SENtral® M&M™

Motion & Measurement Modules

General Description

PNI's SENtral M&M motion and measurement modules provide accurate heading and orientation data in a small, low-power-consumption, and easy-to-integrate package. A module incorporates the SENtral motion coprocessor, a magnetometer, an accelerometer, a gyroscope, and an optional barometric pressure sensor with different SENtral M&M versions comprising different sensor models.

Unlike other inertial measurement units (IMUs) requiring extensive sensor fusion algorithm development and sensor calibration work, the Sentral M&M modules are pre-engineered to provide high accuracy motion tracking, heading, environmental data. And this is obtained at a fraction of the power used by any other solution on the market.

The SENtral M&M comes ready to integrate into a user's systemThe on-board EEPROM contains SENtral's configuration file and this automatically uploads into SENtral RAM when powered up.

With the SENtral M&M modules you can quickly and easily incorporate industry-leading motion-tracking and orientation measurement in your mobile device. We're sure you'll be impressed.

Features

- All-in-one motion & orientation tracking module, incorporates the SENtral motion coprocessor, 3-axis gyroscope, 3-axis accelerometer, 3-axis magnetometer, and barometric pressure sensor.
- Low power consumption.
- 11x11 mm footprint and SMT design for ease of integration into a user's system
- Multiple versions with different sensors.

Applications

- Personal Navigation & LBS
- Gaming & Augmented Reality
- Movement Science & Fitness

Ordering Information

Item	Sensors	Part #
White	none (Sentral only)	13734
Orange	BMI055+AK8963	13771
Red	MPU6500+AK8963	13763
Green	LSM330+AK8963	13736
Yellow	LSM9SD0	13738
Blue	LSM330+RM3100	13759
Purple	BMI160+BMM150+BMP280	13807
Pink	LSM6DS3+AK9911+LPS25H	13813

Table of Contents

List of Figures

List of Tables

1 Product Overview

The SENtral M&M Motion and Measurement Module is a castellated printed-circuit assembly that makes it easy to quickly integrate a complete motion-sensor-fusion system into a wearable or mobile device. A module incorporates the SENtral Motion Coprocessor, a magnetometer, an accelerometer, a gyroscope, and an optional barometric pressure sensor with different SENtral M&M versions integrating different sensor models. The SENtral Motion Coprocessor manages and uses data from the sensors to provide reliable motion tracking and an accurate compass heading, while consuming about 1% of the power of a comparable ARM-based sensor fusion microprocessor. SENtral outputs Euler angles (aka heading, pitch, and roll), quaternions, and sensor data. Quaternions uniquely define orientation and, unlike Euler angles, do not experience a singularity (i.e. gimbal lock) when pointing straight up. They easily can be converted to Euler angles, the rotation vector, and the rotation matrix (aka DCM), as discussed in Appendix I.

1.1 SENtral Features and Benefits

At the heart of the SENtral M&M module is PNI's revolutionary SENtral Motion Coprocessor. Listed below are some of the features and benefits of this device.

- **Low power consumption.** Offloads sensor processing from the less efficient host CPU, consuming <1% of the power of a general purpose microprocessor running a comparable sensor fusion algorithm. Provides the ability to tailor the tradeoff between power consumption and motion-tracking performance.
- **Industry-leading heading accuracy.** Unparalleled heading accuracy for consumer electronics applications.
- **Continuous hard and soft-iron auto-calibration.** Unlike other motion-tracking products, SENtral calibrates for both hard-iron and soft-iron magnetic distortion. Specifically, soft-iron distortion is quite difficult to correct, and can contribute up to 90° of error. It can be caused by materials widely used in mobile and consumer electronic devices, such as EMI shielding tape and other shielding. Additionally, since a host system's magnetic signature can change over time and temperature, SENtral's continuous auto-calibration ensures accuracy over time.
- **Magnetic anomaly compensation.** With SENtral, heading and motion tracking is unaffected by short-term magnetic anomalies, such as rebar in buildings, desks, speakers etc., that can easily throw off the accuracy. SENtral establishes if a transient magnetic anomaly is present and compensates for this.
- **Sensor flexibility.** SENtral works with most common consumer electronics motion sensors, so designers can choose the sensors most appropriate for their systems.

- **Small form-factor.** 1.6x1.6x0.5 mm chip-scale package on 0.4 mm pitch. Uses little PCB real estate, allowing for painless integration.
- **I^{2}C interface.** Uses industry-standard I^{2}C protocol in a low-power implementation to interface to the sensors and the host, so system integration is straightforward. Standard, Fast, Fast Plus, and High Speed are supported on the host bus.
- **Outputs.** SENtral natively outputs Euler angles (heading, pitch, and roll), quaternions, rotational velocity, linear acceleration, and magnetic field.
- **Pass-Through** allows for direct communication with devices on the I^{2}C sensor bus.

1.2 SENtral M&M System Overview

Figure 1-1 provides a reference schematic for SENtral M&M modules. While this diagram applies for most versions of the SENtral M&M, the White and Blue M&M modules differ from what is shown and the Purple and Pink include an additional pressure sensor (not shown). Section 7 addresses additional functionality provided by the pressure sensor on the Pink & Purple M&M modules. Specific schematics for each module are available from PNI on request. How to interface with the SENtral M&M is covered in more detail in Section 3.

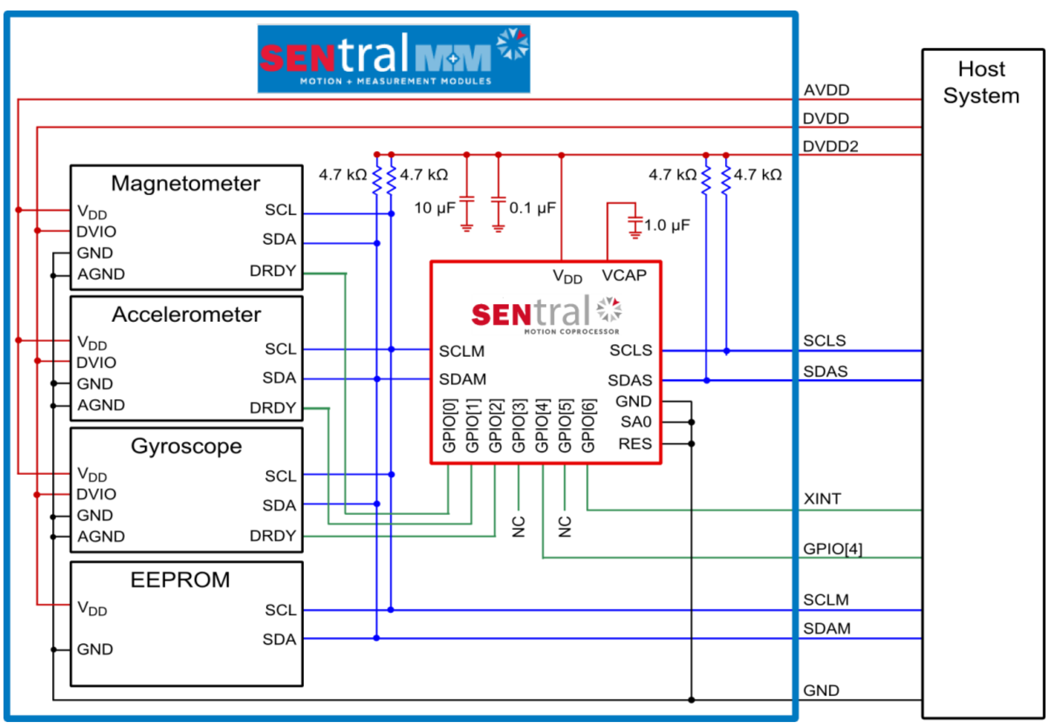

Figure 1-1: SENtral M&M Module Reference Schematic

A few points on diagram:

- The layout shows a discrete magnetometer, accelerometer, and gyroscope. SENtral M&M modules generally incorporate a combo sensor that combines the gyroscope and accelerometer into a single device or all three sensors into a single device.
- SENtral acts as a slave on a host system I^2C bus. This does not need to be a dedicated bus, although it is shown this way in the schematic.
- The SCLM and SDAM lines can be used to monitor SENtral's I^2C sensor bus, but this is not necessary. These lines are optional and may be left unconnected.
- If the host will poll SENtral, rather than running in an interrupt-driven manner, it is not necessary to connect GPIO[6], the host interrupt line, to the host system.
- GPIO[4] is intended for future use and currently serves no purpose. This can be left unconnected.

2 SENtral Specifications[1]

2.1 Performance Characteristics

Table 2-1: Performance Characteristics

Parameter	Minimum	Typical	Maximum	Units
Heading Accuracy		2		° rms
Output Data Rate		200	400	Hz

2.2 Electrical Characteristics

Table 2-2: Absolute Maximum Ratings

Parameter	Symbol	Minimum	Maximum	Units
Supply Voltage	V_{DD}	-0.3	+3.6	VDC
Input Pin Voltage	V_{IN}	GND – 0.3	V_{DD} + 0.3	VDC
Storage Temperature		-50°	+150°	C

CAUTION:

Stresses beyond those listed above may cause permanent damage to the device. These are stress ratings only. Operation of the device at these or other conditions beyond those indicated in the operational sections of the specifications is not implied.

Footnote

1. Specifications subject to change.

Table 2-3: Operating Conditions

Parameter	Symbol	Min	Typ	Max	Units
Digital Supply Voltage – Sensors & EEPROM	DVDD	1.71		AVDD	VDC
Supply Voltage – SENtral	DVDD2	1.6		3.3	VDC
Analog Supply Voltage – Sensors	AVDD	2.4		3.6	VDC
Power-On Reset Threshold	V_{POR}		1.0		VDC
High Level Input Voltage	V_{IH}	$0.7*V_{DD}$		V_{DD}	VDC
Low Level Input Voltage	V_{IL}	0		$0.3*V_{DD}$	VDC
High Level Output Current, $V_{OH} = V_{DD} – 0.3V$	I_{OH}			-1	mA
Low Level Output Current, $V_{OL} = 0.3V$	I_{OL}	1			mA
I^2C Interface Data Rate[1] – Host Bus				3400	kbits/sec
I^2C Interface Data Rate[1] – Sensor Bus				1000	kbits/sec
I^2C Interface Data Rate[1] – Pass-Through				400	kbits/sec
Operating Temperature	T_{OP}	-40	+25	+85	C

Footnote:

1. SENtral's I^2C Host Interface supports Standard, Fast, Fast Plus, and High Speed Modes. High Speed Mode (3400 kHz) is supported with a reduced range of V_{DD} and bus capacitance. SENtral's I^2C sensor bus interface supports Standard, Fast, and Fast Plus Modes. Pass-Through state, which connects the sensor bus and host bus, supports Standard and Fast Modes.

3 Interface

The SENtral M&M pin-out is given in Table 3-1. Pin-outs also are given alongside the device mechanical drawings in Section 5. See Table 2-3 for the operating ranges of DVDD, DVDD2, and AVDD. A discussion of the communication interface follows the table.

Table 3-1: SENtral M&M Module Pin Assignments

Pin Name	Description	M&M Orange, Red, Green, & Yellow Pin#	M&M White Pin #	M&M Blue Pin #
DVDD	Digital Supply Voltage – Sensors & EEPROM	1	NA	2
DVDD2	Supply Voltage – SENtral	2	D1	2
AVDD	Analog Supply Voltage – Sensors	7	NA	7
GND	Ground	8	D2	8
SCLS	I^2C host bus SCL clock line	3	B1	3
SDAS	I^2C host bus SDA data line	5	A1	5
SDAM	I^2C sensor bus SDA data line	9	A4	9
SCLM	I^2C sensor bus SCL clock line	10	B4	10
GPIO[0]	SENtral Accelerometer Interrupt	--	D4	--
GPIO[1]	SENtral Magnetometer Interrupt	--	C4	--
GPIO[2]	SENtral Gyroscope Interrupt	--	A3	--
GPIO[3]	Reserved	--	B3	--
GPIO[4]	Reserved	6	A2	6
GPIO[5]	Reserved	--	B2	--
GPIO[6]	Host Event Interrupt	4	C1	4
SA0	Slave Address Pin 0	--	C3	--
VCAP	Regulator Capacitor	--	D3	--
Reserved	Reserved (not connected)	--	C2	1, 12, 13

Communication with the host processor is via SENtral's I^2C host interface, where the SENtral M&M acts as a slave device and the host's processor acts as the master. The host interrupt line informs the host system when SENtral has updated measurement data. The SENtral Motion Coprocessor on the SENtral M&M module communicates with the module's sensors over the sensor bus, where SENtral is the I^2C master and the sensors are slave devices.

SENtral's I^{2}C interfaces comply with NXP's UM10204 specification and user manual, rev 04. Standard, Fast, Fast Plus, and High Speed modes of the I^{2}C protocol are supported by SENtral's I^{2}C host interface. Below is a link to this document.

http://www.nxp.com/documents/user_manual/UM10204.pdf

3.1 I^{2}C Timing

SENtral's I^{2}C timing requirements are set forth below, in Figure 3-1 and Table 3-2. For the timing requirements shown in Figure 3-1, transitions are 30% and 70% of V_{DD}.

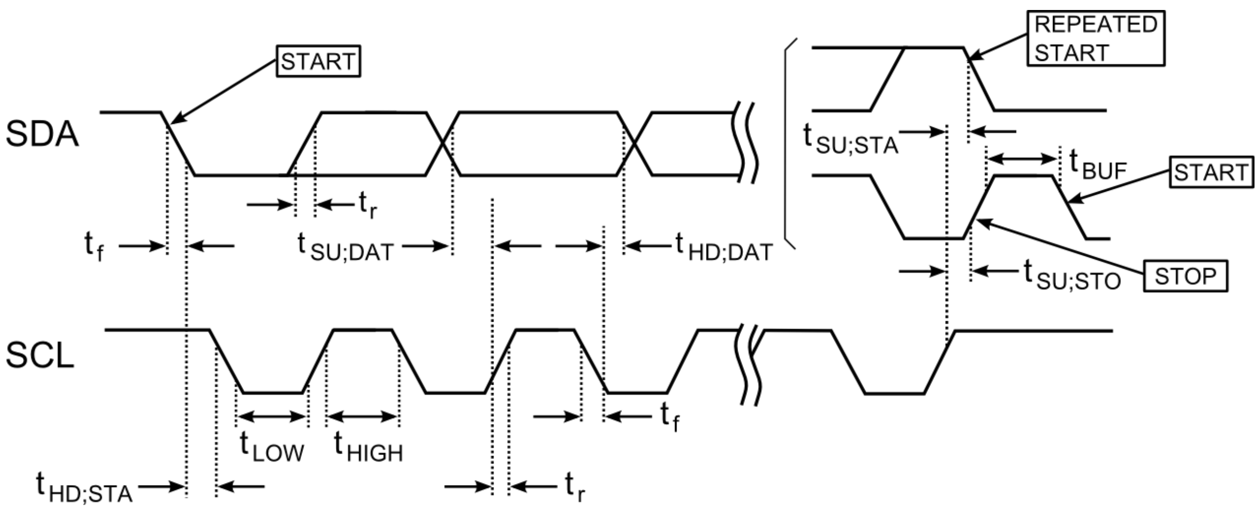

Figure 3-1: I^{2}C Timing Diagram

Table 3-2: I^{2}C Timing Parameters

		Standard		Fast		Fast Plus		Units
Symbol	**Parameter**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	
f_{SCL}	SCL Clock	0	100	0	400	0	1000	kHz
t_r	SDA & SCL Rise Time	-	1000	20	300		120	ns
t_f	SDA & SCL Fall Time	-	300	20*(V_{DD}/ 5.5V)	300	20*(V_{DD}/ 5.5V)	120	ns
t_{LOW}	LOW period of SCL Clock	4.7	-	1.3	-	0.5	-	µs
t_{HIGH}	HIGH period of SCL Clock	4.0	-	0.6	-	0.26	-	µs
$t_{HD;STA}$	Hold time (repeated) START	4.0	-	0.6	-	0.26	-	µs
$t_{HD;DAT}$	Data hold time	0	-	0	-	0	-	µs
$t_{SU:DAT}$	Data set-up time	250	-	100	-	50	-	ns
$t_{SU;STA}$	Set-Up time for repeated Start	4.7	-	0.6	-	0.26	-	µs
$t_{SU;STO}$	Stop set-up time	4.0	-	0.6	-	0.26	-	µs
t_{BUF}	Bus free time between STOP & START	4.7	-	1.3	-	0.5	-	µs

3.2 I^{2}C Host Interface (Host Bus)

The host will control the SENtral M&M on the host bus via SENtral's I^{2}C host interface. The host interface consists of 2 wires: the serial clock, SCLS, and the serial data line, SDAS. Both lines are bi-directional. SENtral is connected to the host bus via the SDAS and SCLS pins, which incorporate open drain drivers within the device. Note the SENtral M&M module incorporates 4.7 kΩ pull-up resistors on the host bus clock and data lines, so if the host system also incorporates pull-up resistors on these line the resistors will act in parallel.

The SENtral M&M's **7-bit** I^{2}C slave address is 0x28 (0b0101000). The shifted address is 0x50.

Data transfer is always initiated by the host. Data is transferred between the host and SENtral serially through the data line, SDAS, in an 8-bit transfer format. The transfer is synchronized by the serial clock line, SCLS. Supported transfer formats are single-byte read, multiple-byte read, single-byte write, and multiple-byte write. The data line can be driven either by the host or SENtral. Normally the serial clock line will be driven by the host, although exceptions can exist when clock-stretching is implemented in Pass-Through State.

3.2.1 I²C Transfer formats

Figure 3-2 illustrates writing data to registers in single-byte or multiple-byte mode.

START	SLAVE ADDRESS							RW	ACK	REGISTER ADDRESS (N)								ACK	DATA TO REGISTER (N)								ACK	DATA TO REGISTER (N+1)								ACK	STOP
S	A6	A5	A4	A3	A2	A1	A0	0	0	R7	R6	R5	R4	R3	R2	R1	R0	0	D7	D6	D5	D4	D3	D2	D1	D0	0	D7	D6	D5	D4	D3	D2	D1	D0	0	P

From Host to SENtral

From SENtral to Host

----------- Data Transferred (n bytes + acknowledge) -----------

Figure 3-2: I²C Slave Write Example

The I²C host interface supports both a read sequence using repeated START conditions, shown in Figure 3-3, and a sequence in which the register address is sent in a separate sequence than the data, shown in Figure 3-4 and Figure 3-5.

START	SLAVE ADDRESS							RW	ACK	REGISTER ADDRESS (N)								ACK	START	SLAVE ADDRESS							RW	ACK	DATA FROM REGISTER (N)								NACK	STOP
S	A6	A5	A4	A3	A2	A1	A0	0	0	R7	R6	R5	R4	R3	R2	R1	R0	0	SR	A6	A5	A4	A3	A2	A1	A0	1	0	D7	D6	D5	D4	D3	D2	D1	D0	1	P

Data Transferred
(n bytes + acknowledge)

Figure 3-3: I²C Slave Read Example, with Repeated START

START	SLAVE ADDRESS							RW	ACK	REGISTER ADDRESS (N)								ACK	STOP
S	A6	A5	A4	A3	A2	A1	A0	0	0	R7	R6	R5	R4	R3	R2	R1	R0	0	P

Figure 3-4: I²C Slave Write Register Address Only

START	SLAVE ADDRESS							RW	ACK	DATA FROM REG. (N)								ACK	DATA FROM REG. (N+1)								NACK	STOP
S	A6	A5	A4	A3	A2	A1	A0	1	0	D7	D6	D5	D4	D3	D2	D1	D0	0	D7	D6	D5	D4	D3	D2	D1	D0	1	P

From Host to SENtral

From SENtral to Host

------------- Data Transferred (n bytes + acknowledge) -------------

Figure 3-5: I²C Slave read register from current address

3.3 I²C Sensor Interface (Sensor Bus)

Understanding how the sensor interface operates is not necessary when using the SENtral M&M module. However, understanding the sensor interface is useful if there is a need to communicate directly with a sensor or the EEPROM in Pass-Through state.

The SENtral Motion Coprocessor on the SENtral M&M module communicates with the module's accelerometer, gyroscope, magnetometer and pressure sensor over the module's sensor bus, where SENtral is the I²C master and the sensors are slave devices. On the sensor bus, SENtral initiates data transfer and generates the serial clock. The two wires comprising the sensor bus are SDAM, the serial data line, and SCLM, the serial clock. Both are bidirectional and driven by open drain transistors within SENtral. These can be monitored by the host, but should not be written to by the host. Each line is attached to a 4.7 kΩ pull-up resistor. SENtral's I²C sensor interface supports Standard mode with a rate up to 100 kbit/s, Fast mode with a rate up to 400 kbit/s, and Fast Plus mode with a rate up to 1000 kbit/s.

3.4 Host Interrupt/GPIO Lines

GPIO[6] provides an interrupt to the host whenever a defined event occurs. Exactly which types of events will trigger an interrupt are set by the EnableEvents register, which is discussed in Section 4.2 This interrupt line can be used to signal the host that new results are available for reading. Alternately, the host may poll SENtral's EventStatus register, discussed in Section 0, to determine if any events of interest have been updated. If polling will be used, PNI recommends polling on a regular interval such that an error event will be identified in a timely manner.

GPIO[4] is not currently used, and generally should be left unconnected. This is also true for GPIO[3] and GPIO[5], which are only accessible on the SENtral White M&M.

4 Operation

Figure 4-1 provides a flow chart of the SENtral M&M module's initialization process, and a discussion of this process follows in Section 4.1 **For the registers, all multi-byte elements are stored and transmitted using the Little Endian convention: the least significant byte is stored at the lowest address and transmitted first over the I^2C bus.**

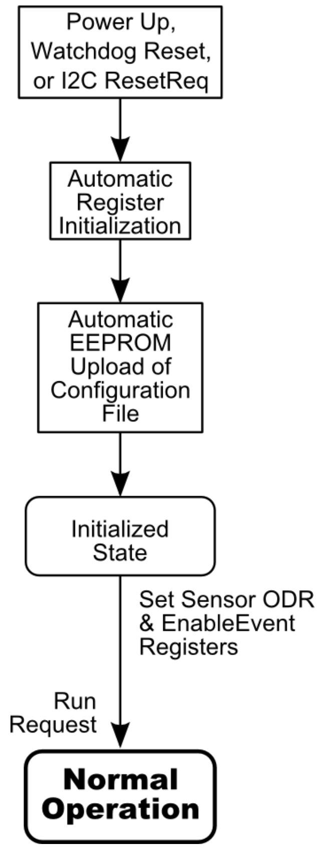

Figure 4-1: SENtral Initialization Sequence

Once the initialization sequence is complete, there are three states in which SENtral may reside: Normal Operation, Standby, and Pass-Through. Figure 4-2 indicates the recommended way to get from one state to another, and these states are discussed in detail in Sections 4.2 and 0 (Normal Operation), 4.4 (Standby), and 4.5 (Pass-Through).

