Gulf Coast Data Concepts

www.gcdataconcepts.com

X2-5
USB Accelerometer
Data Logger

User Manual

Document Revision:	**Rev A**
Firmware Version:	**1836**
Date:	**January 29, 2020**

Table of Contents

Index of Figures

Index of Tables

1 Introduction

1.1 About This Manual

Thank you for purchasing the X2-5 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 X2-5. If the logger fails to operate as expected, please refer to the troubleshooting guide (page 23).

1.2 Document Conventions

The quick start guide in section 1.8 provides a basic summary of operation to begin using the X2-5 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 X2-5 and how to use the data.

1.4 Product Summary

The X2-5 is a compact self-recording data logger with a very low noise accelerometer sensor. 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 X2-5 appears as a standard mass storage device containing the comma delimited data files and the user setup file. The X2-5 includes an internal 500mAh lithium-polymer rechargeable battery, which will recharge using USB power.

1.5 Feature List

- Analog Devices ADXL355 digital 3-axis accelerometer

- High gain ±2g mode and low gain ±8g mode

- 20-bit resolution with internal analog and digital filters

- User selectable sample rates between 4 and 2000 Hertz

- Accurate time stamped data using Real Time Clock (RTC)

- Operates from internal 500mAh lithium-polymer rechargeable battery

- 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)

- Weight 1.3oz (48g)

- Size 1x1x4.1 inch (26x26x104 mm)
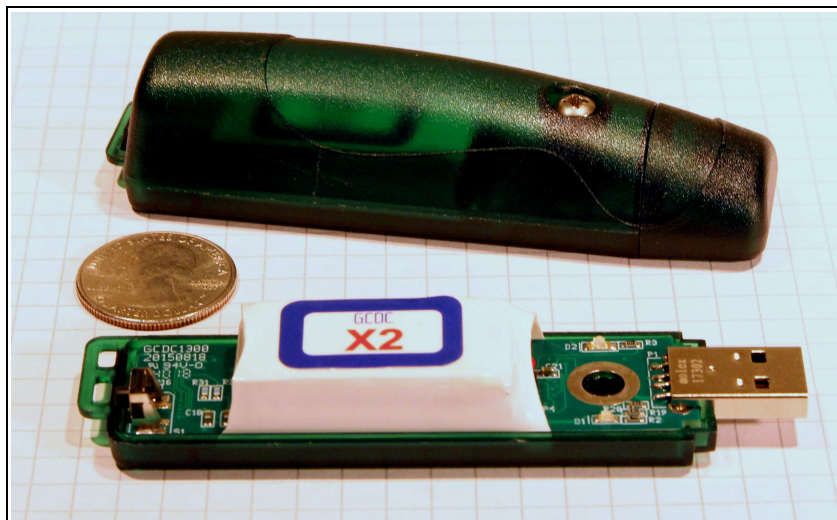
Figure 1: X2-5 Data Logger

1.6   Items Included with X2-5

1.6.1   Single Unit Purchase

The X2-5 is packaged with the logger, a USB extender cable, and a screwdriver.

Figure 2: X2-5 and Accessories

1.6.2   5 Unit Kit

A kit includes 5 X2-5 loggers, a USB extender cable, and a screwdriver.

