

PLC Programming Guide

<u>CONTENTS:</u>	<u>PAGE:</u>
1.0 SCOPE	1
2.0 GENERAL INFORMATION	1
3.0 PLC ACCEPTANCE	2
4.0 PROGRAM CHANGES	2
5.0 SECURITY	3
6.0 SYMATE PROGRAMMING	4
7.0 CONCEPT PROGRAMMING	6
8.0 DFB EXAMPLES	19

1.0 SCOPE

It is the intent of this document to set forth a set of minimal standards that shall be used for programming Programmable Logic Controllers (PLCs).

The Wastewater Group (WG) has standardized on new projects using Quantum PLC'S. Communication is done through Ethernet ports, are used at Tryon Creek and Columbia Treatment plants. There are other locations where PLCs are networked. At some locations where stand-alone PLCs are used, such as small pump stations Momentum PLCs have become the new standard. Existing locations use the series 400 or 600 PLCs, without Ethernet ports are permitted for use. Using PLCs without Ethernet ports must be approved by WG's Automation Planning Team (APT).

2.0 GENERAL

The current standard for PLC Programming is Concept 2.5 Service Release 2, D Patch. This software will be used on all new projects using Quantum or Momentum PLCs.

SquareD® "SymatePlus" shall be used to write all PLC programs on existing Square D PLCs. We have existing SquareD PLCs at various pump stations, Columbia BLVD. WWTP and Tryon Creek WWTP.

PLC programs are to be written in a simple and straightforward manner. Programs for Quantum PLCs will be written using AEC standard function

PLC Programming Guide

blocks. Appropriate DFBs shall be used. The programmer will review the existing DFBs that are available upon request. Examples follow this standards section. The programmer will then use these standard DFBs along with specific DFBs developed for this project. The specific DFBs will be saved with the program. Where appropriate, sequential function blocks and IEC ladder logic are also acceptable. The use of 984 ladder logic is not approved.

It is important that maintenance personnel with little or no familiarity with the program be able to analyze and understand the program in a minimal amount of time.

A flow chart of the PLC program shall be given to the APT for review at the beginning of program design. While in the development stage, the program shall be periodically submitted to APT for review and approval.

3.0 PLC ACCEPTANCE

Current, good copies of the PLC program files shall be given to the WG Electrical and Instrumentation Department (E&I) when the PLC becomes operational.

Testing: At a minimum the following tests shall be performed. There may be additional tests that the contractor, consultant or the PLC programmer may deem necessary. Testing must be done to ensure that the control logic of the entire system functions properly and performs its intended purpose.

- ❖ The PLC inputs shall be tested by individually activating each and all field I/O devices and confirming that the assigned PLC address receives the input.
- ❖ Each PLC output shall be individually activated and the associated field device shall be checked to ensure that it receives the correct PLC signal.
- ❖ All analog I/O shall be properly calibrated.
- ❖ All route statements shall be tested to ensure they are correct and that they function properly.
- ❖ All communications wiring shall be tested for proper functioning. For further details on data communications see that section of the E&I Standards.

4.0 PROGRAM CHANGES

After the WG accepts operational responsibility for a facility that has an associated PLC, any changes to the PLC **must** be coordinated with the

PLC Programming Guide

E & I Supervisor. **Also, after any changes are made to a PLC, the most recent copy of the program must be given to the E/I department with a copy of the previous program files (in case we need to reload the old program).**

Note: SymatePlus software will normally automatically create this file for you by changing the last letter of the file extension to a tilde (~). E & I keeps several copies of these files for documentation as well as for use in operation and maintenance. These files **must be kept current**. We have experienced considerable trouble in the past when this has not been done.

When PLC program changes are made, the date, name of programmer and a brief description of the change shall be entered in the title section of the PLC program.

The latest version of a program will be stored on the Server Romeo on the PLC Share. Two copies will also exist on floppy disk and stored in the E/I Shop. These copies will be designated as MASTER and SHOP copies. The SHOP copy will ordinary be used in the field to initiate program changes

5.0 SECURITY

PLC Security will be maintained by use of the memory protect key function. All programming changes shall be downloaded locally unless special arrangements are made. When possible program software passwords will also be initiated.

PLC Programming Guide

6.0 PROGRAMMING GUIDELINES FOR SYMATE PLUS

A 1 second scan delay shall be used at the beginning of the program to allow the PLC to boot up.

Any constant values set in registers shall be done in a subroutine at the beginning of the program. A separate register file shall not be used to store and/or load these values.

```
DEMO/LADDER
| FIRST                                1 SCAN
|-----| |-----()--
| SCAN                                DELAY
|
| FIRST
|-----| |-----()--
|                                SCAN
|
| 1 SCAN
|-----| |-----TLET S0150= 04000; 04800; 09999; 02000
| DELAY
|
| 1 SCAN
|-----| |-----TLET S0158= 04500; 04500; 00000; 05600
| DELAY
```

6.1 SUBROUTINES

PLC programs are to use subroutines where possible.

The function of the subroutine and a brief narrative of its logic shall be written in a label header.

If the rung is part of a subroutine, the first characters of the header shall be “Subroutine XX” where XX is the subroutine number. This is to allow you to easily know which subroutine your are viewing when looking at a PLC program with a PC instead of a hard copy listing.

Alarms shall be grouped in 1 subroutine.

PLC Programming Guide

6.2 LABELS

Symate Plus allows 3, 6-character fields to be used for labels, with only the first 2 displayed on a personal computer (PC) screen. Therefore the programmer shall use abbreviations which describe the contacts, etc., as descriptively as possible.

Abbreviations listed at the end of this document shall be used in applicable PLC labels. If labels are used that are not on this list, a list shall be provided to the WG so that it can be incorporated in the abbreviation list.

If the label is describing an analog loop, the first 6 characters **shall be the loop number.**

Coils and normally open (NO) contact “descriptions” shall be written so as to be True when the coil is energized. Normally closed (NC) contact descriptions shall be written so as to be False when the associated coil is energized.

Each rung of ladder logic shall have a Header label explaining the logic of the rung. (This prevents headers and rungs from being misaligned when ladder logic is revised). If the rung is part of a subroutine, the first characters of the header shall be “Subroutine XX” where XX is the subroutine number.

