Overview

The MTM-Relay (S78-MTM-RELAY), as part of the Acroname® MTM (Manufacturing Test Module) product series, is a ruggedized set of software-controlled SSR (Solid State Relay) modules with user-accessible serial communication. The MTM-Relay allows MTM system designers to easily and modularly add relay-based power switching to their test system designs.

Ideal for use in high-reliability manufacturing or development testing environments, all features of the MTM-Relay, including relays, 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-Relay are controlled via Acroname's powerful and extensible BrainStem technology and software APIs.

Typical Application

– Manufacturing functional testing
– Validation testing
– Automated test development
– Embedded system development
– Power switching

System Features

– 4 60V, 6A optically isolated solid state relays (NO)
– 4 Digital GPIOs (overvoltage and current protected)
– 1 BrainStem I²C FM+ (1Mbit/s) bus

Description

As part of Acroname's MTM product series, the MTM-Relay module is a key component for manufacturing test and R&D of devices requiring high power switching using durable relays. BrainStem interface and APIs are at https://acroname.com/reference.

The MTM-Relay implements an onboard BrainStem controller running an RTOS (Real-Time Operating System), which provides a host connection, independent operating capability and the BrainStem interface, for control of the MTM resources identified in this datasheet (Relay X, GPIO, etc.).

The MTM-Relay primarily provides four main software-controlled SSR relays to implement high-power switching in a test system. The relays are controlled by the BrainStem API.

Within the MTM platform architecture, the MTM-Relay module can operate either independently or as a component in a larger network of MTM modules. Each MTM-Relay 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 BrainStem 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-Relay. The MTM-Relay 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-Relay 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.

Description	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 0-3	-0.5	13.2	V
Module Address 0-3	-0.5	13.2	V
Reset	-0.5	13.2	V
Relay Blocking Voltage	-	60	V
USB D+, D-	-0.5	5.5	V
USB V_{bus}	-0.5	6.0	V
Relay Input Power Dissipation (each relay)	-	150	mW
Relay Total Power Dissipation (each relay)	-	2400	mW
Relay Isolation Voltage	-	1.0	kV_{rms}

Table 1: Absolute Maximum 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
Ambient Operating Temperature, T_A	Non-Condensing	0	25	70	°C
Relative Humidity Range	Non-Condensing	5	-	95	%RH
Storage Temperature, T_{STG}		-10	-	85	°C
Electrostatic Discharge, V_{ESD}	IEC 61000-4-2, level 4, contact discharge to edge connector interface	-8	-	+8	kV

Table 2: 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
Input Voltage, V_{supply}		6.0	-	12.0	V
Voltage to any IO pin		0	-	3.3	V
Voltage to any I2C pin		0	-	3.3	V
Relative Humidity Range	Non-Condensing	5	-	95	%RH

