SFM2 Sensor Fusion Module User Manual

The SFM2 is the smallest form-factor and lowest power wireless 9 DOF Sensor Fusion Module with optional Pressure, Temperature and Humidty sensors. The SFM2 offers both a Bluetooth Low Energy and a USB/COM port interface. The SFM2 provides an extremely flexible BLE and USB interfaces and allows complete customization of sensor output streams and sensor configurations. **The SFM2 can easily be worn using the Velcro strap ran through slot in the bottom of the enclosure.**

Shown with Black Enclosure. For other color options please inquire

1.0 Product Highlights

- 40.9 x 28 x 14.3mm (1.61" x 1.10 x 0.563") Form Factor
- Wearable: Enclosure has slot for Velcro strap.
- USB and BLE Interfaces: **Connect up to 5 SFM2's running 208Hz Fusion ODR and Quaternion data with newer iOS and Android Devices. High ODR Rates possible depending on mobile device performance.**
- Field Updatable using 'Over the Air' update
- 9 DOF: 3-Axis Gyro, Accel, Mag
  - with Pressure, Temp, and Humidity Sensors
- Up to 833Hz Sensor Fusion Operation. Multiple Streams can be enabled at same time.
  - Tared & Un-Tared Quaternion Output
  - Fusion Compass Heading and Tilt Output
  - Fusion Linear Acceleration Output
  - Fusion Euler Angle Output
- Sensor Fusion Output via BLE or USB Interface
  - 833, 417, 208, 104, 52, 26Hz ODR Rate Selections
- Raw Sensor Output availability for all sensors. Accel/Gyro Max ODR = 1667Hz
- USB(1Mbps) and BLE Interfaces(DLE and 2Mb PHY capability)
- **iOS** and **Android** apps for sensor and data output configuration.
- USB/COM command set for sensor and data output configuration.
- Python API Library to quickly and easily get application up and running
- TARE function to zero orientation for quaternion output.
- GLOBAL REFERENCE enable/disable command
- TIMESTAMP sensor data with 25us resolution with improved accuracy using RTC synchronization
- Timestamp-timestamp delta accuracy ± 1µs for data logs
- Multi-Mode Time Synchronization to time synchronized multiple SFM2's used simultaneously with Average Standard Deviation between timestamps of multiple SFM2's typical <10ms
- Rechargable Battery (130mAh)
  - ~73hour battery life with Quaternion output @ 26Hz using BLE interface
  - ~47hour battery life with Quaternion output @ 104Hz using BLE interface
  - ~30hour battery life with Quaternion output @417Hz using BLE interface
- Individual Control of each sensor output to be included in Data Stream
  - Sensors and Sensor Fusion can be completely customized including Sensor Filters.
- Custom OEM/Logo Enclosure available
- Example Application Software

2.0 Table of Contents

3.0 Custom Development and OEM Solutions

Sensor Maestros can provide customized solutions using the SFM2 as a foundation. Below are some examples of the common customizable options for the SFM2.

- Customized Hardware
  - Additional Sensors
    - Temperature/Humidity
    - Gas: CO2, CO, VOC's
    - PIR
  - Customized Battery options
- Custom Motion Algorithms
  - Pattern Recognition
  - Machine Learning
  - Vibration Analysis
  - Fitness/Sports Analysis
- Custom Software Applications
  - iOS
  - Android
  - PC Applications
- Custom Cloud Applications
- Custom Enclosure Options

4.0 Future Options

- Compressed Audio Recording Streaming
- AI / Machine Learning Application Development
- User Application Flash Memory area to all for User applications to be programmed directly onto the SFM2
- SENSR-POD
  - BLE to Cellular Bridge
- Vibration Analysis
- Cloud Enablement

5.0 SFM2 Hardware Description

The SFM2 is comprised of a 64MHz Cortex M4F BLE SOC, Motion Sensors, and optional Pressure, Relative Humidity and Temperate Sensors. A block diagram is shown below with further details below.

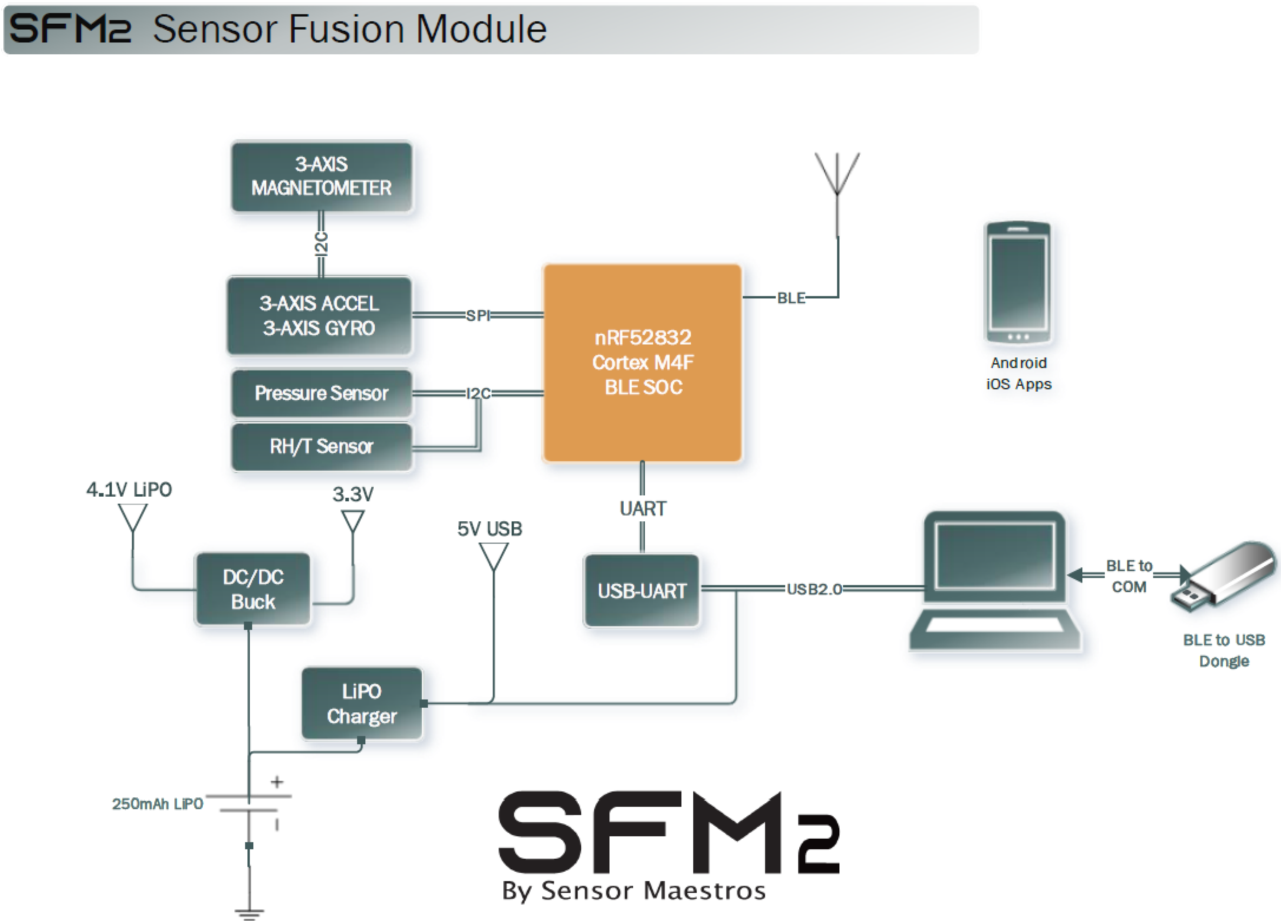


Figure 1: SFM2 Block Diagram

- 64MHz Cortex M4F MCU
- 9DOF Sensors
  - LSM6DSO – 6 Axis Accelerometer/Gyroscope
  - LIS2DML – 3 Axis Magnetometer
- Optional Environmental Sensors
  - LPS22HH – Pressure Sensor
  - ENS210 – Relative Humidity and Temperature Sensor
- BLE interface to provide easy means for application developers to make use of the SFM2 for developing custom applications.
- Example Software provided for BLE Client side for how to interface to the SFM2
  - Example Python 3D Cube application used to interface to the SFM2 from a USB/CDC Connection.
    - Python API Library
  - Native C++ Library(coming soon)
- USB2.0 Support:
  - SFM2 can be controlled/configured from a COM Terminal program using Serial Commands described in USB Commands and for higher level Python API's see Python API Library.
  - Sensor Fusion Data can be sent via USB2.0
  - USB provides Battery charging
- Highly efficient BLE/Characteristic architecture
  - Every Sensor has its own Sample Rate, Full-Scale Range, Data Enable
  - Every Sensor has a Notification Enable

6.0 ACCEL/GYRO/SENSOR FUSION Specifications

Symbol	Parameter	Test conditions	Min.	Typ.[1]	Max.	Unit
FS_ODR	Sensor Fusion Output Data Rate		12.5		833	*Hz*
LA_FS	Linear acceleration measurement range			±2		*g*
				±4		
				±8		
				±16		
G_FS	Angular rate measurement range			±125		dps
				±250		
				±500		
				±1000		
				±2000		
LA_So	Linear acceleration sensitivity[2]	FS = ±2 *g*		0.061		m*g*/LSB
		FS = ±4 *g*		0.122		
		FS = ±8 *g*		0.244		
		FS = ±16 *g*		0.488		
G_So	Angular rate sensitivity[2]	FS = ±125 dps		4.375		mdps/LSB
		FS = ±250 dps		8.75		
		FS = ±500 dps		17.50		
		FS = ±1000 dps		35		
		FS = ±2000 dps		70		
LA_SoDr	Linear acceleration sensitivity change vs. temperature[4]	from -40° to +85°		±0.01		%/°C
G_SoDr	Angular rate sensitivity change vs. temperature[4]	from -40° to +85°		±0.007		%/°C
LA_TyOff	Linear acceleration zero-g level offset accuracy[5]			±20		m*g*
G_TyOff	Angular rate zero-rate level[5]			±1		dps
LA_OffDr	Linear acceleration zero-g level change vs. temperature[4]			±0.1		m*g*/ °C
G_OffDr	Angular rate typical zero-rate level change vs. temperature[4]			±0.010		dps/°C
Rn	Rate noise density in high-performance mode[6]			3.8		mdps/√Hz
RnRMS	Gyroscope RMS noise in normal/low-power mode[7]			75		mdps
An	Acceleration noise density in high-performance mode[8]	FS = ±2 *g*		70		µ*g*/√Hz
		FS = ±4 *g*		75		
		FS = ±8 *g*		80		
		FS = ±16 *g*		110		

Symbol	Parameter	Test conditions	Min.	Typ.[1]	Max.	Unit
RMS	Acceleration RMS noise in normal/low-power mode[9] [10]	FS = ±2 *g*		1.8		m*g*(RMS)
		FS = ±4 *g*		2.0		
		FS = ±8 *g*		2.4		
		FS = ±16 *g*		3.0		
LA_ODR	Linear acceleration output data rate			12.5 26 52 104 208 416 833 1666		Hz
G_ODR	Angular rate output data rate			12.5 26 52 104 208 416 833 1666		

1. *Typical specifications are not guaranteed.*
2. *Sensitivity values after factory calibration test and trimming.*
3. *Subject to change.*
4. *Measurements are performed in a uniform temperature setup and they are based on characterization data in a limited number of samples. Not measured during final test for production.*
5. *Values after factory calibration test and trimming.*
6. *Gyroscope rate noise density in high-performance mode is independent of the ODR and FS setting.*
7. *Gyroscope RMS noise in normal/low-power mode is independent of the ODR and FS setting.*
8. *Accelerometer noise density in high-performance mode is independent of the ODR.*
9. *Accelerometer RMS noise in normal/low-power/ultra-low-power mode is independent of the ODR.*
10. *Noise RMS related to BW = ODR/2.*

7.0 Magnetometer Specifications

Symbol	Parameter	Test conditions	Min.	Typ.	Max.	Unit
FS	Magnetic dynamic range			±49.152		gauss
So	Sensitivity		-7%	1.5	+7%	mgauss/ LSB
TcyOff	Magnetic Sensor Offset	With offset cancellation	-60	1.5	+60	mgauss
RMS	RMS Noise	High Performance Mode		3		mgauss (RMS)

8.0 Pressure Sensor Specifications

Symbol	Parameter	Test conditions	Min.	Typ.	Max.	Unit
P	Pressure Operating Range		260		1260	hPa
P_{bits}	Pressure bits of data		-7%	24	+7%	mgauss/ LSB
P_{sens}	Pressure Sensitivity			4096		LSB/hPa
P_{AccRel}	Relative Accuracy over pressure	P = 800-1100 hPa T = 25°C		±0.025		hPa
P_{AccT}	Absolute accuracy over temperature	Pop, T = -20 to 80°C		±0.5		hPa
ODR_{Pres}	Pressure output data rate			1 10 25 75 100 200		Hz

9.0 Relative Humidity Sensor Specifications

Symbol	Parameter	Test conditions	Min.	Typ.	Max.	Unit
P	Pressure Operating Range		260		1260	hPa
P_{bits}	Pressure bits of data		-7%	24	+7%	mgauss/ LSB
P_{sens}	Pressure Sensitivity			4096		LSB/hPa
P_{AccRel}	Relative Accuracy over pressure	P = 800-1100 hPa T = 25°C		±0.025		hPa
P_{AccT}	Absolute accuracy over temperature	Pop, T = -20 to 80°C		±0.5		hPa
ODR_{Pres}	Pressure output data rate			1 10 25 75 100 200		Hz

10.0 Temperature Sensor Specifications

Symbol	Parameter	Test conditions	Min.	Typ.	Max.	Unit
P	Pressure Operating Range		260		1260	hPa
P_{bits}	Pressure bits of data		-7%	24	+7%	mgauss/ LSB
P_{sens}	Pressure Sensitivity			4096		LSB/hPa
P_{AccRel}	Relative Accuracy over pressure	P = 800-1100 hPa T = 25°C		±0.025		hPa
P_{AccT}	Absolute accuracy over temperature	Pop, T = -20 to 80°C		±0.5		hPa
ODR_{Pres}	Pressure output data rate			1 10 25 75 100 200		Hz

11.0 LED Operation Description

COLOR	FUNCTION	NOTES
YELLOW	Connected	Will blink twice every ~2.5 seconds to indicate SFM2 is connected to a BLE Client
YELLOW	Advertising	Will blink once every ~1 second to indicate SFM2 is in advertising mode
RED	Sensor/Sensors Enabled	Connected: Blink twice every ~2.5 seconds Unconnected: Blink once every ~1 second This LED is sequenced with the YELLOW LED if a Sensor is Active/ON.
BLUE	Active Battery Charging	Solid BLUE Led indicates battery is charging via USB.

Table 1: LED Operation

12.0 BLE Operation

The SFM2 provides BLE Peripheral(slave) operation and allows any BLE Client(master) to connect to the SFM2. The SFM2 does not require Bonding. Description of operation is provided below.

1) **BLE ADVERTISING:** Upon powering the device the SFM2 Performs BLE Advertising and advertises the Device name "SFM2".
    a. YELLOW BLE LED Blinks at ~1 second interval
    b. Advertising Interval = 500ms
    c. Currently the device does not allow user adjustment of the Advertising Interval.

    NOTE: The Device Advertises as a Connectable BLE Peripheral and does not require any Bonding/Secure Pairing.
2) **BLE CONNECTED:** Once a BLE Client has connected to the SFM2 the device will transition to what is shown below and allows the BLE Client to read all the Services, Characteristics, and Descriptors to allow an application to be developed.
    a. YELLOW BLE LED Blinks at ~2.5 second interval
    b. All BLE Services, Characteristics, Descriptors are available to be read from the Client.
    c. The SFM2 Max/Min Connection interval is set for Maximum throughput via a BLE Connection. Not all BLE Clients in particular Mobile Devices will allow for the minimum 7.5ms Connection interval. Sensor Sample Rates and Sensor Fusion Output Rates should be configured according to the maximum throughput allowed by the Client BLE device.
3) **BLE DISCONNECTED:** Upon a BLE Client(master) disconnecting from the SFM2 the device will return to the **BLE ADVERTISING** mode.

13.0 SFM2 Firmware Update

The SFM2 embedded firmware is not available in source code as open source. Licensing can be made available upon requests. Developers/users of the SFM2 can make use of the USB Commands and/or the BLE Characteristics to configure and retrieve data from the SFM2. Sensor Maestros can provide design services for customization of the embedded firmware for specific use cases on a case by case basis. The SFM2 firmware can be updated via a BLE Connection.

The SENSR-LOGR mobile app will automatically detect if there is a new firmware version available for the SFM2. The SENSR LOGR App can be found in the Apple App Store and on Google Play by searching for Sensor Maestros.

Android: https://play.google.com/store/apps/details?id=sensormaestros.SensorMaestros
iOS: Go to the 'App Store' on your mobile device and search for SENSR-LOGR or Sensor Maestros

If the SENSR-LOGR app detects that there is a newer version of firmware it will automatically prompt you to update to the latest version of firmware. It is HIGHLY RECOMMENDED to update to the newest version of firmware.

14.0 Sensor Fusion

The SFM2 provides for 9DOF Sensor Fusion output that can be enabled by the user. The following Sensor Fusion output streams are provided by the SFM2 and can be individually enabled/disabled using either the BLE or USB interface.

- Quaternion
- Tared Quaternion
- Compass Heading and Tilt
- Linear Acceleration
- Euler Angle

Sensor Fusion output rates (ODR) can be selected by the user from 0(Disabled), 12.5, 26, 52, 104, 208, 417, and 833Hz.
The Sensor Fusion ODR(SFOR) can be set to an ODR rate equal to the highest ODR rate of either the GYRO or ACCEL. For example if 417Hz SFOR ODR rate is desired either the ACCEL or GYRO or both should be configured also to 417Hz. For optimum results it is recommended to set the ACCEL & GYRO ODR rates to 2X the SFOR ODR rate. Technically the SFOR ODR can be greater than the ODR rate of either the ACCEL or GYRO but not both. This configuration is NOT recommended though. The Magnetometer does not need to be ODR rate correlated with the Sensor Fusion ODR rate though for any Sensor Fusion ODR rates 104Hz or greater it is highly recommended to set the Magnetometer to its max value of 104Hz. The MAG ODR rate does have a requirement that the ODR rate can't be greater than both the ACCEL and GYRO. An acceptable configuration is shown below though NOT recommended.

