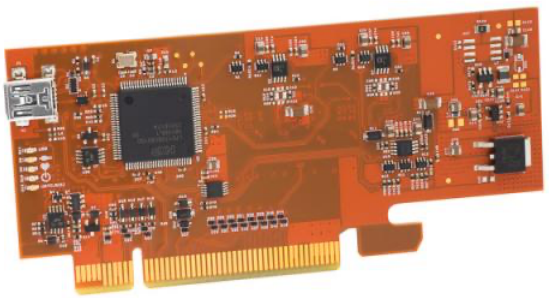


Overview

The Acroname® 18-channel MTM Analog I/O Module (MTM-DAQ-2), part of Acroname's MTM (Manufacturing Test Module) product series, is a ruggedized software-controlled analog measurement (ADC) and voltage source (DAC) module. The MTM-DAQ-2 allows MTM system designers to easily and modularly add analog voltage measurement and sources to their test system designs.

Ideal for use in high-reliability manufacturing or development testing environments, all features of the MTM-DAQ-2, including analog inputs, analog outputs, digital IO, and I²C communication channels are electrically protected for ESD strikes, overvoltage, overcurrent and short circuit conditions.

Built using Acroname's industry-proven and well-adopted BrainStem® technology, resources on the MTM-DAQ-2 are controlled via Acroname's powerful and extensible BrainStem software APIs.

Typical Application

- Manufacturing functional testing
- Validation testing
- Automated test development
- Embedded system development

System Features

- 16 differential bi-polar 16-bit analog inputs (ADC) with programmable gain
- 2 bipolar 12-bit analog voltage sources (DAC) with programmable gain
- 1 BrainStem I²C FM+ (1Mbit/s) bus
- 2 Digital GPIOs (overvoltage and current protected)

Description

The MTM-DAQ-2 module is a key component for manufacturing test and R&D of devices requiring precision voltage measurements. BrainStem interface and APIs are at https://acroname.com/reference.

The MTM-DAQ-2 implements an onboard BrainStem controller running an RTOS (Real-Time Operating System), which provides a USB host connection, Independent operating capability and the BrainStem interface, for control of the MTM resources identified in this datasheet. These resources can be controlled from a host computer over USB or in a network of MTM modules.

The MTM-DAQ-2 primarily provides 16 channels of 16-bit, differential analog voltage measurements with two independent ADCs. Every input channel has individually programmable gain allowing for selectable ranges of ±10.24V, 0-10.24V, ±5.12V, 0-5.12V, ±2.56V, 0-2.56V, ±1.28V, 0-1.28V and ±0.64V. Two of the input channels also have the additional ranges of ±1.024V, ±512mV, ±256mV, ±128mV, ±64mV. Further, the MTM-DAQ-2 provides two 12-bit, buffered-output analog DAC channels with selectable ranges of 0-2.048V, 0-4.096V, and ±10.24V. All of these features are easily controlled via the BrainStem API.

Within the MTM platform architecture, the MTM-DAQ-2 module can operate either independently or as a component in a larger network of MTM modules. Each MTM-DAQ-2 is uniquely addressable and controllable from a host by connecting via the onboard USB connection, the card-edge USB input or through other MTM modules on the local MTM/BrainStem I2C bus.

Acroname's BrainStem link is established over the selected input connection. The BrainStem link allows a connection to the onboard controller and access to the available resources in the MTM-DAQ-2. The MTM-DAQ-2 can then be controlled via a host running BrainStem APIs or it can operate independently by running locally embedded, user-defined programs based on Acroname's BrainStem Reflex language in the RTOS.

IMPORTANT NOTE

The MTM-DAQ-2 utilizes a PCIe connector interface but is for use strictly in MTM-based systems. It should never be installed in a PCI slot of a host computer directly. Insertion into a PC or non-MTM system could cause damage to the PC.

Absolute Maximum Ratings

Stresses beyond those listed under ABSOLUTE MAXIMUM RATINGS can cause permanent damage to the device. These are stress ratings only and functional operation of the device at these or any other conditions beyond those indicated under RECOMMENDED OPERATING CONDITIONS is not implied. Exposure to absolute-maximum rated conditions for extended periods affects device reliability and may permanently damage the device.

Voltage Rating	Minimum	Maximum	Units
Input Voltage, V_{supply}	-13.2	13.2	V
I2C0 SDA, SCL	-0.5	13.2	V
UART TX/RX	-0.5	13.2	V
DIO 0-1	-0.5	13.2	V
Module Address 0-3	-0.5	13.2	V
Reset	-0.5	13.2	V
USB D+, D-	-0.5	5.5	V
USB V_{bus}	-0.5	6.0	V
A_{IN-nP}, A_{IN-nN} to GND, n=[0-15]	-13.2	13.2	V
A_{out_n} to GND	-12.0	12.0	V

Table 1: Absolute Maximum Ratings

Current Rating	Minimum	Maximum	Units
Input Current, I_{supply}	0.0	5.5	A

Table 2: Absolute Maximum Current Ratings

The MTM system is designed to be used in a system where V_{supply} is the highest voltage connected to all MTM modules. Each module is designed to withstand V_{supply} continuously connected to all IOs, excepting those specified above, including accidental reverse polarity connection between V_{supply} and ground (0V). As with all products, care should be taken to properly match interface voltages and ensure a well-architected current-return path to ground. As with all devices utilizing USB interfaces, care should be taken to avoid ground loops within the USB subsystem. When using the USB interface, ground must be at 0V potential to avoid damaging connected host systems.

