RT1

Resistance Temperature Detector

Function Module

MODULE MANUAL

Revision History

Module Manual - RT1 Revision History		
Revision	**Revision Date**	**Description**
C	2022-09-15	ECO C09637, transition to docbuilder format. Replaced "Specifications" with "Data Sheet". Pg.6, added Pt2000 to RTD interface & excitation specs. Pg.6, changed 4-wire accuracy to 0.1%. Pg.6, changed Update Rate to Sample Rate; updated spec. Pg.6, changed Power to 450 mA. Pg.7, removed (default) from 'four-wire mode' in Introduction. Pg.7, added Pt2000 range & excitation current source. Pg.8, added 3-wire mode excitation current description. Pg.8, added Temp Threshold Detect & Status and Interrupts. Pg.10, changed Update Rate to Sample Rate. Pg.11, added Pt2000 to RTD Type. Pg.12, added Suspend Background Maintenance Operations Register sub-section. Pg.16, added Summary Status. Pg.18, changed Update Rate to Sample Rate. Pg.19, added Suspend Background Maintenance Operations/Run Open-Line/Run BIT offsets. Pg.21, added Summary Status offsets. Pg.22-24, added Interrupt Registers offsets.

Module Manual - Status and Interrupts Revision History		
Revision	**Revision Date**	**Description**
C	2021-11-30	C08896; Transition manual to docbuilder format - no technical info change.

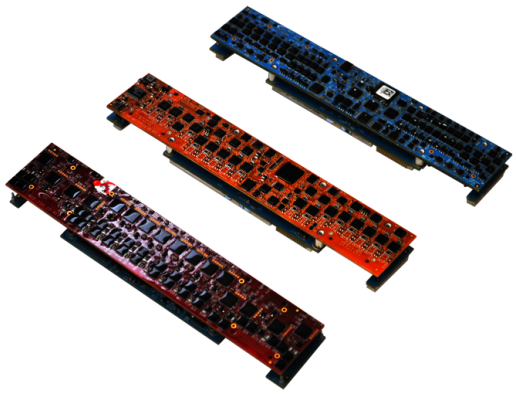
RT1 Measurement & Simulation Modules Thermocouple and RTD Measurement Function Modules

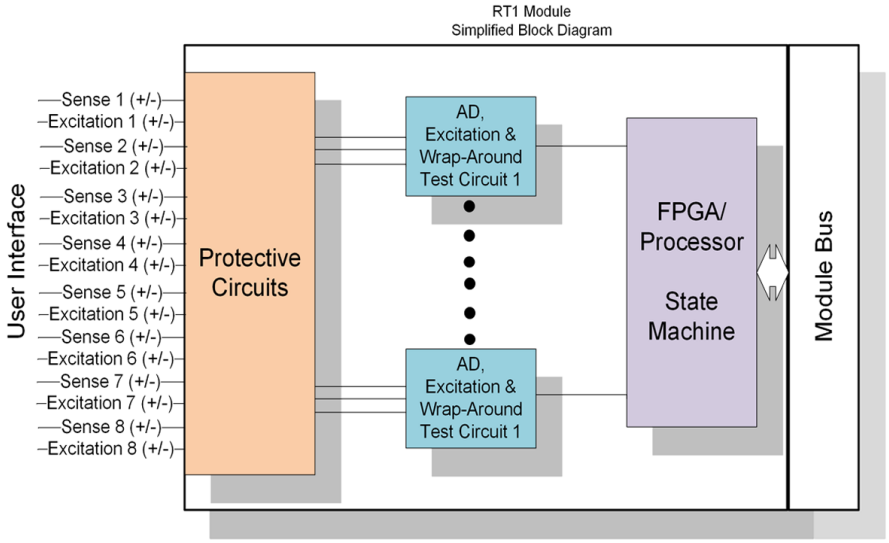
8 Channels, RTD Measurement

The module provides 8 measurement channels, and can be programmed for interfacing to 2, 3, and 4-wire platinum RTD sensor configurations. The 4-wire mode is the most accurate, providing excellent stability and repeatability. The RTD channels feature individual A/D converters and precision excitation/current drive. Programmable lead-wire compensation is provided for inherently less accurate 2 and 3-wire configurations. All RTD channels are self-aligning because each channel is automatically "aligned" on a rotating basis to eliminate offset and gain errors throughout the operating envelope. Programmability for expected resistance range and wire modes allows for optimization of scaling/resolution, as well as flexibility in reading many RTD types.

The RT1 is used to measure temperature by providing the measured resistance of the RTD element. By correlating the characteristic resistance with the temperature algorithm of the specific RTD utilized, the temperature can be resolved. Most RTD elements consist of a length of fine coiled wire wrapped around a ceramic or glass core. The element is usually quite fragile, so it is often placed inside a sheathed probe to protect it. The RTD element is made from a pure material, typically platinum, nickel, or copper. The material has a predictable change in resistance as the temperature changes; it is this predictable change that is used to determine temperature.

Due to higher accuracy and repeatability, RTDs are increasing in use as compared with thermocouples in many industrial/embedded and test applications below 600° C.





Features

- **RTD Measurement**
  - Higher accuracy and repeatability as compared with thermocouples in applications below 600 °C
  - Two, three or four-wire mode
  - Channels are calibrated at the factory for Pt100, Pt500, Pt1000 and Pt2000 RTDs
  - Single Precision Floating Point Value (IEEE-754) format
  - Open sensor connections are detected and reported
  - 1 mA, 500 µA, 250 µA & 100 µA excitation sources for Pt100, Pt500, Pt1000 and Pt2000 ranges
- **Independently Programmable**
  - Up to 8 RTD channels
- **Programmable Sample Rate**
  - Sets the sampling rate of the A/D
- **Offset Temperature**
  - Provides ability to null for any system induced measurement errors

Specifications

Number of Channels	**8 Channels**
Analog Input Resolution	**24-bits per channel**
RTD Interface	**4, 3, or 2-wire RTD interface capability. Specifically designed for use with 100Ω, 500Ω, 1000Ω and 2000Ω RTDs, or any RTD whose maximum operating resistance is less than 8000Ω.**
Open Line Detection	**Ability to detect an open in any line or RTD in all wire modes.**
Excitation	**1mA (Pt100), 500µA (Pt500), 250µA (Pt1000) or 100µA (Pt2000) for 2- & 4-wire mode; 500µA (Pt100), 250µA (Pt500), 125µA (P1000) or 50µA (Pt2000) for 3-wire mode**
Accuracy	**±0.1% of full-scale value @ 5 samples per second (4-wire mode only), ±0.2% of full-scale value @ 5 samples per second (3-wire mode only), ±1.2% of full-scale value @ 5 samples per second (2-wire mode only)**
Sample Rate	**Programmable between 3 - 4800 Hz**
Output Format	**Resistance/Temperature**
ESD Protection	**Designed to meet the testing requirements of IEC 801-2 Level 2. (4 KV transient with a peak current of 7.5 A and a Tc of approximately 60 ns.)**
Power	**5 VDC @ 450 mA typical**
Ground	**All channel grounds are common and are isolated from system ground.**
Weight	**1.5 oz. (42 g)**

Architected for Versatility

NAI's Configurable Open Systems Architecture™ (COSA®) offers a choice of over 100 smart I/O, communications, or Ethernet switch functions, providing the highest packaging density and greatest flexibility of ruggedized embedded product solutions in the industry. Preexisting, fully-tested functions can be combined in an unlimited number of ways quickly and easily.

One-Source Efficiencies

Eliminate man-months of integration with a configured, field-proven system from NAI. Specification to deployment is a seamless experience as all design, state-of-the-art manufacturing, assembly and test are performed - by one trusted source. All facilities are located within the U.S. and optimized for high-mix/low volume production runs and extended lifecycle support.

Product Lifecycle Management

From design to production and beyond, NAI's product lifecycle management strategy ensures the long-term availability of COTS products through configuration management, technology refresh and obsolescence component purchase and storage.

INTRODUCTION

This module manual provides information about the North Atlantic Industries, Inc. (NAI) Resistance Temperature Detector Function Module: RT1. This module is compatible with all NAI Generation 5 motherboards.

Resistance Temperature Detector (RTD) RT1 is a 32-Bit module that provides eight measurement channels, and can be programmed for interfacing to two, three and four-wire platinum RTD sensor configurations. The four-wire mode is the most accurate, providing excellent stability and repeatability.

