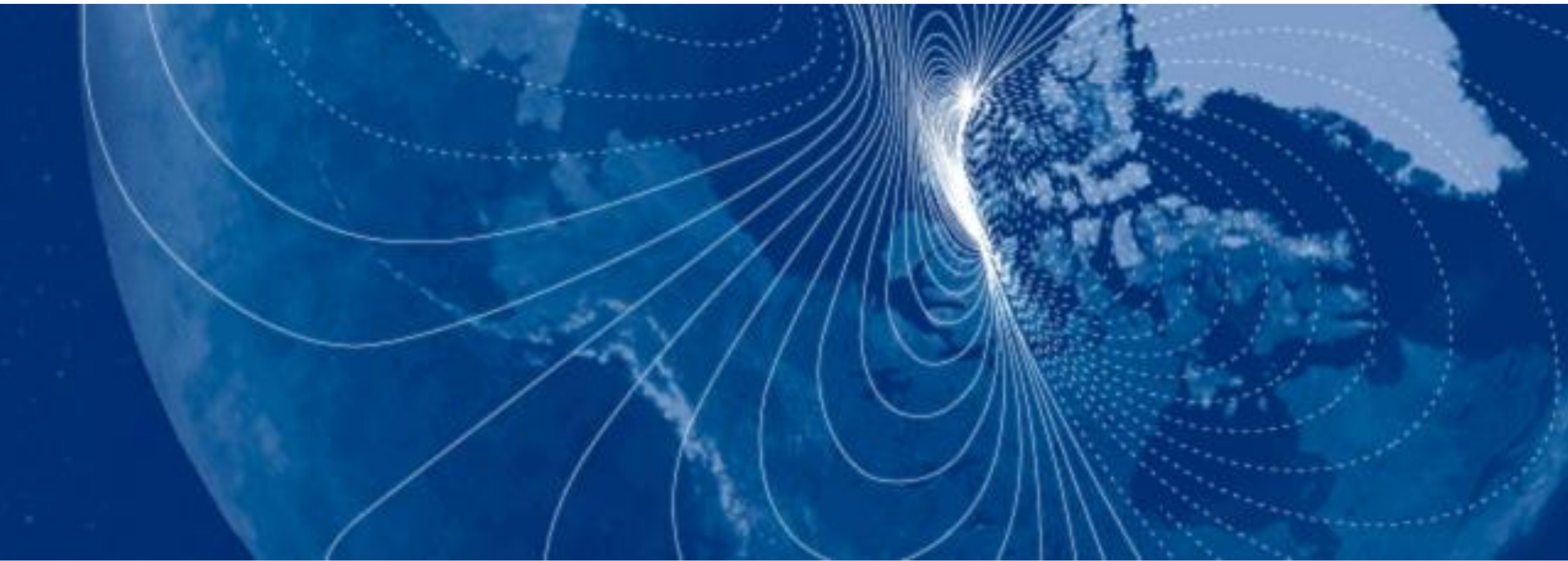


User Manual

SENtral Development Kit

PNI
SENSOR CORPORATION

Table of Contents

List of Tables

List of Figures

1. Copyright & Warranty Information

Revised June 2014. For most recent version 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 ("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 one year 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 OEM, 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, OEM's sole and exclusive remedy and PNI's sole and exclusive liability will be, at PNI's option, to repair, replace, or credit OEM's account with an amount equal to the price paid for any such Product which fails during the applicable warranty period provided that (i) OEM 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 OEM'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 OEM 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 OEM'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.

2. Introduction & Quick Start

Thank you for purchasing PNI's SENtral Development Kit, pn 90058. The SENtral Development Kit allows developers the ability to easily and quickly evaluate the SENtral sensor fusion coprocessor, and assists system designers in understanding how to integrate SENtral into their system. The SENtral Development Kit consists of a:

- SENtral Development Board,
- Micro-USB-to-USB (Standard A type) Cable,
- 2 ea Shunts,
- 3x 3" Jumper Wires, and
- USB Thumb Drive containing:
    - SENtral Utility Tools:
        - RunConfigFileLoader.bat
        - Configuration file uploading tool: i2cCom.exe
        - EEPROM programming tool: EEPROMprogrammer.exe
    - SENtral Demo Program: SentralDemo_x.exe, where x = revision
    - Sample code IAR project
    - User Manual.
- On a case-by-case basis, PNI also may supply one or more sensor modules for use in conjunction with the SENtral Development Kit.

The SENtral Demo Program makes qualitative analysis of SENtral simple and straightforward with an intuitive GUI interface. Additionally the Demo Program enables data logging for quantitative orientation-tracking analysis.

The SENtral Development Board enables the user to easily change sensors by swapping out sensor modules which are based on the Qualcomm Gen7 sensor card design. The development board allows for CMOS-level UART, UART VCP over USB, or USB communication with its on-board STM32L151RD 32-bit ARM microprocessor. Various test points allow for effortless monitoring of the voltage lines, digital signal lines, and current draw. Debugging is aided by the incorporation of a JTAG connector, while reprogramming of the microprocessor is possible via the JTAG or the microprocessor's bootloader feature. The development board can be powered either from a micro-USB connector or from external DC voltages.

Getting Started with the Development Kit and an All-in-One Sensor Module

First confirm you are using the development board with an all-in-one sensor module that contains SENtral, a 3-axis magnetometer, a 3-axis accelerometer, and a 3-axis gyroscope. For more on the sensor modules, see Section 4.1.

- Plug the all-in-one sensor module into the SENtral Master Socket (E) on the SENtral Development Board.
- Ensure the VCP/USB shunt on the development board is set to select for UART VCP communication, and that the VCP driver for FTDI's FT232R IC is installed on your PC. See Section 4.2.1 for more information.
- Connect the SENtral Development Board to your PC using the micro-USB-to-USB (Standard A type) cable.
- Launch and run the SENtral Demo Program, as discussed in Section 6.

Getting Started with the Development Kit and Multiple Sensor Modules

First confirm you are using the development board with multiple sensor modules, including a SENtral-only sensor module and additional sensor modules that incorporate a 3-axis magnetometer, a 3-axis accelerometer, and a 3-axis gyroscope. For more on the sensor modules, see Section 4.1.

- Generate the appropriate SENtral Configuration File for your specific sensors and sensor configurations. The Configuration File is generated by the SENtral Configuration Tool, which is accessed at www.SentralToolKit.com. The tool is for SENtral customers, and registration is required. Registration requests are reviewed at PNI. Typically it takes less than one business day to activate a registration request. See the SENtral Technical Datasheet for more information.
- Plug the SENtral-only sensor module in the SENtral Master Socket (E) on the SENtral Development Board, and plug the other sensor modules into the Sensor Module Sockets, A through D. Any of the Sensor Module Sockets may be used, as long as the combination of sensor modules includes a 3-axis magnetometer, a 3-axis accelerometer, and a 3-axis gyroscope.
- Install the Interrupt-to-GPIO jumper cables, as discussed in Section 4.1.2. Note that which sensor interrupt is required at each GPIO was defined when the SENtral Configuration File was generated with the Configuration Tool.
- Ensure the VCP/USB shunt on the development board is set to select for UART VCP communication, and that the VCP driver for FTDI's FT232R IC is installed on your PC. See Section 4.2.1 for more information.
- Connect the SENtral Development Board to your PC using the micro-USB-to-USB (Standard A type) cable.
- Write the Configuration File into one of the development board's EEPROMs, as discussed in Section 5.
- Launch and run the SENtral Demo Program, as discussed in Section 6.

3. Specifications

Table 3-1: Development Board Specifications

Parameter		Value
Communication Interface	Micro-USB	UART VCP or USB
	6-pin SIP	UART
Protocol		ASCII / Binary
Dimensions (*l* x *w* x *h*)	with standoffs	124.0 x 85.9 x 19.8 mm
	without standoffs	124.0 x 85.9 x 15.8 mm
Connectors		Micro-USB, and 6-pin Right-Angle SIP

Dimensions in mm [inches]

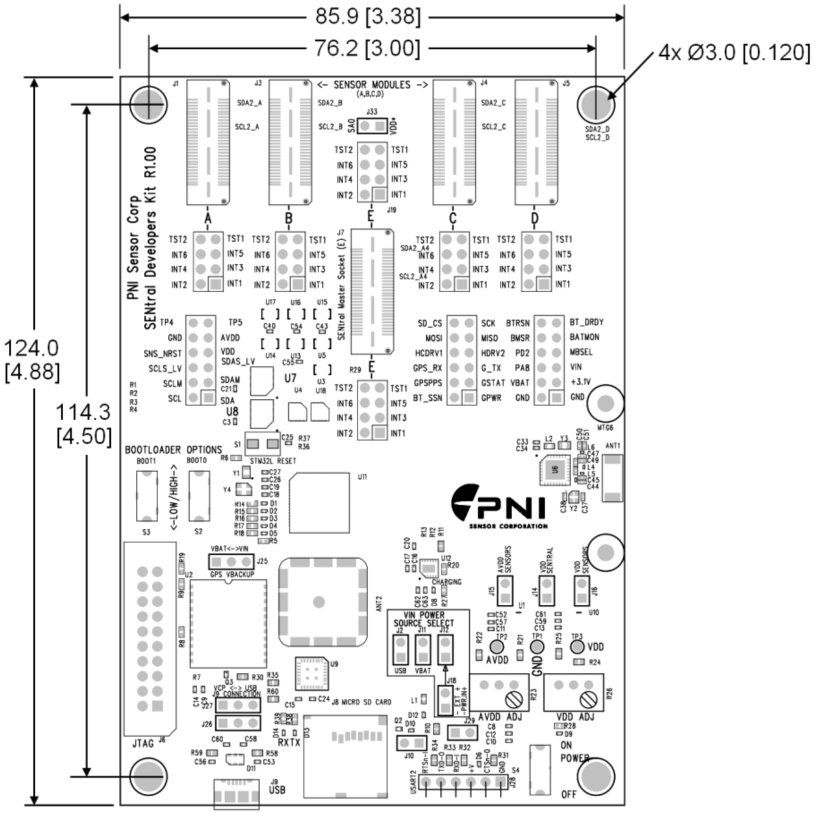
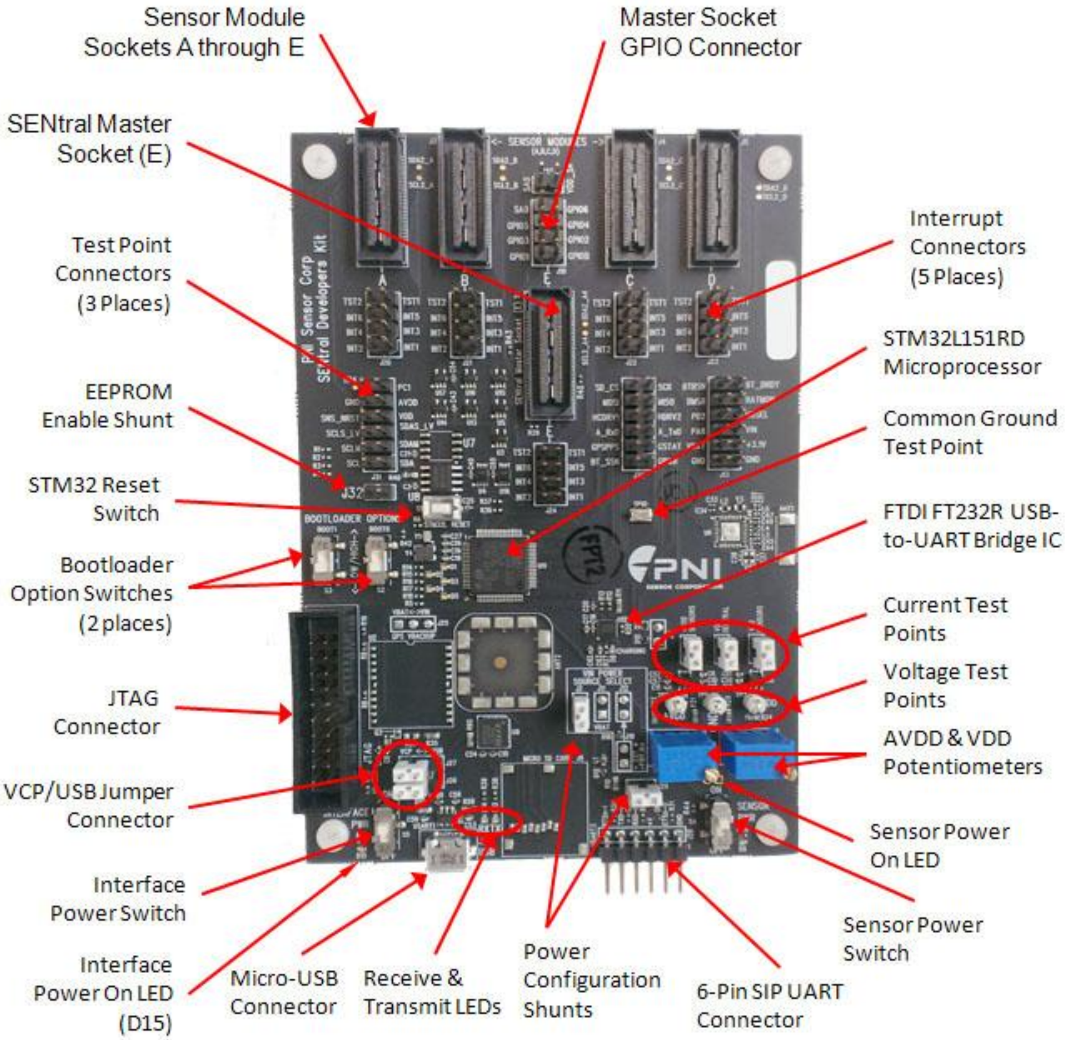