Handling Ratings

Parameter	Conditions/Notes	Minimum	Typical	Maximum	Units
Storage Temperature, T_{STG}		-10	-	85	°C
Relative Humidity Range	Non-Condensing	5	-	95	%RH
Electrostatic Discharge, V_{ESD}	IEC 61000-4-2, level 4, contact discharge to edge connector interface	-8	-	+8	kV

Table 3: Handling Ratings

Recommended Operating Ratings

Specifications are valid at 25°C unless otherwise noted. Intended for indoor use only.

Parameter	Conditions/Notes	Minimum	Typical	Maximum	Units
Ambient Operating Temperature, T_A	Non-Condensing	0	25	70	°C
Relative Humidity Range	Non-Condensing	5	-	95	%RH
Input Voltage, V_{supply}		6.0	-	12.5	V
Voltage to any IO pin		0	-	3.3	V
Voltage to any I2C pin		0	-	3.3	V
A_{IN-nP}, A_{IN-nN}	For *n* = [0-15]	-10.24	-	10.24	V
A_{OUT-nP}	For *n* = [16-17]	-10.24	-	10.24	V

Table 4: Recommended Operating Ratings

Block Diagram

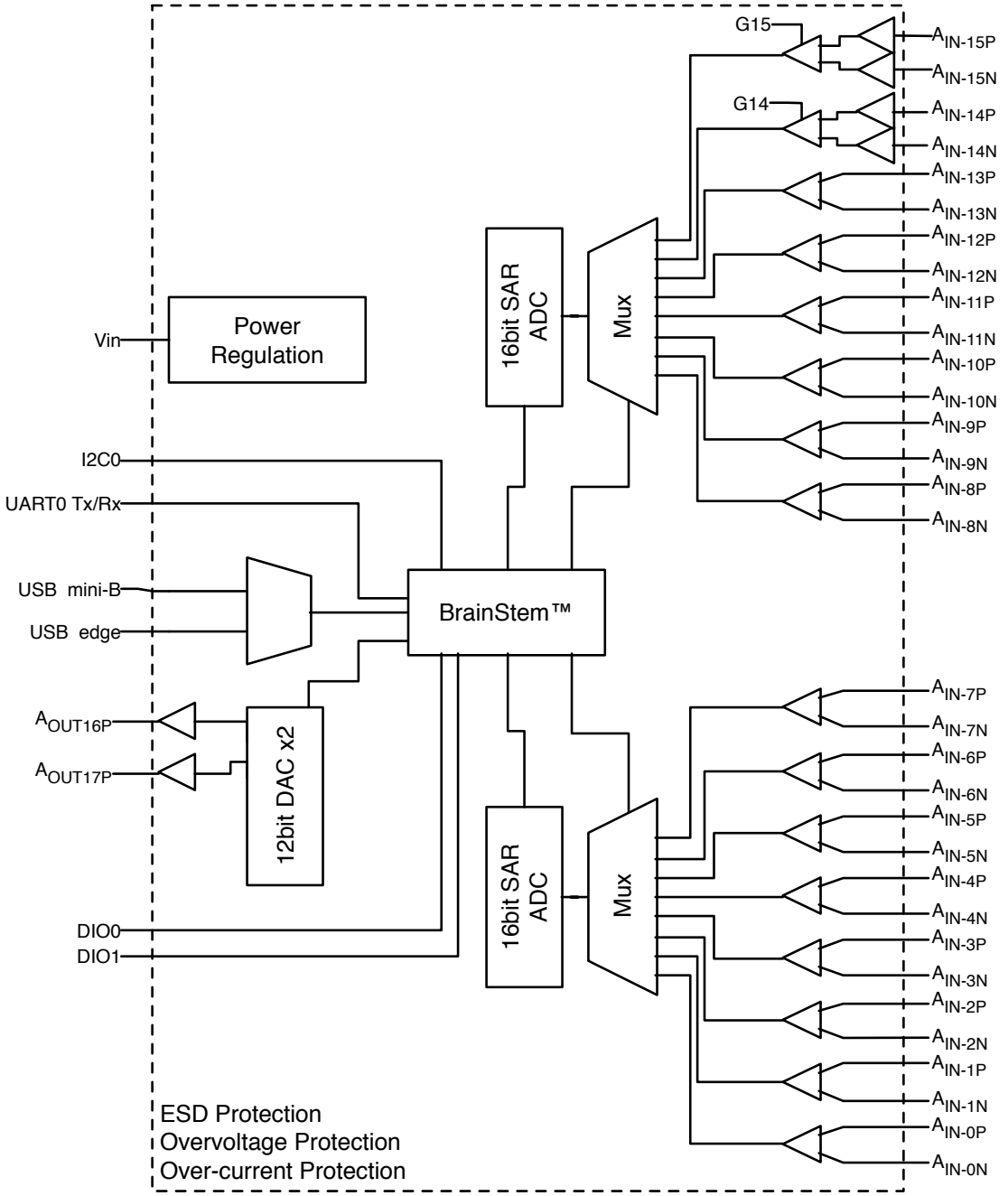


Figure 1: MTM-DAQ-2 Block Diagram

Typical Performance Characteristics

Specifications are valid at 25°C unless otherwise noted. Indoor application use only. Sample rates are typically limited by the USB throughput of the host operating system except where bulk capture is supported.

