shimmer
DISCOVERY IN MOTION

LogAndStream for Shimmer3
Firmware User Manual
Rev 0.1a

Legal Notices and Disclaimer

Table of Contents

1. Introduction

This document is an accompaniment to the *LogAndStream Firmware v0.1.0* image and later for *Shimmer3*. No previous development experience is required. **Note:** *LogAndStream Firmware v0.1.0* is a Beta release.

LogAndStream firmware is a complete data recording solution which merges features from our previous *Shimmer3* firmware releases. The *LogAndStream* firmware facilitates logging of data from a *Shimmer3* to the on-board SD card while also providing the ability to simultaneously stream data wireless connection to a Bluetooth-enabled PC. The firmware allows for full user configuration of the *Shimmer3* using a configuration file stored on the SD card, or over Bluetooth using the provided *ShimmerCapture* PC application.

It is recommended that the *LogAndStream* firmware be used in conjunction with the *ShimmerCapture* and/or *ShimmerLog* PC applications, which are available for download from the Shimmer website (www.shimmersensing.com).

1.1. Scope of this User Manual

The purpose of this *User Manual* is to guide the user through the features of the *LogAndStream* firmware and to provide the required instructions to configure the data logging options, configure the data streaming options and to parse the recorded and received data. The *User Manual* does not provide an extensive explanation of the source code for the firmware.

1.2. Pre-Requisites

The *LogAndStream* firmware can be used with a *Shimmer3* device that has a connected microSD card. For *Shimmer3* compatibility, the chosen microSD card must be less than 32GB in capacity, not be an XC (eXtended Capacity) card and it must support 1-bit SPI mode.

For a wireless connection to the *Shimmer3*, a Bluetooth enabled PC and the *ShimmerCapture* is required to interface with *Shimmer3* and receive the streamed data. The *ShimmerCapture* application allows a user to fully configure the *Shimmer3* whilst providing the ability to save streaming data locally to the PC in real-time. Concurrently, a *Shimmer Dock* is required to allow access the microSD card on the Shimmer from a PC. The *Shimmer Dock* provides an alternative method for configuring the*Shimmer3* logging preferences (using the *ShimmerLog* application) whilst also facilitating data transfer (either RAW or calibrated) from the microSD card. Please note that legacy (black) Shimmer docks are not suitable for this purpose and a newer (white) dock is required.

For more information on the *Shimmer3* platform, please refer to the *Shimmer3 User Manual* which is available for download at www.shimmersensing.com.

1.3. Installation

Install the *LogAndStream Firmware* v0.1.0 image (LogAndStream_v0.1.0_shimmer3.txt) onto a *Shimmer3* device, using the *Shimmer3 Bootstrap Loader* (*Shimmer3 BSL*) application. The Shimmer3BSL.exe application is available via download from the Members area at www.shimmersensing.com.

2. Firmware Features

As stated previously, the *LogAndStream* firmware provides a seamless integration of the existing *Shimmer3* firmware releases of *SDLog* (*i.e.*, logging *Shimmer3* data to an on-board microSD card) and *BtStream* (*i.e.*, streaming *Shimmer3* data back to "host" device). As such, many users will already be familiar with the operation of the firmware and many its features. The *LogAndStream* firmware has been designed such that a user can still utilise the *Shimmer3* in a purely *BtStream* or *SDLog* mode if they desire. What sets the *LogAndStream* firmware apart from previous firmware releases is its ability to perform both *SDLog* and *BtStream* operations simultaneously.

A *Shimmer3* programmed with *LogAndStream* firmware can be in one of five states: *Idle*, *BT Connected*, *BT Streaming*, *BT Streaming + SD Logging* or *SD Logging* - as shown in Figure 4-1. The active states form two operational branches from which the user can choose to operate the *Shimmer3* device - one initiated by a Bluetooth connection (blue shaded area in Figure 4-1) and the other based on SD Logging operation (orange shaded area in Figure 4-1).

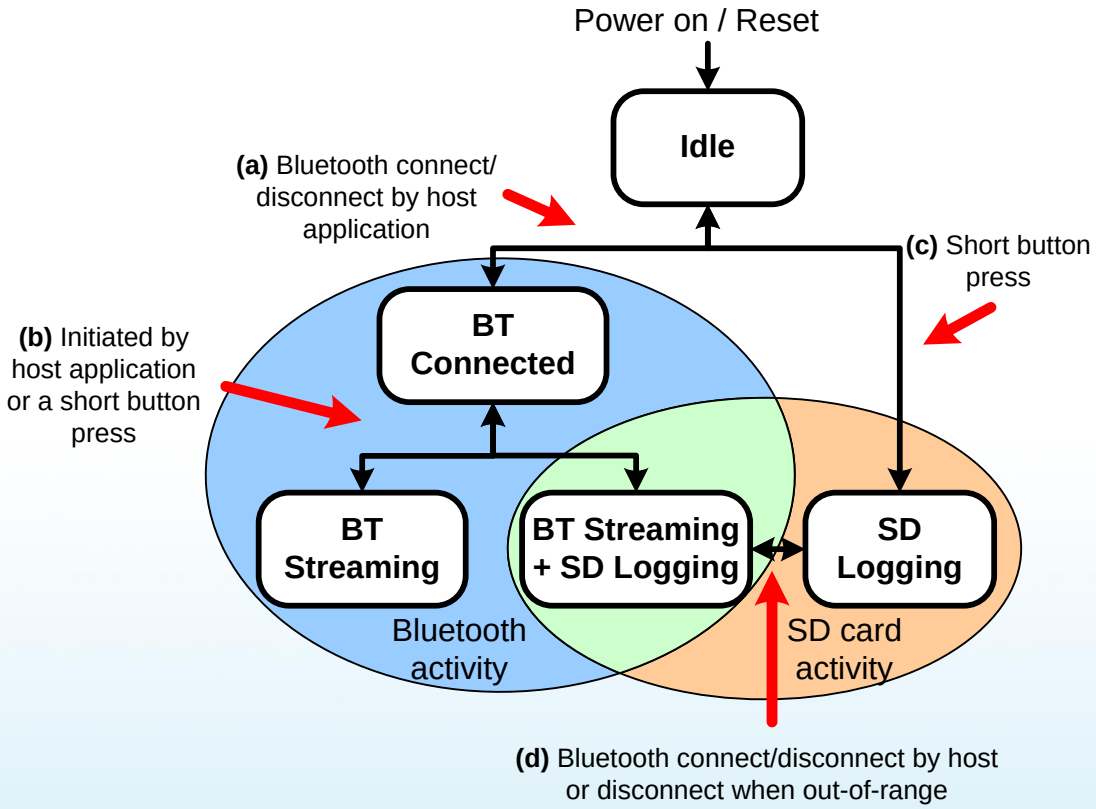

Figure 2-1 LogAndStream firmware operational hierarchy.

The key feature of the *LogAndStream* firmware can be viewed at the intersection of the two active branches (green shaded area in Figure 4-1). From a "host" application, the user can select to stream *Shimmer3* sensor data over a Bluetooth connection at the same time as saving it locally to the on-board microSD card. An instance where this is of key benefit is the case where, in the process of data collection, a *Shimmer3* might travel out of range from the "host" device. In this circumstance, the *Shimmer3* will continue to save data to the microSD card such that there will be no break in the

data recording. Similarly, during normal SD logging operation, continuous Bluetooth streaming is not always desirable or required (*e.g.*, during long data recordings where a lower battery consumption is key). By combining Bluetooth support with SD logging capability, the new *LogAndStream* firmware now facilitates the ability for the user to check the recording activity of a *Shimmer3* at any stage during SD logging operation - without interrupting the data recording.

2.1. General operation

State 1 - Idle

When a *Shimmer3*, programmed with the *LogAndStream* firmware, is first powered on or reset, it enters the *Idle* state. It will remain in the *Idle* state until either a connection is made over the Bluetooth link (*i.e.*, by opening a serial connection), or the user instigates a single button press which will initiate *SD logging* mode. It should be noted that this operational case is valid when "Undock Start" is disabled in the *Shimmer3* configuration - see Section 3.5. When "Undock Start" is enabled, the *Shimmer3* will progress directly from *Idle* mode into *SD Logging* mode after a power-on or reset condition.

> **Note:** To use the Bluetooth features of the *LogAndStream* firmware, the *Shimmer3* device must first be paired with a "host" device (*e.g.*, a PC), as outlined in the *Shimmer3 User Manual*.

Path (a): To enter *BT Connected* state, a Bluetooth connection must be initiated by a "host" side application (*e.g.*, *ShimmerCapture*). A subsequent disconnection of the Bluetooth connection will send the *Shimmer3* back to the *Idle* state.

Path (c): If "*Undock Start*" is disabled, a short button press on the *Shimmer3* will initiate SD *Logging* state. Similarly, another short button press will terminate *SD Logging* mode and return the *Shimmer3* to *Idle* state.

State 2 - BT Connected

In the *BT Connected* state, the *Shimmer3* can process various commands to configure its sensors and sampling parameters, set calibration parameters, send configuration settings back to the "host" (PC, mobile or other) and start sampling. When the *Shimmer3* is in the *BT Connected*, *BT Streaming* or *BT Streaming* + SD Logging states, there can be active communication between the *Shimmer3* and the host over the Bluetooth serial connection. Packets of bytes are sent in both directions and these can consist of commands, responses or data.

Path (b): The host application (*e.g.*, *ShimmerCapture*) allows the user to fully configure the *Shimmer3* and choose between either *BT Streaming* or *BT Streaming + SD Logging* recording modes. Alternatively, the short button press action can be used to start or stop *BT Streaming + SD Logging* mode from the *initial BT Connected* branch shown in Figure 2-1.

State 3 - BT Streaming

When a command to start *BT Streaming* is received, the *Shimmer3* enters the *BT Streaming* state and starts sampling data from its sensors and sending that data over the Bluetooth link. This state will continue until a command to stop logging is received, whereupon the *Shimmer3* returns to the *BT Connected* state. When in the *BT Streaming* state, a short button press can also be used to halt streaming and return the *Shimmer3* to *BT Connected* state. Closing the serial connection will put the *Shimmer3* back into the *Idle* state. Similarly, if the *Shimmer3* goes out-of-range from the "host" device for a certain period (≈ 150 s), the *Shimmer3* will also return to *Idle* state.

State 4 - BT Streaming + SD Logging

As described previously, the *BT Streaming + SD Logging* state can be initiated either by using the "host" application (*e.g.*, *ShimmerCapture*) or using a short button press from the initial *BT Connected* state. When the *Shimmer3* enters the *BT Streaming+ SD Logging* state, the *Shimmer3* will start sampling data from its sensors and send this data over the Bluetooth link as well as store it locally in raw file format to the microSD card (see Section 5).

The *Shimmer3* will continue in this mode until either one of two conditions. Firstly, if a stop logging command is received from the "host" or, as mentioned previously, the user instigates a short button press, the *Shimmer3* will return to the *BT Connected* state. Secondly, in the event of either a Bluetooth disconnection, the serial port being closed or the *Shimmer3* travels out of range for a certain period (≈ 150 s), the *Shimmer3* will automatically enter the *SD Logging* state (as illustrated by "Path (d)" in Figure 4-1).

State 5 - SD Logging

In the *SD Logging* state, the *Shimmer3* will continuously read data from its sensors and store this data locally to a raw data file in its on-board microSD card (for more information see Section 5). The *Shimmer3* will remain open to a Bluetooth connection by a host device whereupon, the Shimmer3 which automatically switch to the *BT Streaming + SD Logging* state without interruption to the SD logging data stream.

2.2. LED indicators

The *Shimmer3* has five LEDs in two locations: the lower LED location (location A) which contains the green (b), yellow[1] and red LEDs; and the upper LED location (location B) which contains the green (a) and blue LEDs.

[1] Note that what is referred to as the yellow LED may appear orange to some users.

