ARTISAN®
TECHNOLOGY GROUP

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

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

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

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

Custom engineering
so your equipment works exactly as you specify.

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

Expert team | Trust guarantee | 100% satisfaction

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

Find the *Honeywell 627-1002R* at our website: *Click HERE*

Honeywell

620 Logic Controller

627 MiniCOP and BASIC09 User Manual

627-8991
12/95

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

Copyright, Trademarks, and Notices

Printed in U.S.A. — © Copyright 1988, 1995 by Honeywell Inc.

While this information is presented in good faith and believed to be accurate, Honeywell disclaims the implied warranties of merchantability and fitness for a particular purpose and makes no express warranties except as may be stated in its written agreement with and for its customer.

In no event is Honeywell liable to anyone for any indirect, special, or consequential damages. The information and specifications in this document are subject to change without notice.

PRODUCT TRADEMARKS

620 LCS® and Honeywell 620 Logic Controller System® are registered trademarks of Honeywell Inc.

BASIC09® is a registered trademark of Microware, Inc.

Belden® is a registered trademark of Cooper Industries.

Decwriter® is a registered trademark of Digital Electronics Corp.

IBM AT® is a registered trademark of IBM Corp.

MS-DOS® is a registered trademark of Microsoft Corp.

OS-9® is a registered trademark of Motorola Corp.

UNIX® is a registered trademark of Bell Laboratories.

Votrax® is a registered trademark of Votrax, Inc.

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

Honeywell

620 Logic Controller

627 MiniCOP and BASIC09 User Manual

627-8991
Rev. F
12/95

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

About This Publication

This publication provides information on the Honeywell 627 MiniCOP modules (Models 627-1002RC and 627-1003RC), the OS-9 Operating System, and the BASIC09 programming language. The following sections are presented:

Section 1. MiniCOP Hardware

Section 2. Terminal Emulator

Section 3. OS-9 Operating System

Section 4. MiniCOP Interfacing

Section 5. BASIC09 Programming Language

Section 6. Appendices/Glossary

Symbol Definitions



This CAUTION symbol on the equipment refers the user to the Product Manual for additional information. This symbol appears next to required information in the manual.



WARNING, risk of electric shock. This symbol warns the user of a potential shock hazard where HAZARDOUS LIVE voltages greater than 30 Vrms, 42.4 Vpeak, or 60 Vdc may be accessible.



Protective earth terminal. Provided for connection of the protective earth (green or green/yellow) supply system conductor.



Functional earth terminal. Used for non-safety purposes such as noise immunity improvement.

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

Table of Contents

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

Table of Contents

PAGE

SECTION 1. MiniCOP HARDWARE 1

INTRODUCTION 3
MiniCOP OVERVIEW 3
GENERAL FEATURES 4
FUNDAMENTAL CONFIGURATIONS 5
CE Conformity (Europe) 5
Block Diagram of 620 Logic Controller System (LCS) 6
620 Platform Controllers Interfaced to TDC 3000 7
Series 9000 Configurations on the 620 Platform 8
MiniCOP in Processor Rack of a 620-36 LCS 9
MiniCOP in Serial I/O Rack of a Logic Manager 10
MiniCOP/Allen-Bradley PLCs/Honeywell 620-35 LCS/TDC 3000 11
MiniCOP used as "Stand-Alone" Device 12
627 MINICOP COMPONENTS, OPTIONS, & RELATED PRODUCTS 13

HARDWARE CHARACTERISTICS 14
627-1002 AND 627-1003 MiniCOP SPECIFICATIONS 14
MAJOR COMPONENTS 15
Module Frontplates 15
Circuit Boards 16
Battery 18
Toggle Switch 18
LED Indicators 18
25-PIN ASYN Ports 19
Modular Keyboard Port 19
Video Ports 20

CONFIGURATION/INSTALLATION 21
SHIPPING CONFIGURATION 21
DIP SWITCH SELECTABLE FEATURES 21
CPU Board 21
Port/Memory Board 23
Video Board 25
JUMPER-SELECTABLE FEATURES 26
ASYN 25-PIN CONNECTOR CABLE 26
VIDEO PORT CABLE 26
GENERAL INSTALLATION NOTES 27
Receipt Inspection 27
Preinstallation Considerations 27
Installation Checklist 27
Precautionary Steps for Safe and Proper Installation 28
General Installation Procedures 31
MiniCOP INSTALLATION 32
Preinstallation Steps 32
MiniCOP Installation Procedures 33
MiniCOP ADDRESS ASSIGNMENTS 34
RACK TYPE CONSIDERATIONS 39
Standard Racks 40
Augmented Racks 42
Special Considerations 44
CABLING CONSIDERATIONS 47
CE Conformity Special Conditions (Europe) 47
To External Control/Sensing Devices 47
To the Terminal Emulator 52

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

Table of Contents *(Continued)*

PAGE

VIDEO FUNCTIONS 53
OVERVIEW 53
VIDEO SIGNAL ADJUSTMENT 54
Color Monitor with Separate SYNC Control 54
Color Monitor with Combined SYNC Control 54
Monochrome Monitor 54
VIDEO COMMAND CODES 55
Syntax 56
Video Color Register 57
Video Attribute Register 57
BASIC09 Example 58

SECTION 2. TERMINAL EMULATOR 59

INTRODUCTION 61
TERMINAL EMULATOR OVERVIEW 61
Shipping Contents 63
Minimum System Requirements 63

INSTALLATION/START-UP 64
CABLING TO THE MiniCOP 64
One PC Serial Port Connected to MiniCOP 65
Two PC Serial Ports Connected to MiniCOP 66
PC Serial Ports and Loader/Terminal Port Connected to MiniCOP 67
TERMINAL EMULATOR START-UP 68
From Floppy Disk 68
From Hard Disk 68
From the 623-6100 or 623-6150 Loader/Terminal 68

CONFIGURATION 69
TERMINAL EMULATOR MAIN MENU 69
Configuration Selections (F1 and F2) 71
File Transfer, Verification, and Display Selections (F3, F4, F5 and F7) 74
Special Function Selections (F8, F9, and F10) 79
Help Screens (ALT-F10) 80

SECTION 3. OS-9 OPERATING SYSTEM 83

OS-9 OPERATIONAL OVERVIEW 85
OS-9 OPERATING SYSTEM 85
MiniCOP MEMORY STRUCTURE 86
Main Memory 87
COPRAM Memory 89
Personal Computer Memory to MiniCOP Memory Analogy 92
Main Memory Management Functions 93
OS-9 STRUCTURE 94
Layer 1 95
Layer 2 95
Layer 3 95
Layer 4 95
Layer 5 96
MEMORY MODULE CONCEPT 97
Memory Module Structure 98
Memory Module Descriptions 99
Device Descriptors 101

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

Table of Contents *(Continued)*

PAGE

OS-9 OPERATIONAL OVERVIEW *(Continued)*

SYSTEM PROGRAM MODULES 102
SHELL 102
BASIC09 102
RUNB 102
REAL-TIME CLOCK OPERATION 102
INTERRUPT PROCESSING 103
MULTITASKING AND OS-9 104
Processor Allocation and Time Slicing 104
Process States 105
CREATION OF A NEW PROCESS 106
Primary Module 106
Parameters 106
Inherited Properties 107
Assigned Properties 107
LOADING PROGRAM MODULES INTO MEMORY 108

OS-9 PROGRAMMING BACKGROUND 109
THE SHELL 109
Operational Overview 109
Command Line Parameters 113
Command Line Processing 114
Built-In Shell Commands and Options 115
Command Grouping 116
OS-9 INPUT/OUTPUT OPERATIONS 117
Unified Input/Output System 117
I/O Device Names 118
Receiving Input from an I/O Device 118
Sending Output to an I/O Device 119
EXECUTION MODIFIERS 120
Alternate Memory-Size Modifier (#) 120
I/O Redirection Modifiers (<, >, >>) 121
Sequential Execution Modifier (;) 123
Multitasking Modifier (&) 124
WORKING IN COPRAM 125
COPRAM Directory Structure 125
Additional Information About COPRAM Directories 127
Data and Execution Directories 128
Creating Directories 130
Deleting Directories 133
Using and Changing Directories 135
Anonymous Directory Names 138
WORKING WITH OS-9 FILES 139
File Usage in OS-9 139
Naming OS-9 Files 140
Physical File Organization 141
File Reading and Writing 141
Executable Program Module Files 141
Random-Access Data Files 142
Text Files 142
Shell Procedure Files 143
File Security System 145
Examining and Changing File Attributes 146
Loading, Linking, Saving, and Unlinking Modules 148
Memory Fragmentation 153

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

Table of Contents *(Continued)*

PAGE

OS-9 PROGRAMMING BACKGROUND *(Continued)*

Unprotected I/O Operations 156

OS-9 SYSTEM COMMANDS LISTING 157
SYNTAX 157
COMMANDS 157
ATTR 158
BINEX 160
EXBIN 160
BUILD 162
CHD 163
CHX 163
CLEAR 164
COPY 165
DATE 166
DEL 167
DELDIR 168
DIR 169
DISPLAY 170
DSKBAK 171
DUMP 172
EX 173
EXBIN 174
FREE 175
IDENT 176
INKEY 177
KILL 178
LINK 179
UNLINK 179
LIST 180
LOAD 181
MAKDESC 182
MAKDIR 185
MDIR 186
MERGE 188
MFREE 189
PD 190
PROCS 191
RENAME 192
RESET 193
SAVE 195
SETIME 197
SETPORT 198
SETPR 199
SHELL 200
SLEEP 201
TLOAD 202
TMODE 203
UNLINK 205
VERIFY 206
XMODE 207

MiniCOP/OS-9 TUTORIAL 209
INTRODUCTION 209

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

Table of Contents *(Continued)*

PAGE

MiniCOP/OS-9 TUTORIAL *(Continued)*

TUTORIAL REQUIREMENTS 209
Software 209
Hardware 209
Documentation 210
MiniCOP SET-UP TUTORIAL 211
Hardware Configuration/Initialization 211
MiniCOP Initialization 216
OS-9 FUNDAMENTAL OPERATIONS TUTORIAL 220
General System Operations 220
Main Memory 221
COPRAM Memory 222
Loading, Linking, and Saving Program Modules 230
CONCLUSION 236

SECTION 4. MiniCOP INTERFACING 237

MiniCOP INTERFACING 239
INTERFACING OVERVIEW 239
MiniCOP TO HOST 620 PLATFORM CONTROLLER 240
MiniCOP Data Table 240
MiniCOP Access to the Data Table (IPC621 DRIVER UTILITY) 244
IPC621 244
Special Condition Flag 246
DATACONTROL 248
HOST 620 PLATFORM CONTROLLER to MiniCOP 249
Data Table Pointer 249
Control Bits 251
LADDER LOGIC COMMUNICATION FUNCTIONS 254
Parallel Programming Block Descriptions 256
Serial Programming Block Descriptions 269
Parallel Programming Block Time Delays 286
MiniCOP/PERSONALITY PROCEDURES TO PLCs 289
MiniCOP to 620 Platform Controller Personality Procedure 289
MiniCOP to Allen-Bradley PLCs Personality Procedure 296
MiniCOP to MODBUS Personality Procedure 302
Updating 627 COP BASIC09 Application Programs 306
MINICOP TO OTHER TARGET DEVICES 306

SECTION 5. BASIC09 PROGRAMMING LANGUAGE 307

BASIC09 OPERATIONAL OVERVIEW 309
INTRODUCTION 309
Programming Features 310
GENERAL PRACTICE 311
MODES OF OPERATION 311
BASIC09 Mode 313
Edit Mode 314
Execution Mode 315
Debug Mode 317
BASIC09 WORKSPACE 318
CHANGING THE WORKSPACE SIZE 320
Method 1 320
Method 2 320

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

Table of Contents *(Continued)*

PAGE

BASIC09 OPERATIONAL OVERVIEW *(Continued)*

NESTED WORKSPACES 321
Interleaved Workspaces 321
Parallel Workspaces 323
MODULAR PROGRAMMING 325

BASIC09 PROGRAMMING BACKGROUND 327
PROCEDURE NAMES 327
LINE NUMBERS 327
COMMANDS AND STATEMENTS 328
Reserved Words 329
Syntax Notation 329
DATA REPRESENTATION, STORAGE, AND STRUCTURE 330
CONSTANTS 330
Numeric Constants 330
String Constants 331
Boolean Constants 331
VARIABLES 332
Declaring Variables 333
Variable Storage 334
PARAMETER PASSING 335
Passing Parameters by Reference 337
Passing Parameters by Value 337
DATA TYPES 338
Real 339
Byte 340
Integer 341
String 342
Boolean 344
Complex 345
Automatic Type Conversions 346
ARRAYS 347
Vector Arrays 347
Matrix Arrays 347
Indexing Arrays 348
EXPRESSIONS 350
OPERATORS 350
Fundamental Operators 350
Arithmetic Operators 351
Logical Operators 352
Relational 353
Operator Precedence 354
Special Function Operators 355
CONCATENATION 357
INPUT/OUTPUT OPERATIONS 358
I/O Paths 358
Sequential File Operations 359
Random Access File Operations 360
WORKING IN THE BASIC09 MODE 361
Creating New BASIC09 Procedures 362

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

Table of Contents *(Continued)*

PAGE

BASIC09 PROGRAMMING BACKGROUND *(Continued)*

WORKING IN THE EDIT MODE 363
Editing Existing BASIC09 Procedures 363
Moving the Edit Pointer 364
Listing BASIC09 Procedures 365
Line Editing Commands 366
String Editing Commands 367
WORKING IN THE EXECUTION MODE 369
Running BASIC09 Procedures 369
WORKING IN THE DEBUG MODE 371
MANIPULATING BASIC09 PROCEDURES 373
Saving Source Code Procedures to COPRAM 373
Saving Source Code Procedures to the Terminal Emulator 374
Packing Source Code Procedures to COPRAM 375
Packing Source Code Procedures to the Terminal Emulator 376
Loading Source Code Procedures from COPRAM 377
Loading Source Code Procedures from the Terminal Emulator 378
Listing Source Code Procedures to a Printer via the Terminal Emulator 379
PROGRAMMING TIPS 380
Tips to Reduce Memory Usage 380
Tips to Increase Procedure Execution 381

BASIC09 COMMANDS AND STATEMENTS LISTING 383
CLASSIFICATIONS 384
BASIC09 Mode Commands 384
Debug Mode Commands 384
Control Statements 384
I/O Statements 385
Internal Data Statements 385
Formatted Output Statement 385
SYNTAX 386
COMMANDS 386
$ 387
BASE 0 388
BASE 1 388
BREAK 389
BYE 390
CHAIN 391
CHD 392
CHX 393
CLOSE 394
CONT 395
CREATE 396
DATA 397
DEG/RAD 398
DELETE 399
DIM 400
DIR 402
DO 403
E/EDIT 404
END 405
ENDEXIT 406
ENDIF 406
ENDLOOP 406
ERROR 407

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

Table of Contents *(Continued)*

PAGE

BASIC09 COMMANDS AND STATEMENTS LISTING *(Continued)*

EXITIF 408
FOR/NEXT 409
GET/PUT 410
GOTO 412
GOSUB/RETURN 413
IF (VERSION 1) 414
IF (VERSION 2) 415
INPUT 416
KILL 417
LET 419
LIST 420
LOAD 421
LOOP/ENDLOOP/EXITIF/ENDEXIT 422
MEM 424
NEXT 425
ON ERROR GOTO 426
ON GOSUB 427
ON GOTO 427
OPEN 428
PACK 429
PARAM 430
PAUSE 431
POKE 432
PRINT 433
PRINT TAB 434
PRINT USING 435
PUT 444
Q 445
RAD 446
READ (External File Operation) 447
READ (Internal Constant Operation) 448
REM 449
RENAME 450
REPEAT/UNTIL 451
RESTORE 452
RETURN 453
RUN 454
SAVE 456
SEEK 457
SHELL 459
STATE 460
STEP 461
STOP 462
THEN 463
TRON/TROFF 464
TYPE 465
UNTIL 467
WHILE/DO 468
WRITE 469

BASIC09 TUTORIAL 471
INTRODUCTION 471

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

Table of Contents *(Continued)*

PAGE

BASIC09 TUTORIAL *(Continued)*

TUTORIAL REQUIREMENTS 471
Software 471
Hardware 471
Documentation 472
BASIC09 TUTORIAL PRELIMINARIES 472
BASIC09 FUNDAMENTAL OPERATIONS TUTORIAL 473
Initializing BASIC09 473
Creating, Editing, and Running BASIC09 Procedures 475
Modular Programming and Parameter Passing 482
Manipulating BASIC09 Procedures 483
CONCLUSION 484

SECTION 6. APPENDICES/GLOSSARY 485

APPENDIX A — MODULE MODEL NUMBER "R" SUFFIX 487
OPERATIONAL OVERVIEW 487

APPENDIX B — CABLE CONSTRUCTION AND CONFIGURATION 489
ASYNC PORT PIN ASSIGNMENTS 490
SERIAL COMMUNICATIONS CABLES 491
20mA Communications Cable 491
RS232 Half-Duplex Communications Cable 492
RS232 Full-Duplex Communications Cable 493
RS422/485 Half-Duplex Point-to-Point Communications Cable 494
RS422/485 Full-Duplex Point-to-Point Communications Cable 495
RS422/485 Half-Duplex Multidrop Communications Cable 496
RS422/485 Full-Duplex Multidrop Communications Cable 497
VIDEO CABLES 498
627-1003 MiniCOP Video Port Cable Construction 498
MULTIDROP COMMUNICATION NETWORK CONFIGURATION 499
MiniCOP Within the Network 499
MiniCOP at the End of the Network 500
BIASING/TERMINATING RESISTORS 501

APPENDIX C — AUGMENTED RACK RESISTOR PACK UPDATE 503
DETERMINE WHICH RACKS MUST BE UPDATED 504
UPDATING AUGMENTED RACKS 506
TEST THE UPDATED AUGMENTED RACK 507
TROUBLESHOOTING 507

APPENDIX D — ASCII CODE CHARACTER SET 509

APPENDIX E — RUNB BACKGROUND 511
INTRODUCTION 511
OPERATIONAL CHARACTERISTICS 511

APPENDIX F — MINICOP ERROR MESSAGES 513
BASIC09 ERROR CODES 513
OS-9 ERROR CODES 515

APPENDIX G — APPLICATION NOTES 517

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

Table of Contents *(Continued)*

PAGE

APPENDIX H — CONVERSION TABLES 521
SELECTED BINARY, DECIMAL, AND HEXADECIMAL CONVERSIONS 521
MATHEMATIC CONVERSIONS 522

APPENDIX I — BASIC09 AND PERSONALITY PROCEDURES 523

APPENDIX J — USING CONF1003 TO LOAD THE V4 AND K4 DEVICE DESCRIPTORS 525

APPENDIX K — PRGMB FILE EXAMPLE 527

GLOSSARY 529

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

List of Figures

NO.	TITLE	PAGE
1	627-1002RC and 627-1003RC MiniCOP Modules	3
2	Block Diagram of 620 Logic Controller System (LCS)	6
3	620 LCS Platform of Controllers Interfaced to TDC 3000	7
4	Series 9000 Configurations on the 620 Platform	8
5	MiniCOP in Processor Rack of a 620-36 LCS	9
6	MiniCOP in Serial I/O Rack of Logic Manager	10
7	MiniCOP/Allen-Bradley PLCs /Honeywell 620-35 LCS/TDC 3000	11
8	MiniCOP Used as "Stand-Alone" Device	12
9	627-1002RC and 627-1003RC MiniCOP Module Frontplates	15
10	627-1002 Circuit Boards	16
11	627-1003 Circuit Boards	17
12	CPU Board DIP Switches	21
13	Port/Memory Board DIP Switch Locations	23
14	Video Board	25
15	CPU Board Jumper Location	26
16	System Grounding	28
17	Power Supply Grounding	29
18	I/O Addressing Example	35
19	MiniCOP Slot Addressing Example #1 (Top View)	37
20	MiniCOP Slot Addressing Example #2 (Top View)	38
21	621-9990C I/O Rack Backplane Layout	40
22	Standard (Nonaugmented) Rack Connections	41
23	621-9992C Augmented I/O Rack Backplane Layout	42
24	Augmented Rack/MiniCOP Connection	43
25	Backplanes on Augmented Racks	45
26	627-1003RC MiniCOP	53
27	Terminal Emulator "Bridge" Between the Personal Computer and MiniCOP	62
28	One PC Serial Port Connected to MiniCOP	65
29	Two PC Serial Ports Connected to MiniCOP	66
30	PC Serial Port and Loader/Terminal Port Connected to MiniCOP	67
31	Terminal Emulator Main Menu Display	69
32	Terminal Port Configuration Menu	71
33	Tape Port Configuration Menu	71
34	<F3> Function Window	74
35	<F4> Function Window	75
36	<F4> Warning Prompt	76
37	<F5> Function Window	77
38	Exit Program Prompt	79
39	F1 and F2 Help Screen	80
40	F3, F4, F5 Help Screen	81
41	Main Menu, F7, and F10 Help Screen	82
42	627 MiniCOP Memory Structure	86
43	Main Memory Detail	87
44	COPRAM Structure After Initialization	89
45	Example COPRAM Directory Structure	91
46	MiniCOP to Personal Computer Memory Analogy	92
47	Main Memory Management	93
48	OS-9 Organization	94
49	Module Directory Listing	98
50	Memory Module Contents	98
51	Listing a File Created Via a Redirection Modifier	122
52	COPRAM Relationship to the System Service Directory	126
53	Root Directory (/D0) Listing	128
54	CMDS Directory Listing	129
55	COPRAM Directory Structure After Initialization	130
56	Directory Structure after Creating DATADIR	131

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

List of Figures *(Continued)*

NO.	TITLE	PAGE
57	Directory Structure after Creating Source	132
58	Directory Deletion Options	133
59	Using and Changing Directories	136
60	File Attributes	146
61	Program Module Transfer	148
62	Fragmented Memory	154
63	Unfragmented Memory	155
64	File Permission Abbreviations and Definitions	158
65	Module Type Characters	187
66	SAVE Command Syntax Definition	196
67	Using One Serial COM Port (COM1 – RS-232)	213
68	Using Two Serial COM Ports (COM1 and COM2 – RS-232)	214
69	Using Loader/Terminal Interface Board (RS-422/485) as COM2	215
70	Module Directory Listing	218
71	COPRAM Structure After Initialization	218
72	Execution Directory Listing	219
73	Available Main Memory Example	221
74	Root Directory (D0) Listing	222
75	Available COPRAM Memory and Charactersistics	222
76	COPRAM Structure with "NEWDIR" Directory ADDED	223
77	COPRAM Directory Listing for /D0 Directory	224
78	COPRAM Directory Listing for Subdirectory "CMDS"	225
79	Listed Textfile	227
80	Available COPRAM Memory and Characteristics	231
81	MiniCOP Data Table	241
82	Changing a Single Bit	242
83	Changing a 16-Bit Register	243
84	Data Table Pointer Example	250
85	MiniCOP Addressing Example	251
86	MiniCOP Address to Control Bits and Programming Block Relationship	252
87	MiniCOP to 620 Platform Controller — Example #1	290
88	MiniCOP to 620 Platform Controller — Example #2	291
89	MiniCOP to 620 Platform Controller — Example #3	292
90	MiniCOP to Allen-Bradley PLC — Example #1	297
91	MiniCOP to Allen-Bradley PLC — Example #2	298
92	MiniCOP to Allen-Bradley PLC — Example #3	299
93	MiniCOP to Modbus Example	303
94	BASIC09 Mode Relationships	312
95	Main Memory Detail	319
96	Interleaved BASIC09 Workspaces	322
97	Parallel BASIC09 Workspaces	323
98	Simplified Modular Program Flowchart	326
99	Parameter Passing Example #1	335
100	Parameter Passing Example #2	336
101	Real String Contents	339
102	String Length Assignments	343
103	Array Indexing Modes	348
104	BASIC09 Procedure Listing	365
105	20mA Communication Cable	491
106	RS232 Half-Duplex Communication Cable	492
107	RS232 Full Duplex Communication Cable	493
108	RS422/485 Half-Duplex Point-to-Point Communication Cable	494
109	RS422/485 Full Duplex Point-to-Point Communication Cable	495
110	RS422/485 Half-Duplex MultiDrop Communication Cable	496
111	RS422/485 Full-Duplex MultiDrop Communication Cable	497
112	627-1003 Video Port Cable Construction	498

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

List of Figures *(Continued)*

NO.	TITLE	PAGE
113	Multidropping ASYNC Port Devices Example (MiniCOP Within Network)	499
114	Multidropping ASYNC Port Devices Example (MiniCOP at End of Network)	500
115	Augmented Rack Backplane Layouts	505
116	Resistor Pack Pin Outs	506

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

List of Tables

NO.	TITLE	PAGE
1	627-1002 and 627-1003 MiniCOP Components, Options, and Related Products	13
2	CPU Board DIP Switch Functions	22
3	Port Memory Board DIP Switch SW1 Functions	24
4	Port/Memory Board DIP Switch SW2 Functions	24
5	Video Board DIP Switch Functions	25
6	MiniCOP Rack Model Number Compatibility	39
7	Tray to Tray Spacing — Inches (mm)	49
8	Tray to Conduit Spacing — Inches (mm)	49
9	Conduit to Conduit Spacing — Inches (mm)	49
10	Raceway Shielding	51
11	Video Command Codes	55
12	Video Color Register	57
13	Video Attribute Register	57
14	Device Driver/Descriptor to MiniCOP Ports Relationship	96
15	OS-9 Command Quick Reference Guide	112
16	I/O Device Names (Descriptors)	118
17	MAKDESC Parameters	183
18	TMODE Parameters	204
19	XMODE Parameters	208
20	Control Bit Definitions	253
21	Read the Real-Time Clock Data Format	256
22	Read the Real-Time Clock Data Format	270
23	Parallel Programming Block Time Delays	287
24	MiniCOP to Host 620 Platform Controller Communication Comparison	293
25	620 LCS Personality Procedure Operations	295
26	Allen-Bradley Personality Procedure Operations	301
27	Modbus Personality Procedure Operations	305
28	BASIC09 Mode Commands Summary	313
29	Edit Mode Commands Summary	314
30	Execution Mode Commands Summary	316
31	Debug Mode Commands Summary	317
32	BASIC09 Reserved Words	329
33	Data Type Overview	338
34	Fundamental Operators	350
35	NOT Truth Table	352
36	AND Truth Table	352
37	OR Truth Table	352
38	XOR Truth Table	352
39	Operator Order of Precedence	354
40	BASIC09 Special Function Operators	356
41	Execution Speed of Integer and Real Arithmetic Functions	381
42	I/O Statements	385
43	MiniCOP ASYNC Port Pin Assignments	490
44	Biasing/Terminating Resistors	501
45	Resistor Pack Update Requirement	504
46	ASCII Code Character Set	509
47	BASIC09 Error Codes	513-514
48	OS-9 Error Codes	515-516
49	Application Note Summary	518-519
50	Number System Conversions	521
51	Mathematical Conversions	522

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

USER MANUAL CROSS-REFERENCE FOR THE 627 MiniCOP SYSTEM

MANUAL	MATERIAL COVERED
620- Logic Controller System Configuration and Ordering Guide Form 6654-PR1	System configurations (general specifications and layouts) and ordering guidelines for all 620 product lines.
620-12/1633/36 Logic Controller System User Manual Form 620-8964	Detailed hardware description; modes of operation; diagnostics; remote serial operation; ABC, RTU and PeerData Network protocol functions supported by the communications port; instruction set and opcodes.
620-11/14/1631 Logic Controller System User Manual Form 620-8976	Detailed hardware description; modes of operation; diagnostics; remote serial operation; ABC, RTU protocol functions supported by the communications port; instruction set and opcodes.
620-25 and 620-35 Logic Controller System User Manual Form 620-8984	Detailed hardware description; 620-25 and 620-35 System configurations; modes of operation; diagnostics; instruction set and opcodes.
620-10 and 620-15 Logic Controller System User Manual Form 620-8999	Detailed hardware description; 620-10 and 620-15 System configurations; modes of operation; diagnostics; instruction set and opcodes.
620-06 Logic Controller System User Manual Form 620-8975	Detailed hardware description; 620-06 System configuration; modes of operation; diagnostics; instruction set.
620 Installation and 621 I/O Communications User Manual Form 620-8996	620 System overview; system configuration for all logic controllers, as well as parallel, serial, and slave I/O; addressing; rack assembly and mounting; module settings (jumpers and DIP switches); module installation; cable and conduit routing; wiring; 621 I/O System description; reference information on superseded model numbers.
620 Installation and 621 I/O Communications for the 620-12/1633/36 Logic Controller Systems User Manual Form 620-8962	Guidelines for selecting and installing components of the 620-12/1633/36 Logic Controller Systems; includes descriptions of components, system set-up procedures, and system configuration examples.
623-60 MS-DOS Loader User Manual (Version 2.X) Form 623-8993	Complete description of the MS-DOS Loader System; configuration and installation; system start-up and menus; editing instructions; edit/display functions; documentation functions; utility functions; 620 instructions set; menu maps.

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

USER MANUAL CROSS-REFERENCE FOR THE 627 MiniCOP SYSTEM *(Continued)*

MANUAL	MATERIAL COVERED
623-60 MS-DOS Loader User Manual (Version 3.X) Form 623-8986	Complete description of the MS-DOS Loader System; configuration and installation; system start-up and menus; editing instructions; edit/display functions; documentation functions; utility functions; 620 instructions set; menu maps.
623-60 MS-DOS Loader User Manual (Version 4.X) Form 623-8983	Complete description of the MS-DOS Loader System; configuration and installation; system start-up and menus; editing instructions; edit/display functions; Function Block programming techniques, PeerData Network configuration, documentation functions; utility functions; 620 instructions set; menu maps.
623-6100/6150 Loader/Terminal User Manual Form 627-8991	Introduction to the L/T uses and literature; detailed system description (hardware and operating system); configuration and startup procedures; basic L/T operation, including operating system applications; MS-DOS operating system commands.
623-6041 Terminal Emulator User Manual Form 623-8989	Description and contents listing of the 623 Terminal Emulator; Cabling to peripheral devices; Terminal Emulator software installation, start-up, configuration and use.

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

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

Section 1.
MiniCOP Hardware

About This Section

This section presents information on the MiniCOP hardware, to include an overview of the available MiniCOP modules, fundamental configurations, module hardware characteristics, installation and configuration procedures, and video functions.

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

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

INTRODUCTION

MiniCOP OVERVIEW

The Honeywell 627 MiniCOP Modules (Models 627-1002RC and 627-1003RC — shown in Figure 1) are industrial rated microprocessor-based computing modules that provide an interface between intelligent factory floor devices and the 620 Platform of Honeywell controllers. Each module has the capability of being programmed in the BASIC09 Programming Language or optionally the OS-9 Assembly Language (Assembly Language Model No. 626-1018). The MiniCOP modules can collect, monitor, store, and display data retrieved from connected intelligent factory floor devices. They can also perform high-level data processing on the data to include such functions as trigonometry, floating point math, or practically any operation capable of being programmed in BASIC09 or the OS-9 Assembly Language.

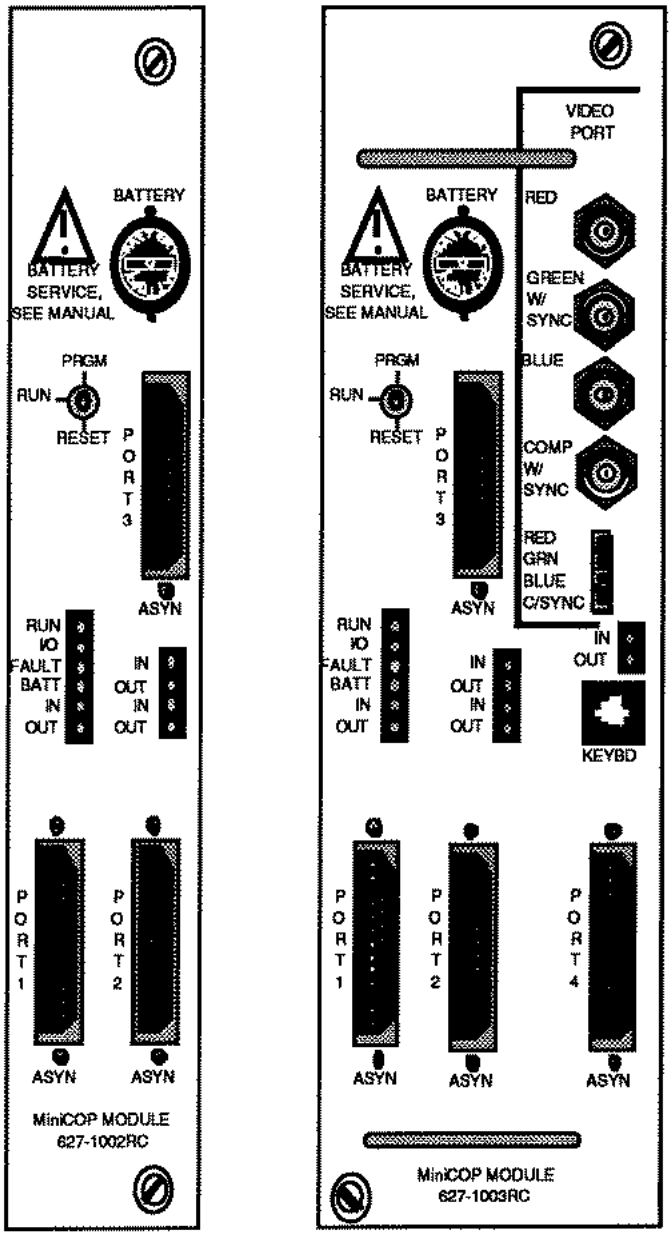


FIGURE 1 - 627-1002RC AND 627-1003RC MINICOP MODULES

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

NOTE

Model numbers 627-1002RC and 627-1003RC are direct replacements for the 627-1002 and 627-1003 MiniCOP modules. All references to model numbers 627-1002 and 627-1003 in this text also pertain to the 627-1002RC and 627-1003RC models unless otherwise noted. Refer to *Appendix A* for specific details regarding the 627-1002RC and 627-1003RC MiniCOP modules.

The 627 MiniCOP is capable of interfacing with printers, barcode readers, LED displays, CRT terminals, or any other devices that can be configured for RS232, RS422/485, or 20mA serial communications. Included in these other devices are the Honeywell 620 Platform controllers and programmable logic controllers (PLCs) from other vendors. The term "620 Platform controllers" refers to the Honeywell 620 Logic Controller System (620 LCS), the Series 9000 and 9000e Loop and Logic Controllers, and the Logic Manager (LM). Three prepackaged software drivers called Personality Procedures are also available for the MiniCOP. These drivers can be used to simplify the task of implementing a control scheme where the MiniCOP module is to interface to 620 LCSs, Allen-Bradley PLCs, or Modicon PLCs via its serial ports.

The 627-1002 and 627-1003 MiniCOPs are double-slot- and triple-slot-width modules respectively. They can be placed in the I/O slots of any I/O rack or processor rack that accepts modules that use PUSH/PULL ladder logic instructions. Both models contain 64K of main memory with 256K of RAM storage. The 627-1003 MiniCOP also provides for color or composite RGB video output.

GENERAL FEATURES

Each MiniCOP module has the following features:

- Programmable in the BASIC09 Programming Language and the optional OS-9 Assembly Language.
- Uses the Motorola 68B09E microprocessor.
- Uses the OS-9 Operating System (a UNIX-like operating system developed by Microware for Motorola).
- 16K EPROM, 48K RAM workspace, 256K CMOS RAM (solid-state RAM Disk).
- 8192 single-bit or 512 sixteen-bit word data table to facilitate interfacing to a host 620 Platform controller.
- 9-month battery backup at full memory.
- Calculation of floating point and trigonometric functions.
- Real-time clock capable of being set and read by the host 620 Platform controller's ladder logic.
- Special condition flags to alert the host 620 Platform controller of predetermined conditions requiring immediate attention.
- Asynchronous serial communications ports to interface to external devices.
- The 627-1003 MiniCOP also features monochrome or color RGB video outputs, easy access video signal level adjustments, and a modular keyboard jack.
- CE Mark (Europe)—Conformity with **73/23/EEC**, the Low Voltage Directive, and **89/336/EEC**, the EMC Directive.

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

FUNDAMENTAL CONFIGURATIONS

Since the 627 MiniCOP can be used with each system in the 620 Platform of controllers, the number of possible specific applications and configurations is almost endless. Throughout the remainder of this section, a general layout of the 620 Platform is presented along with several possible sample configurations and applications for the MiniCOP to provide a general "feel" for the module and its uses.

CE CONFORMITY (EUROPE)

Directives

This system is in conformity with the protection requirements of the following European Council Directives: **73/23/EEC**, the Low Voltage Directive, and **89/336/EEC**, the EMC Directive. Conformity of this product with any other "CE Mark" Directive(s) shall not be assumed.

Deviation from the installation conditions specified in this manual, and the special conditions for CE conformity in the Important Cable Considerations section, may invalidate this product's conformity with the Low Voltage and EMC Directives.

Product Classification

Class I equipment with protective earthing (grounding). Programmable Controllers are considered as components of a control system and may be installed as enclosed or open equipment. As such, overall system safety including installation and application is the user's responsibility. (EN 61131-2)

Overvoltage Category

Category II: The user shall insure as part of the installation, that transient overvoltages at the point of connection to the mains power supply will be controlled, and do not exceed overvoltage category II. (EN 61131-2)

Pollution Degree

Pollution Degree 2: Normally non-conductive pollution with occasional conductivity caused by condensation. (IEC 664-1)

Altitude

This equipment is suitable for operation up to 2000 m (6560 ft.).

EMC Classification

Group 1, Class A, ISM Industrial Equipment.

ATTENTION

The emission limits of EN 50081-2 are designed to provide reasonable protection against harmful interference when this equipment is operated in an industrial environment. Operation of this equipment in a residential area may cause harmful interference. This equipment generates, uses, and can radiate radio frequency energy and may cause interference to radio and television reception when the equipment is used closer than 30 meters (98 feet) to the antenna(e). In special cases, when highly susceptible apparatus is used in close proximity, the user may have to employ additional mitigating measures to further reduce the electromagnetic emissions of this equipment.

Declarations of Conformity

51197616-000 attests to this system's conformity to EN 61131-2, Programmable Controllers—Part 2: Equipment Requirements and Tests, in conformity with **73/23/EEC**, the Low Voltage Directive. 51197614-000 attests to this system's conformity with **89/336/EEC**, the EMC Directive using a Technical Construction File (TCF).

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

BLOCK DIAGRAM OF 620 LOGIC CONTROLLER SYSTEM (LCS)

Figure 2 presents a fundamental block diagram illustrating the 620 LCS. As can be seen, the MiniCOP is part of a broad logic control package.

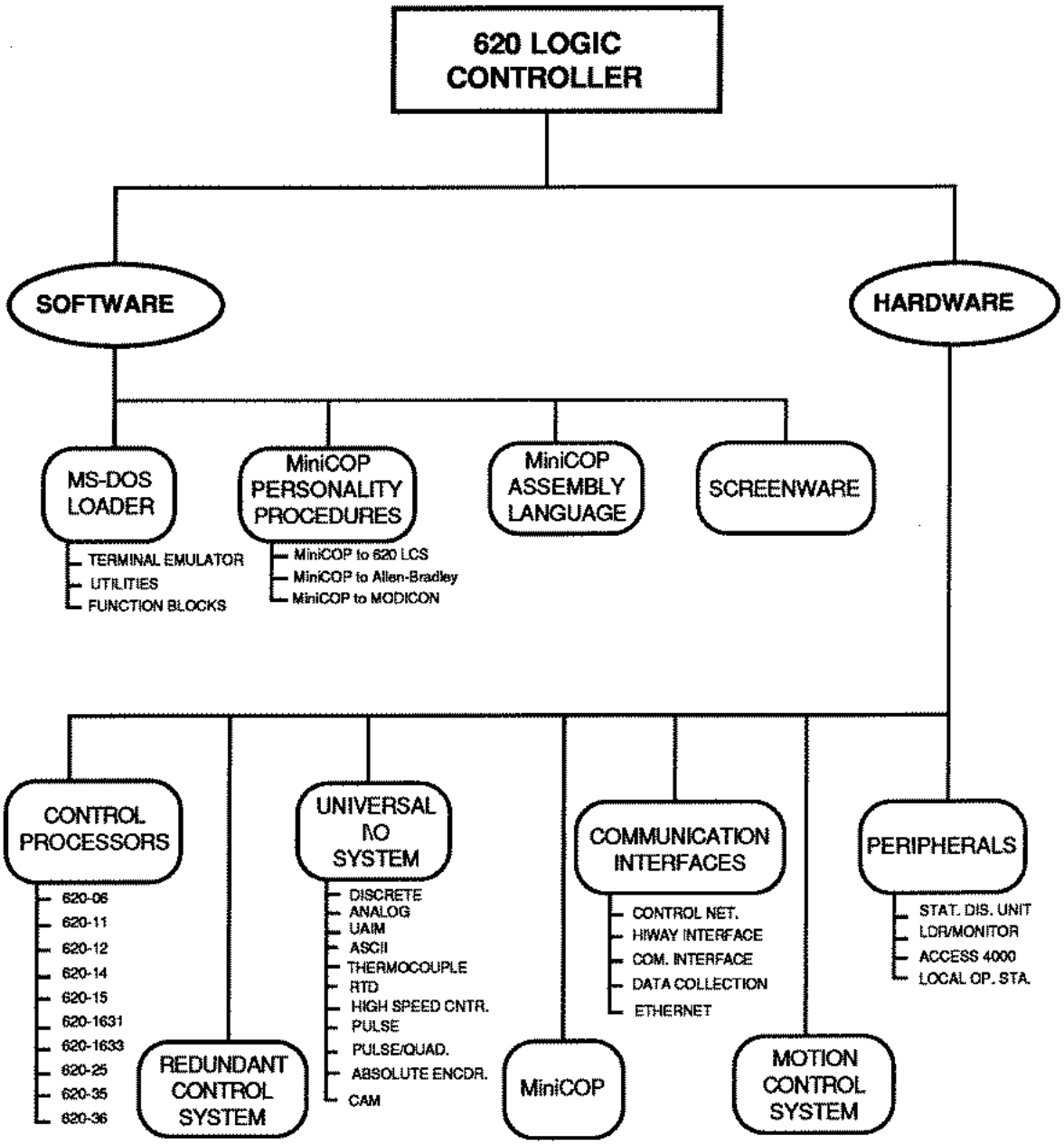
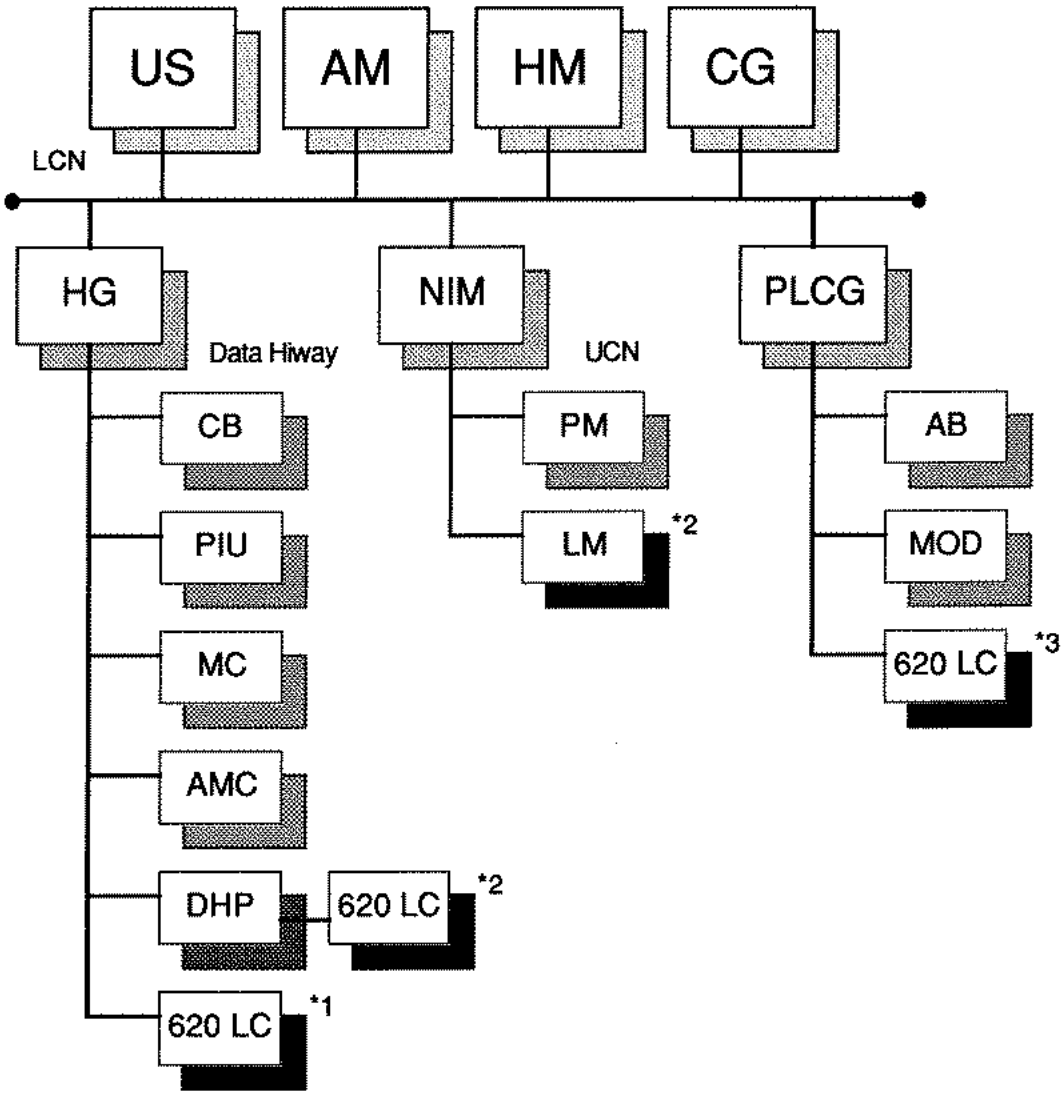


FIGURE 2 - BLOCK DIAGRAM OF 620 LOGIC CONTROLLER SYSTEM (LCS)

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

620 PLATFORM CONTROLLERS INTERFACED TO TDC 3000

Figure 3 illustrates different methods that can be used to interface the 620 Platform controllers to the TDC 3000. In all cases the MiniCOP can be implemented within the 620 Platform controller portion of the control strategy. Additionally, Figure 3 illustrates that the Logic Manager (LM) and the Series 9000 Loop and Logic Controller are 620 Platform devices that can also utilize the MiniCOP.



*1 Any 620 LC Processor connected to the Data Hiway via the Hiway Interface Module (HIM).

*2 Any 620 LC Processor connected to the Data Hiway through the Data Hiway Port (DHP), or connected to the Programmable Controller Gateway (PLCG), via the Communications Interface Module (CIM).

*3 TDC 3000 Logic Manager connected to the Universal Control Network (UCN) via the Logic Manager Module (LMM).

FIGURE 3 - 620 LCS PLATFORM OF CONTROLLERS INTERFACED TO TDC 3000

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

SERIES 9000 CONFIGURATIONS ON THE 620 PLATFORM

Figure 4 shows that each model of the Series 9000e Loop and Logic Controller is built on the third generation 620 Platform. Because of this all Series 9000e models can take full advantage of the 627 MiniCOP in the same manner as the 620 LCS, and the Logic Manager (LM).The Series 9000 Loop and Logic Controller is built on the second generation of 620 Platform.

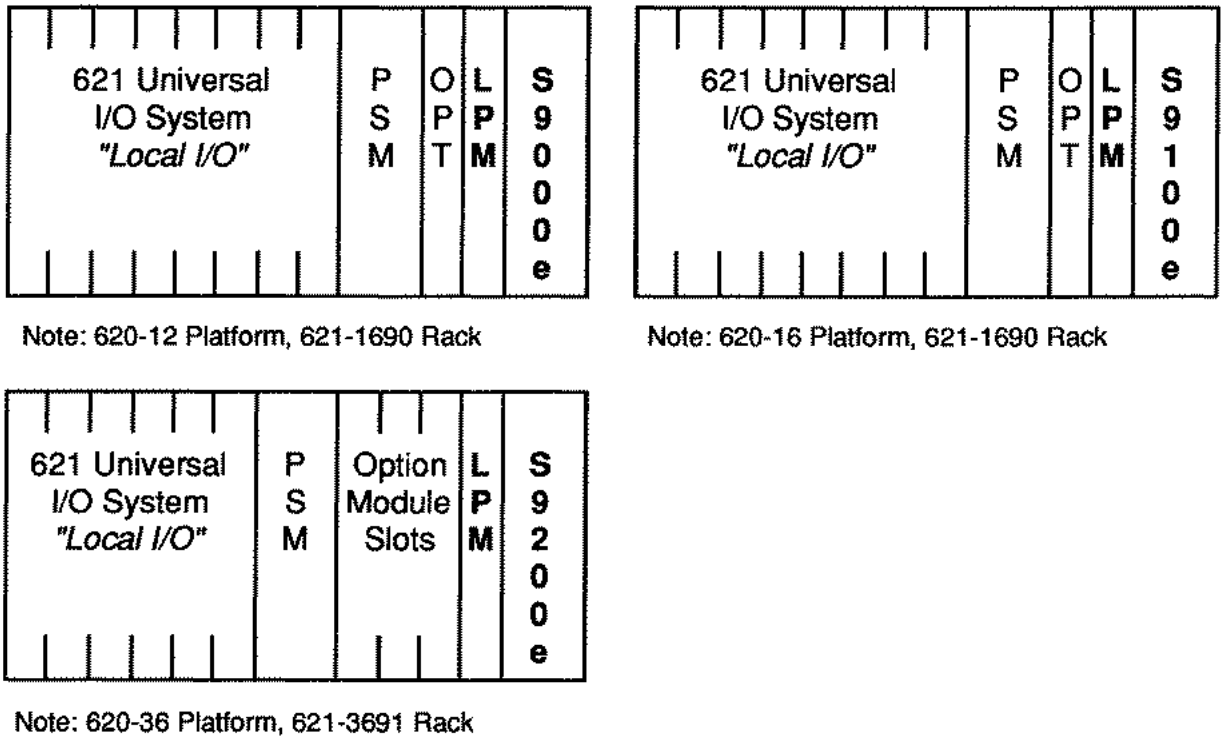


FIGURE 4 - SERIES 9000 CONFIGURATIONS ON THE 620 PLATFORM

NOTE

The Series 9000 Loop and Logic Controllers use the second generation 620 Platform as follows:

SE9000 — 620-15 platform
SE9100 — 620-25 platform
SE9200 — 620-35 platform

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

MiniCOP IN PROCESSOR RACK OF A 620-36 LCS

Figure 5 illustrates a MiniCOP installed in the processor rack I/O slots of a 620-36 LCS which is providing parallel I/O control. This MiniCOP can sample register and ON/OFF conditions from the 620 LCS that pertain to its I/O operations. The MiniCOP is also interfaced to a weigh scale and a barcode reader that are part of the same process that the parallel I/O is controlling. Based on a user-developed program, the MiniCOP can use the data provided collectively from the 620 LCS, the weigh scale, and the barcode reader to determine the operations that must take place in order to achieve the desired results. The MiniCOP may also provide periodic printouts of the ongoing operation via the connected serial printer.

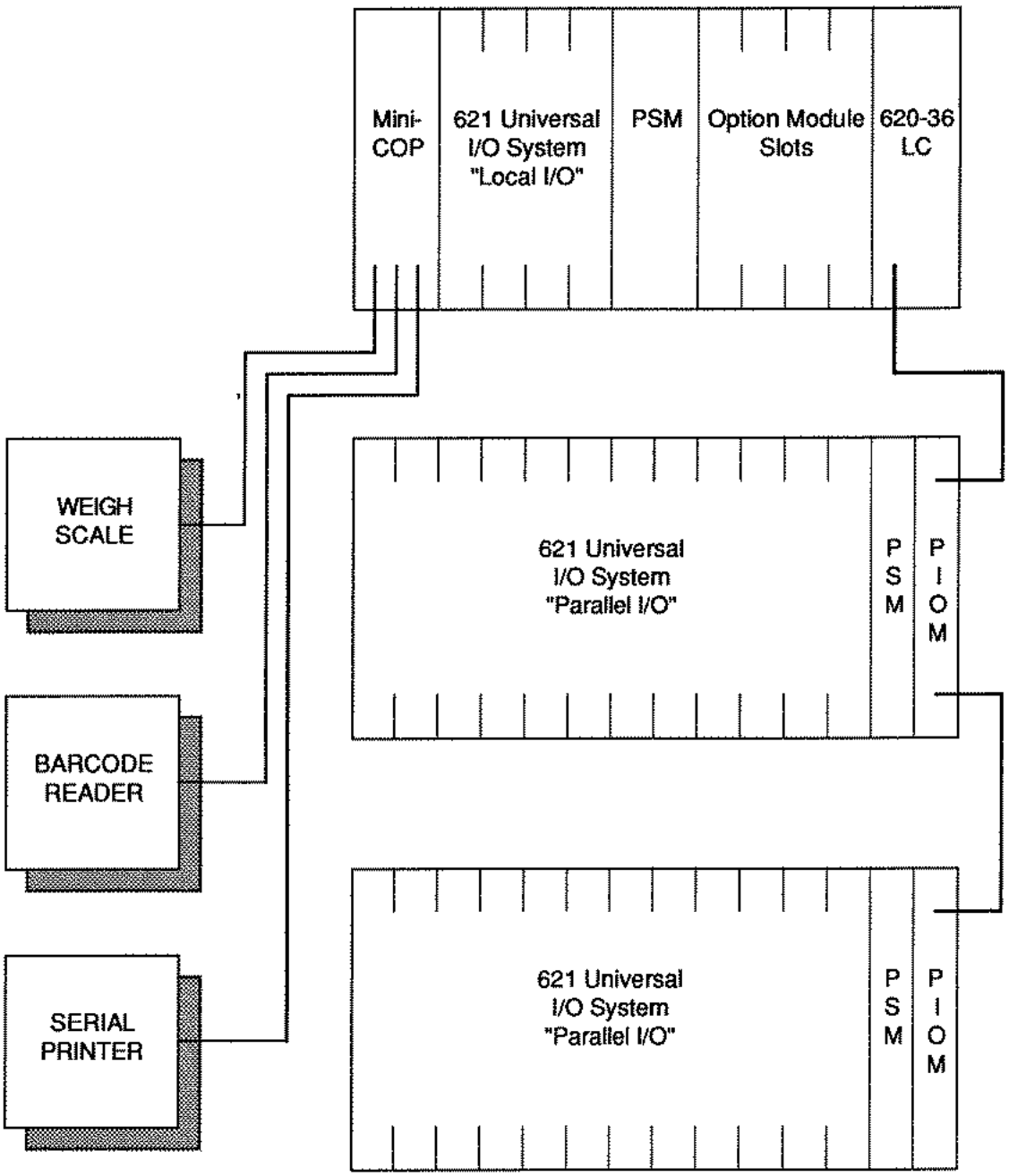


FIGURE 5 - MiniCOP IN PROCESSOR RACK OF A 620-36 LC

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

MiniCOP IN SERIAL I/O RACK OF LOGIC MANAGER

Figure 6 shows a MiniCOP installed in the serial I/O rack of a Logic Manager. This MiniCOP is interfaced to a self-contained operator station on the plant floor and to several DMCS*-connected controllers. It is also configured to access data from the Logic Manager which pertains to the serial I/O operations or which has been passed to the LM from the TDC 3000-connected boxes. In this setup the MiniCOP might be monitoring the DMCS network and then making this information available to the TDC 3000 through the LM, or the TDC 3000 may be providing data to the MiniCOP (through the LM) which then passes it along to the DMCS-connected devices. The operator station that is connected to the MiniCOP may be used by plant floor employees to monitor the ongoing operation and to provide input if and when necessary.

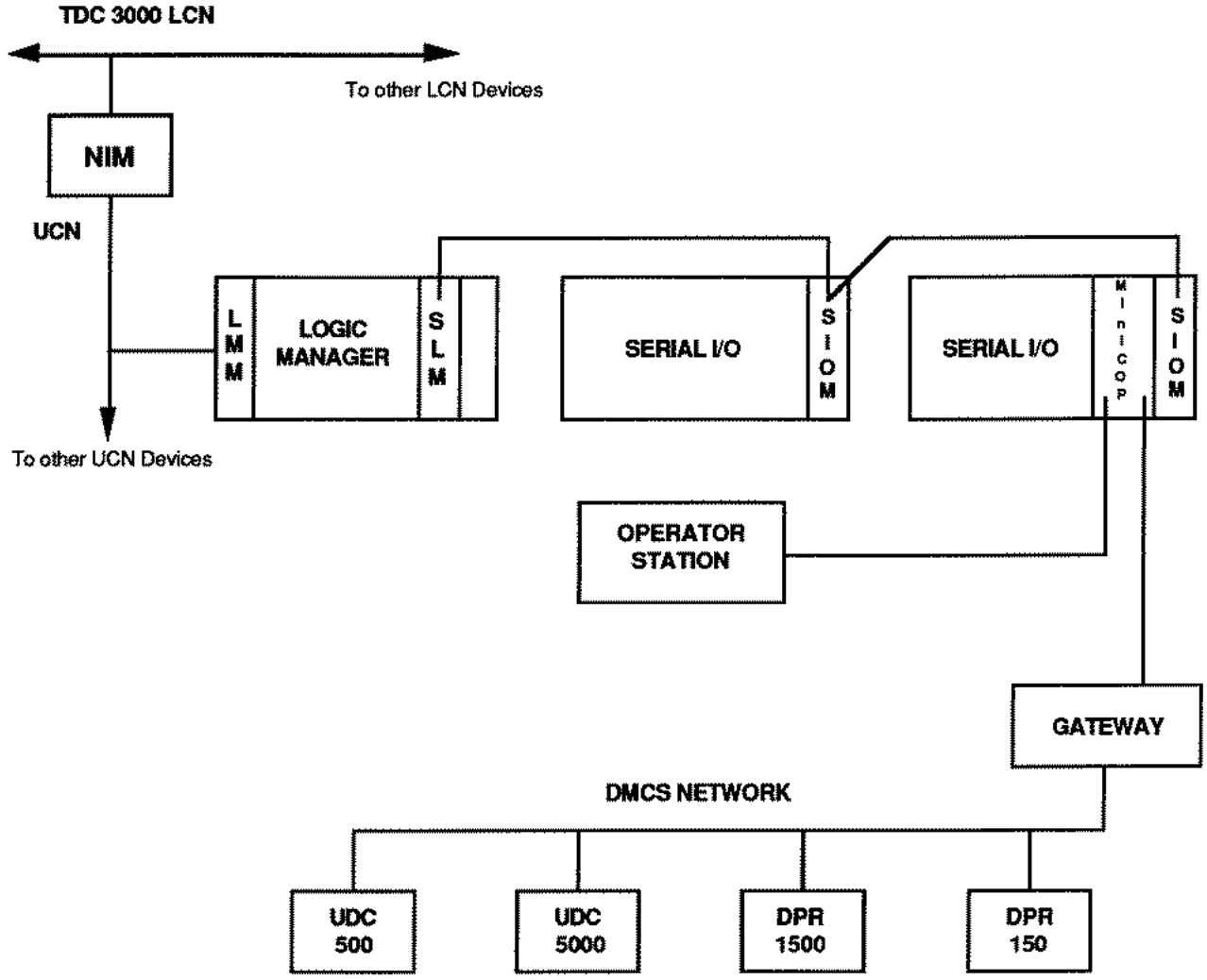


FIGURE 6 - MiniCOP IN SERIAL I/O RACK OF LOGIC MANAGER

* DMCS = Distributed Manufacturing Control System.

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

MiniCOP/ALLEN-BRADLEY PLCs/HONEYWELL 620-35 LCS/TDC 3000

Figure 7 shows a MiniCOP being used to interface two Allen-Bradley PLCs to a Honeywell 620-35 LCS, which in turn is connected to the TDC 3000 via the Hiway Interface Module (HIM) and the Data Hiway. For this particular configuration it has been determined that:

- Another operation will be added to the process that will require an additional PLC,
- All plant operations are to be integrated through the TDC 3000, and
- The new operation will be controlled by a 620 LCS with parallel I/O.

The MiniCOP can aid in facilitating the integration of the existing PLC systems into the new plantwide control system. Using the proper Personality Procedures, data can be passed back and forth between the 620 LCS and the Allen-Bradley PLCs. Since the 620 LCS is interfaced to the TDC 3000, it is possible to either share this information with any other connected boxes or to pass information or commands down from this higher-level order of control.

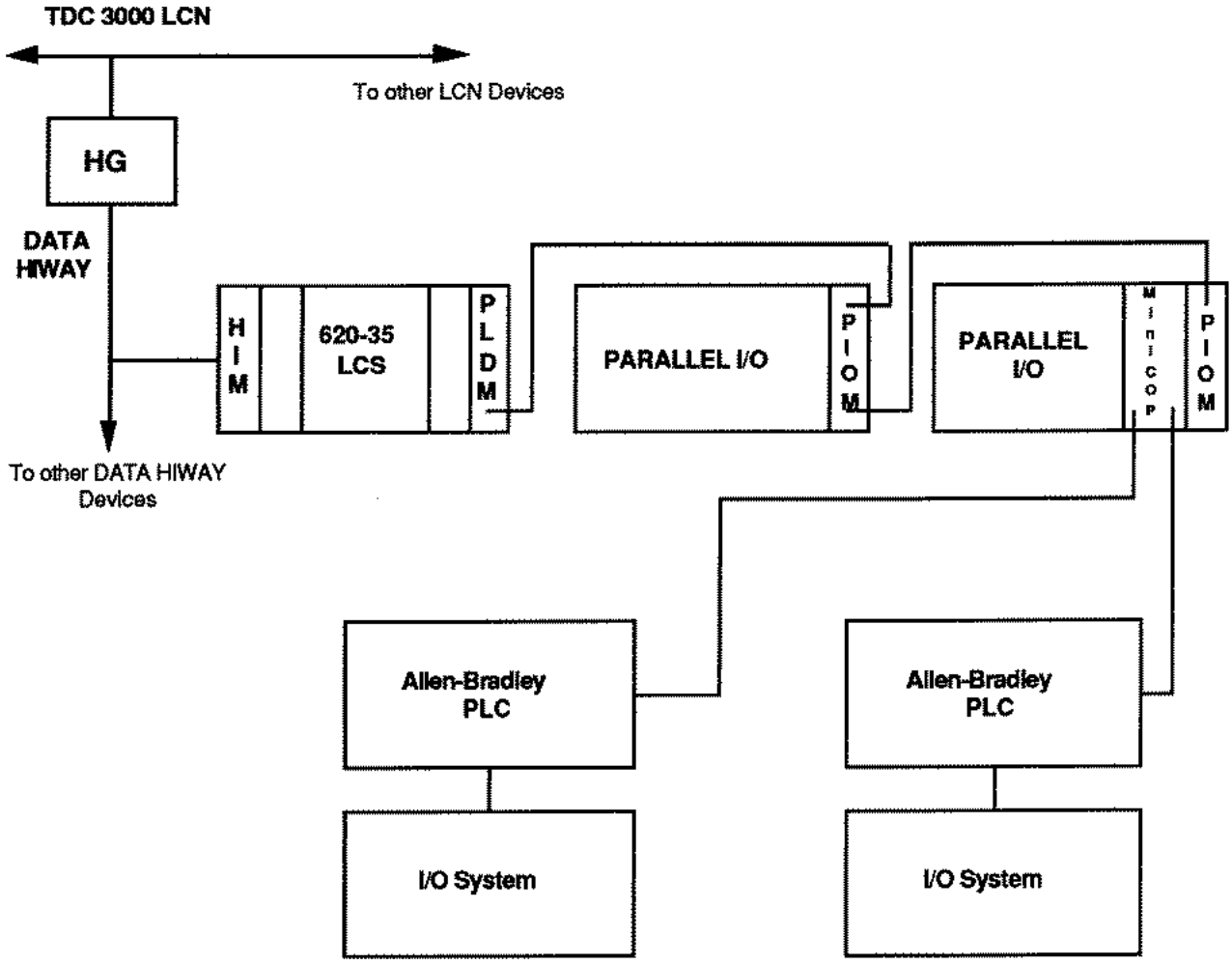


FIGURE 7 - MiniCOP/ALLEN-BRADLEY PLCs /HONEYWELL 620-35 LCS/TDC 3000

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

MiniCOP USED AS "STAND-ALONE" DEVICE

Figure 8 shows a "stand-alone" configuration which is not a common application but is entirely possible. In this configuration, the MiniCOP is used as a means to interface two or more intelligent field devices that are incompatible with each other but which can communicate with the MiniCOP. In the particular configuration shown in Figure 8, a MiniCOP module is installed in a half-size (10-inch) I/O rack along with a Power Supply Module (PSM); all other slots are blank. Through its serial ports the MiniCOP is interfaced to both an Allen-Bradley PLC and a Modicon PLC. Using the proper Personality Procedure, the MiniCOP can act as a mailbox between the two PLCs by taking information from one PLC and passing it on to the other. In addition, the MiniCOP can also provide a higher level of control by monitoring data from both PLCs and issuing commands, error flags, etc.

