GS316USB

USB G-Sensor Dongle

DATASHEET

June 2014

140013DTS01-00.05.docx

sysacom

www.sysacom.com

Features

- 3-axis accelerometer (X,Y,Z)
- 4 Acceleration Range (±2g, ±4g, ±8g and ±16g)
- Configurable sampling rate (10 to 200 sps)
- Configurable alarm threshold for each axis
- USB 2.0 compatible
- Lead-Free / ROHS compliant

Applications

- Motion activated functions
- Free-fall detection
- Impact recognition and logging
- Vibration monitoring and compensation

Description

The GS316USB is a USB dongle containing a 3-axis accelerometer. It measure acceleration in 3-axis (X, Y, Z) and has 4 acceleration ranges (±2g, ±4g, ±8g and ±16g). The sampling rate can be set from 10SPS to 200SPS. Also, a configurable threshold can be set to generate an alarm if the acceleration is greater than said threshold (can be enable for one or multiple axes).

This USB dongle is based on the LIS3DH accelerometer from STMicroelectronics. Specific information related to the accelerometer can be found at:

http://www.st.com/web/catalog/sense_power/FM89/SC444/PF250725?s_searchtype=partnumber

The GS316USB is easy to install and simple to use with direct access to the accelerometer register using a protocol commands for the USB port. For simplicity it is supplied with C# graphical user interface and source code is included for an easier software development. The USB commands are designed to mimic the interface provided by the accelerometer itself. So an application code using the provided source file would be easily usable in an embedded application.

Contents

Revision History

Revision Number	Revision Date	Description of Changes
00.00	10/23/2013	Preliminary release
00.03	06/03/2014	Update GS device command
00.04	06/13/2014	Minor text correction & add picture of product
00.05	06/16/2014	Add description of pc interface

1. Electrical Characteristics

Specifications*	GS316USB	Unit
Performance		
USB Compliance	2.0	
Support Data Rate	1.5	Mbps
Environment		
Operating Temp Range	-40 to 85°C	°C
Electrical		
Supply Voltage (from USB)	5.0	V
Supply Current (from USB)	25	mA
MTBF	TBD	Hours
Physical		
Weight	5.62	Grams

* All specifications are preliminary estimate.

2. Principle of operation

The GS316USB unit uses the USB protocol to receive commands sent from a PC. These commands are used to configure the accelerometer (sampling rate, acceleration range and alarm threshold) and access data from the accelerometer.

Two acquisition modes can be used with the GS316USB, the single read mode and the sampling read mode. In the single read mode, the PC sends a command to the device and it respond by sending a single set of acceleration data. In sampling read mode, the PC sends a command to start sampling mode and the devices responds by sending sets of acceleration data at fixed interval until the PC sends the command to stop the sampling read mode.

3. Using the software

The GS316USB is provided with a simple PC application that shows how to use many of its features. This section is focused on helping you to use the source code for communicating with the GS316USB.

3.1 Starting the Application

- Double-click on the GSFamilyInterface desktop icon.

OR

- Choose GSFamilyInterface from the Start menu → Programs → Sysacom.

OR

- Choose Run from the Start menu. In the Run window, type the installation directory and type \ GSFamilyInterface.exe. (For example: C:\Program Files\Sysacom\GS Family Interface\ GSFamilyInterface.exe)

Then select the GS316USB from the dropdown list.

