Overview

The Acroname MTM Power Module (MTM-PM-1) is a single-channel, software-controlled voltage and current limiting modular power supply designed for powering devices during manufacturing or R&D testing. It can provide stable, consistent and robust power to a wide range of devices and is optimized for devices using LiPo or similar batteries. In particular, the MTM-PM-1's ultra fast transient response is ideal for devices with cellular radios (GSM, UMTS, LTE, CDMA, etc.). All features, including accurate voltage, temperature and current measurements, are controlled by Acroname's proven, extensible and well-adopted BrainStem® technology and software API which is accessible via USB 2.0 (full-speed) or the BrainStem bus.

Features

– 1 fully software-controllable rail with ~1mV resolution up to a 3A current limit
– 1 pass-through voltage (from V_{supply}), adjustable current limit, power supply rail with software enable control
– 2 overvoltage and current protected digital GPIO
– 1 Rail output current voltage-mirror
– 1 Rail output enable indicator
– 1 BrainStem I²C FM+ (1Mbit/s) bus

Description

As part of Acroname's MTM (Manufacturing Test Module) series, the MTM-PM-1 is a key component for manufacturing test and R&D of devices requiring highly accurate, controllable power supply and measurement. For noise sensitive applications such as sensitivity testing of radios, the MTM-PM-1 can be operated in a fully linear power supply mode; no potentially RF-noisy switchers. When thermal and efficiency considerations are more important, the S65-MTM-PM-1 can utilize a first-stage, pre-conditioning switch mode power supply (SMPS) followed by the high-accuracy, ultra fast response linear stage. Regardless of the application requirements, the MTM-PM-1 provides stable and accurate power, even under transient loads up to 3A, and line transients within its operating input voltage range.
The MTM-PM-1 has an optional K-type thermocouple interface, allowing for remote temperature measurements from -25° to +400°C. This remote temperature measurement is intended to monitor the temperature of the device under test or other test equipment. There is a standard on-board temperature sensor for monitoring MTM-PM-1 and over-temperature shutdown.
The MTM-PM-1 can be utilized independently or as an integrated part of a BrainStem network. Simple and convenient networking of multiple modules, in addition to the USB interface on the MTM-PM-1, makes connecting to and controlling the power supply in a variety of ways quick and easy. The powerful BrainStem APIs in C++ and Python makes development simple and portable. The modularity of the device and robust MTM design means less downtime and maintenance on manufacturing lines. Finally, scaling throughput from prototype-stage to mass production is simply plugging in additional modules.

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.

Parameter	Minimum	Maximum	Units
Input Voltage, V_{supply}	6.0	26.0	V
V_{supply} current	0.0	14.0	A
Voltage to any IO pin	-0.5	12.5	V
Voltage to any I2C pin	0.0	5.5	V

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	0.0	25.0	70.0	°C
Storage Temperature, T_{STG}		-10.0	-	85.0	°C
Electrostatic Discharge, V_{ESD}	IEC 61000-4-2, level 4, contact discharge	0.0	-	±8000	V

Table 2: Handling Ratings

Recommended Operating Ratings

Values presented apply to the full operating temperature range.

Parameter	Conditions/Notes	Minimum	Typical	Maximum	Units
Input Voltage, V_{supply}		6.0	-	12.0	V
Current Draw, I_{supply}		0.125	-	6.2	A
Reset Low Threshold		-	1.2	-	V
I2C SDA, SCL Pins		0.0	3.3	5.5	V
UART Tx/Rx Logic High, V_{IH}		2.15	-	-	V
UART Tx/Rx Logic Low, V_{IL}		-	-	1.1	V
UART, Digital Input Output		-	3.3	-	V
Digital Input Logic High, V_{IH}		2.15	-	-	V
Digital Input Logic Low, V_{IL}		-	-	1.1	V

`rail0` Output Voltage, V_{rail0}	Software controlled	1.8	-	5.0	V
`rail0` Output Current Limit	Software controlled	0.05	-	3.0	A
`rail1` Output Voltage, V_{rail1}	V_{supply} pass-through	-	V_{supply}	-	V
`rail1` Output Current Limit	Software controlled	0.001	-	3.0	A
Digital Output Drive Current	Output high; short to GND Output high into 2.97V	- -	20.0 3.5	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

