Integrated Flight Computer for High Altitude Balloon

Dominic Critchlow[1], Bryan Gaither[1], Justin Oelgoetz[1]

Austin Peay State University[1]

Abstract

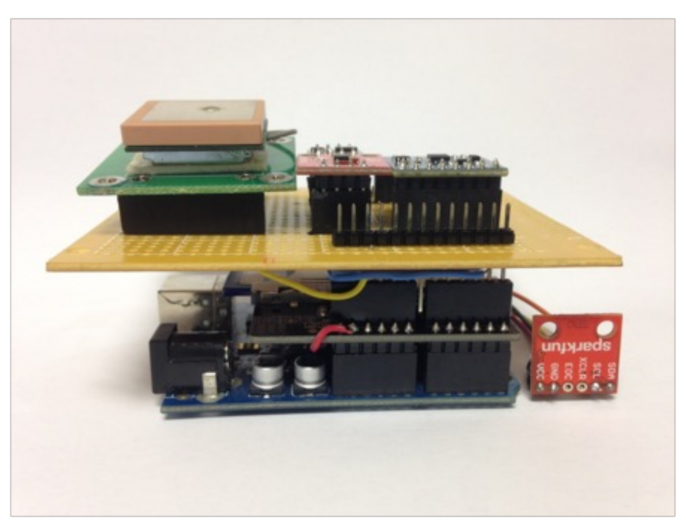
Integrating a number of pieces of scientific equipment for in flight use on a stratospheric balloon requires a common flight computer. This is especially important if one wants to tie data from sensors such as an accelerometer, magnetometer, and GPS to a piece of equipment such as a Geiger counter. We are designing and building an integrated flight computer that combines flight sensors, and can be used to collect data from other systems based around the Arduino micro-controller. The system logs data for altitude, pressure, and temperature as well as a 3 axis magnetometer and a 3 axis accelerometer to record forces on the payload. GPS data is also being recorded to the data that is acquired to a specific location. The system is expandable to record data from other devices (such as a Geiger counters) and to control external hardware (such as a camera or sample collector).

Hardware

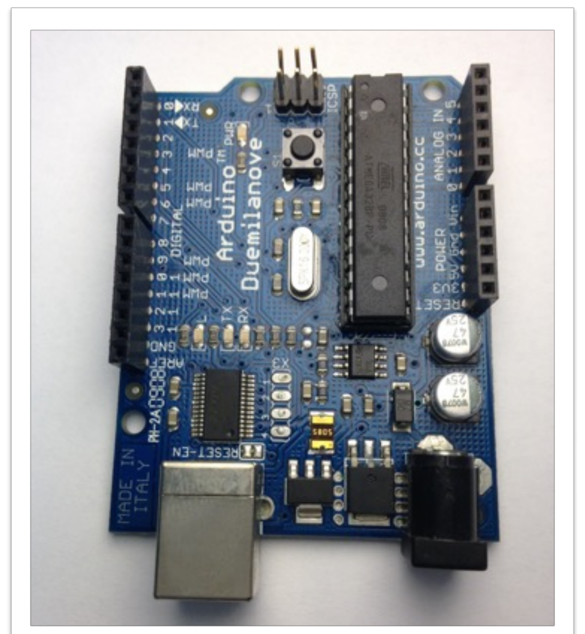
- The individual parts were selected to make a system that can give location, orientation, and forces for a high altitude balloon.
- Each sensor is mounted on a shield so that they are easily removable and replaceable [Image 1.4].
- The Pressure sensor is attached with wires so it can be mounted outside on the payload of the ballon.
- The whole system is based around an Arduino micro controller [Image1.2].

List of Parts:

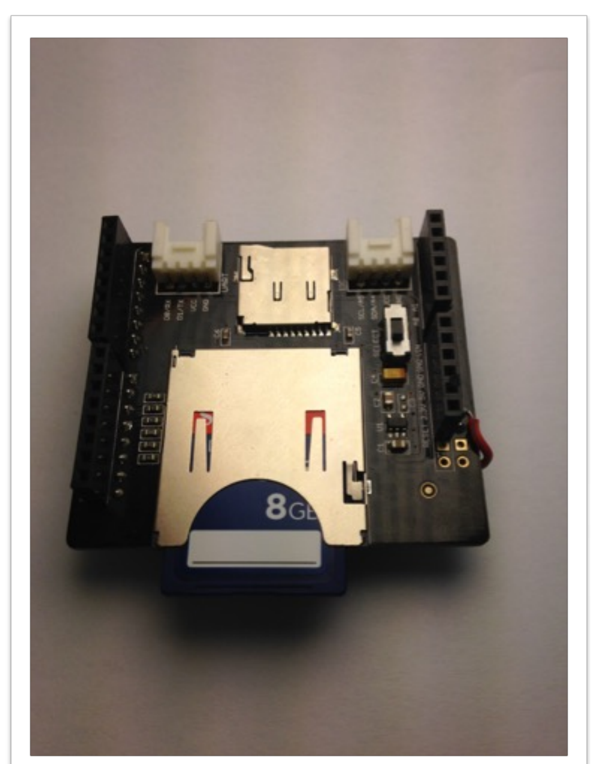
- Arduino Duemilanove
- SD Card Shield - Seed Studio
- 3 axis Gyroscope - Parallax
- 3 axis Accelerometer - Sparkfun
- Magnetometer - Sparkfun
- BMP Pressure Sensor - Sparkfun
- GPS Receiver - Inventek System
- Prototyping board



