HF CONTROLS CONTROL SYSTEM

HFC-6000 Product Line Components

TSAP
Requirements Specification

700901-09

Rev A
August 7, 2003

Author JONATHAN TAYLOR

Reviewer JOHN STEVENSON

Approval ALLEN HSU

[ ]

Copyright© 2003 HF Controls Corporation

Revision History

Date	Revision	Author	Changes
8/03	A	J Taylor	Original document divided into separate requirements documents for each major component.

Table of Contents

Section	Title	Page
1.0	SCOPE	4
2.0	PURPOSE	4
3.0	REFERENCES	4
3.1	Related Publications	4
3.2	Terms and Acronyms	4
4.0	FUNCTIONAL ANALYSIS	5
4.1	General Requirements	5
4.2	Input Requirements	5
4.3	Processing Requirements	6
4.3.1	Accuracy Test Support	7
4.3.2	Response Time Tests	8
4.3.3	Discrete Operability Tests	9
4.3.4	Timer Test	9
4.3.5	Prudency Burst of Events Test	9
4.3.6	Heartbeat Monitor	10
4.3.7	Simulated Control Loops	10
4.4	Output Requirements	11
4.5	Conceptual Module Description	11
4.6	Assumptions	11
4.7	Critical Issues	12

List of Figures

Number	Title	Page
Figure 1.	Analog Accuracy Test Waveform	7
Figure 2.	Digital Response Time Test Algorithm	8
Figure 3.	Analog Response Time Test Algorithm	8
Figure 4.	Timer Test Algorithm	9
Figure 5.	TSAP DO Control Algorithm for Burst of Events	10

1.0

SCOPE

This document covers Test Application Software (TSAP) installed in the HFC-6000 controller Test Specimen during qualification testing. The Test Specimen will be a modular, fully functional control system composed of a single redundant remote, I/O modules, and ancillary equipment.

2.0 PURPOSE

The TSAP will operate as an application program of the main processor on the controller card. It will accept operator inputs from control switch modules (CSM), M/A stations, a flat panel display (FPD), and a JCRT PC workstation. It will include both control logic for simulated functional control loops and support for prudency and operability tests that are designed to exercise control system logic functions. The TSAP will use data inputs from configured AI and DI channels to update values for AO and DO points as well as alarm status. The TSAP retrieves all input data from memory on the HFC-SBC06 controller and transfers all output values to memory during each processing cycle. Utilities for running the I/O scan and transferring status to appropriate displays are external to the TSAP.

3.0 REFERENCES

3.1 RELATED PUBLICATIONS

EPRI TR-107330 Generic Requirements Specification for Qualifying a Commercially Available PLC for safety-Related Applications in Nuclear Power Plants

700901-01	HFC-6000 Product Line Specification
TN0401	Master Test Plan
TP0402	Operability Test Procedure
TP0403	Prudency Test Procedure

3.2 TERMS AND ACRONYMS

AI	Analog Input
AO	Analog Output
CSM	Control Switch Module
DI	Digital Input
DO	Digital Output
EWS	Engineering Workstation
FL	Flag
HPAT	HFC Plant Automated Tester
I/O	Input/Output
JCRT	Java CRT Workstation
M/A	Manual/Automatic
MS	Message
TI	Timer

4.0 FUNCTIONAL ANALYSIS

The following paragraphs define the functional requirements of the TSAP to be installed in the HFC-6000 Test Specimen.

4.1 GENERAL REQUIREMENTS

[

7.

]

4.2 INPUT REQUIREMENTS

The following inputs correspond to the AI, DI, and pulse input signals, which will be generated by the HPAT hardware. Most input channels used for operability and

prudency testing will be connected directly to corresponding output points without intermediate gating. Inputs for the simulated control loops will be connected to the logic algorithms for those loops.

[

5.

]

4.3 PROCESSING REQUIREMENTS

The TSAP will provide logic to perform the following functions.

- Read inputs associated with the operability and prudency tests.
- Drive outputs required by the operability and prudency tests.
- Provide logic for timer testing.
- Provide algorithms for the simulated control loops.

The following paragraphs provide a detailed list of requirements to ensure adequate support for the prudency and operability tests.

4.3.1 Accuracy Test Support

Refer to TP0402 for a detailed description of accuracy testing run as part of the operability group of tests. The following are the specific test support requirements for the TSAP.

[

]

Figure 1. Analog Accuracy Test Waveform

4.3.2 Response Time Tests

The TSAP shall provide separate algorithms to support both digital and analog response time measurements. Refer to TP0402 paragraph 3.3.3 for detailed information about the operability response time tests.

[

2.

]

Figure 2. Digital Response Time Test Algorithm

Figure 3. Analog Response Time Test Algorithm

4.3.3 Discrete Operability Tests

No special logic support will be required for the discrete input and discrete output operability tests. (Refer to TP0402 paragraphs 3.3.4 and 3.3.5.)

4.3.4 Timer Test

The TSAP shall provide the algorithm required to support the operability timer test. Refer to TP0402 paragraph 3.3.7 for detailed information about the timer test.

[

2.

]

Figure 4. Timer Test Algorithm

4.3.5 Prudency Burst of Events Test

The prudency burst of events test exercises a significant number of channels simultaneously. The external HPAT controls the AI and DI signals supplied to the input interface of the Test Specimen, and the TSAP provides analog and digital algorithms to control the corresponding output channels.

[

3.

]

Figure 5. TSAP DO Control Algorithm for Burst of Events

4.3.6 Heartbeat Monitor

The TSAP shall contain a dedicated counter configured to increment by one count during each execution cycle of the equation interpreter. This counter will provide a direct indication that the controller is executing the application code.

4.3.7 Simulated Control Loops

The TSAP will provide logic for a minimum of 10 (five analog and five digital) simulated control loops. Each control loop will receive simulated inputs from the HPAT and execute a control algorithm appropriate for the simulated field device under control. The resulting outputs will control output signals to the simulated process as well as status and alarm signals. Minimum requirements for the simulated control loops are as follows.

[

7.

]

4.4 OUTPUT REQUIREMENTS

The following outputs correspond to the AO and DO signals, which will be generated by the TSAP. Most output channels used for operability and prudency testing will be connected directly to corresponding input points of the HPAT for logging purposes. Output signals for the simulated control loops will be connected to the simulation algorithms in the HPAT for field devices or processes.

[

4.

]

4.5 CONCEPTUAL MODULE DESCRIPTION

[

]

4.6 ASSUMPTIONS

The design requirements for the TSAP of the Test Specimen is based on the following assumptions:

[

•

]

4.7 CRITICAL ISSUES

None

ATTACHMENT 7.1

COMMERCIAL GRADE SOFTWARE EVALUATION

EVALUATION NUMBER: CGSE002

REVISION: A

Safety Related: [X] YES [ ] NO

MANUFACTURER: HF Controls

DESCRIPTION: HFC-SBC06 Main Processor System Firmware

SOFTWARE NUMBER: 9120905-10 Rev: CKSM 147C

Assigned Engineer: Jonathan Taylor Date: 12/15/05

Technical Reviewer: Steve Yang Date: 12/15/05

Approved By: Allen Hsu Date: 12/15/05

V & V Review: Bob Cain Date: 12/15/05

COMMERCIAL GRADE SOFTWARE EVALUATION

Software P/N: 9120905-10
CGS CLASSIFICATION SHEET

CGSE002, Rev A

Rev: CKSM 147C Description: Main Processor Firmware

1. What is the safety function of the parent component in which this item is to be installed?

   Controls safety system operation.

2. What is the function of the item?

   Main processor initialization.
   Overall status monitoring and control.
   Runs system application code.
   Primary executes status drop to secondary controller.
   Secondary receives status drop from primary.

3. What are the ways in which the item should fail in actual service?

   Program error causes processor stall or invalid branching operation.
   Program error produces invalid data image.
   Program error disables required function.
   Heavy loading results in failure to execute utility task.

4. Would any of these postulated failures of the item prevent the parent component or associated components from performing their intended safety function?

   Incorrect program branching without processor stall could prevent the safety system from performing its intended function.
   Generation of invalid data image could result in incorrect or unexpected control of an output channel.
   Secondary processor stall without failover could disable the corresponding portion of the safety system.

COMMERCIAL GRADE SOFTWARE EVALUATION

Software P/N: 9120905-10
CGS CLASSIFICATION SHEET

CGSE002, Rev A

Rev: CKSM 147C Description: Main Processor Firmware

Yes [X] (Item has safety function)

No [ ] (Item has non-safety function)

Method (Mark all that apply}

A "YES" in all three criteria is required for the item to be classified as commercial grade.

		YES	NO
1	Is the item not subject to design or specification requirements that are unique to facilities or activities licensed or regulated by the USNRC?	[X]	[ ]
2	Is the item used in applications other than facilities or activities licensed or regulated by the USNRC?	[X]	[ ]
3	Is the item ordered from manufacturer/supplier on the basis of specifications set forth in the manufacturer's published description?	[X]	[ ]
	Commercial Grade Software Item	[X]	[ ]

PHYSICAL CRITICAL CHARACTERISTICS:

Source code files
Compiled object code designation
Module checksum generated during compilation and when PROM is encoded.

COMMERCIAL GRADE SOFTWARE EVALUATION

Software P/N: 9120905-10
CGS CLASSIFICATION SHEET

CGSE002, Rev A

Rev: CKSM 147C Description: Main Processor Firmware

PERFORMANCE CRITICAL CHARACTERISTICS:

Initialization control
System status monitoring and display
Hardware watchdog control function
[
Application functional processing
[
]
Performance under heavy loading conditions

DEPENDABILITY CRITICAL CHARACTERISTICS:

Built-in quality
Configuration management
Operating history
Problem reporting and resolution

METHOD (MARK ALL THAT APPLY):

[X] Method 1 - Special Tests and Inspections.

[X] Method 2 - Design Documentation and/or Commercial Grade survey of vendor.

[ ] Method 3 - Source Verification.

[X] Method 4 - Historical Performance Records.

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1

CGS CRITICAL CHARACTERISTICS SHEET

Software P/N: 9120905-10 **Description:** SC Firmware **CGSE** 002 **Rev:** A

Physical Critical Characteristic	Acceptance Criteria	Method of Verification
CC#001 Source code files	Source files are controlled with limited access. Revision record is included in the header of each source file. Source code reviews generated for critical source code files. Module-level test of PDS files.	Direct examination of source files Source code review documentation Source code test reports Prototype test report
CC#002 Compiled object code designation	Object code file designated as 912090510.hex	Direct verification
CC#003 Module checksum and PROM part number	One byte compiler checksum value (32H) is inserted as the first byte of the object code header. Two byte object code checksum (147CH) calculated at the time the PROM is programmed. Checksum value and PROM part number are printed on a label, and that label is affixed to the outside of the PROM.	Checksum 147C is based on the SC program code with a generic database configuration. Changing the database size will result in a different checksum value and a different part number. Physical verification of U52 label on assembly. Direct verification of one-byte checksum value from PROM header.

Performance Critical Characteristic	Acceptance Criteria	Method of Verification
CC#004 Initialization control	Runs initialization tests per design specification to verify operability status	Source code review. Source code test reports

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1

CGS CRITICAL CHARACTERISTICS SHEET

Software P/N: 9120905-10 **Description:** SC Firmware **CGSE** 002 **Rev:** A

Performance Critical Characteristic	Acceptance Criteria	Method of Verification
CC#005 System status monitoring and display	Detects any failure during initialization sequence and displays failure code Detects run-time failure and displays failure code. Displays tasking code throughout normal operation.	Prototype test report UG004-000-08
CC#006 Hardware/software watchdog control function	Per design specification, as revised for 0.5 second failover.	TS901-000-42 TR901-000-02
CC#007 Support for redundant controller configuration	Per design specification.	Source code review. Source code test reports Functional test.
CC#008 Support for application processing	Supports standard Boolean functions Supports programmable timer function Supports block algorithms for analog processing	Application Objects Test Report TR001-000-02 DS001-000-003
CC#009 [ ]	Per design specification.	Prototype test report
CC#010 [•]	Per design specification.	Prototype test report

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1

CGS CRITICAL CHARACTERISTICS SHEET

Software P/N: 9120905-10 **Description:** SC Firmware **CGSE** 002 **Rev:** A

Performance Critical Characteristic	**Acceptance Criteria**	**Method of Verification**
CC#011 Performance under abnormal loading conditions	[ ]	[ ]
Dependability Critical Characteristic	**Acceptance Criteria**	**Method of Verification**
CC#012 Built-in quality	NQA1 QA program	Records of internal and external audits Source code review records Source code test reports Prototype test report
CC#013 Configuration control	Source files are retained in CMS library with limited access; current object file is retained in the assembly release directory and matches BOM	Direct inspection
CC#014 Operating history	Revision record in file header indicates up to 20 years operating history, including the past several years an Ulchin NPP.	Direct inspection of source files CR and SCR database Cumulative operating history of generic operating system
CC#015 Reporting and problem resolution	Corrective action program in place and being used for evaluating and resolving reported problems. HFC supports CFR Part 21 reporting	QPP 16.1 and CR/SCR database Record of reported errors for generic operating system QPP 16.3

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1

CGS CRITICAL CHARACTERISTICS SHEET

Software P/N: 9120905-10 **Description:** SC Firmware **CGSE** 002 **Rev:** A

M&TE CIN#	M&TE Desc.	Cal. Due Date	M&TE CIN#	M&TE Desc.	Cal. Due Date

Condition Report (CR): [ ] Yes [ ] No [ ] N/A If yes, identify below:
CR# [ ]Acceptable [ ] Not Acceptable Initials/Date: _____
CR# [ ]Acceptable [ ] Not Acceptable Initials/Date: _____
CR# [ ]Acceptable [ ] Not Acceptable Initials/Date: _____
Other:

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1

CGS CRITICAL CHARACTERISTICS SHEET

Software P/N: 9120905-10 **Description:** SC Firmware **CGSE** 002 **Rev:** A

Sheet ____ **of** ____

CC#	P/F/N	Preformed By	Date	Verified By	Date
001					
002					
003					
004					
004					
006					
007					
008					
009					
010					
011					
012					
013					
014					
015					

P = Pass, F = Fail, N = Not Applicable

Characteristics Record Sheet Competition:

Verified By _____ **Approved By** _____

Date:_____ **Date:**_____

ATTACHMENT 7.1

COMMERCIAL GRADE SOFTWARE EVALUATION

EVALUATION NUMBER: CGSE004

REVISION: A

Safety Related: [ ] YES [X] NO

MANUFACTURER: HF Controls

DESCRIPTION: HFC-SBC06 SEP System Firmware

SOFTWARE NUMBER: 9120907-11 Rev: CKSM 6FF8

Assigned Engineer: Jonathan Taylor Date: 12/15/05

Technical Reviewer: Steve Yang Date: 12/16/05

Approved By: Allen Hsu Date: 12/16/05

V & V Review: Bob Cain Date: 12/16/05

COMMERCIAL GRADE SOFTWARE EVALUATION

Software P/N: 9120907-11
CGS CLASSIFICATION SHEET

CGSE004, Rev A

Rev: CKSM 6FF8 Description: SEP Firmware

1. What is the safety function of the parent component in which this item is to be installed?

   Controls safety system operation.

2. What is the function of the item?

   SEP processor initialization.

   [.

   ]

3. What are the ways in which the item should fail in actual service?

   Program error causes processor stall or invalid branching operation.
   Program error results in communication error.
   Program error results in corruption of data.

4. Would any of these postulated failures of the item prevent the parent component or associated components from performing their intended safety function?

   No. C-Link is used for transfer of status data only.

COMMERCIAL GRADE SOFTWARE EVALUATION

Software P/N: 9120907-11
CGS CLASSIFICATION SHEET

CGSE004, Rev A

Rev: CKSM 6FF8 Description: SEP Firmware

Yes [ ] (Item has safety function)

No [X] (Item has non-safety function)

Method (Mark all that apply}

A "YES" in all three criteria is required for the item to be classified as commercial grade.

		YES	NO
1	Is the item not subject to design or specification requirements that are unique to facilities or activities licensed or regulated by the USNRC?	[X]	[ ]
2	Is the item used in applications other than facilities or activities licensed or regulated by the USNRC?	[X]	[ ]
3	Is the item ordered from manufacturer/supplier on the basis of specifications set forth in the manufacturer's published description?	[X]	[ ]
	Commercial Grade Software Item	[X]	[ ]

COMMERCIAL GRADE SOFTWARE EVALUATION

Software P/N: 9120907-11
CGS CLASSIFICATION SHEET

CGSE004, Rev A
Rev: CKSM 6FF8 Description: SEP Firmware

PHYSICAL CRITICAL CHARACTERISTICS:

N/A

PERFORMANCE CRITICAL CHARACTERISTICS:

N/A

DEPENDABILITY CRITICAL CHARACTERISTICS:

N/A

METHOD (MARK ALL THAT APPLY):

[ ] Method 1 - Special Tests and Inspections.

[ ] Method 2 - Design Documentation and/or Commercial Grade survey of vendor.

[ ] Method 3 - Source Verification.

[ ] Method 4 - Historical Performance Records.

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1
CGS CRITICAL CHARACTERISTICS SHEET

Software P/N: 9120907-11 **Description:** SEP Firmware **CGSE** 004 **Rev:** A

Physical Critical Characteristic	Acceptance Criteria	Method of Verification
N/A		

ATTACHMENT 7.1
CGS CRITICAL CHARACTERISTICS SHEET

Software P/N: 9120907-11 **Description:** SEP Firmware **CGSE** 004 **Rev:** A

Performance Critical Characteristic	Acceptance Criteria	Method of Verification
N/A		

ATTACHMENT 7.1
CGS CRITICAL CHARACTERISTICS SHEET

Software P/N: 9120907-11 **Description:** SEP Firmware **CGSE** 004 **Rev:** A

Dependability Critical Characteristic	Acceptance Criteria	Method of Verification
N/A		

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1
CGS M&TE SHEET

Software P/N: 9120907-11 **Description:** SEP Firmware **CGSE** 004 **Rev:** A

Record all M&TE used for this test:

M&TE CIN#	M&TE Desc.	Cal. Due Date	M&TE CIN#	M&TE Desc.	Cal. Due Date

Condition Report (CR): [ ] Yes [ ] No [ ] N/A If yes, identify below:
CR# [ ]Acceptable [ ] Not Acceptable Initials/Date: _____
CR# [ ]Acceptable [ ] Not Acceptable Initials/Date: _____
CR# [ ]Acceptable [ ] Not Acceptable Initials/Date: _____
Other:

ATTACHMENT 7.1
CGS CRITICAL CHARACTERISTICS RECORD SHEET

Software P/N: 9120907-11 **Description:** SEP Firmware **CGSE** 004 **Rev** A

Sheet ___ **of** ___

N					

P = Pass, F = Fail, N = Not Applicable

Characteristics Record Sheet Competition:

Verified By _____ **Approved By** _____
Date:_____ **Date:**_____

ATTACHMENT 7.1

COMMERCIAL GRADE SOFTWARE EVALUATION

EVALUATION NUMBER: CGSE006

REVISION: A

Safety Related: [Y] YES [ ] NO

MANUFACTURER: HF Controls

DESCRIPTION: HFC-AI4K Firmware

SOFTWARE NUMBER: 9120683-12 Rev: CKSM 3B01

Assigned Engineer: Jonathan Taylor Date: 12/15/05

Technical Reviewer: Steve Yang Date: 12/16/05

Approved By: Allen Hsu Date: 12/15/05

V & V Review: Bob Cain Date: 12/16/05

INFORMATIONAL

COMMERCIAL GRADE SOFTWARE EVALUATION

Software P/N: 9120683-12
CGS CLASSIFICATION SHEET

CGSE006, Rev A

Rev: CKSM 3B01 Description: HFC-AI4K Firmware

1. What is the safety function of the parent component in which this item is to be installed?

   Controls safety system operation.

2. What is the function of the item?

   Controls internal operation of 4-channel pulse input module.

3. What are the ways in which the item should fail in actual service?

   Program error causes processor stall or invalid branching operation.
   Program error results in corruption of data images.
   Program error results in loss of communication with controller.

4. Would any of these postulated failures of the item prevent the parent component or associated components from performing their intended safety function?

   Corruption of input image (either accumulated count or rate value) results in invalid control of field process.
   Loss of communication with controller or loss of image data prevents reliable control of field process.

   Yes [X] (Item has safety function)

   No [ ] (Item has non-safety function)

COMMERCIAL GRADE SOFTWARE EVALUATION

Software P/N: 9120683-12
CGS CLASSIFICATION SHEET

CGSE006, Rev A

Rev: CKSM 3B01 Description: HFC-AI4K Firmware

Method (Mark all that apply}

A "YES" in all three criteria is required for the item to be classified as commercial grade.

		YES	NO
1	Is the item not subject to design or specification requirements that are unique to facilities or activities licensed or regulated by the USNRC?	[X]	[ ]
2	Is the item used in applications other than facilities or activities licensed or regulated by the USNRC?	[X]	[ ]
3	Is the item ordered from manufacturer/supplier on the basis of specifications set forth in the manufacturer's published description?	[X]	[ ]
	Commercial Grade Software Item	[X]	[ ]

PHYSICAL CRITICAL CHARACTERISTICS:

Source code files
Compiled object code designation
Module checksum generated at the time that the PROM is encoded.

PERFORMANCE CRITICAL CHARACTERISTICS:

Initialization control
Internal status monitoring
Saves pulse channel configuration parameters in non-volatile RAM.
Hardware watchdog strobe function
Input scan cycle control
Input count value processing
Response to messages from the HFC-6000 controller.

COMMERCIAL GRADE SOFTWARE EVALUATION

Software P/N: 9120683-12
CGS CLASSIFICATION SHEET

CGSE006, Rev A

Rev: CKSM 3B01 Description: HFC-AI4K Firmware

DEPENDABILITY CRITICAL CHARACTERISTICS:

Built-in quality
Configuration control
Operating history
Problem reporting and resolution

METHOD (MARK ALL THAT APPLY):

[X] Method 1 - Special Tests and Inspections.

[X] Method 2 - Design Documentation and/or Commercial Grade survey of vendor.

[ ] Method 3 - Source Verification.

[X] Method 4 - Historical Performance Records.

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1
CGS CRITICAL CHARACTERISTICS SHEET

Software P/N: 9120683-12 **Description:** HFC-AI4K Firmware CGSE 006 **Rev:** A

Physical Critical Characteristic	Acceptance Criteria	Method of Verification
CC#001 Source Code Files	[ ]. Revision record is included in the header section of each source code file. Source code files are under configuration control. Source code reviews generated for critical source code files.	Direct examination of source files Source code review documentation
CC#002 Compiled object code designation	Object code file is designated 912068312.hex.	Direct verification
CC#003 Module checksum and PROM part number.	Two-byte checksum (3B01) calculated at the time that the PROM is programmed. PROM part number and checksum value are printed on a label, and this label is attached to the PROM.	Checksum 3B01 is generated by ChipWin program at the time the PROM is encoded. Physical verification of label on U9.

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1
CGS CRITICAL CHARACTERISTICS SHEET

Software P/N: 9120683-12 **Description:** HFC-AI4K Firmware CGSE 006 **Rev:** A

Performance Critical Characteristic	Acceptance Criteria	Method of Verification
CC#004 Initialization control	Runs initialization tests per design specification to verify operability.	Source code review Prototype test report
CC#005 Internal status monitoring function	Maintains count of communication errors. Verifies that valid configuration data is available. Detects loss of ICL communication and enables hunt mode.	Source code review Prototype test report
CC#006 Saves pulse channel configuration data in non-volatile RAM	Per design specification and maintenance manual documentation.	Source code review Prototype test report
CC#007 Controls watchdog strobe	Per design specification and maintenance manual documentation.	Source code review Prototype test report
CC#008 Input scan function	Per design specification	Source code review Prototype test report
CC#009 Input count value processing function	Processing algorithm uses configuration parameters to produce rate or count value image per design specification	Source code review Prototype test report
CC#010 ICL response function	Per design specification and maintenance manual documentation.	Source code review Prototype test report

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1
CGS CRITICAL CHARACTERISTICS SHEET

Software P/N: 9120683-12 **Description:** HFC-AI4K Firmware **CGSE** 006 **Rev:** A

Dependability Critical Characteristic	Acceptance Criteria	Method of Verification
CC#011 Built-in quality	NQA1 QA Program	Records of internal and external audits. Software review records. Prototype test records.
CC#012 Configuration control	[	Direct inspection
CC#013 Operating history	Header in source files provide revision record; software used for more than 2 years at Ulchin NPP.	Direct inspection
CC#014 Problem reporting and resolution	Corrective action process is in place and being used for evaluating and resolving reported problems. HFC supports CFR Part 21 reporting.	QPP 16.1 and CR/SCR database QPP 16.3

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1
CGS M&TE SHEET

Software P/N: 9120683-12 **Description:** HFC-AI4K Firmware CGSE 006 **Rev:** A

Record all M&TE used for this test:

M&TE CIN#	M&TE Desc.	Cal. Due Date	M&TE CIN#	M&TE Desc.	Cal. Due Date

Condition Report (CR): [ ] Yes [ ] No [ ] N/A If yes, identify below:

CR# [ ]Acceptable [ ] Not Acceptable Initials/Date: _____

CR# [ ]Acceptable [ ] Not Acceptable Initials/Date: _____

CR# [ ]Acceptable [ ] Not Acceptable Initials/Date: _____

Other:

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1

CGS CRITICAL CHARACTERISTICS RECORD SHEET

Software P/N: 9120683-12 **Description:** HFC-AI4K Firmware CGSE 006 **Rev:** A

Sheet ____ **of** ____

CC#	P/F/N	Performed By	Date	Verified By	Date
001					
002					
003					
004					
005					
006					
007					
008					
009					
010					
011					
012					
013					
014					

P = Pass, F = Fail, N = Not Applicable

Characteristics Record Sheet Competition:

Verified By _____ **Approved By** _____

Date:_____ **Date:**_____

ATTACHMENT 7.1

COMMERCIAL GRADE SOFTWARE EVALUATION

EVALUATION NUMBER: CGSE008

REVISION: A

Safety Related: [Y] YES [ ] NO

MANUFACTURER: HF Controls

DESCRIPTION: HFC-AO8F Firmware

SOFTWARE NUMBER: 9120679-16 Rev: CKSM 87C6

Assigned Engineer: Jonathan Taylor Date: 12/15/05

Technical Reviewer: Steve Yang Date: 12/15/05

Approved By: Allen Hsu Date: 12/15/05

V & V Review: Bob Cain Date: 12/16/05

COMMERCIAL GRADE SOFTWARE EVALUATION

Software P/N: 9120679-16
CGS CLASSIFICATION SHEET

CGSE008, Rev A

Rev: CKSM 87C6 Description: HFC-AO8F Firmware

1. What is the safety function of the parent component in which this item is to be installed?

   Controls safety system operation.

2. What is the function of the item?

   Controls internal operation of 8-channel 4- to 20-mA analog output module.

3. What are the ways in which the item should fail in actual service?

   Program error causes processor stall or invalid branching operation.

   Program error prevents transfer of image data to output channel.

   Program error results in loss of communication with controller.

4. Would any of these postulated failures of the item prevent the parent component or associated components from performing their intended safety function?

   Failure to update one or more AO channels disables control of field process.

   Loss of communication with controller or loss of image data prevents reliable control of field process.

   Yes [X] (Item has safety function)

   No [ ] (Item has non-safety function)

COMMERCIAL GRADE SOFTWARE EVALUATION

Software P/N: 9120679-16
CGS CLASSIFICATION SHEET

CGSE008, Rev A

Rev: CKSM 87C6 Description: HFC-AO8F Firmware

Method (Mark all that apply}

A "YES" in all three criteria is required for the item to be classified as commercial grade.

		YES	NO
1	Is the item not subject to design or specification requirements that are unique to facilities or activities licensed or regulated by the USNRC?	[X]	[ ]
2	Is the item used in applications other than facilities or activities licensed or regulated by the USNRC?	[X]	[ ]
3	Is the item ordered from manufacturer/supplier on the basis of specifications set forth in the manufacturer's published description?	[X]	[ ]
	Commercial Grade Software Item	[X]	[ ]

PHYSICAL CRITICAL CHARACTERISTICS:

Source code files
Compiled object code designation
Module checksum generated at the time that the PROM is encoded.

PERFORMANCE CRITICAL CHARACTERISTICS:

Initialization control
Internal status monitoring
Responds to messages from the HFC-6000 controller.
Controls AO scan function.
Hardware watchdog strobe function

COMMERCIAL GRADE SOFTWARE EVALUATION

Software P/N: 9120679-16
CGS CLASSIFICATION SHEET

CGSE008, Rev A

Rev: CKSM 87C6 Description: HFC-AO8F Firmware

DEPENDABILITY CRITICAL CHARACTERISTICS:

Built-in quality
Configuration control
Operating history
Problem reporting and resolution

METHOD (MARK ALL THAT APPLY):

[X] Method 1 - Special Tests and Inspections.

[X] Method 2 - Design Documentation and/or Commercial Grade survey of vendor.

[ ] Method 3 - Source Verification.

[X] Method 4 - Historical Performance Records.

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1

CGS CRITICAL CHARACTERISTICS SHEET

Software P/N: 9120679-16 Description: HFC-AO8F Firmware CGSE 008 Rev: A

Physical Critical Characteristic	Acceptance Criteria	Method of Verification
CC#001 Source Code Files	[Revision record is included in the header section of each source code file. Source code files are under configuration control. Source code reviews generated for critical source code files.	Direct examination of source file Source code review documentation
CC#002 Compiled object code designation	Object code file is designated 912067916.hex.	Direct verification
CC#003 Module checksum and PROM part number.	Two-byte checksum (87C6) calculated at the time that the PROM is programmed. PROM part number and checksum value are printed on a label, and this label is attached to the PROM.	Checksum 87C6 is generated by ChipWin program at the time the PROM is encoded. Physical verification of label on U5.

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1

CGS CRITICAL CHARACTERISTICS SHEET

Software P/N: 9120679-16 **Description:** HFC-AO8F Firmware CGSE 008 **Rev:** A

Performance Critical Characteristic	Acceptance Criteria	Method of Verification
CC#004 Initialization control	Runs initialization tests per design specification to verify operability.	Source code review Prototype test report
CC#005 Internal status monitoring	Maintains count of communication errors. Records module failsafe configuration. [ ]	Source code review Prototype test report
CC#006 ICL response function	Responds to defined ICL messages per design specification.	Source code review Prototype test report
CC#007 Controls AO scan function	Per design specification	Source code review Prototype test report
CC#008 Controls watchdog strobe	Per design specification	Source code review Prototype test report

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1

CGS CRITICAL CHARACTERISTICS SHEET

Software P/N: 9120679-16 Description: HFC-AO8F Firmware CGSE 008 Rev: A

Dependability Critical Characteristic	Acceptance Criteria	Method of Verification
CC#009 Built-in quality	NQA1 QA Program	Records of internal and external audits. Software review records. Prototype test records.
CC#010 Configuration control	Source files are retained in source save; current object file retained in the assembly release directory and matches BOM	Direct inspection
CC#011 Operating History	Revision record in file header indicates up to 15 years operating history; including	Direct inspection CR and SCR database
CC#012 Problem reporting and resolution	Corrective action process is in place and being used for evaluating and resolving reported problems HFC supports CFR part 21 reporting	QPP 16.1 and CR/SCR database QPP 16.3

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1
CGS M&TE SHEET

Software P/N: 9120679-16 **Description:** HFC-AO8F Firmware CGSE 008 **Rev:** A

Record all M&TE used for this test:

M&TE CIN#	M&TE Desc.	Cal. Due Date	M&TE CIN#	M&TE Desc.	Cal. Due Date

Condition Report (CR): [ ] Yes [ ] No [ ] N/A If yes, identify below:
CR# [ ]Acceptable [ ] Not Acceptable Initials/Date: _____
CR# [ ]Acceptable [ ] Not Acceptable Initials/Date: _____
CR# [ ]Acceptable [ ] Not Acceptable Initials/Date: _____
Other:

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1

CGS CRITICAL CHARACTERISTICS RECORD SHEET

Software P/N: 9120679-16 **Description:** HFC-AO8F Firmware CGSE 008 **Rev:** A

Sheet ___ **of** ___

CC#	P/F/N	Performed By	Date	Verified By	Date
001					
002					
003					
004					
005					
006					
007					
008					
009					
010					
011					
012					

P = Pass, F = Fail, N = Not Applicable

Characteristics Record Sheet Competition:

Verified By _____ **Approved By** _____

Date:_____ **Date:**_____

ATTACHMENT 7.1

HFC

COMMERCIAL GRADE SOFTWARE EVALUATION

EVALUATION NUMBER: CGSE010

REVISION: A

Safety Related: [Y] YES [ ] NO

MANUFACTURER: HF Controls

DESCRIPTION: HFC-DC34 Firmware

SOFTWARE NUMBER: 9120678-14 Rev: CKSM 2164

Assigned Engineer: Jonathan Taylor Date: 12/15/05

Technical Reviewer: Steve Yang Date: 12/16/05

Approved By: Allen Hsu Date: 12/16/05

V & V Review: Bob Cain Date: 12/16/05

INFORMATIONAL ONLY

COMMERCIAL GRADE SOFTWARE EVALUATION

Software P/N: 9120678-14
CGS CLASSIFICATION SHEET

CGSE010, Rev A

Rev: CKSM 2164 Description: HFC-DC34 Firmware

1. What is the safety function of the parent component in which this item is to be installed?

   Controls safety system operation.

2. What is the function of the item?

   Controls internal operation of 12-channel DI, 2-channel 125-vdc DO module with coil continuity monitoring.

3. What are the ways in which the item should fail in actual service?

   Program error causes processor stall or invalid branching operation.

   Program error prevents transfer of image data to output channels or corrupts DI image.

   Program error results in loss of communication with controller.

4. Would any of these postulated failures of the item prevent the parent component or associated components from performing their intended safety function?

   Failure to update DO channels prevents reilable control of field process.

   Loss of DI data prevents reliable status monitoring of field process.

   Loss of communication with controller or loss of image data prevents reliable control of field process.

Yes [X] (Item has safety function)

No [ ] (Item has non-safety function)

COMMERCIAL GRADE SOFTWARE EVALUATION

Software P/N: 9120678-14
CGS CLASSIFICATION SHEET

CGSE010, Rev A

Rev: CKSM 2164 Description: HFC-DC34 Firmware

Method (Mark all that apply)

A "YES" in all three criteria is required for the item to be classified as commercial grade.

		YES	NO
1	Is the item not subject to design or specification requirements that are unique to facilities or activities licensed or regulated by the USNRC?	[X]	[ ]
2	Is the item used in applications other than facilities or activities licensed or regulated by the USNRC?	[X]	[ ]
3	Is the item ordered from manufacturer/supplier on the basis of specifications set forth in the manufacturer's published description?	[X]	[ ]
	Commercial Grade Software Item	[X]	[ ]

PHYSICAL CRITICAL CHARACTERISTICS:

Source code files
Compiled object code designation
Module checksum generated at the time that the PROM is encoded.

PERFORMANCE CRITICAL CHARACTERISTICS:

Initialization control
Internal status monitoring
Responds to messages from the HFC-6000 controller.
Controls digital I/O scan function.
Hardware watchdog strobe function

COMMERCIAL GRADE SOFTWARE EVALUATION

Software P/N: 9120678-14
CGS CLASSIFICATION SHEET

CGSE010, Rev A

Rev: CKSM 2164 Description: HFC-DC34 Firmware

DEPENDABILITY CRITICAL CHARACTERISTICS:

Built-in quality
Configuration control
Operating history
Problem reporting and resolution

METHOD (MARK ALL THAT APPLY):

[X] Method 1 - Special Tests and Inspections.

[X] Method 2 - Design Documentation and/or Commercial Grade survey of vendor.

[ ] Method 3 - Source Verification.

[X] Method 4 - Historical Performance Records.

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1
CGS CRITICAL CHARACTERISTICS SHEET

Software P/N: 9120678-14 Description: HFC-DC34 Firmware CGSE 010 Rev: A

Physical Critical Characteristic	Acceptance Criteria	Method of Verification
CC#001 Source Code Files	[ ]. Revision record is included in the header section of each source file. Source files are under configuration control. Source code reviews generated for critical source code files.	Direct examination of source files Source code review documentation
CC#002 Compiled object code designation	Object code file is designated 912067614.hex.	Direct verification
CC#003 Module checksum and PROM part number.	Two-byte checksum (2C30) calculated at the time that the PROM is programmed. PROM part number and checksum value are printed on a label, and this label is attached to the PROM.	Checksum 2C30 is generated by ChipWin program at the time the PROM is encoded. Physical verification of label on U8.

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1
CGS CRITICAL CHARACTERISTICS SHEET

Software P/N: 9120678-14 Description: HFC-DC34 Firmware CGSE 010 Rev: A

Performance Critical Characteristic	Acceptance Criteria	Method of Verification
CC#004 Initialization control	Runs initialization tests per design specification to verify operability.	Source code review Prototype test report
CC#005 Monitors overall hardware status	Monitors excitation voltage status, 125 vdc status, and coil continuity status. Maintains count of communication errors. [.]	Source code review Prototype test report
CC#006 Controls I/O scan function	Per design specification	Source code review Prototype test report
CC#007 ICL response function	Responds to defined ICL messages per design specification.	Source code review Prototype test report
CC#008 Controls watchdog strobe	Per design specification	Source code review Prototype test report

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1
CGS CRITICAL CHARACTERISTICS SHEET

Software P/N: 9120678-14 Description: HFC-DC34 Firmware CGSE 010 Rev: A

Dependability Critical Characteristic	Acceptance Criteria	Method of Verification
CC#009 Built-in quality	NQA1 QA Program	Records of internal and external audits. Software review records. Prototype test records.
CC#010 Configuration control	[	Direct inspection
CC#011 Operating history	Revision record in source files indicate that all components have more than 4 years operating history, including the past several years at Ulchin NPP	Direct inspection CR and SCR database
CC#012 Problem reporting and resolution	Corrective action process is in place and being used for evaluating and resolving reported problems. HFC supports CFR Part 21 reporting.	QPP 16.1 and CR/SCR database QPP 16.3

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1
CGS M&TE SHEET

Software P/N: 9120678-14 **Description:** HFC-DC34 Firmware CGSE 010 **Rev:** A

Record all M&TE used for this test:

M&TE CIN#	M&TE Desc.	Cal. Due Date	M&TE CIN#	M&TE Desc.	Cal. Due Date

Condition Report (CR): [ ] Yes [ ] No [ ] N/A If yes, identify below:
CR# [ ]Acceptable [ ] Not Acceptable Initials/Date: _____
CR# [ ]Acceptable [ ] Not Acceptable Initials/Date: _____
CR# [ ]Acceptable [ ] Not Acceptable Initials/Date: _____
Other:

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1

CGS CRITICAL CHARACTERISTICS RECORD SHEET

Software P/N: 9120678-14 **Description:** HFC-DC34 Firmware CGSE 010 **Rev:** A

Sheet ___ **of** ___

CC#	P/F/N	Performed By	Date	Verified By	Date
001					
002					
003					
004					
005					
006					
007					
008					
009					
010					
011					
012					

P = Pass, F = Fail, N = Not Applicable

Characteristics Record Sheet Competition:

Verified By _____ **Approved By** _____

Date:_____ **Date:**_____

ATTACHMENT 7.1

COMMERCIAL GRADE SOFTWARE EVALUATION

EVALUATION NUMBER: CGSE012

REVISION: A

Safety Related: [Y] YES [ ] NO

MANUFACTURER: HF Controls

DESCRIPTION: HFC-DO8J Firmware

SOFTWARE NUMBER: 9120677-14 Rev: CKSM AD95

Assigned Engineer: Jonathan Taylor Date: 12/15/05

Technical Reviewer: Steve Yang Date: 12/15/05

Approved By: Allen Hsu Date: 12/15/05

V & V Review: Bob Cain Date: 12/16/05

COMMERCIAL GRADE SOFTWARE EVALUATION

Software P/N: 9120677-14
CGS CLASSIFICATION SHEET

CGSE012, Rev A

Rev: CKSM AD95 Description: HFC-DO8J Firmware

1. What is the safety function of the parent component in which this item is to be installed?

   Controls safety system operation.

2. What is the function of the item?

   Controls internal operation of 8-channel DO module.

3. What are the ways in which the item should fail in actual service?

   Program error causes processor stall or invalid branching operation.

   Program error corrupts DO image or prevents transfer of image to output channel.

   Program error results in loss of communication with controller.

4. Would any of these postulated failures of the item prevent the parent component or associated components from performing their intended safety function?

   Loss or corruption of DO control prevents automated control of field process.

   Yes [X] (Item has safety function)

   No [ ] (Item has non-safety function)

COMMERCIAL GRADE SOFTWARE EVALUATION

Software P/N: 9120677-14
CGS CLASSIFICATION SHEET

CGSE012, Rev A

Rev: CKSM AD95 Description: HFC-DO8J Firmware

Method {Mark all that apply}

A "YES" in all three criteria is required for the item to be classified as commercial grade.

		YES	NO
1	Is the item not subject to design or specification requirements that are unique to facilities or activities licensed or regulated by the USNRC?	[X]	[ ]
2	Is the item used in applications other than facilities or activities licensed or regulated by the USNRC?	[X]	[ ]
3	Is the item ordered from manufacturer/supplier on the basis of specifications set forth in the manufacturer's published description?	[X]	[ ]
	Commercial Grade Software Item	[X]	[ ]

PHYSICAL CRITICAL CHARACTERISTICS:

Source code files
Compiled object code designation
Module checksum generated at the time that the PROM is encoded.

PERFORMANCE CRITICAL CHARACTERISTICS:

Initialization control
Responds to messages from the HFC-6000 controller.
Controls digital DO scan function.
Hardware watchdog strobe function

COMMERCIAL GRADE SOFTWARE EVALUATION

Software P/N: 9120677-14
CGS CLASSIFICATION SHEET

CGSE012, Rev A

Rev: CKSM AD95 Description: HFC-DO8J Firmware

DEPENDABILITY CRITICAL CHARACTERISTICS:

Built-in quality
Configuration control
Operating history
Problem reporting and resolution

METHOD (MARK ALL THAT APPLY):

[X] Method 1 - Special Tests and Inspections.

[X] Method 2 - Design Documentation and/or Commercial Grade survey of vendor.

[ ] Method 3 - Source Verification.

[X] Method 4 - Historical Performance Records.

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1

CGS CRITICAL CHARACTERISTICS SHEET

Software P/N: 9120677-14 **Description:** HFC-DO8J Firmware CGSE 012 **Rev:** A

Physical Critical Characteristic	Acceptance Criteria	Method of Verification
CC#001 Source Code Files	[Revision record is included in header section of each source file. Source code files are under configuration control Source code reviews generated for critical source code files.	Direct verification of source files Source code review documentation
CC#002 Compiled object code designation	Object code file is designated 912067714.hex.	Direct verification
CC#003 Module checksum and PROM part number.	Two-byte checksum (AD95) calculated at the time that the PROM is programmed. PROM part number and checksum value are printed on a label, and this label is attached to the PROM.	Checksum AD95 is generated by ChipWin program at the time the PROM is encoded. Physical verification of label on U5.

