Gulf Coast Data Concepts

www.gcdataconcepts.com

X16-1E USB Accelerometer Data Logger

User Manual

Document Revision:	Rev New
Firmware Version:	2037
Date:	May 25, 2021

Table of Contents

Index of Figures

Index of Tables

1 Introduction

1.1 About This Manual

Thank you for purchasing the X16-1E 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 X16-1E. If the logger fails to operate as expected, please refer to the troubleshooting guide (page 22).

1.2 Document Conventions

The quick start guide in section 1.8 provides a basic summary of operation to begin using the X16-1E 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 X16-1E and how to use the data.

1.4 Product Summary

The X16-1E 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 X16-1E appears as a standard mass storage device containing the comma delimited data files and the user setup file. The X16-1E is powered from a replaceable AA type alkaline battery (or LR6).

1.5 Feature List

- 3-axis ±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)
- Uses a standard replaceable "AA" type alkaline battery
- Weight 2oz (55g) with alkaline battery
- Size 1x1x4.1 inch (26x26x104 mm)

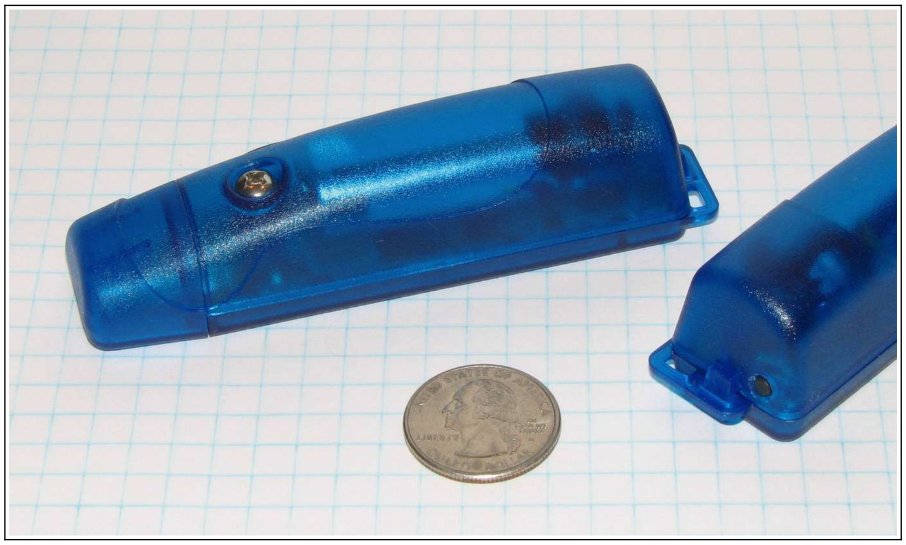


Figure 1: X16-1E Data Logger

1.6 Items Included with X16-1E

1.6.1 Single Unit Purchase

The X16-1E is packaged with the logger, a USB extender cable, a screwdriver, and an AA alkaline battery.

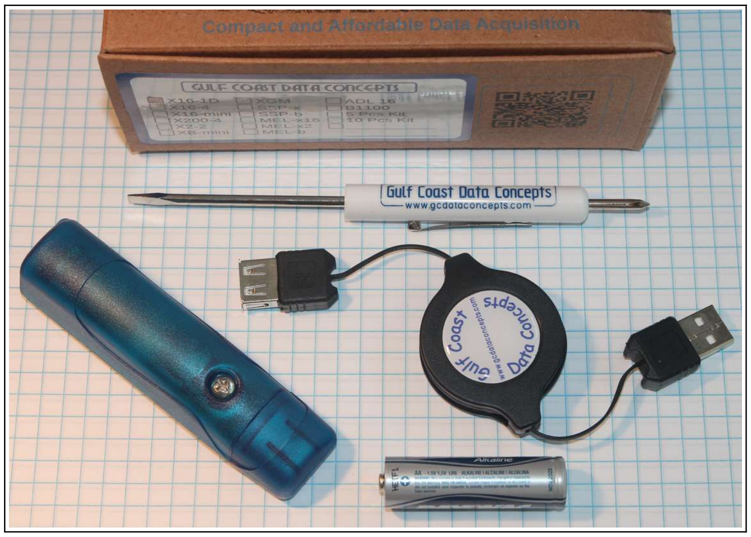


Figure 2: X16-1E and Accessories

1.6.2 5 Unit Kit

A kit includes 5 X16-1E loggers, a USB extender cable, and a screwdriver. Batteries are not included with kits.

