Smart Water Rockets (SWR)
for
Space Trek Space Camp at
Kennedy Space Center

Group 6

Gabriel Fernandez: Electrical Engineering

Kyle Moran: Computer Engineering

Connor Phillips: Electrical Engineering

Matthew Turner: Electrical Engineering

Sponsor

Space Trek

1. Executive Summary 1
2. Project Description 2
2.1 Project Motivation and Goals 2
2.2 Objectives 7
2.3 Requirements Specifications 8
2.4 System Summary 9
3. Project Research 12
3.1 Existing Similar Technologies 12
3.1.1 Jolly Logic AltimeterThree (SpaceTrekAltimeter) 12
3.1.2 Perfectflite Firefly 13
3.1.3 Software in use at Space Trek 14
3.1.4 Space Trek Rocket Nose Cones 14
3.2 Strategic Components 15
3.2.1 Rocket Processing 16
3.2.1.1 TI MSP430 16
3.2.1.2 Arduino 17
3.2.1.3 Raspberry Pi 18
3.2.2 Sensors 18
3.2.2.1 Pressure Sensors 18
3.2.2.1.1 BMP180 19
3.2.2.1.2 MPL3115A2 20
3.2.2.1.3 BMP280 20
3.2.2.2 Accelerometer 21
3.2.2.2.1 MMA8451 23
3.2.2.2.2 LIS3DH 23
3.2.2.2.3 ADXL345 24
3.2.3 Batteries 25
3.2.3.1 Lead-acid Batteries 26
3.2.3.2 Nickel Batteries 27
3.2.3.3 Lithium Batteries 28
3.2.3.4 Voltage Regulators 30
3.2.4 Launchpad Automation 30
3.2.4.1 Launchpad Microcontroller 31
3.2.4.1.1 Arduino 31
3.2.4.1.2 TI 31
3.2.4.2 Motor 32
3.2.4.2.1 Servo Motors 32
3.2.4.2.2 Stepper Motors 33
3.2.4.3 Air Compressor 34
3.2.4.4 Tubing and Valves 35
3.2.6 Briefcase 36
3.2.6.1 Briefcase Controls 37
3.2.6.1.1 Ignition Control 37
3.2.6.1.2 Keypad 38
3.2.6.1.3 Launch Trigger 39
3.2.6.1.4 Configurations 41

3.2.6.2 Display 42
3.2.6.2.1 Display Technologies 43
3.2.6.2.2 Tablet Approach 45
3.2.6.2.3 Tablet Options 46
3.2.6.2.4 Microcontroller Approach 48
3.2.7 Communications 49
3.2.7.1 Radio 50
3.2.7.1.1 Transmission 50
3.2.7.1.2 Cellular Signals 50
3.2.7.1.3 Spectrum Allocation 51
3.2.7.2 Wi-Fi 51
3.2.7.2.1 Transmission 52
3.7.2.3 Bluetooth 52
3.2.7.3.1 Specifications 53
3.2.7.3.2 Versions 53
3.2.8 Software Components 54
3.2.8.1 Code Composer Studio 54
3.2.8.2 Arduino IDE 55
3.2.8.3 Android Studio 56
3.2.8.4 Apple IDE 57
3.2.8.5 Windows IDE 58
3.2.8.6 Eagle 59
3.3 Related Technologies 61
3.3.1 Power Management 61
3.3.1.1 Power Banks and Large Capacity Rechargeable Batteries 61
3.3.1.2 Solar Panels 62
3.3.2 MEMS 62
3.3.3 Automated Launch Pad 63
3.3.4 Radar Altimeter 63
4. Related Standards 64
4.1 I2C Standard 64
4.2 Launch Standards 64
4.3 Health and Safety Standards 65
4.4 Wi-Fi Standards 65
5. Related Constraints 66
5.2 Microcontroller Constraints 66
5.3 Communication Constraints 66
5.4 Health and Safety Constraints 67
5.5 Economic and Time Constraints 68
6. Software Design 69
6.1 Software Decisions 69
6.1.1 Rocket Microcontroller Software Summary 69
6.1.2 Mobile applications 71
6.1.2.1 Android 71
6.1.2.2 iOS 72
6.2 Software Class Descriptions 72

6.2.1 UML Diagrams 72
6.2.1.1 Sequence 73
6.2.1.2 Sequence Revised 75
6.2.1.3 Class 76
6.2.1.4 Class Revised 78
7. Hardware Design Details 79
7.1 Hardware Designs 79
7.1.1 Rocket Launchpad Designs 79
7.1.1.1. Existing Structure Modification 80
7.1.1.2. Motor Mounting Design 82
7.1.1.3 Fuel and Pressurization Flow 84
7.1.1.4 Waterproofing the Electronics 85
7.1.1.5 IPX Certifications 86
7.1.2 Briefcase Design 88
7.1.2.1 Power 89
7.1.3. Altimeter Design 90
7.1.3.1 Microcontroller 90
7.1.3.2 Pressure Sensor 91
7.1.3.3 Accelerometer 91
7.1.3.4 Design Constraints and Parameters 92
7.1.3.5 Design Schematics 92
7.1.3.6 Power 97
7.1.3.7 Bluetooth 97
7.1.3.7.1 Block Diagram 100
7.1.3.7.2 Schematic 101
7.1.3.8 Future Goals and Hardware Revisions 102
8. Project Prototype Construction and Coding 103
8.1 Breakout Boards 103
8.2 PCB Vendor and Assembly 104
8.4 Final Coding Plan 105
9 Project Testing 105
9.1 Hardware Test Environment 106
9.2 Hardware Specific Testing 106
9.2.1 Communications Network Testing 106
9.2.2 Automated System Testings 107
9.2.3 Launch Sequence Testing 109
9.2.4 Keypad Testing 109
9.3 Software Testing Environment 111
9.3.1 Motherboard 111
9.3.2 Processor 112
9.3.3 Random Access Memory 112
9.3.4 Storage 112
9.3.5 Graphics Card 112
9.3.6 Power Supply 112
9.4 Software Testing Procedure 113
10 Administrative Content 116

10.1 Budget and Finances 117
10.2 Progress Updates 117
10.3 Milestones 118
10.4 Group Responsibilities 119
10.5 Sponsor: Space Trek 121
Appendix A: Sources 123
Appendix B: Permissions 124

1. Executive Summary

Atlantis Education Services at Kennedy Space Center offers two space camp programs; Camp KSC (Kennedy Space Center) and Space Trek. The Smart Water Rocket system (SWR) will be an upgrade to the water rocket program currently in use by the Space Trek space camp program. The SWR will be reusing Space Trek's rocket design both to cut down on costs and because they already have a well written pre-flight analysis program. The SWR aims to replace or upgrade three main components of the old system: launch data collection, post-launch data analysis, and launch experience.

In the current system the students can vary a few parameters of their flight including: water volume, air pressure, vertical launch angle, and water release valve size. They utilize a pre-flight analysis program to choose the parameters they think will get them closest to the goal the instructors choose (speed, accuracy, etc.). Once they have all of their data, the launchpads are set up in a field behind the education center. The students then use beakers to fill their rockets with water, install their nozzle of choice, install the altimeter in the nose cone, then manually load the rocket and set the launch angle. Finally, they pressurize the rockets and launch them using a large red button. After launch, the altimeter is retrieved and the data read manually off of a small screen.

The SWR will improve this process by upgrading a few key steps in the procedure. The pre-flight analysis will remain the same and the students will still have the opportunity to attempt to find the correct parameters for launch but the physical launch process will be much more efficient. One of the main components of the new system, the launch briefcase, will control all of the automated portions of the launch from a cart near the launch pads and display flight data as close to real time as possible. The briefcase will have controls to: operate an air compressor that will pump water into the rocket and pressurize it automatically, precisely set the vertical angle of launch, and finally launch the rocket using a launch process that has been designed to be more engaging. The team has designed a new altimeter to work with the briefcase that will transmit flight data to the briefcase. The data will be displayed on a large, easy to read screen and will include a model of the flight trajectory. The SWR system will perform all of these tasks efficiently and reliably while remaining portable and easy to use by both the students and instructors.

The SWR team is dedicated to helping Space Trek overhaul their current system and create a system to engage students of all ages and backgrounds. Hopefully our combined efforts will encourage students to learn more about STEM (science, technology, engineering, and math) and even pursue a career in the space industry.

2. Project Description

NASA's Kennedy Space Center has been the center of technological innovation and space exploration since its founding in 1958 as a response to the Soviet Union's successes in space flight. The American people were immediately enthralled with the Space Race and the first golden age of space exploration began. The Apollo missions to the Moon garnered a worldwide interest in NASA and introduced children all across the globe to the wonders of manned space travel.

There has been a demand for a space camp that children can attend to learn more about rocketry and space in general since the first manned space launches but NASA did not start offering space camps until 1982. There are two space camp programs at Kennedy Space Center (KSC) currently: Camp KSC and Space Trek. Atlantis Educational Services manages the Space Trek program at KSC and their goal with the program is to provide students from across the globe in middle school, high school, and all levels of college with the opportunity to learn about the incredible history of space travel at KSC and how the technologies required are being developed today. The program is the first introduction to STEM (science, engineering, and mathematics) topics for many students and can spark an interest in the space industry in students of any age. Atlantis Educational Services is committed to educating both the students and educators by constantly developing new technology to improve their already stellar programs. For these reasons, this team chose to ask Atlantis Educational Services for sponsorship to improve an aspect of the Space Trek camp program.

2.1 Project Motivation and Goals

Space Trek has three separate programs: near space exploration, robotics, and rocketry. All three programs are day camps, the students arrive in the morning and leave in the late afternoon. The near space exploration program guides the students through designing experiments for a weather balloon that they eventually launch and recover. It is the most in depth camp and requires five days to complete. The other two camps only require three days but still involve plenty of design and the eventual piloting of a robot or launching of a rocket. Both the near space exploration camp and the robotics camp are currently undergoing improvements to be completed by summer 2016 at the latest so when the team approached Space Trek about potential projects the rocketry program was the first choice.

The rocketry program, described in the opening section, currently has several small faults. The program breaks the students into groups of three to five depending on the total class size. There are often more than one hundred students in a class resulting in upwards of twenty five individual rockets. Each rocket, at peak efficiency, takes 4 minutes to launch once the students are ready. More often than not it is difficult to keep the students focused and each step takes several minutes longer. A good turnaround time for these launches is between 5 and 10 minutes. The students are allowed two or three launches, depending on the total class size, and the launch cart has two pads on it which allows two groups to launch almost simultaneously. To launch twenty five rockets at average time efficiency (7 minutes) takes 3 hours with both pads and 6 hours with only one. The steps to complete for each launch are summarized in the **Table 1-1** below, along with the estimated time to complete each step.

Step	**Time to Complete**
The altimeter must be loaded into the rocket and setup to read data	30 seconds
The parachute must be loaded into the rocket	30 seconds
The rocket must be fueled, secured to the launch pad, the angle set, and the parachute release mechanism must be readied for launch	1-1.5 minutes
The rocket must be pressurized	1-3 minutes

Table 1-1: Time estimates to complete each step in the launch sequence

If there are more than one hundred students then the current manual rocket launches become very time consuming, resulting in less time for the other activities at KSC. The Space Trek team has stated that they like the option to do everything manually because it illustrates the physical labor required in launching the rockets but that they would like the option to automate some of the processes to decrease the turnaround time for launches. The launches are performed in a large field (see **Figure 1-1**) behind the education complex at KSC so the educators and children end up having to spend several hours in the sun. Faster turnaround time will be a relief to the educators on hot days while still giving the students the opportunity to complete all of their launches.

The field is used because of its proximity to Space Treks offices and classrooms. Sometimes the field is used for other purposes and the launches must take place elsewhere on the premises. The field is used for large events in the summer like the Robotic Mining Competition and is unavailable for several weeks. The launch carts must be able to travel to the field with ease but preferably they should be able to go farther. There are several other sites where launches take place and they are all within walking distance of the education complex but they are 3-4 minutes farther away. The educators pull the carts from location to location to reduce the chance of accidents. The carts only need to be light enough to be moved by an adult.

Figure 1-1: Launches are performed at this field behind the education complex that Space Trek is located in. The students and educators stand on the concrete and the pads are set up in the field a safe distance away.

Not all of the steps in the launch sequence can be optimized with automation. The parachute must still be loaded and readied manually, there is simply no way to modify their current system to simplify or quicken this step. The rocket must also be secured to the launch pad manually for the same reasons. The other steps in the process can all be improved. A new altimeter will be designed to send data directly to a portable launch computer (launch briefcase) to be viewed by the students. This will relieve the educators of the responsibility of logging the data manually and remove the need to reset the altimeter. Ideally, it will be cheap enough to produce one for every rocket. In the event that it is not that cheap to produce it must be very simple to move the altimeter from rocket to rocket. Automating this task will reduce the chance for user error significantly and improve the turnaround time by at least twenty seconds on that step. Streamlining this process will hopefully reduce the chance of the students losing interest and becoming distracted which will further improve the turnaround time.

The launch angle is particularly difficult to set accurately. The current system uses a thumbscrew to hold the pad in place at the intended angle. This is inefficient because the screw is always either too tight to remove for the next launch or too loose to hold the rocket steady. It is incredibly dangerous for the rocket to fall and launch in the horizontal position. The full force of the rocket is directed horizontally and it becomes a torpedo like projectile. If it were to strike a guest or student there is a massive chance for injury. This must be improved upon in the new system. The team's goal is to use a motor on each launch pad to control the angle of launch. This will not only improve the overall safety of all of the launches but also reduce the turnaround time because neither students nor educators will have to deal with the thumbscrew that currently causes so many problems.

Space Trek discussed the possibility of adding horizontal movement to the launch pad automations. A second set of motors would be used to turn the rockets a few degrees clockwise or counterclockwise with respect to the ground. The team discussed this idea in detail before deciding that it is simply not worth the extra effort in design. The second motor will add excess complexity to the motor mounting design. Both motors would have to work in conjunction with each other to rotate both axis. The team has limited mechanical experience so this would end up being a massive resource sink. The rockets would have to be severely limited in their turning radius to avoid hitting each other on the pad. The rockets are quite large and close together so they would only be able to turn to 1pm clockwise and 11pm counterclockwise to avoid collision. There is no benefit to being able to launch the rockets at a slight horizontal angle. It will provide no boost to speed or altitude and it will be no safer. The team discussed the issue with Space Trek and they agreed that it would offer limited benefit to the program while taking excess resources that could be used to improve other systems of the SWR.

Fueling and pressurizing the rocket will be the trickiest part of automating the launch pad. The easiest way to do this is to use gravity and pressure to move both air and water into the rocket while it is on the launch pad. The most difficult constraint to work around will be ensuring the whole system is still portable. The compressor must be robust enough to perform the task quickly relative to the manual option, but remain small enough to be solar or battery powered and stored on a cart. The students often spend a lot of time debating about the fuel level in the rocket, reducing the turnaround time. If the process is automated then the desired amount of water will be precisely delivered to the rocket. Similarly, the pressure is currently provided by bike pumps attached to the carts shown in **Figure 1-2**. There is no way to precisely deliver pressure to the rocket using this system. Since the pump is manual, it is up to the user to regulate how much air goes into the rocket. It is very easy to get carried away and pump more air than intended into the rocket. It is also very time consuming to pump up the rocket to the maximum pressure of 90 psi. The students often get tired when pumping and have to take breaks, which adds more to the turnaround time. Automating this process will save the most time of all of the steps. It takes the educators at least a minute to pressurize the rockets and it takes the students up to three times as long. A table of the ideal launch procedures is shown below as **Table 1-2**.

Figure 1-2: The bike pump is attached using a heavy duty twist tie. The students use it to pump air into the rocket through vinyl tubing (the blue tubing behind the pump). The pump has a pressure gage so the students can track the amount of pressure in the rocket.

Step	**Time to Complete**
The altimeter must be loaded into the rocket and setup to read data	5-10 seconds
The parachute must be loaded into the rocket	30 seconds
The rocket must be fueled, secured to the launch pad, the angle set, and the parachute release mechanism must be readied for launch	30-60 seconds
The rocket must be pressurized	1-2 minutes

Table 1-2: Time goals to complete each step in the launch sequence

Reducing the tediousness of the launch will help retain the student's interest but it doesn't help engage uninterested students. A launch briefcase was proposed to increase the functionality of the system as well as make it look more appealing to the students. The idea for using a briefcase was proposed by Space Trek during our initial design meeting. They had a spare briefcase that they wanted to repurpose and expressed their desire to make a professional looking launch case for the rockets. The briefcase is a medium sized metal container with a key lock. It has the look that they are going for and is large enough to contain any hardware required for the project. At this time the concept of a launch computer to perform data processing on site was developed as well leading to the project name Smart Water Rocket. Smart refers to the launch briefcase and computer controlling the system and water rocket refers to the water rockets being launched.

The briefcase must contain all of the systems controls. It will be used by the students and educators so it needs to have a user friendly interface and still look professional and appealing. Space Trek allowed the team plenty of freedom in designing the internals of the briefcase. The team decided that the launch process needs to engage multiple members of the group by including a several step button sequence to be completed before launch. First, the briefcase must be opened, the rocket primed, then it can launch. The priming of the rocket will include automating the fuel and pressure and setting the launch angle. This will be achieved using a keypad for numeric entry. As each parameter is entered, the SWR will perform its tasks automatically.

Each team will have a unique identifier so that the data for each launch can be kept separate. This identifier will be entered into the briefcase by each team before launch. Then, a series of switches will be pressed followed by a large button to initiate the launch. The launch will be accompanied by a countdown and success signal. In the air the newly designed altimeter will communicate with the launch pad for as long as it is in range. When it is out of range the altimeter will store the data on board and begin transmitting again once it is in range. The computer will then process this data and display the maximum height, speed, and acceleration. It will then simulate the rocket's trajectory so the students can the flight path their rocket just took. This data will be stored for further analysis in the classroom at a later time. The briefcase will be connected to both launch pads so that the launches can happen in quick succession.

The entire system needs to be portable so that it can be moved around on carts that Space Trek has designated to be used for rocket launches. The launch pads cannot remain in the field because they are not resilient to outside exposure. They can spend several hours outside but the components will eventually begin to rust and decompose if not cared for. The team had several meetings with Space Trek to ensure that the final SWR system works well for them. The main goal of this system is to provide an exciting and educational launch experience for the students.

2.2 Objectives

The Space Trek camp runs year round but they have the most classes in the summer. The Smart Water Rocket system needs to be operational by summer 2016 at the latest so that they can use it in the upcoming classes. Ideally, the

SWR systems will start being implemented in early 2016 so that any bugs can be worked out in time for the summer. The objectives for this project are easily broken down by semester:

Fall Semester 2015

- Perform research into the topic to discover potential design constraints and determine how realistic each subsystem is to design
- Complete preliminary designs for each subsystem
- Complete the software roadmap
- Research potential parts and create a budget for Space Trek to approve
- Discuss areas where the system can be further modified
- Begin testing of preliminary designs for each subsystem

Spring Semester 2016

- Finalize designs for each subsystem
- Construct the software required for communication, processing, and display
- Modify existing structures in preparation of the new systems
- Construct existing systems, ensuring that the software and the hardware are cooperating when required
- Perform extensive testing of each subsystem and software component
- Add cosmetic details to each cart to make them more exciting and appealing

These objectives should be performed in the order listed for efficiency but steps can be rearranged as difficulties occur. Space Trek gave the team a maximum budget of $1000 but hope that we can stay under that. The SWR team needs to achieve the goals for this project using as little of the budget as possible while still maintaining the professional aesthetic that Space Trek is looking for. The detailed steps the SWR team is taking to ensure that the project is completed on time is included later in the paper.

2.3 Requirements Specifications

Since the Smart Water Rocket will be an integral part of the rocketry program Space Trek gave the team several specifications to follow for the project. The following specifications were chosen to ensure that the SWR is accurate and easy to use by the instructors and students.

- Preserve the original mass, flight profile, and shape of the rocket so that the pre-flight analysis program does not need modification
- Whenever possible reuse existing systems to reduce overall costs and remain under the $1000 budget
- Increase the efficiency of the launch process to increase the frequency of launches

- Create a system that is engaging and user friendly for students aged 10 and up
- Create a system that is repairable and replaceable to ensure its longevity
- Develop a launch sequence that includes several students from each group
- Maintain the portability of the launch system

These specifications are to be followed when developing the new system so that Space Trek is happy with the final product. The SWR team has broken the project up into several sections to be researched before beginning the design.

Existing technologies will be considered to help reduce the need for design. The team needs to determine which parts of the system already exist before time is wasted attempting to create a custom design. Many of the components of the SWR will need to be custom made by the team, reducing the total number of new designs will increase the chance of completing the entire system before the deadline. This research will delve into the technology currently in use at Space Trek. Many of their systems will be reused in the SWR and require extensive modification so knowledge of their design and operation will be essential to the success of the project.

The launches will involve children so it is imperative that all the proper standards are adhered to. The team is lucky in that Space Trek has been performing launches at Kennedy Space Center for several years and are very familiar with the rules and limitations imposed on water rocket launches. The SWR team must also adhere to communications and hardware standards to ensure that the system is legal and repairable. In addition to standards, the team must follow constraints set by Space Trek and the limitations of the available hardware.

Each member of the team took a lead role in designing a different part of the system. Each design required input from all members but it was easiest for one person to take charge of a section. Several of the sections proved too difficult and required significant collaboration between members. The overall design of the system has required a significant amount of work to remain within the specifications set by Space Trek. The SWR team has developed a test plan to be approved by Space Trek to construct and test each portion of the system and complete it within the allotted time frame.

2.4 System Summary

Based on the detailed objectives and specifications required for this project, the team devised a block diagram to describe how the different subsystems will interact with each other. The diagram below (**Diagram 2-1**) presents the interaction between the main subsystems, which will be expanded on in the following sections of this paper.

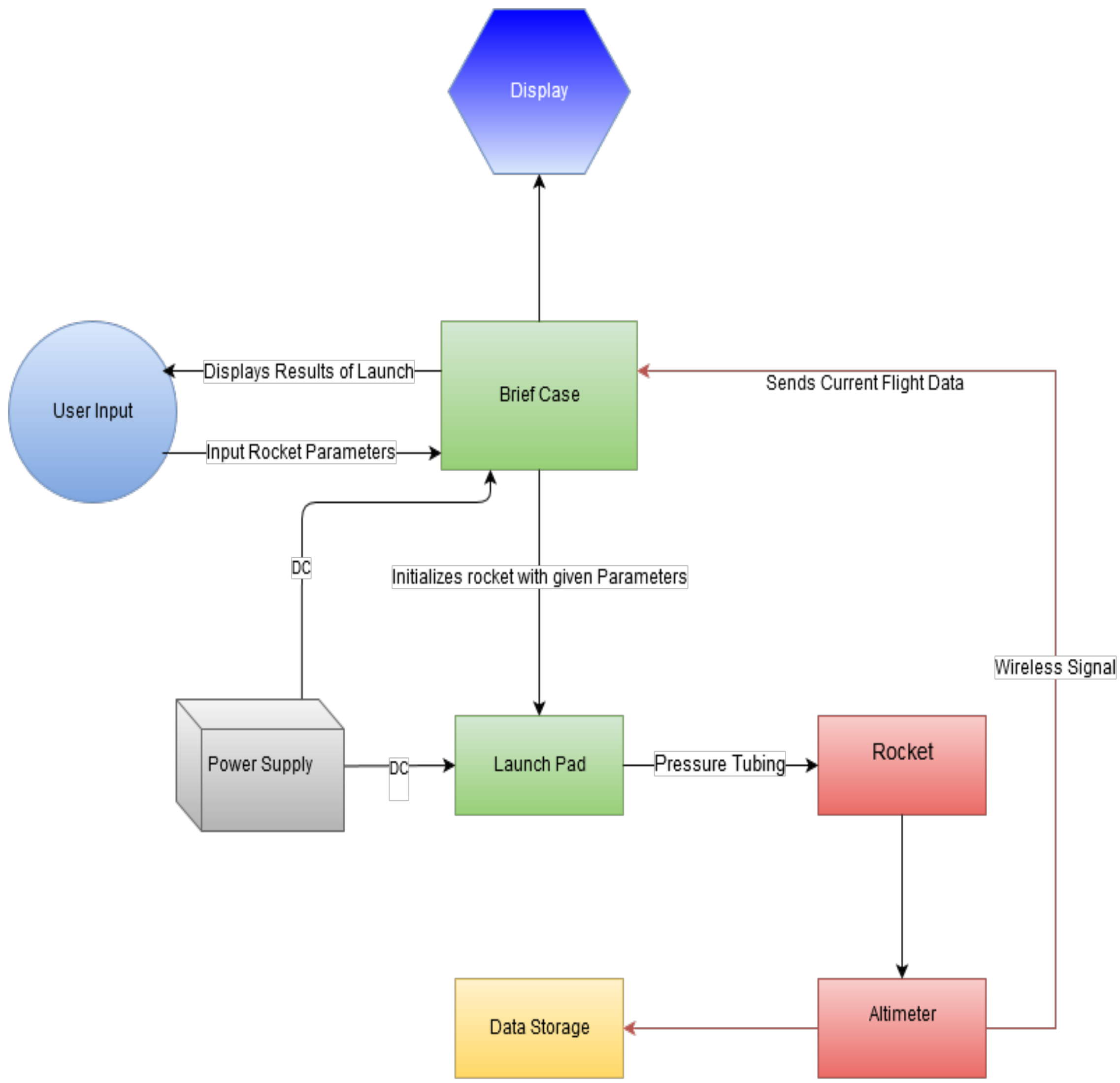


Diagram 2-1: Block diagram describing the interaction of all the subsystems.

Our first goal will be to obtain accurate readings without worrying to much about size and weight. By approaching our design of the altimeter this way, we will be able to work in conjunction with the other parts of our system and move our project further. Our current plan is to utilize accelerometers, gyros and some form of wireless transmission (radio, wifi or bluetooth) to build the onboard altimeter. We're currently planning on integrating a microcontroller to receive the sensor data, store it and relay it to our launchpad.

Analyzing the data provided from the rocket's flight is the most important aspect of our senior design project. Currently the process for acquiring the information from the altimeter is both slow and very challenging to keep organized. After each Launch the altimeter must be taken out of the nose cone and the team must record their data individually by scrolling through a single-line LCD display until they come across the data that they want to record. To deal with this problem we will create a program that can set up as many teams as needed and when each team launches their rocket, their data will automatically be stored and can be reviewed later.

To enhance the experience of launching rockets for both the students and faculty we will design the program to control all aspects of the launch as well as mid-flight data. Keeping everything in one concise place will save time and effort to make the experiencing more enjoyable so that there is less time setting up the rocket and more time launching.

The Software design for the rocket is extremely variable and can be achieved with most microprocessors using whichever language suits them the most. The flight data will be acquired from the sensors and then store the important intervals of information. The data will be sent to the launch pad where it will appear on the display. Our communication between the rocket and the launchpad is another very important aspect because this determines how reliable the data received can be. Once the data has been processed by the rocket and successfully sent to the launchpad, the students can see the stats of their rocket in real time. When the rocket has finally landed and come to a stop, the launchpad will be able to graph a trajectory of the flight and given students a graphical visual of what they just witnessed in real time.

Designing the touch screen display for the launch pad poses a few more issues and can be achieved in many ways such as the rocket’s microprocessor except for the limiting factors of cost and reliability. Currently we have to choose between designing our own LCD display or using a tablet. While the tablet will be more reliable is will be much more expensive, likewise a custom LCD display will be much cheaper but can lack in reliability. If the tablet fits in our budget, we can design our own application that can automatically prep and engage in the launch sequence and acquire all the data from the launch and analyze the results in real time. On the other hand if we use a custom LCD Display we can create a system of buttons to control the prep and launch of the rocket and store all the data to a removable device, which can be later uploaded to a computer and then analyzed.

Choosing which mode of communication to use is an important decision to the team as it will directly affect how reliable our design is. The first and most appealing option that we can use is Bluetooth, because of how affordable it will be to design and how straightforward data manipulation will be. By simply having a Bluetooth connection between launchpad and rocket we can send the data as soon as it is measured and processed.

The launch pad briefcase will be a completely new component to the water rocket program. Preferably we would like to make it simple and eye-catching so the kids partaking will be interested by something other than the rocket launch itself. The launch sequence is one of the features that will be interesting to watch on the display and phones. One idea we had was to start the launch process you must first insert a key to complete the circuit, (or one for each rocket). Another option would be a unique launch code to every rocket that can be input on the display. A physical button to initiate the launch will be more fun for the kids rather than a digital button. Once the countdown is initiated large numbers accompanied by a relevant countdown tone can really enhance the experience.

An abort sequence has also been thought of. It could be a failsafe that may be unnecessary but could be fun for the kids whether it be a big red button on the program or a tangible big red button (ideal experience). Even with a display a physical big red button could be implemented.

3. Project Research

The Smart Water Rocket will require several different subsystems to work in complete harmony. The team did significant research into which sections can be purchased and work out of the box and which systems will require completely custom designs. The research was split between the team members to prevent anyone from taking the entire brunt of the research.

3.1 Existing Similar Technologies

This project requires the understanding of multiple technologies; most of them already available to consumers. In order to create the best possible device, the group has to research and study current technologies. By looking into what has been successful in the past, the project is able to move quicker past the research stage and into development.

3.1.1 Jolly Logic AltimeterThree (SpaceTrekAltimeter)

Currently one of the most respected and accurate altimeters for hobbyists, this is the device the group are looking to replace and improve upon. Currently there are three versions of the device: the AltimeterOne, AltimeterTwo, and AltimeterThree; each one improving over the previous one.

The AltimeterThree is the one that we have focused on even though the AltimeterTwo is the one being used at Space Trek. The AltimeterThree, unlike its previous iterations provides Bluetooth connectivity to start recording and download the information gathered by the flight. The device works by simply being placed inside your “flyer” and from there it records height and 3-axis acceleration.

Other features that make this device so popular is the LCD display on the device, the integration with social media to share flights and the ability to use your phone for “geo tagging”.

The Smart Water Rocket's altimeter will use the physical dimensions and constraints of the AltimeterTwo (see **Figure 3-1**) as design parameters in order to keep the already designed rocket's center of mass. As for connectivity, The Smart Water Rocket altimeter borrowed Bluetooth integration from the AltimeterThree.

Figure 3-1: A photo of the AltimeterTwo used at Space Trek. The display (currently showing the battery level) is the only way to see data output from the device. Photo taken by the SWR team.

3.1.2 Perfectflite Firefly

The Perfectflite Firefly (see **Figure 3-2**) is a more do it yourself (DIY) and less consumer friendly solution, that makes up for its lack of aesthetics with an extremely power efficient and cheap design. The Firefly is a lightweight altimeter that records the peak altitude and maximum speed. To retrieve this information, a data port on the device is used to communicate with a computer. As an added step for "security" the device features a high resistance to false triggering from sunlight or high wind gusts.

This altimeter will be used as a way to direct our spending costs and power requirements. Unlike our altimeter's future design, the Perfectflite Firefly doesn't support Bluetooth capabilities and as such our power draw will definitely be higher, but the cost and efficiency should be on par with it. Unlike the AltimeterTwo and AltimeterThree from Jolly Logic, this device lacks a screen, and more importantly Bluetooth integration, which the smart water rocket's altimeter will use to transfer data.

Figure 3-2: The PerfectFlite Firefly is an alternative altimeter for the SWR system.
Photo courtesy of PerfectFlite

3.1.3 Software in use at Space Trek

At the present moment, Space Trek's water rocket launches utilize minimal hardware to achieve their goal of collecting and analyzing launch data. The altimeter that is being used is the Jolly Logic AltimeterTwo, which must be reset before each launch to capture the data. When the rocket has landed then the instructor will retrieve the altimeter and wait for the information to scroll by on a single LCD display. The team will improve not only the way data is gathered but also will create custom software to cater specifically to Space Trek's needs. Currently there is no software being used that is specifically for the launches.

3.1.4 Space Trek Rocket Nose Cones

Space Trek has four iterations of nose cone designs. The nose cone of the rocket can completely reshape the flight profile and must be chosen carefully. Space Trek has stated that the SWR team is not responsible for designing a new nosecone because of the complications involved. The altimeter that Space Trek uses is currently housed in the nose cone of the rocket so that it can be removed and replaced easily.

The first iteration of the Space Trek nose cone resembled Styrofoam. It proved to be too heavy and made the rocket tip. There were not many versions of this iteration made because of its overall poor performance. The next iteration was significantly lighter and improved the flight path of the rocket. Unfortunately it is also very brittle and if the parachute malfunctions is prone to breaking. Since the altimeter is housed in the nose cone it is imperative that it remains whole during flight. The third iteration is a miss-match of the fourth iteration. Space Trek was trying a new method of making nose cones and many of them turned out rubbery and misshaped. These nose cones were similar to the first, too heavy and incorrectly shaped to be of any use. The final iteration is the lightest and most rugged. It is still prone to shattering if the parachute mechanism fails but the nose cones shatter much less often. These cones help the rocket make a perfect arc as it flies. The SWR team will attempt to integrate the altimeter into the nose cones already in use at Space Trek (see **Figure 3-3**). If possible, new nose cone designs will be researched to increase the performance of the rocket and further protect the altimeter from damage.

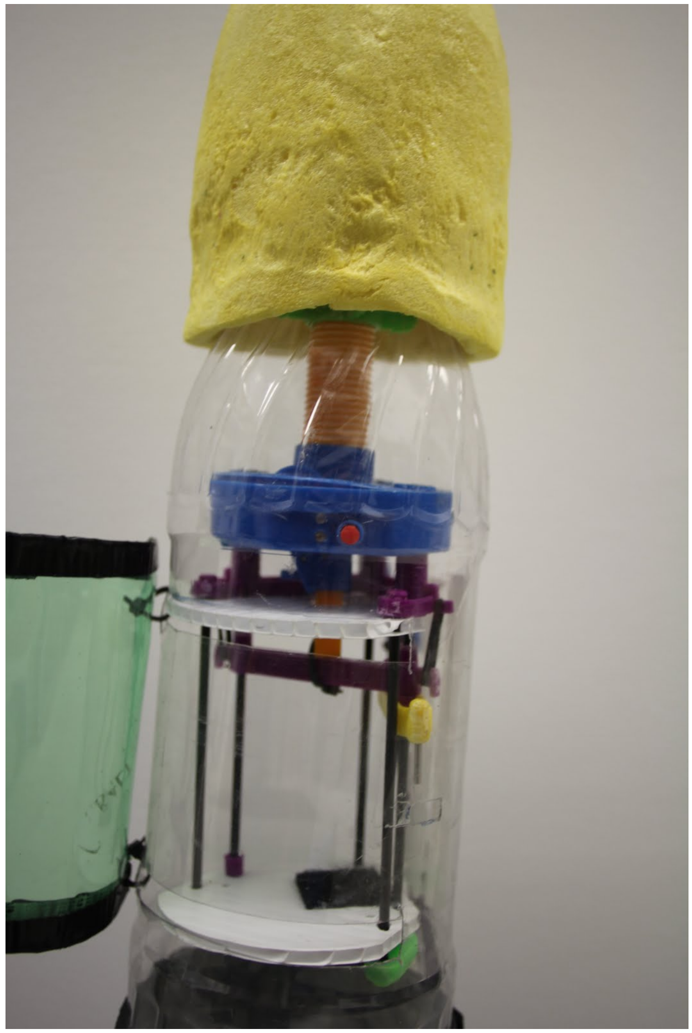


Figure 3-3: Iteration 4 of the nose cone screwed into place on a rocket. The parachute release mechanism is the blue puck attached to the orange threaded rod. The altimeter rests inside the nose cone and is held in place by the rocket itself. Picture taken by the SWR team.

3.2 Strategic Components

The components the team researched for the Smart Water Rocket were often quite similar. It took several discussions to reduce the total number of choices to a reasonable amount. The final components the SWR uses will have to be approved by Space Trek so the SWR team reduced the more difficult choices down to only a few components. Research into these topics will ensure that the team is prepared to use whichever component Space Trek determines is best for the project.

