



**InvenSense Inc.**  
1197 Borregas Ave., Sunnyvale, CA 94089 U.S.A.  
Tel: +1 (408) 988-7339 Fax: +1 (408) 988-8104  
Website: www.invensense.com

Document Number: AN-MPU-9150IMF-03  
Revision: 1.0  
Release Date: 09/18/2012

# InvenSense MotionFit™ 5.1 SDK User Guide

## Release 1.0

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 This is an unpublished work protected under the United States copyright laws.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

<b>1</b>	<b>PURPOSE AND SCOPE</b> .....	<b>3</b>
<b>2</b>	<b>REVISION HISTORY</b> .....	<b>3</b>
<b>3</b>	<b>REFERENCE DOCUMENTS AND SOFTWARE PACKAGES</b> .....	<b>4</b>
<b>4</b>	<b>MOTIONFIT™ SDK</b> .....	<b>5</b>
4.1	SUPPORTED HARDWARE .....	5
4.2	SUPPORTED SOFTWARE .....	6
<b>5</b>	<b>HARDWARE AND SOFTWARE SETUP</b> .....	<b>7</b>
5.1	HARDWARE SETUP .....	7
5.2	SOFTWARE SETUP TO RUN THE DEMO APPLICATION .....	8
<b>6</b>	<b>PROGRAMMING THE SDK</b> .....	<b>10</b>
6.1	MODIFYING THE FIRMWARE .....	10
6.2	HARDWARE NEEDED FOR PROGRAMMING USING JTAG .....	10
6.3	SOFTWARE COMPONENTS NEEDED TO PROGRAM THE SDK .....	11
6.4	HOW TO USE EMBEDDED MOTIONAPPS 5.1 FEATURES AND INTEGRATE WITH AN APPLICATION.....	12
6.5	EMBEDDED MOTIONAPPS™ PLATFORM OVERVIEW .....	12
6.6	CALIBRATION AND SELF-TEST.....	13
6.7	ULTRA-LOW POWER FEATURES (APPLICABLE TO THE MPU-6050 AND MPU-9150) .....	13
6.8	COMPILATION SETUP .....	13
6.9	STEPS TO BRING UP THE PROJECT IN CODE COMPOSER STUDIO.....	13
6.10	REVIEW OF THE MAIN.C FILE TO BUILD A SAMPLE APPLICATION WITH EMBEDDED MOTIONAPPS 5.1 .....	15



## 1 Purpose and Scope

This document describes the InvenSense MotionFit™ 5.1 SDK User Guide. It is intended as a guide for the user to run the MotionFit™ SDK with the PC application.

## 2 Revision History

Revision Date	Revision	Description
09/18/2012	1.0	InvenSense MotionFit™ 5.1 SDK User Guide



### 3 Reference Documents and Software Packages

1. **MotionFit SDK v5.1 APIs Specification.**
2. **MPU-9150™ MotionFit™ Reference Board User Guide: (This document covers the board components, hardware schematics, and other features.)**
3. **MotionFit with eMPL 5.1 release package: (This is the firmware which is running Embedded MotionApps 5.1.)**
4. **Teapot application software package: (This is the application program running on the PC which can be used to run the MotionFit™ SDK demonstration.)**

