Gulf Coast Data Concepts

www.gcdataconcepts.com

MEL
Multifunction Extended Life Data Logger

User Manual

Document Revision: Rev C
Firmware Version: 2514
Date: June 20, 2023

Table of Contents

Index of Figures

Index of Tables

1 Introduction

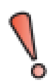
1.1 About This Manual

Thank you for purchasing the MEL 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 MEL.

1.2 Document Conventions

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

1.4 Product Summary

The Multifunction Extended Life (MEL) data logger uses precise time stamped data logging, microSD memory storage, real-time data access and USB connectivity. Available with a 16g or a 2g accelerometer sensor. The additional GPS option provides location data and precision GPS time synchronization. The MEL data logger operates as a standard USB mass storage device containing the comma delimited text data files and user setup files. The commercial standard “D” batteries (IEC R20) provides extended life operation suitable to long term data acquisition applications.

1.5 Feature List

Figure 1: MEL Data Logger

1.5.1 General Features

- Accurate time stamped data using RTC disciplined to GPS time
- Data recorded to a removable microSD card
- Easily readable comma separated text data files
- Data transfer compatible with Windows or Linux via Universal Serial Bus (USB) interface (no special software)
- System appears as USB Mass Storage Device to Windows and Linux OS’s.
- Standard replaceable “D-cell” type batteries (IEC R20)
- Weighs 500g (1.1lbs) with batteries

1.5.2 MEL-x16 (16g Accelerometer)

- Analog Devices ADXL345 3-axis 16-bit +/-16g digital accelerometer
- User selectable sample rate of 12, 25, 50, 100, 200 and 400 Hertz

1.5.3 MEL-x2 (2g Accelerometer)

- Analog Devices ADXL355 3-axis high performance 20-bit +/-2g accelerometer
- User selectable sample rates between 1 Hz and 1000Hz

1.5.4 +GPS Option

- The GPS option adds location data and precision time alignment to the MEL-x16 or MEL-x2
- u-Blox CAM-M8 GPS receiver module
  - compatible with GPS/Galileo, BeiDou, GLONASS systems
- Internal high gain patch antenna

1.6 Items Included with MEL

1.6.1 Accessories

GCDC provides a microSD card, two D-cell batteries, and a microB USB cable so the MEL data logger is operational “out of the box”. A computer is needed by the end-user to access, analyze, and view the recorded data.

1.7 Component Names

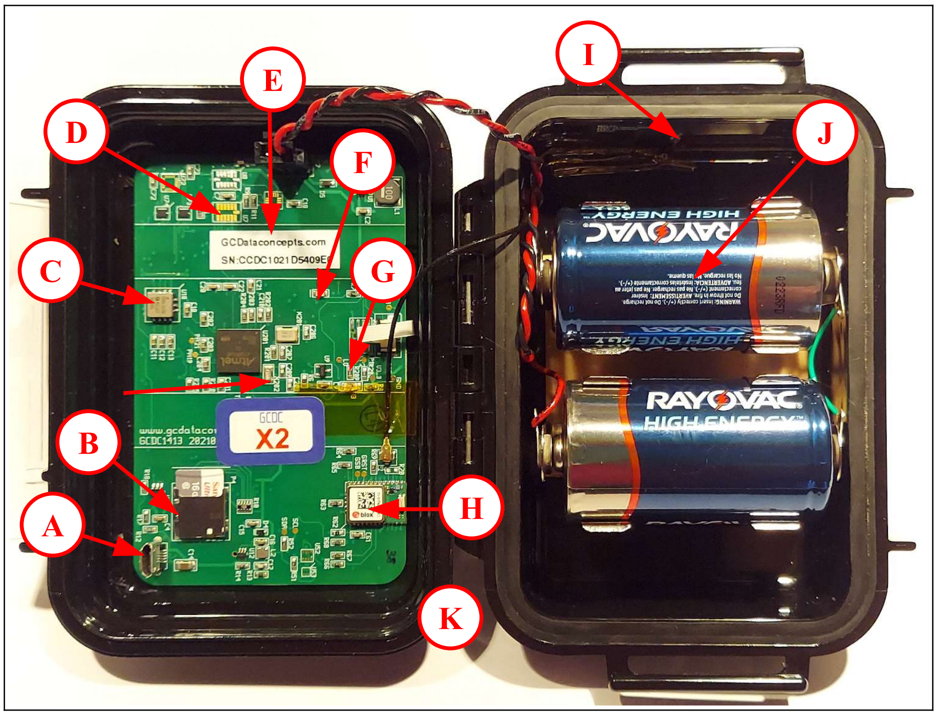


Figure 2: MEL Data Logger Components

A	microB USB connector	H	GPS receiver
B	MicroSD card	I	GPS antenna
C	ADXL355 2g sensor	J	Two D-cell batteries (2x IEC R20)
D	ADXL345 16g sensor	K	External On/Off button (not pictured)
E	Serial number unique to specific logger		
F	Blue LED status indicator *		* same function as external LED indicators located with on/off button
G	Red LED data indicator *		

1.8 Quick Start Guide

The MEL data logger is a simple, economical solution to capture extended length data sets and quickly deliver the information for analysis. The following instructions outline the steps to begin using the data logger. Configuration settings and mounting methods will depend on the particular application.

Step 1: Pull the latch to open the enclosure. Place the "D" size batteries (IEC R20) into the battery holder as indicated by the battery label.

Step 2: Plug the MEL data logger into a computer and allow the computer operating system to register the device as a Mass Storage Device.

Step 3: Configure the MEL data logger by editing the appropriate tags in the config.txt file located in the root directory of the microSD card. Use a standard text editor such as Wordpad or Notepad++ to modify and save the configuration file. See section 2.6for configuration options.

Step 4: If necessary, initialize the RTC clock by creating a time.txt file (see section 2.4). Unplug the logger and press the power button to load the time.txt file to the RTC. After initialization, the time.txt file is deleted.