3.2 Main Window

3.2.1 Description

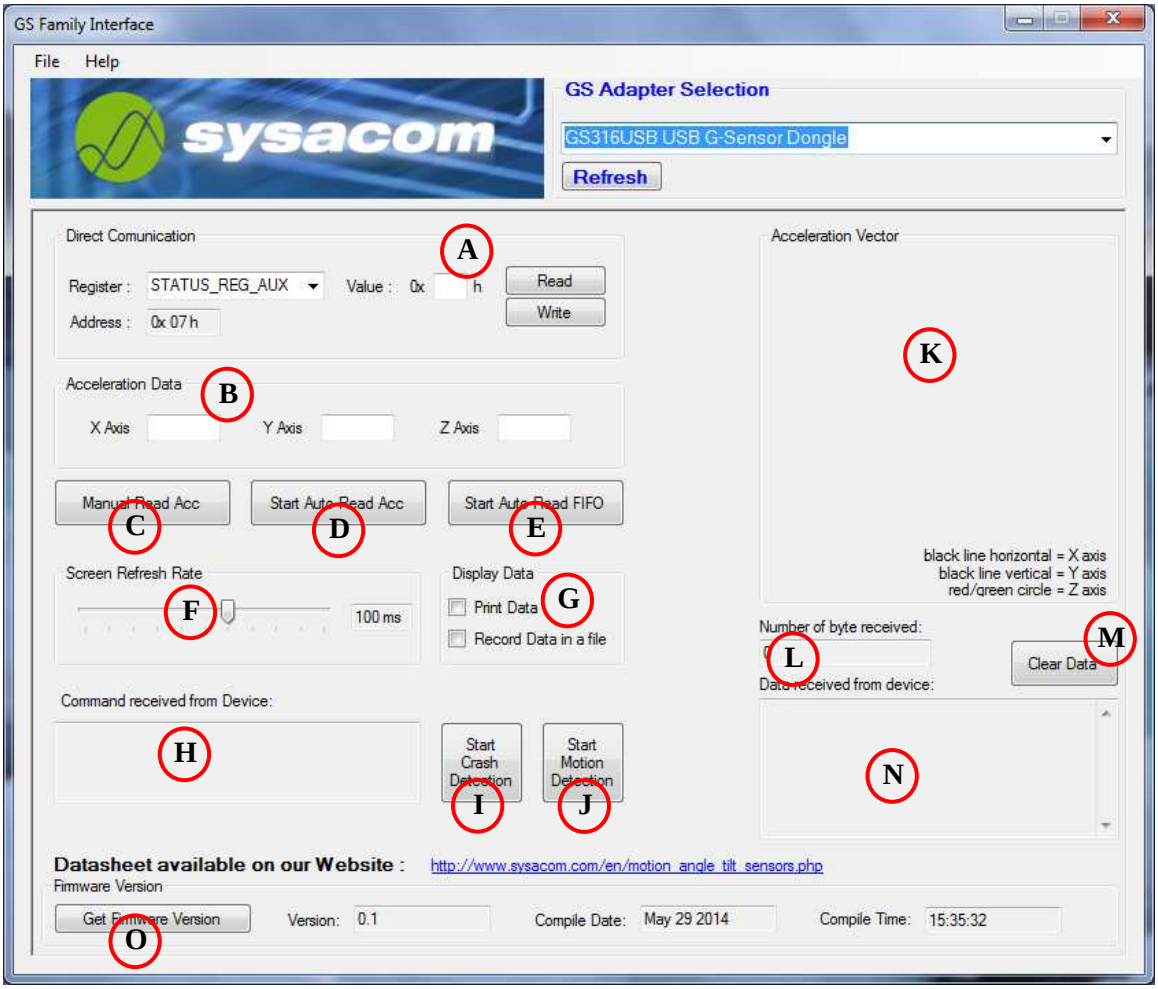


Figure 1: GS316USB Window.

Reference Designator	Description
A	Direct Communication
B	Acceleration Value (in g)
C	Read Acceleration Value Once
D	Auto Read Acceleration
E	Read Acceleration FIFO
F	Screen Refresh Rate
G	Display Data
H	Command Received from Device
I	Start Crash Detection
J	Start Motion Detection
K	Acceleration Vector
L	Number of byte Received
M	Clear Data
N	Data Received from Device
O	Get Firmware Version

Table 1: GS316USB Window Description.

3.2.2 Setting Parameters and Functionalities

3.2.2.1 Direct Communication

- Read accelerometer register
- Write accelerometer register

Refer to LIS3DH accelerometer datasheet for register description:
http://www.st.com/web/en/resource/technical/document/datasheet/CD00274221.pdf

3.2.2.2 Auto Read Acceleration

Read acceleration at the rate set in "Screen Refresh Rate"

3.2.2.3 Read Acceleration FIFO

Read acceleration FIFO with the following setting:

- CTRL_REG1 = 0x17 (Set ODR at 1Hz)
- CTRL_REG3 = 0x04 (FIFO Watermark interrupt generation is routed to INT1 pin)
- CTRL_REG4 = 0x80 (FS = ±2g normal mode with BDU bit enabled)
- CTRL_REG5 = 0x40 (FIFO_EN bit enabled)
- FIFO_CTRL_REG = 0x5F (FIFO mode, trigger event when FIFO full)

Data are displayed on the interface when the FIFO buffer is full.

3.2.2.4 Display Data

- Check “Print Data” to display the acceleration raw data in the interface
- Check “Record Data in a file” to record in a csv file the acceleration value (in g). The csv file will be located in C:\Program Files\Sysacom\GS Family Interface.