## 4 MotionFit™ SDK

### 4.1 Supported Hardware

MotionFit™ Board

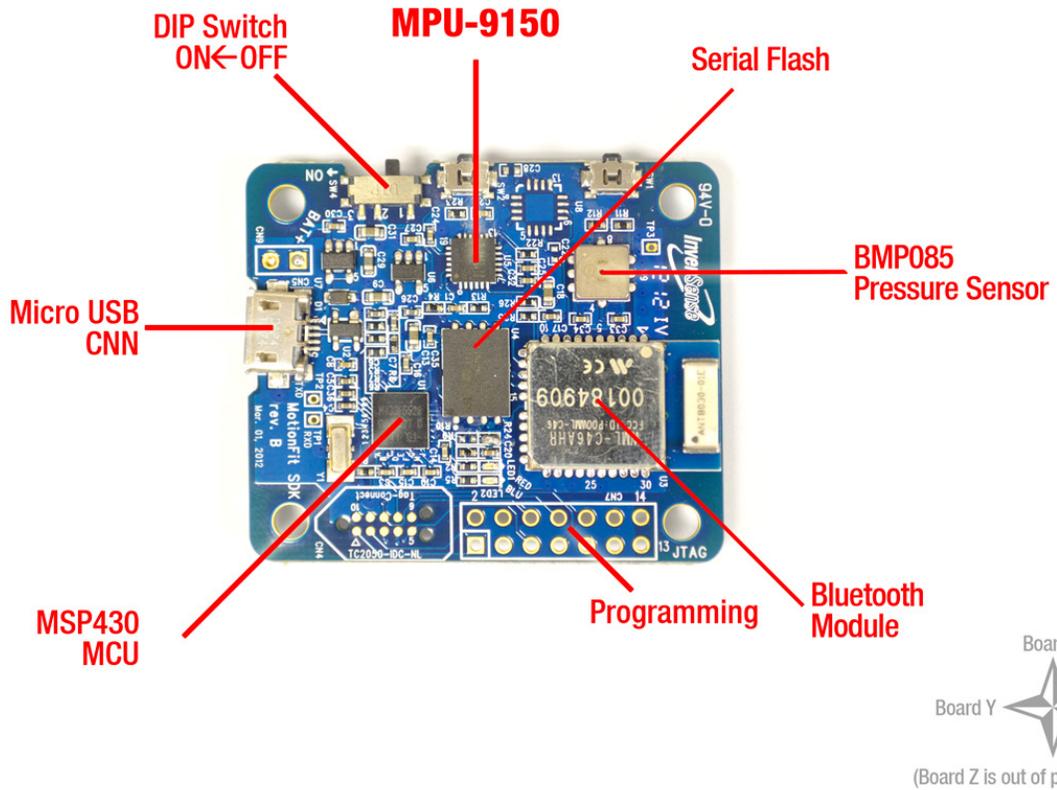


Figure 4.1.0 MPU 9150 MotionFit™ Board

Please refer to the MPU-9150™ MotionFit™ Reference Board User Guide for details on each of these sensors and hardware schematics.



## 4.2 Supported Software

### Firmware eMPL-5.1

1. The MotionFit™ board comes with preloaded firmware which outputs quaternion via a 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\\_Compiler\\_Studio\\_v5](http://processors.wiki.ti.com/index.php/Category:Code_Compiler_Studio_v5) .

2. The default software settings for eMPL can be changed according to the MotionFit SDK v5.1 API Specification document. Please check the main.c file in the application release package.

Feature	Settings
Quaternion Update Rate	50Hz
Quaternion Resolution	32 Bits
Quaternion Range	Full 360°

### Application Test Software 1 (Teapot demo using visual studio)

1. The Teapot demo software is included in the package which is a PC based application that 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 installed on the computer. This cube demo lets users do more than just observe the orientation. For a complete set of functionality please look at the python eMPL-client.py script.

## 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 the ON and OFF positions 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 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](#).

