

**Project DZERV**

**Disaster Zone Emergency Response Vehicle**

Spring 2012-Fall 2012

Group F

Group Members:

Marcial Rosario

Robert Smith

Michael Lopez

Table of Contents

1.0 Executive Summary.......................................................................................................1

2.0 Project Description.........................................................................................................2

2.1 Project Motivation and Goals...................................................................................2

2.2 Objectives.................................................................................................................3

2.3 Project Requirements and Specifications.................................................................4

2.3.1 Risks................................................................................................................6

2.3.2 Team Management..........................................................................................6

3.0 Research Related to Project Definition..........................................................................6

3.1 Existing Similar Projects and Products....................................................................6

3.1.1 Stevens Institute of Technology Search and Rescue.......................................7

3.1.2 OSU Search and Rescue Robot.......................................................................7

3.1.3 Autonomous Ball Collector (ABC).................................................................9

3.1.4 Other Available Products...............................................................................10

3.1.4.1 Unmanned Aerial Vehicles (UAVs)..................................................10

3.1.4.2 Kinect Robot......................................................................................11

3.2 Relevant Technologies...........................................................................................12

3.2.1 Wireless Protocols.........................................................................................12

3.2.1.1 Wi-Fi (IEEE Standard 802.11).........................................................12

3.2.1.2 Bluetooth..........................................................................................13

3.2.1.3 Infrared Communications.................................................................15

3.3 Strategic Components............................................................................................15

3.3.1 Mobile Base...................................................................................................16

3.3.2 Surface Mount vs. Through-Hole..................................................................17

3.3.3 Microcontrollers.............................................................................................17

3.3.4 Cameras..........................................................................................................18

3.3.5 Sensors............................................................................................................18

3.3.6 RF Transmitters and Receivers.......................................................................19

3.3.7 Power Supplies................................................................................................19

3.3.8 Base Computer.................................................................................................20

3.3.9 Graphical User Interface..................................................................................21

3.3.10 Voltage and Current Regulation....................................................................21

4.0 Project Hardware and Software Design.......................................................................21

4.1 Microcontroller......................................................................................................21

4.1.1 Research Phase.............................................................................................22

4.1.2 PIC16F886...................................................................................................24

4.1.3 Design Phase................................................................................................24

4.1.4 Programming................................................................................................26

4.1.5 Testing and Debugging.................................................................................28

4.2 Sensors/Instruments.....................................................................................................29

4.2.1 Pressure Sensor (MPLII5A1).............................................................................29

4.2.2 Temperature and Humidity Sensor (HTM1735LF)...........................................31

4.2.3 Ultrasonic Range Finder (LV-EZ1)...................................................................33

4.2.4 LED Bar..........................................................................................................35

4.3 Camera (AirSight Wireless IP Camera)...................................................................36

4.3.1 MAX232 Chip................................................................................................39

4.4 Software/Interface....................................................................................................41

4.4.1 GUI.................................................................................................................42

4.4.2 Robot Control.................................................................................................44

4.5 Voltage Regulators...................................................................................................45

4.5.1 Linear Regulators vs. Switching Regulators...................................................46

4.5.2 Linear Regulator Options................................................................................48

4.5.3 Switching Regulators......................................................................................51

5.0 Power and Communication..........................................................................................52

5.1 Chassis....................................................................................................................52

5.2 Motors.....................................................................................................................54

5.2.1 Continuous DC..............................................................................................54

5.2.2 Stepper Motor................................................................................................56

5.2.3 RC Servo Motors...........................................................................................57

5.2.4 Price Comparison...........................................................................................59

5.3 Communications...................................................................................................61

5.3.1 Infrared Communication..............................................................................61

5.3.2 RF (Radio Wave) Communication...............................................................62

5.3.3 Wireless Adapters ........................................................................................63

5.4 Power Supply........................................................................................................67

5.4.1 Electronics Power Supply.............................................................................67

5.4.2 Alkaline.........................................................................................................68

5.4.3 Lithium..........................................................................................................69

5.4.4 Lead Acid......................................................................................................70

5.4.5 Nickel Cadmium (NiCd)...............................................................................70

5.4.6 Nickel Metal Hydride (NiMH)......................................................................70

5.5 Choosing a Battery...............................................................................................71

5.5.1 Battery Charger.............................................................................................73

5.6 Antenna.................................................................................................................74

5.6.1 Ceramic Antenna...........................................................................................74

5.6.2 Planar Antenna...............................................................................................75

5.6.3 Whip Antenna.................................................................................................76

5.7 Motor Controller....................................................................................................77

6.0 Project Prototype Construction and Coding................................................................79

6.1 Strategic Locations................................................................................................79

6.1.1 EECS Senior Design Laboratory..................................................................80

6.1.2 Group Members’ Homes..............................................................................80

6.2 Parts Acquisition....................................................................................................81

6.2.1 Vendor Selection...........................................................................................81

6.3 Printed Circuit Boards............................................................................................82

6.3.1 ExpressPCB....................................................................................................82

6.3.2 Sunstone Circuits............................................................................................83

6.3.3 PCBFabExpress...............................................................................................84

6.3.4 Final Decision..................................................................................................85

7.0 Testing..........................................................................................................................85

7.1 Chassis...................................................................................................................85

7.2 Speed Test..............................................................................................................86

7.3 Battery Life............................................................................................................87

7.4 Battery Charging....................................................................................................88

7.5 Video Camera........................................................................................................88

7.6 PCB Testing...........................................................................................................89

7.7 Wireless Control Testing........................................................................................90

7.8 Sensor Testing........................................................................................................90

7.9 Microcontroller Testing..........................................................................................91

8.0 Bill of Materials and Responsibilities..........................................................................92

8.1 Bill of Materials......................................................................................................92

8.2 Responsibilities.......................................................................................................95

8.3 Project Milestones...................................................................................................96

Appendix A: Bibliography.................................................................................................97

Appendix B: Permissions.................................................................................................101

Appendix C: Index of Figures..........................................................................................108

Appendix D: Index of Tables...........................................................................................110

**1.0 Executive Summary**

The major focus for this project is to serve as a search and rescue robot which can be implemented in the field in dangerous, disaster-like situations. This robot is to serve as a means of making human involvement in dangerous search-and-rescue operations much more safe, thus enabling such operations to become much more successful.

To accomplish this task, we will be using a variety of sensors, such as temperature sensors, pressure sensors, as well as a camera and a rangefinder to help our robot “see” and assist with navigation. This data will be continuously measured by the various sensors and streamed to and manipulated by the microcontroller. This data will then be sent back to a computer so that it can be interpreted by the operator who will be able to see what is being captured by the camera, as well as a variety of other parameters about the area, such as the ambient temperature, atmospheric pressure, as well as distances to objects using the rangefinder.

We will be using an RF transmitter and receiver in order to communicate back and forth between the robot and the control unit, which will be the operator’s laptop computer. We will also have measures in place such that the robot will be able to controlled using the laptop’s keyboard. This will allow the user to not only guide the robot where it needs to go, but also, using the data collected by the sensors on the robot, will enable the operator to determine whether it will be safe for a person to enter the area to complete the rescue operation.

The microcontroller, wireless communication assembly, sensors, and camera will all be mounted onto a remotely controllable vehicle, which we will be purchasing separately, and modifying to suit our needs. This will require us to completely gut the factory-built vehicle and essentially rebuild its electrical circuitry from the ground up, as we will not be utilizing the remote control that comes with it. That means that in addition to programming the microcontroller to handle the data being fed to it by the sensors and camera, we will also need to interface it with the vehicle’s motors such that we can manage to control it remotely via a computer.

In summary, the goals of this project include designing a search and rescue vehicle capable of detecting whether the conditions in an area are conducive to human survivability and relaying this information back to a remotely located operator. We intend to have a working prototype accomplish this task as efficiently as possible, consuming as little power as possible, while minimizing costs and resources for our group members. Upon successful completion of this prototype, such a robot could be used in an incredibly broad range of applications, from military reconnaissance operations, to aiding rescuers in areas ravaged by fires, floods, or other similar natural disasters.

**2.0 Project Description**

**2.1 Project Motivation and Goals**

When we first embarked on this mission to choose a project, we were rather indecisive in the beginning, as neither member of our group really had a solid idea of what exactly it was that they wished to accomplish in completing this senior design project. For a group as diverse as ours, with each member coming from different backgrounds with different areas of expertise, settling down and choosing one project on which to focus our efforts was a rather difficult task to accomplish. We had gone back and forth over many different projects, ranging from the simplistic to the overly complex. We discussed whether we wanted a mos1tly software-based project, or whether we wanted a more concrete, mostly hardware-based project.

Ultimately, we decided that we would like to have something that would be a split down the middle in terms of software and hardware, as we have two electrical engineers and one computer engineer. We therefore arrived at the conclusion that some type of robot would likely be the best fit for our group, as this would mean plenty of hardware and software work to go around for everyone, and we felt that this would be a great way for all of us to maximize our exposure to both sides of electrical and computer engineering.

Once we all descended upon the idea that a robot would be the best and most useful way for us to proceed, we still needed to decide what type of robot we would build. After a great deal of research, we mulled through the seemingly endless possibilities. We could move forward with a quad-copter type project, which seemed to be quite popular at other universities. We could have gone with something that could help automate some mundane everyday task around the home, such as retrieving a beverage from a refrigerator. We very well could have even designed a robot for entertainment purposes, such as for a game or something similar.

After many sessions of lengthy discussion on what we would be designing, we finally settled on the idea that we could design and implement something for the greater good: a search-and-rescue robot. We felt that proceeding with such a project would allow for a great deal of freedom in our design and implementation, while also exposing us to a vast array of design techniques, as well as the product development lifecycle as a whole. Designing something of this nature would give us each the ability to focus on our interests and goals, while allowing us to work together to collectively design and build something that could one day prove to be helpful in real- world applications, especially given the extent to which robots are used today. This project would not only provide us with the opportunity to test and hone our design and implementation abilities, but it will also provide us with invaluable experience which will be necessary for a successful career in engineering.

As two of our members currently have fairly close connections to the defense industry, it seems logical that this is the direction in which we would be proceeding, as there is already a presence of these types of machines on the market, and thus gives a large pool from which to draw inspiration, as well as gain relevant, useful knowledge on how to start designing and constructing this robot. We hope that with this project, we will be able to accomplish the goals that we have set before us, including the successful interfacing of the microcontroller with the various sensors, as well as successfully managing to network all the wireless communication in such a way that the data will be able to travel in such a manner that the operator will be able to make use of real-time data and video being streamed from the robot.

Apart from the technical goals that we had in mind for this project, we truly wanted to create something that could benefit mankind. For example, whenever an earthquake strikes, or whenever an area is destroyed by a hurricane or tornado, people risk their lives to aid in the search and rescue operation. With the aid of something like our project, we could assist in helping to speed up the operation by locating victims and survivors. We could also use our robot to locate potential safety hazards, such as downed power lines and fires so that people proceed with caution as necessary.

**2.2 Objectives**

Search and Rescue robots are a topic of both wide discussion and importance in today’s society. With catastrophes such as the tragedy of September 11th, 2011 and the recent tornadoes in the Midwest, search and rescue robots can be of vital and beneficial use in aiding and guiding rescue workers through disaster zones. Search and rescue robots might be deployed in disasters such as mining accidents, urban disasters, hostage situations, and explosions.

These robots can accomplish tasks on the rescue sites where either human or canine resources are rendered powerless, or where the situation might be too dangerous for a human to enter. Other benefits of such robots include reduced personnel usage, reduced fatigue of rescue workers, and gaining access to areas which may be unreachable by either human or canine.

The first major objective of DZERV is to be able to deploy it into an area where neither human nor canine could gain access to, whether it be due to severe conditions or scabrous terrain. Aside of simply gaining access to such areas, DZERV should be able to retrieve some sort of useful information. Such information could come in the form of video feed locating live victims which need aid, or sending back sensor data, informing rescue workers of issues such as extreme heat or fire, hazardous environments, or possible explosions.

The second major objective of DZERV is to transmit video data back to the rescue worker’s computer station. Such a feature could work for numerous other applications besides search for live victims. The use of video feedback could be specifically useful for bomb detection in various situations. Take for example military checkpoints, while military officials are checking identifications, other officials could be operating the robot from a remote location checking under cars for explosives or other dangerous devices.

**2.3 Project Requirements and Specifications**

Our project is to consist of three major components, a mobile platform, an array of sensors, and a wireless communication link. DZERV must be able to survey and gather information from its environment including temperature, humidity, and barometric pressure. Upon acquiring necessary sensor data, measured values will be wirelessly transmitted and displayed to the base computer in real time. Aside of transmitting sensor data, DZERV must also be capable of transmitting real-time video feedback to the operator at the base computer. Movement of the robot will be based on commands sent over a wireless link from the operator’s base computer. Essentially, DZERV will send both video and sensor data the to the base computer, while the base computer will send signals to control the car. All of this will be done wirelessly.

As previously mentioned, DZERV should send temperature, humidity, and barometric pressure values retrieved from its surroundings back to the user operating it from the base computer. Ideally, DZERV should be placed in areas where living human rescue workers cannot gain access to. Due to the conditions and environments which DZERV will be operating in through human control, it is necessary to implement some form of “eyes” on our robot. For our application, we will be mounting a wireless camera onto the robot. This will allow the user operating DZERV to know not only the area the robot is operating in, but also whether or not there is a potentially dangerous situation at hand. Aside of such benefits, mounting a wireless camera onto the robot will also alleviate some pressures from the operator. The operator will not need a direct line of sight with the robot to guide it, rather, he/she can maneuver the robot using the video feedback through the camera software installed on the base computer.

Another aspect of DZERV are the sensors mounted onto it. We must carefully consider the wireless transmission of sensor readings back to the operator at the base computer. The operator must be able to be operating the robot form the base computer and have dependable real-time data being retrieved from the sensors. Such information must be real-time in order to allow the user to make active decisions about the environment that it is in. Such real-time information could lead to possible life or death situations in a harsh environment such as a fire or explosion.

We must also take into consideration obstacle avoidance and proper maneuvering. Due to this, we will be mounting an ultrasonic range finder onto our robot as well. By implementing a range finder, we will be able to properly avoid obstacles which could impede the movement of the robot. In the case of our applications an ultrasonic range finder was deemed more useful and beneficial than a bumper switch or limit switch. Unlike a bumper or limit switch which alert you only when they have been hit, the range finder will alert us to an obstacle in its path prior to collision, thus not only allowing the user time to properly and safely navigate around obstacles, but protecting the robot itself. Table 1 below provides a brief overview of the requirements for our project.