Step 5: Close the enclosure and lock latch to ensure a proper seal. Be careful not to pinch the battery wires in the seal. Firmly attach the system to the target object. Depending on the g-force intensity expected, tape, tie-wraps, and clamps are suitable methods of attachment.

Step 6: Press the button located at the top of the enclosure to initiate data recording. The "Data" LED will blink as the configuration file is accessed. Then, the "Status" LED will begin to blink at a 1 second interval indicating the system is operating. If the GPS feature is enabled, the status LED will flicker while the GPS searches for a suitable location fix. The data LED will blink periodically as data is written to the microSD card.

Step 7: To stop recording, press and hold the on/off button for about 3 seconds. The LEDs will begin to blink rapidly for 2 seconds. Release the button and the MEL data logger turns off. Data recording is restarted by pressing the button again.

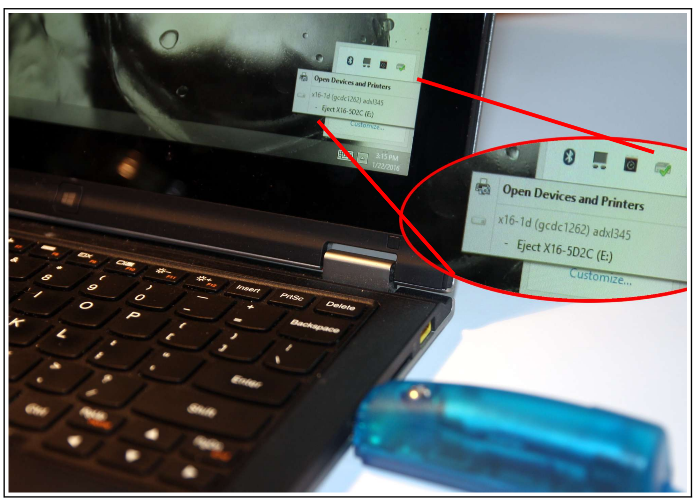


Figure 3: Connecting to PC

2 Operation

2.1 USB Interface

The MEL data logger connects to a PC using a standard microB to 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 MEL as a typical USB external memory drive. Therefore, the MEL will allow file transfers to the microSD card like a common USB flash drive. When connected to a PC, the MEL 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 MEL.

2.2 Memory Card

The MEL data logger stores data to a removable microSD flash memory card. GCDC supplies a 16GB card with the logger. The logger uses FAT32 file structure on the microSD card so the theoretical maximum memory capacity is 16TB. GCDC has tested up to 32GB UHS-1 speed (Class 10).

The logger needs only the config.txt file to operate. The MEL 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 MEL will create a folder called "GCDC", if not already present, to place the 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 MEL data logger is powered by two "D" sized batteries (IEC R20). Figure 2 illustrates the battery orientation. Operation times depends on system configuration, battery quality, and microSD card type. The RTC continues to operate from the batteries when the device is "off". Removing the batteries will reset the RTC and lose GPS location information, which will require additional time to search for an acceptable location fix when power is restored.

The batteries are not used when the device is connected to a computer USB port.

2.3.1 Expected Battery Life

Figure 4 illustrates the "preliminary" expected continuous logging time versus certain configuration conditions of the MEL-x2+GPS, which is based on insitue power measurements during operation. The MEL-x16+GPS will perform better when compared to similar configuration settings. Logging duration depends heavily on the battery quality, operating temperature, microSD card performance, and GPS behavior. These factor are difficult to predict so battery life can vary from the presented estimates.

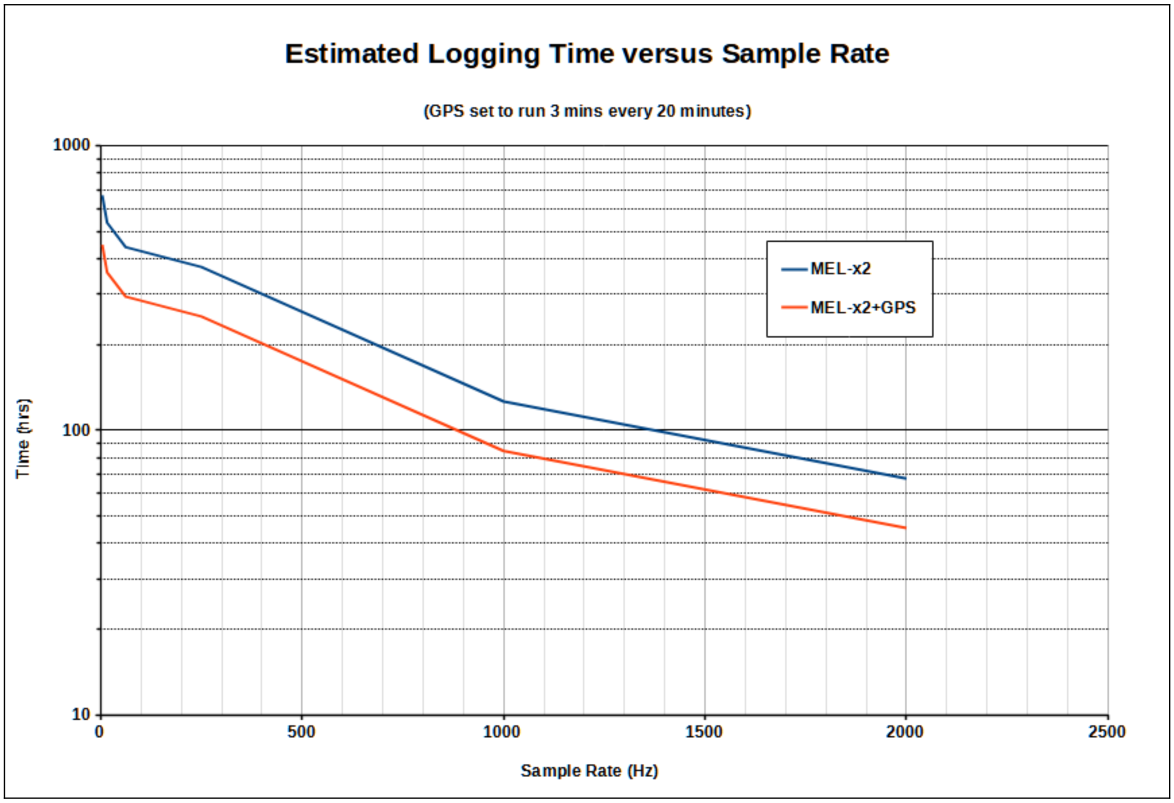


Figure 4: Expected Battery Life

2.4 Setting The RTC

2.4.1 Manual RTC Initialization

A real time clock (RTC) integrated into the MEL data logger determines the time for each line of data recorded. The RTC is initialized manually 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 5 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".

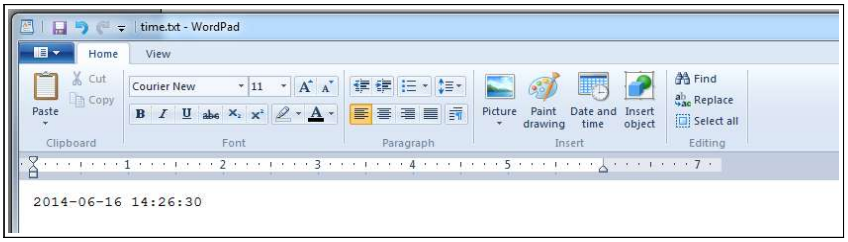


Figure 5: Example Time Entry in time.txt File

> *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.*

2.4.2 GPS synchronization of RTC

The RTC is initialized automatically to Coordinated Universal Time (UTC) based on time information acquired from the GPS (see section 2.6.4.1). Therefore, when set to "absoluteTime" mode, the data logger will record time stamps relative to UTC. The end-user must convert the time stamps manually to the appropriate local time zone. Visit https://en.wikipedia.org/wiki/Time_zone for a list of UTC offsets for common geographic locations.

2.5 Status Indicators

System status is indicated by the two LEDs located on the outside box lid (see Figure 6). The "Status" LED flickers when searching for GPS satellites and it blinks once per second to indicate a suitable GPS lock is acquired. A steady blinking "Status" LED, once per second, indicates a properly operating system. The "Data" LED blinks when data is written from the internal cache to the microSD memory card. The period at which the "Data" LED blinks depends on the logging parameters. Use the "statusindicators" tag in the configuration to deactivate the status indicators.

Figure 6: LED Status Indicators

2.6 System Configuration Options

The MEL data logger 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. A tag is followed by an equal sign ("=") and an applicable tag setting. A line finishes with a newline character (For Windows systems, Wordpad is recommended for editing the config.txt file. Notepad does not terminate lines appropriately). 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 different MEL data logger versions use the same configuration file syntax but the valid tags are particular to the sensor type. Section 2.6.1 discusses the general options and sections 2.6.2 , 2.6.3 and 2.6.4 discuss the specific options for the -x16, -x2 and GPS features.

2.6.1 General Configuration Options

Table 1 lists the common configuration options for setting up the MEL data logger.

Table 1: General Configuration Options

Tag	Valid Settings	Description
absoluteTime	-	Time stamps relative to epoch
deadBand	An integer between 0 and 32767	Defines the minimum difference between recorded sensor readings in terms of "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 65535	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.
dirName		
dwell	An integer between 0 and 65535	Defines the number of consecutive samples recorded at the set sample rate after a deadband threshold event
fileName	text	File name prefix, limited to 5 characters
minBattVoltage	Integer	Low battery cutoff level to stop logger, millivolts
rebootonDisconnect	-	The presence of this tag causes the system to start recording after disconnect from a USB port.
samplesPerFile	An integer >0 and <500000	The number of lines of data per file before a new file is created
statusIndicators	"Normal", "High", "Off"	LED status indicators can be activated with normal brightness (Normal), activated with high brightness (High), or completely deactivated (Off).
WakeUpTime	-	A set of integers to start the data logger at time(s)

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.

2.6.1.1 absoluteTime

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

2.6.1.2 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 7 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.

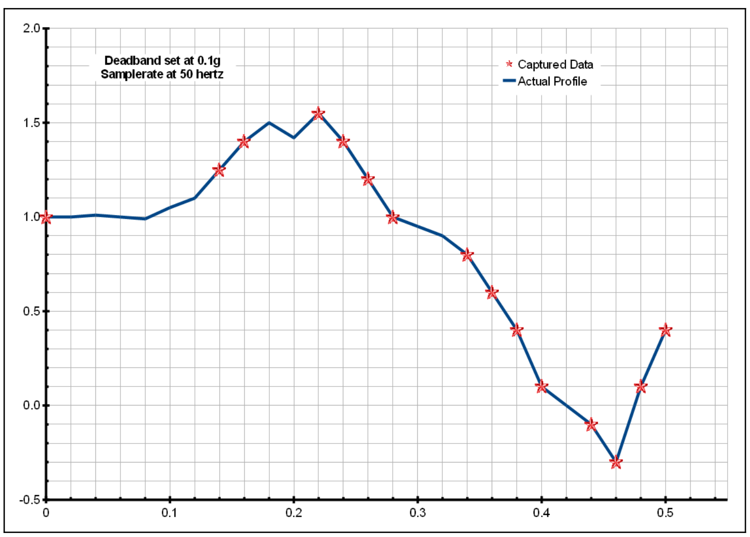


Figure 7: Graphical Illustration of the Deadband Feature

2.6.1.3 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.

2.6.1.4 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.

2.6.1.5 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 2.7.4 for an example implementation of the deadBand/dwell features.

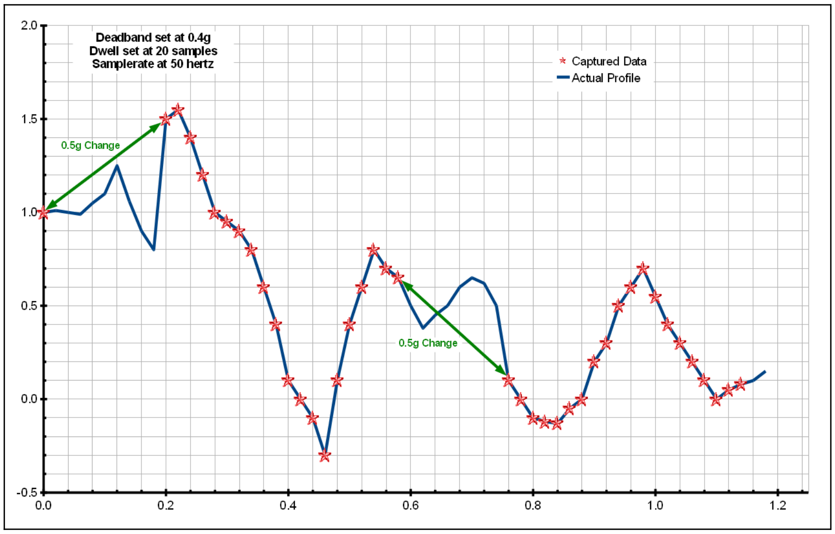


Figure 8: Graphical Illustration of the Dwell Feature

2.6.1.6 fileName

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

2.6.1.7 minBattVoltage

The logger will initiate a low-battery shutdown when the minBattVoltage is detected. By default, the minBattVoltage is set to 3200 millivolts. In the case of non-rechargeable type battery configurations, the minBattVoltage can be set to a custom value to optimize the battery usage. minBattVoltage value is millivolts.

2.6.1.8 rebootonDisconnect

The MEL 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.

2.6.1.9 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.

2.6.1.10 statusIndicators

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

2.6.1.11 WakeUpTime

WakeUpTime is an experimental feature and may not work in all configurations.

The "wakeUpTime" option configures the logger to turn on at specific times and days of month. Parameters are in order of minutes, hours, and days and are separated by spaces. 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.*

2.6.2 -x16 Options

Table 2 lists the configuration options for the MEL-x16 accelerometer feature.

Table 2: X16 Configuration Options

Tag	Valid Settings	Description
sampleRate	12, 25, 50, 100, 200, 400, 800, 1600, 3200	Sets sample rate of ADXL345 sensor
filterOff	-	Deactivates oversample and FIR filter

2.6.2.1 sampleRate

The "sampleRate" tag defines the data rate in Hertz, or samples per second. Valid sample rate settings are 12, 25, 50, 100, 200, 400, 800, 1600, and 3200 Hz. Sample rates of 800, 1600, and 3200 Hz deactivates the oversampling and FIR filter and records the native 13-bit resolution from the sensor. The data conversion factor remains 2048 counts/g. See section 4.1.1 for more information.

2.6.2.2 filterOff

Deactivates the 8X oversample and FIR filter, regardless of the sample rate setting. Accelerometer data is recorded in the native 13-bit resolution but the conversion factor is still 2048 counts/g. See section 4.1.1 for further information regarding the oversample and FIR filter.

2.6.3 -x2 Options

Table 3 lists the configuration options for the MEL-x2 accelerometer feature.

Table 3: X2 Configuration Options

Tag	Valid Settings	Description
sampleRate	1-16, 3.9, 7.8, 15.6, 31.2, 62.5, 125, 250, 500, 1000	Sets sample rate of ADXL355 sensor
gain	"high" or "low"	Sets the range to +/-2g or +/-8g
adxl355_hpf	1,2,3,4,5,6	Sets the high pass filter cutoff frequency

2.6.3.1 sampleRate

The "sampleRate" tag defines the data rate in Hertz, or samples per second. Valid settings range from 1 to 16 Hz in integer increments and then 3.9, 7.8, 15.6, 31.2, 62.5, 125, 250, 500, 1000. The settings between 1 and 16 Hz are disciplined by the logger's timer, which is synchronized to the GPS time. The settings from 3.9 Hz to 1000 Hz are controlled by the ADXL355 sensor clock, which can vary slightly.

2.6.3.2 gain

The MEL-x2 accelerometer sensor provides a high and low gain mode that sets the range to ±2g or

±8g, respectively. Set "gain = high" for ±2g range or "gain = low" for ±8g range. See Section 3.3.4 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 256000 counts. The "low" gain mode reduces the sensitivity so 1g is equal to 64000 counts.

2.6.3.3 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 4. 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 4: 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.

2.6.4 +GPS Options

Table 5 lists the configuration options for the GPS feature.

Table 5: GPS Configuration Options

Tag	Valid Settings	Description
gps_clockThreshold	Floating point number	maximum clock error between RTC and GPS
gps_dynamicModel	0, 1, 2, 3, 4, 5, 6, 7, 8	Motion model for GPS navigation engine
gps_minLock	Floating point number	Minimum location error
gps_noBad	-	Records all GPS location samples
gps_numSamples	Integer number	Number of location samples to record per powerOn event
gpsOn	-	Turns the GPS module on
gps_powerOnTime	Integer number	The number of seconds the GPS module will be activate after a powerOn event
gps_powerPeriod	Integer number	The number of seconds between powerOn events
gps_sampleInterval	Floating point number	Defines the GPS sample rate

2.6.4.1 gps_clockThreshold

"gps_clockThreshold" defines the acceptable error band, in milliseconds, between the RTC and the GPS time of day. The GPS time of day information is received every 5-10 seconds so the GPS must be recording acceptable location data for ~10 seconds before the clock error is checked. If the clock correction feature is not needed, then set the gps_clockThreshold to a very large number, such as 3600.

> *Note that the clock correction process will initialize the RTC to GPS time, which is formated to Coordinated Universal Time (UTC). A diagnostic comment will occur in the data file noting when this correction occurred. A list of UTC offsets for common geographic locations is available at https://en.wikipedia.org/wiki/Time_zone*

2.6.4.2 gps_dynamicModel

The GPS module uses several dynamic models to help calculate subsequent location samples. Choose an appropriate model depending on the expected motion of the logger. There are 8 different modes:

Table 6: Supported Dynamic Modes for GPS

gps_DynamicModel Value	Mode	Description
0	Portable	Applications with low acceleration
1	Stationary	Stationary applications where velocity is 0
2	Pedestrian	Low acceleration and speed
3	Automotive	Dynamics of a car, assumes no vertical acceleration
4	At Sea	Assumes zero vertical velocity, sea level
5	Airborne <1g	Higher dynamic range and greater vertical acceleration
6	Airborne <2g	Typical airborne applications
7	Airborne <4g	Extreme dynamic environments

2.6.4.3 gps_minLock

"gps_minLock" sets the minimum Horizontial Dilution of Precision (HDOP) error, in meters, required to define an acceptable GPS location sample. HDOP is basically the error bubble around the location sample. The number of visible satellites and the data collection time affects the HDOP. Objects that block the view of the sky, such as trees and buildings, will increase the HDOP.

2.6.4.4 gps_noBad

The "gps_noBad" option makes the logger record all GPS location samples regardless of the hdop setting. This is good for diagnostic purposes to determine how quickly the location error focuses down to the hdop threshold.

2.6.4.5 gps_numSamples

"gps_numsamples" defines the number of samples to record that meet the gps_minLock criteria.

2.6.4.6 gpsOn

"gpsOn =1" turns the GPS module on. "gpsOn=0" deactivates the GPS module and saves power (no GPS data recorded).

2.6.4.7 gps_powerOnTime

"gps_powerOnTime" defines the number of seconds that the GPS module will be active after turning on from the gps_powerPeriod option. The GPS module will search for location data and the logger will record acceptable location samples according to the gps_minLock criteria. Once the gps_powerOnTime is complete, the GPS module is turned off to save power.

2.6.4.8 gps_powerPeriod

"gps_powerPeriod" defines the number of seconds between GPS power-on events. The gps_powerPeriod must be larger than the gps_powerOnTime value. The ephemeris, almanac, last position, and time of day are stored between events to reduce the time to location fix of the subsequent power on event.

2.6.4.9 gps_sampleInterval

"gps_sampleInterval" defines number of seconds between GPS location data samples, which is the GPS sample rate. The interval value is a floating point number. The fastest rate is 0.1000 seconds (10 Hz) and the slowest can be very large, such as 10080 seconds (weekly).

2.7 Example Configuration Files

The MEL data logger supports a very wide range of configuration options. The following sections break-down the config.txt file into manageable segments for discussion purposes. Combining Figure 9, Figure 10 and Figure 12 or Figure 9, Figure 11 and Figure 12 creates a complete config.txt file for the MEL data logger.

2.7.1 General Configuration Options

Figure 9 shows the general settings for the logger, including options such as file size, directory name, file name, time stamps, etc. The files will use "aero" as the prefix and the files will be stored in a directory named "data". The deadband is set to zero so there is no accelerometer event trigger. The time stamps are relative to epoch, which will be corrected to UTC epoch when the GPS locks location. The logger LEDs are active and the logger will start when it is removed from the PC.

```
; FILE PARAMETERS
samplesperfile = 180000
; custom directory name
dirname = /DATA
; custom file prefix
filename = aero
; EVENT DETECTOR PARAMETERS
deadband = 0
DeadBandTimeout = 5
dwell=100
; set timestamp to UTC seconds since Jan 1, 1970 instead of time since start of file
absoluteTime
;control brightness of LEDs, values are 'off' or 'high'
statusindicators = high
;uncomment following line to activate logger upon disconnect from USB
rebootOnDisconnect
```

Figure 9: Example of General Configuration Options

2.7.2 -x16 Configuration Options

The configuration in Figure 10 sets the sample rate to 200 Hz. The time stamps are set to absolute timing. Data is recorded continuously and each file is 5 minutes.

```
;Example MEL-x16 config file
;set sample rate
samplerate = 200
absoluteTime
;record constantly
deadband = 0
deadbandtimeout = 0
;set file size to 5 minutes of data
samplesperfile = 300000
;set status indicator brightness
statusindicators = high
;rebootOnDisconnect
;see MEL user manual for other config options
```

Figure 10: Example -x16 Configuration Settings

2.7.3 -x2 Configuration Options

The following configuration sets the ADXL355 sensor range to 8g and records data at 62.5 hertz. Deadband and deadbandtimeout are set to zero so the logger will record constantly at the set sample rate. Each data file is 56,250 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 MEL-x2+GPS config file
;set gain to 8g
gain=low
;set sample rate
samplerate = 62.5
;record constantly
deadband = 0
deadbandtimeout = 0
;set file size to 15 minutes of data
samplesperfile = 56250
;set status indicator brightness
statusindicators = high
;rebootOnDisconnect
;see MEL user manual for other config options
```

Figure 11: Example -x2 Configuration Settings

2.7.4 GPS Configuration Options

The example in Figure 12 sets the GPS to turn on every 5 minutes (300 seconds) and search for location data for 50 seconds. It is using model 3 and capturing location every 1 second. The GPS has 50 seconds to find a suitable location with an hdop less than 40 meters. If a suitable location can be found within 40 seconds, then it has enough time to record 10 locations before turning off. Typically, the time needed to find an acceptable location improves with each power-up period. The GPS will eventually have enough time to get the 10 location samples and update the RTC.

```
; GPS PARAMETERS
gpsOn = 1
; minimum hdop required, in meters
gps_MinLock = 40
; discipline RTC with GPS time
gps_clockThreshold = 0.010
;gps_update interval, in seconds
gps_sampleinterval = 1.0
gps_dynamicModel = 3
; discard gps readings that exceed the minimum HDOP
;gps_nobad
; used in on/off mode number of good samples to collect before turning off
gps_numSamples = 10
; used in on/off mode, time in seconds device will spend trying to collect good samples
gps_powerOnTime = 50
; used in on/off mode, time, in seconds between on events
gps_powerPeriod = 300
```

Figure 12: Example GPS Configuration Settings

3 Data Interpretation

3.1 Data Files

The MEL 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 MEL 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 13 represents an example data file created by the MEL-x2+GPS logger.

