SEL-3355/SEL-3360 Main Board Controller Software Developer's Guide

Introduction

This document describes the interface for communicating with and accessing registers within the main board controller on either an SEL-3355 Rack-Mount Rugged Computer or SEL-3360 Compact Industrial Computer. Use this document to develop device driver or application software necessary for configuring the main board.

Interface Definition

The interface to the main board controller consists of two 8-bit registers exposed to the Low Pin Count (LPC) I/O bus interface.

Register Definition

Data

Table 1 Data Register Definition

Data (Address 0x0192)								
Bit	7	6	5	4	3	2	1	0
Read	OBR							
Write	IBR							

OBR: The Output Buffer Register (OBR) is a holding buffer the main board controller uses to present data to the LPC interface. This is a single-register deep buffer, so successive reads always return identical data. The main board controller sets the Output Buffer Flag (OBF) to indicate new data are inserted into the OBR (see *Command/Status* for register information).

IBR: The Input Buffer Register (IBR) is a holding buffer on the LPC interface used to present data to the main board controller. This is a single-register deep buffer, so successive writes overwrite previous data. The main board controller can signal it is ready to receive new data by clearing the Input Buffer Flag (IBF) (see *Command/Status* for register information).

Command/Status

Table 2 Command/Status Register Definition

Command/Status (Address 0x0193)								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read	ST		–	BUS	C_D	–	IBF	OBF
Write	CMD							

ST: These bits indicate the present operating condition of the interface state machine. *Table 3* lists valid states.

Table 3 Interface States

ST	State
00	IDLE
01	WRITE
10	READ
11	ERROR

BUS: This bit indicates that the main board controller is busy processing. It should be used in conjunction with the IBF to determine when you can use the LPC interface to send another command.

C_D: This bit indicates the last register to which you have written. When this bit is clear, the last write was to the Data register; when this bit is set, the last write was to the Command/Status register.

IBF: The main board controller sets this flag automatically when you write to either the IBR or CMD register to indicate that it has not processed the data. The main board controller clears this bit when it has finished processing the last byte written. You should check that this bit has cleared before writing to either the IBR or CMD register.

OBF: The main board controller sets this flag when it updates the OBR with new data. The controller clears the OBF automatically when the LPC interface reads the OBR. The main board controller will not write new data to the OBR until this bit clears.

CMD: The command (CMD) register is an 8-bit register you use to issue a command to the main board controller. You should check that the IBF and BUS bits have cleared before writing to this register.

Commands

All following command flows make use of a WAIT process, defined by *Figure 1* and following text.

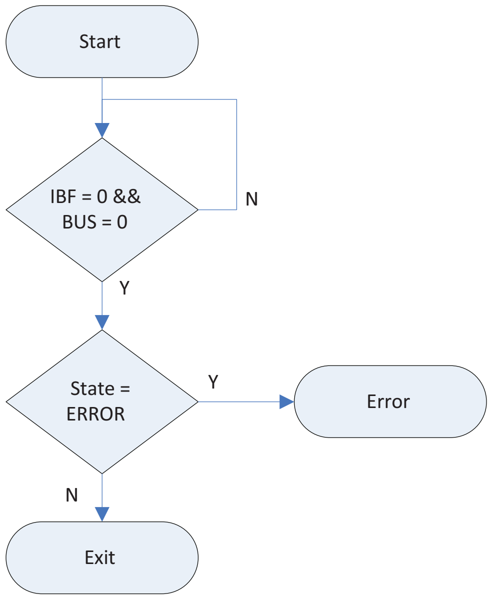

Figure 1 Wait Process

WRITE_CONFIG (0x10), WRITE_DIAG (0x11), WRITE_IRIG (0x12), WRITE_PS_DIAG (0x13)

A WRITE command initiates the writing of a byte of data to the main board controller register interface. After you write the WRITE command to the CMD register, you write the register address to the IBR and then subsequently send the data to write. Perform appropriate IBF and BUS bit checks to prevent race conditions. *Figure 2* describes the write command procedure.

