InvenSense MotionFit™ SDK Quick Start Guide
Release 1.2

InvenSense Confidential - Pre Release version

A printed copy of this document is
NOT UNDER REVISION CONTROL
unless it is dated and stamped in red ink as,
"REVISION CONTROLLED COPY."

This information furnished by InvenSense is believed to be accurate and reliable. However, no responsibility is assumed by InvenSense for its use, or for any infringements or patents or other rights of third parties that may result from its use. Specifications are subject to change without notice. Certain intellectual property owned by InvenSense and described in this document is patent protected. No license is granted by Implication or otherwise under any patent or patent rights of InvenSense. This is an unpublished work protected under the United States copyright laws. This work contains proprietary and confidential information of InvenSense Inc. Use, disclosure or reproduction without the express written authorization of InvenSense Inc. is prohibited. Trademarks that are registered trademarks are the property of their respective companies.

This publication supersedes and replaces all information previously supplied. InvenSense sensors should not be used or sold for the development, storing, production and utilization of any conventional or mass-destructive weapons or any other weapons or life threatening applications, as well as to be used in any other life critical applications such as medical, transportation, aerospace, nuclear, undersea, power, disaster and crime prevention equipment.

Copyright ©2011 InvenSense Corporation.

Table of Contents

1 PURPOSE AND SCOPE 3
2 REFERENCE DOCUMENTS AND SOFTWARE PACKAGES 4
3 MOTIONFIT™ SDK 5
3.1 SUPPORTED HARDWARE 5
3.2 SUPPORTED SOFTWARE 6
4 HARDWARE AND SOFTWARE SETUP 7
4.1 HARDWARE SETUP 7
4.2 SOFTWARE SETUP TO RUN THE DEMO APPLICATION 8

InvenSense Confidential - Pre Release version

1 Purpose and Scope

This document describes the InvenSense MotionFit™ SDK quick start guide. It is intended as a user to run the MotionFit™ SDK with PC application.

2 Revision History

Revision Date	Revision	Description
02/14/2012	1.0	MotionFit™ SDK Quick Start Guide
02/22/2012	1.1	Updates to reflect sales team feedback
3/7/2012	1.2	Added section on programming the SDK

InvenSense Confidential - Pre Release version

3 Reference Documents and Software Packages

1. **MotionFit™ SDK application note**: This document covers the board components and hardware schematics and other features.
2. **MotionFit™ SDK functional specification**: This document covers the eMPL lite specification and functional modules.
3. **Embedded MotionApps Lite release package**: This is the firmware which is running embedded MotionApps lite
4. **Teapot application software package**: This is the application program running on the PC which can be used to run the MotionFit™ SDK demonstration.