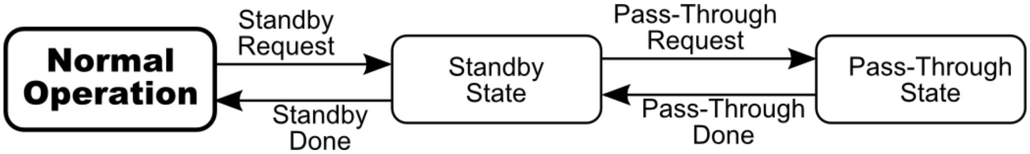

Figure 4-2: SENtral Operational States

4.1 Power-Up

After powering up or issuing a ResetReq command, SENtral automatically initializes the registers and loads the SENtral Configuration File from the onboard EEPROM, as indicated in Figure 4-1. The Configuration File contains information specific to the particular SENtral M&M flavor, and is discussed more thoroughly in the SENtral Motion Coprocessor Technical Datasheet. Once the upload is complete, SENtral enters Initialized State and waits for instructions from the host.

Table 4-1: Configuration File Upload from EEPROM Registers

Register Name	Address	Register Value
SentralStatus	0x37	[0] EEPROM. 1 = EEPROM detected [1] EEUploadDone. 1 = EEPROM upload completed [2] EEUploadError. 1 = Calculated CRC of EEPROM is incorrect. Only valid when EEUploadDone = 1. [3] Idle. 1 = Device in Unprogrammed or Initialized state. [4] NoEEPROM. 1 = No EEPROM detected.
ResetReq	0x9B	[0] ResetRequest. 1 = Emulate a hard power down/power up.

The host should confirm a successful EEPROM upload by following the steps below:

- Read the value from the SentralStatus register.
- Check bit [0], the EEPROM bit, to ensure an EEPROM is detected by SENtral.
- Check bit [1], the EEUploadDone bit. If this is '0' then the Configuration File upload is not complete, and reread the SentralStatus register until bit [1] = 1.
- Once bit [1] = 1, check bit [2], the EEUpload Error bit. If this is '0', then the upload was successful.

If the Configuration File upload failed, send a Reset command by writing 0x01 to the ResetReq register or power off/power on the device. If the issue persists, refer to the SENtral Motion Coprocessor datasheet for debugging hints.

4.2 Initial Register Set-Up

After the initialization process is complete, it is necessary to configure a few of SENtral's registers before running in Normal Operation. These registers are given in Table 4-2.

Table 4-2: Registers for Initial Set-up

Register Name	Address	Register Value
MagRate	0x55	Requested magnetometer output data rate
AccelRate	0x56	Requested accelerometer output data rate divided by 10
GyroRate	0x57	Requested gyroscope output data rate divided by 10
QRateDivisor	0x32	Along with GyroRate, establishes output data rate for quaternion data.
AlgorithmControl	0x54	[0] 1 = StandbyEnable 0 = Disable Standby State (return to Normal Operation) [1] RawDataEnable. 1 = Raw data provided in MX, MY, MZ, AX, AY, AZ, GX, GY, & GZ. 0 = Scaled sensor data. [2] HPRoutput. 1 = Heading, pitch, and roll output in QX, QY, & QZ. QW = 0.0. 0 = Quaternion outputs.
EnableEvents	0x33	'1' indicates an interrupt to the host will be generated for the event. [0] CPUReset. Non-maskable [1] Error [2] QuaternionResult [3] MagResult [4] AccelResult [5] GyroResult

Perform the following operations to run SENtral as desired.

- Set the sensor output data rates (ODRs): MagRate, AccelRate, and GyroRate. If a sensor rate is set to 0x00, SENtral will shutdown the sensor and disable SENtral background calibration. There are two major points regarding setting these registers:
  - The AccelRate and GyroRate register values should be 1/10th the desired ODR, while the MagRate value should match the desired ODR. For example, if the desired ODR is 30 Hz for the magnetometer, 100 Hz for the accelerometer, and 200 Hz for the gyroscope, then the respective register values should be 0x1E (30_d), 0x0A (10_d), and 0x14 (20_d).
  - The actual accelerometer and gyro ODRs are limited to the ODRs supported by the specific sensors. **If the AccelRate or GyroRate register values do not correspond to a supported ODR, then the next highest ODR will be used.** For instance, if the GyroRate register is set to 0x14, which corresponds to 200 Hz, but the gyro supports 95 Hz, 190 Hz, and 380 Hz, then the actual gyro ODR will be 380 Hz since this is the closest supported rate above that requested by the register.

- Establish the quaternion output data rate, where the quaternion output data rate equals GyroRate divided by QRateDivisor. The default for QRateDivisor is 0x00, which is interpreted as ‘1’ and results in the quaternion output data rate equaling GyroRate.
- Establish how SENtral’s orientation and sensor data is to be output. The AlgorithmControl register allows the user to select either quaternion or Euler angles (heading, pitch, and roll) for orientation outputs, and either scaled or raw sensor data outputs. The defaults are quaternions and scaled sensor data.
- Establish which events will trigger an interrupt to the host by configuring the EnableEvent register. PNI specifically recommends enabling bit [1], the Error interrupt bit, in addition to whichever other interrupts the user wants.

Example steps to do this are below:

- Write 0x1E0A0F to the MagRate register. Since SENtral automatically increments to the next register, this also populates the AccelRate and GyroRate registers. This sets MagRate to 100 Hz, AccelRate to 100 Hz, and GyroRate to 150 Hz.
- Write 0x02 to the QRateDivisor Register. This sets the quaternion output data rate to be half the GyroRate. This step is optional, as the default register value of 0x00 sets the quaternion output data rate equal to GyroRate.
- Write 0x06 to the AlgorithmControl register. This enables heading, pitch, and roll orientation outputs and raw sensor data outputs. This step is optional, as the default register value of 0x00 results in outputs of quaternions and scaled sensor data.
- Write 0x07 to the EnableEvents register. This sets the host to receive interrupts from SENtral whenever the quaternion results registers (QX, QY, QZ, and QW) are updated, an error has been detected, or SENtral has been Reset but the Configuration File has not been uploaded. If the host regularly will poll SENtral, rather than run in an interrupt-driven manner, it is not necessary to set the EnableEvents register.

Note: *It is necessary to set the MagRate, AccelRate, AND GyroRate registers to non-zero values for the SENtral algorithm to function properly and to obtain reliable orientation and scaled sensor data. If a [Sensor]Rate register is left as 0x00 after power-up, or is changed to 0x00, this effectively disables that sensor within the SENtral algorithm. Also, the CalStatus, MagTransient, and AlgorithmSlow bits become undefined.*

4.3 Running in Normal Operation

After performing the steps listed above, SENtral is ready to start generating orientation data. The registers used to run in Normal Operation are given in Table 4-2, the steps to follow comes after this, and a flow diagram is given in Figure 4-3.

Table 4-3: Normal Operation Registers

Register Name	Address	Register Value
HostControl	0x34	[0] 1 = RunEnable 0 = Enable Initialized State (Standby State generally is preferred since enabling Initialized State resets the SENtral algorithm, including calibration data.)
EventStatus	0x35	'1' indicates a new event has been generated. [0] CPUReset [1] Error [2] QuaternionResult [3] MagResult [4] AccelResult [5] GyroResult

Below are the steps to follow when operating in Normal Operation state.

a) Write 0x01 to the HostControl register. This sets the RunEnable bit to '1' and enables the sensors and the SENtral algorithm.

b) If operating in an interrupt-driven mode, then the host waits until it receives an interrupt signal from SENtral. Alternatively the host may operate on a polling basis, rather than an interrupt-driven basis, in which case the interrupt line may not be used.

c) Once an interrupt is received by the host or the host otherwise decides to read new data, read the EventStatus register.

d) Interpret and act on the EventStatus register in the priority shown in Figure 4-3. If bit [1], the Error bit, is '1', see Section 4.3.1. If bits [2], [3], [4], or [5], the Results bits, are '1', see Section 4.3.2. Bit [0], the CPUReset bit, should never be '1', since this bit only can be '1' after a Reset or powering up and prior to loading the Configuration File, and on the SENtral M&M module loading of the Configuration File is automatically performed after powering up.

e) Repeat steps c and d until new orientation data is not needed and/or the host decides to enter a different state.

Reading the EventStatus register clears it. It is possible for more than one bit position to be '1' in the EventStatus register, especially if the host does not always read the EventStatus register after receiving an interrupt. Similarly, if multiple bits are set to '1' in the EventStatus register, once the register is read all the bits will be set to '0'. For this reason the EventStatus register should be processed in the priority shown in Figure 4-3, as information will be cleared for events that are not handled.

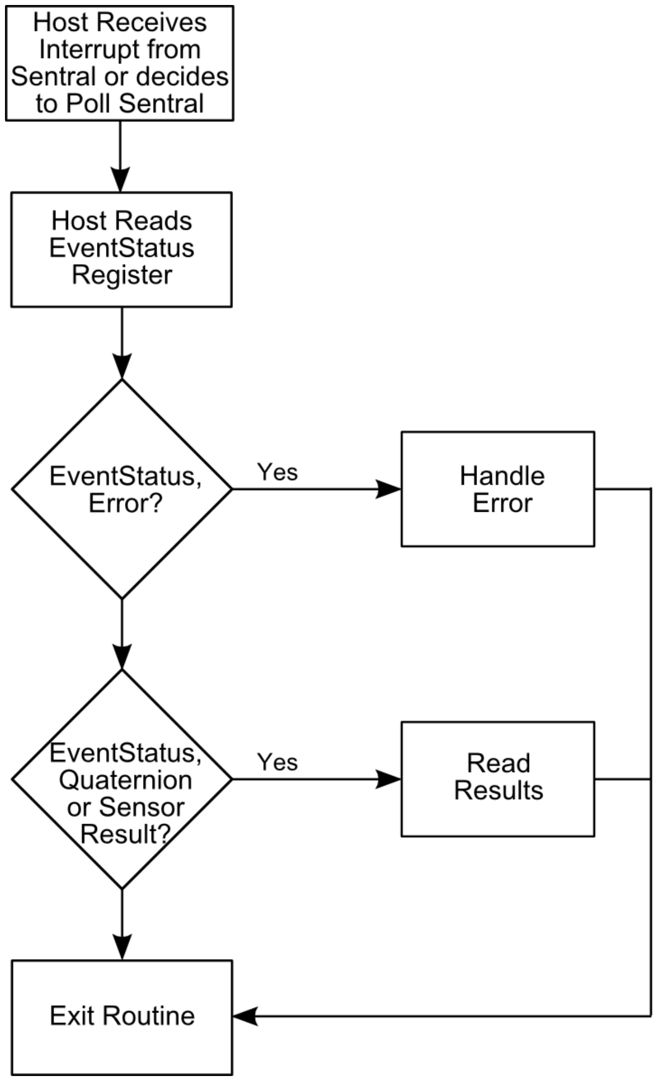

Figure 4-3: SENtral Normal Operation Flow

A discussion of how to handle the events follows.