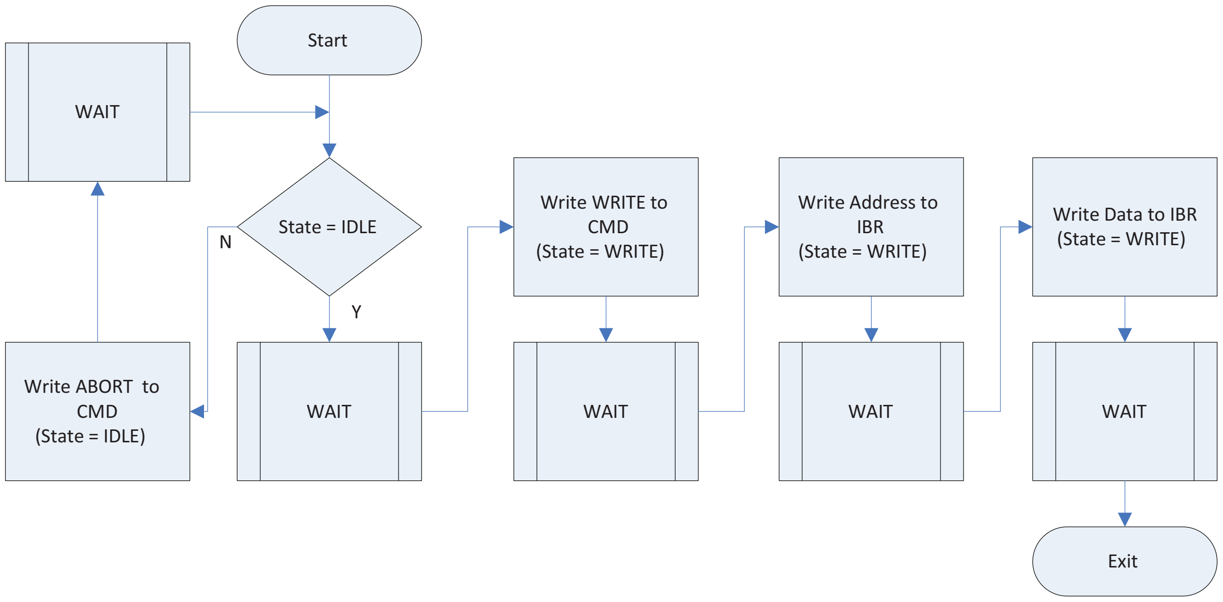

Figure 2 Write Process

READ_CONFIG (0x20), READ_DIAG (0x21), READ_IRIG (0x22), READ_PS_DIAG (0x23)

A READ command initiates the reading of a byte of data from the main board controller register interface. After you write the READ command to the CMD register, you write the register address to the IBR. After the main board controller receives the address, it writes the requested data to the OBR. Perform appropriate IBF and BUS bit checks to prevent race conditions. *Figure 3* describes the READ command procedure.