InvenSense Confidential - Pre Release version

4 MotionFit™ SDK

4.1 Supported Hardware

MotionFit™ Board

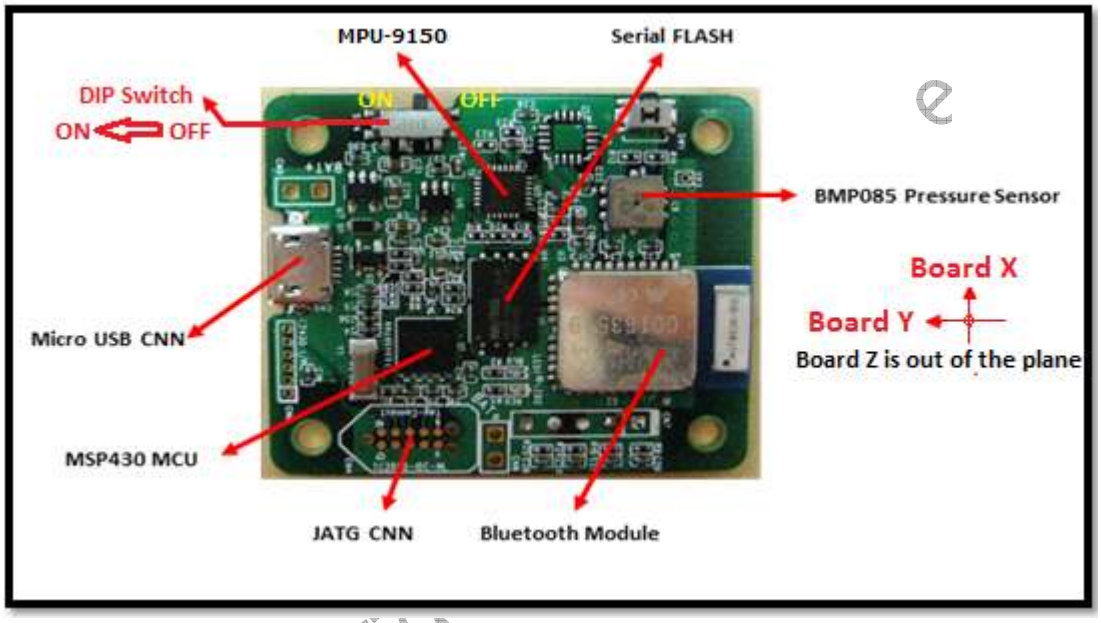

Figure 4.1.0 MPU 9150 MotionFit™ Board

Please refer to the MPU-9150™ MotionFit™ SDK Application Notes for details on each of these sensors and hardware schematics.

InvenSense Confidential Version

4.2 Supported Software

Firmware eMPL-wSDK-Test

1. The MotionFit™ board comes with preloaded firmware which outputs quaternion via Bluetooth link. The code is implemented using code composer studio from TI. The link below provides the documentation for code composer

   http://processors.wiki.ti.com/index.php/Category:Code_Composer_Studio_v5 .

2. The software setup for eMPL lite default which can be changed according to the Function Specification Document referred as MotionFit™ SDK Functional Specifications. Please check the main.c in the application release package. "embedded MotionApps Lite release package"

Feature	Settings
Quaternion Update Rate	20Hz
Quaternion Resolution	16 Bits
Quaternion Range	Full 360º

Application Test Software 1 (Teapot demo using visual studio)

1. The Teapot demo software is included with the package which is PC based application which rotates the teapot based on the MotionFit™ SDK. The mounting matrix for the device assumes the direction as in the figure 4.1.0. The board connects to the PC based on the Bluetooth link. Teapot demo software is provided with source as well.

Application Test Software 2 (debug console app with cube demo using python)

2. The cube demo is written in python script and needs python install on the computer. This cube demo lets user do more than just the orientation. He can use the keyboard to press "t" to do self-test. Press "mfa" to show accel value. For complete set of functionality please look at the "Embedded MotionApps Lite release package" and also the python eMPL_wSDK-client.py script.

InvenSense Confidential - Pre Release version

5 Hardware and Software Setup

5.1 Hardware Setup

Charge the unit via USB connector for about 2 hours. Turn on the SDK using the DIP switch on the board. Check the figure 4.1.0 for ON and OFF position of the switch.

1. Configure the Bluetooth interface on the PC. Go to "*control Panel\All Control Panel Items\Devices and Printers\"* then choose *"Add a device"* and choose blue "BlueRadios" device (figure 4.1.0)
2. If the computer asks for the password, enter 0000 (Zeros in numerical)
3. The PC will add the required driver to successfully add the device to the PC
4. Go to the Control Panel\All Control Panel Items\Devices and Printers\ to properties and the check the port number of the Bluetooth serial port (figure 5.1.1). In this example the port number is 66, but we will call it just COM# from now on. After successfully pairing, power cycle the MotionFit™ SDK once by flipping the DIP switch OFF and ON as in figure 4.1.0

