SDLog for Shimmer3 Firmware User Manual

Rev 0.6a

Legal Notices and Disclaimer

Table of Contents

1. Introduction

This document is an accompaniment to the *SDLog Firmware v0.6.0* image and later for *Shimmer3*. No previous development experience is required.

Note: There was significant change to the *SDLog* configuration header and the data format in *SDLog Firmware v0.6.0*. Users of *SDLog Firmware v0.5.0* should refer to the Appendices section of this document for parsing information. Users of *SDLog Firmware v0.5.0* are advised to upgrade to *SDLog Firmware v0.6.0 to ensure support for future Shimmer3 offerings.* Versions of *SDLog Firmware* prior to *v0.5.0* are unsupported by this document and are unsupported in applications provided by Shimmer.

Note: *SDLog Firmware v0.6.0* is a Beta release.

SDLog Firmware is a firmware image which allows logging of data from a Shimmer to the on-board SD card. The firmware allows full user configuration of the Shimmer via a configuration file, stored on the SD card. Many useful features, such as time synchronisation among multiple devices, start/stop logging on one or more devices by a single button press and user-defined naming of devices are enabled by this firmware image.

It is recommended that *SDLog Firmware* be used in conjunction with the *ShimmerLog* software, which is available for download from the Shimmer website (www.shimmersensing.com).

2. Scope of this User Manual

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

3. Pre-Requisites

SDLog Firmware can be used with a *Shimmer2*, *Shimmer2r* or *Shimmer3* device with a microSD card, with capacity up to 2GB. For Shimmer compatibility the microSD card chosen must support 1-bit SPI mode. For *Shimmer2* or *Shimmer2r*, please refer to the *SDLog for Shimmer2/2r Firmware User Manual* which is available for download at www.shimmersensing.com.

A Shimmer dock is required to allow access the SD card on the Shimmer from the PC for configuration of logging preferences and data transfer. Please note that legacy (black) Shimmer docks are not suitable for this purpose and a newer (white) dock is required.

4. Installation

Install the *SDLog Firmware* v0.6.0 firmware image (SDLog_v0.6.0_shimmer3.txt) onto a *Shimmer3* device, using the *Shimmer3 Bootstrap Loader* (*Shimmer3 BSL*) program. The Shimmer3BSL.exe program is available via download from the Members area at www.shimmersensing.com.

5. Using the firmware

To use this firmware image, the user must provide the desired configuration parameters of each Shimmer. The configuration parameters are saved in a configuration file, named *sdlog.cfg*, from which they will be loaded by the Shimmer at initialisation. 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 extra whitespaces will cause parsing errors.

If the configuration file is not found by the firmware or if the file has a glitch, such that the firmware cannot read it, a blue/green LED indicator on the Shimmer will intermittently flash to warn the user of the error. Note that this error will not occur in the event of parsing errors (e.g. typographical errors, extra whitespace, repeated commands); it is the user's responsibility to ensure that the *sdlog.cfg* file is well-written.

It is recommended to use the *ShimmerLog (*single *Shimmer) or Multi Shimmer Sync for SD*, 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.

5.1. 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 and their corresponding *sdlog.cfg* entries are listed in Table 1 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` and `exp_power=1`	`intch13=0` and `exp_power=0`
ECG	`(exg1_24bit=1 and exg2_24bit=1 and exp_power=1) or (exg1_16bit=1 and exg2_16bit=1 and exp_power=1)`	`(exg1_24bit=0 and exg2_24bit=0 and exp_Power=0) or (exg1_16bit=0 and exg2_16bit=0 and exp_Power=0)`
EMG	`exg1_24bit=1 or exg1_16bit=1`	`exg1_24bit=0 or exg1_16bit=0`

Table 1 - Sensor enable/disable commands for Shimmer3

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.

There are currently no sensor conflicts for *Shimmer3*.

5.2. Configuring the sensors

The following options can be used to set sensor configuration parameters, using the commands in Table 2.

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) 3 (Auto Range)	4 (Auto Range)
Pressure Resolution	`pres_bmp180_prec=0`	0 (Low) 1 (Standard) 2 (High) 3 (Very High)	0 (Low)

Table 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.

5.3. Configuring the experiment

Logging preferences can be configured using the commands in Table 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 - Logging configuration options

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 (*True Fs*) can be calculated from the specified sampling rate (*Fs*) using the following formula:

$$True\ Fs = 32768/(\mathrm{ceil}(32768/Fs)),$$

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

The *Synchronisation* setting allows samples from multiple Shimmers to be synchronised in time. If this setting is enabled, exactly one Shimmer must be designated as *Master* and the others as *Slaves* (see *Master* setting below). It is vital that the *Configuration ID* parameter (described below) is equal for all Shimmers in the experiment if synchronisation is enabled. For more information on this feature, see Section 6.1.