PLC Programming Guide

7.0 PROGRAMMING GUIDELINES FOR CONCEPT

7.1 INTRODUCTION

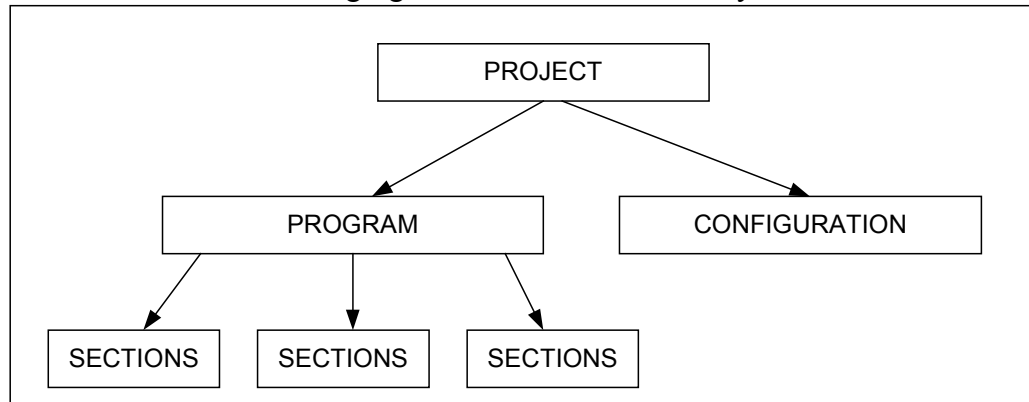
OBJECTIVE

The objective of the PLC Programming Guidelines is to develop the minimum set of standards for PLC programming.

OVERVIEW

A. General:

1.0 Programming of the PLC using Concept is done hierarchically, in projects, in the PLC configuration, in programs, and in Sections. The following figure illustrates the hierarchy.



2.0 A project represents the function of a specific PLC. The program consists of one or more sections that describe the program of the PLC.

3.0 Each PLC project shall be designed around a structure that breaks the PLC program into logical functional sections.

7.2 - DETAILS

PROJECTS

A. Files and File Naming

1. From a file standpoint, a project consist of a series of files all with the same core name but different extensions (e.g. PLC Name.). Each project is named to correspond to the PLC to which it is downloaded.
2. Project names are limited to 8 characters by Concept.
3. PLC program file names will follow the following identified below. Note the last two digits accommodate versions.
 - a. FACLxxVZ, where:
 - 1) FACL= Facility/contract identifier, four character (e.g. HIDC)

PLC Programming Guide

- 2) xx = PLC number within a facility (e.g. 01..99 or –1..99, the later is preferred)
 - 3) V = Program version character (e.g. A..Z)
 - 4) Z = Program subversion number (e.g. 0..9)
 - 5) Completed example: HIDC-1A3.¹
4. Projects can also consist of local Definable Function Blocks (or DFB's), as opposed to global DFB's. For DFB's to be local only to one project, they need to reside in a DFB directory under the directory where the primary project files (PLC¹) reside. For this reason, it is suggested that each project reside in its own directory.
 5. Note that global DFB's are located under the ...\\Concept\\DFB directory. Refer to Definable Function Block document.

B. PLC Names

1. PLC names will follow almost the same format as for the program file names, with the last two version control characters dropped (e.g. simply HIDC-1).

C. Version Control

1. The most recent copy of the program (change) will be given to AST on floppy disk or CD ROM.
2. AST will archive the program on the PLC share/ROMEO server. A master copy will also be provided to the E/I shop.

7.3 CONFIGURATION

A. Suggested PLC I/O Rack Layout

1. To maintain maximum separation between signal types, I/O in racks will be laid out with analog I/O on left (starting next to communication or CPU modules), and discrete I/O on right (starting next to redundant power supply module). Both types shall fill to center.
2. Rack (drops) can support a maximum of 64 words In and 64 words Out. Note: 64 words in and 64 words out **can** be exceeded if an entire 16 slot rack is filled with high density analog modules, or all 8 point analog inputs modules.

¹ PLC Name

PLC Programming Guide

3. The following shows a typical rack layout for a Modicon 16-slot rack containing a PLC

C P S	C P U	C R P	A C C O	A C C O	A C C O	D A I	D A I	D A O	D A I	D A I	D A I	D A O	C P S
1	4	9	0	0	0	5	5	8	5	5	5	8	1
2	2	3	3	2	3	4	4	4	4	4	4	4	2
4	4	2	0	0	0	0	0	0	0	0	0	0	4

4. The following shows a typical rack layout for a Modicon 16-slot rack as remote I/O.

C P S	C R A I	A C C O	A C C O		D A I	D A I	D A O	D A I	D A I	D A O	C P S
1	9	0	0		5	5	8	5	5	8	1
2	3	3	2		4	4	4	4	4	4	2
4	2	0	0		0	0	0	0	0	0	4

B. Memory Usage

1. Available memory address types and defaults (for a Modicon 424 processor) are as listed below. These default ranges may need to change depending on size of PLC system. Note defaults require about 3% of the 64K state memory.
 - a. Coil (0:x) (default 1-1536)
 - b. Discrete Inputs (1:x) (default 1-512)
 - c. Input Resisters (3:x) (default 1-48)
 - d. Holding Registers (4:x) (default 1-1872)
 - e. Xxx (6:x)
2. The following criteria was used to establish a guideline for assigning PLC memory to I/O.
 - a. Analog cards will not exceed 8 points per card (requiring a maximum of 8 register addresses)
 - b. Discrete cards will not exceed 32 points per card (requiring 32 coil or discrete input addresses).
 - c. When adding a card, addresses are available in same memory area as other I/O, and in a sequential order. Gaps in memory are allowed
 - d. If the rack, slot, point, and I/O card type are known, the memory address can be directly determined (a table is available as a supplement)
 - e. A maximum of 14 slots are available for I/O, since power supply takes one slot (two for redundancy) and processor or communication module will take another.

PLC Programming Guide

- f. Assign enough bits or words to accommodate all of one type of module in a rack.
 - 1) 14 slots available for I/O x 32 bits/slot = 448 bits maximum per rack (for DI's or DO's)
 - 2) 14 slots available for I/O x 8 words/slot = 112 words maximum per rack (for all AI's), but drops limited to maximum of 64 words in and 64 words out. Assign maximum supported words in and out to each rack (64/64), regardless of whether needed or not.
 - g. Both analog register assignment and discrete coil and inputs will increase going from left to right.
 - h. Before the assignment of addresses for I/O, leave a block of memory address available for system level status monitoring (e.g. PLC health, system clock, etc.)
3. The following table shows the overall assignment scheme. Refer to supplement for a typical detailed breakdown, including slot and point.