3.2.1 Rocket Processing

In order to create a ‘smart’ rocket for Space Trek, there needs to be a microcontroller unit (MCU) within the rocket to collect information about the flight in real-time. When deciding which microcontroller to use for the rocket the same conditions apply as did with the decisions for the mobile device platform. The ideal microcontroller should have a well-established language and an easy to use development environment. While the drawback is that most environments for microprocessor developments aren’t as easy to use when compared to higher-level environments, the benefit is the efficiency of assembly language.

The microprocessor in the rocket should work in conjunction with all of the sensors that comprise the altimeter to transmit the collected data from the launch and send it to the mobile device on the launch pad via the communication medium of choice. While it is important to have excellent performance when reading such important data, the most important factor is the size and weight of the microcontroller. Granted most microcontrollers are intended to be very small and compact, the nose cone of the rocket in which the printed circuit board will be located is very small which limits our team to small microcontroller units.

Choosing a microcontroller for the rocket is essential to what must be accomplished in order to call the project a success. The goal is to design an altimeter, which can manipulate the data and transmit it to another display. Without a microcontroller there is no way to accomplish this task. When comparing other small companies to Texas Instruments, Samsung, Qualcomm, Intel and other large producers of semiconductor technologies, it doesn’t seem worthwhile to invest in a product that could otherwise be unreliable or obsolete. Nanode, Pinguino, STM32, Teensy and other smaller companies will more than likely not be considered as the microcontroller of choice within the rocket itself. While these companies produce outstanding microcontroller technologies, the reliability factor is very important to the design team as well as the sponsors at Space Trek.

The rocket is going to ideally be as low maintenance as possible with cheap replacements, and when thinking about using smaller, more specialized products, these conditions become increasingly harder to be met. The choice of MCU will come down to which one of Texas Instrument or Arduino’s products will produce the desired size to performance ratio, which has yet to be determined.

3.2.1.1 TI MSP430

Texas Instruments is the third largest producer of semiconductors, with an extensive family of microprocessors that can be made to fit a vast number of specialized tasks. The MSP family of microprocessors by Texas Instruments is a great option because of many years of experience and the amount of processing power compared to its relatively small size. Using the MSP430 for many engineering courses has allowed the team a head start on assembly coding by being familiar with the language as well as the development environment, Code Composer Studio. While not as easy to use and debug when compared to higher level development environments such as eclipse or visual studies, it is a very powerful and easy to understand tool. This gives using a Texas Instruments Microcontroller Unit a strong choice for the current experience, as well as the reliability of a well-known and respected company.

Texas Instruments' MSP430 benefits from very small voltage requirements, low price and low power consumption. To make power saving more efficient, the board also features five different power saving modes as well as standby mode. **Table 3-1** below represents the key points that make the MSP430 a valuable solution.

Parameter	**Minimum**	**Typical**	**Maximum**
Supply Voltage	1.8V	2.5V	3.6V
RAM		2kB	
Communications		SPI, I2C and UART	
Flash memory		16kB	

Table 3-1: All values taken from the MSP430 datasheet.

3.2.1.2 Arduino

Arduino is an open-source computer hardware and software company that creates microcontroller-based kits for personal and commercial use. In contrast to Texas Instruments, who create many other forms of semiconductors other than microprocessors, Arduino is based solely around their microprocessors. This gives Arduino the advantage in being an innovator in the industry because of their undivided focus when compared to other bigger companies that have many more technologies to pay attention to.

Using a microcontroller unit from Arduino would allow the advantage of modularity and flexibility, as well as having better access to advice from the online forums. The altimeter could benefit from an Arduino MCU that would allow for faster processing and more potential when connecting external sensors to the circuit board. Given the nature of the rocket design, using the microcontroller from Arduino could allow for better gathering and transmission of data which is essential to the success of this project.

Currently the most common offering from Arduino is the Arduino Uno Rev3 board, which incorporates the ATMega328p MCU into its board. Utilizing this board and consequently the ATMega328p on the altimeter offers certain benefits when compared to the MSP430. From **Table 3-2** below, we can observe how it offers more flash storage and requires a lower operating voltage.

Parameter	Minimum	Typical	Maximum
Supply Voltage	1.5V	2.5V	5.5V
RAM		2kB	
Communications		SPI, I2C and UART	
Flash memory		30kB	

Table 3-2: All values taken from the ATMega328p datasheet.

3.2.1.3 Raspberry Pi

Raspberry Pi is a series of single-board computers designed with the idea to promote learning and innovation in computer science. An excellent benefit to Raspberry Pi is that they provide credit card sized boards with tremendous performance that can easily handle all the data gathered during a given flight. The only downside is that as small as a credit card is, it will be far too big to fit well within the nose cone of the rocket. This is why the Raspberry Pi won't be used in the rocket but given its great performance there could still be a use for it within the Launchpad itself and can even run Android for the development of an application to use to gather the launch data.

3.2.2 Sensors

Like most other systems, especially those which require interaction with external cues, the SWR requires the use of sensors to properly determine the rocket's state. For the purpose of the SWR system an altimeter will be required to determine the rockets max altitude. Also, the use of gyroscopes and accelerometers could be implemented in conjunction with the altimeter to provide an accurate flight path.

The sensors should easily interact with an MCU within the rocket, while also keep cost, weight and power consumption to a minimum. The main power requirement will be its ability to work in a system with or around a 3.3V input. This will give us the option to easily integrate the system to USB standards.

3.2.2.1 Pressure Sensors

The altimeter will be a pressure-based altimeter; this means it will use a pressure sensor to relate the pressure to the altitude. This will be achieved by integrating the pressure sensor to an MCU and using the following equation:

$$altitude = 44330 * \left(1 - \left(\frac{p}{p_0}\right)^{\frac{1}{5.255}}\right)$$

.

Equation 3-1: Pressure to altitude relationship equation.

To obtain the best possible sensor, the group had to investigate a variety of pressure sensors that satisfied our low cost and power need, while keeping weight into consideration. When it came to the final decisions it came down to three highly popular sensors: two solutions from Bosch, the BMP180 and BMP280, and lastly Freescale's MPL3115A2. All of three of these sensors are available as breakout boards from different distributors like SparkFun or Adafruit, which provides a definite benefit during our prototyping phase.

The three main design aspects used to narrow our alternatives were package size and weight, I2C compliant and power efficiency. Resolution and sensitivity were taken into consideration, but were only considered as secondary priorities next to compatibility.

3.2.2.1.1 BMP180

The ultra-low power, low voltage electronics of the BMP180 are optimized for use in mobile phones, PDAs, GPS navigation devices and outdoor equipment. This makes it one of the top choices for use in an altimeter The I2C interface allows for easy system integration with a microcontroller which is one of the key factors looked at when looking at this pressure sensor.

Parameter	**Test Conditions**	**Minimum**	**Typical**	**Maximum**
Supply Voltage (VDD)		1.8V	2.5V	3.6V
Relative Accuracy (VDD=3.3V)	950-1050 hPa @25□		±0.12 hPa	
Absolute Accuracy (VDD=3.3V)	300-1100 hPa @25□	-6.0 hPa	-1.0 hPa	+2.0 hPa
Long Term Stability	12 Months		±1.0 hPa	

Table 3-3: All values taken from Bosch's BMP180 datasheet

From the datasheet values above (see **Table 3-3**), this sensor is capable and has been tested for a 3.3V input while also providing a high-resolution output. By adding some form of reset or “normalize” button we should be able to resolve any stability issues. In the case of this data sheet the information starts to vary significantly around the 12 months mark.

3.2.2.1.2 MPL3115A2

From the table below (see **Table 3-4**) we can see that Freescale’s solution provides a slightly larger and heavier device, but includes more user programmable options like power saving modes and interrupts. The I2C is also present, allowing for easy system integration with a microcontroller.

Parameter	**Test Conditions**	**Minimum**	**Typical**	**Maximum**
Supply Voltage (VDD)		1.95V	2.5V	3.6V
Relative Accuracy	70-110 kPa @25□		±0.05 hPa	
Altitude Resolution		.0625 kPa	0.3 kPa	
Long Term Stability	12 Months		±0.1 kPa	

Table 3-4: All values taken from Freescale’s MPL3115A2 datasheet.

Just like Bosch’s BMP180 the MPL3115A2 works with a 3.3V input while providing more accurate and higher resolution readings. Also, the MPL3115A2 provides a much greater long term stability under ideal conditions.

3.2.2.1.3 BMP280

Just like the BMP180, the BMP280 has been developed by Bosch for small mobile applications. When compared the two options benefit from extremely small packages as well as full I2C compatibility. Unlike the BMP180 the BMP280 also features an SPI interface, which could provide useful in the case of an MCU revision.

Parameter	BMP180	BMP280
Footprint	3.6mm x 3.8mm	2.0mm x 2.5mm
Vdd(min)	1.8V	1.7V
Current Consumption	12uA	2.7uA
Pressure Resolution	1 Pa	0.16 Pa
Interfaces	I2C	SPI and I2C
Measurement Rate	120 Hz	157 Hz

Table 3-5: Taken from the BMP280's datasheet.

Based on the comparison above (see **Table 3-5**), the BMP280 seem to be the logical option between the two. This pressure sensor features and even smaller size, although 0.3mm thicker, as well as extremely low power consumption. To further set the two apart, the BMP280 is also capable of working at a minimum voltage of 1.7V, which is well below the planned 3.3V voltage supply that will be going into the pressure sensor.

3.2.2.2 Accelerometer

Unlike the average altimeter, our final design will be able to provide not only the water rocket's altitude but also its velocity. To attain these values an accelerometer will be integrated with the pressure sensor. The final design should provide, in theory, accurate reading for altitude, velocity and acceleration. The latter two will be obtained solely from the accelerometer

For our accelerometer two different types were considered: dual axis or triple axis. The idea of using a single-axis accelerometer was dropped due to the limited capabilities of such a device. By utilizing a multi-axis accelerometer we're able to obtain more data with about the same board real-estate. Also, utilizing a dual or triple axis accelerometer provides us with the ability to obtain the water rocket's position in a two dimensional or three dimensional space.

Most projects will usually suffice with a dual-axis accelerometer (see **Figure 3-4**). This type of accelerometer provides accurate readings along an x-y plane, breaking down the values into two vector components which can be used for two dimensional positioning.

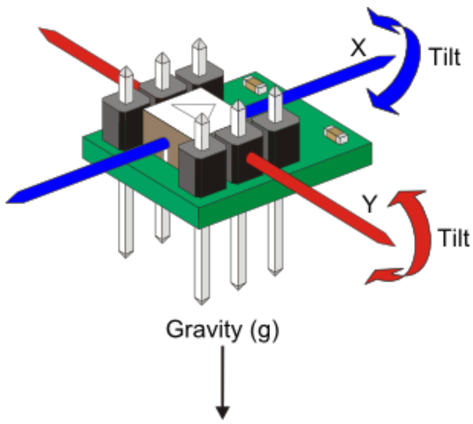


Figure 3-4: Dual-Axis Accelerometer
Permission requested Parallex

In the case of a triple-axis accelerometer, the device will provide accurate readings along an x-y-z plane (see **Figure 3-5**). This accelerometer will breakdown the acquired values into three vector components which can be used for three dimensional positioning.

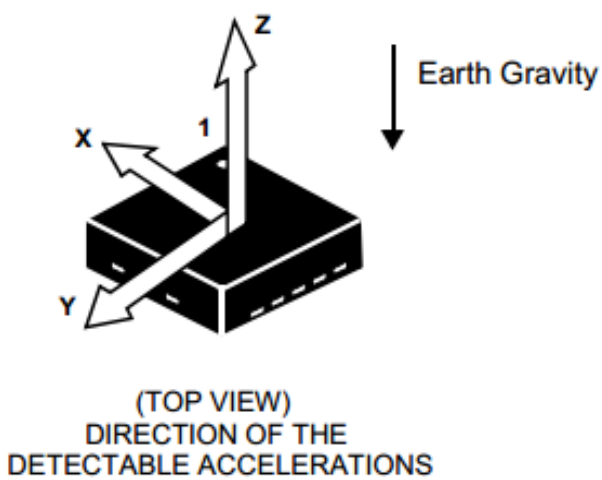


Figure 3-5: Triple-Axis Accelerometer illustration.
Taken from the MMA8451 datasheet.

As is the case with the Smart Water Rocket System, 3D positioning would be ideal. Although 2D positioning would suffice, at the current state, pricing between single and multi-axis accelerometers is basically the same which makes it much more logical to use a triple axis accelerometer and “cut down” its capabilities if 3D positioning will not properly benefit our final system during testing.

Below are the three main accelerometers considered. Just like with our pressure sensors the key factors are low voltage and power, high sensitivity, size and I2C compatible.

3.2.2.2.1 MMA8451

The MMA8451 from Freescale is a perfect candidate for our accelerometer. This incredibly small triple-axis accelerometer works with I2C and is available from Adafruit as a breakout board at a low cost. Even more to our benefit, the accelerometer was built with mobile applications in mind, which makes it a perfect option for use in low power applications.

Freescale's MMA8451 features an automatic ODR (Octal Data Rate: Eight bits per clock cycle per lane) change for Auto-Wake and return to sleep. This is an incredibly useful tool for maintaining power consumption down and possibly "zeroing-out" the rocket at its starting position, identifying the launch and subsequently the landing.

Parameter	**Test Conditions**	**Minimum**	**Typical**	**Maximum**
Supply Voltage		1.95V	2.5V	3.6V
Sensitivity Accuracy			±2.64%	
Operating Temperature		-40°C		+85°C
Sensitivity vs. Temperature	25°C		±0.008 %/°C	
Low Power Mode	ODR=50Hz		14 uA	
Normal Mode	ODR=50Hz		24uA	

Table 3-6: All values taken from the MMA8451 datasheet.

The data above (see **Table 3-6**) is useful at showing how little temperature will affect this accelerometer. With only a 0.008% change per degree Celsius from a 25 starting position, this accelerometer should be able to handle Florida's temperatures with ease.

3.2.2.2.2 LIS3DH

The LIS3DH triple-axis accelerometer from STMicroelectronics is both I2C and SPI compliant, just like our other accelerometers. This device actually belongs to the "nano" family based on its size which makes it a prime candidate for our altimeter.

Parameter	Test Conditions	Minimum	Typical	Maximum
Supply Voltage		1.7V	2.5V	3.6V
Sensitivity Accuracy			±2.64%	
Operating Temperature		-40°C		+85°C
Sensitivity vs. Temperature	25°C		±1 digit/°C	
Low Power Mode	ODR=50Hz		6uA	
Normal Mode	ODR=50Hz		11uA	

Table 3-7: All values taken from the MMA8451 datasheet.

The main characteristics that stand out on this accelerometer make it very tempting for our final design. This accelerometers power consumption is significantly lower than Freescale's MMA8451. At just 6uA in low power mode, this accelerometer uses less than half of what Freescale's solution uses. Not only is the power consumption impressive, the LIS3DH will also work under the same temperatures and can be powered with even less voltage than the MMA8451.

Unlike Freescale's accelerometer the LIS3DH features free fall detection, which could be useful to stop our altimeters function inside the rocket and effectively cut down on power consumption. This could potentially half our power consumption since our rockets time on free fall will almost always be higher than the time ascending, mainly due to the parachute.

3.2.2.2.3 ADXL345

The ADXL345 from Analog Devices is the third and last candidate for the altimeter design. This triple-axis accelerometer features a high resolution (13-bit) measurement as well as an I2C interface. It's intended for small mobile applications and features low power modes. This device measure just 3mm x 5mm x 1mm, which is about the same as the MMA8451, but bigger than the LIS3DH.

Parameter	Test Conditions	Minimum	Typical	Maximum
Supply Voltage		2V	2.5V	3.6V
Sensitivity @ 10-bit resolution	±4g	116 LSB/g	128 LSB/g	143 LSB/g
Operating Temperature		-40□		+85□
Weight			20 mg	
Low Power Mode	ODR=50Hz		6uA	
Normal Mode	ODR=50Hz		11uA	

Table 3-8: All values taken from the ADXL345 datasheet.

Just like the LIS3DH the ADXL345 feature automatic free fall detection as well as a standby mode. Although its features are very similar and uses more power, this accelerometer has one feature that makes it very unique. Analog Devices use a patent pending FIFO technology that minimizes the host processor load, which could potentially help our MCU.

3.2.3 Batteries

The SWR system will need two different batteries to operate. The automated systems and briefcase environment, the main system, must be battery powered so they remain portable and the altimeter must be battery powered because it will remain inside the rocket for the duration of the flight. Both batteries need to be rechargeable to keep the system reusable but otherwise the batteries have different design constraints. The altimeter battery must be very small and lightweight so that it does not affect the center of mass of the water rocket. It also needs to be either long lasting or easy to change because the altimeter will be used for multiple flights per session. The main system battery can be larger because it will be stored on a cart but it must still be small enough to fit in the cart and light enough to not significantly impede the cart's motion. Both batteries will be used often so they will require a long lifespan or they must be cheap to replace.

Due to financial constraints the batteries in the SWR may not be the most efficient for effective for the application. Two of the goals of this project are to make it ecologically stable and reproducible so that Space Trek and easily perform maintenance and upgrade the system as they need to. Ideally, the best battery would be used in both the main system and altimeter but as long as the design functionality is unimpeded compromises can be made to ensure that the final design is both energy efficient and cost effective.

3.2.3.1 Lead-acid Batteries

Many vehicles utilize lead-acid batteries because they are comparatively long lasting and can store large amounts of power for long periods of time with minimal decay (see **Figure 3-6**). Unfortunately if the battery is not fully charged after use it will start to degrade through sulfation. Even if the battery is maintained properly it will still suffer sulfation after repeated and extended use. Lead-acid batteries are charged slowly because high voltages can cause the battery to corrode but care must be taken because if the battery is charged too slowly there is a risk of sulfation due to a lack of charge.

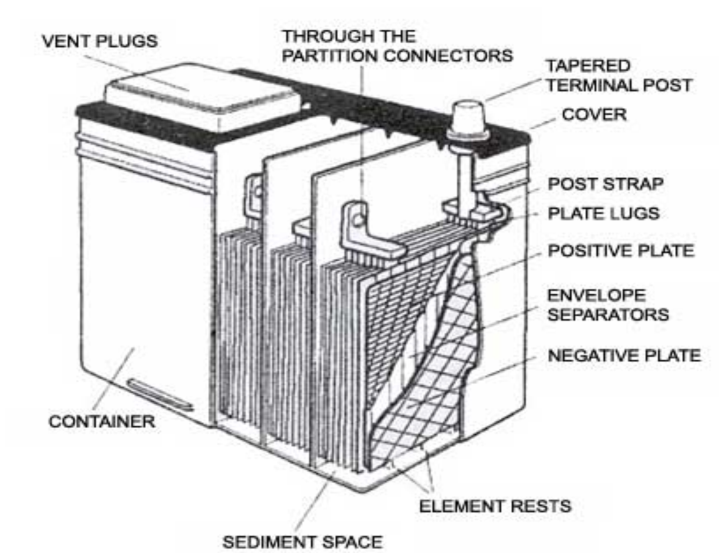


Figure 3-6: Breakdown of Lead-Acid battery

Lead is a toxic metal and consumption of it will lead to various health problems. Fortunately, the lead in the battery will be inaccessible and will not be an immediate issue. The battery will need to be properly disposed of at the end of its life cycle and proper measures must be taken to ensure that the battery does not leak any dangerous material. Generally, lead acid batteries will last between one and four years but with proper maintenance can last up to ten years. Lead is a very heavy element, which means these batteries can be very heavy. There are two main distinct types of lead-acid batteries; flooded and absorbent glass mat (AGM). Flooded batteries require more maintenance and care because they require an electrolytic fluid. Water must be added to flooded batteries occasionally to maintain the level of fluid inside the battery. AGM batteries are sealed and do not require the same amount of maintenance. The main difference between the two types is that AGM batteries are less susceptible to sulfation and can be charged much faster. AGM batteries also cost more than their flooded counterparts but are overall more reliable and perform more efficiently.

A lead-acid battery will make a poor choice for the altimeter battery due to its size and weight but, may be ideal for the main system. The main system would benefit from the high power output of the battery and will not draw enough power from it to reduce its lifespan or cause any major immediate sulfation. The constraints for using this type of battery for the main system will be the recharge rate and reusability of the battery.

3.2.3.2 Nickel Batteries

Nickel batteries are low voltage, fast charging, and long lasting (see **Figure 3-7**). The five main variations of nickel batteries are: nickel-cadmium, nickel-metal-hydride, nickel-iron, nickel-zinc, and nickel-hydrogen. Nickel batteries are commonly used in consumer electronics because they are lightweight, charge quickly, and require little maintenance. Nickel-cadmium and nickel-metal-hydride are the only two variations that would make sense for the SWR system because the other variations are either too low voltage or too expensive for this application. Both variations suffer from the “memory effect”; they must be completely discharged periodically or they will “remember” the level they were most often discharged to as the new empty level, which results in reduced capacity.

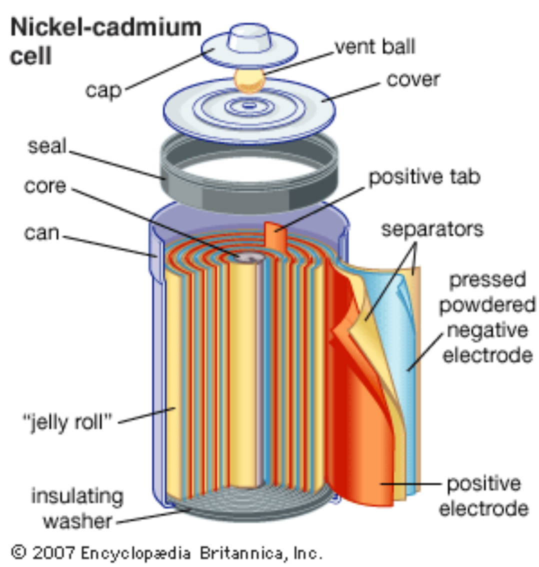


Figure 3-7: Breakdown of a Nickel-Cadmium

Fortunately, they can be stored completely discharged for extended periods with minimal consequences. The differences between nickel-cadmium and nickel-metal-hydride are minimal. Cadmium is a toxic metal so like the lead acid batteries special care must be taken when the batteries are disposed of. Nickel-cadmium batteries are all sealed to toxic leaks will not be a concern. Nickel-metal-hydride batteries do not contain any toxic materials so they inflict much less harm on the environment overall. They have a higher self-discharge rate than nickel-cadmium and must be recharged often but they are also more susceptible to damage from overcharging and undercharging so special care must be taken when designing the recharging system.

Nickel batteries are not realistically feasible for the main system power. They do not produce enough voltage to make an array of them viable. However, their low weight and high capacity make them completely viable for the altimeter battery. The constraints on using a nickel battery in the altimeter are weight and size. The cavity where the altimeter will be installed is very small and close to the nose of the rocket. Any significant addition of mass to the altimeter will change the flight trajectory and reduce the overall efficiency of the rocket.

3.2.3.3 Lithium Batteries

Lithium batteries are not rechargeable but both lithium-ion and lithium-polymer batteries are. Both variations are functionally similar but are packaged and produced in very different ways. Lithium-polymer batteries often come in a pouch or flexible container to reduce weight and rigidity. Less rigidity means the overall product can be more flexible but the battery is more vulnerable to damage. Aside from the shape and additional applications, lithium-polymer batteries are identical to lithium-ion batteries.

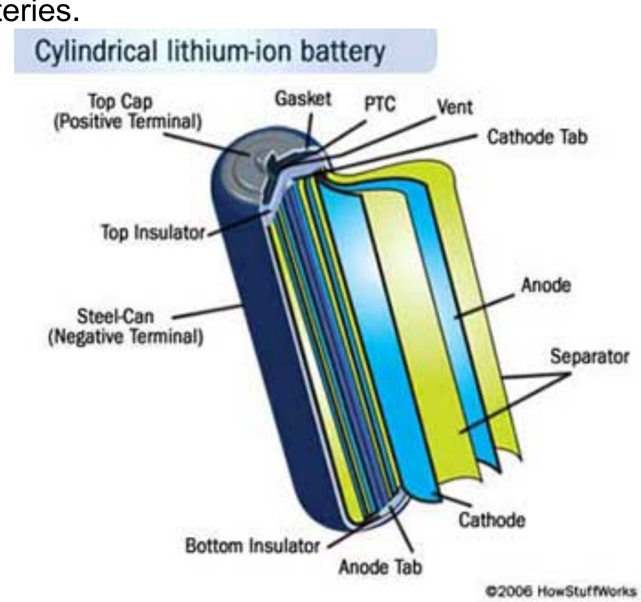


Figure 3-8: Breakdown of Lithium-Ion battery

Lithium-ion batteries are very similar to nickel batteries but offer higher performance (see **Figure 3-8**). Just like nickel batteries, lithium-ion batteries are lightweight, low maintenance, and can be stored in a fully discharged state. Historically they have been very costly but recently they have dropped in price substantially. Again, like nickel batteries there are several variations of lithium-ion batteries built for specific applications but they all have similar operating characteristics. In general, these batteries can have voltages twice as large as nickel batteries and have higher current capacities. They can be charged quickly and have no toxic components. The main downside to using lithium-ion batteries in the SWR system is that they are very sensitive to thermal changes. The SWR system will be outside for hours at a time so special care must be taken to keep the batteries cool.

An array of lithium-ion batteries could be powerful enough to run the main system but will be costly to produce compared to a lead-acid battery. The reduced weight and charge times of the lithium-ion array may outweigh the cost benefits of the lead-acid battery but this largely depends on the market. The lifecycle of the battery is also a factor. Since batteries are expensive it is imperative to ensure the battery lasts as long as possible. It may make more economic sense to purchase a more expensive lithium-ion array to ensure the longevity of the system. A comparison of the lifecycles of lead-acid and lithium-ion batteries is shown in **Figure 3-9**. If an array of lithium-ion batteries can be purchased at a reduced price then that will be the most cost and weight effective option for the main system, otherwise a lead-acid battery will prevail. For the altimeter battery the benefits of lithium-ion are immense. The increased capacity will significantly reduce the need to charge the battery and will allow for more flights between charges. The reduced weight and size will also be of great benefit. The altimeter will have to have a heat management design included to reduce the chance of thermal runaway in the battery.

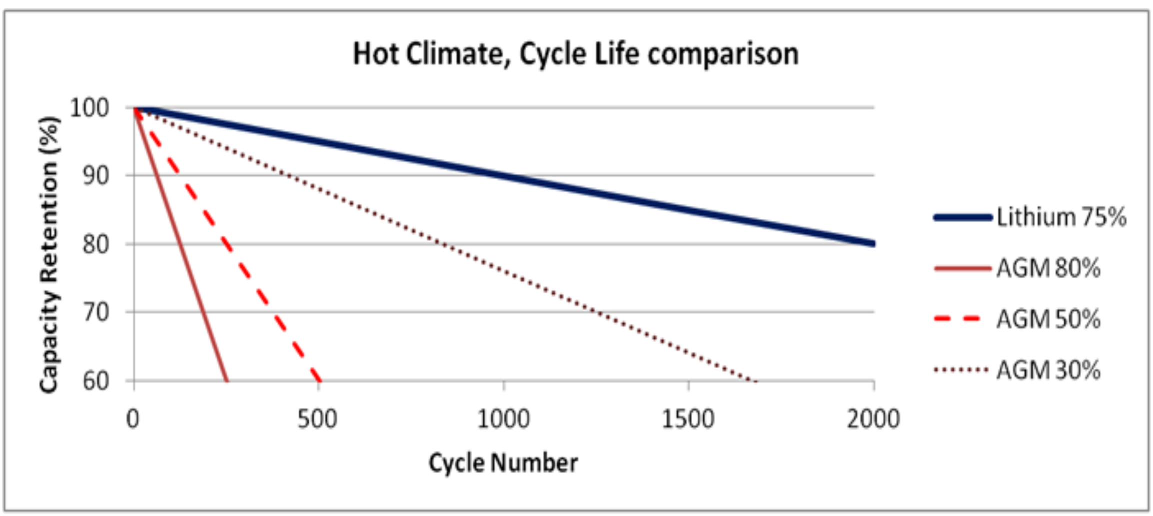


Figure 3-9: From Altenergymag, an online trade magazine specializing in topics related to energy, this diagram compares how many cycles a lithium-ion and lead-acid AGM batteries last. It is obvious that lithium batteries last significantly longer than even the highest quality AGM battery. (Permission requested 11/11/15)

3.2.3.4 Voltage Regulators

To properly power the small electronics in areas like the Smart Water Rocket's altimeter a specific voltage supply will be required. By providing a voltage source anywhere from 9V to 5V we risk damaging our electronics since the maximum operating voltage is well below those voltages.

With the Smart Water Rocket's altimeter a step-down voltage regulator will be required. Based on the other possible components, the ideal target voltage for each component will be 3.3V. Currently the main choice is the LM2576 by TI fixed at 3.3V. This voltage regulator will effectively provide a constant 3.3V output as well as 3A output current which is ideal for our components. Although it is not the cheapest of the voltage regulators analyzed, it is intended for use in small applications like ours and the 3A output will help make sure that all of our components have sufficient power going in.

Regulated Outputs (#)	1
Vin (Min) (V)	4
Vin (Max) (V)	40
Vout (Min) (V)	3.3
Vout (Max) (V)	37
Iout (Max) (A)	3

Table 3-9: All values taken from TI's LM2576 datasheet

From the datasheet (see **Table 3-9**) we can observe that this voltage regulator only provides a single regulated output, which means we will need to implement one before each component's VCC. This will only be the case if the provided source is above the max operating voltage for each separate component. In the case of an Arduino Uno Rev3 board for example, a step-down voltage regulator is not required.

3.2.4 Launchpad Automation

The three functions of the launch process that must be automated are: setting the vertical launch angle, loading the fuel (water), and pressurizing the vessel. The launch angle will be controlled by a microprocessor and controlled using a motor. This will be limited by the precision of the motor, which is limited by the power provided by the battery. The pressurization and fueling have similar constraints. Water will be pumped into the vessel using an air compressor of reasonable strength and the same air compressor will be used to pressurize the vessel. This will be accomplished using a series of tubes and check valves to prevent water from flowing back into the compressor and allow for both air and water to flow both simultaneously and independently. This system will require a microcontroller, motor, air compressor, tubing, and valves.

3.2.4.1 Launchpad Microcontroller

The motor will need to be controlled by a microcontroller. The team has experience with both TI and Arduino microcontrollers so both will be considered. Ideally the microcontroller should consume as little power as possible. This can be accomplished by limiting the functions of the microcontroller, which will also reduce the cost. The microcontroller should be near the motor to reduce the overall complexity of the design.

3.2.4.1.1 Arduino

There are two low power Arduino based boards available that will fit the basic requirements of this system. Usually, bi-polar stepper motors require five pins to run. The cart has two launch pads on it and both require a motor so ideally the microcontroller will control both, independently. Both the Arduino Nano and Micro have enough digital output pins to control the motors independently. The relevant specifications of each are listed in **Table 3-10**.

The Nano is slightly smaller, cheaper, and performs all of the required functions. The only advantage of the Micro is the additional pins, which could be useful in upgrading the system in the future. Based on the cost the ideal choice for an Arduino microcontroller for this system is the Arduino Nano however, Space Trek already has several other Arduino boards that they would like to recycle for this project. The board choice will ultimately be up to the sponsor.

	Arduino Micro	**Arduino Nano**
Operating Voltage	5 volts	5 volts
Input Voltage	6-20 volts	6-20 volts
Digital I/O Pins	14	20
DC Current per I/O Pin	40 mA	20 mA
Average Cost	$15	$25

Table 3-10: Specification comparison of Arduino based microcontrollers.

3.2.4.1.2 TI

Another family of microcontroller that could get the job done is the MSP430 Launchpad from TI. The launchpad family has a low-power operation mode that is suitable for the SWR's stepper motor implementation. From the MSP430Fxx series there is slightly more pins and power available (see **Table 3-11**) but this is not necessary. With the MSP430G2xx provided the same functionality needed for operating two stepper motors it would be the more reasonable choice because it is a bit cheaper. Although the MSP430 can work for the launchpad automation, it would require more work than implementing and Arduino as our microcontroller. Arduino simply has more resources online for the team to utilize when designing the automated launchpad.

	MSP430G2xx Series	**MSP430Fxx Series**
Operating Voltage	1.8-3.6 volts	5 volts
Input Voltage	3.6-15 volts	3.6-15 volts
Digital I/O Pins	40	47
DC Current per I/O Pin	23 mA	24 mA
Average Cost	$18	$32

Table 3-11: Comparing Launchpad G2xx and Fxx series

3.2.4.2 Motor

The launch angle automation motor must be controlled by a microcontroller to ensure that it is as precise as possible. The motor will attach to the launch pad using a belt or chain. There are many types of motors to choose from but this system will utilize either a servomotor or a stepper motor. These motors were chosen because they are simple to set up and sufficiently powerful for this application.

3.2.4.2.1 Servo Motors

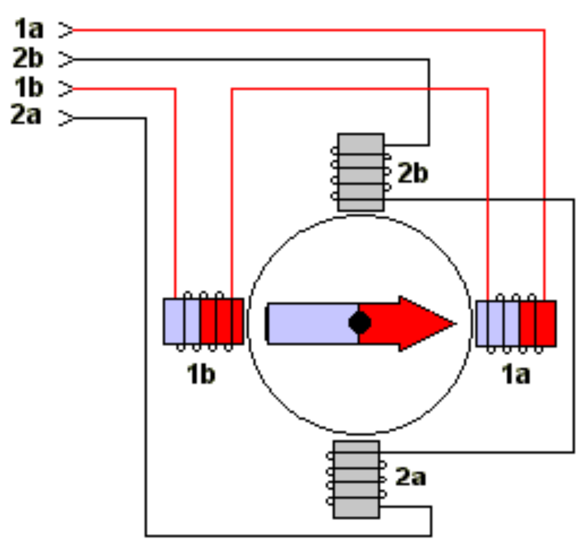
All members of the team have experience with servomotors from a prior course so those were the first motors researched. Servo motors rotate to precisely control velocity and acceleration. Utilizing a feedback loop, they retain the information on their position after moving. This helps compensate for rotating too far and other faults in the system. They are often used in applications where precise movement and torque are required. However, the extra precision comes with extra cost and the motors are more expensive than a stepper motor of similar power.

The launch pad motors will need enough torque to move the dry weight of the rocket into position and hold that position while the rocket is fueled, pressurized, and launched. Servo motors are best used in situations where a component needs to move a specific distance in a specific period, for example robots, CNC mills, and laser cutters. The SWR system does not need the precision offered by servo motors and the cost is too high to justify. Unless a servo motor is donated or found at a much lower price than usual, it will not be feasible to use in this system.

3.2.4.2.2 Stepper Motors

Stepper motors operate differently from servo motors in that servo motors use a feedback system to record its position and stepper motors do not. A servo motor can be told to rotate to a degree and a stepper motor must be given an amount of degrees to rotate. This is because the mechanism that rotates the shaft of the motor is different in each. Stepper motors use a series of gears and magnets to rotate to a set position and clamp down. The gear mechanism it uses gives the motor excellent holding stability and allows it to precisely move heavy loads. **Figure 3-10** shows how the shaft rotates using magnets between different poles to control movement in two directions, also known as bidirectional movement. However there are several disadvantages, the worst of which is the power consumption. Stepper motors consume power whether they are operating or not. The battery must be large enough to compensate for the continuous power drain from the microcontroller and motor.

Stepper motors are ideal for setting the launch angle because they hold their position well and still offer precise movements. Motors strong enough for this application are well within the budget constraints of the project and the lack of a feedback loop will simplify the programming component of the automation. The launch pad cart has room for two pads on it so at least two motors must be purchased.



Conceptual Model of Bipolar Stepper Motor

Figure 3-10: Permission requested on 11/11/15

3.2.4.3 Air Compressor

The SWR is currently pressurized by using a bike pump connected with thin vinyl tubing (see **Figure 3-11**). The team currently describes the bike pump as a physical aspect that shows the students of the program what work must be involved to pressurize the rockets before launch. Although there is a lesson in putting in work to see the resulting rocket launch, this is another opportunity for the water rocket to be made ‘smarter’.