Figure 3: 5 Unit kit of Loggers

1.7   Component Names
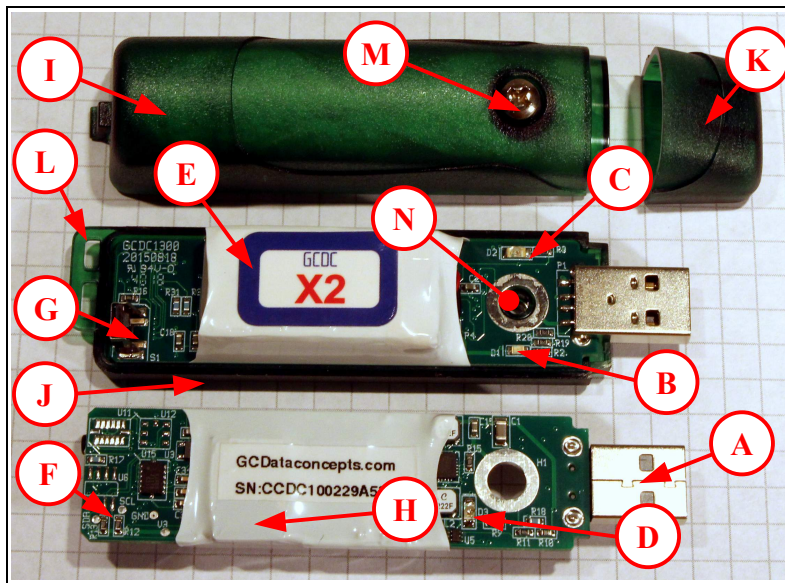
Figure 4: X2-5 Data Logger Components

A	Type-A USB connector	H	MicroSD card (under wrap)
B	Blue LED status indicator	I	Enclosure top
C	Red LED data indicator	J	Enclosure bottom
D	Red LED charge indicator	K	Enclosure cap
E	500mAh Battery Pack	L	Enclosure hinge
F	ADXL355 sensor	M	#6-32 3/4" screw
G	On/Off button	N	#6 nut
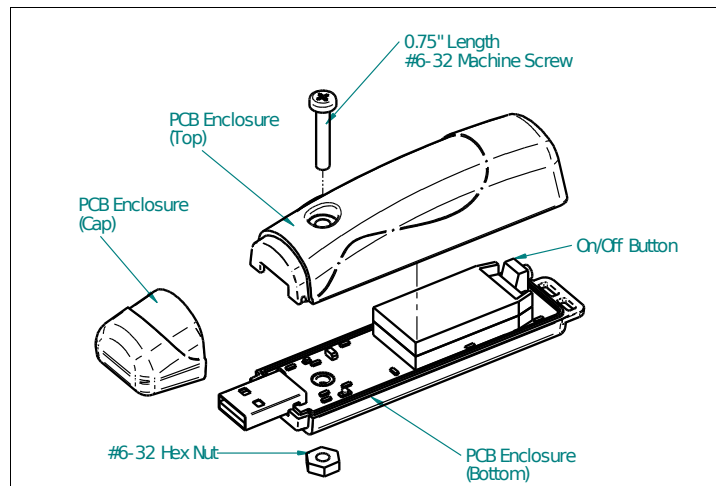
Figure 5: Exploded View of the X2-5

1.8   Quick Start Guide

The X2-5 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 X2-5. Configuration settings and mounting methods will depend on the particular application.

Step 1:   Plug the X2-5 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.
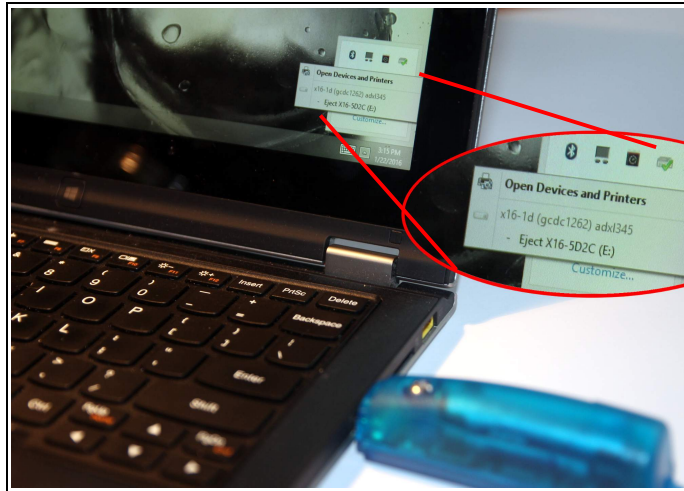
Figure 6: Connecting to PC

Step 2:   Configure the X2-5 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.
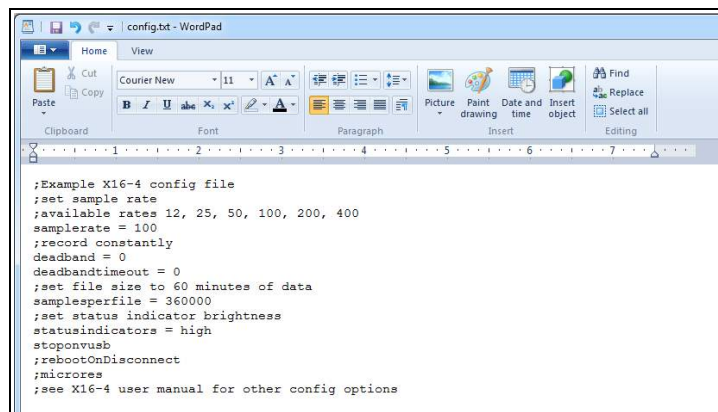
Figure 7: Editing the Config.txt File

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 X2-5 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 X2-5 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 X2-5 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 X2-5

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 X2-5 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 16384 (high gain) or 4096 (low gain) to determine g units. See section 3.3 for a complete discussion of data conversion.

2   Operation

2.1   USB Interface

The X2-5 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 X2-5 as a typical USB external memory drive. Therefore, the X2-5 will allow file transfers to the microSD card like a common USB flash drive. When connected to a PC, the X2-5 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 X2-5.

2.2   Memory Card

The X2-5 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 micoSDHC type cards.

The logger needs only the config.txt file to operate. The X2-5 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 X2-5 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 X2-5 is powered by an internal 500mAh lithium-polymer rechargeable battery pack. The internal battery management system recharges the battery when the X2-5 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.