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Requirement | Description | Testing |
| 1 | Communication  range of 75 ft. | Communication  from TX should  send to RX from  100 ft. | Drive DZERV  until  communication  failure and  record distance. |
| 2 | Be able to  provide  audio/video | DZERV must  send real-time  audio/video to  base computer. | Check the live  feed of the robot  as it operates. |
| 3 | Battery life of  approximately  30 minutes |  | Run the motor  until it dies  while recording  the time. |
| 4 | Be able to  navigate  multiple types of  terrain  (pavement,  grass, etc.) | DZERV should  be prepared  for whatever  terrain it  encounters. | Drive the robot  over various  terrains to  ensure it can  move  efficiently. |
| 5 | Must weigh less  than 15 lbs. | DZERV should  be light-weight. | Measure the  robot on a scale. |
| 6 | Must maintain  low enough  temperature | There should be  no failure of  electronic  equipment. | Use a  temperature gun  to measure the  robot while it  operates. |
| 7 | Simple to use  GUI | The operator  should be able to  control DZERV  with ease. | Find someone  unfamiliar with  the project and  have them test it  and listen to  their feedback. |
| 8 | Portability | Aside of being  lightweight, it  should be easy  to transport. | Attempt to make  the project a one  piece robot. |
| 9 | Auto  notification | The system  should alert the  operator if a  condition that  needs attention  arises. | Place interrupts  within the GUI  code and test  dummy  conditions to see  if they pop up. |

Table 1: Table of Requirements

**2.3.1 Risks**

Just like any other project or task in life, we must prepare to deal with possible risks. Since we are all full time students and two of hold demanding internships, we must prepare for risks. One risk separating us from other groups is the fact that we only have three group members. This is a positive aspect in the sense that our project will not become overcomplicated. However, we will have more research and design work to accomplish than a group with four members. Another broad risk is the fact that none of us have ever even attempted to embark on such a mission or really held any experience designing, building, and testing a robot. More specific risks are involved with the robot itself. What if a sensor electronically fails? What if the user interface does not properly receive values? What if we miscalculate power and current, solder a PCB together, and a component busts? These are all things that we must prepare for by making sure we triple check all of our work and calculations as we go. Of course, like any other project, we will have some bugs and kinks to work out, but we should prepare for major risks.

**2.3.2 Team Management**

Since the original formation of our group, we all decided that this would be a collaborative effort, and that we would all take equal responsibility in our project by having no set leader. This was strongly influenced by the fact that none of us had ever been faced with such a feat, so we decided tackling it together equally would be an optimal solution. Despite having no team leader, we all took initiative to make sure we were all kept well informed and up to date as to what everyone was working on. We scheduled regular weekly meetings where we would discuss what he had researched/worked on/etc. Doing this insured we were all kept up to date and that no two people were completing the same task.

For our documentation, we chose to utilize Dropbox, a virtual storage cloud, and have one main document, where each team member could write updates kept on personal machines to. We also chose to divide the documentation into small portions that each team member could individually manage. When splitting up the research and documentation, we decided that each team member was to choose on either what they were interested in working with, or had prior experience or knowledge of. If we were working on a section in which we expressed interest in or held knowledge in, we felt our productivity would be at an optimal level.

**3.0 Research Related to Project Definition**

**3.1 Existing Similar Projects and Products**

At the time we had sat down to discuss and finalize our project plans and goals, neither one of our group’s members had a great deal of experience working with much of the technologies which we’ll be using to accomplish our objectives for this project. It therefore became a necessity to spend a great deal of time researching in order for us to acquire the knowledge necessary to face the challenge before us. Much of the challenge in completing this senior design project will be from absorbing and assimilating information from the work of groups and companies who have traveled this path previously. Fortunately, there are countless projects and products similar to what we are planning to accomplish from which we will be able to draw inspiration.

**3.1.1 Stevens Institute of Technology Search and Rescue**

One such example was a project built by a group at the Stevens Institute of Technology in New Jersey. This particular group had come to nearly the same conclusion that we did, in that a robot of some sort would be a great way to combine the disciplines of computer, electrical, and in their case, mechanical engineering. Their particular design included some of the things that we intend to use, such as a camera, a rangefinder, and various other sensors. One of the major differences between that project and the one that we intend to implement lies in that particular group’s decision on how to proceed with actual construction of the robot. They had contemplated constructing their own chassis for their robot. This is something that our group neither has the experience and technical know-how, nor the desire to do, as this would complicate our design greatly, and would also likely greatly increase the time and costs involved in prototyping and manufacturing. This is also something that could lead to issues with scalability at a later time during the actual building of the working prototype, as this could be a drain of quite a bit of resources for our group members should we find that what we had constructed was not exactly something that we really, truly needed or wanted. However, the Stevens group members decided that purchasing a robot kit from a manufacturer called Vex would be the best way for them to proceed. We have decided that this too would also complicate things for us due to some of the design constraints for our Senior Design class, and so we will therefore not be purchasing a robotics kit in order to build our project. This group also intended to make their robot autonomous with in that it could navigate on its own

using some path prediction algorithms in conjunction with some limit and bump switches located on the vehicle, for collision detection. While this would make our robot a far more effective search and rescue vehicle, we feel that this would increase the complexity of our project immensely, as none of our members have much experience with programming for Artificial Intelligence. Furthermore, they also included a passive infrared sensor to detect changes in temperature to indicate the presence of a human or other animal in the area. While we had given this some thought at first, we decided that we would rather not include such a sensor in our design. Furthermore, they had also intended on building their own obstacle course on which to test their device. Time and budget permitting, such a requirement may be something that we can achieve, but we do not expect to be doing this for the testing of our final prototype. Overall, this project seemed like a great source off which to base our design, but we still had quite a few unanswered questions, and so further research was required.

**3.1.2 OSU Search and Rescue Robot**

Another project that we had encountered during our research was for a search and rescue type robot made by a senior design group at Oklahoma State University. This project, from which we have based much of our design, has quite a bit of detailed information, which has actually helped us a great deal for the theory behind the functionality of our project. This group had essentially the same idea as the group from Stevens, in that creating a search and rescue robot would be a great way to integrate the disciplines of electrical and computer engineering. Unlike the group from Stevens, however, these students decided against purchasing an actual robotics kit like the Vex robotics kit that the Stevens students purchased. However, they also did not venture to build their own robot either, as this would again greatly increase the complexity of the project with the given time constraints with which they were working. Therefore, they ended up purchasing an already working remote-controlled vehicle to serve as the mobile platform for their robot upon which everything else was added or built. This is the route which we will also be taking, as this simplifies our design by quite a bit, and allows us to focus more of our time on what really matters, which is interfacing our hardware and software to work properly together. In addition to using an already working mobile platform, the members of the OSU group went one step further and created a few inventive ways to control the vehicle. They not only created their own remote control for the vehicle, but they also created a program such that the vehicle could be controlled via user inputs on a laptop computer’s keyboard. Furthermore, they also came up with a very creative way to control the vehicle via voice commands. However, as the idea behind this vehicle is that it should be able to be operated remotely, it seems rather infeasible to include such a feature, as this would require the operator to be near the vehicle, unless through the use of some peripheral device, the operator could transmit their voice commands directly to the

search and rescue module, which would also increase the complexity of the project. Apart from these features, this project is essentially quite similar to the Stevens group’s project in that there is a camera and various sensors, which all communicate wirelessly with the operator’s base computer in order to transmit data remotely and receive commands. One major difference between this project and the Stevens project is that while the Stevens group built their robot for autonomous control, the Oklahoma State group did not, and actually intended for their robot to be controlled by an operator at all times. That being said, the technical design for this robot was actually a little complicated. Their PIC microcontroller allowed five ten-bit-wide channels for Analog to Digital Conversion. Because more than one timer would need to be used in order for each sensor to be able to handle measurements in real time simultaneously, they needed to utilize MOSFETs to act as switches between the different devices so that they could all interface with the microcontroller’s Analog to Digital Converters in order to get the data back to the user in real time. The PIC processor that we intend to use only has three timers on-chip, and we will therefore likely need to implement a similar design in order to ensure that everything works as intended so that we can take measurements as accurately as possible. Radio Frequency (RF) communications played a major role in this group’s design, as this is how they were able to successfully have the data sent from the mobile robot platform and received at the operator’s base computer. The communication aspect of a search and rescue robot is perhaps one of the most important factors in its completion because without a proper means of communication, there is no way for the data, video, or any other type of information to be sent back and forth between the base computer and the robot itself, which would not be very conducive to a successful search and rescue operation, or any other type of application where such a robot could prove to be useful. Our decision to use RF as a means of communication, which will be discussed later in the Design Details section, was loosely based on this project, as well as many others, and as we have discovered, will be much simpler to implement than other types of communications protocols.

**3.1.3 Autonomous Ball Collector (A.B.C)**

Another project that our group members investigated briefly was a project done by a group of University of Central Florida students back in 2010. Their project essentially was an autonomous robot which was used for seeking and collecting tennis balls on a tennis court. Though this project is not necessarily the direction in which we are planning to go, we were still able to gain some inspiration from the work that this group had done. This project was certainly on the more software-intensive side, as there was quite a bit of artificial intelligence to program into this robot, as it had to determine whether it was collecting tennis balls, it had to know where to position itself to do so, along with a variety of other functions that were performed autonomously. However, as previously mentioned, autonomy is not the route which our group is planning to take, due to the overall complexity that this additional requirement would bring to our project. Additionally, the A.B.C. team fabricated their robot entirely from scratch, which again, is something we will not be doing, as this will add significant cost and man-hours to our overall design, which would be better spent on mission-critical items, such as the microcontroller or the user interface. The main inspiration we’ve drawn from this project was their in-depth analysis of various types of wireless communications. Just as they had done, we also have investigated into the possibility of using Radio Frequency, Bluetooth, IEEE standard 802.11 Wi- Fi, and even infrared wireless, which we will again be discussing in our section on communications. To implement their design from an electronics standpoint, they utilized an Atmel microcontroller. This worked to their advantage for a variety of reasons. First, their members had very limited programming ability. The fact that this microcontroller could be programmed in C meant that they could use the limited knowledge they had with the C language as opposed to having to learn a completely different language, which would ultimately, ideally, save them time in that they wouldn’t have to struggle with learning a new programming language and could get to work in the controller straightaway. Additionally, they were able to find a variety of pre-written libraries that would be of great assistance to them once they actually began working with the controller to interface all their components. They also used two additional Atmel microcontrollers to control the various other functions of their vehicle and to simplify the design for their group members. We have found that such an arrangement will not be necessary for our robot, as the controller we will be using will be sufficient to handle the things that we need it to do. More on that will be discussed in our section on microcontroller selection. After reviewing this project, our group members decided that this was in reality, quite different from what we were attempting to accomplish, and the amount of inspiration we were able to draw from this machine was actually quite limited. While we did gain a bit of insight as to how we can get our communications for our own project working, much of what we found with the project by this UCF group is unfortunately, not applicable for our purposes. For this reason, we felt the need to continue investigating into other projects and products.

**3.1.4 Other Available Products**

When we sat down to begin performing research for our project, we knew that we would definitely be able to draw inspiration from things already available on the market, as search and rescue robots are not a new technology, and have been employed by various military branches, police departments, fire departments, as well as other agencies, for quite some time. As such, we found a vast source of knowledge by looking into these various applications of real-world applications of search and rescue robots.

**3.1.4.1 Unmanned Aerial Vehicles (UAVs)**

We are seeing rapid growth in the field of unmanned aviation, especially for military purposes. More and more, we are hearing reports of strategic enemy targets being eliminated by these unmanned aerial vehicles. Similarly to other autonomous robots, these can be pre-programmed to fly along certain paths. However, they are not completely autonomous, as they are for the most part controlled by a human operator located in a remote command center.



Figure 1. An image of an MQ-9 Reaper

## (Reprinted from Wikipedia under the Creative Commons Attribution-ShareAlike 3.0 Unported License)

## The above is an image of a General Atomics MQ-9 Reaper, or a Predator B, as it is most commonly referred to. This UAV is perhaps one of the most recognizable UAVs on the market, as these are the Predator drones that are often alluded to in the news stories about targets having been eliminated. However, these drones were primarily first developed b to aid intelligence agencies in their operations in gathering intelligence and other information about certain areas. Because these types of aircraft were originally developed for the same reconnaissance / search and rescue operations, we decided that some research into these aircraft would have some merit. While we will not be developing anything near the scale or magnitude of one of these types of aircraft, it would, in theory, operate under the same basic fundamental principles of our own search and rescue robot. For example, the Predator is remotely operated, and is equipped with various sensors, cameras, and is able to process and transmit this information back to the operator at the control post on the fly. This is similar to our own robot in that it will perform the same basic functions of reconnaissance, and the user will be able to see this data being transmitted from our robot in real time as well. While the underlying principles for how the UAV and the operator communicate wirelessly may be substantially more complex than what we intend to do for our robot, the same idea of controlling the device remotely is still there, and the same considerations must be accounted for. For example, what kind of range would we look to have on such a device? What types of frequencies would we be operating with? Questions such as these must be answered for the development of these unmanned aerial vehicles just the same as they are being answered for our project. The very fact that this technology is still relatively new, and the fact that it has revolutionized the way reconnaissance, and even combat are carried out makes an incredibly strong case that this type of machinery is most certainly paving the way to the future. It is reassuring for our group members that the type of robot we are designing has some real-world use, and inspires us to ensure that the utmost care is taken throughout this process so that our machine can have some semblance to these incredible machines.

## 3.1.4.2 Kinect Robot

Engineers at the University of Warwick in Coventry, UK have accomplished great some very remarkable things in the field of search and rescue robotics. Using a Microsoft Xbox Kinect, they have managed to implement an incredible powerful search and rescue robot capable of mapping its environment and giving the operators a greater amount of flexibility in what they can do with the robot. For instance, using the Kinect module, the robot can create a 3-D model of the area and allow its operators to get a better feel for the environment and also assist in pointing out possible locations for survivors. With this robot, the engineers from the university won first place in the European Robocup Rescue championship in 2010. The most remarkable thing about this vehicle is that despite the level of detail and sophistication that went into its construction, it is still remotely operated by a human being, and is fairly close to what we are looking to do with our project. The Kinect module acts as a camera of sorts, and also allows this robot to map the area it is in. With an accurate model of the area, rescuers can use the robot to search out survivors from the safety of a remote location much more effectively without needing to worry about risking their lives by entering a dangerous building with the possibility that there may be nobody there. Similar to what we intend to create, this vehicle is not really meant for actually assisting in the removal of survivors from the disaster area, but is meant to serve a pivotal role in actually finding survivors and helping rescuers in searching for them. While we will not be using the camera mounted on our vehicle to create a three-dimensional map of the environment, it will be serving as our primary means of exploration and our primary means of detecting the presence (or lack thereof) of any survivors/victims. Additionally, their unit is equipped with an infrared camera for the detection of body heat as well as a CO2 sensor to detect the presence of expelled CO2 gas, which would indicate the presence of a living, breathing human being. We actually did consider both of these as viable additions for our robot in our initial planning stages, but we felt that neither was truly necessary, and we therefore decided not to proceed in purchasing these parts for our robot.

Even after investigating all these projects and inventions, we were still left with many, many unanswered questions. How are we going to get all this working together? How do we bring all the hardware and software together to create a fully- functioning robot? How do we make control this robot wirelessly without the use of a remote control? What can we do to ensure that our search and rescue vehicle will work properly? These questions, along with dozens of others, are all questions that needed to be answered before we could finalize our design plans and actually begin looking into the construction and testing stages of our project. Fortunately, with the help of many internet knowledge sources, as well as the help of many skilled individuals, we were able to have most of these questions answered, or at the very least, we were given a little bit more knowledge than we had previously, which served as a great place to start.

**3.2 Relevant Technologies**

When we first sat down to begin researching things for our project, one of the largest factors that helped us determine the direction in which we wanted to move was the availability of hardware and knowledge about the hardware/software that we would need in order to complete the task that has been laid before us. Among the first things that we researched were forms of wireless communication.

**3.2.1 Wireless Protocols**

When we first decided on what we wanted to do with our project, we had absolutely no idea what possibilities there were on the market for us to use in terms of communications. We were aware of the existence of things like Wi-Fi, and RF communication, but we had no idea what they really were or how they work. This portion of our paper will delve a little deeper in to each of the technologies we researched, and how they work or could be incorporated into our design. We’ll first begin with an investigation of Wi-Fi.

**3.2.1.1 Wi-Fi (IEE Standard 802.11)**

Much of the beginning of our research stemmed from the little bit of wireless communications that we did know. We knew about Wi-Fi, and were aware that it was used in the creation of things such as local area networks (LANs), but we were unsure as to how exactly this technology worked, or how we would be able to utilize it in our design. After a great deal of digging and research, we were able to solve this mystery and gain a much more detailed understanding of how Wi-Fi works.

Wireless Fidelity, as it was first called, much like many other forms of wireless communication, functions by the way of radio waves. Simply put, Wi-Fi works in a manner very similar to the way in which two-way radios or walkie-talkies communicate. Essentially, a computer converts digital information into analog radio waves and sends this data to a router via an antenna located inside the computer or laptop. This data must then undergo analog-to-digital conversion at the router level, which is then transmitted to the internet via a wired Ethernet link down on the router level of things. This also must happen when data is being moved back from the internet through the router to the computer, but in the reverse order, with the digital to analog conversion taking place at the router, and then the analog to digital conversion taking place at the computer.

There are many flavors of the IEEE 802.11 standard, all of which have very different applications, ranging from use inside buildings to use inside moving vehicles. The one that would likely be the best fit for our search and rescue robot would be the 802.11b standard. It is important to note that for long time, this was the least expensive standard to use, but also is the slowest of the 802.11 family, operating in the 2.4 GHz frequency band with a baud rate of about 11 megabits per second. While this clearly would not be a very good choice for a user’s main internet connection, something like this would definitely be more than able to satisfy our purposes in getting this project completed. After further investigation, we discovered that we could even implement a Wi-Fi network using a simple embedded device that we’d be able to place right on our printed circuit board, which would save us a great deal of space, as well as help us minimize our costs, as these units are actually relatively inexpensive. Additionally, such a unit would be able to communicate with a computer via the serial port, which in turn could help simplify our design a bit.

We did investigate into some of the advantages and disadvantages of using 802.11 as our form of communication, as demonstrated in Table 2 below.

|  |  |
| --- | --- |
| Advantages | Disadvantages |
| Long range | Relatively Slow Data transfer rate |
| Low Cost | Slight signal interference possible |
| Signal not easily blocked | High Power Consumption |
|  | Network Configuration can be  Complex |

Table 2: Pros and Cons of 802.11

The fact that Wi-Fi modules do consume a great deal of power was somewhat of a cause for concern, as one of the major goals for our project is to minimize the power consumption for our robot as much as we possibly can. After researching this technology, it became clear to our group members that proceeding with Wi-Fi technology would most likely not be the way to go because of the power consumption requirement that we have imposed on this project. While Wi-Fi would serve as a great backup idea, we clearly needed to take some more time to gather a little more data before we could finalize our plans.

**3.2.1.2 Bluetooth**

Another great resource for us to investigate was Bluetooth technology. This is yet another wireless communication standard that could actually be quite useful for our project. Similar to 802.11, this standard also relies on the transmission of radio waves in order to send and receive data. The major difference is that where Wi-Fi operates on the basis of Wide Area Networking (WAN), the Bluetooth protocol operates on the concept of piconets, which are much smaller networks, as a Bluetooth module’s range is actually much less than that of Wi-Fi, at about ten meters in all directions from the device. Similar to Wi-Fi and many other technologies, Bluetooth operates in the 2.45 GHz frequency range. While this protocol does share the same frequency spectrum as many other types of devices and protocols, the way in which Bluetooth operates actually makes it quite robust. This protocol operates using what is known as spread-spectrum frequency hopping. This simply means that the Bluetooth transmitter will change frequencies many times per second to ensure that it does not interfere with other devices in the area. The big advantage to this, is that if the device does end up running on a frequency that is already being used by another device, it will not stay on that frequency long enough for it to cause interference, and even if any interference were to take place, its effect would be negligible.

The reason that such a device could be useful for our needs is that we could then use a wireless device to control our robot remotely. Because it would constantly be changing frequencies, there would be less of a need to worry about interference with the other components that we’ll be using, as we’ll need to not only control our vehicle wirelessly, but we’ll need to ensure that the data we’ll be measuring is being transmitted and received properly as well. Essentially, we would have a Bluetooth module connected to our vehicle, which would communicate with either the Bluetooth module embedded into the operator’s mobile phone, or even their laptop computer. From there, the operator could use either the user interface we are developing to manage and control the vehicle. From the research we have done, we have constructed table of the advantages and disadvantages of Bluetooth.

|  |  |
| --- | --- |
| Advantages | Disadvantages |
| Spread-spectrum frequency hopping | Very short range |
| Little radio wave interference | Relatively slow data transmission speeds |
| Very low power consumption | Pricing is not very cost-effective |
| Deployable across a wide range of devices |  |
| “Piconets” are easily configurable |  |

Table 3: Pros and Cons of Bluetooth

Though our research into Bluetooth technology offered us yet another great option for how we will have our communications working, we still felt the need to look into other various means of communications. This in turn would provide our group members with a much more broad knowledgebase so that we would be better able to make a decision with what we feel will be the best method for communication for our project. The reason that this is such a difficult decision is that the way we choose to do communications for our robot will greatly influence the approach that we’ll need to take in actually building and testing our model. We next investigated the applications and utility of infrared communications as a possibility for our robot.

**3.2.1.3 Infrared Communications**

The reason that we looked into infrared as a means of communication for our robot is the fact that infrared is so popular in modern electronics. The infrared portion of the electromagnetic spectrum has countless applications in modern science and engineering, but we limited our investigation and discussion to its applications in the realm of communications.

The concept behind infrared (IR) communication is relatively straightforward. On the transmission side of things, an infrared-based light-emitting diode gives off infrared radiation. This radiation is then focused into a small beam that is modulated via a pulse width modulator in a way that the data being sent gets encoded using either the RC-5 or SICRC IR communication protocol. On the receiving end, the infrared energy is sent into a photodiode, which generates an electric current, which is then sent through the receiving device and decoded such that the data ends up in the right spot.

Unfortunately, the more research we did on this type of technology, the less and less it appeared that we would be able to use this as a viable method of communication for our project. For one thing, we discovered that due to the light-sensitive nature of infrared communications, that it would not be a feasible means of communication for outdoor applications. This would be an issue for our search and rescue robot, as a majority of its use and testing would occur outdoors. This would, however, be perfect for something that would be used strictly for night time. Another disadvantage that we found is that devices operating via IR communications work on a “line of sight” basis. This means that in order for devices to communicate using infrared energy, there must be a direct path between the devices. As an example, this works perfectly for the remote control for a television, because there is a direct line of site between the remote control and the television while they are in the same room. If one were to attempt to change the channel on the television from another room, they would find that this clearly would not work, due to the lack of a direct line of sight. For applications like this, this is perfect because it limits interference significantly, as this infrared radiation is unable to penetrate walls or interact with other devices.

For our purposes, however, this would prove rather ineffective. Because our intention is to have this vehicle controlled remotely from a location where a direct line of sight between the operator and the vehicle is most likely not an option, this option appears to be rather infeasible. Despite all the advantages that IR can provide, it is likely that we will not be able to use this for our project.

**3.3 Strategic Components**

One of the most difficult stages of the design process is actually choosing which components will be used. This is especially true with a group of senior design students, who all have very little experience in comparison with the components engineers that large corporations have access to. That being said, we spent a great deal of time doing research on the kinds of components that we would need to complete our project. We found this research to be invaluable in that it taught us such a great deal about component selection for our own project. It also gave us insight into how this process is done in the industry, and we have all benefitted tremendously for having done this research together.

**3.3.1 Mobile Base**

After deciding that we wished to make a robot, we then needed to decide how exactly we would implement this robot. Would we create it all on or own? Would we design a chassis for it? Would we do all the welding for the robot? How would we pay to make something like this turn into a reality? These were all questions that we had in our initial design meetings. After taking a look at other projects where the groups designed their own robot chassis, it seemed like a great idea, and would be an incredible accomplishment for our team members, but none of us had any experience with robot design. Our group members had neither the technical know-how, nor the tools and experience to make something like this happen. Additionally, in determining an initial budget for our project, this idea seemed rather costly. In addition to the other components we would need to buy for our robot, we would also have to worry about paying for parts and materials for the robot chassis as well. Due to our lack of experience, we would likely need to also pay some sort of costs for machining, along with many other related expenses. We agreed that this would probably not be a viable way of proceeding to do the massive costs that would be incurred just from constructing the chassis alone. Also, due to the time constraints with which we are working, we decided that this in and of itself would take much too long for us to be able to complete our project on time. We then began to look to other sources of inspiration for our project.

After dawning on the realization that creating our own robot from the ground up would be much too risky from a financial and temporal standpoint, we came to the agreement that this simply would not be how we will be proceeding. It would simply complicate things, make it much more expensive, and would more than likely not leave us enough time to actually focus on getting the major aspects of the project finished. After leaving the idea of building our own robot behind, we decided that we would need to look into other alternatives for our project.

Because of the need for a mobile platform for search and rescue purposes, we then decided to look into using remote-controlled vehicles. There are an incredible amount of remote-controlled vehicles on the market, ranging from boats, to airplanes, to cars, to tanks, among a wide variety of others. We decided that we’d like to have something that is ground-based, and therefore ruled out the boats or the airplanes. We then needed to finalize what we are going to be working with. After more research and planning, we decided that it would be best if we just purchased a simple remote- controlled car from a manufacturer such as Mattel. This way, we can focus more on the things that are more important, such as communications, the sensors, and our user interface.

This train of thought has brought us to our current state. We will be purchasing a toy remote-controlled car, and will be removing its internal parts such that once we are finished, our printed circuit board, sensors, camera, and everything else will be able to fit inside and be moved by this mobile base of operation. We all have come to the agreement that with this, we will be much better able to concentrate on the important parts of our project, which are all the electrical components, as well as the software.

**3.3.2 Surface Mount vs. Through-Hole**

While investigating components, one of the largest factors in determining which would be the right parts for us, we needed to investigate the advantages and disadvantages of through-hole and surface mount components. The reason for this is that most manufacturers make quite a few of their parts in both styles, so it was important for us to know which would be of more benefit for us when we actually create our PCB.

After careful consideration and further research, we discovered quite a few things about the differences between surface mount technology and through-hole technology. For example, the advent of surface mount technology in the 1980s has caused the size of electronics to decrease incredibly. The fact that surface mount components are many times smaller than their through-hole counterparts means that the density of components on a printed circuit board can be increased dramatically, allowing for more components on a single board, and therefore vastly increases the overall capabilities of what can be done with a single board. In addition, we discovered that there a variety of other advantages of using surface mount parts over through-hole components. One example is that many surface mount parts actually dissipate slightly less power than through-hole parts. Also, many manufacturers make certain components only in a surface mount package. While there are many manufacturers who do make both surface mount and through-hole packages for their parts, the through-hole parts are starting to become a thing of the past as more and more circuit designers begin to go with surface mount parts. Apart from these few differences, however, we noticed not very many other differences between through- hole and surface mount technology. After much deliberation, we decided that we would do as much prototyping as possible with through-hole components due to the fact that they are much simpler to work with, as we have all had experience working with through-hole parts in various labs. For prototyping, the through-hole parts will be perfect because they are easy to place and remove, and soldering and unsoldering

would be much easier as well, which will make the prototyping process much more smoothly for us. Once we are ready to begin constructing our finalized prototype for demonstration, we will switch over to surface mount components. Because we will be working with a rather limited amount of space, we feel more and more that the space- saving qualities of surface mount parts will be invaluable to us.

**3.3.3 Microcontrollers**

Of course, perhaps the single most important component for any piece of embedded electronic equipment these days is the microcontroller. This piece of circuitry is actually what will be serving as the brain of our operation, as it will be handling all the data that we will be utilizing. It will serve essentially as the command center for our robot. This is where all the programmed information will be stored, that will tell the controller what to do with the data being sent to it by the various sensors. It is the piece that will communicate with the transmitters and receivers to get the data back and forth between our base computer and the vehicle. This component will be what takes the information from the sensors, manipulates, and turns it into something that will make sense for the search and rescue team operating our robot. The microcontroller will be also be processing the commands coming from our base computer to control the vehicle’s movement. Needless to say, the microcontroller we choose to use will have a significant amount to do with the success of our robot, and so it is imperative that we make this decision very wisely, as choosing the wrong controller for the job could lead to catastrophic failure. More specific details about the selection process for our microcontroller will be provided in the microcontroller section of our hardware and software description.

**3.3.4 Camera**

Perhaps the second most important feature for our project will be the camera. Whereas the microcontroller will be serving as the “brain” for our robot, our camera will most definitely be serving as the eyes and ears for the robot. The camera is what will allow the operator for the robot to see the world as the robot sees it from a remote location. The camera is also what will allow the operator to see where to guide the vehicle. That being said, it is clear that the camera is perhaps the second most important component for this project, if not just as important as the microcontroller. Because of the significance of the camera for our robot, we knew that great care needed to be taken in its selection because the camera, like the microcontroller, can either make or break this project. We therefore did quite a bit of research on cameras, ranging from small embedded surface-mount cameras, to the very expensive cameras with features including built-in rangefinders. After a great deal of research and healthy debate, we decided to stick with a small, relatively inexpensive IP camera that would allow us to transmit the data over a wireless internet network, instead of needing to purchase an additional transmitter/receiver pair for this as well. Due to the massive importance of this component, we feel that the research that we have done for this component was adequate enough for us to choose something that will provide us with a high level of success for this project.

**3.3.5 Sensors**

Another extremely important aspect of this robot’s design would be the sensors. We spent a great deal of time researching into exactly which types of sensors we would like to include in our robot. Would we be including an ultrasonic sensor to pick up on sounds in the area? Would we include in infrared camera? Will we be in need of a temperature sensor? All these questions, we were able to answer after investigating various other similar projects. After a bit of discussion and further investigation, we decided that including an ultrasonic sensor would be rather superfluous, given that the camera that we are going to be using has a built-in microphone, which would render an ultrasonic sensor unnecessary. After more investigation, we also decided that going with an infrared camera would also likely not be the greatest of ideas. We decided this based on the fact that we are already going to be dealing with tight spacing constraints, as well as budgetary constraints. We therefore decided against purchasing one of these in favor of going with a more robust camera with quite a few more capabilities. We ended up deciding that we would try and limit the amount of sensors for our robot for a few reasons. First, we did not want to make this task overly complicated, because adding more and more sensors would greatly increase the complexity of the circuits we’d need to build, as well as the cost for our printed circuit board. We also simply just did not feel the need to overcrowd our PCB with dozens of unnecessary sensors, as again, this would drive up our costs because these additional sensors would give rise to the need for additional copper traces, which would lead to increased PCB costs.

We therefore decided on a few basic components that would help us get the job done. We decided that we would certainly be in need of a temperature sensor. This will help our robot determine the ambient temperature of the surrounding area, which will help the operator determine whether it will be safe for rescuers to enter the area to help seek out survivors. Much more detail will be given on how this particular component was chosen will be given in the Sensors subsection of our hardware details section.

In addition to the temperature sensor, we also decided to include a pressure sensor for the robot. The rationale behind this is that our vehicle would be able to detect an unsafe rise in air pressure in the area as the result of an explosion. While this would certainly be easy to see/hear with the use of the camera, it would also be helpful to have empirical data taken in the field to see the kind of effect this has on our vehicle. More detail on the actual selection for this item, as well as how it will operate and how we intend to have it connected to the microcontroller can be found in the Sensors subsection of the Hardware description.

We also decided that a rangefinder would be an essential piece of equipment to include on our robot. This actually is useful for a variety of reasons. First, this will help us determine how far away our vehicle is from potential survivors. Secondly, this will also let us know how far away from certain obstacles our vehicle is so that the operator can make the necessary adjustments in controlling the vehicle such as to ensure that the obstacles are avoided. We actually spent a great deal of time researching this, knowing it would play a major role in our operation and came to the conclusion that an ultrasonic rangefinder would suit our needs perfectly. As with the other sensors, a more detailed discussion of this part’s operation will be given in the Sensors subsection of the hardware description.

**3.3.6 RF Transmitters and Receivers**

In order for our robot to function properly, we would need some way of having the data, video, and commands from the computer to all be transmitted wirelessly. We looked into a great variety of technologies for this. We at last decided on Radio Frequency (RF) technology to help us accomplish our task. One of the major things we knew we would need is a way of transmitting data from the vehicle, as well as a way of receiving that data as input to our base computer. This is how the operator will be able to operate the vehicle, because this is how we will be receiving our video feed from the robot. This is also the way we would be controlling the vehicle, as we will also be sending data wireless to the vehicle as well in the form of commands to tell the robot which direction to move.

While the details of how we came to settle on radio frequency technology as the main means of communications will be discussed in our design section, it is important to note that we realized ahead of time that our communications will be the major challenge of getting our robot to work. We decided developing our own wireless communication protocol would be a senior design project in and of itself, and an incredibly difficult one, at that. We therefore opted to look into the other existing technologies and decided that RF would be the way that we would like to proceed.

**3.3.7 Power Supplies**

Because one of our major goals for this project was to have our robot consume as little power as possible, it is only logical that we consider our power supply to be a strategic component. Since this is what will be providing the necessary power for our vehicle, it is fitting that we take extra care in making this selection. Due to such a wide variation in the available power supplies on the market, we needed to do a great deal of research to find out which would be the best supply for our needs. The most difficult part of choosing our power supply was the fact that we weren’t very sure exactly how much voltage needed to be supplied to the various components as a whole. Thankfully, after studying dozens and dozens of components’ datasheets, we were able to determine how much power would need to be supplied to everything, and from there we were then able to do research into what kinds of power supplies to use, as there are an incredible variety of supplies available, and more work needed to be done to figure out which would be the best for our own applications. More information on this will be discussed in great detail in our design section.

**3.3.8 Base Computer**

Another incredibly important aspect of our project will be the base computer from which everything is going to be controlled. The reason this is important is that this is where the operator will be able to control the robot and tell it where to go and look. This is a relatively important part of our setup, as this is where the data collected by the robot will be transmitted so that the operator can see what is happening, as well as see the various sensor readings as measured by our robot. This in turn will allow the operator to react as necessary in getting the proper resources deployed to the affected area. We decided that the best equipment for the job would be one of our group members’ laptop computers, as this would ensure portability, as well as compatibility. One of the other factors that came into play in this making this decision was that we would need something that would work with all the parts that we’ll be using. This means that we’ll need something capable of communicating via a USB-to-serial port so that we can get the data to the computer wirelessly. Because none of us have a serial port on our laptop, we’ll need to look into purchasing a serial-to-USB adapter, which should be both relatively simple to find, and fairly inexpensive.

**3.3.9 Graphical User Interface**

While not necessarily a “component” in the physical sense of the word, another incredibly vital part of our project will be the graphical user interface that we’ll be using to make the robot simpler to use. The reason that this is an important part of our robot is that this interface is how the operator will see the world as the robot sees it. This is where all our sensor readings will be sent, and is also where the video captured by our vehicle is going to be displayed, which is what will allow us to issue commands and control it from our base computer. This is where we will see temperature, pressure, and other readings, and this is what will be allowing the operator to make the necessary decisions. We decided that the user interface would need to be detailed enough such that it would be user-friendly. This will allow just about anyone to operate the vehicle and make search and rescue operations more streamlined so that anyone can help out whenever necessary. Another functionality we would like for this to have would be the ability to integrate captured video into the user interface somehow. We had seen similar projects where this was attempted, but was not accomplished successfully. We would therefore like to be able to accomplish this, as this would make everything a lot more streamlined, and would be much nicer from an aesthetic point of view. Our graphical user interface will be discussed more in-depth in our discussion on interface in the design section of this document.

**3.3.10 Voltage and Current Regulation**

Another important thing for us to investigate was the idea of voltage regulation. How will we ensure that all of our components are getting the proper voltage? How will we ensure that no parts are drawing more current than their maximum rating? What types of technologies can we put in place to keep our parts from exceeding their rated values? The answer to these questions, quite simply, is that we would need to implement things that we have been learning about over the last few years that we have spent in school. We are faced with the issue of current and voltage regulation for our robot. We will now have the chance to implement things like voltage dividers and voltage regulators in a real-world application. At our disposal, we have components such as voltage regulators and current limiter circuits. Using tools such as these, and the skills that we have honed over the years, we will be able to ensure that all of our parts are operated within the limits of operation specified on their respective datasheets, as well as use this knowledge to ensure that they dissipate as little power as possible while operating.

**4.0 Project Hardware and Software Design**

**4.1 Microcontroller (PIC16F886)**

The microcontroller we utilize for our project is one of the most important assets and must be taken into careful consideration. The microcontroller we use will essentially serve as the “brain” of DZERV, reading in all of the sensor data, and sending that data out to the transmitter to be sent to the base computer’s operator. It will constantly be transmitting data to the computer’s receiver, as well as receiving data from the sensors. It must be able to correctly tie to and read data from all of the electronic components which we plan on utilizing. The block diagram below shows the matter in which we plan on utilizing our microcontroller and interfacing it with our electronic components.

μC

RF Transmitter

RF Receiver

User Controlled PC

Ultrasonic Range Finder

Temp./Humidity Sensor

Pressure Sensor

Figure 2: uC Block Diagram (Created by Group)

**4.1.1 Research Phase**

Initially, we were unsure as to what type or brand of microcontroller we would be utilizing for our project. The possibilities were endless, PIC, AVR, Freescale, or Texas Instruments. Before immediately choosing the first microcontroller that we came across, we had to contemplate a few things. What kind of programming experience do we hold, and with what language? How much memory do we need? Should we get an 8-bit, 16-bit, or 32-bit controller? How many pins will we need? Do we want a RISC or CISC CPU?

Upon further investigation, our group had reached some conclusions as to what type of microcontroller we would utilize. We had come to the conclusion that we wanted an either 8-bit or 16-bit RISC microcontroller which was easy to work with, had plenty of development tools and online support, and contained internal D/A conversion. We chose to go with a RISC architecture due to the fact that this architecture has a simpler instruction set, making it much easier for compiler to write programs in. Aside of this, we had come to discover that the CISC architecture was beginning to become obsolete compared to RISC. We also decided to either go with an 8-bit or 16-bit microcontroller. We decided upon this due to the fact that such microcontrollers are better suited for a smaller application such as our robot, where we are more concerned about cost, size, and power, as opposed to ultimate precision.

**Texas Instruments MSP430**

Originally, we had decided on the 16-bit MSP430 microcontroller produced by Texas Instruments. We happened to stumble upon this microcontroller through our Senior Design course, which offered us an instruction session with a representative of TI regarding the MSP430. After attending this, we concluded this would be sufficient for our application as the top CPU speed was 25 MHz, supply voltage of 1.8 V to 3.6 V, and 51 instructions. Despite the specifications, one of the most attractive features of the MSP430, is price. You can purchase a development board for testing purposes for $4.30 seen in the figure below, and are even granted free access to use their Code Composure Studio IDE for writing code for the MSP 430. Although we were not allowed to utilize a development board in our final project, having such a feature would greatly simplify things, as we had the ability to test out components before permanently soldering them.

**PIC Microcontrollers**

Upon further research of the MSP430, we came to the realization that this microcontroller would not be as simple to implement with our sensors as we expected. For instance, we intended on using SPI to communicate with our sensors, as we had prior experience with SPI programming, but the MSP430 utilizes USART, which would be a learning curve for the group. Also, the MSP430 contained 51 instructions, which could eventually prove to be too complex for our application. Due to a group member’s prior experience with PIC microcontrollers, we began looking into this family. Upon further research, we concluded that a PIC microcontroller was the way to go. Now the only question was, which one do we choose?

We now had concluded that an 8-bit microcontroller would be better suited for our application, as well as being easier to deal with and program, since we would not be dealing with large-numbered mathematics. We began to explore the PIC16 and PIC18 models, which were both 8-bit microcontrollers. We will reach a final decision based off of several factors, instruction sets, data space, and code space. We will also be utilizing a PDIP, as it will be easier to test with a development board or breadboard, as well as being easier to solder onto our final PCB. The table below displays a few PIC microcontroller which we took into consideration.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **PIC Model** | **CPU Speed** | **RAM Bytes** | **EEPROM Bytes** | **Program Memory** | **Supply Voltage** |
| PIC16F886 | 5 MIPS | 368 | 256 | 14 KB | 2 V – 5.5 V |
| PIC16F877 | 5 MIPS | 368 | 256 | 14 KB | 2 V– 5.5 V |
| PIC16F690 | 5 MIPS | 256 | 256 | 7 KB | 2 V – 5.5 V |
| PIC16F73 | 5 MIPS | 192 | N/A | 7 KB | 2 V – 5.5 V |
| PIC18F2553 | 12 MIPS | 2,048 | 256 | 32 KB | 2 V – 5.5 V |
| PIC18F2680 | 10 MIPS | 3,328 | 1024 | 64 KB | 2 V – 5.5 V |
| PIC18F24J11 | 12 MIPS | 3,800 | N/A | 16 KB | 2 V – 3.6 V |
| PIC18F26J50 | 12 MIPS | 3,800 | N/A | 64 KB | 2 V - 3.6 V |

Table 3: Possible PIC MCUs

All of the PIC microcontrollers listed in the table above contain many of the features which we are looking for in our microcontroller such as a low operating voltage, CPU speed, and program memory. Our ultimate decision was to choose the PIC16F886. We felt as if choosing this microcontroller would be sufficient for our application. Despite it having a smaller RAM and Program Memory size, we felt as if this smaller microcontroller would be simpler to work with and implement.

**4.1.2 PIC16F886**

Ultimately, we chose to go with the PIC16F886 produced by Microchip. We decided that this microcontroller was powerful enough and would be simple to work with and program. One of the major factors in choosing this microcontroller, was the fact that it only contained 35 single word instructions, which would greatly simply programming, as we had three different sensors which will send data to the chip. As well as this, this chip features 256 bytes of EEPROM data memory, and 11 different channels of 10-bit A/D conversion, which will be perfect for handling the conversion in our sensor data. This chip also features a synchronous serial port which can be configured as 3-wire Serial Peripheral Interface for communication between devices. This chip also features 28 pins, which we felt would be more than sufficient in order to tie all of our necessary components to it. Some of the other key features of this microcontroller include (Info taken from datasheet):

* 8 MHz to 32 kHz
* Power-Saving Sleep Mode
* In-Circuit Debug via two pins
* 25 I/O Pins
* Master Synchronous Serial Port module SPI mode, I2C mode capability
* 10-bit 11 channel Analog-to-Digital Converter
* Operating current: 11 A at 32 kHz, 220 A at 4 Mhz
* Standby current: 50 nA at 2 V

**4.1.3 Design Phase**

Now that we have selected our microcontroller, we must decide and design its configuration. Since we plan on using as small of a PCB as possible, we must make sure to consolidate room on our chassis. Because of this, we plan on constructing one main PCB to house in our chassis. This main board will house the microcontroller, sensors, and extra parts such as resistors, capacitors, regulators, etc. The figure below displays the schematic, which we will implement for our microcontroller wiring.

