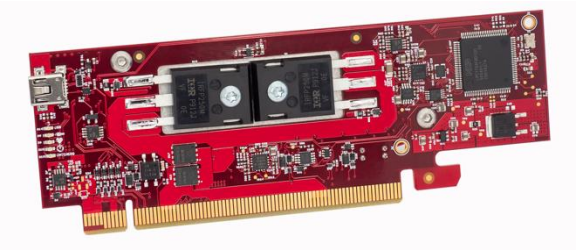


Overview

The Acroname® MTM DC Load Module (MTM-Load-1), part of Acroname's MTM (Manufacturing Test Module) product series, is a software-controlled electronic load module with voltage, current, power, and resistance control. The MTM-Load-1 allows MTM system designers to easily and modularly add power supply loading to their test system designs.

The MTM-Load-1 provides a stable, consistent and robust way to load a wide range of devices, optimized for constant current applications.

Built using Acroname's industry-proven and well-adopted BrainStem® technology, resources on the MTM-Load-1 are controlled via Acroname's powerful and extensible BrainStem software APIs.

Typical applications include:

- Manufacturing functional testing
- Validation testing
- Automated test development
- Motor driver testing
- Battery testing

Features

- 1 Fully software-controlled regulated input with ~100µV/~100µA resolution, up to a 30V/10A current limit (Rail0)
- 1 BrainStem I²C FM+ (1Mbit/s) bus
- 4 overvoltage, short-circuit and overcurrent protected digital GPIOs
- Remote/Kelvin sense (Rail0)
- Rail output current voltage-mirror (Rail0)
- Rail output enable indicator (Rail0)
- On-board local temperature sensor for monitoring and overtemperature shutdown

Description

The MTM-Load-1 module is a key component for manufacturing test and R&D of devices requiring precision voltage measurements. For more information on the MTM platform architecture, please refer to https://acroname.com.

The MTM-Load-1 implements an on-board 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.

The MTM-Load-1 provides a main power input rail. The rail is a fully regulated input designed for precise DC loading applications for a wide input voltage (0 to 30V) and current (0 to 10A) range. The voltage and current measurements are captured by a 24bit ADC to allow for high resolution (100µV and 100µA) over the full range of the load.

Within the MTM platform architecture, the MTM-Load-1 module can operate either independently or as a component in a larger network of MTM modules. Each MTM-Load-1 is uniquely addressable and controllable from a host by connecting via the on-board 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 on-board controller and access to the available resources in the MTM-Load-1. The MTM-Load-1 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-Load-1, like all MTM modules, 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
Rail0 Voltage	-40.0	40.0	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

Table 1: Absolute Voltage Ratings

Current Rating	Minimum	Maximum	Units
Input Current, I_{supply}	0.0	2.0	A
Rail0 Current	-12.0	12.0	A

Table 2: Absolute 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
Ambient Operating Temperature, T_A	Non-Condensing	0.0	25.0	70.0	°C
Relative Humidity Range	Non-Condensing	5	-	95	%RH
Storage Temperature, T_{STG}		-10.0	-	85.0	°C
Electrostatic Discharge, V_{ESD}	IEC 61000-4-2, level 4, contact discharge to edge connector interface	0.0	-	±8000	V

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
Input Voltage, V_{supply}	V_{supply} below 12V will reduce max Rail0 power	6.0	12.0	12.5	V
Rail0 Voltage		0.0	-	30.0	V
Rail0 Current		0.0	-	10.0	A
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 4: Recommended Operating Ratings

