shimmer
DISCOVERY IN MOTION

Shimmer MATLAB™ Instrument Driver User Manual

Revision 2.2b

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

The *Shimmer* MATLAB™ *Instrument Driver* allows for realtime streaming of data from a Shimmer device into MATLAB™. It is designed to work with MATLAB™ 2008a (version 7.6) or later on a Windows OS (the driver has been tested on Windows 7).

2. Pre-Requisites

In order to use the *Shimmer* MATLAB™ *Instrument Driver* you will need the following:

1. A copy of the Shimmer User Manual available for download from the documentation section of the Shimmer website (http://www.shimmersensing.com/support/wireless-sensor-networks-documentation/).

2. MATLAB™[1] 2008a (version 7.6) or later installed on your PC.

3. A *Shimmer2*/*Shimmer2r*/*Shimmer3* device programmed with the latest version of BtStream. This is available for download from http://www.shimmersensing.com/support/wireless-sensor-networks-download/. These images can be loaded onto the Shimmer devices using the *Shimmer Bootstrap Loader* applications for *Shimmer2/2r* and *Shimmer3*. See the *Shimmer User Manual* for details.

   The Shimmer needs to be paired with the PC (over Bluetooth); please refer to the *Shimmer User Manual* for information on how to pair your Shimmer with your PC.

4. Ensure you are using the latest version of the Instrument Driver and User guide, by checking the Downloads section of the Shimmer website for updates (http://www.shimmersensing.com/support/wireless-sensor-networks-download/).

3. Installation

3.1. Download

Download the folder *Shimmer* MATLAB™ *Instrument Driver* and copy it to a suitable location.

3.2. Install Realterm

Download and install an application called *Realterm Serial Terminal*, which acts as the communication link between the Shimmer device and MATLAB™. Realterm version 2.0.0.57 was used in the development of this driver and can be downloaded from http://sourceforge.net/projects/realterm/files/Realterm/2.0.0.57/.

Important 1: When you are installing Realterm you need to ensure that the option to *register Realterm automation server* is ticked when presented with the window at the end of the installation process as highlighted in Figure 1 below.

[1] The MathWorks Inc., 3 Apple Hill Drive, Natick, MA 01760-2098, USA.

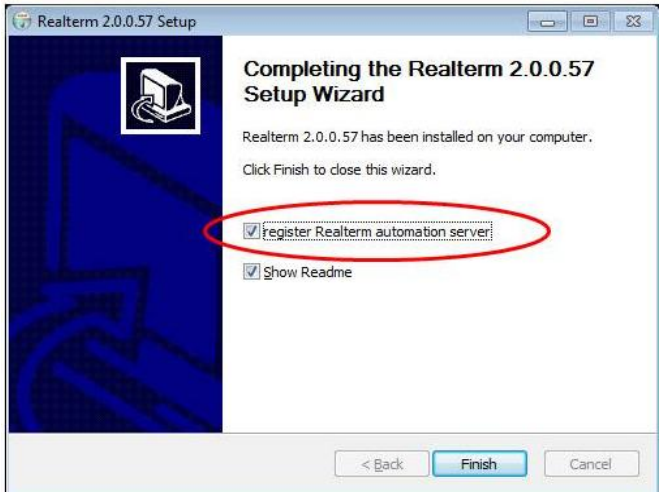


Figure 1 register Realterm automation server

3.3. Windows Users - Change User Account Control Settings

Users of Windows need to disable the *User Account Control Settings* to allow Realterm to run with administrator privileges always. For Windows 7, in order To locate the *User Account Control Settings* control panel you should search for it in the Start Menu. The control panel is illustrated in Figure 2. You need to set this control to *Never Notify*. You will need to restart your computer for the setting to take effect.

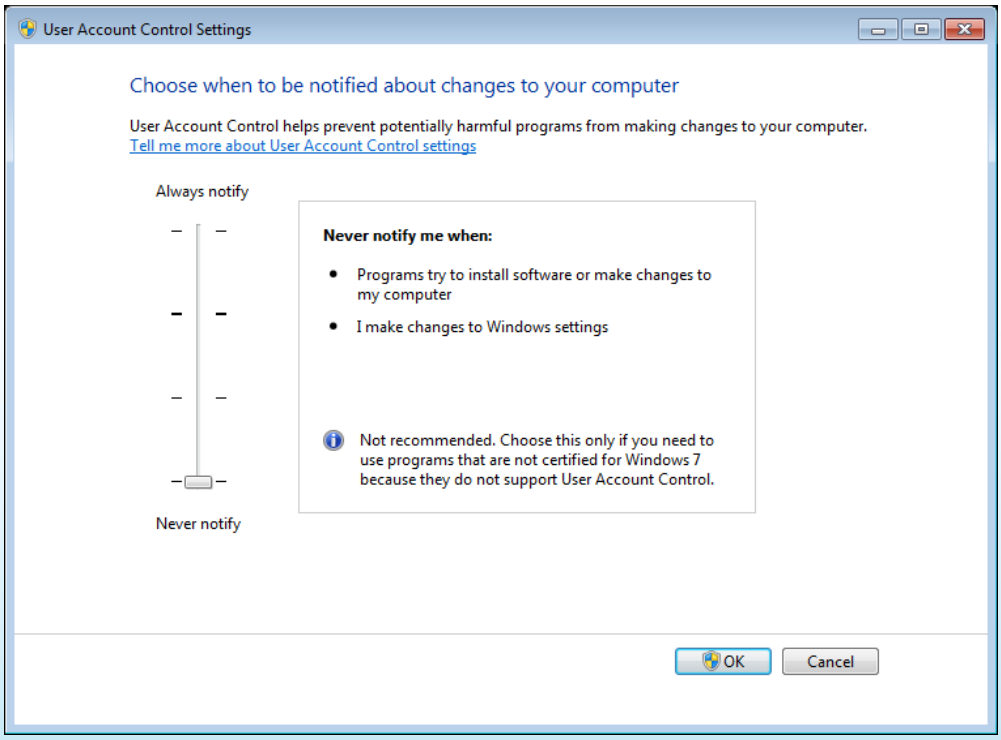


Figure 2 User Account Control Settings (Windows 7)