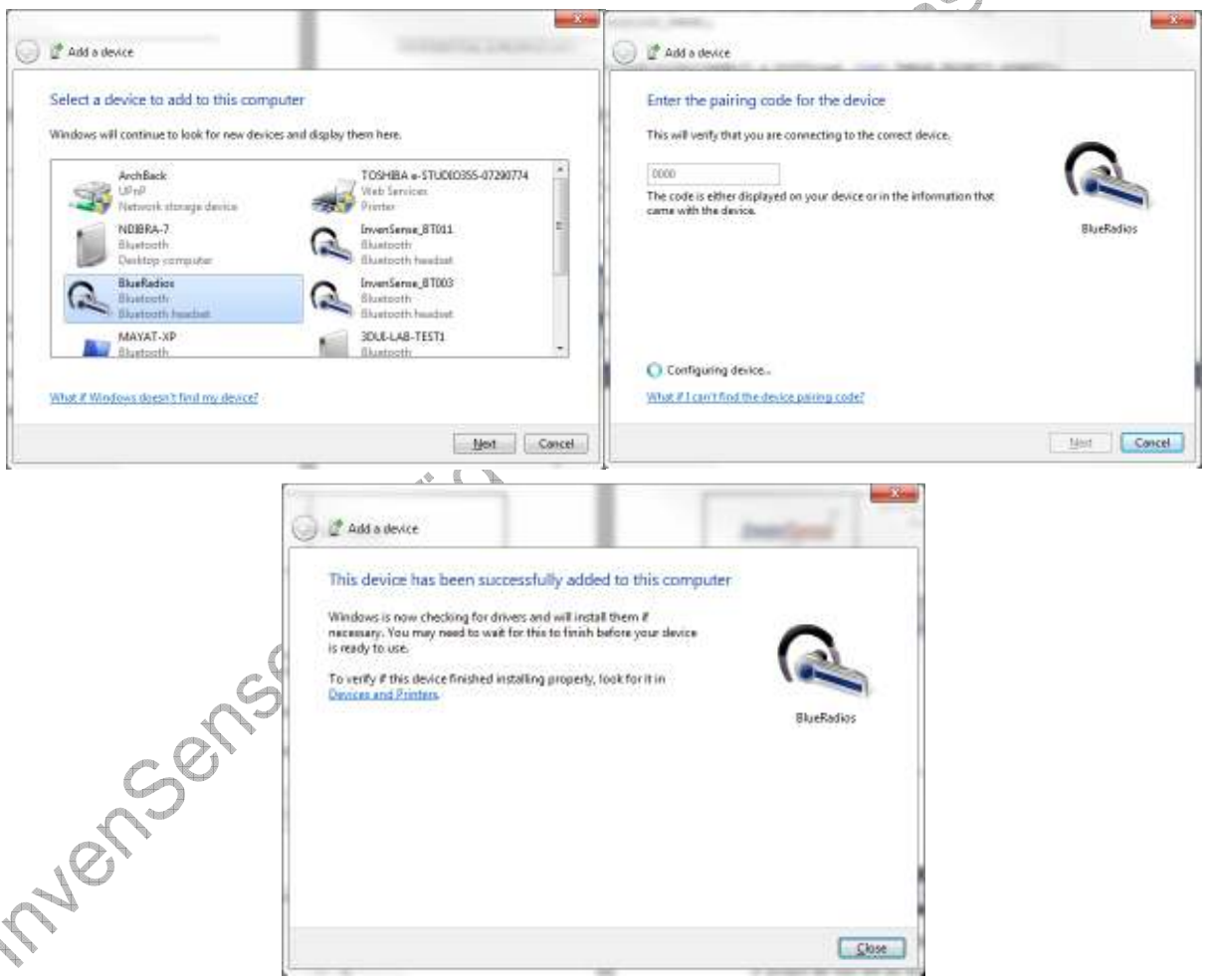

Figure 5.1.0 Adding BlueRadios to the PC (Steps 1, 2, 3)

InvenSense Confidential Version

Figure 5.1.1 Right click the added device to go to properties and check the port number (Step 4)

5.2 Software setup to run the Demo Application

Teapot Application

In order to demonstrate Embedded MotionApps Platform, a "Teapot" application has been provided. Download the file named 'MotionFit Wireless SDK' under the software download section of the developer corner (http://www.invensense.com/developers/index.php?_r=downloads). This application was developed using Microsoft Visual Studio and the entire solution is included in the release package. You will find the Teapot Application executable in the release folder. You have to create a shortcut to the executable, right click on the shortcut, select properties, and add the COM# number in the target field. Com# is represented by 15, in the example shown in Figure 5.2.0.