Block Diagram

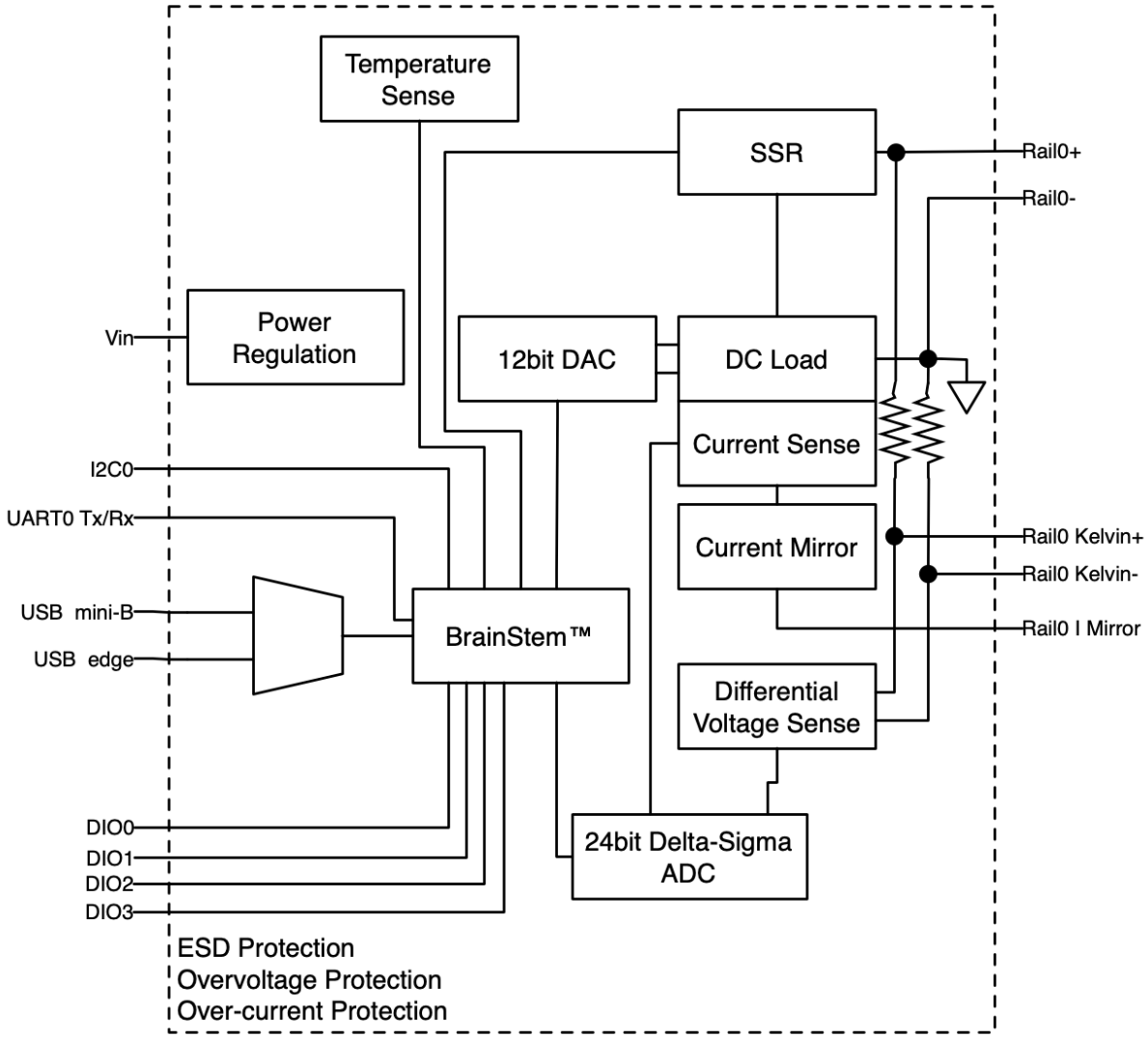


Figure 1: MTM-Load-1 Block Diagram

Typical Performance Characteristics

Specifications are valid at 25°C and V_{supply} = 12V unless otherwise noted. Indoor application only. Sample rates are typically limited by the USB throughput of the host operating system except where bulk capture is supported.