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

> *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. The logger does not implement power saving features when connected to an external power supply so power consumption will be higher than using the internal battery.*

2.3.1   Expected Battery Life

The battery provides approximately 6 to 18 hours of operation, depending on the selected sample rate. Figure 9 illustrates the expected continuous logging time versus sample rate.
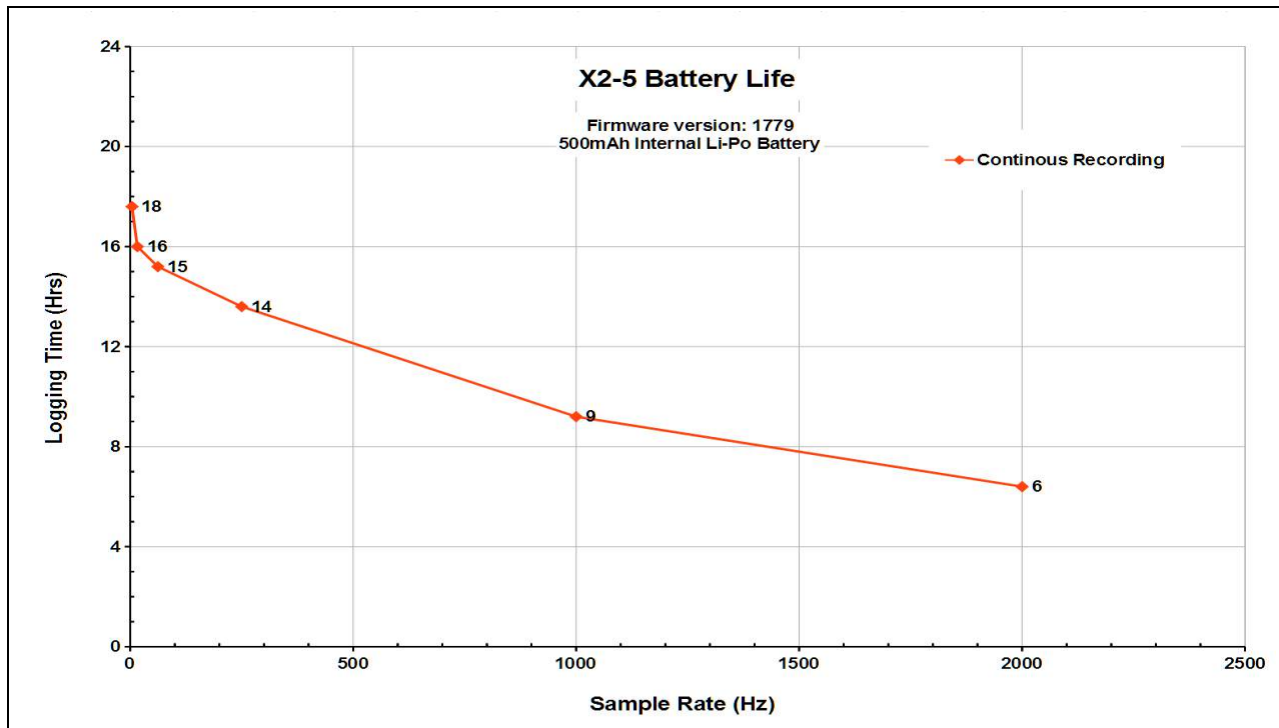

Figure 9: Expected Battery Life

2.3.2   Power Saving Strategies

Write operations to the mircoSD card is the most significant power drain to the battery. Switching the microSD card to low-power SLC type memory or very fast MLC (UHS-I or better) will increase the logger's battery life. Finding these cards can be difficult or cost prohibitive. Note that the microSD card must be formatted to FAT32. However, there are several feature configurations that will reduce power consumption and help extend the battery life.

- Use the deadband feature to reduce the recorded data. Since the microSD card is the most significant power draw, then reducing the data writes will translate into much better battery life. For example, setting the deadband=32000 will keep the logger from writing data to a file when motion is less than 0.5g. Reducing the data by 20% can increase the battery life by 30%.

- Turn off micro-resolution to reduce the number of characters written to the file, which saves about 3% battery life.

2.4   Setting The RTC

A real time clock (RTC) integrated into the X2-5 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 10 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 initialize 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".
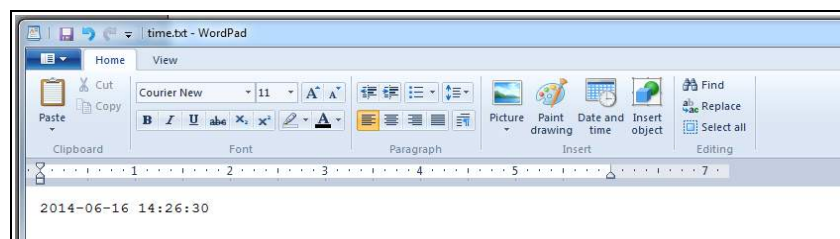
Figure 10: Example Time Entry in time.txt File

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 X2-5 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 11). 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 11: LED Status Indicators

2.6   System Configuration Options

The X2-5 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 s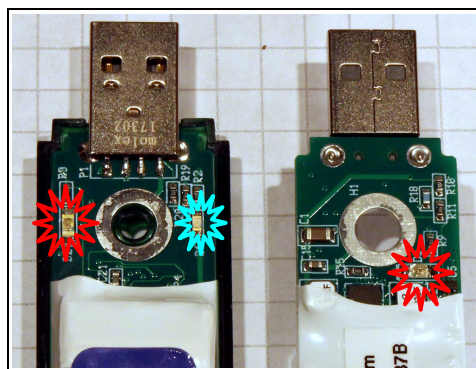etting. A line finishes with a newline character. Tags are not case sensitive. Tab and space characters are ignored. Lines starting with a semicolon (";") are treated as comments and ignored by the system. The system will use default settings if the config.txt file is not found.