3.2.2.5 Start Crash Detection

Start crash detection with the following setting:

- CTRL_REG1 = 0x67 (Set ODR at 200Hz)
- CTRL_REG3 = 0x40 (AOI1 interrupt generation is routed to INT1 pin)
- CTRL_REG4 = 0x90 (FS = ±4g normal mode with BDU bit enabled)
- CTRL_REG5 = 0x0C (Interrupt on INT1 pin is latched with D4D_INT1 bit enabled)
- INT1_THS = 0x80 (Threshold = 128LSBs * 15.625mg/LSB = 2g)
- INT1_DURATION = 0x02 (Duration = 2LSBs * (1/200Hz) = 10mS)
- INT1_CFG = 0x7F (6D movement detection)

When the accelerometer detect 2 consecutive acceleration value higher than 2g then a big red circle appear on the interface.

3.2.2.6 Start Motion Detection

Start motion detection with the following setting:

- CTRL_REG1 = 0x47 (Set ODR at 50Hz)
- CTRL_REG3 = 0x40 (AOI1 interrupt generation is routed to INT1 pin)
- CTRL_REG4 = 0x80 (FS = ±2g normal mode with BDU bit enabled)
- CTRL_REG5 = 0x0C (Interrupt on INT1 pin is latched with D4D_INT1 bit enabled)
- INT1_THS = 0x20 (Threshold = 32LSBs * 15.625mg/LSB = 500mg)
- INT1_DURATION = 0x00 (interrupt will be generated immediately)
- INT1_CFG = 0x7F (6D movement detection)

When the accelerometer detect a movement then a big red circle appear on the interface.

4. Software Function

The GS316USB module use the generic WinUSB driver (Winusb.sys) and its user-mode component (Winusb.dll) provided by Microsoft to communicate.

The functions supported by the source code are:

```
public ErrorCode InitComm(String reqDevID)
public void CloseComm(void)
public byte registerRead(byte reg, out byte value)
public byte registerWrite(byte reg, byte value)
public byte accelerationRead(byte[] data)
public byte fifoRead(byte[] data)
public byte interruptEnable(void)
public byte interruptDisable(void)
```

4.1 InitComm() detail

Prototype : DWORD InitComm(void);

This function is used to initialize the communication with the GS316USB. First of all, this function creates the link between the driver and the DLL. With this link the handles (IN and OUT) are initialized to communicate with the driver. When both handles is initialized correctly, the thread to read the GS316USB in continuous is created and started.

Error Returned	VALUE	Description
ERR_SUCCESS	0x00	The communication with the GS316USB is done
ERR_INVALID_HANDLE	0x04	The handles were not initialized correctly. The communication is not done.
System error code	xxxx	The error code returned after an error with the creation of the thread. The error system code is defined in the WinError.h file or to this Web address : http://msdn.microsoft.com/en-us/library/ms681381(VS.85).aspx

Table 2: InitComm error Message

Example usage:

```
if( InitComm() == ERR_SUCCESS )
       /* Then the communication is done… */
```

4.2 CloseComm() detail

Prototype: DWORD CloseComm(void);

This function closes the communication between the application program and the GS316USB driver. It kills the thread and closes both handles (IN and OUT).

Error Returned	VALUE	Description
ERR_SUCCESS	0x00	The communication with the GS316USB is correctly closed
ERR_INVALID_FUNCTION	0x05	A problem is arrived during the closure and the communication is not correctly closed.

Table 3: CloseComm error Message

Example usage:

```
if( CloseComm() == ERR_SUCCESS )
      /* The communication is closed, thus the application can be close… */
```

4.3 registerRead() detail

Prototype: `byte registerRead(byte reg, out byte value)`

This function is used to read a specific register from LIS3DH accelerometer.

Parameter	Description
reg	This contains the accelerometer register address to read.
value	This return the value in the selected register.

Table 4: registerRead parameter

Error Returned	VALUE	Description
ERR_SUCCESS	0x00	The accelerometer register was successfully read.
ERR_NOT_COMPLET	0x01	The device was unable to read register.

Table 5: registerRead error Message

4.4 registerWrite() detail

Prototype: `byte registerWrite(byte reg, byte value)`

This function is used to write a specific register from LIS3DH accelerometer.

Parameter	Description
reg	This contains the accelerometer register address to write
value	This contains the value to write at the selected register

Table 6: registerWrite parameter

Error Returned	VALUE	Description
ERR_SUCCESS	0x00	The accelerometer register was successfully write.
ERR_NOT_COMPLET	0x01	The device was unable to write register.

Table 7: registerWrite error Message

4.5 accelerationRead() detail

Prototype: `byte accelerationRead(byte[] data)`

This function is used to get the acceleration value for all axes.

Parameter	Description
data	This points to the 6 bytes returned by the device: Bytes 0-1: X axis acceleration value. Bytes 2-3: Y axis acceleration value. Bytes 4-5: Z axis acceleration value.