ASR = 12.5Hz GSR = 208Hz MSR = 104Hz

14.1 Coordinate Systems

The SFM2 uses two different Coordinate Systems(CS):

- Local Coordinate System(LCS)
- Global Coordinate System(ENU)

The Global Coordinate System is fixed with the external world and it is ENU standard(East—North-Up, acceleration positive) the same that is used in the Android coordinate system. It means that Global Coordinate System axes are oriented as follows:

- X is East
- Y is North
- Z is Up

The LCS is a rigidly fixed coordinate system affixed to the SFM2 device and by default is as shown below:

Using the 'TARE' function this coordinate system can be re-oriented to any position the SFM2 has been moved to and will be effectively Zero'd at the position it is in prior to using the 'TARE' function.

This means that when the SFM2 rotates(changes its orientation) the LCS rotates with it in the exact same manner. All the SFM2 attitude streams except SFQ provide attitude of current LCS provided in ENU. SFQ provides attitude of default LCS given in ENU. The sensor measurement vectors(acceleration, angular rate, magnetic) and SFLA(Sensor Fusion Linear Acceleration) are by default given in LCS but can also be given in ENU by enabling GLOBAL REFERENCE by setting GLOBREF = 1.

14.2 Tare Quaternion

The SFM2 variable SFTARE! provides the ability to change the LCS orientation. Executing SFTARE! creates a new LCS that is the same as ENU in the moment of executing the command. After taring the SFM2 the LCS orientation is exactly the same as ENU so all angles are zero. This allows you to change the default LCS to any other you want. This is **EXTREMELY USEFUL** if the SFM2 has been mounted or moved the user can use the SFTARE! function to re-orient the LCS to this new orientation. This provides an easy means for the user to always start from a known orientation starting point regardless of the physical orientation of the SFM2.

Example: Attach the SFM2 to some piece of equipment that is stationary during the attachment. The orientation of the SFM2 with respect to the equipment doesn't matter. Orient the equipment in such a way that its X axis point East, Y axis points North and Z axis points Up(the orientation of equipment is the same as ENU). With the SFM2 in a stable position execute the SFTARE! command or execute this via the Mobile App. Now the SFM2 outputs are given in the new LCS which is the same as the equipment's coordinate system.

14.3 Headtare

Sometimes it is not practical to rotate the equipment to match the ENU axes for taring. Then you can 'TARE' in any orientation(making sure the equipment is not tilted) and then manually correct the SFM2 using the Headtare function (HEADTARETARE=<angle_in_deg>) according to the current equipment heading angle. The angle is measured between the West-East axis and the equipment axis is in the counter-clockwise manner(looking from above). Headtare allows for a manual alignment adjustment to the SFM2 in degrees.

14.4 Global Reference

To understand the Global Reference option, please get familiar with the 'Coordinate Systems' first being LCS and ENU.

All sensor measurement vectors(acceleration, angular rate, magnetic) and SFLA(Sensor Fusion Linear Acceleration) are by default given in LCS.

Tracking Acceleration Example: It is useful for example when you want to measure the acceleration of your car. You fix the SFM2 to your car, tare and adjust the heading using headtare if needed. Then all the accelerations related to car acceleration or breaking appear on the X axis. When the car accelerates there are positive readings on the X axis, when it is braking there are negative readings on the X axis. It is no matter how the car is oriented in the external world the X axis is always affixed to forward and backward motion. No matter whether you are driving South-North or East-West the reading are always on the X axis.

In the presented above car example using measurements given in the LCS is the best solution because it gives you exactly what you want.

The case is completely different when you want to track the trajectory of a ball.

Tracking Trajectory Example: In this case you would put the SFM2 into the ball and affix it in some manner and throw it. Now you want to draw the flight trajectory. In the simplest form to get the position(relative to the start point)you have to integrate the acceleration twice. It would be easy if the ball would not be spinning. When the ball(and SFM2) is spinning its LCS axis are changing orientation all the time so the acceleration measured on the X axis sometimes refers to North-South movement, sometimes East-West and sometimes Up and Down. It all depends what is the actual SFM2 orientation in the external world at that moment in time. Calculating trajectory using measurements given in LCS are not easy and require a bit of math.

The trajectory calculation would be GREATLY SIMPLIFIED if the measurement would be given in some external(global) coordinate system that is fixed NOT with the SFM2/Ball BUT with the external world. This can be achieved using the 'Global Reference' option by setting GLOBREF=1.

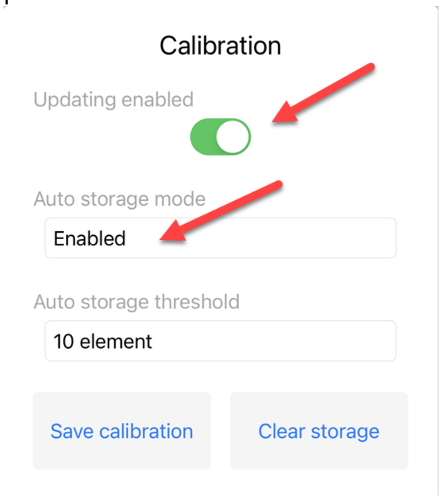
When GLOBREF=1 is set all the measurements are given in the ENU. It means that all the acceleration on the X axis corresponds to East-West movement, on the Y axis to North-South movement and on the Z axis to Up-Down movement no matter what the ball orientation is. Calculating the trajectory using measurements given in ENU is MUCH EASIER because you don't need to take the ball spin into account.

- When GLOBREF=1 all vector outputs(raw sensor data and SFLA) are represented in the global reference frame and are not affected if the SFM2 is 'tared'.

- When GLOBREF=0 and the device is 'tared' all vector outputs are rotated to match the new orientation. This means that after taring the SFM2 the Accelerometer output AD should always read approximately x=0, y=0, z=1, until the orientation is changed.

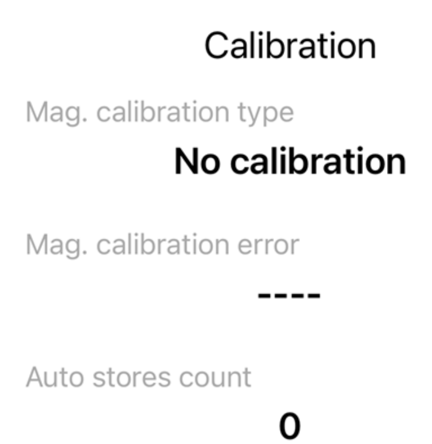
15.0 Manual Calibration

The SFM2 is provided with factory calibration. In general it is not recommended to perform a full re-calibration however there may be some cases where this may be necessary. Also it should be noted that with 'Updating enabled' and 'Auto storage mode' enabled the SFM2 will continuously look for a better calibration than what is currently being used. If a better calibration set of parameters is found as may be the case if the SFM2 has been mounted to a particular object those new parameters will be used in place of the prior calibration values.

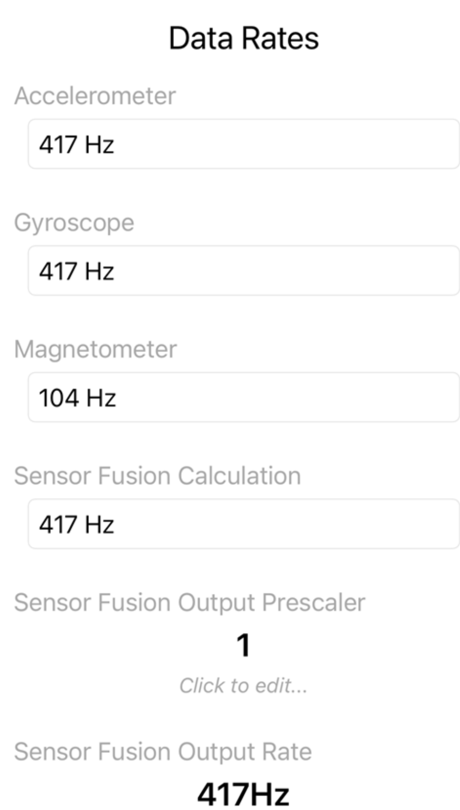


To determine if the SFM2 already has had a calibration performed go to the Stats->Calibration section. If it is blank as shown below a manual calibration will need to be performed as there is no calibration values currently stored in the SFM2.

No Calibration Parameters stored in SFM2



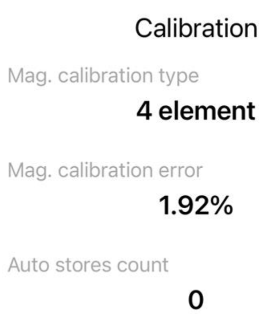
Typical Config Values for performing a manual calibration.

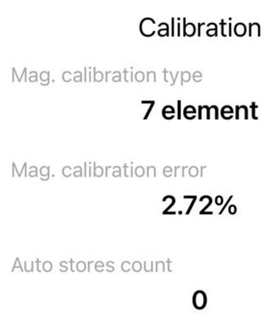


15.1.1 Calibration Steps

1) STEP 1: Recommended Manual process is to hold the SFM2 away from any large metal objects as metallic objects will affect the magnetic calibration process.
2) STEP 2: Hold the SFM2 in open air and rotate/roll the SFM2 around its X axis multiple times trying to keep the rotations smooth and at a consistent speed.

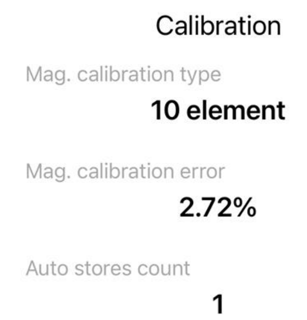
3) STEP 3: Do this while you are viewing the Stats->Calibration view so you can see the Calibration parameters and Error being updated in real time.
4) STEP 4: After you see the initial calibration values appear starting with 4 element values continue the same rotation and you should soon see the Calibration values update to the 7 Element values as shown below…



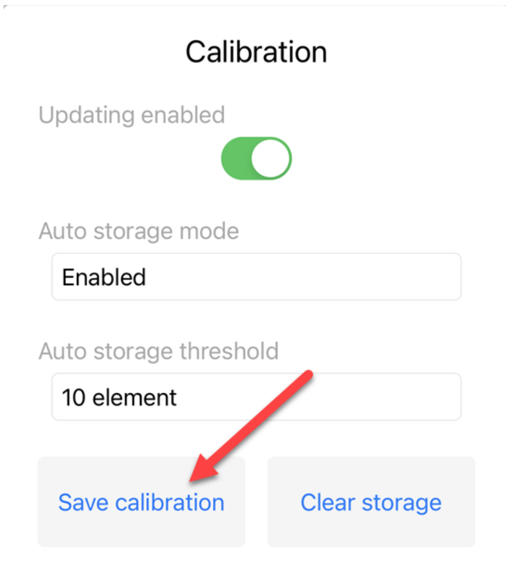


5) After seeing the Calibration values update to 7 Element values now angle the SFM2 slightly so it is pointing either slightly up or slightly down. **NOTE: The strap can be ignored in the picture below.**

6) You will typically see the Calibration values update again to 10 element values after angling the SFM2 while still rotating it around the X axis. Now angle the SFM2 in the opposite direction you had it in STEP 4: to see if the 'Mag calibration error' value improves. We are looking for 10 Element Calibration with 'Mag Calibration Error' < 3%.



7) If the 10 element calibration shows a 'Mag calibration error' < 3% then you can go to Config-Calibration and select the 'Save Calibration' button. This will store the current calibration values in Non-Volatile memory so that these values will be used the next time the SFM2 is powered on.



15.1.2 Calibration Notes

1) It can happen that the 4 element values show a Mag Cal error that is less than the 7 element or 10 element. There are fewer elements taken into account but this is still not as good as a 10 element calibration.
2) If after you perform the Calibration the 'Mag calibration error' shows an error > 3% it is recommended to go to Config->Calibration and click the 'Clear Storage' button to ensure any prior Calibration values are cleared from memory. Power cycle the SFM2 and perform the calibration again.
3) Calibration can be performed with NO Data Streams active. The motion sensors should be configured for ODR rates that are suitable for the application.

16.0 Timestamps

Timestamps are crucial to being able to accurately analyze data. The SFM2 and the SENSR-LOGR app provide methods to allow for highly accurate, consistent timestamps to be provided in data streams that would be logged whether using the BLE or USB interface. The default timestamp clocks are defined below.

Clock Source	Tick	Accuracy	Enabled
LSM6DSx	25us	Poor	ALWAYS ON
RTC	30.518us	High	TSDE = 1 & BLE Interface

SENSR-LOGR app by default sets TSDE=1 upon connection to a SFM2. From the USB Interface this can be enabled/disabled however. If a user desires to develop their own mobile app it is highly recommended to ALWAYS set TSDE=1 to allow for providing for accurate timestamps.

An example of RTC output in the binary stream is shown below which could apply for BLE or USB connection.
Suppose that AD stream is enabled @208Hz rate and TS data is turned on.

We receive the following binary frames:

Frame idx	Samples	LSM6DSx clock (25us tick)
1	AD	100000
2	AD	100192 (Sample Delta: 192 ticks * 25us = 4.8ms or 208Hz)
3	AD + TS = 630	100384
4	AD	100576
5	AD	100768
6	AD	100960
7	AD + TS = 1260	101152
8	AD	101344
9	AD	101536
10	AD	101728
11	AD + TS = 1890	101920

The SENSR-LOGR app then uses the TS data to calculate the RTC timestamps corresponding to the remaining AD samples. The timestamps shown in LOG file from the SENSR-LOGR app are already calculated to an actual time value versus being displayed in LSM6DSx 'ticks'.

NOTE: This timestamp synchronization is completely different than synchronizing multiple SFM2's to each other. Synchronizing multiple SFM2's does leverage the highly accurate 32.768kHz clock but there are several additional items that are monitored and adjusted if necessary to synchronize multiple SFM2's.

16.1 SENSR-LOGR timestamps

Timestamps shown in Log files from logging Data from a SFM2 or multiple SFM2's are already converted to an actual time value out to 7 decimal points. If you are using the SENSR-LOGR app it will automatically adjust individual LSM6DSx timestamps if needed by utilizing the RTC time that is output in the binary frames which is highly accurate and provides a reference clock to make minor adjustments in the LSM6DSx timestamp which is output in an Integer value.

Example of a LOG file using the SENSR-LOGR app with SFQT @833Hz which is one of the fastest sampling rates available on the SFM2 for Sensor Fusion outputs. Raw Accelerometer and Gyroscope data can be output as high as 1667Hz.

NOTE: Timestamp Deltas = 1.2268ms and 1.2269ms translating to ~815Hz output rate. The exact output rate can vary slightly but it will be consistent and the timestamps using the SENSR-LOGR app will be very accurate.

Time [s]	W	X	Y	Z	Delta (µs)
1992.2697675	0.42324	0.07057	-0.04128	0.90232	
1992.2709943	0.42172	0.06684	-0.03926	0.90340	1226.800
1992.2722212	0.41955	0.06162	-0.03637	0.90491	1226.900
1992.2734480	0.41857	0.05992	-0.03616	0.90548	1226.800
1992.2746749	0.41550	0.05734	-0.03538	0.90710	1226.900
1992.2759017	0.41486	0.05676	-0.03358	0.90749	1226.800
1992.2771285	0.41432	0.05564	-0.03172	0.90788	1226.800

1992.2783554	0.41513	0.05741	-0.03185	0.90739	1226.900
1992.2795822	0.41694	0.06068	-0.03216	0.90634	1226.800
1992.2808091	0.41911	0.06434	-0.03223	0.90508	1226.900
1992.2820359	0.42085	0.06716	-0.03211	0.90407	1226.800
1992.2832627	0.42198	0.06913	-0.03226	0.90339	1226.800
1992.2844896	0.42384	0.07217	-0.03278	0.90226	1226.900
1992.2857164	0.42410	0.07223	-0.03248	0.90215	1226.800
1992.2869432	0.42549	0.07499	-0.03336	0.90123	1226.800
1992.2881701	0.42568	0.07555	-0.03383	0.90108	1226.900
1992.2893969	0.42532	0.07473	-0.03351	0.90133	1226.800
1992.2906238	0.42448	0.07269	-0.03252	0.90193	1226.900
1992.2918506	0.42362	0.07063	-0.03150	0.90253	1226.800
		STD Deviation 20 Deltas(µs)			**0.050**

16.2 USB/COM Port Timestamps

USB/COM port timestamps are output in a LSM6DSx tick format. An example of the COM port output on a terminal window such as Putty is shown below. Example of COM Port output for 833Hz TSDE=1 Non-Binary mode.

