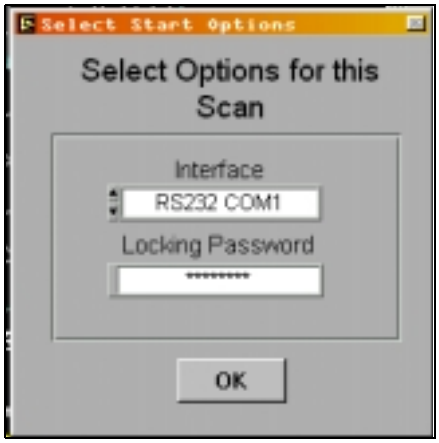
16.1.1 Connecting Controller to PC

The controller should already be connected to the PC and switched on as described in Section 2 of this document before commencing with this section.

16.1.2 Starting the Program

Start the program by double clicking the program icon or selecting its start menu option. After a short delay the program will load and the user will be presented with the Select Start Options dialog box. Select the required interface, enter the locking password and press [OK].

Figure 16.1: Start Options



The locking state should be displayed on the NanoScan front panel when it has loaded.

If NanoScan cannot find a controller a "Controller Comms Error!…" will be reported in the status bar, and nonsense values for the controller type, serial number etc will be returned in the status area .

If NanoScan correctly detects the controller and its' stages it will load their status information and display "Controller Communicating…." in the status bar.

Clicking on the relevant channel Tab (Chan1, Chan2 or Chan3) will allow inspection of the Stage connected to each channel.

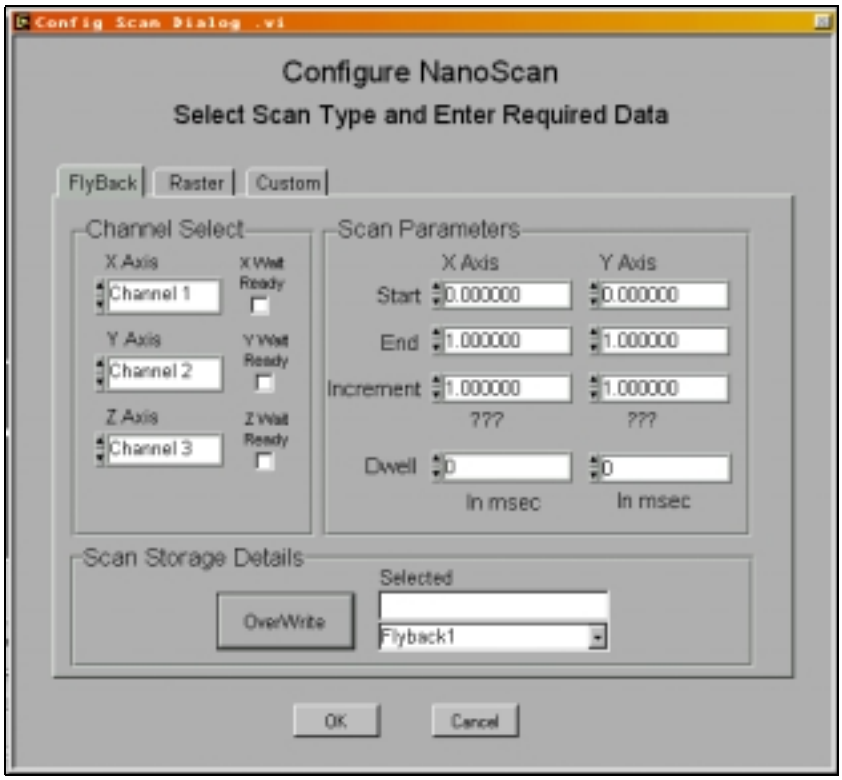
Online help can be displayed by either checking the Help checkbox or selecting Help >> Show Help from the Help Menu. Also available on the Help Menu is About NanoScan… disclaimer and information box.

Version and part number information can be obtained by selecting Windows >> Show VI Info from the Windows Menu.

16.2 Configuring a Scan

16.2.1 Description

Figure 16.2: Configuring a Fly back Scan

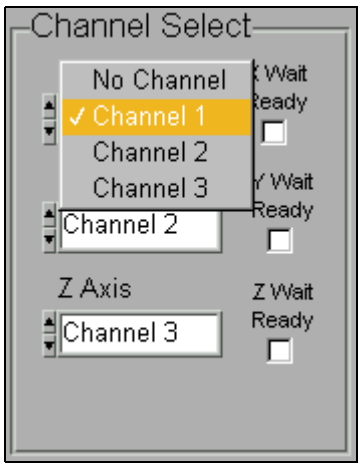


Select [Configure Scan] from the NanoScan front panel.

Select the Fly Back or Raster Tab from the Dialog box displayed.

Map the scan Axis to its relevant channel using the Channel Select Axis drop down menus.

Figure 16.3: Map Channel to Axis



Check Scan Axis Wait Ready checkbox if required. This will make the program wait until the axis is in position from the previous move before commencing the next move.

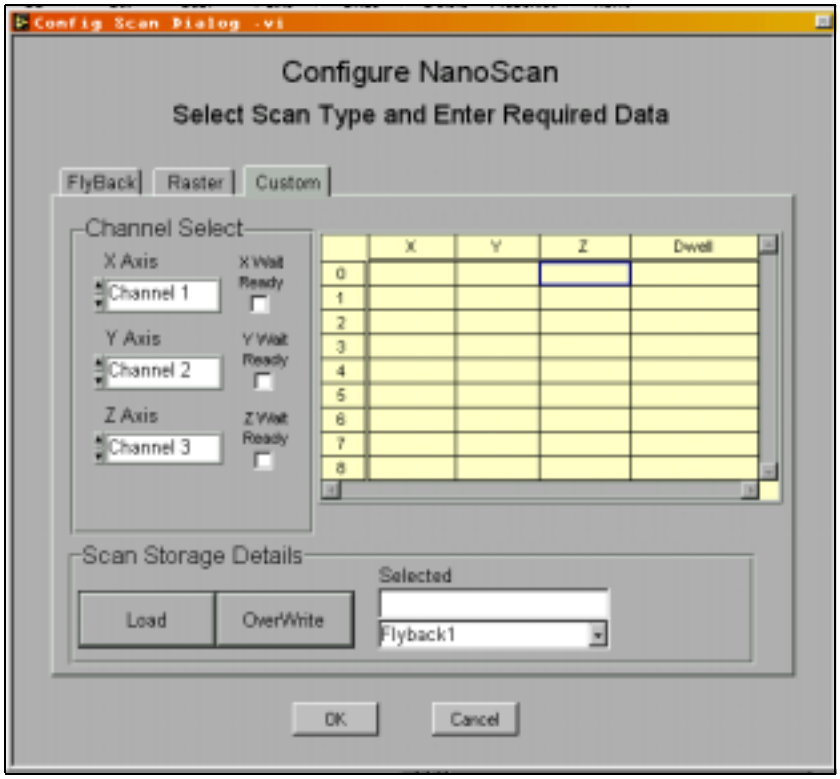
Enter the scan parameters in the displayed units as required: Start Position, End Position, step Increment and step Dwell Time.

Enter a new name for this scan or select an existing scan to overwrite and press [Save] or [OverWrite].

Press [Ok] to set scan in memory or [Cancel] to ignore scan.

The Scan Type icon on NanoScan's front panel will indicate the type of scan.

Figure 16.4: Configuring a Custom Scan



Any scan can be loaded, changed and saved using the custom scan facilities.

The left-hand column corresponds to the scan step, X, Y and Z columns to the scan axis and the dwell is for each scan step.

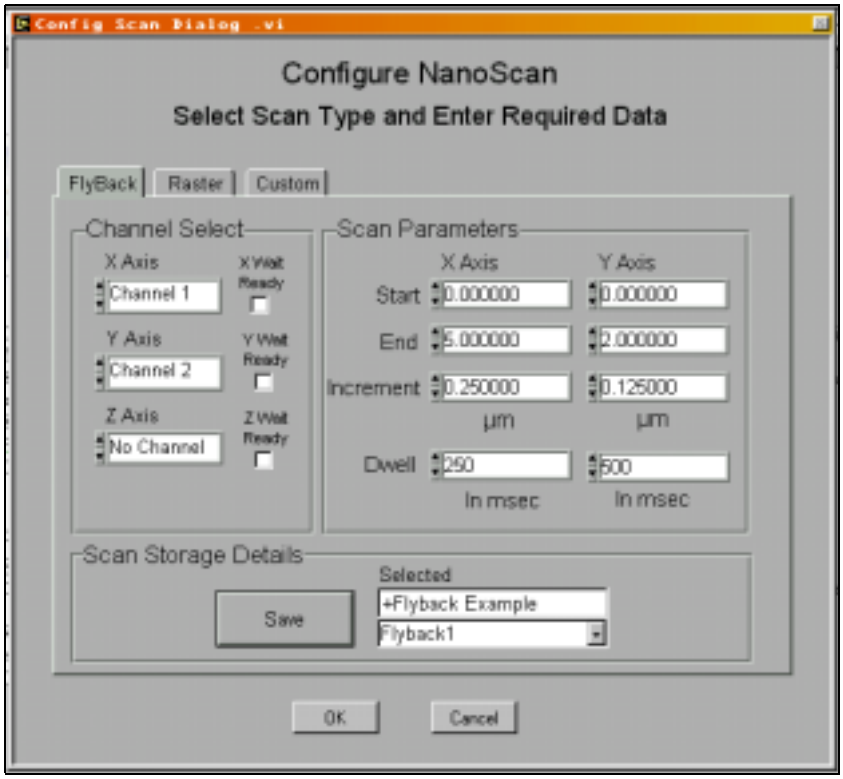
16.2.2 Positive Fly Back Example

Requirement
To move from 0 - 5µm in the X-axis using 0.25µm steps, the end of each step will increment the Y-axis from 0 to 2 in 0.125µm steps. The dwell for each X movement will be 250ms, the dwell for each Y movement will be 500ms. The X-axis will be connected to Channel 1 and the Y-axis to Channel 2. The control mode will be command mode so there is no need to set the Wait for Ready check box.

The example shown below will be stored as +Fly back Example.

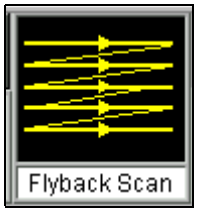
Settings

Figure 16.5: Positive Fly back Example



When [OK] is pressed the Fly back Icon will be displayed on the NanoScan front panel.

Figure 16.6: Fly back Icon



Resultant Graphs

Figure 16.7: Positive Fly back 1-Dimension Graph

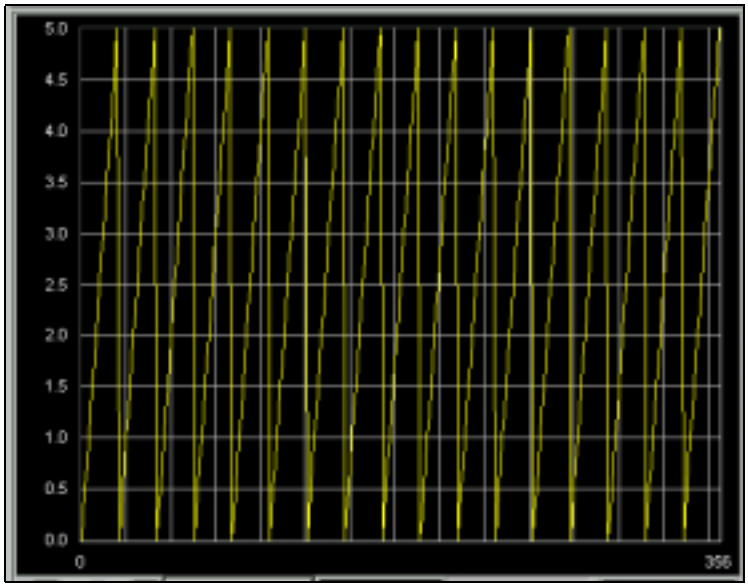


Figure 16.8: Positive Fly back 2-Dimension Graph

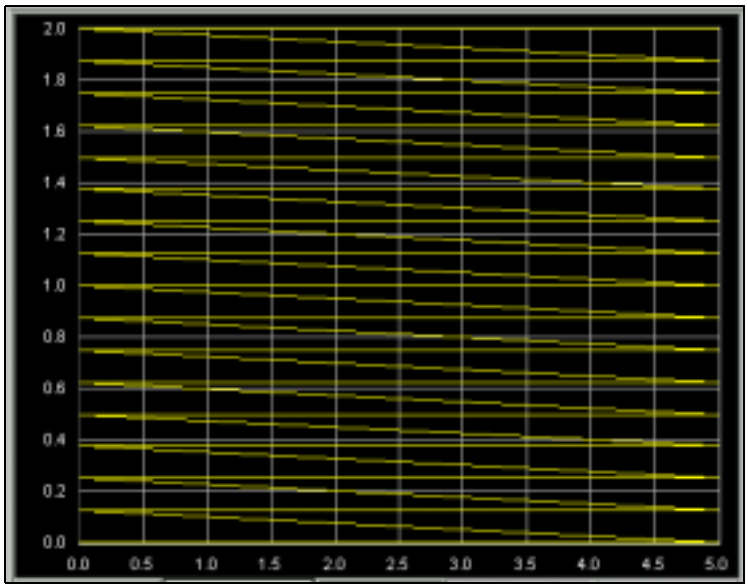
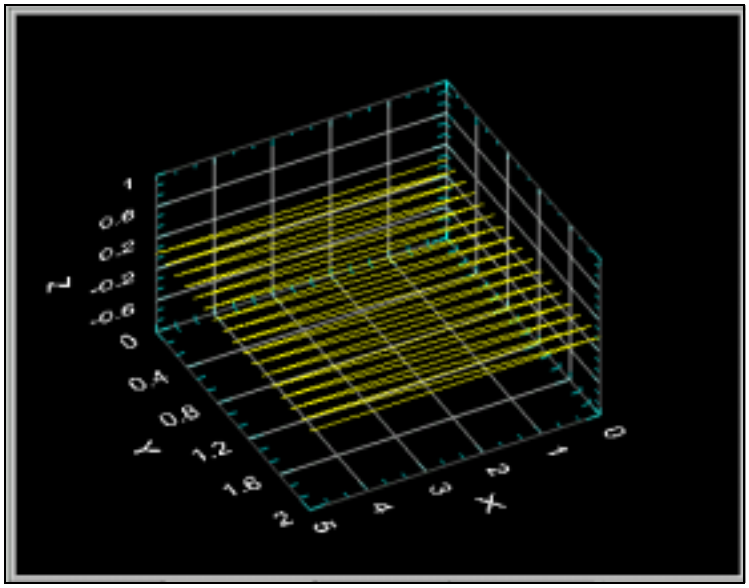


Figure 16.9: Positive Fly back 3-Dimension Graph



The 1-dimension graph maps the X-axis against time, the 2-dimensio maps X and Y and the 3-dimension maps X and Y with reference to Z.

