

ENGINEERING CHANGE NOTICE

Page 2 of 5

1. ECN (use no. from pg. 1)

643391

16. Design Verification Required

Yes
 No

17. Cost Impact

ENGINEERING

CONSTRUCTION

Additional \$
Savings \$ *NA*

Additional \$
Savings \$ *NA*

18. Schedule Impact (days)

Improvement

Delay *NA*

19. Change Impact Review: Indicate the related documents (other than the engineering documents identified on Side 1) that will be affected by the change described in Block 13. Enter the affected document number in Block 20.

SDD/DD	<input checked="" type="checkbox"/>	Seismic/Stress Analysis	<input type="checkbox"/>	Tank Calibration Manual	<input type="checkbox"/>
Functional Design Criteria	<input type="checkbox"/>	Stress/Design Report	<input type="checkbox"/>	Health Physics Procedure	<input type="checkbox"/>
Operating Specification	<input type="checkbox"/>	Interface Control Drawing	<input type="checkbox"/>	Spares Multiple Unit Listing	<input type="checkbox"/>
Criticality Specification	<input type="checkbox"/>	Calibration Procedure	<input type="checkbox"/>	Test Procedures/Specification	<input type="checkbox"/>
Conceptual Design Report	<input type="checkbox"/>	Installation Procedure	<input type="checkbox"/>	Component Index	<input type="checkbox"/>
Equipment Spec.	<input type="checkbox"/>	Maintenance Procedure	<input type="checkbox"/>	ASME Coded Item	<input type="checkbox"/>
Const. Spec.	<input type="checkbox"/>	Engineering Procedure	<input type="checkbox"/>	Human Factor Consideration	<input type="checkbox"/>
Procurement Spec.	<input type="checkbox"/>	Operating Instruction	<input type="checkbox"/>	Computer Software	<input type="checkbox"/>
Vendor Information	<input type="checkbox"/>	Operating Procedure	<input type="checkbox"/>	Electric Circuit Schedule	<input checked="" type="checkbox"/>
OM Manual	<input type="checkbox"/>	Operational Safety Requirement	<input type="checkbox"/>	ICRS Procedure	<input type="checkbox"/>
FSAR/SAR	<input type="checkbox"/>	IEFD Drawing	<input type="checkbox"/>	Process Control Manual/Plan	<input type="checkbox"/>
Safety Equipment List	<input type="checkbox"/>	Cell Arrangement Drawing	<input type="checkbox"/>	Process Flow Chart	<input type="checkbox"/>
Radiation Work Permit	<input type="checkbox"/>	Essential Material Specification	<input type="checkbox"/>	Purchase Requisition	<input type="checkbox"/>
Environmental Impact Statement	<input type="checkbox"/>	Fac. Proc. Samp. Schedule	<input type="checkbox"/>	Tickler File	<input type="checkbox"/>
Environmental Report	<input type="checkbox"/>	Inspection Plan	<input type="checkbox"/>	I/O Channel List	<input checked="" type="checkbox"/>
Environmental Permit	<input type="checkbox"/>	Inventory Adjustment Request	<input type="checkbox"/>		<input type="checkbox"/>

20. Other Affected Documents: (NOTE: Documents listed below will not be revised by this ECN.) Signatures below indicate that the signing organization has been notified of other affected documents listed below.

Document Number/Revision	Document Number/Revision	Document Number/Revision
<i>WHS</i> - Document Number/Revision		
HNF-SD-WM-CSDD-008 / Rev. 3		
HNF-SD-WM-EL-001 / Rev. 6		

21. Approvals

	Signature	Date	Signature	Date
Design Authority	W. G. Brown <i>W. G. Brown</i>	<i>11/1/97</i>	Design Agent	_____
Cog. Eng.	G. J. Gauck <i>G. J. Gauck</i>	<i>11/25/97</i>	PE	_____
Cog. Mgr.	R. P. Tucker <i>R. P. Tucker</i>	<i>11/25/97</i>	QA	_____
QA	R. R. True <i>R. R. True</i>	<i>1-7-98</i>	Safety	_____
Safety	L. S. Krogsrud <i>L. S. Krogsrud</i>	<i>1/7/98</i>	Design	_____
Environ.			Environ.	_____
Other	A. M. Ermi <i>A. M. Ermi</i>	<i>10-31-97</i>	Other	_____
Other				_____
Other				_____
			DEPARTMENT OF ENERGY	_____
			Signature or a Control Number that tracks the Approval Signature	_____
			ADDITIONAL	_____

**UNREVIEWED SAFETY QUESTION
SCREENING/DETERMINATION FORM**
(Per WHC-IP-0842)

Page 1 of 3

USQ Tracking No.
TF-97-0174

Rev. 1

AREA: East West GeneralFacility: 242-A DST SST LERF
 Aging Waste Other

ECN No. 637503/643391

PCA No. N/A

Work Pkg No. N/A

Other (Specify) HNF-SD-WM-SDD-045, Rev 3

**TITLE: SYSTEM DESIGN DESCRIPTION FOR SY-101 HYDROGEN MITIGATION TEST PROJECT
DATA ACQUISITION AND CONTROL SYSTEM (DACCS-1)**

Description of the Proposed Activity/REPORTABLE OCCURRENCE or PIAB:

This ECN changes the systems design description support document describing the computers system used to control, monitor and archive the processes and outputs associated with the Hydrogen Mitigation Test Pump installed in SY-101.

Introduction:

There is no new activity or procedure associated with the updating of this reference document. The updating of this system design description maintains an agreed upon documentation program initiated within the test program and carried into operations at time of turnover to maintain configuration control as outlined by design authority practicing guidelines. Any changes made to controlled components in the field will be updated after the time of implementation to support the engineers and operators understand, maintain, train to and operate the system.

There are no new credible failure modes associated with the updating of information in a support description document. The failure analysis of each change was reviewed at the time of implementation of the Systems Change Request for all the processes changed. This document simply provides a history of implementation and current system status.

The incorporation of the two documents, Computer Systems Design Description (HNF-SD-WM-CSDD-008) and the Input/Output Channel List (HNF-SD-WM-EL-001), as appendices allow for fewer errors in changes. Because the documents are all together, they will be approved as one document, not three separate entities which could be updated at different times, creating a situation which does not accurately depict field conditions.

Scope:

This USQ screening examines the Systems Design described above, HNF-SD-WM-SDD-045, Rev 3 and no other. The incorporation of the Computer Systems Design Description (HNF-SD-WM-CSDD-008) and the Input/Output Channel List (HNF-SD-WM-EL-001) to this document brings along with them all previously screen and accepted reviews against the authorization basis in effect at the time.

Authorization Basis:

HNF-SD-WM-BIO-001, Rev 0-D Tank Waste Remediation System Basis for Interim Operation

UNREVIEWED SAFETY QUESTION SCREENING/DETERMINATION FORM
 (Continued)

 USQ Tracking No.
 TF-97-0174

Rev. 1

HNF-SD-WM-TSR-006, Rev 0-E Tank Waste Remediation System Technical Safety Requirements

LA-UR-92-3196, REV 14a, Los Alamos National Laboratory 1995 Table 6-6.

Conclusion:

The update and revision of the computer system design description does not represent a USQ and the conditions surrounding the change which occurred in order to change this document have been analyzed in the authorization basis documents. Therefore no USQ determination is required. The incorporation of information into this revision of the Systems Design Description has been examined against the flammable gas check sheet, Appendix D of Appendix E of the BIO, and found not to be applicable because there is no field work associated with this ECN. The expected outcome of this USQ is ease of documentation incorporation and safe operation of the DACS by consolidation of information into one reference document.

References

TF-97-0175, 241-SY-101 HYDROGEN MITIGATION TEST PROJECT DATA ACQUISITION AND CONTROL SYSTEM (DACCS-1) INPUT/OUTPUT CHANNEL LIST. Gauck, GJ & Brown, WG. (1997)

TF-97-0581, COMPUTER SYSTEM DESIGN DESCRIPTION FOR SY-101 HYDROGEN MITIGATION TEST PROJECT DATA ACQUISITION AND CONTROL SYSTEM (DACCS-1); HNF-SD-WM-CSDD-008, Rev.3, (ECN 637502). Gauck, GJ & Brown, WG. (1997)

Van Vleet, R.J., 1994 "Safety Basis for Activities in Double-Shell Flammable Gas Watch List Tanks," WHC-SD-WM-SARR-002, Westinghouse Hanford Company, Richland WA.

Straalsund E.K. & Mendoza R.E., June 19, 1995, "SYSTEM DESIGN DESCRIPTION FOR SY-101 HYDROGEN MITIGATION TEST PROJECT DATA ACQUISITION AND CONTROL SYSTEM (DACCS-1)," USQ Screening Against ECN 198628

USQ Screening:

- A. Does the PROPOSED ACTIVITY represent a change to the facility as described in the AUTHORIZATION BASIS?

 No Yes N/A

Basis: The document, HNF-SD-WM-SDD-045, Rev 3, makes no changes to the facility as described in the authorization basis. The hardware system design description is being updated to reflect current status of the DACS system and do not drive any change to the SY-101 tank farm facility. The Safety Assessment, LA-UR-92-3196, REV 14a, Los Alamos National Laboratory 1995, requirements for DACS system hardware and software control testing and verification are discussed in Appendix S. The updating of this document is in compliance with the maintaining of current revisions of the

UNREVIEWED SAFETY QUESTION SCREENING/DETERMINATION FORM
 (Continued)

 USQ Tracking No.
 TF-97-0174

Rev. 1

software which operates the pump and provides alarms, shutdown sequences and surveillance/archiving functions.

- B. Does the PROPOSED ACTIVITY represent a change to procedures as described in the AUTHORIZATION BASIS?

 No Yes N/A

Basis: This is not a change to a procedure nor control associated with the operation of the pump or its recording surveillance devices.

- C. Does the test or experiment represent a test or experiment not described in the AUTHORIZATION BASIS documentation?

 No Yes N/A

Basis: This document up date is not a test nor experiment.

- D. Does the PROPOSED ACTIVITY or REPORTABLE OCCURRENCE, impact:

- OSRs or IOSRs?
- Approved IOSR Compliance Implementation Plan?

 No Yes N/A

Basis: There are no changes in any OSR, IOSR, or TSR documents or requirements governing the operation or configuration of this pump control system or this document outlining the systems design descriptions.

- E. Does the REPORTABLE OCCURRENCE or PIAB involve analytical errors, omissions, and/or deficiencies in the AUTHORIZATION BASIS?

 No Yes N/A

Basis: This is a change to a systems design description document not documentation noting a reportable occurrence or PIAB.

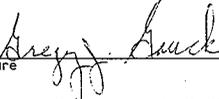
USQE No. 1 Greg J Gauck

Print Name

USQE No. 2 Warren G Brown

Print Name

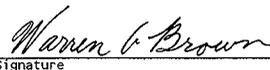
Signature



Date

11/25/97

Signature



Date

11/25/97

IF "YES", USQE CONTINUE WITH DETERMINATION BELOW

USQ DETERMINATION:

SYSTEM DESIGN DESCRIPTION FOR THE SY-101 HYDROGEN MITIGATION TEST PROJECT DATA ACQUISITION AND CONTROL SYSTEM (DACS-1)

A. M. Ermi

SGN Eurisys Services Corporation, P.O. Box 840, Richland, WA 99352
U.S. Department of Energy Contract DE-AC06-96RL13200

ECN: 643391	UC: 2030
Org Code: S4000	Charge Code: NF22Y
B&R Code: EW3120072	Total Pages: 241

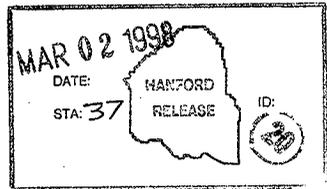
Key Words: DACS, DATA ACQUISITION AND CONTROL SYSTEM, 241SY101, MIXER PUMP, HARDWARE, SOFTWARE

Abstract: This document provides descriptions of the the hardware subsystems, software components, and functions of the automated data acquisition and control system (DACS) in support of hydrogen mitigation for waste tank 241SY101 at the Hanford Nuclear Reservation. The system was designed and implemented by Los Alamos National Laboratory, supplied to Westinghouse Hanford Company, and now operated by Lockheed Martin Hanford Corporation.

TRADEMARK DISCLAIMER. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof or its contractors or subcontractors.

Printed in the United States of America. To obtain copies of this document, contact: Document Control Services, P.O. Box 950, Mailstop H6-08, Richland WA 99352, Phone (509) 372-2420; Fax (509) 376-4989.

Jannis Aardal 3-2-98
Release Approval Date



Release Stamp

Approved for Public Release

HNF-SD-WM-SDD-045

Revision 3

**System Design Description for the
SY-101 Hydrogen Mitigation Test Project
Data Acquisition and Control System (DACS-1)**

October 1997

Prepared by:

A. M. Ermi,
SGN Eurisys Services Corporation

and

S. O. Smith & R. W. Truitt,
PLCs Plus

Prepared for:

G. J. Gauck,
Lockheed Martin Hanford Corporation

TABLE OF CONTENTS

1.0 INTRODUCTION	1
1.1 BACKGROUND	1
1.2 FUNCTION AND DESIGN REQUIREMENTS	2
1.2.1 Basis for Requirements	2
1.2.2 Functional Requirements	3
1.2.3 Design Requirements	3
1.3 DOCUMENT OVERVIEW	3
1.4 ACRONYMS	4
2.0 HARDWARE DESIGN DESCRIPTION	6
2.1 INTRODUCTION	6
2.1.1 Overall Block Diagram	6
2.1.2 Site Support	7
2.2 UNINTERRUPTABLE POWER SUPPLY	7
2.2.1 General	7
2.2.2 UPS Unit	7
2.2.3 HVAC	8
2.2.4 Batteries	9
2.2.5 Auxiliary Elements	9
2.3 HVAC SYSTEM	9
2.4 DACS TRAILER	9
2.4.1 General	9
2.4.2 Trailer Mechanical	10
2.4.2.1 <u>Bench Area</u>	10
2.4.2.2 <u>Instrument Rack Area</u>	10
2.4.2.3 <u>Computer Console Area</u>	11
2.4.3 Trailer Electrical	11
2.4.3.1 <u>Instrument Power</u>	11
2.4.3.2 <u>Utility Power</u>	11
2.4.4 Trailer Fire Protection	12
2.4.5 Emergency Lighting	12
2.5 PROGRAMMABLE LOGIC CONTROL (PLC) SYSTEM	12
2.5.1 General	12
2.5.2 Controllers	14
2.5.3 Locations (I/O Drops)	14
2.5.3.1 <u>General</u>	14
2.5.3.2 <u>Location 1</u>	15
2.5.3.3 <u>Location 2</u>	15
2.5.3.4 <u>Location 3</u>	16

	2.5.3.5	<u>Location 4</u>	16
	2.5.3.6	<u>Locations 5 and 6</u>	16
	2.5.3.7	<u>Location 7</u>	16
	2.5.3.8	<u>Location 8</u>	17
	2.5.3.9	<u>Location DACS</u>	18
2.5.4		Auxiliary Systems for the PLC	18
2.5.5		Strain Gauge Recording System	18
	2.5.5.1	<u>General</u>	18
	2.5.5.2	<u>Pump Column Strain Gauges</u>	18
	2.5.5.3	<u>MIT, VDTT Strain Gauges</u>	19
	2.5.5.4	<u>Nicolet High-Speed Recorder (STATION4)</u>	20
	2.5.5.5	<u>Strain Signal Processor</u>	20
	2.5.5.6	<u>Ectron Calibrator</u>	21
2.5.6		Modicon Console (STATION1)	21
2.5.7		Modicon-GENESIS Interface	22
2.6		GENESIS (DATA ACQUISITION AND CONTROL)	22
	2.6.1	General	22
	2.6.2	GENESIS Stations	22
	2.6.2.1	<u>STATION11, STATION13, STATION15, and STATION17</u>	25
	2.6.2.2	<u>STATION5</u>	25
	2.6.2.3	<u>STATION6, STATION7 and STATION10</u>	25
	2.6.2.4	<u>STATION7</u>	26
	2.6.2.5	<u>STATION8</u>	26
	2.6.2.6	<u>STATION9</u>	26
2.6.3		GENESIS Network (ARCNET)	26
	2.6.3.1	<u>General</u>	26
	2.6.3.2	<u>Physical Connection</u>	27
	2.6.3.3	<u>ARCNET to HLAN Connection</u>	27
	2.6.3.4	<u>Software Communication</u>	27
2.7		AUXILIARY SYSTEMS	28
	2.7.1	Printers	28
	2.7.1.1	<u>System Laser Printer</u>	28
	2.7.1.2	<u>GENESIS Printer and Security Keys</u>	29
	2.7.1.3	<u>STATION8 Printer</u>	30
	2.7.1.4	<u>Video Copier</u>	31
	2.7.2	A-B Switch for STATION7 and STATION8	31
	2.7.3	Time Code System	31
	2.7.4	External Speakers	32
	2.7.5	DACS to AF-5000+ Interface	32

2.7.6	Video System	33
2.7.6.1	<u>Internal Surveillance Camera</u>	34
2.7.6.2	<u>External Surveillance Camera</u>	34
2.7.6.3	<u>Camera Controllers</u>	35
2.7.6.4	<u>Video Recorders and Modulators</u>	35
2.7.6.5	<u>Monitors</u>	36
2.7.7	Telephone System	36
2.7.7.1	<u>AT&T Spirit Rotary PBX</u>	36
2.7.7.2	<u>Fax Machine</u>	37
2.7.7.3	<u>Modem Lines</u>	37
2.7.7.4	<u>Other Lines</u>	37
2.7.8	Weather Station	37
2.7.8.1	<u>Wind Speed and Direction Indicator</u>	37
2.7.8.2	<u>Relative Humidity Indicator</u>	38
2.7.8.3	<u>Outside Temperature Indicator</u>	38
2.7.8.4	<u>Barometric Pressure Indicator</u>	38
2.7.8.5	<u>Trailer Internal Temperature</u>	38
2.8	AF5000 VARIABLE SPEED DRIVE	38
2.8.1	General	38
2.8.2	STATION8 Interface	39
2.8.3	Remote Programming Units	39
2.9	GAS MONITORING SYSTEM	39
2.9.1	Description	39
2.9.2	DACS-GMS Interface	39
2.10	CONTROLS	40
2.11	EMERGENCY STOP CIRCUIT	40
2.12	PASSWORDS AND SECURITY	41
2.12.1	Passwords	41
2.12.2	Security Keys	42
3.0	SYSTEM LIMITATIONS, CASUALTY EVENTS AND RESPONSES	43
3.1	SYSTEM LIMITATIONS	43
3.1.1	Physical Limitations	43
3.1.2	Electrical Limitations	43
3.1.3	Control Software Limitations	43
3.1.3.1	<u>Consequences of Abort Conditions</u>	44
3.1.3.2	<u>Pump Over-Speed Protection</u>	44
3.1.3.3	<u>GENESIS vs Modicon Speed</u>	44
3.2	CASUALTY EVENTS AND RESPONSES	44
3.2.1	Fire	45
3.2.2	Earthquake	45

3.2.3	Lightning	45
3.2.4	High Winds	45
3.2.5	Other Heat and Spark Sources	45
4.0	OPERATIONS	47
4.1	INTRODUCTION	47
4.2	PUMP OPERATION	47
4.2.1	Alarms and Aborts	47
4.2.1.1	Rollover	48
4.2.1.2	ARPs and AOPs	48
4.3	DATA LOGGING	48
4.4	CONFIGURATION CHANGES	48
5.0	MAINTENANCE	49
5.1	COMPUTER MAINTENANCE	49
5.1.1	PC Filter Cleaning	49
5.1.2	PC Disk Cleaning	49
5.1.3	Other PC Maintenance	49
5.1.4	Modicon PLC Communication Checks	50
5.1.5	Startup and Shutdown	50
5.2	SYSTEM MAINTENANCE	50
5.2.1	HVAC Maintenance	50
5.2.2	UPS Battery Load Tests	51
5.2.3	UPS HVAC General Maintenance	51
5.2.4	Winterization Procedures for the UPS	51
5.2.5	Other	51
5.2.6	Startup and Shutdown Procedures	51
5.2.6.1	HVAC	51
5.2.6.2	UPS	51
5.2.6.3	Trailer	52
5.3	SIGNAL CONDITIONING MAINTENANCE	52
5.3.1	Strain Channel Signal Checks	52
5.4	CALIBRATION PROCEDURES	52
5.4.1	Modicon B875 Calibration	52
5.4.2	Modicon B883 Calibration	53
5.4.3	UPS Relay Calibrations	53
5.4.4	Weather Station Calibrations	53
6.0	COMPUTER SYSTEM LAYOUT	54
6.1	COMPONENT LAYOUT	54
6.2	LOGICAL FLOW OF DATA	57

6.3	TASK ALLOCATION TO COMPONENTS	58
6.3.1	Modicon PLC	58
6.3.2	Eaton AF-5000+ Variable Frequency Drives	58
6.3.3	TEST Strategy Terminal, STATION5	58
6.3.4	MOTOR Strategy Terminal, STATION8	58
6.3.5	Backup Station, STATION7	59
6.3.6	Local RSS Terminals, STATION6, STATION7 and STATION10	59
6.3.7	Remote RSS Terminals, STATION11, STATION13, STATION15, and STATION17	60
6.3.8	File Server Support, STATION9	60
7.0	INSTRUMENTATION	61
7.1	MIXER PUMP OPERATION MEASUREMENTS	61
7.1.1	Mixer Pump Motor Operation	62
7.1.2	Discharge Nozzle and Pump Column	62
7.2	TANK AND WASTE MEASUREMENTS	62
7.2.1	Tank and Waste Temperatures	62
7.2.2	Tank Waste Level	62
7.2.3	Tank Pressures	62
7.2.4	Vent Header	63
7.2.5	Gas Concentrations	63
7.2.6	Miscellaneous Tank Measurements	63
7.3	STRAIN MEASUREMENTS	63
7.4	TRAILER AND AREA MONITORS	64
8.0	SYSTEM FUNCTIONS AND DETAILS	65
8.1	INSTRUMENT MONITORING AND DATA COLLECTION FUNCTION (TEST STRATEGY)	65
8.1.1	TEST Strategy Communications	67
8.1.1.1	<u>Communications between the TEST Strategy and the Modicon PLC</u>	67
8.1.1.2	<u>Communications with the MOTOR Strategy</u>	72
8.1.1.3	<u>Communications from the TEST Strategy to the RSS Stations</u>	72
8.1.1.4	<u>Communications from the TEST Strategy to the Data Archiving Computer</u>	72
8.1.2	Data Monitoring	72
8.1.3	PLC Control	76
8.1.4	Data Logging	77
8.1.5	Alarming Functions	85

8.2	PUMP CONTROL FUNCTION (MOTOR STRATEGY)	89
8.2.1	Position Control	90
8.2.1.1	<u>Start</u>	92
8.2.1.2	<u>Stop</u>	93
8.2.1.3	<u>AF5000+ Communications</u>	95
8.2.1.4	<u>Position Feedback</u>	98
8.2.1.5	<u>Direction Control</u>	99
8.2.1.6	<u>Disable Movement</u>	102
8.2.2	Pump Control	103
8.2.2.1	<u>Start</u>	104
8.2.2.2	<u>Stop</u>	104
8.2.2.3	<u>AF5000+ Communications</u>	105
8.2.2.4	<u>Parameter Verification</u>	108
8.2.3	Abort	109
8.2.4	Test Operation	110
8.2.4.1	<u>Test Creation and Selection</u>	111
8.2.4.2	<u>Elapsed Time Calculation</u>	114
8.2.4.3	<u>Operator Test Information</u>	115
8.2.4.4	<u>Test Operation Strategy Details</u>	117
8.2.5	PLC Communications	127
8.2.6	Alarming	129
8.2.7	System Parameters	129
8.2.8	Stop Test	130
8.3	MODICON PLC FUNCTIONS	130
8.3.1	Data Collection From the Field	131
8.3.1.1	<u>ASCII/BASIC Module Operation</u>	133
8.3.1.2	<u>ASCII/BASIC Module Communications with the Gas Chromatograph</u>	133
8.3.1.3	<u>ASCII/BASIC Module Communications with the RGA5 Computer</u>	136
8.3.1.4	<u>Thermocouple Module Operation</u>	137
8.3.2	Abort Functions	141
8.3.3	Control Points and Alarm Outputs	144
8.3.4	Test Timers and Enable Logic	145
8.3.5	Data Transfer and Status Information	146
8.3.6	Directional Motor Simulator	149
8.4	NETWORK LAYOUT	150
9.0	REFERENCES	152

APPENDICES

Appendix A: Hardware Configurations	153
Appendix B: Key Macros and State Fields	163
Appendix C: Report and Recipe File Listings	164
Appendix D: PLC Register List	182
Appendix E: Input/Output Channel List	183
Appendix F: ASCII/BASIC Listings	191
Appendix G: PLC I/O Status Registers	197
Appendix H: Network Configuration Settings	199
Appendix I: Computer Configurations and Software Versions	201
Appendix J: File Listings	204
Appendix K: Software License Agreements	217
Appendix L: DeCipher Plus and DeTerminal Software	218

LIST OF FIGURES

Figure 1.	DACS Block Diagram	6
Figure 2.	DACS Electrical Power Block Diagram	8
Figure 3.	PLC System Hierarchical Diagram	12
Figure 4.	Strain Gauge System Essential Elements	19
Figure 5.	GENESIS Computer Block Diagram	23
Figure 6.	Laser Printer Connection	28
Figure 7.	X24E Printer Connection	29
Figure 8.	IBM Proprinter II Connection	30
Figure 9.	STATION8 to AF5000 Interface	32
Figure 10.	Trailer Video Systems Block Diagram	33
Figure 11.	E-Stop Circuit	41
Figure 12.	Diagram of the Physical Connections Between Components.	56
Figure 13.	Data and Control Paths Between Components.	57
Figure 14.	General Overview of Test Strategy Functional Regions.	65
Figure 15.	PLC I/O Strategy Detail.	71
Figure 16.	Startup Logic.	76
Figure 17.	History Data Logging Overall View.	77
Figure 18.	Timer, Report, and Pulse Portion of History Strategy Section.	79
Figure 19.	Phasing and HIST Blocks.	81
Figure 20.	GENFXR Detail.	84
Figure 21.	Abort Logic Detail.	86
Figure 22.	Instrument Fail Detail.	88
Figure 23.	Communication Pathway Failure Detection.	89
Figure 24.	DGAP Block Output	99
Figure 25.	Ladder Logic for Thermocouple Setup Table Creation.	137
Figure 26.	Ladder Logic for Thermocouple Command Table Creation.	138
Figure 27.	Ladder Logic for Configuring Thermocouple Modules.	139
Figure 28.	Thermocouple Operation: Extraction of Commands from Command Table.	140
Figure 29.	The Thermocouple Modules Reading Thermocouple Data Values.	141
Figure 30.	Abort Limit Testing.	141
Figure 31.	Composite Abort Contact.	142
Figure 32.	GENESIS to PLC Watchdog Timer.	144
Figure 33.	Test Enable and Timer Logic for Pump Bump.	145
Figure 34.	High-Frequency Strain Alarm Monitoring Logic.	147
Figure 35.	Block Moves for Transmitting I/O Health Status.	149
Figure 36.	GEN-NET Network Configuration	150

LIST OF TABLES

Table 1.	GENESIS Password Levels and Permissions	42
Table 2.	Device Setup Parameters for TEST Strategy DEV 1	69
Table 3.	Device Setup Parameters for TEST Strategy DEV 2	70
Table 4.	TEST Strategy Screens	74
Table 5.	History File Names and Contents	81
Table 6.	MOTOR Strategy Screens	90
Table 7.	Tags That Control Directional Motor Operating Conditions	95
Table 8.	Tags Used as Directional Motor Feedback Signals	96
Table 9.	Variables Used in the AF5000+ Device Block <DMOTOR>	97
Table 10.	Tags That Control Pump Motor Operating Conditions	105
Table 11.	Tags Used as Pump Motor Feedback Signals	106
Table 12.	Variables Used in the AF5000+ Device Block <PMOTOR>	107
Table 13.	Mixer Pump Signals Causing Pump Aborts	109
Table 14.	Device Setup Parameters for MOTOR Strategy DEV 3	128
Table 15.	Device Setup Parameters for MOTOR Strategy DEV 4	128
Table 16.	Report & Recipe Codes Used in the MOTOR Strategy	129
Table 17.	Input and Output Modules Used in DACS	132
Table 18.	GC-3 Data Format	134
Table A-1.	Computer #1 - Texas Microsystems in the DACS Trailer	153
Table A-2.	Computer #2 - Texas Microsystems in the DACS Trailer	154
Table A-3.	Computer #3 - Texas Microsystems in the DACS Trailer	155
Table A-4.	Computer #4 - Texas Microsystems in the DACS Trailer	156
Table A-5.	Computer #5 - Texas Microsystems in the DACS Trailer	157
Table A-6.	Computer #6 - Texas Microsystems in the DACS Trailer	158
Table A-7.	Computer #7 - Texas Microsystems in the DACS Trailer	159
Table A-8.	Computer #8 - Texas Microsystems in the DACS Trailer	160
Table A-9.	Computer #9 - Texas Microsystems in the DACS Trailer	161
Table A-10.	Computer #10 - Texas Microsystems in the DACS Trailer	162
Table G-1.	I/O Health Status Implementation Table	197
Table H-1.	ARCNET Card Parameters and Iconics Software	201
Table H-2.	Contents of GEN372, TEST and MOTOR Directories for Each Station	202
Table K-1.	Commercial Software That Is Used in the DACS System	217

TRADEMARK CREDITS

3Com is a registered trademark of 3Com Corporation
Agiler is a trademark of Sysgration Ltd.
ARCLINK is a trademark of Datapoint Corporation
ARCNET is a registered trademark of Datapoint Corporation
Av ens is a trademark of Avens Signal Equipment Corporation
AT&T is a registered trademark of AT&T Corp.
Bard is a registered trademark of Bard Manufacturing Company
Bernoulli is a registered trademark of Iomega Corporation
Black Box is a registered trademark of Black Box Corporation
Calex is a trademark of Calex Mfg. Co., Inc.
Centronics is a registered trademark of Centronics Data Computer Corporation
Custom Power is a trademark of Custom Power, Inc
Datataker Plus is a registered trademark of Data Electronics (Aust) Pty. Ltd.
Datum is a trademark of Datum Inc.
DeCipher Plus is a trademark of Data Electronics (Aust) Pty. Ltd.
DeTerminal is a trademark of Data Electronics (Aust) Pty. Ltd.
DT50, DT200 and DT500 are trademarks of Data Electronics (Aust) Pty. Ltd.
Eaton is a registered trademark of the Eaton Corporation
Ectron is a registered trademark of the Ectron Corporation
EMM386 is a trademark of Microsoft Corporation
Ethernet is a trademark of the Xerox Corporation
Expert Mouse is a trademark of Kensington Microware Ltd.
FAMOS is a trademarks of Nicolet Instrument Corporation
Fenwal is a registered trademark of Kidde-Fenwal, Inc.
Gateway 2000 is a registered trademark of Gateway 2000, Inc.
GENESIS Control Series is a trademark of Iconics, Inc.
Gen-Net is a trademark of Iconics, Inc.
Genius Mouse is a trademark of KYE International Corporation
HP LaserJet is a registered trademark of Hewlett-Packard Company
IBM PC is a trademark of the International Business Machine Corporation
IBM Proprinter is a trademark of the International Business Machine Corporation
Iomega is a registered trademark of Iomega Corporation
Jaz is a registered trademark of Iomega Corporation
JVC is a registered trademark of the Victor Company of Japan, Ltd.
Kensington is a registered trademark of Kensington Microware Ltd.
Kinometrics is a trademark of Kinometrics, Inc.
Lotus is a registered trademark of Lotus Development Corporation
Magic I/O is a trademark of Everex Systems, Inc.
Modsoft is a registered trademark of AEG Schneider Automation, Inc.

Modicon, Modbus Plus and 984 PLC are trademarks of AEG Schneider Automation, Inc.
MS-DOS and Windows are registered trademarks of Microsoft Corporation
Multisync is a trademark of NEC Corporation
NETROOM is a trademark of the Helix Software Company, Inc.
Nicolet is a registered trademarks of Nicolet Instrument Corporation
Norton DiskLock is a trademark of Symantec Corporation
Norton Utilities is a trademark of Symantec Corporation
NEC is a registered trademark of NEC Corporation
Nicolet is a registered trademarks of Nicolet Instrument Corporation
Panasonic is a registered trademark of Matsushita Electric Corporation of America
PC Tools is a trademark of Symantec Corporation
PELCO is a registered trademark of PELCO
Radio Shack is a registered trademark of the Tandy Corporation
Relco is a trademark of Relco Products, Inc.
SA-85 Modbus Plus Card is a trademark of AEG Schneider Automation, Inc.
Shinko is a trademark of Shinko Technologies, Inc.
Skyvane is a trademark of Qualimetrics, Inc.
SMC is a registered trademark of of Standard Microsystems Corporation
Soft Touch Operator Panel is a trademark of Eaton Corporation
Sony is a trademark of Sony Electronics, Inc.
Spirit is a registered trademark of AT&T Corp.
Texas Microsystems is a registered trademark of Texas Microsystems, Inc.
Vaisala is a trademark of Vaisala Oy
Whittaker is a registered trademark of Whittaker Corporation

Other product names (i.e. Bufo Enterprises, Rotronics, Shure-Lite) may be trademarks or registered trademarks of their respective companies and are hereby acknowledged.

SYSTEM DESIGN DESCRIPTION FOR SY-101 HYDROGEN MITIGATION TEST PROJECT DATA ACQUISITION AND CONTROL SYSTEM (DACS-1)

1.0 INTRODUCTION

This document describes the hardware and computer subsystems for the Data Acquisition and Control System (DACS) used in mitigation tests conducted on waste tank 241-SY-101 at the Hanford Nuclear Reservation. The system was designed and implemented by Los Alamos National Laboratory (LANL), supplied to the Westinghouse Hanford Company (WHC), and is currently being operated by Lockheed Martin Hanford Corporation.

This document reflects the DACS-1 information for the software currently in use in the DACS trailer: Test Strategy "TEST305", Motor Strategy "MOTOR302" and Modicon Ladder Logic software "PLC306".

1.1 BACKGROUND

Prior to the mixer pump installation in 1993, tank 241-SY-101 experienced recurrent periodic gas releases of hydrogen, nitrous oxide, ammonia, and methane. The hydrogen gas represents a danger, as some of the releases are in amounts above the lower flammability limit (LFL). These large gas releases had to be mitigated. Several instruments were added to the tank to monitor the gas compositions, the tank level, the tank temperature, and other parameters. A mixer pump was developed to stir the tank waste to cause the gases to be released at a slow rate. It is the function of the DACS to monitor those instruments and to control the mixer pump in a safe manner.

During FY93 and FY94 the mixer pump was installed with associated testing operations support equipment and a mitigation test project plan was implemented. These activities successfully demonstrated the mixer pump's ability to mitigate the tank 241-SY-101 hydrogen gas hazard.

The mitigation testing uses a pump immersed in the waste tank, directed at certain angles and operated at different speeds and time durations. The positioning of the pump and operation of the pump is controlled by the DACS, with the test operators commanding the system. There are many instruments used to monitor process variables within the tank at all times. The DACS collects data from these instruments, displays real-time data for operators and archives the data for later analysis; it also interfaces with control elements and safeguards the operation of the test equipment.

The DACS is composed of several components working in parallel to perform the tasks needed for test operation and monitoring.

There is a variable frequency drive that controls a motor that will move the position of the pump to different angles. A second variable frequency drive controls a motor which drives the pump that circulates the waste.

A PLC (Programmable Logic Controller) interfaces with instrumentation in the waste tank and controls elements for safe operation. This PLC has embedded logic to assure safe operation of the pump and to process data for interfacing to a supervisory networked computer system.

A networked computer system, using a multitasking software shell called GENESIS, brings together all data gathered from operator input, the PLC, and the variable frequency drives. The GENESIS system is used to collect data from the PLC and from the variable frequency drives controlling the motors. GENESIS displays this data in real time and in a graphic format to the operators. The system also stores data to files that can be analyzed at any time and these files are archived for later analysis.

The GENESIS system also is used to control the variable frequency drives' operation from operator requests. There is embedded logic in the system to warn operators of critical conditions and ensure safe operation of the pump motors.

The GENESIS system supports peer-to-peer network communications to enable data to be shared between terminals. The GENESIS network also supports host communications to multiple supervisory stations, allowing several consoles to access common data.

1.2 FUNCTION AND DESIGN REQUIREMENTS

1.2.1 Basis for Requirements

The requirements for the DACS were based on the following two WHC documents: "Test Plan for Run-In of 101-SY Mitigation Mixer Pump" (WHC-SD-WM-TP-139) [ref. 1], and "Test Plan for Tank 101-SY Mitigation-by-Mixing Test" (WHC-SD-WM-TP-140) [ref. 2]. These documents provided the basis for conducting the run-in tests and experiments required to conduct the 241-SY-101 mitigation tests.

Other documents, "Function and Design Requirements for Test Mixer Pump" (WHC-SD-WM-FDC-022, hereafter called FDC-022) [ref. 3], and "Functional Design Criteria for Tank 241-SY-101 Hydrogen Mitigation Test Project Data Acquisition and Control System (DACS-1)" (WHC-SD-WM-FDC-031, hereafter called FDC-031) [ref. 4], contain more information on DACS function and design requirements. This section comprises excerpts from these documents.

In July 1992, FDC-022 was finalized and released as Revision 1. In addition to presenting the functional design criteria (FDC) for design and analysis of candidate hydrogen mitigation concepts, FDC-022 provided extensive criteria for design of the DACS, which supports mitigation testing operations. While some general DACS criteria and/or descriptions appear throughout FDC-022, Sections 2.3, 3.6, 3.7, 3.8, 3.9, 3.10, and 3.13 are, for the most part, specific to the DACS. FDC-022 Table 3-1 (in Section 3.6) provides a detailed listing of all mitigation testing instrumentation.

1.2.2 Functional Requirements

The DACS is required to monitor and control all of the tank 241-SY-101 hydrogen mitigation testing systems. This edict is from the Safety Assessment (SA) [ref. 5]. Associated ancillary systems and equipment must also be monitored and controlled, including heating ventilation and air conditioning (HVAC), uninterruptable power supply (UPS), and other support systems required to achieve safe and reliable operation of the testing systems in the tank. Monitoring and control functions include data display, alarm annunciation, and data storage. The DACS is also required to have methods for monitored data to be transferred to outside data management and analysis organizations.

Data monitored by the DACS include physical and chemical properties of the waste and the gases in the dome space and ventilation system. The DACS tank data do not include tank integrity data such as the leak detection system or concrete and annular space temperatures.

The DACS control functions include pump manual control, pump automatic control, operation monitoring, test system operating condition indication, and selected displays to advise operating personnel of the current operating status of the systems. Manual emergency shutdown capabilities are required, in order to provide safety backup for equipment and personnel.

For specific DACS functional requirements, see FDC-031.

1.2.3 Design Requirements

No specific design requirements for hydrogen mitigation methods were applied to the DACS. The trailer, HVAC system, and UPS were all originally designed to fulfill missions at the Nevada Test Site (NTS) and were adapted for use on the mitigation project. The internal layout and selection of equipment were all done by consensus of project engineers.

1.3 DOCUMENT OVERVIEW

This document provides detailed descriptions of the tank 241-SY-101 DACS hardware and computer systems. The hardware layout is described in Section 2, while areas pertaining to its limitations, operation and maintenance are covered in Sections 3, 4 and 5. The computer system and associated instrumentation is described in Sections 6 and 7, while details pertaining to how the tasks are accomplished through the software is covered in Section 8.

Note: This document now encompasses the information formerly found in the DACS-1 "Computer System Design Description . . ." (HNF-SD-WM-CSDD-008, Rev. 3) and the ". . . Input / Output Channel List" (HNF-SD-WM-EL-001, Rev. 6).

1.4 ACRONYMS

The following acronyms are used:

AFD	Adjustable Frequency Drive (see also VFD, VSD)
AOP	Alarm Operational Procedure
ARP	Alarm Response Procedure
ASCII	American Standard Code for Information Interchange
BASIC	Beginner's All-Purpose Symbolic Instruction Code
BCD	Binary Coded Decimal
BNC	Bayonet connector type N, size C
CSDD	Computer System Design Description
DACS	Data Acquisition and Control System
FTIR	Fourier Transform InfraRed
GMS	Gas Monitoring System
HLAN	Hanford Local Area Network (actually a WAN)
HP	Hewlett Packard
hp	horsepower
HSDD	Hardware System Design Description
HVAC	Heating, Ventilating, Air Conditioning
ips	inches per second
IRIG	Inter-Range Instrument Group
LAN	Local Area Network
MIT	Multifunction Instrumentation Tree
NTS	Nevada Test Site
PBX	Private Bell Switch (X)
PC	Personal Computer
pc	printed circuit
PLC	Programmable Logic Controller
PNL	Pacific Northwest Laboratory (Battelle)
R&R	Report and Recipe
RSS	Remote Supervisory Station
RZ	return to zero
RTD	Resistance Thermal Device
SA	Safety Assessment
SDD	System Design Description
SHMS	Standard Hydrogen Monitoring System
SSP	Strain Signal Processor
UPS	Uninterruptable Power Supply
VCO	Voltage-controlled oscillators
VDTT	Velocity, Density, Temperature Tree

VFD Variable Frequency Drive (see also AFD, VSD)

VSD Variable Speed Drive (see also AFD, VFD)

Note: VSD and VFD are synonymous

WAN Wide Area Network

WHC Westinghouse Hanford Company

2.0 HARDWARE DESIGN DESCRIPTION

2.1 INTRODUCTION

LANL provided a DACS to monitor and control the SY-101 tank mixer pump mitigation tests at Hanford. The DACS controls the mixer pump, records, displays, and archives the required data from the instrumentation described in the I/O list. The data are made available for off-line analysis. The majority of the data are recorded at a low frequency (approximately one sample per second or longer) for long periods of time (hours to months). The DACS displays the data in simple, clear displays on computer monitors. The DACS controls the mitigation equipment through the use of commercial process control software and I/O modules. Automatic alarms alert operators of potentially unsafe conditions, and abort sequences shut down the tests when unsafe conditions occur.

2.1.1 Overall Block Diagram

Figure 1 shows a block diagram of the DACS. It is divided into six major subsystems:

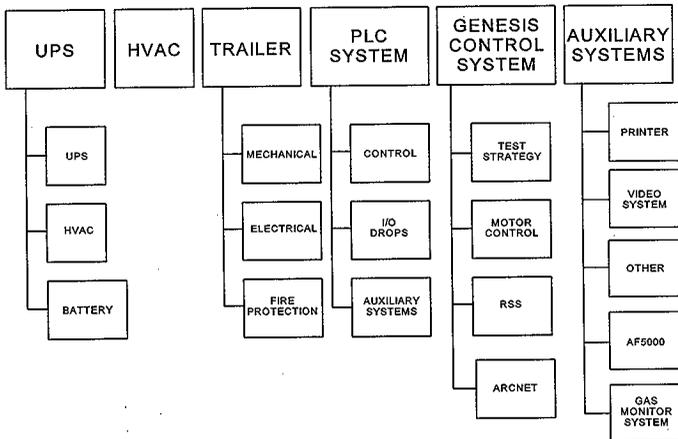


Figure 1. DACS Block Diagram

1. Uninterruptable Power Supply (UPS)
2. Heating, Ventilation, and Air Conditioning (HVAC)
3. Instrumentation trailer
4. Programmable Logic Controller (PLC) system for instrumentation
5. GENESIS control system
6. Auxiliary systems and components

Each major subsystem is further divided into its own subsystems and will be described on that basis.

2.1.2 Site Support

Site support is limited to electrical power, telephones, including the local-area network (LAN), and trailer emplacement. There is no plumbing in the DACS trailer or DACS equipment in the tank farm. The HVAC is self-contained (see Section 2.3).

2.2 UNINTERRUPTABLE POWER SUPPLY

2.2.1 General

The UPS is housed separately from the trailer and is located immediately to the west of the HVAC skid, outside the trailer. The UPS is housed in a metal "shack" mounted on a skid. Inside the shack are the following:

1. UPS unit
2. HVAC system for the shack (attached to the wall of the shack).
3. Batteries
4. Utilities (lights, breaker panels)

Outside the shack are auxiliary cables, connectors, connection boxes, a transformer, and a disconnect switch.

For the UPS system mechanical and electrical drawings, see "System Design Description 241-SY-101 DACS Uninterruptible Power Supply" (WHC-SD-WM-SD-076) [ref. 6]. See Section 2.4.3.1 for information on trailer instrument power.

2.2.2 UPS Unit

The UPS is a Custom Power model 120-10KNT. It converts 208-V 3-phase to 120-V 1-phase. It is rated for 10 kVA and can supply over 20 min of power to a fully running DACS and over 1 h of power to the DACS if only essential equipment is turned on (see Figure 2). Upon loss of power from the mains, the UPS will instantly (within 1/2 cycle of 60 Hz) take power from the batteries, invert it to 120-V 1-phase, and supply the DACS instrument power. While the power mains are active, the UPS supplies charging current to the batteries.

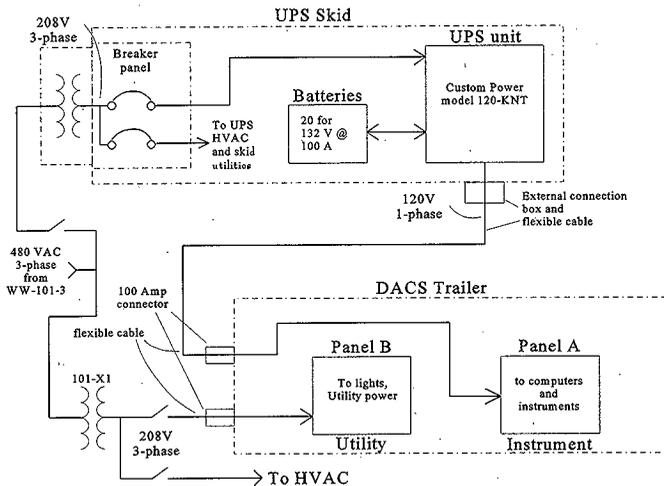


Figure 2. DACS Electrical Power Block Diagram

Switches to manually bypass the UPS and to test the batteries under load (both the DACS and a dummy load) are available. The UPS monitors both the incoming line voltage and the battery voltage and provides indicators and alarms. These indicators and alarms are displayed both at the UPS panel and, remotely, in the DACS.

2.2.3 HVAC (for UPS)

The HVAC system is mounted to the outside wall of the UPS shack and supplies forced cool air for the batteries and the UPS unit. It is a Bard model 20WA1, 1-ton unit.

2.2.4 Batteries

Twenty-five batteries are in the UPS shack. Each is a 12-V lead-acid sealed gel battery (shaped like a large car battery). Each battery is rated for 800 amp-hours.

The batteries are located in a cabinet opposite the UPS unit inside the UPS shack. The batteries are on pull-out shelves, arranged in an array of 10 by 2 (with five spares) to give a maximum output of 132 V at 100 amps nominal. The batteries are connected via heavy buss wire and the output of the battery cabinet before it is input to the UPS unit. There is a 100-amp circuit breaker to protect the battery array.

2.2.5 Auxiliary Elements

Attached to the outside of the UPS shack are power cables, junction boxes, and disconnect switches. The essential cables, disconnects, and junction boxes are identified in drawing EG&G NF-50721. Those cables and switches not identified are left over from the time the UPS shack was used at the Nevada Test Site (NTS). They were used for connecting dummy loads and supplying power to systems other than the trailer.

2.3 HVAC SYSTEM

During the 4th quarter of FY94, Morrison Refrigeration Company replaced the HVAC system at DACS-1. A contract was established with Morrison for future maintenance of the system.

The HVAC is connected to the trailer via flexible 24-in. ducts. Two ducts are provided for cool air into the trailer and hot air exiting the trailer.

2.4 DACS TRAILER

2.4.1 General

The trailer is a totally self-contained 52 x 12 ft instrumentation trailer designed to contain electronic and computer equipment. Instrument power is supplied via a 10-kVA external UPS (see Section 2.2), while separate utility power comes directly from the mains (see Section 2.1.2). The temperature is maintained by an external HVAC system (see Section 2.3). The trailer is located just outside the fence north of the SY-101 between the fence and the steam line. Details of the trailer mechanical, electrical, fire protection, rack layout, and other systems are given in drawings EG&G NF-2649 and H-2-822861, H-2-822862 and H-2-822863.

The trailer provides a suitable environment to house the computers, data acquisition equipment and other electronics. In addition to the computers and data acquisition equipment the trailer also houses the video camera controls, video monitors, and power supplies.

2.4.2 Trailer Mechanical

The trailer is mounted on a 3-ft-high structure to anchor it to the ground and keep the wheels off the ground. The underside is skirted with aluminum siding. Inside, the trailer is divided into three sections: a workbench, an instrument rack, and a computer console. In addition to those three areas are other amenities located in various places. A partial wall separates the computer consoles from the instrument racks. There are separate outside entrances to both areas. For details on the trailer mechanical see drawing EG&G NF-2649.

2.4.2.1 Bench Area

At one end of the trailer is a small workbench with equipment manuals in the cabinet above the bench and general spare parts (fasteners, cable connectors, some electronic parts).

2.4.2.2 Instrument Rack Area

The rack layout is found in drawing H-2-822861. In the rack area are 9 each standard 24 x 30 in. deep (19-in. internal) instrument racks, containing the following (listed in order from the bench area):

Rack 14:	Video monitors
Rack 13:	Open
Rack 12:	GENESIS STATION10
Rack 11:	I/O drop 14, PLC control systems
Rack 10:	Gas-monitoring computers and equipment
Rack 9:	Open
Rack 8:	GENESIS STATION9, ARCNET equipment, signal conditioning equipment, test equipment, I/O drop 15
Rack 7 & 6:	Signal conditioning equipment, test and troubleshooting equipment, and fast data-logging equipment.

(Note: a distinction is made between the physical trailer and the DACS. Thus the gas monitoring equipment located inside the trailer is not necessarily part of DACS. Likewise, all DACS equipment is not necessarily located inside the trailer.)

Signals flowing into and out of the DACS enter and exit via cables connected to a patch panel located across from racks 8, 9, and 10. The patch panel signals and layout are described in drawing H-2-815401. Modicon communication cables pass through the floor of the trailer.

2.4.2.3 Computer Console Area

In the console area are eight computer stations, each with a Texas Microsystems model 8014 industrial-type PC. The computers are numbered 1 to 8 in a clockwise order. The rack layout is given in drawing H-2-822861. The functionality of the GENESIS STATIONS 5 through 8 are described in Section 2.6.2. The functionality of STATION1 is described in Section 2.5.6.

In addition to the computer consoles is a cabinet for three printers and a fax machine (these functions are described in Section 2.7.1), a copier, and several cabinets containing documentation and backup disks. Above most of the computer video displays are 5-inch panels containing various alarm annunciators, switches, and readouts.

2.4.3 Trailer Electrical

The trailer has two separate power systems, one for instruments and one for utilities. Utility power is 208 V 3-phase. Instrument power is 120 V 1-phase from the UPS.

2.4.3.1 Instrument Power

Instrument power is supplied via the external UPS to a 4-conductor (3-phase and ground) 100-amp connector on the outside north wall of the trailer. The UPS converts 208 V 3-phase to 120 V 1-phase. At the UPS output connector, three conductors are tied together as the "hot" and the fourth conductor is the return/ground. The power enters the side of the trailer and is routed to breaker panel A. Panel A is wired for 3-phase, but as stated before, all three phases are the same. From the breakers the power is routed to the inside of each rack and console to power strips in each instrument rack and console rack. Voltage and current are monitored at the breaker panel for display in the DACS (via PLC input and GENESIS display).

The essential parts of the instrument power connection to the trailer and through the UPS are shown in drawings in the vendor information file.

2.4.3.2 Utility Power

Utility power is supplied via the power mains to a connector on the outside north wall of the trailer. Power from the connector is routed to breaker panel B. The 208 V 3-phase circuit panel is located in the wall dividing the rack area from the console area, adjacent to the

instrument power panel A. From the breakers the power is routed to the lights and power strips around the perimeter of the trailer.

The essential parts of the utility power connection to the DACS are shown on drawings in the vendor information file.

2.4.4 Trailer Fire Protection

Trailer fire protection is provided by a Fenwal model 1301 Halon fire suppression system. The Halon bottle is located in a shielded container high outside the East of the trailer and the control system is located in the wall dividing the rack area from the console area. Heat and smoke detectors are located on the ceiling around the trailer. Fire alarm pull-boxes are located next to each door. The trailer system is integrated into the Hanford fire alarm system.

2.4.5 Emergency Lighting

When utility power is lost to the trailer, two Shure-Lite model LM-1 emergency lights provide up to 90 min of emergency lighting. One is located above the copier in the console area. The other is located in the bench area. Power for charging the batteries and to detect the loss of power is provided by outlets on a utility power strip on the wall. However, to be NEC compliant, this will be changed so the (charging) power comes directly from the light power circuit.

2.5 PROGRAMMABLE LOGIC CONTROL (PLC) SYSTEM

2.5.1 General

Figure 3 shows a breakdown of the PLC system as an extension to Figure 1. The PLC system consists of the following:

1. Controllers
2. I/O drops
3. Auxiliary systems

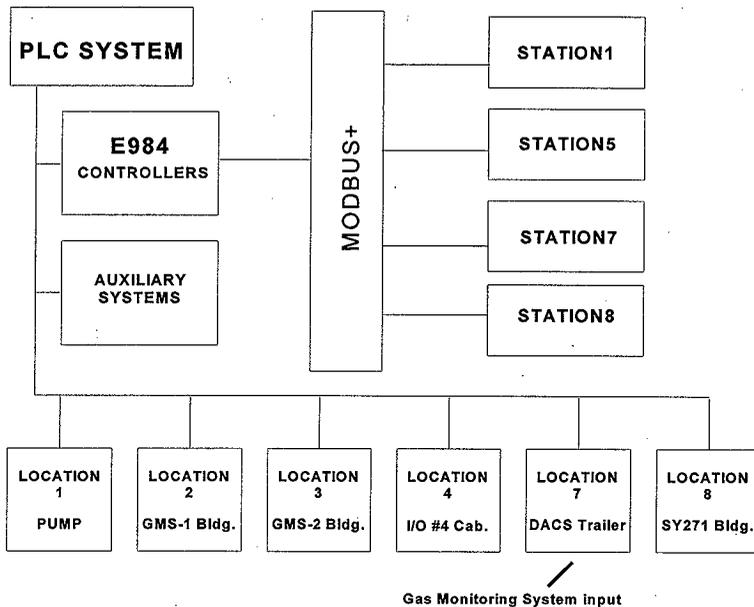


Figure 3. PLC System Hierarchical Diagram

2.5.2 Controllers

The controllers are AEG-Modicon model E984. DACS uses the E984 configured for complete redundancy via a second E984 unit operating in a "hot standby" mode and allows switchover from a failed primary E984 to the secondary E984. In the hot standby mode, data from the primary E984 are "shadowed" to the identically configured secondary I/O on a periodic basis. If internal diagnostics detect a watchdog failure, the switchover is executed and provides for a bumpless (i.e., less than one 50-ms scan time) transfer from one controller to the other.

The hot standby is connected using Modicon recommendations. For detailed information on that connection circuitry, consult the Modicon 984 manual. For detailed connection information on the programming, see Section 2.5.6.

2.5.3 Locations (I/O Drops)

2.5.3.1 General

The I/O field is made up of Modicon-AEG 800 series I/O. Typical I/O modules being used on the DACS are: model B875-101 analog input, model B824-16 24-Vdc output, model B885-002 ASCII BASIC, and model B883-200 thermocouple. The I/O is configured in several drops and distributed in the I/O clusters at strategic locations. The I/O is connected to the E984 via dual coaxial cables to the E984 PLC.

There are currently more than 200 channels of I/O that are described in the "Tank 241-101-SY Hydrogen Mitigation Test Functional Design Criteria" [ref. 3]. The I/O modules are housed in environmentally controlled housings with other electronics.

I/O drops 5 and 6 no longer exist. At one time, location 5 was to be used for the MCC shack for the Eaton VSDs.

The word "location" as listed in the I/O channel list is listed as "locations" in some older drawings and documents. Other documents use the term, "I/O drop" for location. The I/O channel list and the DACS software use the term, "I/O drop" to indicate a tap in the Modicon communications link. Furthermore the drops are not numbered sequentially, but are related to the "location" by multiplying by 2 (and adding 1 for multiple drops per location). Thus, location 1 (pump instrumentation cabinet) has drop 1, location 2 (Bldg. GMS-1) has drop 4, and so on. The exception to this is drop 13, which is at location 8. It was changed to 13 from 17 due to addressing limits of the E984 PLCs - they cannot address beyond 16.

This document will use the term "location" to indicate a physical location, and "I/O drop," or just "drop" to indicate a communication tap, to be consistent with the current H-2 drawings, the DACS software, and the I/O channel list.

2.5.3.2 Location 1

Location 1 is the set of Modicon modules in a single I/O drop (drop 2), for the mixer pump instrumentation. It consists of two Modicon racks. In the first rack are 3 model B875-101 modules set for 4-20 mA inputs. In the second rack is a model B827-024 digital input (24-V switch) module and a B865-002 BCD input module. There are no outputs (controls) in this I/O drop. The drop is physically located in a NEMA-12 instrument cabinet on top of SY-101 tank southeast of riser 16A. The following pump signals are monitored:

1. Intake and discharge pressure
2. Intake and discharge flow
3. Intake and discharge temperature
4. Motor oil temperature
5. Motor oil moisture limit
6. Rotational motor position
7. Rotational motor limits

2.5.3.3 Location 2

Location 2 is a set of Modicon modules in a single I/O drop (drop 4), for various instruments, but mainly the thermocouples on MIT 17B and MIT 17C. It consists of 2 Modicon racks. In the first rack are two model B875-101 modules set for 1 to 5-V inputs, 1 model B827-024 digital input (24-V switch) module, and one B883-200 thermocouple input module set for type-K thermocouples. In the second rack are five model B883-200 thermocouple input modules set for type-K thermocouples. These modules have been modified. There are no outputs (controls) in this I/O drop. The drop is in a small cabinet on the east wall floor inside Bldg. GMS-1. The following signals are monitored:

1. Temperatures for MIT 17B
2. Temperatures for MIT 17C
3. Vent header temperature
4. Vent header relative humidity
5. Tank dome pressure
6. Hydrogen concentration
7. Camera nitrogen pressures
8. ENRAF level gauge, riser 1A
9. ENRAF level gauge, riser 1C

2.5.3.4 Location 3

Location 3 is a set of Modicon modules in a single I/O drop (drop 6) for various gas monitoring diagnostics. It consists of one Modicon rack with four modules: three B875-101 modules and a B827-016 module. There are no outputs (controls) in this I/O drop. The drop is physically located in a cabinet on the south wall of Bldg. GMS-2. The following inputs are monitored:

1. SHMS cabinet trouble
2. Hydrogen concentrations
3. FTIR diagnostics
4. Gas chromatograph diagnostics
5. GMS-2 building temperature

2.5.3.5 Location 4

Location 4 is the set of Modicon modules in two I/O drops (drops 8 and 9) for various instruments. It consists of three Modicon racks. In the first rack are four B875-101 modules set for 4-20 mA. In the second rack is one B875-101 module set for 4-20 mA, one B827-032 module, and one B824-016 module. In the third rack (drop 9) are four model B875-101 modules set for 4-20 mA. There are no outputs (controls) in either I/O drop. The drops are located in I/O cabinet #4 near Riser 1B. The following inputs are monitored:

1. VDTT velocities and temperatures
2. I/O #4 cabinet temperature
3. Hydrogen concentration
4. Area gamma radiation and instrumentation diagnostics

2.5.3.6 Locations 5 and 6

Locations 5 and 6 are not used. Both were to be used in the original system, but have been deleted. Location 5 was the motor control center (MCC) shack, but it turned out no Modicon I/O was needed.

2.5.3.7 Location 7

Location 7 is the set of Modicon modules in two I/O drops (drops 14 and 15) for various instruments. It consists of three Modicon racks with nine modules. In the first rack are three model B875-101 modules set for 0-10 V. In the second rack is a model B827 digital input module, a B824 24-V output module, and three B875-101 modules all set for 1-10 V. In the third rack (drop 15) are two model B829 modules, a B828 module, and a B885 ASCII

BASIC module. This location is the only one with outputs (controls). The drops are physically located in the DACS trailer (see drawing H-2-822875). The following inputs are monitored:

1. Trailer instrumentation cabinet temperatures
2. UPS diagnostics
3. Video camera diagnostics
4. DACS trailer instrumentation ac current (1-phase)
5. DACS trailer instrumentation ac voltage (1-phase)
6. DACS weather station
7. Emergency stop monitor
8. Pump accelerometers
9. All strain signals
10. All high-frequency strain alarms

In addition to the input modules are instrument output modules B828 and B824-016. The following signals are output from the modules:

1. Emergency stop signal
2. Nicolet trigger
3. Communications timer (internal)

2.5.3.8 Location 8

Location 8 is the set of Modicon modules in two I/O drops (drops 13 and 16) for various instruments, mainly the thermocouples for the tank bottom and side. It consists of two Modicon racks with ten modules. In the first rack are five model B875-101 modules set for 0-5 V. In the second rack are two model B875-101 modules set for 4-20 mA and 0-5 V, and three model B883-200 thermocouple modules set for type-J thermocouples. There are no outputs (controls) on this I/O drop. The drops are located in the DACS trailer (see drawing H-2-822875). The following inputs are monitored:

1. Vent header flow
2. Tank side and bottom thermocouples
3. Vent header hydrogen concentration

2.5.3.9 Location DACS

This location is used as a place holder for internal signal names or "tags." There are no input or output signals, only internal signals. The physical location is in the DACS trailer. Internal signals in this location are used mainly for diagnostics such as instrument failures, or communication failures. The main PLC abort coil is a tag in this location.

2.5.4 Auxiliary Systems for the PLC

Each I/O drop has a model J890 remote I/O interface module. The J890 is used to control the internal working of the rack(s) in the particular drop, and to communicate with the E984 PLCs in the DACS trailer. Each drop is connected to the E984 controller via coaxial cables configured according to H-2-140169. The J890 modules communicate module status and data to the E984s. The J890s also receive setup information from the E984s for the modules in their rack(s). Each I/O drop also has a model P810 rack power supply.

Where there is more than one Modicon rack in a particular drop, one rack acts as the primary rack while the others are secondary. The primary rack contains the P810 and J890 modules. Communications to the secondary racks are provided via a model W801 multiconductor cable. Power is supplied to the secondary racks via a model W808 cable connecting the two racks.

Where external power is required (i.e., to control relays), power supplies are used. Normally these are 24-V high-current (several Amps) instrument-grade power supplies, physically located beneath the appropriate Modicon model B824-016 module. An example of this is an HP adjustable power supply (set to 24 V) located in the trailer in rack 11 below drop 14 to provide relay power for the E-stop circuits.

2.5.5 Strain Gauge Recording System

2.5.5.1 General

Figure 4 shows the essential components of the strain gauge recording system. In all channels, the strain element is a single active component in a bridge circuit with the bridge completion resistors located above the tank. The bridge excitation voltage for all channels is 10 V. The nominal strain gauge resistance is 350 ohms for all channels.

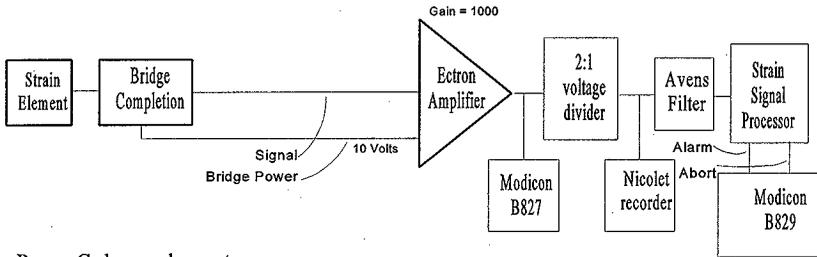
2.5.5.2 Pump Column Strain Gauges

These strain measuring systems are shown in the top half of Figure 4. They do not have a Calnex amplifier in the field, nor do they have intrinsic barriers (the pump column interior is not an explosive environment). The Ectron model 776B strain gauge signal processors (amplifiers) are set for a gain of 1000. Strain bridge excitation is provided by the

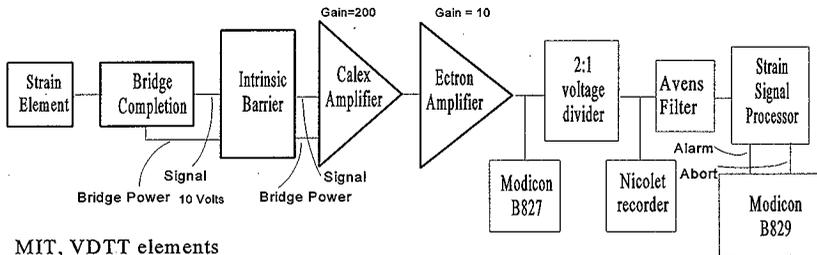
Ectron unit. The Ectron output signals pass through a 6-pole, 1-kHz low-pass Bessel filter. Full-scale output of the Ectron amplifier is ± 10 -V.

After the Ectron amplifiers, the individual signals are routed via RG-58 and a BNC connector, to a tee. Both paths after the tee provide high impedance, so this tee is a simple BNC tee. One side of the tee is routed to a 2:1 voltage divider, the other side is routed to the input of a Modicon model B875-101 module set for ± 10 -V input.

2.5.5.3 MIT, VDTT Strain Gauges



Pump Column elements



MIT, VDTT elements

Figure 4. Strain Gauge System Essential Elements

These strain measuring systems are shown in the bottom half of Figure 4. A Calex model 460-115 amplifier is used in the field to boost the signal into the trailer and into the Ectron model 776B amplifier. The gain on the Calex amplifiers is set to 200 and the gain on the Ectron amplifier is set to 10. The Calex amplifier module provides the excitation voltage for the strain bridge. The Calex output signals pass through a 2-pole, 9.5-Hz low-pass Bessel filter.

After the Ectron amplifiers, the individual signals are routed via RG-58 and a BNC connector to an individual tee. Both paths after the tee provide high impedance, so the tee is a simple BNC tee. One side of the tee is routed to a 2:1 voltage divider, the other side is routed to the appropriate input of a Modicon model B875-101 module set for +/-10-V input. The Modicon inputs are used to feed the strain signals directly to the GENESIS control software for display and archive. These are the "slow" strain signals.

Currently, two of the four strain gauges on the pump column are operational. The remaining strain gauges on all the other instrument trees are out of service.

2.5.5.4 Nicolet High-Speed Recorder (STATION4)

From the 2:1 divider (after the Ectron amplifier), the signals are again teed with a simple BNC tee. One of each of the signals from the tees is fed via RG-58 and BNC connectors to the appropriate input of the Nicolet model Multipro DAS data recorder with model 120 4-channel input cards. The Nicolet data recorder is set for a maximum input of +/-5-V.

The Nicolet data recorder is controlled by software running under Windows on STATION4. A GPIB cable connects STATION4 to the Nicolet. Setups, data retrieval, and data archive to disk are all handled from STATION4. Sampling speed, record length, offset, gain, and pretriggers may be loaded to the Nicolet data recorder from STATION4.

The Nicolet has two external triggers and both are used. One trigger is a TTL level input, and the other trigger takes a high-level pulse (although configured for 5 V). The TTL trigger input is connected to the output of a Modicon model B828-016 TTL output module (drop 15). *This signal is activated when any low- or high-frequency strain alarms occur.* The other Nicolet trigger input is connected to the output of a Modicon model B824-016 24-V switch module (drop 14). *This signal is activated when the pump is running.* Because the pump-run trigger signal is 24 V, it is divided by a simple 5:1 voltage divider in line with the signal (47 K ohms series, 10 K ohms shunt, inside a BNC in-line connection box). Both trigger signals are routed via a BNC connector and RG-58 to pigtailed wires at the Modicon module terminal blocks.

2.5.5.5 Strain Signal Processor

From the 2:1 divider (after the Ectron amplifier), the signals are, again, teed with a simple BNC tee. One of each of the signals from the tees are fed via RG-58 and BNC connectors to the appropriate input of an Avens model 4000 high-pass filter chassis located in the bottom of Rack 7. The individual channel filters are all set for a cutoff frequency of 0.1 Hz with a gain of 1.

The filter output signals are routed via RG-58 and BNC connectors to the appropriate inputs of the EG&G N-DP-172 strain signal processor (SSP). The SSP consists of

16 individual channels that each compare their input signals with preset levels. One preset level is for an "alarm" and one level is for an "abort" (an output for the "mean" level is available but not used). The alarm and abort outputs for each channel are 5-V TTL outputs active when the input level is above the respective preset level. The individual alarm and abort outputs are routed via RG-58 cable and BNC connectors to their respective input of a Modicon B829-116 TTL input module. This provides the PLC ladder logic with a direct, discrete signal for alarm and abort for the strain signals.

2.5.5.6 Ectron Calibrator

The Ectron amplifiers have built-in calibration signals that can be switched into the amplifier input. The calibration signal is a three-level signal: full-scale positive, full-scale negative, and zero. The full-scale voltage may be adjusted internally (see the Ectron manual for instructions). The three levels are applied to the input of the amplifier as a return-to-zero (RZ) pulse train. An 8-bit ID code may be programmed into this pulse train (see the Ectron manual for instructions).

An EG&G model N-TF-56 HPSCU calibrator, in rack 7, is used to control the calibration signal to all of the Ectron amplifiers. Cal Enable and Cal Clock signals (with signal return/common) are generated by the model N-TF-56 and are routed to terminal boards in rack 6. From the terminal boards, those signals are routed to the individual amplifiers to pins on J2 of the Ectron amplifiers.

To operate the model N-TF-56, turn the power on, select the rate of the RZ clock with the front panel switch, and switch Cal Enable to Manual (also on the front panel). The Ectron amplifiers will now output their individual calibration signals.

2.5.6 Modicon Console (STATION1)

The first computer rack in the trailer houses the computer that communicates and programs the Modicon E984 PLCs. The computer is a Texas Microsystems model 8014. The monitor is a Texas Microsystems model 1006R-G. The connection from the computer to the Modicon PLCs is via a Modbus Plus network connection. The computer is connected to the system printer through port LPT1 (see Section 2.7.1). A trackball is used for cursor manipulation, and is connected to a bus mouse card (IRQ10, port 338H).

Installed on STATION1 is Modsoft application software. Modsoft, Modicon's programming and utility software, provides the following functionality: offline program development, online program maintenance, and I/O module status monitoring. The Modsoft protocol uses these utilities for reading/writing registers or discrete inputs/outputs and for broadcasting data.

2.5.7 Modicon-GENESIS Interface

Interface to the GENESIS system is via Modbus-Plus protocol (see Modicon 984 User's manual). A Modbus-Plus connector is on the front of each of the E984 controllers. The Modbus-Plus network begins at the PLC at address 33, then continues to the PLC at address 01, STATION1, STATION5, STATION7, and to STATION8. The end-points (PLC at address 33 and STATION8) require a terminated connector (light grey), while all nodes in between require an in-line connector (dark grey).

2.6 GENESIS (DATA ACQUISITION AND CONTROL)

2.6.1 General

GENESIS is an industrial process control system produced by Iconics in Foxborough, MA. The software is built on a real-time operating system, AMX, providing multitasking and preemptive scheduling of events. This networked computer system, using this multi-tasking software shell, is designed to collect data from operator input, the PLC, and the variable frequency drives, display the data in real time to operators, and log the data for archival and future analysis. Functions embedded in the system for controlling and positioning the pump allow the GENESIS system to use the data collected from the PLC and variable frequency drives to warn operators of critical conditions.

The nucleus of the DACS is a system of six PCs running GENESIS. The PCs are designated as STATION5, STATION6, STATION7, STATION8, STATION9 and STATION10. STATION9 is located in rack 8 of the equipment rack area of the DACS trailer, and STATION10 is located in rack 12. The four remaining stations are located in the computer console area of the DACS trailer. In addition to these six GENESIS stations in the DACS trailer, there are four remote stations designated as STATION11, STATION13, STATION15, and STATION17. STATION11 is located in a West Tank Farms office in Bldg. 278WA, and STATION13 is located in the 306E Bldg. STATION15 is located in the 200E area, Bldg. 2750. STATION17 is a mobile DACS Developer RSS. Figure 5 shows a block diagram of the connection between the computers, and Section 2.6.2 explains the primary function of each PC.

2.6.2 GENESIS Stations

STATION5 is the GENESIS data collection runtime station.
STATIONS 6, 7 and 10 are local remote supervisory stations.
STATION7 is also a backup station to each runtime station.
STATION8 is the pump control runtime station.
STATION9 is a file server interface to the local real-time network.
STATIONS 11, 13, 15 and 17 are remote supervisory stations.

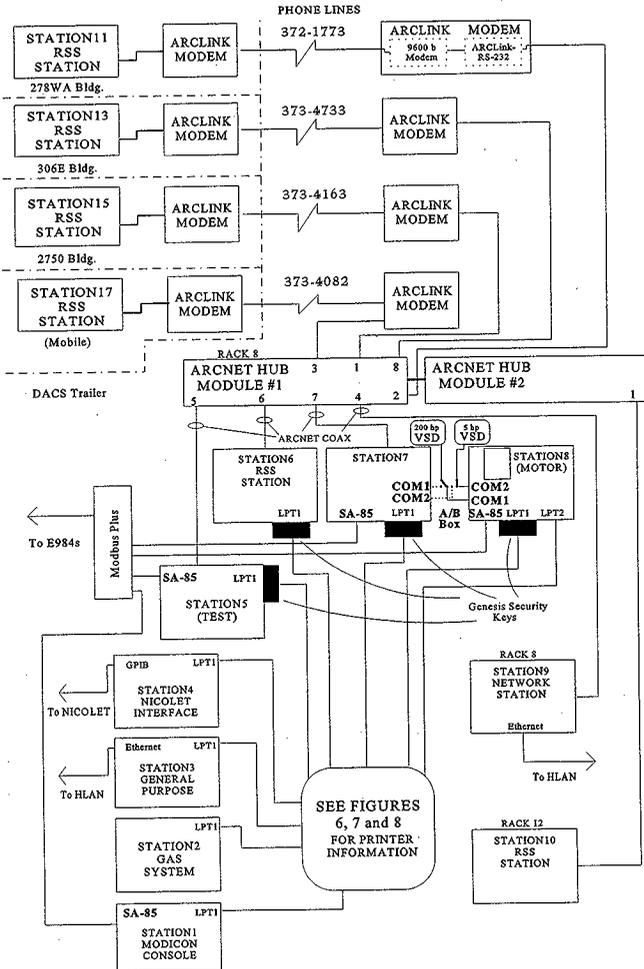


Figure 5. GENESIS Computer Block Diagram

The computers used in the DACS trailer and STATION11 are Texas Microsystems model 8014. The monitors used are Texas Microsystems model 1006R-G. STATION13 and STATION15 are Gateway 2000 computers with NEC Multisync 5FG monitors. Kensington Expert Mouse V. 2.0 model trackballs are used with each PC. The STATION17 computer is located in the 328 building, and its configuration is controlled by DACS developers. In the DACS trailer console area the OEM trackballs are replaced with color coded and numbered billiard balls corresponding to the station number. Due to the close proximity of each PC, this practice alleviates potential confusion. The internal configuration for each PC is documented in Appendix A.

The main functions of the TEST strategy on STATION5 are instrument monitoring, data logging, alarm generation, and PLC control. Essential to these functions are communications with the Modicon PLC, with the STATION8 MOTOR strategy (via the PLC) and with other stations on the network. In addition to the main functions, the strategy also provides the ability to generate hard-copy reports of the data. Given below is an overview of these TEST strategy functions and the communication paths needed to accomplish them. The GENESIS TEST strategy on STATION5 communicates with the MOTOR strategy on STATION8 and the Modicon PLC via Modbus Plus network communications. It communicates via the ARCNET with the RSS stations for operator displays, and with STATION9, the data-archiving computer.

Data monitoring is one of the primary functions of the TEST strategy. Tank instrumentation data and system status information are presented on a number of operator display screens. The operator displays fall into three basic categories plus a MAP screen. The MAP screen shows the organization of all of the other screens in the system. The three screen categories are used for monitoring and are discussed below.

The TEST strategy is responsible for logging data to the STATION5 hard drive and transmitting the data to the archiving computer STATION9. The data are used for long-term tank monitoring and to evaluate the results of the pump tests.

The runtime MOTOR strategy on STATION8 contains logic for controlling and positioning the pump. This logic uses operator input, the current status of the pump and directional motors, and the current feedback signal levels of the critical variables in the waste tank to determine what the directional motor should do. This task will command the directional motor's frequency drive to move the pump to the operator's desired position when all conditions are correct.

The mixing pump motor is controlled through its variable frequency drive by the logic associated with pump operation in the MOTOR strategy. The logic uses the current values of the critical variables in the tank to decide if commands from the operators to start or stop the pump motors should be honored. If allowable, commands are sent to change the pump's operating status.

There is logic embedded in the MOTOR strategy to abort the directional or pump motor operation. This task will immediately shut down operation of either motor if any of the abort criteria becomes active. STATION8 will display alarm conditions in the alarm summary

and warn the operator with an audible alarm when a signal goes into alarm state. See Section 6.2.1 also.

The tests that are performed on the waste in the tank involve operating the pump at different speeds and at different angles. A test is selected using a display of the MOTOR strategy on the STATION8 console. This selection determines a predetermined set of speeds and angles so that the test parameters can be downloaded easily to the active parameters of the position and mixing pump control logic.

2.6.2.1 STATION11, STATION13, STATION15, and STATION17

Four remote supervisory station (RSS) computers are located outside of the DACS trailer. STATION11 is located in a West Tank Farms office in Bldg. 278WA, STATION13 is located in the 306E Bldg., STATION15 is located in Bldg. 2750, and STATION17 is a mobile RSS used by the DACS software developers. They transfer their network data packets over phone lines at 9600 baud using special ARCNET/RS-232 conversion units called Arc-Links. The connection of the Arc-Links will be discussed in Section 2.6.3.2.

These terminals are outside of the DACS trailer. They function just as an RSS terminal in the DACS trailer except these stations are "read only." These PCs are able to receive information from the TEST strategy, but no changes to values in the strategy can be made from these terminals.

2.6.2.2 STATION5

STATION5, the primary data acquisition computer, is connected to the Modicon 984 PLC through the Modbus Plus network. The strategy functions STATION5 performs are as follows:

1. Send and retrieve field values from the Modicon 984 I/O drops
2. Write those engineering values to disk as GENESIS history files
3. Participate in the watchdog timer with the 984 ladder logic
4. Look for alarm and abort conditions in the strategy
5. Send aborts to the 984 for inclusion in the abort limit ladder logic.

2.6.2.3 STATION6, STATION7 and STATION10

The local RSSs, STATION6, STATION7 and STATION10, are linked to STATION5 and can receive and send values to the TEST strategy allowing the operator to display and change values in the strategy. These RSS terminals are allowed access to the runtime station strategy through graphic displays developed for the strategy.

2.6.2.4 STATION7

The primary role for STATION7 is as an RSS. This terminal can bring up dynamic displays that update directly from the TEST strategy. Another function of STATION7 is that of being a "backup" to either STATION5 running the TEST strategy or STATION8 running the MOTOR strategy. This station can act as the TEST strategy runtime station or the MOTOR strategy runtime station. The tasks embedded in each strategy would be performed by this station if it becomes a runtime station. See section 2.7.2.

2.6.2.5 STATION8

This personal computer is dedicated to running the GENESIS Runtime MOTOR strategy. The Gen-Net network node name STATION8 is used to reference this computer. It is connected to the PLC through the Modbus Plus network. The main task STATION8 performs during communication with the AF5000+ variable frequency drives are monitoring data from the drives and sending commands to them. This is performed with the Eaton AF5000+/GENESIS communications driver.

2.6.2.6 STATION9

STATION9 is a PC dedicated to file transfer; it has both an ethernet card to access the Westinghouse WAN (HLAN) and an ARCNET card to receive files over the Gen-Net network. The data files logged on STATION5 are archived every 2 hours by using a file transfer function to transfer data files from STATION5 to STATION9. The data files are copied to a removable media drive on STATION9 and are transferred to a file server on the HLAN. STATION9 on the Gen-Net network is logged onto a Hanford ethernet file server and is operated in Gen-Net file server mode. This allows files from STATION5 to be transferred to a Hanford file server without broadcasting real-time data packets over the HLAN.

2.6.3 GENESIS Network (ARCNET)

2.6.3.1 General

The GENESIS system supports peer-to-peer network communications to enable data to be shared between terminals. The GENESIS network also supports host communications to multiple supervisory stations, allowing several consoles to access common data. The physical network protocol used by GENESIS is ARCNET hardware implemented in a star configuration. The ARCNET and other network connections are coupled to an active hub with coaxial cable as shown in Figure 5.

2.6.3.2 Physical Connection

Each of the GENESIS stations is connected via an ARCNET link in a "star" configuration. Each station has an SMD model PC-130 ARCNET card. A BNC connector on the card is attached to a length of RG-62 cable. Each of the cables is connected to a BNC connector on one of two ARCNET hub modules located in trailer equipment rack 8. The hub modules are either a CCSI model MODHUB-16, or SMD model HUB-03-120 (both are interchangeable).

The remote RSS stations are also on the ARCNET. Since ARCNET does not support lengths of RG-62 greater than 1500 ft, special modems are used to translate the ARCNET information packets to RS-232 9600 baud for transmission over the phone lines. An ArcLink translator converts the RS-232 serial data to ARCNET (and vice-versa) data for transmission on the RG-62 cable. A synchronous modem transmits the RS-232 data over the phone lines. One pair of the modem/ArcLink-translator combination is used for each station. The modems in the trailer are set for "answer" and the remote modems are set for "dial."

2.6.3.3 ARCNET to HLAN Connection

In addition to the ARCNET, GENESIS has a connection to the HLAN. This is accomplished through STATION9. STATION9 is attached to both the ARCNET star, and the HLAN via an ethernet card. Although not a direct gateway, STATION9 runs special software to transfer archived data received from the GENESIS system, to a server on the HLAN. A broadband Ethernet card is installed in STATION9 and the connection to HLAN is made by RG-58 cable with an F connector.

2.6.3.4 Software Communication

Information must be exchanged between system components to integrate all of the data and control signals into a workable whole. Communication routines are employed to read signals and data from the PLC and AF5000 devices and to send control commands to the pump equipment. The display terminals share data over ARCNET (a real-time data network) and Modbus Plus. These networks permit multiple users to share information and allow strategies to share critical signals and control commands.

Communication between the TEST and MOTOR strategies is accomplished via PLC. Data relating to the variable-speed drives for pump motor control and directional motor control are sent from the MOTOR strategy to the PLC. The TEST strategy accesses this data for logging and display.

Pump tests are initiated from the MOTOR strategy. Critical data and other information for the control of the tests are shared by the MOTOR strategy and the PLC. These are such things as test-enable signals and timer durations so the PLC can time the length

of a test independently and use abort limit values that vary based upon the pump speed used for a particular test.

Data from the positional encoder are needed in the MOTOR strategy. These data originate in the PLC. The TEST and MOTOR strategies communicate with the Modicon PLC via Modbus Plus. Figure 5 shows the connection between the Modicon PLC and GENESIS.

Data are logged first locally on STATION5 by the TEST strategy. The data are then transferred to the data archiving computer, STATION9. After a successful transfer, the data are then deleted from STATION5. The computers are linked with ARCNET, and the transfer is accomplished by using a GENFXR block located in the TEST strategy. The data archiving computer is networked to both ARCNET and to the HLAN.

2.7 AUXILIARY SYSTEMS

Auxiliary systems include printers, printer switches, A-B switch for STATION7 and STATION8, the video copier system, PC external speakers, and annunciator panels.

2.7.1 Printers

There are three printers in use in the console area adjacent to console rack 1. Figures 6, 7, and 8 show block diagrams of the printer configuration.

2.7.1.1 System Laser Printer

Five computers (STATION1, STATION2, STATION3, STATION4, and STATION8) in the console area of the trailer share a single HP Laserjet IIP+ printer. The computers share the printer using a Black Box model #SW 809 Auto Parallel Centronics interface switch. This "smart" switch detects when a particular computer needs the printer and routes that computer's parallel port to the printer (and tells the other computers the printer is busy). The printer interface switch is mounted on a panel in the bottom of the console rack for STATION4. Standard Centronics parallel cables connect from ports LPT1 of each computer to the smart switch with the exception of STATION8, which uses port LPT2. A standard Centronics cable connects the smart switch to the laser printer. Figure 6 shows some details.

An HP laser fax machine is also connected to the printer (which produces the fax output). The connection between printer and fax is via a proprietary HP cable (see Section 2.7.7.2).

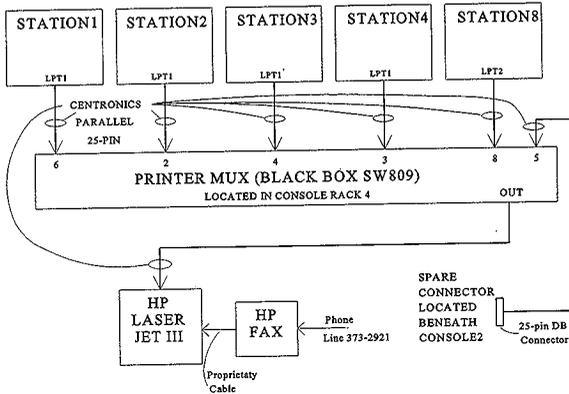


Figure 6. Laser Printer Connection

2.7.1.2 GENESIS Printer and Security Keys

STATION5, STATION6 and STATION7 share a single IBM Proprinter X24E. The computers share the printer using a Bufo Enterprises model KC 6012 Auto Data switch, similar to the Black Box switch, described above (except this is set up for three ports vs up to eight ports for the Black Box switch). This switch is located inside the console rack for STATION3. Standard Centronics parallel cables connect from ports LPT1 of each computer to the smart switch. A standard Centronics cable connects the smart switch to the printer. Figure 7 shows some details. For more information on the hardware security devices see Section 2.12.2.

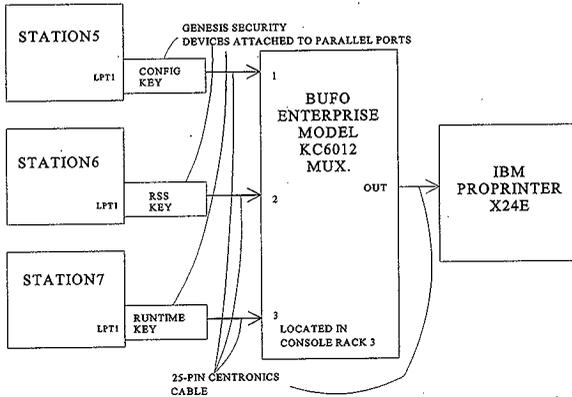


Figure 7. X24E Printer Connection

2.7.1.3 STATION8 Printer

The LPT1 port of STATION8 is connected directly to an IBM Proprinter II via a standard Centronics cable (albeit a very long cable). A GENESIS configuration security device key is also attached to port LPT1. Figure 8 shows the connection.

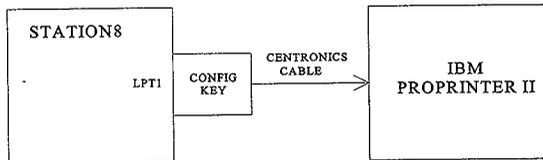


Figure 8. IBM Proprinter II Connection

2.7.1.4 Video Copier

Five computers (STATION4 through STATION8) in the console area of the trailer share a single Shinko model CHC-443MV2 color video screen copier. The copier has its own multiplexers, model SPA-9-4, tied to the video output of each computer. A push button at each station causes the copier to print the image on the selected screen.

Video signals from the backs of the computers are split out from the 15-pin DA connector to separate coaxial lines for each of the signals (a multiheaded "snake"). Each signal is connected to BNC connectors on the SPA-9-4 units. A separate multiheaded connector attaches from the SPA-9-4 unit to the video monitor, with a 15-pin gender adapter to mate with the monitor connector. Drawing LANL 5Y169636D-1 shows the details for the red, green, blue, horizontal and vertical sync signals. A composite signal called CNI (R, G, B, H, V, and C) is carried on RG-59/U coaxial cable.

Separate RG-59/U coaxial cables with BNC connectors for each signal (R, G, B, V, H, and C) connect the SPA-9-4 units together in a chain, with the Shinko copier at the end of the chain.

2.7.2 A-B Switch for STATION7 and STATION8

STATION7 acts as a backup for STATION5 and STATION8. When it is not directly configured for backup for STATION5 or STATION8, it is configured as an RSS station. When STATION7 is needed to replace STATION8, a RS-232 dual A-B switch performs the task of connecting the AF5000 serial communication lines to STATION7. The A-B switch is located on a panel above STATION5. Figure 9 shows the connections.

Software configuration to utilize STATION7 as the MOTOR strategy system is automatic. The user types "RUNTIME MOTOR" from the C:\MOTOR subdirectory, and the MOTOR application begins. *STATION8 MUST NOT BE RUNNING TO USE STATION7 AS THE RUNTIME MOTOR STATION.*

Software configuration to utilize STATION7 as the TEST strategy system is also automatic. The user types "RUNTIME TEST" from the C:\TEST subdirectory, and the TEST application begins. *STATION5 MUST NOT BE RUNNING TO USE STATION7 AS THE RUNTIME TEST STATION.*

2.7.3 Time Code System (Clock)

A Kinometrics/True Time Division model 60DC time-code receiver is located at the top of rack 12. A special antenna on the roof of the trailer picks up WWVB time code on 10 MHz from Fort Collins, CO. and sends the signal via RG-58C/U cable to the model 60DC receiver, which in turn, displays the time. The displayed time is accurate to +/- 50 ms. The receiver is set up to display Pacific time and adjusted to compensate for the delay in time from Fort Collins, CO.

Switching from daylight savings time to standard time, and vice-versa, are automatically performed by the receiver.

Two time-code Datum model 9300 generator/translator units are connected to the model 60DC receiver via RG-58C/U cable. Time-code type IRIG-B is output from the receiver to the time code translators in a daisy-chain fashion. The other translator, located high above STATION5 console (and the monitors), displays the accurate time for general-purpose use.

2.7.4 External Speakers

To increase the volume of the alarms from the GENESIS stations, external amplified speakers are connected to the internal PC speakers. For all stations, the connection is made by splicing a 3-mini stereo connector to each PC speaker, then wiring amplified speakers to that connector. The speakers are mounted under the console platforms for the keyboards.

For STATION7 and STATION8, a Radio Shack 40-1267 amplified stereo speaker is mounted below the respective keyboards. It is powered by a 9-V plug-in power supply (Radio Shack 273-1651B).

For STATION5 and STATION6 a Radio Shack 40-167 amplified mono speaker is mounted below the respective keyboards. It is powered by a 9-V plug-in power supply (Radio Shack 273-1651B). Different types of speakers were used to provide different tones for each of STATION5 and STATION8.

2.7.5 DACS to AF-5000+ Interface

The communications link between the AF-5000+ Adjustable Frequency Drive (AFD) units (see Section 2.8) and the DACS are two dedicated RS-232 serial connections between STATION8 (and STATION7) and the two motor drives. Figure 9. shows the connection. In addition to the RS-232 serial links, the AFDs are interfaced with two remote keypads, called Soft Touch Operator Panels (S.T.O.P.), located in the DACS trailer computer console area above STATION7. These keypad displays provide drive status and diagnostic information. For more details regarding how these function, consult VI files #22570 and 22571 regarding the Eaton AF-5000+. The wiring scheme showing the drive units in the MCC building (Bldg. 271-SY) connecting to the DACS is on drawing H-2-815171. The complete data path for the AF-5000+ AFDs is from the drives to STATION8 (or STATION7) using two serial links, then to the PLC and to STATION5 utilizing Modbus Plus as described in Sections 2.6.2 and 2.5.6 respectively.

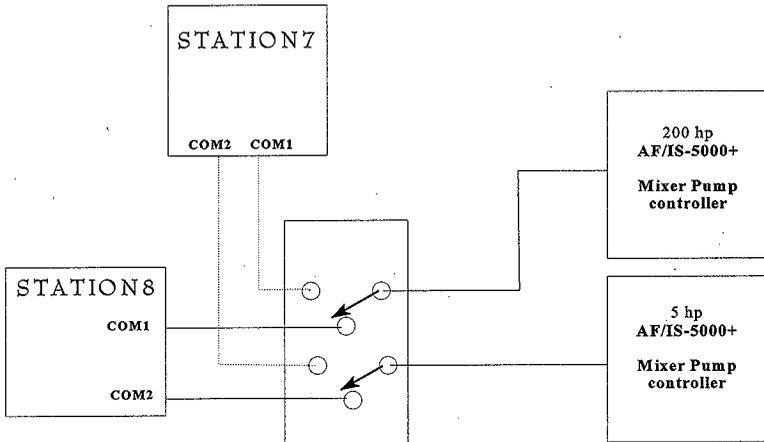


Figure 9. STATION8 to AF5000 Interface

2.7.6 Video System

The DACS trailer has three separate video systems; two from Westinghouse and one from LANL. The SY-101 in-tank video and the trailer internal surveillance cameras are Westinghouse systems. They will be briefly described with details found in the H-2 drawings. The external surveillance camera on top of the trailer is provided by LANL and will be described in detail. As an auxiliary to the video systems are the camera controllers, the videotape cassette recorders, and the video modulators. A block diagram of all the trailer video systems is shown in Figure 10.

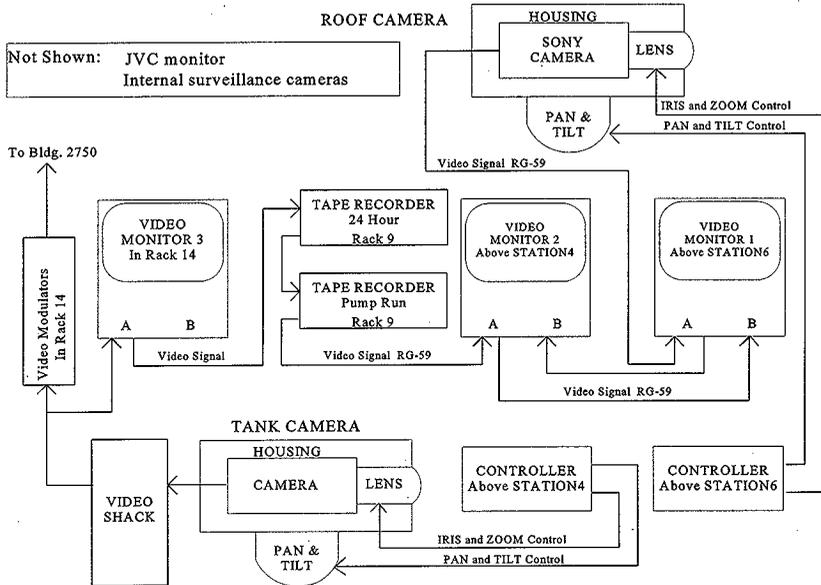


Figure 10. Trailer Video Systems Block Diagram

2.7.6.1 Internal Surveillance Camera

Two CCD cameras are mounted in the console area of the trailer positioned to observe the console operators. The video signal from these cameras is routed through the trailer according to drawing H-2-822870. Video from these cameras is available in the DACS trailer and is modulated on a separate modulator from that shown in Figure 10, on the broadband line for viewing in Bldg. 2750.

2.7.6.2 External Surveillance Camera

A Sony model SSC-M354 black-and-white CCD camera with iris controller, a Rainbow model MEA-II zoom lens with remote control constitute the camera subsystem (for controllers, see Section 2.7.6.4). This subsystem is mounted inside of a Pelco model EH550

camera enclosure (water-tight). The camera subsystem and enclosure are attached to a Pelco model PT570-24P pan and tilt unit (for controller, see Section 2.7.6.4).

2.7.6.3 Camera Controllers

The trailer roof camera is controlled by a Pelco model MLZ6DT lens control and a Pelco model MPT24DT pan and tilt control. Both controllers are mounted on a panel immediately above STATION6. A separate 24 VAC power supply (transformer) is mounted behind the control panel in console rack 6 and provides power for the pan and tilt motors. The controllers have their own internal power supplies and are connected to 120 VAC -- the controllers provide power to the lens DC motor. The 24 VAC may be monitored at terminals on the control panel. A power switch for all the camera instruments is also located on the panel. Connection between controllers and the camera is made by multiconductor cable. For connection details, see drawing H-2-822870.

NOTE: It is not possible to pan (rotate) the roof camera in a full 360 deg arc. The limit is to the east coming from both clockwise or counterclockwise directions. Damage may result to cables and connectors if the camera is forced to pan further than its limit.

The functions zoom, iris, focus, pan, and tilt are all available. Operation is simple, but for detailed instructions on how to use the controllers, consult the manufacturer's manual (VI file #22671).

For the in-tank camera, a separate remote control unit is provided by the Westinghouse video maintenance crew. The functions: zoom, iris, focus, pan, and tilt are all available. Operation is simple, but for detailed instructions on how to use the controllers, consult the manufacturer's manual for the Panasonic control unit. Connection is made between controller and camera by coaxial cable.

2.7.6.4 Video Recorders and Modulators

Video from the in-tank camera flows through the video shack, then into the trailer via RG-59/U coaxial cable and BNC connectors, to a modulator located in the bottom of rack 14 (see Figure 10). The baseband video signal is teed off and routed to the monitor in rack 14 on the "A" side. The signal is modulated onto a broadband cable for transmission to Bldg. 2750.

From the output of the monitor in rack 14 the signal is routed via RG-59 coaxial cable and a BNC connector to two VCRs located in rack 7. The signal is looped through from video-in to video-out using F-type connectors. The first VCR is a special slow recorder that

can record 24 h onto one cassette. The second VCR is a high-quality standard VHS recorder, used when the pump is running.

2.7.6.5 Monitors

The first monitor in line for the in-tank signal is a Sony SSM-221 monochrome (black- and-white) monitor. It has an internal A-B switch to select from two video sources; only the in-tank video is connected. The signal loop is not terminated; the termination switch on back of the monitor is set to Hi-Z.

After the VCRs, the in-tank signal is routed to another Sony SSM-221 located above STATION4 via RG-59 cable and a BNC connector. Here, side A is the in-tank video, and side B is the roof video. The in-tank video is then routed through the monitor above STATION4, to another monitor above STATION6 via RG-59 and BNC connectors. The monitor above STATION4 is set for HI-Z termination for the in-tank video and 75 ohms for the roof video on side B.

The monitor above STATION6 is another Sony SSM-221. The in-tank video is now on side B, and the signal line is terminated by switching the internal termination switch on the back of the monitor to 75 ohms. The roof video is side A and it is not terminated (Hi-Z).

A JVC model VM-R140SU color monitor was added to the trailer recently, above STATION5. For connection details see drawing H-2-822870. This monitor shows in-tank video.

2.7.7 Telephone System

There are ten separate dial lines into the trailer. The lines enter the trailer at terminal blocks located outside, above the power connectors. The lines are:

- | | | |
|--------------|-------------|-------------|
| 1. 373-4854 | 2. 373-2630 | 3. 373-4250 |
| 4. 373-2921 | 5. 372-1773 | 6. 373-4733 |
| 7. 373-4163 | 8. 373-4082 | 9. 372-2025 |
| 10. 373-4680 | | |

2.7.7.1 AT&T Spirit Rotary PBX

The first three lines listed above are connected to an AT&T model 308/616 3-line rotary switcher (mini-PBX). The output of the switcher is connected to four phones around the trailer. The telephone for output IC#10 is located on the rear wall by rack 12. For output IC#11 the phone is located on the front wall by the white board. For output IC#12: by STATION5 in the corner console area. For output IC#13: by STATION1 (the printer rack). Each of these phones has three lines, which are selected by pushing the appropriate button on

the phone instrument. For incoming calls, the designated line will be used and if busy, will signal busy. For outgoing calls, the next available line will be used if one or more is busy.

2.7.7.2 Fax Machine

The fax is connected to dedicated line 373-2921. See Section 2.7.1.1. for other details.

2.7.7.3 Modem Lines

There are four lines used strictly for modems. Lines 372-1773 (used for STATION11 in Bldg. 278WA), 373-4733 (used for STATION13 in the 306E Bldg.), 373-4163 (used for STATION15 in room D221, Bldg. 2750), and 373-4082 (used for STATION17) are used for the ArcLink Modems (see Section 2.6.3.2). The modems in the trailer are set for "answer," and the remote modems are set for "dial," so when communications are lost, no one has to man the trailer to reestablish communications (unless a computer needs rebooting).

2.7.7.4 Other Lines

Line 372-2025 is used by gas monitoring system personnel for voice communication to GMS-1 and GMS-2. In addition, there is one separate dedicated phone line in the trailer, 373-4680, used strictly in the console area as a "red line" to Bldg. 2750 (although is not hard-wired).

2.7.8 Weather Station

The weather station sits on the roof of the trailer and consists of separate components to measure wind speed and wind direction. Barometric pressure, temperature, and relative humidity instruments are located on the side of the trailer near the patch panel cable entrance. All roof signals come in on two cables and, with the other signals from the side, are eventually recorded through a Modicon B875-101 set for 0-10 V in Location 7, in the trailer.

2.7.8.1 Wind Speed and Direction Indicator

A SkyVane model 2102/2107 is used to measure wind speed and direction outside the trailer. Separate transducers are used for each measurement. The wind speed transducer is a mini DC generator supplying 0-5 V to a Modicon model B875-101 channel set for 0-5 V (located at drop 14 in the trailer).

The direction transducer is a rotary potentiometer. A separate 5-Vdc power supply located in the trailer patch panel area supplies current to the potentiometer. The 0-5 V direction signal is input a Modicon channel adjacent to the wind-speed input channel.

2.7.8.2 Relative Humidity Indicator

A Rotronics model 1298WD11CF005V-01 humidity gauge is used to measure relative humidity outside the trailer. A separate 24-V power supply is used to power the gauge. This power supply is mounted in rack 11. Relative humidity (RH) is a voltage in the range 0-10 V and is routed to a Modicon B875-101 channel set for the same (located in Drop 14 in the trailer).

2.7.8.3 Outside Temperature Indicator

A Relco model 1018A ambient temperature probe is used to measure outside air temperature. It uses an RTD powered by a separate 24-V power supply in the trailer. The power supply is mounted in the rack 11. Temperature is a voltage in the range from 0-10 V and is routed to a Modicon model B875-101 channel set for the same (located in Drop 14 in the trailer).

2.7.8.4 Barometric Pressure Indicator

A Vaisala model PTA 427 barometer is used to measure barometric pressure outside the trailer. Like the RH gauge, a separate 24-V power supply is used to power the barometer. The power supply is located in rack 11. The pressure signal is routed to a Modicon model B875-101 channel set for 0-10 V (located in Drop 14 in the trailer).

2.7.8.5 Trailer Internal Temperature

Several Relco RTD probes are located at various points inside the trailer to monitor the temperature. These inputs are fed to Modicon model B875-101 channels for recording and display on the GENESIS stations. See drawing EG&G NF-2649 for details on location and connectivity.

2.8 AF5000 VARIABLE SPEED DRIVE

2.8.1 General

The mixer pump and the rotation motor are both controlled by Eaton model AF/IS-5000+ variable (adjustable frequency) speed drives. The mixer pump motor is controlled by a 200 HP, unit and the rotational motor is controlled by a 5 hp unit. Both units use a 480 Vac pulse width modulated (PWM) controller. This is a state-of-the-art electronic microprocessor-controlled speed controller manufactured by Eaton, Inc.

The AF-5000+ AFD is designed to provide adjustable speed control for three phase motors. These microprocessor-based pulse-width-modulated (PWM) drives have features

allowing custom programming of the drive's performance. The mixer pump and rotation motors are both controlled by AF-5000+ AFDs.

2.8.2 STATION8 Interface

The RS-232 serial communication lines between the DACS STATION8 (or STATION7) and the serial RS-232 port on each of the drives allows the DACS to remotely control the speed of the mixer pump and the rotation motor. See Figure 9 for details.

Data are transferred from the Eaton controllers to STATION8 (or STATION7), then to the PLC and STATION5 via Modbus Plus. This AFD information is used to calculate alarms and aborts for pump operations.

2.8.3 Remote Programming Units

Located above STATION7 are two remote Soft Touch Operator Panels (S.T.O.P.) to control the AF5000 AFDs directly. For more information consult the manufacturer's operations manual.

2.9 GAS MONITORING SYSTEM

2.9.1 Description

The details of the gas monitoring system (GMS) are beyond the scope of this manual. Briefly, the GMS is composed of an FTIR instrument, several Whittaker hydrogen monitors, and other sampling equipment for measuring concentrations of gasses. Instrument integrators that control the sampling of the gasses are in turn controlled by PCs on the tank farm. The tank farm PCs are connected via ethernet to a PC in rack 10 in the DACS trailer.

2.9.2 DACS-GMS Interface

All of the GMS data are transmitted via ethernet to the PC in the trailer. There, the data are encoded and transmitted via RS-232 line from the PC serial port to a Modicon B885-002 ASCII BASIC module (located in Drop 14). Once the data have been received in the ASCII BASIC module, they are decoded and sent to the GENESIS system for display and archive.

2.10 CONTROLS

As mentioned in previous sections, transducers located on the mixer pump that monitor the pump parameters such as flow, pressure, temperature, and strain, interface to the DACS through the Modicon I/O. Other transducers that monitor SY-101 tank parameters such as temperature, waste height, gas pressure, flow, and percent hydrogen are also interfaced through Modicon I/O modules. The DACS monitors, displays, and records the outputs of these transducers to indicate the status of the mixer pump and Tank 101-SY during operation. The DACS assists the test operator in controlling the mixer pump and the mitigation test parameters to within predetermined upper and lower safe operating limits. The DACS compares the incoming data with the stored alarm and abort limits in the software and automatically alerts the operator when limits are exceeded. *The DACS automatically shuts off current to the pump when abort limits are exceeded.*

Under automatic control of the DACS software, the mixer pump can be operated either manually or in preprogrammed sequences. This allows the execution of a disciplined test sequence within a predefined envelope. In this manner, all operations of the pump can be planned in advance.