Table 1: Configuration File Tags and Descriptions

Tag	Valid Settings	Default	Description
absoluteTime	-	Off	Time stamps are relative to epoch (Jan 1, 1970)
adxl355_hpf	0,1,2,3,4,5,6	Off	Sets the high pass filter cutoff
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.
dwell	An integer between 0 and 65535	1	The number of samples recorded after a deadband threshold triggered event
gain	High, low	Low	Sets the range as 2g (high gain) or 8g (low gain)
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	4, 8, 16, 31, 62, 125, 250, 500, 1000, 2000	100	Sets the rate at which data is collected and recorded to the microSD card.
starttime and stoptime	See section 2.6.11	-	Defines when to start and stop recording
stoponvusb	-	Off	Stops data logging if 5v USB power is present (see section 2.6.12)
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   absoluteTime

The absoluteTime option changes the time stamp entry to elapsed seconds relative to January 1, 1970, referred to as epoch or UNIX time. Add "absoluteTime" on a single line in the config.txt file to active this option. Absolute time stamps are more readily handled and formatted in analysis software, such as Matlab or R. The default mode keeps the time stamps relative to the start_time entry listed in the file header.

2.6.2   adxl355_hpf

The ADXL255 sensor includes an internal digital high-pass filter. By default, the filter is disabled. The high pass corner frequency, where the signal is attenuated by 50% (-3db), is related to the sample rate. The available modes are list in Table 2. For example, adxl355_hpf=2 will remove signals below 0.31 Hz when the sample rate is 500 Hz ($6.2084 \times 10^{-4} \times 500 = 0.31$). The high pass filter can be used to remove the effects of gravity from the accelerometry data. Note that the sensor will need 5-10 seconds to stabilize the filter after recording has started.

Table 2: High Pass Filter Response

adxl355_hpf	Corner Frequency (-3db)
0 (default)	disabled
1	$24.7 \times 10^{-4} \times$ SR
2	$6.2084 \times 10^{-4} \times$ SR
3	$1.5545 \times 10^{-4} \times$ SR
4	$0.3862 \times 10^{-4} \times$ SR
5	$0.0954 \times 10^{-4} \times$ SR
6	$0.0238 \times 10^{-4} \times$ SR
SR = sample rate	

A high pass filter allows signals higher than a certain frequency to pass. For example, a 100 Hz high pass filter will allow signal above 100 Hz to be recorded but signals below 100 Hz are attenuated.

2.6.3   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 12 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.
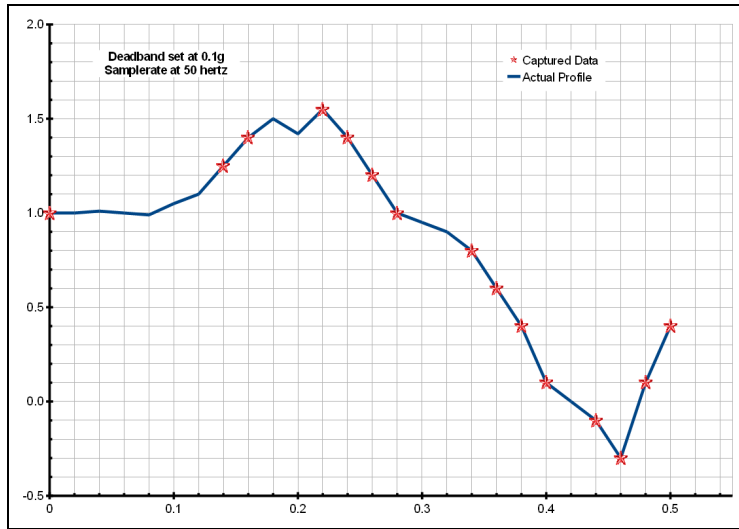
Figure 12: Graphical Illustration of the Deadband Feature

2.6.4   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.

2.6.5   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 2.7.3 for an example implementation of the deadband/dwell features.
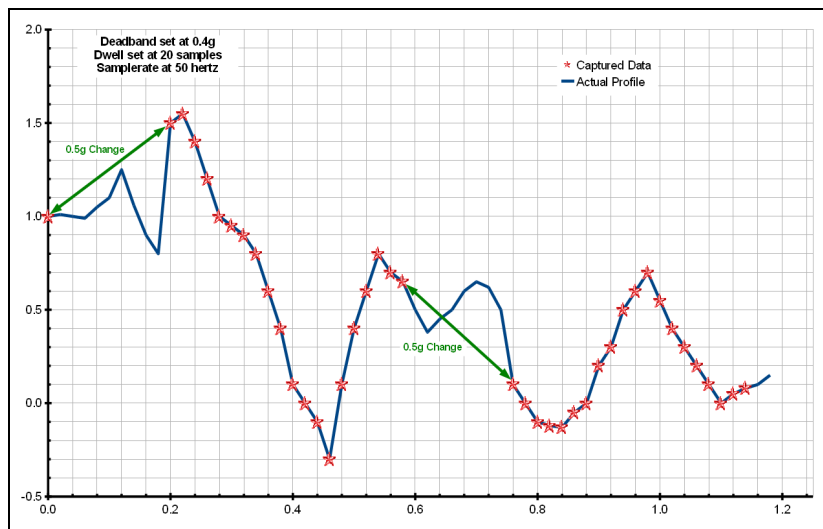
Figure 13: Graphical Illustration of the Dwell Feature

2.6.6 gain

The X2-5 provides a high and low gain modes that sets the range to ±2g or ±8g, respectively. Set "gain = high" for ±2g range or "gain = low" for ±8g range. See Section 3.3.3 for instructions regarding data conversion.

