GE Fanuc Automation

Programmable Control Products

Series Six Parallel I/O Transmitter Module for the Series 90-70 PLC

User's Manual

GFK-1883 *May 2002*

Warnings, Cautions, and Notes as Used in this Publication

Warning

Warning notices are used in this publication to emphasize that hazardous voltages, currents, temperatures, or other conditions that could cause personal injury exist in this equipment or may be associated with its use.

In situations where inattention could cause either personal injury or damage to equipment, a Warning notice is used.

Caution

Caution notices are used where equipment might be damaged if care is not taken.

Note

Notes merely call attention to information that is especially significant to understanding and operating the equipment.

Revision 3.270

Released with firmware build 127

The S6PIO module is manufactured and supported by Real Time Consultants Pty Ltd under license from Real Time & Embedded Systems Pty Ltd.

Content of This Manual

Related Publications

GFK-1179H ***Installation Requirements for Conformance to Standards – June 2000*** (GE Fanuc Automation North America, Inc).

GEK-96602A ***Series Six Plus Programmable Logic Controller User's Manual*** (GE Fanuc Automation North America, Inc).

GFK-0171B ***Series Six Genius Bus Controller User's Manual*** (GE Fanuc Automation North America, Inc).

GFK-0398C ***90-70 Genius Bus Controller User's Manual*** (GE Fanuc Automation North America, Inc).

GEK-25364A ***Series Six Programmable Controller CCM Communications User's Manual*** (GE Fanuc Automation North America, Inc).

GEK-25367E ***Series Six Programmable Controllers Data Sheets*** (GE Fanuc Automation North America, Inc).

GEK-25379D ***Logicmaster TM 6 Programming and Documentation Software - User Manual*** (GE Fanuc Automation North America, Inc).

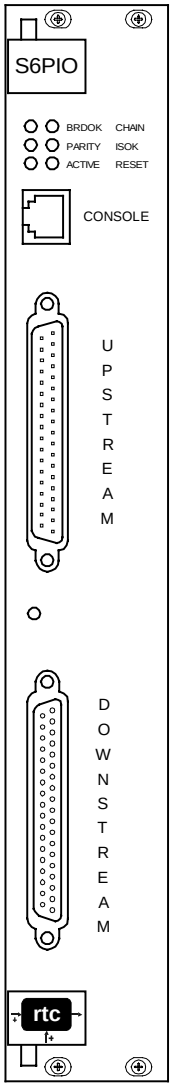
Figures

Tables

Chapter 1 INTRODUCTION

The aim of this chapter is to provide a brief overview of the S6PIO module.

Series Six Parallel I/O Interface Transmitter Module (S6PIO)



This User Manual is for the Series Six Parallel I/O Transmitter Module (S6PIO), which provides an interface to GE-Fanuc Series Six™ Input and Output (I/O) modules. The S6PIO module complies with the VME standard (VME REV C.1; October 1985) and may be used in GE-Fanuc Series 90-70 PLC systems and standard VME systems. The S6PIO module allows users to update their existing Series Six installations without replacing all of the installed I/O modules.

Figure 1-1. The S6PIO Front Panel, shows the current front panel layout. The S6PIO was previously released with different front panel markings for the LEDs. The function of each LED however remains the same as the previous superceded front panel.

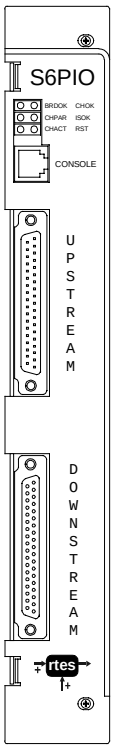


Figure 1-1. The S6PIO Front Panel (left) Superceded Front Panel (right)

Modes of Operation

The S6PIO module has two major modes of operation:

Master Mode

In Master Scan mode, the S6PIO module acts as a Series Six Bus Master. To the I/O system, the S6PIO module is indistinguishable from a Series Six CPU. In Master Scan mode, the S6PIO module can perform standard I/O scans or provide Direct Memory Access (DMA). When a window is opened to a DMA device (such as an I/O Communication Control Module (IOCCM)), data transfers occur between the DMA device and *emulated* Series Six memory tables that are resident on the S6PIO module.

Passive Mode

In passive mode, the S6PIO module can listen for standard I/O scans issued by another Series Six Bus Master (either a Series Six CPU or another S6PIO module). Such scan data is accumulated into S6PIO module memory. When a given synchronization address is recognized on the bus, the scan data can be transferred to the Series 90-70 PLC. Usually, the Series Six Program Development Terminal (PDT) window is used for synchronization.

Module features

Selectable Scan Rates

The default I/O scan rate of the S6PIO module is close to that of the standard Series Six CPU. The default setting provides a scan of 1000 I/O in approximately 5.6msec bus time. The S6PIO module has the potential to scan at faster rates. Scan rates are user-selectable (see Table 1-1. Module Specifications below).

Status LEDS

The S6PIO module has 6 green LED indicators mounted on the front panel. During power-on initialization, all of the LEDs will cycle twice through an LED test routine before being set to their standard states. Each of the LEDs are described below:

BRDOK (Board OK) LED.

The initial state (after reset) of the BRDOK (Board OK) LED is off. After configuration is complete and the S6PIO module is ready for operation, the BRDOK (Board OK) LED should be on. If the application program has not been programmed into on-board flash memory, or an error is detected with the initialization of the S6PIO module, the BRDOK (Board OK) LED will remain off.

CHAIN (Chain OK) LED

The CHAIN (Chain OK) LED is on when power is on at all downstream stations and continuity is OK to all upstream points. If any of these conditions are not met then the CHAIN (Chain OK) LED is off. Previously this LED was marked CHOK.

PARITY (Chain Parity) LED

The PARITY (Chain Parity) LED is on if output parity is OK at all downstream stations. The PARITY (Chain Parity) LED is off if there is an output parity error at one or more stations. This LED is based on the state of the bus parity signal and will flicker during DMA cycles (in passive mode) as the parity bus signal line is used for another purpose during DMA windows. Previously this LED was marked CHPAR on the front panel.

ISOPWR (Isolated Power) LED

The ISOPWR (Isolated Power) LED is on if the output voltage of the onboard, isolated +5V DC-DC converter is within tolerance. If the DC-DC converter voltage is out of tolerance then the ISOPWR (Isolated Power) LED is off.

ACTIVE (Command Handler Active) LED

Indicates when the S6PIO command handler is processing a command. This may be VME communications via the back-plane, or S6PIO VME commands issued through the serial port. In normal operation, the commands arrive via the VME back-plane from the 90-70 and hence this LED indicates communications is active between the S6PIO and the 90-70 CPU.
To present an acceptable user interface, the ACTIVE LED remains on for 200 milliseconds whenever a command is processed. This results in constant brightness during normal scan operations. This LED was previously marked CHACT on the superceded front panel.

RESET LED

The RESET (RST) LED FLASHES if the S6PIO is in master mode AND RST is asserted.
The LED is ON if the S6PIO is in master mode AND RST de-asserted.
The LED is OFF if the S6PIO is in passive mode or the S6PIO is idle (idle occurs after reset or power up and before any C blocks have been executed)

Special LED Functions

When the S6PIO module is placed in Self-test mode, the 6 LEDs described above assume special functions. These special LED functions are described later.
(refer Chapter 8 TROUBLESHOOTING in relation to self-test mode).

Small Onboard LED

This LED is on at power-up and is turned off after module initialization has been completed.

Onboard Reset Button

This button resets the onboard processor. It can be used during module setup and testing or during the FLASH download procedure.

Caution

This button is NOT intended to be used during normal operation.

Jumpers

The S6PIO module has 27 jumpers, the functions of which are discussed in the section entitled Jumper Settings (refer Chapter 2 INSTALLATION & WIRING, below). If the S6PIO module is used in a 90-70 PLC system then the factory default settings for these jumpers are generally adequate.

C Blocks

A number of C Blocks are supplied to interface Series 90-70 PLC system software with the S6PIO module. The C Blocks are described in the section entitled C Blocks.
(Refer Chapter 4 SOFTWARE INTERFACE, below).

Module Specifications

This section details specifications of the S6PIO module.

Environmental

Operating Temperature	0° to +50°C, no forced air cooling, 0° to +60°C, for forced air cooling, inlet air to bottom of VME rack.
Storage Temperature	–40° to +85°C (-40° to +185°F)
Humidity	5% to 95% non condensing.
Vibration	1G @ 40-150Hz, 0.012in p-p @ 10-40Hz
Shock	15G's for 11msec.

Mechanical

Board Height	6U.(double Euro card height).
Slot Spacing	Alternate front panel designs allow for 0.8in or 1.6in slot spacing.
Backplane	J1 backplane only.

General Module Details

Technology	Designed using state of the art technology, incorporating FPGA and embedded microcontroller.
Configuration Jumpers	Jumpers for setting bus termination and plug and play operation. Also jumpers for manufacturing testing and diagnostic purposes.
Front Panel Port	RS232 front panel port. for configuration and debug use.
LED Indicators	6 LED indicators mounted on the front panel.
Standalone Operation	For testing, can be set for standalone scanning. Allows a host device access to the I/O bus as a VME host.
Firmware Upgrade	Field upgradeable via the RS232 front panel port.
Power Supply	Operates on +5V power rail only. Consumes < 2.0 amps @ 5V.

I/O Interface

General	GE Fanuc Series Six I/O bus, or GE Drive Systems C-Bus.
Connectors	One DB37 male and one DB37 female connector.
Cable	Any Series Six compatible I/O cable.

Isolation	1,500 volts dielectric isolation.
Total Bus length	Maximum 2000ft (600 meters).
I/O Bus segment length	Maximum 500ft (150 meters).
Number of Segments	Maximum four segments downstream from the S6PIO module.
Number of Racks	10 daisy chained Series Six I/O racks.
Genius Bus Controller Specific I/O Chain limits	When a Series Six Genius Bus Controller is to be driven by an S6PIO, the total I/O chain length between the S6PIO and the I/O rack containing the GBC must not exceed 1050ft (320 meters). There must be no more than two I/O transmitter modules and three daisy chained I/O racks between the S6PIO and the GBC.
Downstream Interface	Series Six I/O Receiver Module or a C-Bus STMA interface module.
Normal or Expanded I/O	Up to 8 channels in Expanded Scanning mode. Only normal mode may be used on C-BUS, unless the C-BUS connection is downstream of an I/O Transmitter module.
User Selectable I/O Scan Times	45us per byte – 5.625ms per 125 addresses, max bus length 2000' 30us per byte – 3.75ms per 125 addresses, max bus length 2000' 15us per byte – 1.875ms per 125 addresses, max bus length 500'
Watchdog Timer	Programmable from 270msec to 2550msec in 10msec increments. (Programmable via a 90-70 C-block)
Passive Mode	This mode allows the module to monitor the I/O bus cycles produced by another S6PIO module running in Master Mode, or by a Series Six CPU
Window Bus Cycles	Can generate block transfer cycles for servicing smart devices requiring the use of a PDT or DPREQ Window.
Parity Error Retries	Number of parity retries may be specified in a command option. If the specified number of retries fail to correct the error, the input table byte is not updated and a fatal parity error is reported..
Priority Input I/O Cycles	The S6PIO can perform priority input cycles when the MS address bit is a "1". This causes the address to be ignored by output cards, however input cards respond normally.
VME Bus Interface	
VME Specification	Meets the VME REV C.1 standard dated October 1985.
Slave Module	Is designed as a VME slave module.
Configuration for 90-70 PLC	When installed in a 90-70 PLC the module configuration is plug and play. The VME slave address and address modifiers are determined by the rack slot number.
Data Transfer Bus	Programmable to allow A16/D8, A16/D16, A24/D8 and A24/D16 transfers in either user or supervisor mode. A16 and A24 addressing shall both map to onboard dual ported RAM. RMW cycles are supported.
Dual Port RAM	4K words (16 bit) of dual ported RAM.
Mailbox Interface	Interrupt mailbox, triggered by VME write to a specified dual port memory address.
Priority Interrupt Bus	Can generate vectored interrupts on any programmable interrupt level from level 1 to level 7.
SYSFAIL	Can assert SYSFAIL on a board error condition. This option is jumper selectable.
Slot Sensing	Automatically senses the slot number in a 90-70 rack.
Rack Compatibility	Can be used in a 90-70 PLC CPU rack, or any local expansion rack. Standard or integrator racks may be used. It can not be installed in a remote I/O rack.
Number of Modules	A maximum of 4 S6PIO modules may be installed per local 90-70 I/O or CPU rack. Allows a maximum of 24 S6PIO modules in a fully configured 90-70 PLC system.
Software	
Emulated Series 6 Memory	Emulates Series Six Plus CPU memory tables. Emulates Series Six scratchpad to ensure interoperability. Function provided to move data between the 90-70 I/O tables and the emulated Series Six I/O tables.
Event Log	A circular PLC time stamped event log of 200 events is maintained
Status	Module status is available in dual port RAM.
Software Interface	A number of C Blocks are provided for use as the Series 90-70 PLC interface.
Asynchronous operation	Option, for commands to be executed asynchronously via C block.
Return Status	16 bit status register is returned indicating the module status at the completion of a command.
Return Error Codes	A number of error codes may be returned by the C Block.
Messages to PLC Fault Table	Messages posted to PLC fault table via the C Block. Fault message includes S6PIO slot number and C Block name.

Table 1-1. Module Specifications

Chapter 2

INSTALLATION & WIRING

Installing and Removing a Module

Installing a Module

General

The S6PIO module plugs into the J1 bus only of any VME, 6U high rack.
Prior to installing the S6PIO module, ensure that all of the jumper settings are appropriate to user requirements. The jumper functions and settings are described in the section entitled Jumper Settings (below).

Always turn rack power off prior to attempting to install the S6PIO module.

Caution

As with any complex electronic equipment, use standard anti-static precautions when handling the S6PIO module.

Installing in a Series 90-70 Rack

The S6PIO module can only be used in a CPU rack, or any local expansion rack. It can not be installed in a remote I/O rack.
The S6PIO module may be installed in any slot to the right of the CPU module in a CPU rack, or to the right of a BRM module in a local expansion rack. The S6PIO should be placed to the right of any modules capable of generating interrupts. If the S6PIO module is to be installed in an existing system then simply select the next available slot.
If the Series 90-70 PLC is used to control factory equipment then schedule a time period when the rack power may be safely shut off without affecting production or equipment.

Note

Once the board is installed into the 90-70 rack it is important to configure the 90-70 CPU with the slot and address information for the S6PIO. This is done via the LM90 software by selecting a third party VME card for the card type. Refer to Chapter 3 CONFIGURATION for more details.

Jumper Settings

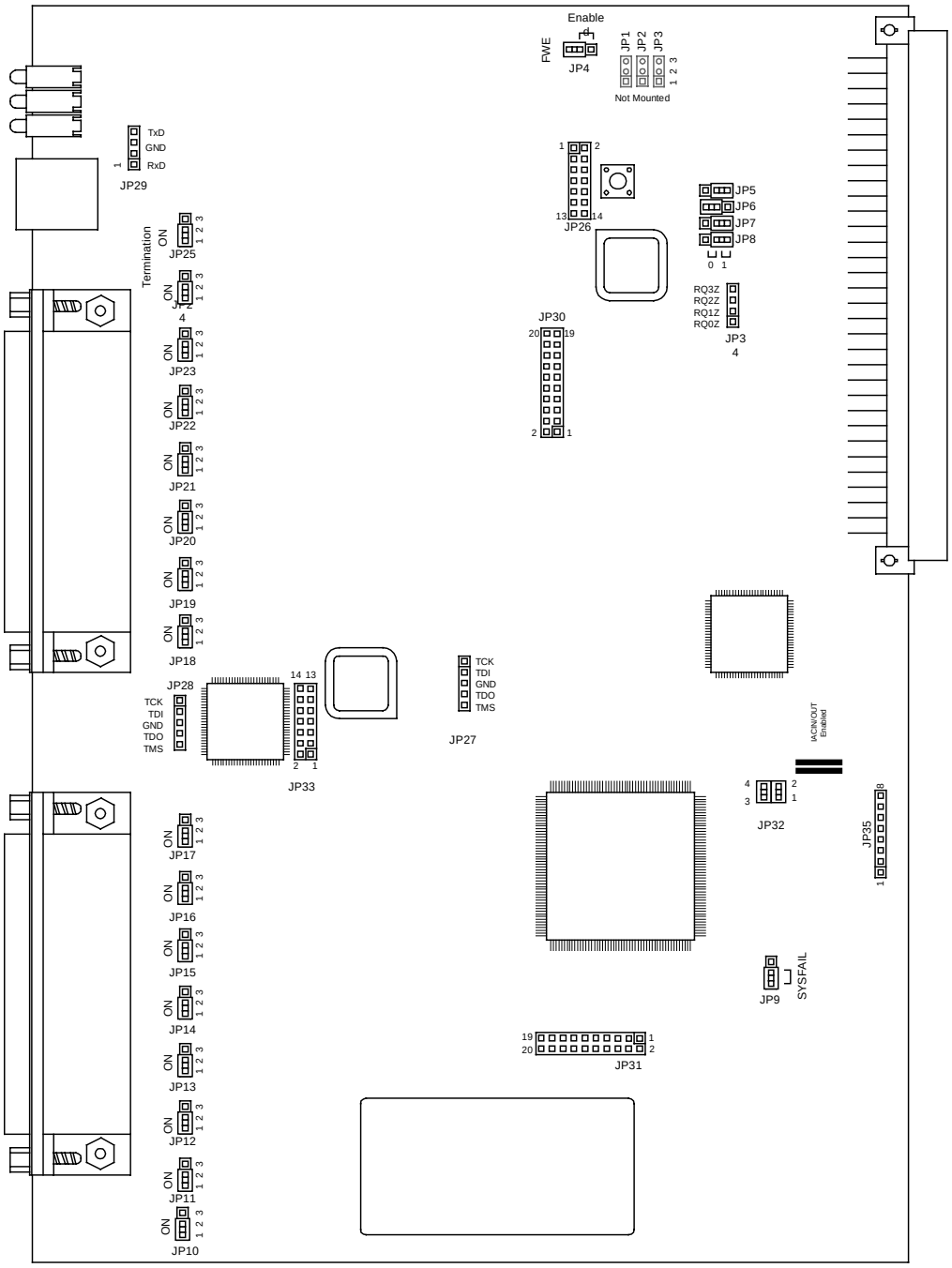


Figure 2-1. S6PIO Module Showing Jumpers in "As Shipped" Position.

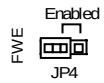
Jumpers JP1, JP2 and JP3

Microcontroller Mode Selection. These are factory set and cannot be changed.
Factory default positions:

JP1	2-3
JP2	1-2
JP3	1-2

On revision 2.00 and later boards these jumpers are not physically fitted.

Jumper JP4 – FLASH Memory Protect

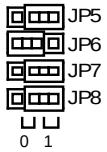


FLASH memory protect jumper. This jumper is set to the protected (disabled) position in the factory and should not be moved unless field upgrading the on board FLASH memory.

Enabled	FLASH memory is write enabled
Protected	FLASH memory is protected (normal position)

The diagram on the left shows the normal position for the jumper

Jumpers JP5, JP6, JP7 and JP8 - Configuration



Configuration jumpers JP5 & JP6 specify the configuration mode for the S6PIO module.
The diagram on the left shows the normal position for these jumpers

Plug & Play Mode

This is the factory default position. In this mode the S6PIO module automatically senses the slot number into which it is plugged and configures its VME slave interface according to the slot number. The actual slave interface address and address modifiers are described later. This mode is the normal mode of operation for a Series 90-70 PLC system.

JP5	1
JP6	0

Self-test Mode

In this mode the S6PIO module continually runs manufacturing self-tests. The progress of the self-tests is indicated on the front panel LEDs and is described in Chapter 8 TROUBLESHOOTING

JP5	0
JP6	0

VME Configuration from FLASH Mode

This mode is intended to be used when the S6PIO module is installed in a standard VME system. The VME slave address and address modifier are programmed into onboard FLASH memory during S6PIO module configuration prior to first use.

JP5 1
JP6 1

VME Disabled Mode

In this mode the VME bus is disabled.

JP5 0
JP6 1

Boot from FLASH or RAM

JP7 specifies whether to boot from a program in FLASH or RAM. This mode is only used during module testing. The factory default position is to boot from FLASH.

Position 0 Boot from RAM
Position 1 Boot from FLASH

Boot into Onboard Monitor

JP8 specifies whether to boot into a monitor program for diagnostic and testing purposes or the application program. The factory default is to boot into the application program.

Position 0 Boot into monitor program
Position 1 Boot application program

Jumper JP9 – VME SYSFAIL

The diagram at left shows the jumper with SYSFAIL enabled (normal position), moving the jumper to the upper two pins disables SYSFAIL.
SYSFAIL is asserted from the S6PIO module at power-up until all self-tests are complete.

Jumpers JP10 through JP25 - Bus Termination

These jumpers are Series Six Bus termination jumpers. In the default position (ON, pins 1-2 jumpered) terminates all signals on the 2 DB37 connectors with 150ohm resistors. This setting is appropriate when the S6PIO module is in Master Scan mode at the start of an I/O chain and there is only one Series Six I/O cable plugged into a DB37 connector (generally, the bottom DB37 connector).

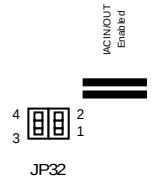
Jumpers JP10 through JP25 should be placed in pins 2 to 3 position when the S6PIO module is used in Passive mode (where the S6PIO module is not at the beginning of an I/O chain). In such circumstance, Series Six I/O bus cables will be plugged into both DB37 connectors.

Alternatively a termination plug (RTC part # J1320-P003) can be used instead of terminating internally. This is convenient if constant switching between master and passive mode is required.

Caution

Care should be taken to ensure the I/O chain is not multiply terminated. If an external termination is used and the internal termination is also present then system mis-operation is likely to occur.

Jumper JP32 – IACKIN/OUT Daisy Chain



Positions 1 to 2 and 3 to 4 to bypass the IACK daisy chain
Positions 1 to 3 and 2 to 4 to enable the IACK daisy chain

The diagram at right shows the chain bypassed.

Header Descriptions

There are a number of headers present on the S6PIO module. The functions of some of these headers are described below.

Header JP26

Connector for low level FLASH programming adapter. This is used in initial factory testing to load the low level bootstrap monitor.

Headers JP27 and JP28

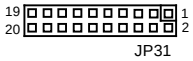
Connectors for EPLD programming bridge cable. This is used if ever the onboard EPLD needs to be field-upgraded.

Header JP29

Second RS232 Port. This serial port is configured to run at 57,600 baud and outputs event log messages in ASCII format.

Pin 1 = RX
Pin 2 = Signal Ground
Pin 3 = Signal Ground
Pin 4 = TX

Header JP31

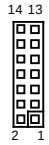


Provides monitoring and debugging signals via the FPGA. Logic high indicates the function is active.

Pin 2 = VME command handler active.
Pin 4 = VME mailbox ISR executing.
Pin 6 = Not used.
Pin 8 = DMA or passive ISR executing.

Pin 10 = Checksum error.
Pin 12 = DMA window, master/passive.
Pin 14 = Spare
Pin 16 = DMA debugging signal

Header JP33



Terminations for spare pins of the EPLD chip. Certain Series Six bus signals are available on this header as follows and is provided for low level debugging purposes only:

Pin 2 = ADS, address strobe
Pin 4 = IS, input strobe
Pin 6 = CP, card present
Pin 8 = PE, parity error

Field Wiring Options.

The S6PIO can be wired in two main modes: passive mode, for monitoring a Series Six I/O chain, or master mode, for driving a Series Six I/O chain. The mode that the S6PIO is operating in depends, not only on the wiring option chosen, but also on the C blocks used in the 90-70 program that control the S6PIO. It is the user's responsibility to ensure the appropriate C blocks are utilized for the selected field-wiring configuration.

On power up the S6PIO is in an idle state and will enter master or passive mode depending on which C block is executed first. Once the device is in passive mode it will not switch into master mode until the S6PSV block (refer Chapter 4 SOFTWARE INTERFACE) is used to stop passive mode or the 90-70 program is restarted.

Passive Mode

In this mode, the S6PIO module may be used to monitor an existing I/O bus. This mode is very useful during system change over as it allows the Series Six I/O to be mirrored into a 90-70 PLC system. The 90-70 PLC has access to all of the inputs and outputs scanned by the Series 6 CPU.

Passive mode can demultiplex analog inputs or outputs "on the fly". This is useful where DOIOs are used in the Series Six to rapidly read analog inputs or write analog outputs. Demultiplexed data is maintained in a separate table. Use of this feature requires the user to specify the layout of analog input and analog output cards in an address map (refer Chapter 4 SOFTWARE INTERFACE for further detail).

The synchronization feature allows the sweep time of the 90-70 to be locked to the sweep time of the Series Six.

Note

It is not recommended to use a single 90-70 to monitor I/O chains spread across multiple Series Six PLCs.

Note

In Passive mode, the entire chain is driven by the Series Six I/O Controller (IOC) or Auxiliary I/O Controller (AUX IOC). Accordingly, care should be taken to position the Series 90-70 rack such that the 50 feet (15.24 meters) total cable limit for the entire chain is not exceeded

With S6 Primary I/O Chain Only

In this configuration, a Series Six CPU controls a primary I/O chain, which is daisy-chained through an S6PIO module, and then out to the I/O system.

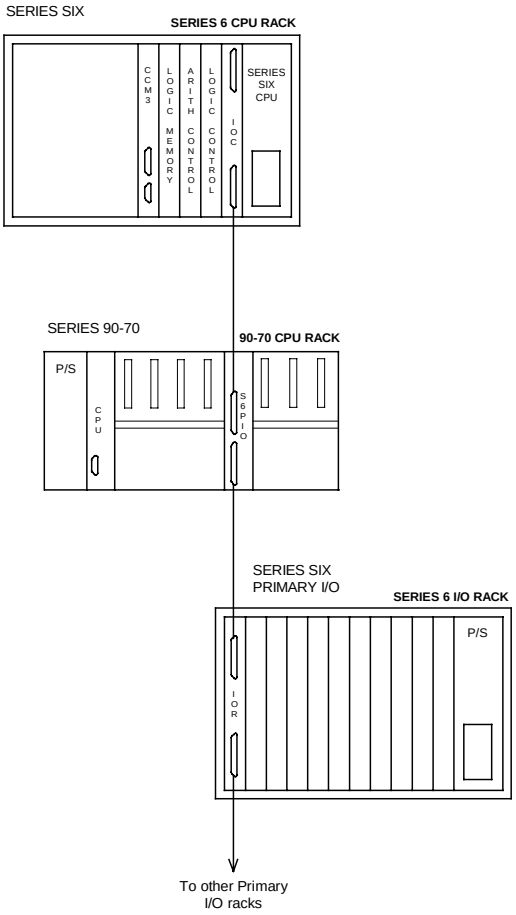


Figure 2-2. Passive Mode wiring with Series Six Primary I/O Only

S6 Primary and Auxiliary I/O Channels

In this configuration, a Series Six CPU controls primary and auxiliary I/O chains. Both chains are daisy-chained through separate S6PIO modules and then out to the respective I/O systems.

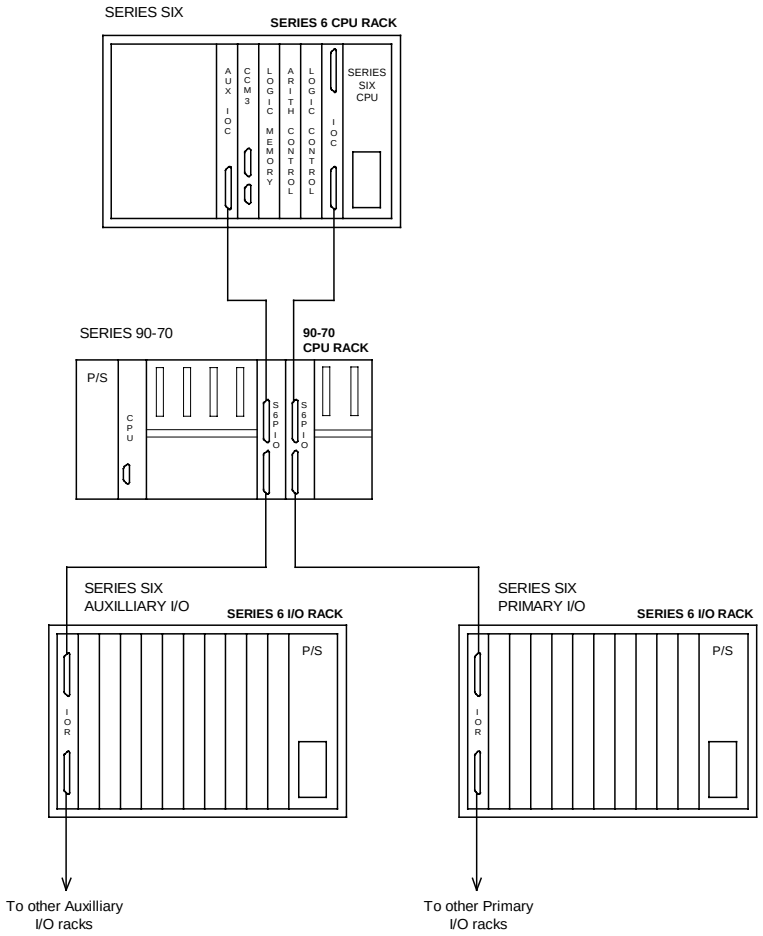


Figure 2-3. Passive Mode wiring with Series Six Primary and Auxiliary I/O's

Master Mode

In this mode, the S6PIO module is used to control an existing Series Six I/O bus with a Series 90-70 processor.

Note

The S6PIO module has the same driving capability as an I/O Transmitter module, therefore in master mode the S6PIO module is capable of driving a 500 feet (152.4 meters) cable segment.

With S6 Primary I/O Chain Only

In this configuration, a Series 90-70 CPU has replaced the original Series Six CPU, and now controls the Series Six primary I/O chain.

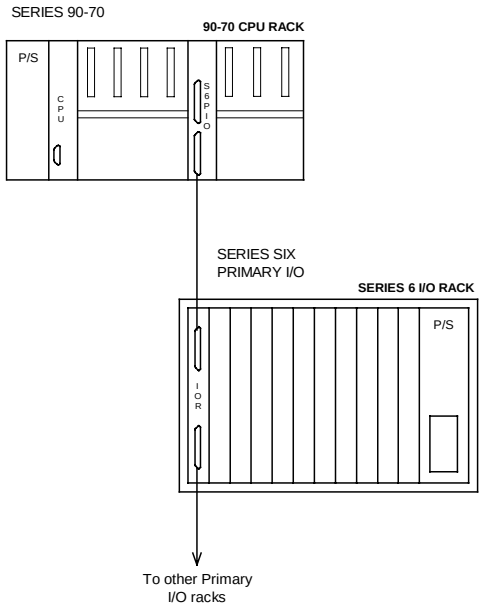


Figure 2-4. Master Mode wiring with Series Six Primary I/O Only

S6 Primary and Auxiliary I/O Channels

In this configuration, a Series 90-70 CPU replaces the original Series Six CPU and controls the Series Six primary and auxiliary I/O chains. Figure 2-5 below, two S6PIO modules are installed in the Series 90-70 CPU rack to provide interface functionality.

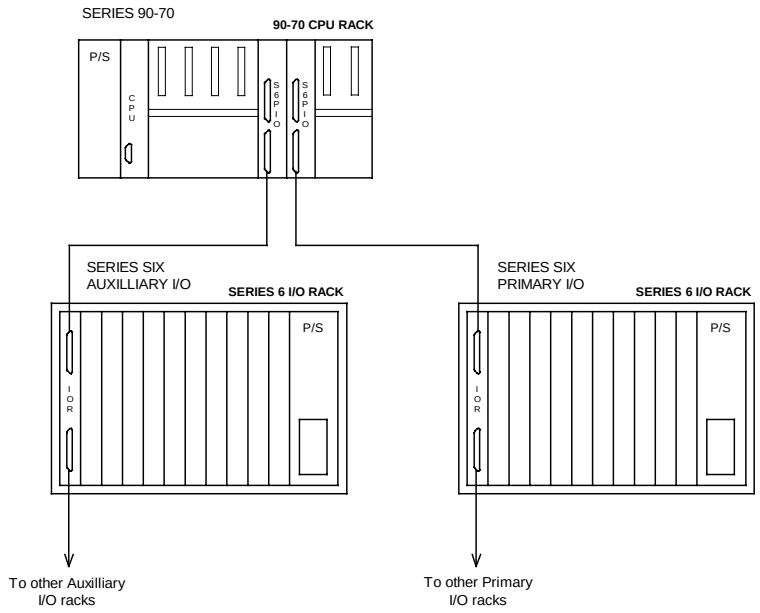


Figure 2-5. Master Mode wiring with Series Six Primary and Auxiliary I/O

90-70 In Redundancy Configuration

In this configuration, two Series 90-70 CPU's replace the original Series Six CPU and control the Series Six primary I/O chain (in a redundancy configuration). In Figure 2-6 below, S6PIO modules are installed in each of the Series 90-70 CPU racks to provide interface functionality.

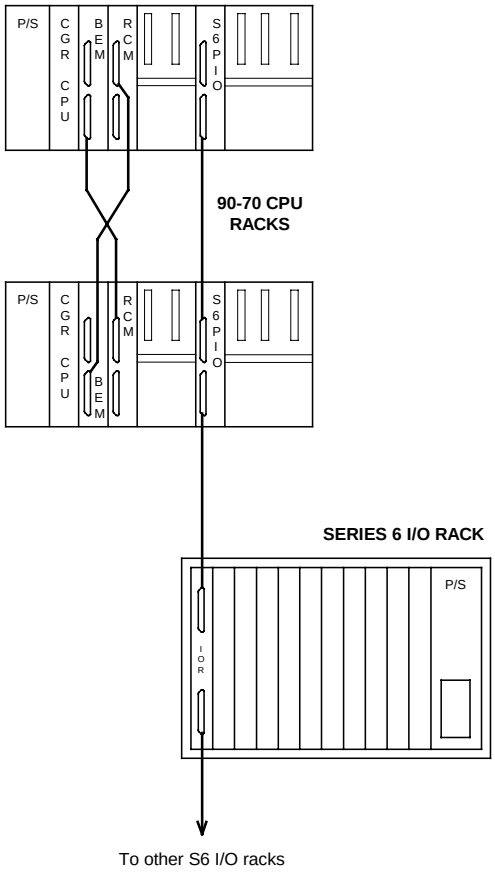


Figure 2-6. Master Mode wiring with two 90-70's in Redundancy Configuration.

An alternate redundancy configuration comprises two S6PIO modules in each Series 90-70 CPU rack to control primary and an auxiliary Series 6 I/O chains.

Refer to "Appendix B EXAMPLE 90-70 PROGRAMS", for example Series 90-70 coding.

Chapter 3 CONFIGURATION

Outlines the various configuration modes and how the user may configure the S6PIO. Refer to Chapter 2 INSTALLATION & WIRING for details on how to set the configuration mode.

Plug & Play Mode

If Plug & Play mode is selected then the S6PIO module will configure its VME interface in accordance with Table 3-1. VME Slave Addressing by Slot Number. This configuration allows the S6PIO module to be used in (Series 90-70) rack 0 or any of the local expansion racks 1-7 without impacting on the use of GE Fanuc modules in the same racks. The use of other third party VME modules should be carefully considered.

Note

A S6PIO module cannot be installed in slot 1 of any rack.

Note

The configuration should also be entered via the LM90 configuration program to ensure the 90-70 CPU is aware of the location and addressing range of the card.

Slot Number	S6PIO AM Code	S6PIO VME A16 Address	Dual Port Memory Size
9	0x29	0xE000	8K, byte access
8	0x29	0xC000	8K, byte access
7	0x29	0xA000	8K, byte access
6	0x29	0x8000	8K, byte access
5	0x2D	0xE000	8K, byte access
4	0x2D	0xC000	8K, byte access
3	0x2D	0xA000	8K, byte access
2	0x2D	0x6000	8K, byte access
1	N/A	N/A	N/A

Table 3-1. VME Slave Addressing by Slot Number

The supplied Series 90-70 C Blocks use special programmed address modifier codes to select the correct Series 90-70 CPU or Expansion rack. These codes are transparent to the user but are documented in the following table for completeness.