2.11 EMERGENCY STOP CIRCUIT

Note: Repeated activation of the shunt-trip will result in early failure of the circuit breaker. Manually open the circuit breaker before pushing the E-stop buttons.

The emergency stop (E-stop) circuit for the mixer pump consists of shunt trip coils implemented on the main disconnects in the MCC and feeder isolation disconnects of the mixer pump and the rotation motors. The E-stop activation will establish two breaks between the power source and the mixer pump and rotation motors. The two breaks provided are the incoming MCC line breaker and the individual feeder breakers (see drawing H-2-140167 for details). Dual redundant buttons ("panic switches") are provided for the operator to initiate the stop circuit. The buttons are located on a panel above STATION7. There is one for the main pump drive and one for the directional drive. Figure 11 shows details but not exact circuitry (for exact circuitry, see drawing H-2-140167). The E-stop circuits may be activated automatically via a Modicon model B824-016 module in Drop 14, or by pressing the appropriate stop button. The buttons are dual position: pulled out for start, and pushed in for stop. When pulled out, internal lamps illuminate the buttons, and power is supplied to the AFDs. When the buttons are pushed in, the lamps go out and the shunt-trip circuit on the main power breaker is activated, opening the breaker. Both circuits are monitored. If either is activated, a signal is sent to a Modicon model B827-024 module channel in drop 14.

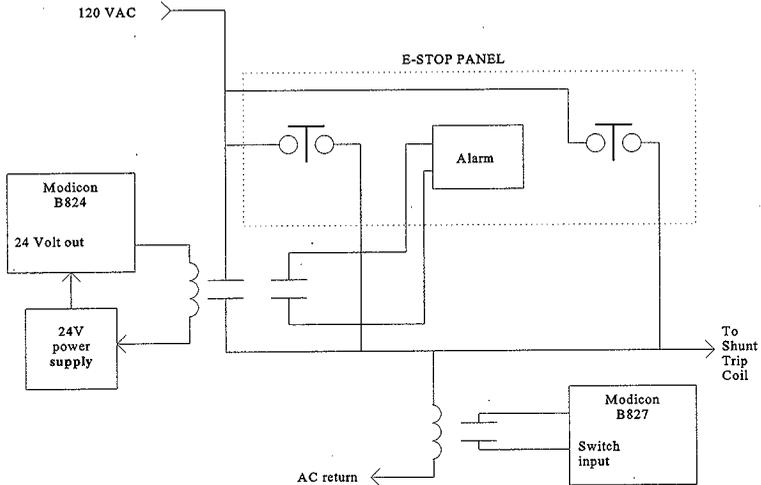


Figure 11. E-Stop Circuit

2.12 PASSWORDS AND SECURITY

There are two types of security implemented on the DACS: passwords and GENESIS security keys. The passwords are kept by the cognizant programmers and the operators.

2.12.1 Passwords

GENESIS contains three levels of password entry into the DACS system (and a non-access level). These levels are listed in Table 1, along with the privileges afforded by each level. In addition to the operational passwords, there is a separate password to protect the configuration.

Table 1. GENESIS Password Levels and Permissions

Level	Scope of permitted operation
View only (0 access)	<ul style="list-style-type: none"> ● No changes ● No submenus ● No exit to DOS
Operator 1	<ul style="list-style-type: none"> ● Data entries in user displays can be modified ● No submenus displayed ● No exit to DOS
Technician 2	<ul style="list-style-type: none"> ● Algorithm submenus displayed and values on first page can be changed ● Mode of operation can be switched between auto and manual
Engineer 3	<ul style="list-style-type: none"> ● All values can be changed during the runtime session

The four levels listed above apply to operation of the GENESIS software. There is another password for the programmer that allows configuration to take place. Software changes are done offline using special configuration software. No one may change the software without the configuration software, password, and the configuration security key (see below). There may be more than one password for each level, to keep track of who has logged on.

2.12.2 Security Keys

The GENESIS software is protected from copying and pirating by use of security devices called "keys" or "dongles." In addition to copy protection, there is *use* protection. There are three levels of user protection, all hierarchical. The highest level is called the configuration level, next is the run-time level, and the lowest is the RSS level. STATION 5 and 8 have configuration keys, STATION6 has an RSS key, and STATION7 and STATION10 have run-time keys. All keys are attached to port LPT1 before the printer cable is attached. Figures 5, 7 and 8 show the configuration.

3.0 SYSTEM LIMITATIONS, CASUALTY EVENTS AND RESPONSES

3.1 SYSTEM LIMITATIONS

There are several limitations to the DACS system in these categories: physical, electrical, control software.

3.1.1 Physical Limitations

As mentioned in Section 2.4.2, the trailer has a size limitation. This limits the amount of equipment that can be placed in the trailer. There is approximately 6 ft of usable rack space left in the rack area. In the console area there is very little usable space left. Also not mentioned in Section 2.4.2 is the ceiling height of 6 ft 5 in. This is a low ceiling by most standards, making it difficult for tall people to work in the trailer. In addition to an already low ceiling are the various protruding fire alarm system sensors.

The cable raceways above the patch panel and in the space between the rack section and the console section, are quite full of cables, making the addition of new cables difficult.

Part of the DACS is on the tank farm (Locations 1 - 4) and needs "hazardous waste-trained" people to operate, repair, and troubleshoot, whereas the trailer does not require "hazardous waste-trained" personnel. This may make troubleshooting and repair difficult.

3.1.2 Electrical Limitations

The UPS supplies about 20 min to 1 hour of instrument power to the trailer, depending on how much equipment is on. Essential data monitoring may be kept "alive" for about 1 hour, while the whole DACS trailer with all equipment on can operate for about 20 minutes.

The pump rotational motor and the main drive motor are electrically interlocked to prevent both from running at the same time. A so-called "tank sweep" with the jets rotating through the waste is not currently possible.

3.1.3 Control Software Limitations

The control software is designed to shut off an operating pump in a very short time based on any one of dozens of parameters being out of a preset range (i.e., hydrogen concentration above 1000 ppm, or vent header flow above 700 cfm). Once a parameter is detected out of range, an abort occurs, and the pump is shut off.

3.1.3.1 Consequences of Abort Conditions

Ladder Logic in the PLC continuously looks for conditions calling for an abort by using limit parameters. If an abort is calculated, a contact is closed in a binary module controlled by the PLC, interrupting power to the pump. In addition, notification of the abort is sent back to GENESIS, and the strategy sends stop pulses to the pump and direction AFDs.

3.1.3.2 Pump Over-Speed Protection

Every AF5000+ motor drive has a parameter field MAXSPD, the maximum speed in the recorded speed unit that can be programmed through the SPDSET command. If an operator tries to command the drive to run at a value greater than the MAXSPD value, the speed set point becomes the value stored in MAXSPD. The MAXSPD value is stored in nonvolatile RAM so that the last programmed value is present upon power cycling.

GENESIS also protects the set-point entry field from being programmed with values out of range. Range values are entered in AIN blocks in the configurator. These are changed by an operator entering the run-time system at the engineer password level or by a programmer knowing the password for altering the strategy. If an operator tries to over- or under-limit a set point, GENESIS restricts the entered value to the appropriate point.

The AF5000+ variable speed motor drives can be controlled by a manual panel or by the DACS. During pump testing the control will be from the DACS over an RS-232 communications link (see Figures 5 and 9). The manual control panels will be disabled to avoid interference with the DACS. Protection against an inadvertent overspeed command is enabled by setting a MAXSPD in nonvolatile memory in the motor drive. This can be done over the communications link. A command to the speed controller to exceed this speed will be disregarded and the speed set to the value of MAXSPD.

3.1.3.3 GENESIS vs Modicon Speed

The PLC is much faster than GENESIS running on the PC. For real-time response (approximately 50 ms) ladder logic must be used. GENESIS minimum response time is approximately 1 second or greater. For higher level functions that do not require "instant" response, genesis may be used.

3.2 CASUALTY EVENTS AND RESPONSES

Several casualty events will be discussed: fire, earthquake, lightning, high winds, and spark or heat sources (other than lightning).

3.2.1 Fire

The Fenwal Halon system should take care of most fires in the DACS. If a fire happens, pull the fire alarm and evacuate the trailer. If the fire is serious, (under supervision of the fire department or cognizant operations) throw to "off" the two disconnect switches located between the HVAC and the MCC shack. The two disconnects are labeled, one for trailer utility and one for trailer instruments.

3.2.2 Earthquake

The DACS trailer was designed to be used in a high-shock environment (at the NTS). The entire structure can withstand an earthquake of high magnitude, including the equipment mounted in the racks (although there is no shock absorption, the structure will remain sound).

Should an earthquake occur, perform the power-down procedure referenced in Section 5.2.6.3, and follow the site emergency action plan.

3.2.3 Lightning

The DACS trailer is designed as an instrumentation trailer, thus it is a basic EM shield at the skin. The trailer is grounded to a lightning grounding system, so little damage should occur inside. Power surges may affect some equipment.

Administrative controls in the SA forbid running the pump when thunderstorms are close (e.g. 25 miles). If lightning is imminent, when the pump is not operating, the trailer should be shut down using the power-down procedure referenced in Section 5.2.6.3.

3.2.4 High Winds

The trailer is mounted and skirted to withstand high winds up to 75-100 mph. No action is necessary for gusts up to 100 mph. If a tornado is imminent, the trailer should be shut down using the power-down procedure referenced in Section 5.2.6.3, and the site emergency action plan should be followed.

3.2.5 Other Heat and Spark Sources

The instrumentation sensor signals on the mixer pump proper are intrinsically safe by virtue of current-limiting barrier strips (except the strain gauges that are in a Nitrogen atmosphere). A mixer pump instrument cabinet, located tank-side in the vicinity of the central pump pit, contains the signal conditioning instrumentation, which includes power supplies, intrinsically safe barrier strips, the I/O drop #1 interface to the DACS, and all other mixer

pump instrumentation support hardware. The MIT and VDTT instruments are also intrinsically safe due to current-limiting barriers.

If an intrinsic barrier should fail shut, it would show up as an instrument failure. Pump operation should cease and the problem repaired. Should an intrinsic barrier fail open, it would not be detected by the DACS.

4.0 OPERATIONS

4.1 INTRODUCTION

The three trailer operations are as follow:

1. Pump operation
2. Data logging
3. Configuration changes.

Each will be discussed in general. Maintenance and repair are discussed in Section 5. Records of these operations have been kept in the operator's log. The operator's log is the official log (defined by an SOP). The operator's log is transcribed semi-weekly and is available, along with the archived data. It contains information on pump running, repairs, alarms acknowledged, and more.

4.2 PUMP OPERATION

Tank level maintenance is accomplished by operating the pump in one of four possible modes:

1. Bump
2. 1 Hour Run
3. Tank Sweep
4. Excavation

There are "tests" pre-programmed in the pump control software. The "tests" are made up of three variables namely: time, speed, and angle. Typical values are listed as follows:

- | | | |
|----|-----------------|---------------------------|
| 1. | Time (minutes) | 5, 25, 30, 60 |
| 2. | Speed (rpm) | 750, 920, 1000 |
| 3. | Angle (degrees) | 28, 65, 97, 125, 155, 185 |

4.2.1 Alarms and Aborts

Normal pump operation is kept under control of the Modicon Ladder logic and the GENESIS strategy software. All of the tank parameters are continually monitored by the software. Each parameter is compared against 2 limits; an alarm limit and an abort limit. If any one of the parameters is above the alarm limit, an alarm sounds in the trailer, and the event is logged electronically. If any parameter is above the abort limit, the same type of

alarm will sound, the event will be logged, and if the pump was running it will be immediately stopped.

The alarm and abort limits are all defined in the SA [ref. 5].

4.2.1.1 Rollover

Several parameters may be simultaneously above the alarm or abort limit. Certain combinations of these parameters (e.g. high vent flow, with high H₂ concentration) define a rollover.

4.2.1.2 ARPs and AOPs

There are written Alarm Response Procedures and Alarm Operational Procedures to follow in cases of certain alarms and aborts. These documents are located in the cabinet adjacent to STATION8.

4.3 DATA LOGGING

The system runs 24 hours a day, even though the pump may not be operating. The system gathers data on the tank. STATION5 is responsible for the data logging operation, also called, "archiving." This function is completely automatic. No operator action is required (except to start the process, and reboot if a lock-up occurs).

4.4 CONFIGURATION CHANGES

Changes to the software to accommodate more equipment, to fix bugs, or to add more data logging all fall under the "System Configuration Management Plan for SY-101 DACS" document (WHC-SD-WM-CSCM-023)[ref. 7]. A change control board takes requests for software and hardware changes, reviews them and approves or disapproves the requests. All necessary documentation, ECNs, procedures, etc. are tracked through the board.

When a change to the software is approved, a cognizant programmer tests the change in a simulator, then implements it in the DACS. The change is tested and verified. Appropriate documentation is updated.

When a change to the hardware is approved, a cognizant engineer writes an ECN, and requests personnel to make the change. The change is implemented, tested and verified. Appropriate documentation is updated.

5.0 MAINTENANCE

Several procedures have been defined for performing DACS preventative maintenance. There are two types of maintenance based on who would perform the procedure: computer maintenance and regular system maintenance. In addition, there are periodic checks to the recording system, and calibrations.

5.1 COMPUTER MAINTENANCE

Computer maintenance consists of several procedures to be performed by qualified personnel or under supervision of the cognizant programmer. The list includes:

1. Texas Microsystems PC filter cleaning
2. Texas Microsystems PC floppy disk cleaning
3. Other PC maintenance (virus checks, disk compression)
4. Modicon PLC communication checks
5. Startup and shutdown

5.1.1 PC Filter Cleaning

This procedure should be performed monthly, especially when the dust blows outside. To perform this procedure, the PC chassis must be slid out from the console racks, and they must be powered down. The procedure is given in the technical manual.

5.1.2 PC Disk Cleaning

This procedure should be performed monthly, especially when the dust blows outside, or a particular floppy drive is heavily used. To perform this procedure, the power must be turned on to the PCs. This procedure is given in the technical manual.

5.1.3 Other PC Maintenance

Other PC maintenance covers incidental items such as cleaning the video screens, replacing printer ribbons and toner, and replacing paper. These do not require formal procedures, but refer to the manufacturer's directions.

Software maintenance includes: erasing unused temporary files, performing virus checks, and defragmenting the hard disk drives. The cognizant engineer/programmer for the DACS should have methods to perform these tasks (i.e., Norton Utilities, PC Tools), and following the direction of those methods, perform them on a regular basis.

5.1.4 Modicon PLC Communication Checks

This procedure should be performed on a regular basis. This procedure is used to check the status of the PLC and can detect problems early, before they are serious. The procedure is given in the technical manual.

5.1.5 Startup and Shutdown

The computers are left on 24 hours a day to log data from the SY-101 tank. If for some reason they were shut down, ensure the instrument panel breakers are all on. The switches for the PCs are located behind their individual locked panels. These switches **SHOULD BE ON**. To ensure they are on, unlock the panels and push the power switch on. Boot procedures and software loading are left to the technical manual. The switches for the monitors are located inside the small door to the right of the screen.

It is sometimes necessary to shut the trailer down. When this happens, the proper procedure for shutting down is simply to open the circuit breakers. **DO NOT** turn off individual equipment switches. An exception to this is when the trailer is excessively hot -- turning off unused monitors is acceptable (it also conserves power).

5.2 SYSTEM MAINTENANCE

System maintenance consists of several procedures to be performed by various personnel. The list includes:

1. HVAC general maintenance (per manufacturer specifications)
2. Winterization procedures for the HVAC
3. UPS battery load tests
4. UPS HVAC general maintenance (per manufacturer specifications)
Note: there is no maintenance required for the UPS batteries
5. Winterization procedures for the UPS
6. Other
7. Startup and shutdown procedures

5.2.1 HVAC Maintenance

The HVAC system was replaced in 1994, and a private contract is in place to maintain the system.

5.2.2 UPS Battery Load Tests

There are two procedures for the UPS: one to check the functionality of the UPS and one to check the battery condition.

The UPS functionality procedure should be performed on a regular basis. This procedure is used to see if the UPS does, in fact, function when the main power is lost, and can detect problems early, before they are serious.

The battery procedure should be performed on a regular basis. This procedure is used to check the condition of the batteries and the UPS unit, and can detect problems early, before they are serious.

5.2.3 UPS HVAC General Maintenance

The manufacturer (Bard) lists no regular maintenance procedure, other than keeping the condenser clean.

5.2.4 Winterization Procedures for the UPS

The manufacturer (Bard) lists no specific procedures for winterization.

5.2.5 Other

The manufacturer's manual (Skyvane) for the weather vane and speed indicator has a yearly oiling schedule.

5.2.6 Startup and Shutdown Procedures

5.2.6.1 HVAC

The HVAC system was replaced in 1994, and a private contract is in place to maintain the system.

5.2.6.2 UPS

The UPS should be tested periodically; thus it is necessary to start and stop the UPS. The *Battery Test Procedure* (see Section 5.2.3) lists the exact order used to start and stop the UPS. Text in the procedure warns that the operator must move certain switches in an exact order, or damage will result to the UPS.

5.2.6.3 Trailer

The trailer as a whole should be shut down in this order:

1. Open the individual instrument breakers
(DO NOT turn off individual equipment switches)
2. Open the main instrument breaker
3. Open the individual utility breakers (the lights will go out)
4. Open the main utility breaker
5. Turn off the light switches (even though it is dark).
6. Close the doors and lock them, as you leave.

To start up the trailer from a cold start, perform the shutdown procedure in reverse order.

5.3 SIGNAL CONDITIONING MAINTENANCE

5.3.1 Strain Channel Signal Checks

This procedure should be performed on a regular basis. This procedure is used to check for noise in the strain signals, and for gross offset and gain errors. The check can (and did) detect problems early, before they are serious. The procedure is given in the technical manual.

5.4 CALIBRATION PROCEDURES

The following DACS equipment requires calibration:

1. Modicon B875
2. Modicon B883
3. UPS relay
4. Weather station

5.4.1 Modicon B875 Calibration

The Modicon manual for the model B875-101 analog input module provides a procedure for calibration of the internal 10-V reference. Modicon recommends annual recalibration of the reference. This procedure should be part of a larger procedure for a loop calibration (i.e., CBRS).

5.4.2 Modicon B883 Calibration

Modicon has provided a separate procedure for calibrating the model B883-200 thermocouple modules. Modicon recommends a 3-year cycle for recalibration. This procedure is extensive and should be done by a standards lab. This procedure should be incorporated into a larger (i.e., CBRS) procedure.

5.4.3 UPS Relay Calibrations

Three special relays that are energized at certain voltages. These relays need calibration. EG&G/EM has provided calibration procedures for these relays and it is recommended these relays receive annual recalibration.

5.4.4 Weather Station Calibrations

The manufacturers of the barometric pressure gauge, temperature gauge, and wind speed indicator, all have procedures for calibrating their instruments. Since the weather station has been deemed "indicator only," these instruments do not need calibrating. This may change in the future. If it does, VI file #22671 has the information to calibrate the gauges.

6.0 COMPUTER SYSTEM LAYOUT

The DACS contains several components that function in parallel to perform data collection and analysis, data display, alarm indication, monitoring, pump control, and archiving functions. Each component is carefully chosen, based on its capabilities, to perform a specific task. Their close choreography is required to meet the performance requirements of the system.

The system is designed to collect data from the field, display the data to operators, and log the data for later analysis. The system has the ability to warn the operators of critical conditions. There are control functions embedded in the system to control the position and operation of the pump. During pump control there are several checks on the critical signals to abort operation when a problem is detected. The system is designed to have enhanced effectiveness by being user friendly, and has built-in protections to make the test operation safe.

Information must be exchanged between components to integrate all of the data and control signals into a workable whole. Communication routines are implemented to read signals and data from the PLC and AF5000 devices and to send control commands to the pump equipment. The display terminals share data over a real-time data network. This network allows multiple users to share information and allows strategies to share critical signals and control commands.

6.1 COMPONENT LAYOUT

The DACS is composed of Modicon PLCs, two Eaton AF5000+ variable frequency drives, a GENESIS data collection runtime station, a GENESIS pump control runtime station, a backup station to each GENESIS runtime station, multiple local consoles for the GENESIS data collection station, multiple remote consoles for the GENESIS data collection station, a file server interface to the local area network and a station in the trailer for access to and programming of the PLC.

The Eaton AF5000+ variable frequency drives are used to control electrical motors. The units have microprocessors that have logic functions specifically designed to control the motor and provide for its safe operation. The reading of parameters and sending of control commands for the variable frequency drives can be done from the front panel of the unit or from serial communication commands. The operating status of the drive can be viewed from the front panel of the unit or it can be requested by a device that supports serial communications.

A driver has been written to communicate with the Eaton AF5000+ variable frequency drives to request data and send control commands. This driver has been specifically designed to function within a personal computer running the GENESIS multitasking software.

The GENESIS multitasking software is used to display real-time data from the field, perform logic functions, and send commands to the control devices. Each GENESIS

application is embodied in a database which is referred to as a strategy. This strategy contains algorithms that are executed at specific rates. The algorithms in the strategy are graphically selected by placing blocks in the strategy configuration environment. There are different kinds of algorithm blocks that can be used in the strategy to read data from hardware devices and write data to hardware devices. GENESIS provides control-based algorithm blocks to build control logic. The block types are denoted by an *ITALIC* font. The block tag names will be encapsulated in angle brackets; for example, <TAGNAME>.

Personal computers that use GENESIS to update a strategy from hardware devices are considered runtime stations. Real-time displays are dynamically updated from the current values of the parameters in the algorithm blocks in the GENESIS strategy. These displays are interactive and are designed to show current signal readings and allow the operators to change operating parameters. The GENESIS software is used to bring data from the field together, perform calculations on that data, and then display the results to the operators.

Each GENESIS runtime application can share its data to other GENESIS stations over a real-time data network called GEN-NET. The GEN-NET is used to transfer selected real-time data and files between personal computers running the GENESIS software. The physical layer of the GEN-NET uses ARCNET hardware in a star configuration. All GEN-NET stations are connected to an active hub with coaxial cable.

Two GENESIS runtime stations are used to pass data to and from the Modicon PLC. Information is retrieved and sent to the PLC with a communication driver that executes in parallel with the other GENESIS runtime functions. One runtime station is called STATION5 on the GENESIS GEN-NET. The GENESIS strategy executed at this station is called TEST.

The other GENESIS runtime station is used for pump control and communicates to the two variable frequency drives used to position and control the pump. This runtime station is called STATION8. It also communicates with the PLC to transfer data from the variable frequency drives. The GENESIS strategy that is executed in this station is called MOTOR.

One of the personal computers on the GEN-NET is used as a backup to either STATION5 running the TEST strategy or STATION8 running the MOTOR strategy. This station is called STATION7 on the GEN-NET.

STATION7 can also become a second console to the TEST strategy running in STATION5. As a Remote Supervisory Station (RSS) the console can bring up dynamic displays that update directly from the TEST strategy. STATION6 and STATION10 serve as RSSs to the TEST strategy at all times.

Four RSS consoles are located outside of the DACS trailer. These are known as STATION11, STATION13, STATION15 and STATION17 to the GEN-NET network software. STATION11 is located in Building MO-278, STATION13 is located in the 306E Building, STATION15 is located at the 2750 Building and STATION17 is located in the office of the software developers. All four remote stations transfer their network data packets over phone lines at 9600 baud using special ARCNET/RS-232 conversion units called Arc-Links.

The Arc-Links can only recognize odd-numbered network addresses (see 6.3.7), so all of the remote stations have odd numbers. When the remote stations were added, stations in the trailer had been assigned the network addresses 5 through 10 so the remote stations were given the next available odd numbers - 11, 13, 15 and 17.

A personal computer is dedicated to transferring files from the GEN-NET to a file server on the Hanford local area network (HLAN). STATION9 on the GEN-NET is logged into the file server and is operated in a GEN-NET file server mode. This allows files from STATION5 to be transferred to the file server without broadcasting real-time data packets over the HLAN.

STATION1 in the trailer runs the commercial package Modsoft. This software allows the PLC to be programmed and configured. It also allows the PLC registers and coils (single bits) to be viewed and manipulated online, by developers only.

Figure 12 is a diagram of the physical connections between components.

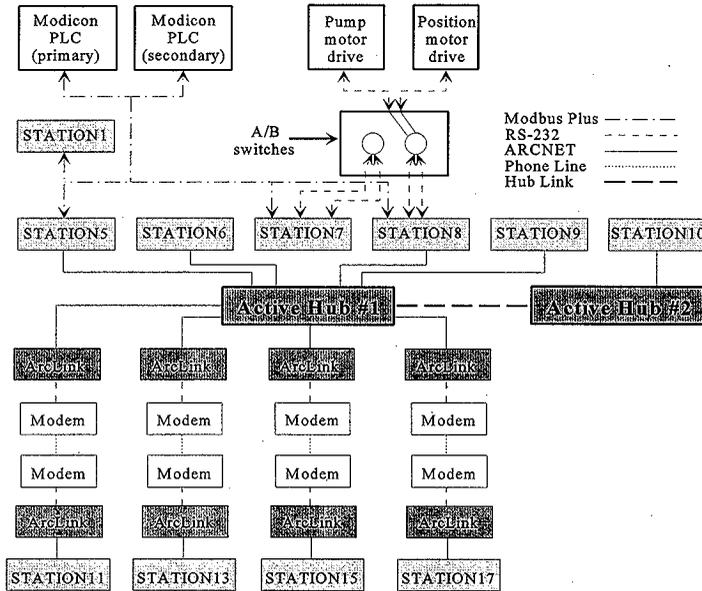


Figure 12. Diagram of the Physical Connections Between Components.

6.2 LOGICAL FLOW OF DATA

Data are retrieved from the PLC registers and status points by the station running the TEST strategy. This updates the TEST strategy algorithm blocks that are used to read data from the PLC. Data are sent back to the PLC from algorithm blocks that are used to write data to the PLC. For backup support STATION7 can run the TEST strategy; for normal operation STATION5 will run the TEST strategy. All discussion related to data flow of the TEST strategy will assume normal operation. For backup operation simply substitute STATION7 for STATION5. The MOTOR strategy is used to communicate bidirectionally with both variable frequency drives. For normal operation STATION8 will run the MOTOR strategy; for backup operation STATION7 can also run the MOTOR strategy. Data which are passed between STATION5 and STATION8 (via registers in the PLC) allow the TEST and MOTOR strategies to share critical data.

The local RSSs linked to the TEST strategy can receive and send values to the TEST strategy. The RSSs outside of the DACS trailer will only be able to receive information from the TEST strategy.

Figure 13 shows the data and control paths between components.

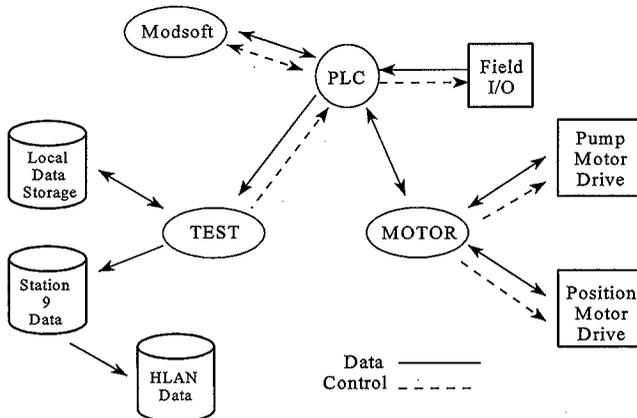


Figure 13. Data and Control Paths Between Components.

6.3 TASK ALLOCATION TO COMPONENTS

Each component performs at least one main task needed for the system. The following section is a breakdown of the tasks that are performed by each component.

6.3.1 Modicon PLC

The functions of the Modicon PLC are to input data from the field, to abort the pump when critical channels are beyond their abort limits, to provide control and alarm signals and the logic to generate them, to provide timers and enable logic for pump testing, and to provide data and status information to GENESIS.

6.3.2 Eaton AF-5000+ Variable Frequency Drives

The variable frequency drives control the motors used to position and operate the pump. Each drive controls its respective motor's critical parameters such as speed, acceleration, deceleration, and maximum speed as well as the starting and stopping of the motor. The drive also monitors current, line voltage, actual speed, and current state of the motor.

The variable frequency drives communicate with the GENESIS MOTOR strategy via RS-232 serial communications. Each variable frequency drive has a dedicated COM port for these communications. The drive that controls the pump motor is on COM1, and the drive that controls the directional motor is on COM2. GENESIS only acts in a supervisory role in controlling the motors; the drives can run the motors independent of GENESIS communication.

A special signal from each drive that indicates when the motor is running is wired directly to a discrete input module. This signal is then used by the PLC for accurate timing of the duration of a test run.

6.3.3 TEST Strategy Terminal, STATION5

STATION5 is the primary data acquisition computer and is connected to the Modicon E984 PLC via Modbus Plus. The strategy functions that STATION5 has to perform are: send and retrieve field values from the Modicon E984 I/O drops, write those engineering values to disk as GENESIS history files, participate in the watchdog timer with the E984 ladder logic, download abort limits to the PLC and generate operator alarms.

6.3.4 MOTOR Strategy Terminal, STATION8

This personal computer is dedicated to running the GENESIS Runtime MOTOR strategy. The GEN-NET node name STATION8 is used to reference this component.

The main task STATION8 performs is to communicate with the AF5000+ variable frequency drives to monitor data from the drives and send commands to them. This is performed with the Eaton AF5000+ GENESIS communications driver.

The MOTOR strategy contains logic to control the positioning of the pump. This logic uses operator input, the current status of the pump and directional motors, and the current feedback signal levels of the critical variables in the waste tank to determine what the directional motor should do. This task will command the directional motor's variable frequency drive to move the pump to the operator's desired position when all conditions are correct.

The mixing pump motor is controlled through its variable frequency drive by the logic associated with the pump operation in the MOTOR strategy. The logic uses the current values of the critical variables in the tank to decide if commands from the operators to start or stop the pump motors should be honored. If allowable, commands are sent to change the pump's operating status.

There is logic embedded in the MOTOR strategy to abort the directional or pump motor operation. This task will immediately shut down operation of either motor if any of the abort criteria becomes active. STATION8 will display alarm conditions in the alarm summary and warn the operator with an audible alarm when a signal goes into alarm state.

The tests that are performed on the waste in the tank involve operating the pump at different speeds and at different angles. A test is selected using a display of the MOTOR strategy on the STATION8 console. This selection determines a predetermined set of time durations, speeds, and angles so that the test parameters can be downloaded easily to the active parameters of the position and mixing pump control logic.

PLC communications functions are enabled in this strategy to pass variables to and from the PLC.

6.3.5 Backup Station, STATION7

This station can act as the TEST strategy runtime station or the MOTOR strategy runtime station. The tasks that are embedded in each strategy would be performed by this station if it becomes a runtime station. To make this station a runtime station, the position of a manual switch must be changed and the appropriate strategy initiated in the computer.

6.3.6 Local RSS Terminals, STATION6, STATION7 and STATION10

Remote supervisory stations (RSS) allow access to a runtime station strategy through graphic displays that were developed for the strategy. These network stations allow the operator to display and change values in the runtime strategy. All RSS terminals on the GEN-NET will be linked to the station runtime TEST strategy. There are three RSS terminals in the DACS trailer: STATION6, STATION7 and STATION10.

6.3.7 Remote RSS Terminals, STATION11, STATION13, STATION15, and STATION17

These terminals are at a site remote to the DACS trailer. They function just as a RSS terminal in the DACS trailer except that the values in the TEST strategy can not be changed from these terminals. This is done by using a different set of key macros which disable functions that would ordinarily allow access to TEST strategy values. Refer to Appendix B for additional information on the GENESIS key macros.

RSS identification numbers must be unique. The broadcast option uses the low-order bit in the identifier. If the bit has the value 1, broadcasts are allowed; if it has the value 0, broadcasts are disabled. The broadcast option is used at DACS, so a 1 is used in the low-order bit making the identifier an odd number and allowing broadcasts to pass through the ARCLINKs. For this reason, all remote RSS terminals using ARCLINKs have odd numbered identification numbers.

6.3.8 File Server Support, STATION9

The data files that are logged on STATION5 must be archived. Every two hours the data files are started over and a file transfer program is executed to transfer data files from STATION5 to STATION9. The data files are copied to a 1GB JAZ drive on STATION9 and are transferred to a file server on the HLAN. STATION9 has both an Ethernet card to access the HLAN and an ARCNET card to receive files via GEN-NET.

7.0 INSTRUMENTATION

DACS instrumentation provides measurements which can be organized in the following categories: measurements for mixer pump operation, waste tank conditions, riser strain, and DACS trailer status and area monitors. Each of these is discussed in the following sections.

Data enter the DACS via one of four general methods. Knowledge of the method used is necessary in order to determine the resolution of the data.

Most of the analog measurements enter the DACS through Modicon I/O modules which are set up to receive signals ranging from 4-20 mA, 1-5 volts, or 0-5 volts. These modules digitize the signals to a resolution of 12 bits. The range column of the I/O list gives the corresponding range in engineering units. This represents the full range of the input signal. The engineering unit range is used to scale the digitized input, and this scaled value is recorded to disk. The useful range does not always correspond to the full range. In some cases, there is an entry in the subrange column of the channel list which specifies the useful range. If the digitized input signal falls outside of the subrange, the recorded value of the measurement will be pegged at one or the other boundaries of the subrange.

The MIT17B, MIT17C, and tank bottom and side thermocouple measurements enter DACS via Modicon thermocouple modules. These modules are set to produce double precision readings with 0.1 degree F resolution as their output which is recorded without any further scaling.

Data from several analytical instruments enter the DACS via Modicon ASCII/BASIC modules. These modules have built-in BASIC interpreters. BASIC programs are written through which the modules communicate with the instruments via RS-232. The data are then placed in Modicon PLC registers in whatever format is agreed upon with the experimenters. This method allows considerable flexibility concerning the ultimate resolution of the data. The GC1, GC2, GC3, FTIR, and Photo NH3 data all enter the DACS via ASCII/BASIC modules.

Data from the variable speed drives (speed, current, and voltage) enter the DACS via the AF5000 device driver. This is the only category of data which does not enter via the PLC.

7.1 MIXER PUMP OPERATION MEASUREMENTS

DACS provides measurements which monitor the mixer pump operation. There are measurements for the operating characteristics of the pump motors and for detecting the characteristics and motion of the waste at the pump discharge nozzles and within the pump column.

7.1.1 Mixer Pump Motor Operation

These measurements provide information about the operating state of the mixer pump motor and the pump directional motor. Most of these measurements originate with the variable speed drive and first enter the MOTOR strategy over the RS-232 link. There are 12 measurements giving the motor speed (RPM), the motor voltage, current, percent of load voltage frequency and set-point speed for each motor.

In addition there are measurements for mixer pump motor oil temperature and directional motor position which enter via the Modicon PLC. There are also digital signals which detect moisture in the pump motor oil and when the pump rotational position is at one of its limits.

7.1.2 Discharge Nozzle and Pump Column

DACS provides a set of measurements for detecting the motion and characteristics of the waste as it moves through the mixer pump. These include measurements for discharge nozzle waste temperature, flow and pressure, nozzle tap pressures, and pump column pressure. There are a total of 16 measurements sampled once every 6 seconds.

7.2 TANK AND WASTE MEASUREMENTS

The bulk of DACS measurements provide information about the characteristics of the tank waste and the conditions inside the tank.

7.2.1 Tank and Waste Temperatures

There are two multifunction instrument "trees" (MIT) within the tank called MIT17B and MIT17C. Each of these instrument trees contain 22 thermocouples spaced along the length of the tree for measuring the temperature of the waste at various levels within the tank. Both MIT17B and MIT17C use type K thermocouples. These are sampled by GENESIS every 12 seconds.

In addition there are 26 type J thermocouples which measure temperatures around the bottom and side of the inner waste tank. These are sampled every 30 seconds.

7.2.2 Tank Waste Level

There are two wire level gauges indicating tank level, located in risers 1A and 1C.

7.2.3 Tank Pressures

There are two instruments which measure dome space pressure.

7.2.4 Vent Header

There are measurements for vent header flow rate, temperature and relative humidity, and hydrogen concentration.

7.2.5 Gas Concentrations

There are three gas chromatographs, called GC1, GC2 and GC3, which provide hydrogen concentration measurements to DACS. These enter the PLC as serial ASCII data through the Modicon ASCII/BASIC modules (see Section 8.3.1.3). The data from GC1 and GC2 are compiled together and enter through one of the ASCII-BASIC modules. The GC3 data is compiled with the infrared spectrometer and photo-NH₃ data and sent as another ASCII stream which enters the other ASCII/BASIC module (see Section 8.3.1.2). Information sent with the hydrogen concentrations gives the time of the sample, file ID numbers which index the sample, and hydrogen retention time.

An infrared spectrometer provides measurements of NH₃ and N₂O concentrations. This is sent along with the gas chromatograph data and includes the file ID and sample time numbers.

A separate set of measurements monitor the gas chromatograph equipment and the IR spectrometer. This information enters DACS via standard Modicon input modules.

Four hydrogen monitors which measure hydrogen concentration in various parts of the dome space and the vent header complete the set of gas-measurement equipment.

7.2.6 Miscellaneous Tank Measurements

There are measurements for the in-tank camera enclosure pressure and nitrogen supply. There are four fluid velocity measurements for measuring waste movement within the tank.

7.3 STRAIN MEASUREMENTS

There are measurements which provide strain readings for the instrument trees and the pump column. These measure the strain on these trees which can be produced by movement of the tank waste. There are measurements for low- and high-frequency strains. There are a total of 9 low-frequency strain measurements; 4 for the pump column, 3 for riser 1B, and 2 for riser 17C.

The high-frequency strain measurements (using the same transducers as the low-frequency strain measurements) are recorded during tank rollover and during pump operation using the Nicolet storage oscilloscope. If these measurements exceed alarm and abort limits, they are sent to the PLC to cause an alarm or abort condition.

7.4 TRAILER AND AREA MONITORS

Environmental measurements provide information about the DACS trailer status. Sensors inside the trailer measure rack temperatures and the power system status. Other devices outside the trailer indicate weather conditions outside the trailer such as air temperature, wind speed and direction, barometric pressure and relative humidity.

There is also an area radiation monitor which provides tank farm radiation readings and alarms.

8.0 SYSTEM FUNCTIONS AND DETAILS

The tasks that are performed by the DACS are allocated to the PLC, the Eaton AF5000+ variable frequency drives, or the GENESIS supervisory control software. The following sections present the functions and details of the DACS design. Included are discussions of the GENESIS TEST and MOTOR strategies, and the PLC, focusing upon how each of these accomplishes their various tasks.

8.1 INSTRUMENT MONITORING AND DATA COLLECTION FUNCTION (TEST STRATEGY)

The functions of the TEST strategy are to monitor the status and values read from the tank instruments, log data, generate alarms, and control the PLC. Essential to these functions is communication with the Modicon PLC, with the MOTOR strategy via the PLC, and with other stations on the network. Given below is an overview of these TEST strategy functions and the communication paths needed to accomplish them, and the details of how these functions are accomplished.

Some of these functions are accomplished by the TEST strategy blocks, some by the display screens, and some via built-in features of the GENESIS system. For those functions that are accomplished via strategy blocks, sample portions of the strategy are shown and explained in the following sections. Figure 14 is a general overview of the TEST strategy that delineates functional regions. This is provided as an aid in placing the sample strategy portions into the context of the strategy as a whole.

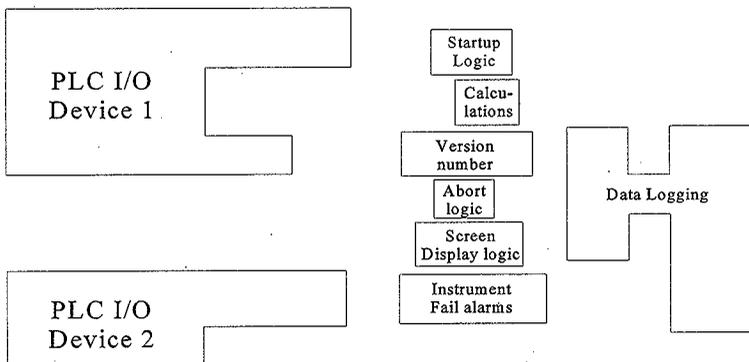


Figure 14. General Overview of Test Strategy Functional Regions.

Some of these regions correspond directly to the TEST strategy functions; others are strategy regions that perform supporting roles to these functions. Some brief comments about the regions follow:

- **PLC I/O Regions (Device 1 and Device 2):** The blocks in these are involved in communications with the Modicon PLC. This will be discussed in detail in Section 8.1.1.1.

The PLC I/O Device 1 regions contains most of the standard instrumentation I/O. It contains the abort limit blocks as well. These blocks are set to the appropriate abort limit values via a report on startup of the strategy (see Section 8.1.3). The values are then sent to the PLC where the actual abort comparison takes place.

The PLC I/O Device 2 region contains I/O for the ASCII/BASIC modules and thermocouple modules. The ASCII/BASIC modules can be used to interface a variety of serial scientific devices (see Section 8.3.1). This region contains blocks that implement the GENESIS portion of the GENESIS-to-PLC communications watchdog timer (see Section 8.1.5). Input via Device 2 is PLC status information used to monitor the status of the PLC and all of the PLC I/O modules (see Section 8.1.2).

- **Startup Logic Region:** This region contains blocks that initiate data logging and the downloading of abort limits to the PLC upon startup (see Sections 8.1.4 and 8.1.3).
- **Calculations Region:** These blocks perform calculations related to providing operator alarms and data logging (see Sections 8.1.4 and 8.1.5).
- **Version Number:** This is an enlarged dummy logic block whose tag name is set to the current strategy version number.
- **Abort Logic Region:** This region produces operator alarms when pump problems occur. It also contains the abort coil reset logic (see Section 8.1.5).
- **Screen Display Logic Region:** This contains logic pertaining to operator displays (see Section 8.1.2).
- **Instrument Fail Alarms Region:** These blocks provide instrument failure alarms; they will be discussed in Section 8.1.5.

- **Data Logging Region:** These blocks implement the Data Logging function, to be discussed in Section 8.1.4.

8.1.1 TEST Strategy Communications

The TEST strategy communicates with the Modicon PLC via the Modbus Plus network protocol. It communicates via the ARCNET with the RSS stations for operator displays, and with the STATION9 data archiving computer. Communication with the STATION8 MOTOR strategy is accomplished via the Modicon PLC and Modbus Plus. The following is a discussion of the details of each of these.

8.1.1.1 Communications between the TEST Strategy and the Modicon PLC

All of the tank instrumentation data enter the GENESIS TEST strategy from the Modicon PLC via the Modbus Plus network. These include data such as tank temperature readings, gas measurements, flow measurements, and much more. In addition to the tank instrumentation data, the PLC also provides status information about the PLC hardware and abort status information about any measurements that have aborts associated with them. It will provide instrument failure status information for selected critical measurements.

The communication with the PLC is a two-way communication, and information is sent from the TEST strategy to the PLC. The TEST strategy sends abort limit values to the PLC for use in the abort comparison logic. It will also send an operator-initiated abort reset signal to the PLC, which resets the PLC abort logic.

Communications between GENESIS and the PLC is accomplished via two device blocks, called DEV 1 and DEV 2. The device blocks accomplish the mapping between the PLC registers and the GENESIS TEST strategy I/O blocks. The device blocks together with the I/O blocks scale the data to engineering units. The data are then available to be logged to the hard disk and displayed on any of a number of user display screens.

DEV 1 is set to scale 12-bit raw data (0-4095) and DEV 2 is set to scale 16-bit raw data (0-65535). The Modicon PLC uses 16-bit registers, but most of the data from the output modules are 12-bit data. In general, DEV 1 is used for most data, and DEV 2 is used when a full 16-bit PLC register is to be transferred.

Each device block has 16 groups that can be set up to accomplish a mapping between PLC registers and GENESIS I/O blocks. For each group to be used, one must specify the type of GENESIS I/O block to be connected, the type of PLC register, the starting PLC register address, the Modbus Plus node address of the source data, and the number of connection points within the group. The GENESIS I/O block types available are *AIN*, *AOUT*, *DIN*, *DOU*, *PAIN*, *PAOT*, *PAIO*, *PDIN*, *PDOT*, and *PDIO*. In the above terminology "A" stands for analog, "D" for digital, "P" for packed, "IN" for input, "OUT" or "OT" for output, and "IO" for bidirectional input output. The PLC registers available are: CS (Coil/Status bits-0xxxxx), IS (Input/Status Bits-1xxxxx), IR (Input Registers-3xxxxx), and

HR (Holding Registers-- 4xxxxx). The starting address specifies the xxxxx part of the above registers.

The GENESIS I/O block and the PLC register type must be compatible.

Coil/Status and Input/Status registers must be paired with digital input or output blocks. Input registers must be paired with analog input blocks, and holding registers can be paired with either analog or digital input or output blocks.

The amount of data that can be transferred within one group depends upon the type of I/O block and PLC register that is used. Each group can have up to 16 connections (numbered 0-15). Nonpacked blocks transfer one register or one bit per connection depending on whether the block is analog or digital. Packed blocks can transfer 16 analog registers or 256 (16x16) digital bits. These bits could be 256 coil/status (0xxxx) bits or 16 holding registers (4xxxx) interpreted digitally. Packed communication is much more efficient and should be used whenever possible.

The device blocks have several configuration parameters which must be specified. One of these is the device # parameter. This codes for the memory base address of the SA-85 interface card. The SA-85 card is the Modbus Plus network interface card which allows the PCs to be connected to the Modbus Plus network. Although the SA-85 card can be set for memory base addresses ranging from C0000-EF800, the Modbus Plus driver only allows four choices: DD000, DD800, DE000, and DE800.

The Scan Task parameter assigns a scan number which corresponds to COM Port number (1 to 4). Although the SA-85 card does not use a COM port, the COM port corresponding to the scan task parameter must be set to INSTALLED in the COMM CNFG menu of the GENESIS configurator (under System Configuration). The Scan Period which is defined here determines how often the Modbus Plus driver is executed. The scan task parameter is currently set to 0.5 second.

The device block configuration menu also has parameters for Modbus plus network routing (not used in our system), BCD interpretation of data (also not used) and to specify the bit-order of holding registers (4xxxxx) interpreted bitwise.

The device setup parameters and the group assignments for DEV 1 and DEV 2 for the TEST Strategy are given in Tables 2 and 3.

Table 2. Device Setup Parameters for TEST Strategy DEV 1

DEV 1					
Device #: 2			Scan Task: Task: 1		
AIN Raw Count-Zero: 0			AOOUT Raw Count - Zero: 0		
AIN Raw Range: 4095			AOOUT Raw Range: 4095		
BCD: NO			Bit Order: 16-1		
Group	Start Address	End Address	Group	Start Address	End Address
G1C0	402001	402008	G6C0	476	491
G1C2	402017	402024	G6C2	508	523
G1C3	402025	402032	G6C3	524	539
G1C4	402033	402040	G6C4	540	555
G1C7	402057	402064	G6C5	556	571
G1C8	402065	402072	G6C6	572	587
G1C9	402073	402080			
G1C10	402081	402088	G7C0	220	235
G1C11	402089	402096	G7C1	236	251
G1C12	402097	402104	G7C2	252	267
G1C13	402105	402112	G7C3	268	283
			G7C4	284	299
G2C0	2001	2016	G7C5	300	315
G2C1	2017	2032	G7C6	316	331
G2C2	2033	2048	G7C7	332	347
G2C3	2049	2064	G7C13	428	443
			G7C14	444	459
G3C0	241	N/A	G7C15	460	475
G3C9	250	N/A			
			G9C0	400200	400207
G4C0	400200	400207	G9C1	400208	400215
G4C3	400224	400231	G9C3	400224	400231
G4C4	400232	400239	G9C4	400232	400239
G4C5	400240	400247	G9C5	400240	400247
G4C6	400248	400255	G9C6	400248	400255
G5C0	2052	2067	G10C0	2150	2165
G5C1	2068	2083			
G5C2	2084	2099			
G5C3	2100	2115			
G5C4	2116	2131			
G5C5	2132	2147			

Table 3. Device Setup Parameters for TEST Strategy DEV 2

DEV 2					
Device #:		2	Scan Task:		Task: 1
AIN Raw Count-Zero:		0	AOUT Raw Count - Zero:		0
AIN Raw Range:		65535	AOUT Raw Range:		65535
BCD:		NO	Bit Order:		16-1
Group	Start Address	End Address	Group	Start Address	End Address
H1C0	213	N/A	H3C0*	400108	N/A
			H3C1*	400109	N/A
H2C0	402129	402136	H3C2*	400110	N/A
H2C1	402137	402144	H3C3*	400111	N/A
H2C2	402145	402152			
H2C3	402153	402160	H4C0*	400112	N/A
H2C4	402161	402168	H4C1*	400113	N/A
H2C5	402169	402176	H4C2*	400114	N/A
H2C6	402177	402184	H4C3*	400115	N/A
H2C7	402185	402192			
H2C8	402193	402200	H5C0*	400116	N/A
H2C9	402201	402208	H5C1*	400117	N/A
H2C10	402209	402216	H5C2*	400118	N/A
H2C11	402217	402224	H5C3*	400119	N/A
H2C12	402225	402232			
H2C13	402233	402240	H6C0*	400120	N/A
H2C14	402241	402248	H6C1*	400121	N/A
			H16C0	409000	N/A

*Modbus Plus driver allows only the first 4 connections using PDIN blocks with 40xxxx registers.

Figure 15 shows a portion of the PLC I/O section of the STATION5 strategy. For efficient communications, most I/O is done via packed blocks. The packed blocks are connected to the device block. To allow each channel to have its own tag name and description, regular AIN blocks are connected to the packed blocks as shown in the figure. The I/O blocks are (with few exceptions) organized by device block group such that each horizontal row in the PLC I/O area contains the I/O for one device block group starting with Group 1 at the top.

The packed blocks are named to reflect their device block group and connection number within the group. Thus, the PAIN block <G1C2> is connected to Group 1, connection 2 of device block 1. The packed blocks connected to device 1 are named beginning with the letter "G." Those connected to device 2 begin with the letter "H." Thus, a block called "H6C3" would be connected to Group 6, connection 3 of device block 2.

The regular *AIN* blocks connected to the packed blocks are arranged by connection number of the packed block starting at the upper left and reading down. Thus, in Figure 15, <PITNO111> is connection 1, <PITNO110> is connection 2, <TT10001> is connection 3, <PITMSY13> is connection 4, <MT10001> is connection 5, <TITMSY15> is connection 6, <FTE50002> is connection 7, and <LIR01A> is connection 8. This convention has been followed throughout.

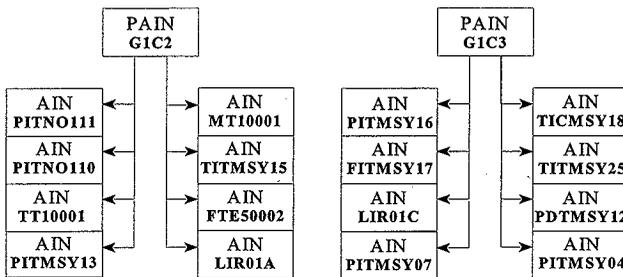


Figure 15. PLC I/O Strategy Detail.

The *PAIN* blocks include range information for the incoming raw values that allow them to be scaled to engineering units. This scaling information is duplicated in the *AIN* blocks. In addition, the *AIN* blocks are configured to include the engineering units, the measurement description, alarm values (if any), and scan period.

The scan period specifies the time between measurement updates. GENESIS only allows scan periods of 0.05, 0.1, 0.25, 0.5, 1, 2, 6, 12, and 30 seconds. However, the scan period actually achievable depends upon the number of blocks in the system and the communication rate with the PLC. In addition, the overall communication scan period for each communication link must be specified. The scan periods of individual blocks cannot exceed the overall communication scan period. The communication link with the PLC is currently set to a 0.5 second scan period. Most of the measurements are set for 6-, 12-, or 30-second scan periods.

There are two cases where additional manipulation is needed to present the data in engineering format. The measurements are <ARMGAMMA> and <PHO-NH3>. The signal sent to GENESIS for <ARMGAMMA> is actually an exponent, so a calculation in a GENESIS *F(X)* block is done to create the actual measurement value from the exponent. The signal <PHO-NH3> is sent originally as two halves, a most significant and least significant

portion. These are combined into one measurement value in the TEST strategy using a $F(X)$ block.

8.1.1.2 Communications with the MOTOR Strategy

There is no direct communication path between the TEST and MOTOR strategies. Instead, communication between them is accomplished via the PLC and Modbus Plus. Data relating to the variable speed drives for pump motor control and directional motor control are sent from the MOTOR strategy to the PLC, then are relayed to the TEST strategy.

Also, the TEST strategy receives alarm and abort limit values for pump motor speed, pump motor current and pump discharge pressure. These values are calculated in the MOTOR strategy and sent to the PLC. The actual abort and alarm comparisons are done in the PLC, but the calculated limit values are sent from the PLC to the TEST strategy.

Also, both the TEST and MOTOR strategies have watchdog timers to monitor the communication between the strategies and the PLC. When one of these communication paths fails, the other strategy is notified of the communication failure.