> 💡 *The gain settings may appear counter-intuitive but the "high" and "low" terms reference the sensitivity of the sensor, or conversion factors. Gain set to "high" means the sensor is in a very sensitive mode and 1g is equal to 16384 counts. The "low" gain mode reduces the sensitivity so 1g is equal to 4096 counts.*

2.6.7 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.*

2.6.8 rebootondisconnect

The X2-5 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 starttime and stoptime

The X2-5 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 X2-5 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 *

2.6.12 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. Use this feature to recharge the logger with a 5v USB
adapter but not record data during the recharging operation.

2.6.13 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 sets the sensor range to 8g and records data at 125 hertz. Deadband and
deadbandtimeout are set to zero so the logger will record constantly at the set sample rate. Each data
file is 112,500 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 X2-5 config file
;set gain to 8g
gain=low
;set sample rate
;available rates 4, 8, 16, 31, 62, 125, 250, 500, 1000, 2000
samplerate = 125
;record constantly
deadband = 0
deadbandtimeout = 0
;set file size to 15 minutes of data
samplesperfile = 112500
;set status indicator brightness
statusindicators = high
;rebootOnDisconnect
;see X2-5 user manual for other config options
```

Figure 14: Configuration File Example A

2.7.2 Example B

Example B configuration sets the range to 2g and the sample rate to 1000 Hz. The high pass filter corner frequency is 0.155 Hz so gravity is effectively removed from the data. The time stamps are set to absolute timing with maximum precision. Data is recorded continuously and each file is 5 minutes.

```
;Example X2-5 config file
;set gain to 2g
gain = high
;set sample rate
;available rates 4, 8, 16, 31, 62, 125, 250, 500, 1000, 2000
samplerate = 1000
absoluteTime
microres
;record constantly
deadband = 0
deadbandtimeout = 0
; set high pass filter to 0.16 Hz
adxl355_hpf = 3
;set file size to 5 minutes of data
samplesperfile = 300000
;set status indicator brightness
statusindicators = high
;rebootOnDisconnect
;see X2-5 user manual for other config options
```

Figure 15: Configuration File Example B

2.7.3 Example C

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 X2-5 Config file
gain = low
; set to 62Hz
samplerate = 62
; trigger at 0.1g
deadband = 410
; record 5 seconds of data
dwell = 310
;force a write every hour
deadbandtimeout = 3600
; set file length
samplesperfile = 30000
; LEDs on
statusindicators = normal
```

Figure 16: Configuration File Example C

2.7.4   Example D

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 500Hz and create 42 data files in the 3.5 hours. The sensor is set to 2g range. The micro-resolution is activated to provide the best timing precision at the 500 Hz sample rate.

```
; Example X16-4 Config file
; set sensor to 2g
gain = high
; set to 500Hz
samplerate = 500
; activate precision timing
microres
; record constantly
deadband = 0
deadbandtimeout = 0
; set file length to 5 min
samplesperfile = 150000
; set logger to turn on with clock
starttime = 30 10
stoptime = 00 14
statusindicators = normal
```

