

FSP201 Development Kit

Quick Start Guide 1000-5132

Rev. 1.1 December 2023

CONFIDENTIAL AND PROPRIETARY UNAUTHORIZED REPRODUCTION PROHIBITED



Documentation Control

History Table

Version	Date	Description
1.0	December 6, 2023	Initial Release
1.1	December 12, 2023	Broken reference fixes, odd page footer update



Disclaimer and Proprietary Information Notice

The information contained in this document does not represent a commitment on any part by CEVA®, Inc., or its subsidiaries (collectively, "CEVA"). CEVA makes no warranty of any kind with regard to this material, including, but not limited to implied warranties of merchantability and fitness for a particular purpose whether arising out of law, custom, conduct, or otherwise.

Additionally, CEVA assumes no responsibility for any errors or omissions contained herein, and assumes no liability for special, direct, indirect, or consequential damage, losses, costs, charges, claims, demands, fees, or expenses, of any nature or kind, which are incurred in connection with the furnishing, performance, or use of this material.

This document contains proprietary information, which is protected by U.S. and international copyright laws. All rights reserved. No part of this document may be reproduced, photocopied, or translated into another language without the prior written consent of CEVA. All product names are registered trademarks of CEVA®, Inc. and/or its subsidiaries, or, of its applicable suppliers if so stated.



Support

CEVA® makes great efforts to provide a user-friendly software and hardware development environment. Along with this, CEVA provides comprehensive documentation, enabling users to learn and develop applications on their own. Due to the complexities involved in the development of DSP applications that might be beyond the scope of the documentation, an online Technical Support Service has been established. This service includes useful tips and provides fast and efficient help, assisting users to quickly resolve development problems.

How to Get Technical Support:

- **FAQs**: Visit our website <u>http://www.ceva-dsp.com</u> or your company's protected page on the CEVA website for the latest answers to frequently asked questions.
- **Application Notes**: Visit our website <u>http://www.ceva-dsp.com</u> or your company's protected page on the CEVA website for the latest application notes.
- **Email**: Use the CEVA central support email address <u>ceva-</u> <u>support@ceva-dsp.com</u>. Your email will be forwarded automatically to the relevant support engineers and tools developers who will provide you with the most professional support to help you resolve any problem.
- License Keys: Refer any license key requests or problems to <u>sdtkeys@ceva-dsp.com</u>. For SDT license keys installation information, see the *SDT Installation and Licensing Scheme Guide*.

Email: <u>ceva-support@ceva-dsp.com</u>

Visit us at: <u>www.ceva-dsp.com</u>



Table of Contents

1.	INT	RODUCTION
	1.1	Scope
	1.2	Audience8
	1.3	Related Documents8
	1.4	Overview
2.	HAF	RDWARE
	2.1	Introduction9
	2.2	Connections 10 2.2.1 Nucleo 10 2.2.2 Host Interface (J3) 11
	2.3	Host Protocol Configuration11
3.	SOF	TWARE
	3.1	FSP201 Development Kit12
	3.2	PC Demo Application with Dev Kit 12 3.2.1 Requirement 12 3.2.2 Running PC Application 13 3.2.3 Backup Bridge Image 19
	3.3	Development Environment19
	3.4	Embedded Example Software20
4.	SIM	IPLE CALIBRATION
	4.1	Hardware Controlled Method25
	4.2	Software Controlled Method25
5.	GLC	26 SSARY

List of Figures

Figure 2-1: FSP201 Development Kit	9
Figure 3-1: Simplified System Diagram with Freespace [™] MotionStudio	12
Figure 3-2: Device Manager to check installed driver for ST Virtual COM Port	13
Figure 3-3: Freespace [™] MotionStudio 2 Startup Window	13
Figure 3-4: Device Connection Panel	14
Figure 3-5: Device Connection Window after Successful Communication in Freespace [™] MotionStudio2	15
Figure 3-6: Sensor Control Panel in Freespace [™] MotionStudio 2	16
Figure 3-7: Virtual Object Panel in Freespace [™] MotionStudio 2	17
Figure 3-8: FSP201 Device Orientation	17
Figure 3-10: Sensor Orientation in Virtual Object Panel in Freespace [™] MotionStudio 2	18
Figure 3-11: Installed driver for ST Virtual COM port	19
Figure 3-12: Simplified System Diagram (blue indicates driver developed by CEVA)	20
Figure 3-13: Source Code Structure	20



Figure 3-14: Terminal Emulator Screenshot	22
Figure 3-15: IAR EWARM Screenshot	23
Figure 3-16: Simple Calibration Procedure Start Position	24
Figure 3-17:Simple Calibration Procedure End Position	24

List of Tables

Table 2-1: FSP201 Shield to Nucleo Interface	10
Table 2-2: Host Interface (J3) pinout	11
Table 2-3: Host Protocol Selection	11
Table 3-1: Acronyms	26



1. Introduction

1.1 Scope

This document describes the features and operation of the FSP201 Development Kit.

