1. Hardware

1.1. Introduction

The FSM30X is a compact IMU module based on CEVA's Hillcrest Labs business unit's BNO080 9-axis SiP. The FSM30X incorporates the BNO080, a 32.768 kHz crystal and passive components into a compact module form factor that can be quickly and easily integrated into a design. The FSM30X provides all the motion based outputs available on the BNO080. It does not support environmental sensors.



Figure 1: FSM30X

1.2. Connections with Nucleo

The FSM30X uses the Sensor Hub Transport Protocol (SHTP) to communicate with a system or application processor (host that connects to the FSM30X). The SHTP protocol is documented in the *BNO080 Datasheet* [1], allowing a customer to potentially develop their own host software if they choose to do so. In order to ease customer integration, Hillcrest has developed software that runs on a host platform such as the STM32F4x1RE Nucleo series. The software driver fully implements the communication protocol used by the FSM30X. Hillcrest provides this software driver package as source code.

Please refer to Hillcrest *FSM30X Datasheet* [4] and *FSM300 Connection with Nucleo* [5] for more information on the pinout and connection.



Figure 2: FSM30X Pinout

Customers who intend to use the FSM30X for their own software development should purchase STM32F4x1RE Nucleo series separately. There are two options to evaluate the FSM30X modules, which will be explained in detail in the next section.



Figure 3: FSM30X connected to STM32F411RE Nucleo Board (I2C mode shown)

2. Software

2.1. FSM30X with STM32 Nucleo Board Running a PC Demo Application

Customers can download STM32 Nucleo board image with Hillcrest software that allows communication between the FSM30X and Freespace[®] MotionStudio 2.

Freespace[®] MotionStudio 2 is a Windows application to allow users to control and configure the FSM30X through a USB interface. This approach can be used for a quick evaluation of the FSM30X. A generalized system diagram is shown in Figure 4.





2.1.1. Program STM32F4 Nucleo Board with the Bridge Binary Download Nucleo Bridge Binary from <u>https://www.hillcrestlabs.com/downloads/bridge-firmware-for-nucleo</u>.

Program STM32 Nucleo board you purchased with the bridge image:

- Open STM32 ST-LINK Utility.
- Target->Connect.
- Target->Program.
- In new window select "File path" to locate the file you just downloaded.
- Click "Start".

5 STM32 ST-LINK	K Utility							_		Х
<u>F</u> ile <u>E</u> dit <u>V</u> iew	<u>T</u> arget ST-LI	NK External Lo	oader <u>H</u> elp							
🖴 🖥 🖕 🖑 🏈 🚿 🧝 🔜										
Memory display						Device	STM32F411xC/E			
Address: 0x08000000 Size: 0x49ED Data Width: 32 bits Device ID 0x431										
Address: 0x0000000 V Size: 0x49FD Data width: 32 bits V Revision ID Rev A						Rev A				
Device Memory @ (x0800000 · =	la : 1000 2059 1	2.1.17 hov			Flash size	512KBytes			
Target memory, Add	ress range: [0x08	3000000 0x08004	49FD]							poate
Address	0	4	8	с	ASCII					^
0x08000000	20008030	08004839	0800486D	08004871	0€.9	Hm.Hq.H	1			_
0x08000010	08004875	08004879	0800487D	00000000	u H y	H}H				
0x08000020	0000000	0000000	0000000	08004881		н				
0x08000030	08004885	0000000	08004889	0800488D	н	%нн				
0x08000040	08004891	08004895	08004899	0800489D	'H•H™H H					
0x08000050	080048A1	080048A5	080048A9	080048AD) ;H¥H©HH					
0x08000060	080048B1	080048B5	080048B9	08001F09	± HµH ¹ H					
0x08000070	08001F0F	08001F19	08001F1F	08001F25	%					_
0x08000080	08001F2B	08001F31	080048BD	00000000	+ 1 ½ H					~
<	1			1						>
16:44:52 : SWD Fre	quency = 4,0 MH:	z.								^
16:44:52 : Connecti	on mode : Normal									
16:44:52 : Debug in 16:44:52 : Device ID	1:0x431	enabled.								
16:44:52 : Device fla	16:44:52 : Device flash Size : 512KBytes									
16:44:52 : Device fa	16:44:52 : Device family :STM32F411xC/E 16:44:57 : [1000-3958 1 3 1 17 bev] opened successfully									
16:44:57 : [1000-3958_1.3.1.17.hex] checksum : 0x001B8BE7										
16:45:10 : Memory p	16:45:10 : Memory programmed in 0s and 813ms.							~		
Debug in Low Power	mode enabled.		Device ID:0x43	1			Core State : Live Update D)isabled		

Figure 5: STM32 ST-LINK Utility Window

2.1.2. Other Requirements

Running Freespace[®] MotionStudio 2 with FSM30X requires the following items.

