



# IMU-GPS Data Logger

## User Manual

**Document Revision:**  
**Firmware Version:**  
**Date:**

**Rev New**  
**2476 and newer**  
**July 3, 2023**



# Table of Contents

1	Introduction.....	1
1.1	About This Manual.....	1
1.2	Document Conventions.....	1
1.3	Appendix.....	1
1.4	Product Summary.....	2
1.5	Feature List.....	2
1.5.1	General Features.....	2
1.5.2	Sensors.....	2
1.5.3	GPS.....	2
1.6	Items Included with the IMU-GPS Logger.....	3
1.6.1	Accessories and Enclosures.....	3
1.7	Component Names.....	3
1.8	Quick Start Guide.....	4
2	Operation.....	5
2.1	USB Interface.....	5
2.2	Memory Card.....	5
2.3	Battery.....	6
2.4	Setting The RTC.....	7
2.4.1	Manual RTC Initialization.....	7
2.4.2	GPS synchronization of RTC.....	8
2.5	Status Indicators.....	8
2.6	System Configuration Options.....	8
2.6.1	General Configuration Options.....	9
2.6.1.1	absoluteTime.....	9
2.6.1.2	deadBand.....	9
2.6.1.3	deadBandTimeOut.....	10
2.6.1.4	dirName.....	10
2.6.1.5	dwell.....	10
2.6.1.6	fileName.....	11
2.6.1.7	minBattVoltage.....	11
2.6.1.8	rebootonDisconnect.....	11
2.6.1.9	samplesPerFile.....	11
2.6.1.10	statusIndicators.....	11
2.6.1.11	WakeUpTime.....	12
2.6.2	IMU Options.....	12
2.6.2.1	ag_SampleRate.....	13
2.6.2.2	ag_GyroHpf.....	13
2.6.2.3	ag_GryoFullScale.....	13
2.6.2.4	mag_SampleRate.....	13
2.6.2.5	ag_AccelFullScale.....	13
2.6.3	GPS Options.....	13
2.6.3.1	gps_clockThreshold.....	14
2.6.3.2	gps_DynamicModel.....	14
2.6.3.3	gps_MinLock.....	14

2.6.3.4	gps_nobad.....	14
2.6.3.5	gps_numsamples.....	15
2.6.3.6	GpsOn.....	15
2.6.3.7	gps_powerOnTime.....	15
2.6.3.8	gps_powerPeriod.....	15
2.6.3.9	gps_sampleinterval.....	15
2.6.4	Barometric Pressure Options.....	15
2.6.4.1	press_pressOn.....	15
2.6.4.2	press_pressureInterval.....	16
2.6.4.3	press_tempOn.....	16
2.6.4.4	press_tempSubSample.....	16
2.6.4.5	press_TriggerHeight.....	16
2.7	Example Configuration Files.....	16
2.7.1	Basic Options.....	16
2.7.2	Sensor Options.....	17
2.7.3	GPS Options.....	17
3	Data Interpretation.....	18
3.1	Data Files.....	18
3.2	Data Format.....	18
3.3	Data Conversion.....	19
3.3.1	Time Stamps (Default mode).....	19
3.3.2	Absolute Time Stamps.....	20
3.3.3	Time Stamps: Absolute Mode.....	20
3.3.4	Motion Sensor Data.....	20
3.3.5	Pressure and Temperature Data.....	21
3.3.5.1	Converting Pressure to Altitude.....	21
3.3.6	GPS Data.....	21
4	System Details.....	22
4.1	Sensors.....	22
4.1.1	Accelerometer/Gyroscope.....	22
4.1.2	Magnetometer.....	24
4.1.3	Barometric Pressure.....	24
4.2	Operating and Storage Conditions.....	25
4.3	Dimensions.....	25
5	Appendix.....	26
5.1	What is an Accelerometer.....	26
5.2	Using “R” to Analyze Data.....	28
5.2.1	What is “R”.....	28
5.2.2	Introduction to R Commands.....	29
5.2.3	Online Resources for R.....	30
5.2.4	Example Scripts in R.....	31

## Index of Figures

Figure 1: IMU-GPS Data Logger.....	2
Figure 2: IMU-GPS Data Logger Components.....	3
Figure 3: Connecting to PC.....	4
Figure 4: Editing the Config.txt File.....	4

Figure 5: Expected Battery Life.....	6
Figure 6: Example Time Entry in time.txt File.....	7
Figure 7: LED Status Indicators.....	8
Figure 8: Graphical Illustration of the Deadband Feature.....	10
Figure 9: Graphical Illustration of the Dwell Feature.....	11
Figure 10: Example of Basic Options.....	17
Figure 11: Example of Sensor Options.....	17
Figure 12: Example of GPS Options.....	18
Figure 13: Example Data File From the IMU-GPS.....	19
Figure 14: Time Stamp Conversion Method.....	20
Figure 15: Sensor Orientation.....	22
Figure 16: Enclosure Dimensions.....	25
Figure 17: Spring-mass Accelerometer.....	26
Figure 18: Simplified MEMS Accelerometer Design (L) and Actual MEMS Accelerometer (R).....	27
Figure 19: R Command Line Interface.....	28
Figure 20: RStudio Interface.....	29

## Index of Tables

Table 1: General Configuration Options.....	9
Table 2: IMU Configuration Tags and Descriptions.....	12
Table 3: GPS Configuration Tags and Descriptions.....	13
Table 4: Supported Dynamic Modes for GPS.....	14
Table 5: Barometric Pressure Configuration Tags and Descriptions.....	15
Table 6: Accelerometer Sensor Characteristics.....	23
Table 7: IMU-Gyroscope Sensor Characteristics.....	23
Table 8: IMU-Magnetometer Sensor Characteristics.....	24
Table 9: Pressure Sensor Characteristics.....	24
Table 10: Operating and Storage Conditions.....	25

# 1 Introduction

## 1.1 About This Manual

Thank you for purchasing the IMU-GPS data logger. Gulf Coast Data Concepts spent considerable efforts developing an easy to use data logger for the scientific researcher, student, or hobbyist. This product collects a complete set of motion and GPS position data simultaneously to provide an advanced insight into motion analysis. Please read this manual to understand the operation and capabilities of the data logger.

## 1.2 Document Conventions

The quick start guide in section 1.8 provides a basic summary of operation to begin using the 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 5.1) as well as software and analysis procedures (section 5.2). These short discussions will help new users learn about the IMU-GPS and how to use the data.

## 1.4 Product Summary

The IMU-GPS is a compact self-recording data logger that collects and stores accurately time stamped data to flash memory in simple text format. When connected via the USB to a personal computer, the logger appears as a standard mass storage device containing the comma delimited data files and the user setup file. The IMU-GPS includes an internal 500mAh lithium-polymer rechargeable battery, which will recharge using USB power. The IMU-GPS logger can be packaged into an alternative enclosure and fitted with different battery configurations.