1.2 Audience

This document is intended for software and hardware engineers integrating or evaluating the FSP201.

1.3 Related Documents

The following documents are related to the information in this document:

- 1. 1000-4819 FSP201 Datasheet, CEVA, Inc.
- 2. 1000-3625 SH-2 Reference Manual, CEVA, Inc.
- 3. 1000-4906 FSP201 Simple Calibration User Guide, CEVA, Inc.
- 4. 1000-3535 Sensor Hub Transport Protocol
- 5. 1000-5106 Schematic, Module, Dev Kit Shield
- 6. 1000-4868 Schematic, FSP201 Module

1.4 Overview

The FSP201 is a 6-axis IMU processor, integrating CEVA's highperformance sensor hub software stack, to provide heading and orientation outputs. When connected to one of several supported sensors, it performs all the accelerometer and gyroscope sensor fusion processing necessary to produce stable and accurate heading and orientation outputs. This document is intended to provide information about the FSP201 Development Kit, the pin connections and software provided by CEVA to facilitate customer integration.



9

2. Hardware

2.1 Introduction

The FSP201 Development Kit includes a shield known as the FSP201 ("target device") Development Board that is designed for quick and easy development and prototyping. The shield is designed to connect to the STM32F411RE Nucleo ("bridge board") which is developed and sold by STMicroelectronics (ST) and is included in the FSP201 Development Kit package. The Nucleo platform includes a STM32F411 MCU that serves as the master to the FSP201. This document includes details for connecting and using the FSP201 Development Kit with the ST Nucleo prototyping platform.

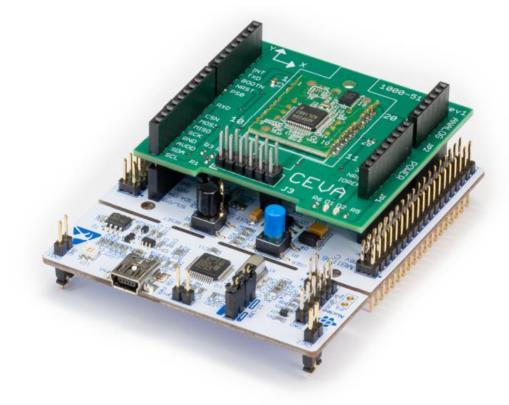


Figure 2-1: FSP201 Development Kit



2.2 Connections

2.2.1 Nucleo

The FSP201 Development Shield simply plugs into the Nucleo board and is ready to use. The reader is encouraged to review the FSP201 Datasheet [1] and the FSP201 Shield Schematic [5] for more information on the pinout configurations. The FSP201 communicates with a host system over the following possible interfaces: I2C-SHTP, UART-RVC, or UART-SHTP.

Dev Board	Signal	Nucleo
JP1.1	NC	CN6.1
JP1.2	VDD	CN6.2
JP1.3	NC	CN6.3
JP1.4	NC	CN6.4
JP1.5	NC	CN6.5
JP1.6	GND	CN6.6
JP1.7	GND	CN6.7
JP1.8	NC	CN6.8

Dev Board	Signal	Nucleo
JP2.1	RED LED	CN8.1
JP2.2	GRN LED	CN8.2
JP2.3	NC	CN8.3
JP2.4	PS1	CN8.4
JP2.5	NC	CN8.5
JP2.6	NC	CN8.6

Dev Board	Signal	Nucleo
JP3.10	H_SCL	CN5.10
JP3.9	H_SDA	CN5.9
JP3.8	NC	CN5.8
JP3.7	GND	CN5.7
JP3.6	NC	CN5.6
JP3.5	NC	CN5.5
JP3.4	H_SA0	CN5.4
JP3.3	NC	CN5.3
JP3.2	NC	CN5.2
JP3.1	H_RX	CN5.1

Dev Board	Signal	Nucleo
JP4.8	NC	CN9.8
JP4.7	PS0	CN9.7
JP4.6	RESETN	CN9.6
JP4.5	BOOTN	CN9.5
JP4.4	H_TX	CN9.4
JP4.3	H_INT	CN9.3
JP4.2	NC	CN9.2
JP4.1	NC	CN9.1

Table 2-1: FSP201 Shield to Nucleo Interface

In UART-SHTP and I2C-SHTP modes, the FSP201 uses the Sensor Hub Transport Protocol (SHTP) to communicate with a system or application processor (host that connects to the FSP201). The SHTP protocol is documented in the Sensor Hub Transport Protocol [4], allowing a customer to potentially develop their own host software if they choose to do so. To ease customer integration, CEVA has developed software that runs on a host platform such as the STM32F411RE Nucleo series. The software driver fully implements the communication protocol used by the FSP201. CEVA provides this software driver package as source code. The FSP201 Development Kit has programmed the ST Nucleo to work with the CEVA PC application to demonstrate functions. Customers who intend to use the FSP201 Development Kit for their own software development should use the driver package to download new firmware.