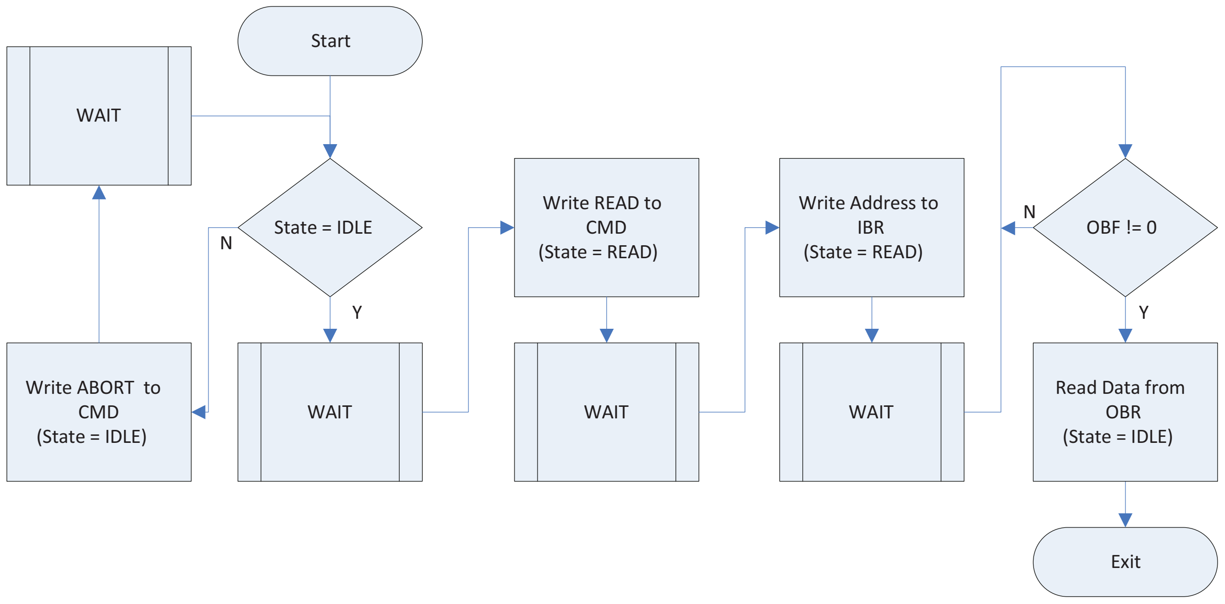
Start
WAIT
State = IDLE
N
Y
Write ABORT to CMD (State = IDLE)
WAIT
Write READ to CMD (State = READ)
WAIT
Write Address to IBR (State = READ)
WAIT
N
OBF != 0
Y
Read Data from OBR (State = IDLE)
Exit

Figure 3 Read Process

READCONT_CONFIG (0x40), READCONT_DIAG (0x41), READCONT_IRIG (0x42), READCONT_PS_DIAG (0x43)

A READCONT command initiates the continuous reading of successive bytes of data from the main board controller register interface. After you write the READCONT command to the CMD register, you write the first register address to the IBR. After the main board controller receives the address, it writes the first requested data byte to the OBR. To receive the next byte, you must write another READCONT command to the CMD register. After receiving the READCONT command again, the main board controller places the next byte in the OBR. To stop the continuous read, you must write a READCONTEND command to the CMD register. Perform appropriate IBF and BUS bit checks to prevent race conditions. *Figure 4* describes the READCONT command procedure.