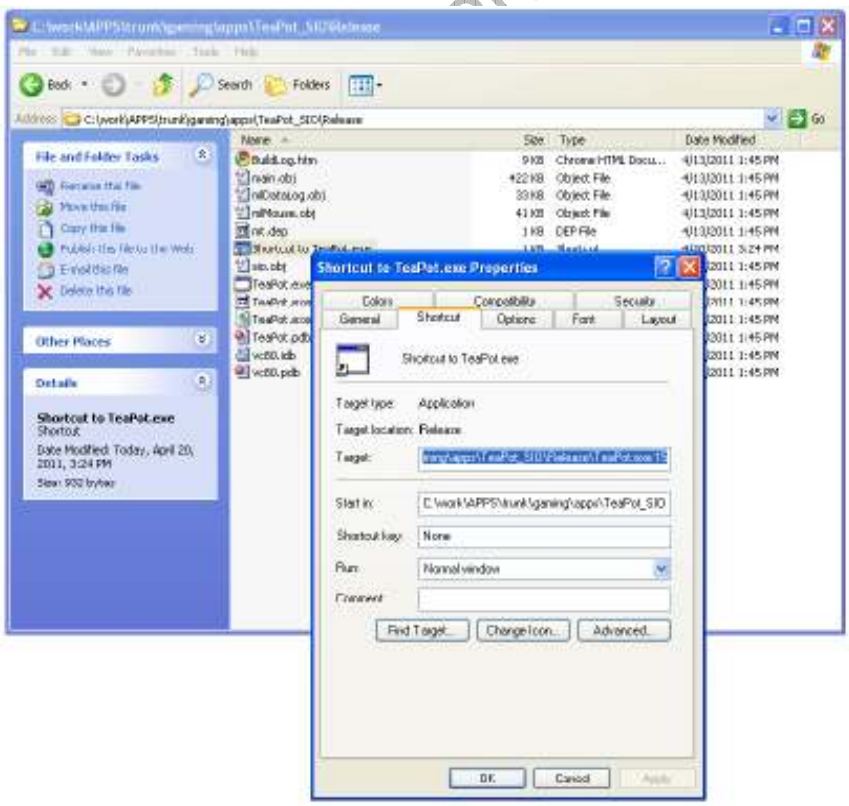

Figure 5.2.0 Adding COM port number to the shortcut of the application

InvenSense Pre Release version

Run the application by double clicking on the shortcut. A 3D teapot will be displayed on another window, as shown in Figure 5.2.1. Notice the orientation of the teapot change in accordance to the Embedded MotionApps Platform sensor fusion output. When the application is launched, you will notice the pointer and teapot drift continuously. This is because the sensors are uncalibrated. Place the platform on a stable surface and do not move it till you see the drift stop. A console window will be launched upon starting the teapot application. Close the console window to exit the application.

Figure 5.2.1 Teapot Application

Please note that prior to moving the SDK you would need to wait for the BLUE LED to turn off. As soon as you start moving the SDK, the BLUE LED will turn on to indicate recognition of a motion event. It is recommended to wait for at least one no motion event before you using the SDK, since a no motion event will remove the GYRO drift. The RED LED on the board indicates the data transfer to the PC

Debug Console app (Optional demo to run if needed)

(eMPL-wSDK-client.py)

A debugging console application is provided to aid developers using the MotionFit™ SDK. This application is provided as Python source code. It has the following functionality:

1. Provides a console for MPL_LOG messages.
2. Provides a simple cube demo for visualizing sensor fusion output. This cube demo provides identical functionality to the Teapot application from within the debug console.

Because the debug console is provided as Python source code, the user will need to install the following components to have a working Python environment.

1. Python 2.5.4: The debug console was tested with Python 2.5.4 on Windows XP and Windows 7. Download the Windows Installer (python-2.5.4.msi) from http://www.python.org/download/releases/2.5.4/
2. Pyserial 2.5: The debug console requires pyserial to communicate on the serial port. Download the Python 2.x Windows Installer (pyserial-2.5.win32.exe) from http://pypi.python.org/pypi/pyserial

InvenSense Confidential - Pre-release version

3. Pygame 1.9.1: The debug console uses the pygame library to render the cube demo. Download the Windows / Python 2.5 release (pygame-1.9.1release.win32-py2.5.exe) from http://www.pygame.org/download.shtml
4. You must install Python before installing the pyserial and pygame libraries. After installing python use the command prompt to install pyserial in command prompt, "python install pyserial.py "
5. You will want to add Python to your PATH for convenience. In Windows, right click on My Computer, select Properties, then in the advanced tab, select Environment Variables. In the user variables (top section), select Path, click Edit, and add the directory C:\Python25 to the end of the list.
6. Once you have a working Python environment, you can run the debug console application from the command line with the command. To accept data one has to give the python location of the python cube relative to the release folder as below
7. Embedded_MotionApps-MPU9150-Rel-MSP430F5528-V4_1_0_INVN_9x_HOST-2012-02-13\pycube

```
python eMPL_wSDK-client.py <comport>
```