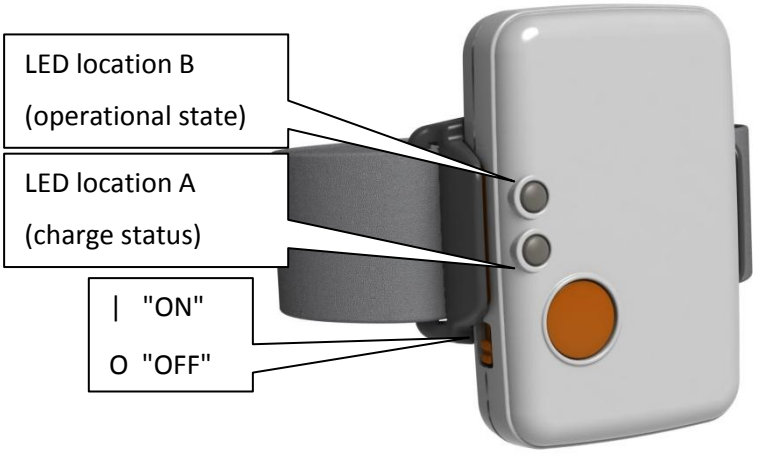

Figure 2-2 Shimmer3 in enclosure.

The two upper LEDs are used to indicate state of operation for the *LogAndStream* firmware. The status of each depends on whether the *Shimmer3* is in a docked or undocked state, as shown in Table 2-1 below.

	UNDOCKED		DOCKED	
LED	**Green (b)**	**Blue**	**Green (b)**	**Blue**
Power OFF	OFF	OFF	OFF	OFF
Idle mode	OFF	0.1 s ON/2 s OFF	N/A	0.1 s ON/2 s OFF
Connected	OFF	ON	N/A	ON
Streaming only	OFF	1 s ON/1 s OFF	N/A	1 s ON/1 s OFF
Logging only	1 s ON/1 s OFF	OFF	N/A	N/A
Streaming+Logging	1 s each 		N/A	N/A
Configuring	OFF	0.1 s ON/0.1 s OFF	N/A	0.1 s ON/0.1 s OFF
Error*	0.1 s ON/0.1 s OFF	0.1 s ON/0.1 s OFF	0.1 s ON/0.1 s OFF	0.1 s ON/0.1 s OFF

Table 2-1 - LED indicators specifc to LogAndStream firmware (undocked and docked). * In Error mode, the Blue and Green (b) LEDs alternate.

Note: The Shimmer unit should never be placed in the dock while the operation LEDs indicate that it is configuring as this may cause a file-system error. Once configuration has begun, you must power off or reset the Shimmer unit before docking.

Note: It is not recommend to place the *Shimmer3* unit in the dock while it is logging data as this can cause SD card access problems. Once logging has begun, you must power off or reset the *Shimmer3* unit before docking.

The lower three LEDs are exclusively used across all *Shimmer3* firmware as a battery charge indicator and will show the colour corresponding to the charge status. As above, the status of each LED depends on whether the *Shimmer3* is in a docked or undocked state. If the *Shimmer3* is on a dock or

multi-charger and is powered on, the battery charge status indicator will show a solid LED. If the *Shimmer3* is undocked, the corresponding LED will flash, as described below in Table 2-2.

	UNDOCKED			DOCKED		
LED	**Green (a)**	**Yellow**	**Red**	**Green (a)**	**Yellow**	**Red**
Power OFF	OFF	OFF	OFF	OFF	OFF	OFF
Low Charge	OFF	OFF	0.1 s ON/5 s OFF	OFF	OFF	ON
Medium Charge	OFF	0.1 s ON/5 s OFF	OFF	OFF	ON	OFF
Full Charge	0.1 s ON/5 s OFF	OFF	OFF	ON	OFF	OFF

Table 2-2 - Power status LED indicators for Shimmer3 firmware (undocked and docked).

2.3. Start/Stop Logging

Dock (automatic start)

This is the default way for the *Shimmer3* to start and stop logging data to the SD card and is enabled if the "Undock Start" option is enabled from ShimmerLog or ShimmerCapture (*i.e.*, configuration file line entry: "userbutton=0"). With this configuration, if the Shimmer is powered on or reset while it is **not** on the dock, the following process will begin immediately:

1. The Shimmer unit will go into standby mode for up to 3 seconds.
2. The Shimmer unit will read the configuration file or infomem and create the required directories on the SD card.
3. The Shimmer unit will start logging.

Alternatively, if the Shimmer is powered on or reset while it **is** on the dock, the three steps above will begin as soon as the Shimmer is removed from the dock.

In either case, logging will continue until the Shimmer unit is reset, powered off, replaced in the dock or the battery runs out, whichever happens soonest.

Repeatedly resetting the Shimmer unit will result in multiple logging sessions on the SD card.

Warning: *Please note that it is not recommend to dock the Shimmer unit while it is being configured. Ideally, the Shimmer unit should either be powered off or in standby mode whenever it is being placed on the dock.*

User button (manual start)

This option relies on the orange user button on the *Shimmer3* enclosure to start/stop logging and is enabled if the "Push Button Start" option is enabled from ShimmerLog or ShimmerCapture (*i.e.*, configuration file line entry: "userbutton=1").

With this setting, undocking the Shimmer unit or powering it on off the dock will trigger the same steps 1 and 2 as above (*i.e.*, standby for 3 seconds followed by configuration). However, logging will not start until the user button is pressed by the user.

If the user button is pressed while the Shimmer unit is on the dock, it will have no effect on logging.

If the user button is pressed while the Shimmer unit is undocked, logging will start immediately.

Logging will continue until one of the following occurs: the user button is pressed, the Shimmer unit is reset, powered off, replaced in the dock, the battery runs out or the Shimmer is told to stop by a host application, whichever happens soonest.

Repeatedly pressing the user button and or docking the *Shimmer3* unit (to stop logging) will result in multiple logging sessions on the SD card.

3. Configuring the *Shimmer3*

3.1. Configuration and calibration loading priority

To use this firmware image, the user must provide the desired configuration parameters to each *Shimmer3*. For the *LogAndStream* firmware, configuration parameters are saved in two locations on the *Shimmer3*: a configuration file named *sdlog.cfg* located on the microSD card; and a flash memory location in the *Shimmer3*'s microcontroller called infomem. On initialisation, the *Shimmer3* will attempt to load configuration parameters from one of these sources in order of priority, initially from the configuration file and subsequently from infomem. If there is a problem loading parameters from either of the sources (*e.g.*, in the case of a new *Shimmer3*), the *Shimmer3* will revert to default values - as illustrated in Figure 3-1.

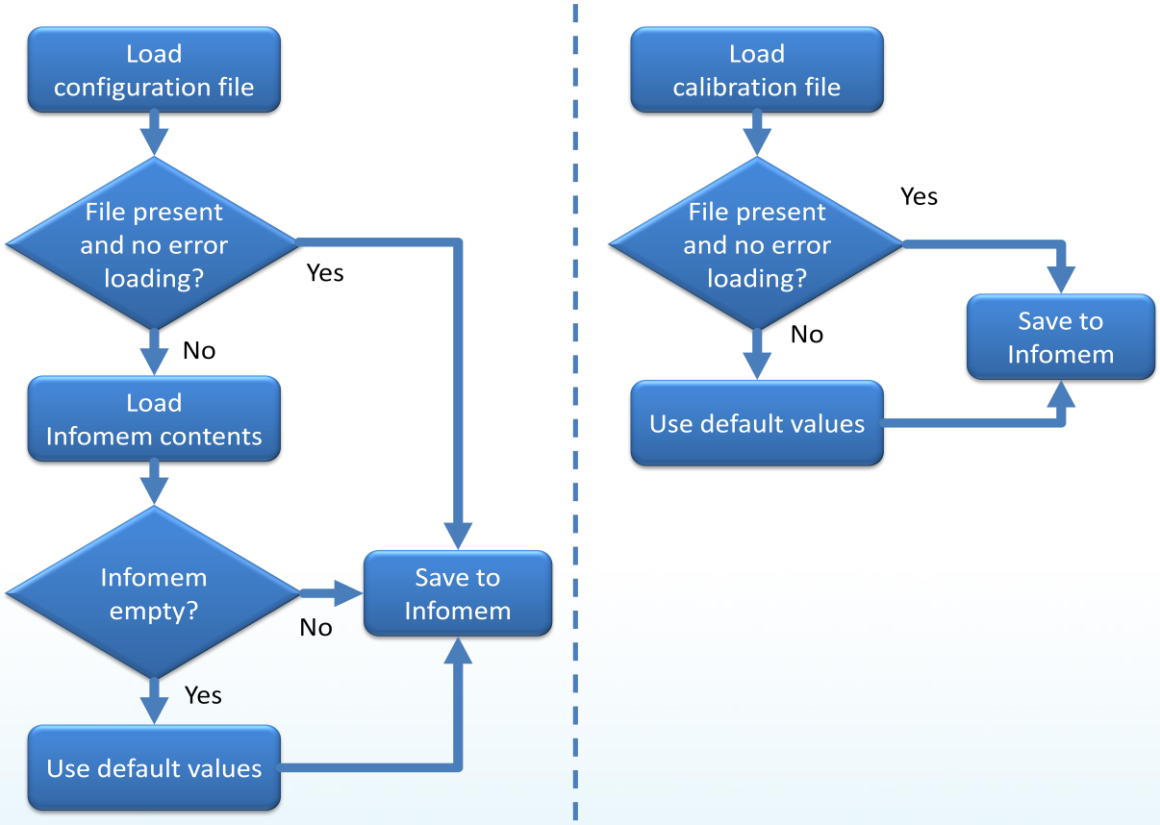

Figure 3-1 Shimmer3 configuration parameter and calibration value loading priority.

It is recommended to use the *ShimmerCapture or ShimmerLog* applications, available for download from the Shimmer website (www.shimmersensing.com), to avoid errors in writing the configuration file. However, it can be written in any text editor and saved to the SD card without the software application, by following the guidelines throughout this section.

The *LogAndStream* firmware is fully compatible with the *ShimmerLog* application. In this case, the user can configure the *Shimmer3* in the same manor as would previously have been done with the SDLog firmware, *i.e.*, with a *Shimmer Dock*. Please refer to the *ShimmerLog User Manual* for more information.

The *ShimmerCapture* application allows for configuration either through the *Shimmer Dock* or wirelessly over Bluetooth. In both cases, when the *Shimmer3* is configured through the *ShimmerCapture* application, the new configuration parameters will be written to the *Shimmer3*'s infomem. If the *Shimmer3* is undocked when it is configured, the *Shimmer3's* microcontroller will automatically try to rewrite the configuration file with the new parameters. If the *Shimmer3* is docked, as SD card access is handed over to the PC, the *ShimmerCapture* application will attempt to write the new configuration file.

Calibration parameters are not used in this firmware image to provide calibrated data, but they are saved in the configuration header of the raw data files on the microSD card. This is so that the data can calibrated post data collection. To supply calibration parameters, the user should create a folder called "Calibration" or "calibration" in the top-level SD card directory and should save the calibration file with the name "calibParams.ini" (name is case sensitive). The format of the file should match that output by the *Shimmer 9DOF Calibration Application* [2].

If a calibration file is found by the firmware, the appropriate parameters from the file will be stored in the configuration header for the enabled sensors. As illustrated previously in Figure 3-1, if no calibration file is found, then the default calibration values will be stored for all sensors. If a calibration file is found but it does not contain parameters for a given enabled sensor, or for the configured sensor range, then default calibration values are stored for that sensor.

> **Note:** When performing calibration with the *Shimmer 9DoF Calibration* application, the calibration data must be saved to the calibration file instead of using the "Save to Shimmer" option. The "Save to Shimmer" options only store the calibration data to the infomem of the *Shimmer3* and do not automatically update the calibration file. Therefore, as illustrated in Figure 3-1, when the *Shimmer3*'s power is cycled or the device is reset, the *LogAndStream* firmware will overwrite the 'new' calibration values in the infomem with the 'old' calibration file contents, if one is present.

3.2. Configuration file

The configuration file is read every time a new logging session is started on the Shimmer; thus, to change the configuration, just modify the *sdlog.cfg* file and reboot the Shimmer (there is no need to reprogram). The configuration file must be saved in the top level directory of the SD card (*i.e.*, not within a subfolder). This file allows configuration of all firmware features, as described throughout this section.

Each new line of the configuration file contains a single configuration parameter command. Users should note that the commands should be written **exactly** as they appear in the following sections so that they will be successfully parsed by the firmware. Any typographical errors, extra whitespaces, or repeated commands will cause parsing errors. It is the user's responsibility to ensure that the *sdlog.cfg* file is well-written. An example configuration file layout can be found in the appendices (*i.e.*, Section 6.2).

[2] For a more detailed description of IMU calibration parameters, refer to the *Shimmer 9DoF Calibration User Manual* and the *Shimmer IMU User Guide*, which are available for download from the members' area of www.shimmersensing.com.

3.3. Enabling sensors

The user may choose which sensors are enabled and disabled on the Shimmer for a particular experiment. The configuration file should contain a new line for each sensor enable/disable command. If any sensor is not included in the configuration file, it will be assumed to be disabled. The currently available sensors for the *Shimmer3*and their corresponding *sdlog.cfg* entries are listed in Table 3-1. There are currently no sensor conflicts for *Shimmer3*.

Sensor		Enable command	Disable command
Low Noise Accelerometer		accel=1	accel=0
Gyroscope		gyro=1	gyro=0
Magnetometer		mag=1	mag=0
Wide Range Accelerometer		accel_d=1	accel_d=0
Battery voltage		vbat=1	vbat=0
External Expansion Channel 7		extch7=1	extch7=0
External Expansion Channel 6		extch6=1	extch6=0
External Expansion Channel 15		extch5=1	extch15=0
Internal Expansion Channel 1		intch1=1	intch1=0
Internal Expansion Channel 12		intch12=1	intch12=0
Internal Expansion Channel 13		intch13=1	intch13=0
Internal Expansion Channel 14		intch14=1	intch14=0
GSR		gsr=1	gsr=0
Pressure & Temperature		pres_bmp180=1	pres_bmp180=0
Pulse/PPG		intch13=1 exp_power=1	intch13=0 exp_power=0
ECG	**24-bit**	exg1_24bit=1 exg2_24bit=1 exp_power=1	exg1_24bit=0 exg2_24bit=0 exp_Power=0
	16-bit	exg1_16bit=1 exg2_16bit=1 exp_power=1	exg1_16bit=0 exg2_16bit=0 exp_Power=0
EMG	**24-bit**	exg1_24bit=1	exg1_24bit=0
	16-bit	exg1_16bit=1	exg1_16bit=0
Strain Gauge		str=1	str=0

Table 3-1 - Sensor enable/disable commands for Shimmer3. ***Note:*** *the ECG and EMG channels can be set at either 16-bit or 24-bit but not both.*

Note: To enable the ExG Test Signal to verify the correct operation of the *Shimmer3 ExG module* see the *ExG User Guide for ECG* or the *ExG User Guide for EMG,* both available for download on www.shimmersensing.com.

3.4. Configuring sensors

The following commands and options can be used to set sensor configuration parameters:

Parameter	Example command	Valid options	Default
Accel range	`acc_range=0`	0 (± 2.0 G) 1 (± 4.0 G) 2 (± 8.0 G) 3 (± 16.0 G)	0 (± 2.0 G)
Gyro range	`gyro_range=0`	0 (±250 dps) 1 (±500 dps) 2 (±1000 dps) 3 (±2000 dps)	0 (±250 dps)
Mag range	`mg_range=1`	1 (±1.3 Ga) 2 (±1.9 Ga) 3 (±2.5 Ga) 4 (±4.0 Ga) 5 (±4.7 Ga) 6 (±5.6 Ga) 7 (±8.1 Ga)	1 (±1.3 Ga)
Accel internal data rate	`acc_internal_rate=1`	0 (power down) 1 (1.0 Hz) 2 (10 Hz) 3 (25 Hz) 4 (50 Hz) 5 (100 Hz) 6 (200 Hz) 7 (400 Hz) 8 (1.620KHz) 9 (1.344 KHz)	5 (100 Hz)
Gyro internal data rate	`gyro_samplingrate=0`	0 – 255 (8000/value -1 Hz)	155 (51.28 Hz)
Mag internal data rate	`mg_internal_rate=0`	0 (0.75 Hz) 1 (1.5 Hz) 2 (3.0 Hz) 3 (7.5 Hz) 4 (15.0 Hz) 5 (30.0 Hz) 6 (75.0 Hz)	6 (75 Hz)
Accel Low Power Mode	`acc_lpm=0`	0 (disabled) 1 (enabled)	0 (disabled)
Accel High Res Mode	`acc_hrm=0`	0 (disabled) 1 (enabled)	0 (disabled)
GSR range	`gs_range=0`	0 (10kOhm – 56kOhm) 1 (56kOhm – 220kOhm) 2 (220kOhm – 680kOhm) 3 (680kOhm – 4.7MOhm) 4 (Auto Range)	4 (Auto Range)
Pressure Resolution	`pres_bmp180_prec=0`	0 (Low) 1 (Standard) 2 (High) 3 (Very High)	0 (Low)

Table 3-2 - Sampling configuration options for Shimmer3.

Note: To change the gain and/or the data rate of the *ExG module*, see the *ExG User Guide for ECG* or the *ExG User Guide for EMG*, both available for download on www.shimmersensing.com.

3.5. Configuring an experiment

Logging preferences can be configured using the commands in Table 3-3. Further details describing these parameters are provided below the table.

Parameter	Example command	Valid options	Default
Sampling rate (in Hz)	`sample_rate=51.2`	0, ..., 1024	51.2
Synchronisation	`sync=0`	0 (no time synchronisation) 1 (master-slave synchronisation)	0
Master	`iammaster=0`	0 (slave) 1 (master)	0
Broadcast interval (s)	`interval=120`	integer value: 54, ..., 255	120
User button	`user_button_enable=0`	0 (disabled) 1 (enabled)	0
Single-touch	`singletouch=0`	0 (disabled) 1 (enabled)	0
Number of Shimmers	`Nshimmer=1`	integer value: 1, ..., 255	1
Shimmer ID number	`myid=1`	integer value: 1, ..., Nshimmer	1
Shimmer name	`shimmername=shimmer1`	string with up to 11 characters	IDxxxx
Experiment ID	`experimentid=expid`	string with up to 11 characters	-
Configuration ID	`configtime=1`	any 32-bit signed integer value	0
Master MAC Address	`center=00066646b6af`	MAC address of the Master BT Module	-

Table 3-3 - Logging configuration options. ***Note:*** *the multi-Shimmer sync features, as highlighted by the shaded rows, are not currently supported by the current version of LogAndStream.*

Sampling rate

The *Sampling rate* is the frequency at which sampling should be performed. It should be noted that the actual sampling rate may not be exactly equal to this parameter because the firmware requires the sampling period to be an integer multiple of 1/32768 s. For example, a desired sampling frequency of 500 Hz requires a sampling period of 2 ms which is equal to 65.536*(1/32768) s. In the firmware, this sampling period will be rounded up to 66*(1/32768) = 2.014 ms, so the true sampling rate will be 496.48 Hz. The actual sampling rate that will be used (*i.e.*, the *True Fs*) can be calculated from the specified sampling rate (*Fs*) using the following formula:

$$True\ Fs = \frac{32768}{ceil\left(\frac{32768}{Desired\ Fs}\right)}$$

where the ceil() function means rounding up to the nearest integer.

Syncronisation, Master, Broadcast interval and Single-touch

As mentioned previously, the multi-Shimmer sync features (usually availble in the *Shimmer3 SDLog* and *BtStream* firmware), are not supported in the current version of *LogAndStream* firmware. For

compatibility reasons, the related configuration file entries (i.e., *Syncronisation*, *Master*, *Broadcast interval* and *Single-touch*) - as highlighted by the shaded rows in *Table 3-3*) are still present in the configuration file but currently have no effect.

User Button

The *User button* refers to the button on the baseboard which is accessible by pressing the circular orange button on the *Shimmer3* enclosure. If the button is enabled, then logging will begin when the user button is pressed. Logging will finish when **either** the user button is pressed again or the Shimmer is placed on the programming dock.

> **Note:** the user button must not be enabled if the Heart Rate sensor is enabled due to conflicts caused by shared pins in hardware.

Number of Shimmers and Shimmer ID number

Number of Shimmers and *Shimmer ID number* are used to describe the total number of Shimmers in an experiment and the ID number of each individual Shimmer. Once again, these parameters do not have any explicit function in firmware.

Shimmer name

The *Shimmer name* parameter allows the user to specify a meaningful name for each device. It is used by the firmware to name the data directories on the SD card. For more information on the directory structure, see Section 5.

Experiment ID

The *Experiment ID* parameter allows the user to specify a meaningful name for an experiment. It is used by the firmware to name the data directories on the SD card. For more information on the directory structure, see Section 5.

Configuration ID

The *Configuration ID* is a numeric identifier for the experiment. It typically refers to the date and/or time at which the configuration file was created (but users may choose any valid numeric value that is meaningful to them). It is appended to the experiment ID name in the data directories to allow the user to identify different experiments in the case that the same experiment ID name is used multiple times. It is vital that this parameter is equal for all Shimmers in the experiment if synchronisation or single-touch start is enabled.

3.6. Infomem contents

The configuration parameter values are stored by the *Shimmer3* in the first 90 bytes of the infomem, which is the part of the *Shimmer3* memory that survives a reset or power cycle but is overwritten when the *Shimmer3* is reprogrammed. The format of the configuration data stored in infomem is as follows:

Infomem Byte	Contents
0 - 1	Sampling rate
2	Buffer size
3 - 5	Selected sensors
6 - 9	Config bytes (Allows for 32 individual boolean settings)
10 - 30	Low Noise Accelerometer calibration values
31 - 51	Gyroscope calibration values
52 - 72	Magnetometer calibration values
73 - 93	Wide Range Accelerometer calibration values

Table 3-4 Informem layout overview.

Selected Sensors - Infomem Bytes 3 to 5

The *Selected sensors* bytes have a single bit assigned to each sensor as follows:

	Bit	Property
Infomem Byte 3	7	Low Noise Accelerometer.
	6	Gyroscope.
	5	Magnetometer.
	4	ExG1_24BIT.
	3	ExG2_24BIT.
	2	GSR.
	1	External Expansion ADC Channel 7.
	0	External Expansion ADC Channel 6.
Infomem Byte 4	7	Strain Gauge.
	6	Not yet assigned.
	5	Battery Monitor.
	4	Wide Range Accelerometer.
	3	External Expansion ADC Channel 15.
	2	Internal Expansion ADC Channel 1.
	1	Internal Expansion ADC Channel 12.
	0	Internal Expansion ADC Channel 13.
Infomem Byte 5	7	Internal Expansion ADC Channel 14.
	6	MPU9150 Accelerometer.
	5	MPU9150 Magnetometer.
	4	ExG1_16BIT.
	3	ExG2_16BIT.
	2	BMP180 Pressure.
	1	BMP180 Temperature.
	0	MSP430 Temperature.

Config Bytes - Infomem Bytes 6 to 9

The *Config bytes* contain the following parameters:

Infomem Byte 6 - Config Setup Byte 0	
Bits 7 – 4	Wide Range (LSM303DLHC) Accelerometer Data Rate.
Bits 3 – 2	Wide Range (LSM303DLHC) Accelerometer Range.
Bit 1	Wide Range (LSM303DLHC) Accelerometer Low Power Mode.
Bit 0	Wide Range (LSM303DLHC) Accelerometer High Resolution Mode.
Infomem Byte 7 - Config Setup Byte 1	
Bits 7 – 0	MPU9150 Data Rate.
Infomem Byte 8 - Config Setup Byte 2	
Bits 7 – 5	(LSM303DLHC) Magnetometer Range.
Bits 4 – 2	(LSM303DLHC) Magnetometer Data Rate.
Bit 1 - 0	MPU9150 Gyroscope Range.
Infomem Byte 9 - Config Setup Byte 3	
Bits 7 – 6	MPU9150 Accelerometer Range.
Bits 5 – 4	BMP180 Pressure Resolution.
Bit 3 - 1	GSR Range
Bit 0	Internal Expansion Power Enable

Calibration Parameters - Infomem Bytes 10 to 93

The calibration parameters for the inertial measurement units (accelerometer, gyroscope and magnetometer) consist of a three-element offset bias vector, a three-element sensitivity vector and a 3x3-element alignment matrix.[3] The structure of these values when they are sent to/from the *Shimmer3* and stored in infomem is as follows:

- Each of the 3 offset bias vector values are stored as 16-bit signed integers (big endian) and are contained in bytes 0-5.
- Each of the 3 sensitivity vector values are stored as 16-bit signed integers (big endian) and are contained in bytes 6-11.
- Each of the 9 alignment matrix values are stored as 8-bit signed integers and are contained in bytes 12-20.

[3] For a more detailed description of IMU calibration parameters, refer to the *Shimmer 9DoF Calibration User Manual* and the *Shimmer IMU User Guide*.

4. Bluetooth Streaming

The first byte of every packet received by the *Shimmer3* or the "host" is an identifier, telling the receiver what action to carry out or how to interpret the subsequent bytes. The full list of identifiers that are used to interface with the *BtStream* application, can be found in the header file, *Shimmer.h*, which can be found in the Appendix in Section 6.3 of this document (most recent version available online).

For every packet that the *Shimmer3* receives, it sends an acknowledgement message (ACK_COMMAND_PROCESSED) back to the host, to acknowledge receipt of the command.

4.1. Set Commands

The "SET" commands are used to set the values of all of the configurable parameters:

- Enabled sensors.
- Sampling rate.
- Accelerometer, gyroscope, magnetometer range.
- Accelerometer, gyroscope, magnetometer data rate.
- Battery monitoring.
- Calibration parameters for Accelerometers, Gyroscope, Magnetometer.
- Blink LED.

The packets sent between the *Shimmer3* and the PC for a SET command are shown in Figure 4-1.

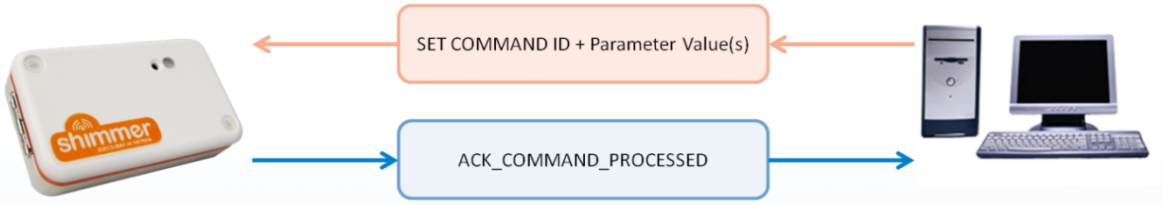

Figure 4-1 Packets sent for SET commands

These commands require that further data be received by the *Shimmer3* after the identifier byte. e.g. the SET_SAMPLING_RATE_COMMAND identifier must be followed by a one-byte value representing the sampling rate that the *Shimmer3* is to use. Another example is the SET_A_ACCEL_CALIBRATION_COMMAND identifier, which must be followed by 21 bytes representing the accelerometer calibration parameters.

4.2. Get Commands

The "GET" commands are requests for information and require that the *Shimmer3* sends data back to the host. The packets sent between the *Shimmer3* and the PC for a SET command are shown in Figure 4-2.

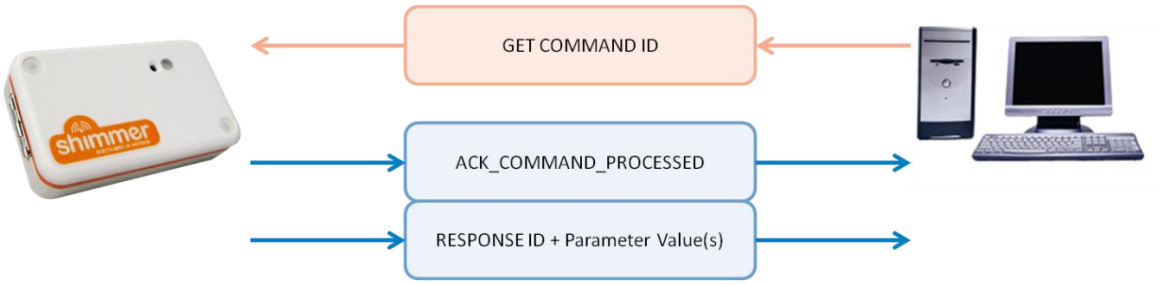

Figure 4-2 Packets sent for GET commands

On receipt of a GET command, the *Shimmer3* will send an acknowledgement message and, then, it will prepare and send a packet containing the appropriate response identifier byte, followed by the data that was requested.

For example, when the *Shimmer3* receives the GET_SAMPLING_RATE_COMMAND, it must send the current value of the sampling rate back to the host; the response packet will contain two bytes: the first byte will be the SAMPLING_RATE_REPONSE identifier and the second byte will be the sampling rate value.

Similarly, if the *Shimmer3* receives a GET_A_ACCEL_CALIBRATION_COMMAND, it will send a packet whose first byte is the A_ACCEL_CALIBRATION_RESPONSE identifier, followed by 21 bytes representing the accelerometer calibration parameters.

The INQUIRY_COMMAND

The INQUIRY_COMMAND is issued by the "host" when it wants to know the entire configuration of the *Shimmer3*, like what is the sampling rate, what is the buffer size, to which channel is each enabled sensor assigned, etc. In response to this command, the *Shimmer3* will send a packet back to the "host" with the structure shown in Table 4-1.

Byte	**0**	**1-2**	**3-6**	**7**	**8**	**9**	**10**	**...**	**x**
Value	Packet Type	Sampling rate	Config Bytes 0-3	Num Chans	Buffer size	Chan1	Chan2	...	ChanX

Table 4-1 Inquiry response packet format

where the Packet Type = INQUIRY_RESPONSE and the value in the channel fields (Chan1, Chan2, ..., ChanX) indicate exactly what data from which sensor will be contained in the equivalent field of the data packet. The total number of bytes sent by the *Shimmer3* will depend on how many data channels are active (i.e. which sensors are enabled).

Signal name, byte values and datatypes

Table 4-2 lists the values in the channel contents bytes of the Inquiry response packet along with the signal names and datatypes for the equivalent sensor signals (* in the datatype column denotes MSB first; otherwise LSB first).

Signal Name	Byte Value	Signal Datatype
Low Noise Accelerometer X*	0	u12
Low Noise Accelerometer Y*	1	u12
Low Noise Accelerometer Z*	2	u12

Battery	3	u12
Wide Noise Accelerometer X*	4	i16
Wide Noise Accelerometer Y*	5	i16
Wide Noise Accelerometer Z*	6	i16
Gyroscope X*	7	i16*
Gyroscope Y*	8	i16*
Gyroscope Z*	9	i16*
Magnetometer X*	A	i16*
Magnetometer Y*	B	i16*
Magnetometer Z*	C	i16*
External ADC 7	D	u12
External ADC 6	E	u12
External ADC 15	F	u12
Internal ADC 1	10	u12
Internal ADC 12	11	u12
Internal ADC 13	12	u12
Internal ADC 14	13	u12
BMP180 Temperature*	1A	u16*
BMP180 Pressure*	1B	u24*
GSR Raw	1C	u16
ExG_ADS1292R_1_STATUS	1D	u8
ExG_ADS1292R_1_CH1_24BIT	1E	i24*
ExG_ADS1292R_1_CH2_24BIT	1F	i24*
ExG_ADS1292R_2_STATUS	20	u8
ExG_ADS1292R_2_CH1_24BIT	21	i24*
ExG_ADS1292R_2_CH2_24BIT	22	i24*
ExG_ADS1292R_1_CH1_16BIT	23	i16*
ExG_ADS1292R_1_CH2_16BIT	24	i16*
ExG_ADS1292R_2_CH1_16BIT	25	i16*
ExG_ADS1292R_2_CH2_16BIT	26	i16*
Strain Gauge High	27	u12
Strain Gauge Low	28	u12

Table 4-2 Signal names, channel contents byte values and datatypes for available sensor signals

4.3. Action Commands

There are a number of available "ACTION" commands, which do not require that parameter values be sent between the PC and the *Shimmer3* but, instead, tell the *Shimmer3* what action it is to carry out. These include the START_STREAMING_COMMAND and STOP_STREAMING_COMMAND and the TOGGLE_LED_COMMAND.

4.4. Streaming

When the START_STREAMING_COMMAND is received by the *Shimmer3*, it will send an acknowledge message back to the "host" and start sampling sensor data. As the sensor data is sampled, the *Shimmer3* will prepare data packets and send them to the "host" over Bluetooth.

The *Buffer size* parameter determines the number of samples that are sent together in a single data packet. The structure of the data packet with *Buffer size = 2* is shown in Table 4-3, where Packet Type = DATA_PACKET, *TS* denotes "Timestamp" and *Ch* denotes "Channel".

Byte	**0**	**1 - 2**	**3 - 4**	**5 - 6**	**...**	**(x-1) - x**	**(x+1) - (x+2)**	**(x+3) - (x+4)**	**(x+5) - (x+6)**	**...**	**(2x-1) - 2x**
Value	Packet Type	TS	Ch1	Ch2	...	ChX	TS	Ch1	Ch2	...	ChX
		Sample 1					**Sample 2**				

Table 4-3 Data packet structure (Buffer size =2)

If *Buffer size* were equal to 1, then the data packet would contain only one timestamp and one sample from each channel (i.e. the bytes denoted "Sample 1" in Table 4-3. If *Buffer size* were any integer value greater than 2, then subsequent timestamps and sample values for each channel would be appended at the end of the packet until the number of samples equals the buffer size.

Sensor data will continue to be sampled and streamed until a STOP_STREAMING_COMMAND is received by the *Shimmer3*.

By default, the application will sample the 3-axis accelerometer at 51.2 Hz and send the data using a data buffer of size 1.

4.5. Bluetooth Latency

Our lab tests have shown up to 100 ms of latency with considerable variation (> 50 ms). These measures result from multiple FIFOs in the data path, as expected in wireless data acquisition systems using conventional computing devices for the data end-points. Actual performance is strongly impacted by end-point system configuration and load.

5. Reading the SD card Data

It is recommended that the *ShimmerLog* software, which can be downloaded from the Shimmer website (www.shimmersensing.com) be used to read the data from the SD card. However, using the following guidelines, the user may use their platform of choice to read and parse the data.

5.1. SD card directory structure

The data directory structure on the SD card is as follows:

- A folder called "*data*" is created by the firmware in the SD card top-level directory if it does not already exist.
- A subfolder is created within the *data* folder each time a new "*experiment id*" is encountered in the *sdlog.cfg* file (default is "*default_exp*").
- A new subfolder is created within the appropriate experiment folder each time logging starts. The naming convention for this folder is the Shimmer name specified in the *sdlog.cfg* file (default is the abbreviated form of the *Shimmer3* id number (e.g. *IDbf5e*)), followed by a three digit number which is sequentially incremented at new logging session.
- For example, the first time a *Shimmer3* with name "*device1*" in experiment "*experiment1*" logs data to the SD card it will create the folder *data/experiment1/device1-000/*. The next time it stops and subsequently restarts logging, it will create *data/experiment1/device1-001/*, etc.
- Within this subfolder, data is logged in files which are named sequentially with a three digit number indicating the order of logging, starting with *000*. The file is closed after 1 hour of continuous data logging and a new file is opened in the same subfolder, such that the second file is called *001* and so on. The last file is closed when logging stops and a new file (in a different folder) is created the next time logging begins.

5.2. Parsing the raw data file

Raw data file configuration header

Each RAW data file (e.g. *data/experiment1/device1-000/000*), begins with a header which contains the *Shimmer3* configuration information. The structure of this header is outlined in the tables below, where each row of each table represents one byte and the individual bits are separated, where appropriate, depending on whether each bit represents an independent binary value or the entire byte contains a single value. Empty cells and cells with constant values represent bits that are reserved for future use. For *Shimmer3*, there are a total 256 bytes in the header.

Byte # 0 – 9: Enabled Sensors

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	ADC Sample Rate (LSB)							
1	ADC Sample Rate (MSB)							
2								

3	Acc (LN)*	Gyr	Mag	EXG1_24BIT	EXG2_24BIT	GSR	ExtCh 7	ExtCh 6
4	strain		Battery	Acc (WR)*	ExtCh 15	IntCh 1	IntCh 12	IntCh13
5	IntCh 14	MPU9150_ACCEL	MPU9150_MAG	EXG1_16BIT	EXG2_16BIT	pres		
6								
7								
8	LSM303 Digtial Accel Rate				D Accel Range		Accel LPM	Accel HRM
9	MPU9150 Gyro Rate							

* There are multiple accelerometers on the *Shimmer3* base board. The low noise (LN) analog accelerometer on the KXRB5-2042 chip is enabled via Byte 3, Bit 7 (Acc (LN)), whilst the wide range digital accelerometer on the LSM303DLHC chip is enabled via Byte 4, Bit 4 (Acc (WR)).

Byte # 10 – 19: Trial Configuration

Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
10	LSM303 Mag Range			LSM303 Mag Rate			MPU9150 Gyro Range	
11	MPU_ACCEL_RANGE		Pressure Precision		GSR Range			EXP_PWR
12								
13								
14								
15								
16			User button		1	Sync	Master	0
17	Single touch	Accel LPM	Accel HRM	txco				
18	Broadcast interval							
19								

Byte # 30 – 39: Firmware and Shimmer Parameters

Byte	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
30	ShimmerVersion (MSB)							
31	ShimmerVersion (LSB)							
32	MyTriald							
33	Nshimmer							
34	FW Version - Type (MSB)							
35	FW Version - Type (LSB)							
36	FW Version - Major (MSB)							
37	FW Version - Major (LSB							
38	FW Version - Minor							
39	FW Version - Release							

Shimmer Version indicates the hardware version for which the code was compiled. *Shimmer3* is denoted by *3*.

The *FW Version* bytes define the firmware version. The *Type* field will always be 2 for *SDLog* firmware images. *Major* and *Minor* versions are indicated by a two- and one-byte value, respectively

(e.g. for LogAndStream*g v1.0*, *Major* = 1 and *Minor* = 0). The *Release* field can be ignored by users; it will have a value of 0.

Byte # 40 – 51: Reserved for future use

Byte # 52 - 55: Configuration Time

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
52	Config Time 0 (MSB)							
53	Config Time 1							
54	Config Time 2							
55	Config Time 3 (LSB)							

This value comes from the *Configuration ID* parameter from Section 3.5.

Byte # 56-181: Calibration parameters

Bytes 56 to 181 store the calibration data for individual *Shimmer3* sensors. These parameters are used to calibrate the RAW data during post-experimentation analysis. The layout and description of these bytes can be found in the Appendix section of this document (*i.e.*, Section 6.2).

Byte # 182 - 251: Reserved for future use

Byte # 252 - 255: Initial Timestamp

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
252	Initial time stamp							
253	Initial time stamp							
254	Initial time stamp							
255	Initial time stamp							

Immediately after the configuration header, there is a 4-byte timestamp, whose value is the time on the local *Shimmer3* clock (in units of ticks of the 32 kHz clock) when the first sample in that file was recorded. It is used for aligning the data from multiple Shimmer devices to a common clock in multi-shimmer sync applications. As the multi-shimmer sync feature is not available in the current LogAndStream firmware, these bytes can be ignored.

Raw data file format

Following these 256 bytes, the logged sensor data begins (i.e. at the 257th byte in the file). The data is written in blocks of up to 512 bytes. These blocks are continuously written until data has been logged to a given file for one hour, at which time the current file is closed and a new file is opened, with the same format as the current one (i.e. the configuration header is repeated in every file).

The exact number of bytes per block and the interpretation of those bytes of data depend on the enabled sensors. The total number of bytes written per block, *Bp*, is given by:

$$Bp = N \times (Bs + 2)$$

where:

Bs = *Nc3* * 3 + *Nc2* * 2 + *Nc1*, is the number of bytes per sample, which depends on the number of 3byte channels (*Nc3*), number of 2byte channels (*Nc2*) and the number of 1-byte channels (*Nc1*) required by the enabled sensors;

N = floor(512/(*Bs* + 2)), is the integer number of samples per block, with a buffer of 512 bytes and the extra 2 bytes per sample are the 16-bit timestamps from the *Shimmer3*'s local 32 kHz clock.

All of the bytes in the block are interleaved local 16-bit timestamps and the associated sensor samples. For example, with the accelerometer and the gyroscope enabled, there are three 2-byte channels for the accelerometer and three 2-byte channels for the gyroscope in each sample and the data would be as shown in the following table until a total of Bp = 504 bytes is reached (Bs = 12; N = floor((512)/(12 + 2)) = 36).

Byte	Parameter	Sensor
0	timestamp 0 byte 0 (MSB)	
1	timestamp 0 byte 1 (LSB)	
2	sample 0 channel 0 byte 0 (MSB)	(XAccel)
3	sample 0 channel 0 byte 1 (LSB)	
4	sample 0 channel 1 byte 0 (MSB)	(YAccel)
5	sample 0 channel 1 byte 1 (LSB)	
6	sample 0 channel 2 byte 0 (MSB)	(ZAccel)
7	sample 0 channel 2 byte 1 (LSB)	
8	sample 0 channel 3 byte 0 (MSB)	(XGyro)
9	sample 0 channel 3 byte 1 (LSB)	
10	sample 0 channel 4 byte 0 (MSB)	(YGyro)
11	sample 0 channel 4 byte 1 (LSB)	
12	sample 0 channel 5 byte 0 (MSB)	(ZGyro)
13	sample 0 channel 5 byte 1 (LSB)	
14	timestamp 1 byte 0 (MSB)	
15	timestamp 1 byte 1 (LSB)	
16	sample 1 channel 0 byte 0 (MSB)	(XAccel)
17	sample 1 channel 0 byte 1 (LSB)	
...	...	...

Table 5-1 - Example of the raw data format for a file stored to the SD card of a Shimmer3.

Order of raw data channels

The order in which the sensors appear in the data channels depends on which sensors are enabled. The firmware will assign the channels in the order outlined in Table 5-2 (top to bottom). Note that “Accel (LN)” refers to the low noise accelerometer, whilst “Accel (WR)” refers to the wide range accelerometer. The number of channels per sensor is listed beside the channel name. The endianess of the bytes for each channel is also specified.

Channel	Channel contents	Number of	Bytes per	Endian

type		channels	channel	
Analog channels	Accel (LN)	3	2	little
	Battery	1	2	little
	Ext Exp A7	1	2	little
	Ext Exp A6	1	2	little
	Ext Exp A15	1	2	little
	Int Exp A12	1	2	little
	Int Exp A13	1	2	little
	Int Exp A14	1	2	little
	Strain High	1	2	little
	Strain Low	1	2	little
	Int Exp A1	1	2	little
Digital channels	Gyro_mpu	3	2	big
	Accel_lsm (WR)	3	2	little
	Mag_lsm	3	2	big
	Accel_mpu	3	2	big
	Mag_mpu	3	2	little
	Temperature_bmp180	1	2	big
	Pressure_bmp180	1	3	big
	EXG1(24-bit/16-bit)	1	7/5	big
	EXG2(24-bit/16-bit)	1	7/5	big

Table 5-2 - Order of data channels for Shimmer3 (top to bottom).

6. Appendices

6.1. Example of the *Shimmer3* configuration file (sdlog.cfg)

```
accel=1
gyro=1
mag=1
accel_d=1
vbat=1
extch7=0
extch6=0
extch15=0
intch1=0
intch12=0
intch13=0
intch14=0
gsr=0
str=0
pres_bmp180=0
exg1_24bit=0
exg2_24bit=0
exg1_16bit=0
exg2_16bit=0
EXG_ADS1292R_1_CONFIG1=0
EXG_ADS1292R_1_CONFIG2=163
EXG_ADS1292R_1_LOFF=16
EXG_ADS1292R_1_CH1SET=69
EXG_ADS1292R_1_CH2SET=5
EXG_ADS1292R_1_RLD_SENS=0
EXG_ADS1292R_1_LOFF_SENS=0
EXG_ADS1292R_1_LOFF_STAT=0
EXG_ADS1292R_1_RESP1=2
EXG_ADS1292R_1_RESP2=1
EXG_ADS1292R_2_CONFIG1=2
EXG_ADS1292R_2_CONFIG2=163
EXG_ADS1292R_2_LOFF=16
EXG_ADS1292R_2_CH1SET=5
EXG_ADS1292R_2_CH2SET=5
EXG_ADS1292R_2_RLD_SENS=0
EXG_ADS1292R_2_LOFF_SENS=0
EXG_ADS1292R_2_LOFF_STAT=0
EXG_ADS1292R_2_RESP1=2
EXG_ADS1292R_2_RESP2=1
exp_power=0
acc_range=1
acc_internal_rate=5
acc_lpm=0
acc_hrm=0
gyro_samplingrate=155
gyro_range=1
mg_internal_rate=6
mg_range=1
user_button_enable=1
```

```
sample_rate=51.2
iammaster=0
sync=1
interval=120
singletouch=0
center=00066646b6af
myid=1
Nshimmer=2
shimmername=device1
experimentid=expidname
configtime=1234567
```

6.2. SD data file configuration header - sensor calibration bytes

Byte # 56-76: ExG Calibration

For *Shimmer3*, these parameters refer to the ExG module which is made up of two ADS1292R chips.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
56	NV_EXG_ADS1292R_1_CONFIG1							
57	NV_EXG_ADS1292R_1_CONFIG2							
58	NV_EXG_ADS1292R_1_LOFF							
59	NV_EXG_ADS1292R_1_CH1SET							
60	NV_EXG_ADS1292R_1_CH2SET							
61	NV_EXG_ADS1292R_1_RLD_SENS							
62	NV_EXG_ADS1292R_1_LOFF_SENS							
63	NV_EXG_ADS1292R_1_LOFF_STAT							
64	NV_EXG_ADS1292R_1_RESP1							
65	NV_EXG_ADS1292R_1_RESP2							
66	NV_EXG_ADS1292R_2_CONFIG1							
67	NV_EXG_ADS1292R_2_CONFIG2							
68	NV_EXG_ADS1292R_2_LOFF							
69	NV_EXG_ADS1292R_2_CH1SET							
70	NV_EXG_ADS1292R_2_CH2SET							
71	NV_EXG_ADS1292R_2_RLD_SENS							
72	NV_EXG_ADS1292R_2_LOFF_SENS							
73	NV_EXG_ADS1292R_2_LOFF_STAT							
74	NV_EXG_ADS1292R_2_RESP1							
75	NV_EXG_ADS1292R_2_RESP2							

Byte # 76-96: Digital Accelerometer Calibration

For *Shimmer3*, these parameters refer to the LSM303DLHC accelerometer. There are other accelerometers also populated on the *Shimmer3* circuit board; the associated calibration parameters for the other devices are stored in a later section of the configuration header.

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
76	Digital Accel Calibration Offset X (MSB)							
77	Digital Accel Calibration Offset X (LSB)							

78	Digital Accel Calibration Offset Y (MSB)
79	Digital Accel Calibration Offset Y (LSB)
80	Digital Accel Calibration Offset Z (MSB)
81	Digital Accel Calibration Offset Z (LSB)
82	Digital Accel Calibration Gain X (MSB)
83	Digital Accel Calibration Gain X (LSB)
84	Digital Accel Calibration Gain Y (MSB)
85	Digital Accel Calibration Gain Y (LSB)
86	Digital Accel Calibration Gain Z (MSB)
87	Digital Accel Calibration Gain Z (LSB)
88	Digital Accel Calibration Align XX
89	Digital Accel Calibration Align XY
90	Digital Accel Calibration Align XZ
91	Digital Accel Calibration Align YX
92	Digital Accel Calibration Align YY
93	Digital Accel Calibration Align YZ
94	Digital Accel Calibration Align ZX
95	Digital Accel Calibration Align ZY
96	Digital Accel Calibration Align ZZ

Byte # 97 - 117: Gyroscope Calibration

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
97	Gyro Calibration Offset X (MSB)							
98	Gyro Calibration Offset X (LSB)							
99	Gyro Calibration Offset Y (MSB)							
100	Gyro Calibration Offset Y (LSB)							
101	Gyro Calibration Offset Z (MSB)							
102	Gyro Calibration Offset Z (LSB)							
103	Gyro Calibration Gain X (MSB)							
104	Gyro Calibration Gain X (LSB)							
105	Gyro Calibration Gain Y (MSB)							
106	Gyro Calibration Gain Y (LSB)							
107	Gyro Calibration Gain Z (MSB)							
108	Gyro Calibration Gain Z (LSB)							
109	Gyro Calibration Align XX							
110	Gyro Calibration Align XY							
111	Gyro Calibration Align XZ							
112	Gyro Calibration Align YX							
113	Gyro Calibration Align YY							
114	Gyro Calibration Align YZ							
115	Gyro Calibration Align ZX							
116	Gyro Calibration Align ZY							
117	Gyro Calibration Align ZZ							

Byte # 118 - 138: Magnetometer calibration

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

118	Mag Calibration Offset X (MSB)
119	Mag Calibration Offset X (LSB)
120	Mag Calibration Offset Y (MSB)
121	Mag Calibration Offset Y (LSB)
122	Mag Calibration Offset Z (MSB)
123	Mag Calibration Offset Z (LSB)
124	Mag Calibration Gain X (MSB)
125	Mag Calibration Gain X (LSB)
126	Mag Calibration Gain Y (MSB)
127	Mag Calibration Gain Y (LSB)
128	Mag Calibration Gain Z (MSB)
129	Mag Calibration Gain Z (LSB)
130	Mag Calibration Align XX
131	Mag Calibration Align XY
132	Mag Calibration Align XZ
133	Mag Calibration Align YX
134	Mag Calibration Align YY
135	Mag Calibration Align YZ
136	Mag Calibration Align ZX
137	Mag Calibration Align ZY
138	Mag Calibration Align ZZ

Byte # 139 - 159: Analog Accelerometer Calibration

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
139	Analog Accel Calibration Offset X (MSB)							
140	Analog Accel Calibration Offset X (LSB)							
141	Analog Accel Calibration Offset Y (MSB)							
142	Analog Accel Calibration Offset Y (LSB)							
143	Analog Accel Calibration Offset Z (MSB)							
144	Analog Accel Calibration Offset Z (LSB)							
145	Analog Accel Calibration Gain X (MSB)							
146	Analog Accel Calibration Gain X (LSB)							
147	Analog Accel Calibration Gain Y (MSB)							
148	Analog Accel Calibration Gain Y (LSB)							
149	Analog Accel Calibration Gain Z (MSB)							
150	Analog Accel Calibration Gain Z (LSB)							
151	Analog Accel Calibration Align XX							
152	Analog Accel Calibration Align XY							
153	Analog Accel Calibration Align XZ							
154	Analog Accel Calibration Align YX							
155	Analog Accel Calibration Align YY							
156	Analog Accel Calibration Align YZ							
157	Analog Accel Calibration Align ZX							
158	Analog Accel Calibration Align ZY							
159	Analog Accel Calibration Align ZZ							

Byte # 160 - 181: Temperature (BMP180) and Pressure Calibration

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
160	Temp & Pres Calibration AC1_MSB							
161	Temp & Pres Calibration AC1_LSB							
162	Temp & Pres Calibration AC2_MSB							
163	Temp & Pres Calibration AC2_LSB							
164	Temp & Pres Calibration AC3_MSB							
165	Temp & Pres Calibration AC3_LSB							
166	Temp & Pres Calibration AC4_MSB							
167	Temp & Pres Calibration AC4_LSB							
168	Temp & Pres Calibration AC5_MSB							
169	Temp & Pres Calibration AC5_LSB							
170	Temp & Pres Calibration AC6_MSB							
171	Temp & Pres Calibration AC6_LSB							
172	Temp & Pres Calibration B1_MSB							
173	Temp & Pres Calibration B1_LSB							
174	Temp & Pres Calibration B2_MSB							
175	Temp & Pres Calibration B2_LSB							
176	Temp & Pres Calibration MB_MSB							
177	Temp & Pres Calibration MB_LSB							
178	Temp & Pres Calibration MC_MSB							
179	Temp & Pres Calibration MC_LSB							
180	Temp & Pres Calibration MD_MSB							
181	Temp & Pres Calibration MD_LSB							