Figure 17: Configuration File Example D

3   Data Interpretation

3.1   Data Files

The X2-5 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 X2-5 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 18 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, X2-5, Analog Dev ADXL355
;Version, 1836, Build date, Dec  3 2019,  SN:CCDC100287EA8D2
;Start_time, 2020-01-29, 14:26:35.760
;Uptime, 5,sec,  Vbat, 3564, mv, EOL: 3200, mv
;ADXL355, SR,500,Hz, Range, +/-2g, Hpf,off*Sr, Sens,256000,LSB/g (typ)
;Deadband, 0, counts
;DeadbandTimeout, 0.000,sec
;Time, Ax, Ay, Az
0.001289,-425,8690,-256130
0.003289,-367,8838,-256210
0.005289,-431,8790,-256116
0.007289,-461,8780,-255994
0.009289,-369,8856,-256078
0.011289,-338,8812,-256058
0.013289,-424,8799,-256054
0.015158,-434,8807,-255888
0.017158,-421,8821,-255932
0.019158,-401,8826,-256098
0.021158,-350,8784,-256018
```

Figure 18: 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. 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 or epoch time (see section 3.3.2).

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 (";").

3.3   Data Conversion

3.3.1   Time Stamps (Default mode)

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 value 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 (",") 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. 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.
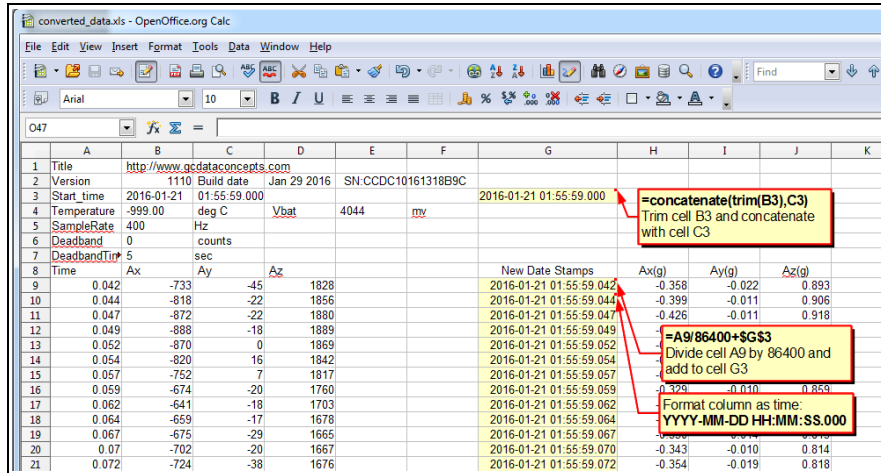
Figure 19: Time Stamp Conversion Method

3.3.2   Absolute Time Stamps

The absoluteTime feature formats the time stamps relative to January 1, 1970. This is commonly referred to as "epoch time" or "UNIX time". Use "datetime" function in Matlab to convert the absolute time stamp entry to a date-time format. In R, use the "as.POSIXct" function included with the base package.   In the case of spreadsheets, use method outlined in section 3.3.1 with January 1, 1970 as the start_time.

3.3.3   Acceleration

The X2-5 records the raw digital data from the accelerometer sensor, which is referred herein as "counts". The "counts" terminology is equivalent to "bits" or LSB. Storing the raw data in digital terms helps reduce processor load, increase sample rate capability, and avoids data errors due to floating point calculations. Divide the raw data by the appropriate conversion factors listed below in Table 2. The appropriate conversion factor is also listed in the data file header.