Parameter	Conditions/Notes	Minimum	Typical	Maximum	Units
Base Current Consumption, I_{supply}	Input voltage = 6V Input voltage = 12V	225 120	250 140	275 160	mA
Reset Low Threshold		-	1.2	-	V
I2C SDA, SCL Pins		0.0	3.3	5.0	V
Z_{id}, Input Differential Impedance	n = [0…15]		20 || 1		GW || pF
Z_{ii}, Input Common-mode Impedance	n = [0…15]		10 || 5		GW || pF
Input Bias Current, I_B	n = [0…15]	-	35	65	nA
Full-scale Analog Input Measurement Resolution *n* = [0…15]	Input range = ±10.24V	-	313	-	µV
	Input range = 0-10.24V	-	156	-	µV
	Input range = ±5.12V	-	156	-	µV
	Input range = 0-5.12V	-	78	-	µV
	Input range = ±2.56V	-	78	-	µV
	Input range = 0-2.56V	-	40	-	µV
	Input range = ±1.28V	-	40	-	µV
	Input range = 0-1.28V	-	20	-	µV
	Input range = ±0.64V	-	20	-	µV
Full-scale Analog Input Measurement Resolution *n* = [14,15]	Input range = ±1.024V	-	32	-	µV
	Input range = ±512mV	-	16	-	µV
	Input range = ±256mV	-	8	-	µV
	Input range = ±128mV	-	4	-	µV
	Input range = ±64mV	-	2	-	µV
Full-scale Analog Input Measurement Accuracy *n* = [0…15]		-1.0	-	1.0	% FSR
Full-scale Analog Input Common-mode Rejection Ratio (CMRR) *n* = [0…15]	V_{cm}=±12VDC	-	95	-	dB
Full-scale Analog Output Range Resolution *n* = [16,17]	Output range = ±10.24V	-	5000	-	µV
	Output range = 0-4.096V	-	1000	-	µV
	Output range = 0-2.048V	-	500	-	µV
Full-scale output range error *n* = [16,17]		-1.0	-	1.0	% FSR
Analog Output Sink Current		-	-	-15.0	mA
Analog Output Source Current		-	-	12.0	mA
Digital Output Logic High, V_{OH}		-	3.3	-	V
Digital Input Logic High, V_{IH}		2.15	-	-	V
Digital Input Logic Low, V_{IL}		-	-	1.1	V

Parameter	Conditions/Notes	Minimum	Typical	Maximum	Units
Digital Output Drive Current	Output high; short to GND Output high into 2.97V	- -	20.0 3.15	30.0 -	mA
Digital Output Sink Current	Output low; short to V_{supply}	-	-20.0	-30.0	mA
Digital Output Short Duration	Output high	-	Infinite	-	hours
Digital Output Overvoltage Duration	V_{supply} on pin	-	Infinite	-	hours
Digital Input Resistance	Configuration mode set to both Input and High-Z	-	4.25	4.45	MΩ
Digital Input Leakage Current	Configuration mode set to both Input and High-Z	-	110	-	uA
Digital Sample Rate[1]	via USB link, C++ Reflex	- -	1000 8200	- -	Hz

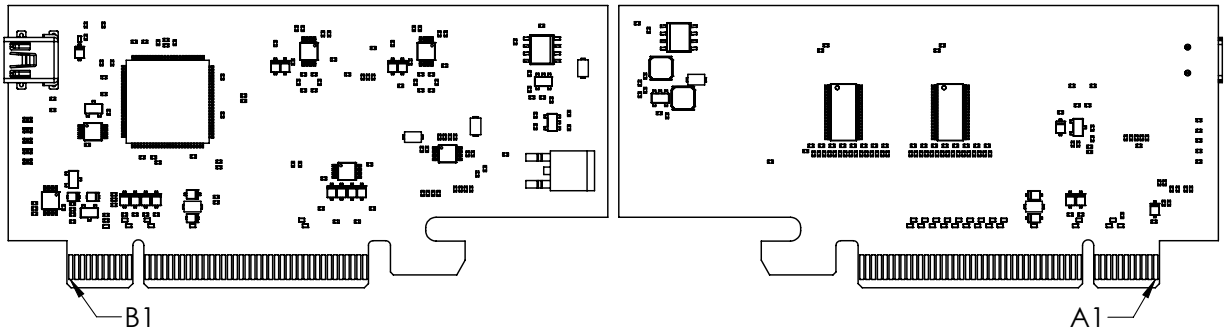
Table 5: Typical Performance Characteristics

[1] Host dependent, test was done as a single instruction, subsequent instructions may affect performance.

Pinout Descriptions

WARNING: MTM modules use a PCIe connector interface that is common in most desktop computers; however, they are NOT intended nor designed to work in these devices. Do NOT insert this product into any PCIe slot that wasn't specifically designed for MTM modules, such as a host PC. Installing this module into a standard PCI slot will result in damage to the module and the PC.