Table 3: Recommended Operating Ratings

Block Diagram

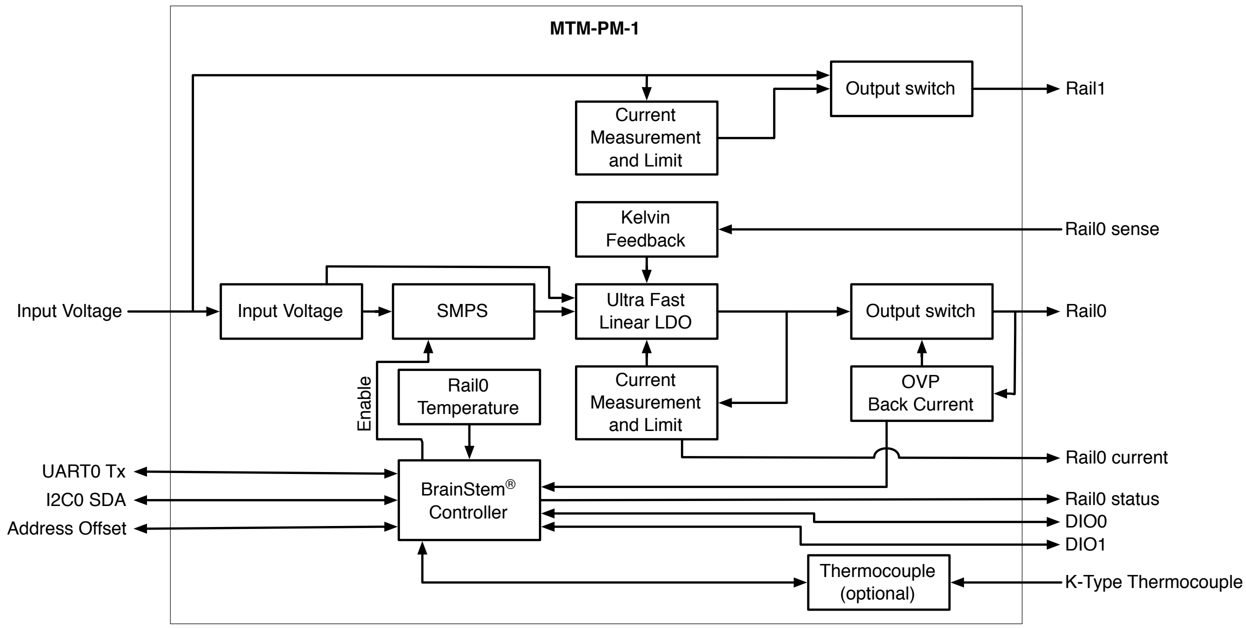
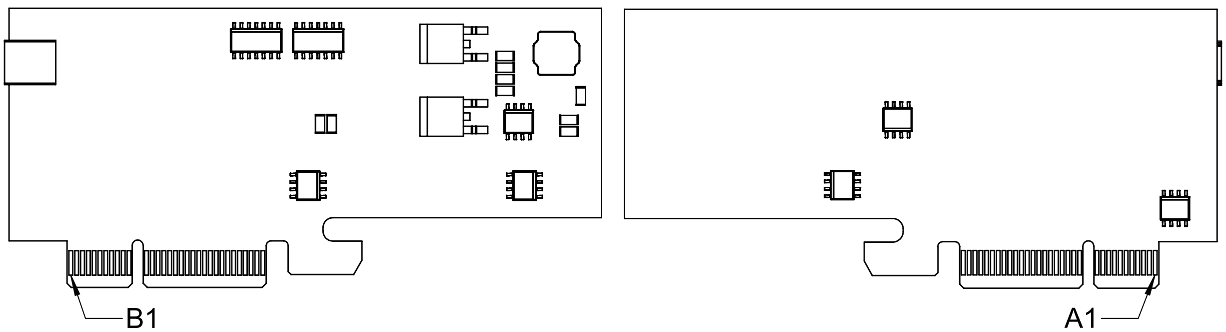