- ST-LINK/V2 USB driver available from the ST website (http://www.st.com/en/embedded-software/stsw-link009.html).
- ST32 Virtual COM Port Driver from ST website (http://www.st.com/en/developmenttools/stsw-stm32102.html). The FSM30X 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 instruction to complete the installation.
- Freespace[®] MotionStudio 2 application from <u>https://www.hillcrestlabs.com/downloads/freespace-motionstudio-2</u>.

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

Ports (COM & LPT)
 STMicroelectronics STLink Virtual COM Port (COM7)

Figure 6: Device Manager to Check Installed Driver for ST Virtual COM Port

August 20181000-4103 FSM30X Quick Start Guide with Nucleo Revision 1.2

Start Freespace[®] MotionStudio 2 (MotionStudio2.exe) after STM32 Nucleo virtual COM port is successfully detected in your PC.

2.1.3. Running PC Application

Start Freespace® MotionStudio 2

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



Figure 7: Startup Window of Freespace® MotionStudio 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.

- "Device Type" of the bridge is set to ST Nucleo.
- ST Link Virtual COM Port available in your PC appears in "Virtual COM Port" box.
- "Target Device" is set to SHTP over I2C by default. Follow the instruction in the Application Note

 FSM300 Connection with Nucleo [5] Section I2C. Please note that Freespace[®] Motion Studio 2
 currently supports SHTP over I2C and SHTP over SPI interface.
- Use "Connect" button to start.

Hillcrest Labs MotionStudio 2							-		×	
- MotionStudio 2 - Device Connection	Device Connection ?			l	Bridge::NC	Target::NC	Panel::Acti	ve		*
Device Connection Device Information Sensor Control Sensor Data Visualization Sensor Data Logging Virtual Object	Device Connection	Virtual COM Port COM7 STMicroelectronics STI CLKSEL0 0 ~	ink Virtual COM Port (COM SAO 0	M7) ~				(°		~
Disconnected										



D Hillcrest Labs MotionStudio 2					- 0	×		
<u>F</u> ile <u>H</u> elp					\frown			
MotionStudio 2 Device Connection	Device Connection ?		Budge::Nucleo	Target::SensorHub	Panel::Active	- ^		
Sensor Control Sensor Data Visualization	Bridge							
Sensor Data Analysis	Device Type Virtual COM Port							
Virtual Object	ST Nucleo 🗸 COM7 STMicroelectronics STLink Virtual COM Port (COM7) 🗸 😰							
	Target Device					_		
	Transport Protocol CLKSELO	540						
	SHTP over I2C V 0	✓ 0 ✓						
		•						
	Connect Disconnect							
	'Value': '1.0.0\x00'}), Container{{'Length': 2, 'lag': 'Maxi (x7f'}), Container{{'Length': 2, 'Tag': 'MaxTransferWrite'	argoPlusHeaderWrite', 'Value': '\x00\x01'}), C Value': '\x00\x01'}), Container({'Length': 2, '1	Lontainer({'Length': 2, 'Tag Tag': 'MaxTransferRead', '\	g': 'MaxCargoPlusHeaderRe /alue': '\xff\x7f'}), Containe	ead', 'Value': '\xff r({'Length': 5, 'Tag':	^		
	'AppName', 'Value': 'SHTP\x00'}), Container{{'Length': 1 Container{{'Length': 4, 'Tag': 'GUID', 'Value': '\x01\x00\;	'Tag': 'NormalChannel', 'Value': '\x00'}), Con /0\x00'}), Container({'Length': 11, 'Tag': 'App	tainer({'Length': 8, 'Tag': ' Name', 'Value': 'executabl	ChannelName', 'Value': 'co le\x00'}), Container({'Length	ntrol\x00'}), i': 1, 'Tag':			
	'NormalChannel', 'Value': '\x01'}), Container({'Length': Container(('Length': 10, 'Tag': 'AppName', 'Value': 'sen	, 'Tag': 'ChannelName', 'Value': 'device\x00'} orhub\x00'}), Container({'Length': 1, 'Tag': 'N), Container({'Length': 4, ' IormalChannel', 'Value': '\	Tag': 'GUID', 'Value': '\x02\x x02'}). Container({'Length':	:00\x00\x00'}), 8. 'Tag':			
	"ChannelName" "Value": "control/v00"3). Container#"Ler	th' 1 'Tao'' 'NormalChannel' 'Value' '\v03	3) Container#Tenoth': 12	'Tag': 'ChannelName' 'Va	lue': 'innutNormal	· · ·		
	INFO: Connect Device: Biogdapp of the Constructions of the Construction of the Constru							
	INFO: Connect Device : BridgeApp Event Channel Success INFO: Connect Device : BridgeApp Event Channel Success INFO: Connect Device : Shift Control Channel Success							
	INFO: Connect Device : Executable Device Channel Suc INFO: Connect Device : SH-2 Control Channel Success	255						
	INFO: Connect Device : SH-2 GyroRv Input Channel Sur	ess						
	INFO: Connect Device : SH-2 Normal Input Channel Su	ess						
						•		
Connected								