In UART-RVC mode, the FSP201 transmits heading and sensor information at 100Hz over the H_TX pin. UART-RVC mode has its own dedicated packet format. It's not configurable, but convenient to get a limited sensor data set for applications.





2.2.2 Host Interface (J3)

Dev Board	Signal	Description
J3.1	SA0	I2C address select
J3.2	SDA_TX	I2C data / UART TX data
J3.3	SCL_RX	I2C clock / UART RX data
J3.4	NA	Not applicable
J3.5	INT	Host interrupt
J3.6	PS1	Host interface protocol select 1
J3.7	NRST	Reset, Active low
J3.8	NC	Not connected
J3.9	VDD	Power supply
J3.10	GND	Ground
J3.11	PS0	Host interface protocol select 0
J3.12	BOOTN	Bootloader select

The Host Interface connector (J3) is available for testing purposes, or this interface could be used to wire up to another host.

Table 2-2	Host In	nterface	(J 3)	pinout
-----------	---------	----------	---------------	--------

2.3 Host Protocol Configuration

The communication interface is configured by setting the protocol selection PS0 and PS1 pins appropriately.

PS1	PS0	FSP201 Transport Protocol
0	0	I2C-SHTP
0	1	UART-RVC
1	0	UART-SHTP
1	1	Not supported

Table 2-3: Host Protocol Selection



3. Software

3.1 FSP201 Development Kit

FSP201 Development Kit contains a pre-programmed STM32 Nucleo board with CEVA software that allows communication between the FSP201 and Freespace[™] MotionStudio 2.

Freespace[™] MotionStudio 2 is a Windows application to allow users to control and configure the FSP201 through a USB interface. FSP201 Development Kit can be used for a quick evaluation of the FSP201. A generalized system diagram is shown below.

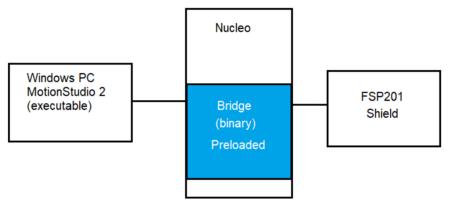


Figure 3-1: Simplified System Diagram with FreespaceTM MotionStudio

3.2 PC Demo Application with Dev Kit

3.2.1 Requirement

Running FreespaceTM MotionStudio 2 with FSP201 development kit requires the following items.

- ST-LINK/V2 USB driver available from the ST website (<u>http://www.st.com/en/embedded-software/stsw-link009.html</u>).
- ST32 Virtual COM Port Driver from ST website (<u>http://www.st.com/en/development-tools/stsw-stm32102.html</u>). The FSP201 software package is tested with STSW version 1.4.0. Once you have downloaded and extracted the driver, follow the readme.txt file for the instructions to complete the installation.
- FreespaceTM MotionStudio 2 application from https://www.ceva-dsp.com/resource/freespace-motionstudio-2-2-0-1/



Connect USB Type A to Mini-B cable to Nucleo board and your PC. The virtual COM port should appear in your Device Manager.

~	ē	Ports (COM & LPT)
	-	STMicroelectronics STLink Virtual COM Port (COM10)

Figure 3-2: Device Manager to check installed driver for ST Virtual COM Port

Start FreespaceTM MotionStudio 2 (MotionStudio2.exe) after FSP201 development kit virtual COM port is successfully detected in your PC.

3.2.2 Running PC Application

3.2.2.1 Start Freespace[™] MotionStudio 2

After you unzip the PC Application package, launch MotionStudio2.exe under MotionStudio2 folder. This will open MotionStudio2 window.

crin CEVA's Hillcrest Labs MotionStudio 2		-	\times
File Help Motionsbudio? - - Device Connection - - Sensor Control - - Sensor Data Analysis - - Sensor Data Logging - - Virtual Object - - General Control -	EXERCISES Hitcrest Labs Motion Studio 2		
Disconnected	ne-Click Connect BNO08X SHTP FSP220 SHTP BNO08X RVC FS9200 RVC		

Figure 3-3: FreespaceTM MotionStudio 2 Startup Window

3.2.2.2 Establish Connection to the Nucleo Board

From the menu panel on the left, select Device Connection. This panel allows users to select device type, transport protocol and more.

• *Product* set to FSP201



- Bridge Device Type set to ST Nucleo
- *Bridge Virtual COM Port* select the available ST Link Virtual COM port
- *Target Device* set to SHTP over UART
 - SHTP over I2C and UART RVC are other supported target devices
- Use the "*Connect*" button to establish the connection.