FEATURES

- 8 measurement channels
- Higher accuracy and repeatability as compared with thermocouples in applications below 600° C
- Two, three or four-wire mode
- Channels are calibrated at the factory

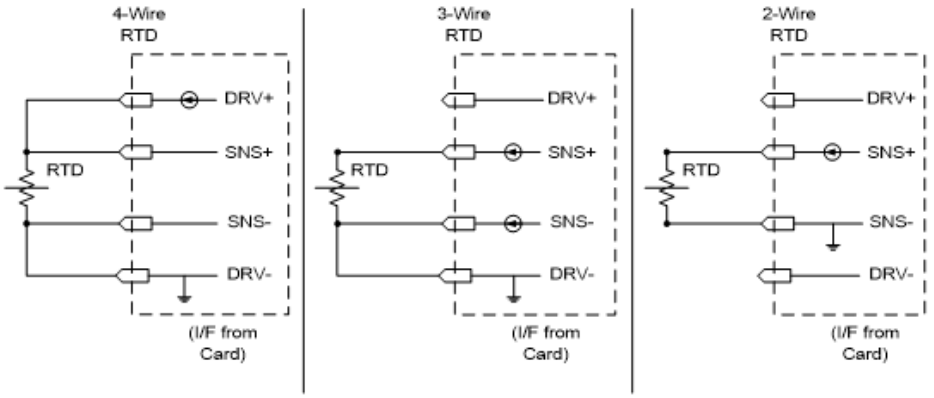
PRINCIPLE OF OPERATION

The RT1 provides 8 RTD measurement channels. Each channel is configurable for use with 4-wire, 3-wire, or 2-wire connections to the RTD sensors. All RTD channels are calibrated at the factory and measurement results provided in Single Precision Floating Point Value (IEEE-754) format.

Open sensor connections are detected and reported in an Open status word. A 1mA excitation source is used for resistance measurement in the lowest range setting (Pt100). For the Pt500, Pt1000 and Pt2000 ranges, the excitation current sources are 500µA, 250µA, and 100µA respectively.

Lead resistance correction is available in 2-wire mode, allowing for user compensation of cabling resistance in the measurement system. Resistance values may be entered for individual channels, which are subtracted from the measurement. The temperature measurements will reflect the compensated resistance values for the RTD for direct readout of the corrected temperature readings.

As with the Thermocouple mode, the temperature offset provisions allow for the same nulling of the temperature offsets in the system and operate similarly.



2-Wire is the simplest resistance measurement configuration, requiring only two wires per sensor. Measurements will be very sensitive to test cabling, as the excitation current and voltage measurements are through the same wires. Short or low resistance test leads are needed for accurate readings. Provisions for nulling the test lead resistances are provided via individual 2-wire offset resistance registers, allowing direct readings of the compensated measurements on a per channel basis.

3-Wire configuration relies on balanced test lead resistance on the Sense (+) and Sense (-) wires, so the voltage drop across each of the two current source lines are equal and cancel each other out. The differential voltage reading between the two sense lines along with the excitation current through the resistor is used in the resulting calculation of the sensor resistance. The test lead wire length will not affect the measurement provided the two lines are equal in resistance and is often the best compromise between wiring requirements and accuracy. In 3-Wire mode, the excitation current is split in half. Half of the total current flows on each of the sense lines (see 3-Wire RTD connection diagram).

4-Wire configuration provides optimal accuracy, allowing precise measurements without any constraints of short or balanced test leads, but requires 4 wires per sensor. The two sense wires measure the voltage at the sensor independently without undue influence of voltage drops due to the excitation current. This configuration is recommended where accuracy is a priority over the additional wiring requirements.

Built-In-Test (BIT)/Diagnostic Capability

Automatic background BIT testing is provided. Each channel is checked for correct A/D operation using an on-board 100 Ω nominal resistor. The open input detection test applies a 0.5 µA current to the A/D converter inputs. The FPGA then tests for a full-scale reading, indicating an open circuit. Any failure triggers an interrupt, if enabled, with the results available in the status registers. The testing is totally transparent to the user and has no effect on the operation of this module. It can be enabled or disabled. It is enabled by default.

Temperature Threshold Detect Programming

The RT1 provides the ability to program two temperature thresholds that will result in temperature alerts. For each threshold, a “low” and a “high” threshold value is specified that will be used to set the Temperature Alert statuses. The *Temperature Threshold Low* registers sets the threshold values to use to set the *Temperature Alert Low* status bit when the Temperature reading is below the low temperature threshold value. Conversely, the *Temperature Threshold High* registers sets the threshold values to use to set the *Temperature Alert High* status bit when the Temperature reading is above the high temperature threshold value. These threshold values are individually configurable on a per channel basis.

A possible usage of the two temperature thresholds is to use the first threshold detection levels as an early warning pre-alarm level and the second threshold detection levels as an alarm limit value. For this purpose, the Detect 2 thresholds should be set at larger deviation values from the nominal temperature than the Detect 1 thresholds.

For example:

[Threshold Low 2] < [Threshold Low 1] < [Nominal Temperature] < [Threshold High 1] < [Threshold High 2]

This allows the Detect 1 thresholds to serve as a pre-alert warning of temperature excursion, while Detect 2 may represent an alarm condition. Note: these detect thresholds are not necessarily set in this order and may be independently set either way.

Status and Interrupts

The RT1 Function Module provide registers that indicate faults or events. Refer to “Status and Interrupts Module Manual” for the Principle of Operation description.

REGISTER DESCRIPTIONS

The register descriptions provide the register name, Function Address Offset, Type, Data Range, Read or Write information, Initialized Value, a description of the function and, in most cases, a data table.

RT1 Measurement Registers

Temperature (°C)

Function: Measures the temperature of the RTD sensor.

Type: Single Precision Floating Point Value (IEEE-754)

Data Range: N/A

Read/Write: R

Initialized Value: N/A

Operational Settings: RTD temperature measurement in degrees Celsius.

Temperature (°F)

Function: Measures the temperature of the thermocouple/ RTD sensor.

Type: Single Precision Floating Point Value (IEEE-754)

Data Range: N/A

Read/Write: R

Initialized Value: N/A

Operational Settings: RTD temperature measurement in degrees Fahrenheit.

Resistance

Function: Measures resistance of the RTD sensor.

Type: Single Precision Floating Point Value (IEEE-754)

Data Range: N/A

Read/Write: R

Initialized Value: N/A

Operational Settings: Measures resistance in ohms. This measurement may optionally be adjusted through a user entry of a 2-wire lead resistance compensation value.

RT1 Control Registers

RTD or Thermocouple

Function: Indicates whether the module is an RTD or Thermocouple.

Data Range: 0 or 1

Read/Write: R

Initialized Value: 1 (RTD Mode)

Operational Settings: On the RT1 the value of the *RTD or Thermocouple* register is set to **1** for Resistance Temperature Detector (RTD) mode.

RTD or Thermocouple															
D31	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**D**

Sample Rate

Function: Sets the sampling rate of the sensor.

Type: unsigned binary word (32-bit)

Data Range: 0x00 to 0x27 (See table)

Read/Write: R/W

Initialized Value: 0x27 (3 Hz)

Operational Settings: Set the value based on Sample Rate table. Note: lower rates provide greater stability and accuracy in the readings. Per channel configuration.

Sample Rate Register Value	Update Frequency (Hz)	Sample Rate Register Value	Update Frequency (Hz)
0x0	4800	0x14	75
0x1	2400	0x15	64
0x2	1600	0x16	60
0x3	1200	0x17	50
0x4	960	0x18	48
0x5	800	0x19	40
0x6	600	0x1A	32
0x7	480	0x1B	30
0x8	400	0x1C	25
0x9	320	0x1D	24
0xA	300	0x1E	20
0xB	240	0x1F	16
0xC	200	0x20	15
0xD	192	0x21	12
0xE	160	0x22	10
0xF	150	0x23	8
0x10	120	0x24	6
0x11	100	0x25	5
0x12	96	0x26	4
0x13	80	0x27	3

Sample Rate															
D31	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
0	0	0	0	0	0	0	0	0	0	**D**	**D**	**D**	**D**	**D**	**D**

RTD Type

Function: RTD nominal resistance at 0°C.