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1
CGS CRITICAL CHARACTERISTICS SHEET

Software P/N: 9120677-14 Description: HFC-DO8J Firmware CGSE 012 Rev: A

Performance Critical Characteristic	**Acceptance Criteria**	**Method of Verification**
CC#004 Initialization control	Runs initialization tests per design specification to verify operability.	Source code review Prototype test report
CC#005 Controls DO scan function	Per design specification	Source code review Prototype test report
CC#006 ICL response function	Responds to defined ICL messages per design specification.	Source code review Prototype test report
CC#007 Controls watchdog strobe	Per design specification	Source code review Prototype test report
Dependability Critical Characteristic	**Acceptance Criteria**	**Method of Verification**
CC#008 Built-in quality	NQA1 QA Program	Records of internal and external audits. Software review records. Pototype test records.
Configuration control	[	Direct inspection
Operating history	Revision records in source files indicate up to 15 years operating history, including the past several years at Ulchin NPP.	Direct inspection CR and SCR database
Problem reporting and resolution	Corrective action program in place and being used for evaluating and resolving reported problems. HFC supports CFR Part 21 reporting	QPP 16.1 and CR/SCR database QPP 16.3

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1
CGS M&TE SHEET

Software P/N: 9120677-14 **Description:** HFC-DO8J Firmware CGSE 012 **Rev:** A

Record all M&TE used for this test:

M&TE CIN#	M&TE Desc.	Cal. Due Date	M&TE CIN#	M&TE Desc.	Cal. Due Date

Condition Report (CR): [ ] Yes [ ] No [ ] N/A If yes, identify below:
CR# [ ]Acceptable [ ] Not Acceptable Initials/Date: _____
CR# [ ]Acceptable [ ] Not Acceptable Initials/Date: _____
CR# [ ]Acceptable [ ] Not Acceptable Initials/Date: _____
Other:

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1

CGS CRITICAL CHARACTERISTICS RECORD SHEET

Software P/N: 9120677-14 **Description:** HFC-DO8J Firmware CGSE 012 **Rev:** A

Sheet ____ **of** ____

CC#	P/F/N	Performed By	Date	Verified By	Date
001					
002					
003					
004					
005					
006					
007					
008					
009					
010					
011					
012					

P = Pass, F = Fail, N = Not Applicable

Characteristics Record Sheet Competition:

Verified By _____ **Approved By** _____

Date:_____ **Date:**_____

HF Controls Corporation

HFC Control System Components

CQ4 Blocks

General Information For CQ4 Blocks

DS001-000-03

Rev A

Effective Date	3/15/05
Author/title	Jonathan Taylor
Reviewer/title	Steve Yang
Approval/title	Terrence A. Geradis

[ ]

Copyright© 2005 HF Controls Corporation

DOOSAN

HFC

Revision History

Date	Revision	Author	Changes
12/09/03	0	D. Bynum	Initial
12/11/03	1	D. Bynum	Review Comments Incorporated
3/2/05	A	J Taylor	Initial Release

Table of Contents

1. INTRODUCTION 3

2. LIST OF CQ4 BLOCKS 3

3. DATA STRUCTURES OF CQ4 BLOCKS 6

4. CQ4 BLOCK COMMON FUNCTIONS 8

4.1 ATW PROCESSING 8
4.2 SPANS 10
4.3 ALARMS 10
4.4 CLAMPS 12
4.5 SET POINTS (SP) 13

Index of Tables

Table 1. CQ4 Blocks Designations and Functions 4
Table 1. CQ4 Blocks Designations and Functions (Cont) 5
Table 2: Block Data Structure 6
Table 3: BL Quality Word Bit/Field Definitions 6
Table 3: BL Quality Word Bit/Field Definitions (Cont) 8
Table 4: ATW Codes and Associated Meanings 9
Table 5: Possible Block Configurations 13

1. Introduction

HFC control systems use a subsystem called CQ4 to process digital images of analog values to implement analog control functions. An analog control loop can be described by the following sequence:

- The state of a process is sensed and converted to a digital image.
- This digital image then is passed to program algorithms that use it to calculate a control value.
- The control value is converted back to an analog signal.
- This signal adjusts the operation of a field device.
- The operation of the field device then affects the process under control in the desired manner.

The CQ4 subsystem is a modular, data-driven, high-level programming structure that uses elements called blocks for constructing the algorithm for individual control loops. These CQ4 blocks are data structures containing static and dynamic data. The static data is user defined and consists of such information as block number, type, processing frequency, and other parameters. Dynamic data consists of inputs from external devices or output values from other blocks. Each time the algorithm for a particular block executes, the following operations take place:

- That block evaluates its internal status.
- If the block is active, it obtains new values for its inputs and inserts those new values into its algorithm.
- The results of the current calculation then overwrite those from the previous cycle, and these values determine the block output value.

This document describes common CQ4 block functions that execute various support operations included in most block algorithms.

2. List of CQ4 Blocks

There are 37 CQ4 blocks available for use in the application code for an HFC control system. The detailed algorithm for each block type is described in a separate appendix to this document. Table 1 lists each block type defined for the CQ4 subsystem, the appendix that provides a detailed description of block operation, and a summary of functions supported.

Table 1. CQ4 Blocks Designations and Functions [

Block	Appendix	Description	ATW	Clamps	Alarms

]

3. Data Structures of CQ4 Blocks

[

Table 2: Block Data Structure

Field	Size	Function

Table 3: BL Quality Word Bit/Field Definitions

Tag	Bit	Description
		BLQ1

Table 3: BL Quality Word Bit/Field Definitions (Cont)

Tag	Bit	Description

4. CQ4 Block Common Functions

4.1 ATW Processing

Antiwindup (ATW) processing is a method for preventing block output values from increasing or decreasing beyond safe operating limits for the external process under control. For example, if a specific block generates a control value for automatic operation while the field device is in manual mode, the block might begin increasing (or decreasing) its output to force a response. If the field device is then put into automatic mode, the value of the control variable could cause severe bumping, which might damage the external hardware. The CQ4 subsystem uses ATW codes to avoid such problems. When a control loop is configured to use ATW processing, each intermediate block in the control loop uses the ATW code from its downstream link to limit the magnitude of its output, and it passes ATW codes to each of its variable upstream input sources. The resulting transfer of ATW codes, combined with

periodic initialization of the intermediate blocks, ensures that the values of the process variable (PV) and control output remain within safe limits for system operation. Specific values for the ATW code and their associated meanings are listed in Table 4.

Table 4: ATW Codes and Associated Meanings

ATW Code	Meaning
0	No ATW restrictions exist on current output value.
1	Current output cannot be less than previous output.
2	Current output cannot be greater than previous output.
3	Current output is not used.

[

6.

.]

4.2 Spans

The important parameters for spans are Input Zero, Input Span, Output Zero, Output Span. Input Zero and Input Span are used to specify a zero offset and full-scale count for the block input. A typical calculation associated with these parameters is as follows:
[

.]

4.3 Alarms

Alarms can be sent to an operator station and, optionally, to a printer. Alarm status is maintained in the quality word of BL, DI, DO, and MS points, and that status can be downloaded to operator stations along with other dynamic data. The operator stations include alarm processing software that formats alarm messages for display on alarm summary graphic pages. The Alarm Parameters are as follows:

- **Hi Trigger and Lo Trigger.** These two parameters specify limit values used by the equations program to trigger digital control of alarm processing.

- **Hi Alarm and Lo Alarm.** These two parameters specify a second set of limit values for alarm processing.

- **Deadband.** This parameter enables insertion of hysteresis into transitions from alarm-on to alarm-off status.

[

>

<

<

>

Figure 1: Significant Change Alarm Points

]

4.4 Clamps

[

-
-
-]

4.5 Set Points (SP)

[

.]

Table 5: Possible Block Configurations

Processing Mode	SP Mode	SP Source	MN bit	CA bit	CM bit

HF Controls

HF Controls Corporation

Redundancy and Failover Mechanism
Component Design Specification

DS001-000-08

Rev C

Effective Date	5/15/07
Author/Title	J. Taylor
Reviewer/Title	J. Stevens
Approval/Title	D. Briner

[ ]

Copyright© 2007 HF Controls Corporation

Revision History

Date	Revision	Author	Changes
05/21/04	0	B Han	Initial draft
12/30/04	A	B Han	Release
10/23/06	B	J. Stevens	KOPEC Comments (sheet 16)
3/9/07	C	J Taylor	SCR 1766

Table of Contents

1. INTRODUCTION **4**

2. FUNCTIONAL DESCRIPTION **5**

2.1 NEGOTIATION MECHANISM OF CONTROLLER OPERATION MODE 5
2.2 FAILOVER MECHANISM AND SYNCHRONIZATION 5
2.3 DPM TRANSFER 5
2.4 CONTROLLER FAILURE DETECTION MECHANISM 6
2.5 MAINTENANCE FAILOVER MECHANISM 6

3. DESCRIPTION OF HARDWARE CIRCUITRY DESIGN AND SIGNALS **6**

3.1 FAILOVER SIGNALS 7
3.1.1 SANE/ Signal *8*
3.1.2 PRIMARY/, PRIMARY+, FAILOVER+ and FAILOVER-/ Signals *8*
3.1.3 MAITENANCE_FAILOVER_ENABLE/ *9*
3.1.4 REQUEST_FOR_PRI/ and REQUEST_FOR_SEC/ Signals *9*
3.1.5 CPUSEL Signals *10*
3.2 FAILOVER MECHANISM 10
3.3 MAINTENANCE FAILOVER MECHANISM 10
3.3.1 Request for Controller A to Become Secondary *10*
3.3.2 Request for Controller A to Become Primary *11*

4. DESCRIPTION OF SOFTWARE DESIGN **11**

4.1 DPM TRANSFER 11
4.1.1 DPM Availability *11*
4.1.2 DPM Ownership *12*
4.1.3 DPM Configuration *12*
4.1.4 CQ3 and CQ4 Data Transfer *14*
4.2 CONTROLLER FAILURE DETECTION 15
4.3 SYNCHRONIZATION AFTER FAILOVER 19

5. SOURCE LISTING **21**

5.1 DPM CONFIGURATION: 21
5.2 DPM CQ3 AND CQ4 DROP AND PICK UP ROUTINES: 21
5.3 DPM STATIC DATA EQUALIZATION: 22
5.4 SYSTEM MAILBOX: 22

1. Introduction

Most of HFC control systems support redundant controller configuration. As shown in Figure1-1, a redundant configuration consists of two controllers with a dual port memory (DPM) in between. Each controller in the redundant pair operates in one of two different modes: primary or secondary mode. The primary controller provides all system control functions and the secondary controller functions as a hot spare to take over operation on failure of the primary controller. The transition of control from the primary controller to the secondary controller is called failover.

The purpose of redundancy is to provide higher reliability of the system. When one controller fails, the system can still function as normal with the other controller taking over control. However, it is important to ensure a smooth transition from one controller to another to prevent potentially hazardous bumping or disturbances to the plant. It is also important to make sure that the failover mechanism is available. The general way of checking on this is to perform a regular maintenance procedure by manually trigger a failover to determine if the secondary is able to take over control properly. This procedure is called maintenance failover.

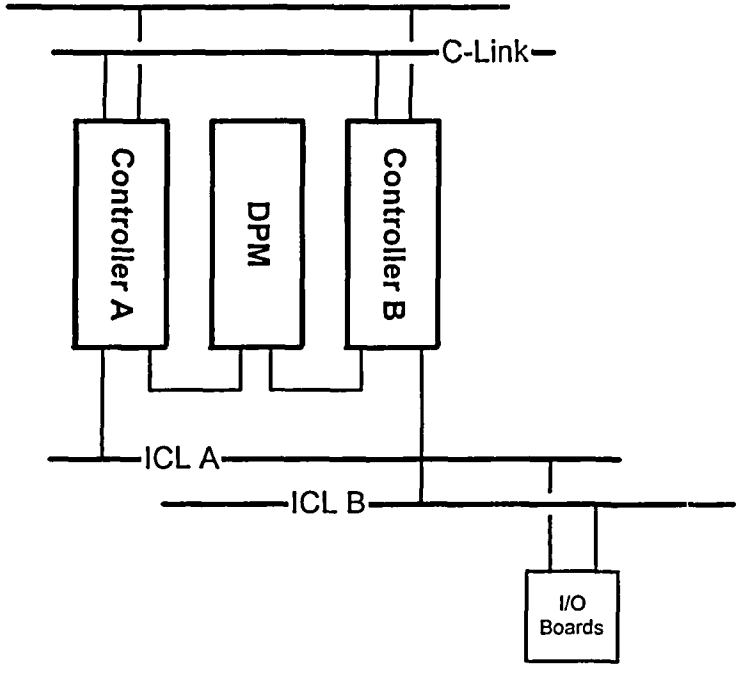


Figure 1 - Architecture of Redundant Controller Configuration

The purpose of this document is to provide functional descriptions of the redundant controller configuration and failover mechanism as well as hardware and software design descriptions to implement these functions.

2. Functional Description

The main functions for a redundant controller configuration are as follows:

- Negotiation mechanism to determine controller operation mode: Primary or Secondary
- Failover mechanism on failure of primary controller and synchronization of tasks and processors
- DPM Data Transfer Mechanism to ensure a smooth transition of control during failover
- Controller failure detection mechanism
- Maintenance failover mechanism

2.1 Negotiation Mechanism of Controller Operation Mode

Each controller must be able to determine in which mode it should operate: Primary or Secondary. Therefore, there must be a negotiation mechanism between the redundant controllers to make sure that one controller is running on Primary mode and the other controller is running on Secondary mode. The system cannot function properly if both controllers are in Primary mode, or if no controller operates as the Primary controller.

For a redundant controller configuration, the first controller that finishes initialization will become the primary controller. The other controller will start operating as the secondary controller after initialization. For a non-redundant configuration with only one controller, that controller will operate in Primary mode.

In a multi-processor controller environment, all tasks of each processor operate their functions in accordance with the operation mode of controller.

2.2 Failover Mechanism and Synchronization

During normal operation, if the primary controller fails, the secondary controller will take over control and become the primary controller. It is very important that all the tasks running on all the processors in the same controller are in the same operation mode, either Primary or Secondary. Otherwise it became an out of SYNC (synchronization) condition. When the controller's operation mode is changed, the whole controller should perform a synchronization process for all the tasks on each processor.

2.3 DPM Transfer

During redundant operation, the Primary controller periodically copies dynamic database and system status data to the DPM. The Secondary controller, on the other hand, periodically validates the transferred data in the DPM and updates its own record after the data validation process. This DPM Transferring Mechanism keeps the secondary controller updated with a current copy of the primary running status and ensures a smooth transition from one controller to the other in the case of failover. During a failover event when control

responsibility switches from the Primary to the Secondary, the Secondary retrieves the most recent application and system status data and uses it as an initial state for starting its control of the system. Starting with an initial state with most recent dynamic database should minimize any disturbance to the plant induced by the controller switch over, and prevent potentially hazardous bumping caused by transfer of control. After switch over is complete, the Secondary, which is now controlling the plant, switches to operation in the Primary control mode.

2.4 Controller Failure Detection Mechanism

[

.]

2.5 Maintenance Failover Mechanism

[

•]

3. Description of Hardware Circuitry Design and Signals

The general circuitry design for the failover mechanism and maintenance failover mechanism is shown in Figure 3-1. CPLD (Complex Programmable Logic Devices) can be used to provide most of the logics. There are several interconnected signals used to implement the failover mechanism as shown in Table 3-1.

3.1 Failover Signals

[

]

Figure 2 - Failover Circuit

Table 1 - Failover Signals

Controller Signal	DPM	Description

3.1.1 SANE/ Signal

[

.]

3.1.2 PRIMARY/, PRIMARY+, FAILOVER+ and FAILOVER-/ Signals

[

.]

3.1.3 MAITENANCE_FAILOVER_ENABLE/

[

]

3.1.4 REQUEST_FOR_PRI/ and REQUEST_FOR_SEC/ Signals

[

]

3.1.5 CPUSEL Signals

[

]

3.2 Failover Mechanism

[

.]

3.3 Maintenance Failover Mechanism

[

]

3.3.1 Request for Controller A to Become Secondary

[

.]

3.3.2 Request for Controller A to Become Primary

[

.]

4. Description of Software Design

Processor software provides support for the redundant controller configuration and controller failure detections.

4.1 DPM Transfer

[

].

4.1.1 DPM Availability

[

]

4.1.2 DPM Ownership

[

.]

4.1.3 DPM Configuration

[

Table 2 - DPM Configuration

Name	Size (byte)	Descriptions
	CQ3 Flags	

4.1.4 CQ3 and CQ4 Data Transfer

[

4.]

Table 3 - Equation Drop Process Error Codes

Status Code	Descriptions

4.2 Controller Failure Detection

[

.]

Figure 3 - System Processor Failure Detection

Table 4 - Failover Information in Diagnostic Area

4.3 Synchronization after Failover

[

.]

5. Source Listing

5.1 DPM configuration:

[

]

5.2 DPM CQ3 and CQ4 drop and pick up routines:

[

]

5.3 DPM static data equalization:

[

]

5.4 System mailbox:

[

.]

HF Controls

HF Controls Corporation

HFC Control Systems

ICL Protocol
Component Design Specification

Inter-Communications Link

DS002-000-02

Rev D

INFORMATION ONLY

Effective Date	5/15/07
Author/Title	J. Taylor
Reviewer/Title	B. Cain
Approval/Title	T. Gerardis

[ ]

Copyright© 2007 HF Controls Corporation

Revision History

Date	Revision	Author	Changes
11/14/03	0	B Lin	Initial draft
12/09/03	1	B Lin	Review Comments Incorporated
12/11/03	2	B Lin	2nd Review
05/12/04	A	B Han	Release document
04/25/06	B	T.Park, S.Kang	Modification to include FPC06 entry for HF1073 SCR #1486
9/20/06	C	J. Taylor	Revised to eliminate duplication in 700009-50 and complete modifications for SCR #1486.
3/8/07	D	J Taylor	SCR 1766

Table of Contents

Number **Title** **Page**

1.0 INTRODUCTION 4

2.0 FUNCTIONAL DESCRIPTION 4

2.1 Redundant Serial Link 4
2.2 Polling 5
2.3 Secondary Loopback Test 6
2.4 Secondary Polling Function 7

3.0 MESSAGE STRUCTURES 7

3.1 Address Code 8
3.2 Message Codes 8
3.3 Data Field 11
3.3.1 Normal Request/Response Message Data Field 11
3.3.2 Request/Response Message Data Field for the HFC-FPC06 14
3.3.3 Special Request/Response Message Data Field 14
3.3.3.1 Diagnostic Request Message 16
3.3.3.2 SOE Messages 19
3.4 Secondary Loop Back Message Data Field 20
3.5 NACK Response 21

3.1 CRC-16 ALGORITHM 21

List of Tables

Number	Title	Page
Table 1.	Defined Message Codes for HFC Control Systems	9
Table 2.	Data Field for Digital Images	11
Table 3.	Data Field for Analog Images	12
Table 4.	SOE Time Sync Message Format	14
Table 5.	Data Formats for FPC06 Message Transfers	15
Table 6.	Special Request Message	15
Table 7.	Special Response Message	15
Table 8.	Data Field for Standard Diagnostic Response	16
Table 9.	Data Field for RTD Diagnostic Response (QIO Protocol)	17
Table 10.	Diagnostic Response Data for DI with Quality Status	18
Table 11.	SOE Record Request Message Data Field Structure	19
Table 12.	SOE Event Record Formats	19
Table 13.	Secondary Loopback Message Exchange	20

1.0 INTRODUCTION

The Inter-Communications Link (ICL) provides the hardware medium for data transfer between a HFC controller board and configured I/O boards. [

.]

2.0 FUNCTIONAL DESCRIPTION

The ICL protocol is an HFC proprietary design used for general communications between a controller board and an I/O board. Its purpose is to provide a fixed format of data transfer for each type of I/O board. [

]

2.1 REDUNDANT SERIAL LINK

[

.]

Figure 1. ICL Architecture

2.2 POLLING

[

]

2.3 SECONDARY LOOPBACK TEST

[

]

Figure 2. Secondary Loop Back Test

2.4 SECONDARY POLLING FUNCTION

[

]

3.0 MESSAGE STRUCTURES

Both the IOIF and the QIO protocols are based on fixed message structures as shown below.
[

IOIF Poll Command Format

QIO Response Message Format

.]

3.1 ADDRESS CODE

[

]

QIO Address Code Structure							

IOIF Address Code Structure							

3.2 MESSAGE CODES

[

.]

Table 1. Defined Message Codes for HFC Control Systems

Code	Description	Bytes in command	Bytes in response

Table 1. Defined Message Codes for HFC Control Systems (Cont)

Code	Description	Bytes in command	Bytes in response
			Variable

3.3 DATA FIELD

The data field required for each message type is determined by the message code. Most messages have a fixed byte count based on the board type of the slave station. Certain message types may have a different byte count depending on specific application. In either case, both the controller and the slave station used the expected byte count as the basis for decoding the message received.

3.3.1 Normal Request/Response Message Data Field

[

]

Table 2. Data Field for Digital Images

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Table 2. Data Field for Digital Images (Cont)

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Table 3. Data Field for Analog Images

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Table 3. Data Field for Analog Images (Cont)

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Table 4. SOE Time Sync Message Format

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

3.3.2 Request/Response Message Data Field for the HFC-FPC06

[

]

3.3.3 Special Request/Response Message Data Field

[

.]

Table 5. Data Formats for FPC06 Message Transfers

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Table 6. Special Request Message

Byte	Description

Table 7. Special Response Message

Byte	Description

3.3.3.1 Diagnostic Request Message

[

]

Table 8. Data Field for Standard Diagnostic Response

Byte	Description
Byte	Description

Table 9. Data Field for RTD Diagnostic Response (QIO Protocol)

Byte	Description

Table 8. Data Field for RTD Diagnostic Response (Cont)

Byte	Description
⋮	⋮

Table 10. Diagnostic Response Data for DI with Quality Status

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

3.3.3.2 SOE Messages

[

]

Table 11. SOE Record Request Message Data Field Structure

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Table 12. SOE Event Record Formats

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Table 12. SOE Event Record Formats (Cont)

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

3.4 SECONDARY LOOP BACK MESSAGE DATA FIELD

[

]

Table 13. Secondary Loopback Message Exchange

Byte	Description							

3.5 NACK RESPONSE

NACK is used to signal that a transfer error or timeout has occurred. The NACK response has no data field.

3.1 CRC-16 ALGORITHM

The CRC-16 is a 16-bit checksum appended to the end of the message. The algorithm is used by both the transmitter and receiver to check the message for accuracy. The HFC control system uses the same CRC-16 algorithm for all message transfers. This algorithm is the same as the one used for the MODBUS protocol. [

]

HF Controls

HF CONTROLS CORPORATION

HFC-6000 Controller
SC, SAP, SEP
Firmware Requirements Specification

RS901-000-37

Rev A

Effective Date 5/15/07

Author/Title J. Taylor

Reviewer/Title D. Briner

Approval/Title T. Gerardis

[ ]

Copyright© 2007 HF Controls Corporation

Revision History

Date	Revision	Author	Changes
10/17/06	A0	J Taylor	Draft
1/12/07	A	J Taylor	Preliminary release

Table of Contents

Section	Title	Page
1.0	**Scope**	**3**
2.0	**Purpose**	**3**
3.0	**References**	**3**
4.0	**Functional Analysis**	**3**
4.1	Requirements Analysis	4
4.1.1	SC Firmware Requirements	4
4.1.2	SEP Firmware Requirements	7
4.1.3	SAP Firmware Requirements	9
4.1.4	CPLD Program Code	10
4.2	Conceptional Design Description	11
4.2.1	Overview of Project	11
4.2.2	Assumptions	11
4.2.3	Critical Issues	11
5.0	**Safety Analyses**	**11**
	Main controller – SC Code	11
5.1.1	Criticality	12
5.1.2	Hazard	12
5.1.3	Security	12
5.1.4	Risk	12
5.2	ICL Controller - SAP Code	13
5.2.1	Criticality	13
5.2.2	Hazard	13
5.2.3	Security	13
5.2.4	Risk	14
5.3	C-Link Controller - SEP Code	14
5.3.1	Criticality	14
5.3.2	Hazard	15
5.3.3	Security	15
5.3.4	Risk	15

1.0 SCOPE

This document covers requirements for the SC, SEP, and SAP firmware code for controllers used in HFC control systems. The HFC-SBC06 and ECS-SBC06 assemblies include one each of these controller types. The HFC-SCG06 assembly contains one SC processor and two SEP processors. Functional differences in the operation of the program code will be determined by hardware configuration, which will be sensed by the code during its initialization sequence.

2.0 PURPOSE

This document defines overall requirements for the SC, SAP, and SEP firmware to be used by the HFC controller assemblies. [
]

3.0 REFERENCES

WI-ENG-202	Development of Software/Firmware Requirements Specifications
DS001-000-01	Operating System Component Design Specification
DS002-000-01	C-Link Protocol Design Specification
DS002-000-03	UCP Protocol Design Specification
RS901-000-01	HFC-6000 Product Line-Requirements
700901-04	HFC-DPM06 Requirements Specification
700901-05	HFC-SBC06 Requirements (Hardware) Specification
RS901-000-32	HFC-SCG06 Requirements (Hardware) Specification
400960-01	HFC-SCG06 Schematic Diagram

4.0 FUNCTIONAL ANALYSIS

The HFC-SBC06 hardware assembly is designed for installation in a standard HFC-6000 chassis. The hardware assembly has one SC processor section, one SAP processor section, and one SEP processor section. The SC processor is designated as the main processor. It executes the application program and monitors overall status for the controller. The SAP and SEP processor operate independently of the SC processor and independently of one another. The SAP processor controls scan cycles with configured I/O modules via a serial ICL. The SEP processors controls redundant serial interfaces with the C-Link to enable controller-to-controller communication. The HFC-SBC06 controller is typically configured to control one or more field processes via a combination of discrete and analog I/O signals. It may be configured for either redundant or non redundant operation.

The HFC-SCG06 hardware assembly is also designed for installation in a standard HFC-6000 chassis. The hardware assembly has one SC processor section and two SEP processor sections. Each of the SEP processors will control redundant serial interfaces for separate C-Links. One of the two C-Links will serve as the common communication link with other HFC-SCG06 and ECS-SBC06 remotes and also provide the path for status transfer to external workstations. The second C-Link will provide the mechanism for communication with a maximum of 64 HFC-SBC06 controllers. The HFC-SCG06

module will perform the function of a Communication Master (CM) remote and will not be used for functional control of any field process. Accordingly, its application program will be restricted to data mapping functions.

[

.]

4.1 REQUIREMENTS ANALYSIS

4.1.1 SC Firmware Requirements

[

5.

Table 1. DIP Switch Configuration Data

Switch	Function

16.

.]

4.1.2 SEP Firmware Requirements

[

.]

4.1.3

SAP Firmware Requirements

[

14.

.]

4.1.4 CPLD Program Code

[

]

4.2 CONCEPTIONAL DESIGN DESCRIPTION

4.2.1 Overview of Project

[

.]

4.2.2 Assumptions

[

•

]

4.2.3 Critical Issues

[

]

5.0 SAFETY ANALYSES

[

.]

5.1.1 Criticality

[

]

5.1.2 Hazard

[

]

5.1.3 Security

[

]

5.1.4 Risk

[

•

.]

5.2 ICL CONTROLLER - SAP CODE

[

]

5.2.1 Criticality

[

]

5.2.2 Hazard

[

•

.]

5.2.3 Security

[

]

5.2.4 Risk

[

- .]

5.3 C-LINK CONTROLLER - SEP CODE

[

- .]

5.3.1 Criticality

[

]

5.3.2 Hazard

[

•

]

5.3.3 Security

[

]

5.3.4 Risk

[

•

]

HF CONTROLS CORPORATION

HFC-6000 Control System

TSAP Design Specification

ADS0401

Rev A

Effective Date _9/11/03_____

Author/Title _John Stevens_____

Reviewer/Title __Jonathan Taylor_____

Approval/Title __Allen Hsu_____

[ ]

Copyright© 2003 HF Controls Corporation

Revision History

Date	Revision	Author	Changes
11/7/02	0	JA Stevens	Draft
9/9/03	1	JA Stevens	Design review corrections

Table of Contents

SECTION 1 ..... **4**

GENERAL INFORMATION **4**

1.1 Introduction 4
1.2 References 4
1.3 Abbreviations and Acronyms 4

SECTION 2 ..... **6**

TSAP DESIGN **6**

2.1 INTRODUCTION 6
2.2 Application Program Software 6
2.2.1 Simulated Control Loop Logic 6
2.2.1.1 Medium Voltage Switchgear and Load Center Circuits (Air Circuit Breaker) .. 6
2.2.1.2 Reversing Motor Starter Circuits 10
2.2.1.3 Non-Reversing Motor Starter Circuits 13
2.2.1.4 Solenoid Valve Circuits 15
2.2.2 TSAP Logic Algorithms 18

Section 1

GENERAL INFORMATION

1.1 INTRODUCTION

This document provides a detailed description of the test system application program (TSAP) that will make up the HFC, HFC-6000 test specimen to be presented for qualification by the NRC.

1.2 REFERENCES

EPRI TR-107330	Generic Requirements Specification for Qualifying a Commercially Available PLC for Safety-Related Application in Nuclear Power Plants
700901-01	HFC-6000 Requirements Specification
700905-01	Analog Composite Logics
700906-01	Digital Composite Logics
700907-01	Schematic Wiring Diagram – TSAP
700908-01	Schematic Wiring Diagram – HPAT
700901-09	TSAP Requirements Specification
700910-01	PCB Card Layout Diagram – TSAP

1.3 ABBREVIATIONS AND ACRONYMS

BIT	Built-in Test
CPC	Communication Protocol Controller
CRC	Cyclical Redundancy Check
CREVAS	Control Room Emergency Ventilation Actuation System
CSM	Control Switch Module
DCS	Distributed Control System
ESF	Emergency Safety Feature
ESFAS	Emergency Safety Feature Actuation System

EWS	Engineering Workstation
HFC	HF Controls
ICL	Intercommunication Link
I/O	Input/Output
JCRT	Java CRT Workstation
LAN	Local Area Network
M/A	Manual/Automatic
MCC	Main Control Center
NRC	Nuclear Regulatory Commission
OIS	Operator Interface System
PCC	Peripheral Communication Controller
PID	Proportional Integral Derivative. A type of analog control algorithm.
RTD	Resistance Temperature Detector
SOE	Sequence of Events
TSAP	Test System Application Program. A "synthetic application" used to check/verify the PLC functionality needed to support the qualification test program.

Section 2

TSAP Design

2.1 INTRODUCTION

The software installed on the HFC hardware for NRC qualification will include operating system software and the TSAP installed on the master controller and a separate executive program on each I/O card. The software for each card will be installed in PROM, but the master controller will copy the program instructions to flash memory and run from flash memory during normal operation.

2.2 APPLICATION PROGRAM SOFTWARE

Document 700901-09 TSAP Requirements Specification defines the requirements of the TSAP. The following will describe the software design required to implement those requirements. The TSAP is based on a set of simulated control loops and program code to support operability and prudency test.

2.2.1 SIMULATED CONTROL LOOP LOGIC

[

]

2.2.1.1 Medium Voltage Switchgear and Load Center Circuits (Air Circuit Breaker)

The following describes the typical control function for a medium voltage switchgear. Section E will describe the specific I/O points to be used for the TSAP.

A. Format

[

2.]

Figure 4-1
Medium Voltage Switchgear and Load Center

B. Functional Circuit Types

[

4.

]

C. General Functions

[

11.

]

D. Inoperable Detector Functions

A flashing inoperable light and alarm operate under the following conditions to indicate equipment inoperable. The flashing inoperable light goes to steady when the start or stop pushbutton is operated and to off when the inoperable condition returns to normal.

[

f.

]

E. I/O Configuration

[

d.

]

2.2.1.2 Reversing Motor Starter Circuits

The following describes the typical control function for a reversing motor starter. Section E will describe the specific I/O points to be used for the TSAP.

A. Format

[

3.

]

Figure 4-2
Reversing Motor Starter

B. Functional Circuit Types

[

2.

]

C. General Functions

[

8.

]

D. Inoperable Detector Functions

A flashing inoperable light and alarm operate under the following conditions to indicate equipment inoperable. The flashing inoperable light goes to steady when the

start or stop pushbutton is operated and to off when the inoperable condition returns to normal.

[

d.

]

E. I/O Configuration

[

c.]

2.2.1.3 Non-Reversing Motor Starter Circuits

The following describes the typical control function for a non-reversing motor starter. Section E will describe the specific I/O points to be used for the TSAP.

A. Format

[

3.

]

Figure 4-3
Non-Reversing Motor Starter

B. Functional Circuit Types

[

2.

]

C. General Functions

[

5.

]

D. Inoperable Detector Functions

A flashing inoperable light and alarm operate under the following conditions to indicate equipment inoperable. The flashing inoperable light goes to steady when the start or stop pushbutton is operated and to off when the inoperable condition returns to normal.

[

e.

]

E. I/O Configuration

[

d.

]

2.2.1.4 Solenoid Valve Circuits

The following describes the typical control function for a solenoid valve. Section E will describe the specific I/O points to be used for the TSAP.

A. Format

[

3.

]

Figure 4-4
Solenoid Valve

B. Functional Circuit Types

[

4.

]

C. General Functions

[

6.

]

D. Inoperable Detector Functions

A flashing inoperable light and alarm operate under the following conditions to indicate equipment inoperable. The flashing inoperable light goes to steady when the start or stop pushbutton is operated and to off when the inoperable condition returns to normal.

[

d.

]

E. I/O Configuration

[

a.]

2.2.2 TSAP LOGIC ALGORITHMS

A. Format

[

2.]

B. Functional Digital Circuits

[

f.

]

C. Functional Analog Circuits

[

7.

]

HF CONTROLS CORPORATION

HFC-6000 Product Line
Nuclear Qualification Project ERD 111

VERIFICATION AND VALIDATION REPORT

VV0415

Rev A

INFORMATION ONLY

Effective Date	12/07/05
Author	Jonathan Taylor
Reviewer	Robert Cain
Approval	Steve Yang

[ ]

Copyright© 2005 HF Controls Corporation

DOOSAN

HFC

Revision History

Date	Revision	Author	Changes
10/12/05	A0	J Taylor	Initial Draft
12/01/05	A	J Taylor	First Release

Table of Contents

Section	Title	Page
1	**INTRODUCTION**	**1-1**
1.1	Scope	1-1
1.2	References	1-1
1.3	Definitions and Acronyms	1-2
1.3.1	Definitions	1-2
1.3.2	Abbreviations	1-3
2	**PROJECT PLANNING PHASE**	**2-1**
2.1	Scope	2-1
2.2	References	2-1
2.3	Summary of Activities	2-1
2.3.1	Preliminary Audit	2-1
2.3.2	Planning Meetings	2-2
2.3.3	Planning Documents	2-4
2.4	Overall Assessment	2-4
3	**REQUIREMENTS PHASE**	**3-1**
3.1	Scope	3-1
3.2	References	3-1
3.3	Summary of Activities	3-1
3.3.1	Project Quality Plan	3-2
3.3.2	Module Requirements Specifications	3-2
3.3.3	HFC-PCC06 Software Requirements	3-2
3.3.4	HFC-FPC06 Software Requirements	3-2
3.3.5	Master Test Plan	3-3
3.3.6	TSAP Requirements	3-3
3.3.7	HPAT Requirements	3-3

Table of Contents

Section	Title	Page
3.4	Assessment of Overall Quality	3-3
3.5	Conclusions and Recommendations	3-4
4	**DESIGN PHASE**	**4-1**
4.1	Scope	4-1
4.2	References	4-1
4.3	Summary of Activities	4-2
4.3.1	Module Design Specifications	4-2
4.3.2	Legacy Software Component Design Specifications	4-3
4.3.2.1	Pseudo Code Analysis	4-5
4.3.2.2	Source Code Review	4-5
4.3.2.3	Source Code Design Descriptions	4-5
4.3.3	Test Specimen Design	4-6
4.3.4	Test procedures	4-6
4.4	External Audit	4-6
4.5	Assessment of Overall Quality	4-7
4.6	Conclusions and Recommendations	4-7
5	**IMPLEMENTATION PHASE**	**5-1**
5.1	Scope	5-1
5.2	References	5-1
5.3	Summary of Activities	5-2
5.3.1	Hardware Fabrication	5-2
5.3.2	Program Code Implementation Tasks	5-2
5.3.2.1	TSAP and HPAT Source Code Review	5-2
5.3.2.2	Application Software Objects Acceptance Test	5-3
5.3.2.3	Legacy Software Component Tests	5-3
5.3.2.4	HFC-PCC06 Software Test	5-4
5.4	Assessment of Overall Quality	5-4
5.4.1	Hardware Implementation	5-4
5.4.2	System Software Implementation	5-4
5.4.3	TSAP and HPAT Code	5-4
5.5	Conclusions and Recommendations	5-4
6	**INTEGRATION PHASE**	**6-1**
6.1	Scope	6-1
6.2	References	6-1
6.3	Summary of Activities	6-1
6.3.1	Burn-in Test	6-1
6.3.2	Integration Test	6-2
6.3.3	Validation Test	6-2
6.4	Test Results	6-2

Table of Contents

Section	Title	Page
6.5	Assessment of Overall Quality	6-3
6.6	Conclusions and Recommendations	6-3
7	**ACCEPTANCE TESTING PHASE**	**7-1**
7.1	Scope	7-1
7.2	References	7-1
7.3	Summary of Activities	7-1
7.3.1	Initial Acceptance Test	7-1
7.3.1.1	Operability Test	7-1
7.3.1.2	Prudency Test	7-2
7.3.2	Acceptance Testing Prior to Seismic Retest	7-2
7.3.2.1	Test Specimen Validation	7-2
7.3.2.2	Operability Tests	7-2
7.3.2.3	Prudency Test	7-2
7.4	Test Results	7-3
7.5	Assessment of Overall Quality	7-3
7.6	Conclusions and Recommendations	7-3
8	**QUALIFICATION TEST PHASE**	**8-1**
8.1	Introduction	8-1
8.2	References	8-1
8.3	Summary of Activities	8-1
8.3.1	Initial Qualification Test Sequence	8-1
8.3.2	Seismic Retest	8-2
8.3.3	Regression Tests	8-2
8.3.3.1	Analog Response Time	8-2
8.3.3.2	Hardware Watchdog Timeout	8-2
8.3.3.3	Software Watchdog Timeout Period	8-3
8.4	External Audit	8-3
8.5	Outstanding Issues	8-4

Section 1

INTRODUCTION

1.1 SCOPE

This document is the Verification and Validation Report for the ERD111 project of HF Controls (HFC). This project was undertaken by HFC to develop a control system product line specifically for nuclear safety-related applications. The report covers Verification and Validation activities conducted during the life cycle phases for the ERD111 project. During the course of this project, a sample control system and synthetic application program were developed and subjected to the regimen of tests recommended by EPRI TR-107330, Generic Requirements for Qualifying a Commercially Available PLC for Safety-Related Applications in Nuclear Power Plants.

1.2 REFERENCES

The following list contains the industry standards used during the development and testing of the HFC-6000 product line.

EPRI TR-102323	Guidelines for Electromagnetic Interference Testing in Power Plants, Revision 1
EPRI TR-106439	Guideline on Evaluation and Acceptance of Commercial Grade Digital Equipment for Nuclear Safety Application
EPRI TR-107330	Generic Requirement Specification for Qualifying Commercially Available PLC for Safety-Related Applications in Nuclear Power Plants
IEC 801-2	Electromagnetic Compatibility for Industrial-Process Measurement and Control Equipment
IEEE Std 279-1971	Criteria for Protection Systems in Nuclear Power Generating Stations
IEEE Std 323-1974	IEEE Standard for Qualifying Class 1E Equipment for Nuclear Power Generating Stations
IEEE Std 344-1975	Recommended Practice for Seismic Qualification of Class 1E Equipment for Nuclear Power Generating Stations
IEEE Std 352-1987	Guide for Principles of Reliability Analysis of Nuclear Power Generating Station Protection Systems
IEEE Std 384-1981	Standard Criteria for Independence of Class 1E Equipment and Circuits

IEEE Std 603-1991	Standard Criteria for Safety Systems for Nuclear Power Generating Stations
IEEE Std 828-1990	Standard for Software Configuration Management Plans
IEEE Std 929-1983	Standard for Software Test Documentation
IEEE Std 830-1993	Recommended Practice for Software Requirements Specifications
IEEE Std 1008-1987	Standard for Software Unit Testing
IEEE Std 1012-1986	Standard for Software Verification and Validation Plans
IEEE Std 1028-1988	Standard for Software Reviews and Audits
IEEE Std 1042-1987	Guide to Software Configuration Management
IEEE Std 1050-1996	IEEE Guide for Instrumentation and Control Equipment Grounding in Generating Stations
IEEE Std 7-4.3.2	Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations
IEEE Std C62.41-1991	Recommended Practice on Surge Voltages in Low-Voltage AC Power Circuits
IEEE Std C62.45-1992	IEEE Guide on Surge Testing for Equipment Connected to Low-Voltage AC Power Circuits
NQA-1-1993	Quality Assurance Program Requirements for Nuclear Facilities

The day-to-day operations within HFC are governed by the HFC Quality Assurance Manual, individual Quality Process Procedures (QPP), and departmental work instructions. The QA Manual as well as the individual procedures and work instructions applicable to nuclear safety-related projects were developed based on NQA-1 and the relevant industry standards.

1.3 DEFINITIONS AND ACRONYMS

1.3.1 DEFINITIONS

Anomaly

Anything observed in the documentation, construction, or operation of the system that deviates from expectations based on previously verified components or reference documents.

Fault Tolerance

A built-in capacity of a system to provide continued correct execution in the presence of a limited number of hardware or software faults.

Legacy Software
Software that had been developed before a comprehensive V&V program had been established.

Software
Programs, procedures, rules, data, and any associated documentation pertaining to the operation of a computer system.

Requirements Phase
The period of a computer system life cycle during which functional and nonfunctional requirements capabilities are defined and documented. (Nonfunctional requirements include performance, interface, reliability, security, safety, and environmental concerns.)

Requirements Specification
Documentation of the essential requirements for the components that make up the overall computer system.

Validation
The process of evaluating an integrated computer system (hardware and software) or an individual component during or at the end of its development process to determine whether it satisfied specified requirements.

Verification
The process of evaluating a system or component to determine whether or not the products of a given development phase satisfy the conditions imposed at the start of that phase.

1.3.2 ABBREVIATIONS

ACE	Adverse Conditions and Events
BOM	Bill of Material
COTS	Commercial off-the-Shelf
CR	Condition Report
HAS	Historical Archiving System
HFC	HF Controls
HPAT	HFC Plant Automated Tester
I/O	Input/Output
JCRT	Java-based interactive graphic software utility used as a control system MMI
EPROM	Erasable PROM
EWS	Engineering Workstation
MMI	Man-Machine Interface

PC	Personal Computer
PCB	Printed Circuit Board
PCC	Peripheral Communication Controller
PDS	Previously Developed Software
PROM	Programmable Read-Only Memory
QA	Quality Assurance
QC	Quality Control
QCRT	Interactive graphic software utility designed to operate in a QNX environment
RAM	Random Access Memory
SBC	Single Board Computer
SOE	Sequence of Events
SVV	Software V&V
SVVR	Software V&V Report
UCP	Universal Communication Protocol
V&V	Verification and Validation

Section 2

PROJECT PLANNING PHASE

2.1 SCOPE

This section covers all V&V tasks accomplished and the overall results of V&V activities during the project planning phase.

2.2 REFERENCES

The following major documents and V&V task reports were generated during this phase of the project life cycle.

Assessment of HF Controls Digital Control Platforms for use in Nuclear Safety Related Service, Hurst Technologies
TN0401 Master Test Plan
VV0401 Product Development Plan

2.3 SUMMARY OF ACTIVITIES

This phase consisted of a series of planning meetings during which HFC management, selected engineers, and various consultant experts developed top-level plans for the HFC-6000 product line. The project that became ERD111 started as an outgrowth of S10079, the Plant Control System (PCS) project for Ulchin Nuclear Power Plant Units 5 and 6, which was then in process. The initial concept was to qualify the control system architecture that had just been developed for domestic use with the retrofit market as the primary targeted customer base.

2.3.1 PRELIMINARY AUDIT

[

]

2.3.2 PLANNING MEETINGS

[

]

2.3.3 PLANNING DOCUMENTS

HFC began this project using the QA program documents and departmental work instructions that were in place at that time. The following major planning documents were applicable to this phase of the project.

QAPM	QA Program Manual
QPP-2.1	Project Quality Plan
QPP-3.2	System Life Cycle and Verification/Validation Program
WI-ENG-003	Configuration Management
WI-ENG-008	Validation and Verification Program

2.4 OVERALL ASSESSMENT

[

]

Section 3

REQUIREMENTS PHASE

3.1 SCOPE

This section covers all V&V tasks and the overall results of V&V activities during the requirements phase for the development of the new nuclear safety-related product line. The project was designated ERD111, and the product line was named HFC-6000. During this phase, the requirements for the basic hardware and software components of the product line were defined. Legacy software modules would be used for practically all of the system-level software to limit requirements for new software development. As a result, the vast majority of new design consisted of hardware packaging only.

3.2 REFERENCES

The following major documents were generated during this phase of the project life cycle.

PQP 2003-005	ERD111 Project Quality Plan
700005-02	CSM Requirements Specification (legacy module)
700005-04	M/A Station Requirements Specification (legacy module)
700009-50	ECS-Series I/O Cards Software Design Specification (legacy module)
700901-01	HFC-6000 Product Line Requirements Specification
700901-03	HFC-PCC06 Serial Channel Multiplexer Requirements Specification
700901-04	HFC-DPM06 Dual-Ported Memory Requirements Specification
700901-05	HFC-SBC06 Controller Card Requirements
700901-06	General I/O Cards Requirements Specification
700901-07	FPD and FPC Requirements Specification
700901-09	TSAP Requirements Specification
700901-10	HPAT Requirements Specification
RS901-000-01	HFC-6000 Product Line Requirements Specification
RS901-000-02	HFC PCB Power Supply 24/48vdc Requirements Specification
RS901-000-04	FPC06 Controller Software Requirements Specification
RS901-000-11	HFC Single Loop Controller System Requirements Specification
RS901-000-12	19 inch Rack Power Supply Requirements Specification
RS901-000-13	HFC-FPC06 Real-Time Operating System Requirements Specification
TN0401	Master Test Plan

3.3 SUMMARY OF ACTIVITIES

During this period of the project life cycle, requirements specifications were prepared for each of the modules that were defined for the initial set of the HFC-6000 product line, and the Master Test Plan was completed. The original list of hardware modules was revised to produce the final design requirements for the qualification test specimen and TSAP.