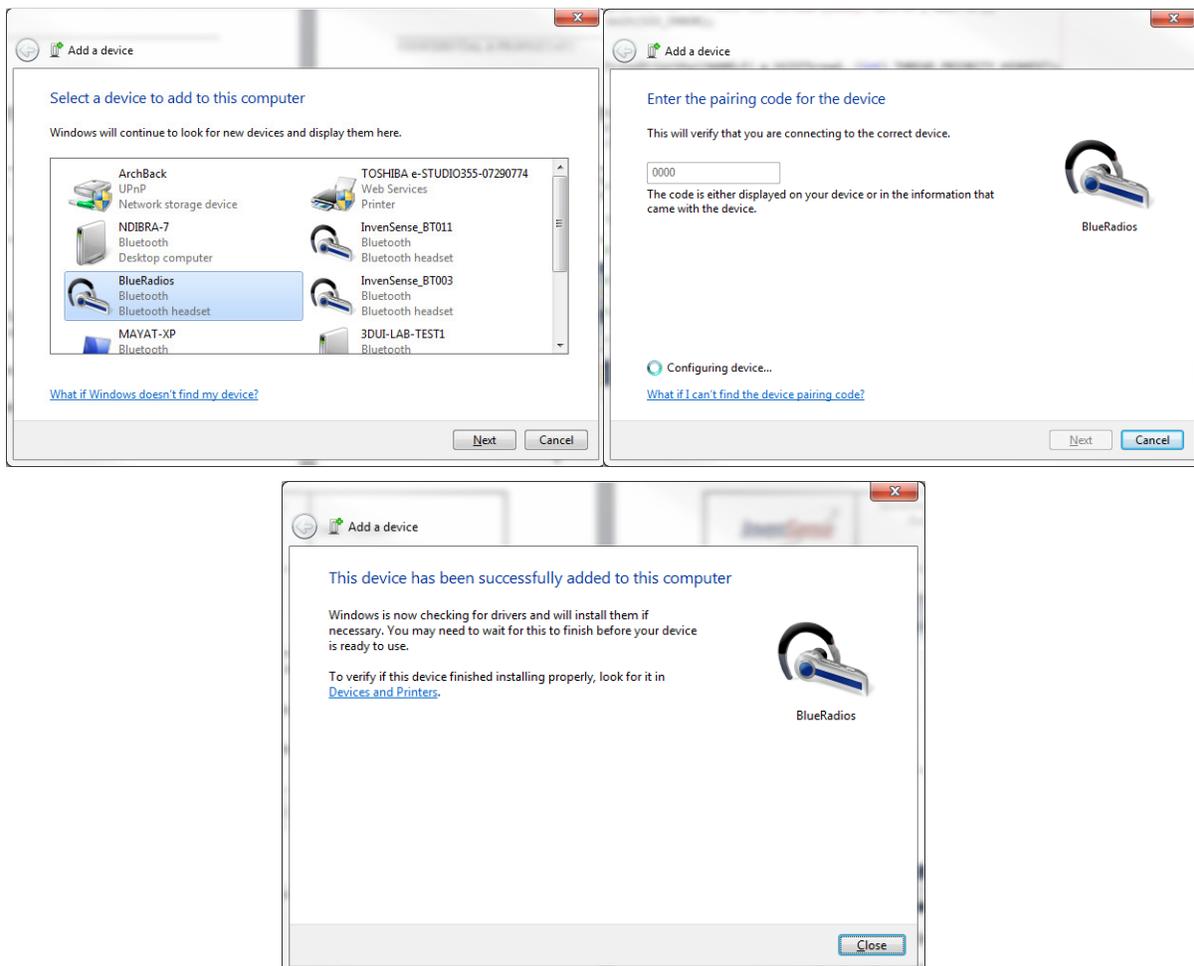


Figure 5.1.0 Adding BlueRadios to the PC (Steps 1, 2, 3)

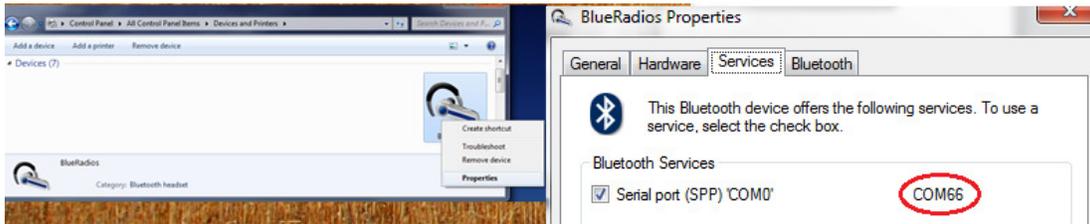


Figure 5.1.1 Right click the added device to go to properties and check the port number (Step 4)

## 5.2 Software Setup to Run the Demo Application

### Teapot Application

In order to demonstrate the Embedded MotionApps Platform, a “Teapot” application has been provided. Download the file named ‘MotionFit Wireless SDK’ under the software download section in the Developers 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# in the target field. Com# is represented by 15, in the example shown in Figure 5.2.0.