16.2.3 Negative Raster Example

Requirement

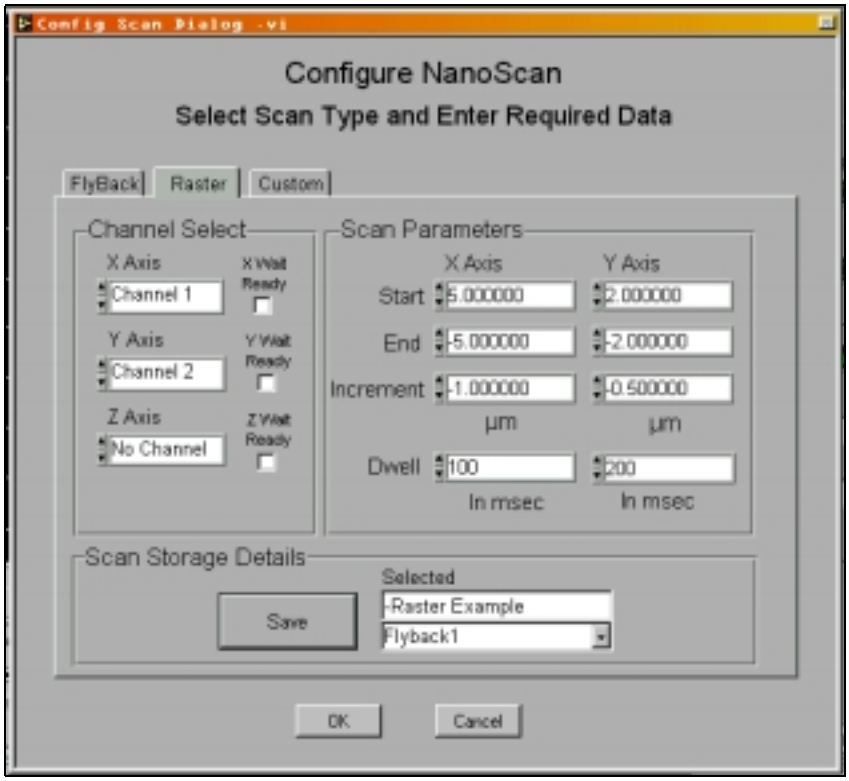
To move from 5µm to -5µm in the X-axis using -1µm steps, the end of each step will decrement the Y-axis from 2µm to –2µm in –0.500µm steps. The dwell for each X movement will be 100ms, the dwell for each Y movement will be 200ms. The X-axis will be connected to Channel 1 and the Y-axis to Channel 2. The control mode will be command mode so there is no need to set the Wait for Ready check box.

The example shown below will be stored as -Raster Example.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

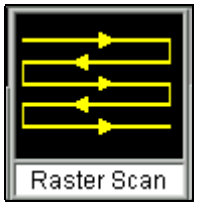
Settings

Figure 16.10: Negative Raster Example



When [OK] is pressed the Raster Icon will be displayed on the NanoScan front panel.

Figure 16.11: Raster Icon



Resultant Graphs

Figure 16.12: Negative Raster 1-Dimension Graph

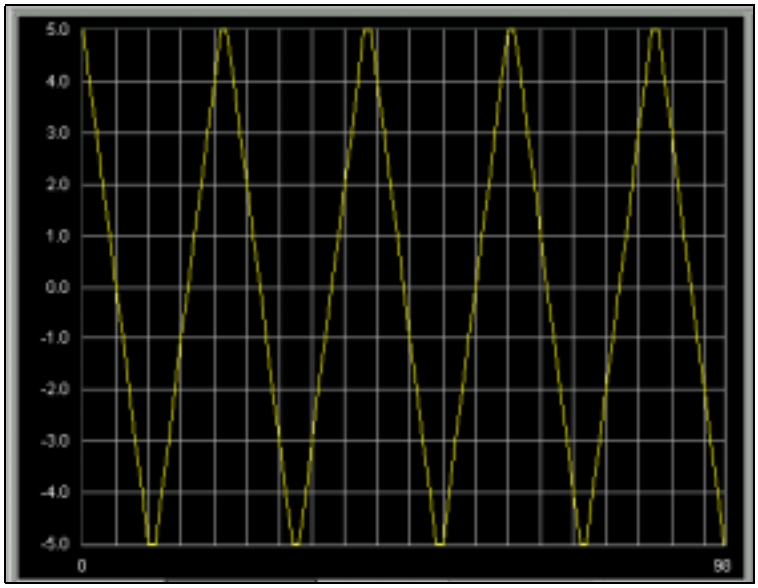


Figure 16.13: Negative Raster 2-Dimension Graph

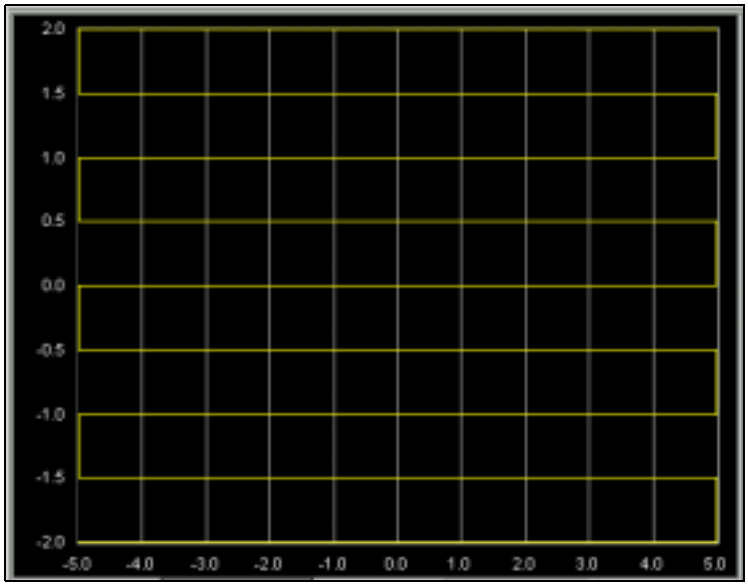
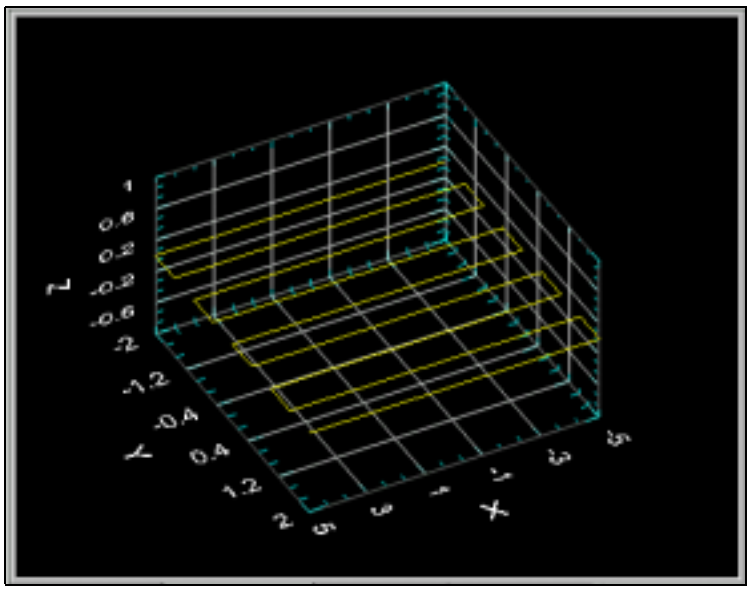


Figure 16.14: Negative Raster 3-Dimension Graph



The 1-dimensino graph maps the X-axis against time, the 2-dimension maps X and Y and the 3-dimension maps X and Y with reference to Z.

16.2.4 Custom Example

Requirement

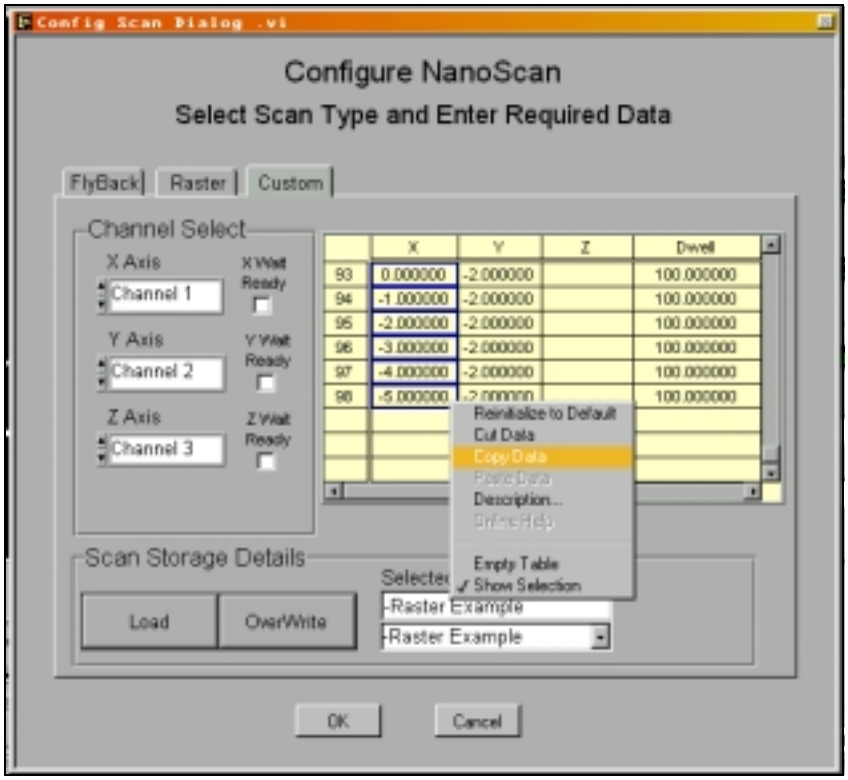
To copy the data from the -Raster Example and add Z-axis data to it. The X-axis will be copied to the Z-axis thus making a 3D Raster. The control mode will be command mode so there is no need to set the Wait for Ready check box.

The example shown below will be stored as Custom Example.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

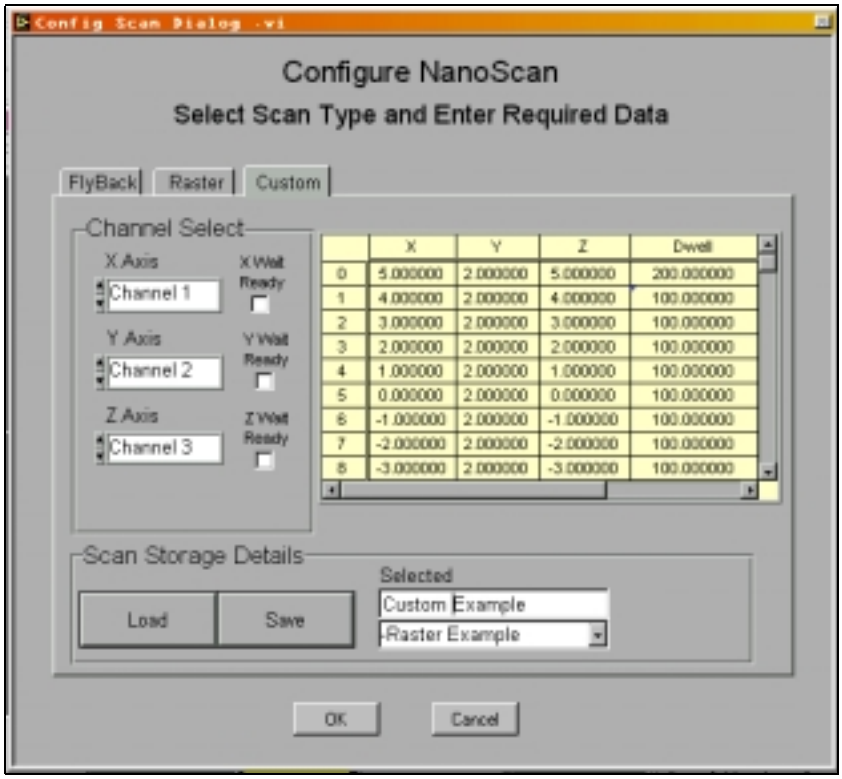
Settings

Figure 16.15: Custom Example Copying Data



The X-axis data is copied by using the mouse and dragging to highlight the data required. Right-click the mouse and a pop-up menu will appear, select Copy Data and then drag the area where the data is to be copied to. Right-click the mouse again and the pop-up menu will now have the option to Paste Data. Select this and check that the inserted data is correct.