Figure 9: Device Connection Window after Successful Communication in Freespace® MotionStudio 2

When connection process is completed, the three status indicator text boxes on the upper right hand corner of the panel and the console window on the bottom would 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.

Sensor Control

The Sensor Control panel allows the users 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 operation 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 specified operating period. The "Requested Period (us)" fields which are left blanked or obtained invalid value 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.



Figure 10: Sensor Control Panel in Freespace® MotionStudio 2

Virtual Object

Virtual Object panel shows the orientation of the device. Please note that you need to enable sensors in **Sensor Control** panel, select the sensor from the drop-down menu in **Virtual Object** panel. The sword in the Virtual Object will move 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 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 11: Virtual Object Panel in Freespace® MotionStudio 2



Figure 12: FSM30X Orientation

Please follow the instruction 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 Hillcrest logo on the ground plane points to your forward direction (heading).
- Hold the FSM30X module Y+ axis points to your forward direction as well.
- Select "Game Rotation Vector" in drop down menu and click "Tare Z". Now, sword will point to the edge of the Hillcrest logo and is aligned with your device Y+.

• Switch to "ARVR Stabilized Game Rotation Vector" in drop down menu to evaluated ARVR Stabilized Game Rotation Vector.



Figure 13: Sensor Orientation in Virtual Object Panel in Freespace® MotionStudio 2

2.2. FSM30X with STM32 Nucleo Board in 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\ directory. Please skip this step if you have installed already from Section 2.1.2.
- ST32 Virtual COM Port Driver from the ST website. Please skip this if you have installed already from Section 2.1.2.



Figure 14: 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.
- 2.3. FSM30X with STM32 Nucleo Board Running Example Application

Hillcrest provides a complete software package for the STM32F4x1RE Nucleo boards.

The example application for the FSM30X source code is available in public github.

https://github.com/hcrest/bno080-nucleo-demo

Clone this repository using the --recursive flag with git. Alternatively, you can download a ZIP file from the link.

git clone --recursive https://github.com/hcrest/bno080-nucleo-demo.git

Everything required to obtain outputs from the FSM30X is included in this package. The software package incorporates the FSM30X sensor hub driver, enabling SH2 functionality for the development system.



Figure 15: Simplified System Diagram (blue indicates driver developed by Hillcrest)



Figure 16: Source Code Structure

The software is organized as an IAR EWARM project that can be dropped into the IAR IDE 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-example-nucleo/EWARM/Project.eww". This should open an IAR workspace with all the files within the project.
- Select one of sh2-demo-i2c, sh2-demo-spi, sh2-demo-uart or demo-rvc in the project configuration based on the board connection. Refer the information in the Application Note *FSM300 Connection with Nucleo* [5]. Please note that all 4 protocols are supported in MCU workspace.
- 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 Hillcrest communications protocol for the BNO080 and User's Guide for Hillcrest's SH-2 driver.

The reader is encouraged to review the *BNO080 datasheet* [1] and the *SH-2 Reference Manual* [2] for details on how to construct messages.

The output from the FSM30X is printed through the serial port. The first few lines indicate that the host has established proper communication ("Product ID Request") with the BNO080 and the BNO080 has responded with version information ("Product ID Response").