CEVA's Hillcrest Labs MotionStur	tio 2			-		Х
Eile Help MotionStudio 2 Device Connection	Device Connection ?	Bridge::NC	Target::NC	Panel::A	ctive	
	Product	Bridge=NC	Target. NC	Panet:A	ctive	
Disconnected	Console					~

Figure 3-4: Device Connection Panel



dio 2						- 🗆 ×
Device Connection ?				Bridge::Nucleo	Target::SensorHub	Panel::Active
Product FSP201 ~]					
Bridge Device Type ST Nucleo V	Virtual COM Port	s STLink Virtual COM Port (COM36				
Target Device Transport Protocol SHTP over UART V	Baud Rate	-				
Connect	Disconnect					
((Tag): MaxCargoPlusHeaderV) 256), Container((Tag): MaxTra Container((Tag): ChannelNam 'NormalChannel', 'Length': 1, 'Value': sensorhub'), Contain 'Value': Sensorhub'), Container((Tag): C 'inputWake'), Container((Tag): b'1.1.0x00'), Container((Tag): 'x00x0b'x00x0x0x0Chanter((Tag):	Vrite', 'Length': 2, 'Value': 256) InsferRead', 'Length': 2, 'Value (e', 'Length': 8, 'Value': 'contro Value': 1)), Container(["Tag': 'C er(["Tag': NormalChannel', 'Length': 129, 'Length': 12, 'Value': b', ': 'NormalChannel', 'Length': 12, Vo03v0ftx00vt10x00vt11x000)), Container(["Tag:: MaxCargoPiu e: 32767)), Container(["Tag:: AppN of]), Container(["Tag:: GUID,: Leng hannelName', Length: 7, Value: nugth: 1, Value: 2)), Container([[Ti alue:: inputNorma1]), Container([Ti , Value: 3), Container([Tag:: Ch xd8xt0xd5xxd9xd3x10xd1xx10xd 12x00xd13x00xd1xx10xd15x10x	HeaderRead, Lengthi 2, Vai ame, 'Lengthi 5, Value: 'SHT thi 4, Value: 1)), Container({ 'device'}), Container({Tag: 6 ag: 'ChannelName', 'Lengthi Tag: 'WakeChannel', 'Lengthi annelName', 'Lengthi 12, Value b\x05\xfa\x05\xfc\x11\xef\x02\x 16\xx05\xfa\x05\xfc\x11\xef\x02\x 16\xx00\x17\xx00\x18\x00\x19\x0	ue': 32767}), Container(('P'), Container(('Tag': N Tag': 'AppName', 'Leng' UID', 'Length: 4, 'Value' 8, 'Value': control')), Co : 1, 'Value': 4)), Contain ue': 'InputGyroRy')), Co vol'In/vx02/n/x03/vx04/vx04/ 0/x14/x00/x16/vx00/vx16/v	[Tag]: 'MaxTransferWrite', lormalChannel', 'Length': th': 11, 'Value': 'executable : 2)], Container[[Tag': Api intainer([Tag': NormalCha er([Tag': ChannelName', ntainer((Tag': 128, 'Length nhw05/x00/x06/n/x07/x10)	'Length': 2, 'Value': 1, 'Value': 0)), e')), Container({'Tag': pName', 'Length': 10, innel', 'Length': 1, Length': 10, 'Value': ': 6, 'Value': ': 6, 'Value':
INFO: Device Connection / Con INFO: Device Connection / Con INFO: Device Connection / Con INFO: Device Connection / Con INFO: Device Connection / Con	nect Device : SHTP Control Ch nect Device : Executable Devic nect Device : SH-2 Control Ch nect Device : SH-2 Normal Inp nect Device : SH-2 Wake Inpu	hannel Success ce Channel Success Jannel Success put Channel Success it Channel Success				
	Product PSP201 Bridge Device Type ST Nucleo Target Device Transport Protocol SHTP over UART Connect (Device, ShtpChrl/Receive ; Co (Tras: ShtpChrl/Receive ; Co (Tras: ShtpChrl/Receive ; Co (Tras: MaxCargoPlusHeader); Container(Tag: MaxTargoPlusHeader); Container(Tag); VaOVabu000x00x00v00v00v0v0v0v0v0v0v0v0v0v0v0v0	Device Connection Product FS201 Bridge Device Type Virtual COM Port ST Nucleo COM36 STMicroelectronic Target Device Transport Protocol Budge Connect Disconnect (Device_ShtpCtri/Receive_Container[[Report]ype: 'Advert [Transport Protocol Connect Disconnect (Connect) Disconnect (Device_ShtpCtri/Receive_Container[[Report]ype: 'Advert [Transport Protocol Container[Trag: 'MaxTansferRead', Length: 2, Vulue: 'Soft Value: 30, Container[Trag: 'MaxTansferRead', Length: 12, Vulue: 'Soft Disconnect (Device_ShtpCtri/Receive_Container[Trag: 'MormalChanner, 'Length: 12, Vulue: 'Soft Disconnect Unavoidswoodswoodsvoodsvoodsvoodsvoodsvoodsvoo	Device Connection Image: Connection Product FSP201 Bridge Device Type Uritual COM Port ST Nucleo Stridge COM36 STMicroelectronics STUnk Virtual COM Port (COM36 Target Device Commed Transport Protocol Baud Rate SHTP over UART 3000000 Connect Disconnect [Device_ShtpCtrl/Receive ', Container[[Report]ype: 'Advertise', Response: [Container[[Tag: MaxCargoPlusHeaderWirle', Length: 2, Value: 256], Container[[Tag: AppleContainer[[Tag: Container[[Tag: Containe	Device Connection Image: Connection Product Image: Connection Bridge Image: Connection Device Type Virtual COM Port ST Nucleo COM36 STMicroelectronics STLink Virtual COM Port (COM36) Target Device Image: Connect Image: Connect Disconnect Connect Disconnect ([Device_ShtpCtrl/Receive_; Container[[Report]ype: 'Advertise', Response'; [Container[[Tag: 'GUID', Length: 4, Value: 3D, Container[Tag: 'GUID', Length: 12, Value: 3D, Container[Tag: 'GUID', Value: 3D, Container[Tag: 'GUID', Length: 12, Value: 3D, Container[Tag: 'GUID', Value: 3D, Value:	Device Connection Image: Nucleo Product Image: Nucleo Bridge Image: Nucleo Device Type Virtual COM Port ST Nucleo COM36 STMicroelectronics STLink Virtual COM Port (COM38) Target Device Image: Nucleo Target Device Image: Nucleo Connect Disconnect (Connect) Disconnect (Device_ShtpCtrl/Receive_; Container[Tag: Autor: Status: Stat	Device Connection ? Bridge: Auccieo Target: SensorHub Product