4. Getting Started

Start MATLAB™ and set the current working directory to the *Shimmer* MATLAB™ *Instrument Driver* folder you downloaded previously. The folder contains the following files:

- *Readme.txt*
- *ShimmerHandleClass.m*
- *plotandwriteexample.m*
- *orientation3Dexample.m*
- *twoshimmerexample.m*
- *plotandwriteemgexample.m*
- *plotandwriteecgexample.m*
- *plotandwriteexgtestsignalexample.m*
- *ppgtoheartrateexample.m*
- *FilterClass.m*
- *ShimmerPPGtoHR_Rev_X_Y.jar*
- *plotandwriteexamplelegacy.m*
- *twoshimmerexamplelegacy.m*

4.1. ShimmerHandleClass

The file *ShimmerHandleClass.m* contains the code for the class, *ShimmerHandleClass*, which can be used to create Shimmer objects in MATLAB™. In the command window, type *doc ShimmerHandleClass* to get an explanation of the properties and methods of the class.

4.2. FilterClass

FilterClass.m contains an implementation of a configurable Chebyshev filter, which does not rely on the Matlab toolboxes. Examples of how to use this filter implementation can be found in the *plotandwriteemgexample.m*, *plotandwriteecgexample.m* and *ppgtoheartrateexample.m*.

4.3. ShimmerPPGtoHR Java Archive

A Java Archive (.jar file) is provided with the Instrument Driver to allow photoplethysmogram (PPG) data from an optical pulse sensor to be converted to heart rate.

To use the Java PPGtoHR library in conjunction with the MATLAB ID, save the ShimmerPPGtoHR_Rev_X_Y.jar file to a local directory on your PC (e.g. C:\Users\username\Documents\MATLAB\Shimmer Matlab Instrument Driver vx.x) and add it to the Java dynamic class path in Matlab, using the following command:

javaaddpath('C:\Users\username\Documents\MATLAB\Shimmer Matlab Instrument Driver vx.x\ShimmerPPGtoHR_Rev_X_Y.jar');

See Section 4.4 for an example of how to use this functionality.

4.4. Examples

The file, *plotandwriteexample.m*, contains a function which is used to demonstrate the use of the *ShimmerHandleClass*. Type *help plotandwriteexample* to get an explanation of the function.

The file, *orientation3Dexample.m*, contains a function which is used to demonstrate the ability of the *ShimmerHandleClass* to estimate the orientation of a Shimmer device in 3D and an example of how to visualise the orientation on screen. Type *help orientation3Dexample* to get an explanation of the function.

The file, *twoshimmerexample.m*, contains a function which is used to further demonstrate the use of the *ShimmerHandleClass*. Type *help twoshimmerexample* to get an explanation of the function.

The three ExG example files, *plotandwriteemgexample.m, plotandwriteecgexample.m* and *plotandwriteexgtestsignalexample.m*, demonstrate the ExG functionality for *Shimmer3* in combination with the *ExG Expansion Board*. For more information refer to Section 10 of this document.

In *ppgtoheartrateexample.m,* photoplethysmogram (PPG) data from an optical pulse sensor is converted to heart rate. The PPG data is pre-filtered using a second order Chebyshev low pass filter (LPF) with a corner frequency of 5Hz and a second order Chebyshev high pass filter (HPF) with a corner freq of 0.5Hz, by using *FilterClass.m*. This example makes use of the Shimmer Java PPGtoHR library.

4.5. Legacy Examples

There are also two legacy files from an early beta release provided for reference, *plotandwriteexamplelegacy.m,* and *twoshimmerexamplelegacy.m*. The legacy examples use the method 'getuncalibrateddata', which is a legacy method from the beta release. The user is advised to use the replacement method, 'getdata', as demonstrated in *plotandwriteexample.m*.

5. Using the Instrument Driver

5.1. Plotandwriteexample

The best way to start using the instrument driver is to go through the *plotandwriteexample.m* mentioned in the previous section. In this section, some key functions within the example which users should take note of when using the instrument driver are presented.

- *shimmer = ShimmerHandleClass(comPort);*
    - creates a *ShimmerHandleClass* which takes in a string value of the Com Port given as the input argument *comPort*; each Shimmer device which has been paired to the PC will have a Com port associated with it.
- *shimmer.connect;*
    - connects the PC over Bluetooth to the Shimmer device whose Com port was specified in the *comPort* field for the *ShimmerHandleClass*.
- *shimmer.setsamplingrate(51.2);*
    - transmits a command to set the sampling rate of the Shimmer device. A list of supported sampling rates can be found in the Resources folder.
- *shimmer.setinternalboard('9DOF');*
    - specifies which daughter board is attached to the Shimmer device you have connected to; in the example, the *Shimmer IMU 9DoF daughterboard* is selected.
- *shimmer.setenabledsensors('Gyro',1,'Mag',1,'Accel',1,'BattVolt',1);*
    - transmits a command to the Shimmer device, telling it which sensors it should enable; in the example, the gyroscope, magnetometer, accelerometer and battery voltage monitor are enabled.
- *shimmer.start;*
    - transmits a start streaming command to the Shimmer device.
- *shimmer.stop;*
    - transmits a stop streaming command to the Shimmer device.
- shimmer.disconnect;
    - disconnects the Bluetooth connection between the PC and the Shimmer device.