If *Master* is enabled, the device will assume the role of master for synchronisation of timestamps and/or for the single-touch start feature (described below). Otherwise, the device will assume the role of slave.

> **Note:** it is the user's responsibility to ensure that there is exactly one master and all other devices are slaves. If multiple Shimmers are acting as master, behaviour of the slaves will not be as expected.

> **Note**: it is the user's responsibility to ensure that all slaves have the MAC address of the master's Bluetooth module stored as the value for the *Master MAC Address* parameter in their *sdlog.cfg* files.

The *Broadcast interval* determines how often the master Shimmer sends a timestamp update broadcast. Its value is the integer number of seconds that the master should wait in between each update. The slave devices use this value to determine when to turn on the radio to listen for a timestamp update.

> **Note:** the minimum allowed value for the broadcast interval is 54 seconds. This is due to power constraints and time constraints of the Bluetooth radio. If the user provides a value less than 54 seconds in the configuration file, the SDLog firmware forces the broadcast interval to be 54 seconds.

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.

Single-touch start allows logging to be started on multiple Shimmers, in response to a button press on a single Shimmer. If this setting is enabled, exactly one Shimmer must be designated as the *Master* and all others are slaves and the user button must be enabled. It is vital that the *Configuration ID* parameter (described below) is equal for all Shimmers in the experiment if single-touch start is enabled. For more details describing this feature, see Section 6.2.

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.

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 7.

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 7.

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.

5.4. Example sdlog.cfg file for *Shimmer3*

```
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
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
```

5.5. Providing calibration parameters

Calibration parameters are not used in this firmware image to provide calibrated data, but they are saved in the configuration header along with the data, so that the data can later be calibrated. 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*, which is available for download from the members' area of www.shimmersensing.com.

If a calibration file is found by the firmware, then the appropriate parameters from the file will be stored in the configuration header for the enabled sensors. If no calibration file is found, then 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.

6. Firmware features

6.1. Time synchronisation

This option requires exactly one Shimmer to be designated as the "master" Shimmer and any other Shimmers to be designated as "slaves". Enabling time synchronisation allows samples from multiple Shimmers to be converted to a common reference clock (the clock of the master Shimmer).

It is important to note that, in this synchronisation method, no effort is made to alter the clock frequency of any Shimmer in the firmware. Instead, each clock runs at its own frequency throughout logging and the alignment of samples to a common reference clock is done in post-processing. The basis of the alignment is a log of offsets between each Shimmer's local timestamp and master timestamps. On the *Shimmer3*, the slaves each connect with the master via Bluetooth (BT) at regular intervals and the master sends its timestamp at that instant to the connected slave. The slaves must be within radio range (approximately 10m) of the master for successful communication.

One important synchronisation parameter is determined by the user in the *sdlog.cfg* file: the broadcast interval (denoted BI in the following). If the user does not provide a value for BI, it defaults to 120 seconds.

<u>Master:</u>

- When the master Shimmer is powered on or reset, its on-board 32 kHz clock will begin counting at zero. This clock will be the common reference clock for all samples in the experiment.
- When the master starts logging, it will save its initial timestamp with a resolution of 32 bits (see Section 6.3 for a description on where this initial timestamp is saved).
- On *Shimmer3*:
  - Throughout logging, the master will periodically turn on its BT radio at regular intervals of BI seconds. When a connection is successfully made with the Bluetooth from a slave device, the master will send its timestamp with a resolution of 32 bits.
  - The master turns on its radio for a pre-determined amount of time and waits for connections from slave devices. After the pre-determined length of time has elapsed, the master turns off its radio.

<u>Slave:</u>

- When a slave Shimmer is powered on or reset, its on-board 32 kHz clock will begin counting at zero.
- When the slave starts logging, it will save its initial timestamp with a resolution of 32 bits (see Section 6.3 for a description on where this initial timestamp is saved).
- On *Shimmer3*:

  - The slave will periodically turn on its BT module and connect with the master device, according to the MAC address provided in the *sdlog.cfg* file.
  - Upon successful connection, the slave will receive a message from the master containing the master's timestamp.
  - The slave Shimmer will turn off its radio immediately after receiving a master timestamp message or if the connection attempt is unsuccessful.
  - If the previous connection attempt was unsuccessful, the slave will repeat the above steps at a short interval; otherwise, the slave will repeat the communication steps every BI seconds.
  - This sequence repeats throughout logging.
- Whenever the slave receives a timestamp, it immediately records the time on its local 32 kHz clock.
- The difference between the slave's local time and the received master time (local time - master time) is calculated, as follows:
  - The *offset sign* (one byte) represents the sign of the difference (0 (positive) if local time >= master time and 1 (negative) otherwise).
  - The magnitude of the difference (|local time - master time|), with a resolution of 32 bits, is represented by a 4-byte unsigned integer.
- The 5 bytes of the offset (sign and magnitude) are written to the SD card at the start of every write operation. There will be many write operations for each time offset update; for example, a write operation might occur many times every second, whilst the default value for the broadcast interval is 120 seconds. At any given write operation, if a new master timestamp has not been received since the previous write operation, then the value 0xFFFFFFFF is written for the offset magnitude, with a sign byte of 0. This value must be recognised in software to denote an "invalid" value and be removed before processing the offsets.

6.2. Start/Stop logging

Dock

This is the default way for the Shimmer to start and stop logging data to the SD card and is enabled if none of the other options below are chosen. 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 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 recommendable 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)

This option relies on the orange user button on the *Shimmer3* enclosure to start/stop logging.

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 or the battery runs out, whichever happens soonest.

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

Single-touch start/stop

This option requires exactly one Shimmer to be designated as the "master" Shimmer and any other Shimmers to be designated as "slaves". This option allows logging to be started on all Shimmers by a single press of the user button on the master Shimmer. Similarly, logging can be stopped on all Shimmers by a single press of the user button on the master Shimmer. Steps 1 and 2 from above (standby mode for three seconds followed by configuration) remain unchanged for this setup.

There are a number of important points to remember with this setting, as outlined below:

<u>**Master:**</u>

- When the master Shimmer is powered on or reset, it will do nothing while it waits for a user button press.
- If the master user button is pressed while the master Shimmer is on the dock, it will have no effect on logging.

- If the master user button is pressed while the master Shimmer is undocked and not logging, the master will immediately start logging. Furthermore, it will include a "start-logging" message in all further Bluetooth communications with slaves, telling them to start logging.

- If the master user button is pressed while the master Shimmer is undocked and logging, the master will immediately stop logging and it will include a "stop-logging" message in all further Bluetooth communications with slaves, telling them to stop logging.

Slave:

- The user button is always enabled on the slave Shimmers in single-touch start configuration. This feature allows the user to manually start logging on the slave Shimmers, either to override the single-touch start option or as a backup, in case the "start-logging" or "stop-logging" messages from the master are missed by the slave.

- When the slave Shimmer is powered on or reset and is undocked, it will periodically connect to the master Shimmer via Bluetooth while it waits for a "start-logging" message or a user button press.

- If a "start-logging" message is received by the slave, while it is not logging, it will immediately start logging.

- If a "start-logging" message is received by the slave, while it is already logging, the message will be ignored.

- While the slave Shimmer is logging, Bluetooth communication with the master will be periodically attempted.

- If a "stop-logging" message is received by the slave, while it is logging, then the slave will immediately stop logging and will periodically connect to the master Shimmer via Bluetooth to wait for the next "start-logging" message.

- If a "stop-logging" message is received by the slave, while it is not logging, then the slave will ignore the message.

- The user button can be used as backup/override as follows:

    - If the slave user button is pressed while the slave Shimmer is undocked and not logging, the slave will immediately start logging.

    - If the slave user button is pressed while the Shimmer is undocked and logging, the slave will immediately stop logging.

- Finally, if the slave Shimmer is docked, logging will stop (if the Shimmer was logging) and the radio will be turned off.

6.3. LED indicators

The Shimmer has three LEDs (green, yellow[1] and red), which are used to indicate operation and battery charge status while the device is undocked, according to Table 4.

			Operation					Battery Status		
	LED	Power OFF	Standby	Logging	Configuring	Radio ON	Error*	Low Charge	Medium Charge	Full Charge
SDLog	Green (a)	OFF	-	-	-	-	-	OFF	OFF	0.1 s ON/5 s OFF
	Yellow	OFF	-	-	-	-	-	OFF	0.1 s ON/5 s OFF	OFF
	Red	OFF	-	-	-	-	-	0.1 s ON/5 s OFF	OFF	OFF
	Green (b)	OFF	0.1 s ON/2 s OFF	1 s ON/1s OFF	0.1 s ON/0.1 s OFF	-	0.1 s ON/0.1 s OFF	-	-	-
	Blue	OFF	OFF	-	-	ON	0.1 s ON/0.1 s OFF	-	-	-

Table 4 - SD Log firmware LED indicators (undocked)

** 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 recommendable to place the Shimmer 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 Shimmer unit before docking.

The LEDs also indicate operation and battery charge status while the device is docked, according to Table 5.

			Operation					Battery Status		
	LED	Power OFF	Standby	Logging	Configuring	Radio ON	Error*	Low Charge	Medium Charge	Full Charge
SDLog	Green (a)	OFF	-	-	-	-	-	OFF	OFF	ON
	Yellow	OFF	-	-	-	-	-	OFF	ON	OFF
	Red	OFF	-	-	-	-	-	ON	OFF	OFF
	Green (b)	OFF	0.1 s ON/2 s OFF	-	-	-	0.1 s ON/0.1 s OFF	-	-	-
	Blue	OFF	OFF	-	-	ON	0.1 s ON/0.1 s OFF	-	-	-

Table 5 - SD Log firmware LED indicators (docked)

Logging and other SD card related operations are not carried out while the device is on the dock.

7. Reading the data

It is recommended that the *ShimmerLog (single Shimmer) or MSS for SD (multiple Shimmers)* 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.

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

7.1. 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 Shimmer id number (e.g. *IDbf5e*)), followed by a three digit number which is sequentially incremented at new logging session.
- For example, the first time Shimmer 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.

7.2. Parsing the data

Configuration header

Each data file (e.g. *data/experiment1/device1-000/000*), begins with a header with the 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 #	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	ADC Sample Rate (LSB)							
1	ADC Sample Rate (MSB)							
2	reserved							
3	Acc (LN)*	Gyr	Mag	EXG1_24BIT	EXG2_24BIT	GSR	ExtCh 7	ExtCh 6
4			Battery	Acc (WR)*	ExtCh 15	IntCh 1	IntCh 12	IntCh 13
5	IntCh 14	MPU9150_ACCEL	MPU9150_MAG	EXG1_16BIT	EXG2_16BIT	pres		
6								
7								
8	LSM303 Digital 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 #	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
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	tcxo				
18	Broadcast Interval							
19								

Byte # 30 – 39: Firmware and Shimmer Parameters

Byte #	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
30	ShimmerVersion (MSB)							
31	ShimmerVersion (LSB)							
32	MyTrialId							
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. *Shimmer1* is denoted by 0, *Shimmer2* by 1, *Shimmer2r* by 2 and *Shimmer3* 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 *SDLog v1.2*, *Major* = 1 and *Minor* = 1). 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 #	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
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 5.3.

Byte # 56-76: ExG Calibration

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

Byte #	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
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 #	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
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

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

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

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

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

Byte # 182 - 251: Reserved for future use

Initial Timestamp

Immediately after the configuration header, there is a 4-byte timestamp, whose value is the time on the local Shimmer 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 Shimmers to a common clock.

252	initial time stamp				
253	initial time stamp				
254	initial time stamp				
255	initial time stamp				

Data log

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).

Note: The block size in SDLog v0.3.0 (and earlier) is 1024 bytes as oppose to 512 bytes so any formula referencing 512 bytes should be replaced with 1024 if SDLog v.0.3.0 or earlier was used.

The exact number of bytes per block and the interpretation of those bytes of data depend on the enabled sensors and whether or not synchronisation is enabled. Introducing the variable, *S*, such that S = 1 if master-slave synchronisation is enabled and 0 otherwise, the total number of bytes written per block, Bp, is given by:

$$Bp = N * (Bs + 2) + 5*S$$

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 -5*S)/(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 Shimmer's local 32 kHz clock.

There are two possible cases, as outlined in the examples below.

Case 1: Synchronisation is enabled.

The first byte represents the sign (0 if positive, 1 if negative) of the time offset between the local slave timestamp and the broadcast master timestamp. The next four bytes represent the 32-bit magnitude of the offset, such that applying the sign byte results in offset = local timestamp – master timestamp. The remainder 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 are as follows:

Byte 0:	offset sign	
Byte 1:	offset magnitude byte 0 (MSB)	
Byte 2:	offset magnitude byte 1	
Byte 3:	offset magnitude byte 2	
Byte 4:	offset magnitude byte 3 (LSB)	
Byte 5:	timestamp 0 byte 0 (MSB)	
Byte 6:	timestamp 0 byte 1 (LSB)	
Byte 7:	sample 0 channel 0 byte 0 (MSB)	(XAccel)
Byte 8:	sample 0 channel 0 byte 1 (LSB)	(XAccel)
Byte 9:	sample 0 channel 1 byte 0 (MSB)	(YAccel)
Byte 10:	sample 0 channel 1 byte 1 (LSB)	(YAccel)
Byte 11:	sample 0 channel 2 byte 0 (MSB)	(ZAccel)

Byte 12:	sample 0 channel 2 byte 1 (LSB)	(ZAccel)
Byte 13:	sample 0 channel 3 byte 0 (MSB)	(XGyro)
Byte 14:	sample 0 channel 3 byte 1 (LSB)	(XGyro)
Byte 15:	sample 0 channel 4 byte 0 (MSB)	(YGyro)
Byte 16:	sample 0 channel 4 byte 1 (LSB)	(YGyro)
Byte 17:	sample 0 channel 5 byte 0 (MSB)	(ZGyro)
Byte 18:	sample 0 channel 5 byte 1 (LSB)	(ZGyro)
Byte 19:	timestamp 1 byte 0 (MSB)	
Byte 20:	timestamp 1 byte 1 (LSB)	
Byte 21:	sample 1 channel 0 byte 0 (MSB)	(XAccel)
Byte 22:	sample 1 channel 0 byte 1 (LSB)	(XAccel)

and so on, until a total of *Bp* = 509 bytes is reached (*Bs* = 3*2 + 3*2 = 12; *N* = floor((512 -6)/(12 + 2)) = 36).

Case 2: Synchronisation is not enabled.

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 are as follows:

Byte 0:	timestamp 0 byte 0 (MSB)	
Byte 1:	timestamp 0 byte 1 (LSB)	
Byte 2:	sample 0 channel 0 byte 0 (MSB)	(XAccel)
Byte 3:	sample 0 channel 0 byte 1 (LSB)	(XAccel)
Byte 4:	sample 0 channel 1 byte 0 (MSB)	(YAccel)
Byte 5:	sample 0 channel 1 byte 1 (LSB)	(YAccel)
Byte 6:	sample 0 channel 2 byte 0 (MSB)	(ZAccel)
Byte 7:	sample 0 channel 2 byte 1 (LSB)	(ZAccel)
Byte 8:	sample 0 channel 3 byte 0 (MSB)	(XGyro)
Byte 9:	sample 0 channel 3 byte 1 (LSB)	(XGyro)
Byte 10:	sample 0 channel 4 byte 0 (MSB)	(YGyro)

Byte 11: sample 0 channel 4 byte 1 (LSB) (YGyro)

Byte 12: sample 0 channel 5 byte 0 (MSB) (ZGyro)

Byte 13: sample 0 channel 5 byte 1 (LSB) (ZGyro)

Byte 14: timestamp 1 byte 0 (MSB)

Byte 15: timestamp 1 byte 1 (LSB)

Byte 16: sample 1 channel 0 byte 0 (MSB) (XAccel)

Byte 17: sample 1 channel 0 byte 1 (LSB) (XAccel)

and so on, until a total of Bp = 504 bytes is reached (Bs = 12; N = floor((512)/(12 + 2)) = 36).

Order of 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 6 (left to right). 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 below the channel name. All channels are 2 bytes each except where otherwise statede.g. Pressure_bmp180. The endianess of the bytes for each channel is also specified.

Analog channels									Digital channels								
Accel (LN)	Battery	Ext Exp A7	Ext Exp A6	Ext Exp A15	Int Exp A12	Int Exp A13	Int Exp A14	Int Exp A1	Gyro_mpu	Accel (WR)_lsm	Mag_lsm	Accel_mpu	Mag_mpu	Temperature_bmp180	Pressure_bmp180	EXG1(24/16)	EXG2(24/16)
3	1	1	1	1	1	1	1	1	3	3	3	3	3	1	1	1	1
														2bytes	3bytes	7/5 bytes	7/5 bytes
little	little	little	little	little	little	little	little	little	big	little	big	big	little	big	big	big	big

Table 6 - Order of data channels for Shimmer3

8. Synchronising the data

8.1. Aligning data to a common clock

If synchronisation has been enabled, the clock of the master Shimmer can be taken as a reference clock and data from all slave Shimmers can be aligned with this reference clock by manipulating the timestamps associated with the slave data. It is important to note that this alignment is **not** done by the firmware and must be done by some processing in software after data logging has ended.

Shimmer offer a software application to automate the process of aligning the synchronised data call *Multi Shimmer Sync for SD*; please check www.shimmersensing.com for more information. Alternatively, the user may choose to write their own software, following the procedure outlined below.

1. Calculate the value of the slave timestamps:

   - Convert the logged 16-bit timestamps to a continuous value (i.e. one that does not return to zero when it reaches 65536).

- Note that these timestamps are relative to when logging started and must later be converted to a value relative to when the Shimmer clock was reset to zero.

2. Remove invalid offsets from the data, leaving only valid offsets, as shown in Figure 8-1:

    - Invalid offsets will have a value of 4294967295 (0xFFFFFFFF in hexadecimal format) in the 4-byte *offset magnitude*.

    - They will have been written to the SD card with every block for which a new offset message was not received.

3. Find the time on the slave clock at which the valid offset messages were received:

    - Use the first timestamp of the block containing the valid offset.

    - This step will generate a set of {*message time*, *offset*} pairs.

