*The Embedded I/O Company*

TEWS TECHNOLOGIES

# TXMC590

## 16 Channel Thermo-/Strain Measurement

Version 1.0

## User Manual

Issue 1.0.6

October 2019

**TEWS TECHNOLOGIES GmbH**

Am Bahnhof 7 25469 Halstenbek, Germany

Phone: +49 (0) 4101 4058 0 Fax: +49 (0) 4101 4058 19

e-mail: info@tews.com www.tews.com

## TXMC590-10R

16 Channel Thermo-/Strain Measurement



TEWS TECHNOLOGIES GmbH has made any effort to ensure that this manual is accurate and complete. However TEWS TECHNOLOGIES GmbH reserves the right to change the product described in this document at any time without notice.

TEWS TECHNOLOGIES GmbH is not liable for any damage arising out of the application or use of the device described herein.

**Style Conventions**

Hexadecimal characters are specified with prefix 0x, i.e. 0x029E (that means hexadecimal value 029E).

For signals on hardware products, an ‚Active Low' is represented by the signal name with # following, i.e. IP_RESET#.

Access terms are described as:

| | |
|---|---|
| W | Write Only |
| R | Read Only |
| R/W | Read/Write |
| R/C | Read/Clear |
| R/S | Read/Set |

| Issue | Description | Date |
|---|---|---|
| 0.0.1 | First Preliminary Issue | December 2015 |
| 0.0.2 | Second Preliminary Issue:<br>- Added Error Status Register in PCIBAR0 (Offset 0x088) | February 2016 |
| 1.0.0 | Initial Issue<br>- Changed the Table Header Format<br>- Changed the functionality of the Calibrate Bit in the Channel Control Registers<br>- Added the possibility to set the cold junction to “0” from the outside | September 2016 |
| 1.0.1 | - Corrected the description of the Error Status Register<br>- Corrected the description of the Interrupt Status Error Register<br>- Added the error indication and handling in the “Getting a temperature/strain value” chapter<br>- Corrected the Access Mode of the Calibrate Bit in the Channel x Control Register which is r/w and not r/s | April 2017 |
| 1.0.2 | - Clarification that because of the non-isolated inputs of the TXMC590 only ungrounded thermocouples can be used | October 2017 |
| 1.0.3 | - Removed the “Resolution” from the Technical Specification Table, because this value has no meaning for the overall accuracy of the TXMC590 and might be misleading<br>- Added a note to the table concept description mentioning the interpolation between two table data values | January 2018 |
| 1.0.4 | - Corrected the PCIe Generation from 2.1 to 1.1 | February 2019 |
| 1.0.5 | - Added figures and description for the JTAG connectors and DIP switch to clarify the intended factory use only | July 2019 |
| 1.0.6 | - Added a chapter to clarify the connection of the external I2C temperature sensor | October 2019 |

# Table of Contents

# List of Figures

# List of Tables

# 1 Product Description

The Thermo-/Strain-Measurement XMC module TXMC590 is able to measure thermocouples, resistive sensors like RTD and thermistors, and strain gauges.

It provides 16 independent non-isolated measurement channels. Each channel consists of a differential analog input and a differential sensor excitation output. The setup of the analog in-/outputs, the signal conditioning and the translation into data values is handled by a microcontroller with an integrated ADC and current source.

The TXMC590 is suitable for measuring any kind of ungrounded thermocouples. The cold junction compensation is done onboard, freeing the user application from this task. There are three possibilities to measure the cold junction temperature:

1. A second channel configured as resistive temperature sensor
2. An onboard I2C temperature sensor
3. An external I2C temperature sensor (not included)

For measuring RTDs ratio metric, there is a high precision reference resistor onboard. The excitation current is set up automatically according to the RTD channel settings.

All Strain Gauge types are supported (e.g. quarter, half and full bridge configurations). However, there is no bridge completion circuit onboard the TXMC590. Therefore, if quarter or half bridges are used, these have to be completed outside of the module. The excitation current is set up automatically according to the strain gauge channel settings. As well, every channel can be calibrated and the measured value is cleared of offset errors.

Each channel can separately be configured for any sensor type. The measured value is translated into a data value according to the sensor type, i.e. into temperature or microstrain. The translation may be set up to provide the data in a format that allows direct usage without further calculations. Measurements can be set up to operate periodically with different time bases or can be triggered via register access.

The TXMC590 features predefined parameter sets for the most common sensor types. Additionally, up to 16 user-defined parameter sets can be persistently stored onboard. These user-defined parameter sets allow supporting even user defined sensors. A software tool to build these user-defined parameter sets is part of TXMC590-SW-xx.

For special customer requirements, the microcontroller is also programmable via PCI Express which allows developing and programming special firmware into the controller.

Software Support (TXMC590-SW-xx) is available for different operating systems.

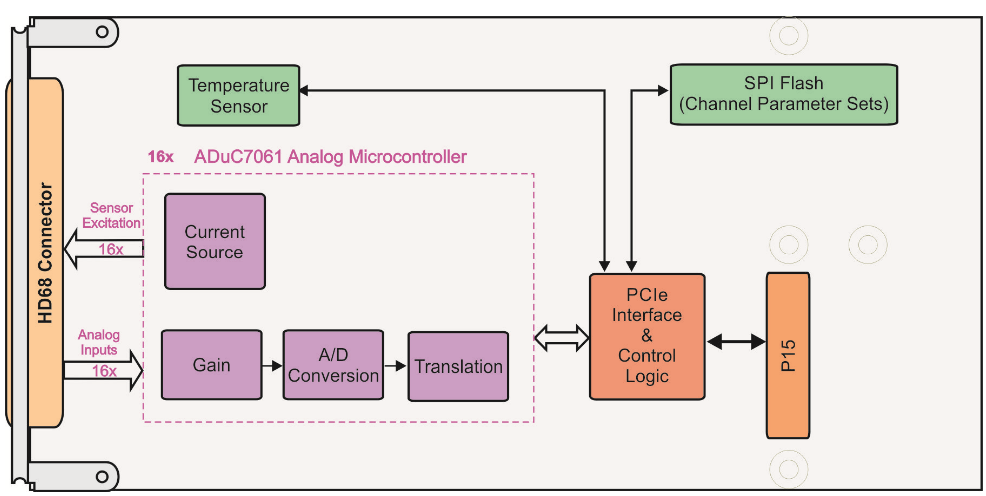



Figure 1-1 : Block Diagram

# 2 Technical Specification

## 2.1 General Specification

| | | |
|---|---|---|
| **XMC Interface** | | |
| **Mechanical Interface** | Switched Mezzanine Card (XMC) Interface conforming to ANSI/VITA 42.0-2008 (R2014)<br>Standard single-width (149mm x 74mm) | |
| **Electrical Interface** | PCI Express (Base Specification 1.1) up to x4 compliant interface conforming to ANSI/VITA 42.3-2006 (R2014)<br>(XMC PCI Express Protocol Layer Standard) | |
| **Onboard Devices** | | |
| **Analog Microcontroller** | ADUC7061 (Analog Devices) | |
| **IPMI FRU Data EEPROM** | M24C02 (STMicroelectronics) | |
| **I/O Interface** | | |
| **Number of Channels** | 16 | |
| **I/O Connector** | 3M N10268-52E2PC | |
| **Analog Inputs** | | |
| **Measurement Duration** | Thermocouple: 22ms<br>RTD: 22ms<br>Strain Gauge: 74ms<br><br>All times are total times from setting the conversion trigger to the time when valid data is ready to be read from the Channel Data Register. | |
| **Excitation Source** | 0 – 2mA (dependent on sensor type) | |
| **Physical Data** | | |
| **Power Requirements** | 400mA typical @ VPWR = +5V DC<br>200mA typical @ VPWR = +12V DC<br><br>2mA typical @ +3.3V DC<br>2mA typical @ +3.3Vaux DC<br>+12V / -12V not used | |
| **Temperature Range** | Operating<br>Storage | -40°C to +85°C<br>-40°C to +85°C |
| **MTBF** | 464000 h<br>MTBF values shown are based on calculation according to MIL-HDBK-217F and MIL-HDBK-217F Notice 2; Environment: $G_B$ 20°C.<br>The MTBF calculation is based on component FIT rates provided by the component suppliers. If FIT rates are not available, MIL-HDBK-217F and MIL-HDBK-217F Notice 2 formulas are used for FIT rate calculation. | |
| **Humidity** | 5 – 95 % non-condensing | |
| **Weight** | 80g | |

Table 2-1 : Technical Specification

## 2.2 Accuracy

Specifying accuracies for temperature sensors is not done with simply presenting one value. The voltage to temperature curves of the temperature sensors are non-linear, varying from range to range. As a result, the accuracy of any sensor is highly dependent on the temperature range it is used in. The following table shows the accuracy that a TXMC590 is able to produce typically in the given temperature range.

The values in the table do not include the accuracy of the temperature sensors themselves.

| Sensor Type | | Range | Typical Accuracy |
|---|---|---|---|
| **Thermocouple** | **B** | +50°C to +1820°C | ±3.0°C |
| | **E** | -270°C to 0°C | ±0.5°C |
| | | 0°C to +1000°C | ±0.3°C |
| | **J** | 0°C to +679°C | ±0.5°C |
| | **K** | -270°C to 0°C | ±0.8°C |
| | | 0°C to +1370°C | ±0.5°C |
| | **N** | -270°C to 0°C | ±1.1°C |
| | | 0°C to +1300°C | ±0.5°C |
| | **T** | -270°C to 0°C | ±0.8°C |
| | | 0°C to +400°C | ±0.4°C |
| | **R** | -50°C to 0°C | ±3.6°C |
| | | 0°C to +1766°C | ±1.5°C |
| | **S** | -50°C to 0°C | ±3.6°C |
| | | 0°C to +1766°C | ±1.8°C |
| **RTD** | **PT100** | -200°C to +860°C | ±0.1°C |
| | **PT500** | -200°C to +860°C | ±0.1°C |
| | **PT1000** | -200°C to +860°C | ±0.1°C |
| | **NI100** | -60°C to +250°C | ±0.2°C |
| | **NI500** | -60°C to +250°C | ±0.1°C |
| | **NI1000** | -60°C to +250°C | ±0.1°C |
| **Strain Gauge** | | -18.75mV to +18.75mV | ±75µV |

Table 2-2 : Accuracy

# 3 Handling and Operation Instruction

## 3.1 ESD Protection

**The TXMC590 is sensitive to static electricity. Packing, unpacking and all other handling of the TXMC590 has to be done in an ESD/EOS protected Area.**

## 3.2 Selecting Thermocouples

**The TXMC590 is suitable for measuring any kind of ungrounded thermocouples.**

**Please refer to chapter "Thermocouples" for more details.**

# 4 PCIe Configuration

## 4.1 IDs

| ID Type | ID Setting |
|---|---|
| **Vendor ID** | 0x1498 |
| **Device ID** | 0x924E |
| **Revision ID** | 0x00 |
| **Subsystem ID** | 0x900A |
| **Subsystem Vendor ID** | 0x1498 |
| **Class Code** | 0x118000 |

Table 4-1 : PCIe ID Configuration

## 4.2 Base Address Register Configuration

| Base Address Register (BAR) | PCIe Space Mapping | Size (Byte) | Port Width (Bit) | Endian Mode | Description |
|---|---|---|---|---|---|
| 0 | MEM | 4K | 32 | Little | Control/Status Registers |
| 1 | MEM | 4K | 32 | Little | Correction Table |
| 2 | MEM | 64K | 32 | Little | In-Field Programming Data |