Figure 3-1: Sentral Development Board Dimensions

4. Development Board Features

The SENtral Development Board incorporates many features which are discussed below and in the following subsections. Figure 4-1 highlights these various features.

> ***NOTE:*** *Figure 4-1 and the following descriptions are not applicable to hardware revisions A-D. Please reference revB of this manual for those revisions.*



Sensor Module Sockets A through E
Master Socket GPIO Connector
SENtral Master Socket (E)
Test Point Connectors (3 Places)
Interrupt Connectors (5 Places)
STM32L151RD Microprocessor
EEPROM Enable Shunt
Common Ground Test Point
STM32 Reset Switch
FTDI FT232R USB-to-UART Bridge IC
Bootloader Option Switches (2 places)
Current Test Points
Voltage Test Points
JTAG Connector
AVDD & VDD Potentiometers
VCP/USB Jumper Connector
Sensor Power On LED
Interface Power Switch
Sensor Power Switch
Interface Power On LED (D15)
Micro-USB Connector
Receive & Transmit LEDs
Power Configuration Shunts
6-Pin SIP UART Connector

Figure 4-1: SENtral Development Board Features

A summary of the various features is given below:

- **Sensor Module Sockets, A through E:** These sockets allow the user to easily swap sensor modules on the development board. The sensor modules should conform to Qualcomm's Gen7 sensor card design. Contact PNI for additional information on the sensor modules.

- **SENtral Master Socket (E):** This socket will hold either a SENtral-only sensor module, or an all-in-one sensor module that contains SENtral, a 3-axis gyroscope, 3-axis accelerometer, and 3-axis magnetometer. Note this is generally considered one of the sensor module sockets, albeit a special one.
- **Master Socket GPIO Connector:** This is used to jumper interrupts from the sensor module interrupts to the GPIO inputs on SENtral.
- **Interrupt Connectors:** These 5 connectors are used to jumper interrupts from the sensor modules to the GPIO inputs on SENtral.
- **STM32L151RD Microprocessor:** This microprocessor communicates with either a PC via USB or UART VCP, or an external processor via CMOS-level UART, and acts as the master device when communicating with SENtral on the host I^2C bus.
- **STM32 Reset Switch:** This switch is directly connected from ground to the RSTN pin on the STM32 microprocessor. Depressing this button causes a hardware reset on the STM32 microprocessor.
- **Test Point Connectors:** These 3 connectors allow for easy monitoring of various voltages, microprocessor lines, and SENtral Master Socket (E) lines.
- **FTDI FT232R IC:** This IC acts as a USB-to-UART bridge between the USB output of a PC and the UART interface of the STM32 microprocessor when the development board is configured for UART VCP operation. These UART signals are connected to the STM32 microprocessor’s USART#2 TX and RX pins.
- **Bootloader Option Switches:** These switches enable the bootloader functions of the STM32 microprocessor. Their default settings are both LOW. These switches are tied directly to STM32’s BOOT0 and BOOT1 input pins.
- **JTAG Connector:** Used for debugging and uploading firmware to the microprocessor.
- **VCP/USB Jumper Connector:** The shunts across this connector establish whether communication across the micro-USB connector will be UART VCP or USB. In UART VCP mode the UART signals are connected to the STM32’s USART#2 TX and RX pins via the FTDI FT232R.
- **Receive & Transmit LEDs:** Indicate when data is being transmitted to/from the STM32 microprocessor via the UART VCP.
- **Micro-USB Connector:** Provides power and a communication interface between the SENtral Development Board and a PC.
- **6-Pin SIP UART Connector:** Provides power and a 3.1 V CMOS logic-level UART communication interface to the STM32’s USART#1 TX and RX pins. Note that some revisions of the development board’s PCB incorrectly identify this as USART#2. This port is generally used when working with an external microprocessor.
- **Power Configuration Shunts:** Currently these shunts should be left in place and not removed. At some future time they may be used to enable additional features.

- **Sensor Power Switch:** Switching this to the ON position provides power to the sensor module sockets. Note that switching it to the OFF position does turn off power to the sensor module sockets but does not turn off power to the microprocessor, etc (see Interface Power Switch)
- **Sensor Power ON LED:** This will be ON when power is applied to the sensor module sockets.
- **AVDD and VDD Potentiometers:** These potentiometers give the user the ability to adjust the analog and digital (IO) supply voltages, AVDD and VDD, respectively, on the sensor module sockets.
- **Voltage Test Points:** These test points allow for easy monitoring of the analog (AVDD) and digital (VDD) voltages across the sensor module sockets.
- **Current Test Points:** These test points allow for monitoring of the current on the digital line to the SENtral Master Socket (VDD Sentral), the digital lines to the other sensor sockets (VDD Sensors), and the analog lines (AVDD Sensors) to all the sensor sockets.
- **Interface Power Switch:** Switching this power ON provides power to the STM32L microprocessor used to interface the host to the SENtral, one side of the level-translator ICs and all associated interface circuits. NOTE: This does not remove power to the battery management IC, U12. See Section 4.4 for more information.
- **Interface Power ON indicator LED:** This LED indicates the presence of power from the Battery Management IC, U12, to the interface circuitry, mainly the STM32L microprocessor. If this LED fails to turn on, a fault has been detected by the Battery Management IC U12. See Section 4.4 for more information.
- **EEPROM Enable Shunt:** J23, when shorted with a jumper (100mil header shunt), enables the on-board EEPROM U8 on the SENtral Master I^{2}C Bus. When enabled this EEPROM is addressable at address 0xA8 (7-bit: 0x54).

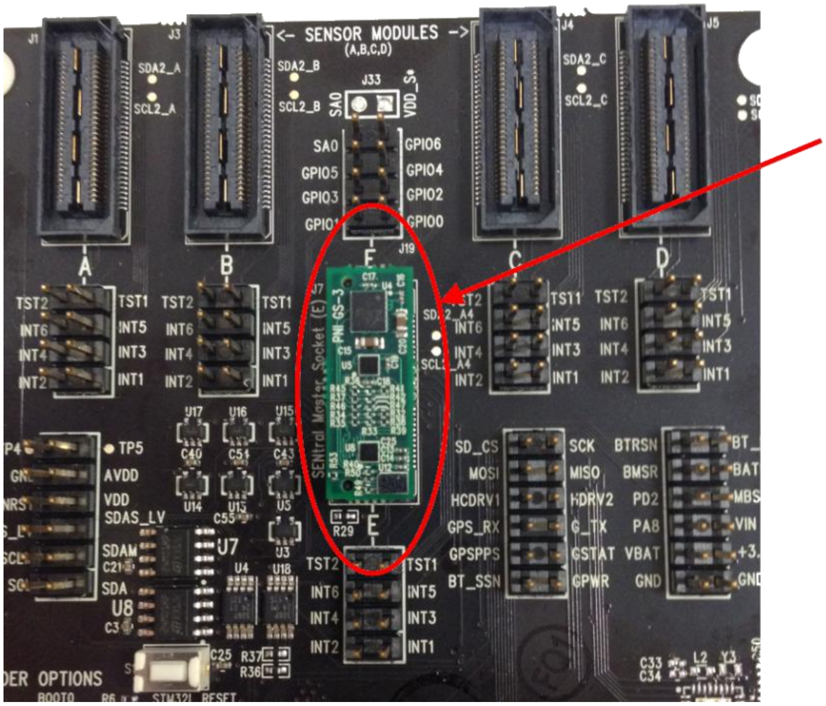
4.1. Sensor Module Installation

The SENtral Development Board allows the user to select which sensors to use in conjunction with the SENtral coprocessor. The sensors and SENtral are mounted on sensor modules which comply with Qualcomm Gen7 sensor card requirements. There are two methods for implementing sensor selection: 1) all-in-one sensor modules, or 2) multiple sensor modules.

4.1.1. All-in-One Sensor Module Approach

For the all-in-one sensor module approach, all the sensors are mounted on a single sensor module, along with SENtral, and this all-in-one sensor module is placed in the SENtral

Master Socket (E). Such an implementation is shown in Figure 4-2. In this specific case, the all-in-one sensor module contains SENtral, a combo accelerometer/gyroscope, and a discrete magnetometer.



All-in-One Sensor Card Containing:
SENtral
3-axis Magnetometer
3-axis Accelerometer
3-axis Gyroscope

Figure 4-2: All-in-One Sensor module Mounted on Development Board

4.1.2. Multiple Sensor Modules Approach

For the multiple sensor modules approach, SENtral is mounted on the PNI-supplied SENtral-only sensor module and placed in the SENtral Master Socket (E), while the sensors are mounted on one or more sensor modules and placed in sensor module sockets, A through D. Note that 1, 2, 3, or all 4 sockets may be populated, but combined they must include a magnetometer, accelerometer, and gyroscope. The user must ensure that none of the sensors share an I2C slave address. For the multiple sensor modules approach it is necessary to install jumpers to connect the interrupt signals from the magnetometer, accelerometer, and gyroscope to SENtral's GPIO inputs. An example of such an implementation is shown in Figure 4-3, and a discussion of the jumpers follows.