## 1.5 Feature List

### 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.
- Operates from internal lithium-polymer rechargeable battery
- Weight 1.5oz (42g)
- Size 3.0L x 1.8W x 0.79H inch (76x46x20 mm)



Figure 1: IMU-GPS Data Logger

### 1.5.2 Sensors

- STMicroelectronics LSM6DMTR 6-DOF inertial sensor
  - Accelerometer, 3-axis,  $\pm 2/\pm 4/\pm 8/\pm 16$  g
  - Gyroscope, 3-axis,  $\pm 125/\pm 250/\pm 500/\pm 1000/\pm 2000$  dps
- Memsic MMC5983MA magnetometer sensor,  $\pm 8$  Gauss range ( $\pm 800000$  nT)
- Bosch Sensortec BMP384 pressure sensor, 30 - 125 kPa

### 1.5.3 GPS

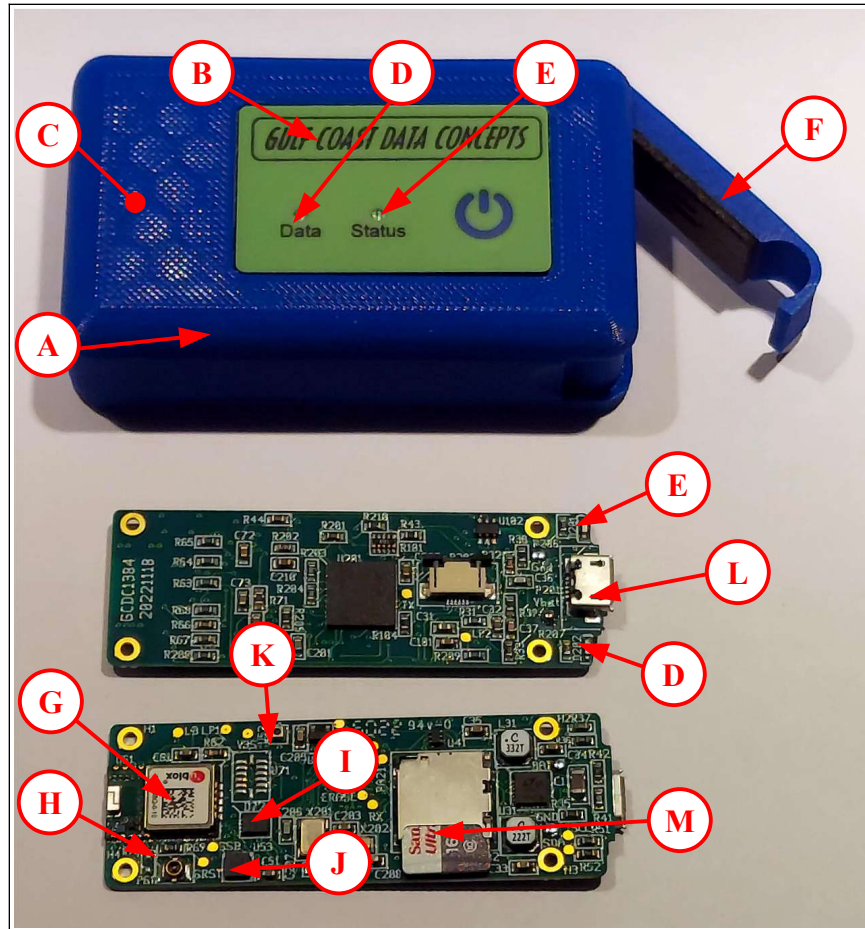
- The GPS option adds location data and precision time alignment
- u-Blox CAM-M8 GPS receiver module
  - GPS/Galileo, BeiDou, GLONASS
- Internal high gain patch antenna

## 1.6 Items Included with the IMU-GPS Logger

### 1.6.1 Accessories and Enclosures

The logger includes a 500mAh lithium-ion battery, 16GB microSD card, and microB USB cable, unless otherwise noted in the purchase agreement. GCDC can provide the IMU-GPS logger in various enclosure configurations, including a 3D printed enclosure or just the electronics system.

### 1.7 Component Names



**Figure 2: IMU-GPS Data Logger Components**

- |   |                                      |   |                         |
|---|--------------------------------------|---|-------------------------|
| A | 3D Printed Enclosure                 | G | uBlox GPS module        |
| B | External button Assembly             | H | Antenna Connector       |
| C | GPS patch antenna (inside enclosure) | I | IMU sensor (accel/gyro) |
| D | Data LED                             | J | Magnetometer sensor     |
| E | Status LED                           | K | Pressure sensor         |
| F | Access door to microB connector      | L | MicroB USB Connector    |
|   |                                      | M | MicroSD card            |

## 1.8 Quick Start Guide

The IMU-GPS 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 data logger. Configuration settings and mounting methods will depend on the particular application.

Step 1: Plug the IMU-GPS logger 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. The “Data” LED will turn off when the battery is fully charged and blink to indicated data transfers to/from the logger. The “Status” LED will flicker when searching for a GPS lock and blink once-per-second when a suitable GPS lock is achieved. The real time clock will automatically initialize with the GPS.

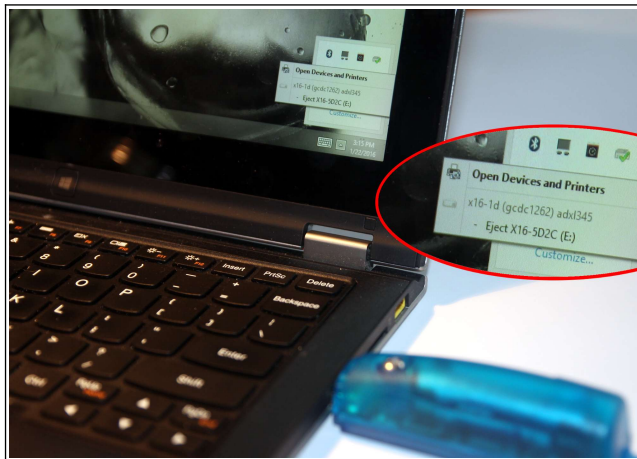


Figure 3: Connecting to PC

Step 2: Configure the IMU-GPS 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.

```
 ; PRODUCT_ID = HAM-IMU
 ; deadband, deadbandtimeout, dwell options act
 ; upon the accelerometer sensor data
 deadBand = 0
 DeadBandTimeout = 5
 ; dwell=100
 ; set the length of the data file
 samplesperfile = 72000
 ; control brightness of LEDs
 statusindicators = Normal
 ; uncomment following line to activate logger
 ; upon disconnect from USB
 rebootOnDisconnect
 ; activate max timing precision
 micros
 ; add gyro to data stream
 gyroOn
 ; add quaternion calculations to data stream
 quaternion
 ; add magnetometer data to data stream
 ; mag data occurs at slower rate than accel/gyro
 magOn
```

Figure 4: Editing the Config.txt File



- Step 3: After removing from the USB port, attach the IMU-GPS logger to the target object.
- Step 4: Press the on/off button to activate the IMU-GPS logger. Logging of sensor data will start about 3-5 seconds after activation and the “Data” LED will blink as data is written to the flash memory.
- Step 5: To stop recording, press and hold the on/off button for about 3 seconds. The LEDs will begin to blink rapidly for 2 seconds and then turn off. Data is maintained in flash memory. Press the on/off button again to restart the logger and create a new data file.
- Step 6: 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. See section 3.3 for a complete discussion of data conversion.