SFQT:5.3619534E-1,-3.3474213E-1,-3.8904034E-2,-7.73905E-1@393955
SFQT:5.361946E-1,-3.347494E-1,-3.8905397E-2,-7.739021E-1@394003
SFQT:5.3619564E-1,-3.3475372E-1,-3.8908757E-2,-7.738995E-1@394051
SFQT:5.3619516E-1,-3.3475843E-1,-3.890932E-2,-7.738977E-1@394099
SFQT:5.361955E-1,-3.3476514E-1,-3.8911525E-2,-7.7389455E-1@394147
SFQT:5.361958E-1,-3.3476412E-1,-3.8911685E-2,-7.738947E-1@394195
SFQT:5.3619444E-1,-3.3477426E-1,-3.89153E-2,-7.7389115E-1@394243
SFQT:5.361948E-1,-3.347757E-1,-3.891803E-2,-7.7389E-1@394291
SFQT:5.361951E-1,-3.3477685E-1,-3.8917627E-2,-7.738895E-1@394339
SFQT:5.3619426E-1,-3.347772E-1,-3.891741E-2,-7.738899E-1@394387
SFQT:5.361937E-1,-3.3477232E-1,-3.8916305E-2,-7.7389234E-1@394435
SFQT:5.3619385E-1,-3.3476683E-1,-3.8914517E-2,-7.7389467E-1@394483
SFQT:5.361941E-1,-3.34764E-1,-3.8915258E-2,-7.738958E-1@394531
SFQT:5.361934E-1,-3.3475825E-1,-3.8915522E-2,-7.7389866E-1@394579
SFQT:5.361922E-1,-3.3475316E-1,-3.891633E-2,-7.739017E-1@394627
SFQT:5.361896E-1,-3.3475327E-1,-3.8911823E-2,-7.739036E-1@394675
SFQT:5.3618896E-1,-3.347374E-1,-3.8905088E-2,-7.7391136E-1@394723
SFQT:5.361908E-1,-3.3471256E-1,-3.8902704E-2,-7.739209E-1@394771

If the data was to be sorted for the first 5 values preceeding the samples shown above it would look like this…

Note: Timestamp Delta = 48ticks => 48 * 25us = 1.2ms or a 833Hz ODR rate.

Time	W	X	Y	Z	Delta(25µs tick)
393715	5.03E-01	-3.39E-01	-2.55E-02	-7.94E-01	
393763	5.03E-01	-3.39E-01	-2.55E-02	-7.94E-01	48
393811	5.03E-01	-3.39E-01	-2.55E-02	-7.94E-01	48
393859	5.03E-01	-3.39E-01	-2.55E-02	-7.94E-01	48
393907	5.03E-01	-3.39E-01	-2.55E-02	-7.94E-01	48

17.0 Synchronize Multiple SFM2's

The SFM2 allows for multiple SFM2's to be synchronized together to allow multiple SFM2's to be used with either the BLE or the USB/COM port interfaces. It is very useful to be able to have multiple SFM2's operating at the same time simultaneously connected via the BLE or the USB interfaces to monitor multiple locations of a person or object. The only way this is useful though is if the data that is logged by each SFM2 is synchronized in time. The SENSR-LOGR app provides an easy means to allow a user to do this using the 'Synchronization' feature in the 'Time' view in the SENSR-LOGR app. This is further described in Multi Mode: Time Synchronization. For USB operation there is an example Python script in the sfm2_python_lib_0_7/examples folder called 'testing_time_synchronization.py' that provides an example of this using the USB interface.

18.0 Binary Frame Mode

The binary frame mode provides a means for optimizing the throughput of the BLE and USB interfaces by packing sensor data and optionally RTC timestamp data in binary frames that can be decoded on either the BLE Client/SENSR-LOGR or USB Application side. For the case of the BLE interface the Binary Frames are sent using the Data Stream 0x0101 Characteristic.

NOTE: The BLE interface ALWAYS operates in Binary Frame mode whereas the USB/COM Port interface can operate either in ASCII or Binary Frame mode. To enable Binary Frame Mode using the USB/COM port interface the BINMODE=1 command should be used.

18.1 Binary Frame Format

Start byte *0xFA*	Data description	Timestamp	Data	End byte *0xFB*
1 byte	2 bytes	4 bytes	Variable length	1 byte

18.2 Start byte

A single byte used as a frame delimiter. It's value is fixed: 0xFA.

18.3 Data description

Indicates what sample types are contained in the frame. The field consists of 16 bits, where each bit indicates if it's corresponding sample type is contained in the frame.

Bit	Sample Type
0	AD - Accelerometer
1	GD – Gyroscope
2	MD – Magnetometer
3	SFQ – Sensor Fusion Un-Tared Quaternion
4	SFQT – Sensor Fusion Tared Quaternion
5	SFLA – Sensor Fusion Linear Acceleration
6	SFEA – Sensor Fusion Euler Angles
7	SFCHT – Sensor Fusion Heading
8	SFM – Sensor Fusion Calibrated Magnetometer
9	PD – Pressure
10	ALT – Altitude
11	TD – Temperature
12	HD – Humidity
13	TS – Time Synch
14-15	Reserved for future use

18.4 Timestamp

Contains a single UInt32 value denoting time in 25us resolution. All samples contained in the frame share the same timestamp.

NOTE: When TSDE=1 the highly accurate RTC time is included in the 'Data' frame. This timestamp is the LSM6DSx time tick.

18.5 Data

Contains measurement sample data.

Sample type	Size	Content
AD - accelerometer	12 bytes	Vector of 3 floats
GD - gyroscope	12 bytes	Vector of 3 floats
MD - magnetometer	12 bytes	Vector of 3 floats
SFQ - Sensor Fusion untared quaternion	16 bytes	Quaternion of 4 floats
SFQT - Sensor Fusion quaternion	16 bytes	Quaternion of 4 floats
SFLA - Sensor Fusion Linear acceleration	12 bytes	Vector of 3 floats
SFEA - Sensor Fusion Euler Angles	12 bytes	3 floats: Roll, Pitch, Yaw
SFCHT - Sensor Fusion Heading	8 bytes	2 floats: Heading, Tilt
SFM - Sensor Fusion calibrated magnetometer	12 bytes	Vector of 3 floats
PD – Pressure	4 bytes	Single float value, in hPa
ALT - Altitude	4 bytes	Single float value, in meters
TD – Temperature	4 bytes	Single float value, in Celsius
HD – Humidity	4 bytes	Single float value, in %
TS – Time Synch	4 bytes	Two uint32 values 1) RTC time in RTC ticks(see TIME command) 2) Configuration index, incremented each time the RTC is set, either with the TIME or TOFFSET commands.

NOTE: If a frame contains more than one sample,the samples are packed one after another in the order as they appear in the table above.

18.6 End byte

A single byte used as a frame delimiter. It's value is fixed: 0xFB.

18.7 Examples

18.7.1 SFQT+SFLA @104Hz

Each data frame contains two samples, one for SFQT, one for SFLA. They always share a common timestamp.

Field	Start byte	Data description	Timestamp	Data		End byte
				SFQT	SFLA	
Value	0xFA	0x30*	4 bytes	16 bytes	12 bytes	0xFB

*0x30 = 0b0000 0000 0011 0000. With ones at 4th and 5th positions (zero-indexed), corresponding to SFQT and SFLA.

18.7.2 AD@208Hz + GD@104Hz

There are twice as many AD samples as there are GD. This means that AD samples are present in every frame, and GD samples in every other frame.

Frame 1 (AD+GD):

Field	Start byte	Data description	Timestamp	Data		End byte
				AD	GD	
Value	0xFA	0x03*	4 bytes	12 bytes	12 bytes	0xFB

*0x03 = 0b0000 0000 0000 0011. With ones at 0th and 1st positions (zero-indexed), corresponding to AD and GD.

Frame 2 (AD):

Field	Start byte	Data description	Timestamp	Data	End byte
				AD	
Value	0xFA	0x01*	4 bytes	12 bytes	0xFB

*0x01 = 0b0000 0000 0000 0001. With a single one at 0th position (zero-indexed), corresponding to AD.

Frame 1 and 2 come alternately, one after the other.

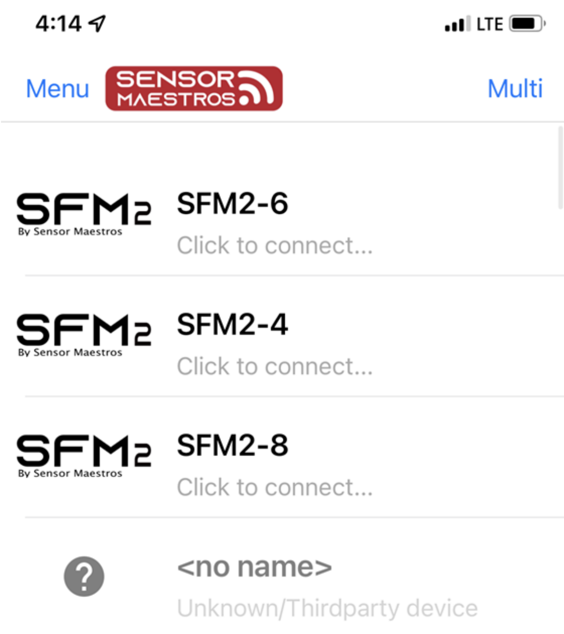
19.0 Mobile App

The SFM2 has both an Android and an iOS App that can be used to Configure, Test, and Data-log the SFM2. Features of the SFM2 Mobile App are listed below.

- Allow multiple SFM2's to be connected simultaneously
- Provides 4 Standard SFM2 configurations:
  - OFF
  - Low Power (ACC=26Hz, GYRO=26Hz, MAG=26Hz, SFOR=26Hz)
  - Balanced (ACC=104Hz, GYRO=104Hz, MAG=104Hz, SFOR=104Hz)
  - Performance (ACC=833Hz, GYRO=833Hz, MAG=104Hz, SFOR=417Hz)
- 3D Cube Representation: Sensor Fusion Output for SFQTE and SFLA
  - Tare SFM2 from 3D View and Manual rotate image of SFM2 to match real-world positioning of SFM2
- BLE & COM Port Throughput Testing
- SFM2 Self-Test
- Individual Sensor Configuration
  - Including Filter Settings for Accel, Gyro. Mag
- Individual Data Enable for each Sensor Stream
  - Both Raw Sensor Output and Sensor Fusion Outputs
- Global Reference Enable/Disable
- Headtare Adjustment: Allows for Tare Offset
- Tare SFM2
- Manual Calibration of SFM2
- Manual Kalman Filter Tuning
- Renaming of individual SFM2
- Store Current Configuration
- Restore Factory Defaults
- Real-Time Graphing for Pressure, Temperature, Humidity, and Altitude
- Display of real-time stream data
- BLE and COM Port Stats
- TIME and TOFFSET Settings
- Data Log to csv file

19.1 Scanner

SFM2 App will show all SFM2's that are active(ON) and in the range of the mobile device. Tap on any of the SFM2's in the list to connect to it.



19.2 Change Name

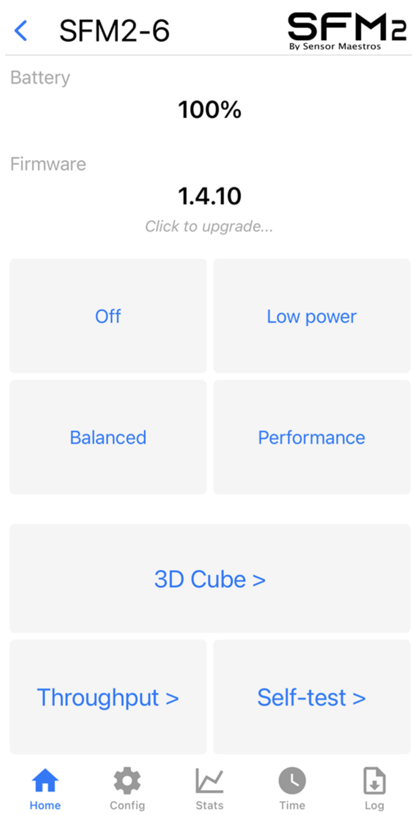
Give a SFM2 a unique name by connecting to it going to the 'Config' tab and scroll to the bottom section to rename the device. After changing the name of the SFM2 the name will immediately take effect and is stored in non-volatile memory and will display as that name until the name would be changed again.

19.3 Home

The 'Home' view provides the user a quick means to configure the SFM2 as shown below and ability to navigate to other views.

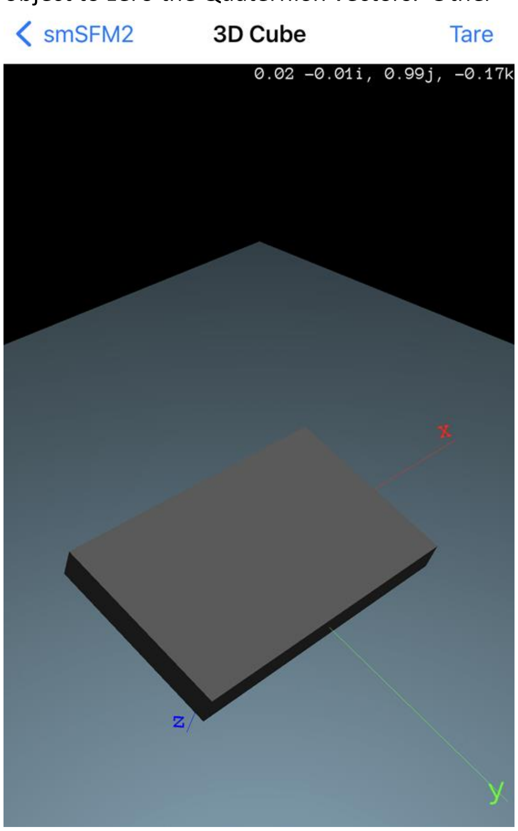
- 4 Performance Modes

Mode	ACC/GYR/MAG (Hz)	SF ODR Div	Pressure Sensor(Hz)	Temp/Hum Interval(sec)	DataStream Enabled
OFF	0	1	0	0	NONE
Low Power	26	1	1	2	SFQT
Balanced	208	1	10	1	SFQT
Performance	833	1	75	Continuous	SFQT



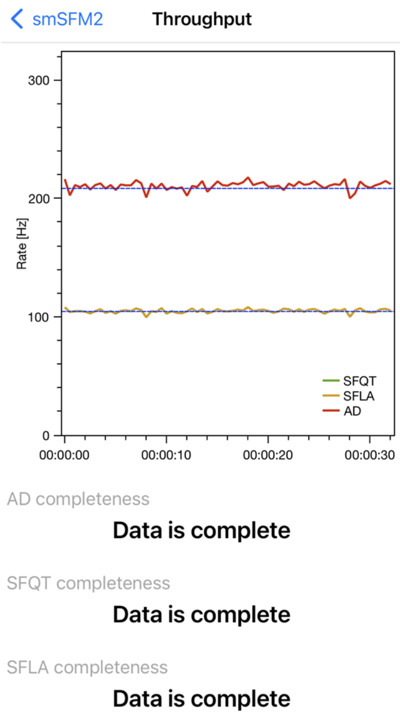
19.3.1 Home: 3D View

The 3D view allows a user to see the SFM2 rotate in 3D space and provide the ability to 'Tare' the SFM2 if it has been mounted to an object to zero the Quaternion vectors. Other



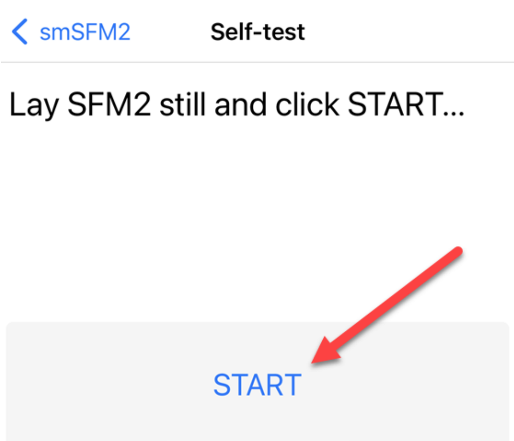
19.3.2 Home: Throughput

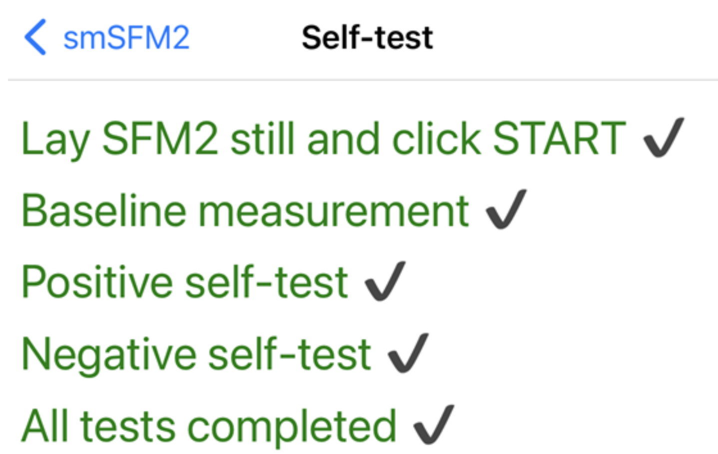
The Throughput view from the Home view can be used to quickly get an understanding if the data throughput is sufficient for the data streams that have been enabled at the particular ODR rates. Shown below the SFQT(tared quaternion) and SFLA(linear acceleration) and Raw Accelerometer Data Streams have all been enabled. NOTE: The Accelerometer has been configured for 208Hz ODR while the Fusion Library is running at 104Hz.



19.3.3 Home: Self-Test

The Self-Test on the Home view allows the user to quickly perform a sensor functionality test to ensure the SFM2 is functioning properly.