Table 4-2 : PCIe Base Address Register Configuration

## 4.3 BAR 0 – Address Map (Control/Status Register)

| Offset (Base = PCIe Base Address 0) | Size (Bit) | Register |
|---|---|---|
| 0x000 | 32 | Channel 0 Control Register |
| 0x004 | 32 | Channel 1 Control Register |
| 0x008 | 32 | Channel 2 Control Register |
| 0x00C | 32 | Channel 3 Control Register |
| 0x010 | 32 | Channel 4 Control Register |
| 0x014 | 32 | Channel 5 Control Register |
| 0x018 | 32 | Channel 6 Control Register |
| 0x01C | 32 | Channel 7 Control Register |
| 0x020 | 32 | Channel 8 Control Register |
| 0x024 | 32 | Channel 9 Control Register |
| 0x028 | 32 | Channel 10 Control Register |
| 0x02C | 32 | Channel 11 Control Register |
| 0x030 | 32 | Channel 12 Control Register |
| 0x034 | 32 | Channel 13 Control Register |
| 0x038 | 32 | Channel 14 Control Register |
| 0x03C | 32 | Channel 15 Control Register |
| 0x040 | 32 | Channel 0 Data Register |
| 0x044 | 32 | Channel 1 Data Register |
| 0x048 | 32 | Channel 2 Data Register |
| 0x04C | 32 | Channel 3 Data Register |
| 0x050 | 32 | Channel 4 Data Register |
| 0x054 | 32 | Channel 5 Data Register |
| 0x058 | 32 | Channel 6 Data Register |
| 0x05C | 32 | Channel 7 Data Register |
| 0x060 | 32 | Channel 8 Data Register |
| 0x064 | 32 | Channel 9 Data Register |
| 0x068 | 32 | Channel 10 Data Register |
| 0x06C | 32 | Channel 11 Data Register |
| 0x070 | 32 | Channel 12 Data Register |
| 0x074 | 32 | Channel 13 Data Register |
| 0x078 | 32 | Channel 14 Data Register |
| 0x07C | 32 | Channel 15 Data Register |
| 0x080 | 32 | Configuration Trigger Register |
| 0x084 | 32 | Conversion Trigger Register |
| 0x088 | 32 | Error Status Register |
| 0x08C | 32 | Interrupt Control Register |
| 0x090 | 32 | Interrupt Status Register |
| 0x094 | 32 | Interrupt Status Configuration Done Register |
| 0x098 | 32 | Interrupt Status Error Register |

| | | |
|---|---|---|
| 0x09C | 32 | Cold Junction I2C Onboard Control Register |
| 0x0A0 | 32 | Cold Junction I2C External Control Register |
| 0x0A4 | 32 | Cold Junction Channel 00 Control Register |
| 0x0A8 | 32 | Cold Junction Channel 01 Control Register |
| 0x0AC | 32 | Cold Junction Channel 02 Control Register |
| 0x0B0 | 32 | Cold Junction Channel 03 Control Register |
| 0x0B4 | 32 | Cold Junction Channel 04 Control Register |
| 0x0B8 | 32 | Cold Junction Channel 05 Control Register |
| 0x0BC | 32 | Cold Junction Channel 06 Control Register |
| 0x0C0 | 32 | Cold Junction Channel 07 Control Register |
| 0x0C4 | 32 | Cold Junction Channel 08 Control Register |
| 0x0C8 | 32 | Cold Junction Channel 09 Control Register |
| 0x0CC | 32 | Cold Junction Channel 10 Control Register |
| 0x0D0 | 32 | Cold Junction Channel 11 Control Register |
| 0x0D4 | 32 | Cold Junction Channel 12 Control Register |
| 0x0D8 | 32 | Cold Junction Channel 13 Control Register |
| 0x0DC | 32 | Cold Junction Channel 14 Control Register |
| 0x0E0 | 32 | Cold Junction Channel 15 Control Register |
| 0x0E4 | 32 | Cold Junction Onboard Data Register |
| 0x0E8 | 32 | Cold Junction External Data Register |
| 0x0EC | 32 | Table Control Register |
| 0x0F0 | 32 | FPGA Firmware Revision Register |
| 0x0F4 | 32 | ADUC7061 Channel 00 Firmware Revision Register |
| 0x0F8 | 32 | ADUC7061 Channel 01 Firmware Revision Register |
| 0x0FC | 32 | ADUC7061 Channel 02 Firmware Revision Register |
| 0x100 | 32 | ADUC7061 Channel 03 Firmware Revision Register |
| 0x104 | 32 | ADUC7061 Channel 04 Firmware Revision Register |
| 0x108 | 32 | ADUC7061 Channel 05 Firmware Revision Register |
| 0x10C | 32 | ADUC7061 Channel 06 Firmware Revision Register |
| 0x110 | 32 | ADUC7061 Channel 07 Firmware Revision Register |
| 0x114 | 32 | ADUC7061 Channel 08 Firmware Revision Register |
| 0x118 | 32 | ADUC7061 Channel 09 Firmware Revision Register |
| 0x11C | 32 | ADUC7061 Channel 10 Firmware Revision Register |
| 0x120 | 32 | ADUC7061 Channel 11 Firmware Revision Register |
| 0x124 | 32 | ADUC7061 Channel 12 Firmware Revision Register |
| 0x128 | 32 | ADUC7061 Channel 13 Firmware Revision Register |
| 0x12C | 32 | ADUC7061 Channel 14 Firmware Revision Register |
| 0x130 | 32 | ADUC7061 Channel 15 Firmware Revision Register |

Table 4-3 : BAR 0 – Address Map (Control/Status Register)

## 4.4 BAR 1 – Address Map (Table Space)

| Offset (Base = PCIe Base Address 1) | | Type |
|---|---|---|
| Start | End | |
| 0x000 | 0x00E | Table Header |
| 0x010 | 0xFFF | Table Data |

Table 4-4 : BAR 1 – Table Space Address Map

## 4.5 BAR 2 – Address Map (In-Field Programming Space)

The Address Map of BAR 2 is not scope of this user manual. For updating the ADUC7061 or the FPGA, please consult the TXMC590 Update User Manual.

# 5 Register Description

## 5.1 BAR 0 – Control/Status Space

### 5.1.1 Channel x Control Registers (0x000 – 0x03C)

The 16 Channel Control Registers are used to enable and set up the channels. Each channel can be configured with a separate table, allowing 16 completely independent measurement channels. Each channel can be set up to issue interrupts, either after a conversion is done or if any of the status bits is set. Additionally, the conversion can be setup to happen periodically and automatically, and each channel may be calibrated before actual measurements are started.

**A write access to this register does not issue a configuration cycle automatically. As a result, it does not matter if a new correction table (if necessary) or the Channel Control Register is written first. A configuration cycle has to be started explicitly through the Configuration Trigger Register.**

**Bit 11 (Calibrate) is used to activate the ADuC internal self-calibration. This function calibrates the ADC against an internal reference inside the ADuC. The duration of the calibration process is typically ten times of a conversion. The calibration is best done on start-up and when the temperature of the system has changed. This can be measured with the onboard I2C temperature sensor.**

<table>
<tr><th>Bit</th><th>Symbol</th><th>Description</th><th>Access</th><th>Reset Value</th></tr>
<tr><td>31:16</td><td>CC_VALUE</td><td>Conversion Clock Value<br>These bits set the starting value for the clock timer. The timer runs with the time base set by CC_BASE.<br>Whenever the timer hits zero, a conversion is started.</td><td>r/w</td><td>0x0000</td></tr>
<tr><td>15:14</td><td>CC_BASE</td><td>Conversion Clock Time Base<br><table><tr><th>CC_BASE</th><th>Sample Clock Time Base</th></tr><tr><td>00</td><td>100 µs</td></tr><tr><td>01</td><td>100 ms</td></tr><tr><td>10</td><td>1 s</td></tr><tr><td>11</td><td>reserved</td></tr></table><br>Note: The minimum time required for a conversion has to be taken into consideration when choosing a conversion time value. Choosing a time value shorter than the conversion time does not make sense.</td><td>r/w</td><td>00</td></tr>
<tr><td>13</td><td>CONV_MODE</td><td>Channel Conversion Mode<br>1 periodically (CC_BASE + CC_VALUE)<br>0 by trigger (Conversion Trigger Register)</td><td>r/w</td><td>0</td></tr>
<tr><td>12</td><td>-</td><td>Reserved - always write as ‘0’, reads are undefined</td><td>-</td><td>-</td></tr>
<tr><td>11</td><td>CALIBRATE</td><td>Calibration<br>This bit enables the ADuC internal self-calibration procedure. If this bit is set, each configuration cycle triggers the calibration.</td><td>r/w</td><td>0</td></tr>
</table>

| | | | | |
|---|---|---|---|---|
| | | 1 enabled<br>0 disabled | | |
| 10 | IRQ_CONV | Conversion Interrupt enable (assert a PCIe interrupt when a conversion is done and the data can be read).<br>1 enabled<br>0 disabled | r/w | 0 |
| 9 | IRQ_CONF | Status Interrupt enable (assert a PCIe interrupt when the status has changed)<br>1 enabled<br>0 disabled | r/w | 0 |
| 8 | CH_EN | Channel enable<br>1 enabled<br>0 disabled<br>If the channel is enabled, the conversion mode is set up according to CONV_MODE bit and conversion speed is set up according to CC_BASE and CC_VALUE bits. | r/w | 0 |
| 7:0 | TABLE_NR | Table Number<br>0001 1111 User Defined Table #15<br>0001 1110 User Defined Table #14<br>0001 1101 User Defined Table #13<br>0001 1100 User Defined Table #12<br>0001 1011 User Defined Table #11<br>0001 1010 User Defined Table #10<br>0001 1001 User Defined Table #9<br>0001 1000 User Defined Table #8<br>0001 0111 User Defined Table #7<br>0001 0110 User Defined Table #6<br>0001 0101 User Defined Table #5<br>0001 0100 User Defined Table #4<br>0001 0011 User Defined Table #3<br>0001 0010 User Defined Table #2<br>0001 0001 User Defined Table #1<br>0001 0000 User Defined Table #0<br><br>Factory Defined Tables (cannot be overwritten)<br>0000 1111 Internal Self-Test<br>0000 1110 RTD Type NI1000<br>0000 1101 RTD Type NI500<br>0000 1100 RTD Type NI100<br>0000 1011 RTD Type PT1000<br>0000 1010 RTD Type PT500<br>0000 1001 RTD Type PT100<br>0000 1000 Reserved<br>0000 0111 Thermocouple Type B<br>0000 0110 Thermocouple Type S<br>0000 0101 Thermocouple Type R<br>0000 0100 Thermocouple Type T<br>0000 0011 Thermocouple Type E<br>0000 0010 Thermocouple Type N<br>0000 0001 Thermocouple Type J<br>0000 0000 Thermocouple Type K | r/w | 0x00 |

Table 5-1 : BAR 0 – Channel x Control Register

### 5.1.2 Channel x Data Registers (0x040 – 0x07C)

| Bit | Symbol | Description | Access | Reset Value |
|---|---|---|---|---|
| 31:0 | CH_X_DATA | Measured and translated data of the given channel.<br><br>Data representation, scale and unit depends on the settings in the corresponding table. | r | 0 |

Table 5-2 : BAR 0 – Channel X Data Register

### 5.1.3 Configuration Trigger Register (0x080)

Trigger the configuration of one or more channels by writing a ‘1’ to the appropriate bit(s). As soon as the configuration for a channel finished, the corresponding bit will reset to ‘0’ automatically. The channels that shall be configured have to be enabled and set up in the channel control register before writing to this register.

| Bit | Symbol | Description | Access | Reset Value |
|---|---|---|---|---|
| 31:16 | - | Reserved - always write as ‘0’, reads are undefined | - | - |
| 15 | CH_15_CONF_TRIG | Trigger configuration of channel X | r/s | 0 |
| 14 | CH_14_CONF_TRIG | | r/s | 0 |
| 13 | CH_13_CONF_TRIG | | r/s | 0 |
| 12 | CH_12_CONF_TRIG | | r/s | 0 |
| 11 | CH_11_CONF_TRIG | | r/s | 0 |
| 10 | CH_10_CONF_TRIG | | r/s | 0 |
| 9 | CH_09_CONF_TRIG | | r/s | 0 |
| 8 | CH_08_CONF_TRIG | | r/s | 0 |
| 7 | CH_07_CONF_TRIG | | r/s | 0 |
| 6 | CH_06_CONF_TRIG | | r/s | 0 |
| 5 | CH_05_CONF_TRIG | | r/s | 0 |
| 4 | CH_04_CONF_TRIG | | r/s | 0 |
| 3 | CH_03_CONF_TRIG | | r/s | 0 |
| 2 | CH_02_CONF_TRIG | | r/s | 0 |
| 1 | CH_01_CONF_TRIG | | r/s | 0 |
| 0 | CH_00_CONF_TRIG | | r/s | 0 |

Table 5-3 : BAR 0 – Configuration Trigger Register

### 5.1.4 Conversion Trigger Register (0x084)

Trigger the conversion of one or more channels by writing a '1' to the appropriate bit(s). As soon as the conversion for a channel finished, the corresponding bit will reset to '0' automatically. The channels that shall do the conversion have to be enabled and set up in the channel control register before writing to this register.