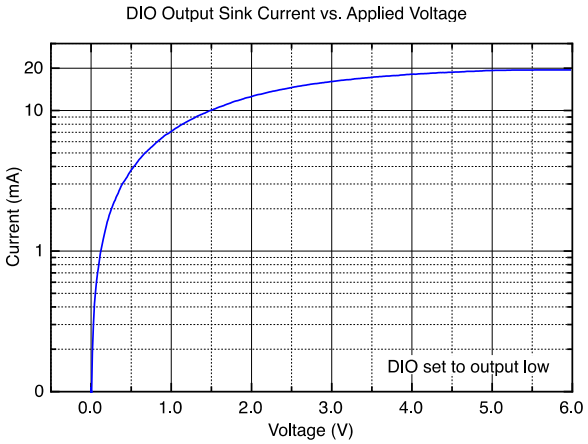
Parameter	Conditions/Notes	Minimum	Typical	Maximum	Units
Base Current Consumption, I_{supply}	Input voltage = 6V Input voltage = 12V	- -	92 95	- -	mA
Rail0 On Current Consumption, I_{supply}	Input voltage = 6V Input voltage = 12V	- -	170 210	- -	mA
Rail0 Input Voltage, V_{RAIL0}	Software controlled	0.0	-	30.0	V
Rail0 Input Voltage Limit Range		-0.7	-	35.0	V
Rail0 Input Current, I_{RAIL0}	Software controlled	0.0	-	10.0	A
Rail0 Input Current Limit Range		-1.0	-	12.0	A
Rail0 Temperature Measurement Resolution		-	0.068	-	°C
Rail0 Thermal Shutdown		-	-	85	°C
Rail0 Thermal Shutdown Hysteresis	After thermal shutdown event	15.0	-	-	°C
Rail0 Voltage Measurement Resolution		-	1	-	µV
Rail0 Current Measurement Resolution		-	1	-	µA
Rail0 Voltage Control Resolution		-	100	-	µV
Rail0 Current Control Resolution		-	100	-	µA
Rail0 Voltage Accuracy		-0.025	-	0.025	%FSR
Rail0 Current Accuracy		-0.080	-	0.080	%FSR
Rail0 Voltage Temperature Drift	Measurement change relative to Rail0 temperature change	-1.5	-	1.5	mV/°C
Rail0 Current Temperature Drift		-1.0	-	1.0	mA/°C
Rail0 Current Voltage-Mirror[1]			0.5		V/A
Rail0 Current Mirror Zero-offset		-20	-	20	mV
Rail0 Current Mirror Gain Error		-2	-	2	%
Rail0 Peak Power		-	-	150	Watts
Rail0 Continuous Power	@25°C Ambient	-	50	-	Watts
Rail0 Minimum Resistance		-	-	150	mΩ
Rail0 Slew Rate	10% to 90% of command	-	0.6	-	A/µs
Reset Low Threshold		-	1.2	-	V
I2C SDA, SCL Pins		0.0	3.3	5.0	V
Digital Output V_{HI}		-	3.3	-	V
Digital Input Logic High, V_{IH}		2.15	-	-	V
Digital Input Logic Low, V_{IL}		-	-	1.1	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	V_{supply} on pin	-	Infinite	-	hours
Digital Output Sink Current		-	-	-20.0	mA
Digital Output Source Current	<10% voltage drop (V_{output} >= 2.97V)	-	-	3.15	mA

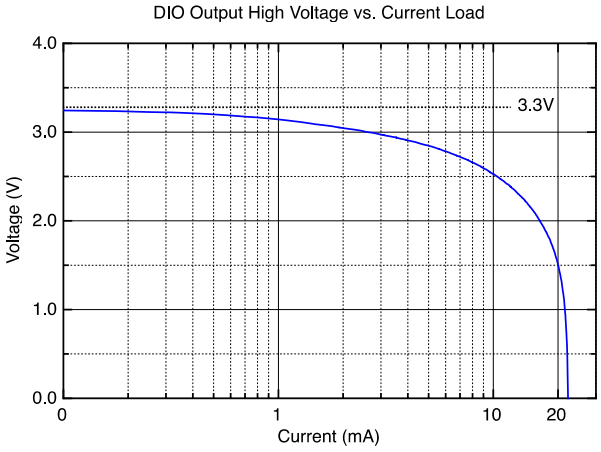
[1] Current output voltage-mirror must be connected to high impedance

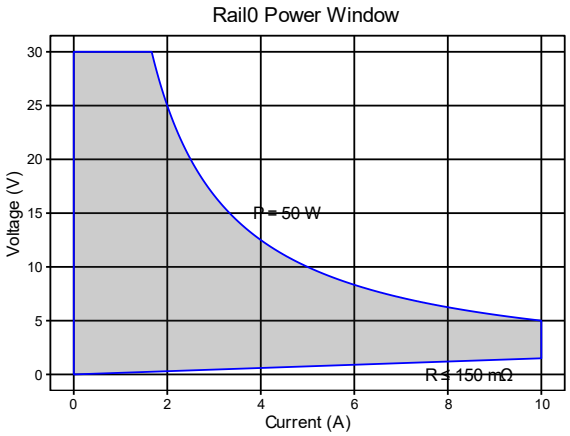
Parameter	Conditions/Notes	Minimum	Typical	Maximum	Units
Digital Sample Rate[2]	via USB link, C++ Reflex	- -	1000 8200	- -	Hz Hz
Digital Input Resistance	Configuration mode set to both Input and High-Z	-	4.25	4.45	MΩ
Digital Input Leakage Current	Configuration mode set to both Input and High-Z	-	110	-	µA

Table 5: Typical Performance Characteristics

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





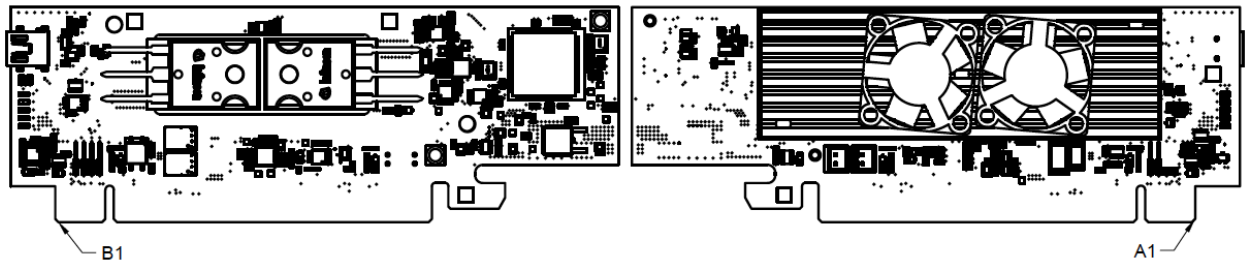


Step response available soon.