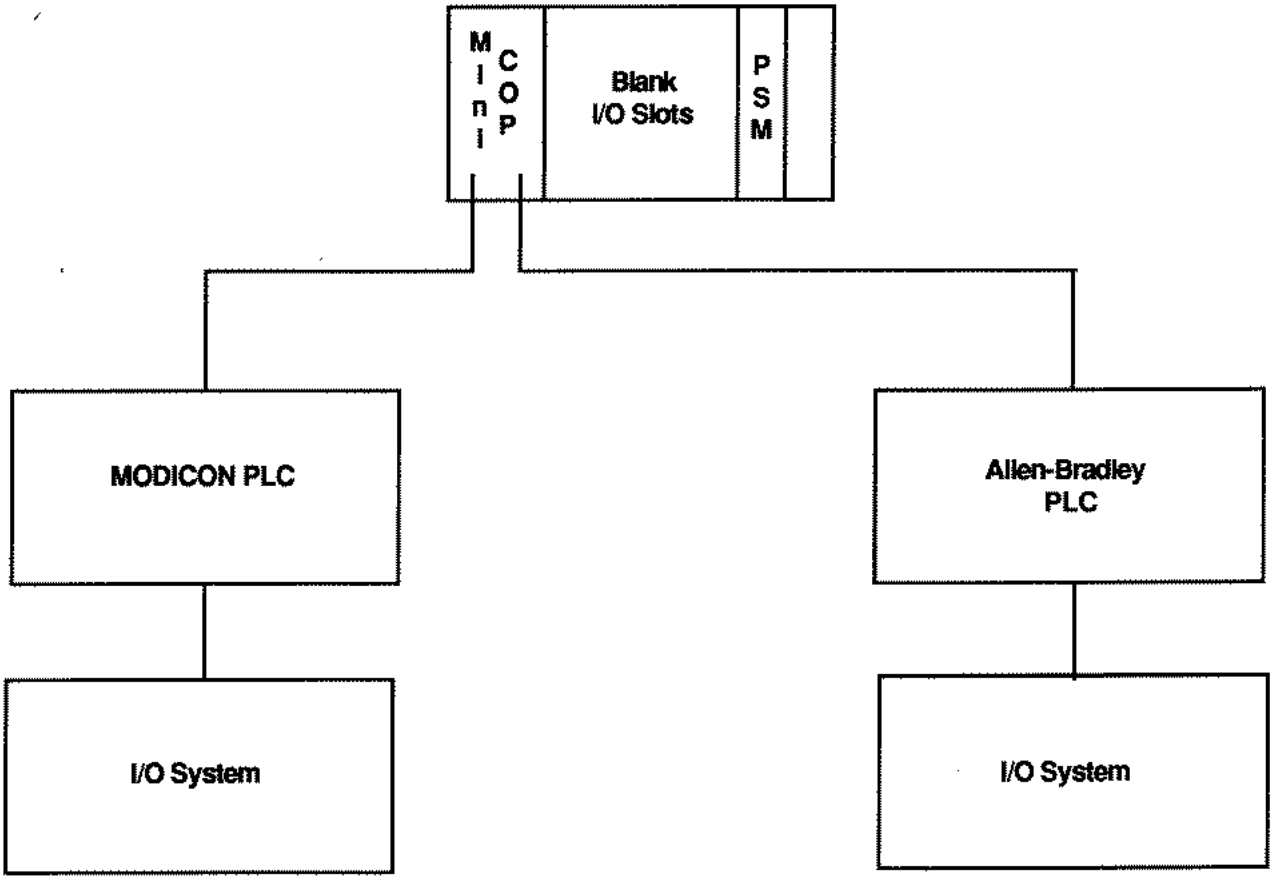


FIGURE 8 - MINICOP USED AS "STAND-ALONE" DEVICE

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

627 MiniCOP COMPONENTS, OPTIONS, & RELATED PRODUCTS

Table 1 lists the various items that are associated with the 627 MiniCOP Systems.

TABLE 1 - 627-1002 AND 627-1003 MINICOP COMPONENTS, OPTIONS, AND RELATED PRODUCTS

CATEGORY	MODEL NUMBER	DESCRIPTION
Modules	627-1002	MiniCOP Module
	627-1003	MiniCOP Module with Video
Software	627-0005 [1,2]	OS-9 Systems Commands Tape
	627-0006[2]	OS-9 Systems Commands Disk (3.5 in. and 5.25 in.)
	626-1011[1]	MiniCOP/620 Personality Procedure Tape
	626-1012	MiniCOP/620 Personality Procedure Disk
	626-1013[1]	MiniCOP/Allen-Bradley Personality Procedure Tape
	626-1014	MiniCOP/Allen-Bradley Personality Procedure Disk
	626-1016	MiniCOP/Modbus Personality Procedure Disk
	626-1018	MiniCOP Assembly Programming Disk
	623-6041	Terminal Emulator Disk
Manuals	627-8991[3]	627 MiniCOP and BASIC09 User Manual
	627-8993	OS-9/627 Hardware Users Manual
	627-8997	627 70 IPC620 Personality Procedure User Manual
	626-9500	627 70 Modbus Personality Procedure User Manual
	626-9501	627 70 AB Data Highway Personality Procedure User Manual
	623-8989	623 Terminal Emulator User Manual
	626-9531	627 MiniCOP Assembly Software User Manual
Misc.	628-3006[1]	MiniCOP to 623 RS232 Cable (10 feet)
	629-3002	AA Size Lithium Battery
	629-1021	0.1 Amp Fast-Blo Fuse, 5 x 20mm

[1]For use with the 623-50/51 Loader Terminal only.
[2]Includes BASIC09 and RUNB *and* is included with both the 627-1002 and 627-1003 MiniCOP modules.
[3]Included with both the 627-1002 and 627-1003 MiniCOP Modules.

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

HARDWARE CHARACTERISTICS

627-1002 AND 627-1003 MiniCOP SPECIFICATIONS

MICROPROCESSOR	68B09E operating at 1.84 MHz
REAL TIME CLOCK BATTERY BACKUP	9-month continuous at full memory, AA size lithium (Model No. 629-3002)
MEMORY	16K EPROM (operating system elements) 48K workspace (user program workspace); 256K CMOS RAM (user program storage, COPRAM)
FUSE	100mA, 250V
POWER	**627-1002:** 5VDC ± .25VDC @ 3.8A +15VDC ± .48VDC @ .285A –15VDC ± .48VDC @ .285A **627-1003:** 5VDC ± .25VDC @ 5.5A +15VDC ± .48VDC @ .379A -15VDC ± .48VDC @ .379A
ASYNC PORT INTERFACE	RS232, RS422/485, 20mA (DIP switch selectable)
BAUD RATES	150 to 19.2K for RS232 & RS422/485 (DIP switch selectable); 9600 max. for 20mA (DIP switch selectable)
DATA WORD	7 or 8 data bits 1 or 2 stop bits Even, odd, or no parity (all DIP switch selectable)
SERIAL CONNECTORS	**627-1002:** Three 25-pin subminiature D-type female **627-1003:** Four 25-pin subminiature D-type female; one modular keyboard jack
OPERATING TEMPERATURE	32°F to 140°F (0°C to 60°C)
STORAGE TEMPERATURE	104°F to 185°F (40°C to 85°C)
HUMIDITY	5% to 95% (noncondensing)
NOISE TEST	NEMA 2-230, IEEE 472-1974, VDE 0871 Class A
ELECTRICAL CODES	UL, CSA OL, VDE
VIDEO CONNECTORS (627-1003 ONLY)	Four BNC connectors (3 color, 1 composite)
VIDEO SIGNAL STANDARD (627-1003 ONLY)	RS170 (analog)

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

MAJOR COMPONENTS

MODULE FRONTPLATES

Figure 9 illustrates the frontplates of both the 627-1002 and 627-1003 MiniCOP modules. Notice from this figure that the left two thirds of the 627-1003 MiniCOP is identical to the entire 627-1002 unit. The differences between the two revolves around the third circuit board, which is located behind the right third portion of the 627-1003 MiniCOP frontplate.

NOTE

Refer to the remainder of this section for detailed descriptions of the connectors, LEDs, switches, etc.

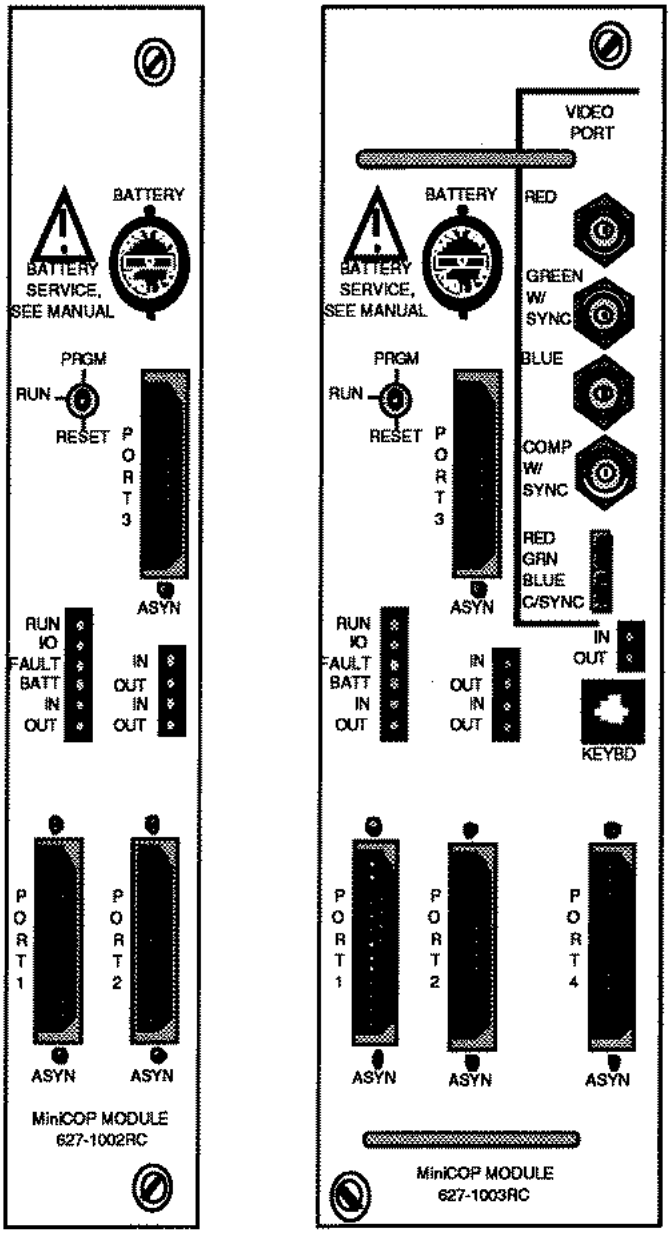


FIGURE 9 - 627-1002RC AND 627-1003RC MINICOP MODULE FRONTPLATES

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

CIRCUIT BOARDS

The 627-1002 and 627-1003 are multiple-board modules. The 627-1002 MiniCOP consists of two circuit boards and requires two I/O slots when installed in a 620 Platform system. The 627-1003 MiniCOP consists of three circuit boards and requires three I/O slots when installed in a 620 Platform system.

627-1002 MiniCOP

The 627-1002 is a dual-board module. As shown in Figure 10, the left-hand board is the CPU board and the right-hand board is the PORT/MEMORY board.

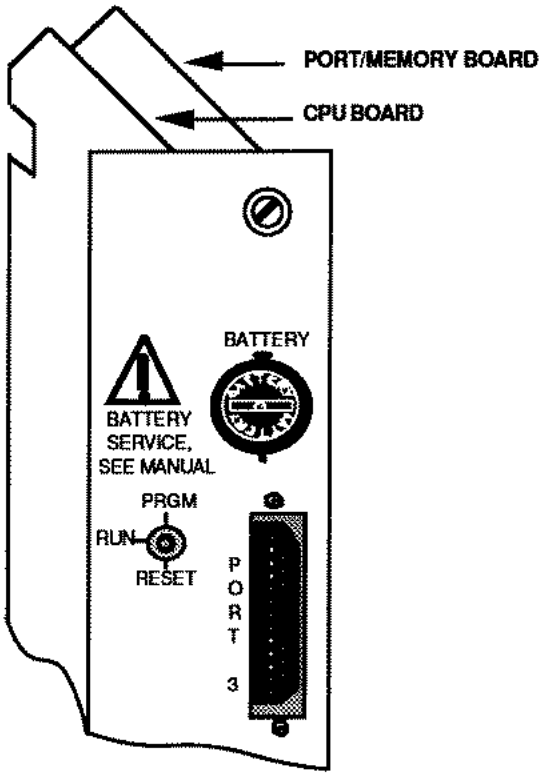


FIGURE 10 - 627-1002 CIRCUIT BOARDS

The CPU board contains the microprocessor, the 16K ROM operating system, and the 48K RAM workspace. It also contains the battery's, electrical connection, fuse, mode switch, a bank of LEDs, and one ASYN 25-pin connector (Port 1).

The PORT/MEMORY board contains the 256K CMOS RAM program storage area known as COPRAM. It also contains a bank of LEDs and two ASYNC 25-pin connectors (Ports 2 and 3). While the battery appears to be part of the PORT/MEMORY board based on its location on the frontplate, there is actually no direct electrical or mechanical connection taking place.

The 627-1002 MiniCOP can be inserted into any two I/O slots that accept PUSH/PULL modules. The MiniCOP can communicate to its "host" 620 Platform processor on the serial or parallel I/O bus. PUSH/PULL ladder logic instructions are used by the processor to send and retrieve data to/from the MiniCOP. The term "host" refers to the 620 Platform controller in which the MiniCOP resides. The MiniCOP can be inserted into any standard full- or half-size I/O rack, augmented I/O rack, or processor rack I/O slots.

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

627-1003 MiniCOP

The 627-1003 is a triple-board module (see Figure 11). The left and center boards are identical to the 627-1002 MiniCOP; the third or right-hand board is the VIDEO board.

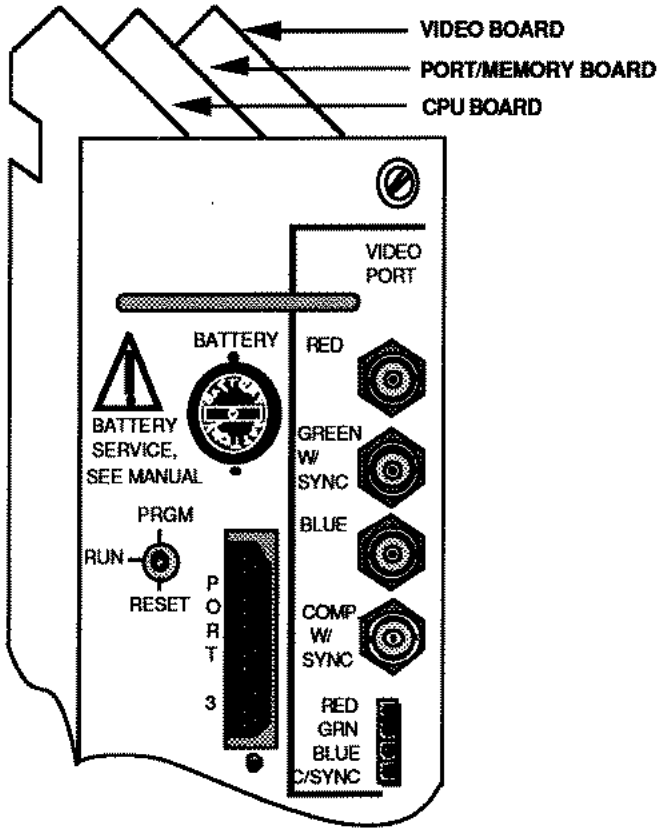


FIGURE 11 - 627-1003 CIRCUIT BOARDS

The CPU board contains the microprocessor, the 16K ROM operating system, and the 48K RAM workspace. It also contains the batteries, electrical connection, fuse, mode switch, a bank of LEDs, and one ASYN 25-pin connector (Port 1).

The PORT/MEMORY board contains the 256K CMOS RAM program storage area known as COPRAM. It also contains a bank of LEDs and two ASYN 25-pin connectors (Ports 2 and 3). While the battery appears to be part of the PORT/MEMORY board based on its location on the frontplate, there is actually no direct electrical or mechanical connection taking place.

The VIDEO board contains the video circuitry including four BNC connectors. The top three BNC connectors provide RGB color video output, while the bottom provides a composite monochrome signal. There are also a bank of LEDs, one modular keyboard jack, and one ASYN 25-pin connector (Port 4). Because the two ports (ASYN port and keyboard jack) use the same interface hardware, they may not be used simultaneously.

The 627-1003 MiniCOP can be inserted into any three I/O slots that accept PUSH/PULL modules. The MiniCOP can communicate to its "host" 620 Platform processor on the serial or parallel I/O bus. PUSH/PULL ladder logic instructions are used by the processor to send and retrieve data to/from the MiniCOP. The term "host" refers to the 620 Platform controller in which the MiniCOP resides. The MiniCOP can be inserted into any standard full or half size I/O rack, augmented I/O rack, or processor rack I/O slots.

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

BATTERY

A lithium battery (AA size, Model No. 629-3002) is required to maintain the system's CMOS RAM memory during power-down. The battery is located on the PORT/MEMORY board and is rated for nine-months of continuous backup at full memory. Under normal operation the battery has a rated life of ten years. If the BATT LED is not lit when the MiniCOP is under power, the battery must be replaced. Access to the battery is gained by way of a round black cover as illustrated in Figure 9.

CAUTION

Replace the battery only while the module is under AC power. Failure to do so will result in lost or corrupted memory. Depressing the battery cover may open the battery circuit, also resulting in a loss or corruption of memory if the MiniCOP is not under AC power.

Battery replacement and servicing of battery-related circuitry should be performed by a qualified technician.

TOGGLE SWITCH

A two-way toggle switch, located on the front panel of the unit (see Figure 9), is used to select the operating mode for the system. The toggle switch can be DIP switch configured on the CPU board to perform one of two basic functions. One configuration allows the user to initiate or reset the MiniCOP's OS-9 operating system. The second option is to allow the user to toggle between user programs called PRGMA and PRGMB.

The toggle switch can also be disabled via a jumper selection on the CPU Board.

NOTE

Refer to the *Configuration/Installation* subsection for DIP switch and jumper setting information.

LED INDICATORS

The MiniCOP has several banks of LEDs on the frontplate, two banks on the 627-1002 and three on the 627-1003 as shown in Figure 9.

IN/OUT LEDs

The modules' ASYN ports each have dedicated IN and OUT LEDs. These green LEDs indicate receive and transmit status for the serial ports. A port's IN LED only illuminates when the MiniCOP is receiving transmission through the port from a connected device. The OUT LED only illuminates when the MiniCOP is transmitting through the port to a connected device.

RUN LED

The RUN LED is lit when the module is in the RUN Mode. This green LED indicates that the module has passed its internal self-test operation and is executing a user program. If there is no user program, the MiniCOP will execute the executive program. If the RUN LED is not lit (OFF) and the FAULT LED is lit (ON), the processor has failed the self-test.

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

FAULT LED

The red FAULT LED is lit if a fault is detected in the operation of the module. Under normal operating conditions the LED is not lit.

BATT LED

The BATT LED is a green LED that indicates the battery is in good condition when lit. If the LED is not lit when the MiniCOP is under AC power, the battery must be replaced. Replace the battery only when the MiniCOP is under AC power. Failure to do so may corrupt or clear memory.

25-PIN ASYN PORTS

The 627-1002 MiniCOP has three 25-pin subminiature female asynchronous communications ports while the 627-1003 has four, as shown in Figure 9. These ports are used to communicate with serial devices, such as printers, data storage devices, barcode readers, and other factory floor control and sensing devices (including PLCs). The ports are compatible with RS232, RS422/485, and 20mA electrical interface standards (DIP switch selectable). The ports can also be DIP switch selected for full- or half-duplex transmission and point-to-point or multidrop configuration. Port baud rate and data word characteristics are software selectable. Port 1 DIP switch selections are made on the CPU board; Ports 2 and 3 on the PORT/MEMORY board; and Port 4 on the VIDEO board.

NOTE

Refer to the *Configuration/Installation* subsection for DIP switch setting information and to *Appendix B* for more information on cable construction and configuration.

MODULAR KEYBOARD PORT

The modular keyboard port (labeled KEYBD as shown in Figure 9) acts as an interface to the 623-51 Loader/Terminal keyboard. This port can be used for the 627-1003 MiniCOP in place of the 25-pin ASYN Port 4 to provide for direct keyboard entry. A DIP switch selection on the Video Board determines which of the two ports (KEYBD or Port 4) are enabled. Only one can be enabled at any given time.

NOTE

Refer to the *Configuration/Installation* subsection for DIP switch setting information and to *Appendix B* for information on cable construction and configuration. Refer to the *623 Loader/ Terminal User Manual* for information on the 623-51 Loader/Terminal keyboard.

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

VIDEO PORTS

The triple-wide 627-1003 MiniCOP can be used to send either monochrome or color RGB video signals to a remote monitor. Four female BNC-type connectors (video ports) are located on the frontplate for this purpose as shown in Figure 9. The RGB video signal is provided in the RS-170 electrical standard.

The bottom BNC port provides a composite monochrome signal, including vertical and horizontal synchronization. The top three BNC ports provide a color signal. Vertical and horizontal synchronization are combined with the GREEN signal.

Monitors may be connected to the ports via RG58 coaxial cable up to a distance of 1000 feet (300 meters).

NOTE

Refer to the *Video Cables* portion of *Appendix B* for information on 627-1003 MiniCOP video port cable construction.

Four potentiometers are available to adjust the video signal levels. These potentiometers may be accessed from the frontplate and are located just beneath the VIDEO PORT label as shown in Figure 9.

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

CONFIGURATION/INSTALLATION

SHIPPING CONFIGURATION

Each MiniCOP is shipped with standard factory settings. Note that these settings may not be those recommended for your particular application. Both MiniCOP models are also shipped with the following components:

- 25-pin male connector kits for each ASYN port (Port 3 or 4),
- OS-9 Systems Commands tape and OS-9 System Commands Disks (3-1/2 in. and 5-1/4 in. copies), and
- One *627 MiniCOP and BASIC09/RUNB User Manual.*

NOTE

The module toggle switch is shipped ENABLED. This switch can be DISABLED by changing the position of jumper W1 on the CPU board as described in this subsection.

The module battery is shipped installed, but DISABLED. In order to have the battery backup for the MiniCOP's RAM memory, the battery must be ENABLED as described in this subsection.

Each ASYN port is set for RS232, point-to-point/full duplex communication. The remaining ASYN port parameters are software selectable. These parameters are factory-set so that each port is configured for 9600 baud with 8 data bits, 1 stop bit, and no parity. You can change these settings via the Terminal Emulator software package.

NOTE

Refer to the *Terminal Emulator* section of this manual or the *623 Terminal Emulator User Manual* for additional information regarding the ASYN port software-selectable functions.

DIP SWITCH SELECTABLE FEATURES

CPU BOARD

The CPU board has a single 8-position DIP switch located at the top of the circuit board, as shown in Figure 12. Using this DIP switch Port 1, the battery, toggle switch, and serial or parallel I/O rack installation can be selected. Table 2 lists the functions of each switch position. Each position is either CLOSED/ON (1) or OPEN/OFF (0).

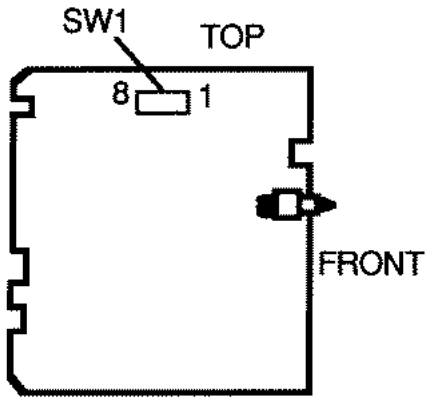


FIGURE 12 - CPU BOARD DIP SWITCHES

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

TABLE 2 - CPU BOARD DIP SWITCH FUNCTIONS

Position	Setting		Function	
1,4	1	4		
	Open/OFF	Open/OFF	Not Used	Port 1
	Open/OFF	Closed/ON	RS422/485	
	Closed/ON	Open/OFF	20mA	
	Closed/ON*	Closed/ON*	RS232	
2	Open/OFF		Not Connected	RS422/485 Receiver/Termination
	Closed/ON*		Connected	Resistor (150Ω)Port 1
3	Open/OFF		Multidrop/Half Duplex	Port 1
	Closed/ON*		Point-to-Point/Full Duplex	
5	Open/OFF*		Battery OFF	
	Closed/ON		Battery ON	
6	Closed/ON*		Not Set by User	
7	Open/OFF		Power Up to OS-9, toggle any direction resets OS-9	
	Closed/ON*		Power Up to PRGMB, toggle up to PRGMA, toggle down to PRGMB	
8	Open/OFF		Serial I/O Rack	
	Closed/ON*		Parallel I/O Rack	

*Factory Setting

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

PORT/MEMORY BOARD

The PORT/MEMORY board has one 4-position DIP switch and one 8-position DIP switch located at the top of the circuit board as shown in Figure 13. Using these DIP switches, Ports 2 and 3 can be configured. Tables 3 and 4 list the functions associated with each position.

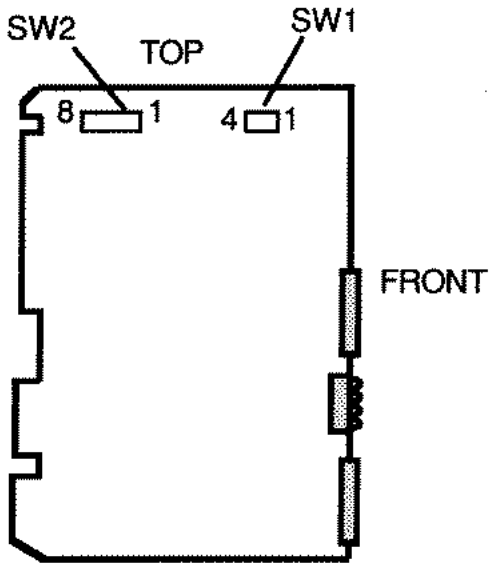


FIGURE 13 - PORT/MEMORY BOARD DIP SWITCH LOCATIONS

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

TABLE 3 - PORT MEMORY BOARD DIP SWITCH SW1 FUNCTIONS

Position	Setting	Functions	
1	Open/OFF	RS422/485	Port 2
	Closed/ON*	RS232	
2	Open/OFF	Not Connected	RS422/485 Receiver Terminator
	Closed/ON*	Connected	Resistor (150Ω)—Port 2
3	Open/OFF	Multidrop/Half-Duplex	Port 2
	Closed/ON*	Point-to-Point/Full Duplex	
4	Open/OFF	RS422/485	Port 3
	Closed/ON*	RS232	

*Factory Setting

TABLE 4 - PORT/MEMORY BOARD DIP SWITCH SW2 FUNCTIONS

Position	Setting	Functions	
1	Open/OFF	Not Connected	RS422/485 Receiver Termination
	Closed/ON*	Connected	Resistor (150Ω)-Port 3
2	Open/OFF	Multidrop/Half-Duplex	Port 3
	Closed/ON*	Point-to-Point/Full Duplex	
3,4,5,6	Not Set by User **DO NOT ALTER** Factory setting should be: Position 3,4,5,6 set ON		
7	Open/OFF	20mA Enabled set SW1 Position 4 to Closed /On	Port 2
	Closed/ON*	20mA Disabled	
8	Open/OFF	20mA Enabled set SW1 Position 4 to Closed/On	Port 3
	Closed/ON*	20mA Disabled	

*Factory Setting

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

VIDEO BOARD

The VIDEO board has one 8-position DIP switch located at the top of the circuit board as shown in Figure 14. Using either this DIP switch, Port 4, or the modular keyboard jack, Port 4 and the video line mode can be configured. Table 5 lists the functions associated with each position.

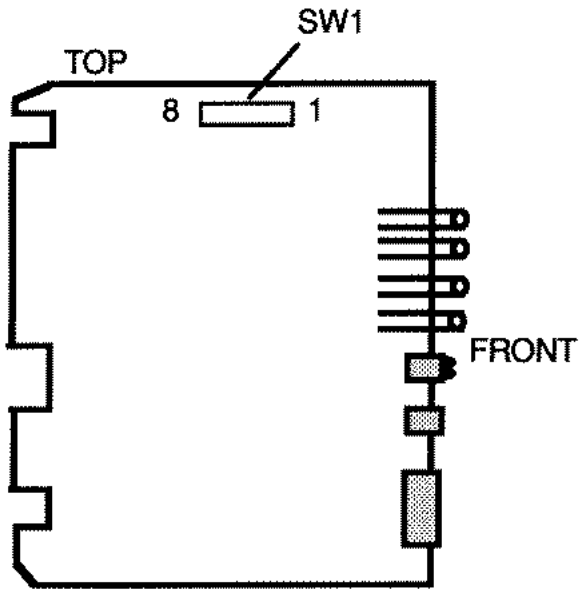


FIGURE 14 - VIDEO BOARD

TABLE 5 - VIDEO BOARD DIP SWITCH FUNCTIONS

Position	Setting		Function	
1,4	1	4		Port 4
	Open/OFF	Open/OFF	Keyboard Jack	
	Open/OFF	Closed/ON	RS422/485	
	Closed/ON	Open/OFF	20mA	
	Closed/ON*	Closed/ON*	RS232	
2	Open/OFF		Not Connected	RS422/485 Receiver Termination
	Closed/ON*		Connected	Resistor (150Ω)-Port 4
3	Open/OFF		Multidrop/Half Duplex	Port 4
	Closed/ON*		Point-to-Point/Full Duplex	
5	Open/OFF*		Not Used	
6	Open/OFF*		24 line mode	
	Closed/ON		48 line mode	
7,8	7	8	Not Used — Do not alter	
	Open/OFF*	Open/OFF*		

*Factory Setting

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

JUMPER-SELECTABLE FEATURES

The MiniCOP has one user-configurable, jumper-selectable feature. The toggle switch on the frontplate can be ENABLED or DISABLED using the three-pin jumper W1. Jumper W1 is located on the CPU board directly behind the toggle switch, as shown in Figure 15. The protective cover must be removed from the CPU board-side of the MiniCOP module in order to gain access to this jumper.

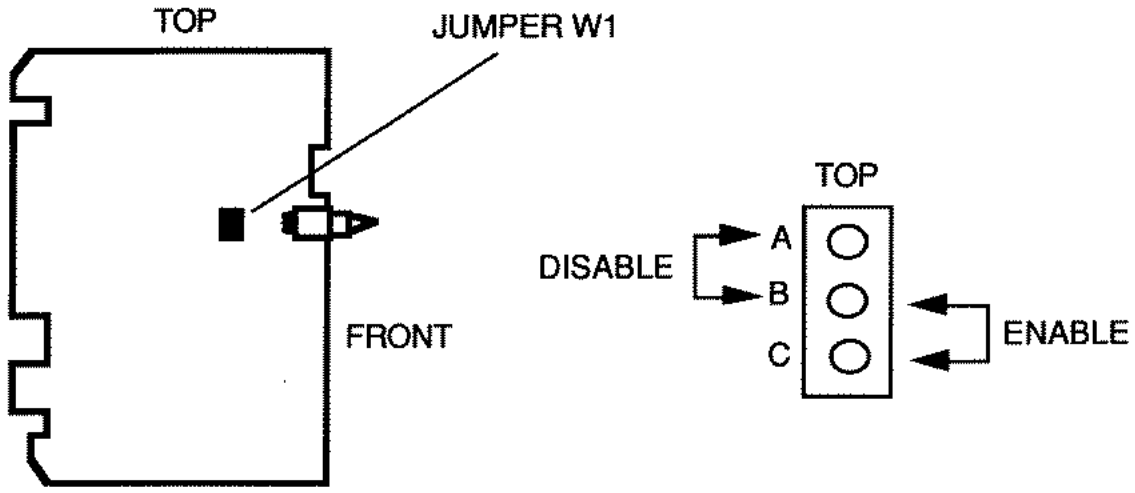


FIGURE 15 - CPU BOARD JUMPER LOCATION

The toggle switch is ENABLED by placing the jumper over pins B and C, or the two pins nearest the bottom of the module. It is DISABLED by placing the jumper over pins A and B, or the two pins nearest the top of the module.

ASYN 25-PIN CONNECTOR CABLE

NOTE
Refer to *Appendix B* for detailed information on the asynchronous, 25-pin connector cable.

VIDEO PORT CABLE

NOTE
Refer to *Appendix B* for detailed information regarding the VIDEO port cable.

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

GENERAL INSTALLATION NOTES

RECEIPT INSPECTION

Each 620 System order is shipped with component parts individually packaged in labeled boxes. The **Product Registration Label** containing the **Model Number** and **Serial Number** of the component is affixed to the shipping carton. Remove and return the label to Honeywell at the noted address to ensure follow-up service and support.

The 620 LCS components are fully inspected, tested, and properly packaged before leaving the factory. They should be inspected, however, for any damage that may have occurred during shipment. If any damage is evident, note it on the carrier's freight bill and make a written request for inspection by the carrier's agent.

PREINSTALLATION CONSIDERATIONS

1. Inspect the system components as they are received.

2. Carefully review the literature shipped with the product, paying close attention to any special precautions.

3. Determine the types of logic controllers, I/O modules, special function modules, and other subsystems or peripheral devices required. Lay out each rack and consider such factors as power consumption and heat dissipation. (Much of this is done prior to purchasing the system — this manual provides a brief review of the procedure for reference.)

4. Determine the starting address of each rack in the system that contains I/O slots (including the processor rack when applicable).

INSTALLATION CHECKLIST

1. Carefully review all literature enclosed with the product, paying close attention to any cautionary notes.

2. Install the 620 Platform controller hardware in accordance with the current edition of the *National Electrical Code ANSI/NFPA 70.*

3. Have all 620 Platform components serviced only by qualified technicians.

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

PRECAUTIONARY STEPS FOR SAFE AND PROPER INSTALLATION

Ensure optimum system performance and operator safety by adhering to the following precautionary steps for safe and proper installation:

System Component Grounding ⏚

1. Follow the system grounding setup as shown in Figure 16. System ground is established in two ways:

   - protective ground (earth) is provided by the power supply protective earth terminal which is mandatory, and
   - functional ground (earth) for non-safety purposes such as noise immunity improvement which is required for proper operation.

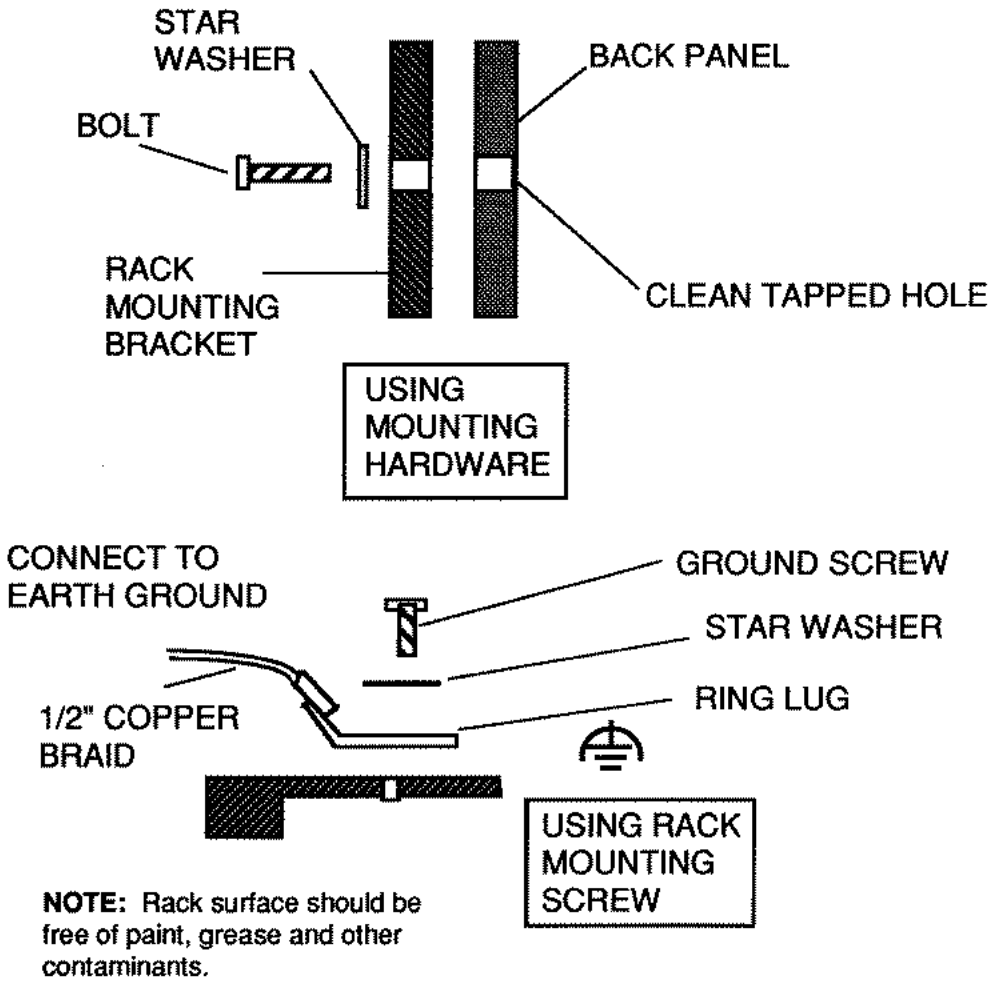


FIGURE 16 - SYSTEM GROUNDING

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

2. Establish system functional ground (earth) by mounting all 620 LCs and I/O racks on a metal back-panel. Connect the back-panel to earth ground using the special rack mounting screws and star washers provided with each 620 rack. Connect the rack/back-panel to the functional earth ground point with 1/2-inch (12 mm) copper braid.

   NOTE: The copper braid should be properly protected with suitable plating and/or supplementary insulation to prevent corrosion.

3. Ensure that all 620 power supply ground terminals are connected to earth ground as shown in Figure 17.

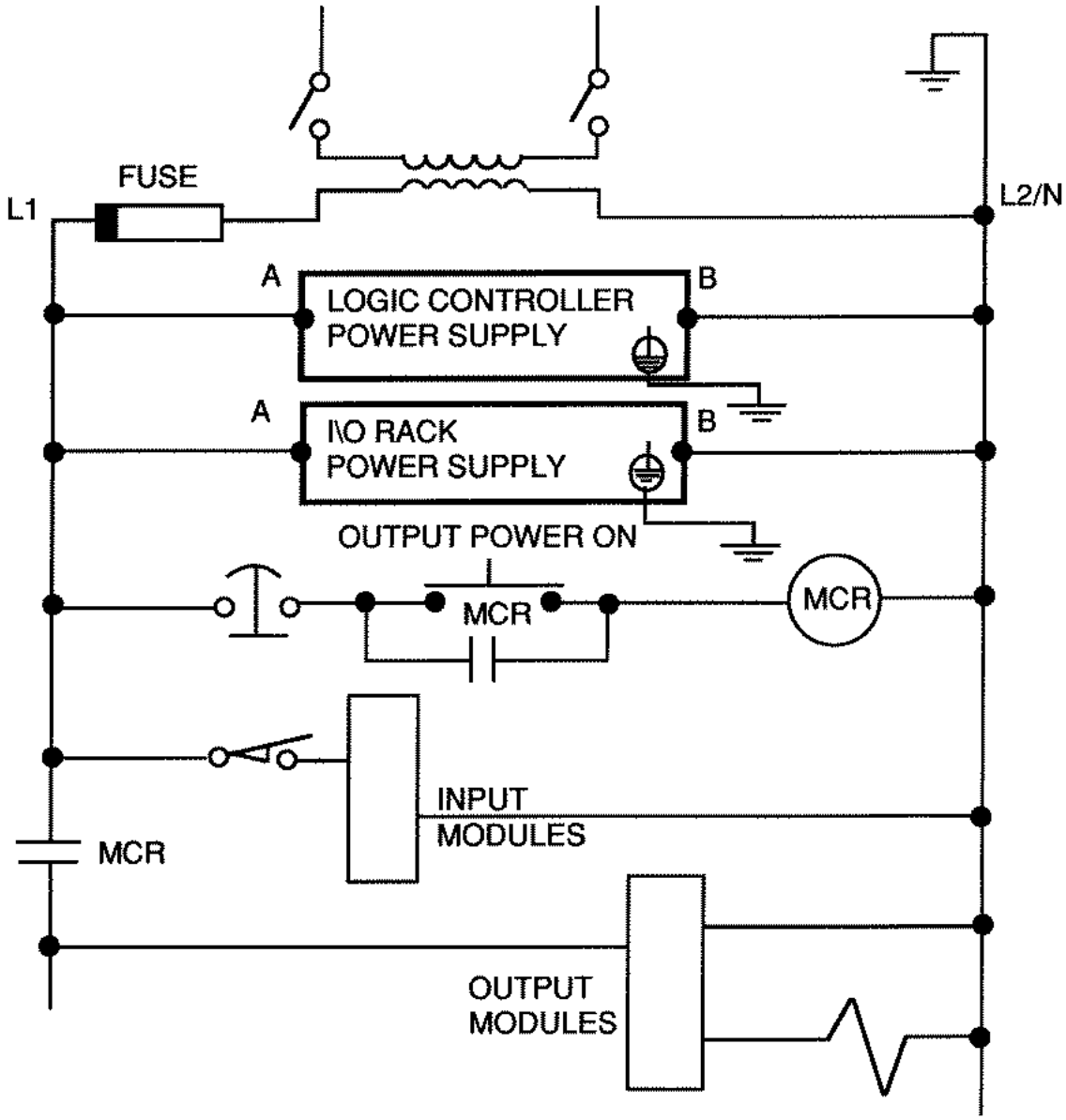


FIGURE 17 - POWER SUPPLY GROUNDING

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

4. Select the proper voltage, 115VAC or 230VAC (on those modules that allow such a selection), before supplying power to the module.

5. Install and remove modules only when power has been removed from the system.

   Most I/O modules, special modules, and other appropriate Honeywell modules (including the MiniCOP) are currently being updated to allow for insertion and removal while the rack is under power. These modules are identified by the suffix "R" at the end of the model number (e.g., the two MiniCOP modules with this circuitry will carry the model numbers 627-1002RC and 627-1003RC respectively; each model number will be silk-screened on module's frontplate). This function can be performed in serial I/O only. Refer to *Appendix A* for additional details about this operation.

CAUTION

Do not attempt to install or remove modules that do not carry the "R" suffix from any rack while the rack is powered. Doing so may cause damage to the module, may affect the backplane communications, and may possibly turn outputs ON unintentionally for one ladder logic program scan.

6. Ensure that coverplates are installed over any and all blank slots. The coverplates supplied with the 620 Platform system racks and modules help to shield the circuitry from external electromagnetic noise.

7. Securely fasten all module frontplates and blank coverplates in order to maintain system integrity and to ensure system component grounding. Do NOT overtighten.

8. Consult the *IEEE Standard 518-1977*, the *National Electrical Code*, and the *Cable and Conduit Routing* procedures presented in the *620 Installation and 621 I/O Communications User Manual* before wiring system components.

NOTE

Refer to the *620 Installation and 621 I/O Communications User Manual* for detailed information regarding the installation of 620 Platform controllers and I/O systems.

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

GENERAL INSTALLATION PROCEDURES

1. Lay out the rack enclosures and mount the processor and I/O racks.

2. Set each module's DIP switches and jumpers. Many control processors, special modules, and I/O modules have DIP switch and/or jumper-selectable features that must be set, depending on the configuration and required operation.

3. Install modules in the positions already determined during the configuration stage.

4. Add address labels based on: rack starting address, I/O module point size, and the number of points assigned to each slot.

5. Route the cabling within and between enclosures following the prescribed rules for spacing and shielding.

6. Install the I/O terminal blocks.

7. Wire the individual I/O module, special function module, and other modules' field control/sense wiring from the system to the field devices.

8. Attach the AC or DC wiring to the power supplies.

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

MiniCOP INSTALLATION

PREINSTALLATION STEPS

1. Inspect the system components as they are received.

2. Carefully review the literature shipped with the product, paying close attention to any special precautions.

3. Determine the type of rack in which the MiniCOP is to be installed, including standard or augmented rack; and parallel, serial, or processor I/O rack slots (note that parallel and processor I/O rack slots are treated in the same manner).

4. Establish the eight I/O addresses that the MiniCOP module will use.

5. Initialize the MiniCOP for its intended application, including the user-developed BASIC09 or OS-9 Assembly Language program and the loading of any Personality Procedures.

6. Develop and program any ladder logic that may be required to intercommunicate between the host 620 Platform controller and the MiniCOP module.

7. Determine the layout and configuration of cabling needed to intercommunicate between the MiniCOP module and any external control and/or sensing devices.

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

MiniCOP INSTALLATION PROCEDURES

1. Remove the MiniCOP module from its packaging and examine the unit carefully.

2. Select the appropriate DIP switch settings to configure the following parameters:

   - Serial port interface characteristics,
   - Toggle switch operation (PRGMA/PRGMB or OS-9 on power-up),
   - Serial or parallel rack installation.

CAUTION

The maximum combined sum of all MiniCOP port baud rates must not exceed 19.2K. If any one port or group of ports on a single MiniCOP are set at a combined rate of 19.2K baud, no other ports can be serviced reliably. If the baud rate of these ports is decreased, to limit the total baud rate, other ports can be added, provided the maximum baud rate of 19.2K for interrupt processing is not exceeded.

3. Enable the battery.

NOTE

Refer to the *DIP Switch Selectable Features* portion of this subsection for detailed information regarding steps 2 and 3 above.

4. ENABLE or DISABLE the frontplate toggle switch, as appropriate.

NOTE

Refer to the *Jumper-Selectable Features* portion of this subsection for additional information on ENABLING/DISABLING the frontplate toggle switch.

5. Use the following procedure if the unit is to be installed in an augmented rack (that is, a rack with the upper/local bus on its backplane).

   - Carefully remove the ribbon cable from the MiniCOP module's circuit boards. This may require the removal of the protective side cover in order to gain an unobstructed access to the connectors.
   - Prepare the rack's backplane by removing any jumpers, and update the resistor packs as necessary.

NOTE

Refer to *Appendix C* for detailed information on the augmented rack resistor pack.

6. Carefully insert the module into the rack.

7. Connect the external devices to the MiniCOP module's ASYN ports as determined by the application. If the MiniCOP has not been initialized, programmed, or loaded with any Personality Procedures/ADI Devices Drivers (if required) it will be necessary to first perform these steps utilizing the 623 Terminal Emulator software package.

NOTE

Refer to *Appendix B* for additional information.

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

MiniCOP ADDRESS ASSIGNMENTS

The following are a number of factors to keep in mind when assigning addresses to I/O slots, when installing the 627 MiniCOP module.

1. The MiniCOP utilizes PUSH/PULL ladder logic instructions to communicate with its host 620 Platform controller via the I/O bus. Therefore, the MiniCOP must be installed in I/O slots that accept modules using the PUSH/PULL interface. This includes any parallel or serial I/O rack and all processor rack I/O slots except the 620-25/35 LCS, the Series 9100 and 9200 Loop and Logic Controllers, and the Logic Manager (this list of unacceptable processor racks does **not** include the Series 9100E and 9200E Loop and Logic Controllers or the Advanced Logic Manager). The specified racks either do not except I/O modules or those that use the PUSH/PULL interface.

2. The 620 Platform controller system I/O modules are always addressed in the following order of priorities:

   - FIRST — Local I/O (I/O slots that reside in the processor rack).
   - SECOND — Parallel or serial I/O.

   Each rack containing I/O is assigned a starting address, and addressing then takes place from left to right in each rack (see Figure 18).

3. Overlapping I/O addresses and identical starting addresses within the same system are prohibited.

4. Skipping addresses or leaving I/O slots empty is permitted.

5. Each I/O slot must be configured for either 0-point, 8-point, 16-point, or 32-point operation.

6. Regarding the MiniCOP, only the CPU board communicates with the host processor via the I/O bus. This circuit board requires eight addresses, and the slot it occupies can be configured using the same rules as any 8-point module.

7. Eight-point modules can be inserted into slots configured for 16-point operation with no adverse effects. In this scenario, the 8-point module will assume the least significant eight addresses of the slot.

Since the MiniCOP is a two- or three-slot wide module and since only the CPU board (left-most) communicates via the I/O bus, only the CPU board slot must be addressed. However, it is also necessary to take into consideration the PORT/MEMORY board and/or the VIDEO board in terms of addresses assigned to the slots they occupy. The 620 Platform controllers provide four point sizes for each I/O slot: 0-point, 8-point, 16-point, and 32-point. Normally it is **NOT** advantageous to use the 0-point option. The MiniCOP is a module that can utilize the 0-point configuration effectively, and without any of the problems typically encountered with this selection.

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

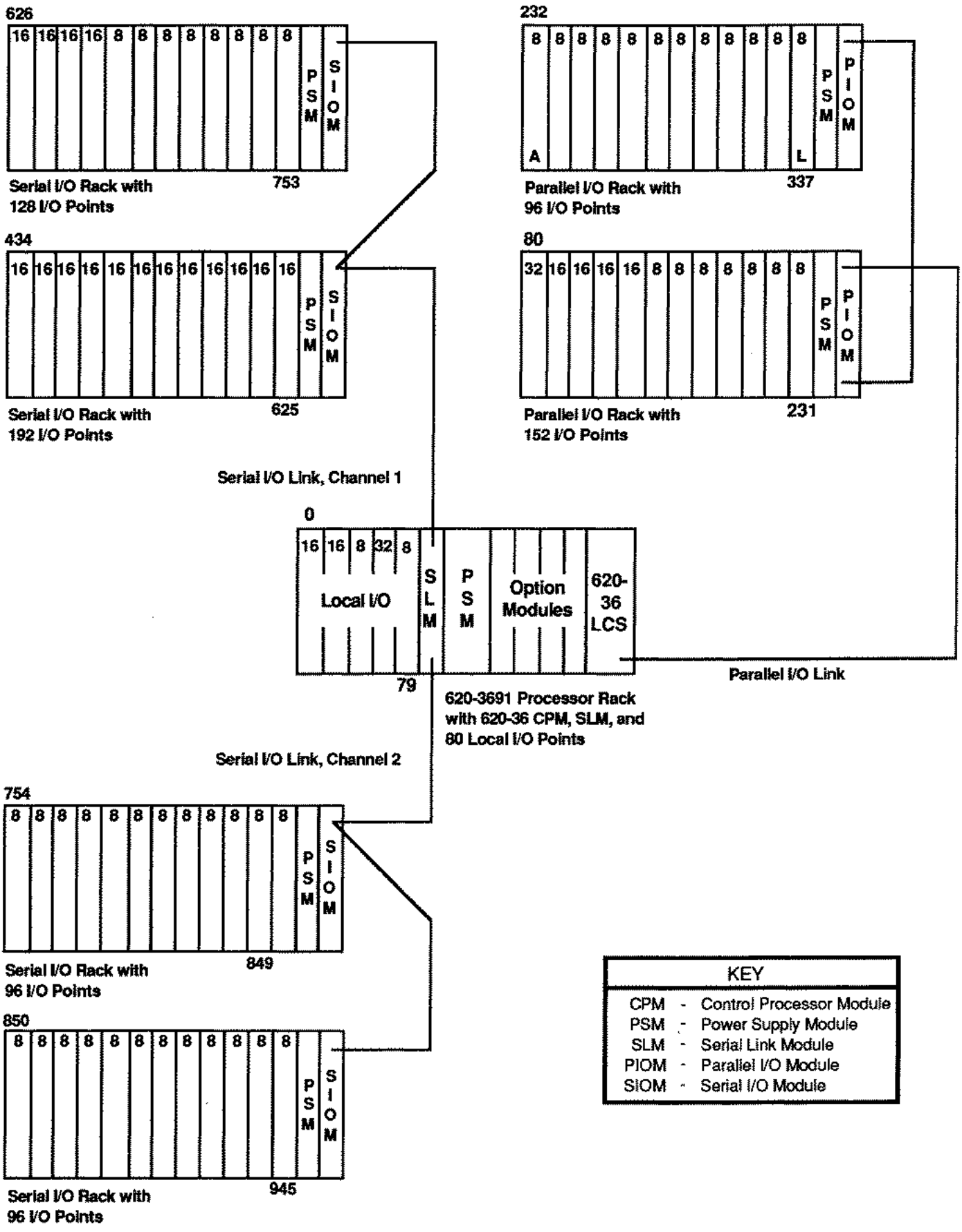


FIGURE 18 - I/O ADDRESSING EXAMPLE address assignments

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

Two MiniCOP I/O slot addressing examples are illustrated in Figures 19 and 20. Figure 19 illustrates a MiniCOP addressing example where both slots have been configured for 8-point operation. This will have no adverse effects on the MiniCOP, but it should be noted that the 8 addresses that have been assigned to the slot containing the PORT/MEMORY board are useless and are lost to the controller. The total number of possible 620 Platform controller real I/O points therefore has been reduced by eight.

Figure 20 illustrates a MiniCOP that has been installed in two I/O slots. The slot containing the CPU board is configured for 8-point operation, while the PORT/MEMORY board's slot is configured for 0-point operation.

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

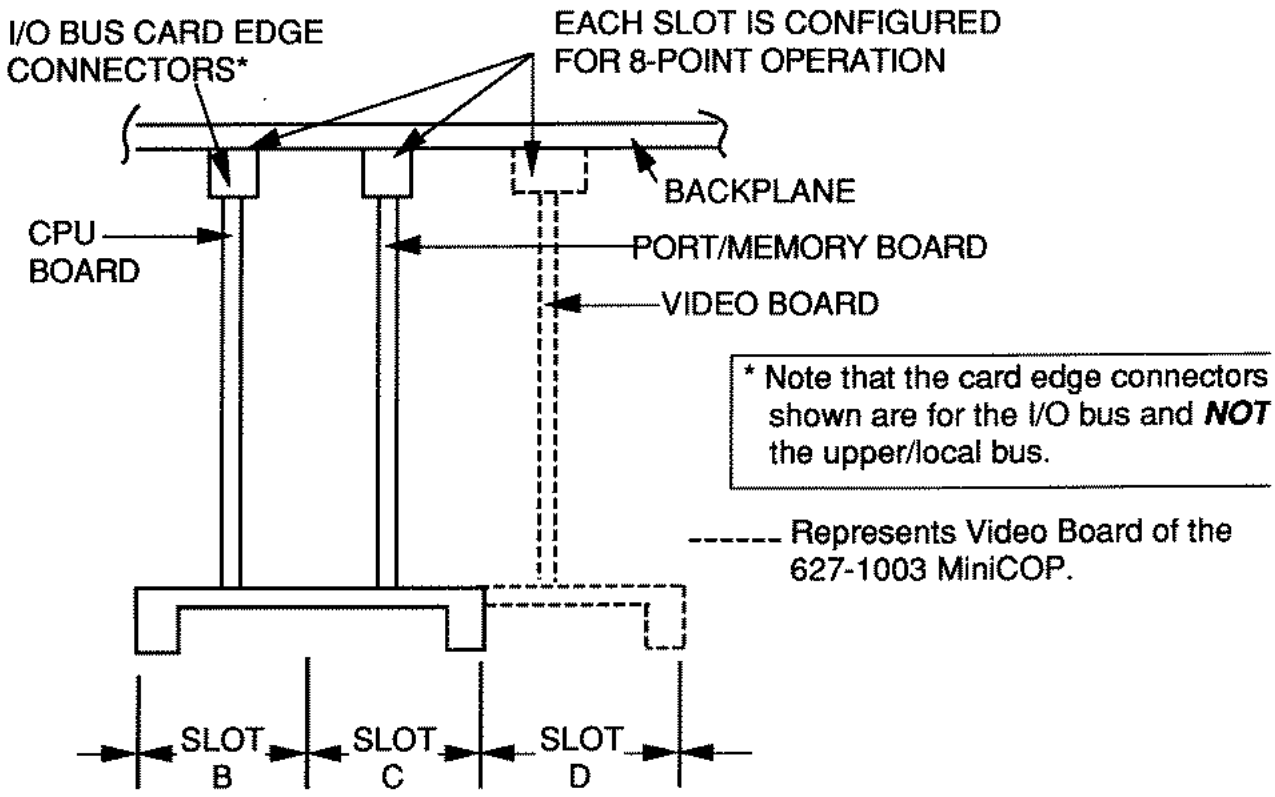


FIGURE 19 - MiniCOP SLOT ADDRESSING EXAMPLE #1 (TOP VIEW)

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

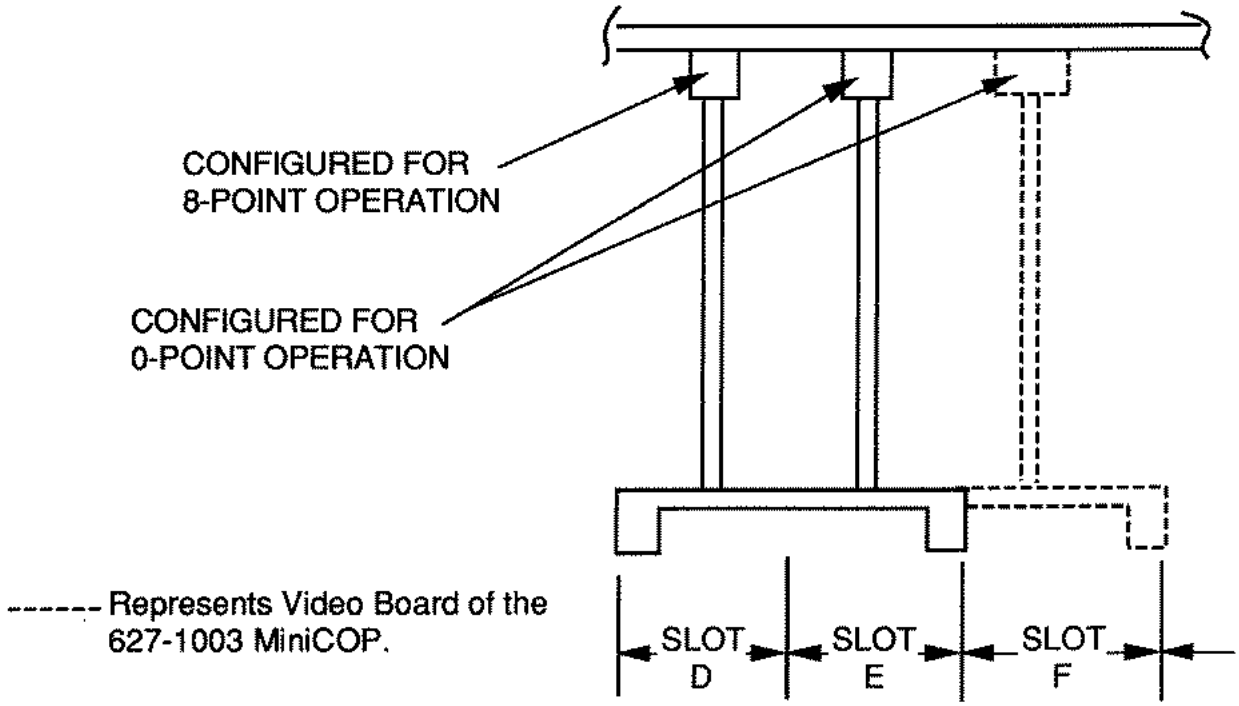


FIGURE 20 - MiniCOP SLOT ADDRESSING EXAMPLE #2 (TOP VIEW)

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

RACK TYPE CONSIDERATIONS

With few exceptions both 627 MiniCOP models can be installed in any standard or augmented rack; any processor or I/O, full- or half-size racks. Table 6 lists 627 MiniCOP-to-620 Platform rack compatibility.

TABLE 6 - MiniCOP RACK MODEL NUMBER COMPATIBILITY

Rack Model No.	Description	627-1002 MiniCOP
620-0090	06/10/15 Standard 19-inch Processor Rack	Yes
620-0091	06/10/15 Augmented 19-inch Processor Rack	Yes
620-0093	06/10/15 Standard 10-inch Processor Rack	Yes
620-1690	11/12/14/16/36 Standard 19-inch Processor Rack	Yes
620-1693	11/12/14/16/36 Augmented 19-inch Processor Rack	Yes
620-1695	11/12/14/16/36 Augmented 10-inch Processor Rack	Yes
620-2590	25 Standard 19-inch Processor Rack	No
620-3590	35 Standard 19-inch Processor Rack	No
620-3691	36 Augmented 19-inch Processor Rack	Yes
621-9990C	Standard 19-inch I/O Rack	Yes
621-9991C	Standard 10-inch I/O Rack	Yes
621-9992C	Augmented 19-inch I/O Rack (3 Segment)	Yes

The 627-1002 MiniCOP consists of two circuit boards and the 627-1003 MiniCOP consists of three. While only the CPU board can communicate via the I/O bus with the host 620 Platform controller, all MiniCOP circuit boards must communicate with each other. A means to establish this board-to-board communication is provided on each circuit board. Each circuit board also has two sets of card-edge connectors: one on the bottom for connection to the I/O bus, and one on the top for board-to-board communications. Though only the CPU board can communicate through the bottom card-edge, all circuit boards receive their operating power through this connection. In addition to the top card-edge connector, each also has a 50-pin connector mounted next to the card-edge connector on the component side of the circuit board. Either the card-edge or the 50-pin connectors are used to establish a path for board-to-board communication depending on the type of rack in which the MiniCOP is installed (that is, either an augmented or standard rack).

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

STANDARD RACKS

Both processor and I/O racks are available in the "Standard Rack" layout. Several standard half-size racks are also available.

A standard rack has a backplane that contains only one set of card-edge connectors. The 620 Platform processor and option module slots use a full-length card-edge connector that runs from the top to the bottom of the backplane. I/O slots use a shorter card-edge connector located at the bottom of the backplane. Figure 21 illustrates an example of a typical standard rack backplane (621-9990C Standard, Full-Size I/O Rack).

