

X16-4 USB Accelerometer Data Logger

User Manual

Document Revision:
Firmware Version:
Date:

Rev B
1110
August 16, 2017



Table of Contents

1	Introduction.....	1
1.1	About This Manual.....	1
1.2	Document Conventions.....	1
1.3	Appendix.....	1
1.4	Product Summary.....	2
1.5	Feature List.....	2
1.6	Items Included with X16-4.....	3
1.6.1	Single Unit Purchase.....	3
1.6.2	5 Unit Kit.....	3
1.7	Component Names.....	4
1.8	Quick Start Guide.....	5
2	Operation.....	7
2.1	USB Interface.....	7
2.2	Memory Card.....	7
2.3	Battery.....	8
2.4	Setting The RTC.....	9
2.5	Status Indicators.....	10
2.6	System Configuration Options.....	11
2.6.1	deadband.....	12
2.6.2	deadbandtimeout.....	12
2.6.3	dwel.....	12
2.6.4	microres.....	13
2.6.5	rebootondisconnect.....	13
2.6.6	samplesperfile.....	13
2.6.7	samplerate.....	14
2.6.8	starttime and stoptime.....	14
2.6.9	stoponvusb.....	14
2.6.10	statusindicators.....	14
2.7	Example Configuration Files.....	14
2.7.1	Example A.....	14
2.7.2	Example B.....	15
2.7.3	Example C.....	15
3	Data Interpretation.....	16
3.1	Data Files.....	16
3.2	Data Format.....	16
3.3	Data Conversion.....	17
3.3.1	Time Stamps.....	17
3.3.2	Acceleration.....	18
4	System Details.....	18
4.1	Sensor.....	18
4.1.1	Sensor Special Features.....	19
4.2	Operating and Storage Conditions	20
4.3	Dimensions.....	20
5	Troubleshooting.....	21

6	Appendix.....	23
6.1	What is an Accelerometer.....	23
6.2	Using “R” to Analyze Data.....	25
6.2.1	What is “R”.....	25
6.2.2	Introduction to R Commands.....	26
6.2.3	Online Resources for R.....	27
6.2.4	Example Scripts in R.....	28

Index of Figures

Figure 1:	X16-4 Data Logger.....	2
Figure 2:	X16-4 and Accessories.....	3
Figure 3:	5 Unit kit of Loggers.....	3
Figure 4:	X16-4 Data Logger Components.....	4
Figure 5:	Exploded View of the X16-4.....	4
Figure 6:	Connecting to PC.....	5
Figure 7:	Editing the Config.txt File.....	5
Figure 8:	Starting the X16-4.....	6
Figure 9:	Sensor Orientation.....	7
Figure 10:	Battery Pack Connectors.....	8
Figure 11:	Expected Battery Life.....	9
Figure 12:	Example Time Entry in time.txt File.....	10
Figure 13:	LED Status Indicators.....	11
Figure 14:	Graphical Illustration of the Deadband Feature.....	12
Figure 15:	Graphical Illustration of the Dwell Feature.....	13
Figure 16:	Configuration File Example A.....	15
Figure 17:	Configuration File Example B.....	15
Figure 18:	Configuration File Example C.....	15
Figure 19:	Example Data File.....	16
Figure 20:	Time Stamp Conversion Method.....	17
Figure 21:	800Hz Sample Configuration.....	20
Figure 22:	Enclosure Dimensions.....	20
Figure 23:	Spring-mass Accelerometer.....	23
Figure 24:	Simplified MEMS Accelerometer Design (L) and Actual MEMS Accelerometer (R).....	24
Figure 25:	R Command Line Interface	25
Figure 26:	RStudio Interface	26

Index of Tables

Table 1:	Configuration File Tags and Descriptions.....	11
Table 2:	Data File Header Tags.....	17
Table 3:	Example Data Conversion.....	18
Table 4:	Accelerometer Sensor Characteristics.....	19
Table 5:	Operating and Storage Conditions.....	20

1 Introduction

1.1 About This Manual

Thank you for purchasing the X16-4 accelerometer data logger. Gulf Coast Data Concepts spent considerable efforts developing an easy to use data logger for the scientific researcher, student, or hobbyist. Please read this manual to understand the operation and capabilities of the X16-4. If the logger fails to operate as expected, please refer to the troubleshooting guide (page 21).

1.2 Document Conventions

The quick start guide in section 1.8 provides a basic summary of operation to begin using the X16-4 data logger. This user manual continues into further details of configurations and capabilities starting in section 2. Each section also presents relevant tips and warnings to help the user.



This icon indicates a helpful tip that may enhance the performance of the logger or aide in the application of the logger.



This icon indicates a warning, restriction, or limitation that the user should be aware of regarding the logger operation.

1.3 Appendix

The appendices to this document include several educational discussions regarding accelerometers (section 6.1) as well as software and analysis procedures (section 6.2). These short discussions will help new users learn about the X16-4 and how to use the data.

1.4 Product Summary

The X16-4 is a low cost and compact self-recording accelerometer data logger. Data from the digital 3-axis accelerometer sensor is time stamped using a real time clock and stored to a microSD card in simple text format. When connected via the USB to a personal computer, the X16-4 appears as a standard mass storage device containing the comma delimited data files and the user setup file. The X16-4 includes an internal 500mAh lithium-polymer rechargeable battery, which will recharge using USB power.

1.5 Feature List

- 3-axis $\pm 16g$ accelerometer
- 16-bit resolution
- User selectable sample rate of 12, 25, 50, 100, 200, 400 Hertz
- Finite Impulse Response filter
- Accurate time stamped data using Real Time Clock (RTC)
- Convenient on/off button
- Data recorded to a removable microSD card (8GB included)
- Easily readable comma separated text data files
- Data transfer compatible with Windows or Linux via Universal Serial Bus (USB) interface (no special software required)
- Operates from internal lithium-polymer rechargeable battery
- Weight 1.3oz (48g)
- Size 1x1x4.1 inch (26x26x104 mm)



Figure 1: X16-4 Data Logger

1.6 Items Included with X16-4

1.6.1 Single Unit Purchase

The X16-4 is packaged with the logger, a USB extender cable, and a screwdriver.



Figure 2: X16-4 and Accessories

1.6.2 5 Unit Kit

A kit includes 5 X16-4 loggers, a USB extender cable, and a screwdriver.



Figure 3: 5 Unit kit of Loggers

1.7 Component Names

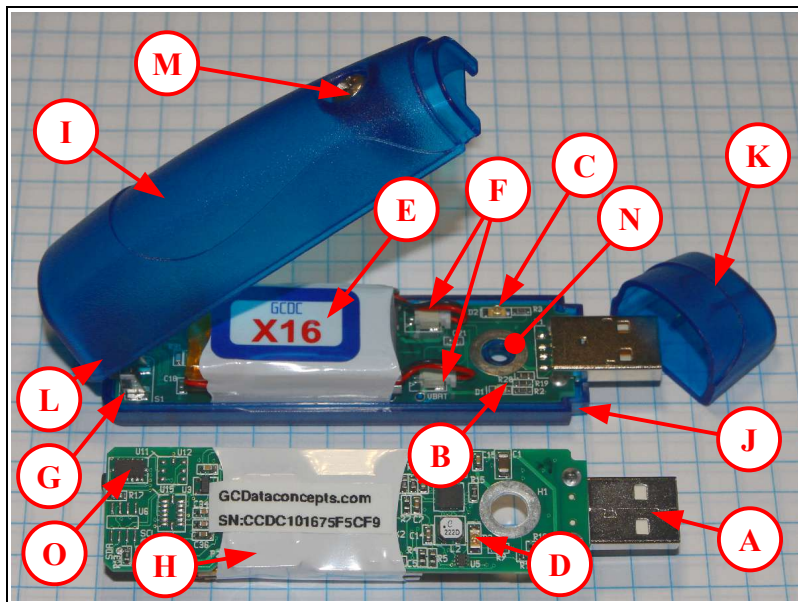


Figure 4: X16-4 Data Logger Components

A	Type-A USB connector	I	Enclosure top
B	Blue LED status indicator	J	Enclosure bottom
C	Red LED data indicator	K	Enclosure cap
D	Red LED charge indicator	L	Enclosure hinge
E	500mAh Battery Pack	M	#6-32 3/4" screw
F	Battery Connectors	N	#6 nut
G	On/Off button	O	ADXL345 sensor
H	MicroSD card (under wrap)		

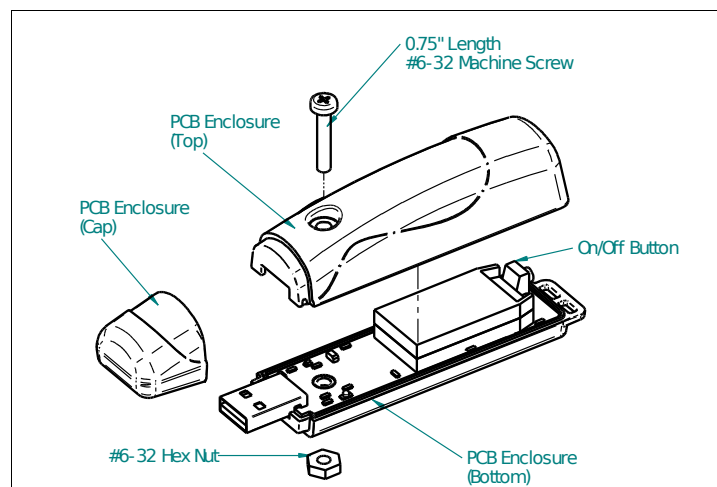


Figure 5: Exploded View of the X16-4

1.8 Quick Start Guide

The X16-4 is a simple, economical solution to capture continuous motion data and quickly deliver the information for analysis. The following instructions outline the steps to begin using the X16-4. Configuration settings and mounting methods will depend on the particular application.

