shimmer™

DISCOVERY IN MOTION

BtStream for Shimmer3

Firmware User Manual

Rev 0.4b

Legal Notices and Disclaimer

Redistribution IS permitted provided that the following conditions are met:
Redistributions must retain the copyright notice, and the following disclaimer. Redistributions in electronic 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 document.

Neither the name of Shimmer Research, or Realtime Technologies Ltd. nor the names of its contributors may be used to endorse or promote products derived from this document without specific prior written permission.

THIS DOCUMENT 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 DOCUMENT, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Table of Contents

1. Introduction

This document is an accompaniment to the *BtStream* Firmware v0.4.0 (or later) image for *Shimmer3*. No previous development experience is required.

Note that *BtStream* v0.4.0 is a Beta release.

BtStream firmware is a general purpose, fully configurable application to be used with the *Shimmer3* platform. As the name suggests, a *Shimmer3* programmed with *BtStream* firmware will stream data via a Bluetooth connection to a PC, mobile or other Bluetooth-enabled device.

BtStream firmware provides a complete solution, ready for use as-is for configurable data streaming and is fully compatible with the Shimmer Instrument Driver Libraries for LabVIEW and MATLAB as well as Shimmer APIs in Android and C#, and with *Multi Shimmer Sync* software applications. The source code is also openly available for any able user who may wish to modify or customise it to their own needs or, indeed, to use it as the basis for a new firmware application.

2. Scope of this User Manual

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

3. Pre-Requisites

BtStream for Shimmer3 firmware can be used with a *Shimmer3* device. A Bluetooth enabled device (PC, mobile, etc.) is required to interface with *Shimmer3* and receive the streamed data.

For *Shimmer2/2r*, please see the *BtStream for Shimmer2/2r Firmware User Manual*, available for download from www.shimmersensing.com.

4. Installation

Install the *BtStream* firmware v0.4.0 firmware image (BtStream_0.4.0_shimmer3.txt) onto a *Shimmer3* device using the *Shimmer3 Bootstrap Loader* (*Shimmer3 BSL*) program. The Shimmer3BSL.exe program is available via download from the user area at http://www.shimmersensing.com.

5. Using the firmware

To use the *BtStream* firmware, the device must first be paired with a PC, mobile or other Bluetooth-enabled device, as outlined in the *Shimmer User Manual*.

A *Shimmer3* programmed with *BtStream* firmware can be in one of three states: *Disconnected*, *Connected* or *Streaming*. When the *Shimmer3* is first powered on or reset, it is in the *Disconnected* state and will remain there until a connection is made over the Bluetooth link (i.e. by opening a serial connection).

In the *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 a command to start sampling is received, the *Shimmer3* goes into the *Streaming* state and starts sampling data from its sensors and sending that data over the Bluetooth link. This continues until a command to stop logging is received, whereupon the *Shimmer3* returns to the *Connected* state. Closing the serial connection will put the *Shimmer3* in the *Disconnected* state.

When the *Shimmer3* is in the *Connected* or *Streaming* 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.

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 7.1 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.

5.1. 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 5-1.

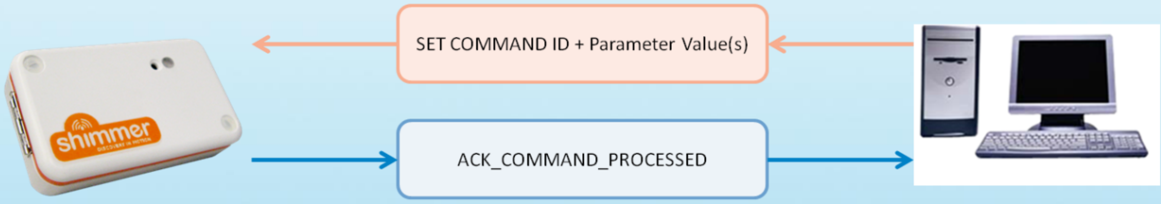

Figure 5-1 Packets sent for SET commands

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.

5.2. 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 5-2.

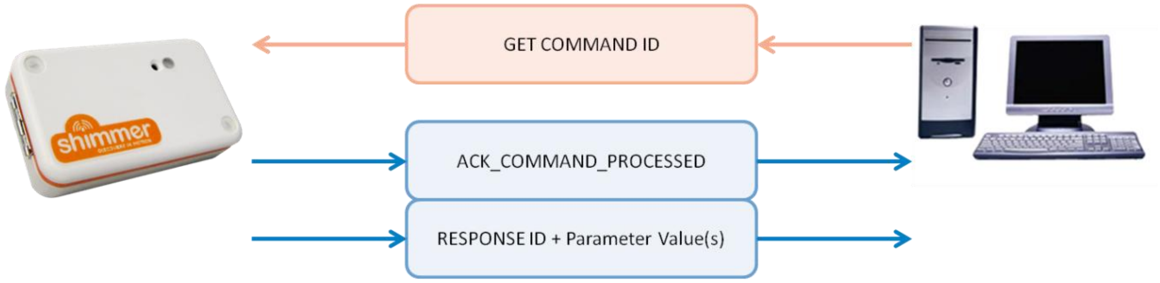

Figure 5-2 Packets sent for GET commands

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 5-1.