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

🜉 COM7:115200baud - Tera Term VT	_	Х
File Edit Setup Control Window Help		
Eur Fair Serah eFunda Wundan Web	_	_
		~
HITCHEST SH-2 Demo.		
SHZ RESEL.		
Part 10003608 : Version 5.2.7 Build 370		
Starting Concern Denets		
Starting Sensor Reports		
A_{1} and B_{1} and B_{2} and B_{2		
0.3008 Relation Vector: 1.0.003 1.0.002 1.0.019 R.0.797 (acc. 100.001 deg)		
0.3109 Rotation Vector: 1.0.003 i0.002 i.0.019 k.0.797 (acc. 140.499 deg)		
0.3210 Rotation Vector: 1.0.003 i0.002 i.0.019 k.0.797 (acc. 118.746 deg)		
0.317 Rotation Vector: r:0.003 i:-0.002 j:0.019 k:0.797 (acc. 118.746 deg)		
0.3417 Rotation Vector: r:0.003 i:-0.002 j:0.019 k:0.797 (acc. 118.746 deg)		
0.3517 Rotation Vector: 1:0.003 i:-0.002 i:0.019 k:0.797 (acc: 118.746 deg)		
0.3719 Potation Vector: r:0.603 i:-0.002 i:0.019 k:0.797 (acc: 118.746 deg)		
0.3819 Rotation Vector: r:0.603 i:-0.002 i:0.019 k:0.797 (acc. 118.746 deg)		
0.3019 Rotation Vector: 1:0.003 i:-0.002 i:0.019 k:0.797 (acc: 118.746 deg)		
0.4019 Rotation Vector: r:0.603 i:-0.002 i:0.019 k:0.797 (acc: 118.746 deg)		
0.4129 Rotation Vector: r:0.603 i:-0.002 i:0.019 k:0.797 (acc: 118.746 deg)		
0 4229 Rotation Vector: r:0 603 i:-0 002 i:0 019 k:0 797 (acc: 118 746 deg)		
0 4329 Rotation Vector: r:0 603 i:-0 002 i:0 019 k:0 797 (acc: 118 746 deg)		
0.4430 Rotation vector: r:0.603 i:-0.002 i:0.019 k:0.797 (acc: 118.746 deg)		
0.4525 Rotation vector: r:0.603 i:-0.002 i:0.019 k:0.797 (acc: 118.746 deg)		
0.4629 Rotation vector: r:0.603 i:-0.002 i:0.019 k:0.797 (acc: 118.746 deg)		
0.4729 Rotation vector: r:0.603 i:-0.002 i:0.019 k:0.797 (acc: 118.746 deg)		
0.4829 Rotation vector: r:0.603 i:-0.002 i:0.019 k:0.797 (acc: 118.746 deg)		
0.4935 Rotation vector: r:0.603 i:-0.002 i:0.019 k:0.797 (acc: 118.746 deg)		
0.5039 Rotation vector: r:0.603 i:-0.002 i:0.019 k:0.797 (acc: 118.746 deg)		
0.5139 Rotation vector: r:0.603 i:-0.002 j:0.019 k:0.797 (acc: 118.746 deg)		\sim