Type: Single Precision Floating Point Value (IEEE-754)

Data Range: See table

Read/Write: R/W

Initialized Value: 100.0 (0x42C8 0000 in floating point)

Operational Settings: Set the RTD Type as specified in the table.

RTD Type	Description
Pt100	0-100 Ω RTD; resistance range of 0 Ω to approximately 500 Ω
Pt500	0-500 Ω RTD; resistance range of 0 Ω to approximately 2000 Ω
Pt1000	0-1000 Ω.RTD; resistance range of 0 Ω to approximately 4000 Ω
Pt2000	0-2000 Ω.RTD; resistance range of 0 Ω to approximately 8000 Ω

Wire Measurement Mode

Function: Sets the RTD sensor configuration: 2, 3 or 4 wire.

Type: unsigned binary word (32-bit)

Data Range: 2 – 4

Read/Write: R/W

Initialized Value: 2 (Value is set to 2-wire default whenever channel configuration mode is changed to RTD.

Operational Settings: Set the Wire Measurement Mode as specified in the table.

Wire Measurement Mode Value	Description
2	2-wire configuration
3	3-wire configuration
4	4-wire configuration

Wire Measurement Mode															
D31	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
0	0	0	0	0	0	0	0	0	0	0	0	0	**D**	**D**	**D**

2-Wire Lead Resistance Compensation

Function: Set a user defined compensation resistance in ohms, primarily required for channels that are configured for 2-wire configuration in the *Wire Measurement Mode* register. This allows test lead or cabling resistances to be cancelled out when using a 2-wire configuration. The *Resistance* measurement reading is adjusted by subtracting the value set in this register to null test lead and cabling resistance. This resistance offset is also applied in 3 and 4 wire modes, though typically not required in those modes.

Type: Single Precision Floating Point Value (IEEE-754)

Data Range: N/A

Read/Write: R/W

Initialized Value: 0.0

Operational Settings: Set the TOTAL lead resistance to be subtracted from the resistance measurement and reported in the *Resistance* register.

Suspend Background Maintenance Operations Register

The default configuration of the module is to run periodic self-test and calibration at 30 second intervals. During these operations, updates to the measurement readings are briefly suspended. For time critical measurements, such as using a channel for low voltage A/D measurements, the periodic internal processes may optionally be suspended for continuous and uninterrupted readings. During this suspended time, the maintenance operations for open-line detect may be triggered manually by the application at suitable intervals.

Suspend Background Maintenance Operations

Function: Holds off the performance of periodic maintenance routines for open line status checking and built in test (BIT). Used for dynamic measurements for continuous reading updates without interruption from the brief maintenance operations.

Type: unsigned binary word (32-bit)

Data Range: 0x0000 0000 to 0x0000 00FF

Read/Write: R/W

Initialized Value: 0 (All channels run maintenance operations on a scheduled basis)

Operational Settings: Suspends periodic operations for open line status check and BIT. Set to **0** to perform periodic operations for channel (default). Set bit to **1** to suspend the background maintenance operations for the specified channel.

Disable Maintenance Operations															
D31	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
0	0	0	0	0	0	0	0	**Ch8**	**Ch7**	**Ch6**	**Ch5**	**Ch4**	**Ch3**	**Ch2**	**Ch1**

Run Open-Line Check

Function: Triggers check for open or unconnected channels to update the open status indication. This is only used when the periodic schedule has been disabled for time critical measurements. This allows the application to run the routine in between measurement sessions.

Type: unsigned binary word (32-bit)

Data Range: 0x0000 0000 to 0x0000 00FF

Read/Write: R/W

Initialized Value: 0

Operational Settings: Write a **1** to the corresponding bit for the channel. Bit is self-clearing and will reset to zero on completion of the routine.

Run Open-line check															
D31	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
0	0	0	0	0	0	0	0	**Ch8**	**Ch7**	**Ch6**	**Ch5**	**Ch4**	**Ch3**	**Ch2**	**Ch1**

Run BIT

Function: Triggers Built-In-Test to detect out of tolerance conditions on the measurement circuitry. Only used when the periodic schedule for the channel has been disabled for time critical measurements. This allows the user to run the routine in between measurement sessions.

Type: unsigned binary word (32-bit)

Data Range: 0x0000 0000 to 0x0000 00FF

Read/Write: R/W

Initialized Value: 0

Operational Settings: Write a **1** to the corresponding bit for the channel. Bit is self-clearing and will reset to zero on completion of the routine.

Run BIT															
D31	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
0	0	0	0	0	0	0	0	**Ch8**	**Ch7**	**Ch6**	**Ch5**	**Ch4**	**Ch3**	**Ch2**	**Ch1**

Temperature Threshold Detect Programming

The RTD Temperature Threshold registers provide the ability program two temperature thresholds that will result in temperature alerts.

Temperature Threshold Detect 1

A “low” and a “high” threshold value is specified for each temperature threshold that will be used to set the Temperature Alert statuses. The *Temperature Threshold Low 1* register sets the threshold value to use to set the *Temperature Alert Low 1* status bit when the Temperature reading is below the low temperature threshold value. Conversely, the *Temperature Threshold High 1* register sets the threshold values to use to set the *Temperature Alert High 1* status bit when the Temperature reading is above the high temperature threshold value. These threshold values are individually configurable on a per channel basis.

Temperature Threshold Low 1

Function: Sets Temperature Threshold Low 1 value in degrees Celsius for each channel.

Type: Single Precision Floating Point Value (IEEE-754)

Data Range: N/A

Read/Write: R/W

Initialized Value: -40° C

Operational Settings: If the temperature drops below this set value, then a *Temperature Alert Low 1 Status* will be set. An interrupt will occur if the *Temperature Alert Low 1 Interrupt Enable* register is set to **1**.

Temperature Threshold High 1

Function: Sets Temperature Threshold High 1 value in degrees Celsius for each channel.

Type: Single Precision Floating Point Value (IEEE-754)

Data Range: N/A

Read/Write: R/W

Initialized Value: 25° C

Operational Settings: If the temperature exceeds the set value, then a *Temperature Alert High 1 Status* will be set. An interrupt will occur if the *Temperature Alert High 1 Interrupt Enable* register is set to **1**.

Temperature Threshold Detect 2

A “low” and a “high” threshold value is specified for each temperature threshold that will be used to set the Temperature Alert statuses. The *Temperature Threshold Low 2* register sets the threshold value to use to set the *Temperature Alert Low 2* status bit when the Temperature reading is below the low temperature threshold value. Conversely, the *Temperature Threshold High 2* register sets the threshold values to use to set the *Temperature Alert High 2* status bit when the Temperature reading is above the high temperature threshold value. These threshold values are individually configurable on a per channel basis.

Temperature Threshold Low 2

Function: Sets Temperature Threshold Low 2 value in degrees Celsius for each channel.

Type: Single Precision Floating Point Value (IEEE-754)

Data Range: N/A

Read/Write: R/W

Initialized Value: 0°C

Operational Settings: If the temperature drops below the set value, then a *Temperature Alert Low 2 Status* will be set. An interrupt will occur if the *Temperature Alert Low 2 Interrupt Enable* register is set to **1**.

Temperature Threshold High 2

Function: Sets Alert Temperature High 2 value in degrees Celsius for each channel.

Type: Single Precision Floating Point Value (IEEE-754)

Data Range: N/A

Read/Write: R/W

Initialized Value: 100° C

Operational Settings: If the temperature exceeds the set value, then a *Temperature Alert High 2 Status* will be set. An interrupt will occur if the *Temperature Alert High 2 Interrupt Enable* register is set to **1.**

Status and Interrupt Registers

The RT1 Module provides status registers for BIT, Open, and Temperature Alert.

Channel Status Enabled

Function: Determines whether to update the status for the channels. This feature can be used to “mask” status bits of unused channels in status registers that are bitmapped by channel.

Type: unsigned binary word (32-bit)

Data Range: 0x0000 0000 to 0x0000 00FF (Channel Status)

Read/Write: R/W

Initialized Value: 0x0000 00FF

Operational Settings: When the bit corresponding to a given channel in the Channel Status Enabled register is not enabled (**0**) the status will be masked and report “0” or “no failure”. This applies to all statuses that are bitmapped by channel (BIT Status, Open Status, Temperature Alerts and Summary Status). Note, Background BIT will continue to run even if the Channel Status Enabled is set to ‘0’.