3.3.1 PROJECT QUALITY PLAN

The Project Quality Plan is a document required by QPP-2.1. Its purpose is to define the overall scope and specific quality requirements for each project undertaken by HFC.

3.3.2 MODULE REQUIREMENTS SPECIFICATIONS

[

]

3.3.3 HFC-PCC06 SOFTWARE REQUIREMENTS

[

]

3.3.4 HFC-FPC06 SOFTWARE REQUIREMENTS

[

]

3.3.5 MASTER TEST PLAN

[

]

3.3.6 TSAP REQUIREMENTS

[

.]

3.3.7 HPAT REQUIREMENTS

[

]

3.4

ASSESSMENT OF OVERALL QUALITY

The individual Requirements Specification Evaluation reports provided a detailed accounting of all discrepancies that were detected as part of each document review. All detected discrepancies were resolved before the documents were finally accepted.

3.5 CONCLUSIONS AND RECOMMENDATIONS

[

]

Section 4

DESIGN PHASE

4.1 SCOPE

This section covers all V&V tasks accomplished and the overall results of V&V activities during the design phase. As previously indicated, the vast majority of new design consisted of hardware packaging only. The scope of new software design was minimal.

4.2 REFERENCES

The following major documents were generated during this phase of the project life cycle.

400401-01	HFC-ILR06 Hardware Design Specification
400409-01	HFC-BPC01-19 Hardware Design Specification
400414-01	HFC-BPE01-19 Hardware Design Specification
400419-01	HFC-SBC06 Hardware Design Specification
400424-01	HFC-DPM06 Design Specification
400429-01	HFC-PCC06 Design Specification
400434-01	HFC-AI16F Hardware Design Specification
400439-01	HFC-AI8L Hardware Design Specification
400444-01	HFC-AI8M Hardware Design Specification
400449-01	HFC-AC4K Hardware Design Specification
400454-01	HFC-DI16I Hardware Design Specification
400459-01	HFC-DO8J Hardware Design Specification
400464-01	HFC-DC33 Hardware Design Specification
400469-01	HFC-DC34 Hardware Design Specification
400474-01	HFC-AO8F Hardware Design Specification
400479-01	HFC-AC36 Hardware Design Specification
400484-03	HFC-6000 Digital and Analog Board Functional Test
ADS0401	TSAP Design Specification
DD0401	Test Specimen Design Description
DS002-000-01	Component Design Specification C-Link Protocol
DS002-000-02	Component Design Specification ICL Protocol
DS002-000-03	Component Design Specification Universal Communications Protocol
DS001-000-01	Component Design Specification, Operating System
DS001-000-02	Component Design Specification, Equation Interpreter
DS001-000-03	Component Design Specification, CQ4 Blocks
DS001-000-07	Component Design Specification, Job Configuration
DS001-000-08	Component Design Specification, Redundancy and Failover
DS002-000-01	Component Design Specification, C-Link
DS002-000-02	Component Design Specification, ICL
DS002-000-03	Component Design Specification, UCP
DS901-000-01	HFC-SBC06 Module Detailed Design Specification

DS901-000-02	HFC-6000 I/O Card Module Detailed Design Specification
DS901-000-03	HFC-DO8J Card Module Detailed Design Specification
DS901-000-04	HFC-DI16I Module Detailed Design Specification
DS901-000-05	HFC-DC33 Module Detailed Design Specification
DS901-000-06	HFC-DC34 Module Detailed Design Specification
DS901-000-07	HFC-AI16F Module Detailed Design Specification
DS901-000-08	HFC-AO8F Module Detailed Design Specification
DS901-000-09	HFC-AC36 Module Detailed Design specification
DS901-000-10	HFC-AI8L Module Detailed Design Specification
DS901-000-11	HFC-AI8M Module Detailed Design Specification
DS901-000-12	HFC-AI4K Module Detailed Design Specification
DS901-000-14	HFC-BPC01-08 Hardware Design Specification
DS901-000-19	HFC-PSR06 Hardware Design Specification
DS901-000-21	FPC06 Controller Software Design Specification
DS901-000-23	HFC-PCC06 ICL I/O Processor Software Design Specification
MS901-000-01	HFC-SBC06 Module Design Specification
MS901-000-02	HFC-6000 I/O Card Module Design Specification
ATP0402	Application Software Objects Test Plan
CGSE001	Commercial Grade Software Evaluation, QNX Operating System
TP0401	Integration Test Plan
TP0402	Operability Test Procedure
TP0403	Prudency Test Procedure
TP0404	Environmental Stress Test Procedure
TP0405	Seismic Test Procedure
TP0406	Surge Withstand Test Procedure
TP0407	EMI-RFI Test Procedure
TP0408	TSAP Validation Test Procedure
TP0409	ESD Test Procedure
TP0410	Burn-in Test Procedure
TP0411	Isolation Test Procedure

4.3 SUMMARY OF ACTIVITIES

Detailed definitions of hardware and software designs were accomplished during this lifecycle phase. In addition, source code evaluation reports and detailed design descriptions for legacy software components were generated to support commercial grade dedication of COTS software that had been developed by HFC and had many years of operating history.

4.3.1 MODULE DESIGN SPECIFICATIONS

[

.]

4.3.2 LEGACY SOFTWARE COMPONENT DESIGN SPECIFICATIONS

[

]

4.3.2.1

Pseudo Code Analysis

[

-]

4.3.2.2 **Source Code Review**

[

.]

4.3.2.3 **Source Code Design Descriptions**

[

]

4.3.3 TEST SPECIMEN DESIGN

[

.]

4.3.4 TEST PROCEDURES

Each of the detailed test procedures identified in TN0401 were generated and reviewed internally and then submitted to external consultants for evaluation. Hurst Technologies provided guidance and critical evaluations during the initial development of these test procedures.

4.4 EXTERNAL AUDIT

[

]

4.5 ASSESSMENT OF OVERALL QUALITY

[

]

4.6 CONCLUSIONS AND RECOMMENDATIONS

HFC revised its internal procedures covering design development and its approach to component documentation based on comments from consultant experts. Because the internal processes were being developed in parallel with the HFC-6000 components, implementation of new procedures had to be phased into execution of the on-going project. Now that internal procedures have been developed to cover development of safety-related equipment and systems, HFC must ensure that those procedures are followed consistently.

Section 5

IMPLEMENTATION PHASE

5.1 SCOPE

The implementation phase is the period during which hardware components are fabricated and software code is written. The only new software that was generated prior to execution of the qualification tests was the TSAP and the application software for the HPAT. (The HPAT software controlled the automated tests and is external to the qualification system.) Implementation of the program code for the PCC06 and the FPC06 continued while the qualification tests were being run. All other tasks related to hardware fabrication only.

5.2 REFERENCES

The following V&V documents and reports were generated during the implementation phase for hardware and software components:

400401-02	HFC-ILR06 Test Procedure
400419-02	HFC-SBC06 Controller, HFC-DPM06 Dual Port Memory, HFC-BPC01-19 Backplane, and HFC-BPE01-19 Backplane Test Procedure
400429-02	HFC-PCC06 Test Procedure
400439-02	HFC-AI8L Test Procedure
400444-02	HFC-AI8M Test Procedure
700715-00	Test Specimen I/O Wiring Schematic
700907-01	TSAP Logic Diagram – Main Controller
700907-02	TSAP Logic Diagram – Single Loop Controller
700908-10	HPAT Logic Diagram
700909-01	System Arrangement
700910-01	Card Layout, Class Q Safety-Related USNRC Test Racks
700910-02	Card Layout, Class S Non-Safety Related USNRC Tester
700911-01	HFC-6000 Seismic Test Equipment Layout
700912-01	System Assembly Drawing
700916-01	Power Distribution, TSAP Configuration
700916-02	Power Distribution, Single Loop Configuration
700918-01	Equipment Assembly, Single Loop Hoffman Cabinet
ASC0401	TSAP Program Listing
DS001-000-07	HFC-6000 Software Configuration for Nuclear Safety Equipment Controller
TS001-000-01	Component Test Procedure
TS901-000-01	HFC-PCC06 Detailed Prototype Test Procedure
TS901-000-02	HFC-SBC06/DPM06 Prototype Test Procedure
TS901-000-03	HFC-AC36 Detailed Prototype Test Procedure
TS901-000-04	HFC-AI16F Prototype Test Procedure

TS901-000-05	HFC-AI4K Prototype Test Procedure
TS901-000-06	HFC-AI8L Detailed Prototype Test Procedure
TS901-000-07	HFC-AI8M Prototype Test Procedure
TS901-000-08	HFC-AO8F Prototype Test Procedure
TS901-000-09	HFC-DC33 Prototype Test Procedure
TS901-000-10	HFC-DC34 Prototype Test Procedure
TS901-000-11	HFC-DI16I Prototype Test Procedure
TS901-000-12	HFC-DO8J Prototype Test Procedure
TS901-000-13	HFC-AC38 Detailed Prototype Test Procedure
TS901-000-14	HFC-PCC06 Detailed Test Procedure
TS901-000-16	HFC-BPC01-08 Detailed Prototype Test Procedure
TS901-000-43	HFC-PCC06 Code Functional Test
TR001-000-01	Block Algorithm Test Report

5.3 SUMMARY OF ACTIVITIES

5.3.1 HARDWARE FABRICATION

HFC uses external commercial vendors to fabricate PCB assemblies. The documentation necessary to support prototype fabrication were created, reviewed, and approved during the design phase, and this package was supplied to the vendor. As prototypes became available, the hardware designer verified functional operation of the components. Because this was viewed as part of the hardware design process, no specific software V&V activities applied. After the designer was satisfied that all design errors had been identified and corrected, the assemblies were released for production. The final release package for each PC assembly includes the following items:

- Schematic diagram.
- Assembly diagram and gerber file.
- Bill of Material.
- Production test procedure.
- PROM code files and CPLD code files (if applicable).

Reviews from outside consultants indicated that the production-level test procedures did not provide adequate formal documentation for prototype approval. Consequently, HFC developed an additional prototype test procedure for each module and then executed those tests to document module performance characteristics.

5.3.2 PROGRAM CODE IMPLEMENTATION TASKS

5.3.2.1 TSAP and HPAT Source Code Review

[

]

5.3.2.2 Application Software Objects Acceptance Test

[

.]

5.3.2.3 Legacy Software Component Tests

[

]

5.3.2.4 HFC-PCC06 Software Test

[

]

5.4 ASSESSMENT OF OVERALL QUALITY

5.4.1 HARDWARE IMPLEMENTATION

Fabrication of all hardware components was accomplished by qualified vendors, and the quality of the resulting components was validated by the test procedure covering that component. Records of all inspections and tests were included in HFC quality records for ERD111. The scope and formality of prototype testing that had been accomplished was deemed inadequate for safety-related applications. To correct this discrepancy, HFC generated a set of detailed prototype tests covering each of the new modules. The new prototype tests were then executed and the results documented.

5.4.2 SYSTEM SOFTWARE IMPLEMENTATION

Nearly all of the system software is based on legacy software that has been used for many years. The general approach was to make minimal changes to the software in either structure or function. Consequently, the V&V tasks for implementation phase consisted of generating the documentation that would be needed to support commercial grade dedication.

5.4.3 TSAP AND HPAT CODE

[

.]

5.5 CONCLUSIONS AND RECOMMENDATIONS

HFC has altered its internal practices to include formally documented prototype tests in addition to the functional tests used for production assemblies. This was a new step not previously done. HFC needs to ensure that this exercise is accomplished for all new hardware and software development. For this purpose, the WI-ENG-107, "Development of Hardware Test Procedures,' was modified to specify the scope of testing.

Section 6

INTEGRATION PHASE

6.1 SCOPE

During the integration phase of the life cycle, the hardware components of the Test Specimen are assembled together and electrically interconnected. Scope of activities during this phase consists primarily of inspections to verify proper assembly of all constituent components.

6.2 REFERENCES

The following V&V documents and reports were generated during the implementation phase for hardware and software components:

TN0401	Master Test Plan
TP0401	Integration Test Plan
TP0408	TSAP Validation Test Procedure
TP0410	Burn-in Test Procedure
TS901-000-30	Summary Report for Burn-in Test, Setup and Checkout Test, TSAP Validation Test

6.3 SUMMARY OF ACTIVITIES

Basic fabrication of components was accomplished by outside qualified vendors, who manufactured the hardware assemblies as commercial grade equipment. The resulting assemblies were received at HFC and inspected under the HFC QA program. When a functional module was accepted, it was scheduled for the burn-in test. Nonfunctional hardware components were assembled by HFC personnel in accordance with process control sheets and relevant assembly diagrams. Once all of the functional components had completed burn-in, the modules were installed in the individual modular chassis prior to the start of the integration test.

6.3.1 BURN-IN TEST

[

]

6.3.2 INTEGRATION TEST

[

.]

6.3.3 VALIDATION TEST

[

•

.]

6.4 TEST RESULTS

[

]

6.5 ASSESSMENT OF OVERALL QUALITY

All the tests except TSAP validation related to hardware performance. No hardware failures were detected at this time. During TSAP validation, several design and implementation errors were detected and corrected. Since the test algorithms were designed 6 months before any part of the hardware existed, existence of these errors is not prejudicial to the Test Specimen or the TSAP.

6.6 CONCLUSIONS AND RECOMMENDATIONS

Completion of this phase of the testing was satisfactory.

Section 7

ACCEPTANCE TESTING PHASE

7.1 SCOPE

The first execution of the Operability and Prudency tests were accomplished during this phase prior to the start of the qualification tests. After completion of the qualification tests, HFC management decided that the complete seismic test would need to be repeated. Consequently, both the Operability and the Prudency tests were completely repeated prior to execution of the seismic retest.

7.2 REFERENCES

The following documents define the tests that were conducted during this phase of the project.

TP0402	Operability Test Procedure
TP0403	Prudency Test Procedure
TP0408	TSAP Validation Test Procedure
TP0408B	Test Specimen Validation Test Procedure
TS901-000-22	Baseline Testing Summary Report
TS901-000-30	Summary Report for Burn-in Test, Setup and Checkout Test, TSAP Validation Test

7.3 SUMMARY OF ACTIVITIES

The primary purpose of the tests run at this time was to establish a performance baseline for the Test Specimen. This baseline performance would then be used during the qualification tests to detect any degradation caused by the stress conditions being imposed on the system hardware. These tests all focused on hardware performance, not execution of the software

7.3.1 INITIAL ACCEPTANCE TEST

7.3.1.1 Operability Test

[

]

7.3.1.2 Prudency Test

[

]

7.3.2 ACCEPTANCE TESTING PRIOR TO SEISMIC RETEST

[

]

7.3.2.1 Test Specimen Validation

[

]

7.3.2.2 Operability Tests

[

.]

7.3.2.3 Prudency Test

[

]

7.4 TEST RESULTS

[

]

7.5 ASSESSMENT OF OVERALL QUALITY

[

]

7.6 CONCLUSIONS AND RECOMMENDATIONS

[

]

Section 8

QUALIFICATION TEST PHASE

8.1 INTRODUCTION

This section covers activities conducted during on site installation concluding with the execution of the field performance test.

8.2 REFERENCES

The following documents provided the basis for the tests conducted during this phase of the ERD111 project.

TP0402	Operability Test Procedure
TP0403	Prudency Test Procedure
TP0404	Environmental Stress Test Procedure
TP0405	Seismic Test Procedure
TP0406	Surge Withstand Test Procedure
TP0407	EMI-RFI Test Procedure
TP0408B	Test Specimen Validation Test Procedure
TP0409	ESD Test Procedure
TP0411	Isolation Test Procedure
TS901-000-22	Baseline Testing Summary Report
TS901-000-23	Environmental Testing Summary Report
TS901-000-25	EMI/RFI, ESD and SWC Testing Summary Report
TS901-000-28	Isolation Testing Summary Report
TS901-000-29	Post Qualification Testing Summary Report
TS901-000-34	Seismic Retest In-house Testing Summary Report
TS901-000-35	Seismic Testing Summary Report
TS901-000-39	AI Response Time Regression Test Report
TS901-000-42	Failover Regression Test Report
TR901-000-02	Mailbox Timeout Regression Test Report

8.3 SUMMARY OF ACTIVITIES

8.3.1 INITIAL QUALIFICATION TEST SEQUENCE

[

]

8.3.2 SEISMIC RETEST

[

.]

8.3.3 REGRESSION TESTS

[

]

8.3.3.1 Analog Response Time

[

]

8.3.3.2 Hardware Watchdog Timeout

[

]

8.3.3.3 Software Watchdog Timeout Period

[

.]

8.4 EXTERNAL AUDIT

[

]

8.5 OUTSTANDING ISSUES

[

]

ATTACHMENT 7.1

COMMERCIAL GRADE SOFTWARE EVALUATION

EVALUATION NUMBER: CGSE003

REVISION: A

Safety Related: [X] YES [ ] NO

MANUFACTURER: HF Controls

DESCRIPTION: HFC-SBC06 SAP System Firmware

SOFTWARE NUMBER: 9120906-11 Rev: CKSM 27D1

Assigned Engineer: Jonathan Taylor Date: 12/15/05

Technical Reviewer: Steve Yang Date: 12/16/05

Approved By: Allen Hsu Date: 12/16/05

V & V Review: Bob Cain Date: 12/16/05

COMMERCIAL GRADE SOFTWARE EVALUATION

Software P/N: 9120906-11
CGS CLASSIFICATION SHEET

CGSE003, Rev A

Rev: CKSM 27D1 Description: SAP Firmware

1. What is the safety function of the parent component in which this item is to be installed?

Controls safety system operation.

2. What is the function of the item?

SAP processor initialization.
Primary scans configured I/O modules, [

.]
Processor validates all data received from input modules.
[.]

3. What are the ways in which the item should fail in actual service?

Program error causes processor stall or invalid branching operation.
Program error results in lost or corrupted I/O data.
Program error disables required function.

4. Would any of these postulated failures of the item prevent the parent component or associated components from performing their intended safety function?

Incorrect program branching without processor stall could result in loss of I/O data.
Corruption of I/O data image could result in incorrect or unexpected control of an output channel.

COMMERCIAL GRADE SOFTWARE EVALUATION

Software P/N: 9120906-11
CGS CLASSIFICATION SHEET

CGSE003, Rev A

Rev: CKSM 27D1 Description: SAP Firmware

Yes [X] (Item has safety function)

No [ ] (Item has non-safety function)

Method (Mark all that apply}

A "YES" in all three criteria is required for the item to be classified as commercial grade.

		YES	NO
1	Is the item not subject to design or specification requirements that are unique to facilities or activities licensed or regulated by the USNRC?	[X]	[ ]
2	Is the item used in applications other than facilities or activities licensed or regulated by the USNRC?	[X]	[ ]
3	Is the item ordered from manufacturer/supplier on the basis of specifications set forth in the manufacturer's published description?	[X]	[ ]
	Commercial Grade Software Item	[X]	[ ]

PHYSICAL CRITICAL CHARACTERISTICS:

Source code files
Compiled object code designation
Module checksum generated during compilation and when PROM is encoded.

COMMERCIAL GRADE SOFTWARE EVALUATION

Software P/N: 9120906-11
CGS CLASSIFICATION SHEET

CGSE003, Rev A

Rev: CKSM 27D1 Description: SAP Firmware

PERFORMANCE CRITICAL CHARACTERISTICS:

Initialization control
System status monitoring and display
[.
]

DEPENDABILITY CRITICAL CHARACTERISTICS:

Built-in quality
Configuration control
Operating history
Problem reporting and resolution

METHOD (MARK ALL THAT APPLY):

[X] Method 1 - Special Tests and Inspections.

[X] Method 2 - Design Documentation and/or Commercial Grade survey of vendor.

[ ] Method 3 - Source Verification.

[X] Method 4 - Historical Performance Records.

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1

CGS CRITICAL CHARACTERISTICS SHEET

Software P/N: 9120906-11 **Description:** SAP Firmware **CGSE** 003 **Rev:** A

Physical Critical Characteristic	Acceptance Criteria	Method of Verification
CC#001 Source code files	Source files are controlled with limited access. Revision record is included in the header section of each source file. Source code reviews generated for critical source code modules. Module-level test executed for PDS modules.	Direct examination of source files. Source code review documentation Source code test reports Prototype test report
CC#002 Compiled object code designation	Object code file designated as 9120906.hex	Direct verification
CC#003 Module checksum and PROM part number	One byte compiler checksum value (39H) is inserted as the first byte of the object code header. Two byte object code checksum (27D1H) calculated at the time the PROM is programmed. Checksum value and PROM part number are printed on a label, and that label is affixed to the outside of the PROM.	Checksum 27D1H is generated by ChipWin program at the time the PROM is encoded. One byte checksum is generated by the compiler and inserted as the first byte of the program header. Physical verification of U7 label on assembly. Direct verification of two-byte checksum value in PROM header.

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1

CGS CRITICAL CHARACTERISTICS SHEET

Software P/N: 9120906-11 **Description:** SAP Firmware **CGSE** 003 **Rev:** A

Performance Critical Characteristic	Acceptance Criteria	Method of Verification
CC#004 Initialization control	Runs initialization tests per design specification to verify operability status.	Source code review. Source code test report.
CC#005 System status monitoring and display	Detects failure during initialization and prevents start of normal operation. Controls display to indicate normal cycling operation.	Prototype test report
CC#006 [ ]	[.]	TR901-000-02
CC#007 [ ]	[ ] [ ]	Source code review. Source code test report. Functional test.

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1

CGS CRITICAL CHARACTERISTICS SHEET

Software P/N: 9120906-11 **Description:** SAP Firmware **CGSE** 003 **Rev:** A

Dependability Critical Characteristic	Acceptance Criteria	Method of Verification
CC#008 Built-in quality	NQA1 QA program	Records of internal and external audits. Source code review records Source code test reports Prototype test report
CC#009 Configuration control	[.]	Direct inspection
CC#010 Operating history	Revision record in file headers indicate up to 20 years operating history, including the past several years at Ulchin NPP.	Direct inspection of source files CR and SCR database Cumulative operating history of generic operating system
CC#011 Problem reporting and resolution	Corrective action program in place and being used for evaluating and resolving reported problems. HFC supports CFR Part 21 reporting.	QPP 16.1 and CR/SCR database Record of reported errors for generic operating system QPP 16.3

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1

CGS CRITICAL CHARACTERISTICS SHEET

Software P/N: 9120906-11 **Description:** SAP Firmware **CGSE** 003 **Rev:** A

Record all M&TE used for this test:

M&TE CIN#	M&TE Desc.	Cal. Due Date	M&TE CIN#	M&TE Desc.	Cal. Due Date

Condition Report (CR): [ ] Yes [ ] No [ ] N/A If yes, identify below:
CR# [ ]Acceptable [ ] Not Acceptable Initials/Date: _____
CR# [ ]Acceptable [ ] Not Acceptable Initials/Date: _____
CR# [ ]Acceptable [ ] Not Acceptable Initials/Date: _____
Other:

COMMERCIAL GRADE SOFTWARE EVALUATION
ATTACHMENT 7.1
CGS CRITICAL CHARACTERISTICS SHEET

Software P/N: 9120906-11 **Description:** SAP Firmware **CGSE** 003 **Rev:** A

Sheet ____ **of** ____

CC#	P/F/N	Preformed By	Date	Verified By	Date
001					
002					
003					
004					
004					
006					
007					
008					
009					
010					
011					

P = Pass, F = Fail, N = Not Applicable

Characteristics Record Sheet Competition:

Verified By _____ **Approved By** _____
Date:_____ **Date:**_____

ATTACHMENT 7.1

HFC

COMMERCIAL GRADE SOFTWARE EVALUATION

EVALUATION NUMBER: CGSE005

REVISION: A

Safety Related: [Y] YES [ ] NO

MANUFACTURER: HF Controls

DESCRIPTION: HFC-AI16F Firmware

SOFTWARE NUMBER: 9120680-17 Rev: CKSM A58B

Assigned Engineer: Jonathan Taylor Date: 12/15/05

Technical Reviewer: Steve Yang Date: 12/15/05

Approved By: Allen Hsu Date: 12/15/05

V & V Review: Bob Cain Date: 12/16/05

INFORMATIONAL ONLY

COMMERCIAL GRADE SOFTWARE EVALUATION

Software P/N: 9120680-17
CGS CLASSIFICATION SHEET

CGSE005, Rev A

Rev: CKSM A58B Description: HFC-AI16F Firmware

1. What is the safety function of the parent component in which this item is to be installed?

   Controls safety system operation.

2. What is the function of the item?

   Controls internal operation of 16-channel 4- to 20-mA analog input module.

3. What are the ways in which the item should fail in actual service?

   Program error causes processor stall or invalid branching operation.

   Program error results in corruption of AI images.

   Program error results in loss of communication with controller.

4. Would any of these postulated failures of the item prevent the parent component or associated components from performing their intended safety function?

   Corruption of AI image results in invalid control of field process.

   Loss of communication with controller or loss of image data prevents reliable control of field process.

   Yes [X] (Item has safety function)

   No [ ] (Item has non-safety function)

COMMERCIAL GRADE SOFTWARE EVALUATION

Software P/N: 9120680-17
CGS CLASSIFICATION SHEET

CGSE005, Rev A

Rev: CKSM A58B Description: HFC-AI16F Firmware

Method (Mark all that apply}

A "YES" in all three criteria is required for the item to be classified as commercial grade.

		YES	NO
1	Is the item not subject to design or specification requirements that are unique to facilities or activities licensed or regulated by the USNRC?	[X]	[ ]
2	Is the item used in applications other than facilities or activities licensed or regulated by the USNRC?	[X]	[ ]
3	Is the item ordered from manufacturer/supplier on the basis of specifications set forth in the manufacturer's published description?	[X]	[ ]
	Commercial Grade Software Item	[X]	[ ]

PHYSICAL CRITICAL CHARACTERISTICS:

Source code files
Compiled object code designation
Module checksum generated at the time that the PROM is encoded.

PERFORMANCE CRITICAL CHARACTERISTICS:

Initialization control
Internal status monitoring
Hardware watchdog strobe function
AI scan cycle control
Checks calibration and corrects input image for component drift.
Controls alarm for out-of-range input signal.
Response to messages from the HFC-6000 controller.

COMMERCIAL GRADE SOFTWARE EVALUATION

Software P/N: 9120680-17
CGS CLASSIFICATION SHEET

CGSE005, Rev A

Rev: CKSM A58B Description: HFC-AI16F Firmware

DEPENDABILITY CRITICAL CHARACTERISTICS:

Built-in quality
Configuration Control
Operating History
Problem Reporting and Resolution

METHOD (MARK ALL THAT APPLY):

[X] Method 1 - Special Tests and Inspections.

[X] Method 2 - Design Documentation and/or Commercial Grade survey of vendor.

[ ] Method 3 - Source Verification.

[X] Method 4 - Historical Performance Records.

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1
CGS CRITICAL CHARACTERISTICS SHEET

Software P/N: 9120680-17 Description: HFC-AI16 Firmware CGSE 005 Rev: A

Physical Critical Characteristic	Acceptance Criteria	Method of Verification
CC#001 Source Code Files	[ ] Revision record is included in the header section of each source code file. Source files are under configuration control. Source code reviews generated for critical source code files.	Direct examination of source file. Source code review documentation.
CC#002 Compiled object code designation	Object code file is designated 912068017.hex.	Direct verification
CC#003 Module checksum and PROM part number.	Two-byte checksum (A58B) calculated at the time that the PROM is programmed. PROM part number and checksum value are printed on a label, and this label is attached to the PROM.	Checksum A58B is generated by ChipWin program at the time the PROM is encoded. Physical verification of label on U5.

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1
CGS CRITICAL CHARACTERISTICS SHEET

Software P/N: 9120680-17 Description: HFC-AI16F Firmware CGSE 005 Rev: A

Performance Critical Characteristic	Acceptance Criteria	Method of Verification
CC#004 Initialization control	Runs initialization tests per design specification to verify operability status.	Source code review Prototype test report
CC#005 Controls watchdog strobe	Per design specification	Prototype test report
CC#006 Internal Status Monitoring	Maintains count of communication errors Detects out-of-calibration status and sets alarm flag Detects loss of ICL communication, and enables hunt mode	Source code review of algorithm Prototype test report
CC#007 Input scan function	Per design specification	Source code review of algorithm Prototype test report
CC#008 Checks calibration and adjusts input image for component drift	Per design specification Assembly operates within specified tolerance after calibration and remains within tolerance under environmental stress.	Source code review of algorithm Prototype test report Qualification test report
CC#009 Controls alarm for out-of-range input signal	Controls alarm indication for out-of-range input signal per design specification	Source code review of algorithm Prototype test report
CC#010 ICL response function	Responds to defined ICL messages and per design specification	Source code review of algorithm Prototype test report

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1
CGS CRITICAL CHARACTERISTICS SHEET

Software P/N: 9120680-17 Description: HFC-AI16F Firmware CGSE 005 Rev: A

Dependability Critical Characteristic	Acceptance Criteria	Method of Verification
CC#011 Built-in quality	NQA1 QA Program	Records of internal and external audits. Software review records. Prototype test records.
CC#012 Configuration Control	[ ]	Direct inspection
CC#013 Operating History	Revision record in file header indicates up to 15 years of operating history, including the past several years at Ulchin NPP.	Direct inspection CR and SCR database
CC#014 Problem Reporting and Resolution	Corrective action process is in place and being used for evaluating and resolving reported problems. HFC supports CFR Part 21 reporting	QPP 16.1 and CR/SCR database QPP 16.3

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1
CGS M&TE SHEET

Software P/N: 9120680-17 **Description:** HFC-AI16F Firmware CGSE 005 **Rev:** A

Record all M&TE used for this test:

M&TE CIN#	M&TE Desc.	Cal. Due Date	M&TE CIN#	M&TE Desc.	Cal. Due Date

Condition Report (CR): [ ] Yes [ ] No [ ] N/A If yes, identify below:
CR# [ ]Acceptable [ ] Not Acceptable Initials/Date: _____
CR# [ ]Acceptable [ ] Not Acceptable Initials/Date: _____
CR# [ ]Acceptable [ ] Not Acceptable Initials/Date: _____
Other:

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1

CGS CRITICAL CHARACTERISTICS RECORD SHEET

Software P/N: 9120680-17 **Description:** HFC-AI16F Firmware CGSE 005 **Rev** A

Sheet ___ **of** ___

CC#	P/F/N	Performed By	Date	Verified By	Date
001					
002					
003					
004					
005					
006					
007					
008					
009					
010					
011					
012					
013					
014					

P = Pass, F = Fail, N = Not Applicable

Characteristics Record Sheet Competition:

Verified By _____ **Approved By** _____

Date:_____ **Date:**_____

ATTACHMENT 7.1

COMMERCIAL GRADE SOFTWARE EVALUATION

EVALUATION NUMBER: CGSE007

REVISION: A

Safety Related: [Y] YES [ ] NO

MANUFACTURER: HF Controls

DESCRIPTION: HFC-AI8M Firmware

SOFTWARE NUMBER: 9120682-14 Rev: CKSM 8178

Assigned Engineer: Jonathan Taylor Date: 12/15/05

Technical Reviewer: Steve Yang Date: 12/16/05

Approved By: Allen Hsu Date: 12/16/05

V & V Review: Bob Cain Date: 12/16/05

COMMERCIAL GRADE SOFTWARE EVALUATION

Software P/N: 9120682-14
CGS CLASSIFICATION SHEET

CGSE007, Rev A

Rev: CKSM 8178 Description: HFC-AI8M Firmware

1. What is the safety function of the parent component in which this item is to be installed?

   Controls safety system operation.

2. What is the function of the item?

   Controls internal operation of an 8-channel RTD input module.

3. What are the ways in which the item should fail in actual service?

   Program error causes processor stall or invalid branching operation.

   Program error results in corruption of data images.

   Program error results in loss of communication with controller.

4. Would any of these postulated failures of the item prevent the parent component or associated components from performing their intended safety function?

   Corruption of AI image results in invalid control of field process.

   Loss of communication with controller or loss of image data prevents reliable control of field process.

Yes [X] (Item has safety function)

No [ ] (Item has non-safety function)

COMMERCIAL GRADE SOFTWARE EVALUATION

Software P/N: 9120682-14
CGS CLASSIFICATION SHEET

CGSE007, Rev A

Rev: CKSM 8178 Description: HFC-AI8M Firmware

Method (Mark all that apply}

A "YES" in all three criteria is required for the item to be classified as commercial grade.

		YES	NO
1	Is the item not subject to design or specification requirements that are unique to facilities or activities licensed or regulated by the USNRC?	[X]	[ ]
2	Is the item used in applications other than facilities or activities licensed or regulated by the USNRC?	[X]	[ ]
3	Is the item ordered from manufacturer/supplier on the basis of specifications set forth in the manufacturer's published description?	[X]	[ ]
	Commercial Grade Software Item	[X]	[ ]

PHYSICAL CRITICAL CHARACTERISTICS:

Source code files
Compiled object code designation
Module checksum generated at the time that the PROM is encoded.

COMMERCIAL GRADE SOFTWARE EVALUATION

Software P/N: 9120682-14
CGS CLASSIFICATION SHEET

CGSE007, Rev A

Rev: CKSM 8178 Description: HFC-AI8M Firmware

PERFORMANCE CRITICAL CHARACTERISTICS:

Initialization control
Internal status monitoring
[ ].
Controls AI scan function.
Checks calibration and corrects input image for component drift.
Controls alarm for out-of-range input signal.
Hardware watchdog strobe function
Response to messages from the HFC-6000 controller.

DEPENDABILITY CRITICAL CHARACTERISTICS:

Built-in quality
Configuration control
Operating history
Problem reporting and resolution

METHOD (MARK ALL THAT APPLY):

[X] Method 1 - Special Tests and Inspections.

[X] Method 2 - Design Documentation and/or Commercial Grade survey of vendor.

[ ] Method 3 - Source Verification.

[X] Method 4 - Historical Performance Records.

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1

CGS CRITICAL CHARACTERISTICS SHEET

Software P/N: 9120682-14 **Description:** HFC-AI8M Firmware CGSE 007 **Rev:** A

Physical Critical Characteristic	Acceptance Criteria	Method of Verification
CC#001 Source Code Files	[ ]. Revision record is included in the header section of each source code file. Source code files are under configuration control. Source code reviews generated for critical source code files.	Direct examination of source file. Source code review documentation
CC#002 Compiled object code designation	Object code file is designated 912068214.hex.	Direct verification
CC#003 Module checksum and PROM part number.	Two-byte checksum (8178) calculated at the time that the PROM is programmed. PROM part number and checksum value are printed on a label, and this label is attached to the PROM.	Checksum 8178 is generated by ChipWin program at the time the PROM is encoded. Physical verification of label on U5.

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1

CGS CRITICAL CHARACTERISTICS SHEET

Software P/N: 9120682-14 **Description:** HFC-AI8M Firmware CGSE 007 **Rev:** A

Performance Critical Characteristic	Acceptance Criteria	Method of Verification
CC#004 Initialization control	Runs initialization tests per design specification to verify operability.	Source code review Prototype test report
CC#005 Internal status monitoring function	Maintains count of communication errors. Senses out-of tolerance calibration status and sets alarm flag. [	Source code review Prototype test report
CC#006 Saves calibration data in EEPROM	Per design specification and Test procedure.	Assembly calibration record
CC#007 Controls AI scan function	Per design specification	Source code review Prototype test report
CC#008 Checks calibration and adjusts input image for component drift	Per design specification Assembly operates within tolerance after calibration and remains within tolerance under environmental stress.	Source code review Prototype test report Qualification test report
CC#009 Controls alarm for out-of-range input signal	Activates alarm for over range, under range, and open RTD element	Source code review Prototype test report
CC#010 Controls watchdog strobe	Per design specification	Source code review Prototype test report
CC#011 ICL response function	Responds to defined ICL messages per design specification	Source code review Prototype test report

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1

CGS CRITICAL CHARACTERISTICS SHEET

Software P/N: 9120682-14 **Description:** HFC-AI8M Firmware CGSE 007 **Rev:** A

Dependability Critical Characteristic	Acceptance Criteria	Method of Verification
CC#012 Built-in quality	NQA1 QA Program	Records of internal and external audits. Software review records. Pototype test records.
CC#013 Configuration control	[	Direct inspection
CC#014 Operating history	Revision record in file header indicates up to 15 years operating history, including the past several years at Ulchin NPP	Direct inspection CR and SCR database
CC#015 Problem reporting and resolution	Corrective action process is in place and being used for evaluating and resolving reported problems. HFC supports CFR Part 21 reporting	QPP 16.1 and CR/SCR database QPP 16.3

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1
CGS M&TE SHEET

Software P/N: 9120682-14 **Description:** HFC-AI8M Firmware CGSE 007 **Rev:** A

Record all M&TE used for this test:

M&TE CIN#	M&TE Desc.	Cal. Due Date	M&TE CIN#	M&TE Desc.	Cal. Due Date

Condition Report (CR): [ ] Yes [ ] No [ ] N/A If yes, identify below:
CR# [ ]Acceptable [ ] Not Acceptable Initials/Date: _____
CR# [ ]Acceptable [ ] Not Acceptable Initials/Date: _____
CR# [ ]Acceptable [ ] Not Acceptable Initials/Date: _____
Other:

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1

CGS CRITICAL CHARACTERISTICS RECORD SHEET

Software P/N: 9120682-14 **Description:** HFC-AI8M Firmware CGSE 007 **Rev:** A

Sheet ___ **of** ___

CC#	P/F/N	Performed By	Date	Verified By	Date
001					
002					
003					
004					
005					
006					
007					
008					
009					
010					
011					
012					
013					
014					
015					

P = Pass, F = Fail, N = Not Applicable

Characteristics Record Sheet Competition:

Verified By _____ **Approved By** _____

Date:_____ **Date:**_____

ATTACHMENT 7.1

HFC

COMMERCIAL GRADE SOFTWARE EVALUATION

EVALUATION NUMBER: CGSE009

REVISION: A

Safety Related: [Y] YES [ ] NO

MANUFACTURER: HF Controls

DESCRIPTION: HFC-DC33 Firmware

SOFTWARE NUMBER: 9120676-14 Rev: CKSM 2C30

Assigned Engineer: Jonathan Taylor Date: 12/15/05

Technical Reviewer: Steve Yang Date: 12/16/05

Approved By: Allen Hsu Date: 12/15/05

V & V Review: Bob Cain Date: 12/16/05

INFORMATIONAL ONLY

COMMERCIAL GRADE SOFTWARE EVALUATION

Software P/N: 9120676-14
CGS CLASSIFICATION SHEET

CGSE009, Rev A

Rev: CKSM 2C30 Description: HFC-DC33 Firmware

1. What is the safety function of the parent component in which this item is to be installed?

   Controls safety system operation.

2. What is the function of the item?

   Controls internal operation of 12-channel DI, 2-channel 120-vac DO module with coil continuity monitoring.

3. What are the ways in which the item should fail in actual service?

   Program error causes processor stall or invalid branching operation.

   Program error prevents transfer of image data to output channels or corrupts DI image.

   Program error results in loss of communication with controller.

4. Would any of these postulated failures of the item prevent the parent component or associated components from performing their intended safety function?

   Failure to update DO channels disrupts control of field process.

   Loss of DI data prevents reliable status monitoring of field process.

   Loss of communication with controller or loss of image data prevents reliable control of field process.

   Yes [X] (Item has safety function)

   No [ ] (Item has non-safety function)

COMMERCIAL GRADE SOFTWARE EVALUATION

Software P/N: 9120676-14
CGS CLASSIFICATION SHEET

CGSE009, Rev A

Rev: CKSM 2C30 Description: HFC-DC33 Firmware

Method (Mark all that apply}

A "YES" in all three criteria is required for the item to be classified as commercial grade.

		YES	NO
1	Is the item not subject to design or specification requirements that are unique to facilities or activities licensed or regulated by the USNRC?	[X]	[ ]
2	Is the item used in applications other than facilities or activities licensed or regulated by the USNRC?	[X]	[ ]
3	Is the item ordered from manufacturer/supplier on the basis of specifications set forth in the manufacturer's published description?	[X]	[ ]
	Commercial Grade Software Item	[X]	[ ]

PHYSICAL CRITICAL CHARACTERISTICS:

Source code files
Compiled object code designation
Module checksum generated at the time that the PROM is encoded.

PERFORMANCE CRITICAL CHARACTERISTICS:

Initialization control
Internal status monitoring
Responds to messages from the HFC-6000 controller.
Controls digital I/O scan function.
Hardware watchdog strobe function

COMMERCIAL GRADE SOFTWARE EVALUATION

Software P/N: 9120676-14
CGS CLASSIFICATION SHEET

CGSE009, Rev A

Rev: CKSM 2C30 Description: HFC-DC33 Firmware

DEPENDABILITY CRITICAL CHARACTERISTICS:

Built-in quality
Configuration control
Operating history
Problem reporting and resolution

METHOD (MARK ALL THAT APPLY):

[X] Method 1 - Special Tests and Inspections.

[X] Method 2 - Design Documentation and/or Commercial Grade survey of vendor.

[ ] Method 3 - Source Verification.

[X] Method 4 - Historical Performance Records.

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1

CGS CRITICAL CHARACTERISTICS SHEET

Software P/N: 9120676-14 **Description:** HFC-DC33 Firmware CGSE 009 **Rev:** A

Physical Critical Characteristic	Acceptance Criteria	Method of Verification
CC#001 Source Code Files	[Revision record is included in the header section of each source file. Source code files are under configuration control. Source code reviews generated for critical source code files.	Direct examination of source files Source code review documentation
CC#002 Compiled object code designation	Object code file is designated 912067614.hex.	Direct verification
CC#003 Module checksum and PROM part number.	Two-byte checksum (2C30) calculated at the time that the PROM is programmed. PROM part number and checksum value are printed on a label, and this label is attached to the PROM.	Checksum 2C30 is generated by ChipWin program at the time the PROM is encoded. Physical verification of label on U8.

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1

CGS CRITICAL CHARACTERISTICS SHEET

Software P/N: 9120676-14 Description: HFC-DC33 Firmware CGSE 009 Rev: A

Performance Critical Characteristic	Acceptance Criteria	Method of Verification
CC#004 Initialization control	Runs initialization tests per design specification to verify operability.	Source code review Prototype test report
CC#005 Monitors overall hardware status	Monitors excitation voltage status, 120 vac status, and coil continuity status. Maintains count of communication errors. [	Source code review Prototype test report
CC#006 Controls I/O scan function	Per design specification	Source code review Prototype test report
CC#007 ICL response function	Responds to defined ICL messages per design specification.	Source code review Prototype test report
CC#008 Controls watchdog strobe	Per design specification	Source code review Prototype test report

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1

CGS CRITICAL CHARACTERISTICS SHEET

Software P/N: 9120676-14 **Description:** HFC-DC33 Firmware CGSE 009 **Rev:** A

Dependability Critical Characteristic	Acceptance Criteria	Method of Verification
CC#009 Built-in quality	NQA1 QA Program	Records of internal and external audits. Software review records. Pototype test records. Up to 20 years operating history for component program files.
CC#010 Configuration control	[	Direct inspection
CC#011 Operating history	Revision record in source file headers indicate that all components have more than 4 years of operating history, including the past several years at Ulchin NPP.	Direct inspection CR and SCR database
CC#012 Problem reporting and resolution	Corrective action process is in place and being used for evaluating and resolving reported problems. HFC supports CFR Part 21 reporting	QPP 16.1 and CR/SCR database QPP 16.3

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1
CGS M&TE SHEET

Software P/N: 9120676-14 **Description:** HFC-DC33 Firmware CGSE 009 **Rev:** A

Record all M&TE used for this test:

M&TE CIN#	M&TE Desc.	Cal. Due Date	M&TE CIN#	M&TE Desc.	Cal. Due Date

Condition Report (CR): [ ] Yes [ ] No [ ] N/A If yes, identify below:
CR# [ ]Acceptable [ ] Not Acceptable Initials/Date: _____
CR# [ ]Acceptable [ ] Not Acceptable Initials/Date: _____
CR# [ ]Acceptable [ ] Not Acceptable Initials/Date: _____
Other:

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1

CGS CRITICAL CHARACTERISTICS RECORD SHEET

Software P/N: 9120676-14 **Description:** HFC-DC33 Firmware CGSE 009 **Rev:** A

Sheet ____ **of** ____

CC#	P/F/N	Performed By	Date	Verified By	Date
001					
002					
003					
004					
005					
006					
007					
008					
009					
010					
011					
012					

P = Pass, F = Fail, N = Not Applicable

Characteristics Record Sheet Competition:

Verified By _____ **Approved By** _____

Date:_____ **Date:**_____

ATTACHMENT 7.1

COMMERCIAL GRADE SOFTWARE EVALUATION

EVALUATION NUMBER: CGSE011

REVISION: A

Safety Related: [Y] YES [ ] NO

MANUFACTURER: HF Controls

DESCRIPTION: HFC-DI16I Firmware

SOFTWARE NUMBER: 9120686-14 Rev: CKSM 3A5B

Assigned Engineer: Jonathan Taylor Date: 12/15/05

Technical Reviewer: Steve Yang Date: 12/15/05

Approved By: Allen Hsu Date: 12/15/05

V & V Review: Bob Cain Date: 12/16/05

COMMERCIAL GRADE SOFTWARE EVALUATION

Software P/N: 9120686-14
CGS CLASSIFICATION SHEET

CGSE011, Rev A

Rev: CKSM 3A5B Description: HFC-DI16I Firmware

1. What is the safety function of the parent component in which this item is to be installed?

   Controls safety system operation.