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, 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	I^2C0 SCL	8	GND
9	I^2C0 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

Pins Specific to MTM-Load-1

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	RAIL0 Kelvin Sense Negative Return	18	RAIL0 Kelvin Sense Positive Return
19	GND (RAIL0 -)	19	RAIL0 +
20	GND (RAIL0 -)	20	RAIL0 +
21	GND (RAIL0 -)	21	RAIL0 +
22	GND (RAIL0 -)	22	RAIL0 +
23	Digital IO 1	23	Digital IO 0
24	RAIL0 Current Mirror	24	RAIL0 Power Enable Status
25:29	Reserved, Do Not Connect	25:29	Reserved, Do Not Connect
30	Digital IO 3	30	Digital IO 2
31:57	Reserved, Do Not Connect	31:57	Reserved, Do Not Connect
58	GND (RAIL0 -)	58	RAIL0 +
59	GND (RAIL0 -)	59	RAIL0 +
60	GND (RAIL0 -)	60	RAIL0 +
61	GND (RAIL0 -)	61	RAIL0 +
62	GND (RAIL0 -)	62	RAIL0 +
63:82	Reserved, Do Not Connect	63:82	Reserved, Do Not Connect

Heatsink Warnings

⚠

Because the heatsink is used to dissipate heat generated by Rail0, it will get very hot during operation. Users should avoid touching it while the MTM-Load-1 is in use to avoid burns.

The heatsink is electrically connected to RAIL0+. Proper care should be taken to isolate or insulate the heatsink from different electrical potentials.

Module Hardware and Software Default Values

The MTM-Load-1 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 6 details the BrainStem API entities and macros used to interface with the MTM-Load-1 module. For C and C++ developers, these macros are defined in `aMTMLoad1.h` from the BrainStem development package. For Python development, the module `MTMLOAD1` class defines the extent of each entity array.

While Table 6 lists the BrainStem API entities available for this module, not all entity methods are supported by the MTM-Load-1. For a complete list of supported entity methods, see Table 8. 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[4].

Parameter	Index	Macro Name or Implemented Options	Notes
Module Definitions:			
Module Base Address	14	`aMTMLOAD1_MODULE _ADDRESS`	See `aMTMLoad1.h`
Entity Class Definitions:			
`rail` Entity Quantity	1	`aMTMLOAD1_NUM_RAILS`	
`Temperature` Entity Quantity	1	`aMTMLOAD1_NUM_TEMPERATURES`	
`digital` Entity Quantity	4	`aMTMLOAD1_NUM_DIGITALS`	
`i2c` Entity Quantity	1	`aMTMLOAD1_NUM_I2C`	
`Store` Entity Quantity	2	`aMTMLOAD1_NUM_STORES`	
`system` Entity Quantity	1		
`timer` Entity Quantity	8	`aMTMLOAD1_NUM_TIMERS`	
`app` Entity Quantity	4	`aMTMLOAD1_NUM_APPS`	
`pointer` Entity Quantity	4	`aMTMLOAD1_NUM_POINTERS`	

Table 6: MTM-Load-1 Hardware and Software Default Values[3]

[3] Refer to `aMTMLoad1.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-LOAD-1. 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-LOAD-1 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-LOAD-1 is attached to a host machine, a user can connect to it via software API:

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

The MTM-LOAD-1 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-LOAD-1 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 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 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[4].

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	14	14
NC	NC	NC	1	2	14	16
NC	NC	1	NC	4	14	18
NC	1	NC	NC	8	14	22
1	NC	NC	NC	16	14	30
1	NC	NC	1	2+16	14	32

Table 7: Module Address Hardware Offset Examples

Upstream USB Connectivity Options

The MTM-Load-1 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, 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-Load-1 away from the factory default settings. Saving system settings preserves the settings to become the new default. Most changes to system settings require a save and reboot before taking effect. Use the following command to save changes to system settings before reboot:

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

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

Store Entities

Every BrainStem module includes several Store entities and onboard memory slots to load Reflex files (for details on Reflex, see BrainStem Reference[4]). 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.