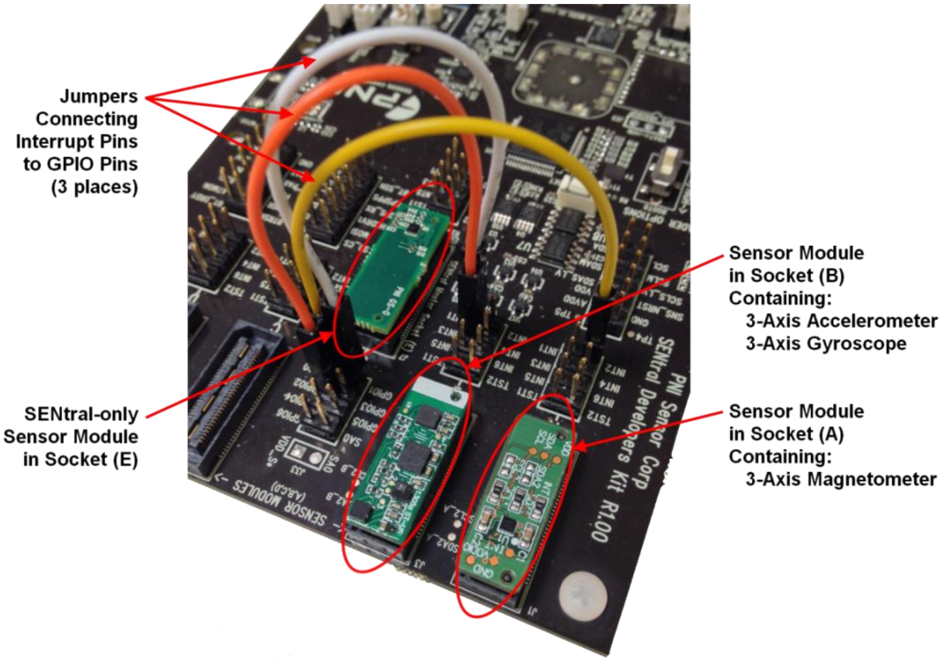

Jumpers Connecting Interrupt Pins to GPIO Pins (3 places)
Sensor Module in Socket (B) Containing:
3-Axis Accelerometer
3-Axis Gyroscope
SENtral-only Sensor Module in Socket (E)
Sensor Module in Socket (A) Containing:
3-Axis Magnetometer
PNI Sensor Corp
SENtral Developers Kit R1.00
SENSOR MODULES

Figure 4-3: Multiple Sensor Modules Mounted on Development Board

To configure the jumpers for a multiple sensor modules implementation requires properly connecting the interrupt pins associated with the sensor modules, to the correct GPIO pins on the sensor-bus-side of the SENtral Master Socket (E). The SENtral Configuration File (see SENtral Technical Datasheet) establishes the GPIO assignments for the magnetometer, accelerometer, and gyroscope. The connector pins on the sensor modules correspond to the INT pins as given in Table 4-1.

Table 4-1: Sensor Module Jumper Connector Pin-Out

Jumper Connector Label*	Sensor Module Connector Pin #
INT1	30
INT2	32
INT3	26
INT4	28
INT5	20
INT6	46
TST1[1]	48
TST2[1]	22

Footnote:

1. TST1 and TST2 functions per Qualcomm Gen7 board definition.

For the example shown in Figure 4-3:

- The SENtral Configuration File has been generated so the sensor GPIO assignments match the defaults, which are the magnetometer on GPIO[0], the accelerometer on GPIO[1], and the gyroscope on GPIO[2].
- The Socket A sensor module, containing the magnetometer, has the magnetometer's interrupt on pin #32, which is tied to INT2.
- The Socket B sensor module, containing the accelerometer and gyroscope, has the accelerometer interrupt on pin #30, which is tied to INT1, and the gyroscope interrupt on pin #26, which is tied to INT3.
- As such, the jumpers connect SOCKET A's INT2 pin with the Sensor-Bus-side SOCKET E's GPIO[0] pin, thus connecting the magnetometer's interrupt line with the GPIO line where SENtral is expecting to see the magnetometer interrupt. Similarly, SOCKET B's INT1 and INT3 pins are connected with the Sensor-Bus-side SOCKET E's GPIO[1] and GPIO[2] pins, respectively, thus properly connecting the accelerometer and gyroscope.

4.2. Connectors

The SENtral Development Board provides two connectors for powering the board and for communicating with the STM32 microprocessor: a micro-USB connector, and a 6-pin right-angle SIP connector. The micro-USB allows for communication with the STM32 microprocessor via UART VCP or via USB, while the 6-pin SIP connector allows for communication via UART. Only one connector need be used.

4.2.1. Micro-USB Connector

The micro-USB connector should be connected to a computer's USB port via the micro-USB-to-USB cable. The micro-USB connector allows for communicating with the on-board STM32L151RD microprocessor using either UART VCP or USB protocols. Figure 4-4 shows how to configure the VCP/USB jumper connector to properly select the desired communication protocol.

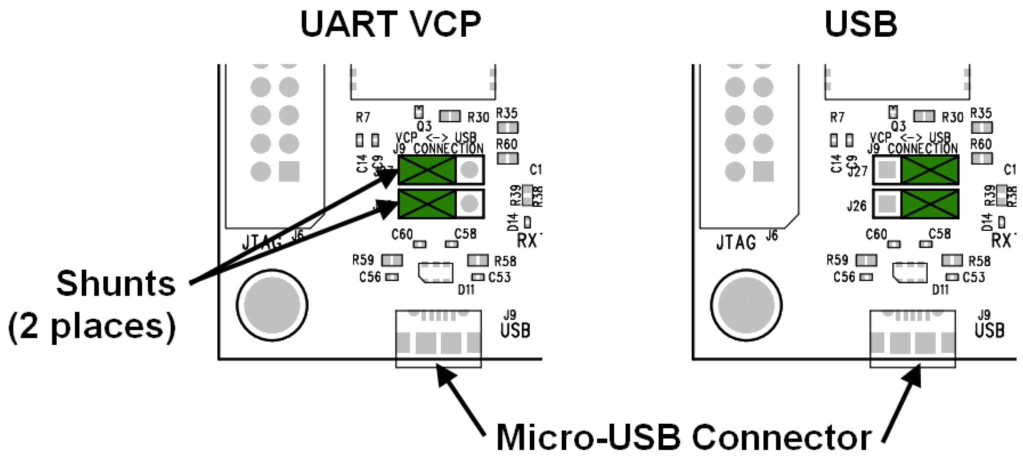


Figure 4-4: VCP/USB Shunt Configuration

UART VCP Configuration

For UART VCP communication, first ensure the VCP/USB shunt is in the "UART VCP" position, as shown in Figure 4-4.

The development board includes an FTDI FT232R IC for bridging the USB output of the PC with the UART#2 port on the STM microprocessor, and the driver for the FT232R needs to be installed on your PC. To confirm if the device driver is installed, connect the SENtral Development Board using a micro-USB-to-USB cable, then navigate to your PC's Windows Device Manager. Under "Ports" in the Device Manager you should see a device labeled "USB Serial Port" along with the port address. Right click on this and select "Properties". This will open a new window. Click the "Driver" tab and confirm "Driver Provider" is FTDI.

If an FTDI VCP driver is not installed, download "CDM 2.08.30 WHQL Certified.zip" at: http://www.ftdichip.com/Drivers/VCP.htm. Use a zip utility to "Extract" the contents, which include: "CDM **v**2.08.30 WHQL Certified.zip" (note the "v" prior to the revision number, which is not in the host zip file name), "CDM v2.08.30 WHQL Certified.exe", and "CDM 2 08 30 Release Info.rtf". Double-click on "CDM v2.08.30 WHQL Certified.exe" to install the VCP driver. This should complete the FTDI driver installation process. Plug in the SENtral Demo Module and check under "Ports" in the Device Manager. You should see "USB Serial Port" along with the port address. If not, reinstall the driver, and next reboot your computer.

USB Configuration

The SENtral Development Board's STM32L151RD microprocessor incorporates a USB interface so it can communicate directly with a PC via USB. For USB communication, ensure the VCP/USB shunt is in the "USB" position, as shown in Figure 4-4.

Prior to using the USB interface it is necessary for ST's DfuSe Demonstration Program to be installed on the PC. If running Windows 7, the PC's operating system automatically will locate and install the tool. If running a prior version of Windows it may be necessary to manually install the DfuSe Demonstration Program. In this case download "stsw-stm32080.zip" and the installation instructions from ST's website at:

http://www.st.com/web/en/catalog/tools/FM147/CL1794/SC961/SS1533/PF257916

Note that while the SENtral Development Board allows for direct USB communication with the STM32 microprocessor, SENtral firmware only supports UART and UART VCP communication. Still, direct USB communication can be useful for certain processes, such as bootloading firmware into the microprocessor.

4.2.2. 6-Pin SIP UART Connector

The 6-pin right-angle SIP connector primarily is intended to be used when communicating directly with a microprocessor. The pin-out and associated voltages are given in Table 4-2

Table 4-2: 6-PIN SIP Connector Pin-Out & Voltages

Label	Description	Voltages
RTS	Not implemented	
TXD	UART Transmit Line	High-Level Output, V_{OH}: >2.4 V
		Low-Level Output, V_{OL}: <0.4 V
RXD	UART Receive Line	High-Level Input, V_{IH}: 2.0 to 5.5 V
		Low-Level Input, V_{IL}: -0.3 to 0.8 V
+V	Supply Voltage	4.75 to 5.4 VDC
CTS	Not implemented	
GND	Ground	-0.3 to 0.0 VDC

4.3. Test Points

The SENtral Development Board incorporates 3x 12-pin test point connectors which can be used to monitor certain outputs of the STM32L151RD microprocessor, the SENtral Master Socket connector, other primary voltages, and ground. The test points are delineated in Table 4-3. "Attached to" references beginning with "SMS" refer to the SENtral Master

Socket (E), while references beginning STM refer to the STM32 microprocessor, and in both cases the number is the socket's or STMs's pin number.

Table 4-3: SENtral Development Board Test Points

Label	Attached to:*	Description
TP4	--	Not implemented
GND	Ground	Ground – same as TP1
SNS_NRST	SMS 24	Tied to pin 24 of all GEN7 sockets (A-E),
SCLS_LV[1]	SMS 34	Sentral Host Bus Slave I2C Clock.
SCLM	SMS 55	Sentral Sensor Bus Master I2C Clock
SCL	STM 61	STM Host Bus Master I2C Clock
TP5	--	Not implemented
AVDD	SMS 38	Sensor Module Sockets' AV_{DD} – same as TP2
VDD	SMS 23	Sensor Module Sockets' V_{DD} – same as TP3
SDAS_LV[1]	SMS 36	Sentral Host Bus Slave I2C Data Line
SDAM	SMS 57	Sentral Sensor Bus Master I2C Data Line
SDA	STM 62	STM Host Bus Master I2C Data Line
SD-CS	STM 38	Not implemented
MOSI	STM 23	STM SPI Master Out, Slave In
HCDRV1	STM 25	Not implemented
GPS_RX	STM 29	Not implemented
GPS_PPS	STM 33	Not implemented
BT_SSN	STM 20	Not implemented
SCK	STM 21	STM SPI Clock
MISO	STM 22	STM SPI Master In, Slave Out
HCDRV2	STM 37	Not implemented
GPS_TX	STM 30	Not implemented
GPS_RFSTAT	STM 34	Not implemented
GPS_ON/OFF	STM 35	Not implemented

BT_RSTN	STM 27	Not implemented
BATMEAS	STM 36	Not implemented
PD2	STM 54	STM32 PD2 GPIO
PA8	STM 41	STM32 PA8 GPIO
VBAT	--	Not implemented
GND	Ground	Ground – same as TP1
BT_DRDY	STM 26	Not implemented
BATMON	STM 15	Not implemented
MBSEL	STM 59	Not implemented, but should be configured as a Push/Pull output on the STM32 and set to LOW.
VIN	+V & VUSB	Input to voltage regulators for VDD and AVDD
+3.1V	STM 13, 19, etc.	Regulated AV_{DD} and V_{DD} for the microprocessor. Derived from USB. This supply source is NOT derived from VIN.
GND	Ground	Ground – same as TP1

Footnote

1. SENtral slave I^2C signal labels ending with "_LV" denote the Low Voltage side, or the SENtral VDD side, of the I^2C level translator that bridges the STM32 (3.1V) to the sensor module VDD sockets.