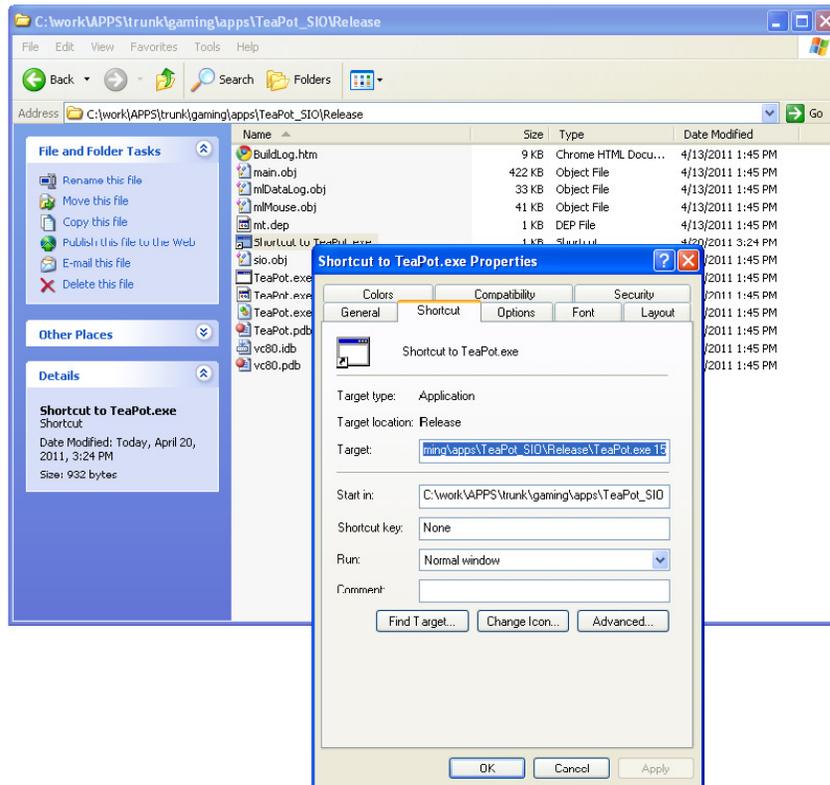


Figure 5.2.0 Adding COM port number to the shortcut of the application

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 changes 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 not calibrated. Place the platform on a stable surface and do not move it until you see the drift stop. A console window will be launched upon starting the teapot application. Close the console window to exit the application.

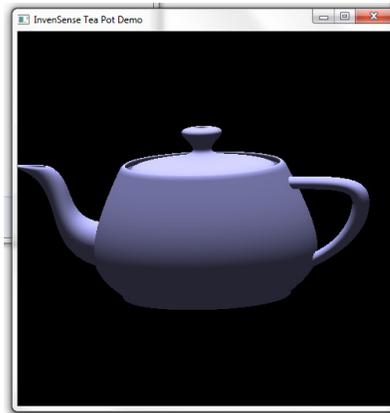


Figure 5.2.1 Teapot Application

Please note that prior to moving the SDK you will 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 start 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-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:

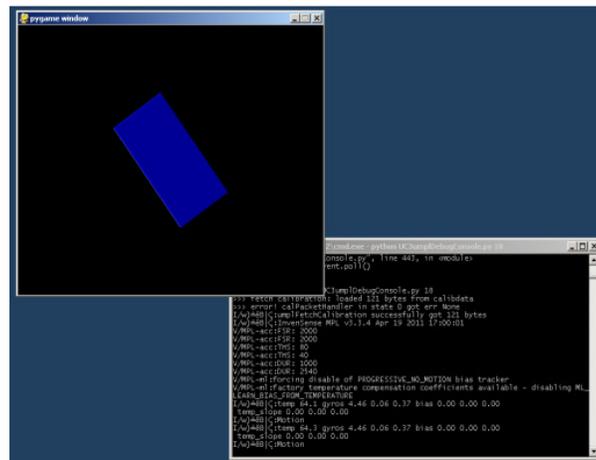
1. Provides a console for MPL\_LOG messages.
2. Provides a simple cube demo for visualizing sensor fusion output. This cube demo provides identical functionality to the Teapot application from within the debug console.

