CPE 496 Spring 2016

Senior Design Final Report

NASA University Student Launch Initiative (Sensor Payload)

Payload Name: G.A.M.B.L.S

TEAM MEMBERS:
Jason G Renner - Project Manager
Tin T Tran – Hardware development
Patrick R Williamson - Software development
Michael A Bizanis – Software development
Course Instructor:
B. Earl Wells, Ph.D., P.E.
Electrical & Computer Engineering
wellsbe@uah.edu (256) 824-6047

Mentor & Customer
Mr. Jason Winningham, Comp. Sys. Engineer
Engineering Dept., UAH
Jason.Winningham@uah.edu (256)824-6132

Charger Rocket Work Design Team
University of Alabama in Huntsville
301 Sparkman Drive, Huntsville, AL 35899

GAMBS Team Members
Jason G Renner - Project Manager
Computer Engineering
jgr0007@uah.edu (865) 805-1806

Tin T Tran – Hardware development
Computer Engineering
ttt0006@uah.edu (256) 348 - 8352

Patrick R Williamson - Software development
Computer Engineering
prw0002@uah.edu (256) 783 - 8540

Michael A Bizanis – Software development
Computer Engineering
mab0035@uah.edu (256) 724 - 0170
Project Summary

GAMBLS will measure rotation, acceleration, direction, and atmospheric pressure while ascending through the atmosphere, beginning at launch and ending at approximately 5280 feet (1 mile). The payload will sample sensor data at a minimum of 500 samples per second and store this data on board. After apogee, the rocket will begin transmitting all data to a ground station so that there will be two copies of acquired data, one on the rocket and one at ground station. GAMBLS will synchronize data sampling by use of a GPS time stamp, and transmit data to ground using an RF transmitter.

Design Team Members:
Jason G Renner - Project Manager
Tin T Tran - Hardware Development
Patrick R Williamson - Software Development
Michael A Bizanis - Software Development

At the beginning of this project, all of the members of our team had similar experience with electronics. We had all taken the embedded systems class for programming microcontrollers and have taken circuits classes. However, none of the team members had built rocketry items or created their own circuit board before, so we spent considerable time learning. The main areas we lacked experience were in circuit board design, surface mount soldering, and Atmel Studio programming. We gained experience in surface mount soldering by creating the power boards with the help of John Jetton. We also became more familiar with Atmel Studio as we studied the code from last year’s senior design team. During the next semester, each member of the team will have to continue learning more about Atmel Studio, and learn how to design boards using Eagle.
Introduction

The USLI project is a competition sponsored by NASA to see which team of university students can create the best rocket which meets a range of requirements. Our requirements as the CPE team are to design a payload which meets the requirements of our customer, Jason Winningham. The College of Engineering at UAH is sponsoring the USLI team which we are a part of. These requirements were obtained through talks with the team leader for the UAH USLI team.

Experiment Concept

Payloads that measure flight characteristics are fairly common. However, storing the GAMBLS inside the rocket in limited space while also keeping it near the nose cone required creativity. The GAMBLS includes three circuit boards and must also be near the nose cone since it uses the pitot probe to monitor airspeed. Additionally, the restricted space for the payload inside the rocket calls for more originality in the design and as a result, every aspect of the design must serve a purpose. This is especially evident in the payload mounting bracket. The sensor boards are mounted on either side of the bracket while the battery had to be encased on the other side of the all-thread so that it would be protected from the rigors of flight. This design allows the boards to stay within their allotted footprint while also protecting electrical components.

Marketing Requirements

This payload has the following requirements:

- Shall operate under the rigors of flight
- Shall operate effectively for multiple launches
- Shall be able to idle on the launch pad for up to forty-five minutes and still be able to operate during flight
Shall store data on the rocket and transmit data to a ground station

Shall take data from an accelerometer, gyroscope, magnetometer, and atmospheric pressure sensor and have the capability to add more sensors

Engineering Requirements

The payload must contain the following instruments:

- 3-axis accelerometer (3 channels)
- 3-axis gyroscope (3 channels)
- 3-axis magnetometer (3 channels)
- One pressure sensor for ambient pressure (up to 15 psi)
- Develop a way to synchronize data between multiple copies of this payload in order to compare events between payloads.
- Five additional channels of data which may be used for sensors chosen by the USLI team
- Minimum 500 Hz sampling rate
- Sensors and five additional channels must have a 12-bit minimum resolution
- Capable of making 5 voltage measurements (0 - 5 V) at up to four feet from the payload.
- Noise tolerant digital or differential analog signaling required for the five additional channels and any other signals traveling more than five inches
- System shall provide a minimum of 1W power to sensors and associated support components (e.g. ADCs, bus transceivers) for remote sensors
- Capable of operating under a 50g acceleration loading
- Capable of operating under vibration experienced during a rocket flight.
- Have a means of confirming operational state when the rocket is on the launch pad
- Have a means of powering on and off via an external switch when the payload is in the assembled rocket
• Must be capable of being integrated with the rest of the rocket, powered up, and operational within 45 minutes
• Must be ready for re-flight (new batteries installed, data transferred to ground station, and empty memory) within 45 minutes
• Capable of operating for up to one hour in the powered up (standby) state on the rocket pad
• Capable of fitting inside of a 3.5-inch cylinder with a 4 inch height
• Weigh under 1 kg
• contain an independent power source (i.e. not require power from other systems in the rocket)

Background:

In general, the package on the market is large, heavy, and expensive. The market packages always have unnecessary features or lack of features for our needs.

For example, R-DAS (http://www.aedelectronics.nl/rdas/) has inadequate data rates and sensors, but it lacks an RF transmitter. BigRedBee (http://www.bigredbee.com/) has GPS with RF transmitter but lacks other sensors.

Our customer has specialized the requirements for the package, and we cannot find any pre-built solution package for them. So we decided to build our own circuit board to meet all requirements from our customer.

Solution

At the beginning, we wanted to use Raspberry Pi 2 Model B with ARMv7 quad-core with 1GB RAM with 64GB micro SD card, and breakout sensor boards off the market to build the system payload. After doing research, we found out that the Raspberry Pi is not capable of meeting our 500Hz sample requirement. Therefore, we decided to redesign circuit boards from the previous year to meet new requirements.
Hardware Design Description

Designing the payload to achieve a high sample rate while also fitting inside the allotted physical footprint was a challenge. Therefore the team chose to design and fabricate custom circuit boards, allowing for a payload with no unnecessary components or software overhead. This approach leads to a highly efficient electrical system which can sample, store, and transmit data at 500 Hertz.

The upper airframe team designed a mounting bracket for payload integration. The mounting bracket is a hollow square frame with the all-thread running lengthwise through the center. The sensor board is on one side of the all-thread, and the transmitter board with LiPo battery is on the other. The sensor board and transmitter board will be connected by header pins, and the sensor board is connected to the pitot board by a wire which runs up to the nose cone where the pitot probe is nested.

The data is collected using high precision sensors, which will work for multiple tests and flights. Testing and analysis of the GAMBLS payload will begin upon fabrication and programming of the design. The GAMBLS payload team will solder the circuit boards together in the lab, and program the AVR and SAM in Atmel Studio. The
GAMBLIS payload team works in pairs so that team members can check each other’s work.

The payload is divided into three circuit boards. The pitot probe board contains two 30 psi and one 150 psi analog pressure sensors, a signal amplifier for each sensor, and one 12 bit ADC. Requirements for the transmitter board consist of only a GPS tracker and radio. The sensor board contains four categories of measurements – acceleration, angular velocity (gyroscope), directional measurement (magnetometer), and altitude (barometric pressure). Logging of data acquired from the sensors requires non-volatile memory. This memory must be capable of storing enough data at the necessary speeds, while being resistant to shock and acceleration. A central processing unit is required to drive the sensors, change states based upon flight regime, and facilitate communication between the sensors, radios, and memory units. A secondary purpose is facilitation of communication between the package and the ground station.

The microprocessor central to the payload is the Atmel SAM4S, and is based on the ARM Cortex-M4 RISC processor. This is a 32-bit processor, with a maximum speed of 120 MHz, and 2 megabytes (MB) of flash storage and 160 kilobytes (KB) of random access memory (RAM).
A primary consideration for the processor is the ability to record up to 500 Hz data from a multitude of sensors, while determining position, velocity, and acceleration data in real-time and communicating via a radio to the ground station. The Atmel SAM4S meets these criteria, while possessing the ability to run at extremely low power. In addition, the SAM4S has an onboard USB controller, allowing for recorded data download via an external cable interface after recovery.