Table 3: Recommended Operating Ratings

Block Diagram

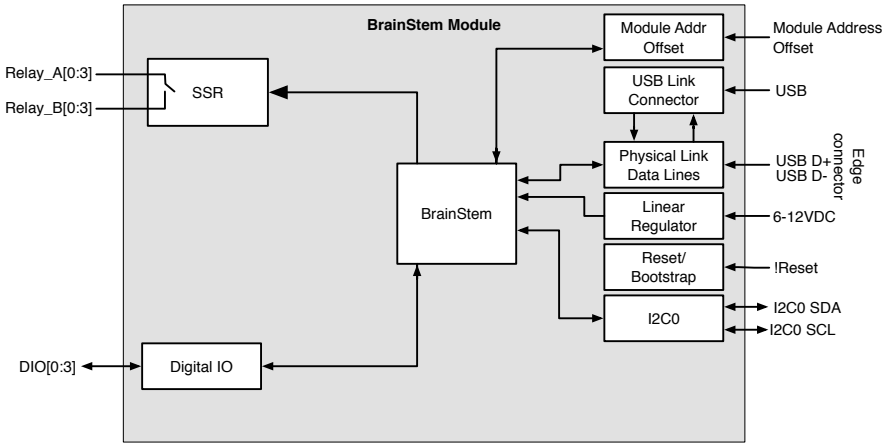


Figure 1: MTM-Relay Block Diagram

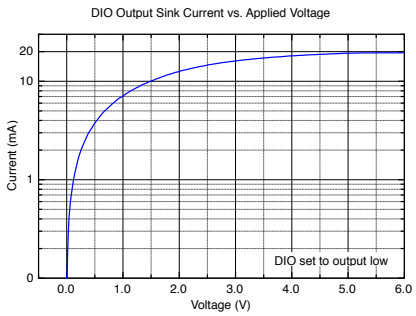
Typical Performance Characteristics

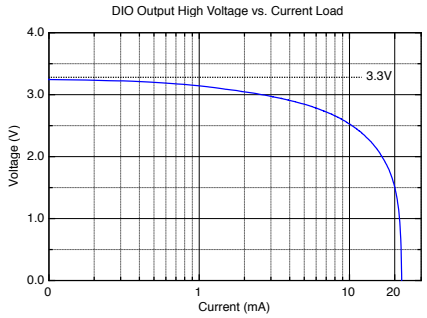
Specifications are valid at 25°C unless otherwise noted. Indoor application use only. Sample rates are typically limited by the USB throughput of the host operating system except where bulk capture is supported.

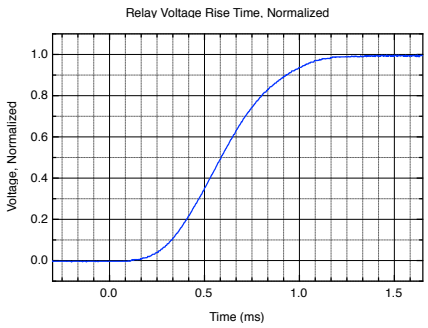
Parameter	Conditions/Notes	Minimum	Typical	Maximum	Units
Base Current Consumption, I_{supply}	V_{supply} = 6V V_{supply} = 12V	- -	78 90	- -	mA
Reset Low Threshold		-	1.2	-	V
I2C SDA, SCL Pins		0.0	3.3	-	V
Digital Input Logic High, V_{IH}		2.15	-	-	V
Digital Input Logic Low, V_{IL}		-	-	1.1	V
Digital Input Leakage Current	Mode set Input or High-Z	-	110	-	µA
Digital Input Resistance	Mode set Input or High-Z	-	4.25	4.45	MΩ
Digital Output Logic High, V_{OH}		-	3.3	-	V
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 Sample Rate[1]	via USB link, C++ Reflex	- -	1000 8200	- -	Hz
Relay Load Current, Continuous, I_L	Free air	-6	-	6	A_{DC}, A_{rms}
Relay Peak Load Current, I_{LPK}	t = 10ms	-20	-	20	A
Relay On-Resistance, R_{ON}	I_L = 1A	-	-	0.06	W
Relay Off-State Leakage Current, I_{LEAK}	V_L = 60V	-	-	1	µA
Relay Turn-On, t_{ON}	I_L=100mA	900	1300	5000	µs
Relay Turn-Off, t_{OFF}	I_L=100mA	10	14	1000	µs
Relay Output Capacitance, C_{OUT}	V_L=50V, f=1MHz	-	340	-	pF

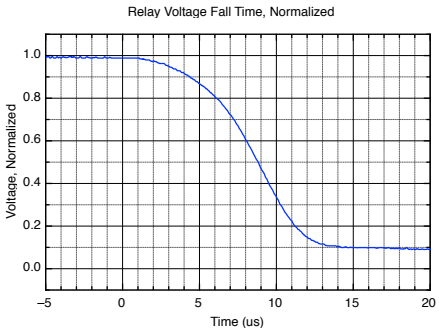
Table 4: Typical Performance Characteristics

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









Controlling Inductive Loads

When controlling inductive loads with the MTM-Relay, a large transient voltage spike (flyback voltage) may be created during switching events due to energy stored in the load inductance. These spikes may cause damage to the MTM-Relay module. Due to the wide variation of voltage ranges and power applications for relays, flyback protection and ESD diodes are not included on the MTM-Relay SSR terminals. Should such additional protection be required in the desired application, Acroname recommends implementing such features using external components.