Readers should note that *Timestamp* data can be returned in a raw (*RAW*) and in a calibrated (*CAL*) format. RAW timestamp data is generated from the crystal oscillator on the Shimmer which has a

frequency of 32768 Hz. It is a 16-bit value and will loop around 0 when it exceeds 65536. CAL timestamp data is in units of msec.

5.2. List of commands

Table 5-1 contains the list of commands available in the *ShimmerHandleClass*. The "Usage" column denotes the versions of Shimmer hardware with which each command is compatible: *Shimmer2* (2), *Shimmer2r* (2r) and/or *Shimmer3* (3).

Command	Description	Usage
setsamplingrate	Set the sampling rate of the Shimmer.	2/2r/3
setenabledsensors	Enables/disables the sensors on the Shimmer.	2/2r/3
setconfigbyte0	Set the config byte0 on the Shimmer.	2/2r
setfivevoltreg	Set the 5 volt regulator bit on the Shimmer.	2/2r
setpmux	Set the PMux bit on the Shimmer.	2/2r
setledblink	Set which LED is blinking on the Shimmer.	2/2r/3
setaccelrange	Set the accelerometer range of the Shimmer.	2/2r/3
setgsrrange	Set the Gsr Range on the Shimmer.	2/2r/3
setmagrange	Set the Mag Range on the Shimmer.	2/2r/3
setmagrate	Set the Mag Data Rate on the Shimmer.	2/2r/3
setinternalboard	Sets the internal daughter board on the Shimmer.	2/2r/3
setexternalboard	Sets the external daughter board on the Shimmer.	2/2r/3
setbattlimitwarning	Set the battery voltage at which a low battery warning will occur.	2/2r/3
setgyroinusecalibration	Enable/disable gyro in-use calibration.	2/2r/3
setbuffersize	Set the number of samples sent in each data packet.	2/2r/3
setemgcalibrationparameters	Store the EMG calibration parameters on the Shimmer.	2/2r/3
setecgcalibrationparameters	Store the ECG calibration parameters on the Shimmer.	2/2r/3
getcomport	Get the Com Port of the Shimmer.	2/2r/3
getstate	Get the state of the Shimmer.	2/2r/3
getshimmerversion	Get the version number of the Shimmer.	2/2r/3
getsamplingrate	Get the sampling rate of the Shimmer.	2/2r/3
getconfigbyte0	Get the config byte0 setting of the Shimmer.	2/2r
getfivevoltreg	Get the 5 Volt Regulator setting of the Shimmer.	2/2r/3
getpmux	Get the PMux setting of the Shimmer.	2/2r
getaccelrange	Get the accelerometer range of the Shimmer.	2/2r/3
getgsrrange	Get the gsr range of the Shimmer.	2/2r/3
getmagrange	Get the mag range of the Shimmer.	2/2r/3
getmagrate	Get the mag data rate of the Shimmer.	2/2r/3
getinternalboard	Get the internal board setting of the Shimmer.	2/2r/3
getexternalboard	Get the external board setting of the Shimmer.	2/2r/3
getcalibrationparameters	Get calibration parameters for Accel/Mag/Gyro.	2/2r/3
getenabledsignalnames	Get the names of the enabled sensor signals.	2/2r/3
getsignalname	Get the name of a sensor signal.	2/2r/3
getsignalindex	Get the index of a sensor signal.	2/2r/3

getdata	Get any type of available sensor data.	2/2r/3
getpercentageofpacketsreceived	calculate percentage of packets received.	2/2r/3
connect	Connect to the Shimmer over Bluetooth.	2/2r/3
start	Send start streaming command to the Shimmer.	2/2r/3
stop	Send stop streaming command to the Shimmer.	2/2r/3
toggleLED	Send command to the Shimmer to toggle the red LED.	2/2r/3
disconnect	Close serial connection with the Shimmer.	2/2r/3
setinternalexppower	Enable to power the external sensor when using the Shimmer GSR+ Expansion Board or Resistance input of Bridge Amplifier+ Expansion Board	3
getuncalibrateddata	Get uncalibrated data from the data buffer. Note: Deprecated function, use getdata instead.	2/2r/3
getconfigbytes	Get the config bytes setting (only config byte0 for Shimmer2 , Shimmer2r)	2/2r/3
getexggain	Get the exg gain.	3
getexgrate	Get the exg rate.	3
getinternalexppower	Get the internal expansion power setting.	3
getpressureresolution	Get the pressure resolution.	3
getaccellpmode	Get acceleration low power mode setting	3
getaccelhrmode	Get acceleration high resolution mode setting	3
setaccelhrmode	Set acceleration high resolution mode	3
setaccellpmode	Set acceleration low power mode	3
setgyrorate	Set the data rate of the gyroscope.	3
setaccelrate	Set the data rate of the digital accelerometer.	3
setgyrorange	Set the range of the gyroscope.	3
setpressureresolution	Set the pressure resolution.	3
setconfigbytes	Set the config bytes.	3
setdefaultecgparameters	Set default ecg parameters.	3
setdefaultemgparameters	Set default emg parameters.	3
setexgconfiguration	Set the exg configuration.	3
setexggain	Set the exg gain.	3
setexgrate	Set the exg rate.	3
setexgtestsignalparameters	Set default exg testsignal parameters.	3

Table 5-1 ShimmerHandleClass Commands

For further information on each command, either use the MATLAB™ help function or directly refer to the *ShimmerHandleClass.m* file. To use MATLAB™ help, select (highlight) the command and press F1 as shown in the example in Figure 3.