Figure 17: Terminal Emulator Screenshot

Roject - IAR Embedded Workbench IDE						_ 0 <u>_ x</u>
File Edit View Project Debug Disasse	embly ST-LINK Tools Window Help					
EIM SWU						
Workspace × main.c		main() 🔻	• ×	Disassembly		×
sh2-demo-i2c • 63	/* USER CODE END PV */		-	Go to	 Memory 	
Files 😤 📴 64				Disassembly		*
⊟ fil sh2-demo ✓ 65	/* Private function prototypes*			int main(void)		
Application 67	static woid MX GPTO Init (woid):			{		
Drivers 68	static void MX I2C1 Init(void);			main:		
Hillcrest 69	<pre>static void MX_USART2_UART_Init(void);</pre>			Cx80063ac: 0	Jxb530	PUSH
Middlewar 70	<pre>static void MX_SPI1_Init(void);</pre>			0x80063ae: 0	Jxb08b	SUB
□ □ Output 71	<pre>void StartDefaultTask(void const * argument);</pre>			HAL_Init();	0	DT
72	/* USER CODE BEGIN PEP */			SwetenClock Co	mfig():	DL
74	/* Private function prototypes*		=	0x80063b4 : 0	0xf000 0xf8b3	BL
75				GPIOC CLK EN	MABLE();	
76	/* USER CODE END PFP */			0x80063b8: 0)x2000	MOVS
77	(* HEED CODE PECTN 0 */			0x80063ba: 0)x9000	STR
70	/ OSER CODE BEGIN 0 -/			HAL_GPIO_Init((GPIOB, &GPIO_	InitStru
80	/* USER CODE END 0 */			0x80063bc: 0	Jx4c81	LDR.N
81				Ux80053be: 0	Jx4882	LDR.N
82	int main (void)			0x80063c0: 0)x6601 0xf041 0x0104	OPP N
83				0x80063c6: 0	0x6001	STR
85	/* USER CODE BEGIN 1 */			0x80063c8: 0	Jx6801	LDR
86				0x80063ca: ()xf001 0x0104	AND. W
87	/* USER CODE END 1 */			0x80063ce: 0)xf000 0xf8a0	BL
88				HAL_GPIO_Init((GPIOB, &GPIO_	InitStru
89	/* MCU Configuration	-*/		0x80063d2: 0	Jxf041 0x0101	ORR.W
90	/* Reset of all peripherals. Initializes the Flash interface and the Systick.	*/		0x80063d6: 0	JX6001	TDP
92	HAL Init();	1		0x80063da: ()x6001 0xf001 0x0101	AND H
93				0x80063de: 0	Jxf000 0xf898	BL
94	/* Configure the system clock */			HAL_GPIO_Init((GPIOB, &GPIO_	InitStru
95	SystemClock_Config();			0x80063e2: 0)xf041 0x0102	ORR.W
97	<pre>/* Initialize all configured peripherals */</pre>			0x80063e6: 0	Jx6001	STR
98	MX_GPIO_Init();			0x80063e8: 0	Jxa901	ADD
99	MX_I2C1_Init();			Ux80063ea: U	Jx6800 0.4000 0.0000	LDR AND U
100	MX_USART2_UART_Init();			0x80063EC. 0	0x9000	STR
101	MX_SPI1_init();			0x80063f2: 0	Jx9800	LDR _
sh2-demo				· · · · · · · · · · · · · · · · · · ·		
		,				
Log						*
Wed Jan 25, 2017 13:27:13: Unload	ded macro file: C:\IAR_Systems\EmbeddedWorkbench_v7.4\arm\config\flashloader\ST\FlashST	M32F4xxx.mac	2			
Wed Jan 25, 2017 13:27:13: Downle	oaded C:\Kyungjin\local_git\bno080-nucleo-demo\EWARM\sh2-demo-i2c\Exe\sh2-demo.out to fla	ash memory.				
Wed Jan 25, 2017 13:27:13: Hardw	rare reset with strategy 0 was performed					
Wed Jan 25, 2017 13:27:14: 27468	bytes downloaded into FLASH and verified (8.30 Kbytes/sec)					
Wed Jan 25, 2017 13:27:14: Loade	ra aebagee. Wikiyangjinijadai_gitijanabob-nucleo-aemolic WAMM(snz-aemo-izd)exe(shz-demo.ou wa rasat was parformad	it.				E
Wed Jan 25, 2017 13:27:14: Soliwa	reset					
le di dan co, com rocerri i raigo						
Debug Log Build		-	_			¥
Ready					N	UM 🔤

Figure 18: IAR EWARM Screenshot

References

- 1. 1000-3927 BNO080 Datasheet, Hillcrest Labs
- 2. 1000-3625 SH-2 Reference Manual, Hillcrest Labs
- 3. 1000-3600 SH-2 SHTP Reference Manual, Hillcrest Labs
- 4. 1000-4086 FSM30X Datasheet, Hillcrest Labs
- 5. 1000-4094 FSM300 Connection with Nucleo, Hillcrest Labs

Copyright © 2018 Hillcrest Laboratories, Inc. All rights reserved.

Notices

© Copyright 09/2019 CEVA, Inc. and/or its subsidiaries ("CEVA") All rights reserved. All specifications are subject to change without notice.

Freespace is a registered trademark of CEVA. Other company and product names mentioned in this document may be the trademark or registered trademark of their respective owners.

Disclaimer: The information furnished herein is believed to be accurate and reliable. However, the information is provided "AS IS", without any express or implied warranty of any kind including warranties of merchantability, non-infringement of intellectual property, or fitness for any particular purpose.

In no event shall CEVA or its suppliers be liable for any claims and/or damages whatsoever arising out of the use of or inability to use the materials. CEVA and its suppliers further do not warrant the accuracy or completeness of the information, text, graphics or other items contained within these materials. CEVA may make changes to these materials, or to the products described within.

www.ceva-dsp.com



FOR MORE INFORMATION:

