High Altitude Weather Balloon Project: Project Log Part 3

Patrick A. Costello
Menlo School
50 Valparaiso Street
Atherton, CA 94027
patrick.costello@students.menloschool.org

ABSTRACT
This paper describes the current state of the high altitude weather balloon project as of March 4, 2008. The high altitude weather balloon project is a project run through the Applied Science Research class at Menlo School. The goal of the project is to launch a weather balloon into near space, approximately 100,000 feet.

Keywords
high altitude, weather balloon, ATV, PIC microcontroller

1. INTRODUCTION

Current work on the high altitude weather balloon project has consisted of software for the dsPIC30F3014 flight controller, hardware development, and radio licensing.

2. PARTS PURCHASING

Total Spent: $1,416.31
Total Spent with Tax: $1,532.45
Budget Remaining: - $1,232.45
The major portion of spending has gone to purchasing equipment for radio transmitting. Unfortunately, this is completely necessary because without it there would be no data recovery and the balloon would be lost. A complete list of purchased parts is in Appendix A.

3. GPS

After replacing the antenna on the GPS module, the GPS outputs meaningful data, which was accurate to the foot.

4. MEMORY

The 16MB flash memory chip was eventually chosen for this project. In order to provide enough data for the entire flight (maximum flight time estimated around 2 hours) with one data acquisition per second, two 16MB flash chips need to be used. The Atmel flash chips have chip select input pin, which allow two chips to be used, where only one is active at a time. Unfortunately, the flash memory chips require data to be written in 512 byte sectors. This means that data can either be stored, and then written every 512 bytes, or data can be written with extra space to fill the 512 bytes. Because there is enough memory for each data packet to take 512 bytes, this method was chosen to eliminate the complexity of maintaining multiple data packet buffers.

5. PIC MICROCONTROLLER

A new microcontroller has been chosen for this project.

5.1 PIC16F887

Unfortunately, the PIC16F887 was: 1. dead, and 2. not supported by the C compiler.

5.2 dsPIC30F3014

The new chip, dsPIC30F3014 not only is supported by the compiler but has a much larger range of features.

5.2.1 UART

The dsPIC30F3014 has 2 full UART drivers. One of these drivers also has selectable alternate pins. This allows full connection to the GPS and radio (or GPS and PC, depending on the operating mode).

6. FLIGHT CONTROLLER

The flight control software is close to completion, and the current version is in Appendix B.

6.1 Program modes

The flight controller has 3 different program modes, determined by jumper configurations in hardware. The three modes are: Acquire Data, Erase Flash, and Transmit to PC.

6.1.1 Acquire Data

The acquire data program mode handles all acquisition of hardware and in-flight operations. Operation is controlled by a combination of interrupts and hard-coded commands. The operation is as follows:

1. Initialize UART1 for GPS receiving.
2. Initialize UART2 for radio interaction.
3. Wait for GPS packet to be read.
4. Read A/D values.
5. Add a checksum.
6. Write packet to flash and radio.
7. Repeat steps 3-6.

6.1.2 Erase Flash

The erase flash program fully erases the flash, which must be done one page at a time. This will be done pre-launch to ensure maximum acquisition of data.

6.1.3 Transmit to PC

The transmit to PC program transmits all the data on the flash to the PC. This program also parses the data to eliminate all meaningless data. This increases the speed at which the flash can be read, because reading the entire flash over the UART would take approximately 30 minutes.

7. HARDWARE

7.1 Design

The hardware design schematics are in Appendix C.

7.2 A/D Hardware

Most of the A/D hardware requires amplification of the signal to provide the PIC with adequate data to provide the most precise signal. Amplification is done using low-current Op Amps.

8. RADIO

8.1 Balloon Radio

The radio finally chosen for transmission from the weather balloon was the Kenwood TH-D7A. This radio is preferred because it has a built-in terminal node controller (TNC). The TNC is what converts the data signals to radio signals (and vice-versa).

8.2 Recovery Vehicle Radio

The radio chosen for the recovery vehicle was the TH-D7A's mobile counterpart, the Kenwood TM-D710A. This radio is built just like the TH-D7A, but it runs off the car battery supply, and has higher sensitivity, allowing it to pick up and transmit better signal. The TM-D710A also has a built-in TNC with a serial port connection.

8.3 Radio License

The author of this paper now holds a Technician Class Radio License (the minimum license required to operate on amateur radio bands).

9. PACKET DATA SYNTAX

The data packets follow similar syntax to that of the NMEA protocol. The packets begin with the GPS data in the form "$GPRMC" followed by the data.