8.1.1.3 Communications from the TEST Strategy to the RSS Stations

The TEST strategy communicates over the ARCNET to RSS stations. These stations can display any of the TEST strategy screens for data monitoring.

8.1.1.4 Communications from the TEST Strategy to the Data Archiving Computer

Data are first logged locally on STATION5 by the TEST strategy. The data are then transferred to the data archiving computer, STATION9. After a successful transfer, the data are then deleted from STATION5. The computers are linked with ARCNET, and the transfer is accomplished by using a *GENFXR* block located in the TEST strategy. The data archiving computer is networked to both ARCNET and to the HLAN. The data are transferred from STATION5 directly to the network server as well as to the JAZ drive on STATION9.

8.1.2 Data Monitoring

Data monitoring is one of the primary functions of the TEST strategy. The data and status monitoring functions of the TEST strategy are accomplished by displaying information to the operators on any of a number of displays including GENESIS-provided system runtime trend and history displays. In addition to providing the monitoring function, some displays allow the operator to input control signals that can initiate processes in GENESIS or ultimately in the PLC. In addition, all displays have a button which, when clicked on with the mouse, allow other displays to be brought up on that particular monitor.

Displays can be brought up independently on the STATION5, STATION6, STATION7 and STATION10 computers in the DACS trailer, as well as the four remote stations (STATION11 in Building MO-278, STATION13 in the 306E Building, STATION15 in Building 2750 and STATION17 in the office of the software developers). All data that travel to and from the remote displays (in RSS stations) go through the master computer (STATION5) via ARCNET.

The displays are organized in a hierarchy based on their function; other displays can be accessed from the current one. At the top of the hierarchy is the WELCOME display. This display appears when the system is first brought up. It has buttons that allow access to the MAP, CSMAIN, MSMAIN, and ASMAIN displays.

- The MAP display shows a representation of the hierarchy and allows access to any other display in the system.
- CSMAIN is the overall abort status monitoring display. It allows access to the other displays listed in the "Abort Status Monitoring Screens" Table 4a, which provide more detailed information about the abort status of various instruments. All the members of this group include a button that allows the user to reset the PLC abort coil. This function is discussed in Section 8.1.5.
- MSMAIN has a schematic representation of the tank showing the locations of the various instrument trees. It allows the user to bring up displays that show data from the instruments on these trees. Its buttons allow the user to call up other tank measurement monitoring displays which present the data in a variety of formats. These displays are listed in the "Tank Measurement Monitoring Screens" Table 4b, excluding those indicated as runtime trend screens.
- ASMAIN allows the operator to access runtime trend and history displays. These displays are also listed in the "Tank Measurement Monitoring Screens" Table 4b. Runtime trends plot various tank measurements as a function of time, beginning with the point when the screen is first brought up. The graph grows from this point until the time allotted for the graph has passed. After that, the older values are scrolled off the edge of the graph as new values become available. If another screen is brought up on the same display, then the process must start over again. This differs from the system runtime screen discussed below.

Accessible from the MAP display are those displays listed in the "System Status Monitoring and Miscellaneous Screens" Table 4c. IOSTATUS and DACS allow monitoring of the PLC racks and modules and the trailer status respectively. The displays named ABRTENAB, ABRTNAME, and ABRTCHEK are used for monitoring and/or enabling/disabling coils used for aborts by the PLC. MININ1 and MININ2 are for

convenience and can be used to check minimum instrument status. The "other" displays are non-monitoring screens, MAP and WELCOME, discussed above.

All of the displays have a similar header bar. The header bar has the title, the current time and date, the application name and version number, and buttons that allow the user to bring up the MAP, CSMAIN, MSMAIN, or ASMAIN displays as well as the display one level above in the hierarchy (by pressing the Page Back button). The displays have been designed so that anything in blue can be clicked on with the mouse. These are either buttons that bring up another display or measurement values that bring up a tag details sub-window. The tag details sub-window appears at the bottom of the display and gives further information about the measurement.

GENESIS provides two other facilities useful for data monitoring. One is the system runtime trend. This allows the display of runtime graphs for up to 20 predefined measurements. These differ from the runtime trend screens discussed above in that the prior data are not lost when the display is exited momentarily.

Additionally, GENESIS provides the ability to examine, graphically or in tabular format, any data that are being logged to a history file on the current computer. STATION5 is the only computer that has history files available, so this feature only works on STATION5. Data logging will be discussed further in Section 8.1.4.

Table 4. TEST Strategy Screens

SCREEN	SCREEN TITLE	COMMENTS
CSMAIN	Automatic Alarms and Aborts	Overall alarm and abort summary display
HVTALARM	Hydrogen-Vent Header Tank	Alarm and abort details
MANABRT	Manual Aborts	H2 and NH3 concentrations
PUMPALRM	Pump Parameters	Alarm and abort details
STRNALM	Strain Gauges	Alarm and abort details
TEMPALM	Temperature	Alarm and abort details

Table 4b. Tank Measurement Monitoring Screens		
SCREEN	SCREEN TITLE	COMMENTS
ASMAIN	Runtime Trend Selection Screen	Selection screen for runtime trend displays
GASSUM	Gas Summary	Collected gas concentrations and related items
MIT17B	Instrument Tree (1) Riser 17B (MIT)	Tank Temperatures
MIT17C	Instrument Tree Riser 17C (MIT)	Tank Temperatures
MSMAIN	Main Riser Profile	Selection screen for monitoring displays
NEW72HR	Current 72 hour history trend.	History trend - critical measurements
OLD72HR	Previous 72 hour history trend.	History trend - critical measurements
PUMP	Mixer Pump Riser 12A	Mixer pump instrument readings
PUMPOPS	Pump Operations	Runtime trend - pump operation measurements
ROLLOVER	Imminent Rollover	Runtime trends and temperature profiles to monitor for possible rollover
SUMMARY	Summary Information	Vent Header, In-tank parameters, MIT temperatures
TBSTC	Tank Bottom and Side Thermocouples	Temperature readings
TEMPRFL	MIT Temperature Profiles	Risers 17B and 17C temperature bar graphs

Table 4c. System Status Monitoring and Miscellaneous Screens		
SCREEN	SCREEN TITLE	COMMENTS
DACS	DACS Facilities Management	DACS trailer power, trailer temperature and weather station
IOSTATUS	I/O Health Status	Modicon rack and module status
MAP	Map	Shows organization of all screens and allows operator to call up any screen
TESTEXIT	Exit to DOS?	Allows exit with level 1 password
WELCOME	DST 241-SY-101 Data Acquisition and Control System (DACS)	Screen that appears upon system startup
ABRTENAB	Abort Enable Checklist	Allows enable/disable of PLC abort coils and indicates coil status
ABRTCHEK	Abort Limit Checklist	Used for pump run
ABRTNAME	Abort Coil Tag Names	Identifies PLC abort coils and indicates coil status
MININ1	Minimum Instrumentation Checklist #1	Convenience
MININ2	Minimum Instrumentation Checklist #2	Convenience

8.1.3 PLC Control

The TEST strategy provides limited operator control of the PLC. Specifically, from any of the abort monitoring screens the operator can reset the PLC abort coil. The pump cannot run when this coil is set. This abort coil is latched, which means that if a measurement exceeds its abort limit only momentarily the abort coil will remain set even after the measurement has returned to a normal value. The only way to reset the abort coil is via the button on the abort monitoring screens. This function was discussed in Section 8.1.2 and the strategy logic to accomplish it will be shown and discussed in Section 8.1.5.

The TEST strategy is also responsible for downloading the abort limits to the PLC. These are the values that the PLC will compare the incoming measurement value with to determine if an abort condition exists. If one of these measurements exceeds its abort limit then the PLC abort coil is set. The abort limits are sent via *AIN* blocks located in the PLC I/O section of the strategy. These *AIN* blocks get their values from a report SETLIMS.RPS, which contains the actual abort limit values. This report is run at startup to load the *AIN* blocks with the abort limit values. The only exceptions to these are those limits that change based upon the pump speed. These limits are calculated in the MOTOR strategy and sent to the PLC when a new test is selected.

Figure 16 show the startup logic. It shows the mechanism for generating a pulse immediately upon startup as well as a delayed startup pulse. This is implemented by connecting output of the <ON> *DIN* block to the <PULON> *pul* block and the <DELAYON> *TON* block. The <ON> block is wired to always be 1. The <OFF> block is wired to be always 0. Since at startup, the inputs of all blocks are initialized to 0, the only time the <PULON> and <DELAYON> blocks will see a 0 to 1 transition on their inputs is at startup. This will trigger the *pul* block <PULON> and cause the *TON* block <DELAYON> to produce its output after the programmed 45-second delay elapses.

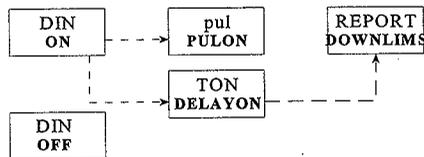


Figure 16. Startup Logic.

The <PULON> block initiates the history file naming the phasing process that was discussed in Section 8.1.4. The *TON* block <DELAYON> is the block that runs the DOWNLIMS report via the <DOWNLIMS> *REPORT* block. The 45-second delay is necessary to allow time for GENESIS to PLC communication to be established before the abort limits are downloaded.

The <OFF> block is used to prevent data logging in a *HIST* block (see following section 8.1.4).

8.1.4 Data Logging

The TEST strategy is responsible for logging data to the hard drive and transmitting the data files over the network to the archiving computer, STATION9. The data are used for long-term tank monitoring and to evaluate the results of the pump tests. Approximately 160 data channels are logged. These are stored in 10 history files at intervals ranging from 6 seconds to 5 minutes.

The TEST strategy creates these history files and assigns file names derived from the time and date the file is created. Data are logged until an even-numbered 2-hour boundary is reached (that is, 2:00, 4:00, 6:00, etc.). At this time the current set of history files is closed and a new set is opened. The old set of history files is then transferred to the STATION9 archiving computer and placed on the HLAN. Upon successful transfer the files are deleted from the STATION5 computer disk. If the file transfer is not successful the files remain on the STATION5 disk and another attempt is made 2 hours later. The files accumulate on the STATION5 disk until the network problem is resolved and they can successfully be transferred to STATION9.

The data logging portion is accomplished in GENESIS by the use of *HIST* blocks. The files are known in GENESIS terminology as history files. Figure 17 shows an overall view of the TEST strategy history/data logging section. More detailed views of portion of this strategy section are shown later.

The top twelve large blocks of the rightmost column in the diagram above are the *HIST* blocks. The remainder of the blocks are involved with timing the creation of the history files, phasing the creation of the files and logging of the data to avoid system overload, naming the files, and copying the files automatically over the network and to different directories on the system.

Two of the twelve *HIST* blocks are not used for the standard data logging which is described below. The blocks <OLD72HR> and <NEW72HR> provide a way to have at least 72 hours of readings available which can be viewed using the history display capabilities of GENESIS. These history files contain measurements which can characterize the tank status for the time between pump runs when the DACS trailer is not manned.

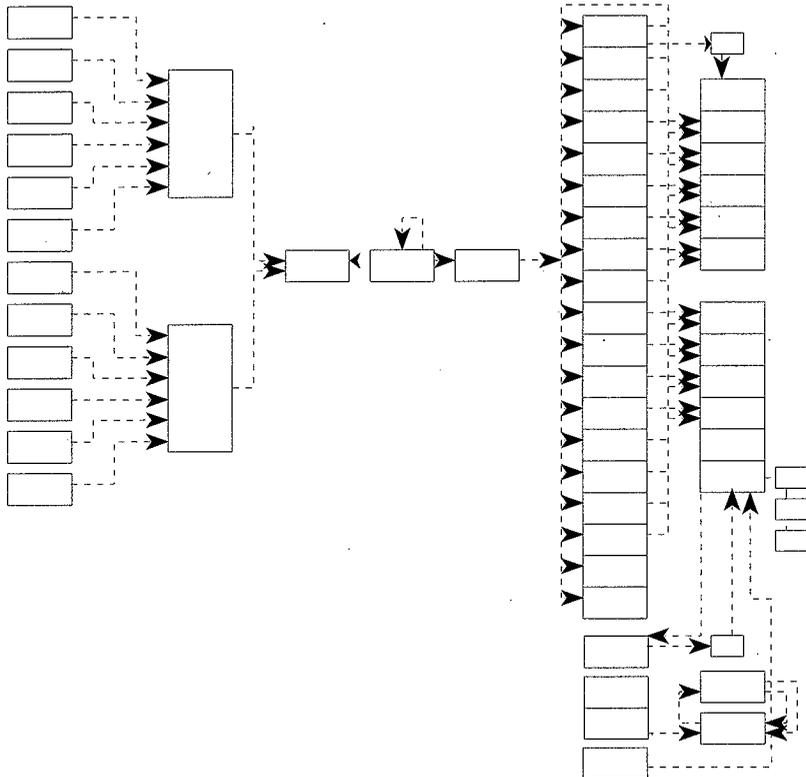


Figure 17. History Data Logging Overall View.

The functionality of the history section of the strategy is by column, and the general flow of the logic is left to right. The leftmost column consists of *TIMER* blocks controlling the timing of the opening and closing of the history files. These feed into two *OR* blocks that feed into a *REPORT* block. The *REPORT* block runs a report (*HISTFNAM*) that creates file names for the about-to-be-opened history files based upon the date and time.

When this process is complete the report sets the *pul* block, the next block to the right above. This block generates a pulse that causes the *SHOT* block (the next one over) to generate a one-shot pulse. This pulse starts the *TON* blocks (the long column of 19 blocks). It is detrimental to start all the history blocks at the same time, so for this reason, the *TON* (timer-on-delay) blocks lie in-line with every history block. Each *TON* block has a different delay, spreading the load on the hard disk drive over the total time interval. These delay timer blocks are set to produce their outputs at varying times from the start of the one-shot pulse. They implement the phased stopping and starting of the history files by feeding into the start and stop inputs of the *HIST* blocks, the rightmost column of blocks. The one exception to this is the <RGA5> block. It is started when new data is received from the RGA5 computer and stopped after it has logged just one sample. This is done to avoid multiple logging of the same data.

These blocks control the timing of the running of another report (*HISTCOPY*), which copies history files to two directories on the disk \HIST and \HIST2. The *GENFXR* block at the bottom of the rightmost column is controlled by the *TON* blocks. The *GENFXR* block copies history files over the network. More details concerning this process follow.

Figure 18 shows a portion of the history strategy section that initiates a history file changeover. The system is designed to produce history files that contain 2 hours worth of data each. Every 2 hours throughout the day, the current set of history files is closed and a new set is opened. The timer blocks are set to create a pulse at a specific time of day. In this case, the timer blocks are set to create their pulses on the hour spaced 2 hours apart. When one of the timers goes off, the signal feeds through the *OR* block and causes the report *HISTFNAM* to be run.

This report implements the history file-naming convention. The files are named "ymdhFFFF.PRN," where

y	is the year, with 2=1992, 3=1993, etc.
m	is the month, numbered 1-9 and A-C
d	is the day, numbered 1-9 and A-V
h	is the hour, numbered 0-9 and A-N, and
FFFF	is the file code that can be any one to four alphanumeric characters.

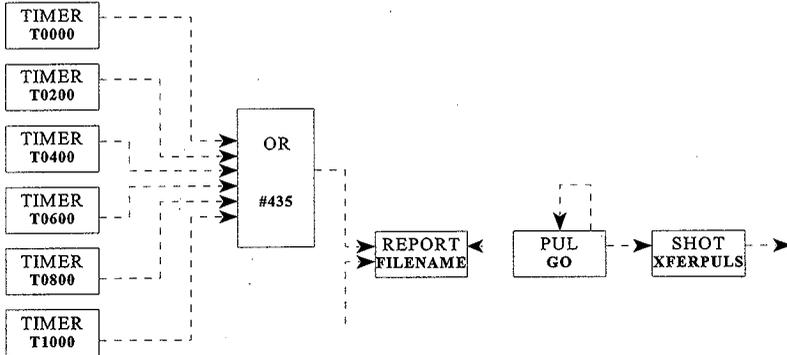


Figure 18. Timer, Report, and Pulse Portion of History Strategy Section.

HISTFNAM reads the system time/date and creates these file names. It loads the file name field of the *HIST* blocks with the new file name. It then sets the *PUL* block <GO> which in turn initiates the one-shot pulse from the *SHOT* block <XFERPULS>. The HISTFNAM report is run at system startup. The history files, sample periods and file contents are summarized in Table 5.

Figure 19 shows some of the *TON* blocks that implement the phased starting and stopping of the *HIST* blocks. This phasing is necessary in order to spread out the load on the system imposed by the data logging operations. The system has limited processing time available for data logging as well as limited buffer space for file operations. If the strategy attempts to write too much data at once, this buffer overflows and a system message appears indicating a loss of data.

Table 5. History File Names and Contents

File Name	Sample Period (sec.)	Contents				
ymdhPUMP.PRN	6 or 300	PCR12A01, VR232040,	PDPBASE, VR232050,	PITNO111, ZIMPE112,	PITNO110, ZIMPE142	VR232020
ymdhMIT2.PRN	12	TIR17B09, TIR17B14, TIR17C12,	TIR17B10, TIR17B15, TIR17C13,	TIR17B11, TIR17C09, TIR17C14,	TIR17B12, TIR17C10, TIR17C15	TIR17B13, TIR17C11,
ymdhTBS1.PRN	300	TBSTC01, TBSTC06, TBSTC11,	TBSTC02, TBSTC07, TBSTC12,	TBSTC03, TBSTC08, TBSTC13	TBSTC04, TBSTC09,	TBSTC05, TBSTC10,
ymdhGAS.PRN	150	FT-FILE, FT-NH3C, PHO-TIME,	FT-TIME, GC3-TIME, PHO-NH3,	FT-N2OA, GC3-FILE, RGA5TND1,	FT-N2OC, GC3-RT, RGA5TND2	FT-NH3A, GC3-H2,
ymdhRGA5.PRN	per sample	RG-RUN, GC1-RT, FITMSY17, PDTMSY12,	RG-STAT, GC2-H2, TICMSY18, PITMSY07,	RG-TIME, GC2-AREA, PITMSY13, PITMSY10	GC1-H2, GC2-RT, TITMSY15,	GC1-AREA, PITMSY04, PITMSY16,
ymdhMIT3.PRN	12	TIR17B16, TIR17B21, TIR17C19,	TIR17B17, TIR17B22, TIR17C20,	TIR17B18, TIR17C16, TIR17C21,	TIR17B19, TIR17C17, TIR17C22	TIR17B20, TIR17C18,
ymdhTBS2.PRN	300	TBSTC14, TBSTC19, TBSTC24,	TBSTC15, TBSTC20, TBSTC25,	TBSTC16, TBSTC21, TBSTC26	TBSTC17, TBSTC22,	TBSTC18, TBSTC23,
ymdhSTRN.PRN	150	WIR12A01, WIR17C02,	WIR12A02, WIR1BA01,	WIR12A03, WIR1BA02,	WIR12A04, WIR1BA03	WIR17C01,
ymdhVOL.PRN	12	LIR01C, PIR17C01, NITJSY06, WSWDIR,	FTE50001, PIR17B04, NIR17B01, WSWSPD,	FTE50002, FTE50003, WSP1, TIR12A01,	TT10001, NIRO5A01, WSH1, TIR12A02,	MT10001, NITKSY06, WST1, LIR01A
ymdhMIT1.PRN	12	TIR17B01, TIR17B06, TIR17C03, TIR17C08	TIR17B02, TIR17B07, TIR17C04,	TIR17B03, TIR17B08, TIR17C05,	TIR17B04, TIR17C01, TIR17C06,	TIR17B05, TIR17C02, TIR17C07,
NEW72HR.PRN	300	FTE50001, GC3-H2S, WYR17C01,	FTY50002, GC2-H2S, FTE50003,	PYR17B04, NIR17B01, WYR12A01	PYR17C01, WYR1BA01,	LIR01C, WYR17B01,

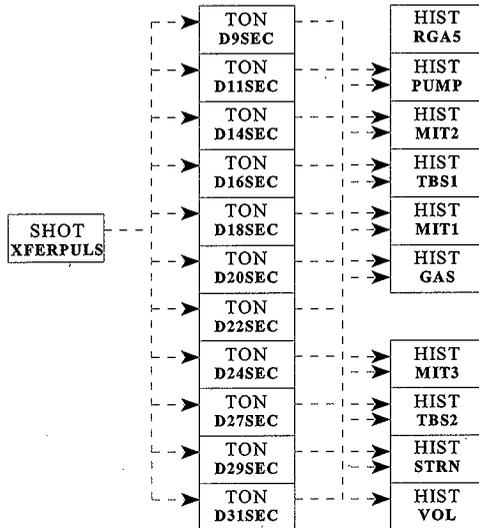


Figure 19. Phasing and HIST Blocks.

Besides the buffer overflow problem, the system resources are strained when files are opened and closed. The phasing scheme is designed to spread out the load on the system as much as possible. The scheme provides that during file changeover, no two files are started or stopped simultaneously. In addition, between changeovers, the sampling of data should be spread out as much as possible. In the current system, no more than two *HIST* blocks sample their data in any one scan period. The scheme is dependent upon the sample period in the *HIST* blocks. If the sample period of any of the blocks is changed, the phasing scheme must be adjusted accordingly. Currently, the changeover occurs as follows:

Second	Action
0	PUMP Stop
1	PUMP Start
5	MIT2 Stop
7	TBS1 Stop, PUMP Sampled
9	MIT2 Start
11	MIT1 Stop
13	PUMP Sampled
14	TBS1 Start
16	GAS Stop
18	MIT1 Start
19	PUMP Sampled
21	MIT2 Sampled
22	GAS Start
24	MIT3 Stop
25	PUMP Sampled
29	MIT3 Start
30	MIT1 Sampled
31	TBS2 Stop, PUMP Sampled
33	STRN Stop, MIT2 Sampled
35	TBS2 Start
37	PUMP Sampled
38	VOL Stop
40	STRN Start
41	MIT3 Sampled
43	VOL Start, PUMP Sampled

Given the start times above for each one of the files and the data sampling periods, one can determine when each of the samples are taken. In this case, for a 12-second period, the samples are:

Second	Files Sampled
1	PUMP
2	TBS1 (every 25th pass beginning with pass 2)
4	GAS (every 25th pass beginning with pass 15) STRN (every 25th pass beginning with pass 4)
5	MIT3
6	MIT1
7	PUMP, VOL
9	MIT2
10	GAS (every 25th pass beginning with pass 2) STRN (every 25th pass beginning with pass 17)
11	TBS2 (every 25th pass beginning with pass 3)
12	
1	(pattern repeats at this point)

The RGA5 file does not sample regularly as the others do. This file is opened, a sample taken, and closed again only when a new sample is received.

Figure 20 shows the *GENFXR* block and an *OR* block. It also shows a few of the *TON* blocks. The *TON* block *D1MIN* initiates the report, *HISTCOPY*, 1 minute after the file changeover begins. This report copies the recently closed files to both the \HIST and \HIST2 directories then deletes them from the \TEST directory. This is done to avoid having history files build up in the directory. The *HISTCOPY* report renames (using the history file naming convention) copies, then deletes the alarm and event summary files.

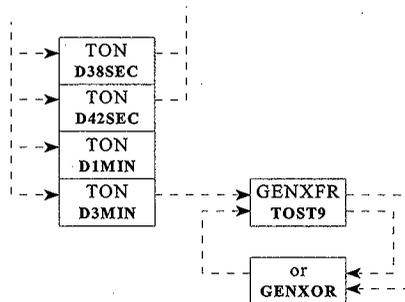


Figure 20. GENFXR Detail.

The *GENFXR* block is set up to copy the history files over the network. This block is initiated 3 minutes after a file changeover begins. The first pass attempts to copy files from the \HIST directory to the \HIST directory on drive C: of STATION9. If this file transfer is successful, the files are deleted from the \HIST directory on STATION5.

The *OR* block detects one of two status outputs: either successful transfer or failure. In either case, it means that the first operation is complete. This signal is used to initiate the second file transfer operation in the *GENFXR* block. This time, files are copied from the \HIST2 directory over the network to the Jaz drive (Drive E:) of STATION9. Once again, the files are deleted from the STATION5 directory if the transfer is successful.

The history file ymdhRGA5 differs from the others in that a new sample is logged only when a new reading is received by GENESIS. This avoids duplication of samples in the file. The instruments are set to produce a new sample every three minutes. If the instruments are not working properly or if there is some other problem, no data will be logged and the file could be empty. The file is still closed every two hours no matter how many samples have been taken during that time.

A discussion of the logic which implements this follows. The *F(X)* block, <FILEUPDT>, continually scans the <RG-RUN> tag. The value of this tag will change whenever a new sample is received. The <FILEUPDT> block contains logic to compare the previous value of <RG-RUN> with the current value. It sets one of its outputs (DO3) to be high (1) if the current and previous scans are equal, and low (0) otherwise. When this output transitions from 0 to 1, it starts a 45 second timer (*TON* block <RGSTLOG>). When the 45 seconds have elapsed, the sample is logged (by starting data logging for the <RGA5> *HIST* block. This 45 second delay is to insure that the value of PHO-NH3 which is calculated from MSB and LSB portions, has enough time to execute before the values are logged.

The *HIST* blocks <OLD72HR> and <NEW72HR> provide a way to have at least 72 hours of readings available which can be viewed using the history display capabilities of GENESIS. These history files contain measurements which can characterize the tank status for the time between pump runs when the DACS trailer is not manned. Only <NEW72HR> is set to actually log data. When a 72 hour period has elapsed the tag <CHNG72OK> is set. The report HISTCOPY.RPS detects this and copies the file NEW72.PRN to OLD72.PRN. New data is then logged to the NEW72.PRN file. The data in the OLD72.PRN file can be examined using the history file display capabilities of GENESIS.

8.1.5 Alarming Functions

An important function of the TEST strategy is to notify the operator of significant events. The system has been designed to produce alarms when critical tank measurements approach safety limits, when critical instruments fail, when communications have been lost between the TEST strategy and the PLC, between the TEST strategy and the MOTOR

strategy, and when the PLC has detected an abort condition. This is accomplished by using the alarm features of the GENESIS software. GENESIS provides a special screen called the alarm/event summary screen. Alarms, operator actions, and other system events are posted to this screen as they occur. In addition, the PC speaker sounds when an alarm arrives. These speakers have been amplified in the trailer producing an effective audible alarm. The operator must hit a keyboard key to silence the alarm. Information about the alarm can be found on the alarm/event summary screen. From there the alarms can be acknowledged by the operator.

Most of the alarms are generated by measurement I/O blocks. The alarm values are programmed into the block and the alarm priority is set. When the measurement value exceeds one of the alarm limits the alarm occurs. Other alarms require additional strategy logic. Portions of the TEST strategy related to the alarming function are shown in Figure 21 and discussed below:

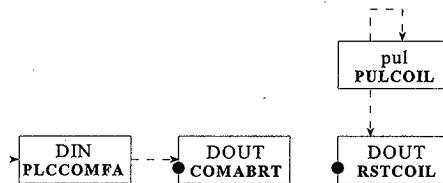


Figure 21. Abort Logic Detail.

This strategy section is responsible for causing a GENESIS alarm if a PLC communication failure is detected. It also contains logic for resetting the PLC abort coil.

Since the PLC abort coil remains latched following an abort, the operator needs some means of resetting the coil after the problem has been resolved. This is provided by the <PULCOIL> and <RSTCOIL> blocks shown above. When the Reset Abort Coil buttons on the abort coil status screens (see Section 8.1.2) are clicked upon, they cause the <PULCOIL> block to produce a pulse. This is sent via the <RSTCOIL> digital output block to the PLC where it causes a reset of the PLC abort coil, as well as all of the individual aborts in the PLC.

The <PLCCOMFA> block is set when GENESIS detects a communication failure from the PLC to GENESIS. This can be detected by the Modicon driver and accessed by a connection to the FAIL bit of an I/O block. The <PLCCOMFA> block causes a GENESIS alarm whenever this condition occurs. It also attempts to send this information to the PLC so that the abort coil can be set. This is accomplished via the <COMABRT> block shown above. This is useful in the unlikely event that the communication failure has been only one way and the PLC has not detected a corresponding GENESIS to PLC failure.

Figure 22 shows a portion of the strategy that generates instrument failure alarms. These blocks generate alarms when a critical instrument reads at or near zero indicating a possible instrument failure. They are also wired to the instrument failure detection mechanisms in the PLC (see Section 8.3.2) and cause an alarm when the PLC detects an instrument failure as well.

Each of the *OR* blocks shown in Figure 22 implements an instrument fail alarm for the measurements whose names correspond most closely to the names of the *OR* blocks. In most cases, the measurement name has an I in place of the Z in the above names.

Five of these instrument failure alarm blocks have to be handled differently. These are the ones shown as <GC1-INST>, <GC2-INST>, <GC3-INST>, <NH3-INST>, and <FT-INST> in Figure 22. These originate from the gas measurement equipment and arrive at the PLC after having traveled through several other computers and devices. An instrument failure in this case could mean either a failure of the instrument itself or of the communication pathways involved in getting the measurement to the PLC.

The first case is handled by detecting when the measurements involved read zero. This is slightly complicated by the fact that <GC1-H2>, <GC2-H2> and <GC3-H2> already have low alarm values associated with them, so separate blocks (<GC1-ZVAL>, <GC2-ZVAL>, and <GC3-ZVAL>) have been provided to detect the zero case.

The second case is handled by the logic shown in Figure 23. It is used to detect a failure in the communications path between the instrument and the PLC. The gas measuring equipment sends a value to the PLC known as the File ID. This value is incremented (independently for FTIR, PHOTO NH3, GC1, GC2 and GC3 data) whenever a new measurement has been taken. The *F(X)* block, <FILEUPDT>, receives this file ID from the measurements. As long as the current value is equal to the previous value, the *TON* blocks are allowed to run. When a new File ID arrives, the *TON* blocks are reset. If the *TON* blocks ever reach their predefined timeout values, the instrument failure alarm occurs. The block <RGSTLOG> is used to start the RGA5 data logging. It was discussed in section 8.1.4.

Values for FTIR, PHOTO NH3, GC1, GC2, and GC3 are forced to zero by GENESIS if the *TON* blocks time out. This is an obvious indication to operators that the instrument has a problem. The gas values are actually forced to zero in the PLC ladder logic by use of a PDOT block (G10C0). As soon as updated data is sent to the PLC from the gas equipment, the *TON* blocks will reset and the display will indicate valid readings.

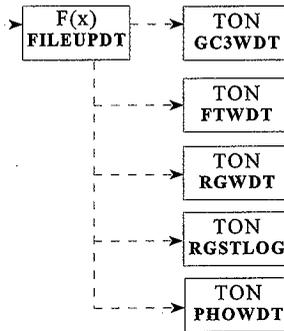


Figure 23. Communication Pathway Failure Detection.

8.2 PUMP CONTROL FUNCTION (MOTOR STRATEGY)

The signals that actually control the pump are assigned by the two Eaton AF5000+ variable frequency drives. There is one drive to control the motor that positions the pump and one drive to control the motor that operates the pump. These variable frequency drives receive commands from the GENESIS MOTOR strategy (using RS-232 asynchronous communications) in the form of ASCII strings from a communication driver. The MOTOR strategy has logic functions that control pump position, pump operation, error checking, operation aborting, test selection, network communications, and alarming. The ASCII driver is imbedded in the GENESIS MOTOR strategy and is represented by two AF5000+ hardware communication device blocks, one for each VFD.

As in the TEST strategy, the MOTOR strategy includes a number of dynamically linked display screens, shown in Table 6. The MOTOR strategy displays permit the operators to select or set up specific mixer pump tests, and to start, monitor, and stop the tests. Diagnostic information is also included on some displays.

Table 6. MOTOR Strategy Screens

SCREEN	SCREEN TITLE	COMMENTS
DMPROB	Position Pump Problem	Help screen - positioning problem display.
MAINMENU	Main Test Menu	Menu display to select other displays.
PBENPROB	Test Enable Problem	Help screen - Test enable problem display.
PSPROB	Test Start Problem	Help screen - Test start problem display.
PUMPRUN	Pump Operation	Combined directional/run pump operation display.
SETPROB	Test Values Set Problem	Help screen - Test values set problem display.
TESTSET	User Test Setup	User test setup display.
EXIT	Exit MOTOR Strategy	Runtime exit display

There are several different tasks that are performed in parallel to control the position and operation of the pump. The results of these tasks are sent as commands to the variable frequency drives or as indications to the operators. The following subsections of Section 8.2 describe each task performed by the MOTOR strategy, and supply the details of how the separate tasks of position control, pump control, aborting, and test operation are implemented in the MOTOR strategy. These will be followed by discussions of PLC communications, alarms, system parameters, and indication of failure of executing tasks.

8.2.1 Position Control

The position control portion of the GENESIS strategy involves several tasks operating in parallel. The desired pump position is selected by the operator or by the current phase of an automated test. The pump position is controlled by an Eaton AF5000+ variable frequency drive. Parameters are read and set via the GENESIS AF5000 driver. The logic portion of the GENESIS strategy tells the directional motor to start or stop; the direction of movement is based on the actual position feedback from the PLC.

The starting of the directional motor is controlled by automated logic and enabled by the operator. The automated logic will seek to move the pump within a desired range. When enabled, the logic will set up the direction the pump needs to move and then start the directional motor if the pump is not within the desired range.

The directional motor cannot be started if the pump motor is running. This is a safeguard to ensure equipment is not damaged. At no time are both the directional motor and the pump motor to operate at the same time.

The actual position of the pump must be outside of the desired range for the directional motor to start. If all safeguard conditions are met and the pump is directed to move by the operator, a command is sent to the Eaton AF5000+ variable frequency drive to start movement of the pump.

The directional motor will stop if the actual position of the pump is within the desired range; it will also stop if the actual position exceeds the limits of movement or overshoots the desired range. An abort condition will also cause the directional motor to stop. The operator can manually stop the directional motor from the console or from the stop button on the VFD control panel. If any logic conditions are met that specify halting of the directional motor, a command is sent to the Eaton AF5000+ variable frequency drive to stop the movement of the pump.

Communications to the Eaton AF5000+ variable frequency drive are enabled at all times. Parameters can only be read from and sent to the drive when the drive is powered up and receiving asynchronous communication requests. The communication task is responsible for sending control commands and desired operating conditions to the variable frequency drive. Feedback status from the directional motor is displayed to the operator and is also passed to the PLC abort logic via Modbus Plus communications.

The current position of the pump is requested from the PLC. When the operator requests a position change, the actual position is compared with the desired position to determine if the directional motor should be started. After being started, it is stopped automatically when it is within range or when it overshoots the desired range. "Within range" is defined by a dead-band of 2 degrees above and below the desired position but this range is adjustable by the operator. When the pump is within this range the directional motor is not started. If the pump is out of the desired range and a request to change the position is made, the direction to move is automatically determined by the position control logic. The direction signal is then passed to the variable frequency drive and the directional motor is started if all critical safety checks are passed.

If for any reason a condition is preventing a successful start of the directional motor an indicator denoted by a question mark will flash on the display. By clicking on the indicator the source of the problem will be displayed to the operator.

When the directional motor is running, limit-checking on the position of the pump assures that the pump will not be moved to a position beyond the maximum bounds set for the pump position. These limits are adjustable by the operator.

The pump can be moved by operator request before or after pump motor operation. There are safeguards in place to disable pump movement during pump operation to protect equipment and test results. There is automated logic that will disable movement after the pump is within the desired range. Any time the directional motor is stopped the movement of the pump is disabled. The logic is set up so the operator must enable position control every time the pump is to be moved.

The following are the details related to controlling the position (i.e., direction) of the pump.

8.2.1.1 Start

The directional motor can move the pump only if the actual position is outside the desired range, the actual position is within the safe boundary, the pump is not running, and the operator enables the pump to move. The following tags of the MOTOR strategy are used to execute the start of the directional motor.

ABORTOR	BACKWARD	DISMOVE	DMOTOR
FORWARD	FSTRTDEL	GOFORWRD	GOREVERS
HILIMAND	LOLIMAND	NOTRUN	POSCNTRL
POSITION	PRUN	PSTOP	RSTRTDEL
START	STARTAND	STARTSHT	STRTAND

When the output of the *one shot* block <STARTSHT> goes high, the directional motor will be commanded to start by setting <DMOTOR.start> high. <STARTSHT> is set high when the output of the small logic *and* block <STARTAND> goes high. <STARTAND> and <STRTAND> are used as the safeguard to disable a start of movement under improper conditions. All of the following conditions must be true before the directional motor can be started.

- <ABORTOR.QNOT> is high if no abort condition is active.
- <START.DOUT> is high indicating a start request is active from the direction control logic.
- <HILIMAND.QNOT> is high indicating the position of the pump is not beyond the maximum boundary.
- <LOLIMAND.QNOT> is high indicating the position of the pump is not beyond the minimum boundary.
- <DISMOVE.QNOT> is high indicating position control is not disabled.
- <PSTOP.DOUT> is high indicating that pump motor is stopped.
- <PRUN.QNOT> is high indicating the pump motor is not running.

The small logic *or* block <START> is the block that will go high when the directional motor is successfully requested to move. <START> has two inputs to move the pump forward or backward.

- <FSTRTDEL.DOUT> goes high when the pump is to move in the forward direction. <FSTRTDEL> will go high when both of the following inputs are true.
 - <GOFORWARD.DOUT> which goes high when the pump is to go forward.
 - <DMOTOR.forward> is high when the position motor is set to move in the forward direction.
- <RSTRTDEL.DOUT> goes high when the pump is to move in the reverse direction. <RSTRTDEL> will go high when both of the following inputs are true.
 - <GOREVERS.DOUT> which goes high when the pump is to go backward.
 - <DMOTOR.reverse> is high when the position motor is set to move in the reverse direction.

8.2.1.2 Stop

The directional motor will be stopped automatically if the position of the pump is within the desired range, the position of the pump exceeds the safe boundary, or an abort condition arises. The directional motor also can be stopped by an operator request from the STATION8 console.

The following signals are involved with the stop task in the MOTOR strategy.

BUTPULSE	DISP1	DMOTOR	FOROVER
FOVRSTOP	GOSTOP	HILIMIT	LOLIMIT
REVOVER	ROVRSTOP	STOP	STOP2

The output of the small logic *or* block <STOP2> controls <DMOTOR.stop>, which commands the directional motor to halt if the signal transitions from 0 to 1.

<STOP.DOUT> is also a small logic *or* block to help compile stop conditions. <STOP2.DOUT> will go high if any one of the following conditions become true.

- <STOPTEST.DOUT> which goes high when the operator requests to stop the pump, an abort condition occurs, or the pump operation time reaches the desired time.

- <GOSTOP.DOUT> goes high, which is automatically set when the pump position comes within the desired range while the directional motor is moving.
- <FOVRSTOP.DOUT> goes high, which indicates an overshoot condition while the pump is moving forward.
- <ROVRSTOP.DOUT> goes high, which indicates an overshoot condition while the pump is moving in reverse.
- <HILIMIT.DOUT> goes high, which indicates the actual pump position has exceeded the maximum boundary. This will be discussed in further detail in the position feedback task in section 8.2.1.4.
- <LOLIMIT.DOUT> goes high, which indicates the actual pump position has exceeded the minimum boundary. This will be discussed in further detail in the position feedback task in section 8.2.1.4.
- <BUTPULSE.DOUT> goes high, which indicates the operator requested the motor to stop.

Each time the position motor is stopped the movement is disabled by the output of <STOP2> setting input 1 of <DISP1> high to execute the macro DISMOV1. The macro DISMOV1 simply forces <DISMOVE> high. The disable movement task will be discussed in further detail in section 8.2.1.6.

An overshoot condition causes the directional motor to stop operation. This condition is recognized when <POSCNTRL.DO2> goes high and the pump is moving forward or <POSCNTRL.DOUT> goes high and the pump is moving in reverse.

The small logic *and* block <FOROVER> goes high to drive the *one-shot* block <FOVRSTOP> that stops the directional motor when all three of the following conditions are true.

- <DMOTOR.inforwd> is high, indicating the motor is moving in the forward direction.
- <POSCNTRL.DO2> is high when the actual position is beyond the desired range and the motor should be moved in reverse to achieve the desired position.
- <DMOTOR.inrun> is high, indicating the directional motor is running.

The small logic *and* block <REVOVER> goes high to drive the *one-shot* block <ROVRSTOP> that stops the directional motor when all three of the following conditions are true.

- <DMOTOR.inrevers > is high, indicating the motor is moving in the reverse direction.
- <POSCNTRL.DOUT > is high when the actual position is beyond the desired range and the motor should be moved in forward to achieve the desired position.
- <DMOTOR.inrun > is high, indicating the directional motor is running.

8.2.1.3 AF5000+ Communications

The AF5000 device driver is a communication task that runs in parallel with all other GENESIS tasks. The device driver is responsible for sending information to and collecting information from the variable frequency drive. The device block with the tag name <DMOTOR> processes all communications for the variable frequency drive controlling the directional motor.

The following signals are used to perform the AF5000+ communications to the position motor.

AF5COMFA	COMMOR	DACCEL	DATSPEED
DDECEL	DFAULT	DFORWARD	DLINE
DMAXSPD	DMOTOR	DREADY	DREVENAB
DREVERSE	DRUN	DSTOP	VR232070
VR232080	VR232090	VR232100	VR232110
VR232120	GOFORWRD	GOREVERS	STARTSHT
STOP2			

The tags listed in Table 7 are used to issue commands to the variable frequency drive to change operating value and operating condition.

Table 7. Tags That Control Directional Motor Operating Conditions

TAG	VARIABLE	DESCRIPTION
GOFORWRD	forwr.d.o0	Changes direction of motor to forward
GOREVERS	revers.d.o0	Changes direction of motor to reverse
STARTSHT	start.d.o0	Causes motor to start
STOP2	stop.d.o0	Causes motor to halt

The tags listed in Table 8 are used as directional motor feedback signals from the AF5000+ device block. Some of these signals are passed to STATION5 with the PLC communication task and some are used for parameter verification.

Table 8. Tags Used as Directional Motor Feedback Signals

TAG	AF5000+ Variable	Description
COMMOR.INP1	ACCEL.fail	Indicates a failure in communications
DACCEL	ACCEL.inp	Desired acceleration rate
DDECEL	DECEL.inp	Desired deceleration rate
VR232090	FREQ.inp	Stator frequency high range
DLINEV	LINEV.hrng	Line voltage
VR232070	LOAD.inp	Motor load
DMAXSPD	MAXSPD.inp	Maximum speed
VR232100	MOTORA.inp	Motor current
VR232080	MOTORV.inp	Motor voltage
VR232120	SPDSET.inp	Desired speed of motor
VR232110	SPEED.inp	Actual speed of motor
DATSPEED	inatspd.di0	Actual speed is within $\pm 5\%$ of desired speed
DFAULT	infault.di0	Variable frequency drive is faulted
DFORWARD	inforward.di0	Direction of motor movement is forward
DREADY	inready.di0	Drive is ready to run
DREVERSE	inrevers.di0	Direction of motor movement is reverse
DRUN	inrun.di0	Drive is running
DSTOP	instoppd.di0	Drive is stopped
DREVENAB	reven.di0	Reverse enable

Table 9 lists the variables in the AF5000+ device block <DMOTOR>.

Table 9. Variables Used in the AF5000+ Device Block <DMOTOR>

Variable	Description
ACCEL	Desired acceleration rate
DECEL	Desired deceleration rate
FREQ	Stator frequency
LINEV	Line voltage
LOAD	Motor load
MAXSPD	Maximum speed
MOTORA	Motor current
MOTORV	Motor voltage
SPDSET	Desired speed of motor
SPEED	Actual speed of motor
forwrđ	Changes direction of motor to forward
inatspd	Actual speed is within $\pm 5\%$ of desired speed
infault	Variable frequency drive is faulted
inforward	Direction of motor movement is forward
inready	Drive is ready to run
inrevers	Direction of motor movement is reverse
inrun	Drive is running
instoppđ	Drive is stopped
reven	Reverse enable
revers	Changes direction of motor to reverse
start	Causes motor to start
stop	Causes motor to halt

If a failure of communications arises from the VFD, the parameter <DMOTOR.ACCEL.fail> will go high which is passed on <COMMOR> which alarms <AF5COMFA>. Communications failure is an abort condition which causes either motor drive to be automatically stopped as discussed further in the abort task section 8.2.3.

8.2.1.4 Position Feedback

There are maximum limits set on the actual position of the pump. The actual position signal comes from the block <POSITION>. The high-alarm value of <POSITION> is set to 190 degrees, and the low-alarm value is set to 15 degrees.

The following signals are integral to the position feedback task in the MOTOR strategy.

ANION	HILIMAND	HILIMIT	INBAND
LOLIMAND	LOLIMIT	POSITION	

The high-alarm bit from <POSITION> will go high when the actual position exceeds the high alarm value. This high-alarm bit is passed on to a small logic *and* block <HILIMAND>, which is used to disable the start of the directional motor.

The output of <HILIMAND> must be low for the directional motor to be enabled to start.

- <POSITION.HALM> is high indicating the pump has exceeded the maximum boundary.
- <DMOTOR.inforwr> is high indicating the variable frequency drive is enabled to move the directional motor forward.

The high-alarm bit from <POSITION> is currently passed to a small logic *and* block <HILIMIT>, which is used to stop the directional motor and disable movement of the pump if it is running and the actual position exceeds the high limit. The output of <HILIMIT> is passed to the block <STOP>, which will cause the directional motor to stop. The output of <HILIMIT> will go high when the following three conditions are true.

- <POSITION.HALM> is high, indicating the pump has exceeded the maximum boundary.
- <DMOTOR.inrun> is high, indicating the directional motor is running.
- <DMOTOR.inforwr> is high, indicating the directional motor will move forward when it is started.

The low-alarm bit from <POSITION> will go high when the actual position is less than the low-alarm value. This low-alarm bit is passed on to a small logic *and* block <LOLIMAND>, the output of which must be low for the directional motor to be enabled to start. The output of <LOLIMAND> will go high when both of the following conditions are true.

- <POSITION.LALM> is high, indicating the pump has exceeded the minimum boundary.
- <DMOTOR.inrevers> is high, indicating the variable frequency drive is enabled to move the directional motor in reverse.

The low-alarm bit from <POSITION> is also passed to a small logic *and* block <LOLIMIT>, which is used to stop the directional motor and disable movement of the pump, if it is running and the actual position falls below the low limit. The output of <LOLIMIT> is passed to the block <STOP>, which will cause the directional motor to stop. The output of <LOLIMIT> will go high when the following three conditions are true.

- <POSITION.LALM> is high, indicating the pump has fallen below the minimum boundary.
- <DMOTOR.inrun> is high, indicating the directional motor is running.
- <DMOTOR.inrevers> is high, indicating the directional motor will move in reverse when it is started.

The position feedback alarming is disabled on startup of Runtime by controlling the inhibit alarm bit of <POSITION> by <ANTON.QNOT> which will go low 10 seconds after the strategy is started. When the feedback position is within the deadband of the desired position <INBAND> is set high.

8.2.1.5 Direction Control

The actual pump position is monitored from the PLC by STATION8 with the block called <POSITION>. The output of the block <POSITION> is passed to the *DGAP* control block <POSCNTRL> as the measured process variable.

The *DGAP* block <POSCNTRL> is used to control forward movement or reverse movement of the directional motor based on the comparison of desired position with

actual position. The *DGAP* block is an on-off control block that sets two digital outputs to various states depending on where the measured position is in relationship to the deadband. Figure 24 is a diagram of these two outputs.

The high gap and low gap of the *DGAP* block <POSCNTRL> is set to a 2-degree dead band around the desired position.

The first output of the *DGAP* block <POSCNTRL.DOUT> goes high when the measured position is less than the desired position minus the low-gap offset. The output goes to a one-shot block <GOFORWRD>, used to start forward movement of the pump.

The output of <GOFORWRD> will stay high for 15 seconds unless the block is reset. The reset of the block <GOFORWRD> is controlled by the inverse of the small logic *and* block <FORWARD>. When the output of <FORWARD> is low, the output of <GOFORWRD> will always be low and will not start forward movement.

The block <FORWARD> has two inputs that must go high for the output to go high to allow the pump to move forward.

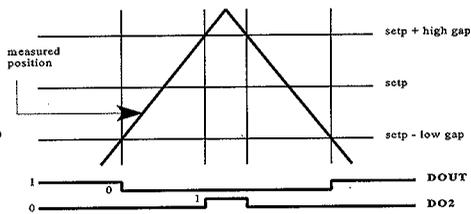


Figure 24. *DGAP* Block Output

- The output of <DISMOVE> must be low. This is the bit used to disable movement mentioned in the disable movement task.
- The output of <NOTRUN> must be high, which is the inverse of the feedback signal from the directional motor variable frequency drive that goes high when the motor is running.

When the output of <GOFORWRD> is allowed to go high it sets <DMOTOR.forwr> high, which commands the variable frequency drive to move the motor forward when it is running. The output of <GOFORWRD> also starts a 6-second *time-on delay* block <FSTRTDEL>.

The 6-second delay of <FSTRTDEL> is to ensure that the forward command is sent to the variable frequency drive before the motor is commanded to start. The output of <FSTRTDEL> sets a small logic *or* block <START> high, which is the beginning of the start task of position control.

The second output of the *DGAP* block <POSCNTRL.DO2> goes high when the measured position is greater than the desired position plus the high-gap offset. The output goes on to a *one-shot* block <GOREVERS> used to start reverse movement of the pump.

The output of <GOREVERS> will stay high for 15 seconds unless the block is reset. The reset of the block <GOREVERS> is controlled by the inverse of the small logic *and* block <BACKWARD>. When the output of <BACKWARD> is low the output of <GOREVERS> will always be low and will not start reverse movement.

The block <BACKWARD> has two inputs that must go high for the output to go high and allow the pump to move backward:

- The output of <DISMOVE> must be low. This is the bit used to disable movement mentioned in the disable move task.
- The output of <NOTRUN> must be high. This is the inverse of the feedback signal from the directional motor variable frequency drive that goes high when the motor is running.

When the output of <GOREVERS> is allowed to go high it sets <DMOTOR.revers> high, which tells the variable frequency drive to move the motor in reverse when it is running. The output of <GOREVERS> also starts a 6-second *time-on delay* block <RSTRTDEL>.

This delay ensures that the reverse command is sent to the variable frequency drive before the motor is commanded to start. The output of <RSTRTDEL> sets a small logic *or* block <START> high, which, as noted above, imitates the position control task.

An overshoot condition causes the directional motor to stop operation. This condition is recognized when <POSCNTRL.D02> goes high and the pump is moving forward or <POSCNTRL.DOUT> goes high and the pump is moving in reverse.

The small logic *and* block <FOROVER> goes high to drive the *one-shot* block <FOVRSTOP> that stops the directional motor when all three of the following conditions are true:

- <DMOTOR.inforwr> is high, indicating the motor is moving in the forward direction.
- <POSCNTRL.D02> is high when the actual position is beyond the desired range and the motor should be moved in reverse to achieve the desired position.
- <DMOTOR.inrun> is high, indicating the directional motor is running.

The small logic *and* <REVOVER> goes high to drive the *one-shot* block <ROVRSTOP> that stops the directional motor when all three of the following conditions are true:

- <DMOTOR.inrevers> is high, indicating the motor is moving in the reverse direction.
- <POSCNTRL.DOUT> is high when the actual position is beyond the desired range and the motor should be moved forward to achieve the desired position.
- <DMOTOR.inrun> is high, indicating the directional motor is running.

8.2.1.6 Disable Movement

The pump is disabled from moving unless the operator enables movement from the console. Each time the pump stops or goes beyond the desired range the pump is disabled from moving automatically.

The following tags in the MOTOR strategy are used to execute the disable movement task.

DISMOV	DISMOV1	DISP1	PBPOSAN2
DISMOVE	DISP1	INBAND	ON
STOP2			

The *DIV* block <DISMOVE> must have a 0 value in order to start the directional motor; This is the value set when the operator enables pump movement from the STATION8 console. The value of <DISMOVE> is set to 1 any time the directional motor is stopped or when the MOTOR strategy is first started. This is accomplished with two custom macros in EXTRA.KMS. The macro DISMOV0 is used to set <DISMOVE.DOUT> low and the macro DISMOV1 is used to set <DISMOVE.DOUT> high which will disable movement of the pump.

The following signal will execute the macro DISMOV0.

- <PBPOSAN2.DOUT> which goes high if the operator selects to position the pump during a full integrated test.

The movement of the pump is also enabled manually by the operator from the DDISPLAY display.

The following signals will execute the macro DISMOV1.

- <STOP2.DOUT> will go high any time the directional motor is stopped.
- <ON.DOUT> when the MOTOR strategy is first started.

- <DISMOV1.DOUT> which goes high any time there is a problem with starting the position motor, the position motor is not running, and the position of the pump is not within the desired deadband.

The *function* block <DISMOV> is used to stop the directional motor and disable movement when the position of the pump is within the desired range. This block also checks the disable movement bit from <DISMOVE.DOUT>, the forward movement bit from <POSCNTRL.DOUT>, and the reverse movement bit from <POSCNTRL.D02>. When all three conditions are not true an internal control bit <DRI> starts an internal counter <ARI>. This counter is used to delay the stop of the pump by 2 seconds after the position is within the dead band. This allows the pump to move closer to the desired position before shutting down movement. When the internal counter reaches 2 seconds the directional motor is commanded to stop and movement is disabled by control bit <DISMOV.DOUT>.