Where `<comport>` is the COM# as described in figure 5.1.1

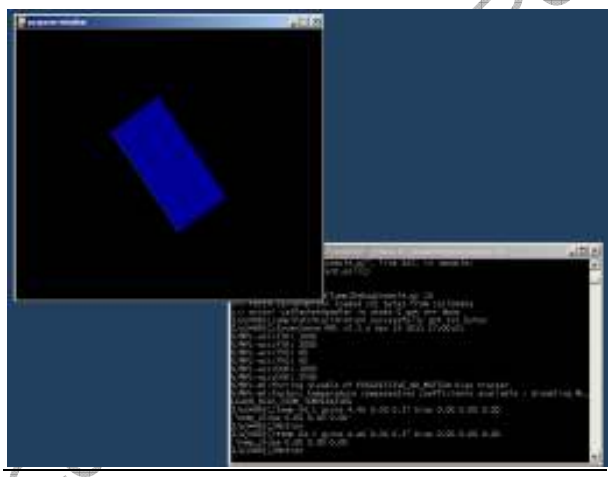

Figure 5.2.2: Cube Demo Application running on console

6 Programming the SDK

6.1 The MotionFit™ SDK comes pre-flashed with software needed to generate calibrated quaternion. The remainder of this section is needed only if the user would need to modify the embedded firmware.

6.2 Hardware needed for programming using JTAG

- MotionFit™ SDK (http://www.invensense.com/developers)

InvenSense Confidential Release version

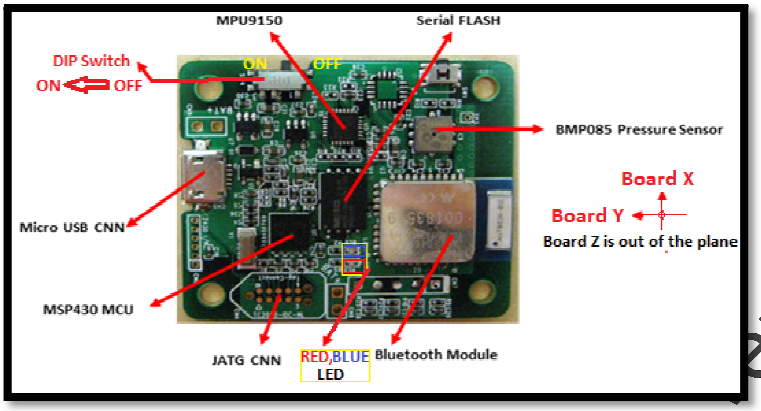

Figure 6.0 : MotionFit™ SDK hardware

- MSP-FET430UIF flash tool (http://www.ti.com/tool/msp-fet430uif)

Figure 6.1: Flash tool required for programming

- Flash extension connector and cable (more details on the extension cable please refer to http://www.tag-connect.com/tag-connect-2050-IDC-TG2050IDC). For obtaining the Connector PCB board please contact InvenSense (sales@invensense.com)

Figure 6.2: The red strip on the cable points to PIN1, arrow on the PCB

InvenSense Confidential Release Version

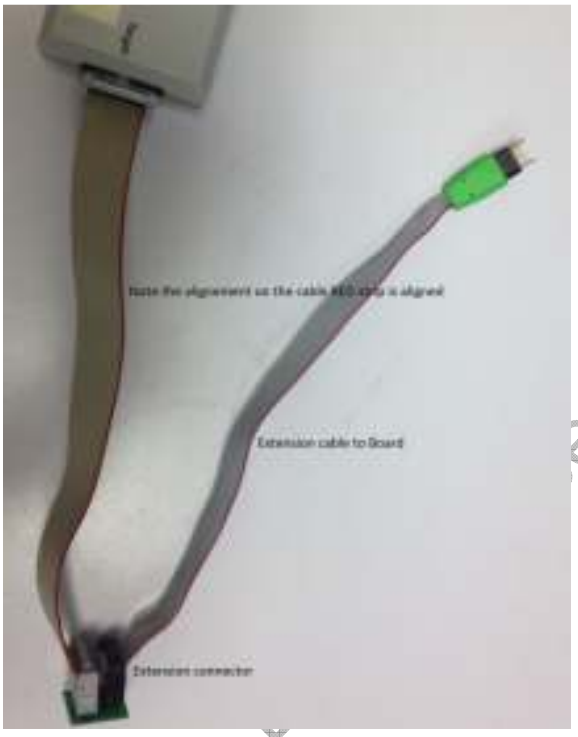

Figure 6.3: Flash extension cable which connects to the wearable sensor. Please align the two connectors so that the red strip of both cables is on the same side.