Channel Status Enabled															
D31	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
0	0	0	0	0	0	0	0	**Ch8**	**Ch7**	**Ch6**	**Ch5**	**Ch4**	**Ch3**	**Ch2**	**Ch1**

BIT Status

There are four registers associated with the BIT Status: *Dynamic*, *Latched*, *Interrupt Enable*, and *Set Edge/Level Interrupt*.

BIT Dynamic Status															
BIT Latched Status															
BIT Interrupt Enable															
BIT Set Edge/Level Interrupt															
D31	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
0	0	0	0	0	0	0	0	**Ch8**	**Ch7**	**Ch6**	**Ch5**	**Ch4**	**Ch3**	**Ch2**	**Ch1**

Function: Indicates the corresponding channels associated with the channel’s BIT status or configuration

Type: unsigned binary word (32-bit)

Data Range: 0x0000 0000 to 0x0000 00FF

Read/Write: R (*Dynamic*), R/W (*Latched*, *Interrupt Enable*, *Edge/Level Interrupt*)

Initialized Value: 0

Open Status

There are four registers associated with the Open Status: *Dynamic*, *Latched*, I*nterrupt Enable*, and *Set Edge/Level Interrupt*.

Open Dynamic Status															
Open Latched Status															
Open Interrupt Enable															
Open Set Edge/Level Interrupt															
D31	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
0	0	0	0	0	0	0	0	**Ch8**	**Ch7**	**Ch6**	**Ch5**	**Ch4**	**Ch3**	**Ch2**	**Ch1**

Function: Sets the corresponding bit associated with the channel's Open status indication for an unconnected input.

Type: unsigned binary word (32-bit)

Data Range: 0x0000 0000 to 0x0000 00FF

Read/Write: R (*Dynamic*), R/W (*Latched*, *Interrupt Enable*, *Edge/Level Interrupt*)

Initialized Value: 0

Temperature Alert Status

There are four registers associated with each of the Temperature Alert Statuses: *Dynamic*, *Latched*, *Interrupt Enable*, and *Set Edge/Level Interrupt*.

Temperature Alert Low 1 Dynamic Status															
Temperature Alert Low 1 Latched Status															
Temperature Alert Low 1 Interrupt Enable															
Temperature Alert Low 1 Set Edge/Level Interrupt															
Temperature Alert High 1 Dynamic Status															
Temperature Alert High 1 Latched Status															
Temperature Alert High 1 Interrupt Enable															
Temperature Alert High 1 Set Edge/Level Interrupt															
Temperature Alert Low 2 Dynamic Status															
Temperature Alert Low 2 Latched Status															
Temperature Alert Low 2 Interrupt Enable															
Temperature Alert Low 2 Set Edge/Level Interrupt															
Temperature Alert High 2 Dynamic Status															
Temperature Alert High 2 Latched Status															
Temperature Alert High 2 Interrupt Enable															
Temperature Alert High 2 Set Edge/Level Interrupt															
D31	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
0	0	0	0	0	0	0	0	**Ch8**	**Ch7**	**Ch6**	**Ch5**	**Ch4**	**Ch3**	**Ch2**	**Ch1**

Function: Sets the corresponding bit associated with the channel's Temperature Alert indication for temperature readings that are below or above the associated thresholds.

Type: unsigned binary word (32-bit)

Data Range: 0x0000 0000 to 0x0000 00FF

Read/Write: R (*Dynamic*), R/W (*Latched*, *Interrupt Enable*, *Edge/Level Interrupt*)

Initialized Value: 0

Summary Status

There are four registers associated with the Summary Status: *Dynamic, Latched, Interrupt Enable,* and *Set Edge/Level Interrupt.*

Summary Status Dynamic Status															
Summary Status Latched Status															
Summary Status Interrupt Enable															
Summary Status Set Edge/Level Interrupt															
D31	**D30**	**D29**	**D28**	**D27**	**D26**	**D25**	**D24**	**D23**	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
0	0	0	0	0	0	0	0	**Ch8**	**Ch7**	**Ch6**	**Ch5**	**Ch4**	**Ch3**	**Ch2**	**Ch1**

Function: Sets the corresponding bit when a fault is detected for BIT or Open on that channel.

Type: unsigned binary word (32-bit)

Data Range: 0x0000 0000 to 0x0000 00FF

Read/Write: R (*Dynamic*), R/W (*Latched*, *Interrupt Enable*, *Edge/Level Interrupt*)

Initialized Value: 0

Summary Events Table

Module	BIT	Overcurrent	External Power Loss	Open Line	External Power Under Volt	External Power Over Volt	Over Temp	Surge Suppressor Fault
RT1	X			X				

Interrupt Vector and Steering

When interrupts are enabled, the interrupt vector associated with the specific interrupt can be programmed (typically with a unique number/identifier) such that it can be utilized in the Interrupt Service Routine (ISR) to identify the type of interrupt. When an interrupt occurs, the contents of the Interrupt Vector registers is reported as part of the interrupt mechanism.

In addition to specifying the interrupt vector, the interrupt can be directed (“steered”) to the native bus or to the application running on the onboard ARM processor.

Note, the Interrupt Vector and Interrupt Steering registers are mapped to the Motherboard Common Memory and these registers are associated with the Module Slot position (refer to Function Register Map).

Interrupt Vector

Function: Set an identifier for the interrupt.

Type: unsigned binary word (32-bit)

Data Range: 0 to 0xFFFF FFFF

Read/Write: R/W

Initialized Value: 0

Operational Settings: When an interrupt occurs, this value is reported as part of the interrupt mechanism.

Interrupt Steering

Function: Sets where to direct the interrupt.

Type: unsigned binary word (32-bit)

Data Range: See Table

Read/Write: R/W

Initialized Value: 0 = Not Defined

Operational Settings: When an interrupt occurs, the interrupt is sent as specified:

Direct Interrupt to VME	1
Direct Interrupt to ARM Processor (via SerDes) *(Custom App on ARM or NAI Ethernet Listener App)*	2
Direct Interrupt to PCIe Bus	5
Direct Interrupt to cPCI Bus	6

FUNCTION REGISTER MAP

Key: ***Bold Italic = Configuration/Control***

Bold Underline = Measurement/Status

*When an event is detected, the bit associated with the event is set in this register and will remain set until the user clears the event bit. Clearing the bit requires writing a 1 back to the specific bit that was set when read (i.e. write-1-to-clear, writing a '1' to a bit set to '1' will set the bit to '0').

~ Data is always in Floating Point.

RT1 Measurement Registers

0x1004	**Temperature (°C) Ch 1~**	R
0x1044	**Temperature (°C) Ch 2~**	R
0x1084	**Temperature (°C) Ch 3~**	R
0x10C4	**Temperature (°C) Ch 4~**	R
0x1104	**Temperature (°C) Ch 5~**	R
0x1144	**Temperature (°C) Ch 6~**	R
0x1184	**Temperature (°C) Ch 7~**	R
0x11C4	**Temperature (°C) Ch 8~**	R

0x1008	**Temperature (°F) Ch 1~**	R
0x1048	**Temperature (°F) Ch 2~**	R
0x1088	**Temperature (°F) Ch 3~**	R
0x10C8	**Temperature (°F) Ch 4~**	R
0x1108	**Temperature (°F) Ch 5~**	R
0x1148	**Temperature (°F) Ch 6~**	R
0x1188	**Temperature (°F) Ch 7~**	R
0x11C8	**Temperature (°F) Ch 8~**	R

0x1000	**Resistance Ch 1~**	R
0x1040	**Resistance Ch 2~**	R
0x1080	**Resistance Ch 3~**	R
0x10C0	**Resistance Ch 4~**	R
0x1100	**Resistance Ch 5~**	R
0x1140	**Resistance Ch 6~**	R
0x1180	**Resistance Ch 7~**	R
0x11C0	**Resistance Ch 8~**	R

RT1 Control Registers

0x2000	**RTD or Thermocouple**	R

0x1028	***Sample Rate Ch 1***	R/W
0x1068	***Sample Rate Ch 2***	R/W
0x10A8	***Sample Rate Ch 3***	R/W
0x10E8	***Sample Rate Ch 4***	R/W
0x1128	***Sample Rate Ch 5***	R/W
0x1168	***Sample Rate Ch 6***	R/W
0x11A8	***Sample Rate Ch 7***	R/W
0x11E8	***Sample Rate Ch 8***	R/W