The processor is powered via the 3.3 volt bus. Communication with the sensors is accomplished via a SPI bus, with a clocking frequency of 10 MHz. In addition, the processor can communicate with the radio via serial UART, or low-speed SPI. The processor is programmed via a Serial Wire Debug (SWD) interface.

A GPS receiver is attached to the transmitter board. Due to ITAR regulations, functionality of the GPS is limited during the accelerated portions of flight. Because of this, the GPS data acquired will be logged, but not used for in-flight computations. The sensor is active during the recovery portion of flight, and is used primarily for position reporting to aid in the post-flight recovery of each stage. The GPS receiver communicates via a UART serial bus only, which is two orders of magnitude slower than the SPI bus used on the sensor board (maximum of 100 KHz versus 10 MHz).
To offload GPS communication from the main ARM processor on the sensor board, an additional AVR microprocessor is added to the transmitter board. This processor controls the GPS receiver and sets the appropriate messages required. The GPS serial data is parsed and buffered in on-board RAM. The AVR also communicates with the fuel gauge circuit and buffers the latest battery status. The AVR is connected to the ARM processor on the sensor board, and can pass the latest buffered GPS data and battery status when requested. Radio communication for each stage is also required and is accomplished with a 900 MHz XBee radio module.

The schematic files of each circuit board display all payload electronics and how they are connected. To ensure the payload does not activate itself, a switch will be added to disconnect the power supply from the battery, so the circuit can be broken by turning off the switch. This will stop power from flowing from the battery out to the system. The payload will be manually turned on when the rocket is prepared for launch. The switch will be accessible from the outside of the rocket through a small hole in the body tube. Figure 78 shows a block diagram of the sensor board.
Notice in that all sensors communicate via serial peripheral interface. The GAMBSLS payload team chose SPI because full duplex communication will be able to match the 500 Hz minimum sampling requirement.
The transmitter board contains the circuitry required for powering the entire GAMBLS payload. The block diagram for this can be seen above. The payload includes a 3.7 volt lithium polymer (LiPo) rechargeable battery, seen below. The battery voltage is not constant, but ranges from 4.2 volts when fully charged to approximately 3.4 volts when discharged. Because of this, two 3.3 volt regulation circuits were designed on this board, each capable of supplying up to 1 amp of current. The battery also requires a precise charging profile to avoid damage which is accomplished with a LiPo charging circuit. Charge status and remaining life of the battery is monitored via a LiPo fuel gauge circuit which communicates via I2C.
Pitot board schematic
PCB CAD Layout

Sensor board layout

PW / RF Board
Finished PCB Board

Sensor Board  RF/Power Board  Pitot Board
Tests and Verifications

The table below shows all of the payload tests. All sensor tests are done in an indoor environment over a period of time of a few hours, except for the pitot probe. Once data is acquired, it is analyzed to determine if any calibration is required to assure correct data values. The sensors tested by being set outside for a time are the MS5607-02BA, LSM9DS0, and H3LIS331DL. The sensors were tested by rotating, moving, and repositioning the payload while outputting data to a computer screen. The IMU partially worked. The accelerometer and gyroscope in the IMU function while the magnetometer does not return good data. The team was unable to determine the cause of this problem. The acceleration and rotation data were determined to be accurate.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Design Feature</th>
<th>Testing Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measure Atmospheric Pressure</td>
<td>MS5607-02BA</td>
<td>Test with computer</td>
</tr>
<tr>
<td>Measure Acceleration Low G</td>
<td>LSM9DS0</td>
<td>Test with computer</td>
</tr>
<tr>
<td>Measure Acceleration High G</td>
<td>H3LIS331DL</td>
<td>Test with computer</td>
</tr>
<tr>
<td>Measure Rotation</td>
<td>LSM9DS0</td>
<td>Test with computer</td>
</tr>
<tr>
<td>Measure Airspeed</td>
<td>LSM9DS0</td>
<td>Test with computer</td>
</tr>
<tr>
<td>Measure Orientation</td>
<td>LSM9DS0</td>
<td>Test with computer</td>
</tr>
<tr>
<td>Option of Adding Sensors to Board</td>
<td>Sensor Board</td>
<td>Inspection</td>
</tr>
<tr>
<td>Data recorded at 500 Hz</td>
<td>Software</td>
<td>Software Analysis</td>
</tr>
<tr>
<td>Data Transmits to Ground</td>
<td>XBee Pro 900HP</td>
<td>Flight Test</td>
</tr>
<tr>
<td>--------------------------</td>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Data Stored to Memory</td>
<td>S25FL512S</td>
<td>Inspection</td>
</tr>
</tbody>
</table>