The *function* block <DISMOV> also has a logic equation to determine if the pump is within the desired range. <DISMOV.D02> is passed to <INBAND>, which is used to indicate the pump is within the dead band range of 2 degrees.

8.2.2 Pump Control

Several tasks that execute in parallel comprise the pump control. The pump is driven by a motor-controlled Eaton AF5000+ variable frequency drive. Commands to start, stop, and change speeds are processed by a GENESIS AF5000 communication driver.

Abort logic is provided to ensure safe operation of the pump. Pump operating status is displayed to the operator and passed on to the PLC for abort logic and for STATION5 to data log critical signals.

An operator request will start the pump provided preventative conditions do not disallow it. An operator can start the pump from the STATION8 console if the directional motor is not operating, all permissives are satisfied, and the enable pump control bit has been set by the operator. If the start logic is satisfied, a command is sent to the Eaton AF5000+ variable frequency drive that controls the pump motor.

If for any reason a condition is preventing a successful start of the pump motor an indicator denoted by a question mark will flash on the display. By clicking on the indicator the source of the problem will be displayed to the operator.

The pump can be halted automatically from the elapsed test timer, an abort condition, or a manual request by an operator. The operator can halt the operation manually by selecting the stop pump button on the STATION8 console or the stop button on the VFD control panel. The stop button on the STATION8 console requires two consecutive presses to prevent accidental stopping of the pump. If the logical conditions to halt the pump motor are met, a command is sent to the Eaton AF5000+ variable frequency drive to stop the pump.

Communications from GENESIS to the Eaton AF5000+ variable frequency drive which control the pump motor are enabled at all times, parameters can only be read and sent.

to the drive when the drive is powered up and receiving asynchronous communication requests. The MOTOR strategy is responsible for sending control commands and desired operating conditions to the pump motor and retrieving feedback status from the pump motor.

The current operating values in the variable frequency drive are compared with the desired operating values set by the operator. If the current operating parameters deviate from the desired operating parameters the operator is warned and the pump is not started. For instance, the desired pump motor speed is compared with the actual pump speed and a warning is displayed to the operator if difference is more than 5 RPM of the desired speed. The acceleration, deceleration, and maximum speed from the pump VFD are also compared with the desired operating conditions.

The predicted motor current draw is calculated based on the desired speed of the motor and an alarm limit is set on the motor current 10 amps above this limit. The operator will be warned during pump operation if the motor current exceeds this limit. An abort limit is set 20 amps above the predicted current draw. The PLC logic will send an abort command if the motor current exceeds this limit. There is a delay of 6 seconds after the pump is started or changes speeds before this alarm and abort logic come into effect for the motor amps.

The following are the details related to controlling the operation of the mixing pump to actually mix the contents of the tank.

8.2.2.1 Start

The AF5000+ device block variable <PMOTOR.start> requires a transition from 0 to 1 to start operation of the pump motor. The <PMOTOR.start> variable is driven by the output of the block with the tag name <PBSTRTDN>. <PBSTRTDN> will go high when the operator successfully starts a fully integrated test for pump operation. Test operation will be explained in detailed sections to follow.

The following blocks are involved with the start pump control task.

PBSTRTDN

PMOTOR

8.2.2.2 Stop

The AF5000+ device block variable <PMOTOR.stop> requires a transition from 0 to 1 to stop operation of the pump motor. The <PMOTOR.stop> variable is driven by the output of a small logic *or* block with the tag name <STOPTEST>.

The <STOPTEST> block will go high when the pump operation is stopped automatically from an abort condition, from the elapsed time exceeding the desired times, the operator manually stopping pump operation, or failure in communications to either VFD.

The following blocks are involved with the stop pump control task.

PMOTOR

STOPTEST

8.2.2.3 AF5000+ Communications

The AF5000+ device driver is a communication task that runs in parallel with all other GENESIS tasks. The device driver is responsible for sending information to and collecting information from the variable frequency drive. The device block with the tag name <PMOTOR> processes all communications for the variable frequency drive controlling the pump motor.

The following signals are used to perform the communications task for the pump control.

AF5COMFA	COMMOR	EQUSPD	PACCEL
PATSPED	PDECEL	PFAULT	PFORWARD
PLINEV	PMAXSPD	PMOTOR	PREADY
PREVENAB	PREVERSE	PRUN	PSTOP
VR232020	VR232030	VR232040	VR232050
VR232060	VSDRS232		

The tags listed in Table 10 are used to issue commands to the variable frequency drive to change operating values and operating conditions.

Table 10. Tags That Control Pump Motor Operating Conditions

TAG	AF5000+ Variable	Description
COMMOR.QNOT	forwrd.do0	Changes direction of motor to forward
PBSTRTDN	start.do0	Causes motor to start
STOPTEST	stop.do0	Causes motor to halt

The tags listed in Table 11 are used as pump feedback signals from the AF5000+ device block. Some of these signals are passed to STATION5 with the PLC communication task, and some are used for parameter verification.

Table 11. Tags Used as Pump Motor Feedback Signals

TAG	AF5000+ Variable	Description
COMMOR.INP2	ACCEL.fail	Indicates a failure in communications
PACCEL	ACCEL.inp	Desired acceleration rate
PDECEL	DECEL.inp	Desired deceleration rate
VR232030	FREQ.hrng	Stator frequency high range
PLINEV	LINEV.inp	Line voltage
VSDRS232	LOAD.inp	Motor load
PMAXSPD	MAXSPD.inp	Maximum speed
VR232040	MOTORA.inp	Motor current
VR232020	MOTORV.inp	Motor voltage
VR232060	SPDSET.inp	Desired speed of motor
VR232050	SPEED.inp	Actual speed of motor
VSDRS232	LOAD.inp	VSD % load
PATSPEED	inatspd.di0	Actual speed is within $\pm 5\%$ of desired speed
PFAULT	infault.di0	Variable frequency drive is faulted
PFORWARD	inforward.di0	Direction of motor movement is forward
PREADY	inready.di0	Drive is ready to run
PREVERSE	inrevers.di0	Direction of motor movement is reverse
PRUN	inrun.di0	Drive is running
PSTOP	instoppd.di0	Drive is stopped
PREVENAB	reven.di0	Reverse enable

Table 12 lists the variables in the AF5000+ device block <PMOTOR> .

Table 12. Variables Used in the AF5000+ Device Block <PMOTOR>

Variable	Description
ACCEL	Desired acceleration rate
DECEL	Desired deceleration rate
FREQ	Stator frequency
LINEV	Line voltage
LOAD	Motor load
MAXSPD	Maximum speed
MOTORA	Motor current
MOTORV	Motor voltage
SPDSET	Desired speed of motor
SPEED	Actual speed of motor
forwr	Changes direction of motor to forward
inatspd	Actual speed is within $\pm 5\%$ of desired speed
infault	Variable frequency drive is faulted
inforward	Direction of motor movement is forward
inready	Drive is ready to run
inrevers	Direction of motor movement is reverse
inrun	Drive is running
instoppd	Drive is stopped
reven	Reverse enable
revers	Changes direction of motor to reverse
start	Causes motor to start
stop	Causes motor to halt

If a failure of communications arises from the VFD the parameter <PMOTOR.ACCEL.fail> will go high which is passed on <COMMOR> which alarms <AF5COMFA>. Communications failure is an abort condition which should cause either motor drive to be automatically stopped as discussed further in the abort task section 8.2.3.

8.2.2.4 Parameter Verification

The following blocks are used for parameter verification in the MOTOR strategy.

CALCUR	RCALDISP	RCURABR	TCURALRM
CURPRED	DISPRABT	DISPRALM	DISPRPRE
EQUQSPD	VR232050	VR232060	

The *function* block <EQUQSPD> contains an equation to compare the desired speed set in the variable frequency drive <VR232060> with the actual speed <VR232050>. The variable <EQUQSPD.DR1> will be high if the actual speed is within 5% of the desired speed. This will be displayed to the operator as a light green indication on the pump speed bar graphs.

The desired speed value is passed to the function block <CALCURR> uses <CURPRSPD> as the desired speed and automatically calculates the predicted motor current, which is passed to <CURPRED>.

The predicted motor current plus 20% is passed to <CURALRM> as the alarm limit. The output of <CURALRM> sets the high alarm value of <VR232040> and is passed to the PLC for the high motor alarm.

The predicted motor current plus 40% is passed to <CURABRT> as the abort limit. The output of <CURABRT> is passed to the PLC as the abort limit for motor current.

The desired pump speed is also passed to the *function* block <CALDISPR> which automatically calculates the predicted discharge pressure, which is passed to <DISPRPRE> for operator display.

The predicted discharge pressure plus 20% is passed to <DISPRALM> as the alarm limit, and the output of <DISPRALM> is passed to the PLC to set the high alarm value of the discharge pressure signal.

The predicted discharge pressure plus 40% is passed to <DISPRABT> as the abort limit, and the output of <DISPRABT> is passed to the PLC as the abort limit for the discharge pressure.

8.2.3 Abort

The directional motor or pump motor will automatically halt operation if an abort condition becomes active from the abort coil in the PLC, a failure in AF5000+ communications, or a failure in PLC communications to STATION8 or STATION5.

There is limit-checking built into the logic used to safeguard the operation of the pump and directional motors. Feedback signals from the PLC are compared with abort limits and desired operating conditions set by the operator (directly or by test selection). Several instrument signals are also monitored by the PLC to ensure safe operating conditions. The PLC can command the mixing pump control and directional control motors to stop if any of these signals exceed their operating range. These instrument and motor signals cause the abort coil in the PLC to become active. The mixer pump signals capable of aborting the pump run are listed in Table 13.

Table 13. Mixer Pump Signals Causing Pump Aborts

Instrument Tag	Problem Indication
VR232050	Mixer pump speed high.
VR232040	Mixer pump motor amps high.
MIP00001	Moisture in pump motor oil.
TIR12A01	Mixer pump motor oil temperature A high.
TIR12A02	Mixer pump motor oil temperature B high.
ZIMPE143	Position CW limit switch.
ZIMPE144	Position CCW limit switch.
WIR12A01	Pump support column strain #1 high.
WIR12A02	Pump support column strain #2 high.
WIR12A03	Pump support column strain #3 high.
WIR12A04	Pump support column strain #4 high.

The status of the PLC communications with STATION5 and STATION8 is monitored at all times, and any time if the PLC communications fails both motors will be halted. A failure in communications with either of the AF5000+ variable frequency drives will also cause an abort condition which will stop both motors. If communications to the VFDs is halted it will probably be unlikely that the stop command will be successfully sent to the VFDs. This condition assumes communications has faulted and revived itself. In any

case an alarm warning will be indicated to the operator that an abort condition has occurred and which one of the four sources has caused the motors to stop.

The following blocks are involved with the abort task.

A5ABRT	ABORT	ABORTOR	ABORTPUL
COMMOR	NODEFAIL	PLCFAIL	

The current abort status is indicated by the small logic *or* block <ABORTOR> which has the following input conditions.

- <ABORT.DOUT> is high if the abort coil in the PLC is active.
- <COMMOR.DOUT> is high if either VFD has a failure in communications to the MOTOR strategy.
- <NODEFAIL.DOUT> is high if a failure in communications from STATIONS5 to the PLC is recognized.
- <PLCFAIL.DOUT> is high if a failure of communications to the PLC is recognized from the MOTOR strategy.

The output of <ABORTOR> is passed to a small logic *pulse* block <ABORTPUL> which is passed on to <STOPTEST> which will stop either VFD with a communication command.

<COMMOR> is a small logic *or* block with two inputs from each hardware device block. If communications to either variable frequency drive goes bad, indicated by a fail bit of the first parameter selected from each hardware device block, <COMMOR.DOUT> will go high. This signal is passed to the PLC with the *digital output* block <A5ABRT>.

8.2.4 Test Operation

There are different tests that can be executed by the operators. Each test specifies the position of the pump, the speed of the pump, the duration of pump operation, and the time delay between pump operations, all of which have an effect on the critical feedback signals. The MOTOR strategy is responsible for controlling the pump under stringent guidelines set up by the design engineers.

Each test has a specific number associated with the parameter settings for the test. The test engineer cannot override the test parameters that control the pump unless the mixing pump motor display is used, which is a manual control screen for pump operation. The operator can select the test to be executed from the console and must set the values of the test to the active parameters, by selecting the SET VALUES button from the console. The test operation must then be enabled before starting the test.

When a test is active an elapsed timer keeps track of the overall duration of the test. Another timer keeps track of how long the pump is in operation. An early warning is activated before the elapsed time of pump operation reaches the desired operation time. If either of these elapsed timers reach the desired time during pump operation the test will be aborted and the pump will be stopped. The PLC times pump operation and the overall test time. The desired test time values are passed to the PLC from the test selection with network communications to STATION5.

The pump may become active several times during a test at different positions. The operator will be prompted to move the pump when the desired time of pump operation has been achieved at the current position. After pump operation has been completed at all desired positions, the end of the test is delayed until all critical data are monitored for an appropriate period of time after pump operation is suspended.

During test selection and operation there are several procedures the test engineer must follow to begin the test. The system sets control points on the STATION8 console to different colors to show the operators which procedures have been performed and what functions have been enabled.

Details of the actual test operation, including their selection, how the elapsed time is calculated, and the information provided to the operator, is included in the following sections.

8.2.4.1 Test Creation and Selection

The following blocks are related to test creation and selection.

ANGLECHG	CHANGED	CHGCHK1	CLRMESS
DECELCHG	DELPUL	HRSCHG	INFILE
MANUAL	MAXINDX	MINSCHG	NEXTINDX
NEXTPUL	OACCEL	OANGLE	ODECEL
OHRS	OLDTEST	OMINS	ORESTIM
OSECS	OSPEED	PBTESTNO	PREVPUL
RESCHG	SAVEPUL	SECSCHG	SPEEDCHG
TESTCHG	TESTSET	TESTSET2	UACCEL
UANGLE	UDECCEL	UHRS	UMINS
UMSG1	URESTIM	USECS	USPEED
UTESTNO			

All test setups are in a file called USERSET.TDF (TDF stands for Test Data File). The test setups can be accessed from the pump operation display PUMPRUN. There is a display called TESTSET which allows the creation and modification of the test setup file USERSET.TDF. This display is not part of the STATION8 menu system but can be accessed via the F1 function key. The TESTSET display calls Report and Recipe code which performs basic operations on the test setup file such as test setup loading, scrolling, insertion and deletion.

The details of all of this are described below.

USERSET.TDF file format. This is an ASCII file which contains records with the following information for each test setup:

test number (or end of file marker)
 angle (degrees)
 pump speed (RPM)
 test duration (hours minutes seconds)
 pump motor acceleration (RPM/Sec)
 pump motor deceleration (RPM/Sec)
 test description
 Reset overall test timer indication (1 = yes, 0 = no)

The format for each record is:

```
tt.t aaa.a rrrr.r hh mm ss ccc.c ddd.d eeeeeeeeeeeeeeeeeeeee f
```

where tt.t is the test number, aaa.a is the angle, rrrr.r is the pump speed, hh is test duration hours, mm is test duration minutes, ss is test duration seconds, ccc.c is pump motor acceleration, ddd.d is pump motor deceleration, e...e is the test description and f is reset overall timer indication.

Each record is separated by a carriage return/line feed. The end of file record has 999 as the test number.

The reset overall test timer field is used to indicate whether the overall test time should be reset at the beginning of a test. This allows for multiple-angle tests; the timer should be reset for the first of the series of tests and not for the subsequent tests.

TESTSET Screen. The TESTSET screen is used to create and maintain the setup file USERSET.TDF. It contains data entry fields for the above parameters as well as buttons to save and delete test setups, and buttons to go to the previous or next record in the file. A test setup can also be loaded by typing the test number in the test number data entry field. There is an indication on the screen telling whether or not the current test number exists in the file.

The buttons on the screen initiate the execution of reports which do the actual file manipulation. The Save Test button executes the report SAVE.RPS, the Delete Test button, DELETE.RPS, the Next button, NEXT.RPS, and the Previous button, PREV.RPS. In addition, there are reports CLRMESS.RPS, ULOAD.RPS and ILOAD.RPS. These reports are discussed in the next section.

Reports for test setup file manipulation. SAVE.RPS is executed by clicking on the Save Test button on the TESTSET screen. It inserts the test setup which has been entered into the data entry fields on that screen into the USERSET.TDF file. The new setup is inserted so as to keep the records in numeric order by test number. If a record already exists with the same test number, it is replaced.

DELETE.RPS is executed by clicking on the Delete Test button on the TESTSET screen. It deletes the current test from the file, if it exists in the file.

NEXT.RPS loads the setup for the next test (in numeric order). This setup is displayed on the screen. It is executed by clicking on the Next button on the TESTSET screen or the up-arrow button on the PUMPRUN screen. If the current test is not in the test setup file, the next button will load the next test from the last test loaded which was in the file.

PREV.RPS works the same as NEXT.RPS only the previous test setup (in numeric order) is loaded. It is executed by clicking on the Previous button, or the down-arrows.

ILOAD.RPS loads the first setup in the test setup file and initializes the housekeeping blocks in the strategy. It is executed upon strategy startup.

ULOAD.RPS loads the setup for the current test number. It is executed when the strategy blocks detect that the user has entered a new test number.

CLRMESS.RPS clears the message field on the TESTSET screen. The reports SAVE and DELETE produce messages on the TESTSET screen giving the status of the save or delete operation. These messages are cleared by CLRMESS after a delay. It is executed by the strategy block CLRMESS going high. This bit is controlled from within the SAVE and DELETE reports.

SETOLD.RPS sets a set of strategy blocks corresponding to all of the test setup parameters to be equal to the current test setup parameters. This allows the strategy to detect when the user changes a parameter. It is executed from within any of the above reports which load a new predefined test setup.

Strategy Blocks. The reports are run from the report blocks <TESTSET> and <TESTSET2>. <TESTSET> executes SAVE.RPS from input 1, DELETE.RPS from input 2 and CLRMESS.RPS from input 3. <TESTSET2> executes ULOAD.RPS from input 1, NEXT.RPS from input 2, PREV.RPS from input 3, and ILOAD.RPS from input 4.

The test setups are loaded into a set of AIN and DIN blocks called <UTESTNO>, <UANGLE>, <USPEED>, <UHRS>, <UMINS>, <USECS>, <UACCEL>, <UDECEL>, <UDESC> and <URESTIM>. There is a corresponding

set of blocks with the first letter of "O" instead of "U" (except for the one corresponding to <UTESTNO>, which is called <OLDTEST> and the one corresponding to <UDESC> which is called <TDESC>). These sets of blocks are set to be equal when a new test setup is loaded (see SETOLD.RPS above). A set of alarm blocks detects when any of the pairs of blocks contain differing values. This indicates that the user has changed a setup parameter and this information is displayed on the screens. In addition, if the test number is changed, the report ULOAD.RPS is executed to load in the setup corresponding to the new test number if it exists. The block <INFILE> is maintained by the reports. It is 1 if the current test is in the file, 0 otherwise. The blocks <NEXTINDX> and <MAXINDX> are maintained by the reports. They keep track of the record number of the current test and the record number of the highest numbered test in the file respectively. The block <UMSG1> is used by the reports to display messages in the message field of the TESTSET screen. The block <CHANGED> is 1 if any of the test setup parameters have been changed since last loaded and 0 if not. The block manual is set when a test setup is actually loaded for use if either <CHANGED> is 1 or <INFILE> is 0. This indicates that a setup is being used that was not predefined.

8.2.4.2 Elapsed Time Calculation

There are two elapsed timers that are used during test operation. The overall timer for test operation keeps track of the total test time. There is a second timer to time pump operation time.

The following blocks are integral in performing the elapsed time calculation task.

DF65535	DFLT0	DISP4	PB30SEC
PBCALSEC	PBDNTM	PBHR	PBMIN
PBRAMP	PBRAMPHL	PBSEC	PBSPLIT
RESETPB	RSECPB	SPLITPB	TSTRUNNG

The *RAMP* block <RSECPB> totals the test time while a test is in operation. When ramp rate 1 becomes active the output of <RSECPB> increments by one every second.

A test is in operation when the *rs flip-flop* block <TSTRUNNG> is high. This signal controls <RSECPB.RPIO>, which activates ramp rate 1. The elapsed timer <RSECPB> is reset at the beginning of test operation by <RESTPB> which is set high with the RR code PBSETVAL.RPS when a new test is selected. The elapsed test time is broken down from seconds into hours, minutes, and seconds by the *function* block <SPLITPB>.

There is a second elapsed timer used to keep track of the pump operation time at each position. The *RAMP* block <PBRAMP> totals the time the pump is in operation.

When ramp rate 1 becomes active, the output of <PBRAMP> increments by one every second.

The pump is in operation when the *rs flip-flop* block <TSTRUNNG> is high. This signal controls <RSECPB.RPIO>, which activates ramp rate 1.

The elapsed timer <PBRAMP> is reset at the beginning of pump operation by <SETVALDN> which goes high when the test values have been successfully set. The high limit of <PBRAMP> is controlled by <PBRAMPHL> which will pass either the value 0 from <DFLTO> or the maximum pump operation time from <DF65535>. <PBRAMPHL> will pass the value of <DF65535> when <SETVALDN> is high when the test values are successfully loaded.

The desired pump operation time is set during test selection by setting the test values. The *AIN* blocks <PBHR>, <PBMIN>, and <PBSEC> contain the desired minutes and seconds for pump operation. The outputs of these blocks are passed to a *function* block <PBCALSEC>, which merges the values into one desired pump operation time value in seconds. This value is passed to the *function* block <PBSPLIT>, which is used to stop pump operation when the elapsed time from <PBRAMP> exceeds the desired time from <PBCALSEC>.

The control bit <PBSPLIT.DOUT> will go high when the elapsed time exceeds the desired time and is passed to the *DIN* <PBDNTM>. The output of this block resets the *rs flip-flop* block <PBRSFF> mentioned above. The output of <PBDNTM> is passed to logic which stops pump operation by setting <STOPTEST> high. <PBSPLIT> also breaks seconds of pump operation into hours, minutes, and seconds for operator display.

The desired pump operation times are reset by the output of <STOPTEST> executing the macro RESPTIM which is referenced in <DISP4>.

The *alarm* block <PB30SEC> is used to warn the operator 30 seconds before the end of pump operation. The high alarm value of <PB30SEC> is set by the second analog output of <PBCALSEC> which is set to 30 less than the desired pump operation time during test operation and to a very large number when a test is not in operation.

8.2.4.3 Operator Test Information

The PUMPRUN screen can be used for any test. The operator actions required for the PUMPRUN screen are discussed below. The operations are:

- 1) Select a test setup or manually enter a pump test configuration
- 2) Set the selected test values (set values).
- 3) Position the pump (if required)
- 4) Enable the pump test.
- 5) Start the pump test.
- 6) Stop the pump test (performed automatically when the preset time elapses or at any time by the operator).

Test Setup. Tests can be selected by typing the test number in the Test entry field on the screen. This will access the USERSET.TDF file to load the parameters for this test into the other fields (see 8.2.4.1). These values can be used as is, or can be changed by entering different values into any of the fields. If new values are entered the test is considered to be a manual test. A message will appear in the lower left corner of the screen indicating that the test values have been changed.

Set Values. The test parameters do not take effect until the SET VALUES button is pressed. The button is only effective if it is green in color. Pressing SET VALUES begins the pump test process, copying the test setup parameters to operational parameters. After the SET VALUES button is pressed the STOP TEST button will turn green. This button will abort the current test and return the system to the test setup mode. This can be done at any time, however the button functions differently if the pump is actually running (see below). After the SET VALUES button is pressed the POSITION PUMP button will turn green if the pump is not oriented at the specified angle. Otherwise the ENABLE TEST button will turn green. If there is a problem with setting the values, a flashing yellow question mark will appear at the bottom of the screen. Clicking on this question mark will bring up a screen showing what the problem was. Similar problem screens exist for all of the operations.

Position Pump. Pressing this button will cause the directional motor to begin to orient the pump to the desired angle. The current angle and the directional motor speed (in RPMs), as well as a directional motor running indicator are displayed above the button. When the pump is in position the POSITION PUMP button will turn gray and the ENABLE TEST button will turn green. As with all of the buttons, if there is a problem with pump positioning, clicking on the flashing question mark will bring up a problem screen.

Enable Test. This button enables a pump test. At this time test times and pump speed based alarm and abort limit values are sent to the PLC. The START TEST button will turn green. The operator has 60 seconds to start the test after the enable button is pressed. If the 60 seconds elapses, the ENABLE TEST button will once again turn green.

Start Test. This button starts the pump. The elapsed time indication on the screen will begin to increment and the pump motor running indicator will turn green. The pump speed is also displayed. The legend for the STOP TEST button will change to read ENABLE STOP.

Stop Test. This button functions to reset the system when the pump is not running. It returns the system to the test setup mode (SET VALUES button is green). When the pump is running, the legend for this button is ENABLE STOP. When clicked upon, the legend of the button will change to STOP TEST. After two seconds, the button will begin to flash. At this point, the user has five seconds to click the button again. This will stop the pump and reset the system to the test setup mode. If the 5 seconds elapses, the legend will change once again

to ENABLE STOP. The pump will stop automatically when the specified time for the test has elapsed. There is also a timer in the PLC to shut off the pump as a backup to this. The STOP TEST button gives the operator a means to manually stop the pump before the allotted time has elapsed. When the pump is stopped either manually or at the end of the time allotted, the system will return to the test setup mode.

8.2.4.4 Test Operation Strategy Details

There is logic within the MOTOR strategy for indication of test operation status. There are status bits for test values set correctly, pump at desired position, test enabled to start, and test running. If a function is selected by an operator and the task was not successfully executed there is logic to indicate what the problem is. The following sections describe each main task execution and status feedback for indication of success of operation.

Set Test Values. The desired test values selected by the operator must be downloaded to the control parameters in the MOTOR strategy. This is done by the operator with the set values button. The blocks in the following table are related to the set test values task.

NORUN	PBENABLE	PBREPORT	PBSETAND
SETTEST	SETVALDN	STOPTEST	TSTRUNNG
VALSET			

The small logic *pulse* block <SETTEST> starts the execution of the set values task. Its output is passed to the small logic *and* block <PBSETAND> which has the following input conditions to be true before executing the RR code PBSETVAL.RPS by setting input 1 of <PBREPORT> high.

- <SETTEST.DOUT> is set high by the operator to set the test values.
- <PBENABLE.QNOT> is high if a test is not already enabled.
- <TSTRUNNG.QNOT> is high if a test is not in progress.
- <NORUN.DOUT> is high if both the pump motor and position motor are stopped.
- <VALSET.QNOT> is high if the test values are not already set.

At the end of the RR code PBSETVAL.RPS <SETVALDN> is set high to indicate the test values are successfully set. <SETVALDN> has a pulsed output and sets the *rs flip-flop* block <VALSET> high. <VALSET.DOUT> is the status bit used to indicate that the test values have been successfully set. <VALSET> is reset when <STOPTEST> goes high when a test is stopped.

Problem with Setting Test Values. If the test values are requested to be set and the task is not successfully completed then the blocks in the following table are used to indicate what the problem with setting test value variables is.

DISP3	NORUN	NOSEPROB	PBBUTSTP
PBPROB	PDISP	RSETPROB	SEDISP
SETPROB	SETPROBA	SETPROBR	

The *rs flip-flop* block <SETPROB> is used to indicate a problem with setting the test values. <SETPROB> is set high if the output of <SETPROBA> is high and the reset input <SETPROBR> is low.

<SETPROBA> is a small logic *and* block with the following inputs.

- <SETTEST.DOUT> is high if the test values are requested to be set.
- <NORUN.DOUT> is high if both motors are stopped.
- <VALSET.QNOT> is high if the test values are not set.

<SETPROBR> will reset <SETPROB> if any of the following conditions are true.

- <RSETPROB.DOUT> which is set high by the operator to reset the problem indication.
- <PBBUTSTP.DOUT> which goes high when the operator manually stops a test.
- <NOSEPROB.DOUT> which is a small logic *and* block with the following input conditions.
 - <PBENABLE.QNOT> is high when a test is not enabled.
 - <TSTRUNNG.QNOT> is high when a test is not running.

- <NORUN.DOUT> is high when both motors are stopped.

When <SETPROB> goes high <PBPROB> is set high which indicates a test operation problem. The small logic *and* block <SEDISP> is used to bring up the problem indication display for setting test values. When <SETPROB> is high and the operator sets <PDISP> high to view the problem the display SETPROB is brought up via the *display* block <DISP3>.

Position Pump Selection. When the test values are set the desired pump position is compared with the actual position. If the pump is not at the desired position the pump will need to be positioned before a test can be enabled. The blocks in the following table are the key variables in the pump position selection.

ABORTOR	INPOSI	NORUN	PBENABLE
PBPOSAN2	PBPOSAND	PBPOSPMP	POSSET
TSTRUNNG	VALSET		

<PBPOSPMP> starts the execution of the pump position selection. Its output is passed to the small logic *and* block <PBPOSAND> which together with <PBPOSAN2> has the following input conditions before <PBPOSAN2> will start the pump position control by executing the macro DISMOV0 as discussed earlier in the disable movement task section.

- <TSTRUNNG.QNOT> which is high if a test is not running.
- <ABORTOR.QNOT> which is high if an abort conditions is not active.
- <PBENABLE.QNOT> is high if a test is not enabled.
- <VALSET.DOUT> is high if the test values are set.
- <NORUN.DOUT> is high if both motors are stopped.
- <PBPOSPMP.DOUT> goes high when the operator selects to position the pump.
- <POSSET.QNOT> which is an *AND* block with the following input conditions.
 - <VALSET.DOUT> is high when the test values are set.
 - <INPOSI.DOUT> which is high when the pump is at the desired position.

Problem with Position Pump Selection. If the pump position selection does not successfully execute, the blocks in the following table are used to indicate what the problems with the position pump variables are.

ABORTOR	DISP3	NOPOPROB	NORUN
PBBUTSTP	PBENABLE	PBOSMPMP	PBPROB
POSDISP	POSPROB	POSPROBA	POSPROBR
POSSET	RPOSROB	TSTRUNNG	VALSET

The *rs flip-flop* block <POSROB> is used to indicate a problem with selecting to position the pump. <POSROB> is set high if the output of <POSROBA> is high and the reset input <POSROBR> is low. <POSROBA> is a small logic *and* block with the following inputs.

- <PBOSMPMP.DOUT> is set high by the operator to position the pump.
- <POSSET.QNOT> which is high when the pump is not in position or the test values are not set.
- <NORUN.DOUT> is high when both motors are stopped.
- <VALSET.DOUT> is high if the test values have been set.

<POSROBR> will reset <POSROB> if any of the following conditions are true.

- <RPOSROB.DOUT> which is set high by the operator to reset the problem indication.
- <PBBUTSTP.DOUT> which goes high when the operator manually stops a test.
- <NOPOPROB.DOUT> which is a small logic *and* block with the following input conditions.
 - <ABORTOR.QNOT> which is high if no abort condition is active.
 - <TSTRUNNG.QNOT> is high if a test is not running.
 - <PBENABLE.QNOT> is high if a test is not enabled.
 - <VALSET.DOUT> is high if the test values are set.

- <NORUN.DOUT> is high if both motors are stopped.

When <POSPROB> goes high <PBPROB> is set high which indicates a test operation problem. The small logic *and* block <POSDISP> is used to bring up the problem indication display for position pump selection. When <POSPROB> is high and the operator sets <PDISP> high to view the problem the display <POSPROB> is brought up via the *display* block <DISP3>.

Position Pump Operation Problem. When the position motor is to be started to position the pump and a problem prevents the actual movement of the pump, the blocks in the following table are used to indicate what the problem with position pump operation variables is.

ABORTOR	DDISP	DISMOV1	DISMOVE
DISP1	DISP3	DMDISP	DMPROB
DMPROBD	IDMPROBR	DRUN	DSTOP
ENMOVPUL	ENMOVTON	HILIMAND	HOLDENMO
INBAND	LOLIMAND	NODMPROB	PBDMPROB
PBPROB	PBPROB2	PDISP	POSDISP
PRUN	PSTOP	RDMPROB	

The *rs flip-flop* block <DMPROB> is used to indicate a problem with positioning the pump. <DMPROB> is set high if the small logic *pulse* block <ENMOVPUL> is high and the reset input <DMPROBR> is low. <ENMOVPUL> is set high by <ENMOVTON> which delays <HOLDENMO> for 4 seconds. <HOLDENMO> stays high for 5 seconds after the pump is requested to move by <DISMOVE> going low. <DMPROBR> will reset <DMPROB> if any of the following conditions are true.

- <RDMPROB.DOUT> which is set high by the operator to reset the problem indication.
- <NODMPROB.DOUT> which is a small logic *and* block with the following input conditions.
 - <INBAND.QNOT> which indicates the pump is not within the desired position.
 - <ABORTOR.QNOT> which is high if no abort condition is active.
 - <HILIMAND.QNOT> is high if the pump position is not beyond the high range.

- <LOLIMAND.QNOT> is high if the pump position is not beyond the low range.
- <PSTOP.DOUT> is high if the pump motor is stopped.
- <PRUN.QNOT> is high pump motor is not running.

If there is a problem indicated by <DMPROB> and the test values are set and the pump is not in position <PBDMPROB> goes high which is then passed to <PBPROB> through <PBPROB2>. The small logic *and* block <DMPROBDI> is used to bring up the problem indication display for positioning the pump. When <PBDMPROB> is high and the operator set <PDISP> high to view the problem and there is no problem with the position selection from <POSDISP.QNOT>, the display DMPROB will be brought up via the *display* block <DISP3>.

The small logic *and* block <DISMOV1> will execute the macro DISMOV1 to disable movement of the pump. <DISMOV1> has the following input conditions.

- <DISMOVE.QNOT> is high if movement is enabled.
- <DRUN.QNOT> is high if the position motor is not running.
- <DSTOP.DOUT> is high if the position motor is stopped.

The small logic *and* block <DMDISP> is used to bring up the problem indication display for positioning the pump. When <DMPROB> is high and the operator sets <DDISP> high to view the problem from the MOTOR display the display DMPROB is brought up via the *display* block <DISP1>.

Test Enable. Before test operation can be started a test must be enabled. The enable task is used to check the values set for a test to ensure proper test operation. The blocks in the following table are the variables involved with the test enable function.

BPOSGATE	CHKCOLOR	DO1-6DEV	DO1-7DEV
DO1DEV	DO2DEV	DO3DEV	DO4DEV
DO5DEV	DO6DEV	DO7DEV	ENCHECK
NORUN	PBCHK1	PBCHKCLR	PBENABLE
PBENAND	PBENOR	PBENREST	PBENRSRS
PBENTIMO	PBENTON	POSORBUM	RSCHECK
RSCHECKR	STOPTEST	TSTRUNNG	

The small logic *pulse* block <CHKCOLOR> starts the execution of the test enable task. Its output is passed to the small logic *and* block <PBCHKCLR> which works with <PBENAND> to require the following input conditions before a test can be enabled by setting <PBENABLE> high.

- <CHKCOLOR.DOUT> which is set high by the operator to enable a test.
- <NORUN.DOUT> is high if both motors are stopped.
- <TSTRUNNG.QNOT> is high if a test is not running.
- <PBENABLE.QNOT> is high if a test is not enabled.
- <POSORBUM.DOUT> is high if the pump is in position.
- <DO1-7DEV.QNOT> is high when all of the following conditions are not active. <DO1-6DEV> is also used to compile the conditions.
 - <PBCHK1.DALM> is high when the PLC test time does not match the desired test time.
 - <BPOSGATE.DOUT> is high if the pump is not in position.
 - <DO1DEV.DALM> is high if the pump is not in position.
 - <DO2DEV.DALM> is high if the desired pump speed does not match the pump speed in the VFD.
 - <DO3DEV.DALM> is high if the desired pump acceleration does not match the acceleration set in the VFD.
 - <DO4DEV.DALM> is high if the desired pump deceleration does not match the deceleration set in the VFD.
 - <DO5DEV.DALM> is high if the desired pump maximum speed does not match the VFD feedback.
 - <DO6DEV.DALM> is high if the set speed alarm does not match the PLC feedback.
 - <DO7DEV.DALM> is high if the pump speed abort limit does not match the PLC feedback.

The test enable bit <PBENABLE> is reset when <PBENOR> has one of the following input conditions are true.

- <STOPTEST.DOUT> goes high when a test is stopped.
- <PBENREST.DOUT> is a 4 second time delay of the small logic *and* block <PBENTIMO>. <PBENTIMO> has the following input conditions.
 - <PBENTON.DOUT> will go high if a test is enabled and a test is not started within 60 seconds.
 - <TSTRUNNG.QNOT> is high if a test is not running.

<ENCHECK> is a status bit used to enable color control on the test operation displays for critical parameters. This bit comes from the *rs flip-flop* block <RSCHECK> which is set high by <PBCHKCLR> and reset by <RSCHECKR> which will go high when any of the following conditions go high.

- <SETVALDN.DOUT> goes high when the test values are set.
- <STOPTEST.DOUT> goes high when a test is stopped.

Problem with Test Enable. If the test enable function was desired but not successfully completed the blocks in the following table are used to indicate what the problem with the test enable variables is.

CHKCLRTN	DISP3	DO1-7DEV	NOPENAB
NORUN	PBBUTSTP	PBENABLE	PBENADIS
PBENRSRS	PBPROB	PDISP	PENAB
PENABA	PENABR	POSORBUM	RPENAB
TSTRUNNG			

The *rs flip-flop* block <PENAB> is used to indicate a problem with enabling a test. <PENAB> is set high if the output of <PENABA> is high and the reset input <PENABR> is low. <PENABA> is a small logic *and* block with the following inputs.

- <PBENABLE.QNOT> is high if a test is not enabled.
- <NORUN.DOUT> is high if both motors are stopped.

- <CHKCLRTRN.DOUT> will go high 1 second after the operator tries to enable a test.
 - <POSORBUM.DOUT> is high if the pump is in position.
- <PENABR> will reset <PENAB> if any of the following conditions are true.
- <RPENAB.DOUT> which is set high by the operator to reset the problem indication.
 - <PBBUTSTP.DOUT> which goes high when the operator manually stops a test.
 - <NOPENAB.DOUT> which is a small logic *and* block with the following input conditions.
 - <DO1-7DEV.QNOT> is high if no problem exists with the feedback parameters.
 - <NORUN.DOUT> is high if both motors are stopped.
 - <TSTRUNNG.QNOT> is high if a test is not running.
 - <POSORBUM.DOUT> is high if the pump is at the desired position.
 - <PBENRSRS.DOUT> is high if the test enable has timed out.

When <PENAB> goes high, <PBPROB> is set high which indicates a test operation problem. The small logic *and* block <PBENADIS> is used to bring up the problem indication display for enabling a test. When <PENAB> is high and the operator sets <PDISP> high to view the problem the display, <PBENPROB> is brought up via the *display* block <DISP3>.

Test Start. After test parameters are downloaded and a test is enabled the operator can start a test which will operate the pump for a desired period of time. The blocks in the following table are related to the test start task.

ABORTOR	NORUN	PBENABLE	PBENTON
PBSPLIT	PBSTRAN2	PBSTRAND	PBSTRTDN
PBSTTEST	PLCPBENA	STOPTEST	TSTRUNNG
VALSET			

The small logic *pulse* block <PBSTTEST> starts the execution of the test start task. Its output is passed to the small logic *and* blocks <PBSTRAND> and <PBSTRAN2> which must have the following input conditions true before starting a test.

- <PLCPBENA.DOUT> which is high if the test enable status bit is high in the PLC.
- <PBSTTEST.DOUT> is set high by the operator to start a test.
- <PBENABLE.DOUT> is high if a test is enabled.
- <TSTRUNNG.QNOT> is high if a test is not running.
- <VALSET.DOUT> is high if the test values are set.
- <NORUN.DOUT> is high if both motors are stopped.
- <ABORTOR.QNOT> is high if no abort condition is active.
- <PBENTON.DO2> is high when the test enable has not timed out.
- <PBSPLIT.DO3> is high if the elapsed test time is not greater than or equal to the desired time.

The output of <PBSTRAN2> is passed onto <PBSTRTDN> which starts the pump motor and sets the *rs flip-flop* block <TSTRUNNG> high. <TSTRUNNG> is reset from <STOPTEST> when a test is stopped.

Problem with Starting a Test. If the start of a test is not permitted, the blocks in the following table are used to indicate problems with the start test variables.

ABORTOR	DISP3	NOPSPRO2	NOPSPROB
NORUN	PBUTSTP	PBENABLE	PBENTON
PBPROB	PBSPLIT	PBSTTEST	PLCPBENA
PSDISP	PSPROB	PSPROBA	PSPROBR
RPSPROB	TSTRUNNG	VALSET	

The *rs flip-flop* block <PSPROB> is used to indicate a problem with starting a test. <PSPROB> is set high if the output of <PSPROBA> is high and the reset input <PSPROBR> is low. <PSPROBA> is a small logic *and* block with the following inputs.

- <PBSTTEST.DOUT> is set high by the operator to start a test.
 - <TSTRUNNG.QNOT> is high if a test is not running.
 - <PBENABLE.DOUT> is high if a test is enabled.
- <PSPROBR> will reset <PSPROB> if any of the following conditions are true.
- <RPSPROB.DOUT> which is set high by the operator to reset the problem indication.
 - <PBBUTSTP.DOUT> which goes high when the operator manually stops a test.
 - <NOPRPRO2.DOUT> which is a small logic *and* block that works with <NOPSPROB> to compile the following input conditions.
 - <PBENABLE.DOUT> is high if a test is enabled.
 - <PLCPBENA.DOUT> is high if the test enable bit in the PLC is high.
 - <ABORTOR.QNOT> is high if no abort condition is active.
 - <VALSET.DOUT> is high if the test values are set.
 - <NORUN.DOUT> is high if both motors are stopped.
 - <PBENTON.DO2> is high if the test enable did not time out.
 - <PBSPLIT.DO3> is high if the elapsed test time is less than the desired time.

When <PSPROB> goes high <PBPROB> is set high which indicates a test operation problem. The small logic *and* block <PSDISP> is used to bring up the problem indication display for starting a test. When <PSPROB> is high and the operator sets <PDISP> high to view the problem the display PSPROB is brought up via the *display* block <DISP3>.

8.2.5 PLC Communications

PLC communications are enabled in the MOTOR strategy for several functions. One is to receive the actual position of the pump to be used in the position pump logic. Also

the PLC abort coil status is read into the strategy to stop either motor and disallow the start of a motor if it is active. The desired operating parameters and abort levels that are set in the MOTOR strategy are sent to the PLC logic to be used for logic. The current values of the critical variables from the VFDs are also sent to the PLC for abort logic and to be transferred to the TEST strategy for display on STATIONS5. Tables 14 and 15 list the connections that are used for communications to the PLC.

Table 14. Device Setup Parameters for MOTOR Strategy DEV 3

DEV 3					
Device #:		2	Scan Task:		Task: 3
AIN Raw Count-Zero:		0	AOUT Raw Count - Zero:		0
AIN Raw Range:		65535	AOUT Raw Range:		65535
BCD:		NO	Bit Order:		16-1
Group	Start Address	End Address	Group	Start Address	End Address
IIC0	402129	402136	I2C1	400208	400215
IIC11	402217	402224	I2C2	400216	400223
IIC12	402225	402232			

Table 15. Device Setup Parameters for MOTOR Strategy DEV 4

DEV 4					
Device #:		2	Scan Task:		Task: 3
AIN Raw Count-Zero:		0	AOUT Raw Count - Zero:		0
AIN Raw Range:		4095	AOUT Raw Range:		4095
BCD:		NO	Bit Order:		16-1
Group	Start Address	End Address	Group	Start Address	End Address
J1C0	402001	402008	J3C1	242	N/A
			J3C3	244	N/A
J2C0	2001	2016	J3C8	249	N/A
J2C1	2017	2032	J3C10	251	N/A
J2C2	2033	2048			

To report a failure in the PLC, the *DIN* block <PLCFAIL> will alarm in the MOTOR strategy. <PLCFAIL> receives the signal from <PLCFAIL2> which is a small logic *or* block that has three input conditions.

- <PLCWDON.DOUT> goes high if the PLC toggle bit <PLCWD stays high for more than 2 seconds.
- <PLCWDOFF.DOUT> goes high if the PLC toggle bit <PLCWD> stays low for more than 2 seconds.
- <IIC0.FAIL> which will go high if there is a failure in communications in the ModPlus GENESIS driver.

8.2.6 Alarming

The operators are notified of signals that exceed limits and of the status of events with the alarming function of genesis. All algorithm blocks with an alarm priority greater than 0 are enabled as alarming signals. When an alarm condition arises the computer drives an amplified audible alarm, and the tag name of the signal that caused it is displayed in the lower right corner of the display. The tag name and description also shows up in the alarm summary to tell the operator which signal is in alarm.

The alarms set in the motor strategy pertain mostly to the pump motor and the directional motor. There are some additional signals that are received from the PLC that are also alarmed.

Operator activity is also tracked in the alarm/event summary. If a value is changed by the operator, a record of the change is logged into the alarm/event summary.

8.2.7 System Parameters

There are special algorithm functions in the MOTOR strategy that are used to execute Report & Recipe (R&R) code, and for version tracking. A list of Report blocks in the strategy with the R&R code that each block serves is shown in Table 16. Refer to Appendix C for a description of each R&R code.

Table 16. Report & Recipe Codes Used in the MOTOR Strategy

TAG	SOURCE1	SOURCE2	SOURCE3	SOURCE4
TESTSET2	ULOAD	NEXT	PREV	ILOAD
PBREPORT	PBSETVAL			
TESTSET	SAVE	DELETE	CLRMESS	
INITENAB	INITENAB			

The *d flip-flop* block in the center of the database contains the version of the strategy as its tag name. For example, <V3.05> would be the tag name for version 3.05 of the strategy. The block <VERSION> contains the version number in the descriptive field.

8.2.8 Stop Test

The stop test task will stop operation of either motor and reset operator status feedback. The following blocks are used for the stop test task.

ABORTPUL	DISP4	PBBUTSTP	PBDNTM
STOPTEST			

The operator can manually stop a test by setting <PBBUTSTP> high. The output of <PBBUTSTP> is passed to <STOPTEST> which stops the motors and resets the desired pump operation time with the macro RESPTIM in the *display* block <DISP4>. <STOPTEST> also resets the operator feedback status bits for test operation. <STOPTEST> also is controlled by <ABORTPUL.DOUT> which goes high if an abort condition becomes active. <PBDNTM.DOUT> goes high if the elapsed pump operation time exceeds the desired time and sets <STOPTEST> high.

8.3 MODICON PLC FUNCTIONS

The Modicon PLC is responsible for inputting data from the field instrumentation, providing an abort signal when the values of critical measurements exceed their abort limits, providing control signals to field instruments, providing a timer and enable logic for pump operation, sending data and status information to GENESIS and providing a directional motor simulation. Details of these functions will be presented in the following sections.

Access to and programming of the PLC is provided by the program Modsoft. Modsoft is run on STATION1 in the DACS trailer and communicates with the PLC via the Modbus Plus network.

Using Modsoft, the PLC I/O configuration can be defined, the PLC registers can be allocated and named, and the ladder logic can be produced (refer to Appendix D for additional information on the registers and corresponding symbol table). All of this can be done offline and later downloaded to the PLC, or can be done online in the PLC as it operates. The latter is not a wise policy for making permanent changes, but can be useful for debugging ladder logic.

Modsoft allows the ladder logic programmer to impose a structure on the ladder logic for ease of understanding. It allows the programmer to break the ladder logic into modules known as objects and to connect the objects using flow control logic. Each object can contain any number of ladder logic networks. This structure is converted by Modsoft to an equivalent flat nonstructured form when the logic is downloaded to the PLC. In addition,

the ladder logic can be distributed in different logic segments. The segments provide another way to organize the ladder logic program. The logic in each segment can be set to be solved either on each pass or conditionally. Also, an I/O drop can be associated with a segment; the drop inputs and outputs are processed when the associated segment is active.

The current PLC program has logic in the first four segments. The first segment contains only the hot standby block. This block controls the hot standby functions of the PLCs. The second segment contains most of the operational logic and will be discussed extensively below. The third segment contains the block copy logic for creating the GENESIS I/O region. The fourth segment contains the directional motor simulator logic.

Segment two is the only segment in which the logic is divided into objects. It currently has seven objects (P000, P001, and P003 through P007) which execute sequentially (no control structures mediating them). Their functions are listed below:

- P000 (15 Networks) - PLC status logic, test timer and enable logic, instrument control logic.
- P001 (1 Network) - PLC rack and module status logic.
- P003 (64 Networks) - Abort logic and instrument fail logic.
- P004 (10 Networks) - RGA5 ASCII/BASIC module control logic.
- P005 (13 Networks) - GC3 and FTIR ASCII/BASIC module control logic.
- P006 (5 Networks) - High frequency strain alarm filtering.
- P007 (54 Networks) - Thermocouple module setup and control logic.

Modsoft has a documentation feature which allows information about the PLC configuration and ladder logic to be either saved to a file or printed.

8.3.1 Data Collection From the Field

The PLC is responsible for collecting data from the field instruments. These data enter the PLC through any of a number of Modicon I/O modules. There are several different types of modules which accept different types of input - analog voltage or current inputs, thermocouple inputs, digital inputs of varying levels or ASCII serial inputs. There are corresponding types of modules for data output. The input and output modules used in the system and their descriptions are listed in Table 17. The output modules are used for various control functions discussed in Sections 8.3.2 and 8.3.3.

Table 17. Input and Output Modules Used in DACS

Module	Description	Signals Accepted
B875-101	8-channel analog input	Configurable for various voltage or current input ranges
B827-024	32-channel digital input	24-V digital input
B885-002	ASCII/BASIC module	RS-232 serial ASCII data
B829-016	16-channel digital input	5-V TTL digital input
B883-200	Thermocouple module	Thermocouple types B, E, J, K, N, R, S, & T
B865-002	TTL Register input	5-V digital output
B824-016	16-chan. digital output	24-V digital output
B828-016	16-chan. digital output	5-V TTL digital output

In order to access data from a module, it must be entered into the PLC configuration table known as the Traffic Cop. This involves specifying in the configuration module of the Modsoft program the module's type, physical location, and the PLC registers which will be used for communication with the module. The physical location is specified in terms of the Modicon I/O drop, the I/O rack within the drop, and the slot within the rack which contains the module.

For most modules, the PLC I/O hardware places the data into the specified registers. From there, the data can be used within the PLC if further logic operations or abort comparisons are required.

The exceptions to this are the B883 thermocouple input modules and the B885 ASCII/BASIC input modules. Both of these require that PLC logic be provided to program module parameters, control the operation of the modules, and handle the interface between the modules and the PLC. In addition, the ASCII/BASIC module contains a built-in BASIC interpreter. A BASIC program must be loaded into this module to control the input of data into the module as well as the interface between the module and the PLC. The logic and programs which operate these modules are discussed in the following sections.

Most of the measurements from the field enter the system through the B875 analog input module. This module can be set to accept inputs with differing characteristics. Commonly used in this system are 1-5 V, 0-5 V and 4-20 mA signals. These inputs are digitized with 12 bits of resolution. This 12-bit value is stored in the lower order 12 bits of the Modicon register corresponding to that channel. The most significant bit of the same register is the out-of-range bit. The B875 module will set this bit if it detects an incoming signal which is out of the range specified when the module is set up. For example, if the module has been set to receive 1 to 5-V inputs and then receives an input of 0 Vs, it will set the out-of-range bit. This out-of-range bit is used for instrument failure detection, a topic which is discussed in Section 8.3.5.

The data from all modules are copied into a contiguous set of registers before being sent to GENESIS. This is done to increase the PLC to GENESIS communication efficiency. Refer to Appendix E for the detailed I/O channel list.

8.3.1.1 ASCII/BASIC Module Operation

The Modicon B885 ASCII/BASIC Module allows ASCII data from a computer or from analytical equipment to be read into the Modicon PLC. The ASCII data are presented via one of two RS-232 ports on the front of the module. The module contains a stripped-down BASIC interpreter. BASIC programs can be written that read the ASCII data from the ports and pass it to the PLC via the command interface registers.

The Module has two operational modes: RUN and PROGRAM. These are selected via a switch on the front panel. In PROGRAM mode, the module communication parameters can be set and the BASIC interpreter accessed for programming. In the RUN mode, the module is under the control of the PLC ladder logic via the command/status registers or under control of its own internal scheduler. These are used to determine which BASIC program stored in the module's memory to execute, and when.

Details of the module configuration, programming, and operation are given in the Modicon B885 ASCII/BASIC Module User Guide.

The DACS uses two ASCII/BASIC Modules: one for the RGA5 gas data (RGA5), and one for gas chromatograph, infrared spectrometer and photo gas concentration data (GC3) (see Appendix F for complete file listings). Both of these require a BASIC program to read the ASCII stream, strip off unwanted characters, format the data for the PLC registers, and send this data to the PLC. They also require ladder logic to control and monitor the operation of the BASIC programs, receive the data from the interface registers, and place it in the appropriate final destination registers.