## 2 Operation

### 2.1 USB Interface

The IMU-GPS logger connects to a PC using a standard micro-B USB connector and supports the USB mass storage device interface for file access and file transfers. Nearly all computer operating systems recognize the logger as a typical USB external memory drive. Therefore, the logger will allow file transfers to the internal flash memory like a common USB flash drive. When connected to a PC, the IMU-GPS deactivates logging and operates only as a USB interface to the flash memory.

### 2.2 Memory Card

The IMU-GPS logger stores data to an internal microSD card. GCDC supplies a 16GB card with the IMU-GPS 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 logger 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 logger will create a folder called “GCDC”, if not already present, to place the data files.



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

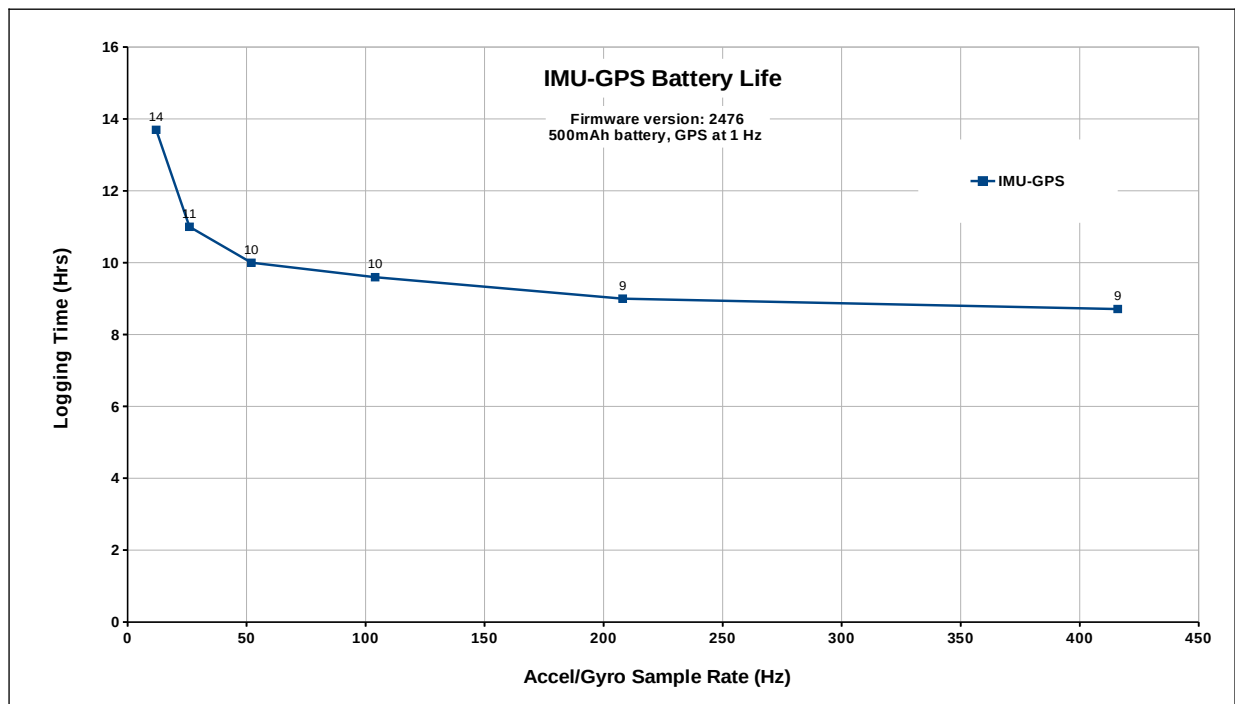
## 2.3 Battery

The IMU-GPS is powered by a lithium-polymer rechargeable battery pack. The internal battery management system recharges the battery when the logger is plugged into a USB port or attached to a USB 5v power adapter. The “Data” LED turns on (see Section 2.5) when the battery is charging and off when the battery reaches full charge.

Battery life can vary greatly due to the GPS module. The number of satellites available and antenna signal will influence the power requirements of the GPS module. Obstructions such as trees and buildings will cause the GPS module to work harder to find satellites and calculate locations. Figure 5 illustrates the battery life of the IMU-GPS data logger when fitted with a 500mAh battery, sampling GPS at 1 Hz, and the antenna positioned with a clear view of the sky.

The data logger can be powered by a non-rechargeable battery but “disableCharger” must be added to the config.txt file to deactivate the charge controller. The non-rechargeable battery must have a nominal 3.7 volts, maximum 4.2 volts, and the logger is set to shutdown at 3.2 volts. A non-rechargeable lithium-thionyl chloride (LTC) chemistry battery meets these requirements. Note that LTC batteries must be de-passivated to ensure a consistent discharge performance.

The RTC continues to operate from the battery when the device is “off”.



**Figure 5: Expected Battery Life**



*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 flash memory.*



*The logger is always “on” as it maintains the real time clock and will eventually discharge the battery completely after several months. The IMU-GPS may require an additional hour of charging from a completely discharged state. Keep in a cool (20°C/ 68°F) dry environment to avoid damage of the battery pack.*

## 2.4 Setting The RTC

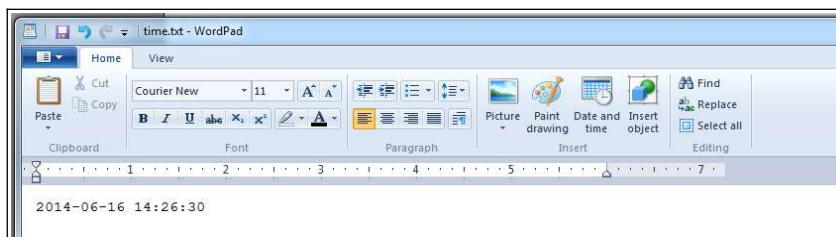
### 2.4.1 Manual RTC Initialization

A real time clock (RTC) integrated into the IMU-GPS 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 6 provides an example time.txt file that will initialize the RTC to 2:26:30 pm June 16, 2014.
- Step 3: Save this file to the root directory of the microSD card (same location as the config.txt file) and close the text editor.
- Step 4: Remove the logger from the PC. The logger will automatically find the time.txt file and initialize the RTC with the time stored in the file. The file is deleted after initialization.