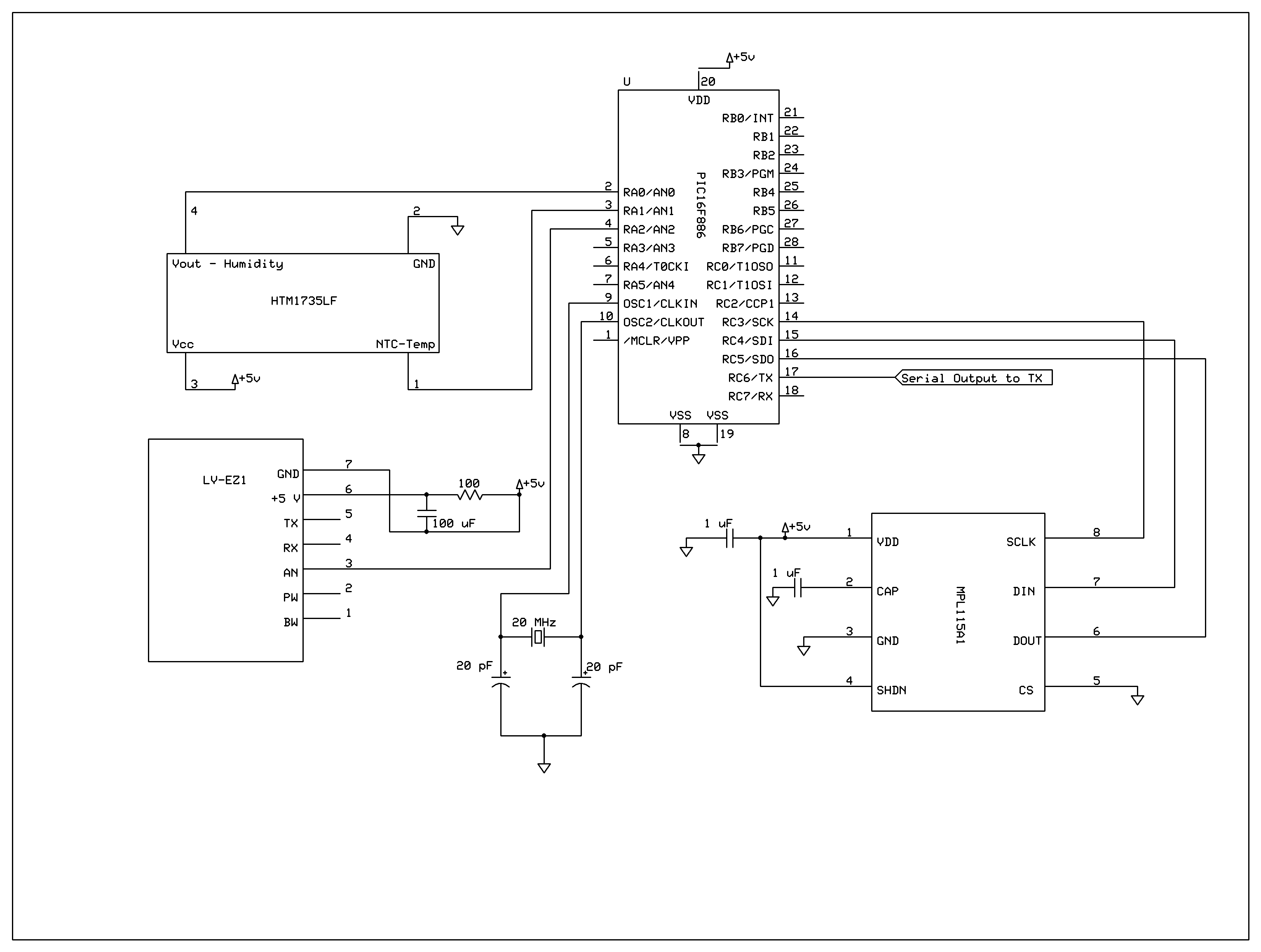


Figure 3: Microcontroller Schematic (Created by Group)

As we can tell from the figure above, regulators have not yet been implemented in our initial schematic. This is due to the fact that we have not yet fully decided as to what type of voltage regulation we will use in our final product. This will be decided in the prototyping stages over the Summer months. From the schematic, we also note that two of our sensors run into the channel A 10-bit A/D inputs of the microcontrollers. These two sensors, the temperature/humidity sensors, and the range finder, will output analog values to the microcontroller, which in turn, will be internally converted to digital values for it to process. The third sensor, the pressure sensor, will communicate through serial peripheral interface. Utilizing SPI will not be difficult since we only have one device on the bus. Therefore, we will not need to worry about the chip select lines or daisy chaining multiple devices onto the bus. A final feature which one will notice from our schematic, is the fact that we chose to utilize an external crystal oscillator, rather than relying on the microcontroller’s internal oscillator. The crystal oscillator we are using is the same frequency as the microcontroller’s internal one, 20 MHz. Going any higher could overclock the microcontroller and lead to serious potential problems with our schematic. We chose to use an external one for one major reason. Using an external oscillator could prove to give us more precise measurements with exact timing, which could be crucial in a life or death situation. Aside of this, using an external crystal is simple and cost effective to implement.

**4.1.4 Programming**

Upon deciding which microcontroller to use and how to tie the sensors to it, our group had to begin to think of the manner in which we would program out microcontroller to read information from the sensors and get it to the base computer. We decided that the manner in which we send data should be both uniform and simple. This way, it would be easy to decode and convert to the values, and we would not run into any confusion. We decided that once the microcontroller has read the 10-bit value from the sensor and stored it as an integer, we will convert this value to a string and append it to an array. The format of the array will have one letter followed by a period, followed by letters representing a 10-bit integer holding the sensor information. The table below gives a clearer idea of our method of transmitting sensor data.

|  |  |  |
| --- | --- | --- |
| **Sensor** | **Data** | **Output** |
| P (Pressure) | ABCD | P.ABCD |
| T (Temperature) | EFGH | T.EFGH |
| H (Humidity) | IJKL | H.IJKL |
| R (Range) | MNOP | R.MNOP |

Table 4: Microcontroller Sensor Output

Once these values have been calculated, we will be utilizing pin C6 on the microcontroller in order to serially send our data to the transmitter, which will in turn send it to the receiver on the base computer where it will be decoded.

In order to program our microcontroller, we had two options, PIC-C or assembly. We chose to go with PIC-C rather than assembly due to C being simpler to deal with versus assembly programming. Upon deciding to use PIC-C to code our microcontroller, we had to search for a compiler and code editor. We came across MPLAB IDE, produced by Microchip. The figure below shows a screenshot of the software.

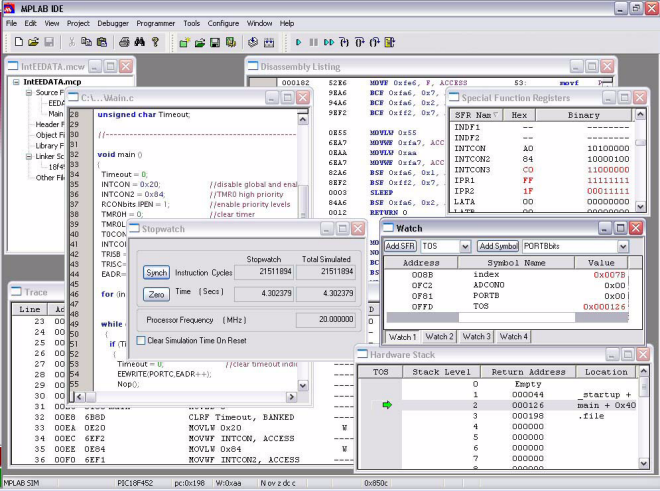


Figure 4: MPLAB IDE (Reprinted with Permission of Microchip)

We chose to utilize MPLAB IDE for two major reasons. One, it was produced by Microchip and is specifically designed for PIC microcontrollers, and two, it was free to download and use. MPLAB also includes a C compiler for writing your code in C, which is something we wanted to do. This suite also included pre-made header files which would be utilized in our code. Using an IDE with such features will allow for simpler and more direct coding. MPLAB works like any other C compiler, and is simple to compile and run your code.

Now that we had chosen our microcontroller, language, and IDE, it time to begin writing code. The flow of our code was to be simple. First, we would initialize the pins which we would use, next, we would set high the pin to read, give a delay, read it, and then the data to our function to be converted into a string. The only sensor data which will be calculated in the microcontroller code rather than computing it on the base computer, is the pressure sensor data. This was due to this sensor having seven steps in calculating the final value, rather than using one or two simple linear equations as the other sensors utilized. The flowchart below depicts the manner in which our code will work.

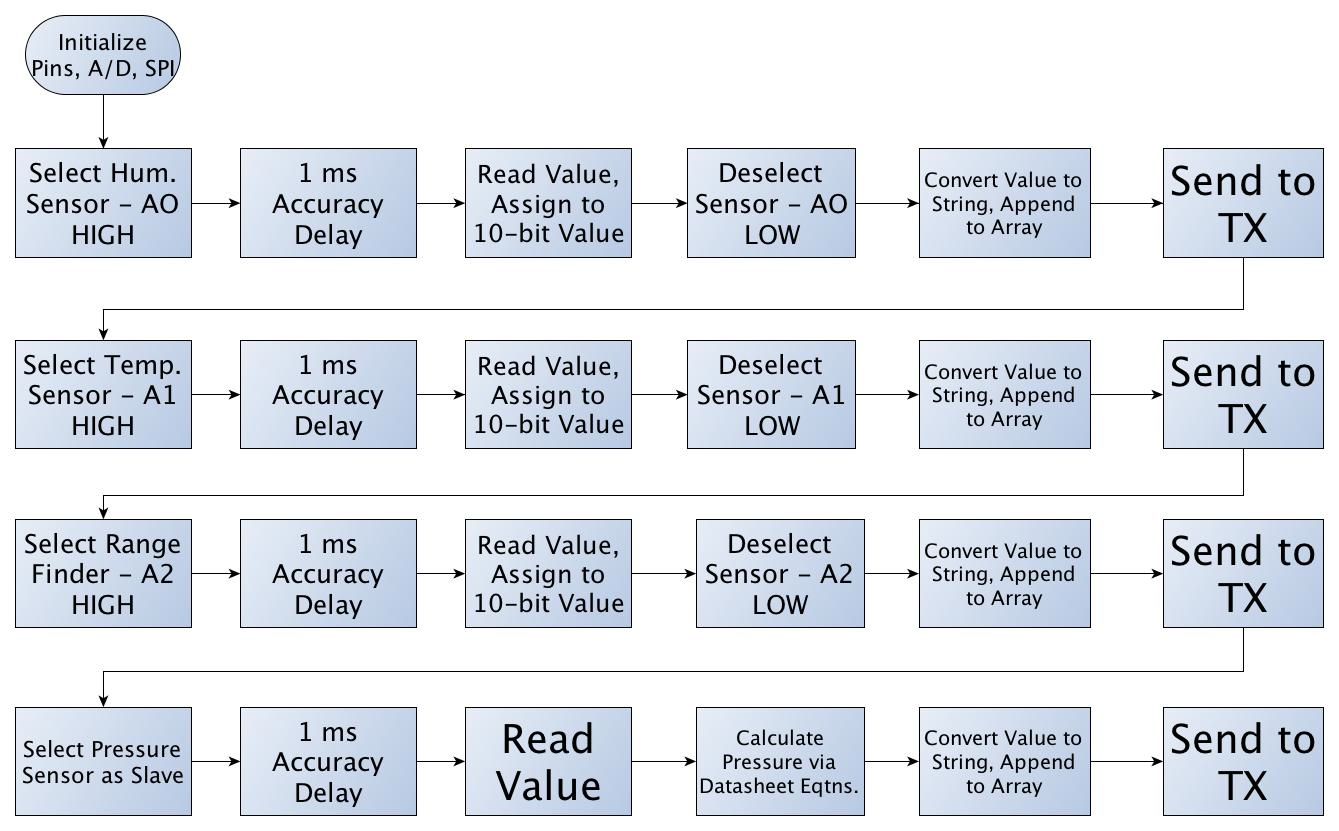


Figure 5: Microcontroller Flowchart (Created by Group)

In order to properly load our code onto our microcontroller, we will need to use a programmer. A programmer allows us to directly connect to the microcontroller via wired cable, and directly load our code onto it to run. The programmer which we will using is the PICKit 2 Debug Express produced by Microchip, a popular programmer among PIC users. The kit we will be purchasing features the programmer along with the MPLAB IDE and a debugging tutorial. The programmer connects to the computer via USB and then will connect to our development board through a connector, which will utilize five pins. The schematic below displays the manner in which we will wire our programmer to the microcontroller.

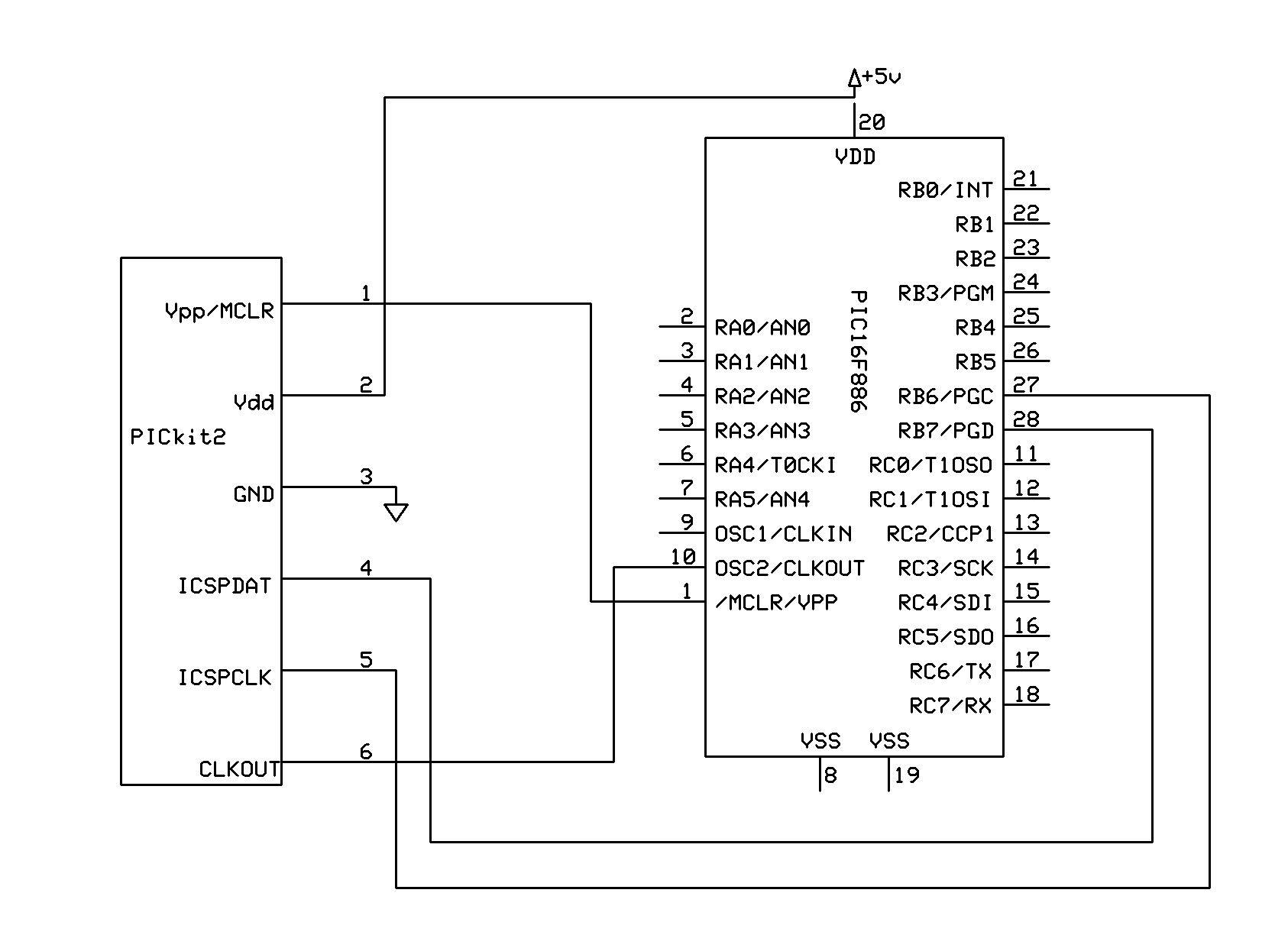


Figure 6: PICKit Connection to MCU (Created by Group)

**4.1.5 Testing/Debugging**

Testing and debugging are inevasible aspects of writing code for a microcontroller, or code in general. Because of this, we must ensure that we have a thorough testing and debugging plan. We must prepare for any possible errors or bugs we may encounter, whether it be incorrect sensor readings, erroneous wiring, or part failure. In order to prepare for such risks, we must create a both efficient and tactical plan to counteract any negative encounters. Because of this, we plan to purchase two copies of our microcontroller. We will be purchasing both a through-hole and surface mount version of our microcontroller. Purchasing a through-hole part will allow us to create circuits on a breadboard with the microcontroller and sensors. This will greatly aid us in the final design of our PCB. Once we have created and tested a circuit using the through-hole part, we will use the surface mount IC on our final PCB which will be used in the final product.