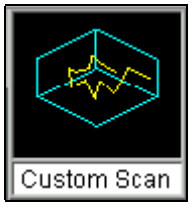
Figure 16.16: Custom Example Pasted Data



Note: It is important to ensure that the Pasted area is exactly the same dimensionally as the Copied area. Erroneous value will result otherwise.

When [OK] is pressed the Custom Icon will be displayed on the NanoScan front panel.

Figure 16.17: Custom Icon



Resultant Graphs

Figure 16.18: Custom 1-Dimension Graph

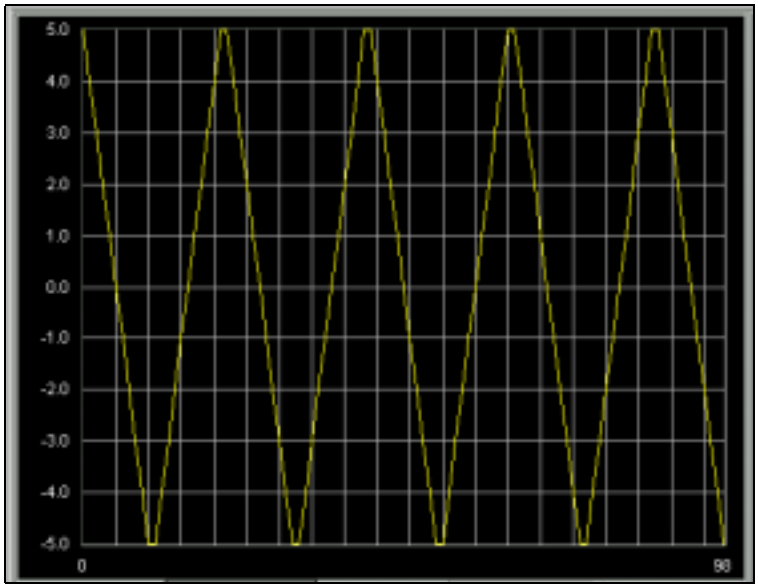


Figure 16.19: Custom 2-Dimension Graph

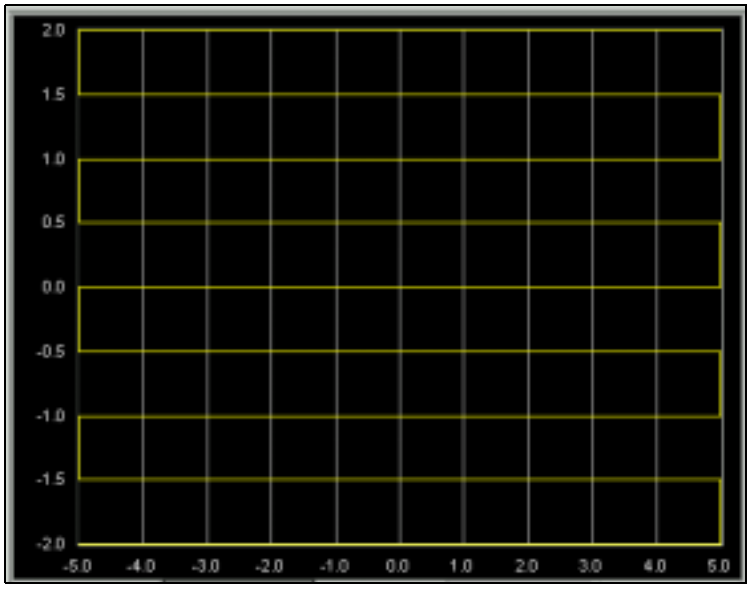
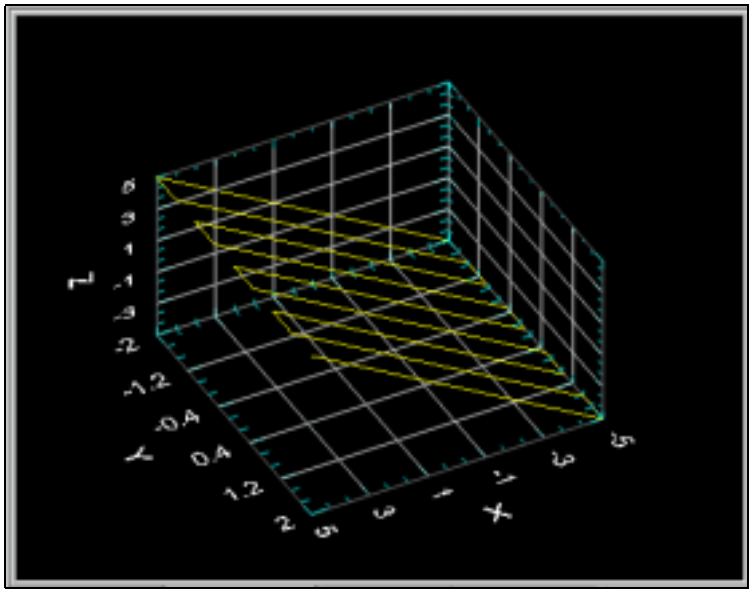
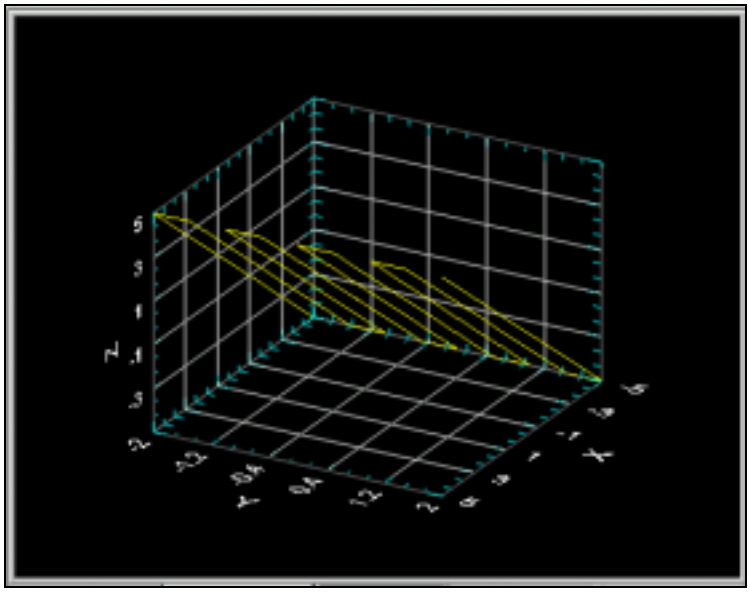


Figure 16.20: Custom 3-Dimension Graph



The 1-dimension graph maps the X-axis against time, the 2-dimension maps X and Y and the 3-dimension maps X and Y with reference to Z.

Figure 16.21: Custom Alternative View



Use the mouse to drag the 3-dimension graph to present alternative views.

16.3 Running a Scan

Configure a scan as in section 16.2.

Select the desired Graphing display Mode.

Figure 16.22: Measured Mode



In ***Measured Mode*** the command position is sent to the controller, dwell time waited and the measured position returned. The measured position in then updated and displayed on the graph. This operating mode is significantly slower than Command Mode.

Figure 16.23: Commanded Mode



In Commanded Mode the command position is sent to the controller, the dwell time is waited and the next position command is then sent to the controller.

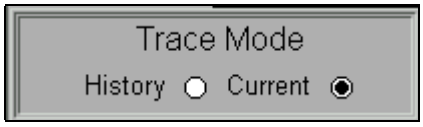
Select Trace Mode

Figure 16.24: History Mode



In History Mode all the previous positions are line plotted on the graph. The Graph can be in Autoscale or Manual scale mode. The graphs in the examples are in History Mode.

Figure 16.25: Current Mode



In Current Mode only the latest position is marked on the graph. The Graph will be in Manual scale mode. Figures 16.26 and 16.27 indicate the expected graphs.

Figure 16.26: 2-Dimension Trace Mode

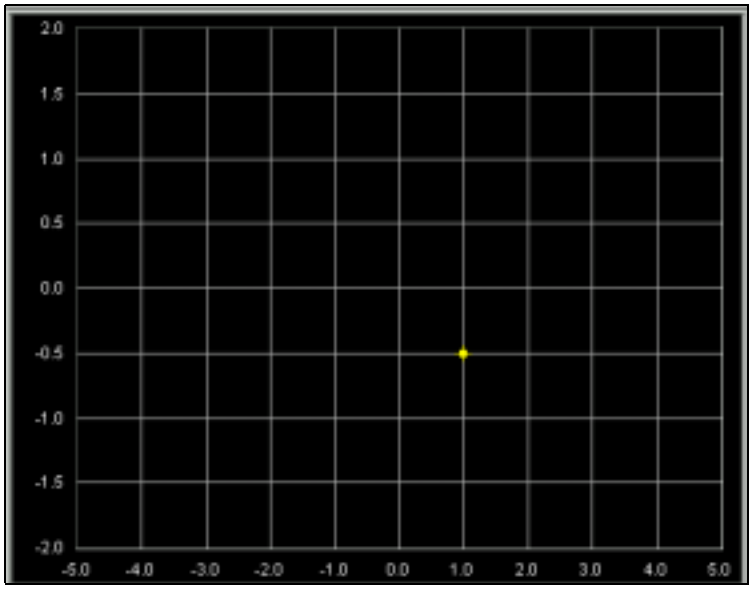
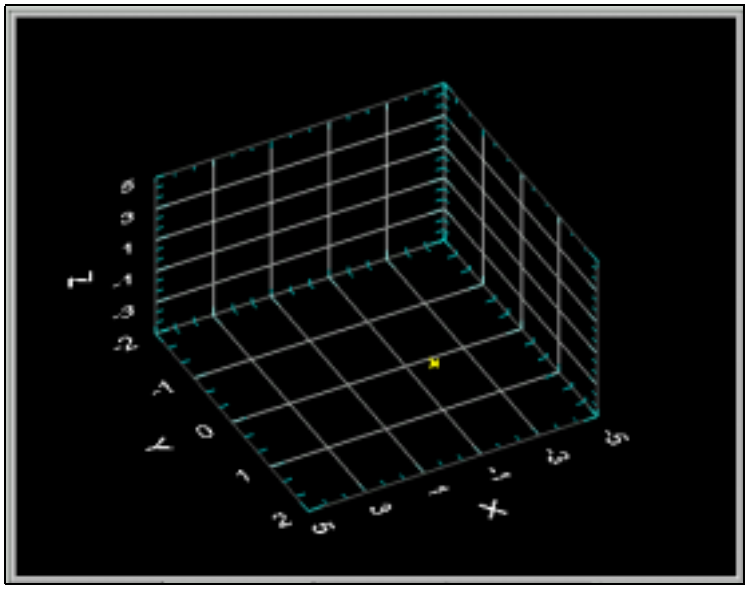


Figure 16.27: 3-Dimension Trace Mode



Check Graph Settings in the System Set-up Dialog (see section 16.4.2)

Press [Start] to start the scan

To pause a scan press [Stop] - The Scan can be Reset or restarted from its stopped position by pressing [Reset] or [Start].

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

16.4 Changing Default Settings

16.4.1 Scan Data File Path

Figure 16.28: Set-up Scan Data File Path



Change this to change the location and name of the scans. This can also be used to set up personalised Scan files.

16.4.2 Graph Settings

Figure 16.29: Set-up Graphs



Use this mode to set the default graph axis minimum and maximum values for when the Manual Scaling mode is selected. Select the value that needs updating and update. The Graph Scaling mode can also be set here, select the required mode (Autoscale or Manual).
Press [OK] to accept the changes or [Cancel] to ignore changes.

Note: Manual Scaling always applied when Current History Trace mode is chosen on the NanoScan front panel.

16.4.3 Hardware

Figure 16.30: Set-up Hardware



The hardware Tab is used to set the default controller interface settings.

16.4.4 INI Path

Figure 16.31: Set-up INI Path



This is the file path where all the system set-up data is stored.

16.4.5 Chan Set up

Figure 16.32: Set-up Channels



The controller can have its ready limit and its units set using this Tab option. This data is not saved because it is stored within the controllers' EPROM memory.

To change the Axis Ready Limit select the channel to change and type new number.

To change the units select required units from the drop down menu.

Press [Set] to send commands to controller.

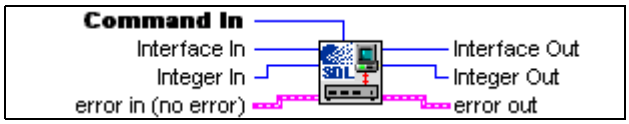
16.5 Quitting Program

It is good practice to quit the program in a controlled manner. Use the [Exit] button on the NanoScan front panel for this purpose.

17 Hardware Drivers

17.1 Hardware Driver Details

17.1.1 NPS3XXX Interface



Queensgate Limited
[Ver 1V0]
PART NUMBER : NGC-3032-S
TITLE : LabVIEW Drivers for NPS3000 Controllers
AUTHOR : SSDC
USED ON : NPS3000 series controllers

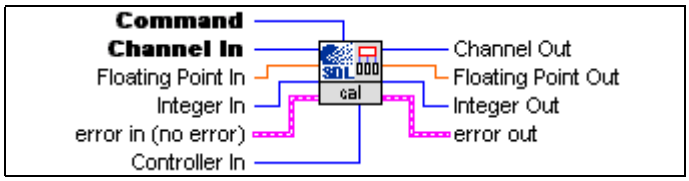
DESCRIPTION
This will set up the interface between the computer and controller.