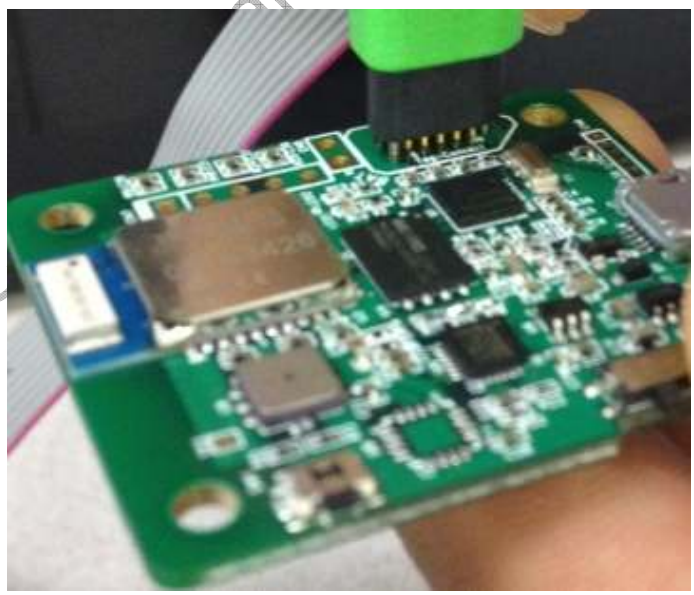

Figure 6.4: How to place green connector on SDK

InvenSense Confidential - For Internal Use version

6.3 Software components needed to program the SDK

Loading software on MCU

Option 1 – PC Program to load the software (that loads firmware on MCU) : Please install the software (http://www.ti.com/tool/flash-programmer) on your PC. For questions on installation or usage of this program please contact TI.

Option 2 - Code composer Studio: The code composer studio from TI can be used for application development on the SDK. Information on CCS can be obtained from http://www.ti.com/tool/ccstudio or http://processors.wiki.ti.com/index.php/Download_CCS

MotionFit™ SDK Software

To integrate the Embedded MotionApps Lite you will need to locate eMPL_*.Hex file which is included in the eMPL package. Refer to the MotionFit™ SDK functional specification for supported API calls needed to build your application.

Pressure sensor Driver for BMA085

For extracting pressure sensor data, please locate drivers on Bosch website (

http://www.bosch-sensortec.com/content/language1/html/3477.htm)

Macronix serial flash Driver for MX25L25635E

For information on the serial flash, please refer to MX25L25635E, LLD, v0.3.zip on Macronix website.

6.4 How to use Embedded MotionApps Lite features and integrate with an application

MotionFit™ package comes with an example application (that instantiates Embedded MotionApps Lite API calls) to transmit sensor fusion data via serial UART. The next sections cover details about the Embedded MotionApps™ architecture and example application code.

6.5 Embedded MotionApps™ Platform Overview

The different functional layers of Embedded MotionApps™ Platform are as shown in Figure 6.5 below.