The GAMBLS must take specific measurements during its ascent in order to meet the requirements. Specific tests are conducted for each sensor in an attempt to verify the functionality of the sensors before they are launched. A table listing each sensor with its individual success criteria, verification method, and status is listed in Table below:

<table>
<thead>
<tr>
<th>Verification Requirement</th>
<th>Success Criteria</th>
<th>Verification Method</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acceleration Measurement Low G</td>
<td>Sample rocket acceleration at 500 Hz up to 16g</td>
<td>GAMBLS Ground Test</td>
<td>Complete</td>
</tr>
<tr>
<td>Acceleration Measurement High G</td>
<td>Sample rocket acceleration at 500 Hz up to 100g</td>
<td>GAMBLS Ground Test</td>
<td>Complete</td>
</tr>
<tr>
<td>Atmospheric Pressure Measurement</td>
<td>Sample atmospheric pressure up to 1200 mbar</td>
<td>GAMBLS Ground Test</td>
<td>Complete</td>
</tr>
<tr>
<td>Orientation Measurement</td>
<td>Sample orientation up to 12 gauss</td>
<td>GAMBLS Ground Test</td>
<td>Failed</td>
</tr>
</tbody>
</table>
Safety Analysis

For this project, we have to fabricate three PCB boards, and soldering can create hot surfaces, fire, and smoke which can damage your eyes and skin; therefore, safety glasses are required when soldering the board. In addition, the rocket has fire risks during flight because of high speeds and fast temperature changes which can abrade the LiPo. The battery also requires a precise charging profile to avoid damage which is accomplished with a LiPo charging circuit. The payload itself poses very little danger as the boards operate at 3.3 volts.

The payloads are subjected to the comprehensive risk identification and assessment plan as the vehicle. The table below has been created to clearly identify areas of focus in this portion of the project life cycle. This table includes hazards to personnel as well as potential risks to payload components that could result in failure to meet NASA SL requirements.

<table>
<thead>
<tr>
<th>Hazard/Risk</th>
<th>Cause</th>
<th>Effect</th>
<th>Pre-RAC</th>
<th>Mitigation</th>
<th>Verification</th>
<th>Post-RAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAMBLS Failure Modes - Operational Phase</td>
<td>Sample rotation up to 2000 degrees per second</td>
<td>GAMBLS Ground Test</td>
<td>Complete</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sample airspeed up to 300 feet per second</td>
<td>Flight test</td>
<td>Not Started</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| Data collection failure | Software or component failure under flight loads | 1. Loss of ability to record data during flight
2. Inaccurate data | 2C | 1. Completion of successful testing prior to launch
2. Ensure ground and flight test results match calculated results |

| Electronics fail under high g-load | Software not tested to high levels of acceleration | 1. Loss of functionality of sensors or controllers | 2C | 1. Simulate high G-loads on all electronic equipment prior to flight |

| Pitot probe instrument becomes damaged during launch activities | Nose cone lands forcibly | Damage to pitot probe resulting in the inability to collect | 3D | 1. Ensure impact upon descent is minimized
2. Fabricate | 1. Vibration table testing |

|  |  |  |  |  | 2E |
| Li-Po Battery damage incurred during launch activities | Friction from vibration in flight | 1. Inoperable battery  
2. Potential chemical exposure | 2D | 1. Battery enclosed in structure in nose cone to minimize damage due to friction | 1. Flight testing | 2E |

**Major results**

Three copies of the GAMBLS were created, and were capable of reaching the requirements for the payload. The flash memory that encountered errors was corrected, and now is capable of writing data to the memory pages. The GAMBLS is capable of interfacing and reading data from the various sensors over the SPI bus. This data can be sent to the flash memory to be stored in the memory pages for storage.

The software has the basic structure set up for the three stages of flight, which are when the rocket is idle on the launch pad, during the rocket’s ascent, and after the rocket reaches apogee. During the first state, when the rocket is idle on the launch pad, the GAMBLS will not perform any significant computation, and will simply wait for the
rocket’s launch to be detected through the use of the accelerometer. During the second state, when the rocket is ascending, the GAMBLS will interface with each sensor to retrieve the data, and will store that data on the flash memory. For the last stage, which begins once the rocket reaches apogee, the sensors will no longer be used, and the GAMBLS will simply transmit the data stored in the flash memory to the ground station, to create a second copy of the data gathered during the ascent.

Lessons learned

Developing embedded code takes much longer than expected, and should be started as soon as possible. Designing and building the circuit boards does not take as long, and should be taken care of quickly. When ordering the parts for the circuit boards, it is important to check that the parts are the correct size, because not all parts will have the same footprint. Getting a part with the wrong size can cause a setback of a week or more before the circuit boards can be finished. When working on the software for the system, it is more efficient to work in pairs, as the progress of each can be assisted by the other, while working in groups of more than two can result in more difficult coordination.

Conclusion

The CPE 495 project we have chosen is very demanding because of the accelerated schedule and depth of knowledge we must learn. Our most difficult problem has been learning where to start. None of the team members knew anything about circuit board design, surface mount soldering, or rocketry in general. Fortunately, the CRW team along with Jason Winningham have explained many of the aviation aspects. Additionally, John Jetton and Dr. Wells have provided valuable advice regarding circuit board soldering and design along with guidance for the direction of the project. During
we have studied circuit board design more, and we were able to redesign the sensor board, and then write software to control the ARM MCU on the board. The hardest problem we are still having is how to write sensor data to memory.

At the end of CPE 496, the team has learned and gained a lot of experience on working with Eagle, Atmel Studio, surface mount soldering, rocketry, as well as the basic understanding and process in developing embedded software to program the circuit board.

Future Work

For those in the future who wish to work on this project, time should not be underestimated. Most of what must be done will take longer than expected, and some will prove to be much more difficult than expected. Tasks should be done as soon as possible, and should not be put off to a later date, doing so can lead to procrastination, and will leave the project in a state of rush upon reaching the end.

Hardware

The GAMBLS team has demonstrated at least one of the new sensor boards work by blinking LEDs on the board. Another board does not work and needs to be troubleshooting further. We believe this board has solder bridges that need to be found by going over the board with a microscope. The third board was taken apart to check resistance values. We found that three of the resistors had the wrong values. The two resistors for the voltage clamp in the USB circuit (R2 and R4) are 150ohm resistors instead of 27ohm resistors. These resistors are probably causing a signal problem with our USB interface to the computer. The third wrong resistor (R3), was a 168Kohm resistor instead of a 100Kohm. We are not sure how these three wrong resistors were placed, but the team recommends replacing them with the correct valued resistors for
future work on the boards. Replacing these three resistors should solve the last hardware problems.

Software

The codebase is partially ported to the new sensor boards. All of the SAM4S pins have been reassigned in the embedded code to work with the new boards, and there are functions written to test the sensors by outputting data to the computer screen. However, the computer cannot interface with the sensor board because the USB circuit does not work. The two USB resistors which are misvalued are causing this problem. If these two resistors are replaced, the current code for testing sensors should work. Future teams will have to implement the state machine that the payload will follow during flight.
Appendix

User Manual

Turning the Payload ON
1. Collect payload components, including sensor board, transmitter board, pitot board, and LiPo battery.
2. Connect transmitter board to sensor board by inserting pin connections between boards. There are total of four headers with 16 pins. Make sure the 3.3V pin on the transmitter board is connected to the 3.3V pin on the sensor board.
3. Connect the pitot board to the sensor board by connecting a wire between the connection headers on the sensor board and pitot board.
4. Complete the circuit on the sensor board by connecting the switch. This is done by twirling the ends of the copper wires on the sensor board.
5. Plug the LiPo battery into the terminal on the transmitter board.

