InvenSense Inc.
1745 Technology Dr., San Jose, CA 95110 U.S.A.
Tel: +1 (408) 988-7339 Fax: +1 (408) 988-8104
Website: www.invensense.com

Document Number: AN-EMAPPS-0.0.6
Revision: 1.0
Release Date: 04/17/2015

Motion Driver 608 – Quick Start Guide

Table of Contents

1 Revision History

Revision Date	Revision	Description
04/17/2015	1.0	Initial Release

2 Purpose

The Motion Driver 608 is an embedded software stack of the sensor driver layer developed for the Invensense ICM20608 6-axis motion sensor. The driver is very similar to the Motion Driver 6.1 and this document will detail the overview. It is recommended to read the App Notes in the Motion Driver 6.1 release to gain a deeper understanding of the driver.

3 Software Features

Here is a quick overview of the features the Motion Driver 608 provides

- Example project using STM32F4 MCU with IAR ARM toolchain
- Motion Processing Library
    - 6-axis Sensor Fusion
    - Gyro Temperature Compensation
    - In-Use Gyro Calibration
    - In-Use Accel Calibration
- Dynamically turn off/on sensors
- Dynamic rate change of sensor sampling
- Example python client with 3D model representation of sensor fusion data
- Data Types
    - Accel Data
    - Gyro Data
    - Euler Angles
    - Rotational Matrix
    - Quaternions
    - Heading
    - Linear Acceleration
    - Gravity Vector
- Low Power Motion Interrupt Mode
- Low Power Accel Only Mode
- Low Power 6-axis Mode
- Factory Hardware Self Test and Calibration algorithms

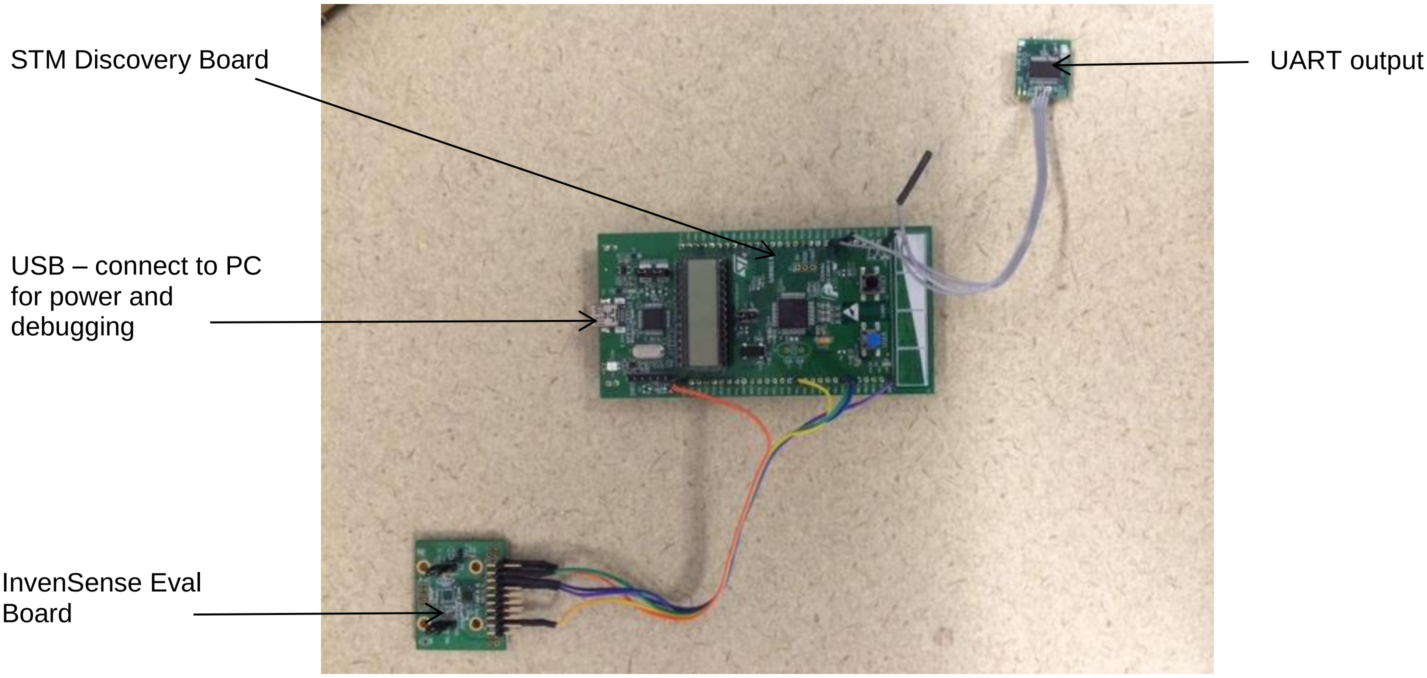
4 Release Package

MD608 release package contains example projects of STM32F4 using IAR toolchain. It also contains binary MPL libraries for the sensor fusion precompiled for ARM processors. MPL libraries for ARM use gcc 4.9.3, IAR, and Kiel ARM compilers. Compiler option settings are in the readme.

- **…\STM32F4_MD20608:** Directory which contains the IAR project for STM32F4 Discovery Evaluation Board and the InvenSense motion solution. The STM32F4 is a Cortex-M4 MCU core. The IAR project file is located under .\STM32F4_MD20608\Projects\eMD20608\EWARM\STM32F4_MD20608.eww
- **...\documentation :** All relevant documentations regarding MD608 is under this directory
- **…\eMPL-pythonclient :** Python client used to test and demonstrate the motion device performance as well as display log information
- **…\mpl libraries :** Directory which contains the InvenSense Proprietary binary MPL (Motion Processing Library) used in the MD608. ARM libraries are compiled it’s relative compilers detailed for the M0, M0+, M3, and M4.

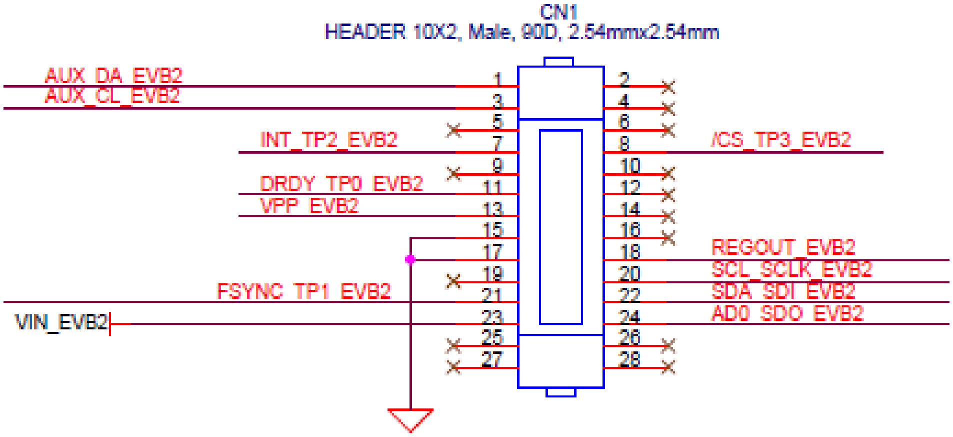
5 STM32F4 (Cortex-M4) Discovery Board Project

Please read the Motion Driver 6.1 App Note 1. The MD608 driver hardware setup is very similar with using the STM32F4 Discovery Board, a UART output, and an Invensense ICM20608 Evaluation Board. Connections of the bare metal boards will be similar to this



- **InvenSense Eval Board connection to Discovery Board**

The connection from the InvenSense eval board to the discovery board will require wiring between the two PCB boards. The InvenSense eval board pin outs are the same across all InvenSense eval boards.



To connect to the Discovery Board you will need to connect these 5 pins

EVB Header Pin Number	Description	Discovery Board GPIO Pin Number
7	INT output	PA1
17	GND	GND
23	VCC_IN	EXT_3V
20	I²C SCL	PB10
22	I²C SDA	PB11

The MD608 outputs data via from the UART port. The data is used by the python client to display information for the user. The pins are

Discovery Board UART Out Pin Number	Description
PA2	UART Tx
PA3	UART Rx

You will need to use a UART convertor to the PC. There are several UART to Serial or UART to USB convertors available.

6 Python Client

A python client is included with the release package to test the performance and display log information. The client can be found in the release package under the directory

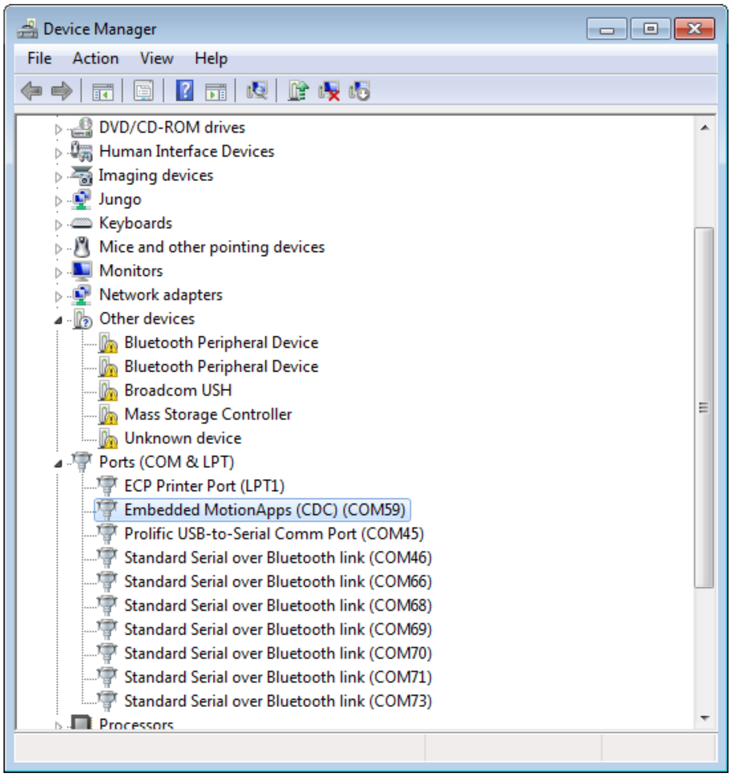
..\eMPL-pythonclient\

The python client also accepts user input and provides the input the sample HAL Application. The user would be able to enable/disable sensors, enable computation algorithms, enable hardware features, and view log information. You would need to install Python (version 2.5 and above), pyserial and pygame for the python script to execute.

- Installing Python 2.7 (32-bits version) or above, pyserial, and pygame

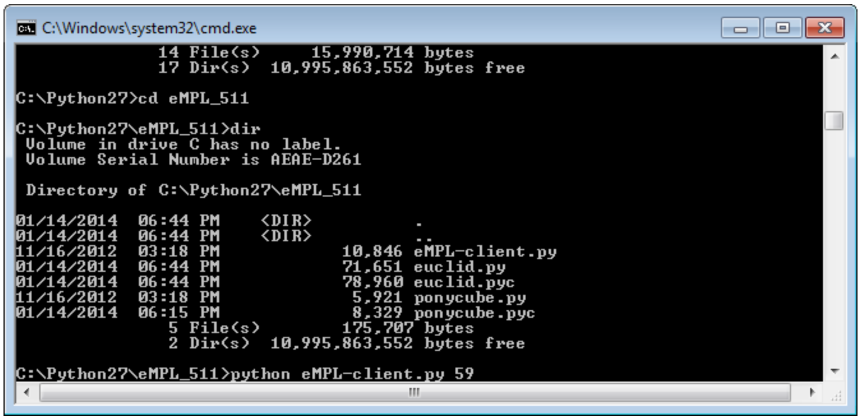
  Python: https://www.python.org/downloads/
  Pyserial: https://pypi.python.org/pypi/pyserial
  Pygame: http://www.pygame.org/download.shtml

- Connect your flashed and working hardware to your PC and find the COM port in the device manager if the connected device



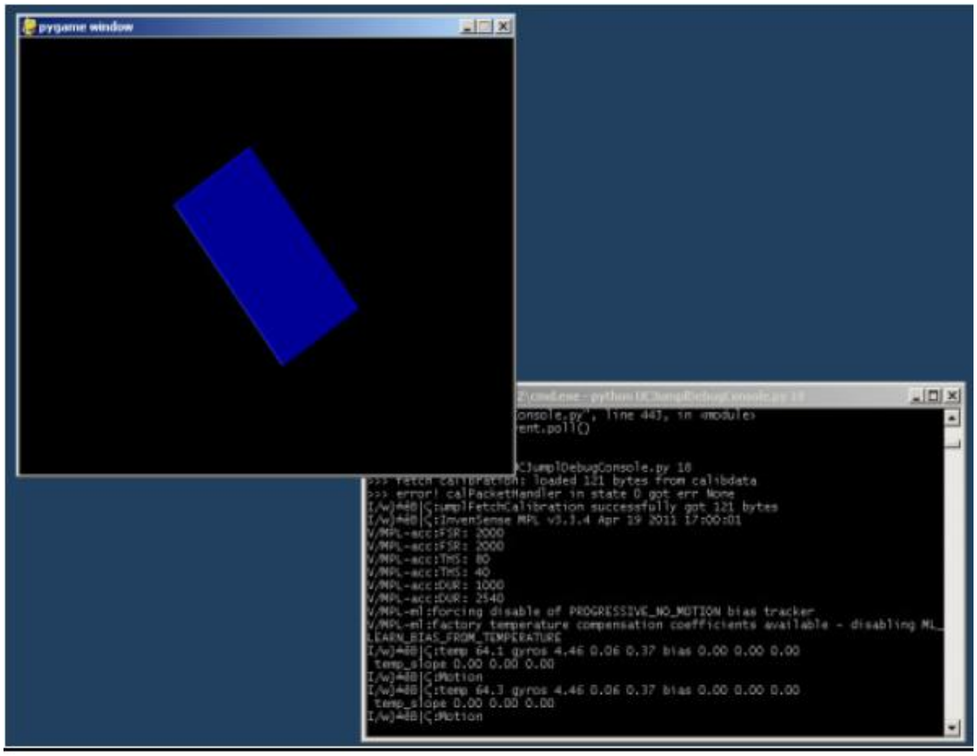
```
DVD/CD-ROM drives
Human Interface Devices
Imaging devices
Jungo
Keyboards
Mice and other pointing devices
Monitors
Network adapters
Other devices
    Bluetooth Peripheral Device
    Bluetooth Peripheral Device
    Broadcom USH
    Mass Storage Controller
    Unknown device
Ports (COM & LPT)
    ECP Printer Port (LPT1)
    Embedded MotionApps (CDC) (COM59)
    Prolific USB-to-Serial Comm Port (COM45)
    Standard Serial over Bluetooth link (COM46)
    Standard Serial over Bluetooth link (COM66)
    Standard Serial over Bluetooth link (COM68)
    Standard Serial over Bluetooth link (COM69)
    Standard Serial over Bluetooth link (COM70)
    Standard Serial over Bluetooth link (COM71)
    Standard Serial over Bluetooth link (COM73)
Processors
```