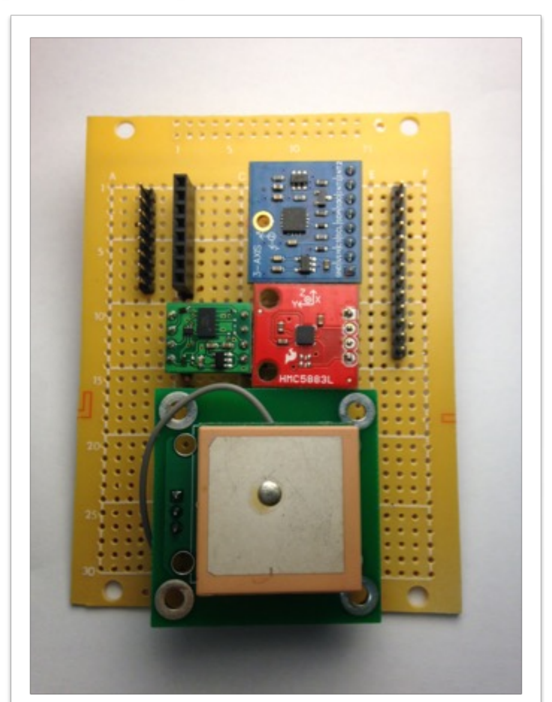
[Image 1.1] Entire Assembly



[Image 1.2] Arduino Duemilanove



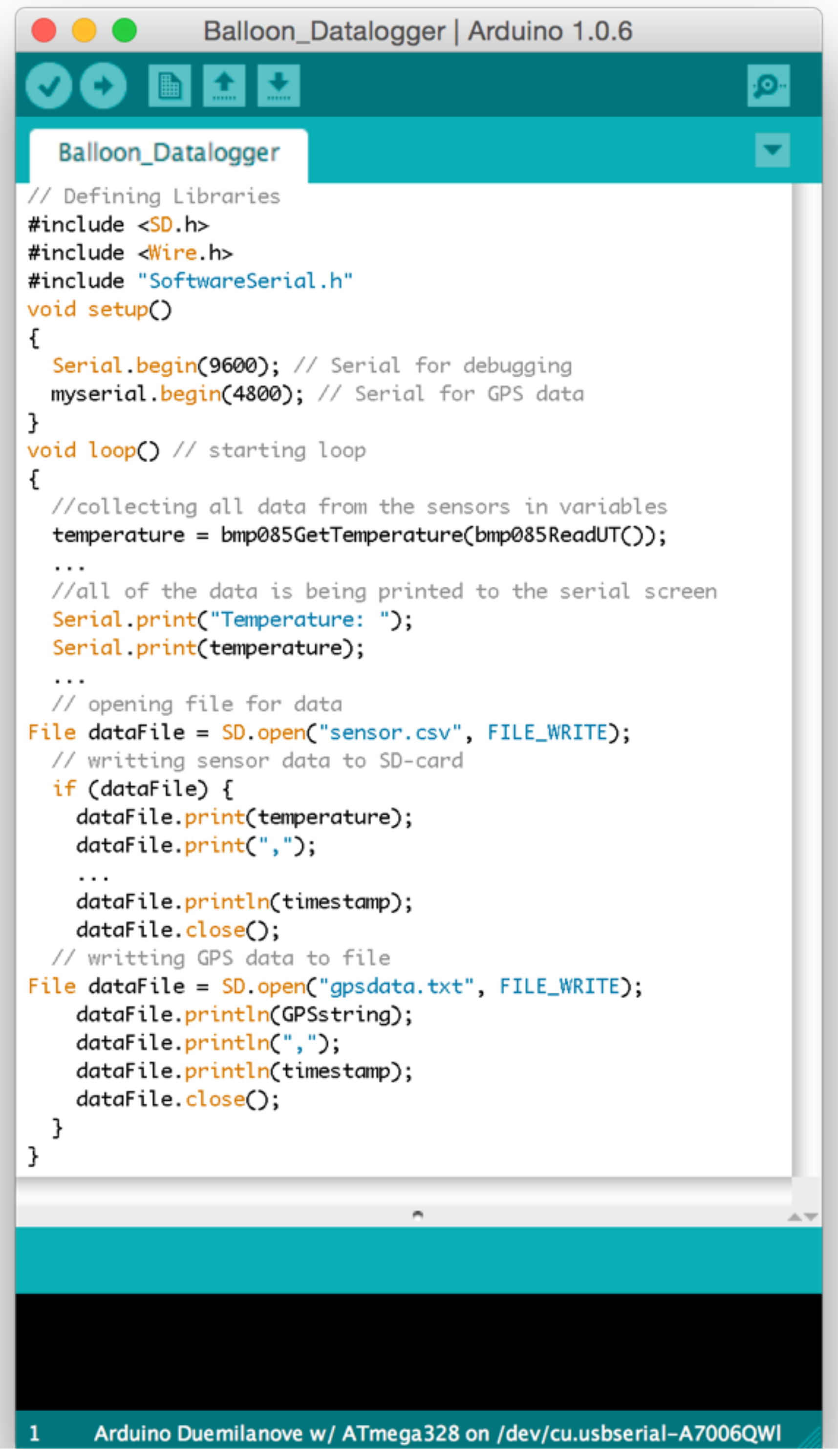
[Image 1.3] SD card shield -Seed Studio



[Image 1.4] Shield with all of the sensors mounted

Software

- The software that the Arduino micro-controller operates on is a C/C++ like language, which is compiled by the Arduino software.
- An Arduino script has two stages, the setup and the loop.
- Most sensors are connected through the I2C bus, which allows them to be addressed individually.
- The GPS (Global Positioning System) is connected through a serial connection, and the NMEA (National Marine Electronics Association) [4] is then read at a baud rate of 4800.
- The sensor data is being stored on the SD-Card with a timestamp as a comma separated variable file. The GPS data is being stored as a separate file with the same timestamp.
- All of the data is also sent through serial for debugging purposes as well as to check if all sensors are connected and the GPS has a satellite lock.