The BASIC programs and ladder logic are therefore quite similar in concept but differ due to the differing data formats and conventions used by the devices. The RGA5 ladder logic is in Segment 2, Object P004 and the gas chromatograph ladder logic is in Segment 2, Object P005.

8.3.1.2 ASCII/BASIC Module Communications with the Gas Chromatograph

The gas monitoring computer sends its data to the ASCII/BASIC module as a stream of ASCII characters beginning with a STX (start of transmission) character and followed by 14 four-digit ASCII encoded numbers separated by carriage returns (↵) (see Table 18). These communication parameters are: 1200 Baud, 8 bits, 1 stop bit, no parity.

Table 18. GC-3 Data Format

Label	Description	Value Transmitted
STX	Start of Transmission	2
GC3-TIME	GC3 Time of Sample	nnnn←
GC3-AREA	GC3 Peak area	nnnn←
GC3-RT	GC3 H2 Retention Time	nnnn←
GC3-H2	GC3 H2 Concentration	nnnn←
GC3-FILE	GC3 File ID	nnnn←
FT-TIME	FTIR Time of Sample	nnnn←
FT-N2OA	FTIR N2O Area	nnnn←
FT-N2OC	FTIR N2O Concentration	nnnn←
FT-NH3A	FTIR NH3 Area	nnnn←
FT-NH3C	FTIR NH3 Concentration	nnnn←
FT-FILE	FTIR File ID	nnnn←
PHO-TIME	Photo NH3 Time of Sample	nnnn←
PHO-MSB	Photo NH3 Conc. - most significant byte	nnnn←
PHO-LSB	Photo NH3 Conc. - least significant byte	nnnn←

nnnn← = four ASCII numeric characters followed by a carriage return.

The BASIC program (filename GC3.BAS) in the ASCII/BASIC module is responsible for reading these incoming characters, interpreting them, placing them in registers and sending them to the PLC through the module's command/data register interface.

The PLC ladder logic is responsible for running the BASIC program, issuing commands to read the status of the module, and to read data from the module and placing the data into the proper registers so it can be sent to GENESIS.

Communication between the PLC and the BASIC module is through the 6 input and 6 output interface registers assigned to the module during configuration. In addition, there is an array of 100 registers internal to the BASIC module which can be accessed by the PLC (up to 5 at a time) through these interface registers. There are also eight status bits which can be set or reset directly by the BASIC program and read by the PLC along with the hardware status bits.

In general, the normal sequence of operation of the system is as follows (for more specifics, see the ladder logic comments, the BASIC program listing and the ASCII/BASIC module manual):

PLC Operation

1. The ladder logic checks the ASCII/BASIC module status register. If there is an error condition, it resets the module. If the BASIC program is not running, it issues the command to run the BASIC program.
2. The ladder logic continues to monitor the status register for error conditions and checks the data ready user flag (GC3_SND8). This is the signal from the BASIC program that a set of data has been read and is ready to be transferred to the PLC.
3. When the data ready flag has been received, the ladder logic goes through a sequence of commands which transfer the data from the ASCII/BASIC module's internal register array. The data are transferred one to four registers at a time and stored in registers in the PLC for transference to GENESIS. The 14 data items to be transferred are stored in the ASCII/BASIC module's internal registers 0-13. The PLC ladder logic issues 7 commands in sequence to transfer these 14 data items.
4. The ladder logic sets the ASCII/BASIC module's internal register 30 to 1 (data transfer complete flag). This is the signal to the BASIC program that the data transference is complete.
5. Go to step 1.

BASIC Program Operation

1. The program is initialized by the PLC ladder logic. It immediately sets the data ready flag to zero (although it should already be zero) and the data transfer complete register (register 30) to zero.
2. The program reads characters from serial port 1 until a STX character is received. The start of transmission character signals the beginning of the gas data stream.
3. After the STX is received, the program reads 14 numeric values and stores them in internal registers 0 through 13.
4. The program sets the data ready flag [SND(8)], indicating to the PLC that there is a set of data in the registers to be transferred.
5. The program waits until the data transfer complete flag is set (Register 30). This will indicate that the PLC has transferred the data.

6. The program clears the data ready flag, then exits. When the PLC detects that the program has exited, it will rerun the program.

8.3.1.3 ASCII/BASIC Module Communications with the RGA5 Computer

Following is the data format for the RGA5 ASCII/BASIC module communication:

```
Run=nnn, Stream =ccccccc, Date=nn-nn-nn, Time=nn:nn
H2A=nnnn.n, H2A Area=nnnnnnnnn, H2ART=nnn, H2Arf=nnnnn.n
H2B=nnnn.n, H2B Area=nnnnnnnnn, H2BRT=nnn, H2Brf=nnnnn.n
```

where, n = any numeric character and c = any printable ASCII character. *Run* is the run number and is incremented by one with each data stream. *Stream* is set to Tank, Flush or Calib to indicate the source of the data. The BASIC program only looks at the first character of this field. *Date* is the sample date. This is ignored by the BASIC program. *Time* is the time of sample in the format hours:minutes. *H2A* is the GC-1 hydrogen concentration in ppm. *H2A Area* is the GC-1 hydrogen peak area. *H2ART* is the GC-1 hydrogen retention time in seconds. *H2Arf* is the GC-1 hydrogen retention factor. This is ignored by the BASIC program. *H2B* refers to the same information as above except it pertains to GC-2 instead of GC-1. The numeric data will be right justified with leading zeros inserted to preserve the field widths.

The communications parameters are: 1200 baud, 8 bits, no parity, 1 stop bit. The stream must enter the ASCII module via port B. The XON/XOFF protocol should be enabled.

The BASIC program operates by searching the ASCII stream for the word "Run", indicating the start of the data. It then issues reads to the ASCII message processor to input the data. After the data has been input, the program places the data into contiguous registers to be read by the PLC. In some cases a datum is placed unchanged into a register. In other cases, the data is scaled or manipulated in some way in order to make it fit into a 16 bit register. The PLC is then notified that data is available. When it has read the data, it notifies the BASIC program that the data has been read. The program then exits and is run anew by the PLC.

The issuing of ASCII read messages by the program means that these message formats must be loaded into the memory of the ASCII portion of the ASCII/BASIC module. The following are the ASCII read message formats:

```
RMSG1: 10 S0,D3
RMSG2: 10 S1,A1
RMSG3: 10 S2
RMSG4: 10 S2,D2,D2
RMSG5: 10 S4,F6.1
```

```

RMSG6: 10 S6, D1, D4, D4
RMSG7: 10 S9, D3
RMSG8: 10 S10
RMSG9: 10 S10, F6.1
RMSG10: 10 S12, D1, D4, D4
RMSG11: 10 S15, D3
RMSG12: 10 S16

```

In addition, the program uses the prefix capability of the ASCII module to determine the beginning point of the ASCII reads. The prefix instructs the ASCII message processor to ignore all incoming characters until the prefix string is read, then begin the formatted read. In this case, the prefix string of the module must be set to “.” The command to be issued is: PR# 3Dh. Also, the delimiter character should be set to null: DL# 0.

The program must be loaded into RAM 2 of the ASCII/BASIC module.

8.3.1.4 Thermocouple Module Operation

The B883 thermocouple modules receive thermocouple inputs from the MIT17B and MIT17C tank temperatures, and the tank bottom and side thermocouples. Three groups of modules are used: three modules to input the MIT17B tank temperatures (22 measurements), three modules to input the MIT17C tank temperatures (22 measurements), and three to input the tank bottom and side temperatures (26 measurements). These modules require ladder logic to set up and operate the modules.

The ladder logic can be broken into three major sections: (1) creation of the setup table and command table, (2) configuring the module, and (3) operating the module. The configuration of the three modules of each group is the same so only two setup tables are needed. The following is an overview of the ladder logic needed to operate one of the modules.

Figure 25 shows a portion of the ladder logic used to create the thermocouple module setup table. Since no predefined constants are available in ladder logic, the contents of the configuration table must be created from scratch when the PLC is first brought online. The SUB blocks are used to load the values in the top node of the SUB blocks into the registers indicated in the bottom node of the SUB blocks. These registers form the setup table. They are grouped in threes, since these represent command parameters which will later be sent to the thermocouple modules. In the figure the #0301-#0000-#0000 sequence is the thermocouple module STOP command, and the #0290-#0000-#0000 sequence is the thermocouple module ENTER CONFIGURATION MODE command. A complete list of the available commands can be found in the thermocouple module manual.

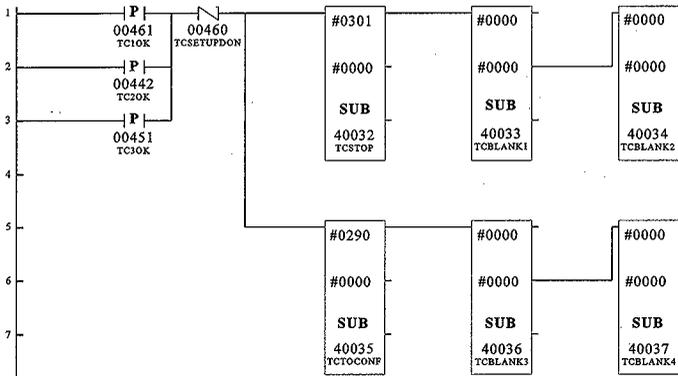


Figure 25. Ladder Logic for Thermocouple Setup Table Creation.

The setup table, when complete, contains a set of commands which will be issued sequentially to each module in the group during configuration mode. These commands will configure the module, giving it the thermocouple type to expect upon input and the data format in which to report the channel values. Each channel must be configured separately so, although in our case all of the channels are configured the same, we must still set the configuration parameters for each channel.

Figure 26 shows a portion of the ladder logic for command table creation. The command table is a list of commands which will be issued to the module one after another when the system is in operational mode. In our case, the command table consists of 10 read channel value commands, one for each of the ten channels available on a single module. In this case, *ADD* blocks are used to set the channel parameters. The E984 PLCs do not allow constants larger than 999 in a block, so, for example, the “read channel one value” command, 1001, is created by adding 2 to 999.

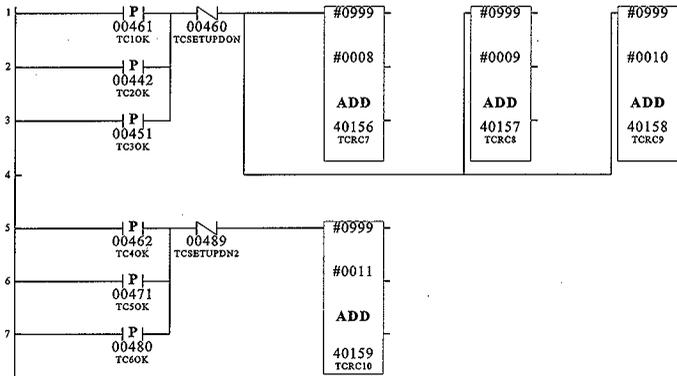


Figure 26. Ladder Logic for Thermocouple Command Table Creation.

After these two tables have been created, the modules must then be configured according to the commands now in the setup table. Figure 27 shows the logic which issues these configuration commands. Register 40173 is the setup table pointer. This logic will load the three register command sequences into the command registers, 40091-40093, incrementing the setup table pointer each time. This is done until the end of the setup table is reached and coil 435 is set, indicating that configuration is done. The next set of commands is issued when coil 434 goes high indicating that the previous command has been accepted by the module. The setup sequence can also be initiated when a module first comes online (coil 461 goes high) or after an error has occurred (coil 437 goes high).

After the configuration is complete, the thermocouple modules enter operation mode. In this mode, the commands in the command table are executed. When the end of the table is reached, the system resets the command table pointer to the beginning of the table and goes through the table again. The command table consists of commands to read each of the 10 thermocouple data values one after the other.

In Figure 28, register 40150 is the beginning of the command table. Register 40171 is the index into the command table, and register 40172 is the register which receives the value pointed to by the combination of 40150 plus the offset in 40171. This value is loaded into the thermocouple module command interface register, 40091, thereby issuing the command.

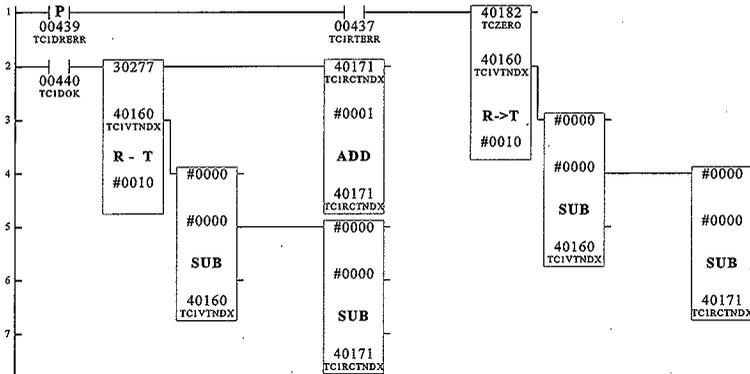


Figure 29. The Thermocouple Modules Reading Thermocouple Data Values.

The upper pathway is activated when there is an error (coil 437, runtime error, is high). This places a zero into the data value table and resets the data value pointer and the command table pointer.

8.3.2 Abort Functions

One of the primary monitoring functions of the Modicon 984 PLC in the DACS system is to check for over- or under-range conditions on various safety parameters. The PLC is responsible for providing an automatic pump shutoff signal (known as the abort signal) whenever these safety-related or other critical measurements exceed predetermined limits. The abort limits are sent to the PLC from GENESIS. The PLC then continually compares the incoming critical measurements to their abort limits and sends the abort signal if any of the limits are exceeded. This function is provided in the PLC rather than in GENESIS since the response time of the PLC is much faster (on the order of 50 ms).

The parameters are sensed by the 984 series I/O at drops or I/O stations distributed around tank 241-SY-101 at surface level, and then transferred to the 984 PLC inside the DACS trailer. Once received by the PLC, these critical values are stored in 30xxx registers. These are two-byte registers that function as repositories for incoming data. Current values of the important parameters are now loaded in the PLC, and are ready for comparison with limit values. This comparison takes place in the ladder logic in the PLC and is of the form below in Figure 30. Register A is the current value register.

Register B is the 40xxx register containing the limit to be compared with the current value. Register C is the 40xxx register containing the difference between the values in A and B. The subtract block actually has three outputs available for user logic. The top output is activated if A is greater than B. This is the case in which the limit abort logic is mainly interested. With this line high and

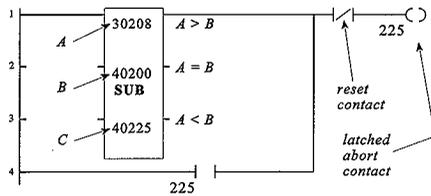


Figure 30. Abort Limit Testing.

the reset contact sense in the nonreset condition, the latched abort contact 225 is closed. When it closes, the contact sense in the bottom leg holds or latches the contact closed. Nothing can interrupt power to contact 225 except if the reset contact should close. The reset contact in the above diagram is a “not” of the true reset contact. This means that if the reset contact is open, the ladder element conducts power (is closed).

Values for the abort and alarm limits stored in the 4xxx registers are written from the TEST strategy in STATION5 to the PLC.

In the PLC, there exists ladder logic similar to the above for every limit abort parameter; however, the latched abort contact is distinct for each parameter. The reset contact is common to all the logic. Figure 31 shows how the many abort contacts are OR’ed together to set a single abort contact. Here the reset contact is identical to that in the previous diagram and all other limit abort logic. The composite abort contact is held closed by the latching leg containing the sense of contact 021. It can only be reset by making the reset contact close. This is done from GENESIS. Notice below that the abort contact 225 from the logic above is represented as only one of many such contacts combined to set the composite abort contact.

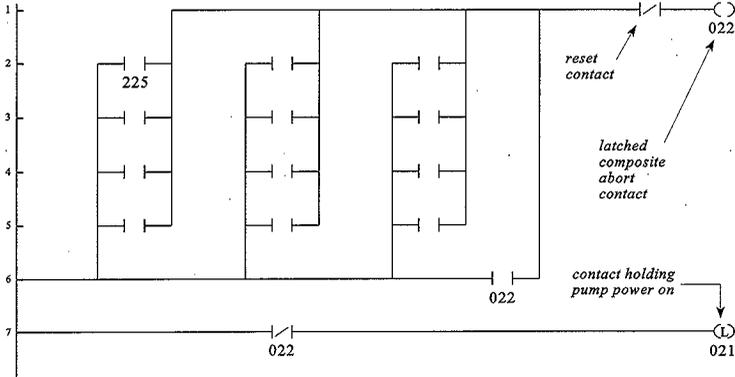


Figure 31. Composite Abort Contact.

When the latched composite abort contact closes, there is a contact shown in the bottom right-hand corner of the logic above that is opened upon an abort. This power contact is directly connected to a digital output Modicon module that normally holds the power breaker to the pump and directional motors closed. When this contact is opened, the power to the motors is interrupted.

When any of the latched abort contacts closes, a display in STATIONS5 called CSMAIN shows the class of the abort along with the alarm limit value, the abort limit value, and a button to another screen with more detail on the abort.

The class of the abort will be one of the following:

- | | |
|---------------------------------|------------------------------|
| High H2 Concentration | High Pump Speed |
| High Pump Flow | High Waste Temperature |
| High Duration Limit | High Pump Motor Current |
| High Pump Motor Oil Temperature | High Pump Motor Oil Moisture |
| High Pump Column Strain | High Pump Column Vibration |
| Low Pump Column Gas Pressure | MIT Column Strain |
| VDTT Column Strain | Low Ventilation Flow |
| High Tank Dome Pressure | |

There is also an indicator of the status of the PLC abort coil and a reset button for it. The text on this screen signifying the class of the abort becomes red if a tag in its class has caused an abort. If the operator presses the button next to a class, another more detailed screen is called up. On this next level of display tag names, tag descriptions, alarm and abort limit values, tag current values, and units are listed. In addition, there is an indicator that other alarms or abort conditions exist as well as an indicator for the PLC abort contact.

As a safety factor, the composite abort contact is read by the GENESIS strategy in STATION8. The tag name of the particular block holding this state is <LIMABORT>. When the MOTOR strategy in STATION8 detects a problem, a report executes that issues stop commands to both the pump motor drive and the directional motor drive.

8.3.3 Control Points and Alarm Outputs

The PLC provides a few control and alarm signals other than the pump motor shutoff signal. Other control outputs provided by the PLC are: a signal which triggers the Nicolet data collection system whenever there are indications of a tank rollover, and a signal which shuts off the VDTT flow meters when a high strain is detected on certain risers.

The PLC also contains a watchdog timer which expects to receive regular updates from GENESIS. If communications from GENESIS STATION5 or STATION8 to the PLC fail, a signal is sent which sounds an audible alarm. Also, an oscillating signal is provided from the PLC which resets watchdog timers in GENESIS to monitor communication from the PLC to GENESIS.

The Nicolet trigger signal is set whenever any of the following occur: (1) there is a high-frequency strain alarm condition, (2) there is a low frequency strain alarm condition, or (3) the pump motor is running. The Nicolet system captures high-speed strain data. This allows the system to be triggered during a tank rollover.

Figure 32 shows the GENESIS to PLC watchdog timer ladder logic. GENESIS sends an alternating digital signal which comes into on PLC coil 213. This signal alternately resets one of the two timers while allowing the other to run. If communication from GENESIS ceases, one of the two timers will be stuck in the run state and will eventually time out, setting coil 20. This coil is wired to an audible alarm circuit, which produces an alarm when the coil is set.

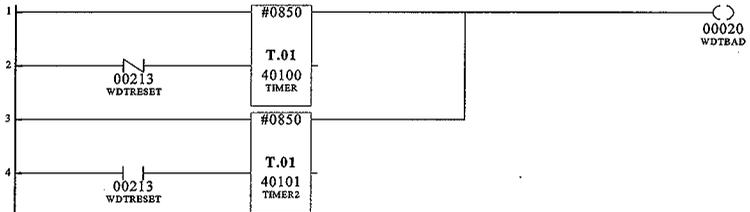


Figure 32. GENESIS to PLC Watchdog Timer.

8.3.4 Test Timers and Enable Logic

The PLC contains a timer which times the pump tests. The time allowed for the test is sent from GENESIS. The PLC timer then runs whenever the pump is running and shuts down the pump whenever the allowed time is exceeded. This provides a backup means for shutting off the pump when the allowed time for the test has elapsed. In most cases, the operator or the MOTOR strategy will have already shut off the pump before the allowed time has elapsed.

The PLC also contains test enable logic. This is a logic mechanism which ensures that the timer is set and abort parameter changes are accomplished before the pump is run.

The PLC ladder logic which implements a test enable mechanism and timer for timing pump tests is shown in Figure 33.

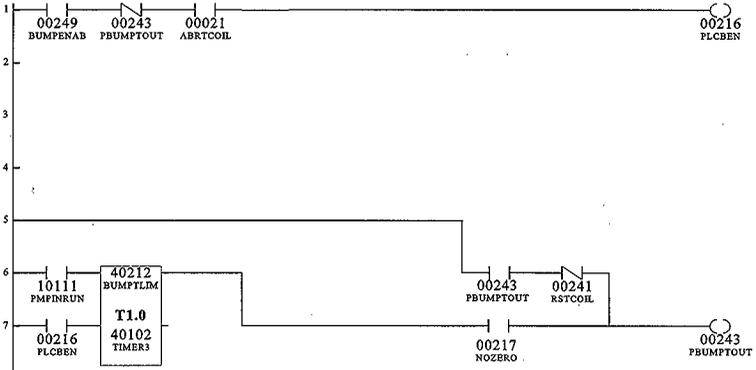


Figure 33. Test Enable and Timer Logic for Pump Bump.

The pathway in the upper part of the figure is the test timer enable mechanism. The test enable signal originates from STATION8. The operator presses the test enable button on one of the pump run screens. The enable signal is passed from the MOTOR strategy to the PLC where it becomes coil 249. This enable signal starts the test timer as long as there is no abort current in the PLC logic (indicated by coil 21). The enable signal must remain set continuously in STATION8 for the duration of the run. Coil 243 causes the timer to reset when the test time has elapsed regardless of the state of the enable from STATION8. This coil is also incorporated into the pump abort logic, causing the pump to shut off when the allowed test time has elapsed.

The timer block is set to time the duration of a pump run and will shut off the pump if the elapsed time exceeds the allowed time. The allowed time is loaded into register 40212 by the Motor strategy. The elapsed time is in register 40102. The timer will run only if the test has been enabled (coil 216 is high) and the pump is running (input 10111 is high). If the allowed time elapses the output of the timer will go high. This sets coil 243 high provided that coil 217 is high. Coil 217 is set only if the timer count and elapsed time are both nonzero. This prevents the timer from causing an abort condition before the test has begun.

8.3.5 Data Transfer and Status Information

Most data to be transferred from the PLC to GENESIS is organized into contiguous blocks before transfer. This increases the efficiency of the data transfer. This is accomplished in segment 3 of the ladder logic by using block moves.

Other than the ASCII/BASIC module data and the thermocouple data discussed in Sections 8.3.1.1 and 8.3.1.2, there is one other case in which additional PLC processing is needed to prepare data for GENESIS. This is required to slow down the possibly momentary high-frequency strain alarms so that GENESIS can have enough time to detect them before they disappear.

Figure 34 shows the PLC ladder logic used to extend the duration of the high-frequency strain alarms. An alarm can appear as a high on any of the input register 10193, 10195, 10197, etc. When a positive-going pulse is detected on any of these channels, and if an extended alarm is not already present for that channel, coil 167 is set. This coil indicates that a new alarm has arrived.

In addition, one of the coils 209-215 is set and latched. The coil set is the one corresponding to the alarm received. These are the coils which are sent to GENESIS to become the high-frequency strain alarms. The incoming signal on the 10xxx register could have been just a momentary signal that disappeared before the alarm was transferred to GENESIS. The latching mechanism ensures that the alarm will remain long enough to be sent to GENESIS.

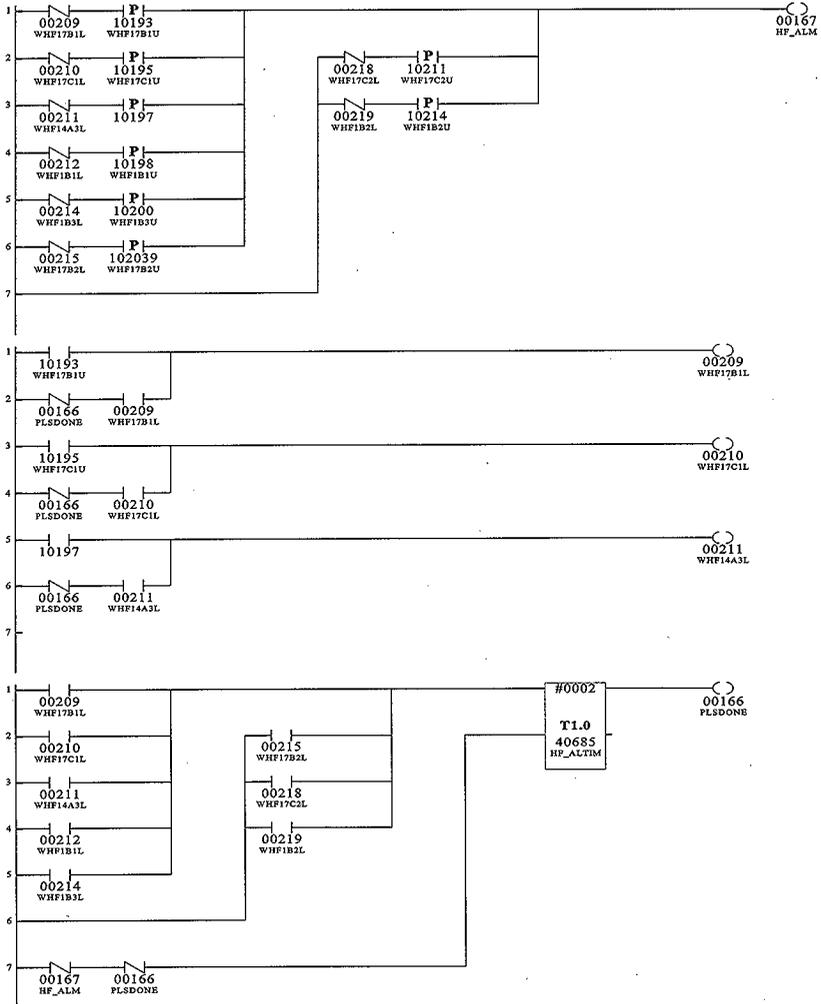


Figure 34. High-Frequency Strain Alarm Monitoring Logic.

To reset the latches, a timer block is provided. This timer will begin to run if an alarm is received and will time out after 2 seconds causing any latched alarms to be reset. If a second alarm is received before the timer has timed out, the timer count will be reset and the timer will continue to run. This ensures that all latched alarms will remain latched for at least 2 seconds.

The PLC also provides status information to GENESIS. It provides a signal which monitors the PLC and the hot standby PLC. This signal is 1 if both PLCs are okay, and 0 if either of them is offline. This information comes from extracting information from the PLC status table which begins at register 40301. Refer to the Modsoft PLC manual "Modicon 984 Programmable Controller Systems Manual" for the details of the status table.

The PLC also provides information to GENESIS about the health status of I/O modules in the system. The following four conditions must be met for a module to indicate good health:

- The slot has been configured in the Traffic Cop.
- The slot contains a module with the correct personality.
- Valid communications exist between the module and the J890.
- Valid communications exist between the J890 and the 984 PLC.

Module health is stored in registers in the Modicon PLC. Different PLCs use different registers. To standardize access to this information, a STAT block is added to the ladder logic. During runtime, the status information is moved to registers defined by the STAT block.

None of the drops are full of modules. Thus, not all of the status registers are of use. A STAT block register contains the I/O health of one crate (alternately referred to as a rack) of modules on a drop. All told, 13 racks are currently in use.

To minimize the communication load, the status registers that correspond to existing racks are copied into a set of transmit registers. It is the transmit registers that GENESIS views.

The movement of the STAT registers to the transmit registers is accomplished by using BLKMs (block moves). The order of the STAT registers is by drop (1 through 32) and by rack (1 through 5) within each drop. Only in cases where two crates have been used within a given drop are the registers contiguous and thus can be handled by a single block move, as shown in Figure 35.

To streamline communication, the GENESIS TEST strategy gathers the transmit registers in as packed data. The 13 registers fit within 13 packed digital inputs (PDIN) fanned out from DEV 4, GC11 (group 11). The names of the PDIN blocks reflect the drop and crate numbers of the involved modules. The drop number is prefaced by a D, the crate

number is prefaced by an R (for Rack). Thus, the *PDIN* block for drop 2 crate 1 modules is named D2R1.

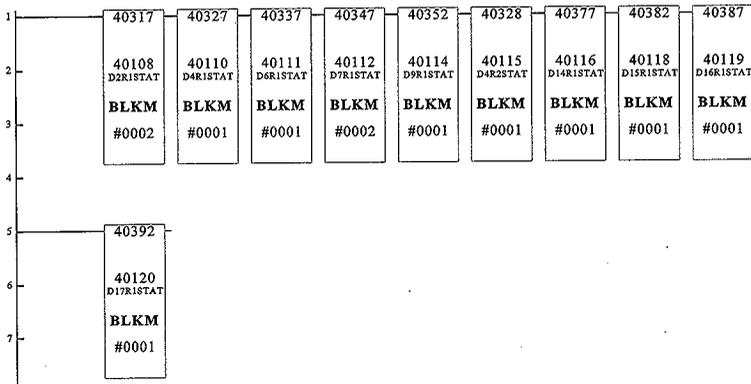


Figure 35. Block Moves for Transmitting I/O Health Status.

Each *PDIN* block contains 16 bits. Each block represents modules 1-11 in a given crate (plus 5 unused bits). Because Modicon defines its bits opposite of GENESIS, this corresponds to bits 16 through 6 in GENESIS. (The data also require byte swapping, which is provided in DEV 4.) Note that alarms for *PDIN* blocks are reported by bit number.

Display of the I/O health is available on the graphic IOSTATUS. Each point in the matrix is tied to the *PDIN* block bit corresponding to the module's health. Thus, the matrix point for drop 2 crate 1 slot (or module) 4 is tied to D2R1.DO13.

Good health is indicated by a green OK; bad health is indicated by a red BAD. On startup of GENESIS, the graphic will show all the modules in bad health until one scan of transmit registers has been made. The scan rate of the transmit registers is 30 seconds.

A complete table of these PLC I/O status registers is given in the "I/O Health Status Implementation Table" in Appendix G.

8.3.6 Directional Motor Simulator

Segment four of the PLC program contains the directional motor simulator logic. This logic simulates the rotation of pump in the lab environment. This logic receives the directional motor acceleration setting, the directional motor speed setpoint and whether or not the motor is in forward or reverse from the Motor strategy. It then increments or decrements the value of ZIMPE112, the position indication.

The simulator is enabled by setting coil 171 to 0 using Modsoft. It also requires the real position encoder (ZIMPE112) be at 0. If there is a non-zero reading for ZIMPE112, then the simulation is disabled and the real value is passed to GENESIS.

The simulation logic calculates the number of seconds required to move the directional motor 1 degree. This calculation is based upon the directional motor simulated speed (RPM). The value is calculated using a value of 13.09 degrees for each revolution of the pump motor.

8.4 NETWORK LAYOUT

The GEN-NET network is composed of eight GEN-NET network stations with ARCNET in a star configuration, as illustrated in Figure 36.

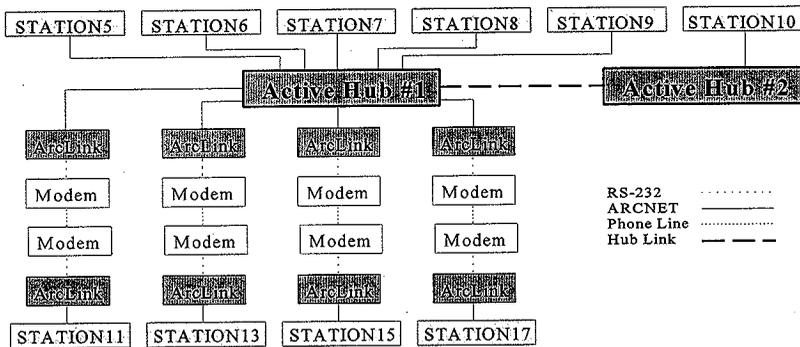


Figure 36. GEN-NET Network Configuration

The GENESIS system allows peer-to-peer communications between GENESIS applications. STATION5 serves data to all other stations on the GEN-NET. STATION6, STATION7, STATION10, STATION11, STATION13, STATION15 and STATION17 all link directly to STATION5 and share the database TEST strategy running on STATION5. These stations accomplish this by running Remote Supervisory Software with the TEST strategy on each station.

STATION5 uses GEN-NET software to enable networking to a Runtime master station. STATION8 is connected to the network for offline file transfer but is not connected when the motor Standby is running.

STATION7 normally serves as a Remote Supervisory Station but is also used as a backup for STATION5 or STATION8. In this instance STATION7 serves as a Runtime master station.

STATION9 acts as a file server on the network to store data files. STATION9 is also connected to the HLAN and is connected to a file server using Novell software.

Each station is addressed on the GEN-NET by the network card address on the ARCNET. A station name is linked to a physical address in the GEN-NET software. Refer to Appendix H for details on the network layout.

9.0 REFERENCES

1. Westinghouse Hanford Company, "Test Plan for Run-In of 101-SY Mitigation Mixer Pump," WHC-SD-WM-TP-139.
2. T. M. Burke, "Test Plan for Tank 101-SY Mitigation-by-Mixing Test," Westinghouse Hanford Company, WHC-SD-WM-TP-140.
3. T. R. Benegas, "Tank 241-101-SY Hydrogen Mitigation Test Functional Design Criteria," Westinghouse Hanford Company, WHC-SD-WM-FDC-022 (1992).
4. R. W. Truitt, "Functional Design Criteria for Tank SY-101 Hydrogen Mitigation Test Project Data Acquisition and Control System (DACS-1)," Westinghouse Hanford Company, WHC-SD-WM-FDC-031, Rev. 0 (1994).
5. L. H. Sullivan, et al, "A Safety Assessment for Proposed Mixing Operations to Mitigate Episodic Gas Releases in Tank 21-SY-101, Hanford Site, Richland, WA.", Los Alamos National Laboratory, LA-UR-92-3195, Rev. 14 (1995) or higher.
6. G. J. Gauck, "System Design Description 241-SY-101 DACS Uninterruptible Power Supply", Westinghouse Hanford Company, WHC-SD-WM-SD-076, Rev. 0 (1997).
7. G. J. Gauck, "System Configuration Management Plan for 101-SY Hydrogen Mitigation Test Project Data Acquisition and Control System of Tank Waste Remediation System", Westinghouse Hanford Company, WHC-SD-WM-CSCM-023, Rev. 0 (1994).

Appendix A: Hardware Configurations

The following is a list of all cards and their settings on all computer stations in the DACS trailer. The addresses below ("Addr" column) are in hexadecimal. The printer ports are multiplexed; the port on the multiplexer is noted in the "Destination" column; the ARCNET star configuration ports are also noted here.

There are other standard IRQs that are being used by the system in addition to those noted in the tables: IRQ 0 (system timer); IRQ 1 (keyboard), IRQ 2 (cascade from higher IRQs), IRQ 6 (floppy disk), IRQ 8 (clock), IRQ 13 (80486DX), and IRQ 14 (hard disk).

Table A-1. Computer #1 - Texas Microsystems in the DACS Trailer
(Running Modsoft; Directly Connected to the PLC)

Slot	Card	Port(s)	IRQ	Addr	Destination
1	Kensington bus mouse	9-pin ♂	10	338	Bus mouse
2	(empty)				
3	SA-85	9-pin ♀		C000	Modbus Plus, E984
4	(empty)				
5	Texas Micro	15-pin ♀ 9-pin ♀			V2VGA Video not used
6	(empty)				
7	CPU (80486)	9-pin ♂ 25-pin ♀			not used not used
8	Magic I/O	9-pin ♂ 25-pin ♀	4 7	3F8 378	not used LaserJet printer (port 6)
9	I/O jumper	9-pin ♂ 25-pin ♀	3	2F8	PLC A/B switch (rack 13)* not used
10	(empty)				
11	(empty)				
12	(empty)				
13	SCSI controller	37-pin ♀	11	330	not used
14	(empty)				

* Connected through T-connector going to Computer #4 and to rack 8, where the cable is curled up near Computer #9. The card is not being used.

Table A-2. Computer #2 - Texas Microsystems in the DACS Trailer
(Used for Gas Monitoring)

Slot	Card	Port(s)	IRQ	Addr	Destination
1	Kensington bus mouse	9-pin ♂	10	338	not used
2	(empty)				
3	3Com Ethernet	15-pin ♀ BNC ♀			TLC fanout buffered repeater not used
4	(empty)				
5	Texas Micro	15-pin ♀ 9-pin ♀			V2VGA Video not used
6	(empty)				
7	CPU (80486)	9-pin ♂ 25-pin ♀	4 7	3F8 378	mouse LaserJet printer (port 2)
8	I/O jumper	9-pin	3	2F8	not used
9	(empty)				
10	3Com Ethernet	15-pin ♀ BNC ♀	5	300	not used Rack 10 GMS computer
11	(empty)				
12	(empty)				
13	SCSI controller	37-pin ♀	11	330	not used
14	(empty)				

Table A-3. Computer #3 - Texas Microsystems in the DACS Trailer
(Utility Station; HLAN Drop; Communicates with ASCII/BASIC Module)

Slot	Card	Port(s)	IRQ	Addr	Destination
1	Kensington bus mouse	9-pin ♂	12	338	bus mouse
2	Iomega host adaptor	37-pin ♂	5		Switch box; branch 2 (Bernoulli)
3	3Com Ethernet II	RF connector RF connector	5	300	terminated (jumpered to slot 11) Hanford LAN
4	(empty)				
5	Texas Micro	15-pin ♀ 9-pin ♀			V2VGA Video not used
6	(empty)				
7	CPU (80486)	9-pin ♂ 25-pin ♀			not used not used
8	Magic I/O	9-pin ♂ 25-pin ♀	4 7	3F8 378	not used LaserJet printer (port 4)
9	I/O jumper	9-pin ♂ 25-pin ♀	3	2F8	ASCII module downloading not used
10	(empty)				
11	3Com Ethernet II	15-pin ♀			not used (jumpered to slot 3)
12	(empty)				
13	SCSI controller	37-pin ♀	11	330	not used
14	(empty)				

Table A-4. Computer #4 - Texas Microsystems in the DACS Trailer
 (High Speed Strain Gauge Data [Nicolet System])

Slot	Card	Port(s)	IRQ	Addr	Destination
1	Kensington bus mouse	9-pin ♂	12	338	bus mouse
2	Iomega host adaptor	37-pin ♂	5		Switch box; branch 1 (Bernoulli)
3	GPIB-AT	IEEE-488	10	280	Nicolet system
4	(empty)				
5	Texas Micro	15-pin ♀ 9-pin ♀			V2VGA Video; Shinko multiplx. not used
6	(empty)				
7	CPU (80486)	9-pin ♂ 25-pin ♀			not used not used
8	Magic I/O	9-pin ♂ 25-pin ♀	4 7	3F8 378	not used LaserJet printer (port 3)
9	I/O jumper	9-pin ♂ 25-pin ♀	3	2F8	not used not used
10	(empty)				
11	(empty)				
12	(empty)				
13	SCSI controller	37-pin ♀	11	330	not used
14	(empty)				

Table A-5. Computer #5 - Texas Microsystems in the DACS Trailer
 (STATION5, running the GENESIS TEST strategy)

Slot	Card	Port(s)	IRQ	Addr	Destination
1	(empty)				
2	(empty)				
3	SA-85	9-pin ♀		D800	Modbus Plus, E984
4	(empty)				
5	Texas Micro	15-pin ♀ 9-pin ♀			V2VGA Video; Shinko multiplex. not used
6	(empty)				
7	CPU (80486)	9-pin ♂ 25-pin ♀			not used not used
8	Magic I/O	9-pin ♂ 25-pin ♀	7	378	not used GENESIS C key; Black Box data switch; Proprinter #3 (port 1)
9	I/O jumper	9-pin ♂ 25-pin ♀	4 3	3F8 2F8	Dual ABC box (A) mouse
10	SMC ARCNET	BNC	5	2E0	ARCNET Hub #1, Port 5
11	(empty)				
12	(empty)				
13	(empty)				
14	(empty)				

Table A-6. Computer #6 - Texas Microsystems in the DACS Trailer
 (STATION6, Remote Supervisory Station for the TEST strategy)

Slot	Card	Port(s)	IRQ	Addr	Destination
1	(empty)				
2	(empty)				
3	(empty)				
4	(empty)				
5	Texas Micro	15-pin ♀ 9-pin ♀			V2VGA Video; Shinko multiplex. not used
6	(empty)				
7	CPU (80486)	9-pin ♂ 25-pin ♀	4 7	3F8 378	mouse GENESIS RSS key; Black Box data switch; Proprinter #3 (port 1)
8	(empty)				
9	SMC ARCNET	BNC	5	2E0	ARCNET Hub #1, Port 6
10	(empty)				
11	(empty)				
12	(empty)				
13	(empty)				
14	(empty)				

Table A-7. Computer #7 - Texas Microsystems in the DACS Trailer
 (STATION7, Remote Supervisory Station and Backup for STATION5 or STATION8)

Slot	Card	Port(s)	IRQ	Addr	Destination
1	(empty)				
2	Kensington bus mouse	9-pin ♂	12	338	bus mouse
3	SA-85	9-pin ♀		D800	Modbus Plus, E984
4	(empty)				
5	Texas Micro	15-pin ♀ 9-pin ♀			V2VGA Video; Shinko multiplex. not used
6	(empty)				
7	CPU (80486)	9-pin ♂ 25-pin ♀			not used not used
8	Magic I/O	9-pin ♂ 25-pin ♀	7	378	not used GENESIS RT key; Black Box data switch; Proprinter #3 (port 2)
9	I/O jumper	9-pin ♂ 25-pin ♀	4 3	3F8 2F8	mouse Dual ABC box (C)
10	SMC ARCNET	BNC	5	2E0	ARCNET Hub #1, Port 7
11	(empty)				
12	(empty)				
13	(empty)				
14	(empty)				

Table A-8. Computer #8 - Texas Microsystems in the DACS Trailer
(STATION8, running the GENESIS MOTOR strategy)

Slot	Card	Port(s)	IRQ	Addr	Destination
1	(empty)				
2	Kensington bus mouse	9-pin ♂	12	338	bus mouse
3	SA-85	9-pin ♀		D800	Modbus Plus, E984
4	Everex I/O	9-pin ♂ 25-pin ♀	7	278	not used Proprinter #8
5	Texas Micro	15-pin ♀ 9-pin ♀			V2VGA Video; Shinko multiplx. not used
6	(empty)				
7	CPU (80486)	9-pin ♂ 25-pin ♀			not used not used
8	Magic I/O	9-pin ♂ 25-pin ♀	7	378	not used GENESIS C key; LaserJet printer multiplx. (port 8)
9	I/O jumper	9-pin ♂ 25-pin ♀	4 3	3F8 2F8	Dual ABC box (B) mouse
10	(empty)				
11	SMC ARCNET	BNC	5	2E0	not used
12	(empty)				
13	(empty)				
14	(empty)				

Table A-9. Computer #9 - Texas Microsystems in the DACS Trailer
(STATION9, stores data files and moves them to a HLAN file server)

Slot	Card	Port(s)	IRQ	Addr	Destination
1	Kensington bus mouse	9-pin ♂	10	338	bus mouse
2	(empty)				
3	3Com Ethernet II	RF connector RF connector	5	300	terminated (jumpered to slot 11) Hanford LAN
4	(empty)				
5	Texas Micro	15-pin ♀ 9-pin ♀			V2VGA Video; Shinko multiplex. not used
6	(empty)				
7	CPU (80486)	9-pin ♂ 25-pin ♀	4 7	3F8 378	not used GENESIS RT key
8	I/O jumper	9-pin ♂ 25-pin ♀			not used not used
9	Iomega host adaptor	37-pin ♂	5		Bernoulli drive
10	SMC ARCNET	BNC	3	2E0	ARCNET Hub #1, Port 4
11	3Com Ethernet II	15-pin ♀			not used (jumpered to slot 3)
12	(empty)				
13	(empty)				
14	(empty)				

Table A-10. Computer #10 - Texas Microsystems in the DACS Trailer
 (STATION10, Remote Supervisory Station for the TEST strategy)

Slot	Card	Port(s)	IRQ	Addr	Destination
1	(empty)				
2	(empty)				
3	(empty)				
4	(empty)				
5	Texas Micro	15-pin ♀ 9-pin ♀			V2VGA Video not used
6	(empty)				
7	CPU (80486)	9-pin ♂ 25-pin ♀	4 7	3F8 378	not used GENESIS RSS key
8	(empty)				
9	SMC ARCNET	BNC	5	300	ARCNET Hub #2, Port 1
10	(empty)				
11	(empty)				
12	(empty)				
13	(empty)				
14	(empty)				

Appendix B: Key Macros and State Fields

GENESIS software has several predefined key macros. The TEST and MOTOR strategies utilize a key macro definition file to redefine one of the key macros, the auto/manual function. RSS stations outside the DACS trailer do not have the redefined auto/manual function enabled. The text file EXTRA.KMS was created to disable <F9> and add <Alt-F9> as the auto/manual macro. The text file is compiled with KEYMAC EXTRA to create EXTRA.KML. This compiled file must be contained in the MOTOR and TEST directories where the respective strategies are contained. The following is the listing of EXTRA.KMS:

```
DISMOV1=ON(DISMOVE.OUT).
DISMOV0=OFF(DISMOVE.OUT).
CHKCLR1=ON(CHKCOLOR.OUT).
SETTES1=ON(SETTEST.OUT).
RESPTIM=DOWNLD_PT(0,PBHR.OUT) DOWNLD_PT(0,PBMIN.OUT)
DOWNLD_PT(0,PBSEC.OUT)
DOWNLD_PT(0,DO5DEV.SETP) DOWNLD_PT(0,DO6DEV.SETP).
```

```
[F9]=DISABLE.
[Alt-F9]=A/M().
```

State fields are useful to tie words to analog or discrete values. The text file STATES.SFS was modified to create a state field for the displays linked to the MOTOR strategy:

- <Enable-Disable Move> is used to indicate and control the enabling of the position motor.

If a state field is revised (by editing STATES.SFS and compiling it with the command STATECMP STATES), the display connection in the display builder must be redone in order for the changes to come into effect.

Appendix C: Report and Recipe File Listings

Report and Recipe (R&R) code is controlled by the state of bits in the strategy. These bits are connected to *REPORT* blocks which reference the code as a source file.

R&R code is composed of a command and function set provided by the R&R software option. These commands and functions must follow a syntax format explained in the R&R Option document provided by Iconics.

Following is a description of each R&R code's function for the MOTOR and TEST strategies and when it is executed. Refer to the code itself for details on how the functions are carried out.

MOTOR Strategy Reports

R&R file: CLRMESS.RPS
REPORT block: TESTSET (File 3)
Controlled by: CLRMESS.DOUT Set high within the reports DELETE and SAVE.
Purpose: This clears the message window on the TESTSET screen.
Related task: Test setup.

```
@MSG(UMSG1.DESC," ")  
@EXIT(CURRENT)
```

R&R file: DELETE.RPS
REPORT block: TESTSET (File 2)
Controlled by: DELPUL.DOUT Set high by the operator when deleting a test setup.
Purpose: This deletes a test setup from USERSET.TDF, the test setup file.
Related task: Test setup.


```

        @out(f2,f1[i,0]:79)
    @endif
    @set(i,i+1)
    @set(testno,f1[i,0]:4)
@endloop
@if(flg==0)
    @msg(UMSG1.desc,"Test not found.")
@endif
@out(f2," 999",nl)
@close(f1)
@close(f2)
@wait(20)
@COPY("newset.tdf","userset.tdf")
@set(INFILE.out,0)
@else
    @msg(UMSG1.desc,"Test file not found.")
@endif
@set(c1rmess.out,1)
@exit(current)

```

R&R file: ILOAD.RPS

REPORT block: TESTSET2 (File 4)

Controlled by: ON.DOUT Executes at startup only.

Purpose: This initializes the test setup blocks.

Related task: Test setup.

```

@FLOAT(testno,i)
@set(OLDTEST.out,UTESTNO.out)
@if(access("userset.tdf")==1)
    @open(f1,"userset.tdf",ASCII,INPUT,80)
    @set(testno,f1[0,0]:4)
    @if(testno<>999)
        @set(i,0)
        @set(UTESTNO.out,f1[i,0]:4)
        @set(UANGLE.out,f1[i,5]:5)
        @set(USPEED.out,f1[i,11]:6)
        @set(UHRS.out,f1[i,19]:2)
        @set(UMINS.out,f1[i,23]:2)
        @set(USECS.out,f1[i,27]:2)
        @set(UACCEL.out,f1[i,30]:5)
        @set(UDECEL.out,f1[i,36]:5)
        @msg(UDESC.desc,f1[i,42]:21)
        @set(URESTIM.out,f1[i,75]:1)
        @set(INFILE.out,1)
        @set(NEXTINDX.out,i)
        @loop((testno<>999) and (not(eof(f1))))
            @set(i,i+1)
            @set(testno,f1[i,0]:4)

```

```

@endloop
@set(MAXINDX.out,i)
@load("setold")
@close(f1)
@else
@set(MAXINDX.out,0)
@set(NEXTINDX.out,0)
@set(INFILE.out,0)
@close(f1)
@endif
@else
@open(f1,"userset.tdf",ASCII,OUTPUT,80)
@out(f1," 999",nl)
@close(f1)
@set(MAXINDX.out,0)
@set(NEXTINDX.out,0)
@set(INFILE.out,0)
@endif
@exit(current)

```

R&R file: INITENAB.RPS

REPORT block: INITENAB (File 1)

Controlled by: ANTON.DOUT which goes high 10 seconds after the start of Runtime.

Purpose: This code is used to initialize the registers in the PLC when the strategy is first started.

Related task: Initialization.

```

; Code: "INITENAB.RPS"
;
; Purpose: This code is used to initialize the registers in the PLC when
; the strategy is first started.
;
; Related task: Initialization
;
; Author: Ken Eldridge 6/23/95
; Modifications: Sam Smith 6/26/95
; To eliminate multiple register setting in the same PAOUT
; block.
; To merge setting values to 1 and setting block back to
; auto into one RRcode source file.
;
; This code is executed by:
; ANTON.DOUT which goes high 10 seconds after the start of Runtime.
;
; Reference: INITENAB.SOURCE_FILE1

```

```

@SET(I2C1.AM,1)      ; Places block in manual.
@SET(I2C2.AM,1)      ; Places block in manual.
@SET(I2C1.P06,0.0015) ; Sets value to 1, never really gets to PLC.
@SET(I2C2.P06,0.0015) ; Sets value to 1, never really gets to PLC.

@SET(I2C1.AM,0)      ; Puts block back to automatic and all values written
                    ; at this time.
@SET(I2C2.AM,0)      ; Same as above.

```

R&R file: NEXT.RPS

REPORT block: TESTSET2 (File 2)

Controlled by: NEXTPUL.DOUT Set high by the operator when using the down arrow button to access the next test in the test setup file.

Purpose: This accesses the next test in the test setup file.

Related task: Test setup.

```

; Code: "NEXT.RPS"
;
; Purpose: This code uses the file index block, NEXTINDX, to load
;          test setup values for the next test in the file
;          USERSET.TDF.
;
; Related task: Test Setup
;
; Author: Jeff Martin 6/7/94
; Modifications:
;
; This code is executed by:
;          NEXTPUL.DOUT is set high by the operator selecting NEXT
;          in the TESTSET display or by selecting the up-arrow
;          in the PUMPOPS or PBTESTS displays.
;
; Reference: TESTSET???.SOURCE_FILE1

```

```

@FLOAT(testno,i) ;setup local variables. testno for the test number, i for the file
index
@if(MAXINDX.out == 0) ;are there any records in the file?
  @exit(current) ;if not, exit
@endif
@if(access("userset.tdf")==1) ;does the file exist?
  @open(f1,"userset.tdf",ASCII,INPUT,80) ;yes. open it.
  @set(i,NEXTINDX.out+1) ;increment the file index pointer
  @set(testno,f1[i,0]:4) ;read the test number
  @if(testno <> 999) ;are we at the end of file?