Rail Entities

Rail 0 on the MTM-Load-1 module is powerful (no pun intended); it allows other devices and peripherals to provide power to the

MTM-Load-1 module where it is precisely loaded. The rail is a software-adjustable constant current sink. This rail is accessed through a BrainStem rail class entity. The MTM-Load-1 module implements a subset of the BrainStem rail class for the load rail. Table 8: Supported MTM-Load-1 BrainStem Entity API Methods summarizes the implemented rail entity options.

The rail can be switched on or off using the `setEnable` API:

```
stem.rail[0].setEnable(1)
```

RAIL0 Current Setting

The current setpoint for the rail can be configured in software from 0A to 10A. Setting values outside the allowable range will return an error (`aErrRange 13`). The rail will attempt to maintain the specified current through all input voltage variations once the rail is enabled with the operational mode set to constant current.

```
stem.rail[0].setCurrentSetpoint(microvolts)
stem.rail[0].getCurrentSetpoint(microvolts)
```

RAIL0 Current Limit Setting

The current limit for the rail can be configured in software from 0A to 12A. The rail will operate normally if the measured current is below the specified current. If the limit is crossed, the load will automatically disable the rail and set the corresponding overcurrent fault bit in the Operational State variable. If the current limit is below the current setpoint the rail will still disable itself when the current limit is exceeded.

```
stem.rail[0].setCurrentLimit(microamps)
```

RAIL0 Voltage Limit Setting (Min/Max)

The voltage limits for the rail can be configured in software from -0.7V to 35V. The rail will operate normally between the minimum and maximum voltage limits. If the upper or lower limit is crossed, the load will automatically disable the rail and set the corresponding over/under voltage fault bit in the Operational State variable. Setting values outside the allowable range will return an error (`aErrRange 13`).

```
stem.rail[0].setVoltageMinLimit(microvolts)
stem.rail[0].setVoltageMaxLimit(microvolts)
```

The voltage minimum limit can be conveniently used for battery discharging to prevent the load from over drawing the battery.

RAIL0 Power Limit Setting

The power limit for the rail can be configured in software from 0W to 150W. The rail will operate normally below this limit. If the limit is crossed, the load will automatically disable the rail and set the corresponding overpower fault bit in the Operational State variable. Setting values outside the allowable range will return an error (`aErrRange 13`).

```
stem.rail[0].setPowerLimit(milliwatts)
```

RAIL0 Operational Mode

The rail operational mode is a bit field combination of the hardware configuration and the operational mode.

Bits	Descriptions
0:3	Hardware Configuration
4:7	Operational Mode

The hardware configuration nibble has two valid values, 0 (auto) or 1 (linear). These two configurations are equivalent. Attempts to set any other hardware configuration values will result in an error (`aErrConfiguration 17`).

The rail has one valid operational mode. Attempts to set any other operational mode will result in an error (`aErrUnimplemented 21`).

Mode	RAIL Operational Mode Description
0	Constant Current (railOperationalModeConstantCurrent)

The operational mode can be configured through the API:

```
mode = railOperationalModeConstantCurrent << 4

stem.rail[0].setOperationalMode(mode)
stem.rail[0].getOperationalMode(mode)
```

RAIL0 Temperature Limit

The rail also features over-temperature protection. An over-temperature condition occurs when the board temperature crosses a fixed safe operating threshold (85°C). The rail is automatically disabled and cannot be enabled until the temperature goes below the hysteresis set point (85°C - 15°C → 70°C). Clear faults with the `clearFaults()` method via the software API.

The MTM-Load-1 utilizes fans for active cooling. The fans run directly off V_{supply}, and the air flow decreases with decreasing V_{supply}. To achieve the longest possible rail operation without reaching the over-temperature limit, use V_{supply} = 12V.

RAIL0 Operational State

The rail operational state will give all the details about the current status of the rail entity.

```
stem.rail[0].getOperationalState(state)
```

The value *state* is a 32-bit value, defined by the following bit fields:

Bits	RAIL Operational State Description
0	Initializing (railOperationalState_Initializing)
1	Enabled (railOperationalState_Enabled)

2	Fault (railOperationalState_Fault)
3-7	Reserved
8-15	Hardware Configuration (railOperationalState_HardwareConfiguration)
16	Overvoltage Fault “OV” (railOperationalStateOverVoltageFault)
17	Undervoltage Fault “UV” (railOperationalStateUnderVoltageFault)
18	Overcurrent Fault “OC” (railOperationalStateOverCurrentFault)
19	Overpower Fault “OP” (railOperationalStateOverPowerFault)
20	Reverse Polarity Fault “RV” (railOperationalStateReversePolarityFault)
21	Overtemperature Fault “OT” (railOperationalStateOverTemperatureFault)
22-23	Reserved
24-31	Operating Mode (railOperationalStateOperatingMode)

On startup, the Initializing bit will be set. Once the system is ready to load, the bit will be cleared, typical time is less than 1 second.