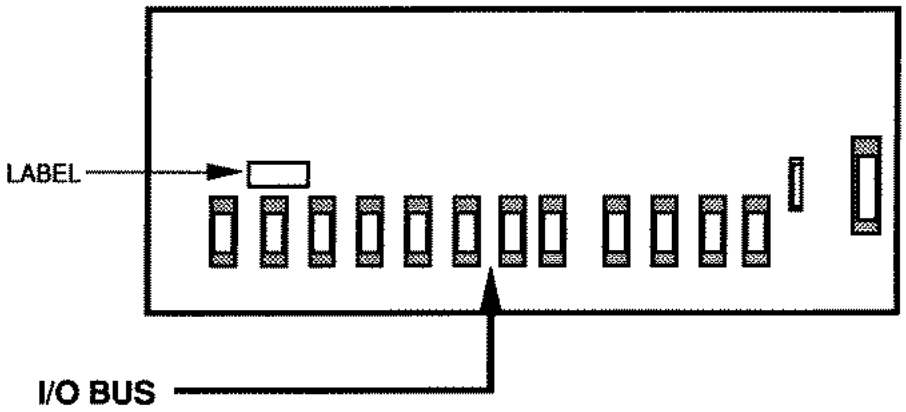
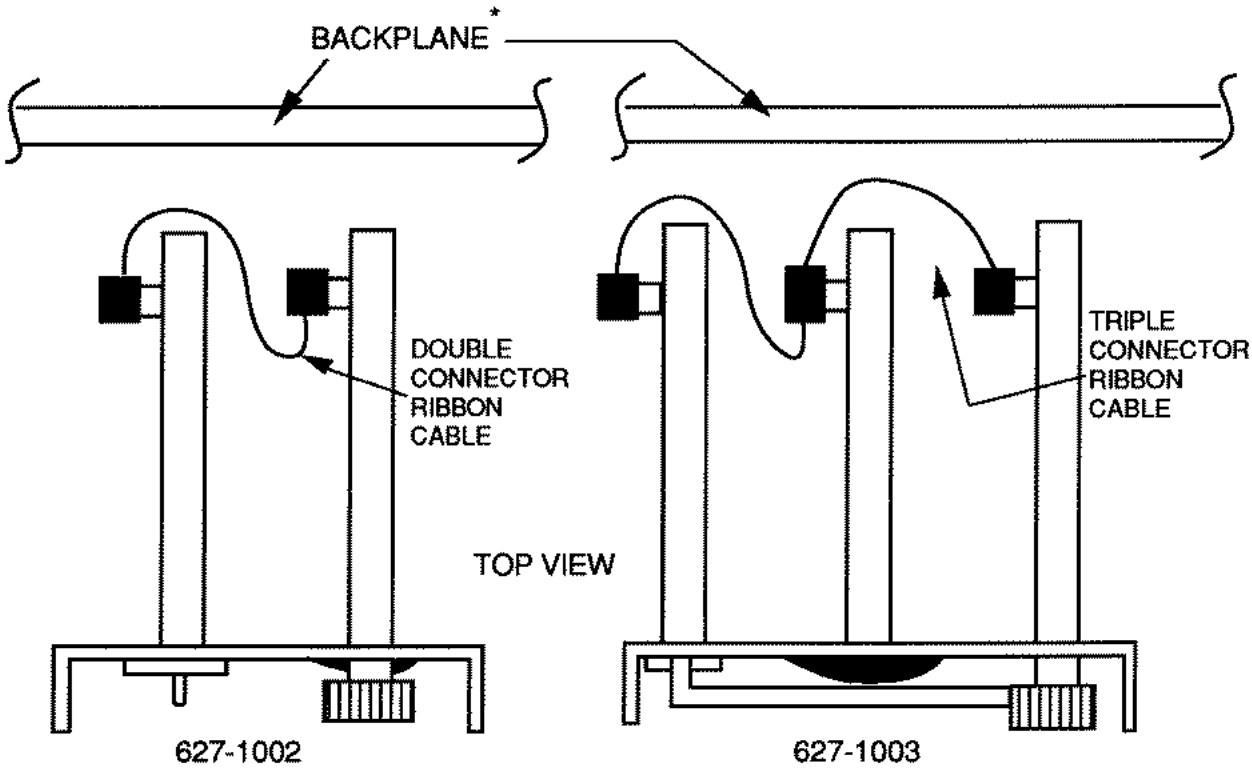


FIGURE 21 - 621-9990C I/O RACK BACKPLANE LAYOUT

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

Since each MiniCOP circuit board must also be able to intercommunicate, it is necessary that 50 conductor ribbon cable be installed. The ribbon cable is included with and factory-installed on each MiniCOP module for installation in a standard rack when shipped (see Figure 22). This ribbon cable provides a means for each MiniCOP circuit board to intercommunicate.



*Note the absence of upper/local bus card edge connectors.

FIGURE 22 - STANDARD (NONAUGMENTED) RACK CONNECTIONS

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

AUGMENTED RACKS

An augmented rack backplane contains two sets of card-edge connectors. An example of a typical augmented rack is illustrated in Figure 23 (621-9992C Augmented I/O Rack). In addition to the standard row of connectors, the augmented rack also has an upper row of card-edge connectors. The upper card-edge connectors (upper/local bus) provide the necessary pathway for the MiniCOP module's circuit boards to intercommunicate.

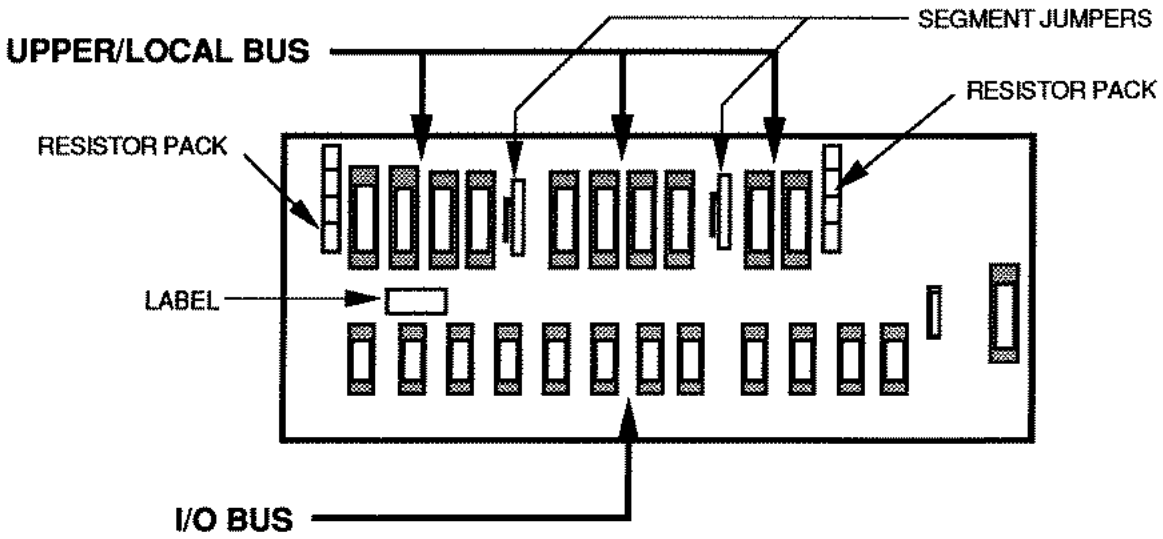


FIGURE 23 - 621-9992C AUGMENTED I/O RACK BACKPLANE LAYOUT

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

In order to install a MiniCOP module into an augmented rack, it is necessary to first remove the factory-installed ribbon cable. Attempting to insert a MiniCOP module into an augmented rack with the ribbon cable installed can damage the rack backplane, the card-edge connectors, and/or the MiniCOP module circuit boards. The installation of a MiniCOP module in an augmented rack is illustrated in Figure 24. Note that it may be necessary to remove the MiniCOP module's side cover in order to gain easy access to the ribbon cable for removal.

CAUTION

Caution must be exercised when removing a MiniCOP module from an augmented rack. Because the battery's electrical connection is made to the CPU board and not to the PORT/MEMORY board, memory will be lost if the MiniCOP is removed from an augmented rack (backup is provided to the PORT/MEMORY board via the upper/local bus). **BEFORE REMOVING** a MiniCOP from an augmented rack, always **SAVE MEMORY** to the floppy or hard disk via the Terminal Emulator. This is not a problem when the MiniCOP is installed in a standard rack since the battery backup is provided to the PORT/MEMORY board through the ribbon cable.

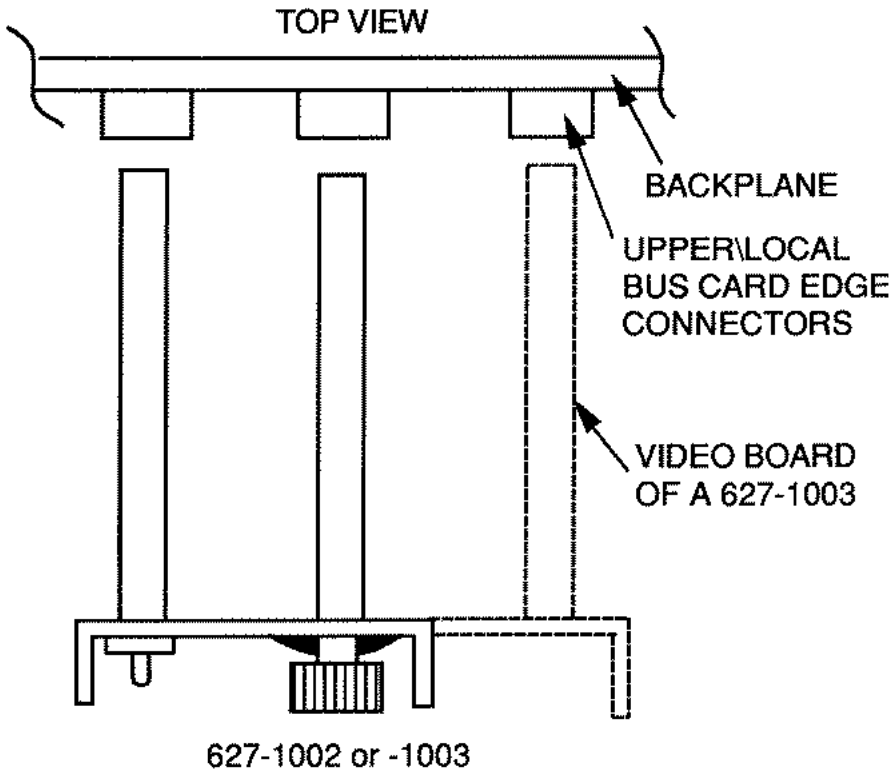


FIGURE 24 - AUGMENTED RACK/MINICOP CONNECTION

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

SPECIAL CONSIDERATIONS

Upper/Local Bus

Both the full- and half-size versions of the augmented racks contain resistor packs. The full-size augmented racks also contain backplane segment jumpers. It is important that these items are considered and appropriately configured so that the MiniCOP operates as expected. Examples of several augmented racks and the locations of the jumpers and/or resistor packs are illustrated in Figure 25.

The backplane segment jumpers allow the upper/local bus to be segmented into separate partitions. When a jumper is removed, pins (lines) 7 through 52 through that jumper are opened. This permits the creation of multiple augmented (upper/local bus) backplane segments which are used in the 622 Motion Control System. When the jumpers are installed all points along the upper/local bus are connected as one. In this instance all points can communicate to all other points along the upper/local bus. Also note that pins 1 through 6 and 53 through 56 are common across the upper/local bus and are not separated by the jumpers.

CAUTION

Multiple MiniCOPs cannot coexist in a single augmented I/O rack. Pin 5 (one of the common pins) is used as a clock signal by the MiniCOP. If multiple MiniCOPs are installed in a single augmented I/O rack, multiple clocks are attempting to share the same line. This will cause incorrect operation of the installed MiniCOP modules.

627-1002RC and 627-1003RC Hot Insertion/Removal MiniCOPs

When installing the 627-1002RC and 627-1003RC MiniCOP modules in augmented processor racks or I/O racks, certain considerations must be observed and noted.

As with the 627-1002 and 627-1003 MiniCOPs, caution must be exercised when removing either the 627-1002RC or 627-1003RC from an augmented rack. When a MiniCOP is installed in a standard rack, one function of the factory-installed ribbon cable is to provide battery backup to memory when power is removed from the unit or when the unit is removed from the rack. The ribbon cable must be removed for the MiniCOP to be installed in an augmented rack. In this installation, the upper/local bus provides a path for battery backup of memory should power be removed from the rack. If a 627-1002RC or 627-1003RC Hot Insertion/Removal MiniCOP is removed from an augmented rack, the link to backup power is broken and memory will be lost just as it is in the 627-1002 or 627-1003 MiniCOP modules.

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

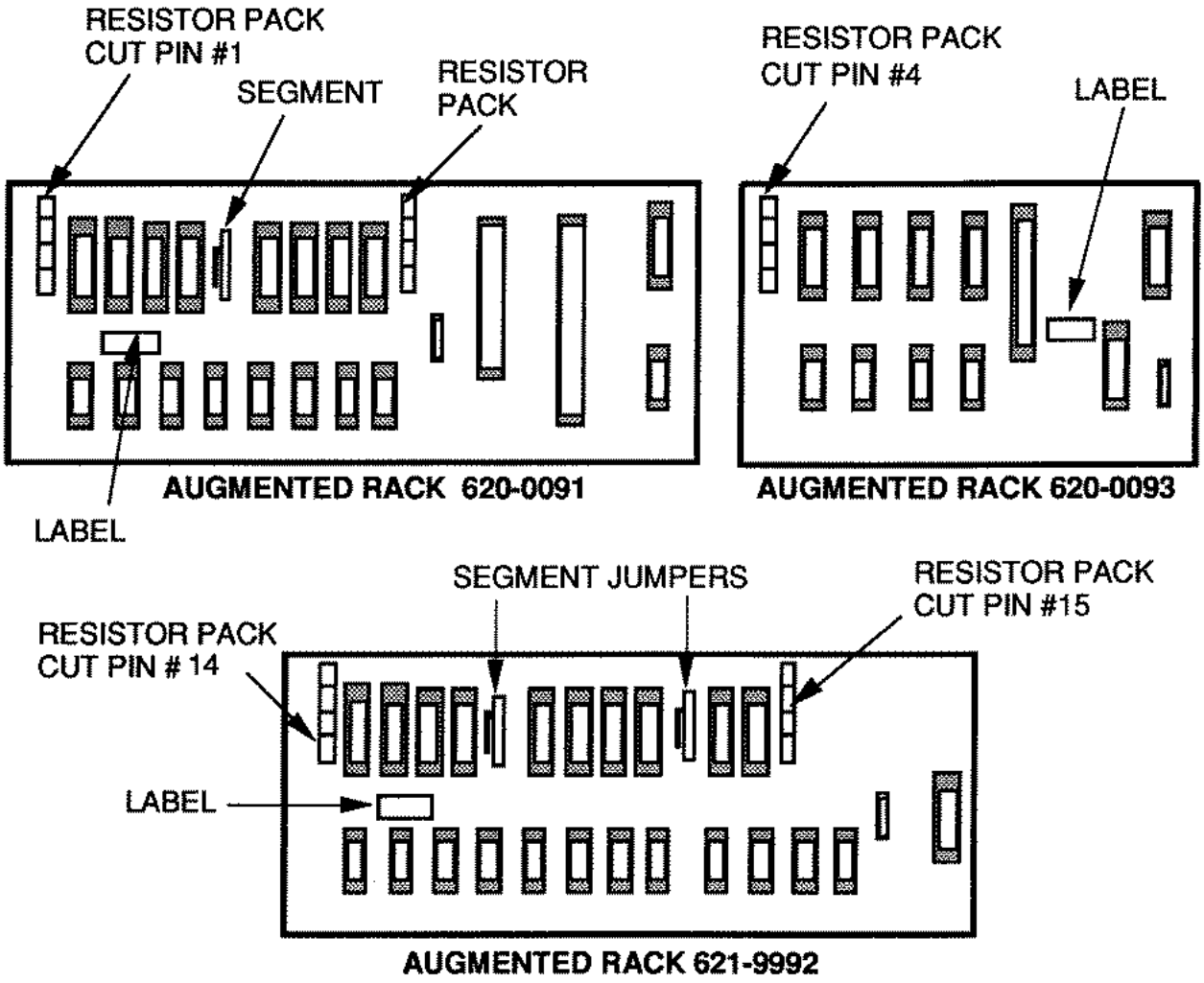


FIGURE 25 - BACKPLANES ON AUGMENTED RACKS

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

Resistor Pack Update

I/O modules that access the upper/local bus, such as the MiniCOP, can lose battery power when installed in an augmented rack.

If AC power is removed from the rack, the battery backup immediately takes over maintaining the MiniCOP's memory. Battery power is supplied to the PORT/MEMORY board through the ribbon cable connecting it to the CPU board where the battery's electrical connection to the MiniCOP is made. If the ribbon cable was removed for installation in an augmented rack, excessive current drain is placed on the battery through the backplane's resistor packs. This drain can significantly shorten the life of the battery's rated 9-month backup capability.

By updating the augmented rack's resistor packs, as shown in Figure 25, this excessive current drain can be eliminated. Additional details concerning the updating of resistor packs can be found in *Appendix C*.

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

CABLING CONSIDERATIONS

CE CONFORMITY SPECIAL CONDITIONS (EUROPE)

Shielded twisted pair cables are required for all Analog I/O, Process Variable, RTD, Thermocouple, dc millivolt, low level signal, 4-20 mA, and computer interface circuits. Additional special conditions for CE conformity of specific modules will be found in the WIRING section of the individual module manuals.

TO EXTERNAL CONTROL/SENSING DEVICES

After the MiniCOP module has been installed, proper cabling, spacing, shielding, and other typical concerns should be addressed. General wiring and cabling theory and practices are presented here for this purpose.

Cable and Conduit Routing

All wires and cables can be classified into two categories as follows:

High-Level Conductors: Carry high voltage and/or high current. Examples include AC I/O lines; high-level DC I/O lines connected to such devices as switches, relays, or solenoids; and AC power lines. These are typically IEEE 518 Data 2, 3, and 4 categories.

Low-Level Conductors: Carry low voltage and/or low current. Typical low-level conductors are serial communication cables; DC I/O lines connected to solid-state devices such as proximity switches or photoelectric sensors; TTL signal lines; analog signal lines; encoder signal lines; motion control signal lines; and parallel I/O communications cables.

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

Important Cable Considerations

Observe the following guidelines when routing cables and wiring. Note that special attention must be paid to separating data cables from power cables to avoid electrical noise problems.

- Maintain a 60°F (16°C) or less ambient air temperature at the bottom of each 620 Platform rack assembly.
- Install all wiring according to the *IEEE Standard 518-1977.*
- When using a wire duct less than 3.5 inches (89 mm) deep, allow at least 1.5 inches (38 mm) between it and the rack assembly. For a wire duct greater than 3.5 inches deep, allow a clearance of 2 inches (51 mm).
- Separate all 440VAC (or higher) electrical supply lines as far from 620 Platform components as possible. Tables 7, 8, and 9 list the **minimum** tray and conduit spacings.
- All 440VAC (or higher) lines must be run in separate conduits from any I/O wiring ducts.
- Separate all AC and low-level DC wiring. Tables 7, 8, and 9 list the **minimum** tray and conduit spacings.
- When 8-inch (203 mm) deep enclosures are used, door stiffeners are not permitted in front of the 620 Platform racks.
- Separate the data cable from AC and DC (above 12V) when connecting cabling between the 623-6100/6150 Loader/Terminal and the 620 Platform controller.
- All data cables should be terminated at both ends, excluding the shield. If a termination is not made or is intermittent, noise immunity is lost even though the system may transmit valid data.
- Shields for communications cables should be terminated at both ends. Shields for analog signals should be terminated to ground at the source end.
- Data and power cables should cross at right angles rather than running parallel for any distance.
- Ensure that the entire communications network is grounded to earth ground.
- Shield continuity must be maintained. Ensure that the shield leads are not broken.
- Cables should be routed around rather than through high-noise areas.
- Route shielded data cable along surfaces such as metal cabinet walls or in conduits or trays. Single-shield cable is most effective when routed along grounded surfaces.
- Allow the minimum amount of unshielded wire that will accommodate the connection.

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

TABLE 7 - TRAY TO TRAY SPACING — INCHES (MM)

Cable Class	Data 1	Data 2	Data 3	Data 3A	Data 4	Data 4A
Data 1		Note 2	6 (152)	6 (152)	26 (660)	26 (660)
Data 2	Note 2		6 (152)	6 (152)	18	26 (660)
Data 3	6 (152)	6 (152)		0	Note 1	12 (305)
Data 3A	6 (152)	6 (152)	0		8 (203)	18 (457)
Data 4	26 (660)	18 (457)	Note 1	8 (203)		0
Data 4A	26 (660)	26 (660)	12 (305)	18 (457)	0	

NOTE 1: Data 3 and 4 cables may be run in a common tray but should be separated by a barrier. This barrier does not have to be grounded.

NOTE 2: Data 1 and 2 cables may be run in a common tray, provided they are separated by a grounded steel barrier.

TABLE 8 - TRAY TO CONDUIT SPACING — INCHES (MM)

Cable Class	Data 1	Data 2	Data 3	Data 3A	Data 4	Data 4A
Data 1		1 (25)	4 (102)	4 (102)	18 (457)	18 (457)
Data 2	1 (25)		4 (102)	4 (102)	12 (305)	18 (457)
Data 3	4 (102)	4 (102)		0	0	8 (203)
Data 3A	4 (102)	4 (102)	0		6 (152)	12 (305)
Data 4	18 (457)	12 (305)	0	6 (152)		0
Data 4A	18 (457)	18 (457)	8 (203)	12 (305)	0	

TABLE 9 - CONDUIT TO CONDUIT SPACING — INCHES (MM)

Cable Class	Data 1	Data 2	Data 3	Data 3A	Data 4	Data 4A
Data 1		1 (25)	3 (76)	3 (76)	12 (305)	12 (305)
Data 2	1 (25)		3 (76)	3 (76)	9 (229)	12 (305)
Data 3	3 (76)	3 (76)		0	0	6 (152)
Data 3A	3 (76)	3 (76)	0		6 (152)	9 (229)
Data 4	12 (305)	9 (229)	0	6 (152)		0
Data 4A	12 (305)	12	6 (152)	9 (229)	0	

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

Cable Spacing

It is recommended that noise-creating sources (those with time-varying voltage or time-varying current) be separated from data signal cables. Cables with similar levels of noise susceptibility should be grouped together, and those with similar levels of noise generation should be grouped in trays or conduits.

IEEE 518 defines four classes of wiring that differ in signal level and noise susceptibility as follows:

DATA 1: High-noise susceptibility. Analog signals of less than 50V and digital signals of less than 15V.

DATA 2: Medium-noise susceptibility. Analog signals of less than 50V, regulating signals of 50V with currents less than 20A; AC feeders less than 20A.

DATA 3: Low-noise susceptibility. Switching signals greater than 50V, regulating signals of 50V with currents less than 20A; AC feeders less than 20A.

DATA 4: ***Power*** — AC and DC buses of 0-1000V with currents of 20-800A.

Data 3 and 4 classes have one subclass each for applications that require special cable and spacing. Tables 7, 8, and 9 list the subclass for Data 3 cables as Data 3A and the subclass for Data 4 cables as Data 4A. Examples of Data 3 and Data 4A cables are as follows.

1. Signals from communicating fields and line resistors.
2. Signals from shunts to regulators.
3. Power greater than 1000V and/or 800A.

NOTE
Refer to *IEEE 518* for complete information on wiring classes and subclasses.

Tray Spacing

Tables 7, 8 and 9 list the recommended minimum distances between trays, trays and conduits, and conduits respectively. The dimensions provided are between the top of one tray/conduit and the bottom of the tray/conduit directly above, or between the sides of any adjacent trays/conduits.

Trays containing Data 1 or 2 cables should have solid bottoms and covers to complete the shielding. Ventilation slots or covers with louvers may be used only in trays containing Data 3 cables.

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

Conduit Shielding

Conduit or cable trays should be used to route the wiring within the plant for short distances or for several miles. The same conduit should not carry both data and power cables. This is because of the problems created by electromagnetic interference. The choice of materials and configuration for the cable can determine the degree of protection from this interference.

Table 10, listing raceway shielding, is taken from the *IEEE Standard 518* for the installation of electrical equipment. It compares different types of raceways for magnetic field attenuation at 60Hz and for electrostatic attenuation at 100 KHz.

NOTE
Appendix B provides wiring diagrams for establishing cabling for the MiniCOP's serial ports.

TABLE 10 - RACEWAY SHIELDING

Raceway Type	Thickness in. (mm)	60 Hz Mag. Field Attenuation		100 KHz Elec. Field Attenuation	
		Ratio	dB	Ratio	dB
Free Air		1:1	0	1:1	0
2-inch Aluminum Conduit	0.154 (3.91)	1.5 : 1	3.3	2,1650: 1	66.5
#16 Gauge Aluminum Tray	0.060 (1.52)	1.6:1	4.1	15,500: 1	83.9
#16 Gauge Aluminum Tray	0.060 (1.52)	3:1	9.4	20,000: 1	86.0
#16 Gauge Galv. Ingot Iron Tray	0.060 (1.52)	3.2:1	10.0	22,000: 1	86.8
2-inch IPS Copper Pipe	0.156 (3.96)	3.3:1	10.2	10,750: 1	80.6
#16 Gauge Aluminum Tray	0.060 (1.52)	4.2:1	11.5	29,000: 1	89.3
#14 Gauge Aluminum Tray	0.075 (1.90)	6:1	15.5	23,750: 1	87.5
2-inch Metallic Tubing	0.065 (1.65)	6.7:1	16.5	3,350:1	70.5
2-inch Rigid Galv. Conduit	0.154 (3.91)	40:1	32.0	8,850:1	78.9

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

TO THE TERMINAL EMULATOR

Before the MiniCOP can be applied to any control and/or monitoring operation, it must be initialized (OS-9 commands loaded) and programmed. The Terminal Emulator is a software package that runs on a DOS-operating system. The Terminal Emulator allows you to interface with the MiniCOP, which is an OS-9-operating system microcomputer. The Terminal Emulator uses the personal computer's serial ports to communicate to the MiniCOP module's Ports 1 and 2.

NOTE

Refer to the *Terminal Emulator* section of this manual for detailed information regarding its use, operation, and characteristics.

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

VIDEO FUNCTIONS

OVERVIEW

The triple-slot-wide 627-1003 MiniCOP provides a dedicated video port that can be used to send monochrome or color RGB video signals to a remote monitor. Four female BNC-type connectors (video ports) are located on the module's frontplate for this purpose, as shown in Figure 26. The RGB video signal is provided in the RS-170 electrical standard.

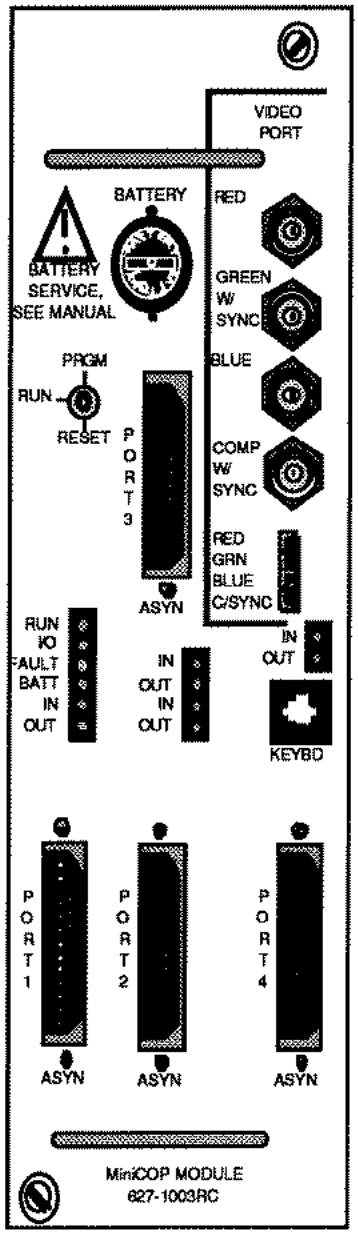


FIGURE 26 - 627-1003RC MINICOP

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

The bottom BNC connector provides a composite monochrome signal, which includes both vertical and horizontal synchronization. The top three BNC connectors provide a color signal. Vertical and horizontal synchronization are combined with the GREEN signal when using a color monitor.

Monitors can be connected to either the monochrome or color ports via RG58 coaxial cable up to a distance of 1000 feet (300 meters).

NOTE
Refer to the *Video Cables* portion of *Appendix B* for information on constructing video cables.

Four potentiometers are available to adjust the video signal levels. These potentiometers can be accessed from the frontplate and are located just beneath the video port label as shown in Figure 26.

VIDEO SIGNAL ADJUSTMENT

The video signal is adjusted while the monitor is cabled to the MiniCOP's video ports and both devices are powered. There are three types of monitors that can be used with the MiniCOP's video ports.

- COLOR MONITOR with SEPARATE Sync Control,
- COLOR MONITOR with COMBINED Sync/Green Control, or
- MONOCHROME MONITOR.

To adjust the video signal, select the appropriate procedure (of the following) that corresponds to the type of monitor being used. These procedures require that the adjustments be made to suit your specific requirements.

COLOR MONITOR WITH SEPARATE SYNC CONTROL

1. Set the monitor's brightness control to the half-way position.
2. Adjust the vertical and horizontal hold using the MiniCOP's C/SYNC potentiometer.
3. Adjust the RED, GREEN, and BLUE signals using the appropriate potentiometers on the MiniCOP module.

COLOR MONITOR WITH COMBINED SYNC CONTROL

1. Set the monitor's brightness control to the half-way position.
2. Adjust the vertical and horizontal hold using the MiniCOP's GREEN (with combined sync) potentiometer.
3. Continue turning the GREEN potentiometer to adjust the GREEN video.
4. Adjust the RED and BLUE signals using the appropriate potentiometers on the MiniCOP module.

MONOCHROME MONITOR

1. Set the monitor's brightness control to the half-way position.
2. Adjust the vertical and horizontal hold using the MiniCOP's C/SYNC potentiometer.

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

VIDEO COMMAND CODES

The 627-1003 MiniCOP's video driver accepts and responds to ASCII codes that have been defined from statements in a user-developed BASIC09 procedure. There are four primary video command codes: ESC, DEL, CR, and LF. Only the ESC command code uses parameters to further define its operation. These command codes are categorized as READ and WRITE operators. The function and effect of the video command codes on an RGB-compatible monitor are presented in Table 11 below.

TABLE 11 - VIDEO COMMAND CODES

Read	Command	Response	Function
	ESC (R)	<x,y>	Read visible cursor position
	ESC (r)	<x,y>	Read blind cursor position
	ESC (N)	<char>, <attr>, <color>	Set READ Mode
	ESC (n)		Reset READ Mode
Write	**Command**		**Function**
	DEL		Delete last character
	CR		Carriage return, set cursor to beginning of next line
	LF		Line feed, move cursor down one line
	ESC (c)		Reset the Video Controller
	ESC(J)		Clear screen
	ESC (W) <x1, y1 ; x2,y2>		Fill screen
	ESC (j) <char>, <attr>, <color>, <x1, y1; x2, y2>		Fill window
	ESC (U)		Delete line at current X, Y
	ESC (V)		Delete line at current X, Y
	ESC(q) <attr> , <color>		Set <attr> and <color> at current cursor position
	ESC (X) <x1,y1,; x2,y2 >		Set window reverse
	ESC (P)<x,y.>		Set visible cursor
	ESC (p) <x, y>, <attr>, <color>		Set visible cursor, <attr> latch and <color> latch
	ESC (8)		Set visible cursor auto move – ON
	ESC (9)		Set visible cursor auto move – OFF
	ESC (A)		Move visible cursor – UP
	ESC (B)		Move visible cursor – DOWN
	ESC (C)		Move visible cursor – RIGHT
	ESC (D)		Move visible cursor – LEFT
	ESC (0)		Set visible cursor – OFF
	ESC (1)		Set visible cursor – ON
	ESC (2)		Set visible cursor – UNDERLINE
	ESC (3)		Set visible cursor – BLINKING
	ESC (4)		Set visible cursor – NON-BLINKING
	ESC (5)		Set visible cursor – BLOCK
	ESC (S) <x,y>		Set blind cursor
	ESC (s) <x,y>, <attr>, <color>		Set visible cursor, <attr> latch and <color> latch

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

SYNTAX

The following is an example of a video command code's syntax with parameters:

ESC (command) <x, y>, <char>, <attr>, <color>

The parameters of a video command code are defined as:

ESC	Calls and executes the video command code as identified in the attached parameters. Represented in a BASIC09 procedure by the CHR$(27) statement, where 27 is the decimal equivalent of the ASCII code escape key.
(command)	A single alphanumeric character that identifies a specific video command code function.

NOTE
Upper- and lower-case representation of the same alphabetic characters have different functions.

<x, y>	Identifies a specific screen coordinate by its vertical and horizontal position. X-position is a value between $80 and $CF (0-79).*. Y-position is a value between $80 and $97 (0-23) or $80 and $AF (0-47).*1.
<x1, y1;x2, y2>	Identifies a specific block of screen coordinates by its upper left (x1, y1) and bottom right (x2, y2) corners. X-position is a value between $80 and $CF (0-79).*. Y-position is a value between $80 and $97 (0-23) or $80 and $AF (0-47).*2.
<char>	Identifies a specific ASCII code alphanumeric character between $20 and $FF (32-255).
<attr>	Identifies a specific ASCII code alphanumeric character between $80 and $FF (128-255)
<color>	Identifies a specific ASCII code alphanumeric character between $80 and $FF (128-255)

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

VIDEO COLOR REGISTER

Table 12 provides a listing of the content and function of the 8-bit Video Color Register. With this register it is possible to control the color of both the background and foreground. A special function (blinking) is also controlled by the Video Color Register.

TABLE 12 - VIDEO COLOR REGISTER

Function	Bit Position	Color	State	Decimal Weight
Background	0	Red	Off = 0 On =1	1
	1	Green	Off = 0 On =1	2
	2	Blue	Off = 0 On =1	4
Foreground	3	Red	Off = 0 On =1	8
	4	Green	Off = 0 On =1	16
	5	Blue	Off = 0 On =1	32
Special Function	6	Blinking	Off = 0 On =1	64
	7	Not Used	N/A	128

VIDEO ATTRIBUTE REGISTER

Table 13 provides a listing of the content and function of the 5-bit Video Attribute Register. With this register it is possible to control the video Attributes and Character Sets.

TABLE 13 - VIDEO ATTRIBUTE REGISTER

Function	Bit Position	Description	State	Decimal Weight
Attribute	0	Reverse	Off = 0 On = 1	1
	1	Underline	Off = 0 On = 1	2
	2	Line Through	Off = 0 On = 1	4
Character Set	3 4	External	On = 1 Off = 0	8 16
	3 4	Alphanumeric	On = 1 On = 1	8 16

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

BASIC09 EXAMPLE

The following example lines of source code demonstrate how the video command codes might appear in a BASIC09 procedure.

```
DIM A$:STRING[7]
A$=CHR$(27)+"j"+CHR$(27)+"P"+CHR$(8)+CHR$(20)+"PROCESS DATA"
OPEN #PATH"/V4":WRITE
PUT #PATH, A$
CLOSE #PATH
```

This procedure will clear the screen and display the word PROCESS DATA beginning at cursor position 8, 20.

Line one creates a one-dimensional array (vector), called A$, as a string array with seven elements.

Line two assigns data to the seven elements in the A$ array.

In **line three**, a channel is opened to the MiniCOP video ports via the V4 device driver.

Line four writes an ASCII formatted data record to the path defined by the OPEN statement.

Finally, in **line five**, the path to the video port is closed.

NOTE
Refer to *Appendix D* for a detailed listing of the ASCII code character set.

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

Terminal Emulator

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

Section 2.
Terminal Emulator

About This Section

This section presents background information on the use and operation of the Terminal Emulator software package. This software package is included with the MS-DOS Loader software package (Model Nos. 623-6010 and 623-6020); is also included with the 623 Loader/Terminal (623-6100 and 623-6150); and.can be ordered separately (Model No. 626-6041)

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

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

INTRODUCTION

TERMINAL EMULATOR OVERVIEW

NOTE
Refer to the *623-6041 Terminal Emulator User Manual* for additional information.

The MS-DOS Terminal Emulator is a software package that essentially turns your personal computer into a "dumb terminal" (when communicating with the MiniCOP, for example). Using the Terminal Emulator you can converse with two separate and unique microprocessors through your PC's keyboard and monitor. One of these microprocessors is the MiniCOP running the OS-9 operating system and the other your personal computer running a DOS operating system.

Figure 27 illustrates the relationship between the personal computer and the MiniCOP. The Terminal Emulator software package acts as a bridge between the MiniCOP and the personal computer. Function keys F1 through F10 allow you to communicate with the MiniCOP. Commands entered at any of the MiniCOP's four prompts (OS9:, B:, E:, and D:) allow you to communicate to the MiniCOP.

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

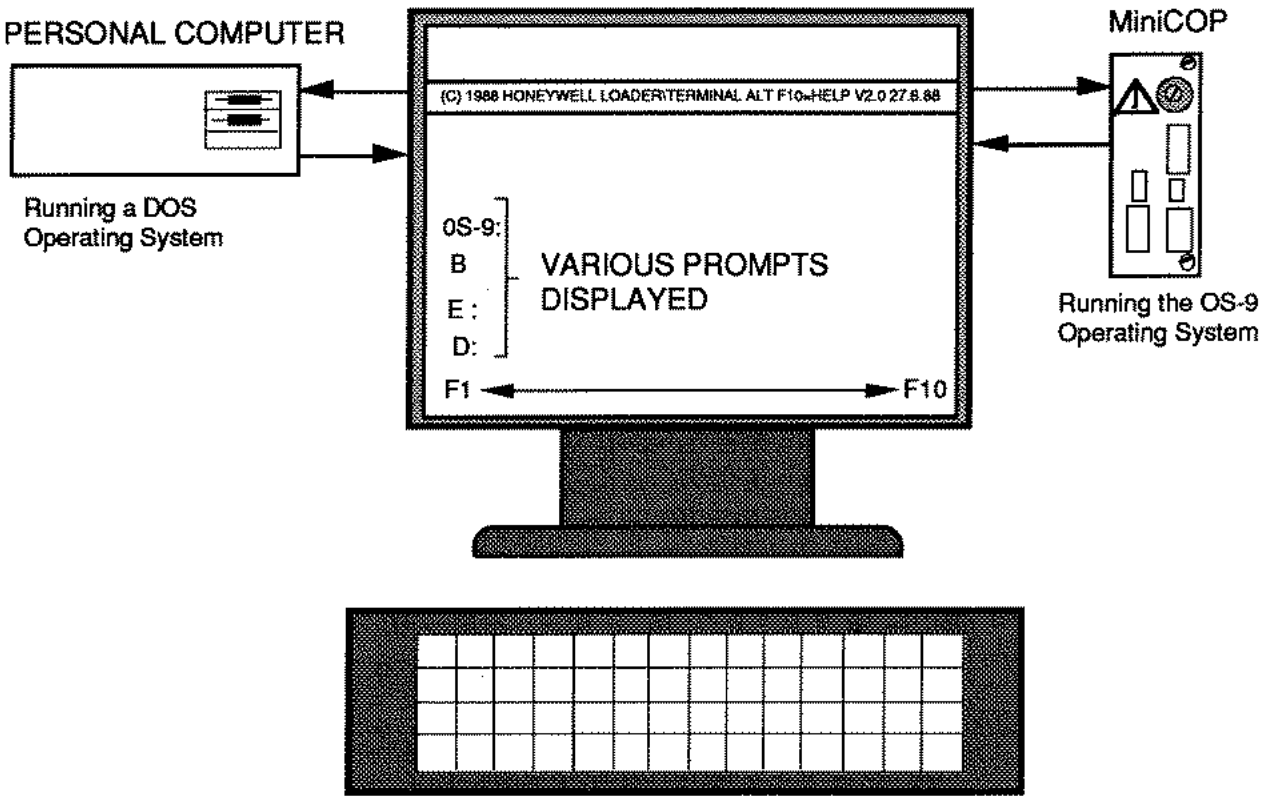


FIGURE 27 - TERMINAL EMULATOR "BRIDGE" BETWEEN THE PERSONAL COMPUTER AND MINICOP

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

SHIPPING CONTENTS

The Terminal Emulator consists of the following items:

- One 5.25-inch double-side, double-density floppy disk labeled Terminal Emulator.
- One 3.5-inch double-side, double-density floppy disk labeled Terminal Emulator.
- One *623 Terminal Emulator User Manual*.

Both diskettes contain the following files:

TERMMESS.DAT
TERMHELP.IND
TERMHELP.DAT
TERMMESS.IND
TERMVARS.C@L

MINIMUM SYSTEM REQUIREMENTS

The Terminal Emulator software is designed to operate on any MS-DOS based personal computer. The following is a list of the **minimum** system requirements.

- MS-DOS software and hardware compatible.
- Monochrome or color monitor, **CGA Graphics Adapter required**.
- One 5.25-inch or 3.5-inch floppy disk drive.
- Hard disk drive (recommended).
- Minimum 384K RAM main memory.

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

CABLING TO THE MiniCOP

The Terminal Emulator software package allows communications to the MiniCOP from the PC's serial communication port(s). The PC will generally have either one or two serial communications ports. It should be noted that although most PCs come standard with just one serial communications port, cable connections between the MiniCOP and a PC with a single serial communications port are generally more cumbersome to make than for PCs that have two serial communications ports.

Procedures are presented in this subsection for making cable connections between the MiniCOP and the following PC configurations:

- PCs with one serial communications port,
- PCs with two serial communications ports, and
- PCs with one serial communications port and a Loader/Terminal port.

NOTE
Refer to *Appendix B* for specific cable construction details.

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

ONE PC SERIAL PORT CONNECTED TO MiniCOP

Figure 28 illustrates the cable connections between a PC with one serial communications port and a MiniCOP module. It should be noted that while this method will work, it requires the Terminal Emulator to communicate to the MiniCOP module's Ports 1 and 2 via one cable. In this arrangement you will be required to move the cable between the two MiniCOP ports in accordance with the ongoing communications needs of the Terminal Emulator.

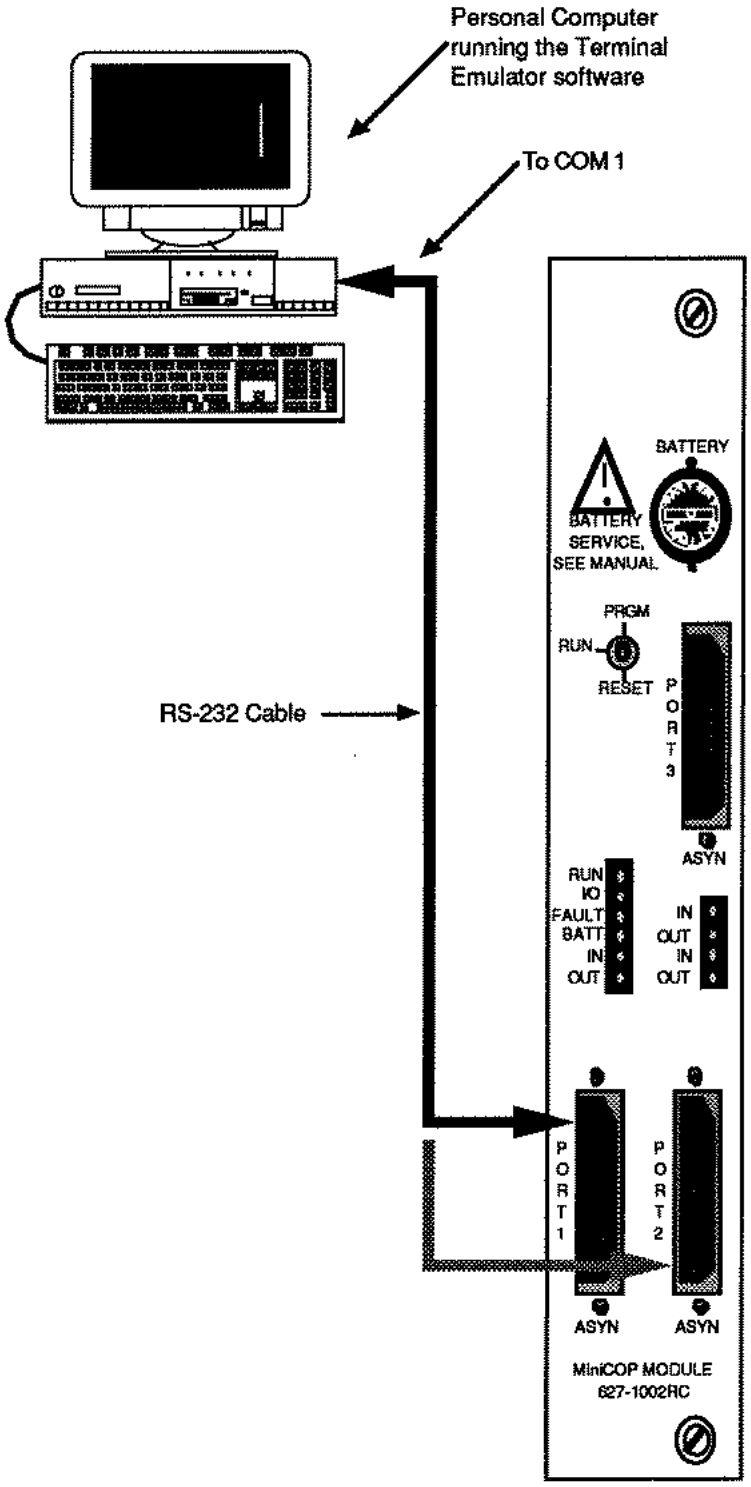


FIGURE 28 - ONE PC SERIAL PORT CONNECTED TO MiniCOP

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

TWO PC SERIAL PORTS CONNECTED TO MiniCOP

Figure 29 illustrates the cable connections between a PC with two serial ports and a MiniCOP module. Since both ports on the MiniCOP have their own connections in this configuration, it is more practical in terms of ease of use than the single PC serial port configuration.

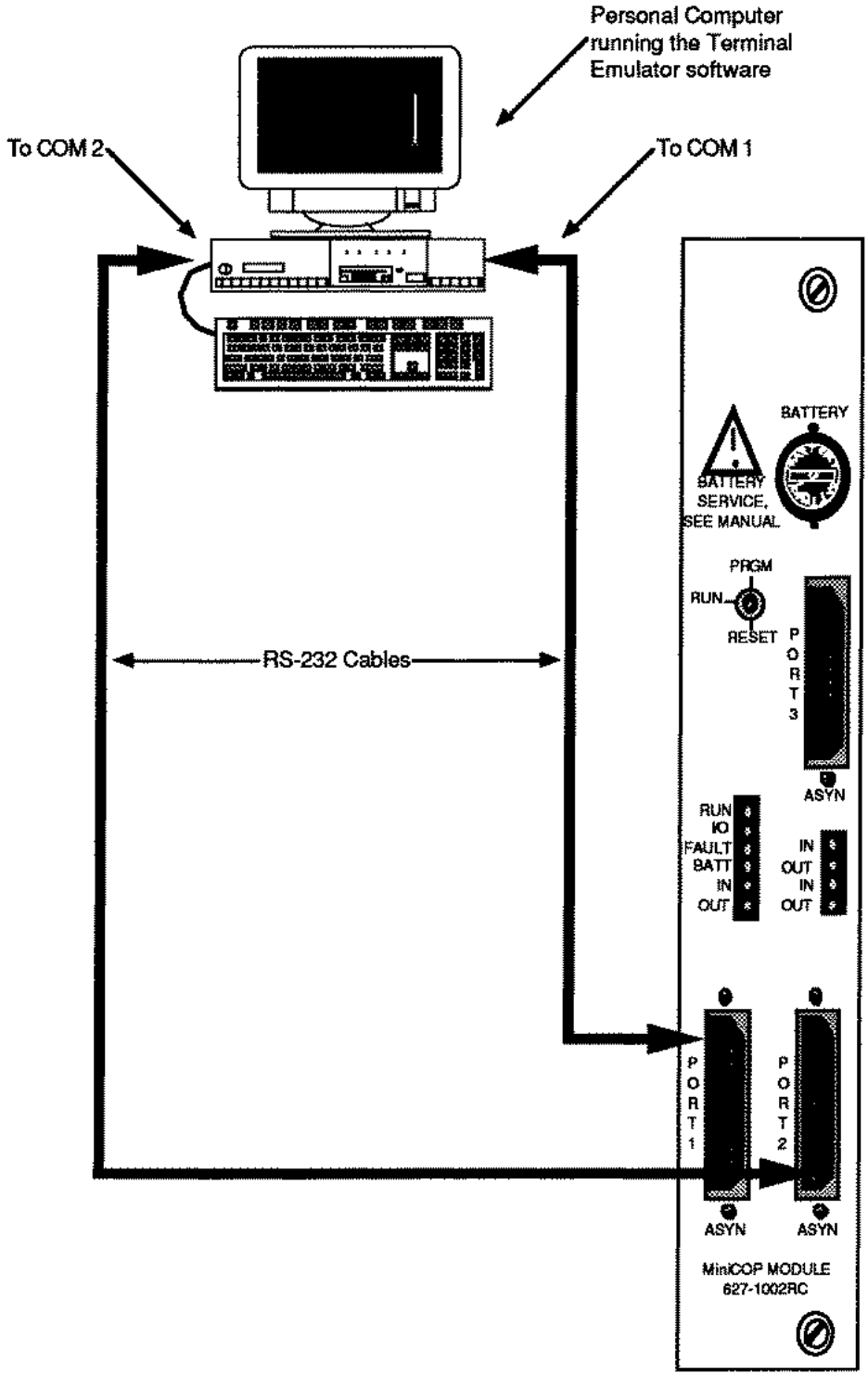


FIGURE 29 - TWO PC SERIAL PORTS CONNECTED TO MiniCOP

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

PC SERIAL PORT AND LOADER/TERMINAL PORT CONNECTED TO MiniCOP

PCs that contain an internal, half-size MS-DOS Loader Interface board (which is used to communicate to 620 Platform controllers) can use the board as a second serial communications port (referred to as the Loader/Terminal or L/T port). This board is an RS422/485 serial interface card that can be used to communicate to virtually any compatible device. Since any MiniCOP port can be set up to operate on the RS422/485 format, the one PC serial port dilemma is solved and the purchase of a second serial card for the PC can be avoided. Figure 30 illustrates the cable connections for this L/T port configuration.

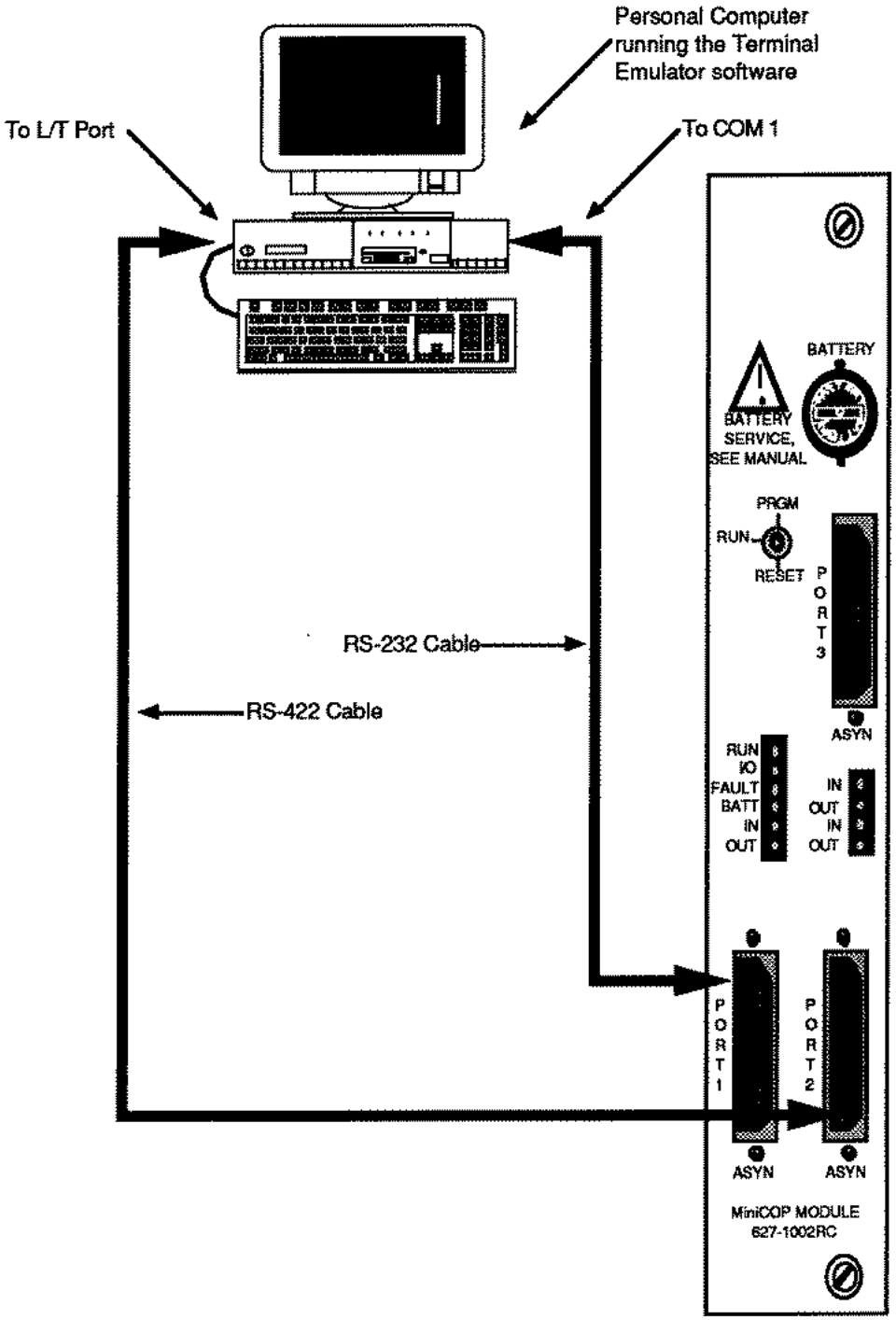


FIGURE 30 - PC SERIAL PORT AND LOADER/TERMINAL PORT CONNECTED TO MiniCOP

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

TERMINAL EMULATOR START-UP

FROM FLOPPY DISK

1. Insert the Terminal Emulator disk in the appropriate drive (available on 3.5-inch and 5.25-inch floppy disks).

2. Change the current drive selection to the one containing the Terminal Emulator disk (i.e. A:, B:, etc.).

3. Enter the filename **TERMINAL** at the DOS prompt.

4. Proprietary information and copyright screen appear.

FROM HARD DISK

1. If necessary change to the drive that contains the directory with the Terminal Emulator files (i.e. C:, D:, etc.).

2. Change to the directory that contains the Terminal Emulator files.

3. Enter the filename **TERMINAL** at the DOS prompt.

4. Proprietary information and copyright screen appear.

FROM THE 623-6100 or 623-6150 LOADER/TERMINAL

1. The 623-6100 and 623-6150 Loader Terminal automatically enter the MS-DOS Loader at power-up. Use the <F7> function key from the Main menu to access the Terminal Emulator.

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

CONFIGURATION

TERMINAL EMULATOR MAIN MENU

Figure 31 illustrates the Main menu display of the Terminal Emulator. Each of the Main menu selections can be entered by simply pressing the corresponding function keys F1 through F10. Help screens are available for the Main menu and all subsequent menus and/or function windows by pressing <ALT> <F10>.

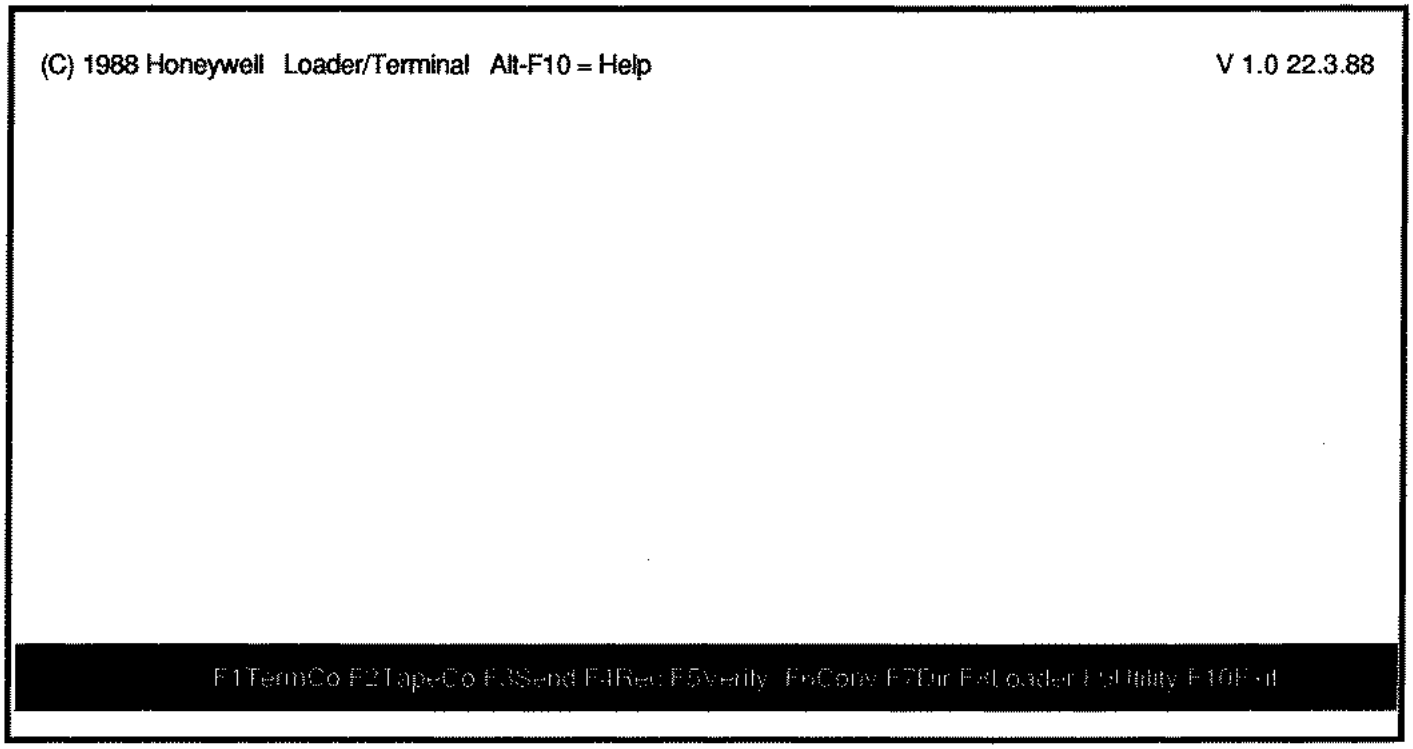


FIGURE 31 - TERMINAL EMULATOR MAIN MENU DISPLAY

MAIN MENU SELECTIONS

F1 = Configuration of Terminal Port
F2 = Configuration of Tape Port
F3 = Send Tape to Port
F4 = Receive from Tape Port
F5 = Verify on the Tape Port
F6 = NOT USED
F7 = Display Directory
F8 = Start LOADER
F9 = Start UTILITY
F10 = Exit to DOS
ALT-F10 = Help Screens

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

NOTES

The term **TERMINAL PORT** identifies the MiniCOP's port 1, which is the port used by the Terminal Emulator to issue commands to the MiniCOP.

The term **TAPE PORT** identifies the MiniCOP's port 2, which is the port used to transfer files between the Terminal Emulator and the MiniCOP.

The Main menu selection <F6> was originally intended to convert labels to MicroVision, a function which is **not** supported by the 626-6041 Terminal Emulator software or the 627-1002 or 627-1003 MiniCOP systems. Please **disregard** this function.

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

CONFIGURATION SELECTIONS (F1 and F2)

The Configuration Selections display the current port communication parameter settings for the PC's COM ports that are to interface to the MiniCOP's terminal and tape ports. You can configure any COM port on the PC as the Terminal Emulator's TERMINAL PORT or TAPE PORT. When using only one serial COM port on the PC, both the TERMINAL PORT and the TAPE PORT should be configured identically.

<F1> TermCo — selects the Terminal Port Configuration menu as illustrated in Figure 32.

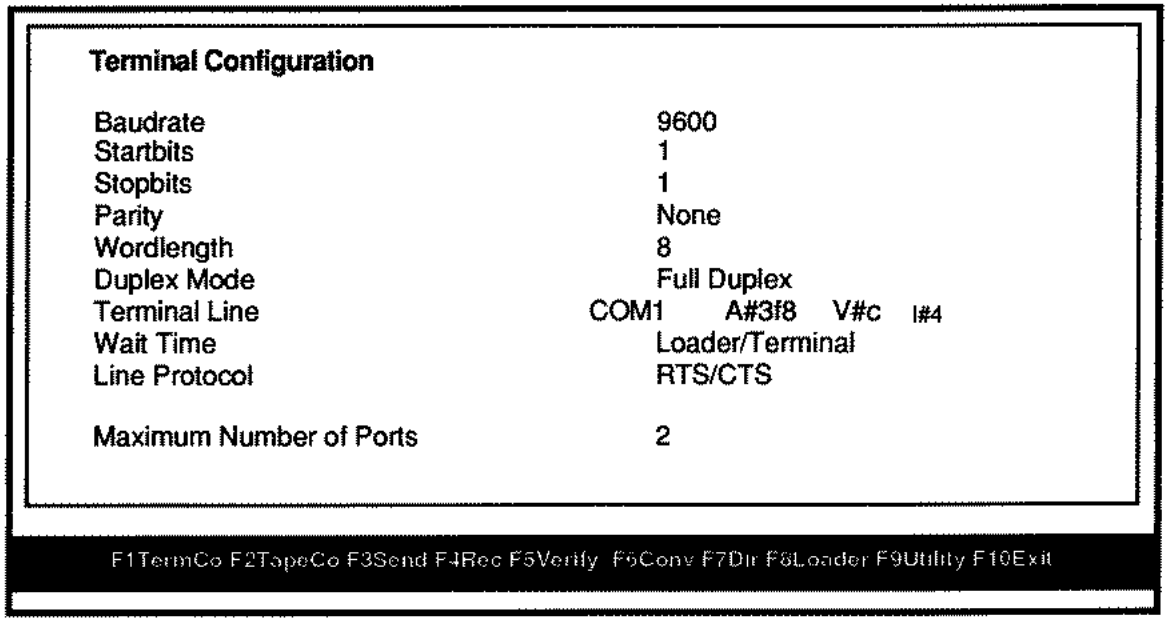


FIGURE 32 - TERMINAL PORT CONFIGURATION MENU

<F2> TapeCo — selects the Tape Port Configuration Menu as illustrated in Figure 33.