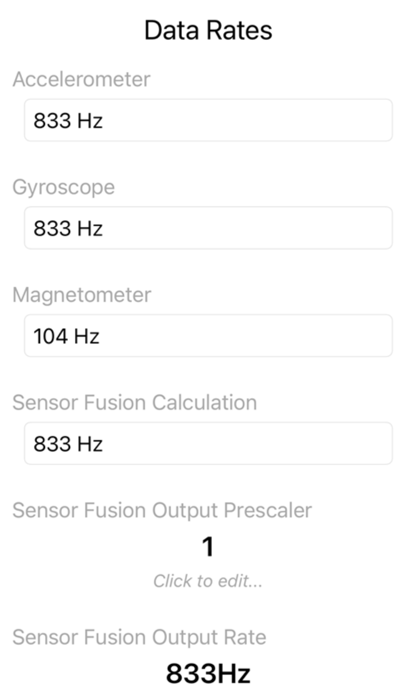
19.4 Config

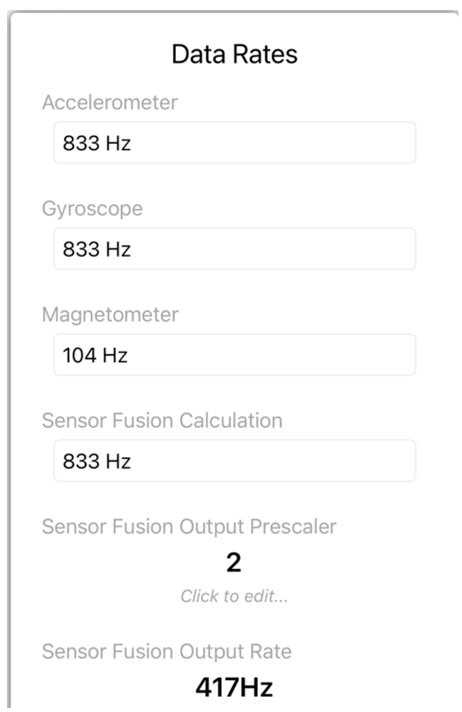
The Config view allows the following configurations

- Data Rates: Accel, Gyro, Magnetometer, Sensor Fusion
- Sensor Fusion Output Divider (allows SF to run at a higher data rate than output data stream, helps reduce BLE Bandwidth)
- Enable Datastreams:
  - Raw Sensor Data: Accel, Gyro, Mag
  - Calibrated Mag Data
  - Sensor Fusion Outputs: SFQ(untared quaternion), SFQT(tared quaternion), SFEA(Euler Angles), SFCHT(Heading, Tilt), SFLA(linear acceleration)
  - Environmental Sensors: Pressure, Altitude, Temperature, Humidity
- Change the Reference Frame
- Configure Temperature/Humidity Sensor
- Configure Pressure/Altitude Sensor
- Configure SFM2 Calibration Settings
- Configure Accelerometer
- Configure Gyroscope
- Configure Magnetometer
- Configure Kalman Filter Parameters
- Change SFM2 Name
- Store User Defined Configurations, Load Factory Defaults

19.4.1 Config: Data Rates

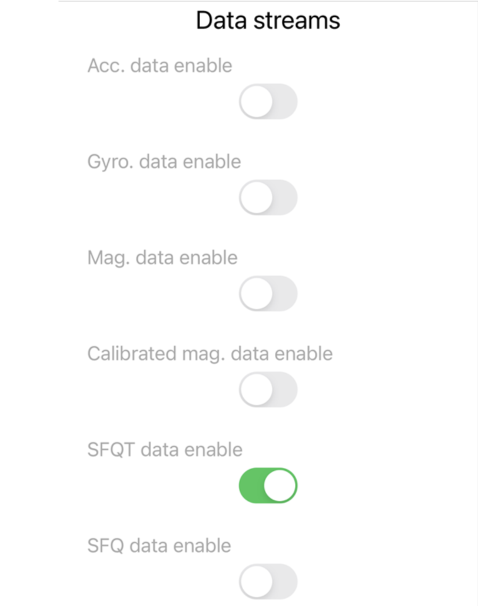
Configure Data rates for the Accel/Gyro/Mag and Sensor Fusion Output. Example shown below.
To reduce the BLE throughput you can use the ‘Sensor Fusion Output Prescaler’ and set to an integer value. This will divide the Fusion ODR by this value example shown below.





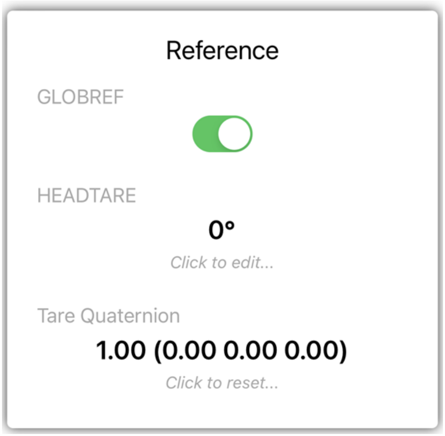
19.4.2 Config: Data Streams

Enable individual data streams desired to be in SFM2 ouptut. Not all data stream enables are shown in image below…



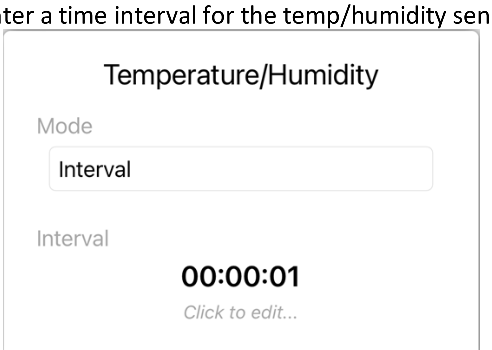
19.4.3 Config: Reference

Enabled Disable Global Reference Frame. Refer to Section 14 for further details on Sensor Fusion Reference Frames.



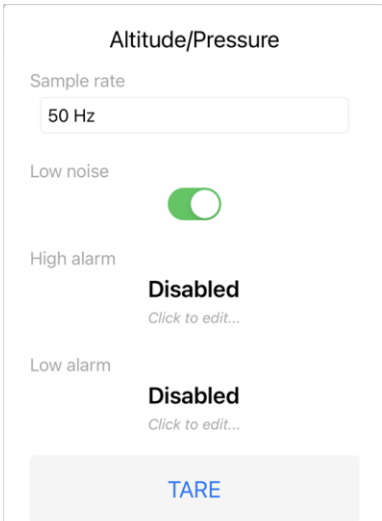
19.4.4 Config: Temperature/Humidity

The Temperature/Humidity sensor can be configured to operate in a continuous mode or in an interval mode. Interval mode allows user to enter a time interval for the temp/humidity sensor to be sampled.



19.4.5 Config: Altitude/Pressure

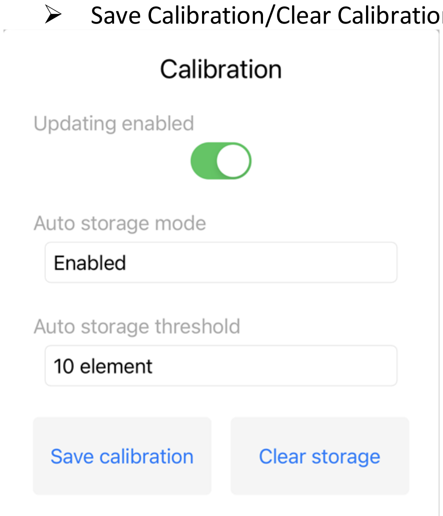
- Configure Altitude/Pressure sensor sample rate
- Enable/Disable low noise operation
- Configure High/Low Alarms
    - Allows a High and/or Low Alarm to be set. Graph will flash RED if Altitude has passed threshold
- 'Tare' to zero altitude.
    - Zero Altitude to measure change of altitude at current altitude



19.4.6 Config: Calibration

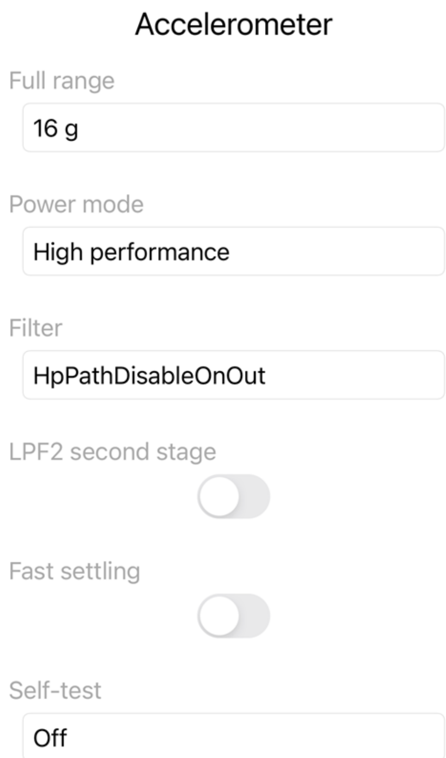
The Calibration section allows the user to enable calibration for the SFM2. The recommended settings for Calibration are shown below.

- Updating Enabled: Allows for dynamic calibration updates
- Auto Storage Mode: Looks for and updates any new calibration values that provide a better match than current values.
- Auto Storage Threshold: Looks for either 7 or 10 parameter calibration
- Save Calibration/Clear Calibration: Save current calibration into NV Memory(uses on POR), clear current Cal values



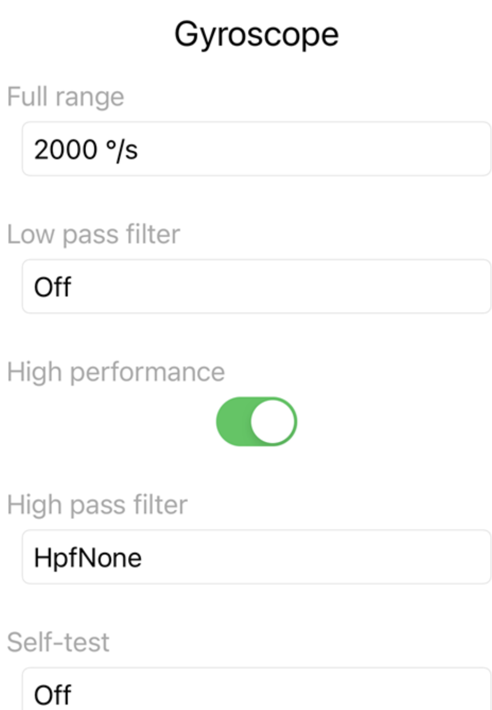
19.4.7 Config: Accelerometer

Provides for configuration of all Accelerometer parameters. For typical usage recommended settings are shown below. For setting options please refer to Section 16.5 Accelerometer Settings 0x0201



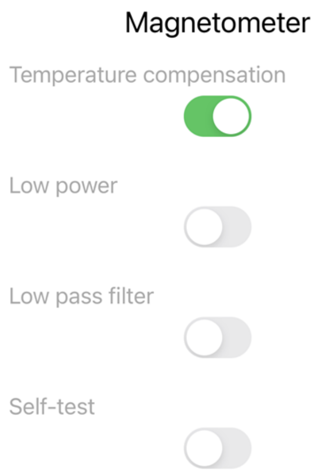
19.4.8 Config: Gyroscope

Provides for configuration of all Gyroscope parameters. For typical usage recommended settings are shown below. For setting options please refer to Section 16.6 Gyroscope Settings 0x0202



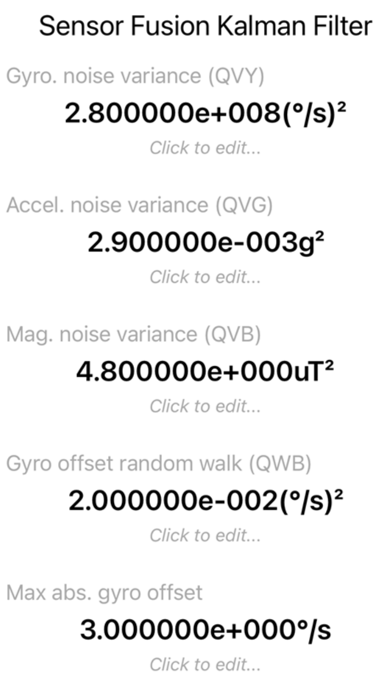
19.4.9 Config: Magnetometer

Provides for configuration of all Magnetometer parameters. For typical usage recommended settings are shown below. For setting options please refer to Section 16.7 Magnetometer Settings 0x0203



19.4.10 Config: Kalman Filter Parameters

The SFM2 allows for Kalman Filter parameter tunings which can be useful for specific application to fine tune for specific movement or conditions. It is NOT recommended to change the parameters that have been configured for the SFM2 from the factory calibration unless these variables are clearly understood. All SFM2's are shipped with a factory calibration including Kalman filter tuning. Typical Kalman filter tunings are shown below.

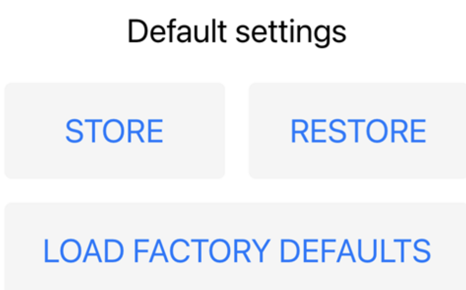


19.4.11 Config: Name Change

Changing the name is already discussed in Section 17.2 Change Name. Use to change name displayed for SFM2 connected to SENSR-LOGR.

19.4.12 Config: Default Settings

Allows user to store in Non-Volatile memory the current SFM2 configuration. If the user stores current configuration the SFM2 will power ON with the stored configuration. Allows user to restore Factory Defaults.



19.5 Stats

The Stats view is SFM2 data streams are shown and scrolling graphs for Environmental Sensors is shown along with system information.

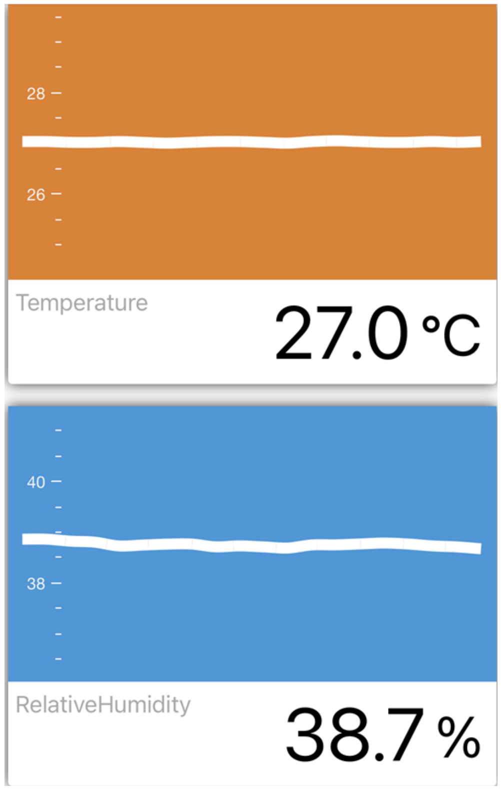
- Enabled Data Streams displayed, Scrolling graphs for Environmental Sensors if data streams are enabled.
- BLE Stats
- Serial Port Stats
- Current/Stored Calibration information
- Device Statistics
- BLE Connection Parameters

19.5.1 Data Streams

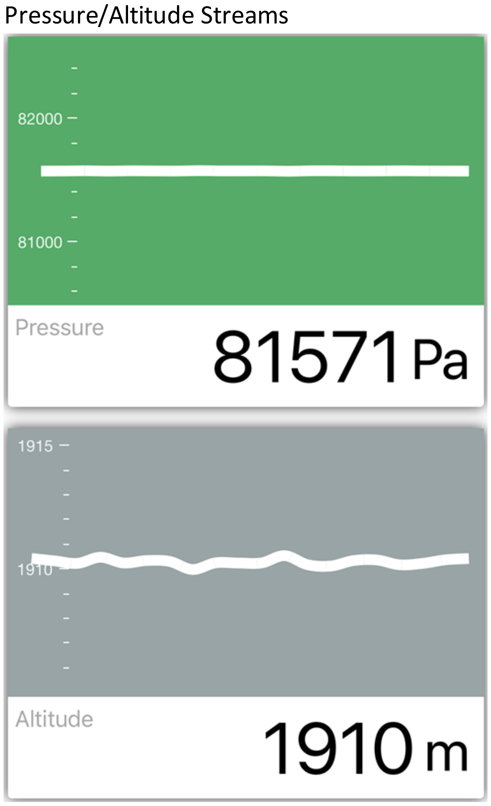
Data Streams section will display all active data streams for motion sensors, sensor fusion, and environmental sensors similar to as shown below.

Motion Sensor/Sensor Fusion Streams

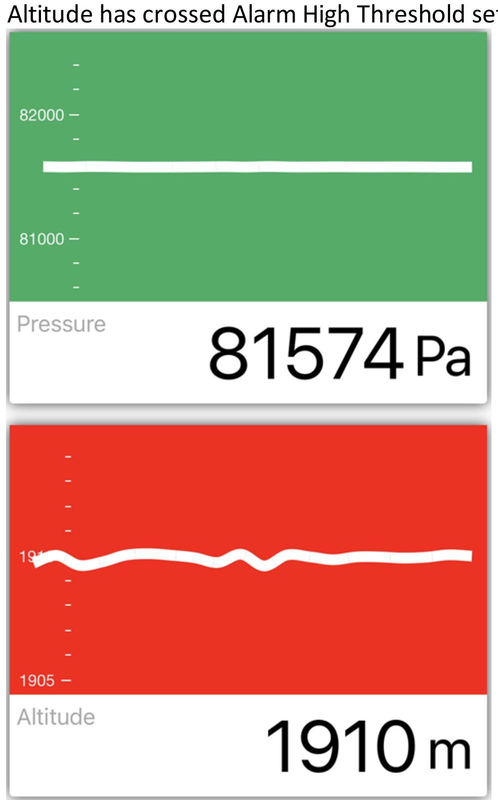
Temperature/Humidity Streams



Pressure/Altitude Streams

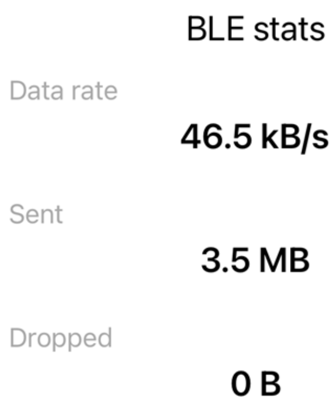


Altitude has crossed Alarm High Threshold set for 1907m



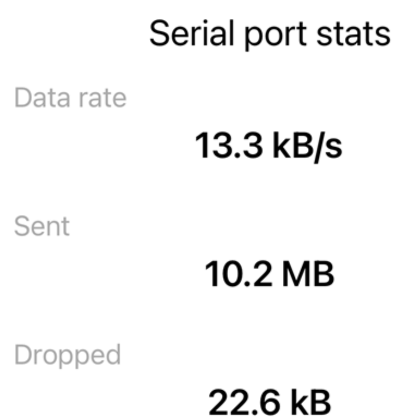
19.5.2 BLE Stats

BLE Stats provides current BLE throughput with data streams that are currently enabled along with a history of the amount of data that has been sent and the number of bytes dropped if any.