Drop / Rack	O:x (Read/Write) Discrete Outputs 448 bits/drop	I:x (Read only) Discrete Inputs 448 bits/drop	3:x (Read only) Analog inputs 64 words/drop	4:x (Read/Write) Analog Outputs 64 words/drop
NA (system) – Global Data – Hydra Data – PLC Status – RIO/DIO Status?	0: 1-128	1: 1-128	3: 1-10	4: 1-110 1 – 32 (Global) 33 – 52 (Hydra) 53 – ... (PLC stat, RIO/DIO stat)
1	0: 129-576	1: 129-576	3: 11-74	4: 111-174
2	0: 577-1024	1: 577-1024	3: 75-138	4: 175-238
3	0: 1025-1472	1: 1025-1472	3: 139-202	4: 239-302
4	0: 1473-1920	1: 1473-1920	3: 203-266	4: 303-366
5	0: 1921-2368	1: 1921-2368	3: 267-330	4: 367-430
6	0: 2369-2816	1: 2369-2816	3: 331-394	4: 431-494
7	0: 2817-3264	1: 2817-3264	3: 395-458	4: 495-558
8	0: 3265-3712	1: 3265-3712	3: 459-522	4: 559-622

C. Loadables Required

1. The following loadables are required in the 140 CPU 424 0x Modicon Quantum processor to fully support the IEC programming functions utilized:
 - a. @1S7 V196
 - b. @217 V196

PLC Programming Guide

D. Special Configuration

1. Battery Coil
2. Timer Register
3. Time of Day
4. Allow Duplicate Coils
5. Watchdog Timeout
6. Redundant
7. I/O Timeslice

E. Segment Scheduler

F. Configuration Extensions

G. Data Protection

H. Peer Cop

I. Hot Standby

J. Ethernet

K. ASCII Setup

1. Total Messages
2. Message Area Size
3. ASCII Ports

L. Section Execution Order

1. The order that sections are executed by the PLC can optionally be specified by the user. A suggested section execution order is as follows:
 - a. System Sections:
 - 1) Administrative (one time)/Analog Configuration
 - 2) Administrative/Communication/Timing Section
 - 3) I/O Processing Section
 - b. Unit Process Sections:
 - 1) Unit Process #1 Section
 - 2) Unit Process #x Section

7.4 PROGRAM AND SECTIONS

A. Usage

1. A program is simply the collection of all sections within the project. Sections will contain logic that is both custom to the process (e.g.

PLC Programming Guide

sequence logic) as well as Derived Function Block's (DFB's) which are reusable blocks of code (other section and programs can typically reference these block)

2. Some primary consideration regarding the use of sections is as following:
 - a. The section is the smallest program element that can be loaded in the PLC at one time.
 - b. A section can only contain a single programming language, IL, LD, etc.
 - c. Sections listed in Concept are alphabetically arranged by section type (FBD, LD, etc.) and then by section name.
 - d. Depending on the size of the section, inserting new rungs into an existing section can be challenging due to the free form logic layout nature of Concept. However, the use of Derived Function Blocks will significantly reduce the physical layout space required.
3. Due to the different needs of each program there can be no hard and fast rules used for program structure. Taking into account the considerations above, the following guidelines shall be used in designing a program.
 - a. Each Section shall be structured in a logical way that takes a unit process or unit operation and breaks it into manageable modules within the section.
 - b. Modules within a section shall correspond to Program Flow Diagram (PFD) developed for a Unit Process (see PROCESS SECTIONS discussion below).
 - c. The subsections shall develop a complete operation such as developing process switches for a pump station and pump alternator, pump control, a level controller, etc.
 - d. The first regularly scanned section of a program shall be program timing section.

B. Section Naming

1. Section names are limited to 32 characters and cannot contain any spaces.
2. Section name should typically be the process description, with word separated by an underbar (e.g. Odor_Control).
3. If more than one section is required for a particular process, identify the main section as _____ and the other related sections as _____, so as to keep the sections listed together. (e.g. WW_Clarifier_Sequence_Control as the main section and WW_Clarifier_1, WW_Clarifier_2, etc. as the related sections).

C. Documenting sections

PLC Programming Guide

1. Since Concept at this time does not allow searching for text notes within the program, but does allow searching for variables, each code subsection shall start with a dummy piece of program logic that is linked to a Boolean variable. Descriptive text shall be placed close to this variable.
2. Each section shall be identified with a title that describes the sections overall function.
3. Each *subsection* shall be identified with a title that describes its function.
4. The PFD shall note each subsection and describe the logical operations of each.
5. Within a section, the objective of PLC program documentation is to convey the operation of a specific rung or group of rungs to the O&M staff for system startup and long term maintenance. The documentation criteria is somewhat subjective but should be developed such that a person with PLC programming skill should be able understand the program in reasonable time. The following are minimum requirements.
 - a. As a general rule each program rung will have a description that describes the function of the rung and how it works in the overall program..
 - b. Loop Number(s) of controlled devices.
 - c. Equipment description. i.e. RAS Pump #1 Equipment Control

D. Reference Data Editor

1. Useful for troubleshooting, the Reference Data Editor allows online monitoring and manipulation of variables within the processor, in a spreadsheet format..
2. Each unit process program section (or sections) should have at least one Reference Data Editor file (*.rde) created containing critical variables.
3. Reference file should have the same name, or an abbreviated form, of the unit process section it corresponds to.

E. Sections Classified

1. For purposes of discussion only, sections will be classified as one of two types, discussed in more detail later in this document:
 - a. System Sections: those section related to administrative details, common to all programs
 - b. Process Section: those sections that have been uniquely developed to meet different process needs

7.5 SYSTEMS SECTIONS

A. PLC Status monitoring

PLC Programming Guide

1. Each PLC shall be monitored using the PLCSTAT (PLC health Status Function Block)
2. The objective of the PLC monitoring is to alert the O&M staff that there is a problem in the PLC or it I/O subsystem. The alarm and status points will be monitored from the SCS.
3. The following lists the specific alarms and status points to be monitored by the SCS

a. Word 1 (PLC status)

Bit	Description
9	Alternating Current On
12	Battery failed

b. Word 2 (Hot Standby status if the system has a hot backup)

Bit	Description
15&16	01 = Local system Offline 10 = Local system Primary 11 = Local system Standby
13&14	01 = Remote system Offline 10 = Remote system Primary 11 = Remote system Standby
11	0 = PLCs have Matching Logic PLCs do not have matching logic

c. Word 3 (PLC status)

Bit	Description
3	First Sweep
12	Battery failed

1. The first sweep may be used for program initialization.

B. Analog Configuration

1. To take advantage of the built in status monitoring blocks provided with Concept, analog card configuration will typically be handled by interconnecting the Quantum function blocks (reference Concept on-line help for details)