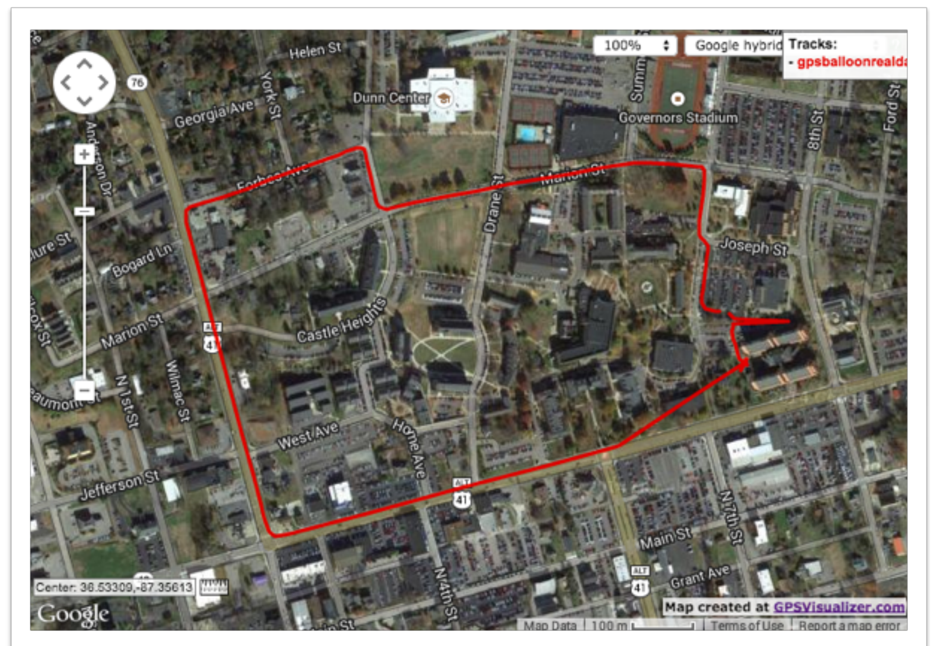



```
// Defining Libraries
#include <SD.h>
#include <Wire.h>
#include "SoftwareSerial.h"
void setup()
{
  Serial.begin(9600); // Serial for debugging
  myserial.begin(4800); // Serial for GPS data
}
void loop() // starting loop
{
  //collecting all data from the sensors in variables
  temperature = bmp085GetTemperature(bmp085ReadUT());
  ...
  //all of the data is being printed to the serial screen
  Serial.print("Temperature: ");
  Serial.print(temperature);
  ...
  // opening file for data
File dataFile = SD.open("sensor.csv", FILE_WRITE);
  // writting sensor data to SD-card
  if (dataFile) {
    dataFile.print(temperature);
    dataFile.print(",");
    ...
    dataFile.println(timestamp);
    dataFile.close();
  // writting GPS data to file
File dataFile = SD.open("gpsdata.txt", FILE_WRITE);
    dataFile.println(GPSstring);
    dataFile.println(",");
    dataFile.println(timestamp);
    dataFile.close();
  }
}
```

[Image 2.1] Arduino programming environment with simplified version of the Datta-logger's code.

GPS Data

The output data from the GPS (Global Positioning System) is received through a Serial connection. The NMEA (National Marine Electronics Association)[4] sentence is then being parsed into its latitude, longitude, and altitude components see [Graphic 2.1]. The coordinates are then plotted onto Google maps as seen in [Image 2.2]. The GPS has a range, stated on the data sheet up too 60,000 feet which is a concern as we will cross that line on a normal flight.



[Image 2.2] GPS data of a trip around Austin Peay's campus plotted onto Google Maps [2]

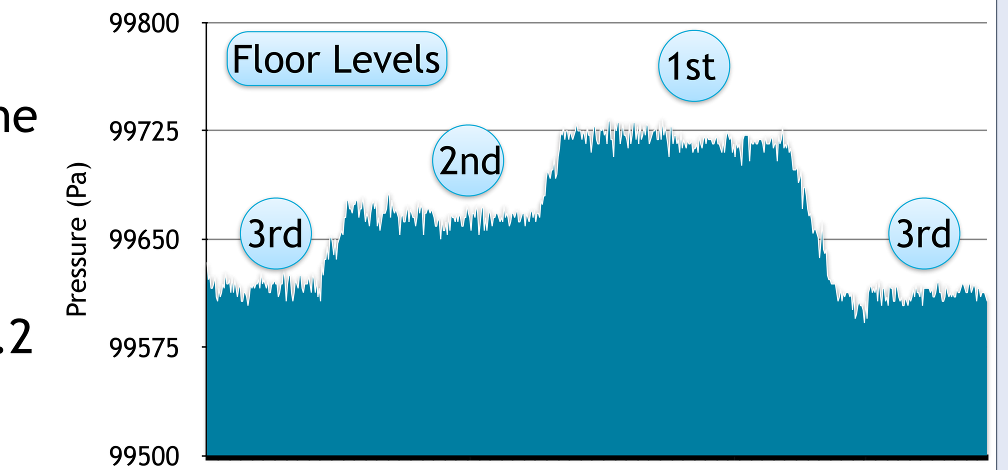
$GPGGA,064146.000,3630.9764,N,
08712.7477,W,1,08,1.4,167.1,M