- Start the python client by opening up a command prompt window and browse to the python client directly and enter the following command
  - python eMPL-client.py <COM PORT NUMBER>



```
              14 File(s)     15,990,714 bytes
              17 Dir(s)  10,995,863,552 bytes free

C:\Python27>cd eMPL_511

C:\Python27\eMPL_511>dir
 Volume in drive C has no label.
 Volume Serial Number is AEAE-D261

 Directory of C:\Python27\eMPL_511

01/14/2014  06:44 PM    <DIR>          .
01/14/2014  06:44 PM    <DIR>          ..
11/16/2012  03:18 PM            10,846 eMPL-client.py
01/14/2014  06:44 PM            71,651 euclid.py
01/14/2014  06:44 PM            78,960 euclid.pyc
11/16/2012  03:18 PM             5,921 ponycube.py
01/14/2014  06:15 PM             8,329 ponycube.pyc
               5 File(s)        175,707 bytes
               2 Dir(s)  10,995,863,552 bytes free

C:\Python27\eMPL_511>python eMPL-client.py 59
```

- 2 Windows will pop one. 1 contains a 3D Cube which corresponds with the quaternion angles outputted from the device. The other window will display any related logs or data



```
onsole.py", line 443, in <module>
ent.poll()

CJumplDebugConsole.py 18
>>> fetch calibration: loaded 121 bytes from calibdata
>>> error! calPacketHandler in state 0 got err None
I/w]+8B|Ç:umplFetchCalibration successfully got 121 bytes
I/w]+8B|Ç:InvenSense MPL v3.3.4 Apr 19 2011 17:00:01
V/MPL-acc:FSR: 2000
V/MPL-acc:FSR: 2000
V/MPL-acc:THS: 80
V/MPL-acc:THS: 40
V/MPL-acc:DUR: 1000
V/MPL-acc:DUR: 2540
V/MPL-ml:forcing disable of PROGRESSIVE_NO_MOTION bias tracker
V/MPL-ml:factory temperature compensation coefficients available - disabling ML_
LEARN_BIAS_FROM_TEMPERATURE
I/w]+8B|Ç:temp 64.1 gyros 4.46 0.06 0.37 bias 0.00 0.00 0.00
 temp_slope 0.00 0.00 0.00
I/w]+8B|Ç:Motion
I/w]+8B|Ç:temp 64.3 gyros 4.46 0.06 0.37 bias 0.00 0.00 0.00
 temp_slope 0.00 0.00 0.00
I/w]+8B|Ç:Motion
```

- Motion Driver 608 can accept input commands and display various different data. You must first make sure the cube window is the focused window then type in the input command.
  - ‘8’ : Toggles Accel Sensor
  - ‘9’ : Toggles Gyro Sensor
  - ‘a’ : Prints Accel Data
  - ‘g’ : Prints Gyro Data
  - ‘e’ : Prints Eular Data in radius
  - ‘r’ : Prints Rotational Matrix Data
  - ‘q’ : Prints Quaternions
  - ‘h’ : Prints Heading Data in degrees
  - ‘i’ : Prints Linear Acceleration data
  - ‘o’ : Prints Gravity Vector
  - ‘d’ : Register Dump
  - ‘6’ : Toggles Low Power 6-axis Mode, duty cycles the gyro while setting averaging filter to 16x
  - ‘p’ : Toggles Low Power Accel Mode at 20Hz sampling

- ‘t’ : run factory self test and calibration routine
- ‘1’ : Change sensor output data rate to 10Hz
- ‘2’ : Change sensor output data rate to 20Hz
- ‘3’ : Change sensor output data rate to 40Hz
- ‘4’ : Change sensor output data rate to 50Hz
- ‘5’ : Change sensor output data rate to 100Hz
- ‘m’ : Enter Low Power Interrupt Mode