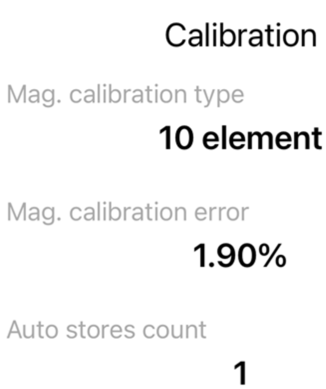
19.5.3 Serial Port Stats

Serial Port Stats provides COM port statistics as shown below.



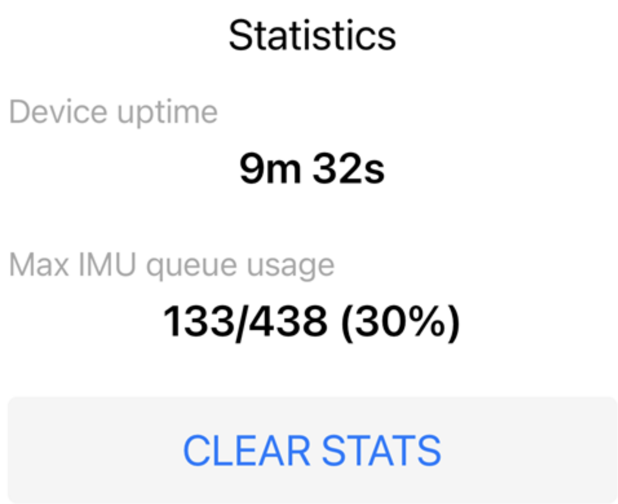
19.5.4 Calibration

Calibration Stats indicate current calibration values and error currently being used. If these are blank then the SFM2 has not been properly calibrated OR possibly the Calibration values have been cleared via the CONFIG->Calibration 'Clear Storage' button and the device has been power cycled where it does not have calibration values to use. If this is the case please refer to Section 15.0 Manual Calibration.



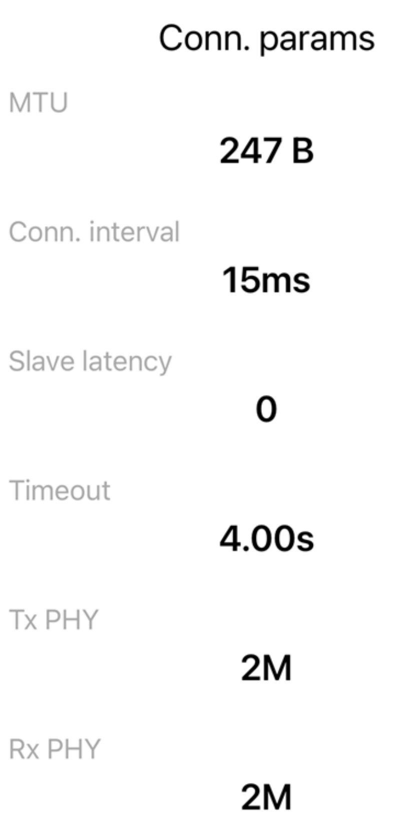
19.5.5 Statistics

The Statistics section provides high level system statistics indicating how long the SFM2 has been connected to the SENSR-LOGR app. The Max IMU queue usage provides an indication as to how much BLE bandwidth is available on the SFM2 side.



19.5.6 Connection Params

Connection Params show the current BLE Connection parameters being used by the SENSR-LOGR/mobile device. The SFM2 will always attempt to establish the connection with the parameters shown below. Some older mobile devices may not support the MTU size of 247bytes and/or 2Mb PHY connections. If these connection parameters are not supported on your mobile device the BLE throughput will be significantly reduced.

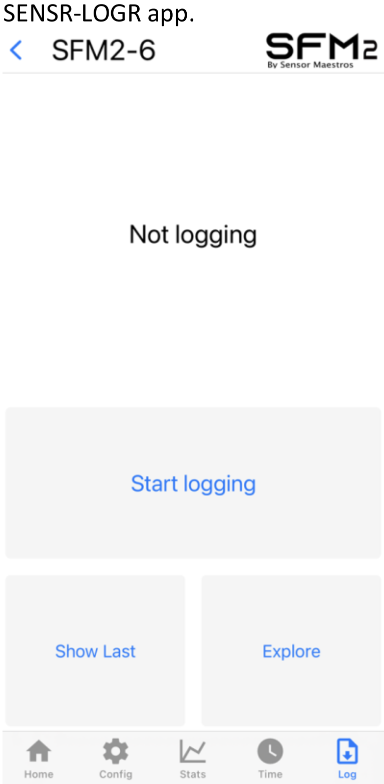


19.6 Time

When ONLY 1 SFM2 is being used at a time the 'Time' view and 'Synchronization' feature should NOT be used. This is ONLY for use when multiple SFM2's are in use simultaneously connected to the SENSR-LOGR app. For this case refer to Section 18.8.3 Multi Mode: Time Synchronization. Future versions of the SENSR-LOGR app will NOT display the Time view when only 1 SFM2 is in use/connected to the SENSR-LOGR app.

19.7 Log

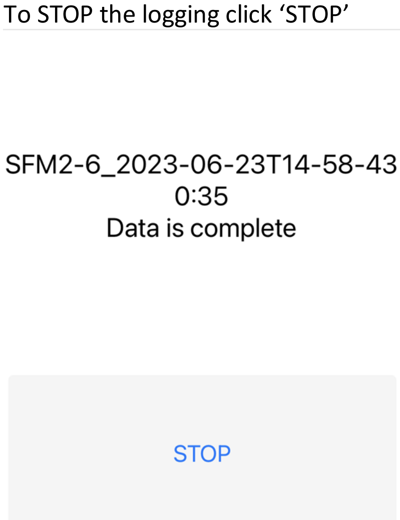
The Log view allows data streams to be logged to a CSV file on the mobile device. These CSV files can then be 'Shared' by sending them via email, pushing them to a cloud storage location, etc. The log files can also be clicked on to have them be graphed via the SENSR-LOGR app.



19.7.1 Start Logging

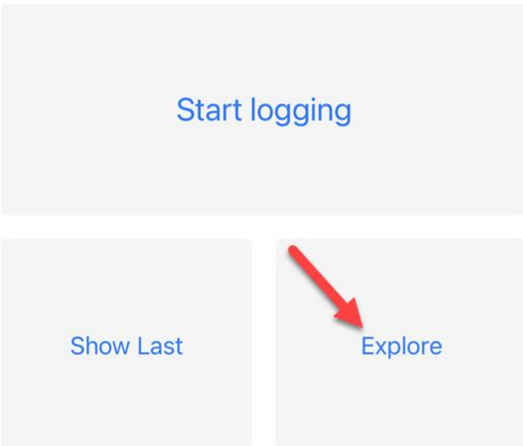
To start data logging to the mobile device ensure you have the Data Streams enabled that you want to be logged and simply click the 'Start Logging button and you will see the view shown below.

To STOP the logging click 'STOP'



19.7.2 View/Explore/Garph Log Files

To view your log files click 'Explore'

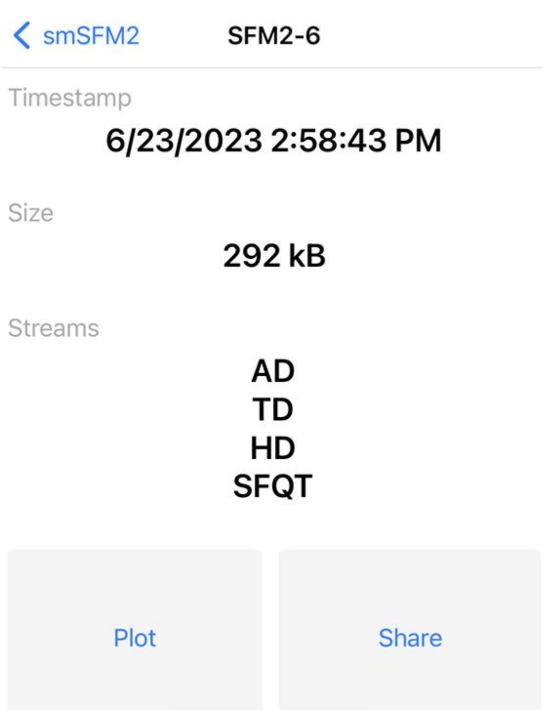


All stored log files in the SENSR-LOGR app will show as seen below

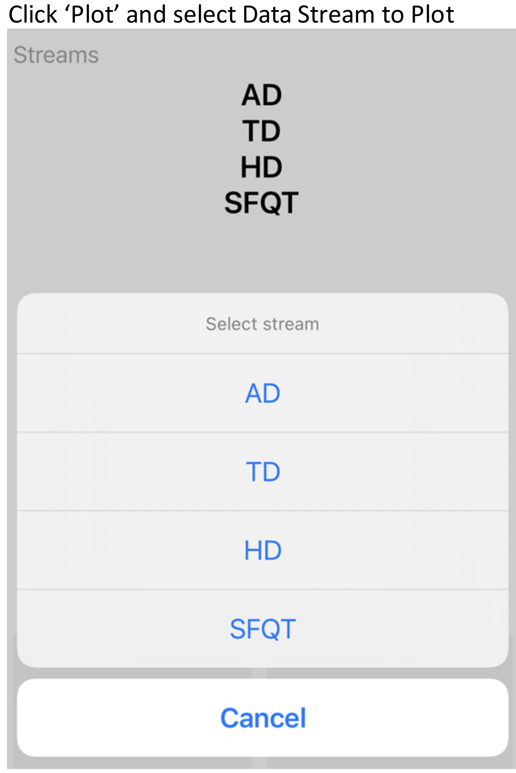
The 'Show Last' button brings up the most recent Log file.

To graph or see the data streams inside a Log file click on the desired Log file

The log file shown below has the AD(accelerometer), TD(Temperature), HD(Humidity), and SFQT(Tared Quaternion) data streams contained in it.

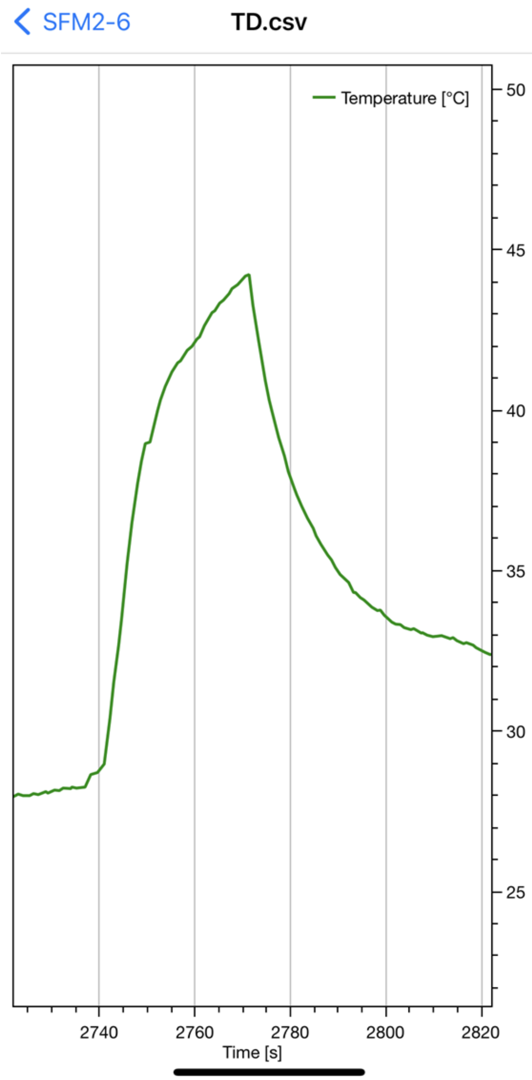


Click 'Plot' and select Data Stream to Plot

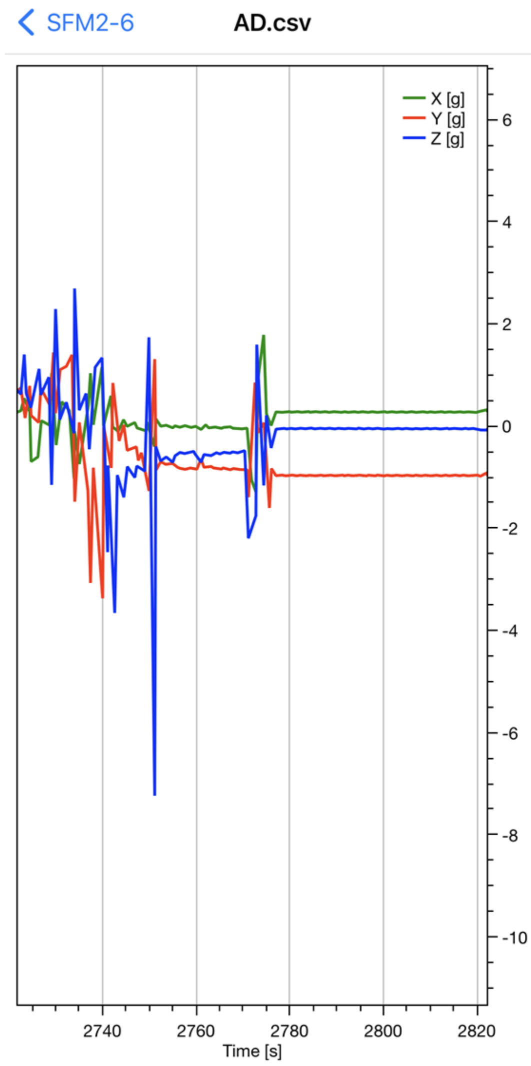


Example plots shown on next page…

Temperature Plot

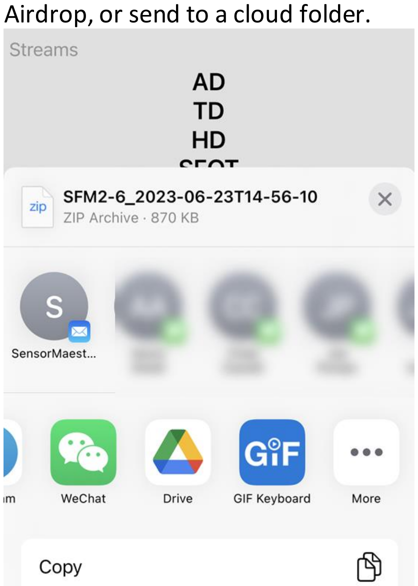


Accelerometer Plot



19.7.3 Share Log File(store remotely)

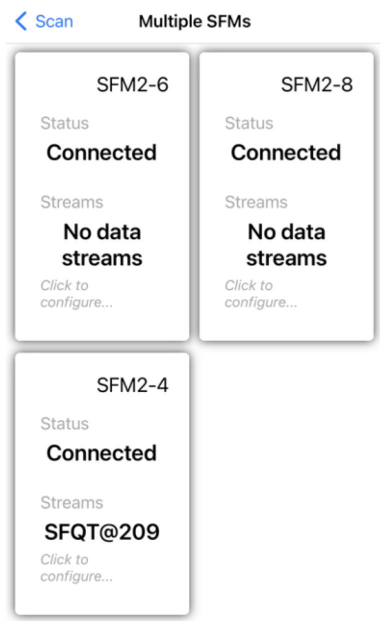
Stored log files can be sent to a remote location using the 'Share' button which provides the ability to send the log files via text, email, Airdrop, or send to a cloud folder.



19.8 Multi Mode(multiple SFM2's)

The SENSR-LOGR app and SFM2 firmware allows for multiple SFM2's to be connected to the app simultaneously and be synchronized together to allow for time multiple SFM2's to be used and have the data streams be synchronized in time. A typical mult-SFM2 use would be to monitor human body motion in which SFM2's would be worn on both wrists and both ankles and possibly one on the chest. While it is possibly to connect up to 6 SFM2's to the SENSR-LOGR app it is not typically recommended as mobile devices tend to start to limit the bandwidth they will allocated once multiple BLE devices have been connected. Sensor Maestros has tested up to 5 devices on newer, high performance iOS and Android mobile devices with success supporting Data Rates in the 40-45kB/s range. Most mobile devices we have tested start to shows noticeable dropped bytes when multiple devices are connected and overall data rates are above 40-45kB/s. This is highly dependent on the specific mobile device and each mobile device will perform differently. For muli-SFM2 use the specific mobile devices should be tested to verify throughput capability.

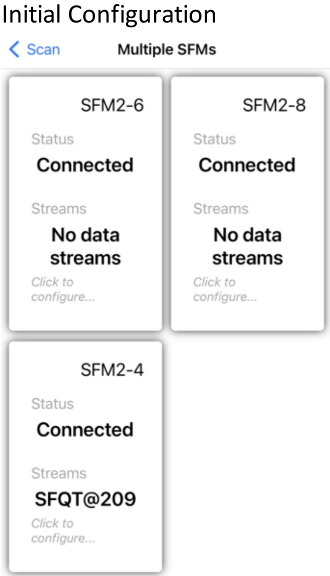
To use multiple SFM2's click on the 'Multi' button and it will connect to each SFM2 it sees in the SCAN window. Afterwards you will see the Multi View and you can click on the individual SFM2's to configure each one individually. OR you can configure them all together as is shown further below…



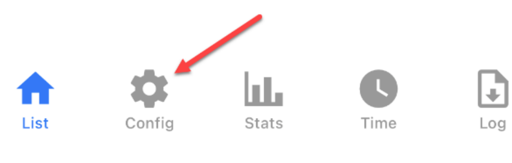
19.8.1 Multi Mode: Config

When using Multiple SFM2's you can configure them to all use the same configuration by using the 'Config' menu in MultiView mode. ***Example:*** In this case we had the SFM2-4 configured to output SFQT(tared quaternion) at a 208Hz output. We want to change all 3 SFM2's to output SFLA(Linear Acceleration) at a 417Hz data rate. This can be done in a single instance using the 'Config' settings.