4.3.1 Error

In the event of an error, SENtral will trigger an error interrupt and SENtral will enter Standby State. See the Section 4.6 for recommendations on Troubleshooting and/or reset SENtral by sending 0x01 to the ResetReq register, at address 0x9B.

4.3.2 Read Results

The Results Registers' addresses, formats, and full-scale ranges are given below in Table 4-4. For an explanation of how to convert quaternions to the rotation vector, the rotation matrix, or Euler angles (heading, pitch, and roll), see Appendix I. The resolution is 32 kHz for all timestamps.

Note: *All multi-byte elements are stored and transmitted using the Little Endian convention: the least significant byte is stored at the lowest address and transmitted first over the I^2C bus.*

Table 4-4: Results Registers

Name	Address (Hex)	Description	Format	Full-Scale Range
QX	00 – 03	Normalized Quaternion – X, or Heading	Float32	0.0 – 1.0, or ±π
QY	04 – 07	Normalized Quaternion – Y, or Pitch	Float32	0.0 – 1.0, or ±π/2
QZ	08 – 0B	Normalized Quaternion – Z, or Roll	Float32	0.0 – 1.0, or ±π
QW	0C – 0F	Normalized Quaternion – W, or 0.0	Float32	0.0 – 1.0
QTime	10 – 11	Quaternion Data Timestamp	UInt16	0 – 2048 msec
MX	12 – 13	Magnetic Field – X Axis, or Raw Mag Data	Int16	±1000 µT when scaled
MY	14 – 15	Magnetic Field – Y Axis, or Raw Mag Data	Int16	±1000 µT when scaled
MZ	16 – 17	Magnetic Field – Z Axis, or Raw Mag Data	Int16	±1000 µT when scaled
MTime	18 – 19	Magnetometer Interrupt Timestamp	UInt16	0 – 2048 msec
AX	1A – 1B	Linear Acceleration – X Axis, or Raw Accel Data	Int16	±16 g when scaled
AY	1C – 1D	Linear Acceleration – Y Axis, or Raw Accel Data	Int16	±16 g when scaled
AZ	1E – 1F	Linear Acceleration – Z Axis, or Raw Accel Data	Int16	±16 g when scaled
ATime	20 – 21	Accelerometer Interrupt Timestamp	UInt16	0 – 2048 msec
GX	22 – 23	Rotational Velocity – X Axis, or Raw Gyro Data	Int16	±5000°/s when scaled
GY	24 – 25	Rotational Velocity – Y Axis, or Raw Gyro Data	Int16	±5000°/s when scaled
GZ	26 – 27	Rotational Velocity – Z Axis, or Raw Gyro Data	Int16	±5000°/s when scaled
GTime	28 – 29	Gyroscope Interrupt Timestamp	UInt16	0.0 – 2.048 sec

4.4 Standby State

In Standby State overall system power consumption is dramatically reduced because both the SENtral algorithm and the sensors are shut down. Table 4-5 provides the registers associated with Standby State.

Table 4-5: Standby Registers

Register Name	Address	Register Value
AlgorithmControl	0x54	[0] 1 = StandbyEnable 0 = Disable Standby State
AlgorithmStatus	0x38	[0] 1 = SENtral in Standby State 0 = SENtral not in Standby State

The steps to enter and exit Standby State are given below:

- Write 0x01 to the AlgorithmControl register. This places SENtral in Standby State.
- Read the AlgorithmStatus register. If bit [0] is '1', then SENtral is in Standby State. This step is optional.
- When you are ready to exit Standby State, write 0x00 to the AlgorithmControl register. This takes SENtral out of Standby State and normally will place it back into Normal Operation.
- Read the AlgorithmStatus register. If bit [0] is '0', then SENtral is not in Standby State. This step is optional.

4.5 Pass-Through State

In Pass-Through State, SENtral's sensor and host interfaces are connected by internal switches so the host system can communicate directly with the sensors or EEPROM. To enter Pass-Through State, SENtral first either should be in Standby or Initialized State. Consequently, in Pass-Through State the SENtral algorithm, host interrupt line, and sensors are disabled, unless a sensor is directly turned on by the host. When exiting Pass-Through State, SENtral will return to its prior state.

Note: *When entering Pass-Through State the sensor's registers retain the values established by SENtral, and when exiting Pass-Through State any register changes will be retained.*

Uses for the Pass-Through State include:

- Direct control of sensors, if desired.
- Debugging.
- Communication with the dedicated EEPROM, if implemented. Specifically, if a new Configuration File is generated, the host can write this into the EEPROM when in Pass-Through State, as discussed in the SENtral Motion Coprocessor datasheet.

Since operating in Pass-Through State requires stopping the SENtral algorithm, Pass-Through State is not recommended for accessing sensor data unless reliable heading data is not required. If sensor data and reliable heading data are both desired, they can both be accessed during Normal Operation from the Results Registers, as given in Table 4-4.

Table 4-6 provides the registers associated with Pass-Through State.

Table 4-6: Pass-Through Registers

Register Name	Address	Register Value
AlgorithmControl	0x54	[0] 1 = StandbyEnable 0 = Disable Standby State
AlgorithmStatus	0x38	[0] 1 = SENtral in Standby State 0 = SENtral not in Standby State
PassThroughControl	0xA0	[0] 1 = Enable Pass-Through State 0 = Disable Pass-Through State
PassThroughStatus	0x9E	[0] 1 = SENtral in Pass-Through State. 0 = SENtral not in Pass-Through State.

The steps to go in and out of Pass-Through State are given below.

- Write 0x01 to the AlgorithmControl register. This places SENtral in Standby State.
- Write 0x01 to the PassThroughControl register. This places SENtral in Pass-Through State.
- Read the PassThroughStatus register. If bit [0] is '1', then SENtral is in Pass-Through State. This step is optional.
- When you are done in Pass-Through State, write 0x00 to the PassThroughControl register. This terminates Pass-Through mode and returns SENtral to Standby State.
- Write 0x00 to the AlgorithmControl register. This takes SENtral out of Standby State and normally will place it back into Normal Operation.

4.6 Troubleshooting

This section provides guidance in troubleshooting SENtral, and is divided into hardware-related and software-related errors.

4.6.1 Hardware-Related Error Conditions

Possible indications of a hardware-related problem are given below in Table 4-7.

Table 4-7: Hardware-Related Error Indications

Register Name	Address	Error Indication
EventStatus	0x35	[0] 1 = CPURest. SENtral Configuration File needs uploading. See Section 4.1.
SentralStatus	0x37	[2] 1 = EEUploadError. Issue with uploading from the dedicated EEPROM. See Section 4.1.
MagRate	0x55	0x00 – Value lost
AccelRate	0x56	0x00 – Value lost
GyroRate	0x57	0x00 – Value lost

In the event of such errors, SENtral will enter Standby State, shut down the sensors, and generate an interrupt to the host. Possible reasons for hardware-related errors include problems with the EEPROM upload, power transients detected by power management, and errors in software detected by Watchdog. Often the error can be cleared by sending the ResetReq command.

4.6.2 Software-Related Error Conditions

Possible indications of software-related errors are given below in Table 4-8:

Table 4-8: Software-Related Error Indications

Register Name	Address	Error Indication
EventStatus	0x35	[1] 1 = Error.
SensorStatus	0x36	[0] MagNACK. 1 = NACK from magnetometer [1] AccelNACK. 1 = NACK from accelerometer [2] GyroNACK. 1 = NACK from gyroscope [4] MagDeviceIDErr. 1 = Unexpected DeviceID from magnetometer [5] AccelDeviceIDErr. 1 = Unexpected DeviceID from accelerometer [6] GyroDeviceIDErr. 1 = Unexpected DeviceID from gyroscope.
SentralStatus	0x37	[3] 1 = Idle. SENtral in Initialized State.
ErrorRegister	0x50	Non-zero value indicated an error. See Table 4-9.
RAMVersion	0x72, 0x73	Unexpected Configuration File revision level.

If the ErrorRegister indicates a non-zero value, then the value provides additional information on the sensor that is causing a problem, as given in Table 4-9.

Table 4-9: ErrorRegister Indications

Value	Error Condition	Response
0x00	No error	
0x80	Invalid sample rate selected	Check sensor rate settings.
0x30	Mathematical Error	Check for software updates
0x21	Magnetometer initialization failed	This error can be caused by a wrong driver, physically bad sensor connection, or incorrect I^2C device address in the driver
0x22	Accelerometer initialization failed	
0x24	Gyroscope initialization failed	
0x11	Magnetometer rate failure	This error indicates the given sensor is unreliable and has stopped producing data.
0x12	Accelerometer rate failure	
0x14	Gyroscope rate failure	

If the RAMVersion register values do not correspond to the expected Configuration File revision level, as given in Table 4-10, certain features or functions that are expected to be available may not be available, or they may not function as expected. This normally can be remedied by generating the latest Configuration File revision level using the SENtral Configuration Tool and then loading this into the onboard EEPROM, as discussed in the SENtral Technical Datasheet.

Table 4-10: RAMVersion Register Values

0x72 Register Value	0x73 Register Value	RAM Version (Hex / Decimal)	Config File Revision
0x04	0x0C	0x0C04 / 3076	1.0
0xD5	0x0C	0x0CD5 / 3285	1.1
0x37	0x0E	0x0E02 / 3639	1.2

5 Package Information

PIN	FUNC
1	DVDD
2	DVDD2
3	SCLS
4	GPIO6
5	SDAS
6	GPIO4
7	AVDD
8	GND
9	SDAM
10	SCLM

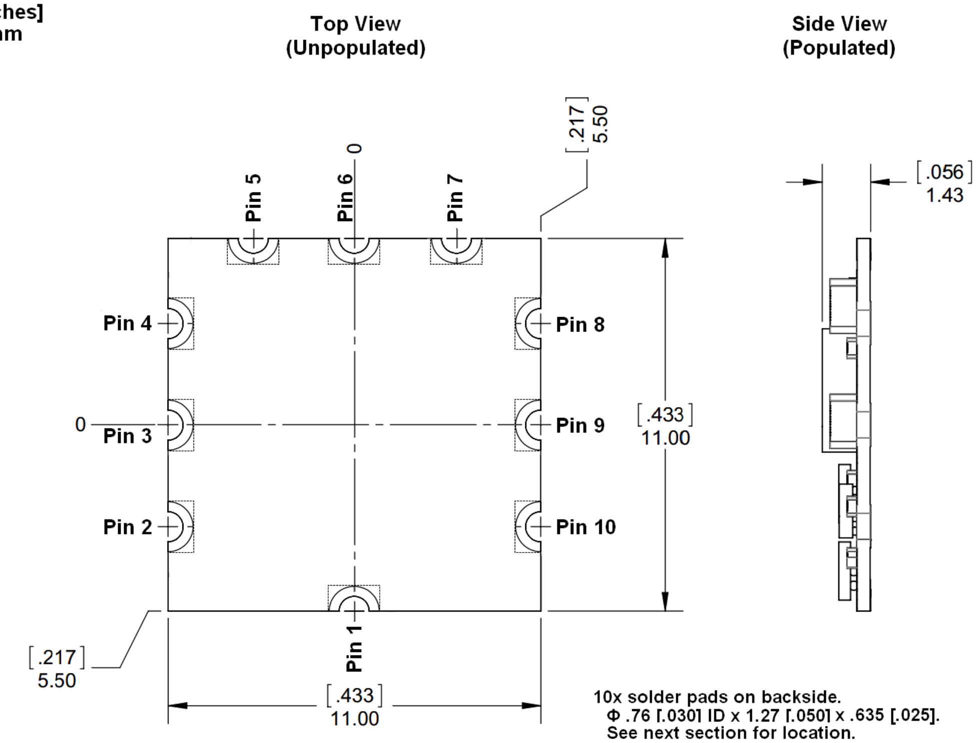

Figure 5-1: SENtral Orange, Red, Green, & Yellow M&M Mechanical Drawing

PIN	FUNC
A1	SDAS
A2	GPIO4
A3	GP1O2
A4	SDAM
B1	SCLS
B2	GPIO5
B3	GPIO3
B4	SCLM
C1	GPIO6
C2	RES
C3	SA0
C4	GPIO1
D1	DVDD
D2	GND
D3	VCAP
D4	GPIO0