The RTC maintains  $\pm 50$ ppm 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 6: 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

If the GPS feature is active, the RTC is initialized automatically to Coordinated Universal Time (UTC) based on time information acquired from the GPS (see section 2.6.3.1). 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](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 external button assembly and duplicated on the circuit board near the USB connector (see Figure 7). 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 file to deactivate the status indicators.

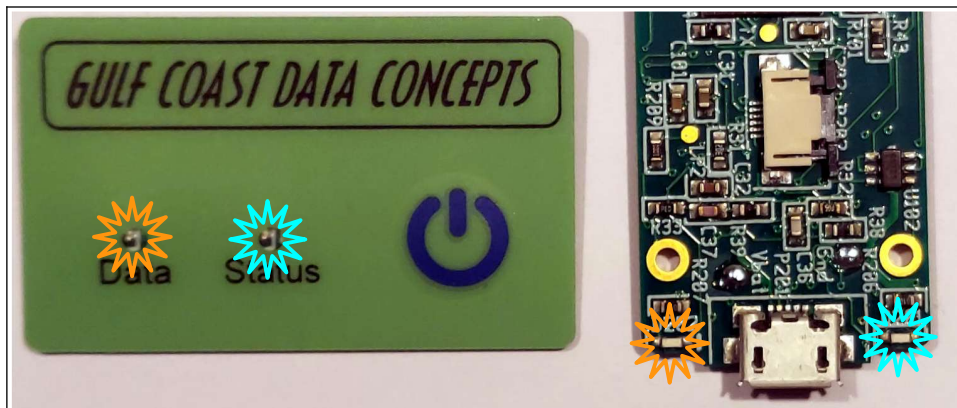


Figure 7: LED Status Indicators

## 2.6 System Configuration Options

The IMU-GPS 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.



*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 General Configuration Options

Table 1 lists the common configuration options for setting up the IMU-GPS 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		
dwel	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.

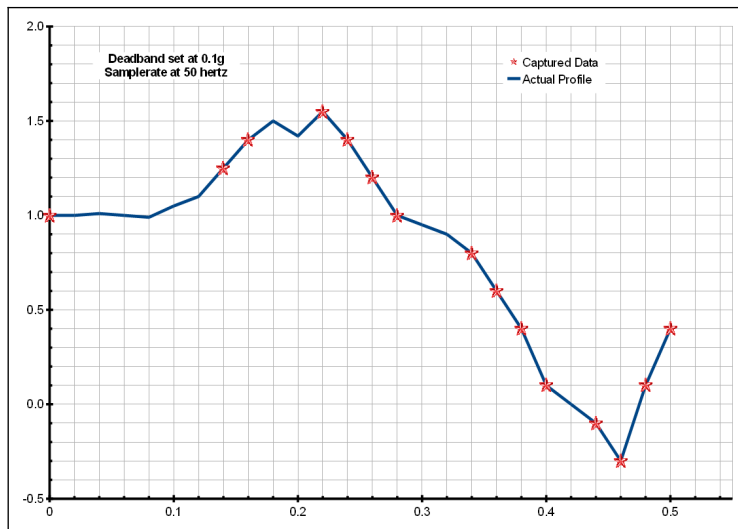
### 2.6.1.2 deadBand

“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 8 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 8: Graphical Illustration of the Deadband Feature**

### 2.6.1.3 deadBandTimeOut

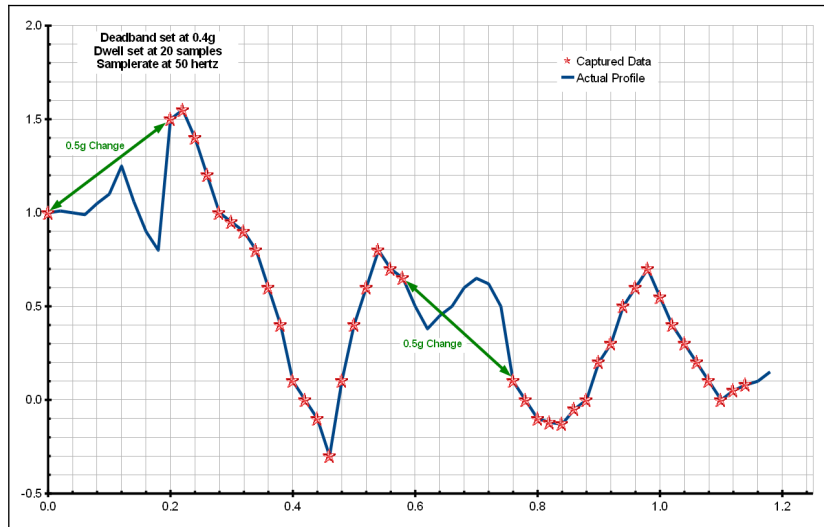
“deadBandTimeOut” defines the period in seconds when a sample is recorded by the logger regardless of the deadBand setting. This feature ensures periodic data is recorded during extended periods of inactivity. A valid setting for the deadBandTimeOut is an integer between 0 and 16384.

### 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 2.7.2 for an example implementation of the deadBand/dwell features.



**Figure 9: 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 IMU-GPS 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 IMU Options

The IMU-GPS requires the additional configuration option for setting the accelerometer, gyroscope, and magnetometer, as described in Table 2.

**Table 2: IMU Configuration Tags and Descriptions**

Tag	Valid Settings	Description
ag_SampleRate	12, 26, 52, 104, 208, 416	Accelerometer/Gyroscope sample rate (Hz)
ag_GyroHpf	0, 16, 65, 260, 1000	Gyroscope cutoff frequency (milli-Hz)
ag_GryoFullScale	125, 250, 500, 1000, 2000	Gyroscope range in degrees per second
mag_SampleRate	1, 10, 20	Magnetometer sample rate (Hz)
ag_AccelFullScale	2,4,8,16	Accelerometer range (g)



### 2.6.2.1 ag\_SampleRate

The accelerometer sensor and gyroscope sensor are packaged within the same chip and their sample rates are tied together. Therefore, the “ag\_SampleRate” setting defines the sample rate for both the accelerometer and gyroscope data. The tag defines the sample rate in Hertz, or samples per second. Valid sample rate settings are 12, 26, 52, 104, 208, and 416.

### 2.6.2.2 ag\_GyroHpf

“ag\_GryoHpf” defines the cut-off frequency for a high pass filter implemented on the gyroscope data. The frequency is defined in milli-Hz and valid settings are 0, 16, 65, 260, and 1000.

### 2.6.2.3 ag\_GryoFullScale

“ag\_GryoFullScale” sets the fulls scale range of the gyroscope, expressed in degrees per second units. Valid settings are 125, 250, 500, 1000, and 2000.