```

```

@set(UTESTNO.out,testno)           ;no. set the test parameter blocks
@set(UANGLE.out,f1[i,5]:5)
@set(USPEED.out,f1[i,11]:6)
@set(UHRS.out,f1[i,19]:2)
@set(UMINS.out,f1[i,23]:2)
@set(USECS.out,f1[i,27]:2)
@set(UACCEL.out,f1[i,30]:5)
@set(UDECEL.out,f1[i,36]:5)
@msg(UDESC.desc,f1[i,42]:21)
@set(URESTIM.out,f1[i,75]:1)
@set(INFILE.out,1)                 ;set in file flag
@set(NEXTINDX.out,i)               ;set new value of index
@load("setold")                   ;queue setold.rps
@close(f1)                         ;close and exit.
@exit(current)

@else
@set(i,0)                          ;we were at the end so wrap to the beginning of the file
@set(testno,f1[i,0]:4)             ;get test number
@if(testno <> 999)                  ;still at end of file?
  @set(UTESTNO.out,testno)         ;no. set the test parameter blocks.
  @set(UANGLE.out,f1[i,5]:5)
  @set(USPEED.out,f1[i,11]:6)
  @set(UHRS.out,f1[i,19]:2)
  @set(UMINS.out,f1[i,23]:2)
  @set(USECS.out,f1[i,27]:2)
  @set(UACCEL.out,f1[i,30]:5)
  @set(UDECEL.out,f1[i,36]:5)
  @msg(UDESC.desc,f1[i,42]:21)
  @set(URESTIM.out,f1[i,75]:1)
  @set(INFILE.out,1)               ;set the in-file flag
  @set(NEXTINDX.out,i)             ;set the index pointer
  @load("setold")                 ;queue setold.rps
  @close(f1)                       ;close and exit.
  @exit(current)
@endif
@endif
@exit(current)

```

R&R file: PBSETVAL.RPS
 REPORT block: PBREPORT (File 1)
 Controlled by: SETTEST.DOUT Set high by the operator selecting to set values.
 Purpose: This code will download the desired test values to the critical parameters in the MOTOR.
 Related task: Set test values

```
Code: "PBSETVAL.RPS"

Purpose: This code is used to update setpoints, alarm limits, and abort
limits based on test selection.

Related task: Test Operation

Author: Darrel Holt 8/3/93
Modifications: Ken Eldridge 1/23/94
              To provide additional comments to code.
              S.G. McNeece 3/08/94
              CR304 to remove duplicate strategy blocks

              S.G. McNeece. 4/21/94
              CR334 to remove demo mode
              Ken Eldridge 5/25/94
              Integrate status indication to strategy
              Ken Eldridge 6/6/94
              Incorporate setting of alarm setpoints for feedback check.
              Jeff Martin 6/7/94
              Values for each test are now maintained in a file (userset.tdf)
              and loaded into algorithm blocks. The actual test settings
              are loaded by this report from these blocks.
              Jeff Martin 7/28/94
              Add position deadband sets.

This code is executed by:
PBSETAND.DOUT is set high by the operator selecting to set values
in the phase B test display.

Reference: PBREPORT.SOURCE_FILE1
```

```
                : REMOVE THIS WHEN PUMP BUMP SCREEN GOES AWAY
@IF(BUMPENAB.OUT == 1)                : Is this a pump bump or a phase b test?
@SET(D03DEV.SETP.PMOTOR.ACCE)         : Pump acceleration check for PBTESTS indication.
@SET(D04DEV.SETP.PMOTOR.DECE)         : Pump deceleration check for PBTESTS indication.
@SET(RESETPB.OUT,1)                   : Reset overall test time.
@SET(D01DEV.SETP.POSCNTNL.SETP)       : Position check for PUMP BUMP indication.
@SET(PMPSPDAL.OUT.PMOTOR.SPDS + 10)   : Set pump motor alarm limit.
@SET(PMPSPDAB.OUT.PMOTOR.SPDS + 20)   : Set pump motor abort limit.
@SET(D02DEV.SETP.PMOTOR.SPDS)         : Desired speed check for PBTESTS indication.
@SET(D05DEV.SETP.PMOTOR.SPDS)         : Pump maximum speed check for PBTESTS indication.
```

```

@SET(D06DEV.SETP.PMOTOR.SPDS+10) ; Pump speed alarm check for PBTESTS indication.
@SET(D07DEV.SETP.PMOTOR.SPDS+20) ; Pump speed abort check for PBTESTS indication.

@ELSE

@SET(DMOTOR.SPDS.100) ; Set desired speed of position motor.
@SET(DMOTOR.ACCE.500) ; Set acceleration of position motor.
@SET(DMOTOR.DECE.500) ; Set deceleration of position motor.
@SET(PMOTOR.ACCE.UACCEL.OUT) ; Set acceleration of pump motor.
@SET(PMOTOR.DECE.UDECEL.OUT) ; Set deceleration of pump motor.

@SET(D03DEV.SETP.UACCEL.OUT) ; Pump acceleration check for PBTESTS indication.
@SET(D04DEV.SETP.UDECEL.OUT) ; Pump deceleration check for PBTESTS indication.
@MSG(TDESC.DESC,UDESC.DESC) ; Test description
@IF(URESTIM.OUT == 1) ; Check if overall test time should be reset.
    @SET(RESETPB.OUT.1) ; Reset overall test time.
@ENDIF

@SET(POSCNTRL.HGAP.2.2) ; Set pump position deadband (high)
@SET(POSCNTRL.LGAP.2.2) ; Set pump position deadband (low)
@SET(POSCNTRL.SETP.UANGLE.OUT) ; Set desired pump position.
@SET(PBMIN.OUT.UMINS.OUT) ; Set desired minutes for pump operation.
@SET(PBSEC.OUT.USECS.OUT) ; Set desired seconds for pump operation.
@SET(PBHR.OUT.UHRS.OUT) ; Set desired hours for test operation.

@SET(D01DEV.SETP.UANGLE.OUT) ; Position check for PBTEST indication.
@SET(PMOTOR.MAXS.USPEED.OUT) ; Set pump motor maximum speed.
@WAIT(10) ; 1 second wait needed for maximum speed to
    ; be set first before desired speed.
@SET(PMOTOR.SPDS.USPEED.OUT) ; Set pump motor desired speed.
@SET(PMPSPDAL.OUT.USPEED.OUT+10) ; Set pump motor alarm limit.
@SET(PMPSPDAB.OUT.USPEED.OUT+20) ; Set pump motor abort limit.

@SET(D02DEV.SETP.USPEED.OUT) ; Desired speed check for PBTESTS indication.
@SET(D05DEV.SETP.USPEED.OUT) ; Pump maximum speed check for PBTESTS indication.
@SET(D06DEV.SETP.USPEED.OUT+10) ; Pump speed alarm check for PBTESTS indication.
@SET(D07DEV.SETP.USPEED.OUT+20) ; Pump speed abort check for PBTESTS indication.

@if((INFILE.OUT == 0) or (CHANGED.OUT == 1)) ;check for test in file or changed.
@SET(MANUAL.OUT.1) ;test changed or not in file, indicate manual test.
@SET(PBTESTNO.OUT.0) ;and clear test number.
@ELSE
    @SET(MANUAL.OUT.0) ;Indicate valid test in file.
    @SET(PBTESTNO.OUT.UTESTNO.OUT) ;Set current test number
@ENDIF

@ENDIF

@SET(SETVALDN.OUT.1) ; Set indication that values are set successfully.

```

R&R file: PREV.RPS
 REPORT block: TESTSET2 (File 3)
 Controlled by: PREVPUL.DOUT Set high by the operator when using the up arrow button to access the previous test in the test setup file.
 Purpose: This accesses the previous test in the test setup file.
 Related task: Test setup.

```

: Code: "PREV.RPS"
: Purpose: This code uses the file index block, NEXTINDX, to load
: test setup values for the previous test in the file
: USERSET.TDF.
:
: Related task: Test Setup
:
: Author: Jeff Martin 6/7/94
: Modifications:
:
: This code is executed by:
: PREVPUL.DOUT is set high by the operator selecting PREVIOUS
: in the TESTSET display or by selecting the down-arrow
: in the PUMPOPS or PBTESTS displays.
:
: Reference: TESTSET???.SOURCE_FILE1

```

```

@FLOAT(testno,i) ;setup local variable, testnumber and i - record index.
@if(MAXINDX.out == 0) ;if maxindx is 0, there is either no setup file
;or a setup file with no tests in it. So exit.

@exit(current)
@endif
@if(access("userset.tdf")==1)
@open(f1,"userset.tdf",ASCII,INPUT,80) ;if test setup file exists, open it.
@set(i,NEXTINDX.out) ;get the index for the current test.
@if(i==0) ;are we at the beginning?
@set(i,MAXINDX.out) ;if so, set the new index to the end of the file
if(i<>0)
@set(i,i-1)
@endif
@else
@set(i,NEXTINDX.out-1) ;we're not at the beginning, so decrement index pointer
@endif
@set(testno,f1[i,0]:4) ;read the next number for the 'previous' record.
@if(testno <> 999) ;is it the end of file pointer?
@set(UTESTNO.out,testno) ;if not, set the test value blocks in the strategy
@set(UANGLE.out,f1[i,5]:5)
@set(USPEED.out,f1[i,11]:6)

```

```

@set(UHRS.out,f1[i,19]:2)
@set(UMINS.out,f1[i,23]:2)
@set(USECS.out,f1[i,27]:2)
@set(UACCEL.out,f1[i,30]:5)
@set(UDESC.out,f1[i,36]:5)
@msg(UDESC.desc,f1[i,42]:21)
@set(URESTIM.out,f1[i,75]:1)
@set(INFILE.out,1) ;set the in-file indication
@set(NEXTINDX.out,i) ;set the new current record index
@load("setold") ;queue setold.rps
@close(f1) ;close and exit
@exit(current)
@endif
@endif
@exit(current)

```

R&R file: SAVE.RPS

REPORT block: TESTSET (File 1)

Controlled by: SAVEPUL.DOUT Set high by the operator to save a test setup.

Purpose: This saves a test setup in the test setup file.

Related task: Test setup.

```

; Code: "SAVE.RPS"
;
; Purpose: This code inserts the current user defined test setup
; parameters into the test setup file, USERSET.TDF.
;
; Related task: Test Setup
;
; Author: Jeff Martin 6/7/94
; Modifications:
;
; This code is executed by:
; SAVEPUL.DOUT is set high by the operator selecting SAVE TEST
; in the TESTSET display.
;
; Reference: TESTSET???.SOURCE_FILE1

```

```

@FLOAT(testno,i) ;setup local variables, testno is the test number read from
; the file. i is the record index into the file
@DIGITAL(flg) ;setup local variable, flg is set when the new test setup
; has been inserted.
@msg(umsg1.desc,"Saving...") ;write message to display
@if(access("userset.tdf")==1) ;check to see if USERSET.TDF exists

```

```

@open(f1,"userset.tdf",ASCII,INPUT,80);USERSET exists so open it.
@open(f2,"newset.tdf",ASCII,OUTPUT,80);open NEWSET.TDF
@set(i,0);initialize flg and record pointer
@set(fl,0)
@set(testno,f1[i,0]:4);read the first test number from file
@loop((not(eof(f1))) and (testno < 999));loop until end of file is reached
  @if(flg==0);have we inserted the new test setup yet?
    ;if not, check to see if this is the place (the file is
    ;maintained so that the tests are in ascending numeric order).
    @if(((testno-0.05) < UTESTNO.out) and ((testno+0.05) > UTESTNO.out))
      ;new test number is equal to a test number already in the file
      ;so replace the old setup with the new one.
      @msg(UMSG1.desc,"Replacing previous test setup.")
      @wait(10)
      @out(f2,UTESTNO.out:2:1," ",UANGLE.out:3:1," ",USPEED.out:4:1)
      @out(f2," ",UHRS.out:2:0," ",UMINS.out:2:0," ",USECS.out:2:0)
      @out(f2," ",UACCEL.out:3:1," ",UDECEL.out:3:1)
      @out(f2," ",UDESC.desc," ",tab(80*i+75),URESTIM.out)
      @out(f2," ",nl)
      @set(NEXTINDX.out,i);set index variable to point to the new test setup
      @set(fl,1);set flag indicating we have inserted new test setup.
    @else
      @if(UTESTNO.out < testno-0.05);are we in the right spot?
        ;yes. So insert new test setup. We didn't find a test with the same
        ;number already in the file.
        @out(f2,UTESTNO.out:2:1," ",UANGLE.out:3:1," ",USPEED.out:4:1)
        @out(f2," ",UHRS.out:2:0," ",UMINS.out:2:0," ",USECS.out:2:0)
        @out(f2," ",UACCEL.out:3:1," ",UDECEL.out:3:1)
        @out(f2," ",UDESC.desc," ",tab(80*i+75),URESTIM.out)
        @out(f2," ",nl)
        @set(fl,1);Set flag indicating we have inserted the new test.
        @set(NEXTINDX.out,i);set index variable to point to the new test.
        @set(MAXINDX.out,MAXINDX.out+1);increment the maximum test pointer since
        ;we have added a new one.
        @out(f2,f1[i,0]:80);now copy the current test setup in the old file to
        ;the new one.
      @else
        @out(f2,f1[i,0]:80);not the right place to insert the new test setup.
        ;copy test setup from the old file to the new one.
      @endif
    @endif
  @endif
  @endif
  ;we have already inserted the new test set up so...
  @out(f2,f1[i,0]:80);copy the remainder of the records from
  ;the old file to the new file.
  @endif
  @set(i,i+1);increment index pointer
  @set(testno,f1[i,0]:4);read next test number
@endloop
@if(flg==0)

```

```

;we've reached the end of file and still not inserted the new
;test. That means we need to insert it at the end.
@out(f2,UATESTNO.out:2:1," ",UANGLE.out:3:1," ",USPEED.out:4:1)
@out(f2," ",UHRS.out:2:0," ",UMINS.out:2:0," ",USECS.out:2:0)
@out(f2," ",UACCEL.out:3:1," ",UDECEL.out:3:1)
@out(f2," ",UDESC.desc," ",tab(80*i+75),URESTIM.out)
@out(f2," ",nl)
@set(NEXTINDX.out,i) ;set index pointer variable to new test
@set(MAXINDX.out,MAXINDX.out+1) ;increment maximum test index pointer
@endif
@out(f2," 999",nl) ;append the end of file marker to the new test file
@close(f1) ;close the old setup file and the new setup file
@close(f2)
@wait(20) ;wait 2 seconds then copy the new file to the old file.
@COPY("newset.tdf","userset.tdf")
@set(INFILE.out,1) ;set the infile flag indicating that the current test
;is in the setup file.
@load("setold") ;queue setold.rps
@else
;the file didn't exist. so create it and add the current setup.
@open(f2,"userset.tdf",ASCII,OUTPUT,80)
@out(f2,UATESTNO.out:2:1," ",UANGLE.out:3:1," ",USPEED.out:4:1)
@out(f2," ",UHRS.out:2:0," ",UMINS.out:2:0," ",USECS.out:2:0)
@out(f2," ",UACCEL.out:3:1," ",UDECEL.out:3:1)
@out(f2," ",UDESC.desc," ",tab(75),URESTIM.out)
@out(f2," ",nl)
@out(f2," 999",nl)
@set(MAXINDX.out,1) ;set the maximum index to 1
@set(NEXTINDX.out,0) ;set the current index to 0 - the first record.
@close(f2)
@set(INFILE.out,1) ;the current test is in the file.
@load("setold") ;queue setold.rps
@endif
@msg(umsg1.desc,"Save Complete.") ;write message to display.
@set(clrmess.out,1) ;run clrmess.rps after a delay.
@exit(current) ;exit

```

R&R file: SETOLD.RPS

REPORT block: None

Controlled by: Called from the reports SAVE, ILOAD, ULOAD, NEXT, PREV.

Purpose: This code updates the old test parameter blocks whenever a new test becomes the active test. This allows the check for test parameter changes to be made.

Related task: Test setup.

@SET(OANGLE.OUT,UANGLE.OUT)

```

@SET(OSPEED.OUT,USPEED.OUT)
@SET(OHRS.OUT,UHRS.OUT)
@SET(OMINS.OUT,UMINS.OUT)
@SET(OSECS.OUT,USECS.OUT)
@SET(OACCEL.OUT,UACCEL.OUT)
@SET(ODECEL.OUT,UDECEL.OUT)
@SET(ORESTIM.OUT,URESTIM.OUT)

```

R&R file: ULOAD.RPS

REPORT block: TESTSET2 (File 1)

Controlled by: TSTCHG.DOUT Set whenever a new test number is entered by the operator on the TESTSET, PBTESTS or PUMPRUN screens.

Purpose: This loads the selected test from the test setup file..

Related task: Test setup.

```

@FLOAT(testno,i)
@set(OLDTEST.out,UTESTNO.out)
@if(access("userset.tdf"==1))
  @open(f1,"userset.tdf",ASCII,INPUT,80)
  @set(i,0)
  @set(testno,f1[i,0]:4)
  @loop(((testno<>999) and (not(eof(f1))))
    @if(((testno-0.05) < UTESTNO.out) and ((testno+0.05) > UTESTNO.out))
      @set(UANGLE.out,f1[i,5]:5)
      @set(USPEED.out,f1[i,11]:6)
      @set(UHRS.out,f1[i,19]:2)
      @set(UMINS.out,f1[i,23]:2)
      @set(USECS.out,f1[i,27]:2)
      @set(UACCEL.out,f1[i,30]:5)
      @set(UDECEL.out,f1[i,36]:5)
      @msg(UDESC.desc,f1[i,42]:21)
      @set(URESTIM.out,f1[i,75]:1)
      @set(INFILE.out,1)
      @set(NEXTINDX.out,i)
    @close(f1)
    @load("setold")
    @exit(current)
  @else
    @if(UTESTNO.out < testno-0.05)
      @set(INFILE.out,0)
      @close(f1)
      @exit(current)
    @endif
  @endif
  @set(i,i+1)
  @set(testno,f1[i,0]:4)

```

```

@endloop
@set(INFILE.out,0)
@set(NEXTINDX.out,0)
@close(f1)
exit(current)
@else
@open(f1,"userset.tdf",ASCII,OUTPUT,80)
@out(f1," 999",n1)
@close(f1)
@set(MAXINDX.out,0)
@set(NEXTINDX.out,0)
@set(INFILE.out,0)
@endif
@exit(current)

```

TEST Strategy Reports

R&R file: HISTCOPY.RPS

REPORT block: FILENAME (File 4)

Controlled by: DIMIN.DOUT Set high one minute after the beginning of a history file changeover.

Purpose: This copies history files from the TEST directory to the HIST and HIST2 directories.

Related task: History file transfer.

```

;Filename: HISTCOPY.RPS

```

```

;Purpose: Copies the history files to HIST and HIST2
;         directories and then deletes them from the test
;         directory.
;         Copies "NEW72" file to "OLD72" file and then
;         deletes "NEW72" so can start another set

```

```
@STRING(SNAM[4])
```

```
@IF(ACCESS("C:\TEST\OLDFN.DT")==1)
```

```
  @OPEN(F1,"OLDFN.DT",FREE,INPUT,0)
```

```
  @SET(SNAM,F1[ ])

```

```
  @COPY("C:\TEST\" + SNAM + "PUMP.PRN","C:\HIST\" + SNAM + "PUMP.PRN")
```

```
  @COPY("C:\TEST\" + SNAM + "RGA5.PRN","C:\HIST\" + SNAM + "RGA5.PRN")
```

```
  @COPY("C:\TEST\" + SNAM + "MIT2.PRN","C:\HIST\" + SNAM + "MIT2.PRN")
```

```
  @COPY("C:\TEST\" + SNAM + "MIT1.PRN","C:\HIST\" + SNAM + "MIT1.PRN")
```

```
  @COPY("C:\TEST\" + SNAM + "TBS1.PRN","C:\HIST\" + SNAM + "TBS1.PRN")

```

```
@COPY("C:\TEST\" + SNAM + "VOL.PRN", "C:\HIST\" + SNAM + "VOL.PRN")
@COPY("C:\TEST\" + SNAM + "TBS2.PRN", "C:\HIST\" + SNAM + "TBS2.PRN")
@COPY("C:\TEST\" + SNAM + "GAS.PRN", "C:\HIST\" + SNAM + "GAS.PRN")
@COPY("C:\TEST\" + SNAM + "MIT3.PRN", "C:\HIST\" + SNAM + "MIT3.PRN")
@COPY("C:\TEST\" + SNAM + "STRN.PRN", "C:\HIST\" + SNAM + "STRN.PRN")
@COPY("C:\TEST\" + SNAM + "PUMP.PRN", "C:\HIST2\" + SNAM + "PUMP.PRN")
@COPY("C:\TEST\" + SNAM + "RGAS.PRN", "C:\HIST2\" + SNAM + "RGAS.PRN")
@COPY("C:\TEST\" + SNAM + "MIT2.PRN", "C:\HIST2\" + SNAM + "MIT2.PRN")
@COPY("C:\TEST\" + SNAM + "MIT1.PRN", "C:\HIST2\" + SNAM + "MIT1.PRN")
@COPY("C:\TEST\" + SNAM + "TBS1.PRN", "C:\HIST2\" + SNAM + "TBS1.PRN")
@COPY("C:\TEST\" + SNAM + "VOL.PRN", "C:\HIST2\" + SNAM + "VOL.PRN")
@COPY("C:\TEST\" + SNAM + "TBS2.PRN", "C:\HIST2\" + SNAM + "TBS2.PRN")
@COPY("C:\TEST\" + SNAM + "GAS.PRN", "C:\HIST2\" + SNAM + "GAS.PRN")
@COPY("C:\TEST\" + SNAM + "MIT3.PRN", "C:\HIST2\" + SNAM + "MIT3.PRN")
@COPY("C:\TEST\" + SNAM + "STRN.PRN", "C:\HIST2\" + SNAM + "STRN.PRN")
@DELETE("C:\TEST\" + SNAM + "PUMP.PRN")
@DELETE("C:\TEST\" + SNAM + "RGAS.PRN")
@DELETE("C:\TEST\" + SNAM + "MIT2.PRN")
@DELETE("C:\TEST\" + SNAM + "MIT1.PRN")
@DELETE("C:\TEST\" + SNAM + "TBS1.PRN")
@DELETE("C:\TEST\" + SNAM + "VOL.PRN")
@DELETE("C:\TEST\" + SNAM + "TBS2.PRN")
@DELETE("C:\TEST\" + SNAM + "GAS.PRN")
@DELETE("C:\TEST\" + SNAM + "MIT3.PRN")
@DELETE("C:\TEST\" + SNAM + "STRN.PRN")
@CLOSE(F1)
@ENDIF

@IF(ACCESS("C:\TEST\FN.DT")==1)
  @COPY("C:\TEST\FN.DT", "C:\TEST\OLDFN.DT")
@ENDIF

@IF(CHNG720K.OUT == 1)
  @IF(ACCESS("NEW72.PRN")==1)
    @COPY("NEW72.PRN", "OLD72.PRN")
    @DELETE("NEW72.PRN")
  @ENDIF
@ENDIF
@ENDIF
```

R&R file: HISTFNAM.RPS
 REPORT block: FILENAME (File 1,2 and 3)
 Controlled by: #435.DOUT, #474.DOUT, PULON.DOUT Set high at two-hour history
 file changeover or at startup.
 Purpose: This renames the history files based upon the time and date.
 Related task: History file transfer.

```

@FLOAT(MNUM,DNUM,HNUM)
@STRING(YSTR[1],MSTR[1],DSTR[1],HSTR[1])
@OPEN(F1,"STORE.DT",ASCII,OUTPUT,8)
@OUT(F1,DATE(YY),DATE(MM),DATE(DD),DATE(HH))
@CLOSE(F1)
@OPEN(F1,"STORE.DT",ASCII,INPUT,8)
@SET(YSTR,F1[0,1]:1)
@SET(MNUM,F1[0,2]:2)
@SET(DNUM,F1[0,4]:2)
@SET(HNUM,F1[0,6]:2)
@CLOSE(F1)
@IF(MNUM > 9)
    @SET(MSTR,CHAR(MNUM + 55))
@ELSE
    @SET(MSTR,CHAR(MNUM + 48))
@ENDIF

@IF(DNUM > 9)
    @SET(DSTR,CHAR(DNUM + 55))
@ELSE
    @SET(DSTR,CHAR(DNUM + 48))
@ENDIF

@IF(HNUM > 9)
    @SET(HSTR,CHAR(HNUM + 55))
@ELSE
    @SET(HSTR,CHAR(HNUM + 48))
@ENDIF

@SET(PUMP.FNAM,YSTR + MSTR + DSTR + HSTR + "PUMP")
@SET(RGAS.FNAM,YSTR + MSTR + DSTR + HSTR + "RGAS")
@SET(MIT2.FNAM,YSTR + MSTR + DSTR + HSTR + "MIT2")
@SET(MIT1.FNAM,YSTR + MSTR + DSTR + HSTR + "MIT1")
@SET(TBS1.FNAM,YSTR + MSTR + DSTR + HSTR + "TBS1")
@SET(VOL.FNAM,YSTR + MSTR + DSTR + HSTR + "VOL")
@SET(TBS2.FNAM,YSTR + MSTR + DSTR + HSTR + "TBS2")
@SET(GAS.FNAM,YSTR + MSTR + DSTR + HSTR + "GAS")
@SET(MIT3.FNAM,YSTR + MSTR + DSTR + HSTR + "MIT3")
@SET(STRN.FNAM,YSTR + MSTR + DSTR + HSTR + "STRN")
@SET(GO.OUT,1)
@OPEN(F1,"FN.DT",FREE,OUTPUT,0)
@OUT(F1,YSTR + MSTR + DSTR + HSTR)
@CLOSE(F1)

```

R&R file: SETLIMS.RPS
 REPORT block: DOWNLIMS (File 1)
 Controlled by: DELAYON.DOUT Set 45 seconds after startup.
 Purpose: Downloads abort limits to the PLC.
 Related task: Abort limits.

```

: Code: "SETLIMS.RPS"
: Purpose: This code is executed to set the abort limits in the PLC.
: Related task: Abort.
: Author: Jeff Martin
: Modifications: Sam Smith/Ross Truitt 04-03-96
: Changes in support of TEST version 3.04 per SCR#384
: Augie Ermi 08-20-97
: Changes in support of TEST version 3.05 per SCR#426
: This code is executed by:
: DELAYON.DOUT
: Reference: DOWNLIMS.SOURCE_FILE1

```

;SETS LIMITS IN AIN BLOCKS FOR OUTPUT THROUGH THE PAOT

```

:
: @SET(HILIM.EVAL,4095)
:
: @SET(HH2LIM.EVAL,0.75)
: @SET(HTEMPLIM.EVAL,135)
: @SET(HPDPLIM.EVAL,21)
: @SET(HPMCLIM.EVAL,140)
: @SET(HPMOTLIM.EVAL,225)
: @SET(LVFLIM.EVAL,400)
: @SET(HTDPLIM.EVAL,-1)
: @SET(HPCSLIM.EVAL,194)
: @SET(HMITSLIM.EVAL,357) Removed per SCR#384
: @SET(H17CSLIM.EVAL,357)
: @SET(HSPDLIM.EVAL,400)
: @SET(HH2LIM2.EVAL,0.75)
: @SET(HTMPLIM3.EVAL,135) Removed per SCR#426
: @SET(LVFLIM2.EVAL,400)
: @SET(HTDPLIM2.EVAL,-1)
: @SET(H1BCSLIM.EVAL,546)
: @SET(LPCSLIM.EVAL,-194)
: @SET(LMITSLIM.EVAL,-357) Removed per SCR#384
: @SET(L17CSLIM.EVAL,-357)
: @SET(L1BCSLIM.EVAL,-546)

```

@SET(LPCGPLIM.EVAL,7)
@SET(HVFLIM1.EVAL,700)
@SET(HVFLIM2.EVAL,700)
@SET(H1BSAL.EVAL,317)
@SET(L1BSAL.EVAL,-317)
@SET(H12ASAL.EVAL,145.5)
@SET(L12ASAL.EVAL,-145.5)
@SET(H17BCSAL.EVAL,267)
@SET(L17BCSAL.EVAL,-267)

Appendix D: PLC Register List

The Modsoft program allows a symbol table to be created. This symbol table allows the programmer to assign symbolic names and comments to the registers used by the PLC. The symbol table is accessed under the Utility Menu of Modsoft. A hard copy print-out can be obtained by using the File I/O and Export menu choices once the symbol table has been called up.

The symbol table is essential for the ladder logic programmer. It is the definitive record of which PLC registers have been used and which are available for use. Any entry in the symbol table which has a symbolic name has been used by the PLC. If there is no entry in the symbol table for a register, or if there is a register entry without a symbolic name, that register is available for use.

It is essential that the symbol table be kept up to date. Any change which adds or deletes PLC registers (including adding I/O modules) must be reflected in the symbol table also.

Appendix E: Input/Output Channel List

Descriptions of the Column Headings Used in the DACS-1 I/O Channel List

COLUMN	DESCRIPTION
DACS TAG	The descriptive name used by the DACS to identify or refer to a specific block such as an input from the field.
TAG DESCRIPTION	The tag description found in the GENESIS strategy block - describes what the tag is (e.g. TANK TEMP RISER 17C - 28").
FIELD TAG	Identifies the tag number assigned to the instrument signal on P&ID drawings.
I/O # AND/OR SOURCE	Identifies the origin of the tag (e.g. the I/O number and location, the PLC, DACS station/block, or other software).
SIGNAL TYPE	Identifies the type of signal that is received from the field instrument (e.g. 4-20 mA).
I/O LOC.	Identifies the I/O location of where the instrument is wired into the Modicon. The four numbers refer to the drop, rack, slot, and channel.
INPUT REG. #	Identifies the Modicon register number used for field I/O.
XFR REG. #	Identifies the Modicon register number used for communication between the PLC and GENESIS.
GENESIS GROUP	Identifies the location of the tag within the GENESIS strategy. The four numbers refer to the device block number, the group within the device block, the connection point within the group, and if the tag is part of a packed block, the channel number within the packed block.
SCAN TIME	Lists the GENESIS scan rates. Each DACS tag will update only as fast as its scan rate.
RANGE: LOW / HIGH	Identifies the instrument range used by the DACS (e.g. 80 / 140)
ENG. UNITS	Identifies the engineering units associated with the range, alarm, and abort values (e.g. °F)
ALARM: LOW / HIGH	Lists the low range and high range alarm setpoints within GENESIS.
ABORT	Lists the low and/or high abort setpoints. If these values are encountered, the PLC will automatically stop the pump.

DACS-1 INPUT/OUTPUT CHANNEL LIST

Sorted By DACS TAG

File "lot5dc45.xls" Page 4 of 7 October 1, 1997

DACS TAG	TAG DESCRIPTION	FIELD TAG	NOT AND/OR	SIGNAL TYPE	LOC.	IO	INPUT REG.#	XFR REG.#	GENESIS GROUP	SCAN TIME	RANGE LOW/HIGH	ENG. UNITS	ALARM LOW	ALARM HIGH	ABORT
RG-RUN	RGAS RUN NUMBER	---	---	ASCII	14.2.7.ASC	---	402137	H2C1.1	30 sec	---	---	---	---	---	---
RG-STAT	RGAS-5 STAT (0-TANK,1-CAL,2-F,3-SH)	---	---	ASCII	14.2.7.ASC	---	402138	H2C1.2	30 sec	---	---	---	---	---	---
RG-TIME	RGAS-5 TIME OF SAMPLE	---	---	ASCII	14.2.7.ASC	---	402139	H2C1.3	30 sec	---	---	---	---	---	---
RGAS1ND1	RGAS TREND OUTPUT 1	---	---	0-10V	61.6.1	---	402065	G1C7.7	30sec	---	---	---	---	---	---
RGAS1ND2	RGAS TREND OUTPUT 2	---	---	0-10V	61.6.2	---	402066	G1C7.8	30sec	---	---	---	---	---	---
RSTCOIL	ABORT COIL RESET	---	---	DOUT	---	---	300058	402064	G1C7.8	30sec	---	---	---	---	---
STCOOPFA	STATION 5 COMM FAIL	---	---	PLC	14.2.4.4	---	20	3048	G2C2.16	2 sec	ON/OFF	---	---	---	ON
STRCPMPA	STATION 5 COMM FAIL	---	---	DIN	14.2.4.9	---	25	3045	G2C2.18	2 sec	ON/OFF	---	---	---	---
TRSM1M2	STRIPPER FORCE HEAT ALARM LOW (85°F)	TAS1-01MS1-2-1	---	DIN	14.2.4.9	---	100068	---	G2C2.18	30 sec	---	---	---	---	---
TRST001	TANK BOT & SIDE TIC #1	---	---	TYPE J	13.1.4	---	---	402116	H2C9.10	30 sec	80/140	°F	---	---	---
TRST002	TANK BOT & SIDE TIC #2	---	---	TYPE J	13.1.5	---	---	402117	H2C9.11	30 sec	80/140	°F	---	---	---
TRST003	TANK BOT & SIDE TIC #3	---	---	TYPE J	13.1.6	---	---	402118	H2C9.12	30 sec	80/140	°F	---	---	---
TRST004	TANK BOT & SIDE TIC #4	---	---	TYPE J	13.1.7	---	---	402119	H2C9.13	30 sec	80/140	°F	---	---	---
TRST005	TANK BOT & SIDE TIC #5	---	---	TYPE J	13.1.8	---	---	402120	H2C9.14	30 sec	80/140	°F	---	---	---
TRST006	TANK BOT & SIDE TIC #6	---	---	TYPE J	13.1.9	---	---	402121	H2C9.15	30 sec	80/140	°F	---	---	---
TRST007	TANK BOT & SIDE TIC #7	---	---	TYPE J	13.1.10	---	---	402122	H2C9.16	30 sec	80/140	°F	---	---	---
TRST008	TANK BOT & SIDE TIC #8	---	---	TYPE J	13.1.11	---	---	402123	H2C9.17	30 sec	80/140	°F	---	---	---
TRST009	TANK BOT & SIDE TIC #9	---	---	TYPE J	13.1.12	---	---	402124	H2C9.18	30 sec	80/140	°F	---	---	---
TRST010	TANK BOT & SIDE TIC #10	---	---	TYPE J	13.1.13	---	---	402125	H2C9.19	30 sec	80/140	°F	---	---	---
TRST011	TANK BOT & SIDE TIC #11	---	---	TYPE J	13.1.14	---	---	402126	H2C9.20	30 sec	80/140	°F	---	---	---
TRST012	TANK BOT & SIDE TIC #12	---	---	TYPE J	13.1.15	---	---	402127	H2C9.21	30 sec	80/140	°F	---	---	---
TRST013	TANK BOT & SIDE TIC #13	---	---	TYPE J	13.1.16	---	---	402128	H2C9.22	30 sec	80/140	°F	---	---	---
TRST014	TANK BOT & SIDE TIC #14	---	---	TYPE J	13.1.17	---	---	402129	H2C9.23	30 sec	80/140	°F	---	---	---
TRST015	TANK BOT & SIDE TIC #15	---	---	TYPE J	13.1.18	---	---	402130	H2C9.24	30 sec	80/140	°F	---	---	---
TRST016	TANK BOT & SIDE TIC #16	---	---	TYPE J	13.1.19	---	---	402131	H2C9.25	30 sec	80/140	°F	---	---	---
TRST017	TANK BOT & SIDE TIC #17	---	---	TYPE J	13.1.20	---	---	402132	H2C9.26	30 sec	80/140	°F	---	---	---
TRST018	TANK BOT & SIDE TIC #18	---	---	TYPE J	13.1.21	---	---	402133	H2C9.27	30 sec	80/140	°F	---	---	---
TRST019	TANK BOT & SIDE TIC #19	---	---	TYPE J	13.1.22	---	---	402134	H2C9.28	30 sec	80/140	°F	---	---	---
TRST020	TANK BOT & SIDE TIC #20	---	---	TYPE J	13.1.23	---	---	402135	H2C9.29	30 sec	80/140	°F	---	---	---
TRST021	TANK BOT & SIDE TIC #21	---	---	TYPE J	13.1.24	---	---	402136	H2C9.30	30 sec	80/140	°F	---	---	---
TRST022	TANK BOT & SIDE TIC #22	---	---	TYPE J	13.1.25	---	---	402137	H2C9.31	30 sec	80/140	°F	---	---	---
TRST023	TANK BOT & SIDE TIC #23	---	---	TYPE J	13.1.26	---	---	402138	H2C9.32	30 sec	80/140	°F	---	---	---
TRST024	TANK BOT & SIDE TIC #24	---	---	TYPE J	13.1.27	---	---	402139	H2C9.33	30 sec	80/140	°F	---	---	---
TRST025	TANK BOT & SIDE TIC #25	---	---	TYPE J	13.1.28	---	---	402140	H2C9.34	30 sec	80/140	°F	---	---	---
TRST026	TANK BOT & SIDE TIC #26	---	---	TYPE J	13.1.29	---	---	402141	H2C9.35	30 sec	80/140	°F	---	---	---
TRST027	TANK BOT & SIDE TIC #27	---	---	TYPE J	13.1.30	---	---	402142	H2C9.36	30 sec	80/140	°F	---	---	---
TRSM1Y8	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.39	---	300055	---	G1C9.7	30 sec	80/140	°F	---	---	---
TRSM1Y9	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.40	---	300056	---	G1C9.8	30 sec	80/140	°F	---	---	---
TRSM1Y10	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.41	---	300057	---	G1C9.9	30 sec	80/140	°F	---	---	---
TRSM1Y11	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.42	---	300058	---	G1C9.10	30 sec	80/140	°F	---	---	---
TRSM1Y12	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.43	---	300059	---	G1C9.11	30 sec	80/140	°F	---	---	---
TRSM1Y13	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.44	---	300060	---	G1C9.12	30 sec	80/140	°F	---	---	---
TRSM1Y14	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.45	---	300061	---	G1C9.13	30 sec	80/140	°F	---	---	---
TRSM1Y15	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.46	---	300062	---	G1C9.14	30 sec	80/140	°F	---	---	---
TRSM1Y16	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.47	---	300063	---	G1C9.15	30 sec	80/140	°F	---	---	---
TRSM1Y17	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.48	---	300064	---	G1C9.16	30 sec	80/140	°F	---	---	---
TRSM1Y18	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.49	---	300065	---	G1C9.17	30 sec	80/140	°F	---	---	---
TRSM1Y19	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.50	---	300066	---	G1C9.18	30 sec	80/140	°F	---	---	---
TRSM1Y20	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.51	---	300067	---	G1C9.19	30 sec	80/140	°F	---	---	---
TRSM1Y21	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.52	---	300068	---	G1C9.20	30 sec	80/140	°F	---	---	---
TRSM1Y22	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.53	---	300069	---	G1C9.21	30 sec	80/140	°F	---	---	---
TRSM1Y23	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.54	---	300070	---	G1C9.22	30 sec	80/140	°F	---	---	---
TRSM1Y24	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.55	---	300071	---	G1C9.23	30 sec	80/140	°F	---	---	---
TRSM1Y25	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.56	---	300072	---	G1C9.24	30 sec	80/140	°F	---	---	---
TRSM1Y26	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.57	---	300073	---	G1C9.25	30 sec	80/140	°F	---	---	---
TRSM1Y27	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.58	---	300074	---	G1C9.26	30 sec	80/140	°F	---	---	---
TRSM1Y28	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.59	---	300075	---	G1C9.27	30 sec	80/140	°F	---	---	---
TRSM1Y29	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.60	---	300076	---	G1C9.28	30 sec	80/140	°F	---	---	---
TRSM1Y30	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.61	---	300077	---	G1C9.29	30 sec	80/140	°F	---	---	---
TRSM1Y31	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.62	---	300078	---	G1C9.30	30 sec	80/140	°F	---	---	---
TRSM1Y32	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.63	---	300079	---	G1C9.31	30 sec	80/140	°F	---	---	---
TRSM1Y33	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.64	---	300080	---	G1C9.32	30 sec	80/140	°F	---	---	---
TRSM1Y34	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.65	---	300081	---	G1C9.33	30 sec	80/140	°F	---	---	---
TRSM1Y35	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.66	---	300082	---	G1C9.34	30 sec	80/140	°F	---	---	---
TRSM1Y36	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.67	---	300083	---	G1C9.35	30 sec	80/140	°F	---	---	---
TRSM1Y37	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.68	---	300084	---	G1C9.36	30 sec	80/140	°F	---	---	---
TRSM1Y38	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.69	---	300085	---	G1C9.37	30 sec	80/140	°F	---	---	---
TRSM1Y39	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.70	---	300086	---	G1C9.38	30 sec	80/140	°F	---	---	---
TRSM1Y40	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.71	---	300087	---	G1C9.39	30 sec	80/140	°F	---	---	---
TRSM1Y41	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.72	---	300088	---	G1C9.40	30 sec	80/140	°F	---	---	---
TRSM1Y42	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.73	---	300089	---	G1C9.41	30 sec	80/140	°F	---	---	---
TRSM1Y43	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.74	---	300090	---	G1C9.42	30 sec	80/140	°F	---	---	---
TRSM1Y44	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.75	---	300091	---	G1C9.43	30 sec	80/140	°F	---	---	---
TRSM1Y45	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.76	---	300092	---	G1C9.44	30 sec	80/140	°F	---	---	---
TRSM1Y46	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.77	---	300093	---	G1C9.45	30 sec	80/140	°F	---	---	---
TRSM1Y47	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.78	---	300094	---	G1C9.46	30 sec	80/140	°F	---	---	---
TRSM1Y48	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.79	---	300095	---	G1C9.47	30 sec	80/140	°F	---	---	---
TRSM1Y49	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.80	---	300096	---	G1C9.48	30 sec	80/140	°F	---	---	---
TRSM1Y50	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.81	---	300097	---	G1C9.49	30 sec	80/140	°F	---	---	---
TRSM1Y51	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.82	---	300098	---	G1C9.50	30 sec	80/140	°F	---	---	---
TRSM1Y52	FIR COOLING WATER TEMP	---	---	TYPE J	13.1.83	---	300099	---	G1C9.51	30 sec	80/140	°F	---	---	---

DACS-1 INPUT/OUTPUT CHANNEL LIST

Sorted By DACS TAG

File "io\td45.xls" Page 5 of 7

October 11, 1997

DACS TAG	TAG DESCRIPTION	FIELD	IO#AND/OR SOURCE	SIGNAL TYPE	LOC.	INPUT REG.#	XFR REG.#	GENESIS GROUP	SCAN TIME	RANGE LOW/HIGH	ENG. UNITS	ALARM LOW	ALARM HIGH	ALARM ABORT
TR17B13	TANK TEMP RISER 17B - 220*	TE-101-SV-17B-15	#2 - GMS-1	TYPE K	4.1,7.5	---	402192	HC27.8	6 sec	70/140	°F	70.01	130	135
TR17B14	TANK TEMP RISER 17B - 234*	TE-101-SV-17B-16	#2 - GMS-1	TYPE K	4.2,7.5	---	402233	HC13.2	6 sec	70/140	°F	70.01	130	135
TR17B15	TANK TEMP RISER 17B - 244*	TE-101-SV-17B-17	#2 - GMS-1	TYPE K	4.2,7.5	---	402234	HC13.2	6 sec	70/140	°F	70.01	130	135
TR17B16	TANK TEMP RISER 17B - 268*	TE-101-SV-17B-18	#2 - GMS-1	TYPE K	4.1,7.8	---	402235	HC13.4	6 sec	70/140	°F	70.01	130	135
TR17B17	TANK TEMP RISER 17B - 282*	TE-101-SV-17B-19	#2 - GMS-1	TYPE K	4.2,3.8	---	402236	HC13.4	6 sec	70/140	°F	70.01	130	135
TR17B18	TANK TEMP RISER 17B - 316*	TE-101-SV-17B-20	#2 - GMS-1	TYPE K	4.2,7.9	---	402237	HC13.5	6 sec	70/140	°F	70.01	130	135
TR17B19	TANK TEMP RISER 17B - 340*	TE-101-SV-17B-21	#2 - GMS-1	TYPE K	7.1,7.7	---	402238	HC13.6	6 sec	70/140	°F	70.01	130	135
TR17B20	TANK TEMP RISER 17B - 384*	TE-101-SV-17B-22	#2 - GMS-1	TYPE K	4.3,7.7	---	402239	HC13.7	6 sec	70/140	°F	70.01	130	135
TR17B21	TANK TEMP RISER 17B - 402*	TE-101-SV-17B-23	#2 - GMS-1	TYPE K	4.3,7.7	---	402240	HC13.8	6 sec	70/140	°F	70.01	130	135
TR17B22	TANK TEMP RISER 17B - 402*	TE-101-SV-17B-24	#2 - GMS-1	TYPE K	4.1,7.8	---	402241	HC13.4	6 sec	70/140	°F	70.01	130	135
TR17C01	TANK TEMP RISER 17C - 16*	---	#2 - GMS-1	TYPE K	4.2,4.1	---	402159	HC23.7	12 sec	70/140	°F	---	---	---
TR17C02	TANK TEMP RISER 17C - 16*	---	#2 - GMS-1	TYPE K	4.2,4.1	---	402160	HC23.8	12 sec	70/140	°F	---	---	---
TR17C03	TANK TEMP RISER 17C - 28*	---	#2 - GMS-1	TYPE K	4.2,6.1	---	402161	HC24.1	12 sec	70/140	°F	---	---	---
TR17C04	TANK TEMP RISER 17C - 52*	---	#2 - GMS-1	TYPE K	4.2,4.2	---	402162	HC24.2	12 sec	70/140	°F	---	---	---
TR17C05	TANK TEMP RISER 17C - 76*	---	#2 - GMS-1	TYPE K	4.2,6.2	---	402163	HC24.3	12 sec	70/140	°F	---	---	---
TR17C06	TANK TEMP RISER 17C - 100*	---	#2 - GMS-1	TYPE K	4.2,6.2	---	402164	HC24.4	12 sec	70/140	°F	---	---	---
TR17C07	TANK TEMP RISER 17C - 112*	---	#2 - GMS-1	TYPE K	4.2,4.3	---	402165	HC24.5	12 sec	70/140	°F	---	---	---
TR17C08	TANK TEMP RISER 17C - 124*	---	#2 - GMS-1	TYPE K	4.2,4.3	---	402166	HC24.6	12 sec	70/140	°F	---	---	---
TR17C09	TANK TEMP RISER 17C - 148*	---	#2 - GMS-1	TYPE K	4.2,4.3	---	402167	HC24.7	12 sec	70/140	°F	---	---	---
TR17C10	TANK TEMP RISER 17C - 172*	---	#2 - GMS-1	TYPE K	4.2,4.4	---	402168	HC24.8	12 sec	70/140	°F	---	---	---
TR17C11	TANK TEMP RISER 17C - 196*	---	#2 - GMS-1	TYPE K	4.2,4.4	---	402169	HC24.9	12 sec	70/140	°F	---	---	---
TR17C12	TANK TEMP RISER 17C - 200*	---	#2 - GMS-1	TYPE K	4.2,4.5	---	402170	HC25.1	12 sec	70/140	°F	---	---	---
TR17C13	TANK TEMP RISER 17C - 200*	---	#2 - GMS-1	TYPE K	4.2,4.5	---	402171	HC25.3	12 sec	70/140	°F	---	---	---
TR17C14	TANK TEMP RISER 17C - 232*	---	#2 - GMS-1	TYPE K	4.2,5.5	---	402172	HC25.4	12 sec	70/140	°F	---	---	---
TR17C15	TANK TEMP RISER 17C - 282*	---	#2 - GMS-1	TYPE K	4.2,5.5	---	402173	HC25.5	12 sec	70/140	°F	---	---	---
TR17C16	TANK TEMP RISER 17C - 316*	---	#2 - GMS-1	TYPE K	4.2,4.6	---	402174	HC25.6	12 sec	70/140	°F	---	---	---
TR17C17	TANK TEMP RISER 17C - 316*	---	#2 - GMS-1	TYPE K	4.2,4.6	---	402175	HC25.7	12 sec	70/140	°F	---	---	---
TR17C18	TANK TEMP RISER 17C - 340*	---	#2 - GMS-1	TYPE K	4.2,6.6	---	402176	HC25.8	12 sec	70/140	°F	---	---	---
TR17C19	TANK TEMP RISER 17C - 364*	---	#2 - GMS-1	TYPE K	4.2,6.7	---	402177	HC25.1	12 sec	70/140	°F	---	---	---
TR17C20	TANK TEMP RISER 17C - 382*	---	#2 - GMS-1	TYPE K	4.2,6.7	---	402178	HC26.2	12 sec	70/140	°F	---	---	---
TR17C21	TANK TEMP RISER 17C - 402*	---	#2 - GMS-1	TYPE K	4.2,6.7	---	402179	HC26.3	12 sec	70/140	°F	---	---	---
TR17C22	TANK TEMP RISER 17C - 425*	---	#2 - GMS-1	TYPE K	4.2,6.7	---	402180	HC26.4	12 sec	70/140	°F	---	---	---
TR17S15	FTK SAMPLE GAS TEMP	TE-GMS1-15-1	#2 - GMS-1	TYPE K	15.7,4.6	300046	402202	GT2.8	6 sec	327/150	°F	---	---	---
TR17S25	MS BUILDING TEMP	TE-GMS1-25-1	#2 - GMS-2	TYPE K	4.20/10A	300056	402202	GT2.8	6 sec	327/150	°F	---	---	---
TR17C23	NO1 INST. CABINET TEMP	TE-GMS1-23-1	#2 - GMS-1	TYPE K	14.1,3.6	300056	402202	GT2.8	6 sec	327/150	°F	---	---	---
TR17C24	NO2 INST. CABINET TEMP	TE-GMS1-24-1	#2 - GMS-1	TYPE K	14.1,3.6	300056	402202	GT2.8	6 sec	327/150	°F	---	---	---
TR17C25	NO3 INST. CABINET TEMP	TE-GMS1-25-1	#2 - GMS-1	TYPE K	14.1,3.6	300056	402202	GT2.8	6 sec	327/150	°F	---	---	---
TR17C26	NO4 INST. CABINET TEMP	TE-GMS1-26-1	#2 - GMS-1	TYPE K	14.1,3.6	300056	402202	GT2.8	6 sec	327/150	°F	---	---	---
TR17C27	NO5 INST. CABINET TEMP	TE-GMS1-27-1	#2 - GMS-1	TYPE K	14.1,3.6	300056	402202	GT2.8	6 sec	327/150	°F	---	---	---
TR17C28	NO6 INST. CABINET TEMP	TE-GMS1-28-1	#2 - GMS-1	TYPE K	14.1,3.6	300056	402202	GT2.8	6 sec	327/150	°F	---	---	---
TR17C29	NO7 INST. CABINET TEMP	TE-GMS1-29-1	#2 - GMS-1	TYPE K	14.1,3.6	300056	402202	GT2.8	6 sec	327/150	°F	---	---	---
TR17C30	NO8 INST. CABINET TEMP	TE-GMS1-30-1	#2 - GMS-1	TYPE K	14.1,3.6	300056	402202	GT2.8	6 sec	327/150	°F	---	---	---
TR17C31	NO9 INST. CABINET TEMP	TE-GMS1-31-1	#2 - GMS-1	TYPE K	14.1,3.6	300056	402202	GT2.8	6 sec	327/150	°F	---	---	---
TR17C32	NO10 INST. CABINET TEMP	TE-GMS1-32-1	#2 - GMS-1	TYPE K	14.1,3.6	300056	402202	GT2.8	6 sec	327/150	°F	---	---	---
TR17C33	NO11 INST. CABINET TEMP	TE-GMS1-33-1	#2 - GMS-1	TYPE K	14.1,3.6	300056	402202	GT2.8	6 sec	327/150	°F	---	---	---
TR17C34	NO12 INST. CABINET TEMP	TE-GMS1-34-1	#2 - GMS-1	TYPE K	14.1,3.6	300056	402202	GT2.8	6 sec	327/150	°F	---	---	---
TR17C35	NO13 INST. CABINET TEMP	TE-GMS1-35-1	#2 - GMS-1	TYPE K	14.1,3.6	300056	402202	GT2.8	6 sec	327/150	°F	---	---	---
TR17C36	NO14 INST. CABINET TEMP	TE-GMS1-36-1	#2 - GMS-1	TYPE K	14.1,3.6	300056	402202	GT2.8	6 sec	327/150	°F	---	---	---
TR17C37	NO15 INST. CABINET TEMP	TE-GMS1-37-1	#2 - GMS-1	TYPE K	14.1,3.6	300056	402202	GT2.8	6 sec	327/150	°F	---	---	---
TR17C38	NO16 INST. CABINET TEMP	TE-GMS1-38-1	#2 - GMS-1	TYPE K	14.1,3.6	300056	402202	GT2.8	6 sec	327/150	°F	---	---	---
TR17C39	NO17 INST. CABINET TEMP	TE-GMS1-39-1	#2 - GMS-1	TYPE K	14.1,3.6	300056	402202	GT2.8	6 sec	327/150	°F	---	---	---
TR17C40	NO18 INST. CABINET TEMP	TE-GMS1-40-1	#2 - GMS-1	TYPE K	14.1,3.6	300056	402202	GT2.8	6 sec	327/150	°F	---	---	---
TR17C41	NO19 INST. CABINET TEMP	TE-GMS1-41-1	#2 - GMS-1	TYPE K	14.1,3.6	300056	402202	GT2.8	6 sec	327/150	°F	---	---	---
TR17C42	NO20 INST. CABINET TEMP	TE-GMS1-42-1	#2 - GMS-1	TYPE K	14.1,3.6	300056	402202	GT2.8	6 sec	327/150	°F	---	---	---
TR17C43	NO21 INST. CABINET TEMP	TE-GMS1-43-1	#2 - GMS-1	TYPE K	14.1,3.6	300056	402202	GT2.8	6 sec	327/150	°F	---	---	---
TR17C44	NO22 INST. CABINET TEMP	TE-GMS1-44-1	#2 - GMS-1	TYPE K	14.1,3.6	300056	402202	GT2.8	6 sec	327/150	°F	---	---	---
TR17C45	NO23 INST. CABINET TEMP	TE-GMS1-45-1	#2 - GMS-1	TYPE K	14.1,3.6	300056	402202	GT2.8	6 sec	327/150	°F	---	---	---
TR17C46	NO24 INST. CABINET TEMP	TE-GMS1-46-1	#2 - GMS-1	TYPE K	14.1,3.6	300056	402202	GT2.8	6 sec	327/150	°F	---	---	---
TR17C47	NO25 INST. CABINET TEMP	TE-GMS1-47-1	#2 - GMS-1	TYPE K	14.1,3.6	300056	402202	GT2.8	6 sec	327/150	°F	---	---	---
TR17C48	NO26 INST. CABINET TEMP	TE-GMS1-48-1	#2 - GMS-1	TYPE K	14.1,3.6	300056	402202	GT2.8	6 sec	327/150	°F	---	---	---
TR17C49	NO27 INST. CABINET TEMP	TE-GMS1-49-1	#2 - GMS-1	TYPE K	14.1,3.6	300056	402202	GT2.8	6 sec	327/150	°F	---	---	---
TR17C50	NO28 INST. CABINET TEMP	TE-GMS1-50-1	#2 - GMS-1	TYPE K	14.1,3.6	300056	402202	GT2.8	6 sec	327/150	°F	---	---	---
TR17C51	NO29 INST. CABINET TEMP	TE-GMS1-51-1	#2 - GMS-1	TYPE K	14.1,3.6	300056	402202	GT2.8	6 sec	327/150	°F	---	---	---
TR17C52	NO30 INST. CABINET TEMP	TE-GMS1-52-1	#2 - GMS-1	TYPE K	14.1,3.6	300056	402202	GT2.8	6 sec	327/150	°F	---	---	---
TR17C53	NO31 INST. CABINET TEMP	TE-GMS1-53-1	#2 - GMS-1	TYPE K	14.1,3.6	300056	402202	GT2.8	6 sec	327/150	°F	---	---	---
TR17C54	NO32 INST. CABINET TEMP	TE-GMS1-54-1	#2 - GMS-1	TYPE K	14.1,3.6	300056	402202	GT2.8	6 sec	327/150	°F	---	---	---
TR17C55	NO33 INST. CABINET TEMP	TE-GMS1-55-1	#2 - GMS-1	TYPE K	14.1,3.6	300056	402202	GT2.8	6 sec	327/150	°F	---	---	---
TR17C56	NO34 INST. CABINET TEMP	TE-GMS1-56-1	#2 - GMS-1	TYPE K	14.1,3.6	300056	402202	GT2.8	6 sec	327/150	°F	---	---	---
TR17C57	NO35 INST. CABINET TEMP	TE-GMS1-57-1	#2 - GMS-1	TYPE K	14.1,3.6	300056	402202	GT2.8	6 sec	327/150	°F	---	---	---
TR17C58	NO36 INST. CABINET TEMP	TE-GMS1-58-1	#2 - GMS-1	TYPE K	14.1,3.6	300056	402202	GT2.8	6 sec	327/150	°F	---	---	---
TR17C59	NO37 INST. CABINET TEMP	TE-GMS1-59-1	#2 - GMS-1	TYPE K	14.1,3.6	300056	402202	GT2.8	6 sec	327/150	°F	---	---	---
TR17C60	NO38 INST. CABINET TEMP	TE-GMS1-60-1	#2 - GMS-1	TYPE K	14.1,3.6	300056	402202	GT2.8	6 sec	327/150	°F	---	---	---
TR17C61	NO39 INST. CABINET TEMP	TE-GMS1-61-1	#2 - GMS-1	TYPE K	14.1,3.6	300056	402202	GT2.8	6 sec	327/150	°F	---	---	---
TR17C62	NO40 INST. CABINET TEMP	TE-GMS1-62-1	#2 - GMS-1	TYPE K	14.1,3.6	300								