Figure 1: MTM-PM-1 Block Diagram

Pinout Descriptions

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

Edge Connector Side A	Edge Connector Side A Description	Edge Connector 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	GND	5	Input Voltage, V_{supply}
6	Reset	6	Reserved, Do Not Connect
7	GND	7	Reserved, Do Not Connect
8	I^2C SCL	8	GND
9	I^2C SDA	9	GND
10	GND	10	UART0 Transmit
11	GND	11	UART0 Receive
12	Module Address Offset 0	12	Module Address Offset 2
13	Module Address Offset 1	13	Module Address Offset 3

Pins Specific to MTM-PM-1

Edge Connector Side A	Edge Connector Side A Description	Edge Connector Side B	Edge Connector Side B Description
14	Reserved, Do Not Connect	14	USB Upstream Data +
15	K-Type Thermocouple +	15	USB Upstream Data -

16	K-Type Thermocouple -	16	Reserved, Do Not Connect
17	Reserved, Do Not Connect	17	Reserved, Do Not Connect
18	RAIL0 Kelvin Sense Negative Return[1]	18	RAIL0 Kelvin Sense Positive Return
19	GND	19	RAIL0 +
20	GND	20	RAIL0 +
21	GND	21	RAIL0 +
22	GND	22	RAIL0 +
23	Digital IO 1	23	Digital IO 0
24	RAIL0 Current Mirror (1V/1A)	24	RAIL0 Power Enable Status
25	Reserved, Do Not Connect	25	RAIL1 +
26	Reserved, Do Not Connect	26	RAIL1 +
27	Reserved, Do Not Connect	27	RAIL1 +
28	Reserved, Do Not Connect	28	RAIL1 +
29	Reserved, Do Not Connect	29	Reserved, Do Not Connect
30	Reserved, Do Not Connect	30	Reserved, Do Not Connect
31	Reserved, Do Not Connect	31	Reserved, Do Not Connect
32	Reserved, Do Not Connect	32	Reserved, Do Not Connect

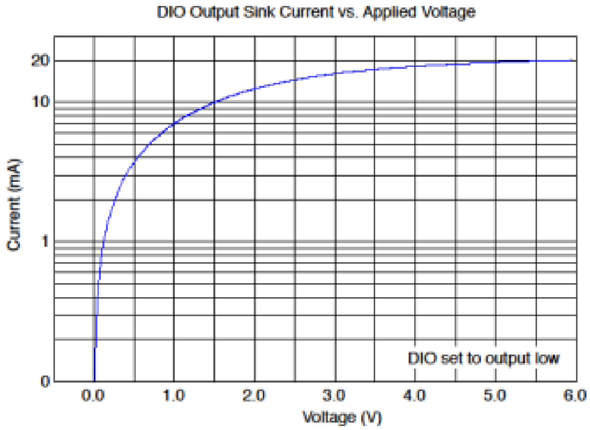
[1] RAIL0 Kelvin Sense Negative return is electrically connected to GND

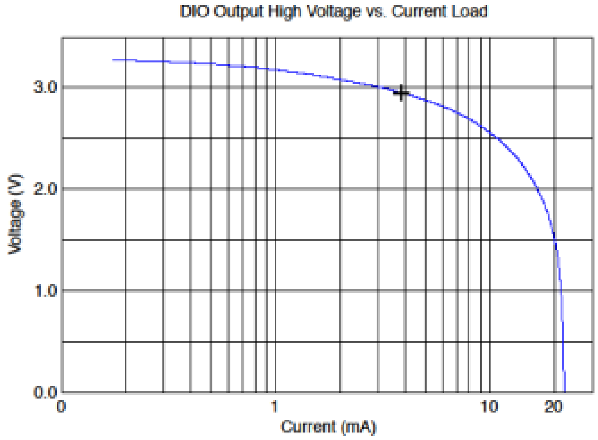
Typical Performance Characteristics

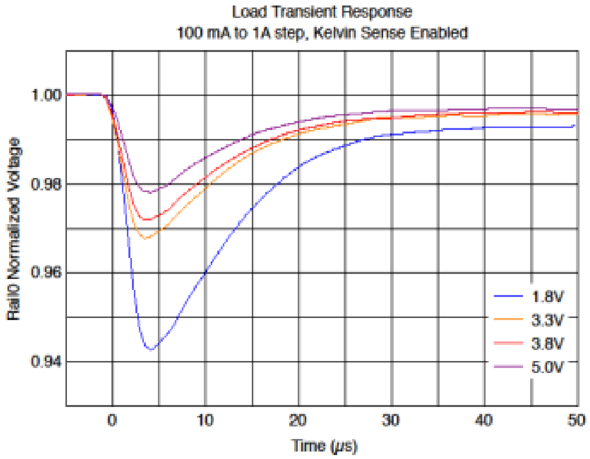
Typical Performance Characteristics are representative of typical performance at 25°C ambient with a 7.0V input supply and linear power supply operational mode unless otherwise noted.