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](http://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>.
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, “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. MotionFit\_With\_eMPL511\_Release.zip\MotionFit\_With\_eMPL511\_Release\core\eMPL-hal\eMPL-Client
8. python eMPL-client.py <comport>

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



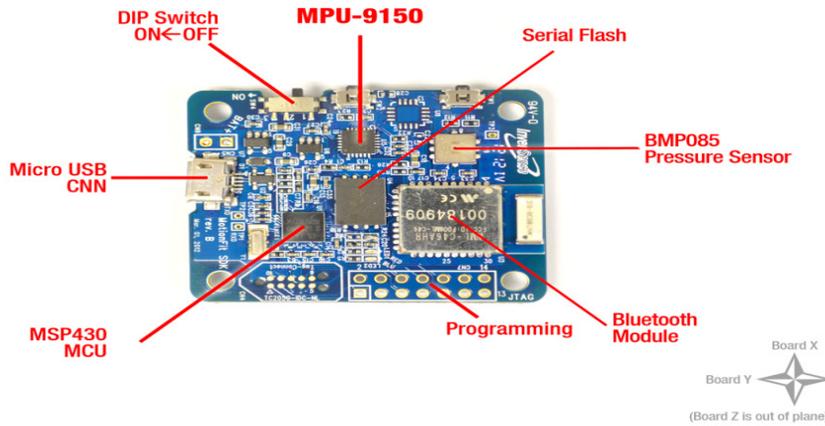


Figure 6.0: MotionFit™ SDK hardware

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



TEXAS INSTRUMENTS  
MSP430

Figure 6.1: Flash tool required for programming



### 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 the 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](http://processors.wiki.ti.com/index.php/Download_CCS).

### **MotionFit™ SDK Software**

To integrate Embedded MotionApps 5.1 you will need to locate the eMPL\_\*.Hex file which is included in the eMPL package. Refer to the MotionFit™ SDK v5.1 API Specification for supported API calls needed to build your application.

### **Pressure Sensor Driver for BMA085**

For extracting pressure sensor data, please locate drivers on the 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 the Macronix website.

## **6.4 How to use Embedded MotionApps 5.1 Features and Integrate with an Application**

The MotionFit™ package comes with an example application (that instantiates MotionFit v5.1 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**

Embedded MotionApps 5.1 has been designed with a new architecture as explained below. The new architecture provides customers with a much more modular MotionTracking solution that consumes much less application processor resources, improving the overall power consumption.

There are three basic components in this architecture:

- Embedded Hardware Abstraction Layer (HAL):
  - a. This is the central point of communication between eMPL, device drivers, and the applications.
  - b. HAL provides function calls to initialize, enable, start, stop, and control all features in eMPL. Please refer to [1] for further details.
- Embedded MotionProcessing Library (eMPL):
  - a. eMPL provides a plug-in framework for single function features (e.g. gyro bias calibration, 9-axis MotionFusion, etc.)
  - b. eMPL provides a well-defined flow to register/unregister, initialization, start, stop, and generate features. Please refer to [1] for further details.
- Kernel driver:
  - a. Device drivers for all supported InvenSense devices are implemented in sysfs using the input subsystem.



## 6.6 Calibration and Self-test

Embedded MotionApps consists of the following key calibration and self-test features:

Calibration:

- Calibration for gyro bias
  - Fast no motion (<1 second convergence time)
  - Standard no motion (8 second convergence time)
  - SW temperature compensation
- InvenSense proprietary compass calibration
  - Standard fit
  - Vector
  - Small Motion using gyro

Self-test:

- Integrated MotionTracking device based self-test for gyroscope in the MPU-6050 and MPU-9150 (Refer to [1] for further details).
- Integrated MotionTracking device-based self-test for accelerometer in the MPU-6050 and MPU-9150 (Refer to [1] for further details).
- Integrated MotionTracking device-based self-test for compass in the MPU-9150 (Refer to [1] for further details).
- Self-test is integrated into the respective device drivers.

Please refer to [1] for a brief description of individual calibration functions and recommended settings.

## 6.7 Ultra-Low Power Features (applicable to the MPU-6050 and MPU-9150)

The ultra-low power features introduced in this release and listed below support fully autonomous operation, completely independent from the application processor:

- DMP-based pedometer
- DMP-based gestures
  - Directional Tap
  - Orientation (including Flip)
  - Screen Orientation
- Low Power Quaternion
- Low Power 6-Axis Sensor Fusion

In general, please refer to [1] on the description, usage, and integration of each of the above features.