PLC Programming Guide

C. Program timing

1. Each PLC program shall have a DFB timing routine that cycles a set of timing outputs to be used in the program. Use of the function block SYSCLOCK will facilitate this.
2. SYSCLOCK
 - a. This function block generates clock pulses of frequencies 0.3125 Hz, 0.6250 Hz, 1.2500 Hz, 2.5000 Hz and 5.0000 Hz. Additionally, the time duration since system cold start is provided.
 - b. Parameter Description
 1. CLK1 BOOL Clock freq. 0.3125 Hz (3.2 s clock)
 2. CLK2 BOOL Clock freq. 0.6250 Hz (1.6 s clock)
 3. CLK3 BOOL Clock freq. 1.2500 Hz (800 ms clock)
 4. CLK4 BOOL Clock freq. 2.5000 Hz (400 ms clock)
 5. CLK5 BOOL Clock freq. 5.0000 Hz (200 ms clock)

D. Communications

1. A single section shall be used to set up any additional peer-to-peer PLC communications outside of the Configure/Peer-Cop environment
2. Typical function blocks used in this section are:
 - a. CREADREG: Continuous Register Reading
 - b. CWRITREG: Continuous Register Write
 - c. MBP_MSTR: Modbus Plus Master
 - d. READREG: Read Register
 - e. WRITEREG: Write Register

E. I/O Processing

1. Analog
 - a. All analog points input into a PLC shall be scaled to engineering units with the ANLG_IN DFB. The scaled value shall be used for communication to the SCS and PLC based control algorithms.
 - b. Process of analog input process shall be distributed using the timing routines to minimize the PLC loading on any one scan time.
 - c. Each analog loop will monitor for analog input signal fail.
2. Discrete
 - a. Discrete I/O modules will normally use isolated inputs. Grouped inputs will only be used under special conditions.
 - b. Discrete AC output modules will have AC voltage supplied from a common, single phase AC power source. I/O modules vary greatly in their application. Please follow the Quantum Hardware Reference Guide for detailed information.

PLC Programming Guide

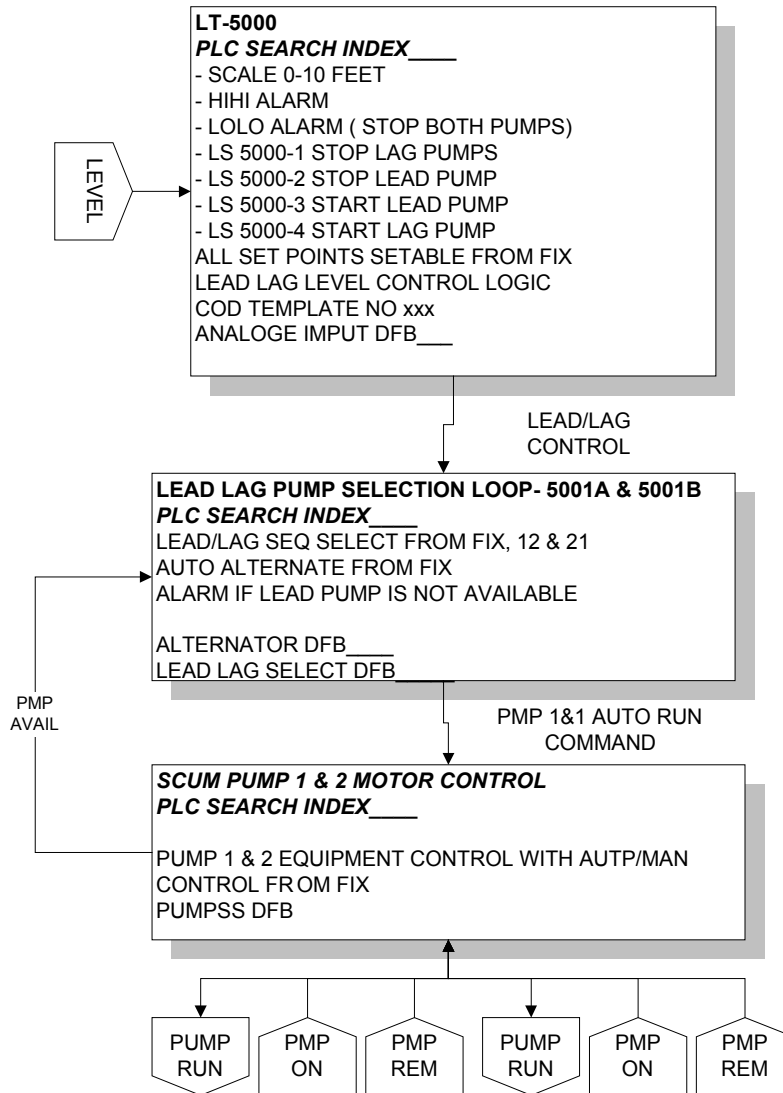
7.6 PROCESS SECTIONS

A. Program Flow Diagram (PFD)

1. The objectives of the Program Flow Diagram (or PFD) is to develop the structure and organization of the PLC program. In addition, the PFD's will aid the O&M staff in analyzing and troubleshooting the program.
2. The functional block diagrams shall illustrate the PLC program in a flow chart format where each program section and subsection is identified. Each subsection shall be shown as a flow chart block. The PFD shall show interrelationships between blocks. The interrelationships shall include functions such as level switches developed in one block (subsection) and passed to another block.
3. The following describes the content for each block (subsection):
 - a. Subsection Title
 - b. Search index variable.
 - c. Operational description
 - d. Mathematical equations, set point ramps, PID loop, loop initialization, etc.
 - e. COD templates used
 - f. Defined Function Blocks (DFB's) used.
 - g. PLC physical I/O points
4. The PFD's shall be developed in VISIO flow charter. The following is a typical PFD:

PLC Programming Guide

SECTION- SCUMPMP
SCUM SUMP LEVEL CONTROL
LOOP 5000 & 5002A AND 5000B



7.7 Miscellaneous

PLC / SCS INTERFACE COILS AND REGISTERS

A. Point Attributes

1. The coils and registers that the SCS reads and writes to is developed by COD² (configuration offline database), based on the **Point Attribute**. The point attribute defines a class of data and ultimately it's update rate.