Figure 3-5: Device Connection Window after Successful Communication in FreespaceTM MotionStudio2

When connection process is completed, the three status indicator text boxes on the upper right corner of the panel and the console window on the bottom provide the result of connection process. The three status indicators show the status of the connected system and the status of the associated panel. If the specific panel supports the protocol used by the connected device, the panel becomes active and shows in green color.

3.2.2.3 Sensor Control

The Sensor Control panel allows the user to enable and disable the various sensors individually. There are two ways to control sensors:

- To enable an individual sensor at a default operation rate, use the check box on the right end of the row for each sensor.
- To enable sensors at specific rates, input the requested operating period, in microseconds, in the "Requested Period (us)" fields. Then click the "Set Sensor Periods" button on the top of the panel. All sensors will be updated with the specified operating period. The "Requested Period (us)" fields which are left blank or have invalid values are assumed to be "zero".



In many cases, the sensors do not operate at the exact rate as requested. The actual operating period is shown in the "Reported Period (us)" field. Users can also use the "Get Sensor Periods" button on top of the panel to refresh the actual operating period for all sensors.

Device Connection	Sensor Control ?							Bridge::Nucleo	Target	SensorHub	Panel::Active
Device Information Sensor Control Sensor Data Visualization Sensor Data Analysis	Sensor Controls Set Sensor Periods Get Sensor F		sable All Senso		r Data Fields				0	ting Unit Interva	al in us
Sensor Data Logging	Set Sensor Periods Get Sensor P	Periods Dis	sable All Senso	rs Clear	Data Fields				Operat	ing onic interva	ii iii us
·· Virtual Object ·· RVC Utilities	Motion Request							Motion Intent			
General Control		REQ Intl/Freq	RPT Intl/Freq	Intent	Request	Status					
	MotionRequest	0	0					0 - Unknown		~	Set
	Special Sensors										
		REQ Intl/Freq	RPT Intl/Freq	Mode	Angle						
	OperatingMode] 🗆					
		REQ Intl/Freq	RPT Intl/Freq	1(X)	J (Y)	K (Z)	Real (W)	х ү	z		
	GyroRotationVector	0	0								j –
		REQ Intl/Freq	RPT Intl/Freq	X	Y	Status					
	StandardCursor]				Sh	ow Cursor			
	Rotation Vectors										
		REQ Intl/Freq	RPT Intl/Freq	I I (X)	J (Y)	K (Z)	Real (W) Accuracy	Status		
	ARVRStabilizedGameRotationVector	0	0							Show Euler	Angles
	ARVRStabilizedRotationVector	0	0							Show Euler	Angles
	GameRotationVector	10000	9615	-0.0023	0.7071	0.0085	0.707	70 0.0000	3 🗟	Show Euler	Angles
	GeomagneticRotationVector	0	0				1			Show Euler	Angles
	RotationVector	0	0				í		[Show Euler	Angles
		L		1	JL	JLJ	J				-
	INFO: Sensor Control / Set ARVRStabiliz INFO: Sensor Control / Set ARVRStabiliz INFO: Sensor Control / Set GyroRotation INFO: Sensor Control / Set MotionReau	edRotationVec edGameRotation Vector interva	tor interval @ onVector inter I @ Ous								

Figure 3-6: Sensor Control Panel in Freespace TM MotionStudio 2

3.2.2.4 Virtual Object

The Virtual Object panel shows the orientation of the device. Please note that you need to enable sensors in Sensor Control panel, then select the sensor from the drop-down menu in Virtual Object panel. The sword in the Virtual Object will move according to the device orientation.

To adjust the camera position, move the cursor to the Virtual object Panel, then press the LEFT mouse button. Hold the button down and move the mouse to change the view position. To reset the camera position, use the "Reset Camera Position" button.

To display the game rotation vectors, select the game rotation vectors from the drop-down menu, the data fields should start updating with the received sensor data. The virtual object will move according to the orientation of the hardware. Use the Sensor Control Panel to enable or disable the specific sensor. This panel does not control the sensor but displays the output data.



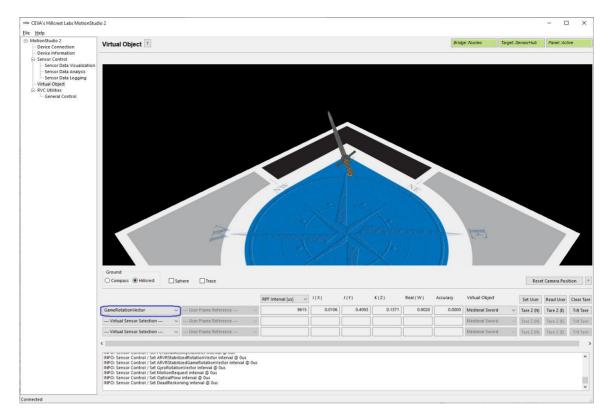


Figure 3-7: Virtual Object Panel in Freespace[™] MotionStudio 2

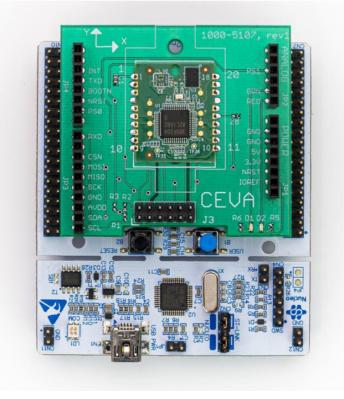


Figure 3-8: FSP201 Device Orientation



X/Y marked on the board, using the right-handed rule

Please follow the instructions below to align your device.

- Enable Game Rotation Vector and ARVR Stabilized Game Rotation Vector in "Sensor Control" panel.
- Switch to Virtual Object panel and move the background so the black corner of the logo on the ground plane points to your forward direction (heading).
- Hold the FSP201 development kit Y+ axis points to your forward direction as well.
- Select "Game Rotation Vector" in drop down menu and click "Tare Z". Now, the sword will point to the edge of the logo and is aligned with your device Y+.

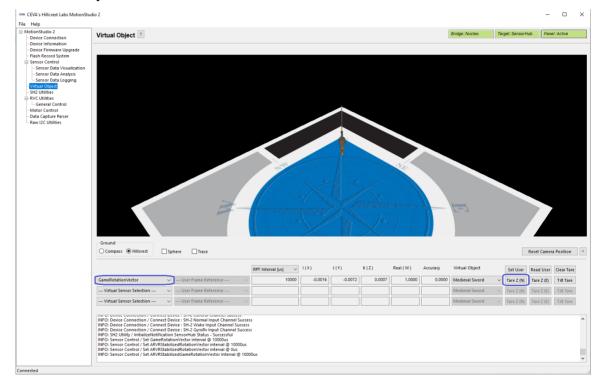


Figure 3-9: Sensor Orientation in Virtual Object Panel in Freespace TM MotionStudio 2



3.2.3 Backup Bridge Image

(1) Please keep in mind that FSP201 Development Kit is programmed to work with a Windows PC Application. Development example code explained in Section 3.3 is different and will overwrite the binary image shipped with the FSP201 Development Kit. If you want to switch back to the bridge code to run the Windows PC Application, please either download the latest bridge firmware from CEVA website (https://www.ceva-dsp.com/resource/bridgefirmware-for-nucleo/), or read the Nucleo device memory using STM32 ST-LINK Utility before downloading a new image into the Nucleo board. Please follow the instructions below to back up the prebuilt bridge image.

To save the contents of Nucleo bridge board:

- Open STM32 ST-LINK Utility
- Target->Connect
- Use "Save the displayed content in a Binary File" on the top left corner to save the image into a file for future use.

To program Nucleo bridge board with the saved image:

- Open STM32 ST-LINK Utility
- Target->Connect
- Target->Program
- In new window select "File path" to locate previously saved file.
- Click "Start"

3.3 Development Environment

The example software requires the following items to execute.

- IAR Embedded Workbench® for ARM (EWARM) by IAR Systems.
- ST-LINK/V2 USB driver. This driver is available from the ST website and is supported by the IAR Embedded Workbench for ARM (EWARM). After installing EWARM, check IAR_INSTALL_DIRECTORY\arm\drivers\ST-Link\ Please skip this step if you have installed already from Section 3.2.1.
- ST32 Virtual COM Port Driver from the ST website. Please skip this if you have installed already from Section 3.2.1.

```
    Ports (COM & LPT)
    STMicroelectronics STLink Virtual COM Port (COM10)
```

Figure 3-10: Installed driver for ST Virtual COM port



• Terminal emulator software like Tera Term or PuTTy. Set up the terminal emulator at 115200 – 8bit – no parity – 1bit stop bit – no flow control.