The MTM edge connector pin assignments are shown in the following table. Please refer to Table 4: Recommended Operating Ratings for appropriate signal levels.



Pins Common to all MTM Modules

Side A	Edge Connector Side A Description	Side B	Edge Connector Side B Description
1	GND	1	Input Voltage, Vsupply
2	GND	2	Input Voltage, Vsupply
3	GND	3	Input Voltage, Vsupply
4	GND	4	Input Voltage, Vsupply
5	Reset	5	Input Voltage, Vsupply
6	GND	6	Reserved, Do Not Connect
7	GND	7	Reserved, Do Not Connect
8	I2C0 SCL	8	GND
9	I2C0 SDA	9	GND
10	GND	10	Reserved, Do Not Connect
11	GND	11	Reserved, Do Not Connect
12	Module Address Offset 0	12	Module Address Offset 2
13	Module Address Offset 1	13	Module Address Offset 3

Table 6: Pins Common to all MTM Modules

Pins Specific to MTM-DAQ-2

Side A	Edge Connector Side A Description	Side B	Edge Connector Side B Description
14	Reserved, Do Not Connect	14	USB Upstream Data +
15	Reserved, Do Not Connect	15	USB Upstream Data -
16:17	Reserved, Do Not Connect	16:17	Reserved, Do Not Connect
18	Digital IO 1	18	Digital IO 0
19	Analog CH0 Positive Input	19	Analog CH1 Positive Input
20	Analog CH0 Negative Input	20	Analog CH1 Negative Input
21	Analog CH2 Positive Input	21	Analog CH3 Positive Input
22	Analog CH2 Negative Input	22	Analog CH3 Negative Input
23	Analog CH4 Positive Input	23	Analog CH5 Positive Input
24	Analog CH4 Negative Input	24	Analog CH5 Negative Input
25	Analog CH6 Positive Input	25	Analog CH7 Positive Input
26	Analog CH6 Negative Input	26	Analog CH7 Negative Input
27	Analog CH8 Positive Input	27	Analog CH9 Positive Input
28	Analog CH8 Negative Input	28	Analog CH9 Negative Input
29:33	Reserved, Do Not Connect	29:33	Reserved, Do Not Connect
34	Analog CH10 Positive Input	34	Analog CH11 Positive Input
35	Analog CH10 Negative Input	35	Analog CH11 Negative Input
36	Analog CH12 Positive Input	36	Analog CH13 Positive Input
37	Analog CH12 Negative Input	37	Analog CH13 Negative Input
38	Analog CH14 Positive Input	38	Analog CH15 Positive Input
39	Analog CH14 Negative Input	39	Analog CH15 Negative Input
40	Analog CH16 Positive Output	40	Analog CH17 Positive Output
41:49	Reserved, Do Not Connect	41:49	Reserved, Do Not Connect

Table 7: Pins Specific to MTM-DAQ-2

Module Hardware and Software Default Values

Software Control

The MTM-DAQ-2 module firmware is built on Acroname's BrainStem technology and utilizes a subset of BrainStem entity implementations that are specific to the hardware's capabilities. Table 8 details the BrainStem API entities and macros used to interface with the MTM-DAQ-2 module. For C and C++ developers, these macros are defined in `aMTMDAQ2.h` from the BrainStem development package. For Python development, the module `MTMDAQ2` class defines the extent of each entity array.

While Table 8 lists the BrainStem API entities available for this module, not all entity methods are supported by the MTM-DAQ-2. For a complete list of supported entity methods, see Table 13. Note that available method options may vary by entity index, as well as by entity, and calling an unsupported entity option will return an appropriate error (e.g.: `aErrInvalidEntity`, `aErrInvalidOption`, `aErrMode`, or `aErrUnimplemented`) as defined in `aError.h` for C and C++ and the `Result` class in Python.

All API example code snippets that follow are pseudocode loosely based on the C++ method calls - Python and Reflex are similar. Please consult the BrainStem Reference for specific implementation details[2].

Parameter	Index	Macro Name or Implemented Options	Notes
Module Definitions:			
Module Base Address	10	aMTMDAQ2_MODULE_BASE_ADDRESS	See aMTMDAQ2.h
Entity Class Definitions:			
`analog` Entity Quantity	18	aMTMDAQ2_NUM_ANALOGS	
`digital` Entity Quantity	2	aMTMDAQ2_NUM_DIGITALS	
`i2c` Entity Quantity	1	aMTMDAQ2_NUM_I2C	
`Store` Entity Quantity	2	aMTMDAQ2_NUM_STORES	
`system` Entity Quantity	1		
`timer` Entity Quantity	8	aMTMDAQ2_NUM_TIMERS	
`app` Entity Quantity	4	aMTMDAQ2_NUM_APPS	
`pointer` Entity Quantity	4	aMTMDAQ2_NUM_POINTERS	

Table 8: MTM-DAQ-2 Hardware and Software Default Values[2]

[2] Refer to `aMTMDAQ-2.h` within the BrainStem Development Kit download for actual file.

Capabilities and Interfaces

BrainStem Link and Module Networking

A BrainStem link can be established that will give the user access to the resources available on the MTM-DAQ-2. The module can then be controlled via a host running BrainStem APIs or operated independently by running locally embedded, user-defined programs based on Acroname's BrainStem Reflex language in the RTOS.

A BrainStem link to the MTM-DAQ-2 can be established via one of three (3) interfaces: the onboard USB connection, the card-edge USB connection, or through another MTM module using the BrainStem protocol (more on this interface below). For the USB connection options, once the MTM-DAQ-2 is attached to a host machine, a user can connect to it via software API:

```
stem.link.discoverAndConnect(linkType,
serialNumber, modelNumber)
```

The MTM-DAQ-2 can also work within a network of other Brainstem modules, such as in a test fixture, to give access to the capabilities of all networked modules. On the MTM platform, networked modules communicate using the Brainstem protocol, which is transmitted over I²C. Each MTM-DAQ-2 is uniquely addressable via hardware or software to avoid communication conflicts on the I²C bus. A software offset can be applied as follows:

```
stem.system.setModuleSoftwareOffset(address)
```

Module Address Hardware Offset Configuration

A hardware offset allows a user to modify the module's address on the BrainStem network. Using hardware offset pins is useful when more than one of the same module type is installed on a single BrainStem network. Applying a different hardware offset to each module of the same type in one network allows for all the modules to seamlessly and automatically configure the network for inter-module communication. Further, modules can be simply swapped in and out of the network without needing to pre-configure a module's address before being added to a network. Finally, when a system has more than one of the same module type in a network, the module address hardware offset can be used to determine the module's physical location and thus its interconnection and intended function. For detailed information on BrainStem networking see the BrainStem Reference[3].

Each hardware offset pin can be left floating or pulled to ground with a 1kΩ resistor or smaller (pin may be directly shorted to ground). Pin states are only read when the module boots, either from a power cycle, hardware reset, or software reset. The hardware offset pin states are treated as a bit state within a 4bit number. This number is multiplied by 2 and added the to the module's base address. The hardware offset calculation is detailed in the following table:

HW Offset Pin				Address Offset	Module Base Address	Final Module Address
3	2	1	0			
NC	NC	NC	NC	0	4	4
NC	NC	NC	1	2	4	6
NC	NC	1	NC	4	4	8
NC	1	NC	NC	8	4	12
1	NC	NC	NC	16	4	20
1	NC	NC	1	(1+8) * 2	4	22

Table 9: Module Address Hardware Offset Examples

Upstream USB Connectivity Options

The MTM-DAQ-2 supports upstream USB connections (to communicate to a host PC) via the mini-B connector, or through pins B14 and B15 of the PCIe edge connector. The module defaults to using the edge connector and will switch to the mini-B connector if 5V is present on V_{bus} at the mini-B connector.

System Entities

Every BrainStem module includes a single System Entity. The System Entity allows access to configuration settings such as the module address and I²C rate, measurements such as input voltage, control over the user LED, and many more.

Saving Entity Settings

Some entities can be configured and saved to non-volatile memory. This allows a user to modify the startup and operational behavior for the MTM-DAQ-2 away from the factory default settings. Saving system settings creates a new default and often requires a reboot of the MTM-DAQ-2 for changes to take effect; see Table 10: Entities Values Saved by system.save() for relevant settings. Use the following command to save changes to system settings before reboot:

```
stem.system.save()
```

Saved Configurations	
Module Software Offset	I2C Rate
Router Address	I2C Pullup State
Heartbeat Rate	Boot Slot

Table 10: Entities Values Saved by system.save()

Store Entities

Every BrainStem module includes several Store entities and onboard memory slots to load Reflex files (for details on Reflex, see BrainStem Reference[3]). One Reflex file can be

stored per slot. Store[0] refers to the internal flash memory, with 12 available slots, and store[1] refers to RAM, with 1 available slot.

Digital Entities

The MTM-DAQ-2 has two (2) digital input/outputs (DIO) controlled by the digital entity. Each DIO is controllable via software and is independently current limited for both source and sink currents.

All DIO are input and output capable:

```
stem.digital[0].setConfiguration(mode)
stem.digital[0].getConfiguration(mode)
```

The *mode* parameter is an integer that correlates to the following:

Value	Configuration
0	Input with Pullup
1	Output
4	Hi Impedance (Hi-Z)
5	Input with Pull Down

Table 11: Digital IO Configuration Values

- Example: If a digital pin is configured as output mode, set the digital logic level:

```
stem.digital[0].setState(level)
```

Example: If a digital pin is configured as input mode, read the digital logic level:

```
stem.digital[0].getState(level)
```

If a digital pin is configured in Hi-Z mode its internal circuitry has been disconnected to create a high impedance. There are no functions that can act on this configuration.

Digital	Input	Output	HighZ	RCServo	Signal
DIO 0	Yes	Yes	Yes	-	-
DIO 1	Yes	Yes	Yes	-	-
DIO 2	Yes	Yes	Yes	-	-
DIO 3	Yes	Yes	Yes	-	-

Table 12: Digital IO Pin Configurations

Analog Entities

The MTM-DAQ-2 has sixteen (16) analog inputs (ADC) and two (2) analog outputs (DAC) all controlled by the analog entity. Each analog is controllable via software and is independently current-limited for both source and sink currents. The analog inputs are connected to a 16-bit ADC, and return a voltage value in microvolts.

Analog[0…13] are fully-differential with individually selectable input ranges of ±10.24V, 0-10.24V, ±5.12V, 0-5.12V, ±2.56V, 2.56V, ±1.28V, 0-1.28V, and ±0.64V:

```
stem.analog[0].setRange(analogRangeP10V24N10V24
)
v = stem.analog[0].getVoltage()
```

Analog[14, 15] are fully differential and have the bipolar ranges above and the additional ranges of ±1.024V, ±512mV, ±256mV, ±128mV, and ±64mV:

```
stem.analog[14].setRange(analogRangeP0V064N0V06
4)
v = stem.analog[14].getVoltage()
```

Analog[16, 17] are analog outputs with individually selectable ranges of 0-2.048V, 0-4.096V, and ±10.24V. The outputs are connected to a 12-bit DAC and take a voltage setpoint in microvolts. These outputs default to having their outputs disabled, so setEnable(1) must be called before their voltage will be present on their respective pins:

```
stem.analog[16].setRange(analogRangeP4V096N0V0)
stem.analog[16].setVoltage(voltage)
stem.analog[16].setENABLE(1)
```

I²C Entities

The MTM-DAQ-2 includes access to a single I²C bus operating at a set 1Mbit/s rate.

Note: The 1Mbit/s bus, while user-accessible, is also used for BrainStem network communication so there may be other, non-user-initiated traffic when other BrainStem modules are linked.

Example: Sending 2 bytes (0xABCD) through the I²C bus to a device with address 0x42:

```
stem.i2c.write(0x42, 2, 0xABCD)
```

Example: Reading 2 bytes of data from a device with address 0x42:

```
stem.i2c.read(0x42, 2, buffer)
```

Where *buffer* would be a char array in C++.

The maximum data size for individual read and write operations on an I²C bus through the BrainStem API is 20 bytes. Sending more than 20 bytes of information must be done as an iterated sequence.

Each I²C bus also includes 330Ω pull-up resistors on the SDA and SCL lines, disabled by default. When using the MTM-DAQ-2 in a linked system (communicating over the 1Mbit/s bus), only a single set of pull-ups along the bus should be enabled in order for the I²C bus to work properly (if more than one set is enabled, the lines cannot be pulled low for communication). Similarly, when using a single MTM device to communicate with an external device over the I²C bus, either the internal pull-ups can be enabled, or external hardware pull-ups added.

```
stem.i2c.setPullUp(bEnable)
```

Reflex RTOS

Reflex is Acroname's real-time operating system (RTOS) language which runs in parallel to the module's firmware. Reflex allows users to build custom functionality directly into the device. Reflex code can be created to run autonomously on the module or a host can interact with it through BrainStem's Timer, Pointer App and other entities.

Timer Entities

The Timer entity provides simple scheduling for events in the reflex system. The MTM-DAQ-2 includes 8 timers per reflex. Each timer represents a reflex definition to be executed upon expiration of a running timer. Timers can be controlled from a host, but the reflex code is executed on the device.

Example: Setting up and starting Timer 0 for single use:

```
stem.timer[0].setMode(timeModeSingle)
stem.timer[0].setExpiration(DELAY)
```

Reflex Definition: Timer 0 expiration callback:

```
reflex timer[0].expiration() { //Do Stuff }
```

Pointer Entities

Reflex and the Brainstem module share a piece of memory called the scratchpad which can be accessed via the Pointer Entity. The MTM-DAQ-2 has 4 pointers per reflex which allow access to the pad in a similar manner as a file pointer.

Example: Configure and access the scratchpad in static mode:

```
stem.pointer[0].setMode(pointerModeStatic)
stem.pointer[0].getByte(byte)
```

Reflex Pad: Single unsigned char definition:

```
pad[0:0] unsigned char byteValue
```

App Entities

Apps are reflex definitions that can be directly trigger by the host. These definitions are also capable of passing a parameter into or out of the app reflex definition. The MTM-DAQ-2 is equipped with 4 App Entities per reflex.

Example: Triggering App 0:

```
stem.app[0].execute(parameter)
```

Reflex Definition: App 0 callback:

```
reflex app[0](int appParam) { //Do Stuff }
```

MTM-DAQ-2 Supported Entity Methods Summary

Detailed entity class descriptions can be found in the BrainStem Reference[3]. A summary of MTM-DAQ-2 class options are shown below. Note that when using Entity classes with a single index (aka, 0), the index parameter can be dropped. For example:

```
stem.system[0].setLED(1) → stem.system.setLED(1)
```

Entity Class	Entity Option	Variable(s) Notes
digital[0-1]	setConfiguration	
	getConfiguration	
	setState	
	getState	
i2c[0]	write	
	read	
analog[0-15]	getVoltage	
	getValue	
	setRange	
	getRange	
analog[16-17]	setVoltage	
	getVoltage	
	setValue	
	getValue	
	setRange	
	getRange	
	setEnable	
	getEnable	
store[0-2]	getSlotState	
	loadSlot	
	unloadSlot	
	slotEnable	
	slotDisable	
	slotCapacity	
	slotSize	
system[0]	save	
	reset	
	setLED	
	getLED	
	setBootSlot	
	getBootSlot	
	getInputVoltage	
	getVersion	
	getModuleBaseAddress	
	getModuleSoftwareOffset	
	setModuleSoftwareOffset	
	getModuleHardwareOffset	
	setHBInterval	