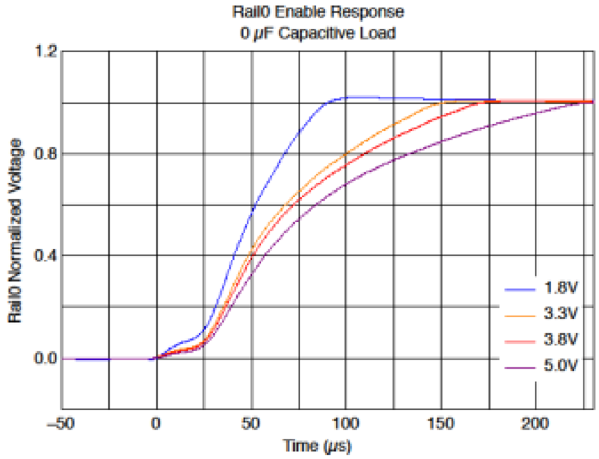
Parameter	Conditions/Notes	Minimum	Typical	Maximum	Units
`rail0` Voltage Accuracy	150mA load	-	0.5	-	%
`rail0` Voltage Output Ripple	150mA load	-	13	20	mV
`rail0` Voltage Control Resolution		-	1.5	-	mV
`rail0` Linear-mode Drop Out		-	0.8	1.2	V
`rail0` Switcher-mode Supply	V_{supply}	7.0	-	-	V
`rail0` Thermal Drift	Full temperature range	-	-	1.5	mV
`rail0` Current Voltage-Mirror[2]			1.0		V/A
`rail0` Current Mirror Zero-offset		-	15	20	mV
`rail0` Current Mirror Output Accuracy	$I_{out} \leq 500mA$ $I_{out} > 500mA$	- -	2 0.5	5 1	%
`rail0` Transient Load Response	1A step load at 5µs[3] 1A step load at 28µs[1] 2A step load at 28µs[1] 3A step load at 28µs[1]	- - - -	80 10 10 25	100 20 25 35	mV
`rail0` Enable Time[4], 0µF [5]	V_{rail} = 1.8V, limit=3A V_{rail} = 3.3V, limit=3A V_{rail} = 3.8V, limit=3A V_{rail} = 5.0V, limit=3A	- - - -	80 125 140 175	- - - -	µs
`rail0` Enable Time[2], 4.7uF [3]	V_{rail} = 1.8V, limit=3A V_{rail} = 3.3V, limit=3A V_{rail} = 3.8V, limit=3A V_{rail} = 5.0V, limit=3A	- - - -	110 165 175 230	- - - -	µs
`rail1` On resistance (R_{on})		-	70	-	mΩ
`rail1` Enable Time, 0 uF [3]		-	570	-	µs
`rail1` Enable Time, 4.7 uF [3]		-	590	-	µs
Current Limit Accuracy	Limit <250mA Limit ≥250mA	- -	25 10	- -	mA %

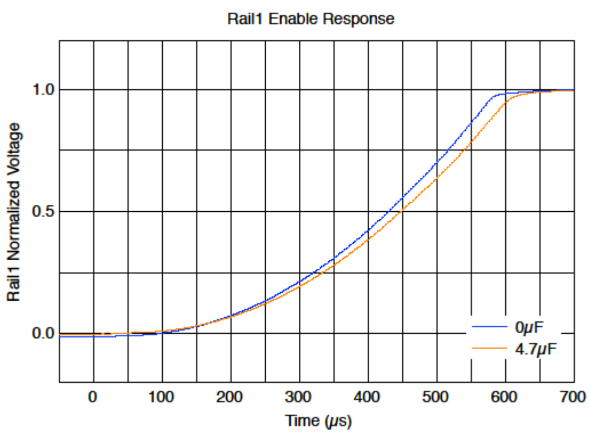
[2] Current output voltage-mirror must be connected to high impedance
[3] Kelvin sense connected and enabled; 5µs is maximum voltage drop due to step load; 28µs for 3GPP reference
[4] Rise time to within 90% of V_{rail}
[5] Enable time depends heavily on the operational load, so the maximum rise time will be application-dependent

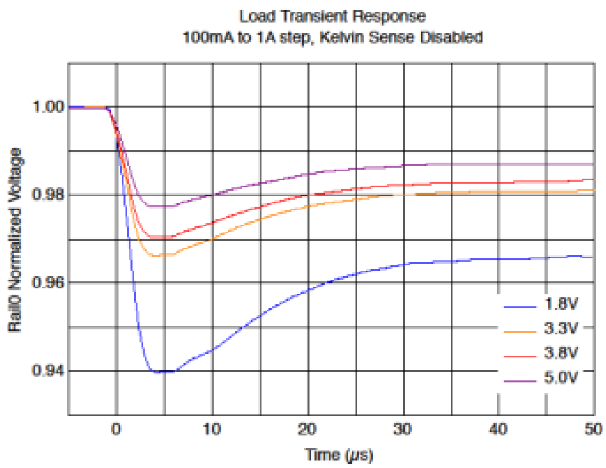

DIO Output Sink Current vs. Applied Voltage
Current (mA)
20
10
1
0
0.0
1.0
2.0
3.0
4.0
5.0
6.0
Voltage (V)
DIO set to output low

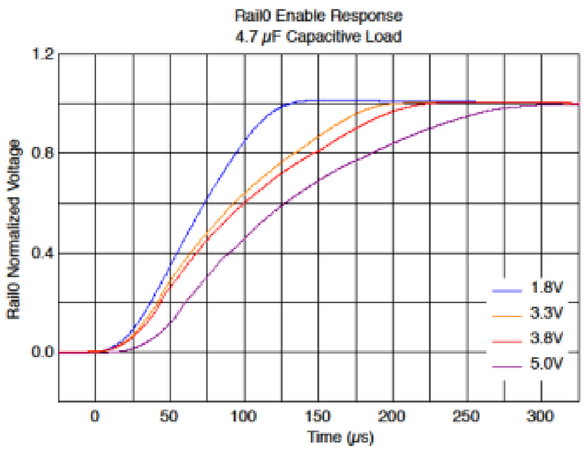

DIO Output High Voltage vs. Current Load
Voltage (V)
3.0
2.0
1.0
0.0
0
1
10
20
Current (mA)


Load Transient Response
100 mA to 1A step, Kelvin Sense Enabled
Rail0 Normalized Voltage
1.00
0.98
0.96
0.94
0
10
20
30
40
50
Time (μs)
1.8V
3.3V
3.8V
5.0V


Rail0 Enable Response
0 μF Capacitive Load
Rail0 Normalized Voltage
1.2
0.8
0.4
0.0
−50
0
50
100
150
200
Time (μs)
1.8V
3.3V
3.8V
5.0V


Rail1 Enable Response
Rail1 Normalized Voltage
1.0
0.5
0.0
0
100
200
300
400
500
600
700
Time (μs)
0μF
4.7μF


Load Transient Response
100mA to 1A step, Kelvin Sense Disabled
Rail0 Normalized Voltage
1.00
0.98
0.96
0.94
0
10
20
30
40
50
Time (μs)
1.8V
3.3V
3.8V
5.0V


Rail0 Enable Response
4.7 μF Capacitive Load
Rail0 Normalized Voltage
1.2
0.8
0.4
0.0
0
50
100
150
200
250
300
Time (μs)
1.8V
3.3V
3.8V
5.0V

Module Hardware and Software Default Values

The MTM-PM-1 module utilizes a subset of BrainStem entity implementations that are specific to the hardware's capabilities. Table 4: MTM-PM-1 Hardware and Software Default Values details the BrainStem API entities and macros used to interface to the MTM-PM-1 module. For C and C++ developers, these macros are defined in `aMTMPM1.h` from the BrainStem development package. For Python development, the module `MTMPM1` class defines the extent of each entity array.
While the BrainStem API entities define the full potential functionality of a given interface, not all features are supported by the MTM-PM-1 module. Table 4: MTM-PM-1 Hardware and Software Default Values defines each of the options implemented with each entity, which varies by entity index. 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.