```
;Title, http://www.gcdataconcepts.com, X2-6, Analog Dev ADXL355, GPS
;Version, 2514, Build date, Jun 13 2023,  SN:CCDC1021D5409E0
;Start_time, 2007-01-01, 00:01:05.341
;Uptime, 6,sec,  Vbat, 2264, mv, EOL, 1800, mv
;Deadband, 0, counts
;DeadbandTimeout, 5.000,sec
;ADXL355, SR,1,Hz, Range, +/-2g, Hpf,off*Sr, Sens,256000,LSB/g (typ)
;BMP384, SI, 0.000,sec, Units, Pa, mdegC
;Alt Trigger disabled
;CAM_M8 Gps, SR,1,Hz
;Gps Sats, TOW, 1000, ver, 1, numSat, 0
;, gnssId, svId, cno, elev, azmith, prRes, flags,inUse
;Time, Ax, Ay, Az, TOW, Lat,Lon, Height(m), MSL(m), hdop(m), vdop(m)
1167609852.002731,226096,-88115,46925
1167609853.002830,205622,-102666,12432
1167609854.002929,212564,-5637,-152842
1167609855.003028,208580,-58474,-140828
1167609856.003127,206656,-59723,-145012
1167609857.003226,204354,-64773,-146860
1167609858.003266,227116,-115432,-8197, 411730.000, 30.2957220,-89.3669477, -69.409,-41.922, 27.478,141.470
1167609859.003357,206295,6643,-164325
1167609860.003448,208722,3772,-156107
1167609861.003539,212280,2343,-148518
1167609862.003630,214120,3556,-147563
1167609863.003721,207400,-58019,-143397
1167609864.003812,208934,-61975,-139738
1167609865.003903,208370,-59809,-140637
1167609866.003937,227615,28,-122988, 411738.000, 30.2957194,-89.3668899, -65.174,-37.687, 25.038,119.888
1167609867.004020,227281,2598,-123695
1167609868.004103,226128,2574,-125831
1167609869.004186,226615,-1186,-124719
```

Figure 13: Example MEL-x2+GPS Data File

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.

A short gap in data may occur between sequential files as data is purged from the cache and a new file is allocated. The time stamps between the two file will indicate the gap.

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.

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. The file header includes information about the logger type, firmware version, sensor configuration, file start time, sample rate, sensor range, sensor units. The file header is followed by the data samples. Each sample contains a time stamp entry followed by the sensor output readings and GPS data. The time entry is seconds elapsed from the start time recorded in the header (default mode) or relative to Jan 1, 1970 UTC (absoluteTime mode).

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

Sensor type, modes, sample rates, units, and conversion factors are listed in the data file header.

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

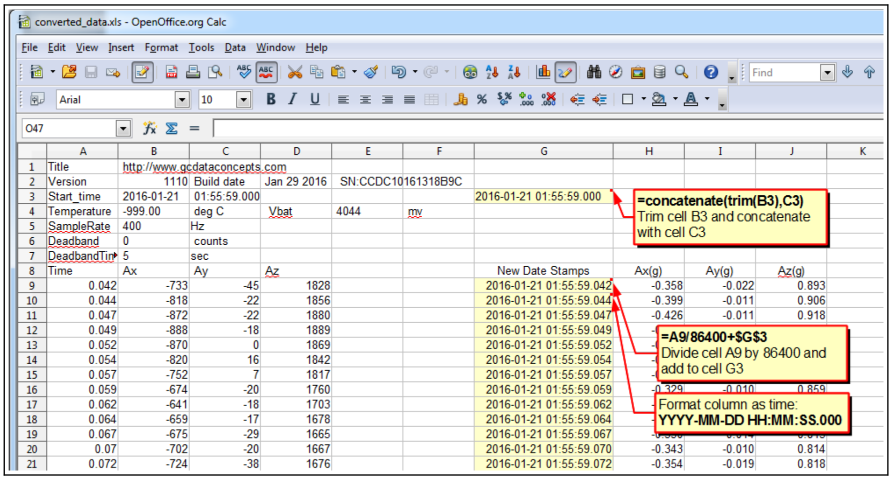


Figure 14: Time Stamp Conversion Method

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 -x16 Acceleration