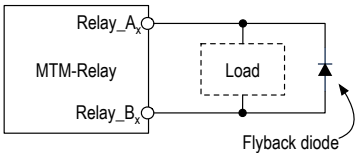
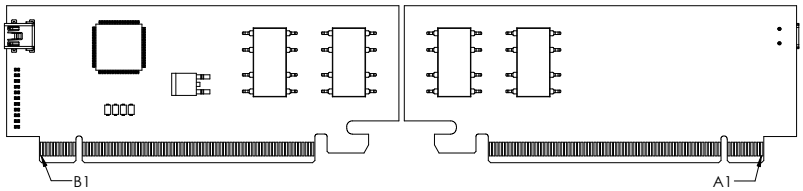


Figure 2: Limiting flyback voltages with inductive loads

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 3: 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, V_{supply}
2	GND	2	Input Voltage, V_{supply}
3	GND	3	Input Voltage, V_{supply}
4	GND	4	Input Voltage, V_{supply}
5	Reset	5	Input Voltage, V_{supply}
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 5: Pins Common to all MTM Modules

Pins Specific to MTM-Relay

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	Reserved, Do Not Connect	16	Reserved, Do Not Connect
17	Reserved, Do Not Connect	17	Reserved, Do Not Connect
18	Digital IO 1	18	Digital IO 0
19	Digital IO 3	19	Digital IO 2
20:49	Reserved, Do Not Connect	20:49	Reserved, Do Not Connect
50	GND	50	Reserved (24V)
51	GND	51	Reserved (24V)
52	GND	52	Reserved (24V)
53	Reserved, Do Not Connect	53	Reserved, Do Not Connect
54	Reserved, Do Not Connect	54	Reserved, Do Not Connect
55	Reserved, Do Not Connect	55	Reserved, Do Not Connect
56	Reserved, Do Not Connect	56	Reserved, Do Not Connect
57	Reserved, Do Not Connect	57	Reserved, Do Not Connect
58	Reserved, Do Not Connect	58	Reserved, Do Not Connect
59	Reserved, Do Not Connect	59	Reserved, Do Not Connect
60	Relay 0, A	60	Relay 2, A
61	Relay 0, A	61	Relay 2, A
62	Relay 0, A	62	Relay 2, A
63	Reserved, Do Not Connect	63	Reserved, Do Not Connect
64	Reserved, Do Not Connect	64	Reserved, Do Not Connect
65	Relay 0, B	65	Relay 2, B
66	Relay 0, B	66	Relay 2, B
67	Relay 0, B	67	Relay 2, B
68	Reserved, Do Not Connect	68	Reserved, Do Not Connect
69	Reserved, Do Not Connect	69	Reserved, Do Not Connect
70	Reserved, Do Not Connect	70	Reserved, Do Not Connect
71	Reserved, Do Not Connect	71	Reserved, Do Not Connect
72	Reserved, Do Not Connect	72	Reserved, Do Not Connect
73	Reserved, Do Not Connect	73	Reserved, Do Not Connect
74	Reserved, Do Not Connect	74	Reserved, Do Not Connect
75	Relay 1, A	75	Relay 3, A
76	Relay 1, A	76	Relay 3, A
77	Relay 1, A	77	Relay 3, A
78	Reserved, Do Not Connect	78	Reserved, Do Not Connect
79	Reserved, Do Not Connect	79	Reserved, Do Not Connect
80	Relay 1, B	80	Relay 3, B
81	Relay 1, B	81	Relay 3, B
82	Relay 1, B	82	Relay 3, B

Table 6: Pins Specific to MTM-Relay

Module Hardware and Software Default Values

Software Control

The MTM-Relay 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 7 details the BrainStem API entities and macros used to interface with the MTM-Relay. For C and C++ developers, these macros are defined in `aMTMRelay.h` from the BrainStem development package. For Python development, the module `MTMRelay` class defines the extent of each entity array.

While Table 7 lists the BrainStem API entities available for this module, not all entity methods are supported by the MTM-Relay. For a complete list of supported entity methods, see Table 12. 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[3].

Parameter	Index	Macro Name or Implemented Options	Notes
Module Definitions:			
Module Base Address	12	`aMTM_RELAY_MODULE_BASE_ADDRESS`	See aMTMRelay.h
Entity Class Definitions:			
`relay` Entity Quantity	4	`aMTM_RELAY_NUM_RELAYS`	
`digital` Entity Quantity	4	`aMTM_RELAY_NUM_DIGITALS`	
`i2c` Entity Quantity	1	`aMTM_RELAY_NUM_I2C`	
`Store` Entity Quantity	2	`aMTM_RELAY_NUM_STORES`	
`system` Entity Quantity	1		
`timer` Entity Quantity	8	`aMTM_RELAY_NUM_TIMERS`	
`app` Entity Quantity	4	`aMTM_RELAY_NUM_APPS`	
`pointer` Entity Quantity	4	`aMTM_RELAY_NUM_POINTERS`	