InvenSense Confidential - Pre Release version

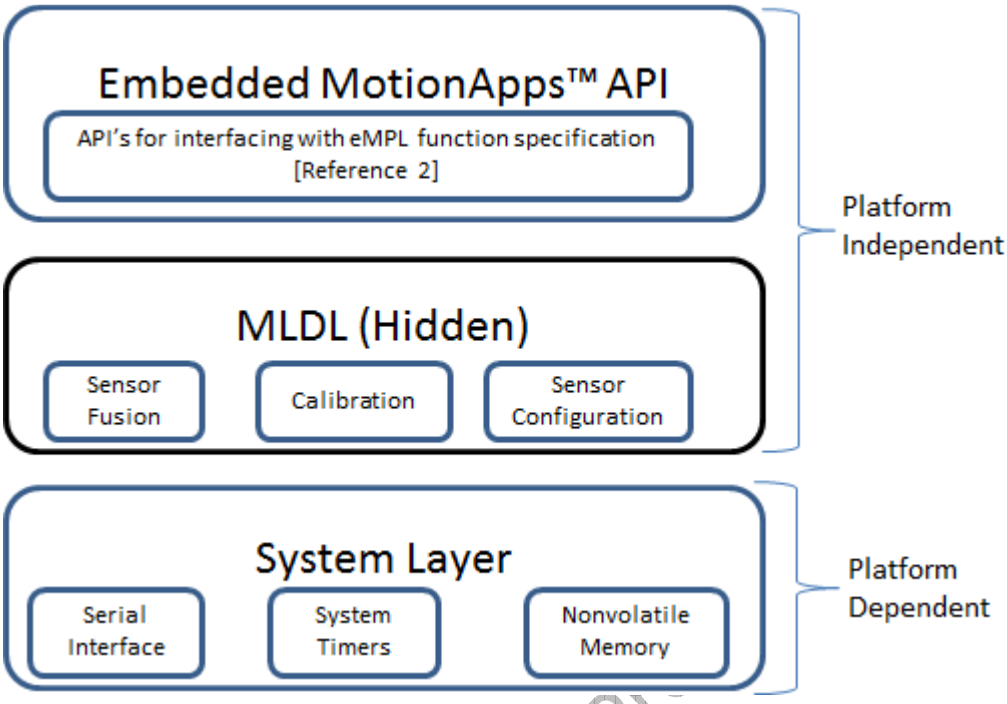

Figure 6.5: Embedded MotionApps™ Platform

6.6 Embedded MotionApps™ API layer

This layer provides the required API's to interface to setup, update data and get sensor fusion outputs. Please check the *.h eMPL_include folder on the API that is available for your application. Please refer to Functional Spec (referenced item #2 in section 3 of this document) for details on API calls.

6.7 MLDL layer

This is the core layer which has the entire algorithm for calibration and sensor fusion. This code interfaces with the sensors to manage configuration of DMP and also perform sensor fusion. This is code is hidden (binary) in the release package.

6.8 System layer (MLSL)

- Serial Interface is the platform dependent implementation for serial communication. The Embedded MotionApps™ platform uses this to communicate with the MPU device over an I^2C bus. Please check the release package *.h file in the main program folder.
- System Timers (MLOS) provides an interface to a system timer and delay routine. The Embedded MotionApps™ platform requires a system timer with millisecond resolution. Please check the release package *.h file in the main program folder on the available API functions.
- Nonvolatile memory (UStore manager) provides an interface to TI's internal flash to store the calibration data when needed. Check the msp430_nvram.h on the details of the function used for storing and retrieving calibration data.

6.9 Embedded MotionApps™ sensor configuration

The Embedded MotionApps™ Platform has a power mode API which allows the user to power on and power off the individual sensors. Once the sensor is not being used the power will be disabled thus enabling a low power mode.

InvenSense Confidential - Pr

eMPL_init must be called before any other MotionApps™ Platform APIs can be used. After this function is called, all devices will still be powered off. The user can then call function as below to select a sensor and the power will be managed accordingly. eMPL_start_mpu (1) enables 9 axis while eMPL_start_mpu (0) will enable 6 axis (Gyro and Accel) and with inv_set_mpu_sensors you can select accel only mode.

Figure 6.6: Embedded MotionApps™ Platform State Machine Diagram

Note:

- When built with input 0, the six axis version with the accelerometer and gyros will turn on and when passed with 1 which represent 9-axis, eMPL_start_mpu will power on the accelerometer, gyros, and magnetometer.
- In inv_set_mpu_sensors if accel only is chosen than accel only will be powered on.

6.10 Compilation Setup

The Embedded MotionApps™ Platform requires code composer studio v4.0 or later and the default chip selected is MSP430F5528 which is the one on the SDK hardware.

6.11 Steps to bring up the project under code composer studio

- Download and install the code composer studio from TI website (http://www.ti.com/tool/ccstudio). Please note that TI offers a free version of CCS which has restrictions on code size that makes it unusable with MotionFit SDK hence you would need the full featured version. TI offers a trial version of the software which can be used for configuring your MotionFit application.
- Download the release package from developer corner and unzip (or 7zip) the folder to extract Embedded MotionApps Lite software.
- Open the code composer, go to File -> Imports as in figure below
- Open Existing CCS/CCE Eclipse Projects
- Browse the folder where the eMPL lite is located and open the project
- Now the project opens up and it is ready to compile, look for main.c to see how the example project is being integrated to the eMPL.