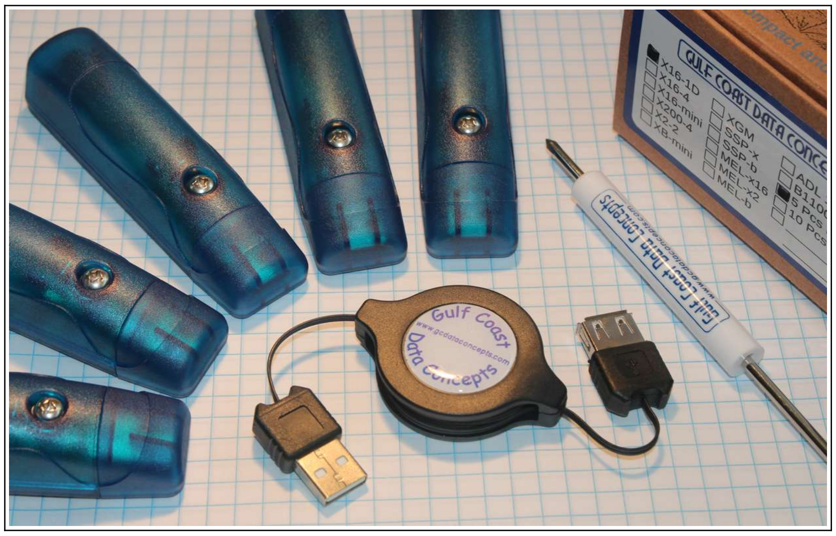


Figure 3: 5 Unit Kit of Loggers

1.7 Component Names

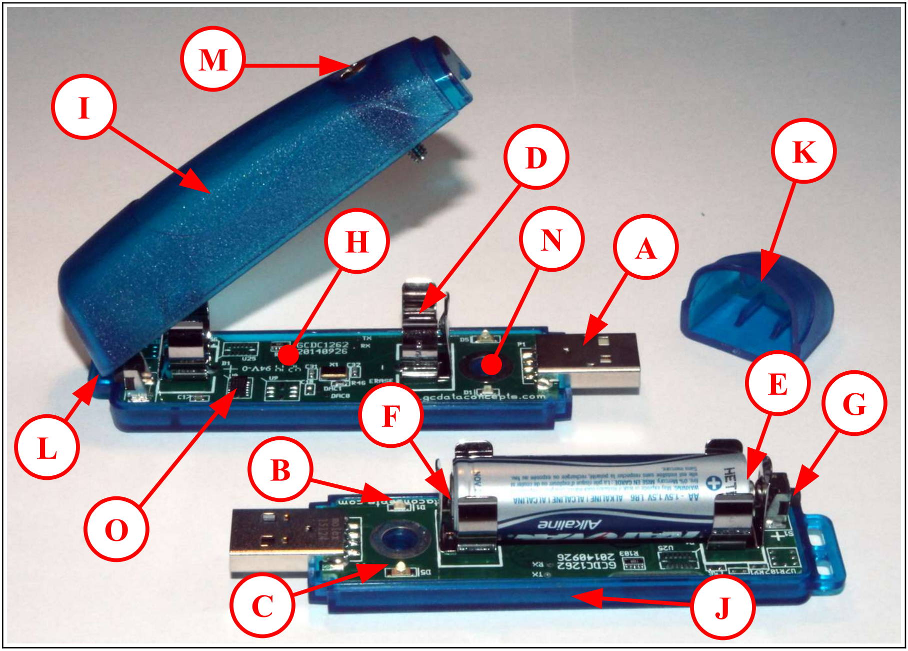


Figure 4: X16-1E Data Logger Components

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	AA Battery holder	L	Enclosure hinge
E	Positive terminal	M	#6-32 3/4" screw
F	Negative terminal	N	#6 nut
G	On/Off button	O	ADXL345 sensor
H	MicroSD card (under circuit board)		

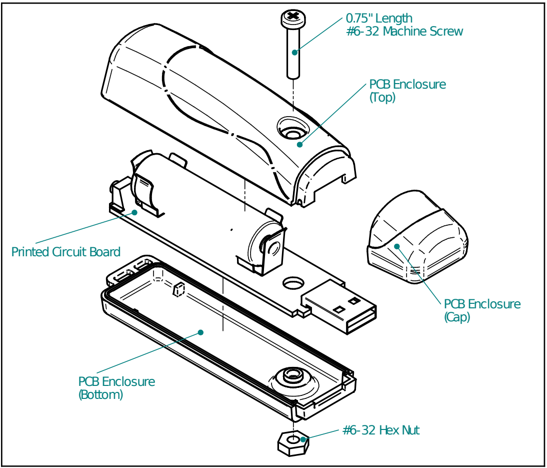


Figure 5: Exploded View of the X16-1E

1.8 Quick Start Guide

The X16-1E 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 X16-1E. Configuration settings and mounting methods will depend on the particular application.

Step 1: Disassemble the enclosure by unscrewing the #6 machine screw and opening the parts like a clam shell. Place an AA type battery into the battery holder with the positive battery terminal facing away from the USB connector. Reassemble the enclosure.

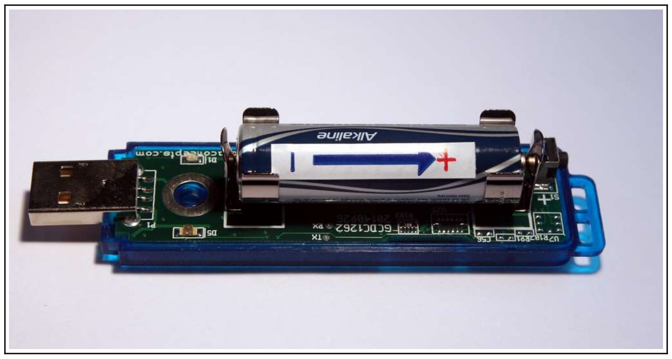


Figure 6: Installing Battery

Step 2: Plug the X16-1E 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 logger's serial number.

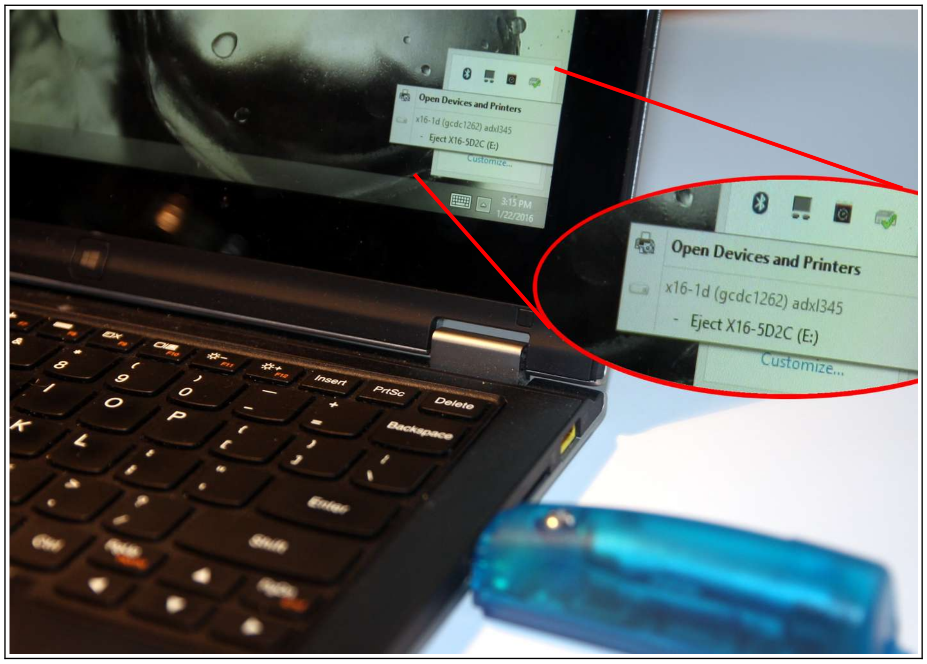
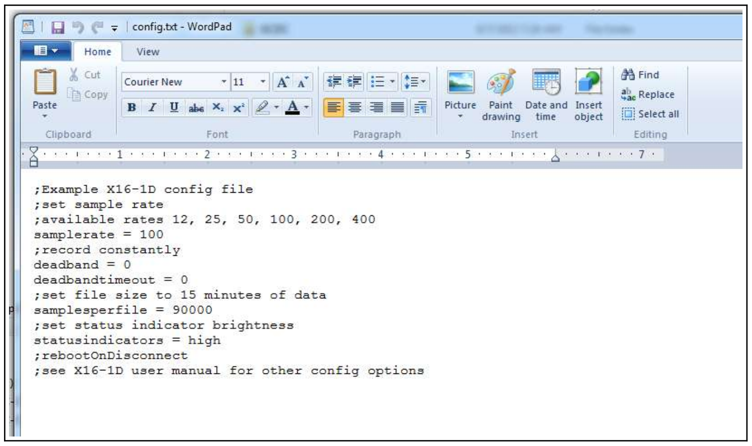


Figure 7: Connecting to PC

Step 3: Configure the X16-1E 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. GCDC recommends Windows Wordpad or Notepad++ to edit the config.txt file. Refer to section 2.6 for a complete list of configuration options.



```
;Example X16-1D 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 X16-1D user manual for other config options
```

Figure 8: Editing Config.txt File

Step 4: 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 so that the data samples can be correlated to a specific date and time.