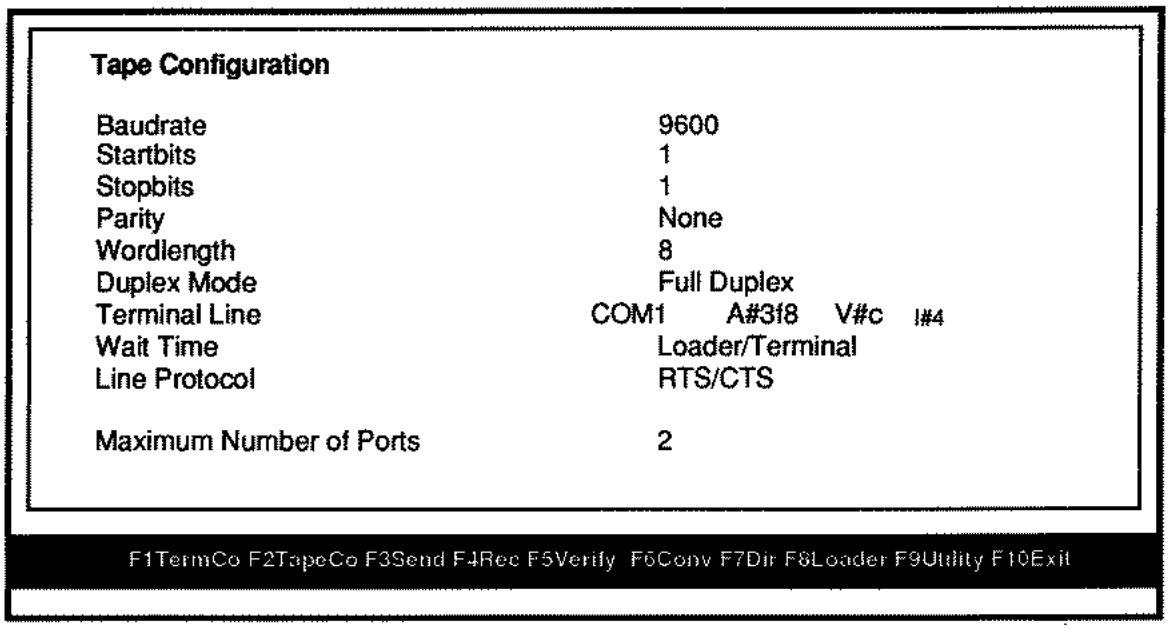


FIGURE 33 - TAPE PORT CONFIGURATION MENU

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

Parameter Definitions

BAUD RATE	Serial data transmission rate in bits/second: 110, 150, 330, 600, 1200, 2400, 4800, 9600, or 19.2K.
START BITS	Number of start bits (first bits) in each word: 1 or 1.5.
STOP BITS	Number of stop bits (last bits) in each word: 1 or 2.
PARITY	Odd, even or none. Word integrity verification: the number of 1's in the word are made odd/even by manipulating the parity bit; none indicates no parity bit.
WORD LENGTH	Length of the actual data word: 5, 6, 7, or 8. Does not include the start, stop, or parity bits.
COMMUNICATION	Refers to the type of duplexed communication being used: full-duplex or half-duplex. Full-duplex - Simultaneous two-way communication. Half-duplex - Communication may flow in either direction, but only one direction at a time.
PORT (Terminal or Tape)	Serial COM ports can be used by the Terminal Emulator for communication to other devices. COM1 Address = $03F8 Vector = $C Interrupt = $4 COM2 Address = $02F8 Vector = $B Interrupt = $3
TERMINAL MODE	This should always be set to Terminal/Monitor.
DELAY	Amount of time the Terminal Emulator will wait to receive data from the Tape Port before aborting the transfer attempt. Also, the amount of time the Terminal Emulator will wait once a transmission has begun and halted before aborting the transfer attempt. Selections: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 40, 50, 60, 70, 80, 90 or 100 seconds. In personal computers without a real-time clock, the amount of delay time may vary slightly.
PROTOCOL	The type of port control used; XON/XOFF (software control lines) or RTS/CTS (hardware control lines).

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

Changing Menu Parameters

Use the UP and DOWN cursor control keys to select the parameter to be changed.

Use the LEFT and RIGHT cursor control keys to select the parameter options. Different parameter options will be displayed as the LEFT and RIGHT cursor control keys are pressed; the options will also wrap-around once the end of the line is reached. If only two options are possible, the LEFT and RIGHT cursor control keys will toggle between them.

To lock in any changes (and/or to exit), press either <RETURN>, <F10>, or <ESC>.

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

FILE TRANSFER, VERIFICATION, AND DISPLAY SELECTIONS (F3, F4, F5 and F7)

<F3> Send

This menu selection accesses the functions needed to transfer a file from the Terminal Emulator to the MiniCOP. The following procedure outlines the steps required for this operation.

1. From the OS9 prompt on the Terminal Emulator, clear the Tape Port of any extraneous data (ex. OS9: CLEAR /TP).

2. From the OS9 prompt on the Terminal Emulator, enter the command syntax to have the MiniCOP receive a file via the Tape Port. **DO NOT PRESS <ENTER>** (for example, OS9:COPY /TP /D0/batchmix).

3. Press <F3> to select the Transmit File Function Window.

4. The prompt will display the directory in which the Terminal Emulator files reside. There are two simple methods for specifying the file to be transferred.

   A. If necessary, use the backspace key to erase the current directory given in the prompt and replace it with the directory pathlist where the file to be transferred resides. Include the name of the file to be transferred from the Terminal Emulator to the MiniCOP;

   or

   B. If necessary, use the backspace key to erase the current directory given in the prompt and replace it with the directory pathlist where the file to be transferred resides. Press <F7> to list all the files in the specified directory. Using the arrow keys highlights the file to be transferred.

5. Press <RETURN> twice to initiate the file transfer.

Figure 34 (below) illustrates the <F3> function window. In Figure 34 the pathlist can be described in the following manner --

C: indicates a specific drive on the personal computer,

623-60 indicates the directory name (this defaults to the directory name where the Terminal Emulator software is located), and

batchmix.SRC indicates the file to be transferred.

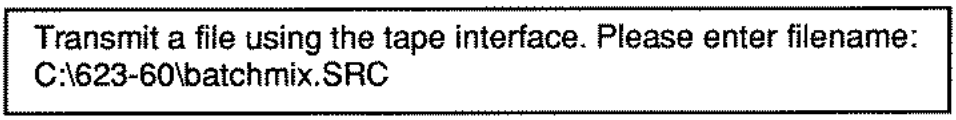


FIGURE 34 - <F3> FUNCTION WINDOW

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

<F4> Receive

This menu selection accesses the functions needed for the Terminal Emulator to receive a file from the MiniCOP. The following procedure outlines the steps required for this operation.

1. From the OS9 prompt on the Terminal Emulator, clear the Tape Port of any extraneous data (ex. OS9: CLEAR /TP).

2. At the OS9 prompt on the Terminal Emulator, enter the command syntax to have the MiniCOP send a file via the Tape Port. **DO NOT PRESS ENTER** (for example, OS9:COPY /D0/cip /TP)

3. Press <F4> to select the Receive function.

4. The prompt will display the directory in which the Terminal Emulator files reside. There are two simple methods for specifying the file to be transferred.

   A. If necessary, use the backspace key to erase the current directory given in the prompt and replace it with the directory pathlist where the file to be transferred resides. Include the name of the file to be transferred from the Terminal Emulator to the MiniCOP;

   or

   B. If necessary, use the backspace key to erase the current directory given in the prompt and replace it with the directory pathlist where the file to be transferred resides. Press <F7> to list all files in the specified directory. Using the arrow keys highlight the file to be transferred.

5. Press <RETURN> twice to initiate the file transfer.

Figure 35 (below) illustrates the <F4> function window. In Figure 35 the pathlist can be described in the following manner:

C: indicates a specific drive on the personal computer,

623-60 indicates the directory name (this defaults to the directory name where the Terminal Emulator software is located), and

reactorsample.pac indicates the file to be transferred.



FIGURE 35 - <F4> FUNCTION WINDOW

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

If the MS-DOS filename already exists, a warning is displayed by the Terminal Emulator. Figure 36 illustrates the warning prompt. If the file is to be overwritten, press the left cursor control key to highlight YES, then press <RETURN>. If the file is not to be overwritten, press <RETURN> while the highlight remains on NO. The time it takes to answer this question is **NOT** a factor in the port timeout process.

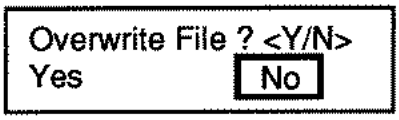


FIGURE 36 - <F4> WARNING PROMPT

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

<F5> Verify

This menu selection accesses the functions needed to verify a file from the Terminal Emulator against a file from the MiniCOP. The following procedure outlines the steps required for this operation.

1. From the OS9 prompt on the Terminal Emulator, clear the Tape Port of any extraneous data. (ex. OS9: CLEAR /TP).

2. From the OS9 prompt on the Terminal Emulator, enter the command syntax to have the MiniCOP send a file via the Tape Port. **DO NOT PRESS <ENTER>.** (for example, OS9:COPY /D0/extruder /TP)

3. Press <F5> to select the Verify File Function window.

4. The prompt will display the directory in which the Terminal Emulator files reside. There are two simple methods for specifying the file to be transferred.

   A. If necessary, use the backspace key to erase the current directory given in the prompt and replace it with the directory pathlist where the file to be transferred resides. Include the name of the file to be transferred from the Terminal Emulator to the MiniCOP;

   or

   B. If necessary, use the backspace key to erase the current directory given in the prompt and replace it with the directory pathlist where the file to be transferred resides. Press <F7> to list all files in the specified directory. Using the arrow keys, highlight the file to be transferred.

5. Press <RETURN> twice to initiate the file verification.

Figure 37 (below) illustrates the Verify with Tape Port Function window. In Figure 37 the pathlist can be described in the following manner:

C: indicates the drive on the personal computer,

623-60 indicates the directory name (this defaults to the directory name where the Terminal Emulator software is located), and

startup indicates the file to be verified.



FIGURE 37 - <F5> FUNCTION WINDOW

As the verification takes place, the byte count will be displayed in the lower right-hand corner of the <F5> Function window.

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

<F7> Directory

This menu selection accesses the functions needed to display all file names in a specified directory on the Terminal Emulator. This function can be used in conjunction with the Send <F3>, Receive <F4>, or Verify <F5> functions as previously described. In addition, this function can be used independently to display files from any drive or directory on the PC that is running the Terminal Emulator software. The following procedure outlines the steps required for this operation:

1. Press <F7> to select the Display Directory function.

2. This function defaults to the directory pathlist where the Terminal Emulator software resides. If necessary, use the backspace key to erase this pathlist and/or drive, then enter the proper pathlist.

3. Press <ENTER> to display all files in the specified pathlist.

A VIEW-ONLY cursor or highlight bar allows you to scan through the directory. This is helpful in the case of a large directory which cannot all be displayed in the window. The cursor allows you to scan through the entire listing. The cursor or highlight bar DOES NOT enable you to select any of the files for execution, however, but it can be used to select a file for the Send <F3>, Receive <F4>, or Verify <F5> function of the Terminal Emulator.

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

SPECIAL FUNCTION SELECTIONS (F8, F9, and F10)

<F8> Loader

Selecting <F8> immediately places you in the Utility software program of the MS-DOS Loader, which allows you to perform ladder logic and text transfer operations. These functions are not directly applicable to the 627 MiniCOP.

<F9> Utility

Selecting <F9> immediately places you in the Loader software of the MS-DOS Loader, which allows you to create, edit, display, and document 620 Platform controller ladder logic programs.

<F10> Exit

Selecting <F10> allows you to exit the Terminal Emulator software package and return to DOS. When this function is selected, the message window illustrated in Figure 38 (below) appears.

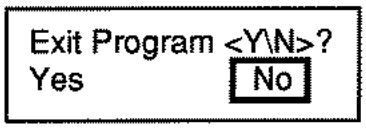


FIGURE 38 - EXIT PROGRAM PROMPT

The left cursor control keys can be used to move the highlight bar to YES. Once the desired choice has been selected, press <RETURN>. Pressing YES immediately exits the Terminal Emulator to DOS.

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

HELP SCREENS (ALT-F10)

By pressing <ALT> and <F10> simultaneously, a help screen is displayed that correlates directly to the Terminal Emulator function presently in use. There are a total of four different help screens related to the Main menu and its selections.

F1 and F2 Help Screen

Functions F1and F2 have an identical help screen that provides additional information on the terminal and the tape port configuration (see Figure 39 below).

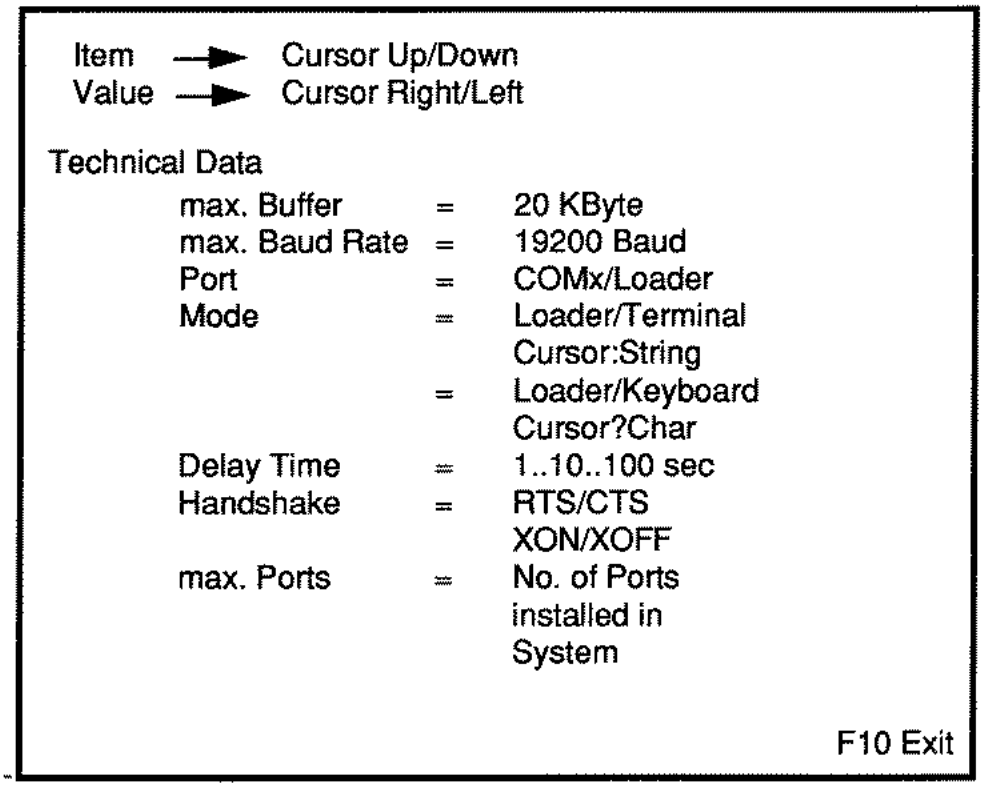


FIGURE 39 - F1 AND F2 HELP SCREEN

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

F3, F4, and F5 Help Screen

F3, F4, and F5 share a help screen that offers general instructions on sending, receiving, and verifying files using the Terminal Emulator (see Figure 40 below).

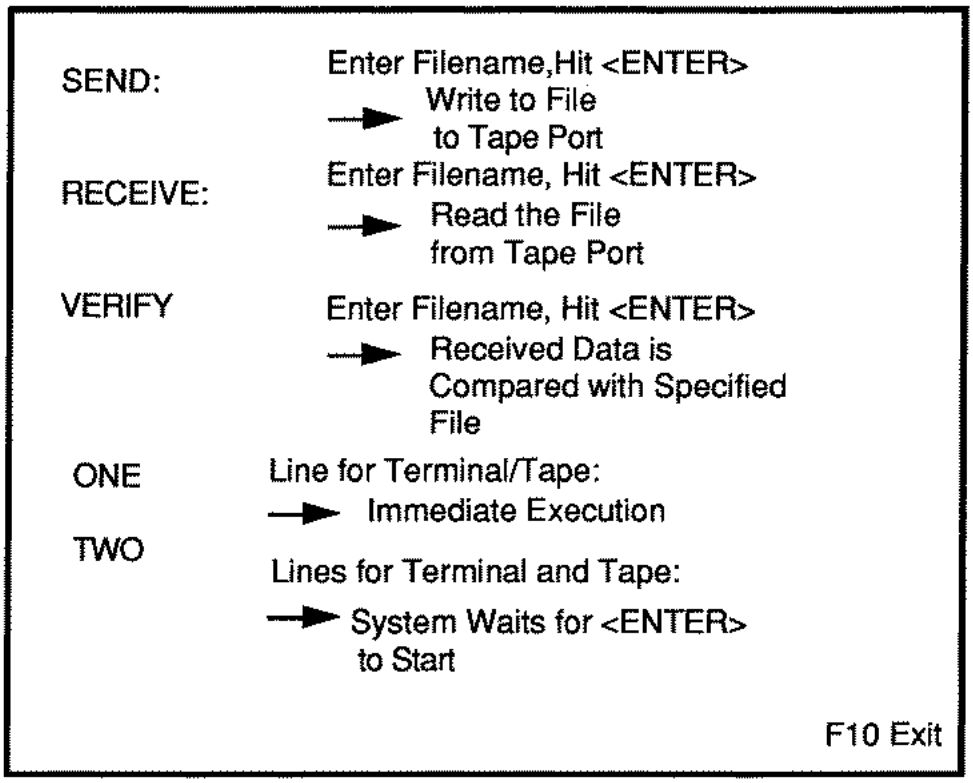


FIGURE 40 - F3, F4, F5 HELP SCREEN

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

Main Menu, F7, and F10 Help Screen

The Main Menu, F7, and F10 also share a Help screen that provides a brief definition of the abbreviated labels associated with each selection (see Figure 41 below).

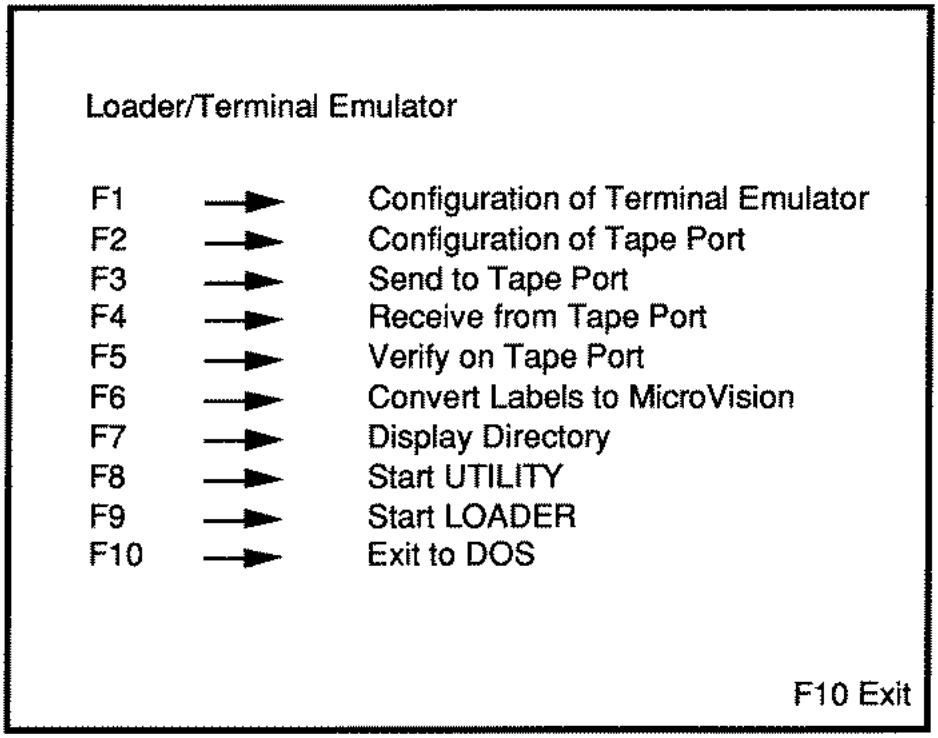


FIGURE 41 - MAIN MENU, F7, AND F10 HELP SCREEN

F8 and F9 Help Screen

The F8 and F9 Help screen offers assistance in interpreting the Main Menus associated with those specific functions.

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

OS-9 Operating System

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

Section 3.
OS-9 Operating System

About This Section

This section presents an operational overview and programming background for the OS-9 Operating System. A commands listing for the OS-9 Operating System is also presented, along with a MiniCOP/OS-9 tutorial.

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

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

OS-9 OPERATIONAL OVERVIEW

OS-9 OPERATING SYSTEM

OS-9 is a versatile multitasking operating system designed for computers based on the Motorola 6809 microprocessor. OS-9 is modeled after Bell Laboratories' UNIX operating system, and has been revised for smaller disk-based systems. OS-9 retains the structure of UNIX, but systems implementation is considerably different.

Users experienced with UNIX should have little difficulty adapting to OS-9. Some of the main differences are summarized in the following:

1. OS-9 is written in 6809 Assembly Language. This improves program size and speed characteristics.

2. OS-9 is designed for a mixed RAM/ROM memory environment and supports reentrant, position-independent code.

3. OS-9 supports the use of reentrant, position-independent firmware that can be shared by several processes at the same time.

4. OS-9 uses the memory module concept for organizing object code, which supports modular software development.

5. OS-9 supports multiple file managers, which are modules that interface a class of devices to the file system.

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

MiniCOP MEMORY STRUCTURE

The basic memory structure for the 627 MiniCOP is illustrated in Figure 42. There are two primary areas that form system memory: main memory and COPRAM.

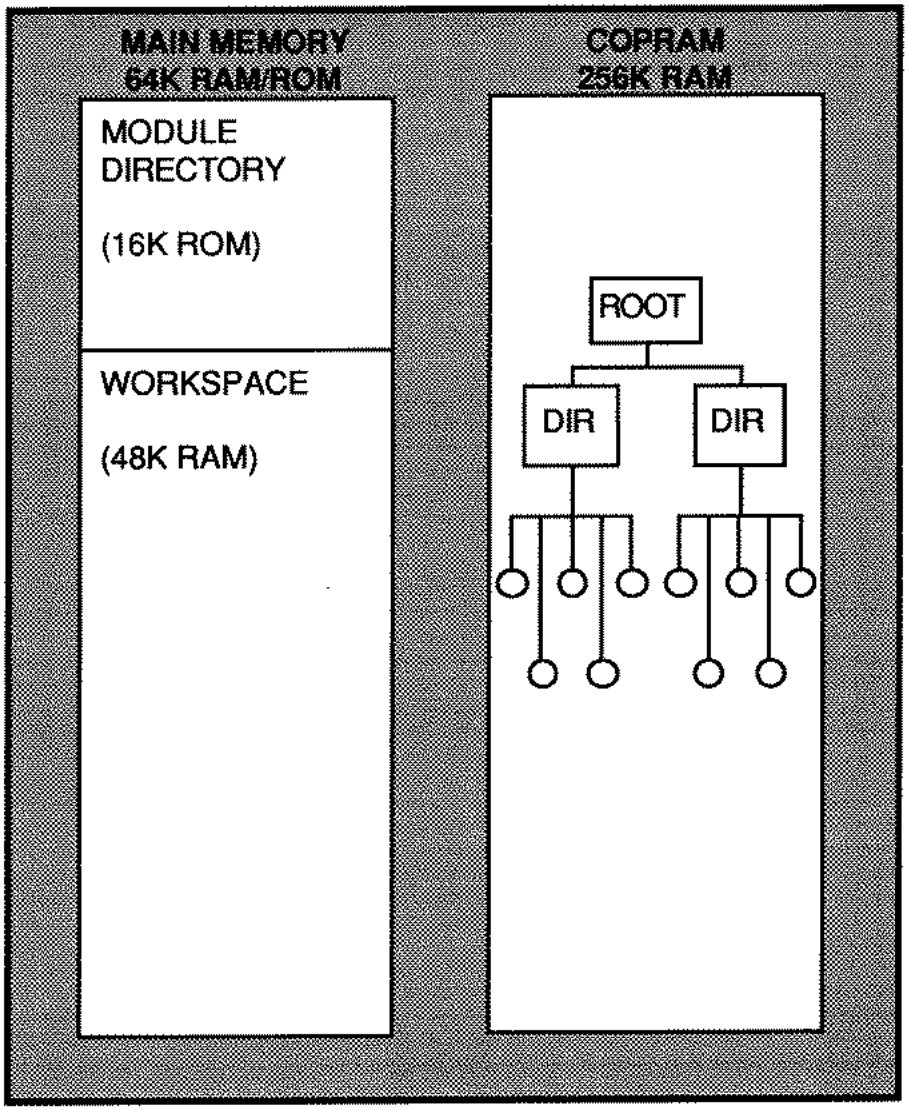
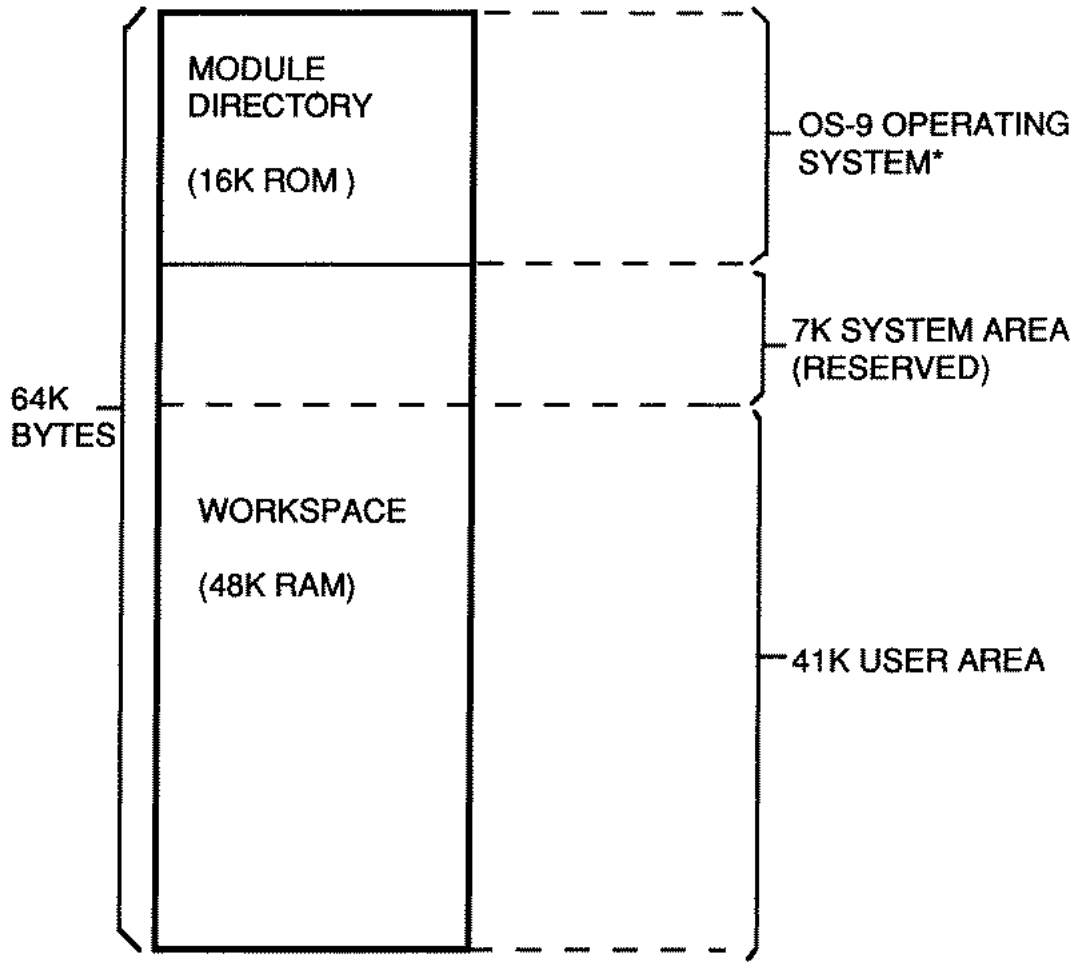


FIGURE 42- 627 MINICOP MEMORY STRUCTURE

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

MAIN MEMORY

The MiniCOP's main memory contains 64K bytes of combined RAM and ROM, which coordinate the activities of the operating system. The MiniCOP's main memory can be further subdivided into two categories: module directory and workspace, as illustrated in Figure 43 below.



NOTE: Workspace sizes are approximate.

* The Module Directory will also (in RAM) contain the names of all programs currently in the workspace.

FIGURE 43 - MAIN MEMORY DETAIL

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

Module Directory

After the MiniCOP has first been initialized, the module directory portion of main memory is a 16K EPROM storage area that contains the OS-9 modules that comprise the operating system (see Figure 43). When a program is loaded into the workspace its name is also listed in the module directory.

Work Space

The workspace portion of main memory is a 48K RAM storage area that includes approximately 41K bytes of user-available space with the remaining 7K Bytes reserved for system usage (see Figure 43). The user area of the workspace is used for BASIC09 program development. Programs in the workspace can access files contained in COPRAM and the memory modules in the Module Directory.

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

COPRAM MEMORY

Structure

This portion of memory is essentially a solid-state RAM disk. In OS-9, COPRAM allows a branching, hierarchical tree structure much like a DOS operating system. Unlike DOS, OS-9 provides for two directory pointers: data directory and execution directory.

The system dynamically allocates memory during the RUN mode. A RAM block can be used by one process, then de-allocated and used by a second process. To allow program allocation and manipulation, each program is stored in a module format. The module format consists of a 16-byte header block, the program, and a 3-byte cyclic redundancy check (CRC). This format allows the system to load and verify programs while manipulating memory.

COPRAM stores data in separate logical entities called files. Each file name is entered within a directory. COPRAM has a root directory, called **D0**, which can include names of files and directories. From the root directory, COPRAM can branch out in a tree-like structure of directories and subdirectories. During the MiniCOP's initialization, part of this directory structure is created as the OS-9 system commands are transferred to COPRAM.

The root directory is created during initialization and contains one subdirectory called **CMDS**. This directory contains the OS-9 system commands and other files written there during initialization. The pathlist for this directory is: **/D0/CMDS** (see Figure 44).

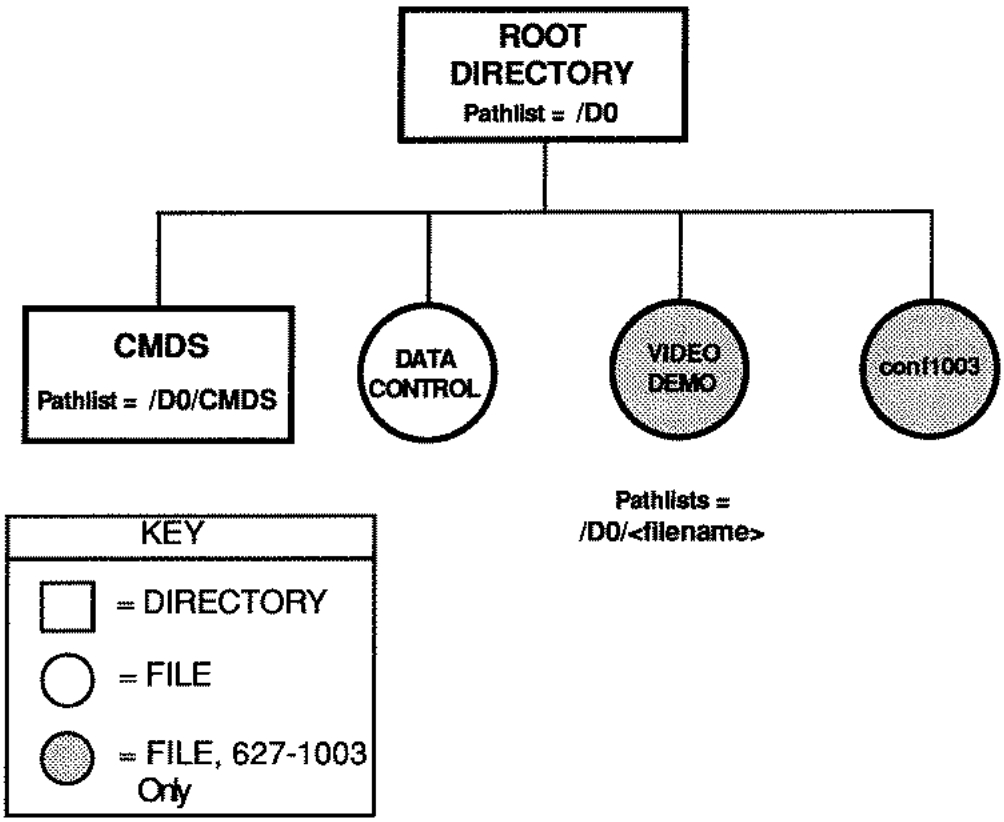


FIGURE 44 - COMPRAM STRUCTURE AFTER INITIALIZATION

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

Figure 45 shows an example of a simple directory structure that might have been created by a user. OS-9 uses pathlists to identify and/or specify specific files in COPRAM. When OS-9 is given a pathlist to process, it reads the pathlist from left to right, and then searches through the directory structure in accordance with the list to arrive at the target location. For example, in Figure 45 the file called **tempstore,** located in the directory DAIRYHISTORY, is identified by the pathlist: **/D0/DAIRYHISTORY/tempstore**. Similarly, the file called **reprtgen,** located in the directory **DAIRYPROCS,** is identified by the pathlist: **/D0/DAIRYPROCS/reprtgen**.

Notice that in Figure 45 the directory names are capitalized while the filenames are lower case. This is a recommended practice that allows directories and files to be easily identified when using the **DIR** command.

Operation

COPRAM consists of RAM-based memory modules. OS-9 treats this memory as though it were a hard disk. COPRAM access is achieved through the Random Block File Manager and is always in 256-byte blocks. The hardware configuration of these memory modules is accomplished through sector mapping. Each block is given a logical sector number that varies between 0 and 572.

Files and Directories

Each file is treated as a separate entity in COPRAM and is referenced by name. No version numbers or file extensions are present in OS-9. OS-9 files should be given a file extension when being transferred to the personal computer which is running the Terminal Emulator software. Only one file per name in a single directory is allowed. Directories are also files that have the directory attribute set. The directory files contain the names of all files and subdirectories that they hold which creates a hierarchical structure.

Speed of COPRAM Transfer

Data can be moved between COPRAM and main memory at a rate of approximately 44K bytes per second. This access speed should allow you to move modular programs in and out of the workspace with minimal delay.

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

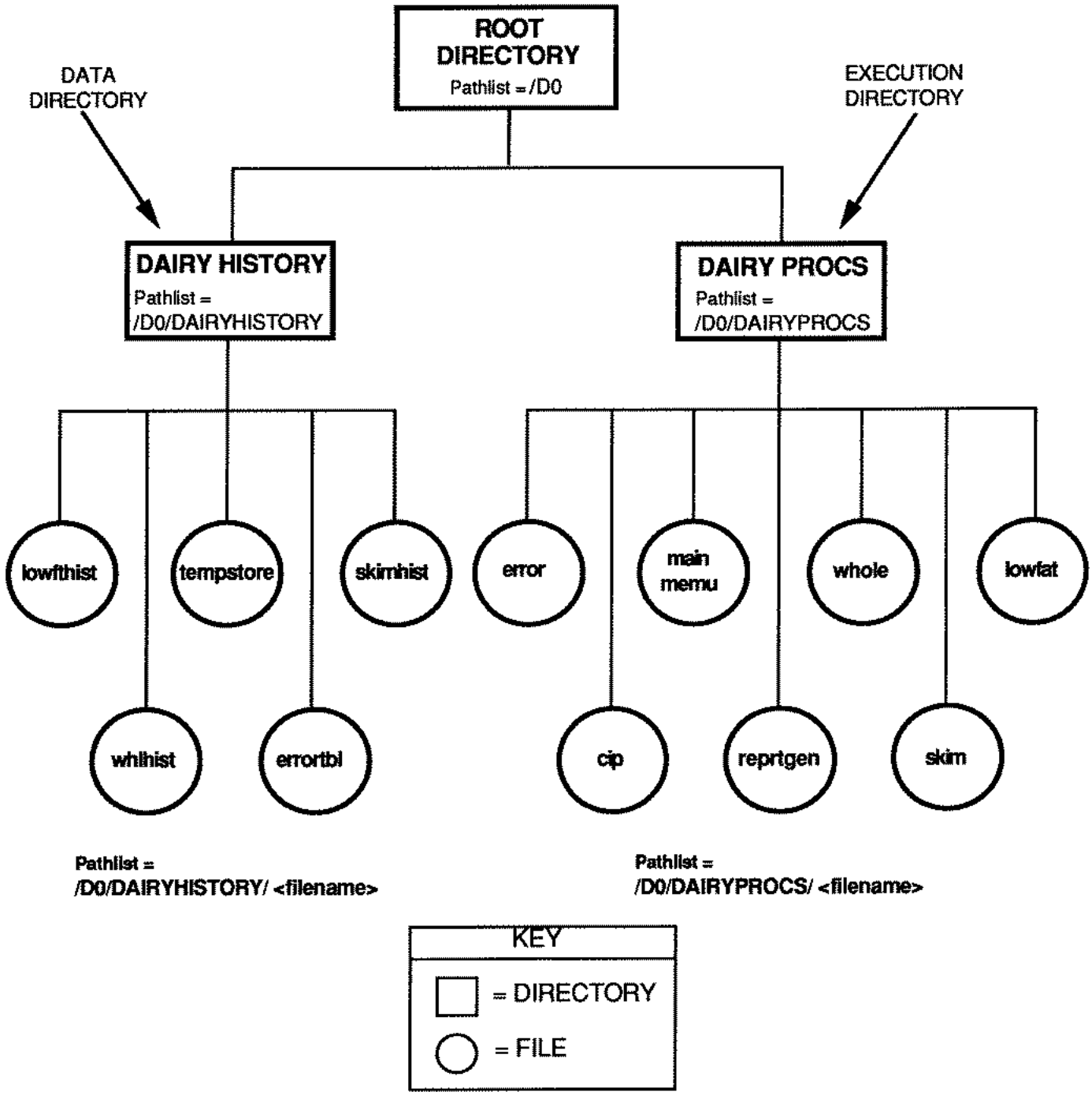


FIGURE 45 - EXAMPLE COPRAM DIRECTORY STRUCTURE

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

PERSONAL COMPUTER MEMORY TO MiniCOP MEMORY ANALOGY

To assist in developing a basic understanding of the MiniCOP's memory structure, an analogy to the memory structure of a typical personal computer can be drawn, as shown in Figure 46.

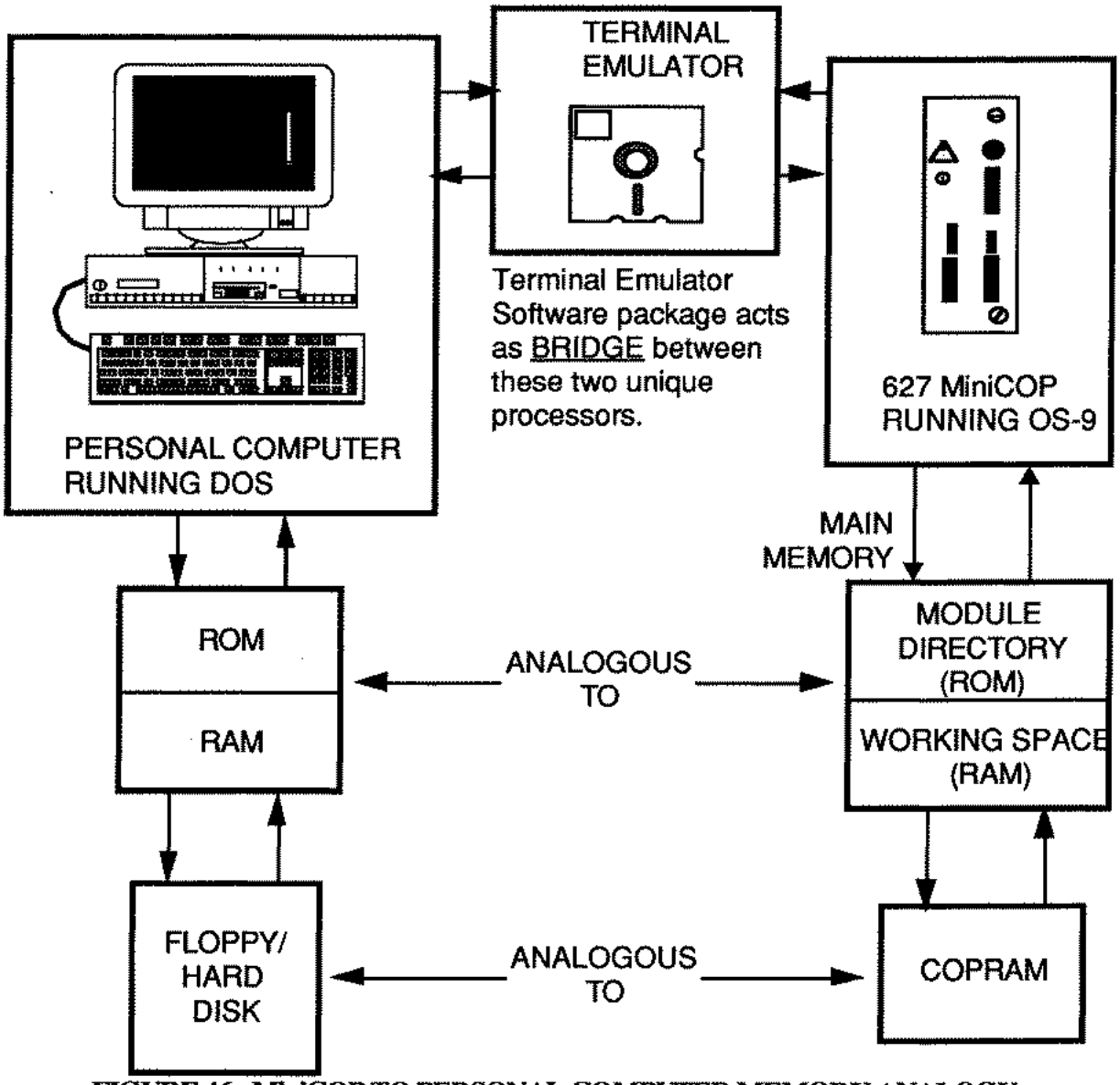


FIGURE 46 - MiniCOP TO PERSONAL COMPUTER MEMORY ANALOGY

The PC's combined ROM and RAM memory is analogous to the MiniCOP's main memory (ROM and RAM). The ROM memory of both the PC and the MiniCOP contain start-up and testing routines which are needed in order to power-up and prepare each device for operation. The RAM memory for both the PC and the MiniCOP is the area in each device where programs are loaded and executed, and where temporary storage of data and variables takes place. The PC's floppy and hard disk drives are used to store programs when they are not in use. In the MiniCOP these programs are stored in COPRAM, a solid-state RAM disk.

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

MAIN MEMORY MANAGEMENT FUNCTIONS

An important OS-9 function is main memory management. OS-9 automatically allocates all system memory to itself and to its processes. It also keeps track of all program modules resident in main memory at any given time.

Using the concept of memory modules, OS-9 manages the logical contents of memory and the physical assignment of memory to programs.

Within main memory, memory is assigned from the higher addresses downward for program modules, and from the lower addresses upward for the data areas. (see Figure 47).

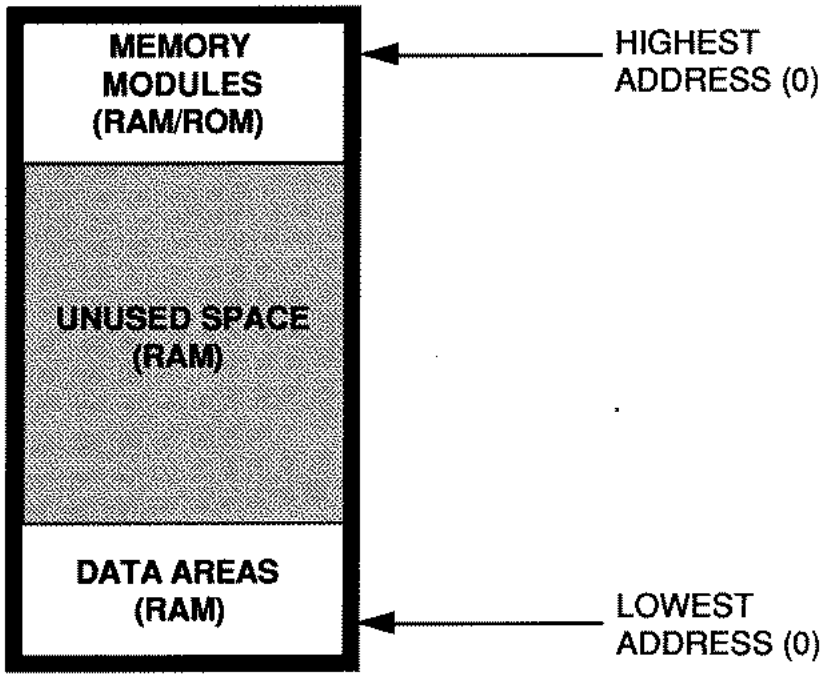


FIGURE 47 - MAIN MEMORY MANAGEMENT

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

OS-9 STRUCTURE

OS-9 is comprised of a group of memory modules, some in RAM and some in ROM, each of which provides specific functions. The modules are arranged in a multilayer organization, as shown in Figure 48. Each of these layers is described in the following paragraphs.

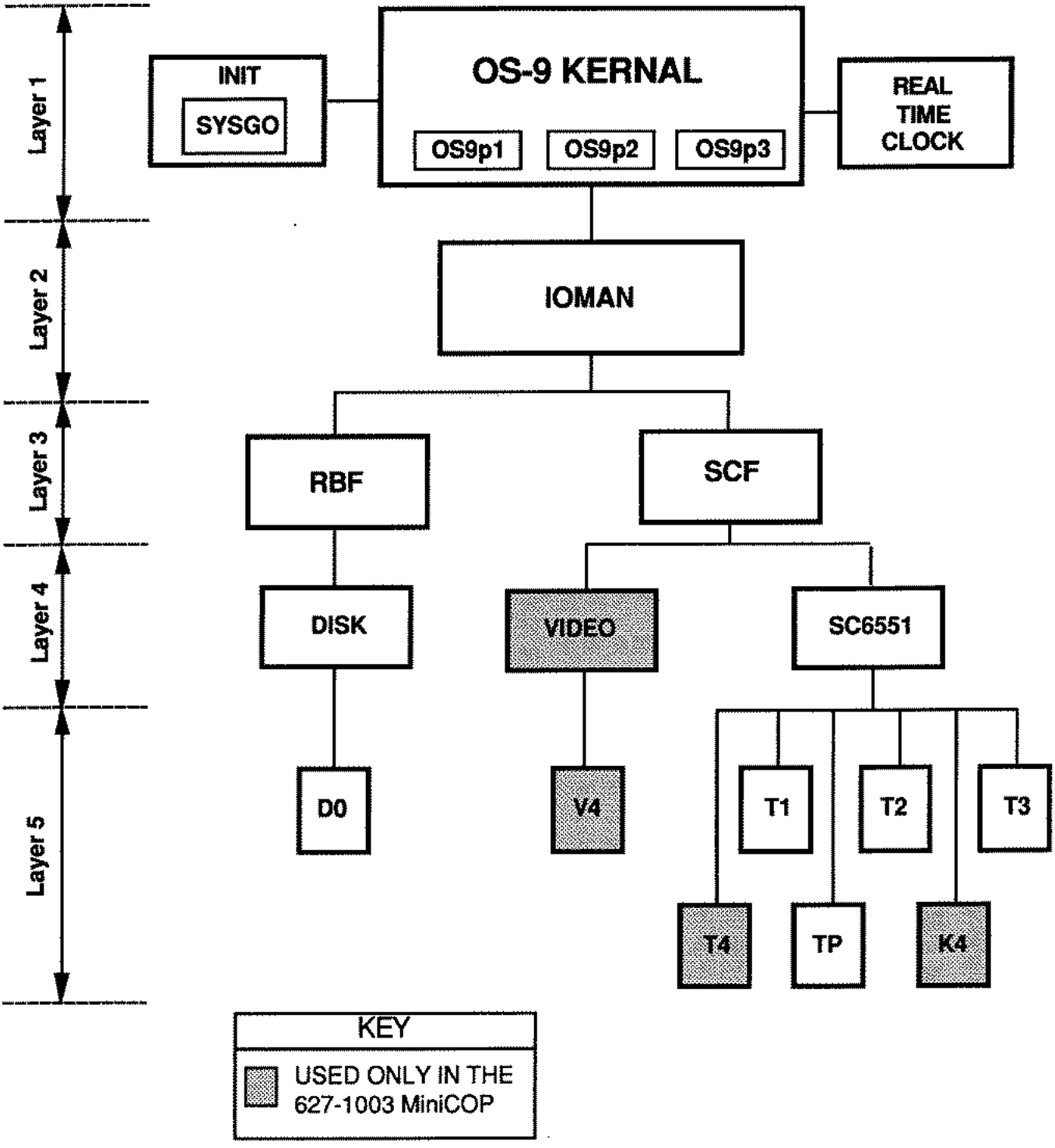


FIGURE 48 - OS-9 ORGANIZATION

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

NOTE

Not shown in Figure 48 is the SHELL, which is the OS-9 Command Interpreter. The SHELL is a program that is **not** necessarily a part of the operating system. Refer to the *OS-9 Programming Background* subsection for additional information regarding the SHELL.

LAYER 1

Layer 1 includes the OS-9 KERNAL (OS9p1, OS9p2, OS9p3), INIT, SYSGO, and the real-time clock. The KERNAL provides services such as multitasking and memory management, and it links all other system modules together to form the OS-9 operating system. INIT is an initialization table used by the KERNAL during start-up. SYSGO is the first process executed upon system start-up.

The KERNAL is the nucleus of the OS-9 operating system; it serves as the system administrator and resource manager. The KERNAL's main functions are:

- System initialization at start-up or after restart,
- Memory management,
- Multitasking management, and
- Low-level interrupt processing.

After a hardware reset, the KERNAL initializes the system, which involves:

- Locating memory modules in ROM,
- Determining the amount of RAM workspace available,
- Locating memory modules in the RAM workspace, and
- Executing the system start-up routine, SYSGO.

Input and output operations are not part of the KERNAL's functions. They are handled by the Input/Output Manager in Layer 2.

LAYER 2

Layer 2 contains the Input/Output Manager (IOMAN) module, which provides common processing for all I/O operations.

LAYER 3

Layer 3 is the file manager level. File managers provide I/O request processing for similar classes of devices. The Random Block File manager (RBF) module processes COPRAM functions. The Sequential Character File manager (SCF) handles the operation of the MiniCOP's serial ports.

LAYER 4

Layer 4 is the device driver level. Device drivers handle the physical I/O functions for specific classes of devices. The MiniCOP uses three device drivers: DISK, VIDEO, and SC6551. The drivers coordinate MiniCOP communication -- the DISK driver with COPRAM, the VIDEO driver with the video ports of the 627-1003 MiniCOP, and the SC6551 driver with the asynchronous serial ports.

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

LAYER 5

Layer 5 is the device descriptor level. Device descriptors are tables that associate I/O ports with their logical names (for example, T2 indicates ASYNC port 2), device drivers, and file managers. They also contain any necessary initialization data. Table 14 lists the device drivers of Layer 4 along with their respective device descriptors in Layer 5 and each descriptor's relationship to the various MiniCOP ports.

TABLE 14 - DEVICE DRIVER/DESCRIPTOR TO MINICOP PORTS RELATIONSHIP

Device Driver	Device Descriptor	MiniCOP Port	Notes
Disk	D0	None	
Video	V4*	Video Ports	4 BNC, RGB Video Connectors
SC6551	T1	Port 1	ASYNC Serial, AKA: Terminal Port
	T2	Port 2	ASYNC Serial, AKA: Tape Port
	T3	Port 3	ASYNC Serial
	T4*†	Port 4	ASYNC Serial
	TP	Tape Port	Used to transfer files to and from the Terminal Emulator
	K4*†	Keybd Port	Modular Keyboard Jack
ATTENTION	MiniCOP Port Buffer sizes equal: • Input — 74 bytes, and • Output — 140 bytes		

* 627-1003 MiniCOP only.
† Cannot be used simultaneously.

NOTE

Device Descriptors V4 and K4 are not included in the module directory after initialization using the OS-9 System Commands disk file 1003SYS. If needed in a particular application they can be installed using the conf1003 utility, which is located in the CMDS (Execution) Directory along with V4 and K4. Refer to *Using conf1003 to Load the V4 and K4 Device Descriptors* in *Appendix J*

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

MEMORY MODULE CONCEPT

OS-9 consists of separate modules that link together to form the operating system. These modules are stored in main memory. OS-9 uses the memory module concept to manage the logical contents of memory and the physical assignment of memory to programs.

All programs that are loaded into main memory utilize the memory module concept. This concept permits OS-9 to maintain the module directory which can consist of both ROM and RAM. After the MiniCOP is initialized, the module directory consists of memory modules contained in the 16K ROM which are responsible for the internal operation of the MiniCOP.

Memory modules can also be added to the module directory. These modules can be prebuilt programs or routines, such as BASIC09, or they can be user-developed programs that have been stored in COPRAM or have been transferred to floppy disk through the Terminal Emulator. As program modules are called for execution they are loaded into main memory's RAM workspace and their names are added to the module directory. When a program module is no longer needed, OS-9 de-allocates its part of memory and removes its name from the module directory.

Memory modules are the basis for the OS-9 operating system's modular environment. Advantages of the memory module concept include:

- Sharing of programs through reentrant coding,
- Modular programs, which allow for ease of programming, editing, debugging, updating, and which reduce memory load, and
- Run-time linking of programs into libraries.

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

MEMORY MODULE STRUCTURE

The memory module concept allows OS-9 to manage the logical and physical contents of memory. The main requirement is that the modules do not modify themselves and that they be written in position-independent code. In this way, OS-9 can load and relocate the memory contents whenever memory space is available.

The operating system keeps track of memory by use of a module directory. The directory contains the addresses of each module and a count of how many processes are using each module. When modules are no longer needed, their memory is de-allocated (modules are unlinked) and their names are removed from the directory. The OS-9 command **MDIR** displays a listing of all modules in main memory. Figure 49 below illustrates the module directory after the MiniCOP has first been initialized. **MDIR** does not actually appear in the module directory until it is entered at the SHELL command line to generate the listing.

```
module directory at 14:35:08

Boottp    TLoad     RBF       D0        Disk
Firq      Video     TP        T1        T2
T3        T4        SC6551    SCF       IOMan
Clock     Sysgo     OS9p3     OS9p1     Init
OS9p1     Shell     Mdir
```

FIGURE 49 - MODULE DIRECTORY LISTING

Each memory module has three parts: module header, module body, and cyclic redundancy check (CRC) value (see Figure 50). The header contains information on the module size, type, attributes, and use. The CRC is a 24-bit value used to check the integrity of the module.

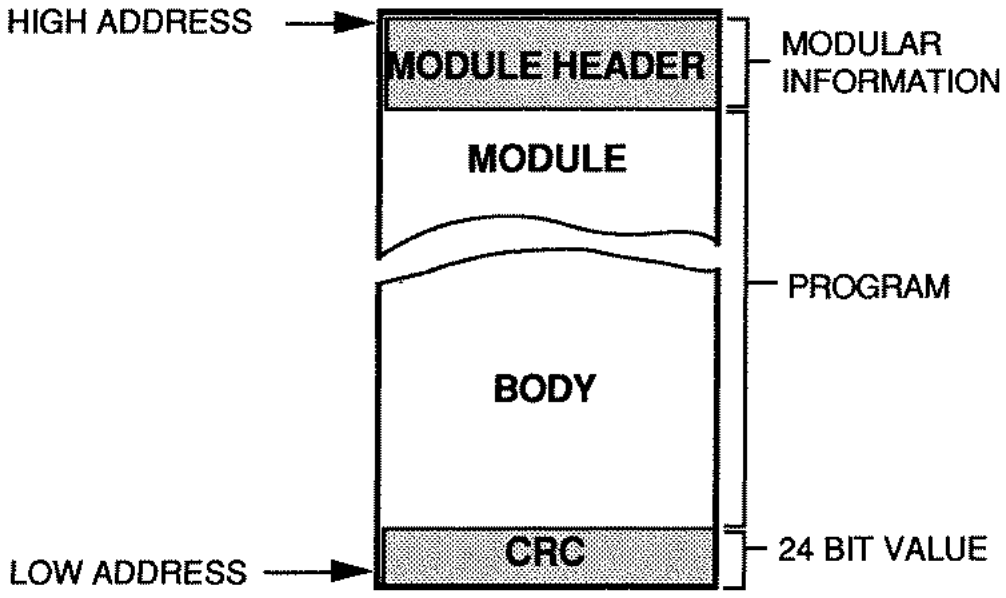


FIGURE 50 - MEMORY MODULE CONTENTS

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

MEMORY MODULE DESCRIPTIONS

OS-9 consists of twelve separate modules that link together to form the operating system. The modules are contained in Layers 1 through 4 of the OS-9 structure. They include: the KERNAL (OS9p1, OS9p2, OS9p3), INIT, SYSGO, the real-time clock, IOMAN, RBF, SCF, DISK, VIDEO, and SC6551.

KERNAL (OS9p1, OS9p2, OS9p3)

The KERNAL modules reside at the very top of memory and contain the vector addresses and interrupt servicing routines. At start-up these modules search for existing BASIC09 procedures (called PRGMA and PRGMB) and execute them if present.

The OS9p1 module is also responsible for setting up memory boundaries, for determining what modules are present in memory, and for reserving main memory for variables and the data table. OS9p3 is also responsible for initializing the real-time clock and its RAM.

INIT

The INIT module contains the defaults for the system. These defaults include the high-free memory boundary, entries in the interrupt polling table, entries in the device table, the standard terminal name, and the initialization module name (SYSGO).

SYSGO

The SYSGO module performs the final initialization before the MiniCOP enters one of its two primary operating modes -- either OS-9 SHELL or RUNB.

IOMAN

The Input/Output Manager (IOMAN) module must be present to perform sequential or random I/O operations. The IOMAN channels data along I/O paths between processes. It has two internal data structures: the device table and the path table.

RBF

The Random Block File (RBF) manager supports random-access, block-oriented devices like COPRAM. It contains the routines necessary to create, open, close, or delete files in COPRAM. It is a reentrant subroutine package that can accommodate any number of such devices. The RBF module buffers the data stream to and from COPRAM and issues requests to the KERNAL for dynamic allocation of buffer memory.

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

SCF

The Sequential Character File (SCF) module supports devices that operate on a character-by-character basis (for example, terminals, printers, modems, barcode readers, etc.). This module specifically contains the GETSTAT and SETSTAT system calls for sequential files. The SCF module is a reentrant subroutine package that can accommodate any number of devices. It includes extensive I/O editing functions such as backspace, line delete, repeat line, and screen pause.

DISK

The DISK module is the device driver that performs the I/O transfers to and from COPRAM. The RBF module calls specific routines in the DISK module through a branch table, which includes:

- Device initialization routine,
- Read from device,
- Write to device,
- Get device status,
- Set device status, and
- Device termination routine.

VIDEO

The VIDEO module is the device driver that performs the video signal transfer to the 627-1003 MiniCOP's video ports.

SC6551

The SC6551 module is an asynchronous communications interface device driver. This module handles all of the register manipulation needed to transfer data to and from the serial ports. It contains the interrupt servicing routine for all ports and also handles interrupt polling.

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

DEVICE DESCRIPTORS

The device descriptors (Layer 5) can be viewed as filters that define all control characters for the three device drivers: DISK, VIDEO, and SC6551. The MiniCOP has eight original device descriptors after initialization: D0, V4, T1, T2, T3, T4, TP, and K4. You can create new device descriptors with the OS-9 command **MAKDESC**, which allows different terminal types to be used. Additionally, to be able to send and receive binary data, the filter must be set to zero. This zeroing is done on the TP (TAPE Port) descriptor. Table 14 lists the device drivers of Layer 4 along with their respective device descriptors in Layer 5 and each descriptor's relationship to the various MiniCOP ports.

Device Driver - DISK

The DISK device driver uses a single device descriptor called D0. D0 allows the DISK device driver to identify certain bit patterns and control characters while transferring information to and from COPRAM.

Device Driver - VIDEO

The VIDEO device driver has a single device descriptor called V4. V4 allows the VIDEO device driver to transfer RGB color and monochrome video signals to the MiniCOP's four BNC connector video ports. The device driver VIDEO and the device descriptor V4 are only available for use with the 627-1003 MiniCOP.

Device Driver — SC6551

The device driver SC6551 has six device descriptors T1, T2, T3, T4, TP, and K4. These device descriptors allow the SC6551 device driver to identify certain bit patterns as control characters. By using the device descriptor for the appropriate port, the SC6551 responds to particular bit patterns in a set manner. Error reporting and control signal processing are done automatically by SC6551. The device descriptors T4 and K4 are only available for use with the 627-1003 MiniCOP as they correspond to port 4 (ASYNC, 25-pin serial) and the modular keyboard port respectively.

NOTE

Device Descriptors V4 and K4 are not included in the module directory after initialization using the OS-9 System Commands disk file 1003SYS. If needed in a particular application they can be installed using the conf1003 utility, which is located in the CMDS (Execution) Directory along with V4 and K4. Refer to *Using conf1003 to Load the V4 and K4 Device Descriptors* in *Appendix J*

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

SYSTEM PROGRAM MODULES

The following three additional modules are present after initialization: SHELL, BASIC09, and RUNB.

SHELL

The SHELL is the command line interpreter that allows you to direct the operating system, through the Terminal Emulator, to perform specific tasks. These tasks can include starting and stopping of processes, changing priorities, changing the current data or execution directory, or other operations. Several SHELL modules can exist in main memory simultaneously. Care must be taken to avoid depleting the systems workspace memory when using multiple SHELLs.

BASIC09

BASIC09 is the complete BASIC editing, debugging, and run-time package. It is approximately 23K bytes in size. Like the SHELL, multiple BASIC09 workspaces can exist in memory simultaneously, but again, care must be used to avoid depleting the workspace memory.

RUNB

The RUNB module contains only the execution portion of the BASIC09 programming language and is approximately 12K bytes in size. RUNB cannot be used to program, edit, or debug BASIC09 procedures. RUNB is utilized when a program or group of programs to be run on the MiniCOP have been tested and verified for proper operation. In this case an additional 11K bytes of workspace (BASIC09 23K - RUNB 12K = 11K) will be available for user programs. Several BASIC09 procedures (programs) can exist in the RUNB workspace simultaneously.

REAL-TIME CLOCK OPERATION

The MC146818 clock IC is a real-time clock plus RAM. The clock counts in seconds, minutes, hours, days, months, and years. The clock generates interrupts at a specified rate. Two types of interrupts are generated by the clock: ticks and seconds. The "tick" interrupt is generated sixty-four times per second for multitasking operations. The "second" interrupt is generated once per second and is used to test the battery level. The clock can also be used for time and date stamping of COPRAM files, or it may be transmitted to external control/sensing devices via the serial ports or backplane connections.

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

INTERRUPT PROCESSING

All OS-9 system interrupts are of the IRQ type. When an interrupt occurs, all of the registers are placed on the stack and the service routine is called. There are only eight interrupt devices. The polling sequence taken by the system to locate the origination of an interrupt is as follows:

1. Real-time clock
2. T1, ASYNC port 1
3. T2, ASYNC port 2
4. TP, TAPE port (Terminal Emulator on ASYNC port 2)
5. T3, ASYNC port 3
6. T4, ASYNC port 4
7. V4, Video port
8. K4, modular keyboard port

The first device checked in the polling scheme is the clock. The real-time clock generates a "tick" every 15.625 msec (1/64th second) and a "second" interrupt every second.

If the interrupt is a tick, the system determines if any other processes are waiting for processor time; checks their priority; and takes the appropriate action.

If the interrupt is a second, the update routine is called and the time values are updated. In addition, the battery's level is tested through the PIA and is recorded in memory.

If the interrupt was **not** generated by the clock, each of the remaining devices are polled until the origination of the interrupt is located. If no originating device is found, the system resumes program execution.

Interrupt processing involving multitasked programs that service multiple ports uses the same interrupt processing routine as programs that are not run concurrently. Multitasking introduces baud rate restrictions for all ports being serviced concurrently.

The maximum combined baud rate for all ports is 19.2K baud. If, for example, two ports on a MiniCOP are being accessed in a system running multitasked programs or are being used concurrently to perform data collection from field devices, the maximum baud rate for each port is 9600 (9600 + 9600 = 19200 or 19.2K baud).

CAUTION

The maximum combined sum of all MiniCOP port baud rates must not exceed 19.2K. If any one port or group of ports on a single MiniCOP are set at a combined rate of 19.2K baud, no other ports can be serviced reliably. If the baud rate of these ports is decreased, to limit the total baud rate, other ports can be added, provided the maximum baud rate of 19.2K for interrupt processing is not exceeded.

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

MULTITASKING AND OS-9

OS-9 is a multitasking operating system which allows several independent programs to be executed concurrently. For several programs to run concurrently and without interferences, OS-9 must perform many coordination and resource allocation functions. The major system resources managed by OS-9 are:

1. CPU time,
2. Memory (main memory and COPRAM), and
3. The input/output system.

In order for the MiniCOP to provide reasonable performance, these resources must be managed in the most efficient manner possible. OS-9 uses many techniques and strategies to optimize throughput and capacity in order to provide the best performance.

Multitasking functions use a hardware real-time clock that generates interrupts at a regular rate of sixty-four per second. The microprocessor (MPU) time is, therefore, divided into periods of 1/64th second duration, known as "ticks". Processes that are active and not waiting for some event are run for a specific period called a time slice.