3.4 Embedded Example Software

CEVA provides a complete software package for the STM32F411RE Nucleo boards. The example application for FSP201 development kit source code is available in public github.

https://github.com/ceva-dsp/sh2-demo-nucleo

Clone this repository using the --recursive flag with git. An example git clone command is shown below.

git clone --recursive https://github.com/ceva-dsp/sh2-demo-nucleo.git

Everything required to obtain outputs from the FSP201 is included in this package. The software package incorporates the FSP201 driver, enabling SH2 functionality for the development system.

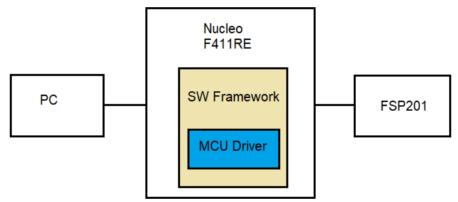


Figure 3-11: Simplified System Diagram (blue indicates driver developed by CEVA)

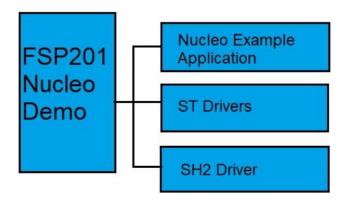


Figure 3-12: Source Code Structure



The software is organized as an IAR EWARM project that can be dropped into the IAR IDE (version 9.32 and higher) on a Windows PC. Follow this procedure to compile the project and download the software to the Nucleo board.

- Open IAR Embedded Workbench for ARM (EWARM)
- In the File menu, select Open and choose Workspace. Browse to where the example package is extracted and select "sh2-demo-nucleo/EWARM/Project.eww". This should open an IAR workspace with all the files within the project.
- In the project configuration, select one of the following for the FSP201.
 - o fsp20x-uart: FSP201 in UART-SHTP mode.
 - o fsp201-i2c: FSP201 in I2C-SHTP mode.
 - fsp201-uart-cal: simple calibration demo for the FSP201 in UART-SHTP mode.
 - fsp201-i2c-cal: simple calibration demo for the FSP201 in I2C-SHTP mode.
 - o demo-rvc: FSP201 in UART-RVC mode.
- In the "Project" menu, select "Rebuild All" to compile the project.
- After the project is successfully compiled, go to the Project menu and select Download and Debug.

The "sh2" directory contains a full implementation of the CEVA communications protocol for the FSP201.

The reader is encouraged to review the FSP201 Datasheet [1] and the SH-2 Reference Manual [2] for details on how to construct messages.

The output from the FSP201 is printed through the serial port. The first few lines indicate that the host has established proper communication ("Product ID Request") with the FSP201 and the FSP201 has responded with version information ("Product ID Response").

• fsp201-i2c:

Game Rotation vector is enabled at 100Hz by default and reports are printed through the serial port.



SOM26:115200baud - Tera Term VT	-	\times
Eile Edit Setup Control Window Help		
CEVA SH2 Sensor Hub Demo.		^
CEUM SH2 Sensor Hub Demo. Part 10004818 : Uersion 3.12.25 Build 36 Part 10004817 : Uersion 5.7.24 Build 26 Part 10004815 : Uersion 5.3.26 Build 28 Part 10004815 : Uersion 3.12.18 Build 18 1.8546 CRU : r80.999939 : i0.807996 j:-0.807996 k:-0.808061 1.8647 CRU : r80.999939 : i0.807996 j:-0.807996 k:-0.808061 1.8647 CRU : r80.999939 : i0.807996 j:-0.807996 k:-0.808061 1.8649 CRU : r80.999939 : i0.807996 j:-0.807996 k:-0.808061 1.8649 CRU : r80.999939 : i0.807996 j:-0.807996 k:-0.808061 1.8748 CRU : r80.999939 : i0.807996 j:-0.807996 k:-0.808061 1.8748 CRU : r80.999939 : i0.807996 j:-0.807996 k:-0.808061 1.8748 CRU : r80.999939 : i0.807996 j:-0.807996 k:-0.808061 1.8758 CRU : r80.999939 : i0.807996 j:-0.807996 k:-0.8080122 1.1851 CRU : r80.999939 : i0.807996 j:-0.807996 k:-0.8080122 1.1254 CRU : r80.999939 : i0.807996 j:-0.807996 k:-0.8080122 1.1254 CRU : r80.999939 : i0.807996 j:-0.807996 k:-0.8080122 1.1255 CRU : r80.999939 : i0.807996 j:-0.807996 k:-0.8080122 1.1355 CRU : r80.999939 : i0.807996 j:-0.807996 k:-0.8080122 1.1557 CRU : r80.999393 : i0.807996 j:-0.807996 k:-0.8080122 1.1557 CRU : r80.999339 : i0.807996 j:-0.807996 k:-0.8080122 1.1558 CRU : r80.999339 : i0.807996 j:-0.807996 k:-0.8080122 1.2264 CRU : r80.999339 : i0.807996 j:-0.8087996 k:-0.8080122 1.2265 CRU : r80.999339 : i0.807996 j:-0.80857 k:-0.808057 k:-0.8080122 1.2265 CRU : r80.999339 : i0.807996 j:-0.808857 k:-0.80		^
1.2770 GRU: ::0.999939 1:0.007996 j:-0.008057 k:-0.000122 1.2871 GRU: ::0.999939 1:0.007996 j:-0.008057 k:-0.000122		
1.2972 GRU: r:0.999939 i:0.007996 j:-0.008057 k:-0.000122 1.3072 GRU: r:0.999939 i:0.007996 j:-0.008057 k:-0.000122 1.3174 GRU: r:0.999939 i:0.007996 j:-0.008057 k:-0.000122 1.3275 GRU: r:0.999939 i:0.007996 j:-0.008057 k:-0.000122		
1.3376 GRU: r:0.999939 1:0.007936 j-0.008057 k:-0.000122 1.3477 GRU: r:0.999939 1:0.007936 j-0.008057 k:-0.000122 1.3477 GRU: r:0.999939 1:0.007935 j-0.008057 k:-0.000122 1.3579 GRU: r:0.999939 1:0.007935 j-0.008057 k:-0.000122 1.3579 GRU: r:0.999939 1:0.007935 j-0.008057 k:-0.000122 1.3579 GRU: r:0.999939 1:0.007935 j-0.008057 k:-0.000122		