6.3. LogAndStream source code header file (Shimmer.h)

```
/*
 * Copyright (c) 2013, Shimmer Research, Ltd.
 * All rights reserved
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions are
 * met:
 *
 *    * Redistributions of source code must retain the above copyright
 *      notice, this list of conditions and the following disclaimer.
 *    * Redistributions in binary form must reproduce the above
 *      copyright notice, this list of conditions and the following
 *      disclaimer in the documentation and/or other materials provided
 *      with the distribution.
 *    * Neither the name of Shimmer Research, Ltd. nor the names of its
 *      contributors may be used to endorse or promote products derived
 *      from this software without specific prior written permission.
 *    * You may not use or distribute this Software or any derivative works
 *      in any form for commercial purposes with the exception of commercial
 *      purposes when used in conjunction with Shimmer products purchased
 *      from Shimmer or their designated agent or with permission from
 *      Shimmer.
 *      Examples of commercial purposes would be running business
 *      operations, licensing, leasing, or selling the Software, or
 *      distributing the Software for use with commercial products.
```

```
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 * "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
 * LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
 * A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
 * OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
 * LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
 * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
 * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
 * OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */
#ifndef SHIMMER_BTSD_H
#define SHIMMER_BTSD_H

//these are defined in the Makefile for BtStream (TinyOS)
#define DEVICE_VER          3      //Represents SR30. 0-3 for shimmer1 to shimmer3
#define FW_IDENTIFIER       3      //Two byte firmware identifier number:  3 for BTSD,
                                   //2 for SDLog, 1 for BTStream,
#define FW_VER_MAJOR        0      //Maor version number: 0-65535
#define FW_VER_MINOR 0             //Minor version number: 0-255
#define FW_VER_INTERNAL     15     //internal version number: 0-255

typedef enum {
       BTSD_DOCKED = 0,
       BTSD_UNDOCKED_BT,
       BTSD_UNDOCKED_SYNC,
       BTSD_STREAMING,
       BTSD_STREAMING_LOGGING,
       BTSD_LOGGING,
       BTSD_WAITING_USR
} BTSD_STATE;

typedef uint8_t bool;
#define TRUE 1
#define FALSE 0

typedef uint8_t error_t;
#define SUCCESS 1
#define FAIL 0

// Packet Types// Packet Types
#define DATA_PACKET                                0x00
#define INQUIRY_COMMAND                            0x01
#define INQUIRY_RESPONSE                           0x02
#define GET_SAMPLING_RATE_COMMAND                  0x03
#define SAMPLING_RATE_RESPONSE                     0x04
#define SET_SAMPLING_RATE_COMMAND                  0x05
#define TOGGLE_LED_COMMAND                         0x06
#define START_STREAMING_COMMAND                    0x07  //maintain compatibility
with Shimmer2/2r BtStream
#define SET_SENSORS_COMMAND                        0x08
#define SET_LSM303DLHC_ACCEL_RANGE_COMMAND         0x09
#define LSM303DLHC_ACCEL_RANGE_RESPONSE            0x0A
#define GET_LSM303DLHC_ACCEL_RANGE_COMMAND         0x0B
#define SET_CONFIG_SETUP_BYTES_COMMAND             0x0E
#define CONFIG_SETUP_BYTES_RESPONSE                0x0F
#define GET_CONFIG_SETUP_BYTES_COMMAND             0x10
#define SET_A_ACCEL_CALIBRATION_COMMAND            0x11
#define A_ACCEL_CALIBRATION_RESPONSE               0x12
#define GET_A_ACCEL_CALIBRATION_COMMAND            0x13
```

```
#define SET_MPU9150_GYRO_CALIBRATION_COMMAND          0x14
#define MPU9150_GYRO_CALIBRATION_RESPONSE             0x15
#define GET_MPU9150_GYRO_CALIBRATION_COMMAND          0x16
#define SET_LSM303DLHC_MAG_CALIBRATION_COMMAND        0x17
#define LSM303DLHC_MAG_CALIBRATION_RESPONSE           0x18
#define GET_LSM303DLHC_MAG_CALIBRATION_COMMAND        0x19
#define SET_LSM303DLHC_ACCEL_CALIBRATION_COMMAND      0x1A
#define LSM303DLHC_ACCEL_CALIBRATION_RESPONSE         0x1B
#define GET_LSM303DLHC_ACCEL_CALIBRATION_COMMAND      0x1C
#define STOP_STREAMING_COMMAND                        0x20//maintain compatibility
                                                       //with Shimmer2/2r BtStream
#define SET_GSR_RANGE_COMMAND                         0x21
#define GSR_RANGE_RESPONSE                            0x22
#define GET_GSR_RANGE_COMMAND                         0x23
#define DEPRECATED_GET_DEVICE_VERSION_COMMAND         0x24
                                  //deprecated because 0x24 ('$' ASCII) as a command
                                  //is problematic if remote config is enabled in
                                  //RN42 Bluetooth module. Replaced with 0x3F command
#define DEVICE_VERSION_RESPONSE                       0x25
                                  //maintain compatibility with Shimmer2/2r BtStream
#define GET_ALL_CALIBRATION_COMMAND                   0x2C
#define ALL_CALIBRATION_RESPONSE                      0x2D
#define GET_FW_VERSION_COMMAND                        0x2E
                                  //maintain compatibility with Shimmer2/2r BtStream
#define FW_VERSION_RESPONSE                           0x2F
                                  //maintain compatibility with Shimmer2/2r BtStream
#define SET_CHARGE_STATUS_LED_COMMAND                 0x30
#define CHARGE_STATUS_LED_RESPONSE                    0x31
#define GET_CHARGE_STATUS_LED_COMMAND                 0x32
#define BUFFER_SIZE_RESPONSE                          0x35
#define GET_BUFFER_SIZE_COMMAND                       0x36
#define SET_LSM303DLHC_MAG_GAIN_COMMAND               0x37
#define LSM303DLHC_MAG_GAIN_RESPONSE                  0x38
#define GET_LSM303DLHC_MAG_GAIN_COMMAND               0x39
#define SET_LSM303DLHC_MAG_SAMPLING_RATE_COMMAND      0x3A
#define LSM303DLHC_MAG_SAMPLING_RATE_RESPONSE         0x3B
#define GET_LSM303DLHC_MAG_SAMPLING_RATE_COMMAND      0x3C
#define UNIQUE_SERIAL_RESPONSE                        0x3D
#define GET_UNIQUE_SERIAL_COMMAND                     0x3E
#define GET_DEVICE_VERSION_COMMAND                    0x3F
#define SET_LSM303DLHC_ACCEL_SAMPLING_RATE_COMMAND    0x40
#define LSM303DLHC_ACCEL_SAMPLING_RATE_RESPONSE       0x41
#define GET_LSM303DLHC_ACCEL_SAMPLING_RATE_COMMAND    0x42
#define SET_LSM303DLHC_ACCEL_LPMODE_COMMAND           0x43
#define LSM303DLHC_ACCEL_LPMODE_RESPONSE              0x44
#define GET_LSM303DLHC_ACCEL_LPMODE_COMMAND           0x45
#define SET_LSM303DLHC_ACCEL_HRMODE_COMMAND           0x46
#define LSM303DLHC_ACCEL_HRMODE_RESPONSE              0x47
#define GET_LSM303DLHC_ACCEL_HRMODE_COMMAND           0x48
#define SET_MPU9150_GYRO_RANGE_COMMAND                0x49
#define MPU9150_GYRO_RANGE_RESPONSE                   0x4A
#define GET_MPU9150_GYRO_RANGE_COMMAND                0x4B
#define SET_MPU9150_SAMPLING_RATE_COMMAND             0x4C
#define MPU9150_SAMPLING_RATE_RESPONSE                0x4D
#define GET_MPU9150_SAMPLING_RATE_COMMAND             0x4E
#define SET_MPU9150_ACCEL_RANGE_COMMAND               0x4F
#define MPU9150_ACCEL_RANGE_RESPONSE                  0x50
#define GET_MPU9150_ACCEL_RANGE_COMMAND               0x51
#define SET_BMP180_PRES_OVERSAMPLING_RATIO_COMMAND    0x52
#define BMP180_PRES_OVERSAMPLING_RATIO_RESPONSE       0x53
#define GET_BMP180_PRES_OVERSAMPLING_RATIO_COMMAND    0x54
#define BMP180_CALIBRATION_COEFFICIENTS_RESPONSE      0x58
```

```
#define GET_BMP180_CALIBRATION_COEFFICIENTS_COMMAND   0x59
#define RESET_TO_DEFAULT_CONFIGURATION_COMMAND        0x5A
#define RESET_CALIBRATION_VALUE_COMMAND               0x5B
#define MPU9150_MAG_SENS_ADJ_VALS_RESPONSE            0x5C
#define GET_MPU9150_MAG_SENS_ADJ_VALS_COMMAND         0x5D
#define SET_INTERNAL_EXP_POWER_ENABLE_COMMAND         0x5E
#define INTERNAL_EXP_POWER_ENABLE_RESPONSE            0x5F
#define GET_INTERNAL_EXP_POWER_ENABLE_COMMAND         0x60
#define SET_EXG_REGS_COMMAND                          0x61
#define EXG_REGS_RESPONSE                             0x62
#define GET_EXG_REGS_COMMAND                          0x63
#define SET_DAUGHTER_CARD_ID_COMMAND                  0x64
#define DAUGHTER_CARD_ID_RESPONSE                     0x65
#define GET_DAUGHTER_CARD_ID_COMMAND                  0x66
#define SET_DAUGHTER_CARD_MEM_COMMAND                 0x67
#define DAUGHTER_CARD_MEM_RESPONSE                    0x68
#define GET_DAUGHTER_CARD_MEM_COMMAND                 0x69
#define ACK_COMMAND_PROCESSED                         0xFF
#define ROUTINE_COMMUNICATION                                 0xE0   //'0'
#define START_SDBT_COMMAND                                    0x70
#define STATUS_RESPONSE                                       0x71
#define GET_STATUS_COMMAND                                    0x72
#define SET_TRIAL_CONFIG_COMMAND                              0x73
#define TRIAL_CONFIG_RESPONSE                                 0x74
#define GET_TRIAL_CONFIG_COMMAND                              0x75
#define SET_CENTER_COMMAND                                    0x76
#define CENTER_RESPONSE                                       0x77
#define GET_CENTER_COMMAND                                    0x78
#define SET_SHIMMERNAME_COMMAND                               0x79
#define SHIMMERNAME_RESPONSE                                  0x7a
#define GET_SHIMMERNAME_COMMAND                               0x7b
#define SET_EXPID_COMMAND                                     0x7c
#define EXPID_RESPONSE                                        0x7d
#define GET_EXPID_COMMAND                                     0x7e
#define SET_MYID_COMMAND                                      0x7F
#define MYID_RESPONSE                                         0x80
#define GET_MYID_COMMAND                                      0x81
#define SET_NSHIMMER_COMMAND                                  0x82
#define NSHIMMER_RESPONSE                                     0x83
#define GET_NSHIMMER_COMMAND                                  0x84
#define SET_CONFIGTIME_COMMAND                                0x85
#define CONFIGTIME_RESPONSE                                   0x86
#define GET_CONFIGTIME_COMMAND                                0x87
#define DIR_RESPONSE                                          0x88
#define GET_DIR_COMMAND                                       0x89
#define INSTREAM_CMD_RESPONSE                                 0x8A

//SENSORS0
#define SENSOR_A_ACCEL                          0x80
#define SENSOR_MPU9150_GYRO                     0x40
#define SENSOR_LSM303DLHC_MAG                   0x20
#define SENSOR_EXG1_24BIT                       0x10
#define SENSOR_EXG2_24BIT                       0x08
#define SENSOR_GSR                              0x04
#define SENSOR_EXT_A7                           0x02
#define SENSOR_EXT_A6                           0x01
//SENSORS1
//#define SDH_SENSOR_STRAIN                     0x80
//#define SDH_SENSOR_HR                         0x40
#define SENSOR_VBATT                            0x20
#define SENSOR_LSM303DLHC_ACCEL                 0x10
#define SENSOR_EXT_A15                          0x08
```

```
#define SENSOR_INT_A1                           0x04
#define SENSOR_INT_A12                          0x02
#define SENSOR_INT_A13                          0x01
//SENORS2
#define SENSOR_INT_A14                          0x80
#define SENSOR_MPU9150_ACCEL                    0x40
#define SENSOR_MPU9150_MAG                      0x20
#define SENSOR_EXG1_16BIT                       0x10
#define SENSOR_EXG2_16BIT                       0x08
#define SENSOR_BMP180_PRESSURE                  0x04
//#define SENSOR_BMP180_TEMPERATURE             0x02

#define MAX_COMMAND_ARG_SIZE     131    //maximum number of arguments for any command
                                        //sent
                                        //(daughter card mem write)
#define RESPONSE_PACKET_SIZE     131    //biggest possibly required  (daughter card mem
                                        //read + 1 byte for ack)
#define MAX_NUM_CHANNELS         28     //3xanalogAccel + 3xdigiGyro + 3xdigiMag +
                                        //3xLSM303DLHCAccel + 3xMPU9150Accel +
                                        //3xMPU9150MAG + BMP180TEMP + BMP180PRESS +
                                        //batteryVoltage + 3xexternalADC +
                                        //4xinternalADC
#define DATA_PACKET_SIZE         84     //3 + (MAX_NUM_CHANNELS * 2) + 1 + 6 (+1 as
                                        //BMP180 pressure requires 3 bytes, +6 for 4
                                        //(3 byte) ExG channels plus 2 status bytes
                                        //instead of 4xinternalADC)

// Channel contents
#define X_A_ACCEL                       0x00
#define Y_A_ACCEL                       0x01
#define Z_A_ACCEL                       0x02
#define VBATT                           0x03
#define X_LSM303DLHC_ACCEL              0x04
#define Y_LSM303DLHC_ACCEL              0x05
#define Z_LSM303DLHC_ACCEL              0x06
#define X_LSM303DLHC_MAG                0x07
#define Y_LSM303DLHC_MAG                0x08
#define Z_LSM303DLHC_MAG                0x09
#define X_MPU9150_GYRO                  0x0A
#define Y_MPU9150_GYRO                  0x0B
#define Z_MPU9150_GYRO                  0x0C
#define EXTERNAL_ADC_7                  0x0D
#define EXTERNAL_ADC_6                  0x0E
#define EXTERNAL_ADC_15                 0x0F
#define INTERNAL_ADC_1                  0x10
#define INTERNAL_ADC_12                 0x11
#define INTERNAL_ADC_13                 0x12
#define INTERNAL_ADC_14                 0x13
#define X_MPU9150_ACCEL                 0x14
#define Y_MPU9150_ACCEL                 0x15
#define Z_MPU9150_ACCEL                 0x16
#define X_MPU9150_MAG                   0x17
#define Y_MPU9150_MAG                   0x18
#define Z_MPU9150_MAG                   0x19
#define BMP180_TEMP                     0x1A
#define BMP180_PRESSURE                 0x1B
#define GSR_RAW                         0x1C
#define EXG_ADS1292R_1_STATUS           0x1D
#define EXG_ADS1292R_1_CH1_24BIT        0x1E
#define EXG_ADS1292R_1_CH2_24BIT        0x1F
#define EXG_ADS1292R_2_STATUS           0x20
#define EXG_ADS1292R_2_CH1_24BIT        0x21
```

```
#define EXG_ADS1292R_2_CH2_24BIT          0x22
#define EXG_ADS1292R_1_CH1_16BIT          0x23
#define EXG_ADS1292R_1_CH2_16BIT          0x24
#define EXG_ADS1292R_2_CH1_16BIT          0x25
#define EXG_ADS1292R_2_CH2_16BIT          0x26

// Infomem contents
//#define NV_NUM_CONFIG_BYTES             100
// Infomem contents
#define NV_NUM_SETTINGS_BYTES             33
#define NV_NUM_CALIBRATION_BYTES          84
#define NV_TOTAL_NUM_CONFIG_BYTES
NV_NUM_SETTINGS_BYTES+NV_NUM_CALIBRATION_BYTES

#define NV_SAMPLING_RATE                  0
#define NV_BUFFER_SIZE                    2
#define NV_SENSORS0                       3
#define NV_SENSORS1                       4
#define NV_SENSORS2                       5
#define NV_CONFIG_SETUP_BYTE0             6      //sensors setting bytes
#define NV_CONFIG_SETUP_BYTE1             7
#define NV_CONFIG_SETUP_BYTE2             8
#define NV_CONFIG_SETUP_BYTE3             9
#define NV_TRIAL_CONFIG0                  10
#define NV_TRIAL_CONFIG1                  11     //debug
#define NV_BT_INTERVAL                    12
#define NV_EXG_ADS1292R_1_CONFIG1         13     // exg bytes, not implemented yet
#define NV_EXG_ADS1292R_1_CONFIG2         14
#define NV_EXG_ADS1292R_1_LOFF            15
#define NV_EXG_ADS1292R_1_CH1SET          16
#define NV_EXG_ADS1292R_1_CH2SET          17
#define NV_EXG_ADS1292R_1_RLD_SENS        18
#define NV_EXG_ADS1292R_1_LOFF_SENS       19
#define NV_EXG_ADS1292R_1_LOFF_STAT       20
#define NV_EXG_ADS1292R_1_RESP1           21
#define NV_EXG_ADS1292R_1_RESP2           22
#define NV_EXG_ADS1292R_2_CONFIG1         23
#define NV_EXG_ADS1292R_2_CONFIG2         24
#define NV_EXG_ADS1292R_2_LOFF            25
#define NV_EXG_ADS1292R_2_CH1SET          26
#define NV_EXG_ADS1292R_2_CH2SET          27
#define NV_EXG_ADS1292R_2_RLD_SENS        28
#define NV_EXG_ADS1292R_2_LOFF_SENS       29
#define NV_EXG_ADS1292R_2_LOFF_STAT       30
#define NV_EXG_ADS1292R_2_RESP1           31
#define NV_EXG_ADS1292R_2_RESP2           32
#define NV_A_ACCEL_CALIBRATION            33
#define NV_MPU9150_GYRO_CALIBRATION       54
#define NV_LSM303DLHC_MAG_CALIBRATION     75
#define NV_LSM303DLHC_ACCEL_CALIBRATION   96

//Config byte masks
//Config Byte0
#define LSM303DLHC_ACCEL_SAMPLING_RATE                  0xF0
#define LSM303DLHC_ACCEL_RANGE                          0x0C
#define LSM303DLHC_ACCEL_LOW_POWER_MODE                 0x02
#define LSM303DLHC_ACCEL_HIGH_RESOLUTION_MODE   0x01
//Config Byte1
#define MPU9150_SAMPLING_RATE                           0xFF
//Config Byte2
#define LSM303DLHC_MAG_GAIN                             0xE0
#define LSM303DLHC_MAG_SAMPLING_RATE                    0x1C
```

```
#define MPU9150_GYRO_RANGE                              0x03
//Config Byte3
#define MPU9150_ACCEL_RANGE                             0xC0
#define BMP180_PRESSURE_RESOLUTION                      0x30
#define GSR_RANGE                                       0x0E
#define EXP_POWER_ENABLE                                0x01
//Unused bits 3-0

//ADC initialisation mask
#define MASK_A_ACCEL        0x0001
#define MASK_VBATT          0x0002
#define MASK_EXT_A7         0x0004
#define MASK_EXT_A6         0x0008
#define MASK_EXT_A15        0x0010
#define MASK_INT_A1         0x0020
#define MASK_INT_A12        0x0040
#define MASK_INT_A13        0x0080
#define MASK_INT_A14        0x0100
#define MASK_MSP_TEMP       0x0200

//LSM303DLHC Accel Range
//Corresponds to the FS field of the LSM303DLHC's CTRL_REG4_A register
//and the AFS_SEL field of the MPU9150's ACCEL_CONFIG register
#define ACCEL_2G            0x00
#define ACCEL_4G            0x01
#define ACCEL_8G            0x02
#define ACCEL_16G           0x03

//LSM303DLHC Accel Sampling Rate
//Corresponds to the ODR field of the LSM303DLHC's CTRL_REG1_A register
#define LSM303DLHC_ACCEL_POWER_DOWN       0x00
#define LSM303DLHC_ACCEL_1HZ              0x01
#define LSM303DLHC_ACCEL_10HZ             0x02
#define LSM303DLHC_ACCEL_25HZ             0x03
#define LSM303DLHC_ACCEL_50HZ             0x04
#define LSM303DLHC_ACCEL_100HZ            0x05
#define LSM303DLHC_ACCEL_200HZ            0x06
#define LSM303DLHC_ACCEL_400HZ            0x07
#define LSM303DLHC_ACCEL_1_620KHZ         0x08   //1.620kHz in Low-power mode only
#define LSM303DLHC_ACCEL_1_344kHz         0x09   //1.344kHz in normal mode, 5.376kHz in
                                                 //low-power mode

//LSM303DLHC Mag gain
#define LSM303DLHC_MAG_1_3G               0x01 //+/-1.3 Gauss
#define LSM303DLHC_MAG_1_9G               0x02 //+/-1.9 Gauss
#define LSM303DLHC_MAG_2_5G               0x03 //+/-2.5 Gauss
#define LSM303DLHC_MAG_4_0G               0x04 //+/-4.0 Gauss
#define LSM303DLHC_MAG_4_7G               0x05 //+/-4.7 Gauss
#define LSM303DLHC_MAG_5_6G               0x06 //+/-5.6 Gauss
#define LSM303DLHC_MAG_8_1G               0x07 //+/-8.1 Gauss

//LSM303DLHC Mag sampling rate
#define LSM303DLHC_MAG_0_75HZ             0x00 //0.75 Hz
#define LSM303DLHC_MAG_1_5HZ              0x01 //1.5 Hz
#define LSM303DLHC_MAG_3HZ                0x02 //3.0 Hz
#define LSM303DLHC_MAG_7_5HZ              0x03 //7.5 Hz
#define LSM303DLHC_MAG_15HZ               0x04 //15 Hz
#define LSM303DLHC_MAG_30HZ               0x05 //30 Hz
#define LSM303DLHC_MAG_75HZ               0x06 //75 Hz
#define LSM303DLHC_MAG_220HZ              0x07 //220 Hz

//calibration info
```

```
#define S_ACCEL                         0
#define S_GYRO                          1
#define S_MAG                           2
#define S_ACCEL_A                       3
//#define S_ECG                         3
//#define S_EMG                         4

//MPU9150 Gyro range
#define MPU9150_GYRO_250DPS             0x00 //+/-250 dps
#define MPU9150_GYRO_500DPS             0x01 //+/-500 dps
#define MPU9150_GYRO_1000DPS            0x02 //+/-1000 dps
#define MPU9150_GYRO_2000DPS            0x03 //+/-2000 dps

//#digital accel_range
#define RANGE_2G                        0
#define RANGE_4G                        1
#define RANGE_8G                        2
#define RANGE_16G                       3

//#mag_gain
#define LSM303_MAG_13GA                 1
#define LSM303_MAG_19GA                 2
#define LSM303_MAG_25GA                 3
#define LSM303_MAG_40GA                 4
#define LSM303_MAG_47GA                 5
#define LSM303_MAG_56GA                 6
#define LSM303_MAG_81GA                 7

//SD Log file header format
#define SDHEAD_LEN                      256// 0-255

#define SDH_SAMPLE_RATE_0                       0
#define SDH_SAMPLE_RATE_1                       1
#define SDH_SENSORS0                            3
#define SDH_SENSORS1                            4
#define SDH_SENSORS2                            5
#define SDH_CONFIG_SETUP_BYTE0                  8 //sensors setting bytes
#define SDH_CONFIG_SETUP_BYTE1                  9
#define SDH_CONFIG_SETUP_BYTE2                  10
#define SDH_CONFIG_SETUP_BYTE3                  11
#define SDH_TRIAL_CONFIG0                       16
#define SDH_TRIAL_CONFIG1                       17
#define SDH_BROADCAST_INTERVAL                  18
//#define SDH_ACCEL_RANGE                       22      //digital accel range
//#define SDH_GSR_RANGE                         23
//#define SDH_MAG_RANGE                         24
//#define SDH_MAG_RATE                          25
//#define SDH_ACCEL_RATE                        26      digital accel rate
//#define SDH_GYRO_RATE                         27
//#define SDH_GYRO_RANGE                        28
//#define SDH_PRESSURE_PREC                     29
#define SDH_SHIMMERVERSION_BYTE_0               30
#define SDH_SHIMMERVERSION_BYTE_1               31
#define SDH_MYTRIAL_ID                          32
#define SDH_NSHIMMER                            33
#define SDH_FW_VERSION_TYPE_0                   34
#define SDH_FW_VERSION_TYPE_1                   35
#define SDH_FW_VERSION_MAJOR_0                  36
#define SDH_FW_VERSION_MAJOR_1                  37
#define SDH_FW_VERSION_MINOR                    38
#define SDH_FW_VERSION_INTERNAL                 39
#define SDH_CONFIG_TIME_0                       52
```

```
#define SDH_CONFIG_TIME_1                       53
#define SDH_CONFIG_TIME_2                       54
#define SDH_CONFIG_TIME_3                       55
#define SDH_EXG_ADS1292R_1_CONFIG1              56
#define SDH_EXG_ADS1292R_1_CONFIG2              57
#define SDH_EXG_ADS1292R_1_LOFF                 58
#define SDH_EXG_ADS1292R_1_CH1SET               59
#define SDH_EXG_ADS1292R_1_CH2SET               60
#define SDH_EXG_ADS1292R_1_RLD_SENS             61
#define SDH_EXG_ADS1292R_1_LOFF_SENS            62
#define SDH_EXG_ADS1292R_1_LOFF_STAT            63
#define SDH_EXG_ADS1292R_1_RESP1                64
#define SDH_EXG_ADS1292R_1_RESP2                65
#define SDH_EXG_ADS1292R_2_CONFIG1              66
#define SDH_EXG_ADS1292R_2_CONFIG2              67
#define SDH_EXG_ADS1292R_2_LOFF                 68
#define SDH_EXG_ADS1292R_2_CH1SET               69
#define SDH_EXG_ADS1292R_2_CH2SET               70
#define SDH_EXG_ADS1292R_2_RLD_SENS             71
#define SDH_EXG_ADS1292R_2_LOFF_SENS            72
#define SDH_EXG_ADS1292R_2_LOFF_STAT            73
#define SDH_EXG_ADS1292R_2_RESP1                74
#define SDH_EXG_ADS1292R_2_RESP2                75
#define SDH_LSM303DLHC_ACCEL_CALIBRATION        76
#define SDH_MPU9150_GYRO_CALIBRATION            97
#define SDH_LSM303DLHC_MAG_CALIBRATION          118
#define SDH_A_ACCEL_CALIBRATION                 139
#define SDH_TEMP_PRES_CALIBRATION               160
#define SDH_MY_LOCALTIME                        252     //252-255

//SENSORS0
#define       SDH_SENSOR_A_ACCEL                0x80
#define SDH_SENSOR_MPU9150_GYRO                 0x40
#define SDH_SENSOR_LSM303DLHC_MAG               0x20
#define SDH_SENSOR_EXG1_24BIT                   0x10
#define SDH_SENSOR_EXG2_24BIT                   0x08
#define SDH_SENSOR_GSR                          0x04
#define SDH_SENSOR_EXTCH7                       0x02
#define SDH_SENSOR_EXTCH6                       0x01
//SENSORS1
//#define SDH_SENSOR_STRAIN                     0x80
//#define SDH_SENSOR_HR                         0x40
#define SDH_SENSOR_VBATT                        0x20
#define SDH_SENSOR_LSM303DLHC_ACCEL             0x10
#define SDH_SENSOR_EXTCH15                      0x08
#define SDH_SENSOR_INTCH1                       0x04
#define SDH_SENSOR_INTCH12                      0x02
#define SDH_SENSOR_INTCH13                      0x01
//SENSORS2
#define SDH_SENSOR_INTCH14                      0x80
#define SDH_SENSOR_MPU9150_ACCEL                0x40
#define SDH_SENSOR_MPU9150_MAG                  0x20
#define SDH_SENSOR_EXG1_16BIT                   0x10
#define SDH_SENSOR_EXG2_16BIT                   0x08
#define SDH_SENSOR_BMP180_PRES                  0x04
//#define SDH_SENSOR_BMP180_TEMP                0x02
//SENSORS3
#define SDH_SENSOR_MSP430_TEMP                  0x01
#define SDH_SENSOR_TCXO                         0x80

//SDH_TRIAL_CONFIG0
#define SDH_IAMMASTER                           0x02
```

```
#define SDH_TIME_SYNC                           0x04
#define SDH_TIME_STAMP                          0x08// not used now, reserved as 1
#define SDH_GYRO_BUTTON_ENABLE                  0x10
#define SDH_USER_BUTTON_ENABLE                  0x20
#define SDH_SET_PMUX                            0x40// not used now, reserved as 0
#define SDH_SET_5V_REG                          0x80// not used now
//SDH_TRIAL_CONFIG1
#define SDH_SINGLETOUCH                         0x80
#define SDH_ACCEL_LPM                           0x40//config has this bit
#define SDH_ACCEL_HRM                           0x20//config has this bit
#define SDH_TCXO                                0x10
#define SDH_EXP_POWER                           0x08//config has this bit
#define SDH_MONITOR                             0x04

//choice of clock
#define TCXO_CLOCK          255765.625
#define MSP430_CLOCK 32768

// BT routine communication
// all node time must *2 in use
// all center time must *4 in use
#define RC_AHD              3
#define RC_WINDOW_N         13
#define RC_WINDOW_C         27
#define RC_INT_N            27
#define RC_INT_C            54          //240
#define RC_CLK_N            16384       //16384=2hz;//32768=1hz;8192=4hz
#define RC_CLK_C            8192        //16384=2hz;//32768=1hz;8192=4hz
#define RC_FACTOR_N 32768/RC_CLK_N      //16384=2hz;//32768=1hz;8192=4hz
#define RC_FACTOR_C 32768/RC_CLK_C      //16384=2hz;//32768=1hz;8192=4hz

//routine communication response text - ack:flag:time4:time3:time2:time1
#define RCT_SIZE            6
#define RCT_ACK             0
#define RCT_FLG             1
#define RCT_TIME            2
#define RCT_TIME_SIZE       4

// sd card write buffer size
#define SDBUFF_SIZE_MAX     4096    //4095  255
#define SDBUFF_SIZE         512     //4095  512

// BATTERY
#define BATT_HIGH           0
#define BATT_MID            1
#define BATT_LOW            2
#define BATT_ITNERVAL       19660800

#define MAX_CHARS           13

//BMP Pressure oversampling ratio
#define BMP180_OSS_1                0x00
#define BMP180_OSS_2                0x01
#define BMP180_OSS_4                0x02
#define BMP180_OSS_8                0x03

//BtStream specific extension to range values : should SDLog keep it?
#define GSR_AUTORANGE               0x04

#endif
```