PROCESSOR ALLOCATION AND TIME SLICING

CPU time is a finite resource that must be allocated wisely to maximize the MiniCOP's throughput. It is characteristic of many programs to spend much unproductive time waiting for some event such as an I/O operation to occur. An example of unproductive time is an interactive program that communicates with an operator at a terminal on a line-by-line basis. Every time the program has to wait for a message to be displayed to the terminal and/or the operator to respond, it typically cannot perform any useful function. An efficient multitasking operating system automatically assigns CPU time to those procedures that can effectively use it, thereby avoiding much of the wasted CPU time.

OS-9 uses time-slicing to share CPU time between active processes. Time-slicing is implemented using both hardware and software functions. Each second of CPU time is sliced up to be shared among several processes. At each tick, OS-9 can suspend execution of one program and begin execution of another. How frequently a process is given time slices depends on its assigned priority relative to the other active processes.

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

PROCESS STATES

The OS-9 Operating System treats each running program as a process. Each process has an ID number and certain attributes. The processes can be ACTIVE, SLEEPING, or WAITING . A process's attributes include parent program, priority, and memory allocation. The process states are defined as:

ACTIVE	A process that is ready for execution.
WAITING	A process that has been suspended until a child process terminates or a signal is received.
SLEEPING	A process that has been suspended for a specific period of time.

A process is entered into a queue and stays there until it has terminated or the process state changes. In the active queue each process has a priority. This priority allows all processes in the active queue to obtain a portion of processor time in relation to its priority. The priority numbers range from 0 to 255, with 255 as the highest priority. OS-9 uses a scheduler algorithm that ensures all active processes receive execution time. At each time slice, every priority number is incremented and the highest priority is given the time slice. This process repeats every 15.625 msec (tick). Once a process has reached 255, it is reset to its original priority number.

Each process can receive its own interrupts or signals. A process can also signal to other processes. These interrupts must be prepared for by setting an interrupt trap.

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

CREATION OF A NEW PROCESS

The sequences of operations required to create a new process and initially allocate its resources are automatically performed by OS-9. If for any reason any part of the sequence cannot be performed, the creation process is aborted and the calling procedure is passed an appropriate error code. The most common reason for failure is the unavailability of required resources, especially memory. The second most common reason for failure is that the program specified cannot be found. A single procedure can create infinite new processes within the limitation of unassigned memory available.

Process creation takes place in a tree-like, hierarchical structure. The tree-like structure makes it easy to describe the relationship between processes (parent and child), and it is used extensively in descriptions of OS-9's multitasking operations.

When a procedure creates a new process, it must specify the primary module and its parameters. The new process will also inherit certain properties from its parent (creating procedure).

Each new process has the following items associated with it:

- Primary module
- Parameters
- Inherited properties
- Assigned properties

PRIMARY MODULE

The primary module is the name of the program to be executed by the new process. The program can already exist in memory or OS-9 can load it from COPRAM. The COPRAM file will have the same name as the primary module.

PARAMETERS

Parameters are pieces of data specified by the calling procedure to be passed to and used by the new process. This data is copied to a part of the new process's memory area. Parameters are used to pass file names, initialize values, etc. The SHELL also passes command line parameters in this manner.

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

INHERITED PROPERTIES

The new process will inherit copies of certain properties of its parent (called process).

User ID Number

The user ID number is used by the file security system to identify all procedures belonging to a specific user. This is not the same as the process ID number which identifies a specific process. The system manager is always set to user number "0".

Standard Input and Output Paths

The standard input and output paths are three paths (input, output, and error/status) used for routine input and output operations. Most paths (files) can be used simultaneously by two or more processes.

Current COPRAM Directories

OS-9 allows for two directory pointers -- data directory and execution directory. The current data and execution directories used by the parent process are passed to the new process (child or called process).

ASSIGNED PROPERTIES

As part of the new process creation, OS-9 automatically assigns the following properties:

Process ID Number

A process ID number from 1 to 255 is assigned to each new process and is used for process identification. Other processes refer to this ID number when initiating communications with the process. The process ID number is not the same as the user ID number.

Memory

OS-9 allocates enough workspace memory to run the new process. Included for the process's exclusive use in this memory allocation is a data area, which is used for the program's parameters, variables, and stack. A second workspace memory area may also be required to load the module if it is not currently in main memory.

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

LOADING PROGRAM MODULES INTO MEMORY

The following steps outline the loading of a program module.

1. OS-9 first attempts to find the referenced program module in the module directory which holds the beginning address of every module in main memory. Program modules are programmed in reentrant code. This allows the exact same module to be shared by two or more different processes simultaneously without effecting each other, provided that each incarnation of the program has an independent memory area for its variables.

   Almost all OS-9 family software is reentrant and can make the most efficient use of memory. For example, the program module BASIC09 uses about 23K bytes of memory. If a request is made to run BASIC09, but another process is using it, both processes will share the same copy. OS-9 automatically keeps track of how many processes are using each program module. OS-9 will delete the module when all processes using it are terminated. This will then free its memory for use elsewhere.

   If the requested program module is not already in main memory, the name is used as a pathlist (file name) and an attempt is made to load the program from COPRAM.

2. Once found, OS-9 checks the module header. The header contains information pertaining to the program such as its size, amount of memory needed for variable storage, etc. OS-9 allocates the specified RAM workspace for the process.

   A program module can be assigned more workspace than called for by the header. This is done by specifying an optional parameter on the SHELL command line or in the calling routine. OS-9 will always allocate the larger of the amounts.

NOTE
For a more detailed explanation and description regarding the assignment of more workspace than specified in the module header, refer to the SHELL in the OS-9 Programming Background subsection.

3. If OS-9 can perform all of the previous steps, it adds the new process to the active process queue for execution scheduling. If it fails to perform **any** of these tasks, loading is aborted and the calling procedure is informed of the error.

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

OS-9 PROGRAMMING BACKGROUND

This section of the manual describes the features of the OS-9 operating system. It also describes the best ways to take advantage of these features. Areas of special note include:

- The SHELL, which provides an interface between the user and OS-9.
- The unified I/O system, along which data is transferred.
- The COPRAM directory and files system.
- Multitasking vs. sequential execution.

NOTE

OS-9 commands are described in this section of the manual to provide a practical reference to the topics discussed. For a more detailed explanation and description of all OS-9 commands, refer to the OS-9 System Commands Listing subsection.

THE SHELL

OPERATIONAL OVERVIEW

The SHELL is a command interpreter program that gives you convenient and versatile access to OS-9's powerful capabilities. It provides an easy-to-use interface to the internal functions of the operating system. When data is entered at the Terminal Emulator, the SHELL reads and interprets this data as a sequence of commands. The SHELL can also be called from a program while it is executing.

The SHELL reads and interprets one line of text at a time from a standard input path (keyboard or program). After each line is read and interpreted by the SHELL, it reads another until it finds an end-of-file marker, at which time it terminates itself. When the SHELL has been called by another program, it interprets the given argument list as the first line of input. If this command line consists of ONLY built-in commands, more lines will be read and processed; otherwise the SHELL terminates and control is returned to the calling program after the single command line is processed.

The SHELL is entered following system start-up or can be called from BASIC09. When the SHELL is active and waiting for a response or input, it displays the prompt **OS9:**.

This prompt indicates that the SHELL is active and awaiting a command from the keyboard. Respond by typing a command line followed by a [RETURN] (note that the carriage return should always be the last character of a command line). OS-9 will make no distinction between upper and lower case alphabetic characters; either case or a combination of both can be used.

The command line always begins with a name of a program module (which must currently exist in memory) or a pathlist (which describes the route to a file in COPRAM). A pathlist may include:

- An **OS-9 COMMAND**. These are actually names of the program modules, which are machine language programs that act as a system utilities (e.g. DIR, COPY, LIST, MDIR, etc.).
- The name of a **PACKED BASIC09 PROCEDURE**. BASIC09 procedures can be packed, or have their unnecessary source code removed (that is, line numbers, nonexecutable statements, etc.).
- The name of a **PROCEDURE FILE** that was created with the BUILD command. Procedure files are a type of text file that contain an OS-9 command or series of commands.

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

As previously mentioned OS-9 provides for two COPRAM directory pointers; data directory and execution directory. The data directory typically is used to store text files and BASIC09 source code procedures. The execution directory typically includes OS-9 commands, packed BASIC09 procedures, or other executable files/programs.

The SHELL searches for the specified program name or pathlist in the following sequence:

1. **MAIN MEMORY** -- if the program is found in main memory it is executed.

2. **EXECUTION DIRECTORY** -- if the program name is found in the current COPRAM execution directory it is executed.

3 **DATA DIRECTORY** -- if the program name is found in the current COPRAM data directory it is executed as a procedure file.

The SHELL itself does not execute commands. System commands are individual machine language programs stored in the COPRAM execution directory CMDS (which was created during initialization) or in main memory. They are called by the SHELL when their names are entered. When a program has ceased execution (whether it was completed or aborted), control is returned to the SHELL.

Table 15 is a quick reference guide that includes the OS-9 commands and a brief description. The structure listed reflects the general command construction and not the formal syntax.

NOTE

Refer to *OS-9 System Commands Listing* for a detailed description of the commands listed in Table 15.

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

TABLE 15 - OS-9 COMMAND QUICK REFERENCE GUIDE

Command Structure	Description
ATTR <pathlist> [permissions]	Display/change file security permissions.
BINEX<pathlist> <pathlist2>	Convert binary file to S-record (hexadecimal).
BUILD <pathlist>	Build a text file from standard input.
CHD <pathlist>	Change current data directory.
CHX <pathlist>	Change current execution directory.
CLEAR </descriptor>	Clear port's input buffer of extraneous characters.
COPY <pathlist> <pathlist2> [mod.]	Copy a file from one pathlist to another.
DATE [t]	Display system date and time.
DEL <pathlist>	Delete a COPRAM file or program modules.
DELDIR <directory name>	Delete a directory and all files in it.
DIR <pathlist> [e] [x]	Display the names of files contained in a directory.
DISPLAY<Hex Number>	Displays the ASCII character equivalent of hexadecimal.
DSKBAK	Saves, loads, verifies COPRAM memory.
DUMP <pathlist>	Displays COPRAM file data in hexadecimal and ASCII.
EX <module> <modifiers>	Execute a program as an overlay.
EXBIN <pathlist> <pathlist2>	Convert S-record (hexadecimal) to binary.
FREE	Display free memory remaining in COPRAM.
INDENT <pathlist>	Display OS-9 module header information.
INKEY <path number>	Reads a path's buffer and returns any character found.
KILL <process ID>	Abort a process.

(Continued)

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

TABLE 15 - OS-9 COMMAND QUICK REFERENCE GUIDE *(Continued)*

Command Structure	Description
LINK <module>	Link a program module into memory.
LIST <pathlist>	List the contents of a text file.
LOAD <pathlist>	Load module from COPRAM into main memory.
MAKDESC	Create a port device descriptor.
MAK DIR <pathlist>	Create a COPRAM directory file.
MDIR	Display module directory.
MERGE <pathlist1> <pathlist2>	Copy and combine files.
MFREE	Display free main memory.
PD [d] [x]	Display current data or execution directory pathlist.
PROCS	Display currently executing system processes.
RENAME <pathlist> <new name>	Change COPRAM file name.
RESET	Initiates a software reset routing.
SAVE <pathlist> <module>	Saves memory modules to a COPRAM file or MiniCOP port.
SETIME	Set system clock.
SETPORT	Modifies baud rate, parity, stop bit, and time out parameters.
SETPR <process ID> <number>	Set process priority.
SHELL <argument list>	Call the OS-9 command Interpreter.
SLEEP <tick count>	Suspend a process for a period of time.
TLOAD <pathlist>	Load a program module from Terminal Emulator or COPRAM.
TMODE <pathnumber> <argument list>	Change terminal operating parameters.
UNLINK <module name>	Remove a program module from main memory.
VERIFY <module name > <module name>	Verifies and/or updates module header and CRC value.
XMODE <device name>	Examine or change port operating parameters.

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

COMMAND LINE PARAMETERS

The program name may be optionally followed by one or more parameters that are passed to the program called by the SHELL. A command line may also include one or more modifiers, which are specifications used by the SHELL to alter the program's standard input/output files or memory assignments.

Command line parameters have two functions: to specify a file name(s), and to select specified options to be used by the program specified on the command line. Parameters are separated from the command name and from each other by space characters (parameter names cannot themselves include spaces). Two examples of SHELL command lines are as follows:

```
OS9:LOAD file1 file2 file3

OS9:MDIR e
```

In these examples:

OS9:	is the OS-9/SHELL command line prompt.
LOAD	is an OS-9 command.
MDIR	is an OS-9 command.
file1 file2 file3	are three file names (parameters) passed to the LOAD command.
e	is the modifier or option given to the MDIR command.

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

COMMAND LINE PROCESSING

The SHELL is a program that reads and processes command lines one at a time from its input path (Terminal Emulator keyboard entry or a SHELL call from an executing program). Each line is first scanned to identify and process any of the following parts that may be present:

- A **PROGRAM, PROCEDURE** or **COMMAND** name.
- **PARAMETERS** to be passed to the program, procedure, or built-in command name.
- Execution **MODIFIERS** to be processed by the SHELL.

Note that only the program, procedure, or command name needs to be present. All other parts are optional. After the program, procedure, or command has been identified, the SHELL processes any modifiers that may be present. Any other text not yet processed is assumed to be a parameter(s) and is passed to the item being called.

Except for the built-in commands, the SHELL will run the item named as a new process or task. The SHELL then deactivates itself until the process has terminated. Once the program terminates, the SHELL reactivates and is ready to receive another command line input. This will occur continuously until an end-of-file ([CONTROL-]]) condition is detected on the SHELL's input path. The SHELL then terminates its own execution.

The following sample SHELL command line and description calls BASIC09 to run a procedure called, **reportgen:**

```
OS9:BASIC09 REPORTGEN #12K>/T2
```

In this example:

OS9:	is the OS-9/SHELL command line prompt.
BASIC09	is the program module to be executed.
reportgen	is the parameter passed to BASIC09.
#12K	is a modifier that requests the process be assigned 12K Bytes of workspace memory.
>/T2	is a modifier that redirects the output (listing) to a printer connected to port 2.

The program, procedure, or command must be the first name listed in the command line. After it has been scanned, the SHELL first checks to see if it is a built-in command. If it is, it is immediately executed. Otherwise the SHELL assumes that it is a program name and attempts to locate and execute it.

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

BUILT-IN SHELL COMMANDS AND OPTIONS

When processing input lines, the SHELL looks for several special names of commands or options that are built into the SHELL. These commands are executed without loading a program and creating a new process. They affect how the SHELL operates and can be used at the beginning of a command line, or following any program separator. Two or more adjacent built-in commands can be separated by the space or comma characters.

The built-in commands and their functions are listed below.

CHD <pathlist>	Change the working COPRAM data directory pointer to the directory specified in the pathlist.
CHX <pathlist>	Change the working COPRAM execution directory pointer to the directory specified in the pathlist.
EX <module name>	Directly execute the module named. It permits a transition from the SHELL to another program without creating another process, thus conserving main memory.
W	Wait for any process to terminate.
KILL <process ID>	Abort the process specified.
SETPR <process ID> <number>	Changes process priority to the number specified.
X	Causes the SHELL to abort on an error.
-X	SHELL does not abort on an error.
P	Turns the SHELL prompt and messages ON (default setting).
-P	Inhibits the SHELL prompt and messages.
T	Causes the SHELL to copy all input lines to output.
-T	Does not copy input lines to output (default setting).

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

COMMAND GROUPING

Sections of SHELL command lines can be enclosed in parentheses. This permits modifiers and separators to be applied to an entire set of programs, procedures, and commands. The SHELL processes them by calling itself recursively (as a new process) to execute the enclosed list. Take for example the following two command lists. Both perform the same function.

```
OS9:DIR >/T2; DIR x >/T2
```

gives the same results as...

```
OS9:(DIR; DIR x) >/T2
```

In both cases the contents of the current data directory (DIR command) and the current execution directory (DIR x command) are listed to the device at T2 (that is, the serial printer). The subtle difference is that the printer is "kept" continuously in the second example. In the first example, another program can "steal" the printer between the execution of the DIR commands in a multitasked environment.

Command grouping can be used to execute a group of programs sequentially with respect to themselves, but concurrently with respect to another process. A case in point:

```
OS9:(DEL file1; DEL file2; DEL file3)&
&004
OS9:LIST file4 /T3
```

In this case the LIST operation shares processor resources with the combined DEL commands. Within the parentheses, the DEL commands are executed sequentially, but that operation is concurrent with the LIST command.

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

OS-9 INPUT/OUTPUT OPERATIONS

UNIFIED INPUT/OUTPUT SYSTEM

OS-9 has a unified input/output system that transfers data to all I/O devices in almost exactly the same manner, regardless of the specific hardware devices involved. The differences between devices are overcome by defining a set of standardized logical functions for all I/O devices. Devices are made to conform to these conventions through software routines that eliminate hardware dependencies. This produces a much simpler and more versatile input/output system.

OS-9's versatile I/O system is based on logical I/O entities called paths. Paths are similar to software I/O channels that route data from a program (process) to a COPRAM file or serial port. All input/output operations require the use of paths.

Paths operate in a uniform manner with respect to the program. Data transferred through paths are made to conform to the hardware requirements of the specific I/O device involved. Data transfers can be bidirectional (read and write) or unidirectional (read-only or write-only). The type depends on the device and the manner in which the path was established.

Data is transferred through a path in a stream of bytes with no specific type or value. The data's representation depends on its application within the program.

Some advantages of unified input/output systems are:

- Programs operate the same regardless of the type of I/O being used.
- Programs are portable from one system to another.
- I/O data can be redirected to alternate files or devices without changing the program.
- New driver routines can be created via the MAKDESC command.

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

I/O DEVICE NAMES

Each input/output device supported by the MiniCOP has its own device descriptor. I/O device descriptors are kept by OS-9 in the module directory. New device descriptors can be developed through the **MAKDESC** command. Table 16 lists each prebuilt device descriptor and the I/O device to which it pertains.

TABLE 16 - I/O DEVICE NAMES (DESCRIPTORS)

Descriptor	I/O Device	Notes
D0	COPRAM	Provides access to COPRAM (Root Directory)
T1	Port 1	ASYNC 25-pin serial
T2	Port 2	ASYNC 25-pin serial
T3	Port 3	ASYNC 25-pin serial
T4	Port 4	ASYNC 25-pin serial
TP	Tape Port	Terminal Emulator (on port 2)
V4	Video Ports	4 BNC connectors (RGB Color, Monochrome)
K4	Keybd Port	Modular keyboard jack

NOTE

Device Descriptors V4 and K4 are not included in the module directory after initialization using the OS-9 System Commands disk file 1003SYS. If needed in a particular application they can be installed using the conf1003 utility, which is located in the CMDS (Execution) Directory along with V4 and K4. Refer to *Using conf1003 to Load the V4 and K4 Device Descriptors* in *Appendix J.*

Using the MAKDESC command you can **copy** a currently defined descriptor from memory. The descriptor's associated parameters can then be changed to fit the specific application desired. The new descriptor is given a unique name and copied back into memory. The original descriptor will not be affected by this procedure and will still reside in memory.

RECEIVING INPUT FROM AN I/O DEVICE

When instructing the SHELL to receive input from an I/O device the descriptor must be the first item in the pathlist after the command. It must also be preceded by a slash (/) to create the device descriptor name. The slash indicates that the name is that of an I/O device. Take for example the following command line.

OS9:COPY /TP chochipcookie /D0/RECIPES/chochipcookie

This command line instructs the system to copy a file from the tape port (terminal emulator) called **chochipcookie**. It is then copied to a COPRAM directory called **RECIPES**. In this example two descriptors or I/O device names were used: **/TP** and **/D0**. TP indicated that the descriptor for exchanging data between the Terminal Emulator and the MiniCOP's tape port (port 2) would be used to receive the named file. D0 indicates that the descriptor for exchanging data with COPRAM will be used.

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

SENDING OUTPUT TO AN I/O DEVICE

By default most of the OS-9 commands and programs will display their output on the Terminal Emulator monitor. The output of these programs can alternately be sent to devices connected to any of the MiniCOP's ports. This technique is known as I/O redirection. For example, the DIR command sends a list of files to the Terminal Emulator monitor by default and no descriptor (I/O device name) is required. However, this list could also be sent to another device such as a serial printer connected to port 3 as is shown in the following example:

```
OS9:DIR /D0/RECIPES >T3
```

This command line instructs the system to list the contents of the COPRAM directory **RECIPES**. The greater than character (>) tells the SHELL to redirect this output to port 3 using the descriptor (I/O device name) **T3**.

NOTE

Refer to *I/O Redirection Modifiers* for additional information regarding the I/O redirection.

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

EXECUTION MODIFIERS

Execution modifiers are processed by the SHELL before a program is run. If an error is detected in any of the modifiers, the run will be aborted and the error reported. Modifier characters are stripped from the command line and passed to the program parameters. The characters reserved for use as modifiers (#, >, <, &) can be used before or after the parameters, but cannot be used inside the parameters.

ALTERNATE MEMORY-SIZE MODIFIER (#)

When programs are executed by the SHELL, they are allocated the minimum amount of working RAM memory specified in the program's module header. It may be desirable to increase this default memory size. Memory can be assigned using the # modifier in two methods.

Method 1 (#n)

In this method, memory is assigned in 256-byte pages. The modifier **#n,** where n is the decimal number of pages, directs the SHELL to assign the amount of memory specified. The following examples illustrate method 1:

OS9:COPY #8 /D0/RECIPES/chochipcookie /D0/TEMP/chocchipcookie

OS9:BASIC09 #16

In the first example, the file **chochipcookie** is copied from the directory **RECIPES** to the directory **TEMP** and is allocated **2048 bytes** of memory (8 x 256 = 2048). In the second example, the program **BASIC09** is called for execution and is allocated **4096 bytes** of workspace memory (16 x 256 = 4096).

Method 2 (#nK)

In this method, memory is assigned in increments of four pages or 1024 bytes. The modifier **#nK**, where **K** specifies 1024 byte blocks and **n** is the number of blocks, directs the SHELL to assign the amount of memory specified. The following examples illustrate method 2:

OS9:COPY #8K /D0/RECIPES/chochipcookie /D0/TEMP/chocchipcookie

OS9:BASIC09 #16K

In the first example, the file **chochipcookie** is copied from the directory **RECIPES** to the directory **TEMP** and is allocated **8192 bytes** of memory (8 x 1024 = 8192). In the second example, the program **BASIC09** is called for execution and is allocated **16384 bytes** of workspace memory (16 x 1024 = 16384).

In either method, the amount of allocated memory specified by the modifier must be greater than the program's default memory.

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

I/O REDIRECTION MODIFIERS (<, >, >>)

These modifiers are used to redirect the program's standard I/O paths to alternate files or devices. OS-9 programs may use these paths for routine I/O. Because programs do not use specific file or device names, it is easy to redirect the I/O to any file or device without altering the program itself. Programs that normally receive data from a terminal use one or more of the standard paths. There are three standard I/O paths: input, output, and error output.

STANDARD INPUT PATH	Normally passes data from the Terminal Emulator's keyboard to the program.
STANDARD OUTPUT PATH	Normally used to output data from the program to the Terminal Emulator's monitor.
STANDARD ERROR OUTPUT PATH	Used to output routine status messages such as prompts and errors to the Terminal Emulator's monitor (defaults to the same path as the standard output path).

When new processes are created, they inherit the standard I/O paths of the current process. The SHELL's standard input path is the Terminal Emulator keyboard and the standard output and error output paths are the Terminal Emulator monitor. When a redirection modifier is used on the command line, the SHELL opens the corresponding paths and passes them to the new process as its standard I/O paths. There are three redirection modifiers, which are defined in the following:

< Redirect the **STANDARD INPUT PATH** to the specified path.

> Redirect the **STANDARD OUTPUT PATH** to the specified path.

>> Redirect the **STANDARD ERROR OUTPUT** path to the specified path.

Redirection modifiers must be immediately followed by a pathlist describing the file or device that constitutes the new path. Files referenced by I/O modifiers are automatically opened or created, and closed (as appropriate) by the SHELL. Redirection modifiers can be used before and/or after the program's parameters, but each modifier can be used only once.

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

The following examples and explanations illustrate the use of redirection modifiers.

OS9:LIST opsmessage>/T1

OS9:DIR /D0/DATA>/D0/SYSCONFIG/datadirconfig

In the first example, assuming that **opsmessage** is a text file, its listing will be redirected to the device at port 1. In the second example the listing of the directory **DATA** is redirected to the directory **SYSCONFIG** and placed into the file **datadirconfig**.

If the LIST command, as shown in the following example, is used on the file **datadirconfig** from the directory **SYSCONFIG** (which was created in example 2) the output from the DIR command will be displayed as illustrated in Figure 51.

OS9:LIST /D0/SYSCONFIG/datadirconfig

```
directory of ./D0/SYSCONFIG    07:5:41
datadirconfig      execdirconfig      mdirconfig
```

*All files and/or directories contained in SYSCONFIG will be listed.
"execdirconfig" and "mdirconfig" are examples of additional files.

FIGURE 51 - LISTING A FILE CREATED VIA A REDIRECTION MODIFIER

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

SEQUENTIAL EXECUTION MODIFIER (;)

Sequential execution of commands means that one command cannot be executed until the previous command's task is complete. Commands are run sequentially when each is entered on a separate command line. Sequential execution of multiple commands can be specified on a single SHELL command line by separating each (along with any parameters) from the next with a semicolon. For example:

```
OS9:COPY /D0/TEMP/startup /D0/APPLICS/startup; DIR /D0/APPLICS>T2
```

This command line will first execute the COPY of the file **startup** from the directory **TEMP** to the directory **APPLICS**. When this finishes, the DIR command provides a listing of the **APPLICS** directory that is redirected to **T2**.

If an error is returned by any program or command, subsequent commands on the same command line are not executed (regardless of the state of the built-in X option). This is not the case for commands on separate lines.

All programs executed sequentially are separate child processes of the SHELL. After initiating execution of a program to be executed sequentially, the SHELL enters a "wait" state until execution of the called program terminates.

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

MULTITASKING MODIFIER (&)

The multitasking modifier (&) implies concurrent execution of commands. The first command is run, but the SHELL does not wait for it to complete before processing the next command.

Two or more programs or commands can be executed simultaneously. The number of programs that can be run at the same time is not fixed. It depends on the amount of free memory in the system and the memory requirements of each. The following examples illustrate the use of the multitasking modifier:

```
OS9:DIR>T4&
&003
```

In this example the DIR command causes the SHELL to send a directory listing to the device at the MiniCOPs port 4. The multitasking modifier (&) on the command line causes the SHELL to wait for another command to be executed. The next command can be entered and executed while the directory is being printed. The ID number of this process (the DIR command), &003, is displayed below the command line. The PROC command can be used to display all of the current processes in the system.

```
OS9:(DEL file1; DEL file2; DEL file3)&
&004
OS9:LIST file4 /T3
```

In this example (from *Command Grouping*), command grouping is being used to execute the three files (**file1**, **file2**, **file3**) sequentially. The multitasking modifier allows this sequential deletion of files to be executed concurrently with the listing of **file4** to **T3**.

```
OS9:COPY /D0/DATA/old new & LIST errmessages >T2;DEL /D0/DATA/old
```

In this example, the COPY command copies a file called **old** from the directory **DATA** and creates a new file called **new** in the same directory. The LIST command sends a copy of the file **errmessages** to the device at port 2 (**T2**). The DEL command deletes the file **old** from the directory **DATA**. Because they are separated by the multitasking modifier (&), COPY and LIST will execute concurrently. The DEL command, because it is preceded by a semicolon, will not run until the last of the two concurrent commands have completed execution.

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

WORKING IN COPRAM

COPRAM DIRECTORY STRUCTURE

COPRAM stores data in separate logical entities called files. Each file has a name that is entered under a directory. COPRAM has a root directory, D0, that includes the names of files, directories and subdirectories. Directories are a convenient way to organize information by file type, application, or other appropriate scheme, convention, or categorization desired.

The root directory is created automatically when the MiniCOP is first initialized. After being initialized it contains the root directory, /D0, and a directory called CMDS (pathlist = /D0/CMDS). Also in the root directory is the file: DATACONTROL. When working with a 627-1003 MiniCOP files, VideoDemo and conf1003 also appear. In both cases, the path lists are constructed as follows; pathlist = /D0/<file name>. You may create additional directories as needed. In addition, it is possible to delete or rename any directory.

COPRAM pathlists locate a file within the directory structure. When OS-9 is given a pathlist to process, it reads the names sequentially from left to right, searching through the various directory branches to obtain the necessary routing information.

The root directory is accessed by the SHELL using the device descriptor, D0. This is also where the root gets its name. Figure 52 illustrates a simplified example of a COPRAM directory structure that includes its relationship/position with the SYSTEM DEVICE DIRECTORY (the OS-9 Operating System Layer 4).

Notice that the device descriptor and directory names are capitalized and the file names are not. This is a recommended practice (not mandatory) that allows directories and files to be easily identified using the DIR command.

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

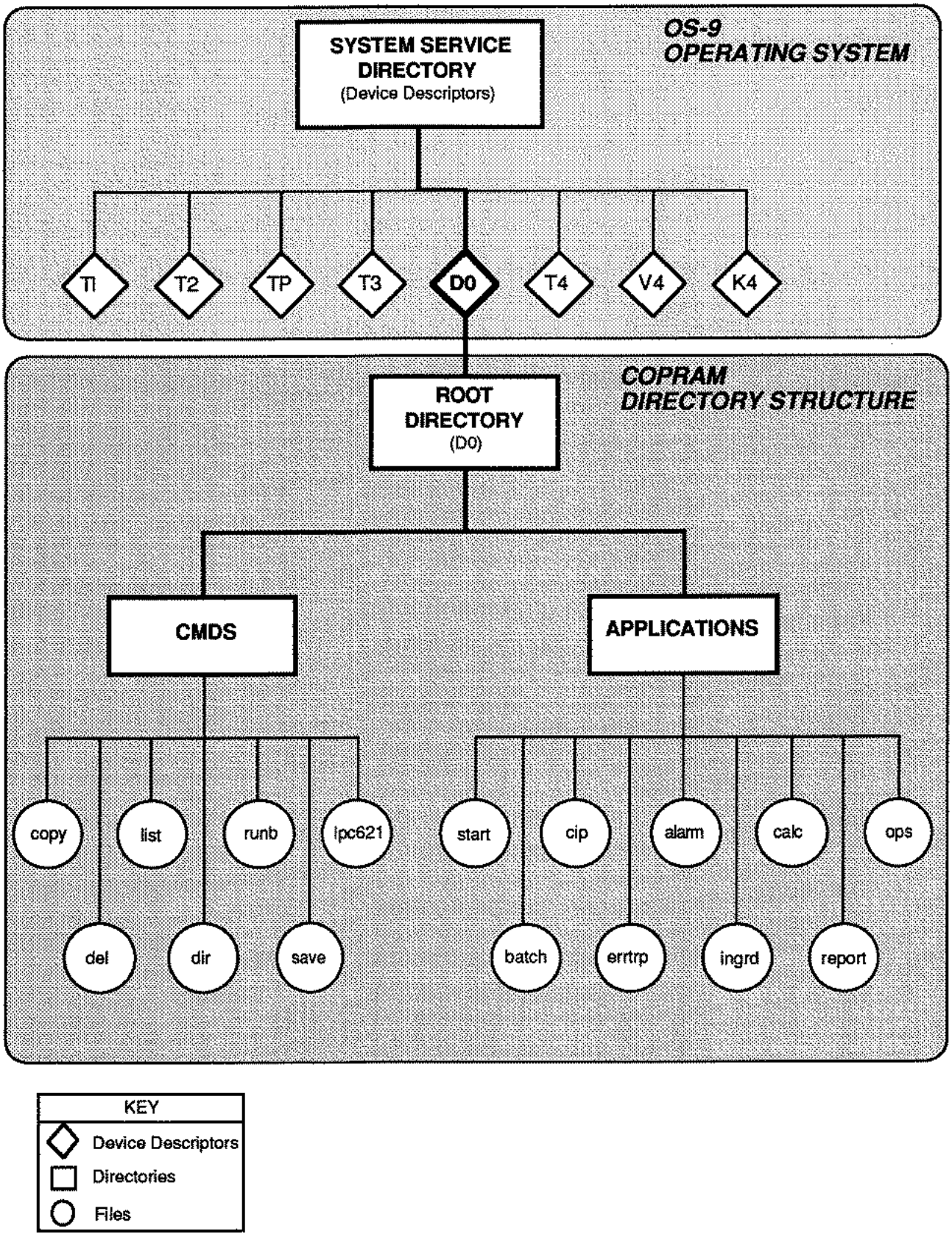


FIGURE 52 - COPRAM RELATIONSHIP TO THE SYSTEM SERVICE DIRECTORY

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

ADDITIONAL INFORMATION ABOUT COPRAM DIRECTORIES

The COPRAM directory system is very useful because it allows a user to privately organize files as desired (project, function, etc.) without affecting other files or other users' files. Other important characteristics relating to the use of COPRAM directory files include:

- Directories are files and have the same ownership, security attributes, and rules as nondirectory files.
- A file name can only appear once under a specific directory, but it may appear in other directories.
- A file can only acquire the directory attribute when it is created with the **MAKDIR** command.

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

DATA AND EXECUTION DIRECTORIES

There are two directory pointers in OS-9, which allows for two working directories referred to as the current data and execution directories. Two working directories are maintained so that executable program modules can be organized in a different directory than the files containing data.

CAUTION

Though it is possible to create more than two directories in COPRAM, extra care must be taken when developing processes. If a pathlist is not explicitly given, OS-9 will default to the current working directories regardless of the actual intention of the user. This could cause unintended results possibly creating a disastrous situation. Use the **PD d** and **PD x** commands to determine the current directories.

Data Directory

Immediately after the MiniCOP is initialized, OS-9 sets the current directory pointer to the root directory, **/D0**. Unlike the execution directory, a unique data directory is not present until created by the user.

Typically, the data directory will contain text files and BASIC09 source code procedures.

A listing of the root directory (current data directory after initialization) contents is illustrated in Figure 53.

```
directory of .  14:37:52

CMDS  DATACONTROL  VideoDemo  conf1003
```

Italicized file names are available for use with the 627-1003 MiniCOP when initialized using the file 1003SYS.

FIGURE 53 - ROOT DIRECTORY (/D0) LISTING

The current data directory can be changed by using the **CHD** (change data directory) command and specifying the desired directory via its pathlist. The pathlist of the current data directory can also be displayed using the PD d(print current directory - data) command.

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

Execution Directory

OS-9 creates an execution directory, called CMDS, during the MiniCOP's initialization. The CMDS directory is the COPRAM location where the OS-9 system commands are transferred from the Terminal Emulator.

Other than the OS-9 system commands, the execution directory is typically used to store executable OS-9 program modules and packed BASIC09 procedures.

A listing of the program modules (OS-9 system commands), loaded to the CMDS directory during initialization, is illustrated in Figure 54.

```
directory of .   14:39:40

Attr          Binex         Build         Clear
Copy          Date          Deldir        Del
Display       Dir           DskBak        Dump
Exbin         Free          Ident         Inkey
Ipc621        Link          List          Load
MakDesc       Makdir        Mdir          Merge
Mfree         Procs         Pd            Rename
Reset         Save          Setime        Setport
Shell         Sleep         Tmode         Unlink
Verify        Xmode         Basic09       RunB
DataControl   VideoDemo     TV            K4
V4
```

Italicized file names are available for use with the 627-1003 MiniCOP when initialized using the file 1003SYS.

FIGURE 54 - CMDS DIRECTORY LISTING

The current execution directory can be changed using the **CHX** (change execution directory) command and specifying the desired directory via its pathlist. The pathlist of the current execution directory can also be displayed using the PD x (print current directory - execution) command.

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

CREATING DIRECTORIES

The D0 and CMDS directories are created by OS-9 during initialization. Originally D0 serves as both the root and current data directories. CMDS serves as the current execution directory. It is not recommended that the root directory be used as the user's data directory. Therefore, at least one new directory should be created for this purpose.

Directories are created by first creating a directory file. The directory file is a special type of COPRAM file in which the directory attribute has been set ON. The directory file is organized into 32-byte records. With few exceptions they are processed by the same I/O functions and commands as any other file.

Assume that COPRAM currently contains the initial directory structure, as shown in Figure 55.

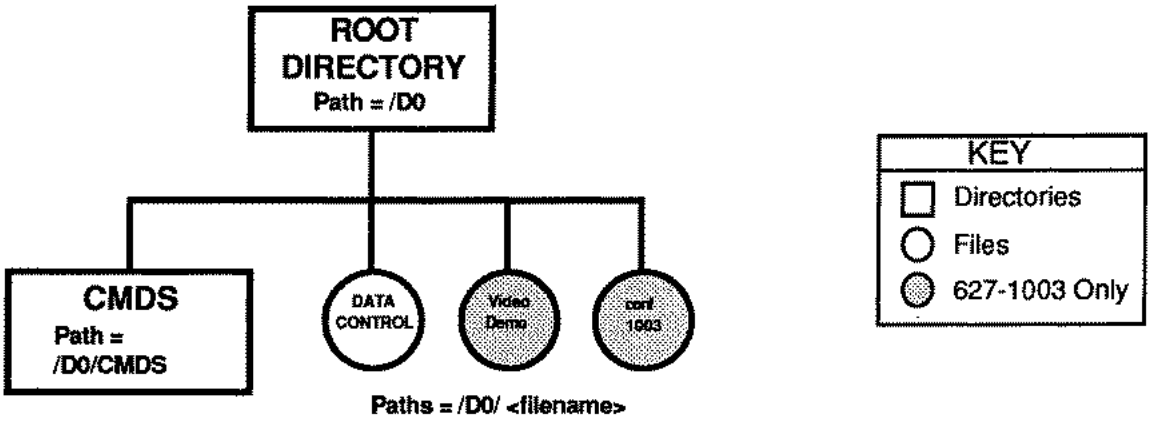


FIGURE 55 - COPRAM DIRECTORY STRUCTURE AFTER INITIALIZATION

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

Directory files are created with the command, which must exist in the current execution directory. A new directory is always created from the current data directory, unless otherwise directed with a pathlist. In the following example a new directory, called **DATADIR**, is created.

OS9:MAKDIR DATADIR

This command line will create the new directory under the current data directory, **D0**. The COPRAM directory structure will now appear as shown in Figure 56.

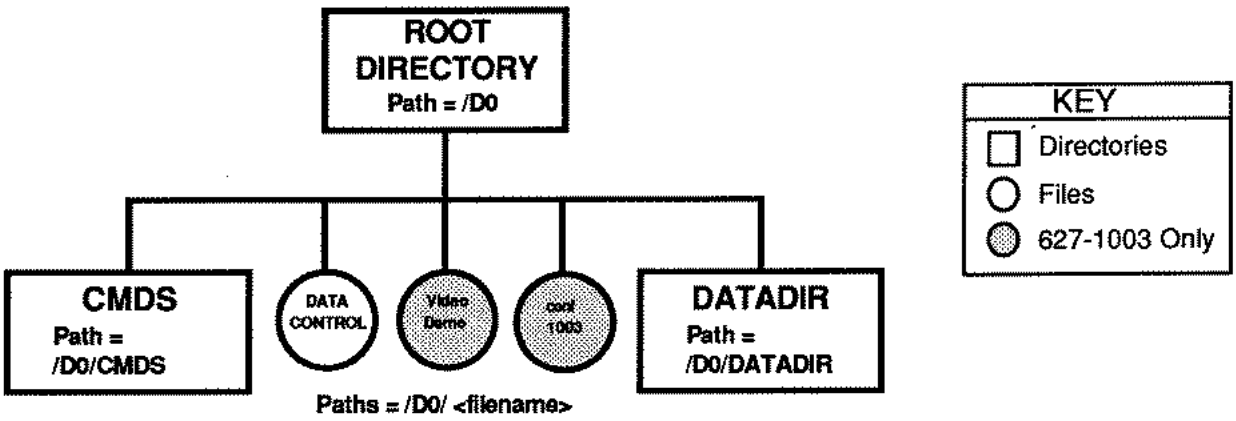


FIGURE 56 - DIRECTORY STRUCTURE AFTER CREATING DATADIR

The current data directory is still the root directory (**D0**). Any newly-created directory can be changed to the current directory (data or execution) using the **CHD** or **CHX** commands.

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

Subdirectories can also be created. Suppose that under the directory **DATADIR**, a subdirectory called **SOURCE** is desired. This new directory can be created in two basic manners as shown in the examples:

```
OS9:CHD /D0/DATADIR
OS9:MAKDIR SOURCE
```

or

```
OS9:MAKDIR /DO/DATADIR/SOURCE
```

In the first example, the current data directory was changed from **D0** to **DATADIR**. Then a new directory was created, called **SOURCE**. In the second example, the new directory, **SOURCE**, was created in the directory, **DATADIR**, from the current directory, **D0**. By including the pathlist, the specific location where the new directory was to be created was identified. In the case of either example, the COPRAM directory structure now appears as shown in Figure 57.

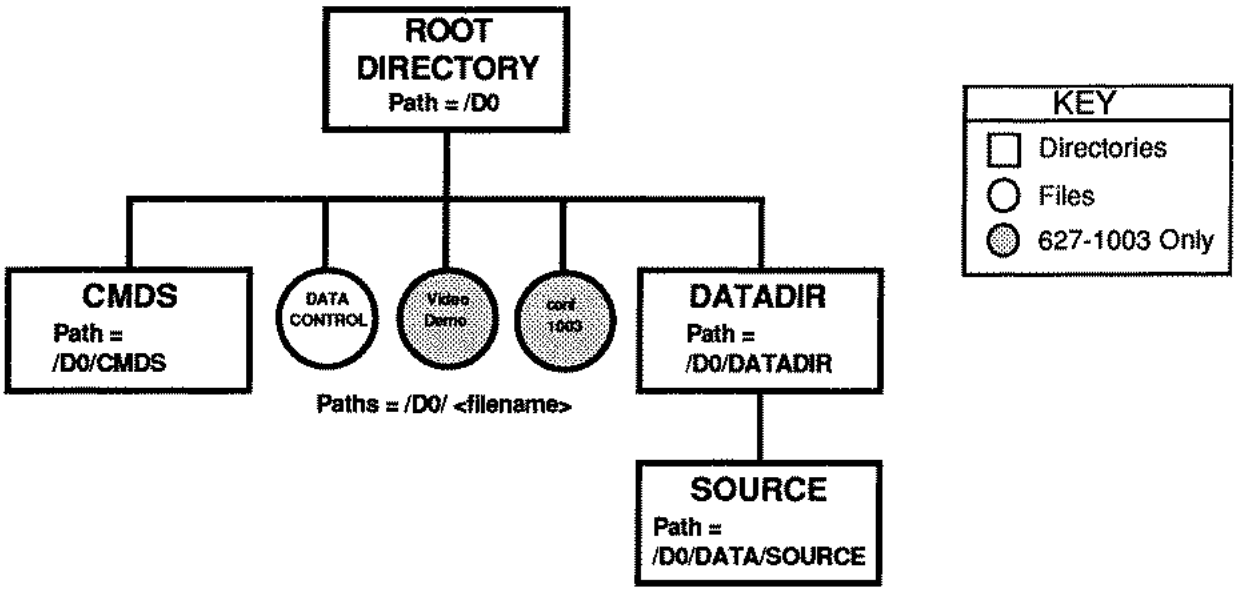


FIGURE 57 - DIRECTORY STRUCTURE AFTER CREATING SOURCE

Directories can be created in this way, and files can be created in or copied to them. Directory files cannot be copied or listed, the **DIR** command is used instead.

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

DELETING DIRECTORIES

The **DEL** command is used to delete a file within a directory, but it cannot be used to delete the directory file itself. There are two methods that can be used to delete a directory file.

Method 1

The first method is to use the DELDIR command. DELDIR is a convenient and fast way to delete a directory file.

It is important to note that the DELDIR command will also delete all files and subdirectories in the specified directory file. When using the DELDIR command a message is displayed requesting you to choose from three options: **LIST** the directory, **DELETE** the directory, or **QUIT** the deletion operation, as shown in Figure 58.

```
Deleting directory file.
List directory, delete directory, or quit? (l/d/q)
```

FIGURE 58 - DIRECTORY SELECTION OPTIONS

Selecting **q** (quit) aborts the delete operation and generates the BASIC09 error message #001 indicating an unconditional abort has occurred.

Selecting **d** (delete) causes an unconditional deletion of the directory and all files (including subdirectories and their files) it holds. As each file is deleted its attribute list is displayed.

Selecting **l** (list) lists the contents of the specified directory including each file's: name, bytecount, sector, attributes, last modified date, and owner. A message is then displayed asking if you wish to: **Continue?, (y/n)**. Selecting **y** (yes), initiates the delete function, while selecting **n** (no), initiates the quit or abort function.

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

Method 2

This method is a little more involved, but achieves the same results. It can be broken down into the following steps:

1. Delete all files in the directory using the **DEL** command. It may be necessary to do this in any subdirectories the specified directory contains.

2. Using the **ATTR** (attribute) command, turn OFF the specified directory's "directory attribute".

3. Delete the file using the **DEL** command.

While method 1 is the quickest it is also the most drastic and easiest in which to delete files or subdirectories that were not intended to be deleted. Also, it may not be possible to use the DELDIR command depending on the MiniCOP's application. Often, commands in the execution directory that are not called on by any of the programs, which are used to control its application, are deleted from the CMDS directory. This technique is used to free up valuable COPRAM memory for more useful purposes. In this case method 2 can be used or the DELDIR command can be copied to the MiniCOP from the OS-9 system commands disk via the Terminal Emulator. Once finished, the command can again be deleted.

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

USING AND CHANGING DIRECTORIES

The use of current working directories allows pathlist construction to start at any level (subtree) of the COPRAM's hierarchy structure. Any directory that the user has permission to access can be made a current working directory.

The two most basic rules for pathlists are:

1. When the first character in the pathlist **is not a slash** (/), processing of the pathlist starts at the COPRAM's current working directories.

2. When the first character **is a slash** (/), processing of the pathlist starts at the system service directory (device descriptors).

   Pathlists starting with a slash (/) must be complete. They must contain all names required to trace the pathlist from the system service directory down through all levels of COPRAM directories (if any).

Use of the current working directory allows all names in its own hierarchy branch to be implied instead of explicitly given. This not only makes pathlists shorter, but typically allows OS-9 to search fewer directories. Assume that COPRAM is structured as shown in Figure 59 and that the current working data directory is **RECIPES**.

The **RECIPES** directory pathlist is **/D0/APPLICATIONS/RECIPES**. If the name product1 is entered at the command line, its path is understood. However, to call the name **rprtgen** without changing the directory pointer the entire pathlist must be provided — for example (**/D0/APPLICATIONS/UTILS/rprtgen**).

Pathlists that use the working directories as their starting point can also specify additional lower-level directories. Assuming that **RECIPES** is still the current working directory, when entering the pathlist, **TESTPRODS/p2test**, the full pathlist of **/D0/APPLICATIONS/RECIPES/TESTPRODS/p2test** is understood.

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

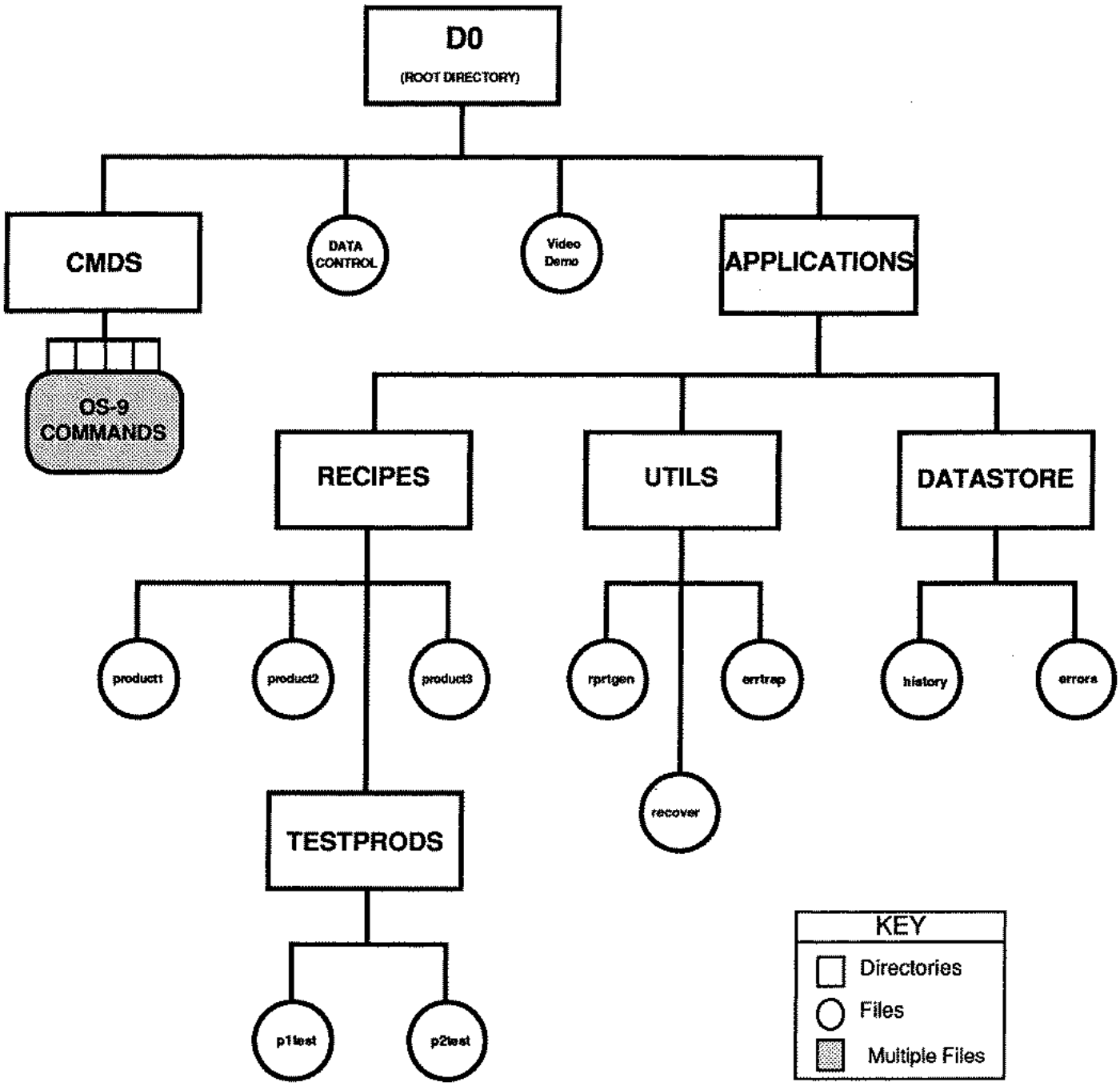


FIGURE 59 - USING AND CHANGING DIRECTORIES

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

The built-in SHELL commands **CHD** and **CHX** are used to change the current working data and execution directories, respectively. The commands must be followed by a pathlist that describes the new directory's location. Permission to access the directory file according to normal file security is also required. The following command line examples illustrate the use of the **CHD** and **CHX** commands and reference Figure 59. Assume that the current working data directory is **D0** and the current working execution directory is **CMDS**.

OS9:CHD APPLICATIONS/DATASTORE

In this example, the current working data directory is changed from **D0** to **DATASTORE**. Notice that the pathlist does not include the root directory name, **D0**. Because **D0** is the current data directory and the first name in the pathlist, **APPLICATIONS**, can be found, the pathlist prior to **APPLICATIONS** is understood. Also notice that the name **APPLICATIONS** is not preceded by a slash (/). A slash tells the SHELL to begin processing the following pathlist from the system service directory since the name **APPLICATIONS** is not located there; error message #221 - Module Not Found, would be displayed.

OS9:CHD /D0/APPLICATIONS/UTILS

In this example, the current working data directory is changed from **DATASTORE** to **UTILS**. Because the **UTILS** directory does not reside below the **DATASTORE** directory, it is necessary to specify the complete pathlist beginning at the root directory (**D0**).

OS9:CHX /D0/APPLICATIONS/UTILS

In this example, the current working execution directory is changed from CMDS to UTILS. Because the **UTILS** directory does not reside below the **CMDS** directory it is necessary to specify the complete pathlist beginning at the root directory (**D0**).

CAUTION

Care must be taken when changing the current working execution directory.

In order for the system to execute any OS-9 commands (except built-ins), they **must be present** in the current working execution directory. Changing the execution directory will limit the available OS-9 commands to **only those that were previously copied** into the new execution directory.

Because the CHX command is a built-in, it is possible to recover should you inadvertently change the execution directory to one that has no or only some of the OS-9 commands.

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

ANONYMOUS DIRECTORY NAMES

Sometimes it is useful to be able to refer to the current directory pointer or the next higher-level directory, but the entire pathlist might not be known or might be lengthy. Special name substitutes can be used in these cases. The name substitute is the period (.). A single period refers to the current directory. Each additional period refers to the next higher-level in that particular branch of the directory tree structure. They can be used in the place of pathlists or they can be the first name in a pathlist.

The following examples illustrate the period used as a name substitute in command line entries. Reference Figure 59 and assume that the current working data directory is **TESTPRODS**.

OS9:DIR . -- In this example, all file names in the current data directory will be displayed (DIR defaults to the data directory).

OS9:DIR .. -- In this example, all file names in the **RECIPES** directory will be displayed.

OS9:DIR ... -- In this example, all files in the **APPLICATIONS** directory will be displayed.

OS9:DEL .../VideoDemo -- In this example, the file, **VideoDemo**, in the **D0** directory will be deleted.

OS9:CHD .. -- In this example, the current data directory will be changed from **TESTPRODS** to **RECIPES**.

OS9:CHD .../DATASTORE -- In this example, the current directory will be changed from **RECIPES** to **DATASTORE**.

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

WORKING WITH OS-9 FILES

FILE USAGE IN OS-9

There is physically only one type of file in OS-9, but the use of files covers a broad spectrum. OS-9 does not use file extensions. Because all files have the same physical type, commands such as COPY, DEL, etc. can be used on most any file. Directory files are an exception to this rule and they utilize several special commands (e.g. DELDIR, MAKDIR) of their own.

Since all files are of only the one type, a particular file can be treated as having a different use at different times by different programs. Files can be used as:

- TEXT FILES,
- RANDOM ACCESS FILES,
- EXECUTABLE PROGRAM MODULE FILES,
- DIRECTORY FILES, and
- PROCEDURE FILES.

The directories are files that have had the directory attribute turned ON. Each directory file contains the names and logical sector numbers of the files and subdirectories they hold.

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

NAMING OS-9 FILES

Use the following three simple rules to name any OS-9 file:

1. Any OS-9 file (including directories and BASIC09 procedures) can consist of up to twenty-nine characters.

2. The first character must always be an alphabetic character.

3. The remaining characters can be alphabetic, numeric, or the period symbol. Special characters, other than the period, are reserved and cannot be used in file names.

It is recommended (not a requirement) that directory files use all upper-case alphabetic characters and that all other files use lower-case alphabetic characters. This naming convention will simplify the identification of files and directories when listed with the DIR command.

The following list provides some examples of legal file names:

raw.data2
QOil.Temp.History
project.backup
X042953
DATAFILE15

The following list and explanations provide some examples of illegal file names:

22November -- does not begin with an alphabetic character.
Max*Min -- uses the reserved special character * (asterisk).
.data -- does not begin with an alphabetic character.
data file -- uses the space character, which is a reserved special character.
plastic.extrusion machine.output.linearfoot.pershift -- contains more than twenty-nine characters.

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

PHYSICAL FILE ORGANIZATION

Each OS-9 file is composed of one or more sectors that are the physical storage units of COPRAM. COPRAM uses sectors that store exactly 256 bytes of data. Each sector is numbered sequentially starting at logical sector number zero (LSN 0). Each file has a directory entry that includes the file's: name; byte count; LSN; attributes; last date modified; and owner.

Unless the file's byte count is zero, the file will have one or two sectors used to store data. The data sectors are grouped in one or more contiguous blocks called segments.

FILE READING AND WRITING

A single file type and format is used for all COPRAM files. Files store an ordered sequence of 8-bit bytes. A given file may store a machine language program, assembly language program, a BASIC09 program, or characters of text. Data is written to and read from files exactly as given. The file can be any size from zero bytes up to the COPRAM memory capacity of 256K bytes.

When a file is first created a file pointer is established for it. Bytes within the file are addressed like memory, and a file pointer holds the address of the next byte in the file to be written or read. READ and WRITE operations update the POINTER as data transfers are performed.

Any part of a file can also be read from or written to nonsequentially by using a function called SEEK to reposition the file pointer to any byte address in the file. This is used when random access of the data is desired.

EXECUTABLE PROGRAM MODULE FILES

These files are used to hold packed BASIC09 procedures or program modules that act as OS-9 utilities. Each file may contain one or more program modules. OS-9 program modules resident in memory have a standard format that includes object code, a module header, and a CRC value. If a program module has been altered in any way, either as a file or in memory, its CRC value will be incorrect and OS-9 will refuse to load the module.

Program modules and packed BASIC09 procedures cannot be listed. The DUMP command can be used to examine the contents of this kind of file in either a hexadecimal or ASCII format.

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

RANDOM-ACCESS DATA FILES

Random-access data files are created from and used within BASIC09. The BASIC09 GET, PUT, and SEEK functions are used to operate on random-access files. The file is organized in an ordered sequence of records. Each record has exactly the same length, so given its numerical index, a record's beginning address within the file can be computed by multiplying the record number by the number of bytes used in each record. This permits records to be directly accessed in any order.

Each record can be subdivided into fields as defined by the TYPE statement in BASIC09. Each field generally has fixed length and usage for all records within the file.

TEXT FILES

These files contain lines of ASCII characters. Each line is terminated by a [RETURN] character. Text files are used for program source code, procedure files, messages, documentation, etc.

Text files are created with the BUILD command. The file can be created in any COPRAM directory, according to the pathlist given in the command line. If no pathlist is provided, the file will be created in the current data directory by default.

Text lines can be up to eighty characters in length and the only limit to the number of lines is the amount of available COPRAM memory.

Text files are usually read sequentially, and are supported by BASIC09 READ and WRITE statements. Though it is possible to randomly access data at any location within a text file, this is rarely done because each line has a variable length. This makes it difficult to locate the beginning of a line without actually reading the data to locate the [RETURN] characters.

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

SHELL PROCEDURE FILES

The SHELL is a reentrant program that can be simultaneously executed by more than one process at a time. It uses the standard I/O paths for routine input and output. Specifically, it requests command lines from the standard input path and writes prompts and other data to the standard output path, unless otherwise directed.

The SHELL can accept and execute command lines from a file instead of the Terminal Emulator's keyboard. The text file to be processed is called a procedure file.

Procedure files contain one or more command lines that are identical to command lines that would be manually entered from the keyboard. This technique is sometimes called batch or background processing.

If the program module specified on the SHELL command line cannot be found in main memory or the current execution directory, the SHELL searches the current data directory. If found, the SHELL automatically executes it as a procedure file. A procedure file may look something like this:

```
LIST file.xfer.message
LOAD COPY
COPY new.data old.data
DEL new.data
UNLINK COPY
```

In this procedure file, the LIST command displays the contents of the file, **file.xfer.message**, to the Terminal Emulator monitor. The LOAD command loads the command (program module) COPY from COPRAM into main memory's module directory. COPY is then used to transfer the contents of the file **new.data** to the file **old.data**. DEL deletes the file **new.data**. UNLINK removes COPY from the module directory.

A procedure file can have a number of valuable applications. It can eliminate repetitive manual entry of commonly-used sequences of commands. It can allow the MiniCOP to execute a lengthy series of programs in the background while the MiniCOP is unattended or while the system is running other programs in the foreground.

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

In addition to redirecting the SHELL's standard output and input to a procedure file, the standard output and error output can be redirected to another file that can record output for later review or printing. This can also eliminate the sometimes annoying output of SHELL messages to the Terminal Emulator monitor on random intervals. An example of this type of procedure file may look like:

```
MERGE data1 data2 >comfile
LIST comfile >/D0/SOURCE/listing
DIR /D0/SOURCE/ >T2
```

In this example, MERGE combines the contents of the files **data1** and **data2** and sends this information to a file called, **comfile**. The LIST command transfers the lines of text from **comfile** to a file called **listing**, which is in the directory **SOURCE**. The DIR command sends a directory listing of all files and subdirectories contained in **SOURCE** to a device, such as a printer, connected to the MiniCOP's port 2.

There are two basic methods to execute a procedure file. Assume a procedure file exists called **procfile**. Either of the two following examples will execute this file.

```
OS9:SHELL procfile
```

or

```
OS9:procfile
```

Both will call and execute the procedure file called **procfile**. The only difference is that the first example creates a new SHELL which is used to execute the file. The second example will execute the file in the current SHELL. The command line in the first example could be used to call the SHELL and execute the named file from a BASIC09 program.

The procedure file can also be executed in the background by adding the multitasking execution modifier (**&**). OS-9 has no constraints as to the number of jobs that can be executed concurrently, provided that sufficient memory is available.

The procedure files themselves can cause sequential or concurrent execution of additional procedure files. Following is an example of the multitasking of two procedure files.

```
OS9:proc1 T >>stat1 & proc2 T >>stat2
```

The error output of **proc1** is redirected to **stat1** and the error output of **proc2** if redirected to **stat2.** The built-in command T is used to direct the SHELL to record all input lines executed to the output file, but without the useless OS-9 prompts intermixed. The "-X" built-in command can be used to prevent the process from stopping if an error occurs. The built-in commands only affect the SHELL that executes them, and not other SHELLs that may also exist.

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

SYSTEM

Each file (including directory files) has ownership attributes that define which processes may access the file and whether it can be read, changed, or executed. OS-9 automatically stores a user number with each file associated with the process that created it. As the user, you are considered to be the owner of the file. For example, a file, such as a text file, constructed manually from the OS-9 command line with the BUILD command will have the same user number assigned to it as the person who constructed it.

If a file has been built by a process with the user number of 4 , then that process would "own" it and the newly built file would also carry the user number of 4. This new file could then not be executed by files with other user numbers.

There are a total of eight attributes, each of which can be turned ON or OFF independently. The first is the directory (**d**) attribute, which when ON indicates the file is a directory. The second is the shareable (**s**) attribute, which when ON, indicates that a file cannot be accessed by two or more tasks simultaneously.

Six other attributes control whether the file can be read from, written to, or executed. Three of these attributes refer to the file's use by the OWNER and the remaining three the PUBLIC. The three public attributes are not used by OS-9 files created for use with the MiniCOP.

NOTE

OS-9 was designed for use on multi-user systems, hence the use of public vs. owner permission attributes. The 627 MiniCOP is a single-user system. The attribute command (ATTR) is therefore used to protect the file from accidental changes by the owner. The public attributes are not used by OS-9 files in the MiniCOP.

The three owner attributes are:

EXECUTE PERMISSION TO OWNER

If ON, the owner can load the file into memory and execute it. The file must contain one or more valid OS-9 memory modules in order to actually load. For example, a text file that contains only a message and no OS-9 commands will generate an error. If OFF, any file can execute it.

WRITE PERMISSION TO OWNER

If ON, the owner may write to or delete the file. If OFF, any file can write to or delete it. This can be used to protect important files from accidental deletion or modification.

READ PERMISSION TO OWNER

If ON, the owner is allowed to read from the file. If OFF any file can read from it. This can be used to prevent binary files from being inadvertently used as text files.