4. Apply the *offset sign* to the *offset magnitude*:

    Offset = (1-2**offset sign*)**offset magnitude*

Offset Sign	Offset Magn.	Timestamp
1	4294967295	2380
1	4294967295	3020
1	1010	3660
1	4294967295	4300
1	4294967295	4940
1	4294967295	5580
1	4294967295	6220
1	4294967295	14540
1	1006	15180
1	4294967295	15820
1	4294967295	367180
1	883	367820

Offset Sign	Offset Magn.	MessageTime
1	1010	3660
1	1006	15180
1	883	367820

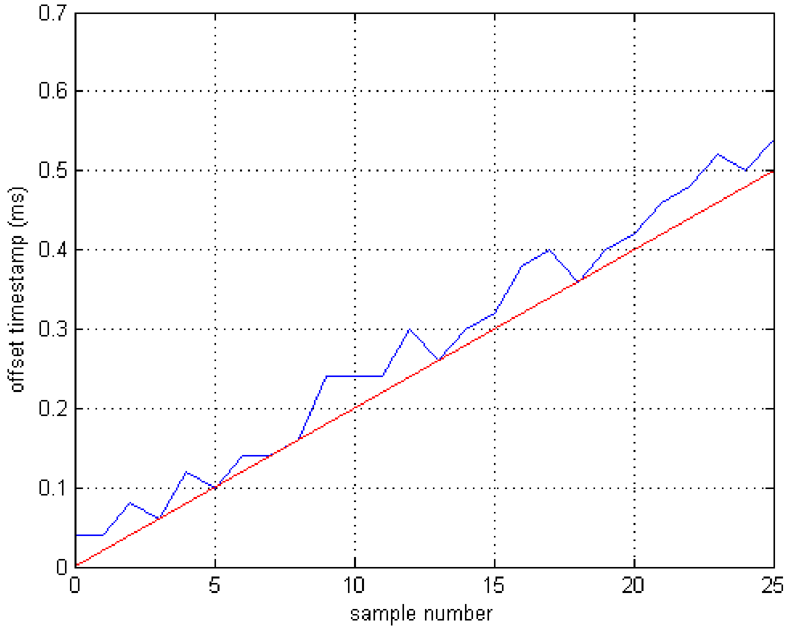
Offset	MessageTime
-1010	3660
-1006	15180
-883	367820

Figure 8-1: Steps 1 - 4

5. Interpolate the {*message time*, *offset*} pairs using the slave timestamps as shown in Figure 8-3:

    - Use linear interpolation.

- A minimum mean square error fit or similar will provide reasonable results. However, the ideal fit would produce a lower bound on the estimated offset values, as illustrated in Figure 8-3: Step 5-2



8-2: Step 5-1

- This step will generate an offset estimate for every sample.

Offset	MessageTime
-1010	3660
-1006	15180
-883	367820

Offset	Timestamp
-1010.44	2380
-1010.22	3020
-1010.0	3660
- 1009.78	4300
- 1009.56	4940
- 1009.33	5580
- 1009.11	6220
-1006.22	14540
-1006.0	15180
-1005.78	15820
-883.22	367180
-883.0	367820

Figure 8-3: Step 5-2

6. Convert the timestamps to the 32-bit slave clock value, relative to when the slave was rebooted (and its local clock reset to zero):

    - The initial timestamp (bytes 252 - 255 of the data file) is the time on the slave clock in 32-bit resolution when the first sample of that file was recorded.

    - The timestamp for the first logged data sample (sample 0) will be the 16-bit equivalent of this 32-bit value and can be used to calculate the constant value that should be added to all subsequent samples.

7. Subtract the offset estimates from the slave timestamps to give the estimated slave timestamps relative to the master clock:

    Aligned timestamp = Slave timestamp - Offset.

Initial Timestamp
67916

Offset	Timestamp
-1010.44	2380
-1010.22	3020
-1010.0	3660
- 1009.78	4300
- 1009.56	4940
- 1009.33	5580
- 1009.11	6220
-1006.22	14540
-1006.0	15180
-1005.78	15820
-883.22	367180
-883.0	367820

Offset	Timestamp
-1010.44	67916
-1010.22	68556
-1010.0	69196
- 1009.78	69836
- 1009.56	70476
- 1009.33	71116
- 1009.11	71756
-1006.22	80076
-1006.0	80716
-1005.78	81356
-883.22	432716
-883.0	433356

Timestamp
68926.4
69566.2
70206.0
70845.8
71485.6
72125.3
72765.1
81082.2
81722.0
82361.8
433599.2
434239.0

Figure 8-4: Steps 6 – 7

9. Appendices

9.1. Legacy Firmware: SDLog v0.5.0

Configuration header

Each data file (e.g. *data/experiment1/device1-000/000*), begins with a header with the 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 178 bytes in the header.