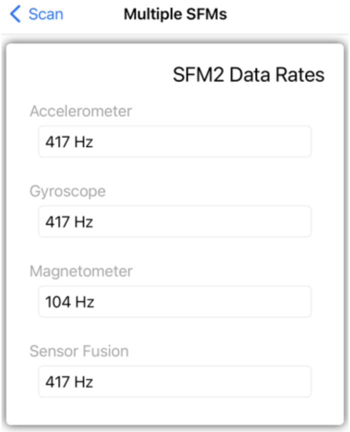
Initial Configuration



Select 'Config'

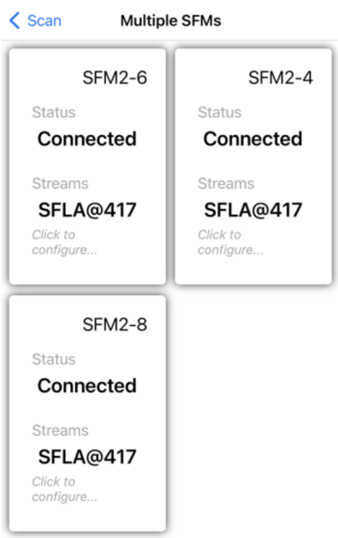


Configure ACC/GYR/SF = 417Hz MAG=104Hz



Scroll down if Data Streams are not in view and select SFLA Data Enable

Return to List View and you will see all 3 SFM2's have been configured the same



19.8.2 Multi Mode: Stats

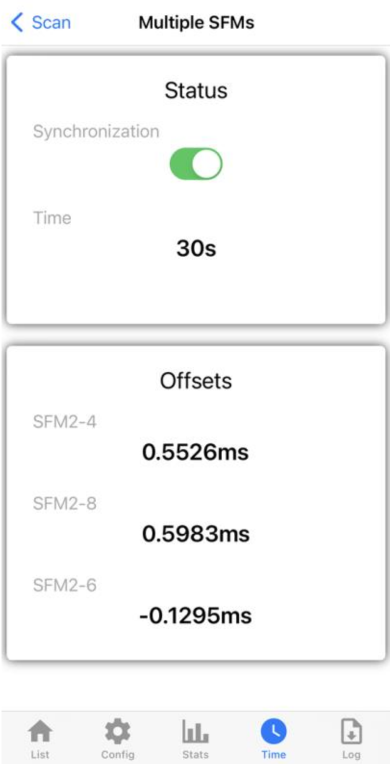
If you have multiple SFM2's configured and connected to the SENSR-LOGR app you can view the Data Streams on the 'Stats' tab as shown below.

19.8.3 Multi Mode: Time Synchronization

The SENSR-LOGR app and SFM2 firmware can operate together to provide accurate time synchronized data between multiple SFM2's. You can easily use this feature if you are using Multi Mode by going to the 'Time' view and selecting 'Synchronize'. This provides the ability for multiple SFM2's to be synchronized in time providing data stream outputs that have time synchronized timestamps. This allows data to be analyzed from multiple SFM2 and be synchronized in time to provide a an entire time-synchronized view of the person or object that is being monitored by the SFM2.

This time synchronization algorithm is constantly being monitored between all SFM2's in use in Multi Mode and constantly updates as needed each SFM2's 'time stamp clock' to keep the SFM2's in Synch. This process is not trivial by any means as one can imagine there are not only clocks on each SFM2 that may differ but also understanding that there are differences between each SFM2 and data sent/received. While this algorithm is proprietary it can be said that a highly accurate clock is used to effectively trim and update clocks on each SFM2 and the SENSR-LOGR app helps coordinate all this activity and synchronization.

It can be observed in a Multi Mode Data log that each of the SFM2's timestamps are closely synchronized using this functionality. A typical data log is shown below for having 3 SFM2's connected after Synchronization has been enabled as shown below.



Example of LOG file with 3 SFM2's connected to the SENSR-LOGR app on next page…

19.8.3.1 Time Synchronization Log File File

Below is an example of 3 SFM2's connected to the SENSR-LOGR app with Fusion Output set to 417Hz with the Output Prescaler set to 2 which provides a 208Hz ODR and the SFQT Datastream enabled for each of the SFM2's.
NOTE: For the sake of space the table below is only showing the W component of a Quaternion output.

SFM #	Time [s]	W	SFM #	Time [s]	W	SFM #	Time [s]	W	STD Deviation
SFM1	5.9926726	0.60209	SFM2	5.9988324	0.63320	SFM3	6.0038994	0.20222	5.6223
SFM1	5.9975040	0.60209	SFM2	6.0036503	0.63161	SFM3	6.0088063	0.20132	5.6584
SFM1	6.0023354	0.60209	SFM2	6.0084682	0.63274	SFM3	6.0137131	0.20199	5.6946
SFM1	6.0071667	0.60210	SFM2	6.0132861	0.63464	SFM3	6.0186200	0.20005	5.7311
SFM1	6.0119981	0.60210	SFM2	6.0181040	0.63207	SFM3	6.0235268	0.19763	5.7677
SFM1	6.0168295	0.60211	SFM2	6.0229219	0.63187	SFM3	6.0284337	0.19650	5.8045
SFM1	6.0216609	0.60210	SFM2	6.0277398	0.63020	SFM3	6.0333405	0.19800	5.8414
SFM1	6.0264923	0.60209	SFM2	6.0325576	0.63049	SFM3	6.0382474	0.20169	5.8785
SFM1	6.0313236	0.60209	SFM2	6.0373755	0.62839	SFM3	6.0431542	0.20448	5.9158
SFM1	6.0361550	0.60208	SFM2	6.0421934	0.62807	SFM3	6.0480610	0.20565	5.9532
SFM1	6.0409864	0.60209	SFM2	6.0470113	0.62872	SFM3	6.0529679	0.20726	5.9908
SFM1	6.0458178	0.60209	SFM2	6.0518292	0.63102	SFM3	6.0578747	0.20702	6.0285
SFM1	6.0506491	0.60210	SFM2	6.0566471	0.63444	SFM3	6.0627816	0.20569	6.0664
SFM1	6.0554805	0.60209	SFM2	6.0614650	0.63429	SFM3	6.0676884	0.20112	6.1043
SFM1	6.0603119	0.60208	SFM2	6.0662829	0.63495	SFM3	6.0725953	0.19847	6.1425
SFM1	6.0651433	0.60208	SFM2	6.0711008	0.63347	SFM3	6.0775021	0.19880	6.1807
SFM1	6.0699746	0.60208	SFM2	6.0759186	0.63319	SFM3	6.0824090	0.19734	6.2192
SFM1	6.0748060	0.60210	SFM2	6.0807365	0.63088	SFM3	6.0873158	0.19973	6.2577
SFM1	6.0796374	0.60210	SFM2	6.0855544	0.63086	SFM3	6.0922227	0.20064	6.2964
SFM1	6.0844688	0.60210	SFM2	6.0903723	0.63040	SFM3	6.0971295	0.20220	6.3351
SFM1	6.0893001	0.60209	SFM2	6.0951902	0.63084	SFM3	6.1020363	0.20337	6.3741
SFM1	6.0941315	0.60208	SFM2	6.1000081	0.63253	SFM3	6.1069432	0.20317	6.4131
SFM1	6.0989629	0.60208	SFM2	6.1048260	0.63293	SFM3	6.1118500	0.20419	6.4523
SFM1	6.1037943	0.60208	SFM2	6.1096439	0.63253	SFM3	6.1167569	0.20642	6.4916
SFM1	6.1086257	0.60207	SFM2	6.1144618	0.63242	SFM3	6.1216637	0.20655	6.5309
SFM1	6.1134570	0.60207	SFM2	6.1192796	0.63205	SFM3	6.1265706	0.20578	6.5705
SFM1	6.1182884	0.60207	SFM2	6.1240975	0.63105	SFM3	6.1314774	0.20505	6.6101
SFM1	6.1231198	0.60206	SFM2	6.1289154	0.62913	SFM3	6.1363843	0.20044	6.6498
SFM1	6.1279512	0.60205	SFM2	6.1337333	0.62766	SFM3	6.1412911	0.19917	6.6896
SFM1	6.1327825	0.60205	SFM2	6.1385512	0.62731	SFM3	6.1461979	0.20132	6.7296
				AVG Deviation all 30 sample periods(ms)					0.3365
				AVG of 30 STD Deviation Values(ms)					6.1667

20.0 SFM2 BLE Services and Characteristics

All custom service and characteristic UUIDs are 128-bit UUIDs of the form
71D3xxxx-E8E7-4F91-AA3C-4A68051247BC, with unique 16-bit values replacing the "xxxx" in bytes 2 and 3.
In the descriptions below, only the unique 16 bits of the UUIDs are given, between hyphens, expressed in hexadecimal, for example, -0102-represents 71D3***0102***-E8E7-4F91-AA3C-4A68051247BC.

Bluetooth specification, which is not explicitly listed below.
https://developer.bluetooth.org/gatt/descriptors/Pages/DescriptorViewer.aspx?u=org.bluetooth.descriptor.gatt.client_characteristic_configuration.xml
Indications may optionally be supported.
Note: the standard Bluetooth Device Information Service must be implemented to provide the identification strings and hardware/firmware/software versions.
https://developer.bluetooth.org/gatt/services/Pages/ServiceViewer.aspx?u=org.bluetooth.service.device_information.xml

20.1 SFM2 BLE Characteristic Map

The SFM2 has a total of 19 characteristics, 16 of them support notifications.

Name	ID	Read	Notify	Write
Data_Stream	0x0101		X	
Data_Stream_Settings	0x0102	X	X	X
Sensor_Fusion_Settings	0x0122	X	X	X
Reference_settings	0x0128	X	X	X
Tare	0x0129			X
Time	0x0131	X	X	X
Time_Offset	0x0132	X	X	
Time_Trim	0x0142	X	X	X
Time_Notify_Interval	0x0143	X	X	X
Name	0x0133	X	X	X
Settings_Storage	0x0134			X
Calibration_Storage	0x0135			X
Stats	0x0137	X	X	X
Connection_Parameters	0x0138	X	X	
Accelerometer_Settings	0x0201	X	X	X
Gyroscope_Settings	0x0202	X	X	X
Magnetometer_Settings	0x0203	X	X	X
SF_Kalman_Settings	0x0204	X	X	X
Calibration	0x0206	X	X	
Calibration Settings	0x0207	X	X	X
Self-Test	0x0208	X	X	X
Env._Sensors_Settings	0x0310	X	X	X
Altitude Tare	0x0311			X

20.2 Data Stream Characteristics

The Data Stream characteristic is used to send sensor data from the SFM2. The transmission is started by enabling BLE notifications of the *Data Stream characteristic* (writing its CCCD). The type of sensor data to be sent can be chosen by enabling the various sensor data types via the Data Stream Settings characteristic as shown further below.

20.2.1 Data Stream 0x0101

The data stream characteristic sends data in the binary frame format (see SFM2 - Binary frame document). The characteristic supports notifications only, each notification contains a chunk of data. The data chunks are not aligned in any way, they should be treated as if received from a serial port. The client is responsible for stitching the chunks together and dividing it into frames.

20.2.2 Data Stream Settings 0x0102

Data Stream Settings allow for individualized control of what sensor data is included in the Data Stream itself. It also allows for individualized control of the USB output to be Binary or ASCII.

Size		Name	Description
15 bytes	1 byte	binmode	Binary data mode. The BLE interface ALWAYS uses the Binary Frame format. While this bit can be changed from the USB interface the BLE Data streams are always sent using the Binary Frame method. X – Don't Care
	1 byte	ade	Enables accelerometer (AD) data. 1-enabled, 0-disabled
	1 byte	gde	Enables gyroscope (GD) data. 1-enabled, 0-disabled
	1 byte	mde	Enables magnetometer (MD) data. 1-enabled, 0-disabled
	1 byte	sfqde	Enables untared quaternion (SFQ) data. 1-enabled, 0-disabled
	1 byte	sfqtde	Enables tared quaternion (SFQT) data. 1-enabled, 0-disabled
	1 byte	sflade	Enables linear acceleration (SFLA) data. 1-enabled, 0-disabled
	1 byte	sfeade	Enables Euler angles (SFLA) data. 1-enabled, 0-disabled
	1 byte	sfchtde	Enables Heading-Tilt (SFCHT) data. 1-enabled, 0-disabled
	1 byte	sfmde	Enables calibrated magnetometer (SFM) data. 1-enabled, 0-disabled
	1 byte	pde	Enables pressure (PD) data. 1-enabled, 0-disabled
	1 byte	altde	Enables altitude (ALT) data. 1-enabled, 0-disabled
	1 byte	tde	Enables temperature (TD) data. 1-enabled, 0-disabled
	1 byte	hde	Enables humidity (HD) data. 1-enabled, 0-disabled

	1 byte	tsde	Enables time sync (TS) data. Enables data timestamps(25us resolution) to be synchronized with a more accurate 32.768kHz real time clock and corrected as needed. 1-enabled, 0-disabled **NOTE: SENSR-LOGR app sets this bit by default upon connecting to a SFM2.**

20.2.2.1 TSDE(Timestamp Syncrhonization)

Setting TSDE to 1 enables the RTC clock to be output in the Binary Frame along with the sensor data that is packed into the binary frame. By default the SENSR-LOGR app will set TSDE=1 upon connecting to a SFM2. This is extremely useful as the default timestamp that is provided by the LSM6DSx IMU which provides 25us resolution can drift significantly over time. When the RTC Tick, based on a 32.768kHz clock, is placed in the binary stream the BLE client can validate if the integer timestamps provided by the LSM6DSx should be adjusted slightly one way or the other by providing a highly accurate reference clock for the timestamps provided by the LSM6DSx. The timestamp adjustment is completely handled by the BLE Client(most likely a mobile app) such as the SENSR-LOGR. The SENSR-LOGR app already handles this and adjusts the timestamps that are located in LOG files for examples to ensure they are accurate in reference to the 32.768kHz reference clock.

An example of RTC output in the binary stream is shown below which could apply for BLE or USB connection.
Suppose that AD stream is enabled @208Hz rate and TS data is turned on.

We receive the following binary frames:

Frame idx	Samples	LSM6DSx clock (25us tick)
1	AD	100000
2	AD	100192 (Sample Delta: 192 ticks * 25us = 4.8ms or 208Hz)
3	AD + TS = 630	100384
4	AD	100576
5	AD	100768
6	AD	100960
7	AD + TS = 1260	101152
8	AD	101344
9	AD	101536
10	AD	101728
11	AD + TS = 1890	101920

The SENSR-LOGR app then uses the TS data to calculate the RTC timestamps corresponding to the remaining AD samples. The timestamps shown in LOG file from the SENSR-LOGR app are already calculated to an actual time value versus being displayed in LSM6DSx 'ticks'.

NOTE: This timestamp synchronization is completely different than synchronizing multiple SFM2's to each other. Synchronizing multiple SFM2's does leverage the highly accurate 32.768kHz clock but there are several additional items that are monitored and adjusted if necessary to synchronize multiple SFM2's.

20.3 Sensor Fusion settings 0x0122

Sensor Fusion Settings provide control over the Fusion Output Data Rate(ODR) and also over the Output Divider setting which allows the Fusion ODR to run at a higher rate than the actual Data Rate that is sent to the BLE and USB interfaces. EXAMPLE – User can run the Fusion Library at 833Hz with the Output Divider set to 2 and the data rate sent to the BLE & USB interfaces is only 833Hz/2. This effectively provides a means to oversample the Sensor Fusion Data.

Size	Name	Description
1 byte	rate	Data_rate
1 byte	Output Divider	Divides Fusion ODR Data Rate to BLE or USB interface

20.3.1 Data Rate

Data rate is a single byte setting that is used to configure data rate.

Value	Rate	Value	Rate
0x00	0 Hz	0x05	208 Hz
0x01	12.5 Hz	0x06	417 Hz
0x02	26 Hz	0x07	833 Hz
0x03	52 Hz	0x08	1667 Hz
0x04	104 Hz		

20.3.2 Fusion ODR Output Divider

The Fusion ODR Output Divider can be set to scale the actual Fusion ODR that is sent across the BLE or USB interface. This effectively provides a means to run the Fusion Library in an 'Oversampling' mode while allowing a slower data rate to the BLE or USB interface.

Example – Configure the Fusion Data Rate to 833Hz(0x07)
Fusion ODR Output Divider = 2, yields 833Hz/2 data rate for the BLE or USB data stream. Fusion ODR Output Divider = 4, yields 833Hz/4 data rate for the BLE or USB data stream.

Size	Name	Description
1 byte	Fusion ODR Output Divider	Integer value between 1 and 255

20.4 Reference Settings 0x0128

Size		Name	Description
17 bytes	4 bytes	w	The real component of tare quaternion, Float32
	4 bytes	x	The i vector component of tare quaternion, Float32
	4 bytes	y	The j vector component of tare quaternion, Float32
	4 bytes	z	The k vector component of tare quaternion, Float32
	1 byte	globref	Global reference frame. 1-enabled, 0-disabled

20.5 Accelerometer Settings 0x0201

Size		Name	Description
7 bytes	1 byte	rate	Data_Rate
	1 byte	full scale	Full_Scale
	1 byte	lpf2	LPF2 second stage selected, 1-enabled, 0-disabled
	1 bytes	filter	Filter_Configuration
	1 byte	fast_settling	Filter fast settling. 1-enabled, 0-disabled
	1 byte	power_mode	Power_Mode
	1 byte	self_test	Self-Test Disturbance Configuration

20.5.1 Accelerometer Full Scale

Used to configure full scale of 3D accelerometer

Value	Rate
0x00	2 G
0x01	4 G
0x02	8 G
0x03	16 G

20.5.2 Accelerometer Filter Configuration

Value	Filter	Value	Filter
0x00	HP_PATH_DISABLE_ON_OUT	0x34	HP_REF_MD_ODR_DIV_100
0x10	SLOPE_ODR_DIV_4	0x35	HP_REF_MD_ODR_DIV_200
0x11	HP_ODR_DIV_10	0x36	HP_REF_MD_ODR_DIV_400
0x12	HP_ODR_DIV_20	0x37	HP_REF_MD_ODR_DIV_800
0x13	HP_ODR_DIV_45	0x01	LP_ODR_DIV_10
0x14	HP_ODR_DIV_100	0x02	LP_ODR_DIV_20
0x15	HP_ODR_DIV_200	0x03	LP_ODR_DIV_45
0x16	HP_ODR_DIV_400	0x04	LP_ODR_DIV_100
0x17	HP_ODR_DIV_800	0x05	LP_ODR_DIV_200
0x31	HP_REF_MD_ODR_DIV_10	0x06	LP_ODR_DIV_400
0x32	HP_REF_MD_ODR_DIV_20	0x07	LP_ODR_DIV_800
0x33	HP_REF_MD_ODR_DIV_45		

20.5.3 Accelerometer Power Mode

Value	Rate
0x00	High performance
0x01	Normal
0x02	Ultra low power

20.5.4 Self-Test Configuration

Self-test disturbance configuration. Used for both accelerometer and gyro.

Value	Rate
0x00	Off
0x01	Positive
0x02	Negative

20.6 Gyroscope Settings 0x0202

Size		Name	Description
6 bytes	1 byte	rate	Data_rate
	1 byte	full scale	Gyro_Full Scale
	1 byte	lpf1	LPF1_Configuration
	1 bytes	high_perf	High performance mode. 1-enabled, 0-disabled
	1 byte	hpf	High Pass Filter Configuration
	1 byte	self_test	Self-test Disturbance Configuration

20.6.1 Gyroscope Full Scale

Used to configure full scale of gyroscope

Value	Rate
0x00	125 °/s
0x01	250 °/s
0x02	500 °/s
0x03	1000 °/s
0x04	2000 °/s

20.6.2 Gryoscope LPF1 Configuration

Value	Rate	Value	Rate
0x00	Ultra Light	0x05	Very strong
0x01	Very Light	0x06	Aggressive
0x02	Light	0x07	Extreme
0x03	Medium	0x08	Off
0x04	Strong		

20.6.3 Gryoscope HPF Configuration

Value	Rate
0x00	None
0x80	16 mHz
0x81	65 mHz
0x82	260 mHz
0x83	1.04 Hz

20.7 Magnetometer Settings 0x0203

Size		Name	Description
5 bytes	1 byte	rate	Data_rate
	1 byte	temp_comp	Temperature compensation. 1-enabled, 0-disabled
	1 byte	low_power	Low power mode. 1-enabled, 0-disabled
	1 byte	low_pass_filter	Low pass filter. 1-enabled, 0-disabled
	1 byte	self-test	Self-test disturbance enabled. 1-enabled, 0-disabled

20.8 SF Kalman Settings 0x0204

Kalman filter settings. **NOT RECOMMENDED TO CHANGE UNLESS YOU UNDERSTAND THESE VALUES AND THEIR IMPACT. CONSULT SENSOR MAESTROS IF SPECIFIC TUNING IS DESIRED.**

Size		Name	Description
20 bytes	4 bytes	qvy	Gyro sensor noise variance units (deg/s)^2
	4 bytes	qvg	Accelerometer sensor noise variance units g^2 defining minimum deviation from 1g sphere.
	4 bytes	qvb	Magnetometer sensor noise variance units uT^2 defining minimum deviation from geomagnetic sphere.
	4 bytes	qwb	Gyro offset random walk units (deg/s)^2
	4 bytes	max_bpl	Maximum absolute permissible power on gyro offsets(deg/s)

20.9 Calibration Settings 0x0207

Size		Name	Description
20 bytes	1 byte	mode	Magnetometer Calibration Auto Storage Mode
	1 byte	threshold	Magnetometer_Calibration_Type

20.9.1 Magnetometer Calibration Auto Storage Mode

Value	Mode	Comment
0x00	Off	Auto storage disabled
0x01	Once	Magnetometer calibration will be stored once.
0x02	Always	Magnetometer calibration will be stored each time new calibration is calculated.

20.9.2 Magnetometer Calibration Type

Value	Threshold
0x00	None
0x01	4 element
0x02	7 element
0x03	10 element

20.10 Environmental Sensor Settings 0x0310

These settings provide configuration for the Pressure(LPS22H) and Temperature/Humidity(ENS210) sensors on the SFM2.

Size		Name	Description
17 bytes	4 bytes	zero_pressure	Pressure at 0 m. Used to tare altitude, in hPa. Float32.
	4 bytes	alarm_up	High altitude alarm threshold, in m. Set to NAN to disable. Float32.
	4 bytes	alarm_down	Low altitude alarm threshold, in m. Set to NAN to disable. Float32.
	1 byte	Pressure_Data_Rate	**Pressure Data Rate**
	1 byte	lps22hh_low_noise	LPS22HH low noise mode. 1-enabled, 0-disabled.
	1 byte	T/H_sampling_mode	**T/H Sampling Mode**
	2 bytes	ens210_interval	ENS210 sampling interval. Used only when sensor is in *Interval* sampling mode. In ms. Uint16

20.10.1 Pressure Data Rate

LPS22HH Pressure Sensor Data Rate

Value	Rate
0x00	0 Hz
0x01	1 Hz
0x02	10 Hz
0x03	25 Hz
0x04	50 Hz
0x05	75 Hz
0x06	100 Hz

20.10.2 Temp/Humidity Sample Mode

ENS210 Combo Temperature and Humidity Sensor Sampling Mode

Value	Mode	Comment
0x00	Off	Sensor disabled
0x01	Interval	Sampling at configured interval
0x02	Continuous	Sampling as fast as possible

20.10.3 Altitude Tare 0x0311

Writing any value to the *Altitude tare characteristic* tares the altitude (sets it to 0). The altitude/pressure sensor must be enabled for the tare to work.

Size	Name	Description
1 byte	tare	Altitude tare command. Write a 1 to Tare the Altitude data output

20.11 Storage Characteristics

Storage characteristic control storing data in non-volatile memory.

20.11.1 Settings Storage 0x0134

Settings storage is a write-only characteristic. It can be written with single byte commands. Each command triggers an action, depending on the value written.

Value	Action
0x01	Store settings in non-volatile memory
0x02	Restore settings from non-volatile memory
0x03	Restore settings to factory defaults. This does not overwrite non-volatile memory.

20.11.2 Calibration Storage 0x0135

Calibration storage is a write-only characteristic. It can be written with single byte commands. Each command triggers an action, depending on the value written.

Value	Action
0x01	Store magnetometer calibration in non-volatile memory
0x02	Clear magnetometer calibration storage.

20.12 Time Characteristics

Time characteristics are used to provide synchronization between multiple SFM2's. The SENSR-LOGR app utilizes these characteristics to effectively synchronize multiple SFM2's that are being used in Multi Mode to keep the SFM2's data timestamps in synch so that analysis can be performed by aligning the data sets of each SFM2 in time. The Multi Mode Syncrhonization is rather complex. For users that are desiring to create their own mobile apps and want to utilize Multi Mode Synchronization please contact Sensor Maestros directly at sales@sensormaestros.com and we can provide further details on this algorithm. These characteristics are described in a general sense below.

20.12.1 Time 0x0131

The time characteristic allows the user to read or set the RTC Clock running on the SFM2. The time is given in RTC ticks. The RTC tick frequency is 32,768Hz.

Size	Name	Description
4 bytes	time	RTC time in ticks. UInt32

20.12.2 Time Offset 0x0132

Time offset characteristic can be used to apply an offset to the RTC time. Each write to the offset characteristic shift the time by the same amount.

Size	Name	Description
4 bytes	offset	Time offset in ticks. Int32

20.12.3 Time Trim 0x0142

Time trim characteristic can be used to (effectively) trim the frequency of the RTC clock. The actual tick frequency can't be changed. Instead, the trim works by either adding or subtracting one additional tick at a given interval. If the trim value is positive then a tick is added at the interval, if it's negative then a tick is subtracted. When the value is 0 there are no additional ticks.

Size	Name	Description
4 bytes	trim	RTC time trim in ticks. Int32

20.12.4 Time Notify Interval 0x0143

Time notify interval characteristic configures the interval at which the *Time characteristic* sends notifications. The notifications are disabled if the interval is set to 0.

Size	Name	Description
4 bytes	interval	Notify interval. In 1/1024 second steps. Int32

20.13 Other Characteristics

20.13.1 Tare 0x0129

Tare characteristic can be used to tare the sensor fusion in current orientation. It is write-only.

Size	Name	Description
1 bytes	tare	Tare command. 1 - full tare, 2 - heading tare

20.13.2 Name 0x0133

Name characteristic can be used to read or configure the device's name. The name gets updated immediately after write. The name will be what the unit shows in the BLE Advertisement.

The name is ASCII encoded, without the null terminator. The characteristic is of variable length, with a maximum length of 16.

20.13.3 Stats 0x0137

Once every 2 seconds the device's statistics are transmitted through the stats characteristic. It can also be read directly to immediately get the statistics. Writing any value to this characteristics clears the stats.

Size		Name	Description
24 bytes	4 bytes	runtime	Device runtime in ticks (1/1024 s). UInt32
	4 bytes	ble_sent	Count of bytes sent through BLE. UInt32.
	4 bytes	ble_dropped	Count of bytes dropped when sending via BLE. UInt32.
	4 bytes	serial_sent	Count of bytes written to serial. UInt32
	4 bytes	serial_dropped	Count of bytes dropped when writing to serial. UInt32
	2 bytes	max_queue	Maximum count of element in IMU queue.
	2 bytes	calibration_stores	The number of times the magnetometer calibration was stored.

20.13.4 Connection Parameters 0x0138

The connection parameters characteristic holds the current BLE connection parameters. It is updated each time any of the parameters changes.

Size		Name	Description
24 bytes	2 bytes	mtu	MTU, in bytes. UInt16.
	2 bytes	connection_interval	Connection Interval in 1.25 ms units. 0 if not set. UInt16.
	2 bytes	slave_latency	Slave Latency in number of connection events. -1 if not set. Int16.
	2 bytes	conn_sup_timeout	Connection Supervision Timeout in 10 ms units. 0 if not set. UInt16.
	1 byte	tx_phy	TX **PHY**
	1 byte	rx_phy	RX **PHY**

20.13.4.1 PHY

Indicates which BLE PHY is being used for the BLE connection.

Value	Phase
0x00	Not set (unknown).
0x01	1M
0x02	2M
0x04	Coded

20.13.5 Calibration 0x0206

Calibration characteristic provides the magnetometer current calibration status.

Size		Name	Description
20 bytes	4 bytes	Error	Calibration error in %. Float32
	1 byte	Type	**Calibration Type**

20.13.6 Self-Test 0x0208

Self-test characteristic can be used to trigger and monitor self-test. The characteristic can be read and emits notifications. Writing any value to it starts self-test.

Size		Name	Description
4 bytes	1 byte	phase	**Self-test Phase**
	1 byte	accel	Accelerometer result. (1 - ok, 0 - failure)
	1 byte	gyro	Gyroscope result. (1 - ok, 0 - failure)
	1 byte	mag	Magnetometer result. (1 - ok, 0 - failure)

20.13.6.1 Self-Test Phase

Provides Self-Test status whether it has been run prior and any disturbances.

Value	Phase
0x00	Never run
0x01	Base measurement
0x02	Positive disturbance measurement
0x03	Negative disturbance measurement
0x04	Completed

20.13.7 SF Kalman Settings 0x0204

Size		Name	Description
20 bytes	4 bytes	qvy	Gyro sensor noise variance units (deg/s)^2
	4 bytes	qvg	Accelerometer sensor noise variance units g^2 defining minimum deviation from 1g sphere.
	4 bytes	qvb	Magnetometer sensor noise variance units uT^2 defining minimum deviation from geomagnetic sphere.
	4 bytes	qwb	Gyro offset random walk units (deg/s)^2
	4 bytes	max_bpl	Maximum absolute permissible power on gyro offsets(deg/s)

20.13.8 Calibration Settings 0x0207

Size		Name	Description
20 bytes	1 byte	mode	Magnetometer_calibration_auto_storage_mode
	1 byte	threshold	Magnetometer_calibration_auto_storage_threshold

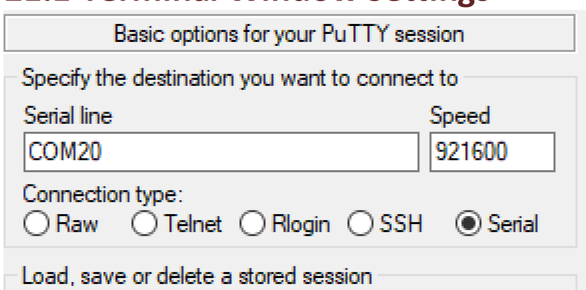
21.0 USB Commands

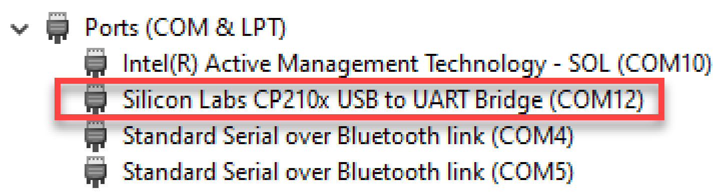
NOTE: This section describes ASCII commands that can be used with the SFM2 over a USB/COM Port interface. If it is desired to use the USB interface it is HIGHLY RECOMMENDED to utilize the SFM2 Python Library. The SFM2 Python Library can be downloaded from the SFM2 website at the link below and is further described in Section 21.1.1 Python API Library. The Python Library provides a high level Python API library that utilizes the lower level ASCII type commands.

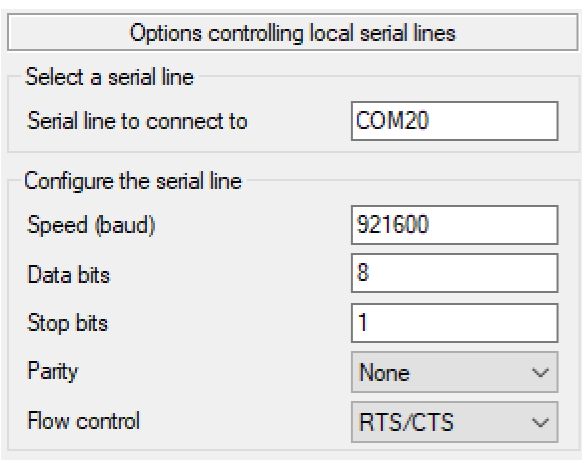
SFM2 Python Library Download: https://sensormaestros.com/wp-content/uploads/sfm2_python_lib_0_7.zip

This protocol is used by the SFM2 to connect to an external computer via a standard COM port. COM port Baud Rate is 921600 baud. Using a terminal window such as PuTTY the COM Port settings would look like this. COM Port would be the COM port that is shown for the SFM2 in the 'Device Manager'.

21.1 Terminal Window Settings







21.2 Command Format:

The top-level protocol elements are commands, queries, responses, and data. Commands and responses are of the general form:

FOO=37

where "FOO" is the designator for a specific command or response, "=" is a separator between the designator and parameter(s), and "37" is a parameter.

Designators such as "FOO" in the above example are considered case insensitive. The receiving side MUST accept "FOO", "foo", and even "FOO" interchangeably, e.g., by using the C standard library strcasecmp() or strncasecmp() functions for case insensitive string comparison, or by converting the received characters to a specific case before comparison. Where the designators are programmatically generated, they SHOULD be generated in upper case.

A command such as "FOO=37" will result in a similar response, typically also "FOO=37". However, in a case where the parameter value is not acceptable, the response may have a different parameter value, which is the actual parameter value that will be used henceforth. This may be the previous value of the parameter, or a new value that is an approximation of the requested parameter value. For instance, if the previous value of the parameter for FOO was 5, and 37 is not an acceptable value, but 35 is acceptable, then a "FOO=37" command might yield a response of "FOO=5" or a response of "FOO=35".

Under some circumstances a command may yield multiple responses. For instance, if the parameters to FOO and BAR are related, such that only certain combinations are permitted, then a command "FOO=37" might yield two consecutive responses "FOO=35" and "BAR=42".

Under some circumstances responses may be generated spontaneously rather than in response to a command. This might happen if a parameter change occurs based on an alternate communication channel. For instance, if the USB CDC channel sends a "FOO=37" command, a "FOO=35" response may be sent both over the USB CDC channel and to the nRF52. This is necessary so that the nRF52 firmware will update the appropriate characteristic values, and send corresponding notifications if enabled.
Queries are similar to commands, but rather than having a "=" followed by parameter(s), they have a "?" and no parameters. The response to a query is formatted in the same way as a command response.
Data is formatted similarly to reponses, but with a ":" separating the designator from the data.

21.3 Special characters:

\r carriage return, 0x13
\n newline, 0x10
All commands and queries MUST be sent with a terminating carriage return. For a command, the carriage return MAY be followed by a line feed (which is highly recommended), but the receiving side **MUST NOT** require the line feed.
Responses and data **MUST** be sent terminated with a carriage return followed by a line feed. The receiving side **MUST NOT** require the line feed. In general, the receiving side of any protocol element should ignore the presence (or absence) of line feed characters.

21.4 Metavariables:

In the descriptions of commands, repsonses, and data, items enclosed in angle brackets, e.g., "<sample_rate-int>" are metavariables. The angle brackets are only for syntactic specification, and are **NOT** actually sent or received on the serial link. Within the metavariable specification, the portion before the hyphen identifies the nature of the data item, and the portion after the hypen indicates the data type.