This register is ignored when the conversion is set up to be periodically in the Channel Control Register.

| Bit | Symbol | Description | Access | Reset Value |
|---|---|---|---|---|
| 31:16 | - | Reserved - always write as '0', reads are undefined | - | - |
| 15 | CH_15_CONV_TRIG | Trigger conversion of channel X | r/s | 0 |
| 14 | CH_14_CONV_TRIG | | r/s | 0 |
| 13 | CH_13_CONV_TRIG | | r/s | 0 |
| 12 | CH_12_CONV_TRIG | | r/s | 0 |
| 11 | CH_11_CONV_TRIG | | r/s | 0 |
| 10 | CH_10_CONV_TRIG | | r/s | 0 |
| 9 | CH_09_CONV_TRIG | | r/s | 0 |
| 8 | CH_08_CONV_TRIG | | r/s | 0 |
| 7 | CH_07_CONV_TRIG | | r/s | 0 |
| 6 | CH_06_CONV_TRIG | | r/s | 0 |
| 5 | CH_05_CONV_TRIG | | r/s | 0 |
| 4 | CH_04_CONV_TRIG | | r/s | 0 |
| 3 | CH_03_CONV_TRIG | | r/s | 0 |
| 2 | CH_02_CONV_TRIG | | r/s | 0 |
| 1 | CH_01_CONV_TRIG | | r/s | 0 |
| 0 | CH_00_CONV_TRIG | | r/s | 0 |

Table 5-4 : BAR 0 – Conversion Trigger Register

### 5.1.5 Error Status Register (0x088)

This register signals errors independent from the interrupt settings. It may be polled for errors in systems where using interrupts is not an option. The error status is cleared by writing a logic ‘1’ to the corresponding bits.

| Bit | Symbol | Description | Access | Reset Value |
|---|---|---|---|---|
| 31 | CH_15_CHAN_ERR | Channel X Channel Error<br><br>Possible causes:<br>- Timeout during the ADC measurement<br>- Timeout during the ADC self-calibration<br>- SPI communication problem | r/c | 0 |
| 30 | CH_14_CHAN_ERR | | r/c | 0 |
| 29 | CH_13_CHAN_ERR | | r/c | 0 |
| 28 | CH_12_CHAN_ERR | | r/c | 0 |
| 27 | CH_11_CHAN_ERR | | r/c | 0 |
| 26 | CH_10_CHAN_ERR | | r/c | 0 |
| 25 | CH_09_CHAN_ERR | | r/c | 0 |
| 24 | CH_08_CHAN_ERR | | r/c | 0 |
| 23 | CH_07_CHAN_ERR | | r/c | 0 |
| 22 | CH_06_CHAN_ERR | | r/c | 0 |
| 21 | CH_05_CHAN_ERR | | r/c | 0 |
| 20 | CH_04_CHAN_ERR | | r/c | 0 |
| 19 | CH_03_CHAN_ERR | | r/c | 0 |
| 18 | CH_02_CHAN_ERR | | r/c | 0 |
| 17 | CH_01_CHAN_ERR | | r/c | 0 |
| 16 | CH_00_CHAN_ERR | | r/c | 0 |
| 15 | CH_15_CONF_ERR | Channel X Configuration Error<br><br>Possible causes:<br>- Invalid table header content (sensor type, CRC, …)<br>- Invalid table data content (wrong number of data pairs)<br>- Measured value out of table | r/c | 0 |
| 14 | CH_14_CONF_ERR | | r/c | 0 |
| 13 | CH_13_CONF_ERR | | r/c | 0 |
| 12 | CH_12_CONF_ERR | | r/c | 0 |
| 11 | CH_11_CONF_ERR | | r/c | 0 |
| 10 | CH_10_CONF_ERR | | r/c | 0 |
| 9 | CH_09_CONF_ERR | | r/c | 0 |
| 8 | CH_08_CONF_ERR | | r/c | 0 |
| 7 | CH_07_CONF_ERR | | r/c | 0 |
| 6 | CH_06_CONF_ERR | | r/c | 0 |
| 5 | CH_05_CONF_ERR | | r/c | 0 |
| 4 | CH_04_CONF_ERR | | r/c | 0 |
| 3 | CH_03_CONF_ERR | | r/c | 0 |
| 2 | CH_02_CONF_ERR | | r/c | 0 |
| 1 | CH_01_CONF_ERR | | r/c | 0 |
| 0 | CH_00_CONF_ERR | | r/c | 0 |

Table 5-5 : BAR 0 – Error Status Register

### 5.1.6 Interrupt Control Register (0x08C)

There are two ways of acknowledging interrupts:

1. Acknowledge selected interrupts by writing a '1' to the appropriate interrupt status bit. This method is used when the IRQ_ACK_CONF bit is set to '0' (default).
2. Acknowledge all interrupts just by reading the appropriate register. This method is used when the IRQ_ACK_CONF bit is set to '1'.

**For a better understanding of the various interrupt status registers, see the following register descriptions.**

| Bit | Symbol | Description | Access | Reset Value |
|---|---|---|---|---|
| 31:1 | - | Reserved - always write as '0', reads are undefined | - | - |
| 0 | IRQ_ACK_CONF | Interrupt Acknowledge Configuration<br>0 Interrupts are acknowledged by writing '1' to the appropriate bit in the Interrupt Status Register<br>1 Interrupts are cleared when the Interrupt Status Register is read | R/W | 0 |

Table 5-6 : BAR 0 – Interrupt Control Register

### 5.1.7 Interrupt Status Register (0x090)

**Interrupt Status functionality has to be enabled in each of the Channel Control Registers, before it can be used.**

The Interrupt Status Register is divided in half:

The 16 upper bits indicate that a conversion is done, which is the most used status change during regular use. These bits are cleared in the way that is set up in the Interrupt Control Register.

The 16 lower bits are in fact a logical OR connection of the *Interrupt Status Configuration Done Register* and the *Interrupt Status Error Register*. If any of the bits in these two registers are set, the corresponding interrupt status bits are also set. So, for the 16 lower bits, after reading the Interrupt Status Register, the user has to check the other two registers in order to get the actual interrupt source. The 16 lower bits are not cleared by an access to the Interrupt Status Register. They are cleared by accessing the *Interrupt Status Configuration Done* and/or the *Interrupt Status Error Register*, each in the way that is set up in the Interrupt Control Register.

**A PCI Express Interrupt (Legacy INTA) is asserted when at least one bit is set in the Interrupt Status Register.**

| Bit | Symbol | Description | Access | Reset Value |
|---|---|---|---|---|
| 31 | CH_15_CONV_DONE | Channel X Conversion Done<br><br>Indicates that a conversion is done.<br>These bits are cleared with the method set up in the Interrupt Control Register. | r/c | 0 |
| 30 | CH_14_CONV_DONE | | r/c | 0 |
| 29 | CH_13_CONV_DONE | | r/c | 0 |
| 28 | CH_12_CONV_DONE | | r/c | 0 |
| 27 | CH_11_CONV_DONE | | r/c | 0 |
| 26 | CH_10_CONV_DONE | | r/c | 0 |
| 25 | CH_09_CONV_DONE | | r/c | 0 |
| 24 | CH_08_CONV_DONE | | r/c | 0 |
| 23 | CH_07_CONV_DONE | | r/c | 0 |
| 22 | CH_06_CONV_DONE | | r/c | 0 |
| 21 | CH_05_CONV_DONE | | r/c | 0 |
| 20 | CH_04_CONV_DONE | | r/c | 0 |
| 19 | CH_03_CONV_DONE | | r/c | 0 |
| 18 | CH_02_CONV_DONE | | r/c | 0 |
| 17 | CH_01_CONV_DONE | | r/c | 0 |
| 16 | CH_00_CONV_DONE | | r/c | 0 |
| 15 | CH_15_IRQ_STATUS | Channel X Interrupt Status<br><br>Indicates that at least one of several possible status changes happened. See the *Interrupt Status Configuration Done Register* and the *Interrupt Status Error Register* for possible status changes.<br>These bits are not cleared via this register. | r | 0 |
| 14 | CH_14_IRQ_STATUS | | r | 0 |
| 13 | CH_13_IRQ_STATUS | | r | 0 |
| 12 | CH_12_IRQ_STATUS | | r | 0 |
| 11 | CH_11_IRQ_STATUS | | r | 0 |
| 10 | CH_10_IRQ_STATUS | | r | 0 |
| 9 | CH_09_IRQ_STATUS | | r | 0 |
| 8 | CH_08_IRQ_STATUS | | r | 0 |

| | | | | |
|---|---|---|---|---|
| 7 | CH_07_IRQ_STATUS | | r | 0 |
| 6 | CH_06_IRQ_STATUS | | r | 0 |
| 5 | CH_05_IRQ_STATUS | | r | 0 |
| 4 | CH_04_IRQ_STATUS | | r | 0 |
| 3 | CH_03_IRQ_STATUS | | r | 0 |
| 2 | CH_02_IRQ_STATUS | | r | 0 |
| 1 | CH_01_IRQ_STATUS | | r | 0 |
| 0 | CH_00_IRQ_STATUS | | r | 0 |

Table 5-7 : BAR 0 – Interrupt Status Register

### 5.1.8 Interrupt Status Configuration Done Register (0x094)

This register is one of the sources for the 16 lower bits of the Interrupt Status Register. It has to be cleared in the way that is set up in the Interrupt Control Register.

| Bit | Symbol | Description | Access | Reset Value |
|---|---|---|---|---|
| 31:16 | - | Reserved - always write as '0', reads are undefined | - | - |
| 15 | CH_15_CONF_DONE | Channel X Configuration Done | r/c | 0 |
| 14 | CH_14_CONF_DONE | | r/c | 0 |
| 13 | CH_13_CONF_DONE | | r/c | 0 |
| 12 | CH_12_CONF_DONE | | r/c | 0 |
| 11 | CH_11_CONF_DONE | | r/c | 0 |
| 10 | CH_10_CONF_DONE | | r/c | 0 |
| 9 | CH_09_CONF_DONE | | r/c | 0 |
| 8 | CH_08_CONF_DONE | | r/c | 0 |
| 7 | CH_07_CONF_DONE | | r/c | 0 |
| 6 | CH_06_CONF_DONE | | r/c | 0 |
| 5 | CH_05_CONF_DONE | | r/c | 0 |
| 4 | CH_04_CONF_DONE | | r/c | 0 |
| 3 | CH_03_CONF_DONE | | r/c | 0 |
| 2 | CH_02_CONF_DONE | | r/c | 0 |
| 1 | CH_01_CONF_DONE | | r/c | 0 |
| 0 | CH_00_CONF_DONE | | r/c | 0 |

Table 5-8 : BAR 0 – Interrupt Status Configuration Done Register

### 5.1.9 Interrupt Status Error Register (0x098)

This register is one of the sources for the 16 lower bits of the Interrupt Status Register. It has to be cleared in the way that is set up in the Interrupt Control Register.