COMMANDS
No Command - Error : Logs or throws error
Get Interface : Returns Interface on Interface Out
Set Interface : Set Interface to Interface In
Get RS232 Baud Rate : Returns value on Integer Out
Set RS232 Baud Rate : Sets value to Integer In
Get Data Format : Returns value on Integer Out
Set Data Format : Sets value to Integer In
Get Sync Period : Returns value on Integer Out
Set Sync Period : Sets value to Integer In
RestartCommPort : resets CommPort to Interface In
RestartCommRate : resets Comm Port Baud Rate to Integer In

ERROR HANDLING
Logs or throws error on No Command. Log or Throws errors returned from Controller.

VERSION CONTROL
1V0 - New Driver SSDC 4-July-2000

17.1.2 NPS3XXX Calibration



Queensgate Limited
[Ver 1V0]
PART NUMBER : NGC-3032-S
TITLE : LabVIEW Drivers for NPS3000 Controllers
AUTHOR : SSDC
USED ON : NPS3000 series controllers

DESCRIPTION
Sets and returns Stage Calibration Information for a selected Stage, Axis or Channel.

COMMANDS
No Command - Error : Logs or Throws Hardware Error
Get Yaw Rate : Returns Reading for controller and channel on Floating Point Out
Set Yaw Rate : Sets Value for controller and channel from Floating Point In
Get Lever Arm Length : Returns Reading from controller and channel on Floating Point Out
Set Lever Arm Length : Sets Value for controller and channel from Floating Point In
Get Lin Coeff b0 : Returns Reading from controller and channel on Floating Point Out
Set Lin Coeff b0 : Sets Value for controller and channel from Floating Point In
Get Lin Coeff b1 : Returns Reading from controller and channel on Floating Point Out
Set Lin Coeff b1 : Sets Value for controller and channel from Floating Point In
Get Lin Coeff b2 : Returns Reading from controller and channel on Floating Point Out
Set Lin Coeff b2 : Sets Value for controller and channel from Floating Point In
Get Lin Coeff b3 : Returns Reading from controller and channel on Floating Point Out
Set Lin Coeff b3 : Sets Value for controller and channel from Floating Point In
Get Lin Coeff b4 : Returns Reading from controller and channel on Floating Point Out
Set Lin Coeff b4 : Sets Value for controller and channel from Floating Point In
Get Charge Amp Zero Offset : Returns Reading from controller and channel on Floating Point Out & Integer Out
Set Charge Amp Zero Offset : Sets Value for controller and channel from Integer In
Get Charge Amp Range : Returns Reading from controller and channel on Floating Point Out & Integer Out
Set Charge Amp Range : Sets Value for controller and channel from Integer In
Get Charge Amp Coarse Gain : Returns Reading from controller and channel on Floating Point Out & Integer Out
Set Charge Amp Coarse Gain : Sets Value for controller and channel from Integer In
Get Charge Amp Fine Gain : Returns Reading from controller and channel on Floating Point Out & Integer Out
Set Charge Amp Fine Gain : Sets Value for controller and channel from Integer In
Get Sensor Scale Factor : Returns Reading from controller and channel on Floating Point Out & Integer Out
Set Sensor Scale Factor : Sets Value for controller and channel from Integer In
Get Sensor Scale Offset : Returns Reading from controller and channel on Floating Point Out
Set Sensor Scale Offset : Sets Value for controller and channel from Floating Point In
Get Actuator Scale Factor : Returns Reading from controller and channel on Floating Point Out
Set Actuator Scale Factor : Sets Value for controller and channel from Floating Point In
Get Actuator Scale Offset : Returns Reading from controller and channel on Floating Point Out
Set Actuator Scale Offset : Sets Value for controller and channel from Floating Point In
Get Anlg Input Scale Factor : Returns Reading from controller and channel on Floating Point Out
Set Anlg Input Scale Factor : Sets Value for controller and channel from Floating Point In
Get Anlg Input Scale Offset : Returns Reading from controller and channel on Floating Point Out
Set Anlg Input Scale Offset : Sets Value for controller and channel from Floating Point In
Get Zero Offset Corr Factor : Not Implemented
Set Zero Offset Corr Factor : Not Implemented

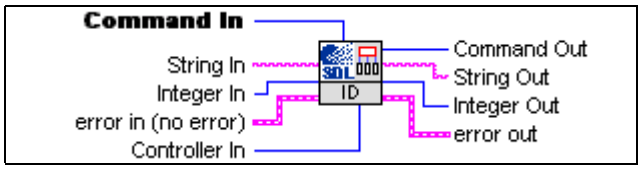
Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Get Range Corr Factor : Not Implemented
Set Range Corr Factor : Not Implemented
Get Coarse Gain Corr Factor : Not Implemented
Set Coarse Gain Corr Factor : Not Implemented
Get Fine Gain Corr Factor : Not Implemented
Set Fine Gain Corr Factor : Not Implemented

ERROR HANDLING
Logs or throws error on No Command. Log or Throws errors returned from Controller.

VERSION CONTROL
1V0 - New Driver SSDC 4-July-2000

17.1.3 NPS3XXX Controller ID



Queensgate Limited
[Ver 1V0]
PART NUMBER : NGC-3032-S
TITLE : LabVIEW Drivers for NPS3000 Controllers
AUTHOR : SSDC
USED ON : NPS3000 series controllers

DESCRIPTION
This sets and returns all Controller ID information. The various parameters will be parsed so that requested data will be in a readable form.
The Controller offers command-locking facilities. The controller handles the locking status. If a command is accessed that is not available in the current locking status then the controller will send back an error message. This will be interpreted by the VI and handled appropriately.

Security Levels:

- User Lock
  - This facility allows you to lock the controller so that the most basic commands can be used.
- Super User Lock
  - This level is provided to prevent access to the 'super user' commands'.
- QI Lock
  - Prevents access to the 'QI commands'.

Please refer to the command language reference to the security level of each command. This will also verify which commands will be read-only.

The returned dates will be in DD MMM YYYY format for the String Out and Days since Jan 1st 1900 for the Integer Out. If the Set Date option is chosen and both String In and Integer In are filled, String In will take precedence.

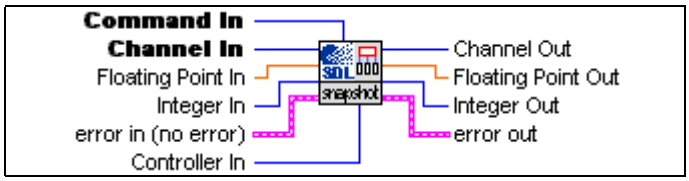
COMMANDS
No Command - Error : Logs or Throws Hardware Error
Get HW Part Number : Returns on String Out
Set HW Part Number : Part Number on String In must be in aaa-0000-a format (NPS-3111-S)
Get Serial Number : Returns on String Out & Integer Out
Set Serial Number : Sets value to Integer In
Get Options : Returns on String Out & Integer Out
Set Options : Set Integer In to number specified in Command Language Documents
Get Interface : Returns on String Out
Set Interface : Set Integer In to number specified in Command Language Documents
Get HW Manufacture Date : Returns on String Out (Formatted) & Integer Out (Days since 30 Dec 1899)
Set HW Manufacture Date : Date on String In must be in format DD MMM YYYY (02 Jul 2000, 25 Sep 1998, valid months are Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec)
Get Calibration Date : Returns on String Out (Formatted) & Integer Out (Days since 30 Dec 1899)
Set Calibration Date : See Set HW Manufacture Date
Get SW Part Number : Returns on String Out
Get Version Number : Returns on String Out
Get SW Release Date : Returns on String Out (Formatted) & Integer Out (Days since 30 Dec 1899)
Get Data Field Index : Returns on String Out & Integer Out
Set Data Field Index : Sets value to Integer In

ERROR HANDLING
Logs or throws error on No Command. Log or Throws errors returned from Controller.

VERSION CONTROL
1V0 - New Driver SSDC 4-July-2000

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

17.1.4 NPS3XXX Controller Snapshot.vi



Queensgate Limited
[Ver 1V0]
PART NUMBER : NGC-3032-S
TITLE : LabVIEW Drivers for NPS3000 Controllers
AUTHOR : SSDC
USED ON : NPS3000 series controllers

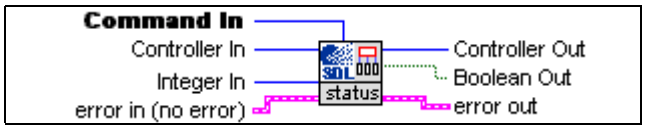
DESCRIPTION
Sets, reads and controls all snapshot functions for a selected Stage, Axis or Channel.

COMMANDS
No Command - Error : Logs or Throws Hardware Error
Get Stimulus Chan Number : Returns value for Controller and Channel on Integer Out
Set Stimulus Chan Number : Sets value to Channel In
Get Stimulus Amplitude : Returns value for Controller and Channel on Floating Point Out
Set Stimulus Amplitude : Sets value to Floating Point In
Get Stimulus Leading Edge : Returns value for Controller and Channel on Integer Out
Set Stimulus Leading Edge : Sets value to Integer In in system cycles
Get Stimulus Trailing Edge : Returns value for Controller and Channel on Integer Out
Set Stimulus Trailing Edge : Sets value to Integer In in system cycles
Get Response Chan Number : Returns value for Controller and Channel on Integer Out
Set Response Chan Number : Sets value to Integer In
Get Response Capture Time : Returns value for Controller and Channel on Integer Out
Set Response Capture Time : Sets value to Integer In in system cycles
Get Response Data : Returns indexed snapshot data on Floating Point Out where Integer In is the index.
Fire Snapshot : Starts the snapshot

ERROR HANDLING
Logs or throws error on No Command. Log or Throws errors returned from Controller.

VERSION CONTROL
1V0 - New Driver SSDC 4-July-2000

17.1.5 NPS3XXX Controller Status



Queensgate Limited
[Ver 1V0]
PART NUMBER : NGC-3032-S
TITLE : LabVIEW Drivers for NPS3000 Controllers
AUTHOR : SSDC
USED ON : NPS3000 series controllers

DESCRIPTION
Controls Controller and returns Controller status.
Status bits are parsed so that they return a boolean Yes or No to a question. This helps block diagrams self document.
The Controller offers command-locking facilities.
The controller handles the locking status. If a command is accessed that is not available in the current locking status then the controller will send back an error message. This will be interpreted by the VI and handled appropriately.

Security Levels:

User Lock
This facility allows you to lock the controller so that the most basic commands can be used.
Super User Lock
This level is provided to prevent access to the 'super user' commands'.
QI Lock
Prevents access to the 'QI commands'.

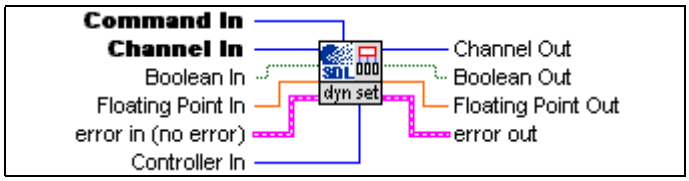
Please refer to the command language reference to the security level of each command. This will also verify which commands will be read-only.

COMMANDS
No Command - Error : Logs or Throws Hardware Error
Set Controller Type : 0 (Master) or 1 (Slave) on Integer In
Reset Controller : Performs a software reset
Store System Configuration : Stores System Status
QI Locked? : Returned on Boolean Out
Super-User Locked? : Returned on Boolean Out
User Locked? : Returned on Boolean Out
Channel 1 Ready? : Returned on Boolean Out
Channel 2 Ready? : Returned on Boolean Out
Channel 3 Ready? : Returned on Boolean Out
Recall STD Controller Config : Not implemented
Set Controller Lock Code : Uses Integer In to set a new user lock code (0 disables user lock facility)
Lock Controller : Locks the controller if a user lock has been defined
Unlock Controller : Uses Integer In to unlock the controller

ERROR HANDLING
Logs or throws error on No Command. Log or Throws errors returned from Controller.

VERSION CONTROL
1V0 - New Driver SSDC 4-July-2000

17.1.6 NPS3XXX Dynamic Setup



Queensgate Limited
[Ver 1V0]
PART NUMBER : NGC-3032-S
TITLE : LabVIEW Drivers for NPS3000 Controllers
AUTHOR : SSDC
USED ON : NPS3000 series controllers

DESCRIPTION
This sets and returns Stage Setting information for a selected Stage, Axis or Channel.