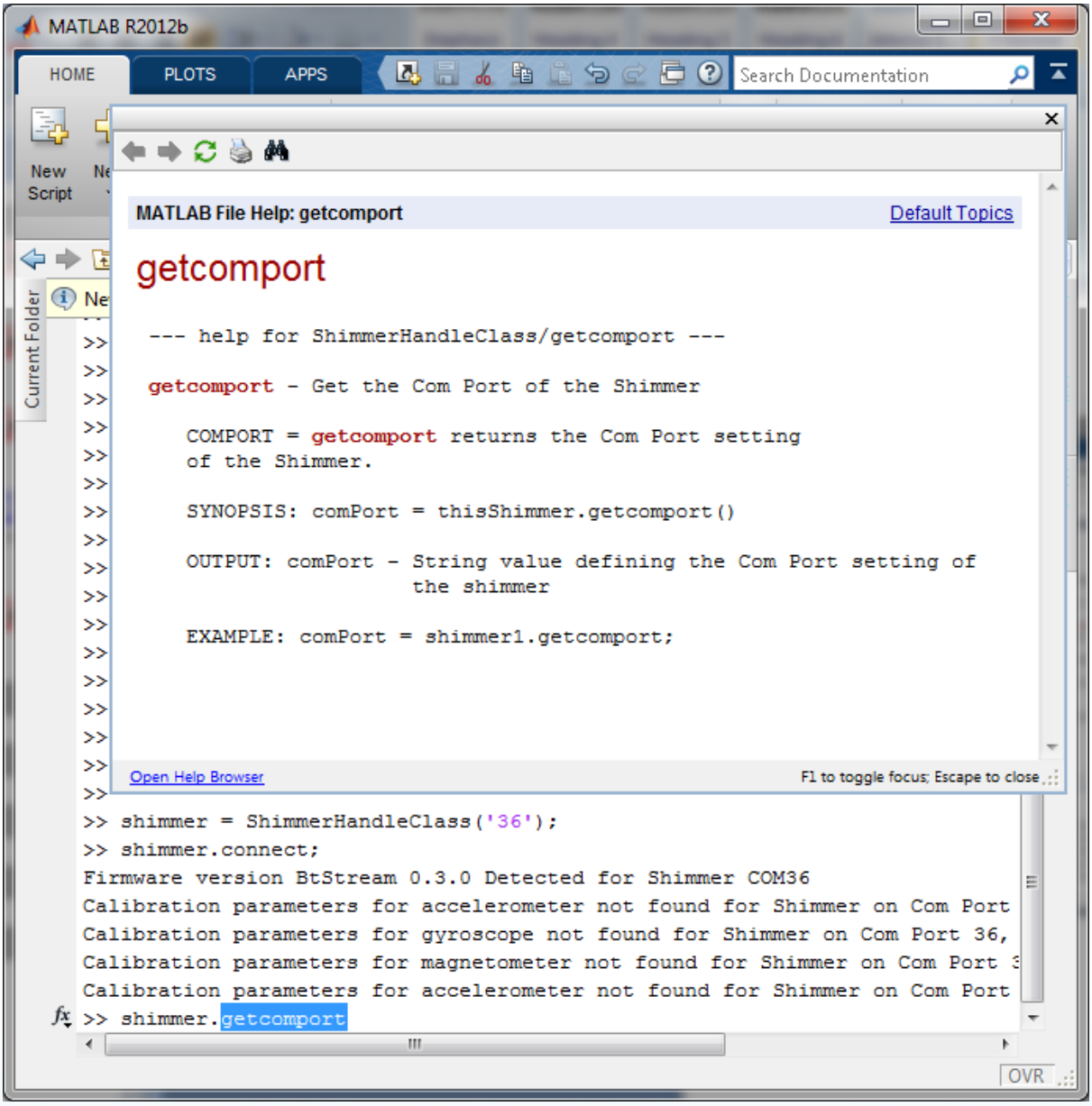


Figure 3: Using Help

6. Usage Considerations

6.1. General

The *getdata()* method is provided for users to retrieve data received from the Shimmer unit. The method relies on the *capturedata()* method which reads data from the buffer. The user should note that once the *capturedata()* method is completed, the buffer is cleared. Thus, to retrieve data from multiple sensors or in multiple formats, data retrieval methods should NEVER be used consecutively; instead, multiple arguments should be used in a single call to the data retrieval method. Figure 4 shows the wrong way (left) and the correct way (right) to retrieve calibrated gyroscope and accelerometer data.

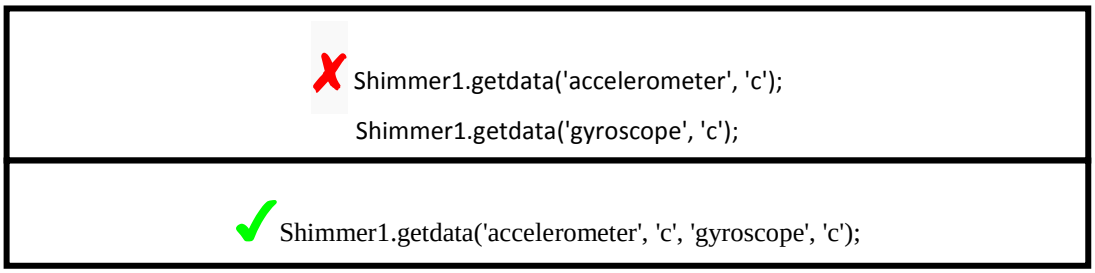


Figure 4: Retrieving Data

The *getdata()* method returns four different arrays, e.g.:

[sensorData,signalName,signalFormat,signalUnit] = Shimmer1.getdata('accelerometer','c');.