21.5 Data types:

boolean:
"0" (false) or "1" (true)
int16:
16-bit integer represented as up to five decimal digits, without leading zeros, and with an optional leading "-" for negative values
int32:
32-bit signed integer
uint32:
32-bit unsigned integer can be preceded by a "-" to indicate negative values
float:
Floating-point value represented as a mantissa portion with optional leading "-" sign and optional "." radix mark, and optional exponent part

21.6 System Commands:

NAME=<=Alpha-Numeric String>
CONFIG?
Shows current configuration of the SFM2 and provides a list of available USB commands.
SFRESET!
Reset sensors to 0Hz ODR including Sensor Fusion output
SELFTEST!
Performs automated Self-Test on Accel, Gyro and Mag sensors.
CALIBSTORE!
Performs SFM2 calibration and stores sensor offsets into Non-Volatile memory. To run the calibration enable each sensor with a >0Hz ODR rate and enable SFOR > 0Hz.
Success => CALIBSTORE=VALID
CALIBCLEAR!
Clears previously stored calibration data.
Success => CALIBSTORE=EMPTY
GLOBREF=<enable-boolean>

Enable/Disable Global Reference Frame

BINMODE=<enable-boolean>

1 = COM Sensor output in Binary

0 = COM Sensor in ASCII format(Default)

21.7 Motion Sensor Configuration Commands:

21.7.1 Accelerometer Commands

ASR=<sample_rate-int16>

Set accelerometer sample rate in Hz (0 for sampling disabled)

AFR=<full_scale_range-int16>

Set accelerometer full-scale range in g

AFASTSET=<enable-boolean>

Set accelerometer Fast Settling

ALPF2=<enable-boolean>

Enable Accelerometer 2nd stage Low Pass Filter

ADE=<enable-boolean>

Set accelerometer data output enable

21.7.2 Gyroscope Commands

GSR=<sample_rate-int16>

Set gyro sample rate in Hz (0 for sampling disabled)

GFR=<full_scale_range-int16>

Set gyro full-scale range in dps

GDE=<enable-boolean>

Set gyro data output enable

GHP=<enable-boolean>

Enable/Disable Gyroscope High Performance Mode

21.7.3 Magnetometer Commands

MSR=<sample_rate-int16>

Set magnetometer sample rate in Hz (0 for sampling disabled)

MFR=<full_scale_range-int16>

Set magnetometer full-scale range in uT

MDE=<enable-boolean>

Enable/Disable magnetometer data output enable

MTEMP=<enable-boolean>

Enable/Disable Mag Temperature Compensation. Recommended to Leave Enabled.

MLP=<enable-boolean>

Enable/Disable Magnetometer Low Power Mode

MLPF=<enable-boolean>

Enable/Disable Magnetometer Low Pass Filter

21.8 Environmental Sensor Configuration Commands:

PSR=<sample_rate-int16>

Set Pressure Sensor Sample Rate in Hz(0 for sampling disabled, 1, 10, 25, 50, 75, 100)

If values outside of these are entered the value will be rounded to the nearest value

PLN=<enable-boolean>

Set Pressure Low Noise operation ON/OFF

ALTDE=<enable-boolean>

Set Altitude data output enable

ALTZERO=<tare_altitude-int16>

Enter value to 'Tare' (zero) altitude. Value entered is to zero Pressure as altitude is calculated from Pressure data.

PDE=<enable-boolean>

Set Pressure data output enable. Pressure output in hPa

TMODE=<OFF, INTERVAL, CONTINUOUS>

Set Temperature/Humidity Sensor sampling: OFF, INTERVAL, CONTINUOUS

If set to INTERVAL need to enter value for Interval using TINT

TINT=<sampling_interval-int16>

Set Temp/Humidity sampling Interval in ms (1000 = 1sec)

HDE=<enable-boolean>

Set Humidity data output enable. Humidity output in % humidity

TDE=<enable-boolean>

Set Temperature data output enable. Temperature output in Celsius.

21.9 Sensor Fusion Commands:

SFOR=<fusion_rate-int16>

Set sensor fusion output rate in Hz

SFOP=<fusion_rate_output_prescaler (integer 1-255)>

Divides Sensor Fusion Output rate by this integer. I.E. – If SFOR = 417 and SFOP = 2 ; actual SFOR ODR = 208Hz

SFQDE=<enable-boolean>

Set sensor fusion untared quaternion data output enable

SFQTDE=<enable-boolean>

Set sensor fusion tared quaternion data output enable

SFCHTDE=<enable-boolean>

Set sensor fusion compass heading and tilt data output enable

SFLADE=<enable-boolean>

Set sensor fusion linear acceleration data output enable

SFEADE=<enable-boolean>

Set sensor fusion euler angle data output enable

SFTARE!

Tare sensor fusion

SFTARE=<w>,<x>,<y>,<z>   (coming soon!!)

Set sensor fusion tare quaternion

21.10 Timestamp Commands:

TIME!

Resets the Timestamp value to 0

TIME=<uint32>

Set the Timestamp value to an integer. Time is given in RTC Ticks from the RTC on the SFM2. The RTC tick frequency is 32,768Hz. TIME can be proceeded value with "-" sign to indicate negative value.

TOFFSET!

Resets the Timestamp value to 0

TOFFSET=<int32>

The Time Offset can be used to apply an offset to the RTC time. Each write to Offset shifts the time by the same amount. Can proceed value with "-" sign to indicate negative value.

TTRIM!

Resets the Time Trim to 0

TRIM=<int32>

The Time Trim can be used to fine tune the SFM2 RTC. When TTRIM is set to a positive value X then one tick is added to the RTC every X ticks. If TTRIM is set to a negative value Y then one tick is subtracted from the RTC every -Y ticks. E.g. Suppose that the SFM2 RTC ticks slightly too fast and every second it gets ahead of the reference clock by 1ms. In that case TTRIM can be set to -1000 to compensate.

TINTERVAL=<uint32>

Sets the Time Notify Interval. When set to non-zero value SFM2 periodically sends time status, as if queried with "TIME?". The interval is given in 1/1024 s.

TSDE=<enable-boolean>

Time Sync data enable. When enabled Time Sync samples are included in the Binary data frame (see SFM2 - Binary frame document). The Time Sync samples are always output at 52 Hz rate. Stamp Data Enabled.

21.11 System Queries:

Config?

Shows current configuration of the SFM2 and provides a list of available USB commands.

NAME?
Provides name of SFM2
TIME?
Provides current 'time' of TIMESFM2
CALIBSTORE?
Returns
=VALID (Calibration has been performed and stored into NV Memory)
OR
=EMPTY (Calibration has not been performed)

21.12 Sensor Queries:

ASR?
Get accelerometer sample rate
AFR?
Get accelerometer full-scale range
ADE?
Get accelerometer data output enable
GSR?
Get gyro sample rate
GFR?
Get gyro full-scale range
GDE?
Get gyro data output enable
MSR?
Get magnetometer sample rate
MFR?
Get magnetometer full-scale range
MDE?
Get magnetometer data output enable
SSAT?
Get sensor saturation

21.13 Sensor Fusion Queries

SFOR?
Get sensor fusion output rate in Hz
SFQDE?
Get sensor fusion untared quaternion data output enable
SQTDE?
Get sensor fusion tared quaternion data output enable
SFCHTDE?
Get sensor fusion compass heading and tilt data output enable
SFEADE?
Get sensor fusion euler angle data output enable
SFLADE?
Get sensor fusion linear acceleration data output enable
SFTARE?
Get sensor fusion tare quaternion

21.14 Responses:

SRESET=<self_test-bool>
Sensor reset response
ASR=<sample_rate-int>
Accelerometer sample rate in Hz (0 for sampling disabled)
AFR=<full_scale_range-int>
Accelerometer full-scale range in g
ADE=<enable-boolean>

Accelerometer data output enable

GSR=<sample_rate-int>
Gyro sample rate in Hz (0 for sampling disabled)

GFR=<full_scale_range-int>
Gyro full-scale range in dps

GDE=<enable-boolean>
Gyro data output enable

MSR=<sample_rate-int>
Magnetometer sample rate in Hz (0 for sampling disabled)

MFR=<full_scale_range-int>
Magnetometer full-scale range in uT

MDE=<enable-boolean>
Magnetometer data output enable

SSAT=<accel>,<gyro>,<mag>
Sensor saturation

SFOR=<fusion_rate-int>
Sensor fusion output rate in Hz

SFQDE=<enable-boolean>
Sensor fusion untared quaternion data output enable

SFQTDE=<enable-boolean>
Sensor fusion tared quaternion data output enable

SFCHTDE=<enable-boolean>
Sensor fusion compass heading and tilt data output enable

SFEADE=<enable-boolean>
Sensor fusion euler angle data output enable

SFLADE=<enable-boolean>
Sensor fusion linear acceleration data output enable

SFTARE=<w>,<x>,<y>,<z>
Sensor fusion tare quaternion

21.15 Data Stream Format:

AD:<x>,<y>,<z>
Accelerometer data 32bit Float

GD:<x>,<y>,<z>
Gyro data 32bit Float

MD:<x>,<y>,<z>
Magnetometer data 32bit Float

SFQ:<w>,<x>,<y>,<z>
Sensor fusion untared quaternion

SFQT:<w>,<x>,<y>,<z>
Sensor fusion tared quaternion

SFCHT:<hdg>,<tilt>
Sensor fusion compass heading and tilt

SFEA:<roll>,<pitch>,<yaw>
Sensor fusion euler angle data

SFLA:<x>,<y>,<z>
Sensor fusion linear acceleration

22.0 Python API Library

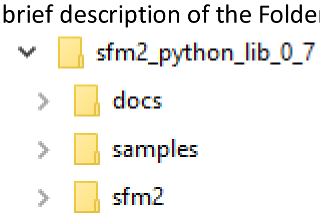
Sensor Maestros will provide example software to assist customers getting their applications developed quickly to interface to the SFM2. Example projects are provided for Python as indicated below. The SFM2 Python Library can be downloaded from the SFM2 webpage at the following link.

SFM2 Python Library and Documentation: https://sensormaestros.com/wp-content/uploads/sfm2_python_lib_0_7.zip

The Python API Library can be used with Python 3.7 or higher. It provides an easy means for users to connect to the SFM2 via a COM/USB port and quickly get an application up and running. The Python Library provides high level API functions that utilize the AT Commands. The Python API Library provides numerous examples and maximizes throughput by enabling Binary Data Streams which will allow for multiple SFM2 to be connected simultaneously. It is certainly feasible to connect 5 SFM2's depending on the exact data streams that have been enabled and Fusion ODR rate. This has also been tested on a Raspberry Pi 4.

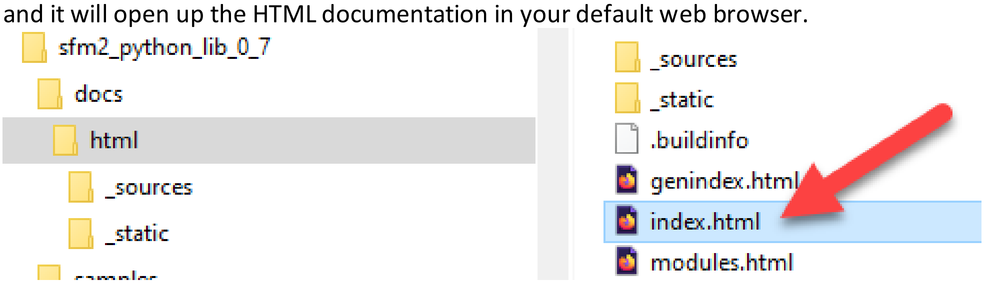
- Numerous examples to quickly and easily get started in the /samples folder
- Data-log timestamped sensor streams to CSV file
- COM Port Throughput Test
- SFM2 Self-Test
- Binary Mode to optimize throughput
- Show 3D Cube representation of SFM2
- Provides 4 Standard SFM2 configurations:
  - OFF
  - Low Power (ACC=26Hz, GYRO=26Hz, MAG=26Hz, SFOR=26Hz)
  - Balanced (ACC=104Hz, GYRO=104Hz, MAG=104Hz, SFOR=104Hz)
  - Performance (ACC=833Hz, GYRO=833Hz, MAG=104Hz, SFOR=417Hz)
- Provides means to measure COM port throughput

A brief description of the Folder Structure Python API's is provided below…

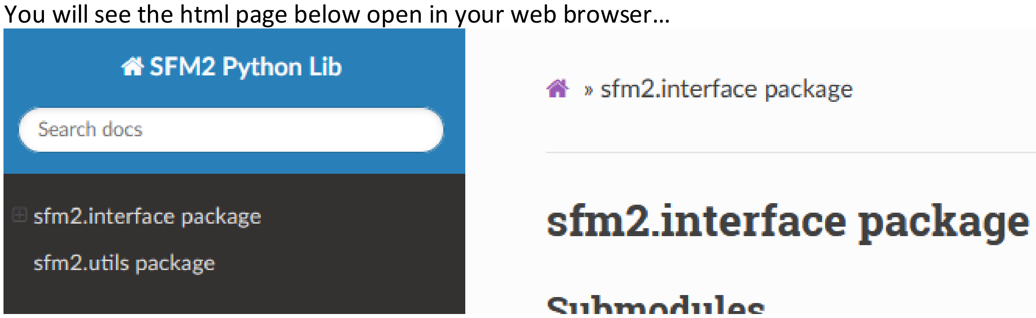


22.1 Python Docs (html documentation)

To read the HTML documentation for the SFM2 Python Library navigate to the docs/html folder and double-click on the 'index.html' file and it will open up the HTML documentation in your default web browser.



You will see the html page below open in your web browser…



The 2 main categories can be expanded by clicking on them…

⊟ sfm2.interface package

- Submodules
- sfm2.interface.config_types module
- sfm2.interface.env_sensor_config_types module
- sfm2.interface.exceptions module
- sfm2.interface.sample_types module
- sfm2.interface.sfm2 module
- sfm2.interface.sfm2 module
- sfm2.interface.sfm2_accelerometer module
- sfm2.interface.sfm2_config module
- sfm2.interface.sfm2_data module
- sfm2.interface.sfm2_env_sensors module
- sfm2.interface.sfm2_gyroscope module
- sfm2.interface.sfm2_init_options module
- sfm2.interface.sfm2_magnetometer module
- sfm2.interface.sfm2_sensor_fusion module
- sfm2.interface.sfm2_time module
- Module contents

⊟ sfm2.utils package

- Submodules
- sfm2.utils.csv_writer module
- sfm2.utils.throughput module
- sfm2.utils.time_sync module
- Module contents

22.2 Python Samples

The 'samples' folder contains numerous high level examples to help show a user how to utilize the Python API library. The example list is shown below…

- _add_lib_to_path.py
- configuring_logs.py
- reading_env_sensors.py
- reading_gyro_data.py
- reading_gyro_data_COMxx.py
- reading_sensor_fusion_data.py
- running_self_test.py
- showing_3d_cube.py
- synchronizing_time.py
- testing_throughput.py
- testing_throughput_multiple.py
- testing_time_synchronization.py
- using_multiple_sfm2s.py
- using_tare_and_globref.py
- using_time.py
- writing_multiple_csvs.py
- writing_to_csv.py
- writing_to_csv_COMxx.py

22.3 Python sfm2 API's

The /sfm2 folder contains 3 folders each containing various API's for either interfacing to or configuring the SFM2 in some form or fashion. Each folder is briefly described below…

- sfm2
  - __pycache__
  - _implementation
  - interface
  - utils

22.3.1 sfm2: implementation

Implementation folder contains API's for interfacing to the SFM2 and configuring SFM2 as described further below.

- _implementation
  - __pycache__
  - commands
  - components
  - internal_utils
  - sample_parsing
  - serial_port

- commands
- components
- internal_utils
- sample_parsing
- serial_port
- __init__.py
- sfm2_discovery.py
- sfm2_on_serial_port.py

Top level includes Serial port functions.

22.3.1.1 Commands

/commands folder contains functions for handling/processing commands…

- command.py
- command_interface.py
- command_list.py
- wait_for_response.py

22.3.1.2 Components

/components folder contains intermediate functions for configuring sensors, enabling various outputs, etc.

- data_enabler.py
- self_test.py
- sfm2_accelerometer_impl.py
- sfm2_config_impl.py
- sfm2_data_impl.py
- sfm2_env_sensors_impl.py
- sfm2_gyroscope_impl.py
- sfm2_magnetometer_impl.py
- sfm2_name.py
- sfm2_reference_impl.py
- sfm2_sensor_fusion_impl.py
- sfm2_time_impl.py
- version_check.py

22.3.1.3 Internal Utils

/internal_utils contains various system/housekeeping functions.

- background_thread_executor.py
- cancellation_token.py
- dispose_on_exception.py
- disposed_token.py
- find_sample_rate.py
- observable_on_new_thread.py
- sample_timestamper.py
- sfm2_rtc_time_conversion.py
- thread_raw_subject.py
- thread_shared_subject.py
- wrap_exception.py

22.3.1.4 Sample Parsing

/sample_parsing folder contains functions for parsing data and handling binary frame mode.

- binary_mode_frame.py
- sample_factories.py
- sample_parser.py

22.3.1.5 Serial Port

/serial_port contains serial port handling functions

- bytes_stream.py
- packetizer.py
- serial_port.py

22.3.2 sfm2: interface

/interface folder contains high level functions for the SFM2. sfm2.py is the main interface function for a connected SFM2.

- config_types.py
- env_sensor_config_types.py
- exceptions.py
- found_sfm2.py
- sample_types.py
- sfm2.py
- sfm2_accelerometer.py
- sfm2_config.py
- sfm2_data.py
- sfm2_env_sensors.py
- sfm2_gyroscope.py
- sfm2_init_options.py
- sfm2_magnetometer.py
- sfm2_reference.py
- sfm2_sensor_fusion.py
- sfm2_time.py

22.3.3 sfm2: Utils

/utils folder contains various SFM2 utility functions such as writing sensor data to a *.csv file, analyzing USB throughput, synchronizing multiple SFM2's, etc.

- csv_writer.py
- sfm2_collection.py
- throughput.py
- time_sync.py

23.0 Revisions

1.0.0: Initial Relase
1.0.1: Updated AD,GD,MD Data values from INT's to Floats
1.0.2: 2022-02-18 Updated SFM2 Pictures
1.0.3: 20230622

- Include new TimeStamp variables and configurations
- Include Binary Frame Mode description
- Include new Multi Mode: Time Synchronization variables and configuration for synching multiple SFM2's
- Update Battery Life estimates for 130mAh battery vs. original estimations for 250mAh battery capacity
- Update mobile App pictures
- Update SFM2 Python API section