Table 4 lists the converted data using the example raw data in Figure 18.

Table 3: Data Conversion Factors

Gain Mode	Range	Conversion Factor to Determine "g" units
Low	±8g	64000 counts/g
High	±2g	256000 counts/g

To determine acceleration in g's, divide the raw data by the appropriate conversion factor listed in Table 3. A "g" is 32.174 ft/sec^2 or 9.807 m/sec^2.

Table 4: Example Data Conversion

Raw Data (Low Gain)				Converted Data			
Time	Ax	Ay	Az	Time	Ax (g)	Ay (g)	Az (g)
0.001289	-425	8690	-256130	01/29/2020 14:26:35.7600	-0.001660	0.033945	-1.000508
0.003289	-367	8838	-256210	01/29/2020 14:26:35.7633	-0.001434	0.034523	-1.000820
0.005289	-431	8790	-256116	01/29/2020 14:26:35.7686	-0.001684	0.034336	-1.000453
0.007289	-461	-8780	-255994	01/29/2020 14:26:35.7759	-0.001801	-0.034297	-0.999977
0.009289	-369	8856	-256078	01/29/2020 14:26:35.7852	-0.001441	0.034594	-1.000305
0.011289	-338	8812	-256058	01/29/2020 14:26:35.7965	-0.001320	0.034422	-1.000227
0.013289	-338	8812	-256054	01/29/2020 14:26:35.8097	-0.001320	0.034422	-1.000211
0.015158	-434	8807	-255888	01/29/2020 14:26:35.8249	-0.001695	0.034402	-0.999563
0.017158	-421	8821	-255932	01/29/2020 14:26:35.8421	-0.001645	0.034457	-0.999734

4   System Details

4.1   Sensor

The X2-5 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 5 lists the basic sensor and logger performance parameters but refer to Analog Devices for detailed sensor specifications.   Figure 20 illustrates the sensor orientation.

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 100 Hz may actually push data at 102 Hz. The X2-5 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. When conducting precision time based analysis, the data set should be re-sampled to ensure a consistent time period between samples.

> *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 -256000 (-1g) in the z-axis when gain is set to high.*

Table 5: Accelerometer Sensor Characteristics