InvenSense Confidential - Pre Release

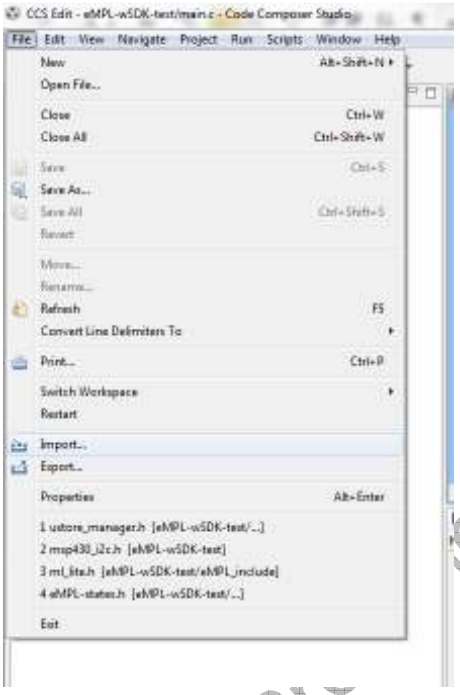

Figure 6.7.0: Select Import to import the project

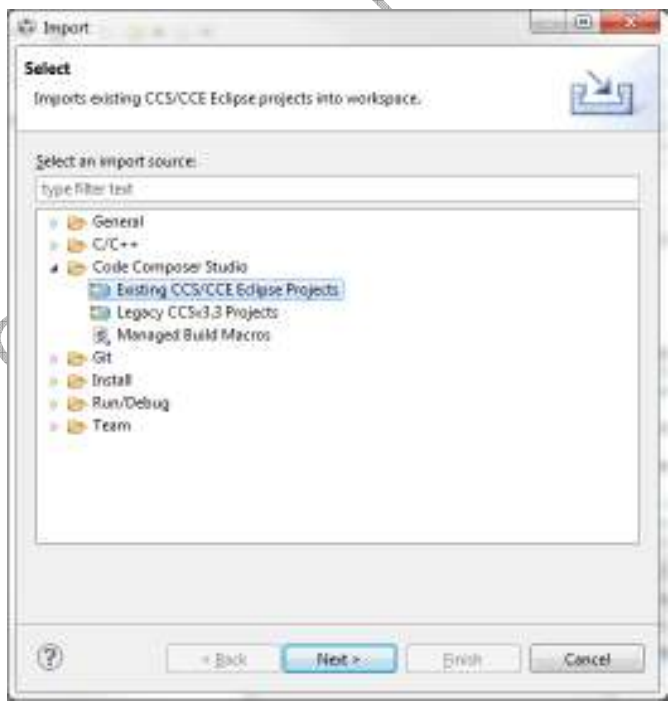

Figure 6.7.1: Open the existing project and put it in the project

InvenSense Release version

6.12 Review of main.c file to build sample application with Embedded MotionApps Lite

The file is self-explanatory in term of inline documentation and therefore we will go through the important steps only. This example application program is designed to continuously transfer quaternion data at 20Hz update rate. The value of the quaternion is transmitted to the UART port and then transferred via Bluetooth module. The previous section shows how to use the program with PC applications.

- set_up_msp430 () is needed to setup the system layer functions. This will enable the I2C, clocks, UART and flash storage functionality in the project
- msp430_uart_register_rx_cb (rx_cb) is optional function call but for the program to work with sample application which runs on the PC, this is required as it will enable the Uart receiver call back functionality. rx_cb is the receiver call back. One can modify this functionality based on the application on how to interpret the Uart receiver data.
- It is very important to initialize using eMPL_init() since this will initialize the calibration data. It also initializes the serial layer for the sensor interface.
- Starting eMPL using eMPL_start_mpu (1) will enable eMPL on 9 axis mode and sets the default bias trackers. Here are the major default values which are initialized. There are other defaults sets as well.
    - 20 Hz fifo output rate (This is the default and can be changed)
    - Range is 2G for accelerometer
    - Gyro range is 2000 dps
    - 9 axis or 6 axis abased on the input 1 or 0
- Motion call back can be enabled optionally. The motion/no motion event is used to light up an LED. Motion call back is triggered in the event of motion and in the event of no motion. Software MLDL layer takes care of the algorithm in detecting motion and no motion events.
- In this example application the MCU continuously transmits the data and is set in a while loop. Based on the application one can change the use case of the application.
- eMPL_update_data() will execute all the required functions and algorithm for sensor fusion
- send_quat () function packages the quaternion and sends it out using Uart module at 115200 baud rate.
- If queried for mfa, mfg, mfm, the accel, gyro and magnetometer calibrated data is transferred in about 1 second.

InvenSense Confidential - Pre Release version