Figure 3-13: Terminal Emulator Screenshot



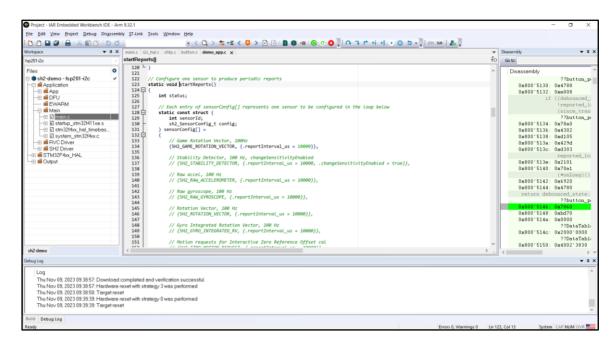


Figure 3-14: IAR EWARM Screenshot

• fsp201-i2c-cal:

Simple Calibration mode (software-controlled method) is entered. The user should follow the instructions on the serial window to perform the calibration. A precise 180 degree rotation is required between the start orientation and the final orientation. After 'Calibration completed successfully', it will run into the next cycle of calibration.

Press key Enter to start simple calibration:

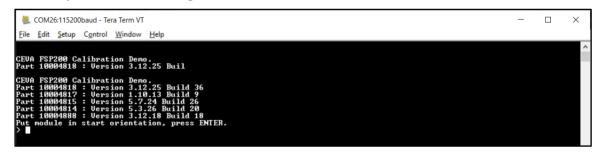






Figure 3-15: Simple Calibration Procedure Start Position

Rotate 180 degrees and press Enter to get calibration results:

🜉 COM26:115200baud - Tera Term VT	_	×
<u>F</u> ile <u>E</u> dit <u>S</u> etup C <u>o</u> ntrol <u>W</u> indow <u>H</u> elp		
CEUA FSP200 Calibration Demo. Part 10004818 : Version 3.12.25 Buil		
CEUA FSP200 Calibration Demo. Part 10004818 : Version 3.12.25 Build 36 Part 10004817 : Version 1.10.13 Build 9 Part 10004815 : Version 5.7.24 Build 26 Part 10004814 : Version 5.3.26 Build 20 Part 10004888 : Version 3.12.18 Build 18 Put module in start orientation, press ENTER.		
Solution of the second		



Figure 3-16:Simple Calibration Procedure End Position



4. Simple Calibration

4.1 Hardware Controlled Method

Not supported on FSP201 Dev Kit hardware. See the FSP201 Datasheet to implement the FSP201 Simple Calibration via hardware control.

4.2 Software Controlled Method

Software Controlled Simple Calibration only works in SHTP mode.

Configuration from the IAR Embedded Workbench "fsp201-i2c-cal" or "fsp201-uart-cal" shows how to execute the FSP201 Simple Calibration. SH-2 Reference Manual [2] Section 6.4.11 also shows the details of the commands used in the example application.

Enter Calibration Mode: startCalStart() shows command to initiate the Simple Calibration. Wait for the command response, startCalRx().

Rotate: Once a valid response is received, rotate the device 180° clockwise, then send finish calibration command, finishCalStart().

Wait for the Calibration Results: wait for the finish calibration response, finishCalRx() and check the calibration status.

When Finish Calibration Response returns any error code, rerun the Simple Calibration for a couple of more times. If the calibration still fails, the device is out of specification.



5. Glossary

Table 5-1 defines the acronyms used in this document.

Table 5-1: Acronyms

Term	Definition
SHTP	Sensor Hub Transport Protocol
MS2	MotionStudio2