4.4. Proper Power Sequencing

Because the PNI SENtral Development Kit includes a battery power option, a power management IC is used on this board. This IC includes several power and battery protection schemes. The protection functions include battery over-charge/discharge, over/under charge, input supply under/over-voltage and 3.1V over current. Because the Dev Kit does not come standard with a battery, there are certain conditions that trigger the battery protection's fault circuitry. Once triggered, the 3.1V regulated output, included in this IC and used as the interface power on this board, will become disabled. Indication of this will be the absence of the Interface Power LED illuminating when the Interface Power switch is turned on. To prevent this from happening, observe the following power sequences:

1. Start with both the Interface Power Switch and the Sensor Power switch in the off position when plugging in or applying external power to the SENtral Development Board.
2. Do not turn the Sensor Power Switch to the position first. Turn it ON after or at the same time as the Interface Power Switch.
3. Never turn the Interface Power Switch to the OFF position with the Sensor Power switch in the ON position. Always turn the Sensor Power Switch to the OFF position first.

4. When unplugging or removing power from the SENtral Development Board wait several seconds with both the Power Switch and the Sensor Power Switch in the OFF position before reapplying power.

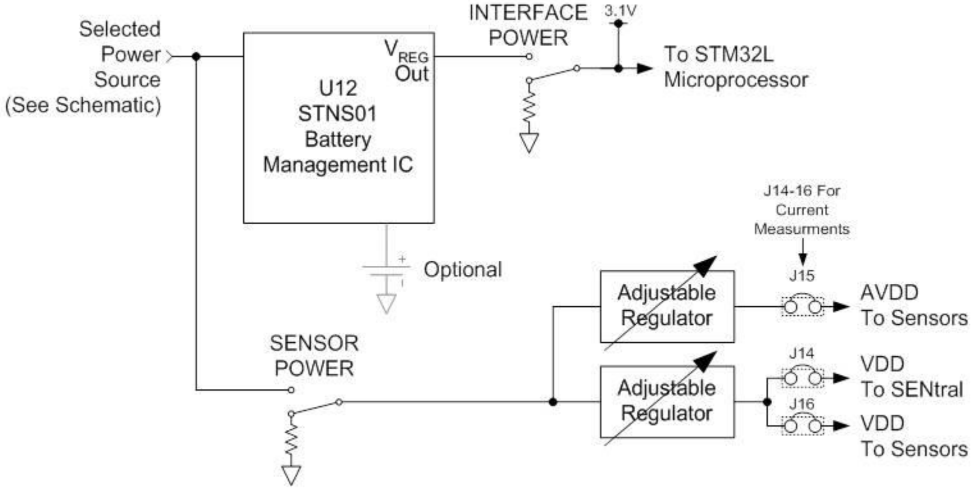


Figure 4-5: SENtral Development Kit Power Scheme

4.5. Uploading STM32 Firmware using the Bootloader

The STM32 microprocessor includes a bootload feature which enables uploading firmware to the microprocessor. It normally is more convenient to perform this operation using the micro-USB connector than using the 6-pin UART connector, but it is possible to upgrade using either connector.

For a detailed explanation of how to upload firmware to the STM32 microprocessor using the micro-USB, see Appendix II.

Bootloading via the 6-pin SIP UART connector is not as convenient as using the micro-USB connector, but is possible. To bootload using the 6-pin SIP UART connector, normally it is necessary to connect your PC to the UART connector with a USB-to-UART cable, such as FTDI's TTL-R232-3V3-WE. This specific cable requires the same UART VCP driver discussed in Section 4.2.1. Information on this cable can be found at http://www.ftdichip.com/Products/Cables/USBTTLSerial.htm. Next, download ST's Flash Loader (stsw-mcu005) and accompanying instructions onto your PC. These can be found at: http://www.st.com/web/en/catalog/tools/FM147/CL1794/SC961/SS1743/PF257525?s_searchtype=keyword

4.6. Setting DVDD & AVDD for Sensors and SENtral

The voltages across the sensor module sockets are established by the VDD (aka VDD_IO) and AVDD potentiometers. These are shown in Figure 4-6, with test leads attached to the VDD and GND test pins to monitor the VDD voltage. Turning the potentiometers clockwise increases the voltage. The potentiometers have a range of 1.4 V to 3.8 V, which likely exceeds the use-case ranges as established by the sensors and SENtral. For SENtral, VDD should be between 1.6 V and 3.3 V. A typical sensor AVDD range is 2.4V to 3.6V and a typical sensor VDD range is 1.71V to AVDD, but please refer to the sensor manufacturer's datasheet for you specific sensors.

CAUTION: The absolute maximum voltage rating for SENtral is 3.6 V. Setting VDD >3.6 V may cause permanent damage to the device.

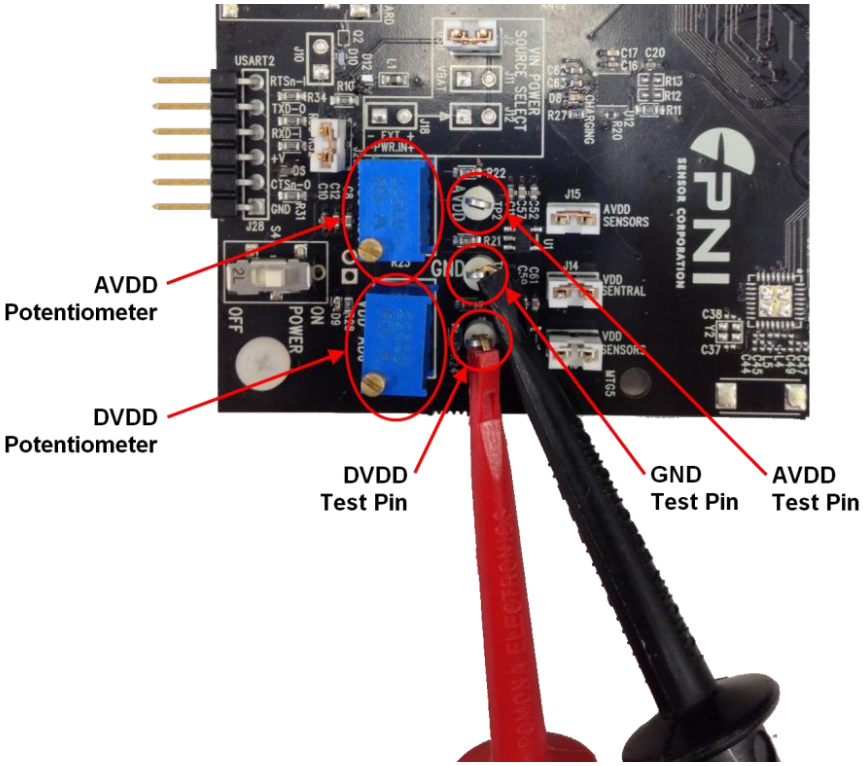


Figure 4-6: DVDD and AVDD Potentiometers and Test Points

4.7. Measuring SENtral & Sensor Current Draw

Current draw to the sensor module sockets can be monitored via the current test points located near the AVDD and VDD potentiometers. To measure current draw, remove the shunt for the current of interest and apply test leads across the jumper pins. The test point labeled “AVDD SENSORS” gives the combined AVDD current consumption for all the sensor module sockets, A through E. “VDD SENSORS” provides the combined VDD current consumption for sensor module sockets A through D, while “VDD SENTRAL” supplies the VDD current consumption for the SENtral Master Socket (E). Figure 4-7 shows the configuration for measuring the current through “VDD SENTRAL”.

Note: *To measure the current draw of only the SENtral coprocessor it is necessary to have the SENtral-only sensor module in socket (E) and the sensors in modules in other sockets. (i.e. a multiple sensor modules approach is necessary to measure the current draw of SENtral.)*

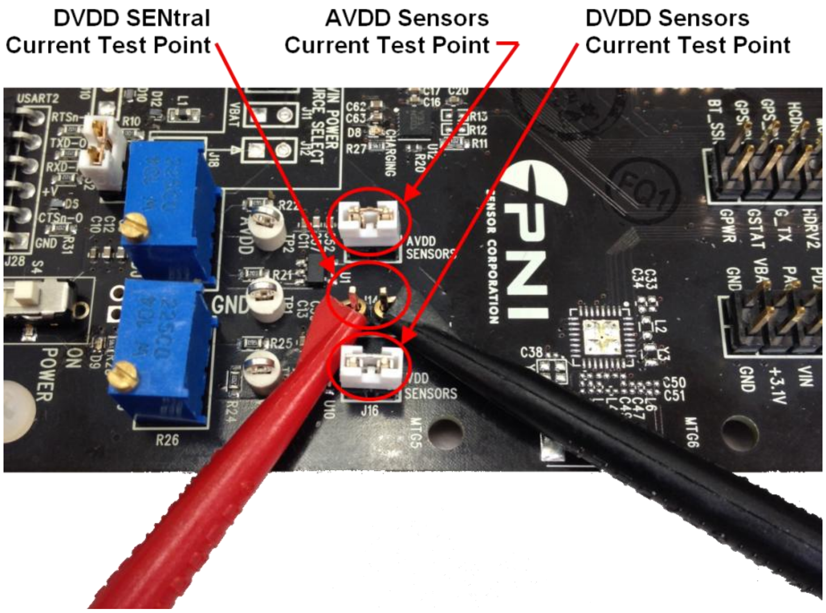


Figure 4-7: SENtral and Sensor Current Test Points

5. Loading the SENtral Configuration File

Note: *If you are using an all-in-one sensor module this section may be skipped since the Configuration File is preloaded on the module at PNI. Read this section if you want to load a different version of the Configuration File onto an all-in-one module or are you using the multiple sensor modules approach.*

Prior to operating the SENtral Development Board, it is necessary to upload the SENtral Configuration File into SENtral RAM. The Configuration File can be loaded into SENtral RAM in two manners: 1) automatically from an EEPROM containing the Configuration File, or 2) manually from the host which contains the Configuration File. Once an EEPROM is programmed, the Configuration File automatically uploads on power-up. If no EEPROM is programmed, the Configuration File must be manually uploaded each time the development board is powered-up.

The Configuration File is generated using the SENtral Configuration Tool, found at www.sentraltoolkit.com, this file specifies which sensor models are used, their slave addresses and interrupt pins, the sensor orientation matrix, etc.. Both the Configuration File and Configuration Tool are discussed further in the SENtral Technical Datasheet.

The following sections discuss how to program an EEPROM with the Configuration File and how to manually load the Configuration File from the host.

5.1. Using the SENtral Utility Tools

To program an EEPROM for automatic upload of the Configuration File into SENtral RAM, the following files should be in the same directory on your PC. These files are supplied by PNI, except the SENtral Configuration File which is generated using the Configuration Tool.

- SENtral Configuration File, ending in .fw
- i2cCom.exe
- EEPROMprogrammer.exe
- RunConfigFileLoader.bat

Additionally, it is necessary that “Microsoft Visual C++ Redistributable” is installed on your PC. This can be found at http://www.microsoft.com/downloads/en/default.aspx.

5.2. Programming an EEPROM with a Configuration File

The development board can accommodate two EEPROMs, 0xA8 is already installed. Table 5-1 below details the EEPROM address for the Dev Kit hardware options.