Appendix F: ASCII/BASIC Listings

```
10 REM
15 REM Filename: GC3.BAS
20 REM
25 REM Version: 2.0
30 REM
35 REM Purpose: GC3, GC2 and FTIR Data Collection
40 REM Reads gas data and places it in registers for PLC access.
45 REM Port 0 for terminal communication
50 REM Port 1 for data from the gc3 computer
55 REM
60 REM Modifications:
65 REM V1.3 - GC2 H2 split into ms byte and ls byte
66 REM V2.0 - Comments added
67 REM V2.1 - GC2 removed, PHOTO NH3 added.
70 REM
75 REM
100 PORT0
110 REM SND(8) is read by the PLC. Indicates that data is available.
120 REM
130 SND(8)=0
140 Z=-1
150 PRINT "GC-3 Data Collection"
160 PORT1
170 REM Clear reg(30). This register is set by the PLC when data
180 REM has been captured.
190 REM
200 REG(30)=0
210 REM
220 REM Look for start of transmission (STX - ASCII 2)
230 DO
240 Q=GET
250 REM Strip off any parity bits.
260 REM
270 IF Q>127 THEN Q=Q-128
280 UNTIL Q=2
290 REM
300 REM STX found. Input 17 data values and place in registers 0-16.
310 FOR I=0 TO 13
320 PORT1
330 INPUT X
340 REG(I)=X
350 PORT0
360 PRINT REG(I)
370 NEXT I
380 REM Data is in registers. Notify PLC by setting SND(8) to 1.
390 REM
400 SND(8)=1
```

```
410 REM Wait until PLC gets data, then exit. PLC will automatically
420 REM Re-run program.
430 REM
440 IF(REG(30)=0)THEN 440
450 SND(8)=0
460 END
```

```
10 REM
15 REM
16 REM  Filename: RGA5.BAS
20 REM  Author:   Jeff Martin
25 REM  Version:  1.0
30 REM
35 REM  Purpose:  RGA-5 Data Collection
40 REM             Extract Data from RGA-5 ASCII stream and send it to
45 REM             the PLC.
50 REM
55 REM  Notes:    Port 0 is for terminal communication,
60 REM             Port 1 receives the RGA-5 data stream.
65 REM
70 REM             The following ASCII formats are used by the program
75 REM             RMSG1: S0,D3
80 REM             RMSG2: S1,A1
85 REM             RMSG3: S2
90 REM             RMSG4: S2,D2,D2
95 REM             RMSG5: S4,F6.1
100 REM            RMSG6: S6,D1,D4,D4
105 REM            RMSG7: S9,D3
110 REM            RMSG8: S10
115 REM            RMSG9: S10,F6.1
120 REM            RMSG10: S12,D1,D4,D4
125 REM            RMSG11: S15,D3
130 REM            RMSG12: S16
135 REM
140 REM            These ASCII formats need to be loaded into the ASCII
145 REM            section of the ASCII/Basic module.
150 REM            In addition, the ASCII/Basic module must be set for no
155 REM            delimiter (DL# 0) and for a prefix string of "=" (PR# 30h).
160 REM            XON/XOFF should be enabled (X1# 1). The communications
165 REM            parameters are 1200 baud, 8 bits, no parity, 1 stop
170 REM            bit.
175 REM            The program should be loaded into RAM:2 of the Basic module.
180 REM
185 REM  Modifications:
190 REM
195 REM


---


280 ONERR 6000
285 PORT0
290 REM  SND(8) is sent when data has been collected.  PLC uses this
295 REM  to determine when to grab data.
300 REM
305 SND(8)=0
310 REM  "Z" is a constant used in Subroutine at line 1390
315 REM
320 Z=-1
325 PRINT "RGA-5 Data Collection"
330 REM  Search ASCII stream for the word "Run"
335 REM  Reg(30) is set to 1 by the PLC when it has read the data
340 REM
345 REG(30)=0
```

```
350 port1
355 DO
360 Q=GET
365 UNTIL Q=ASC("R")
370 REM Keep reading characters until an "R" is found.
375 REM
380 GOSUB 1390
385 REM Subroutine 1390 will return the next character in Q
390 IF Q->ASC("u")THEN 355
395 GOSUB 1390
400 IF Q->ASC("n")THEN 355
405 PORT0
410 PRINT "Message detected"
415 PORT1
420 REM Read data by issuing rmsg's to the ASCII processor
425 REM
430 FOR I = 1 TO 12
435   RMSG(I)
445 NEXT I
450 REM
455 REM Put data to be sent to PLC into registers 50-60.
460 REM Some data is just copied over, others are converted.
465 REM
470 REM Copy run number
475 REM
480 reg(50) = reg(0)
485 REM
490 REM
495 REM
500 REM
505 REM We are expecting stream to be "Tank", "Calibrate" or "Flush".
510 REM Check the first character, set status to 0 for tank, 1 for calibrate
515 REM 2 for flush and 3 for unknown.
520 REM
525 reg(51)=3
530 if reg(1) = asc("t") .OR. reg(1) = asc("T") then reg(51)=0
535 if reg(1) = asc("c") .OR. reg(1) = asc("C") then reg(51)=1
540 if reg(1) = asc("f") .OR. reg(1) = asc("F") then reg(51)=2
545 REM
550 REM
555 REM Store time as hhmm.
560 REM
565 reg(52)=reg(2)*100+reg(3)
570 REM
575 REM
580 REM Scale concentration to a 16 bit register
585 REM
590 reg(53) = int((reg(4)*1000.0 + reg(5)/10.0) * (65535.0/9999.9))
595 REM
600 REM Break the area down so that it fits into two registers.
605 REM To re-constitute: H2A Area = reg(54)*32000 + reg(55)
610 REM
615 A = (reg(6)*3125.0) + (reg(7)*0.3125) + (reg(8)/32000.0)
620 reg(54) = int(A)
625 reg(55) = int((A-int(A))*32000.0+0.5)
```

```
630 REM
635 REM
640 REM Copy H2A Retention time
645 REM
650 reg(56) = reg(9)
655 REM
660 REM
665 REM H2B Concentration scaled to a 16 bit register
670 REM
675 reg(57) = int((reg(10)*1000.0 + reg(11)/10.0) * (65535.0/9999.9))
680 REM
685 REM H2B Area
690 REM
695 A = (reg(12)*3125.0) + (reg(13)*0.3125) + (reg(14)/32000.0)
700 reg(58) = int(A)
705 reg(59) = int((A-int(A))*32000.0+0.5)
710 REM
715 REM H2B Retention Time
720 reg(60) = reg(15)
725 REM
730 REM Print data to terminal
735 port0
740 print
745 print "Run =                ",reg(50)
750 print "Status =               ",reg(51)
755 print "Time =                  ",reg(52)
760 print
765 print "H2A Concentration =     ",(reg(53)/65535.0)*9999.9
770 print "H2A Area =              ",reg(54)*32000.0+reg(55)
775 print "H2A retention time =    ",reg(56)
780 print
785 print "H2B Concentration =     ",(reg(57)/65535.0)*9999.9
790 print "H2B Area =              ",reg(58)*32000.0+reg(59)
795 print "H2B retention time =    ",reg(60)
800 print
805 REM Data is in the registers. Set SND(8) to notify PLC.
810 REM
815 SND(8)=1
820 REM Wait until PLC grabs data (it will set REG(30) to 1), then
825 REM exit. PLC will automatically re-run the program.
830 REM
835 IF(REG(30)=0)THEN 835
840 SND(8)=0
845 END

1350 REM
1360 REM Subroutine for reading characters. If no character is available,
1370 REM GET returns a -1. This subroutine strips off these -1's until a
1380 REM character is available.
1390 DO : Q=GET : UNTIL Q<>Z
1400 RETURN

6000 port0
```

6005 print "Error ",ERRNO," Detected
6010 END

Appendix G: PLC I/O Status Registers

The Modicon I/O health status alarms GENESIS when a module is no longer in good health. In Section 8.3.5, the facility for GENESIS determining the health of the I/O modules in the system was discussed. This included the conditions for good health, the STAT blocks used in the ladder logic to define the module health registers that are transmitted to GENESIS, and the PDIN blocks in the GENESIS strategy that receive them.

The following is a complete list of the PLC registers and their corresponding GENESIS strategy variable names.

Table G-1. I/O Health Status Implementation Table

Drop	Rack	Slot	STAT block Register	Transmit Register	PDIN Name	Output
2	1		40317	40108	H3C0	
		4				DO13
		5				DO12
		6				DO11
2	2	3	40318	40109	H3C1	DO14
		4				DO13
4	1	4	40327	40110	H3C2	DO13
		5				DO12
		6				DO11
		7				DO10
4	2	3	40328	40115	H3C3	DO14
		4				DO13
		5				DO12
		6				DO11
		7				DO10
6	1	4	40337	40111	H3C3	DO13
		5				DO12
		6				DO11
		7				DO10

Drop	Rack	Slot	STAT block Register	Transmit Register	PDIN Name	Output
8	1	4	40347	40112	H4C0	DO13
		5				DO12
		6				DO11
		7				DO10
8	2	3	40348	40113	H4C1	DO14
		4				DO13
		5				DO12
9	1	4	40352	40114	H4C2	DO13
		5				DO12
		6				DO11
		7				DO10
13	1	4	40372	40120	H6C0	DO13
		5				DO12
		6				DO11
		7				DO10
		8				DO9
14	1	4	40377	40116	H5C0	DO13
		5				DO12
		6				DO11
14	2	3	40378	40117	H5C1	DO14
		4				DO13
		5				DO12
		6				DO11
		7				DO10
15	1	4	40382	40118	H5C2	DO13
		5				DO12
		6				DO11
		7				DO10
16	1	4	40387	40119	H5C3	DO13
		5				DO12
		6				DO11
		7				DO10
		8				DO9

Appendix H: Network Configuration Settings

Station Name:	STATION5	STATION6	STATION7	STATION8	STATION9
Function:	Runtime Master	Local RSS	Local RSS	Runtime Master	DOS File Server
GEN-NET S/N:	GN-001950	GN-001948	GN-001949	---	---
GENESIS Key S/N:	GEN-6-C 1494	RSS 465	GEN-6-RT 752	GEN-6-C 1479	---
Memory Buffer Address	D980	D980	D980	D980	D980
I/O Port Address:	2E0	2E0	2E0	2E0	2E0
IRQ:	5	5	5	5	5
Data Check Mode:	CRC-16	CRC-16	CRC-16	CRC-16	CRC-16
Network Address:	5	6	7	8	9
ARCNET Hub Port	#1-5	#1-6	#1-7	---	#1-4
Node Fail Time:	20	20	20	20	20
Watchdog Message Rate:	8	8	8	8	8
Performance Monitor Rate:	4	4	4	4	4
Network Scan Time:	1	1	1	1	1
Message Retry Count:	3	3	3	3	---
Transmit Timeout:	8	8	8	8	8
Primary ACK setting:	80	80	80	80	80
Net Alarm Squelch After Startup:	0	---	---	0	---
Net Alarm Squelch During Startup:	10	---	---	10	---
Net Alarm Buffer size:	4096	---	---	512	---
Transmit Buffer Full Level:	80	---	---	10	---
RX Buffers:	32,32,16,16,32	---	---	8,8,4,4,8	---
Rename Duplicates on file transfer:	Yes	---	---	Yes	Yes
Alarm Broadcasts:	ENABLE	---	---	DISABLE	---
Network Point Database Size:	---	300	300	---	---
RSS Master Identification:	---	MANUAL	MANUAL	---	---
RSS Auto Reconfiguration:	---	ENABLED	ENABLED	---	---
RSS Master Node Name	---	STATION5	STATION5	---	---

Station Name:	STATION10	STATION11	STATION13	STATION15	STATION17
Function:	Local RSS	Remote RSS	Remote RSS	Remote RSS	Remote RSS
GEN-NET S/N:	GN-002933	GN-003430	GN-003288	GN-001942	GN-003227
GENESIS Key S/N:	GEN-6-RT 702	RSS 773	RSS 524	RSS 523	RSS 822
Memory Buffer Address	CC00	CC00	CC00	CC00	CC00
I/O Port Address:	300	300	300	2E0	2E0
IRQ:	5	5	5	5	5
Data Check Mode:	CRC-16	CRC-16	CRC-16	CRC-16	CRC-16
Network Address:	19	11	13	15	17
ARCNET Hub Port	#2-1	#1-2	#1-8	#1-1	#1-3
Node Fail Time:	20	20	20	20	20
Watchdog Message Rate:	8	8	8	8	8
Performance Monitor Rate:	4	4	4	4	4
Network Scan Time:	4	20	20	20	20
Message Retry Count:	3	3	3	3	3
Transmit Timeout:	8	8	8	8	8
Primary ACK setting:	80	80	80	80	80
Net Alarm Squelch After Startup:	---	---	---	---	---
Net Alarm Squelch During Startup:	---	---	---	---	---
Net Alarm Buffer size:	---	---	---	---	---
Transmit Buffer Full Level:	---	---	---	---	---
RX Buffers:	---	---	---	---	---
Rename Duplicates on file transfer:	---	---	---	---	---
Alarm Broadcasts:	---	---	---	---	---
Network Point Database Size:	---	---	---	---	---
RSS Master Identification:	MANUAL	MANUAL	MANUAL	MANUAL	MANUAL
RSS Auto Reconfiguration:	DISABLED	DISABLED	DISABLED	DISABLED	DISABLED
RSS Master Node Name	STATIONS	STATIONS	STATIONS	STATIONS	STATIONS

Appendix I: Computer Configurations and Software Versions

See Appendix H for a general description of the functions and network setup parameters (ARCNET VO Port, Memory Address, and IRQ) of the stations in the network.

All stations are running DOS, site standard version. The versions of Iconics software that are in use are:

- GENESIS (GEN-6C and GEN-RT) 3.72
- Remote Supervisory System (RSS) 3.72
- GEN-NET 3.64
- Report & Recipe option (RRCOMBO) 3.62

The following table gives the parameters for the setup of the ARCNET cards and the Iconics software packages that are on the stations. GEN-NET and RSS serial numbers are provided as well, for those stations that have that software.

Table H-1. ARCNET Card Parameters and Iconics Software

	ARCNET Hub Port	ARCNET I/O Port Address	ARCNET Board IRQ	Iconics Software
STATION5	#1-5	2E0	5	GEN-6C, GEN-NET, R&R
STATION6	#1-6	2E0	5	RSS, R&R
STATION7	#1-7	2E0	5	GEN-6RT, GEN-NET, R&R
STATION8	N/A	2E0	5	GEN-6C, GEN-NET, R&R
STATION9	#1-4	2E0	5	GEN-NET
STATION10	#2-1	300	5	RSS, R&R
STATION11	#1-2	300	5	RSS, R&R
STATION13	#1-8	300	5	RSS, R&R
STATION15	#1-1	2E0	5	RSS, R&R
STATION17	#1-3	2E0	5	RSS, R&R

The directory organization on the computers is the same for the following directories:

- \(\Root directory) - AUTOEXEC.BAT and CONFIG.SYS system boot files
- \DOS - MS-DOS system files
- \MOUSE - Mouse driver
- \NETROOM3 - Expanded Memory Manager

Where directories differ between stations, the following table shows what those directories contain.

Table H-2. Contents of GEN372, TEST and MOTOR Directories for Each Station

Station Name	Directory	Contents
STATION5	\GEN372	GENESIS system, GEN-NET, AF5000+/GENESIS driver, Modicon/GENESIS driver, RRCOMBO option, Help system files
	\TEST	TEST strategy files, related displays, R&R code, key macro files, state field files, startup batch files
STATION6	\GEN372	RSS software, AF5000+/GENESIS driver, Modicon/GENESIS driver, RRCOMBO option, Help system files
	\TEST	TEST strategy files, related displays, R&R code, key macro files, state field files, startup batch files
STATION7	\GEN372	RSS software, GEN-NET, AF5000+/GENESIS driver, Modicon/GENESIS driver, RRCOMBO option, Help system files
	\TEST	TEST strategy files, related displays, R&R code, key macro files, state field files, startup batch files
STATION8	\GEN372	GENESIS system, GEN-NET, AF5000+/GENESIS driver, Modicon/GENESIS driver, RRCOMBO option, Help system files
	\MOTOR	MOTOR strategy files, related displays, R&R code, key macro files, state field files, startup batch files
STATION9	\GEN372	GEN-NET

Station Name	Directory	Contents
STATION10	\GEN372	RSS software, AF5000+/GENESIS driver, Modicon/GENESIS driver, RRCOMBO option, Help system files
	\TEST	TEST strategy files, related displays, R&R code, key macro files, state field files, startup batch files
STATION11	\GEN372	RSS software, AF5000+/GENESIS driver, Modicon/GENESIS driver, RRCOMBO option, Help system files
	\TEST	TEST strategy files, related displays, R&R code, key macro files, state field files, startup batch files
STATION13	\GEN372	RSS software, AF5000+/GENESIS driver, Modicon/GENESIS driver, RRCOMBO option, Help system files
	\TEST	TEST strategy files, related displays, R&R code, key macro files, state field files, startup batch files
STATION15	\GEN372	RSS software, AF5000+/GENESIS driver, Modicon/GENESIS driver, RRCOMBO option, Help system files
	\TEST	TEST strategy files, related displays, R&R code, key macro files, state field files, startup batch files
STATION17	\GEN372	RSS software, AF5000+/GENESIS driver, Modicon/GENESIS driver, RRCOMBO option, Help system files
	\TEST	TEST strategy files, related displays, R&R code, key macro files, state field files, startup batch files

Appendix J: File Listings**DACS STATION 5 FILE LISTINGS****"ALMCOPIY.BAT" on Station 5**

```
copy \test\alarm00.txt \hist\686029AL.TXT
copy \test\alarm00.txt \hist\2\686029AL.TXT
```

"AUTOEXEC.BAT" on Station 5

```
@ECHO OFF
PROMPT $p$g
C:\NETROOM\XLOAD.EXE C:\MOUSE\MOUSE
goto %config%

:GEN354
PATH C:\DOS;C:\;C:\GENESIS;C:\NETROOM;C:\EXPERT
goto end

:GEN372
PATH C:\DOS;C:\;C:\GEN372;C:\NETROOM;C:\EXPERT
SHARE
goto end

:end
```

"CONFIG.SYS" on Station 5

```
[MENU]
menuitem=GEN354,Genesis system version 3.54
menuitem=GEN372,Genesis system version 3.72
menudefault=GEN372[,0]

[COMMON]
DEVICE=C:\NETROOM\RM386.EXE AUTO X=D900-DDFF
DEVICE=C:\NETROOM\SYSCLOAK.EXE
DEVICE=C:\NETROOM\XLOAD.SYS -O
BUFFERS=20,0
FILES=55
LASTDRIVE=E
FCBS=4,0
STACKS=0,0
SHELL=C:\NETROOM\XLOAD.EXE -SDE01 -M57021 -E C:\DOS\COMMAND.COM C:\DOS\ /p
```

DEVICE=C:\NETROOM\STACKS.EXE 9,256

[GEN354]

[GEN372]

"RUNTIME.BAT" on Station 5

```
:Filename:  RUNTIME.BAT rev.1  05-25-95  sos/rwt
:
:Purpose:   This batch file creates runtime files
:           for TEST and MOTOR strategies on Station 7.
:           Format of command:
:           runtime %1
:           where %1 is the strategy name to be used runtime
:
:
```

```
ECHO
CLS
@ECHO OFF
```

```
IF "%1"=="test" GOTO TEST
IF "%1"=="TEST" GOTO TEST
IF "%1"=="motor" GOTO MOTOR
IF "%1"=="MOTOR" GOTO MOTOR
```

```
REM will run other strategies
C:\GEN372\RUNTIME %1
GOTO END
```

```
:TEST
COPY C:\TEST\RUNTEST.DB C:\TEST\TEST.DB > XXX.XXX
COPY C:\TEST\RUNTEST.MDB C:\TEST\TEST.MDB > XXX.XXX
COPY C:\TEST\RUNTEST.XDB C:\TEST\TEST.XDB > XXX.XXX
COPY C:\TEST\RUNTEST.CI C:\TEST\TEST.CI > XXX.XXX
COPY C:\TEST\RUNTEST.CA C:\TEST\TEST.CA > XXX.XXX
COPY C:\TEST\RUNTEST.CFG C:\TEST\TEST.CFG > XXX.XXX
COPY C:\TEST\WELCOME.GRP C:\GEN372\BOOT.GSP
C:\TEST\ALMRENAM
CALL C:\ALMCOPY
DEL ALARM*.*
C:\GEN372\RUNTIME %1
GOTO END
```

```
:MOTOR
COPY C:\MOTOR\MAINMENU.GRP C:\GEN372\BOOT.GSP
C:\GEN372\RUNTIME %1
```

```
:END
```

"RENAME72.BAT" on Station 5

:Filename: RENAME72.BAT

:
:Purpose: This batch file deletes OLD72.PRN and renames
: NEW72.PRN to OLD72.PRN to help minimize the
: file size build up of NEW72.PRN
: Executed manually at DOS prompt when desired.
:

DEL OLD72.PRN
REN NEW72.PRN OLD72.PRN

DACS STATION 6 FILE LISTINGS

"AUTOEXEC.BAT" on Station 6

```
@ECHO OFF
PROMPT $P$G
C:\NETROOM\XLOAD.EXE -SB001 -M23248 C:\EXPERT\MOUSE
goto %config%
```

```
:GEN354
PATH C:\DOS;C:\;\C:\GENESIS;C:\NETROOM;C:\EXPERT
goto end
```

```
:GEN372
PATH C:\DOS;C:\;\C:\GEN372;C:\NETROOM;C:\EXPERT
SHARE
goto end
```

```
:end
```

```
CD \TEST
RSS TEST
```

"CONFIG.SYS" on Station 6

```
[MENU]
menuitem=GEN354,Genesis system version 3.54
menuitem=GEN372,Genesis system version 3.72
menudefault=GEN372[,0]
```

```
[COMMON]
DEVICE=C:\NETROOM\RM386.EXE AUTO X=D900-DDFF
DEVICE=C:\NETROOM\SYSCLOAK.EXE
DEVICE=C:\NETROOM\XLOAD.SYS -0
BUFFERS=20,0
FILES=65
LASTDRIVE=E
FCBS=4,0
SHELL=C:\NETROOM\XLOAD.EXE -SDE01 -M57021 -E C:\DOS\COMMAND.COM C:\DOS\ /p
STACKS=0,0
```

```
[GEN354]
```

```
[GEN372].
```

"RSS.BAT" on Station 6

```
:Filename: RSS.BAT rev.2 05-25-95 sos/rwt
:
:Purpose: This batch file creates RSS files for all RSS stations
:Format of command:
:      rss %1
:           where %1 is the strategy name (typically TEST)
:
:
:
```

```
ECHO
CLS
@ECHO OFF
```

```
IF "%1"=="test" GOTO TEST
IF "%1"=="TEST" GOTO TEST
f:\GEN372\RSS %1
GOTO END
```

```
:TEST
COPY f:\TEST\RSSTEST.DB f:\TEST\TEST.DB > XXX.XXX
COPY f:\TEST\RSSTEST.CI f:\TEST\TEST.CI > XXX.XXX
COPY f:\TEST\RSSTEST.XDB f:\TEST\TEST.XDB > XXX.XXX
COPY f:\TEST\RSSTEST.CA f:\TEST\TEST.CA > XXX.XXX
COPY f:\TEST\RSSTEST.CFG f:\TEST\TEST.CFG > XXX.XXX
COPY f:\TEST\RSSTEST.MDB f:\TEST\TEST.MDB > XXX.XXX
COPY WELCOME.GRP f:\GEN372\BOOT.GSP
COPY WELCOME.GRP f:\GEN372\RSSBOOT.GSP
COPY WELCOME.GRP RSSBOOT.GSP
f:\GEN372\RSS %1
```

```
:END
```

DACS STATION 7 FILE LISTINGS

"AUTOEXEC.BAT" on Station 7

```
@ECHO OFF
PROMPT $p$g
C:\EXPERT\MOUSE
goto %config%

:GEN354
PATH C:\DOS;C:\;\C:\RUNTIME;C:\GENESIS;C:\NETROOM
goto end

:GEN372
REM PATH C:\DOS;C:\;\C:\RUN372;C:\GEN372;C:\NETROOM
PATH C:\DOS;C:\;\C:\GEN372;C:\NETROOM;C:\EXPERT
SHARE
goto end

:end

CD \TEST
RSS TEST
```

"CONFIG.SYS" on Station 7

```
[MENU]
menuitem=GEN354,Genesis system version 3.54
menuitem=GEN372,Genesis system version 3.72
menudefault=GEN372[,0]

[COMMON]
DEVICE=C:\NETROOM\RM386.EXE AUTO X=D900-DDFF
DEVICE=C:\NETROOM\SYSCLOAK.EXE
DEVICE=C:\NETROOM\XLOAD.SYS -O
BUFFERS=20,0
FILES=55
LASTDRIVE=E
FCBS=4,0
SHELL=C:\NETROOM\XLOAD.EXE -SDE01 -M57021 -E C:\DOS\COMMAND.COM C:\DOS\ /p
STACKS=0,0
```

[GEN354]

[GEN372]

"RUNTIME.BAT" on Station 7

```
:Filename:  RUNTIME.BAT rev.1  05-25-95 sos/rwt
:
:Modified:  04-26-96 sos/rwt to disable runtime capability.
:
:Purpose:  This batch file creates runtime files
:          for TEST and MOTOR strategies on Station 7.
:          Format of command:
:          runtime %1
:          where %1 is the strategy name to be used runtime
```

```
ECHO
```

```
CLS
```

```
@ECHO OFF
```

```
IF "%1"=="test" GOTO TEST
```

```
IF "%1"=="TEST" GOTO TEST
```

```
IF "%1"=="motor" GOTO MOTOR
```

```
IF "%1"=="MOTOR" GOTO MOTOR
```

```
REM will run other strategies
```

```
C:\GEN372\RUNTIME %1
```

```
GOTO END
```

```
:TEST
```

```
REM COPY C:\TEST\RUNTEST.DB C:\TEST\TEST.DB > XXX.XXX
```

```
REM COPY C:\TEST\RUNTEST.MDB C:\TEST\TEST.MDB > XXX.XXX
```

```
REM COPY C:\TEST\RUNTEST.XDB C:\TEST\TEST.XDB > XXX.XXX
```

```
REM COPY C:\TEST\RUNTEST.CI C:\TEST\TEST.CI > XXX.XXX
```

```
REM COPY C:\TEST\RUNTEST.CA C:\TEST\TEST.CA > XXX.XXX
```

```
REM COPY C:\TEST\RUNTEST.CFG C:\TEST\TEST.CFG > XXX.XXX
```

```
REM COPY C:\TEST\WELCOME.GRP C:\GEN372\BOOT.GSP
```

```
REM C:\TEST\ALMRENAM
```

```
REM CALL C:\ALMCOPI
```

```
REM DEL ALARM*.*
```

```
REM C:\GEN372\RUNTIME %1
```

```
ECHO RUNTIME IS NOT AVAILABLE ON STATION 7
```

```
GOTO END
```

```
:MOTOR
```

```
REM COPY C:\MOTOR\MAINMENU.GRP C:\GEN372\BOOT.GSP
```

```
REM C:\GEN372\RUNTIME %1
```

```
ECHO RUNTIME IS NOT AVAILABLE ON STATION 7
```

```
:END
```

"RSS.BAT" on Station 7

```
:
:Filename:  RSS.BAT rev.2   05-25-95 sos/rwt
:
:Purpose:  This batch file creates RSS files for all RSS stations
:          Format of command:
:          rss %1
:          where %1 is the strategy name (typically TEST)
:
:
```

ECHO

CLS

@ECHO OFF

IF "%1"=="test" GOTO TEST

IF "%1"=="TEST" GOTO TEST

f:\GEN372\RSS %1

GOTO END

:TEST

COPY f:\TEST\RSSTEST.DB f:\TEST\TEST.DB > XXX.XXX

COPY f:\TEST\RSSTEST.CI f:\TEST\TEST.CI > XXX.XXX

COPY f:\TEST\RSSTEST.XDB f:\TEST\TEST.XDB > XXX.XXX

COPY f:\TEST\RSSTEST.CA f:\TEST\TEST.CA > XXX.XXX

COPY f:\TEST\RSSTEST.CFG f:\TEST\TEST.CFG > XXX.XXX

COPY f:\TEST\RSSTEST.MDB f:\TEST\TEST.MDB > XXX.XXX

COPY WELCOME.GRP f:\GEN372\BOOT.GSP

COPY WELCOME.GRP f:\GEN372\RSSBOOT.GSP

COPY WELCOME.GRP RSSBOOT.GSP

f:\GEN372\RSS %1

:END

DACS STATION 8 FILE LISTINGS

"AUTOEXEC.BAT" on Station 8

```
@ECHO OFF
PROMPT $p$g
C:\EXPERT\MOUSE
goto %config%

:GEN354
PATH C:\DOS;C:\;C:\GENESIS;C:\NETROOM
goto end

:GEN372
PATH C:\DOS;C:\;C:\GEN372;C:\NETROOM
SHARE
goto end

:end
```

"CONFIG.SYS" on Station 8

```
[MENU]
menuitem=GEN354,Genesis version 3.54
menuitem=GEN372,Genesis version 3.72
menudefault=GEN372[,0]

[COMMON]
DEVICE=C:\NETROOM\RM386.EXE AUTO X=D900-DDFF
DEVICE=C:\NETROOM\SYSCLOAK.EXE
DEVICE=C:\NETROOM\XLOAD.SYS -0
BUFFERS=20,0
FILES=55
LASTORIVE=E
FCBS=4,0
SHELL=C:\NETROOM\XLOAD.EXE -SDE01 -M67021 -E C:\DOS\COMMAND.COM C:\DOS\ /p
STACKS=0,0

[GEN354]

[GEN372]
```

"RUNTIME.BAT" on Station 8

:Filename: RUNTIME.BAT rev.1 05-25-95 sos/rwt
:
:Purpose: This batch file creates runtime files
: for TEST and MOTOR strategies on Station 7.
: Format of command:
: runtime %1
: where %1 is the strategy name to be used runtime
:
:

ECHO
CLS
@ECHO OFF

IF "%1"=="test" GOTO TEST
IF "%1"=="TEST" GOTO TEST
IF "%1"=="motor" GOTO MOTOR
IF "%1"=="MOTOR" GOTO MOTOR

REM will run other strategies
C:\GEN372\RUNTIME %1
GOTO END

:TEST
COPY C:\TEST\RUNTEST.DB C:\TEST\TEST.DB > XXX.XXX
COPY C:\TEST\RUNTEST.MDB C:\TEST\TEST.MDB > XXX.XXX
COPY C:\TEST\RUNTEST.XDB C:\TEST\TEST.XDB > XXX.XXX
COPY C:\TEST\RUNTEST.CI C:\TEST\TEST.CI > XXX.XXX
COPY C:\TEST\RUNTEST.CA C:\TEST\TEST.CA > XXX.XXX
COPY C:\TEST\RUNTEST.CFG C:\TEST\TEST.CFG > XXX.XXX
COPY C:\TEST\WELCOME.GRP C:\GEN372\BOOT.GSP
C:\TEST\ALMRENAM
CALL C:\ALMCPY
DEL ALARM*. *
C:\GEN372\RUNTIME %1
GOTO END

:MOTOR
COPY C:\MOTOR\MAINMENU.GRP C:\GEN372\BOOT.GSP
C:\GEN372\RUNTIME %1

:END

DACS STATION 9 FILE LISTINGS

"AUTOEXEC.BAT" on Station 9

```
C:\DOS\SETVER SCPLUS.EXE 5.00
REM @ECHO OFF
SET OAD_DRIVER=C:\OADDOS
C:\NETROOM\XLOAD.EXE -C -AB001,27664 -AC156,12496 -ACE01,47232 C:\OADDOS\SCPLUS\SCPLUS
INSTALL
SET OAD_UTILITY=C:\OADDOS
PROMPT $p$g
goto %config%
```

```
:GEN354
PATH C:\DOS;C:\;C:\GENESIS;C:\MOUSE;C:\BAT;
goto finish
```

```
:GEN372
PATH C:\DOS;C:\;C:\GEN372;C:\MOUSE;C:\BAT;
SHARE
goto finish
```

```
:finish
C:\NETROOM\XLOAD.EXE -SCE01 -M23248 -X C:\MOUSE\MOUSE
```

```
@REM ==== LANMAN 2.1a ==== DO NOT MODIFY BETWEEN THESE LINES ==== LANMAN 2.1a ====
PATH C:\DOS;C:\LM\NETPROG;%PATH%
C:\NETROOM\XLOAD.EXE -C -AB001,4448 -AC156,13488 -AF001,35520 NET START WORKSTATION
NET LOGON DACCS9 DACCS9 /DOMAIN:STANDALONE
net use f: \\whc324\dpu
@REM ==== LANMAN 2.1a ==== DO NOT MODIFY BETWEEN THESE LINES ==== LANMAN 2.1a ====
guest
f:
cd \raw\transfer
```

```
REM =====GENESIS=====
GENCMDR
REM =====GENESIS=====
```

"CONFIG.SYS" on Station 9

[MENU]

```
menuitem=GEN354,Genesis system version 3.54
menuitem=GEN372,Genesis system version 3.72
menudefault=GEN372[,0]
```

[COMMON]

```
DEVICE=C:\NETROOM\RM386.EXE AUTO X=C800-CBFF X=D900-DDFF
rem DEVICE=C:\NETROOM\VIDCLOAK.EXE /PMBIOS
rem DEVICE=C:\NETROOM\SYSCLOAK.EXE
DEVICE=C:\NETROOM\XLOAD.SYS -O
REM DOS=HIGH
BUFFERS=17,0
FILES=55
LASTDRIVE=P
FCBS=4,0
DEVICE=C:\NETROOM\XLOAD.SYS -SC156 -M13728 C:\DOS\SETVER.EXE
DEVICE=C:\LOADDOS\DOSCFG.EXE /M1 /V
DEVICE=C:\NETROOM\XLOAD.SYS -SF001 -M51376 C:\LOADDOS\DOSOAD.SYS

SHELL=C:\NETROOM\XLOAD.EXE -SCE01 -M57021 -E C:\DOS\COMMAND.COM C:\DOS\ /p
STACKS=0,0
DEVICE=C:\NETROOM\XLOAD.SYS -SF001 -M2832 C:\NETROOM\STACKS.EXE 9,256
device=c:\scsi\aspipc2.sys
[GEN354]
```

[GEN372]

Appendix K: Software License Agreements*Table K-1. Commercial Software That Is Used in the DACS System*

Company	Software	Version
Helix	Netroom3	3.02
Kensington Microware, Ltd.	Expert Mouse	N/A
Iconics, Inc.	GENESIS Control Series	3.72
Iconics, Inc.	GEN-NET	3.64
Iconics, Inc.	Remote Supervisory Station	3.72
Modicon, Inc	Modsoft	2.1
Microsoft Corp.	MS-DOS	6.0
Microsoft Corp.	Windows	3.1
Nicolet Instrument Corp.	FAMOS	N/A
Symantec Corp.	Norton DiskLock	3.5

Appendix L: DeCipher Plus and DeTerminal Software

The DeCipher Plus software and DeTerminal software are used with the Datataker 50 data logger in support of the tank 241SY101 mixer pump removal. Descriptions of the software are summarized as follows:

DeCipher Plus Program Disk, version 1.1, July 1, 1992

Volume in drive A is DCP11_1

Directory of A:\

```

READ   ME           12,709 06-30-92  9:37a
README BAT          22 06-26-92  1:01p
INSTALL EXE        104,645 02-26-92 11:34p
DCP01100 001       448,830 06-30-92  9:51a
      4 file(s)          566,206 bytes
      160,768 bytes free

```

"READ.ME" File Listing

DeCipher Plus Release 1.1 - Release Notes

Contents

1. How to install version 1.1
2. Running the DeCipher Plus demonstration
3. Vital hints for getting started
4. Known software problems
5. Known documentation problems
6. What to do when you have problems
7. Revision History

1. How to install version 1.1

To install DeCipher Plus :-

- the DeCipher Plus program disk contains the installation program:-

INSTALL.EXE

- execute this program either by

```
drive_name:INSTALL <Enter>
(e.g. >a:INSTALL)
```

or

```
drive_name: <Enter>
INSTALL <Enter>
(e.g. >a:)
( >INSTALL)
```

where drive_name should be replaced by the letter denoting the drive in which the program disk now resides.

- as the INSTALL program executes, it

- * creates the DeCipher Plus directory structure
- * copies the DeCipher Plus files
- * manages any changes to CONFIG.SYS and AUTOEXEC.BAT which may be required (with your permission)

2. Running the DeCipher Plus Demonstration

The \DCP directory on the installation drive contains the demonstration program

```
DEMO.BAT
```

To run the demonstration, enter the DOS command

```
cd \DCP<Enter>
DEMO<Enter>
```

The DeCipher Plus demonstration will execute various application displays using data previously acquired already installed onto your computer.

3. Vital Hints for Getting Started

- * DeCipher Plus requires a very specific directory structure to be set up on disk in order for it to run. When this directory structure is not present DeCipher Plus will not run. This directory structure begins at the root directory level with the \DCP directory. The \DCP directory must always be at root directory level. By default the database is found on the drive on which DeCipher Plus is executing, although the DB command line parameter allows this to be modified.

* DeCipher Plus file connections can only be made to files with the extension DCP. These files are in a very special format and can only be generated through the File application of DeCipher Plus. The .DCP files are very different in format to any data saved in the top window of the Command application.

IT IS NOT POSSIBLE to connect, plot or translate any data saved using Alt-S in the Command application, nor is it possible to connect to any data saved in the \$Session.DAT file.

The correct sequence of actions to save data from a logger into a disk file, which can then be used by DeCipher Plus is as follows

- Connect to the logger
- Select the desired retrieval mode (real time or logged) in the data window.
- Select the desired channels in the File application
- Select DECIPHER format in the File application (indicated by *)
- Select a file name for the disk file (e.g. DATA)
- Use the Run menu to select and run the File application

As a result the file DATA.DCP will be created in the \DCP\DCP_DATA directory.

To access this data file, we must now create a connection to the file as follows

- Enter the Connect menu
- Highlight Filedata
- Press Alt N to create a new connection
- Select the Datafile type
- Tab to the Datafile tile
- Ensure that the path is set for \DCP\DCP_DATA
- Change the filename to DATA
- Tab to the top menu
- Press Alt-S to save
- Name the connection DATA

As a result a connection definition named DATA will now be listed in the connection list. Connect to the DATA connection as by pressing <Enter> or Alt C. The original data from the logger is now available for use within DeCipher Plus.

* Note that the files created by the File application are for IMPORTing into the spreadsheets only. They are NOT worksheet files.

- * The inbuilt context sensitive help system is extensive and is available throughout DeCipher Plus. It may prove useful to browse the help files before using the product. These files and other useful information can be accessed through the ? window. Simply highlight the desired subject and press <Enter>.

4. Known Software Problems with Version 1.1

- * DeCipher Plus does not exit gracefully to DOS if it cannot find a database structure with which to execute (either by default or as specified on the command line). Reboot the machine to restart DOS. Investigate why no database (\DCP directory structure) exists where DeCipher Plus was executed.
- * There are some memory problems associated with running DeCipher Plus under Windows as a DOS application. If the EMM386 expanded memory manager is being used with the NOEMS parameter, DeCipher Plus will crash on initialization. It is not compatible with this expanded memory arrangement. Run EMM386 without the NOEMS parameter.
- * We have had reports of DeCipher Plus crashes in the plot application connected to use of the axis changing keys and non-standard graphics hardware. As yet no problems have been reported with VGA systems but we have received one report each for Hercules and EGA. It should be stressed that in general DeCipher Plus does work with these graphics adaptors. In cases where the axis change facility causes problems, it should not be used. Check that there are no TSRs loaded which may be making sophisticated or unusual use of the graphics adaptor.
- * Some problems have been reported when a DeCipher Plus connection is made to a COM port on which a serial mouse driver is resident. Some mouse drivers have been found to interfere with DeCipher Plus, which causes various phenomena such as no communications when running applications. We have found this to be a problem especially with

Genius Mouse driver
Agiler Mouse driver

The Microsoft mouse driver seems to function correctly. As a rule don't install mouse drivers installed on any COM port which you intend to use to connect to loggers.

- * A small number of problems have been found with the saving of tasks
 - File application special date and time format are not saved (The first three for all file formats except DECIPHER).
Use Date and Time from logger channels.

- Tasks which include channels which have been deselected in the Data window will not load correctly. Ensure all deselection is managed within the individual application windows.
- * The Lotus default file formats are incorrect for date and time. You should select .dddd format for time and ddd elapsed format for date manually. Once in Lotus it is necessary to import these as numbers and to convert them to TIME_VALUE and DATE_VALUE with a formula. The TIME_VALUE converts directly but the DATE_VALUE must be adjusted by adding 32509 to the imported number (converts to the Lotus date offset from 01\01\1900).
- * It is possible to receive an alarm or an error message during the transmission of a response to DeCipher Plus program query. However this results in eventual loss of connection after a period displaying the Waiting window. Try using protocol mode if this is a problem. Otherwise just reconnect as this is only likely to be a problem in situations where large numbers of alarms or errors are being returned continuously or frequently.
- * On entering the File application after a task load, the file destination may be selected but the filename may not be visible. To display the filename simply toggle the file destination selection.
- * The Setup window always shows the default graphics printer after restart, even after another printer has been selected and saved. In fact the correct printer driver is active, the problem is that it has not been displayed correctly.
- * Creation of a connection file with the name NEW_CON prevents creation of any other connection. The connections may be defined but they cannot be saved. The NEW_CON.CON should be deleted from \DCP\CON_DB\CT. Unfortunately NEW_CON is the default connection name suggested on a new connection save.
- * The available data range for a connection which includes an internal memory card (as displayed in the data window) is not visible. The data still exists on the card, but DeCipher Plus cannot access its range. Any applications executed can define any time/date range for retrieved data as required without problem.
- * It is possible to select any format for date and time channels in the DECIPHER file format. However these have no effect, and will be forced to integer days and integer seconds due to the requirements for reconnection. This will be apparent by returning to the File application window.
- * The units for date and time channels are inconsistent between different applications. However this does not affect running of the applications

in any way.

- * Y axis labeling on some plots is not quite correct. On occasion the largest Y axis label is not displayed.
- * When connecting to a Dataloader using an address not directly attached to the COM port of the computer, the 'Connected to' message is issued once DeCipher Plus has contacted the network and NOT the logger itself. If no logger with the specified address exists on the network, the first communication to that logger will be the first failure. This occurs on an attempt to agree on the program map with the Waiting window displayed. A network error is displayed on the Alarm line. At this point use Alt R to release the connection.

5. Omissions from the Manual in version 1.1

Three sections are missing from the manual.

- 6.3.8 Setup Module
- 6.3.9 Help overview
- A2 System messages

Exiting DeCipher Plus should be section 6.3.10

Synopsis

6.3.8 Setup Module

The setup module allows definition and saving of the contents of the DCP_CFG.CFG file. All overall system parameters can be set here.

It also allows loading and saving of task files. A task file is a complete copy of all DeCipher Plus menu definitions. Loading a task file results in DeCipher Plus menus being configured to the contents of the task file.

Examples are provided in the demonstration.

- DEMO_1.DDT
- DEMO_2.DDT
- etc...

6.3.9 Help overview

All DeCipher Plus help files (and other useful text files provided by Data Electronics) can be browsed within the help overview menu. Simply highlight the desired file and press <Enter>.

The filename is provided on the right hand side of the list to allow

access to the text files for printing. The files can be found in
\\DCP\HELP.

The file KEY_HELP.HLP provides on line access to the manual section
6.2 Keyboard Usage.

6. What to do when you have Problems

- * Please check both the manual appendix and help files for possible solutions to the problem.
- * If that fails, fill out the problem report sheet as fully as possible. In particular describe the sequence of DeCipher Plus operations that highlighted the problem, and details of the Datataker program and configuration you were using. These are of paramount importance.
- * Fax the information to us as follows
 - Within Australia (03)-764-8997
 - Outside Australia 61-3-764-8997
- * For Telephone support you can contact us as follows
 - Within Australia (03)-764-8600
 - Outside Australia 61-3-764-8600

6. Revision History

24-April-1992	(JPM)	Original Version
01-July-1992	(JPM)	Upgrade Revision

Enjoy using DeCipher Plus!

DeTerminal Software, version 2., Science/Electronics Part #71025

Directory of A:\

```

BLOKSCAN CMD      1,098  09-16-93  2:46p
DEPOLY  EXE      37,254  06-25-93  11:07a
DOWNTIME CMD      2,732  09-16-93  2:46p
DT       CFG       374    03-13-94  11:14p
DT       EXE     292,984  03-11-94  3:28p
DT       HLP     61,964  03-13-94  11:07p
EXAMPLE  CMD      2,283  01-20-94  5:01p
FUELFLOW CMD      1,767  09-16-93  2:46p
LOGSCAN  CMD      1,247  09-16-93  2:46p
LSQ      DOC      2,176  04-14-88  2:31p
LSQ      EXE     85,585  04-19-88  4:16p
PROGRAM  CMD      1,138  11-09-93  4:24p
READ     ME       4,958  03-16-94  12:19p
README  BAT        44    02-09-94  12:07a
STEPSCAN CMD      1,355  09-16-93  2:46p
TEMPLATE CMD      1,218  11-09-93  4:23p
UNLD_100 CMD      1,050  03-16-94  11:55a
UNLD_200 CMD      1,348  03-16-94  11:56a
WIND_SET CMD      3,632  09-16-93  2:46p
  19 file(s)      504,207 bytes
   0 dir(s)       947,200 bytes free

```

"READ.ME" File Listing

DeTerminal Command Files

=====

Introduction

=====

DeTerminal is a terminal type program which provides an easy to use interface to the Datataker data loggers from IBM-PC and compatible computers.

DeTerminal is used to create and edit command files, to send commands and command files to the Datataker, and to view and save data returned by the Datataker.

This DeTerminal distribution disc contains the following files

```

DT.EXE      - DeTerminal program file
DT.HLP      - DeTerminal help file, accessed via F1
DT.CFG      - Last used configuration of the DeTerminal system

READ.ME     - This information file

```

- PROGRAM.CMD - A suggested layout for Datataker programs
 TEMPLAT.CMD - A template for Datataker programs, based on the suggested layout. A simple program can be quickly developed by entering details into the template.

Command Files

=====

The distribution disc also contains a number of Datataker command files which provide a guide to using DeTerminal to program the Datataker, and some useful applications programs, as follows

- BLOKSCAN.CMD - Demonstrates programming of the Datataker to scan input channels for short periods of time, at longer regular intervals of time.
- STEPSCAN.CMD - Demonstrates programming of the Datataker to scan input channels at increasing intervals as the data logging session proceeds. This technique can be used to collect data for a parameter which asymptotes with time, such that data is collected more slowly as the parameter changes more slowly.
- LOGSCAN.CMD - Demonstrates programming of the Datataker to scan input channels at logarithmic intervals as the data logging session proceeds. This technique can also be used to collect data for a parameter which asymptotes with time.
- DOWNTIME.CMD - Demonstrates programming of the Datataker to perform calculations based on time. Operator buttons are monitored to indicate reason for machine downtime, which provide triggers for measuring total elapsed time for each cause of downtime.
- FUELFLOW.CMD - Demonstrates the use of Datataker expressions to combine data from two related measurements, to provide corrected results in real time.
- WINDSET.CMD - Demonstrates use of Datataker expressions to calculate average wind direction and wind sigma from a wind direction and wind speed sensor.
- UNLD_100.CMD - Provides assistance in unloading the DT100 family of loggers, by stepping the user through the decisions to be made about data delimiters, time formats, and filenames. The logger is then set as desired and unloaded automatically.

Note that UNLD_100.CMD is supplied for DT100 users as DeTerminal's Alt Unload feature is specifically for supporting DT500's and DT50's.

UNLD_200.CMD - Provides assistance in unloading the DT200 family of loggers, by stepping the user through the decisions to be made about data delimiters, time and date formats, and filenames. The logger is then set as desired and unloaded automatically.

Note that UNLD_200.CMD is supplied for DT200 users as DeTerminal's Alt Unload feature is specifically for supporting DT500's and DT50's.

Utilities

=====

There are also two utility programs on the distribution disc, for extracting polynomials from data sets of sensor calibrations. These polynomials can then be programmed into the Datataker to return data in engineering units from sensors with simple analog or digital outputs.

DEPOLY.EXE - Calculates polynomials mathematically from selected data points from a data set.

LSQ.EXE - Calculates polynomials for data sets using the least squares method.

DISTRIBUTION SHEET

To	From	Page 1 of 1
Distribution	SESC / Laboratory Services	Date October 31, 1997
Project Title/Work Order		EDT No.
System Design Description for the SY-101 Hydrogen Mitigation Test Project Data Acquisition and Control System (DACS-1) / NF22Y		ECN No. 643391

Name	MSIN	Text With All Attach.	Text Only	Attach./ Appendix Only	EDT/ECN Only
W. G. Brown	T4-07	X			
D. W. Crass	H6-11				X
J. A. Ellingsworth	R2-87	X			
A. M. Ermi (5)	L6-37	X			
G. J. Gauck	T4-07	X			
J. C. Krogness	L6-38				X
L. S. Krogsrud	T4-07	X			
D. C. Larsen	T4-08	X			
D. D. Tate	L6-37				X
R. W. Truitt	L6-37	X			
R. R. True	T4-07	X			
R. P. Tucker	T4-08				X
S. O. Smith	L6-37	X			
DACS Project File	L6-37	X			
Central Files	B1-07	X			