GPS

<https://github.com/adafruit/Adafruit-GPS-Library> (this is the library)

<https://learn.adafruit.com/adafruit-ultimate-gps-logger-shield/overview>

Info for SD

BNO sensor <https://learn.adafruit.com/adafruit-bno055-absolute-orientation-sensor/overview>

Pressure

<https://www.parallax.com/sites/default/files/downloads/29124-APPNote_520_C_code.pdf> (will this work for arduino?)

<https://www.parallax.com/sites/default/files/downloads/29124-MS5607-02BA03-Datasheet.pdf>

SD

<https://www.arduino.cc/en/Reference/SD>

(Library already in Arduino)

Libraries

<https://www.arduino.cc/en/Guide/Libraries>

(how to download library)

<https://learn.adafruit.com/adafruit-bno055-absolute-orientation-sensor/device-calibration>

GPS:

1. May need external GPS antenna
2. Establish fix: go outside and wait 45 sec to 30 min
3. Get data from wither direct of soft serial

\*\*\*Do we want direct connect or soft serial?

Direct Connect: (Arduino would act like a USB->UART bridge)

* upload a blank sketch
* Wiring: VIN to 5V, GND to ground, RX to D 0, TX to D 1.
* Flip switch on board to direct
* Plug in USB and select 9600 baud
* Establish a fix to get data
* Look for **$GPRMC,194509.000,A,4042.6142,N,07400.4168,W,2.03,221.11,160412,,,A\*77**
* Latitude: DDMM.MMMM (The first two characters are the degrees.) Longitude: DDDMM.MMMM (The first three characters are the degrees.)

Soft Serial:

* Turn switch to soft serial
* Download libraries
* Parsing example: change SoftSerial mySerial(3,2) to SoftSerial mySerial(8,7)
* upload to arduino and open serial monitor

1. SD Logging:
   * Slide SD into socket
   * don’t have to update SD library b/c using Uno
   * open **Adafruit\_GPS->shield\_sdlog** in GPS library
   * find if(!SD.bigin(chipSelect, 11, 12, 13)){

//if(SD.begin(chipSelect)){

uncomment second line and comment out first one.

* + makes sure switch is set for soft serial
  + connect to 115000 baud

1. PWR-power, L13-SD card access, Fix-blinks once every 15 sec when it has a fix

SD Logging: I am assuming that you use an adapter to get the data from the GPS

* SD card attached to SPI(serial Peripheral Interface) bus (MOSI(Master Out Slave In) - pin11, MISO (Master In Slave Out)- pin12, CLK(Serial Clock) - pin13, CS - pin4)

-Check if SPI\_MODE0,1,2, or 3 for setting (SPISettings mySettting(speedMaximum, dataOrder, dataMode)

* Include SPI.h and SD.h
* Make a string for assembling the data to log
* Add to the string any info necessary
* Create a File object. Method .println(string with info) for instantiated object. close() to close

RTC:

The real time clock in the GPS is NOT 'writable' or accessable otherwise from the Arduino. Its in the GPS only! Once the battery is installed, and the GPS gets its first data reception from satellites it will set the internal RTC. Then as long as the battery is installed, you can read the time from the GPS as normal. Even without a proper "gps fix" the time will be correct.

Abs. Orientation:

PINS:

* Sensor VIN to 5V
* 3vo - voltage regulator
* Sensor GND to ground
* Sensor SCL to microcontroller’s I2C clock line --- A5 (or SCL??)
* Sensor SDA to microcontroller's I2C data line ---- A4 (or SDA??)
* Sensor RST:low then high to reset
* INT - capable of interrupting
* Sensor ADR \*\*\*MAY HAVE TO USE SINCE TWO ICS!!

GPS coordinates