Table 5-1: Dev Kit Hardware EEPROM Addresses

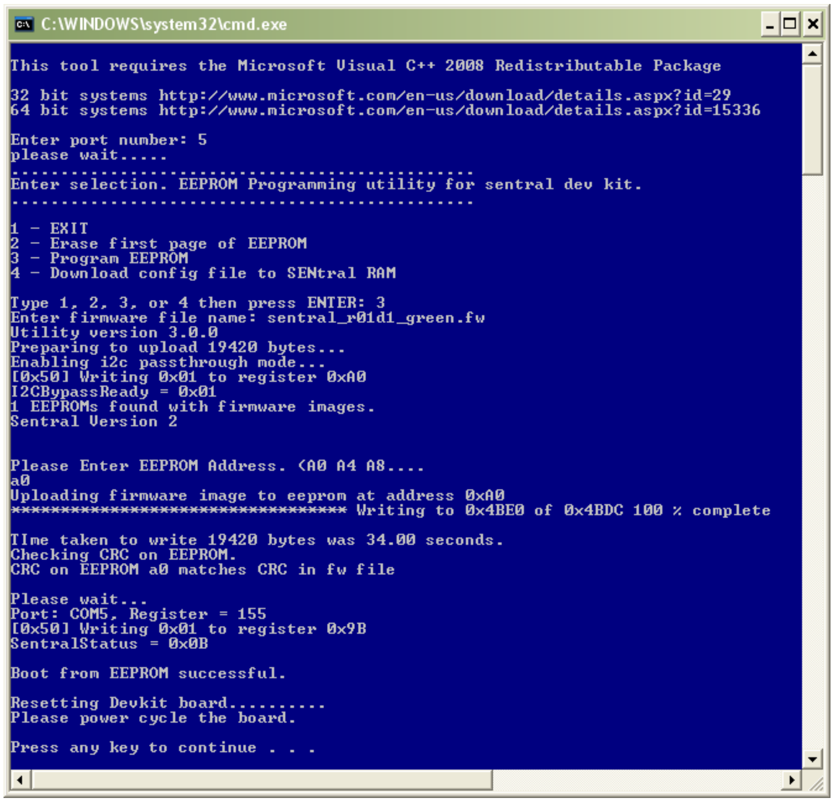
<table>
<tr><th rowspan="2">Device</th><th colspan="6">Address</th></tr>
<tr><th>0xA0</th><th>0xA2</th><th>0xA4</th><th>0xA6</th><th>0xA8</th><th>0xAA</th></tr>
<tr><td>DevBoard</td><td colspan="2">(U7 not installed)</td><td></td><td></td><td colspan="2">●</td></tr>
<tr><td>GS Modules</td><td></td><td></td><td colspan="2">●</td><td></td><td></td></tr>
<tr><td>M&M Modules</td><td colspan="2">●</td><td></td><td></td><td colspan="2">(U5 not installed)</td></tr>
<tr><td colspan="7">PNI uses 1 Mbit (128K-Byte) EEPROMs that span 2 addresses as shown above, however, the data is accessible sequentially in the lower address space.</td></tr>
</table>

SENtral automatically uploads the Configuration File found at the lowest address EEPROM. If A0 is programmed, this will be used. If A0 is not programmed, SENtral will look to A4, and next to A8. **It is important to know which EEPROMs are programmed to ensure the desired Configuration File is uploaded.** See Section 5.3 for how to erase an EEPPROM.

To program an EEPROM, follow the steps below:

- Ensure the VCP/USB shunts are set for USB operation, per Figure 4-4, and ensure the Bootloader Option switches are in the LOW position.
- Connect the development board to a PC with the micro-USB-to-USB cable.
- Open RunConfigFileLoader.bat. This launches the Windows command-line screen.
- Enter the port number for the development board. The port number should be ≤9. If it is ≥10, or you do not know the port number, see Appendix I.
- Enter the number 3 to "Program EEPROM".
- Enter the SENtral Configuration File name.
- Enter the desired EEPROM address. The tool will now write the Configuration File into the EEPROM.
- After the firmware uploads and indicates "Boot from EEPROM successful", press any key and press the STM32 Reset Switch to power cycle the board.

A sample command-line screen for programming an EEPROM is shown below.



```
This tool requires the Microsoft Visual C++ 2008 Redistributable Package

32 bit systems http://www.microsoft.com/en-us/download/details.aspx?id=29
64 bit systems http://www.microsoft.com/en-us/download/details.aspx?id=15336

Enter port number: 5
please wait.....
-----------------------------------------------
Enter selection. EEPROM Programming utility for sentral dev kit.
-----------------------------------------------

1 - EXIT
2 - Erase first page of EEPROM
3 - Program EEPROM
4 - Download config file to SENtral RAM

Type 1, 2, 3, or 4 then press ENTER: 3
Enter firmware file name: sentral_r01d1_green.fw
Utility version 3.0.0
Preparing to upload 19420 bytes...
Enabling i2c passthrough mode...
[0x50] Writing 0x01 to register 0xA0
I2CBypassReady = 0x01
1 EEPROMs found with firmware images.
Sentral Version 2


Please Enter EEPROM Address. <A0 A4 A8....
a0
Uploading firmware image to eeprom at address 0xA0
********************************** Writing to 0x4BE0 of 0x4BDC 100 % complete

TIme taken to write 19420 bytes was 34.00 seconds.
Checking CRC on EEPROM.
CRC on EEPROM a0 matches CRC in fw file

Please wait...
Port: COM5, Register = 155
[0x50] Writing 0x01 to register 0x9B
SentralStatus = 0x0B

Boot from EEPROM successful.

Resetting Devkit board..........
Please power cycle the board.

Press any key to continue . . .
```

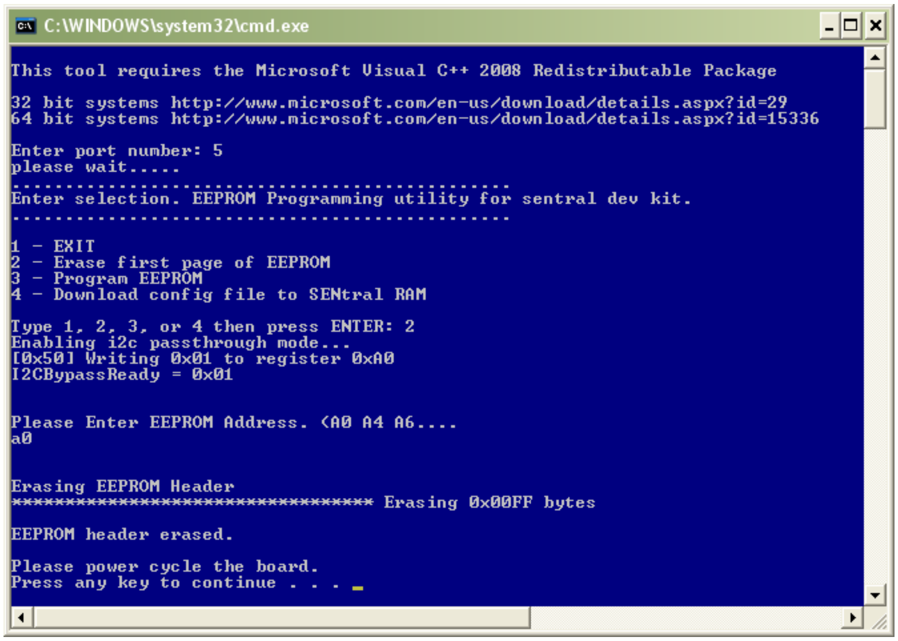
5.3. Erasing an EEPROM

As mentioned in Section 5.2, it is important to know which EEPROMs are programmed to ensure the desired Configuration File is uploaded into SENtral RAM. PNI recommends erasing any EEPROM memory locations that you are not intentionally using. Specifically, erase 0xA0 if you have not intentionally programmed it.

To erase an EEPROM, follow the steps below:

- Ensure the VCP/USB shunts are set for USB operation, per Figure 4-4, and ensure the Bootloader Option switches are in the LOW position.
- Connect the development board to a PC with the micro-USB-to-USB cable.
- Open RunConfigFileLoader.bat. This launches the Windows command-line screen.
- Enter the port number for the development board. The port number should be ≤9. If it is ≥10, or you do not know the port number, see Appendix I.
- Enter the number 2 to "Erase first page of EEPROM".
- Enter the EEPROM address to be erased.
- The tool will now erase the first page of the EEPROM.
- After the firmware indicates "EEPROM header erased", press any key and press the STM32 Reset Switch to power cycle the board.

A sample command-line screen for erasing an EEPROM is shown below.



```
This tool requires the Microsoft Visual C++ 2008 Redistributable Package

32 bit systems http://www.microsoft.com/en-us/download/details.aspx?id=29
64 bit systems http://www.microsoft.com/en-us/download/details.aspx?id=15336

Enter port number: 5
please wait.....
...............................................
Enter selection. EEPROM Programming utility for sentral dev kit.
...............................................

1 - EXIT
2 - Erase first page of EEPROM
3 - Program EEPROM
4 - Download config file to SENtral RAM

Type 1, 2, 3, or 4 then press ENTER: 2
Enabling i2c passthrough mode...
[0x50] Writing 0x01 to register 0xA0
I2CBypassReady = 0x01


Please Enter EEPROM Address. (A0 A4 A6....
a0


Erasing EEPROM Header
*********************************** Erasing 0x00FF bytes

EEPROM header erased.

Please power cycle the board.
Press any key to continue . . . _
```

5.4. Manual Upload of a Configuration File

To manually upload the Configuration File into SENtral's RAM, the following files should be in the same directory on your PC. These files are supplied by PNI, except the SENtral Configuration File which is generated by the user using the SENtral Configuration Tool. **Note that manually uploading the Configuration File will overwrite any previously loaded Configuration File.**

- SENtral Configuration File, ending in .fw
- i2cCom.exe
- RunConfigFileLoader.bat

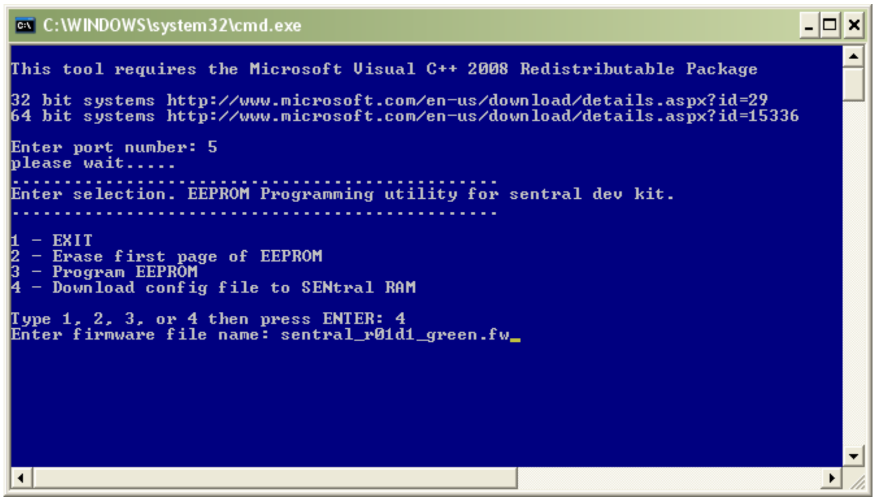
Additionally, it is necessary that "Microsoft Visual C++ Redistributable" is installed on your PC. This can be found at http://www.microsoft.com/downloads/en/default.aspx.

Note: *The Configuration File is stored in SENtral's volatile RAM. If it is not stored on an EEPROM, then it needs to be manually uploaded whenever the SENtral Development Board is powered up.*

To upload a Configuration File into SENtral RAM, follow the steps below:

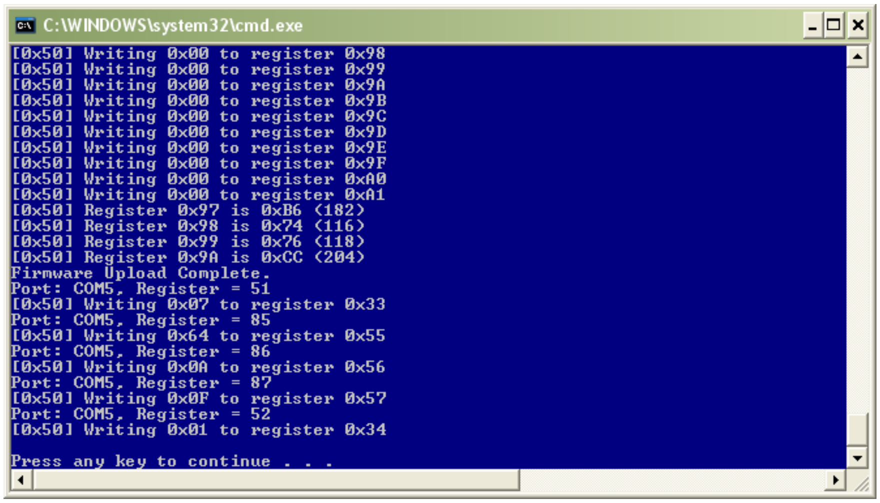
- Ensure the VCP/USB shunts are set for VCP, per Figure 4-4, and the Bootloader Option switches are in the LOW position.
- Connect the SENtral Development Board to a PC with a micro-USB-to-USB cable.
- Open RunConfigFileLoader.bat. This will launch a Windows command-line screen.
- Enter the port number for the Development board and press Enter. The port number should be ≤9. If it is not or you do not know the port number, see Appendix I.