The MEL-x16 records the raw digital data from the ADXL345 accelerometer sensor. This helps reduce processor load, increase sample rate capability, and avoids data errors due to floating point calculations. In the 16-bit mode, 65536 discreet counts cover the full range of the +/-16g sensor. Therefore, the conversion factor is 65536 / 32 = **2048 counts/g**. In the 13-bit mode, the data word is right padded to create a 16-bit output so the conversion remains 2048 counts/g. See section 4.1.1 for further description of MEL-x16 sensor performance and data collection methods.

3.3.4 -x2 Acceleration

The MEL-x2 data logger records the raw digital data from the ADXL355 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 7. The appropriate sensor settings and conversion factor are also listed in the data file header.

Table 7: Data Conversion Factors

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

3.3.5 +GPS Data

GPS data is appended to the sample row after the barometric temperature reading. The GPS data includes 7 samples of information in order of TOW, Lat, Lon, Height, MSL, hdop, and vdop. These values are defined as follows:

TOW – Time of Week. The TOW count is a value ranging from 0 to 403,199 representing the number of 1.5 second periods elapsed since the last Sunday 12:00AM. This value can be used to verify the more precise time stamp value generated by the logger.

Lat – Latitude. Lat is a coordinate that specifies the north–south position of a point on the surface of the Earth as an angle that ranges from –90° at the south pole to 90° at the north pole.

Lon – Longitude. Lon is a geographic coordinate that specifies the east–west position of a point on the surface of the Earth expressed in degrees, whereas 0° denotes the International Reference Meridian.

Height(m) – Height is the distance, in meters, above the Earth assuming a ellipsoid shape. This value is based on the GPS satellites and not the IMU-GPS internal barometric pressure sensor.

MSL(m) – Mean Sea Level. MSL is the distance, in meters, above the Earth using the MSL model of the Earth. This value is based on the GPS satellites and not the IMU-GPS internal barometric pressure sensor.

hdop(m) – Horizontal Dilution of Precision. HDOP is the 2D error around the latitude/longitude location and is expressed in meters. Lower values are better and indicate higher precision.

vdop(m) - Vertical Dilution of Precision. HDOP is the 2D error around the latitude/longitude location and is expressed in meters. The VDOP values can be rather large and vary greatly as compared to the HDOP since the VDOP is based on satellite triangulation instead of barometric pressure.

4 System Details

4.1 Sensors

The MEL digital sensors "push" 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 MEL 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.

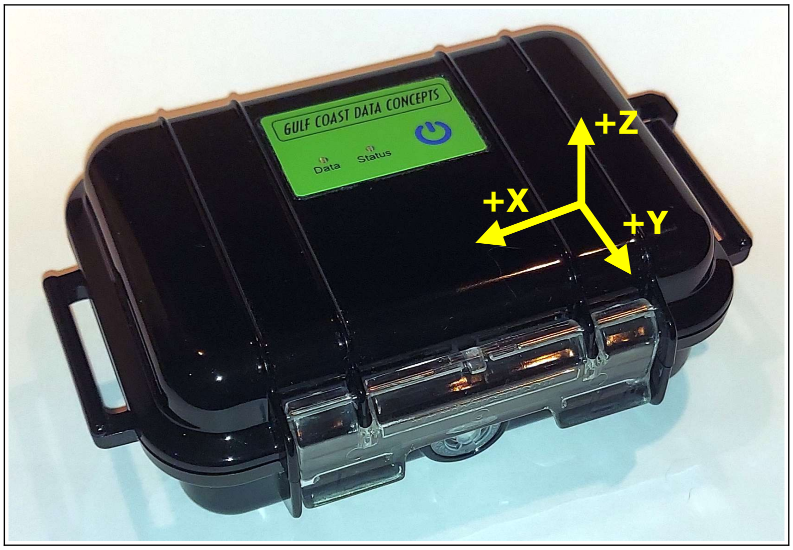


Figure 15: Sensor Orientation

4.1.1 16g ADXL345 Accelerometer