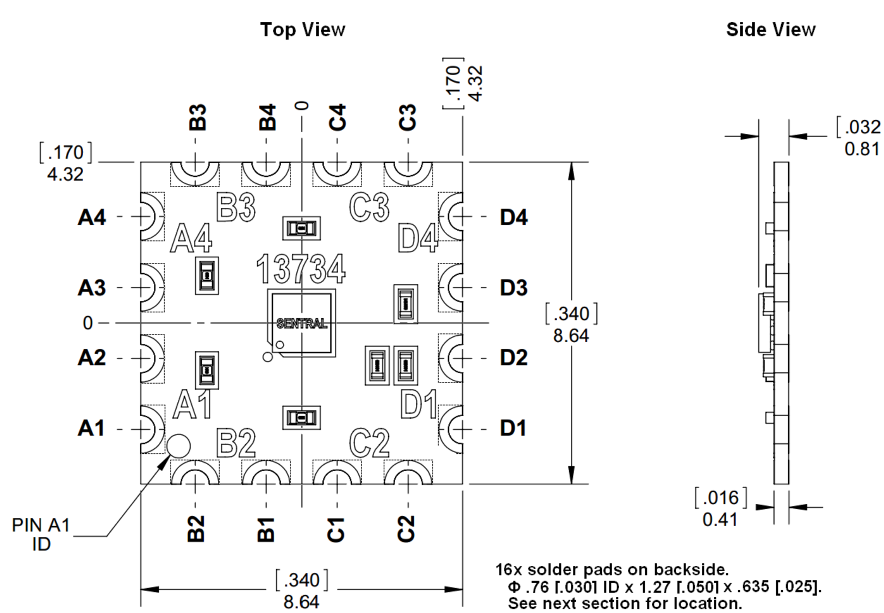

Figure 5-2: SENtral White M&M Mechanical Drawing

Dimensions in [inches] mm

PIN	FUNC
1	N/C
2	DVDD
3	SCLS
4	GPIO6
5	SDAS
6	GPIO4
7	AVDD
8	GND
9	SDAM
10	SCLM
11	GND
12	N/C
13	N/C

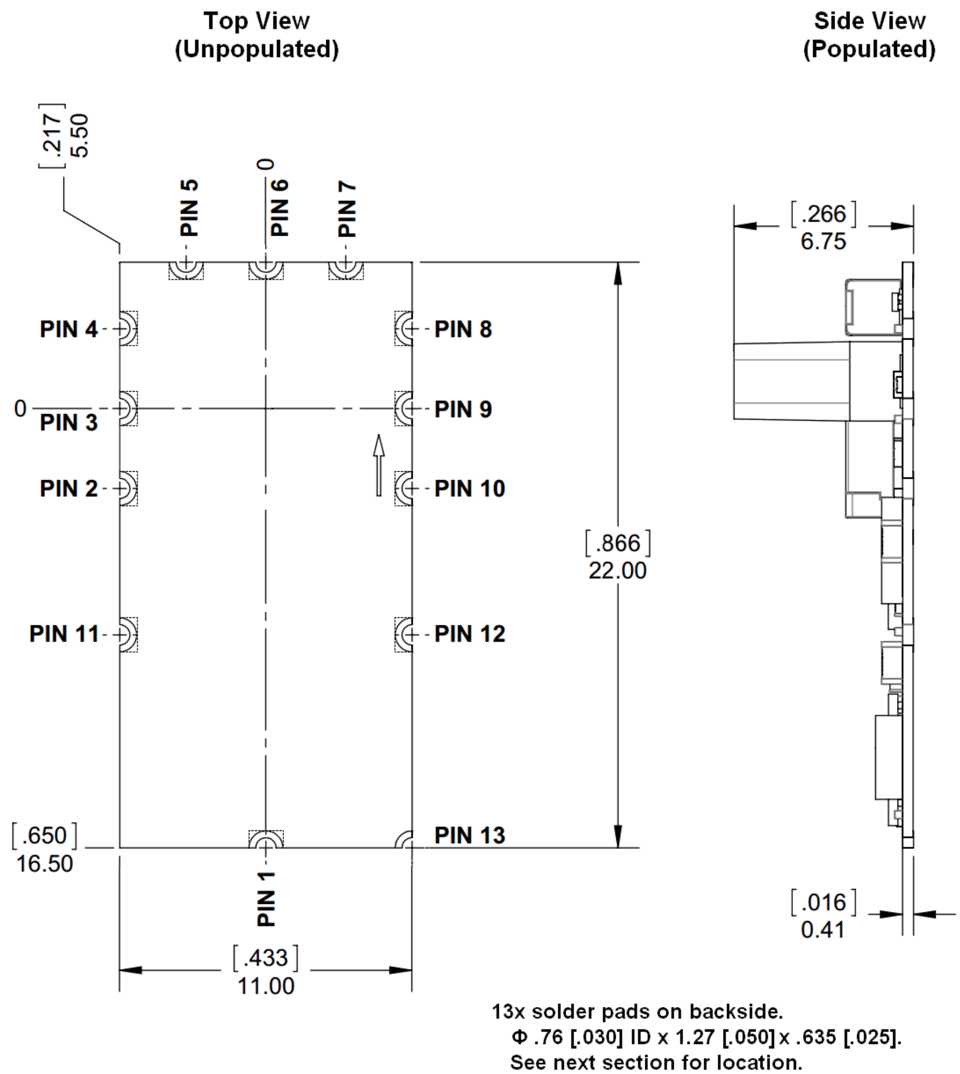

Figure 5-3: SENtral Blue M&M Mechanical Drawing

6 Assembly Guidelines

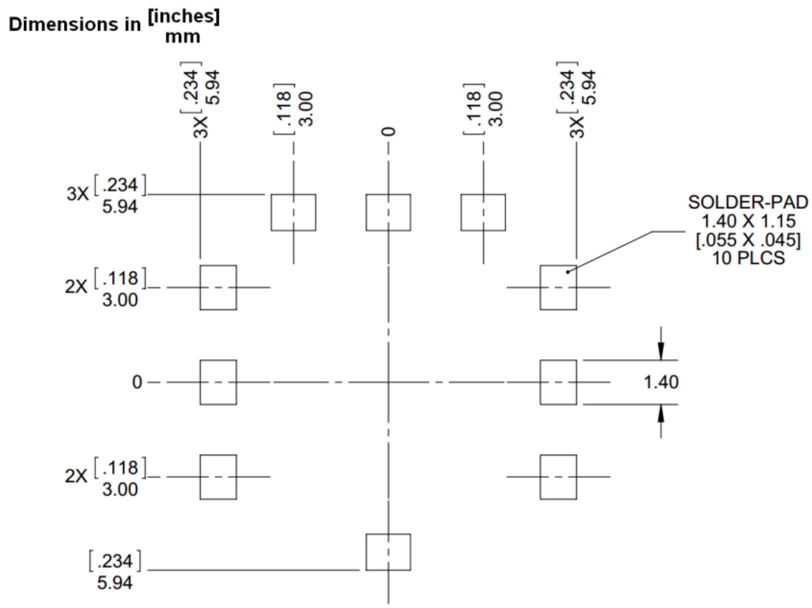

Figure 6-1: SENtral Orange, Red, Green, & Yellow M&M Solder Pad Layout

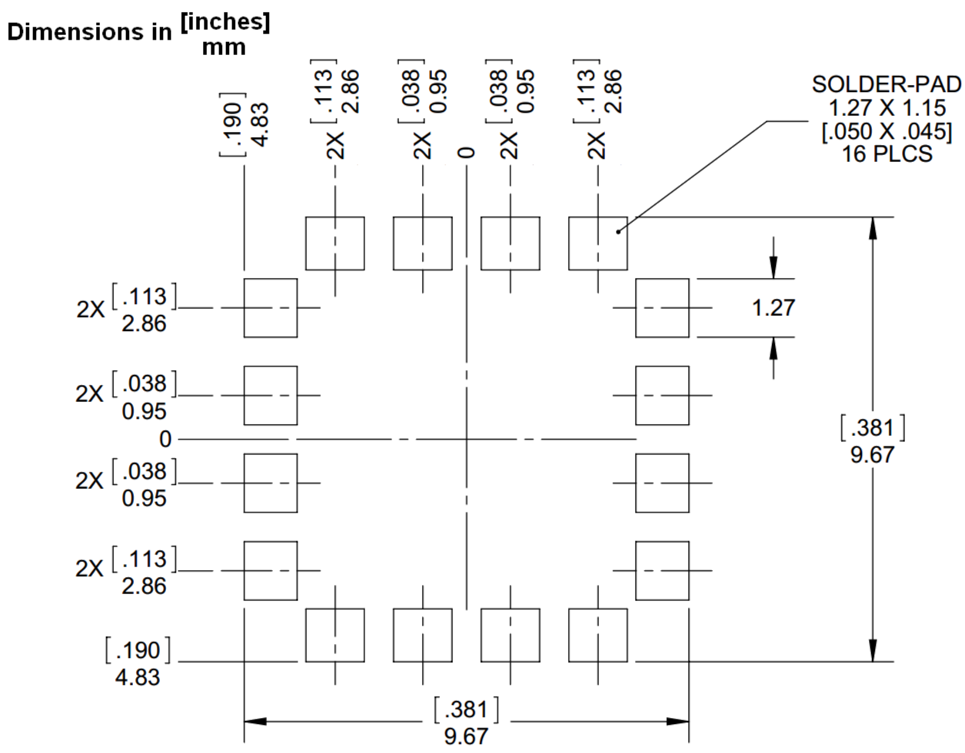

Figure 6-2: SENtral White M&M Solder Pad Layout

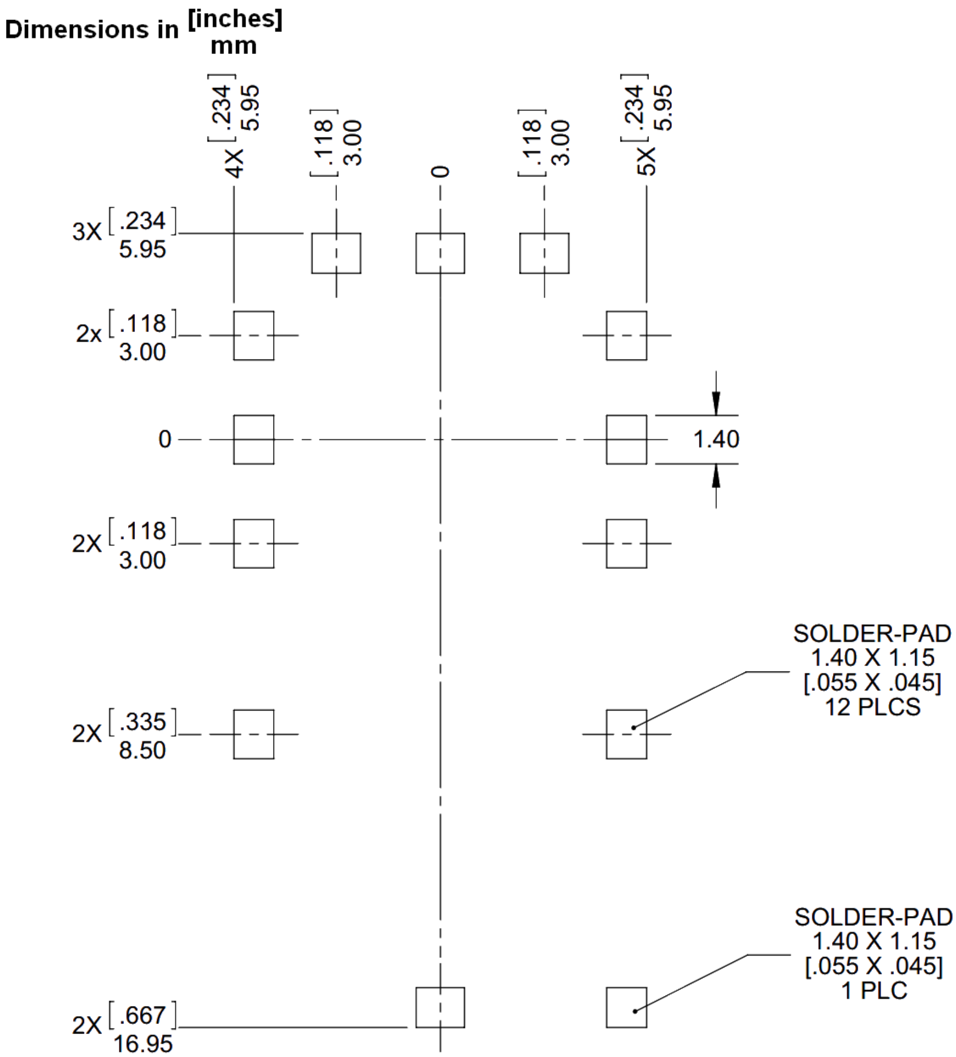

Figure 6-3: SENtral Blue M&M Solder Pad Layout

7 SENtral M&Ms with Pressure Sensors Overview

SENtral M&M Pink and M&M Purple include barometric pressure sensors. In addition to expanding the range of applications and features that the M&M's support, pressure sensors are an important component in accurate dead-reckoning and indoor navigation. The pressure sensors used also provide temperature.

Application Functionality

- Barometric pressure for weather forecasting
- Floor and elevator detection for indoor navigation
- Slope detection and altimeter for outdoor navigation
- Climbing speed

Ambient temperature
This data is available in addition to the SENtral quaternion and algorithm outputs.

7.1 System Schematics

Our M&M modules are an ideal way to test the functionality of a SENtral based system. Full size schematics are provided upon request for those wishing to integrate a discrete solution into their system.

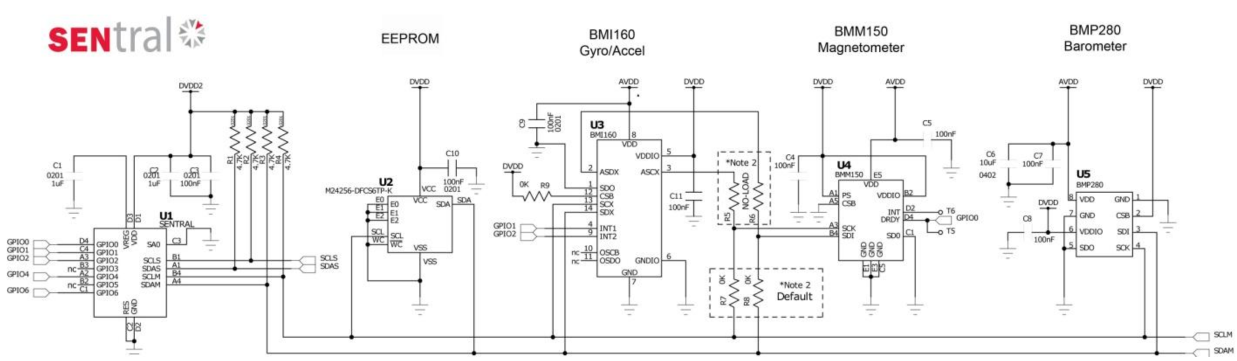

Figure 7-1: SENtral M&M Purple Schematic

Key components include PNI SENtral, Bosch Sensortech BMI160 Gyro/Accel , BMM150 Magnetometer and BMP280 Pressure sensor.

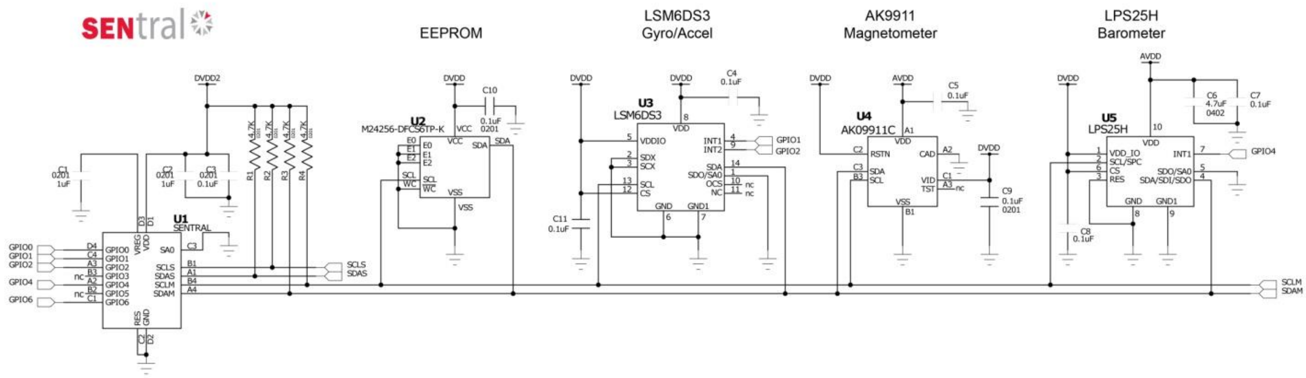

Figure 7-2: SENtral M&M Pink Schematic

Key components include PNI SENtral, ST Micro LSM6DS3 Gyro/Accel, AKM AK9911 Magnetometer and ST Micro LPS25H Pressure Sensor

7.2 Specifications

Sensor data available through SENtral is output at scale factors detailed in the following table;

Product	Scale Factor	Range
M&M Purple – Pressure	3Pa/LSB[1]	300-1100 hPa
M&M Purple – Temperature	0.01°C/LSB	-40 - 85°C
M&M Pink – Pressure	3Pa/LSB	260 – 1260 hPa
M&M Pink – Temperature	0.01°C/LSB	-40 – 85°C

1. Least Significant Bit

7.3 Interface

Pressure sensor data is available separate from the quaternion outputs and can be accessed without the need to wakeup the inertial sensors on the board. The data can be accessed at the following registers.

Table 7-1 Pressure Senor Data Accesss

Register Name	Register Address	Register Value
Feature0	0x2A-0x2B	Barometer Data
Feature1	0x2C-0x2D	Barometer Timestamp
Feature2	0x2E-0x2F	Temperature Data
Feature3	0x30-0x31	Temperature Timestamp

Appendix I – Converting Quaternions

