MN Space Grant Ballooning Workshop – August 2013

Arduinos (3 configurations) Documentation

Spencer McDonald [mcdon773@umn.edu](mailto:mcdon773@umn.edu)

This documentation will explain the three ballooning payloads used in a workshop. There were three Arduino bundles collecting sensor data for the duration of the flight. These microcontrollers are easily programmed to collect and store data using various sensors and an SD card. There is a wide variety of sensors and applications of sensors, and this document should give some of the basics,

1. **Purpose**

These Arduino bundles were used as basic flight computers. They collected sensor data and recorded it onto an SD card that can be looked at after the flight. Each of the three Arduino bundles were a little different regarding what sensors they were used and the difficulty of coding the sensors. The code was provided for the workshop and the participants just looked at the code as they plugged in the sensors. PowerPoint presentations were provided to instruct the participants on the way to wire the Arduino bundles. The Arduino flight computer uses its own software to be program, and the majority of the sensors were purchased from Sparkfun Electronics or Adafruit. The data sheets for these sensors are easily found on the internet, or should be in a folder along with this documentation sheet.

1. **Sensor list**
   1. Analog Pressure Sensor: TrueStability SSC Series, Honeywell 015A24R
   2. Analog Humidity Sensor: HHI4030
   3. Analog Accelerometer: MMA7361
   4. Analog Temperature Sensor: TMP36
   5. Digital Accelerometer: ADXL345
   6. Digital Temperature Sensor: DS18B20
   7. Triple Axis Magnetometer: MicorMag, MAG3110
   8. GPS Module: Adafruit Breakout
   9. Sparkfun Real Time Clock Breakout: BOB-00099: DS1307
   10. SparkfunmicroSD Shield: DEV-09082
   11. Adafruit SD Shield with Real Time Clock
2. **Sensor implementation**

All of the analog sensors were used as input to the Arduino using the analog input ports. Simple equations given in the datasheets allowed for the analog signals to be converted into proper values. Any calibration necessary would have been done in the equations that converted the raw voltage to the proper value.

The Real Time Clock uses the I2C bus (SDA and SCL) for its communication with the Arduino. These are A4 and A5 on the Arduino UNO and D20 and D21 on the Arduino MEGA.

The SD and microSD shields use SPI and a chip select pin. The Sparkfun Shield uses D10-D13 for SPI and D8 for chip select, but you still need to set D10 as output for it to work. The Adafruit Shield is basically the same, but the chip select is D10.

The digital magnetometer uses SDA and SCL like the Real Time Clock, whereas the MicroMag uses ISP protocol on digital pins.

The digital temperature sensor uses a 1-Wire digital bus that can support multiple sensors on one pin. Each device is given an address, and can be read by the Arduino.

As the sensors are plugged in it is important to insure that the ports being used are not being used by some internal process. For example the SD shield uses five digital pins, yet the port on top of the shield still looks like it can be used. Problems can be avoided if you understand what exactly each sensor is doing and what port they are using.

1. **Running the code**

The code provided to the workshop is included in the folder with this documentation, the majority of the code is commented, and can be understood with a little time. The general format of the code is as follows.

First the libraries and variables are initialized before the setup loop begins. During the setup loop, the GPS, SD card, 1-Wire bus, etc. are initialized. After everything is initialized, the loop that runs for the entire flight begins. This loop reads all the sensors, and outputs all the data with a timestamp. For the details that goes into the code, look into the three code sketches. The comments should help to understand every step of the code.

1. **Flight readiness**

After the Arduino bundle is working and all ready to go, it is important to get the bundle flight ready. This consists of making sure that none of the wires or sensors will be pulled out under extreme flight conditions. This might mean hot glue or string used to ensure the bundle is secure. Before the Arduino bundle box is taped shut, make sure that the LED is flashing. Each LED flash means that the Arduino has taken data.