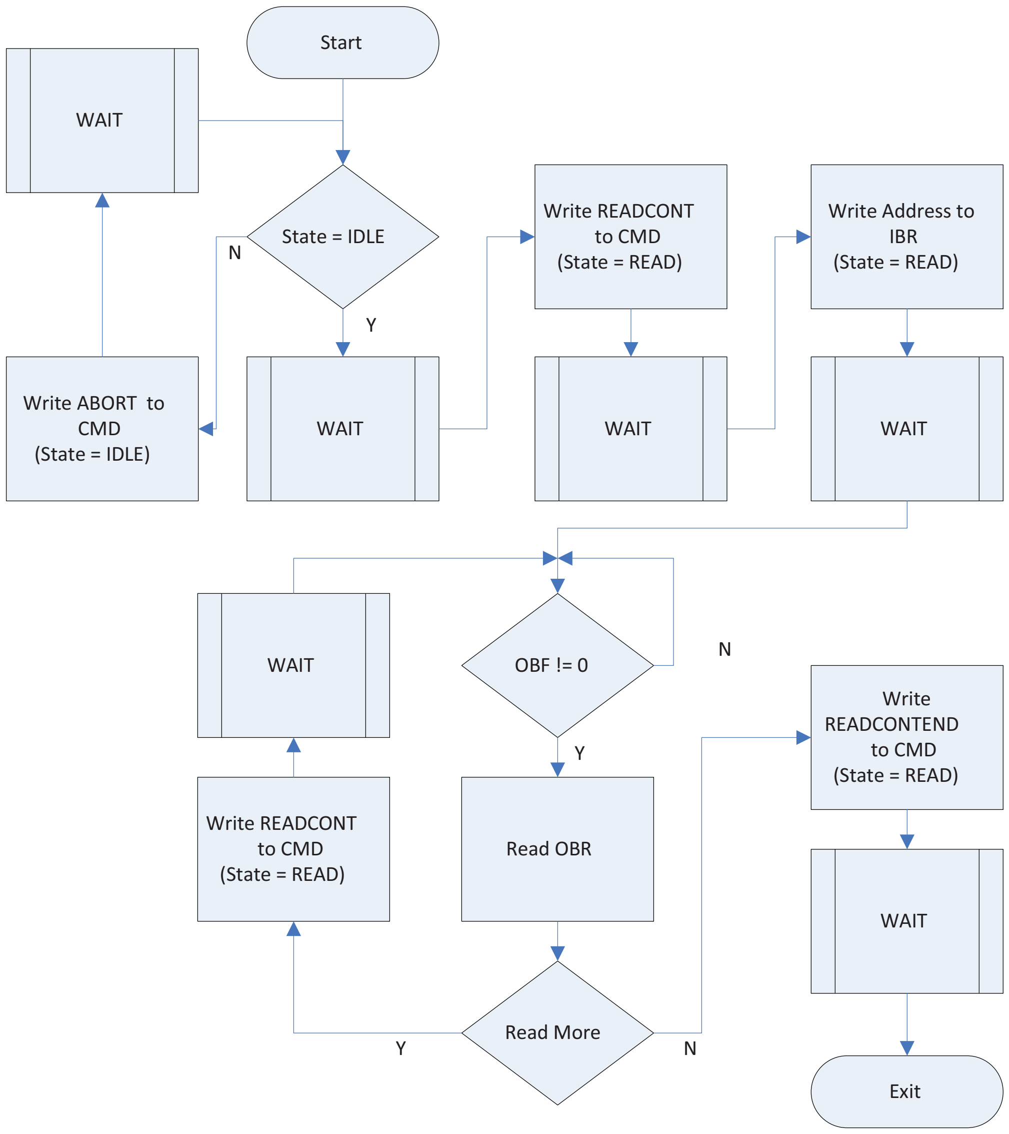
Start
WAIT
State = IDLE
N
Y
Write ABORT to CMD (State = IDLE)
WAIT
Write READCONT to CMD (State = READ)
WAIT
Write Address to IBR (State = READ)
WAIT
WAIT
OBF != 0
N
Y
Write READCONT to CMD (State = READ)
Read OBR
Write READCONTEND to CMD (State = READ)
WAIT
Read More
Y
N
Exit

Figure 4 Read Continuous Process

READCONTEND (0x60)

You write a READCONTEND command to the CMD register to end a continuous read from the main board controller. See the READCONT command for a state machine describing the operation of the READCONTEND command.

ABORT (0x00)

You write an ABORT command to the CMD register at any time to reset the main board controller to the IDLE state, which aborts any transaction in progress. You must send the ABORT command to the main board controller to acknowledge and clear an ERROR state.

Register Map Definition

Note: Treat all addresses not listed in the following tables as reserved and do not read from or write to such addresses.

Note: Write all reserved and read-only bits in registers as ‘0’ unless otherwise noted. All reserved and write-only bits shall return undefined values if read.

CONFIG Register Map (WRITE_CONFIG, READ_CONFIG, READCONT_CONFIG)