Step 1: Plug the X16-4 into a computer and allow the computer operating system to register the device as a Mass Storage Device. Notice that the logger will mount with a drive label using the last digits of the serial number. A red LED located on the bottom side will indicate the battery is charging. The LED will turn off when the battery is fully charged, which takes about 2 hours.

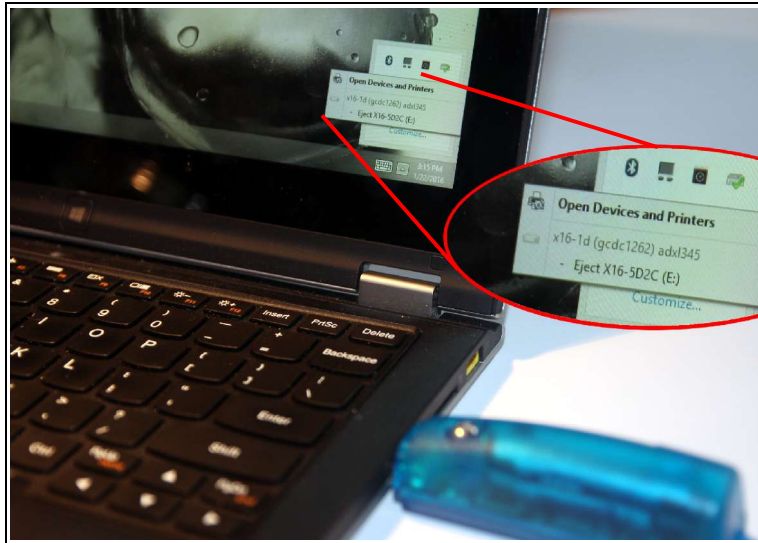


Figure 6: Connecting to PC

Step 2: Configure the X16-4 by editing the appropriate tags in the config.txt file using a simple text editor. In Windows, do not use Notepad as the editor does not terminate new lines properly. GDCDC recommends Windows Wordpad or Notepad++ to edit the config.txt file. Refer to section 2.6 for a complete list of configuration options.

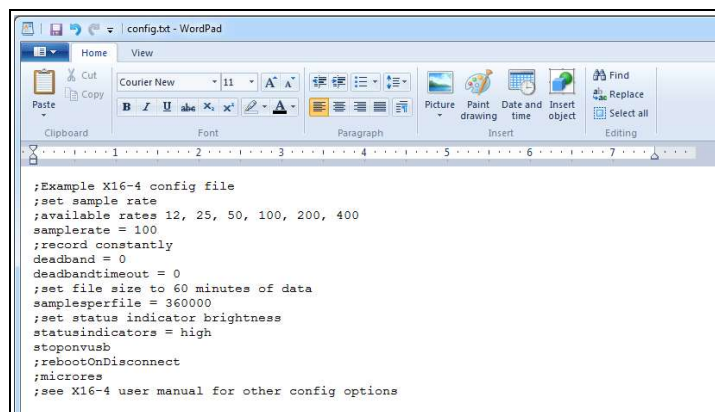


Figure 7: Editing the Config.txt File

- Step 3: If necessary, initialize the RTC clock by creating a time.txt file (see section 2.4). Once the time.txt file is saved, immediately unplug the logger to start the initialization process. The logger will load the time.txt file, initialize the clock, and delete the time.txt file. Initializing the RTC ensures the data files include the correct year, month, and day and that the data samples can be correlated to a specific date and time.
- Step 4: After removing from the USB port, attach the X16-4 logger to the target object. The logger is small and light weight so double-sided tape, Velcro, or a spot of cyanoacrylate glue are sufficient methods of attachment.



The X16-4 is small and light weight so attachment methods do not need to be substantial. Double-sided tape, a spot of cyanoacrylate glue (contact cement), zip-ties, magnetic base, or adhesive putty are example methods of attachment. These methods do not cause adverse signal attenuation considering the relatively low frequency bandwidth of the X16-4 logger. Command Poster Adhesive strips by 3M offer excellent temporary attachment of the logger to most surfaces.

- Step 5: Press the on/off button located at the rear of the enclosure to initiate data recording, (see Figure 8). Logging will start about 3-5 seconds after pressing the button. The red LED will blink as the configuration file is accessed. If the time.txt file is present, the RTC is initialized with the time written in the file. Then, the blue LED will begin to blink at a 1 second interval indicating the system is operating. The red LED will blink periodically as data is written to the microSD card.



Figure 8: Starting the X16-4

- Step 6: To stop recording, press and hold the button for about 3 seconds. The red and blue LEDs will begin to blink rapidly for 2 seconds and then turn off. Release the button and the X16-4 turns off. Pressing the button again restarts the logger and data is recorded to a new file.
- Step 7: Plug the logger into a PC and allow the logger to mount as a USB drive. The data files will appear in the “GCDC” directory.
- Step 8: The data recorded to the files must be converted to determine acceleration in “g” units. Divide the Ax, Ay, Az columns by 2048 to determine g units. See section 3.3 for a complete discussion of data conversion.

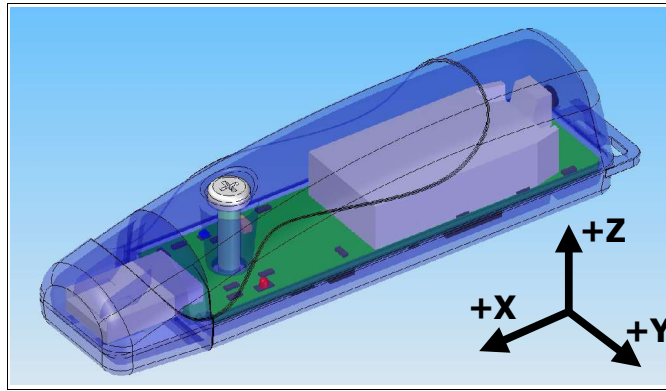


Figure 9: Sensor Orientation

2 Operation

2.1 USB Interface

The X16-4 connects to a PC using a standard Type-A USB connector and supports the USB mass storage device interface for file access and file transfers. Nearly all computer operating systems recognize the X16-4 as a typical USB external memory drive. Therefore, the X16-4 will allow file transfers to the microSD card like a common USB flash drive. When connected to a PC, the X16-4 deactivates logging and operates only as a USB interface to the microSD card. Note that some tablet operating systems block access to USB mass storage devices and will not recognize the X16-4.

2.2 Memory Card

The X16-4 stores data to a removable microSD flash memory card located beneath the white plastic wrap. The included 8GB card is sufficient for most applications so the card does not need to be removed or upgraded. However, the plastic wrap can be cut away to allow access to the card. The logger uses FAT32 and is compatible with microSD and microSDHC type cards.

The logger needs only the config.txt file to operate. The X16-4 will use default configuration settings if the config.txt is not present. The “config.txt” and “time.txt” files must occur in the root directory (see section 2.6 and section 2.4). The X16-4 will create a folder called “GCDC”, if not already present, to place the data files.



The white plastic wrap is used to aid the assembly process at GCDC but it can be removed to allow access to the microSD card. The logger is USB 2.0 compliant but is limited to about 700 KB/sec. Removing the microSD card and using a standard USB card reader will permit faster transfer of data files.



Interrupting the power to the logger can result in corruption of the microSD card. For example, removing the logger from the USB port during file transfers to the PC or removing the battery during logging activity. Reformat the card if it becomes corrupted (FAT32 file structure). If data transfers to/from the card become slow, consider formatting the card using “SD Card Formatter” software provided by the SD Association (www.sdcard.org).