2. What is the function of the item?

   Controls internal operation of 16-channel DI module.

3. What are the ways in which the item should fail in actual service?

   Program error causes processor stall or invalid branching operation.

   Program error corrupts DI image.

   Program error results in loss of communication with controller.

4. Would any of these postulated failures of the item prevent the parent component or associated components from performing their intended safety function?

   Corruption of DI data produces inaccurate indication of process status.

   Loss of communication with controller or loss of image data prevents reliable monitoring of process status.

   Yes [X] (Item has safety function)

   No [ ] (Item has non-safety function)

COMMERCIAL GRADE SOFTWARE EVALUATION

Software P/N: 9120686-14
CGS CLASSIFICATION SHEET

CGSE011, Rev A

Rev: CKSM 3A5B Description: HFC-DI16I Firmware

Method (Mark all that apply}

A "YES" in all three criteria is required for the item to be classified as commercial grade.

		YES	NO
1	Is the item not subject to design or specification requirements that are unique to facilities or activities licensed or regulated by the USNRC?	[X]	[ ]
2	Is the item used in applications other than facilities or activities licensed or regulated by the USNRC?	[X]	[ ]
3	Is the item ordered from manufacturer/supplier on the basis of specifications set forth in the manufacturer's published description?	[X]	[ ]
	Commercial Grade Software Item	[X]	[ ]

PHYSICAL CRITICAL CHARACTERISTICS:

Source code files
Compiled object code designation
Module checksum generated at the time that the PROM is encoded.

PERFORMANCE CRITICAL CHARACTERISTICS:

Initialization control
Monitors operating status
Responds to messages from the HFC-6000 controller.
Controls digital DI scan function.
Hardware watchdog strobe function

COMMERCIAL GRADE SOFTWARE EVALUATION

Software P/N: 9120686-14
CGS CLASSIFICATION SHEET

CGSE011, Rev A

Rev: CKSM 3A5B Description: HFC-DI16I Firmware

DEPENDABILITY CRITICAL CHARACTERISTICS:

Built-in quality
Configuration control
Operating history
Problem reporting and resolution

METHOD (MARK ALL THAT APPLY):

[X] Method 1 - Special Tests and Inspections.

[X] Method 2 - Design Documentation and/or Commercial Grade survey of vendor.

[ ] Method 3 - Source Verification.

[X] Method 4 - Historical Performance Records.

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1

CGS CRITICAL CHARACTERISTICS SHEET

Software P/N: 9120682-14 **Description:** HFC-AI8M Firmware CGSE 007 **Rev:** A

Physical Critical Characteristic	Acceptance Criteria	Method of Verification
CC#001 Source Code Files	[ ]. Revision record is included in the header section of each source code file. Source code files are under configuration control. Source code reviews generated for critical source code files.	Direct examination of source file. Source code review documentation
CC#002 Compiled object code designation	Object code file is designated 912068214.hex.	Direct verification
CC#003 Module checksum and PROM part number.	Two-byte checksum (8178) calculated at the time that the PROM is programmed. PROM part number and checksum value are printed on a label, and this label is attached to the PROM.	Checksum 8178 is generated by ChipWin program at the time the PROM is encoded. Physical verification of label on U5.

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1

CGS CRITICAL CHARACTERISTICS SHEET

Software P/N: 9120686-14 **Description:** HFC-DI16I Firmware CGSE 011 **Rev:** A

Performance Critical Characteristic	Acceptance Criteria	Method of Verification
CC#004 Initialization control	Runs initialization tests per design specification to verify operability.	Source code review Prototype test report
CC#005 Monitors overall hardware status	Monitors excitation voltage status. Maintains count of communication errors. [ ].	Source code review Prototype test report
CC#006 Controls DI scan function	Per design specification	Source code review Prototype test report
CC#007 ICL response function	Responds to defined ICL messages per design specification.	Source code review Prototype test report
CC#008 Controls watchdog strobe	Per design specification	Source code review Prototype test report

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1

CGS CRITICAL CHARACTERISTICS SHEET

Software P/N: 9120686-14 **Description:** HFC-DI16I Firmware CGSE 011 **Rev:** A

Dependability Critical Characteristic	Acceptance Criteria	Method of Verification
CC#009 Built-in quality	NQA1 QA Program	Records of internal and external audits. Software review records. Prototype test records.
CC#010 Configuration control	[ ].	Direct inspection
CC#011 Operating history	Revision record in source files indicate up to 15 years operating history, including the past several years at Ulchin NPP.	Direct inspection CR and SCR database
CC#012 Problem reporting and resolution	Corrective action program is in place and being used for evaluating and resolving reported problems. HFC supports CFR Part 21 reporting.	QPP 16.1 and CR/SCR database QPP 16.3

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1
CGS M&TE SHEET

Software P/N: 9120686-14 **Description:** HFC-DI16I Firmware CGSE 011 **Rev:** A

Record all M&TE used for this test:

M&TE CIN#	M&TE Desc.	Cal. Due Date	M&TE CIN#	M&TE Desc.	Cal. Due Date

Condition Report (CR): [ ] Yes [ ] No [ ] N/A If yes, identify below:
CR# [ ]Acceptable [ ] Not Acceptable Initials/Date: _____
CR# [ ]Acceptable [ ] Not Acceptable Initials/Date: _____
CR# [ ]Acceptable [ ] Not Acceptable Initials/Date: _____
Other:

COMMERCIAL GRADE SOFTWARE EVALUATION

ATTACHMENT 7.1

CGS CRITICAL CHARACTERISTICS RECORD SHEET

Software P/N: 9120686-14 **Description:** HFC-DI16I Firmware CGSE 011 **Rev:** A

Sheet ___ **of** ___

CC#	P/F/N	Performed By	Date	Verified By	Date
001					
002					
003					
004					
005					
006					
007					
008					
009					
010					
011					
012					

P = Pass, F = Fail, N = Not Applicable

Characteristics Record Sheet Competition:

Verified By _____ **Approved By** _____

Date:_____ **Date:**_____

HF Controls Corporation

Component Design Specification

HFC-Controller Equation Interpreter

DS001-000-02

Draft A

Effective Date 2-16-05

Author/Title ANGELA HAN

Reviewer/Title STEVE YANG

Approval/Title ALLEN HSU

[ ]

Copyright© 2004 HF Controls Corporation

Revision History

Date	Revision	Author	Changes
10/30/2004	0	J Taylor	Draft
2/2/2005	A	B Han	Release

Table of Contents

Number	Title	Page
1.0	INTRODUCTION	6
2.0	FUNCTIONAL DESCRIPTION	6
2.1	*Point Types*	*7*
2.2	*Equation Program Structure*	*7*
2.3	*CQ3 Program Struction*	*10*
2.4	*Instruction Processing*	*10*
2.5	*Program Register Assignments*	*12*
3.0	OVERVIEW OF EUQATION INTERPRETER OPERATION	12
3.1	*Initialization following Powerup*	*13*
3.2	*Normal Processing Sequences*	*13*
3.3	*Task Initialization*	*13*
3.4	*Application Code Processing (Primary Controller)*	*18*
3.5	*ADT_SCE Routine*	*19*
3.6	*Opcode Processing Routines*	*23*
3.6.1	Instruction Jump Table	23
3.6.1.1	NOOP_INS (Opcode 00_H and 60_H)	30
3.6.1.2	INIT_BINS	31
3.6.1.3	IF_X (Opcodes 01_H through $0B_H$)	31
3.6.1.4	OUT_X (Opcodes $0C_H$ through 15_H)	32
3.6.1.5	HLD_RSM_TI (Opcodes 17_H)	33
3.6.1.6	TST_X (Opcodes 19_H through 23_H)	34
3.6.1.7	CQ3AND and CQ3OR (Opcodes 24_H and 25_H)	36
3.6.1.8	GOTO_INS (Opcode $2C_H$)	36
3.6.1.9	JMP_INS (Opcode $2D_H$ or 61_H)	37
3.6.1.10	CALL_CANR (Opcode $2E_H$)	38
3.6.1.11	LIST_INS and LIST_INS_CO (OpcodeS $2F_H$ and 62_H)	38
3.6.1.12	GOBK_INS (Opcode 30_H)	39
3.6.1.13	MATH_INS and MATH_INS_ (Opcodes $3A_H$, $3B_H$, $B8_H$, and $B9_H$) 40	
3.6.1.14	TIME_C (Opcode $3C_H$ or $6B_H$)	40
3.6.1.15	DATE_C (Opcode $3D_H$ or $6C_H$)	41
3.6.1.16	GET_Digital Status	42
3.6.1.17	SAV_Digital Status	42
3.6.1.18	SAV_DF (Opcode 97_H)	43
3.6.1.19	AND_Digital Status	44
3.6.1.20	OR_Digital Status	44
3.6.1.21	SGET_Digital Status	45
3.6.1.22	SAVS_Digital Status	46
3.6.1.23	SAVR_Digital Status	47
3.6.1.24	SAVA_Point and MSG_Point	48
3.6.1.25	GET_REMxxx	49
3.6.1.26	IF_SKP (Opcode $7F_H$)	50

3.6.1.27 CO Point Instructions 50
3.6.1.28 TI Point Instructions 54
3.6.1.29 Instructions for Floating-Point BL Values 62
3.6.1.30 Miscellaneous Instructions 64
3.6.1.31 Parameter Instructions 71
3.6.2 Math Instruction Jump Table 72
3.6.2.1 Standard Arithmetic Operations 75
3.6.2.2 Comparison Operations 78
3.6.2.3 Combination Operations 81
3.6.2.4 SAV_TIPRV Operation (Opcode CE_H) 84
3.6.2.5 Parameter Math Instructions 85
3.7 Remote Data Jump Table 85
3.7.1 Remote Instruction Operations 87
3.7.1.1 GET_REMQW (Opcodes 80_H, 81_H, 82_H, 86_H, 87_H, and 88_H) 87
3.7.1.2 GET_REMTIS, GET_REMINT, GET_REMFLP 88
3.7.1.3 AND_REMQW (Opcodes $A0_H$, $A1_H$, $A2_H$, $A4_H$, $A5_H$, and $A6_H$) 88
3.7.1.4 AND_REMTIS (Opcode $A3_H$) 89
3.7.1.5 OR_REMQW (Opcodes $A9_H$, AA_H, AB_H, AD_H, AE_H, and AF_H) 90
3.7.1.6 OR_REMTIS (Opcode AC_H) 90
3.7.1.7 REM_MATHINS (Opcode $B8_H$) 91
3.7.1.8 SGET_x (Opcodes $C0_H$, $C1_H$, $C2_H$, $C3_H$, $C4_H$, $C5_H$, $C6_H$, $C7_H$, and $C8_H$) 91
3.7.2 REMMATHTBL Operations 92
3.8 Canned Routines Address Table 93
3.9 Parameter Instruction Processing 97
4.0 Equation Tasks Table 97
4.1 Equation Interpreter Status Table 97

List of Figures

Number **Title** **Page**

Figure 1. Controller Conversion of Inputs into Outputs 6
Figure 2. Mechanism for Processing Application Code Instructions 11
Figure 3. Equation Interpreter Initialization Sequence 16
Figure 4. Equation Interpreter Reentrant Path 16
Figure 5. Normal Instruction Processing Cycle for Primary Controller 18
Figure 6. ADT_SCE Processing for Primary 20
Figure 7. ADT_SCE Processing for Secondary 21

List of Tables

Number	Title	Page
Table 1.	Point Types Valid for Application Code	8
Table 2.	Source Code Header Structure	9
Table 3.	Four-Byte Instruction Format	10
Table 4.	Control Instruction Jump Table	24
Table 5.	Math Instruction Address Table	72
Table 6.	Get Remote Data Address Table	85
Table 7.	Remote Math Instruction Address Table	87
Table 8.	Canned Routines Address Table	93
Table 9.	CQ4 Algorithm Address Table	94
Table 10.	CQ4 Integer Algorithm Address Table	96
Table 11.	Equation Interpreter Status Table	97

1.0 INTRODUCTION

The Equation Interpreter runs as an automatic task of the Operating System (OS) for HFC controllers. After completion of power-up initialization, the OS calls the Equation Interpreter task, and it begins operation in a continuous loop. Normally, the Equation Interpreter task is the last task of OS execution cycle and will exhaust the majority portion of OS execution cycle time (100 ms usually). [

] As the Equation Interpreter processes each instruction, updated output images become available to other processors running on this controller or to other controllers operating in the same control system (Figure 1).

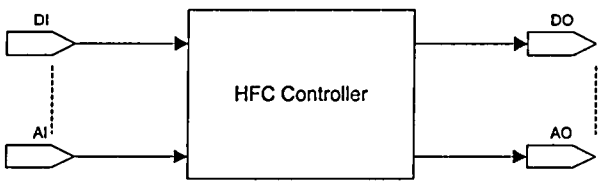


Figure 1. Controller Conversion of Inputs into Outputs

2.0 FUNCTIONAL DESCRIPTION

While the Equation Interpreter is running on a primary controller, it performs the following operations in a continuous sequence:

- Executes a set of preliminary instructions at the beginning of each processing cycle to verify internal operating status.
- Processes each instruction in the application program from the section of memory reserved for this program.
- Updates internal status and checks for UCP messages addressed to it.
- This operation runs as a continuous cycle until the end of OS execution cycle occurs.
- This operation has to complete at least one complete cycle, otherwise a system alarm will be reported.

While the Equation Interpreter is processing instructions from the application code, it fetches values from memory, performs the operation(s) defined by the instruction or passes control to another subroutine. Some of the instructions involve read access to

specified locations of memory, and others return data to specified locations of memory. Because the Equation Interpreter typically executes the same combination of instructions in the same sequence during each processing cycle, the values transferred to memory typically overwrite the contents generated during the previous processing cycle. The Interpreter normally continues this manner of operation until it reaches the last instruction in the application program. At that point the Interpreter updates its internal status, checks for any messages in its task queue, and then returns to the beginning of the program. [

]

2.1 POINT TYPES

The architecture of an HFC control system is based on a set of entities called 'points.' A particular point type within an HFC control system represents a specific function or range of functions within the control system. Each point type is represented by an entry in an Access database, and the point types assigned to each remote within a control system determine the content of the database directory for that remote. Certain point types (Table 1) constitute functional entities that can be used within an application program. All of these point types have a fixed data structure, and memory is allocated within a remote for its configured points during the software build for that remote. The application program uses the data structures associated with these point types to determine the digital image for inputs and the current operating status of the process under control; the various operations defined for the application program enable the controller to update the digital image of outputs as well as their operating status during each execution cycle.

2.2 EQUATION PROGRAM STRUCTURE

The source code file for an application program using equation statements consists of a fixed program header followed by a sequential list of program statements. The program header provides unique identification for this program file, enabling the user to distinguish one version of an application program from another version of the same program. The program statements may be represented either in text file format or in equivalent ladder format. In either case, each equation statement typically uses the current image of one or more points as input data, executes a specified Boolean or arithmetic operation to generate a result, and then writes that result into the data structure for another point.

The format of the text file is designed so that an application design engineer or technician can easily read the intended operation represented by each line of code. HFC has designed an equation compiler program that operates as one of the EWS (Engineering Workstation) suite of programs. [

]

Table 1. Point Types Valid for Application Code

Point Type	Definition	Structure

Point Type	Definition	Structure

Table 2. Source Code Header Structure

Name	Size (byte)	Description

[

Table 3. Four-Byte Instruction Format

Byte	Name	Description

2.3 CQ3 PROGRAM STRUCTION

CQ3 is a programming language for that was used to generate application code for HFC control systems before the equation language was created. A CQ3 program consists of a sequence of statements with each statement having a single Opcode and one or two operands. The Opcode for each statement determines the specific operation to be performed by the equation interpreter, and the operands supply the data to be manipulated. The point types valid for CQ3 programs and the general structure of object code instructions are generally identical to those for equation programs, but a different set of Opcode values are used for CQ3 programs. Macro statements in the source code distinguish between subroutines used for CQ3 support and those for regular equation statements. The CQ3 Opcode was used for HFC ECS-1200 product line only. It is not applicable for HFC-6000 product line.

2.4 INSTRUCTION PROCESSING

[

Figure 2. Mechanism for Processing Application Code Instructions

.]

2.5 PROGRAM REGISTER ASSIGNMENTS

The Equation Interpreter program module is written in ASM86 assembly language. This program employs the 8086/8088 register set for the following specific purposes:

- Register AX normally contains the first two bytes of each instruction statement. The low byte AL contains byte 0 (usually the bit mask), and the high byte AH contains byte 1 (Opcode).

- Register BX normally contains the second two bytes of each instruction statement, which usually provides the point offset parameter for the instruction.

- Register CX contains count data and is used for string operations and loops.

- Register DX is used for data. The low byte DL is used as the Boolean accumulator for logic operations. The high byte DH contains questionable status for the point being processed.

- The stack top serves as the value accumulator.

- DS:DI points to the start of the next instruction in the application code.

3.0 OVERVIEW OF EQUATION INTERPRETER OPERATION

[

]

3.1 INITIALIZATION FOLLOWING POWERUP

[

]

3.2 NORMAL PROCESSING SEQUENCES

[

]

3.3 TASK INITIALIZATION

[

:

Figure 3. Equation Interpreter Initialization Sequence

Figure 4. Equation Interpreter Reentrant Path

3.4 APPLICATION CODE PROCESSING (PRIMARY CONTROLLER)

[

Figure 5. Normal Instruction Processing Cycle for Primary Controller

]

.]

3.5 ADT_SCE ROUTINE

[

.]

Figure 6. ADT_SCE Processing for Primary

Figure 7. ADT_SCE Processing for Secondary

[

.]

3.6 OPCODE PROCESSING ROUTINES

[

.]

3.6.1 Instruction Jump Table

[

]

Table 4. Control Instruction Jump Table

Instruction	Opcode (hex)	Description

Table 4. Control Instruction Address Table (Cont)

Instruction	Opcode (hex)	Description

Table 4. Control Instruction Address Table (Cont)

Instruction	Opcode (hex)	Description

Table 4. Control Instruction Address Table (Cont)

Instruction	Opcode (hex)	Description

Table 4. Control Instruction Address Table (Cont)

Instruction	Opcode (hex)	Description

Table 4. Control Instruction Address Table (Cont)

Instruction	Opcode (hex)	Description

Table 4. Control Instruction Address Table (Cont)

Instruction	Opcode (hex)	Description

[

.]

3.6.1.1 NOOP_INS (Opcode 00_H and 60_H)

[

]

3.6.1.2 INIT_BINS

[

]

3.6.1.3 IF_X (Opcodes 01_H through $0B_H$)

[

Byte	Description

]

3.6.1.4 OUT_X (Opcodes $0C_H$ through 15_H)

[

Byte	Description

.]

3.6.1.5 HLD_RSM_TI (Opcodes 17_H)

[

Byte	Description

]

3.6.1.6 TST_X (Opcodes 19_H through 23_H)

[

Byte	Description

.]

3.6.1.7 CQ3AND and CQ3OR (Opcodes 24_H and 25_H)

[

Byte	Description

.]

3.6.1.8 GOTO_INS (Opcode $2C_H$)

[

Byte	Description

]

3.6.1.9 JMP_INS (Opcode $2D_H$ or 61_H)

[

Byte	Description

.]

3.6.1.10 CALL_CANR (Opcode $2E_H$)

[

Byte	Description

.]

3.6.1.11 LIST_INS and LIST_INS_CO (Opcodes $2F_H$ and 62_H)

[

LIST_INS Format	
Byte	**Description**

Byte	Description

]

3.6.1.12 GOBK_INS (Opcode 30_H)

[

.]

3.6.1.13 MATH_INS and MATH_INS_ (Opcodes $3A_H$, $3B_H$, $B8_H$, and $B9_H$)

[

]

3.6.1.14 TIME_C (Opcode $3C_H$ or $6B_H$)

[

Byte	Description

.]

3.6.1.15 DATE_C (Opcode $3D_H$ or $6C_H$)

[

Byte	Description

.]

3.6.1.16 GET_Digital Status

[

Byte	Description

.]

3.6.1.17 SAV_Digital Status

[

.

Byte	Description

.]

3.6.1.18 SAV_DF (Opcode 97_H)

[

Byte	Description

]

3.6.1.19 AND_Digital Status

[

Byte	Description

]

3.6.1.20 OR_Digital Status

[

Byte	Description

.]

3.6.1.21 SGET_Digital Status

[

Byte	Description

]

3.6.1.22 SAVS_Digital Status

[

Byte	Description

.]

3.6.1.23 SAVR_Digital Status

[

Byte	Description

]

3.6.1.24 SAVA_Point and MSG_Point

[

Byte	Description

]

3.6.1.25 GET_REMxxx

[

Byte	Description

]

3.6.1.26 IF_SKP (Opcode $7F_H$)

[

Byte	Description

]

3.6.1.27 CO Point Instructions

[

]

3.6.1.27.1 INC_CTR (Opcode $7D_H$)

[

Byte	Description

]

3.6.1.27.2 DEC_CTR (Opcode $7E_H$)

[

Byte	Description

.]

3.6.1.27.3 GET_CO (Opcode 85_H)

[

Byte	Description

.]

3.6.1.27.4 SAV_CO (Opcode 95_H)

[

Byte	Description

.]

3.6.1.27.5 SGET_CO (Opcode $C5_H$)

[

Byte	Description

]

3.6.1.28 TI Point Instructions

[

]

3.6.1.28.1 GET_TIS (Opcode 83_H)

[

Byte	Description

]

3.6.1.28.2 GET_TIV (Opcode 84_H)

[

Byte	Description

.]

3.6.1.28.3 SAV_TIS (Opcode 93_H)

[

Byte	Description

]

3.6.1.28.4 SAV_TIPR (Opcode 94_H)

[

Byte	Description

]

3.6.1.28.5 SAV_HT (Opcode $9B_H$)

[

Byte	Description

]

3.6.1.28.6 AND_TI (Opcode $A3_H$)

[

]

3.6.1.28.7 OR_TI (Opcode AC_H)

[

.]

3.6.1.28.8 SGET_TIS (Opcode $C3_H$)

[

]

3.6.1.28.9 SGET_TIV (Opcode $C4_H$)

[

]

3.6.1.28.10 SAVS_TI (Opcode $D3_H$)

[

]

3.6.1.28.11 SAVR_TI (Opcode DB_H)

[

Byte	Description

.]

3.6.1.28.12 SAVS_HT (Opcode $E8_H$)

[

]

3.6.1.28.13 SAVR_HT (Opcode $E9_H$)

[

]

3.6.1.29 Instructions for Floating-Point BL Values

[

]

3.6.1.29.1 GET_BLV (Opcode 89_H)

[

.]

3.6.1.29.2 SAV_BLV and SAV_BLV_

[

]

3.6.1.29.3 SGET_BLV (Opcode $C9_H$)

[

]

3.6.1.30 Miscellaneous Instructions

The following paragraphs describe processing for instructions that provide input data directly for instructions that transfer data to or from temporary registers. Temporary registers are used for storage of intermediate values while processing instructions internal to an equation statement. Normally, all data transferred to a temporary register while executing the instructions that make up an equation will be fetched back and processed before the last instruction of that equation is processed.

3.6.1.30.1 GET_TEMPS (Opcode $8B_H$)

[

]

Byte	Description

[

]

3.6.1.30.2 GET_TEMPV (Opcode $8C_H$)

[

.]

3.6.1.30.3 GET_VALB (Opcode $8C_H$)

[

]

3.6.1.30.4 GET_VALV (Opcode $8E_H$)

[

.]

3.6.1.30.5 GET_VALF (Opcode $8F_H$)

[

.]

3.6.1.30.6 SAV_TEMPS or SAV_TEMPV (Opcode $9E_H$ or $9F_H$)

[

]

3.6.1.30.7 AND_TEMP (Opcode $A8_H$)

[

.]

3.6.1.30.8 OR_TEMP (Opcode $B1_H$)

[

]

3.6.1.30.9 SGET_xx (Opcode CB_H, CC_H, CD_H, and CE_H)

[

]

3.6.1.30.10 COPY_DFQW Instruction (Opcode $7B_H$)

[

Byte	Description

.]

3.6.1.30.11 COPY_BLRC Instruction (Opcode $7C_H$)

[

]

3.6.1.31 Parameter Instructions

[

]

3.6.2 Math Instruction Jump Table

]

Table 5. Math Instruction Address Table

Instruction	Opcode (hex)	Description

Table 5. Math Instruction Address Table

Instruction	Opcode (hex)	Description

Table 5. Math Instruction Address Table

Instruction	Opcode (hex)	Description

.]

3.6.2.1 Standard Arithmetic Operations

3.6.2.1.1 Integer Operations for CO Points (Opcodes 00_H, $0C_H$, 18_H, and 24_H)

[

]

3.6.2.1.2 FP Operations for BL Points (Opcodes 02_H, $0E_H$, $1A_H$, and 26_H)

[

]

3.6.2.1.3 FP Operations for TMP Data (Opcodes 04_H, 10_H, $1C_H$, and 28_H)

[

]

3.6.2.1.4 Integer Operations for VA Data (Opcodes 06_H, 12_H, $1E_H$, and $2A_H$)

[

]

3.6.2.1.5 FP Operations for FP Data (Opcodes 08_H, 14_H, 20_H, and $2C_H$)

[

.]

3.6.2.2 Comparison Operations

3.6.2.2.1 COMP_CO Operation (Opcodes 30_H thru 38_H)

[

.]

3.6.2.2.2 COMP_BL Operation (Opcodes 40_H thru 48_H)

[

]

3.6.2.2.3 COMP_TMP Operation (Opcodes 50_H thru 58_H)

[

.]

3.6.2.2.4 COMP_VA Operation (Opcodes 60_H thru 68_H)

[

]

3.6.2.2.5 COMP_FP Operation (Opcodes 70_H thru 78_H)

[

.]

3.6.2.2.6 CMP_TSE

[

]

3.6.2.3 Combination Operations

[

.]

3.6.2.3.1 ADD/SUB/MUL/DIV_SAV_CO (Opcodes 90_H, 96_H, $9C_H$, and $A2_H$)

[

.]

3.6.2.3.2 ADD/SUB/MUL/DIV_SAV_BL (Opcodes 92_H, 98_H, $9E_H$, and $A4_H$)

[

.]

3.6.2.3.3 SETZ/SET1 Operations (Opcodes $B0_H$, $B2_H$, $B6_H$, and $B8_H$)

[

.]

3.6.2.3.4 INC/DEC/SAV CO Operations (Opcodes BC_H, $C2_H$, and $C8_H$)

[

.]

3.6.2.3.5 INC/DEC/SAV BL Operations (Opcodes BE_H, $C4_H$, and CA_H)

[

]

3.6.2.4 SAV_TIPRV Operation (Opcode CE_H)

[

]

3.6.2.5 Parameter Math Instructions

[

]

3.7 REMOTE DATA JUMP TABLE

[

]

Table 6. Get Remote Data Address Table

Instruction	Opcode (hex)	Description

Table 7. Remote Math Instruction Address Table

Instruction	Opcode (hex)	Description

3.7.1 Remote Instruction Operations

[

.]

3.7.1.1 GET_REMQW (Opcodes 80_H, 81_H, 82_H, 86_H, 87_H, and 88_H)

[

Byte	Description

]

3.7.1.2 GET_REMTIS, GET_REMINT, GET_REMFLP

[

Byte	Description

]

3.7.1.3 AND_REMQW (Opcodes $A0_H$, $A1_H$, $A2_H$, $A4_H$, $A5_H$, and $A6_H$)

[

Byte	Description

.]

3.7.1.4 AND_REMTIS (Opcode $A3_H$)

[

Byte	Description

.]

3.7.1.5 OR_REMQW (Opcodes $A9_H$, AA_H, AB_H, AD_H, AE_H, and AF_H)

[

Byte	Description

]

3.7.1.6 OR_REMTIS (Opcode AC_H)

[

Byte	Description

.]

3.7.1.7 REM_MATHINS (Opcode $B8_H$)

[

.]

Byte	Description

3.7.1.8 SGET_x (Opcodes $C0_H$, $C1_H$, $C2_H$, $C3_H$, $C4_H$, $C5_H$, $C6_H$, $C7_H$, and $C8_H$)

[

Byte	Description

.]

3.7.2 REMMATHTBL Operations

[

Byte	Description

.]

3.8 CANNED ROUTINES ADDRESS TABLE

[

]

Table 8. Canned Routines Address Table

Routine	Description

Table 9. CQ4 Algorithm Address Table

CQ4 Block	Type Index	Description

Table 9. CQ4 Algorithm Address Table (Cont)

CQ4 Block	Type Index	Description

Table 9. CQ4 Algorithm Address Table (Cont)

CQ4 Block	Type Index	Description

Table 10. CQ4 Integer Algorithm Address Table

CQ4 Block	Type Index	Description

3.9 PARAMETER INSTRUCTION PROCESSING

4.0 EQUATION TASKS TABLE

[

.]

4.1 EQUATION INTERPRETER STATUS TABLE

[

.]

Table 11. Equation Interpreter Status Table

Name	Size (byte)	Description

HF Controls

HF Controls Corporation

Control Systems

C-Link Protocol
Software Component Design Specification

DS002-000-01

Rev C

Effective Date	5/15/07
Author/Title	J. Taylor
Reviewer/Title	B. Cain
Approval/Title	T. Gerardis

[ ]

Copyright© 2007 HF Controls Corporation

Revision History

Date	Revision	Author	Changes
11/21/03	0	B Lin	Initial draft
12/18/03	1	B Lin	New draft
12/23/03	2	B Lin	Review Comments Incorporated
04/27/04	A	B Han	Update
02/08/05	B	A Hsu	Modify it for generic use
3/9/07	C	J Taylor	SCR 1766

Table of Contents

1. INTRODUCTION **5**

2. FUNCTIONAL DESCRIPTION **5**

2.1 HARDWARE INTERFACE 6
2.2 TOKEN PASSING 6
2.3 SYNCHRONIZATION ON DUAL-CHANNELS 7
2.3.1 Structure of the Sequence Number Tables *8*
2.3.2 Rules for Synchronization *8*
2.4 INTERFACE WITH OTHER PROCESSORS OR CONTROLLERS 9
2.5 MESSAGE TYPES 9

3. MASTER FOR A MOMENT TOKEN PASSING PROTOCOL **10**

3.1 CPC STATES AND THEIR ASSOCIATED TIMERS 10
3.1.1 T0 - Offline State *10*
3.1.2 T4 - Initial state *10*
3.1.3 T1 - Token Holding state *12*
3.1.4 T2 - Slave Talking State *12*
3.1.5 T3 - Waiting ACK Token state *13*
3.2 RECOVERY SOLUTIONS TO FAULT SCENARIOS 13
3.2.1 One node fails to pass the Token *13*
3.2.2 One node passed Token but with communication error *14*
3.2.3 One node fails to claim Token *14*
3.2.4 One node claimed Token but with communication error *14*
3.2.5 Token Claim Collision (multiple-tokens) *14*
3.2.6 Deaf Nodes *14*

4. DDB BROADCAST **14**

4.1 DDB BROADCAST TABLE AND DDB FILTER TABLE 14
4.2 DDB STRUCTURE 15

5. CPC PROCESSOR SOFTWARE **16**

5.1 LAYER STRUCTURE 16
5.1.1 Public Memory *17*
5.1.2 Interface Layer *17*
5.1.2 LLC Layer *17*
5.1.3 MAC Layer -NIC driver *18*
5.2 PROTOCOL DATA UNITS (PDU) OF EACH LAYER 18
5.3 SOFTWARE MODULES 20
5.3.1 Communication Initialization *23*
5.3.2 Transmit *24*
5.3.2.1 Interface Transmit Task 24
5.3.2.2 LLC Transmit Task 24

5.3.2.3 NIC Transmit Driver 25
5.3.3 Receive 25
5.3.3.1 Receive Interrupt Handler of NIC layer 25
5.3.3.2 LLC Receive Task 26
5.3.3.3 Interface Receive Task 26
5.3.4 Timer ISR 27
5.3.4.1 Timer 0 ISR 27
5.3.4.2 Timer 1 ISR 27
5.3.4.3 Timer 2 ISR 27
5.3.5 DDB Process Routine 27
5.3.5.1 Save DDB 27
5.3.5.2 Put DDB 27
5.3.5.3 Get DDB 28
5.4 NUCLEAR SAFETY CONTROL APPLICATION 28
5.4.1 Nuclear Safety Communication Link (C-Link) Software 28
5.4.1.1 Token Passing Scheme 29
5.4.1.2 Synchronization on Dual-Channels 29
5.4.1.3 Message Types 30
5.4.1.4 C-Link Processor Software Architecture 30
5.5 SOURCE LISTING 30

1. Introduction

An HFC control system is a distributed control system that provides plant monitoring and control capabilities spread over multiple remote control units. The C-Link LAN enables a system controller to broadcast data and exchange information among remotes or communicate with the Operator Station PC. [

]

Figure 1-1. Architecture of an HFC Control System

2. Functional Description

[

]

2.1 Hardware Interface

[

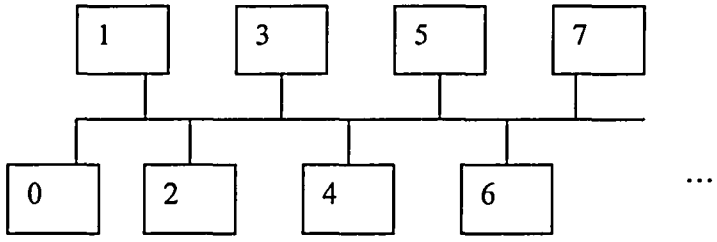


Figure 2-1 C-Link Structure

.]

2.2 Token Passing

[

]

2.3 Synchronization on Dual-Channels

[

]

2.3.1 Structure of the Sequence Number Tables

[

]

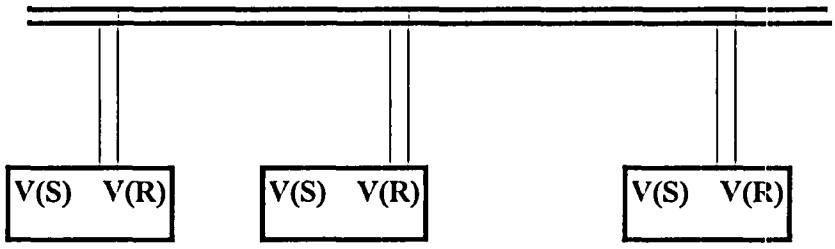


Figure 2-2. Sequence Number Tables

].

2.3.2 Rules for Synchronization

[

.]

2.4 Interface with Other Processors or Controllers

[

]

2.5 Message Types

[

]

3. Master For a Moment Token Passing Protocol

[

.]

3.1 CPC States and their associated Timers

3.1.1 T0 - Offline State

[

]

3.1.2 T4 - Initial state

[

]

Figure 3-1 State Diagram for Token Passing Mechanism

3.1.3 T1 - Token Holding State

[

.]

3.1.4 T2 - Slave Talking State

[

]

3.1.5 T3 - Waiting ACK Token State

[

:]

3.2 Recovery Solutions to Fault Scenarios

3.2.1 One node fails to pass the token

[

]

3.2.2 One node passed token but with communication error

[

]

3.2.3 One node fails to claim token

[

]

3.2.4 One node claimed token but with communication error

[

]

3.2.5 Token claim collision (multiple-tokens)

[ ]

3.2.6 Deaf nodes

[

]

4. DDB Broadcast

[

.]

4.1 DDB Broadcast Table and DDB Filter Table

[

Figure 4-1. Example of DDB Filter

]

4.2 DDB Structure

Table 4-1. DDB Header

Bytes	Name	Description

Table 4-2. Point Types for DDB

Index	Type	Size

5. CPC Processor Software

[

]

5.1 Layer Structure

[

.]

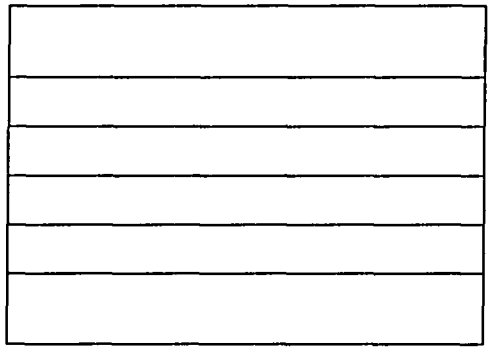


Figure 5-1 Structure of Layers

5.1.1 Public Memory

[

]

5.1.2 Interface Layer

[

]

5.1.2 LLC Layer

[

.]

5.1.3 MAC Layer -NIC driver

[

]

5.2 Protocol Data Units (PDU) of Each Layer

[

Table 5-1. PDU Header for Interface Layer

Bytes	Name	Description

Table 5-2. PDU Header for LLC Layer

Bytes	Name	Description

Table 5-3. Control Byte in the PDU Header for LLC Layer

Bit	Name	Description

Table 5-4. PDU Header for MAC Layer

Bytes	Name	Description

Table 5-5. PDU Header for NIC Receive Layer

Bytes	Name	Description

Total 4 bytes.

Table 5-6. Receive Status Byte

Bit	Description

.]

5.3 Software Modules

The CPC processor software is designed based on an operating system component. [

]

Figure 5-2. CPC Modules and Data Storage

Figure 5-3. Overview of CPC Communication Module Interactions

5.3.1 Communication Initialization

[

.]

5.3.2 Transmit

5.3.2.1 Interface Transmit Task

[

]

5.3.2.2 LLC Transmit Task

[

]

5.3.2.3 NIC Transmit Driver

[

]

5.3.3 Receive

5.3.3.1 Receive Interrupt Handler of NIC layer

[

]

5.3.3.2 LLC Receive Task

[

]

5.3.3.3 Interface Receive Task

[

]

5.3.4 Timer ISR

[

]

5.3.4.1 Timer 0 ISR

[ ]

Type	Value	Description

5.3.4.2 Timer 1 ISR

[

]

5.3.4.3 Timer 2 ISR

[

]

5.3.5 DDB Process Routine

5.3.5.1 Save DDB

[

.]

5.3.5.2 Put DDB

[

]

5.3.5.3 Get DDB

[

.]

5.4 Nuclear safety control application

[

]

Figure 5-4 - Communication Paths of HFC Nuclear Safety Controller

5.4.1 Nuclear Safety Communication Link (C-Link) Software

[

.]

Table 7 - The comparison between normal C-link operation and nuclear safety application

Functions	Normal C-link	Nuclear safety application

5.4.1.1 Token Passing Scheme

[

]

5.4.1.2 Synchronization on Dual-Channels

[

]

5.4.1.3 Message Types

[

]

5.4.1.4 C-Link Processor Software Architecture

[

.]

5.5 Source Listing

[

]

HF Controls Corporation

Universal Communications Protocol
Component Design Specification

DS002-000-03

Rev B

Effective Date 02/16/06

Author/Title Allen Hsu

Reviewer/Title Angela Han

Approval/Title Allen Hsu

[ ]

Copyright© 2005 HF Controls Corporation

Revision History

Date	Revision	Author	Changes
12/30/03	0	B Lin	Initial draft
12/31/03	1	B Lin	Review Comments Incorporated
05/06/04	A	A Han	Update
2/8/2005	B	A Hsu	Modify as generic S/W component document

Table of Contents

1. INTRODUCTION **4**

2. FUNCTIONAL DESCRIPTION **4**

2.1 MESSAGE STRUCTURES 5
2.2 TWO ADDRESSING METHODS 6

3. DETAILED UCP PROTOCOL DESCRIPTIONS **7**

3.1 PATH ELEMENTS 7
3.2 REMOTE STATUS TABLES FOR THE BRIDGE PATH METHOD 10

4. UCP NETWORK LAYER STRUCTURE **12**

4.1 TB STRUCTURE 13
4.1.1 TB_STAT Status Byte *14*
4.1.2 TB_ERRC Error Code Byte *15*
4.1.3 TB_PCT Packet Size Word *15*
4.1.4 TB_LNK and TB_STA (Source Path Element) *15*
4.1.5 TB_CNTL Control byte *15*
4.2 UCP/APPLICATIONS INTERFACE 16
4.2.1 Network Interface *16*
4.2.2 UCP_SEND Interface *18*
4.3 APPLICATION TASK LAYER 19
4.4 APPLICATION/PATH INTERFACE 19
4.5 COMMUNICATION PATH LAYER 20
4.6 PATH/HARDWARE INTERFACE 20
4.7 HARDWARE LAYER 20
4.8 RESPONSE MESSAGES 21

5. SOURCE LISTING **21**

1. Introduction

Universal Communications Protocol (UCP) is an HFC proprietary protocol that allows data to be transferred among nodes/processors/tasks/devices in a multi-layer distributed HFC control system. The data path can be upward, downward, or within the same layer of communication links. UCP protocol provides a defined standardized software interface for multiple layers within the communication system. With UCP protocol, each intelligent agent within a control system can communicate with other intelligent agents without concern for what types of media, what kind of configuration or arrangement, or what kind of protocol the message is to traverse. UCP protocol also guarantees the reliable return of a response message.

The communication software at each layer interface will take care of hardware arrangement and specific network topology while abiding with the UCP protocol. There can be different implementations for the UCP protocol. With UCP protocol as the standard software interface, the development of communication functions at one layer of the network does not require detailed knowledge of other layers, while a message can be recognized at all points in the system and the response message can find its correct path to return to the originator of a request reliably.

The purpose of this document is to provide both an overview of the functional usage of this communications protocol and specific message structures currently defined by the protocol.

2. Functional Description

Figure 1 presents the architecture of a typical HFC control system. An HFC control system is a distributed control system that provides plant monitoring and control capabilities spread over multiple remote control units. An operator can send commands or inquiries to a specific node in the control system through an HMI (Human Machine Interface) such as an operator station PC. The mechanism behind this communication is based on the UCP protocol.

[

]

Figure 1. A Typical HFC Control System Architecture

2.1 Message Structures

The general structure of all UCP messages consists of a UCP map, followed by the message, and closing with the trailer.

Figure 2. UCP Message Structure

[

]

2.2 Two Addressing Methods

[

]

Figure 3. Structure of Addressing

3. Detailed UCP Protocol Descriptions

3.1 Path Elements

[

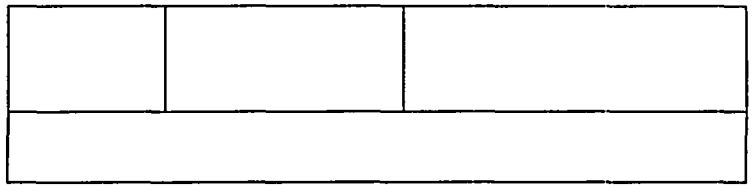


Table 1. Defined Processor IDs

Name	Code	Descriptions

Table 2. Defined Application Channel IDs

Name	Code	Descriptions

Table 3. Defined Standard UCP Request Codes

UCP Standard Request		

Table 4. Defined MTAGs (Application Task Numbers)

Name	Code	Descriptions

3.2 Remote Status Tables for the Bridge Path Method

[

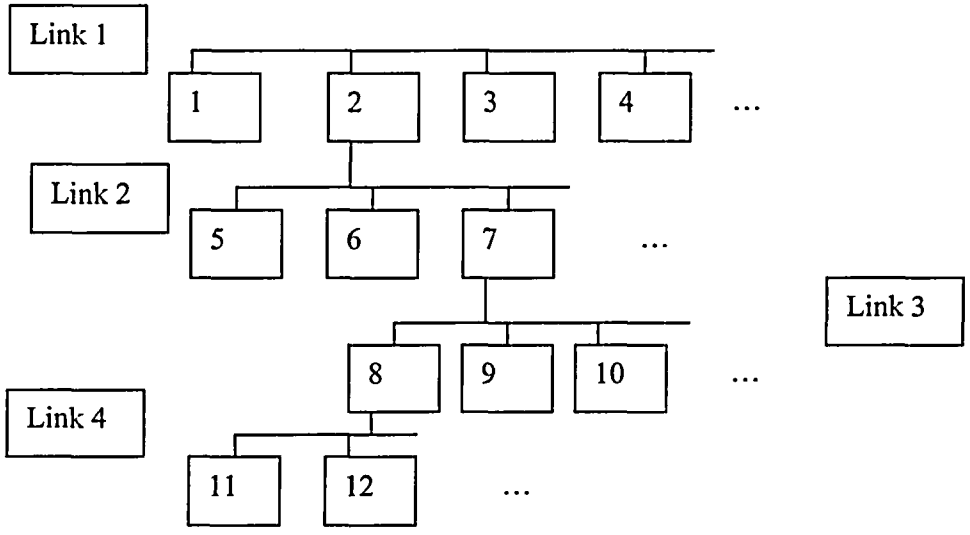


Figure 5. Network Topology

Table 5. Remote Status Array-REMOTE_DEF_STRUC

Name	Description

Table 6. Remote Status Array Entry – REMOTE_STATUS_STRUC

Name	Size (byte)	Description

4. UCP Network Layer Structure

[

Figure 6. Network Layer Structure

]

4.1 TB Structure

[

Status	TB Header	UCP Packet

Figure 7. TB Structure

]

Table 7. Universal TB Elements

4.1.1 TB_STAT Status Byte

[

Table 8. UCP Message Status

Name	Value	Description

]

4.1.2 TB_ERRC Error Code Byte

[

.]