- Enter the number 4 to “Download Config File to SENtral RAM”.
- Enter the SENtral Configuration File name. The screen should similar to that shown below. Press “Enter” to continue.



```
This tool requires the Microsoft Visual C++ 2008 Redistributable Package

32 bit systems http://www.microsoft.com/en-us/download/details.aspx?id=29
64 bit systems http://www.microsoft.com/en-us/download/details.aspx?id=15336

Enter port number: 5
please wait.....
-----------------------------------------------
Enter selection. EEPROM Programming utility for sentral dev kit.
-----------------------------------------------

1 - EXIT
2 - Erase first page of EEPROM
3 - Program EEPROM
4 - Download config file to SENtral RAM

Type 1, 2, 3, or 4 then press ENTER: 4
Enter firmware file name: sentral_r01d1_green.fw_
```

- The tool will now write the Configuration file directly into SENtral’s RAM.



```
[0x50] Writing 0x00 to register 0x98
[0x50] Writing 0x00 to register 0x99
[0x50] Writing 0x00 to register 0x9A
[0x50] Writing 0x00 to register 0x9B
[0x50] Writing 0x00 to register 0x9C
[0x50] Writing 0x00 to register 0x9D
[0x50] Writing 0x00 to register 0x9E
[0x50] Writing 0x00 to register 0x9F
[0x50] Writing 0x00 to register 0xA0
[0x50] Writing 0x00 to register 0xA1
[0x50] Register 0x97 is 0xB6 (182)
[0x50] Register 0x98 is 0x74 (116)
[0x50] Register 0x99 is 0x76 (118)
[0x50] Register 0x9A is 0xCC (204)
Firmware Upload Complete.
Port: COM5, Register = 51
[0x50] Writing 0x07 to register 0x33
Port: COM5, Register = 85
[0x50] Writing 0x64 to register 0x55
Port: COM5, Register = 86
[0x50] Writing 0x0A to register 0x56
Port: COM5, Register = 87
[0x50] Writing 0x0F to register 0x57
Port: COM5, Register = 52
[0x50] Writing 0x01 to register 0x34

Press any key to continue . . .
```

- After the firmware uploads and indicates “Firmware Upload Complete”, as shown above, press any key to complete the process. If you do not see “Firmware Upload Complete”, but rather see a CRC error, then power cycle the Development board by unplugging it and reconnecting it, and then retry the process

6. Operation with the SENtral Demo Program

This section reviews how to operate the SENtral Development Board using the SENtral Demo Program. The purpose of the SENtral Demo Program is to easily demonstrate and test the SENtral Development Board.

6.1. Demo Program Set-Up

Note: *The SENtral Demo Program is built on a Unity platform and has been demonstrated to run on Windows XP, Windows Vista, and Windows 7 computers. However, no guarantees are made regarding the ability of the SENtral Demo Program to operate on any or all computer configurations.*

Before starting:

Ensure your PC and the SENtral Development Board are configured for UART VCP communication.

See Section 4.2.1 for a discussion on configuring the VCP/USB jumper, and installing the FTDI driver.

Install the SENtral Demo Program on your computer.

"Extract" all the files from the "SENtralDemo_x.zip" file, where "x" is the revision level. This will put "SENtralDemo_x.exe" and the folder "SENtralDemo_x_Data" into the same directory as the zip file. These two items must remain located within the same directory on your computer's hard drive.

6.2. Running the SENtral Demo Program

To run the program:

Help ensure the SENtral Demo Program runs smoothly.

Close all other applications and ensure you are running on a single-monitor computer system.

Plug the SENtral Development Board into your computer's USB port.

If this is the first time plugging in a specific SENtral Development Board, Windows automatically will launch the "Found New Hardware" wizard, then indicate it has found the module and installed the device. After this, go to the Windows Device Manager to confirm which port is assigned to the SENtral Development Board by looking under "Ports", where you should see a device labeled "USB Serial Port" along with the port address.

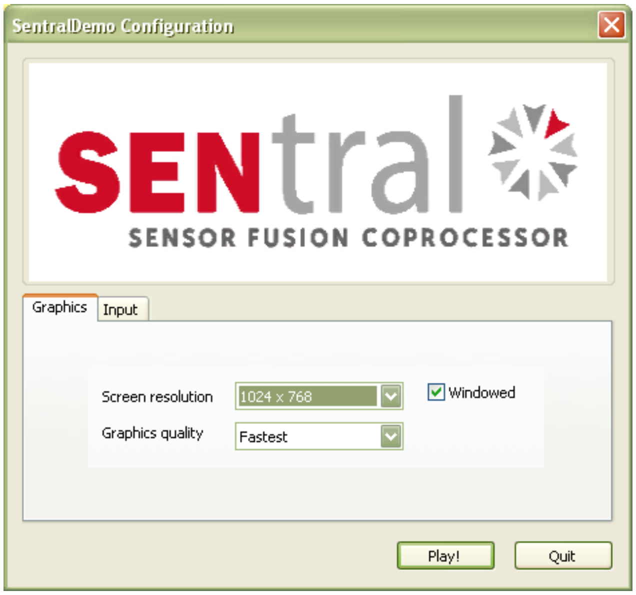
Upload the Configuration File.

If you are using an all-in-one sensor module or if you have loaded one of the EEPROMs with the Configuration File, then the Configuration File will upload automatically and you should proceed to the next step.

If this is not true, then upload the Configuration File as described in Section 5.3. In this case the SENtral Configuration File needs to be uploaded whenever the SENtral Development Board has been powered up, as the file is stored in volatile memory.

Launch and Configure the SENtral Demo Program.

Double-click on the "SENtralDemo_x.exe" icon to launch the program, where "x" indicates the revision level. Ensure the SENtral Development Board is fully at rest for at least 3 seconds immediately after launching the program to allow the gyros time to initialize. The Configuration window will appear, as shown below. Set the desired screen resolution and graphics quality. Graphics quality options are established by the Unity engine and will depend on your computer's configuration. PNI generally recommends "Fastest" graphics quality, but the user can experiment with other settings. Deselecting the "Windowed" box will cause the program run full-screen. Click the "Play!" button or press the <Enter> key.

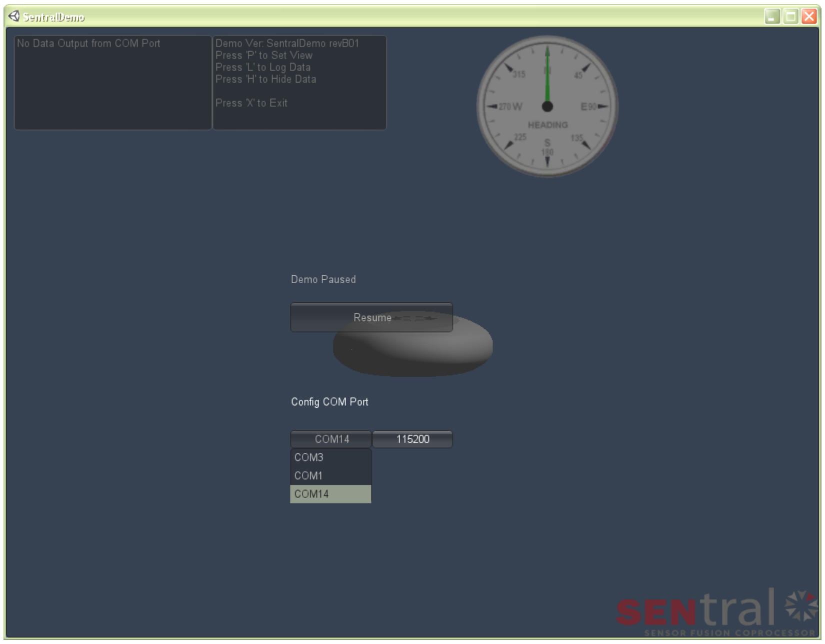


Screen resolution 1024 x 768 ☑ Windowed

Graphics quality Fastest

Initiate Running of the Program.

The main program window will now open, as shown below. Select the correct port number for the SENtral, and set the baud rate to 115200. Click the <Resume> button.



SentralDemo

```
No Data Output from COM Port
```

```
Demo Ver: SentralDemo revB01
Press 'P' to Set View
Press 'L' to Log Data
Press 'H' to Hide Data

Press 'X' to Exit
```

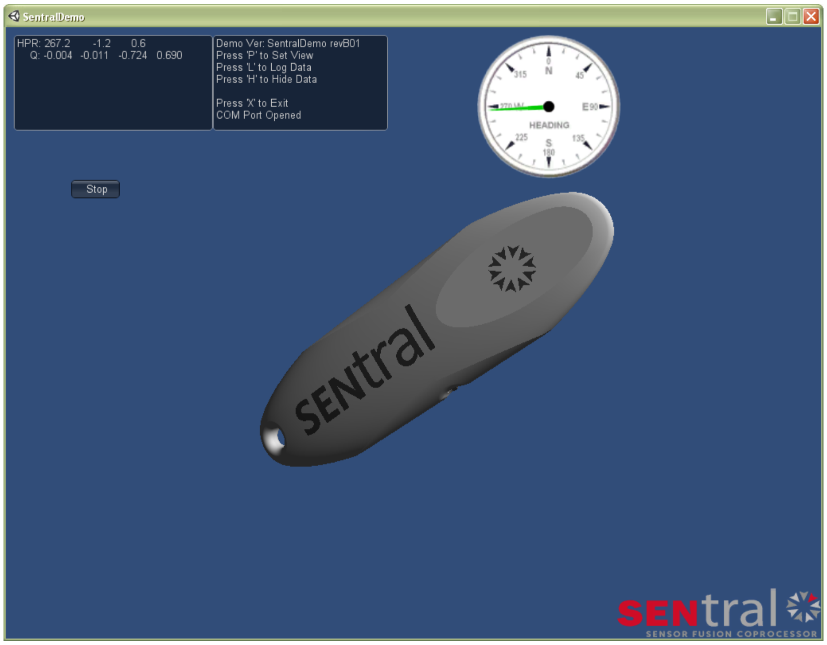
Demo Paused

Config COM Port

COM14 115200

COM3
COM1
COM14

A <Start> button will appear in the upper left quadrant. Click this to send data and initiate the program. Once the program is running: 1) the whole screen will brighten, 2) the values in the upper left-most box (the data text box) are no longer "0" and will change when the module is moved, 3) the compass dial will start tracking heading.



SentralDemo

```
HPR: 267.2      -1.2      0.6
   Q: -0.004  -0.011   -0.724   0.690
```

```
Demo Ver: SentralDemo revB01
Press 'P' to Set View
Press 'L' to Log Data
Press 'H' to Hide Data

Press 'X' to Exit
COM Port Opened
```

The data text box displays heading, pitch, and roll, in degrees, on the top line, and on the second line the quaternion values, Qx, Qy, Qz, and Qw. If desired, you may hide the data and command text boxes by pressing "h" on your keyboard. Pressing "h" again will toggle them back.

The compass dial provides real-time heading data in a common graphical format. The heading given on the compass dial matches the heading value in the data window.