The first array, *sensorData*, holds the sensor data, whilst the other three arrays describe the content of the data. The second array identifies the Property/Signal Name; this is the name of the signal (e.g. Timestamp, Accelerometer X, EMG, etc). The third array identifies the format of the data, (e.g. CAL or RAW). The forth array identifies the units of the signal (e.g. deg/s, mVolts, etc.).

Note: An asterisk after the *Units* indicates that default offset and sensitivity values from the sensor data sheet have been used to calibrate the sensor data (e.g. *mVolts**).

6.2. Differences between *Shimmer2r* and *Shimmer3*

When using the *Shimmer MATLAB™ Instrument Driver*, readers should take note of some fundamental differences between *Shimmer2r* and *Shimmer3*. The first is that the following two commands for *Shimmer2r* hardware are not available for *Shimmer3*:

SET_5V_REGULATOR_COMMAND (when using the *External Expansion Board* with *Shimmer2r*)

SET_PMUX_COMMAND (when monitoring battery voltage on *Shimmer2r*)

Inertial Sensors

In terms of accelerometers, the *Shimmer3* has more than a single Accelerometer. Currently the *Shimmer MATLAB™ Instrument Driver* supports the Low Noise analog accelerometer (KXRB5-2042) and the Wide Range Accelerometer (LSM303DLHC). Since *Shimmer MATLAB™ Instrument Driver v2.2*

using the *getdata* method to retrieve accelerometer data, only inserts the Low Noise and Wide Range Accelerometer for *Shimmer3* as shown in Figure 5.

A	B	C	D	E	F	G	H	I	J	K	L
TimeStamp	LowNoiseAccelerometerX	LowNoiseAccelerometerY	LowNoiseAccelerometerZ	GyroscopeX	GyroscopeY	GyroscopeZ	MagnetometerX	MagnetometerY	MagnetometerZ	VSenseBatt	VSenseBatt1
NUMBER	NUMBER	NUMBER	NUMBER	NUMBER	NUMBER	NUMBER	NUMBER	NUMBER	NUMBER	NUMBER	NUMBER
Time Stamp	Low Noise Accelerometer X	Low Noise Accelerometer Y	Low Noise Accelerometer Z	Gyroscope X	Gyroscope Y	Gyroscope Z	Magnetometer X	Magnetometer Y	Magnetometer Z	VSenseBatt	VSenseBatt
19.531	-0.40964	0.26506	9.7229	0	0	0	-162	471	579	2873	4209.5
39.063	-0.42169	0.24096	9.7108	43.298	-2.5038	23.985	-159	479	591	2865	4197.8
58.594	-0.44578	0.24096	9.6988	0.36641	1.7863	2.4427	-160	491	580	2850	4175.8
78.125	-0.43373	0.25301	9.6627	0.24427	1.7252	2.3511	-162	504	575	2865	4197.8
97.656	-0.39759	0.21687	9.6627	0.10687	1.6489	2.3053	-169	496	587	2865	4197.8
117.19	-0.40964	0.21687	9.7108	0.35115	1.7099	2.3969	-161	475	578	2873	4209.5
136.72	-0.43373	0.22892	9.7229	0.30534	1.8473	2.3206	-171	483	584	2864	4196.3
156.25	-0.42169	0.24096	9.6506	0.50382	1.7557	2.2443	-162	493	580	2850	4175.8
175.78	-0.42169	0.25301	9.6506	0.36641	1.8931	2.2748	-159	478	588	2868	4202.2
195.31	-0.44578	0.22892	9.7349	0.42748	1.6794	2.2595	-165	495	588	2862	4193.4
214.84	-0.44578	0.22892	9.6988	0.30534	1.7405	2.4122	-159	476	586	2852	4178.8
234.38	-0.43373	0.25301	9.6386	0.42748	1.8321	2.2901	-173	495	589	2867	4200.7

Figure 5: Extra Fields for Accelerometer

In earlier versions of the instrument driver three extra fields: 'Accelerometer X', 'Accelerometer Y', 'Accelerometer Z') were inserted with duplicated values of either the Low Noise or Wide Range Accelerometer. These extra fields were included to maintain backwards compatibility in case 'Accelerometer X', 'Accelerometer Y', 'Accelerometer Z' were used by developers using the instrument driver.

Also, readers should note that the data rate of the gyroscope (MPU9150), accelerometer (LSM303DLHC) and magnetometer (LSM303DLHC) are configurable for the *Shimmer3*.

Pressure Sensor

The *Shimmer3* also includes a Pressure sensor (BMP180). The pressure sensor's resolution is configurable (ultra low power, standard, high resolution and ultra high resolution). Users should take note of the Max Conversion time when selecting the sampling rate and the resolution mode; further details can be found in the BMP180 data sheet.

Optical Pulse

There is an external sensor option when using the Shimmer *GSR+ Expansion Board* for Shimmer3, which was not available for *Shimmer2r*. Among other things, this can be used to measure photoplethysmogram (PPG) data, using the *Shimmer Optical Pulse Sensor Probe*. To measure PPG with a GSR+ Expansion Board, the following must be enabled:

- Int ADC A13/PPG - *The output of the sensor connected to the GSR+ Expansion Board is measured by the internal ADC A13*
- Internal Exp Power (*to power the external sensor e.g. PPG*)

For example, when using the *GSR+ Expansion Board,* executing the following commands will enable the internal expansion power and the Internal ADC A13 channel:

shimmer.setinternalboard('GSR');

shimmer.setenabledsensors(' INT A13',1);

shimmer.setinternalexppower(1);

PPG data output will be under the signal name, 'Internal ADC A13'. For further information on the *GSR+ Expansion Board*, users are encouraged to read the *Shimmer GSR+ Expansion Board User Guide*.