The Enabled bit represents the actual state of the rail and will be set whenever the rail is active and cleared whenever the rail is inactive.

The Fault bit will be set whenever the rail has experienced a fault condition. Along with the Fault bit, a bit(s) will be set in the Fault bits region of the state variable (bits 16-21). Clear faults with the clearFaults() method via the software API.

The Hardware Configuration bit field will represent the current hardware configuration of the rail entity. The MTM-Load-1 will only ever report being in linear mode.

Mode Enum	RAIL Hardware Configuration Description
0	Linear (railOperationalStateLinear)

Fault Bits:

- Overvoltage fault: the max voltage limit has been broken.
- Undervoltage fault: the min voltage limit has been broken.
- Overcurrent fault: the current limit has been exceeded.
- Overpower fault: the power limit has been exceeded.
- Reverse polarity fault: the hardware RVP protection has been triggered due to a reverse voltage detected on the rail pins.
- Overtemperature fault: the temperature limit of the rail (85°C) has been exceeded.

The Operating Mode bit field reflects the control mode.

Mode Enum	RAIL Operating Modes Description
0	Constant Current (railOperationalStateConstantCurrent)

RAIL0 Temperature

RAIL0’s subsystem power stage temperature can be monitored above the load circuitry. Reading this value is possible through the API.

```
stem.rail[0].getTemperature(temperature)
```

Temperature monitoring is also used internally to prevent the power regulation stage from over-heating and preserving the power stage. If an over-temperature condition occurs, the MTM-Load-1 module will disable load circuitry and disconnect the rail until the user cycles the rail enable.

RAIL0 Kelvin Sensing

The MTM-Load-1 provides a “4-wire” Kelvin interface to enable accurate voltage measurements. The kelvin connections are always enabled and can be connected externally to eliminate wire drop for improved voltage accuracy during high-current operation. If NOT connected externally, the MTM-Load-1 includes 1 kΩ connections on the board which will provide local voltage measurements without compensating for any wire drop.

Digital Entities

The MTM-Load-1 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:

- 0 (digitalConfigurationInput)
- 1 (digitalConfigurationOutput)
- 4 (digitalConfigurationHiZ)

If a digital pin is configured as output mode, setting the digital logic level:

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

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

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

If a digital pin is configured in HighZ 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	Hi-Z	RCServo	Signal
DIO0	Yes	Yes	Yes	-	-
DIO1	Yes	Yes	Yes	-	-
DIO2	Yes	Yes	Yes	-	-
DIO3	Yes	Yes	Yes	-	-

I²C Entities

The MTM-Load-1 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-Load-1 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)
```

Temperature Entities

The MTM-Load-1 has on board temperature sensor located by the load circuitry to keep track of the temperature of the board and load circuitry. The temperature value can be read using the following function:

```
stem.temperature.getTemperature(temp)
```

Please note that this temperature sensor is the RAIL0 temperature sensor. There is only one temperature sensor on the MTM-Load-1.

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-Load-1 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-Load-1 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-Load-1 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-Load-1 Supported Entity Methods Summary

Detailed entity class descriptions can be found in the BrainStem Reference[4]. A summary of MTM-Load-1 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	
	setStateAll	
	getStateAll	
i2c[0]	setSpeed	
	getSpeed	
	write	
	read	
	setPullup	Disabled by default. I2C communication requires a single set of pull-ups enabled across the bus.
rail[0]	getCurrent	
	setCurrentSetpoint	
	getCurrentSetpoint	
	setCurrentLimit	
	getCurrentLimit	
	getTemperature	
	setEnable	
	getEnable	
	getVoltage	
	setVoltageMinLimit	
	getVoltageMinLimit	
	setVoltageMaxLimit	
	getVoltageMaxLimit	
	getPower	
	setPowerLimit	
	getPowerLimit	
	getResistance	
	setOperationalMode	
	getOperationalMode	
	getOperationalState	
	setKelvinSensingEnable	Kelvin Sensing is always enabled and cannot be disabled.
	getKelvinSensingState	
	clearFaults	
store[0-1]	getSlotState	
	loadSlot	
	unloadSlot	
	slotEnable	
	slotDisable	
	getSlotCapacity	
	getSlotSize	
system[0]	reset	
	save	

Entity Class	Entity Option	Variable(s) Notes
	setLED	
	getLED	
	setBootSlot	
	getBootSlot	
	getInputVoltage	
	getVersion	
	getModuleBaseAddress	
	getModuleSoftwareOffset	
	setModuleSoftwareOffset	
	setHBInterval	
	getHBInterval	
	getRouterAddressSetting	
	getModule	
	getSerialNumber	
	setRouter	
	getRouter	
	getModel	
temperature[0]	getTemperature	
timer[0-8]	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 8: Supported MTM-Load-1 BrainStem Entity API Methods[4]

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

LED Indicators

The MTM-Load-1 board has five LED indicators to assist with MTM system development, debugging, and monitoring. These LEDs are shown in the diagrams below.