2.3 Battery

The X16-4 is powered by a internal 500mAh lithium-polymer rechargeable battery pack. The internal battery management system recharges the battery when the X16-4 is plugged into a USB port or attached to a USB 5v power adapter. The red charge indicator LED turns on (see Section 2.5) when the battery is charging and off when the battery reaches full charge. The battery is not used when the system is connected to a computer USB port. Separate connectors are used for the two individual 250mAh batteries, as noted in Figure 10. Disconnecting both connectors will deactivate the logger and reset the system (data on the microSD card is not be affected).

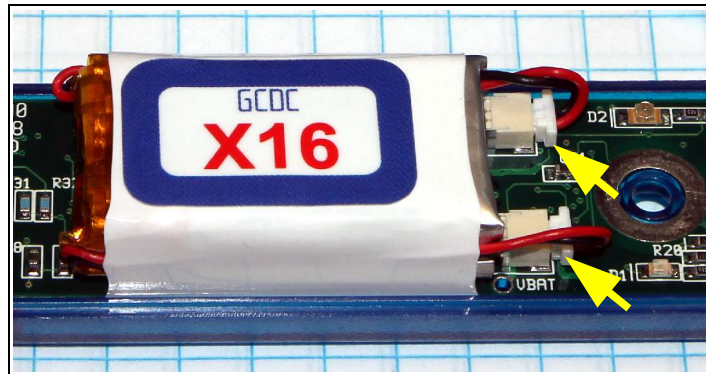


Figure 10: Battery Pack Connectors

The battery provides approximately 16 hours of operation sampling at 400 Hz with the deadband set to zero (maximum data recording capability). Operating time increases with lower sample rates. Figure 11 illustrates the expected continuous logging time versus sample rate.

The RTC continues to operate from the battery when the device is “off”. The RTC should be reinitialized if the battery is removed or completely depleted (see Section 2.4).



Some X16-4 data loggers are assembled without battery connectors and the battery is hardwired to the PCB. See the Troubleshooting guide at the end of this document for further information regarding the hardware differences.



The data logger may draw up to 250mA from the USB supply to recharge the battery. Plugging multiple data loggers into a USB hub can exceed the power capacity of the hub. This can cause “brown-outs” of the logger and possibly damage the microSD card.



The logger is always “on” maintaining the real time clock and will eventually discharge the battery completely after several months. The completely discharged battery will take additional time to recharge. Keep in a cool (20°C/ 68°F) dry environment to avoid damage of the battery pack.



A 5v supply via the USB connector provides extended operation of the device independent of the internal battery. Add “rebootondisconnect” to the config.txt file so the logger will automatically switch to internal battery when the 5v is removed. Common 5v power adapters used for consumer electronics can provide the required 5v supply but most USB battery packs are not compatible with the X16-4. The logger consumes very little power and “smart” USB battery packs will turn off when the power consumption is too low.

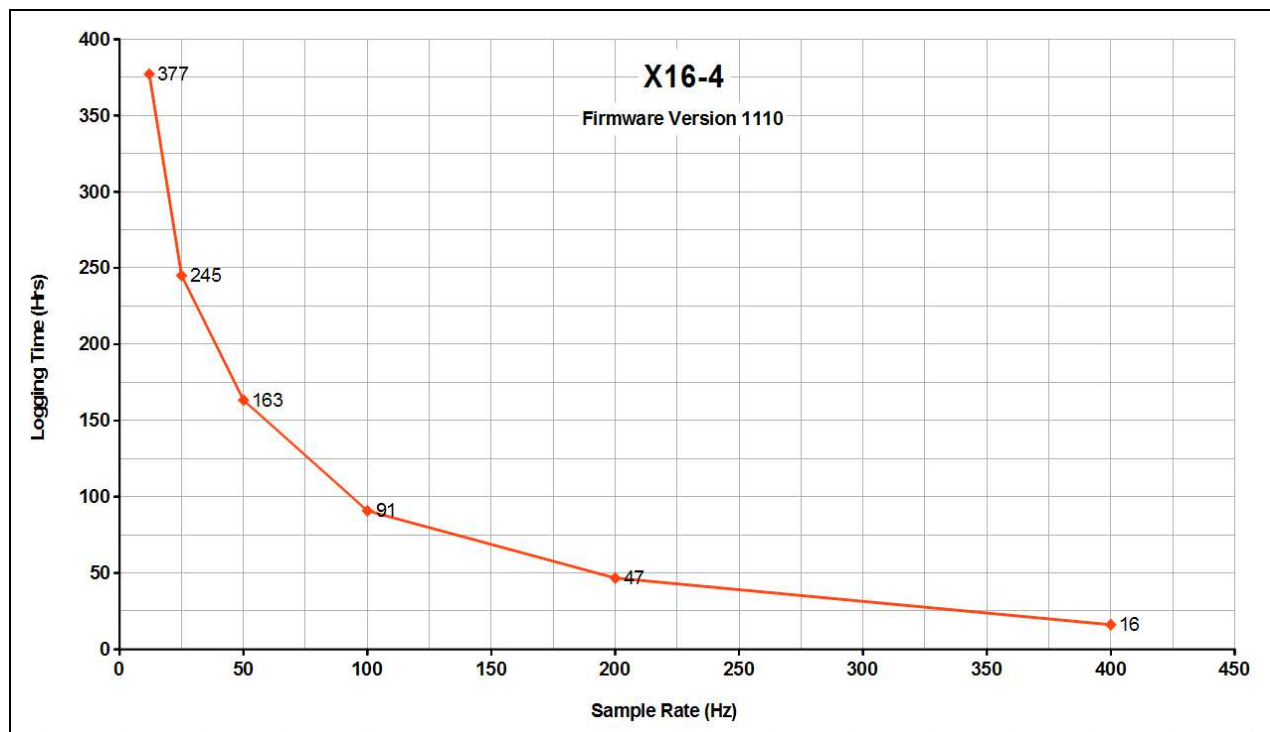


Figure 11: Expected Battery Life

2.4 Setting The RTC

A real time clock (RTC) integrated into the X16-4 determines the time for each line of data recorded. The RTC is initialized using a user-created text file named “time.txt” that is loaded by the logger upon booting. The time file method of setting the RTC does not require special communication drivers, so it can be implemented using a simple text editor. Direct initialization of the RTC is possible but requires specific device drivers and software from Gulf Coast Data Concepts.

Initializing the RTC with a time.txt file is accomplished as follows:

- Step 1: Use Wordpad, or an equivalent text editor, to create a simple text file called “time.txt”.
- Step 2: Enter on the first line the current date and time as “yyyy-MM-dd HH:mm:ss” in 24-hr format. Figure 12 provides an example time.txt file that will initialize the RTC to 2:26:30 pm June 16, 2014.
- Step 3: Save this file to the root directory of the microSD card (same location as the

config.txt file) and close the text editor.

Step 4: Remove the logger from the PC. The logger will automatically find the time.txt file and initialize the RTC with the time stored in the file. The file is deleted after initialization.

The RTC maintains ± 50 ppm accuracy (-40°C to $+85^{\circ}\text{C}$), which means that the accuracy may drift about 4 seconds every day. The RTC is powered by the battery at all times, even when the logger is “off”.

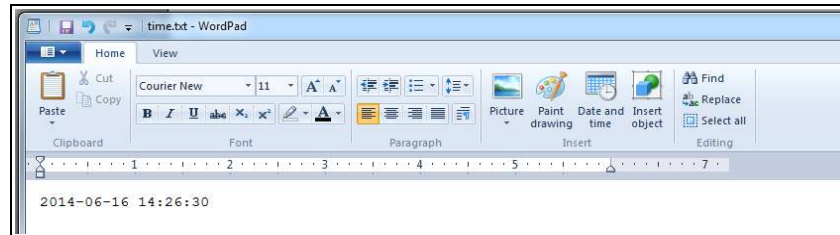


Figure 12: Example Time Entry in time.txt File



Initializing the RTC ensures that the start time and individual time stamps can be correlated to an absolute time – the year, month, day, hour, minute, second, and fractional second. An uninitialized RTC or reset of the RTC will lead to indeterminate start time recorded in the data file header.



After unplugging the logger from the USB port, the logger will load the config.txt file and time.txt file, if present. Therefore, there is a delay between when the time.txt was created and when the logger actually loads the time information. For most applications, this simple method of initializing the clock results in sufficient accuracy.



Initialization of the RTC is limited to ± 1 second. The RTC register that handles the fractional seconds counter is not accessible so the initialization process can not reset the seconds to an even value.