[Graphic 2.1] GPS sentence Latitude, Longitude, Altitude

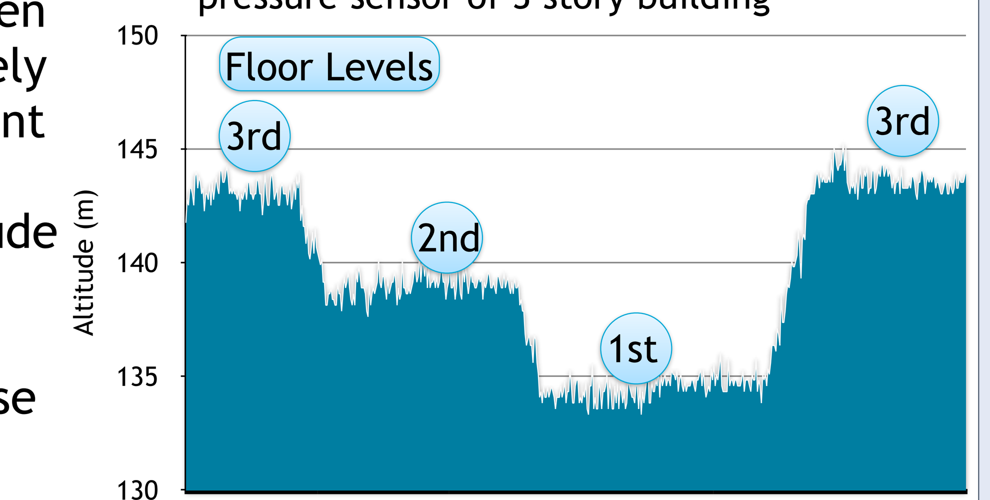
Sensor Data

Graph 3.1-3.2 show measurements taking by the pressure sensor which we use to determine altitude readings. 3.1 Shows the pressure readings where 3.2 shows a conversion to altitude, because pressure has an inverse relationship with altitude. The data taken shows that we can accurately distinguish between different floor level, which is a good accuracy when a high altitude balloon rises to 60,000-100,000 feet. We use this altitude measurement incase the GPS stops operating at those altitudes.

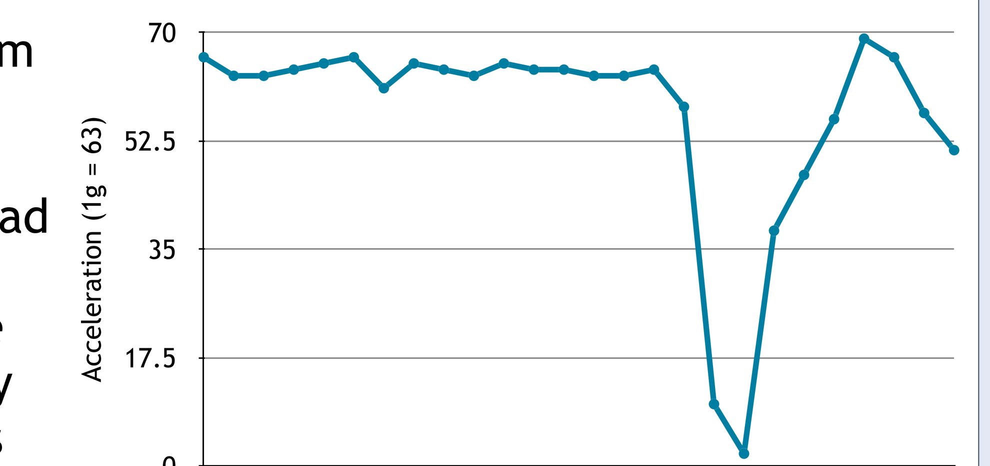
Graph 3.3 displays accelerometer readings from a two story free fall. This simulates changes in acceleration that the payload will undergo. The free fall can be seen in the negative spike as the acceleration by gravity goes to 0 and spikes up to normal values when it was caught again.



[Graph 3.1] Air pressure measured by the pressure sensor of 3 story building



[Graph 3.2] Converted from [Graph3.1] to display altitude (m).



[Graph 3.3] Accelerometer data from free fall in Hexadecimal

Future Work

This setup built the basic platform for any future balloon launches. We are planning on integrating a Geiger counter to measure radiation levels at specific altitudes, and orientations. External hardware that can be controlled on the flight such as a movable camera or sample collectors. We are planning a launch in the near future to test this equipment at the extreme conditions that are present in a near space environment.

References

- [1] "Arduino Software", http://Arduino.cc/
- [2]"Google Maps", http://www.gpsvisualizer.com/
- [3]"National Marine Electronics Association", http://www.nmea.org/
- [4] "Sparkfun Electronics", https://www.sparkfun.com/

Acknowledgements

- The parts for the project were purchased under the Tennessee Space Grant
- The project originated in PHYSICS 2468 a Freshman honors class to explore undergraduate research