Once the rocket is filled with the predetermined amount of water, it is ideally pressurized to around 90 psi. This level of pressure has produced the best overall results for the rocket's trajectory while remaining within NASA’s standards for safe operation. The bike pump achieves this, but the idea is to make the water rocket more automated. Another option for pressurization would be a compact air compressor.

When selecting a compressor there are three constraints: size, weight, and power consumption. Though most compressors can pressurize to 100 psi, it is a matter of acquiring one that can do so while still being compact and light. This is necessary because the water rocket program uses two carts about the size of a small shopping cart, which has limited space and cannot carry the size and weight of an average compressor. For our purposes though we will only be pressurizing a two liter bottle so the capabilities of larger compressors are not needed. Another issue is power consumption; though the compressor will need consistent power for operation the smaller size will drain too much energy from the rest of the SWR system. Ideally, the compressor chosen will operate within the power parameters but if it cannot there still needs to be a manual option for pressurization.

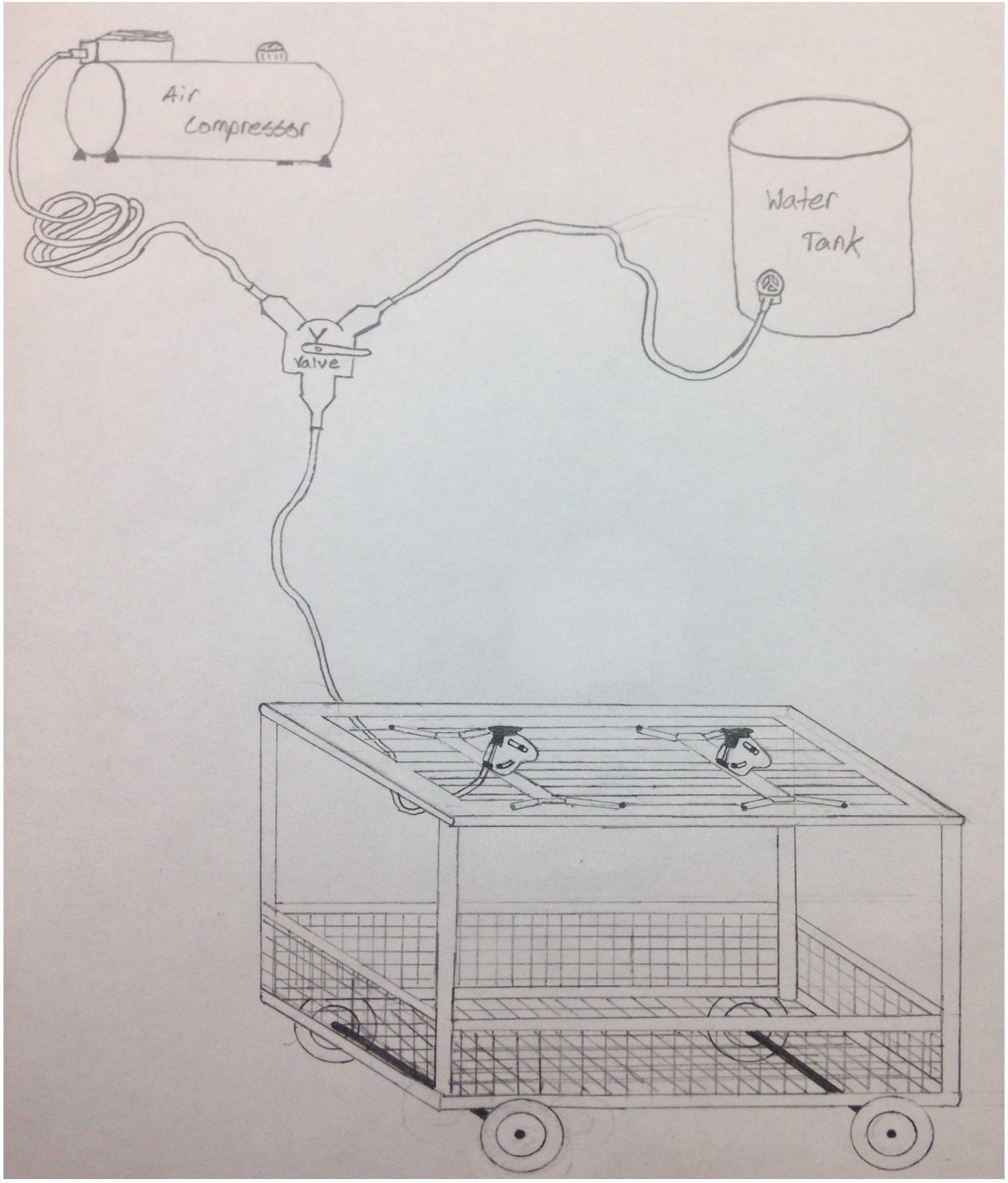


Figure 3-11: Sketch of the compressor system

3.2.4.4 Tubing and Valves

The pressurized air from the air compressor must be directed through a container of water to refuel the rockets. The tubing needs to be strong enough to withstand the pressure of the air compressor and flexible enough to be coiled and stored on the launch pad cart. The container of water needs to hold enough water to last the duration of all the flights of a session.

The water container can be a modified water cooler or similar large closed container. Any openings in the system will need to be closed to retain pressure and reduce stress on the compressor. The weight of the container must be minimized to compensate for the weight of the water stored within. The container and water must not be too heavy or it will impede the movement of the cart and reduce the portability of the system. Similarly, the tubing must be lightweight as well. Three easy to find tubing materials are vinyl, PVC, and rubber. PVC and rubber tubing are resilient and long lasting but are heavier and less rigid than vinyl tubing. Vinyl tubing will wear out over time and need to be replaced, but the increased rigidity will make it easier to coil and store. Vinyl tubing of sufficient diameter will be strong enough to contain the pressurized water and air and is the most suitable choice for the system.

The system will need water pushed into it, then it needs to be separately pressurized. Check valves will have to be used to prevent water from running into the compressor instead of the rocket and to allow air to go into the rocket independently of water. Check valves physically block the flow of a gas or liquid in one direction (see **Figure 3-12**). They are perfect for this application because they are cheap and resilient. Even though they will eventually become damaged or clogged; check valves are a low cost, easy to maintain solution for automating the refueling and pressurizing components of the SWR.

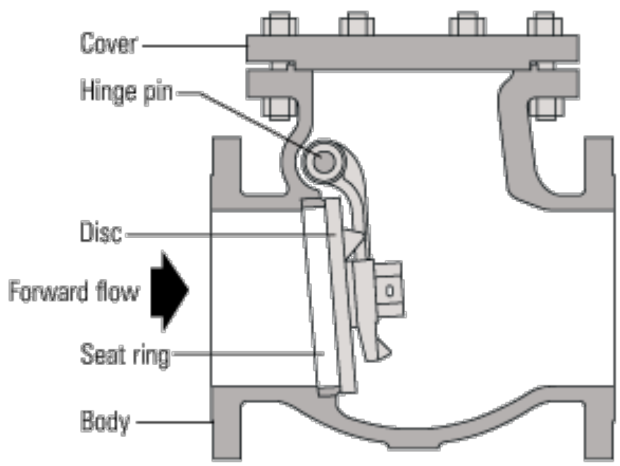


Figure 3-12: Picture courtesy of Flow Solutions International (permission requested 11/11/15). It shows the internal mechanism of a swing check valve. The forward flow represents pressurized air flowing into the system. The disk/hinge pin mechanism prevents air (or water) from flowing out of the system by closing, forcing the substance out through the other exit to the right of the picture.

3.2.6 Briefcase

The launch pad briefcase will be a completely new component to the water rocket program. Preferably the briefcase design will be simple and eye-catching so that the kids partaking in the Water Rocket Program will be interested by something other than the rocket launch itself.

Implementing a screen to display information about the launch is a necessity to the SWR's design. Acquiring a display at low cost is an option, but having a more desirable display at a moderate price would take up a larger portion of our budget. If a fancier display is the design focus then costs and features will be cut elsewhere as the features can be implemented as a program that can in turn be distributed to the kid's phones as an app; overall providing a more interactive experience.

3.2.6.1 Briefcase Controls

The addition of this briefcase will surely make the water rocket "smarter" because it includes a screen for displaying information about the different rockets, which will be controlled by an app, connected to several external buttons/controls that can be used as part of the launch sequence. With the addition of these buttons coupled with a screen to display information, it is our hope to enhance the overall experience of the children attending the space program, as well as make it more convenient for the people administering the launch.

This entire process can be one of two designs: there can be a single component for each ignition check, or two of each - one for each team. The idea is that maybe it will encourage the kids to have their own ignition keys, codes, and launch buttons. However, the launch does not fire off two rockets at the same time. So other than unlocking the briefcase to start the launch sequence, there isn't a point to have more than one keypad or launch button. With the rockets being fired individually, the designation of which team is entering their key code and launching their rocket can be accomplished by the app itself by switching between launch profiles. This will still allow us to have unique codes to each teams rocket profile while eliminating the unnecessary extra components.

3.2.6.1.1 Ignition Control

The launch sequence is one of the features that will be interesting to watch on the display and phones. One idea for the launch sequence was to start the process, you must first insert a key (or a unique key for each team) to complete the circuit and initiate the overall launch process. However, the use of keys as a separate piece of the system would leave the possibility of losing them in which case there would need to be a large set of the keys for the instance of this happening. Also providing the children with the keys creates a further strain on the Space Trek team by having to account for and recollect them - an inconvenience that would stray away from one of the major goals.

Even with the exclusion of this feature comes another more viable, feasible option, an ignition switch (see **Figure 3-13**). Using this as part of the SWR system would still allow another creative step to the launch sequence as well as eliminating the potential inconvenience of losing keys. This switch would act as an actual electrical switch. This switch would essentially complete the circuit and then let the user press the button. If the user were to try to push the button without first "flipping the switch" it would do nothing.

Figure 3-13: Example of a toggle switch with cover to be used in the launch briefcase. Permission granted from SparkFun

3.2.6.1.2 Keypad

Another feature for the system would be the use of keypad to enter team specific launch codes that will be coded to each rocket team's profile. Each rocket team can collaborate to create their own unique launch codes by using an app on their smartphones, or a collection of pre-loaded launch codes can be programmed into the system's app by default.

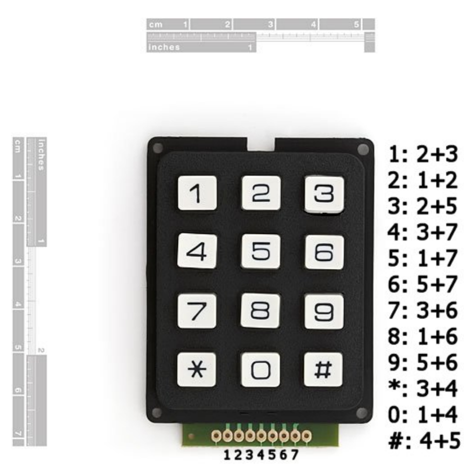


Figure 3-14: Example of the keypad to be used in the launch briefcase. Permission granted from SparkFun

To properly program the average twelve-button keypad, we would need to utilize seven pins as illustrated above in **Figure 3-14**. This could be an effective method of user input, but mandates the use of an MCU with around ten to twelve pins so we have space for more input devices. This method will be explored more in depth and see if using a method with five or six pins is possible to provide the ability to use an MCU with less pins.

3.2.6.1.3 Launch Trigger

Lastly, as the final component of the launch process will of course be a launch button to fire off the rocket. This method is currently being used by Space Trek and remains one of the most interactive things about the whole rocket launching experience (see **Figure 3-15**).

This can be better implemented as a digital button on the tablet or a series of buttons on the children's smartphones that must all be pressed to activate one final launch button. However this is unnecessary and redundant so a physical button to initiate the launch will be more fun for the kids rather than a digital button. The launch button can be a simple red push button that will launch the rocket after every other component of the launch process has been initiated. If the launch button were to light up it could really add excitement to the launch sequence, for instance the button could glow red until the preceding launch components are activated and once it reaches this point it will turn green to designate the rocket launch is ready.

Figure 3-15: Launch buttons currently in use at Space Trek. These could be reused or replaced with more aesthetically pleasing buttons. Photo taken by SWR team.

Once the countdown is initiated large numbers accompanied by a relevant countdown tone can really enhance the experience. While the app displays the countdown to launch an external piezoelectric speaker can sound off the predetermined launch tone, which will keenly enhance the experience of the sequence. Another external feature can be a synchronized seven-segment display that can countdown with the tablet (see **Figure 3-16**).

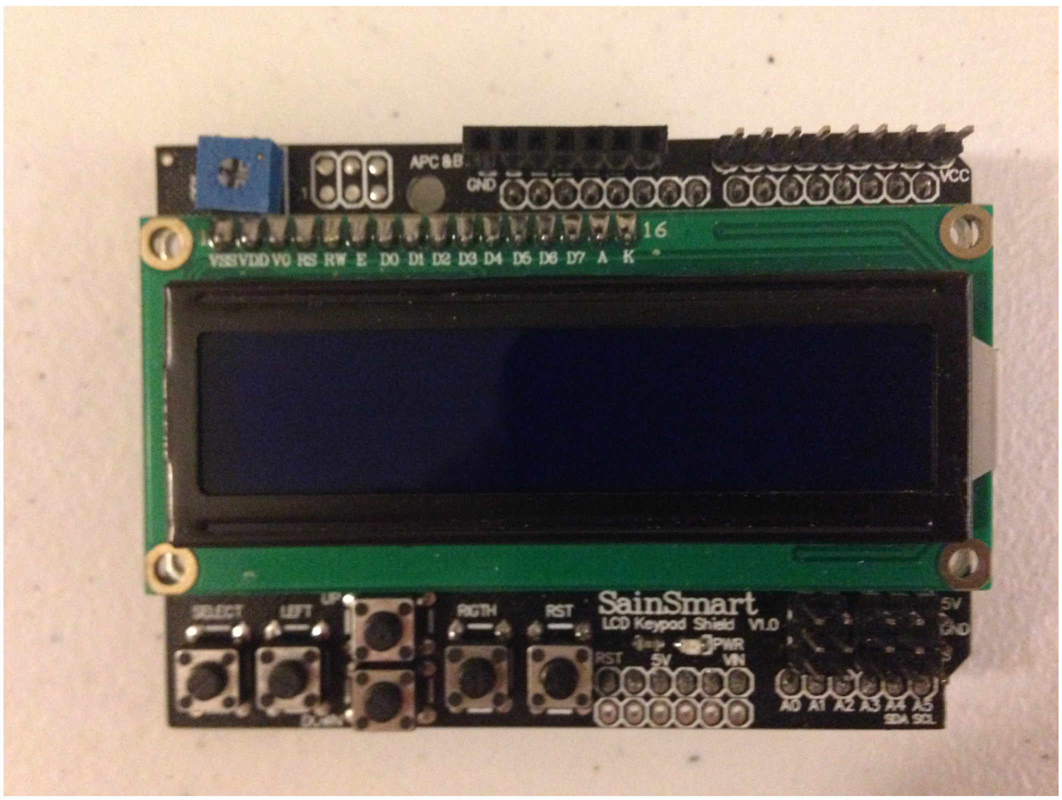


Figure 3-16: LCD Keypad Shield which can be attached to a microcontroller and adjusted using the onboard control buttons. Photo taken by SWR team.

An LCD seven segment display like the one above would require a few hardware components to be effectively integrated with a microcontroller unit. If this LCD display were to be connected directly into the microcontroller, the amount of pins required would be somewhere between seven and twelve depending on the situation. Obviously, this can be achieved, but keeping the amount of required pins down is a priority for our briefcase design.

To effectively implement an LCD like this one, the decision will be to either acquire an MCU with sufficient inputs and outputs or ideally, to use a CMOS BCD to seven segment latch/driver. One of TI's solutions like the CD4543B, would work.

3.2.6.1.4 Configurations

After selecting the above launch sequence controls, there the matter of how they will be laid out in the design. This thought process is mostly for aesthetics and the idea is to get the most natural layout for the briefcase design. **Figure 3-17** below illustrates the briefcase with a blank space demonstrating where the buttons will be located.

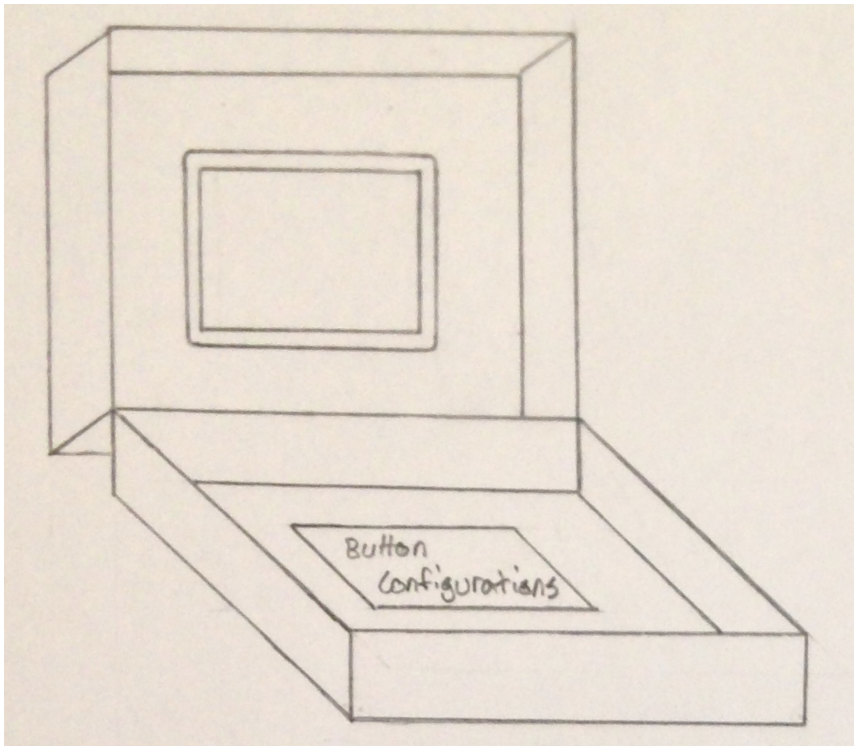


Figure 3-17: Sketch of the launch briefcase.

There are of course many different options for button configurations in the briefcase but the three will be a total of four different options considered. Though there are six possible orderings of the buttons there are two which have been discarded because it is not a natural layout for the launch sequence. These two layouts would have the ignition switch centered with the keypad pad and launch button being on either sides.

Another consideration for the button layouts is that the launch sequence will be designed to have a specific ordering in which the buttons are pressed. The most practical order in which the buttons will be pressed starts with the ignition switch being pressed, followed by entering a code on the keypad, and ending with the launch button of course to finally launch the smart water rocket.

<u>Briefcase Button Configurations (**Figure 3-18**):</u>

1. The first layout is configured to have the ignition switch at the right-hand side of the briefcase. This is possibly the most natural feeling as most people are right handed.
2. The second layout is configured to have the ignition switch at the left-hand side of the briefcase. With this setup the understanding is to have a natural left to right feel when going through the launch sequence.
3. The third layout is supposed to have the launch button centered. This option has is one of two versions of the red button being in the middle once the other two components have been initiated. With the ignition switch on the right it will feel the most natural and possibly another team member could be waiting on the left hand side to enter the key code after the ignition is activated.
4. The fourth layout is configured to have the launch button centered. The feel with this layout is to have the ignition switch and keypad on either side activated to prepare the launch button. Once ready the big red button will be right in the middle of everything for a very satisfying blastoff.

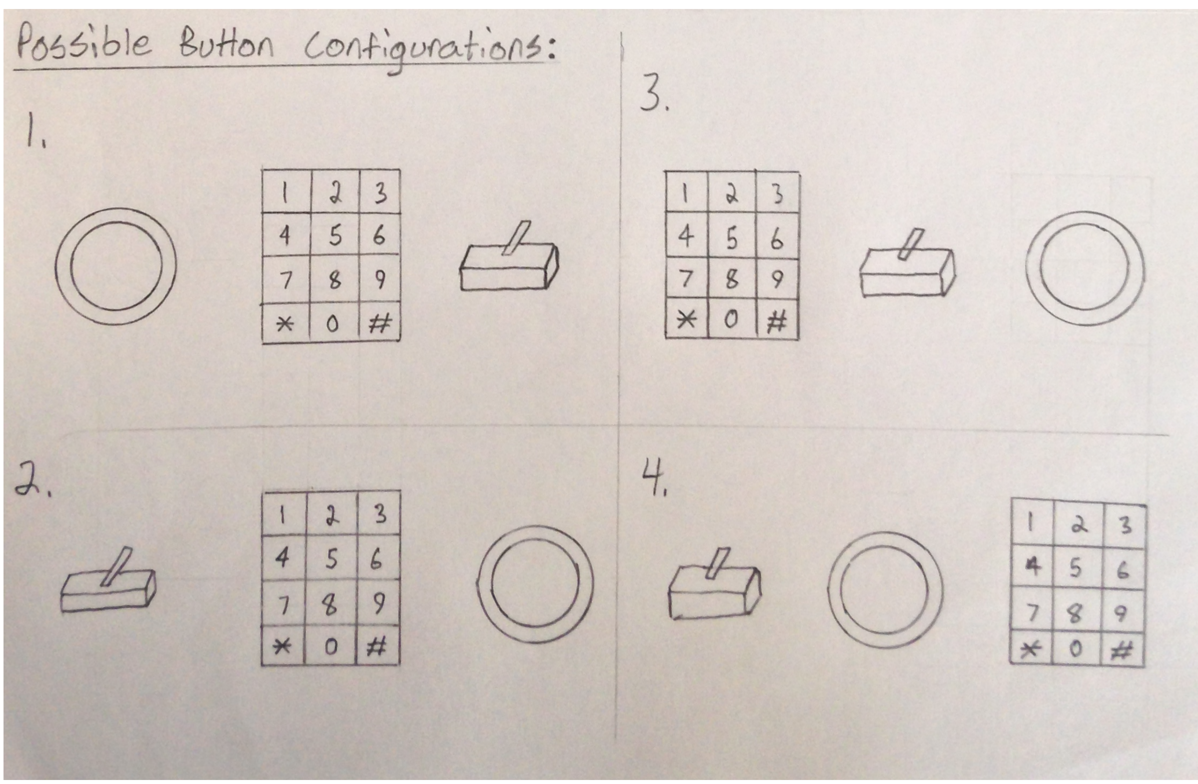


Figure 3-18: Sketch of possible button configurations in the launch briefcase.

3.2.6.2 Display

The display will be a centralized component of all the variables included in the briefcase. Having purchased a display we could link it up to a microcontroller to control all variables of the launch sequence. Conversely, linking a tablet up to the external launch control buttons accomplishes our goal as well. While using a tablet would tackle both processing and displaying information in one unit, the display-micro controller system is has more variability and is potentially cheaper. Though both design approaches have their strengths and weaknesses, thorough investigation leans towards the use of the microcontroller layout for a more feasible and efficient briefcase system.

A pivotal component of the researching and design process is which form of display should be chosen to integrate into the briefcase to display information. There are two options for this task, both of which have their pros and cons: tablet display or component display connected to a microcontroller.

3.2.6.2.1 Display Technologies

When choosing a particular display for the microcontroller approach, two types of screens typically come to mind: LCD and LED. The difference between the two is actually not what is expected because 'LED' screens are actually a type of LCD technology. The difference between the two is their method of backlighting.

So, an LCD or liquid crystal display, is a panel that is made up of two sheets of polarizing material with a liquid crystal solution in between. This panel is configured as such so that when a current passes through the solution it distributes the crystals in a way that either lets the light shine through or not. This is how LCDs work; the difference between backlighting of LCD vs. LED is that LCDs use cold cathode fluorescent lamps (CCFLs) whereas LEDs use an array of light emitting diodes (LED's).

There are multiple benefits of LEDs over LCDs one of which is that LEDs are much smaller than CCFLs and therefore the displays can be made much thinner. LEDs are also much more efficient because they use less power. The major advantage LEDs have over LCDs is the ability of local dimming which is a selective lighting technique that results in a much richer color. CCFLs for instance use a constant backlighting and the intensity of the light is controlled by the pattern of the liquid crystals. The problem with this is with such bright light the display cannot completely eliminate the light shining through. LEDs use of local dimming allows them to dim or completely turn off portions of the backlighting which results in much deeper blacks and a better overall color contrast.

The distinction between both technologies (see **Figure 3-19**) is which form of backlighting they use and it is LEDs that prevail in the determination of a better display. Although LCDs may be a bit cheaper, the use of an LED display will be more efficient and worth the extra cost for a much richer color. When choosing a particular display for implementation into the briefcase, the LED display has become cheaper and more available and thus and LCD display will not be found in many tablets or external component displays.

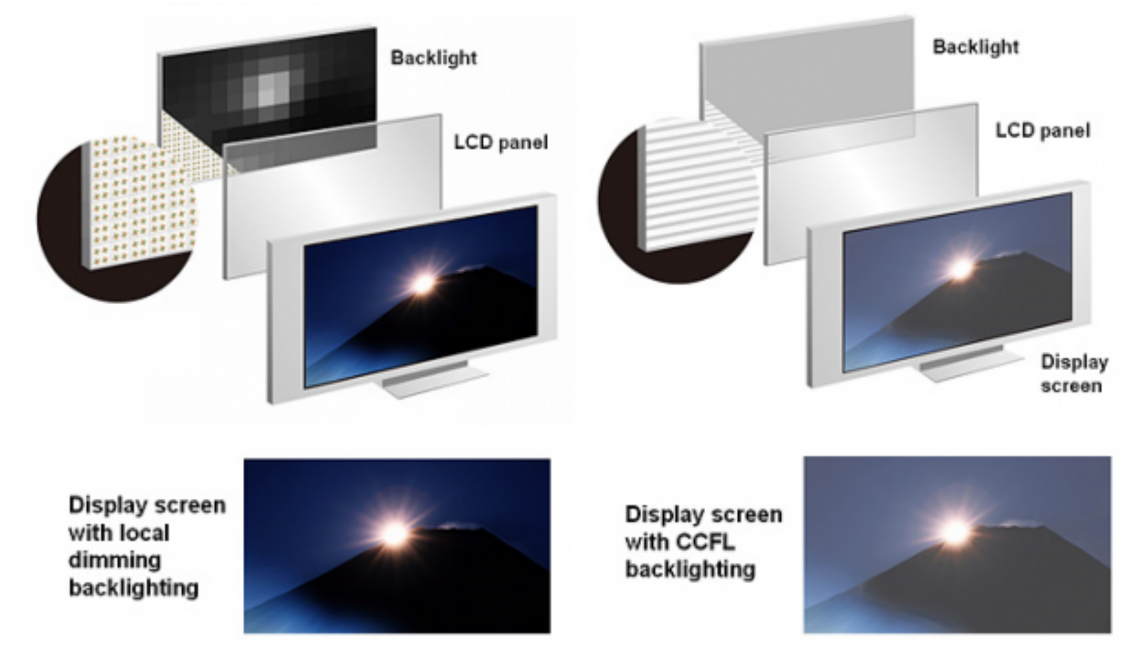


Figure 3-19: Backlighting of LED vs LCD.
Permission requested from Drew Prindle at DigitalTrends.com

Another type of screen display technology is the OLED (organic light emitting diode) which is a type of LED technology consisting of an organic compound that produces light when receiving current. Using this form of lighting is another great way to produce rich color and deep black colors because of lighting of individual pixels.

AMOLED (active matrix light emitting diode) refers to the same type of technology as OLED. The difference between the two is the type of TFT (thin film transmitter) that illuminates the individual pixels. The TFT matrix in AMOLED technology is capable of both storing power (in a capacitor) and charging the pixel. This is the dual function of AMOLED TFTs compared to the single function of OLD TFTs.

The primary method of active matrix OLED TFT is the power consumption. Because of the dual function TFT this type of technology is very effective at saving battery power and thus are beneficial in smartphone and tablet devices.

OLED displays for the most part have the best viewing angle perception. When viewing the screen from different angles there is almost change in the screens picture. However changing the angle may slightly affect the color of the picture, perhaps adding a slight hue to the overall color perspective. This all depends on the specific technology used in the product.

In LCD LED displays a change in viewing angle will much more harshly affect the color. Again this is also very dependent on the specific type of LCD or LED display used. For instance cheaper displays used in older monitors, laptops and smartphones can really show a difference in picture quality.

3.2.6.2.2 Tablet Approach

One design involves the use of a tablet to receive, process, and display information of the rocket's flight. This type of system has given us a number of ideas on how to implement the tablet. Picture fastening the tablet to the lid [top half] of the briefcase, and then running the various launch controls (launch button, key code panel, launch key ignition, etc.) to a USB hub that is connected to the tablet. The problem with this would be how to charge the tablet, which should stay mounted in the briefcase, while having it connected to the controls of the briefcase. With the charger connected to the tablet, power will be split amongst the various controls which is not efficient or acceptable for the external components.

A possible solution to this problem would be to provide the USB hub with switches for each input. Doing this the system would alternate the buttons so charging the tablet will not disturb or harm the external controls. The switches can be manual, in which the user must determine when the tablet needs to be charged and they can switch on the charger and off the buttons and visa versa.

Another option would be virtual switches that can be programmed to switch automatically when the tablet needs a charge. All of this considered, the USB hub still has the issue of not being able to charge and use the buttons at the same time. The idea of connecting the tablet with a USB hub seemed simple in theory but the complications of having the charger and external components connected all at once resulted in a not so ideal outcome for an otherwise desirable system.

Another method of using the tablet for the display would involve taking apart the tablet to get to the motherboard. By doing this the charger can be connected to the tablet’s micro USB port exclusively and the external buttons can be connected to ports on the tablets motherboard, thus solving the issues with using a USB hub. Though this solves one problem, it has its own drawbacks. Exposing the board would require a new cover to allow the buttons, as well as the extra work of wiring them to the board rather than simply connecting components via USB. The application of this method may require more time, work, and money than is necessary for this portion of our project. Although it would be an accomplishment, there are more efficient methods that can showcase the briefcase system as well as be more appealing to the eye as a finished product.

The aforementioned method is not the only idea that would solve the issue of connecting charger and buttons. Another design approach that would solve the problem would be connecting the buttons and charger separately by utilizing the tablets auxiliary port. In theory this would allow us to use the micro USB for charging exclusively, and then the auxiliary would receive information from the buttons. Aux cords have three poles: left and right speaker audio, and the third data. This method would resolve the charging issue, and not require us to disassemble the tablet to connect buttons separately. Although it still does have its own learning curve because it's not as simple as plugging the buttons into the aux port. To do this there would need to be an aux to USB splitter and the accompanying time and testing required to get it working, both on the physical input side, and the receiving side (app programming to register inputs).

There a various formats to the tablet design approach that can be accomplished but the matter of finding which one is the most feasible and efficient is a matter of budget constraints and what the desired result is for the briefcase layout. Though using a tablet in the briefcase will simplify the system by allowing us to receive, process, and display information all in one, it has its own drawbacks when it comes down to application.

3.2.6.2.3 Tablet Options

In today’s industry there is an abundance of options for tablets as the market for them took off when Apple came out with their first iPad. Since then most major electronics companies have developed multiple models of tablets with ranging prices relative to processing power, design, and storage. Every year comes a new iteration slightly better than the last as is for most electronics. Because there are so many companies developing tablets, the table that follows is a selection of some of the top rated tablets currently on the market.

Tablets	Display Size & Resolution	Processor	RAM	Storage	Weight	Dimensions LxWxH
Dragon Touch M7 ~$60	7” 1024 x600	Quad Core 1.3 GHz	1GB	8 GB Flash	1.20 lbs	6.80 x 3.80 x 0.40 inches
Dragon Touch A1X Plus II ~$90	10.1” 1024 x600	Quad Core 1.3 GHz	1GB	16 GB NAND Flash	1.23 lbs	10.24 x 6.50 x 0.39 inches
NeuTab N10+ ~$100	10.1” 1024 x600	Octa Core 1.8 GHz	1GB	16 GB HD	0.8125 lbs	10.60 x 6.70 x 0.40 inches
iPad Air ~$353	9.7” Retina 2048x 1536	A7 1.4GHz	1GB	16 GB Flash	1.00 lbs	9.40 x 6.60 x 0.29 inches
iPad Air 2 ~$450	10” Retina 2048x 1536	A8 1.5GHz	1GB (2GB capacity)	16 GB Flash	0.96 lbs	9.40 x 6.60 x 0.24 inches
Lenovo Tab2 A8 ~$110	8” 1280x 800	Quad Core 1.5GHz	1GB	16 GB Flash	0.8 lbs	8.54 x 5.35 x 0.35 inches

Table 3-12: Comparison of potential tablets. All data from Amazon.com

Programming the tablet will be a matter of choosing between Apple and Android because both an iPad and Lenovo tablet are available for testing from Space Trek. Though an iPad may be more familiar to most of the children because they either have their own or have iPhones, the SWR team has no experience with programming apps for Apple’s iOS. Android OS however uses apps coded with Java for which the team is much more experienced. For this reason an Android tablet would be the best choice because the learning curve would be too great for programming an iOS app for the SWR system.

When selecting a display the major differences between tablets are more noticeable as you go up in price (see **Table 3-12**). For instance, in the ~$100+

price range there will be a more significant increase in processing power, screen resolution, storage, etc. Conversely, the sub-$100 price range would provide average specifications and lower quality materials. The briefcase system will not require a very expensive tablet if it used solely for the SWR app so it can be designed using an average tablet that will not soak up a large portion of the budget. However, if the Space Trek team prefers to acquire the newest model of a high-end manufacturer’s product then the price will not be reflected in our budget because the selected tablet could be detachable from the case and used for other purposes of the Space Program.

If it is desired to program on an apple device, the SWR team already has an iPad Air available from Space Trek. Children attending the space program may be most familiar with iPads compared to other Android and Windows tablets. The problem with implementing this tablet as a display into the the briefcase isn’t just a matter preference though. For the team to use an iPad for the briefcase the application for the rocket launch would have to be programmed in the Swift programming language or another compatible language for the iOS architecture. This would be a major issue with the software side of things because the team programmer does not have any experience with any of the compatible languages. Also, the languages that the team is familiar with do not translate well to compatible iOS application languages.

One perk with using an apple device would be that the SWR team has an iPad available for testing from Space Trek. The particular device is an iPad Air and the team would be able to work with it for a while before purchasing the briefcase tablet.

Another option for implementing a tablet in the briefcase design would be the Surface Pro. This tablet would runs Windows operating system and would thus be compatible with most programming languages i.e. C, C++, Java. Choosing this tablet would be a better option because of the familiarity with programming languages compatible with Windows OS.

3.2.6.2.4 Microcontroller Approach

Another route to implementing a screen into the briefcase would be the use of a component display connected to a microcontroller. With this approach the system could either use a touch screen or a normal LED/LCD screen. Using this design approach for the briefcase in the SWR system was an external display connected to a raspberry pi. The first idea is that this will require a smaller portion of the budget without sacrificing processing power or a lackluster of features. The goal for the SWR is to make it more appealing to the kids as well as making it more convenient for the Space Trek team during operation. Although the raspberry pi connected to an external display may not seem as eye catching as a tablet wired up to some external buttons, it is only one step away from being just as appealing while potentially eliminated excess wiring configurations. The exposed raspberry pi and factory display can have housings created to protect them and either conceal the enclosed hardware or there can be a translucent casing which will not only protect but show exactly the system inside and out, depending on which may be more appealing as this is a science program in which, ideally, the hardware design should be shown off rather than hidden behind a wall that would prevent from further intrigue of the children. The compatibility of the raspberry pi with external devices would eliminate the struggle of designing the system that was identified in the tablet display design approach. The raspberry pi has a total of 4 USB inputs, which each have the capability of systematically connecting 127 total inputs. The briefcase design will only require one of the USB input, having connected all of our components to a hub that can eliminate the use of excessive wiring to the microcontroller.

For testing purposes the display-microcontroller system can be demonstrated using a Raspberry Pi 1 Model B+ that is currently owned by the SWR team before purchasing a new microcontroller for the finished SWR system. This model has a 700MHz Broadcom BCM2835 CPU with 512 MB of SDRAM, full size HDMI port, MicroSD slot, 4 USB 2.0 ports, and 40 general-purpose input/output (GPIO) pins, 26 of which are pinout. Though this microcontroller will work well for testing, the SWR would be designed with the newest Raspberry Pi in mind. Once the system is tested and approved, the finished product will be built with the more powerful current model so that the Space Trek team has the best available to provide a long lasting product.