Parameter	Index	Macro Name or Implemented Options	Notes
Module Definitions:			
Module Base Address	6	aMTMPM1_MODULE_BASE_ADDRESS	See aMTMPM1.h
UART Quantity	1		Pass-through to host (Available in BrainStem Version 2.2)
Entity Class Definitions:			
`digital` Entity Quantity	2	aMTMPM1_NUM_DIGITALS	
`rail` Entity Quantity	2	aMTMPM1_NUM_RAILS	
Adjustable (`rail0`)	1		
V_{supply} pass-through (`rail1`)	1		
`Store` Entity Quantity	2	aBRAINSTEM_NUM_STORES	
`system` Entity Quantity	1		
`timer` Entity Quantity	8	aMTMBRAINSTEM_NUM_TIMERS	

Table 4: MTM-PM-1 Hardware and Software Default Values[6]

[6] Refer to `aMTMPM1.h` within the BrainStem Development Kit download for actual file.

Device Drivers

MTM-PM-1 devices running firmware 2.1.4 and earlier require a BrainStem specific USB driver to be installed for a BrainStem device to properly get loaded into the operating system. Informational details on installing the USB driver can be found within the BrainStem Development Kit under the “drivers” folder.

Capabilities and Interfaces

The MTM-PM-1 module software is built on Acroname’s BrainStem technology. The module adheres to the BrainStem protocol on I²C and uses BrainStem software APIs. For the most part, functionality that is unique to the MTM-PM-1 is described in the following sections; refer to Table 5: Supported MTM-PM-1 BrainStem Entity API Methods for a complete list of all available API functionality. All shortened code snippets are loosely based on the C++ method calls – Python and Reflex are virtually the same. Please consult the BrainStem Reference for implementation details[7].

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.

Serial Number

Every MTM-PM-1 is assigned a unique serial number at the factory. This facilitates an arbitrary number of Manufacturing Test Module (MTM) devices attached to a single host computer or network. The following method call can retrieve the unique serial number for each device.

```
stem.system.getSerialNumber(serialNumber)
```

Module Default Base Address

All BrainStem modules come with a default network I2C base address for identification on the I2C bus. The default module base address for MTM-PM-1 is factory-set as 6, which is defined in the C source file `aMTMPM1.h` (in Python interface, the `brainstem.stem` module).

```
stem.system.getModule(module)
```

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-PM-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()
```

Rail Entities

Rails on the MTM-PM-1 module are powerful (no pun intended); they allow other devices and peripherals to consume power from the MTM-PM-1 module in a precisely controlled fashion. Two (2) different rails are available for use in a variety of an adjustable voltage rail (`rail0`). These rails are accessed through an array of BrainStem rail class entities. The MTM-PM-1 module implements a subset of the BrainStem rail class for each of these rails. Table 5: Supported MTM-PM-1 BrainStem Entity API Methods summarizes the implemented rail entity options for each entity index.
Both rails can be switched on or off through using the `setEnableExternal` and `setDisableExternal` API. The current limit for each rail can be configured in software from 0A to 3A, although the behavior of each rail on an over-current event differs slightly:

- `rail0` will simply reduce the output voltage to drive the specified current.
- `rail1` will be turned off by the hardware if the output current goes above the set limit, and automatically re-enable through hardware once the over-current condition is removed.

RAIL0 Voltage Setting

The output voltage is adjustable through the API across the operating range. Setting values outside the allowable range will return an error.

```
stem.rail[0].setVoltage(microvolts)
```

RAIL0 Operational Mode

`rail0` is unique as it contains an optional SMPS (switch mode power supply) pre-regulation stage power path. The pre-regulation power path stage can be configured through the API:

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

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

- 0 (operationalModeAuto)
- 1 (operationalModeLinear)
- 3 (operationalModeSwitcherLinear)

Default operational mode is configured as operationalModeLinear which disables the SMPS pre-regulator.
For applications such as RF receiver sensitivity testing, it is suggested to operate only in linear regulation mode to eliminate potential EMI sources. When operating in linear mode, make sure to consider power dissipation through the linear regulation stage; a higher input voltage will result in higher power dissipation. When linear mode is desired as well as high current operation, it is recommended to run the input voltage close to the MTM-PM-1 module's minimum input voltage to reduce power regulation losses. Switch mode power supply operation will allow a broader range of input voltages while maintaining high current demand limits. Default behavior is to automatically change between linear-mode and switcher-mode if an input voltage greater than 7.25V is applied with a 100mV hysteresis.

RAIL0 Operational State

`rail0` also features over-temperature and over-voltage protection. An over-temperature condition occurs when the board temperature crosses the safe operating threshold, and the rail is automatically disabled. The rail can then be re-enabled by first disabling and then re-enabling the rail via the software API. An over-voltage condition occurs when a voltage above the rail's set point is applied to the `rail0` output. When an over-voltage condition is detected, the hardware automatically disables the rail output. Once the over-voltage condition is removed, the rail output will resume desired operation without any software intervention.

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

The value *mode* is an 8-byte value, defined as the following:

mode	RAIL Operational Mode Description
0	Initializing (operationalState_Initializing)
1	Power Good (operationalState_PowerGood)
2	Power Fault (operationalState_PowerFault)
3	Linear Regulator Over-Temperature (operationalState_LDOOverTemp)

RAIL0 Temperature

RAIL0's subsystem power stage temperature can be monitored at the adjustable rail (`rail0`) linear regulation stage. 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-PM-1 module will disable the linear regulator until a safe operating temperatures is reached.

RAIL0 Kelvin Sensing

Remote sensing can be applied to compensate for line loss in a system often found in high transient load applications. The MTM-PM-1 provides a "3-wire" interface to provide feedback to the MTM-PM-1 power supply to adjust appropriately and dynamically.

```
stem.rail[0].getKelvinSensingMode(bEnable)
stem.rail[0].setKelvinSensingMode(bEnable)
```

The bEnable parameter is an integer that correlates to the following:

- 0 (kelvinSensingOff)
- 1 (kelvinSensingOn)

Determine whether kelvin sensing is enabled or disabled. Kelvin sensing can be disabled if the power stage incurs a fault on the rail0 power stage.

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

The state parameter is an integer that correlates to the following:

- 0 (kelvinSensingOff)
- 1 (kelvinSensingOn)

Digital Entities

The MTM-PM-1 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:

- 0 (bitDigitalConfigurationInput)
- 1 (bitDigitalConfigurationOutput)

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