Table 9. UCP Message Error Code

Name	Description

4.1.3 TB_PCT Packet Size Word

[

]

4.1.4 TB_LNK and TB_STA (Source Path Element)

[

.]

4.1.5 TB_CNTL Control byte

[

]

4.2 UCP/Applications Interface

[

]

4.2.1 Network Interface

[

]

Table 10. Status Structure

Status	Size (byte)	Description

Table 11. Network Buffer (NRB) Structure

4.2.1.1 NTW_STAT Status Byte

[

]

4.2.1.2 NTW_ERRC Error Code

[

.]

4.2.1.3 RES_COUNT Response Count

[

]

4.2.1.4 RES_PTR Response Pointer

[

]

4.2.1.5 REQCOD Request Code

[

]

4.2.1.6 PROCID Processor ID

[

.]

4.2.1.7 MTAG Applications Task Number

[

.]

4.2.1.8 REMOTN Encoded Controller or Remote Number

[

.]

4.2.1.9 LOC_BUF_PTR Local Buffer Pointer

[

.]

4.2.1.10 RMT_BUF_PTR Remote Buffer Pointer

[

]

4.2.1.11 BYNCT Byte Count

[

]

4.2.2 UCP_SEND Interface

[

.]

4.3 Application Task Layer

[

]

4.4 Application/Path Interface

[

.]

4.5 Communication Path Layer

[

.]

4.6 Path/Hardware Interface

[

]

4.7 Hardware Layer

[

]

4.8 Response Messages

[

]

5. Source Listing

[

]

HF Controls Corporation

Design Specification

System Software Components

DS001-000-06

Rev A

INFORMATION ONLY

Effective Date	3/10/05
Author	Angela Han
Reviewer	Steve Yang
Approval	Allen Hsu

[ ]
Copyright© 2004 HF Controls Corporation

DOOSAN

HFC

Revision History

Date	Revision	Author	Changes
10/20/2003	0	B Han	Initial
12/30/2004	A	A Han	Release

Table of Contents

1 INTRODUCTION **5**

2 FUNCTIONAL DESCRIPTION **5**

2.1 FUNCTIONAL SOFTWARE GROUPS 5
2.2 GENERIC SOFTWARE 5
2.3 ENHANCED NETWORK INTERFACE 5
2.4 SOFTWARE ARCHITECTURE 6
2.5 GENERAL SOFTWARE METHODOLOGY 7

3 THE SYSTEM SOFTWARE COMPONENTS DESCRIPTION **8**

3.1 OPERATING SYSTEM SOFTWARE COMPONENT 8
3.1.1 Processor Initialization *8*
3.1.1.1 System Processor Initialization 8
3.1.1.2 Subordinate Processor Initialization 10
3.1.1.3 OPERATING MODES 11
3.1.1.3.1 Self Test Operating Mode 11
3.1.1.3.2 System Processor Self Test Mode 12
3.1.1.3.3 Subordinate Processor Self Test Mode 12
3.1.1.3.4 OSX88 Multi-tasking Operating Mode 12
3.1.1.3.5 Multitasking Operating Mode Initialization 12
3.1.1.3.6 Multitasking Operating Mode Applications Execution 15
3.1.2 Processor Synchronization *17*
3.1.2.1 Interface Functions 17
3.2 UNIVERSAL COMMUNICATIONS COMPONENT 18
3.3 SYSTEM DIAGNOSTIC SOFTWARE 18
3.3.1 Diagnostic Software *18*
3.3.1.1 Power up initialization diagnostic software 18
3.3.1.2 On-line diagnostic software 19
3.3.1.2.1 On-line diagnostic software for controller components 19
3.3.1.2.2 On-line diagnostic software for memory components 21
3.3.2 Memory locations of diagnostic area – *23*
3.3.2.1 SUBPROC segment 23
3.3.2.2 Base on BEG_PUBM_PTR then find public memory root directory 24
3.3.2.3 Base on PRSEGDP to find each processor's segment directory 24
3.3.2.4 Base on PRSEG_PP pointer to find processor related segment directory 25
3.3.2.5 Base on DIAGSP pointer to find diagnostic segment with zero offset 25
3.4 SOFTWARE FILES 27
3.4.1 Operating System files *27*
3.4.1.1 Major files 27
3.4.1.2 System Processor Modules 28
3.4.1.3 Structure Definition 29
3.4.1.3.1 Controller Configuration Structure 29

Index of Figures

Figure 1 - Configuration Structure Definition 30

Index of Tables

Table 1 - On-line diagnostic software – mailbox validation 19
Table 2 - On-line diagnostic software and impact hardware 20
Table 3 - On-line diagnostic functions 21

Introduction

The system software components of HF Controls control systems are based around the filed proven "Rev 6.1" software version, with the following differences. The software is intended to be organized into independent groups based on functionality. Software for all Intel x86 processor based cards utilizes the same "generic" software modules. [

]

1 Functional Description

1.1 FUNCTIONAL SOFTWARE GROUPS

The purpose of software groups is to be able to include or omit a particular function as desired without affecting any other software. This is accomplished by creating a group of software modules that define any and all of their required parameters and/or support functions, except for standard "operating system" parameters and functions. To accomplish this, each functional module must include the assignment and allocation of all parameters and memory that it requires. Each group must also provide any required initialization software that will be executed by the operating system during initialization of the card. Each software group is assigned a unique function code. This function code is used by the initialization software to define which groups are to be made available for that particular configuration. This concept allows software functions to be easily included in or deleted from other printed circuited cards capable of performing that function. It also allows many different functional groups to be included in the firmware for a particular card, and through proper configuration only have those required for a particular application be made available during initialization. This feature can better utilize available memory and processor performance.

1.2 GENERIC SOFTWARE

The goal of generic software is to use the exact same software components wherever possible. Engineers configure selected components and dedicate them to particular application. The approach for accomplishing this is to separate all hardware dependent functions from the remainder of the software for a particular functional group. The software components containing no hardware specific code can be used in any card requiring that function along with the hardware specific code for that card. This requires that a standard interface exist between the non-hardware specific modules and any hardware specific module. Also a standard interface must exist for the "generic operating system".

1.3 ENHANCED NETWORK INTERFACE

[

]

1.4 SOFTWARE ARCHITECTURE

[

]

1.5 GENERAL SOFTWARE METHODOLOGY

All system utility software was strictly follow standard "Structured Programming" techniques where ever possible. All parameter definitions, structure definitions, and macro definitions required for more than one file should be specified in a single location (file), typically in an "INCLUDE" file.

The underlying principle behind the methodology is to be able to use the exact same files in as many different applications during the software configuration process. In support of the principle, the software is organized in self-supporting modules that can be inserted or deleted as desired for various applications. This requires that modules define all of their required resources. All files within the module are divided into two categories. The first category of files contains only applications specific information and contains no hardware specific requirements. Any hardware specific information must be included in a dedicated file (or files). This provides the ability to include this module in different hardware by using all of the same files with the exception of the hardware specific file(s). New hardware specific files must be created, and included in the module for this new hardware.

This method follows classical layered software techniques, where each layer supports different aspects of the module. Applications layer software files should contain neither protocol layer nor hardware layer information. Protocol layer files can be used on any hardware where that protocol is to be implemented. Hardware layer software files contain all hardware specific information for that protocol.

The software is designed to support two configurable controller platforms. These two platforms include multiple bus master controller platforms (such as "Multibus") and Personal Computer (PC) based platforms. The multiple bus master configuration can also be used to support single board applications. [

]

2 The System Software Components Description

This section presents descriptions of all system software components for this version. It includes HFC Generic Operating System (OS) and its utility software. The detail design descriptions of OS are listed in DS001-000-01, Operating System Component Design Specification.

2.1 OPERATING SYSTEM SOFTWARE COMPONENT

2.1.1 Processor Initialization

[

]

2.1.1.1 System Processor Initialization

[

AREA	STRUCTURE NAME	FILE NAME

]

2.1.1.2 Subordinate Processor Initialization

[

.]

2.1.1.3 OPERATING MODES

[

]

2.1.1.3.1 Self Test Operating Mode

[

]

2.1.1.3.2 System Processor Self Test Mode

[

]

2.1.1.3.3 Subordinate Processor Self Test Mode

[

]

2.1.1.3.4 OSX88 Multi-tasking Operating Mode

[

]

2.1.1.3.5 Multitasking Operating Mode Initialization

[

.]

2.1.1.3.6 Multitasking Operating Mode Applications Execution

[

.]

2.1.2 Processor Synchronization

[

.]

2.1.2.1 Interface Functions

The operating system provides several interface functions that are available to all applications to use during multi-tasking operation. The detail design specifications of all interface functions are listed in DS001-000-01 OS Component Design Specification.

2.2 UNIVERSAL COMMUNICATIONS COMPONENT

The HFC Universal Communications Protocol (UCP) is a general purpose protocol intended to allow information to be sent from any location within a system and received by any other location within the system. The primary goal of the protocol is to allow any available communications media and arrangement to be transparent to the original source and final destination of a message. Another goal is to provide a standardized software interface to the communications system, so that any applications layer software requiring the sending and/or receiving of a message can achieve its goal independent of the arrangement of the system configuration. Also, the creation of any new or different applications, requiring communications can be accomplished by adhering to the documented standard communications software interface procedure. The UCP is divided into multiple software layers, with defined standard software interfaces between the various layers. This allows software within the protocol to be expanded to support newly available hardware configurations without violating the overall UCP requirements. These new software procedures can interface with the other layers of the overall UCP thru the defined standard interfaces. Another goal of the UCP is to provide general purpose communications capability in the most efficient manner possible. For example, every effort is made to eliminate the need of copying messages from one memory area to another.

The detail design specification is defined in DS002-000-03 UCP protocol Component Design Specification.

2.3 SYSTEM DIAGNOSTIC SOFTWARE

The hardware watchdog timer resides on the controller/Controller chassis is the main control mechanism to report the hardware "Sanity" status. However, in order to strobe this watchdog timer continuously, all hardware components need to be functioned properly and passed all checks that are performed by on-line diagnostic software. HFC uses two major groups of software to prove the operating status of hardware components; they are 1) hardware initialization diagnostic software and 2) on-line diagnostic software. The checked hardware components include microprocessors, memory chips and their related circuitries. A 16 bytes "signature" memory area located at 8FF0:0 had been used as reference for all diagnostic software. It contains the signature word, starting address of public memory pointer and remote ID number. The information will be validated during the execution of diagnostic and communication software continuously.

2.3.1 Diagnostic Software

2.3.1.1 Power up initialization diagnostic software

During the power up initialization process, the software checks the following hardware components and their functions.

- It validates the PROM (EPROM) and FLASH by read the signature.
- It validates the RAM (both private and public) and Dual Ported Memory by write and read back a particular pattern.
- Validates the status of all configured microprocessors by using the dedicated diagnostic memory area for each processor.
- Validates the firmware and application code between PROM (EPROM) and Flash
- Synchronize all processors and I/O process by control a system-wide "GO" flag

2.3.1.2 On-line diagnostic software

2.3.1.2.1 On-line diagnostic software for controller components

[

Table 1 - On-line diagnostic software – mailbox validation

Step	Controller	Function	Hardware Involved	Diagnostic Software	Note

Table 2 - On-line diagnostic software and impact hardware

Step	Controller	Function	Hardware Involved	Diagnostic Software	Note

]

2.3.1.2.2 On-line diagnostic software for memory components

[

Table 3 - On-line diagnostic functions

Step	Controller	Function	Hardware Involved	Diagnostic Software	Note

]

2.3.2 Memory locations of diagnostic area –

[

.]

2.3.2.1 SUBPROC segment

[

]

2.3.2.2 Base on BEG_PUBM_PTR then find public memory root directory

[

]

2.3.2.3 Base on PRSEGDP to find each processor's segment directory

[

]

2.3.2.4 Base on PRSEG_PP pointer to find processor related segment directory

[

]

2.3.2.5 Base on DIAGSP pointer to find diagnostic segment with zero offset (The offset of DIAGSP is 60H)

[

]

2.4 SOFTWARE FILES

All codes for the operating system and its utility software, where ever possible, were configured to be in the same "Code Segment". All local RAM memory required by the operating system is configured to be in the same "Data Segment", including all processor stack areas. This allows any task or function to access any operating system memory variable using the "SS" register.

2.4.1 Operating System files

[

]

2.4.1.1 Major files

[

]

2.4.1.2 System Processor Modules

[

]

2.4.1.3 Structure Definition

2.4.1.3.1 Controller Configuration Structure

[

)

]

Figure 1 - Configuration Structure Definition

[

HF Controls

HFCONTROLS CONTROL SYSTEM

One-Step Software User's Guide
Software Revision 2.50
UG04-000-04

Rev C

Effective Date	9/4/07
Author/Title	G. Rochford
Reviewer/Title	J. Taylor
Approval/Title	T. Gerardis

[ ]

Revision History

Date	Revision	Author	Changes
6/1/02	A	H Bin	Initial Issue
12/9/04	Change 1	J Taylor	SCR 733 – Appendix A updated to include all symbols recognized by One Step; Appendix D modified to provide fault isolation guidance.
2/18/05	B	J Taylor	Reissued as standard user's guide for nuclear control systems.
4/10/07	C	G Rochford	Add OneStep.ini file description; Macros; Update with new symbols in Appendix A

PROPRIETARY NOTICE

The contents of this publication are proprietary data of HF Controls Corporation. Reproduction or use of any part of the publication for purposes other than the support of the equipment for which it is published is permissible only if expressly authorized in writing by HFC.

Additional copies of this publication may be purchased from HFC. When ordering or requesting cost information, refer to the publication number appearing on the title page. Address to the attention of the Sales Department, HF Controls Corporation, 1624 W. Crosby Road, Suite 124, Carrollton, TX 75006.

INTRODUCTION

This manual contains user instructions for the One-Step Software supplied with an HFC Control System and an overview of basic promis·e features.

This manual is part of a set covering the operation and maintenance of the HFC Control System. All personnel should become thoroughly familiar with the contents of those volumes relating to their specific responsibilities before attempting to use the control system. Because it is virtually impossible to cover every situation that might occur during operation and maintenance of the equipment described in this publication, personnel are expected to use good judgment when confronted with situations that are not specifically mentioned herein.

TABLE OF CONTENTS

SECTION	TITLE	PAGE
1	SYSTEM OVERVIEW	1-1
1.1	Introduction	1-1
1.2	System Requirements	1-1
1.3	Support Documentation	1-1
2	OVERVIEW OF ONE-STEP CODE GENERATION	2-1
2.1	Introduction	2-1
2.2	Make promis·e Drawings	2-2
2.2.1	promis·e Drawing Organization	2-2
2.2.2	Symbols	2-3
2.2.3	Device types	2-4
2.3	Generate promis·e Reports	2-4
2.4	Folder Structure for One-step Software	2-4
2.5	Outfiles from One-step Software	2-5
2.6	Contents Of OneStep.ini	2-6
3	promis·e DRAWING CREATION AND MODIFICATION	3-1
3.1	Introduction	3-1
3.2	Preliminary Installation	3-1
3.3	Overview of promis·e functions	3-3
3.3.1	promis·e menu bar	3-3
3.3.2	promis·e tool bars	3-3
3.3.2.1	ECS_promis·e Toolbar	3-3
3.3.2.2	View Toolbar	3-4
3.3.2.3	ECS_Modify Toolbar	3-4
3.3.2.4	ECS_Draw Toolbar	3-5
3.3.2.5	ECS_DATA Toolbar	3-5
3.3.2.6	DRAW/EDIT Toolbar	3-6
3.3.2.7	Properties Toolbar	3-6
3.3.3	Special keys and mouse functions	3-6
3.4	Devices	3-6
3.4.1	Device Type	3-7
3.4.2	Device ID	3-7
3.4.3	Connection Point	3-7
3.5	Detailed Instructions for Using promis·e	3-9
3.5.1	Start up	3-9
3.5.2	Add new loop schematic	3-9
3.5.2.1	Create a New Project	3-9
3.5.2.2	Create a New Page	3-12
3.5.2.3	Copy a Project	3-13
3.5.2.4	Copy an Installation	3-14
3.5.2.5	Copy a Page	3-15
3.5.3	Open a page	3-15
3.5.4	Edit a loop schematic	3-16

SECTION	TITLE	PAGE
3.5.4.1	Edit Symbols	3-17
3.5.4.1.1	Modify Symbols	3-17
3.5.4.1.1.1	Edit Device ID	3-17
3.5.4.1.1.2	Edit Connection Point	3-19
3.5.4.1.1.3	Edit Address (PLC Text)	3-20
3.5.4.1.1.4	Edit Symbol Text	3-21
3.5.4.1.2	Add Circuit Card Symbols	3-22
3.5.4.1.2.1	By Inserting with Macro	3-22
3.5.4.1.2.2	By Inserting Symbol Only	3-27
3.5.4.1.3	Add a Logic Gate	3-36
3.5.4.1.4	Add a Process Control Permissive	3-39
3.5.4.2	Edit Text	3-42
3.5.4.3	Edit Wire lines	3-42
3.5.4.4	Edit Page Variables	3-44
4	promis·e REPORTS EXTRACTION	4-1
4.1	Preliminary configuration	4-1
4.2	Generation of report files	4-1
5	ONE-STEP CODE GENERATION AND CONVERSION	5-1
5.1	Preliminary operation	5-1
5.2	Start up	5-1
5.3	Generating Macro Code	5-4
5.4	Generating Application Code	5-6
5.5	Converting Graphic Files	5-7
6	WRITING PROGRAM CODE TO EPROM	6-1
6.1	Introduction	6-1
6.2	Start up	6-1
6.3	Load Application Code	6-2
6.4	Program EPROM	6-5
6.5	Install EPROM	6-6
6.6	Test EPROM	6-8

LIST OF TABLES

LIST OF FIGURES

Figure 2-1. System View of Using One-Step Software 2-2
Figure 2-2. promis·e Drawing Organization 2-3
Figure 2-3. Folder Structure of Input and Output Files for One-Step Software 2-5

Section 1

SYSTEM OVERVIEW

1.1 INTRODUCTION

The HFC One-Step Software enables the user to generate the application code executable in the ECS-1200 controller and AFS-1000 loop controller from promis·e drawings. It also generates graphic files compatible with HFC MCRT software from the same promis·e drawings. Thus, the engineers can create and modify control logic application codes of a loop controller in a completely graphic programming environment. The binary application code generated can be loaded directly into an EPROM as the application firmware of a loop controller card. The graphic files can be displayed by MCRT software for monitoring the dynamic operation of control loop logic of the application firmware.

The One-Step software is a utility of HFC EWS/MCRT software suite. The promis·e software is a commercial software package that can be used with HFC-defined macros and symbols to generate promis·e drawings and report data files. The report files supply the source of data used by One-Step software to generate its outputs.

1.2 SYSTEM REQUIREMENTS

Minimum hardware and software requirements to run One-Step Software are as follows:

- Intel Pentium PC with a minimum of 128-Mb RAM
- Monitor with 1024*768 screen resolution
- Hard-disk drive with 4 GB or greater storage capacity (mirroring optional)
- 1.44 Mb floppy disk drive
- Mouse and keyboard
- Microsoft Windows 2000
- AutoCAD Release 2006
- HFC EWS/MCRT software suite
- Promis·e software version 5.06

1.3 SUPPORT DOCUMENTATION

- Hardware Manual
- EWS User's Guide
- MCRT User's Guide
- AutoCAD Documentation for Release 2006
- promis·e System Manual software version 5.x

Section 2

OVERVIEW OF ONE-STEP CODE GENERATION

2.1 INTRODUCTION

Application engineers use the promis·e software to define control loop algorithms in the form of promis·e drawings. The promis·e software runs as an Add-on to AutoCAD and allows users to set up various attributes for the drawings and their components. The information for each promis·e project is stored in an associated database, therefore it is extractable from promis·e.

[

]

As a whole control system job may involve a great number of drawings, files, application codes, and EPROMs, strict configuration management is indispensable. Each single item needs to have a specific identification number and revision number. After drawings are modified for a new revision, regenerate the corresponding report files and application codes, and load these new application codes to the EPROMs. The revision of the final application code in the EPROM must match the latest revision of promis·e drawings.

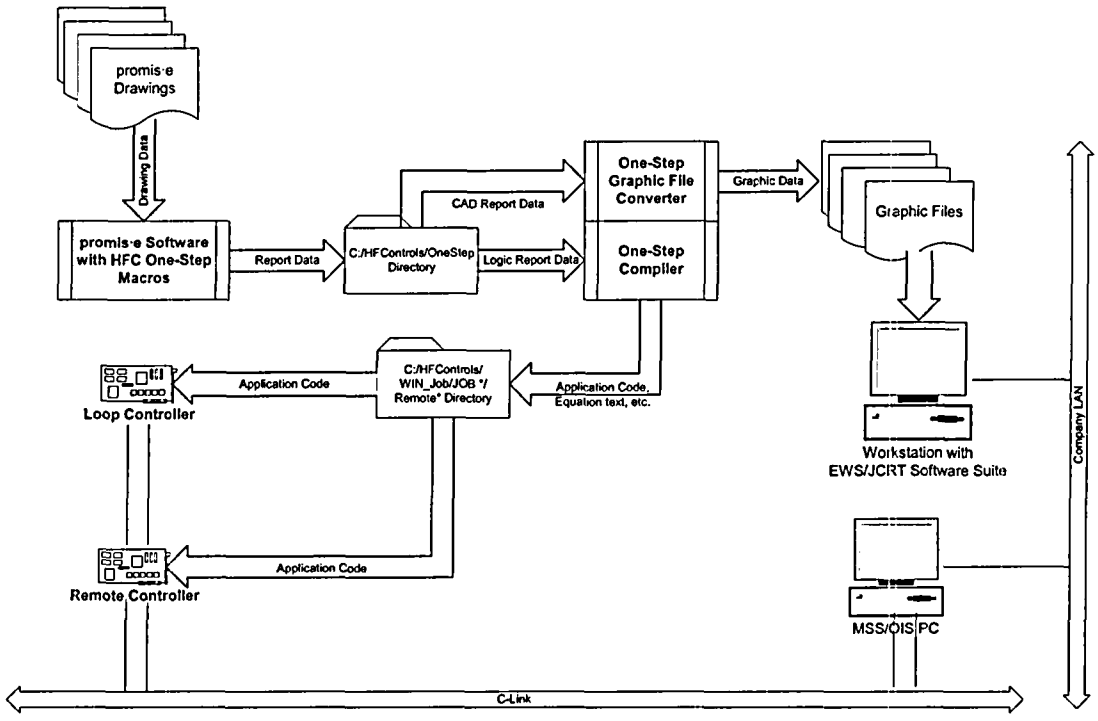


Figure 2-1. System View of Using One-Step Software

2.2 MAKE PROMIS·E DRAWINGS

2.2.1 promis·e DRAWING ORGANIZATION

The promis·e software organizes files into Projects, Installations, Locations and Pages as shown in Figure 2-2. A project includes multiple installations, and an installation includes multiple locations and multiple pages. Page numbering is unique within an installation.

The corresponding set up of PCS control algorithm structure in the promis·e is as follows:

- A control system, for example, SYS_455 is defined as a promis·e project.
- A control loop, for example 0251 is defined as a promis·e installation.
- Symbols in each page define specific logic functions to be performed by the controller running that logic.

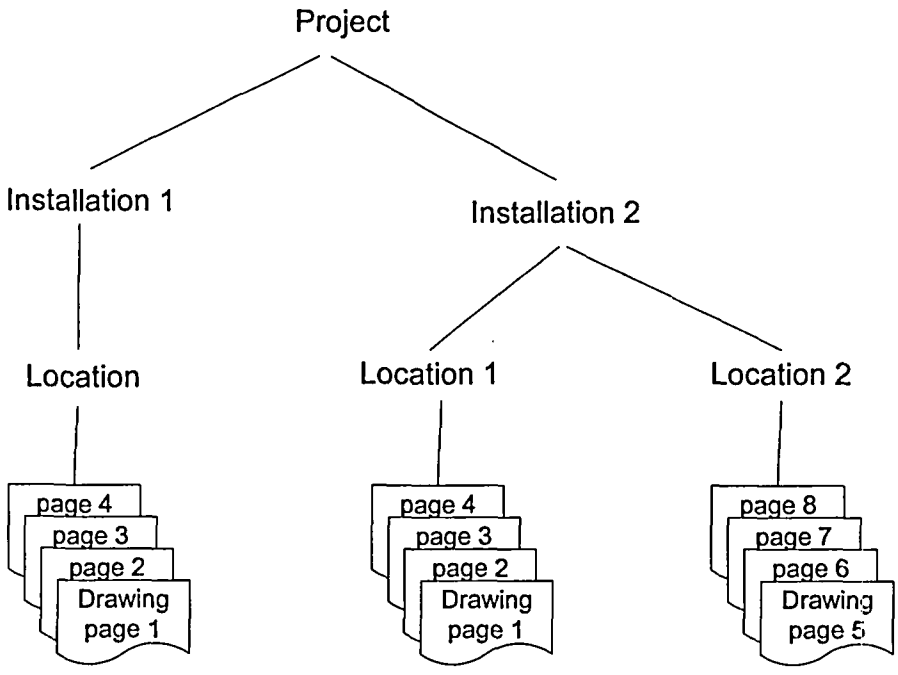


Figure 2-2. promis·e Drawing Organization

2.2.2 SYMBOLS

The One-Step Software can recognize only schematic symbols that have been defined for the Plant Control System (PCS). If undefined symbols are used to make promis·e drawings, application codes cannot be generated successfully. Appendix A provides a complete list of symbols that are recognized by the One-Step Software. The defined symbols are as follows:

- I/O Card Symbols represent all types of circuit cards used in the PCS system, such as AFS series I/O cards, ECS series I/O cards, AFS-SBC04N controller cards, etc.
- I/O Card Field Symbols for field inputs and outputs.
- Digital Logic Symbols, such as AND, OR, NOT, TIMER, etc.
- Analog Logic Symbols, such as PID block, DHA block, DLA block, etc.
- DDB Symbols and DDB Filter symbols
- Power Related Symbols.
- Miscellaneous hardware symbols for control switch modules and M/A stations, etc.

2.2.3 DEVICE TYPES

For the same reason as the symbols, only defined device types can be used to make promis·e drawings for the One-Step Software. Appendix C provides a complete list of device types that are recognized by the One-Step Software and the configuration of connection points for each device type.

2.3 GENERATE promis·e REPORTS

Running HFC One-Step Macros in promis·e will generate the following four logic reports at the same time for each system:

- Wire List: wire_list.txt. Wirelist data in this report define how symbols in a drawing are connected to one another.
- PLC (Programmable Logic Controller) List: plc_list.txt. This report defines how I/O card parent and child symbols are related to one another.
- Loop List: loop_list.txt. It identifies all loops included in a particular system.
- Data List: data_list.txt. It provides specific data for each individual symbol within the system.

Running HFC One-Step Macros for CAD report will generate CAD reports, ".txt" files.

The HFC One-Step Macros transfer the generated reports to appropriate subfolders automatically.

2.4 FOLDER STRUCTURE FOR ONE-STEP SOFTWARE

[

]

Figure 2-3. Folder Structure of Input and Output Files for One-Step Software

2.5 OUTFILES FROM ONE-STEP SOFTWARE

[

]

2.6 CONTENTS OF ONESTEP.INI

[

.]

Section 3

promis·e DRAWING CREATION AND MODIFICATION

3.1 INTRODUCTION

promis·e is commercial software that runs as an add-on to AutoCAD and uses the AutoCAD graphics engine to generate promis·e drawings. Designing in a graphic environment creates schematics that are easy to read and understand. By using a customized library of symbols, the user can also create or modify a drawing quickly and easily. Promis·e pages, lines and symbols have attributes associated with them, and these data attributes are extractable from the promis·e drawings. In order to provide inputs for the One-Step software, the PCS schematics have the required data attributes defined, such as logic gates, the timer values, the analog control blocks, the tuning data, the scaling data, other associated data, as well as the I/O points and loop numbers, etc.

This section is a brief overview of the features offered in promis·e. For a complete description, refer to the promis·e system manual and AutoCAD user's guide. Training in the use of AutoCAD and promis·e is recommended.

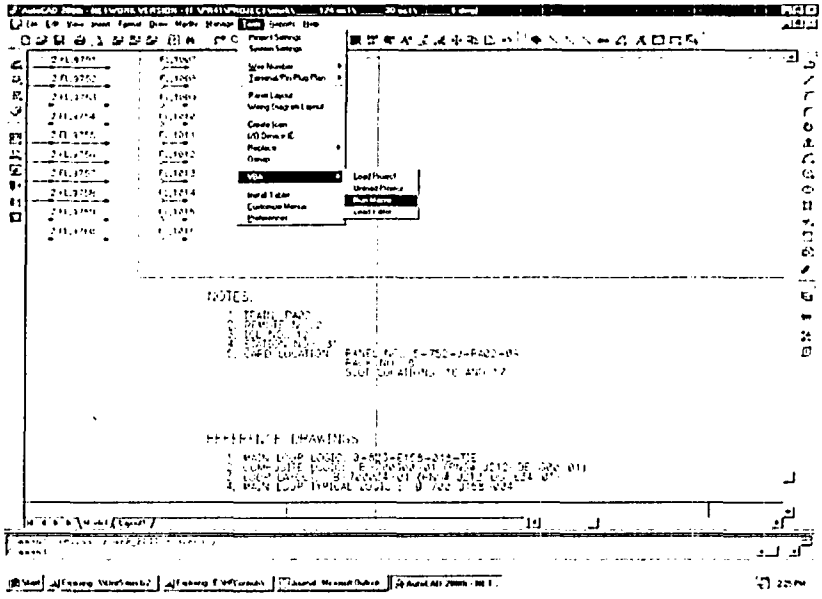
Custom symbols and device types have been defined and created to show all of the control loop logic diagrams required by the system. Only these symbols and device types should be used to make promis·e drawings for the One-Step software. Appendix A and C provide a list of the defined symbols and device types. These symbols and device types are available in the promis·e after the HFC library for the promis·e is installed.

3.2 PRELIMINARY INSTALLATION

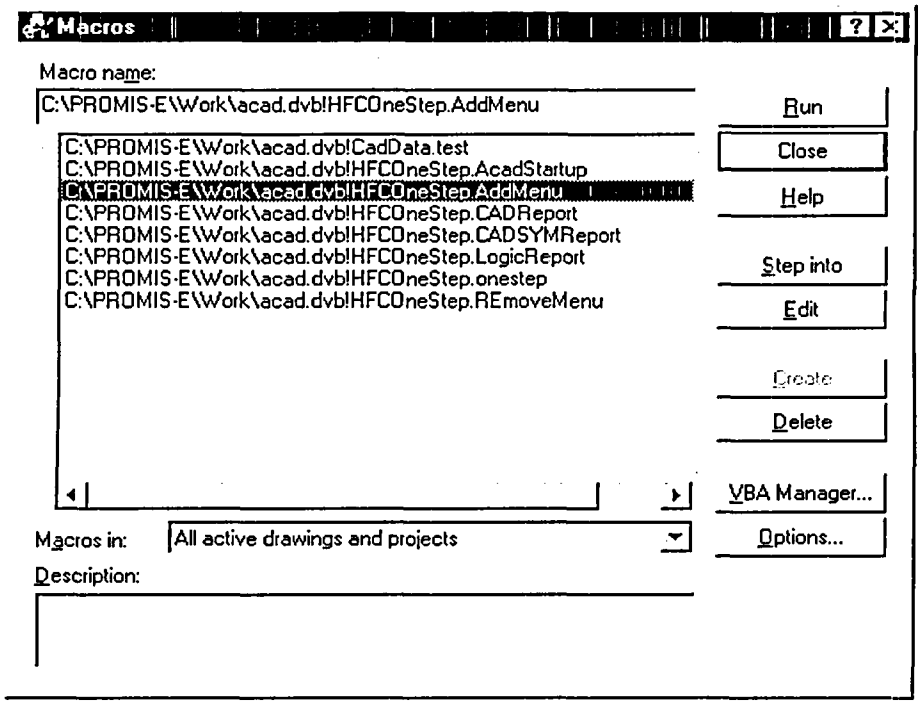
1. Install promis·e software as an Add-on to AutoCAD software. Refer to promis·e System Manual for information about installing and using promis·e.

2. Install HFC library for designing the PCS system under promis·e by copying the HFC folder to the Catalog folder under promis·e directory. After copying, make sure that the HFC folder includes "Block", "Device", "Macro", and "Wmf" subfolders.

3. Add **HFC Report Macro** Function to promis·e menu bar under **Tools**. First click **Tools > VBA > Run Macro.**



4. When the following dialog window pops up, select the **HFCOneStep.AddMenu** macro and click the **Run** button. The **HFC Report** Selection appears under **Tools** when it is running and ready to be used.



3.3

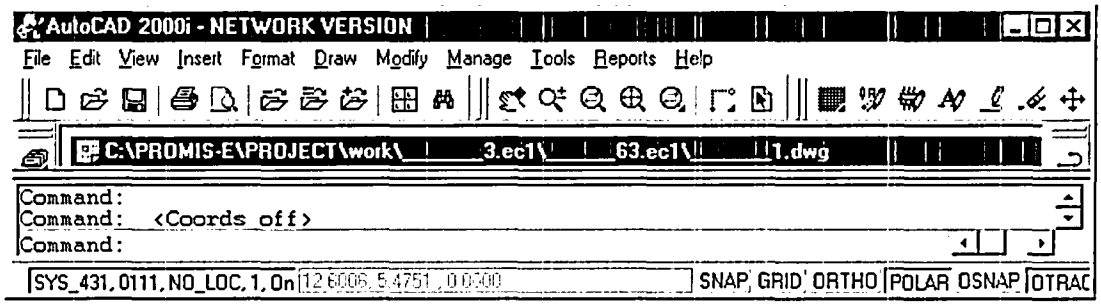
OVERVIEW OF promis·e FUNCTIONS

NOTE

promis·e has several ways to perform the same function, such as menu items, tool bar, shortcut keys, etc. For a more complete description, see promis·e system manual.

3.3.1 promis·e MENU BAR

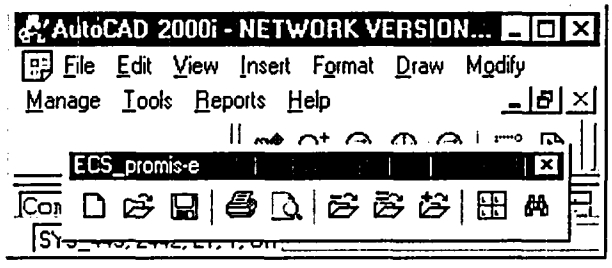
The promis·e software supports standard functions such as File, Edit, View, Help functions, etc, as well as functions related to drawings and reports. The menu bar provides access to most of the functions as shown:



3.3.2 promis·e TOOL BARS

The promis·e tool bars present most of the commands and functions needed to create and edit control loop diagrams. The tool bars appear around the border of the drawing. These icons are single click short cuts to items in the pull down menus.

3.3.2.1 ECS_promis·e Toolbar

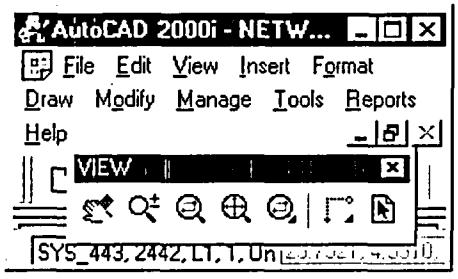


It includes the following icons:

- New page
- Page open
- Save
- Print
- Print preview

- Previous page
- Page refresh
- Next page
- Browse
- Navigator

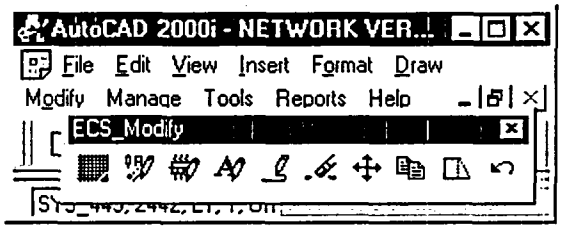
3.3.2.2 View Toolbar



It includes the following icons:

- Dynamic Pan
- Dynamic Zoom
- Zoom Window
- Zoom Extent
- Snap from point
- Redraw

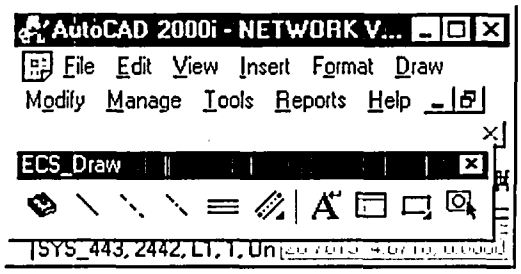
3.3.2.3 ECS Modify Toolbar



It includes the following icons:

- GridSnap
- Power-editor
- Change Symbol
- Edit Text
- Modify line
- Delete
- Move
- Copy
- Stretch
- Undo draw

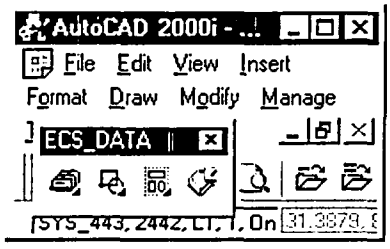
3.3.2.4 ECS Draw Toolbar



It includes the following tools for drawing:

- Select symbol
- Draw wire line
- Dashed wire line
- Center Wire line
- 3-Phase Wire
- Last Multi-line
- Text
- Log Field
- Log Box
- Graphical box empty

3.3.2.5 ECS DATA Toolbar



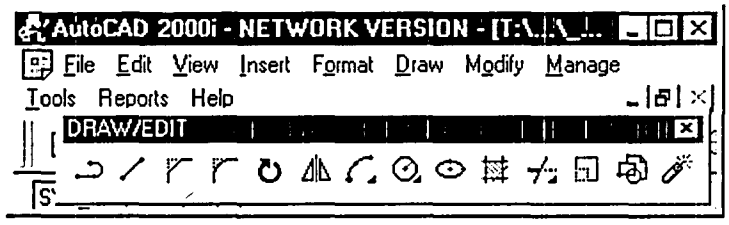
It includes the following tools:

- New Project
- Symbol Definition
- Macro Definition

3.3.2.6

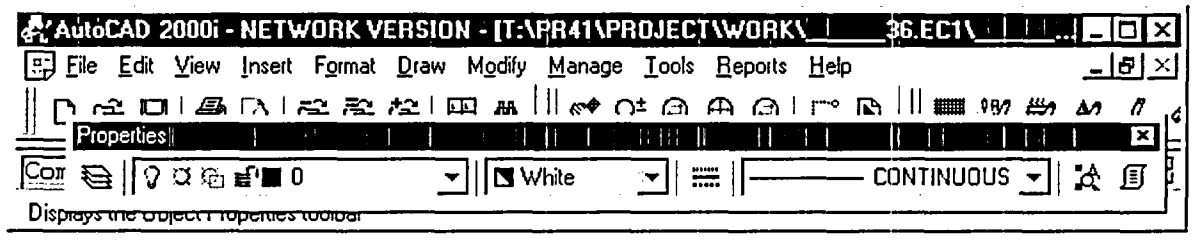
DRAW/EDIT Toolbar

It includes standard AutoCAD tools for drawings.



3.3.2.7 Properties Toolbar

It includes various icons for definitions of page properties.



3.3.3 SPECIAL KEYS AND MOUSE FUNCTIONS

The special keys and mouse functions are similar to AutoCAD.

- Left mouse button is the select button
- Right mouse button is the properties button and generally escape/repeat button
- Scroll wheel zooms in and out
- Scroll wheel button
  - Push and hold pans
  - Double click zooms to extent

NOTE

Pressing the space bar repeats last command.

3.4 DEVICES

Each circuit card used in the control system is considered as a device. A device symbol has attributes like Device ID, Device Type, and Connection Points. Configuring these attributes for each circuit card is very important for designing the control systems, and essential for the success in generating application codes with One-Step software.

3.4.1 DEVICE TYPE

A device type is a definition that specifies how many and what type of children symbols are associated with a parent symbol. For a circuit card, the parent symbol is the basic card, and the children symbols are the I/O points of the card. Therefore, generally the device name and the parent symbol name are the same for the circuit card symbols defined in the HFC library.[

]

3.4.2 DEVICE ID

The device ID identifies a circuit card in the PCS system. Each Device ID must be unique in a control system and should be predefined for each circuit card. The PCS system uses the following conventions to define the Device IDs for circuit card symbols:

- For AFS I/O cards, the device ID is like "bay##-rack##-slot##" that indicates the physical bay, rack, and slot where the card is installed in the system.

- For ECS I/O (MUX I/O) cards, the device ID is like "remote##-link##-station##" that indicates the remote, link, and station required to communicate with the card in the system.

- For a circuit card symbol, the parent symbol (the card) and the children symbols (the I/O points) share the same Device ID. Thus the I/O points are superimposed onto the circuit card.

3.4.3 CONNECTION POINT

Connection Point definition assigns a predefined point address in the control system to each of the field I/O points, i.e., the children symbols of a circuit card.

Connection Points

Select Connection Points

Device ID : =DATA+L1-03-4-19
Device : A103

Installation	Pg.Line	Symbol	Connection Point	Addresses
=0342	2.082	A103	1 2 3 4 5 6 7 8	
=0342	2.082	KINPUT	1	AI,41
=0342	2.088	KINPUT	2	AI,42
=0342	2.094	KINPUT	3	AI,43
=0342	2.100	KINPUT	4	AI,44
=0342	2.106	KINPUT	5	AI,45
=0342	2.112	KINPUT	6	AI,46
=0342	2.118	KINPUT	7	AI,47
=0342	2.124	KINPUT	8	AI,48

5

OK Cancel Help

- For AFS I/O points, the point address is like "Point Type, Point Number". Such as "DI, 31".

- For MUX I/O points, the point address is like "Remote ##, Point Type, Point Number". Such as "15, DI, 31".

3.5 DETAILED INSTRUCTIONS FOR USING promis·e

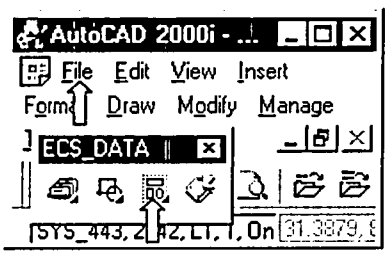
3.5.1 START UP

Invoke the promis·e program by double clicking the promis·e shortcut on the Windows desktop.