Rack Number	S6PIO AM Code	90-70 Programmed AM Code
0	0x29, 0x2D	0x29, 0x2D
1	0x29, 0x2D	0x1E, 0x16
2	0x29, 0x2D	0x1D, 0x15
3	0x29, 0x2D	0x1C, 0x14
4	0x29, 0x2D	0x1B, 0x13
5	0x29, 0x2D	0x1A, 0x12
6	0x29, 0x2D	0x19, 0x11
7	0x29, 0x2D	0x18, 0x10

Table 3-2. 90-70 Rack Selection AM Codes

User Configuration Mode

In this mode, the S6PIO uses configuration data that has been written to FLASH memory. This mode is not supported in a Series 90-70 PLC.

The user may specify A16 or A24 addressing using address modifiers 0x29, 0x2D, 0x39 or 0x3D. No other address modifiers are supported.

The S6PIO occupies an 8K-memory window. In either A16 or A24 mode the address may be set on any 8K boundary.

Refer to Chapter 7 UTILITY PROGRAMS, for details on how to program configuration data into FLASH memory.

Chapter 4

SOFTWARE INTERFACE

90-70 Module Configuration

Once the board is installed into the 90-70 rack it is important to configure the 90-70 CPU with the slot and address information for the S6PIO. This is done via the LM90 software by selecting a third party VME card for the card type. Refer to "Chapter 3 CONFIGURATION" for more details.

C Blocks

The 90-70 PLC interface software is in the form of a collection of C Blocks. The C blocks provide the interface between Series 90-70 PLC application programs and the S6PIO module. At the completion of processing a C block returns a comprehensive status to the user program.

Name	Description	Number of Parameter Pairs
S6NOOP	Simple block to test presence of S6PIO module	1
S6STAT	Block to get or clear S6PIO status counters	3
S6OPT	Read/Write options	3
S6SCAN	Master Scan interface	6
S6AOP	Master Scan analog output interface	6
S6AIN	Master Scan analog input interface	6
S6PSV	Passive Mode interface	6
S6WIN	DMA Windowing interface	4
S6XWIN	Expanded DMA Windowing interface	6
S6MOVE	Data Move interface	5
S6PMAP	Set or clear special card map for passive mode analog I/O demultiplexing.	4
S6EVTLG	Clear or retrieve event log, system error log or DMA headers	5

Table 4-1. C Block Summary

90-70 Interface

The C Blocks communicate with the S6PIO module via VME dual-ported memory.

Adding C Blocks to 90-70 Block Library

Before using a C Function Block it must be added to the 90-70 block library using the Program Block Librarian in the 90-70 Programming Package. To do this you must firstly ensure that the C Block is not read only. Then carry out the following:

Step 1 Go into 90-70 Programming Software

Step 2 Select F6, Program Block Librarian

Step 3 Select F6, Add Element to Library

Step 4 Specify the full path name of the EXE file supplied

Step 5 Specify the Element Type as External Block

Step 6 Press Enter to add the Block to the library

Step 7 Answer correctly to the number of input/output pairs and press enter. Refer to Table 4-1. C Block Summary, for the number of parameter pairs corresponding to each C Block.

Step 8 Press Y (yes) to accept the parameters entered

Step 9 Press ESC twice to return to main menu

Importing C Blocks to Program Folder

When the C block is required for use in a particular program folder it must be imported into the folder as detailed below:

Step 1 Go into 90-70 Programming Software

Step 2 Select F6, Program Block Librarian

Step 3 Select F3, Import Library Element to Folder

Step 4 Select the C Block as the library element

Step 5 Press Enter to import the block

Step 6 Press ESC twice to return to main menu

Bit Addressing

Most C Blocks take the Series Six I/O addresses in bit format. Valid bit numbers range from 1 to 8192. "Table 4-2. Bit Addressing Verses Channel Number" details the bit numbers for normal and expanded channel modes of operation.

Note

Only the ranges for I/O are listed in "Table 4-2. Bit Addressing Verses Channel Number"

Channel Number	Series Six Designation	Start Bit Number	End Bit Number
0	I/O0001-I/O1000	1	1000
1	I/O1+0001-I/O1+1000	1025	2024
2	I/O2+0001-I/O2+1000	2049	3048
3	I/O3+0001-I/O3+1000	3073	4072
4	I/O4+0001-I/O4+1000	4097	5096
5	I/O5+0001-I/O5+1000	5121	6120
6	I/O6+0001-I/O6+1000	6145	7144
7	I/O7+0001-I/O7+1000	7169	8168

Table 4-2. Bit Addressing Verses Channel Number

IO Status Register

Most C Blocks return an I/O status register as a part of their return data. This register consists of 16 bits with the following meanings (bits are numbered 1 to 16):

Bit	Mnemonic	Description
1	TIMEOUT	Command timeout
2	RESERVED	Always reads 0.
3	RESERVED	Always reads 0.
4	PARITY_RETRIES	Parity retries occurred during an I/O scan.
5	PARITY_FATAL	Parity retries exceeded during an I/O scan.
6	RESERVED	Always reads 0.
7	RESERVED	Always reads 0.
8	ISO_PS_BAD	Isolated power supply and/or interface is BAD
9	CHAIN_OK	Chain OK is BAD.
10	CHAIN_PARITY	Chain parity OK is BAD.
11	RST_ASSERTED	RST Asserted.
12	RESERVED	Always reads 0.
13	RESERVED	Always reads 0.
14	MODE_1	S6PIO mode bit 1
15	MODE_2	S6PIO mode bit 2 The two mode bits together indicate the following: (Mode_2 Mode_1) 00 Scanning in passive mode or manual mode 01 Slave mode or idle mode 10 Master I/O mode 11 Master DMA mode
16	PRIORITY_INPUT	Priority input mode

Table 4-3. I/O Status Register

Note

Reserved bits may be defined in future releases of software.

Scan Options

C Blocks that initiate I/O Scans accept a 16 bit scan options register with the following meanings (bits are numbered 1 to 16):

Bit	Mnemonic	Description
1	RESERVED	Set to zero for future compatibility.
2	PRIORITY INPUT MODE	If set, scan in priority input mode. Allows inputs to be read without updating outputs.
3-4	SCAN SPEED	Bit4=0,Bit3=0 default scan speed Bit4=0,Bit3=1 medium scan speed Bit4=1,Bit3=0 fast scan speed. Bit4=1,Bit3=1 very fast scan speed Note fast & very fast must not be used with Series Six I/O.
5	EXPANDED MODE	Assume expanded channel mode scanning.
6	USE_CP	Only copy input data for which a valid CP was received. Only for S6SCAN & S6AOP C Blocks.
7	PDT WINDOW	Issue PDT window after scan. Only for S6SCAN C Block.
8	FORCE IDLE MODE	Force idle mode on watchdog timeout. Only for S6SCAN, S6AOP & S6AIN C Blocks. Reverts To Idle Mode if the 90-70 trips or is selected to stop. This option should only be used for redundant configurations. As in normal configuration, S6PIO Assert Reset is required. It is also required for the Advanced I/O receiver to be in the first I/O rack so that Reset can be asserted when both redundant 90-70 racks trip
9	DMA FLOW THROUGH	Causes DMA data to be copied across to the corresponding 90/70 memory location. Applicable only to S6SCAN blocks with the PDT window option set.
10-16	RESERVED	Set to zero, may be used in future versions of software.

Table 4-4. Scan Options Register

Asynchronous Mode

Selected C Blocks support an asynchronous mode of operation. Currently supported asynchronous blocks are S6SCAN and S6PSV. Asynchronous operation is very useful when the 90-70 is driving multiple S6PIO cards as it allows the S6PIOS to scan or monitor their respective I/O chains at the same time, significantly reducing 90-70 CPU scan times.

When asynchronous mode is used, at least two blocks are required (i.e. two S6PSV blocks or two S6SCAN blocks) for each S6PIO. The first has the asynchronous bit set and initiates background collection and transfer of data in the S6PIO and is generally early in the program. A second block, generally at the end of the program, is used to ensure data collection and transfer has completed successfully and to trigger the transfer of the scanned input data (which has been buffered inside the S6PIO). The second block does not have the asynchronous bit set.

Emulated Series Six Memory

The S6PIO emulates Series Six memory locally onboard (refer to Series Six Table Descriptions for memory type descriptions). To devices on the Series Six Bus the S6PIO looks like a version 130 Microcode Series Six CPU with 32K Logic Memory and 16K Register memory and a CPUID of 1 (the CPUID is selectable, refer to "Setting the CPU ID" on page 24 of this chapter for details).

Note

DMA windows initiated by the S6PIO result in data transfers to and from the emulated Series Six memory, not the 90-70 memory tables. Refer to S6WIN C Block for details.

Error Codes

The C Block may return a number of error codes. Errors may be generated by the C Blocks themselves or by the S6PIO firmware during the execution of a command.

Errors that may be generated by the C blocks include:

Value	Hex Value	Description
0	0	Successful command completion.
1	1	A block parameter was missing.
3	3	An invalid data address was supplied to a C Block.
11	0B	The S6PIO is not present at the specified rack/slot number.
12	0C	An invalid rack address was specified.
13	0D	An invalid slot address was specified.
14	0E	There is a software version conflict between the C Block and the S6PIO.
15	0F	An error occurred when reading or writing the VME interface.
20	14	An invalid function code was specified.
21	15	A timeout occurred.
41	29	The defined series 6 I/O start address is out of the PLC memory range.
42	2A	The defined series 6 I/O range is out of the PLC memory range.
43	2B	The defined series 6 channel number is out of range.
44	2C	Size exceeds capacity of transfer buffer. The transfer buffer is a dual port memory buffer used for buffering data between the 90-70 and the S6PIO.
45	2D	Bad DMA header encountered when transferring data to the 90-70.

Table 4-5. C Block Error Codes

Error codes may also be generated by the S6PIO firmware, these include:

Val	Hex Val	Description
101	65	Passive mode failed to start properly.
102	66	Passive mode failed to stop properly.
103	67	Invalid mode command received by S6PIO
104	68	The specified command is not implemented
105	69	Low level I/O error present.

Val	Hex Val	Description
106	6A	Invalid channel number specified
107	6B	Errors occurred in high level I/O scan.
108	6C	Error in analog output scan.
109	6D	Error in analog input scan.
110	6E	Timeout waiting for PDT window in passive mode.
111	6F	Invalid command received.
112	70	Error changing channels in expanded mode.
113	71	General window error.
114	72	Window timeout
115	73	Window header checksum error.
116	74	Window header is invalid.
117	75	FPGA failed header done sequence.
118	76	DMA interrupt received incomplete header.
119	77	DMA interrupt didn't receive checksum.
120	78	Invalid memory type received.
121	79	Invalid special passive mode card type received.
122	7A	Reboot command failed
123	7B	Invalid register parameter address.
124	7C	Invalid register parameter size.
125	7D	Invalid event log number.

Table 4-6. S6PIO Firmware Error Codes

S6NOOP C Block

Interface

The S6SNOOP C Block provides a simple interface to the S6PIO module that performs No Operation. Use of this block provides for testing to determine whether a S6PIO module exists in a particular slot in the Series 90-70 rack and whether that S6PIO module is healthy.

The C Block consists of 1 pair of parameters as follows.

Parameter	Data Type	Comment
X1	Word	Rack in high 8 bits, slot in low 8 bits.
Y1	Word	Return error status

Table 4-7. S6NOOP Parameter List

The C Block asserts power flow if and only if there are no errors encountered.
If errors are encountered then the return error status register will identify the problem.

Example Usage

Refer to "Appendix B EXAMPLE 90-70 PROGRAMS" for an example use of the S6NOOP C block.

S6STAT C Block

Interface

The S6SSTAT C Block provides a means to set or clear various status counters within the S6PIO module.

The C Block consists of 3 pairs of parameters as follows.

Parameter	Data Type	Comment
X1	Word	Rack in high 8 bits, slot in low 8 bits.
X2	Word	Set to 0 to get status from S6PIO. Set to 1 to clear status in S6PIO
X3	Word	Dummy parameter, not used
Y1	Word	Return error status
Y2	Word	An array of 16 registers holding 8 32 bit status counters. The meaning of each counter is described below.
Y3	Word	Array of two registers containing software build numbers.

Table 4-8. S6STAT Parameter List

The C Block asserts power flow if and only if there are no errors encountered.
If errors are encountered then the return error status register will identify the problem.

The 32 bit status counters are defined as follows:

Register Offset	Data Type	Comment
0	DWORD	Total number of VME commands issued to S6PIO module. VME commands are issued to the S6PIO as a result of executing C Blocks, so normally this number will be incrementing at a rate of one or more counts per sweep.
2	DWORD	Total number of VME command errors. VME command errors arise when any C Block function causes a fatal error, e.g. parameters out of range and detected by the S6PIO firmware, or a fatal parity error or some other fatal I/O error condition.
4	DWORD	Total number of chain errors. Chain errors arise when an I/O cable is unplugged or an I/O rack is switched off.
6	DWORD	Total number of fatal parity errors. A fatal parity error arises when the preprogrammed number of parity retries is exceeded. The current scanning function is aborted.
8	DWORD	Total number of parity error retries. Parity retries are recoverable up to the retry limit specified. This counter counts the total number of retries, whether recoverable or not.
10	DWORD	Total number of analog input scan errors. These errors typically arise when an S6AIN block scans a set of addresses where one or more analog input cards are missing.
12	DWORD	Total number of DMA window errors. A DMA window

Register Offset	Data Type	Comment
		error can be caused by a checksum error, timeout or invalid DMA window header being received or by data under-run or overrun.
14	DWORD	Total number of expanded channel errors. The channel error count will increment each time a scan fails to change from one channel to another.

Table 4-9. Description of S6STAT Status Counters

Example Usage

Refer to “Appendix B EXAMPLE 90-70 PROGRAMS” for an example use of the S6STAT C block.

S6OPT C Block

Interface

The S6OPT block provides a means of reading or writing S6PIO option settings. Currently the only option parameter able to be set is the board watchdog timer.

S6OPT C block was created with 3 sets of inputs/outputs.

Parameter	Data Type	Comment
X1	Word	Rack in high 8 bits, slot in low 8 bits.
X2	Word	1 = Write option value 0 = Read option value (default)
X3	Word	Watchdog reset timeout value in milliseconds (only used if X2 = write) Valid range 50 – 1000 ms
Y1	Word	Return error status
Y2	Word	Returned options list, currently only the watchdog timeout value is returned
Y3	Word	Dummy value

This C block was created to read and write the S6 options (currently only watchdog timeout). It is recommended this C block only runs when needed i.e. not continually.

The watchdog time timeout, by default, is set to 270 milliseconds (valid range 50 to 1000 ms). This value closely approximates the Series Six watchdog timeout.

The watchdog monitors command activity from the 90-70 CPU. If a command is not received within the timeout period and the S6PIO is in master mode, RST will be asserted. There is an option within the S6SCAN block to command idle mode. If this option has been set, then the watchdog timeout will cause the S6PIO to go to the idle state. Refer to Table 4-4. Scan Options Register for further details.

S6SCAN C Block

Interface

The S6SCAN C Block provides an interface to the S6PIO module for use in Master Scan mode. This C Block provides the ability to scan in normal or expanded mode and can optionally issue a PDT window at the completion of the scan. A PDT window is required if Genius Bus controllers or Remote I/O Transmitters are to be supported, or if there is another S6PIO listening on the bus in passive mode. The S6SCAN C Block consists of 6 pairs of parameters as follows:

Parameter	Data Type	Comment
X1	Word	Bits 1-8 = slot number Bits 9-15 = rack number Bits 16 = scan type Bit 16 = 1 Initiate an asynchronous scan Bit 16 = 0 If asynchronous scan has been initiated then complete the scan, other wise carry out a normal scan
X2	Byte Array	Pointer to output table. This is the table within the 90-70 to be output to the Series Six I/O.
X3	Word	Start channel and address. The address is specified in standard Series Six bit notation. Refer to Table 4-2. Bit Addressing Verses Channel Number for details. This address refers to the FIRST bit in the scan.
X4	Word	End channel and address. The address is specified in standard Series Six bit notation. Refer to Table 4-2. Bit Addressing Verses Channel Number for details. This address refers to the LAST bit in the scan.
X5	Word	RST status. Bit1(first bit)=0 means RST de-asserted. RST is the Series Six Bus reset line and if asserted will cause the I/O modules to go to their default states.
X6	Word	Scan options, refer to Table 4-4. Scan Options Register for details.
Y1	Word	Return error status.
Y2	Byte Array	Pointer to input table. This is the table within the 90-70 to receive input data.
Y3	Word	I/O status register, refer to Table 4-3. I/O Status Register for details.
Y4	Word Array	Software build numbers. This is an array of 2 registers to receive the C block build number and the S6PIO firmware build number.
Y5	Word	Dummy parameter, not used
Y6	Word	Dummy parameter, not used

Table 4-10. S6SCAN Parameter List

Notes

The S6SCAN C Block asserts power flow only if there are no errors encountered.

If errors are encountered then the return error status register and the I/O status register will identify the problem. S6SCAN does not copy any input data if an error is encountered.

Parity retries are set to 1. There is no provision in S6SCAN to change the number of retries. If more than two consecutive I/O cycles to the same address have a parity error, the scan is aborted.

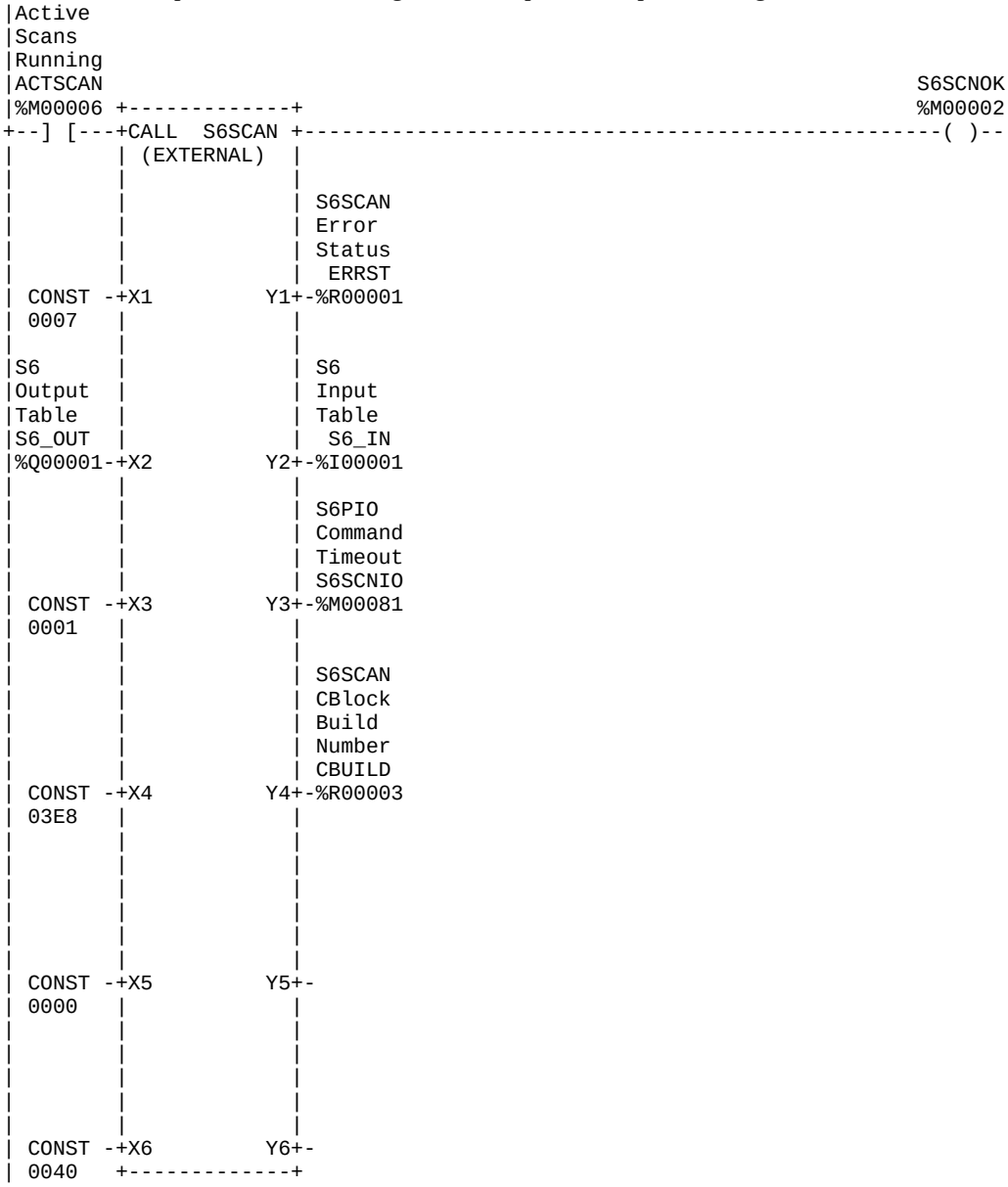
Use the priority input mode to scan inputs without updating outputs. This mode should be used for the first couple of scans after a power up until the input states have been read at least once. Refer to "Appendix B EXAMPLE 90-70 PROGRAMS" for an example on the use of priority input mode.

Use the RST status X5 to assert or de-assert RST. It is recommended that the state of RST follow the run/run-disabled/stop CPU keyswitch. Refer to "Appendix B EXAMPLE 90-70 PROGRAMS" for an example on the use of the CPU keyswitch.

Note the scan option "USE_CP", this option tells the block to copy input data only if the corresponding CP was received. This means that input data in the 90-70 with no corresponding physical input will not be disturbed.

Example Usage

The following example scans I/O from bits 1 to 1000 and issues a PDT window at the completion of the scan. Outputs are taken starting at %Q1, Inputs are copied starting at %I1.



S6AOP C Block

Introduction

The S6AOP C Block provides a simple interface to the S6PIO module for use in master scan mode to output data to analog output cards. The S6PIO module automatically scans and multiplexes the data to output to the four analog output channels on each card. The S6AOP C Block consists of 6 pairs of parameters as follows:

Parameter	Data Type	Comment
X1	Word	Rack in high 8 bits, slot in low 8 bits.
X2	Word Array	Pointer to analog output data. 4 registers per card
X3	Word	Start channel and address. The address is specified in standard Series Six bit notation. Refer to Table 4-2. Bit Addressing Verses Channel Number for details. This address refers to the FIRST bit in the scan. Note that a channel may NOT be crossed.
X4	Word	Number of analog output cards. Maximum 62 cards if starting at the beginning of a channel.
X5	Word	RST status. Bit1 (first bit)=0 means RST de-asserted. RST is the Series Six Bus reset line and if asserted will cause the I/O modules to go to their default states.
X6	Word	Scan options, refer to Table 4-4. Scan Options Register for details.
Y1	Word	Return error status
Y2	Byte Array	Pointer to input table. This is the table within the 90-70 to receive input data. Note that scanning analog output cards also results in input data being returned to the 90-70 PLC.
Y3	Word	I/O status register, refer to Table 4-3. I/O Status Register for details.
Y4	Word Array	Software build numbers. This is an array of 2 registers to receive the C block build number and the S6PIO firmware build number.
Y5	Word	Pointer to the output table. The S6PIO copies analog output bit data to the output table image. If this pointer is not NULL, the C Block copies the image data to this address. It is recommended that this pointer point to the area in 90-70 I/O memory corresponding to the analog output module address.
Y6	Word	Dummy parameter, not used

Table 4-11. S6AOP Parameter List

Notes

Keep in mind that scanning analog outputs also refreshes any inputs from input modules present at the same addresses.

Parity retries are set to 1. There is no provision in S6AOP to change the number of retries. If more than two consecutive I/O cycles to the same address have a parity error, the scan is aborted.

Use the priority-input mode to scan inputs without updating outputs. This mode should be used for the first couple of scans after a power up until the input states have been read at least once. Refer to “Appendix B EXAMPLE 90-70 PROGRAMS” for an example on the use of priority input mode.

Use the RST status X5 to assert or de-assert RST. It is recommended that the state of RST follow the run/run-disabled/stop CPU keyswitch. Refer to “Appendix B EXAMPLE 90-70 PROGRAMS” for an example on the use of the CPU keyswitch.

Note the scan option USE_CP, this option tells the block to copy input data only if the corresponding CP was received. This means that input data in the 90-70 with no corresponding physical input will not be disturbed.

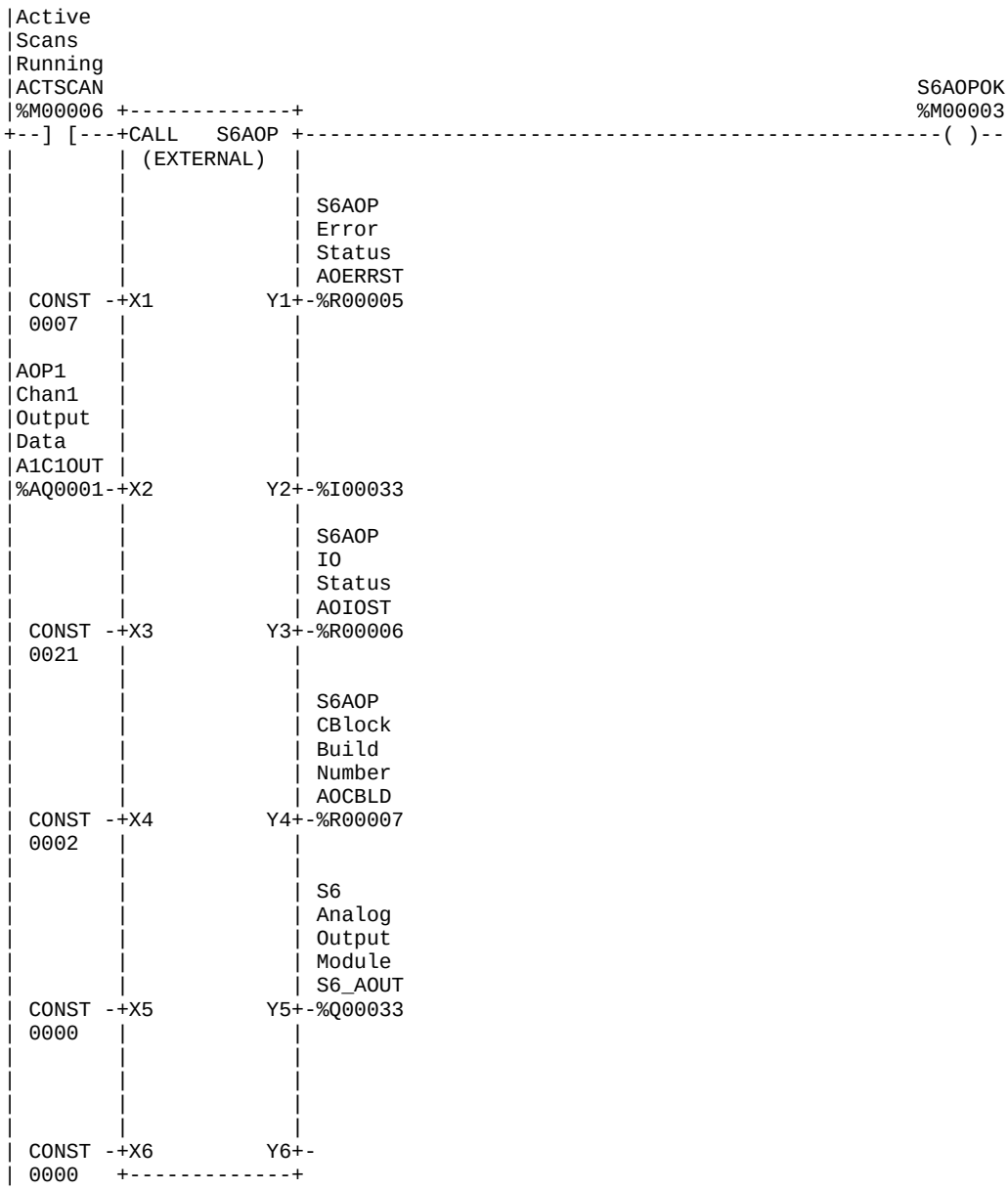
Also note that it is important to ensure that the output table parameter Y5 is correctly set as the S6SCAN block may also output data to the analog output cards if the S6SCAN blocks scanning address range includes the analog output address. If a program has a S6AOP block followed by a S6SCAN block then the S6SCAN block must be presented with appropriate data to output to the analog card, if the output cards address is included in the S6SCAN blocks scanning address range. Failure to do so may cause a subsequent S6SCAN to scan over the analog output card address range and to corrupt the analog output data, resulting in an erroneous output.

Example Usage

The following example scans 2 analog output cards beginning at address 33. Outputs are taken from 90-70 analog output registers %AQ1-%AQ8. The output data is copied to the output table at %Q33, input data is read into the input table at %I33.

Note

It is important to ensure that the output table parameter Y5 is correctly set. Failure to do so may cause a subsequent S6SCAN over the analog output card address range to corrupt the analog output data, resulting in an erroneous output.



S6AIN C Block

Introduction

The S6AIN C Block provides a simple interface to the S6PIO module for use in master scan mode to input data from analog input cards. The S6AIN C Block consists of 6 pairs of parameters as follows. The S6AIN C Block consists of 6 pairs of parameters as follows:

Parameter	Data Type	Comment
X1	Word	Rack in high 8 bits, slot in low 8 bits.
X2	Word Array	Pointer to 90-70 output data to be sent out during the analog input scan.
X3	Word	Start channel and address. The address is specified in standard Series Six bit notation. Refer to Table 4-2. Bit Addressing Verses Channel Number for details. This address refers to the FIRST bit in the scan. Note that a channel may NOT be crossed.
X4	Word	Number of analog input cards. Maximum 31 cards if starting at the beginning of a channel.
X5	Word	RST status. Bit1 (first bit)=0 means RST de-asserted. RST is the Series Six Bus reset line and if asserted will cause the I/O modules to go to their default states.
X6	Word	Scan options, refer to Table 4-4. Scan Options Register for details.
Y1	Word	Return error status
Y2	Byte array	Pointer to analog input data. This is arranged as a block of 8 16 bit registers per analog card.
Y3	Word	I/O status register, refer to Table 4-3. I/O Status Register for details.
Y4	Word	Pointer to analog input status data. 4 bits of status data per channel, packed 4 channels per 16 bit register. The meanings of the 4 bits are as follows: Bit1 – Set if board OK and valid data Bit2 – Set if out of range[1] Bit3 – Set if open wire[2] Bit4 – Sign bit[3]
Y5	Word Array	Software build numbers. This is an array of 2 registers to receive the C block build number and the S6PIO firmware build number.
Y6	Word	Pointer to the input table. The S6PIO copies analog input bit data to the input table image. If this pointer is not NULL, the C Block copies the image data to this address. It is recommended that this pointer point to the area in 90-70 I/O memory corresponding to the analog input module address.

Table 4-12. S6AIN Parameter List

[1] As of build 123 (previously under range)

[2] As of build 123 (previously over range)

[3] As of build 123 (previously open wire)

Notes

Keep in mind that scanning analog inputs also sends out output data to output modules at overlapping addresses.

Parity retries are set to 1. There is no provision in S6AIN to change the number of retries. If more than two consecutive I/O cycles to the same address have a parity error, the scan is aborted.

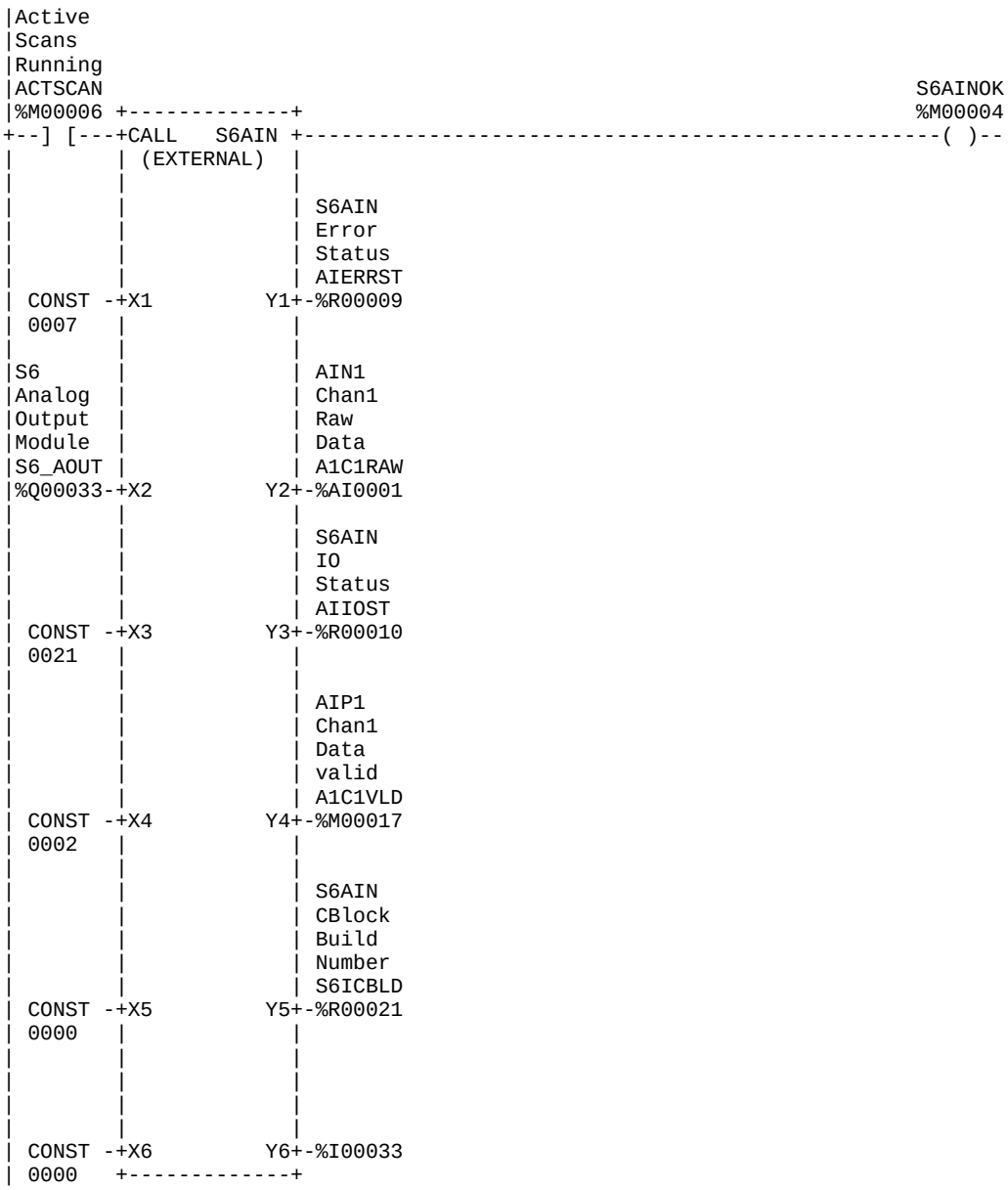
Use the priority input mode to scan inputs without updating outputs. This mode should be used for the first couple of scans after a power up until the input states have been read at least once. Refer to "Appendix B EXAMPLE 90-70 PROGRAMS" for an example on the use of priority input mode.

Use the RST status X5 to assert or de-assert RST. It is recommended that the state of RST follow the run/run-disabled/stop CPU keyswitch. Refer to "Appendix B EXAMPLE 90-70 PROGRAMS" for an example on the use of the CPU keyswitch.

The S6PIO is very particular about analog input card error conditions. The user program MUST CHECK the analog input status data for each analog input channel and take appropriate action if the Board OK/Data valid bit is reset. In the case where this bit is reset, the data cannot be considered to be valid, in fact it may very well have been set to zero by the S6PIO. The user should take appropriate action, for example preserving the last known good value. Refer to "Appendix B EXAMPLE 90-70 PROGRAMS" for an example on the use of the data valid bits for checking analog input data. Checking the range error bit is optional although, in conjunction with the scanned value, can indicate if the input is in limit (minimum or maximum). Care should be taken, however, when using the open wire error bit as it is defined only for 4-20mA current loop type analog cards and is undefined for other card types. For a 0-10V analog card, for example, the open wire bit will be set for some perfectly valid readings.