In order to efficiently test and debug our microcontroller, we decided to use a development board rather than simply using a breadboard. We decided that using a development board would be more beneficial in testing, as it contained more features than a simple breadboard. With a development board, we had the ability to easily interface our transmitter, as well as our sensors. The development board we will utilize in our testing phase is a 28 pin PIC development board sold by Sparkfun. This development board comes equipped with a +5 V voltage regulator, filtering capacitors, an external 20 MHz crystal oscillator, and a power plug-in jack. The figure below depicts the development board we will be using.

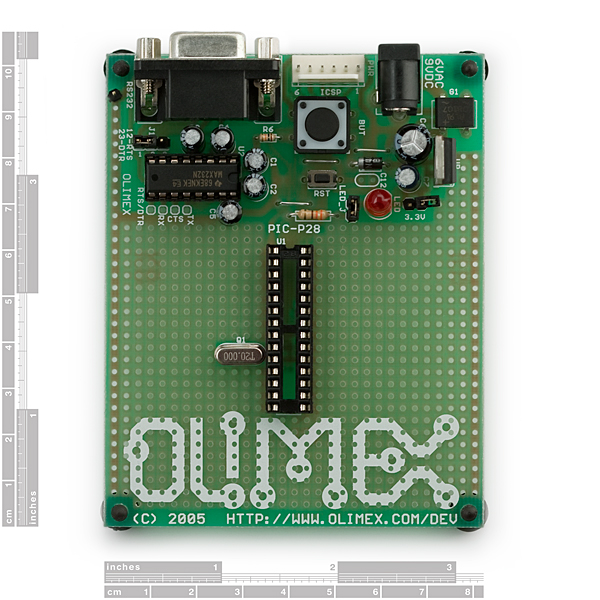


Figure 7: PIC Development Board (Reprinted with Permission of Sparkfun)

**4.2 Sensors/Instruments**

As our design stated, we will be implementing temperature, humidity, and barometric pressure sensors. Aside of these sensors, we will also be mounting an ultrasonic range finder onto our robot in order to avoid obstacle collisions. We will use one chip (HTM1735LF) to gather both temperature and humidity, one chip (MPL115A1) to gather barometric pressure readings, and one chip (LV-EZ1) as our range finder.

**4.2.1 Pressure Sensor (MPLII5A1)**

The pressure sensor we chose for our application is the Miniature SPI Digital Barometer, part number MPL115A1, produced by Freescale semiconductor. The sensor will be purchased from Sparkfun Electronics for $11.95. This sensor was selected based on its characteristics such as output voltage and current, accuracy, and conversion time.

Another sensor which we took into serious consideration was the MS5561 produced by Servoflo. It was extremely similar to the chosen sensor in the sense that it contained an internal A/D converter and operated with a 3-wire serial interface. Ultimately, we chose the MPL115A1 over the Servoflo sensor due to the fact that the Servoflo sensor was almost double the price of the MPL115A1.

Another major reason that this specific sensor was selected was due to the fact that it utilizes the SPI bus to send output temperature and humidity readings. This was useful to us because the PIC16F886 provides three pins specifically dedicated to SPI, and since we are only using one SPI device, it will integrate rather well. Also, by utilizing the SPI bus, we will free up lines dedicated for analog to digital conversion. The sensor provides a reading of 50 kPa to 115 kPa absolute pressure with a 1 kPa accuracy, which is more than enough for our application. The sensor runs on a 2.375 V to 5.5 V power supply and also includes an integrated ADC. One of the few downsides with this selected sensor is the fact that it is light sensitive. If we have direct light exposure to the port hole, we run into the possibility of having flawed measurements. Due to this, the pressure sensor will be placed either in the housing of the circuitry, or in its own separate housing.

The matter in which we will connect the sensor to our microcontroller is as follows: SLCK to RC3, DIN to RC4, and DOUT to RC5. We will not be utilizing the CHIP SELECT or the SHDN (sleep) lines. This is because we only have one slave and one master in our SPI interface, so we have no need to be switching between devices. The SHDN line is only grounded when we want the device disabled, and since we will be constantly refreshing data from the sensor, we will simply connect to the power supply connection, VDD. The figure below depicts the matter in which we will connect our sensor to our microcontroller.

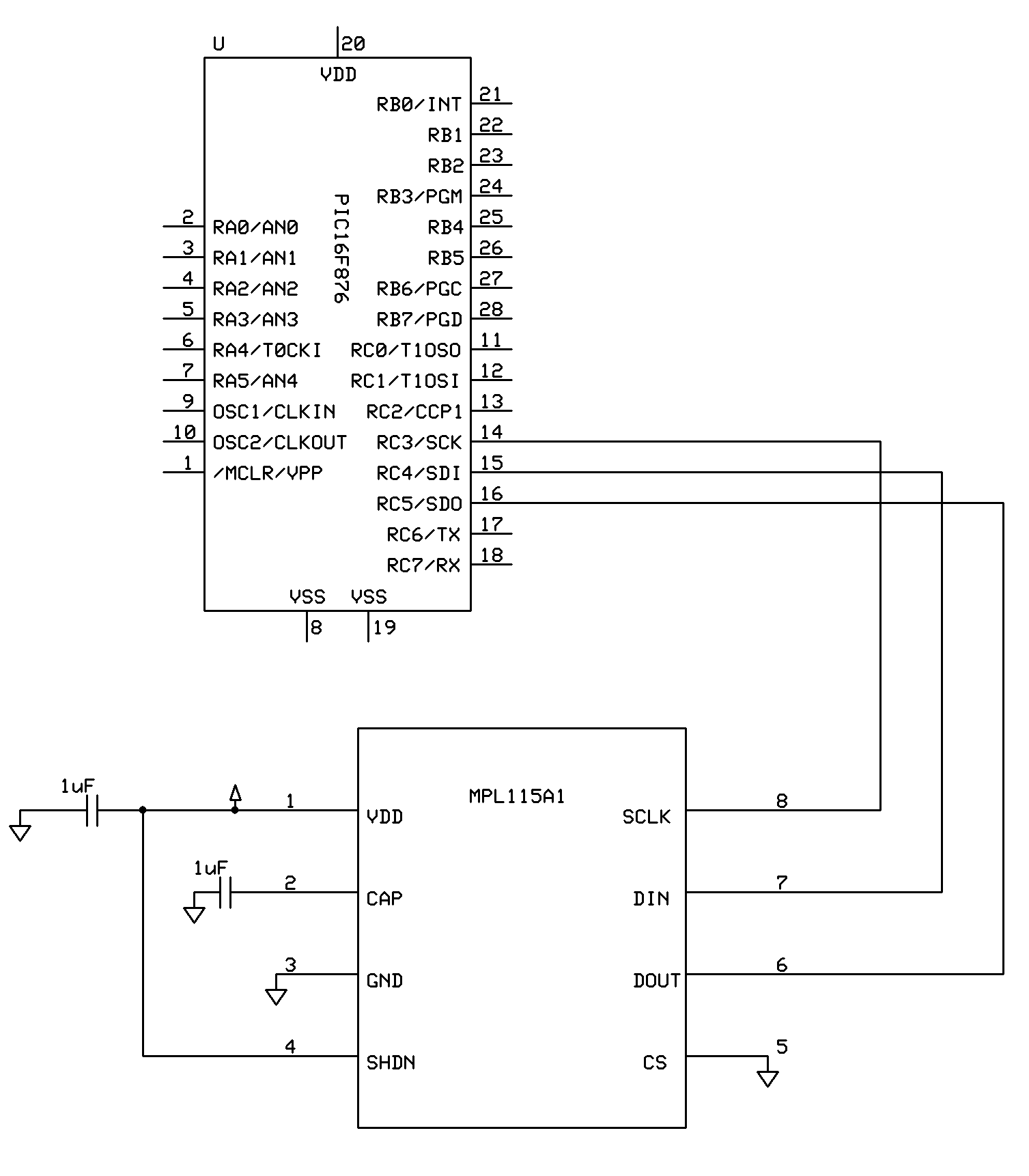


Figure 8: MPL115A1 Connection to MCU (Created by Group)

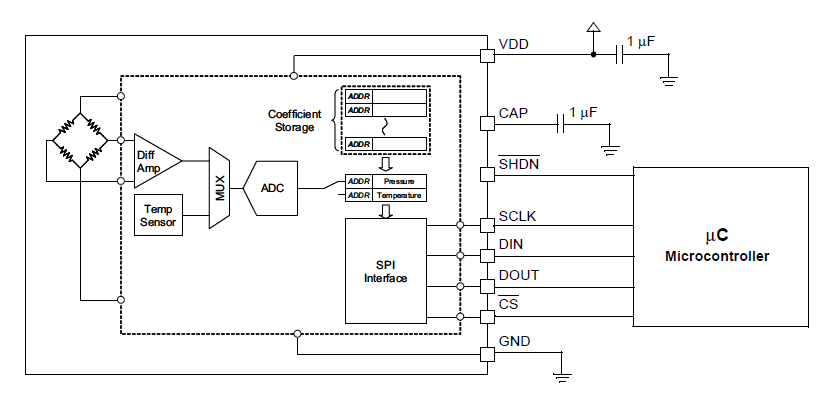


Figure 9: Pressure Sensor Internal Block Diagram (Copyright of Freescale, Inc. 2012, Used by Permission)

As we can see from the internal block diagram above, there is an internal A/D converter built into the sensor we chose, thus simplifying calculations and avoid the use of bus lines or switches to connect devices to our microcontroller. The 10-bit compensated pressure output, which the sensor calculates will be according to the following equation:

*P = a0 + (b1 + c12\*Tadc) \* Padc + b2 \* Tadc*

Where:

Padc is the 10-bit pressure output of the sensor A/D converter

Tadc is the 10-bit temperature output of the sensor A/D converter

a0 is the pressure offset coefficient

b1 is the pressure sensitivity coefficient

b2 is the temperature coefficient of offset

c12 is the temperature coefficient of sensitivity

All of these coefficients will come from the coefficient address map as provided in the datasheet. Once we have this equation solved, we can solve for this pressure in kPa according to the following equation:

*Pressure(kPa) = Pcomp \* + 50*

**4.2.2 Temperature and Humidity Sensor (HTM1735LF)**

The temperature and humidity sensor which we chose for our application was the HTM1735LF sensor produced by Measurement Specialties. We will be purchasing the sensor from Digi-Key, where it is sold for $25.95. Originally, we had chosen the RHT-03 sensor produced by Maxdetect. The RHT-03 was originally chosen due to the fact that it had extremely low power consumption, could run on a power supply as low as 3.3 V, and only used a 1-wire bus for communication between it and our MCU. Despite these characteristics, we ran into some trouble were we to use this sensor. Since DZERV should be capable of handling harsh and possible dangerous environments, it must be equipped with sensors to withstand such conditions. The RHT-03 is not suitable for such harsh conditions, and may have its internal parts or sensitivity interfered with due to chemical materials.

As previously stated, DZERV must carry sensors which are capable of being placed under potentially hazardous situations. The chosen sensor has been tested under harsh chemical conditions, therefore making it capable of withstanding acidic, salty, or extreme chemical conditions. Aside of this, our sensor is not light sensitive, which is a major benefit in the design of DERV. Due to the sensor not being light sensitive, we can have an external location for the sensor, thus assuring accurate results. The temperature operating range is -40 to 100 °C and humidity range is 0 to 100% RH, thus making it more than sufficient for our application.

Direct interface with our PIC microcontroller is made possible with the module’s humidity linear voltage and direct NTC outputs for temperatures. This sensor measures humidity with respect to voltage based on two linear equations:

*Vout = 25.68RH + 1079*

*RH = 0.03892 Vout – 41.98*

|  |  |
| --- | --- |
| **RH (%)** | **Vout (mV)** |
| 10 | 1325 |
| 20 | 1600 |
| 30 | 1860 |
| 40 | 2110 |
| 50 | 2360 |
| 60 | 2605 |
| 70 | 2860 |
| 80 | 3125 |
| 90 | 3405 |

Table 5: Humidity to Voltage

Figure 10: Voltage vs. Humidity Graph of Temp/Humidity Sensor (Created by Group)

The table above represents some sample data, which we might encounter upon deploying DZERV. Upon graphing this sample data, we can see that the output is a linear curve, as was expected. Our sensor outputs temperature according to the following equation:

Where:

RT is the NTC resistance measured in ohms at temperature T in K

RN is the NTC resistance measured in ohms at rated temperature T in K

T, TN is the temperature in K

β is the beta value, material specific constant of NTC

e is the base of natural logarithm (2.71828)

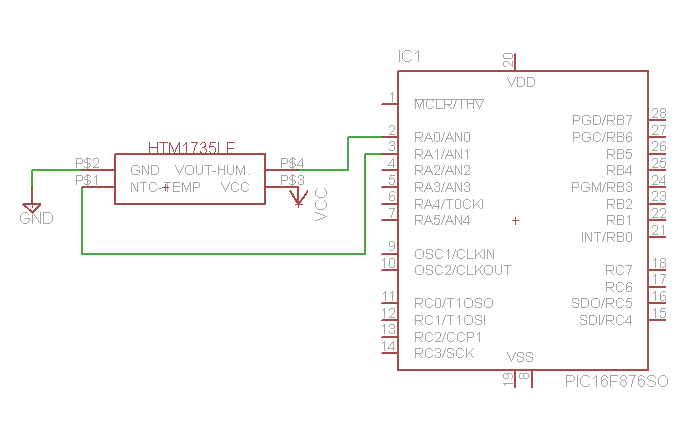


Figure 11: Temp/Humidity Sensor Connection to MCU (Created by Group)

In the figure above, we can see the connection of the sensor to our PIC microcontroller. The connection is as follows: Vcc is connected to a 5 V power supply, GND is grounded, NTC-Temp is connected to analog input A1, and Vout-Humidity is connected to analog input A0. The maximum output voltage that the Vout-Humidity pin will send is 3.6 V. The maximum input voltage that our PIC microcontroller can take in is 5 V, therefore, we will not need to utilize a voltage divider to step down the voltage.