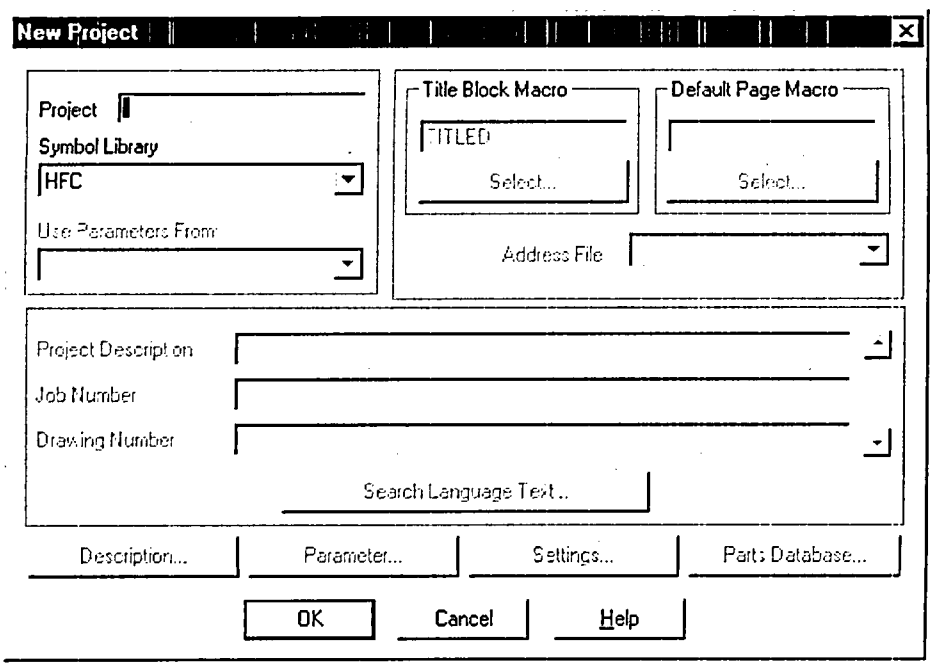
3.5.2 ADD NEW LOOP SCHEMATIC

3.5.2.1 Create a New Project

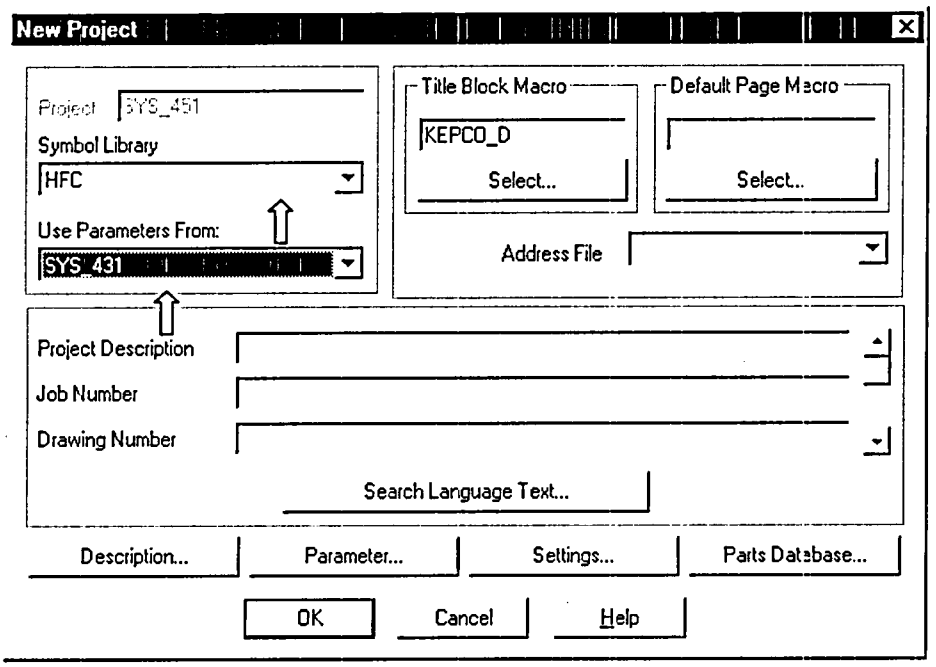
1. Click **Project** icon or click **File/New/Project** from the menu bar as shown.



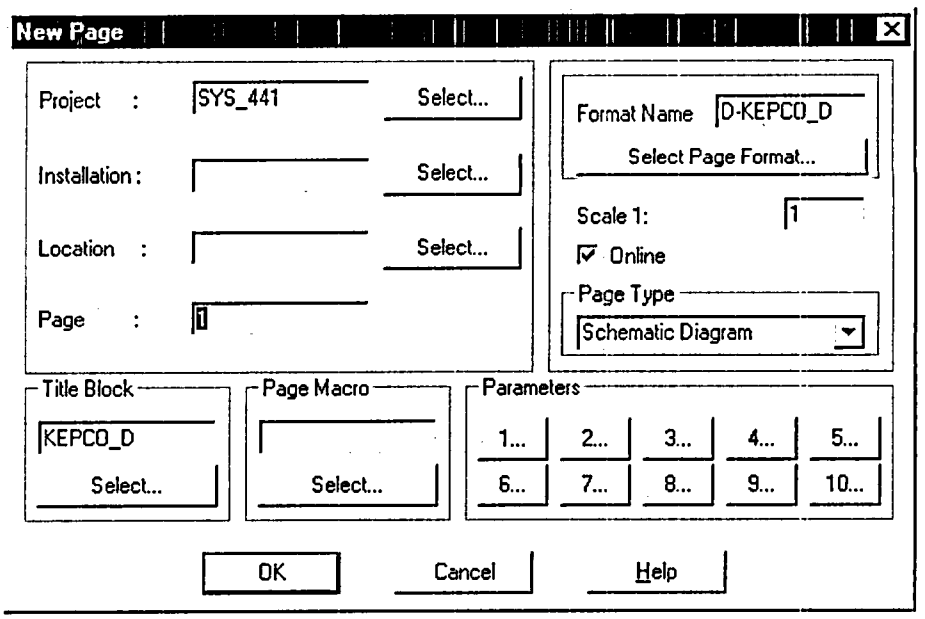
2. The following window will show up.



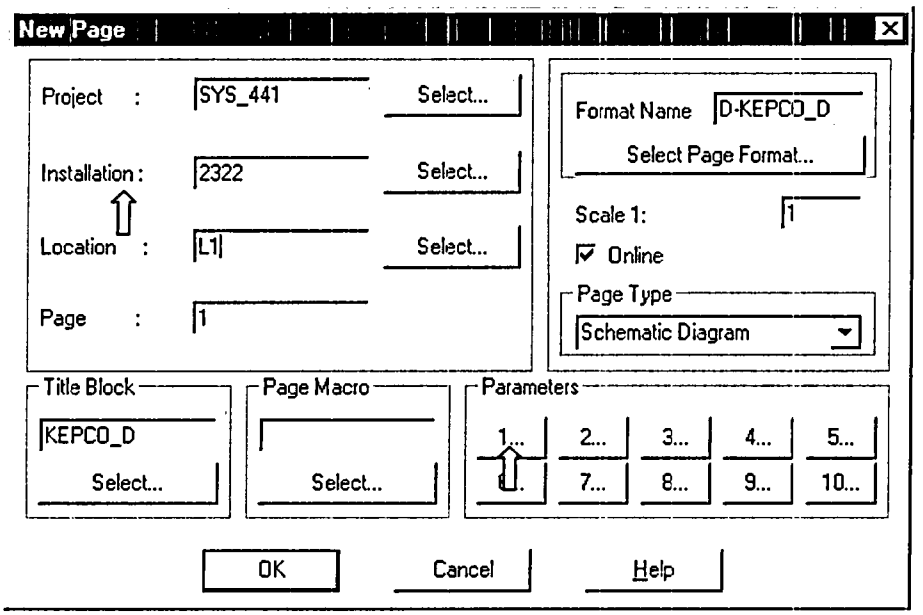
3. Enter the **Project** name first. For example SYS_441 here. **Symbol Library** should be HFC for the design of PCS control systems. Then select an existing system in the **Use Parameters From** window, such as SYS_431. Enter **Project Description, Job Number,** and **Drawing Number** as required. Click **OK.**



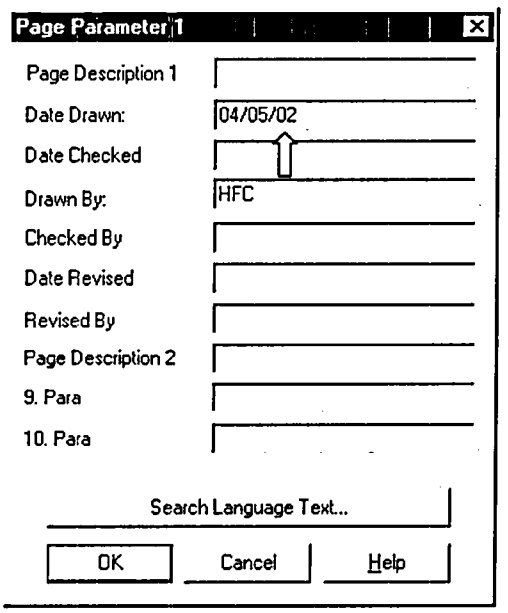
4. The following window will show up.



Enter loop number to the **Installation** window, and typically L1 in the **Location** window.



5. The **Parameters** window provides parameters to define the page. Click button **1** and the following window will show up.



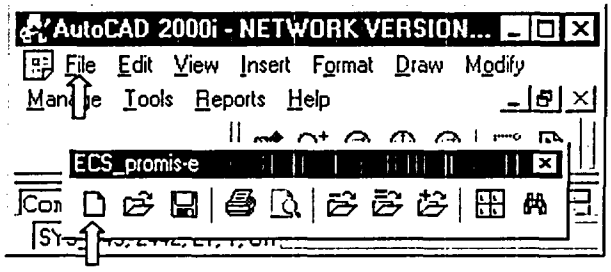
6. The Data Drawn is generated automatically. Add or change any parameters as need and click **OK**.

7. A blank page in standard format will be generated and is ready to begin drawing on it.

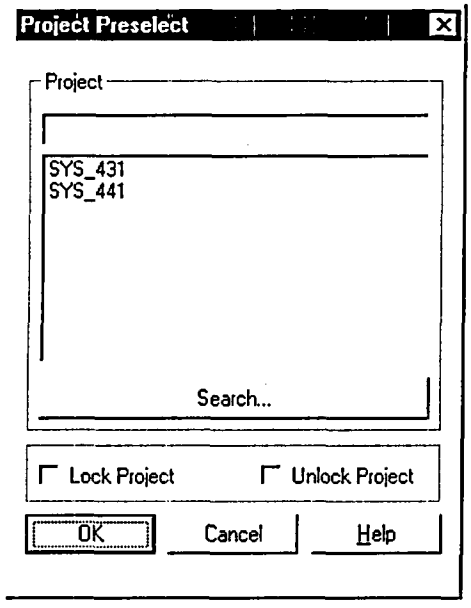
3.5.2.2

Create a New Page

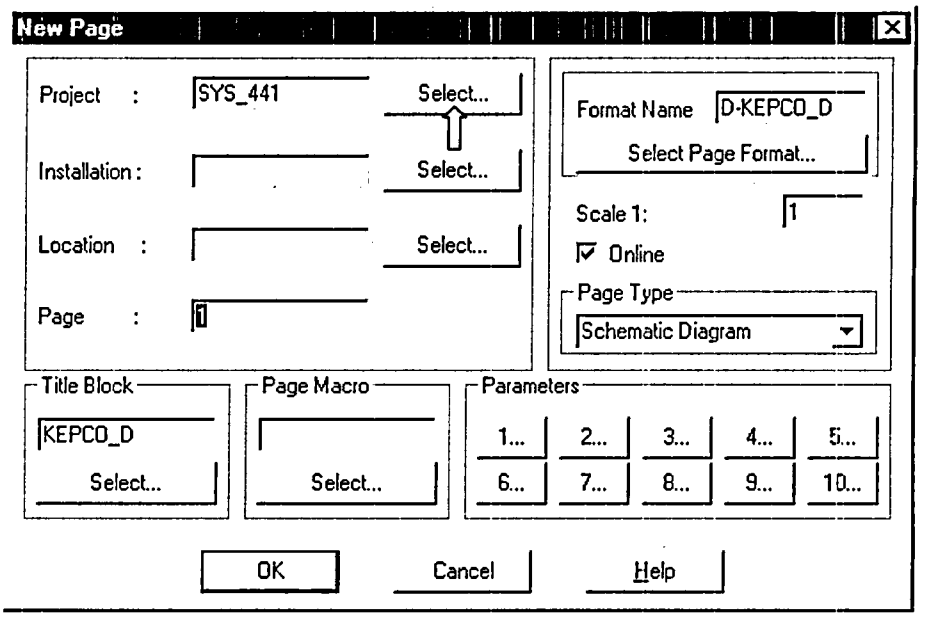
1. Click **new page** icon or click **File/New/Page** from the menu bar as shown.



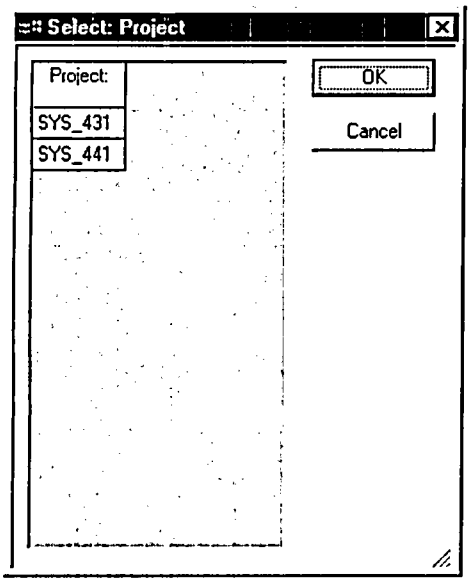
2. If no project is currently open in promis·e, the following window will show up. Select a project and click **OK**, the same window as in Step 4 in Section 3.4.1.1, Create a New Project, will show up. The same window will show up directly if there is already a project open in the promis·e.



3. If the **Project** number is not the one desired, click the **Select** button besides it and a new window will show up.



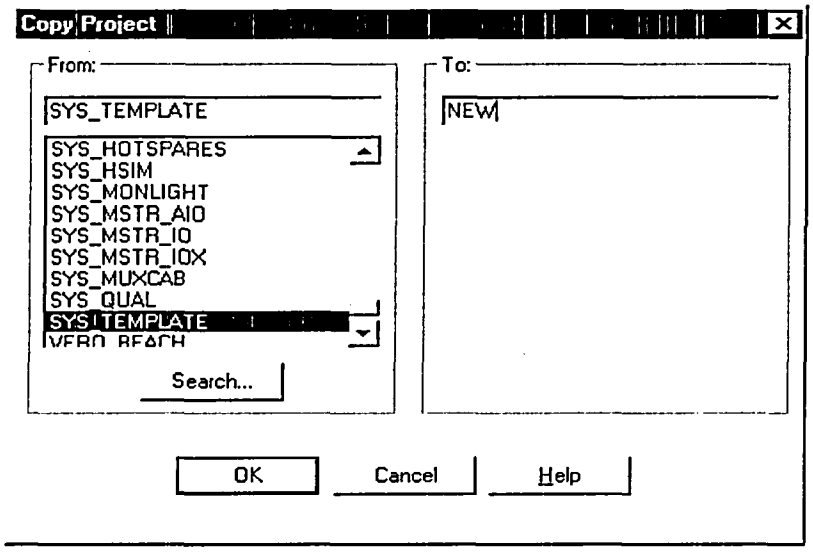
4. Select a project from the list and click **OK**.



5. Follow the instructions from Step 5 in Section 3.4.1.1, Create a New Project, to finish creating the new page.

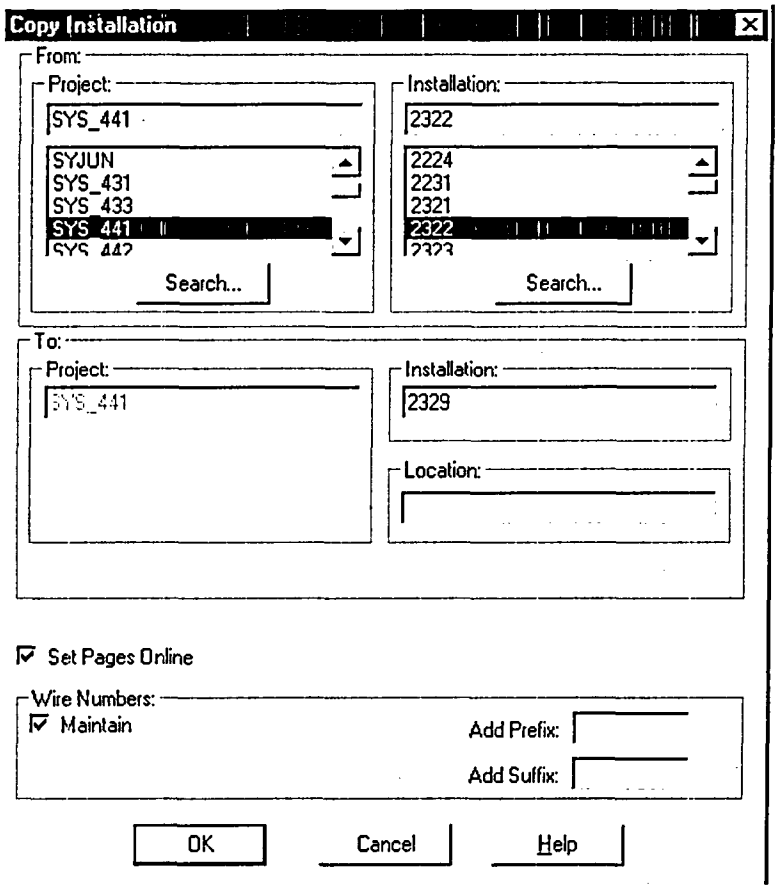
3.5.2.3 Copy a Project

Click **Manage>Project>Copy** from the menu bar and the following window will show up. Select a project in the **From** window, and enter a project name in the **To** window. Click OK and the whole project will be copied.



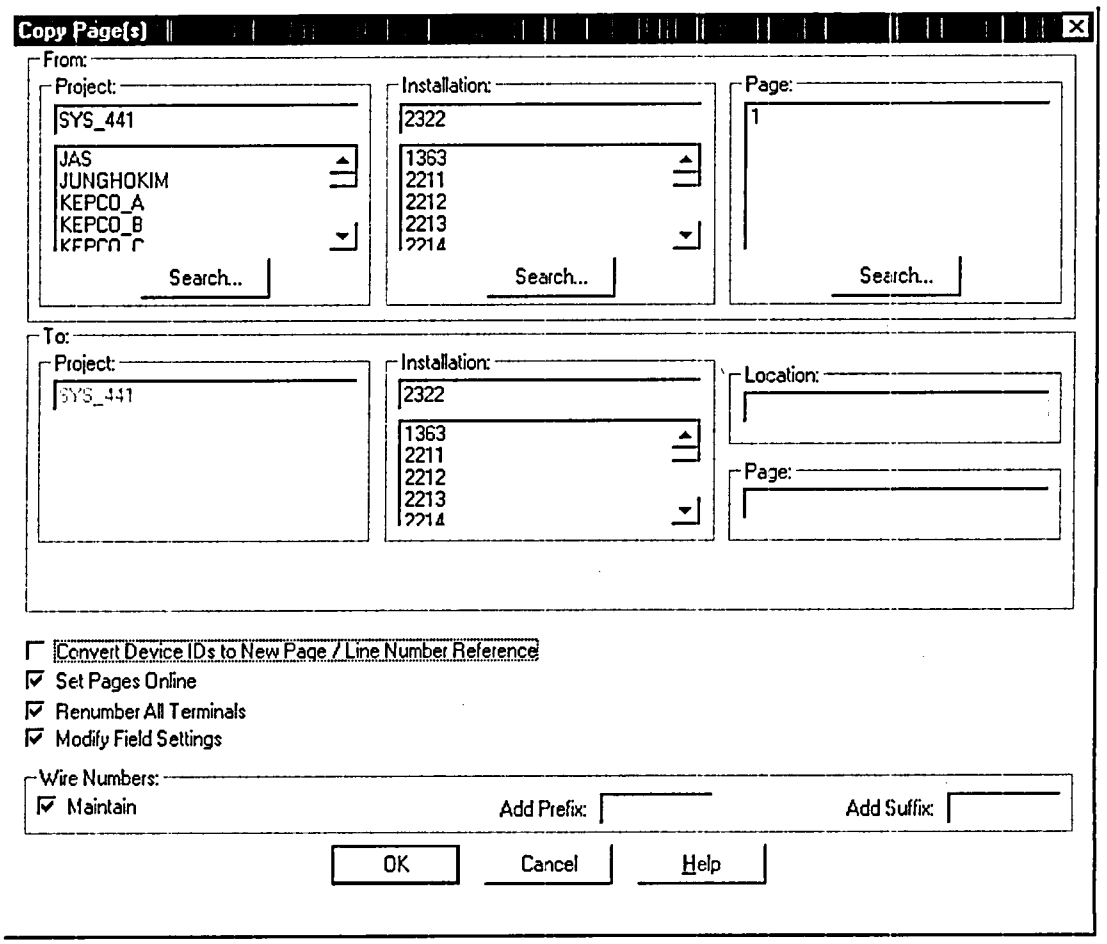
3.5.2.4 Copy an Installation

Click **Manage>Installation>Copy** from the menu bar and the following window will show up. Select an installation in the **From** window, and enter an installation in the **To** window. Click **OK** and the installation will be copied.



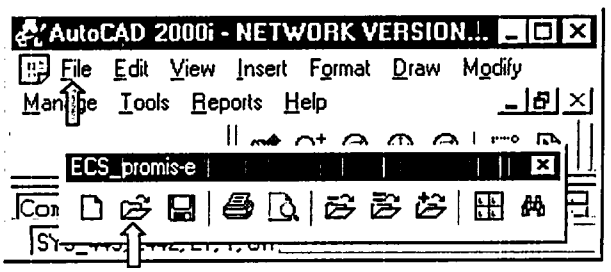
3.5.2.5 Copy a Page

Click **Manage>Page>Copy** from the menu bar and the following window will show up. Select a page in the **From** window, and enter a page in the **To** window. Click **OK** and the page will be copied.



3.5.3 OPEN A PAGE

1. Click **page open** icon or click **File/Open** from the menu bar as shown.



2. The following window will show up. The **Project** window displays all the projects currently in the promis·e. For the PCS, the projects correspond to the control systems. The **Installation** window displays all the installations for the selected Project. For the PCS, the installations correspond to the control loops under a selected system. The **Page** window displays all the drawing pages for a selected loop. Normally, there is only one page for each loop. The Preview window displays a preview of the selected page.

[

]

3. Select proper **Project, Installation** and **Page.** Then click **OK** and the drawing will show up in the promis·e window.

3.5.4 EDIT A LOOP SCHEMATIC

A loop schematic contains symbols, lines, and text. Right click on a component, an appropriate editor window will show up with a list of selections to edit. Menus can also be used to invoke the editors.

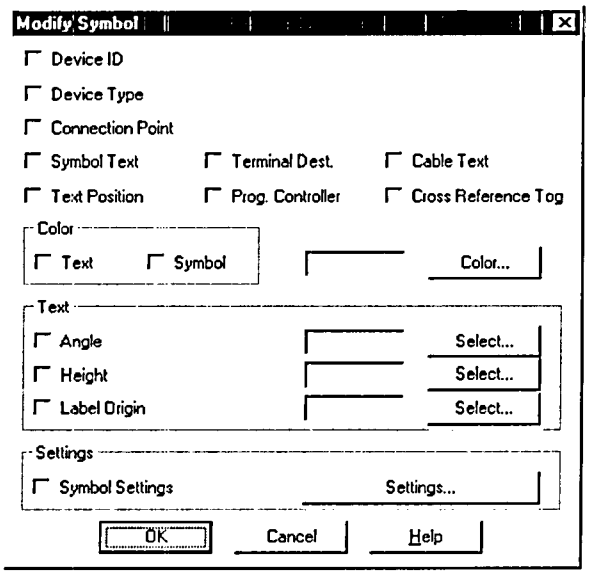
To delete, copy or move a component, click menu item **Modify>delete, Modify>copy,** or **Modify>move.** After that, left click on the component in the schematic to select, and then right click to accomplish the selected operation.

3.5.4.1

Edit Symbols

3.5.4.1.1 Modify Symbols

1. To modify a symbol, click **Modify>Symbol** from the menu bar. Left click the symbol on the schematic to select, and then right click. The following window will show up. Select the proper item to modify, and then click **OK**.



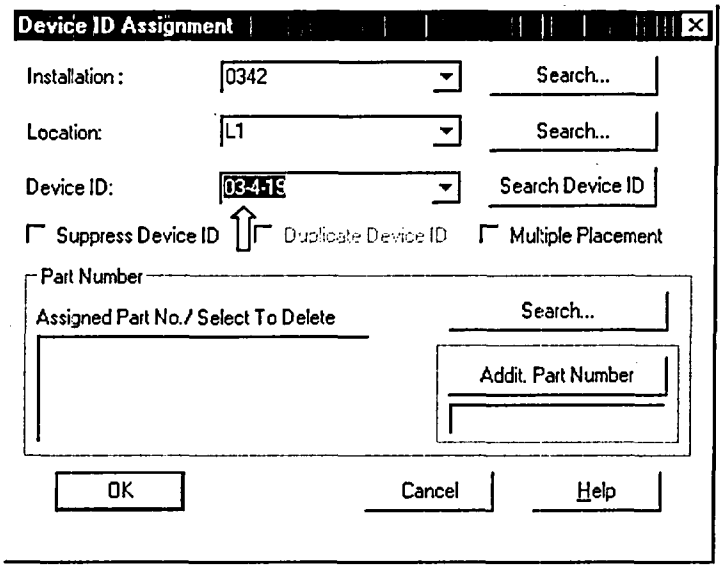
2. An edit window can also show up by right clicking on the symbol as shown. Then select the proper item to modify.

[

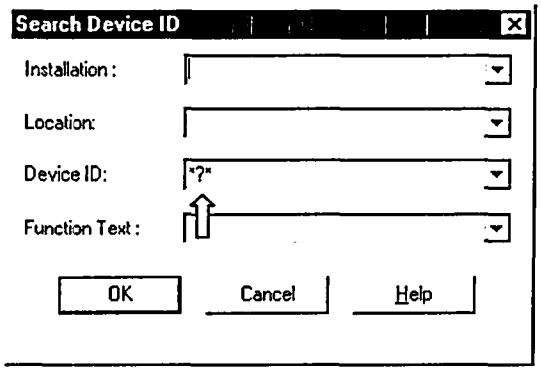
]

3.5.4.1.1.1 Edit Device ID

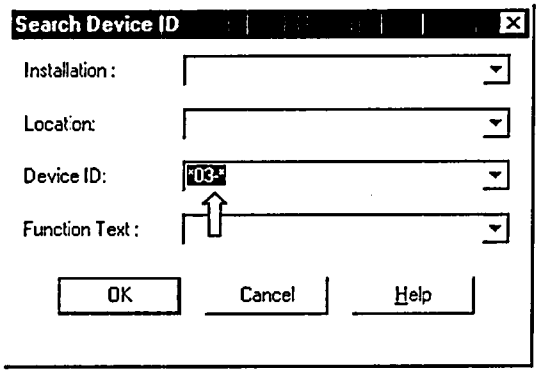
1. Select **Device ID** in the **Modify Symbol** window and click **OK**. The following window will show up. After modifying the Device ID, for example by entering "03-4-17", click **OK**.



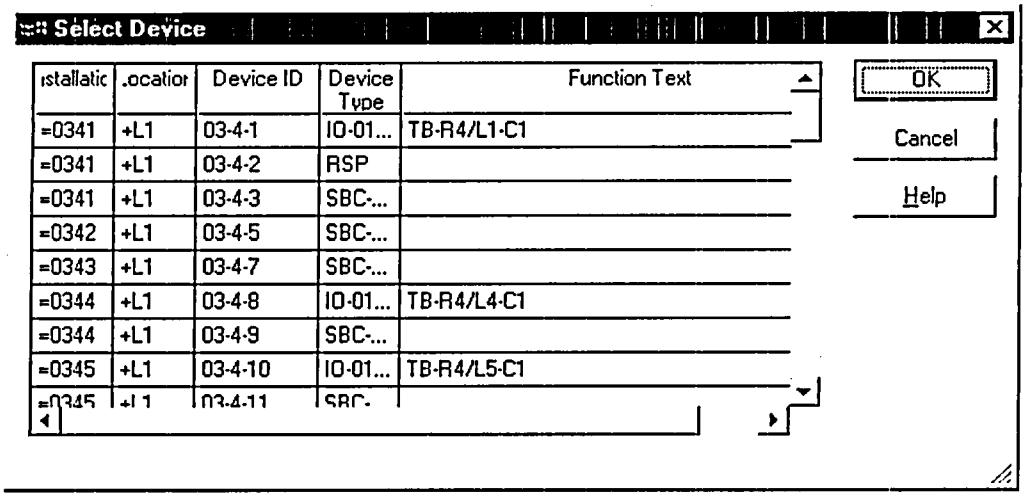
2. If **Search Device ID** is clicked, a new window will show up to obtain a list of currently defined Device ID for the system.



   a. The above search string ***?*** is for Device ID in any pattern. Refine the search by enter a search string that matches the pattern of Device IDs. For example, this circuit card is for bay 03. All Device IDs of circuit cards for this bay are like "03-##-##". Therefore enter ***03-*** and click **OK** to continue.



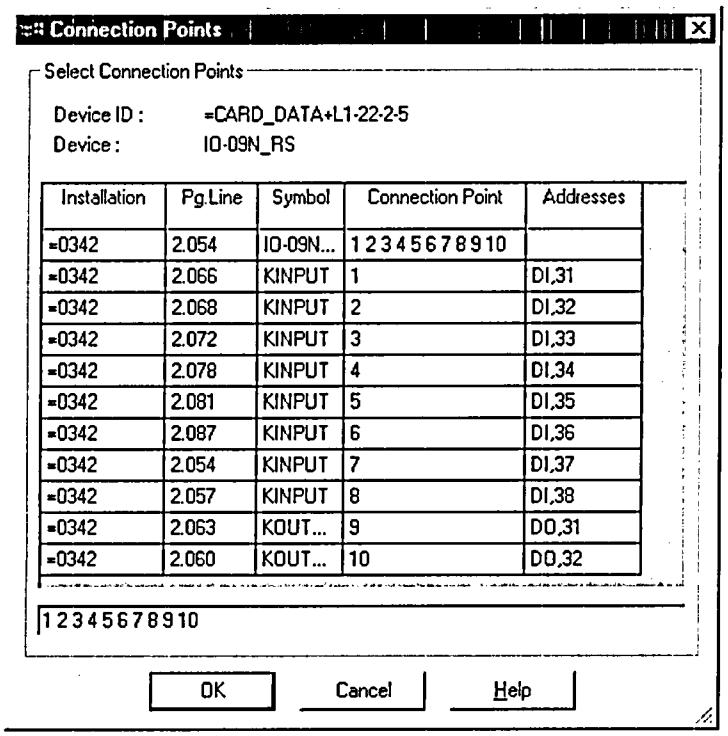
b. A new window will show up. This window lists Device Ids currently defined for the system. Make sure the new Device ID doesn't duplicate any existing Device ID.



Installatic	Location	Device ID	Device Type	Function Text
=0341	+L1	03-4-1	IO-01...	TB-R4/L1-C1
=0341	+L1	03-4-2	RSP	
=0341	+L1	03-4-3	SBC-...	
=0342	+L1	03-4-5	SBC-...	
=0343	+L1	03-4-7	SBC-...	
=0344	+L1	03-4-8	IO-01...	TB-R4/L4-C1
=0344	+L1	03-4-9	SBC-...	
=0345	+L1	03-4-10	IO-01...	TB-R4/L5-C1
=0345	+L1	03-4-11	SBC-	

3.5.4.1.1.2 Edit Connection Point

Select **Connection Point** in the **Modify Symbol** window and click **OK**. The following window will show up. Verify that all I/O points are defined, i.e. the Installation and Page Line columns are filled out. If not, the children symbols must be added to the schematic. Instructions for adding new children symbols are discussed in later sections.



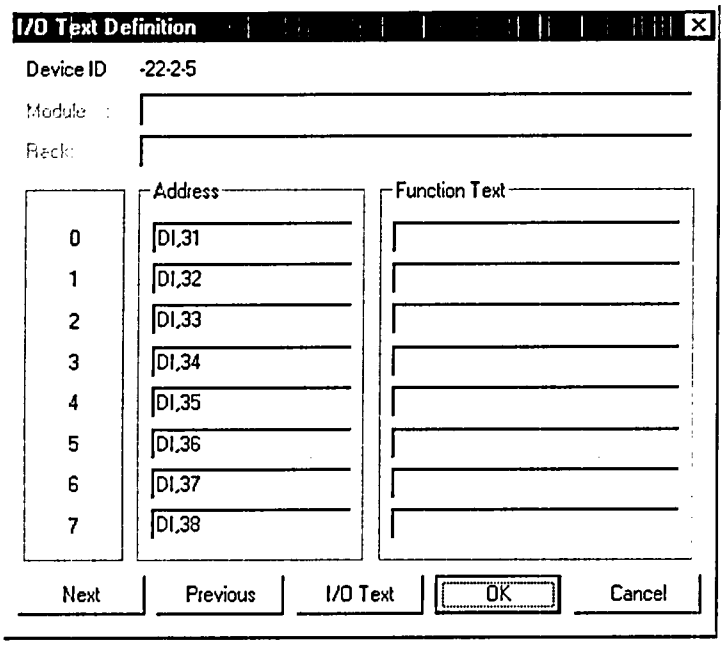
Installation	Pg.Line	Symbol	Connection Point	Addresses
=0342	2.054	IO-09N...	1 2 3 4 5 6 7 8 9 10	
=0342	2.066	KINPUT	1	DI,31
=0342	2.068	KINPUT	2	DI,32
=0342	2.072	KINPUT	3	DI,33
=0342	2.078	KINPUT	4	DI,34
=0342	2.081	KINPUT	5	DI,35
=0342	2.087	KINPUT	6	DI,36
=0342	2.054	KINPUT	7	DI,37
=0342	2.057	KINPUT	8	DI,38
=0342	2.063	KOUT...	9	DO,31
=0342	2.060	KOUT...	10	DO,32

1 2 3 4 5 6 7 8 9 10

3.5.4.1.1.3 Edit Address (PLC Text)

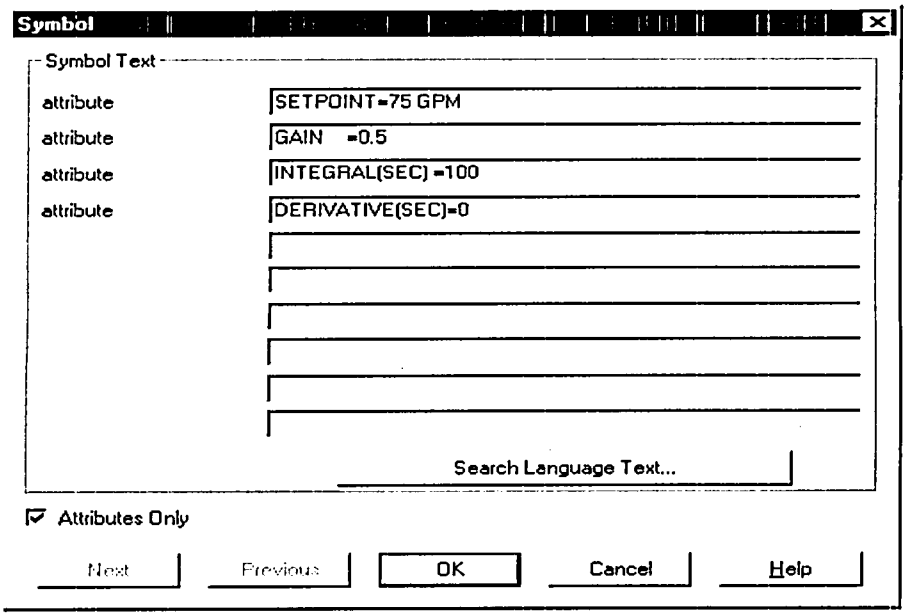
1. Right click the child symbol of the circuit card and select **PLC Text** in the pop up editor window, and the following window will show up.

2. Change addresses in the window to desired ones, and click **OK**. After changing the first point, the other points are populated. For example, change point 0 to "DI, 41" and click **OK** or **Next**, the addresses for other points change to "DI, 42", "DI, 43", etc.



3.5.4.1.1.4 Edit Symbol Text

Select **Symbol Text** in the **Modify Symbol** window and click **OK**. The following window will show up. After modifying attribute value, click **OK**.



3.5.4.1.2 Add Circuit Card Symbols

3.5.4.1.2.1 By Inserting with Macro

Using macros is a quick way to add circuit symbols to the schematic, because the parent symbol and children symbols are added all at one time and combined together for the circuit card. However, the following steps must be performed **EXACTLY** as shown to avoid extra efforts to detect and fix errors caused by adding macros improperly.

For example, to add an A103 card symbol to the schematic, perform the followings:

1. Click **Insert>Select Icon, Insert>Symbol by name,** or **Insert>Macro by name** from the menu bar, or the **Select Symbol** icon from the tool bar, and the following window will show up.
   [

]

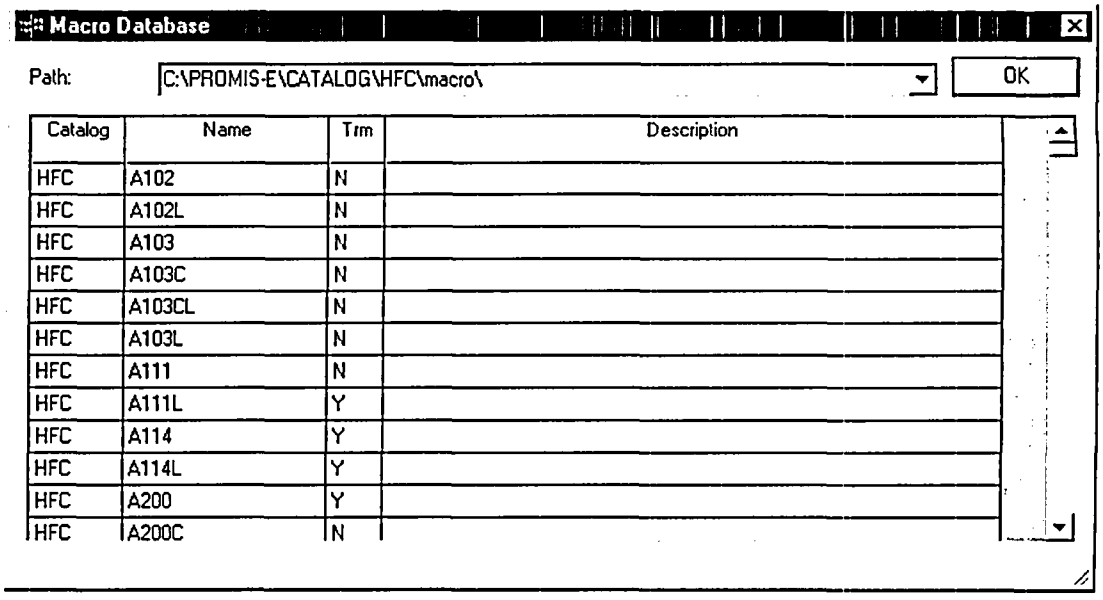
2. Make sure that **Macro** is selected as shown. Select **Yes** for the **Macro Settings:>Maintain Device IDs As Created**. Then select the A103 icon from icon list or enter the exact device type name "A103" in the **Name** box and click **OK**.
   [

   ]

3. If the symbol name is not known, use Search button to search the symbol name.
   [

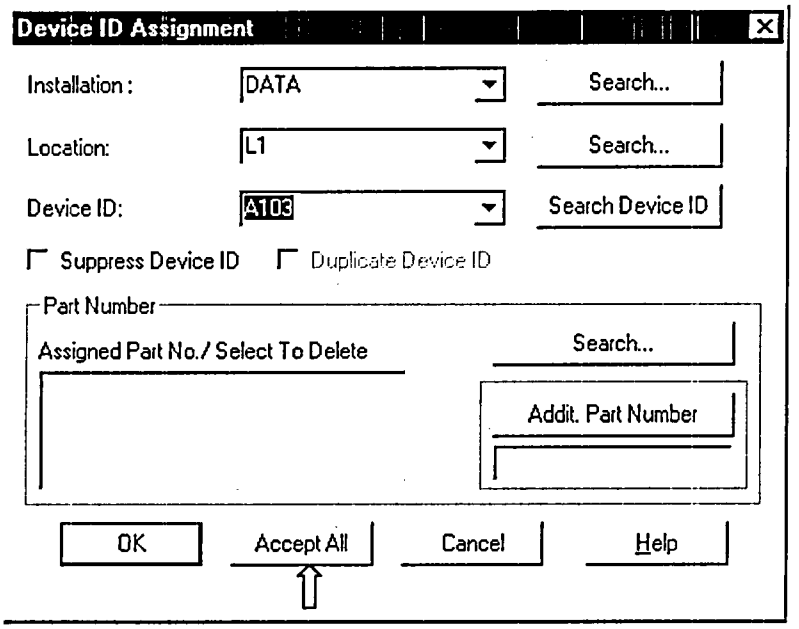
]

4. Select the desired symbol name and click **OK** to continue.

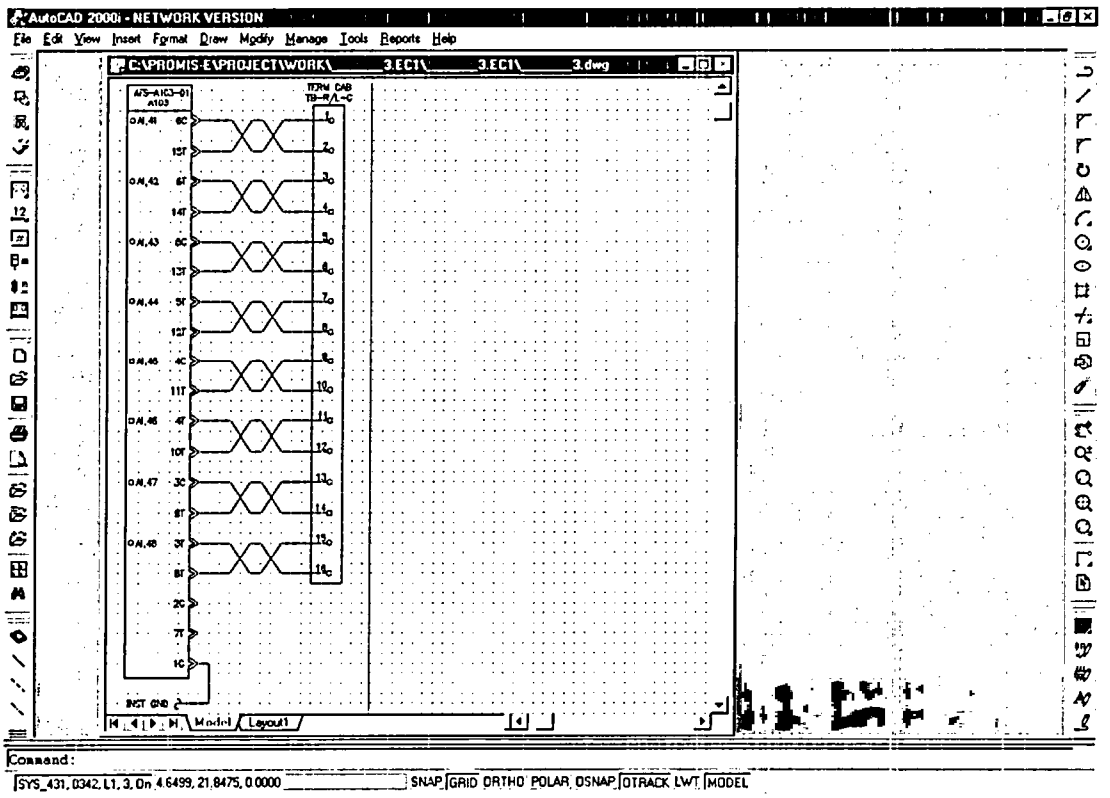


Catalog	Name	Trm	Description
HFC	A102	N	
HFC	A102L	N	
HFC	A103	N	
HFC	A103C	N	
HFC	A103CL	N	
HFC	A103L	N	
HFC	A111	N	
HFC	A111L	Y	
HFC	A114	Y	
HFC	A114L	Y	
HFC	A200	Y	
HFC	A200C	N	

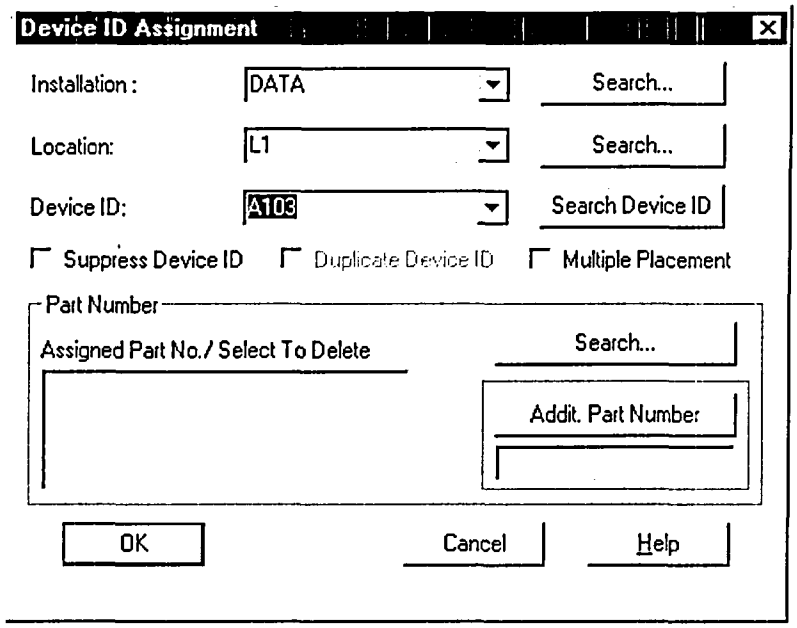
5. After a new window pops up, click **Accept All.** Do not change anything in this dialog window.



