Balloon logger electronics

Philip Schmidt

Introduction

The balloon logger is a small data logging package containing an Arduino, SD card socket, BMP180 barometric pressure sensor and an Ublox Neo-6M GPS module which is mounted externally. All parts except the GPS are contained on a PCB measuring 80mm x 60mm and the whole logger runs from a standard 9V battery. The Arduino starts by receiving data from the GPS, from that it plucks out the UTC date, time, latitude, longitude, speed, course and altitude. After the GPS data has been collected the temperature and barometric pressure are read from a BMP180 sensor. Upon completing these tasks the Arduino formats all the collected information and writes it all onto the SD card. This process takes approximately 1.5 to 2 seconds and runs continuously. The Balloon Logger will not start writing to the SD card unless it has a valid GPS lock on its current position.

The Data Logger

An Arduino forms the basis of the balloon logger module. It is responsible for collecting data from the GPS unit and the BMP180 sensor. It collates the important fields of information and writes it all to a SD card. This process starts when the GPS gets a valid lock on location and continues until power is disconnected. Upon start up if a SD card is not detected the program halts and a red error LED is turned on. The Arduino does no further work until a SD card is inserted and the power is disconnected and re-connected.

Arduino software

There are always two main parts to any Arduino program, the setup and the loop function. In the "setup()" function the Arduino prepares the digital I/O, the hardware serial port, software serial port, BMP180 communications and the SD card. The SD is also checked to see if it is compatible and present. If the SD card test fails the Error LED is turned on the program halts. The "loop()" part of the program first spends 1 second receiving data from the GPS. It then sorts through the data and if a valid GPS fix is found then the data if decoded and the required information is retrieved. This information is then turned into a string ready for the SD card. After the GPS data is sorted the temperature and barometric pressure is read from the BMP180 sensor. Temperature has to always be read first as it is required for the pressure calculation. The temperature and pressure data is then added to data already received from the GPS and it all is written to the SD card. Once writing to the card has completed the full information string that was written to the SD card is sent out the serial port for end user checking. The string just written to the SD card is cleared and the program then returns to the start for another round.

Arduino libraries

The logger software uses two additional libraries not included in the main Arduino IDE release. The first is the "TinyGPS.h" library by Mikal Hart and the Sparkfun BMP180 library "SFE_BMP180.h". Please see the links at the end of the document on where to get these libraries. The other libraries used are "Wire.h" for the I2C bus communications to the BMP180 sensor, "SD.h" for writing information to the SD card, "SoftwareSerial.h" for receiving data from the GPS and "String.h" which assists with data value conversions to strings.

GPS information

The Ublox Neo-6m unit has been re-programmed to send out the NMEA strings $GPGGA and $GPRMC. These two NMEA strings contain all the information required by the data logging software.

SD Card

The SD card should be formatted to FAT32 and needs only to be 2Gb in size. For the data logger to work the SD card MUST have a test file already on the SD called

BLog.txt

Once the logging has been completed this file will contain rows of the following formatted data.

T,Date,Time,Latitude,Longitude,Altitude,Course,Speed,Pressure,Temperature

This data can be imported into Excel by opening BLog.txt. Press CRTL+A to select all text, CTRL+C to copy. Open an Excel file and select box A1 and then press CTRL+V. With all the data in column A selected the “data” pull down menu and then click "text to columns". In step 1 leave the "Delimited" radio button selected and press next. In step 2 tick the box next to "Comma" and un-tick the box next to "Tab". In step 3 click finish button. The data should now be sorted into the spread sheet columns.

Plotting the data in Google earth

Open the text file (or a copy of it) and insert a blank line at the top of the data. Type the following text into the new line and the top of the file.

type,date,time,latitude,longitude,alt,course,speed

This information is needed by the conversion website. Go to the following website

http://www.gpsvisualizer.com/

In the get started now box click the "Browse" button and find the BLog.txt file where ever it is on you computer. Select the file and press "Open". Next select the output file type from the pull down menu. Several output types are available however click on "Google Earth". You can now click the "Go!" button.

If you have done the above steps correctly the website will produce a file that can be opened by Google Earth. This will produce a line that represents the path taken by the logger unit.

Converting pressure to altitude

The air pressure recorded by the data logger can be used to calculate your altitude. You will however need the following three pieces of data. The first is sea level pressure, the second is the air pressure from the BMP180 sensor on the data logger board and the last is the temperature from the BMP180 sensor. Information about this calculation can be found at the following website.

http://keisan.casio.com/exec/system/1224585971

You will need to convert the formula found on the above website into an Excel formula. "Po" is the sea level pressure and this can be read from the Bureau of Meteorology website. This measurement will have to come from the Adelaide airport weather station as it is the closest to sea level available at 2m altitude. Go to the following link to this station's observations.

http://www.bom.gov.au/products/IDS60901/IDS60901.94672.shtml

By now you might have figured out the formula for excel and it could look something like this

=((POWER((Sea level pressure / BMP180 pressure),(1/5.257))-1)*(BMP180 temp +273.15))/0.0065

Apply this formula to the data in the spread sheet and plot the GPS altitude along with the calculated altitude. The plotted lines for the two data sets should follow each other roughly. The reason for this is GPS altitude is not accurate for civilian GPS units and air pressure is a tricky thing with lots of different variables affecting it.

Links

Additional libraries for Arduino

http://arduiniana.org/libraries/tinygps/
https://github.com/sparkfun/BMP180_Breakout

Revision History

Rev 1.0 17/8/2014 – Document creation.

Rev 1.1 25/8/2014 – Minor updates on SD card formatting information and overall grammar and wording.