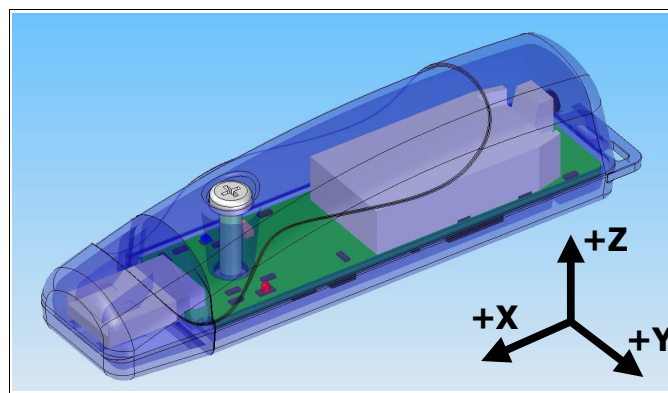| Parameter	Condition	Min	Typical	Max	Units
Sensitivity	±8g		65536		count/g
	±2g		256000		count/g
Sensitivity Deviation Due to Temperature	-40°C to +125°C		±0.01		%/°C
Nonlinearity	±2g		0.1		%FS
Noise Density	±2g	-75		+75	mg
Inter-Axis Alignment Error	±2g		25		µg/√Hz
Cross-Axis Sensitivity			±1		%

Figure 20: Sensor Orientation

4.2   Operating and Storage Conditions

The X2-5 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 6: 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 X2-5 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 X2-5 weighs 1.3oz (48g).
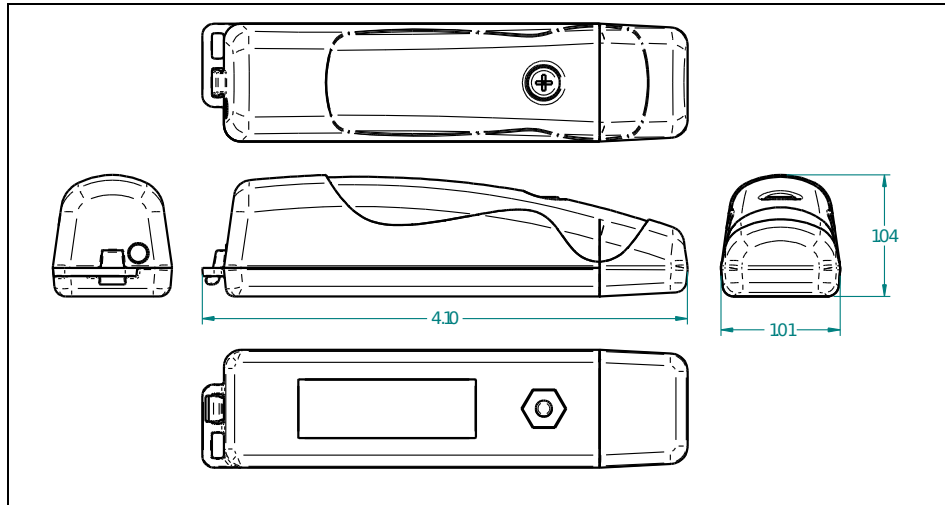

Figure 21: Enclosure Dimensions

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.
	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 regarding file extensions.
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 X2 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?	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. The RTC will need re-initialization.

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 22). 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.
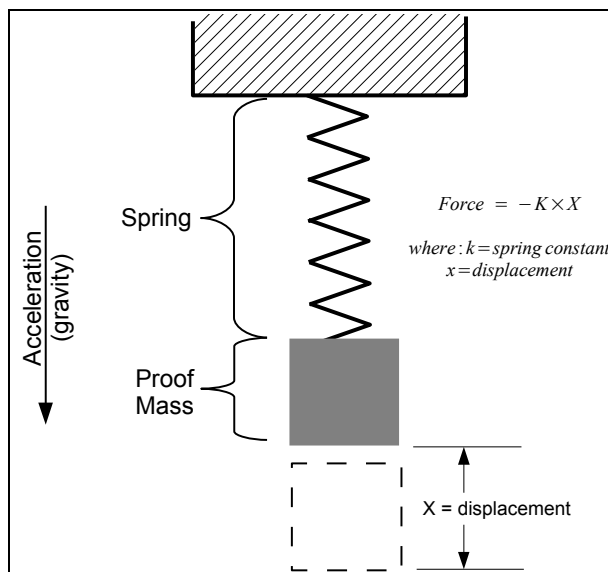
Figure 22: Spring-mass Accelerometer

Micro electro machined sensor (MEMS) technology takes the spring-mass concept and miniaturizes it onto a semiconductor chip. Figure 23 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.
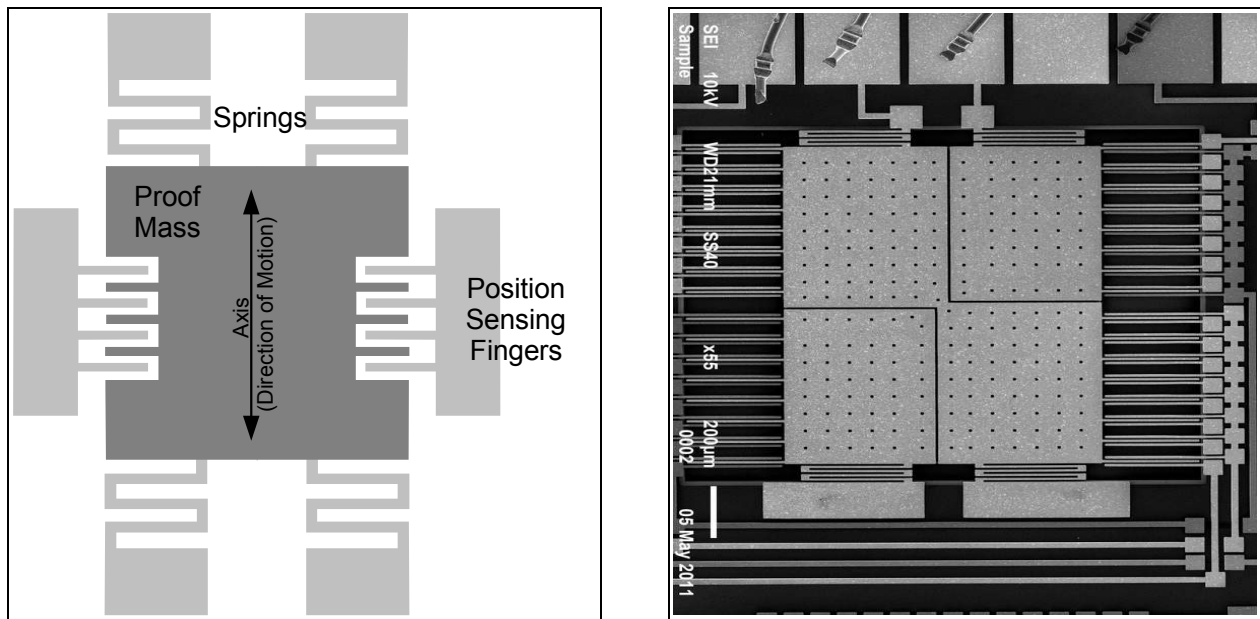
Figure 23: Simplified MEMS Accelerometer Design (L) and Actual MEMS Accelerometer (R)

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.
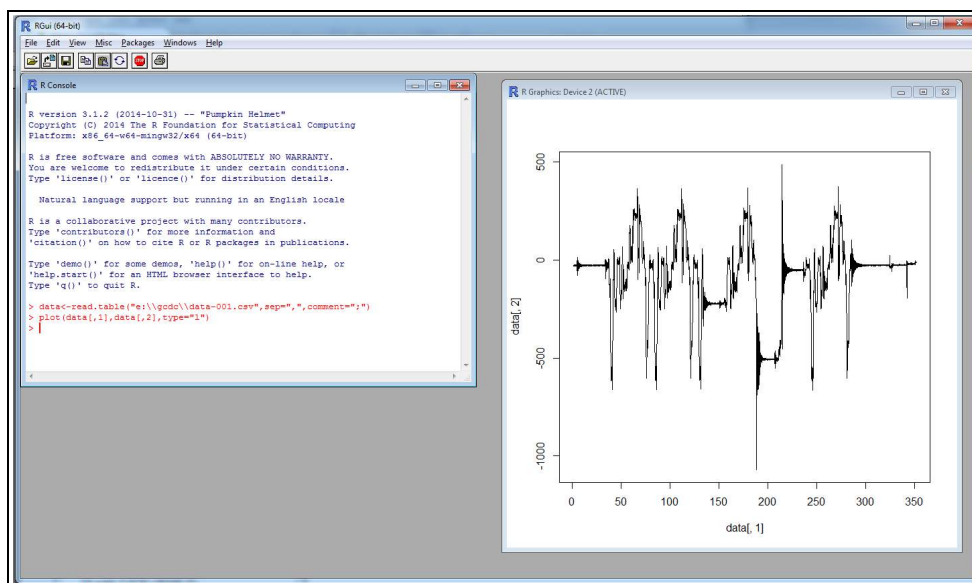

Figure 24: R Command Line Interface

R is implemented from a command line interface as seen in Figure 24. 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
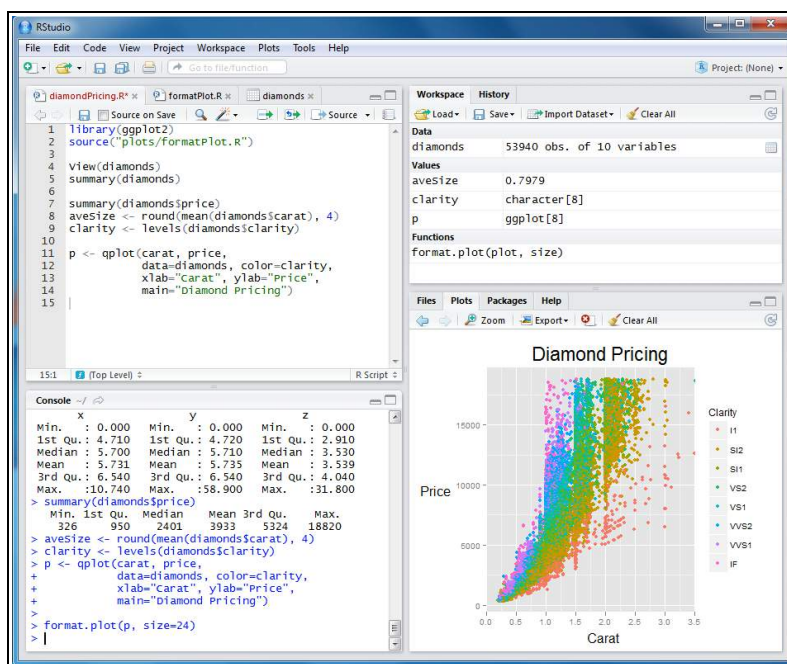

Figure 25: RStudio Interface

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 X2-5 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]/64000        ;convert the x-axis to g and assign to new vector
> dataY_g<-data[,3]/64000
> dataZ_g<-data[,4]/64000
```

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 X2-5 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