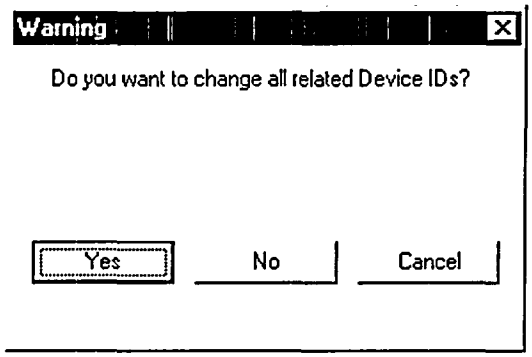
6. Place the symbol at the desired location in the schematic.



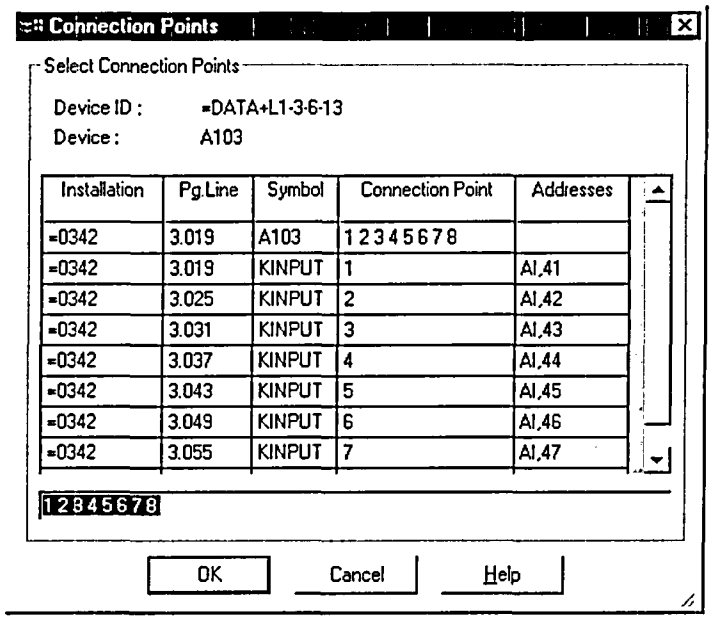
7. Modify the Device ID for the new symbol to the desired one. For example, in the following window, change "A103" to "03-4-19", and click **OK**.



8. Click **OK** to change the Device ID for all related components (the card, parent symbol and children symbols) all at once.



9. Select **Connection Point** in the **Modify Symbol** window and click **OK.** Verify that the Installation and Page Line columns are not blank to ensure that al I/O points for the newly inserted card are defined.



3.5.4.1.2.2 By Inserting Symbol Only

1. Click **Insert>Select Icon**, or **Insert>Symbol by name** from the menu bar, or the **Select Symbol** icon from the tool bar, and the following window will show up.

2. Make sure that **Symbol** is selected as shown and the **Macro Settings:>Maintain Device Ids As Created** shows **No**. Then enter the exact device type name "A103" in the **Name** box and click **OK** to continue.

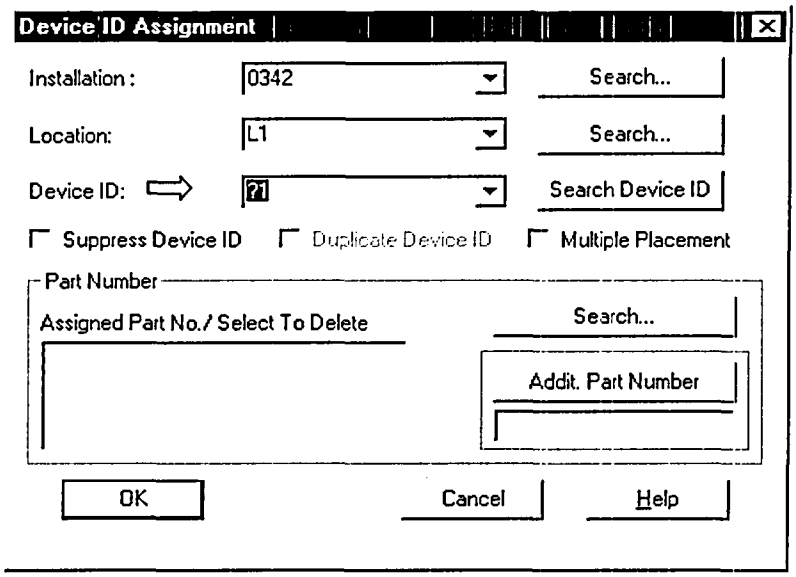
Caution

Enter name only and do NOT select an icon from the icon window. Clicking on an icon causes the symbol to be added with incorrect settings for its associated macros.

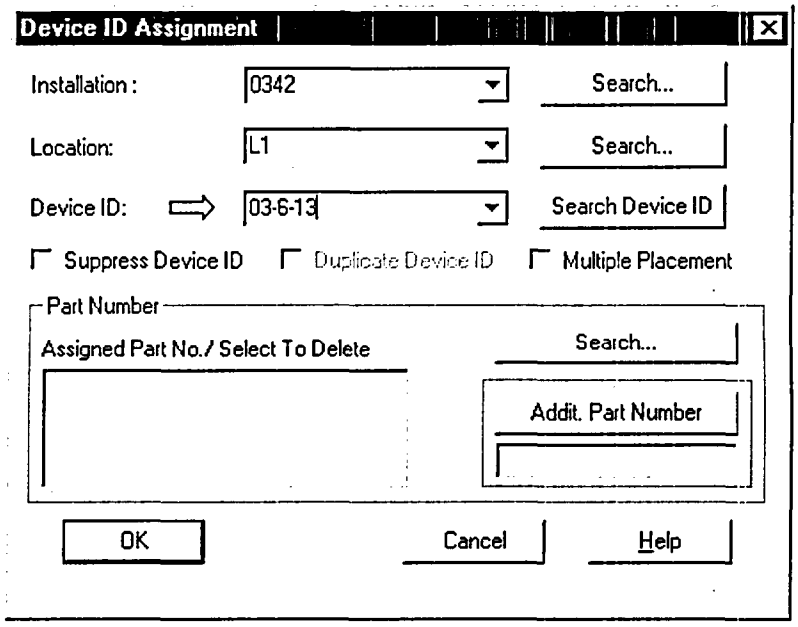
[

]

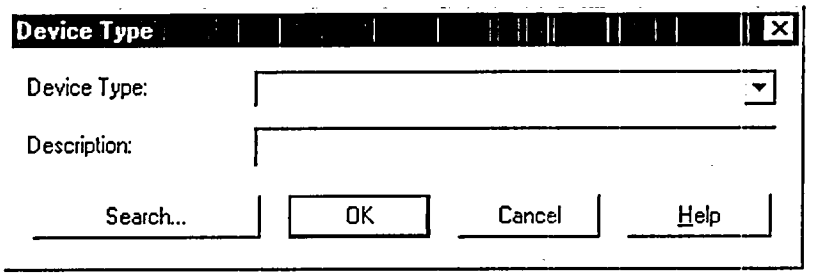
3. After the symbol is placed in the schematic, the following window will show up:



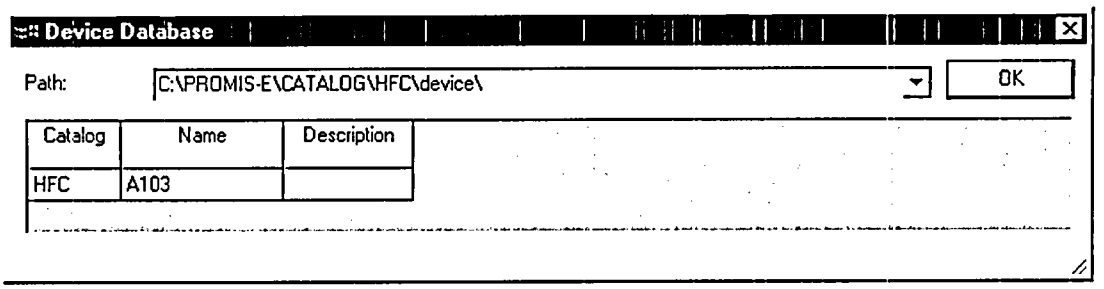
4. Enter Device ID, for example "03-6-13" and click **OK** to continue.



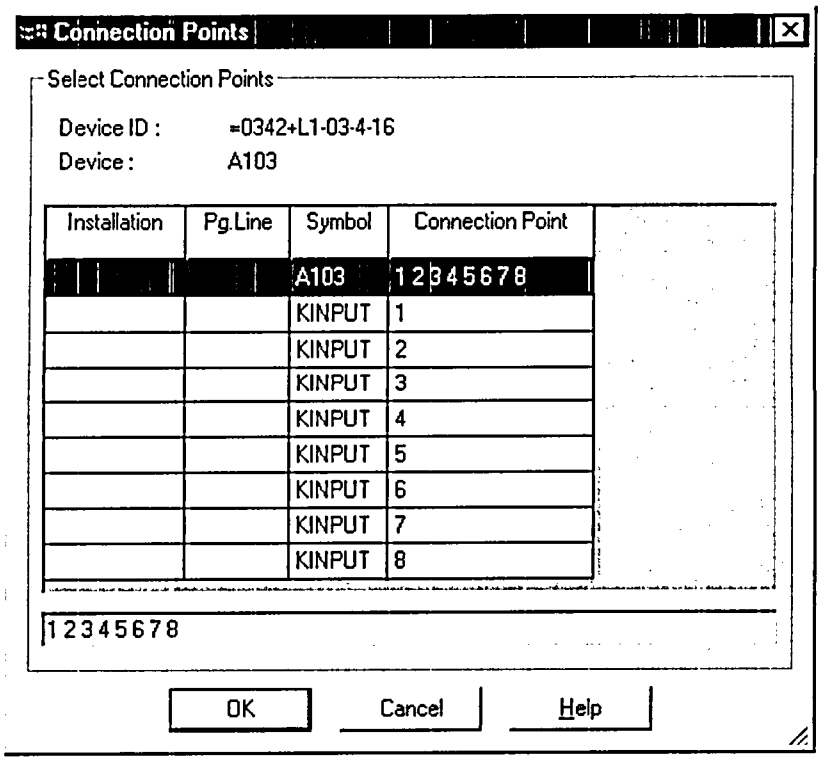
5. The following window will pop up. Enter the exact name for device type, for example "A103", and click **OK** to continue.



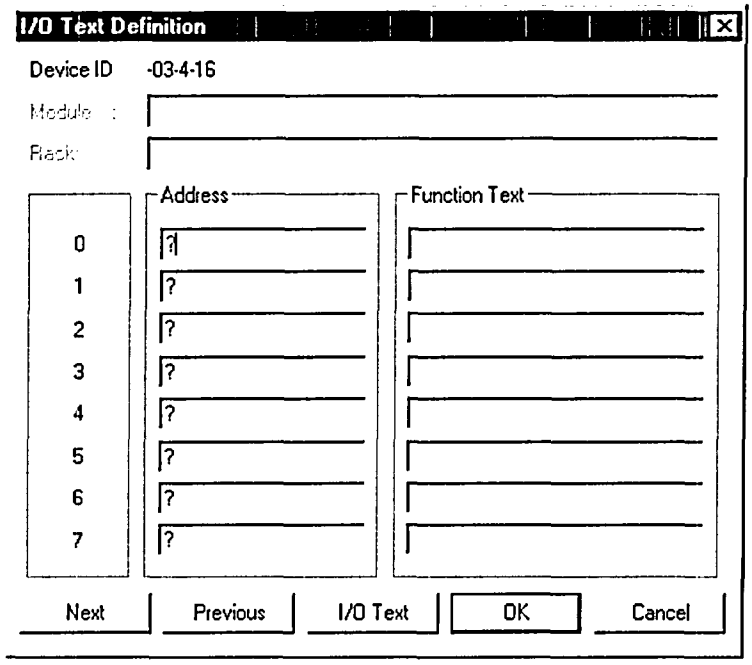
6. Alternatively, click **Search** button, and then select "A103" in the following window.



7. The following window will show up for defining connection point.



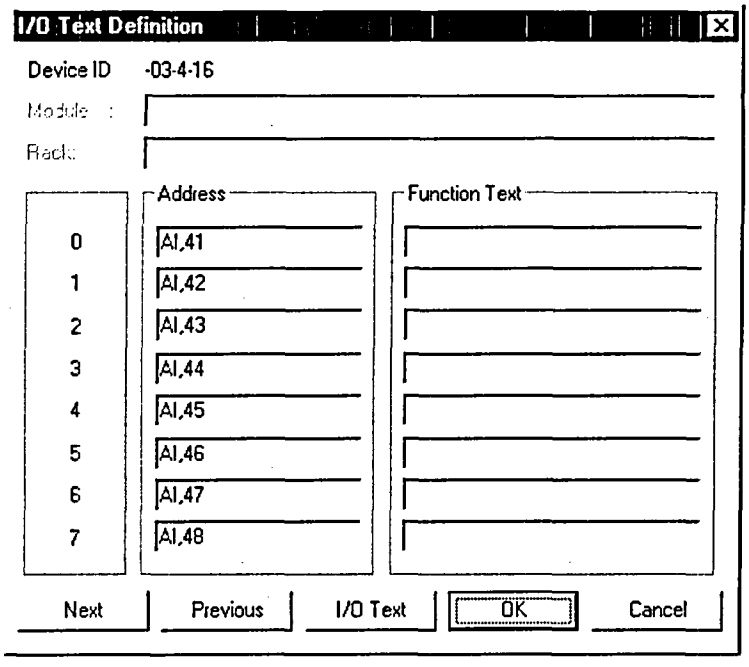
8. Verify that the parent symbol and child symbols are correct for the device type. Click **OK,** and the following window will show up.



9. Enter the predefined point address for point 0, and the address for other points will populate from point 0. Then click OK.

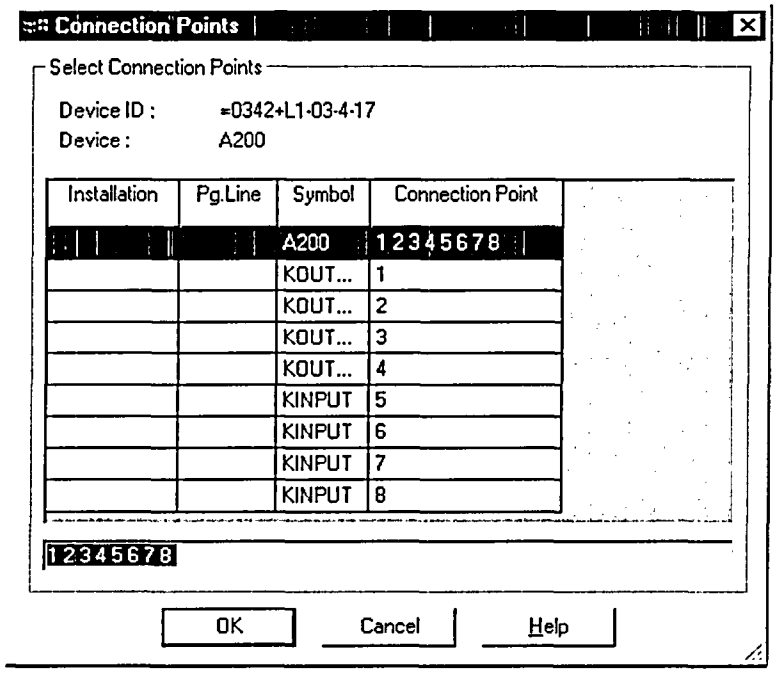
Caution

Connection point numbers are 1 based, for example from 1 to 8, whereas the corresponding point addresses are 0 based, for 0 to 7.

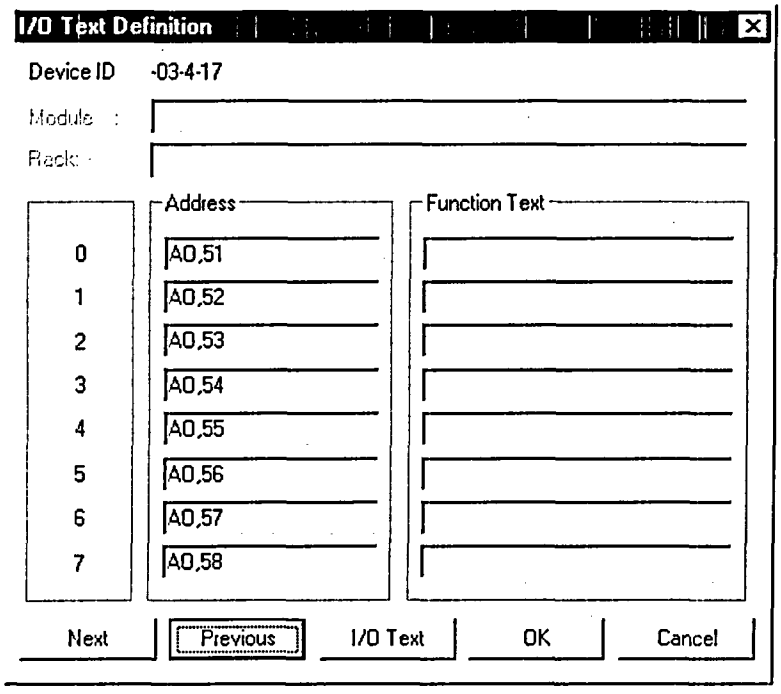


10. For circuit cards with both input and output points, make sure that the populated addresses are correct.

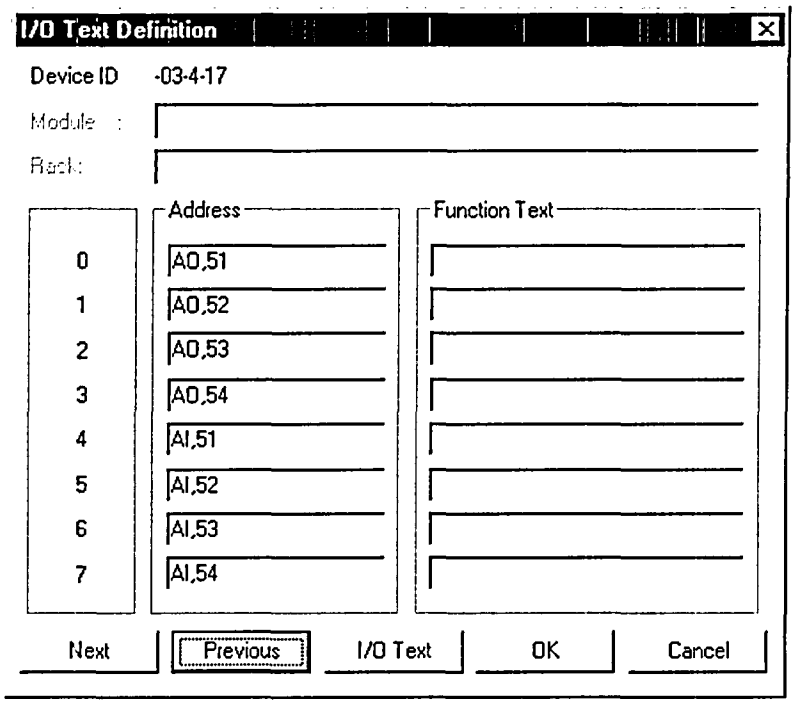
    a. For example, an A200 card has the following child symbols.



    b. Enter the first point, and the other points could be populated as follows:

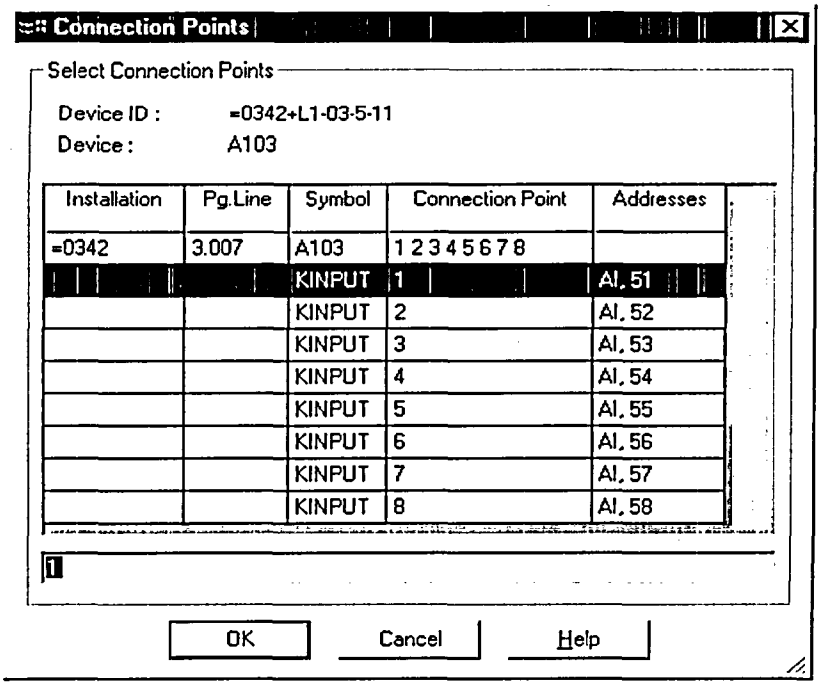


c. Make sure to correct them for the other types of points.



11. Place the symbol at the desired location in the schematic.

12. Select **Connection Point** in the **Modify Symbol** window and click **OK** to have a look at the connection point list. The Installation and Page Line columns are blank, which means that the children symbols need to be added.

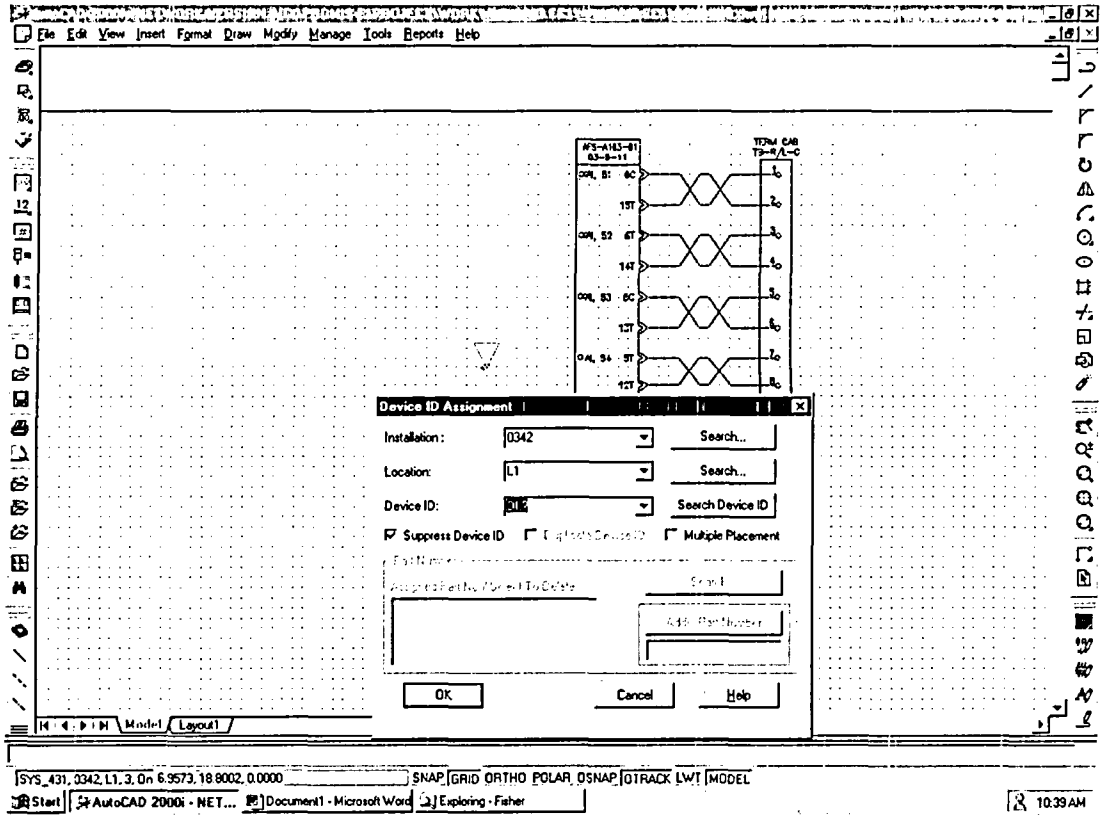


13. Click **Insert>Select Icon**, or **Insert>Symbol by name** from the menu bar, or the **Select Symbol** icon from the tool bar, and the following window will show up.
[

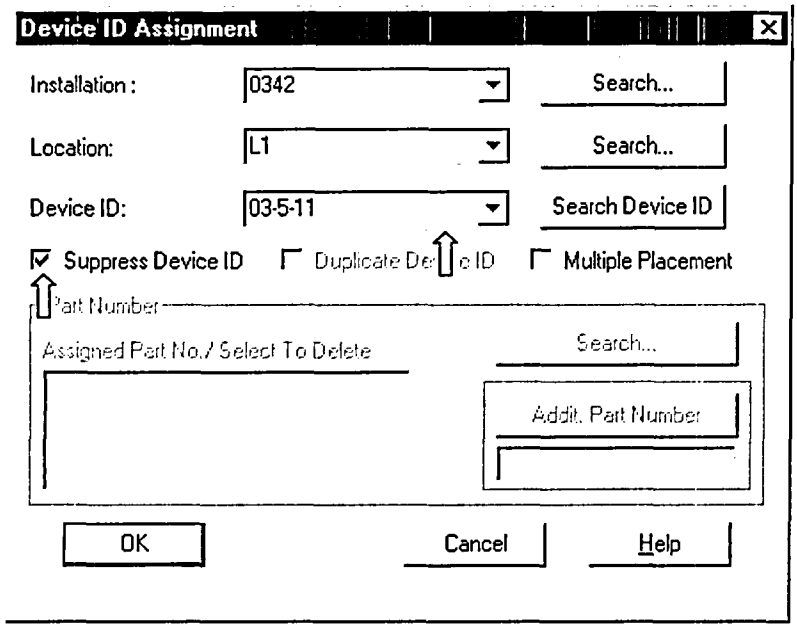
⇧

]

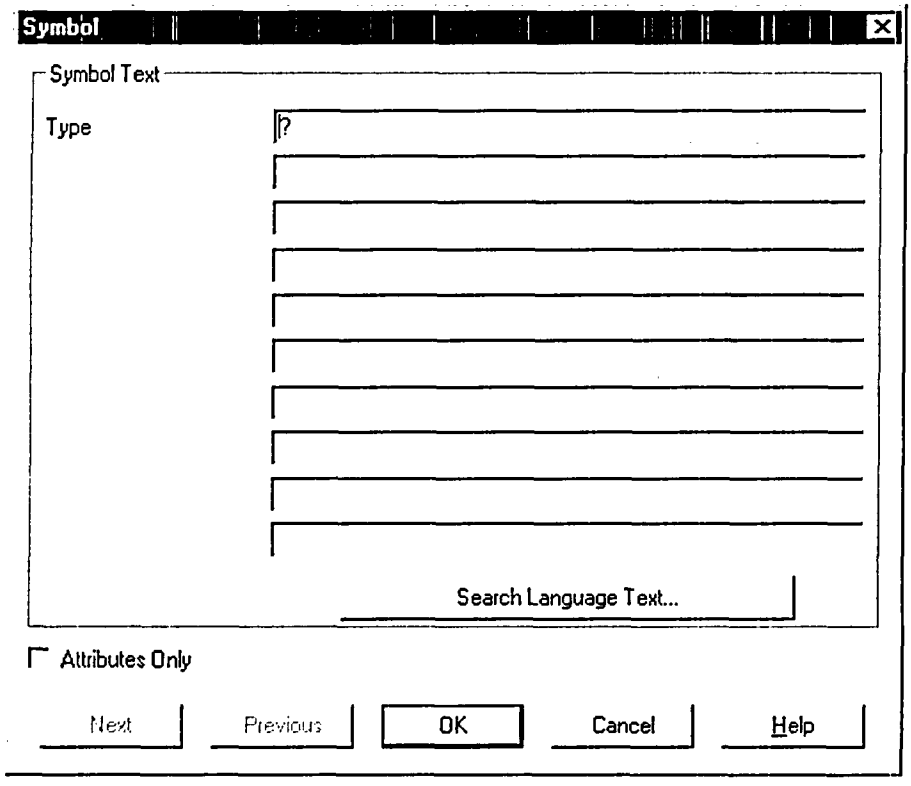
14. Enter "KINPUT" in the **Name** box as the device type for a child symbol of A103 card is "KINPUT". Click OK and place the child symbol to somewhere close to the card symbol. And the following window will show up.



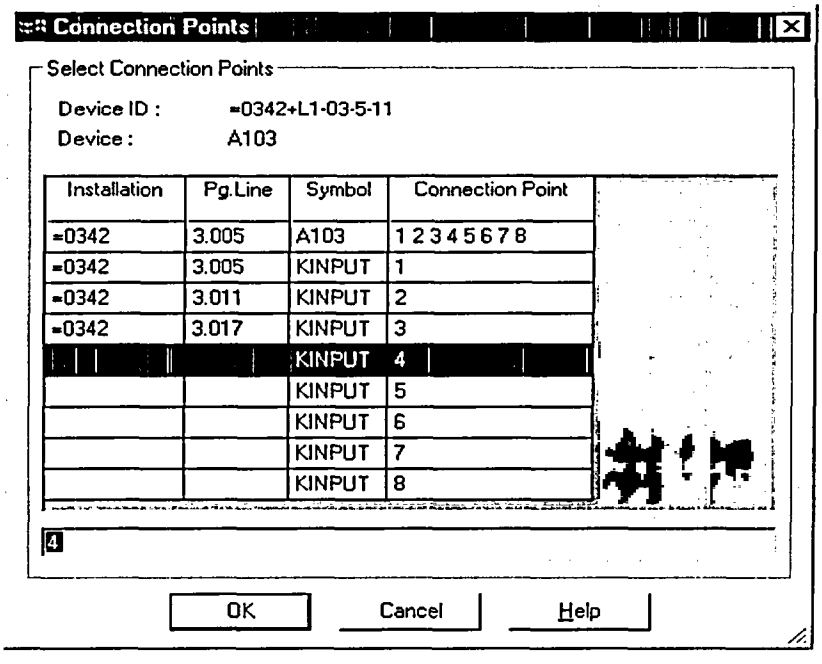
15. Enter the same device ID as the parent symbol, for example "03-5-11". Check **Suppress Device ID** so that this device ID won't be displayed on the schematic. Click **OK** to continue.



16. Delete "?" as we don't need any text for the child symbol.



17. After adding several child symbols in this way, check the **Connection Points** window of the circuit card to make sure that these points have been added to the circuit card.



18. After all the children symbols are added, move them to the circuit card.

3.5.4.1.3 Add a Logic Gate

The following paragraphs outline the method for adding a "NOT" gate input to MS, 53.
[

]

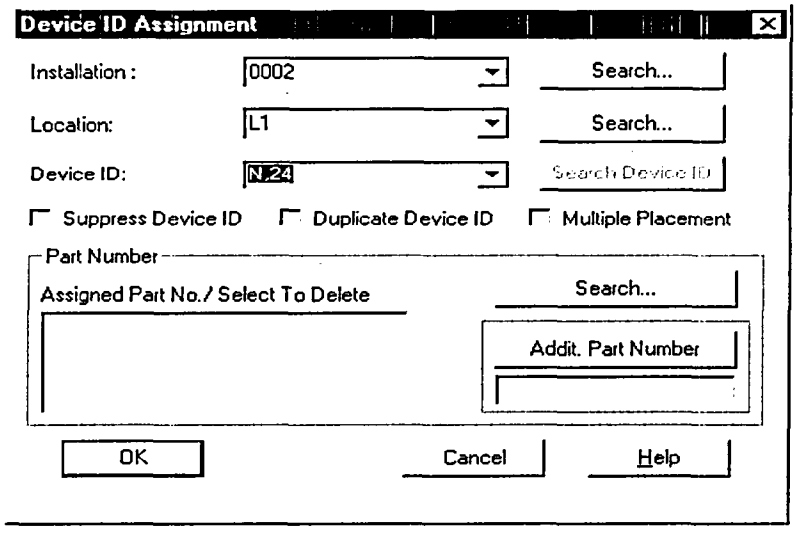
1. Select **Insert>Symbol by name**. The following dialog will show up. Select a "NOT" gate and click **OK** to continue.
   [

   ]
2. Place the "NOT" gate at the desired place. The line automatically breaks around the gate.

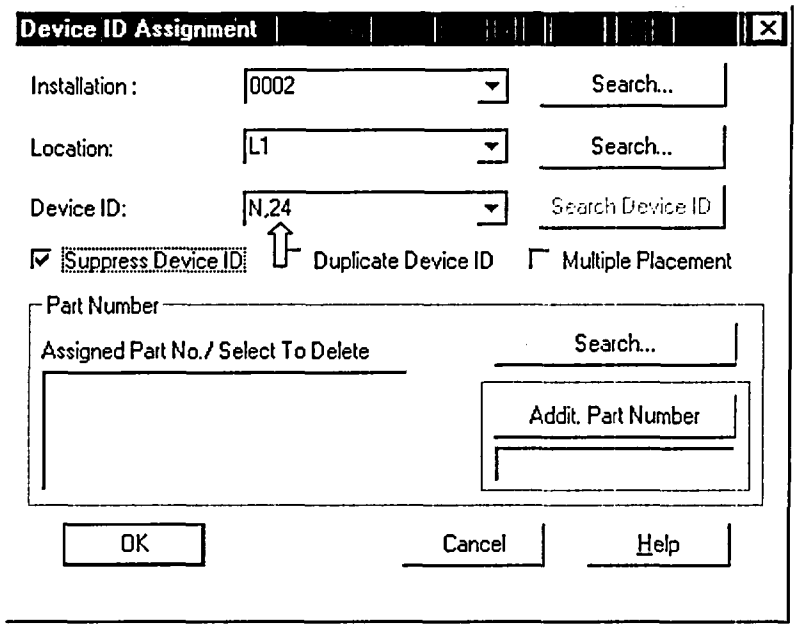
[

]

3. By default, a Device ID will appear at the right corner of the newly added "NOT" gate. However, it is not necessary to have a Device ID for logic symbols on the drawings. Click on the "NOT" gate, and the following dialog will show up.



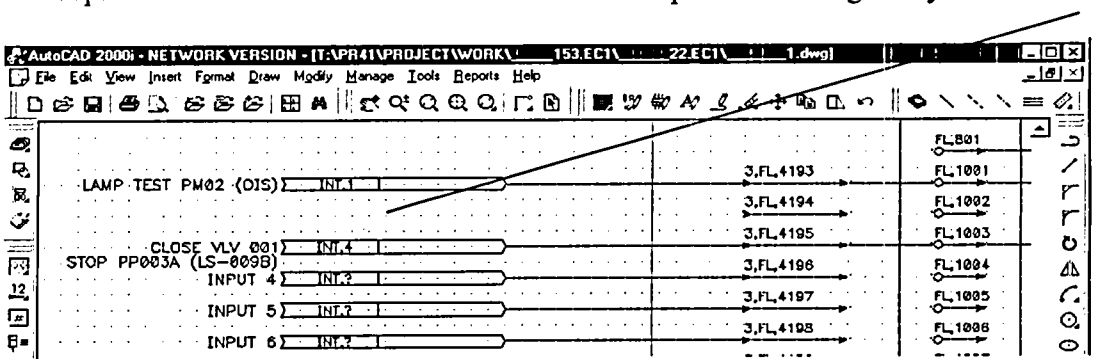
4. Check **Suppress Device ID** in the dialog as shown and click **OK**, the Device ID will hide from display.


Device ID Assignment
Installation :
0002
Search...
Location:
L1
Search...
Device ID:
N.24
Search Device ID
Suppress Device ID
Duplicate Device ID
Multiple Placement
Part Number
Assigned Part No./ Select To Delete
Search...
Addit. Part Number
OK
Cancel
Help

3.5.4.1.4 Add a Process Control Permissive

1. Find the source device ID of the signal, e.g., 3,FL,11165 from the source loop.
   [

   ]
2. Add a DDB filter "KINTC" symbol with the same device ID 3,FL,11165 to destination loop. The window below shows the destination loop before adding the symbol.

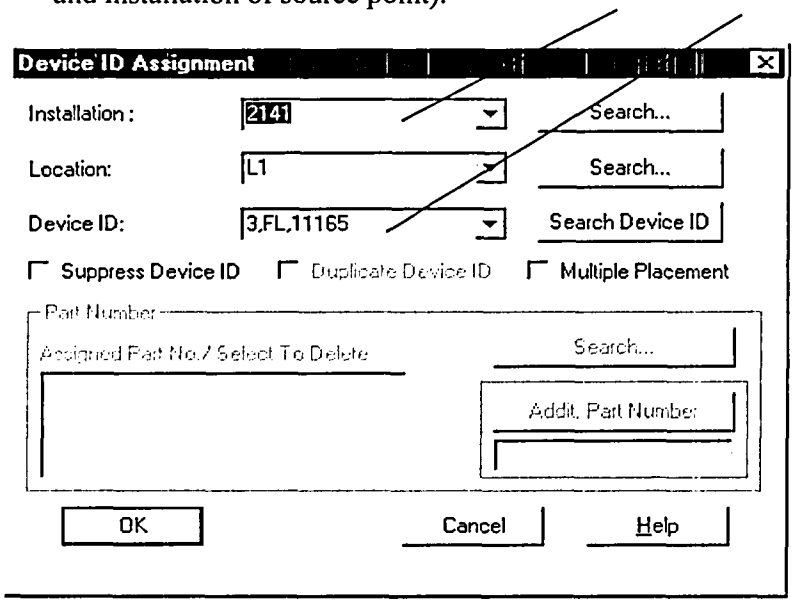


3. Follow the same steps as adding a circuit card symbol. Just enter "KINTC" as the name, select Symbol, and click OK.

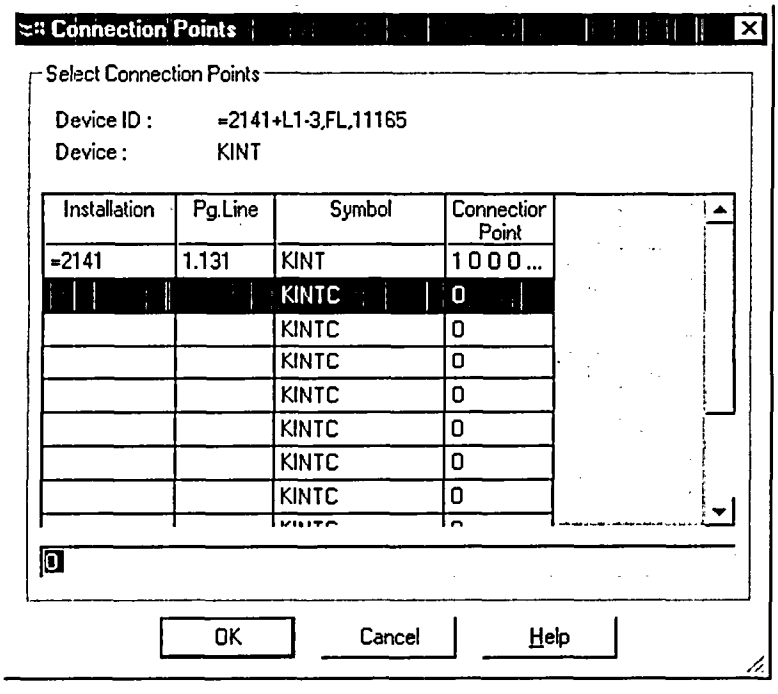
[

]

4. When the following window shows up, enter device ID and installation (the device ID and installation of source point).



5. Click OK when the following table shows up.



6. Connect the new"KINTC" symbol to ICL flag "TF_R" symbol by drawing a wire line between them.

[

]

7. Connect ICL flag "TF_R" symbol to loop flag "FI_R" symbol by drawing a wire line between them.

[

]

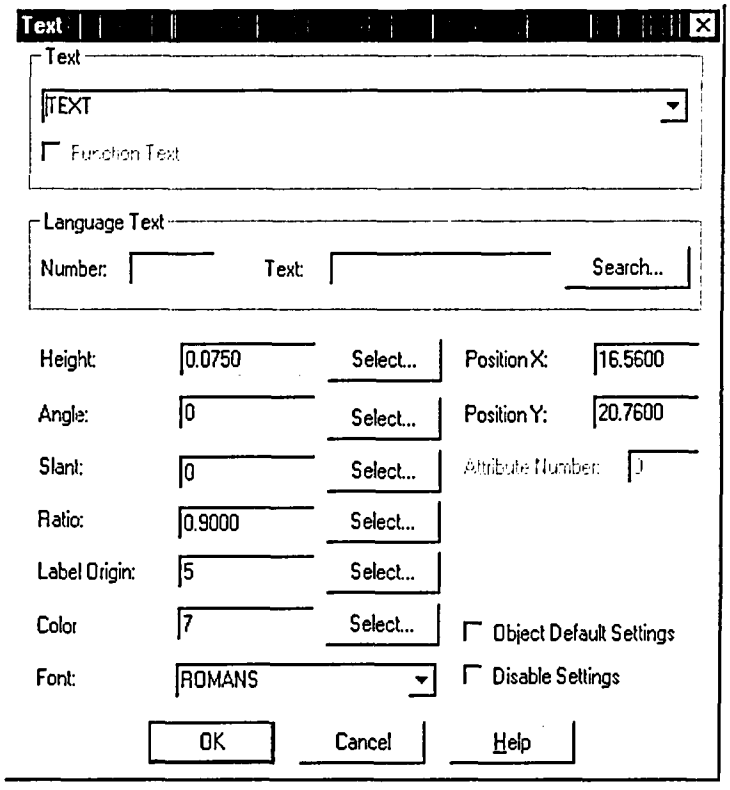
8. Connect loop flag "FI_R" symbol to logic gate by adding a wire line.

[

]

3.5.4.2 Edit Text

Text here refers to stand-alone text. To add text to the drawing, click **Draw>text** from the menu bar, and the following window will show up. After setting up attribute value, click **OK**.



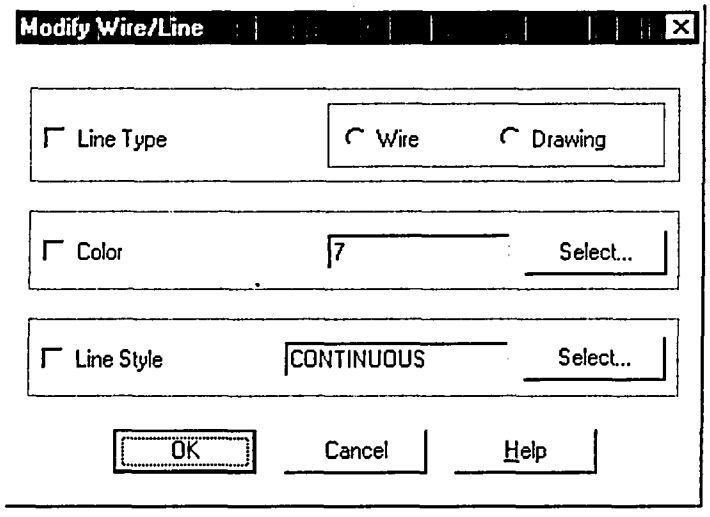
To edit text, right click the text on the drawing and then make changes in the editor window.

3.5.4.3 Edit Wire lines

Wire lines carry the signals from one symbol to the next. They are used to connect all logic symbols, I/O symbols, ICL and DDB symbols within a loop or to and from the loop.

1. To add a wire line, click **Draw>wire>continuous** from the menu bar, and place the wire line at the desired location in the schematic.

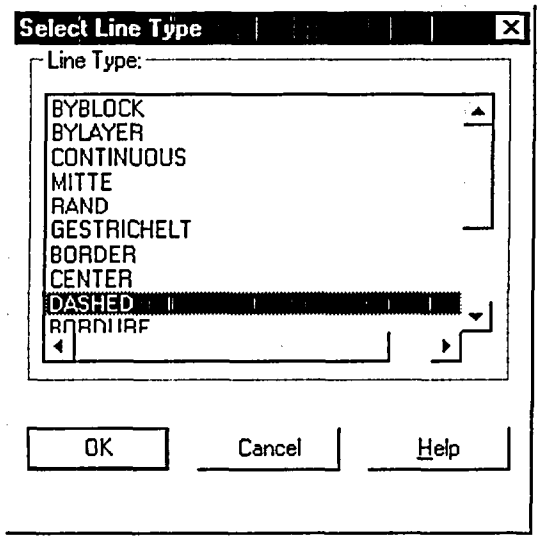
2. To modify a wire line, right click the wire line and select **Modify Line**, or click **Modify>Line** from the menu bar, select the line and then right click. The following window will show up with options to modify wire lines. The **Wire** option should always be selected to connect logic symbols.



NOTE

Drawing lines are used for non-logic related information where no data is passed between the line connections. Drawing lines are not used to connect logic symbols.