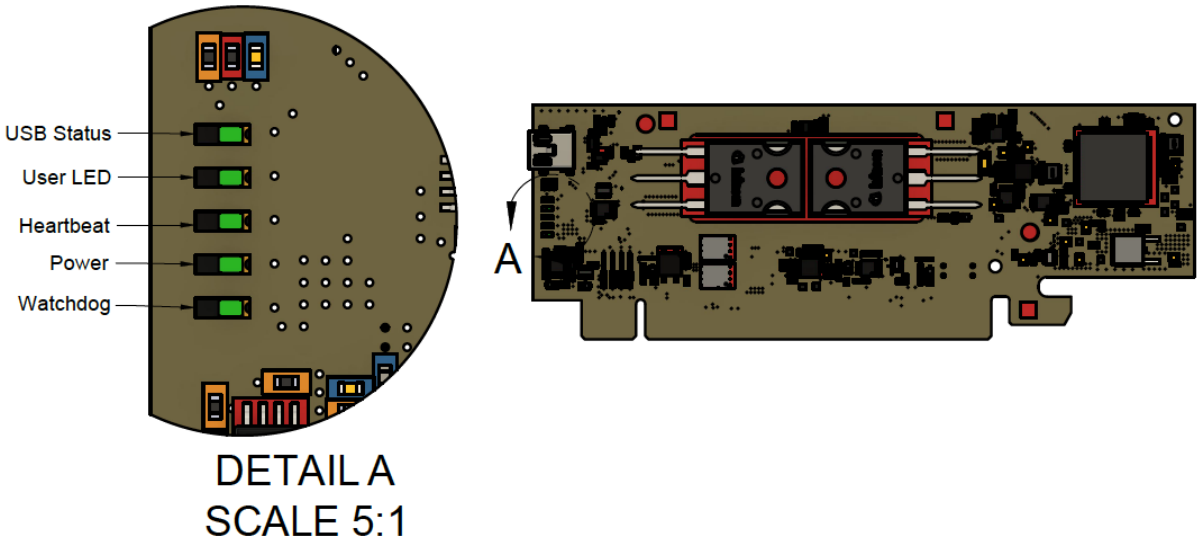


Figure 2: MTM-Load-1 LED Indicators

Application Examples

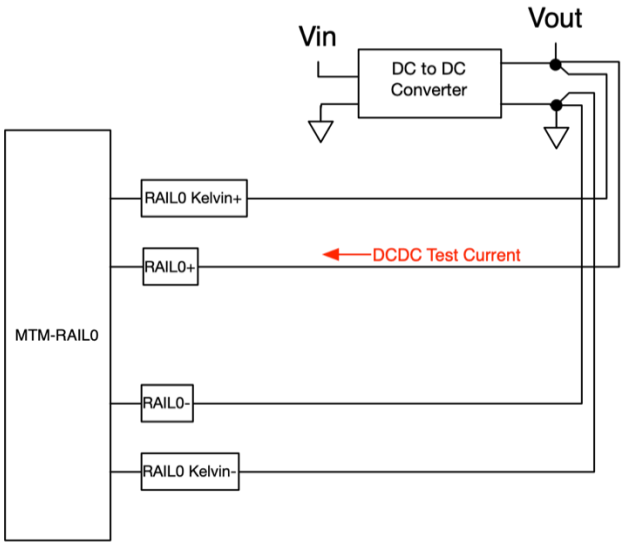


Figure 3 Loading a DC to DC Converter for verifying output-current-ability.

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 shown in Table 9, 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 9: PCI-Express Edge Connectors for MTM Products

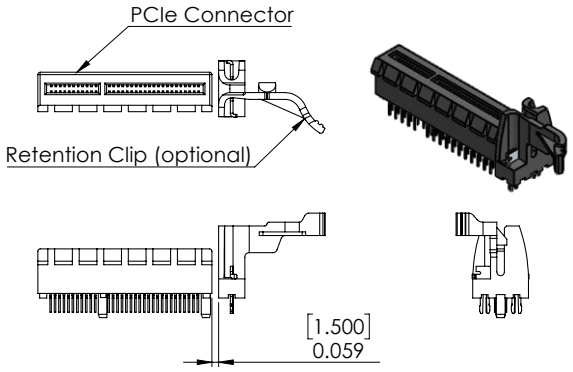


Figure 4: PCIe Vertical Connector with optional Retention Clip

MTM Edge Connector Specifications	Description
Contact Finish	Gold
Card Thickness	0.0625” [1.59mm]
Number of Rows	2
Number of Positions	Variable (see Table 9: PCI-Express Edge Connectors for MTM Products)
Pitch	0.039” (1.00mm)

Table 10: MTM Edge Connector Specifications

Mechanical

Dimensions are shown in inches [mm]. 3D CAD models are available through the MTM-Load-1 product page's Downloads section. A 3D CAD viewer with many different CAD model formats available for download is available at https://a360.co/2AwLNH4.

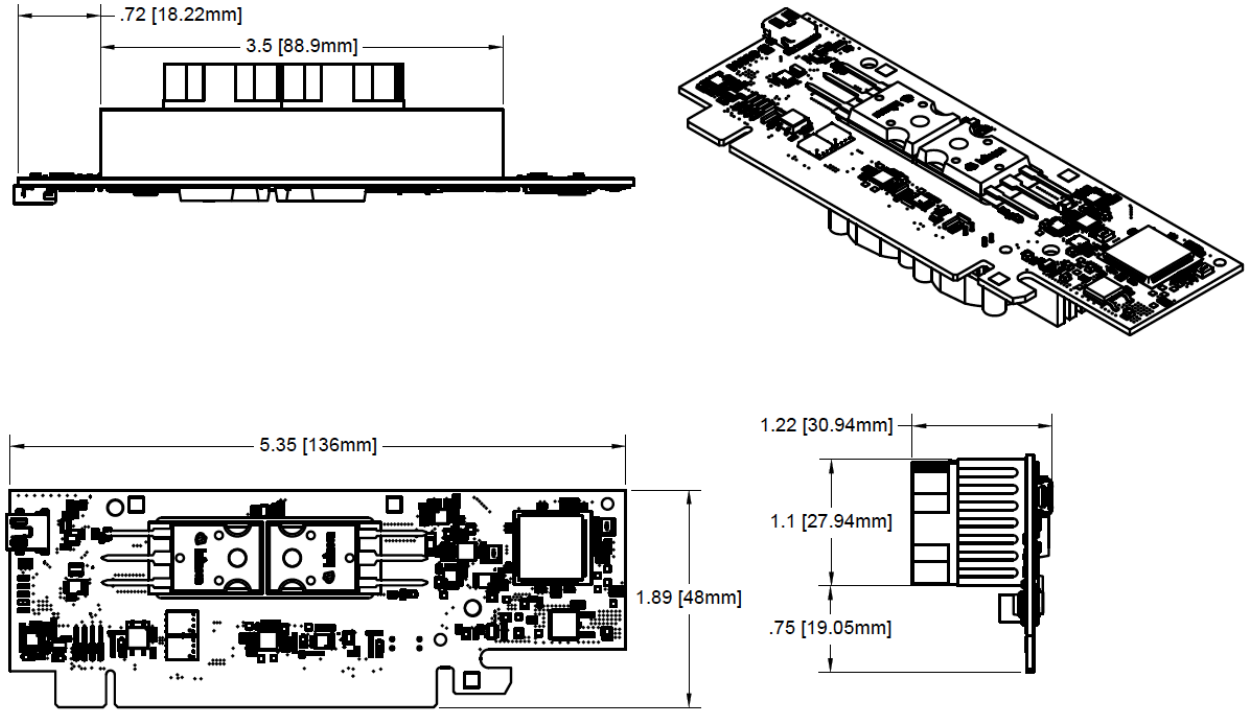


Figure 5: MTM-Load-1 Mechanical

Product Support

Questions about the product operation or specifications are welcome through Acroname's contact portals. Software downloads, reference API and application examples are available online at:

https://acroname.com/support

Direct communication and additional technical support are available at:

https://acroname.com/contact-us

2741 Mapleton Avenue

Boulder, CO, USA 80304-3837

720-564-0373 (phone)

Document Revision History

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

Revision	Date	Engineer	Description
1.0	October 2019	GCF	Initial Release
1.1	Feb 2020	JG	Update product naming
1.2	June 2020	ACRO	Formatting, entity updates, diagram updates
1.3	September 2020	TDH	Updates for official product release
1.4	February 2021	MJK	Contact information for technical support.
1.5	January 2022	TDH	Typical Characteristics updates