Byte # 0 – 9: Enabled Sensors

Byte #	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	Acc (LN)*	Gyr	Mag	N/A	N/A	GSR	ExtCh 7	ExtCh 6
1	N/A	N/A	Battery	Acc (WR)*	ExtCh 15	IntCh 1	IntCh 12	IntCh 13
2	IntCh 14					pres	N/A	exp_pwr
3								
4								
5								
6								
7								
8								
9								

* There are multiple accelerometers on the Shimmer3 circuit board. The low noise (LN) analog accelerometer on the KXRB5-2042 chip is enabled via Byte 0, Bit 7 (Acc (LN)), whilst the wide range digital accelerometer on the LSM303DLHC chip is enabled via Byte 1, Bit 4 (Acc (WR)).

Byte # 10 – 19: Trial Configuration

Byte #	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
10	5V Reg	PMUX	User button	Gyro button	1	Sync	Master	0
11	Single touch Start	Accel LPM	Accel HRM	tcxo				
12								
13								
14								
15	Broadcast Interval							
16								
17								
18								
19								

Byte # 20 – 29: Sensor Configuration

Byte #	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
20	Sample Rate (MSB)							
21	Sample Rate (LSB)							
22	Digital Accel Range							
23	GSR Range							
24	Mag Range							
25	Mag Rate							
26	Digital Accel Rate							
27	Gyro Rate							
28	Gyro Range							
29	Pressure Precision							

Byte # 30 – 39: Firmware and Shimmer Parameters

Byte #	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
30	ShimmerVersion (MSB)							
31	ShimmerVersion (LSB)							
32	MyTrialId							
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. *Shimmer1* is denoted by 0, *Shimmer2* by 1, *Shimmer2r* by 2 and *Shimmer3* 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 *SDLog v1.2*, *Major* = 1 and *Minor* = 1). 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 #	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
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 5.3.

Byte # 56-76: 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 #	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
56	Digital Accel Calibration Offset X (MSB)							
57	Digital Accel Calibration Offset X (LSB)							
58	Digital Accel Calibration Offset Y (MSB)							
59	Digital Accel Calibration Offset Y (LSB)							
60	Digital Accel Calibration Offset Z (MSB)							
61	Digital Accel Calibration Offset Z (LSB)							
62	Digital Accel Calibration Gain X (MSB)							
63	Digital Accel Calibration Gain X (LSB)							
64	Digital Accel Calibration Gain Y (MSB)							
65	Digital Accel Calibration Gain Y (LSB)							
66	Digital Accel Calibration Gain Z (MSB)							
67	Digital Accel Calibration Gain Z (LSB)							
68	Digital Accel Calibration Align XX							
69	Digital Accel Calibration Align XY							
70	Digital Accel Calibration Align XZ							
71	Digital Accel Calibration Align YX							
72	Digital Accel Calibration Align YY							
73	Digital Accel Calibration Align YZ							
74	Digital Accel Calibration Align ZX							
75	Digital Accel Calibration Align ZY							
76	Digital Accel Calibration Align ZZ							

Byte # 77 - 97: Gyroscope Calibration

Byte #	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
77	Gyro Calibration Offset X (MSB)							
78	Gyro Calibration Offset X (LSB)							
79	Gyro Calibration Offset Y (MSB)							
80	Gyro Calibration Offset Y (LSB)							
81	Gyro Calibration Offset Z (MSB)							
82	Gyro Calibration Offset Z (LSB)							
83	Gyro Calibration Gain X (MSB)							
84	Gyro Calibration Gain X (LSB)							
85	Gyro Calibration Gain Y (MSB)							
86	Gyro Calibration Gain Y (LSB)							
87	Gyro Calibration Gain Z (MSB)							
88	Gyro Calibration Gain Z (LSB)							
89	Gyro Calibration Align XX							
90	Gyro Calibration Align XY							
91	Gyro Calibration Align XZ							
92	Gyro Calibration Align YX							
93	Gyro Calibration Align YY							
94	Gyro Calibration Align YZ							
95	Gyro Calibration Align ZX							
96	Gyro Calibration Align ZY							
97	Gyro Calibration Align ZZ							

Byte # 98 - 118: Magnetometer calibration

Byte #	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
98	Mag Calibration Offset X (MSB)							
99	Mag Calibration Offset X (LSB)							
100	Mag Calibration Offset Y (MSB)							
101	Mag Calibration Offset Y (LSB)							
102	Mag Calibration Offset Z (MSB)							
103	Mag Calibration Offset Z (LSB)							
104	Mag Calibration Gain X (MSB)							
105	Mag Calibration Gain X (LSB)							
106	Mag Calibration Gain Y (MSB)							
107	Mag Calibration Gain Y (LSB)							
108	Mag Calibration Gain Z (MSB)							
109	Mag Calibration Gain Z (LSB)							
110	Mag Calibration Align XX							
111	Mag Calibration Align XY							
112	Mag Calibration Align XZ							
113	Mag Calibration Align YX							
114	Mag Calibration Align YY							
115	Mag Calibration Align YZ							
116	Mag Calibration Align ZX							
117	Mag Calibration Align ZY							
118	Mag Calibration Align ZZ							