The Raspberry Pi 2 Model B is the desired microcontroller for the job because it is the newest and most powerful. Having 1GB of ram and a 900 Mhz processor, it will be able to tackle all of the tasks the SWR app will need to do. This raspberry pi processors are recommended to be used with python, but C, C++, Java, Scratch, and Ruby are all installed by default. Another perk of using the raspberry pi is that it includes four USB ports which is more than enough to link up all of our buttons for the launch sequence. The plan with using a microcontroller and externally connected display is to make the design cheaper, more efficient, and ultimately it shall relieve obstacles that would otherwise be encountered by the tablet design approach.

3.2.7 Communications

The Smart Water Rocket will require a communications network to send data from the altimeter in the rocket to the data processing software in the launch briefcase. This system is the most important part of the project because without it the SWR cannot transmit data from the launch. The communications system must have enough range to reach the altimeter or have fast transfer speeds to transmit the data to the processing software quickly after launch.

When designing the Smart Water Rocket system there is a need for data transmission between several components of the system. The primary need for communications is between the altimeter and the briefcase. Whether it be the use of radio, Wi-Fi, or Bluetooth, there cannot be data loss and the range of communication must be a reliable up to 150 feet. Other than the briefcase-altimeter communications there will be a data link between the briefcase tablet and the children's smartphones. The transmission of information between these two devices will likely be carried out by Bluetooth because both devices already have this technology installed and ready to use.

3.2.7.1 Radio

The first that comes to mind when hearing the term radio is AM/FM radio broadcasting. Everyone who has a car more than likely has a receiver for music and talk radio. However, this is a large scale one-way wireless communication of voice and sound via radio waves. Amplitude Modulation (AM) sends information by changing the amplitude of a radio's carrier wave while Frequency Modulation (FM) changes the base wave's frequency.

3.2.7.1.1 Transmission

Using radio waves as a medium for data transmission requires two components. The first is a network device such as a gateway or router that will handle all of the internet traffic of the data communication. Second will be a radio transmitter and receiver, which can be collectively be transceiver. The radio transmitter modulates and electrical signal so that its frequency is compliant with radio communications. Once the signal is modified, it is sent a receiver or a radio antenna. Antennas will be further discussed in a separate section as they are an important component for reliable signal transmission. Antennas are the receiving end of the radio transceiver and they demodulate the radio signal back to its original frequency. The resulting electrical signal is then sent to the gateway/router so that it can be processed by the network. Radio transceivers communicate in both directions, that is, a system can communicate with its network device and visa versa. When signal is sent from the system to a network device the transceivers are in transmission mode and receiver mode respectively. When the signal is sent in the opposite direction, from network device to the system, the transceivers are flipped accordingly. These transceivers can change modes as many as a thousand times per second and this leads to a delay in data communication called latency. Though data transmission through radio waves is potentially cheaper than other forms of communication, the latency would be a major drawback to communication of SWR system.

3.2.7.1.2 Cellular Signals

When referring to briefcase communications the options include radio, Wi-Fi, and Bluetooth. After going through how Wi-Fi transmits data between devices it is important to understand the difference between wifi and the smartphone signal i.e. 3G, 4G, LTE, etc. The first thing to clarify is that Wi-Fi and the 3G, 4G connections are different; Wi-Fi is used between network devices to transmit data over a wireless local area network (WLAN), whereas 3G, 4G connections are signals that smartphones use to communicate with cell towers which operate in much larger ranges.

So, as smartphones improved over the years there was more data processing over the air waves and thus the signal had to be improved as a result. The difference between 3G, 4G is simply the coming of new generations. 3G is now less common as 4G has been phased in, and now 4G LTE is the newest generation. Every generation brings better radio technology and data encoding so that bandwidth can be increased and data transfer rates are also increased.

3.2.7.1.3 Spectrum Allocation

The radio spectrum is the radio frequency (RF) section of the electromagnetic spectrum. This RF part spans from 1Hz to 3000GHz and is shared between about 40 different radio communication services that follow the Radio Regulations (RR) of the International Telecommunication Union (ITU).

For the Unite States, radio spectrum regulation is divided between the Federal Communications Commission (FCC) and the National Telecommunications and Information Administration (NTIA). The FCC is an independent regulations agency which retains spectrum allocation for non-Federal use which includes: state, local government, commercial, private internal business, and personal use. The NTIA operates within the Department of Commerce and allocates radio spectrum for Federal use which includes: Army, the FAA, and the FBI

3.2.7.2 Wi-Fi

Wi-Fi is another form of wireless communication that could be utilized in the Smart Water Rocket system. The wireless local area network (WLAN) is established by using a cable connection to the wall and a network device such as a router or gateway. The router or gateway then communicates with wireless devices within a range of 100-200 ft. The problem with using Wi-Fi for data transmission is the availability of a network. First, Space Trek does not have an available wireless network for use outdoors where the SWR will be operated. If the systems were to connect to a wireless network used for indoor activities, the connection would likely be unreliable and experience transmission issues. So to work around this the SWR would require its own network which include a lot of extra equipment and a much larger portion of the budget. One possible direction is using a Wi-Fi range extender so the indoor network can be relayed to a router outdoors that can be used with the SWR system. Nevertheless using Wi-Fi creates too many obstacles and is not the most efficient or cheap way of performing the task of data transmission between the water rocket and altimeter, especially because there will be such a small amount of data to communicate.

Most people think that the word Wi-Fi is an acronym or has a meaning behind its name but in reality it is just a made up term. Another common misconception is that Wi-Fi is the internet, but it's really just another medium for accessing the web. Wi-Fi is actually a certain type of radio signal. All sound and data is sent through the air in waves. A carrier wave's frequency, amplitude, and phases can be changed to signal information. Wi-Fi is a wireless network that allows network devices such as routers, laptops, tablets, and smartphones to send and receive data. This wireless network is a lot like two-way radio communication.

3.2.7.2.1 Transmission

The process of wireless communication works in a few steps: First a computer's wireless adapter encodes data on a radio signal and then transmits the signal using an antenna. Next a wireless router receives the radio waves and decodes the information to send to the internet through a wired connection. Reversing this process the routers will take information from the internet and send it to the computers.

The CC3000 Wi-Fi module by TI for the MSP family of microprocessors would allow access to the internet and transmit data online as opposed to the Bluetooth device which would be a direct link to the mobile launch pad. Some benefits of using Wi-Fi is that data can be accessed online, making the information accessible without needing a direct connection between the altimeter and the launch pad. This would allow an opportunity to set up a server in which to store all the data from the launches and make it very user friendly for both the kids and the instructors. Some drawbacks include having a poor/no connection if the Wi-Fi signal is not strong enough which would be a serious problem because all the data is sent in real time and unless there is a backup storage there will be no way to guarantee all the data has been received from the flight. With flights that last about 2 second, it's a very precise window of data that we need and to miss any of it would make the WiFi route useless.

3.7.2.3 Bluetooth

Bluetooth is another form of wireless communication that transmits data by manipulating radio frequency signals. The major difference between Wi-Fi and Bluetooth technology is the range, speed, and cost. Bluetooth is generally used for short range, small data transfers and the Bluetooth chips are therefore designed to be much cheaper and require less power. This type of wireless communication is used for transmission of simplistic data between devices such as wireless headsets, mice, keyboards, portable speakers, etc. Even though Bluetooth may be more limited than Wi-Fi, it can still be useful in the SWR because the system will not require a large bandwidth for data transmission.

3.2.7.3.1 Specifications

There are 3 classes with different ranges for Bluetooth: class 1 which is 100 meters but is mostly industrial use, class 2 which is found in most smart phones has a range of 10 meters and lastly class 3 has a range of 1 meter. For this project which can launch a rocket as far as 150 feet, then a class 1 Bluetooth device will be needed. The Texas Instruments CC2540 module is a low powered Bluetooth device that can be used but the range on the device.

Also, newer 3.0 and 4.0 bluetooth technologies are capable of much longer ranges than the 1.0 and 2.0 generations. Therefore, this form of communication will meet the needs of wireless transmission and reduce the cost and power consumption of the SWR system.

3.2.7.3.2 Versions

When using bluetooth technologies it is important to understand the differences between the generations and the capabilities of every version to find a particular bluetooth module that will best fit the needs of the SWR system. Below is a collective list of most bluetooth version which can identify the major improvements that have been made with the technology and help with selection and identification of hardware. Being aware of the limitations of older versions will help bypass any issues with communication needs. Also, specifying what the newer versions of bluetooth are capable of will help in product selection.

- Bluetooth v1.0 & 1.0b
    - The first generation of bluetooth had a lot of problems which made products manufactured with this technology inoperable
    - Another problem with this version of bluetooth was it required hardware device address (BD_ADDR) transmission during the connection process which prevented certain companies from implementing bluetooth services because of the lack of anonymity.
- Bluetooth v1.1
    - Accredited by IEEE Standard 803.15.1-2002
    - Fixed many of the problems with v1.0b
- Bluetooth v1.2
    - Provided faster connection and discovery of devices
    - Transmission speeds up to 721 kbit/s
- Bluetooth v2.0 + EDR
    - Introduced Enhanced Data Rate (EDR) which provided a nominal transfer rate of 3 Mbit/s but the practical rates is about 2.1 Mbit/s
    - Bluetooth v2.0 without EDR was available and had minor improvements but did not offer the higher data rates of EDR.
- Bluetooth v2.1 + EDR
    - The core feature release with this iteration was the addition of secure simple pairing (SSP)

    - SSP again increased the speed of pairing as well as providing better strength of security.
- Bluetooth v3.0 + HS
    - This generation offered a theoretical data transfer of up to 24 Mbit/s although it was not over the bluetooth link.
    - The bluetooth link was used to pair the devices and then the data transfer must be carried out over a 802.11 WLAN.
- Bluetooth v4.0
    - The Bluetooth SIG established the new as Bluetooth Smart which included Classic Bluetooth, Bluetooth high speed, and Bluetooth low energy.
    - Classic Bluetooth followed the old connection protocols whereas BT high speed was used WiFi for higher data transfer rates.
    - Bluetooth low energy was introduced to provide much quicker pairing of simple devices and the technology was used primarily with ultra low power devices powered by coin cell batteries.

3.2.8 Software Components

There are several software components that will be used for the Smart Water Rocket system since there will be multiple types of hardware across different platforms. For the rocket microcontroller design, depending on whether TI or Arduino based MCU's are used, Code Composer Studio or the Arduino IDE will be used respectively. When designing the launch pad MCU / tablet, the software components being used depends solely on which piece of hardware is chosen. If a tablet is chosen, it can either be programmed using Android, Apple or Windows operating systems.

3.2.8.1 Code Composer Studio

Code Composer Studio is an integrated development environment specifically for the development of applications on Texas Instruments embedded processors. Code Composer was acquired by T.I in 1997 and later appended with the world studio to become code composer studio. CCS was released up until version 3.3, when it released an IDE called Code COmposer Essentials, which was based on Eclipse, specifically for the MSP430 line of microcontrollers. Code Composer will be the IDE of choice given the option of using one of T.I.'s microcontroller units.

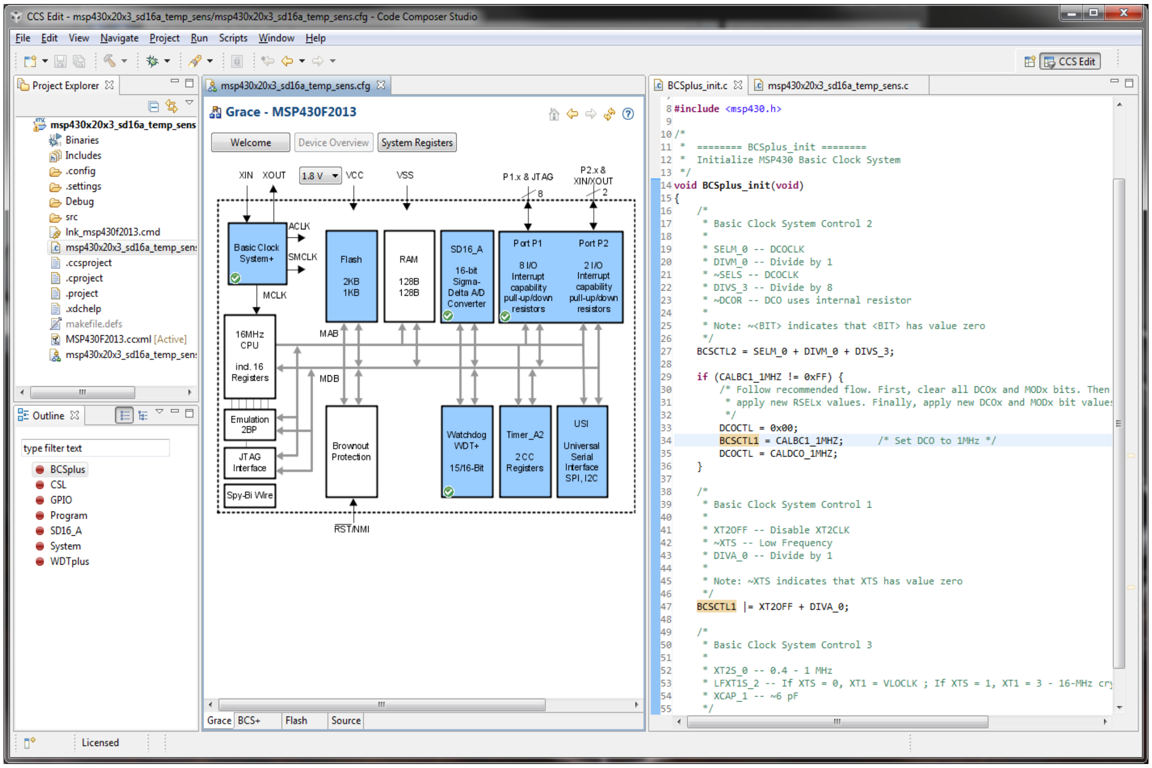


Figure 3-20: Screenshot of Code Composer Studio (Requested 12/1715).

As with most IDE's Code Composer Studio has a simple layout with project files on the left, main coding in the middle and additional coding or pin layouts on the side. CCS allows the user to choose which family of MSP's that they are using and will automatically update the registers and pin layouts accordingly. **Figure 3-20** gives an example of what the device overview will look like in Code Composer Studio.

3.2.8.2 Arduino IDE

The Arduino IDE is a comprehensive integrated development environment especially suited for students and aspiring developers. Arduino is a much smaller and more specialized company when compared to Texas Instruments, as such their focus on microcontroller and kit design is much higher than those of T.I. This means that the community behind Arduino is much stronger and getting help through FAQ's and forums is much easier and most times guaranteed.

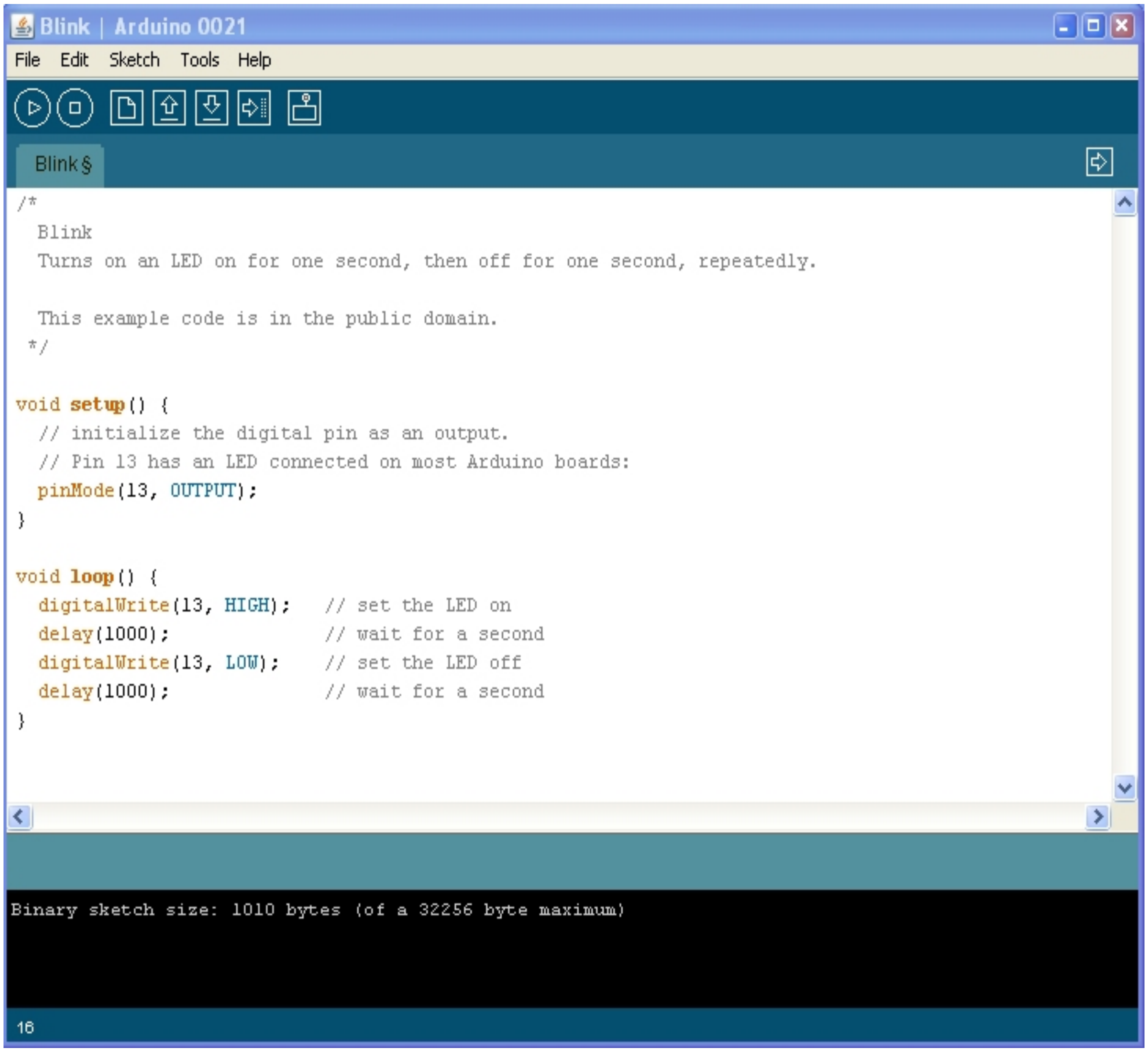


Figure 3-21: Screenshot of the Arduino IDE (Requested 12/7/15).

From **Figure 3-21** it can be seen just how simple in design the Arduino IDE is from the TI IDE. By being much smaller and simpler the Arduino IDE takes up much less space and is usually more efficient than CCS. Execution times across both are exceptionally fast and any differences would be negligible.

3.2.8.3 Android Studio

Android is an operating system developed by Google which is primarily used for mobile touchscreen devices. It was first developed by Android Inc., which was later bought by Google, for which they have been developing since. It is the most widely used operating system in the mobile market as of January 2015 it has sold over a billion devices which is approximately 4 times what IOS and OSX sold together according to the Gartner Report. Android studio is the IDE in which android applications are developed. It was in an early access version 0.1 as of May 2013 and has just released its latest release of version 1.5 as of November 18. 2015.

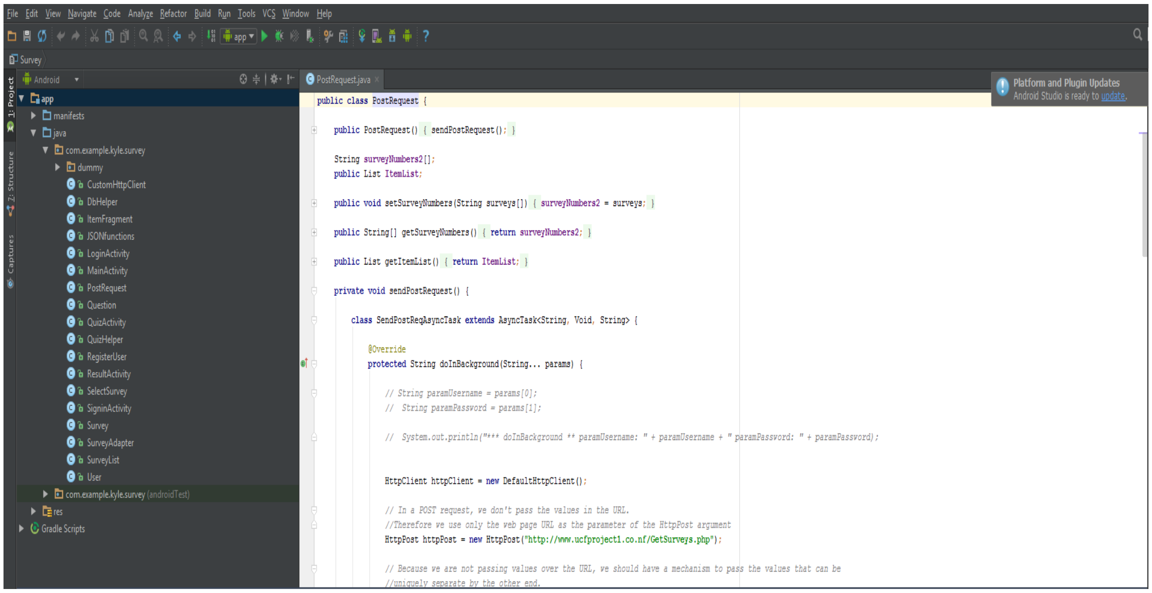


Figure 3-22: Screenshot of Android Studio

When looking at **Figure 3-22** above, the Android Studio development environment is clearly a very elaborate environment. There are a lot of features that come standard with Android Studio and many more additional features. When testing an android application, the IDE comes equipped with an emulator that can act as a mobile device.

3.2.8.4 Apple IDE

Apple created and developed iOS which is the operating system that is used exclusively for all mobile devices created by Apple. The iPhone was the first mobile device of its kind and set the standard for new phones in the industry. With the ability for commercial and private development of apps, a whole new industry was created for the development in mobile applications.

The IDE that is used to create iOS applications is called Xcode. Xcode was first released in 2003, with the latest stable release being version 7.1. Xcode supports many languages including C, C++, Java, Python, Ruby, Swift and more.

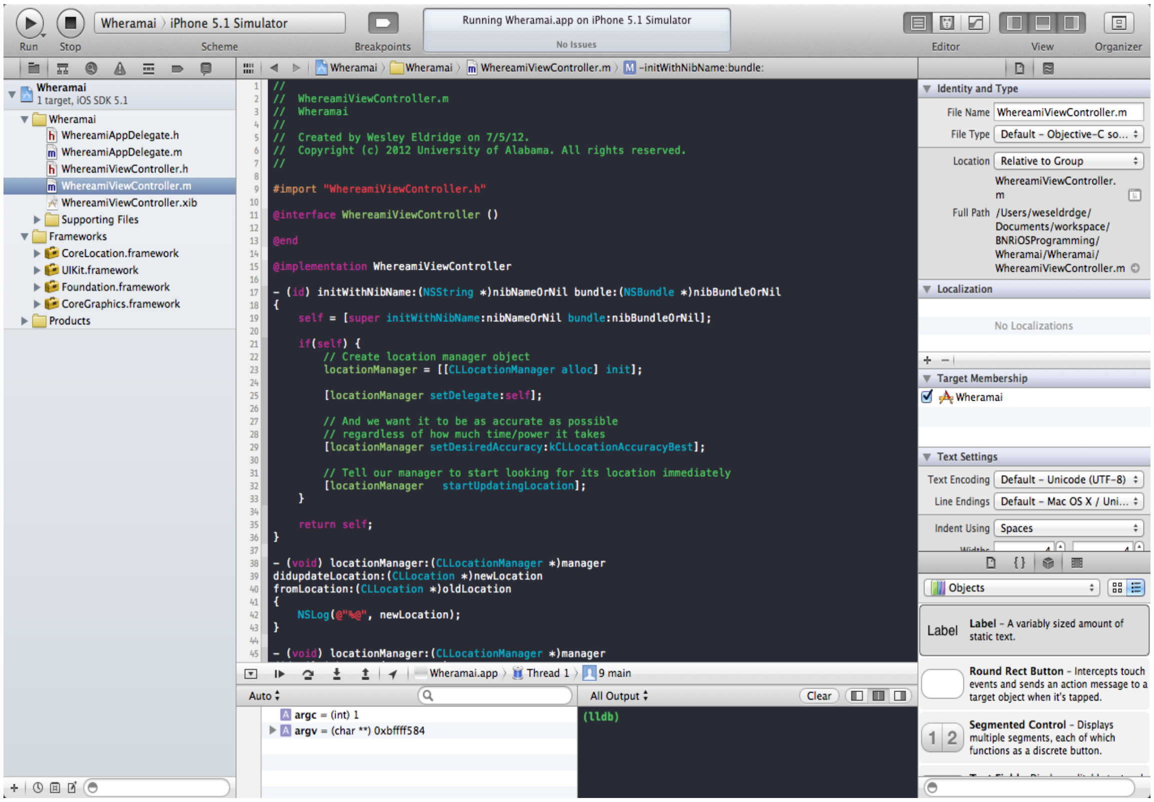


Figure 3-23: Screenshot of Xcode (Requested 12/7/15)

As with all Apple products, the Xcode IDE is sleak and user friendly, with many helpful toolbars located on the top and sides. **Figure 3-23** is an example of what the Xcode development environment looks like, with projects on the left side and the main code in the center.

3.2.8.5 Windows IDE

While Apple and Android have the biggest share in the mobile operating systems market, windows is still a sizeable contender comparatively. With the release of Windows 10, Microsoft has just made it possible to now program cross-platform between mobile devices and personal computers. In order to create a windows application the windows 10 SDK will be used. The Windows 10 SDK was released July 24, 2015 included with Visual Studio 2015. Visual Studio 2015 will be used to write an application for a windows 10 mobile device but can also write programs for many other languages.

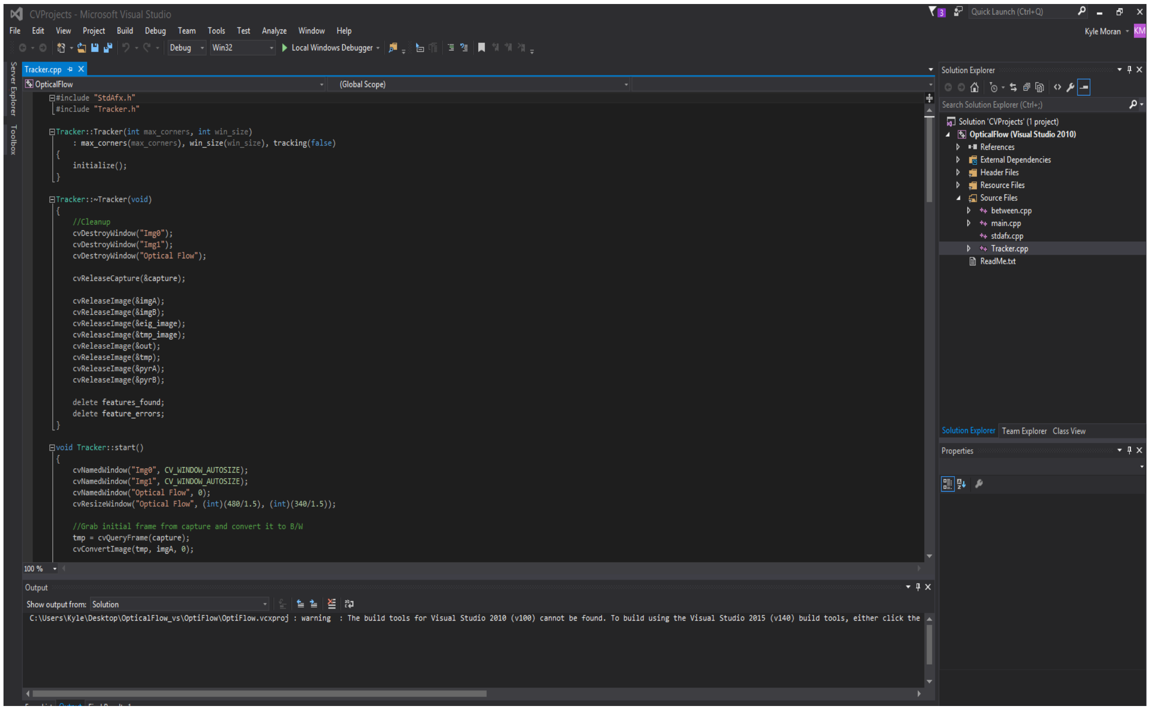


Figure 3-24: Screenshot of Visual Studio 2015

Visual Studio (see **Figure 3-24**) is an extremely power development environment and is capable of compiling many different types of languages. There are still many more features that can be added to visual studio that can allow for very large scale projects to be developed nationwide. Microsoft continues to update Visual Studios with the latest build being November 2015, which means that there will be plenty of online help and resources.

3.2.8.6 Eagle

Eagle stands for Easy Applicable Graphical Layout Editor and is a powerful PCB design tool specifically for engineers both in the job and at school. Eagle has been one of the best choices for PCB design in the past 25 years and has supported hundreds of thousands of electrical engineers worldwide. The simplicity of the software provides a fast learning curve and the openness of eagle design resources allows for an extensive library for any component that a student would need. In addition, the flexibility of the software has been growing in its capabilities as well as its workflow compatibility.

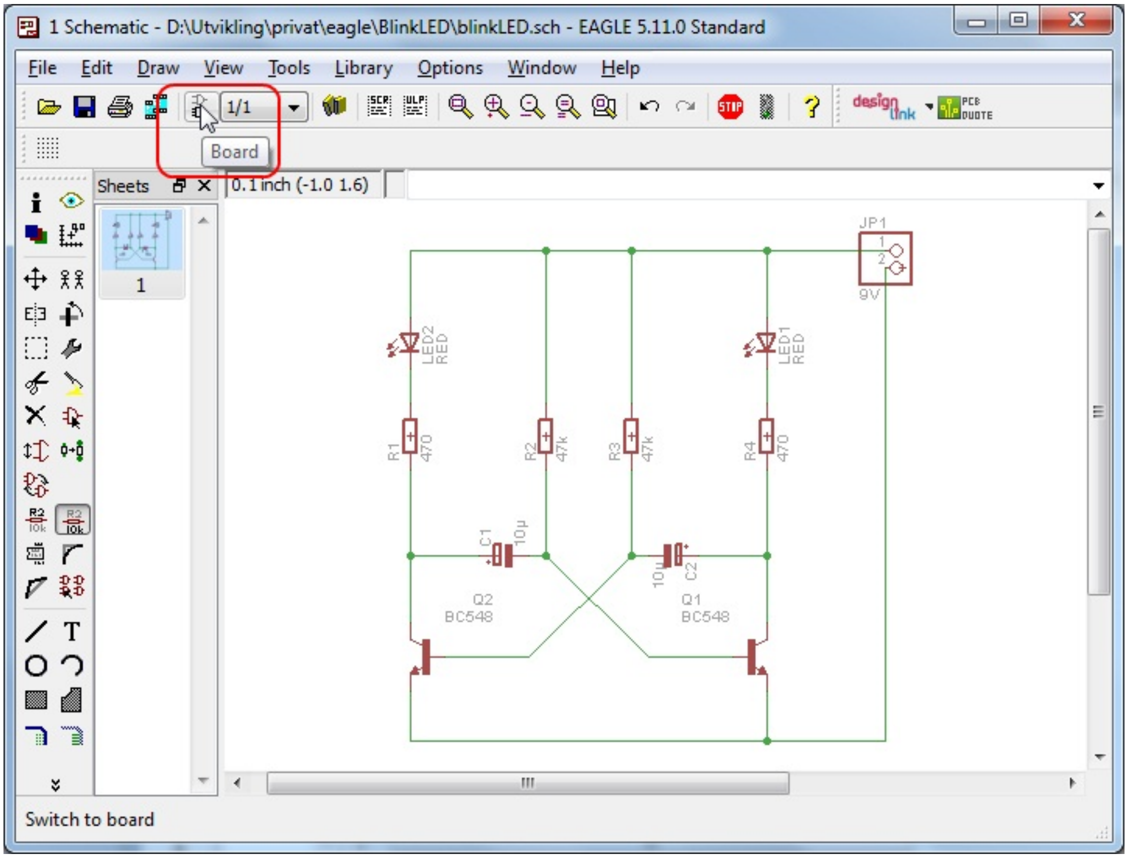


Figure 3-25: Screenshot of EAGLE's layout editor

There are three main modules to eagle: Schematic editor (see **Figure 3-25**), layout editor and autorouter. The Schematic editor will allow the user up to 999 sheets of precision schematics. It also includes Electrical Rule Check which will error check the schematic between modifications. Also it allows for Hierarchical designs so that schematics designs can be structured. The schematic editor also includes automatic board generation, support of assembly variants, net class matrix as well as other features.

Additionally there is the layout editor which allows for quick manual design of the entire circuit board. It also includes an error checking and differential pair routing, which allow for easier design by catching mistakes early on. Another feature of the layout editor is that it can meander command for length compensation of signals and output manufacturing data with the CAM processor.

Lastly the autorouter allows for easy integration between manual and automatic routing. Some features that come with the autorouter include a Grid-based Ripup & Retry router, TopRouter with gridless routing algorithm and interactive Follow-me router for DRC verified. All of these features allow for easy automatic routing as well as precision manual routing when necessary. All of these modules include error checking and use the layout's design rules.

For the purposes of this project, all of the schematics and PCB designs will be designed using Eagle. Not only is this one of the most common resources available, it is available in Windows, Mac and Linux which makes it compatible between all of the team members.

3.3 Related Technologies

Several custom systems have been researched for the Smart Water Rocket because there are no commercial products available that will work within the constraints of the project. To try to reduce the design workload the team researched pre-existing and related technologies. Research into these technologies proved essential in understanding what is required to design other parts of the project.

3.3.1 Power Management

The autonomous systems in the SWR require consistent power to remain functional. There are many options for pre-existing technologies to power equipment under various conditions and time intervals. These technologies are field tested and guaranteed to function but will not be wholly applicable to the SWR system.

3.3.1.1 Power Banks and Large Capacity Rechargeable Batteries

A power bank is a device that functions as a high capacity rechargeable battery and are often used to recharge consumer electronics. They are most often powered by either lithium-ion or lithium-polymer batteries and can deliver low voltages with varying capacities. The largest of these devices can deliver up to 20 volts and store 34.2 amps (rated at 4 amps), which will provide enough power to run the autonomous systems for at least one launch. This specific bank can recharge itself while delivering power so it can be trickle charged using solar panels to allow for additional launches. There are several drawbacks to using a power bank like this though. These devices are unreliable when delivering their full voltage because delivering the full voltage, or close to it, will overload the voltage regulator and cause the system to shut down. This is included as a safety feature to reduce the chance of the product breaking due to overheating but its inclusion hampers its usefulness to the SWR system. Additionally, these devices are very expensive. They can cost between $50 and $200 for the unit by itself and there are additional costs for every required cable. A power banks are very useful for other applications but there is no real application for this system.

Many solar panel kits come with large capacity rechargeable batteries and these batteries can be purchased separately to be used for other purposes. Most large capacity batteries are either lead-acid or arrays of lithium-ion cells. Lead-acid batteries are cheaper than their lithium-ion counterparts but will still cost between

$50 and $200 depending on whether it is flooded or AGM. Lithium-ion batteries must be put into an array or bank to deliver more than 3.7 volts and store more current. A lithium-ion array with the same power specifications of a lead-acid battery will be three to five times more expensive than the lead-acid battery. Currently the cost of a pre-fabricated lithium-ion array is too great for the SWR system, but smaller arrays and singular cells are cheap enough that a custom array could be constructed and remain cost effective. Lead-acid is still the most cost effective technology for prefabricated large capacity batteries.