Example Usage

The following example scans 2 analog input cards beginning at address 33. Inputs are copied to 90-70 analog input registers %AI1-%AI8. The output data is copied to the output table at %Q33, input data is read into the input table at %I33. Analog input channel status is copied to the 4 registers beginning at %R21.


Active
Scans
Running
ACTSCAN
%M00006
CALL
(EXTERNAL)
S6AIN
S6AINOK
%M00004
S6AIN
Error
Status
AIERRST
CONST
0007
Y1
%R00009
S6
Analog
Output
Module
S6_AOUT
AIN1
Chan1
Raw
Data
A1C1RAW
%Q00033
X2
Y2
%AI0001
S6AIN
IO
Status
AIIOST
CONST
0021
X3
Y3
%R00010
AIP1
Chan1
Data
valid
A1C1VLD
CONST
0002
X4
Y4
%M00017
S6AIN
CBlock
Build
Number
S6ICBLD
CONST
0000
X5
Y5
%R00021
CONST
0000
X6
Y6
%I00033

S6PSV C Block

The S6PSV C Block provides an interface to the S6PIO module for use in passive mode to monitor a Series Six I/O Chain. The S6PSV C Block consists of 6 pairs of parameters as follows:

Parameter	Data Type	Comment
X1	Word	Bits 1-8 = slot number Bits 9-15 = rack number Bits 16 = scan type Bit 16 = 1 Initiate an asynchronous passive scan, issue the command to the S6PIO and return immediately. Bit 16 = 0 If asynchronous passive scan has been initiated then complete the scan, other wise carry out a normal scan
X2	Word	Synchronization wait timeout in milliseconds.
X3	Word	Mode and desired channel bitmap Bits 1&2 define the mode (bits are numbered 1 to 16), 0=stop, 1=run single channel, 2=run expanded channel mode. Bits 3-8 are set to 0. Bites 9-16 define the desired channels in expanded channel mode. These bits represent channels 0-7 respectively and if set, the data for that channel is copied to the next available buffer space. This feature allows the user to tailor exactly what expanded channel data is returned to the 90-70 program.
X4	Byte Array	Pointer to input table.
X5	Byte Array	Pointer to output table.
X6	Word	Synchronization address. This address is monitored by the S6PIO and when a bus cycle occurs to this address, the C block will return with the most current passive mode data. Usually this address is set to 0xFF, the PDT window address.
Y1	Word	Return error status.
Y2	Word Array	Card Present map. A map of CP status encountered since the last time the passive mode C block was executed. Consists of 1 bit per address, starting at address 0. The respective bit is set if a Card Present signal is received during the scan to that address. Card Present is a signal on the Series I/O Bus that is asserted whenever an input module is scanned. The CP map consists of 8 words per channel. When running the S6PSV block in expanded channel mode ensure that sufficient space is allocated for 8 expanded channels, ie 64 words.
Y3	Word Array	Address map. A map, similar to the CP map that indicates the addresses scanned since the last time the passive mode C block was executed. The Address map consists of 8 words per channel. When running the S6PSV block in expanded channel mode ensure that sufficient space is allocated for 8 expanded channels, ie

Parameter	Data Type	Comment
		64 words.
Y4	Word	I/O status register, refer to Table 4-3. I/O Status Register for details.
Y5	Word Array	Software build numbers. This is an array of 2 registers to receive the C block build number and the S6PIO firmware build number.
Y6	Word Array	Must point to a register area at least 3 registers long. Data written to this area is as follows: 16 bit channel map, bits 0-7 (first 8 bits) represent channels 0-7. (Bit set to 1 if the corresponding channel was scanned, else 0). 16 bit Synchronization counter, incremented each time the synchronization address is encountered. 16 bit size of special passive mode data. Following the size field is the special passive mode data. The user must ensure there is sufficient space here to hold all of the returned special mode data.

Table 4-13. S6PSV Parameter List

Notes

Passive mode continually monitors the I/O bus and copies address, output & input data information to memory. When a synchronization address is received, the data is copied to a second memory buffer for quick access by the S6PSV C Block. It is recommended that the PDT window address be used as the synchronization address. The reason for this is that the PDT window occurs at the end of the sweep period and provides a natural sweep synchronization point. The effect of synchronization is that the 90-70 and Series Six will generally synchronize to the same sweep time. (Depending on the work to be done by the 90-70, the 90-70 sweep time may be a multiple of the Series Six sweep time). The S6PSV block will return shortly after the synchronization address has been received (or the time out reached). Since the data in the S6PIO is double buffered, the data returned by S6PSV reflects the Series Six bus cycles that have occurred since the previous synchronization address.

Due to the need to synchronize to a Series Six sweep to ensure data coherency, it is necessary to use the S6PSV block in asynchronous mode when monitoring a Series Six with both primary and auxiliary chains. For this reason it is difficult for a 90-70 PLC with multiple S6PIOs to monitor multiple Series Six systems in passive mode. If it is necessary to perform this type of monitoring, the coherency of the Series Six data cannot be guaranteed. In this case the block should be used in asynchronous mode (refer X1 parameter) so that the different passive blocks for the different S6PIO cards can have their requests queued and therefore not block until a synchronization event has occurred. Refer to “Appendix B EXAMPLE 90-70 PROGRAMS” for an example program on how this is achieved.

When running passive mode on a Series Six I/O bus with a Series Six PLC performing a normal I/O scan, expect the address map to contain 125 consecutive bits indicating a scan of I/O 1-1000. It is normal to see the top three address bits not scanned. The address map provides useful information if a SUSIO instruction is programmed and one or more DOIO instructions are used to scan the I/O. In this case the address map shows exactly what addresses have been scanned between synchronization addresses.

The Card present map is useful to see a picture of what input addresses respond during the I/O scan between synchronization addresses.

When determining a suitable value for the synchronization time out period, keep in mind that the value should be longer than the normal Series Six sweep time. Indeed a general rule of thumb is to use a time 50% greater than the average Series Six sweep time. This allows time for DMA devices to complete, even when there is heavy data flow. For example if the average Series Six sweep time is 50 ms then the synchronization time out period should be set to 75 ms.

When using an S6PIO to monitor a Series Six I/O chain running in expanded channel mode, it is preferable to use the PDT window as the synchronization address. Any normal I/O address will be seen on a per channel basis and would not be desirable as a synchronization point.

Example Usage

Refer to “Appendix B EXAMPLE 90-70 PROGRAMS” for example uses of the S6PSV block. Examples of normal use of the block as well as asynchronous use are provided.

S6WIN C Block

Introduction

Smart Series Six I/O modules communicate by becoming bus masters on the Series Six I/O bus. The mechanism for doing this is termed a DMA Window. The S6PIO supports DMA windows and allows a particular addressed device to become a bus master and direct the flow of data between the device and emulated Series Six memory tables within the S6PIO. The DMA referred to here is termed Master DMA mode, in this case the S6PIO is normally the bus master and allows a slave device to become a bus master and "own" the bus for the duration of a window. The S6PIO does not support slave DMA mode.

The S6WIN C Block provides an interface to the S6PIO module for use in Master DMA mode to open a DMA window at a specific address. The S6WIN C Block consists of 4 pairs of parameters as follows:

Parameter	Data Type	Comment
X1	Word	Bits 1-8 = slot number Bits 9-14 = rack number Bit 15 set = use expanded channel mode Bit 16 = not used, should be 0
X2	Word	DMA address. The address is specified in standard Series Six bit notation. Refer to Table 4-2. Bit Addressing Verses Channel Number for details. Bit 16 set = DMA flow through
X3	Word	Command timeout in milliseconds. Specifies the maximum time the C Block will wait for a window to complete.
X4	Word	Window timeout in units of 111usecs. Specifies the window timeout. Normally a value of 45 should be used to give a timeout of 5 milliseconds.
Y1	Word	Return error status
Y2	Word	I/O status register, refer to Table 4-3. I/O Status Register for details.
Y3	Word	Number of window headers processed, including the close window header.
Y4	Word	Software build numbers. This is an array of 2 registers to receive the C block build number and the S6PIO firmware build number.

Table 4-14. S6WIN Parameter List

Notes

The window duration is limited by the timeout specified in X4. In reality the S6PIO will allow the timeout to extend a few milliseconds beyond the programmed timeout, provided that window headers continue to be received.

Each data transfer to or from the S6PIO is specified by a window header sent to the S6PIO by the smart device. The window header specifies the direction of transfer, the memory table and offset address and the length of the transfer. The window header is protected by a checksum byte. Following the window header the data transfer takes place.

A special window header is used to close the window. The window is also closed if the window timeout is exceeded without a new header being received.

The S6PIO stores the window headers and first few bytes of every data transfer in a DMA header log. This log is accessible via the console port and is sometimes useful when diagnosing DMA windowing problems.

Emulated Series Six Memory

The full complement of Series Six memory tables is emulated on the S6PIO module. The emulation is equivalent to that of a Series Six Plus running version 130 microcode. At reset time all memory tables are cleared to zero and the scratchpad initialized with data appropriate to a version 130 microcode CPU scratchpad. The CPU ID is set to 1.

The sizes of the emulated memory tables are as follows:

Memory Table	Size
Scratchpad	256 bytes
I/O Status Table	256 bytes
Transition Table	256 bytes
Override Table	256 bytes
Registers	16384 words
Logic Memory	32768 words

Table 4-15. S6WIN Emulated Memory Sizes

It is important to note that the emulated memory tables are distinct from the scanned I/O. This behavior is different to the Series Six and has been designed this way for maximum flexibility. The user is responsible for moving scanned I/O data to or from the emulated memory tables for access during a DMA window. (Refer to RTC technical notes for further details).

The user should also note that during a DMA window to an IOCCM or ASCII/Basic module, the command byte and status bytes are updated by the window, not by an I/O scan at the module address. Therefore to ensure that a module receives the correct command byte it must be moved into the appropriate status or register table before executing the DMA window. After the DMA window has completed the user may use a move command to move the DMA module status byte back to 90-70 memory.

Setting the CPU ID

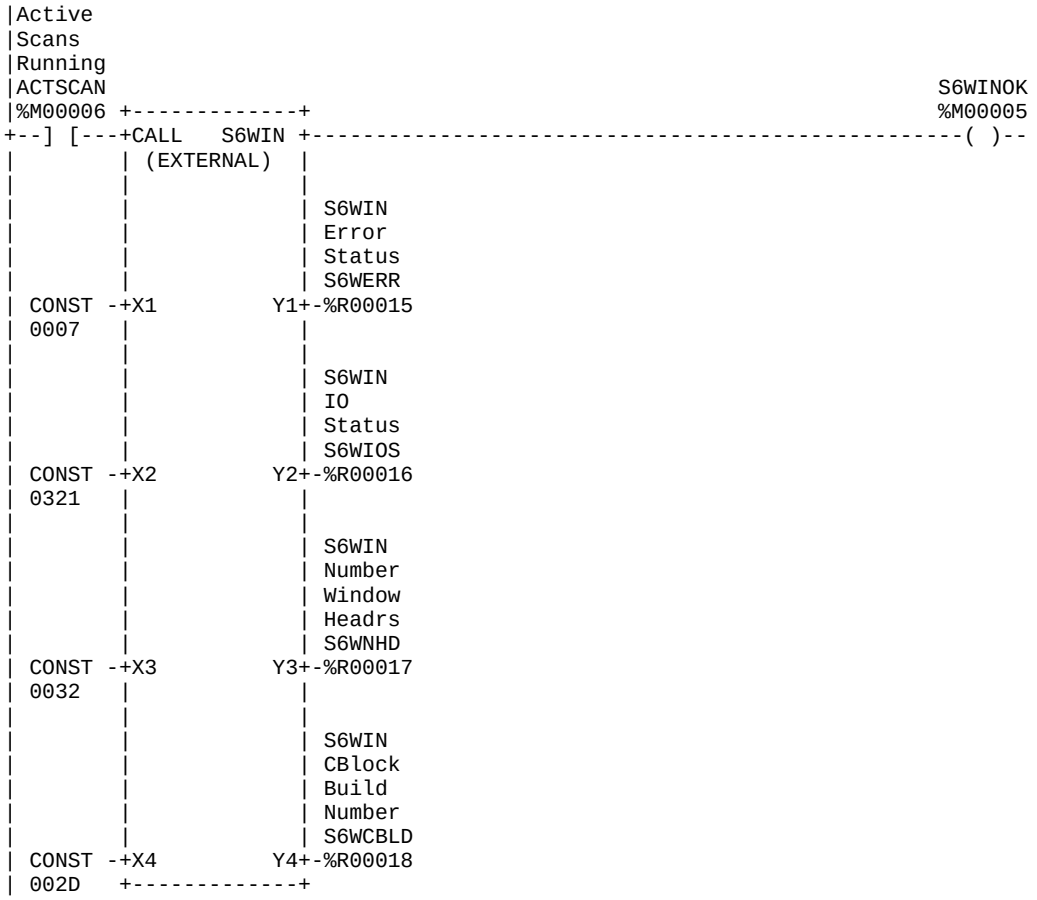
The emulated scratchpad defines a number of items, including the CPU ID. By default the CPU ID is set to 1 when the S6PIO module is reset. The CPU ID is read by an IOCCM module and is used in Master/Slave CCM communications. To change the CPU ID use an S6MOVE block to move a new CPU ID to S6PIO scratchpad byte location 0x16. This move only needs to be done once after the S6PIO module is initialized after a power up sequence.

Example Usage

The following example opens a DMA window to a module at bit address 801. The window has a 5 millisecond timeout, the overall C Block timeout is set at 50 milliseconds.

Note

A DMA window results in data transfers to and from emulated Series Six memory tables on the S6PIO module. This means that to get data in and out of the 90-70 a data move via an S6MOVE block is required. The example program in “Appendix B EXAMPLE 90-70 PROGRAMS” illustrates how to do this.



S6XWIN C Block

Introduction

The S6XWIN C Block provides an enhanced S6WIN interface to the S6PIO module for use in Master DMA mode to open a DMA window at a specific address.

The S6XWIN block has two additional pairs of parameters. These allow the user to specify a block of registers and a command byte to transfer to the S6PIO prior to opening the DMA window and a status byte to copy back after the window has closed. Since certain modules, such as the Genius Bus Controller (GBC) require access to the DPREQ or WINDOW instruction OPCODE in logic memory, the S6XWIN block instructs the S6PIO to transparently emulate the scratchpad and logic memory entries as required. This emulation saves the user from coding several S6MOVE blocks.

Parameter	Data Type	Comment
X1	Word	Bits 1-8 = slot number Bits 9-14 = rack number Bit 15 set = use expanded channel mode Bit 16 = not used, should be 0
X2	Word	DMA address. The address is specified in standard Series Six bit notation. Refer to Table 4-2. Bit Addressing Verses Channel Number for details. Bit 16 set = DMA flow through.
X3	Word	Command timeout in milliseconds. Specifies the maximum time the C Block will wait for a window to complete.
X4	Word	Window timeout in units of 111usecs. Specifies the window timeout. Normally a value of 45 should be used to give a timeout of 5 milliseconds.
X5	Word Array	2 words. 1] Register parameter block address in S6PIO register memory. 2] Number of registers to transfer to S6PIO emulated memory, up to 256 bytes (128 words) can be transferred.
X6	Word Array	Register parameter block in 90-70 register memory. Up to 128 registers (256 bytes) may be specified here. This block will be transferred to the location in S6PIO emulated S6 memory as given in X5
Y1	Word	Return error status
Y2	Word	I/O status register. Refer to Table 4-3. I/O Status Register for details.
Y3	Word	Number of window headers processed, including the close window header.
Y4	Word Array	Software build numbers. This is an array of 2 registers to receive the C block build number and the S6PIO firmware build number.
Y5	Word	Command byte to copy to S6PIO status table output command byte. Applicable to ASCII module
Y6	Word	Status byte location, for status byte copied from S6PIO

Parameter	Data Type	Comment
		status table input status byte. Applicable to ACII module and IOCCM.

Table 4-16. S6XWIN Parameter List

Example Usage

A detailed example on the use of the S6XWIN block with a GBC controller card is given in an applicable RTC application note. RTC application notes are available on the companion product CD or directly from RTC.

S6MOVE C Block

Introduction

The S6MOVE C Block provides an interface to the S6PIO module for use in moving data between the S6PIO (Series Six emulated memory) and 90-70. The S6MOVE C Block consists of 5 pairs of parameters as follows:

Parameter	Data Type	Comment
X1	Word	Rack in high 8 bits, slot in low 8 bits.
X2	Word	Source memory type
X3	Word	Source memory offset
X4	Word	Source memory ID, 0=S6PIO, 1=90-70
X5	Word	Transfer length in bytes.
Y1	Word	Return error status
Y2	Word	Target memory type
Y3	Word	Target memory offset
Y4	Word	Target memory ID, 0=S6PIO, 1=90-70
Y5	Word	Build number

Table 4-17. S6MOVE Parameter List

Memory types are specified according to the following table, depending on whether the memory is resident in the S6PIO or the 90-70. Note that the C block allows source and target to be the same type, i.e. memory can be moved from 90-70 to 90-70, 90-70 to S6PIO, S6PIO to 90-70 or S6PIO to S6PIO. Of particular importance for the S6MOVE block is the correct specification of the source and target memory offsets. For all 90-70 memory types, the source and target offsets are byte offsets. For Series Six memory source and target types, registers and logic memory are word offsets, all other memory types are byte offsets.

Memory Type	Value	Offset
90-70 %L Memory	0	Byte
90-70 %P Memory	4	Byte
90-70 %R Memory	8	Byte
90-70 %AI Memory	10	Byte
90-70 %AQ Memory	12	Byte
90-70 %I Memory	16	Byte
90-70 %Q Memory	18	Byte
90-70 %T Memory	20	Byte
90-70 %M Memory	22	Byte
90-70 %G Memory	56	Byte
90-70 %GA Memory	7	Byte
90-70 %GB Memory	9	Byte
90-70 %GC Memory	11	Byte
90-70 %GD Memory	13	Byte
90-70 %GE Memory	15	Byte
S6PIO Transition Table	1	Byte
S6PIO Override Table	2	Byte

Memory Type	Value	Offset
S6PIO Scratchpad	3	Byte
S6PIO I/O Status Table	4	Byte
S6PIO Registers	5	Word
S6PIO Logic Memory	6	Word

Table 4-18. S6MOVE Memory Types

Series Six Table Descriptions

The S6PIO emulates the Series Six memory tables while in master mode, however, these emulated tables generally do not serve their original function. For example, the emulated Series Six override table does not perform any override function when the S6PIO is in master mode (this is done by the 90-70 override table).

The following Series 6 descriptions are based on information provided in GEK-25379 Logicmaster TM 6 Programming and Documentation Software - User Manual (Section 12). The reader is encouraged to consult this reference if further information is required.

Transition Table:

The Transition table is 256 bytes long and is divided into two parts. The first 1024 bits (00->7F) of the table correspond in layout (one to one) to the output bits in the Output Status table. Counter and one-shot functions use these bits to store the state of the enabling contact string. With this information, the functions sense the OFF-to-ON transition that either enables counting or fires a one-shot.

The second half (80->FF) is used by the auxiliary output counter and one-shot functions with the advanced functions. These 1024 bits correspond in layout (one to one) to the auxiliary output table (which is of course the lowest 64 words (x16 bits) of the register memory).

Override Table:

The Override tables store override information for I/O references in the Main and Auxiliary I/O chains. The Override tables are used by the following functions: relays, counters, timers, latches, and one-shots. Mnemonic functions ignore the Override tables. When an input, output, or auxiliary I/O reference is overridden, it is maintained in the state (on or off) it was in when the override was applied. This state remains the same until the override is removed or its state is changed. Overridden bits can be changed by the mnemonic functions. References can be overridden, expanded Inputs and Outputs cannot be overridden.

The table is 256 bytes long with the first 1024 bits (00 to 7F) corresponding in layout (one to one) to the outputs and the second half (80 to FF) corresponding in layout (one to one) to the inputs.

I/O Status Table:

This table is 256 bytes long with the first 1024 bits (00 to 7F) corresponding in layout (one to one) to the outputs and the second half (80 to FF) corresponding in layout (one to one) to the inputs.

Scratchpad Memory:

This memory block emulates the memory used by the series six microcode and is 256 bytes long.

Logic Memory:

Logic memory normally stores the series six logic program. Logic memory is accessed by certain modules, such as the GBC, that require the DPREQ or WINDOW instruction opcode be present. Information in the opcode is used to enable the GBC to determine the location of its requisite parameter block.

Expanded CPU Operation.

The Series Six Plus PLC can operate in either normal or expanded mode and the S6PIO can be used in either situation.

If further information on expanded CPU operation is required, the user is encouraged to refer to GEK-96602 Series Six" Plus Programmable Logic Controller User's Manual, chapter 4.

Example Usage

Refer to "Appendix B EXAMPLE 90-70 PROGRAMS" for an example use of the S6MOVE C Block.

S6PMAP C Block

The S6PMAP C Block is an initialization block supporting the S6PSV block. It provides a means to set or clear the passive mode special card table in the S6PIO. The S6PIO uses this table to know what special cards are present in the I/O map so that it can demultiplex the data. Currently analog output and input cards are supported.

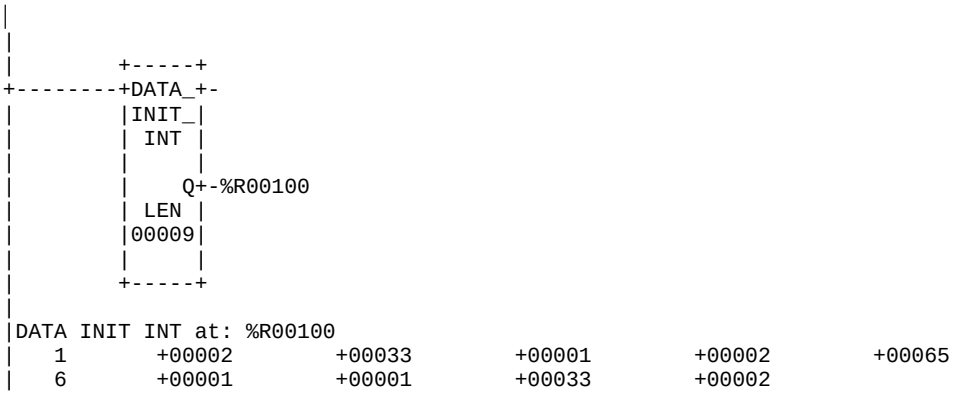
Parameter	Data Type	Comment
X1	Word	Rack in high 8 bits, slot in low 8 bits.
X2	Word	If =1 clear the special card map in the S6PIO before inserting the card map pointed to by X4. If = 0 just add to the existing special card map from the data pointed to by parameter X4.
X3	Word	Number of special card segments in the card map at X4.
X4	Word	The special card map. This consists of a set of segments, the number is defined by X3. For each segment there are three 16 bit parameters, the card type, 1=analog input, 2=analog output, the card bit address and the number of cards at consecutive addresses.
Y1	Word	Return error status
Y2	Word	16 bit word indicating the number of card entries in the special card map.
Y3	Word	16 bit word indicating the number of bytes of special card data that will be returned by a S6PMAP call.
Y4	Word	Software build numbers. This is an array of 2 registers to receive the C block build number and the S6PIO firmware build number.

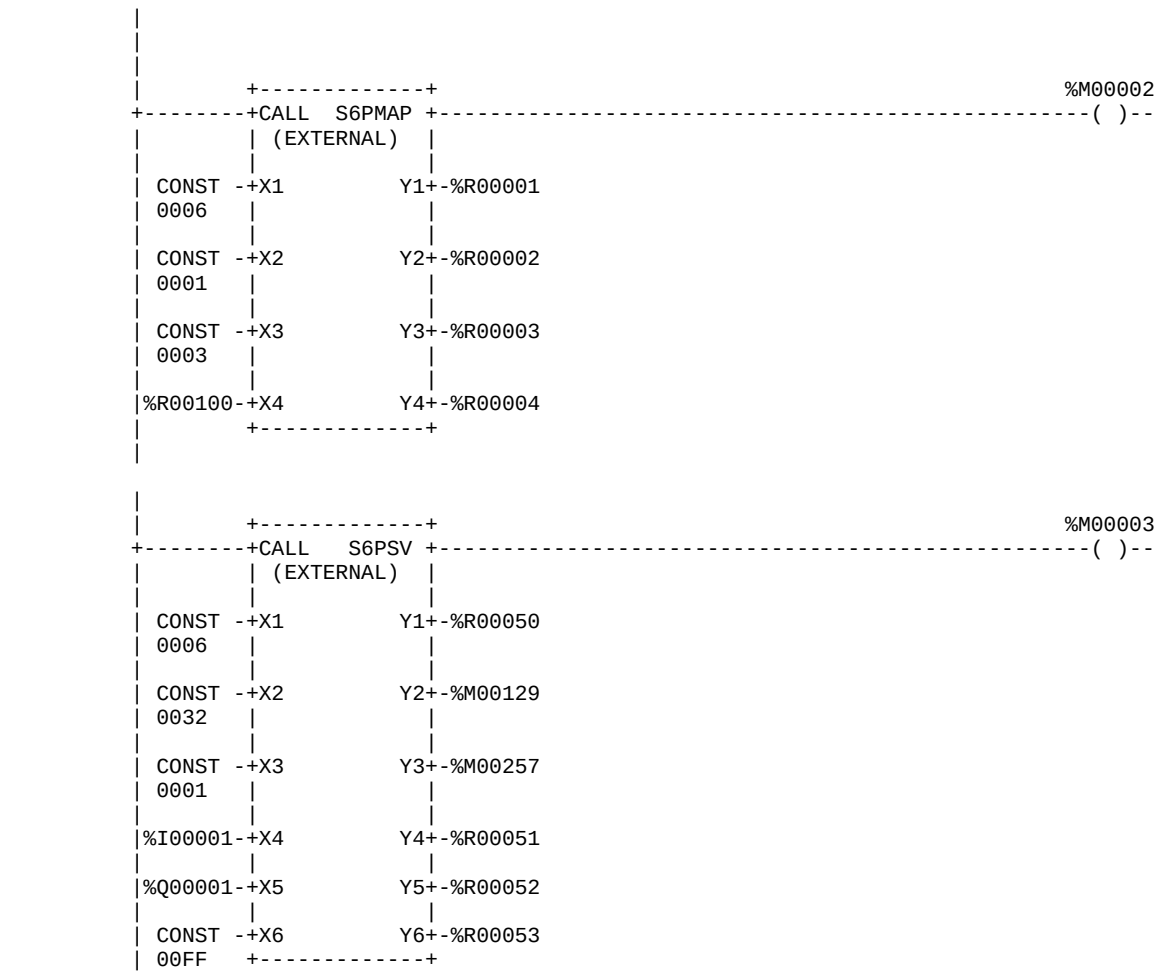
Table 4-19. S6PMAP Parameter List

Note that as the S6PMAP block is basically an initialization block it generally only needs to be run on startup. However, care should be taken when using “first scan” as this may be missed if the S6PIO is initialized after the 90-70 CPU. It is be better to use the S6NOOP block to detect a healthy S6PIO first and when a healthy S6PIO is initially detected then run any initialization blocks (such as the S6PMAP block)

Example Usage

The following example sets up a card map with an analog output at 33 (2,33,1), another analog output at 65 (2,65,1) and two analog input cards at 33 (1,33,2)





Refer to “Appendix B EXAMPLE 90-70 PROGRAMS” for further example use of the S6PMAP C Block.

S6EVTLG C Block

The S6EVTLG C Block provides an interface to the S6PIO event logs. The S6PIO module maintains three event logs, the system event log, the system error log and the DMA header log. The event logs are available using the BootLdr utility described in Chapter 7 UTILITY PROGRAMS. By using the S6EVTLG C Block, application code in the 90-70 may retrieve the event logs and present the information to higher level alarming and reporting systems to assist in routine system maintenance.

The S6EVTLG C Block provides the ability to clear the event logs and to retrieve the most recent information.

Parameter	Data Type	Comment
X1	Word	Rack in high 8 bits, slot in low 8 bits.
X2	Word	Log type. 0=DMA log, 1=system error log, 2=system event log
X3	Word	If =1, returns header only
X4	Word	Maximum number of log entries to return. This parameter effectively sizes the buffer to use at Y4. Maximum number of entries is limited by the buffer space available in the S6PIO transfer buffer. This is approximately 1024 bytes. The maximum number of events that can be returned is 21 for the system event log and 64 for the system error log and 42 for DMA header log. The most recent log entries are returned.
X5	Word	If =1 clear the event log. No data will be returned.
Y1	Word	Return error status
Y2	Word Array	The log header for the specified event log. Requires 16 registers to store the log header. Event Log No. – 16 bit number Next Event – 16 bit number Max Events – 32 bit number Total Events – 32 bit number Name of Event – 16 byte ASCII description Info Size – 16 bit number Padding – Not Used
Y3	Word	Number of log entries retrieved. May be less than the number asked for.
Y4	Word Array	Log data. Requires 32 bytes per entry for the DMA & system error logs and 96 bytes per entry for the system event log. Entries consist of the following: Milliseconds – 32 bit number Day number – 32 bit number Event number – 32 bit number Fault code – 32 bit number Data – 32 bytes for DMA log, 16 bytes for system error log and 80 bytes for system event log.
Y5	Word Array	Software build numbers. This is an array of 2 registers to receive the C block build number and the S6PIO firmware build number.

Table 4-20. S6EVTLG Parameter List

Note that care should be taken to ensure sufficient space is reserved for the output parameters, in particular 16 registers must be reserved for Y2 and sufficient space for the number of requested log entries should be reserved for Y4. The block will limit the number of entries that can be returned.

Event Log Header

The event log header specifies information about the event log, including its name, the pointers to the next free entry in the log and the number of events posted to the log.

Byte Offset	Data Type	Comment
0	WORD	Event log number, . 0=DMA log, 1=system error log, 2=system event log.
2	WORD	Index to next free event.
4	DWORD	Maximum number of events allowed.
8	DWORD	Total number of events posted to the log.
12	STRING	Name of the event log. 16 bytes long NULL terminated string
28	WORD	Size of the info part of the event log entry
30	WORD	Padding, set to zero. Aligns header size to 32 bytes.

Table 4-21. Event Log Header Layout

Common Event Log Information

Individual event log entries consist of common information such as a timestamp, log sequence number and fault number, followed by log specific information. The common log information occupies 16 bytes of memory per event log entry.

The common information is as follows:

Byte Offset	Data Type	Comment
0	DWORD	Milliseconds time. Either the milliseconds part of the day number since S6PIO reboot, or if the time has been set in the S6PIO module, the number of elapsed milliseconds since midnight.
4	DWORD	Day number. Either the number of whole days of uptime, or if the time has been set, the day number since 1st January 2000.
8	DWORD	Ever increasing event entry sequence number.
12	DWORD	Fault code. Specific to the log entry.

Table 4-22. Event Log Common Information Layout

Event Log Type

The three event log types serve different purposes in the S6PIO. They are described in the following sections.

DMA Log

The DMA log provides information about DMA window headers processed by the S6PIO module. DMA headers provide information about data transfers that have occurred in a DMA window and are useful for troubleshooting problems associated with DMA data transfers. The structure of a DMA log entry is as follows.

Byte Offset	Data Type	Comment
0	WORD	Byte count. Number of bytes following. Maximum is 30
2	WORD	Channel number and byte address[4]
4-8	BYTE ARRAY	5 byte DMA header.
9-33	BYTE ARRAY	DMA Data. Up to the first 25 bytes of the DMA data transfer are logged.

Table 4-23. DMA Log Layout

System Error Log

The system error log provides basic system event & error information. The system error log consists of the following entries.

The following fault codes are logged in the system error log.

Fault Code	Mnemonic	Description
1	CHAIN_GOOD	The CHAIN OK signal has transitioned from BAD to GOOD.
2	CHAIN_BAD	The CHAIN OK signal has transitioned from GOOD to BAD.
3	PARITY_ERR	An unrecoverable parity error has occurred. The log information field contains the channel number and byte address where the parity error occurred.
4	PARITY_RETRY	A recoverable parity error has occurred. The log information field contains the channel number and byte address where the parity retry occurred.
5	ISOPWR_GOOD	The interface across the S6PIO isolation barrier is good and the +5V isolated supply is OK.
6	ISOPWR_BAD	There has been a transition to a bad state across the isolation barrier.
7	RST_ASSERTED	The RST line on the IO bus has been asserted.
8	RST_DEASSERTED	The RST line on the IO bus has been de-asserted.
9	CHANNEL_FAIL	Indicates a failure in setting or changing channels
10	CHANNEL_SUCCESS	Successful channel change after a failed attempt

[4] From build 126 onwards

Fault Code	Mnemonic	Description
11	DMA_HDR_TIMEOUT	Indicates a timeout waiting for a header byte from the RXFIFO
12	DMA_CHECKSUM_ERR	Indicates checksum error in DMA window
13	DMA_CHECKSUM_TIME OUT	Time-out waiting for DMA checksum to arrive to indicate a window has been correctly received
14	DMA_INVLD_HDR	Invalid DMA header was received.

Table 4-24. System Error Log Fault Codes

Fault Codes 3 & 4, unrecoverable and recoverable parity errors, include the following information:

1st byte: Channel Number:- Indicates the channel number that the parity error occurred on, this shall be channel 0 for broadcast channel (0x80) or 0 – 7 if an expanded channel. The broadcast channel is indicated as channel zero.

2nd Byte:- Byte address location

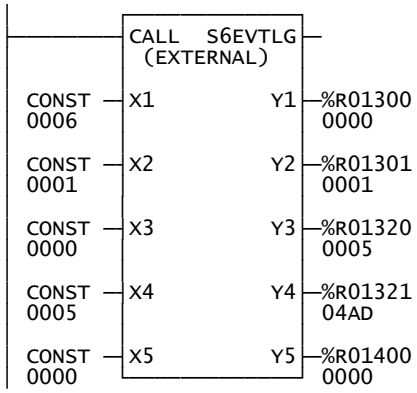
3rd Byte:- Bit Address:- Bit location on the bus 1 – 8192 ((Byte location x 8) + 1)

System Event Log

Unlike the previously described event logs which only provide binary data, the system event log provides general ASCII text messages. Each ASCII message is a NUL terminated string up to 80 bytes in size.

Example Usage

The example below illustrates an example of using the S6EVTLG to retrieve the Event Log data and storing it in a designated %R register. The same data can be retrieved using the bootldr utility from the console port.




```
BootLdr> evtrd 5 1
usEventIdx:             1
usNextEvent:            12
ulMaxEvents:            800
ulTotalNumEvents:       12
qname:                  SYS Error Log
usInfoSize:             16
```

```
Maximum entries = 30, buffer size = 966
00:01:06.733     7  8 Reset De-Asserted
00:01:06.734     8 10 Changed Channel Successfully
00:01:43.336     9  3 Parity Error Channel 0, byte 26, bit 0209
00:01:43.339    10  2 Chain OK Bad
00:01:43.340    11  7 Reset Asserted
```

Event header starts at location R1301 and Log data at R1321. The parity error data starts at location 1361 and has been displayed as hex for ease of reading.

```
                                    REGISTER
                    %R01311              00100000 01110010
01300       00000  00000  00000  00000  00000  00000  00000  00000  00000  00000
01310       28530  29253  08275  22867  00000  00012  00000  00800  00012  00001
01320       00005  00000  00000  00000  00000  00016  00000  00103  28492  08306

01330       00000  00000  00000  00008  00000  00007  00000  00001  00001  01197
01340       00000  00001  00001  01198  00000  00000  00000  00000  00000  00000
01350       00000  00000  00000  00000  00000  33023  00000  00010  00000  00008

01360       00000  00003  00000  00009  00000  00001  00001  37800  00000  00000
01370       00001  37803  00000  00000  00000  00000  00000  00000  00000   1A00
01380       00000  00000  00000  00000  00000  00002  00000  00010  00000  00001

01390       00000  00011  00000  00001  00001  37804  00000  00000  00000  00000
01400       00000  00000  00000  00000  00000  00000  00000  00000  00000  00007
01410       00000  00000  00000  00000  00000  00000  00000  00000  00000  00124
```

Chapter 5

SYSTEM DESIGN

Describes the requirements for good system design, including safety considerations and architecture issues. It is strongly recommended that the software and hardware are unit and system tested before implementation in a live environment.