Step 5: After removing from the USB port, attach the X16-1E logger to the target object. The logger is small and lightweight, so double-sided tape, Velcro, or a spot of cyanoacrylate glue are sufficient methods of attachment.

The X16-1E is small and lightweight, so attachment methods do not need to be substantial. Double-sided tape, a spot of cyanoacrylate glue (contact cement), zip-ties, a 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 X16-1E logger. Command Poster Adhesive strips by 3M offer excellent temporary attachment of the logger to most surfaces.

Step 6: Press the on/off button located at the rear of the enclosure to initiate data recording, (see Figure 9). Logging will start about 3-5 seconds after pressing the button. The red LED will blink as the configuration file is accessed. 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.

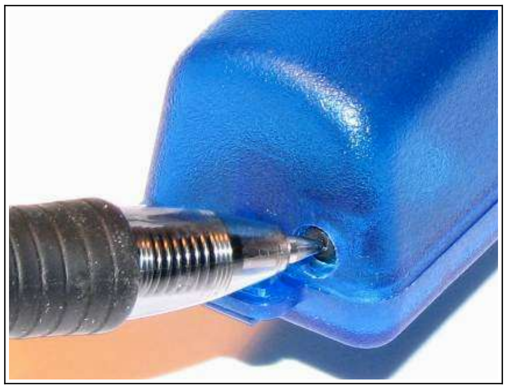


Figure 9: Starting the X16-1E

Step 7: 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 X16-1E turns off. Pressing the button again restarts the logger, and data is recorded to a new file.

Step 8: 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 9: The data recorded to the files must be converted to determine acceleration in “g” units. Divide the Ax, Ay, and Az columns by 2048 to determine g units. See section 3.3 for a complete discussion of data conversion.

The data files contain raw digital data from the sensor. To determine acceleration in g's, divide the raw data by 2048. A “g” is 32.174 ft/sec^2 or 9.807 m/sec^2.

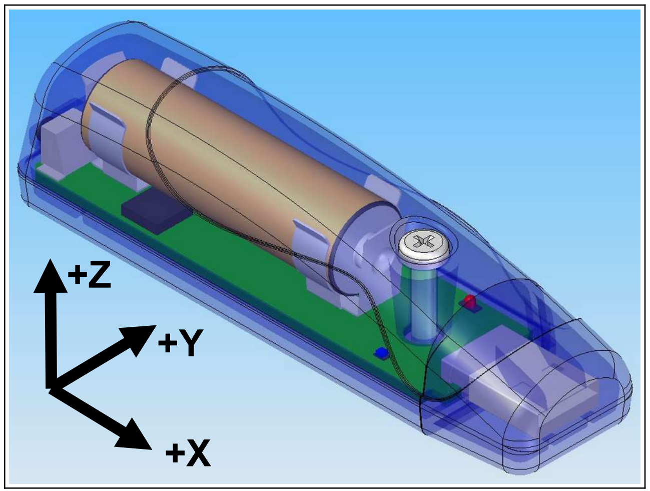


Figure 10: Sensor Orientation

2 Operation

2.1 USB Interface

The X16-1E 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 X16-1E as a typical USB external memory drive. When connected to a PC, the X16-1E 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 X16-1E.

Note that the logger does not record data while connected to a PC.

2.2 Memory Card

The X16-1E stores data to a removable 8GB microSD flash memory card and is compatible with microSD and microSDHC type cards. The logger uses FAT32 file structure, so cards larger than 8GB are supported.

The logger needs only the config.txt file to operate. The X16-1E 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 X16-1E will create a folder called "GCDC", if not already present, to place the data files.

Interrupting the power to the logger, for example, removing the logger from the USB port during file transfers to the PC or removing the battery during logging activity, can result in corruption of the microSD card. 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 X16-1E is powered by a single "AA" sized battery. Gulf Coast Data Concepts recommends an alkaline battery (ANSI type 15A or IEC type LR6) or lithium battery (ANSI type 15L or IEC FR6) to operate the X16-1E. The battery is not used when the device is connected to a computer USB port. The X16-1E will log constantly for approximately 50 hours at 50 Hz and approximately 29 hours at 100 Hz when using a standard alkaline AA battery. 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).

Use a lithium primary AA battery to improve low temperature performance and extend operating time. The lithium chemistry has a wider operating temperature of -40°F – 140°F (-40°C – 60°C) and about 30% more capacity over a standard alkaline battery.

The X16-1E will not operate from an external 5v USB power supply, unlike previous versions of this logger. Extending the operating time is only achieved with larger battery capacity added directly to the AA battery terminals.

The logger is always "on" as it maintains the real time clock and will eventually discharge the battery completely after several months. The discharged battery may leak chemicals and corrode the electronics. Remove the AA battery prior to long term storage of the X16-1E.

The X16-1E is not compatible with NiMH type rechargeable batteries. The NiMH battery delivers 1.2 volts nominally, which is near the voltage cut-off limit of the logger. The X16-1E will not operate very long or may not activate at all.