**4.2.3 Ultrasonic Range Finder (LV-EZ1)**

Obstacle avoidance and collision is a substantial aspect to take into consideration with our robot. Not only do we want to avoid any potentially dangerous objects in the robot’s path, but we also want to protect our product and make sure it returns intact. Due to this, we have chosen to integrate an ultrasonic range finder. We decided to go with an ultrasonic range finder versus and infrared one due to the fact that we will be using the robot outside, thus causing an infrared rangefinder to be deemed useless, as it will not accurately work in the sunlight. Also, ultrasonic range finders are more accurate and project a wider beam to detect obstacles. The only downside of using an ultrasonic range finder is that they do not accurately detect absorbent materials such as sponges and they are more costly than infrared technology.

For our application, we chose the LV-EZ1 produced by Maxbotix. We will be purchasing the device from Sparkfun Electronics for $25.95. We chose this range finder due its sonar range of 6-inches to 254-inches, providing us with very short to long-range detection of obstacles. The rangefinders offered by Maxbotix come in five different beam width sizes, EZ0 through EZ4. The EZ0 model has both the widest and longest beam of detection, while the EZ4 has the smallest. As we can see from the figure below, the field of view for our range finder is in an elliptical form. The figure below also depicts how the different rod sizes and beam widths will affect the overall ability to see objects in the robot’s peripheral vision and how far ahead it will detect objects.

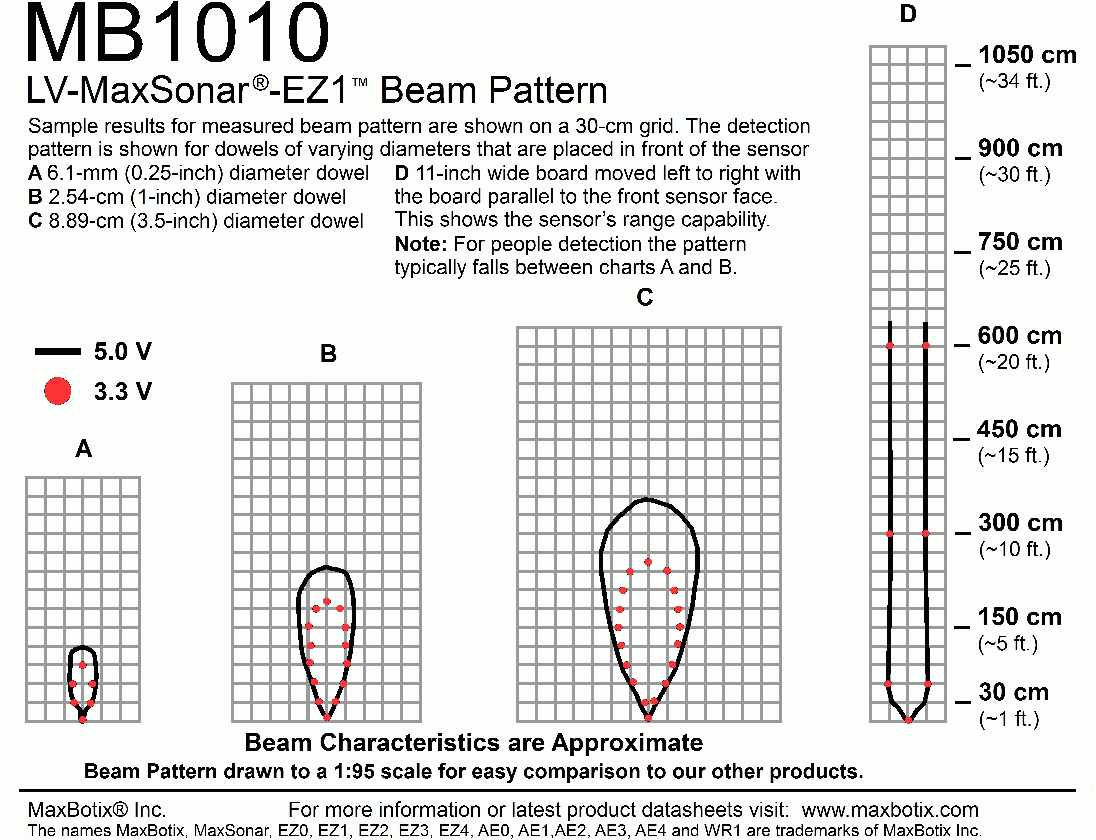


Figure 12: Range Finder Beam Pattern (Reprinted with permission of MaxBotix)

Also we have the option of interfacing our output in three separate formats, pulse width, analog voltage, or serial digital output. For our application, we chose to go with the analog voltage output, as it will be the simplest to deal with and interface with our microcontroller. In fact, by using the analog voltage output method, we will only utilize three out of the seven pins on the device, with only one line running into the microcontroller. Also, readings from this device can occur up to every 50 ms, which is more than sufficient for our application. The only negative aspect of this device is that the project beam is a bit narrow, thus not being able to identify objects in the robot’s peripheral vision.

By supplying the LV-EZ1 range finder with a supply voltage of +5 V and utilizing the analog voltage output, our range finder will yield approximately 9.8 mV/in. Since our device provides sonar range from 6-inches to 254-inches, our analog voltage output will transmit a value of 0.058 V to 2.48 V. From the datasheet, we know that the device produces approximately 9.8 mV per inches with the supply voltage as 5 V with a 9-bit factor. However, since our PIC microcontroller uses 10-bits for A/D conversion, we must make use a separate equation to properly measure distance from objects. The following equation will provide us with the distance:

*(Total inches of capable sensitivity/total a-d value)\*2\*(sensed value)*

From the datasheet, we know that our max capability of sensing is 254 inches, and having a 10-bit A/D conversion, 210 is equal to 1024. Therefore our final equation for total inches sensed isas follows:

*\* 2 \*(sensed value) = actual inches sensed*

Wiring the LV-EZ1 range finder to our microcontroller will be a rather simple process. We will only have one line running from the device to the microcontroller, the AN pin will run to analog input pin RA2 on the MCU. Also, in order to provide superior results, our line must be free of any electrical noise. In order to accomplish this, we will place a resistor in series with a capacitor at the input power pin. The figure below is a schematic of the manner in which we will wire our range finder.

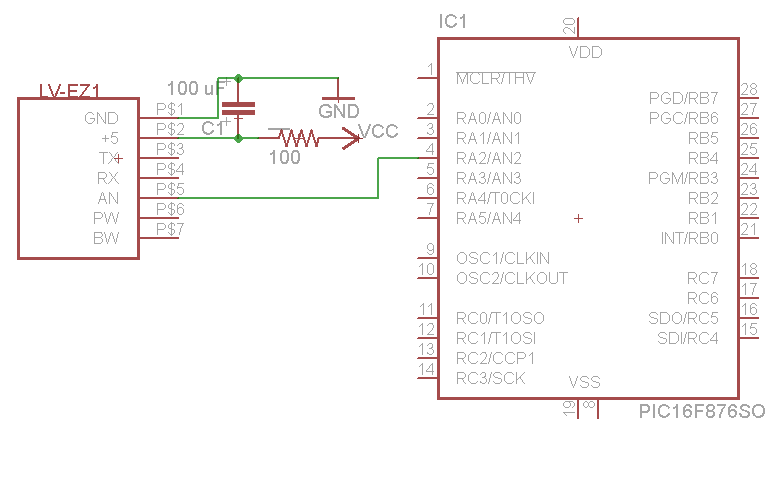


Figure 13: Range Finder Connection to MCU (Created by Group)

**4.2.4LED BAR**

Since DZERV will be operating under potentially hazardous conditions, it might encounter poor lighting, whether it be sent on a mission in during nightfall, or

into a rubbled building in which visibility is poor, it must be equipped to handle such situations. Due to this, we will be desiging an LED bar which is capable of detecting darkness in order to conserve energy. However, our LED bar circuit will be a stand-alone device and will not run off the robots power source, rather it will run on its own. Originally, we planned on purchasing an LED light bar from Sparkfun Electronics. However, we realized that the LED bar ran off a 12 V power supply, which was much more power than we needed for the rest of our robot. We all concurred that it would also be much more intriguing to create our own LED bar.

The photosensitive element which we will implement in order to detect darkness is a phototransistor, type LTR-4206E produced by Lite-On Electronics. The phototransistor which we chose is an infrared-sensitive type with a dark lens. The only downside of this is that it will not see fluorescent or discharge lamps, only sunlight and incandescent lights. We will also be implementing two transistors in our circuit in order to turn on our LEDs at proper times. We will be using 10mm White Super Bright LEDs, part number NTE30071, which we will purchase from Radioshack. We wil be powering our circuit with its own separate power supply, 3 AA alkaline batteries, which will provide 4.5 V and will be housed inside of a purchased battery holder. The LEDs which we picked have a forward Voltage of 3.3 V and diode rated current of 25 mA, therefore, we calculated our load resistors to be about 50 Ω. The figure below is the schematic of our circuit which we will eventually place onto a PCB.

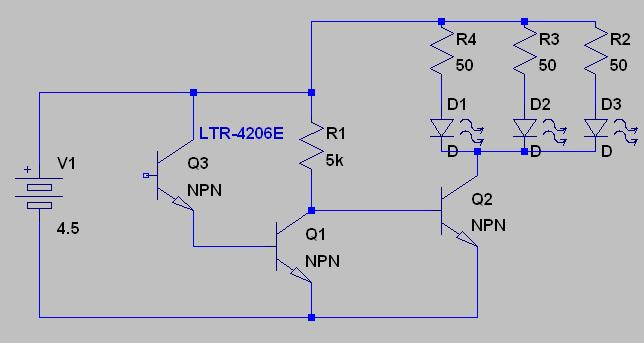


Figure 14: Dark-Detecting LED Circuit (Created by Group)

**4.3 Camera (AirSight Wireless IP Camera)**

In order to produce a superior search and rescue robot, we must have some means of visual navigation. A robot which cannot be visually guided by the base operator would be of no use for our application. Due to this, we had to find some sort of means of transmitting video and audio back to the operator’s base computer, we chose to utilize a camera. We decided our camera should be affordable, reliable, sturdy, and transmit both video and audio.

Originally, we thought that going with a color sensor module with a digital output would be our safest bet, as it was a cheaper alternative to buying a camera. We had found the C3038 color sensor module sold on Electronics123. However, we soon realized that utilizing a color sensor module would prove to be both more costly, and more complicated, as we would have to interface it with the microcontroller and originate a server client program in order to transmit video to the base computer. Aside of this, the color sensor module was not capable of transmitting audio.

After eliminating the idea of utilizing a color sensor module, we went with our only other option, purchasing a ready-to-go camera. Originally, we had decided on purchasing the Planetcam kit produced by x10. X10 produces a kit for $129.99 which includes a wirless PC transceiver, motion sensor, USB video capture adapter, control software, and the camera itself. This camera seemed as if it would be ideal for us, it was weather resistant, sturdy, small, and simple to mount. Also, it already came packaged with a USB adapter in order to transmit the video straight to the machine and software to control the camera. Such items would greatly simplify the idea of controlling our robot with a camera. The only problem we ran into with this camera, was that it did not have an ability to pan and tilt, which would only give us a straight line of sight, thus dampening the efficiency of our robot. However, the greatest problem we ran into is that the camera required a power supply of 12 V. This would prove to be a problem for is, and we were soon searching for a new camera.

Upon erasing the idea of the x10 PlanetCam, we stumble upon IP cameras. By utilizing wireless internet and router, we could directly communicate with and control our camera using wireless internet with no extra adapters. According to IEEE standards, 802.11b and 802.11g WAPs and routers support a range of up to 150 feet indoors and 300 feet outdoors, which was more than sufficient for our project. We chose to go with the Airsight wireless IP camera (Model XX34A) produced by x10 as well, retailing for $99.99. The camera would come packaged with the camera itself and the software in order to stream the audio and video to, as well as controlling the pan and tilt features.

The AirSight camera comes equipped with automatic nightvision, down to 05. lux, and up to 32 feet. We saw this feature as being extremely beneficial to our application. The pan and tilt of the camera is 270 degrees horizontally, and 120 degrees vertically. The camera also contains infrared motion detection which can alert us whenever motion is detected which the base operator may not be able to see due to poor visibility. The camera also comes equipped with either WEP, WPA, or WPA2 secure encryption.



Figure 15: Airsight IP Camera (Reprinted with Permission of X10)

As we can see from the figure above, the camera is rather small, which is perfect for our smaller chassis which we are utilizing for our robot. Aside of this, audio is transmitted to the computer as we required, and the camera also contains a speaker. This could be useful in situations where we are searching for survivor or live victims. One of the major benefits of this camera, is the fact that it only requires a 5 V power supply instead of a 12 V power supply, which will be much easier to incorporate into our robot. We will not be utilizing the I/O interface pins, as those are used in order to send alerts from the IR motion detector to e-mail. The only problem we will face in utilizing this camera, is that it is not designed to withstand harsh outdoor conditions. Due to this, we will be creating a small clear plastic housing for it. The housing will not cover the lens, as to not interfere with video quality, but will only protect the camera from outdoor conditions such as rain or dust. The only thing which we must take into consideration, is the thickness of the plastic. If we make it too thick, we might run into the problem of interference with our wireless signal.



Figure 16: Camera Software (Reprinted with Permission of X10)

Another major benefit of this camera is that it already comes with its own client-server GUI software, which we simply install on the base computer. A screenshot of the software can be seen in the figure above. As we can see from this screenshot, the software brings full control of the camera, giving us the ability to control everything from the pan and tilt features to the audio and video resolution. By utilizing this camera and software, it will make it simple for someone with either no engineering or real technological knowledge to operate the camera and the robot itself. The only problem with the software for this camera is the fact that it cannot be incorporated with the GUI module which we will create for the sensor data, it must be a stand-alone module. Due to this, the base operator will be forced to have two windows open simultaneously running while operating the robot, one to process incoming sensor readings, and the other window open with this software running in order to properly navigate the robot.