PPG data can also be measured on *Shimmer3* via the PROTO3 Deluxe expansion board. For details on which channels should be enabled, please refer to the Optical Pulse Sensor Probe User Guide .

ECG and EMG

Support for using the *ExG Expansion Board* for *Shimmer3* is incorporated in the instrument driver from *v2.1* and later, allowing ECG or EMG data to be measured. In order to use the *ExG Expansion Board* with *Shimmer3,* the *internalboard* setting must be set to 'EXG', 'ECG' or 'EMG' and the relevant sensors need to be enabled. For *Shimmer3*, the ExG data resolution is either 16-bit or 24-bit. For *Shimmer2/2r*, the ECG/EMG data resolution is always 12-bit.

For example, executing the following commands:

```
shimmer.setinternalboard('EMG');

shimmer.setenabledsensors('EMG',1);
```

sets the internalboard to 'EMG' and enables the sensor 'EMG'.

In Section 10 of this document, the ExG functionality in the *Shimmer* MATLAB™ *Instrument Driver v2.1* is further explained. For further information on using the *ExG Expansion Board* in combination with *Shimmer3*, users are also encouraged to read the *ExG User Guide for EMG* and/or the *ExG User Guide for ECG*.

Bridge Amplifier+

The *Bridge Amplifier+ Expansion Board* for *Shimmer3* has an extra input that can be used to measure resistance-based metrics, such as temperature. To use this input, the following must be enabled:

- Int ADC A1 - The output of the sensor connected to the Bridge Amplifier+ Expansion Board is measured on the Internal ADC A1 Channel.
- Internal Exp Power (to power the external sensor)

Users should refer to the *Bridge Amplifier+ User Guide* for more details.

7. Battery Monitoring and Low Battery Warnings

In the *Shimmer* MATLAB™ *Instrument Driver v1.4* and later, battery monitoring can be done by using the `setenabledsensors` command.

For *Shimmer2/2r*, battery monitoring is done via the same ADC channels that are used by ExpBoard A0 and ExpBoard A7; thus, while monitoring the battery you will be unable to use those ports. For *Shimmer3*, there is a dedicated battery monitoring channel.

There is also a low battery warning functionality built in, which will cause the green LED of the Shimmer device to turn yellow when the voltage has dropped below a specified value. By default, the value of the limit is set to 3.4 Volts. This value can be specified via the command `setbattlimitwarning`. Low battery notification is only supported on Shimmer devices which have the *BtStream* firmware installed.

In order to use both battery monitoring and low battery notification on the Shimmer device, the battery voltage sensor has to be enabled prior to streaming. In the case of low battery, the warning is sent to the Shimmer device after the *getdata* (e.g. `shimmer.getdata('Time Stamp','c','Battery Voltage','a'))` is executed. When *getdata* is executed, the driver checks the battery data, and sends a command to change the LED on the Shimmer device if the battery value is below the limit. Users should note that after the warning is transmitted the ACK from the Shimmer device is only retrieved when *getdata* is executed again.

8. 3D orientation estimation

In the *Shimmer* MATLAB™ *Instrument Driver v1.5* and later, the 3D orientation of the Shimmer device can be estimated; this requires that a *Shimmer IMU 9DoF* daughterboard be attached to the baseboard, that the internal board be set to *9DOF* and that the accelerometer, gyroscope and magnetometer are all enabled.

The orientation estimates for each sample are calculated and output in quaternion format if the option, *'quaternion'*, is passed to the *getdata()* function. For an example of how this method is used, refer to the *orientation3Dexample.m* sample code.

9. Gyroscope in-use calibration

In the *Shimmer* MATLAB™ *Instrument Driver v1.5* and later, the offset bias for the gyroscope on the Shimmer device can be estimated during use to improve calibration stability over time.

For *Shimmer2r*, this requires that a *Shimmer IMU 9DoF* or *Shimmer IMU Gyro* daughterboard be attached to the baseboard. The internal board needs to be set to *9DOF* or *Gyro* and the gyroscope must be enabled. For Shimmer3, the 9DoF sensors are all contained on the mainboard.

The method is enabled by calling the function, *setgyroinusecalibration*, with an argument of 1. If this method is enabled, the uncalibrated gyroscope data is continuously buffered in a buffer of length 2 seconds, and the function, *nomotiondetect*, is called from the *getdata* function to determine if the device is moving or motionless. If the device remains motionless for 2 seconds, the gyroscope offset bias vector is updated. For an example of how this method is enabled, refer to the *orientation3Dexample.m* sample code.

Note: the updated value of the offset bias vector is used locally in the instrument driver only and is not sent to the Shimmer device.

10. ExG functionality for *Shimmer3*

Support for using *Shimmer3* with the *ExG Expansion Board* is incorporated in the *ShimmerHandleClass*.m of Shimmer MATLAB™ *Instrument Driver v2.1* and later.

The following examples are provided to demonstrate this functionality:

- *plotandwriteemgexample.m.*
- *plotandwriteecgexample.m.*
- *plotandwriteexgtestsignalexample.m*.

The first two examples explain how to plot and write EMG and ECG data, respectively. The third example shows how to use the ExG testsignal that is generated by the chips on the *ExG Expansion Board*; this is a square wave with a frequency of 1Hz and an amplitude of ±1mV.