### 2.6.2.4 mag\_SampleRate

“mag\_SampleRate” sets the magnetometer sample rate in Hz. Valid settings are 1, 10, and 20.

The magnetometer sensor operates independently of the accelerometer/gyroscope. The magnetometer data is appended to the most recent gyroscope sample so the magnetometer sensor sample rate must be set less than the gyroscope sample rate.

### 2.6.2.5 ag\_AccelFullScale

“ag\_AccelFullScale” sets the fulls scale range of the accelerometer, expressed in G’s. Valid settings are 2, 4, 8, and 16 G. A “G” is 32.17 ft/sec<sup>2</sup> or 9.81 m/sec<sup>2</sup>.

## 2.6.3 GPS Options

Table 3 lists the configuration options for the GPS feature.

**Table 3: GPS Configuration Tags and Descriptions**

<b>Tag</b>	<b>Valid Settings</b>	<b>Description</b>
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

### 2.6.3.1 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 formatted 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](https://en.wikipedia.org/wiki/Time_zone)*

### 2.6.3.2 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 4: 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

### 2.6.3.3 gps\_MinLock

“gps\_MinLock” sets the minimum Horizontal 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.

### 2.6.3.4 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.

### 2.6.3.5 gps\_numsamples

“gps\_numsamples” defines the number of samples to record that meet the gps\_MinLock criteria.

### 2.6.3.6 GpsOn

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

### 2.6.3.7 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.

### 2.6.3.8 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.

### 2.6.3.9 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).

## 2.6.4 Barometric Pressure Options

The IMU-GPS includes a high precision barometric pressure sensor for calculating altitude.

**Table 5: Barometric Pressure Configuration Tags and Descriptions**

<b>Tag</b>	<b>Valid Settings</b>	<b>Description</b>
press_pressOn	-	Adds pressure samples to data file
press_pressureInterval	Integer between 50 and 32768	Sets the period in milliseconds between pressure samples
press_tempOn	-	Adds temperature samples to data file
press_tempSubSample	Integer between 1 and 255	Sets the number of pressure samples between temperature samples
press_TriggerHeight	Integer number	Meters above the baseline altitude to start logging data

### 2.6.4.1 press\_pressOn

“pressOn” adds the temperature compensated pressure readings from the sensor to the data stream.

The BMP180 pressure sensor includes a temperature sensor that is used for temperature compensation

of the pressure measurements.

#### **2.6.4.2 press\_pressureInterval**

“pressureInterval” defines the time period, in milliseconds, between pressure samples. The data logger will support intervals between 50 milliseconds (20 Hz) and 32678 milliseconds (9 hr interval). The pressure and temperature data is added to the previously available accelerometer/gyroscope sample stored in the cache. Therefore, the exact time at which pressure data is collected and the associated time stamp will be slightly different. This is rarely a problem in post-process analysis since pressure changes much slower than motion data.

#### **2.6.4.3 press\_tempOn**

“tempOn” adds the temperature values to the data stream. Remove “tempOn” or comment the term (“;”) to hide the temperature data, but the temperature compensation algorithm for pressure will continue to operate.

#### **2.6.4.4 press\_tempSubSample**

The temperature samples are collected at sub-intervals of the pressure readings. “interleave” defines the number of pressure samples taken before a temperature sample is collected. The temperature data is used in the temperature compensation algorithm.

#### **2.6.4.5 press\_TriggerHeight**

The IMU-GPS logger can be set to initiate recording at a specific altitude. “press\_TriggerHeight” defines the altitude in meters above the baseline measurement. The baseline measurement is established when the logger is first turned on. The logger needs a few seconds to create the baseline. Then, it enters a standby mode (status LED blinks) while it waits for the trigger height. Altitude data is calculated once every 3 seconds.

### **2.7 Example Configuration Files**

The IMU-GPS logger supports a very wide range of configuration options. The following sections break-down the config.txt file into manageable segments for discussion purposes.

#### **2.7.1 Basic Options**

Figure 11 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 10: Example of Basic Options**

## 2.7.2 Sensor Options

Figure 12 continues the example configuration file. The accelerometer and gyroscope sensors are set to 104 samples per second and the magnetometer samples slower at 10 Hz. The gyroscope scale is 250 dps and the filter is not used (commented out). The accelerometer scale is set to 4G range. The barometric pressure is recorded at 10 Hz and temperature is sampled with each pressure value. The logger is set to start recording data when the altitude reaches 50 meters above the initial level.

```

: ACCEL GYRO PARAMETERS
; available sample rates are 12, 26, 52, 104, 208, and 416
; sample rate affects both accel and gyro
ag_sampleRate = 104
; available HPF filter rate are 0, 16, 65, 260, 1000 milliHz
;ag_gyroHpf = 65
; available full scale ranges are 125 250, 500, 1000 and 2000 degrees per sec (dps)
ag_gyroFullScale = 250
; available full scale ranges are 2, 4, 8 and 16 g
ag_accelFullScale = 4
; MAGNETOMETER PARAMETERS
; availble sample rates are 1, 10, 20 50, 100 and 200 Hz, YES
mag_SampleRate= 10
; PRESSURE SENSOR PARAMETERS
;add pressure to data stream
press_pressOn
; the pressure interval is the pressure transducer measurement interval, in milliseconds
press_sampleInterval=100
;add temperature to data stream
press_tempOn
press_tempSubSample=0
; the trigger is measured in meters above turn-on location
;press_triggerHeight = 50

```

**Figure 11: Example of Sensor Options**

## 2.7.3 GPS Options

The GPS is set 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 50 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 of GPS Options**

## 3 Data Interpretation

### 3.1 Data Files

The IMU-GPS logger creates a new data file when the system is booted or when the maximum number of data lines is reached in the previous data file. A system boot condition occurs when the on/off button is pressed or when the logger is removed from a computer USB port with the “rebootondisconnect” feature enabled. Data files are placed in a folder named “GCDC” (default) 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 shows 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. The time stamps between the two file will indicate the gap.*