3.3.1.2 Solar Panels

Solar panels are large panels consisting of multiple solar cells that rely on the photoelectric effect to produce electricity. The SWR system intends to use solar panels to charge the battery for the autonomous systems in order to increase the number of launches that can be performed before the batteries need to be completely recharged. There are many companies whose sole purpose is to fabricate and sell solar panel systems that cater to all needs. These systems range from 640 watts, used to power entire houses, to 5 watts, used to power small consumer electronics. These systems can cost thousands of dollars but include a rechargeable battery and all of the required accessories. The benefit to a pre-made system is that all of the components are guaranteed to work together and last for a predefined period of time. Many companies will offer to design a customized system to meet a specific need. The main downside to a pre-made system is the cost. Individual solar cells can be purchased and assembled into an array for only a fraction of the cost of a pre-assembled panel. Even the cost of a solar panel alone is much less than the cost of an entire solar system. The SWR will use either a custom made solar panel or a series of individual solar panels to reduce the overall cost.

3.3.2 MEMS

Microelectromechanical Systems (MEMS) usually consist of a central unit that processes data (the microprocessor) and several components that interact with the surroundings such as micro sensors. Utilizing material properties, sensors can be designed directly from, or on a material like silicon.

The MEMS field is full of different devices and its applications vary from consumer electronics all the way to industrial devices. MEMS are made up of components between 1 to 100 micro meters in size (i.e. 0.001 to 0.1 mm), and MEMS devices generally range in size from 20 micro meters to a millimeter. At the time of choosing a viable pressure sensor, size and weight were very important design considerations. By utilizing sensors with MEMS sensing elements, the team will be able to keep the size of the Smart Water Rocket's altimeter within our constraints.

The utilized pressure sensor, BMP180 by Bosch, is based on piezo resistive principles. This means the sensing element uses semiconductor properties to analyze the change in electric resistivity and output the information. The principle of piezo-resistivity is widely used in MEMS and allows sensors like the BMP180 to have high resolutions in such small packages.

From a manufacturing and cost point of view, MEMS is also very appealing. Considering MEMS principles allow for the manufacturing of multiple sensors from a single silicon wafer, the return from the manufacturing process is very high. In many cases the cost of the product will depend on the size of the package. With the BMP180 the team was able to obtain an inexpensive, small sensor that clearly benefited from the use of MEMS.

3.3.3 Automated Launch Pad

There are no specific water rocket launch pad stores but there are several designs freely available on the internet for automated water rocket launch platforms. These designs are perfect for an instructor to follow to construct a sophisticated automated launch platform to use in demonstrations. However, none of these will work for the Space Trek rocket because they would prefer to modify their pre-existing structures. Additionally, none of the pre-existing systems account for the repeated use the Space Trek system requires and will not have the reliability and reusability required.

Space Trek uses custom designed 3D printed parts to build their rockets out of reusable materials. They also have special software, designed in house, to do pre-flight analysis by using different amounts of fuel and exit nozzle sizes. They do not want to modify this system so the SWR must conform to the constraints set by the old system. The same launch pad mechanism is used in the pad and in the test chamber so, the launch pad parts are an integral part of the system and cannot be severely modified. To maintain the functionality of the pre-existing system, SWR cannot utilize any pre-existing automated launch pad designs.

3.3.4 Radar Altimeter

Just like pressure altimeters, there are other ways to effectively obtain altitude readings. One of the most commonly used for spacecraft is the Radar Altimeter. This type of altimeter works by sending out a beam of radio signals to the ground directly below. When this signal gets reflected on the ground below, an antenna on the aircraft measures the time it takes the signal to reflect back to the aircraft. These altimeters are extremely useful for military aircraft flying in low altitude to avoid detection and targeting.

Unlike a pressure altimeter, that measures the change in pressure between a set initial conditions, usually sea level, a radar altimeter measures the distance between the ground and the object. Although an advantage low altitude, the higher the altitude, the greater the delay of the wave reflecting back at the aircraft and as such making the data not as accurate. Also, weather and terrain conditions can affect the wave propagation and increase the reflection’s delay.

This type of reading is extremely useful in aircraft, but in the case of small rockets like our Smart Water Rocket, such a device would be too much. Since the Smart Water Rocket device is not intended to fly and keep a set distance from the ground, which is why a pressure altimeter is such a convenient solution. This however does not change how effective a radar altimeter is, and given different goals, a radar altimeter could have been implemented inside the Smart Water Rocket.

4. Related Standards

The Smart Water Rocket must adhere to several standards imposed by different organizations. These standards ensure that the rocket will interface with other technology, remain upgradable and repairable, and operate within all governmental safety standards.

4.1 I^2C Standard

Philips Semiconductors (now NXP Semiconductors) developed a simple bidirectional 2-wire bus for efficient inter-IC control. This bus is called the Inter-IC or I2C-bus. Only two bus lines are required: a serial data line (SDA) and a serial clock line (SCL). The I2C-bus is now a world standard with companies like Siemens, Motorola, TI, Bosch and many others implementing it into their devices.

For the purpose of the SWR system, using a compatible microcontroller is essential since the looked at pressure sensors, accelerometers and Bluetooth module are all I2C compliant. By utilizing this standard the pressure sensor will be able to effectively communicate with the microcontroller and obtain the most accurate data.

4.2 Launch Standards

The federal government has strict regulations for amateur rocket launches. The SWR must be especially careful in following these standards because of Kennedy Space Center's proximity to the air force base at Cape Canaveral. There are three classes of amateur rockets which increase in capability and regulation; class one, class two and class three. Class one amateur rockets are defined using the following list of qualifications from the electronic code of federal regulations: unmanned, uses less than 125 grams of slow burning propellant, constructed out of wood, paper, or breakable plastic, contains no substantial metal component, weighs no more than 1,500 grams. The rocket in use by Space Trek is well within these guidelines and can be launched at Kennedy Space Center. Class one amateur rockets must launch on a suborbital trajectory, remain in US territory, and not create a hazard to other persons or aircraft. The SWR poses no danger to persons, property, or other aircraft. The maximum launch height and distance of the rocket are low enough that Space Trek will not have to worry about breaking any federal regulations.

4.3 Health and Safety Standards

NASA has a published set of standards for water rocket launches. Fortunately, all of these standards are already adhered to in Space Trek's system.

- Only plastic bottles should be used, and should be discarded after 10-15 launches
- Children must be supervised at all times during the launch
- The launch area should be at least 30 meters long
- The launcher must be anchored in place
- Rockets must not be pointed at other people, animals, or objects
- The team member pressurizing the rocket should wear eye protection
- The bottle rocket should be pumped no higher than 50 psi and never higher than 90 psi
- All students and observers must stand behind the launch pad during and after pressurization
- Only start the countdown to launch once the area in front of the launch pad is clear
- In the event of an unsuccessful launch, take care in approaching the rocket because it is still pressurized and may launch or explode unexpectedly
- Only team members should retrieve the rocket

The current launch pad is heavy enough to be considered anchored into the ground on launch and the SWR will maintain the weight distribution of the cart. To ensure that bottles are not used past their intended life cycle a launch count will be included in the data sent by the altimeter and a warning will be present when the number of launches exceeds 10 launches. To prevent unintended over-pressurization a hard limit of 100 psi will be set on the compressor so that the instructors can use higher pressure for testing. Using software, a soft limit of 90 psi can be set so that the students do not exceed the safe parameters. If there is an unsuccessful launch, a remote launch abort system will force the rocket to depressurize safely using a release valve so that nobody has to approach the pressurized rocket. By including those features in the rocket, the SWR will ensure the safety of all participants by adhering to the standards set by NASA.

4.4 Wi-Fi Standards

- 802.11a - Transmits at 5GHz frequency with data speeds up to 54 megabits per second. Utilizes orthogonal frequency-division multiplexing (OFDM) to reduce interference by splitting the radio signal in several signals before reaching a receiver.
- 802.11b - Transmits at 2.4GHz frequency with data speeds up to 11 megabits per second. Utilizes complementary code keying (CCK)

modulation to improve data speeds. This is the slowest and most inexpensive standard so it was once widely used. However, faster standards have become increasingly more affordable so this standard has mostly been phased out.

- 802.11g - Transmits at 2.4GHz frequency like 802.11b but can send up to 54 megabits per second because of the OFDM coding that 802.11a also uses. Though it is capable of 54 mbps, this standard usually only gets up to 24 mbps because network traffic.
- 802.11n - Transmits at 2.4GHz frequency with data speeds up to 140 megabits per second. Currently the most widely used, this 802.11 standard has significantly better range and data speeds than the previous standards. 802.11n can transmit up to four channels of data, each reaching up to 150 megabits per second. Most routers only allow for two or three channels though. 802.11n is also backwards compatible with 802.11a/b/g wireless frequency bands.
- 802.11ac - Transmits at the 5.GHz frequency band with data speeds up to 450 megabits per second on a single channel. Backwards compatible with 802.11n and thus 802.11a/b/g. The 802.11ac standard is as expected much faster than the previous standards, as well as less susceptible to interference. Transmission is capable of up to eight channels.

5. Related Constraints

The Smart Water Rocket system must operate within several constraints. These constraints are set by physical limitations, limitations set by Space Trek, and limitations within the team. The SWR team has several plans made to ensure that the project will be completed, to Space Trek's standards, within the time allotted.

5.2 Microcontroller Constraints

The microcontroller must be able to meet a certain expectation set by the team within the rocket in order to be viable for Space Trek. A high demand for reliability will rely on strong resistance to water and impact damage, as well as transmitting data. Without meeting these requirements it would not be worth replacing the system if it does not last long enough. In order to meet these constraints we will custom fit a nose cone to protect the unit from all forms of damage. While also maximizing space efficiency on the board to keep all of the components as compact as possible to save as much space as possible. When these constraints are met the microcontroller should be able to live a durable life flying on many rockets.

5.3 Communication Constraints

Transmitting data is one of the crucial aspects of this project which will help maximize efficiency and save time for the instructors to gather all of the data from the launches. The limitations on the communications have created constraints that must be met in order to create a reliable system, which are a minimization of data lost, and transmission time as well as maximizing the amount of valuable data that is sent. If these constraints are met the communication device will be the one with the best overall score in these categories. Since each of the different communication choices each have different advantages there should be specialized constraints for each of those as well.

For wireless communication networks, there should be a proper source of wireless signal at any given time for the rocket whether it is in the classroom, outside or in the air. Without this the signal will not be strong enough to transmit data, and with a limited amount of internal storage on the rocket, without the ability to transmit data the choice of using wireless networks would be futile. When designing a rocket use a wireless network the consistency of the connection is key for the constraint to be met.

When designing a rocket using Bluetooth communications to transmit data directly to the launch pad there are certain constraints that must be made in addition to maximizing efficiency. The Bluetooth signal must be strong when within 33 feet as specified by class 2 Bluetooth device specifications. When designing the rocket transmission, it is possible to store data that is outside the range of the Bluetooth device, given that it is for a short amount of time and that all of the data can be sent at once, as soon as the rocket is within range of the launch pad. With these constraints met the rocket will be consistent and reliable which are the key components in the design of this project.

5.4 Health and Safety Constraints

The SWR system has to follow safety constraints set by Space Trek. These constraints are in place to ensure the safety of the students and instructors by reducing the chance of accidental injury. There have been no injuries as a result of water rocket launches at Space Trek and the SWR intends to maintain that record.

One of the biggest safety issues with the current system is that bystanders are often unaware that rockets are being launched and can wander in range of the rockets. The SWR will rectify this using the launch briefcase. The new launch procedures will be loud enough to ensure that anybody near the launch site will know that a launch is happening and will get to a safe area.

Space Trek has eye protection for every student and instructor and insists that proper eye protection be worn whenever a rocket is under pressure. Currently, they store the goggles in bins stored on the carts. There will be plenty of room for both bins of goggles on the briefcase cart. The bins will be secured to the briefcase cart so that they remain stationary during transport but can be removed so that the goggles can be dispersed.

The rockets can only be launched between 10 and 30 degrees for safety. Below 10 degrees the rocket launches laterally and becomes a missile. Above 30 degrees the rocket has a high chance of landing too close to the launch pad and hitting a student. The software on the SWR will ensure that the rocket can only be launched between 10 and 30 degrees for the safety of the students. The instructors will need be able to bypass this limit for demonstrations so a hardware limit will not work for this design.

5.5 Economic and Time Constraints

Space Trek has imposed a set of constraints on the Smart Water Rocket team that detail how the budget can be used and when the project must be completed. Space Trek is very flexible with these constraints but the SWR team is dedicated to adhering to these constraints as much as possible.

The SWR project has a maximum budget of $1000 for materials. Space Trek has stressed that the cost of the project must be kept as low as possible by reusing any components already available in Space Trek’s workshop. Since Space Trek tries to design and build all of the systems for their programs in house they have accumulated many excess components that they are letting the team use. Reusing components will also help keep the project environmentally friendly as the team is trying to use recycled materials as much as possible. The SWR team is to make a list of components required to build each system and Space Trek will determine which components can be salvaged, which they already own, and which need to be ordered. Space Trek and the SWR team will keep track of the total cost of all purchases to ensure that the project remains within the $1000 budget. The most expensive components of the project will be the computer and the batteries. Space Trek hopes to have the computer donated and the batteries can be salvaged from other projects. In the event that either of these components cannot be acquired for free, the SWR team will have to find them for a reduced cost. The other components of the system are not particularly expensive but the combined cost will certainly take up a significantly portion of the total budget.

Space Trek operates its programs year round but they are busiest in the summer. They have told the SWR team that the modifications to their systems need to be completed by summer 2016 so that it can be used in their programs. It will be difficult to modify the systems while classes are going on. The carts need to be functional while classes are in session so that the students do not miss out on launching their rockets. Any modifications must be done during off-hours to ensure that the pads remain operational. In addition, Space Trek has said that they want parts of the system to be operational while the rest of it is being constructed. They would like the launch pad automations to be in place before the simulation software or altimeter are completed to increase efficiency and reduce strain on the educators. The SWR team is going to work quickly to ensure that the automations are completed, tested, and installed all at once. The remaining features will be built as time allows until the entire system is assembled. The whole system must be tested as a cohesive unit so the assembly must occur before the deadline. Again, Space Trek has stated that they are willing to be flexible with their time constraints but the system must be working for the children. It is unacceptable for the launch system to be non-functioning for any classes.

6. Software Design

The Smart Water Rocket requires a robust software to receive and process data from the rocket. Without the software, the SWR will be no more efficient or appealing than it was before. The software must decipher all the transmissions from the altimeter, process and store the data it receives, and display it in a user friendly manner. To achieve this goal there will be two microcontrollers, one on the ground and one on the rocket, which will be programmed to communicate with each other during the launch. During the flight, the onboard rocket MCU will be equipped with sensors to gather the data to then be sent back to the launch pad in as close to real-time as possible. Once the rocket has landed, all of the data will be displayed at the launchpad for immediate viewing.

6.1 Software Decisions

The team has limited resources and must apply these resources where they are needed most. There is only one computer engineer in the group so the software must be prioritized according to necessity. The team must complete all of the goals set by Space Trek while still finishing the project within the given timeframe. There are several main decisions that need to be made for this project: first the hardware that will be used for the rocket and the launchpad and which communications device will be used to transmit the data during the launch. When choosing the microcontroller for the rocket it is important to note the processing power when compared to the sample rate of the sensors. Next when choosing the hardware for the launch pad it will be important for the operating system on which the program will run to be powerful enough to handle a sophisticated application capable of meeting the goals for this project. Lastly the communications device, whether Bluetooth or Wi-Fi, needs to be able to maintain an otherwise consistent connection or be able to transmit data reliable with minimal loss. Designing the software depends on which hardware is chosen and will be designed accordingly.

6.1.1 Rocket Microcontroller Software Summary

With most of the programming emphasis going towards the design of the mobile application for the touch screen launch pad device, there is still a great amount of significance in the design of the microcontroller in the rocket. The main goals of the rocket is to use sensors to collect data on its height, velocity and acceleration. With many calculations per second it becomes important to be able to send that information to a larger storage device. On the launch pad will be the mobile device which will receive the information from the microcontroller and manipulate the data as described in the mobile application section. The design of the microcontroller will present a challenge because of the components that will be connected directly to the controller unit and will need to be arranged accordingly between all the members. Communication not only between the rocket and Launchpad but also between the team members is important here, since multiple components will be working in conjunction between the software and hardware.

As the information is being gathered by the microcontroller, the data will then be transmitted to the mobile device at the launch pad through the communication medium of choice. There are three main communication mediums that can be readily used with the microcontroller, these include Bluetooth, Wi-Fi and radio. Since radio waves are intended for large distances the precision within a 100 foot range can cause problems for the team. Using a wireless network connection to transmit the data to an online database where the mobile device can then execute a script to pull that data from the database and give access to the users to extrapolate the results. While this would be an excellent feature and worthy of the time it would take to complete, this wireless network located at Kennedy Space Center is not fully covered, especially in the area where the launches are performed, the wireless signal will not be strong enough for the performance that will be demanded of it. Lastly is the use of Bluetooth which would create a direct link from the rocket to the mobile launch platform in which to directly store the results. By transmitting the raw data, a large amount of memory will be saved and allow for many launches to be performed before any memory will need to be cleared to make space for new data. In light of this, there can also be an additional storage unit installed onto the mobile device to allow for up to hundreds of launches before a new memory device is required.

When developing the microcontroller there needs to also be careful consideration to how many calculations are performed per second. While sending raw data will take strain off the MCU, there still needs to be calculations performed in order to gather the data, which means there needs to be a balance between accuracy and performance. Keeping this in mind will prove to an excellent design strategy when working with the microcontroller and will help to maximize efficiency in the design of the custom altimeter.

Data that has been sent to the online database has many other units and calculations that can be performed to produce more interesting results. With these possibilities units such as, power, work force, kinetic energy and others can be easily calculated and theoretical graphs can be created such as time until escape orbit, or a low earth orbit. This creates many interesting and unique possibilities but the design of a cloud database has been listed as a stretch goal because it presents a lot of time to initialize as well as maintain and the priority rests in transmitting data faster and it might not be the fastest way.

Considering all these factors the design of the microcontroller will use the most effective communication medium in order to transmit the data from the launch. After some tests are conducted from the team it will be easy to determine which medium will be the best to use, but until then the team has decided to go ahead and start focusing on one idea which has the most potential, which will be the Bluetooth adapter. By using Bluetooth there is a possibility that the range of the rocket is too far from the launch pad to receive any results in real time but with creating an onboard storage device, the data can be stored internally until it reaches the ground where it can transmit all the collected data at one time once it is in range. This saves the problem of a bad connection because the Bluetooth may be out of range at the furthest point in its flight but at some point in time will come down and will be within range to transmit to the launch pad. When compared to the wireless network which could have a faulty connection throughout the entire launch or radio networks which may or may not work at all, which is just simply too much of a risk to try and work with given the range of the project, the Bluetooth option seems to be the most reliable.

6.1.2 Mobile applications

The main component of the software will belong to the application that will gather the information from the launch. With the option of a touchscreen device, a mobile application would be developed for the launch pad in which a student will use both the touch screen and physical buttons to launch the rocket. When deciding with which platform to use the most obvious choices are between Apple and Android. In order to determine which would be a better choice to implement, there are certain conditions that should be met: First, the Integrated Design Environment should be consistent and easy to use. And second, the language used should be efficient and concise.

6.1.2.1 Android

Developing mobile applications for the android would allow for many more users to download the application as compared to Apple. When deciding which operating system to develop on the Android has one very big advantage in the design and that is the market for Android devices is much more competitive and therefore more cost efficient. With a better bargain and more popular market, Android has a very high chance of being chosen as the operating system used for development on the mobile application. Since the team has the most experience with Android, this is also a more efficient option in terms of learning curve, allowing for a quicker start on development. Android Studio is the development environment that would be used in order to create the mobile application, which is written in Java one of the most popular object-oriented languages. The experience with Android Studio as well as Java gives Android the edge over Apple.

6.1.2.2 iOS

When deciding if an Apple device is the right choice for this project, the mediums for obtaining an Apple device have to be taken in consideration. When looking at the price of tablets, the price difference between an Android and an iPad is very obvious. Since there is a budget for this project it is important to minimize cost wherever possible, by using a cheaper tablet the team is guaranteed more money to develop the electronic and mechanical components to a greater extent. An Apple mobile device was considered for donation to use from a member of the Space Trek team which would have made it a cheaper option than buying an Android device. With this option, the reduced cost and greater performance of the iPad would outweigh the learning curve time. The team would need to learn how to use Apple's development environment, Xcode, and without a device to test it on, an Emulator will be needed. Learning Swift could prove another delay in development due to a learning curve, which is yet another reason why developing for Android would save the team more time if not money.

6.2 Software Class Descriptions

Keeping software organized is one of the most challenges aspects of working in groups where multiple people will be modifying the code. Since there are many different types of software design approaches, a top-down design will be implemented. In this approach all of the requirements will be met by putting the most important features first, and allowing room for any additional features if time permits. This allows for a more dynamic flow and gives each member of the group the ability to work on an independent portion of the program and to then combine the work into one bigger program.

With a proper workflow the design for the rocket and the launchpad should be fairly straightforward. By using UML diagrams our team is able to make sure that all requirements are met and that a proper flow is maintained. Essentially by outlining the features of the program and giving a basis for how it will interact in its environment, each member of the group knows exactly which classes will be doing what. That way this minimizes confusion, duplicate work and unnecessary errors that would be otherwise caused by people trying to merge codes that are completely different.

In this design we have utilized several UML diagrams, such as the Class diagram, which represents each class as a separate entity and shows how it interact

6.2.1 UML Diagrams

UML diagrams are an excellent tool for both organizing and designing programs in a team setting. By utilizing a systematic flow, each member of the team will be able to know where they are within the design and know how their segment of code will interact with other member's. There are three classifications of UML diagrams:

- **Behavior diagrams** - A type of diagram that depicts the behavioral features of a system or business process. This includes activity, state machine, and use case diagrams as well as the four interaction diagrams.
- **Interaction diagrams** - A subset of behavior diagrams which emphasize object interactions. This includes communication, interaction overview, sequence, and timing diagrams.
- **Structure diagrams** - A type of diagram that depicts the elements of a specification that are irrespective of time. This includes class, composite structure, component, deployment, object, and package diagrams.

By utilizing diagrams across these classifications the team can create an elaborate outline for all procedures, timing and interactions within the software. The two main diagrams being used are sequence and class diagram. Sequence is an interaction diagram which shows not only who interacts with what but also the timing in which a function will run and when its results will be returned. Additionally the class diagram is irrespective of time but shows how each of the classes will interact with each other and the main member variables that will make up each class. Both of these diagrams help illustrate the flow and interaction within the program that will govern the rocket and launch pad MCU's.

6.2.1.1 Sequence

The sequence diagram helps to visualize the sequential order in which function will flow within the design. When a student is ready to manually enter in their data for their group's launch the instructor must have already set up the groups otherwise the flow will stop, because there needs to be a hierarchy to maintain in order to reduce the amount of bugs and other issues that can occur when information is accessed at random times. By keeping the flow organized the students will be able to enter in all the information they have calculated and then be able to compare them once the launch has been completed. If there was a way for a student to go around the access of the instructor then this could cause many problems for the mobile applications and will most likely crash the software.

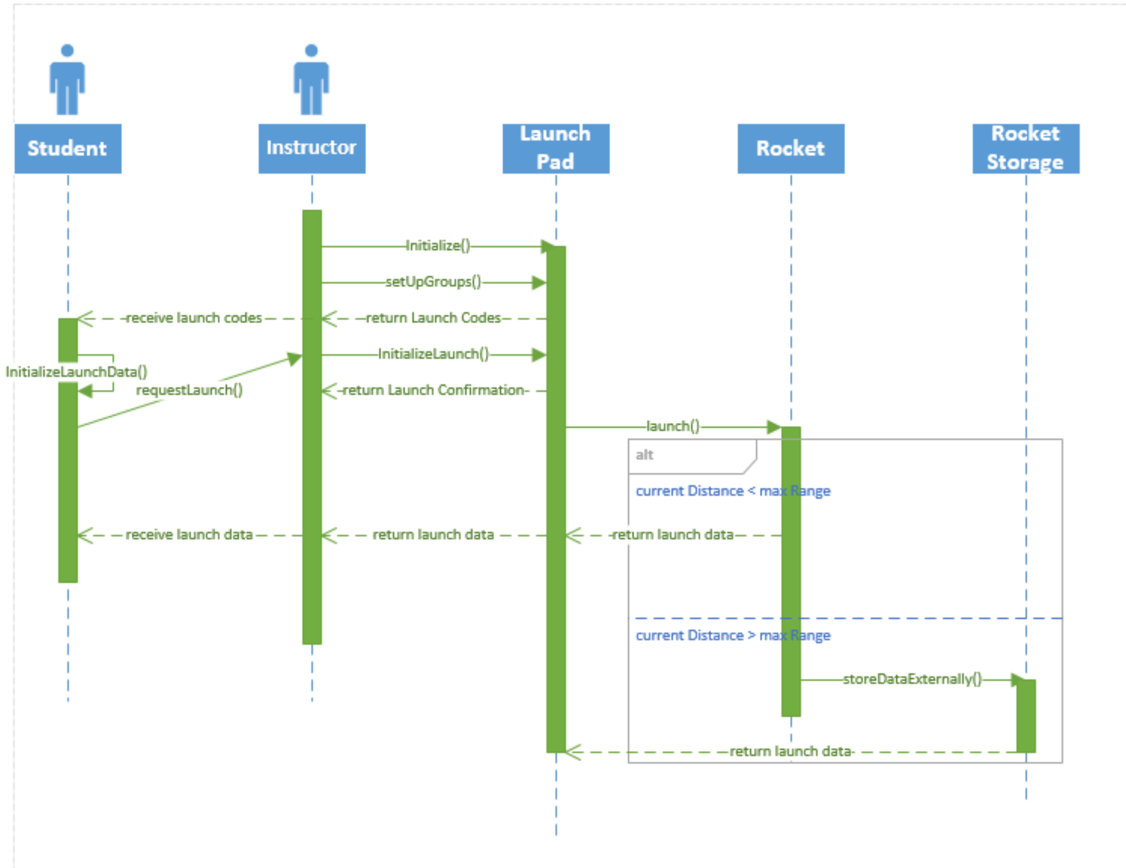


Diagram 6-1: Sequence Diagram of the Rocket Module

According to **Figure 6.1**, nothing can begin until the instructor has initiated the launch pad in preparation for a launch, once this has occurred, then each group must be initialized as well. After each group has been created, each student from the group may enter in their calculations for the last day of the launch in which they will apply their calculations to the actual launch. When launch day arrives and each group has made all their calculations, the students will be given a launch code for each group to use when performing the actual launch of their rocket. When a launch is about to be performed by the group the instructor must give it the go-ahead for the students to unlock their launch button and launch the rocket into the air. This is a failsafe method to use just in case the instructor notices something wrong with the rocket and decides they want to delay the launch to correct the problem.

Once the rocket has been launched, sensors onboard will capture telemetry data and send it back to the launch pad to be simplified into more understandable values. When each group has launched and received all of their data, then the instructors can review everyone's data and make graphical comparisons.

6.2.1.2 Sequence Revised

An updated version (see **Diagram 6-2**) of the sequence diagram for the software design has just been released, it includes an additional object lifeline and has another section of activation for the student and instructor. This means that there are two main phases that the students and instructors will be active in. First the initialization portion of the sequence which includes setting up the groups and distributing launch codes to all the students. This will be done by the instructor, once all the groups have been initialized, each individual group will have an opportunity to fill out all of the data they will need for their launch day within a specific section for their group. This is the first activation phase, the second activation phase comes when the launch is completed and the student wants to look at their results and compare them. Since this action can be performed at any time, it is separate from the initialization phase of the sequence diagram. The following is the updated Sequence diagram:

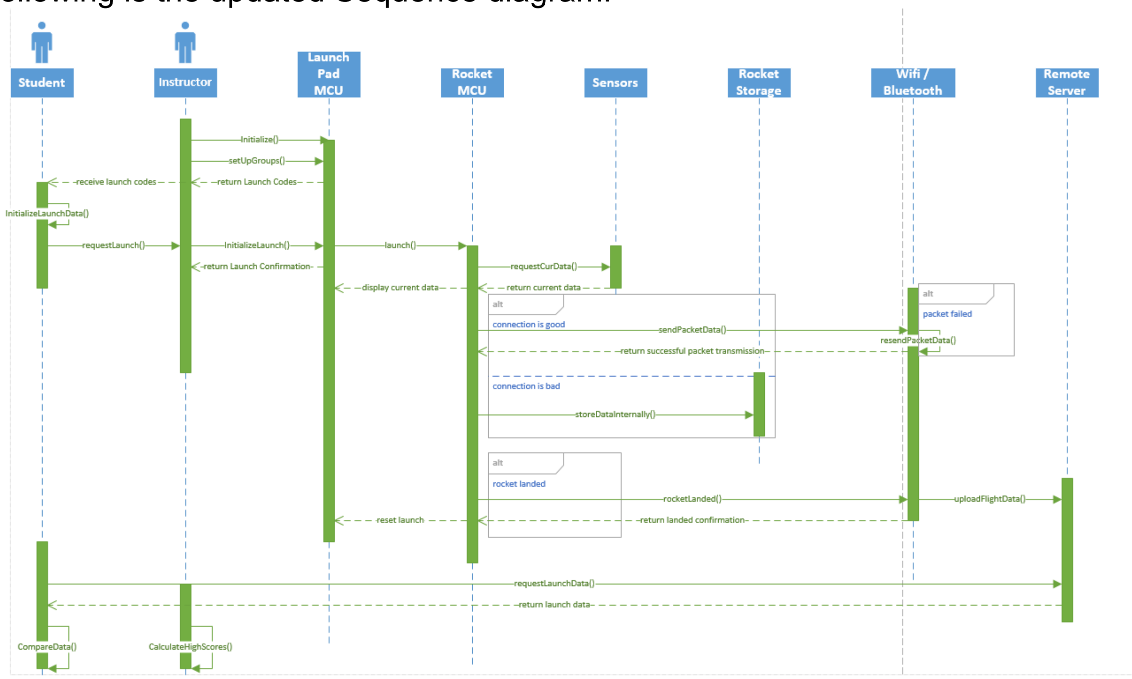


Diagram 6-2: Sequence Diagram (Revision 1)

The first major difference in this revision is that additional objects have been added: sensors, Wi-Fi / Bluetooth, and server. The communications module will have a direct connection between the rocket and the launch pad, giving it the ability to send data freely between the two hosts. When using this module, the data can be sent wirelessly to the launch pad for storage and later review. There are advantages to using Bluetooth over other means of wireless communications, such as Wi-Fi, but at the time of this revision the testing phase has not yet concluded which communication medium will work the best.

In addition to the Bluetooth module object, a new object has been added to represent the remote server for which the launch data can be stored once the rocket has landed. The condition has been modified for a more general connection based requirement instead of a Bluetooth pairing specific condition. With the new condition of a good connection being met, the data will be sent immediately back to the launch pad where it can be reviewed in nearly real-time.

Furthermore, new conditions have been created for the rocket MCU and the communications medium. If the packet fails to be sent back to the launchpad MCU then the packet will resend until it has successfully transmitted. The rocket MCU will also check periodically if the rocket has landed, an easy way to check would be a transmission of a height at zero. Once the rocket has confirmed it has landed, all of the flight data will be sent to the remote server, where the students can look up online the overall results of their launches.

Lastly when the data for all the launches has been accounted for, the instructor can run a report on the remote server to calculate the overall results for each team and find out who came closest to the goals. Each student can readily look at the data for their team's launch at any time and compare numerical results with another team, but once the instructor has analyzed the data and created a high score, each team can find out their ranking.

6.2.1.3 Class

The Class diagram is an important tool is software design in order to describe how classes will interact with each other and show the relationship between them. In the first revision of the Class Diagram there are many simplified and generalized classes because many decisions have thus far been speculative, giving way to a good idea of what the end product will be without an exact idea of how to get there. With that being said, the class diagram has shown that the program will require a tremendous amount of time and effort, not only to complete the tasks and reach the goal of the team, but to also last and outperform over time.

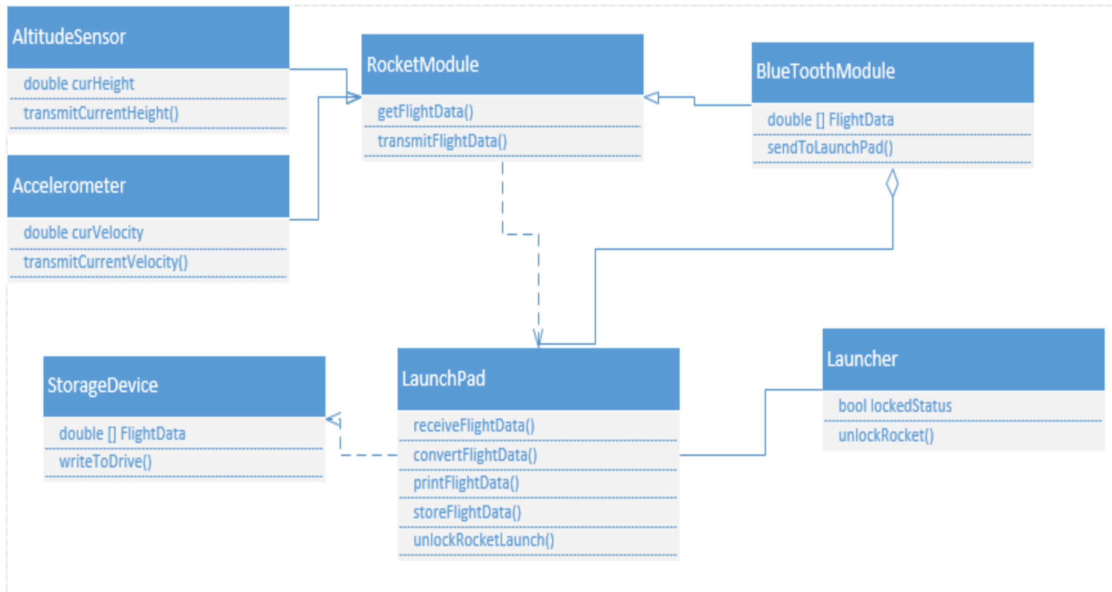


Diagram 6-3: Class Diagram of Rocket/Launchpad Module

With reliability in question, the class diagram gives a clearer picture as to which microcontroller units will need the most processing power in order to perform their required tasks. For instance, the rocket only needs to collect data from the sensors onboard and then transmit them to the launch pad. Meanwhile, the launch pad, has to take care of organizing all the groups and data, control the launch pad activation, then once it has received all the data from the rocket, it must convert the data and then store it in either a media device or some other server for comparison later. Therefore the launchpad MCU is going to need to be much more powerful than the once located within the rocket module because of how much more work it will need to do and how much data it must store.

According to **Diagram 6-3** first the rocket module must acquire all the flight data from each of its sensors, simplified to altitude and accelerometer (height and speed), we can then take this data and send it via Bluetooth, which acts as an aggregate to the rocket module class. Once all the data is send to the launch pad, the data is the stored and organized based on the time it was received. Since this is the first draft of the class diagram, there will be a lot more to elaborate on since the launch pad has many more functions than the rocket module has. In addition to what is labeled in figure 6.2 the launch pad will also require a launch code sequence in order to perform the unlock function, which can then only be approved by an instructor. Many of these additional functions and features will make their way into the diagram as the design becomes more refined and the components are chosen.