COMMANDS
No Command - Error : Logs or throws error
Get Unit Conversion Factor : Returns Reading from controller and channel on Floating Point Out
Set Unit Conversion Factor : Sets Value for controller and channel from Floating Point In
Get Unit Conversion Offset : Returns Reading from controller and channel on Floating Point Out
Set Unit Conversion Offset : Sets Value for controller and channel from Floating Point In
Get Ready Limit F : Returns Reading from controller and channel on Floating Point Out
Set Ready Limit F : Sets Value for controller and channel from Floating Point In
Get Ready Limit INT : Returns Reading from controller and channel on Floating Point Out
Set Ready Limit INT : Sets Value for controller and channel from Floating Point In
Get Data Sampling Time : Returns Reading from controller and channel on Floating Point Out
Get Integrator Time Const : Returns Reading from controller and channel on Floating Point Out
Set Integrator Time Const : Sets Value for controller and channel from Floating Point In
Get Differentiator Time Const : Returns Reading from controller and channel on Floating Point Out
Set Differentiator Time Const : Sets Value for controller and channel from Floating Point In
Get Tracking Time Const : Returns Reading from controller and channel on Floating Point Out
Set Tracking Time Const : Sets Value for controller and channel from Floating Point In
Get Prop Gain : Returns Reading from controller and channel on Floating Point Out
Set Prop Gain : Sets Value for controller and channel from Floating Point In
Get Differential Gain : Returns Reading from controller and channel on Floating Point Out
Set Differential Gain : Sets Value for controller and channel from Floating Point In
Get Setpoint Weighting : Returns Reading from controller and channel on Floating Point Out
Set Setpoint Weighting : Sets Value for controller and channel from Floating Point In
Get Max Int Error : Returns Reading from controller and channel on Floating Point Out
Set Max Int Error : Sets Value for controller and channel from Floating Point In
Get Min Int Error : Returns Reading from controller and channel on Floating Point Out
Set Min Int Error : Sets Value for controller and channel from Floating Point In
Stage Ready? : Returns Reading from controller and channel on Boolean Out & Floating Point Out
Disable Channel? : Returns Reading from controller and channel on Boolean Out & Floating Point Out
Closed Loop? : Returns Reading from controller and channel on Boolean Out & Floating Point Out
Invert Phase? : Returns Reading from controller and channel on Boolean Out & Floating Point Out
Freeze Output? : Returns Reading from controller and channel on Boolean Out & Floating Point Out
Snapshot Stimulus? : Returns Reading from controller and channel on Boolean Out & Floating Point Out
Snapshot Response? : Returns Reading from controller and channel on Boolean Out & Floating Point Out
Analogue Input? : Returns Reading from controller and channel on Boolean Out & Floating Point Out
Set Stage Ready : Sets Value for controller and channel from Boolean In
Set Disable Channel : Sets Value for controller and channel from Boolean In
Set Closed Loop : Sets Value for controller and channel from Boolean In

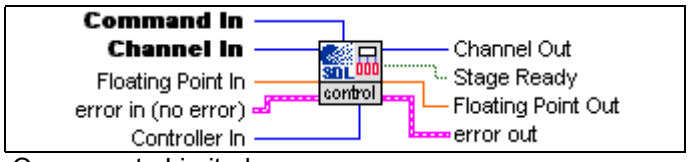
Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Set Invert Phase : Sets Value for controller and channel from Boolean In
Set Freeze Output : Sets Value for controller and channel from Boolean In
Set Analogue Input : Sets Value for controller and channel from Boolean In

ERROR HANDLING
Logs or throws error on No Command. Log or Throws errors returned from Controller.

VERSION CONTROL
1V0 - New Driver SSDC 4-July-2000

17.1.7 NPS3XXX Stage Control



Queensgate Limited
[Ver 1V0]
PART NUMBER : NGC-3032-S
TITLE : LabVIEW Drivers for NPS3000 Controllers
AUTHOR : SSDC
USED ON : NPS3000 series controllers

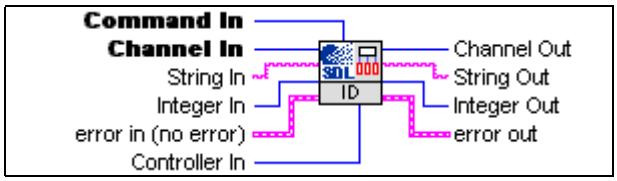
DESCRIPTION
Controls stage and returns stage status for a selected stage, axis or channel.
Status bits are parsed so that they return a boolean Yes/No.

COMMANDS
No Command - Error : Throws or logs error depending on Error Control Setting.
Get Command Position F : Returns value on Floating Point Out
Set Command Position F : Sets value dependant on Floating Point In
Get Command Position INT : Returns value on Floating Point Out (Convert to INT)
Set Command Position INT : Sets value dependant on Floating Point In (Convert to INT)
Get Measured Position F : Returns value on Floating Point Out
Get Measured Position INT : Returns value on Floating Point Out (Convert to INT)
Store Dynamic Setup : Store setup in position specified by Floating Point In
Recall Dynamic Setup : Store setup in position specified by Floating Point In
Recall Std Channel Config : Not Implemented
Store Stage Config : Stores Stage Config for selected channel
Wait For Stage Ready : When Ready indicate on Stage Ready (False = Timeout)
Get Default Setup Number : Returns on Floating Point Out (Convert to INT)
Set Default Setup Number : Uses value on Floating Point In (Convert to INT)
Get Current Setup Number : Returns on Floating Point Out (Convert to INT)

ERROR HANDLING
Default conditions Throw or log and error depending on setting of error control.
All interactions with NPS3XXX are error trapped and will be thrown or logged depending on setting of error control.
Other Errors ignored and passed through.

VERSION CONTROL
Ver 1V0 SSDC 12/05/2000

17.1.8 NPS3XXX Stage ID



Queensgate Limited
[Ver 1V0]
PART NUMBER : NGC-3032-S
TITLE : LabVIEW Drivers for NPS3000 Controllers
AUTHOR : SSDC
USED ON : NPS3000 series controllers

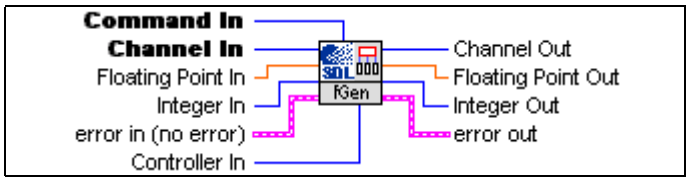
DESCRIPTION
This sets and returns all Stage ID information for a selected Stage, Axis or Channel. The various parameters are parsed so that requested data will be in a readable form.

COMMANDS
No Command - Error : Logs or Throws Hardware Error
Get Part Number : Returns value on String Out
Set Part Number : Part Number on String In must be in aaa-XYZ-000a format (NPS-XY-100A)
Get Serial Number : Returns value on String Out & Integer Out
Set Serial Number : Sets value to Integer In
Get Sensor Configuration : Returns value on String Out & Integer Out
Set Sensor Configuration : On String In set Sensor ID to XYZ for linear axes or ASCII 79 (y), 71 (q) , 66 (f) for the radial axes.
Get Actuator Configuration : Returns value on String Out & Integer Out
Set Actuator Configuration : As Set Sensor Configuration
Get Range : Returns value on String Out & Integer Out
Get HW Manufacture Date : Returns value on String Out & Integer Out
Set HW Manufacture Date : Date on String In must be in format DD MMM YYYY (02 Jul 2000, 25 Sep 1998, valid months are Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec)
Get Calibration Date : Returns value on String Out & Integer Out
Set Calibration Date : See Set HW Manufacture Date
Get Test Controller : Returns value on String Out & Integer Out
Set Test Controller : Sets value of Test Controller serial number to Integer In
Get Test Channel : Returns value on String Out & Integer Out
Set Test Channel : Sets value to Integer In (1,2 or 3)
Get Number Of Axes : Returns value on String Out & Integer Out

ERROR HANDLING
Logs or throws error on No Command. Log or Throws errors returned from Controller.

VERSION CONTROL
1V0 - New Driver SSDC 4-July-2000

17.1.9 NPS3XXX Controller Function Generator



Queensgate Limited
[Ver 1V0]
PART NUMBER : NGC-3032-S
TITLE : LabVIEW Drivers for NPS3000 Controllers
AUTHOR : SSDC
USED ON : NPS3000 series controllers

DESCRIPTION
Sets, reads and controls all Function Generator functions.

COMMANDS
No Command - Error : Logs or Throws Hardware Error
Get Function Gen Mode : Returns Reading for controller and channel on Integer Out
Set Function Gen Mode : Sets Value for controller and channel from Integer In
Get Waveform type : Returns Reading for controller and channel on Integer Out
Set Waveform type : Sets Value for controller and channel from Integer In
Get Waveform Period : Returns Reading for controller and channel on Floating Point Out
Set Waveform Period : Sets Value for controller and channel from Floating Point In
Get Waveform Scale Factor : Returns Reading for controller and channel on Floating Point Out
Set Waveform Scale Factor: Sets Value for controller and channel from Floating Point In
Get Waveform Scale Offset : Returns Reading for controller and channel on Floating Point Out
Set Waveform Scale Offset: Sets Value for controller and channel from Floating Point In
Get Waveform Data : Returns Reading for controller and channel on Integer Out
Set Waveform Data : Sets Value for controller and channel from Integer In
Store Waveform Data : Stores Waveform for controller and channel from Integer In

ERROR HANDLING
Logs or throws error on No Command. Log or Throws errors returned from Controller.

VERSION CONTROL
1V0 - New Driver SSDC 4-July-2000

17.1.10 NPS3XXX Error



Queensgate Limited
[Ver 1V0]
PART NUMBER : NGC-3032-S
TITLE : LabVIEW Drivers for NPS3000 Controllers
AUTHOR : SSDC
USED ON : NPS3000 series controllers

DESCRIPTION
This will convert the number out to an error code and message based on the error codes given in document NGC-3003-D.

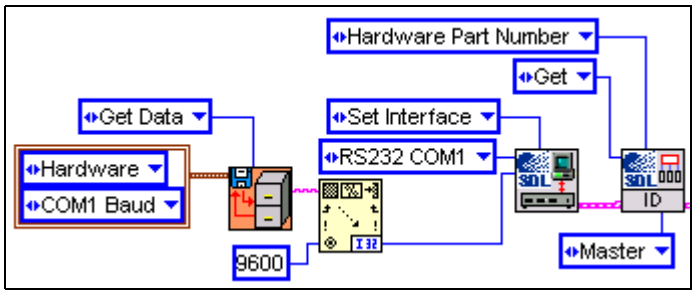
ERROR HANDLING
Traps and stores any errors

VERSION CONTROL
Ver 1V0 ssdc 20/06/2000

17.2 Usage Examples

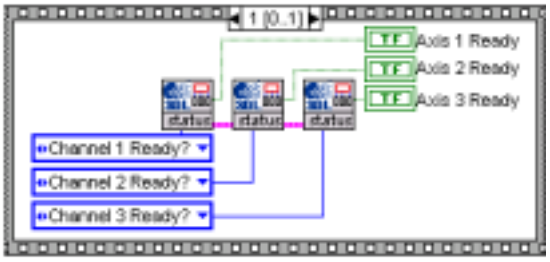
The first Operation should be to set up the interface

Figure 17.1: Set-up Driver



Next it is simply a matter of selecting the Driver to complete the operation and setting up the commands as required. The example below retrieves the Controller Ready Status and sets the display up accordingly.

Figure 17.2: Use Driver



18 Software Examples

18.1 Introduction

This Chapter provides examples of the Controller embedded in user application software.

The following software examples illustrate how to use:

- NGCMOD32.dll with Windows SDK.
- NGCMOD32.dll with Visual C++4.0 and MFC.
- OLE control NGC_OCX.ocx with Visual Basic 4.0.

The code for the following examples is available on the Software Sample Disk supplied.

18.2 Example 1: Using the command dll *NGCMOD32.dll* with Windows SDK

The following example code makes use of the Win API function GetProcAddress.

Since the function names are used to extract the addresses of functions within the dll we do not need to include the header file containing functions declarations.

1. **Define the names of the functions**, within ngcmod32.dll, to be used:

```
#define INITCOMM                              "InitComm"

#define READ_CHARGEAMPZOFFSET                 "Read_Chargeamp_Zoffset"
```

2. **Define the function type pointers**:

```
typedef int (FAR PASCAL *lpfnNGC)(LPSTR port, LPSTR baud, LPSTR parity, LPSTR
databits, LPSTR stopbits, int FAR* error);

typedef int (FAR PASCAL *lpfnRead_Chargeamp_Zoffset)(int far *controller,int far
*channel, long far *dest);
```

3. **To start communications** with the command dll call the InitComm function with default transmission values. You must do this at the very start of any program that intends to call controller functions;

```
strcpy(port,"COM1");

strcpy(baud,"9600");

strcpy(parity,"n");

strcpy(databits,"8");

strcpy(stopbits,"1");



/* Load NGC Command DLL */

hLib = LoadLibrary(LIB);

status = (int)hLib;

if((int)hLib > 32)

{
```

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
fptr = (lpfnNGC)GetProcAddress(hLib,INITCOMM);
 if(fptr != NULL)
{
 status = (*fptr)((LPSTR)port,(LPSTR)baud,(LPSTR)parity,(LPSTR)databits,
(LPSTR)stopbits,&error);
 if(status == 0)
           { ActiveStatus = TRUE; } /* NGC active */
}
```

4. **Declare** pointer to function:

```
lpfnRead_Chargeamp_Zoffset Zfptr;
```

5. **Extract** the address of the required function and call it.:

```
Zfptr = (lpfnRead_Chargeamp_Zoffset)GetProcAddress(hLib,READ_CHARGEAMPZOFFSET);
if(Zfptr != NULL)
{    status = (*Zfptr)((int far*)&controller,(int far*)&channel, (long
far*)&zoffset);
}
```

6. Here the value of Chargeamp Zero Offset is passed to the zoffset parameter.

Figure 18.1: Typical Dialog Box



18.3 Example 2: Using the command dll *NGCMOD32*.dll with Visual C++ 4.0 and MFC

1. **Place** `#include "ngcmod32.h"` within any file that calls command dll functions.
2. **Write a function** that uses *`InitComm`* to start communicating with controller:

```
//
// StartTransmission - example of how to initiate
// communications with an NPS3000 series controller.
//
BOOL CExample1Dlg::StartTransmission(void)
{
  char port[21];
  char baud[21];
  char parity[21];
  char databits[21];
```

```
 char stopbits[21];
 int status, error;

 strcpy(port,"COM1");
 strcpy(baud,"9600");
 strcpy(parity,"n");
 strcpy(databits,"8");
 strcpy(stopbits,"1");

 status = InitComm(port, baud, parity, databits, stopbits, (int far*) &error);
 if(status != 0) return FALSE;

 return TRUE;
}
```

3. **Use command dll functions** to extract data from the controller. Here we are using ***Read_Chargeamp_Zoffset*** to extract Chargeamp data from the controller:

```
//    LoadCalibrationParameters - example of how to call
//    NPS3000 series controller functions.
//
void CExample1Dlg::LoadCalibrationParameters(void)
{
     int controller = 0;
     int channel = 2;
     long dest;
     float fdest;
     char buff[21];

     Read_Chargeamp_Zoffset((int far*)&controller, (int far*)&channel, (long far*)&dest);
     sprintf(buff,"%ld",dest);
     GetDlgItem(IDC_ZERO)->SetWindowText(buff);
}
```

4. **Place function declarations** within window / dialog class:

```
class CExample1Dlg : public CDialog
{
       .
       .