Testing the Sensors
1. Connect the payload to the computer by connecting a micro USB cord to the outlet on the sensor board and the USB outlet on the computer.
2. Find the which port the payload is connected to by looking up the COM in the device manager. The payload is usually connected to a port between COM10-COM15.
3. Start hyperterminal using the following settings:
   Baud Rate - 115200
   Flow Control - None
4. Connect the Atmel ICE programmer to the sensor board using the JTAG header. Make sure the ICE is set for SAM and not AVR.
5. Start the project in Atmel Studio.
6. Download the sensor code to the sensor board by pressing the debug button in Atmel Studio.
7. Monitor the output on HyperTerminal for correct data.

In the main.c file of the code, at the bottom of the second while loop, there are several function calls that are commented out that are used to test each sensor. The name of the function is indicative of which sensor it will test. A one second delay is placed between each function call, this is used so that the data does not get displayed on HyperTerminal too quickly to read. Simply uncomment the function for the sensor you wish to test, as well as one delay for each function call.

**Estimated Cost and Schedule**

CPE 496 Project Cost Evaluation

<table>
<thead>
<tr>
<th>Type</th>
<th>Item</th>
<th>Cost ea.</th>
<th>Qty required</th>
<th>Total Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microcontroller</td>
<td>Atmel ATSAM-4S</td>
<td>6.17</td>
<td>2</td>
<td>12.34</td>
</tr>
<tr>
<td>Accelerometer, 3-axis, ±100G</td>
<td>STMicroelectronics H3LIS331DL</td>
<td>10.96</td>
<td>2</td>
<td>21.92</td>
</tr>
<tr>
<td>Accelerometer, 3-axis, ±12G</td>
<td>STMicroelectronics LIS331HH</td>
<td>5.37</td>
<td>2</td>
<td>10.74</td>
</tr>
<tr>
<td>Barometric Pressure / Altimeter</td>
<td>Measurement Specialties MS5607-02BA03</td>
<td>5.42</td>
<td>2</td>
<td>10.84</td>
</tr>
<tr>
<td>Flash Memory IC</td>
<td>Spansion S25FL256S</td>
<td>3.44</td>
<td>4</td>
<td>13.76</td>
</tr>
<tr>
<td>Gyroscope, 3-axis</td>
<td>Bosch BMG160</td>
<td>6.25</td>
<td>2</td>
<td>12.50</td>
</tr>
<tr>
<td>Magnetometer, 3-axis</td>
<td>STMicroelectronics LIS3MDL</td>
<td>1.79</td>
<td>2</td>
<td>3.58</td>
</tr>
<tr>
<td>Description</td>
<td>Model/Brand</td>
<td>Unit Price</td>
<td>Quantity</td>
<td>Total</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>------------------------------</td>
<td>------------</td>
<td>----------</td>
<td>--------</td>
</tr>
<tr>
<td>ADC, 4-channel</td>
<td>Maxim Integrated MAX11060</td>
<td>6.48</td>
<td>2</td>
<td>12.96</td>
</tr>
<tr>
<td>GPS Module</td>
<td>GlobalTop Technology FGPMMOPA6H</td>
<td>29.95</td>
<td>1</td>
<td>29.95</td>
</tr>
<tr>
<td>Radio Module</td>
<td>Xbee Pro S3B 900HP</td>
<td>39.00</td>
<td>2</td>
<td>78.00</td>
</tr>
<tr>
<td>LIPO Battery, 2000 mAh, 3.7V</td>
<td>MIKROE-1120</td>
<td>13.51</td>
<td>2</td>
<td>27.02</td>
</tr>
<tr>
<td>LIPO Recharging IC</td>
<td>Microchip MCP73831T-2ATI/OT</td>
<td>0.76</td>
<td>2</td>
<td>1.52</td>
</tr>
<tr>
<td>Battery Fuel Gauge</td>
<td>Maxim DS2782E+</td>
<td>7.46</td>
<td>2</td>
<td>14.92</td>
</tr>
<tr>
<td>3.3V Regulator</td>
<td>Micrel MIC5219-3.3YM5 TR</td>
<td>1.54</td>
<td>2</td>
<td>3.08</td>
</tr>
<tr>
<td>60 PSI Pressure Sensor</td>
<td>Honeywell NBPDLNN015</td>
<td>12.32</td>
<td>1</td>
<td>12.32</td>
</tr>
<tr>
<td>100 PSI Pressure Sensor</td>
<td>TruStability NSCDANN100</td>
<td>29.91</td>
<td>1</td>
<td>29.91</td>
</tr>
<tr>
<td>Passive components (R, C, L, etc.)</td>
<td>Various</td>
<td>30.00</td>
<td>1</td>
<td>30.00</td>
</tr>
<tr>
<td>Wires, Cables, Connectors</td>
<td>Various</td>
<td>25.00</td>
<td>1</td>
<td>25.00</td>
</tr>
<tr>
<td>Solder Paste</td>
<td>Zephytronics SPE-0012</td>
<td>24.75</td>
<td>1</td>
<td>24.75</td>
</tr>
<tr>
<td>PCBs, 2-layer, 1 oz copper</td>
<td>Various, 2&quot;x3&quot;</td>
<td>30.00</td>
<td>6</td>
<td>180.00</td>
</tr>
<tr>
<td>Item</td>
<td>Model/Link</td>
<td>Quantity</td>
<td>Unit Cost</td>
<td>Total</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-------------------------------------</td>
<td>----------</td>
<td>-----------</td>
<td>--------</td>
</tr>
<tr>
<td>Antenna, 900 MHz omnidirectional</td>
<td>Abracon APAMS-104</td>
<td>1</td>
<td>6.00</td>
<td>6.00</td>
</tr>
<tr>
<td>Antenna, 900 MHz, directional</td>
<td>Data Alliance AYA-9012</td>
<td>1</td>
<td>21.99</td>
<td>21.99</td>
</tr>
<tr>
<td>Shielded Cable</td>
<td><a href="http://www.mcmaster.com/#shielded-communication-wire=tzr1zz">http://www.mcmaster.com/#shielded-communication-wire=tzr1zz</a></td>
<td>15</td>
<td>0.42</td>
<td>6.30</td>
</tr>
<tr>
<td>Structural Components</td>
<td>Hardware</td>
<td></td>
<td>300.00</td>
<td></td>
</tr>
<tr>
<td>Ground Component</td>
<td>FTDI Serial to USB</td>
<td>1</td>
<td>25.00</td>
<td>25</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td></td>
<td></td>
<td>951.72</td>
<td></td>
</tr>
<tr>
<td><strong>TAX</strong></td>
<td></td>
<td></td>
<td>85.65</td>
<td></td>
</tr>
<tr>
<td><strong>SHIPPING</strong></td>
<td></td>
<td></td>
<td>47.59</td>
<td></td>
</tr>
<tr>
<td><strong>Grand Total</strong></td>
<td></td>
<td></td>
<td>1084.96</td>
<td></td>
</tr>
</tbody>
</table>