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

EXAMINING AND CHANGING FILE ATTRIBUTES

The DIR e command may be used to examine the security permission of the files in a particular directory. An example of this operation is given below:

```
OS9:DIR e /D0/DATADIR
directory of /D0/DATADIR 10:04:36
```

Owner	Last Modified	Attributes	Sector	Bytecount	Name
0	89/12/23 1409	----r-wr	47	42	datafile
0	91/06/17 1011	----r-wr	51	22	errortable
0	90/09/17 0830	d-ewrewr	6D	800	SOURCE

Entries under the attribute column display those that are currently ON. Figure 60 illustrates the eight attributes and their definitions:

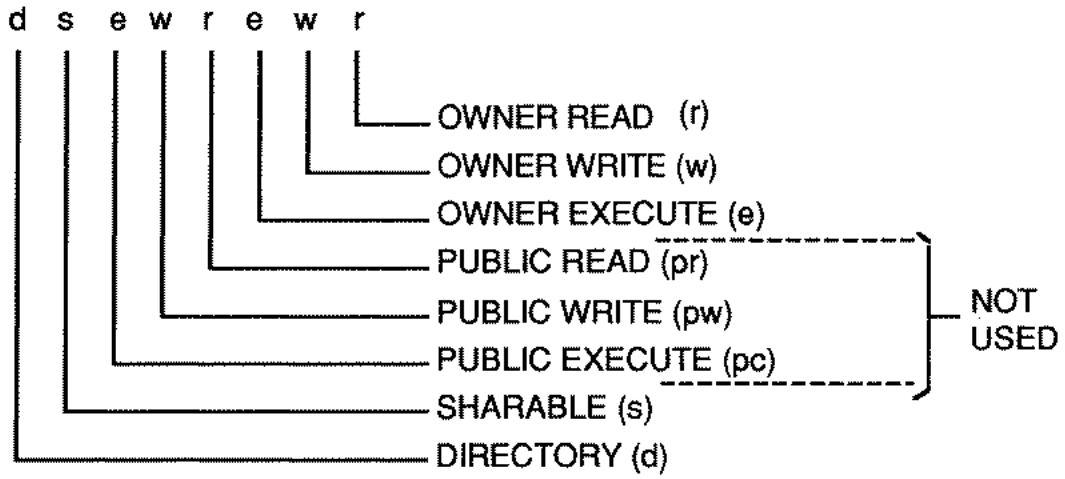


FIGURE 60 - FILE ATTRIBUTES

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

The ATTR (attributes) command is used to examine or change a file's attributes. Entering ATTR followed by a file and pathlist, if necessary, will display the file's attributes. The ATTR command can be used with a list of one or more attribute abbreviations to change a file's attributes. If the abbreviation is preceded by a minus sign, the attribute will be turned OFF. If it is preceded by no sign, it will be turned ON. The new attribute listing will be printed beneath the command line when executed. The following examples illustrate the use of the ATTR command in changing a file displaying or changing a file's attributes.

```
OS9:ATTR sampletest
------wr

OS9:ATTR sampletest -w
-------r

OS9:ATTR sampletest w
-------wr

OS9:ATTR sampletest -w -a
```

In the first example, The ATTR command was used to display the current file attributes of **sampletest**. These attributes will be displayed directly below the command line. In the next example, **-w**, is added to remove the owner write permission from this file. Next, the write permission was restored using the **w** attribute. In the last example the owner write permission was once again removed, but notice that in this case the new attributes were not displayed beneath the command line. The option **-a** was included in the command line to inhibit OS-9 from displaying the file's attributes after the ATTR command.

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

LOADING, LINKING, SAVING, AND UNLINKING MODULES

You can transfer program modules between the module directory, COPRAM, and the Terminal Emulator (floppy or hard disk). The LOAD, LINK, SAVE, and UNLINK commands are used to perform these operations. Figure 61 illustrates the transfer of program modules between these three areas and the commands responsible for each operation.

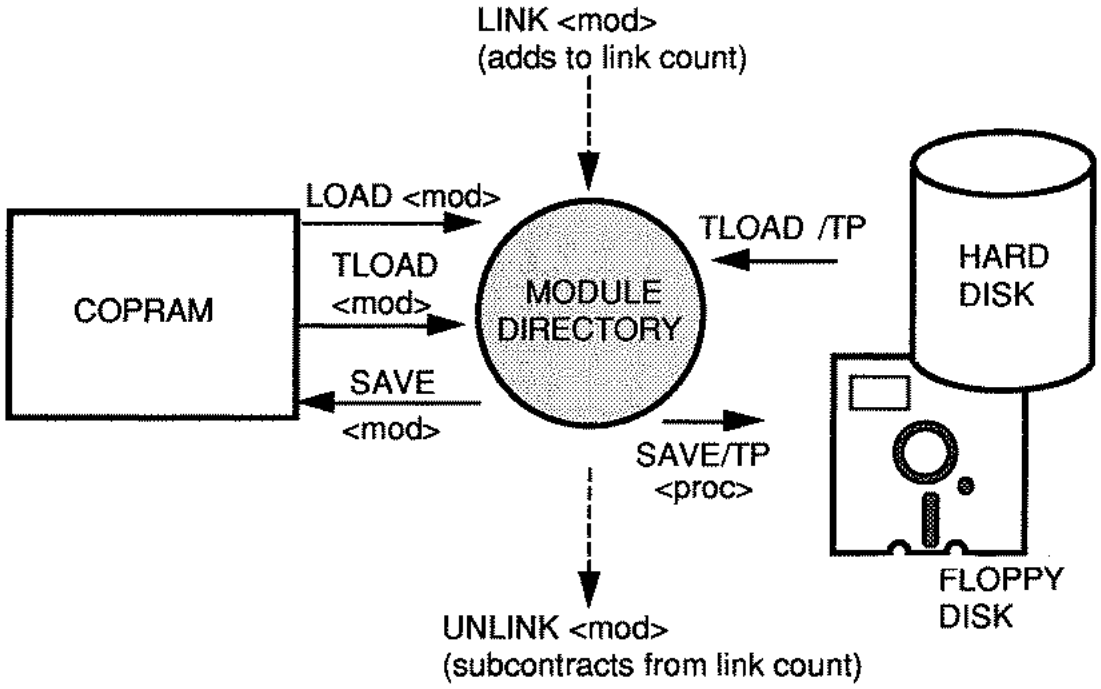


FIGURE 61 - PROGRAM MODULE TRANSFER

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

The LOAD command is used to transfer program modules from COPRAM to the module directory (in main memory) and "lock" them in. Only program modules and packed BASIC09 procedures can be loaded. If the module being loaded has the same name and is of the same type as the module already in memory, the module with the highest revision level (last modified) will be used or kept.

The following example illustrates a typical load command line.

OS9:LOAD /D0/UTILITIES/historize

In this example, the file **historize** is loaded in main memory from the COPRAM directory **UTILITIES**.

The OS-9 command LOAD should not be confused with the BASIC09 command of the same name; the two have totally different meanings.

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

The TLOAD command can be used to perform the same operation as LOAD. The advantage of using LOAD is that in loading multiple program modules on the same command line, all modules are linked into memory, while TLOAD will link only the first module listed. Both commands will load all named modules into memory.

TLOAD is also used to transfer program modules from floppy or hard disk on the Terminal Emulator to the module directory. In this use the command is followed by /TP to indicate that the file is to be received on the tape port via the Terminal Emulator.

The following two examples illustrate typical OS-9 command lines using the TLOAD command.

OS9:TLOAD batchmix

OS9:TLOAD /TP

In the first example, the file **batchmix** is loaded from COPRAM and placed into main memory. In the second example, a file is being loaded into main memory from the Terminal Emulator via the tape port. The specific file is selected using the Terminal Emulator's menu functions and is not identified on the command line. The COPY command can also be used to transfer files from the Terminal Emulator to the MiniCOP.

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

SAVE

The SAVE command is use to transfer modules from main memory to COPRAM or the floppy or hard disk on the Terminal Emulator. The following three examples illustrate typical OS-9 command lines using the SAVE command.

OS9:SAVE ATTR

OS9:SAVE /D0/SOURCE.DIR/mod4 mod4

OS9:SAVE /TP linestartup

In the first example, the module **ATTR** is saved from the module directory to the current COPRAM execution directory. Executable program modules should generally be saved to the execution directory. The current execution directory is the default for the save command, modules to be saved elsewhere must include the necessary pathlist.

In the second example, the file **mod4** is saved as **mod 4** to the **SOURCE.DIR** directory.

In the last example, the file **linestartup** is being saved to the Terminal Emulator's floppy or hard disk via the tape port. The specific location where this file is to be located is determined using the Terminal Emulator's menu functions.

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

The LINK command is used to lock a module into main memory by increasing its link count. The module must already be loaded in order to be "linked". The UNLINK command is used to unlock a module when it is no longer needed. If the module is being used by multiple processes, each must be "unlinked" if they no longer need the linked file. Unlinking unneeded modules provides a more efficient use of MiniCOP's memory resources. The following example illustrates the LINK and UNLINK commands.

```
LOAD /D0/PACK/extrusion
LINK extrusion
UNLINK extrusion
UNLINK extrusion
```

First the file **extrusion** is loaded into main memory by the LOAD command. The module's name and other pertinent information are recorded in the module directory and the module's link count increments to 1. Next, the LINK command links **extrusion** into main memory and the link count increments to 2. Once the module is no longer needed it can be unlinked. The first UNLINK command decrements the link count by 1 and it then equals 1. The module is still present in memory since the link count is greater than 0. This prevents one process from removing the module when others may still be using it. To remove the module, the UNLINK command is repeated until the link count equals 0. The final command line in the example unlinks **extrusion** again, and the link count now equals 0. The module **extrusion** is now removed from main memory.

CAUTION

Do not totally unlink modules that are still in use by other processes. Doing so will cause any programs using the unlinked module to crash.

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

MEMORY FRAGMENTATION

Position-independent code programs can be loaded at any free memory address, but they cannot be dynamically relocated. Programs must remain at the address where they are loaded. This can lead to a situation known as memory fragmentation.

Memory fragmentation occurs because a program is assigned to the first free memory space large enough to hold it. If at the beginning of a process several modules are loaded into memory and during operation these modules are repeatedly loaded/linked and unloaded (unlinked), memory fragmentation will occur. These fragments can be relatively small on an individual basis, but taken together can add up to a large amount of memory. Eventually it is possible that a module will fail to load due to a lack of a large enough area of contiguous memory even though sufficient free memory exists.

The MFREE command can be used to display the total amount of memory available in numbers of pages (1 page = 256 bytes), the hexadecimal address of each free block of memory (block equals one or more contiguous pages), and the size of each free block in numbers of pages. The MDIR e command can be used to display the address of each module in main memory. The information from these commands when taken together can be used to gauge the amount of memory fragmentation in the system.

Memory fragmentation can be reduced by unlinking scattered memory modules (being careful not to unlink any modules in use) and reloading them. The modules will then be placed into memory with no or limited gaps. This technique of "housecleaning" should be performed on a regular basis, determined by the volume of memory fragmentation being created by any one MiniCOP application. Additionally, all necessary ports should be opened upon entering the OS-9 SHELL. This causes OS-9 to allocate port buffers in a contiguous fashion, again avoiding fragmentation.

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

Figure 62 illustrates the typical problem associated with memory fragmentation. Notice that there are eight active programs in memory, but that they are not contiguous. This has created several memory fragments. The eight programs are using 36.1K bytes of the available 48K in the main memory's workspace, leaving 11.9K bytes free. If an attempt is made to load the 8.3K byte program 9 the load operation will fail since the largest contiguous block of free memory is only 2.7K bytes in size.

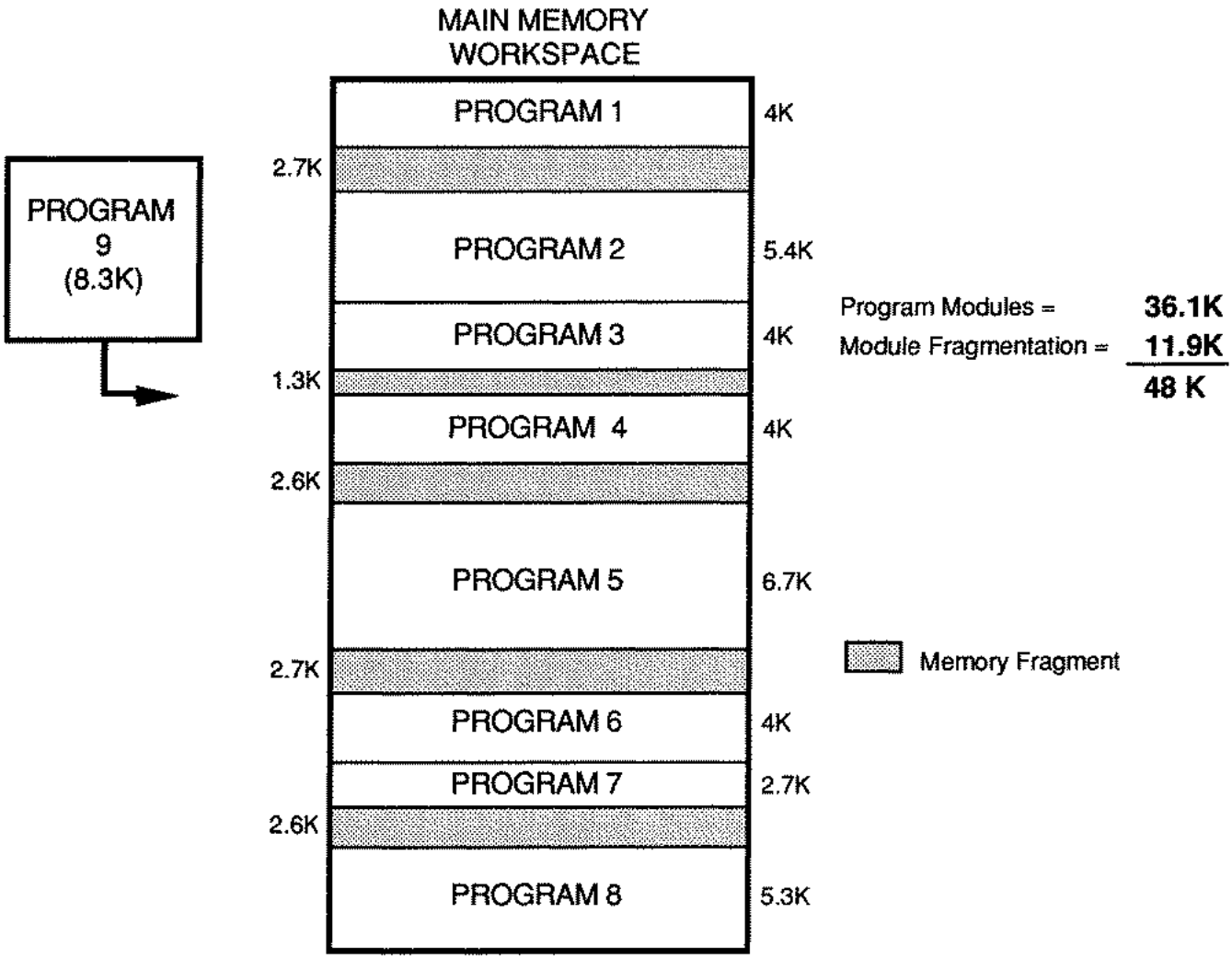


FIGURE 62 - FRAGMENTED MEMORY

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

Figure 63 illustrates the same memory once the fragmentation has been eliminated. The eight current program modules have been unlinked and reloaded to form contiguous blocks of used memory. The remaining free memory is no longer fragmented and program 9 can be loaded with room to spare.

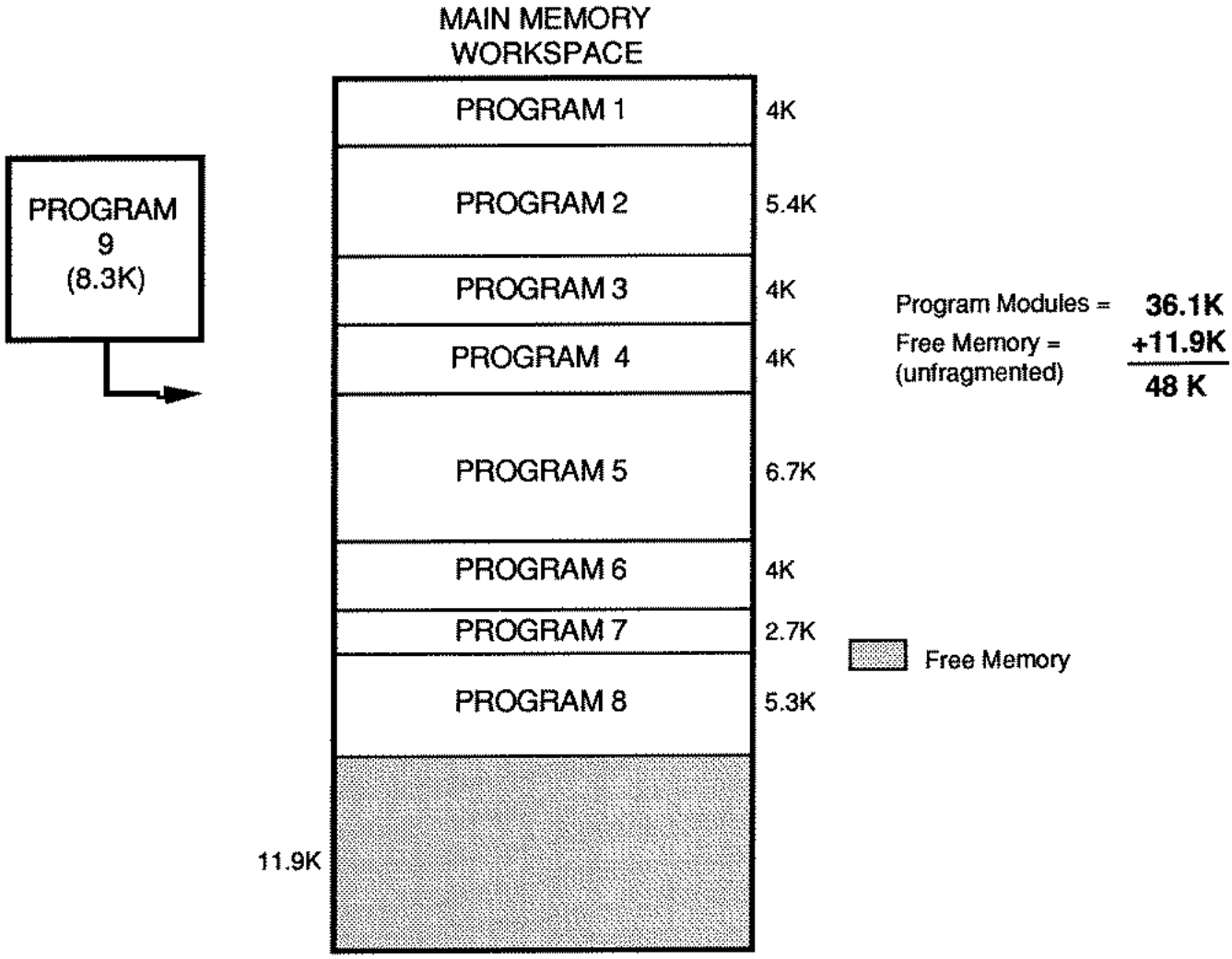


FIGURE 63 - UNFRAGMENTED MEMORY

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

UNPROTECTED I/O OPERATIONS

The special pathlist (**/D0@**) can be used to perform raw I/O operations in COPRAM. When this pathlist is used, all of COPRAM memory is treated as one logical file. Any byte or sector in COPRAM can be read or written to without regard for the actual file structure. This feature is used by utility programs such as DIR, ATTR, and MFREE to access sectors of COPRAM that are not part of data areas of a file (such as the header area).

To convert a byte address of an @ file to a logical sector number, multiply the sector number by 256 (number of bytes per sector). Conversely, the logical sector number of a byte address is the byte address module of 256. As an example, the following command will perform a complete dump of every byte of data in COPRAM

OS9:DUMP /D0@

WARNING

Use extreme caution if attempting unprotected I/O operations (physical sector I/O). Since any sector can be written on, it is possible to severely damage file structure and data in COPRAM. Using @ completely defeats the file security and record-locking systems.

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

OS-9 SYSTEM COMMANDS LISTING

SYNTAX

Each OS-9 Command includes a specific syntax. Syntax refers to all rules that specify precisely how each command and its parameters are to be given so that the operating system can recognize and process the command correctly. Some parameters are optional in a commands syntax while others are required.

[ ]	=	brackets indicate that the enclosed item or items are optional.
{ }	=	braces indicate that the enclosed item or items can be either omitted or repeated.
<arglist>	=	represents a list of arguments (parameters).
<devname>	=	represents any legal device name.
<hexvalue>	=	represents a hexadecimal value.
<modname>	=	represents any legal module name.
<opts>	=	represents one or more options defined in the command description.
<pathlist>	=	represents any legal pathlist.
<pathlist#>	=	# represents start or destination; 1 = start pathlist, 2 = destination pathlist.
<permabbv>	=	represents attribute permission abbreviations.
<procID>	=	represents a process number.
<text>	=	represents a string of characters terminated by and end-of-line.

The previous commands syntax list does not include the SHELL's built-in command options (such as alternate memory size, I/O redirection, etc.). This is because the SHELL will filter out its options from the command line prior to passing it to the program or module called.

COMMANDS

This section consists of a list of OS-9 commands in alphabetical order. Each command description includes a syntax listing. To the right of each command name you will also find values for module size and running size. Module size indicates the amount of memory occupied by the module itself, while running size indicates the amount of memory occupied when the module is executing. Both of these values are given in 256 byte blocks (for example, a module size of 2 equals two 256-byte blocks or 512 bytes).

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

CHANGE FILE SECURITY ATTRIBUTES	Module Size: 2
SYNTAX: ATTR <pathlist> [<permabbv>]	Running Size: 4

The ATTR command is used to examine or change the security permissions of a file. To enter the command, enter ATTR followed by the pathlist for the file that has security permissions to be changed. This is followed by the list of permissions to be turned ON or OFF. A permission is turned ON by giving its abbreviation, or turned OFF by preceding its abbreviation with a minus (-) sign. Permissions not explicitly named are not affected. If no permissions are given, after an ATTR operation, the current or new file attributes are listed.

The file permission abbreviations and their definitions are illustrated in Figure 64.

NOTE

OS-9 was designed for use on multi-user systems, hence the use of public vs. owner permission attributes. The ATTR command is therefore used to protect a file from accidental changes by the owner. The public attributes are not used by OS-9 files in the MiniCOP.

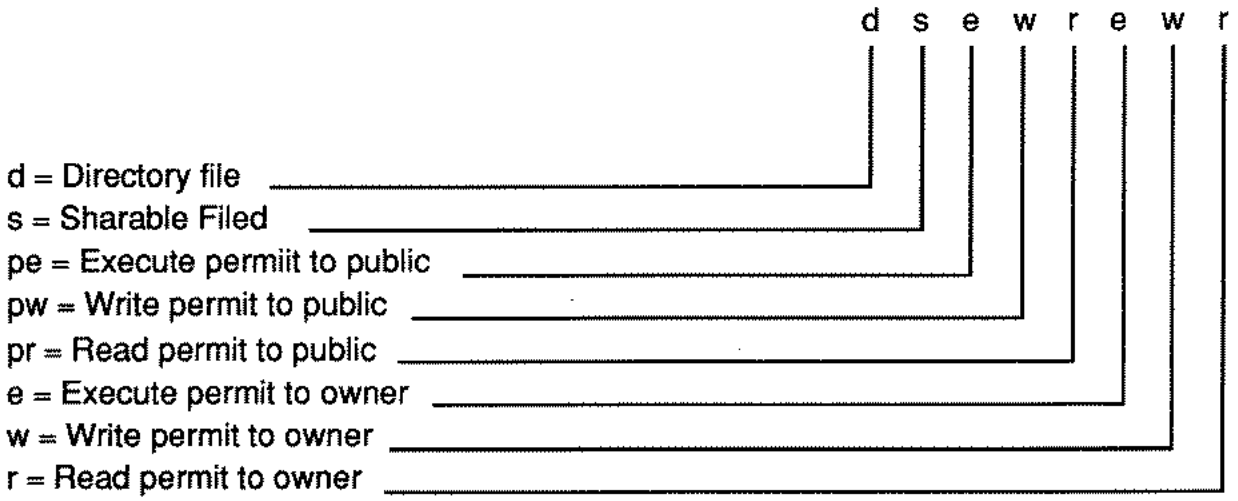


FIGURE 64 - FILE PERMISSION ABBREVIATIONS AND DEFINITIONS

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

The "**-a**" option is also available. It inhibits the listing of new file attributes after an ATTR operation.

EXAMPLES:

OS9:ATTR DATA.DIR **d-ewrewr**	(displays the file's current attributes) (all attributes except -- shareable)
OS9:ATTR DATA.DIR -d **--ewrewr**	(turns the directory attribute OFF)
OS9: ATTR /D0/SOURCE/myfile -w **---wre-r**	(turns the owner write permission OFF)
OS9:ATTR /D0/SOURCE/myfile -r -a	(turns the owner read permission OFF, does not print the new attribute listing)
OS9:ATTR /D0/SOURCE/myfile w **---wrew-**	(turns the owner write permission ON)

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

BINEX
EXBIN

CONVERT A BINARY FILE TO AN S-RECORD FILE **SYNTAX: BINEX <pathlist1> <pathlist2>**	**Module Size: 3** **Running Size: 4**
CONVERT AN S-RECORD FILE TO A BINARY FILE **SYNTAX: EXBIN <pathlist1> <pathlist2>**	**Module Size: 2** **Running Size: 4**

S-Record files contain records that represent binary data in hexadecimal character form. The S-Record format is often directly accepted by commercial PROM programmers, emulators, logic analyzers, and similar devices that can communicate to the MiniCOP via an RS232 interface. The format can also be useful in applications where files are to be transferred over data links that can only handle character-type data.

BINEX

This module converts an OS-9 binary format file from <pathlist1> to an S-Record format file. <pathlist2> is used to indicate the file name and path where the S-Record version is to be located. If invoked on a nonbinary file, a warning message is displayed and you are asked if BINEX should proceed anyway. A "Y" response indicates yes; any other response will terminate the operation. S-Record files have a header that stores the file name for informational purposes and each data record has an absolute memory address. The absolute memory address is not meaningful to OS-9 since it uses position-independent code, however, the S-Record format requires both. BINEX will therefore prompt you for a program name and starting address.

EXAMPLE:

OS9:BINEX /D0/CMDS/scanner /D0/SRECORDS/scanner.SR
Enter starting address for file: $100 (enter hexadecimal starting address)
Enter name for header record: (enter S-Record file name)

The S-record file **scanner.SR** can now be downloaded to a device connected to one of the MiniCOP's ASYNC ports.

EXAMPLE:

OS9:LIST /D0/SRECORDS/scanner.SR >/T3 (sends **scanner.SR** to the device at port 3)

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

EXBIN

EXBIN is the inverse of the BINEX operation. <pathlist1> is assumed to be an S-Record format test file that is to be converted to a pure binary format. <pathlist2> is used to indicate the file name and path where the binary version is to be located. The load addresses of each data record must describe contiguous data in ascending order.

EXBIN does not generate a check for the proper OS-9 module header or CRC value that is required to actually load the binary file. The INDENT or VERIFY commands can be used to check the validity of the modules if they are to be loaded or run.

EXAMPLE:

OS9:EXBIN /D0/SRECORDS/reader.SRec /D0/DATADIR/reader
(converts the S-Record file **reader**.
SRec to a binary file **reader**)

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

BUILD A TEXT FILE FROM THE STANDARD INPUT PATH	Memory Size: 1
SYNTAX: BUILD <pathlist>	Running Size: 4

The BUILD command is used to construct text files. A file is created in the COPRAM directory under the specified pathlist and name. A "?" prompt requests you to enter a line of text. Each line entered can be a maximum of 80 characters in length and is written to the standard output pathlist (specified file). The [RETURN] key marks the end-of-line, which produces another "?" prompt to enter the next line. A [RETURN] with no line of text indicates the end-of-file, which causes the BUILD operation to terminate.

The BUILD command can be used to construct pure text files used to display messages to an output when listed by another program.

EXAMPLES:

OS9:BUILD opsmessage1
? ******WARNING******
? Quench Oil Level
?BELOW MINIMUM!

(creates a text file called **opsmessage1** in the current data directory)

OS9:BUILD /T2

(copies standard input path [keyboard] entries to the device [printer] at port 2)

The standard input path can also be redirected to a file.

EXAMPLES:

OS9: BUILD mytext /T2

(copies the file **mytext** to the device at port 2)

OS9:BUILD mytext /T1

(lists the file **mytext** to the device at port 1)

It is also possible to use text files as procedure file routines when they consist of OS-9 command lines. When the file is called by the SHELL, each command is executed as if it were entered from the OS9: prompt and all rules for command line execution are in affect.

EXAMPLE:

OS9:BUILD
? MAKDIR /D0/SOURCE/BATCHISTORY
? CHD /D0/SOURCE/BATCHISTORY
? CLEAR /T3
? COPY /T3 reactor

(creates a directory called BATCHISTORY; under the SOURCE directory, changes current data directory to BATCHISTORY; clears port 3 of any extraneous data; copies the data from port 3 to a file called reactor)

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

CHANGE THE CURRENT WORKING DATA DIRECTORY
SYNTAX: CHD <pathlist>

CHANGE THE CURRENT WORKING EXECUTION DIRECTORY
SYNTAX: CHX <pathlist>

CHD and CHX are built-in SHELL commands used to change OS-9's current working data and execution directories, respectively. Many OS-9 commands work with user data in the form of text files, programs, etc. These commands default to the current working directory, while some others default to the current execution directory. These commands allow you to assign the current working data and execution directory pointers to any valid directory in COPRAM.

Since these are built-in SHELL commands, they will not appear in the CMDS directory.

EXAMPLES:

OS9:CHD /D0/SOURCE	(change the current working directory to SOURCE)
OS9:CHX /D0/PACK	(change the current working directory to PACK)
OS9:CHD ..	(change the current working directory to the next higher-level directory)
OS9:CHX /D0/CMDS; CHD/D0	(sequentially change the current working execution directory to CMDS and the current working data directory to /D0)

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

CLEAR

CLEAR THE PORTS INPUT BUFFER OF EXTRANEOUS CHARACTERS	Module: 1
SYNTAX: CLEAR /<serial device descriptor name> [R]	Running Size: 1

The CLEAR command clears extraneous characters from the input buffer of a designated port (T1, T2, T3, T4, TP, or user-defined descriptor). Such characters may be generated when a transmission (for example, a save or load operation) is not properly completed. The port must be cleared before performing subsequent data transmissions. The port may also be cleared as a matter of course before any data transfer to ensure error-free transmission.

The **R** option clears the port buffer and resets the port's device driver to the specified port descriptor parameters (for example, 1 stop bit, 8 data bits, no parity).

EXAMPLES:

OS9:CLEAR /TP	(clears the port 2 buffer for data transmission between port 2 and the Terminal Emulator)
OS9:CLEAR /T3	(clears the port 3 buffer and resets the port's device driver to the specified descriptor parameters)

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

COPY

COPY A FILE FROM ONE PATHLIST TO ANOTHER	Module Size: 3
SYNTAX: COPY <pathlist1> <pathlist2>	Running Size: 6

The COPY command makes a copy of the file from <pathlist1> and places it in the location designated by <pathlist2>. The file from the <pathlist1> must exist, either in COPRAM or be available from the Terminal Emulator's floppy or hard disk drives. The destination file is automatically created in the location designated by the <destination pathlist>. The <pathlist2> can direct the copied file to either COPRAM or a MiniCOP port. File data can be of any type.

If the <start pathlist> designates a MiniCOP port, the system will go into a wait state when the COPY command is executed. The system will remain in the wait state until a character is received, then the file contents will be transferred to the <destination pathlist>. For example, the COPY command can be used to transfer files from the Terminal Emulator to the MiniCOP's memory. In this case, the <start pathlist> will designate the tape port (/TP), and the first character received is the [RETURN] key from the Terminal Emulator, which will initiate the transmission.

A file is transferred using large block reads and writes until an end-of-file character is read from the <start pathlist>. Because block transfers are used, normal output processing of data does not occur on connected character-oriented devices such as terminals or printers. The LIST command is a more efficient method of transferring a text file to a printer or terminal.

There are two optional modifiers available for use with the COPY command. The "**-p**" option removes the write protect attribute from the file identified in the <destination pathlist>. This is used to rewrite over an existing file that normally can not be written to. The "**-s**" option removes all control characters from the file being transferred, except for the standard line feed (LF) and carriage return (CR). The OS-9 alternate memory-size modifier can be used to give the COPY operation a larger memory buffer in which to execute. This will increase the speed of the transfer by reducing the number of read/write operations.

EXAMPLES:

OS9:COPY /D0/TEMP/cip /D0/APPLCS/cip	(a copy of **cip** is transferred form TEMP to APPLCS)
OS9:COPY /D0/CMDS/DIR /D0/DIR2	(a copy of the command **DIR** is transferred from CMDS to D0 and is renamed **DIR2**)
OS9:COPY /TP /D0/SOURCE//test -p #10K	(a file from the Terminal Emulator is received from the tape port, transferred to **SOURCE**, named **test**, rewritten if it already exists, and given a #10K buffer to speed up the operation)

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

DATE

DISPLAYS THE CURRENT SYSTEM DATE AND TIME	Module Size: 1
SYNTAX: DATE [t]	Running Size: 2

The DATE command displays the current system date from the MiniCOP's real-time clock, and if the "**t**" option is given, the current system time is also displayed. To set the current date and time use the SETIME command.

EXAMPLES:

OS9:DATE (displays the current date)
June, 27, 1991

OS9:DATE t (displays the current date and time)
June 27, 1991 15:37:45

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

DEL

DELETES A COPRAM DIRECTORY	Module Size: 1
SYNTAX: DEL <pathlist> {<pathlist>} [-x]	Running Size: 3

The DEL command is used to delete the COPRAM file or files specified by the <pathlist>. You must have write permission for the file in order to delete it. Directory files cannot be deleted with the DEL command, use the DELDIR command for this operation.

If no pathlist precedes the given file name, OS-9 will default to the current working data directory. The "**-x**" option can be used to direct the DEL operation to a file in the current working directory, thus eliminating the need to enter its full pathlist.

EXAMPLES:

OS9:DEL warning1	(deletes **warning1** from the current data directory)
OS9:DEL /D0/RECIPES/small	(deletes **small** from RECIPE)
OS9:DEL stacker.pk -x	(deletes **stacker.pk** from the current execution directory)
OS9 DEL data1 data2 data3	(deletes **data1**, **data2** and **data3** from the current data directory)

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

DELDIR

DELETE A DIRECTORY AND ALL FILES IN IT	Module Size: 3
SYNTAX: DELDIR <pathlist>	Running Size: 3

The DELDIR command is a convenient alternative used to manually delete directories and the files they contain. It is only used when all files the directory contains and the directory itself are to be deleted.

EXAMPLE

```
OS9:DELDIR TEMPFILES
Deleting directory file.
List directory, delete directory or quit? (l/d/q)
```

This example is requesting that the directory TEMPFILES be deleted, including all files it contains. When executed OS-9 will display the message as illustrated in the example.

An "**l**" response causes the DIR e command to be executed on the directory. This will list all files with their individual data, to include: name, byte count, sector, attributes, when last modified, and owner. A further prompt requests "**Continue? (y/n)**". If you respond with a "**y**" or "**Y**" for yes the DELDIR is executed. Any other response will abort the operation.

A "**d**" response initiates the DELDIR operation without first listing the directory contents. All files will be unconditionally deleted.

A "**q**" response aborts the DELDIR operation.

The directory to be deleted may include directory files, which may themselves contain directory files. In this case, DELDIR operates recursively (that is, it calls itself) so all lower-level directories are deleted as well. The lower-level directories are always deleted first.

You must have write permission for all files and directories encountered in the DELDIR operation. If not, the operation will abort upon detecting the first that is write protected. The DELDIR command automatically calls the DIR, ATTR, and DEL program modules, which must reside in the current execution directory. The absence of any of these modules causes an error and the DELDIR operation is aborted.

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

DIR

DISPLAY THE NAMES OF ALL FILES IN A DIRECTORY	Module Size: 3
SYNTAX: DIR [<pathlist>] [e] [x]	Running Size: 4

The DIR command displays a formatted list of file names in a COPRAM directory file. If no <pathlist> is given, the current data directory is displayed. If the "**x**" option is given, the current execution directory is displayed. If a <pathlist> is given, the directory it relates to is displayed.

The "**e**" option can be combined with the DIR, DIR x, or DIR <pathlist> to additionally display each file's name, byte count, sector, attributes, when last modified, and owner.

EXAMPLES:

OS9:DIR	(displays the current data directory)
OS9:DIR e	(displays the entire current data directory description)
OS9:DIR x	(displays the current execution directory)
OS9:DIR x e	(displays the entire current execution directory description)
OS9:DIR ..	(displays the contents of the parent of the current data directory)
OS9:DIR /D0/SOURCE e	(displays the entire contents of the directory **SOURCE**)

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

DISPLAY

DISPLAYS CONVERTED CHARACTERS	Module Size: 1
SYNTAX: DISPLAY <hexvalue> {<hexvalue>}	Running Size: 2

The DISPLAY command reads one or more hexadecimal values given as parameters, converts them to ASCII characters, and writes them to the standard output. It is commonly used to send special characters such as cursor and screen control codes to terminals and other I/O devices.

EXAMPLES:

OS9:DISPLAY 41 42 43 43 44 45 46 (converts hexadecimal to ASCII, display to standard output)

ABCDEF

OS9:DISPLAY 15 >/T2 (converts hexadecimal to ASCII, sends to MiniCOP port 2)

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

DSKBAK

SAVES, LOADS, VERIFIES COPRAM MEMORY	Module Size: 3
SYNTAX: DSKBAK	Running Size: 3

The DSKBAK command is used to save, load, and verify the contents of COPRAM memory. The entire contents of COPRAM is operated on as a whole. An exact copy of all sectors is sent to the specified port. An example of the DSKBAK command is as follows:

EXAMPLE:

OS9:DISBAK	(the system prompts you with the following:)
(1) Save to, (2) retrieve from, (3) verify tape?	(enter the desired selection number)
(O) Only allocated Sectors or (A) All Sectors?	(enter the desired selection letter)
Tape Ready? Hit Return to Continue	(if media to save to, retrieve from, or verify with is ready, press [RETURN] to complete the operation)

After indicating the desired operation (save, retrieve, verify), choose either "Only allocated sectors" or "All sectors".

"Only allocated sectors" (O) operates on only those sectors that contain data. Selecting this option reduces the amount of space required in the media by ignoring blank areas.

"All sectors" (A) operates on all available sectors regardless of whether they contain data or are blank.

NOTE

The term "Tape", as used with the DSKBAK command, should be interpreted differently depending on whether a tape-storage-medium device is being used (such as the 623-50/51 Loader/Terminal) or a disk-storage-medium device is being used (such as the 623-60 MS-DOS Loader).

For applications where a tape-storage-medium device is being used (such as the 623-50/51 Loader /Terminal — which uses cassette tapes as its storage medium), equate the word "Tape" with the tape drive of the Loader/Terminal. For such applications, note also that the tape must be positioned to the beginning of the desired program before saving to or retrieving from the tape.

For applications where a disk-storage-medium device is being used (such as the 623-60 MS-DOS Loader), equate the word "Tape" on the display screen to mean the appropriate disk drive of the Loader (that is, either the floppy or hard disk drive).

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

DUMP

FORMATTED FILE DUMP IN HEXADECIMAL AND ASCII	Module Size: 2
SYNTAX: DUMP [<pathlist>]	Running Size: 4

The DUMP command produces a formatted display of the data bit contents of a program module or file in COPRAM. If no pathlist is given, input from the Terminal Emulator keyboard is converted to hexadecimal form. The output is written to the standard output. This command is commonly used to examine the contents of nontext files and is primarily a diagnostic toll.

The data is displayed 16 bytes per line in both hexadecimal and ASCII character formats. Data bytes that have nondisplayable values are represented by periods (.) in the character area. The addresses displayed by the DUMP are relative to the beginning of the file. Because memory modules are position-independent and are stored in files exactly as they exist in memory, the addresses shown correspond to the relative load addresses of memory module files.

EXAMPLES:

OS9:DUMP	(displays any keyboard input in hexadecimal form)
OS9:DUMP product3	(displays the contents of product3 from the current data directory in hexadecimal and ASCII formats)
OS9:DUMP /D0/SOURCE/mastermix	(displays the contents of **mastermix** from **SOURCE** in hexadecimal and ASCII formats)
OS9:DUMP purge.control >/T2	(dumps the contents of purge.control to the device at port 2)

SAMPLE OUTPUT:

```
Addr  01   23   45   67   89   AB   CD   EF   0 2 4 6 8 A C E

0000  87CD 0038 002A F181 2800 2E00 3103 FFE0 . M . 8 * q . ( . . . 1 . . .'
0010  0418 0000 0100 0101 0001 1808 180D 1B04 . . . . . .
0020  0117 0311 0807 1500 002  5445 52CD 5343 . . . . . * T E R M S C
0030  C641 4349 C10E 529E                     F A C I A . R .
```

In this sample output:

Addr	=	starting address
01 through EF	=	data bytes in hexadecimal format
0 through E	=	data bytes in ASCII format

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

EX

EXECUTE THE PROGRAM AS AN OVERLAY
SYNTAX: EX <modname>

The EX command is a built-in SHELL command that causes the process executing the SHELL to start executing another program. It permits a transition from the SHELL to another program without creating another process, thus conserving system memory.

EX is often used when the SHELL is called from another program to execute a specific program, after which the SHELL is not needed. For instance, applications that only use BASIC09 need not waste memory space on the SHELL.

The EX command should always be the last command on a SHELL command line because any successive lines will not be processed.

Since EX is a built-in SHELL command, it does not appear in the CMDS directory.

EXAMPLES

OS9:EX BASIC09	(if BASIC09 is exited, OS-9 will reset since no SHELL is waiting)
OS9:EX SHELL	(aborts the current SHELL and creates a new one)

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

EXBIN

CONVERT AN S-RECORD FILE TO A BINARY FILE — **Module Size: 2**
SYNTAX: EXBIN <pathlist1> <pathlist2> — **Running Size: 4**

NOTE
For additional information, refer to the BINEX command.

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

FREE

DISPLAY THE FREE MEMORY REMAINING IN COPRAM	Module Size: 3
SYNTAX: FREE	Running Size: 5

The FREE command displays the number of unused 256-byte sectors available in COPRAM for use as new files or to expand existing files. The FREE command displays the creation data, the total number of sectors (1024), the number of free sectors, and the largest block of contiguous free sector memory.

A file or program module cannot be loaded into COPRAM memory unless there is a contiguous block of free memory available that is large enough to contain it. Memory is not automatically shifted to consolidate free memory.

NOTE

For additional information regarding free memory, refer to *Memory Fragmentation* under *Working with OS-9 Files* in the *OS-9 Programming Background* subsection.

EXAMPLES:

OS9:FREE (displays the free memory and other information as shown)

"RAMDISK" created on 88/01/28
Capacity: 1, 024 sectors (1-sector clusters)
699 free sectors, largest block 699 sectors

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

IDENT

DISPLAYS FREE MEMORY IN COPRAM	Module Size: 7
SYNTAX: IDENT <pathlist> [-m] [-v] [-x] [-s]	Running Size: 11

The IDENT command is used to display header information from OS-9 memory modules. IDENT displays the module size and CRC bytes (with verification). For Program and device driver modules, it also displays the execution offset and the permanent storage requirement bytes. IDENT will print and interpret the type/language and attribute/revision bytes. In addition, IDENT displays the byte immediately following the module name, since most modules use this byte to indicate module edition.

IDENT can display information for any module contained in a COPRAM file. Additional parameters can be used to modify the commands execution.

PARAMETERS:

[-m]	=	<pathlist> is assumed to be a main memory module
[-v]	=	No verification of the modules CRC value
[-x]	=	Implies the pathlist begins in the execution directory
[-s]	=	Causes IDENT to display the following module information on a single line:

Edition Byte	(first byte after module name)
Type/Language Byte	
Module CRC Value	(a "." = verify ok, "?" = verify failed, blank = parameter -v selected)

EXAMPLES:

OS9:IDENT -m ident (displays the header information on the module **ident**, which is located in main memory as defined by the **-m** parameter)

Header for:Ident	(Module name)
Module size:$06A5 #1701	(Module size)
Module CRC: $1CE78A (Good)	(Good or Bad Module CRC value)
Hdr parity: $8B	(Header parity value)
Exec. off: $0222 #546	(Execution offset)
Data size: $0CA1 $3233	(Permanent storage requirement)
Edition: $05 #5	(First byte after module name)
Ty/La At/Rv: $11 $81	(Type/Language Attribute/Revision)
Prog mod, 6809 obj, re-en	(Module type, Language, Attribute)

OS9:IDENT /D0/CMDS/List -s (displays the header information of **/D0/CMDS/List** on a single line "**-s**")

5	**$11**	**$58BC12**	**.**	**List**	(5 = Edition Type) ($11 = Type/Language) ($58BC12 = CRC Value) (. = CRC Check verifies ok)

(List = Module Name)

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

INKEY

READS A PATH'S BUFFER AND RETURNS ANY CHARACTER FOUND	Module Size: 1
SYNTAX: INKEY ([input path], STRVAR, STRLEN, ER)	Running Size: 0

The INKEY command is actually a subroutine used from BASIC09 to read a designated input path's buffer and return a single character, along with the number of bytes available in the buffer. If no character is given, INKEY defaults to the standard input path, which in most cases will be the Terminal Emulator (keyboard). The variables STRVAR and STRLEN may be dimensioned as BYTE or STRING. The variable ER should be dimensioned as an INTEGER or BYTE.

EXAMPLES: (BASIC09 Statements)

```
DIM PATH:BYTE
OPEN #PATH, "/T3"
RUN INKEY (PATH, STRVAR, STRLEN, ER)
CLOSE #PATH
```

(a single character is read from T3's buffer and returned as the variable STRVAR while the byte length is returned as the variable STRLEN)

```
RUN INKEY (STRVAR, STRLEN)
```

(a single character is read from the standard input buffer and returned as the variable STRVAR while the byte length is returned as STRLEN)

```
DIM STRVAR:STRING[5]
DIM STRLEN:BYTE
REPEAT
PRINT "Enter Z to terminate."
RUN INKEY (STRVAR, STRLEN)
UNTIL STRVAR="Z"
```

(the program remains in the loop until the character "Z" is read from the designated input buffer)

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

KILL

ABORT A PROCESS
SYNTAX: KILL <procID>

Kill is a built-in SHELL command that sends an abort signal to a process with the specified ID number. The PROCS command can be used to obtain process ID numbers. Since it is a built-in command, KILL does not appear in the CMDS directory.

NOTES

The OS-9 KILL command should not be confused with the BASIC09 command of the same name; the two have totally different functions.

If a process is killed and the PROCS command shows it still exists, it is most likely waiting to receive data before it can die. When a process is waiting for I/O operations, it will not die until the operation has completed.

EXAMPLES:

OS9:KILL 5 (aborts the process with ID number 5)

OS9:KILL 22 (aborts the process with ID number 22)

OS9:PROCS (calls and executes the PROCS command)

Usr#	Id	pty	state	Mem	Primary module
0	2	0	active	2	Shell <T1
0	1	0	waiting	1	Sysgo <T1
0	3	0	sleeping	20	Copy <T1

OS9:KILL 3 **(aborts the process with the ID number 3)**

OS9:PROCS (calls and executes the PROCS command)

Usr#	Id	pty	state	Mem	Primary module
0	2	0	active	2	Shell <T1
0	1	0	waiting	1	Sysgo <T1

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

LINK
UNLINK

LINK A MODULE INTO MAIN MEMORY **SYNTAX: LINK <modname> {<modname>}**	**Module Size: 1** **Running Size: 2**
REMOVE A MODULE FROM MAIN MEMORY **SYNTAX: UNLINK <modname> {<modname>}**	**Module Size: 1** **Running Size: 2**

LINK

The LINK command is used to lock a previously loaded module into main memory's module directory. The link count of the module is incremented by one each time it is linked. Linking a module into main memory once for each process that uses it will prevent it from being removed until all processes have unlinked it.

UNLINK

The UNLINK command is used to unlock a linked module from main memory's module directory. The unlinked module may or may not be destroyed and its memory reassigned, depending on whether it is being used by other processes. The link count is decremented by one each time the unlink command is executed. When the link count equals zero the module will be removed from main memory. Any copy of the module which may reside in COPRAM will not be affected. It is good practice to unlink modules when they are no longer in use to make the most efficient use of the available memory resources.

EXAMPLES:

OS9:LOAD COPY	(loads **COPY** into the module directory [link count = 1])
OS9:LINK COPY	(increments COPY's link count by one [link count = 2])
OS9:UNLINK COPY	(decrements COPY's link count by one [link count =1])
OS9:UNLINK COPY	(decrements COPY's link count by one [link count = 0] and removes it from the module directory)
OS9:LOAD ATTR	(loads **ATTR** into the module directory [link count = 1])
OS9:UNLINK	(decrements ATTR's link count [link count = 0] and removes it from the module directory)
OS9:LOAD /D0/PACK/procd1 procd2 procd3	(loads three modules into the module directory)
OS9:LINK procd1 procd2 procd3	(increments each module's link count)
OS9:UNLINK procd1 procd2 procd3	(decrements each module's link count)
OS9:UNLINK procd1 procd2 procd3	(removes each module from the module directory)

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

LIST

LIST THE CONTENTS OF A TEXT FILE	Module Size: 1
SYNTAX: LIST <pathlist> {<pathlist>} {<devname>}	Running Size: 4

The LIST command transfers text lines from one or more COPRAM files to the specified output device or pathlist. If no origination pathlist is given, OS-9 will attempt to locate the specified file names in the current data directory. If no destination pathlist or device is given, OS-9 assumes the default and sends the lines of text to the Terminal Emulator's monitor.

When listing multiple text files from the current data directory, each file name is separated by a space character. When listing multiple files from a directory other than the current data directory, it is necessary to repeat the entire pathlist for each file separated by a space character. This is true even if all files being listed are from the same directory (if it is not the current data directory). Multiple files can also be listed from different directories, using one LIST command, as long as the entire pathlist is given for each text file that is not in the current data directory.

EXAMPLES:

OS9:LIST rec001	(lists the file **rec001** to the Terminal Emulator monitor)
OS9:LIST myproc >T2	(lists the file **myproc** to the device at port 2, T2)
OS9:LIST /DO/TESTPROCS/errval	(lists the file **errval** to the Terminal Emulator monitor from the directory **TESTPROCS**)
OS9:LIST proc01 > mastprocs	(lists the file **proc1** to a file called **mastprocs**)

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

LOAD

LOAD A MODULE FROM COPRAM INTO MAIN MEMORY	Module Size: 1
SYNTAX: LOAD <pathlist>	Running Size: 3

The LOAD command transfers copies of one or more COPRAM files into main memory. The names of the modules are added to the module directory. Only program modules and packed BASIC09 procedures can be loaded into main memory. Attempting to LOAD text files or BASIC09 source procedures will generate **Error Code #216 - Path Not Found**. In this case, the error code indicates that neither a program module or packed BASIC09 procedure was found (note that OS-9 file names are considered as part of the pathlist). If the module identified in the <pathlist> carries the same name and type as a module already in main memory, the module with the highest revision level is kept. The LOAD command defaults to the current working execution directory unless otherwise directed. Each module that is loaded into main memory is linked, even if multiple modules are loaded in the same SHELL command line.

NOTE

The OS-9 LOAD command should not be confused with BASIC09 command of the same name; the two have totally different functions.

EXAMPLES:

OS9:LOAD ATTR	(loads and links the program module **ATTR**)
OS9:LOAD /D0/PACK/pakbas1	(loads and links the packed BASIC09 procedure **pakbas1**)
OS9:LOAD COPY LIST	(loads and links the program modules **COPY** and **LIST**)

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

MAKDESC

CREATE A PORT DEVICE DESCRIPTOR	Module Size: 7
SYNTAX: MAKDESC	Running Size: 3

The MAKDESC command copies a currently defined port descriptor from the module directory or from COPRAM. It allows you to modify the operating parameters of the descriptor, give it a new name, and attach it to any MiniCOP ASYNC ports. The MAKDESC command is the only way to change a descriptor's parameters (for example, turn the echo OFF on port 4).

The command is called by entering MAKDESC on the command line. You are then asked a series of questions that build the descriptor. These questions are:

1. COPY FROM WHICH PATH NAME?

   The pathlist name must be an existing device descriptor for one of the MiniCOP's ASYNC ports. The ASYNC port device descriptors available after the MiniCOP has been initialized include; T1, T2, T3, T4, and TP.

   Descriptors that have been previously created can also be used. If copying the descriptor from the module directory, simply enter the module's name when requested. If copying a descriptor from anywhere in COPRAM, the entire correct pathlist must be given (for example, /D0/CMDS/<descriptor name>).

NOTE

If copying from the port descriptor TP, an additional parameter will be presented for definition by the user. This parameter, called "delay," defines the amount of time (in 16 msec increments) between when the port using this descriptor is opened and when the FIRQ (backplane servicing) routine is disabled. Default value is 0DHex or 208msec.

2. WHAT IS THE NEW PATH DESCRIPTOR NAME?

   The new name must be unique, otherwise it cannot be loaded into the module directory. The descriptor naming convention follows the same rules as any OS-9 file name as outlined in the following:

   - Maximum of 29 characters,
   - First character must be an alphabetic character,
   - Remaining characters can be: alphabetic, numeric, or a period (.),
   - All other special characters including the space are invalid,
   - Alphabetic characters may be upper or lower case, and
   - File extensions are not required in OS-9.

3. WHICH PORT WILL THE NEW DESCRIPTOR ACCESS?

   The new descriptor must be attached to one of the three or four MiniCOP ASYNC ports, in the same manner as TP is associated with port 2. Enter the desired port number: 1, 2, 3, or 4 (port 4 is only available for use on the 627-1003 MiniCOP).

4. HOW DO YOU WANT TO SET THE NEW DESCRIPTOR'S PARAMETERS?

   There are a total of twenty-eight parameters that can be set for a device descriptor. These parameters are listed in Table 17. Each parameter will be displayed in turn. Pressing <RETURN> selects the existing parameter setting. For numeric parameter values, entering a number and pressing <RETURN> will select the new setting. For nonnumeric parameter values, entering a "+" will cause it to be present (set ON), while entering a "-" causes it to be absent (set OFF).

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

TABLE 17 - MAKDESC PARAMETERS

Parameters	Note	Description
bsb		Erase on backspace: backspace character echoed as a backspace-space-backspace sequence.
-bsb		No erase on backspace: echoes single backspace character only.
bsl		Backspace over line: lines are "deleted" by sending backspace-space-backspace sequences until the line is erased. For use with video terminals only.
-bsl		No backspace over line: lines are "deleted" by printing new line sequences. For use with hardcopy terminals only.
echo		Input characters echoed (repeated) back to terminal.
-echo		Input characters are not echoed (repeated) back to terminal.
lf		Auto-line feed ON: line feeds automatically echoed to terminal on input and output carriage returns.
-lf		Auto-line feed OFF.
pause		Screen pause ON: output is suspended upon full screen. Output resumed by pressing any key. (See "pag" parameter for screen size definition.)
-pause		Screen pause OFF.
upc		Upper-case characters only. Lower-case characters are automatically converted to upper-case.
-upc		Both upper-case and lower-case characters.
abort=h	*2	Set abort character to "h".
bell=h	*2	Set bell (alert) character to "h".
bse=h	*2	Set output backspace character to "h".
bsp=h	*2	Set input backspace character to "h".
del=h	*2	Set input delete line character to "h".
delay=h	*1, *2	Set the amount of time this descriptor will delay to "h" sec's before disabling the backplane communications service routine (FIRQ). Delay is calculated as: h x 16msec = delay.
dup=h	*2	Set duplicate last input line character to "h".
eof=h	*2	Set end-of-file input character to "h".
eor=h	*2	Set end-of-record (carriage return) input character to "h".
null=d	*3	Set null to "d" counts: number of null characters ($00) transmitted after carriage returns for return delay.
pag=d	*3	Set video page length to "d" lines. Used with screen pause (see "pause" parameter for screen pause definition.)
psc=h	*2	Set pause screen character to "h".
quit=h	*2	Set quit character to "h".
reprint=h	*2	Set reprint line character to "h".
xon=h		Set DC1 resume output character to "h".
xoff=h		Set DC2 suspend output character to "h".
command=00	*4	
control=1E	*4	
reserved=00	*4	
reserved=2E	*4	
tout (m)	*5	16-bit timeout value.
tout (l)	*5	16-bit timeout value.

*1 = This selection only appears when copying the /TP descriptor.
*2 = Value is entered in hexadecimal.
*3 = Value is entered in decimal.
*4 = Do not modify these parameters.
*5 = May also be modified using the SETPORT command.

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

5. WILL THIS DESCRIPTOR BE SAVED ON DISK (D) OR THROUGH A PORT (P)?

Save to disk refers to COPRAM memory and is selected by pressing <D> <RETURN>. If save to disk is selected, OS-9 automatically saves the descriptor to the current working execution directory, typically CMDS. MAKDESC displays the pathlist of the location where the descriptor was written when saving to disk. If a descriptor by that name already exists, you will be prompted to REWRITE (Y/N)? A "**Y**" response overwrites the existing descriptor. An "**N**" response aborts the MAKDESC operation. Once a descriptor has been saved to COPRAM, it can be transferred to the module directory via the LOAD command.

Save through a port refers to the MiniCOP's ASYNC ports and is selected by pressing [P] <RETURN>. Once a port save through port is selected you will be prompted for the port name to save through. This operation may require you to prepare the receiving device to accept this information as would be the case with the Terminal Emulator. In that instance, it will be necessary to select the receive function from the Terminal Emulator's menu, designate the drive, and then designate a file name under which the information should be saved.

EXAMPLE:

OS9:MAKDESC
COPY FROM WHICH PATH NAME? T3
WHAT IS THE NEW PATH DESCRIPTOR NAME? EPROMPROG
WHICH PORT WILL THE NEW DESCRIPTOR ACCESS? 3

(DESCRIPTOR PARAMETERS ARE SELECTED)

FILE SAVED TO /D0/CMDS/EPROMPROG
OS9:LOAD /D0/CMDS/EPROMPROG

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

MAKDIR

CREATE A COPRAM DIRECTORY FILE	Module Size: 1
SYNTAX: MAKDIR <pathlist>	Running Size: 3

The MAKDIR command creates a new COPRAM directory file according to the given <pathlist>. You must have "write" permission for all parent directories in the <pathlist> to create a subdirectory. The new directory initially does not contain any files except for the "." and the ".." directory pointers, which indicate its parent ".." and itself ".".

It is recommended, but not mandatory, that directory names be capitalized and other file names use lower-case letters. This type of naming convention will allow you to easily identify directories from files when using the DIR command.

To conserve memory resources, it is recommended that the minimum of new directories be created.

EXAMPLES:

OS9:MAKDIR DATAFILES	(creates a directory DATAFILES under the current data directory)
OS9:MAKDIR /D0/APPLCS/RECIPES	(create RECIPES under APPLCS)
OS9:MAKDIR ../PACKFILES	(creates PACKFILES under the current data directory's parent)

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

MDIR

DISPLAYS THE CONTENTS OF THE MODULE DIRECTORY	Module Size: 1
SYNTAX: MDIR [e]	Running Size: 3

The MDIR command displays the names of all modules currently in the module directory. If the "e" option is given, a full listing of the name, use, attributes, revision, type, size, and physical address of each module is displayed. Not all modules listed by the MDIR command are executable as processes. Always check the type code (via the "e" option) before attempting to execute a module.

EXAMPLE:

```
OS9:MDIR e

   module directory at 09:16:33

Addr   Size   typ   rev   attr   use   module name

C30B   2F8    C1    1     r...         Boottp
C603   23     11    1     r...         TLoad
C626   DC0    D1    1     r...         RBF
D3E6   2D     F1    1     r...         D0
D413   C4     E1          r...         Disk
D4D7   27C    C1    1     r...         Firq
D753   37     F1    1     r...         V4
D78A   709    E1    1     r...         Video
DE93   3C     F1    1     r...         K4
DECF   3C     F1    1     r...         TP
DF0B   3C     F1    1     r...         T1
DF47   3C     F1    1     r...         T2
DF83   3C     F1    1     r...         T3
DFBF   3C     F1    1     r...         T4
DFFB   335    E1    1     r...         SC6551
E330   419    D1    1     r...         SCF
E749   72D    C1    1     r...         IOman
EE76   10F    C1    1     r...         Clock
EF85   139    C1    1     r...         Sysgo
F0BE   DF     C1    1     r...         OS9p3
F19D   4AE    C1    1     r...         OS9p2
F64B   25     C0    1     r...         Init
F670   925    C1    1     r...         OS9p1
B300   4F8    11    1     r...         Shell
B100   1CA    11    1     r...         mdir
```


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

Module Type

The module type consists of two hexadecimal characters each of which carries its own significance. The first character describes the function of the module while the second describes the language of the code in which the module was written. The definition of each possible first and second module type character is listed below. An example breakdown of the module type characters is illustrated in Figure 65.

FIRST CHARACTER

1 = Program Module
2 = Subroutine Module
3 = Data Module
C = OS-9 System Module
D = OS-9 File Manager Module
E = OS-9 Device Driver
F = OS-9 Device Descriptor

SECOND CHARACTER

0 = Data (non-executable)
1 = 6809 Object Code
2 = BASIC09 I-code

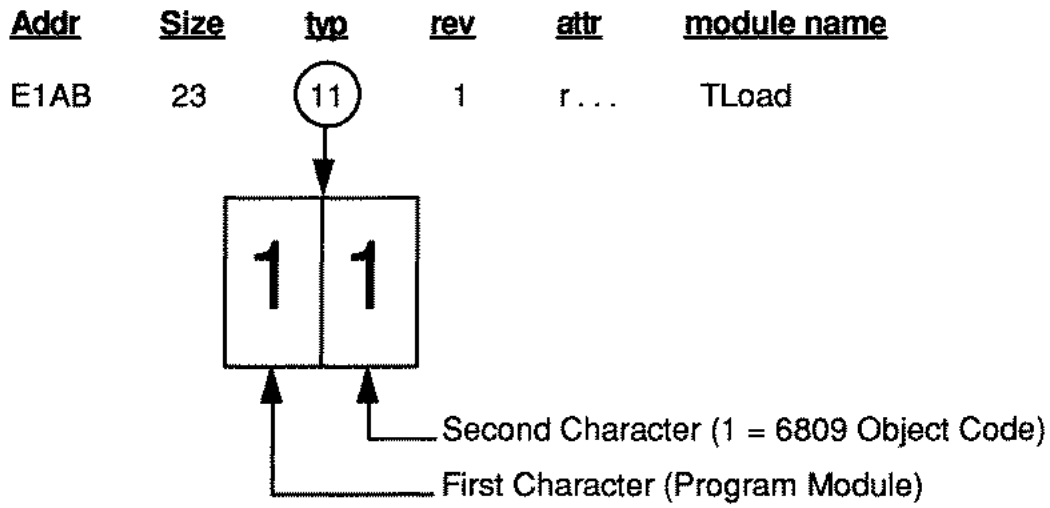


FIGURE 65 - MODULE TYPE CHARACTERS

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

MERGE

COPY AND COMBINE TWO FILES TO THE SPECIFIED PATH	Module Size: 1
SYNTAX: MERGE <pathlist1> {<pathlist1>} <pathlist2>	Running Size: 11

The MERGE command copies and combines several COPRAM files and transfers the combined data to the specified <pathlist2>. Data is copied in the order the files are given. MERGE does not output line editing commands (such as automatic line feed). The MERGE command can be used to perform the same functions as the COPY and RENAME commands.

EXAMPLES:

OS9:MERGE shift1 shift2 shift3 >dailyreport (merges three files into one called **dailyreport**)

OS9:MERGE /D0/TEMP/ redfile bluefile /D0/DATA/purplefile
(**redfile** and **bluefile** are merged from **TEMP** into **DATA** as **purplefile**)

OS9:MERGE message1 message2 >/T3 (merges two files and transfers to the device at port 3)

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

MFREE

DISPLAYS THE AMOUNT AND LOCATION OF FREE MAIN MEMORY	Module Size: 2
SYNTAX: MFREE	Running Size: 3

The MFREE command displays a listing of the addresses of each contiguous main memory block not currently in use and available for assignment. The starting address, ending address, and number of pages are displayed for each block. Each page equals 256 bytes. Included in this display is the total number of free pages. This information can be useful to detect and correct memory fragmentation.