10.1. Key Functions

Some key functions regarding ExG are highlighted below:

- setinternalboard
    - The internalboard can be set to 'EXG', 'ECG' or 'EMG'.
- setenabledsensors
    - The following ExG sensors can be set: EMG, ECG, EXG1, EXG2, EMG 16BIT, ECG 16BIT, EXG1 16BIT, EXG2 16BIT. If EMG, ECG, EMG 16BIT or ECG 16BIT is set, the default configuration parameters for ECG or EMG are automatically loaded, as appropriate; see setdefaultemgparameters, setdefaultecgparameters.
    - The ExG data output format can be either 16-bit or 24-bit. For 16-bit format '16BIT' needs to be appended to the name of the sensor, as shown above.
- setexgtestsignalparameters(chipIdentifier)
    - Sets the ExG configuration for the testsignal for SENSOR_EXG1 or SENSOR_EXG2 depending on the value of the chipIdentifier,which can have a value of either 1 or 2. If the testsignal for a sensor is enabled, the testsignal is fed to both channels of that sensor.
- setdefaultemgparameters
    - When sensor 'EMG' or 'EMG 16BIT' is set (shimmer.setenabledsensors('EMG',1) e.g.), the default configuration parameters for EMG are loaded automatically.
    - By default, the EMG data rate is set as close to the sampling rate of the Shimmer as possible and always higher than this sampling rate.
    - For details on the EMG parameters, refer to the *ExG User Guide for EMG.*

- setdefaultecgparameters
    - When sensor 'ECG' or 'ECG 16BIT' is set (shimmer.setenabledsensors('ECG',1) e.g.), the default configuration parameters for ECG are loaded automatically.
    - By default, the ECG data rate is set as close to the sampling rate of the Shimmer as possible and always higher than this sampling rate.
    - For details on the ECG parameters, refer to the *ExG User Guide for ECG.*
- setexgconfiguration
    - Sets the ExG configuration parameters. Refer to *ExG User Guide for ECG / ExG User Guide for EMG* for detailed information.
- setexggain
    - Sets the ExG gain. Use the MATLAB™ help function to check the valid gain settings.
- setexgrate
    - Sets the ExG rate. Use the MATLAB™ help function to check the valid rate settings.

For more information on each command regarding the ExG functionality in the *Shimmer MATLAB™ Instrument Driver*, either use the MATLAB™ help function or directly refer to the *ShimmerHandleClass.m* file.

Users may also refer to the *ExG User Guide for EMG* and *ExG User Guide for ECG* for more information about the use of the *ExG Expansion Board* with *Shimmer3.*

10.2. Filtering

As described in the *ExG User Guide for ECG / ExG User Guide for EMG* filtering the ExG signals might be necessary in order to enhance the desired information by suppressing undesired frequency components.

In the *plotandwriteemgexample.m and plotandwriteecgexample.m* examples, filters are implemented. Please note that there are no functions for filtering in the *ShimmerHandleClass.* For filtering in the examples, functions of the class, *FilterClass.m*, are used.

In case of measuring EMG, we recommend the use of a high pass filter with a corner frequency of 5Hz. When measuring ECG in a diagnostic setting, we recommend setting the corner frequency of the high pass filter to 0.05Hz. For ECG monitoring applications, we recommend setting this corner frequency to 0.5Hz.

The band stop filter is used to suppress mains interference. If the mains frequency (power line frequency) in your area is 50Hz, set the corner frequencies of the band stop filter to 49Hz and 51Hz. For a mains frequency of 60Hz, set the corner frequencies to 59Hz and 61Hz. For details on the filter settings refer to *FilterClass.m*.

11. Appendices

11.1. More than 7 Shimmers via Bluetooth on a Single Computer

If the user wishes to stream data from more than 7 Shimmers simultaneously on a single computer using Bluetooth a solution is proposed here. The solution requires the use of 2 or more Bluetooth dongles (a dongle is required for every 7 Shimmers) with each Bluetooth dongle having its own Bluetooth Driver. This method has been verified in-house using a Toshiba driver and a Microsoft driver.

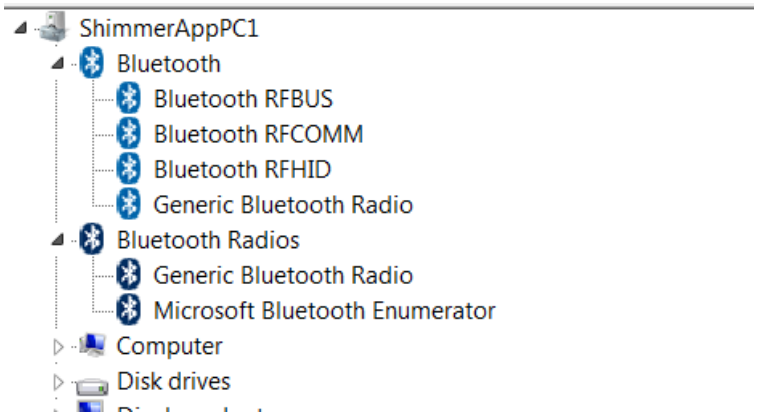


Figure 6: Multiple Bluetooth Driver

Figure 5 shows an example of what the device manager will show when two Bluetooth dongles with different stacks are used simultaneously. The top one uses a Bluetooth Toshiba driver while the bottom one uses a Microsoft driver.