| Bit | Symbol | Description | Access | Reset Value |
|---|---|---|---|---|
| 31 | CH_15_CHAN_ERR | Channel X Channel Error<br><br>Possible causes:<br>- Timeout during the ADC measurement<br>- Timeout during the ADC self-calibration<br>- SPI communication problem | r/c | 0 |
| 30 | CH_14_CHAN_ERR | | r/c | 0 |
| 29 | CH_13_CHAN_ERR | | r/c | 0 |
| 28 | CH_12_CHAN_ERR | | r/c | 0 |
| 27 | CH_11_CHAN_ERR | | r/c | 0 |
| 26 | CH_10_CHAN_ERR | | r/c | 0 |
| 25 | CH_09_CHAN_ERR | | r/c | 0 |
| 24 | CH_08_CHAN_ERR | | r/c | 0 |
| 23 | CH_07_CHAN_ERR | | r/c | 0 |
| 22 | CH_06_CHAN_ERR | | r/c | 0 |
| 21 | CH_05_CHAN_ERR | | r/c | 0 |
| 20 | CH_04_CHAN_ERR | | r/c | 0 |
| 19 | CH_03_CHAN_ERR | | r/c | 0 |
| 18 | CH_02_CHAN_ERR | | r/c | 0 |
| 17 | CH_01_CHAN_ERR | | r/c | 0 |
| 16 | CH_00_CHAN_ERR | | r/c | 0 |
| 15 | CH_15_CONF_ERR | Channel X Configuration Error<br><br>Possible causes:<br>- Invalid table header content (sensor type, CRC, …)<br>- Invalid table data content (wrong number of data pairs)<br>- Measured value out of table | r/c | 0 |
| 14 | CH_14_CONF_ERR | | r/c | 0 |
| 13 | CH_13_CONF_ERR | | r/c | 0 |
| 12 | CH_12_CONF_ERR | | r/c | 0 |
| 11 | CH_11_CONF_ERR | | r/c | 0 |
| 10 | CH_10_CONF_ERR | | r/c | 0 |
| 9 | CH_09_CONF_ERR | | r/c | 0 |
| 8 | CH_08_CONF_ERR | | r/c | 0 |
| 7 | CH_07_CONF_ERR | | r/c | 0 |
| 6 | CH_06_CONF_ERR | | r/c | 0 |
| 5 | CH_05_CONF_ERR | | r/c | 0 |
| 4 | CH_04_CONF_ERR | | r/c | 0 |
| 3 | CH_03_CONF_ERR | | r/c | 0 |
| 2 | CH_02_CONF_ERR | | r/c | 0 |
| 1 | CH_01_CONF_ERR | | r/c | 0 |
| 0 | CH_00_CONF_ERR | | r/c | 0 |

Table 5-9 : BAR 0 – Interrupt Status Error Register

## 5.1.10 Cold Junction I2C Onboard Control Register (0x09C)

This register controls the onboard temperature sensor (SE95 type). Whenever data is fetched from the sensor, it will be saved into the Cold Junction I2C Onboard Data Register. Therefore, the data register always contains the latest data from the sensor (last update is dependent on the setting here).

<table>
<tr><th>Bit</th><th>Symbol</th><th>Description</th><th>Access</th><th>Reset Value</th></tr>
<tr><td>31:19</td><td>-</td><td>Reserved - always write as ‘0’, reads are undefined</td><td>-</td><td>-</td></tr>
<tr><td>18</td><td>CJ_ONBOARD_EN</td><td>Cold Junction Measurement Timer Enable<br>1 enabled<br>0 disabled</td><td>r/w</td><td>0</td></tr>
<tr><td>17:16</td><td>CJ_ONBOARD_BASE</td><td>Cold Junction Measurement Timer Base<br><table><tr><th>CC_BASE</th><th>Sample Clock Time Base</th></tr><tr><td>00</td><td>100 µs</td></tr><tr><td>01</td><td>100 ms</td></tr><tr><td>10</td><td>1 s</td></tr><tr><td>11</td><td>reserved</td></tr></table></td><td>r/w</td><td>00</td></tr>
<tr><td>15:00</td><td>CJ_ONBOARD_VALUE</td><td>Cold Junction Measurement Timer Value<br>These 16 bits set the starting value for the sample clock timer. The timer runs with the time base set by CJ BASE.<br>Whenever the timer hits zero, a conversion is started.</td><td>r/w</td><td>0x0000</td></tr>
</table>

Table 5-10: BAR 0 – Cold Junction I2C Onboard Control Register

### 5.1.11 Cold Junction I2C External Control Register (0x0A0)

This register controls the external temperature sensor (SE95 type). Whenever data is fetched from the sensor, it will be saved into the Cold Junction I2C External Data Register. Therefore, the data register always contains the latest data from the sensor (last update is dependent on the setting here).

<table>
<tr><th>Bit</th><th>Symbol</th><th>Description</th><th>Access</th><th>Reset Value</th></tr>
<tr><td>31:19</td><td>-</td><td>Reserved - always write as ‘0’, reads are undefined</td><td>-</td><td>-</td></tr>
<tr><td>18</td><td>CJ_EXTERNAL_EN</td><td>Cold Junction Measurement Timer Enable<br>1 enabled<br>0 disabled</td><td>r/w</td><td>0</td></tr>
<tr><td>17:16</td><td>CJ_EXTERNAL_BASE</td><td>Cold Junction Measurement Timer Base<br><table><tr><th>CC_BASE</th><th>Sample Clock Time Base</th></tr><tr><td>00</td><td>100 µs</td></tr><tr><td>01</td><td>100 ms</td></tr><tr><td>10</td><td>1 s</td></tr><tr><td>11</td><td>reserved</td></tr></table></td><td>r/w</td><td>00</td></tr>
<tr><td>15:00</td><td>CJ_EXTERNAL_VALUE</td><td>Cold Junction Measurement Timer Value<br>These 16 bits set the starting value for the sample clock timer. The timer runs with the time base set by CJ BASE.<br>Whenever the timer hits zero, a conversion is started.</td><td>r/w</td><td>0x0000</td></tr>
</table>

Table 5-11: BAR 0 – Cold Junction I2C External Control Register

### 5.1.12 Cold Junction Channel x Control Register (0x0A4 – 0x0E0)

These registers are used to set up the cold junction measurement option for each channel that is configured as a thermocouple channel. There are three general options available:

- A second channel configured as resistive sensor
- An onboard I2C (SE95) temperature sensor
- An external I2C (SE95) temperature sensor (not included)

All channels have a cold junction data register internally that is updated from one of the sources set up in the CJ_OPTION bits of this register.

Depending on the timer configuration of the Channel x Control Register (if a second channel is used for cold junction measurement), the *Cold Junction I2C Onboard Control Register* (if the onboard temperature sensor is used for cold junction measurement) or the *Cold Junction I2C External Control Register* (if the external temperature sensor is used for cold junction measurement), the cold junction value gets updated automatically. If the second channel is configured to conversion by trigger, the user has to take care that a trigger of the cold junction channel is issued.

| Bit | Symbol | Description | Access | Reset Value |
|---|---|---|---|---|
| 31:8 | - | Reserved - always write as '0', reads are undefined | - | - |
| 7:0 | CJ_OPTION | Cold Junction Measurement Source<br>0001 1111 Use analog channel 15<br>0001 1110 Use analog channel 14<br>0001 1101 Use analog channel 13<br>0001 1100 Use analog channel 12<br>0001 1011 Use analog channel 11<br>0001 1010 Use analog channel 10<br>0001 1001 Use analog channel 9<br>0001 1000 Use analog channel 8<br>0001 0111 Use analog channel 7<br>0001 0110 Use analog channel 6<br>0001 0101 Use analog channel 5<br>0001 0100 Use analog channel 4<br>0001 0011 Use analog channel 3<br>0001 0010 Use analog channel 2<br>0001 0001 Use analog channel 1<br>0001 0000 Use analog channel 0<br><br>0000 0010 Set the cold junction temperature value to "0" (useful for relational measurements)<br>0000 0001 I2C Temperature Sensor (external)<br>0000 0000 I2C Temperature Sensor (onboard) | r/w | 0 |

Table 5-12: BAR 0 – Cold Junction Channel x Control Register

### 5.1.13 Cold Junction I2C Onboard Data Register (0x0E4)

This register holds the last measured value of the onboard temperature sensor for reference.

| Bit | Symbol | Description | Access | Reset Value |
|---|---|---|---|---|
| 31:0 | CJ_DATA | Measured data of the onboard SE95 I2C cold junction temperature sensor<br><br>The measured data value of the SE95 temperature sensor is stored as 13-bit two's complement data giving a temperature resolution of 0.03125 °C.<br><br>To actually calculate the temperature from the data value, the following formula has to be used:<br>Temp (°C) = Data × 0.03125°C | r | 0 |

Table 5-13: BAR 0 – Cold Junction I2C Onboard Data Register

### 5.1.14 Cold Junction I2C External Data Register (0x0E8)

This register holds the last measured value of the external temperature sensor for reference.

| Bit | Symbol | Description | Access | Reset Value |
|---|---|---|---|---|
| 31:0 | CJ_DATA | Measured data of the onboard SE95 I2C cold junction temperature sensor<br><br>The measured data value of the SE95 temperature sensor is stored as 13-bit two's complement data giving a temperature resolution of 0.03125 °C.<br><br>To actually calculate the temperature from the data value, the following formula has to be used:<br>Temp (°C) = Data × 0.03125°C | r | 0 |

Table 5-14: BAR 0 – Cold Junction I2C External Data Register

### 5.1.15 Table Control Register (0x0EC)

The table in PCI BAR 1 is mapped to one of the tables in the onboard flash memory. Only one table can be read or written at a time; set up in this register. There are up to three steps necessary to handle the tables:

(1) If a table shall be fetched from the onboard flash memory, it is necessary to write the table number into bits 7:0 of this register. As soon as this happens, the BRAM_LOADED bit will be automatically reset to '0', and when the table is completely loaded from the flash, the BRAM_LOADED bit will be automatically set to '1'. Afterwards, the selected table can be read in PCI BAR 1.
(2) If the loaded table shall be overwritten (only supported for table numbers 15 to 31), the BRAM_WE bit has to be set to '1'. After setting this bit, the table may be overwritten by writing to PCI BAR 1. However, after this step, the table is only overwritten in the FPGA, not in the flash memory.
(3) For storing the modified table in the non-volatile flash memory, the FLASH_WRITE bit has to be set to '1'. When the transfer to the memory is finished, this bit will reset automatically.