0x100C	*RTD Type Ch 1*	R/W
0x104C	*RTD Type Ch 2*	R/W
0x108C	*RTD Type Ch 3*	R/W
0x10CC	*RTD Type Ch 4*	R/W
0x110C	*RTD Type Ch 5*	R/W
0x114C	*RTD Type Ch 6*	R/W
0x118C	*RTD Type Ch 7*	R/W
0x11CC	*RTD Type Ch 8*	R/W

0x1010	*Wire Measurement Mode Ch 1*	R/W
0x1050	*Wire Measurement Mode Ch 2*	R/W
0x1090	*Wire Measurement Mode Ch 3*	R/W
0x10D0	*Wire Measurement Mode Ch 4*	R/W
0x1110	*Wire Measurement Mode Ch 5*	R/W
0x1150	*Wire Measurement Mode Ch 6*	R/W
0x1190	*Wire Measurement Mode Ch 7*	R/W
0x11D0	*Wire Measurement Mode Ch 8*	R/W

0x1014	*2-Wire Lead Resistance Compensation Ch 1~*	R/W
0x1054	*2-Wire Lead Resistance Compensation Ch 2~*	R/W
0x1094	*2-Wire Lead Resistance Compensation Ch 3~*	R/W
0x10D4	*2-Wire Lead Resistance Compensation Ch 4~*	R/W
0x1114	*2-Wire Lead Resistance Compensation Ch 5~*	R/W
0x1154	*2-Wire Lead Resistance Compensation Ch 6~*	R/W
0x1194	*2-Wire Lead Resistance Compensation Ch 7~*	R/W
0x11D4	*2-Wire Lead Resistance Compensation Ch 8~*	R/W

0x2008	*Suspend Background Operations*	R/W
0x2010	*Run Open-Line Check*	R/W
0x2014	*Run BIT*	R/W

Temperature Threshold Detect Programming Registers

0x1018	*Alert Temperature Low 1 Ch 1~*	R/W
0x1058	*Alert Temperature Low 1 Ch 2~*	R/W
0x1098	*Alert Temperature Low 1 Ch 3~*	R/W
0x10D8	*Alert Temperature Low 1 Ch 4~*	R/W
0x1118	*Alert Temperature Low 1 Ch 5~*	R/W
0x1158	*Alert Temperature Low 1 Ch 6~*	R/W
0x1198	*Alert Temperature Low 1 Ch 7~*	R/W
0x11D8	*Alert Temperature Low 1 Ch 8~*	R/W

0x101C	*Alert Temperature Low 2 Ch 1~*	R/W
0x105C	*Alert Temperature Low 2 Ch 2~*	R/W
0x109C	*Alert Temperature Low 2 Ch 3~*	R/W
0x10DC	*Alert Temperature Low 2 Ch 4~*	R/W
0x111C	*Alert Temperature Low 2 Ch 5~*	R/W
0x115C	*Alert Temperature Low 2 Ch 6~*	R/W
0x119C	*Alert Temperature Low 2 Ch 7~*	R/W
0x11DC	*Alert Temperature Low 2 Ch 8~*	R/W

0x1020	***Alert Temperature High 1 Ch 1~***	R/W
0x1060	***Alert Temperature High 1 Ch 2~***	R/W
0x10A0	***Alert Temperature High 1 Ch 3~***	R/W
0x10E0	***Alert Temperature High 1 Ch 4~***	R/W
0x1120	***Alert Temperature High 1 Ch 5~***	R/W
0x1160	***Alert Temperature High 1 Ch 6~***	R/W
0x11A0	***Alert Temperature High 1 Ch 7~***	R/W
0x11E0	***Alert Temperature High 1 Ch 8~***	R/W

0x1024	***Alert Temperature High 2 Ch 1~***	R/W
0x1064	***Alert Temperature High 2 Ch 2~***	R/W
0x10A4	***Alert Temperature High 2 Ch 3~***	R/W
0x10E4	***Alert Temperature High 2 Ch 4~***	R/W
0x1124	***Alert Temperature High 2 Ch 5~***	R/W
0x1164	***Alert Temperature High 2 Ch 6~***	R/W
0x11A4	***Alert Temperature High 2 Ch 7~***	R/W
0x11E4	***Alert Temperature High 2 Ch 8~***	R/W

RT1 Status Registers

0x02B0	***Channel Status Enabled***	R/W

BIT Registers

0x0800	**Dynamic Status**	R
0x0804	**Latched Status***	R/W
0x0808	***Interrupt Enable***	R/W
0x080C	***Set Edge/Level Interrupt***	R/W

Status Registers

Open

0x0810	**Dynamic Status**	R
0x0814	**Latched Status***	R/W
0x0818	***Interrupt Enable***	R/W
0x081C	***Set Edge/Level Interrupt***	R/W

Temperature Alert Low 1

0x0820	**Dynamic Status**	R
0x0824	**Latched Status***	R/W
0x0828	***Interrupt Enable***	R/W
0x082C	***Set Edge/Level Interrupt***	R/W

Temperature Alert Low 2

0x0830	**Dynamic Status**	R
0x0834	**Latched Status***	R/W
0x0838	***Interrupt Enable***	R/W
0x083C	***Set Edge/Level Interrupt***	R/W

Temperature Alert High 1

0x0840	**Dynamic Status**	R
0x0844	**Latched Status***	R/W
0x0848	***Interrupt Enable***	R/W
0x084C	***Set Edge/Level Interrupt***	R/W

Temperature Alert High 2

0x0850	**Dynamic Status**	R
0x0854	**Latched Status***	R/W
0x0858	***Interrupt Enable***	R/W
0x085C	***Set Edge/Level Interrupt***	R/W

Summary

0x09A0	**Dynamic Status**	R
0x09A4	**Latched Status***	R/W
0x09A8	***Interrupt Enable***	R/W
0x09AC	***Set Edge/Level Interrupt***	R/W

Interrupt Registers

The Interrupt Vector and Interrupt Steering registers are located on the Motherboard Memory Space and do not require any Module Address Offsets. These registers are accessed using the absolute addresses listed in the table below.

0x0500	***Module 1 Interrupt Vector 1 - BIT***	R/W
0x0504	***Module 1 Interrupt Vector 2 - Open***	R/W
0x0508	***Module 1 Interrupt Vector 3 - Temperature Alert Low 1***	R/W
0x050C	***Module 1 Interrupt Vector 4 - Temperature Alert Low 2***	R/W
0x0510	***Module 1 Interrupt Vector 5 - Temperature Alert High 1***	R/W
0x0514	***Module 1 Interrupt Vector 6 - Temperature Alert High 2***	R/W
0x0518 to 0x0564	***Module 1 Interrupt Vector 7-26 - Reserved***	R/W
0x0568	***Module 1 Interrupt Vector 27 - Summary***	R/W
0x056C to 0x057C	***Module 1 Interrupt Vector 28-32 - Reserved***	R/W

0x0600	***Module 1 Interrupt Steering 1 - BIT***	R/W
0x0604	***Module 1 Interrupt Steering 2 - Open***	R/W
0x0608	***Module 1 Interrupt Steering 3 - Temperature Alert Low 1***	R/W
0x060C	***Module 1 Interrupt Steering 4 - Temperature Alert Low 2***	R/W
0x0610	***Module 1 Interrupt Steering 5 - Temperature Alert High 1***	R/W
0x0614	***Module 1 Interrupt Steering 6 - Temperature Alert High 2***	R/W
0x0618 to 0x0664	***Module 1 Interrupt Steering 7-26 - Reserved***	R/W
0x0668	***Module 1 Interrupt Steering 27 - Summary***	R/W
0x066C to 0x067C	***Module 1 Interrupt Steering 28-32 - Reserved***	R/W

0x0700	***Module 2 Interrupt Vector 1 - BIT***	R/W
0x0704	***Module 2 Interrupt Vector 2 - Open***	R/W
0x0708	***Module 2 Interrupt Vector 3 - Temperature Alert Low 1***	R/W
0x070C	***Module 2 Interrupt Vector 4 - Temperature Alert Low 2***	R/W
0x0710	***Module 2 Interrupt Vector 5 - Temperature Alert High 1***	R/W
0x0714	***Module 2 Interrupt Vector 6 - Temperature Alert High 2***	R/W
0x0718 to 0x0764	***Module 2 Interrupt Vector 7-26 - Reserved***	R/W
0x0768	***Module 2 Interrupt Vector 27 - Summary***	R/W
0x076C to 0x077C	***Module 2 Interrupt Vector 28-32 - Reserved***	R/W