Register	Reset Value	Address	Description
BOARDID0	–[a]	0x00	Bits 7–0 (Read-Only): Board ID bits 7–0
BOARDID1	–[a]	0x01	Bits 7–0 (Read-Only): Board ID bits 15–8
BOARDID2	–[a]	0x02	Bits 7–0 (Read-Only): Board ID bits 23–16
BOARDID3	–[a]	0x03	Bits 7–0 (Read-Only): Board ID bits 31–24
MCVERSION0	–[a]	0x0C	Bits 7–0 (Read-Only): Main Board Controller Hardware Version bits 7–0
MCVERSION1	–[a]	0x0D	Bits 7–0 (Read-Only): Main Board Controller Hardware Version bits 15–8
MCVERSION2	–[a]	0x0E	Bits 7–0 (Read-Only): Main Board Controller Hardware Version bits 23–16
MCVERSION3	–[a]	0x0F	Bits 7–0 (Read-Only): Main Board Controller Hardware Version bits 31–24
LED_CONTROL0	0x01	0x10	Bits 7–2: Reserved Bit 1 (Read/Write): Enabled LED—Set to ‘1’ to turn the Enabled LED green. Set to ‘0’ to turn off the Enabled LED. Bit 0 (Read/Write): Alarm LED—Set to ‘1’ to turn the Alarm LED red and energize the contact. Set to ‘0’ to turn off the Alarm LED and de-energize the contact.
LED_CONTROL1	0x00	0x11	Bits 7–6: Reserved Bits 5–4 (Read/Write): Aux LED2—00b to turn off the LED, 01b to turn the LED red, 10b to turn the LED green, 11b is undefined. Bits 3–2 (Read/Write): Aux LED1—00b to turn off the LED, 01b to turn the LED red, 10b to turn the LED green, 11b is undefined. Bits 1–0 (Read/Write): Aux LED0—00b to turn off the LED, 01b to turn the LED red, 10b to turn the LED green, 11b is undefined.
JUMPER0	0x00	0x20	Bit 7 (Read-Only): State of Jumper H Bit 6 (Read-Only): State of Jumper G Bit 5 (Read-Only): State of Jumper F Bit 4 (Read-Only): State of Jumper E Bit 3 (Read-Only): State of Jumper D Bit 2 (Read-Only): State of Jumper C Bit 1 (Read-Only): State of Jumper B Bit 0 (Read-Only): State of Jumper A **Note:** See the device instruction manual for details on jumper assignment.
WATCHDOG	0x00	0x30	Bits 7–0 (Read/Write): A write of 0x00 disables the watchdog. A write of anything else indicates the number of twos-of-seconds until the next watchdog expiration. A read will return the amount of time left until expiration in twos-of-seconds.

Register	Reset Value	Address	Description
SERIAL_CONTROL	0x00	0x40	Bits 7–4: Reserved Bit 3 (Read/Write): UART1DIS—Writing a ‘1’ to this bit disables UART1. Writing a ‘0’ to this bit enables UART1. Bit 2 (Read/Write): UART0DIS—Writing a ‘1’ to this bit disables UART0. Writing a ‘0’ to this bit enables UART0. Bit 1 (Read/Write): UART1PWR—Writing a ‘1’ to this bit turns on +5 V for UART1. Writing a ‘0’ to this bit turns off +5 V for UART1. Bit 0 (Read/Write): UART0PWR—Writing a ‘1’ to this bit turns on +5 V for UART0. Writing a ‘0’ to this bit turns off +5 V for UART0.
MISC_CONTROL	0x10	0x42	Bits 7–6: Reserved Bit 5 (Read/Write): BIOS_CLEAR—Clear the BIOS settings at next restart. Bits 4–3 (Read/Write): PWRSTATE **Bits** / **Configuration** 00 — INITIALLYOFF—Leave the processor off after the main board controller boots and allow it to shut down or go to sleep. 01 — ALWAYSON—Always turn the processor on when it shuts down or goes to sleep. 10 — FIRSTBOOT—Turn on the processor after the main board controller boots and then allow it to shut off and sleep. 11 — Reserved One of the buttons (see MISC_CONTROL1) must be configured to either Power or Lamp Test for these bits to be writable. If they are not, the setting is read-only and is forced to ALWAYSON. Bit 2: Reserved Bit 1: Reserved Bit 0: Reserved
MISC_CONTROL1	0x12	0x43	Bits 7–4 (Read/Write): LAMPTESTBTNCFG—These bits define the behavior of the Lamp Test button. The bits may be set as shown in the following table. Bits 3–0 (Read/Write): PINHOLEBTNCFG—These bits define the behavior of the Pinhole button. The bits may be set as shown in the following table. **Bits** / **Configuration** 0000 — Disabled 0001 — Lamp Test 0010 — Power 0011 — Reset Others — Reserved
SERIALNO	0x00	0x50	Bits 7–0 (Read-Only): Reading from this register returns the device serial number one byte at a time. The serial number is in ASCII text. The end of the string is indicated by a read of a 0x00 byte. After you read the 0x00 byte, the main board controller resets the register to the beginning of the string and each successive read returns the same string.
MODEL	0x00	0x51	Bits 7–0 (Read-Only): Reading from this register returns the device MODEL one byte at a time. The MODEL is in ASCII text. The end of the string is indicated by a read of a 0x00 byte. After you read the 0x00 byte, the main board controller resets the register to the beginning of the string and each successive read returns the same string.