The MEL-x16 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. Table 8 lists the basic sensor and logger performance parameters. Refer to Analog Devices for detailed sensor specifications.

The MEL-x16 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 ADXL345 sensor to stream at 3200 Hz, which is the maximum capability of the sensor. 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 MEL-x16 will support sample rates of 800, 1600, and 3200 Hz but the logger 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.

Table 8: -x16 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		%

4.1.2 2g ADXL355 Accelerometer

Table 9: -x2 Accelerometer Sensor Characteristics

Parameter	Condition	Min	Typical	Max	Units
Sensitivity	±8g		65000		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		%

4.2 Operating and Storage Conditions

The MEL 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, which begins to degrade at about 40°F (4°C).

Table 10: Operating and Storage Conditions

Parameter	Value
Temperature Range (Operating)	40°F ~ 130°F (4°C ~ 55°C)
Temperature Range (Storage)	-5°F ~ 80°F (-20°C ~ 25°C)
Relative Humidity (Operating and Storage)	<90%

4.3 Mechanical

The MEL is packaged in a water resistant, crush proof, and dust proof enclosure. The MEL data logger weighs 500g (1.1 lbs) with two alkaline batteries.

4.3.1 Dimensions

Figure 16: Enclosure Dimensions

5 Software

The MEL logger records data to comma delimited text files and uses text based files for configuration settings. Therefore, no special software is required to utilize the MEL. For data analysis, Gulf Coast Data Concepts recommends using a commercial or open source mathematics package, such as MatLab, Microsoft Excel, OpenOffice Calc, Octave, R, or similar applications..

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 17). 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.

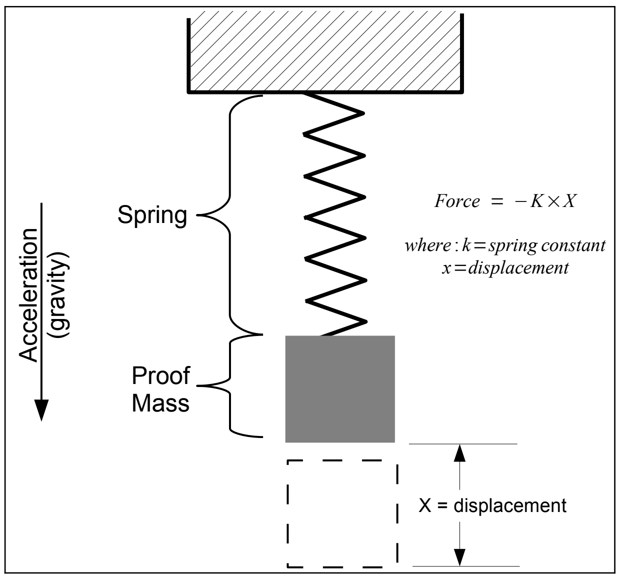


Figure 17: Spring-mass Accelerometer

Micro electro machined sensor (MEMS) technology takes the spring-mass concept and miniaturizes it onto a semiconductor chip. Figure 18 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.

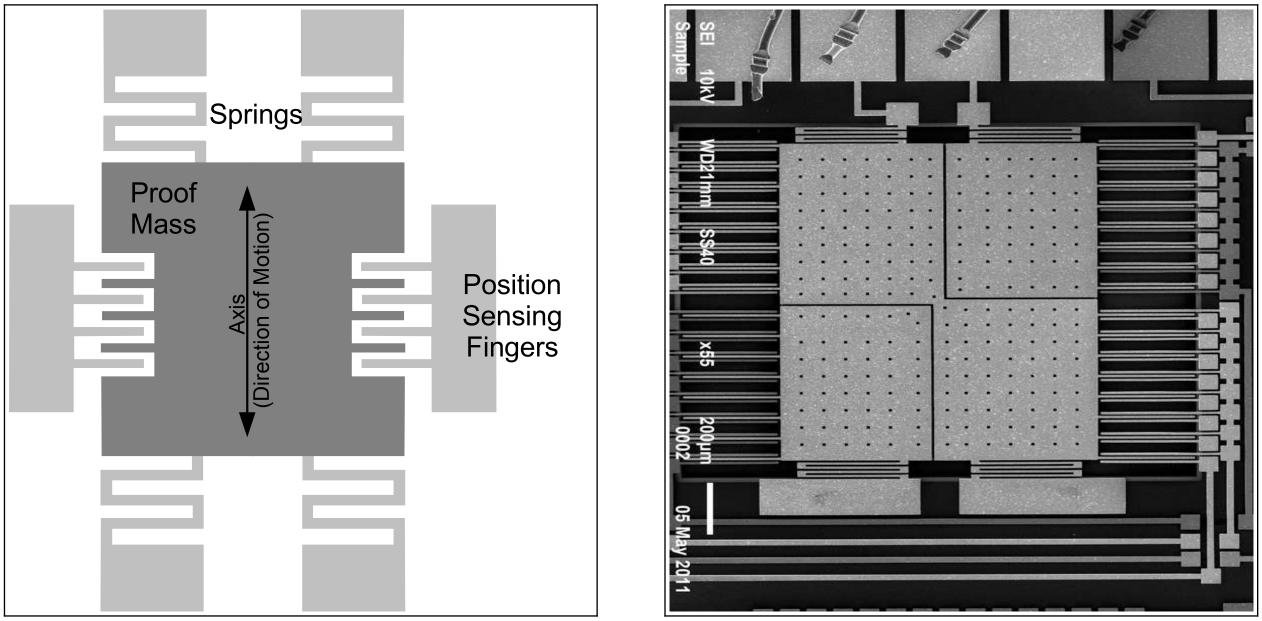


Figure 18: Simplified MEMS Accelerometer Design (L) and Actual MEMS Accelerometer (R)

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.

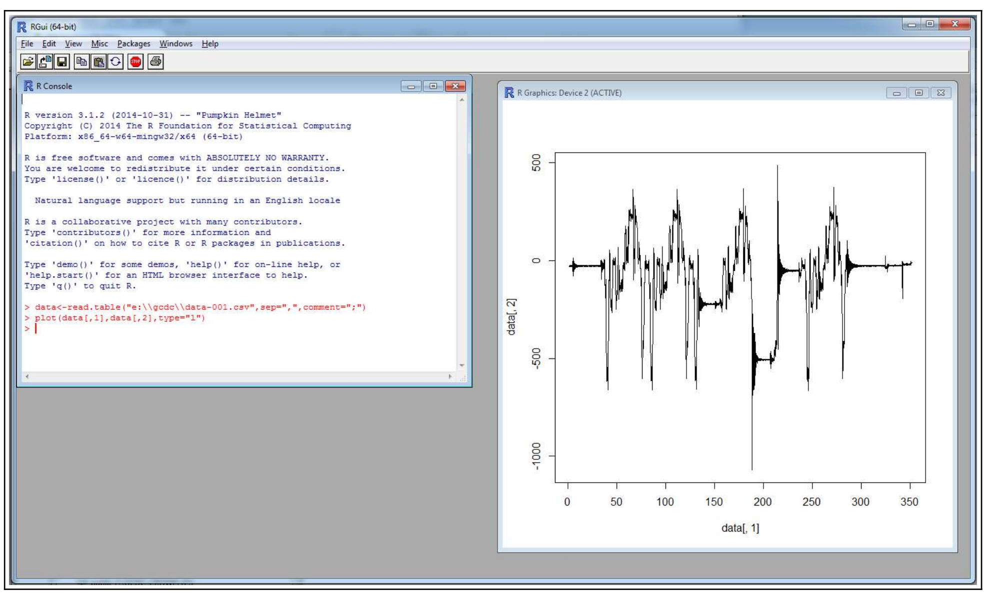


Figure 19: R Command Line Interface

R is implemented from a command line interface as seen in Figure 19. 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

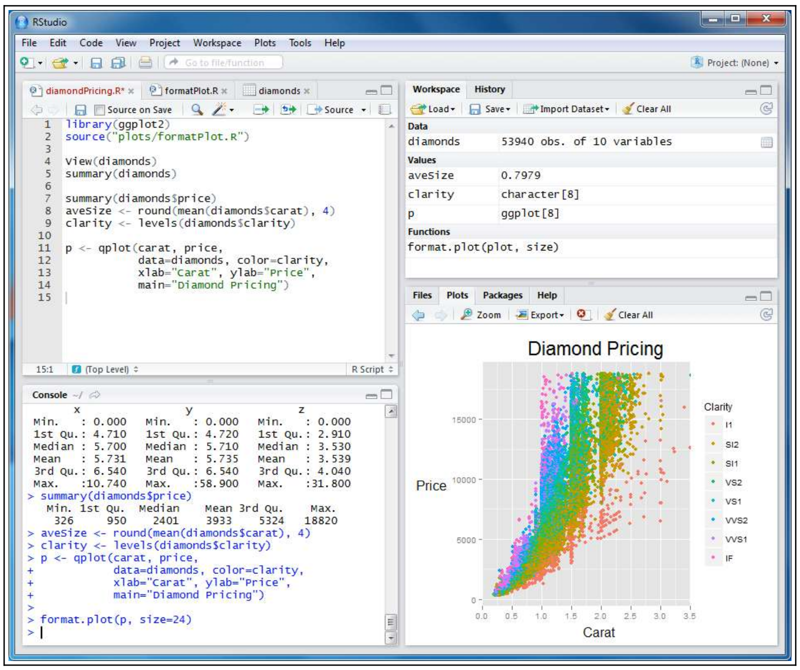


Figure 20: RStudio Interface

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 MEL 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 MEL 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