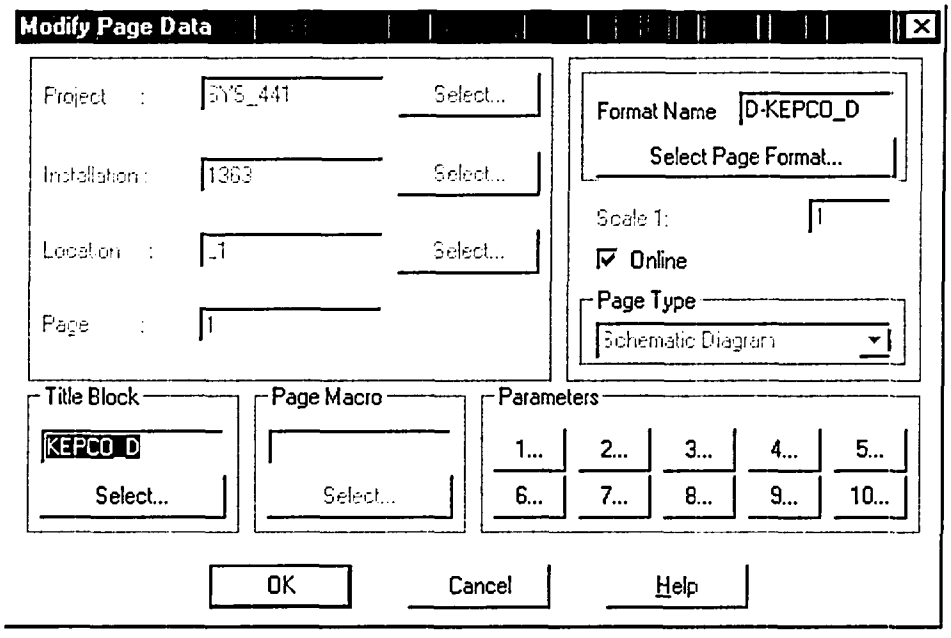
3. Select **Line Style** in the **Modify Wire>Line window** and click **Select.** The following window will show up. **Continuous** wire lines are used in most cases. **Dashed** are used to show digital signals on analog logic diagrams.



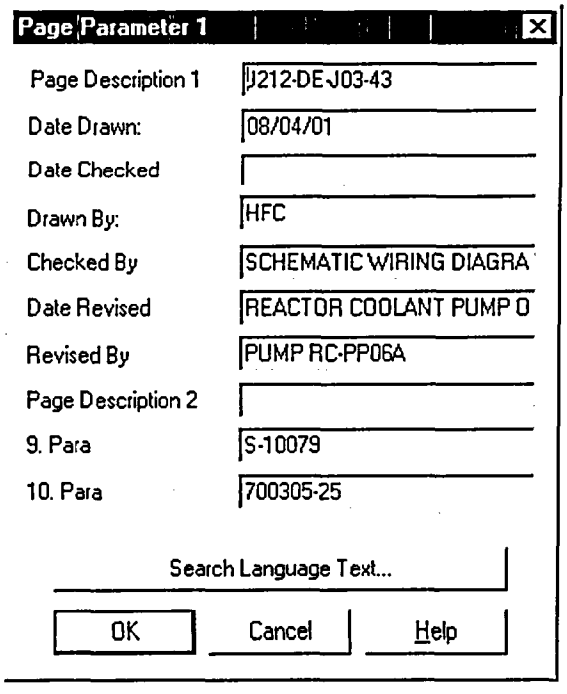
3.5.4.4

Edit Page Variables

1. Select **Manage>Page>Modify,** the following dialog will show up. Each **Parameter** button manages different page Variables.



2. Selecting **Parameters 1** can access the dialog as shown to change current revision and current date of the drawing.



Section 4

promis·e REPORTS EXTRACTION

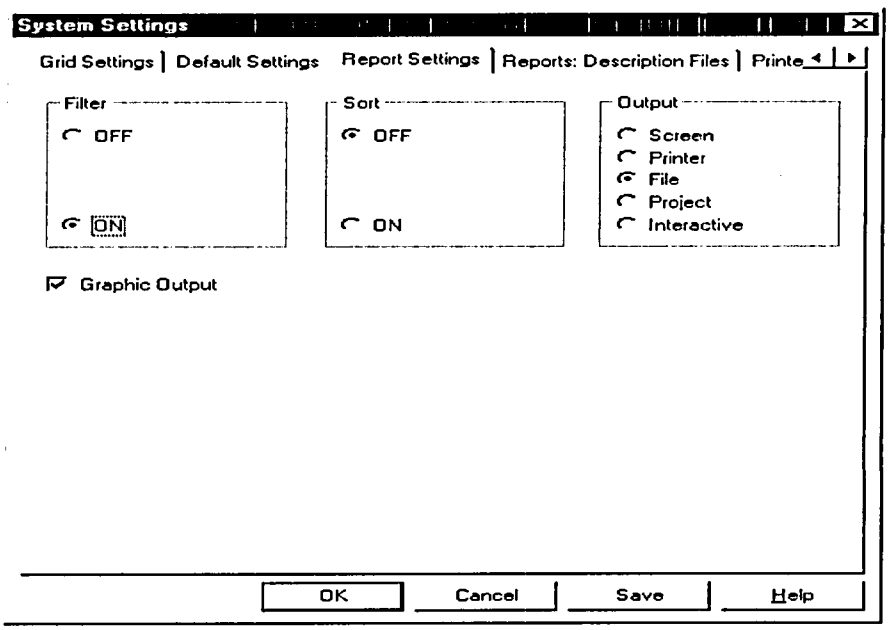
4.1 PRELIMINARY CONFIGURATION

Install HFC EWS/MCRT software suite and create a job for the control system being developed. Refer to the EWS User's Guide and the MCRT User's Guide for installation and configuration information.

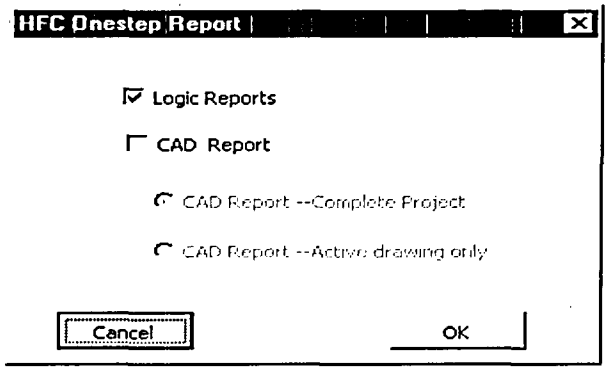
4.2 GENERATION OF REPORT FILES

The following procedures describe how to generate One-Step report files using promis·e.

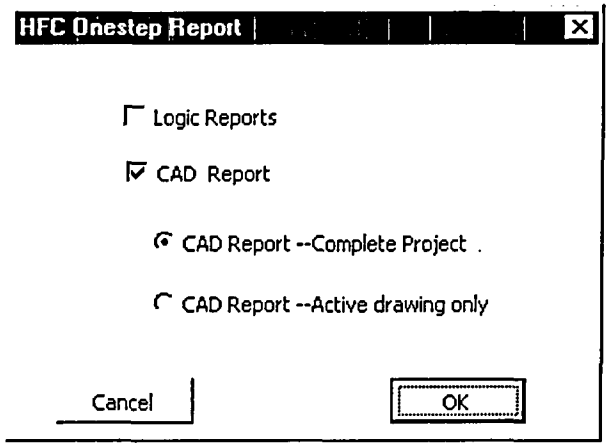
1. Set up system settings of features in reports by clicking **Tools > System settings** on the menu bar of promis·e. Enable **Filter,** disable **Sort,** and select **file output.**



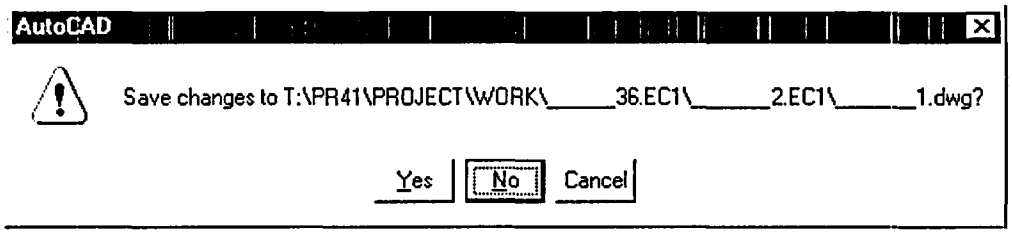
2. Click **Tools > HFC Report** on the menu bar, and the following dialog box will pop-up. Select the type of reports to be generated and click **OK.**



3. If **CAD Report** is selected in step 1, the ".txt" CAD report files will be generated and saved under HFControls/OneStep/System#/CAD_DATA. CAD reports can be selected for either the complete project or the current drawing only by selecting the appropriate target.

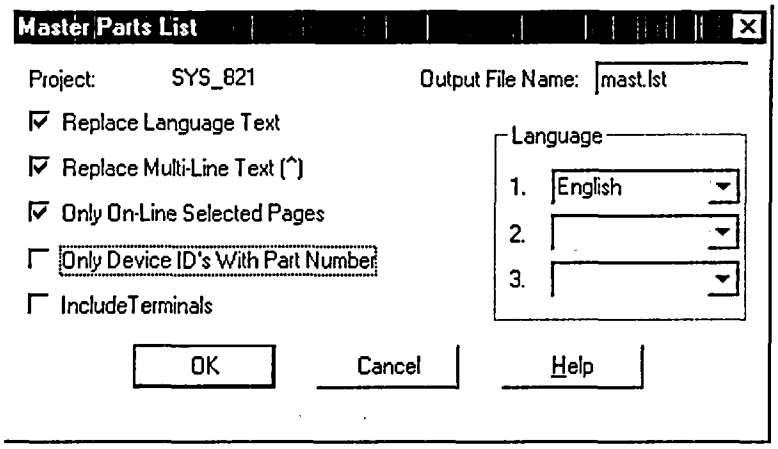


When the dialog box below appears, select **No**.

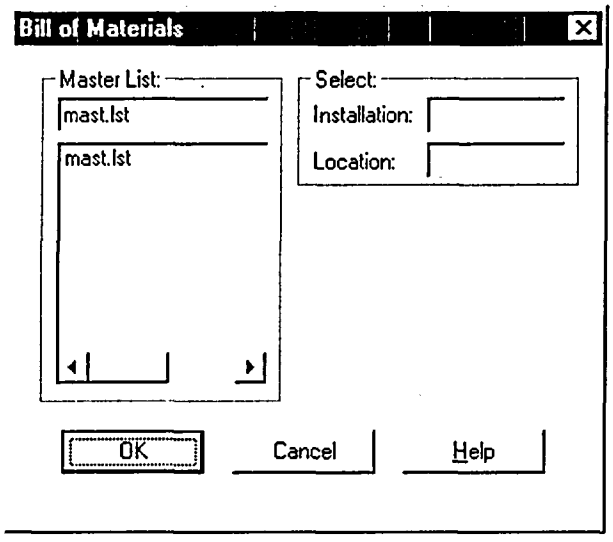


4. If **Logic Reports** is selected in step 1, several dialog boxes will appear to permit selection of specific options.

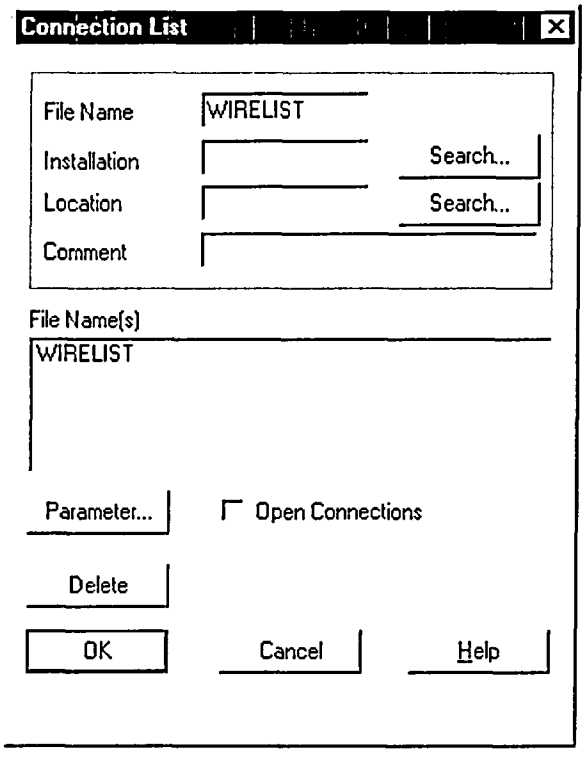
5. When the dialog box below pops up, select options as indicated. Make sure "**Only device ID's with part number**" is **NOT** selected. Then click **OK**.



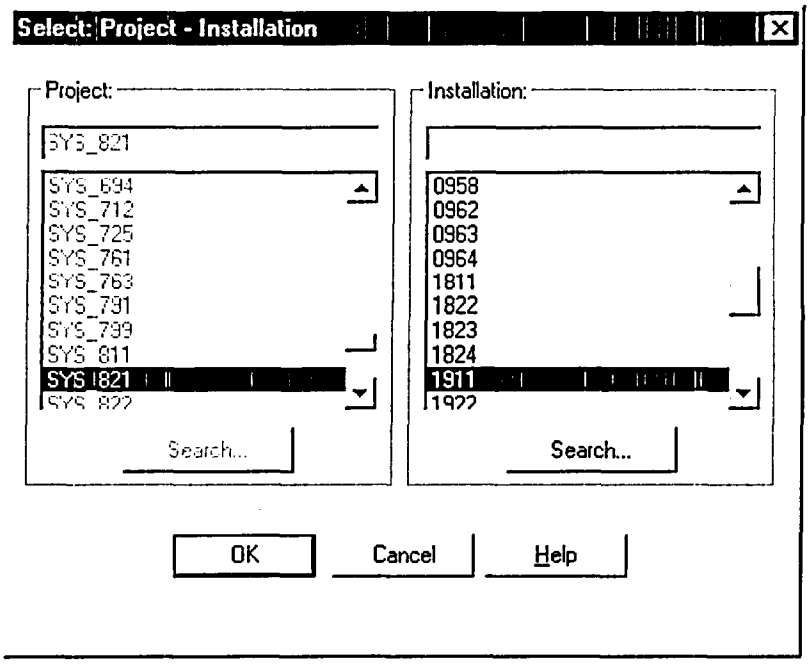
6. Click **OK** for this dialog box to continue.



7. Setup options in the dialog as shown. Location is filtered to "L1" in the **Connection** template.



8. Setup dialog is shown. Click **CANCEL.**



9. After step 8, the logic reports - ".txt" files will be generated in folder HFControls/OneStep/System##.

Section 5

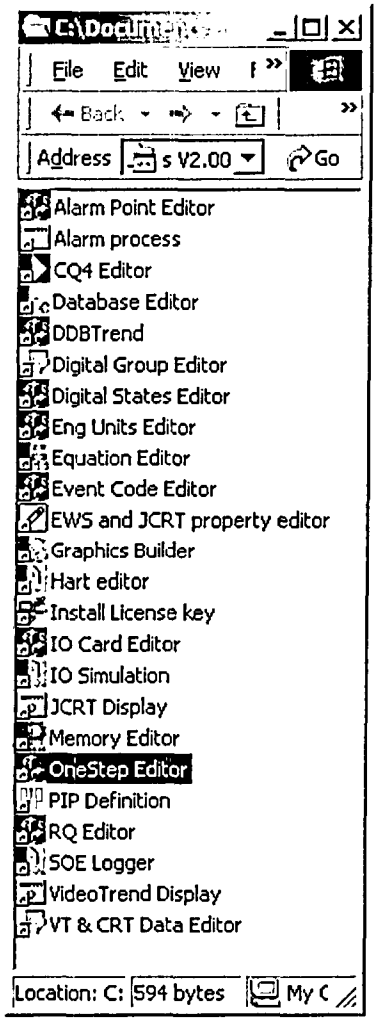
ONE-STEP CODE GENERATION AND CONVERSION

5.1 PRELIMINARY OPERATION

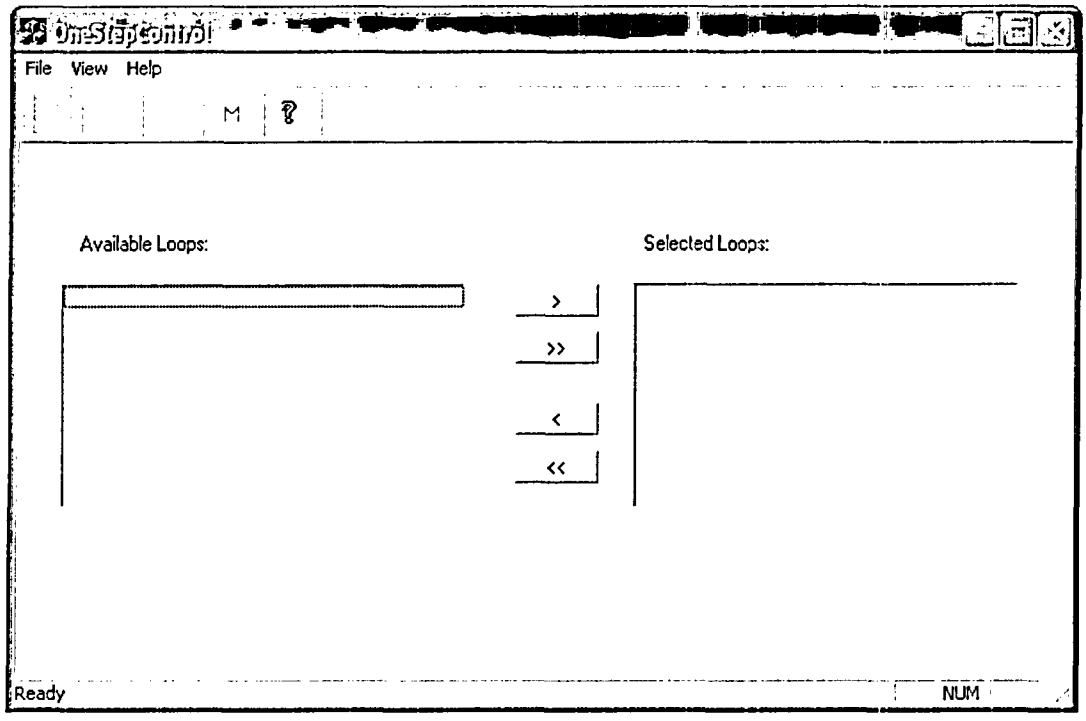
Generate **promis·e** reports first before start using One-Step Software to produce application codes.

5.2 START UP

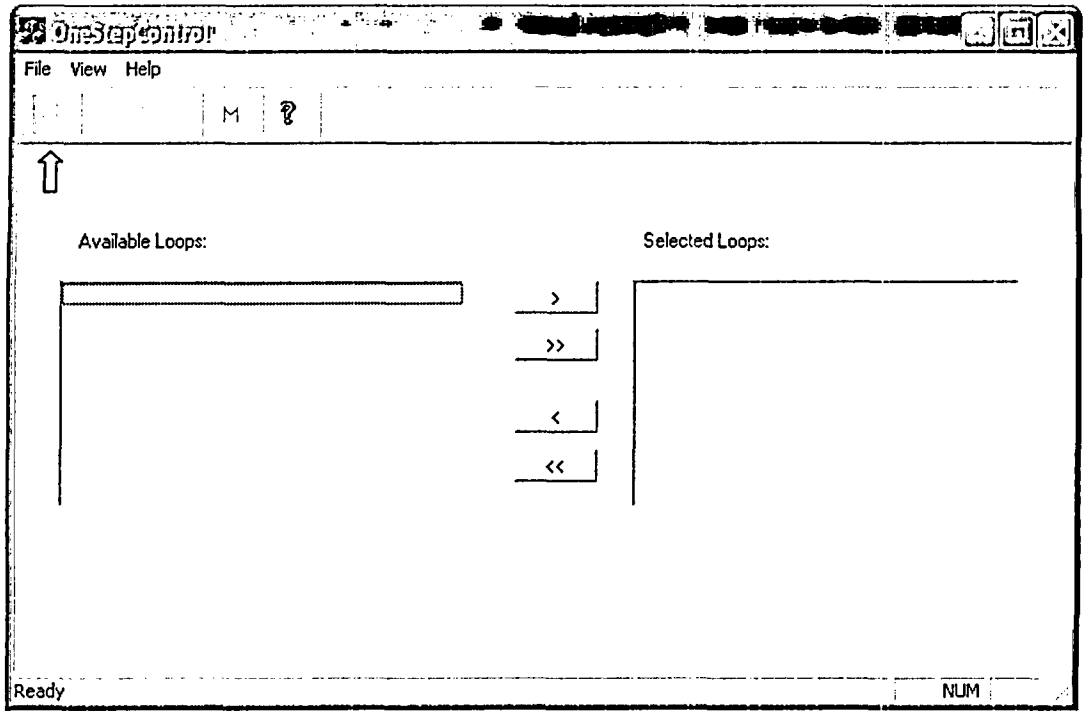
1. Start the One-Step program by double clicking on the **OneStep Editor** icon in the HFControls V2.50 folder as shown below.



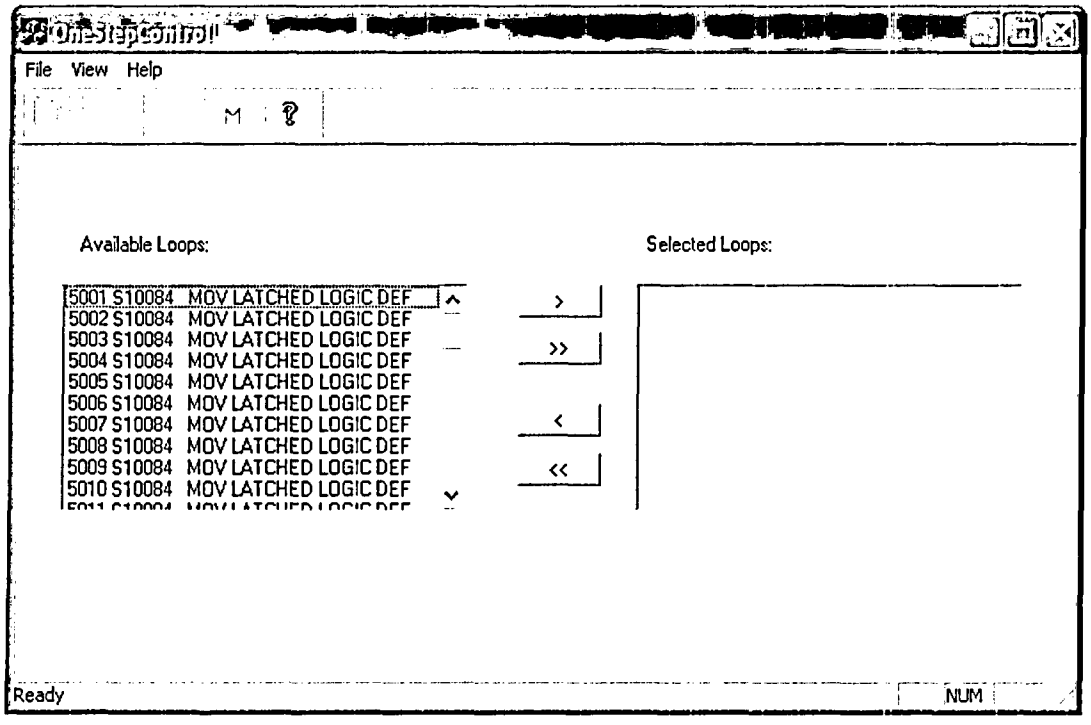
2. After the One-Step software is started, the following window will pop up. The first time **OneStep Editor** is started, no loops are available in the **Available Loops window.**



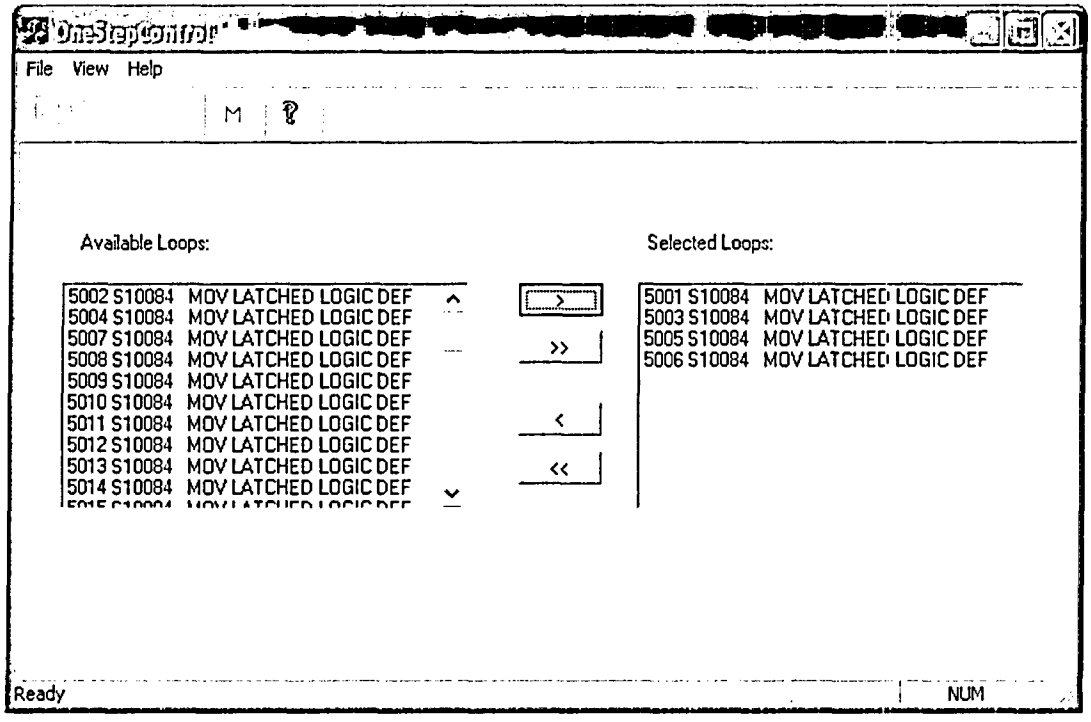
3. Click **File>Reload** or click the **LD** button on the menu bar to load loop numbers.



4. After loading loop numbers, all loop numbers for the current job will appear in the **Available Loops** window as shown below. The next time the One-Step Editor is started, all loop numbers that have been loaded previously will already be displayed in the **Available Loops window.**



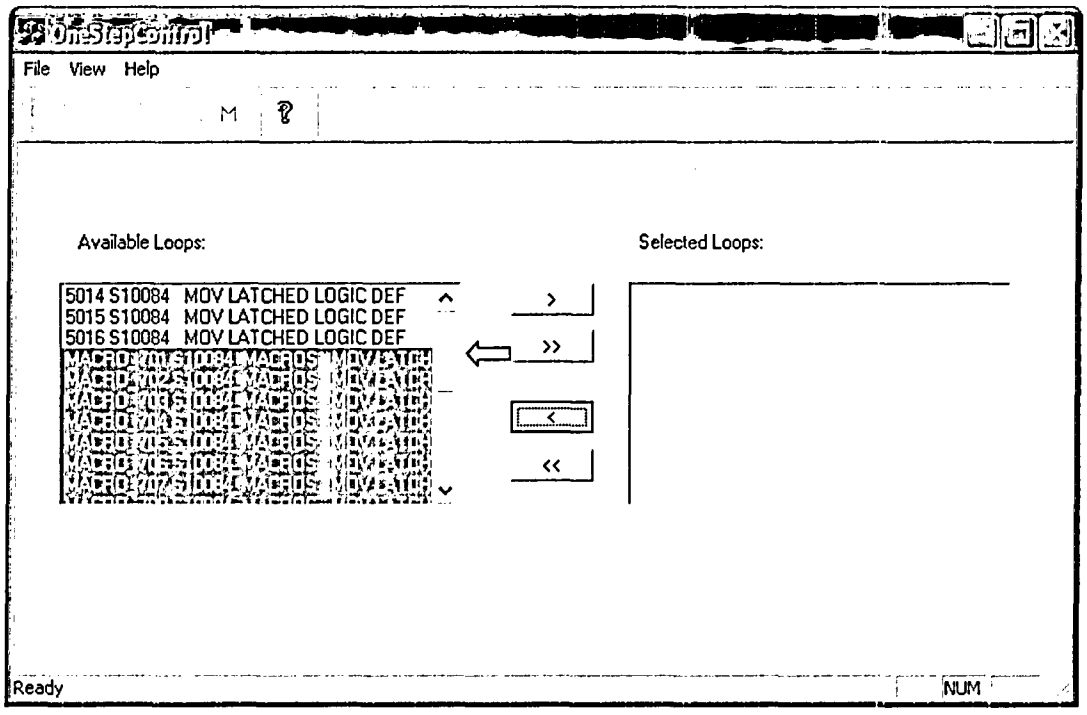
5. Selecting any entry in the **Available Loops** window and clicking the ">" button copies that entry to the **Selected Loops** window. Single loop and multiple loops can be selected and processed at one time. Clicking the ">>" button will send all of the loops to the **Selected Loops** window. Clicking the "<" or the "<<" button will deselect the selected loops or all loops in the **Selected Loops** window.



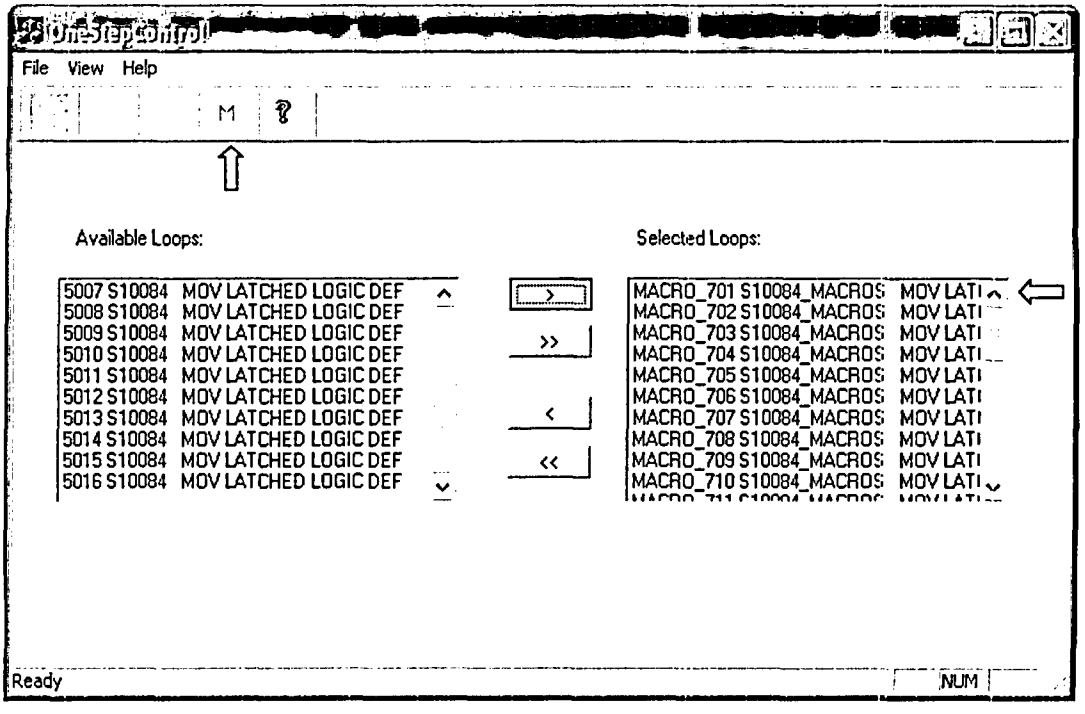
5.3 GENERATING MACRO CODE

If the application contains macro definitions, then the macros must be generated first before any application code can be generated. The following steps describe the sequence for generating macro code for one or more loop controllers or remote.

1. Select the macro to be built. Typically all macros will be selected.



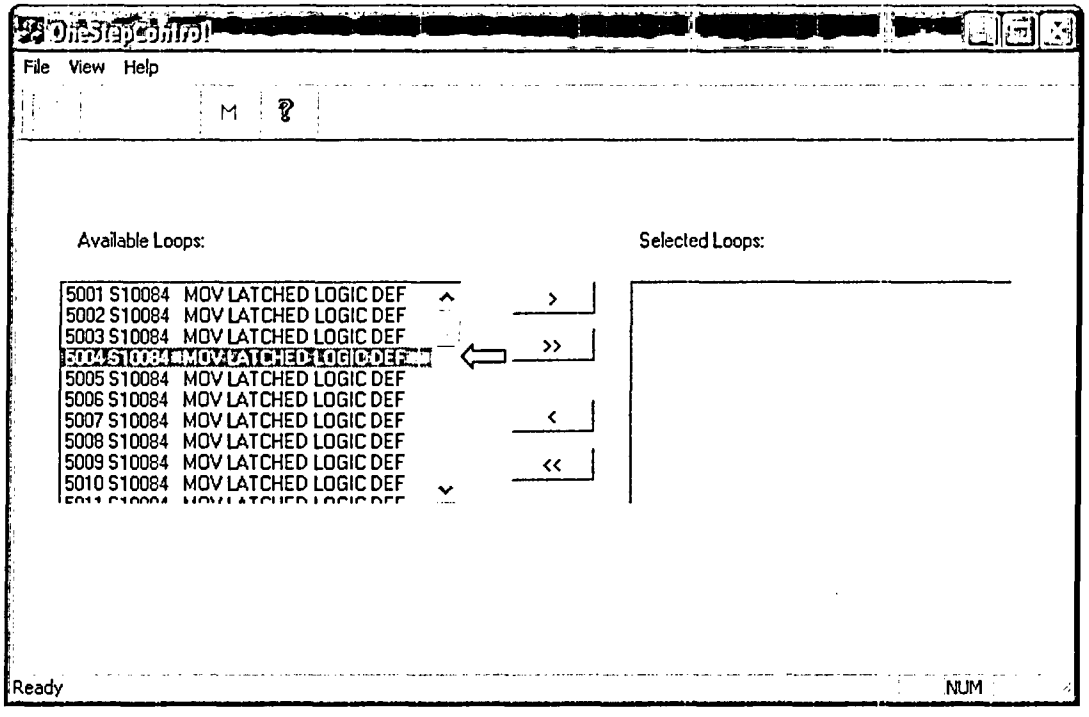
2. Use the right arrow button to move the macro(s) from the **Available Loops** window to the **Selected Loops** window. Click **File>Process macros** or the "M" button to start compiling and generating code for the selected macro(s).



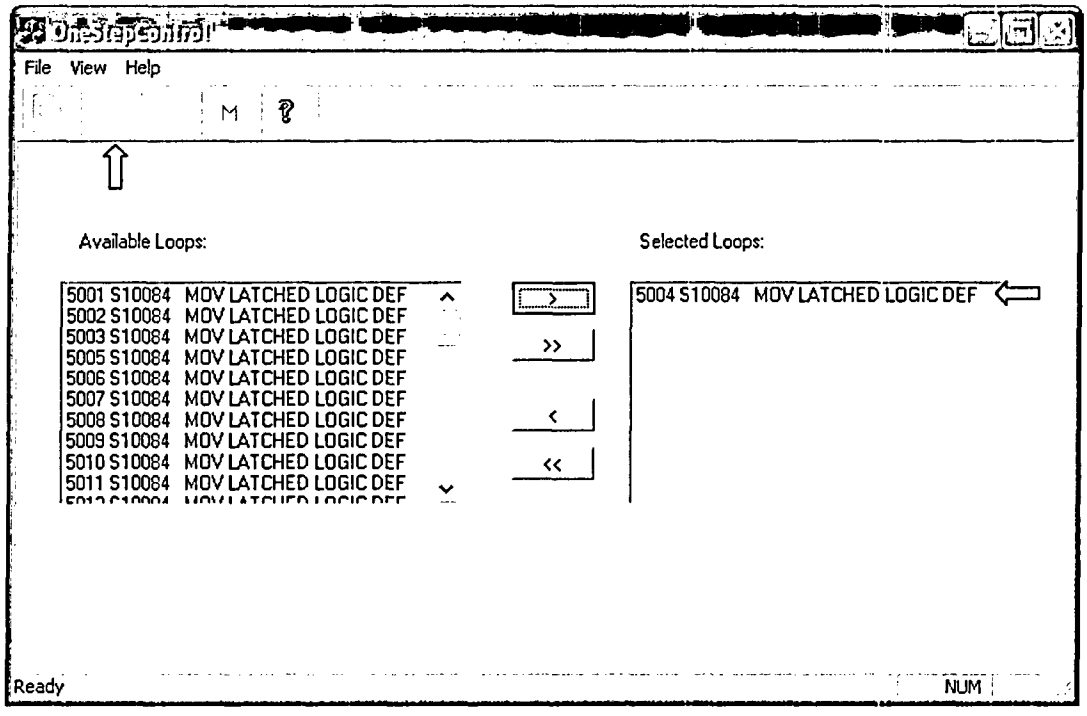
5.4 GENERATING APPLICATION CODE

The following steps describe the sequence for generating program code for one or more loop controllers or remote.

3. Select the loop number for the Application EPROM to be built. In this example, the loop 5004 has been selected.



4. Use the right arrow button to move the LOOP from the **Available Loops** window to the **Selected Loops** window. Click **File>Process Selected Loops** or the "**P**" button to start compiling and generating application codes for the selected loop(s).



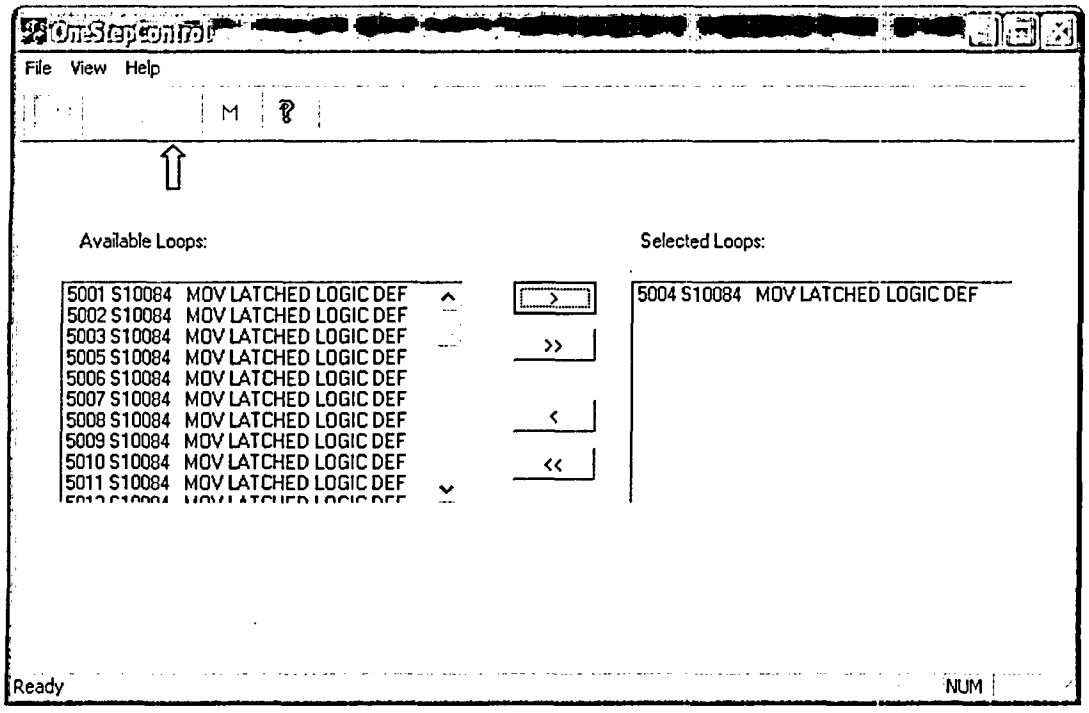
5. If processing is successful, go to output folders to look for the application code generated. Make sure no error message exists in the error files. The application code "**Apps####.prm**" can then be loaded into an EPROM for testing.

6. If any debug window pops up with error messages, use information in the debug window as well as error files in the output folders to debug. For example, errors in the original promis·e drawings can be a source of compiling failure. Refer to Appendix D for the error lists of the error files.

5.5 CONVERTING GRAPHIC FILES

The following steps describe the sequence required to generate ".gra" graphic files from a promis·e drawings file.

1. To convert CAD reports to graphic files, select the loop first, and then click on the button "**G**" in menu bar.



2. After the converting is done, go to check the WIN_JCRT/gra folder to make sure that the ".gra" files are there.

Section 6

WRITING PROGRAM CODE TO EPROM

6.1 INTRODUCTION

This section describes the general procedures of writing application code generated by One-Step software to EPROM. There are different brands of EPROM programmer devices or chip writer. Refer to the vendor's publication of the specific device for detailed instructions. This section uses Dataman EPROM programmer as an example.

6.2 START UP

1.[

]

2. [ ]
[

]

6.3 LOAD APPLICATION CODE

[

]

]

6.4 PROGRAM EPROM

[

⇦

]

6.5 INSTALL EPROM

[

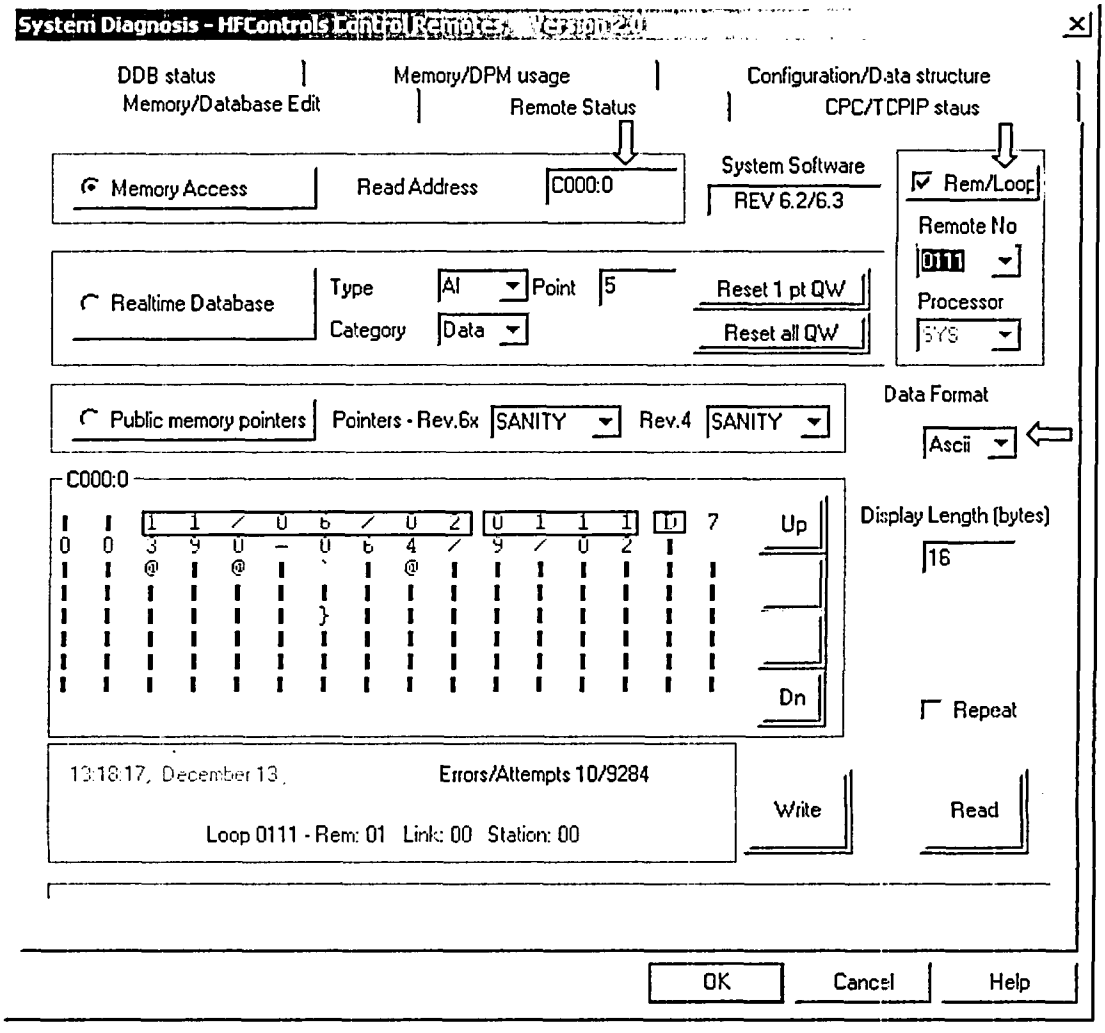
16.

.]

6.6

TEST EPROM

17. Start Memory Editor software, and check Rem/Loop in order to select a loop. After the loop numbers are loaded to the drop down menu, select the loop, for example "0111". Then enter the memory address for the Application file header, "C000:0" in the address box, select "Ascii" for display format, and click Read button. The application program will be displayed. Verify the application file date "11/06/02", the loop number "0111" and the revision number "D" in the file are correct.



18. Start MCRT software to test the correctness of the control logic in the new EPROM against schematic drawings. Refer to appropriate test procedures for detailed instructions.

[

]