²Developed by CH2mHill

PLC Programming Guide

2. The following table lists the point attributes defined:

Point Attribute	Description	Memory Range
DR	Discrete Read of the SCS from the PLC. Updated on a 1 second rate.	
DW	Discrete write from the SCS to the PLC	
FPR	Floating Point Read of the SCS from the PLC. Updated on a 2 second poll rate.	
FPRS	Floating Point Read of the SCS from the PLC, on a slow poll. Updated on a 5 second poll rate.	
FPW	Floating Point Write from the SCS to the PLC	
IR	Integer Read of the SCS from the PLC. Updated on a 2 second poll rate.	
IRS	Integer Read of the SCS from the PLC, on a slow poll, for counters, runtimes, and totalizers. Updated on a 10 to 30 second poll rate.	
IW	Integer write from the SCS to the PLC	

7.8 VARIABLE NAMING USAGE AND NAMING CONVENTION

A. Variable not interfaced to SCS

1. For variables not read by the SCS, the PLC programs shall use defined variables rather than direct memory address. The variable shall follow a similar format as the SCS tag. The following outlines the variable naming criteria.
 - a. Loop
 - b. Description
 - c. Extension

7.9 PLC PROGRAMMING CONSIDERATIONS FOR USE WITH iFix

WG uses iFix® (Intellution Inc. – Foxboro Mass.) as the only operator interface to our PLCs. Several factors must be considered in relation to iFix.

Registers of like data-type acquired by iFix shall be grouped together to minimize the number polls or data requests.

The programmer is to determine all pertinent I/O points and registers that need to be used by iFix to present a complete picture of the process being programmed.

The PLC programmer shall provide all PLC data needed to complete the iFix development. We use a PLC / iFix development program provided by CH2mHill, Inc. called COD (configuration offline database) to develop import files for PLC and iFix products. Our license allows only contractors

PLC Programming Guide

working on PLC and iFix programming (for the City of Portland) to use this product. If you are not using COD to provide data for the iFix database, you must provide the following information in an Excel97® spreadsheet:

- PLC register address.
- Tag name.
- Data type.
- Range (for analog registers).
- The description of how the register is used.
- PLC Rack, Slot and Point(s) for that register.
- The engineering units for analog data in the register and the engineering unit label (GPM, PSI, pH, etc.).
- The “True” state (open or closed) for discreet points.
- For all alarms, provide a description of the alarm and the conditions that cause the alarm. Include the analog levels that set and reset the alarm.

7.9.1 LABELS USED IN PLC PROGRAMING

The following abbreviations shall be used in applicable PLC labels:

- ackn - acknowledge
- alarm
- amp
- c-t-run - call to run **ca2run**
- cmpare
- cnflct - conflict
- flshr - flasher
- hi
- horn
- intrlk - interlock
- lag
- laglag
- latch
- lead
- li - level indicator
- lo
- lt - level transmitter
- lvl - level
- mtr - motor
- plc
- plc/ld - plc lead

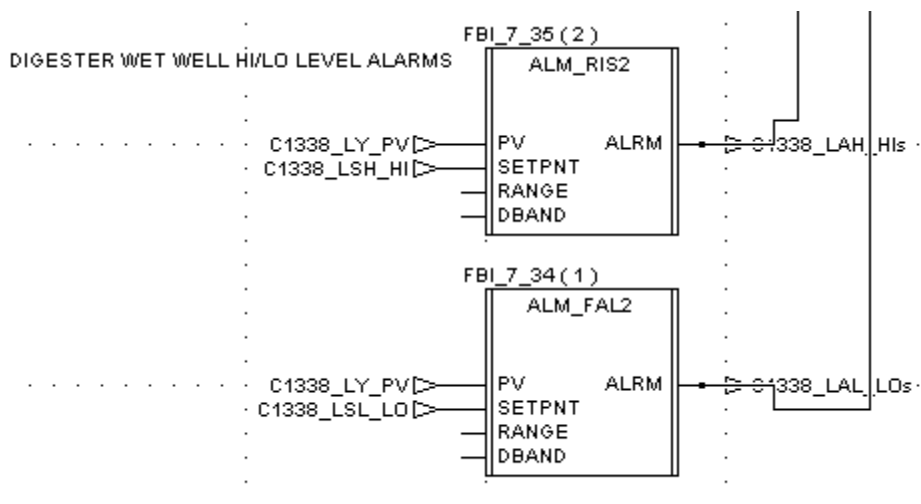
PLC Programming Guide

- pmp - pump
- purg - purge
- run
- sec - second
- seq
- shtdwn
- stashn - station
- w/w
- workng - working
- ww/lvl - wet well level

8.0 DFB EXAMPLES

BES SCS/PLC Standard: (ALRM_RIS2) Single Variable Rising Alarm w/ Dead band (below SP)

Date: March 7, 2000	Lead: Jane Doe	Status: PreFinal
Description:	Single Variable Rising Alarm w/ Dead band	
Comments:	Alarms when a process variable rises above the set point. RANGE times DBAND calculates Hysteresis before alarm will clear.	
Version:	Developed: 06/06/2001 10:41:26	
Accepted by:		
Accepted by:		



PLC Programming Guide

Function Description:

1. To prevent nuisance alarms, provide an (adjustable) deadband below a single process variable. Used as both a standalone function block as well as a sub function block in the ANLG_IN3 DFB.

Inputs:

Input	Description	Data Type
PV	Process Variable	REAL
SETPNT	Alarm set point	REAL
RANGE	Range of process variable	REAL
DBAND	Dead band in real (e.g. 0.02 = 2%)	REAL

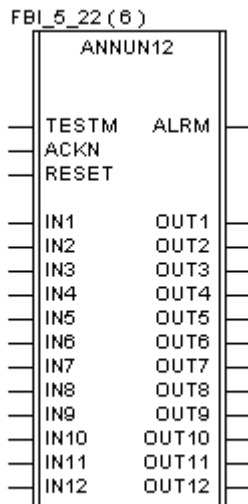
Outputs:

Output	Description	Data Type
ALRM	Process variable alarm	BOOL

BES SCS/PLC Standard: ANNUN12 Annunciator w/ 12 alarm points

Date: January 7, 2000	Lead: Jack Smith	Status: PreFinal
Description:	Modified ISA Sequence R annunciator	
Comments:	Designed for use with a Light box; 12 alarm points with single audible alarm	
Version:	Version: 5/26/99 6:09 AM	
Accepted by:		
Accepted by:		

PLC Programming Guide



Function Description:

1. This DFB is designed as a Modified ISA Sequence R annunciator.
2. Uses three control switches, “Acknowledge”, “Test” and “Reset”.
3. It has 12 inputs and 12 visual alarm outputs and one common audible alarm output.
4. The visual alarm fast flashes and the audible alarm sounds when an alarm is present. If the alarm condition goes away, the visual alarm slow flashes and the audible alarm cancels.
5. Acknowledged: visual alarm, steady light; audible alarm, silent.
6. After ACKN; if the alarm is not present, then the visual alarm light is out.
7. TESTM true: Alarm fast flashes but does not seal in: This is normal when the test is no longer true.
8. If using multiple blocks for a large light box with a single audible, then OR the audible alarms together outside the DFBs.