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 5-1 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 5-2 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
Magnetometer X*	7	i16*
Magnetometer Y*	8	i16*
Magnetometer Z*	9	i16*
Gyroscope X*	A	i16*
Gyroscope Y*	B	i16*
Gyroscope 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 5-2 Signal names, channel contents byte values and datatypes for available sensor signals

5.3. 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.

5.4. 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 5-3, 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 5-3 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 5-3. 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 low noise accelerometer, gyroscope, magnetometer and battery voltage at a rate of 51.2 Hz, with the gyroscope range set to +/- 500dps and the magnetometer range set to +/-1.3 Ga, and the data will be sent using a data buffer of size 1.

5.5. Configuration

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:

- Byte 0 - 1: Sampling rate
- Byte 2: Buffer size
- Bytes 3 - 5: Selected sensors
- Byte 6 - 9: Config bytes (Allows for 32 individual boolean settings)
- Bytes 10 - 30: Low Noise Accelerometer calibration values
- Bytes 31 - 51: Gyroscope calibration values

- Bytes 52 - 72: Magnetometer calibration values
- Byte 73 - 93: Wide Range Accelerometer calibration values

Selected Sensors

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

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

Config Bytes

The *Config bytes* contain the following parameters:

- Config Setup Byte 0:
  - Bit 7 – 4: Wide Range (LSM303DLHC) Accelerometer Data Rate.
  - Bit 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.
- Config Setup Byte 1:
  - Bit 7 – 0: MPU9150 Data Rate.

- Config Setup Byte 2:
    - Bit 7 – 5: (LSM303DLHC)Magnetometer Range.
    - Bit 4 – 2: (LSM303DLHC)Magnetometer Data Rate.
    - Bit 1 – 0: MPU9150 Gyroscope Range.
- Config Setup Byte 3:
    - Bit 7 – 6: MPU9150 Accelerometer Range.
    - Bit 5 – 4: BMP180 Pressure Resolution.
    - Bit 3 – 1: GSR Range
    - Bit 0: Internal Expansion Power Enable

Calibration Parameters

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.[1] 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.

5.6. BtStream firmware LED indicators

The *Shimmer3* has five LEDs in two locations: lower location A (green, yellow[2] and red); upper location B (green , blue), as shown in Figure 5-3.

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

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

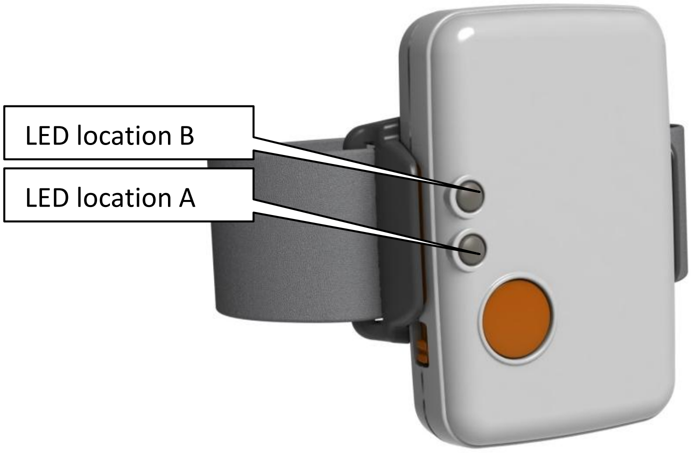

Figure 5-3 Shimmer3 LED Locations

The LEDs in Location A are used to indicate battery charge status, as outlined in Table 5-4.

On Dock/Multi Charger	**LED Pattern**	**Description**
Full Charge	🟢🟢🟢🟢🟢🟢	Green Solid ON
Charging	🟡🟡🟡🟡🟡🟡	Yellow Solid ON
Undocked		
Full Charge	🟢⚪⚪🟢⚪⚪	Green 3 ms ON/2s OFF
Medium Charge	🟡⚪⚪🟡⚪⚪	Yellow 3 ms ON/2s OFF
Low Charge	🔴⚪⚪🔴⚪⚪	Red 3 ms ON/2s OFF

Table 5-4 BtStream Battery Charge Status Indication

The LEDs in Location B are used to indicate operation status, as outlined in Table 5-5.