Safety Considerations:

Warning

Of paramount importance in planning a changeover between a Series Six CPU and the combination of S6PIO and 90-70 PLC is consideration of safety. For this reason attention is drawn to the following sections which outline critical points which must be taken into account during the system design and implementation phase.

Priority Input Mode

Series Six PLC's have a Priority Input Mode which is automatically performed for the first scan when the CPU is selected to RUN mode. Priority Input Mode allows the PLC to read I/O chain inputs without updating any outputs, allowing the CPU to correctly solve logic prior to starting normal scanning where outputs are updated. The S6PIO module supports this feature via the S6SCAN C block. It is up to the user to add code in the 90-70 to select Priority Input Mode for the desired number of scans prior to changing to normal scan mode.

Watchdog Timeout

The S6PIO includes a watchdog timer that causes RST to be asserted on the IO BUS if commands are not received from the 90-70 PLC for a period exceeding the watchdog timeout setting (default = 270 milliseconds). There is an option to prevent the watchdog timer from asserting RST that is necessary when programming a redundant mode system. However if the S6PIO watchdog function is disabled, the user must ensure that equivalent functionality is provided downstream in the IO chain by using a suitably configured Advanced IO receiver Module.

Run/Disable

The S6PIO should follow the state of the 90-70 CPU Run/Disable keyswitch. Refer to Appendix B EXAMPLE 90-70 PROGRAMS for details on how to do this.

Redundant Systems

When designing a redundant system due attention must be paid to the possibility of both systems coming up in Solo mode. Application code must be added to detect IO bus activity and prevent the S6PIO master scan mode being programmed if such activity is detected. Failure to do this will mean that both 90-70 systems may attempt to control the I/O bus with consequent parity errors and misoperation.

Analog Inputs

When using the S6AIN C Blocks, the S6PIO is quite strict about analog input card status. It is therefore necessary to check the data valid bit of the analog input status prior to using any analog input data. Failure to do this may result in analog input data being momentarily set to zero if an error condition occurs.

I/O Bus Terminations

Caution

Care should be taken to ensure the I/O chain is not multiply terminated. If an external termination is used and the internal termination is also present then system mis-operation is likely to occur.

When running an S6PIO in a system where there may be cabling changes to allow for both passive and master mode operation there is often a need to change bus terminations. It is important that the correct terminations be applied to an S6PIO module. Failure to do so may cause parity errors or other misoperation. To make terminating the S6PIO simpler and avoid having to change onboard jumpers, the user may purchase a D37 terminator plug, RTC #J1320-P003. The onboard terminator jumpers may be removed and the plug used for quick change in termination status. A general rule of thumb is that if the S6PIO has an open connector, the terminations must be applied. If both connectors are used, e.g. for passive mode monitoring, the terminations must be removed.

Program Conversion

When converting Series Six programs to run in the 90-70 PLC there are a number of architecture considerations that must be taken into account. The most important considerations are the differences in memory architecture between the 90-70 and Series Six and the emulated Series Six memory facilities provided by the S6PIO.

S6 Emulated Memory

The S6PIO emulates Series Six memory. The emulated memory consists of the STATUS table of 1024 outputs and 1024 inputs, 16K words of register memory and 32K words logic memory. Associated transition tables for I/O are also supported.

The memory reserved for I/O scanning is distinct from the emulated memory. The purpose of this distinction is to provide as much flexibility as possible when designing systems.

Users must remember to transfer data between the 90-70 PLC and the emulated Series Six memory in the S6PIO. This is particularly important for applications involving smart devices that use DMA windowing.

Expanded S6 I/O

In the Series Six, expanded I/O overlays register memory. This must be considered when translating Series Six programs that make reference to expanded I/O.

Testing

Generally, the Series Six systems to be replaced by 90-70 plus S6PIO combination are highly complex systems. It is essential that a replacement system be given adequate unit and system testing before being brought online.

Chapter 6

STANDARDS & APPROVALS

The S6PIO meets the following standards:

	STANDARD	CONDITIONS
EMC Emissions		
Radiated/Conducted	EN55011 FCC Part 15	Class A, tested in a GE 90-70 integrator rack configuration.[5]
EMC Immunity		
ESD	IEC 1000-4-2	8KV Air, 4KV Conducted
RF Susceptibility	IEC 1000-4-3	10Vrms/m, 80MHz to 1000MHz, 80%AM [5]
Fast Transient Burst	IEC 1000-4-4	1KV I/O communications
Surge Withstand	IEC 1000-4-5	
Conducted RF	IEC 1000-4-6	10Vrms,150Khz to 80MHz, 80%AM [5]
Magnetic Immunity	IEC 1000-4-8	
Voltage Interruptions	IEC 1000-4-11	
Isolation		
Dielectric Withstand	IEC 950	1.5KV for modules rated from 51V to 250V
Power Supply		
N/A		
Environmental		
Vibration	IEC 68-2-6	1G@40-150Hz, 0.012in p-p@10-40Hz
Shock	IEC 68-2-27	15G, 11ms
Enclosure Protection		N/A
Safety		
Safety for industrial control equipment	IEC 950	Pollution degree 3, industrial environments

Table 6-1. Standards and Approvals

[5] General system configuration to be as per GFK-1179H

Chapter 7

UTILITY PROGRAMS

BootLdr Utility

Introduction

The bootldr utility is a Windows® console application that communicates with the S6PIO module through the front panel console port. The bootldr utility may be used to field upgrade the S6PIO module FLASH memory or to retrieve diagnostic information from the card.

Console Interface

To use the bootldr utility, connect a cable from the S6PIO module console port to the PC serial port. In the following examples it will be assumed that the PC serial port is COM1.
The console port pin out is compatible with the GE Fanuc Station Manager Cable IC693CBL316B, which may be used to connect to a standard PC DB9 serial port.
Console port communication speed is 57600 baud. This may present operational problems with older PCs.

RJ-11 Pin	Signal	RJ-11 Port Pin Description	DB9 Pin	Signal
1	CTS	Clear to send (i/p)	7	RTS
2	TD	Transmitted data (o/p)	2	RD
3	SG	Signal ground	5	SG
4	SG	Signal ground	5	SG
5	RD	Received data (i/p)	3	TD
6	RTS	Request to send (o/p)	8	CTS

Table 7-1. Console Connections

Invoking BootLdr

Invoke the bootldr utility by typing:

```
C:\> bootldr com1 s6pio
```

The bootldr utility will respond with a signon message and a prompt similar to the following:

```
RTC Boot Monitor version 1.28, May  9 2002
Entering boot monitor, type help for help
BootLdr>
```

BootLdr help Command

To get help on the available commands type:
BootLdr>**?**

The bootldr utility will respond with the following brief help on commands.

```
BootLdr> ?
Help
? or help                    Print this message
reset                        Reset the target board
boot [<baud>]                Boot a target processor (optional baud)
baud <baudrate>              Set new baudrate
exec                         Execute a procedure file
md <start> [<end>]           Memory display
mm <start addr>              Memory modify
ferase <device> <sector>|<sector-sector>|all Erase FLASH sector(s)
go <addr> <stack> [term] [baud] Execute from specified HEX address
upload <dev> <st> <end> <file>   Upload from target FLASH to binary file
rupload <st> <end> <file>  Upload from target RAM to binary file
download <dev> <st> <end> <file> [<file offset>] Download bin file to tgt FLASH
rdownload <st> <end> <file> [<file offset>] Download bin file to tgt RAM
verify <dev> <st> <end> <file> [<file offset>] Verify FLASH with binary file
rverify <st> <end> <file> [<file offset>] Verify RAM with binary file
term [baud rate]             Go to terminal mode, return to BootLdr with ^]
delay <msec>                 Delay for specified number of milliseconds
pkt [type]                   Set packet or simple protocol
evthdr [log id]              Read and display specified event log headers
evtrd [num entries] [log id] Read and display specified event log
evtclr [log id]              Clear the specified event log
evtdma 0|1                   Set simple or decoded mode for DMA header log
read_id                      Read module id structure
exit                         Exit BootLdr
BootLdr>
```

The above commands are the most frequently used. For help on additional commands used mainly for manufacturing testing type:

```
BootLdr> ? ?
```

BootLdr exit Command

The exit command exits the bootldr utility and returns to the DOS prompt.

BootLdr exec Command

The exec command allows a file of commands to be executed as a script. The exec command is most often used to invoke the upgrade of onboard FLASH memory. This is detailed in the section titled Upgrading Flash(below)

BootLdr read_id Command

This command displays the board version information. This is useful for confirming that the latest firmware is installed. The following is an example of the command's usage. Note that the firmware of the user's board should be equal to or later than that shown in the following example.

```
BootLdr> read_id
usModuleType:    1
usSwMajorRev:    1
usSwMinorRev:    0
usSwBuild:       127
BuildDate:       May 10 2002
BuildTime:       03:52:12
VersionString:   [S6PIO,0x00001000] version 1.00, build 127, May 10 2002 03:52:12
sFPGAModule:     XCS40
sFPGANCDFile:    xcs40.ncd
sFPGAPackage:    s40xlpq208
sFPGADate:       2002/05/09
sFPGATime:       11:11:16
usFPGA_OK:       OK
BootLdr>
```

BootLdr evtrd Command

The evtrd command allows the BootLdr utility to retrieve diagnostic information from the S6PIO. The evtrd command allows the user to look at the three event logs maintained by the S6PIO module:

- System Event Log;
- System Error Log
- DMA Header Log.

Each event is time-stamped with a time since boot, an event number, a priority field and a message field. By default, evtrd displays the last 50 entries in the System Event Log.

Examples of the use of the evtrd command are as follows:

DMA Header Log

The following is an excerpt from a DMA log generated using the boot loader command **evtrd 50 0**. It shows one complete DMA window for a GBC and IOCCM card as well as a PDT window.

```
00:20:44.349 1003058  0  9 | BR0961 [W:SP 00055-00056] 0000
00:20:44.349 1003059  0  9 | BR0961 [R:RM 00961-00961] 7017
00:20:44.350 1003060  0 19 | BR0961 [R:RM 00961-00966] 701705000100C30B40004D04
00:20:44.350 1003061  0  8 | BR0961 [R:TR 00000-00000] 00
00:20:44.350 1003062  0  9 | BR0961 [W:RM 00961-00961] 0000
00:20:44.351 1003063  0  8 | BR0961 [W: I00961-I00968] 80
00:20:44.351 1003064  0  9 | BR0961 [R:SP 00055-00056] 02F3
00:20:44.351 1003065  1  7 | BR0961 [CLOSE WINDOW]
00:20:44.353 1003066  0  9 | BR0513 [W:SP 00055-00056] 0000
00:20:44.353 1003067  0  9 | BR0513 [R:SP 00035-00036] 7FF9
00:20:44.354 1003068  0  9 | BR0513 [R:SP 00055-00056] 0178
00:20:44.354 1003069  0  9 | BR0513 [W:SP 00055-00056] 0000
00:20:44.354 1003070  0  9 | BR0513 [R:LM 07ff8-07ff8] 64E4
00:20:44.354 1003071  0  9 | BR0513 [R:SP 00055-00056] 0148
00:20:44.354 1003072  0  9 | BR0513 [W:SP 00055-00056] 0000
00:20:44.354 1003073  0 25 | BR0513 [R:RM 00102-00110] 0D0001001F006F0006001EA0D30086001F20
```

```
00:20:44.355 1003074  0  9 | BR0513 [R:SP 00055-00056] 02F8
00:20:44.355 1003075  0  9 | BR0513 [W:SP 00055-00056] 0000
00:20:44.355 1003076  0  9 | BR0513 [W:RM 00103-00103] 0100
00:20:44.355 1003077  0  9 | BR0513 [R:SP 00055-00056] 0001
00:20:44.356 1003078  1  7 | BR0513 [CLOSE WINDOW]
00:20:44.359 1003079  0 30 | PDT    [R:SP 00000-00016] 0303011F00007000000000FFFFFFFF0F000A0079000001
00:20:44.359 1003080  0 30 | PDT    [R:SP 00000-00016] 0303011F00007000000000FFFFFFFF0F000A0079000001
```

The data between the brackets […] or {…} is header data that has been decoded for the users convenience. This is the default action. If undecoded data is desired for low level debugging purposes, use the `evtdma` command to toggle between decoded and undecoded headers.

`evtdma 0`

Sets the header mode to undecoded.

`evtdma 1`

Sets the mode back to decoded.

The following fragment illustrates the format of the undecoded log.

```
23:57:02.422 290937588  1  7 | 80F8 0002008082
23:57:02.423 290937589  0  9 | 80C0 5510010066 0000
23:57:02.423 290937590  0  9 | 80C0 35100180C6 7FF9
23:57:02.423 290937591  0  9 | 80C0 55100180E6 0178
23:57:02.423 290937592  0  9 | 80C0 5510010066 0000
23:57:02.423 290937593  0  9 | 80C0 F8FF008077 64E4
23:57:02.424 290937594  0  9 | 80C0 55100180E6 0148
23:57:02.424 290937595  0  9 | 80C0 5510010066 0000
23:57:02.424 290937596  0 25 | 80C0 654008802D 0D0001001F006F0006001EA0D30086001F20
23:57:02.424 290937597  0  9 | 80C0 55100180E6 02F8
23:57:02.425 290937598  0  9 | 80C0 5510010066 0000
23:57:02.425 290937599  0  9 | 80C0 66400000A6 0100
23:57:02.439 290937604  0  9 | 80C0 35100180C6 7FF9
23:57:02.439 290937605  0  9 | 80C0 55100180E6 0178
23:57:02.439 290937606  0  9 | 80C0 5510010066 0000
23:57:02.439 290937607  0  9 | 80C0 F8FF008077 64E4
23:57:02.439 290937608  0  9 | 80C0 55100180E6 0148
23:57:02.439 290937609  0  9 | 80C0 5510010066 0000
23:57:02.439 290937610  0 25 | 80C0 654008802D 0D0001001F006F0006001EA0D30086001F20
23:57:02.440 290937611  0  9 | 80C0 55100180E6 02F8
23:57:02.440 290937612  0  9 | 80C0 5510010066 0000
23:57:02.440 290937613  0  9 | 80C0 66400000A6 0100
23:57:02.440 290937614  0  9 | 80C0 55100180E6 0001
23:57:02.440 290937615  1  7 | 80C0 0002000002
```

Using the following example entry from a DMA log:
`00:22:02.735 790203 0 9  | BR 961 [W:SP 00055-00056] 0000`
as an example, the data can be interpreted as follows:

Field	Description
00:22:02.735	Timestamp
790203	Sequence Number
0	Error Indicator Alternatives include: 0 = no error 1 = error
9	Bytes In Header
BR	Channel Alternatives include: BR = broadcast channel

	n+ = expanded channel n, where n = 0 to 7 PDT = PDT window
961	Card Address
[	Header Type Indicator. Header data between [and] has been decoded for the users convenience. Alternatives include: […] = version 1 header {…} = version 2 header
W	Data Direction Alternatives include: W = write by target device R = read by target device
SP	Memory Type Alternatives Include: SP = scratchpad memory (byte addressed in hex) LM = logic memory (word addressed in hex) RM = register memory (word addressed in decimal) TR = transition table (bit addresses) OV = override table (bit addressed) <space> = I/O status memory (bit addressed)
55-56	Data Address Range This can be byte word or bit address range, depending on memory type (refer Memory Type above) In this example, the address range given is 55 to 56 in scratch pad memory, hence they are byte addresses.
0000	Data Presented in hex. In this example 00 is being written to scratch pad memory 55 and 00 is being written to scratch pad memory 56

Table 7-1. Anatomy of a DMA Header Log

System Error Log

```
BootLdr> evtrd 20 1
usEventIdx:           1
usNextEvent:          6
ulMaxEvents:          800
ulTotalNumEvents:     6
qname:                SYS Error Log
usInfoSize:           16
Maximum entries = 30, buffer size = 966
00:00:02.250 0 1 Chain OK Good
00:00:02.251 1 5 Isolated Power OK Good
00:00:02.254 2 8 Reset De-Asserted
00:00:02.255 3 10 Failed to change I/O channels
00:02:22.399 4 7 Reset Asserted
00:07:48.150 5 8 Reset De-Asserted
```

Refer to Figure 7-1. Anatomy of an Error Log Entry for details on the layout of the error log.

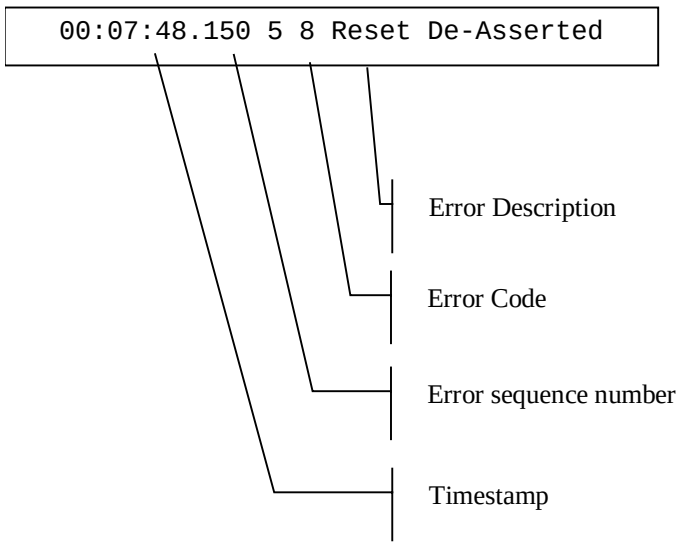


Figure 7-1. Anatomy of an Error Log Entry

System Event Log

The following is a typical dump from the system event log.

```
BootLdr> evtrd 50 2
usEventIdx:         2
usNextEvent:        26
ulMaxEvents:        200
ulTotalNumEvents:   26
qname:              Event Log
usInfoSize:         80
Maximum entries = 9, buffer size = 870
00:00:00.103     0  0 Event system initialised
00:00:00.593     1  0 [S6PIO,0x00001000] version 1.00, build 127, May 10 2002 03:52:12
00:00:00.595     2  0 Testing for modules, start = 0x00001000, size = 0x0001F000
00:00:00.844     3  0 Module S6PIO, size 70308, type CODE, addr 0x00001000
00:00:00.991     4  0 Module XCS40, size 41436, type FPGA_XILINX, addr 0x000122B0
00:00:01.034     5  0 Module XC9572, size 11665, type XSVF, addr 0x0001C490
00:00:01.037     6  0 Module ERASE, size 316, type XSVF, addr 0x0001F230
00:00:01.040     7  0 Module COPYRIGHT, size 422, type TEXT, addr 0x0001F370
00:00:01.055     8  0 Writing XCS40: xcs40.ncd, s40xlpq208, 2002/05/09, 11:11:16
00:00:01.637     9  0 FPGA Config complete
00:00:01.638    10  0 Configuration mode = MODE_PLUG_AND_PLAY
00:00:01.640    11  0 VME: slot = 6, slave = 0x00008000, AM = 0x29, Enable = TRUE
00:00:01.663    12  0 DPRAM Request:  0x00400060
00:00:01.664    13  0 DPRAM Response: 0x00400080
00:00:01.666    14  0 DPRAM CP:       0x004000B0
00:00:01.667    15  0 DPRAM AM:       0x00400130
00:00:01.668    16  0 DPRAM Inputs:   0x004001B0
00:00:01.669    17  0 DPRAM Outputs:  0x004005B0
00:00:01.670    18  0 DPRAM XfrBuff:  0x004009B0, size 0x1000
00:00:01.672    19  0 DPRAM pucChIdx: 0x004019B0
00:00:01.673    20  0 DPRAM mem_used: 0x19C0, free space: 1584 bytes
00:00:01.678    21  0 Chain OK Bad
00:00:01.679    22  0 Isolated Power OK Good
00:00:01.680    23  0 Chain OK Good
00:00:01.682    24  0 RST De-Asserted
00:00:01.683    25  0 IO successful change from Ch 255 to Ch 128. Total Chan errs 0
BootLdr>
```

Note

The system event log is output on the second serial port in ASCII format. Data rate 57600 baud.

BootLdr Boot Command

The boot command allows the BootLdr utility to capture control of the S6PIO and force it into a diagnostic monitor mode. When the S6PIO is in this mode all front panel LEDs are off, apart from ACTIVE LED which flashes at approximately 4Hz rate. A typical dialog is as follows:

```
BootLdr> boot
Setting baudrate to 57600 baud
Reset target processor, hit any key to abort
Valid leadin 'RTES,BOOT,S6PIO,0.00.6' received
BootLdr>
```

Note that the display pauses after “Reset target processor, hit any key to abort” line, waiting for the user to press the reset button on the board (or power cycle the 90-70 rack off-on). Alternatively, the user can press any key to abort the boot command.

Upgrading Flash

To upgrade FLASH memory perform the following procedure:

1. Unprotect FLASH memory by moving the Jumper JP4 to the 2-3 position.
2. Connect the console cable between the S6PIO console port and COM1 on the PC.
3. Start the bootldr utility
4. At the bootldr prompt type the “boot” command.
5. Press the reset button or power cycle the board.
6. When boot mode is entered, type the exec command to initiate the FLASH download. The file s6pio_f.bin is assumed to contain the FLASH binary image.
7. When the download is complete protect the FLASH by moving jumper JP4 to the 1-2 position.

An example FLASH download session follows.

```
BootLdr> boot
Setting baudrate to 57600 baud
Reset target processor, hit any key to abort
Valid leadin 'RTES,BOOT,S6PIO,0.00.6' received
BootLdr> exec prog_flash.txt
BootLdr> rdownload ffe000 fff7ff bootmon.bin
Starting download... press any key to abort
0x00FFF780
Download from file bootmon.bin completed
BootLdr> go ffe018 0
Executing from address FFE018
BootLdr> delay 100
BootLdr> ferase 0 4-9
Erasing sector 4
Erasing sector 5
Erasing sector 6
```

```
Erasing sector 7
Erasing sector 8
Erasing sector 9
BootLdr> download 0 1000 1ffff s6pio_f.bin
Starting download... press any key to abort
0x0001FF80
Download from file s6pio_f.bin completed
BootLdr> exit
```

User Configuration Mode

In this mode the S6PIO uses configuration data that has been written to FLASH memory. This mode is not supported in a 90-70 PLC but is supported for S6PIO use in a standard VME system as well as Innovation controllers.
The use may specify A16 or A24 addressing using address modifiers 0x29, 0x2D, 0x39 or 0x3D. No other address modifiers are supported. Note that the S6PIO will respond only to the programmed address modifier. It cannot respond to multiple address modifier codes.
The S6PIO occupies an 8K memory window. In either A16 or A24 mode the address may be set on any 8K boundary within the 64K or 16M address space.
The S6PIO as shipped has a blank VME configuration. This means that if the card is installed in user config mode the configuration will fail. The mode will flash the ACTIVE LED at approximately 1 second period and turn all other LEDs off to indicate that it has entered console mode. If the S6PIO has already been configured with a valid VME configuration, use JP8 to set the S6PIO to boot into the command line monitor.
Connect a serial terminal emulator to the console port. Setting should be 57,600 baud, 8 bits no parity. Terminal emulator may be an application like Hyperterminal or it could be the BootLdr application shipped with the S6PIO. If using the bootldr application use the following command line to start the bootldr and enter terminal mode.

```
C:\>bootldr com1 s6pio
BootLdr>
BootLdr> term
Entering terminal mode, exit with ^]

S6PIO>
```

Use the config command to set the desired parameters into FLASH. Before doing so ensure that the FLASH is write enabled via JP4.

```
S6PIO> config
No valid configuration found
S6PIO> ?
?            display brief help
help         display brief help
exit         exit from monitor
md           byte memory dump
mdw          word memory dump
mm           byte memory modify
mmw          word memory modify
byte         set byte
word         set word
modbld       build the module directory
exec         execute a text module as a sequence of commands
evt          display the event log
fpga_reset   reset the fpga
```

```
dl_xcs       download the xcs40 fpga
epld_dwnld   download the epld
cmdhdlr      go to command handler
date         display date & time
setime       set date & time
mdir         display the module directory
dump         dump a module in binary format
vme          vme address vme addr modifier
config       config vme address vme addr modifier vme enable
S6PIO>
S6PIO> config 200000 3d 1
No valid configuration found
Writing configuration data...
Configuration data written
VME Slave = 0x200000, VME AM = 0x3D, VME Enable = TRUE
S6PIO>
S6PIO> config
VME Slave = 0x200000, VME AM = 0x3D, VME Enable = TRUE
S6PIO>
```

Chapter 8

TROUBLESHOOTING

The following section may help in troubleshooting problems.

Troubleshooting Strategy

When the system does not operate as expected it is useful to check the following items:

1. *S6PIO Front Panel LEDs*.

   During power-on initialization, all of the LEDs will cycle twice through an LED test routine before being set to their standard states (refer to Status LEDS for a description of each LED). If an error occurs during power on the Active LED will flash rapidly and the remaining LEDS will display an error code. If this condition is encountered the board has become unserviceable. If, however, the active LED is flashing slowly, then it is possible that the board has been jumpered to "Config from FLASH" but there is no configuration data in the FLASH memory. In this case, ensure the board is jumpered correctly for the mode of operation expected (refer Jumper Settings) and, if required, that the FLASH memory contains configuration data (refer User Configuration Mode).

   The following diagrams illustrate the state of the front panel LEDs after power on, under certain operating conditions.

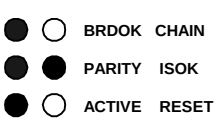
○ = OFF
● = ON
◌ = FLASH
Key

Normal Operation - Master mode

In this mode all LEDs should be on

Normal Operation - Passive mode.

The Reset (RST) LED should be off. The parity LED may flicker if DMA headers are being received.



Fault - Break in I/O Chain (passive mode)

The Chain LED is off, check I/O chain connections.

Fault - Break in I/O Chain (master mode)

Check I/O chain connections

If the Active LED is permanently off it is probable that the 90-70 CPU is halted or there is no 90-70 application loaded that is communicating with the S6PIO (i.e there are no S6PIO C blocks being run)

2. *Cable Connections.*
   Check all connections to ensure they have been firmly made.

3. *Terminations.*
   Check all I/O chain terminations. Make sure the S6PIO is not double terminated (i.e. make sure that the internal and external terminators have NOT both been applied).

4. *C-block error returns.*
   Often the C block error returns indicate what the problem is. It is highly recommended that the error return values be monitored and appropriate action taken. Refer to Trouble Shooting C-Block Error Returns (below) for possible actions.

5. *Event and Error Logs.*
   The event and error logs can provide additional information that may allow the user to pin point the area providing trouble. Refer to BootLdr evtrd Command (above) for information on how to read the event and error logs.

6. *DMA headers.*
   Check the DMA headers to ensure the information interchange is as expected. The DMA headers provide useful information, refer to Table 7-1. Anatomy of a DMA Header Log for an example on how the data is interpreted. The event log can also be read using the S6EVTLOG C block, block (refer Chapter 4 SOFTWARE INTERFACE for more detail).

7. *Software Version.*

Ensure the firmware revision and C-block revisions are the latest revision and are compatible. Many of the C-blocks return a software build number. Check this return to ensure you have the latest C-blocks installed. Also check the firmware revision, this can be done by running the ReadID command in BootLdr, as shown below. Note that the version data is provided for firmware and the FPGA. It may be necessary to quote this data in the event of a technical support call to the distributors of the board.

```
BootLdr> read_id
usModuleType:    1
usSwMajorRev:    1
usSwMinorRev:    0
usSwBuild:       127
BuildDate:       May 10 2002
BuildTime:       03:52:12
VersionString:   [S6PIO,0x00001000] version 1.00, build 127, May 10 2002 03:52:12
sFPGAModule:     XCS40
sFPGANCDFile:    xcs40.ncd
sFPGAPackage:    s40xlpq208
sFPGADate:       2002/05/09
sFPGATime:       11:11:16
usFPGA_OK:       OK
BootLdr>
```

8. *System Design.*
   Refer to Chapter 5 SYSTEM DESIGN to help determine if the problem is a system design issue.

Trouble Shooting C-Block Error Returns

Most of the 90-70 C blocks provide a standard error return number to give some feedback on what the mis-operation is likely to be. Table 4-5. C Block Error Codes lists the possible error code returns. This section lists some of the possible causes for particular errors and suggests some actions that may be taken to avoid the error.

Error Code 1 A block parameter was missing.
Possible Cause(s):
The user has failed to provide an essential parameter to the block.
Corrective Action(s):
Check the data sheet / user manual for a description of each of the parameters of the C block returning the error. Ensure all parameters have been correctly applied.

Error Code 3 An invalid data address was supplied to a C Block.
Possible Cause(s):
The user's program has provided an invalid address as one of the parameters to the C block.
Corrective Action(s):
Ensure the addresses being supplied exist and are of the appropriate type.
Error Code 11 The S6PIO is not present at the specified rack/slot number.
Possible Cause(s):
1. An incorrect rack/slot parameter value was provided to the C block.
2. S6PIO not detected as present on the slot / rack.
Corrective Action(s):
1. Make sure the correct slot / rack number is provided to the C block.
2. Possible hardware failure, run the board under self-test (refer Self-test Mode) to ensure board is operating correctly. If self test passes, check the 90-70 rack to ensure it is operating correctly.

Error Code 12 An invalid rack address was specified.
Possible Cause(s):
The rack number supplied is incorrect or invalid, possibly because the rack does not exist or is not accessible. Only rack addresses 0 to 7 are valid.
Corrective Action(s):
Enter the correct rack number.
Make sure the rack is properly configured and is accessible.

Error Code 13 An invalid slot address was specified.
Possible Cause(s):
The slot number is invalid. Only slots 2 to 9 are valid.
Corrective Action(s):
Ensure a valid slot number is entered as a parameter to the C block.

Error Code 14 There is a software version conflict between the C Block and the S6PIO.
Possible Cause(s):
Both the S6PIO firmware and the C blocks have their own version or build numbers. Generally these numbers should be in agreement (indicating the blocks and the firmware are from the same release). If the version numbers are in conflict it means either the firmware or the C block was updated without the other component being updated.
Corrective Action(s):

Update the out of date component. Refer to Adding C Blocks to 90-70 Block Library for details on adding C blocks to the block library. Refer to Upgrading Flash for details on upgrading the S6PIO firmware.

Error Code 15 An error occurred when reading or writing the VME interface.

Possible Cause(s):

The 90-70 encountered an error while trying to communicate across the back plane to the S6PIO.

Corrective Action(s):

Ensure the S6PIO is properly seated in the card rack and that the S6PIO is free of hardware faults.

Error Code 20 An invalid function code was specified.

Possible Cause(s):

A C block has asked the firmware to perform an unknown task. This can occur when a new C block with new functionality is run on a system which has old S6PIO boards with earlier versions of the firmware. The firmware in this case does not know how to deal with the new functionality.

Corrective Action(s):

Update the firmware to the latest version.

Error Code 21 A timeout occurred.

Possible Cause(s):

A time limit was exceeded, usually resulting from an expected event failing to occur within a set time. For example the S6PSV block waits for t milliseconds for a synchronization address to appear on the I/O chain (where t is set via the X2 parameter of the block). The synchronization address is usually 0xFF (the PDT window). A timeout in this case means the PDT window failed to appear within t milliseconds.

Corrective Action(s):

Ensure the timeout period is appropriately set. As a rule of thumb for the passive block the timeout should be about 50% greater than the average Series Six sweep time. If the timeout period is reasonable, but timeout errors still occur, check the user application program to ensure the synchronization address is occurring as expected.

Error Code 41 The defined series 6 I/O start address is out of the PLC memory range.

Possible Cause(s):

The PLC does not physically contain the I/O address specified.

Corrective Action(s):

Ensure the parameter specifying the I/O address refers to a physical address.

Error Code 42 The defined series 6 I/O range is out of the PLC memory range.

Possible Cause(s):

The address range provided means that when added to the start address some or all of the later addresses do not physically exist.

Corrective Action(s):

Make sure the address range specified does not put the later addresses out side the usable range.

Error Code 43 The defined series 6 channel number is out of range.

Possible Cause(s):

The specified channel number is outside the range 0 to 7.

Corrective Action(s):

Ensure the channel number is inside the range 0 to 7.

Error Code 44 Size exceeds capacity of transfer buffer.

Possible Cause(s):

The transfer buffer is a dual port memory buffer used for buffering data between the 90-70 and the S6PIO. In this case the amount of data requested to be transferred is greater than this buffer can hold.

Corrective Action(s):

Transfer less data per block. This can be achieved by splitting the transfer across two blocks.

Error Code 45 Bad DMA header encountered when transferring data to the 90-70.

Possible Cause(s):

A DMA header failed to decode to a valid type or valid address, or header checksum errors were encountered. This may occur if the I/O chain is unreliable, either through poor connections or high levels of induced noise. Exceeding cable length limits may also cause this problem. In particular, Genius I/O on very long cables can also generate this problem.

Corrective Action(s):

Refer to the Series Six installation manual to details on the proper installation of the Series Six equipment. If possible, reduce I/O chain cable lengths.

Self-test Mode

If hardware problems are suspected then the S6PIO module may be placed in Self-test mode. In Self-test mode, continuous tests are performed on the FPGA, DPRAM, SRAM and the isolated Series Six interface. Before running the Self-test, ensure that all jumpers (other than JP5 & JP6) are in the factory default positions. The Self-test function is invoked by placing both JP5 and JP6 in the 1-2 position (i.e. the position away from the back-plane connector).

Caution

Do not invoke the self-test function, unless the S6PIO has been disconnected from the I/O system (both up-stream and down-stream connections should be disconnected). Failure to disconnect from the I/O system may cause physical damage.

While in Self-test mode, the front panel LEDs indicate the following:

Bottom LEDs (Active, RST)	Flash in unison to indicate self test mode
Middle LEDs (PARITY, ISOK)	Binary coded progress indicator
Top LH LED (BRDOK)	Binary coded progress indicator
Top RH LED (CHAIN)	Error indicator

In Self-test mode, the S6PIO module executes the following tests in a continuous repeating sequence:

Test	Parity LED Middle LHS	ISOK LED Middle RHS	BRDOK LED Top LHS
Start self test sequence	0	0	0
FPGA Download	1	0	0
Dual port RAM memory test.	0	1	0
SRAM memory test	1	1	0
FPGA Register check	0	0	1
Manual Mode loopback test	1	0	1

Table 8-1. Self-test LED Sequence Description

Note

The binary coding of the LEDs are indicated on the right hand side of the above table.

During Self-test (and while no errors are detected), the Chain LED (top right) will be off. The step-through sequence of the other LEDs is indicated below. If an error is detected, the Chain LED (top right) will be turned on and Parity, ISOK and BRDOK LEDs will indicate which test has failed. The test is then suspended (awaiting operator intervention or reset).

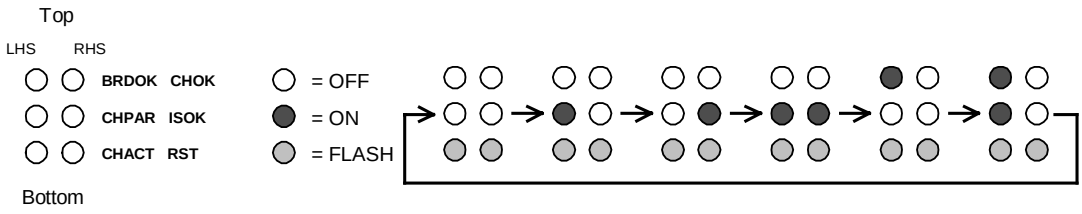


Figure 8-1. Self Test LED pattern

Appendix

A

SWITCHING BETWEEN MODES

This section illustrates the areas that need to be taken into account when switching between passive and master mode. This situation often arises in redundant systems.