### 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 (“;”).

```

;Title, http://www.gcdataconcepts.com, LSM6DSM, BMP384, GPS
;Version, 2476, Build date, Apr 17 2023, SN:CCDC20235263491
;Start_time, 2023-06-09, 10:39:35.425
;Uptime, 5,sec, Vbat, 4134, mv, EOL, 3500, mv
;Deadband, 0, counts
;DeadbandTimeout, 5.000,sec
;LSM6DSM, SR,104,Hz, Units, mG, mdps, fullscale gyro 250dps, accel 4g
;Magnetometer, SR,10,Hz, Units, nT, Temperature, 23,degC
;BMP384, SI, 0.100,sec, Units, Pa, mdegC
;Alt Trigger disabled
;CAM_M8 Gps, SR,1,Hz
;Gps Sats, TOW, 0, ver, 0, numSat, 0
;, gnssId, svId, cno, elev, azmith, prRes, flags,inUse
;Time, Ax, Ay, Az, Gx, Gy, Gz, Mx, My, Mz, P, T, TOW, Lat,Lon, Height(m), MSL(m), hdop(m), vdop(m)
1686307175.910554,-322,-832,509,29050,-1698,44205
1686307175.919842,-75,-977,198,38640,17228,57067
1686307175.929130,-200,-1053,279,50689,21866,57715
1686307175.938418,-388,-887,541,-6037,-27878,44108
1686307175.947706,-381,-841,550,-51196,-48694,48028
1686307175.957001,-256,-954,494,-27370,-34274,71347,-17376,-6011,-14508
1686307175.966282,-127,-1061,407,5758,-26574,94718
1686307175.975563,2,-1181,423,20029,-34764,102620
1686307175.984844,-83,-1125,399,31780,-47802,82083
1686307175.994125,-214,-1021,369,23958,-74218,50846
1686307176.003406,-295,-977,454,5285,-118222,37520
1686307176.012687,-315,-891,417,-6151,-142179,43120
1686307176.021968,-306,-898,421,-8942,-144620,68416
1686307176.031250,-233,-959,503,-11725,-132012,106067,,,,101104,26765
1686307176.040531,-17,-852,515,-13265,-114442,141277
1686307176.049812,0,-793,468,-11926,-106234,155251
1686307176.059093,50,-761,441,-5022,-104773,136508
1686307176.068374,-102,-745,228,-3683,-93380,83912
1686307176.077655,-158,-609,268,-18392,-71374,56131
1686307176.086936,-137,-675,352,-67410,-52378,60908
1686307176.096217,-118,-836,477,-100826,-34152,54398
1686307176.105652,-81,-927,550,-118308,-22847,39173,-15460,-6964,-14801,101105,26768
1686307176.114940,-61,-997,615,-118860,-20825,24193
1686307176.124228,-35,-1019,597,-108570,-15925,12215
1686307176.133516,-127,-1051,560,-83186,-10177,3412
1686307176.142804,-301,-1114,582,-53007,-13309,4847
1686307176.152092,-147,-1181,634,-30275,-18857,27877
1686307176.161380,-113,-1019,602,-14236,-23529,63682
1686307176.170668,-132,-652,628,718,-25848,100082
1686307176.179962,-118,-572,557,22313,-22007,116383,-12896,-8367,-14361,101102,26765,, 488406.000, 30.3603282,-
89.1171898, -29.411,-1.766, 0.001,0.001
1686307176.189250,-73,-727,359,54522,6212,102488

```

**Figure 13: Example Data File From the IMU-GPS**

### 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.

	A	B	C	D	E	F	G	H	I	J	K
1	Title	http://www.gcdataconcepts.com									
2	Version	1110	Build date	Jan 29 2016	SN:CCDC10161318B9C						
3	Start_time	2016-01-21 01:55:59.000									
4	Temperature	-999.00	deg C	Vbat	4044	mv	2016-01-21 01:55:59.000				
5	SampleRate	400	Hz								
6	Deadband	0	counts								
7	DeadbandTir	5	sec								
8	Time	Ax	Ay	Az			New Date Stamps	Ax(g)	Ay(g)	Az(g)	
9	0.042	-733	-45	1828			2016-01-21 01:55:59.042	-0.358	-0.022	0.893	
10	0.044	-818	-22	1856			2016-01-21 01:55:59.044	-0.399	-0.011	0.906	
11	0.047	-872	-22	1880			2016-01-21 01:55:59.047	-0.426	-0.011	0.918	
12	0.049	-888	-18	1889			2016-01-21 01:55:59.049				
13	0.052	-870	0	1869			2016-01-21 01:55:59.052				
14	0.054	-820	16	1842			2016-01-21 01:55:59.054				
15	0.057	-752	7	1817			2016-01-21 01:55:59.057				
16	0.059	-674	-20	1760			2016-01-21 01:55:59.059	-0.329	-0.010	0.859	
17	0.062	-641	-18	1703			2016-01-21 01:55:59.062				
18	0.064	-659	-17	1678			2016-01-21 01:55:59.064				
19	0.067	-675	-29	1665			2016-01-21 01:55:59.067				
20	0.07	-702	-20	1667			2016-01-21 01:55:59.070	-0.343	-0.010	0.814	
21	0.072	-724	-38	1676			2016-01-21 01:55:59.072	-0.354	-0.019	0.818	

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 Time Stamps: Absolute Mode

Using the “absolutetime” parameter in the configuration file sets the time stamp format to absolute seconds elapsed since Epoch, which is January 1, 1970. Figure 13 includes an example data file using the absolute mode time stamps. Programs such as Matlab, Octave, and R will directly import this time stamp format and automatically convert it to a standard date and time format.

### 3.3.4 Motion Sensor Data

Sensor data follows the time stamp and the sensor with the fastest sample rate occurs first. Slower sensors are then added to the data stream. Referencing Figure 14, the column headers (line 14) lists the sensor order: accelerometer, gyroscope, magnetometer, pressure, temperature, and GPS data.

The accelerometer, gyroscope, and magnetometer data is recorded in engineering units as described in lines 7-9 of the file header. The accelerometer and gyroscope range is listed after the units. For example, in Figure 13, the accelerometer is milli-G (mG) and set to 4g range, the gyroscope is degrees per second (dps) and set to 250 dps, and the magnetometer is nano-Tesla (nT).



### 3.3.5 Pressure and Temperature Data

The logger sets the BMP384 sensor to high-precision mode to provide the most accurate measurement of pressure. Pressure data is recorded to the data file in Pascal units and includes the appropriate compensation based on temperature collected at the pressure sensor.

Temperature is recorded in milli-degree Celsius. Divide the temperature value by 1000 to determine degrees Celsius.

#### 3.3.5.1 Converting Pressure to Altitude

Altitude is calculated from the pressure data using the following equation:

$$Altitude = 44330 \times \left( 1 - \left( \frac{P}{P_o} \right)^{\frac{1}{5.255}} \right)$$

where *Altitude* = meters above baseline altitude

*P* = pressure in Pascal

*P<sub>o</sub>* = pressure in Pascal at the baseline altitude (mean sea level = 101325 Pa)

Nominally, sea level pressure is 101325 Pa. However, *P<sub>o</sub>* should be the pressure at the baseline for which the calculated altitude is relative to.

### 3.3.6 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 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 50 Hz may actually push data at 52 Hz. The logger 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 of the accelerometer and gyroscope data.