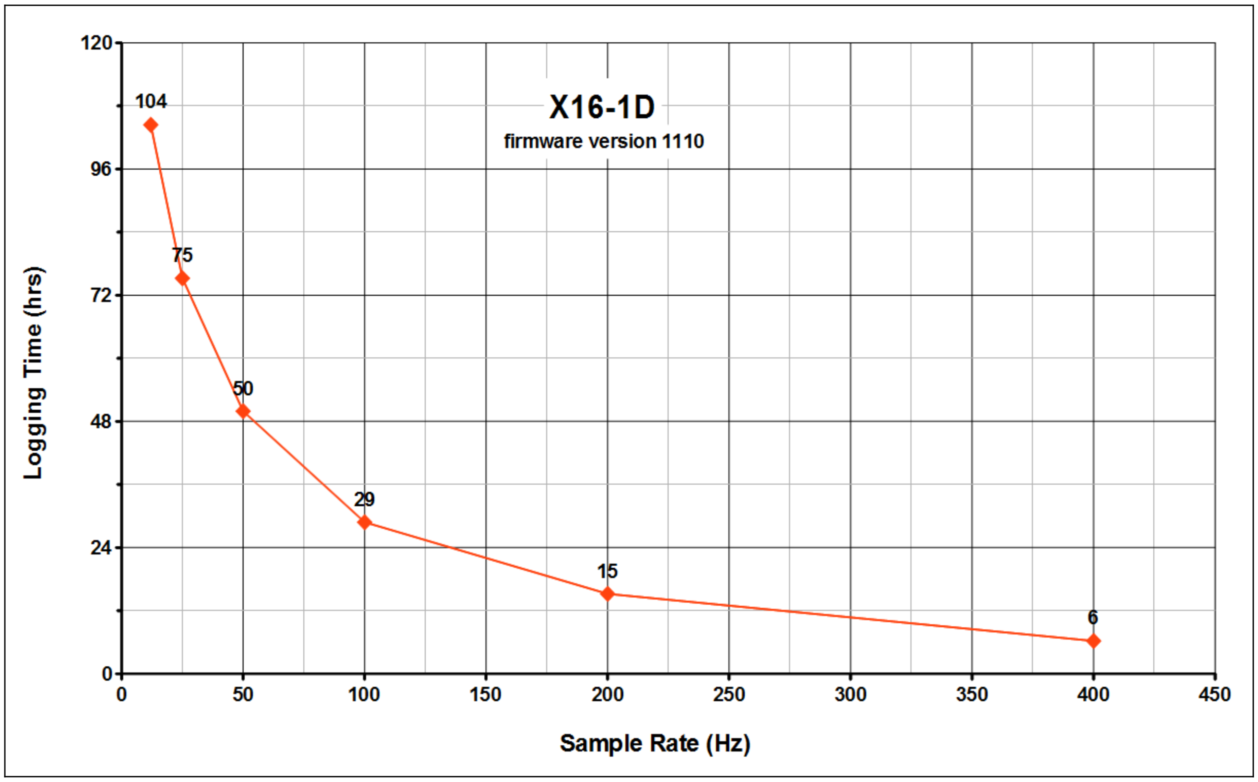


Figure 11: Expected Battery Life

2.4 Setting The RTC

A real time clock (RTC) integrated into the X16-1E 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:

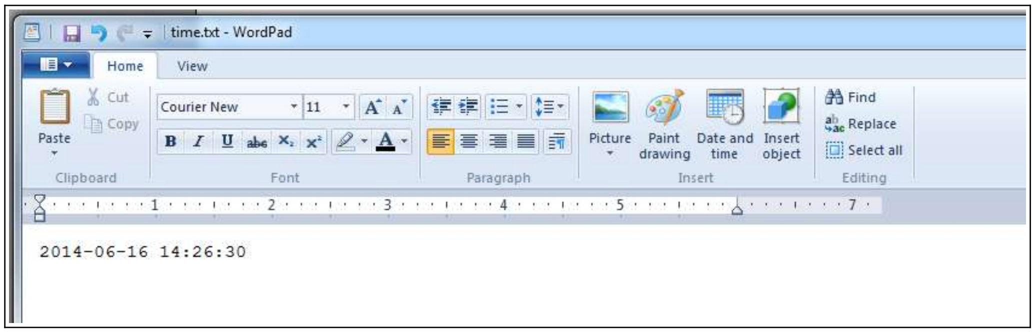
Step 1: Use Wordpad, or an equivalent text editor, to create a simple text file called "time.txt".

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 form the PC. The logger will automatically find the time.txt file and intialize the RTC with the time stored in the file. The file is deleted after initialization.

The RTC maintains ±50ppm accuracy (-40°C to +85°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".



```
2014-06-16 14:26:30
```

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 cannot 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 X16-1E is recording data, in standby mode, or 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 2.6.11).

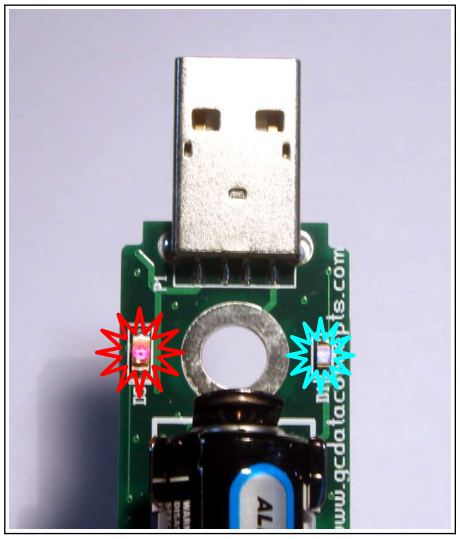


Figure 13: LED Status Indicators

2.6 System Configuration Options

The X16-1E 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 the default settings listed in Table 1 if the config.txt file is not found.

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.

Table 1: Configuration File Tags and Descriptions

Tag	Valid Settings	Default	Description
absoluteTime	-	Off	Time stamps relative to epoch (Jan 1, 1970)
deadBand	An integer between 0 and 16384	0	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
dirName	Character text	/GCDC	Defines directory name to store data files
dwell	An integer between 0 and 65535	1	The number of samples recorded after a deadband threshold triggered event
fileName	Character text	DATA-	File name prefix for the data files
minBattVoltage	An integer greater than 0	1200	set the minimum operating battery voltage (mV)
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.
statusIndicators	“Normal”, “High”, “Off”	Normal	LED status indicators can be activated with normal brightness (Normal), activated with high brightness (High), or completely deactivated (Off).
wakeUpTime	integers	Off	Activates logger at specific times

2.6.1 absoluteTime

By default, the time stamps represent the elapsed seconds since the start_time value listed in the file header. “absoluteTime” changes the start reference to midnight January 1, 1970, otherwise known as “epoch” or Unix time 0.

2.6.2 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.

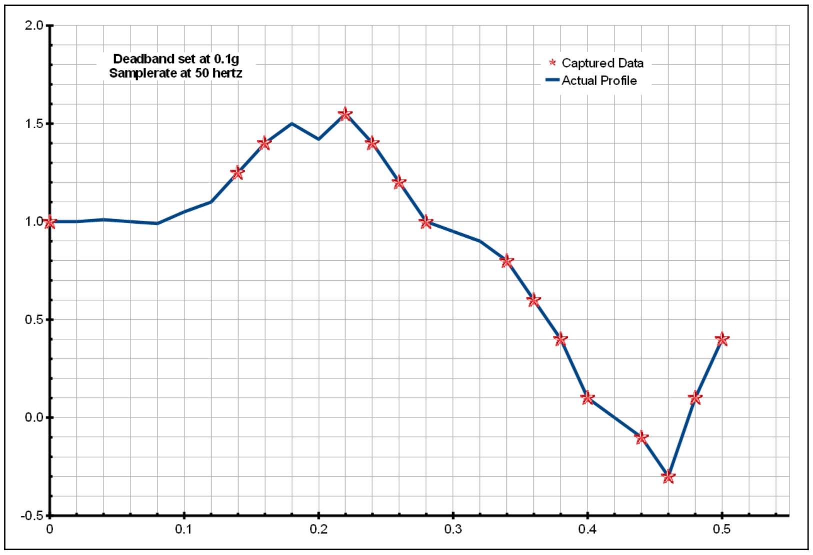


Figure 14: Graphical Illustration of the Deadband Feature

2.6.3 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.

2.6.4 dirName

The logger will store data files into the directory defined by "dirName". The directory must be defined with a preceding slash, such as "dirName=/GCDC". By default, the data directory is set to the root location /GCDC.

2.6.5 dwell

Use "dwell" together with "deadBand" to create an event trigger configuration. 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.