// Implementation
protected:

       void LoadCalibrationParameters(void);
       BOOL StartTransmission(void);
       .
       .
};
```

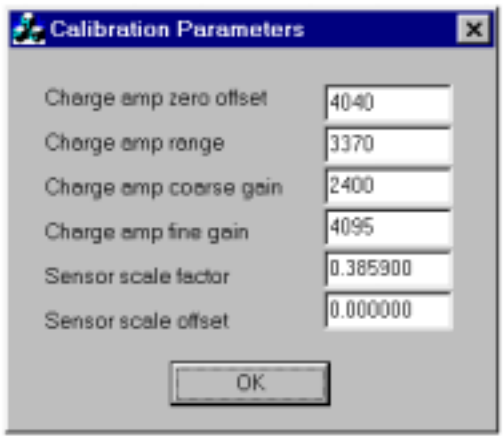
5. **Start communications** with controller at program start-up:

```
BOOL CExample1Dlg::OnInitDialog()
{
        CDialog::OnInitDialog();

        //  Start communicating with controller
        if(StartTransmission() == TRUE)
        {
        //Interrogate calibration parameters
        LoadCalibrationParameters();
        }
        return TRUE;
}
```

6. **Place *ngcmod32.lib*** within your project file list:

Figure 18.2: Typical Dialog Box



18.4 Example 3: Using the OLE control ***NGC_OCX.ocx.*** with Visual Basic 4.0

The following example demonstrates how to use ***NGC_OCX*** within the Visual Basic 4.0 development environment.

1. **To add NGC_OCX** to your project, go to tools, custom controls, browse and select NGC_OCX.ocx. This will load the control into the controls list. To include the control within your project click the checkbox to the left of the control name.
2. **Drag the control onto a form.** The control will immediately try to communicate with the controller. Assuming that a controller is hooked up and running you can change controller parameters both at design time and run time. Changing property values within the property box automatically changes the appropriate controller parameters.

This code snippet fills a list box with controller Linearisation and Calibration parameters.

```
Private Sub Form_Load()

Text1.Text = NGC_OCX1.Controller
Text2.Text = NGC_OCX1.Channel

List1.AddItem "Linearisation B0" + Str(NGC_OCX1.B0)
List1.AddItem "Linearisation B1" + Str(NGC_OCX1.B1)
```

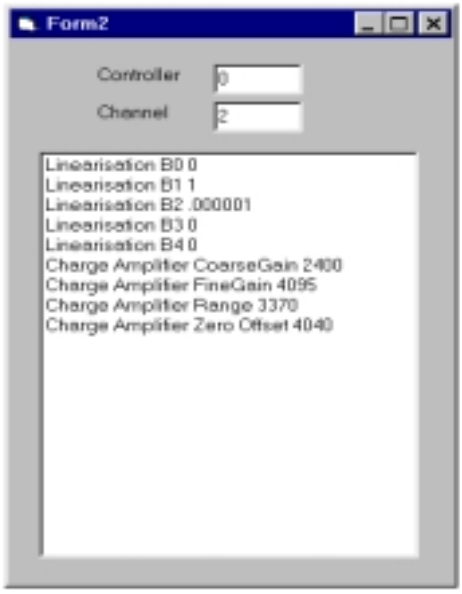
Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

```
List1.AddItem "Linearisation B2" + Str(NGC_OCX1.B2)
List1.AddItem "Linearisation B3" + Str(NGC_OCX1.B3)
List1.AddItem "Linearisation B4" + Str(NGC_OCX1.B4)

List1.AddItem"ChargeAmplifierCoarseGain"+Str(NGC_OCX1.ChargeAmplifierCoarseGain)
List1.AddItem "Charge Amplifier FineGain" + Str(NGC_OCX1.ChargeAmplifierFineGain)
List1.AddItem "Charge Amplifier Range" + Str(NGC_OCX1.ChargeAmplifierRange)
List1.AddItem"ChargeAmplifierZeroOffset" + Str(NGC_OCX1.ChargeAmplifierZeroOffset)