Point the SENtral Development Board at the screen and press "P" for the rendered SENtral module image to track the actual SENtral Development Board's motion. Pressing the "P" key not only initiates motion-tracking, but also (re)initializes the orientation of the rendered object on the screen, such that when the key is pushed the rendered image will point straight into the screen with no pitch or roll.

6.3. Logging Data

To initiate logging of the orientation data, press the 'l' key (lower-case 'L', not the number '1'). The command text box will now say "LoggingData, Press 'L' to Stop". When you are done, press 'l' again to stop logging data. The data will be saved in a file in the same directory as the program. The file name will be "logMM-DD-YY_HH_MM_SS.csv" where MM-DD-YY indicates the month, day, and year, while HH_MM_SS indicates the hour, minute and second. This .csv file can be opened using Excel or other programs that recognize a text format.

6.4. Exiting the Program

Prior to exiting the program, click the <Stop> button to stop the SENtral algorithm. Then press the "x" key to exit the program. When you do this, a file titled "state.csv" will be saved in the same directory as the program. This file contains information that helps the SENtral quickly initialize itself when you restart the program. This file is not required for the SENtral Demo Program to work.

Table 6-1: Summary of SENtral Demo Program Commands

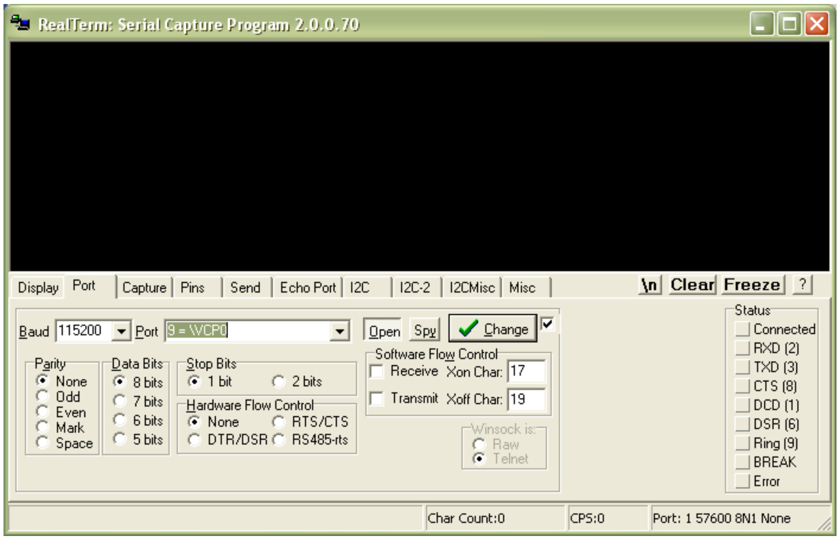
Key	Function
p	Sets rendered object orientation to point directly into screen.
h	Hides the data and command text boxes
l	Initiates logging of data and stops logging of data
x	Exits Program

7. Communicating via RealTerm

This section reviews communication with the SENtral Development Board via the RealTerm terminal emulation program, which is available for no charge at: http://realterm.sourceforge.net/. Other terminal emulation programs should work similarly. The I²C interface operates in hexadecimal, except where noted.

Establish Communication with the SENtral Development Board

Open RealTerm and click on the "Port" tab. On this tab, set "Baud" to 115200 and set "Port" to whichever virtual comm. port is assigned to the SENtral Development Board. Click <Change> to implement these changes.



RealTerm: Serial Capture Program 2.0.0.70

Baud 115200 Port 9 = \VCP0

Parity: ● None ○ Odd ○ Even ○ Mark ○ Space

Data Bits: ● 8 bits ○ 7 bits ○ 6 bits ○ 5 bits

Stop Bits: ● 1 bit ○ 2 bits

Hardware Flow Control: ● None ○ RTS/CTS ○ DTR/DSR ○ RS485-rts

Software Flow Control: □ Receive Xon Char: 17 □ Transmit Xoff Char: 19

Winsock is: ○ Raw ● Telnet

Char Count:0 CPS:0 Port: 1 57600 8N1 None

The port number should be <10. If it is not or you do not know the port number, see Appendix I.

Place RealTerm in Verbose Terminal Mode (optional)

If it is desirable to operate in verbose terminal mode, "T" turns on verbose terminal mode. "t" turns off terminal mode.

Read or Write Command Packets

The syntax of a Read command packet is: `{SLA REG NUM}`

The syntax of a Write command packet is: `[SLA REG DATA DATA….DATA]`

Where:

- **{ }** or **[ ]** initiate and terminate a command packet, where curly brackets are for Read commands and square brackets are for Write commands.

- **SLA** is the 7-bit slave address, plus the r/w bit in the LSB. For the SENtral Development Board, SLA = 0x50. (The SENtral Development Board's 7-bit slave address is 0x28 and the value of the r/w bit does not matter as the development board ensures the r/w bit is properly set. Shifting the 7-bit slave address over one bit yields SLA = 0x50.)
- **<SP>** is the delimiter between bytes. A space character should be inserted between SLA, REG, NUM, and each DATA.
- **REG** is the first register address to be read from or written to. If NUM>01, SENtral will automatically increment to the next register until NUM registers have been read from or written to.
- **NUM** is the number of data bytes to read, in decimal. This can be written as a single character, such as '7' instead of '07'.
- **DATA** are the data bytes to write. These can be written as single characters, such as 'A' instead of '0A'. Recall that a <SP> delimiter should be between DATA values. Up to 62 bytes can be written in one packet.

Examples:

`{50 00 10}` - Requests and returns 16 bytes of data starting at register 0x00. Specifically, this will return SENtral's current 4 quaternion values, Qx, Qy, Qz, and Qw, each of which is 4 bytes.

`{50 35 01}` - Requests and returns the 1 byte of data held at register 0x35. Specifically, this will return the contents of SENtral's EventStatus register.

`[50 55 1e a 13]` - Writes the value 0x1E to register 0x55, value 0x0A to register 0x56, and value 0x13 to register 0x56. Specifically, this sets the MagRate, AccelRate, and GyroRate registers to 30, 10, and 19, respectively, so that MagRate = 30 Hz, AccelRate = 100 Hz, and GyroRate = 190 Hz. Note that the leading '0' is optional for the data values, and the hex value 'A' can be upper or lower case.

`[50 34 1]` - Writes the value 0x01 to register 0x34. Specifically, this sends a RunRequest to SENtral's HostControl register.

Microprocessor Communication Error

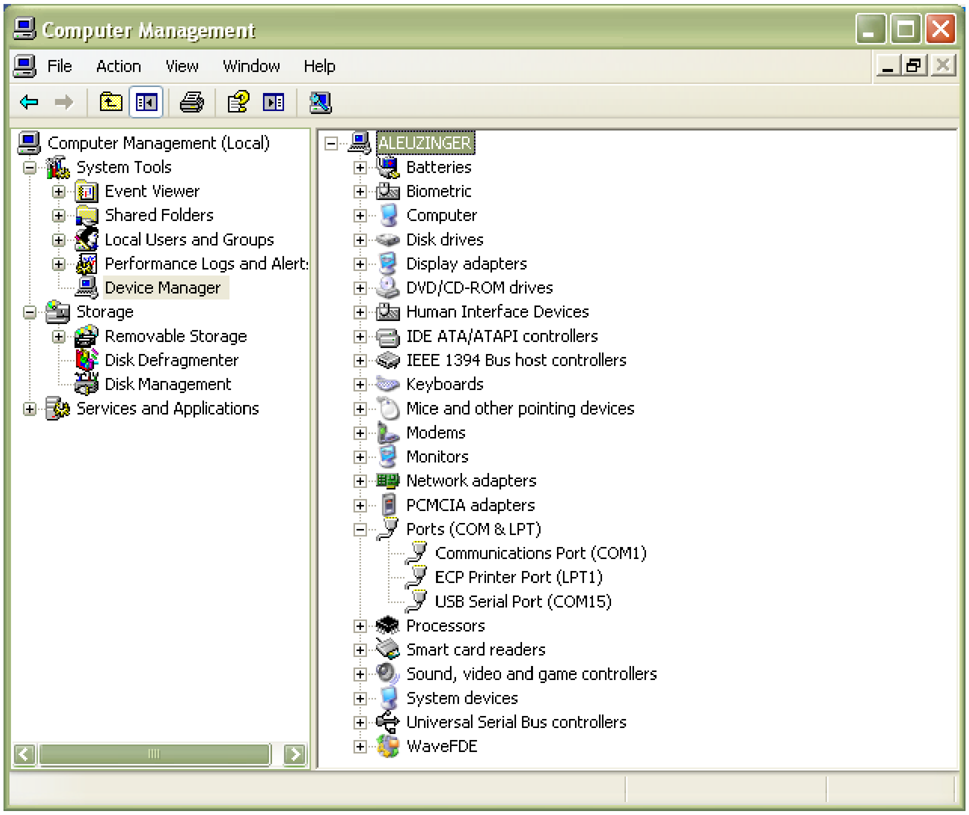
If the microprocessor cannot communicate properly with SENtral or the sensors in Pass-Through Mode, it will send '!'. Normally this can be cleared by rebooting the SENtral Development Board.

Appendix I – Reading & Setting the COM Port

When using SENtral Development Board, it is often important to know the COM port assigned to the board and, if the COM port is >9, it may be necessary to change the assigned COM port. The following describes how to determine the assigned COM port number and how to change it.

Reading the COM Port Number

Navigate to the Windows Device Manager. There are several ways to do this, but one is to locate "My Computer" on your PC, then right-click on this and select "Manage". In the left pane you will see "Device Manager". Select this, which will populate the right screen as shown below.

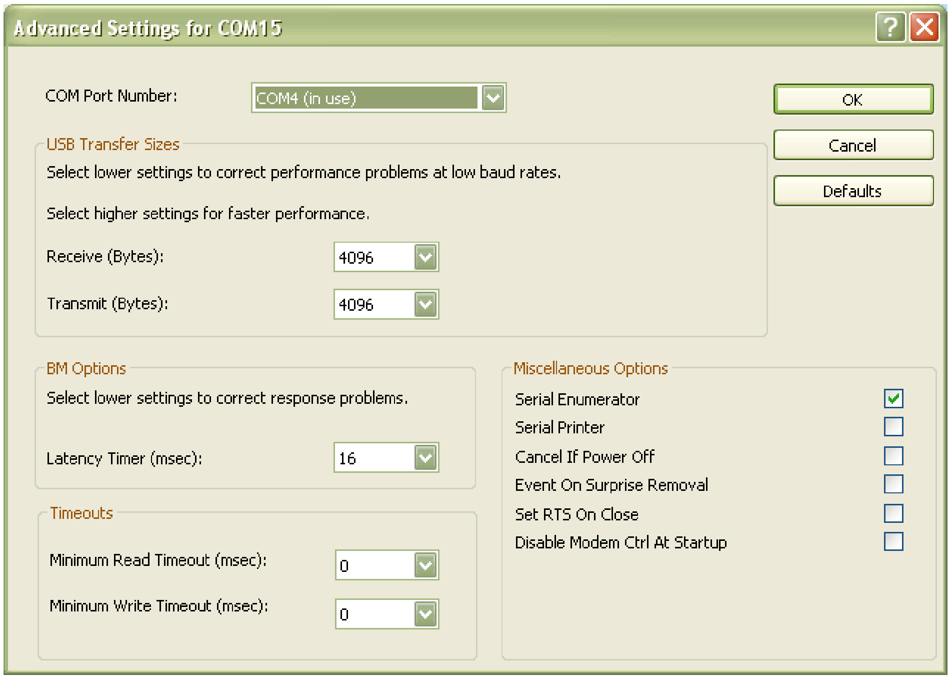


If "Ports" does not show the various assigned ports, click on the "+" to expand this. The SENtral Development Board will be assigned a port that reads "USB Serial Port". In the case above, the SENtral Development Board is assigned to COM port 15.