6.2.1.4 Class Revised

There have been some modifications and additions made to the first version of the class diagram for rocket and launch pad software design (see **Diagram 6-4**). The student and instructor classes have been added to show how they would interact with a remote server, as well as denoting a separation in storage between the rocket and the launchpad. Both will be able to store data and in worse case scenarios, all the data can be preserved on the rocket and extracted once it has landed. The following is the updated class diagram:

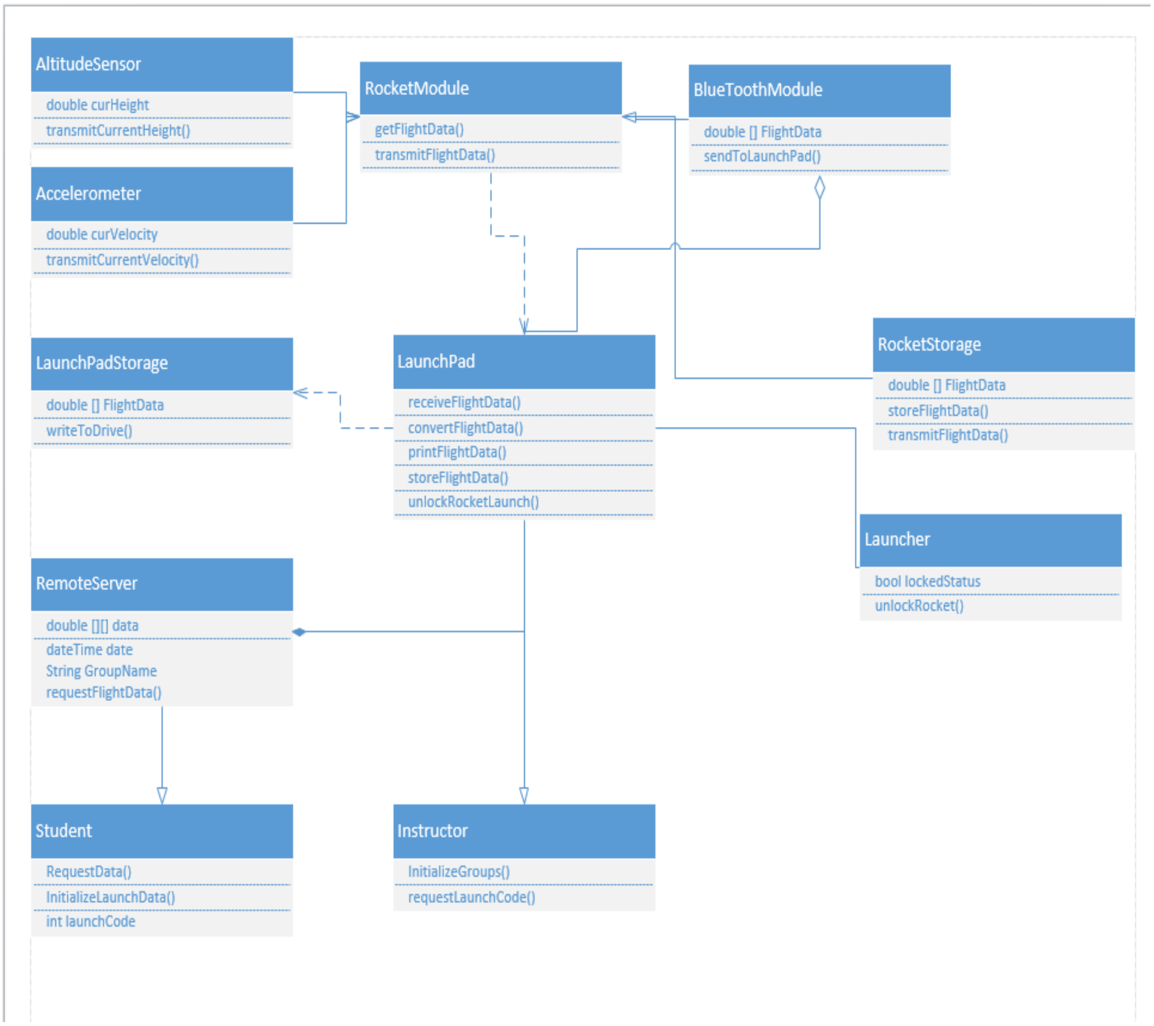


Diagram 6-4: Class Diagram (Revision 1)

Some noticeable additions are the remote server, student and instructor classes. With the addition of a remote server, the ability to save all the data to a remote location for secure storage, there needs to be a class that can control all of that data, hence the Remote Server class. With the addition of this class, there creates an opportunity for one class to handle all data that is transmitted from the rocket to the launch pad, by saving this data remotely, there is a decreased chance the data could be damaged or lost. With the addition of a remote server though, this increases the workload, timeframe and even the budget beyond the team's original projections. Nonetheless, the goal of this project can be reached without a remote server, so this goal is if time permits, then it will happen.

In addition the remote server class, a class for students and instructors have been added as well. Contrasting the class diagram to the sequence diagram, there is a common functionality for the students and instructors, which is to set up groups and to enter all of their calculated information into the application for later reviewing when launch time comes. This will help the students stay more organized as well as being able to automatically compare their calculated results with the actual results. The instructor has the responsibility of setting up groups for the students and once done, the program will automatically assign each group a launch code; which they can only use to unlock their rocket when it comes time to launch.

7. Hardware Design Details

The Smart Water Rocket system will use several custom designs to achieve the goals set by Space Trek. Each member of the team ensured that their design followed the constraints and specifications set. The decisions were discussed during the weekly meetings the team held to research and design the systems.

7.1 Hardware Designs

The team will have to design and modify several components for the SWR system. The launch pad must be modified to be controlled from the launch briefcase and the briefcase itself must be modified as well. The team will also be designing a new altimeter for the rocket to work with the computer system installed in the briefcase. All of these components must be compatible with each other or the system will fail to operate correctly.

7.1.1 Rocket Launchpad Designs

The design of the launchpad is important because it is a constraint of the project to make an automated launchpad but also because this is where students will be able to immediately be able to review their launch data. By creating an automated launchpad the students will be able to enter in their test data and have the rocket automatically configured to their specifications. The angle of the rocket will be adjusted according to the students as well as the volume of water and air. When the rocket has been filled accordingly then the students will be able to use the launchpad controls to send their water rocket into the sky.

While automating the launchpad is important, it is also crucial to have an impressive display on the launchpad that can make the experience that much better for the students. There are several different types of displays that can be used and most will achieve the aesthetic result that the team is looking for. Any tablet or multimedia device will be programmed with an application that will keep track of all the data for the students and their launches, which is discussed in depth in the software section. With a properly designed launch pad, the students will be able to automatically initialize and launch their rockets.

7.1.1.1. Existing Structure Modification

The launch pads are currently mounted to carts with two layers so that extra supplies (safety glasses, beakers, etc.) can be stored as well. The layers are separated by thick plastic with ventilation holes cut in the tip (see **Figure 7-1**). The sides are not enclosed, it looks like two tables stacked on top of each other on wheels. The carts work very well for this application and replacing them will add excess costs to the project so the carts will not be replaced.

The stepper motors will have to be mounted near each pad so the component that attaches the launch pad to the cart will have to be modified (see **Figure 7-2**). The electronics for the motors must be protected from any water that leaks from the rocket (or is expelled after launch) and easy to access if they require maintenance. The microcontroller will be stored in a water resistant container underneath the launch pads. The motors will be mounted upside-down, beneath each launch pad, and any wiring will be protected by heat shrink wrap. A piece of acrylic will be cut to fit around each launch pad and provide a water resistant layer between the pad and the electronics below. Pictures of the current setup can be seen in the figures below.

Figure 7-1: Photo of the launch pad cart currently in use by Space Trek. Photo taken by SWR team.

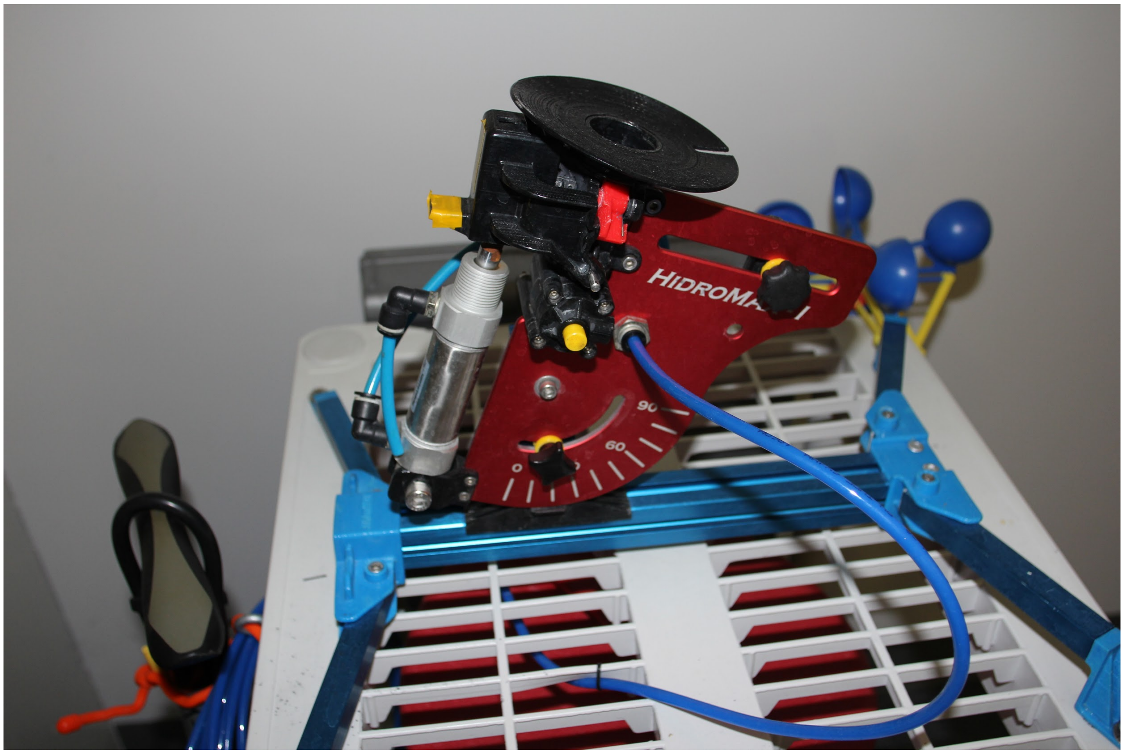


Figure 7-2: Close up of the launch pad for the water rockets. The nozzle of the bottle is inserted into the hole at the top of the black disk. Photo taken by SWR team.

The fuel and pressurization automation modifications will be more in depth. The nozzle of the rocket is locked into the pad before pressurization. For the new SWR design, a tube must be fed through the base of the pad and feed into the nozzle of the rocket so that fuel and pressure can be dispersed. The rocket is currently loaded into a socket on the pad nozzle first. The team intends to drill a small hole in the side of the socket where the rocket is loaded and feed a tube through the hole. After the modification, when the rocket is loaded the hose will fit perfectly into the nozzle of the rocket. It is imperative that this is a closed system. If it is not or if it has any major leaks the compressor will not be able to fill it with fuel or pressurize the rocket. Silicon or vinyl sealant will stick to both the tube and metal to improve the seals and decrease pressure loss to the surroundings. The tube that feeds into the nozzle will have to accommodate three different sizes so it must be conical and taper to a smaller tip. The smaller nozzles will sit higher on the tube and still retain a seal. **Figure 7-3** shows how the tube will fit into the nozzle of the rocket regardless of nozzle diameter.

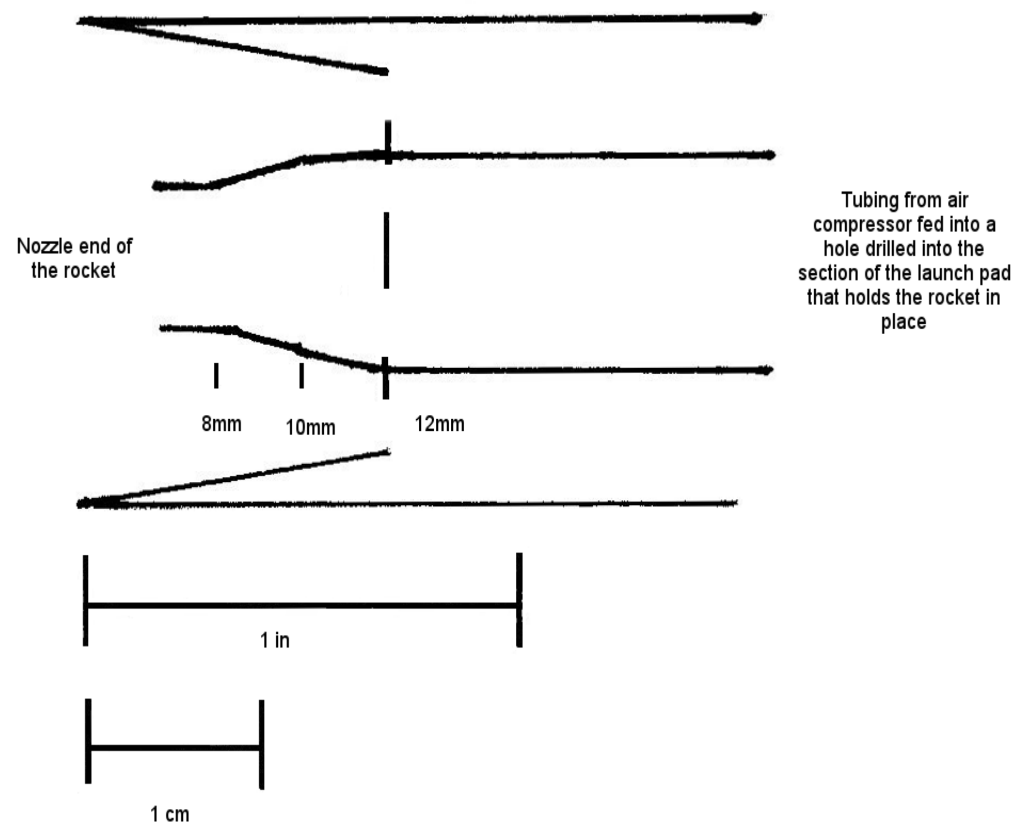


Figure 7-3: The tubing from the air compressor will taper down to the smallest nozzle size (8 mm) to accommodate all three nozzle sizes. Drawing by the SWR team.

7.1.1.2. Motor Mounting Design

There are two possible designs for the motor mount; piston based and belt based. The piston based design will connect the stepper motor and launch pad by a fixed arm. The belt based design will connect the motor and launch pad by a belt. Both designs require the motors to be in close proximity to the launch pad because they are directly connected. The most ideal place to mount each motor is underneath its corresponding launch pad because the cart will provide water resistance and protection to the motor. Additionally, the cart can easily be modified to allot for any components necessary to connect the motor and pad.

The belt based design will use a belt to control the angle of launch. The belt will be attached to the stepper motor and to a spoked wheel attached to the launch pad; as the belt turns the spoked wheel will catch it and rotate the launch pad. This design is ideal because it will be simple to integrate into the current system.

There are only three new components to install and none of them require extensive mechanical engineering knowledge. It will be easy to program the motor to precisely control the angle; one direction will increase it and one will decrease it by precisely one degree per step. The goal of automating the vertical angle of the launch pad is to allow the students to move the rocket to any degree of their choosing with a high level of precision and this design is the simplest and most efficient way to achieve that goal.

The piston based design will require an arm to be attached to the spinning section of the motor and near points (a) and (b) in **Figure 7-4**. The motor and arm must connect in such a way that when the motor turns one direction the angle of launch increases (approaches 90 degrees) and decreases (approaches 0 degrees) when turning the other way. The piston arm must also be lightweight so that it does not add extra stress to the motor. This design is complex and will require more mechanical engineering knowledge than the team possesses. In the event that the belt based design completely fails the team has a backup option that can be further developed into a working prototype.

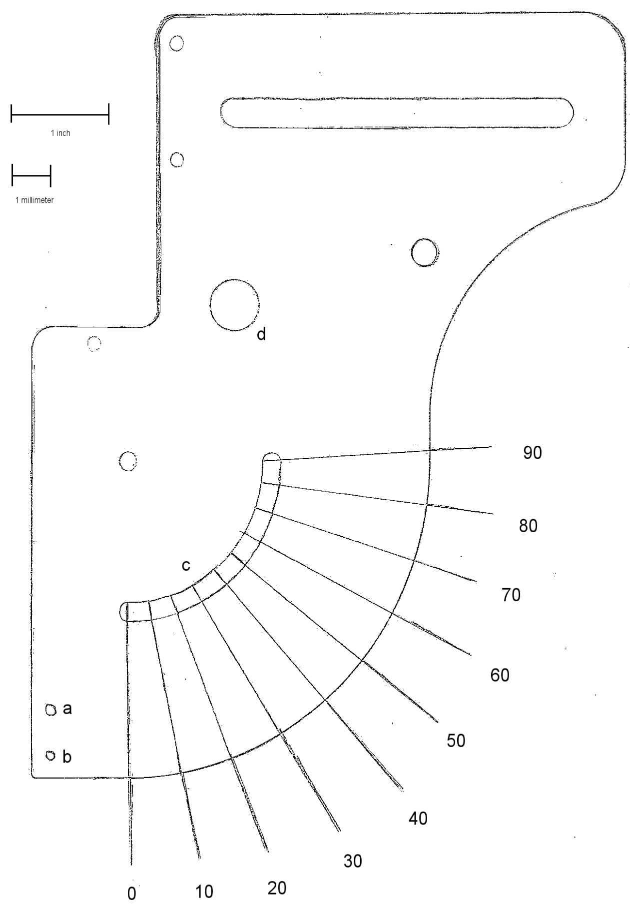


Figure 7-4: Line drawing of angle plate in use by Space Trek. The SWR will adapt the current hardware to work with the new automation designs. Drawing by the SWR team.

7.1.1.3 Fuel and Pressurization Flow

The air compressor will be responsible for fueling and pressurizing the rocket. The fuel for the rocket will come from a pressurized vessel acting as a keg. The keg will hold enough water for one session of launches. The optimal fuel amount in the rocket is 23% or 460 milliliters. There will not be more than 20 groups at a time, assuming each group will perform two launches the vessel will need to hold 18.4 liters of water. To simplify the vessel selection process this value will be approximated to be 19 liters or 5 gallons.

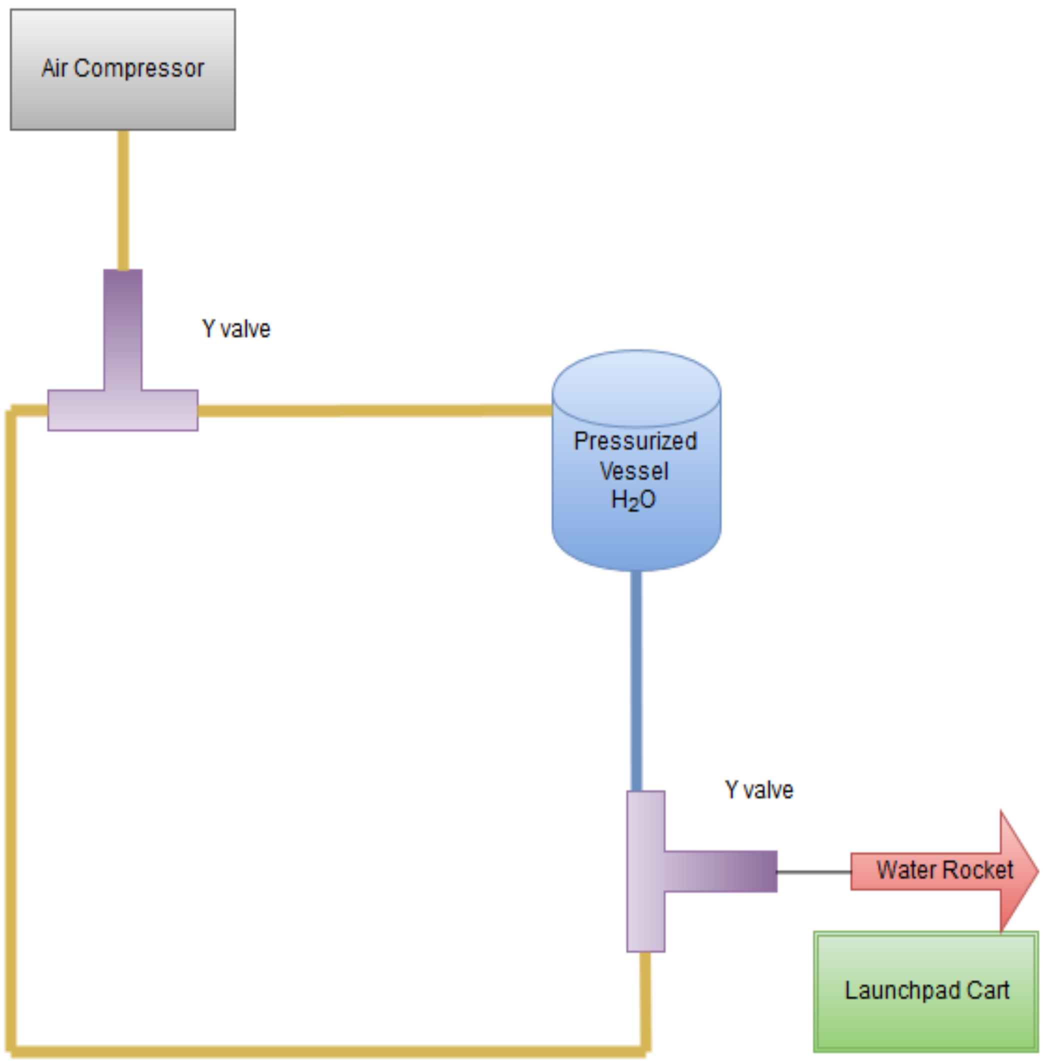


Diagram 7-1: Block diagram of the air flow in the pressurization/fueling system. The yellow lines represent air flow towards the rocket and the blue line represents water flow. The black line from the second valve to the rocket

represents the dual purpose line that will carry both air and water. The launch pad card has no relation to the system besides holding the water rocket and is purely symbolic.

From **Diagram 7-1**, starting at the air compressor, air will flow into a Y valve controlled by a switch. When the switch is in position 0 it will allow air to flow directly into the rocket and pressurize the chamber, when it is in position 1 it will direct air into the keg. The keg will have a check valve on the input and output to prevent either air or water from flowing backwards into the compressor or backflow. A sensor will be used to track how much water has been dispersed. A pressure sensor could be used at the keg or launch pad to calculate changes in the weight of the water or a sensor that detects flow could be used at the output valve of the keg. Both the water and air lines will meet back up at a second Y valve. A pressure gage will be used at either the valve or the launch pad to determine the current pressure in the rocket. The tank may become partially pressurized during fueling by pressurized air bleeding through into the tubing. There will be check valves on both of the inputs to the Y valve and on the output to the rocket to prevent backflow. The amount of fuel and pressure dispersed will be controlled with precision by the launch briefcase and the students will be able to input the exact amounts of each that they want.

7.1.1.4 Waterproofing the Electronics

When working with the Smart Water Rocket there are several components of the system that require protection from exposure to water. An important part of the SWR that needs water protection is the altimeter, which will be contained in the rocket. The rocket's lower half bottle is pressurized with water and air up to 100 PSI making it a volatile capsule that could potentially leak into the upper half of the rocket where the altimeter is located. If the altimeter is not sealed off from water or protected the team would have to reorder parts and components and build a new one from scratch every time the circuit shorts. Another part of the system that requires protection is the automated launchpad. To change the system from a manually operated launchpad to an automated one the team will include a microcontroller connected to stepper motors that will fix onto the platforms angle plate. From this setup the motors and microcontroller will be located close to the rocket's thrust and will thus have to be able to withstand some water saturation.

To protect the altimeter PCB and microcontroller the most common method of water resistant protection is referred to as conformal coating. This is a thin polymeric film material that 'conforms' to the textures of the printed circuit board, which protects the components. PCBs are coated using one of three methods: brushed, sprayed, or dipped. Brush coating is typically used for low volume application, finishing and repair. This technique is the least used because unless a skilled worker performs the task, defects such as bubbles and thicker parts of the film could result. Spray applications are commonly found in aerosol canister form but can also be issued with a spray gun by a professional. This technique is best for low and medium volume coatings. If a skilled professional uses the spraying technique on a PCB that is clean this method can be the best choice. The last method of application is the dipping technique. Typical applied in repetition this method can be the highest volume and is superb at covering every tight space on the circuit board because of submerging in the material. If the PCB is not properly masked however this could create a problem is certain designs.

For waterproofing stepper motors the most common method is known as potting. This process involves filling the internal components with a gelatinous silicone substance that is not only water and corrosive resistant, but adds shock and vibration absorption as well. Once the internal components of the motor are filled with silicone the housing for the motors can be sealed with epoxy to prevent any moisture from seeping in through the cracks.

7.1.1.5 IPX Certifications

The Ingress Protection Marking or IP Code classifies the rates and degree of protection from bodily intrusion, dust, accidental contact, and water. The team is interested in the liquid ingress protection ratings in particular.

Based on first-hand experience observing current launches, the ideal IP ratings that this device will be trying to attain, would be IPX4 to IPX6. This will not only be the case for the altimeter, but also the launchpad. By making this device resistant to water jets, like those of the rockets, we can assure our sponsors at Space Trek of the durability and reliability of our final system.

Level	Protected against	Effective against	Details
0	Not Protected	-	-
1	Falling Water	Vertically falling water	**Test duration:** 10 minutes **Water** ~1mm rainfall/minute
2	Falling water 15°	Vertically falling water up to 15°	**Test duration:** 10 minutes **Water** ~3mm rainfall/minute
3	Spraying water	Falling water in spray form 60°	**Test duration:** 5 minutes **Water volume:** 0.7 litres/minute **Pressure:** 50-150 kPA
4	Splashing water	Splashing water from all directions	**Test duration:** 5 minutes **Water volume:** 10 litres/minute **Pressure:** 50-150 kPa
5	Water jets	Water projected from any direction by a nozzle	**Test duration:** at least 3 minutes **Water volume:** 12.5 litres/minute **Pressure:** 50-150 kPa
6	Powerful water jets	Water projected any direction by a powerful nozzle	**Test duration:** at least 3 minutes **Water volume:** 100 litres/minute **Pressure:** 100 kPa at 3 m
6K	Powerful jets with increased pressure	Water projected from powerful increased pressure	**Test duration:** at least 3 minutes **Water volume:** 75 litres/minute **Pressure:** 1000 kPa at 3 m
7	Immersion up to 1m	Ingress of water in up 1 m of submersion	**Test duration:** 30 minutes
8	Immersion beyond 1m	Protected beyond 1 m if conditions met.	**Test duration:** continuous Depth up to 3 m approximately
9K	Powerful high temperature water jets	Protected against close-range high pressure, high temperature sprays	**Test duration:** - **Water volume:** 14-15 litres/minute **Pressure:** (8000-10000 kPA/80-100 Bar) at 0.1-0.15m **Water temperature:** 80□

Table 7-1: IP ratings for liquid ingress protection

7.1.2 Briefcase Design

When designing the briefcase there will be several methods for implementing the button configurations as well as many options for selecting a display. The exact button configuration is not set in stone because the team has not had the chance to acquire parts and actually work with a tangible product. To truly decide on a configuration there must be multiple prototyping stages in which the team can get a hands on feel for how the launch sequence would best be received by the space program attendants as well as our project presentation reviewers. The tablet display is also another later stage consideration as the finished product may have a different tablet outside of our budget once the system is tested and working. For all intents and purposes the SWR team will test and program with the same OS and the device that will be purchased later on.

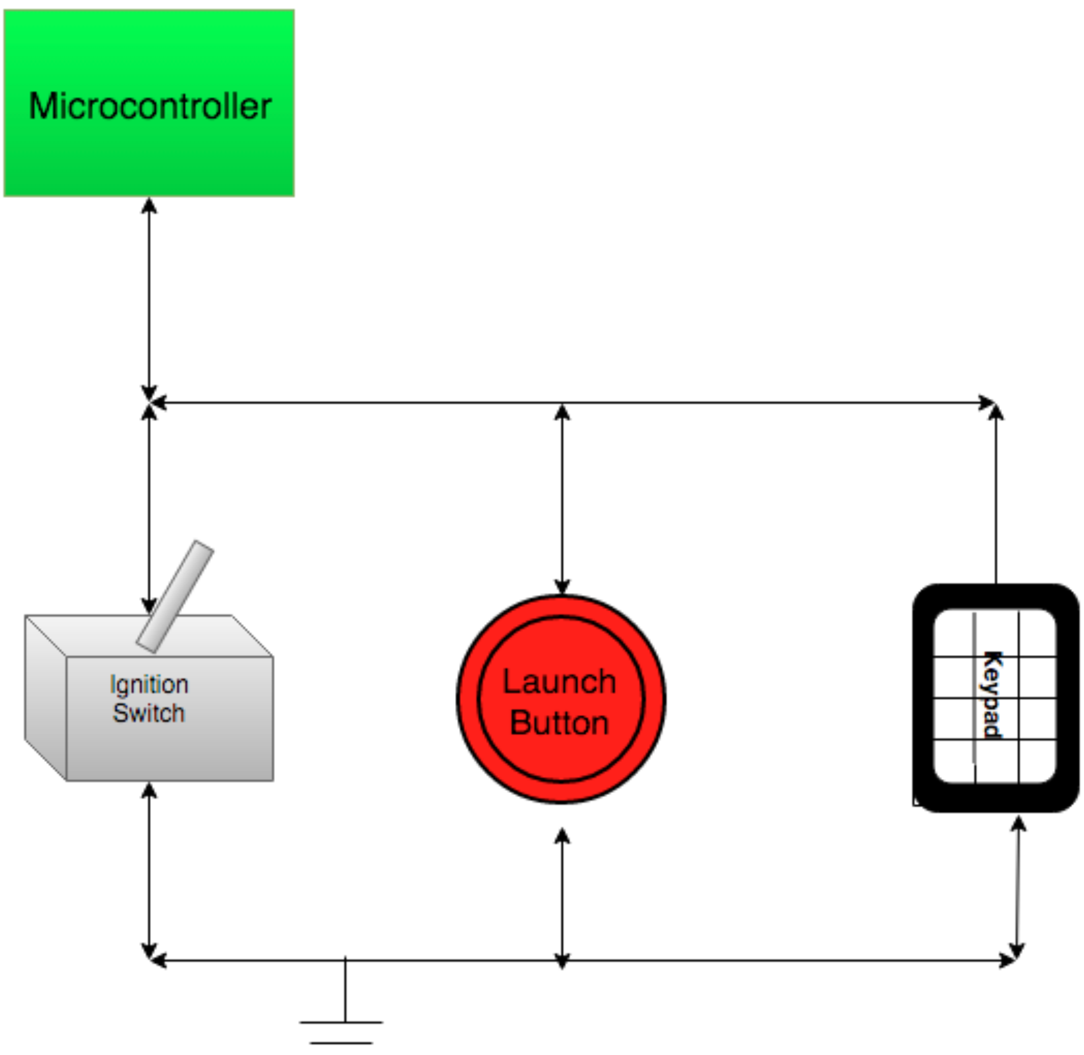


Diagram 7-2: Block diagram / rough schematic of briefcase design

7.1.2.1 Power

To power the briefcase we will be initially using a USB interface with the provided Lenovo All in One desktop provided by our sponsors. This USB interface should work to transfer and read all of the data coming in or out of our onboard microcontroller which will be controlling all of the mechanical systems. Also, the USB will provide constant power to our microcontroller.

After the team is finished with all of the other subsystems and the entire communications system is in place a solar panel charging interface will be implemented. This is ideal considering that the device should be mobile and will be mostly used outside. If we can effectively power all of our mechanical components with a solar power source the team will have effectively designed a self-sufficient Smart Water Rocket system.

The solar panel is intended to charge a lithium battery that will be on-board the briefcase. This battery will most likely be a 9V lithium-ion rechargeable battery, which is more than enough to power our components. The goal is to have the battery charge fully and after that, the typical two to three hours spent using these rockets should go by and still have about half the battery charged.

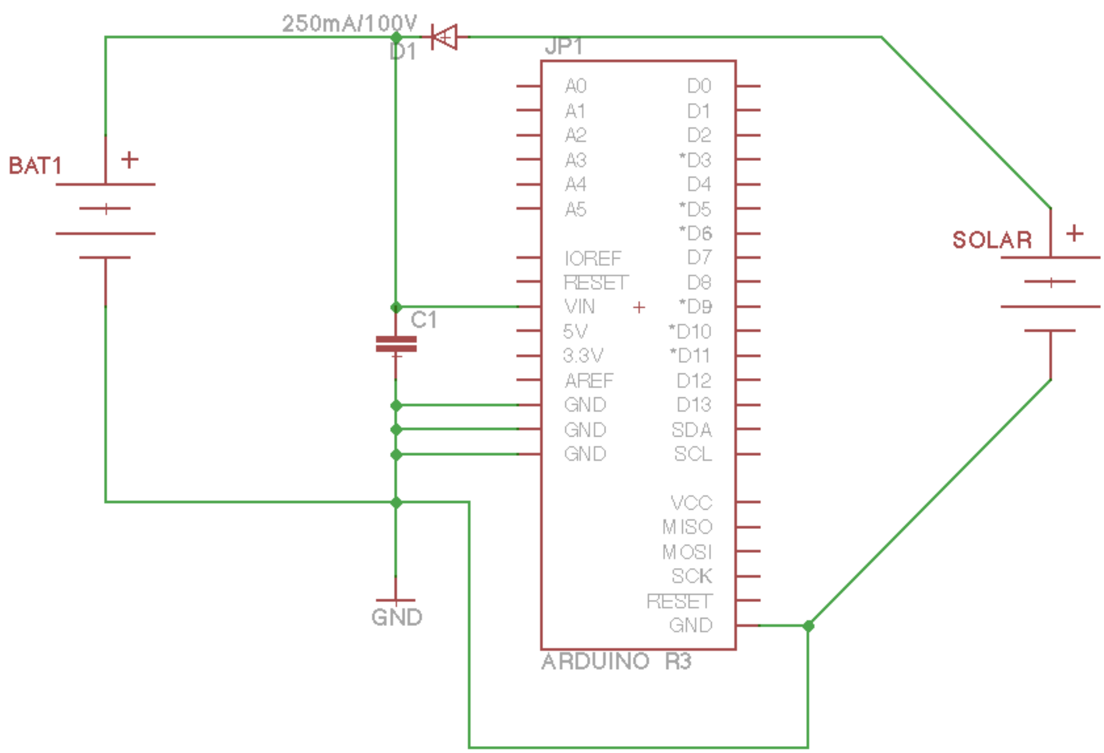


Figure 7-5: Schematic of power system for the Briefcase subsystem.

The schematic above (Figure 7-X) details the mentioned system's schematic. The diode in this design is to prevent any current flow to the solar cell. Sending current to the Solar panels could cause electrical failure and this should be avoided at all costs. Another issue that will arise in the future is the fact that there is no way to prevent the solar cell from overcharging the battery. To fix this a solar panel charging control will have to be added or designed, but an Eagle compatible library that included such a device couldn't be found at the time of this writing. The capacitor in the circuit is intended to be used for stability purposes.

7.1.3. Altimeter Design

Based on the research conducted (see sections 3.2.3.1 and 3.2.3), designing the altimeter will benefit from low weight and low power components. The final altimeter design will most likely go through various iterations before all of the design parameters are satisfied.

7.1.3.1 Microcontroller

To handle all of the data within the altimeter an Arduino Uno Revision 3 is going to be implemented for the initial prototype. This board is based on the ATmega328p which has become on of the most popular MCUs to work with and provides all the basic functionality the altimeter needs. It will be used to communicate with the accelerometer and pressure sensor to obtain the desired data and then relay the information to the “briefcase” by Bluetooth.