Tables that are saved into flash memory are persistently saved through boot cycles and can be used by all channels. Tables in the Block RAM are volatile and cannot be used by the channels. So, in order to make a table permanently accessible to the channels, all three steps have to be executed for all user defined tables, but only once.

| Bit | Symbol | Description | Access | Reset Value |
|---|---|---|---|---|
| 31:24 | FACT_PROT | Factory Table Protection<br>Reserved – always write as '0', reads are undefined | r/w | 0 |
| 23:11 | - | Reserved - always write as '0', reads are undefined | - | - |
| 10 | FLASH_WRITE | Write BRAM table into onboard flash memory<br>1 Write operation of current table to flash memory in progress<br>0 Write operation finished<br><br>Bit will reset automatically when the table is transferred into the onboard flash memory. As long as this bit is '1', no other operations are permitted on the BRAM. | r/s | 0 |
| 9 | BRAM_WE | BRAM Write Enable<br>1 Table in BRAM can be overwritten<br>0 Table in BRAM is write protected | r/w | 0 |
| 8 | BRAM_LOADED | Table loaded from onboard flash memory into BRAM<br>1 Table in BRAM can be read<br>0 Table in BRAM is not ready to be read | r | 0 |
| 7:0 | TAB_NR | Table Number<br><br>The table number that shall be read or written<br>00 – 15 Factory supplied (read-only)<br>15 – 31 Custom programmable<br>31 – 255 Reserved | r/w | 0x00 |

Table 5-15: BAR 0 – Table Control Register

## 5.1.16 FPGA Firmware Revision Register (0x0F0)

| Bit | Symbol | Description | Access |
|---|---|---|---|
| 31:24 | MAJOR | Major Version Number | r |
| 23:16 | MINOR | Minor Version Number | r |
| 15:8 | REVISION | Revision Number | r |
| 7:0 | BUILD_CNT | Build Count Number | r |

Table 5-16: BAR 0 – Firmware Revision Register

## 5.1.17 ADUC7061 Channel x Firmware Revision Registers (0x0F4 – 0x130)

| Bit | Symbol | Description | Access |
|---|---|---|---|
| 31:24 | MAJOR | Major Version Number | r |
| 23:16 | MINOR | Minor Version Number | r |
| 15:8 | REVISION | Revision Number | r |
| 7:0 | BUILD_CNT | Build Count Number | r |

Table 5-17: BAR 0 – Channel Firmware Revision Register

# 5.2 BAR 1 – Table Space

## 5.2.1 Table Header Area

| Table Header Area | | |
|---|---|---|
| **Address** | **Data** | **Width** |
| 0x0000_0000 | **Sensor Type**<br>0 off<br>1 Thermocouple<br>2 RTD<br>3 Strain Gauge<br>FF Internal Self-Test | 8-Bit unsigned |
| 0x0000_0001 | **Table ID**<br>Custom defined value | 8-Bit unsigned |
| 0x0000_0002 | **Table Length**<br>Number of value pairs, without header (→ 0 for Strain Gauges) | 16-Bit unsigned |
| 0x0000_0004 | **CRC** | 16-Bit unsigned |
| 0x0000_0006 | **Calculation Factor**<br>Table Value == Measured Value * $10^{\text{Calculation Factor}}$ | 8-Bit unsigned |
| 0x0000_0007 | **Decimal Places of the calculated data**<br>Result Value == Measured Temperature * $10^{\text{Decimal Places}}$ | 8-Bit unsigned |
| 0x0000_0008 | **Temperature Unit** (for Thermocouple and RTD only)<br>0 Degrees in Celsius<br>1 Kelvin<br>2 Degrees in Fahrenheit | 8-Bit unsigned |
| 0x0000_0009 | **Bridge Factor** (for Strain Gauges only)<br>1 Quarter Bridge<br>2 Half Bridge<br>4 Full Bridge | 8-Bit unsigned |
| 0x0000_000A | **Nominal Resistance in Ω** (for RTD and Strain Gauges only) | 32-Bit unsigned |
| 0x0000_000E | **Gauge Factor** (for Strain Gauges only)<br>Specified in 0.01 steps, e.g. “194” for Gauge Factor 1.94 | 16-Bit unsigned |

Table 5-18: BAR 1 – Table Header Area

### 5.2.2 Table Data Area

| Table Data Area | | |
|---|---|---|
| **Address** | **Data** | |
| 0x0000_0010 | #1 Measured Value | 32-Bit signed |
| 0x0000_0014 | #1 Corresponding Temperature | 16-Bit signed |
| 0x0000_0016 | #2 Measured Value | 32-Bit signed |
| 0x0000_001A | #2 Corresponding Temperature | 16-Bit signed |
| … | … Measured Value | 32-Bit signed |
| … | … Corresponding Temperature | 16-Bit signed |

Table 5-19: BAR 1 – Table Data Area

**The data in the data area has to be stored sorted by the measured value in ascending order. The values have to be strictly monotonic increasing!**

## 5.3 BAR 2 – In-Field Programming Space

The In-Field Programming Space is described in a separate document that is dedicated to updating the ADUC7061 and the FPGA. It is not part of this User Manual.

# 6 Interrupts

## 6.1 Interrupt Sources

The PCIe Interface of the TXMC590 provides one interrupt; the legacy INTa.

Internally, there are several sources that may trigger this interrupt:

- **Channel x Conversion Done Interrupt (16x)**

  Event-based interrupts that become active, when one of the Channel x Conversion Done bits in the *Interrupt Status Register* changes from ‘0’ to ‘1’. This happens, when a channel has finished a conversion and the data can be read from its Channel X Data Register.

  The Conversion Done events can be enabled/disabled per channel.

- **Channel x Status Interrupts (48x)**

  Event-based interrupts that become active, when one of the Channel x Status bits in the *Interrupt Status Register* changes from ‘0’ to ‘1’. This happens either when a channel has finished configuration or when one of the channels signals errors. Errors are further differentiated between configuration errors and conversion errors.

  The events are indicated in the *Interrupt Status Configuration Done Register* and in the *Interrupt Status Error Register*. See the description of these registers and especially the description of the *Interrupt Status Register* for more details.

  These Status Interrupt events can be enabled/disabled per channel.

## 6.2 Interrupt Handling

There are two ways of acknowledging interrupts:

1. Acknowledge selected interrupts by writing a ‘1’ to the appropriate interrupt status bit. This method is used when the IRQ_ACK_CONF bit is set to ‘0’ (default).
2. Acknowledge all interrupts of one type just by reading the appropriate register. This method is used when the IRQ_ACK_CONF bit is set to ‘1’.

See the *Interrupt Control Register* and the *Interrupt Status Register* description for more details.

# 7 Functional Description

## 7.1 Table Concept

The TXMC590 measures voltages from thermocouples, from RTDs or from strain gauges. These voltages have to be translated into temperatures and strain in some way, depending on the specific sensor type. The TXMC590 uses tables to translate from a measured value to a corresponding temperature or strain value. Each channel is configured through an individual table, but tables may also be used for more than one channel.

These tables are divided into a header area and a data area.

The header contains metadata like the sensor type, table length, calculation factor for the measurement values and other important information about the connected sensor. Strain gauges, for example, need to have a gauge factor included, a nominal resistance and a maximum strain. RTDs on the other hand only need the nominal resistance, while thermocouples don't need any of this information. The layout and the content of the header area is described in chapter 5.2.1 Table Header Area.

The data area consists of up to 680 pairs of measured value and corresponding sensor type data value. The measured value is a 32-bit signed value while the corresponding data value is 16-bit signed.

**The returned values are interpolated between the two nearest values present in the table data area (i.e. for thermocouples and RTDs).**

The TXMC590 comes with 16 factory tables that cover the most common thermocouple and RTD types. These tables are meant as a starting point to get up and running fast. They are not necessarily suitable for each and every use case. Different sensor types and different environments may require different tables. In these cases the custom tables should be used. The content of the factory tables is described in the following sensor type chapters.

### 7.1.1 Program a new table

The steps that are necessary in order to program a new table into the TXMC590 are described in the chapter 5.1.15 Table Control Register (0x0EC).

The creation of a new table from scratch is a tedious task that is greatly simplified when the TEWS Table Builder is used, which is delivered together with TXMC590-SW-xx.

# 7.2 Thermocouples

## 7.2.1 Wiring Scheme / Sensor Installation

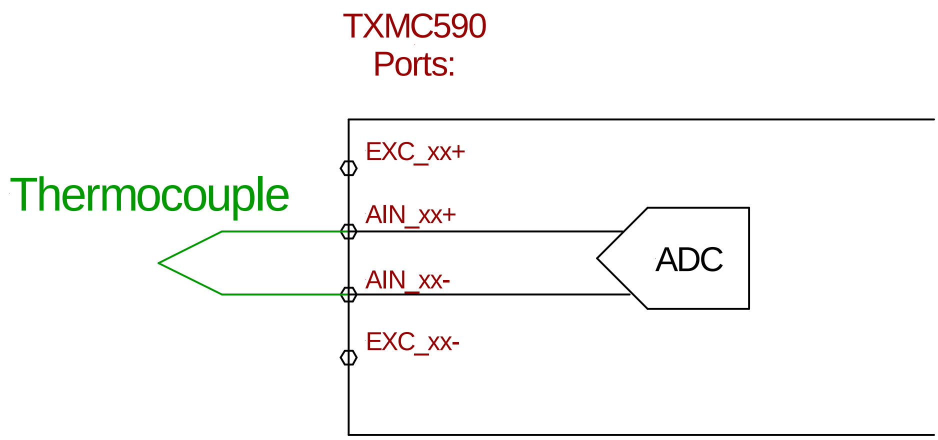



Figure 7-1 : Wiring scheme - Thermocouples

Thermocouples are wired directly to the analog inputs of the ADUC7061. There is no excitation needed, these pins may be left unconnected.

**The analog inputs of the TXMC590 are non-isolated and in total, the TXMC590 offers 16 thermocouple channels that may be distributed to several different locations. This might cause severe ground loops and couple AC noise into the measurement when using grounded thermocouples. Therefore, the TXMC590 may only be used with ungrounded thermocouples.**

## 7.2.2 Cold Junction Compensation

To interpret the measurement of a thermocouple correctly, the temperature at the point where the thermocouple cabling material connects to a different cabling material (e.g. copper) has to be known. This point is called the “cold junction compensation point”, and the temperature at this point is called “cold junction temperature”. This can be achieved in different ways, which are all be supported by the TXMC590:

- The use of a second channel of the TXMC590; configured as an RTD for example. (➔ good accuracy possible, but this solution occupies one additional channel of the TXMC590 that cannot be used for other purposes)
- Onboard temperature sensor (➔ the TXMC590 has probably a different temperature than the point where the thermocouple is connected to copper which means that this is not the best choice for good accuracy)
- The Front-I/O I2C input for an external I2C temperature sensor. This temperature sensor could be mounted on a terminal block where the thermocouples are connected. This way, the cold junction temperature can be measured accurately in the necessary spot without occupying an additional channel.

The cold junction temperature measurement method is set up individually for each channel in the corresponding *Cold Junction Channel x Control Register.* If one of the other channels is used for cold junction measurement, this channel has to be separately set up in its *Channel x Control Register* (e.g. sensor type, conversion mode, etc.). If one of the two I2C sensors is used for cold junction temperature measurement, the device has to be set up either in the *Cold Junction I2C Onboard Control Register* or in the

*Cold Junction I2C External Control Register*. As these sensors may be used for more than one channel, changing the settings affects all channels using them. The value of both sensors may be read back over the PCIe interface in the *Cold Junction I2C Onboard/External Data Registers*.

All channels that are configured for thermocouple sensors automatically fetch the cold junction value prior to each measurement.