Additional sensor data, such as the magnetometer, pressure, temperature, and GPS, arrive at slower rates than the accelerometer and gyroscope data. The logger appends the new data to the last accelerometer/gyroscope entry available in the memory cache. Therefore, the magnetometer, pressure, temperature, and GPS values are not synchronized exactly to the particular time stamp. This method was chosen to simplify the data file format and allow easier parsing of the file. Typically, the slower sensors do not change significantly relative to accelerometer and gyroscope data so they can be assumed to be synchronized.



Figure 15: Sensor Orientation

#### 4.1.1 Accelerometer/Gyroscope

The IMU-GPS logger incorporates the STMicroelectronics LSM6DMTR 6-DOF inertial sensor, which includes an accelerometer and gyroscope. Table 6 lists the basic sensor and logger performance parameters. Refer to STMicroelectronics for detailed sensor specifications.

**Table 6: Accelerometer Sensor Characteristics**

Parameter	Condition	Min	Typical	Max	Units
Acceleration range			±16.0		g
Sensitivity			0.001		g
Sensitivity Deviation			±1.0		%
Nonlinearity	X, Y, Z axis		±0.5		%FS
Zero-g Offset Level Accuracy	X, Y axis	-150		+150	mg
	Z axis	-250		+250	mg
Inter-Axis Alignment Error			±0.1		Degrees
Cross-Axis Sensitivity			±1		%



*The 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 1000 mG in the z-axis.*

**Table 7: IMU-Gyroscope Sensor Characteristics**

Parameter	Condition	Min	Typical	Max	Units
Sensitivity	±250 °/sec		131.072		counts/°/sec
	±500 °/sec		65.536		
	±1000 °/sec		32.768		
	±2000 °/sec		16.384		
Nonlinearity	X, Y, Z axis		0.1		%FS
Zero Rate	X, Y, Z axis		±5		°/sec
Sensitivity Tolerance	X, Y, Z axis		±1.5		%
Cross-Axis Sensitivity			±2		%

## 4.1.2 Magnetometer

**Table 8: IMU-Magnetometer Sensor Characteristics**

Parameter	Condition	Min	Typical	Max	Units
Full Scale Range	X, Y, Z axis		±800000		nT
Sensitivity Accuracy			±5		%
Sensitivity Linearity			±0.1		%
Null Field Output			±50000		nT



*The magnetic sensor detects the Earth's magnetic field lines, which is used to help determine orientation. Other magnetic fields, such as from permanent magnets and electromagnetic systems, will affect the sensor output. Therefore, do not use magnets as an attachment method for the IMU logger.*

## 4.1.3 Barometric Pressure

The IMU-GPS logger includes the Bosch Sensortec BMP384 digital pressure sensor. Table 9 lists the basic sensor characteristics. Refer to Bosch for complete sensor details. The sensor is configured to operate in high-precision mode to maximize the measurement accuracy. Refer to section 3.3.5.1 for converting the pressure values to altitude.

**Table 9: Pressure Sensor Characteristics**

Parameter	Condition	Min	Typical	Max	Units
Operating temperature	Operational	-40		+85	°C
Absolute accuracy pressure $V_{DD}=3.3\text{ V}$	30000 - 70000 Pa (0 to +65°C)		±50		Pa
	30000 - 110000 Pa (-20 to 65 °C)		±65		Pa
Resolution of output data	Pressure		0.016		Pa
	Temperature		0.1		°C
Relative accuracy pressure	90000 – 110000 Pa (25° to 40°C)	-0.15	±9	+15	Pa

## 4.2 Operating and Storage Conditions

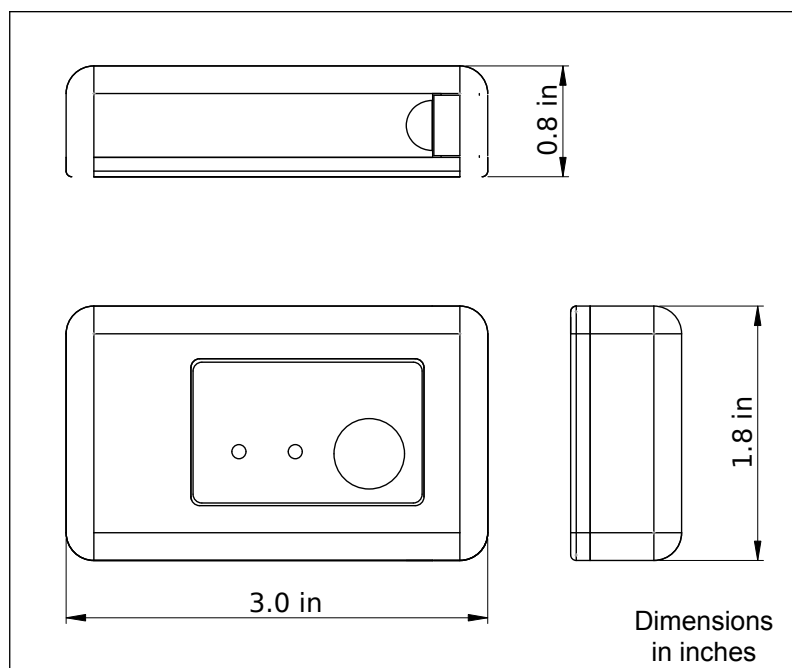
The operating temperature range is limited primarily by the lithium-polymer battery capabilities. The 3D printed enclosure should not be considered waterproof so care must be exercised to protect the logger from water, humidity, dust, and other adverse environmental conditions.

**Table 10: 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 overall IMU-GPS enclosure dimensions are 3.0 inches long, 1.8 inches wide, and 0.8 inches high (76x45x20 mm), and it weighs 1.7 ounces (50g).