## 6.8 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.

## 6.9 Steps to Bring Up the Project in Code Composer Studio

- Download and install the Code Composer Studio from the 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 the 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 the Developers Corner and unzip (or 7zip) the folder to extract the Embedded MotionApps eMPL-5.1 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 is located and open the project.
- Now the project opens up and it is ready to compile, look for the main.c file to see how the example project is being integrated into eMPL.

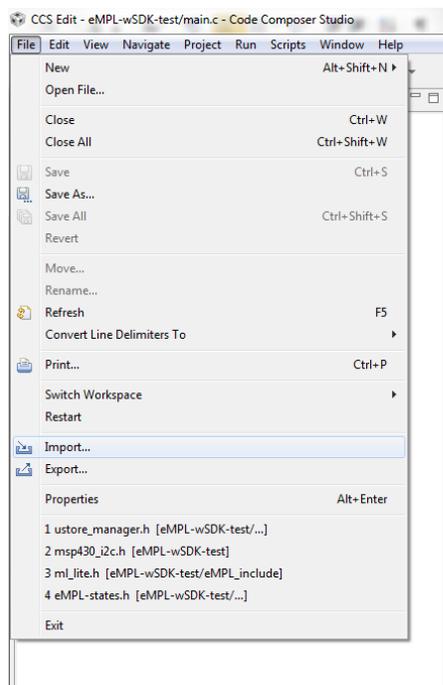


Figure 6.9.0: Select import to import the project

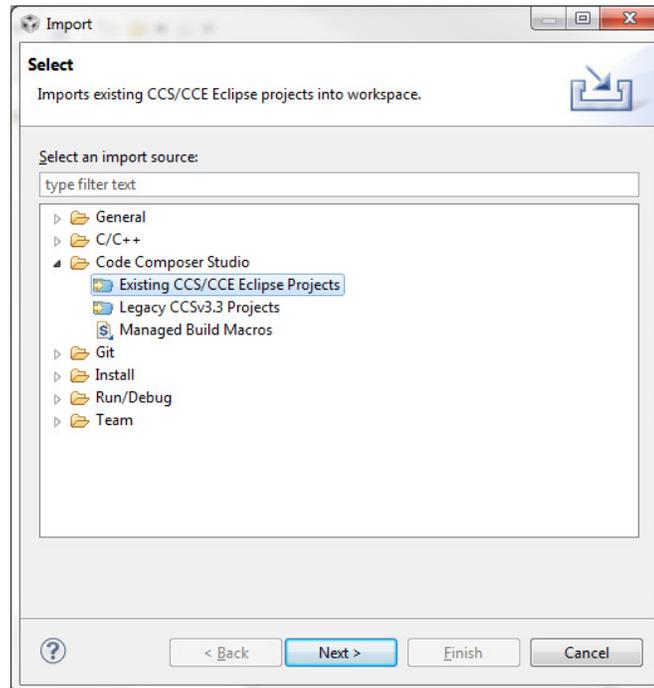


Figure 6.9.1: Open the existing project and put it in the project

## 6.10 Review of the Main.c File to Build a Sample Application with Embedded MotionApps 5.1

The file is self-explanatory in terms 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 50Hz 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.

- `msp430_clock_init(12000000L, 2); msp430_i2c_enable(); msp430_int_enable(); msp430_uart_init();` is needed to setup the system layer functions. This will enable the I2C, clocks, and UART functionality in the project.
- It is very important to initialize using `inv_init_mpl()`. It should be called first and once.
- Starting eMPL using `inv_start_mpl()` will enable eMPL. Here are the major default values which are initialized. There are other defaults sets as well.
  - 50 Hz fifo output rate (This is the default and can be changed)
  - Range for accelerometer can be set using `gyro_set_accel_fsr();`
  - Gyro range can be set using `gyro_set_gyro_fsr()`
- 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.
- `inv_execute_on_data()` will process the data it has received and update all the internal states and features that have been turned on. This function should be called after at least one of `inv_build_gyro()`, `inv_build_accel()`, or `inv_build_compass()` has been called.
- `eMPL_send_quat(data);` this function packages the quaternion and sends it out using the UART module at 115200 baud rate.