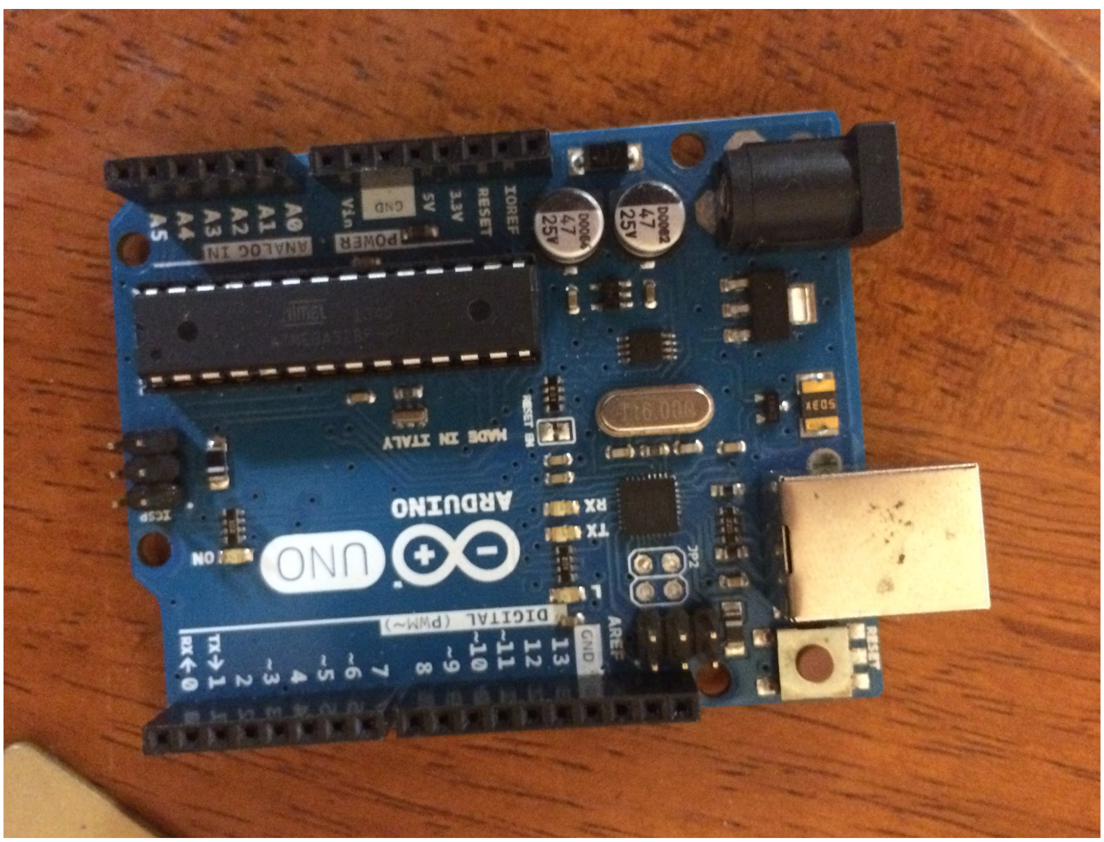


Figure 7-6: Photo of an Arduino Uno by the SWR team.

Using an Arduino Uno Rev3 provides an excellent platform to work with and obtain compatible parts for. The communication between the components will be handled with I2C standards and each component is intended to work with 3.3V. This should provide easy integration with multiple components.

7.1.3.2 Pressure Sensor

The altimeter will feature a BMP180 pressure sensor to effectively calculate the altitude of the rocket and communicate with the Arduino Uno Rev 3 (ATmega328p) over I2C. This device was chosen for it's price, I2C compatibility and voltage requirements. Also, when compared to the MPL3115A2 the BMP180 offers a smaller package which should make it ideal to keep our PCB square footage down and in turn, the price down.

Bosch's BMP180 is also one of the most widely available pressure sensors online and the datasheet provided by Bosch was extremely more detailed than Freescale's MPL3115A2. This highly detailed information is ideal for the inclusion of this device on our PCB.

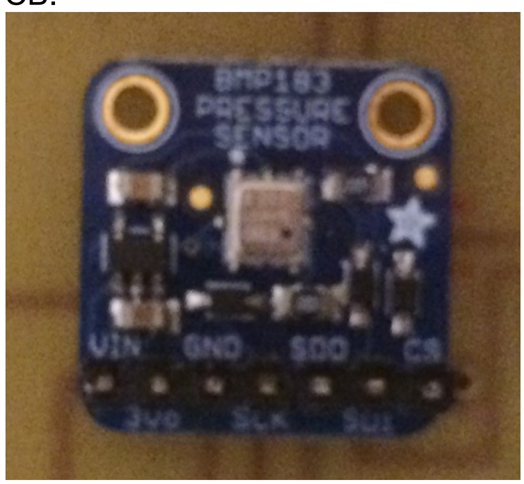


Figure 7-7: Picture of a BMP183 module, a variant of the BMP180 with the same specifications, provided by Space Trek. The photo was taken by the SWR team.

Currently an Adafruit BMP180 breakout board, pictured above, is available from our sponsors at Space Trek and will be used for initial testing. This breakout board will be implemented to our first prototype and used in conjunction with the Arduino Uno Rev3 board.

7.1.3.3 Accelerometer

For the accelerometer, the altimeter will use the ADXL345 by Analog Devices. Although this wasn't the most power efficient model or the smallest (see section 3.2.3.2.3), it is the most cost effective option as well as a very popular option with lots of resources available unlike the LIS3DH.

By implementing this triple-axis accelerometer the altimeter will be able to take advantage of automatic free fall detection as well as activity and inactivity detection. These features will work effectively with a smart water rocket system where speed and altitude are our main goals. By utilizing those features the altimeter will be able to save processing time from the MCU and lower power consumption by turning off other systems during free fall.

As with any accelerometer, the purpose of this implementation is to effectively obtain the changes in acceleration and determine the object's velocity as well as potentially determine its position. As previously stated (see section 3.2.3.2), the use of 3D positioning is entirely possible with a triple-axis accelerometer.

7.1.3.4 Design Constraints and Parameters

Just like any project, our altimeter system needs to satisfy certain parameters. As is the case with a good majority of projects that require objects to stay off the ground, weight is a main concern when updating the current system in place. At the time of this writing, the altimeter used by Space Trek weighs in at 9.9g. This weight requirement is currently the single most important constraint when designing the altimeter.

Given the scope of the system and the required components within the altimeter, 9.9g will prove to be a design challenge. Currently, the initial altimeter will be over twice the weight limit since the initial prototype will be used to work on communications, programming and prove of concept (see section 8 for more details).

Another main constraint is the Bluetooth module. Given the fact that we're dealing with a rocket, long range communications via Bluetooth could prove very difficult. Considering that we need to keep our altimeter under 9.9g we can't increase our range with an onboard antenna without significantly increasing the weight. This leaves us with the option of implementing a small module within the rocket and using a very intensive Bluetooth component on the other end.

Once we have a working prototype, the next step will be to reduce the weight to as little as possible. To achieve this, a custom PCB will be used instead of breakout boards, which should potentially decrease the majority of the weight. To do this we will be using all of the parts mentioned above and mount them to a custom PCB.

7.1.3.5 Design Schematics

Based on the selected parts, various schematics were designed throughout the research process. What initially started as a block diagram began to be substituted by the selected parts and slowly started to take the form of a schematic, but still kept a block diagram essence to it (see **Diagram 7-3**).

Essentially once the I2C standard was set and we had the two sensors the diagram was sketched including a space for a future Bluetooth module.

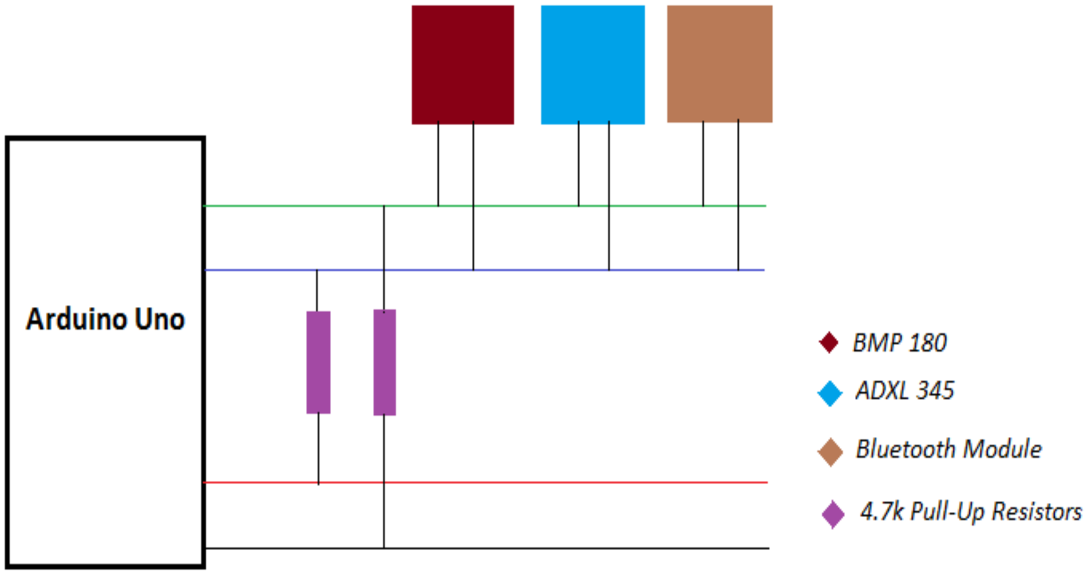


Diagram 7-3: Block diagram of the SWR altimeter design

Realizing that more in depth schematics were needed, each component was designed individually. The reason for an approach like this one was to keep the schematics as small as possible and carefully analyze each component before it's implemented in a solderless breadboard. This decision was reached due to the fact that breakout boards will be used to prototype our first altimeter.

The schematics were designed using Eagle, a popular design tool that's also available for free. Unlike previous projects worked by the team, no one had needed to use Eagle and in turn, no one knew how to properly use it. After various videos, guides, and a somewhat steep learning curve, the team was ready to design.

In order to design each component separately, every single datasheet was thoroughly analyzed and the required Eagle libraries were downloaded. Using some reference designs provided by the manufacturers, a schematic was obtained for every sensor being implemented in the final altimeter.

The first, and possibly the hardest choice of the components to be selected, was the BMP180. This pressure sensor is both widely available as well as extensively used in DIY projects. Not to mention, this sensor is currently available from the project Sponsors at no cost. This reason alone biased the decision to use the BMP180 over the clearly more efficient BMP280. To effectively integrate it to our system and develop the schematic below (see Figure 7-8), the datasheet reference design was combined with the breakout board schematic from Adafruit.

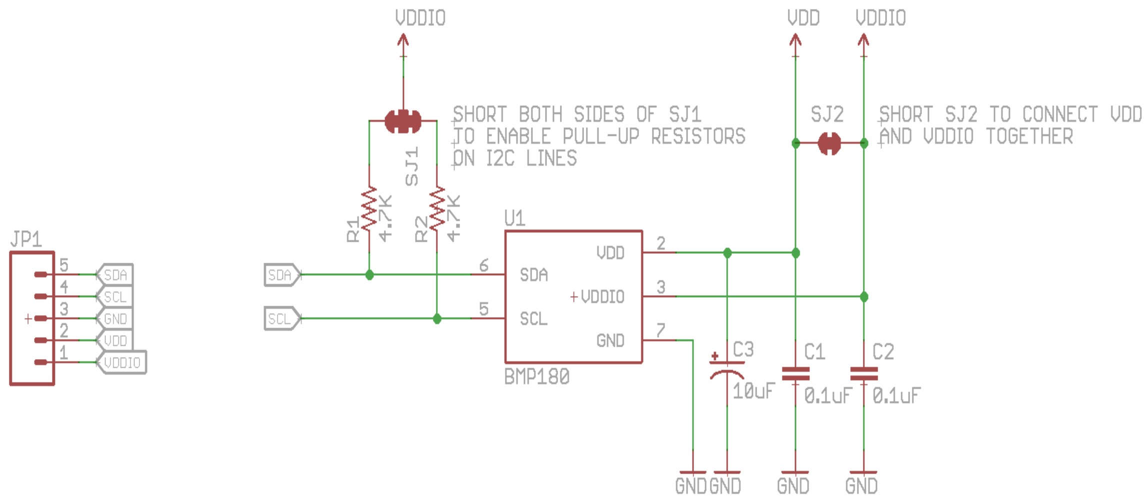


Figure 7-8: Schematic detailing how the BMP180 will be integrated into the SWR altimeter.

The next sensor chosen, the ADXL345, was designed using the same approach (see **Figure 7-9**). By looking at the datasheet and utilizing the reference design, the schematic below was designed. Unlike with the BMP180 the pull up resistors were left out since the idea was to combine both sensor's SDA and SCL in the same bus going to the ATMega328p. Other common practices like capacitors to maintain a stable voltage were implemented in both designs.

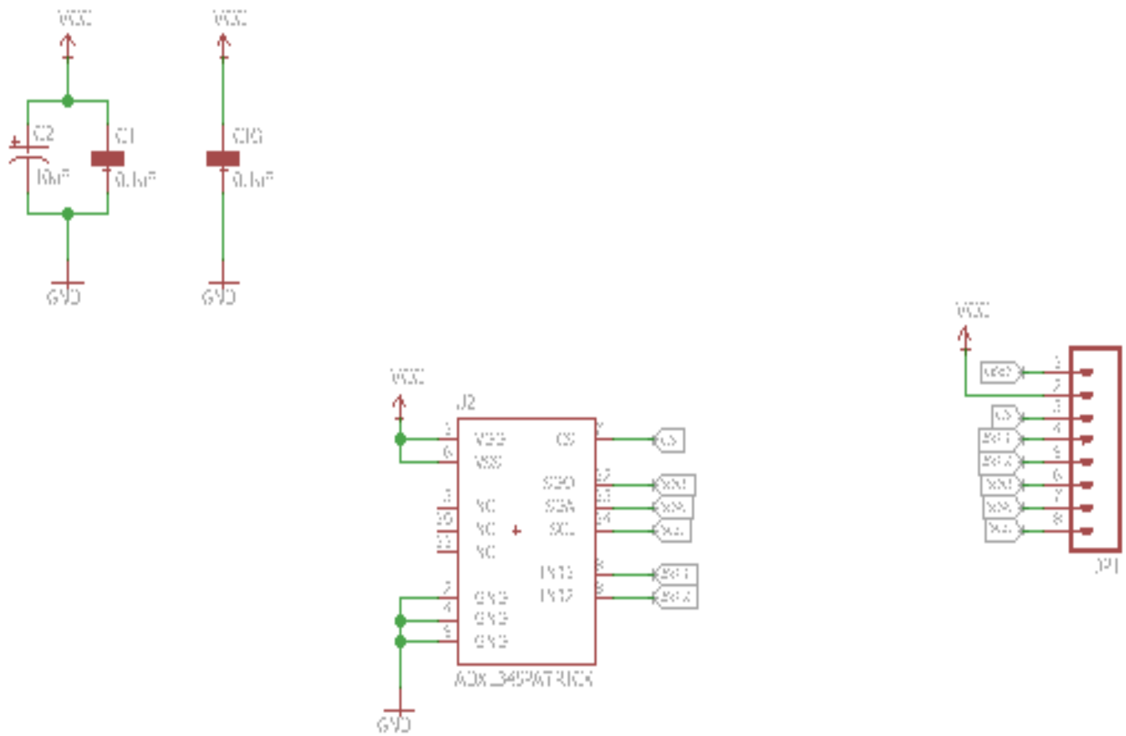


Figure 7-9: Schematic detailing how the ADXL345 will be integrated into the SWR altimeter.

Once the pressure sensor and accelerometer schematics were designed, the next step was to combine it with the MCU. In this case we used the ATmega328p since it's the board on which the Arduino Uno is based on. Doing this we obtained the schematic below, detailing some of the connections between the devices as well as which pins on the ATmega328P need to be used for I2C communications.

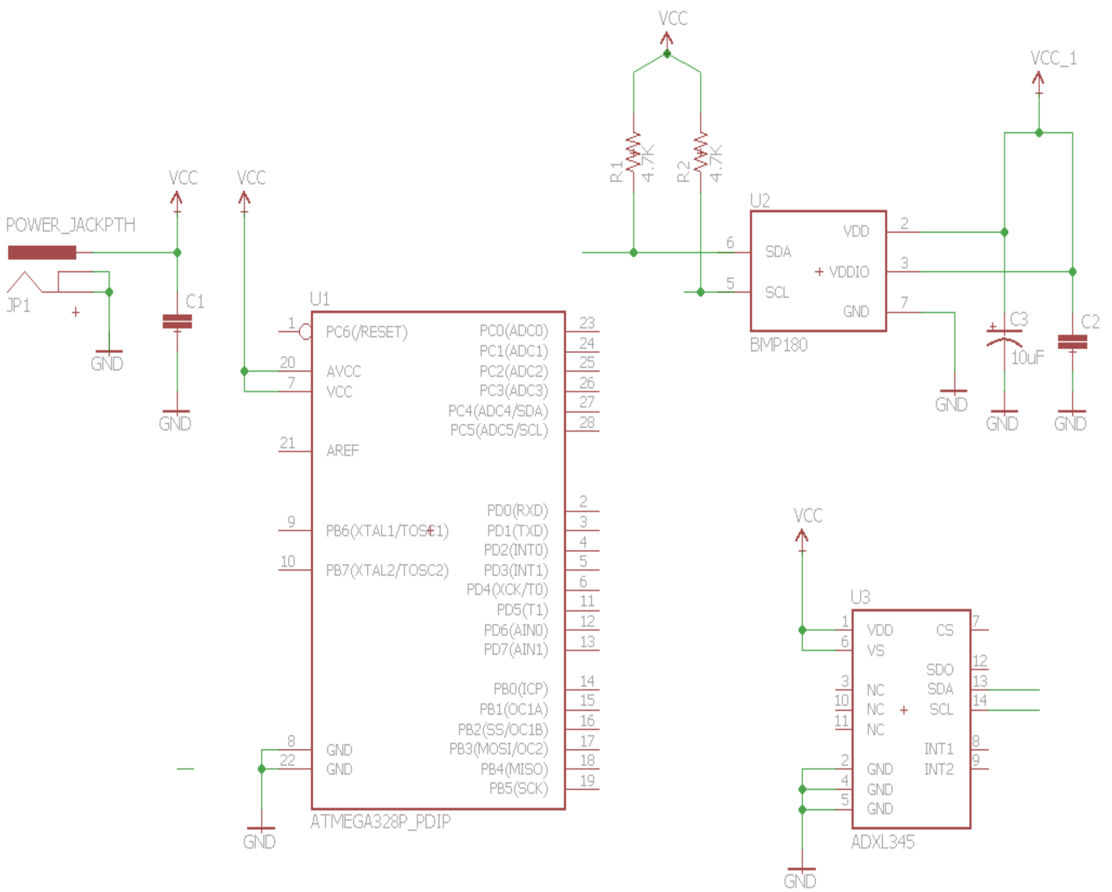


Figure 7-10: Schematic showing the connections between the ATMega328P of the Arduino and how they will connect to the other sensors in the altimeter.

This schematic (see **Figure 7-10**) is by no means the final design, but it provides a clear plan to combine the breakout boards to our MCU for our first prototype. Based on this schematic it's clear we are missing important components like a proper battery source, not a DC power jack, and a step-down regulator fixed at 3.3V to power our sensors.

From the data obtained on step-down voltage regulators (see Section 3.2.4) a LM2576 will be implemented to supply adequate voltages to both the pressure sensor and the triple-axis accelerometer. This device will be implemented following the input voltage in parallel with a capacitor (see the picture below), with the output supplying the selected component; in this case two will be used, one for the pressure sensor and one for the triple-axis accelerometer.

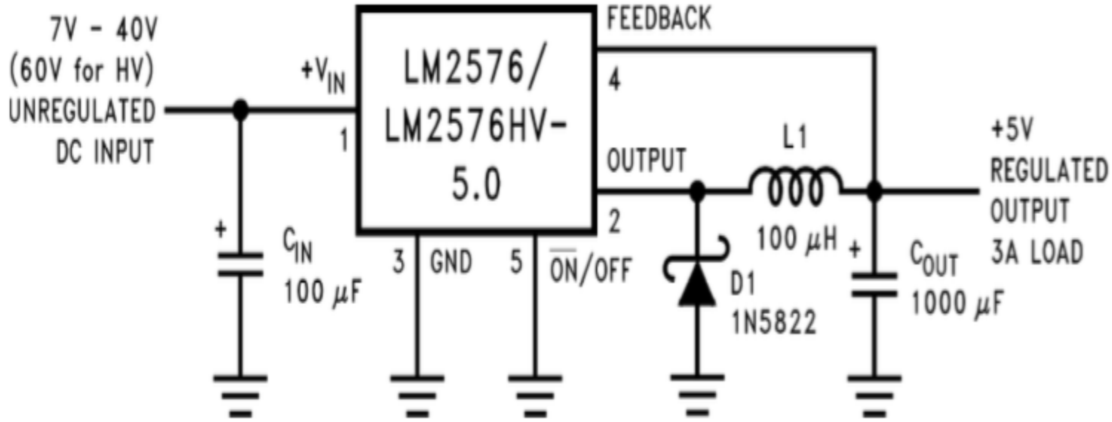


Figure 7-11: Picture obtained from TI's datasheet

Texas Instrument's data sheet includes a reference schematic for a typical application of the LM2576. Unlike their design, the altimeter's version of the LM2576 will be fixed at 3.3V instead of the shown 5V.

7.1.3.6 Power

For the altimeter's power supply, a battery will be used for the final design. Initially the device will be powered around the 9V to 5V range by the Arduino's DC power supply or USB interface. After the early stages of our prototyping, we will move to a lithium-ion rechargeable battery to power the entire device. This will allow for a finished product that should never have to be opened.

The lithium-ion battery approach was selected over the coin-battery based on its benefit of reusability given the fact that the team will be constantly utilizing the device. If a situation were to occur were battery runs out, it will be far easier to recharge the battery and in the current stages, where power draw is not efficient, such a situation could happen often.

The lithium-ion battery will most likely be obtained at the 3.6V range since it offers the target voltage for our system, as well as ease of acquisition. Currently at the largest online electronic component retailers the most commonly used lithium-ion battery is the 3.6V one.

7.1.3.7 Bluetooth

The altimeter will include Bluetooth connection (see **Figure 7-12**) in order to communicate with the briefcase and transmit data. There are many options to establish a Bluetooth link but there is one in particular that has a lot of resources as far as implementation. It is also the most commercially available and a cheap option.

The HC-05 module (see **Table 7-2**) is a simple Bluetooth Serial Port Protocol module designed for wireless serial connection setup. HC-05 has Bluetooth v2.0 + EDR (Enhanced Data Rate) 3Mbps data transmission sent over a 2.4GHz radio transceiver and baseband.

Hardware features

- -80dBm sensitivity
- Up to +4dBm RF transmit power
- Low Power 1.8V Operation, 1.8 to 3.6V I/O
- UART interface with programmable baud rate
- With integrated antenna
- With edge connector

Software features

- Defaults Baud rate: 38400, Data bits: 8, Stop bit: 1, Parity: No parity, Data control: has. Supported baud rate: 9600, 19200, 38400, 57600, 115200, 230400, 460800.
- Given a rising pulse in PIO0, device will be disconnected.
- Status instruction port PIO1: low-disconnected, high-connected.
- PIO10 and PIO11 can be connected to red and blue led separately. When master and slave are paired, red and blue led blinks 1time/2s in interval, while disconnected only blue led blinks 2times/s.
- Auto-connect to the last device on power as default.
- Permit pairing device to connect as default.
- Auto-pairing PINCODE:”0000” as default
- Auto-reconnect in 30 min when disconnected from range.

PIN Name	PIN #	Pad type	Description
GND	13, 21, 22	VSS	Ground pot
3.3 Vcc	12	3.3V	Integrated 3.3V (+) supply with On-chip linear regulator output within 3.15-3.3V
AIO0	9	Bi-Directional	Programmable input/output line
AIO1	10	Bi-Directional	Programmable input/output line, control output for LNA
PIO0	23	Bi-Directional RX EN	Programmable input/output line, control output for LNA
PIO1	24	Bi-Directional RX EN	Programmable input/output line, control output for PA
PIO2 - PIO11	25 - 34	Bi-Directional	Programmable input/output line

Table 7-2: Bluetooth HC-05 module pins and their functions

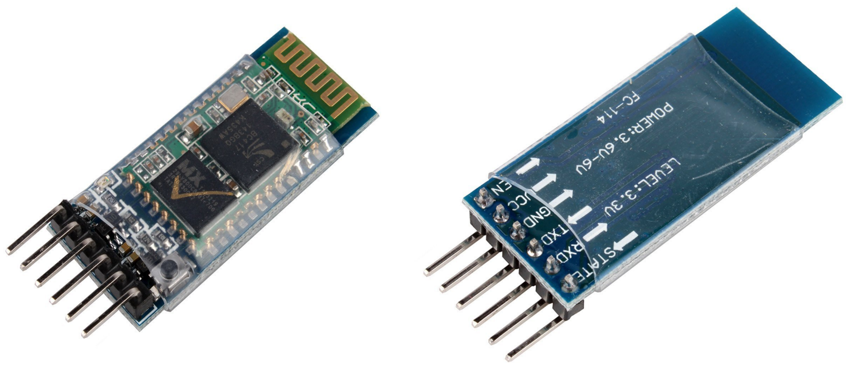


Figure 7-12: HC-05 Bluetooth module, images from Amazon.com

7.1.3.7.1 Block Diagram

To get an idea of how the Bluetooth module itself operates **Diagram 7-4** below shows the flow of components in the system used in the transceiver module necessary for Bluetooth communications. The first block is the antenna which is responsible transmitting and receiving, thus it is also the transceiver. The antenna is then responsible for the next block, which is modulation and demodulation of radio waves into the 2.4GHz band. The information is then sent to the digital signals processor to decipher the radio's encoded information. Connected to the DSP is a 3.3V Power supply required for operation and then random access memories that will store information as it is processed. Lastly is the input/output pins that will be used to connect the whole module to microcontroller. This will handle transmitting and receiving information.

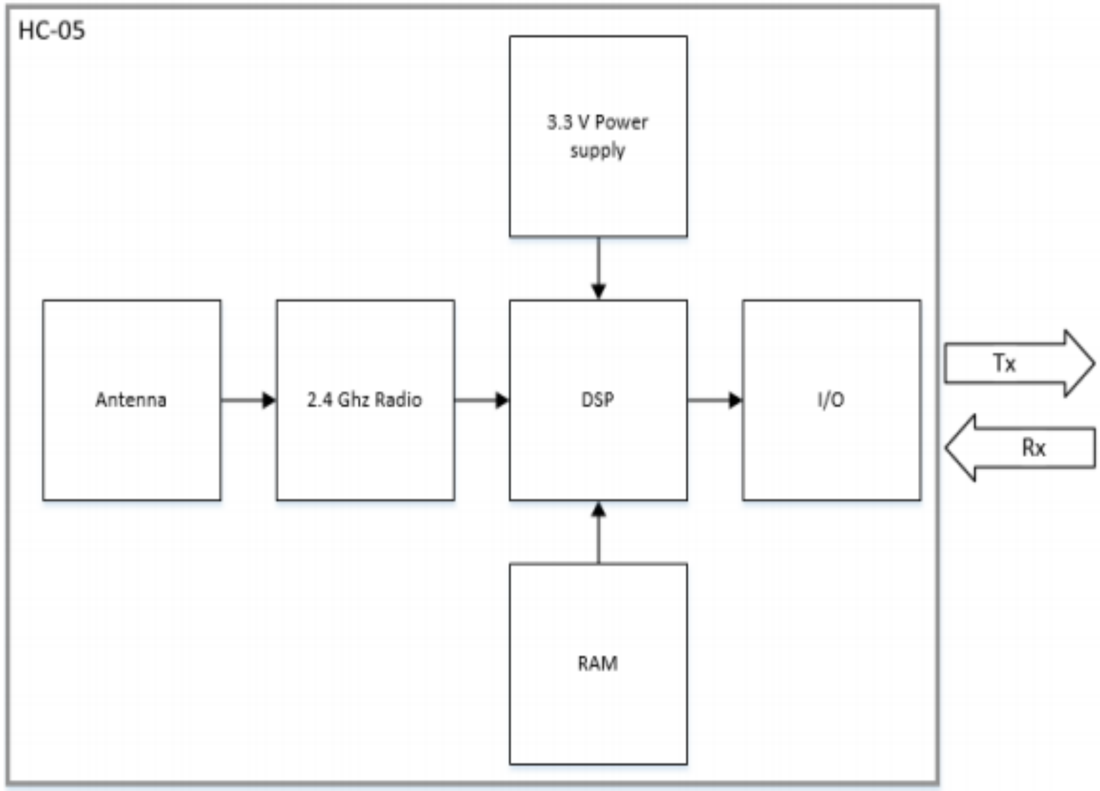


Diagram 7-4: Block Diagram of HC-05 Module

7.1.3.7.2 Schematic

To further understand the process of implementing the HC-05 Bluetooth module a schematic was drawn up using Multisim. In the **Figure 7-13** below, the schematic shows how the HC-05 will be wired to the microcontroller.

The Receiver of the module is connected to the transmitter of the MCU, and the transmitter of the module connected to the receiver of the MCU. This is how the module and microcontroller can send and receive information. Next ground is connected to ground so there is not a short in the circuit. Finally the HC-05's power is received by the microcontroller's output voltage.

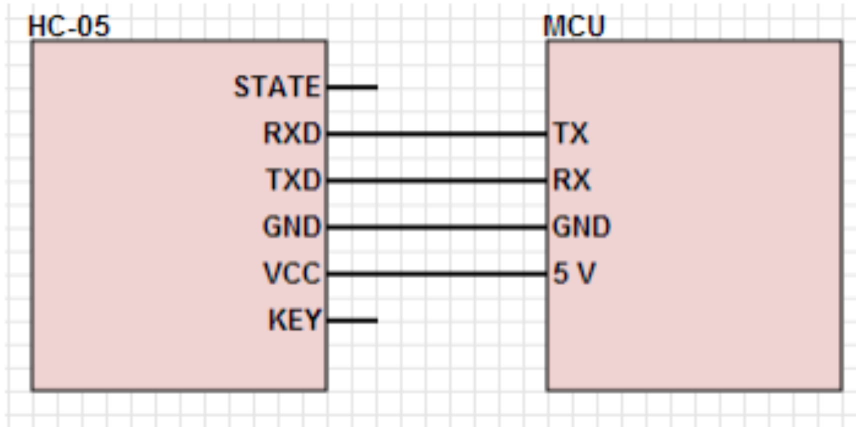


Figure 7-13: Schematic of HC-05 wired to MCU.

7.1.3.8 Future Goals and Hardware Revisions

After the initial prototype the main goal will be to effectively reduce the weight and the voltage requirements of the altimeter. By reducing the voltage requirements the weight could be in turn reduced by effectively changing the altimeter's battery. Also, the use of a custom PCB with a fully integrated system in one board should provide the necessary weight reductions to reach the 9.9g required by Space Trek.

The main way to effectively reduce the altimeters voltage requirements will be to lower the operating frequency for the on board ATMega328p. This will be achieved by using an 8MHz-10MHz crystal oscillator to reduce the voltage requirements. From **Figure 7-14** below we can observe that it is entirely possible to power the ATmega328p with a 3.3V source. With a source like this, the altimeter would be able to be powered by a coin battery and significantly reduce the effective weight. Although a good option, using a coin battery would also remove the benefit of utilizing a rechargeable battery. This is why the coin battery option was dropped from the initial schematic. The altimeter will be initially developed with a lithium-ion rechargeable battery and if the weight requirements are not satisfied then a revision will be in place.

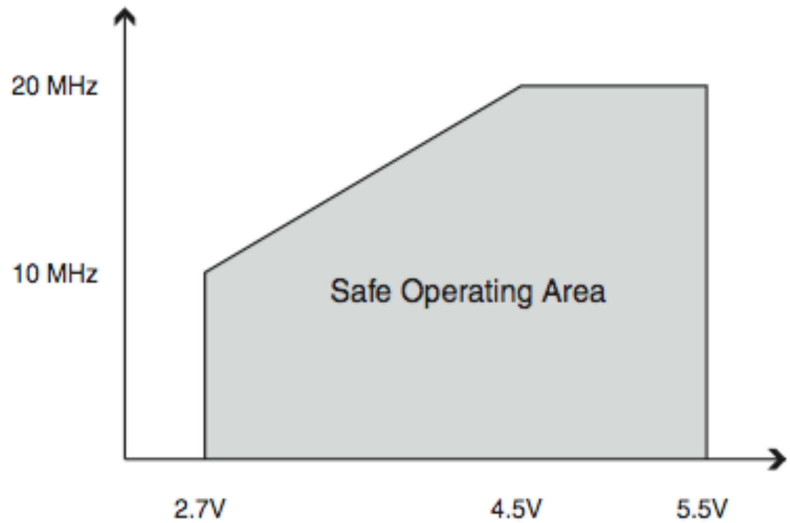


Figure 7-14: Operating frequency to operating voltage graph.
Permission pending since 12/7/2015.

Thanks to the extensive research conducted for the accelerometers, the LIS3DH triple-axis accelerometer is still an option. Based on the specifications in Section 3.2.2.2 this device should reduce footprint and weight, which could be used as a last resort to further reduce the weight of the system.

Probably the single most important revision will be the addition of a Bluetooth module to our altimeter. This module will be a low-cost 4 pin Bluetooth module that will be interfaced with the RX and TX pins on the ATMega328p. Unlike the rest of the components picked for the altimeter, this solution made it very hard to design on Eagle given the team's knowledge in the software. To properly do the schematic including this module, a part number is required, which was basically unattainable with our known resources.

8. Project Prototype Construction and Coding

In the following section, the process of how the team will obtain the first prototype will be detailed. In the case of the Smart Water rocket, many of the mechanical aspects of the system have already been designed since we are improving upon a current system and as such, components like the water rockets themselves, don't need to be acquired.

8.1 Breakout Boards

Like with any other project, prototyping is very important. In the case of this project, obtaining a working prototype as soon as possible is crucial so we can properly work on the communications and software sections of our device. To better satisfy our communications and programming goals in a timely fashion we are going to be using breakout boards for our initial prototyping.

Considerably the main advantage when using a breakout board is the speed with which the group will be able to tackle programming. Most of the programming will require constant testing and editing since a majority will be focused on data transmission and analysis. These are two aspects that will be almost impossible to properly program for without the correct components and environments.

Compared to just using our own custom design, a breakout board provides better time management, since getting our own board manufactured could take months. A breakout board will also provide useful “real-world” information on our selected components during testing since it could always be the case that the end result does not reflect the initial values obtained from our research.

Based on the programming needs of this project and the importance of having a working prototype in due time, the consensus has been to develop the first working prototype using breakout boards and dev boards. When selecting all of the parts, using popular components within the DIY community benefited us with easy access to breakout boards, reference schematics and lots of component feedback assuring the reliability of these parts.

8.2 PCB Vendor and Assembly

Given the current state of our design and the early stages of the project in general, a final PCB vendor has yet to be determined. This does not mean that vendors have not been investigated. Currently the main choices are PCBWay, Advanced Circuits and OSHPark thanks to their outstanding reputation and fast delivery options. Although mounting services are definitely intriguing, we have to make sure that the service falls within our budget which is why mounting will be a secondary priority when selecting the vendor.

Both PCBWay and Advanced Circuits offer student deals as well as manufacture multi-layer boards. The Smart Water Rocket System under its current design, won’t need more than a double-sided circuit board which, but it’s beneficial for future revisions to be in contact with a vendor that is able to manufacture more complex designs such as a multi-layer board.

OSHPark is the cost effective option between the three. With OSHPark we’re able to calculate the price ourselves since pricing is $5 per square inch for the standard dual-layer PCB. This would appear to be in the higher end of pricing, but your “standard” order comes with 3 PCBs which is perfect for a project where things may go wrong and we’re able to save the time of waiting for a new board. Not to mention that we need to design around three altimeters for our sponsor.

Thanks to our sponsors Space Trek, we have access to their PCB prototyping equipment at the Kennedy Space Center. This will provide the team with the ability to quickly work with lower quality boards while constantly improving the final PCB design. Also, we will benefit from soldering stations that could potentially be used for the mounting of our final PCB.

8.4 Final Coding Plan

With the development of a mobile application as well as designing a program for the MCU that will be inside a rocket, there are still several obstacles to overcome. First, when the rocket is in the air, there can either be a direct link between the launch pad and the rocket where all data is sent synchronously or the other option is to store the data. Yet these two options do not have to be mutually exclusive because it is still perfectly viable to have a backup storage just in case the connection is lost and there is no place to store the data. That is why for the rocket the team has decided to go with the route of establishing a permanent connection with the ground while also maintaining a backup storage so that in case the link is lost the data can still be preserved. With this plan there is a redundancy to help increase the accuracy and reliability of the team's rocket design.

In addition to the rocket there is also the question of what features will be on the launchpad itself. Based on the team's ability with software design, a mobile application will be developed specifically for Space Trek that will allow the students and instructors to keep track of the data for all the launches. In addition to a sleek design, an important functionality that will be included is the ability to 'lock' or 'unlock' the launch activation, through the mobile application, in order to press a manual mechanical switch to allow the rocket to launch. This will pose a series of challenges for both the electrical and software components of this design, but once completed will create an elegant design for all to enjoy.