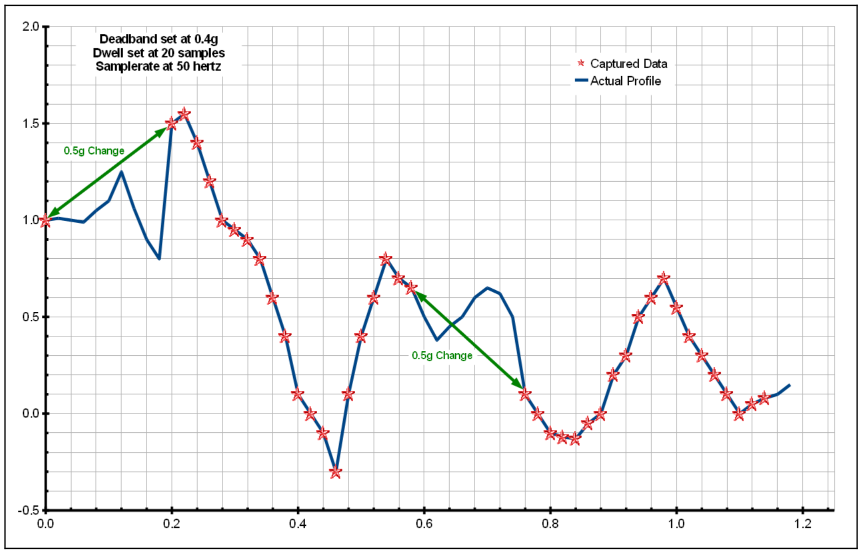


Figure 15: Graphical Illustration of the Dwell Feature

2.6.6 fileName

"fileName" sets the prefix name of the data files. By default, fileName is set to "DATA-".

2.6.7 minBattVoltage

The logger will initiate a low-battery shutdown when the minBattVoltage is detected. By default, the minBattVoltage is set to 1200 millivolts.

The default minimum battery voltage is set to 1200 millivolts to ensure that the alkaline battery can provide the necessary current to support microSD write operations. Care must be taken when modifying the minBattVoltage parameter.

2.6.8 rebootOnDisconnect

The X16-1E 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.

2.6.9 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.

2.6.10 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.

2.6.11 statusIndicators

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

2.6.12 wakeUpTime

The "wakeUpTime" option configures the logger to turn on at specific times each day. Parameters are separated by spaces and are in order of minutes then hours. Multiple parameters are separated by commas. For example, "wakeUpTime=5,20 4,15" turns the logger on at 5 minutes and 20 minutes past the hour of 4am and 3pm. "wakeUpTime=*" will turn the logger on with each minute. There are three additional parameters needed to complete the wakeUpTime option and each must be on a separate line in the config.txt file:

"secsToRecord" defines the time period of data to record in seconds. For example, "secsToRecord=50" will record 50 seconds of data after a wake up event.

"fileAppend" will append new data to the previous available data file. The logger will create a new file with each wake up event if fileAppend is not used.

"offOnEndRecord" turns the logger off after the completion of each wake up event. This option saves power since the logger is not active between wake up events. Otherwise, the logger will stay in a standby mode (blue LED blinks) while waiting for the next wake up event.

Each time the logger completes a wake up event, the remaining portion of the memory sector is filled with a repeating comment string (";sectalign"). This procedure ensures that the next wake up event starts on a new memory sector, which makes flash memory allocation easier for the logger. For the end-user, ignore these ";sectalign" comment strings.

A wakeUpTime event is triggered upon the first time the logger is turned on, regardless of the clock time. After this event completes, the logger will record data at the times specified by the wakeUpTime option.

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 X16-1E config file
;set sample rate
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
```

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 X16-1E Config file
; set to 25Hz
samplerate = 25
; trigger at 0.1g
deadband = 205
; 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 wake up time. The logger will start recording at 4:30pm and record 5 minutes of data every day.

```
; Example X16-1E Config file
; set to 400Hz
samplerate = 400
; record constantly
deadband = 0
deadbandtimeout = 0
; wake up at 4:30pm
wakeUpTime = 30 16
; record 5 min
secToRecord = 300
; set file length
samplesperfile = 100000
statusindicators = normal
```

Figure 18: Configuration File Example C

3 Data Interpretation

3.1 Data Files

The X16-1E 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 or when the X16-1E 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 directory and file prefix are configurable parameters (see sections 2.6.4 and 2.6.6). 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.

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. 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 (default mode) or relative to Jan 1, 1970 (absoluteTime mode). Figure 19 represents an example data file.

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 (";").

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, x16-1e, ADXL345
;Version, 2037, Build date, Dec 18 2020,  SN:CCDC5016091F6D5
;Start_time, 2021-05-25, 10:25:14.000
;Uptime, 5,sec,  Vbat, 1292, mv, EOL, 3500, mv
;SampleRate, 100,Hz
;Deadband, 0, counts
;DeadbandTimeout, 0.000,sec
;Time, Ax, Ay, Az
0.003,799,650,-1773
0.013,805,661,-1808
0.023,766,687,-1844
0.033,790,670,-1818
0.042,801,663,-1808
0.052,769,657,-1786
0.062,790,683,-1795
0.072,813,719,-1853
0.081,824,670,-1784
```