Entity Class	Entity Option	Variable(s) Notes
	getHBInterval	
	getRouterAddressSetting	
	getModule	
	getSerialNumber	
	setRouter	
	getRouter	
	getModel	
	routeToMe	
timer[0-7]	getExpiration	
	setExpiration	
	getMode	
	setMode	
Pointer[0-3]	getOffset	
	setOffset	
	getMode	
	setMode	
	getTransferStore	
	setTransferStore	
	initiateTransferToStore	
	initiateTransferFromStore	
	getChar	
	setChar	
	getShort	
	setShort	
	getInt	
	setInt	
App[0-3]	execute	

Table 13: Supported MTM-DAQ-2 BrainStem Entity API Methods[3]

[3] See BrainStem software API reference at https://acroname.com/reference for further details about all BrainStem API methods and information.

LED Indicators

The MTM-DAQ-2 board has five LED indicators to assist with MTM system development, debugging, and monitoring. These LEDs are shown in Figure 2: MTM-DAQ-2 LED Indicators.

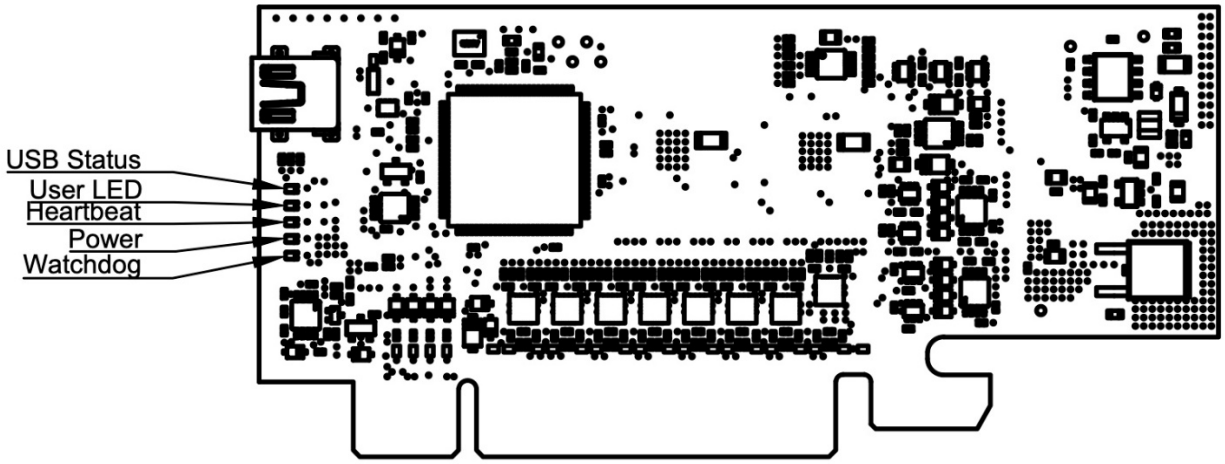
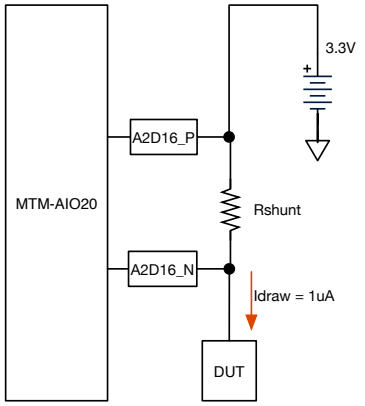


Figure 2: MTM-DAQ-2 LED Indicators

Application Examples



A2D Range	A2D LSB	Rshunt
±2048uV	62.5pV	62.5uΩ
±20.48mV	62.5nV	625mΩ
±204.8mV	6.25uV	6.25Ω
±1.024V	31.25uV	31.25Ω
±2.048V	62.5uV	62.5Ω

Figure 3 Measuring 1uA current consumption.

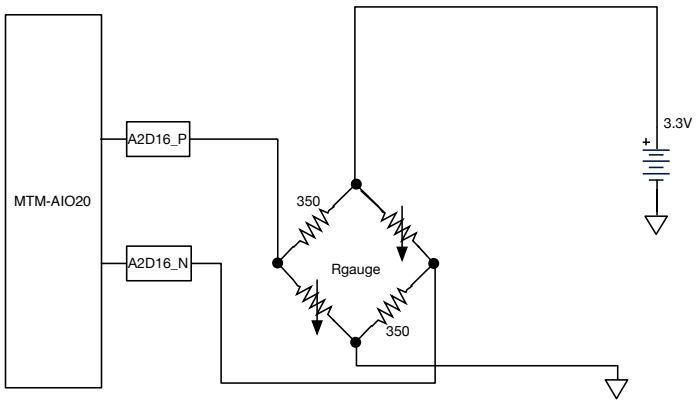


Figure 4 Strain Gauge Interface.