Register	Reset Value	Address	Description
CONFIGURATIONID	0x00	0x52	Bits 7–0 (Read-Only): Reading from this register returns the device Configuration ID one byte at a time. The Configuration ID is in ASCII text. The end of the string is indicated by a read of a 0x00 byte. After you read the 0x00 byte, the main board controller resets the register to the beginning of the string and each successive read returns the same string.
PCIBOARDID0		0x70	Bits 7–0 (Read-Only): Lower byte of the SEL PCI Expansion board ID. If the expansion board is not present, the 16-bit word formed from PCIBOARDID1 and PCIBOARDID0 will be 0xffff.
PCIBOARDID1		0x71	Bits 7–0 (Read-Only): Upper byte of the SEL PCI Expansion board ID. If the expansion board is not present, the 16-bit word formed from PCIBOARDID1 and PCIBOARDID0 will be 0xffff.

a. The present firmware version in use by the board or Flash determines the reset value for these registers.

DIAG Register Map (WRITE_DIAG, READ_DIAG, READCONT_DIAG)

Note: All Diagnostics are 32-bit values presented in little-endian format with the first byte starting at the address indicated. Reading the first byte of a value latches the remaining three bytes, allowing for the reading of a consistent 32-bit value.

Register	Reset Value	Address	Description
CPUCORETEMPx	0x00000000	0x00	CPU core temperature (in millidegrees kelvin)
PSTEMPx	0x00000000	0x04	Power supply temperature (in millidegrees kelvin)
MBTEMPx	0x00000000	0x0C	Main board ambient temperature (in millidegrees kelvin)
DIMM0TEMPx	0x00000000	0x10	DIMM0 temperature (in millidegrees kelvin)
DIMM1TEMPx	0x00000000	0x14	DIMM1 temperature (in millidegrees kelvin)
PCHTEMPx	0x00000000	0x18	PCH temperature (in millidegrees kelvin)
V12x	0x00000000	0x34	12 V power rail (in mV)
VMB_BAT	0x00000000	0x6C	3.3 V main board battery voltage (in mV)
POHOURSx	0x00000000	0xD0	Power on hours
SYSCYCLEx	0x00000000	0xD4	System power cycles
MAXTEMPx	0x00000000	0xD8	Maximum temperature reached (in millidegrees kelvin)
MINTEMPx	0x00000000	0xDC	Minimum temperature reached (in millidegrees kelvin)
WDEXPx	0x00000000	0xE0	A register set to a non-zero value if a watchdog expiration occurred since the last sleep transition or system reset
OVERCURx	0x00000000	0xE4	Overcurrent status: Bitmask of overcurrent bits. VIDEO: 0x1 SERIAL: 0x2 USB FRONT: 0x4 USB REAR: 0x8 USB INTERNAL: 0x10

IRIG Register Map (WRITE_IRIG, READ_IRIG, READCONT_IRIG)

Note: All IRIG values are 32-bit values presented in little-endian format with the first byte starting at the address indicated.

Note: For definitions of IRIG bits, refer to the IRIG standard. Obtain this standard at www.irigb.com.

Note: Position identifiers do not count as bits.