SENtral outputs orientation data in quaternions, using a North-East-Down (NED) convention. This is done to avoid the singularities inherent in using Euler angles (heading, pitch, and roll), and because the fusion algorithms are easier to implement with quaternions. However, normally quaternions are not the desired final output format. Most end users will want heading, pitch, and roll, while Android looks for a rotation vector and generally uses a rotation matrix for orientation. Plus, Android and Win8 both expect data to be presented in the East-North-Up (ENU) convention. This appendix discusses how to convert SENtral's output quaternions into these other output formats.

Converting from NED to ENU

While the North-East-Down (NED) convention is common in many industries, both Android and Windows 8 use the East-North-Up convention. Below is the equation to convert from NED to ENU.

$$Q_{ENU} = \left(\begin{bmatrix} 0.707 & 0.707 & 0 & 0 \end{bmatrix} \begin{bmatrix} Qw & -Qz & Qy & -Qx \\ Qz & Qw & -Qx & -Qy \\ -Qy & Qx & Qw & -Qz \\ Qx & Qy & Qz & Qw \end{bmatrix}_{NED} \right) \begin{bmatrix} 0 & 0 & -0.707 & 0.707 \\ 0 & 0 & 0.707 & 0.707 \\ 0.707 & -0.707 & 0 & 0 \\ -0.707 & -0.707 & 0 & 0 \end{bmatrix}$$

Heading, Pitch, and Roll

Most end users will want orientation data reported as heading, pitch, and roll. Below are the Excel transformation equations. Note that for other programs, such as Matlab, the ATAN2 arguments may be reversed.

- Heading = atan2[$(Qx^2 – Qy^2 – Qz^2 + Qw^2)$, 2*(QxQy + QzQw)]
- Pitch = asin[-2*(QxQz – QyQw)]
- Roll = atan2[$(–Qx^2 – Qy^2 + Qz^2 + Qw^2)$, 2*(QxQw + QyQz)]

Where:

- Results are in radians.
- The quaternions are the outputs from SENtral in NED convention.
- Heading increases as the device rotates clockwise around a positive Z axis, and the range is 0° – 360°. (i.e. it matches what you would expect on a compass.)
- Pitch increases when pitching upward and the range is ±180°.
- Roll increases when rolling clockwise and the range is ±90°.

Rotation Vector

The rotation vector is the first three elements of the quaternion output, Qx, Qy, and Qz. The fourth element, Qw, is not included in the rotation vector. The rotation vector in ENU convention will be the first three elements of Q_{ENU}, discussed above.

Rotation Matrix, or Direction Cosine Matrix (DCM)

The rotation matrix, also known as the direction cosine matrix (DCM), can be established from the quaternion output using the following conversion. Q_{ENU} values can be substituted to give the rotation matrix with an ENU convention.

	$Qw^2 + Qx^2 – Qy^2 – Qz^2$	2*(Qx*Qy + Qw*Qz)	2*(Qx*Qz – Qw*Qy)
R =	2*(Qx*Qy – Qw*Qz)	$Qw^2 – Qx^2 + Qy^2 – Qz^2$	2*(Qy*Qz + Qw*Qx)
	2*(Qx*Qz + Qw*Qy)	2*(Qy*Qz – Qw*Qx)	$Qw^2 – Qx^2 – Qy^2 + Qz^2$

Appendix II – Parameter Transfer

Note: *Implementing the parameter transfer process is not necessary when using SENtral, but can be useful for enabling a warm start, for setting the sensor ranges to non-default values, and/or for reading the device driver IDs.*

This appendix provides the protocol for implementing SENtral's parameter transfer process. A parameter transfer involves the host either loading parameter values into SENtral, or retrieving parameter values currently used by SENtral.

Register Usage

Table A2-0-1 provides the registers used for the parameter transfer process.