The last major feature of the design that could be implemented, given the method of communication has a constant reliable connection, is to set up a remote server for all the data of the rocket launches be uploaded to. By putting all of the information online, through a simple jQuery web interface for example, all the students can easily look up their information at any time and with proper maintenance will continue to be there for years. Including many of these features will create a lush mobile application that will be appealing for everyone to use, as well as the functionality behind it to create a very fun and stress-free environment.

9 Project Testing

This section includes testing procedures for all of the components of the Smart Water Rocket system. The SWR is made up of the altimeter, automated launchpad, the briefcase, and the software for the briefcase. In order for the entire system to work properly as a whole the SWR team has included various testing procedures for every system at play within the system as well as individual components that require additional testing.

9.1 Hardware Test Environment

To ensure proper communication between the launch briefcase and altimeter there must be a testing procedure to ensure a secure and reliable data link. Before the Smart Water Rocket can be fired off for the project presentation or for the space program, the following tests must be executed. Following the testing procedures prior to either of the aforementioned presentations will smooth out the launch sequence and result in less mistakes and delays.

9.2 Hardware Specific Testing

Based on the time constraints the team will have to follow the coming weeks, a list of hardware tests is crucial to get the Smart Water Rocket system designed properly and on time. Before the testing phase moves to actual subsystem tests, the initial tests will be done on the breakout boards themselves. Once the team has received all of the required components, using the equipment provided at Space Trek or the University of Central Florida, voltage and current tests will be conducted. Ideally we will connect each breakout board separately and make sure we are obtaining the necessary inputs. The following set of tests will be conducted in the reliability of our Bluetooth module and verify it is indeed enough.

If any of the tests were to fail, the team will have enough time to ensure the necessary part orders are placed. This will ideally save the team unforeseen malfunctions later in the prototyping phase.

9.2.1 Communications Network Testing

To ensure proper communication between the launch briefcase and altimeter there must be a testing procedure to ensure a secure and reliable data link. Before the Smart Water Rocket can be fired off for the project presentation or for the space program, the following test must be executed.

- Step 1 - Power on the tablet and navigate to settings to ensure that Bluetooth pairing is active on the device.
- Step 2 - Connect or power on the Bluetooth module or chip connected with the altimeter.
- Step 3 - Search for devices on the tablet and pair with the altimeter to establish a connection.
- Step 4 - Now that the devices are paired, test launches must be demonstrated at various water pressure levels to determine the range and reliability of data transmission.
- Step 5 - Launch the rocket at ~100 psi (optimal launch trajectory) and see if the altimeter can send information from the distance it landed.

9.2.2 Automated System Testings

For the automated launchpad the previously manually operated construction will be redesigned to incorporate a microcontroller and stepper motors to make the system more convenient for the Space Trek team. With using stepper motors that are cheap and efficient on power but their mechanical operation is not the most precise type of motor control. The following **Table 9-1** will be implemented as a test to see how accurate the automated launchpad actually is.

The values of degrees selected are not normal to standard SWR launches they are simply to test the accuracy of the finished product and hopefully tune the system prior to launching the actual rockets.

Degree	Measured	Actual	Percent Difference
0			
20			
30			
40			
45			
50			
55			
60			
70			

Table 9-1: Degrees testing.

Another component of the automated launchpad is the new water/air compression system. Previously the rocket was pressurize using a bike pump and some tubing connected to the launchpad but with the addition of an air compressor the launchpad will be made even more automated and convenient for the Space Trek team.

The following **Table 9-2** and **Table 9-3** are included for testing of the air-compression system that will operated using a light-weight, compact air compressor connected through similar tubing to the bike pump setup. The air compressor and a water tank will be connected using this tubing with a Y-valve that can be switched to change between which volume will be brought into the water rocket.

Values in the tables below are purely for testing of accuracy that the air-compression system maintains. Whether the system can perform more reliable results at higher pressures is most important but lower pressure values will also be tested for accuracy to acquire the relative reliability of the system.

Water Pressure (PSI)	Measured (PSI)	Actual (PSI)	Percent Difference
20			
30			
45			
60			
70			
80			
90			
100			
110			

Table 9-2: Water Pressure Testing

Air Pressure (PSI)	Measured (PSI)	Actual (PSI)	Percent Difference
20			
30			
40			
50			
60			
70			
80			
90			
100			

Table 9-3: Air Pressure Testing

9.2.3 Launch Sequence Testing

The launch briefcase is a major component of the Smart Water Rocket system and must be tested and working prior to launch. There are several important software and hardware checks that if followed

- Step 1 - Power on the tablet and make sure there is a secure connection with the launch sequence buttons module.
- Step 2 - Check to make sure the launch button hub is receiving power and the indicator light is lit red.
- Step 3 - Now that the system is connected and ready, launch the Smart Water Rocket application on the tablet.
- Step 4 - Select the rocket profile tab and enter the team's unique rocket code. With the particular launch team's rocket profile selected, the SWR system is ready to acquire simulated launch information.
- Step 5 - Begin entering water pressure and angle of deployment for the SWR into the launch information section. This information will be tested and distributed to each group following a controlled simulation demonstrated inside the Space Trek building.
- Step 6 - Now that the application has the necessary data to begin the launch sequence, the program will activate the buttons and await user input to initiate the launch.
- Step 7 - Open the ignition switch cover and flip the metal toggle to proceed onto the key code activation.
- Step 8 - Enter the rocket team's unique launch code on the 12-button keypad in order to activate the final launch button.
- Step 9 - When the launch button changes from red to green press it to fire off the Smart Water Rocket.

9.2.4 Keypad Testing

To integrate the keypad into the launch sequence design there must be a procedure for identifying which pins are connected to every key (see **Figure 9-1**). The outer lying pins are for connecting the keypad to a circuit. The remaining seven pins will be tested using an Ohmmeter. The following procedure was used to test which pins will be triggered when pressing each key.

- Step 1 - Connect the Ohmmeter to pins 1 and 2.
- Step 2 - Press through each of the 0-9 keys until the meter indicates closure (current passes through the pins)
- Step 3 - Write down the pin numbers next to the number that has been identified as a connection. For instance when connected to pins 1 and 2, the Ohmmeter responded when the number 2 was pressed on the keypad.
- Step 4 - Repeat the process by connecting the Ohmmeter with pins 2 and 3.
- Step 5 - Press the keys until the next number has been indicated.

- Step 6 - Continue this process until all keys have their proper pins identified.
- Step 7 - Create a diagram for the keypad with columns and rows indicating which pins they are connected to.

On the following page is a picture illustrating the tested information in several different ways:

1. The first recordings were to identify the pins connected to every keypad push button. This list was first made to clarify and from there could be used to label the following diagrams.

2. With all of the keypad numbers identified the keypad was illustrated with the proper columns and rows highlighted for another way of picturing where the pins are connected.

3. Now that the keypads are illustrated with a diagram of the columns highlighted and a diagram of the rows highlighted, the corresponding push buttons have been re-listed with the proper color codes.

4. The final image on the top right is the initial labeling of what pins are connecting to lined up with the actual rows and columns.

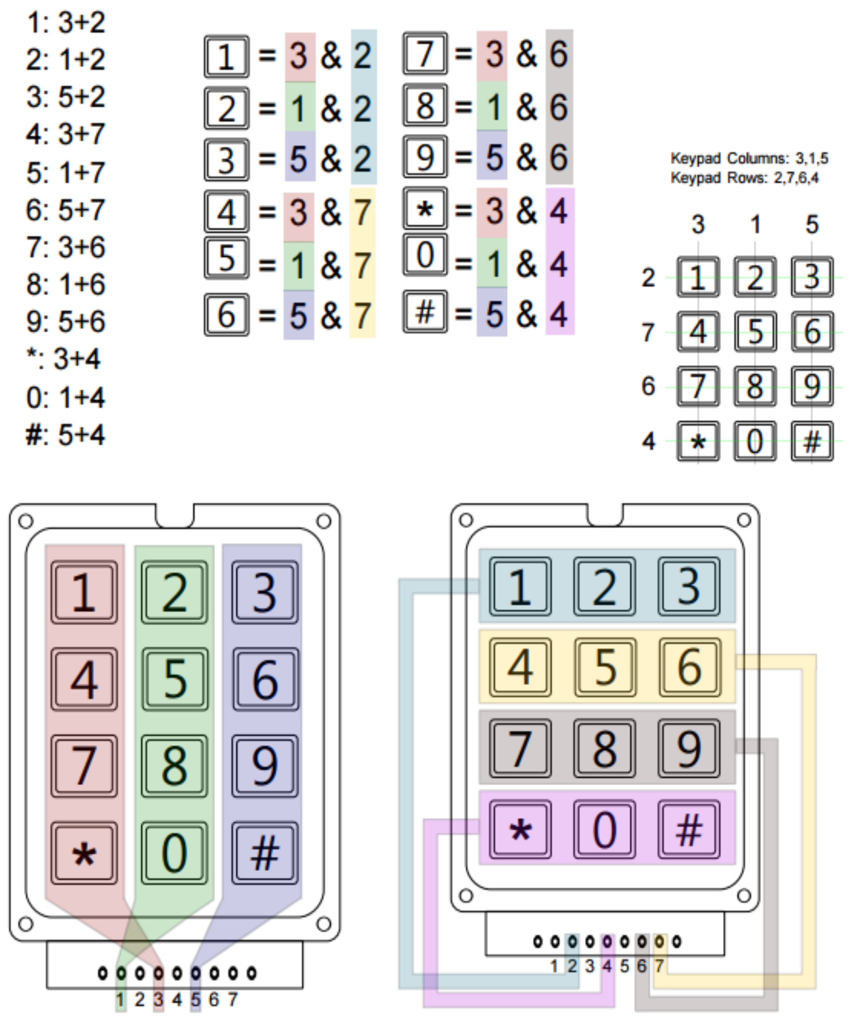


Figure 9-1: Keypad pins identified with columns and rows labeled

9.3 Software Testing Environment

The software will need to be tested in order to ensure a reliable and accurate end result for this project. First is where the testing will be done, given the ample amount of resources at the disposal of the group there are a few locations where the testing can be implemented. For instance, the major locations would be the engineering lab at UCF or the lab that Space Trek has at Kennedy Space Center. Either of these locations would suffice and provide many great tools for us to be able to use. When those options are not available, the testing will be done on the team’s Computer Engineer’s computer. With ample processing power it will be able to handle anything that the team has in store for it.

9.3.1 Motherboard

EVGA X58 SLI Motherboard

- Intel Socket 1366
- Intel X58 Chipset
- Dual or Triple Channel DDR
- Up to 3-Way SLI Support
- 12 USB Ports
- 3 GB/s SATA
- 1x PCIe x16
- 1x PCIe x8/x16
- 1x PCIe x8
- 1x PCIe x1
- 2x PCI

9.3.2 Processor

Intel Core I7-920 Processor

- 8 MB Cache
- 2.66 Ghz Base Frequency (Up to 2.93 Ghz Boosted)
- 4.8 GT/s QPI
- Supports up to 24GB of memory
- 4 Cores
- 8 Threads
- 25.6 GB/s Max memory bandwidth

9.3.3 Random Access Memory

Corsair 2GB Desktop Memory (2GB x 6)

- DDR 1333
- Timing 9-9-9-24
- Cas Latency 9
- Voltage 1.5V

9.3.4 Storage

Western Digital Blue 500 GB (500GB x 2)

9.3.5 Graphics Card

EVGA GeForce GTX 660 SUPERCLOCKED (2-way SLI)

- Max Resolution 2560 x 1600
- Processor 1046 MHz
- 2048 MB GDDR5 SDRAM
- Memory Speed 6008MHz
- 2 GB VRAM

9.3.6 Power Supply

Corsair 600 Watt Power supply

9.4 Software Testing Procedure

The procedure for testing the software is very straight-forward, but extremely vital for knowing what our team can achieve from the design. There will be testing for the communications, storage of data, functionality of mobile application, and sensor readings. These are the major features that will be tested for first but that's not to say there won't be smaller tests that will need to be performed as well.

To start the communications is the major focus of not only testing the software but for the project as a whole it will not succeed without successful communications. Without a proper communication line between the rocket and the launchpad, this project would change nothing about the procedure of the current water rocket launch system. With that in mind, it is important for the team to note which communications medium will work the best. There are three main modes of communication that will be tested: Bluetooth, Wi-Fi, and Radio transmission.

For Bluetooth, the group will test the distances at which a pair will be lost, and test how quickly the pairing is reconnected and to gauge the total data lost in the transaction. It is very important to lose as minimal data as possible which is why it's one of the first and most essential test that will be performed. In addition to Bluetooth, Wi-Fi and radio will be tested in a similar fashion.

Distance (ft)	Connection	Data Transferred	Transfer Rate
0			
15			
30			
45			
60			
75			
90			
105			
120			

Table 9-4: Testing Table for Communications.

The test procedure (see **Table 9-4**) consists of testing the distance at which the communications will remain intact, and how much data was transferred at that distance. The following will be the testing procedure for the communications.

1. Connect the rocket and launch pad devices together
2. Move the rocket in increments of 15 feet and test the connection
3. If a proper connection exists, test if the data will transfer.
4. If a successful transmission of data occurs, record the transfer rate

Testing all the forms of communication is paramount to moving forward with the project and having a successful project over all. Since this is a preliminary testing phase, all of these modules will be tested on experiment boards and the most successful mode of communication will be chosen.

Next step in the testing phase is to test data storage, which will be important if there is no way to guarantee a consistent connection throughout the entirety of the launch. If the communication tests proves to be lacking in this constraint, then an on-board storage device will be used to store additional flight data (if it is more than the MCU can safely store).

Test Number	Data Stored	Transmission	Data Loss
1			
2			
3			
4			
5			

Table 9-5: Testing Table for data storage

The following **Table 9-5** is the testing procedure for the on board data storage:

1. Initialize Rocket with controlled conditions
2. Store entire launch data within onboard memory
3. Transmit the data back to the launch pad once landed
4. Record any data lost

Once the data has been stored it can be sent to the launch pad once the connection has been reestablished between the two. To test the functionality of the storage, there will be an additional altimeter (the Jolly Logic currently in use at Space Trek) used to compare the results and see how accurate the stored data is.

In addition to storing the data, it is important to test the sensors themselves, to know if that data is precise when compared to the current altimeter in use. Since data collection and storage are going to be working in conjunction, this will be a two-part test that will cover all the grounds. When the data is collected it will all be stored in real-time, which will allow for an accurate comparison between the two.

Time (s)	Height	Velocity	Accuracy
0			
0.25			
0.5			
0.75			
1			
1.25			
1.5			
1.75			
2			

Table 9-6: Testing Table for Sensor Data

This test (see **Table 9-6**) is very important to ensure the sensors are obtaining accurate information and that the data is being saved, which is a determining factor in the success of this project as a whole. The following is the test procedure for testing the Sensor Data

1. Initialize Sensor and control altimeter
2. Launch Rocket and record readings at specific intervals
3. Compare results with control

When it has been revealed how reliable the storage will be for the data is when the group can move forward with the testing phase. In the case that a consistent connection can't be made and the additional storage is not capturing all the required data, then another initiative will be taken in order to complete the goal for this project.

After all the data capturing, storage and transmissions have been dealt with, the last major obstacle to overcome is connecting the launch pad touch screen device, with the mechanical switches connected to the rocket itself. While creating a mobile application, and designing a simple switch circuit are not complex on their own, trying to combine them will prove to be a challenge.

Launch Code	Activation	Output (V)	Error
1111			
2222			
3333			
4444			
5555			
6666			
7777			
8888			
9999			

Table 9-7: Testing Table for Launchpad Activation

Table 9-7 above details testing conditions of our onboard launch system. When testing the switches, it is important that they do not activate unless the launch code has been put in, which will prevent misfires and otherwise unplanned launches. The following is the testing procedure for the keypad and launch button activation:

1. Initialize Launch pad with given launch code
2. Enter launch code
3. Test Launch Button
4. Record output Voltage from button
5. Record any errors received

With a successful integration of digital and mechanical systems, the launch station would be considered a success, thus concluding the testing phase of the software components of this project.

10 Administrative Content

The Smart Water Rocket System is just like any other project of this magnitude. To succeed various measures have been implemented to enable a smooth progress, a budget has been implemented and contact with our sponsor Space Trek has been essential.

10.1 Budget and Finances

Thanks to our sponsor we've been given access to not only equipment, but also a very generous budget of $1000. This budget is intended to cover the costs of any additional parts that our sponsors don't have available at their disposal.

Table 10-1 below is an initial estimate of costs per part based on current prices. The prices reflected here are based on the first prototype, which is mostly comprised of breakout boards and no single component will be mounted to a custom PCB This is only an estimate, and the changes in our product will most likely change the required parts, and in turn the final cost.

Component	Price($)
Pressure Sensor	9.95 (x3)
Accelerometer	17.95 (x3)
Voltage Regulator	1.15 (x9)
Arduino Uno Rev3	24.95 (x2)
Bluetooth Modules	15 (x4)
Motor	14
Solar Panels	30
Batteries	30
Misc. Mechanical Parts	30
Misc. Electrical Parts	20
Total	**$298**

Table 10-1: Budget Estimates

Considering these estimated costs, the $1000 financing from Space Trek is more than sufficient to say the least. This will give us incredible flexibility if or when problems arise. By keeping our initial budget controlled, taking advantage of mounting services or express shipping components and PCBs might be possible.

10.2 Progress Updates

To properly keep up with the work at hand, we set a goal of 3-5 pages per week per team member. Also, after analyzing each member's availability, each Monday of every week was set to organize the aforementioned work. To keep matters organized and create a record of our progress, a table was implemented for every meeting.

Month	Progress
September (1 wk)	● First meeting ● Roles were set for each member ● Page count goal implemented
October (4 wk)	● Page Count (22) ● Design changes were made as requested by our sponsors
November (4 wk)	● Page Count (45 at the time of the first draft) ● Focus on design schematics, block diagrams, and familiarizing with Eagle
December (1 wk)	● Page Count (120+) ● Final revisions and formatting

Table 10-2: Monthly progress updates

Table 10-2 above is a monthly summary of the work conducted during those months' meetings. This table will keep to be updated as the project moves into its prototyping phase which will be detailed more in depth in the following section (see Section 10.3).

10.3 Milestones

Going into the prototyping and final design phases of the project will be a fast paced experience. To prepare the team for what's to come, we have devised bi-weekly goals (see Table 10-3) to ensure the success of the Smart Water Rocket System. Starting with the return from Winter Break we will be working closely with our sponsor and using any resources at our disposal to manage the workload.

Week Number (Jan 11- April 14)	Tasks and Deadlines
Week 1	● Order required prototyping boards and parts ● Redesign schematics ● Begin coding the microcontroller units
Week 3	● First prototype electrical and mechanical tests ● Begin app design and launchpad automatization ● Deadline: Bluetooth efficiency test
Week 5	● Test flights using prototype board ● Deadline: Working mechanical system ● Begin work on custom PCB design
Week 7	● Begin automating the mechanical system
Week 9	● Deadline: PCB design ● Deadline: Microcontroller coding ● Deadline: Communications system (Bluetooth)
Week 11	● Deadline: Solder components ● Deadline: Assemble all subsystems
Week 13	● Performance tests ● Deadline: App integration
Week 14	● Deadline: Final aesthetic details

Table 10-3: Bi-weekly upcoming project goals

10.4 Group Responsibilities

Each SWR team member has their own primary roles in their desired areas. **Table 10-4** below details what are the main roles of each member, but it doesn't cover instances of collaboration between teammates. More in depth descriptions of each member's responsibilities and target goals are detailed below.

Altimeter, Gabriel Fernandez

- Retain center of mass
- Dry weight of rocket is 340 grams
- Transmit in both imperial and metric units
- Current altimeter mass is 9.9 grams
- Must collect acceleration, height, and speed data for whole flight

- Achieve a reliable data transmission

Computer Analysis Program and Data Storage, Kyle Moran

- Retain all flight data so it can be saved to another location later
- Data can be saved to a remote server, whether on-site or cloud-based, to manage
- all the flight data and analyze it back in the classroom
- Display data in flight briefcase (Touch Screen Tablet or LCD Display with buttons)
- Will display current stats (Velocity, Height, Acceleration)
- Also will show max height and max velocity once rocket has peaked

Software

- TI-MSP Family of Micro Processors - Code Composer Studio
- Arduino - Arduino IDE
- Raspberry Pi - Linux
- LabView / MatLab - Create graphs of trajectory to be displayed on launch pad display
- Minimal data loss - efficient use of assembly to maximize data transfer and storage
- Android Studio - Create an application for Space Trek to have a Graphical User Interface
- complete with launch functionality and managing flight data

Communications Network, Group

- Bluetooth, wifi, LTE
- Cost must be minimized
- All data must be transmitted either in real time or immediately after launch
- Example launch distance: 104ft max height, 75 mph max velocity

Launch Briefcase Controls, Connor Phillips

- Display (tablet or screen connected to raspberry pi)
- Launch sequence (Seven Segment LCD Display, Countdown multi-tone, Key activated system, separate launch codes for each rocket, tangible launch button or digital)
- Launch abort system (Big red button to abort launch sequence)
- Every modifiable variable should be controlled from here
- Bluetooth receiver

Launchpad Automation, Matthew Turner

- Portability
- Automatically set the launch angle of the rocket
- Automate the water dispersion and pressurization of the rocket

- Power

Member	Primary Role(s)
Fernandez, Gabriel (EE)	Altimeter schematic and PCBs
Moran, Kyle (CPE)	App design and MCU coding
Phillips, Connor (EE)	Power system and briefcase schematic
Turner, Matt (EE)	Launchpad design and electro-mechanical interface

Table 10-4:_Group member and primary role(s). To keep the work distribution equal each member of the team chose a primary role.

10.5 Sponsor: Space Trek

The Space Trek program at Kennedy Space Center (located at Florida's Space Coast in Cape Canaveral) is designed to promote exploration of the wide field of cutting-edge space research undertaken by NASA. Intended to help students get involved in STEM related fields, Space Trek's simulated situations challenge students to provide radical technological solutions to crucial problems being faced by the Space industry. A versatile robotics program dares students to design, build, program and work with robots to experience life-like engineering. Students are encouraged to develop mathematical models for real-life problems that can be applied to the space industry.

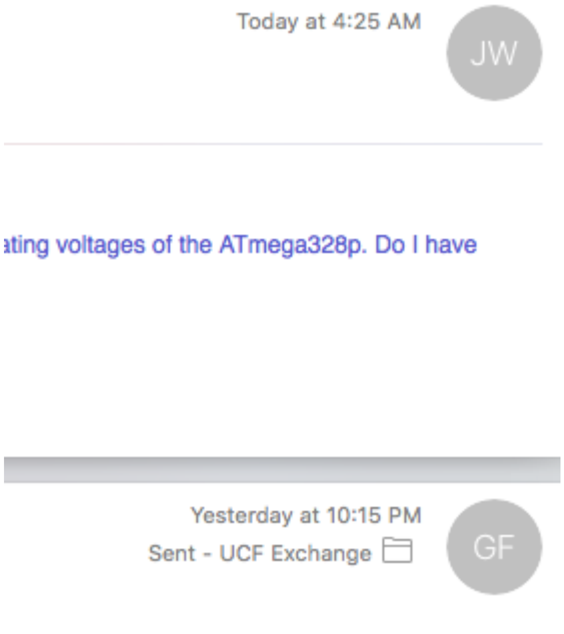
Atlantis Educational Services (Space Trek's parent company) was founded by a young visionary who wanted to help students to compete and excel at the international level. He perceived that while there are many options for students to attend schools specializing in science, technology, engineering, and mathematics; that the curriculums of those programs had become static. He wanted to create something new and fresh.

In order to do achieve this he assembled a team of investors who would demand the opportunity to influence the product from inception and throughout its evolution. Each investor brought with them a wealth of knowledge as the result of their successful engineering careers. He next looked for a person to lead development of the actual educational programs and found an award winning teacher and administrator with years of experience with both NASA and international programs. This team understands the challenge of preparing students for the future both from an education and industry viewpoint. This team recognizes what it takes for students to succeed at the next level because they live and work at that level every day.

Appendix A: Sources

1. https://www.sparkfun.com/pages/accel_gyro_guide
2. http://www.dimensionengineering.com/info/accelerometers
3. http://www.electroschematics.com/9798/reading-temperatures-i2c-arduino/
4. http://www.homautomation.org/2014/04/03/best-ways-to-power-a-arduino-according-to-your-need/
5. http://www.instructables.com/id/The-Ultimate-Altimeter-A-compact-Arduino-altimeter/
6. https://www.dropbox.com/sh/xd3vv0lubuwmndv/AABlJuhs6xGoxZ-HGetreOQJa?dl=0
7. http://www.amazon.com/Dragon-Touch-Bluetooth-Technical-TabletExpress/dp/B00T5TIL80/ref=sr_1_2?s=pc&ie=UTF8&qid=1445131638&sr=1-2&keywords=dragon+touch&refinements=p_72%3A1248879011
8. https://www.raspberrypi.org/blog/the-eagerly-awaited-raspberry-pi-display/
9. http://www.instructables.com/id/External-Bluetooth-Antenna-for-Audio-Injecting/
10. http://www.altenergymag.com/content.php?post_type=1884
11. http://www.digitaltrends.com/home-theater/led-vs-lcd-tvs/
12. http://www.amazon.com/Microsoft-Surface-Pro-Intel-Core/dp/B01606KJ6C/ref=sr_1_1?s=pc&ie=UTF8&qid=1449444754&sr=1-1&keywords=surface+pro
13. http://www.amazon.com/Apple-MD786LL-9-7-Inch-Touchscreen-Tablet/dp/B00G2X1VIY/ref=sr_1_3?s=pc&ie=UTF8&qid=1449444596&sr=1-3&keywords=ipad+air
14. http://www.amazon.com/JBtek-Wireless-Bluetooth-Transceiver-Arduino/dp/B00L083QAC/ref=sr_1_1?ie=UTF8&qid=1449444091&sr=8-1&keywords=hc-05
15. https://www.sparkfun.com/products/11310
16. https://www.sparkfun.com/products/8653
17. http://www.spacetrek.us/whatwedo.html
18. http://www.instructables.com/id/Self-Sufficient-Arduino-Board/?ALLSTEPS
19. http://www.reuk.co.uk/Lead-Acid-Batteries.htm
20. http://kids.britannica.com/comptons/art-52969/A-cutaway-diagram-shows-a-nickel-cadmium-rechargeable-cell-Its
21. http://physicscentral.com/explore/action/lithium.cfm
22.

Appendix B: Permissions



Mr. Wippler,

I am an Electrical Engineering student from the University of Central Florida. I would like to include the attached picture found in your post "Voltage: 3.3 vs. 5" to illustrate the operating voltages of the ATmega328p. Do I have your permision?

Sure - it comes from the ATmega328 datasheet.

Cheers,
-jcw

Request for use of image from one of your articles.

CO connor.phillips
To: dprindle@digitaltrends.com;

Reply all

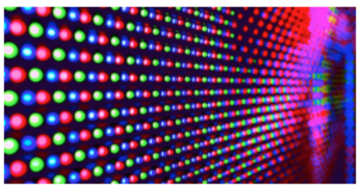
Mon 11/23/2015 5:24 PM

Hello Drew Prindle

I was doing some research for a technical paper I'm writing and I would like to ask your permission to use the image in the article linked below that shows the difference in picture quality of LED vs. LCD TVs.

Thanks,
Connor Phillips

http://www.digitaltrends.com/home-theater/led-vs-lcd-tvs/



LED vs LCD | Digital Trends

Our LED vs LCD TV buying guide explains why LED and LCD TVs are fundamentally similar, how they differ, and what to look for in buying an LED TV.

Read more...

Connor Phillips
Subject: Request for use of images

NOV 30, 2015 | 01:13PM MST
Chris F replied:

Hi Connor.

As long as you are using the images non commercially and give us credit, you can use them.

Chris

Technical Support
SparkFun Electronics, Inc.
6333 Dry Creek Parkway
Niwot, CO 80503
E-mail: techsupport@sparkfun.com
Phone: 303-284-0979

NOV 29, 2015 | 09:12AM MST
Original message
Connor wrote:

Hello,

I am writing a research paper for a school project and I must get permission to use the following images in my document. These items are possible components for the design of a control briefcase I will must build and I will later order the parts when ready.

Thank you,

Connor Phillips

University of Central Florida

COM-09181

(big dome pushbutton - Red)

COM-08653

(Keypad - 12 button)

COM-11310

(toggle switch and cover - illuminated (red))

Hello Connor,

Thank you for writing to us.

I can confirm that you can use the pictures for your project for educational purposes.

Hope this helps.

We'd appreciate your feedback. Please use the links below to tell us about your experience today.

Best regards,
Mounica

Did I solve your problem?

Yes No

Your feedback is helping us build Earth's Most Customer-Centric Company.

Thank you.

Amazon.com

12/06/15 15:35:52
Your Name: Connor
Other info: Request for use of images
Comments: Hello Amazon Customer Support Representative,

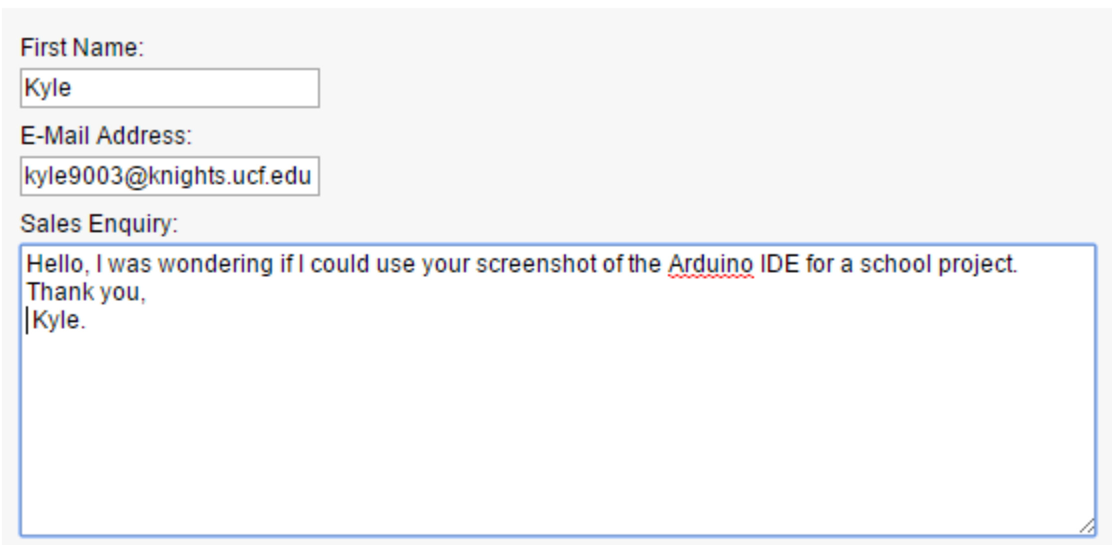
I am a student at the University of Central Florida working on my senior design project and I'm interested in purchasing the following items for use in the design of my teams project. These are possible items that will be included in our research and I must request the use of their images available in the links below.

Thank you,
Connor Phillips

Surface Pro 4
http://www.amazon.com/Microsoft-Surface-Pro-Intel-Core/dp/B01606KJ6C/ref=sr_1_1?s=pc&ie=UTF8&qid=1449444754&sr=1-1&keywords=surface+pro

iPad Air
http://www.amazon.com/Apple-MD786LL-9-7-Inch-Touchscreen-Tablet/dp/B00G2X1VIY/ref=sr_1_3?s=pc&ie=UTF8&qid=1449444596&sr=1-3&keywords=ipad+air

JBTek HC-05 bluetooth module
http://www.amazon.com/JBtek-Wireless-Bluetooth-Transceiver-Arduino/dp/B00L083QAC/ref=sr_1_1?ie=UTF8&qid=1449444091&sr=8-1&keywords=hc-05


First Name:
Kyle
E-Mail Address:
kyle9003@knights.ucf.edu
Sales Enquiry:
Hello, I was wondering if I could use your screenshot of the Arduino IDE for a school project.
Thank you,
Kyle.

To | WE | webmaster@eclipse.org

Cc

using a screenshot of code composer studio for a school project

Hello, I was wondering if it would be okay to use your screenshot of code composer studio for a Senior Design Project that my group is doing for the University of Central Florida.
Thank you,

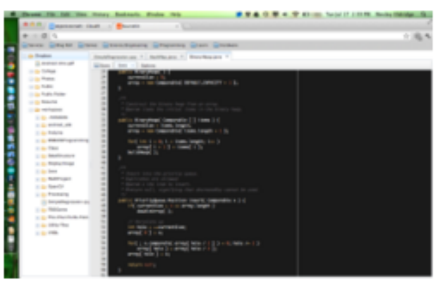
Kyle

To | WE | weseldridge@gmail.com

Cc

permission to use pictures

Hello, I was wondering if I could use the pictures from your blog https://weseldridge.wordpress.com/2012/07/17/summer-code-ii-tools-part-1/ for a school project.



Summer Code II: Tools - Part 1

Tools play a big part in the development process. However, they should not distract from your work. The key is to find a set of tools that fit your needs. From here on out, I will discusse some of ...

Read more...

Remove preview

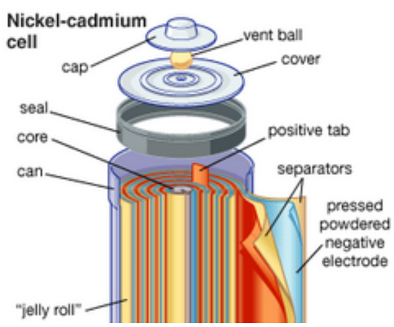
thank you,
Kyle

To | CO | contact@eb.com

Cc

Permission to use image

Hello I am a student at the University of Central Florida working on a senior Design Project and I was wondering if I could have permission to use your image of a nickel cadmium cell as found in this article: http://kids.britannica.com/comptons/art-52969/A-cutaway-diagram-shows-a-nickel-cadmium-rechargeable-cell-Its



battery: nickel-cadmium battery -- Kids Encyclopedia

...

A cutaway diagram shows a nickel-cadmium rechargeable cell (or battery). Its "jelly roll" construction allows high current to be delivered efficiently.

Read more...

Contact Us

Email*:
le9003@knights.ucf.edu

Subject:
Permission to use image

Direct your comment to*:
PhysicsCentral

Comment/Question*:

I could have permission to use your picture of a lithium-ion battery as found here: http://physicscentral.com/explore/action/lithium.cfm

Thank you,
Kyle.

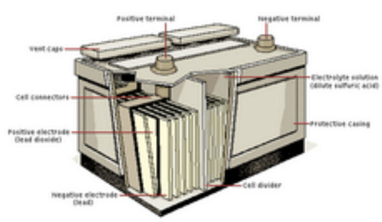
Submit

To NE neil@reuk.co.uk

Cc

permission to use pictures from article

Hello, I am a student at University of Central Florida and I was wondering if I could use the image of the lead-acid battery from this article for a Senior Design Project: http://www.reuk.co.uk/Lead-Acid-Batteries.htm.



Lead Acid Batteries - Storage - REUK

Lead-acid batteries store energy using a reversible chemical reaction between lead plates and dilute sulphuric acid (electrolyte). There are three basic types of lead ...

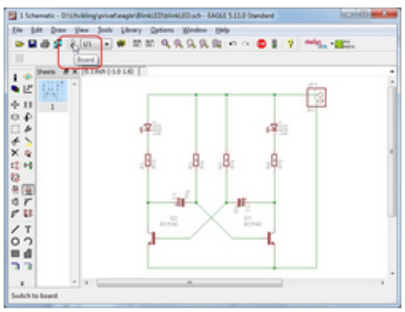
Read more...

To DA Dahl@build-electronic-circuits.com

Cc

Request to use Image

Hello, I am a student from the University of Central Florida and I was wondering if I could have permission to use your picture of EAGLE for a senior design project. The image can be found at this link: http://www.build-electronic-circuits.com/pcb-design-tutorial/



PCB Design Tutorial for Eagle - Build Electronic Circuits

This is a PCB design tutorial that I designed as a simple way to learn how to create a PCB layout in Cadsoft Eagle. What we need: A computer with any version of ...

Read more...