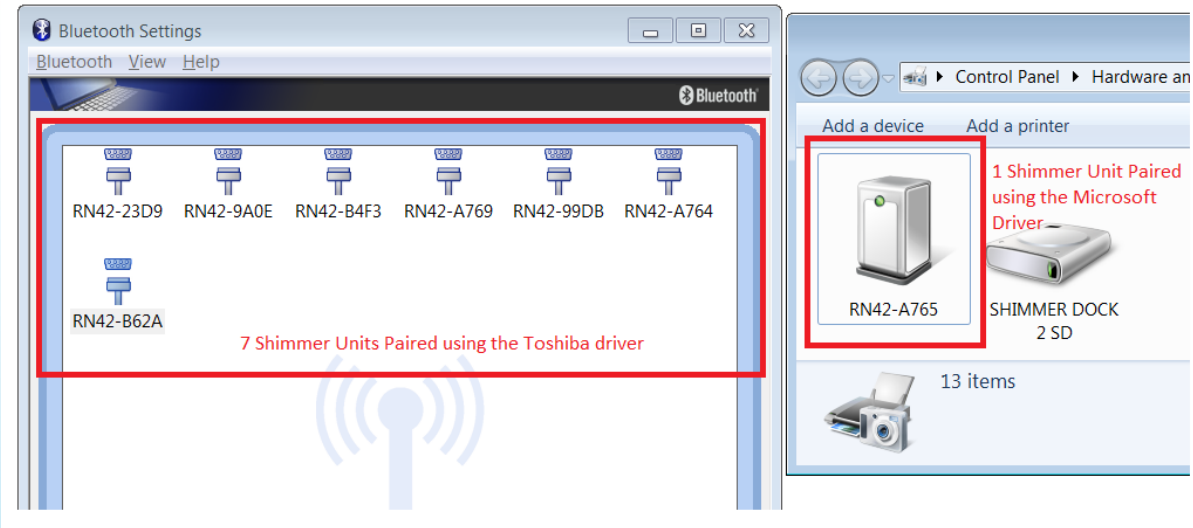


Figure 7: Pairing with Multiple Bluetooth Dongles

When using more than 7 Shimmer units ensure that a maximum of 7 are paired and used with one driver and the others are paired and used with the other driver. To avoid confusion it is advised that each device is paired with only one driver as demonstrated in Figure 7.

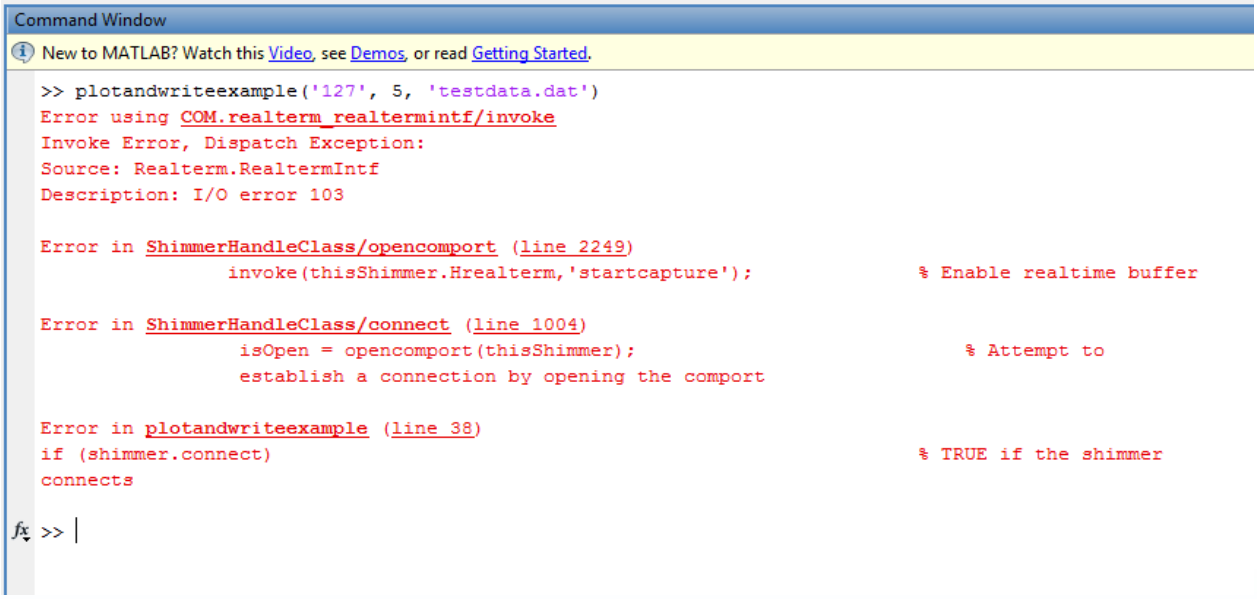
11.2. Known Bugs

Bug 0.1.1 - There is a known bug in the Shimmer Firmware Boilerplate v0.1. The bug occurs occasionally and **only when the magnetometer is enabled**. The Shimmer will connect normally and stream data normally but when the Shimmer receives the command to stop streaming the Shimmer will enter an unknown state and will require a reset. To resolve this issue, users should upgrade their firmware to *BtStream v0.2* or later.

11.3. Troubleshoot

I/O Error 103

Problem: Every time I run example functions I receive an I/O error 103 as illustrated below.



Solution: Ensure that your current working directory contains a folder called realtermBuffer and that you have permission to write to that folder.

Shimmer International Offices:
Europe – Dublin, Ireland.
USA – Boston, MA.
Asia – Kuala Lumpur, Malaysia.

Web: www.ShimmerSensing.com
Email: info@ShimmerSensing.com


www.Shimmersensing.com

/ShimmerResearch

@ShimmerSensing

/company/Shimmer

/ShimmerSensing

/ShimmerResearch