Docked / Undocked	**LED Pattern**	**Description**
Standby	🔵⚪⚪🔵⚪⚪	Blue 3 ms ON/2s OFF
Connected	🔵🔵🔵🔵🔵🔵	Blue Solid ON
Streaming	🔵⚪🔵⚪🔵⚪	Blue 1s ON / 1s OFF
Configuring	🔵⚪🔵⚪🔵⚪🔵⚪🔵⚪🔵⚪	Blue 0.1s ON/ 0.1s OFF
Error	🟢🔵🟢🔵🟢🔵🟢🔵🟢🔵🟢🔵	0.1s Blue/0.1s Breen

Table 5-5 BtStream Operation Status Indication

6. Further resources

There are sample python scripts available on https://github.com/ShimmerResearch/shimmer3 or from support@shimmersensing.com which will help to get new users up to speed with interfacing with a *Shimmer3* running *BtStream*. The *README.txt* document accompanying the scripts describes what each script does and, also, outlines how to bind the MAC address of the *Shimmer3* to an rfcomm port in Linux, in order to allow serial connections over Bluetooth.

7. Appendices

7.1. Bluetooth latency

Our lab tests have shown up to 100ms 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.

7.2. Shimmer.h file

```
/*
 * 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.
 *
 * @author Mike Healy
 * @date December, 2013
 */


#ifndef SHIMMER_H
#define SHIMMER_H

//these are defined in the Makefile for BtStream (TinyOS)
```

```
#define DEVICE_VER      3     //Represents shimmer3
#define FW_IDENTIFIER   1     //Two byte firmware identifier number
#define FW_VER_MAJOR    0     //Major version number: 0-65535
#define FW_VER_MINOR    4     //Minor version number: 0-255
#define FW_VER_REL      0     //Release candidate version number: 0-255


// 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  //maintain compatibility
with Shimmer2/2r BtStream
                                                            //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     //TODO: remove from
release version
#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 SET_BT_COMMS_BAUD_RATE                        0x6A     //11 allowable options:
0=115.2K(default), 1=1200, 2=2400, 3=4800,
                                                               //4=9600, 5=19.2K,
6=38.4K, 7=57.6K, 8=230.4K, 9=460.8K, 10=921.6K
                                                               //Need to disconnect BT
connection before change is active
#define BT_COMMS_BAUD_RATE_RESPONSE                   0x6B
#define GET_BT_COMMS_BAUD_RATE                        0x6C
//0x70 to 0x87 and 0xE0 reserved for Log+Stream
#define ACK_COMMAND_PROCESSED                         0xFF
```

```
//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 SENSOR_STRAIN            0x80     //higher priority than SENSOR_INT_A13 and
SENSOR_INT_A14
#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 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         66    //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
#define STRAIN_HIGH                      0x27
#define STRAIN_LOW                       0x28


// Infomem contents
#define NV_NUM_SETTINGS_BYTES            31
#define NV_NUM_CALIBRATION_BYTES         84
#define NV_TOTAL_NUM_CONFIG_BYTES        115

#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
#define NV_CONFIG_SETUP_BYTE1            7
#define NV_CONFIG_SETUP_BYTE2            8
#define NV_CONFIG_SETUP_BYTE3            9
#define NV_EXG_ADS1292R_1_CONFIG1        10
#define NV_EXG_ADS1292R_1_CONFIG2        11
#define NV_EXG_ADS1292R_1_LOFF           12
#define NV_EXG_ADS1292R_1_CH1SET         13
#define NV_EXG_ADS1292R_1_CH2SET         14
#define NV_EXG_ADS1292R_1_RLD_SENS       15
#define NV_EXG_ADS1292R_1_LOFF_SENS      16
#define NV_EXG_ADS1292R_1_LOFF_STAT      17
#define NV_EXG_ADS1292R_1_RESP1          18
#define NV_EXG_ADS1292R_1_RESP2          19
#define NV_EXG_ADS1292R_2_CONFIG1        20
#define NV_EXG_ADS1292R_2_CONFIG2        21
#define NV_EXG_ADS1292R_2_LOFF           22
#define NV_EXG_ADS1292R_2_CH1SET         23
#define NV_EXG_ADS1292R_2_CH2SET         24
#define NV_EXG_ADS1292R_2_RLD_SENS       25
#define NV_EXG_ADS1292R_2_LOFF_SENS      26
#define NV_EXG_ADS1292R_2_LOFF_STAT      27
```

```
#define NV_EXG_ADS1292R_2_RESP1           28
#define NV_EXG_ADS1292R_2_RESP2           29
#define NV_BT_COMMS_BAUD_RATE             30
#define NV_A_ACCEL_CALIBRATION            31
#define NV_MPU9150_GYRO_CALIBRATION       52
#define NV_LSM303DLHC_MAG_CALIBRATION     73
#define NV_LSM303DLHC_ACCEL_CALIBRATION   94


//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
//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 INT_EXP_POWER_ENABLE                    0x01


//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_GSR     0x0020   //uses ADC1
#define MASK_STRAIN  0x0180   //uses ADC13 and ADC14


//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

//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

//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
#define GSR_AUTORANGE               0x04

#endif
```

Shimmer
Innovation House
DCU Innovation Campus
Glasvevin
Dublin 9
Ireland

T: +353 (1)687 5760
E: info@shimmersensing.com

Shimmer North America
8th Floor (#8100)
28 State Street Boston
Boston
MA 02109
USA

T: +1 617-619-5311
E: info@shimmersensing.com

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