Table 8: accelerationRead parameter

Error Returned	VALUE	Description
ERR_SUCCESS	0x00	There was no error when getting the acceleration data.

Table 9: accelerationRead error Message

4.6 fifoRead() detail

Prototype: `byte fifoRead(byte[] data)`

This function is used to get the X/Y/Z axis acceleration value from the 32 fifo registers.

Parameter	Description
data	This points to the 32 x 6 bytes acceleration data returned by the device.

Table 10: fifoRead parameter

Error Returned	VALUE	Description
ERR_SUCCESS	0x00	There was no error when getting the acceleration data.

Table 11: fifoRead error Message

4.7 interruptEnable() detail

Prototype: `byte interruptEnable(void)`

This function is used to enable the device interrupt detection. The device will then send the value 0xFA through the USB port when an interrupt is detected

Error Returned	VALUE	Description
ERR_SUCCESS	0x00	The device interrupt detection was successfully enabled.

Table 12: interruptEnable error Message

4.8 interruptDisable() detail

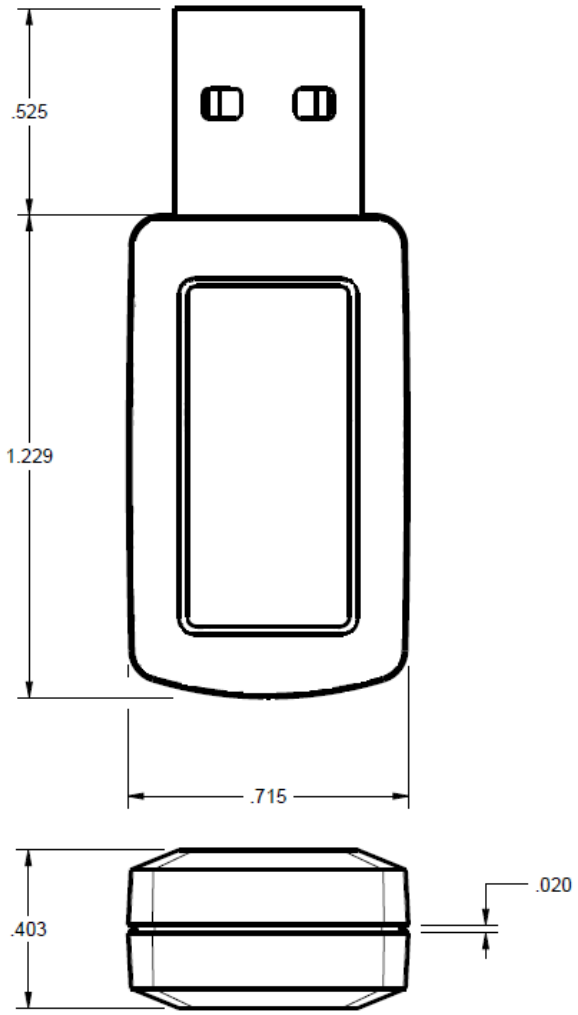
Prototype: `byte interruptDisable(void)`

This function is used to stop the interrupt detection. The device will stop to send interrupt detection value through the USB port.

Error Returned	VALUE	Description
ERR_SUCCESS	0x00	The device interrupt detection was successfully disabled.

Table 13: interruptDisable error Message

5. Mechanical



6. Ordering Information

Part Number	Temperature Range
GS316USB	-40°C ~ 85°C

7. Warranty

Sysacom R&D plus Inc warrants the GS316USB to be free from malfunctions and defects in both materials and workmanship for one year from the date of purchase.

If the equipment does not function properly during the warranty period due to defects in either materials or workmanship, Sysacom R&D plus Inc will, at its option, either repair or replace the equipment without charge, subject to limitations stated herein. Such repair service will include all labour, as well as any necessary adjustments and / or replacement parts.

LIMITATIONS

The warranty becomes null and void if you fail to pack your GS316USB in a manner consistent with the original product packaging and damage occurs during shipment.

Sysacom R&D plus Inc makes no other warranties, express, implied, or of merchantability or fitness for a particular purpose for this equipment or software. Repair or replacements without charge are Sysacom R&D plus Inc only obligation under this warranty. Sysacom R&D plus Inc will not be responsible for any special, consequential or incidental damages resulting from the purchase, use, or improper functioning of this equipment regardless of the cause.

Depending on your geographical location, some limitations may not apply.

Sysacom R&D plus inc.
275A Pierre-Le Gardeur Blvd
Repentigny, Quebec
Canada, J5Z 3A7
www.sysacom.com
(450)585-6396