**For thermocouple use, it is important to note that the lowest temperature to be returned by the TXMC590 is determined by the lowest value in the table plus the cold junction temperature. This is because the voltage difference measured on the thermocouple has no direct meaning. It has to be combined with the cold junction temperature to get an actual temperature reading.**

**So, if the lowest value in the table is, for example, 0° == 0mV (value), then the minimum temperature to be returned is 0° + Cold Junction Temperature. The actual voltage difference measured on the thermocouple might be negative but because the table does not contain these negative values, it returns the minimum value, which is 0mV.**

## 7.2.3 Factory Tables

### 7.2.3.1 Type K

| Header Area | | |
|---|---|---|
| **Address** | **Data** | **Value** |
| 0x0000_0000 | Sensor Type | 1 (Thermocouple) |
| 0x0000_0001 | Table ID | 0 |
| 0x0000_0002 | Table Length | 0x0291 (657) |
| 0x0000_0004 | CRC | 0xE82A |
| 0x0000_0006 | Calculation Factor | 6 |
| 0x0000_0007 | Decimal Places of the calculated data | 2 |
| 0x0000_0008 | Temperature Unit | 0 (°C) |
| 0x0000_0009 | Bridge Factor | - |
| 0x0000_000A | Nominal Resistance in Ω | - |
| 0x0000_000E | Gauge Factor | - |

| Data Area | |
|---|---|
| **Temperature [°C]** | **Measured Value (Scaled with Calculation Factor)** |
| -270 °C | -6458 (i.e. -6.458mV) |
| -269 °C | -6457 |
| -266 °C | -6453 |
| Temperature steps of ±3 °C from -263°C to +1363°C. | |
| +1366 °C | 54683 |
| +1369 °C | 54785 |
| +1370 °C | 54819 |

Table 7-1 : Type K

### 7.2.3.2 Type J

| Header Area | | |
|---|---|---|
| **Address** | **Data** | **Value** |
| 0x0000_0000 | Sensor Type | 1 (Thermocouple) |
| 0x0000_0001 | Table ID | 1 |
| 0x0000_0002 | Table Length | 0x02A8 (680) |
| 0x0000_0004 | CRC | 0xBAB9 |
| 0x0000_0006 | Calculation Factor | 6 |
| 0x0000_0007 | Decimal Places of the calculated data | 2 |
| 0x0000_0008 | Temperature Unit | 0 (°C) |
| 0x0000_0009 | Bridge Factor | - |
| 0x0000_000A | Nominal Resistance in Ω | - |
| 0x0000_000E | Gauge Factor | - |

| Data Area | |
|---|---|
| **Temperature [°C]** | **Measured Value (Scaled with Calculation Factor)** |
| 0 °C | 0 (i.e. 0 µV) |
| 1 °C | 50 |
| 2 °C | 101 |
| Temperature steps of ±1 °C from 3°C to +676°C. | |
| +677 °C | 37712 |
| +678 °C | 37773 |
| +679 °C | 37835 |

Table 7-2 : Type J

### 7.2.3.3 Type N

| Header Area | | |
|---|---|---|
| **Address** | **Data** | **Value** |
| 0x0000_0000 | Sensor Type | 1 (Thermocouple) |
| 0x0000_0001 | Table ID | 2 |
| 0x0000_0002 | Table Length | 0x0274 (628) |
| 0x0000_0004 | CRC | 0x39FD |
| 0x0000_0006 | Calculation Factor | 6 |
| 0x0000_0007 | Decimal Places of the calculated data | 2 |
| 0x0000_0008 | Temperature Unit | 0 (°C) |
| 0x0000_0009 | Bridge Factor | - |
| 0x0000_000A | Nominal Resistance in Ω | - |
| 0x0000_000E | Gauge Factor | - |
| **Data Area** | | |
| **Temperature [°C]** | **Measured Value (Scaled with Calculation Factor)** | |
| -269 °C | -4345 | |
| -266 °C | -4343 | |
| -263 °C | -4340 | |
| Temperature steps of ±3 °C from -260°C to +1293°C. | | |
| +1296 °C | 47369 | |
| +1299 °C | 47477 | |
| +1300 °C | 47513 | |

Table 7-3 : Type K

### 7.2.3.4 Type E

| Header Area | | |
|---|---|---|
| **Address** | **Data** | **Value** |
| 0x0000_0000 | Sensor Type | 1 (Thermocouple) |
| 0x0000_0001 | Table ID | 3 |
| 0x0000_0002 | Table Length | 0x027C (636) |
| 0x0000_0004 | CRC | 0xDC33 |
| 0x0000_0006 | Calculation Factor | 6 |
| 0x0000_0007 | Decimal Places of the calculated data | 2 |
| 0x0000_0008 | Temperature Unit | 0 (°C) |
| 0x0000_0009 | Bridge Factor | - |
| 0x0000_000A | Nominal Resistance in Ω | - |
| 0x0000_000E | Gauge Factor | - |

| Data Area | |
|---|---|
| **Temperature [°C]** | **Measured Value (Scaled with Calculation Factor)** |
| -270 °C | -9835 (9.835mV) |
| -268 °C | -9831 |
| -266 °C | -9824 |
| Temperature steps of ±2 °C from -264°C to +994°C. | |
| +996 °C | 76072 |
| +998 °C | 76223 |
| +1000 °C | 76373 |

Table 7-4 : Type E

### 7.2.3.5 Type T

| Header Area | | |
|---|---|---|
| **Address** | **Data** | **Value** |
| 0x0000_0000 | Sensor Type | 1 (Thermocouple) |
| 0x0000_0001 | Table ID | 4 |
| 0x0000_0002 | Table Length | 0x029E (670) |
| 0x0000_0004 | CRC | 0xDE33 |
| 0x0000_0006 | Calculation Factor | 6 |
| 0x0000_0007 | Decimal Places of the calculated data | 2 |
| 0x0000_0008 | Temperature Unit | 0 (°C) |
| 0x0000_0009 | Bridge Factor | - |
| 0x0000_000A | Nominal Resistance in Ω | - |
| 0x0000_000E | Gauge Factor | - |

| Data Area | |
|---|---|
| **Temperature [°C]** | **Measured Value (Scaled with Calculation Factor)** |
| -269 °C | -6256 (9.835mV) |
| -268 °C | -6255 |
| -267 °C | -6253 |
| Temperature steps of ±1 °C from -266°C to +397°C. | |
| +398 °C | 20748 |
| +399 °C | 20810 |
| +400 °C | 20748 |

Table 7-5 : Type T

### 7.2.3.6 Type R

| Header Area | | |
|---|---|---|
| **Address** | **Data** | **Value** |
| 0x0000_0000 | Sensor Type | 1 (Thermocouple) |
| 0x0000_0001 | Table ID | 5 |
| 0x0000_0002 | Table Length | 0x0227 (551) |
| 0x0000_0004 | CRC | 0x093E |
| 0x0000_0006 | Calculation Factor | 6 |
| 0x0000_0007 | Decimal Places of the calculated data | 2 |
| 0x0000_0008 | Temperature Unit | 0 (°C) |
| 0x0000_0009 | Bridge Factor | - |
| 0x0000_000A | Nominal Resistance in Ω | - |
| 0x0000_000E | Gauge Factor | - |

| Data Area | |
|---|---|
| **Temperature [°C]** | **Measured Value (Scaled with Calculation Factor)** |
| -50 °C | -226 (226.0µV) |
| -49 °C | -223 |
| -46 °C | -211 |
| Temperature steps of ±3 °C from -43°C to +1757°C. | |
| +1760 °C | 21003 |
| +1763 °C | 21040 |
| +1766 °C | 21077 |

Table 7-6 : Type R

### 7.2.3.7 Type S

| Header Area | | |
|---|---|---|
| **Address** | **Data** | **Value** |
| 0x0000_0000 | Sensor Type | 1 (Thermocouple) |
| 0x0000_0001 | Table ID | 6 |
| 0x0000_0002 | Table Length | 0x0227 (551) |
| 0x0000_0004 | CRC | 0xAB4E |
| 0x0000_0006 | Calculation Factor | 6 |
| 0x0000_0007 | Decimal Places of the calculated data | 2 |
| 0x0000_0008 | Temperature Unit | 0 (°C) |
| 0x0000_0009 | Bridge Factor | - |
| 0x0000_000A | Nominal Resistance in Ω | - |
| 0x0000_000E | Gauge Factor | - |

| Data Area | |
|---|---|
| **Temperature [°C]** | **Measured Value (Scaled with Calculation Factor)** |
| -50 °C | -236 (236.0µV) |
| -49 °C | -232 |
| -46 °C | -219 |
| Temperature steps of ±3 °C from -43°C to +1757°C. | |
| +1760 °C | 18609 |
| +1763 °C | 18640 |
| +1766 °C | 18672 |

Table 7-7 : Type S

### 7.2.3.8 Type B

| Header Area | | |
|---|---|---|
| **Address** | **Data** | **Value** |
| 0x0000_0000 | Sensor Type | 1 (Thermocouple) |
| 0x0000_0001 | Table ID | 7 |
| 0x0000_0002 | Table Length | 0x0214 (532) |
| 0x0000_0004 | CRC | 0xBC1C |
| 0x0000_0006 | Calculation Factor | 7 |
| 0x0000_0007 | Decimal Places of the calculated data | 2 |
| 0x0000_0008 | Temperature Unit | 0 (°C) |
| 0x0000_0009 | Bridge Factor | - |
| 0x0000_000A | Nominal Resistance in Ω | - |
| 0x0000_000E | Gauge Factor | - |
| **Data Area** | | |
| **Temperature [°C]** | **Measured Value (Scaled with Calculation Factor)** | |
| +50 °C | 20 (20.0µV) | |
| +53 °C | 30 | |
| +56 °C | 40 | |
| Temperature steps of ±3 °C from +59°C to +1810°C. | | |
| +1813 °C | 137400 | |
| +1816 °C | 137750 | |
| +1820 °C | 138200 | |

Table 7-8 : Type B

# 7.3 RTDs

## 7.3.1 Wiring Scheme / Sensor Installation

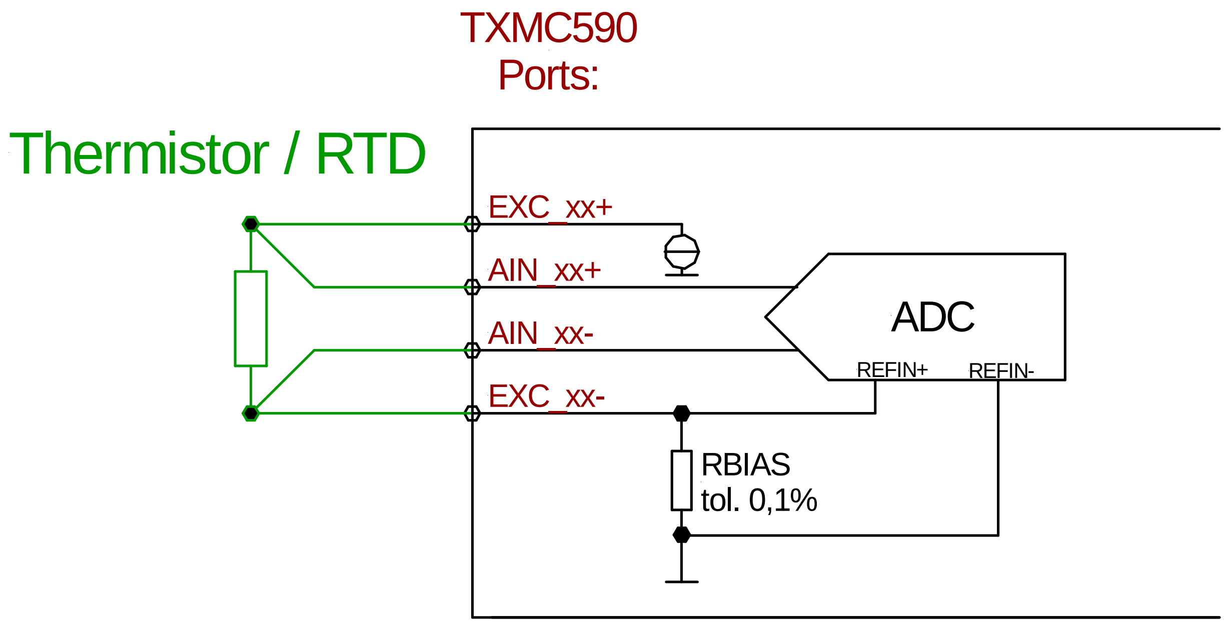



Figure 7-2 : Wiring scheme – RTDs / Thermistors

RTD/Thermistors are measured ratio metric. The excitation current flowing through the sensor is also flowing through the known high precision resistor RBIAS. The voltage over RBIAS is then used as the reference voltage for the measurement. As a result, fluctuations in the excitation are cancelled out. This wiring scheme is known as 4-wire installation.

The µC sets up the gain and the excitation current automatically according to the sensor table.

