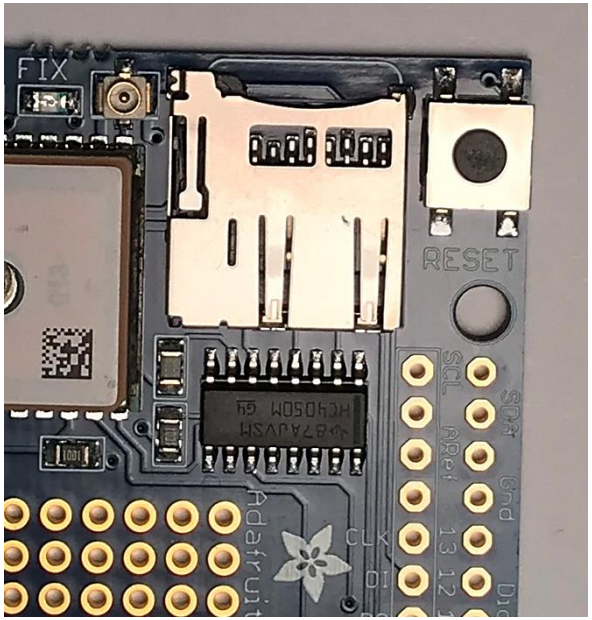
The Ultimate GPS Logger Shield - SD

LaACES Student Ballooning Course

SD Card

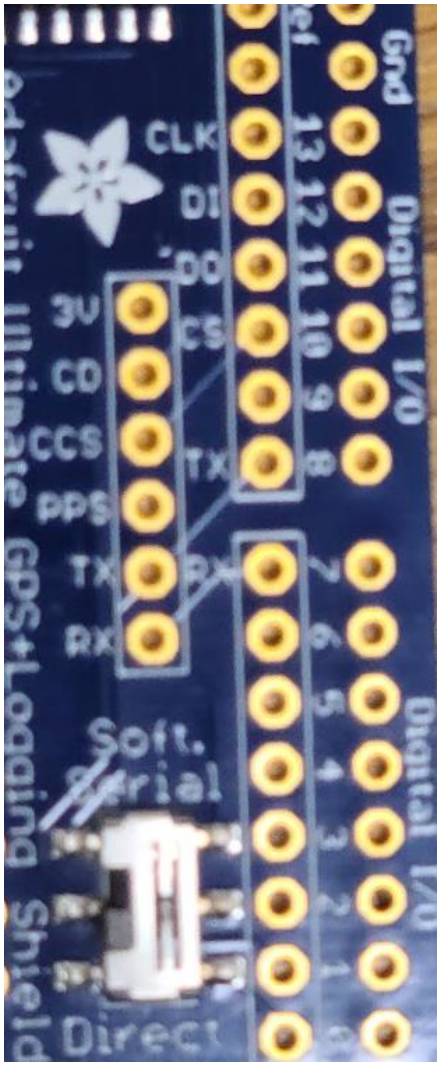
- The GPS shield has a microSD card slot. Use this to save the GPS and flight data
- Communication between the Mega and SD card uses SPI
- The microSD can be any capacity, but be aware of the limitations of the SD library being used
- When inserting the microSD, ensure that it latches. If it does not latch, no data will be logged
- Data saved in files on SD card



The Adafruit Ultimate GPS Logger Shield's microSD socket. It is next to the Reset button. When inserting a microSD card, it will latch once fully inserted. To eject simply push the card in again

SD Card Communications

- SPI communication requires 4 lines: MISO, MOSI, Clock, and Chip Select
- On the Adafruit Shield pins 10, 11, 12, 13 are CS, MISO (DO), MOSI (DI) and CLK, these are the Mega SPI pins
- On the Mega2560 MISO, MOSI, and SCK are digital pins 50, 51, and 52
  - Because of this we either need a library that will do Software SPI or to solder jumpers between the pins
  - CS pin is selectable in most libraries



SD Library: Installation

- Because of this for the activites, an older SD library will be used. It can be found at: https://github.com/adafruit/SD
- We will need to manually install the library
  - Download the zip file. Extract it into the libraries folder.
  - The folder will be named “SD-master.” Rename it as “SD”.
  - Default windows location: Documents >> Arduino >> libraries
- This library was chosen because it allows you to easily implement software defined SPI pins.
- This must be done before launching the IDE to load the library

SD Library: Limitations and Characteristics

- Card must be FAT32 or FAT16
  - ExFAT cards will not be recognized
- Only ~2 GB storage of the SD card will be recognized/ accessible
- File names must follow 8.3 format
- Software SPI will lead to slower write speeds
- It is important to note while this library installed it will override the default Arduino SD library

