

Motion Driver 6.12 – Getting Started Guide



Table of Contents

1	REVIS	SION HISTORY	3
2	PURP	OSE	4
3	RELE	ASE PACKAGE	4
4	STAR	TING WITH THE TI-MSP430 PROJECT	5
	4.1	REQUIREMENTS	5
	4.2	CONNECTING THE HARDWARE	5
	4.3	OPENING AND COMPILING THE MSP430 PROJECT	6
5	STM3	2F4 (CORTEX-M4) DISCOVERY BOARD PROJECT	8
	5.1	REQUIREMENTS	8
	5.2	CONNECTING THE HARDWARE	9
	5.3	OPENING AND COMPILING THE IAR PROJECT	10
6	РҮТН	ON CLIENT	12



1 Revision History

Revision Date	Revision	Description
06/27/2014	1.0	Initial Release
07/17/2014	1.1	Added STM32F4 information
05/05/2015	1.2	Updated for MD 6.12 release



2 Purpose

Motion Driver is an embedded software stack of the sensor driver layer that easily configures and leverages many of the features of the InvenSense motion tracking solutions. The motion devices supported are MPU6050/MPU6500/MPU9150/MPU9250. Many of the features of the hardware and the on board Digital Motion Processor (DMP) are encapsulated into modular APIs which can be used and referenced.

Motion Driver is designed as a solution which can be easily ported to most MCUs. With the release of the Motion Driver 6.0 it includes a 9-axis solution for ARM MCUs and the TI-MSP430. 6-axis only solutions should continue to reference the Motion Driver 5.1.2 for easier understanding of the software.

This document details how to set up the hardware and get the default projects up and running. It is recommended as a good way to understand the Motion Driver algorithms, DMP, and MPU HW features.

3 Release Package

MD6.0 release package contains example projects of the TI-MSP430 using Code Composer as well as STM32F4 and STM32L using IAR. It also contains binary MPL libraries for 9-axis fusion precompiled for ARM processors and TI-MSP430 processors. MPL libraries for arm uses gcc 4.7.2 compiler.

- ...\arm\STM32F4_MD6 : 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 .\STM3F4L_MD6\Projects\eMD6\EWARM\STM32F4_MD6.eww
- ...\documentation : All relevant documentations regarding MD6.12 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 MD6.0. ARM libraries are compiled using GCC 4.9.3 while the TI libraries are using Code Composer 5.5
- ...\msp430\eMD-6.0 : Contains the Code Composer project for MD6.12.



4 Starting with the TI-MSP430 Project

4.1 Requirements

- Code Composer Studio to compile MSP430 example
- TI-MSP430 JTAG for downloading and debugging
- Motion Driver 6.12 source files
- o InvenSense CA-SDK evaluation board (can be purchased through invensense.com)

4.2 Connecting the Hardware

TI-MSP430 JTAG – connect to PC with Code Composer Software and also to the CA-SDK.



InvenSense CA-SDK evaluation board

Micro-USB – Connect to PC for power and CA-SDK output



4.3 Opening and Compiling the MSP430 Project

- Select import under the file menu.
- Choose existing CCS eclipse project.
- Click the browse button to select the Motion Driver folder.

😯 CCS Edit - Code Composer Studio	😵 Import	import CCS Eclipse Projects
Hile Edit View Navigate Project Run Scripts New Alt+Sh Open File	Select Imports existing CCS Eclipse projects into workspace.	Select Existing CCS Eclipse Project Select a directory to search for existing CCS Eclipse projects.
Close C	Select an import source: type filter text	Select sgarch-directory: Browse Select archive file: BrowA
Switch Workspace Restart Step -1	Code Composer Studio Code Composer Studio	Discovered projects:
∠ Export		Mutamatically import referenced projects STEP 3
	(2) (Cancel

• Open the Motion Driver project by clicking finish.

💱 Import CCS Eclipse Projec	cts	_ 0 X	1	😜 CCS Edit - Code Composer	Studio			of the last live		٢
Select Existing CCS Eclipse Project				<u>File E</u> dit View <u>N</u> avigate	<u>P</u> roject <u>R</u> un	Scripts <u>W</u> indo	w F	<u>l</u> elp		
of the second second				📑 🖛 🔛 🕼	≪ -	恭• 🛷 •		📑 🏷 CCS De	bug	
Select a directory to search	for existing CCS Eclipse projects.			🌾 🔶 🕶 🔿 📼				CCS Edit		
		_		🏠 Project Explorer 🙁		🕒 🔄 🗸 🖓				
Select search-directory:	C:\Invensense_Karthik_Local\UMPL\Motion[B <u>r</u> owse		motion_driver-2.0 [A b M Includes	Active - Release]					
Select archive file		Pressure		🔺 🗁 core						
Select archive file:		Browse		a 🗁 driver						
Discovered projects:				a 🗁 eMPL						
				⊳ 🔝 dmpKe	y.h					
🔽 👕 motion_driver-2	2.0 [C:\Invensense_Karthik_Local\UMPL\Motio	Select All		⊳ 🔝 dmpma	ap.h					
				⊳ <u>ic</u> inv_gyn	o_dmp_android.c					
		Deselect All		⊳ <u>in</u> inv_gyn	o_amp_anaroia.n					
				b inv_gyn	0.c					
		R <u>e</u> fresh		4 🕞 msn430	0.11		=			
				▷ (= F5xx F6)	xx Core Lib					
•	4 111			🛛 🕞 USB_AP	ч – – Ч				c 🗠 », 😐	Ē
_				> 🗁 USB_eN	IPL				3 ~~ 2	_
Copy projects into work	space			⊳ 💽 msp430	_clock.c			No consoles to disp		Ĭ
Automatically import re	eferenced projects			⊳ 脑 msp430	_clock.h				🔗 🔳 🐓 🔻	
				⊳ 💽 msp430	_i2c.c				1	
				⊳ li i msp430	_i2c.h				No search resu	Ite
				⊳ 💽 msp430	_interrupt.c				available. Start	a
Open the Resource Explorer	and browse available example projects			⊳ <u>iii</u> msp430	_interrupt.h				search from th	e
				⊳ [<i>i</i> ∂ msp430	USB.cmd				search dialog	
				A B simple_apps						
				msp430	md		-			
				 Image: Image: Ima		•	1			
(?)	Back Next > Finish	Cancel		D* Licensed	😅 mot	ion_driver-2.0			L	