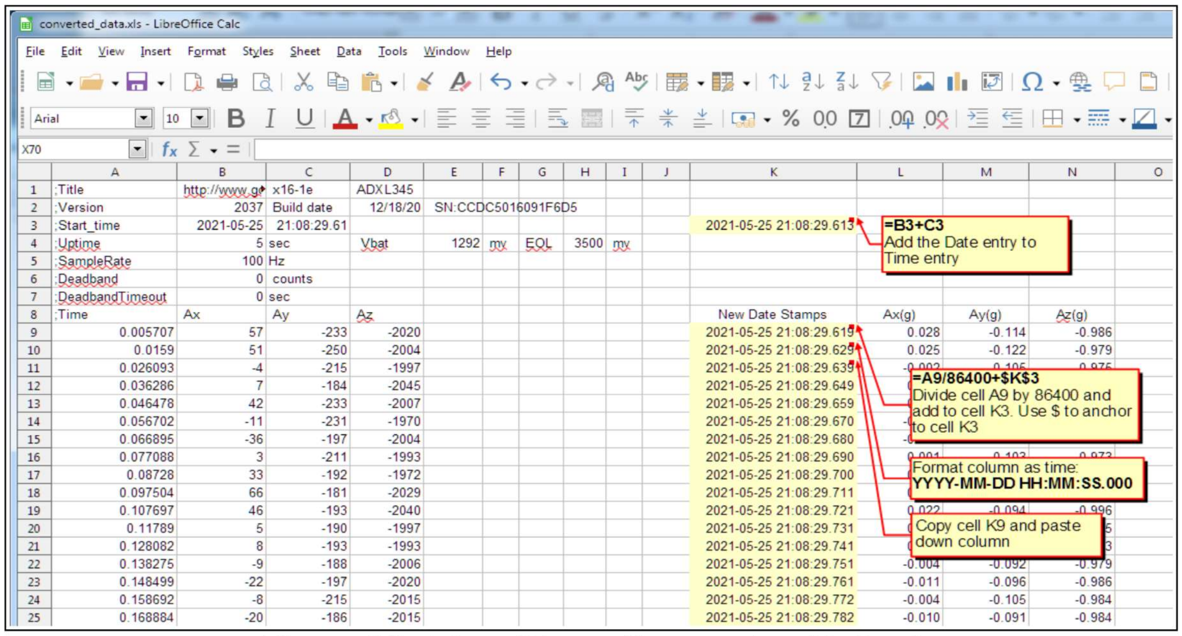
Figure 19: Example Data File

3.3 Data Conversion

3.3.1 Time Stamps: Relative Mode

By default, the time stamps represent the elapsed seconds relative to 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 Microsoft Excel, LibreOffice Calc, or Google Sheets. First, open the data file in a spreadsheet and parse on the comma (",") deliminator. Most spreadsheets will automatically parse the data using the "," character. The parsing operation will separate the start_time into two cells – date and time. Add the two cells together to create the 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.


converted_data.xls - LibreOffice Calc
File Edit View Insert Format Styles Sheet Data Tools Window Help
Arial
10
X70

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	;Title	http://www.g	x16-1e	ADXL345											
2	;Version	2037	Build date	12/18/20	SN:CCDC5016091F6D5										
3	;Start_time	2021-05-25	21:08:29.61								2021-05-25 21:08:29.613				
4	;Uptime	5	sec	Vbat	1292	mv	EOL	3500	mv						
5	;SampleRate	100	Hz												
6	;Deadband	0	counts												
7	;DeadbandTimeout	0	sec												
8	;Time	Ax	Ay	Az							New Date Stamps	Ax(g)	Ay(g)	Az(g)	
9	0.005707	57	-233	-2020							2021-05-25 21:08:29.619	0.028	-0.114	-0.986	
10	0.0159	51	-250	-2004							2021-05-25 21:08:29.629	0.025	-0.122	-0.979	
11	0.026093	-4	-215	-1997							2021-05-25 21:08:29.639	-0.002	[illegible]	[illegible]	
12	0.036286	7	-184	-2045							2021-05-25 21:08:29.649	[illegible]	[illegible]	[illegible]	
13	0.046478	42	-233	-2007							2021-05-25 21:08:29.659	[illegible]	[illegible]	[illegible]	
14	0.056702	-11	-231	-1970							2021-05-25 21:08:29.670	[illegible]	[illegible]	[illegible]	
15	0.066895	-36	-197	-2004							2021-05-25 21:08:29.680	[illegible]	[illegible]	[illegible]	
16	0.077088	3	-211	-1993							2021-05-25 21:08:29.690	0.001	[illegible]	[illegible]	
17	0.08728	33	-192	-1972							2021-05-25 21:08:29.700	[illegible]	[illegible]	[illegible]	
18	0.097504	66	-181	-2029							2021-05-25 21:08:29.711	[illegible]	[illegible]	[illegible]	
19	0.107697	46	-193	-2040							2021-05-25 21:08:29.721	0.022	-0.094	-0.996	
20	0.11789	5	-190	-1997							2021-05-25 21:08:29.731	[illegible]	[illegible]	[illegible]	
21	0.128082	8	-193	-1993							2021-05-25 21:08:29.741	[illegible]	[illegible]	[illegible]	
22	0.138275	-9	-188	-2006							2021-05-25 21:08:29.751	-0.004	-0.092	-0.979	
23	0.148499	-22	-197	-2020							2021-05-25 21:08:29.761	-0.011	-0.096	-0.986	
24	0.158692	-8	-215	-2015							2021-05-25 21:08:29.772	-0.004	-0.105	-0.984	
25	0.168884	-20	-186	-2015							2021-05-25 21:08:29.782	-0.010	-0.091	-0.984	

=B3+C3
Add the Date entry to Time entry

=A9/86400+K3
Divide cell A9 by 86400 and add to cell K3. Use $ to anchor to cell K3

Format column as time:
YYYY-MM-DD HH:MM:SS.000

Copy cell K9 and paste down column

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 Time Stamps: Absolute Mode

Using the "absolutetime" parameter in the configuration file sets the time stamp format to absolute seconds elapsed since Epoch, which is January 1, 1970. Programs such as Matlab, Octave, and R will directly import this time stamp format and automatically convert it to a standard date and time format.

3.3.3 Acceleration

The X16-1E 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 discreet counts, covers the full range of the +/-16g sensor. Therefore, the conversion factor is 65536 / 32 = **2048 counts/g**.

Table 2 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/sec^2 or 9.807 m/sec^2.

Table 2: Example Data Conversion

Raw Data (Low Gain)				Converted Data			
Time	Ax	Ay	Az	Time	Ax (g)	Ay (g)	Az (g)
0.003	799	650	-1773	05/25/2021 10:25:14.003	0.390137	0.317383	-0.865723
0.013	805	661	-1808	05/25/2021 10:25:14.013	0.393066	0.322754	-0.882813
0.023	766	687	-1844	05/25/2021 10:25:14.023	0.374023	0.335449	-0.900391
0.033	790	670	-1818	05/25/2021 10:25:14.033	0.385742	0.327148	-0.887695
0.042	801	663	-1808	05/25/2021 10:25:14.042	0.391113	0.323730	-0.882813
0.052	769	657	-1786	05/25/2021 10:25:14.052	0.375488	0.320801	-0.872070
0.062	790	683	-1795	05/25/2021 10:25:14.062	0.385742	0.333496	-0.876465
0.072	813	719	-1853	05/25/2021 10:25:14.072	0.396973	0.351074	-0.904785
0.081	824	670	-1784	05/25/2021 10:25:14.081	0.402344	0.327148	-0.871094

4 System Details

4.1 Sensor

The X16-1E uses the Analog Devices ADXL345 3-axis digital accelerometer sensor, which is based on micro-electro machined semiconductor (MEMS) technology. 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 3 lists the basic sensor and logger performance parameters. Refer to Analog Devices for detailed sensor specifications.

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 X16-1E 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.

Table 3: 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 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.

The accelerometer sensor is based on micro-electro mechanical systems (MEMS) technology and is not affected by magnetic fields. Gluing a magnet to the bottom of the plastic enclosure can facilitate easy attachment to iron surfaces and will not disturb the accelerometer sensor.