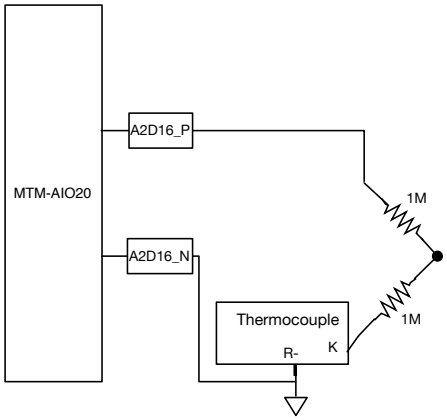


Figure 5 K-Type Thermocouple Interface.

Edge Connector Interface

All MTM products are designed with an edge connector interface that requires a compatible board-edge connector on the carrier PCB. Acroname recommends the through-hole PCI-Express (PCIe) Vertical Connector. The connectors can be combined with an optional retention clip, as shown below. Representative part numbers are show in Table 14, and equivalent connectors are offered from a multitude of vendors.

Manufacturer	Manufacturer Part Number	Description
Amphenol FCI Samtec	10018784-10202TLF PCIE-098-02-F-D-TH	PCI-Express 98-position vertical connector
Amphenol FCI	10042618-003LF	PCI-Express Retention Clip (optional)

Table 14: PCI-Express Edge Connectors for MTM Products

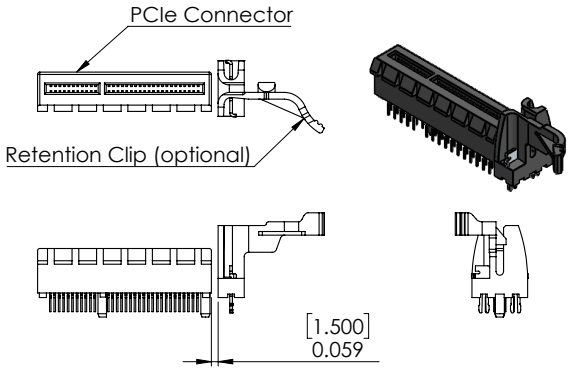


Figure 6: PCIe Vertical Connector with optional Retention Clip

MTM Edge Connector Specifications	Description
Contact Finish	Gold
Card Thickness	0.0625” [1.59mm]
Number of Rows	2
Number of Positions	Variable (see Table 14: PCI-Express Edge Connectors for MTM Products)
Pitch	0.039” (1.00mm)

Table 15: MTM Edge Connector Specifications

Mechanical

Dimensions are shown in mm. 3D CAD models are available through the MTM-DAQ-2 product page's Downloads section. A 3D CAD viewer with many different CAD model formats available for download is available at https://a360.co/2YO3dY4

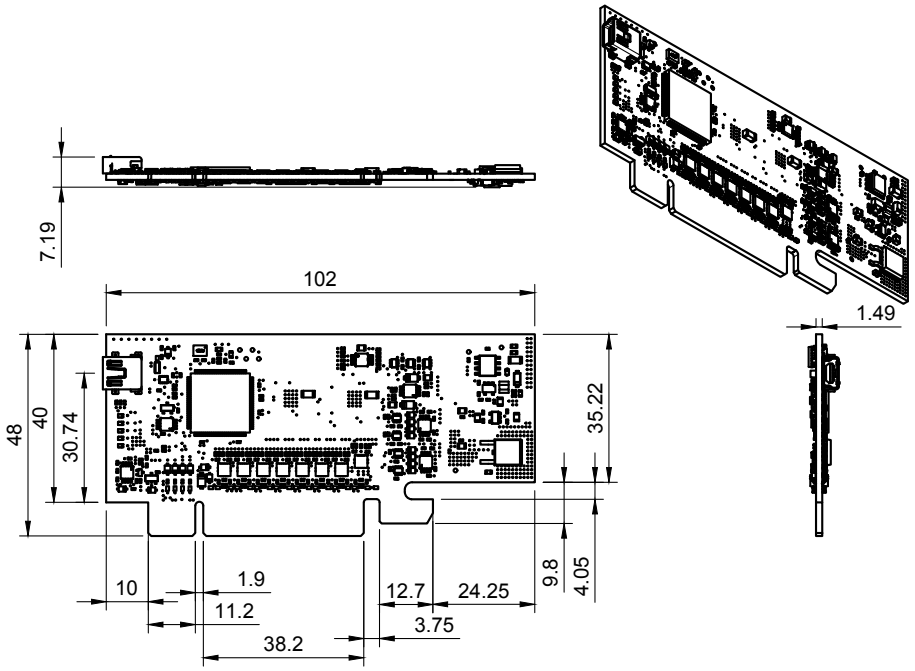


Figure 7: MTM-DAQ-2 Mechanical

Document Revision History

All major documentation changes will be marked with a dated revision code

Revision	Date	Engineer	Description
1.0	September 2018	LCD	Initial Release
1.1	September 2018	JLG	Update analog[16-17] entities and spec table
1.2	October 2019	GCF	Updated DAQ 1 to DAQ 2 typos Fixed #define reference names Fixed table 7
1.3	July 2020	ACRO	Review of specifications, updates to CAD models, formatting, entity updates, diagram updates
1.4	July 2020	ACRO	Updated tables and fixed module name in Reflex section
1.5	July 2020	JLG	Add CMRR spec; update recommended input voltage; formatting fixes
1.6	August 2020	RMN	Fixed typo in pinout.

Table 16: Document Revision History