End Sub
```

Figure 18.3: Typical Dialog Box



19 Application Note 1: Setting up Custom NanoMechanism

19.1 Introduction

This Section is intended to allow users to set-up and configure Queensgate NPS3000-series Digital Controllers for use with NanoSensors® and actuators, to give closed loop operation.

19.2 Installation Procedure

1.0 Install the actuator into the system.
Clean and dry the actuators fixing interfaces with alcohol ☐ Tick
If a flat end piece or ball end piece is fitted ensure that this is tight ☐ Tick
Fit the actuator into the system using spring washers and the specified fixing torques ☐ Tick
(The specified fixing torque can be found on the installation drawing and should not be exceeded!!)

1. Set up the NanoSensors® and Actuator Mechanically

1.1 Mount the Probe NanoSensor® to the NanoMechanism or system to be measured.
Clean the front and back surfaces of the probe NanoSensor® with alcohol ☐ Tick
Clean the fixing face of the mating object with alcohol ☐ Tick
Attach the probe sensors to the moving object (to achieve the best operation the sensor should be mounted with fixing screws and spring washers for strain relief). Alternatively magnetic mountings or silicon rubber can be used but these methods are more susceptible to position drift over time ☐ Tick

1.2 Mount the Target NanoSensor® to the NanoMechanism or system to be measured.
Clean the front and back surfaces of the target NanoSensor® with alcohol ☐ Tick
Clean the fixing face of the reference object with alcohol ☐ Tick
Attach the target sensor to the reference block ☐ Tick
The reference block should have a low resonant frequency and ideally be rigidly attached to the test system to prevent motion of the reference sensor.

1.3 Connect the ADP-MTP/NX/NPS cable to the required channel of the NPS3000 controller. ☐ Tick
Connect the actuator cable to the SMB connector on the ADP-MTP/NX/NPS cable. ☐ Tick
Connect the probe and target NanoSensors® to the Lemo connectors on the ADP-MTP/NX/NPS cable, *ensure that the sensors are connected probe-probe, target-target, and that they are electrically isolated from each other.* ☐ Tick

2 Set-up the NPS3000

2.1 Plug the NPS3000 into the mains and the RS232 port of the computer. ☐ Tick

2.2 Switch the NPS3000 on. ☐ Tick

3. Set the Free State NanoSensor Gap

3.1 Run the Queensgate Instruments NanoControl Panel software. ☐ Tick

3.2 In the software, select the required NPS3000 channel. Command the actuator to 0 position (the midpoint of the HV range) by entering the command position into the white command box in the menu, or entering the value into the green display window ☐ Tick

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

3.3 Set up the nominal NanoSensor® gap.
Select the required nominal gap for the NanoSensor®. This should be between the specified (-L to -S) nominal gaps for the NanoSensor®. □ Tick
Clean a piece of electrically insulating film shim, either 20, 50, 100 or 500µm, with Alcohol to remove dust and debris. □ Tick
Place the film between the Target and Probe NanoSensors®. □ Tick
Move the Target sensor (mounted onto a reference block) closer to the Probe sensor (mounted on the mechanism until it lightly pinches the film). □ Tick
Rotate the sensor fixing block so that the film is pinched evenly across the sensor area.
If the system is to be set up required permanently then the reference block should be rigidly attached to the ground plate, to prevent drift over time. □ Tick
Remove the film from the NanoSensor® gap. □ Tick
The sensor nominal gap and parallelism is now set up

4. Unlock the NPS3000 into User mode

4.1 In the locking menu in the NanoControl Panel software, select 'Unlock' controller. □ Tick
Enter the system lock code, normally the NPS3000 system serial number unless the user has changed it. □ Tick
The controller is now unlocked in User mode. User mode allows system calibration and dynamic data to be stored to the controller EEPROM.

5. Set up the NPS3000 Charge Amplifiers

5.1 In the NanoControl Panel software select the NPS3000 channel that the NanoSensors® are connected to.

Set the following channel parameters:

Master / Slave - Master
Mode Menu - Set to Open loop
- Non freeze output
-Non invert phase

Channel Parameters Menu (Calibration Menu)
Press 'Apply' to download the values into the controller
- Sensor scale factor = 1
- Sensor scale offset = 0
- Actuator scale factor = 1
- Actuator scale offset = 0
- Charge amplifier zero offset = 0
- Charge amplifier range = 128
- Charge amplifier coarse gain = 0
- Charge amplifier fine gain = 4095

Channel Parameters Menu (Linearisation Menu)
- Linearisation Coefficient b0 = 0
- Linearisation Coefficient b1 = 1
- Linearisation Coefficient b2 = 0
- Linearisation Coefficient b3 = 0
- Linearisation Coefficient b4 = 0

Channel Parameters Menu (Units Menu)
- Microns (unit conversion factor = 1)

Channel Parameters Menu (Yaw Compensation Menu)
- Yaw rate = 0
- Lever arm length = 0 □ Tick

5.2 Command the actuator to its maximum positive position (+32767 system units). □ Tick

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

5.3 Note the **maximum** NanoSensor® read back position (the green window in the controller graphic).
Maximum =

5.4 Command the actuator to its minimum negative position (-32767 system units). □ Tick

5.5 Note the **minimum** NanoSensor® read back position.
Minimum =

5.6 Calculate the **average** read back position as (Maximum + Minimum) / 2.
Average =

5.7 Calculate:
Range 1 = 10,400,000 / **Maximum** read back position
Range 2 = 8,250,000 / **Average** read back position
Range 1 =
Range 2 =

5.8 Set the ***charge amplifier range*** using the NanoControl Panel software to the lower of **Range 1** or **Range 2**. If the lower of **Range 1** and **Range 2** is between 4095 and 4500 then set the **Range** to 4095.
Range =

5.9 If **Range 1** was used in section 5.8 then set the ***charge amplifier offset*** to 3900 * (Range 1 / Range 2)
If **Range 2** was used then set the ***charge amplifier offset*** to 3900.
If the ***change amplifier range*** had to be set to 4095 because **Range 1** and **Range 2** were greater than 4095, then set the ***charge amplifier offset*** to
3900 * 4095 / **Range 2**.
Offset =

5.A Command the actuator to its maximum positive position □ Tick

5.B Note the NanoSensor® read back position =

5.C Command the actuator to its minimum negative position □ Tick

5.D Note the NanoSensor® read back position =

5.E Adjust the ***Charge amplifier offset*** until the NanoSensor® read back positions are the same magnitude with the driven stage moving +/- maximum extension □ Tick

5.F Command the actuator to its maximum positive position □ Tick
Adjust the ***Charge amplifier offset coarse gain*** until the magnitude of the read back position is 26500+/-500. If the read back position is greater than 26500+/-500 with the ***Charge amplifier coarse gain*** set to 0, then reduce both the ***Charge amplifier offset*** and ***Charge amplifier range*** by the same ratio until it is 26500 +/-500. □ Tick
Check that the NanoSensor® read back position is the same magnitude with the actuator commanded to its minimum negative position □ Tick
If necessary adjust the ***charge amplifier offset*** until the system is balanced If the NanoSensor® read-back is inverted compared to the actuator command position then set the Mode to 'invert' □ Tick

5.G In the 'mode' menu set the mode to closed loop. □ Tick

6. Calibrate the Closed loop system using a known reference standard

6.1 Command the actuator to + 22000 system units
Record the measured position on the reference standard = **Position 1 (µm)**

Position 1=

6.2 Command the actuator to - 22000 system units
Record the measured position on the reference standard = **Position 2 (µm)**
Position 2=

6.3 Calculate (Position 1+ magnitude (Position 2)) = Measured range
Measured range (µm)=

6.4 Calculate...
NanoSensor® digital scale factor = **(measured range in microns / 44000) =** b_{x1}
b_{x1} **=**

6.5 In NanoControl Panel software select the 'Channel Parameters' menu ☐ Tick
Select the 'Calibration' menu ☐ Tick
Enter the value of b_{x1} **=** as the 'Sensor Scalefactor' ☐ Tick
Press 'Apply' and 'OK' ☐ Tick
The NanoSensors® will now be operating in units of Micrometers ☐ Tick

6.9 In the 'Channel parameters' menu select the 'units' menu. Set the units to the required operating units. ☐ Tick

7. Save the calibration data in the controller EEPROM

7.1 In the 'Configure' menu select the 'Store Channel Set-up, 0' ☐ Tick

7.2 In the 'Configure' menu select 'Save System Configuration', this will store the NanoSensor® calibration data into the controller. ☐ Tick

The NanoSensors® are now calibrated and ready for use. For further information about calibration or Linearisation, please contact Queensgate Ltd (see Section 23).

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

20 Application Note 2: Measurement of Resonant Frequency

20.1 Introduction

A limiting factor to the positioning speed of a NanoMechanism is the resonant frequency. Trying to move such devices too quickly can excite unstable resonances within the mechanism. This instability is often obvious as it is manifested audibly, but sometimes may also be measured as an increase in position noise.

Control of servo-loop PID parameters allows Queensgate or the user to achieve optimised performance of the NanoMechanism. Queensgate quotes stiffness, resonant frequency and bandwidth figures (maximum recommended BW) for our NanoMechanisms in our literature. We also routinely measure RF and BW during test and calibration. Typically we find that the 3dB BW should be around one-third to one-fifth of the RF for stable operation.

In many applications, the user operates the NanoMechanism within a larger mechanical frame. The mechanism may be carrying a large load, be experiencing an external force, or clamped rigidly to the external metrology frame. In these circumstances, the RF (and hence the BW) of the system can change significantly. It is important to be able to measure this parameter in situ. The purpose of this application note is to offer guidance to users wishing to make these measurements.

20.2 Calculation of RF and BW

The RF of a system is related to the moving mass and stiffness of that system by the following:

$$RF = \frac{1}{2\pi}\sqrt{\frac{k}{m}}$$

Where k is the stiffness of the system and m the moving mass. Given the stiffness and RF of the Queensgate Instruments stage, the user can calculate the moving mass of the stage (before any loaded mass). Armed with this information, the user can then re-calculate the RF of the mechanism for any added load. For example, take the NPS-Z-15B closed loop stage. The stiffness of the mechanism is $20N \bullet \mu m^{-1}$, and the RF (unloaded f_o) is 1800 Hz. From this we can calculate the moving mass in the unloaded stage is 156g. With an added load of 100g, the calculated RF ($f_{0.100}$) is around 1400 Hz.

20.3 Measurement

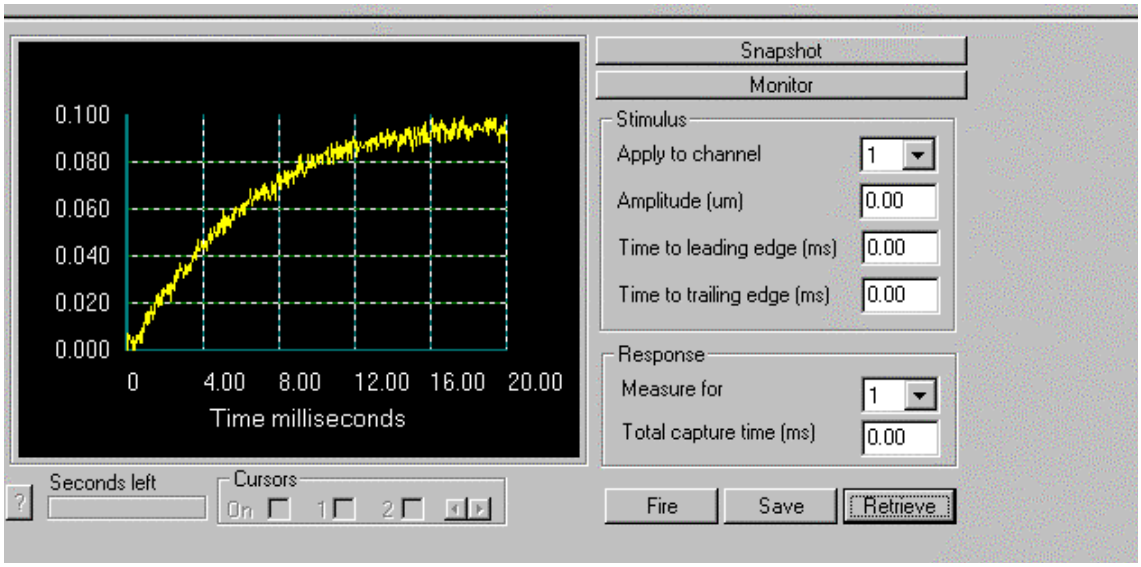
In addition to the calculation, the user also has the facility to make RF and BW directly in situ. This operation makes use of the snapshot facility within the virtual control panel software, supplied with the system. The snap shot mode allows the user to apply an impulse to the mechanism and monitor the response. Traditionally, this feature is used to optimise stage movement for a specific applied mass and step size. The impulse applied to the stage is a step function and the stage response (settle time etc) monitored:

If the stage is operated under open loop (i.e. outside servo control), this snapshot mode can be used to apply an impulse to the stage to force the stage to resonate. The snapshot mode is used to

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

apply an impulse (a quickly applied, high amplitude signal). A good analogy to use is like tapping a fine crystal wineglass, and listening to it ring.

Figure 20.1: Snapshot Window



The stage response to this impulse can be recorder and saved (using the save function in the virtual control panel). This data can be imported directly into a spreadsheet system for further analysis. A Fourier transform can be applied to the data to convert the time data to the frequency domain, from which the FR and BW can be measured.

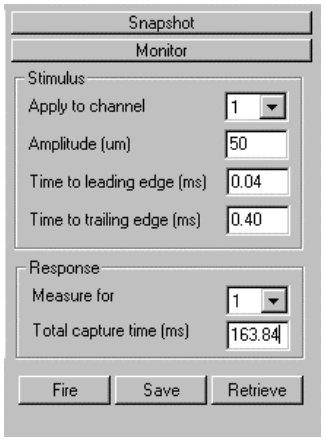
19.4 Example

For example, we are going to measure the RF and BW of the Queensgate NanoMechanism NPS-XY-100A, XY stage with typically 100µm travel in each axis.

Fasten the stage into its standard operating fixture, and apply the mass to be moved during the experiment. Activate the NPS3000-series Digital Controller and virtual control panel. Run the stage in open loop.

Apply an impulse, for example:

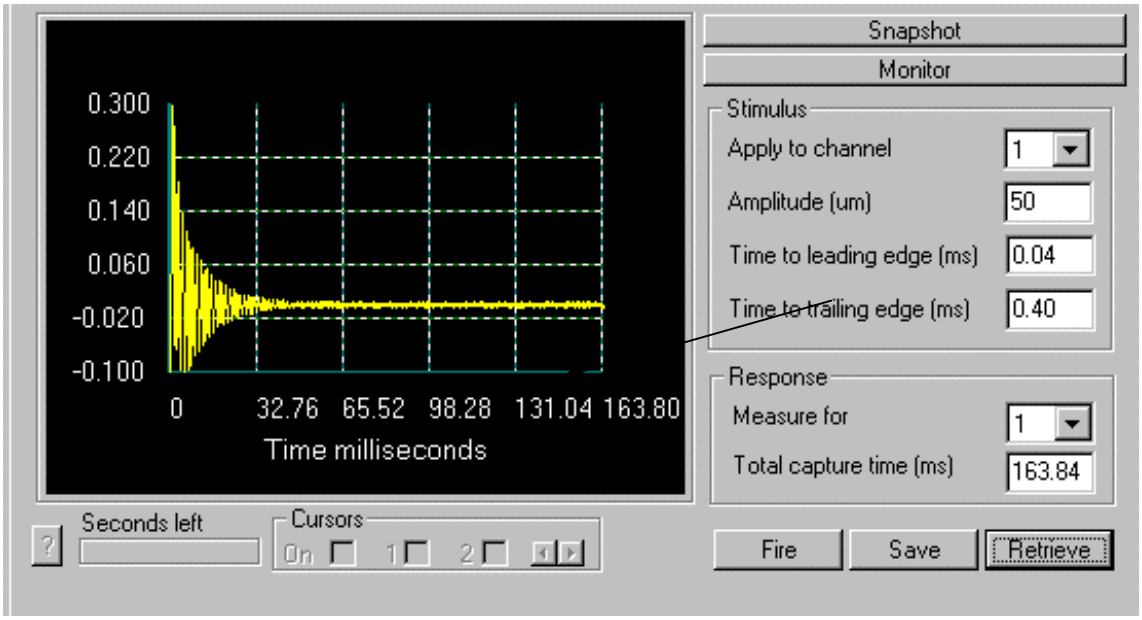
Figure 20.2: Example of Impulse Parameters



This may look a random number, but you will see where this comes from later

And monitor and save the stage response:

Figure 20.3: Snapshot Window after



At Queensgate, we use Microsoft® Excel software, so we find it best to save this data in the .txt format for import purposes. Once in Microsoft® Excel, one has a column of snapshot output (in nanometres). It is now possible to add a frequency base to this data. In our snapshot mode, 4096 data points are taken at 40 microsecond intervals. The spreadsheet should look something like:

Figure 20.4: Excel Spreadsheet showing Data after importing it

Microsoft Excel - Fourier transform sheet (PMR)

File Edit View Insert Format Tools Data Window Help

Arial 12

D11

	A	B	C	D
1	Fourier Analysis Spreadsheet For NPS3XXX Digital Controller Systems			
2	Written Philip Rhead			
3	Version 1V0			
4	7/30/98			
5	Point Number	Frequency Hz	Snapshot Output nm	
6				
7				
8				
9				
10				
11	1	6.103515625	-0.0087337	
12	2	12.20703125	-0.0013436	
13	3	18.31054688	-0.0033591	
14	4	24.4140625	-0.0087337	
15	5	30.51757813	0.0033591	
16	6	36.62109375	0.0033591	
17	7	42.72460938	-0.0087337	
18	8	48.828125	0.00739	
19	9	54.93164063	-0.0127646	
20	10	61.03515625	0.0127644	
21	11	67.13867188	-0.0020155	