0x0800	***Module 2 Interrupt Steering 1 - BIT***	R/W
0x0804	***Module 2 Interrupt Steering 2 - Open***	R/W
0x0808	***Module 2 Interrupt Steering 3 - Temperature Alert Low 1***	R/W
0x080C	***Module 2 Interrupt Steering 4 - Temperature Alert Low 2***	R/W
0x0810	***Module 2 Interrupt Steering 5 - Temperature Alert High 1***	R/W
0x0814	***Module 2 Interrupt Steering 6 - Temperature Alert High 2***	R/W
0x0818 to 0x0864	***Module 2 Interrupt Steering 7-26 - Reserved***	R/W
0x0868	***Module 2 Interrupt Steering 27 - Summary***	R/W
0x086C to 0x087C	***Module 2 Interrupt Steering 28-32 - Reserved***	R/W

0x0900	***Module 3 Interrupt Vector 1 - BIT***	R/W
0x0904	***Module 3 Interrupt Vector 2 - Open***	R/W
0x0908	***Module 3 Interrupt Vector 3 - Temperature Alert Low 1***	R/W
0x090C	***Module 3 Interrupt Vector 4 - Temperature Alert Low 2***	R/W
0x0910	***Module 3 Interrupt Vector 5 - Temperature Alert High 1***	R/W
0x0914	***Module 3 Interrupt Vector 6 - Temperature Alert High 2***	R/W
0x0918 to 0x0964	***Module 3 Interrupt Vector 7-26 - Reserved***	R/W
0x0968	***Module 3 Interrupt Vector 27 - Summary***	R/W
0x096C to 0x097C	***Module 3 Interrupt Vector 28-32 - Reserved***	R/W

0x0A00	***Module 3 Interrupt Steering 1 - BIT***	R/W
0x0A04	***Module 3 Interrupt Steering 2 - Open***	R/W
0x0A08	***Module 3 Interrupt Steering 3 - Temperature Alert Low 1***	R/W
0x0A0C	***Module 3 Interrupt Steering 4 - Temperature Alert Low 2***	R/W
0x0A10	***Module 3 Interrupt Steering 5 - Temperature Alert High 1***	R/W
0x0A14	***Module 3 Interrupt Steering 6 - Temperature Alert High 2***	R/W
0x0A18 to 0x0A64	***Module 3 Interrupt Steering 7-26 - Reserved***	R/W
0x0A68	***Module 3 Interrupt Steering 27 - Summary***	R/W
0x0A6C to 0x0A7C	***Module 3 Interrupt Steering 28-32 - Reserved***	R/W

0x0B00	***Module 4 Interrupt Vector 1 - BIT***	R/W
0x0B04	***Module 4 Interrupt Vector 2 - Open***	R/W
0x0B08	***Module 4 Interrupt Vector 3 - Temperature Alert Low 1***	R/W
0x0B0C	***Module 4 Interrupt Vector 4 - Temperature Alert Low 2***	R/W
0x0B10	***Module 4 Interrupt Vector 5 - Temperature Alert High 1***	R/W
0x0B14	***Module 4 Interrupt Vector 6 - Temperature Alert High 2***	R/W
0x0B18 to 0x0B64	***Module 4 Interrupt Vector 7-26 - Reserved***	R/W
0x0B68	***Module 4 Interrupt Vector 27 - Summary***	R/W
0x0B6C to 0x0B7C	***Module 4 Interrupt Vector 28-32 - Reserved***	R/W

0x0C00	***Module 4 Interrupt Steering 1 - BIT***	R/W
0x0C04	***Module 4 Interrupt Steering 2 - Open***	R/W
0x0C08	***Module 4 Interrupt Steering 3 - Temperature Alert Low 1***	R/W
0x0C0C	***Module 4 Interrupt Steering 4 - Temperature Alert Low 2***	R/W
0x0C10	***Module 4 Interrupt Steering 5 - Temperature Alert High 1***	R/W
0x0C14	***Module 4 Interrupt Steering 6 - Temperature Alert High 2***	R/W
0x0C18 to 0x0C64	***Module 4 Interrupt Steering 7-26 - Reserved***	R/W
0x0C68	***Module 4 Interrupt Steering 27 - Summary***	R/W
0x0C6C to 0x0C7C	***Module 4 Interrupt Steering 28-32 - Reserved***	R/W

0x0D00	***Module 5 Interrupt Vector 1 - BIT***	R/W
0x0D04	***Module 5 Interrupt Vector 2 - Open***	R/W
0x0D08	***Module 5 Interrupt Vector 3 - Temperature Alert Low 1***	R/W
0x0D0C	***Module 5 Interrupt Vector 4 - Temperature Alert Low 2***	R/W
0x0D10	***Module 5 Interrupt Vector 5 - Temperature Alert High 1***	R/W
0x0D14	***Module 5 Interrupt Vector 6 - Temperature Alert High 2***	R/W
0x0D18 to 0x0D64	***Module 5 Interrupt Vector 7-26 - Reserved***	R/W
0x0D68	***Module 5 Interrupt Vector 27 - Summary***	R/W
0x0D6C to 0x0D7C	***Module 5 Interrupt Vector 28-32 - Reserved***	R/W

0x0E00	***Module 5 Interrupt Steering 1 - BIT***	R/W
0x0E04	***Module 5 Interrupt Steering 2 - Open***	R/W
0x0E08	***Module 5 Interrupt Steering 3 - Temperature Alert Low 1***	R/W
0x0E0C	***Module 5 Interrupt Steering 4 - Temperature Alert Low 2***	R/W
0x0E10	***Module 5 Interrupt Steering 5 - Temperature Alert High 1***	R/W
0x0E14	***Module 5 Interrupt Steering 6 - Temperature Alert High 2***	R/W
0x0E18 to 0x0E64	***Module 5 Interrupt Steering 7-26 - Reserved***	R/W
0x0E68	***Module 5 Interrupt Steering 27 - Summary***	R/W
0x0E6C to 0x0E7C	***Module 5 Interrupt Steering 28-32 - Reserved***	R/W

0x0F00	***Module 6 Interrupt Vector 1 - BIT***	R/W
0x0F04	***Module 6 Interrupt Vector 2 - Open***	R/W
0x0F08	***Module 6 Interrupt Vector 3 - Temperature Alert Low 1***	R/W
0x0F0C	***Module 6 Interrupt Vector 4 - Temperature Alert Low 2***	R/W
0x0F10	***Module 6 Interrupt Vector 5 - Temperature Alert High 1***	R/W
0x0F14	***Module 6 Interrupt Vector 6 - Temperature Alert High 2***	R/W
0x0F18 to 0x0F64	***Module 6 Interrupt Vector 7-26 - Reserved***	R/W
0x0F68	***Module 6 Interrupt Vector 27 - Summary***	R/W
0x0F6C to 0x0F7C	***Module 6 Interrupt Vector 28-32 - Reserved***	R/W

0x1000	***Module 6 Interrupt Steering 1 - BIT***	R/W
0x1004	***Module 6 Interrupt Steering 2 - Open***	R/W
0x1008	***Module 6 Interrupt Steering 3 - Temperature Alert Low 1***	R/W
0x100C	***Module 6 Interrupt Steering 4 - Temperature Alert Low 2***	R/W
0x1010	***Module 6 Interrupt Steering 5 - Temperature Alert High 1***	R/W
0x1014	***Module 6 Interrupt Steering 6 - Temperature Alert High 2***	R/W
0x1018 to 0x1064	***Module 6 Interrupt Steering 7-26 - Reserved***	R/W
0x1068	***Module 6 Interrupt Steering 27 - Summary***	R/W
0x106C to 0x107C	***Module 6 Interrupt Steering 28-32 - Reserved***	R/W

APPENDIX: PIN-OUT DETAILS

Pin-out details (for reference) are shown below, with respect to DATAIO. Additional information on pin-outs can be found in the Motherboard Operational Manuals.