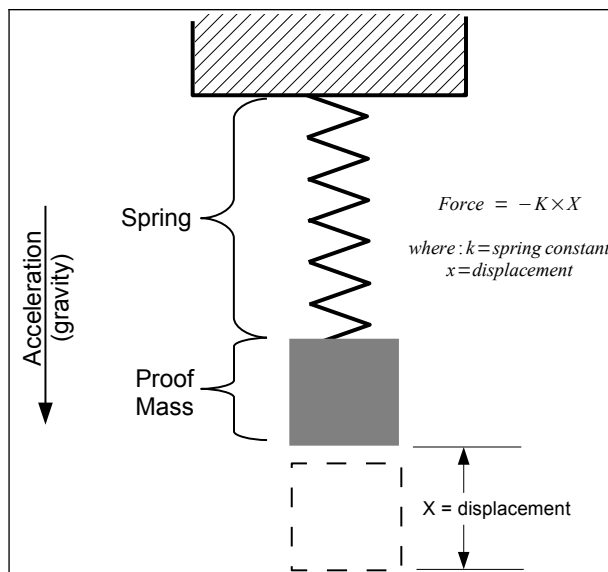
**Figure 16: Enclosure Dimensions**

## 5 Appendix

### 5.1 What is an Accelerometer

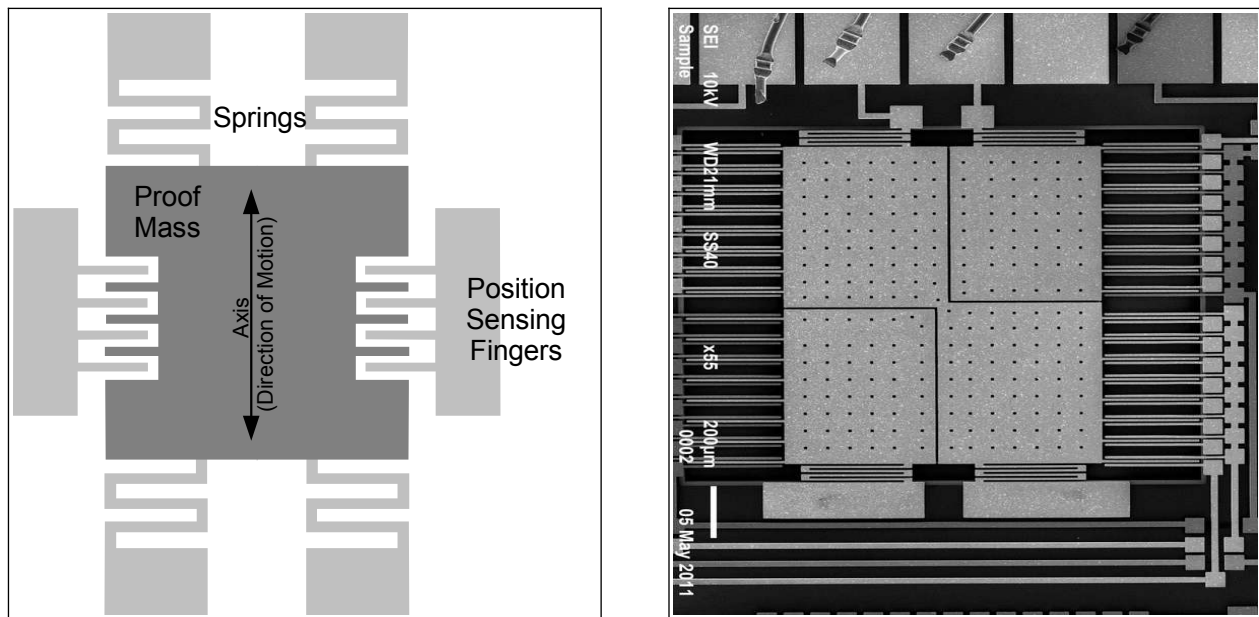
Acceleration is the change in velocity. A stationary body has no acceleration and a body moving at a constant velocity has no acceleration. However, a body changing from a stationary condition to motion experiences acceleration by means of a force. Newton's second law establishes this relationship as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

An accelerometer [ak-se-lə-'rā-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.

## 5.2 Using “R” to Analyze Data

### 5.2.1 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](http://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.

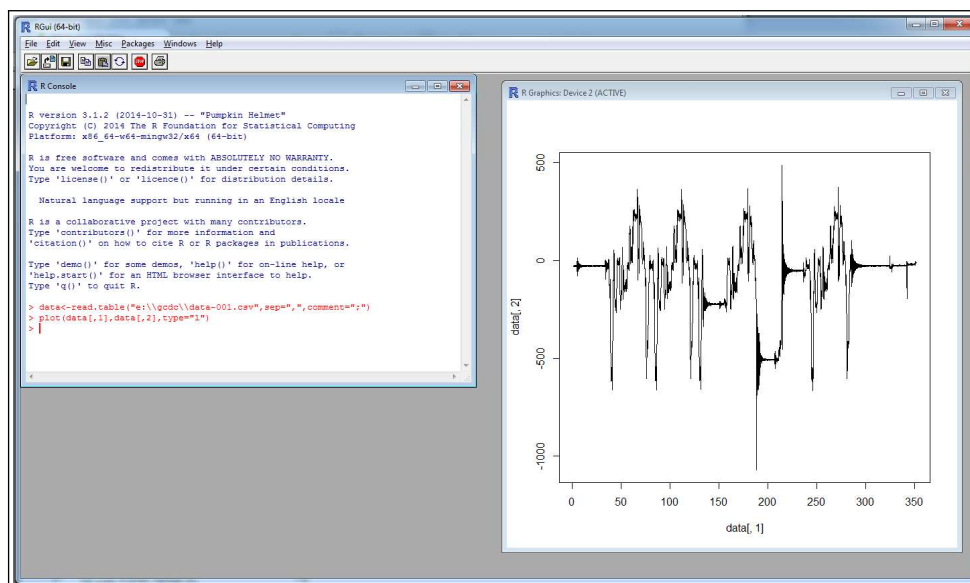


Figure 19: R Command Line Interface



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](http://www.rstudio.com)

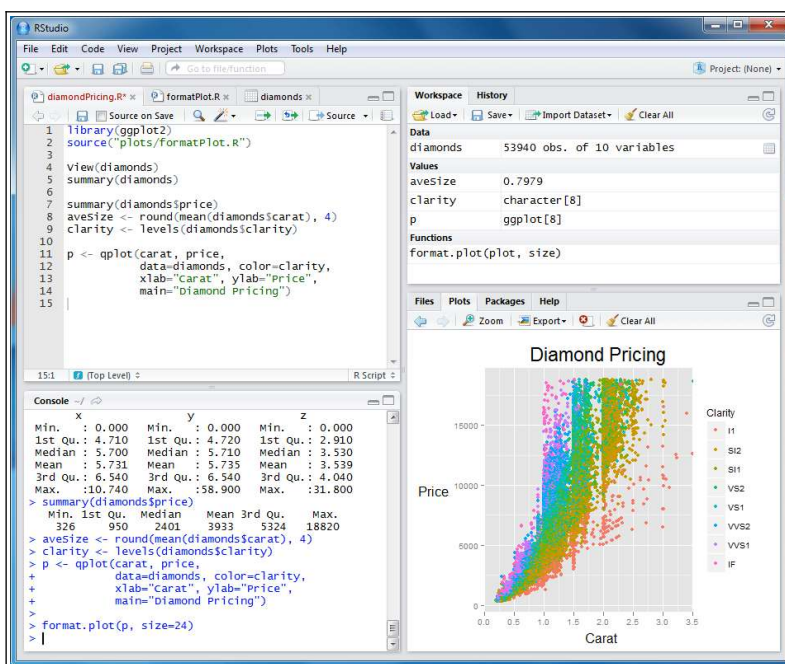


Figure 20: RStudio Interface

## 5.2.2 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 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”
```

### 5.2.3 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](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.

#### **5.2.4 Example Scripts in R**

Several example applications using R scripts are available at the GCDC website or are included with the 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