22	12	73.2421875	0.0053745	
23	13	79.34570313	-0.0020155	

Next, a Fourier Transform of the data can be made. Microsoft® Excel has this function built in.

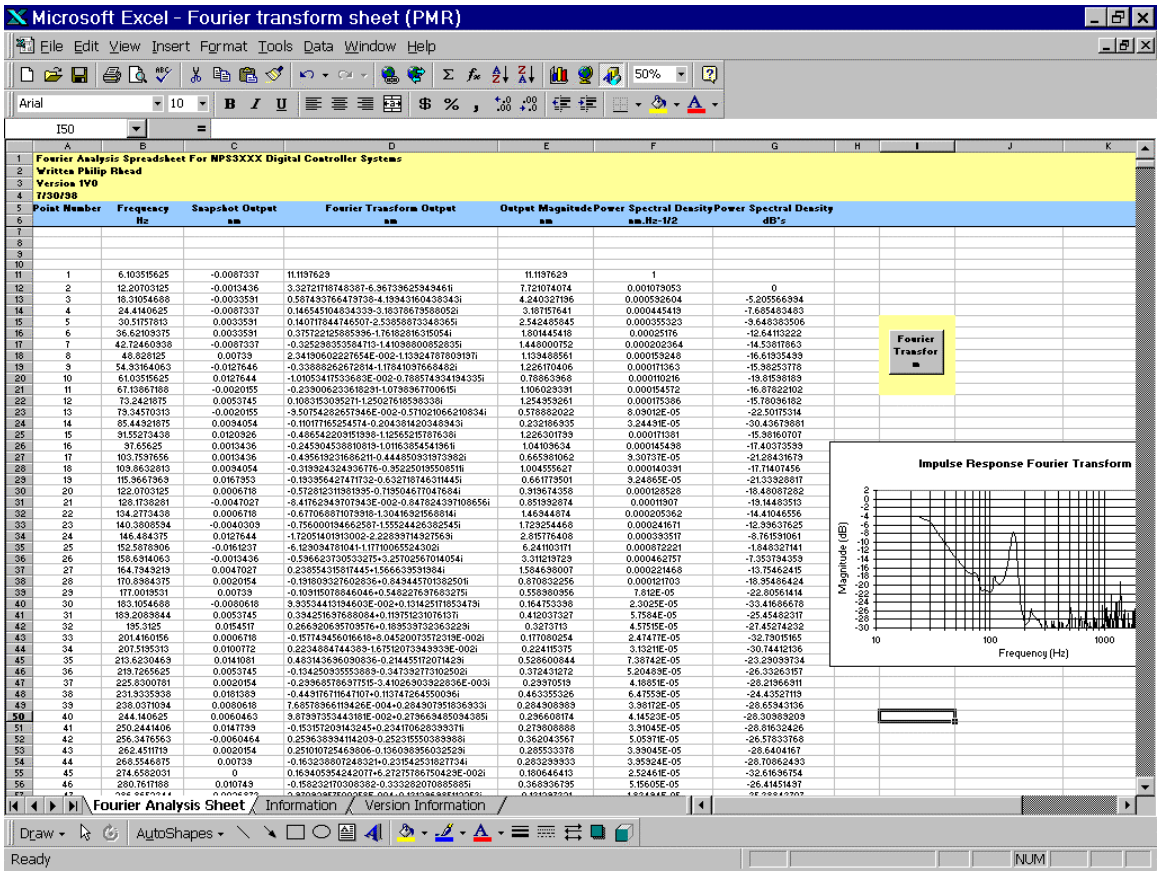
Testing Tip: The FT routine within Microsoft® Excel, requires that the number of data points to be transformed is a function of 2^n. At Queensgate, we obtain 4096 data points (2^{12}) in our snapshots. 4096 data points is equal to 163.84 microseconds (4096 x 40 microseconds).

Hence the monitor times illustrated in the earlier snapshot example is not such a random number after all.

The Fourier results comprise real and imaginary part. We are only interested in the absolute solution so we use the IMABS function to give us our frequency response data.

Finally, we convert this data as a power spectral density ($nm \bullet Hz^{-1}$) and then as dB's. The spectral density is then plotted as a function of frequency. The full spreadsheet looks like the following:

Figure 20.5: Excel Spreadsheet showing Graph of Impulse Response Fourier



From the graph we can directly measure the bandwidth (3dB down) and the resonant frequency).

20.5 Conclusion

The example illustrated above shows how the RF and 3dB bandwidth of a Queensgate system can be measured within the experimental frame.

This measurement is made very simple by use of the snapshot mode in the virtual control panel. Evaluation of the RF and BW can be made very quickly and routinely. Queensgate has made this process even simpler with Microsoft® Excel spreadsheet and macro. Please ask for a copy of this spreadsheet from Queensgate.

21 Troubleshooting

21.1 Introduction

This chapter provides troubleshooting guidelines in the event of faulty operation or failure of the NPS3110, NPS3220 and NPS3330 Digital Controller system.

Table 21.1: Troubleshooting

Symptom	Possible Cause	Action
NPS3000 Power-On LED is off	NPS3000 Mains On/Off switch is Off.	Set to On.
	Mains ac supply is off.	Switch supply On
	Mains ac connector cable not connected	Switch Off the NPS3000 Mains On/Off switch and the Mains ac supply Reconnect cable Switch On the Mains ac supply and the NPS3000 Mains On/Off switch
	Mains ac connector cable fuse blown	Replace fuse
	Mains ac supply is out of tolerance	Rectify ac supply fault
	NPS3000 is defective	Report fault to Queensgate (see Appendix C)
NPS3000 does not respond to computer control	NPS3000 Mains On LED is Off	Switch On NPS3000
	Application software fault	Run NanoControl Panel on computer. If NPS3000 responds to NanoControl Panel commands then fault lies with user application software. Possible user software faults include wrong baud rate set up, addressing the NPS3000 as a slave, addressing an invalid channel (e.g. channel 3 on a 2-Channel Digital Controller) or sending a positioning command while a Freeze command is active.
	Interface Cable not connected at both ends	Switch Off NPS3000. Reconnect Interface Cable between the computer and NPS3000. Switch On NPS3000.
	Interface Cable defective or incorrectly wired	Switch Off NPS3000. Check Interface Cable wiring (see Sections 8 to 12). Replace/repair cable and reconnect between the computer and NPS3000. Switch On NPS3000.
	Computer RS232C Port incorrectly set up	Reconfigure the computer RS232C Port to provide: No parity 8 data bits 1 start bit 1 stop bit
	Incorrect baud rate	When NPS3000 is powered on it defaults to factory default/power-up setting of 9600 baud. The baud rate can be set by software to be 1200, 2400, 4800, 9600, 19200 or 38400.
	NPS3000 is defective	Report fault to Queensgate (see Appendix C)

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Symptom	Possible Cause	Action
NPS3000 Ready LED is Off for most of the time	The NanoMechanism is not settling to the levels specified by the Ready Limit before receipt of the next position command	If you are commanding the NanoMechanism with a continually changing input (e.g. a ramp) there will be a time lag between the commanded and the measured position. This can be reduced by: Changing the READY Limit (see Section 14) Increasing the Closed Loop bandwidth to reduce settling times (see Section 5 and 14**Error! Reference source not found.**) Reduce the command rate change to allow more time for the NanoMechanism to settle
NanoMechanism Oscillates	NanoMechanism mounting and load	The stability of a NanoMechanism is affected by its mounting method and the load placed on it, particularly if the load is heavy (>100g) or has its own resonant structure. See Section 5 for further details or consult Queensgate (see Section 23).
	Closed loop parameters are unstable	Adjust PID parameters, see Sections 3,4 and 5
System Too Noisy	NanoMechanism mounting and/or Environment	The installation and environment of the NanoMechanism is vital in achieving the required performance levels. Vibration, acoustic noise, drafts and ambient temperature changes can all increase noise levels. See Section 5 for further details or consult Queensgate (see Section 23).
	Closed loop parameters are not optimised for low noise	Adjust PID parameters, see Sections 3,4 and 5
NanoControl Panel does not run on computer	Computer software/hardware incompatibility	Check that computer hardware/ software is compatible: 486 PC (minimum) Windows 95 or NT
	NanoControl Panel software corrupted	Re-install NanoControl Panel software from floppy.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

22 Maintenance and Configuration

22.1 Introduction

This Section provides user maintenance and configuration information. The Controller contains no user serviceable parts.

22.2 Routine Maintenance

22.2.1Tasks

Routine maintenance of the NPS3110, NPS3220 and NPS3330 Digital Controllers and associated NanoMechanisms consists of:

- Periodic visual inspection of the system for cable and connector security/integrity, mechanical damage and foreign body/liquid contamination.
- Periodic surface cleaning of the units. Use a lint-free cloth moistened, if necessary, with Iso-Propanol Alcohol (IPA).

DO NOT USE ACETONE, OTHER STRONG SOLVENTS OR ABRASIVE CLEANERS

22.2.2 Periodicity

Queensgate does not specify maintenance intervals. Maintenance periodicity should be set in accordance with local experience and practice.

22.3 Configuration

22.3.1 Controllers

The Controller may be factory pre-configured to:

- Your requirements,

or

- Factory default configuration.

The factory configuration includes 3 sets of PID parameters for each NanoMechanism channel. You can save up to 5 further sets of dynamic PID parameters for each channel using the NanoControl Panel (Section 4.7). These dynamic set-ups are saved in EEPROM and are preserved at power down. details optimisation of the PID parameters.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

22.3.2 NanoMechanisms

Each NanoMechanism contains a factory pre-configured EEPROM containing optimised parameters and calibration information. The parameters in the NanoMechanism EEPROM cannot be re-configured by the user. Contact Queensgate (Section 23) if you need to modify these parameters.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

23 Queensgate After-Sales Office

COUNTRY	**NAME & ADDRESS**	**Phone No.**	**Fax No.**
UK and Rest of the World	Sifam Instruments Ltd Woodland Road Torquay Devon TQ2 7AY	+ 44 1344 350 000	+ 44 1344 350 035

One e-mail address: support@nanopositioning.com

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Appendix A: Specifications

Parameter	Symbol	Value			Units	Notes
		Minimum	Typical	Maximum		
Size			288 x 307 x 70		mm	
Line Voltage		90	-	260	V	
Line Frequency		45	-	65	Hz	
Stage to controller cable length		-	2	5	m	Note 1.
Control interface			RS232			Note 2.
Command format			Single precision floating point (7 digits)			Note 3.
Measurement format			Single precision floating point (7 digits)			Note 3.
Scale factor	a_{x1}, b_{x1}		1		µm	Note 4.
NanoSensor bandwidth				5	KHz	Note 5.
Intrinsic sensor noise	$k_{xm\cdot ndens}$		100×10^{-9}		$Hz^{-1/2}$	Note 6.
HV amplifier output swing		-20	-	120	V	
HV amplifier bandwidth				10	KHz	Note 7.
Intrinsic HV noise (rms)			0.3		mV	
HV amplifier current limit				50	mA	Note 8.
Warm up time			25		Minutes	
Warm up drift			50		nm	
Thermal Drift - LD Option			200×10^{-6} 70×10^{-6}		K^{-1}	Note 9.
Operating temperature		0	-	40	ºC	Note 10.
Survival temperature		-20	-	70	ºC	Note 10.
Number of independent closed loop channels		1	3	6		Note 11.

1 2 m maximum for low noise option (-LN).

2 Included as Standard. Other interfaces are available.

3 The digital resolution is better than the system noise in most cases. Integer format is also available. The format is user controllable, factory default is IEEE standard format.

4 Factory default, User settable.

5 The ‘Read Position’ command returns position information in a 5kHz bandwidth. The ‘Snapshot’ mode returns position information in a 12.5kHz bandwidth.

6 This is the noise for the low noise option (-LN). The standard noise is three times higher. Multiply this number by the square of the sensor gap and the square root of the bandwidth, then divide by the sensor gap for 10pF capacitance to arrive at the rms noise in meters; that is, noise = $k_{xm\cdot ndens} \times G^2 \sqrt{d_{10pF}}$ where G is the capacitor gap and d_{10pF} is the capacitor gap for a 10pF capacitance.

7 The bandwidth for 4µF load. The bandwidth at higher loads is 6.1kHz (6µF load), 3.7kHz (10µF load). Typical stage capacitance is 4µF.

8 100mA current limit is available with less than one minute short circuit protection. Contact Queensgate for details.

9 For example a positioner with a sensor gap of 100µm with the –LD option has a thermal drift of $7nm \cdot K^{-1}$. Note that this is the contribution from the controller only; it does not include thermal expansion of the NanoMechanism.

10 Non-condensing.

11 To obtain six channels, two controllers are linked together. Channels can be open or closed loop (user settable).

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Appendix B: Glossary

Abbe error	A measurement error produced by rotational error ($\delta\gamma$, $\delta\theta$,$\delta\phi$) when the point of interest is not on the measurement axis
BS	British Standards
BSI	British Standards Institute
COM	PC Serial Communications Port
CR-LF	Carriage Return - Line Feed
DLL	Dynamic Link Library. A library of software functions available from all Windows compatible software.
DPT	Digital Piezo Translator
DSP	Digital Signal Processor. A microprocessor IC optimised for signal processing applications
DSP Port	A high speed serial interface built into the DSP
DSR	Data Set Ready (RS232C handshake line)
DTE	Data Terminal Equipment
DTR	Data Terminal Ready (RS232C handshake line)
EEPROM	Electronically Erasable/Programmable Read Only Memory. Non-volatile memory (i.e. it retains the data during power off) which can be re-programmed electronically.
EMC	Electromagnetic Compatibility
ESD	Electrostatic sensitive device
IC	Integrated Circuit
IEC	International Electrotechnical Commission
IPA	Iso-Propanol Alcohol
LabVIEW®	A Microsoft application for simple control of electronic laboratory equipment
Micrometer (µm)	$1x10^{-6}m$
NanoControl Panel	A Queensgate computer program giving access to the functions and controls of the NPS3000 series Digital Controller through a high level display.
NanoMechanism	A mechanism providing motion control with sub-nanometre resolution

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Nanometre (nm)	$1x10^{-9}$m
NanoPositioning	The Queensgate® technology of moving and measuring with sub-nanometre precision
NanoSensor®	Capacitance displacement sensor capable of sub-nanometre resolution with linearity error below 0.2%
NPS	NanoPositioning System
NPS-ANA-A	NPS3000-series Digital Controller analogue interface
NPS-PAR-A or –B	NPS3000-series Digital Controller high speed parallel interface
NTE	Not To Exceed
Null Modem cable	A type of RS232C interface cable used to connect the computer to the NPS3000-series Digital Controller RS232C port
OEM	Original Equipment Manufacturer
Open Loop / Closed Loop operation	The two different modes of operation of the NPS3000-series Digital Controller: Closed-Loop A capacitance displacement sensor measures the stage position and corrections are automatically applied to the command input to compensate for any non-linearity and hysteresis associated with the stage's piezo translator. Open-Loop The command input is connected directly to the piezo drive amplifier. The motion is non-linear and hysteretic. The capacitance displacement sensor, if used, simply monitors the stage position.
PC	IBM Compatible Personal Computer
PID	Proportional, Integral and Differential
$ppmK^{-1}$	Parts per million per degree Kelvin
rms	root mean square
Snapshot mode	An NPS3000 series Digital Controller facility which enables the capture of a time period of position data. This data can be displayed graphically on the NanoControl Panel or downloaded to determine the transient response for Fourier analysis or for measuring noise levels.
Yaw compensation	A facility to reduce the Abbe error when working away from the axis of motion.

Artisan Scientific - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisan-scientific.com

Appendix C: Customer Return Report, QCD6115f

QUEENSGATE LTD			CUSTOMER RETURN REPORT
RA No. QI Sales Order Number: SYSTEM(s) SERIAL No(s).		NAME / ADDRESS (CUSTOMER / AGENT): FAX: PHONE:	
REASON FOR RETURN / NATURE OF COMPLAINT:			
RETURN ☐ **SOLVED BY ADVICE** ☐ **FIXED IN FIELD** ☐			
SUGGESTED ACTION:			
❑ WARRANTY ❑ NON-WARRANTY	NTE:	DATE:	COMPILED BY:
RETURNED UNITS: (**PLEASE INCLUDE SERIAL No's, DETAILS OF RETURNED ITEMS AND CONDITION OF UNITS AND PACKAGING ALSO THE QI MANUFACTURING PART NUMBERS**			
UNITS CHECKED IN BY:		DATE:	

SUMMARY OF FAULT FOUND & ACTIONS TAKEN	PARTS REPLACED & PART No.	NAME IN CAPITALS	TOTAL HOURS	COST

INSPECTED / TESTED	CORRECTION COMPLETE (REA / PM ONLY)	REPAIR REPORT ISSUED (ATTACH COPY) (REA / PM ONLY)	DESPATCHED
SIGN DATE	SIGN DATE	SIGN DATE	INITIALS DATE

CHARGEABLE AMOUNT: SIGNED: DATE:

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