Switching Modes

A fully redundant system may require a standby system to be monitoring in passive mode while the main system is operating in master mode. On change over to the standby system it would be required for the standby system to become the master and to switch from passive to master scan mode. (The previous master may also be required to switch from master scan mode to passive scan mode) However, it is not desirable to enter master scan mode immediately, as it is required that the passive blocks be stopped first (failure to stop any running passive blocks will result in the S6PIO being held in passive mode indefinitely). Additionally no unit should switch into master mode if I/O chain activity is detected (chain activity may be the result of a fault where communication is lost and both units go into solo mode with "Local Active" set).

A program example is provided in "Appendix B EXAMPLE 90-70 PROGRAMS". The example illustrates how both units should start up in passive mode with the master unit only going to master mode if no chain activity is detected. The passive block is stopped on the transition from passive mode to master mode. Note also that the passive block is not being used to scan any shared I/O. The passive block is primarily being used to monitor chain activity, not to maintain data coherency between the master and stand-by units. Shared I/O should still be used to maintain data coherency between the two units.

The user should be aware that when in solo mode, the shared I/O and registers are no longer shared. If the master scan PLC should stop, and the other PLC comes on automatically, depending on the code, the result on the I/O and internal PLC status may be unpredictable.

The S6SCAN module does have a feature that is specific to redundancy configuration and allows the user to force the S6PIO to idle mode if the 90-70 trips or is selected to stop (refer Table 4-4. Scan Options Register). In this mode, if the watchdog times out, while master scanning, it is assumed that the PLC has stopped, and will place the S6PIO module into idle mode so as not to freeze the I/O chain and allow the running PLC to control it.

Appendix B EXAMPLE 90-70 PROGRAMS

This section lists several example programs and illustrates the use of selected C blocks. All the examples are intended as a guide only and are provided to illustrate particular uses for particular blocks. In most cases the supporting code required for a safe live system has not been included.

The included examples are:

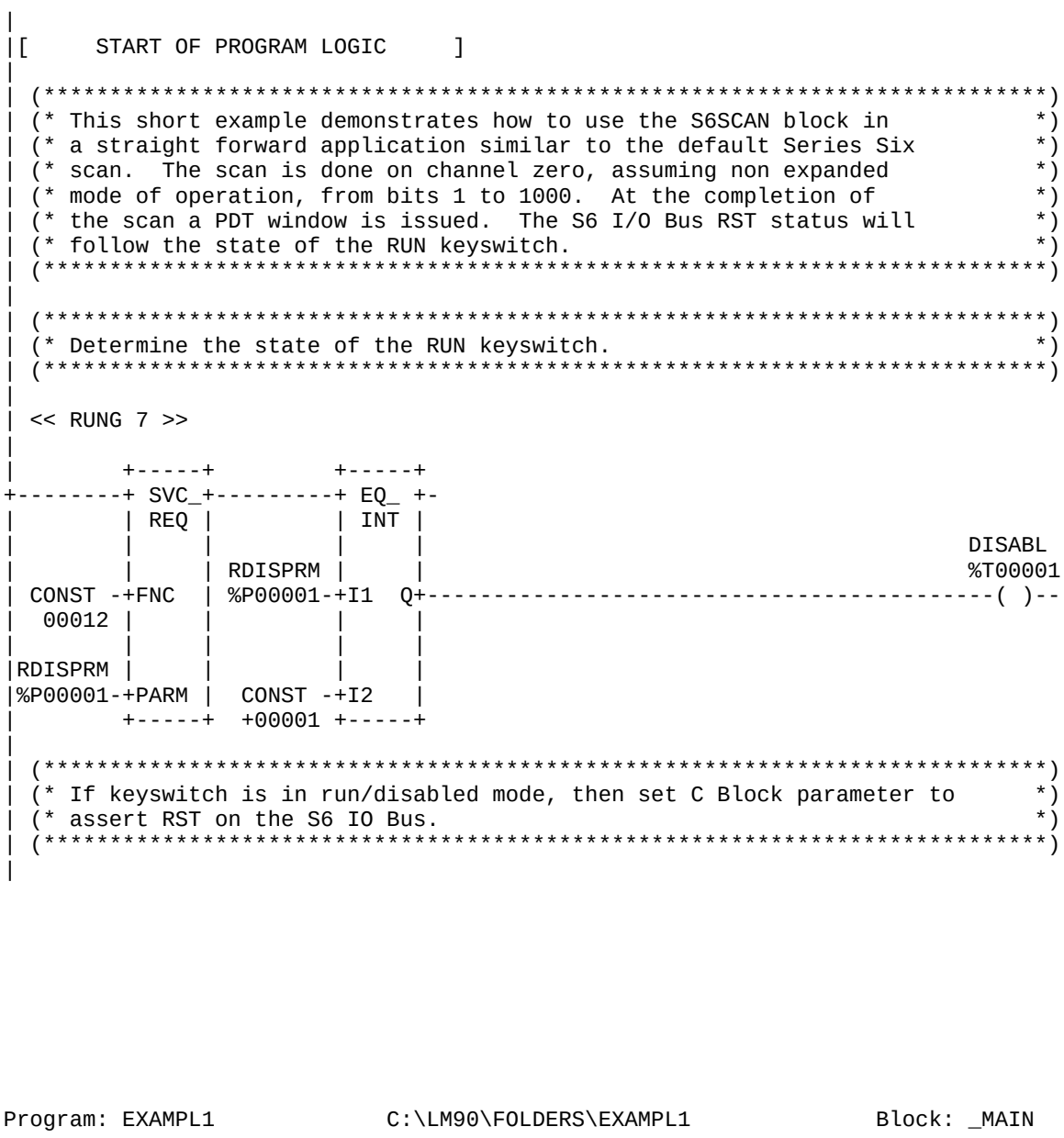
Example 1: Demonstrate Use of the S6SCAN Block

This example program illustrates the use of the S6SCAN C Block. The example is only intended as a guide when constructing real world applications. This example makes the following assumptions:

1 A single S6PIO module is used to replace a Series Six Primary I/O Chain.
2 Normal mode scanning is used.
3 The Series Six I/O chain follows the RUN/DISABLE key-switch status.
4 The S6PIO module is installed in slot 7 of the CPU rack.

Program Listing

EXAMPLE 1
This example is for reference only



Program: EXAMPL1 C:\LM90\FOLDERS\EXAMPL1 Block: _MAIN

```
04-22-02  14:37          GE FANUC SERIES 90-70 (v7.02)              Page     4
                                  EXAMPLE 1
                        This example is for reference only

| << RUNG 9 >>
|
|                 DISABL
|        +-----+ %T00001         +-----+
+--------+MOVE_+---] [-----------+MOVE_+-
|        | INT |                 | INT |
|        |     |                 |     |
|        |     |   RST           |     |   RST
| CONST -+IN  Q+-%P00002  CONST -+IN  Q+-%P00002
| +00000 | LEN |          +00001 | LEN |
|        |00001|                 |00001|
|        |     |                 |     |
|        +-----+                 +-----+
|
| (*************************************************************************)
| (* Scan the Series Six I/O.  Power flow occurs if                       *)
| (* the scan is successful without errors.                               *)
| (*                                                                      *)
| (* X1 - Rack/Slot number, set for CPU rack, slot 7                      *)
| (* X2 - Pointer to output table, set to %Q1                             *)
| (* X3 - Start address in bit format, set to 1                          *)
| (*      Scan channel 0, starting at bit 1                               *)
| (* X4 - End address in bit format, set to 1000                         *)
| (*      End the scan on channel 0 at bit 1000                           *)
| (* X5 - RST status, follows the state of the RUN                       *)
| (*      keyswitch, 1 means asserted, 0 deasserted                       *)
| (* X6 - Scan options, set to issue PDT window at                       *)
| (*      the completion of scanning                                      *)
| (* Y1 - Return error status                                            *)
| (* Y2 - Pointer to the input table, set to %I1                         *)
| (* Y3 - I/O Status register                                           *)
| (* Y4 - Software build number, 2 registers, first                     *)
| (*      is the C Block build number, second is the                      *)
| (*      S6PIO build number                                              *)
| (* Y5 - Not used                                                       *)
| (* Y6 - Not used                                                       *)
| (*************************************************************************)
|

Program: EXAMPL1          C:\LM90\FOLDERS\EXAMPL1          Block: _MAIN
```

```
04-22-02  14:37          GE FANUC SERIES 90-70 (v7.02)              Page     5
                                   EXAMPLE 1
                         This example is for reference only

| << RUNG 11 >>
|
|ALW_ON
|%S00007 +-------------+                                                %M00002
+--] [---+CALL  S6SCAN +-----------------------------------------------( )--
|        | (EXTERNAL)  |
|        |             |
|        |             |  ERRST
| CONST -+X1         Y1+-%R00001
| 0007   |            |
|        |            |
|S6_OUT  |            |  S6_IN
|%Q00001-+X2         Y2+-%I00001
|        |            |
|        |            | IOSTAT
| CONST -+X3         Y3+-%R00002
| 0001   |            |
|        |            |
|        |            | CBUILD
| CONST -+X4         Y4+-%R00003
| 03E8   |            |
|        |            |
|  RST   |            |
|%P00002-+X5         Y5+-
|        |            |
| CONST -+X6         Y6+-
| 0040   +-------------+
|
|[      END OF PROGRAM LOGIC      ]
|
```

Program: EXAMPL1 C:\LM90\FOLDERS\EXAMPL1 Block: _MAIN

Example 2: Demonstrate Use of C Blocks

This example program illustrates the use of the various C Blocks. The example is intended to be used as a guide when constructing real world applications. This example makes the following assumptions:

1 An S6PIO module set up in expanded channel master mode is used to replace a Series Six Primary I/O Chain.
2 Expanded mode scanning is used.
3 A single IOCCM module is present in the system at address 961 channel 1, communicating serially to a CCM module in a external system.
4 Priority input mode is used for the first active scan.
5 The Series Six I/O chain follows the RUN/DISABLE keyswitch status.
6 The S6PIO module is installed in slot 6 of the CPU rack.
7 Two high density digital input modules are present at %I129 – %I160 (channel 0) and %I1 - %I32 (channel 1).
8 Two high density digital output modules are present at %O129 – %I160 (channel 1) and %Q1 - %Q32 (channel 1).
9 Two analog input modules are present at address %I33 (channel 0) and %I65 (channel 1).
10 Two analog output modules are present at address %Q257 (channel 0) and %Q289 (channel 1).

Program Listing

```
05-03-02  18:25          GE FANUC SERIES 90-70 (v7.02)              Page     1
                               Reliabilty Test
                          Reliabilty test example 2

|[  START OF LD  PROGRAM RELIAB1  ]      (*                                *)
|
|[     VARIABLE DECLARATIONS      ]

             V A R I A B L E   D E C L A R A T I O N   T A B L E

          REFERENCE     NICKNAME          REFERENCE DESCRIPTION
          ---------     --------     --------------------------------
          %I00001       INTABL
          %I00017       INTABH
          %I00033       AIN1TAB
          %I00257       AOP1IN
          %I01313       AOP2IN
          %Q00001       OPTAB1L
          %Q00017       OPTAB1H
          %Q00257       AOP1TAB
          %Q01313       AOP2TAB
          %M00002       SAMPLE1      Sample rate for digital IO
          %M00003       SCN1OK
          %M00004       SCN2OK
          %M00007       WIN1OK
          %M00008       WIN2OK
          %M00101       SLOT6OK
          %M00104       MAXCNT2      Max value of counter for A out
          %M00105       MAXERR
          %M00106       MECNT
          %M00107       STATCNT
          %M00112       SLOT8OK
          %M00181       CMPERR       Mis compare error for IOCCM
          %M00201       DATOPOK
          %M01080       MEOVR
          %M01081       STSOVR
          %M01110       IP1ERR1      Monitor error
          %M01111       IP2ERR1
          %T00001       KEYOFF
          %P00001       SWSTAT
          %P00002       RSTSTS       Reset status from RUN switch
          %P00004       PRINP
          %P00006       SL6STAT
          %P00008       SL8STAT
          %R00001       COUNT1
          %R00002       COUNT2
          %R00011       SLOT
          %R00021       WIN1ERR
          %R00022       WIN1HDR
          %R00023       WIN1BLD
          %R00025       WIN2ERR
          %R00026       WIN2HDR
          %R00027       WIN2BLD
          %R00030       AOPSTS
          %R00031       AOP1STS
          %R00032       AOP1BLD
          %R00033       DUMMY1
```

```
          %R00040       AINSTS

Program: RELIAB1             C:\LM90\FOLDERS\RELIAB1             Block: _MAIN
```

```
05-03-02  18:25          GE FANUC SERIES 90-70 (v7.02)              Page     2
                               Reliabilty Test
                          Reliabilty test example 2

          %R00041       AIN1IOS
          %R00043       AIN1BLD
          %R00045       AIN2STS
          %R00046       AIN2IOS
          %R00051       SCNSTS
          %R00053       SCN1IOS
          %R00054       SCN1BLD
          %R00056       DUMMY3
          %R00057       DUMMY4
          %R00061       SCNSTS2
          %R00063       SCN2IOS
          %R00064       SCN2BLD
          %R00071       AOPSTS2
          %R00072       AOP2STS
          %R00073       AOP2BLD
          %R00075       DUMMY2
          %R00081       MOV1ERR
          %R00082       MEM1TYP
          %R00083       MEM1OFF
          %R00084       MEM1ID
          %R00085       MOV1BLD
          %R00091       MOV2ERR
          %R00092       MEM2TYP
          %R00093       MEM2OFF
          %R00094       MEM2ID
          %R00095       MOV2BLD
          %R00101       AOUTX4
          %R00102       AOUT1P2
          %R00103       AOUT1P3
          %R00104       AOUT1P4
          %R00111       AINBUF
          %R00121       AIN2BUF
          %R00131       AINSBTS
          %R00141       TEMP1
          %R00151       MOV3ERR
          %R00152       MEM3TYP
          %R00153       MEM3ID
          %R00154       MEM3OFF
          %R00155       MOV3BLD
          %R00201       MONAIO
          %R00211       MONSTAT
          %R00301       SCNCNT1
          %R00311       ERRCNT1      Digital loopback error counter
          %R00321       ERRCNT2      Static Analog value monitor
          %R00331       ERRCNT3      Analog counter error check
          %R00341       ERRCNT4      Window error count for IOCCM
          %R00351       ERRCNT5      IOCCM serial loopback er count
          %R00411       MAXLIM       10% band plus 10
          %R00412       TEMP2
          %R00413       MINLIM       10% baqnd minus 10
          %R00501       TSTERR1      Wrt compare value on error IOCCM
          %R00502       TSTERR2      RD compare value on error IOCCM
          %R00503       TSTERR3      Dig Loopback OP Low Err Value
          %R00504       TSTERR4      Dig Loopback IP Low Err Value

Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1            Block: _MAIN
```

```
05-03-02  18:25          GE FANUC SERIES 90-70 (v7.02)              Page     3
                               Reliabilty Test
                          Reliabilty test example 2

          %R00505       TSTERR5      Dig Loopback OP High Err Value
          %R00506       TSTERR6      Dig Loopback IP High Err Value
          %R00507       TSTERR7      Static Analog Err Value
          %R00508       TSTERR8      Analog Cnt OP Value
          %R00509       TSTERR9      Analog Cnt IP Value
          %R00961       IOCCMAD
          %R00968       WRTREG
          %R00969       RDREG
          %R01015       TMR2
          %R01020       CNTR2
          %R01182       RAWTP1L      Data test block raw o/p data
          %R01183       RAWTP1H
          %R01184       TESTP1L      Digital IO point test lower reg
          %R01185       TESTP1H
          %R02000       LOGERR
          %R02011       LOGHDR
          %R02031       LOGCNT
          %R02032       LOGBLD
          %R02041       LOGDATA
          %AI0034       AIN2TAB


                        I D E N T I F I E R    T A B L E

          IDENTIFIER     IDENTIFIER TYPE          IDENTIFIER DESCRIPTION
          ----------     ---------------     --------------------------------
          S6AOP          EXTERNAL BLOCK
          S6AIN          EXTERNAL BLOCK
          S6SCAN         EXTERNAL BLOCK
          S6NOOP         EXTERNAL BLOCK
          S6EVTLG        EXTERNAL BLOCK
          S6MOVE         EXTERNAL BLOCK
          S6XWIN         EXTERNAL BLOCK
          DATATST        EXTERNAL BLOCK
          INIT           PROGRAM BLOCK
          DIGLB          PROGRAM BLOCK
          ANLGLB         PROGRAM BLOCK
          SCAN           PROGRAM BLOCK
          S6STAT         EXTERNAL BLOCK
          CCMTST         PROGRAM BLOCK
          S6OPT          EXTERNAL BLOCK
          S6WIN          EXTERNAL BLOCK
          RELIAB1        PROGRAM NAME

|[   PROGRAM BLOCK DECLARATIONS   ]

               +-------+
               | S6AOP |       LANG: EXT (*                                *)
               +-------+       PARAM:6

               +-------+
               | S6AIN |       LANG: EXT (*                                *)
               +-------+       PARAM:6


Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1             Block: _MAIN
```

```
05-03-02  18:25          GE FANUC SERIES 90-70 (v7.02)              Page     4
                               Reliabilty Test
                          Reliabilty test example 2

                  +-------+
                  |S6SCAN |       LANG: EXT (*                              *)
                  +-------+       PARAM:6

                  +-------+
                  |S6NOOP |       LANG: EXT (*                              *)
                  +-------+       PARAM:1

                  +-------+
                  |S6EVTLG|       LANG: EXT (*                              *)
                  +-------+       PARAM:5

                  +-------+
                  |S6MOVE |       LANG: EXT (*                              *)
                  +-------+       PARAM:5

                  +-------+
                  |S6XWIN |       LANG: EXT (*                              *)
                  +-------+       PARAM:6

                  +-------+
                  |DATATST|       LANG: EXT (*                              *)
                  +-------+       PARAM:4

                  +-------+
                  | INIT  |       LANG: LD  (*                              *)
                  +-------+

                  +-------+
                  | DIGLB |       LANG: LD  (*                              *)
                  +-------+

                  +-------+
                  |ANLGLB |       LANG: LD  (*                              *)
                  +-------+

                  +-------+
                  | SCAN  |       LANG: LD  (*                              *)
                  +-------+

                  +-------+
                  |S6STAT |       LANG: EXT (*                              *)
                  +-------+       PARAM:3

                  +-------+
                  |CCMTST |       LANG: LD  (*                              *)
                  +-------+

                  +-------+
                  | S6OPT |       LANG: EXT (*                              *)
                  +-------+       PARAM:3



Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1            Block: _MAIN
```

```
05-03-02  18:25          GE FANUC SERIES 90-70 (v7.02)              Page     5
                               Reliabilty Test
                          Reliabilty test example 2

                +-------+
                | S6WIN |       LANG: EXT (*                                *)
                +-------+       PARAM:4

|[           INTERRUPTS           ]
|
|[     START OF PROGRAM LOGIC     ]
|
| << RUNG 5 >>
|
|ALW_ON
|%S00007          +-------------+
+--] [------------+CALL   INIT  +
|                 +-------------+
|
| << RUNG 6 >>
|
|ALW_ON
|%S00007          +-------------+
+--] [------------+CALL  CCMTST +
|                 +-------------+
|
| << RUNG 7 >>
|
|ALW_ON
|%S00007          +-------------+
+--] [------------+CALL   DIGLB +
|                 +-------------+
|
| << RUNG 8 >>
|
|ALW_ON
|%S00007          +-------------+
+--] [------------+CALL  ANLGLB +
|                 +-------------+
|
| << RUNG 9 >>
|
|ALW_ON
|%S00007          +-------------+
+--] [------------+CALL   SCAN  +
|                 +-------------+
|


Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1            Block: _MAIN
```

```
05-03-02  18:25          GE FANUC SERIES 90-70 (v7.02)              Page     6
                               Reliabilty Test
                          Reliabilty test example 2

| << RUNG 10 >>
|
|ALW_ON
|%S00007 +-------------+                                                %M00009
+--] [---+CALL  S6STAT +-----------------------------------------------( )--
|        | (EXTERNAL)  |
|        |             |
| SLOT   |             |
|%R00011-+X1         Y1+-%R01400
|        |            |
| CONST -+X2         Y2+-%R01401
| 0000   |            |
|        |            |
|%R00900-+X3         Y3+-%R01420
|        +-------------+
|
| << RUNG 11 >>
|
|SLOT6OK
|%M00101 +-------------+
+--] [---+CALL  S6EVTLG+-
|        | (EXTERNAL)  |
|        |             |
|        |             | LOGERR
| CONST -+X1         Y1+-%R02000
| 0006   |            |
|        |            |
|        |            | LOGHDR
| CONST -+X2         Y2+-%R02011
| 0002   |            |
|        |            |
|        |            | LOGCNT
| CONST -+X3         Y3+-%R02031
| 0000   |            |
|        |            |
|        |            | LOGDATA
| CONST -+X4         Y4+-%R02041
| 0002   |            |
|        |            |
|        |            | LOGBLD
| CONST -+X5         Y5+-%R02032
| 0000   +-------------+
|
|[      END OF PROGRAM LOGIC      ]
|

Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1            Block: _MAIN
```

```
05-03-02  18:25          GE FANUC SERIES 90-70 (v7.02)              Page     7
                               Reliabilty Test
                          Reliabilty test example 2

+[  START  OF  LD  BLOCK  INIT    ]
|
|[     VARIABLE DECLARATIONS      ]

             V A R I A B L E   D E C L A R A T I O N   T A B L E

          REFERENCE     NICKNAME          REFERENCE DESCRIPTION
          ---------     --------     -------------------------------
                          NO  VARIABLE  TABLE  ENTRIES


                        I D E N T I F I E R    T A B L E

          IDENTIFIER     IDENTIFIER TYPE          IDENTIFIER DESCRIPTION
          ----------     ---------------     --------------------------------
                         NO  IDENTIFIER  TABLE  ENTRIES


+[      START OF BLOCK LOGIC      ]
|
| (*************************************************************************)
| (* Determine the state of the RUN keyswitch                               *)
| (*************************************************************************)
|
| << RUNG 4 >>
|
|        +-----+         +-----+
+--------+ SVC_+---------+ EQ_ +-
|        | REQ |         | INT |
|        |     |         |     |
|        |     |         |     |                                          KEYOFF
|        |     | SWSTAT  |     |                                          %T00001
| CONST -+FNC  | %P00001-+I1  Q+------------------------------------------( )--
|  00012 |     |         |     |
|        |     |         |     |
|SWSTAT  |     |         |     |
|%P00001-+PARM |  CONST -+I2   |
|        +-----+  +00001 +-----+
|
| (*************************************************************************)
| (* If keyswitch is in run/disable mode, then set C clock parameter to      *)
| (* assert RST on the S6IO Bus.                                            *)
| (*************************************************************************)
|

Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1             Block: INIT
```

```
05-03-02  18:25          GE FANUC SERIES 90-70 (v7.02)              Page     8
                               Reliabilty Test
                          Reliabilty test example 2

| << RUNG 6 >>
|
|                KEYOFF
|        +-----+ %T00001         +-----+
+--------+MOVE_+---] [-----------+MOVE_+-
|        | INT |                 | INT |
|        |     |                 |     |
|        |     | Reset s         |     | Reset s
|        |     | tatus f         |     | tatus f
|        |     | rom RUN         |     | rom RUN
|        |     | switch          |     | switch
|        |     | RSTSTS          |     | RSTSTS
| CONST -+IN  Q+-%P00002  CONST -+IN  Q+-%P00002
| +00000 | LEN |          +00001 | LEN |
|        |00001|                 |00001|
|        |     |                 |     |
|        +-----+                 +-----+
|
| << RUNG 7 >>
|
|        +-----+
+--------+MOVE_+-
|        | INT |
|        |     |
|        |     |  SLOT
| CONST -+IN  Q+-%R00011
| +00006 | LEN |
|        |00001|
|        |     |
|        +-----+
|
| (*************************************************************************)
| (* Check to ensure S6PIO is in slot 6                                    *)
| (*************************************************************************)
|
| << RUNG 9 >>
|
|ALW_ON                                                          SLOT6OK
|%S00007 +-------------+                                         %M00101
+--] [---+CALL  S6NOOP +-------------------------------------------------( )--
|        | (EXTERNAL)  |
|        |             |
| SLOT   |             | SL6STAT
|%R00011-+X1         Y1+-%P00006
|        +-------------+
|


Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1            Block: INIT
```

```
05-03-02  18:25          GE FANUC SERIES 90-70 (v7.02)              Page     9
                               Reliabilty Test
                          Reliabilty test example 2

| (*************************************************************************)
| (* First of all clear counter and error registers on first scan          *)
| (* SCNCNT1 (301R) = Number of scans                                       *)
| (* ERRCNT1 (311R) = Random digital loopback error count                  *)
| (* ERRCNT2 (321R) = Static Analog value monitor error count              *)
| (* ERRCNT3 (331R) = Analog counter loopback error count                 *)
| (* ERRCNT4 (341R) = IOCCM window error count                            *)
| (* ERRCNT5 (351R) = IOCCM serial loopback mis-compare error count       *)
| (* TSTERR1 (501R) = IOCCM last read error value                         *)
| (* TSTERR2 (502R) = IOCCM last write error value                        *)
| (* TSTERR3 (503R) = Digital Loopback Output error value (low)           *)
| (* TSTERR4 (504R) = Digital Loopback Input error value (low)            *)
| (* TSTERR5 (505R) = Digital Loopback Output error value (high)          *)
| (* TSTERR6 (506R) = Digital Loopback Input error value (high)           *)
| (* TSTERR7 (507R) = Static Analog error value                           *)
| (* TSTERR8 (508R) = Analog counter output error value                   *)
| (* TSTERR9 (509R) = Analog counter input error value                    *)
| (* TSTERR10 (510R) = S6XWIN window return error value                  *)
| (*************************************************************************)
|
| << RUNG 11 >>
|
|SLOT6OK
|%M00101 +-----+                +-----+                +-----+
+--]^[---+MOVE_+----------------+MOVE_+----------------+MOVE_+-
|        | INT |                | INT |                | INT |
|        |     |                |     |                |     |
|        |     |                |     | Digital        |     | Static
|        |     |                |     | loopbac        |     | Analog
|        |     |                |     | k error        |     | value
|        |     |                |     | counter        |     | monitor
|        |     | SCNCNT1        |     | ERRCNT1        |     | ERRCNT2
| CONST -+IN  Q+-%R00301  CONST -+IN  Q+-%R00311  CONST -+IN  Q+-%R00321
| +00000 | LEN |          +00000 | LEN |          +00000 | LEN |
|        |00001|                |00001|                |00001|
|        |     |                |     |                |     |
|        +-----+                +-----+                +-----+
|

Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1            Block: INIT
```

```
05-03-02  18:25          GE FANUC SERIES 90-70 (v7.02)              Page    10
                               Reliabilty Test
                          Reliabilty test example 2

| << RUNG 12 >>
|
|SLOT6OK
|%M00101 +-----+
+--]^[---+MOVE_+-
|        | INT |
|        |     |
|        |     | Analog
|        |     | counter
|        |     | error
|        |     | check
|        |     | ERRCNT3
| CONST -+IN  Q+-%R00331
| +00000 | LEN |
|        |00001|
|        |     |
|        +-----+
|
| << RUNG 13 >>
|
|SLOT6OK
|%M00101 +-----+                 +-----+
+--]^[---+BLKMV+-----------------+BLKMV+-
|        | INT |                 | INT |
|        |     |                 |     |
|        |     | Wrt com         |     |
|        |     | pare va         |     | Analog
|        |     | lue on          |     | Cnt OP
|        |     | error I         |     | Value
|        |     | TSTERR1         |     | TSTERR8
| CONST -+IN1 Q+-%R00501  CONST -+IN1 Q+-%R00508
| +00000 |     |          +00000 |     |
|        |     |                 |     |
| CONST -+IN2  |          CONST -+IN2  |
| +00000 |     |          +00000 |     |
|        |     |                 |     |
| CONST -+IN3  |          CONST -+IN3  |
| +00000 |     |          +00000 |     |
|        |     |                 |     |
| CONST -+IN4  |          CONST -+IN4  |
| +00000 |     |          +00000 |     |
|        |     |                 |     |
| CONST -+IN5  |          CONST -+IN5  |
| +00000 |     |          +00000 |     |
|        |     |                 |     |
| CONST -+IN6  |          CONST -+IN6  |
| +00000 |     |          +00000 |     |
|        |     |                 |     |
| CONST -+IN7  |          CONST -+IN7  |
| +00000 +-----+          +00000 +-----+
|
```

```
Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1            Block: INIT
```

```
05-03-02  18:25          GE FANUC SERIES 90-70 (v7.02)              Page    11
                               Reliabilty Test
                          Reliabilty test example 2

| (*************************************************************************)
| (* Derive priority input mode. Used during the first active scan of the     *)
| (* S6PIO.                                                                   *)
| (* Priority input mode allows the S6PIO to read input data from the I/O     *)
| (* bus without allowing output data to be received by output cards. It      *)
| (* is used for the first scan after the S6PIO becomes active.               *)
| (*************************************************************************)
|
| << RUNG 15 >>
|
|                 SLOT6OK
|        +-----+ %M00101         +-----+
+--------+MOVE_+---]^[-----------+MOVE_+-
|        | INT |                 | INT |
|        |     |                 |     |
|        |     |  PRINP          |     |  PRINP
| CONST -+IN  Q+-%P00004  CONST -+IN  Q+-%P00004
| +00016 | LEN |          +00002 | LEN |
|        |00001|                 |00001|
|        |     |                 |     |
|        +-----+                 +-----+
|
+[       END OF BLOCK LOGIC       ]
|
```

```
Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1            Block: INIT
```

```
05-03-02  18:25          GE FANUC SERIES 90-70 (v7.02)              Page    12
                               Reliabilty Test
                          Reliabilty test example 2

+[  START  OF  LD  BLOCK  DIGLB   ]
|
|[     VARIABLE DECLARATIONS      ]

             V A R I A B L E   D E C L A R A T I O N   T A B L E

          REFERENCE     NICKNAME          REFERENCE DESCRIPTION
          ---------     --------     -------------------------------
                          NO  VARIABLE  TABLE  ENTRIES


                        I D E N T I F I E R    T A B L E

          IDENTIFIER     IDENTIFIER TYPE          IDENTIFIER DESCRIPTION
          ----------     ---------------     --------------------------------
                         NO  IDENTIFIER  TABLE  ENTRIES


+[      START OF BLOCK LOGIC      ]
|
| (*************************************************************************)
| (* Set up sample rate for digital scan checks. Sample is taken every 2      *)
| (* scans.                                                                   *)
| (*************************************************************************)
|
| << RUNG 4 >>
|
|ALW_ON
|%S00007 +-----+                                    +-----+
+--] [---+ EQ_ +----------------------------------+ ADD_+-
|        | INT |                                  | INT |
|        |     |                                  |     |
|SCNCNT1 |     |         +-----+         SCNCNT1 |     | SCNCNT1
|%R00301-+I1  Q+---------+MOVE_+-        %R00301-+I1  Q+-%R00301
|        |     |         | INT |                  |     |
|        |     |         |     |                  |     |
|        |     |         |     | SCNCNT1          |     |
| CONST -+I2   |  CONST -+IN  Q+-%R00301  CONST -+I2   |
| +32767 +-----+  -32768 | LEN |           +00001 +-----+
|                        |00001|
|                        |     |
|                        +-----+
|

Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1             Block: DIGLB
```

```
05-03-02  18:25          GE FANUC SERIES 90-70 (v7.02)              Page    13
                               Reliabilty Test
                          Reliabilty test example 2

| << RUNG 5 >>
|
|        +-----+
+--------+ GE_ +-
|        | INT |
|        |     |
|        |     |                                                        SAMPLE1
|SCNCNT1 |     |         +-----+                                        %M00002
|%R00301-+I1  Q+---------+MOVE_+----------------------------------------( )--
|        |     |         | INT |
|        |     |         |     |
|        |     |         |     | SCNCNT1
| CONST -+I2   |  CONST -+IN  Q+-%R00301
| +00002 +-----+  +00000 | LEN |
|                        |00001|
|                        |     |
|                        +-----+
|
| (*************************************************************************)
| (* Monitor the returned input data against the raw output digital random    *)
| (* data. Both lower and upper registers are monitored.                      *)
| (*************************************************************************)
|
| << RUNG 7 >>
|
|Sample
|rate fo
|r digit
|al IO
|SAMPLE1
|%M00002         +-----+
+--] [-----------+ NE_ +-
|                | INT |
|                |     |
|        Digital |     |
|        IO poin |     |
|        t test  |     |
|        lower r |     |                                                IP1ERR1
|        TESTP1L |     |                                                %M01110
|        %R01184-+I1  Q+-------------------------------------------------(S)--
|                |     |
|        INTABL  |     |
|        %I00001-+I2   |
|                +-----+
|
```

```
Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1             Block: DIGLB
```

```
05-03-02  18:25          GE FANUC SERIES 90-70 (v7.02)              Page    14
                               Reliabilty Test
                          Reliabilty test example 2

| << RUNG 8 >>
|
|Sample
|rate fo
|r digit
|al IO
|SAMPLE1
|%M00002         +-----+
+--] [-----------+ EQ_ +-
|                | INT |
|                |     |
|        Digital |     |
|        IO poin |     |
|        t test  |     |
|        lower r |     |                                              IP1ERR1
|        TESTP1L |     |                                              %M01110
|        %R01184-+I1  Q+--------------------------------------------------(R)--
|                |     |
|        INTABL  |     |
|        %I00001-+I2   |
|                +-----+
|
| << RUNG 9 >>
|
|Sample
|rate fo
|r digit
|al IO
|SAMPLE1
|%M00002         +-----+
+--] [-----------+ NE_ +-
|                | INT |
|                |     |
|                |     |                                              IP2ERR1
|        TESTP1H |     |                                              %M01111
|        %R01185-+I1  Q+--------------------------------------------------(S)--
|                |     |
|        INTABH  |     |
|        %I00017-+I2   |
|                +-----+
|

Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1            Block: DIGLB
```

```
05-03-02  18:25          GE FANUC SERIES 90-70 (v7.02)              Page    15
                               Reliabilty Test
                          Reliabilty test example 2

| << RUNG 10 >>
|
|Sample
|rate fo
|r digit
|al IO
|SAMPLE1
|%M00002         +-----+
+--] [-----------+ EQ_ +-
|                | INT |
|                |     |
|                |     |                                                  IP2ERR1
|        TESTP1H |     |                                                  %M01111
|        %R01185-+I1  Q+--------------------------------------------------(R)--
|                |     |
|        INTABH  |     |
|        %I00017-+I2   |
|                +-----+
|
| (*************************************************************************)
| (* This error register records any differences in digital loopback.      *)
| (* ERRCNT1 = number of mis-compares in digital loopback                   *)
| (* TSTERR3 = Output lower reg value                                       *)
| (* TSTERR4 = Input lower reg value                                        *)
| (* TSTERR5 = Output Upper reg value                                      *)
| (* TSTERR6 = Input upper reg value                                       *)
| (*************************************************************************)
|
| << RUNG 12 >>
|
|Sample
|rate fo
|r digit Monitor
|al IO   error
|SAMPLE1 IP1ERR1
|%M00002 %M01110 +-----+                +-----+                 +-----+
+--] [-----] [---+ ADD_+----------------+MOVE_+-----------------+MOVE_+-
|                | INT |                | INT |                 | INT |
|                |     |                |     |                 |     |
|        Digital |     | Digital Digital |     | Dig Loo         |     | Dig Loo
|        loopbac |     | loopbac IO poin |     | pback O         |     | pback I
|        k error |     | k error t test  |     | P Low E         |     | P Low E
|        counter |     | counter lower r |     | rr Valu         |     | rr Valu
|        ERRCNT1 |     | ERRCNT1 TESTP1L |     | TSTERR3 INTABL  |     | TSTERR4
|        %R00311-+I1  Q+-%R00311 %R01184-+IN  Q+-%R00503 %I00001-+IN  Q+-%R00504
|                |     |                | LEN |                 | LEN |
|                |     |                |00001|                 |00001|
|                |     |                |     |                 |     |
|         CONST -+I2   |                +-----+                 +-----+
|         +00001 +-----+
|

Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1             Block: DIGLB
```

```
05-03-02  18:25          GE FANUC SERIES 90-70 (v7.02)              Page    16
                               Reliabilty Test
                          Reliabilty test example 2

| << RUNG 13 >>
|
|Sample
|rate fo
|r digit
|al IO
|SAMPLE1 IP2ERR1
|%M00002 %M01111 +-----+                +-----+                +-----+
+--] [-----] [---+ ADD_+----------------+MOVE_+----------------+MOVE_+-
|                | INT |                | INT |                | INT |
|                |     |                |     |                |     |
|        Digital |     | Digital        |     | Dig Loo        |     | Dig Loo
|        loopbac |     | loopbac        |     | pback O        |     | pback I
|        k error |     | k error        |     | P High         |     | P High
|        counter |     | counter        |     | Err Val        |     | Err Val
|        ERRCNT1 |     | ERRCNT1 TESTP1H |     | TSTERR5 INTABH  |     | TSTERR6
|        %R00311-+I1  Q+-%R00311 %R01185-+IN  Q+-%R00505 %I00017-+IN  Q+-%R00506
|                |     |                | LEN |                | LEN |
|                |     |                |00001|                |00001|
|                |     |                |     |                |     |
|         CONST -+I2   |                +-----+                +-----+
|         +00001 +-----+
|
| (*************************************************************************)
| (* Produce random data to output to digital address cards, and value and    *)
| (* hold in the following registers.                                         *)
| (* TESTP1L = Test point 1 lower                                            *)
| (* TESTP1H = Test point 1 upper                                            *)
| (*************************************************************************)
|

Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1            Block: DIGLB
```

```
05-03-02  18:25          GE FANUC SERIES 90-70 (v7.02)              Page    17
                               Reliabilty Test
                          Reliabilty test example 2

| << RUNG 15 >>
|
|Sample
|rate fo
|r digit
|al IO
|SAMPLE1                                                                DATOPOK
|%M00002 +-------------+                                                %M00201
+--] [---+CALL  DATATST+-----------------------------------------------------( )--
|        | (EXTERNAL)  |
|        |             |
|        |             | Data te
|        |             | st bloc
|        |             | k raw o
|        |             | /p data
|        |             | RAWTP1L
| CONST -+X1         Y1+-%R01182
| 0002   |             |
|        |             |
| CONST -+X2         Y2+-
| 0004   |             |
|        |             |
| CONST -+X3         Y3+-
| 0001   |             |
|        |             |
| CONST -+X4         Y4+-
| 0000   +-------------+
|
| << RUNG 16 >>
|
|SLOT6OK
|%M00101         +-----+
+--] [-----------+ AND_+-
|                | WORD|
|                |     |
|        Data te |     | Digital
|        st bloc |     | IO poin
|        k raw o |     | t test
|        /p data |     | lower r
|        RAWTP1L |     | TESTP1L
|        %R01182-+I1  Q+-%R01184
|                | LEN |
|                |00001|
|                |     |
|         CONST -+I2   |
|           FFFF +-----+
|

Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1            Block: DIGLB
```

```
05-03-02  18:25          GE FANUC SERIES 90-70 (v7.02)              Page    18
                                Reliabilty Test
                          Reliabilty test example 2

| << RUNG 17 >>
|
|SLOT6OK
|%M00101         +-----+
+--] [-----------+ AND_+-
|                | WORD|
|                |     |
|        RAWTP1H |     | TESTP1H
|        %R01183-+I1  Q+-%R01185
|                | LEN |
|                |00001|
|                |     |
|         CONST -+I2   |
|           FFFF +-----+
|
| << RUNG 18 >>
|
|ALW_ON
|%S00007 +-----+                 +-----+
+--] [---+MOVE_+-----------------+MOVE_+-
|        | INT |                 | INT |
|        |     |                 |     |
|Digital |     |                 |     |
|IO poin |     |                 |     |
|t test  |     |                 |     |
|lower r |     |                 |     |
|TESTP1L |     | OPTAB1L TESTP1H |     | OPTAB1H
|%R01184-+IN  Q+-%Q00001 %R01185-+IN  Q+-%Q00017
|        | LEN |                 | LEN |
|        |00001|                 |00001|
|        |     |                 |     |
|        +-----+                 +-----+
|
+[       END OF BLOCK LOGIC       ]
|
```