Table 7: MTM-Relay Hardware and Software Default Values[2]

[2] Refer to `aMTMRelay.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-Relay. 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-Relay 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-Relay is attached to a host machine, a user can connect to it via software API:

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

The MTM-Relay 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-Relay 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)
```

Upstream USB Connectivity Options

The MTM-Relay 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 when 5V is present on V_{bus} at the mini-B connector.

Module Address Hardware Offset Configuration

A hardware offset is one of two ways to modify the module's address on the BrainStem network. Using hardware offset pins is useful when more than one of the same type of module 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 type of module 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 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 pins are treated as a binary number which 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	8	8
NC	NC	NC	1	2	8	10
NC	NC	1	NC	4	8	12
NC	1	NC	NC	8	8	16
1	NC	NC	NC	16	8	24
1	NC	NC	1	2+16	8	26

Table 8: Module address hardware offset examples

System Entities

Every BrainStem module includes a single System Entity. The System Entity allows access to configuration settings such as the module address, router information, 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-Relay *away* from the factory default settings. Saving system settings creates a new default and often requires a reboot of the MTM-Relay for changes to take effect; see Table 9: 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	
Software Offset	I2C Rate
Router Address	Boot Slot
Heartbeat Rate	

Table 9: 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 memory, with 12 available slots, and store[1] refers to RAM, with 1 available slot.

Digital Entities

The MTM-Relay has four (4) 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
1	Output
4	Hi Impedance (Hi-Z)
5	Input with Pull Down

Table 10: 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
DIO0	Yes	Yes	Yes	None	None
DIO 1	Yes	Yes	Yes	None	None
DIO 2	Yes	Yes	Yes	None	None
DIO 3	Yes	Yes	Yes	None	None

Table 11: Digital IO Pin Configurations

Relay Entities

The MTM-Relay has four (4) optically isolated solid-state relays controlled by the relay entity. Each relay is controllable via software and capable of 60V and 6A continuous current load.

Example: Enabling the relay:

```
stem.relay[0].setState(state)
```

I²C Entities

The MTM-Relay 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-USBStem 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-USBStem 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-USBStem 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-Relay Supported Entity Methods Summary

Detailed entity class descriptions can be found in the BrainStem Reference[3]. A summary of MTM-Relay 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[0-3]	setConfiguration	
	getConfiguration	
	setState	
	getState	
i2c[0]	write	
	read	
relay[0-3]	setEnable	
	getEnable	
	getVoltage	
store[0-1]	getSlotState	
	loadSlot	
	unloadSlot	
	slotEnable	
	slotDisable	
	slotCapacity	
	slotSize	
system[0]	save	
	reset	
	setLED	
	getLED	
	setBootSlot	
	getBootSlot	
	getInputVoltage	
	getVersion	
	getModuleBaseAddress	
	getModuleSoftwareOffset	
	setModuleSoftwareOffset	
	getModuleHardwareOffset	
	setHBInterval	
	getHBInterval	
	getRouterAddressSetting	
	getModule	
	getSerialNumber	
	setRouter	
	getRouter	
	getModel	
	routeToMe	
timer[0-7]	getExpiration	
	setExpiration	
	getMode	
	setMode	
Pointer[0-3]	getOffset	
	setOffset	
	getMode	
	setMode	

Formatted Table

Entity Class	Entity Option	Variable(s) Notes
	getTransferStore	
	setTransferStore	
	initiateTransferToStore	
	initiateTransferFromStore	
	getChar	
	setChar	
	getShort	
	setShort	
	getInt	
	setInt	
App[0-3]	execute	

Table 12: Supported MTM-Relay BrainStem Entity API Methods[3]

LED Indicators

The MTM-Relay board has a number of LED indicators to assist with MTM system development, debugging, and monitoring. These LEDs are shown in the diagrams below.

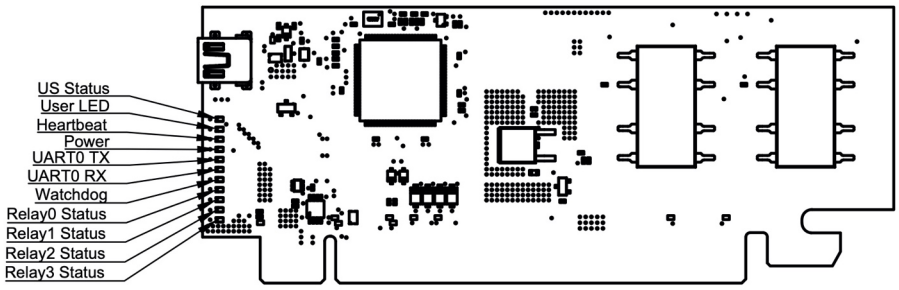


Figure 3: MTM-Relay LED Indicators

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

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 13: PCI-Express Edge Connectors for MTM Products, and equivalent connectors are offered from a multitude of vendors.

Manufacturer	Manufacturer Part Number	Description
Amphenol FCI Samtec	10018784-10203TLF PCIE-164-02-F-D-TH	PCI-Express 164-position vertical connector
Amphenol FCI	10042618-003LF	PCI-Express Retention Clip (optional)

Table 13: PCI-Express Edge Connectors for MTM Products

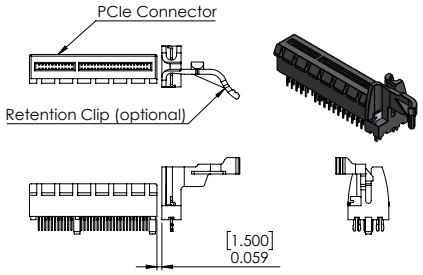


Figure 4: PCIe Vertical Connector with optional Retention Clip

MTM Edge Connector Specifications	Description
Contact Finish	Gold
Card Thickness	0.0625" [1.59mm]
Number of Rows	2
Number of Positions	164
Pitch	0.039" (1.00mm)

Table 14: MTM Edge Connector Specifications

Mechanical

Dimensions are shown in mm. 3D CAD models are available through the MTM-Relay product page's Downloads section. A 3D CAD viewer with many different CAD model formats available for download is available at https://a360.co/3erLYT3

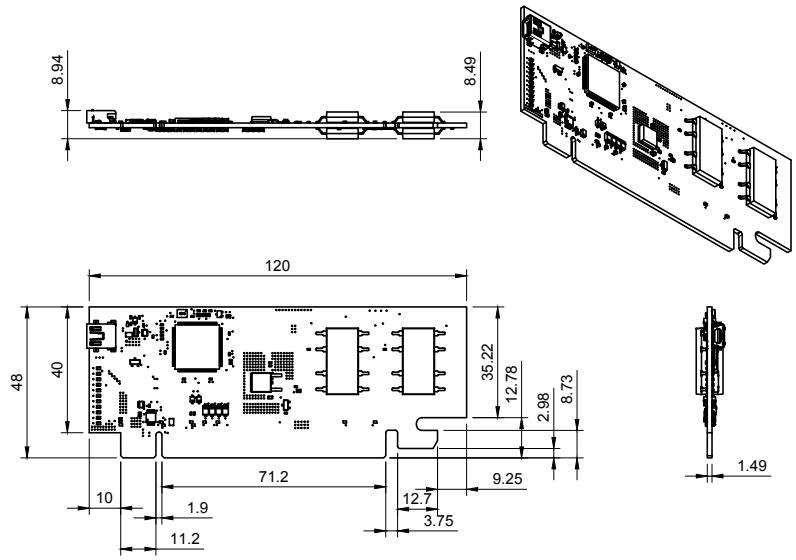


Figure 5: MTM-Relay Mechanical

Document Revision History

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

Revision	Date	Engineer	Description
1.0	April 2016	JTD	Initial Revision
1.1	May 2016	JTD	Fixed pin mapping
1.2	September 2016	RMN	Formatting, Error checking, updates
1.2.1	October 2016	LCD	Update Overview, Features, Description, added DO jitter
1.3	December 2016	JG	Clarified I2C pull-ups; update supported API calls
1.31	September 2018	LCD	Diagram corrections and application update
1.4	October 2018	LCD	Removed getVoltage() function
1.5	June 2020	ACRO	Formatting, links, entity updates, and performance characteristics
1.6	July 2020	ACRO	Updated Software Control section and updated table numbers

Table 15: Document Revision History