2.5 Status Indicators

System status is indicated by the two LEDs located near the USB connector. The blue LED indicates system operation and blinks once per second to indicate a properly operating system. The blue LED blinks when the X16-4 is recording data, in standby mode, or is connected to a computer via the USB port. The red LED blinks when data is written or read from the microSD memory card. In data logging mode, the period at which the red LED blinks depends on the sample rate and other configuration settings. The LEDs will flicker during user initiated shutdown. The “statusindicators” tag in the configuration file turns off or changes the brightness of the status indicators (see section 3.2.4.10). The red charge indicator LED is located on the reverse side of the circuit board and illuminates when charging is in process (see Figure 13). The charge indicator LED will turn off when the battery is fully charged. A fully discharged battery will charge in about 2 hours. A blinking charge indicator LED means there is a problem with the lithium-polymer battery pack.

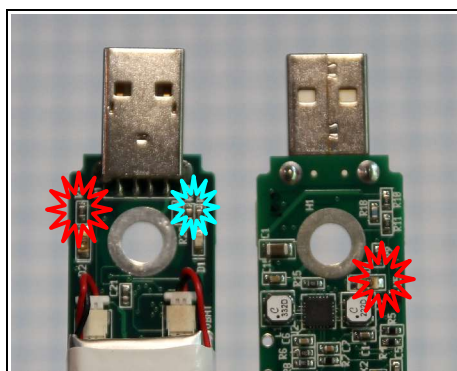


Figure 13: LED Status Indicators

2.6 System Configuration Options

The X16-4 is configured using a set of tags and settings stored in a text file named “config.txt”, which is located in the root directory of the microSD card. The system reads the configuration file at boot time. Table 1 lists the configuration file tags. Tags that require a setting must be followed by an equal sign (“=”) and an applicable tag setting. A line finishes with a newline character. Tags are not case sensitive. Tab and space characters are ignored. Lines starting with a semicolon (“;”) are treated as comments and ignored by the system. The system will use default settings if the config.txt file is not found.

Table 1: Configuration File Tags and Descriptions

Tag	Valid Settings	Default	Description
deadband	An integer between 0 and 16384	0	Sets the deadband to a range expressed in “counts”. A new sample is recorded if any sensor axis exceeds the previous recorded reading by the deadband value
deadbandtimeout	An integer between 0 and 16384	3	Specifies the period in seconds when a sample is recorded regardless of the deadband setting. This feature ensures periodic data is recorded during very long periods of inactivity.
dwel	An integer between 0 and 65535	1	The number of samples recorded after a deadband threshold triggered event
microres	-	Off	The presence of this tag sets the device to record time stamps with 0.1ms effective precision.
rebootondisconnect	-	Off	The presence of this tag causes the system to start recording after disconnect from a USB port.
samplesperfile	An integer greater than 0	90000	The number of lines of data per data file before a new file is created
samplerate	12, 25, 50, 100, 200, 400	100	Sets the rate at which data is collected and recorded to the microSD card.
starttime and stoptime	See section 2.6.8	-	Defines when to start and stop recording
stoponvusb	-	Off	Stops data logging if 5v USB power is present (see section 2.6.9)
statusindicators	“Normal”, “High”, “Off”	Normal	LED status indicators can be activated with normal brightness (Normal), activated with high brightness (High), or completely deactivated (Off).



Do not use the Windows Notepad editor because it does not terminate new lines properly. GCDC recommends Windows Wordpad or Notepad++ to edit the config.txt file.

2.6.1 deadband

“deadband” defines the minimum difference between recorded sensor readings. A new sample from the accelerometer sensor must exceed the previous recorded reading before the logger records the data. The deadband setting is expressed in "counts" units and is applied to the output of each axis. The deadband value can be set to an integer between 0 and 16384. The deadband function is an effective way to reduce the amount of data collected by defining the granularity of the data.

The deadband functions as a event threshold limit when used in conjunction with the “dwell” feature.

Figure 14 illustrates the deadband feature filtering out small changes in acceleration from the recorded data. Only when the deadband limit is exceeded will a new data sample be pushed to the file. Note that this feature will result in samples with inconsistent time periods. Therefore, the data sets should be re-sampled to establish uniform time periods.

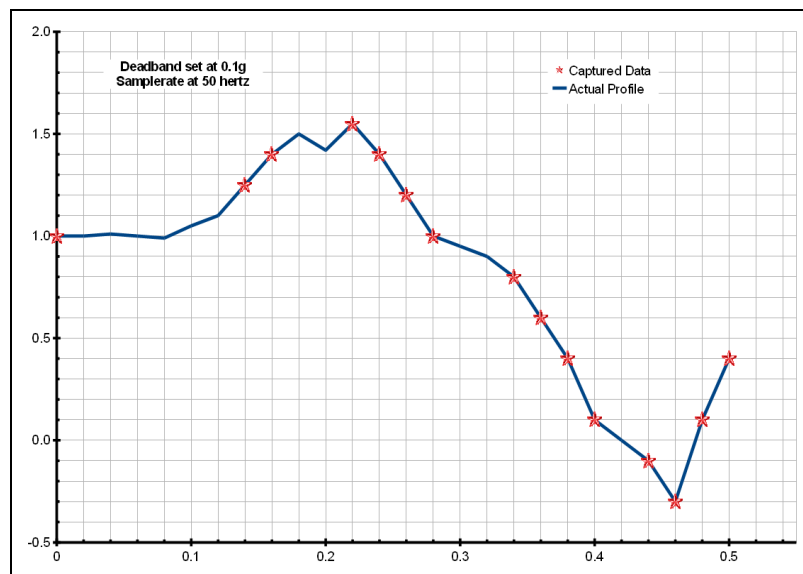


Figure 14: Graphical Illustration of the Deadband Feature

2.6.2 deadbandtimeout

“deadbandtimeout” defines the period in seconds when a sample is recorded by the logger regardless of the deadband setting. This feature ensures periodic data is recorded during extended periods of inactivity. A valid setting for the deadbandtimeout is an integer between 0 and 16384.

2.6.3 dwell

Use “dwell” together with “deadband” to create an event trigger configuraion. The “dwell” tag defines the number of consecutive samples recorded at the set sample rate after a deadband threshold event.

The deadband threshold event occurs when a sensor reading exceeds the last recorded value by the deadband setting. A valid dwell setting is an integer between 0 and 65535. See section 2.7.2 for an example implementation of the deadband/dwell features.

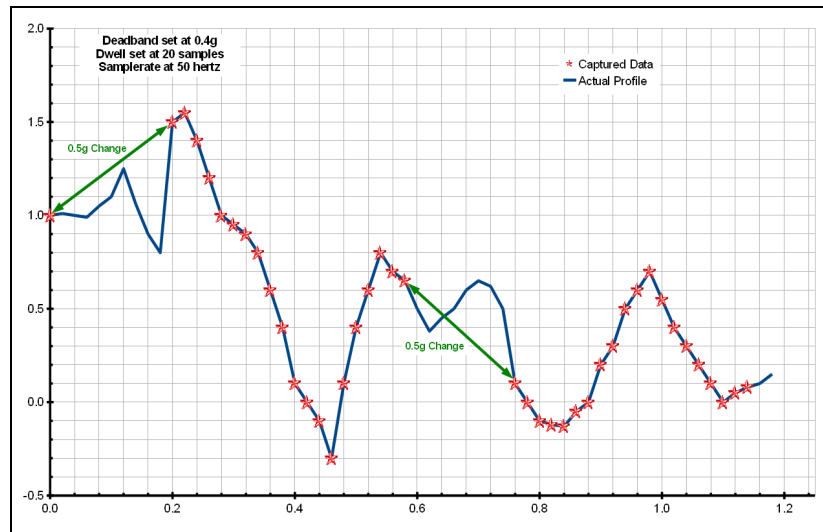


Figure 15: Graphical Illustration of the Dwell Feature

2.6.4 microres

The “microres” option sets the device to record time stamps with 0.1ms precision. In micro-resolution mode, the time stamps are recorded as XX.YYYYZZ where XX are seconds, YYYY are 0.1 milliseconds, and ZZ are spurious digits beyond the precision capability. The micro-resolution option should be implemented at sample rates greater than 200 hertz to provide the best timing precision.



Micro-resolution is best suited for applications requiring precise timing, such as vibration analysis, and is recommended for sample rates above 200 Hz. The standard timing precision (default) of 1 milli-second is suitable for most general applications, such as monitoring human motion.

2.6.5 rebootondisconnect

The X16-4 incorporates an on/off button for initiating and terminating the data recording process. Data recording is automatically started upon disconnect from a computer USB port if the tag word “rebootondisconnect” is included in the configuration file.

2.6.6 samplesperfile

“samplesperfile” defines the number of data lines each file can have before a new file is created. This tag controls the size of the data files into easily manageable lengths for later processing. This setting is loaded as a signed 32-bit integer, which can translate into very large data files. The user should exercise caution before setting large files and test the end-user software application for data limitations.

2.6.7 samplerate

The “samplerate” tag defines the data rate in Hertz, or samples per second. Valid sample rate settings are 12, 25, 50, 100, 200, and 400 Hz. See section 4.1 for special features regarding the sample rates.

2.6.8 starttime and stoptime

The X16-4 starts and stops data recording based on the times defined using the “starttime” and “stoptime” tags. The times must be in “MM HH DD” 24-hr format with the three entries separated by a space. Entries marked with “*” operate as a wild card. The X16-4 continues to record after the start time unless defined otherwise by the stoptime tag. Note that the configuration option does not include the month. Example timing configurations:

Example 1: On the 15th day, start recording at 12:30pm and stop recording at 6:00pm.

starttime = 30 12 15

stoptime = 00 18 15

Example 2: Start recording at the beginning of every hour and stop recording 45 minutes later.

starttime = 00 *

stoptime = 45 *

2.6.9 stoponvusb

The “stoponvusb” tag stops data logging operations when a 5v supply is detected on the USB connector. Without the stoponvusb option (default), the device switches power from the internal battery to the USB 5v and continues to log data.

2.6.10 statusindicators

The brightness intensity of the LED status indicators is defined using the “statusindicators” tag and valid settings of “normal”, “high”, and “off”.

2.7 Example Configuration Files

2.7.1 Example A

The following configuration records data at 100 hertz. Deadband and deadbandtimeout are set to zero so the logger will record constantly at the set sample rate. Each data file is 90,000 lines long, which is 15 minutes of data. The status indicators are set to high brightness. The logger is activated with the on/off button (notice “rebootondisconnect” is not active).

```
;Example X16-4 config file
;set sample rate
;available rates 12, 25, 50, 100, 200, 400
samplerate = 100
;record constantly
deadband = 0
deadbandtimeout = 0
;set file size to 15 minutes of data
samplesperfile = 90000
;set status indicator brightness
statusindicators = high
;rebootOnDisconnect
;see X16-4 user manual for other config options
```

Figure 16: Configuration File Example A

2.7.2 Example B

The deadband and dwell settings configure the device to record at least 5 seconds of data when a change greater than 0.1g is detected. The deadbandtimeout setting forces a sample write every hour.

```
; Example X16-4 Config file
; set to 25Hz
samplerate = 25
; trigger at 0.1g
deadband = 100
; record 5 seconds of data
dwell = 125
;force a write every hour
deadbandtimeout = 3600
; set file length
samplesperfile = 30000
; LEDs on
statusindicators = normal
```

Figure 17: Configuration File Example B

2.7.3 Example C

The logger must be turned on with the on/off button. It will enter a standby mode (blue LED blinks) while it waits for the start time. The logger will start recording at 10:30am and turn off at 2:00pm. The logger will record constantly at 400Hz and create 51 data files in the 3.5 hours. The micro-resolution is activated to provide the best timing precision at the 400 Hz sample rate.

```
; Example X16-4 Config file
; set to 400Hz
samplerate = 400
; activate precision timing
microres
; record constantly
deadband = 0
deadbandtimeout = 0
; set file length
samplesperfile = 100000
; set logger to turn on with clock
starttime = 30 10
stoptime = 00 14
; LEDs on
statusindicators = normal
```

Figure 18: Configuration File Example C

3 Data Interpretation

3.1 Data Files

The X16-4 creates a new data file when the system is booted or when the maximum number of data lines is reached in the previous data file. A system boot condition occurs when the on/off button is pressed, 5v power is restored to the system via the USB connector, or when the X16-4 is removed from a computer USB port with the “rebootondisconnect” feature enabled. Data files are placed in a folder named “GCDC” and are named data-XXX.csv, where XXX is a sequential number starting with 001. The system will create up to 999 files. At the beginning of each file, a header is written describing the system configuration and the current time when the file was created. Figure 19 represents an example data file.



A short gap in data may occur between sequential files as data is purged from the cache and a new file is allocated on the microSD card.

```
;Title, http://www.gcdataconcepts.com, X16-4, Analog Dev ADXL345
;Version, 1110, Build date, Dec 30 2015, SN:CCDC1016D9857FA
;Start_time, 2016-05-11, 10:15:24.390
;Temperature, -999.00, deg C, Vbat, 3804, mv
;SampleRate, 100,Hz
;Deadband, 0, counts
;DeadbandTimeout, 0,sec
;Headers, time,Ax,Ay,Az
0.163,10,24,-2054
0.173,21,28,-2063
0.183,17,12,-2090
0.194,3,7,-2097
0.204,30,5,-2061
0.214,24,24,-2061
0.224,17,26,-2082
0.234,44,16,-2063
0.245,19,33,-2072
0.255,-2,26,-2093
0.265,39,3,-2061
```

Figure 19: Example Data File

3.2 Data Format

Data is written to files in comma separated text format starting with the file header information and followed by event data entries. Table 2 lists the valid header tags, although not all tags may occur in the header. Each data line contains a time entry and the raw accelerometer sensor readings from the X, Y, and Z axes. The time entry is seconds elapsed from the start time recorded in the header. Add the elapsed time to the start time to determine the complete date and time of the sample.

The last line of the final data file records the reason for the termination, such as “shutdown: switched off”, “shutdown: low battery”, “shutdown: max files exceeded”, “shutdown: vbus disconnect”, or “connected to computer”. The line is designated as a comment with a semicolon (“;”).

Table 2: Data File Header Tags

Tag	Description
Deadband	A new sample from the sensor must exceed the last reading by the deadband value
DeadbandTimeout	The period in seconds when a sample is recorded regardless of the deadband setting
Headers	The names of each column of data in the file
SampleRate	Rate at which data is recorded to the microSD card
Start Time	The current time when the data file was created
Temperature	Not supported on X16-4, reports -999
Title	The name of the USB Accelerometer X16-4 unit and sensor type
Vbat	Battery voltage measured at the file start time
Version	The version control information of the firmware, including unique serial number

3.3 Data Conversion

3.3.1 Time Stamps

Each sample starts with a time stamp, which is the seconds elapsed from the start time listed in the file header. Add the time stamp to the start time to determine the complete date/time of each sample.

The time stamp calculation is incorporated easily into a spreadsheet, such as Excel or Calc. First, open the data file in a spreadsheet and parse on the comma (“,”) delimiter. Most spreadsheets will automatically parse the data using the “,” character. The parsing operation will separate the start_time into two cells – date and time. Use the “trim” function to strip the white space around the date cell and use “concatenate” to combine the text into a new start date. The spreadsheet will automatically format the new text into a date. Next, divide the time stamp entry by 86400. This converts the time stamp into a value compatible with the spreadsheet date functions. Finally, add the new time stamp to the new start date and a complete data/time is generated. Format the column as a “time” category and include the trailing “.000” to present the millisecond precision.

	A	B	C	D	E	F	G	H	I	J	K
1	Title	http://www.gcdataconcepts.com									
2	Version	1110	Build date	Jan 29 2016	SN:CCDC10161318B9C						
3	Start_time	2016-01-21	01:55:59.000				2016-01-21 01:55:59.000				
4	Temperature	-999.00	deg C	Vbat	4044	mv					
5	SampleRate	400	Hz								
6	Deadband	0	counts								
7	DeadbandTimeout	5	sec								
8	Time	Ax	Ay	Az							
9		0.042	-733	-45	1828		New Date Stamps	Ax(g)	Ay(g)	Az(g)	
10		0.044	-818	-22	1856		2016-01-21 01:55:59.042	-0.358	-0.022	0.893	
11		0.047	-872	-22	1880		2016-01-21 01:55:59.044	-0.399	-0.011	0.906	
12		0.049	-898	-18	1889		2016-01-21 01:55:59.047	-0.426	-0.011	0.918	
13		0.052	-870	0	1869		2016-01-21 01:55:59.049	-0.328	-0.010	0.859	
14		0.054	-820	16	1842		2016-01-21 01:55:59.052	-0.343	-0.010	0.814	
15		0.057	-752	7	1817		2016-01-21 01:55:59.054	-0.354	-0.019	0.818	
16		0.059	-674	-20	1760		2016-01-21 01:55:59.057	-0.354	-0.019	0.818	
17		0.062	-641	-18	1703		2016-01-21 01:55:59.059	-0.354	-0.019	0.818	
18		0.064	-659	-17	1678		2016-01-21 01:55:59.062	-0.354	-0.019	0.818	
19		0.067	-675	-29	1665		2016-01-21 01:55:59.064	-0.354	-0.019	0.818	
20		0.07	-702	-20	1667		2016-01-21 01:55:59.067	-0.354	-0.019	0.818	
21		0.072	-724	-38	1676		2016-01-21 01:55:59.070	-0.354	-0.019	0.818	

Figure 20: Time Stamp Conversion Method

The time stamps can be added directly to the start_time entry (no need to divide by 86400) when using Matlab, Octave, or R.

3.3.2 Acceleration

The X16-4 records the raw digital data from the accelerometer sensor. This helps reduce processor load, increase sample rate capability, and avoids data errors due to floating point calculations. The 16-bit data, or 65536 discrete counts, covers the full range of the +/-16g sensor. Therefore, the conversion factor is $65536 / 32 = 2048 \text{ counts/g}$.

Table 3 lists the converted data using the example raw data in Figure 19.



To determine acceleration in g's, divide the raw data by 2048. A "g" is 32.174 ft/sec² or 9.807 m/sec².

Table 3: Example Data Conversion

Raw Data (Low Gain)				Converted Data			
Time	Ax	Ay	Az	Time	Ax (g)	Ay (g)	Az (g)
0.163	10	24	-2054	05/11/2016 10:15:24.553	0.004883	0.011719	-1.002930
0.173	21	28	-2063	05/11/2016 10:15:24.726	0.010254	0.013672	-1.007324
0.183	17	12	-2090	05/11/2016 10:15:24.909	0.008301	0.005859	-1.020508
0.194	3	7	-2097	05/11/2016 10:15:25.103	0.001465	0.003418	-1.023926
0.204	30	5	-2061	05/11/2016 10:15:25.307	0.014648	0.002441	-1.006348
0.214	24	24	-2061	05/11/2016 10:15:25.521	0.011719	0.011719	-1.006348
0.224	17	26	-2082	05/11/2016 10:15:25.745	0.008301	0.012695	-1.016602
0.234	44	16	-2063	05/11/2016 10:15:25.979	0.021484	0.007813	-1.007324
0.245	19	33	-2072	05/11/2016 10:15:26.224	0.009277	0.016113	-1.011719

4 System Details

4.1 Sensor

The X16-4 uses the Analog Devices ADXL345 3-axis digital accelerometer sensor, which is based on micro-electro machined semiconductor technology (MEMS). This accelerometer sensor is similar to those used in cellphones, laptops, hard drives and other consumer electronics. Appendix 6.1 describes how a MEMS accelerometer sensor works. Table 4 lists the basic sensor and logger performance parameters but refer to Analog Devices for detailed sensor specifications.

Table 4: Accelerometer Sensor Characteristics

Parameter	Condition	Min	Typical	Max	Units
Acceleration range			±16.0		g
Sensitivity			2048		count/g
Sensitivity Deviation			±1.0		%
Nonlinearity	X, Y, Z axis		±0.5		%FS
Zero-g Offset Level Accuracy	X, Y axis	-150		+150	mg
	Z axis	-250		+250	mg
Inter-Axis Alignment Error			±0.1		Degrees
Cross-Axis Sensitivity			±1		%

The ADXL345 accelerometer sensor “pushes” data to the logger at selected rates based on a clock internal to the sensor. The sensor's clock precision and drift are undefined. For example, a selected sample rate of 50 Hz may actually push data at 52 Hz. The X16-4 incorporates a precise real time clock to independently time stamp the data as it leaves the sensor and to ensure that accurate timing is recorded to the data file. Therefore, the time stamps should be used as the reference for determining the actual sample rates.



The accelerometer sensor is based on microelectromechanical systems (MEMS) technology and is not affected by magnetic fields. Glue a magnet to the bottom of the plastic enclosure to facilitate easy attachment to iron surfaces.



The MEMS accelerometer sensor will detect the acceleration of gravity, which is a convenient feature for validating the sensor operation. Setting the logger on a flat level surface will result in -2048 counts (-1g) in the z-axis.

4.1.1 Sensor Special Features

The X16-4 implements an 8X over-sample and finite impulse response (FIR) filter algorithm at sample rates up to 400Hz. This means that the digital accelerometer sensor provides 8X the sample rate requested in the config.txt file. For example, “samplerate=400” sets the sensor to stream at 3200 Hz, which is the maximum capability of the ADXL345. The eight samples are averaged and processed through the FIR filter to improve the response characteristics. The oversampling and FIR algorithm increases the sensor's native 13-bit resolution to the 16-bit data recorded in the data file.

The X16-4 will support sample rates of 800, 1600, and 3200 Hz but the X16-4 deactivates the oversampling and FIR filter and records the native 13-bit resolution from the sensor. However, these sample rates are not guaranteed and the time stamps may become inaccurate or the logger operation could become unstable. Performance is dependent on the microSD card capability.

Figure 21 shows an example configuration setting the logger to record at 800 Hz. The 13-bit data from the sensor is right padded (LSB) into a 16-bit value to maintain consistency with the oversampled data. Therefore, the conversion factor is still 2048 counts/g.

```

; Example X16-4 Config file
; set to 800Hz
samplerate = 800
; record constantly
deadband = 0
deadbandtimeout = 0
; set file length
samplesperfile = 100000
; set status indicators
statusindicators = normal

```

Figure 21: 800Hz Sample Configuration

4.2 Operating and Storage Conditions

The X16-4 is protected from general handling conditions by the plastic enclosure but is not protected from adverse environmental conditions, such as rain, sweat, splashes, and water submersion. The temperature range is limited primarily by the lithium-polymer battery capabilities.

Table 5: Operating and Storage Conditions

Parameter	Value
Temperature Range (Operating)	-5°F ~ 130°F (-20°C ~ 55°C)
Temperature Range (Storage)	-5°F ~ 80°F (-20°C ~ 25°C)
Relative Humidity (Operating and Storage)	<90%

4.3 Dimensions

The X16-4 electronics are enclosed in a three-part semi-transparent blue plastic enclosure. The top and bottom enclosure components and the printed circuit board are secured together with a 0.75" long #6-32 screw and nut. A slip-on cap protects the USB connector. The X16-4 weighs 1.3oz (48g).

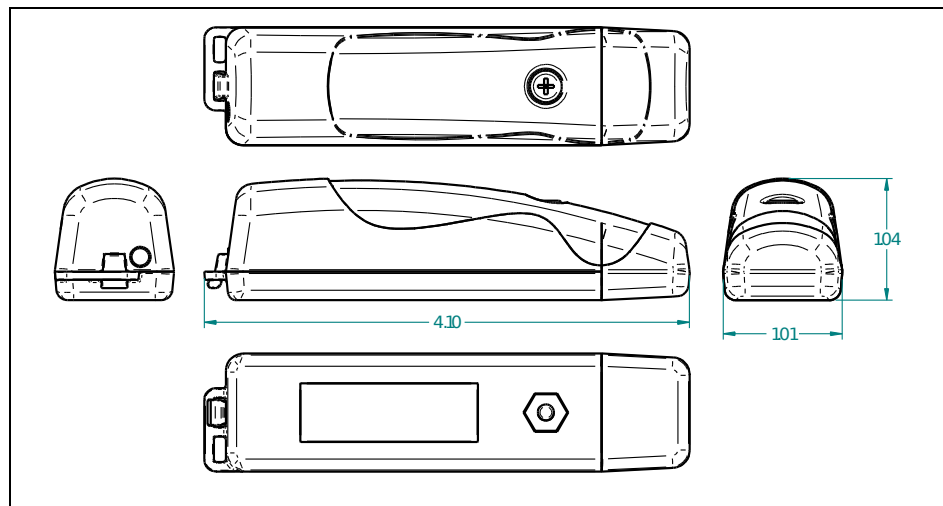


Figure 22: Enclosure Dimensions

5 Troubleshooting

Problem	Resolution
I press the on/off button but the logger does not appear to activate and no LEDs blink.	Make sure the battery is charged.
	Check that the batteries connections are securely locked into the receptacles.
	The logger could be operating correctly but the status indicators are turned off. Check the “statusindicator” option in the config.txt file.
I press the on/off button, the blue LED blinks once per second but the red LED does not indicate logging.	The deadband setting is set too high and the logger is waiting to detect an event.
	The logger is in standby mode waiting for a start time to occur. Check the config.txt file for the start/stop settings.
The blue LED blinks slowly.	The microSD card is not present or is corrupted. Check that the card is inserted properly and the card is not corrupted.
I press the on/off button but the logger records only for a short period of time.	Check that the battery is fully charged.
	The microSD card is full and data files must be deleted.
I plug the logger into a USB port but the PC does not indicate an external drive present.	The microSD card is not present in the logger or is not inserted properly. Check that the card is fully inserted into the logger.
	The microSD card is corrupted or damaged. Reformat the card or replace the card.
	The on/off button could be jammed in the plastic enclosure and the logger is stuck in the “off” state. Check that the button moves freely and “clicks” when pressed.
	The USB connection could be faulty or the extender cable (if present) could be faulty. Remove the extender cable and plug the logger into another USB port.
How do I calculate displacement from acceleration?	The inherent noise and drift characteristics of the MEMS accelerometer sensor introduces errors into the displacement calculations. The errors accumulate quickly over long time periods, which produce poor displacement results. Multiple sensors types and complex filtering algorithms are required to reduce errors in the displacement calculations.

Problem	Resolution
The logger seems to ignore the config.txt file and use default settings.	Check that the config.txt file is properly formatted and not corrupted. Each setting should occur on a separate line.
	Some IT organizations implement an automatic encryption of all removable media devices. This will encrypt the config.txt file and the logger will not be able to access the file. Do not allow encryption of the device.
The start time in the data file header is incorrect.	Initialize the RTC.
The time.txt file doesn't load to the RTC	Check that the date/time is correctly formatted and the file name is "time.txt". Windows may hide the file extension so naming the file as "time.txt" may actually be named "time.txt.txt". Check the Windows file explorer options the view of regarding file extensions.
The Z-axis data is missing in the file.	No, it's present but the column headers are shifted in your spreadsheet due to the presence of the "headers" tag.
The logger is stationary but it registers 1g.	This is normal and indicates Earth's gravity is operating correctly.
But the logger actually registers something other than 1g when stationary.	The sensor will exhibit a slight offset error. Add or subtract the appropriate amount to correct the error.
I want to calibrate my X16 logger.	The logger does not have provisions for implementing user-defined calibration coefficients and any corrections must be performed post-process.
My logger is doing crazy stuff, can I reset it?	Disconnect both battery plugs to reset the logger's processor. This will restart the logger but it will not affect the microSD card or the logger firmware. The RTC will need re-initialization.
My logger does not have battery connectors as shown in Figure 10.	A shipment of batteries did not include the appropriate connectors due to a supplier error so these batteries are hardwired. If a reset of the logger is required, use a small screwdriver to briefly short across capacitor C18 (located between the on/off switch and battery pack). This will take the power supply controller offline and reset the logger.

6 Appendix

6.1 What is an Accelerometer

Acceleration is the change in velocity. A stationary body has no acceleration and a body moving at a constant velocity has no acceleration. However, a body changing from a stationary condition to motion experiences acceleration by means of a force. Newton's second law establishes this relationship as $F=ma$.

An accelerometer [ak-se-lə-'ră-mə-tər] sensor measures the force acting on a known mass to determine “proper” acceleration. There are many ways to measure force with each method having benefits and limitations. A simple method measures the displacement of a spring-mass system (see Figure 23). As a force acts upon the mass, such as gravity, the spring will stretch a certain distance relative to the spring constant. Knowing the spring constant and spring displacement, the force acting upon the mass is calculated. Acceleration is the force divided by the mass.

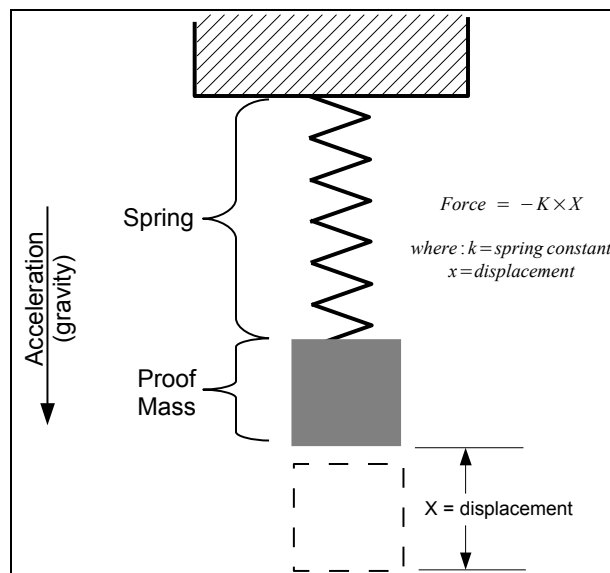


Figure 23: Spring-mass Accelerometer

Micro electro machined sensor (MEMS) technology takes the spring-mass concept and miniaturizes it onto a semiconductor chip. Figure 24 illustrates the general concept of a MEMS accelerometer system and Error: Reference source not found shows the internal layout of an actual MEMS accelerometer sensor. The mass and spring system is etched into the semiconductor layer. When the sensor experiences an acceleration, the proof mass moves and the distance between the interleaving “fingers” changes. The change in electrical capacitance between the fingers is proportional to the displacement of the spring-mass system. The capacitance is measured, filtered, and converted to a digital output representing acceleration.

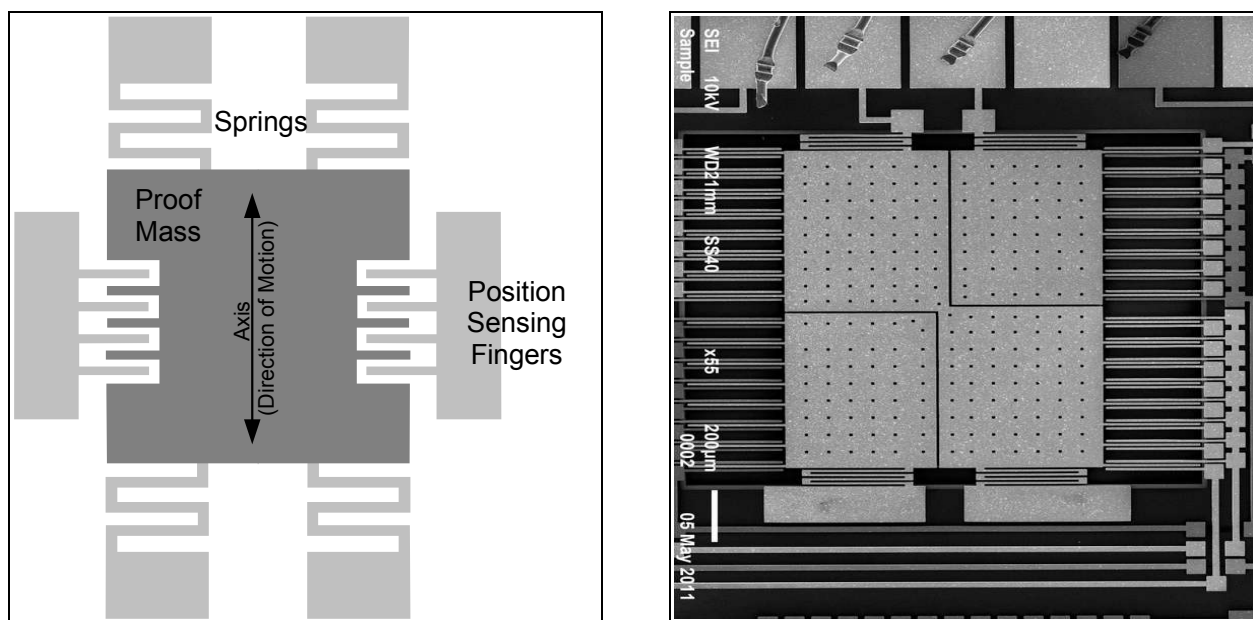


Figure 24: Simplified MEMS Accelerometer Design (L) and Actual MEMS Accelerometer (R)

Accelerometer sensors exhibit several types of limitations, including offset error, drift error, sensitivity error, and noise. Offset and sensitivity errors are corrected by calibrating the sensor against known accelerations. Drift errors are typically related to temperature changes but can be minimized by maintaining a consistent environment temperature. Noise is the random variations introduced into the sensor system. Oversampling algorithms and signal filters minimize the effects of sensor noise. Each of these sensor errors affect how the data is processed into a usable result. For example, integrating acceleration to determine displacement is heavily skewed due to the drift and noise characteristics of the sensor.

Accelerometer sensors detect translation motion within the axis of the proof-mass. Rotational motion causes centripetal acceleration that is interpreted as translational motion by the accelerometer. For example, spinning about the z-axis will cause acceleration in the x/y axes even though there is no translational motion in the x/y plane. A gyroscope sensor, which measures rotational velocity about an axis, is needed to discern rotational motion from translational motion. An accelerometer sensor and gyroscope sensor are required to determine the six variables of 3D motion. This combination of sensors is considered an inertial measurement unit (IMU) system. Some IMU systems include an additional magnetometer sensor (compass) and GPS to further aid the calculations of motion.