Byte # 119 - 130: Reserved for future use

Byte # 131 - 151: Analog Accelerometer Calibration

Byte #	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
131	Analog Accel Calibration Offset X (MSB)							
132	Analog Accel Calibration Offset X (LSB)							
133	Analog Accel Calibration Offset Y (MSB)							
134	Analog Accel Calibration Offset Y (LSB)							
135	Analog Accel Calibration Offset Z (MSB)							
136	Analog Accel Calibration Offset Z (LSB)							
137	Analog Accel Calibration Gain X (MSB)							
138	Analog Accel Calibration Gain X (LSB)							
139	Analog Accel Calibration Gain Y (MSB)							
140	Analog Accel Calibration Gain Y (LSB)							
141	Analog Accel Calibration Gain Z (MSB)							
142	Analog Accel Calibration Gain Z (LSB)							
143	Analog Accel Calibration Align XX							
144	Analog Accel Calibration Align XY							
145	Analog Accel Calibration Align XZ							
146	Analog Accel Calibration Align YX							
147	Analog Accel Calibration Align YY							
148	Analog Accel Calibration Align YZ							
149	Analog Accel Calibration Align ZX							
150	Analog Accel Calibration Align ZY							
151	Analog Accel Calibration Align ZZ							

Byte # 152 - 173: Temperature (BMP180) and Pressure Calibration

Byte #	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
152	Temp & Pres Calibration AC1_MSB							
153	Temp & Pres Calibration AC1_LSB							
154	Temp & Pres Calibration AC2_MSB							
155	Temp & Pres Calibration AC2_LSB							
156	Temp & Pres Calibration AC3_MSB							
157	Temp & Pres Calibration AC3_LSB							
158	Temp & Pres Calibration AC4_MSB							
159	Temp & Pres Calibration AC4_LSB							
160	Temp & Pres Calibration AC5_MSB							
161	Temp & Pres Calibration AC5_LSB							
162	Temp & Pres Calibration AC6_MSB							
163	Temp & Pres Calibration AC6_LSB							
164	Temp & Pres Calibration B1_MSB							
165	Temp & Pres Calibration B1_LSB							
166	Temp & Pres Calibration B2_MSB							
167	Temp & Pres Calibration B2_LSB							
168	Temp & Pres Calibration MB_MSB							
169	Temp & Pres Calibration MB_LSB							
170	Temp & Pres Calibration MC_MSB							
171	Temp & Pres Calibration MC_LSB							
172	Temp & Pres Calibration MD_MSB							
173	Temp & Pres Calibration MD_LSB							

Initial Timestamp

Immediately after the configuration header, there is a 4-byte timestamp, whose value is the time on the local Shimmer 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 Shimmers to a common clock.

174	initial time stamp
175	initial time stamp
176	initial time stamp
177	initial time stamp

Data log

Following these 178 bytes, the logged sensor data begins (i.e. at the 179th 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).

Order of 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 below (left to right). 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 below the channel name. All channels are 2 bytes each except where otherwise stated i.e. Pressure_bmp180 and all channels are little endian.

Analog channels										Digital channels						
Accel (LN)	Battery	Ext Exp A7	Ext Exp A6	Ext Exp A15	Int Exp A1	Int Exp A12	Int Exp A13	Int Exp A14	msp430_Temperature	Gyro_mpu	Accel (WR)_lsm	Mag_lsm	Accel_mpu	Mag_mpu	Temperature_bmp180	Pressure_bmp180
3	1	1	1	1	1	1	1	1	1	3	3	3	3	3	1	1
																3bytes

9.2. Battery life estimates

Table 7 provides some typical values for the battery life of the Shimmer in a heavily loaded configuration. Future updates to this manual will include further information on battery life for other possible configurations. The user should note that battery life is affected by the age of the battery and the number of charge/discharge cycles that it has been put through. The data in Table 7 refer to a Shimmer 3.7V - 450mAh battery and should be used as a guideline only - these are not performance guarantees.

Firmware	Sensors Enabled	Sampling Rate (Hz)	Battery Life Duration (hh:mm)
SDLog	Accel (low noise)	51.2	120:00
SDLog	Accel (LSM303DLHC), Gyro, Mag (LSM303DLHC), Battery, Ext A7, Ext A6, Int A12.	256	23:00

Table 7 - Battery life estimates

Users should note that a high sampling frequency and/or a large number of channels (dependent on the enabled sensors) will mean that SD card write operations occur more frequently and, hence, the battery life will be significantly shorter in such configurations. The decrease in battery life with increased frequency of SD card write operations in non-linear.

10. Troubleshoot

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

The green and blue LEDs flashing on the Shimmer 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 Shimmer (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 Shimmer. 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 mayfail 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 Shimmer 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 7.2.

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