**It is important to note that $R_{BIAS}$ of the TXMC590 is 5.6kΩ. This limits the usable RTD sensors to those that do not exceed this value.**

### 7.3.2 Factory Tables

#### 7.3.2.1 PT100

| Header Area | | |
|---|---|---|
| **Address** | **Data** | **Value** |
| 0x0000_0000 | Sensor Type | 2 (RTD) |
| 0x0000_0001 | Table ID | 9 |
| 0x0000_0002 | Table Length | 0x0212 (530) |
| 0x0000_0004 | CRC | 0xC4F8 |
| 0x0000_0006 | Calculation Factor | 3 |
| 0x0000_0007 | Decimal Places of the calculated data | 2 |
| 0x0000_0008 | Temperature Unit | 0 (°C) |
| 0x0000_0009 | Bridge Factor | - |
| 0x0000_000A | Nominal Resistance in Ω | 100 |
| 0x0000_000E | Gauge Factor | - |

| Data Area | |
|---|---|
| **Temperature [°C]** | **Measured Value (Scaled with Calculation Factor)** |
| -200 °C | 18520 (18.52Ω) |
| -198 °C | 19384 |
| -196 °C | 20247 |
| Temperature steps of ±2 °C from -194°C to +852°C. | |
| +854 °C | 391651 |
| +856 °C | 392235 |
| +858 °C | 392819 |

Table 7-9 : PT100

### 7.3.2.2 PT500

| Header Area | | |
|---|---|---|
| **Address** | **Data** | **Value** |
| 0x0000_0000 | Sensor Type | 2 (RTD) |
| 0x0000_0001 | Table ID | 10 |
| 0x0000_0002 | Table Length | 0x0212 (530) |
| 0x0000_0004 | CRC | 0x7B15 |
| 0x0000_0006 | Calculation Factor | 3 |
| 0x0000_0007 | Decimal Places of the calculated data | 2 |
| 0x0000_0008 | Temperature Unit | 0 (°C) |
| 0x0000_0009 | Bridge Factor | - |
| 0x0000_000A | Nominal Resistance in Ω | 500 |
| 0x0000_000E | Gauge Factor | - |

| Data Area | |
|---|---|
| **Temperature [°C]** | **Measured Value (Scaled with Calculation Factor)** |
| -200 °C | 92600 (92.60Ω) |
| -198 °C | 96920 |
| -196 °C | 101235 |
| Temperature steps of ±2 °C from -194°C to +852°C. | |
| +854 °C | 1958255 |
| +856 °C | 1961175 |
| +858 °C | 1964095 |

Table 7-10: PT500

### 7.3.2.3 PT1000

| Header Area | | |
|---|---|---|
| **Address** | **Data** | **Value** |
| 0x0000_0000 | Sensor Type | 2 (RTD) |
| 0x0000_0001 | Table ID | 11 |
| 0x0000_0002 | Table Length | 0x020E (526) |
| 0x0000_0004 | CRC | 0xC2C7 |
| 0x0000_0006 | Calculation Factor | 3 |
| 0x0000_0007 | Decimal Places of the calculated data | 2 |
| 0x0000_0008 | Temperature Unit | 0 (°C) |
| 0x0000_0009 | Bridge Factor | - |
| 0x0000_000A | Nominal Resistance in Ω | 1000 |
| 0x0000_000E | Gauge Factor | - |

| Data Area | |
|---|---|
| **Temperature [°C]** | **Measured Value (Scaled with Calculation Factor)** |
| -200 °C | 185201 (185.201Ω) |
| -198 °C | 193840 |
| -196 °C | 202465 |
| Temperature steps of ±2 °C from -194°C to +852°C. | |
| +846 °C | 3893096 |
| +848 °C | 3898956 |
| +850 °C | 3904811 |

Table 7-11: PT1000

#### 7.3.2.4 NI100

| Header Area | | |
|---|---|---|
| **Address** | **Data** | **Value** |
| 0x0000_0000 | Sensor Type | 2 (RTD) |
| 0x0000_0001 | Table ID | 12 |
| 0x0000_0002 | Table Length | 0x0137 (311) |
| 0x0000_0004 | CRC | 0x2788 |
| 0x0000_0006 | Calculation Factor | 3 |
| 0x0000_0007 | Decimal Places of the calculated data | 2 |
| 0x0000_0008 | Temperature Unit | 0 (°C) |
| 0x0000_0009 | Bridge Factor | - |
| 0x0000_000A | Nominal Resistance in Ω | 100 |
| 0x0000_000E | Gauge Factor | - |

| Data Area | |
|---|---|
| **Temperature [°C]** | **Measured Value (Scaled with Calculation Factor)** |
| -60 °C | 69520 (69.52Ω) |
| -59 °C | 69987 |
| -58 °C | 70456 |
| Temperature steps of ±1 °C from -57°C to +247°C. | |
| +248 °C | 288004 |
| +249 °C | 289067 |
| +250 °C | 290133 |

Table 7-12: NI100

### 7.3.2.5 NI500

| Header Area | | |
|---|---|---|
| **Address** | **Data** | **Value** |
| 0x0000_0000 | Sensor Type | 2 (RTD) |
| 0x0000_0001 | Table ID | 13 |
| 0x0000_0002 | Table Length | 0x0137 (311) |
| 0x0000_0004 | CRC | 0x3C8A |
| 0x0000_0006 | Calculation Factor | 3 |
| 0x0000_0007 | Decimal Places of the calculated data | 2 |
| 0x0000_0008 | Temperature Unit | 0 (°C) |
| 0x0000_0009 | Bridge Factor | - |
| 0x0000_000A | Nominal Resistance in Ω | 500 |
| 0x0000_000E | Gauge Factor | - |

| Data Area | |
|---|---|
| **Temperature [°C]** | **Measured Value (Scaled with Calculation Factor)** |
| -60 °C | 347600 (347.60Ω) |
| -59 °C | 349935 |
| -58 °C | 352280 |
| Temperature steps of ±1 °C from -57°C to +247°C. | |
| +248 °C | 1440020 |
| +249 °C | 1445335 |
| +250 °C | 1450665 |

Table 7-13: NI500

### 7.3.2.6 NI1000

| Header Area | | |
|---|---|---|
| **Address** | **Data** | **Value** |
| 0x0000_0000 | Sensor Type | 2 (RTD) |
| 0x0000_0001 | Table ID | 14 |
| 0x0000_0002 | Table Length | 0x0137 (311) |
| 0x0000_0004 | CRC | 0xF050 |
| 0x0000_0006 | Calculation Factor | 3 |
| 0x0000_0007 | Decimal Places of the calculated data | 2 |
| 0x0000_0008 | Temperature Unit | 0 (°C) |
| 0x0000_0009 | Bridge Factor | - |
| 0x0000_000A | Nominal Resistance in Ω | 1000 |
| 0x0000_000E | Gauge Factor | - |

| Data Area | |
|---|---|
| **Temperature [°C]** | **Measured Value (Scaled with Calculation Factor)** |
| -60 °C | 695200 (695.20Ω) |
| -59 °C | 699870 |
| -58 °C | 704560 |
| Temperature steps of ±1 °C from -57°C to +247°C. | |
| +248 °C | 2880040 |
| +249 °C | 2890670 |
| +250 °C | 2901330 |

Table 7-14: NI1000

# 7.4 Strain Gauges

## 7.4.1 Wiring Scheme / Sensor Installation

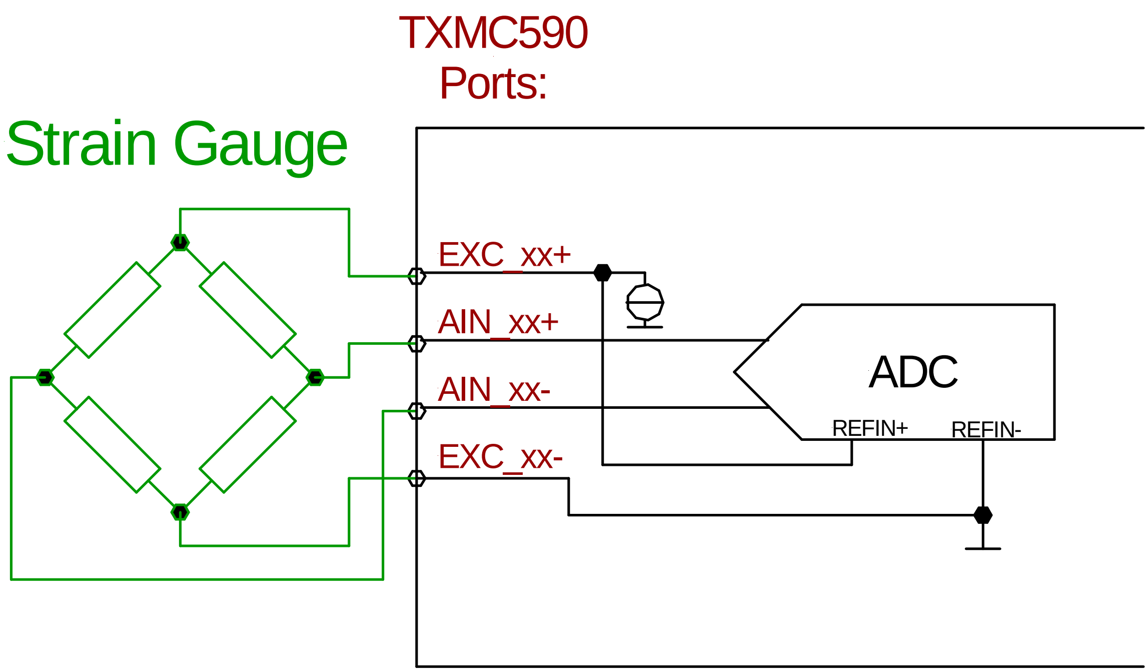



Figure 7-3 : Wiring Scheme / Sensor Installation

Strain gauges are measured best ratio metric; the voltage drop across the bridge is used as the reference voltage. The TXMC590 supports any type of strain gauge (in quarter, half and full bridge configuration), but it does not offer a bridge completion circuit. Therefore, if quarter or half bridges are used, these have to be completed outside of the module.

The TXMC590 excites Strain Gauges from its onboard current source, which is automatically set up according to the sensor table. The maximum excitation current of the TXMC590 is 2mA, so higher impedance sensors will get better reading results.

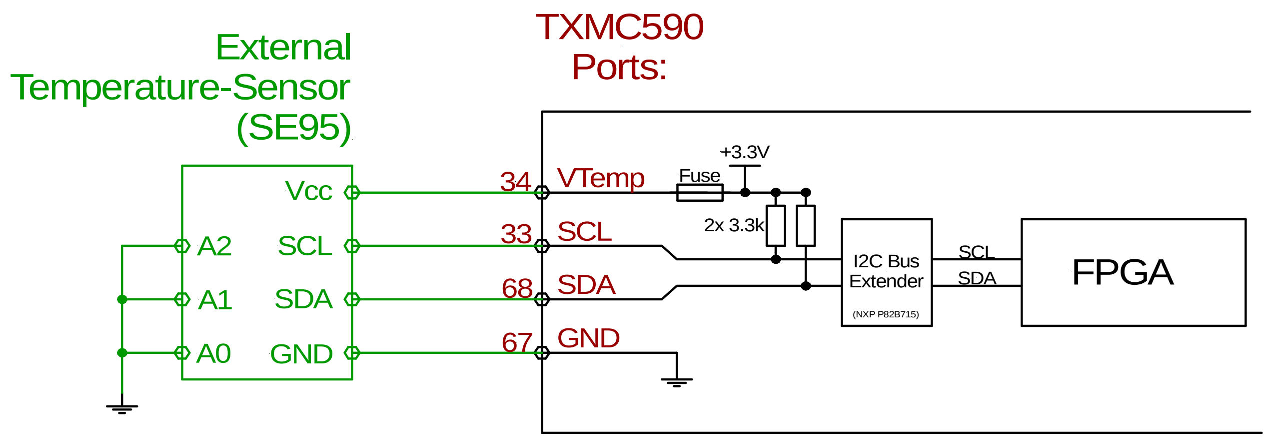
The optimal gain of the analog input amplifier is also set up automatically.

## 7.4.2 Factory Tables

There are no factory supplied tables for strain gauges, because of the various gauge factors common strain gauges have. It is very simple to generate a user-defined table because strain gauges do not need any data area. The calculation is done based on the gauge factor, maximum strain and the nominal resistance.