6.2 Using “R” to Analyze Data

6.2.1 What is “R”

You collected a data set using a GCDC logger and realized, “Wow, that's a lot of data! Now what?”. Data analysis is tedious and the process is particular to each user's application. Don't expect to find a magic software solution that will reduce your data into your perfect answer. However, don't despair. There are several options available, combined with a little bit of user effort, that provide powerful and versatile analysis capabilities.



Spreadsheets, such as Microsoft Excel or OpenOffice Calc, are great choices for plotting moderately sized data sets. The user interfaces are highly polished and customized plotting is easy to handle. Although, most spreadsheets can handle only about 100,000 lines of data before performance begins to slow. Furthermore, scripting complex analysis procedures in a spreadsheet is cumbersome. We recommend trying “R” because it is more powerful than a spreadsheet and it is easy to learn.

“R” is a high-level programming language used most commonly for statistical analysis of data. R is based on the “S” language, which was developed by the Bell Laboratories in the 1970s. R provides a simple workspace environment that can manipulate large data sets using simple math commands and complex function libraries. R is widely used by statisticians and data miners and the language is well supported by the open source community. The software is compact, free, and available for Windows, Mac, and Linux (visit www.r-project.org).

Matlab is another common software application for analyzing data but it is usually reserved to universities or businesses with copious budgets (it's expensive software!). Octave is a free open source adaptation of Matlab with nearly the same capabilities. Although, Octave is a significantly larger download and more complicated installation than R. We favor R because it's small, easy to learn, and free.

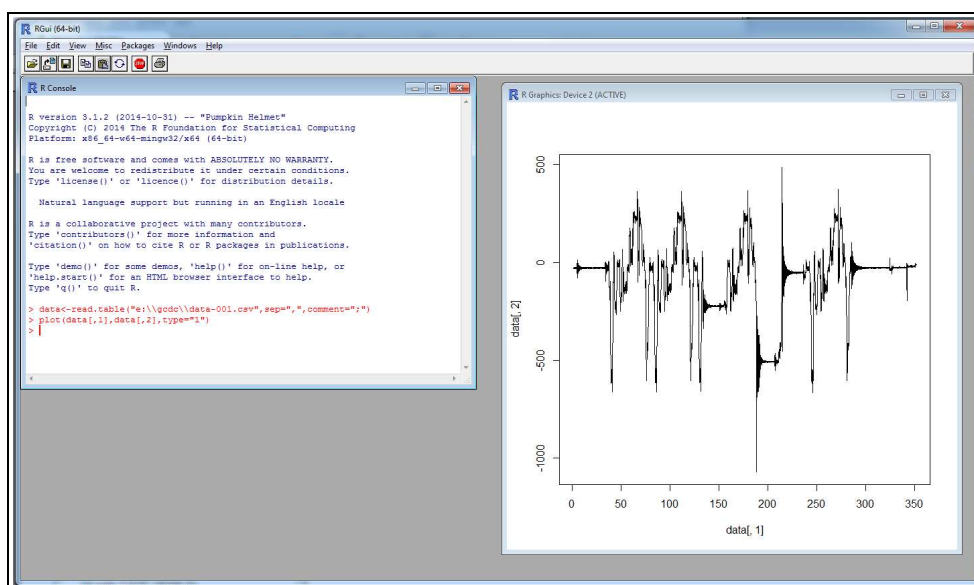


Figure 25: R Command Line Interface

R is implemented from a command line interface as seen in Figure 25. If you are an experienced programmer, you may even cringe at some of the constructs used in R. Don't worry, it just works. User input occurs at the “>” prompt and the R interpreter responds with the results. A single result is preceded by a [1] to indicate the response number. The “;” character is used to add comment information that the R interpreter ignores.

The R workspace includes a single command line interface window and a separate graphics window for displaying plots. “RStudio” is free software package that provides a more versatile interface to the R interpreter. RStudio is available at www.rstudio.com

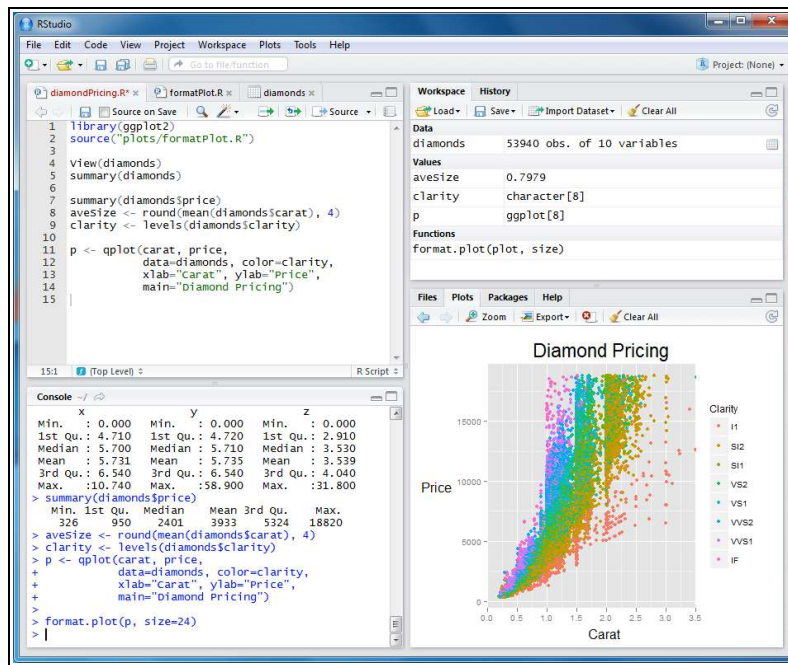


Figure 26: RStudio Interface

6.2.2 Introduction to R Commands

R recognizes basic math operators, such as +, -, *, and /. Assignments are made using “<-”. For example:

```

> 2+2
[1] 4
> a<-2+2      ;assign "a" the result of 2+2
> a
[1] 4

```

In the above example, 'a' was assigned the value '4' and can be used later. R works with vectors and matrices as well.

```

> b<-c(1,2,3) ;"c" is a function call that creates a vector
> b
[1] 1 2 3
> a*b
[1] 4 8 12

```

More complex math steps are handled in separate functions or external scripts.

```
> fun<-function(a,b){
+ c<-a+b           ; the "+" indicates the user input wrapped into next line
+ return(c)
+ }
> fun(2,2)
[1] 4
```

R will read a data file from the X16-4 data logger using the “read.table” function.

```
> data<-read.table("d:\\GCDC\\data-001.csv", sep="," ,comment=";", fill=TRUE)
```

“data” is a matrix of 4 columns containing the time, Ax,Ay,Az values from the file. Values within the matrix are accessed as follows:

```
> data[100,2]   ;row 100, column 2
[1] 101
> a*data[100,2]
[1] 404
```

The raw data is converted and assigned to new vectors.

```
> dataX_g<-data[,2]/2048      ;convert the x-axis to g and assign to new vector
> dataY_g<-data[,3]/2048
> dataZ_g<-data[,4]/2048
```

Now, the acceleration in g's is plotted against the elapsed time.

```
> plot(data[,1], dataX_g, type="l")      ; create a line plot of x-axis values
> lines(data[,1], dataY_g, type="l", col="blue")  ; add another line to plot
```

The converted data can be combined into a new matrix and then exported to a new csv data file.

```
> output<-array(c(data[,1],dataX_g, dataY_g, dataZ_g), dim=c(length(data[,1]),4))
> write.table(output, "c:\\output_data.csv", sep="," )
```

An analysis can be automated by saving the commands into an external text file. Use “source” to call the file and R will execute the script inside workspace.

```
>source("d:\\hello_world.r")
[1] hello world
```

Documentation of the available commands is accessed using “help” or by using an internet search engine.

```
> help("plot")           ; opens a browser with the help documentation for “plot”
```

6.2.3 Online Resources for R

Home page for R to download the software:

<https://www.r-project.org/>

A complete introduction to R at their website:

<https://cran.r-project.org/doc/manuals/r-release/R-intro.html>

Another good tutorial for R beginners:

<http://www.cyclismo.org/tutorial/R/>

R is widely supported by user created packages that expand the capabilities of the language. These packages are libraries of functions built for specific applications. Here is a list of available packages:

https://cran.r-project.org/web/packages/available_packages_by_name.html

If you ever get stuck trying to solve an issue with R, it's very likely someone else faced the same challenge and posted the question to R forums. Search the internet and you will find a solution to get you back on track.

6.2.4 Example Scripts in R

Several example applications using R scripts are available at the GCDC website or are included with the X16-4 data logger. These examples educate the user on basic operation of the data logger, interpretation of acceleration data, and the use of R scripts.

End of User Manual