4.1.1 Sensor Special Features

The X16-1E 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 X16-1E will support sample rates of 800, 1600, and 3200 Hz but the X16-1E deactivates the oversampling and FIR filter and records the native 13-bit resolution from the sensor. However, these sample rates are not guaranteed. 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 remains 2048 counts/g.

```
; Example X16-1E Config file
; set to 800Hz
samplerate = 800
; record constantly
deadband = 0
deadbandtimeout = 0
; set file length
samplesperfile = 100000
; set logger to turn on with clock
statusindicators = normal
```

Figure 21: 800Hz Sample Configuration

4.2 Operating and Storage Conditions

The X16-1E is protected from general handling conditions by the plastic enclosure, but it is not protected from adverse environmental conditions, such as rain, sweat, splashes, and water submersion. The temperature range is limited primarily by the AA battery capabilities.

Table 4: Operating and Storage Conditions

Parameter	Value
Temperature Range (Operating/Storage)	0°F ~ 130°F (-18°C ~ 55°C)
Relative Humidity (alkaline battery)	<90%

4.3 Dimensions

The X16-1E 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 X16-1E weighs 2oz (55g) with an alkaline battery.

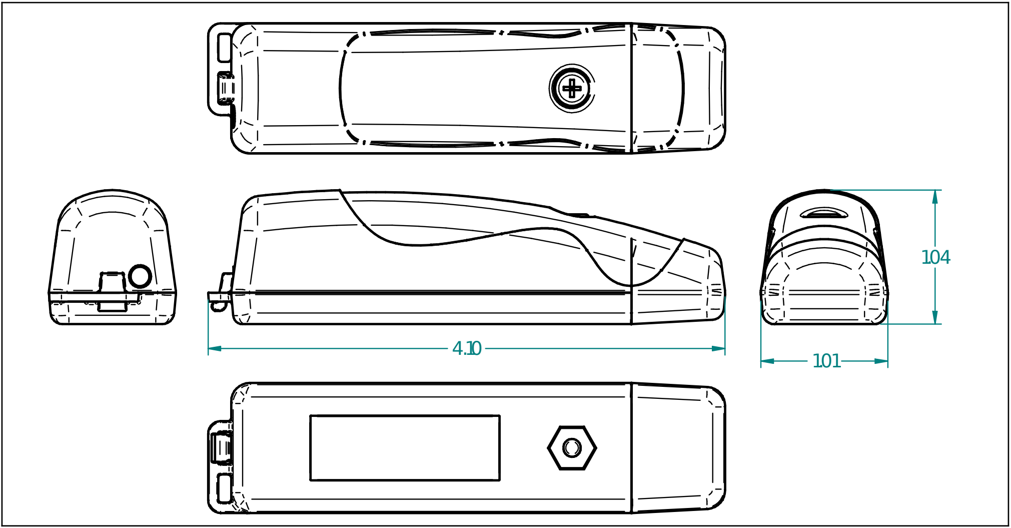


Figure 22: X16-1E Mechanical Dimensions

5 Troubleshooting

Problem	Resolution
I press the on/off button, but the logger does not appear to activate, and no LEDs blink.	Install a new battery.
	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, and 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 not corrupted.
I press the on/off button, but the logger records only for a short period of time.	Install a new battery.
	The microSD card is full, and data files must be deleted.
The logger seems to ignore the config.txt file and uses 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 blue LED blinks irregularly and/or the red LED stays lit. The logger will not mount to a PC.	The logger is unstable due to a configuration problem. Remove the microSD card and use a card reader to access the config.txt file. Edit the config.txt file such that samplerate is 400 Hz or less. Re-install the microSD card. Pull and replace the battery to reset the system. Press the on/off button, and the logger will return to normal operation.

Problem	Resolution
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. Remove and reinsert the card to ensure that the card is correctly seated into the card holder.
	The microSD card is corrupted or damaged. Reformat or replace the card.
	The on/off button could be jammed in the plastic enclosure, and, therefore, the logger is stuck in the “off” state. Check that the button moves freely and “clicks” when pressed.
	The USB connection or the extender cable (if present) could be faulty. Remove the extender cable and plug the logger into another USB port.
The start time in the data file header is incorrect.	Initialize the RTC.
The logger is stationary, but it registers 1g.	This is normal and indicates that Earth's gravity is fully operational and stable.
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. A 3-axis tumble calibration test is the best method to determine the sensor offset error. The errors are particular to the sensor and are normally consistent throughout all data sets.

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-ˈrä-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.

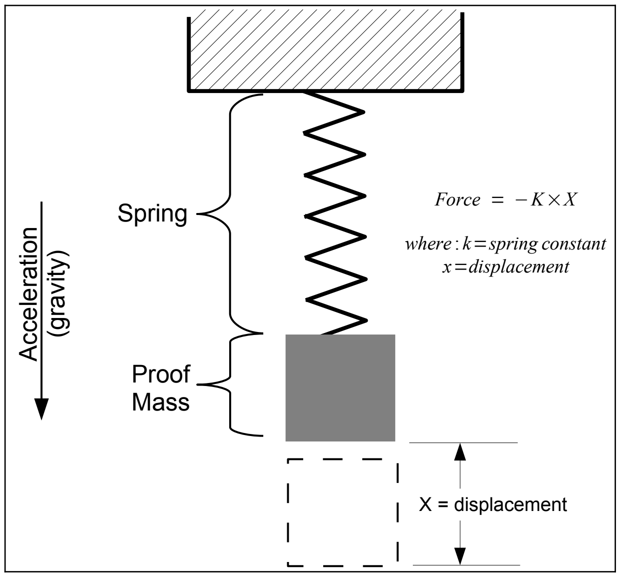
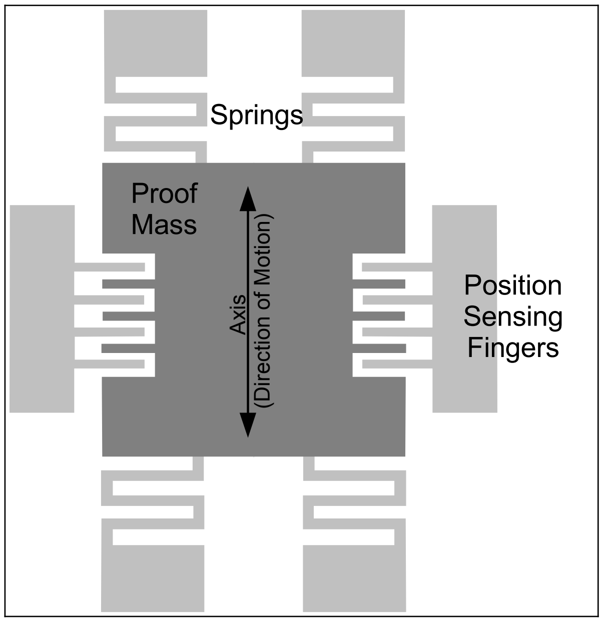


Figure 23: Spring-mass Accelerometer

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 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.