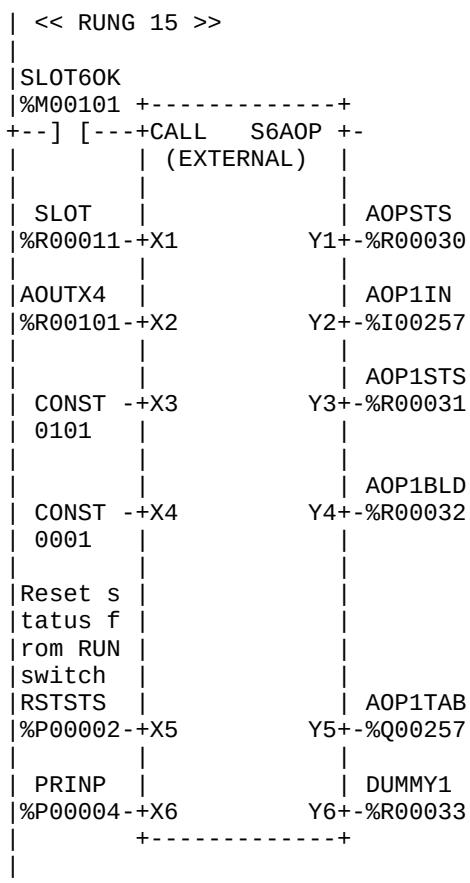
```
Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1            Block: DIGLB
```

```
05-03-02  18:25          GE FANUC SERIES 90-70 (v7.02)              Page    19
                               Reliabilty Test
                          Reliabilty test example 2

+[  START  OF  LD  BLOCK ANLGLB   ]
|
|[     VARIABLE DECLARATIONS      ]

             V A R I A B L E   D E C L A R A T I O N   T A B L E

          REFERENCE     NICKNAME          REFERENCE DESCRIPTION
          ---------     --------     -------------------------------
                          NO  VARIABLE  TABLE  ENTRIES


                        I D E N T I F I E R    T A B L E

          IDENTIFIER     IDENTIFIER TYPE          IDENTIFIER DESCRIPTION
          ----------     ---------------     --------------------------------
                         NO  IDENTIFIER  TABLE  ENTRIES


+[      START OF BLOCK LOGIC      ]
|
| (*************************************************************************)
| (* Testing expanded channel mode                                        *)
| (*                                                                       *)
| (* Channel 0 has digital I/O starting at 129 (to 145)                   *)
| (* And analog input at 33                                               *)
| (* And Analog output at 257                                            *)
| (*                                                                       *)
| (*                                                                       *)
| (* Channel 1 das digital I/O stating at 0 (to 17)                       *)
| (* And analog input at 65                                              *)
| (* And Analog output at 289                                           *)
| (*                                                                       *)
| (* The upcounter counts 0 to 1024 (FS) incrementing each second        *)
| (* The CV is used as the analog output value for analog channel 1      *)
| (* Analog channel 2 has a constant value 1024                         *)
| (* Analog channel 3 has Channel 1 / 2                                 *)
| (* Analog channel 4 has channel 1 / 4                                 *)
| (*                                                                       *)
| (*************************************************************************)
|
| << RUNG 4 >>
|
|SLOT6OK
|%M00101                                                          %M00120
+--]^[-------------------------------------------------------------------( )--
|


Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1            Block: ANLGLB
```

```
05-03-02  18:25          GE FANUC SERIES 90-70 (v7.02)              Page    20
                               Reliabilty Test
                          Reliabilty test example 2

| << RUNG 5 >>
|
|Max val
|ue of c
|ounter
|for A o
|MAXCNT2
|%M00104                                                                %M00120
+--]/[-----------------------------------------------------------------------(S)--
|
| << RUNG 6 >>
|
|        +-----+                                                        %M00109
+--------+ONDTR+-----------------------------------------------------------(^)--
|        |1.00s|
|        |     |
|%M00109 |     |
+--]v[---+R    |
|        |     |
| CONST -+PV CV+-%R04001
| +00005 |     |
|        +-----+
|
|        %R04011
|
| (*************************************************************************)
| (* Simple count up and down routines for controlling changing analog      *)
| (* values to test varying outputs and inputs.                             *)
| (*************************************************************************)
|
| << RUNG 8 >>
|
|                                                                       MAXCNT2
|%M00120 %M00109 +-----+                                                %M00104
+--] [-----] [---->UPCTR+-------------------------------------------------(S)--
|               |     |
|        Max val |     |
|        ue of c |     |
|        ounter  |     |
|        for A o |     |
|        MAXCNT2 |     |
|%M00120 %M00104 |     |
+--] [-----] [---+R    |
|               |     |
|               |     | COUNT2
|         CONST -+PV CV+-%R00002
|         +02047 |     |
|                +-----+
|
|                %R04021
|


Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1            Block: ANLGLB
```

```
05-03-02  18:25          GE FANUC SERIES 90-70 (v7.02)              Page    21
                               Reliabilty Test
                          Reliabilty test example 2

| << RUNG 9 >>
|
|                                                                        MAXCNT2
|%M00120 %M00109 +-----+                                                 %M00104
+--]/[-----] [--->DNCTR+-------------------------------------------------(R)--
|                |     |
|        Max val |     |
|        ue of c |     |
|        ounter  |     |
|        for A o |     |
|        MAXCNT2 |     |
|%M00120 %M00104 |     |
+--]/[-----]/[---+R    |
|                |     |
|                |     | COUNT1
|         CONST -+PV CV+-%R00001
|         +02047 |     |
|                +-----+
|
|                %R04031
|
| (*************************************************************************)
| (* Set up channel values.                                                *)
| (* AOUTX4 = Count (0 -> 2047, 2047 -> 0)                                 *)
| (* AOUT1P2 = 1000 (static value)                                         *)
| (* AOUT1P3 = Count/2                                                     *)
| (* AOUT1P4 = Count/4                                                     *)
| (*************************************************************************)
|
| << RUNG 11 >>
|
|ALW_ON
|%S00007 +-----+                 +-----+
+--] [---+MOVE_+-----------------+MOVE_+-
|        | INT |                 | INT |
|        |     |                 |     |
|COUNT2  |     | AOUTX4          |     | AOUT1P2
|%R00002-+IN  Q+-%R00101  CONST -+IN  Q+-%R00102
|        | LEN |          +01000 | LEN |
|        |00001|                 |00001|
|        |     |                 |     |
|        +-----+                 +-----+
|

Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1            Block: ANLGLB
```

```
05-03-02  18:25          GE FANUC SERIES 90-70 (v7.02)              Page    22
                               Reliabilty Test
                          Reliabilty test example 2

| << RUNG 12 >>
|
|ALW_ON
|%S00007 +-----+
+--] [---+ DIV_+-
|        | INT |
|        |     |
|COUNT2  |     | AOUT1P3
|%R00002-+I1  Q+-%R00103
|        |     |
| CONST -+I2   |
| +00002 +-----+
|
| << RUNG 13 >>
|
|ALW_ON
|%S00007 +-----+
+--] [---+ DIV_+-
|        | INT |
|        |     |
|COUNT2  |     | AOUT1P4
|%R00002-+I1  Q+-%R00104
|        |     |
| CONST -+I2   |
| +00004 +-----+
|
| (*************************************************************************)
| (* Output analog output to first analog card at 257                      *)
| (*                                                                       *)
| (* The output to second analog card at 289                              *)
| (*************************************************************************)
|

Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1            Block: ANLGLB
```


| << RUNG 15 >>
|SLOT6OK
|%M00101 +-------------+
+--] [---+CALL S6AOP +-
| | (EXTERNAL) |
| SLOT | | AOPSTS
|%R00011-+X1 Y1+-%R00030
|AOUTX4 | | AOP1IN
|%R00101-+X2 Y2+-%I00257
| CONST -+X3 Y3+-%R00031
| 0101 | |
| CONST -+X4 Y4+-%R00032
0001	
Reset s	
tatus f	
rom RUN	
switch	
RSTSTS		AOP1TAB
%P00002-+X5 Y5+-%Q00257
PRINP		DUMMY1
%P00004-+X6 Y6+-%R00033
+-------------+

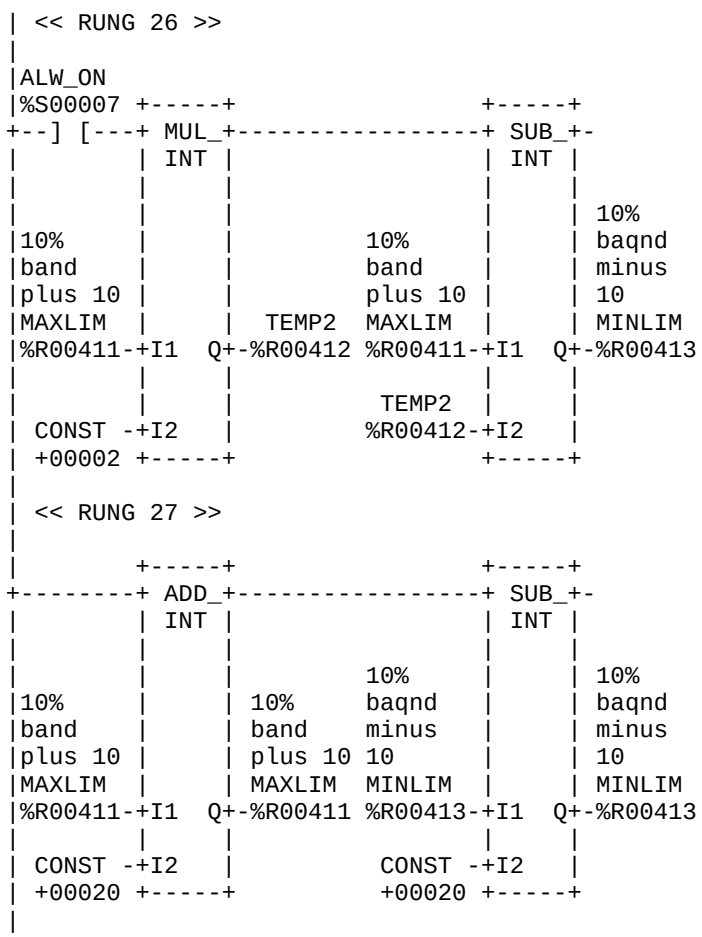
```
05-03-02  18:25          GE FANUC SERIES 90-70 (v7.02)              Page    24
                               Reliabilty Test
                          Reliabilty test example 2

| << RUNG 16 >>
|
|SLOT6OK
|%M00101 +-------------+
+--] [---+CALL   S6AOP +-
|        | (EXTERNAL)  |
|        |             |
| SLOT   |             | AOPSTS2
|%R00011-+X1         Y1+-%R00071
|        |             |
|AOUTX4  |             | AOP2IN
|%R00101-+X2         Y2+-%I01313
|        |             |
|        |             | AOP2STS
| CONST -+X3         Y3+-%R00072
| 0521   |             |
|        |             |
|        |             | AOP2BLD
| CONST -+X4         Y4+-%R00073
| 0001   |             |
|        |             |
|Reset s |             |
|tatus f |             |
|rom RUN |             |
|switch  |             |
|RSTSTS  |             | AOP2TAB
|%P00002-+X5         Y5+-%Q01313
|        |             |
| PRINP  |             | DUMMY2
|%P00004-+X6         Y6+-%R00075
|        +-------------+
|
| (*************************************************************************)
| (* Monitor Analog input 2 (static value) to make sure it never goes       *)
| (* outside the 5% range                                                   *)
| (*                                                                        *)
| (* Get data from both analog input cards starting at 33                  *)
| (*                                                                        *)
| (*************************************************************************)
|
| << RUNG 18 >>
|
|ALW_ON
|%S00007 +-----+
+--] [---+ AND_+-
|        | WORD|
|        |     |
|        |     |  TEMP1
|%R00112-+I1  Q+-%R00141
|        | LEN |
|        |00001|
|        |     |
| CONST -+I2   |
|   0FFF +-----+

Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1             Block: ANLGLB
```

```
05-03-02  18:25          GE FANUC SERIES 90-70 (v7.02)              Page    25
                               Reliabilty Test
                          Reliabilty test example 2

| << RUNG 19 >>
|
|ALW_ON
|%S00007 +-----+                 +-----+
+--] [---+ SUB_+-----------------+RANGE+-
|         | INT |                 | INT |
|         |     |                 |     |
|         |     |                 |     |                                  STATCNT
| TEMP1   |     | MONSTAT         |     |                                  %M00107
|%R00141-+I1  Q+-%R00211  CONST -+L1  Q+----------------------------------( )--
|         |     |           -00010 |     |
|         |     |                 |     |
| CONST -+I2   |          CONST -+L2   |
| +01000 +-----+          +00010 |     |
|                                 |     |
|                         MONSTAT |     |
|                         %R00211-+IN   |
|                                 +-----+
|
| << RUNG 20 >>
|
|STATCNT                                                                   STSOVR
|%M00107 +-----+                                                           %M01081
+--]/[---+ONDTR+-----------------------------------------------------------( )--
|         |1.00s|
|         |     |
|STSOVR   |     |
|%M01081  |     |
+--] [---+R    |
|         |     |
| CONST -+PV CV+-
| +32767 |     |
|         +-----+
|         Static
|         Analog
|         value
|         monitor
|         ERRCNT2
|         %R00321
|
```

```
Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1            Block: ANLGLB
```

```
05-03-02  18:26          GE FANUC SERIES 90-70 (v7.02)              Page    26
                               Reliabilty Test
                          Reliabilty test example 2

| << RUNG 21 >>
|
|STATCNT
|%M00107 +-----+
+--]/[---+MOVE_+-
|        | INT |
|        |     |
|        |     | Static
|        |     | Analog
|        |     | Err
|        |     | Value
| TEMP1  |     | TSTERR7
|%R00141-+IN  Q+-%R00507
|        | LEN |
|        |00001|
|        |     |
|        +-----+
|
| << RUNG 22 >>
|
|SLOT6OK
|%M00101 +-------------+
+--] [---+CALL   S6AIN +-
|        | (EXTERNAL)  |
|        |             |
| SLOT   |             | AINSTS
|%R00011-+X1         Y1+-%R00040
|        |             |
|        |             | AINBUF
|%AQ0033-+X2         Y2+-%R00111
|        |             |
|        |             | AIN1IOS
| CONST -+X3         Y3+-%R00041
| 0021   |             |
|        |             |
|        |             | AINSBTS
| CONST -+X4         Y4+-%R00131
| 0001   |             |
|        |             |
|Reset s |             |
|tatus f |             |
|rom RUN |             |
|switch  |             |
|RSTSTS  |             | AIN1BLD
|%P00002-+X5         Y5+-%R00043
|        |             |
| PRINP  |             | AIN1TAB
|%P00004-+X6         Y6+-%I00033
|        +-------------+
|


Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1            Block: ANLGLB
```

```
05-03-02  18:26          GE FANUC SERIES 90-70 (v7.02)              Page    27
                               Reliabilty Test
                          Reliabilty test example 2

| << RUNG 23 >>
|
|SLOT6OK
|%M00101 +-------------+
+--] [---+CALL   S6AIN +-
|        | (EXTERNAL)  |
|        |             |
| SLOT   |             | AIN2STS
|%R00011-+X1         Y1+-%R00045
|        |             |
|        |             | AIN2BUF
|%AQ0034-+X2         Y2+-%R00121
|        |             |
|        |             | AIN2IOS
| CONST -+X3         Y3+-%R00046
| 0441   |             |
|        |             |
|        |             | AINSBTS
| CONST -+X4         Y4+-%R00131
| 0001   |             |
|        |             |
|Reset s |             |
|tatus f |             |
|rom RUN |             |
|switch  |             |
|RSTSTS  |             | AIN1BLD
|%P00002-+X5         Y5+-%R00043
|        |             |
| PRINP  |             | AIN2TAB
|%P00004-+X6         Y6+-%AI0034
|        +-------------+
|
| (*************************************************************************)
| (* Monitor analog in and out 1 to ensure they are within the allowed      *)
| (* boundry range                                                          *)
| (*************************************************************************)
|
| << RUNG 25 >>
|
|ALW_ON
|%S00007 +-----+                +-----+
+--] [---+ MUL_+----------------+ DIV_+-
|        | INT |                | INT |
|        |     |                |     |
|        |     | 10%     10%    |     | 10%
|        |     | band    band   |     | band
|        |     | plus 10 plus 10|     | plus 10
|COUNT2  |     | MAXLIM  MAXLIM |     | MAXLIM
|%R00002-+I1  Q+-%R00411 %R00411-+I1  Q+-%R00411
|        |     |                |     |
| CONST -+I2   |          CONST -+I2   |
| +00010 +-----+          +00100 +-----+
|

Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1            Block: ANLGLB
```


| << RUNG 26 >>
|ALW_ON
|%S00007 +-----+ +-----+
+--] [---+ MUL_+----------+ SUB_+-
| INT | | INT |
| | | | 10%
|10% | | 10% | baqnd
|band | | band | minus
|plus 10 | | plus 10 | 10
|MAXLIM | | TEMP2 MAXLIM | MINLIM
|%R00411-+I1 Q+-%R00412 %R00411-+I1 Q+-%R00413
| | | TEMP2 | |
| CONST -+I2 | %R00412-+I2 |
| +00002 +-----+ +-----+
| << RUNG 27 >>
| +-----+ +-----+
+--------+ ADD_+----------+ SUB_+-
| INT | | INT |
| | 10% | | 10%
|10% | | 10% baqnd | | baqnd
|band | | band minus | | minus
|plus 10 | | plus 10 10 | | 10
|MAXLIM | | MAXLIM MINLIM | | MINLIM
|%R00411-+I1 Q+-%R00411 %R00413-+I1 Q+-%R00413
| | | | |
| CONST -+I2 | CONST -+I2 |
| +00020 +-----+ +00020 +-----+

```
05-03-02  18:26          GE FANUC SERIES 90-70 (v7.02)              Page    29
                               Reliabilty Test
                          Reliabilty test example 2

| << RUNG 28 >>
|
|ALW_ON
|%S00007 +-----+                 +-----+
+--] [---+ SUB_+-----------------+RANGE+-
|         | INT |                 | INT |
|         |     |                 |     |
|         |     |         10%     |     |
|         |     |         baqnd   |     |
|         |     |         minus   |     |
|         |     |         10      |     |                                  MECNT
|COUNT2   |     | MONAIO  MINLIM  |     |                                 %M00106
|%R00002-+I1  Q+-%R00201 %R00413-+L1  Q+-----------------------------------( )--
|         |     |                 |     |
|         |     |         10%     |     |
|         |     |         band    |     |
|         |     |         plus 10 |     |
|AINBUF   |     |         MAXLIM  |     |
|%R00111-+I2    |         %R00411-+L2   |
|         +-----+                 |     |
|                                 |     |
|                         MONAIO  |     |
|                         %R00201-+IN   |
|                                 +-----+
|
| << RUNG 29 >>
|
| MECNT                                                                MEOVR
|%M00106 +-----+                                                      %M01080
+--]/[---+ONDTR+------------------------------------------------------------( )--
|         |1.00s|
|         |     |
| MEOVR   |     |
|%M01080  |     |
+--] [---+R    |
|         |     |
| CONST -+PV CV+-
| +32767  |     |
|         +-----+
|         Analog
|         counter
|         error
|         check
|         ERRCNT3
|         %R00331
|

Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1            Block: ANLGLB
```

```
05-03-02  18:26          GE FANUC SERIES 90-70 (v7.02)              Page    30
                               Reliabilty Test
                          Reliabilty test example 2

| << RUNG 30 >>
|
| MECNT
|%M00106 +-----+                 +-----+
+--]/[---+MOVE_+-----------------+MOVE_+-
|        | INT |                 | INT |
|        |     |                 |     |
|        |     | Analog          |     | Analog
|        |     | Cnt OP          |     | Cnt IP
|        |     | Value           |     | Value
|COUNT2  |     | TSTERR8 AINBUF  |     | TSTERR9
|%R00002-+IN  Q+-%R00508 %R00111-+IN  Q+-%R00509
|        | LEN |                 | LEN |
|        |00001|                 |00001|
|        |     |                 |     |
|        +-----+                 +-----+
|
+[       END OF BLOCK LOGIC       ]
|
```

```
Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1            Block: ANLGLB
```

```
05-03-02  18:26          GE FANUC SERIES 90-70 (v7.02)              Page    31
                               Reliabilty Test
                          Reliabilty test example 2

+[  START  OF  LD  BLOCK  SCAN    ]
|
||     VARIABLE DECLARATIONS      ]

             V A R I A B L E   D E C L A R A T I O N   T A B L E

          REFERENCE     NICKNAME          REFERENCE DESCRIPTION
          ---------     --------     -------------------------------
                          NO  VARIABLE  TABLE  ENTRIES


                        I D E N T I F I E R    T A B L E

          IDENTIFIER     IDENTIFIER TYPE          IDENTIFIER DESCRIPTION
          ----------     ---------------     --------------------------------
                         NO  IDENTIFIER  TABLE  ENTRIES


+[      START OF BLOCK LOGIC      ]
|
| (*************************************************************************)
| (* Scan digital I/O on both channels                                     *)
| (*************************************************************************)
|

Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1            Block: SCAN
```

```
05-03-02  18:26          GE FANUC SERIES 90-70 (v7.02)              Page    32
                               Reliabilty Test
                          Reliabilty test example 2

| << RUNG 4 >>
|
|SLOT6OK                                                                SCN1OK
|%M00101 +-------------+                                               %M00003
+--] [---+CALL  S6SCAN +-----------------------------------------------------( )--
|        | (EXTERNAL)  |
|        |             |
| SLOT   |             | SCNSTS
|%R00011-+X1         Y1+-%R00051
|        |             |
|OPTAB1L |             | INTABL
|%Q00001-+X2         Y2+-%I00001
|        |             |
|        |             | SCN1IOS
| CONST -+X3         Y3+-%R00053
| 0081   |             |
|        |             |
|        |             | SCN1BLD
| CONST -+X4         Y4+-%R00054
| 00A1   |             |
|        |             |
|Reset s |             |
|tatus f |             |
|rom RUN |             |
|switch  |             |
|RSTSTS  |             | DUMMY3
|%P00002-+X5         Y5+-%R00056
|        |             |
| PRINP  |             | DUMMY4
|%P00004-+X6         Y6+-%R00057
|        +-------------+
|
```

```
Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1            Block: SCAN
```

```
05-03-02  18:26          GE FANUC SERIES 90-70 (v7.02)              Page    33
                               Reliabilty Test
                          Reliabilty test example 2

| << RUNG 5 >>
|
|SLOT6OK                                                                SCN2OK
|%M00101 +-------------+                                               %M00004
+--] [---+CALL  S6SCAN +-----------------------------------------------------( )--
|        | (EXTERNAL)  |
|        |             |
| SLOT   |             | SCNSTS2
|%R00011-+X1         Y1+-%R00061
|        |             |
|COUNT1  |             |
|%R00001-+X2         Y2+-%I01025
|        |             |
|        |             | SCN2IOS
| CONST -+X3         Y3+-%R00063
| 0401   |             |
|        |             |
|        |             | SCN2BLD
| CONST -+X4         Y4+-%R00064
| 0421   |             |
|        |             |
|Reset s |             |
|tatus f |             |
|rom RUN |             |
|switch  |             |
|RSTSTS  |             | DUMMY3
|%P00002-+X5         Y5+-%R00056
|        |             |
|        |             | DUMMY4
| CONST -+X6         Y6+-%R00057
| 0070   +-------------+
|
+[       END OF BLOCK LOGIC       ]
|

Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1            Block: SCAN
```

```
05-03-02  18:26          GE FANUC SERIES 90-70 (v7.02)              Page    34
                               Reliabilty Test
                          Reliabilty test example 2

+[  START  OF  LD  BLOCK CCMTST   ]
|
||     VARIABLE DECLARATIONS      ]

             V A R I A B L E   D E C L A R A T I O N   T A B L E

          REFERENCE     NICKNAME          REFERENCE DESCRIPTION
          ---------     --------     -------------------------------
          %Q00961       Q_S6CCM      S6IOCCM Output Address
          %M00001       S6_6_OK      S6PIO in slot 6 OK
          %M00081       WT_CCM       Write serial to IOCCM
          %M00082       RD_CCM       Read serial IOCCM
          %M00119       SAMPLE       Indication to start sampling
          %T00013       CCMCOMP      IOCCM command status complete
          %T00014       CCMBUSY      IOCCM busy
          %T00015       CCMXFER      IOCCM serial transfer error
          %T00018       CCMWCOM      IOCCM Write cmd complete
          %T00019       CCMRCOM      IOCCM Read cmd complete
          %L00031       REGPADD      S6PIO Register Parameter Address
          %R00035       WIN1STS      IOCCM return status
          %R00281       XFERCNT      IOCCM Xfer complete with error
          %R00291       WIN1TO       Window Timeout error count
          %R00361       DIAGDAT      IOCCM Diagnostics data value
          %R00371       ERRDATA      Xfer error data value
          %R00381       ERRCODE      J1 Port Error code for diag data
          %R00391       CCMSTS       IOCCM status data
          %R00711       BITREF
          %R00800       SCNCNT       Scan counter
          %R01101       RDREG
          %R01520       BITNO
          %R01521       OUTCPY
          %R03011       WRTCMP       Write compare register


                        I D E N T I F I E R    T A B L E

          IDENTIFIER     IDENTIFIER TYPE          IDENTIFIER DESCRIPTION
          ----------     ---------------     -------------------------------
                         NO  IDENTIFIER  TABLE  ENTRIES


+[      START OF BLOCK LOGIC      ]
|
| (*************************************************************************)
| (* Clear error counters on first scan                                    *)
| (*************************************************************************)
|


Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1            Block: CCMTST
```

```
05-03-02  18:26          GE FANUC SERIES 90-70 (v7.02)              Page    35
                               Reliabilty Test
                          Reliabilty test example 2

| << RUNG 4 >>
|
|FST_SCN
|%S00001         +-----+                +-----+                +-----+
+--] [-----------+MOVE_+----------------+MOVE_+----------------+MOVE_+-
|                | INT |                | INT |                | INT |
|                |     |                |     |                |     |
|                |     | IOCCM s        |     | Window         |     | IOCCM X
|                |     | erial l        |     | error c        |     | fer com
|                |     | oopback        |     | ount fo        |     | plete w
|                |     | er coun        |     | r IOCCM        |     | ith err
|                |     | ERRCNT5        |     | ERRCNT4        |     | XFERCNT
|         CONST -+IN  Q+-%R00351  CONST -+IN  Q+-%R00341  CONST -+IN  Q+-%R00281
|         +00000 | LEN |          +00000 | LEN |          +00000 | LEN |
|                |00001|                |00001|                |00001|
|                |     |                |     |                |     |
|                +-----+                +-----+                +-----+
|
| << RUNG 5 >>
|
|FST_SCN
|%S00001         +-----+                +-----+                +-----+
+--] [-----------+MOVE_+----------------+MOVE_+----------------+MOVE_+-
|                | INT |                | INT |                | INT |
|                |     |                |     |                |     |
|                |     | Xfer           |     | Window         |     |
|                |     | error          |     | Timeout        |     |
|                |     | data           |     | error          |     | Scan
|                |     | value          |     | count          |     | counter
|                |     | ERRDATA        |     | WIN1TO         |     | SCNCNT
|         CONST -+IN  Q+-%R00371  CONST -+IN  Q+-%R00291  CONST -+IN  Q+-%R00800
|         +00000 | LEN |          +00000 | LEN |          +00000 | LEN |
|                |00001|                |00001|                |00001|
|                |     |                |     |                |     |
|                +-----+                +-----+                +-----+
|
| << RUNG 6 >>
|
|FST_SCN
|%S00001 +-----+                +-----+                +-----+
+--] [---+MOVE_+----------------+MOVE_+----------------+MOVE_+-
|        | INT |                | INT |                | INT |
|        |     |                |     |                |     |
|        |     | BITREF         |     |  BITNO         |     | OUTCPY
| CONST -+IN  Q+-%R00711  CONST -+IN  Q+-%R01520  CONST -+IN  Q+-%R01521
| +00000 | LEN |          +00000 | LEN |          +00000 | LEN |
|        |00001|                |00001|                |00001|
|        |     |                |     |                |     |
|        +-----+                +-----+                +-----+
|
| (*************************************************************************)
| (* Start with write serial data until status complete then read serial     *)
| (* data until complete and repeat.                                         *)
| (*************************************************************************)

Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1             Block: CCMTST
```

```
05-03-02  18:26          GE FANUC SERIES 90-70 (v7.02)              Page    36
                               Reliabilty Test
                          Reliabilty test example 2

| << RUNG 8 >>
|
|FST_SCN                                                                WT_CCM
|%S00001                                                                %M00081
+--] [----------+-----------------------------------------------------------( )--
|               |
|Write   IOCCM W|
|serial  rite cm|
|to      d compl|
|IOCCM   ete    |
|WT_CCM  CCMWCOM|
|%M00081 %T00018|
+--] [-----]/[--+
|               |
|        IOCCM R|
|Read    ead cmd|
|serial  complet|
|IOCCM   e      |
|RD_CCM  CCMRCOM|
|%M00082 %T00019|
+--] [-----] [--+
|
| << RUNG 9 >>
|
|Write
|serial
|to
|IOCCM
|WT_CCM                                                                 RD_CCM
|%M00081                                                                %M00082
+--]/[-----------------------------------------------------------------------( )--
|
| (*************************************************************************)
| (* If pulsed to write to the IOCCM then call DATATST to write some random   *)
| (* data into buffer memory for transmission                                 *)
| (*************************************************************************)
|
```

```
Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1            Block: CCMTST
```

```
| << RUNG 11 >>
|
|Write
|serial
|to
|IOCCM
|WT_CCM
|%M00081 +-------------+
+--] [---+CALL  DATATST+-
|        | (EXTERNAL)  |
|        |             |
|        |             | WRTREG
| CONST -+X1         Y1+-%R00968
| 0002   |             |
|        |             |
| CONST -+X2         Y2+-
| 0080   |             |
|        |             |
| CONST -+X3         Y3+-
| 0000   |             |
|        |             |
| CONST -+X4         Y4+-
| 0000   +-------------+
|
| << RUNG 12 >>
|
|Write
|serial
|to
|IOCCM
|WT_CCM
|%M00081 +-----+
+--]^[---+MOVE_+-
|        | INT |
|        |     |
|        |     | Write
|        |     | compare
|        |     | registe
|        |     | r
|WRTREG  |     | WRTCMP
|%R00968-+IN  Q+-%R03011
|        | LEN |
|        |00064|
|        |     |
|        +-----+
|
| (*************************************************************************)
| (* The IOCCM will be sent an internal NOOP command between read and write   *)
| (* commands to prevent overflow in the IOCCM. The command parameters        *)
| (* contain the command number only.                                         *)
| (* Location 1 = 6000   NOOP                                                *)
| (*************************************************************************)
|
```

```
05-03-02  18:26          GE FANUC SERIES 90-70 (v7.02)              Page    38
                               Reliabilty Test
                          Reliabilty test example 2

| << RUNG 14 >>
|
|        +-----+
+--------+DATA_+-
|        |INIT_|
|        | INT |
|        |     |
|        |     | IOCCMAD
|        |    Q+-%R00961
|        | LEN |
|        |00001|
|        |     |
|        +-----+
|
|DATA INIT INT at: %R00961
|   1       +06000
|
| (*************************************************************************)
| (* Set up for IOCCM cmd Read From Target To  Source Register Table       *)
| (* Location 1 = 6101   = Command No to read from target to source reg table *)
| (* Location 2 = 5      = Target Id                                       *)
| (* Location 3 = 1      = Target Memory Type                              *)
| (* Location 4 = 3011   = Target Memory Address on S6                     *)
| (* Location 5 = 64     = Data Length (64 registers)                      *)
| (*                                                                       *)
| (* Location 6 = 1101   = Source memory Address                          *)
| (*************************************************************************)
|
| << RUNG 16 >>
|
|Read
|serial
|IOCCM
|RD_CCM
|%M00082 +-----+
+--]^[---+DATA_+-
|        |INIT_|
|        | INT |
|        |     |
|        |     | IOCCMAD
|        |    Q+-%R00961
|        | LEN |
|        |00006|
|        |     |
|        +-----+
|
|DATA INIT INT at: %R00961
|   1       +06101        +00005        +00001        +03011        +00064
|   6       +01101
|


Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1             Block: CCMTST
```

```
05-03-02  18:26          GE FANUC SERIES 90-70 (v7.02)              Page    39
                               Reliabilty Test
                          Reliabilty test example 2

| (*************************************************************************)
| (* Set-up Register parameter block for read command.                      *)
| (* 961 = register parameter block address in S6PIO memory                 *)
| (* 6 = number of reisters to copy                                         *)
| (*************************************************************************)
|
| << RUNG 18 >>
|
|Read
|serial
|IOCCM
|RD_CCM
|%M00082 +-----+
+--] [---+DATA_+-
|        |INIT_|
|        | INT |
|        |     | S6PIO R
|        |     | egister
|        |     | Paramet
|        |     | er Addr
|        |     | REGPADD
|        |    Q+-%L00031
|        | LEN |
|        |00002|
|        |     |
|        +-----+
|
|DATA INIT INT at: %L00031
|   1       +00961        +00006
|
| (*************************************************************************)
| (* Set up for IOCCM cmd Write From Target To Source Register Table       *)
| (* Location 1 = 6111   = Command No.                                     *)
| (* Location 2 = 5      = Target Id                                      *)
| (* Location 3 = 1      = Target Memory Type                             *)
| (* Location 4 = 3011   = Target Memory Address                          *)
| (* Location 5 = 64     = Data Length                                    *)
| (* Location 6 = 968    = Source memory Address                          *)
| (*************************************************************************)
|

Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1            Block: CCMTST
```