Filenames

- File names must follow 8.3 format – FILENAME.EXT
  - Filenames can be shorter than 8 characters but cannot be longer and include a 3 character extension
  - Common extension types are .txt and .csv
- If possible, use filenames that convey information
  - Ex: Use a timestamp for the filename
- Approved characters in Adafruit SD library filenames
  - Letters, numbers, _, - (not all inclusive)
- Characters not allowed in filename by Adafruit library
  - Spaces, periods (not all inclusive)

Extension Types

- Some common extension types are .txt and .csv
- CSV is a comma-separated values file
  - Figure 2 shows an example of what this data could look like
  - Excel automatically separates each row by the delimiter (,)
- TXT file can have the same data format, but user must manually tell Excel what the delimiter is

```
1  ***Logging in file 01153040.csv created at
2  START,Timestamp,Altitude,# Satellites,Fix Q
3  START,_____No Fix!_____,0.00,0,0,5142,0,0
4  START,_____No Fix!_____,0.00,0,0,10142,0,
5  START,_____No Fix!_____,0.00,0,0,15142,0,
6  START,_____No Fix!_____,0.00,0,0,20142,0,
7  START,09/01/2019 15:31:04,1204.10,6,1,28116
8  START,09/01/2019 15:31:09,1207.20,6,1,33116
9  START,09/01/2019 15:31:14,1206.80,6,1,38116
10 START,09/01/2019 15:31:19,1208.10,6,1,43116
11 START,09/01/2019 15:31:24,1209.80,6,1,48116
```

	A	B	C	D	E	F	G
1	***Logging in file 01153040.csv created at 09/01/2019 15:30:40 for code version FlightC						
2	START	Timestamp	Altitude	# Satellite	Fix Qualit	FC Millis	LC Upper
3	START	_____No Fix!_____	0	0	0	5142	0
4	START	_____No Fix!_____	0	0	0	10142	0
5	START	_____No Fix!_____	0	0	0	15142	0
6	START	_____No Fix!_____	0	0	0	20142	0
7	START	9/1/2019 15:31	1204.1	6	1	28116	0
8	START	9/1/2019 15:31	1207.2	6	1	33116	0
9	START	9/1/2019 15:31	1206.8	6	1	38116	0
10	START	9/1/2019 15:31	1208.1	6	1	43116	0
11	START	9/1/2019 15:31	1209.8	6	1	48116	0

Figure 2: This is an example of a .csv file. Left – opened in a text editor. Right – opened in Excel, the commas are used to separate the columns.

Designing a Data Packet

- For this program we recommend you record your data in plain text in CSV files
  - CSV format easily opened excel and other software
- Plain text allow easily viewing the data and can determine if an error occurred during writing
  - SD Library uses same format as Serial libraries to write text to a file
- Comma delimiters separate the data into discrete spots
  - If a particular piece of data is missing can easily see the rest of data
- Use a header as the first line of the file as a label

```
1  Num1,Num2,Num3,Num4
2  1.50,3.25,5.00,7.57,
3  2.00,3.75,5.50,8.07,
4  2.50,4.25,6.00,8.57,
5  3.00,4.75,6.50,9.07,
6  3.50,5.25,7.00,9.57,
7  4.00,5.75,7.50,10.07,
8  4.50,6.25,8.00,10.57,
```

Other Data File Recommendations

- How do you need to format the numbers you are recording
  - Number decimal places, positive or negative signs
- Its easiest to have a data format rather than multiple types of formats, even if data will be recorded at different rates
  - Have a column for each possible type of data if not recorded in that record can just not write a number
  - NUMBER,TYPE,TIME,TEMP1,TEMP2,ALT,
  - 1,A,3,34.0,12.0,,
  - 2,A,8,33.0,12.0,,
  - 3,B,10,,,500.2,
- Make sure you can easily identify each data pack uniquely and include sort of type identifier if you have multiple sorts of data records to allow sorting after recording

Process of Writing Data to SD Card

- SD card communication initialized in setup()
  - SD.begin(CS, MOSI, MISO, CLK);
- Open/create SD file
  - myFile = SD.open(filename, FILE_WRITE);
- Write to SD card – same as printing to Serial Monitor
  - myFile.println("This sentence will be written to my SD file");
- Flush the data
  - myFile.flush();
- When finished, close the file
  - myFile.close();

SdFat

- If you get comfortable with the SD card library used here recommend moving to SDFAT library
- SdFat removes the size and filename limitations
- Can install jumpers to SPI connections :
  - MISO, MOSI, and SCK (50, 51, and 52) must be hardwired to pins (12, 11, and 13)
  - Since SPI is synchronous match all pin jumper lengths and keep lead as short as reasonable
- Also has Software SPI implementation but requires editing of the library files
- The SdFat library can be found at https://github.com/greiman/SdFat