- Total cost to produce three sensor boards, three transmitter boards, and three pitot boards: $1084
- Cost of producing one functional payload package: $362

Expected Cost of Mass Production

The cost of producing one payload was $362. However, it is worth noting that many of the parts we ordered are cheaper when ordered in bulk. For instance, 100 resistors cost about one penny, while ten resistors cost about fifteen cents, according to
digikey’s website. Additionally, we ordered three SAM4S microcontrollers for $5 per microcontroller. If you order 100 microcontrollers, you are charged $3.50 per microcontroller. Every part we ordered on digikey has it’s price drastically reduced if you order by quantity of 100. With this method in mind, it is reasonable to conclude that the price of producing 100 functional payload packages will reduce the cost of each payload package by half.

In other words, we believe the cost of producing functional payload packages for mass production will be about $180 per functional payload package, or $18000 to create 100 functional payload packages.

**Project Schedule**

September 1-20

Finalize Requirements Document

September 21 - October 15

Choose microcontrollers and sensors

October 16-31

Familiarize with previous year’s progress

November

Demonstrate project feasibility

January 6-15

Design Sensor Board
Critical Design Review

January 20-29

Order parts

Begin ARM embedded Code

January 30-February 11

Solder circuit boards

Finish Software

February 12-14

Payload Test Flight

February 15-March 10

Correct Software Problems

Acceptance Tests

March 11-14

Flight Readiness Review

March 15

Final test launch

April 16

Competition Launch

- Total time to produce payload package: 7 months