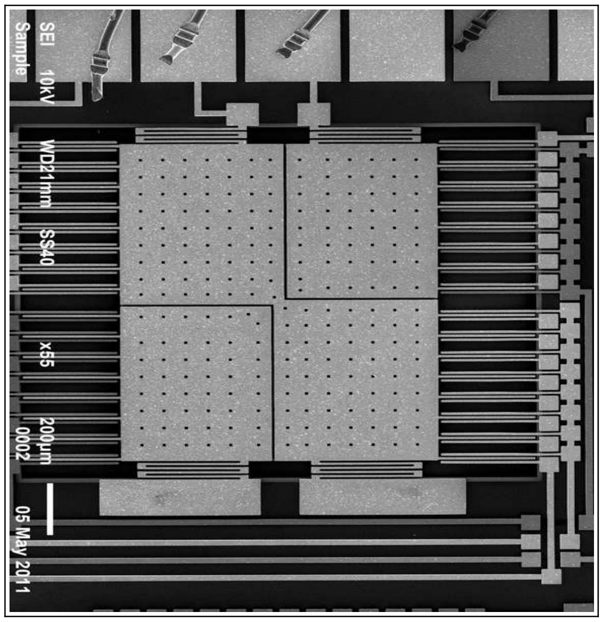


Figure 24: Simplified MEMS Accelerometer Design (L) and Actual MEMS Accelerometer (R)

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

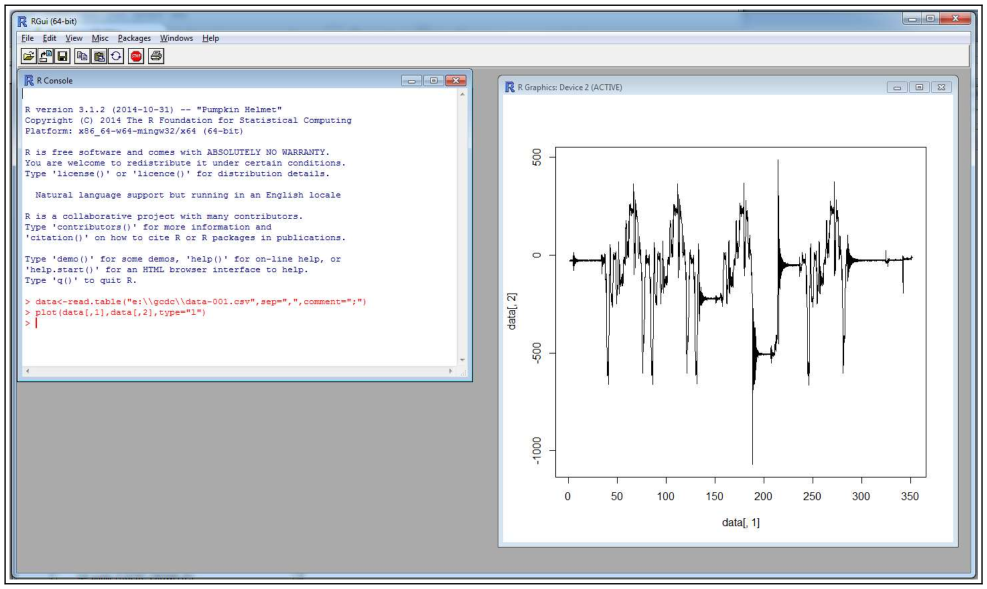
6.2.1 What is “R”

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. However, 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 all the same capabilities. However, Octave is a significantly larger download and more complicated installation than R. We favor R because it's small, easy to learn, and free.

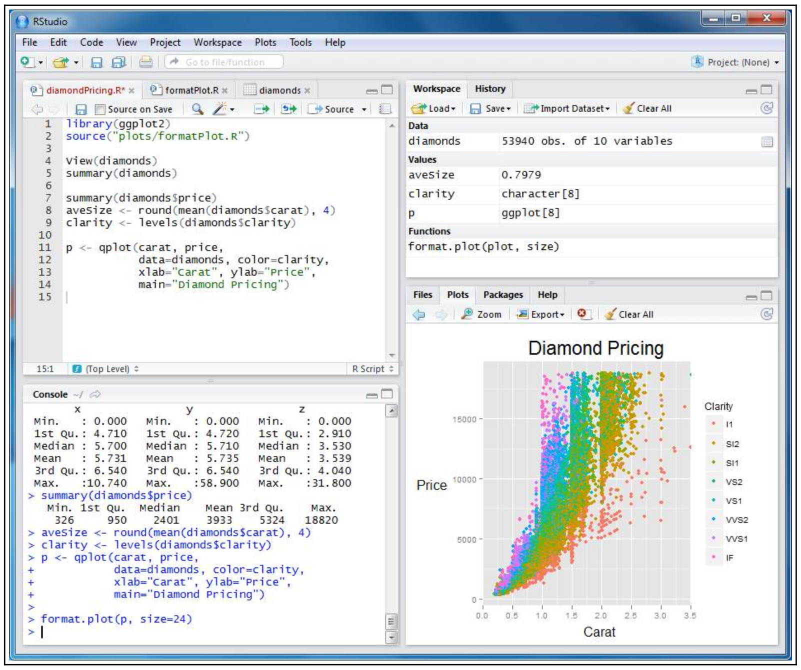


```
R version 3.1.2 (2014-10-31) -- "Pumpkin Helmet"
Copyright (C) 2014 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> data<-read.table("e:\\gcdc\\data-001.csv",sep=",",comment=";")
> plot(data[,1],data[,2],type="l")
> 
```

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 a free software package that provides a more versatile interface to the R interpreter as seen in Figure 25. RStudio is available at www.rstudio.com



```
library(ggplot2)
source("plots/formatPlot.R")

View(diamonds)
summary(diamonds)

summary(diamonds$price)
aveSize <- round(mean(diamonds$carat), 4)
clarity <- levels(diamonds$clarity)

p <- qplot(carat, price,
           data=diamonds, color=clarity,
           xlab="Carat", ylab="Price",
           main="Diamond Pricing")

```

```
       x                 y                 z
 Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
 1st Qu.: 4.710   1st Qu.: 4.720   1st Qu.: 2.910
 Median : 5.700   Median : 5.710   Median : 3.530
 Mean   : 5.731   Mean   : 5.735   Mean   : 3.539
 3rd Qu.: 6.540   3rd Qu.: 6.540   3rd Qu.: 4.040
 Max.   :10.740   Max.   :58.900   Max.   :31.800
> summary(diamonds$price)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    326     950    2401    3933    5324   18820
> aveSize <- round(mean(diamonds$carat), 4)
> clarity <- levels(diamonds$clarity)
> p <- qplot(carat, price,
+            data=diamonds, color=clarity,
+            xlab="Carat", ylab="Price",
+            main="Diamond Pricing")
>
> format.plot(p, size=24)
> |
```

```
Data
diamonds        53940 obs. of 10 variables
Values
aveSize         0.7979
clarity         character[8]
p               ggplot[8]
Functions
format.plot(plot, size)
```

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 X16-1E 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 and Ax, Ay, and 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:

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 has 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 X16-1E 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