# 7.5 External I2C Cold Junction Temperature Sensor

## 7.5.1 Wiring Scheme / Sensor Installation




The Front-I/O pins 33, 34, 67 and 68 can be used to connect an external I2C temperature sensor. The TXMC590 supports SE95-type sensors or compatible ones.

The supply voltage for the sensor is 3.3V and is fused onboard.

SCL and SDA each have a 3.3kΩ pull-up resistor onboard.

Connect the three address pins (A2, A1, A0) of the SE95 to GND, because the TXMC590 expects the SE95 on the device address “1001000”.

## 7.6 Getting a temperature/strain value

Given that the correct table is already available in the onboard flash memory (see previous chapters for that task), the process of measuring a temperature or strain value is simple:

1. Set up the channel in the Channel Control Register without the CONV_MODE bit set
2. Issue a configuration cycle for that channel in the Configuration Trigger Register, wait for it to finish (bit is reset to ‘0’). Errors that happen during the configuration cycle are indicated in the Error Status Register and/or the Interrupt Status Error Register and have to be handled before continuation. Otherwise no valid data will be written to the Channel Data Register
3. *Thermocouples only: If a Thermocouple shall be read, also set up the cold junction measurement for that channel in the Cold Junction Channel X Control Register*
4. *Thermocouples only: Depending on the setting in Cold Junction Channel X Control Register, either set up the onboard or the external I2C temperature sensor, or set up another channel as described in 1.) and 2.).*
5. Start the conversion, either periodically in the Channel Control Register or manually in the Conversion Trigger Register
6. Check the Error Status Register and/or the Interrupt Status Error Register for errors. If errors are indicated they have to be cleared and another conversion cycle has to be issued. If no errors are indicated, read the measured value from the Channel Data Register

# 8 Pin Assignment – I/O Connector

## 8.1 X1 – Front-I/O Connector

| | | | | |
|---|---|---|---|---|
| **Connector Type** | M3 Mini D Ribbon Connector | | | |
| **Source & Order Info** | 3M 10268-52E2PC | | | |
| **Pin Assignment** | | | | |
| **Description** | **Pin** | **Connector View** | **Pin** | **Description** |
| EXC_00+ | **1** | | **35** | EXC_00- |
| AIN_00+ | **2** | | **36** | AIN_00- |
| EXC_01+ | **3** | | **37** | EXC_01- |
| AIN_01+ | **4** | | **38** | AIN_01- |
| EXC_02+ | **5** | | **39** | EXC_02- |
| AIN_02+ | **6** | | **40** | AIN_02- |
| EXC_03+ | **7** | | **41** | EXC_03- |
| AIN_03+ | **8** | | **42** | AIN_03- |
| EXC_04+ | **9** | | **43** | EXC_04- |
| AIN_04+ | **10** | | **44** | AIN_04- |
| EXC_05+ | **11** | | **45** | EXC_05- |
| AIN_05+ | **12** | | **46** | AIN_05- |
| EXC_06+ | **13** | | **47** | EXC_06- |
| AIN_06+ | **14** | | **48** | AIN_06- |
| EXC_07+ | **15** | | **49** | EXC_07- |
| AIN_07+ | **16** | | **50** | AIN_07- |
| EXC_08+ | **17** | | **51** | EXC_08- |
| AIN_08+ | **18** | | **52** | AIN_08- |
| EXC_09+ | **19** | | **53** | EXC_09- |
| AIN_09+ | **20** | | **54** | AIN_09- |
| EXC_10+ | **21** | | **55** | EXC_10- |
| AIN_10+ | **22** | | **56** | AIN_10- |
| EXC_11+ | **23** | | **57** | EXC_11- |
| AIN_11+ | **24** | | **58** | AIN_11- |
| EXC_12+ | **25** | | **59** | EXC_12- |
| AIN_12+ | **26** | | **60** | AIN_12- |
| EXC_13+ | **27** | | **61** | EXC_13- |
| AIN_13+ | **28** | | **62** | AIN_13- |
| EXC_14+ | **29** | | **63** | EXC_14- |
| AIN_14+ | **30** | | **64** | AIN_14- |
| EXC_15+ | **31** | | **65** | EXC_15- |
| AIN_15+ | **32** | | **66** | AIN_15- |
| SCL | **33** | | **67** | GND |
| $V_{TEMP}$ | **34** | | **68** | SDA |

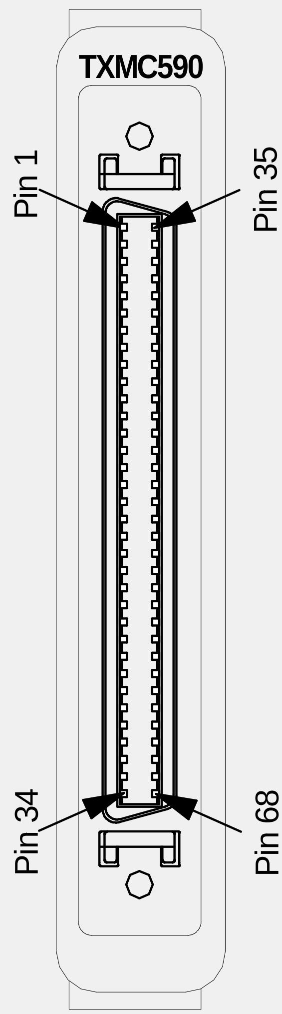



Table 8-1 : Pin Assignment Front I/O Connector

# 8.2 X2 – JTAG Connector ADUC7061

This connector is intended for factory use only and not mounted by default.

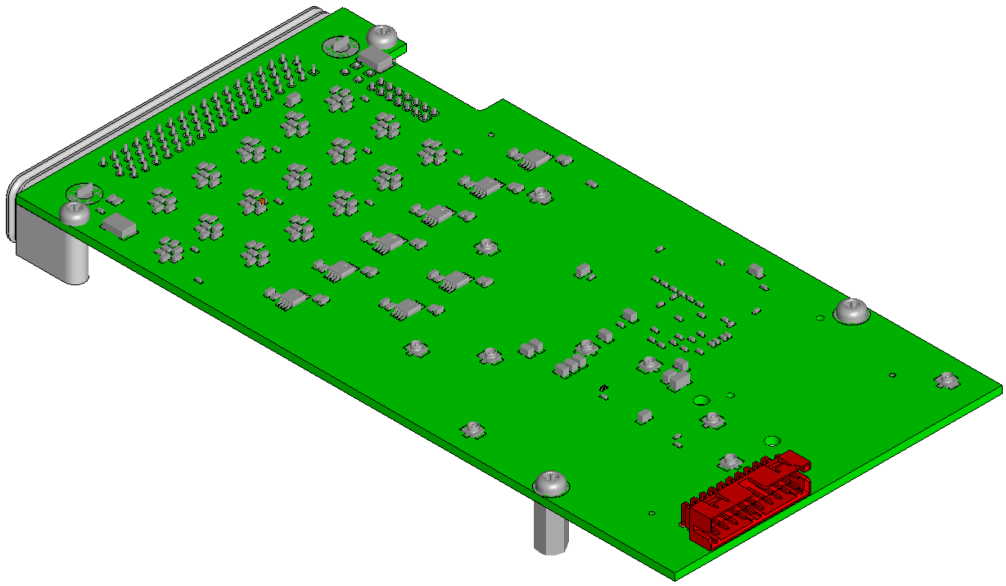


Figure 8-1 : X2 – JTAG Connector ADUC7061 (3D Model)

# 8.3 S1 – DIP Switch for JTAG Chain Selection

The 16x ADUC7061 are grouped by four into four JTAG chains to overcome limitations of several JTAG programmers.

This DIP switch selects the ADUC7061 JTAG chain group. It is, however, only intended for factory use and not necessary to be set up during normal operation.

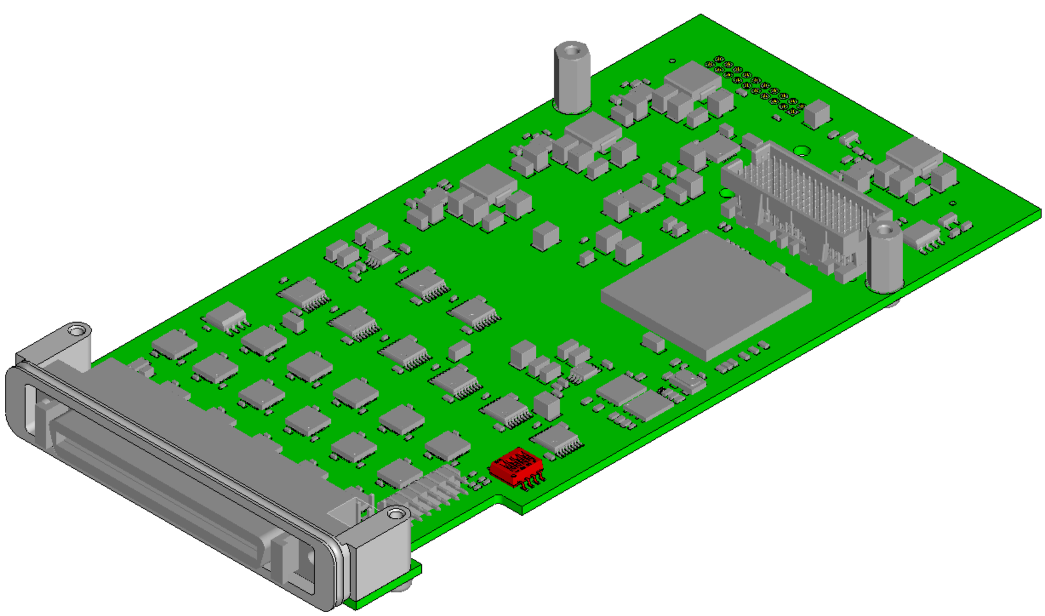


Figure 8-2 : S1 – DIP Switch for JTAG Chain Selection (3D Model)

## 8.4 X3 – JTAG Connector FPGA

This connector is connected to the JTAG chain of the FPGA and is intended for factory use only.

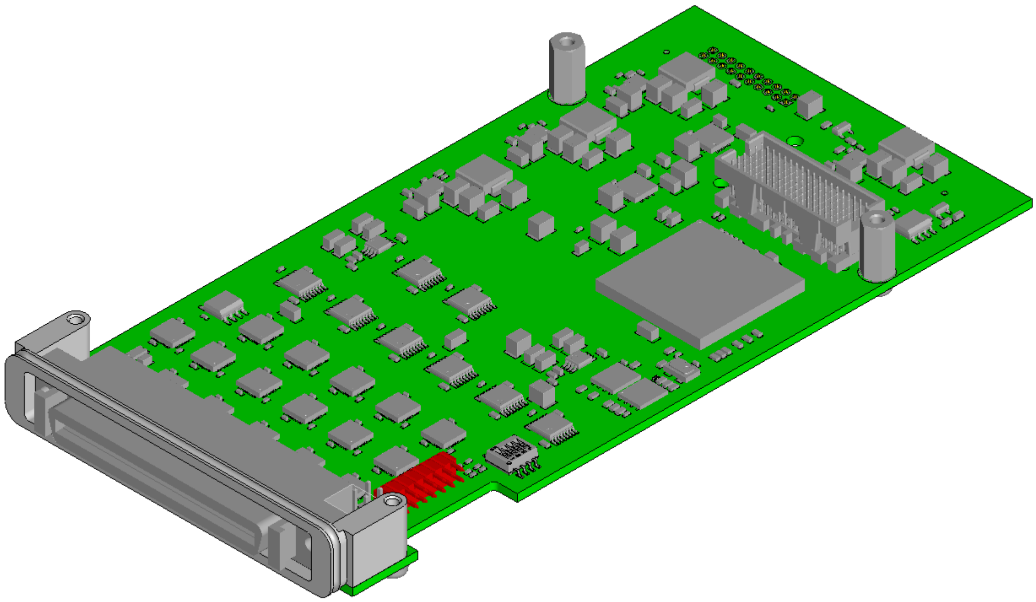


Figure 8-3 : X3 – JTAG Connector FPGA (3D Model)