EXAMPLE:

```
OS9:MFREE

Address          pages

 E00-B1FF        164
B900-B9FF          1

Total pages free = 165
```


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

PD

DISPLAYS THE CURRENT DATA OR EXECUTION PATHLIST	Module Size: 2
SYNTAX: PD [d] [x]	Running Size: 3

The PD command displays the pathlist from the COPRAM root directory to the current working data or execution directory. It can be used by a program or user to determine the actual location of files. The "**d**" option selects the current data directory while the "**x**" option selects the current execution directory.

EXAMPLES:

OS9:PD d	(calls the current data directory pathlist for display)
/D0	
OS9:CHD /D0/SOURCE	(changes the current data directory to SOURCE)
OS9:PD d	(calls the current data directory
/DO/PACK	pathlist for display)
OS9:PD x	(calls the current data directory pathlist for display)
/DO/CMDS	(changes the current data directory to
OS9:CHX /D0/PACK	PACK)
OS9:PD x	(calls the current data directory
/DO/PACK	pathlist for display)

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

PROCS

DISPLAYS THE CURRENT SYSTEM PROCESSES	Module Size: 3
SYNTAX: PROCS	Running Size: 12

The PROCS command displays a list of processes currently running in the system. The display reflects system status at the instant the command is executed. Once executed the display is simply a static snapshot of the status of all current processes at the moment of execution. Processes can constantly change states and be removed from or added to the system, so the display may be deceiving.

PROCS displays the user number, ID number, priority, state, memory size, and primary module name. The primary module name will also include the process's standard input pathlist.

The process state can be active, waiting, or sleeping, and memory size is given in the number of 256 byte pages.

EXAMPLE:

```
OS9:PROCS

Usr#   Id   pty   state     Mem   Primary module
  0     2     0   active      3   Shell <T1
  0     1     0   waiting     1   Sysgo <T1
```


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

RENAME

CHANGE COPRAM FILE NAME	Module Size: 2
SYNTAX: RENAME <pathlist> <new name>	Running Size: 2

The RENAME command gives a new name to the specified COPRAM file. You must have "write" permission for the file in order to change its name. It is not possible to change the name of files in: the module directory, OS-9 commands, user-created device descriptors, or the anonymous directory names "." and "..".

NOTE

The OS-9 command RENAME should not be confused with the BASIC09 command of the same name; the two have totally different functions.

EXAMPLES:

OS9:RENAME oldfile newfile (renames the current data directory file **oldfile** as **newfile**)

OS9:RENAME /D0/SOURCE/test comtest (renames the file **test** in **SOURCE** as **comtest**)

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

RESET

INITIATES A SOFTWARE RESET ROUTINE	Module Size: 1
SYNTAX: RESET [F]	Running Size: 1

The RESET command lets you initiate a warm restart of the MiniCOP from the Terminal Emulator keyboard, rather than via the RESET toggle switch located on the unit's frontplate (the reset function of the toggle switch can be disabled via DIP switches). The RESET command is executed from the OS-9 SHELL command line.

EXAMPLE

OS9:RESET (initiates a warm boot of the MiniCOP)

RESET can also be executed as an instruction in a procedure file or as part of a BASIC09 SHELL statement.

The command has the following effects on the MiniCOP and its memory:

1. All program modules that have been loaded into the module directory are removed.
2. Any procedure or procedures running at the time of RESET are killed.
3. The current data and execution directory pointers are reset to /D0 and /D0/CMDS respectively.
4. The current toggle switch configuration is read (CPU board, SW1 position 7).
   - If set to power-up to OS-9, this operation is performed.
   - If set to power-up to PRGMB, this operation is performed.

"F" Modifier

The RESET command has an optional modifier, "F". Specifying the "F" modifier in the RESET command line instructs the system to check the reset history flag (RHF). The "F" modifier provides the MiniCOP with a system RESET function that may be remotely executed or embedded in an application.

The RHF is set during normal system power-up or as part of a previous RESET F command. Execution of a RESET F command tests the RHF. If the flag is SET, the reset command is instructed to clear the RHF and return to the program that called RESET, typically the OS-9 SHELL.

- If the reset history flag is SET:
  1. Clear the RHF.
  2. Do not execute the RESET command. Return to the calling procedure, typically OS-9.

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

- If the reset history flag is CLEAR:

  1. SET the RHF.

  2. Clear the module directory of all RAM modules (program modules not in the module directory after MiniCOP initialization).

  3. Execute the standard reset operation as described above.

The MiniCOP can be configured to power-up to OS-9 or to immediately execute a program called PRGMB, which you as the user might have written. The modified RESET command may be used as the first item in this BASIC09 procedure as shown in the following example:

EXAMPLE:

```
SHELL RESET F
DIM X:INTEGER
DIM port3:BYTE
OPEN #port3,"/T#":WRITE
        •
        •
(additional lines of BASIC09 code)
        •
        •
NEXT x
CLOSE #port3
BYE
```

This example use of the RESET F command might be used to prepare the MiniCOP to resume execution after a power-fail, should one occur. When PRGMB is executed, the RESET F command tests the reset history flag. If the flag is CLEAR, all RAM modules are removed from the module directory, the RHF is SET, and the standard reset operation is executed. After reset, PRGMB is executed again. This time the RHF is SET and the system will clear the RHF and return to the calling procedure (PRGMB). With the flag cleared, the system is prepared to continue program execution.

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

SAVE

SAVES MODULE DIRECTORY FILES TO COPRAM OR A PORT	Module Size: 1
SYNTAX: SAVE <pathlist> <modname> {<modname>}	Running Size: 2

The SAVE command copies a file or files from the module directory to a specified COPRAM directory or port. The module names must exist in the module directory in order to be saved. The SAVE command's syntax appears somewhat differently than most other OS-9 commands. If no <pathlist> is given, OS-9 will default the save operation to the current execution directory. When a <pathlist> is specified it must include the name under which the saved module is to be labeled. The current name (<modname>) of the module being saved must follow the <pathlist>, and must be separated from it by a space. Multiple <modnames > can be given and all will be saved under the name provided in the pathlist. Figure 66 illustrates the SAVE command's syntax.

NOTE

The SAVE command should not be confused with the BASIC09 statement of the same name; the two have totally different functions.

EXAMPLES:

OS9:SAVE ATTR	(**ATTR** is saved from the module directory to the current execution directory)
OS9:SAVE /D0/CMDS/test01 test01	(module **test01** is saved to **CMDS** as test)
OS9:SAVE /D0/EXEC/rstore mod2	(module **mod2** is saved to as **rstore**)
OS9:SAVE /D0/USRMODS/master mod1 mod2 mod3	(modules **mod1**, **mod2**, and **mod3** are saved to **USRMODS** as **master**)
OS9:SAVE /TP prime	(module **prime** is saved to the Terminal Emulator via TP (T.E. must be prepared to receive the file])

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

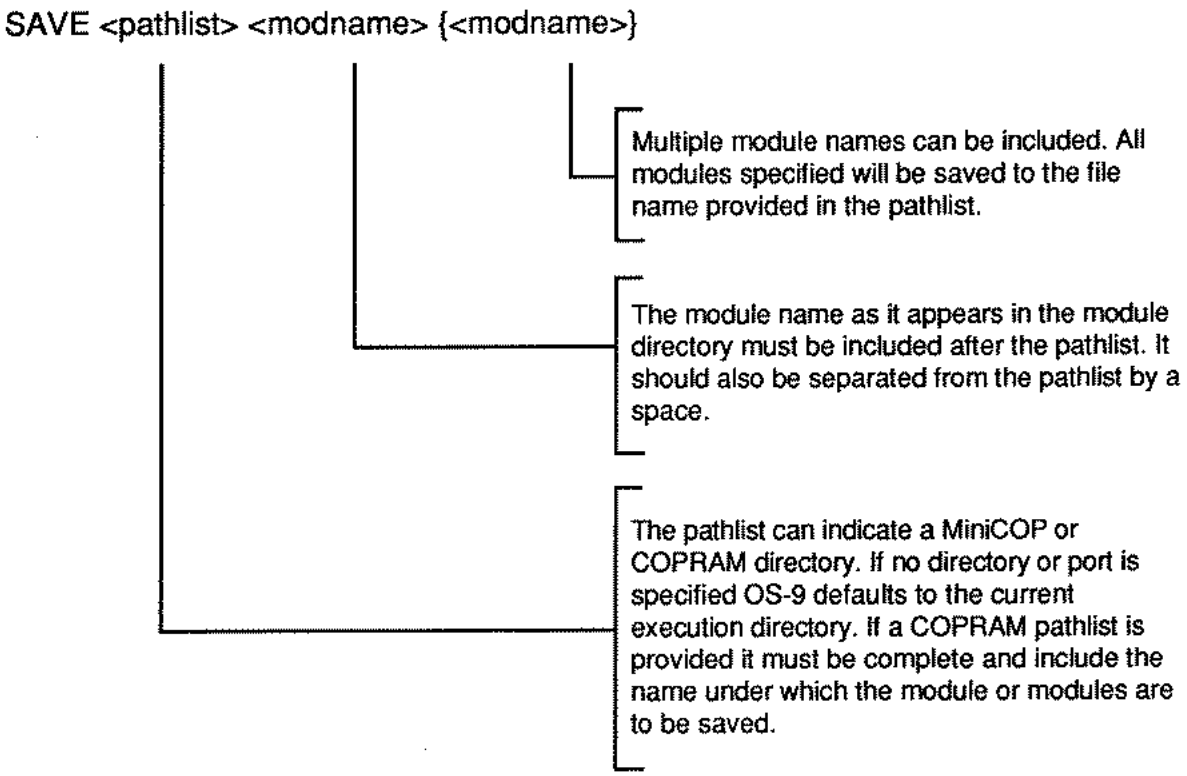


FIGURE 66 - SAVE COMMAND SYNTAX DEFINITION

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

SETIME

SETS THE SYSTEM CLOCK	Module Size: 2
SYNTAX: SETIME [yy,mm,dd,hh,mm,ss]	Running Size: 2

The SETIME command is used to set the system date and time. The date and time are entered as parameters. If no parameters are given, SETIME will issue a prompt requesting the system clock be set and will display the order in which the parameters are to be entered (as shown in their syntax). Each parameter consists of two digits and can be separated from other parameters using the space, colon, semicolon, or slash characters. The time is entered in the twenty-four hour format. If no time is given, the system clock will reset to zero.

EXAMPLES:

OS9:SETIME 91/06/24/11/29/15	(sets the system date and time to: June 24, 1991, 11:29:15AM)
OS9:SETIME 91:12:20:18:07:00	(sets the system date and time to: December 20, 1991, 6:07PM)
OS9:SETIME 91 01 22 06 46 19	(sets the system date and time to: January 22, 1991, 6:46:19AM)
OS9:SETIME 91;04;01;01;15;09	(sets the system date and time to: April 1, 1991, 1:15:09AM)
OS9:SETIME 91/08/03	(sets the system date and time to: August 3, 1991, 00:00:00:)

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

SETPORT

MODIFIES THE BAUD RATE, PARITY, STOP BIT AND/OR TIME-OUT PARAMETERS
SYNTAX: SETPORT </Tx> <b* = b> <p* = p> <w* = w> <s* = s> <t* = t>

The SETPORT command is a utility that allows the you to modify the baud rate, parity, word length, and/or stop bits for any MiniCOP port. The SETPORT command followed by a port descriptor and no parameters will display the port's current settings.

PARAMETER DEFINITIONS

/Tx	=	specific MiniCOP port where x equals the port number (1=T1, 2=T2, 3=T3, 4=T4, P=TP)
b*	=	baud rate
b	=	19200, 9600, 7200, 4800, 3600, 2400, 1800, 1200, 600, 300, 150, 135, 110, 75, or 50
p*	=	parity
p	=	no, odd, or even
w*	=	word length
w	=	8 or 7 bits
s*	=	stop bits
s	=	1 or 2 bits
t*	=	timeout
t	=	set in "ticks" (1/64th sec. or 15.625 msec. increments; for example, timeout of 64 = 1 second)

NOTE
A read request will generate **Error Code #199 – Timeout Error**, when no characters are received within the timeout period. Setting the timeout to zero disables the timeout function.

EXAMPLES:

OS9:SETPORT /T1 b*=4800 p*=odd w*=7 s*=1 t*=640 (each of the five parameters for port 1 have been set as follows: baud rate = 4800; parity = odd; word length = 7 bits; stop bits = 1; and timeout = 10 seconds)

OS9:SETPORT /T3 b*19200 p*=no (only two of the parameters for port 3 are changed, baud rate=19200, and parity=no, all other port parameters remain in their last setting)

OS9:SETPORT /T2 (displays the port's current settings)
BAUD=9600 PARITY=NO WORDLENGTH=8 STOPBITS=1 TIMEOUT=0

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

SETPR

SET OR CHANGE A PROCESS'S PRIORITY
SYNTAX: SETPR <procID> <number>

SETPR is a built-in SHELL command that changes the CPU priority of a process. Priority may range from 0-255, with a higher priority number resulting in more CPU time per time slice. Each process is identified by a decimal <procID> number from 1 to 255, with 255 being the highest priority and 1 the lowest. The PROCS command can be used to display: the user number, ID number, priority, state, memory size, primary module name and the process's standard input pathlist of each process in memory.

SETPR can be used in multitasking applications to increase or decrease a process's priority in relation to other processes. Increasing a process's priority provides it with more processing time per time slice. Since SETPR is a built-in command it will not appear in the CMDS directory.

EXAMPLE:

OS9:PROCS (displays current processes)

Usr#	Id	pty	state	Mem	Primary module
0	2	0	active	3	Shell <T1
0	1	0	waiting	1	Sysgo <T1

OS9:SETPR 2 5 (change process 2 from 0 to 5 priority)

OS9:PROCS (displays current processes)

Usr#	Id	pty	state	Mem	Primary module
0	2	5	active	3	Shell <T1
0	1	0	waiting	1	Sysgo <T1

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

OS-9 COMMAND INTERPRETER	Module Size: 5
SYNTAX: SHELL <arglist>	Running Size: 3

The SHELL command is the OS-9 command interpreter program. It reads data from the standard input path and interprets the data as a sequence of commands. The function of the SHELL is to initiate and control execution of other OS-9 programs.

The SHELL reads and interprets one line of text at a time from the standard input path. Typically, the standard input path is the Terminal Emulator keyboard, or the SHELL is called from another program. In both cases the SHELL interprets each command line until an end-of-file [CONTROL-]] condition occurs, which terminates the SHELL. When the SHELL is called from another program, it interprets the rest of the command line as the first line of input. If this command consists of built-in commands only, additional lines will be read and processed. Otherwise, control will be returned to the calling program after the line has been processed.

The following is a technical syntax specification for the SHELL:

SHELL INPUT LINE FORMAL SYNTAX:

pgm line = <pgm> <pgm>
pgm = [<params>] [<name>] [<modif>] [<pgm params>] [<modif>]

PROGRAM SPECIFICATIONS:

<name> = <module name>
<pathlist>
(<pgm list>)

PARAMETERS:

<params>	=	<module name>	
<delim>	=	space or comma	
<param>	=	CHD <pathlist>	(change current data directory)
	=	CHX <pathlist>	(change execution directory)
	=	EX <name> [<modif>]	(chain to the specified program)
	=	KILL <procID>	(abort process)
	=	p	(turn OS-9 prompt ON; OS9:)
	=	-p	(turn OS-9 prompt OFF)
	=	SETPR <procID> <pty>	(change process priority)
	=	t	(echo input lines to standard output)
	=	-t	(do not echo input lines)
	=	w	(wait for process to die)
	=	*<text>	(comment line; not processed)

MODIFIERS:

<modif>	=	<mod> {<delim> <mod>}	
<mod>	=	;	(sequential execution modifier)
	=	&	(concurrent execution modifier)
	=	< <pathlist>	(redirect standard input)
	=	> <pathlist>	(redirect standard output)
	=	>> <pathlist>	(redirect standard error output)
	=	# <integer>	(set process memory size in pages)
	=	# <integer>K	(set process memory size in 1K blocks)

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

SLEEP

SUSPEND A PROCESS FOR A GIVEN PERIOD OF TIME	Module Size: 1
SYNTAX: SLEEP <tick count>	Running Size: 2

The SLEEP command puts the process to sleep (suspends) for a specified number of clock ticks. It is generally used to generate time delays or break-up CPU-intensive jobs. The duration of a tick is 1/64th second or 15.625 milliseconds. A tick count of zero (0) causes the process to sleep indefinitely (usually awaken by a signal). A tick count of one (1) causes the process to give up its current time slice. Tick counts of greater then one (1) cause the process to sleep a specified number of seconds or a fraction of a second. For example, a tick count of 200 causes the process to sleep for 3.125 seconds (200 x 15.625 msec. = 3125 msec. or 3.125 sec.).

The SLEEP command can be called from BASIC09 procedures also to generate time delays. A BASIC09 statement such as SLEEP 100 prevents an infinite loop from monopolizing CPU time. Another method for preventing a task from monopolizing CPU time is to set its priority low relative to other tasks.

EXAMPLES:

As an OS-9 command line entry:

```
OS9:SLEEP 1000
```

(suspends the current process for 1000 ticks or 15.625 seconds)

As an OS-9 command called from a BASIC09 procedure:

```
WHILE TRUE DO
    DIM path:BYTE
    OPEN #path, "history":READ
    FOR X=1 TO 5000
        READ #path, h$
        OPEN #path, "/T3"
        WRITE #path, h$
        SHELL "SLEEP 100"
    NEXT X
    CLOSE #path
ENDWHILE
```

(built-in time delay of 1.56 seconds per loop)

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

TLOAD

LOAD PROGRAM MODULE FROM THE TERMINAL EMULATOR OR COPRAM	Module Size: 1
SYNTAX: TLOAD <pathlist>	Running Size: 2

The TLOAD command has two primary functions. It can be used to load and link program modules into the module directory from the Terminal Emulator. It can also be used to load and link program modules or packed BASIC09 procedures from COPRAM into the module directory.

When used to load and link program modules from the Terminal Emulator it is always used with the tape port descriptor, /TP. Additionally, Terminal Emulator menu selections are used to prepare, specify, and transmit the file being transferred from floppy or hard disk to the MiniCOP.

Using the TLOAD command to load and link files from COPRAM into the module directory has one distinct disadvantage. TLOAD can only load and link one file at a time. If multiple program modules or packed BASIC09 procedures are specified, only the first file will actually be loaded and linked into the module directory. No error message is generated and all other files are ignored.

NOTE

When loading COPRAM program modules or packed BASIC09 procedures into the module directory, **it is recommended that the LOAD command be used** for this purpose. TLOAD will only load the first file named if multiple modules are named.

EXAMPLES:

OS9:TLOAD /TP	(loads and links the program module or modules received from the Terminal Emulator at /TP)
OS9:TLOAD batchmix1.pac	(loads the file named **batchmix1.pac**.from the current execution directory*)
OS9:TLOAD /D0/APPLCS/PACK/recipe1	(loads **recipe1** from **PACK***)
OS9:TLOAD ATTR BINEX BUILD	(loads **ATTR** from the current execution directory; all other modules ignored*)

* Refer to the previous note regarding this command.

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

TMODE

CHANGE THE TERMINAL EMULATOR OPERATING PARAMETERS	Module Size: 3
SYNTAX: TMODE [.<pathnumber>] [<arglist>]	Running Size: 3

The TMODE command is used to display or change the operating parameters of the Terminal Emulator. A period followed by a number (<pathnumber>) may be included to optionally specify the affected path. If this command is used in a SHELL procedure file, the optional <pathnumber> is used to specify one of the standard I/O paths (0, 1, or 2) to change the Terminal Emulator's operating characteristics. The change will stay in effect until the path is closed.

A list of arguments can be given to change specific parameters. Any number of parameters can be given, separated by spaces or commas. Arguments included in the command line are processed and the parameters changed to reflect these new operating characteristics. If no arguments are given, the present values for each parameter are displayed. Some parameters can be turned ON or OFF while others can be set to perform a specific function by changing their default values as desired. The default values are hexadecimal representations of ASCII code.

If TMODE is used in a procedure file, a standard output path (.1 or .2) must be specified. The SHELL's standard input path is redirected to the procedure file. TMODE is used only on devices that communicate through the Sequential Character File (SCF) Manager.

Table 18 lists the TMODE command parameters, definitions, and default values where needed.

NOTE

Refer to *Appendix D* for a complete listing of the ASCII code character set and its decimal and hexadecimal numeric values used by MiniCOP.

EXAMPLE:

OS9:TMODE upc lf null=4 pause	(upper case only, auto-line feed ON, 4 nulls after each carriage return, screen pause ON)
OS9:TMODE -echo	(echo OFF, input characters are not echoed back to the Terminal Emulator monitor)
OS9:TMODE .1 pag=24	(set page length to 24 lines at the standard output)
OS9:TMODE bsp=08 del=7F	(sets the backspace key to delete on character each time it is depressed; sets the delete key to delete the entire last line each time it is depressed)

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

TABLE 18 - TMODE PARAMETERS

Parameters	Note	Description
bsb	*1	Erase on backspace; backspace character echoed as a backspace-space-backspace sequence.
-bsb		No erase on backspace; echoes single backspace character only.
-bsl	*1	Backspace over line; lines are "detected" by sending backspace-space-backspace sequences until the line is erased. For use with video terminals only.
echo	*1	Input characters echoed (repeated) back to terminal.
-echo		Input characters echoed (repeated) back to terminal.
lf		Auto-line feed ON: linefeeds automatically echoed to terminal on input and output carriage returns.
-lf		Auto-line feed OFF.
pause		Screen pause ON; output is suspended upon full screen. Output resumed by pressing any key. (See "pag" parameter for screen size definition).
-pause	*1	Screen pause OFF.
upc		Upper case characters only. Lower-case characters are automatically converted to upper-case.
-upc	*1	Both upper-case and lower-case characters.
abort=h	*2	Set abort character to "h". Default =03.
bell=h	*2	Set bell (alert) character to "h". Default = 07.
bse=h	*2	Set output backspace character to "h". Default = 08.
bsp=h	*2	Set input backspace character to "h". Default =7F.
del=h	*2	Set input delete line character to "h". Default=1F.
dup=h	*2	Set duplicate last input line character to "h". Default =01.
eof=h	*2	Set end-of-file input character to "h". Default = 1D.
eor=h	*2	Set end-of-record (carriage return) input characters to "h". Default = 0D.
null=d	*3	Set null to "d" counts; Number of null characters ($00) transmitted after carriage returns for return delay. Default = 0
pag=d	*3	Set video page length to "d" lines. Used with screen pause. Default = 24 (see "pause" parameter for screen pause definition).
psc=h	*2	Set pause screen character to "h". Default =17.
quit = h	*2	Set Quit character to "h". Default = 05.
reprint=h	*2	Set reprint line character to "h".\ Default = 04.
type=h	*2	Set SC6551 initialization value to "h". Sets parity, word size, etc. Default=15.
xon=h	*2	Set DC1 resume output character to "h". Default =11.
xoff=h	*2	Set DC2 resume output character to "h". Default =13.

*1 = Default selection.
*2 = Value entered in hexadecimal.
*3 = Value entered in decimal.

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

UNLINK

REMOVE A MODULE FROM MAIN MEMORY — **Module Size: 1**
SYNTAX: UNLINK <modname> {<modname>} — **Running Size: 2**

NOTE
For additional information regarding this command, refer to the LINK command description.

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

VERIFY

VERIFY AND/OR UPDATE A MODULE HEADER AND CRC VALUE	Module Size: 2
SYNTAX: VERIFY [U] <modname> <modname>	Running Size: 6

The VERIFY command is used to verify that the module header parity and the CRC value of one or more modules is correct. Modules are read from the standard input path, and messages are sent to the standard error output path.

If the "U" (update) option is specified, the modules are copied to the standard output pathlist with the module's header parity and CRC replaced with the newly computed values. A message is displayed to indicate whether or not the module's original values match those computed by the VERIFY command.

If the "U" option is not specified, the module is not copied to the standard output to be updated. A message is displayed to indicate whether or not the module's header parity and CRC match the newly computed values.

EXAMPLE:

OS9:VERIFY /D0/PACK/primary (module header and CRC are verified not updated)

Module's header parity is correct.
CRC matches module's.

OS9:VERIFY <datafile >newdatafile (module header and CRC are verified and updated)

Module's header parity is correct.
CRC matches module's.

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

XMODE

EXAMINE OR CHANGE PORT OPERATING PARAMETERS	Module Size: 4
SYNTAX: XMODE <devname> [<arglist>]	Running Size: 3

The XMODE command is used to display or change the operating parameters of user-generated device descriptors created via the MAKDESC command. The parameters for original device descriptors (T1, T2, T3, T4, TP) may be displayed, but may not be changed with this command.

XMODE requires the user-created descriptor name be given (the name created by MAKDESC) and be located in the module directory. The descriptor name can be followed by a list of arguments. If no arguments are given, OS-9 will display the current values for each parameter. Otherwise the parameters given in the list are processed and changed accordingly. Any of the parameters listed in Table 19 may be altered. Any number of parameters may be given, separated by spaces or commas.

EXAMPLES:

OS9:XMODE /T2	(display parameters for port 2 descriptor, CANNOT be changed)
OS9:XMODE /TT	(display parameters for user created descriptor TT, CAN change)
OS9:XMODE /PRINTER pag=15 upc null=1	(change user-created descriptor to; page length = 15 lines, upper case characters only, 1 null after each carriage return)

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

TABLE 19 - XMODE PARAMETERS

Parameters	Note	Description
bsb		Erase on backspace; backspace character echoed as a backspace-space-backspace sequence.
-bsb		No erase on backspace; echoes single backspace character only.
bsl		Backspace over line: lines are "deleted" by sending backspace-space-backspace sequences until the line is erased. For use with video terminals only.
-bsl		Backspace over line; lines are "deleted" by sending backspace-space-backspace sequences until the line is erased. For use with video terminals only.
echo		Input characters echoed (repeated) back to terminal.
-echo		Input characters echoed (repeated) back to terminal.
lf		Auto-line feed ON: linefeeds automatically echoed to terminal on input and output carriage returns.
-lf		Auto-line feed OFF.
pause		Screen pause ON: output is suspended upon full screen. Output resumed by pressing any key. (See "pag" parameter for screen size definition)
-pause		Screen pause OFF.
upc		Upper-case characters only. Lower-case characters are automatically converted to upper-case.
-upc		Both upper-case and lower-case characters.
abort=h	*1	Set abort character to "h". Default =03.
bell=h	*1	Set bell (alert) character to "h". Default = 07.
bse=h	*1	Set output backspace character to "h". Default = 08.
bsp=h	*1	Set input backspace character to "h". Default = 7F.
del=h	*1	Set input delete line character to "h". Default = 1F.
dup=h	*1	Set duplicate last input line character to "h". Default = 01.
eof=h	*1	Set end-of-file input character to "h". Default = 1D.
eor=h	*1	Set end-of-record (carriage return) input character to "h". Default = 0D.
null=d	*2	Set null to "d" counts: Number of null characters ($00) transmitted after carriage returns for return delay. Default = 0.
pag=d	*2	Set video page length to "d" lines; used with screen pause;. default = 24 (see "pause" parameter for screen pause definition).
psc=h	*1	Set pause screen character to "h". Default = 17.
quit=h	*1	Set quit character to "h". Default = 05.
reprint=h	*1	Set reprint line character to "h". Default = 04.
type=h	*1	Set SC6551 initialization value to "h". Sets parity, word size, etc. Default = 15.
xon=h	*1	Set DC1 resume output character to "h". Default = 11.
xoff	*1	Set DC2 suspend output character to "h". Default = 13.

*1 = Value entered in hexadecimal.
*2 = Value entered in decimal.

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

MiniCOP/OS-9 TUTORIAL

INTRODUCTION

This tutorial is a step-by-step exploration of the basic operating characteristics and functionality of the 627 MiniCOP and the OS-9 Operating System. It is intended for use with a personal computer running the Terminal Emulator software.

This tutorial includes hands-on instruction on how to:

- Set-up the system (cabling, switch settings, etc.).
- Set the real-time clock.
- List the available main and COPRAM memories.
- Display the module directory.
- Create, display, and delete COPRAM directories.
- Transfer program modules and files between the Terminal Emulator and the MiniCOP.
- Transfer program modules between main memory and COPRAM.
- Load, link, and unlink program modules in main memory.

TUTORIAL REQUIREMENTS

SOFTWARE

To perform all of the operations in this tutorial, the following software is required:

- Terminal Emulator software (Model No. 626-6041*); and
- OS-9 System Commands disk V.R 1.1(Model No. 627-0006).

HARDWARE

To perform all of the operations in this tutorial, the following hardware is required:

- MS-DOS-based personal computer -or- 623 Loader/Terminal (Model No. 623-6100/-6150);
- 627 MiniCOP (Model No. 627-1002/-1003) with firmware V.R 1.2;
- 621 I/O rack**;
- 621 power supply module***; and
- Serial cables as required per personal computer or 623 Loader/Terminal.

---

* The Terminal Emulator software can be purchased separately using Model No. 626-6041. It is also included with the 623 MS-DOS Loader software and is factory installed on both models of the 623-61 Loader/Terminal (623-6100, 623-6150).

** Any 621 I/O (standard, augmented, full-size, or half-size) is acceptable. A 620 processor rack with I/O slots is also acceptable and the processor module is not required. (The MiniCOP uses PUSH/PULL instructions to communicate to a host processor via the I/O bus and is not typically installed in the 620-25 processor rack.)

*** Use a 620 power supply module if using a processor rack (processor half-size racks use 621 power supplies).

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

NOTE

This tutorial was written for use with the OS-9 System Commands disk V.R 1.1 and either the 627-1002 or 627-1003 MiniCOP using firmware V.R 1.2. This tutorial can also be used with previous releases of the OS-9 System Commands disk and firmware, but be advised that the information displayed on your Terminal Emulator screen may differ slightly.

DOCUMENTATION

The following documentation is also required:

- *627 MiniCOP and BASIC09 User Manual* (Form 627-8991);
- *623 Terminal Emulator User Manual* (Form 623-8989);
- *620 Installation and 621 I/O Communications User Manual* (Form 620-8986); and
- *620 Installation and 621 I/O Communications for 620-12/1633/36 Logic Controller Systems User Manual* (Form 620-8962).

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

MiniCOP SET-UP TUTORIAL

HARDWARE CONFIGURATION/INSTALLATION

The specific steps that are required to configure and install the hardware to be used in this tutorial will depend largely on your personal computer. The instructions in this tutorial, as given, are for use with a personal computer having two serial communications ports.

If the personal computer you are using for this tutorial has two standard RS-232 communications ports (COM1 and COM2) or one standard RS-232 communications port and a 620 Loader/Terminal interface board installed, follow the instructions as given.

NOTE

A personal computer with only one serial communications port can also be used, however the cable between the personal computer and the MiniCOP must be moved from port 1 to port 2 at various times. In this case follow these guidelines for moving the cable:

- Commands entered at the keyboard are transferred between the Terminal Emulator and the MiniCOP via the terminal port (MiniCOP port 1, T1), and
- Files are transferred via the tape port (MiniCOP port 2, TP).

The Terminal Emulators terminal and tape port configuration menus must identify the same personal computer communications port.

1. Determine the number and type of serial communications ports in your personal computer or 623-61 Loader/Terminal, including the MS-DOS Loader interface board. (Type: 9/25-pin, male/female, RS-232/RS-422)

2. Referencing the *MiniCOP Hardware* section of this manual, set the MiniCOP's DIP switches for the proper serial communications format used by each of your personal computer's serial ports or port. If needed, reference *Appendix B* to construct the appropriate serial cables. Refer to Figures 67, 68, and 69 for cabling connections.

3. Referencing the *MiniCOP Hardware* section, set the MiniCOP's DIP switches and jumper for the following configuration:

   - battery "on" (CPU board);
   - power up to OS-9 (enable the toggle switch -- CPU board); and
   - set the parallel/serial DIP switch (CPU board) according to the rack in which the MiniCOP is being installed†.

† Installing the MiniCOP into processor rack I/O slots is treated the same as installation in parallel I/O.

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

4. If the MiniCOP is to be installed in an augmented rack for this tutorial, remove the ribbon cable. If the MiniCOP is to be installed in a standard (nonaugmented) rack for this tutorial, install the ribbon cable (refer to the *MiniCOP Hardware* section).

5. Install the MiniCOP into any I/O slots in the selected rack. Also install the power supply module into its appropriate slot. **DO NOT APPLY POWER AT THIS TIME.**

CAUTION

When installing the 627-1002RC and 627-1003RC MiniCOPs in an augmented processor or I/O racks, certain considerations must be observed and noted.

As with the 627-1002 and 627-1003 MiniCOPs, caution must be exercised when removing either the 627-1002RC or 627-1003RC from an augmented rack. When a MiniCOP is installed in a standard rack, one function of the factory-installed ribbon cable is to provide battery backup to memory when power is removed from the unit or when the unit is removed from the rack. The ribbon cable, however, must be removed for the MiniCOP to be installed in an augmented rack. In this installation the upper/local bus provides a path for battery backup of memory should power be removed from the rack. If a 627-1002RC or 627-1003RC Hot Insertion/Removal MiniCOP is removed from an augmented rack, the link to backup power is broken and memory will be lost just as it is in the 627-1002 or 627-1003 MiniCOPs.

NOTE

Refer to the *620 Installation and 621 I/O Communications User Manual* regarding the location of rack I/O and power supply slots.

6. Connect the cables between the MiniCOP and the personal computer for your tutorial system configuration as shown in Figure 67, 68, or 69 and according to the MiniCOP ASYNC port DIP switch selections made in Step 2.

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

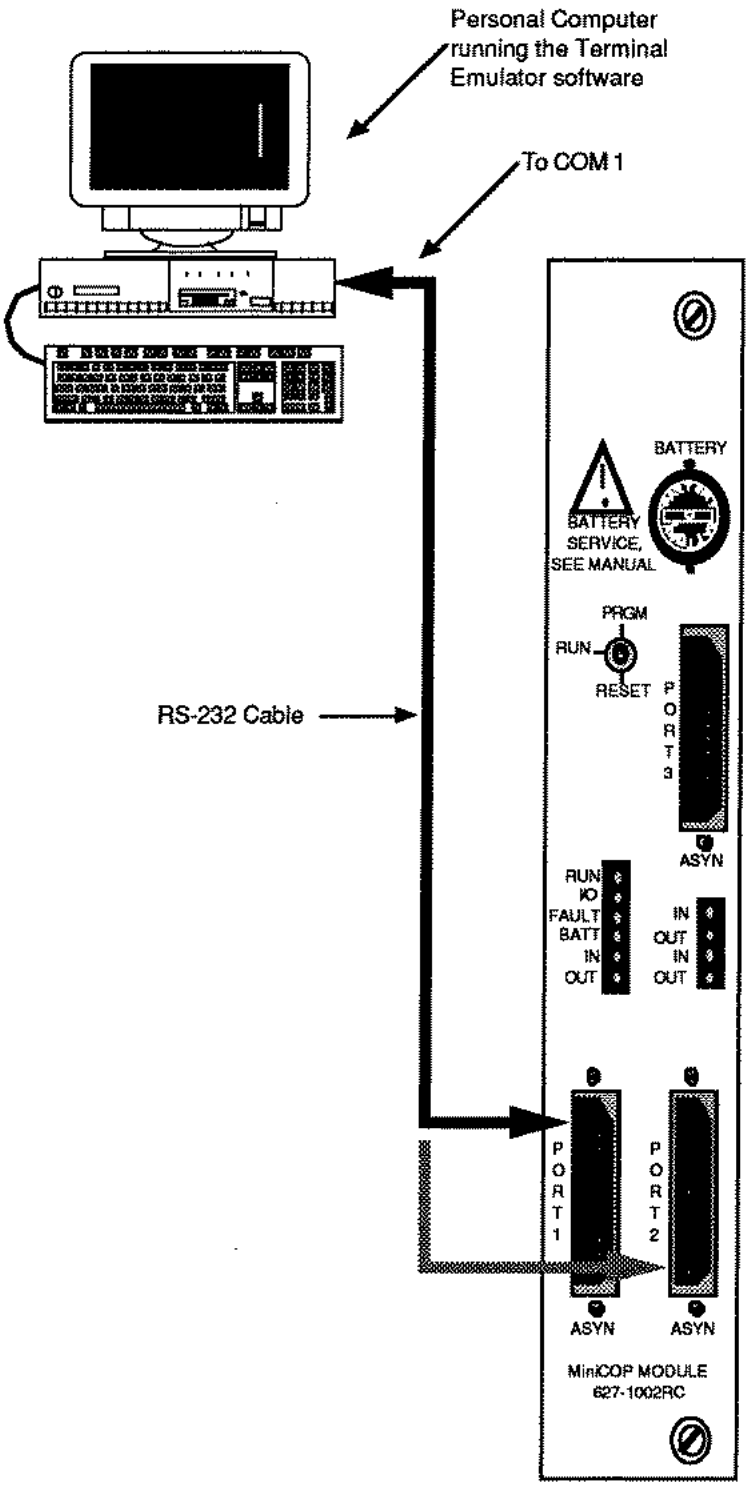


FIGURE 67 - USING ONE SERIAL COM PORT (COM1 – RS-232)

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

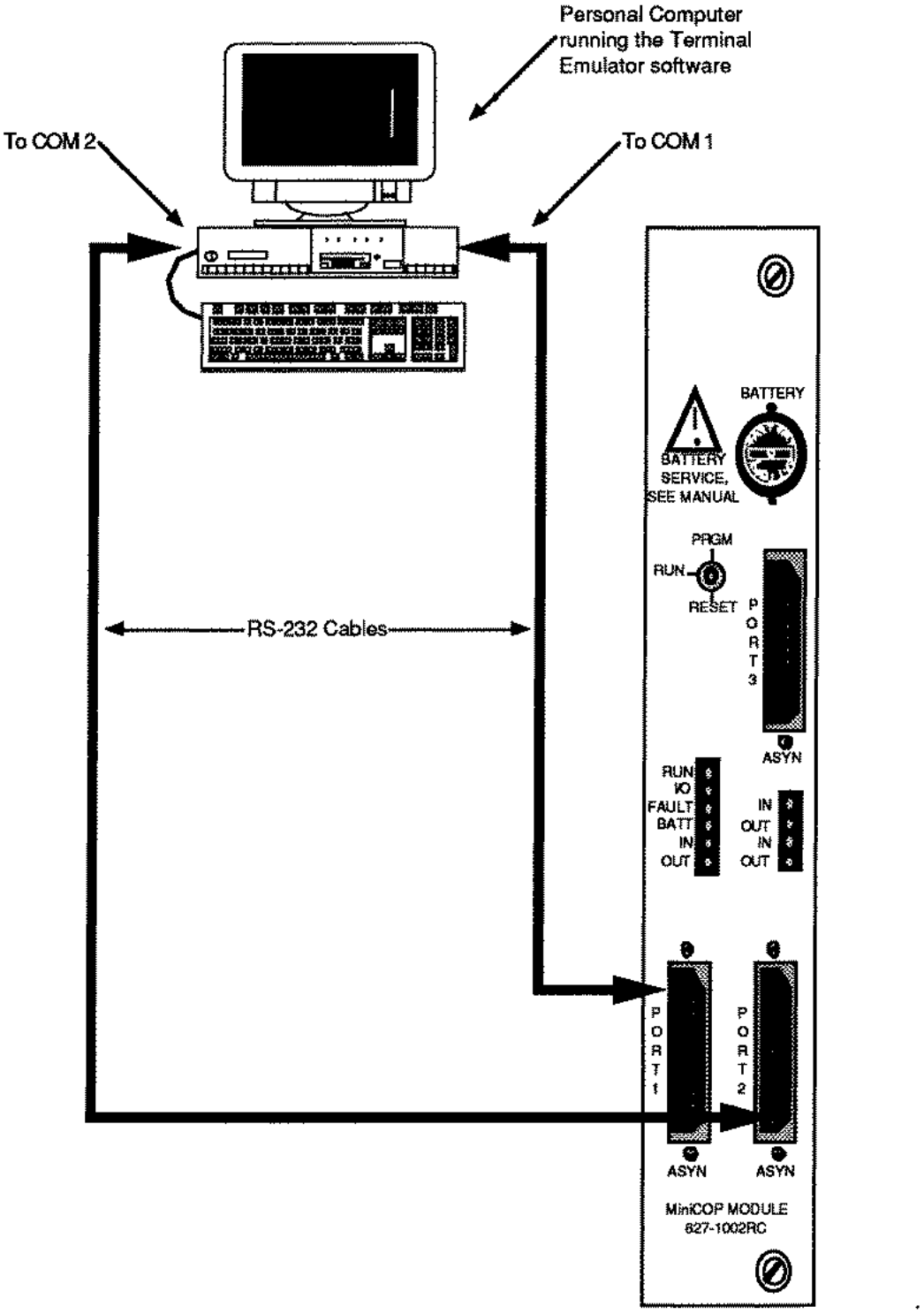


FIGURE 68 - USING TWO SERIAL COM PORTS (COM1 AND COM2 – RS-232)

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

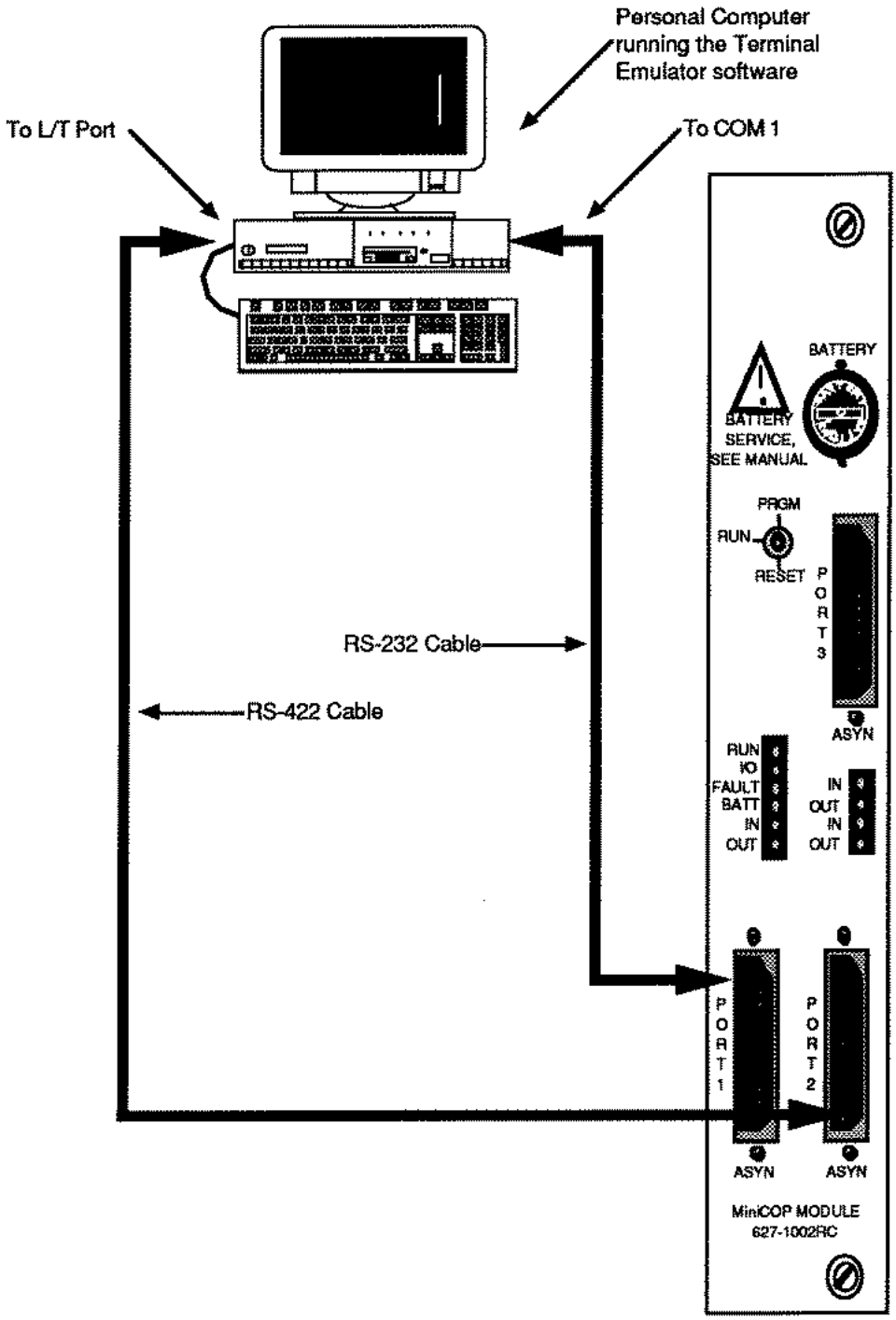


FIGURE 69 - USING LOADER/TERMINAL INTERFACE BOARD (RS-422/485) AS COM2

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

MiniCOP INITIALIZATION

Initializing the MiniCOP involves applying power to the system and transferring the OS-9 system commands from the Terminal Emulator to COPRAM. HELP is available for most Terminal Emulator functions by pressing [ALT-F10].

7. Turn on your personal computer or 623-61 Loader/Terminal and proceed to the Terminal Emulator screen.

   If using a personal computer:

   - Apply power to the computer and access the directory which contains the MS-DOS Loader or Terminal Emulator software package.
   - Type **TERMINAL** and then press **[RETURN].**
   - Press **[RETURN]** after reading the proprietary information screen. The Terminal Emulator screen will then display.

   If using the 623-6100 or 621-6150 Loader/Terminal:

   - Apply power to the Loader/Terminal. The main menu of the MS-DOS Loader software will appear in 10 to 20 seconds.
   - Press **[F7] Terminal Emulator**. This should display the Terminal Emulator screen.

8. Apply power to the MiniCOP.

   A **"Testing sector = XXXX"** message should appear where XXXX is a rapidly changing hexadecimal value. When the count reaches 03FF, the test is complete and the following message will be displayed:

   Please connect 623 Tape to Port TP for loading Boot Tape
   Tape Ready? Hit Return to Continue!

IMPORTANT

Do **NOT** press **[RETURN]** at this time. If you have already done so, reset the MiniCOP using the toggle switch or remove power for several seconds and then restore. The steps as listed in this tutorial will first prepare the Terminal Emulator to transfer files.

9. Press **[F1] TermCo** (terminal configuration). Verify the information provided and change any items if necessary. Press **[RETURN]** to accept the settings.

10. Press **[F2] TapeCo** (tape configuration). Verify the information provided and change any items if necessary. If using the MS-DOS Loader interface board as your second COM port, the word LOADER should be flashing under Tape Port. Press **[RETURN]** to accept the settings.

11. Insert the OS-9 System Commands Disk (Model No. 627-0006) into the Terminal Emulator's floppy drive.

12. Press **[F3] Send** to display the Send File to Tape Port selection. Using the [BACKSPACE] key erase the drive and path listed and change this information to the drive in which the OS-9 system commands disk was placed. For example, if the disk was placed into drive A, enter **A:**.

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

13. Press **[F5] Dir** to display a directory listing of the drive and/or path given in step 12. Use the arrow keys to move the highlight bar to either file **1002SYS** or **1003SYS**. 1002SYS pertains to the 627-1002 MiniCOP and 1003SYS pertains to the 627-1003 MiniCOP. Select the file name which corresponds to the MiniCOP.

14. Press **[RETURN]** twice . The Terminal Emulator will begin transferring the OS-9 Operating System to the MiniCOP and the screen will display the message **Tape Port Active.** In the bottom right-hand corner of the display a running count of the total number of bytes is displayed. The MiniCOP's tape port (port 2) IN and OUT LEDs should be flashing. This indicates that hand-shaking and data transfer activities are taking place.

    If for any reason the LEDs are not flashing, press **[F10] Abort Transmission** and review the following:

    - Terminal Emulator's terminal and tape configuration selections ([F1], [F2]),
    - cable connections,
    - cable construction,
    - MiniCOP port 1 and 2 DIP switch selections, and
    - personal computer serial COM ports (that is, either RS-232 or RS-422) to see if correctly identified.

    Repeat Steps 11 through 14.

15. When the OS-9 system commands have all been transferred to the MiniCOP (627-1002 running count = 79872 bytes; 627-1003 running count = 84992), the following messages and time prompt will appear:

    Load Complete

    627-10 Ver. 1.2

    TIME ?

16. Enter the current date and time using the following format (time entered in the 24 hour format):

    YY/MM/DD/HH/MM/SS [RETURN]

    The following message and OS-9 prompt will appear:

    Shell

    OS9:

    The OS9: prompt is the SHELL command line interpreter. The Terminal Emulator is a bridge between the two unique processors: the personal computer running a DOS operating system, and the MiniCOP running the OS-9 operating system. Selections made through the Terminal Emulator's main menu functions (F1 - F10) allow you to communicate with the personal computer. Entries made at the OS9: prompt (command line) or subsequent BASIC09 prompts (B:, E:, D:) allow you to communicate with the MiniCOP.

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

The MiniCOP is now initialized. During initialization the MiniCOP's memory areas were prepared for operation. In main memory, the Module Directory now contains the modules shown in Figure 70.

```
module directory at 14:35:08

Boottp    TLoad     RBF       D0        Disk
Firq      Video     TP        T1        T2
T3        T4        SC6551    SCF       IOMan
Clock     Sysgo     OS9p3     OS9p1     Init
OS9p1     Shell     Mdir
```

FIGURE 70 - MODULE DIRECTORY LISTING

OS-9 uses two directory pointers: execution directory and data directory. After initialization the COPRAM directory structure appears as in Figure 71. CMDS is the current execution directory and the root directory (/D0) is the current data directory. Up to three additional files may also appear in the root directory: DATACONTROL is available in both the 627-1002 and 627-1003 while VideoDemo and conf1003 are available in only the 627-1003 MiniCOP. The 627-1002 MiniCOP is initialized using 1002SYS and the 627-1003 MiniCOP is initialized using 1003SYS, both from the OS-9 System Commands disk V.R 1.1.

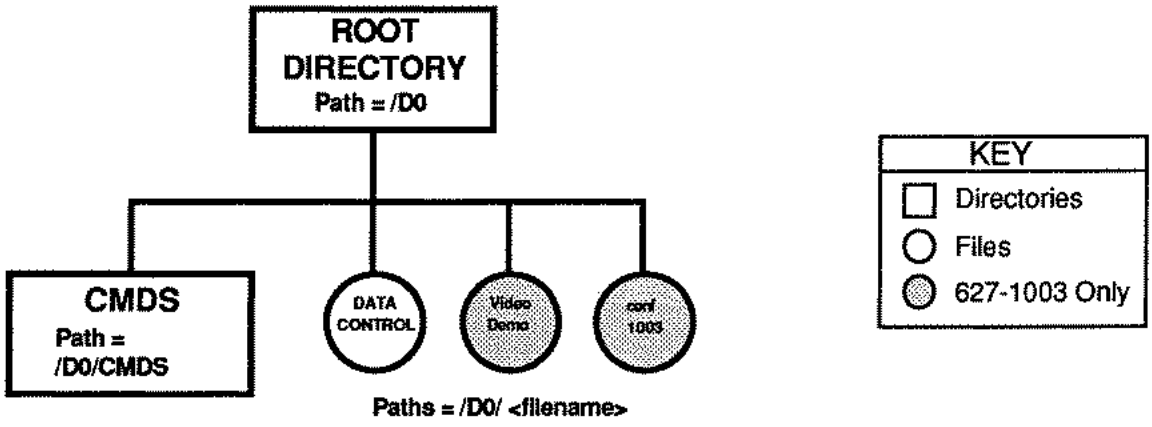
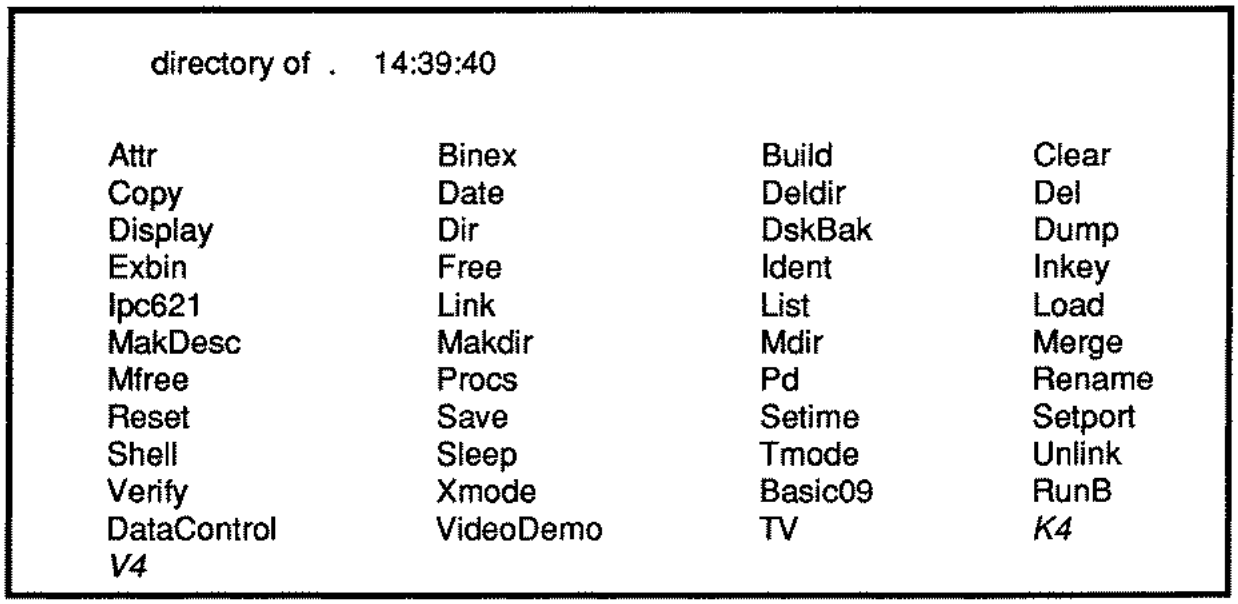


FIGURE 71 - COPRAM STRUCTURE AFTER INITIALIZATION

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

The execution directory contains the OS-9 system commands and four other items: BASIC09, RUNB, DataControl, and VideoDemo as illustrated in Figure 72.



Italicized file names are available for use with the 627-1003 MiniCOP when initialized using the file 1003SYS.

FIGURE 72 - EXECUTION DIRECTORY LISTING

Depending on the particular application, not all of the commands may be needed. To acquire more COPRAM memory, the unused or infrequently used modules can be deleted from the execution directory. Later, if necessary, the specific deleted module or modules can be copied from the OS-9 systems commands disk back to the MiniCOP, then used and deleted again, as needed.

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

OS-9 FUNDAMENTAL OPERATIONS TUTORIAL

GENERAL SYSTEM OPERATIONS

In this section of the tutorial you will exercise several OS-9 commands that will allow you to control certain system operating characteristics.

17. Display the current system date using the DATE command. This command should display the date that you entered in Step 16.

    OS9:DATE [RETURN]

18. Display the current data and time by including the "t" option with the DATE command.

    OS9:DATE t [RETURN]

19. Set or change the current date and time to 7:45:00PM June 27, 1991 using the SETIME command.

    OS9:SETIME 91/06/27/19/45/00 [RETURN]

    OS9:DATE t [RETURN]
    June 27, 1991 19:45:02

    When the **[RETURN]** key is pressed, the MiniCOP's real-time clock is immediately updated. The DATE t command reads the current date and time from the real-time clock. The time difference between the value entered and the value displayed corresponds to the delay in execution between the two commands.

    Reset the MiniCOP's real-time clock to the current date and time.

    OS9:SETIME [Current date and time]

    OS9:DATE t
    [Current date and time]

20. Change the operating parameters of the Terminal Emulator using the TMODE command.

NOTES
The TMODE command allows you to customize the Terminal Emulator. In this step you will use TMODE to allow the [BACKSPACE] key to delete one character at a time and the [DELETE] key to delete an entire line. You will also request that the screen display scroll pause when full (after a pause, pressing any key will resume the scroll). Also, the reset operation, which was generated by either the OS-9 RESET command or the MiniCOP's toggle switch, will remove the current TMODE parameters. The TMODE command and desired parameters can be included in a procedure file created with the BUILD command to provide for ease of terminal configuration.

OS9:TMODE bsp=08 del=7F pause [RETURN]

Type a random string of characters on the command line. Press the **[BACKSPACE]** key several times and observe. Press the **[DELETE]** key once and observe.

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

21. Display a listing of the MiniCOP main memory's newly-created module directory with the MDIR command. Compare this listing with Figure 70.

OS9:MDIR [RETURN]

22. To obtain a more complete listing of the memory module characteristics (name, use, attributes, revision, type, size, and address) use the "e' option.

OS9:MDIR e [RETURN]

Because the pause parameter was included in the TMODE command issued in Step 20, the screen scroll stopped when the screen was full. You can resume by pressing any key. If the pause had not been turned on, the entire listing would have scrolled by. The scroll can also be stopped at any time by pressing **[CONTROL-W]**, which is the control character command.

23. The amount of available (free) main memory is displayed in pages with the MFREE command. A page is defined as a 256-byte block of memory. Compare this listing with Figure 73.

OS9:MFREE [RETURN]

Address	pages
E00-B1FF	163
B900-B9FF	1

Total pages free = 164

FIGURE 73 - AVAILABLE MAIN MEMORY EXAMPLE

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

24. Display a listing of the COPRAM root directory (/D0) with the DIR command. Compare this listing with Figure 74.

OS9:DIR /D0 [RETURN]

```
directory of .  14:37:52
CMDS  DATACONTROL  VideoDemo  conf1003
```

Italicized file names are available for use with the 627-1003 MiniCOP when initialized using the file 1003SYS.

FIGURE 74 - ROOT DIRECTORY (D0) LISTING

25. Display a listing of the current data directory. Entering the DIR command with no parameters or pathlist will display the current data directory.

OS9:DIR [RETURN]

Since OS-9 attaches the current data directory pointer to the root directory during initialization, the contents of the current data directory now displayed are identical to the contents of the root directory displayed in Figure 74.

26. Display the COPRAM memory characteristics with the FREE command. Compare this listing with Figure 75. A sector is defined as a 256-byte block of contiguous COPRAM memory.

OS9:FREE [RETURN]

Using the 627-1002:

```
"RAM Disk" created on: 88/01/28
Capacity: 1,024 sectors (1-sector clusters)
712 Free sectors, largest block 512 sectors
```

Using the 627-1003:

```
"RAM Disk" created on: 88/01/28
Capacity: 1,024 sectors (1-sector clusters)
692 Free sectors, largest block 493 sectors
```

FIGURE 75 - AVAILABLE COPRAM MEMORY AND CHARACTERISTICS

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

27. Create a new directory called NEWDIR using the MAKDIR command.

OS9:MAKDIR NEWDIR [RETURN]

When a directory is created without specifying a pathlist, OS-9 will place that directory under the current data directory as a default. You can display a listing of the new data directory using the DIR command. Compare this new listing of the data directory with Figure 74 and the new directory structure illustrated in Figure 76 with Figure 71.

OS9:DIR [RETURN]

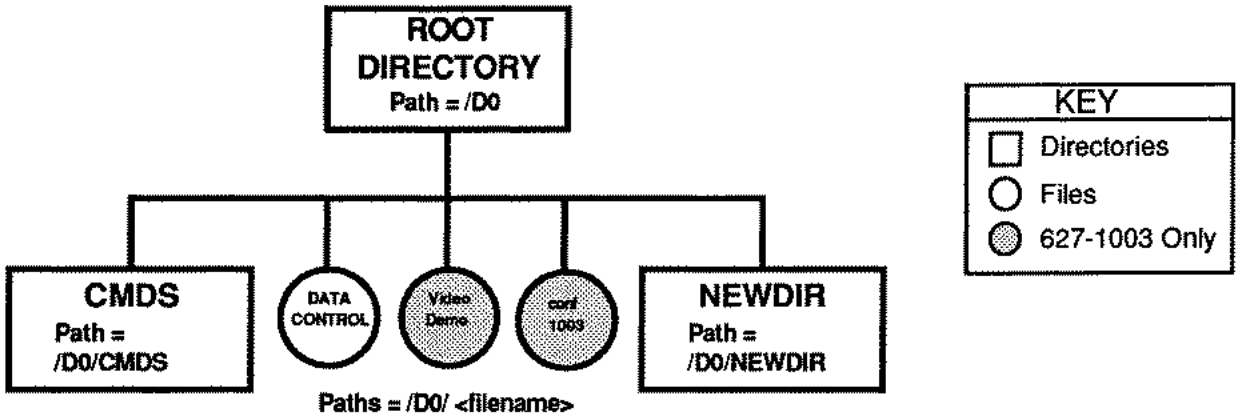


FIGURE 76 - COPRAM STRUCTURE WITH "NEWDIR" DIRECTORY ADDED

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

28. Change or move the current directory pointer using the CHD command. The directory structure is unaffected, except that the data directory pointer will now be attached to NEWDIR.

OS9:CHD /D0/NEWDIR [RETURN]

29. The pathlists to the data and execution directories can be displayed using the PD command. The "**d**" option displays the data directory pathlist and the "**x**" option the execution directory pathlist. These commands are particularly helpful when working with complex directory structures.

OS9:PD d [RETURN]
/D0/NEWDIR

OS9:PD x [RETURN]
/D0/CMDS

30. The "**e**" option can be combined with the DIR command to display a complete listing of any COPRAM directory. Compare this listing with Figure 77.

OS9:DIR /D0 e [RETURN]

Using the 627-1002:

Owner	Last Modified	attributes	sector	bytecount	name
0	57/02/80 6411	d-ewrewr	5	5	CMDS
0	88/03/12 1701	----r-wr	12B	AAE	DATACONTROL
0	91/12/18 0913	d-ewrewr	137	40	NEWDIR

Using the 627-1003:

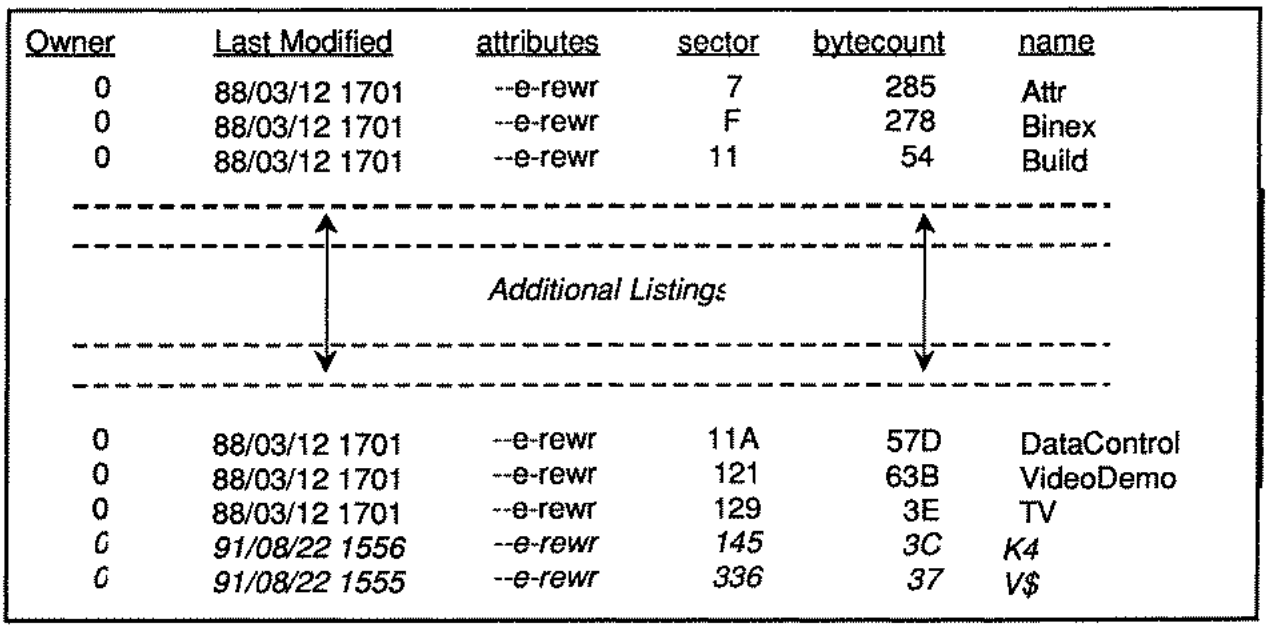
Owner	Last Modified	attributes	sector	bytecount	name
0	57/02/80 6354	d-ewrewr	5	5A0	CMDS
0	88/03/12 1701	----r-wr	12B	AAE	DATACONTROL
0	88/03/12 1701	------wr	137	CAC	VideoDemo
0	57/02/80 6357	----r-wr	1471	12	conf1003
0	99/07/22 1106	d-ewrewr	45	40	NEWDIR

FIGURE 77 - COPRAM DIRECTORY LISTING FOR /DO DIRECTORY

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

Compare this listing to Figure 78.