• Select the 'Project' pull-down menu and 'Build All' to compile the project



 With the JTAG and CA-SDK hardware connected download the firmware by selecting the 'Run' pull down menu and 'Debug'

😯 CCS Edit - eMD-6.0/simple_apps/msp430/mlite_test.c - Code Composer Studio						
<u>File Edit View Navigate Project Run</u> Scripts	s <u>W</u> indow <u>H</u> elp					
📑 🕶 🗟 🔦 🕶 🏇 🕶 🚨 Load	•	😭 🎭 CCS Debug 🕞 🔭				
Project Explorer 🛛 📄 🦄 Debug	F11					
<pre> Comparison of the second se</pre>	Histony , er_tap_cb(tap_cb); er_android_orient_cb(androi Configurations 943 Known Bug - 944 DNP when enabled will sample senso 945 specified in the dmp_set_fifo_rate 946 * a sample has been put into the FIF 947 there will be a 25Hz interrupt fro 948 * 949 There is a known issue in which if 950 then the interrupts will be at 260 951 * is set at a different rate. To avo 952 * 953 DMP sensor fusion works only with 954 */ 955 hal.dmp_features = DMP_FEATURE_6X_LP_ 956 Console S	d_orient_cb); or data at 200Hz and output to FIFO at the right API. The DMP will then sent an interrupt or 0. Therefore if the dmp_set_fifo_rate is at m the MPU device. Tyou do not enable DMP_FEATURE_TAP Hz even if fifo_rate tid this issue include the DMP_FEATURE_TAP EVEN at +-2000dps and accel +-26 QUAT DMP_FEATURE_TAP FEATURE SEMD DAW ACCEL DMD_EEATURE_CEMD () Cerrors, 6 warnings, 0 others Description Marrings (6 items)				
☐ [♦] Licensed [∰] eMD-6.0						



• You can then run the firmware through the debugger or turn off and on the CA-SDK to run off flash.

5 STM32F4 (Cortex-M4) Discovery Board Project

5.1 Requirements

- IAR ARM Workbench Compiler
- STM32F4-Discovery Board Evaluation Board (purchasable through DigiKey, Mouser, etc...)



- Motion Driver 6.12 source files
- InvenSense evaluation boards for MPU6050 or MPU6500 or MPU9150 or MPU9250





5.2 Connecting the Hardware



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 all similar





EVB Header Pin Number	Description	Discovery Board GPIO Pin Number
3	INT output	PA1
13	GND	GND
19	VCC_IN	EXT_3V
16	I ² C SCL	PB10
18	I ² C SDA	PB11

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

• Discovery Board UART Output

The MD6.12 outputs via data via its UART1 pins. 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.

5.3 Opening and Compiling the IAR Project

• Double click on the IAR ARM project file to automatically open the workspace in IAR ARM compiler. Project file is under the directory

../STM32F4_MD6\Projects\eMD6\EWARMSTM32F_MD6.eww



• Select the 'Project' pull down menu and 'Rebuild All'



• With the hardware connected, hit the 'Download and Debug' ICON





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: <u>https://www.python.org/downloads/</u>

Pyserial: <u>https://pypi.python.org/pypi/pyserial</u> 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



- 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>

C:\Windows\system32\cmd.exe	×
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.pyc 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</dir></dir>	
C:\Python27\eMPL_511>python eMPL-client.py 59	Ŧ
J <	•



• 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



• Motion Driver 6.12 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.

For TI-MSP430 you must first type 'inv' then the command.

For IAR project you only need to type the command only.

Commands 'l', 's', and 'x' are for MSP430 only.

- '8' : Toggles Accel Sensor
- \circ '9' : Toggles Gyro Sensor
- o '0' : Toggles Compass Sensor
- 'a' : Prints Accel Data
- o 'g' : Prints Gyro Data
- 'c' : Prints Compass Data
- 'e' : Prints Eular Data in radius
- o 'r': Prints Rotational Matrix Data
- 'q' : Prints Quaternions



- 'h' : Prints Heading Data in degrees
- 'i' : Prints Linear Acceleration data
- o 'o' : Prints Gravity Vector data
- 'w': Get compass accuracy and status
- o 'd' : Register Dump
- 'p': Turn on Low Power Accel Mode at 20Hz sampling
- o 'I': Load calibration data from flash memory
- o 's': Save calibration data to flash memory
- o '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
- o '4': Change sensor output data rate to 50Hz
- o '5' : Change sensor output data rate to 100Hz
- ',' : set interrupts to DMP gestures only
- '.' : set interrupts to DMP data ready
- o '6' : Print Pedometer data
- o '7': Reset Pedometer data
- o 'f': Toggle DMP on/off
- 'm' : Enter Low Power Interrupt Mode
- 'x' : Reset the MSP430
- 'v' : Toggle DMP Low Power Quaternion Generation