Setting the COM Port Number

If the assigned COM port is <10 it will be necessary to change the port assignment when working with RealTerm or the EEPROM programming tool. Navigate to the Device Manager, as discussed above, and identify the port assigned to the SENtral Development

Board. Right click on the port assigned to the SENtral Development Board and select “Properties”. Select the “Port Settings” tab, then click on the <Advanced> button. This will show the current “COM Port Number”. Change this to a number <10 that does not conflict with any other current devices. The screen shot below shows the reassignment of the SENtral Development Board from COM15 to COM4. In this case the “(in use)” designation is not relevant since nothing is actually assigned to the port at this time. But Windows remembers that something has been assigned in the past and this is why it indicates “in use”.


Advanced Settings for COM15

COM Port Number: COM4 (in use)

USB Transfer Sizes

Select lower settings to correct performance problems at low baud rates.

Select higher settings for faster performance.

Receive (Bytes): 4096

Transmit (Bytes): 4096

BM Options

Select lower settings to correct response problems.

Latency Timer (msec): 16

Timeouts

Minimum Read Timeout (msec): 0

Minimum Write Timeout (msec): 0

Miscellaneous Options

Serial Enumerator ☑

Serial Printer □

Cancel If Power Off □

Event On Surprise Removal □

Set RTS On Close □

Disable Modem Ctrl At Startup □

Click <OK> and <OK> again to implement the port number change.

Appendix II – Bootloading new STM32 Firmware

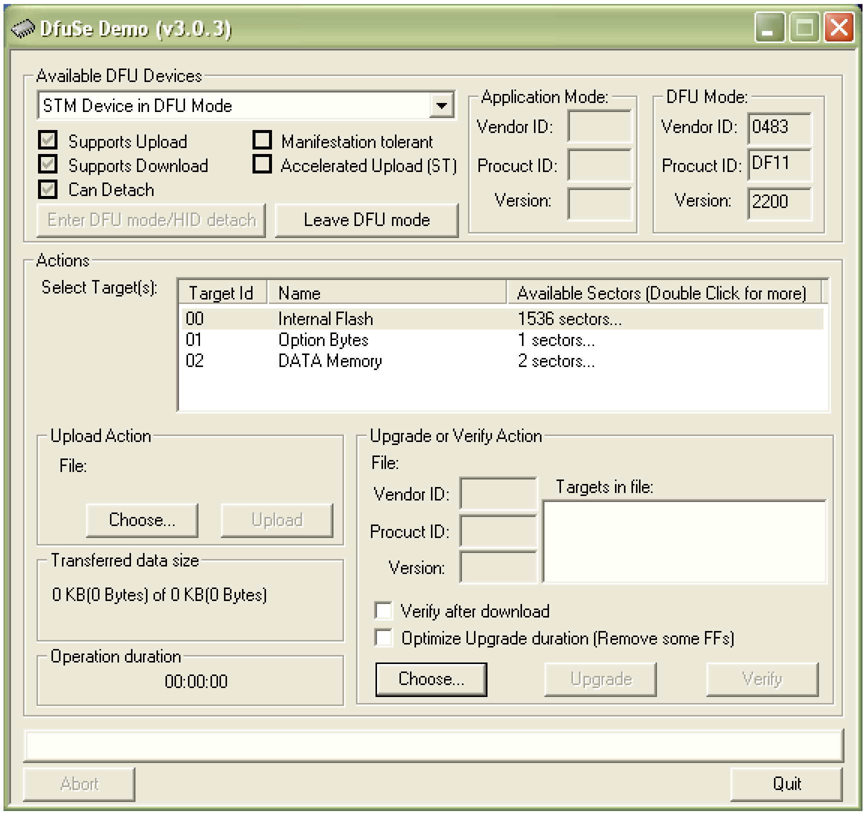
To bootload new firmware into the STM32 microprocessor PNI generally recommends using the micro-USB connector as this is the simplest approach. Before proceeding, ensure ST's USB DfuSe Demonstration Program is installed on your PC, as discussed in Section 4.2.1.

With the SENtral Development Board unplugged:

- Ensure the VCP/USB jumper is set to enable USB communication, as shown in Figure 4-4,
- Set the Bootloader Option Switches. Boot 0 should be HIGH and Boot 1 should be LOW.

Now plug the micro-USB-to-USB cable into the development board and the PC.

- Locate and open ST's "DfuSE Demonstration" program, which can be identified as given or as "DfuSeDemo.exe". Once opened, it should appear as follows:



DfuSe Demo (v3.0.3)

Available DFU Devices

STM Device in DFU Mode

☑ Supports Upload
☑ Supports Download
☑ Can Detach
☐ Manifestation tolerant
☐ Accelerated Upload (ST)

Application Mode:
Vendor ID:
Procuct ID:
Version:

DFU Mode:
Vendor ID: 0483
Procuct ID: DF11
Version: 2200

Actions

Select Target(s):

Target Id	Name	Available Sectors (Double Click for more)
00	Internal Flash	1536 sectors...
01	Option Bytes	1 sectors...
02	DATA Memory	2 sectors...

Upload Action
File:

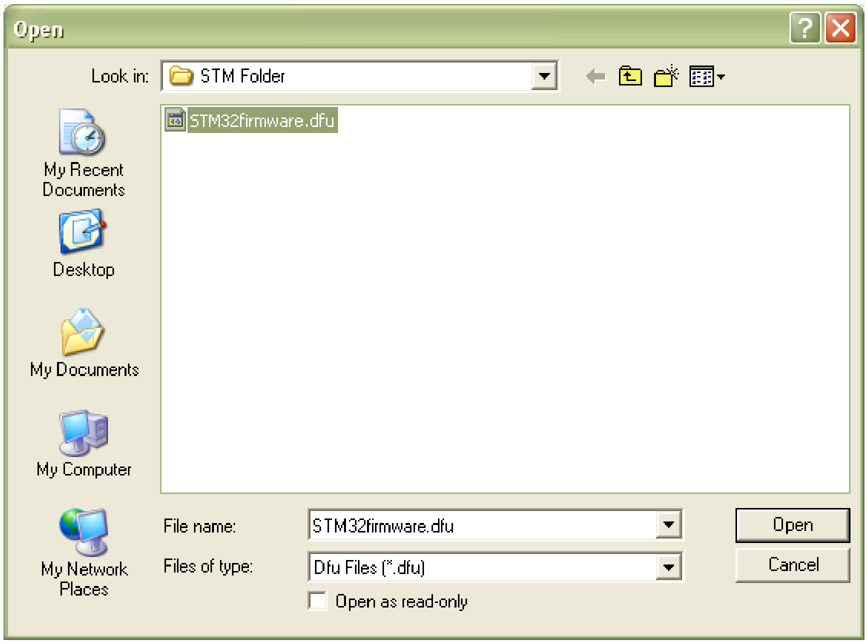
Transferred data size
0 KB(0 Bytes) of 0 KB(0 Bytes)

Operation duration
00:00:00

Upgrade or Verify Action
File:
Vendor ID:
Procuct ID:
Version:
Targets in file:

☐ Verify after download
☐ Optimize Upgrade duration (Remove some FFs)

- Click the <Choose> button in the center bottom of the screen, within the "Upgrade or Verify Action" box. Note there is another <Choose> button towards the left middle of the screen in the "Upload Action" box, which should not be selected.
- Locate and select the desired file to upload. This file should have a .dfu extension.



Look in: STM Folder

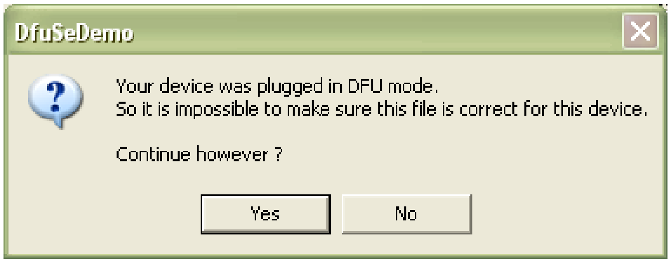
STM32firmware.dfu

File name: STM32firmware.dfu

Files of type: Dfu Files (*.dfu)

□ Open as read-only

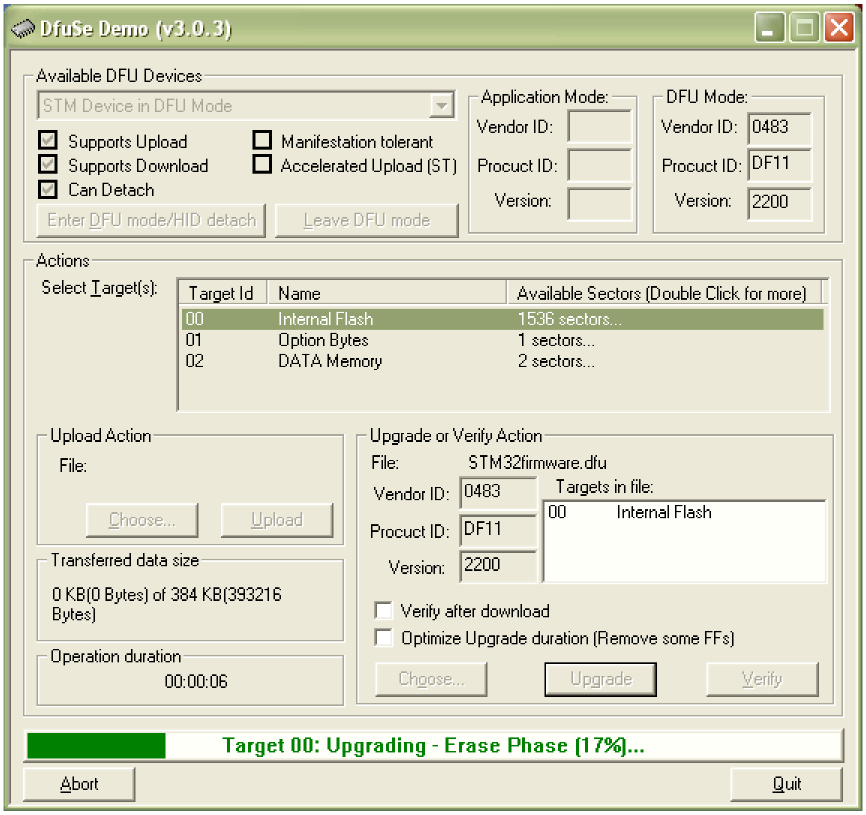
- Click <Open>, which should bring you back to the prior screen, except the bottom of the screen should say "File Correctly Loaded."
- Click <Upgrade>. This may launch a pop-up window indicating "it is impossible to make sure this file is correct for this device." Normally it is okay to click <Yes> to continue.



Your device was plugged in DFU mode.
So it is impossible to make sure this file is correct for this device.

Continue however ?

- The file will now upload to the STM32



Available DFU Devices

STM Device in DFU Mode

☑ Supports Upload
☑ Supports Download
☑ Can Detach
□ Manifestation tolerant
□ Accelerated Upload (ST)

Application Mode:
Vendor ID:
Procuct ID:
Version:

DFU Mode:
Vendor ID: 0483
Procuct ID: DF11
Version: 2200

Actions

Select Target(s):

Target Id	Name	Available Sectors (Double Click for more)
00	Internal Flash	1536 sectors...
01	Option Bytes	1 sectors...
02	DATA Memory	2 sectors...

Upload Action
File:

Transferred data size
0 KB(0 Bytes) of 384 KB(393216 Bytes)

Operation duration
00:00:06

Upgrade or Verify Action
File: STM32firmware.dfu
Vendor ID: 0483
Procuct ID: DF11
Version: 2200
Targets in file:
00 Internal Flash

□ Verify after download
□ Optimize Upgrade duration (Remove some FFs)

Target 00: Upgrading - Erase Phase (17%)...

- When complete, the screen will indicate "Upgrade successful!" Click <Quit>.
- Move Boot1 back to LOW and press the STM32 Reset Switch on the development board.