6.4. Troubleshoot

Green and Blue LEDs flash when I undock or reboot my Shimmer.

The green and blue LEDs flashing on the *Shimmer3* indicate that the firmware either cannot create the required directories or cannot write to the required files on the SD card. Try the following steps to rectify the problem:

1. Check that the SD card is correctly inserted.
2. Ensure that the SD card has a capacity of 2 GB or less and is compatible with *Shimmer3* (refer to the *Shimmer User Manual*, available for download from www.shimmersensing.com).
3. Ensure that the SD card memory is not full.
4. Ensure that the *experimentid* and *shimmername* parameters, specified in the *sdlog.cfg* file contain only alphanumeric characters (a,..., z, A,..., Z, 0,..., 9), dash ('-') and underscore ('_').

When I undock or power on my Shimmer, the green LED continuously flashes at a rate of 5Hz

Try power-cycling the *Shimmer3*. The problem may be due to an error in bluetooth initialisation.
If the problem persists after power-cycling, try changing the SD card for a newer one - if the SD card is corrupt, the firmware may fail to correctly read the configuration file.

The data file is empty after logging

Ensure that you log for a minimum of one minute in order for data to be written to the SD card.

The configuration does not match the parameters in the sdlog.cfg file.

The *sdlog.cfg* file is read once each time the *Shimmer3* is rebooted or undocked so the configuration will always match the most recent configuration file at the time of logging. If the parameters in the configuration header of your data files do not match those in the *sdlog.cfg* file, it is likely that you have changed the *sdlog.cfg* file contents since logging the data in question.

The calibration parameters in the configuration header do not match the Calibration/calibParams.ini file.

Calibration parameters will only be loaded for sensors that are enabled; calibration parameters for disabled sensors will all have zero value.

If a sensor is enabled and the calibration parameters do not match the calibration file, try the following steps to rectify the problem:

1. Ensure that the calibration file is stored in the correct file location from the SD card top level directory, according to the following (case-sensitive) options:

    - /Calibration/calibParams.ini
    - /calibration/calibParams.ini

    No other file path will be recognised by the firmware.

2. Ensure that you have implemented the correct byte-order and endianness when you read the calibration parameters from the configuration header, according to the information in Section 5.2.

Shimmer
The Realtime Building
Clonshaugh Technology Park
Dublin 17
Ireland
T: +353 (1)848 6112
E: info@ShimmerSensing.com

Shimmer North America
101 Tremont St, Suite 500
Boston
MA, 02108
USA
T: +1 857 362 7254
E: info@ShimmerSensing.com

www.ShimmerSensing.com
/ShimmerResearch
@ShimmerSensing
/company/Shimmer
/ShimmerResearch
/ShimmerResearch