Table A2-0-1: Registers Used for Parameter Transfer

Register Name	Address	Access	Format	Description
AlgorithmControl	0x54	R/W	[7:0]	[0] StandbyEnable. 1 = Enable Standby state [1] RawDataEnable. 1 = Raw data provided in MX, MY, MZ, AX, AY, AZ, GX, GY, & GZ. 0 = Scaled sensor data. [2] HPRoutput. 1 = Heading, pitch, and roll output in QX, QY, & QZ. QW = 0.0. 0 = Quaternion outputs. [7] ParamTransfer. 1 = Enable Parameter Transfer.
LoadParamByte0	0x60	R/W	Float 8	Parameter value to be loaded – LSB
LoadParamByte1	0x61	R/W	Float 8	Parameter value to be loaded – LSB + 1
LoadParamByte2	0x62	R/W	Float 8	Parameter value to be loaded – MSB – 1
LoadParamByte3	0x63	R/W	Float 8	Parameter value to be loaded – MSB
ParamRequest	0x64	R/W	[7:0]	Bits [0] – [6] provide the parameter number to be uploaded or retrieved. [7] Load/Save bit. 1 = Load, 0 = Save.
ParamAcknowledge	0x3A	R/O	[7:0]	Bits [0] – [6] provide the parameter number that was uploaded or retrieved. [7] Load/Retrieve bit. 1 = Load, 0 = Retrieve.
RetrieveParamByte0	0x3B	R/O	Float 8	Parameter value read from Sentral – LSB
RetrieveParamByte1	0x3C	R/O	Float 8	Parameter value read from Sentral – LSB + 1
RetrieveParamByte2	0x3D	R/O	Float 8	Parameter value read from Sentral – MSB – 1
RetrieveParamByte3	0x3E	R/O	Float 8	Parameter value read from Sentral – MSB

The parameter transfer process is invoked and terminated by appropriately setting the ParamTransfer bit in the AlgorithmControl register. Ten (10) registers are used for the transfer and for handshaking between SENtral and the host. One set of four registers is allocated to upload a parameter value to SENtral, and another set of four registers is used to retrieve a currently saved parameter from SENtral. Values shorter than four bytes can be transferred using only some of the registers. Two registers implement the handshake mechanism between SENtral and the host. Note that data is stored in little Endian format.

Parameter Load

Figure A2-0-1 shows the Parameter Load process by which the host loads parameter data into SENtral.

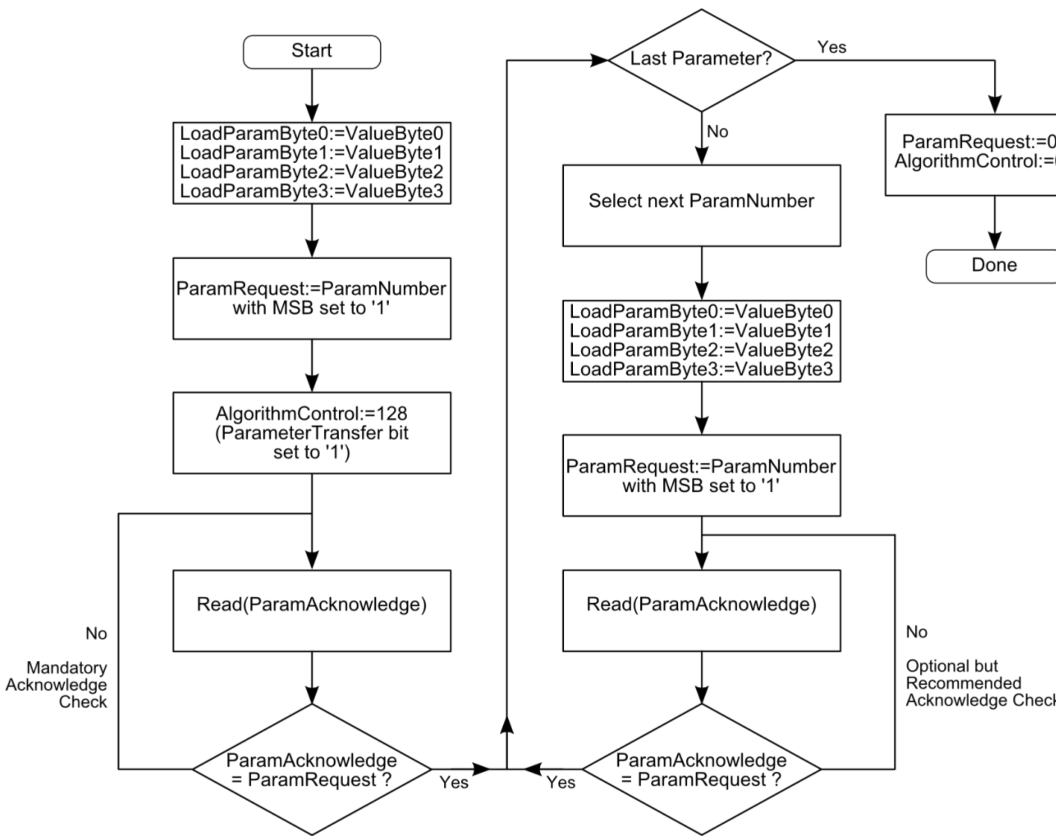

Figure A2-0-1: Parameter Load Process

Initially the parameter values must be written into the LoadParamByte registers followed by sending a non-zero parameter number into the ParamRequest register. The parameter numbers are given in Table A2-0-2. **The MSB of the ParamRequest register should be set to '1' to indicate a Load procedure.** All five bytes can be written using a single I^2C

transaction. **AFTER** the first parameter is written, the ParamTransfer bit in the AlgorithmControl register must be set to ‘1’. Sentral acknowledges receipt of a parameter value by setting ParamAcknowledge equal to ParamRequest, and the host should check the ParamAcknowledge register after writing the first parameter.

Once SENtral acknowledges successfully uploading the first parameter, the host can begin writing the remaining parameters in a loop. Reading the ParamAcknowledge register is optional for subsequent parameters. The host terminates the load procedure by setting the ParamRequest register to 0x00 and the AlgorithmControl register’s ParamTransfer bit to ‘0’.

Parameter Retrieve

The Parameter Retrieve flowchart is given in Figure A2-0-2.

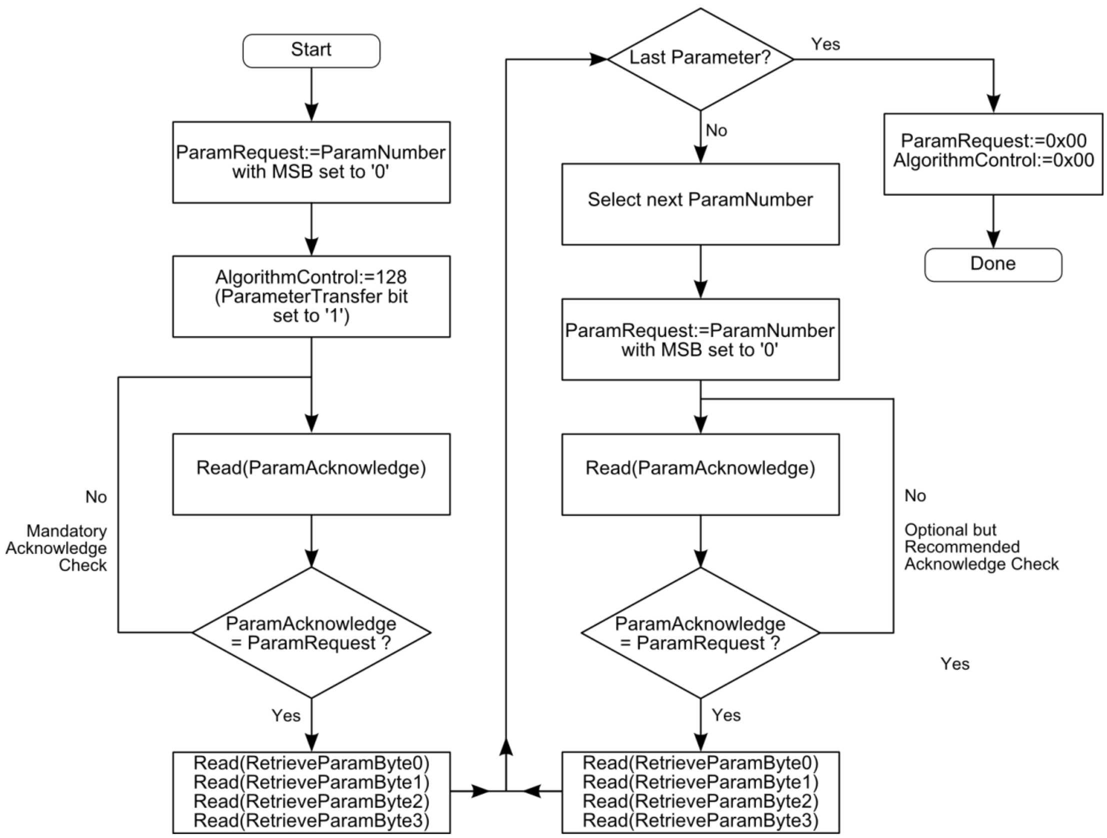

Figure A2-0-2: Parameter Retrieve Process

The process is initiated by the host writing to the ParamRequest register the desired (non-zero) parameter number. **The MSB of ParamRequest register should be ‘0’ to indicate a Retrieve procedure.** After writing to the ParamRequest register, the ParamTransfer bit in the AlgorithmControl register must be set to ‘1’. Next, the host should perform repetitive reads of the ParamAcknowledge register until it contains the requested parameter number.

Now the host can read the RetrieveParamByte registers to obtain the parameter value. Note the host can read the ParamAcknowledge and RetrieveParamByte registers using a single five-byte read transaction. Also, the RetrieveParamByte values are given in little Endian format, such that RetrieveParamByte3 contains the least significant byte of the parameter's 4-byte float value. The host can continue reading other parameters by varying (normally incrementing) the parameter number contained in the ParamRequest registers. Reading the ParamAcknowledge register is optional for subsequent parameters. The procedure is terminated by the host writing 0x00 to the ParamRequest and AlgorithmControl registers.

Interleaving Parameter Load and Retrieve

The host can interleave the Parameter Load and Parameter Retrieve processes during a single process invocation. This can be done for each parameter by setting the MSB bit of the ParamRequest register appropriately. Note that SENtral can be copying a new value into a RetrieveParamByte register while a Parameter Load operation is requested. Interleaving can be utilized by the host as an additional check that the parameter value was updated correctly.

Parameters

The parameter numbers and associated names are given below in Table A2-0-2. A discussion on the WarmStart, SensorRange, and DriverID parameters follows.

Table A2-0-2: Parameter Numbers

Parameter Number	Parameter Name	ParamRequest Value	
		Load	Retrieve
1 – 35	WarmStart[1] to WarmStart[35]	0x81 to 0xA3	0x01 to 0x23
36 – 73	Reserved		
74	SensorRange[mag:accel]	0xCA	0x4A
75	SensorRange[gyro]	0xCB	0x4B
77	DriverID[mag:accel]	--	0x4D
78	DriverID[gyro]		0x4E
80	AlgorithmID	--	0x50

WarmStart

A significant number of parameters are used in the SENtral algorithm as it executes, and these parameters are refined as the SENtral device is used. These include parameters associated with SENtral's continuous background calibration function and gyro bias correction. When SENtral is powered down or otherwise re-initialized, these parameters also are re-initialized and the parameter refinement process must start over. The parameter transfer process provides the ability to save these parameters to the host as they are refined, and to reload them if the parameters within SENtral are re-initialized. Thus, if the WarmStart parameters periodically are retrieved from SENtral and saved by the host, it is possible to effectively warm-start SENtral after it is re-initialized by reloading the WarmStart parameters into SENtral that previously were saved to the host.