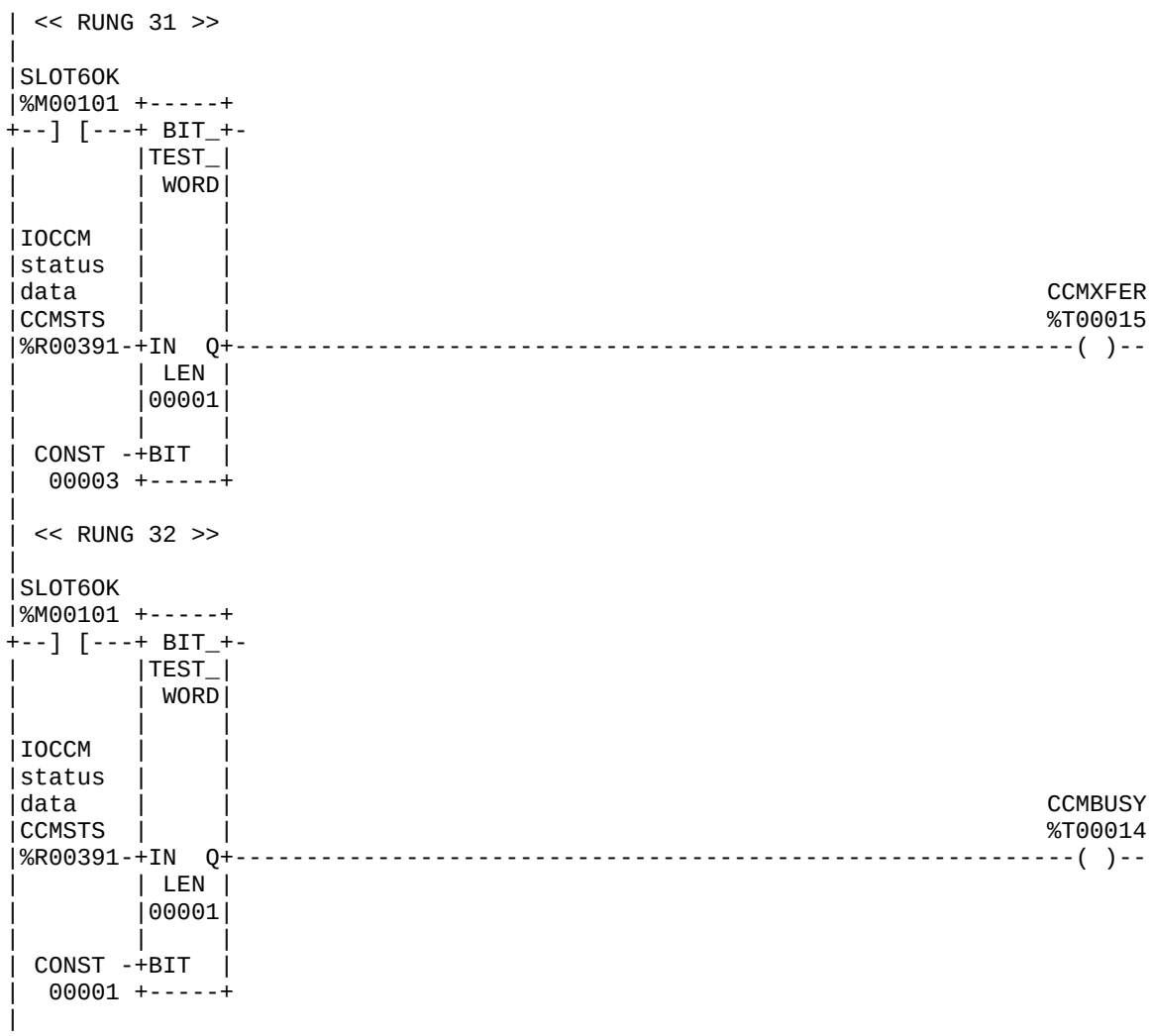
```
05-03-02  18:26          GE FANUC SERIES 90-70 (v7.02)              Page    40
                               Reliabilty Test
                          Reliabilty test example 2

| << RUNG 20 >>
|
|Write
|serial
|to
|IOCCM
|WT_CCM
|%M00081 +-----+
+--]^[---+DATA_+-
|        |INIT_|
|        | INT |
|        |     |
|        |     | IOCCMAD
|        |    Q+-%R00961
|        | LEN |
|        |00006|
|        |     |
|        +-----+
|
|DATA INIT INT at: %R00961
|   1       +06111        +00005        +00001        +03011        +00064
|   6       +00968
|
| (*************************************************************************)
| (* Set-up register parameter block                                       *)
| (* 961 = register parameter address in S6PIO where data will be moved to  *)
| (* 71 = 64 regs of write data + 7 regs of command data                   *)
| (*************************************************************************)
|
| << RUNG 22 >>
|
|Write
|serial
|to
|IOCCM
|WT_CCM
|%M00081 +-----+
+--] [---+DATA_+-
|        |INIT_|
|        | INT |
|        |     | S6PIO R
|        |     | egister
|        |     | Paramet
|        |     | er Addr
|        |     | REGPADD
|        |    Q+-%L00031
|        | LEN |
|        |00002|
|        |     |
|        +-----+
|
|DATA INIT INT at: %L00031
|   1       +00961        +00071
|

Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1            Block: CCMTST
```

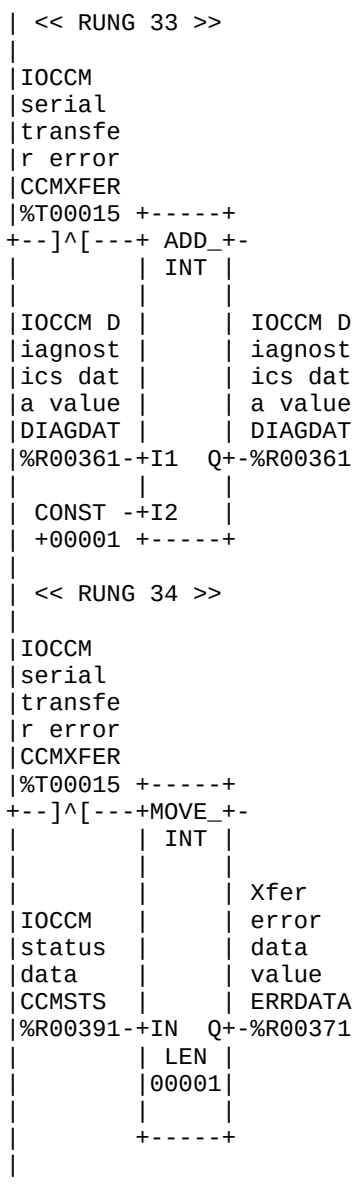
```
05-03-02  18:26          GE FANUC SERIES 90-70 (v7.02)              Page    41
                               Reliabilty Test
                          Reliabilty test example 2

| << RUNG 23 >>
|
|IOCCM
|serial  Read
|transfe serial
|r error IOCCM
|CCMXFER RD_CCM
|%T00015 %M00082 +-----+
+--]^[-----] [---+DATA_+-
|                |INIT_|
|                | INT |
|                |     |
|                |     | IOCCMAD
|                |    Q+-%R00961
|                | LEN |
|                |00006|
|                |     |
|                +-----+
|
|DATA INIT INT at: %R00961
|   1       +06003        +00000        +00000        +00001        +00020
|   6       +00361
|
| (*****************************************************************************)
| (* Clear Diagnostics data on first scan                                      *)
| (*                                                                           *)
| (*****************************************************************************)
|
| << RUNG 25 >>
|
|FST_SCN
|%S00001 +-----+
+--] [---+DATA_+-
|        |INIT_|
|        | INT |
|        |     |
|        |     | IOCCMAD
|        |    Q+-%R00961
|        | LEN |
|        |00001|
|        |     |
|        +-----+
|
|DATA INIT INT at: %R00961
|   1       +06002
|

Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1            Block: CCMTST
```

```
| (*************************************************************************)
| (* Open window to IOCCM card at address location 961. Command block will    *)
| (* be picked up by the IOCCM and executed.                                  *)
| (* The IOCCM status will be returned at parameter Y6 and used to indicate   *)
| (* when the command is complete.                                            *)
| (*                                                                         *)
| (* The error status will be checked for a window timeout and a counter      *)
| (* will be incremented each time a timeout occurs.                         *)
| (*************************************************************************)
|
```

```
05-03-02  18:26          GE FANUC SERIES 90-70 (v7.02)              Page    43
                               Reliabilty Test
                          Reliabilty test example 2

| << RUNG 27 >>
|
|        Write
|        serial
|        to
|        IOCCM
|SLOT6OK WT_CCM                                                        WIN1OK
|%M00101 %M00081                      +-------------+                 %M00007
+--] [--+--] [--+---------------------+CALL  S6XWIN +-----------------( )--
|       |       |                     | (EXTERNAL)  |
|       |       |                     |             |
|       |Read   |                     |             |
|       |serial |                     |             |
|       |IOCCM  |                     |             |
|       |RD_CCM |                     |             |
|       |%M00082|                     |             | WIN1ERR
|       +--] [--+             CONST -+X1         Y1+-%R00021
|                             4006   |             |
|                                    |             |
|                                    |             | IOCCM
|                                    |             | return
|                                    |             | status
|                                    |             | WIN1STS
|                             CONST -+X2         Y2+-%R00035
|                             07C1   |             |
|                                    |             |
|                                    |             | WIN1HDR
|                             CONST -+X3         Y3+-%R00022
|                             0032   |             |
|                                    |             |
|                                    |             | WIN1BLD
|                             CONST -+X4         Y4+-%R00023
|                             002D   |             |
|                                    |             |
|                            S6PIO R |             |
|                            egister |             | S6IOCCM
|                            Paramet |             | Output
|                            er Addr |             | Address
|                            REGPADD |             | Q_S6CCM
|                            %L00031-+X5         Y5+-%Q00961
|                                    |             |
|                                    |             | IOCCM
|                                    |             | status
|                                    |             | data
|                            IOCCMAD |             | CCMSTS
|                            %R00961-+X6         Y6+-%R00391
|                                    +-------------+
|

Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1            Block: CCMTST
```

```
05-03-02  18:26          GE FANUC SERIES 90-70 (v7.02)              Page    44
                               Reliabilty Test
                          Reliabilty test example 2

| << RUNG 28 >>
|
|ALW_ON
|%S00007 +-----+
+--] [---+ EQ_ +-
|        | INT |
|        |     |
|WIN1ERR |     |         +-----+
|%R00021-+I1  Q+---------+ ADD_+-
|        |     |         | INT |
|        |     |         |     |
|        |     |         |     | Window
|        |     |         |     | Timeout
|        |     |         |     | error
|        |     |         |     | count
|        |     |         |     | WIN1TO
| CONST -+I2   |  CONST -+I1  Q+-%R00291
| +00114 +-----+  +00001 |     |
|                         |     |
|                Window   |     |
|                Timeout  |     |
|                error    |     |
|                count    |     |
|                WIN1TO   |     |
|                %R00291-+I2   |
|                         +-----+
|
| (*************************************************************************)
| (* Test bit 2 of the status byte for command complete without error.     *)
| (* When complete clear status register for next command.                 *)
| (*************************************************************************)
|
| << RUNG 30 >>
|
|SLOT6OK
|%M00101 +-----+
+--] [---+ BIT_+-
|        |TEST_|
|        | WORD|
|        |     |
|IOCCM   |     |
|status  |     |
|data    |     |                                                  CCMCOMP
|CCMSTS  |     |                                                  %T00013
|%R00391-+IN  Q+-----------------------------------------------------( )--
|        | LEN |
|        |00001|
|        |     |
| CONST -+BIT  |
|  00002 +-----+
|


Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1            Block: CCMTST
```


| << RUNG 31 >>
|SLOT6OK
|%M00101 +-----+
+--] [---+ BIT_+-
|TEST_|
| WORD|
|IOCCM
|status
|data
|CCMSTS
|%R00391-+IN Q+
| LEN |
|00001|
| CONST -+BIT |
| 00003 +-----+
CCMXFER
%T00015
( )--
| << RUNG 32 >>
|SLOT6OK
|%M00101 +-----+
+--] [---+ BIT_+-
|TEST_|
| WORD|
|IOCCM
|status
|data
|CCMSTS
|%R00391-+IN Q+
| LEN |
|00001|
| CONST -+BIT |
| 00001 +-----+
CCMBUSY
%T00014
( )--


| << RUNG 33 >>
|IOCCM
|serial
|transfe
|r error
|CCMXFER
|%T00015 +-----+
+--]^[---+ ADD_+-
| INT |
|IOCCM D |
|iagnost |
|ics dat |
|a value |
|DIAGDAT |
|%R00361-+I1 Q+-%R00361
| CONST -+I2 |
| +00001 +-----+
| << RUNG 34 >>
|IOCCM
|serial
|transfe
|r error
|CCMXFER
|%T00015 +-----+
+--]^[---+MOVE_+-
| INT |
| Xfer
|IOCCM |
|status |
|data |
|CCMSTS |
|%R00391-+IN Q+-%R00371
| LEN |
|00001|

```
| << RUNG 35 >>
|
|        IOCCM c
|        ommand
|        status
|        complet
|        CCMCOMP
|        %T00013         +-----+
+-------+--]^[--+--------+MOVE_+-
|       |       |        | INT |
|       |IOCCM  |        |     |
|       |serial |        |     |
|       |transfe|        |     | IOCCM
|       |r error|        |     | status
|       |CCMXFER|        |     | data
|       |%T00015|        |     | CCMSTS
|       +--]^[--+ CONST -+IN  Q+-%R00391
|                 +00000 | LEN |
|                        |00001|
|                        |     |
|                        +-----+
|
| (*************************************************************************)
| (* Check the window return error status for any errors and record in the    *)
| (* error counter register.                                                  *)
| (*************************************************************************)
|
| << RUNG 37 >>
|
|SLOT6OK
|%M00101 +-----+
+--] [---+ NE_ +-
|        | INT |
|        |     |
|WIN1ERR |     |         +-----+
|%R00021-+I1  Q+---------+ ADD_+-
|        |     |         | INT |
|        |     |         |     |
|        |     | Window  |     | Window
|        |     | error c |     | error c
|        |     | ount fo |     | ount fo
|        |     | r IOCCM |     | r IOCCM
|        |     | ERRCNT4 |     | ERRCNT4
| CONST -+I2   | %R00341-+I1  Q+-%R00341
| +00000 +-----+         |     |
|                        |     |
|                 CONST -+I2   |
|                 +00001 +-----+
|
| (*************************************************************************)
| (* If there has been a transfer error, read in the status diagnostics      *)
| (* to determine what caused the failure.                                   *)
| (*************************************************************************)
|
```

```
05-03-02  18:26          GE FANUC SERIES 90-70 (v7.02)              Page    48
                               Reliabilty Test
                          Reliabilty test example 2

| << RUNG 39 >>
|
|                IOCCM
|        Read    serial
|        serial  transfe
|        IOCCM   r error
|SLOT6OK RD_CCM  CCMXFER
|%M00101 %M00082 %T00015 +-----+                 +-------------+
+--] [-----] [-----] [---+DATA_+-----------------+CALL  S6MOVE +-
|                        |INIT_|                 | (EXTERNAL)  |
|                        | INT |                 |             |
|                        |     |                 |             |
|                        |     | MEM1TYP         |             | MOV1ERR
|                        |    Q+-%R00082  CONST -+X1         Y1+-%R00081
|                        | LEN |          0006   |             |
|                        |00003|                 |             |
|                        |     |                 |             |
|                        |     |                 |             | MEM1TYP
|                        +-----+          CONST -+X2         Y2+-%R00082
|                                         0005   |             |
|                                                |             |
|                                                |             | MEM1OFF
|                                         CONST -+X3         Y3+-%R00083
|                                         0168   |             |
|                                                |             |
|                                                |             | MEM1ID
|                                         CONST -+X4         Y4+-%R00084
|                                         0000   |             |
|                                                |             |
|                                                |             | MOV1BLD
|                                         CONST -+X5         Y5+-%R00085
|                                         0028   +-------------+
|
|DATA INIT INT at: %R00082
|   1       +00008        +00720        +00001
|

Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1            Block: CCMTST
```

```
05-03-02  18:26          GE FANUC SERIES 90-70 (v7.02)              Page    49
                               Reliabilty Test
                          Reliabilty test example 2

| << RUNG 40 >>
|
|IOCCM
|serial
|transfe
|r error
|CCMXFER
|%T00015 +-----+
+--] [---+ GT_ +-
|        | INT |
|        |     |
|IOCCM D |     |
|iagnost |     |
|ics dat |     |
|a value |     |
|DIAGDAT |     |         +-----+
|%R00361-+I1  Q+---------+MOVE_+-
|        |     |         | INT |
|        |     |         |     |
|        |     | IOCCM D |     | J1 Port
|        |     | iagnost |     | Error c
|        |     | ics dat |     | ode for
|        |     | a value |     | diag da
|        |     | DIAGDAT |     | ERRCODE
| CONST -+I2   | %R00361-+IN  Q+-%R00381
| +00000 +-----+         | LEN |
|                        |00001|
|                        |     |
|                        +-----+
|
| (*************************************************************************)
| (* If we have just executed a read serial data and it has completed,     *)
| (* then copy the data from S6PIO to 90-70 register memory.               *)
| (*                                                                       *)
| (* Set up parameter block for S6MOVE to move IOCCM data back to 90-70.    *)
| (*                                                                       *)
| (* X1 = 6     = S6PIO slot number                                        *)
| (* X2 = 5     = Source memory type = S6PIO register memory               *)
| (* X3 = 1100  = Source memory word offset 1100 = R1101                   *)
| (* x4 = 0     = Source device ID = S6PIO                                 *)
| (* x5 = 128   = Number of bytes to transfer                             *)
| (* y2 = 8     = Target memory type = 90-70 register memory               *)
| (* y3 = 2200  = Target memory byte offset 2200 = %R1101                  *)
| (* y4 = 1     = Target device ID = 90-70                                 *)
| (*************************************************************************)
|

Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1            Block: CCMTST
```

```
05-03-02  18:26          GE FANUC SERIES 90-70 (v7.02)              Page    50
                               Reliabilty Test
                          Reliabilty test example 2

| << RUNG 42 >>
|
|        Read
|        serial
|        IOCCM
|SLOT6OK RD_CCM
|%M00101 %M00082 +-----+                 +-------------+
+--] [-----] [---+DATA_+-----------------+CALL  S6MOVE +-
|                |INIT_|                 | (EXTERNAL)  |
|                | INT |                 |             |
|                |     |                 |             |
|                |     | MEM2TYP         |             | MOV2ERR
|                |    Q+-%R00092  CONST -+X1         Y1+-%R00091
|                | LEN |          0006   |             |
|                |00003|                 |             |
|                |     |                 |             |
|                |     |                 |             | MEM2TYP
|                +-----+          CONST -+X2         Y2+-%R00092
|                                 0005   |             |
|                                        |             |
|                                        |             | MEM2OFF
|                                 CONST -+X3         Y3+-%R00093
|                                 044C   |             |
|                                        |             |
|                                        |             | MEM2ID
|                                 CONST -+X4         Y4+-%R00094
|                                 0000   |             |
|                                        |             |
|                                        |             | MOV2BLD
|                                 CONST -+X5         Y5+-%R00095
|                                 0080   +-------------+
|
|DATA INIT INT at: %R00092
|   1       +00008        +02200        +00001
|
| (*************************************************************************)
| (* When writing/reading data has returned correct status pulse           *)
| (* command complete.                                                     *)
| (*************************************************************************)
|
| << RUNG 44 >>
|
|        Write   IOCCM c
|        serial  ommand
|        to      status  IOCCM
|        IOCCM   complet busy
|SLOT6OK WT_CCM  CCMCOMP CCMBUSY                                    CCMWCOM
|%M00101 %M00081 %T00013 %T00014                                   %T00018
+--] [-----] [-----]^[-----]/[-----------------------------------------(^)--
|

Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1            Block: CCMTST
```

```
05-03-02  18:26          GE FANUC SERIES 90-70 (v7.02)              Page    51
                               Reliabilty Test
                          Reliabilty test example 2

| << RUNG 45 >>
|
|                IOCCM c
|        Read    ommand
|        serial  status  IOCCM
|        IOCCM   complet busy
|SLOT6OK RD_CCM  CCMCOMP CCMBUSY                                        CCMRCOM
|%M00101 %M00082 %T00013 %T00014                                       %T00019
+--] [-----] [-----]^[-----]/[-----------------------------------------(^)--
|
| (*************************************************************************)
| (* Monitor serial IOCCM data loopback after 20 scans to ensure start-up     *)
| (* errors are not indicated.                                               *)
| (* Each write word is compared against the read word.                      *)
| (*************************************************************************)
|
| << RUNG 47 >>
|
|IOCCM R Indicat
|ead cmd ion to
|complet start s
|e       ampling
|CCMRCOM SAMPLE
|%T00019 %M00119         +-----+
+--] [-----] [-----------+MASK_+-
|                        |COMP_|
|                        | WORD|
|                Write   |     |
|                compare |     |
|                registe |     |
|                r       |     |                                        CMPERR
|                WRTCMP  |     |                  +-----+               %M00181
|                %R03011-+I1 MC+-----------------+ ADD_+-------------------(^)--
|                        | LEN |                  | INT |
|                        |00064|                  |     |
|                        |     |         IOCCM s  |     | IOCCM s
|                        |     |         erial l  |     | erial l
|                        |     |         oopback  |     | oopback
|                        |     |         er coun  |     | er coun
|                 RDREG  |     | OUTCPY  ERRCNT5  |     | ERRCNT5
|                %R01101-+I2  Q+-%R01521 %R00351-+I1  Q+-%R00351
|                        |     |                  |     |
|                BITREF  |     |  BITNO           |     |
|                %R00711-+M  BN+-%R01520  CONST -+I2   |
|                        |     |          +00001 +-----+
|                        |     |
|                 CONST -+BIT  |
|                  00000 +-----+
|

Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1             Block: CCMTST
```

```
05-03-02  18:26          GE FANUC SERIES 90-70 (v7.02)              Page    52
                               Reliabilty Test
                          Reliabilty test example 2

| << RUNG 48 >>
|
|Mis com
|pare er
|ror for
|IOCCM
|CMPERR
|%M00181 +-----+                 +-----+
+--]^[---+MOVE_+-----------------+MOVE_+-
|        | INT |                 | INT |
|        |     |                 |     |
|Write   |     | Wrt com         |     | RD comp
|compare |     | pare va         |     | are val
|registe |     | lue on          |     | ue on e
|r       |     | error I         |     | rror IO
|WRTCMP  |     | TSTERR1  RDREG  |     | TSTERR2
|%R03011-+IN  Q+-%R00501 %R01101-+IN  Q+-%R00502
|        | LEN |                 | LEN |
|        |00001|                 |00001|
|        |     |                 |     |
|        +-----+                 +-----+
|
| << RUNG 49 >>
|
|SLOT6OK
|%M00101 +-----+                                  +-----+
+--] [---+ EQ_ +---------------------------------+ ADD_+-
|        | INT |                                 | INT |
|        |     |                                 |     |
|Scan    |     |                         Scan    |     | Scan
|counter |     |                         counter |     | counter
|SCNCNT  |     |         +-----+         SCNCNT  |     | SCNCNT
|%R00800-+I1  Q+---------+MOVE_+-        %R00800-+I1  Q+-%R00800
|        |     |         | INT |                 |     |
|        |     |         |     |                 |     |
|        |     |         |     | Scan            |     |
|        |     |         |     | counter         |     |
|        |     |         |     | SCNCNT          |     |
| CONST -+I2   |  CONST -+IN  Q+-%R00800  CONST -+I2   |
| +32767 +-----+  +00025 | LEN |          +00001 +-----+
|                        |00001|
|                        |     |
|                        +-----+
|

Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1            Block: CCMTST
```

```
05-03-02  18:26          GE FANUC SERIES 90-70 (v7.02)              Page    53
                               Reliabilty Test
                          Reliabilty test example 2

| << RUNG 50 >>
|
|SLOT6OK
|%M00101 +-----+
+--] [---+ GE_ +-
|        | INT |
|        |     |
|Scan    |     |
|counter |     |                                                      SAMPLE
|SCNCNT  |     |                                                      %M00119
|%R00800-+I1  Q+------------------------------------------------------------( )--
|        |     |
| CONST -+I2   |
| +00020 +-----+
|
+[       END OF BLOCK LOGIC       ]
|
```

```
Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1            Block: CCMTST
```

```
05-03-02  18:26          GE FANUC SERIES 90-70 (v7.02)              Page    54
                               Reliabilty Test
                          Reliabilty test example 2


THE LOGIC FOR THE FOLLOWING EXTERNAL BLOCKS CANNOT BE PRINTED:
          S6AOP
          S6AIN
          S6SCAN
          S6NOOP
          S6EVTLG
          S6MOVE
          S6XWIN
          DATATST
          S6STAT
          S6OPT
          S6WIN




Program: RELIAB1            C:\LM90\FOLDERS\RELIAB1
```

***** L O G I C T A B L E O F C O N T E N T S *****

Example 3: Demonstrate Use of the S6PSV Block

This example program illustrates the use of the S6PSV C Block. The example is intended to be used as a guide when constructing real world applications. This example makes the following assumptions:

1 A single S6PIO module is used in series with a Series Six Primary I/O Chain.
2 Normal mode scanning is used.
3 The S6PIO module is installed in slot 6 of the CPU rack.

Program Listing

```
04-22-02  15:02           GE FANUC SERIES 90-70 (v7.02)              Page     1
                                    EXAMPL3
         Example of using the S6PIO to only monitor the S6 I/O activity

|[     START OF PROGRAM LOGIC     ]
|
| (*************************************************************************)
| (* Sample program for a Passive Mode configuration using an S6PIO         *)
| (* module in the 90-70 CPU rack to monitor the I/O activity of             *)
| (* a single Series Six I/O chain.                                          *)
| (*************************************************************************)
|
| (*************************************************************************)
| (* Test to see if an S6PIO card exists in Slot 6 and is healthy.           *)
| (*************************************************************************)
|
| << RUNG 7 >>
|
|ALW_ON                                                              SLOT6OK
|%S00007         +-------------+                                     %M00106
+--] [-----------+CALL  S6NOOP +-----------------------------------------( )--
|                | (EXTERNAL)  |
|                |             |
|                |             | SL6STAT
|          CONST -+X1         Y1+-%P00006
|          0006   +-------------+
|
```

Program: EXAMPL3 C:\LM90\FOLDERS\EXAMPL3 Block: _MAIN

```
04-22-02  15:02          GE FANUC SERIES 90-70 (v7.02)              Page     2
                                  EXAMPL3
         Example of using the S6PIO to only monitor the S6 I/O activity

| (*************************************************************************)
| (* If the S6PIO module is present and OK, load the various parameters       *)
| (* and call the S6PSV block.                                                *)
| (*                                                                         *)
| (* The 90-70 location pointed to by X4 will contain the status of the       *)
| (* 1000 input bits of the Series Six I/O chain being monitored.             *)
| (*                                                                         *)
| (* the 90-70 location pointed to by X5 will contain the status of the       *)
| (* 1000 output bits of the Series Six I/O chain being monitored.            *)
| (*                                                                         *)
| (* The 8 register locations pointed to by Y2 contain the Card Present       *)
| (* Map showing where each Series Six input card is located. One 90-70       *)
| (* bit for each 8 bit Series Six address, hence a 32 bit input module      *)
| (* will show 4 bits set in the map. So bit 1 of register 1 covers          *)
| (* Series Six input address 1 to 8, bit 2 covers 9 to 16 etc. Note         *)
| (* that bits 14, 15 & 16 of register 8 covers addresses 1001 to 1024       *)
| (* which are not scanned by the S6PSV block as they have special           *)
| (* functions in the Series Six and not used for normal I/O.                *)
| (*                                                                         *)
| (* The 8 register locations pointed to by Y3 contain the Series Six        *)
| (* Address Map showing the entire address area being monitored by the      *)
| (* S6PSV block. As for the Card Present Map, each 90-70 register bit       *)
| (* represents each 8 bits of Series Six address being monitored.           *)
| (*************************************************************************)
|

Program: EXAMPL3          C:\LM90\FOLDERS\EXAMPL3          Block: _MAIN
```

```
04-22-02  15:02          GE FANUC SERIES 90-70 (v7.02)              Page     3
                                  EXAMPL3
         Example of using the S6PIO to only monitor the S6 I/O activity

| << RUNG 9 >>
|
|SLOT6OK                                                                PSVOK
|%M00106 +-----+                        +-------------+                %M01032
+--] [---+BLKMV+------------------------+CALL   S6PSV +-------------------( )--
|        | INT |                        | (EXTERNAL)  |
|        |     |                        |             |
|        |     | R&S_PSV         R&S_PSV |             | ERRSTAT
| CONST -+IN1 Q+-%R01301         %R01301-+X1         Y1+-%R01310
| +00006 |     |                        |             |
|        |     |                        |             |
|        |     |                 TIMEOUT |             | CPMAP01
| CONST -+IN2  |                 %R01302-+X2         Y2+-%R01311
| +00030 |     |                        |             |
|        |     |                        |             |
|        |     |                 POPMODE |             | ADMAP01
| CONST -+IN3  |                 %R01303-+X3         Y3+-%R01331
| +00000 |     |                        |             |
|        |     |                        |             |
|        |     |                        |             | IO_STAT
| CONST -+IN4  |                 %I00001-+X4         Y4+-%R01349
| +00000 |     |                        |             |
|        |     |                        |             |
|        |     |                        |             | PSVBLD#
| CONST -+IN5  |                 %Q00001-+X5         Y5+-%R01350
| +00000 |     |                        |             |
|        |     |                        |             |
|        |     |                 SYNCADD |             | PSVDB01
| CONST -+IN6  |                 %R01306-+X6         Y6+-%R01351
| +00124 |     |                         +-------------+
|        |     |
| CONST -+IN7  |
| +00000 +-----+
|
|[      END OF PROGRAM LOGIC      ]
|

Program: EXAMPL3            C:\LM90\FOLDERS\EXAMPL3            Block: _MAIN
```

Example 4: Two 90-70 CPU's In Redundancy Configuration

This program illustrates the use of two Series 90-70 CPU's in redundancy configuration driving a single Series Six I/O chain.
With additional coding, multiple I/O chains can be controlled by the use of the redundancy CPU configuration. In such a case, a S6PIO module should be used for each chain in each Series 90-70 CPU rack.
Bumpless change over between CPU's is achieved when either CPU rack is powered-down for maintenance or modifications, or in the event of a failure.
The example is intended to be used as a guide when constructing real world applications.
This example makes the following assumptions.

1 Two Series 90-70 CPU racks, with redundancy processors wired (as per Figure 2-6. Master Mode wiring with two 90-70's in Redundancy Configuration) to a Series Six primary I/O chain.
2 A single S6PIO module in slot 6 of each of the 90-70 CPU racks, are used to control the Series Six I/O Chain.

Also note that the example does not cater for the situation where both units are inactive, in which case both units will remain in passive mode. It is possible for the system to operate with both units active (both in Solo mode), in which case the first unit to enter master mode will prevent the other unit from entering master mode as well. If the units are operating in solo mode then there will be no data coherency between the units.

Refer to "Appendix A SWITCHING BETWEEN MODES" for more detail.