OS9:DIR /D0/CMDS e [RETURN]



Owner	Last Modified	attributes	sector	bytecount	name
0	88/03/12 1701	--e-rewr	7	285	Attr
0	88/03/12 1701	--e-rewr	F	278	Binex
0	88/03/12 1701	--e-rewr	11	54	Build
		Additional Listings			
0	88/03/12 1701	--e-rewr	11A	57D	DataControl
0	88/03/12 1701	--e-rewr	121	63B	VideoDemo
0	88/03/12 1701	--e-rewr	129	3E	TV
0	*91/08/22 1556*	*--e-rewr*	*145*	*3C*	*K4*
0	*91/08/22 1555*	*--e-rewr*	*336*	*37*	*V$*

Italicized file names are available for use with the 627-1003 MiniCOP when initialized using the file 1003SYS.

FIGURE 78 - COPRAM DIRECTORY LISTING FOR SUBDIRECTORY "CMDS"

The "**e**" option can also be combined with the DIR command and the "**x**" option to provide a complete listing of the execution directory rather than typing in its specific pathlist. Compare this listing to Figure 78.

OS9:DIR x e [RETURN]

If the DIR command and the "**e** " option is given with no pathlist, OS-9 defaults to the current data directory. This list will be blank since the current data directory is NEWDIR, which is empty.

OS9:DIR e [RETURN]

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

31. Create a text file using the BUILD command. Text files contain lines of ASCII characters that can be output to display or print specific messages. They are created with the BUILD command and are displayed with the LIST command.

```
OS9:BUILD textfile
? Text files can be used to output messages to the
? Terminal Emulator screen or other output devices
? connected to the MiniCOP's ports, such as a printer or
? video monitor. Each line can be up to 80 characters
? in length with no limit to the number of lines (within
? the MiniCOP's COPRAM memory resource limitation).
? The LIST command is used to display Text Files. They
? can be called from one or more of the BASIC09
? programs that you may be using to control or monitor
? your application. Pressing the [RETURN] key on a blank
? line terminates the BUILD command. If no pathlist is
? provided, OS-9 will create or save the file in the current
? data directory.
? [RETURN]
```

NOTE

Entering **[RETURN]** as the first character following a prompt ("?") signals OS-9 that the end-of-file has been reached.

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

32. Use the DIR command to list the contents of the current data directory and compare this listing to Figure 79. Then use the LIST command to display the contents of the text file.

OS9:DIR [RETURN]

OS9:LIST textfile [RETURN]

```
OS9: LIST textfile

Text files can be used to output messages to the Terminal
Emulator screen on other output devices connected to the
MiniCOP's ports, such as a printer or video monitor. Each
line can be up to 80 characters in length with no limit to
the number of lines (within the MiniCOP's COPRAM
memory resource limitation). The LIST command is used
to display Text Files. They can be called from one or more
of the BASIC09 programs that you may be using to control
or monitor your application. Pressing the [RETURN] key
on a blank line terminates the BUILD command. If no
pathlist is provided, OS-9 will create or save the file in the
current data directory.


OS9:
```

FIGURE 79 - LISTED TEXTFILE

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

33. Create a SHELL procedure file using the BUILD command. A SHELL procedure file is actually a form of text file. Each line in the file will start with an OS-9 command which causes the SHELL to interpret the text as if it were entered on the SHELL command line. SHELL procedure files do not typically contain text to be displayed as a message. They often will contain a line which will call another text file that does contain a message using the LIST command.

OS9:BUILD procfile
? PD x
? PD d
? DIR x
? DIR
? DIR /D0
? LIST textfile
? [RETURN]

Next you will display the new contents of the current data directory and then enter this file's name on the command line. OS-9 interprets this action as a request to execute the named file. When you press the key, watch the Terminal Emulator display carefully. You should see the result of each line in the file's execution. If necessary, execute the file again.

OS9:DIR [RETURN]

OS9:procfile [RETURN]

34. Change the name of an existing file using the RENAME command.

OS9:RENAME textfile message.01 [RETURN]

OS9:DIR [RETURN]

The RENAME command changes the name of the first file listed to the second. It is possible to change the name of any COPRAM file (including directory names) as long as it has not been write protected.

35. Using the COPY command, you can copy a file from one directory to another. In this exercise we will make a copy of **procfile** and place the copy into the root directory (/D0). Then you will use the DIR command to observe the contents of the current data directory (NEWDIR) and the root. You will notice that the original copy still remains in the data directory.

OS9:COPY procfile /D0/procfile [RETURN]

OS9:DIR [RETURN]

OS9:DIR /D0 [RETURN]

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

36. To delete a file (excluding directories and write protected files), use the DEL command. You can specify a file anywhere in COPRAM by including its pathlist if the target does not reside in the current data directory. For example, COPRAM now contains two copies of the file **procfile**; one in the root directory and one in NEWDIR. If you want to delete the copy in the root, you could use the CHD command to make the root your current data directory and then simply enter **[DEL procfile [RETURN]]** on the command line. Or you can specify the file in the root by including it's pathlist as a command parameter.

OS9:DEL /D0/procfile [RETURN]

Once this is complete you can again list the contents of these two directories and observe the change which has occurred.

OS9:DIR [RETURN]

OS9:DIR /D0 [RETURN]

37. To delete a directory use the DELDIR command. In this exercise you will be deleting the directory NEWDIR that you created earlier. The directory to be deleted does not need to be empty in order to be deleted, but all files the directory holds will also be removed. To execute this command, the target directory cannot be the current data directory or any of its parents. Since NEWDIR is the current data directory, you must first change directories.

OS9:CHD /D0 [RETURN]

OS9:DELDIR NEWDIR [RETURN]

Deleting directory file.
List directory, delete directory, or quit? (l/d/q)

When the DELDIR command is executed, a message and prompt is displayed offering three options: delete, list, or quit.

Selecting "**q**" aborts the deletion operation and generates Error CODE #001 - Unconditional Abort.

Selecting "**l**" displays a complete listing of all files in the target directory (as if the DIR e command was executed) and then asks if the delete operation should: Continue (y/n) -- yes (**y**) deletes all files and the directory while no (**n**) aborts the operation.

Selecting "**d**" executes the delete operation without listing the directory's contents.

Deleting directory file.
List directory, delete directory, or quit? (l/d/q) d [RETURN]

If you now execute the DIR command, you can observe that NEWDIR is no longer in COPRAM.

OS9:DIR [RETURN]

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

LOADING, LINKING, AND SAVING PROGRAM MODULES

Once the MiniCOP has been programmed for its intended application it may be desirable to delete any program modules (OS-9 commands, etc.) from COPRAM that will not normally be used. This will free additional COPRAM memory for use by the application programs. If, at a later date, certain deleted commands become needed due to a change in programming or for use in general system maintenance and housekeeping, they can be reloaded. Once they are no longer required, they or any other program module can be deleted.

CAUTION

Care must be taken when deleting OS-9 program modules from the execution directory. Be certain that these commands are not being called by any portion of the application program to prevent any inadvertent program deletion. Also note that some program modules call others in the course of their execution.

Assume that the program modules ATTR, BASIC09, BINEX, EXBIN, RENAME, and TV will not be needed by the application that is to be run on the MiniCOP. We can delete these six items from the CMDS directory using the DEL command. To delete files in the execution directory, there are two basic methods that can be used.

Method 1

In the first method, there are two basic options. First the entire pathlist for each file can be included with the DEL command (for example, DEL /D0/CMDS/attr). Second the "-x" option can be used in place of the pathlist (for example, DEL attr -x). Since only one file at a time can be deleted from the execution directory per DEL command, both methods must be repeated for each file. There is also the additional option of deleting one file at a time from separate command lines or using the sequential execution modifier (;) to list all commands on the same command line.

Method 2

In the second method, you can temporarily change the current data directory to the current execution directory. Here multiple files, separated by spaces, can be deleted with one DEL command. After deleting the files, return the current data directory pointer to the proper directory.

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

In this exercise you will be using Method 2. You will also use several other commands to observe the contents of the results of your actions.

38. Display the available (free) memory in COPRAM. Compare this listing to Figure 80.

OS9:FREE [RETURN]

Using the 627-1002:

"RAM Disk" created on: 88/01/28
Capacity: 1,024 sectors (1-sector clusters)
712 Free sectors, largest block 512 sectors

Using the 627-1003:

"RAM Disk" created on: 88/01/28
Capacity: 1,024 sectors (1-sector clusters)
692 Free sectors, largest block 493 sectors

FIGURE 80 - AVAILABLE COPRAM MEMORY AND CHARACTERISTICS

39. Display the contents of the current execution directory, then move the data directory pointer to /D0/CMDS (current execution directory) and display the current directory's pathlists.

OS9:DIR x [RETURN]

OS9:CHD /D0/CMDS

OS9:PD d [RETURN]
/D0/CMDS

OS9:PD x [RETURN]
/D0/CMDS

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

40. Delete the six program modules.

OS9:DEL attr basic09 binex exbin rename tv [RETURN]

41. Change the data directory back to the root (/D0) and display the current directory's pathlists.

OS9:CHD /D0 [RETURN]

OS9:PD d [RETURN]
/D0

OS9:PD x [RETURN]
/D0/CMDS [RETURN]

42. Observe that in the listing created by this next command the six files that were specified are gone. Also note that in the following command the amount of free COPRAM memory has increased by 108 sectors or 27648 bytes (108 x 256 = 27648).

OS9:DIR x [RETURN]

OS9:FREE [RETURN]

Most of this additional free memory was the result of deleting BASIC09, which requires about 23K bytes. Also note that RUNB still resides in COPRAM. This is a version of BASIC09 that has been stripped of practically all data that is nonessential to executing packed BASIC09 procedures.

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

Now assume that the program modules ATTR, BASIC09, and RENAME are again needed in COPRAM either permanently due to a change being implemented in the application or for temporary use in system maintenance or housekeeping. We can transfer these three items from the OS-9 system commands disk in the Terminal Emulator to the MiniCOP using the TLOAD command. TLOAD transfer files received from the Terminal Emulator into the MiniCOP's module directory.

In this exercise you will also be using several other commands in addition to TLOAD.

43. Clear the tape port's buffer of any extraneous characters.

OS9:CLEAR /TP [RETURN]

44. Key in the TLOAD command, but **DO NOT PRESS the [RETURN] key.** Although doing so will not cause any problems, the procedures in step 45 assume that you have not done so.

OS9:TLOAD /TP

45. Prepare the Terminal Emulator to transfer the first file using the following procedure:

    - Insert the OS-9 system commands disk into a floppy drive. Press **[F3] Send** and change the displayed prompt to the drive that holds the commands disk. For example, if the commands disk is in floppy drive "**A**", use the **[BACKSPACE]** to erase the displayed prompt and enter **A:**.

    - Press **[F7] Dir**. A listing of the files contained on the OS-9 system commands disk is displayed.

    - Use the arrow keys to move the highlight bar to the file named ATTR.

    - Press **[RETURN]** twice.

    The message "Tape port active" appears and a running count of the bytes of information being transferred from the Terminal Emulator to the MiniCOP is displayed. The IN LED corresponding to port 2 (tape port) on the MiniCOP should be lit indicating the file transfer is taking place.

    Once the transfer is complete, the Terminal Emulator cursor appears beneath the last OS9: prompt. At this point the TLOAD command is still active and is waiting for either the Terminal Emulator to provide additional files or for a signal to exit the currently running program (TLOAD).

46. Repeat the procedure in step 45 for BASIC09 and RENAME. When this is completed, proceed to step 47.

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

47. The control character command ([CONTROL-E]) will exit the currently running program (TLOAD). This will produce Error Code #002 - Keyboard Abort to indicate the action which took place and will also return the OS9: prompt.

48. The three files that were just transferred to the MiniCOP via TLOAD now reside in the module directory. Verify this by using the MDIR command. Also note that they have not been transferred to COPRAM.

OS9:MDIR [RETURN]

OS9:DIR [RETURN]

OS9:DIR x [RETURN]

49. The SAVE command can be used to transfer a copy of these three files from the module directory to the specified COPRAM location. Since the SAVE command defaults to the current execution directory (which is /D0/CMDS) it will not be necessary to specify a pathlist.

OS9:SAVE attr [RETURN]

OS9:SAVE basic09 [RETURN]

OS9:SAVE rename [RETURN]

50. Observe that in the listing created by this next command, the three files again reside in the execution directory. Also note that in the following command the amount of free COPRAM memory has decreased by 99 sectors or 25344 byte (99 x 256 = 275344).

OS9:DIR x [RETURN]

OS9:FREE [RETURN]

ATTR, BASIC09, and RENAME also still reside in the module directory. When a program module is no longer needed in the module directory, it can be removed using the UNLINK command.

51. Display a listing of the module directory before and after unlinking the three program modules.

OS9:MDIR [RETURN]

OS9:UNLINK attr basic09 rename [RETURN]

OS9:MDIR [RETURN]

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

It is also possible to load and link program modules and packed BASIC09 procedures from COPRAM into the module directory. In fact, anytime a program module is executed a copy is loaded and linked into the module directory. When it has completed executing it will be unlinked and removed, so long as no other operations are using it. In this next exercise you will use the LOAD command to load and link a program module into the module directory. Because the module is being loaded and not executed it will remain in the module directory until removed with the LINK command.

52. Load the program module DUMP into the module directory. Continue by displaying a listing of both the module and execution directories.

OS9:LOAD dump [RETURN]

OS9:MDIR [RETURN]

OS9:DIR x [RETURN]

53. Remove the program module DUMP from the module directory and display the contents of the module directory. Observe that DUMP has been removed.

OS9:UNLINK dump [RETURN]

OS9:MDIR [RETURN]

Each time a program module or packed BASIC09 procedure is loaded its link count is incremented by one. If the same module is needed by two concurrently executing procedures, the link count will be incremented by one for both. In this manner the same module or procedure is used rather than loading a new copy for each calling procedure. If one calling procedure no longer requires the module or procedure, it will be unlinked and its link count decremented by one.

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

54. Load the program module SLEEP three times. Each is loaded and linked incrementing the link count to "3". Display the module directory and observe that SLEEP has been loaded.

OS9:LOAD sleep [RETURN]

OS9:LOAD sleep [RETURN]

OS9:LOAD sleep [RETURN]

OS9:MDIR [RETURN]

55. To remove SLEEP from the Module Directory it must be unlinked three times.

OS9:UNLINK sleep [RETURN]

OS9:MDIR [RETURN]

OS9:UNLINK sleep [RETURN]

OS9:MDIR [RETURN]

OS9:UNLINK sleep [RETURN]

OS9:MDIR [RETURN]

After the third UNLINK command is issued, the link count is decremented to zero and SLEEP is removed from the module directory.

56. Load the program module DISPLAY four times using one LOAD command. Each is loaded and linked sequentially incrementing the link count to four. Display the module directory and observe that SLEEP has been loaded.

OS9:LOAD display display display [RETURN]

OS9:MDIR [RETURN]

57. To remove DISPLAY from the module directory, it must be unlinked four times. Use one UNLINK command to remove DISPLAY. Display the module directory and observe that DISPLAY has been removed.

OS9:UNLINK display display display [RETURN]

OS9:MDIR [RETURN]

CONCLUSION

By completing this tutorial you have gained a hands-on familiarity with the basic operating characteristics and functionality of the 627 MiniCOP and the OS-9 Operating System. Though not all functions and characteristics have been practiced, you should be able to explore these operations on your own.

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

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

Section 4.
MiniCOP Interfacing

About This Section

This section presents information on providing an interface between Honeywell 620 Platform Controllers and intelligent factory floor devices, such as printers, barcode readers, LED displays, CRT terminals, and practically any devices that can communicate via the RS-232, RS-422/485, or 20mA electrical formats, including programmable logic controllers from other vendors (for example, Allen-Bradley and Modicon PLCs).

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

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

MiniCOP INTERFACING

INTERFACING OVERVIEW

The Honeywell 627 MiniCOP Modules (Models 627-1002 and 627-1003) are industrial rated microprocessors. Typically they are utilized to provide an interface between Honeywell 620 Platform Controllers and intelligent factory floor devices, including programmable logic controllers from other vendors. The MiniCOP can be interfaced with printers, barcode readers, LED displays, CRT terminals, or practically any device that can communicate via the RS-232, RS-422/485, or 20mA electrical formats. It can be used to collect, monitor, store, manipulate, and display data retrieved from these connected devices.

Using BASIC09 or the OS-9 Assembly Language, you can develop and implement your own programs on the MiniCOP. Personality procedures are also available for use with the MiniCOP.

The personality procedures are prepackaged software drivers used to simplify the task of implementing a control scheme where the MiniCOP is to interface, via it's ASYNC ports, to a 620 Platform Controller, Allen-Bradley PLC, or Modicon PLC.

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

MiniCOP TO HOST 620 PLATFORM CONTROLLER

The term "host" simply means that the MiniCOP is installed in the 620 Platform Controller's I/O slots. The MiniCOP is an eight point (address) module that is installed in the host's I/O slots. These slots may be in a parallel or serial I/O rack or a processor rack. The location chosen must be addressable from the host as real I/O points and be capable of using the PUSH/PULL interface. Thus, the MiniCOP can communicate with a host 620 Platform Controller (and vice versa) via the host's I/O bus.

MiniCOP DATA TABLE

In addition to the MiniCOP's main and COPRAM memories, it also has a temporary storage area called the data table. Both the MiniCOP and the host 620 Platform Controller can access this area. Using ladder logic, the 620 Platform Controller is able to read from or write to the data table. The MiniCOP uses a system utility called the IPC621 driver to read from or write to the data table. In this manner the data table is used as a temporary storage area or mail box which either device can access and thereby communicate with the other.

The MiniCOP and the host 620 Platform Controller "view" the data table in a slightly different manner. It should be noted however that there is **ONLY ONE DATA TABLE** which both devices can access. The data table can be "viewed" as 512 16-bit registers or 8192 single-bit registers as illustrated in Figure 81.

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

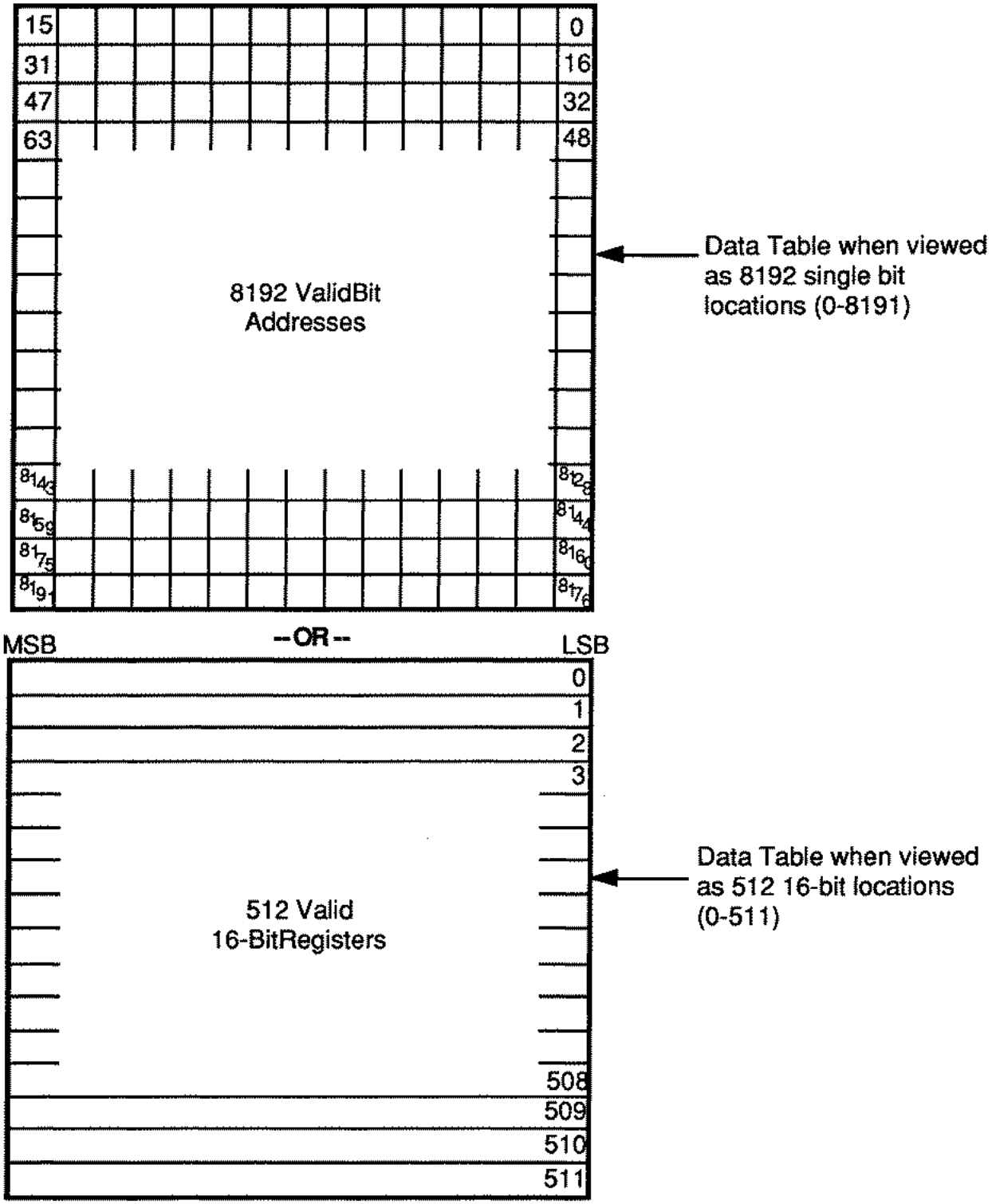


FIGURE 81 - MINICOP DATA TABLE

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

Changing any single-bit also affects the 16-bit register location in which the single-bit resides (for example, single-bit locations 0 through 15 make up the 16-bit register location 0). For example, assume that a value of one (1) has been written into the single-bit location 2 and single-bit locations 0 and 1, and 3 through 15 all hold zeros (0). Since the 16-bit register location 0 and the single bit locations 0 through 15 occupy the same space you would read a value of 4 when viewing these same bits as a 16-bit register location. See Figure 82.

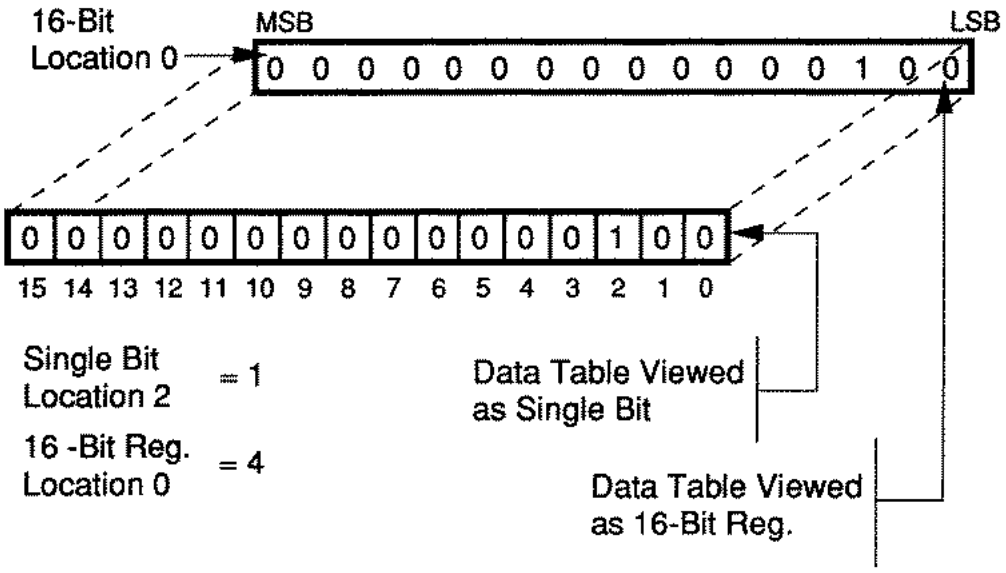


FIGURE 82 - CHANGING A SINGLE BIT

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

Changing the contents of any 16-bit register also affects the single-bit locations that reside in those locations (for example, the 16-bit register location 511 is made up of the single-bit locations 8176 through 8191). For example, assume that the 16-bit binary equivalent of the decimal value 8 has been written into the 16-bit location 511. Single-bit location 8179 would now equal 1 while single-bit locations 8176 through 8178, and 8180 though 8191 now equal zero regardless of what their previous contents were. See Figure 83.

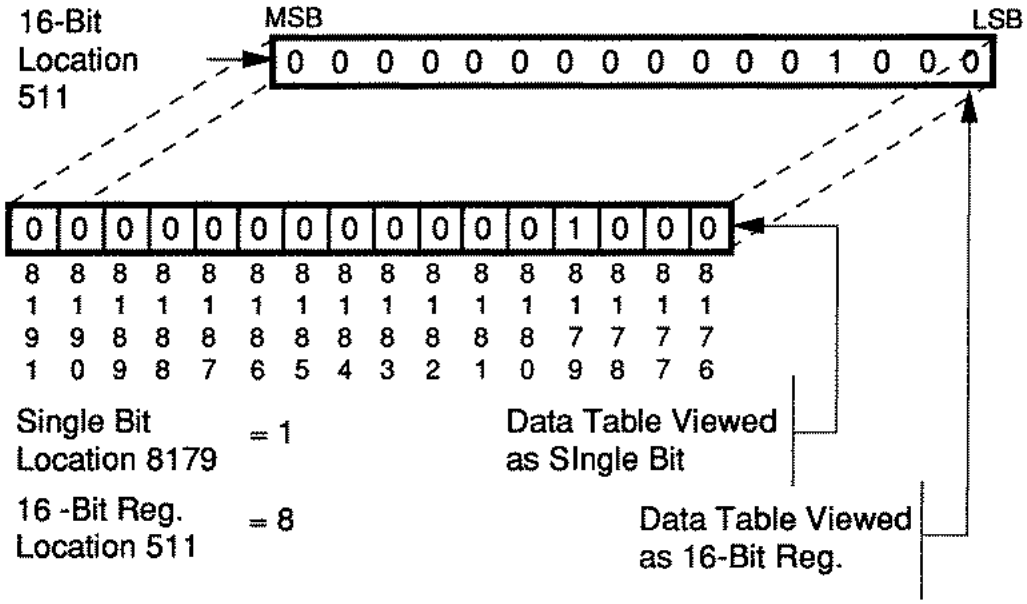


FIGURE 83 - CHANGING A 16-BIT REGISTER

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

MiniCOP ACCESS TO THE DATA TABLE (IPC621 DRIVER UTILITY)

The MiniCOP can view the data table both as 511 16-bit register locations or 8192 single-bit locations. The MiniCOP accesses the data table via the IPC 621 Driver Utility, which is called and executed from a BASIC09 procedure.

The IPC 621 Driver Utility is essentially a personality procedure that allows the MiniCOP to communicate to its host via the 621 I/O System Bus by way of the data table. IPC621 is an executable program that is loaded to the MiniCOP's execution directory along with the OS-9 System Commands, BASIC09, RUNB, and other programs. The IPC621 utility requires the BASIC09 statement RUN in order to execute and has its own syntax, commands, and parameters as follows:

IPC621

ACCESS THE MiniCOP'S DATA TABLE
SYNTAX: RUNIPC621 ([quantity] <command><address><value><error>)

Where:

[quantity] Used to define the number of multiple bits or registers to read or write.
<command> Entered by parameter number.

Clear Data Table	=	0	
Get a Bit	=	1	
Put a Bit	=	2	
Get a Register	=	3	
Put a Register	=	4	
Read Multiple Bits	=	5	
Read Multiple Registers	=	6	
Write Multiple Bits	=	7	
Write Multiple Registers	=	8	
Set User Bit	=	98	
Battery Status	=	99	*1

<address> Single-bit or 16-bit register location.
<value> Name of variable or array name to be written into or read from. *2
<error>

***1** Battery Status = 1 indicates: PASS
Battery Status = 0 indicates: FAIL

***2** When using arrays the quantity parameter must equal the size of the array. The array name must include the lowest array location in parentheses to indicate the array mode.

One (1) = BASE 1 mode array
Two (2) = BASE 0 mode array

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

For example, if the BASIC09 procedure calling the IPC621utility has a BASE 1 mode array named TSTSAMPLE, its dimensioning line of code may appear as:

DIM TSTSAMPLE(60), er:INTEGER

A RUN statement, similar to the one that follows, would then be used to call and execute the IPC621 utility and might appear as:

RUN IPC621(8,8,100,TSTSAMPLE(1),er)

NOTE
ALL PARAMETERS (quantity, command, address, value, and error) must be dimensioned as integers. The quantity parameter is only required when specifying multiple bit or register operations (commands 5, 6, 7, and 8).

EXAMPLES:

OS9:RUN IPC621(0,0,0,er)	(Clears the data table)
OS9:RUN IPC621(99,0,batt,er)	(Tests battery status; 1 = Pass, 0 = Fail)
OS9:RUN IPC621(98,0,0,er)	(Sets the special function bit, D6)

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

SPECIAL CONDITION FLAG

The MiniCOP may flag the host 620 Platform Controller at any time during the execution of a BASIC09 procedure. This is done by using the IPC621 utility with command 98. The special condition flag can be used in the 620 Platform Controller to initiate a block of ladder logic that normally isn't executed. The flag is reset when a PULL of data from the MiniCOP is performed. For instance, in the following example BASIC09 procedure, the flag is set after a string of data has been received and the data has been written to the MiniCOP's data table.

Ladder logic, running in the 620 Platform Controller, recognizes the special condition flag and executes a block of normally skipped or jumped ladder logic. This ladder logic performs a PULL of the fresh data from the MiniCOP's data table, sends this data to the register module, and then again disables the block of logic. This allows the 620 Platform Controller to update itself from the MiniCOP only after new data has been provided for it in the data table. The special condition flag resets itself after each PULL of data from the 620 Platform Controller.

EXAMPLE SPECIAL CONDITION FLAG PROGRAM:

```
      DIM buffer:STRING [1]
      DIM number:STRING[5]
      DIM da(10), timeout, count, x, left, er:INTEGER
      ON ERROR GOTO 5000
      REM GET 'number' BY READING CHARACTERS FROM THE KEYBOARD ONE AT
      REM A TIME UNTIL A SPACE IS RECEIVED OR UNTIL THE TIMEOUT VALUE IS
      REM EXCEEDED.
100   number = ""
      timeout = 0
      REPEAT
            RUN inkey (buffer, left)
            IF left = 0 THEN
                  timeout = timeout + 1
            ELSE
                  number = number + buffer
                  timeout = 0
            ENDIF
      UNTIL buffer = "" OR timeout > 500
      REM IF A NUMBER WAS NOT RECEIVED DUE TO A TIMEOUT OR IF THE
      REM 10th NUMBER WAS RECEIVED, WRITE THE DATA TO THE DATA TABLE
      REM OTHERWISE, READ ANOTHER NUMBER FROM THE KEYBOARD.
      IF LEN(number) = 0 THEN 200
      count = count + 1
      da(count) = VAL(number)
      IF count = 10 THEN 200
      GOTO 100
      REM WRITE ALL DATA RECEIVED TO THE DATA TABLE.
200   FOR x = 1 TO count
            PRINT
            PRINT da(x);
            RUN IPC621 (4,99+(x),,da(x),er)
      NEXT x
      REM FLAG THE 620 PLATFORM CONTROLLER THAT NEW DATA IS AVAILABLE
      RUN IPC621(98, 0, 0, er)
      PRINT
5000  END
```


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

If the values 111, 222, 333 and 444 are entered at the keyboard during the execution of this program they will be stored in the Data Table as indicated.

DATA RECEIVED
111 222 333 444

DATA TABLE AFTER PROGRAM

Address	Value
100	111
101	222
103	333
104	444

NOTE

The FOR/NEXT loop, which begins with line number 200 and ends with NEXT x, can also be programmed in the following manner:

```
200     FOR x = 1 TO count
            PRINT da(x);
        NEXT x
        RUN IPC621 (count,8,100,da(1),er)
```


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

DATACONTROL

is an executable BASIC09 utility program that is loaded to COPRAM during initialization. It is written into the root directory (/D0), which is also the initial data directory. The DATACONTROL function can only be executed from the OS9: prompt. It permits user access to the data table resident on the MiniCOP.

Using DATACONTROL you can: clear the data table; GET and PUT single bits or registers; WRITE and READ multiple bits or registers; SET the user bit; GET battery status; and EXIT the utility program. When DATACONTROL is executed, a short menu is displayed as shown in the following example.

EXAMPLE:

```
OS9:DATACONTROL

CLEAR DATA TABLE      =    0
GET BIT               =    1
PUT BIT               =    2
GET REGISTER          =    3
PUT REGISTER          =    4
READ N BIT            =    5
READ N REGISTER       =    6
WRITE N BIT           =    7
WRITE N REGISTER      =    8
SET USER BIT          =   98
GET BATTERIESTATUS    =   99
EXIT                  =  100  ?
```

The utility prompts you (?) for the desired function and value. At the prompt, enter the numeric code that corresponds to the desired function. The GET and READ functions return the value(s) found at the specified location(s). The PUT and WRITE functions store the provided value(s) in the specified location(s) and return those values for a visual verification of the data that was sent to the data table. The CLEAR function clears or sets all data table locations to zero (0). The SET USER BIT function allows you to signal the host by setting the special condition flag. The GET BATTERIESTATUS function retrieves the batteries status bit for display; 1 = PASS, 0 = FAIL. The EXIT function quits the utility program.

The GET BIT, PUT BIT, and WRITE N BITs valid starting addresses are 0 to 8191. The GET REGISTER, PUT REGISTER, and WRITE N REGISTERs' valid starting addresses are 0 to 511. Bit value prompts accept either 1 or 0 while the register value prompts accept 0 to 65535.

The DATACONTROL utility program is generally used as a test or debug tool when developing applications programs for the MiniCOP. If the MiniCOP's programming will include routines to transfer information to and from a host 620 Platform Controller, DATACONTROL allows BASIC09 procedures to be written and tested without being connected to its host. It allows you to "force" bit and data word values into the data table where your program can retrieve them. You can also retrieve bit and data word values that have been written to the data table by the program or programs executing in the MiniCOP. DATACONTROL is essentially identical to the IPC621 utility program, but provides you with a mechanism to read and write to and from the data table by way of the Terminal Emulator. DATACONTROL is not used in place of the IPC621 utility as it cannot be called from a BASIC09 procedure.

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

HOST 620 PLATFORM CONTROLLER TO MiniCOP

Data movement between the MiniCOP and the host 620 Platform Controllers via the I/O bus and data table is controlled by communication functions. Communication functions are lines of ladder logic instructions executed in the host. The functions are called program blocks and consist of lines of ladder logic that perform specific communication data transfer operations between the 620 Platform Controller and the MiniCOP's data table. Control bits, set by ladder logic, determine what specific function is to occur.

The data table also includes a DATA TABLE POINTER that must be properly positioned before data transfer can occur. The position of the data table pointer determines which data table locations may be accessed.

DATA TABLE POINTER

The data table pointer marks a position in the MiniCOP's data table from which a data transfer is to occur. The pointer can access contiguous data table addresses one address at a time. Because the 620 Platform Controller can only view the MiniCOP's data table as 512 16-bit locations all address values refer to these 16-bit locations.

The data table pointer can access consecutive data table locations by incrementing or decrementing one address at a time. The pointer will increment or decrement as each ladder logic PUSH or PULL is executed. The condition of the control bits at the time of the PUSH or PULL dictates whether the pointer increments or decrements (SET = increment, CLEAR = decrement).

It is necessary to set the data table pointer ONLY when the desired data table location is beyond the increment/decrement range of the pointer. The increment/decrement range is 8 addresses, including the current address (pointer position) in either direction from the current pointer location. The current position of the data table pointer can be read with the Read Real-time Clock communication function. The increment/decrement range and the starting pointer address are set using the Set Data Table Pointer communication function. Figure 84 illustrates a data table that has had its pointer set to address 236 and its increment/decrement range set at 7.

Additional PUSH and PULL instructions within the program blocks may or may not change the position of the pointer depending on the setting of the control bits.

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

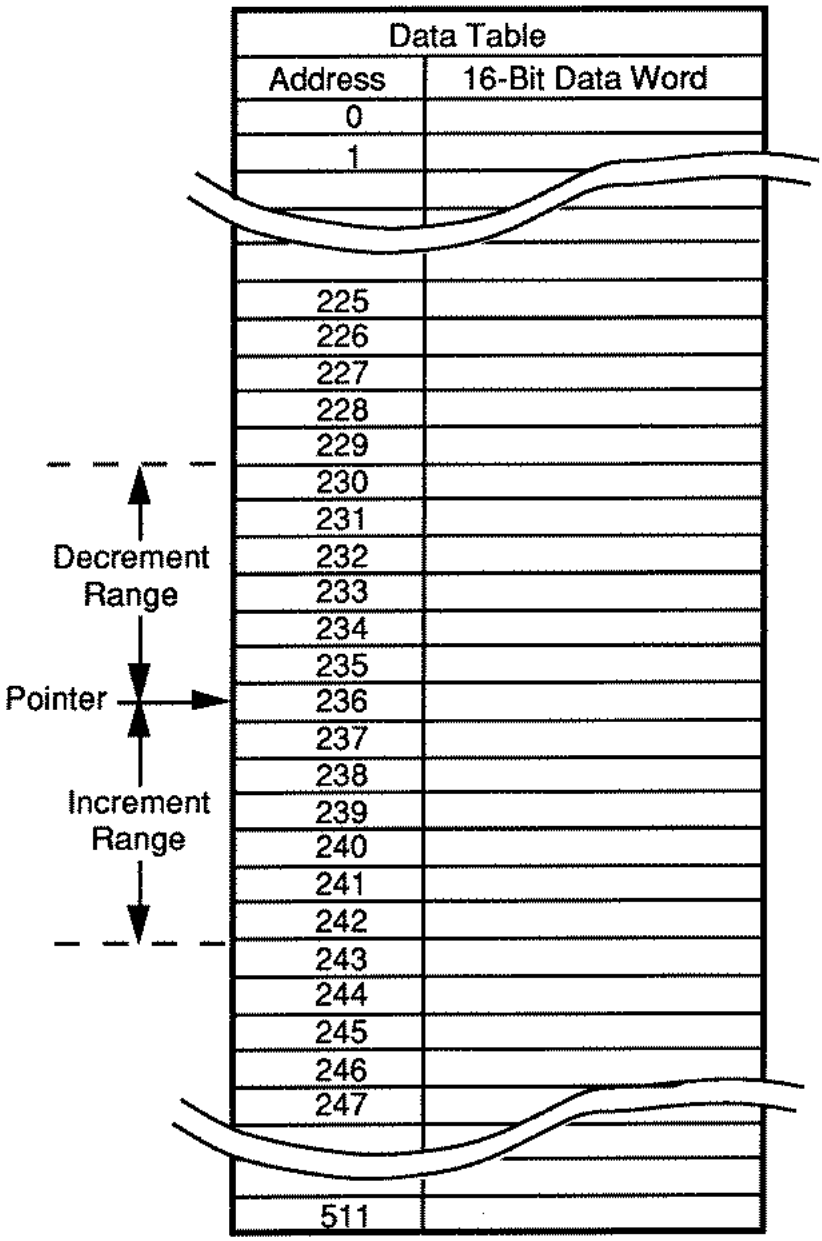


FIGURE 84 - DATA TABLE POINTER EXAMPLE

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

CONTROL BITS

There are six communication functions that are set by control bits. The MiniCOP executes a particular communication function based on the condition of these control bits. Control bits are dedicated single-bit addresses that determine a specific function of MiniCOP operation. Each bit is set by an output address in the ladder logic. This output address is a real address assigned to the MiniCOP according to its placement in the 621 Universal I/O System and is typically assigned to a nonretentive coil.

The MiniCOP references the control bits, which correspond to I/O addresses assigned to the slot where it resides, each time a PUSH or PULL instruction is executed. Based on these control bits the MiniCOP executes a particular communication function. Communication functions are programmed into the host 620 Platform Controller's ladder logic using "program blocks". This method is an efficient way to establish the host to MiniCOP communications. The program blocks are often enclosed in skip blocks. A skip block is executed on demand rather than continuously in every ladder logic scan. In this manner the program block is executed only when its operation is needed.

Figure 85 illustrates a 627-1002 MiniCOP that has been installed into a 620-1690 processor rack with a 620-36 processor module, two 16-point digital input modules, two 16-point digital output modules, one 8-point analog input module, one 4-point analog output module (8-point I/O slot), and a power supply.

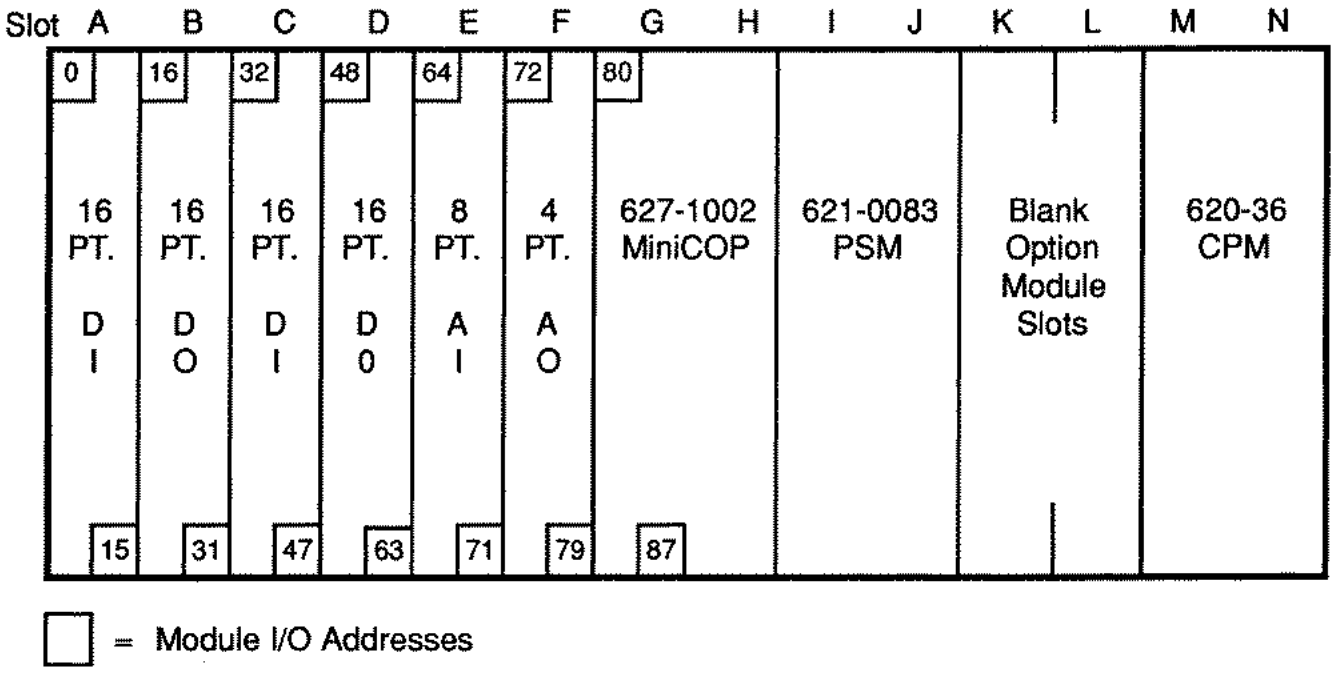


FIGURE 85 - MINICOP ADDRESSING EXAMPLE

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

The MiniCOP is an 8-point I/O module that requires two or three slots depending on the model. The left-most circuit board is the only part of the MiniCOP that communicates on the I/O bus, therefore the MiniCOP assumes the I/O addresses are assigned to the left-most of the two or three slots in which it is installed.

NOTE
Refer to MiniCOP address assignments in the *Configuration/Installation* subsection of the *MiniCOP Hardware* section for additional details regarding addressing.

In Figure 85 the MiniCOP has been assigned the I/O addresses 80 to 87. These addresses will be used as a basis of reference for the remainder of the control bit discussion. Since the MiniCOP can be effectively installed in any valid real I/O address, you as the user will have to relate these addresses to those used in your particular application.

The MiniCOP communication functions utilize eight control bits identified as D0 through D7. These bits correspond to the MiniCOP's I/O addresses as illustrated in Figure 86.

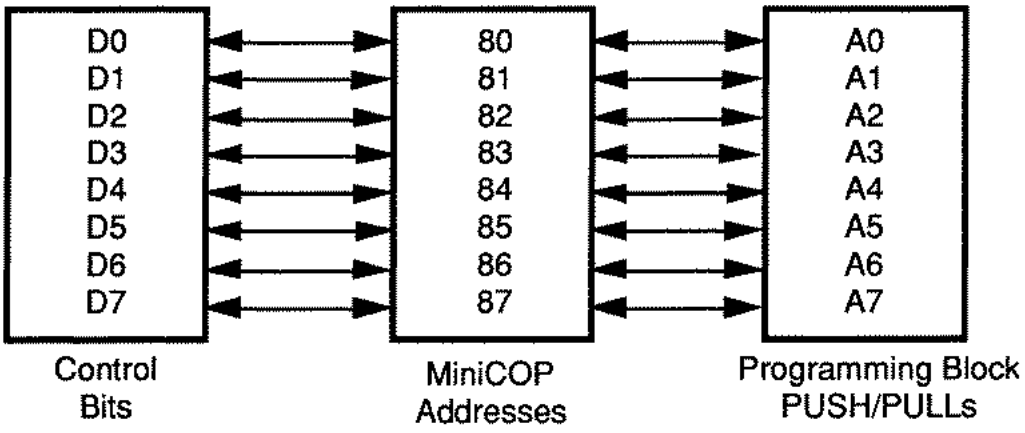


FIGURE 86 - MINICOP ADDRESS TO CONTROL BITS AND PROGRAMMING BLOCK RELATIONSHIP

In addition to the control bit's relationship to the MiniCOP's I/O addresses, Figure 86 illustrates the communication functions programming block identifiers for the ladder logic PUSH/PULL instructions. These identifiers are labeled A0 through A7. In either case, the identifier A7 always corresponds to the MiniCOP's most significant address, and identifier A0 always corresponds to the least significant.

Table 20 defines the communication function that is associated with each control bit. A communication function may require one or two control bits for its operation. The control bits given in Table 20 must be SET in order to execute the function.

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

TABLE 20 - CONTROL BIT DEFINITIONS

Control Bits	Operation type	Function Definition
D2, D0	PULL DATA	Get the real-time clock data.
D2, D1	PUSH DATA	SET the data table pointer location and the quantity of data to be transferred.
D1, D5	PUSH DATA	PUSH data to the MiniCOP. The number of registers sent should correspond to the quantity specified in the set data table pointer program block.
D0, D5	PULL DATA	PULL data from the MiniCOP. The number of registers retrieved should correspond to the quantity specified in the set data table pointer program block.
D3, D1	PUSH DATA	Set the real-time clock.
D3, D0		RESET the MiniCOP.
D4		RESET the data table pointer. Setting this bit resets the data table pointer to the last pointer position specified by the set data table pointer program block.
D5		The status of the bit determines the direction the data table pointer will move during a PUSH or PULL data operation. If this bit is set when used with D1 or D2, the pointer increments during the data transfer. If this bit is clear when used with D1 or D2, the pointer increments during the data transfer.
D6		This bit is enabled by the IPC621 driver utility using code 98. This address is reset only when the PULL data from the MiniCOP function is used.
D7		This bit is used by the MiniCOP to indicate that the tape port is being used. This bit is normally SET (ON, enabled), but is cleared (OFF, disabled) during any operation involving the TP device descriptor. A normally-open contact with this bit's address can be used to condition a Skip or Jump instruction. The Skip or Jump instruction would be used to control the execution of a program block that accesses the MiniCOP. This will allow the MiniCOP to devote full attention to its ongoing procedures without being interrupted via the I/O bus during each control processor ladder logic scan.

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

LADDER LOGIC COMMUNICATION FUNCTIONS

Communication functions are best represented as programming blocks. Programming blocks are a group of lines of ladder logic that are executed within the 620 Platform Controller. Typically, these blocks of logic are skipped or jumped during the control processor's (620 Platform Controller's control processor module) normal sequential scan. When the block's function is required, its logic is executed and then disabled again. This prevents or limits unnecessary communication operations, which adds to each device's workload and can reduce ladder logic scan time when Jump instructions are used.

Each block of ladder logic contains coils that manipulate the control bits; PUSH/PULLs that transfer the data; and time delays that ensure data integrity of the operation. A programming block can be used to perform its associated function anywhere in the control processor's ladder logic program. Any number of the blocks can be used and placed in any order without affecting the internal function of the blocks. Each block is self-contained and no additional logic is needed. Within each programming block, control bits D0 and D1 may be enabled only after all other control bits are changed and must be disabled before any other control bits can change.

NOTE

All MiniCOP/control bit addresses that are energized or set within a programming block must be de-energized or cleared prior to the block's termination. The unlatch coil instruction is used to de-energize or clear a control bit. The ladder logic used to satisfy this requirement is included in each program block.

There are eight programming block communication functions. Each function is provided in two formats: parallel or serial. If the MiniCOP is installed in a parallel I/O rack (which includes the processor rack's local I/O) select the appropriate communication function from the parallel programming blocks. If the MiniCOP is installed in a serial I/O rack, select the appropriate communication function from the serial programming blocks.

The available communication functions include:

- READ - the real-time clock
- SET - the real-time clock
- SET - the data table pointer
- PUSH - data to the MiniCOP
- PULL - data from the MiniCOP*
- SET - the pointer direction*
- RESET - the MiniCOP

* These functions define some attribute of the data table pointer. The data pointer functions are used to set-up for the PUSH and PULL instructions in the programming blocks.

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

The programming blocks that represent each communication function are illustrated in the following examples. The programming block examples are divided into two categories: parallel and serial. Follow the parallel block examples when the MiniCOP is installed in a parallel I/O rack or the local I/O slots of a processor rack. Use the serial programming block examples when the MiniCOP is installed in a serial I/O rack. Each communication function is represented in both the parallel and serial programming block formats.

The I/O addresses in the example programming blocks are to be used for the purpose of reference. The actual addresses required by any MiniCOP installation must be determined according to that specific installation. All ladder logic elements, which reference the MiniCOP, in these examples are based on the MiniCOP's use of I/O addresses 80 through 87.

NOTE

For parallel I/O installations: the ladder logic delay lines in the parallel programming blocks include a blank line of logic which represents a time delay. Time delays are necessary in a parallel installation to provide sufficient time for valid data transfer and MiniCOP operations. Different models of the various generations of 620 Platform Controllers execute instructions at different speeds. Time delay logic can vary from installation to installation. The blank time delay lines must be replaced with the appropriate logic to satisfy the time delay requirements. Refer to *Parallel Programming Block Time Delays* in this section for additional information.

Once these programming blocks are entered into ladder logic, they can be converted to Function Blocks using the 623 MS-DOS Loader (Model 623-60). Function Blocks are part of a modular programming technique available in the version 4.0, or higher, MS-DOS Loader. A Function Block is a group of lines of ladder logic that is represented on the screen display as a Function Block Display box. In this format, the lines of ladder logic are handled and displayed as if they were a single logic element, which simplifies the monitoring of the lines included in the block.

NOTE

For additional information regarding Function Block Programming refer to the *623 MS-DOS Loader User Manual.*

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

Read the Real-time Clock (Parallel I/O)
Enable Control Bits D2 and D0

The data contained in the real-time clock can be pulled from the MiniCOP by invoking this function. The error count and current data table pointer location can also be accessed using this sequence of control bits. The Read the Real-time Clock communications function PULLs the formatted information referencing address A7 as listed in Table 21.

TABLE 21 - READ THE REAL-TIME CLOCK DATA FORMAT

MiniCOP Address	Data Transfer Format
A7	Second
A6	Minute
A5	Hour
A4	Day
A3	Month
A2	Year
A1	Error Count
A0	Current Data Table Pointer Location

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

```
                READ T
                  0
LINE #1   --]/\[-------------------------------------------------------------------------------NSKR--
                                                                                                 0

                                                                                               D2
                                                                                               82
LINE #2   -------------------------------------------------------------------------------------(  )--

                                                                                               D0
                                                                                               80
LINE #3   -------------------------------------------------------------------------------------(  )--

LINE #4   ------------------------(Minimum Time Delay of 190µsec)-------------------------------

                                                                                               D0
                                                                                               80
LINE #5   -------------------------------------------------------------------------------------(U)--

                                                                                               D2
                                                                                               82
LINE #6   -------------------------------------------------------------------------------------(U)--

                  A7                                                                          TIME
                  87                                                                          4107
LINE #7   --[PUL]-------------------------------------------------------------------------------(PSH)--
                  8                                                                             8

LINE #8   -------------------------------------------------------------------------------------EOS--
                                                                                                 0
```

Lines 1 and 8 enclose the Read the Real-time Clock communication function in a skip block that is executed only on command (that is, when transitional contact is true).

Lines 2 and 3 unconditionally SET control bits D2 and D0, which directs the MiniCOP to provide the current time, date, error count and data table pointer location to its most significant I/O address (A7).

Line 4 is a time delay of at least 190µsec that allows the MiniCOP to process the control bit's request prior to receiving further instructions from the programming block.

Lines 5 and 6 unlatch (CLEAR or disable) control bits D0 and D2.

Line 7 performs a PULL of eight 16-bit data words from A7 (MiniCOP I/O address 95). This is the same information that was written to this location by the MiniCOP as directed by control bits D2 and D0 in lines 2 and 3. The PUSH instruction stores this information in the 620 Platform Controller's data registers for use by your ladder logic program.

ATTENTION When using this program block in a model 620-12, 620-1633, or 620-36 processor, refer to the *Parallel Programming Block Time Delays* in this section of the manual.

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

Set the Real-Time Clock (Parallel I/O)
Enable Control Bits D3 and D1

The MiniCOP's real-time clock can be set by ladder logic in the 620 Platform Controller. In this example the programming block sets the real-time clock using constant instructions. In reality, the ladder logic program would use Bring In or Pull instructions to obtain the time and date data to be sent to the MiniCOP. This data may be provided from internal ladder logic routines, the operator's station/panel, dedicated process reference clock, TDC 3000 Universal Station, etc.

```
                SET T
                  1
LINE #1   --]/\[-------------------------------------------------------------------------------------NSKR--
                                                                                                      1

                                                                                                     A7
                 N/A      N/A      YR       MO       DAY      HR       MIN      SEC               87
LINE #2   --[K2]-----[K2]-----[K2]-----[K2]-----[K2]-----[K2]-----[K2]-----[K2]--------(PSH)--
                  0        0       91       12       11        8        0        0                 8

LINE #3   ----------------------------(Minimum Time Delay of 40µsec)-----------------------------

                                                                                                     D3
                                                                                                     83
LINE #4   ------------------------------------------------------------------------------------------(  )--

                                                                                                     D1
                                                                                                     81
LINE #5   ------------------------------------------------------------------------------------------(  )--

LINE #6   ----------------------------(Minimum Time Delay of 190µsec)----------------------------

                                                                                                     D1
                                                                                                     81
LINE #7   ------------------------------------------------------------------------------------------(U)--

                                                                                                     D3
                                                                                                     83
LINE #8   ------------------------------------------------------------------------------------------(U)--

LINE #9   -------------------------------------------------------------------------------------EOS--
                                                                                                      1
```


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

Lines 1 and 9 enclose the Set the Real-time Clock communication function in a Skip block that is executed only on command (that is, when the transitional contact is true).

Line 2 provides the MiniCOP with the necessary time and date data to set the real-time clock. Note that the Set the Real-Time Clock programming block requires a PUSH of 8 to execute the function. The first two values PUSHed must have a value of zero; these values do not affect the operation of the MiniCOP or its real-time clock.

Line 3 is a time delay of at least 40µsec that allows the MiniCOP to receive all the PUSH data before receiving further instructions from the programming block.

Lines 4 and 5 unconditionally SET control bits D3 and D1 which directs the MiniCOP to use the data just received as its most significant I/O address (A7) as the current time and date.

Line 6 is a time delay of at least 190µsec that allows the MiniCOP to process the control bits request prior to receiving further instructions from the programming block.

Lines 7 and 8 unlatch (CLEAR or disable) control bits D1 and D3.

ATTENTION When using this program block in a model 620-12, 620-1633, or 620-36 processor, refer to the *Parallel Programming Block Time Delays* in this section of the manual.

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

Set the Data Table Pointer (Parallel I/O)
Enable Control Bits D2 and D1

The MiniCOP's data table pointer and PUSH/PULL quantity can be set by ladder logic in the 620 Platform Controller. Two 16-bit values are PUSHed to the MiniCOP's most significant I/O address (A7). The first value sets the data table pointer to a specific location. This value must be a valid data table register location (0-511). A second value (1-8) determines the maximum number of data values that can be transferred by a single PUSH or PULL.

```
                SET DTP
                  2
LINE #1  --]/\[-------------------------------------------------------------------------------------NSKR--
                                                                                                     2

                                                                                                    A7
                 DTP     QTY                                                                        87
LINE #2  --[K2]------[K2]-------------------------------------------------------------------------(PSH)--
                  32      8                                                                         2

LINE #3  ----------------------------------(Minimum Time Delay of 40µsec)-----------------------------

                                                                                                    D2
                                                                                                    82
LINE #4  ---------------------------------------------------------------------------------------(  )--

                                                                                                    D1
                                                                                                    81
LINE #5  ---------------------------------------------------------------------------------------(  )--

LINE #6  ----------------------------------(Minimum Time Delay of 140µsec)----------------------------

                                                                                                    D1
                                                                                                    81
LINE #7  ---------------------------------------------------------------------------------------(U)--

                                                                                                    D2
                                                                                                    82
LINE #8  ---------------------------------------------------------------------------------------(U)--

LINE #9  -------------------------------------------------------------------------------------EOS--
                                                                                                     2
```


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

Lines 1 and 9 enclose the Set Data Table Pointer communication function in a Skip block that is executed only on command (that is, when transitional contact is true).

Line 2 provides the MiniCOP with two 16-bit values. The first value is used as the register location to which the data table pointer is to be set. The second value determines the number of data table registers that can be transferred into or out of the data table by a single PUSH or PULL instruction.

Line 3 is a time delay of at least 40µsec that allows the MiniCOP to receive all the PUSH data before receiving further instructions from the programming block.

Lines 4 and 5 unconditionally SET control bits D2 and D1 which directs the MiniCOP to use the data just received at its most significant I/O address (A7) as the location at which to set the data table pointer and the maximum quantity of data table locations that can be transferred.

Line 6 is a time delay of at least 140µsec that allows the MiniCOP to process the control bits request prior to receiving further instructions from the programming block.

Lines 7 and 8 unlatch (CLEAR or disable) control bits D1 and D2.

ATTENTION When using this program block in a model 620-12, 620-1633, or 620-36 processor, refer to the *Parallel Programming Block Time Delays* in this section of the manual.

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

PUSH Data to the MiniCOP (Parallel I/O)
Enable Control Bits D1 and D5

This communication function PUSHes data to the MiniCOP's data table through its most significant I/O address (A7). The PUSH Data to the MiniCOP programming block can only transfer the number of values specified by the Set the Data Table Pointer function. Control bit D5, which determines the direction the data table pointer will move, is included in this programming block. A PUSH of data to the MiniCOP will place this data in the data table starting from the current position of the pointer.

```
                PSHDATA
                   3
LINE #1  --]/\[-------------------------------------------------------------------------------------NSKR--
                                                                                                      3

                                                                                                    D5
                                                                                                    85
LINE #2  -------------------------------------------------------------------------------------------(  )--

                                                                                                    A7
                                                                                                    87
LINE #3  --[K2]-----[K2]-----[K2]-----[K2]-----[K2]-----[K2]-----[K2]-----[K2]-------(PSH)--
           800      700      600      500      400      300      200      100          8

LINE #4  ----------------------------------(Minimum Time Delay of 40µsec)-----------------------

                                                                                                    D1
                                                                                                    81
LINE #5  -------------------------------------------------------------------------------------------(  )--

LINE #6  ----------------------------------(Minimum Time Delay of 289.2µsec)---------------------

                                                                                                    D1
                                                                                                    81
LINE #7  -------------------------------------------------------------------------------------------(U)--

                                                                                                    D5
                                                                                                    85
LINE #8  -------------------------------------------------------------------------------------------(U)--

LINE #9  -------------------------------------------------------------------------------------------EOS--
                                                                                                      3
```


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

Lines 1 and 9 enclose the PUSH to the MiniCOP communication function in a Skip block that is executed only on command (that is, when transitional contact is true).

Line 2 unconditionally enables control bit D5, which directs the pointer to increment from its current location when data is PUSHed to the MiniCOP's data table.

Line 3 and provides the MiniCOP with eight 16-bit data values. The PUSH data may result from any instruction that produces data values (Constant, Bring In, PULL, Etc.).

Line 4 is a time delay of at least 40µsec that allows the MiniCOP to receive all the PUSH data before receiving further instructions from the programming block.

Line 5 unconditionally SETs control bit D1 which directs the MiniCOP to store the data just received at its most significant I/O address (A7) into the data table.

Line 6 is a time delay of at least 289.2µsec that allows the MiniCOP to process the control bit's request prior to receiving further instructions from the programming block.

Lines 7 and 8 unlatch (CLEAR or disable) control bits D1 and D3.

NOTE

When multiple PUSHes of data to the MiniCOP (more than 8 data words) are required, repeat **Lines 3 through 6** as necessary.

ATTENTION

When using this program block in a model 620-12, 620-1633, or 620-36 processor, refer to the *Parallel Programming Block Time Delays* in this section of the manual.

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

PULL Data from the MiniCOP (Parallel I/O)
Enable Control Bits D0 and D5

This communication function PULLs data from the MiniCOP's data table through its most significant I/O address (A7). The PULL Data from the MiniCOP programming block can only transfer the number of values specified by the Set the Data Table Pointer function. Control bit D5, which determines the direction the data table pointer will move, is included in this programming block. A PULL of data from the MiniCOP will retrieve data from the data table starting from the current position of the pointer.

```
                PULDATA
                   4
LINE #1  --]/[-----------------------------------------------------------------------------NSKR--
                4

                                                                                          D5
                                                                                          85
LINE #2  ---------------------------------------------------------------------------------(  )--

                                                                                          D0
                                                                                          80
LINE #3  ---------------------------------------------------------------------------------(  )--

LINE #4  ----------------------------------(Minimum Time Delay of 258.8µsec)----------------------

                  A7                                                                    DATA
                  87                                                                    4127
LINE #5  --[PUL]-------------------------------------------------------------------------(PSH)--
                   8                                                                      8

                                                                                          D0
                                                                                          80

LINE #6  ---------------------------------------------------------------------------------(U)--

                                                                                          D5
                                                                                          85
LINE #7  ---------------------------------------------------------------------------------(U)--

LINE #8  -----------------------------------------------------------------------------------EOS--
                                                                                            4
```


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

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

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

ARTISAN® TECHNOLOGY GROUP