Module Signal (Ref Only)	RTD (RT1)
DATIO1	SENSE+_CH1
DATIO2	SENSE-_CH1
DATIO3	DRIVE+_CH1
DATIO4	DRIVE-_CH1
DATIO5	DRIVE+_CH2
DATIO6	DRIVE-_CH2
DATIO7	SENSE+_CH2
DATIO8	SENSE-_CH2
DATIO9	SENSE+_CH3
DATIO10	SENSE-_CH3
DATIO11	DRIVE+_CH3
DATIO12	DRIVE-_CH3
DATIO13	SENSE+_CH5
DATIO14	SENSE-_CH5
DATIO15	DRIVE+_CH5
DATIO16	DRIVE-_CH5
DATIO17	DRIVE+_CH6
DATIO18	DRIVE-_CH6
DATIO19	SENSE+_CH6
DATIO20	SENSE-_CH6
DATIO21	SENSE+_CH7
DATIO22	SENSE-_CH7
DATIO23	DRIVE+_CH7
DATIO24	DRIVE-_CH7
DATIO25	DRIVE+_CH4
DATIO26	DRIVE-_CH4
DATIO27	SENSE+_CH4
DATIO28	SENSE-_CH4
DATIO29	DRIVE+_CH8
DATIO30	DRIVE-_CH8
DATIO31	SENSE+_CH8
DATIO32	SENSE-_CH8
DATIO33	
DATIO34	
DATIO35	
DATIO36	
DATIO37	
DATIO38	
DATIO39	
DATIO40	
N/A	




Status and Interrupts

MODULE MANUAL

STATUS AND INTERRUPTS

Status registers indicate the detection of faults or events. The status registers can be channel bit-mapped or event bit-mapped. An example of a channel bit-mapped register is the BIT status register, and an example of an event bit-mapped register is the FIFO status register.

For those status registers that allow interrupts to be generated upon the detection of the fault or the event, there are four registers associated with each status: *Dynamic, Latched, Interrupt Enabled*, and *Set Edge/Level Interrupt*.

Dynamic Status: The *Dynamic Status* register indicates the *current* condition of the fault or the event. If the fault or the event is momentary, the contents in this register will be clear when the fault or the event goes away. The *Dynamic Status* register can be polled, however, if the fault or the event is sporadic, it is possible for the indication of the fault or the event to be missed.

Latched Status: The *Latched Status* register indicates whether the fault or the event *has occurred* and keeps the state until it is cleared by the user. Reading the *Latched Status* register is a better alternative to polling the *Dynamic Status* register because the contents of this register will not clear until the user commands to clear the specific bit(s) associated with the fault or the event in the *Latched Status* register. Once the status register has been read, the act of writing a **1** back to the applicable status register to any specific bit (channel/event) location will "clear" the bit (set the bit to **0**). When clearing the channel/event bits, it is strongly recommended to write back the same bit pattern as read from the *Latched Status* register. For example, if the channel bit-mapped *Latched Status* register contains the value 0x0000 0005, which indicates fault/event detection on channel 1 and 3, write the value 0x0000 0005 to the *Latched Status* register to clear the fault/event status for channel 1 and 3. Writing a "1" to other channels that are not set (example 0x0000 000F) may result in incorrectly "clearing" incoming faults/events for those channels (example, channel 2 and 4).

Interrupt Enable: If interrupts are preferred upon the detection of a fault or an event, enable the specific channel/event interrupt in the *Interrupt Enable* register. The bits in *Interrupt Enable* register map to the same bits in the *Latched Status* register. When a fault or event occurs, an interrupt will be fired. Subsequent interrupts will not trigger until the application acknowledges the fired interrupt by clearing the associated channel/event bit in the *Latched Status* register. If the interruptible condition is still persistent after clearing the bit, this may retrigger the interrupt depending on the *Edge/Level* setting.

Set Edge/Level Interrupt: When interrupts are enabled, the condition on retriggering the interrupt after the Latch Register is "cleared" can be specified as "edge" triggered or "level" triggered. Note, the Edge/Level Trigger also affects how the Latched Register value is adjusted after it is "cleared" (see below).

- *Edge triggered*: An interrupt will be retriggered when the Latched Status register change from low (0) to high (1) state. Uses for edge-triggered interrupts would include transition detections (Low-to-High transitions, High-to-Low transitions) or fault detections. After "clearing" an interrupt, another interrupt will not occur until the next transition or the re-occurrence of the fault again.
- *Level triggered*: An interrupt will be generated when the Latched Status register remains at the high (1) state. Level-triggered interrupts are used to indicate that something needs attention.

Interrupt Vector and Steering

When interrupts are enabled, the interrupt vector associated with the specific interrupt can be programmed with a unique number/identifier defined by the user such that it can be utilized in the Interrupt Service Routine (ISR) to identify the type of interrupt. When an interrupt occurs, the contents of the Interrupt Vector registers is reported as part of the interrupt mechanism. In addition to specifying the interrupt vector, the interrupt can be directed ("steered") to the native bus or to the application running on the onboard ARM processor.

Interrupt Trigger Types

In most applications, limiting the number of interrupts generated is preferred as interrupts are costly, thus choosing the correct Edge/Level interrupt trigger to use is important.

Example 1: Fault detection

This example illustrates interrupt considerations when detecting a fault like an “open” on a line. When an “open” is detected, the system will receive an interrupt. If the “open” on the line is persistent and the trigger is set to “edge”, upon “clearing” the interrupt, the system will not re-generate another interrupt. If, instead, the trigger is set to “level”, upon “clearing” the interrupt, the system will re-generate another interrupt. Thus, in this case, it will be better to set the trigger type to “edge”.

Example 2: Threshold detection

This example illustrates interrupt considerations when detecting an event like reaching or exceeding the “high watermark” threshold value. In a communication device, when the number of elements received in the FIFO reaches the high-watermark threshold, an interrupt will be generated. Normally, the application would read the count of the number of elements in the FIFO and read this number of elements from the FIFO. After reading the FIFO data, the application would “clear” the interrupt. If the trigger type is set to “edge”, another interrupt will be generated only if the number of elements in FIFO goes below the “high watermark” after the “clearing” the interrupt and then fills up to reach the “high watermark” threshold value. Since receiving communication data is inherently asynchronous, it is possible that data can continue to fill the FIFO as the application is pulling data off the FIFO. If, at the time the interrupt is “cleared”, the number of elements in the FIFO is at or above the “high watermark”, no interrupts will be generated. In this case, it will be better to set the trigger type to “level”, as the purpose here is to make sure that the FIFO is serviced when the number of elements exceeds the high watermark threshold value. Thus, upon “clearing” the interrupt, if the number of elements in the FIFO is at or above the “high watermark” threshold value, another interrupt will be generated indicating that the FIFO needs to be serviced.

Dynamic and Latched Status Registers Examples

The examples in this section illustrate the differences in behavior of the Dynamic Status and Latched Status registers as well as the differences in behavior of Edge/Level Trigger when the Latched Status register is cleared.