Inputs:

Input	Description	Data Type
TESTM	Test system (while maintained true)	BOOL
ACKN	Acknowledge alarms	BOOL
RESET	Reset system	BOOL
IN1	Alarm input 1	BOOL
IN2	Alarm input 2	BOOL
IN3	Alarm input 3	BOOL

PLC Programming Guide

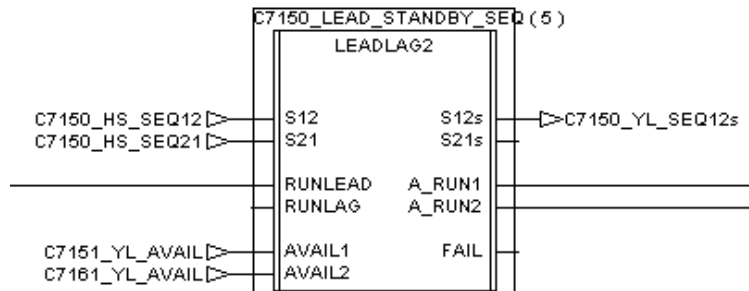
IN4	Alarm input 4	BOOL
IN5	Alarm input 5	BOOL
IN6	Alarm input 6	BOOL
IN7	Alarm input 7	BOOL
IN8	Alarm input 8	BOOL
IN9	Alarm input 9	BOOL
IN10	Alarm input 10	BOOL
IN11	Alarm input 11	BOOL
IN12	Alarm input 12	BOOL

Outputs:

Output	Description	Data Type
ALRM	Audible alarm	BOOL
OUT1	Alarm light 1	BOOL
OUT2	Alarm light 2	BOOL
OUT3	Alarm light 3	BOOL
OUT4	Alarm light 4	BOOL
OUT5	Alarm light 5	BOOL
OUT6	Alarm light 6	BOOL
OUT7	Alarm light 7	BOOL
OUT8	Alarm light 8	BOOL
OUT9	Alarm light 9	BOOL
OUT10	Alarm light 10	BOOL
OUT11	Alarm light 11	BOOL
OUT12	Alarm light 12	BOOL

BES SCS/PLC Standard: LEADLAG2 Lead/Lag or Lead/Standby Control

Date: January 7, 2000	Lead: Rick Komraus	Status: PreFinal
Description:	Lead/Lag or Lead/Standby Control	
Comments:	Allows 2 pumps to run either lead/ lag as called to run; or 2 nd pump can be a standby replacement for lead pump called to run but not responding.	
Version:	Version: 01/19/00 7:21:22	
Accepted by:		
Accepted by:		



Function Description:

1. This DFB provides lead/lag or lead/standby control over two pumps.
2. The lead pump can run continuously or when called to run. The lag pump can function as a standby pump by not tying logic to the start or stop lag pump inputs.
3. The lag pump runs when the lead pump is not available and the lead pump was called to run, or when the lag pump is called to run.
 - a. Default sequence upon power-up is to lowest numbered pump available.
 - b. If no pumps available at power-up; default to first pump to become available.
4. Provide a FAIL output that goes true if there is no sequence selected or no pumps are available.
5. Accept 1-2 and 2-1 lead/lag sequence selection. Provide outputs to indicate sequence selection.

PLC Programming Guide

Inputs:

Input	Description	Data Type
S12	Sequence selection desired is pump 1 as lead and pump 2 as lag	BOOL
S21	Sequence selection desired is pump 2 as lead and pump 1 as lag	BOOL
RUNLEAD	Run Lead pump, set to TRUE for continuous lead operation	BOOL
RUNLAG	Run Lag pump	BOOL
AVAIL1	Pump 1 is available	BOOL
AVAIL2	Pump 2 is available	BOOL

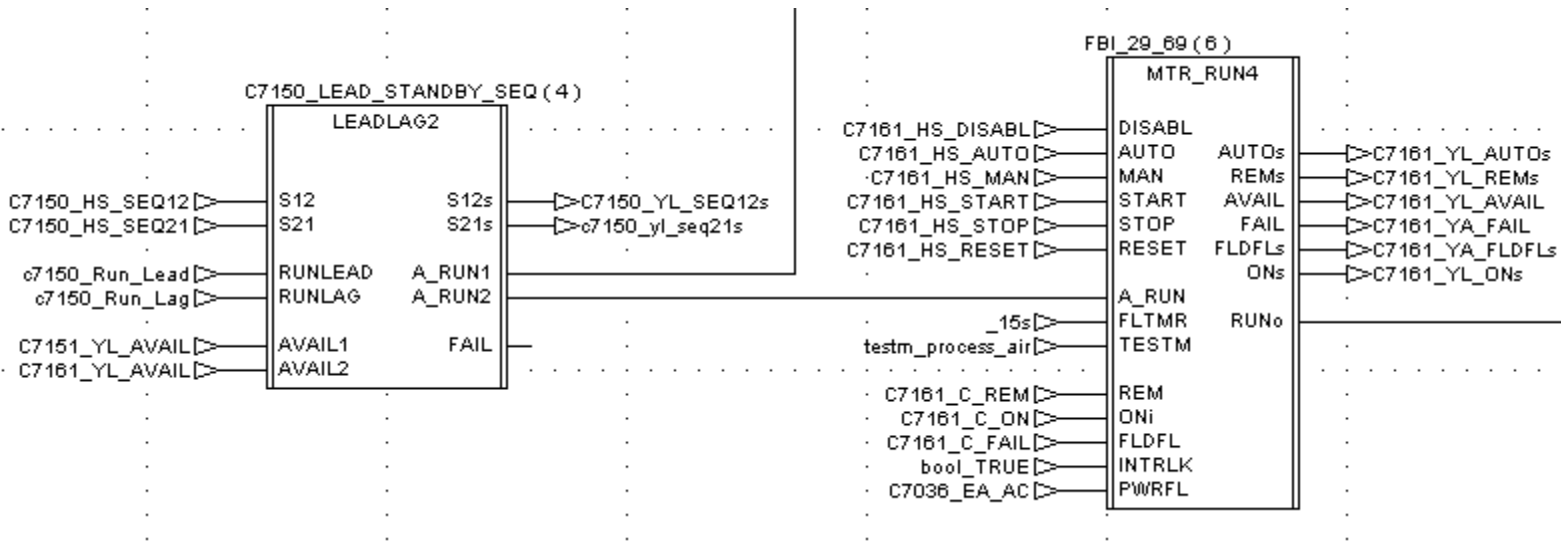
Outputs:

Output	Description	Data Type
S12s	Sequence selection status is 1,2	BOOL
S21s	Sequence selection status is 2,1	BOOL
A_RUN1	Auto run pump 1	BOOL
A_RUN2	Auto run pump 2	BOOL
FAIL	Fail	BOOL

BES SCS/PLC Standard: MTR_RUN4 Motor Control w/ RUNo output

Date: February 29, 2000	Lead: Bonnie Jackson	Status: PreFinal
Description:	Motor Control w/ Maintained RUNo output	
Comments:	Motor Control – Upon power fail DFB remains in AUTO (Only works if UPS powers PLC) If auto restart is not wanted following POWER fail use MTR_RUN2.	
Version:	Version: 9/11/2001 06:35:54	
Accepted by:		
Accepted by:		

PLC Programming Guide



Function Description:

MTR_RUN4 (HOR - Maintained PLC RUN output):

1. Controlled equipment has HAND-OFF-REMOTE (HOR) switches with ON and REMOTE inputs to the PLC. PLC outputs include a RUN signal (or separate START and STOP signals) to start the equipment.
2. If the equipment is not in REMOTE, or DISABLED, the SCS mode shall be forced to MANUAL (not AUTO).
3. Provide equipment AVAILABLE status that is logically true when the equipment is in REMOTE, SCS AUTO, INTERLOCKED, and not FAILED (by either internal or external FAILS).
4. The PLC logic will not attempt to start equipment by the PLC unless not DISABLED, the HOR switch is in the REMOTE position, the INTERLOCK is true, and not FAILED.
5. When the controlled equipment is in REMOTE the equipment mode can be set to AUTO or MANUAL, typically via the SCS. When the controlled equipment is in MANUAL the SCS and PLC shall be configured to RUN the equipment from the SCS. When the controlled equipment is in AUTO the automatic PLC routine shall control the equipment.
6. Monitor the ON status of equipment. Output the ON status to the SCS when not in TEST mode.
7. To provide a mechanism to easily test the systems without forcing physical I/O, accept TEST input. When the TEST input is true the RUN (or START/STOP) output shall be prohibited from energizing. When in this mode, the ON status output will follow the command to run, instead of the field ON input.
8. Monitor and output the REMOTE status of equipment.

PLC Programming Guide

9. If the PLC calls for the equipment to RUN (or START) (equipment must be in REMOTE) and does not receive an ON signal after a suitable time delay, lock out the equipment in the PLC and generate a FAIL alarm to the SCS. Generate the same alarm if the PLC calls for the equipment not to RUN (or STOP) and the ON signal is true after a suitable time delay. Prevent FAIL is DISABLEd.
10. The FAIL alarm shall be cleared on the transition of the REMOTE signal from unset to set, or cleared from a SCS RESET.
11. To eliminate FAILS due to power fails, accept a POWER FAIL input. When detected and the command to RUN or START still exists, inhibit the FAIL timer and FIELD FAIL output until the power is restored. Note PLC CPU is on an uninterruptible power supply.
12. Upon PWRFL (power failure) the automatic mode is retained.

Inputs:

Input	Description	Data Type
DISABL	Disable equipment and alarms	BOOL
AUTO	Switch to Auto mode	BOOL
MAN	Switch to Manual mode	BOOL
START	Start motor	BOOL
STOP	Stop motor	BOOL
RESET	Reset PLC generated FAIL	BOOL
A_RUN	Auto Run	BOOL
FLTMR	Fail Timer	TIME
TESTM	Test Mode	BOOL
REM	Hand switch in Remote	BOOL
ONi	Equipment on	BOOL
FLDFL	Field fail	BOOL
INTRLK	Interlock, normally closed (opens to prevent operation)	BOOL
PWRFL	Power Fail	BOOL

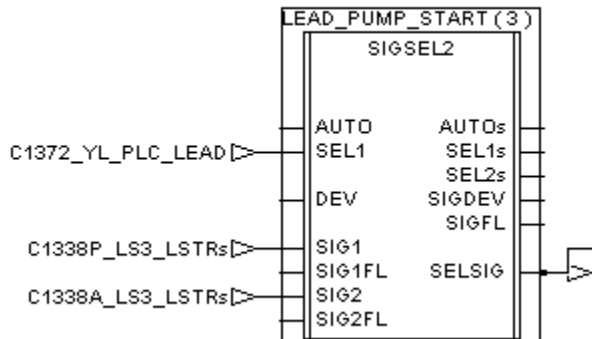
PLC Programming Guide

Outputs:

Output	Description	Data Type
AUTOs	Auto status	BOOL
REMs	Remote status	BOOL
AVAIL	Equipment Available	BOOL
FAIL	Fail (PLC generated)	BOOL
FLDFLs	Field fail status (from field contact)	BOOL
ONs	On status	BOOL
RUNo	Run output (to motor controller)	BOOL

BES SCS/PLC Standard: SIGSEL2 Two Signal Select

Date: January 7, 2000	Lead: George Jeffries	Status: PreFinal
Description:	Select between two signals	
Comments:	Selects between Two signals. In Auto selects signal 1, if signal 1 fails will switch to signal 2. In Manual follows selected signal without switch on fail.	
Version:	Version: / 06/06/01 11:08:35	
Accepted by:		
Accepted by:		



PLC Programming Guide

Function Description:

1. This DFB selects between analog signals.
2. In AUTO;
 - a. Default to signal 1.
 - b. If signal 1 fails automatically switch to signal 2.
 - c. Upon restoration of signal 1, after 5 second delay switch back automatically to signal 1.
3. Not AUTO; follow the selected signal.
4. Provide a FAIL output true when;
 - a.. In AUTO and both signals failed.
 - b. Not AUTO and selected signal fails.
5. Provide Signal Deviation output true when the difference between signals is greater than DEV input.