To effectively enable a warm-start process, it is necessary to periodically save all 35 WarmStart parameters, and to reload all of them after SENtral is re-initialized.

SensorRange

The dynamic ranges of the sensors used in conjunction with SENtral normally are set as part of the Configuration File. Typically the gyroscope will be set to 2000 dps, the accelerometer to ±2 g or ±4 g, and the magnetometer to ±1 μT. However, there may be instances when it is desirable to change the dynamic range. For instance, if SENtral will be used in an application with frequent shock, such as jogging, it may be necessary to increase the accelerometer range to something greater than ±4 g.

SensorRange[mag:accel] loads or retrieves the magnetometer range data in ParamByte0 and ParamByte1, while the accelerometer range data is in ParamByte2 and ParamByte3. For example, a likely readout for SensorRange[mag:accel] in the 4x RetrieveParamByte registers is 0xE8030200, corresponding to a magnetometer dynamic range of 0x03E8 (±1000 μT) and an accelerometer dynamic range of 0x0002 (±2 g). SensorRange[gyro] loads or retrieves the gyroscope range in ParamByte0 and ParamByte1, while ParamByte2 and ParamByte3 are reserved and should be 0x00.

DriverID and AlgorithmID

Sensor driver and algorithm revision information can be retrieved using the Parameter Transfer process. Table A2-0-3 indicates how these parameters are defined. ParameterBytes 2 and 3 for Parameter Numbers 78 and 80 are 0x00 and reserved for future use.

Table A2-0-3: DriverID & AlgorithmID Definition

Parameter Number	Parameter Name	Parameter Byte	Usage
77	DriverID[mag:accel]	0	Mag Driver Revision
		1	Mag Driver ID
		2	Accel Driver Revision
		3	Accel Driver ID
78	DriverID[gyro]	0	Gyro Driver Revision
		1	Gyro Driver ID
80	AlgorithmID	0	Algorithm Revision, Minor
		1	Algorithm Revision, Major

Appendix III – Measuring Current Consumption

All SENtral M&M modules, except the White and Blue versions, have two distinct electrical supply lines. One line is for both the EEPROM and the sensors, and one is for just SENtral. The pins for these voltages are labeled DVDD and DVDD2, respectively. To measure the current on these lines, PNI recommends placing a 1 Ω resistor in series with the DVDD pin to measure combined current consumption for the EEPROM and sensors, and a 100 Ω resistor in series with the DVDD2 pin to measure current consumption by SENtral.

The SENtral Blue M&M has a single DVDD pin that supplies current for SENtral, the EEPROM and the sensors. However, the current consumption of only the SENtral Motion Coprocessor can be measured by modifying the module, as given in the two options listed below.

1. Replace a zero-ohm resistor with a 100Ω resistor and measure voltage across the resistor.
2. Remove the zero-ohm resistor, then solder wires in series with a connected ammeter.

The location of the zero-ohm resistor is given below, and a discussion of the two implementation methods follows.

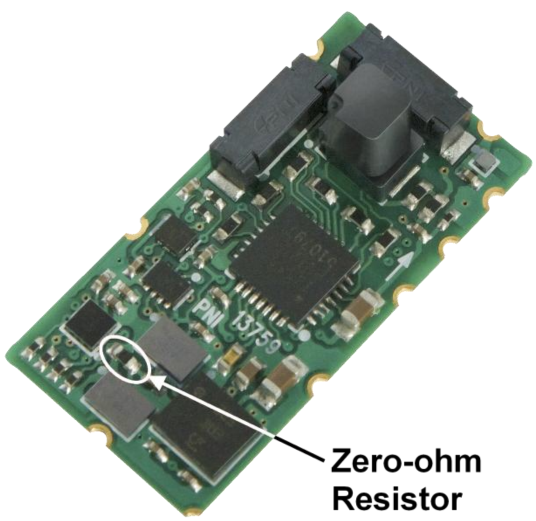

Figure A3-0-1: SENtral Blue M&M Zero-Ohm Resistor Location

Method 1: Replace zero-ohm resistor with 100 Ω resistor.

This method provides flexibility in terms of measuring with either a voltmeter or an oscilloscope, although it may be slightly difficult to implement as holding the probes in the proper position can be tricky. As long as the resistor is ≤100 Ω, there is no need to remove it, as it should not affect performance.

To measure average current consumption, simply touch either side of the 100 Ω resistor with the voltmeter's probe tips and measure the voltage drop. Convert to current consumption using: μA = 10*mV, assuming a 100 Ω resistor.

It is possible to observe the current consumption waveform using an oscilloscope. In this case, place a 100 μF capacitor in parallel with the 100 Ω resistor. This reduces the measurement bandwidth so the waveform can be better observed.

Note that SENtral's bypass capacitors are electrically connected nearest the device after the sense resistor or the voltage meter's resistor. This will bandlimit the measurement to ~1.5 kHz for a 100 Ω resistor. The onboard bypass capacitance totals 1.1 μF.

Method 2: Remove zero-ohm resistor and place ammeter in series.

This method is relatively straight forward to implement, as the probes are physically soldered to the PCB. To help prevent damage to the PCB surface pads, PNI strongly recommends implementing a strain relief for the wires.

Note that the burden voltage of a typical digital multimeter (ammeter) is ~100μV/μA, or 100 Ω. PNI has tested such an ammeter in the Method 2 scenario and seen that it does not affect operation. Also note that negative voltages produced by transient currents are smoothed by the local bypass capacitors.

Also, it may be difficult to measure DC current using ammeters with very fast measurement times due to the periodic wake/sleep cycles of SENtral. Consequently, handheld DMMs with relatively long measurement integration times work well for making average current measurement. Precision benchtop meters with an averaging or smoothing filter also can work well.

Revised April 2014: for the most recent version of this document visit our website at www.pnicorp.com

PNI Sensor Corporation
2331 Circadian Way
Santa Rosa, CA 95407, USA
Tel: (707) 566-2260
Fax: (707) 566-2261

Warranty and Limitation of Liability. PNI Sensor Corporation ("PNI") manufactures its Products from parts and components that are new or equivalent to new in performance. PNI warrants that each Product to be delivered hereunder, if properly used, will, for ninety (90) days following the date of shipment unless a different warranty time period for such Product is specified: (i) in PNI's Price List in effect at time of order acceptance; or (ii) on PNI's web site (www.pnicorp.com) at time of order acceptance, be free from defects in material and workmanship and will operate in accordance with PNI's published specifications and documentation for the Product in effect at time of order. PNI will make no changes to the specifications or manufacturing processes that affect form, fit, or function of the Product without written notice to the Customer, however, PNI may at any time, without such notice, make minor changes to specifications or manufacturing processes that do not affect the form, fit, or function of the Product. This warranty will be void if the Products' serial number, or other identification marks have been defaced, damaged, or removed. This warranty does not cover wear and tear due to normal use, or damage to the Product as the result of improper usage, neglect of care, alteration, accident, or unauthorized repair.

THE ABOVE WARRANTY IS IN LIEU OF ANY OTHER WARRANTY, WHETHER EXPRESS, IMPLIED, OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, ANY WARRANTY OF MERCHANTABILITY, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION, OR SAMPLE. PNI NEITHER ASSUMES NOR AUTHORIZES ANY PERSON TO ASSUME FOR IT ANY OTHER LIABILITY.

If any Product furnished hereunder fails to conform to the above warranty, Customer's sole and exclusive remedy and PNI's sole and exclusive liability will be, at PNI's option, to repair, replace, or credit Customer's account with an amount equal to the price paid for any such Product which fails during the applicable warranty period provided that (i) Customer promptly notifies PNI in writing that such Product is defective and furnishes an explanation of the deficiency; (ii) such Product is returned to PNI's service facility at Customer's risk and expense; and (iii) PNI is satisfied that claimed deficiencies exist and were not caused by accident, misuse, neglect, alteration, repair, improper installation, or improper testing. If a Product is defective, transportation charges for the return of the Product to Customer within the United States and Canada will be paid by PNI. For all other locations, the warranty excludes all costs of shipping, customs clearance, and other related charges. PNI will have a reasonable time to make repairs or to replace the Product or to credit Customer's account. PNI warrants any such repaired or replacement Product to be free from defects in material and workmanship on the same terms as the Product originally purchased.

Except for the breach of warranty remedies set forth herein, or for personal injury, PNI shall have no liability for any indirect or speculative damages (including, but not limited to, consequential, incidental, punitive and special damages) relating to the use of or inability to use this Product, whether arising out of contract, negligence, tort, or under any warranty theory, or for infringement of any other party's intellectual property rights, irrespective of whether PNI had advance notice of the possibility of any such damages, including, but not limited to, loss of use, revenue or profit. In no event shall PNI's total liability for all claims regarding a Product exceed the price paid for the Product. PNI neither assumes nor authorizes any person to assume for it any other liabilities.

Some states and provinces do not allow limitations on how long an implied warranty lasts or the exclusion or limitation of incidental or consequential damages, so the above limitations or exclusions may not apply to you. This warranty gives you specific legal rights and you may have other rights that vary by state or province.

Revision Control Block

Revision	Description of Change	Effective Date	Approval
A	Initial Preliminary Release	Dec. 4, 2013	A.Leuzinger
B	Renamed product to SENtral M&M. Rewrote front page. Removed Performance Characteristics table. Updated and moved pin-out table. Added White & Green mechanical drawings and pad layouts. Updated Blue, Red, & Yellow (LSM330) mechanical drawing and pad layout. Added Red electrical schematic. Added TOC, List of Figures, and List of Tables.	Dec. 30, 2013	A. Leuzinger
C	Changed picture on front page. Added M&M Orange module. Added Blue electrical schematic. Removed castellated board from Red & Green schematics.	Jan. 3, 2014	A. Leuzinger
D	Renamed product description to be "Motion & Measurement Module(s)". Renamed module types by putting color prior to "M&M". Added Sections 1, 3, and 4, and Appendix II. Added Table 2-1, Performance Characteristics. Corrected Operating Conditions and Pin Assignment tables regarding DVDD & DVDD2. Removed Test Point references from Pin Assignment Table and from Mechanical Drawings. Corrected solder pad references on Mechanical Drawings. Removed the section with specific schematics and added a note that schematics are available from PNI. (Section 1.2 has a "generic" schematic.)	Jan. 30, 2014	A. Leuzinger
E	Pertains to both SDK1.1 (build 3285) and SDK1.2 (build 3639). Corrected Ordering Information, Yellow M&M to read LSM9DS0. Added "TM" to "M&M". In Section 3.2.1, changed the notation in the figures so not every bit begins with "A". Added sentence at beginning of Section 4 regarding Little Endian formatting. Added note in Section 5.2 about requirement to set [Sensor]Rate registers to non-zero values. Corrected Table 4-1 to indicate bit [2] of SentralStatus is '1' when the CRC is ***incorrect***. In Section 5.6.2, added discussion of RAMVersion registers and added Table 5-13. Added Appendix II on Parameter Transfer.	Apr. 16, 2014	A. Leuzinger
F	Corretion to DCM conversion table, cell 2,3 and 3,2	7/2/14	R Thompson D McKenzie
G	Addition of M&M Pink and Purple, required new section (7) to detail functionality of additional pressure sensor		