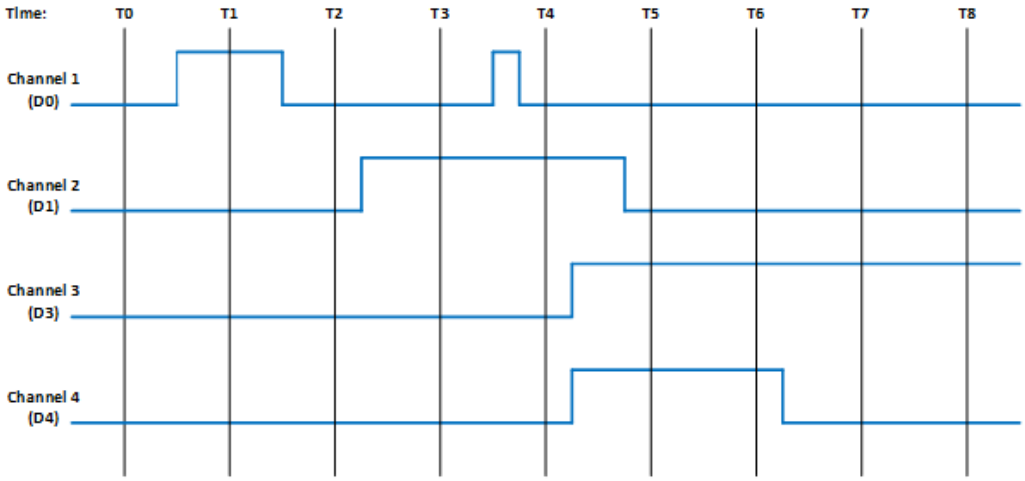


Figure 1. Example of Module's Channel-Mapped Dynamic and Latched Status States

Time	Dynamic Status	No Clearing of Latched Status	Clearing of Latched Status (Edge-Triggered)		Clearing of Latched Status (Level-Triggered)	
		Latched Status	Action	Latched Status	Action	Latched
T0	0x0	0x0	Read Latched Register	0x0	Read Latched Register	0x0
T1	0x1	0x1	Read Latched Register	0x1		0x1
			Write 0x1 to Latched Register		Write 0x1 to Latched Register	
				0x0		*0x1*
T2	0x0	0x1	Read Latched Register	0x0	Read Latched Register	0x1
					Write 0x1 to Latched Register	
						0x0
T3	0x2	0x3	Read Latched Register	0x2	Read Latched Register	0x2
			Write 0x2 to Latched Register		Write 0x2 to Latched Register	
				0x0		*0x2*
T4	0x2	0x3	Read Latched Register	0x1	Read Latched Register	0x3
			Write 0x1 to Latched Register		Write 0x3 to Latched Register	
				0x0		*0x2*
T5	0xC	0xF	Read Latched Register	0xC	Read Latched Register	0xE
			Write 0xC to Latched Register		Write 0xE to Latched Register	
				0x0		*0xC*
T6	0xC	0xF	Read Latched Register	0x0	Read Latched Register	0xC
					Write 0xC to Latched Register	
						0xC
T7	0x4	0xF	Read Latched Register	0x0	Read Latched Register	0xC
					Write 0xC to Latched Register	
						0x4
T8	0x4	0xF	Read Latched Register	0x0	Read Latched Register	0x4

Interrupt Examples

The examples in this section illustrate the interrupt behavior with Edge/Level Trigger.

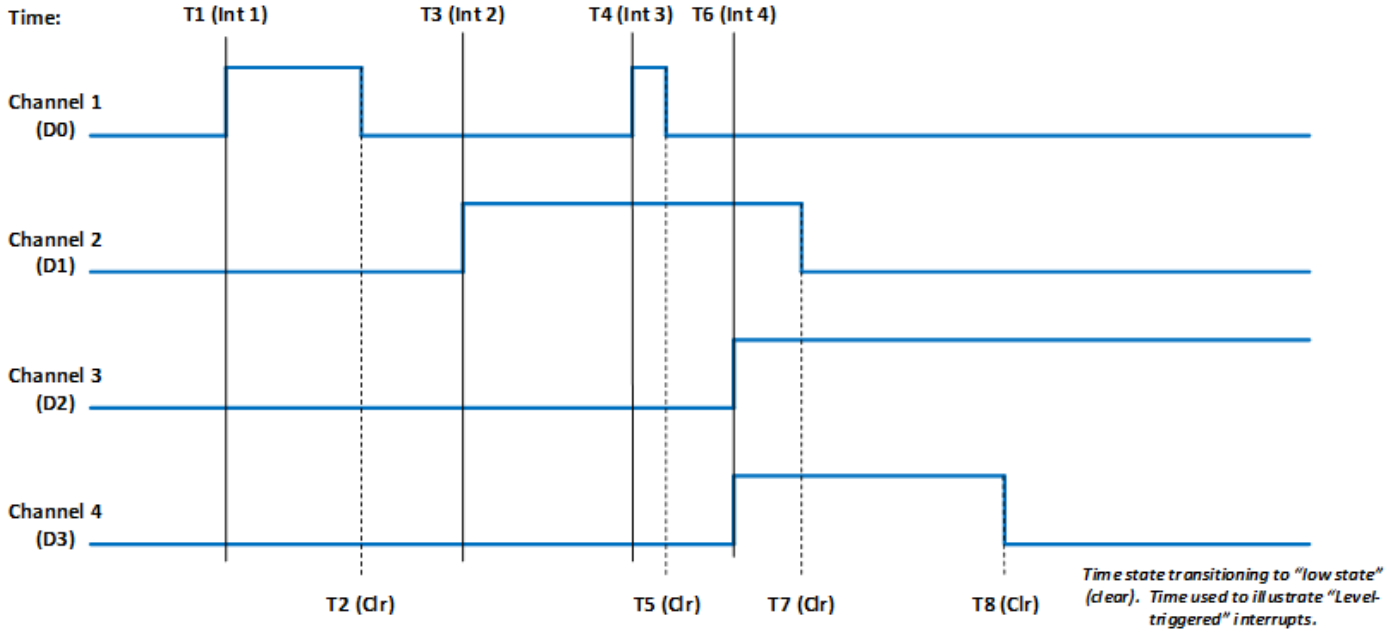


Figure 2. Illustration of Latched Status State for Module with 4-Channels with Interrupt Enabled

Time	Latched Status (Edge-Triggered – Clear Multi-Channel)		Latched Status (Edge-Triggered – Clear Single Channel)		Latched Status (Level-Triggered – Clear Multi-Channel)	
	Action	Latched	Action	Latched	Action	Latched
T1 (Int 1)	**Interrupt Generated** Read Latched Registers	0x1	**Interrupt Generated** Read Latched Registers	0x1	**Interrupt Generated** Read Latched Registers	0x1
	Write 0x1 to Latched Register		Write 0x1 to Latched Register		Write 0x1 to Latched Register	
		0x0		*0x0*	***Interrupt re-triggers*** Note, interrupt re-triggers after each clear until T2.	*0x1*
T3 (Int 2)	**Interrupt Generated** Read Latched Registers	0x2	**Interrupt Generated** Read Latched Registers	0x2	**Interrupt Generated** Read Latched Registers	0x2
	Write 0x2 to Latched Register		Write 0x2 to Latched Register		Write 0x2 to Latched Register	
		0x0		*0x0*	***Interrupt re-triggers*** Note, interrupt re-triggers after each clear until T7.	*0x2*
T4 (Int 3)	**Interrupt Generated** Read Latched Registers	0x1	**Interrupt Generated** Read Latched Registers	0x1	**Interrupt Generated** Read Latched Registers	0x3
	Write 0x1 to Latched Register		Write 0x1 to Latched Register		Write 0x3 to Latched Register	
		0x0		*0x0*	***Interrupt re-triggers*** Note, interrupt re-triggers after each clear and 0x3 is reported in Latched Register until T5.	*0x3*
					Interrupt re-triggers Note, interrupt re-triggers after each clear until T7.	*0x2*

Time	Latched Status (Edge-Triggered – Clear Multi-Channel)		Latched Status (Edge-Triggered – Clear Single Channel)		Latched Status (Level-Triggered – Clear Multi-Channel)	
	Action	Latched	Action	Latched	Action	Latched
T6 (Int 4)	**Interrupt Generated** Read Latched Registers	0xC	**Interrupt Generated** Read Latched Registers	0xC	**Interrupt Generated** Read Latched Registers	0xE
	Write 0xC to Latched Register		Write 0x4 to Latched Register		Write 0xE to Latched Register	
		0x0	***Interrupt re-triggers*** Write 0x8 to Latched Register	*0x8*	***Interrupt re-triggers*** Note, interrupt re-triggers after each clear and 0xE is reported in Latched Register until T7.	*0xE*
				0x0	***Interrupt re-triggers*** Note, interrupt re-triggers after each clear and 0xC is reported in Latched Register until T8.	*0xC*
					Interrupt re-triggers Note, interrupt re-triggers after each clear and 0x4 is reported in Latched Register always.	*0x4*

NAI Cares

North Atlantic Industries (NAI) is a leading independent supplier of Embedded I/O Boards, Single Board Computers, Rugged Power Supplies, Embedded Systems and Motion Simulation and Measurement Instruments for the Military, Aerospace and Industrial Industries. We accelerate our clients' time-to-mission with a unique approach based on a Configurable Open Systems Architecture™ (COSA®) that delivers the best of both worlds: custom solutions from standard COTS components.

We have built a reputation by listening to our customers, understanding their needs, and designing, testing and delivering board and system-level products for their most demanding air, land and sea requirements. If you have any applications or questions regarding the use of our products, please contact us for an expedient solution.

Please visit us at: www.naii.com or select one of the following for immediate assistance:

FAQ

http://www.naii.com/faqs

Application Notes

http://www.naii.com/applicationnotes

Calibration and Repairs

http://www.naii.com/calibrationrepairs

Call Us

(631) 567-1100

Accelerate Your Time-to-Mission™