Inputs:

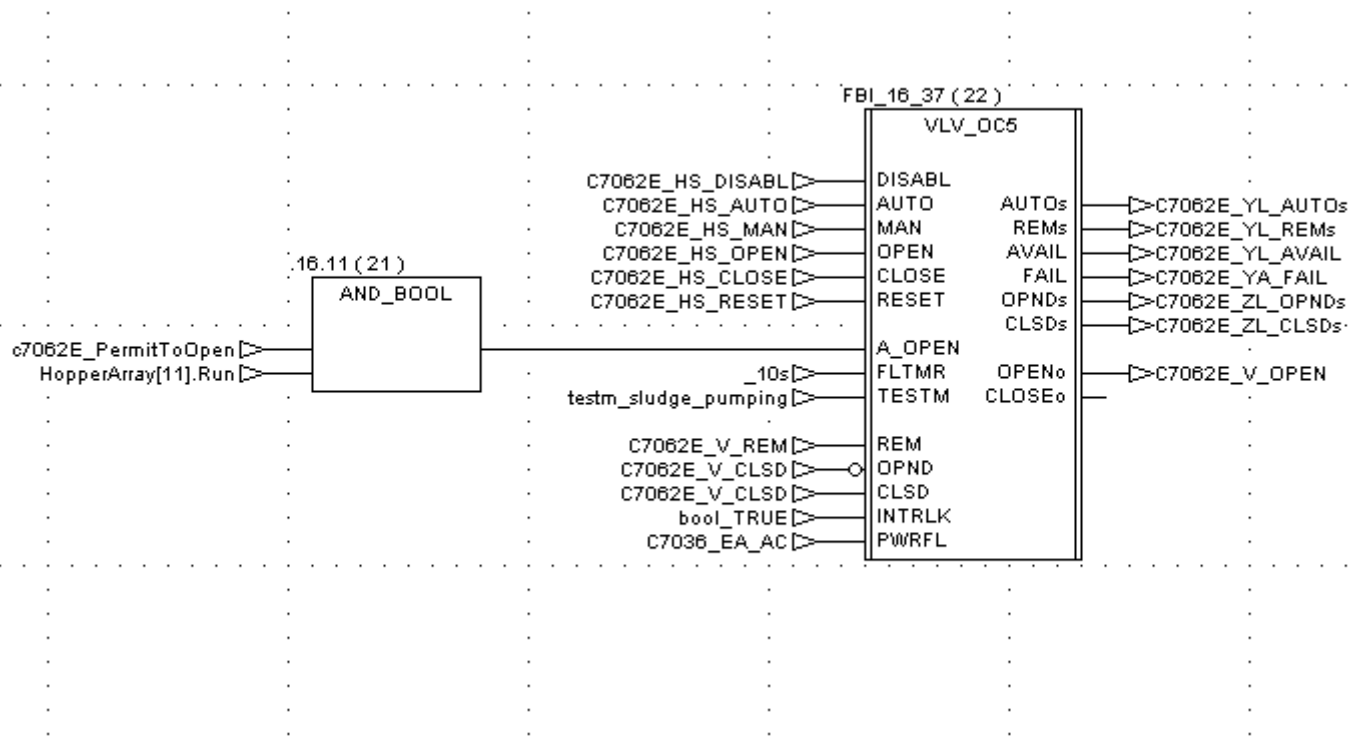
Input	Description	Data Type
AUTO	Automatic selected.	BOOL
SEL1	Signal 1 selected	BOOL
DEV	Maximum deviation allowed	REAL
SIG1	Signal 1	REAL
SIG1FL	Signal 1 failed	BOOL
SIG2	Signal 2	REAL
SIG2FL	Signal 2 failed	BOOL

Outputs:

Output	Description	Data Type
AUTOs	AUTO status	BOOL
SEL1s	Signal 1 selected status	BOOL
SEL2s	Signal 2 selected status	BOOL
SIGDEV	Signal deviation excessive	BOOL
SIGFL	Controlling signal/s failed	BOOL
SELSIG	Signal selected for control	REAL

BES SCS/PLC Standard: VLV_OC5 Valve Control w/ OPEN/CLOSE Outputs

Date: February 29, 2000	Lead: Deana Jones	Status: PreFinal
Description:	Valve Control w/ OPEN/CLOSE PLC outputs <i>(Remains in AUTO if system was in AUTO before power fail)</i>	
Comments:	Valve/Gate Control – PLC OPENo/CLOSEo outputs with DISABL function; Use VLV_OC3 if default to Manual desired after power fail	
Version:	Version: 9/11/01 08:19:50	
Accepted by:		
Accepted by:		



Function Description:

1. Controlled valves have OPEN-CLOSED-REMOTE (OCR) switches with OPEN, CLOSED and REMOTE inputs to the PLC. PLC outputs include an OPEN and CLOSE signal to operate the valve. Do not attempt to operate valve by the PLC unless the HOR switch is in the REMOTE position.
2. Equipment Available status: Provide equipment available status that is logically true when the equipment is in REMOTE, SCS AUTO and not FAILED.
3. When the controlled valve is in REMOTE the SCS and PLC shall be configured to set the valve mode to SCS AUTO or SCS MANUAL. When the controlled valve is in

PLC Programming Guide

SCS Manual the SCS and PLC shall be configured to Open and Close the valve from the SCS. When the Controlled valve is in SCS Auto the automatic PLC routine shall control the Open/Close position of the valve

4. Monitor the OPEN/CLOSED status of valve in the REMOTE position. If the SCS/PLC calls for the valve to OPEN and does not receive an OPEN signal after a suitable time delay, lock out the valve in the PLC and generate an alarm to the SCS.
5. If the PLC calls for the valve to CLOSE and continues to receive an OPEN signal or does not receive a CLOSED signal after a suitable time delay, lock out the valve in the PLC and generate an alarm to the SCS.
6. The alarm condition and the valve shall be cleared when the valve is RESET.
7. After a FAIL and RESET the CLOSEo will be true.
8. FAIL, or DISABL, or not REM; default is to MAN mode.
9. If in AUTO when power fails restart in AUTO after power restored. Requires PLC be on an uninterruptable power supply (UPS) for this to happen.

Inputs:

Input	Description	Data Type
DISABL	Disable equipment and alarms	BOOL
AUTO	Switch to Auto mode	BOOL
MAN	Switch to Manual mode	BOOL
OPEN	Open valve	BOOL
CLOSE	Close valve	BOOL
RESET	Reset PLC generated FAIL	BOOL
A_OPEN	Auto Open	BOOL
FLTMR	Fail Timer	TIME
TESTM	Test Mode	BOOL
REM	Hand switch in Remote	BOOL
OPND	Valve Opened	BOOL
CLSD	Valve Closed	BOOL
INTRLK	Interlock, normally closed (opens to prevent operation)	BOOL
PWRFL	Power Fail	BOOL

PLC Programming Guide

Outputs:

Output	Description	Data Type
AUTOs	Auto status	BOOL
REMs	Remote status	BOOL
AVAIL	Equipment Available	BOOL
FAIL	Fail (PLC generated)	BOOL
OPNDs	Opened status to SCS	BOOL
CLSDs	Closed status to SCS	BOOL
OPENo	Open output (to valve actuator)	BOOL
CLOSEo	Close output (to valve actuator)	BOOL