Register	Reset Value	Address	Description
DECODE_SMHx	0x00000000	0x00	A read of this register latches all IRIG registers to provide a consistent time message. Bit 31: IRIG Good. ‘1’ = IRIG decoded correctly. ‘0’ = no valid IRIG signal. Bits 30–26: Reserved. Bits 25–17: Correspond to bits P28–P20 of the IRIG stream. Bits 16–8: Correspond to bits P18–P10 of the IRIG stream. Bits 7–0: Correspond to bits P8–P1 of the IRIG stream.
DECODE_DAYSx	0x00000000	0x04	Bit 31: IRIG Good. ‘1’ = IRIG decoded correctly. ‘0’ = no valid IRIG signal. Bits 30–24: 7-bit counter for one hundredth of a second. 0x00 = top of second, 0x63 = bottom of second (99 hundredths). Bits 23–18: Reserved. Bits 17–9: Correspond to bits P48–P40 of the IRIG stream. Bits 8–0: Correspond to bits P38–P30 of the IRIG stream.
DECODE_CONTROLx	0x00000000	0x08	Bit 31: IRIG Good. ‘1’ = IRIG decoded correctly. ‘0’ = no valid IRIG signal. Bits 30–27: Reserved. Bits 26–18: Correspond to bits P78–P70 of the IRIG stream. Bits 17–9: Correspond to bits P68–P60 of the IRIG stream. Bits 8–0: Correspond to bits P58–P50 of the IRIG stream.
DECODE_SBSx	0x00000000	0x0C	Bit 31: IRIG Good. ‘1’ = IRIG decoded correctly. ‘0’ = no valid IRIG signal. Bits 30–18: Reserved. Bits 17–9: Correspond to bits P98–P90 of the IRIG stream. Bits 8–0: Correspond to bits P88–P80 of the IRIG stream.

PS_DIAG Register Map (WRITE_PS_DIAG, READ_PS_DIAG, READCONT_PS_DIAG)

Register	Reset Value	Address	Description
PS1_STATUS	0x00	0x00	Bits 7–2: Reserved Bit 1 (Read-Only): PSGOOD—Power Supply Good Bit 0 (Read-Only): PSPRES—Power Supply Present
PS1_PRODUCTID	0x00	0x34	Bits 7–0 (Read-Only): Reading from this register returns the product string of the first power supply one byte at a time. The string is in ASCII text. A 0x00 byte indicates the end of the string or that the power supply is no longer present (!PSPRES). After you read the 0x00 byte, the main board controller resets the register to the beginning of the string and each successive read returns the same string.
PS1_SERIALNO	0x00	0x35	Bits 7–0 (Read-Only): Reading from this register returns the serial number of the first power supply one byte at a time. The string is in ASCII text. A 0x00 byte indicates the end of the string or that the power supply is no longer present (!PSPRES). After you read the 0x00 byte, the main board controller resets the register to the beginning of the string and each successive read returns the same string.
PS2_STATUS	0x00	0x80	Bits 7–2: Reserved Bit 1 (Read-Only): PSGOOD—Power supply good Bit 0 (Read-Only): PSPRES—Power supply present

Register	Reset Value	Address	Description
PS2_PRODUCTID	0x00	0xB4	Bits 7–0 (Read-Only): Reading from this register returns the product string of the second power supply one byte at a time. The string is in ASCII text. A 0x00 byte indicates the end of the string or that the power supply is no longer present (!PSPRES). After you read the 0x00 byte, the main board controller resets the register to the beginning of the string and each successive read returns the same string.
PS2_SERIALNO	0x00	0xB5	Bits 7–0 (Read-Only): Reading from this register returns the serial number of the second power supply one byte at a time. The string is in ASCII text. A 0x00 byte indicates the end of the string or that the power supply is no longer present (!PSPRES). After you read the 0x00 byte, the main board controller resets the register to the beginning of the string and each successive read returns the same string.

Factory Assistance

We appreciate your interest in SEL products and services. If you have questions or comments, please contact us at:

Schweitzer Engineering Laboratories, Inc.
2350 NE Hopkins Court
Pullman, WA 99163-5603 USA
Telephone: +1.509.332.1890
Fax: +1.509.332.7990
Internet: selinc.com
Email: info@selinc.com

⚠WARNING
Operator safety may be impaired if the device is used in a manner not specified by SEL.

⚠AVERTISSEMENT
La sécurité de l'opérateur peut être compromise si l'appareil est utilisé d'une façon non indiquée par SEL.

SCHWEITZER ENGINEERING LABORATORIES, INC.
2350 NE Hopkins Court • Pullman, WA 99163-5603 U.S.A.
Tel: +1.509.332.1890 • Fax: +1.509.332.7990
selinc.com • info@selinc.com