Program Listing

```
04-22-02  15:20           GE FANUC SERIES 90-70 (v7.02)              Page     1
                              EXAMPLE 4 (RDNDTEX)
          Example of using redundancy PLC configuration with S6PIO cards

|[     START OF PROGRAM LOGIC     ]
|
| (*************************************************************************)
| (* Sample test program for a redundancy configuration using an S6PIO       *)
| (* module in each 90-70 CPU rack to drive a single Series Six I/O chain.    *)
| (*                                                                         *)
| (* Both 90-70's should be running this same program.                       *)
| (*************************************************************************)
|
| (*************************************************************************)
| (* Determine the status of both Redundancy System PLC's.                  *)
| (*************************************************************************)
|
| << RUNG 7 >>
|
|PRI_UNT                                                            PRIMARY
|%S00033                                                            %M00133
+--] [-------------------------------------------------------------------( )--
|
| << RUNG 8 >>
|
|SEC_UNT                                                            SECNDRY
|%S00034                                                            %M00134
+--] [-------------------------------------------------------------------( )--
|
| << RUNG 9 >>
|
|LOC_RDY                                                            L_READY
|%S00035                                                            %M00135
+--] [-------------------------------------------------------------------( )--
|



Program: EXAMPL4            C:\LM90\FOLDERS\EXAMPL4             Block: _MAIN
```

```
04-22-02  15:20           GE FANUC SERIES 90-70 (v7.02)              Page     2
                              EXAMPLE 4 (RDNDTEX)
         Example of using redundancy PLC configuration with S6PIO cards

| << RUNG 10 >>
|
|LOC_ACT                                                                L_ACTIV
|%S00036                                                                %M00136
+--] [-------------------------------------------------------------------( )--
|
| << RUNG 11 >>
|
|REM_RDY                                                                R_READY
|%S00037                                                                %M00137
+--] [-------------------------------------------------------------------( )--
|
| << RUNG 12 >>
|
|REM_ACT                                                                R_ACTIV
|%S00038                                                                %M00138
+--] [-------------------------------------------------------------------( )--
|
| << RUNG 13 >>
|
|LOGIC=                                                                  EQUAL
|%S00039                                                                %M00139
+--] [-------------------------------------------------------------------( )--
|
| (*************************************************************************)
| (* Test to see if an S6PIO card exists in Slot 6 and is healthy.         *)
| (*************************************************************************)
|
| << RUNG 15 >>
|
|ALW_ON                                                                 SLOT6OK
|%S00007         +-------------+                                        %M00106
+--] [-----------+CALL  S6NOOP +----------------------------------------( )--
|                | (EXTERNAL)  |
|                |             |
|                |             | SL6STAT
|         CONST -+X1         Y1+-%P00006
|         0006   +-------------+
|
| (*************************************************************************)
| (* OPERATION MODE - SEQUENCE OF EVENTS:                                 *)
| (*                                                                       *)
| (* 1. Both units power-up into passive mode                             *)
| (* 2. Before a unit can move to master mode the following must be true:  *)
| (*       Local Unit is the active unit                                   *)
| (*       There is no I/O chain activity, PSVMODE is held while there is  *)
| (*       chain activity.                                                 *)
| (* 3. On the transition from passive mode to master mode the passive blocks *)
| (*    are stopped                                                        *)
| (*                                                                       *)
| (*    The test for chain activity ensures that if both units are in SOLO *)
| (*    (i.e. redundant communications lost) then there will still be      *)
| (*    only one master mode system                                        *)
| (*************************************************************************)

Program: EXAMPL4            C:\LM90\FOLDERS\EXAMPL4            Block: _MAIN
```

```
04-22-02  15:20          GE FANUC SERIES 90-70 (v7.02)              Page     3
                              EXAMPLE 4 (RDNDTEX)
         Example of using redundancy PLC configuration with S6PIO cards

| (*************************************************************************)
| (* Set passive mode                                                       *)
| (* Note that passive mode is held while there is chain activity.          *)
| (* The NOCHACT coil is only valid in passive mode.                        *)
| (*************************************************************************)
|
| << RUNG 18 >>
|
|L_ACTIV                                                              PSVMODE
|%M00136                                                              %M01028
+--]/[----------+-------------------------------------------------------( )--
|               |
|FST_SCN        |
|%S00001        |
+--] [----------+
|               |
|PSVMODE NOCHACT|
|%M01028 %M01029|
+--] [-----]/[--+
|
| (*************************************************************************)
| (* Set up parameters for normal passive mode                             *)
| (* Be careful not to scan shared I/O                                     *)
| (*                                                                       *)
| (* IN1 = X1 = rack 0 slot 6                                              *)
| (* IN2 = X2 = time out of 30 ms, this would have to be increased for a   *)
| (*            large program.                                             *)
| (*            As a general rule of thumb use 1.5 times the average       *)
| (*            sweep time                                                 *)
| (* IN3 = X3 = mode = single channel mode                                 *)
| (* IN4 =      not used                                                   *)
| (* IN5 =      not used                                                   *)
| (* IN6 = X6 = Sync Address = 0FFH = PDT window                          *)
| (* IN7 =      not used                                                   *)
| (*************************************************************************)
|

Program: EXAMPL4            C:\LM90\FOLDERS\EXAMPL4            Block: _MAIN
```

```
04-22-02  15:20           GE FANUC SERIES 90-70 (v7.02)              Page     4
                              EXAMPLE 4 (RDNDTEX)
         Example of using redundancy PLC configuration with S6PIO cards

| << RUNG 20 >>
|
|PSVMODE
|%M01028         +-----+
+--] [-----------+BLKMV+-
|                | INT |
|                |     |
|                |     | R&S_PSV
|          CONST -+IN1 Q+-%R01301
|          +00006 |     |
|                |     |
|          CONST -+IN2  |
|          +00030 |     |
|                |     |
|          CONST -+IN3  |
|          +00001 |     |
|                |     |
|          CONST -+IN4  |
|          +00000 |     |
|                |     |
|          CONST -+IN5  |
|          +00000 |     |
|                |     |
|          CONST -+IN6  |
|          +00255 |     |
|                |     |
|          CONST -+IN7  |
|          +00000 +-----+
|
| (*************************************************************************)
| (* Set up parameters to stop passive mode                                *)
| (*                                                                       *)
| (* Note that IN3 is now set to 0                                         *)
| (*************************************************************************)
|
```

```
Program: EXAMPL4            C:\LM90\FOLDERS\EXAMPL4            Block: _MAIN
```

```
04-22-02  15:20           GE FANUC SERIES 90-70 (v7.02)              Page     5
                               EXAMPLE 4 (RDNDTEX)
         Example of using redundancy PLC configuration with S6PIO cards

| << RUNG 22 >>
|
|PSVMODE
|%M01028         +-----+
+--]v[-----------+BLKMV+-
|                | INT |
|                |     |
|                |     | R&S_PSV
|          CONST -+IN1 Q+-%R01301
|          +00006 |     |
|                |     |
|          CONST -+IN2  |
|          +00030 |     |
|                |     |
|          CONST -+IN3  |
|          +00000 |     |
|                |     |
|          CONST -+IN4  |
|          +00000 |     |
|                |     |
|          CONST -+IN5  |
|          +00000 |     |
|                |     |
|          CONST -+IN6  |
|          +00255 |     |
|                |     |
|          CONST -+IN7  |
|          +00000 +-----+
|
| (*************************************************************************)
| (* Passive block, the scanned I/O should not overlap with any shared I/O    *)
| (*************************************************************************)
|

Program: EXAMPL4            C:\LM90\FOLDERS\EXAMPL4            Block: _MAIN
```

```
04-22-02  15:20           GE FANUC SERIES 90-70 (v7.02)              Page     6
                               EXAMPLE 4 (RDNDTEX)
         Example of using redundancy PLC configuration with S6PIO cards

| << RUNG 24 >>
|
|PSVMODE SLOT6OK                                                        PSCANOK
|%M01028 %M00106 +-------------+                                        %M01051
+--] [--+--] [---+CALL   S6PSV +----------------------------------------( )--
|       |        | (EXTERNAL)  |
|PSVMODE|        |             |
|%M01028|R&S_PSV |             | PSV_ERR
+--]v[--+%R01301-+X1         Y1+-%R01310
|                |             |
|        TIMEOUT |             | CPMAP01
|        %R01302-+X2         Y2+-%R01311
|                |             |
|        POPMODE |             | ADMAP01
|        %R01303-+X3         Y3+-%R01331
|                |             |
|                |             | PSVSTAT
|        %I01025-+X4         Y4+-%R01349
|                |             |
|                |             | PSVBBLD
|        %Q01025-+X5         Y5+-%R01350
|                |             |
|        SYNCADD |             | CHNL0&1
|        %R01306-+X6         Y6+-%R01352
|                +-------------+
|
| (*************************************************************************)
| (* Test for chain activity                                               *)
| (*                                                                       *)
| (* A chain is considered to be inactive if the pdt window                *)
| (* is not detected within the timeout period.  It is essential          *)
| (* that the timeout period therefore be set to an appropriate value.    *)
| (* (see previous comment)                                                *)
| (*                                                                       *)
| (* An error of 6E or 110 will be reported on the output of the          *)
| (* passive block when the timeout has been exceeded.                    *)
| (*                                                                       *)
| (* The NOCHACT coil is really only valid while in passive mode.         *)
| (*************************************************************************)
|
| << RUNG 26 >>
|
|PSVMODE
|%M01028         +-----+
+--] [-----------+ EQ_ +-
|                | INT |
|                |     |                                                  NOCHACT
|        PSV_ERR |     |                                                  %M01029
|        %R01310-+I1  Q+-------------------------------------------------( )--
|                |     |
|         CONST -+I2   |
|         +00110 +-----+
|

Program: EXAMPL4            C:\LM90\FOLDERS\EXAMPL4            Block: _MAIN
```

```
04-22-02  15:20          GE FANUC SERIES 90-70 (v7.02)              Page     7
                              EXAMPLE 4 (RDNDTEX)
          Example of using redundancy PLC configuration with S6PIO cards

| (*************************************************************************)
| (* Passive mode is not turned off until the local is active AND there is no *)
| (* I/O chain activity.  Hence the master mode S6SCAN block will not         *)
| (* be activated until these requirements are met.                          *)
| (*                                                                         *)
| (* Additionally, the S6SCAN block is executed after the S6PSV block        *)
| (* has been turned off.                                                    *)
| (*                                                                         *)
| (* For the S6SCAN block                                                    *)
| (* X1 = rack 0 slot 6                                                      *)
| (* X2 = Output table starting at %Q00001                                  *)
| (* X3 = Starting S6 address = channel 0 address 1                         *)
| (* X4 = End S6 address = channel 0 address 1000                           *)
| (* X5 = 0 RST de-asserted.  It is recommended that the state of RST follow  *)
| (*      the run/run-disable/stop CPU keyswitch.  This is not illustrated    *)
| (*      here                                                               *)
| (* X6 = Scan options = 11100000                                           *)
| (*                   = Force Idle mode                                     *)
| (*                   = Issue PDT window                                    *)
| (*                   = Only copy input data for which a valid CP was       *)
| (*                     recieved.                                           *)
| (* Force idle mode ensures the I/O chain is relinquished when the          *)
| (* firmware watchdog timer expires.  This watchdog will expire when       *)
| (* the 90-70 CPU stops communicating with the S6PIO.  It is               *)
| (* important that Force Idle Mode is selected for redundant applications.  *)
| (*************************************************************************)
|

Program: EXAMPL4            C:\LM90\FOLDERS\EXAMPL4            Block: _MAIN
```

```
04-22-02  15:20           GE FANUC SERIES 90-70 (v7.02)              Page     8
                              EXAMPLE 4 (RDNDTEX)
         Example of using redundancy PLC configuration with S6PIO cards

| << RUNG 28 >>
|
|SLOT6OK PSVMODE                                                          MSCANOK
|%M00106 %M01028 +-----+                +-------------+                  %M01061
+--] [-----]/[---+BLKMV+----------------+CALL  S6SCAN +------------------( )--
|                | INT |                | (EXTERNAL)  |
|                |     |                |             |
|                |     | R&SSCAN R&SSCAN |             | SCANERR
|         CONST -+IN1 Q+-%R01411 %R01411-+X1         Y1+-%R01421
|         +00006 |     |                |             |
|                |     |                |             |
|         CONST -+IN2  |         %Q00001-+X2         Y2+-%I00001
|         +00000 |     |                |             |
|                |     |                |             |
|                |     |         STAADDR |             | SCNSTAT
|         CONST -+IN3  |         %R01413-+X3         Y3+-%R01423
|         +00001 |     |                |             |
|                |     |                |             |
|                |     |         ENDADDR |             | SCNBBLD
|         CONST -+IN4  |         %R01414-+X4         Y4+-%R01424
|         +01000 |     |                |             |
|                |     |                |             |
|                |     |         SCNRSET |             |
|         CONST -+IN5  |         %R01415-+X5         Y5+-
|         +00000 |     |                |             |
|                |     |                |             |
|                |     |         SCNOPTN |             |
|         CONST -+IN6  |         %R01416-+X6         Y6+-
|         +00224 |     |                +-------------+
|                |     |
|         CONST -+IN7  |
|         +00000 +-----+
|
| (*************************************************************************)
| (* ADD USER LOGIC CODE HERE.                                             *)
| (*                                                                       *)
| (* The following is included for example purposes only                   *)
| (*                                                                       *)
| (*************************************************************************)
|
| << RUNG 30 >>
|
|        +-----+                                                    %M00001
+--------+ONDTR+-----------------------------------------------------------( )--
|        |0.01s|
|        |     |
|%M00001 |     |
+--] [---+R    |
|        |     |
| CONST -+PV CV+-
| +00010 |     |
|        +-----+
|
|        %P00026

Program: EXAMPL4            C:\LM90\FOLDERS\EXAMPL4             Block: _MAIN
```

```
04-22-02  15:20           GE FANUC SERIES 90-70 (v7.02)              Page     9
                              EXAMPLE 4 (RDNDTEX)
         Example of using redundancy PLC configuration with S6PIO cards

| << RUNG 31 >>
|
|%M00001 +-----+
+--]^[---+ BIT_+-
|        | SEQ |
|        |     |
|%Q00145 |     |
+--] [---+R    |
|        | LEN |
|        |00016|
|        |     |
+--------+DIR  |
|        |     |
| CONST -+STEP |
| +00001 |     |
|        |     |
|%Q00129-+ST   |
|        |     |
|        +-----+
|
|        %R00001
|
| << RUNG 32 >>
|
|FST_SCN
|%S00001                                                                %Q00161
+--] [-----------------------------------------------------------------------( )--
|
| << RUNG 33 >>
|
|%M00001 +-----+
+--]^[---+ BIT_+-
|        | SEQ |
|        |     |
|%Q00161 |     |
+--]^[---+R    |
|        | LEN |
|        |00016|
|        |     |
+--------+DIR  |
|        |     |
| CONST -+STEP |
| +00001 |     |
|        |     |
|%Q00145-+ST   |
|        |     |
|        +-----+
|
|        %R00004
|
|[      END OF PROGRAM LOGIC      ]
|


Program: EXAMPL4            C:\LM90\FOLDERS\EXAMPL4             Block: _MAIN
```

Example 5: Illustrate Use of S6PSV Block in Asynchronous Mode

Generally a S6PSV block will not return until it has received the synchronizing event (usually the PDT window). Hence if two passive blocks are required, as may be the case for a system with main and auxiliary I/O, there is no guarantee both will be able to synchronize to the Series Six I/O. Indeed it is possible one of the blocks will return after a time out with a time out error.

An alternative is to use the S6PSV blocks in asynchronous mode. In this case the block will return before the synchronizing event, but the request will have been queued. In this manner two S6PSV blocks can effectively queue their requests thus allowing both blocks to synchronize on the same event. A third S6PSV block would be used to block until the event is actually received.

The following program illustrates the point.

Program Listing

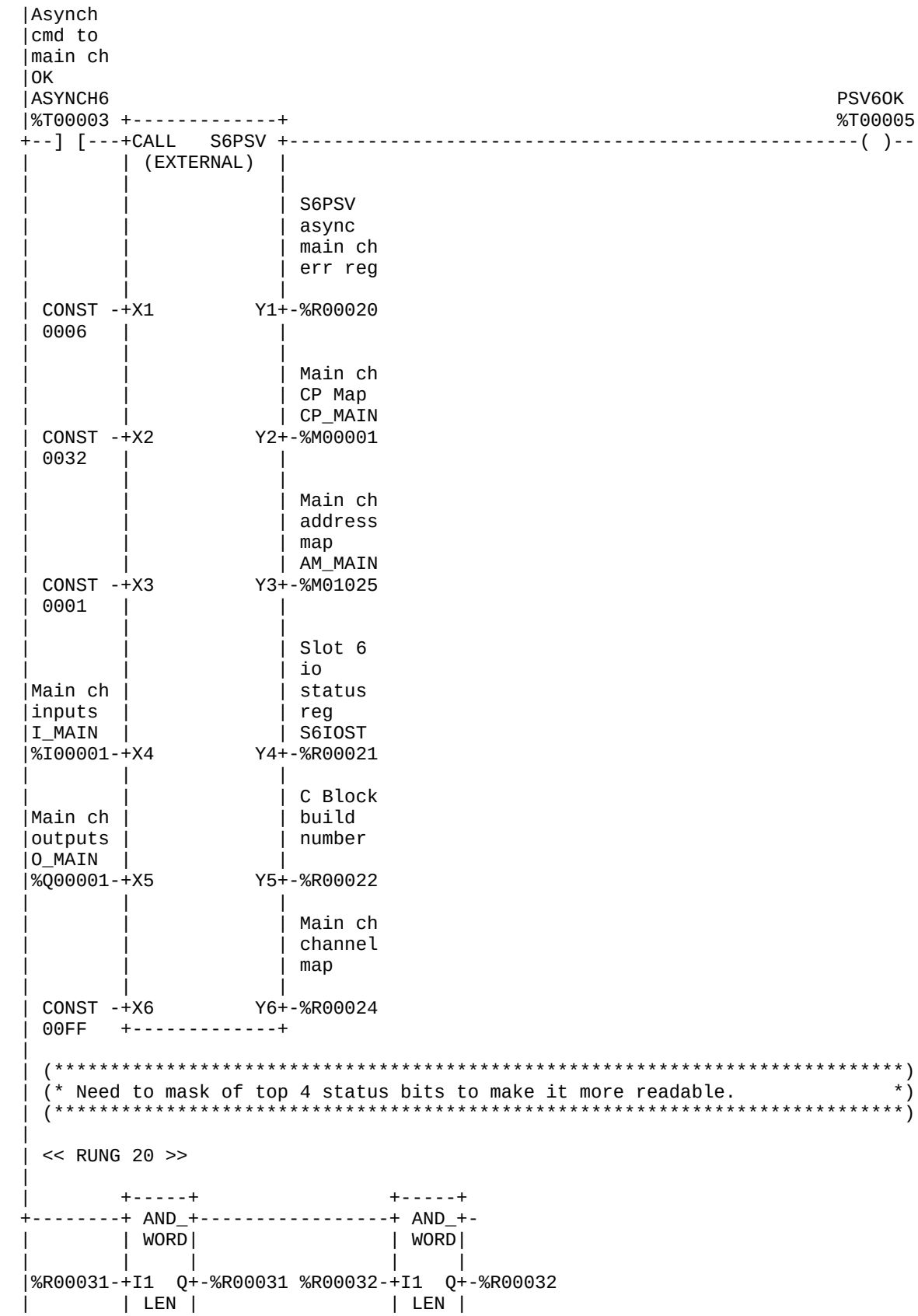
```
04-30-02  16:56          GE FANUC SERIES 90-70 (v7.02)              Page     1
                      Expanded Channel Passive Mode Test


|[     START OF PROGRAM LOGIC     ]
|
| (*************************************************************************)
| (* This program tests expanded channel passive mode and demonstrates the    *)
| (* asynchronous passive mode operation.                                      *)
| (*                                                                          *)
| (* In this example it is assumed that there are two S6PIO modules           *)
| (* installed in slots 6 & 8 of a 90-70 CPU rack.                            *)
| (*                                                                          *)
| (* The 90-70 PLC is cabled up to the primary & auxilliary chains of a       *)
| (* Series Six Plus PLC.  The chains are terminated on the S6PIO modules.    *)
| (* The module in slot 6 is connected to the primary chain, the module in    *)
| (* slot 8 is connected to the auxilliary chain.                             *)
| (*                                                                          *)
| (* The Series Six Plus PLC has all 8 channels enabled.                      *)
| (*************************************************************************)
|
| (*************************************************************************)
| (* Detect whether S6PIOs are installed in slots 6 & 8 and active.           *)
| (*************************************************************************)
|
| << RUNG 7 >>
|
|                                                                    SLOT6OK
|        +-------------+                                             %T00001
+--------+CALL  S6NOOP +-----------------------------------------------( )--
|        | (EXTERNAL)  |
|        |             |
|        |             | S6NOOP
|        |             | for
|        |             | main ch
|        |             | err rg
|        |             |
| CONST -+X1         Y1+-%R00010
| 0006   +-------------+
|
| << RUNG 8 >>
|
|                                                                    SLOT8OK
|        +-------------+                                             %T00002
+--------+CALL  S6NOOP +-----------------------------------------------( )--
|        | (EXTERNAL)  |
|        |             |
|        |             | S6NOOP
|        |             | for Aux
|        |             | ch err
|        |             | reg
|        |             |
| CONST -+X1         Y1+-%R00011
| 0008   +-------------+
|
| << RUNG 9 >>
|
```

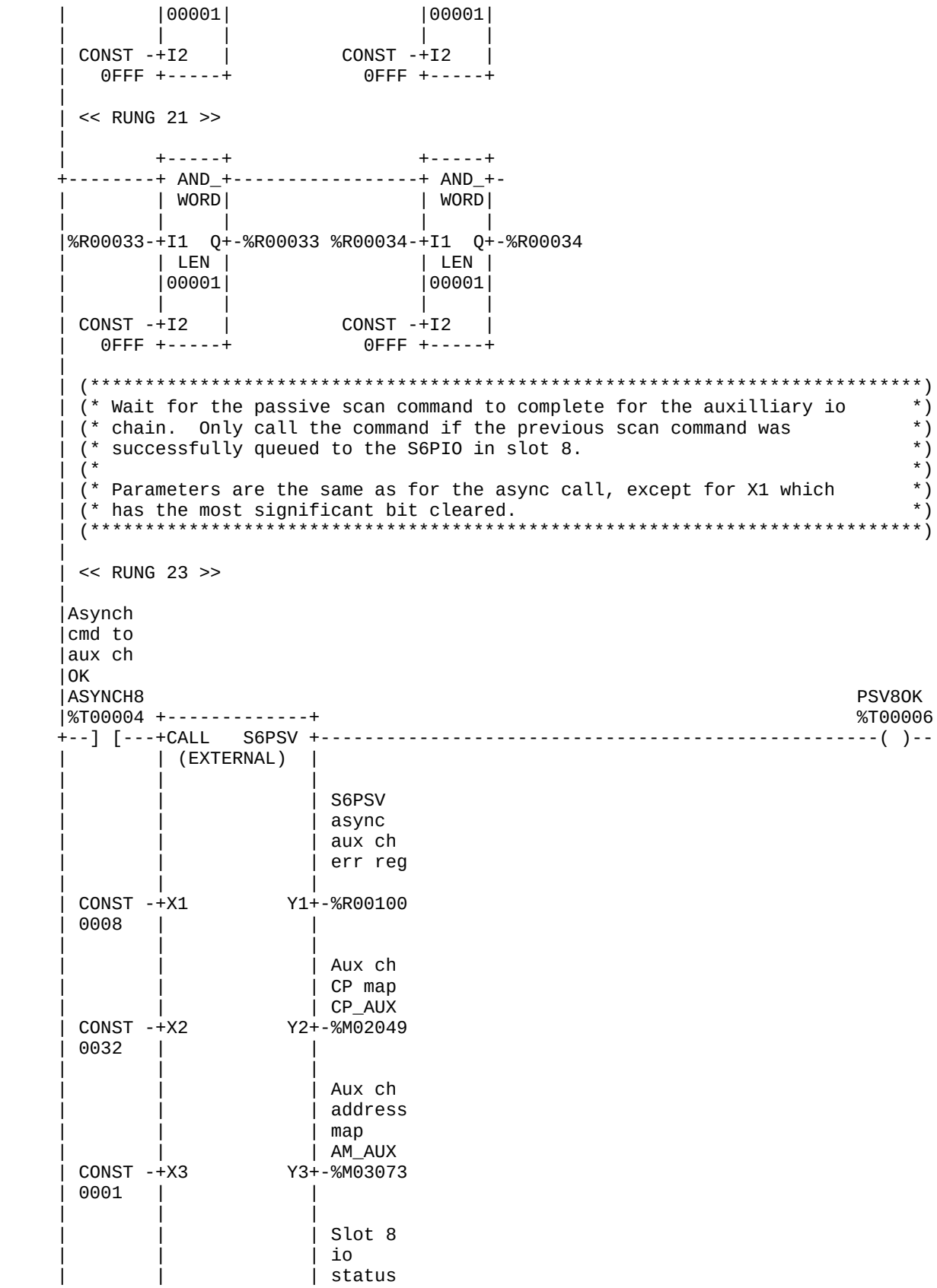
```
|        +-----+
+--------+DATA_+-
|        |INIT_|
|        | INT |
|        |     | S6PSV
|        |     | async
|        |     | aux ch
|        |     | err reg
|        |     |
|        |    Q+-%R00100
|        | LEN |
|        |00006|
|        |     |
|        +-----+
|
|DATA INIT INT at: %R00100
|   1       +00002        +00257        +00001        +00001        +00033
|   6       +00001
|
| << RUNG 10 >>
|
|FST_SCN
|%S00001 +-------------+                                                %M00002
+--] [---+CALL  S6PMAP +-----------------------------------------------( )--
|        | (EXTERNAL)  |
|        |             |
| CONST -+X1         Y1+-%R00001
| 0006   |             |
|        |             |
| CONST -+X2         Y2+-%R00002
| 0001   |             |
|        |             |
| CONST -+X3         Y3+-%R00003
| 0002   |             |
|        |             |
|S6PSV   |             |
|async   |             |
|aux ch  |             |
|err reg |             |
|        |             |
|%R00100-+X4         Y4+-%R00004
|        +-------------+
|
| (*************************************************************************)
| (* Issue asynch passive scan on the main io chain.  The C block          *)
| (* will return immediately without error if the passive scan             *)
| (* command was successfully queued to the S6PIO in slot 6.               *)
| (*                                                                       *)
| (* X1 - 0x8006 Use slot 6 and scan in asynch mode.  Ie just queue        *)
| (*              the command to the S6PIO in slot 6 and return.           *)
| (* X2 - 0x0064 Specifies a timeout of 100msec on sync address.           *)
| (* X3 - 0xFF02 Require data for all 8 expanded channels.  Run in         *)
| (*              expanded chanel mode.                                    *)
| (* X4 - %I0001 Use inputs %I00001 - %I08192 for input data              *)
| (* X5 - %Q0001 Use outputs %Q00001 - %Q08192 for output data            *)
| (* X6 - 0x00FF Synchronisation address, use the PDT window to           *)
| (*              synchronise.  It is always recommended to use the        *)
| (*              PDT window.                                              *)
| (* Y1 - %R0020 Error register.                                          *)
| (* Y2 - %M0001 Card Present (CP) map.  This is a bitmap indicating      *)
| (*              what input address bytes have responded with input       *)
| (*              data.                                                    *)
```

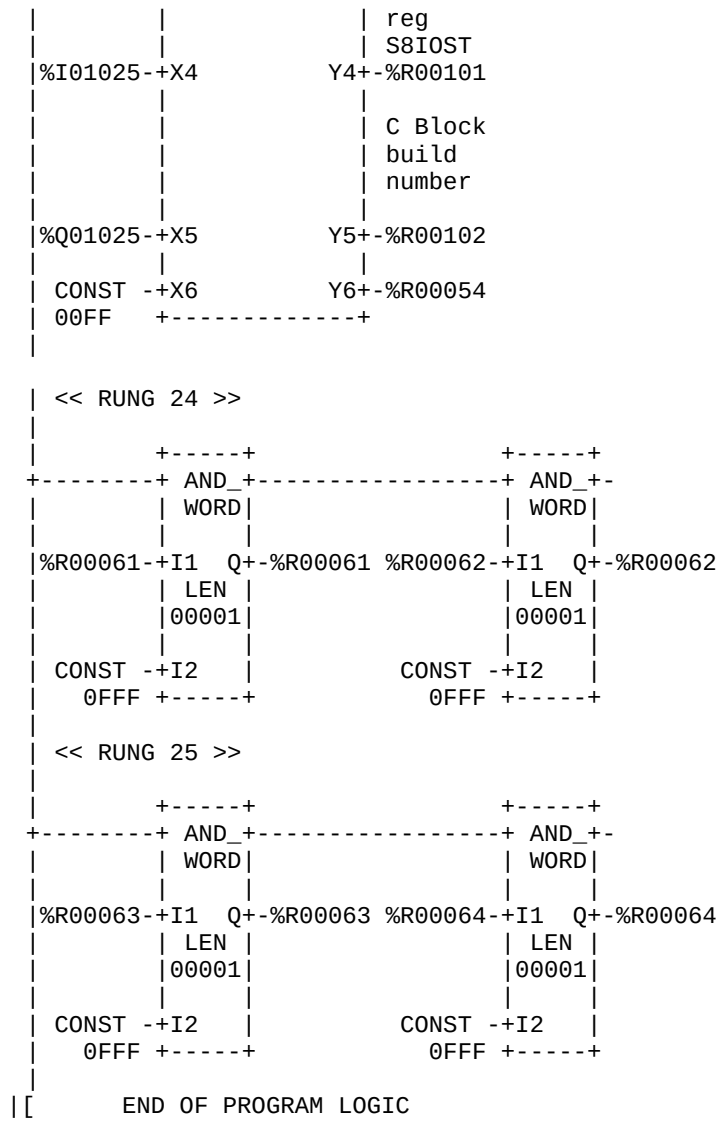
```
| (* Y3 - %M1025 Address (AM) map.  This is a bitmap indicating              *)
| (*             what address cycles have been sent out by the Series       *)
| (*             Six CPU.                                                   *)
| (* Y4 - %R0021 IO status register                                          *)
| (* Y5 - %R0022 2 register array indicatiing the C block and S6PIO          *)
| (*             build numbers.                                             *)
| (* Y6 - %R0024 Array of at least 3 registers.  First register is a         *)
| (*             channel bitmap indicating what channels have been          *)
| (*             scanned.  Second register is a counter which increments    *)
| (*             each time a synchronisation event occurs.  Third register  *)
| (*             is the size of the special passive mode data that may      *)
| (*             be returned.  For this example this will be zero.          *)
| (*                                                                        *)
| (*                                                                        *)
| (**************************************************************************)
|
| << RUNG 12 >>
|
|S6PIO
|in slot
|6 OK
|SLOT6OK                                                              ASYNCH6
|%T00001 +-------------+                                              %T00003
+--] [---+CALL   S6PSV +-----------------------------------------------( )--
|        | (EXTERNAL)  |
|        |             |
| CONST -+X1         Y1+-%R00015
| 8006   |             |
|        |             |
|        |             | Main ch
|        |             | CP Map
|        |             | CP_MAIN
| CONST -+X2         Y2+-%M00001
| 0032   |             |
|        |             |
|        |             | Main ch
|        |             | address
|        |             | map
|        |             | AM_MAIN
| CONST -+X3         Y3+-%M01025
| 0001   |             |
|        |             |
|Main ch |             |
|inputs  |             |
|I_MAIN  |             |
|%I00001-+X4         Y4+-%R00016
|        |             |
|Main ch |             |
|outputs |             |
|O_MAIN  |             |
|%Q00001-+X5         Y5+-%R00017
|        |             |
|        |             | Main ch
|        |             | channel
|        |             | map
|        |             |
| CONST -+X6         Y6+-%R00024
| 00FF   +-------------+
|
| << RUNG 13 >>
|
|FST_SCN
```

```
|%S00001 +-----+
+--] [---+DATA_+-
|        |INIT_|
|        | INT |
|        |     |
|        |    Q+-%R00300
|        | LEN |
|        |00006|
|        |     |
|        +-----+
|
|DATA INIT INT at: %R00300
|   1       +00002        +00289        +00001        +00001        +00065
|   6       +00001
|
| << RUNG 14 >>
|
|FST_SCN
|%S00001 +-------------+                                                %M00003
+--] [---+CALL  S6PMAP +-----------------------------------------------------( )--
|        | (EXTERNAL)  |
|        |             |
| CONST -+X1         Y1+-%R00350
| 0008   |             |
|        |             |
| CONST -+X2         Y2+-%R00351
| 0000   |             |
|        |             |
| CONST -+X3         Y3+-%R00352
| 0002   |             |
|        |             |
|%R00300-+X4         Y4+-%R00353
|        +-------------+
|
| (*************************************************************************)
| (* Issue asynch passive scan on the auxilliary io chain.  The C block       *)
| (* will return immediately without error if the passive scan command        *)
| (* was successfully queued to the S6PIO in slot 8.                          *)
| (*                                                                         *)
| (* X1 - 0x8008 Use slot 8 and scan in asynch mode.  Ie just queue           *)
| (*              the command to the S6PIO in slot 8 and return.             *)
| (* X2 - 0x0064 Specifies a tieout of 100msec on sync address.              *)
| (* X3 - 0x0F02 Require the first 4 expanded channels only.  Only           *)
| (*              data for these channls will be copied.  If any             *)
| (*              chanels are not scaned by the Series Six, the data         *)
| (*              areas are set to zero.                                     *)
| (* X4 - %I8193 Use inputs %I8193 - %I12288 for input data.                 *)
| (*              WARNING: Beware of writing beyond end of configured        *)
| (*              table memory limits.                                       *)
| (* X5 - %Q8193 Use outputs %Q8193 - %Q12288 for output data.              *)
| (*              WARNING: Beware of writin beyond end of configured         *)
| (*              table memory limits.                                       *)
| (* X6 - 0x00FF Synchronisation address, use th PDT window to              *)
| (*              synchronise.  It is always recommended to use the          *)
| (*              PDT window.                                                *)
| (* Y1 - %R100  Error register.                                            *)
| (* Y2 - %M2049 Card Present (CP) map.  This is a bitmap indicating        *)
| (*              what input address bytes have responded with input         *)
| (*              data.                                                      *)
| (* Y3 - %M3073 Address (AM) map.  This is a bitmap indicating             *)
| (*              what address cycles have been sent out by the Series       *)
| (*              Six CPU.                                                   *)
```

```
| (* Y4 - %R101  IO status register                                          *)
| (* Y5 - %R102  2 register array indicating the C block and S6PIO           *)
| (*             build numbers.                                             *)
| (* Y6 - %R104  Array of at least 3 registers.  First register is a         *)
| (*             channel bitmap indicating what channels have been           *)
| (*             scanned.  Second registr is a counter which increments      *)
| (*             each time a synchronisation event occurs.  Third register   *)
| (*             is the size of the special passive mode data that may       *)
| (*             be returned.  For this example this will be zero.           *)
| (*************************************************************************)
|
| << RUNG 16 >>
|
|S6PIO
|in slot
|8 OK
|SLOT8OK                                                              ASYNCH8
|%T00002 +-------------+                                              %T00004
+--] [---+CALL   S6PSV +-----------------------------------------------( )--
|        | (EXTERNAL)  |
|        |             |
| CONST -+X1         Y1+-%R00050
| 8008   |             |
|        |             |
|        |             | Aux ch
|        |             | CP map
|        |             | CP_AUX
| CONST -+X2         Y2+-%M02049
| 0032   |             |
|        |             |
|        |             | Aux ch
|        |             | address
|        |             | map
|        |             | AM_AUX
| CONST -+X3         Y3+-%M03073
| 0001   |             |
|        |             |
|%I01025-+X4         Y4+-%R00051
|        |             |
|%Q01025-+X5         Y5+-%R00052
|        |             |
| CONST -+X6         Y6+-%R00054
| 00FF   +-------------+
|
| (*************************************************************************)
| (* Wait for the passive scan command currently pending on the             *)
| (* main io chain to complete.  Only issue command if the previous         *)
| (* asynch command was successfully queued.                                *)
| (*                                                                       *)
| (* Parameters are the same as for the async call, except for X1 which     *)
| (* has the most significant bit cleared.                                  *)
| (*************************************************************************)
|
| << RUNG 18 >>
|
|
|
|
|
|
|
|
```


Asynch
cmd to
main ch
OK
ASYNCH6
%T00003
CALL
(EXTERNAL)
S6PSV
PSV6OK
%T00005
S6PSV
async
main ch
err reg
CONST
0006
X1
Y1
%R00020
Main ch
CP Map
CP_MAIN
CONST
0032
X2
Y2
%M00001
Main ch
address
map
AM_MAIN
CONST
0001
X3
Y3
%M01025
Slot 6
io
status
reg
S6IOST
Main ch
inputs
I_MAIN
%I00001
X4
Y4
%R00021
C Block
build
number
Main ch
outputs
O_MAIN
%Q00001
X5
Y5
%R00022
Main ch
channel
map
CONST
00FF
X6
Y6
%R00024
(* Need to mask of top 4 status bits to make it more readable. *)
<< RUNG 20 >>
AND_
WORD
LEN
%R00031
%R00032


|00001|
CONST -+I2
0FFF +-----+
|00001|
CONST -+I2
0FFF +-----+
<< RUNG 21 >>
AND_+
WORD|
%R00033-+I1 Q+-%R00033
LEN |
|00001|
CONST -+I2
0FFF +-----+
%R00034-+I1 Q+-%R00034
(* Wait for the passive scan command to complete for the auxilliary io *)
(* chain. Only call the command if the previous scan command was *)
(* successfully queued to the S6PIO in slot 8. *)
(* Parameters are the same as for the async call, except for X1 which *)
(* has the most significant bit cleared. *)
<< RUNG 23 >>
Asynch
cmd to
aux ch
OK
ASYNCH8
%T00004
CALL S6PSV
(EXTERNAL)
PSV8OK
%T00006
S6PSV
async
aux ch
err reg
CONST -+X1
0008
Y1+-%R00100
Aux ch
CP map
CP_AUX
CONST -+X2
0032
Y2+-%M02049
Aux ch
address
map
AM_AUX
CONST -+X3
0001
Y3+-%M03073
Slot 8
io
status


reg
S8IOST
%I01025-+X4 Y4+-%R00101
C Block
build
number
%Q01025-+X5 Y5+-%R00102
CONST -+X6 Y6+-%R00054
00FF +-------------+
<< RUNG 24 >>
AND_+ AND_+-
WORD WORD
%R00061-+I1 Q+-%R00061 %R00062-+I1 Q+-%R00062
LEN LEN
00001 00001
CONST -+I2 CONST -+I2
0FFF +-----+ 0FFF +-----+
<< RUNG 25 >>
AND_+ AND_+-
WORD WORD
%R00063-+I1 Q+-%R00063 %R00064-+I1 Q+-%R00064
LEN LEN
00001 00001
CONST -+I2 CONST -+I2
0FFF +-----+ 0FFF +-----+
END OF PROGRAM LOGIC

A

B

C

D

E

F

H

I

T

U

V

W