```
stem.digital[0].setState(mode)
stem.digital[0].getState(mode)
```

I^{2}C Entities

The MTM-PM-1 includes access to a single I^{2}C bus, operating at a set 1Mbit/s rate.
NOTE: This bus, while user-accessible, is also used for primary BrainStem communication so there may be other, non-user-initiated traffic as well, particularly with linked BrainStem units.
The maximum data size for individual `read` and `write` operations on this bus through the BrainStem API is 20 bytes. For example, sending 2 bytes (0xBEEF) through the I^{2}C bus to a device with an address 0x42 would be written:

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

Reading 2 bytes of data from a device with an address 0x42 would be written:

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

Where *buffer* would be a char array in C++.
The I^{2}C bus also includes, as a convenience, software-controllable 330Ω pull-up resistors on the SDA and SCL lines, disabled by default. When using the MTM-PM-1 in a linked system, only a single set of pull-ups should be enabled in order for the I^{2}C bus to properly work (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 I^{2}C bus, either the internal pull-ups can be enabled, or external hardware pull-ups added.

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

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

- 0 (I^{2}C pull-ups off)
- 1 (i2cSetPullup)

Temperature Entities

An optional thermocouple interface can be ordered installed from the factory. Standard orders do not include this interface.

K-Type Thermocouple

If the MTM-PM-1 was purchased with the thermocouple interface installed, one can use a K-type thermocouple. The input from a connected and compatible K-type thermocouple can be read using the following function:

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

MTM-PM-1 Supported Entity Methods Summary

Detailed entity class descriptions can be found in the BrainStem Reference (http://acroname.com/entities/index.html). A summary of MTM-PM-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[0-1]`	`setConfiguration`	
	`getConfiguration`	
	`setState`	
	`getState`	
`i2c[0]`	`write`	
	`read`	
	`setPullup`	Disabled by default. I2C communication requires a single set of pull-ups enabled across the bus.
`rail[0-1]`	`setEnableExternal`	

	getEnableExternal	
	setCurrentLimit	
	getCurrent	
	getCurrentLimit	
	getVoltage	
rail[0]	setVoltage	
	setOperationalMode	
	getOperationalMode	
	getOperationalState	
	getTemperature	
	getKelvinSensingMode	
	setKelvinSensingMode	
	getKelvinSensingState	
store[0-1]	getSlotState	
	loadSlot	
	unloadSlot	
	slotEnable	
	slotDisable	
	slotCapacity	
	slotSize	
system[0]	save	
	reset	
	setLED	
	getLED	
	setSleep	
	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`	

Table 5: Supported MTM-PM-1 BrainStem Entity API Methods[7]

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

Mechanical

Dimensions are shown in inches [mm]. 3D CAD models are available through the MTM-PM-1 product page's Downloads section.

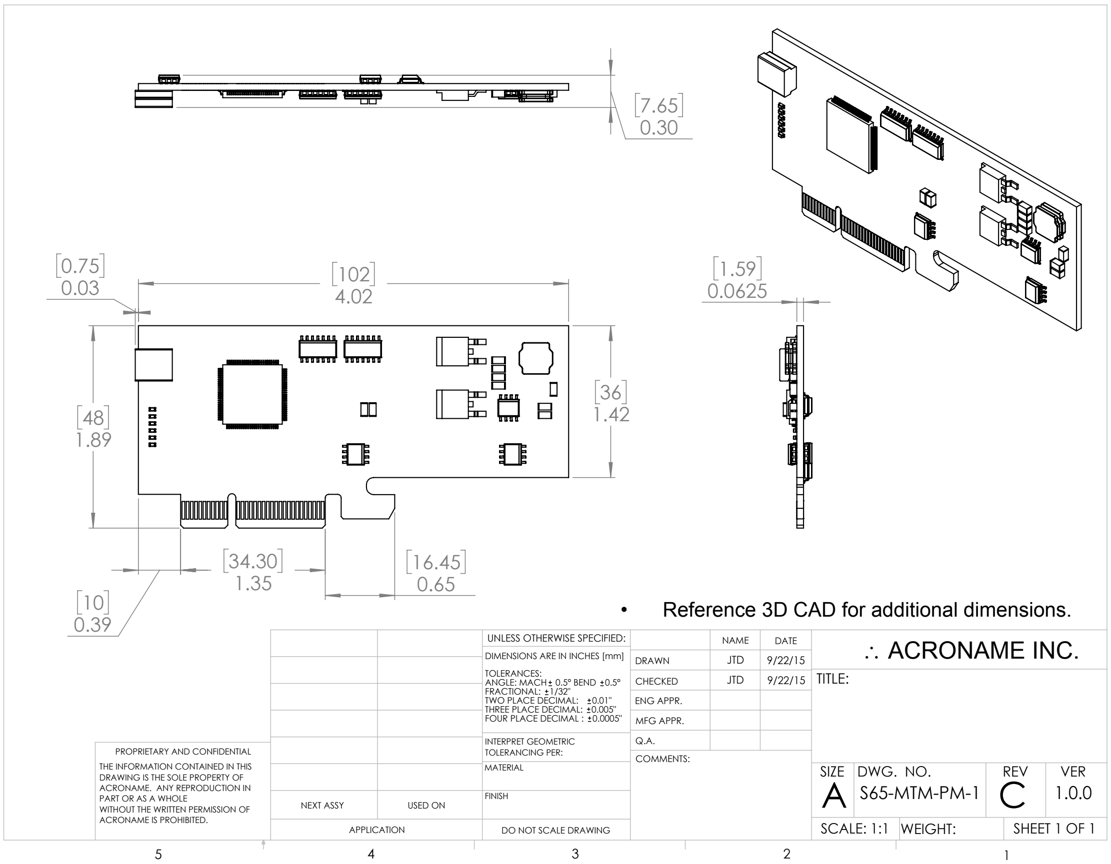


Figure 2: MTM-PM-1 Mechanical

Document Revision History

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

Revision	Date	Engineer	Description
1.0	September 29, 2015	JTD	Initial Revision
1.1	October 19, 2015	JTD	Reformatted, added Entity Section Specifics
1.2	October 23, 2015	MJK	Added pinout description