**4.3.1 MAX232 Chip**

Since we will be reading incoming sensor data through the serial port of our base computer, we must have some method of converting the digital input into signals suitable for our RS-232 serial port. In order to accomplish this, we will have to build a separate PCB which will connect to the base computer incorporating a MAX232 IC produced my Maxim Electronics. This IC will be purchased from Mouser Electronics for $6.11. This chip will provide RS-232 voltage level from ±3 V to ±15 V from a +5 V power supply to the chip itself. The table below describes the logic of the MAX232 chip. Whenever it receives a logic 0 or 1, it changes it to the appropriate voltage.

|  |  |
| --- | --- |
| RS232 Line | RS232 Voltage (V) |
| Rx/Tx Logic 0 | +3 to +15 |
| Rx/Tx Logic 1 | -3 to -15 |
| RTS/CTS/DTR/DSR Logic 0 | -3 to -15 |
| RTS/CTS/DTR/DSR Logic 1 | +3 to +15 |

Table 6: RS232 Outputs

In order to properly receive the TTL signals being produced by the microcontroller, we will have to incorporate the RXM-916-ES transmitter and the MAX232 chip on the same PCB which will connect to the serial port. We will be connecting our MAX232 chip into a DB9 232 connector. sThe pins which we will be utilizing for our PCB will be 2, 3, and 5. Pin 2 is the RD (Receive Data), pin 3 is the TD (Transmit Data), and pin 5 is GND.

In order to properly pass data to the base computer, the MAX232 chip must receive to correct incoming signals. In order to achieve this, it will have to be wired to our RXM-916-ES RF receiver. Essentially, the PIC microcontroller will send sensor data to the TXM-916-ES, which will then feed that data to the RXM-916-ES receiver. Upon receiving that digital data, it will be passed to the MAX232 chip to transform into the correct analog voltage to be fed into the base computer. The figure below displays a bock diagram of how these modules will interact.

PIC uC

RF Receiver

MAX232

RF Transmitter

Base Computer

Figure 17: Sensor Data Flow

Both the MAX232 and RXM-916-ES receiver require only a +5 V power supply, which will simplify our PCB design greatly. The RF receiver is rather easy to deal with, as it only requires three connections and two groundings. Pin 1 (ANT) will run to our 50-ohm RF input antenna, Pins 2 and 4 (GND) will be grounded, Pin 5 (Vcc) will be connected to our +5 V power supply, and pin 12 (DATA) is our digital data output, which will run into our MAX 232. The pins labeled NC on the RXM-916-ES in the figure below are not an electrical connection, but rather soldered for physical support only. The MAX232 chip is also rather simple to deal with. It only requires wiring the power supply pins, five external capacitors, and our connection to the RF receiver. Connecting it to the serial port will be dealt with later. The schematic below displays the manner in which we will wire our RXM-916-ES RF receiver to our MAX232 chip.

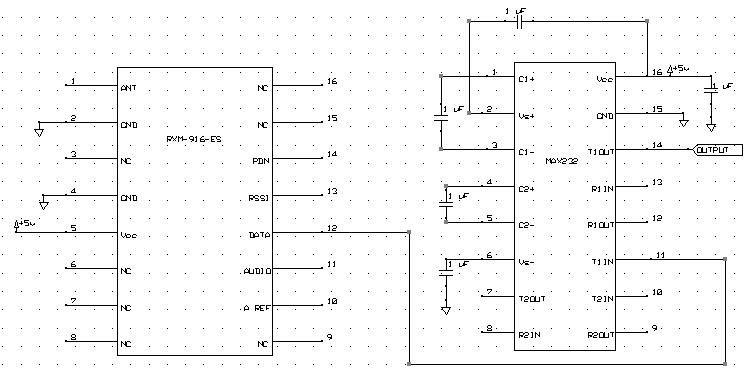


Figure 18: RF RX to MAX232 Wiring (Created by Group)

**4.4 Software/Interface**

The figure below is a flowchart of the way in which our software will operate. The user will first go through the proper setup of connecting the receiver box to the base computer and powering on the robot. After this, they will open the camera software in order to obtain visual data. Once this initial setup has been complete, they will open the DZERV.exe icon, which will start the program. Once the serial port has been opened to receive incoming sensor data, the user will have the option of navigating the robot via the keyboard arrows or an onscreen GUI of arrows. While all of this is occurring, the PIC microcontroller will be sending out sensor data to the transmitter, which will in turn send it to the receiver, all of this will occur in the background.

Open Serial Port

Navigate Robot via Keyboard Arrows

Process Incoming Sensor Readings

Print Sensor Readings to User

Quit Motion/Program

No

Yes

Figure 19: Software Flowchart (Created by Group)

Upon receiving the sensor data on the base computer, it will be written to a text file with a time and date stamp. The data in this text file will have a refresh rate of about 100 ms in order to ensure up-to-date readings. A sample of what the text file will appear as can be seen in the figure below.

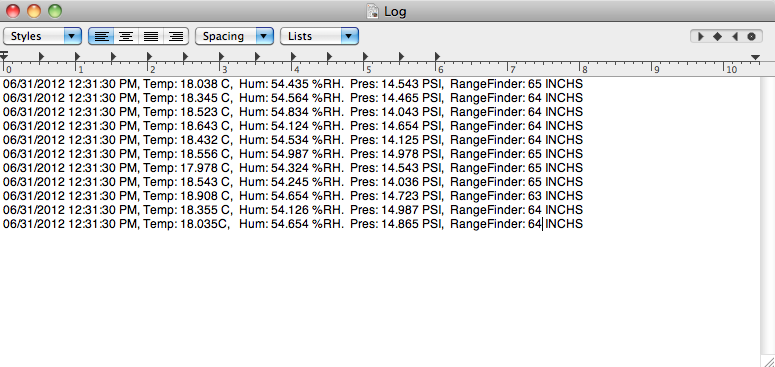


Figure 20: Log Data (Created by Group)

This portion of the application will be coded using Visual Basic, as it is a straightforward language to use. Due to it being straightforward and simple to grasp, the group members with electrical engineering backgrounds can work on the coding as well. The Visual Basic code will be written to take in the “X.ABCD” formatted strings from the serial port connected to the receiver. Once our code receives this string, it will decipher and arrange according to temperature, humidity, and distance. Once we have this information, the code will execute the proper equations to arrive at our sensor values, which will be printed to the text file described above.

**4.4.1 GUI**

The graphical user interface, which we create, must be designed in order to allow a non-technical user operating the base computer to utilize the robot. This means it must be simple and easy to use. The user will begin by opening all necessary software as stated in the previous section. After this initial setup, the program will monitor the serial port for new data, and write this data to a text file at a refresh rate of 100 ms. Such a process will occur in a continuous loop, or until the user decides to terminate the program. Our GUI software will be coded in an object oriented fashion implementing the Java programming language.

**openFile Class**

This class is the main component of our GUI program. It will serve as a means for the operator on the base computer to open and select the log file which contains all of the sensor data. It contains the following members.

* **private JButton openButton, saveButton:** These buttons allow the user to open a directory to search for the log file, as well as allowing a user to permanently save the log file to their machine for future use or record keeping.
* **private JTextArea logArea:** This will serve as a logging area where the program will let the user know what is currently happening with their file. It can have logs such as: “Opening: “ +file.getName() + “.” + newline”, “Saving: “+file.getName() + “.” + newline”, “Couldn’t fine file: “+path”, and “Save command cancelled by operator”.
* **private JFileChooser chooser:** This is a Java API which makes it easy to bring up open and save dialogs for the operator on the base computer.

The openFile class will contain the following methods:

* **public openFile:** Here we will create our log area, as well as creating and laying out our buttons using FlowLayout.
* **public void actionPerformed(ActionEvent e):** This method will handle the actions of when the user clicks the open and save buttons. If the user clicks “Open a File”, it will handle opening the file, as well as handling the saving of a file if the user clicks “Save a File”.
* **private static ImageIcone createIcon(String filePath):** This method will return a Java ImageIcon or null if the file path provided by the user is invalid.
* **private static void createDisplay():** This method will create our GUI and display it to the user.

Below, we can see a screenshot of our GUI in action opening a log file which contains our sensor data. In the future, we will make the GUI more appealing to the user, as well as adding more functionality to it.

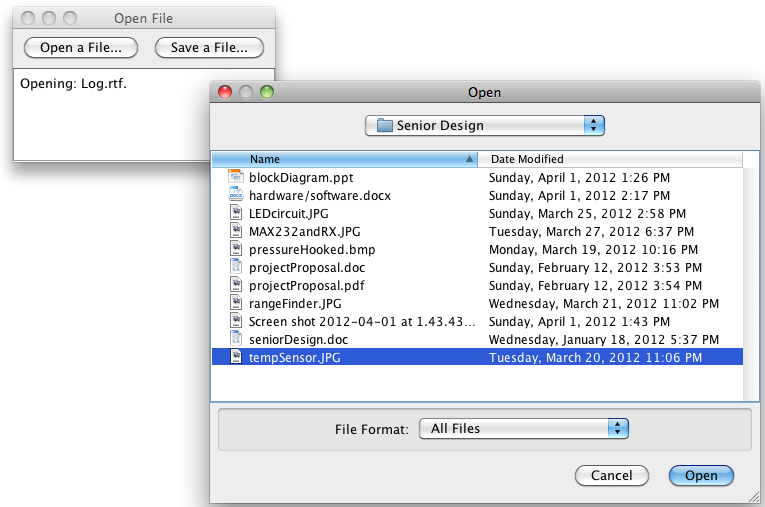


Figure 21: Display File GUI (Created by Group)

**4.4.2 Robot Control**

As previously mentioned, the base operator has the ability to control the robot by using the arrow keys from the keyboard, it will the the up the user’s discretion as to which they implement for control. In order to use the arrow keys, a user may press any combination of the arrows in order to produce a result as seen in Table 7 below.

|  |  |
| --- | --- |
| **Command** | **Result** |
| ↑ | Move robot straight forward/Increase Speed |
| ↓ | Move the robot straight backward/Decrease Speed |
| → | Move the robot right |
| ← | Move the robot left |
| B or b | Brake/Reduce Speed |
| E or e | Emergency Brake |

Table 7: Robot Control

In order to control our robot from the base computer, we decided it would be easier and more efficient to utilize a wireless module versus utilizing the computer’s serial port. While utilizing a wireless module might be less cost effective, it will require less components to tie to the base computer. In order to accomplish this wireless manuevering our our robot, we will be utilizing the XBee wireless modules, a motor controller, and the Processing open source programming language. We will buy two XBee modules, one to implement as a transmitter from the base computer, and one to implement as a receiver mounter on the robot. The receiver mounted on the robot will be connected to the motor controller, which will in turn be connected to the motor our of our robot. The code we write will be sent to the transmitter via USB, which will wirelessly be sent to our receiver in order to control our car. The figure below depicts the manner in which our wireless control of the robot will work.

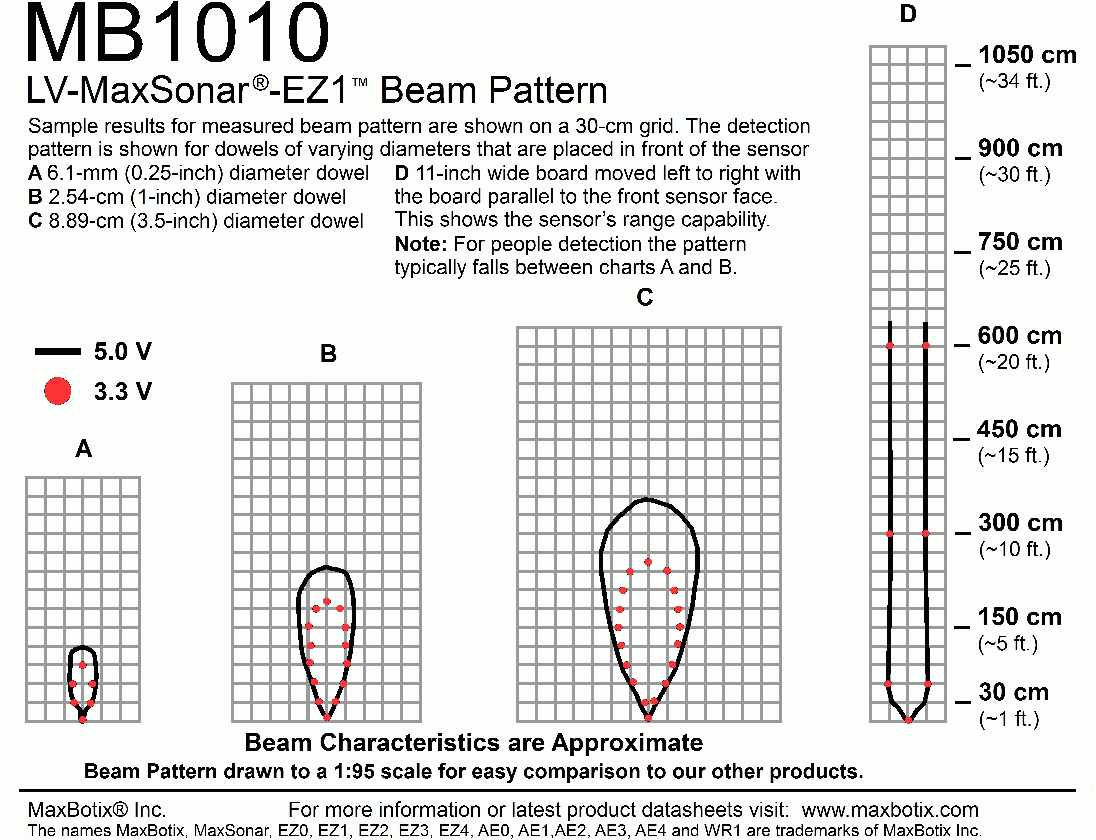


Figure 22: Wireless Control (Created by Group, XBee and Controller Reprinted with Permission of Sparkfun)

For our coding environment and language, we will be utilizing a free open source language, Processing. The structure and syntax of Processing is very similar to the language utilized by Arduino, thus making it rather simple and straightforward to use. Since we are communicating with a communication port to send code to our XBee transmitter, we will use the serial library provided by Processing. In utilizing this library, our code will consist of if statements reading user inputs, and write() functions to send data to the communication port. Once the code has been completed, it will be converted into a .exe file, which will allow the user to simply click the icon to begin maneuvering the robot. The flowchart below depicts the manner in which our code will act.

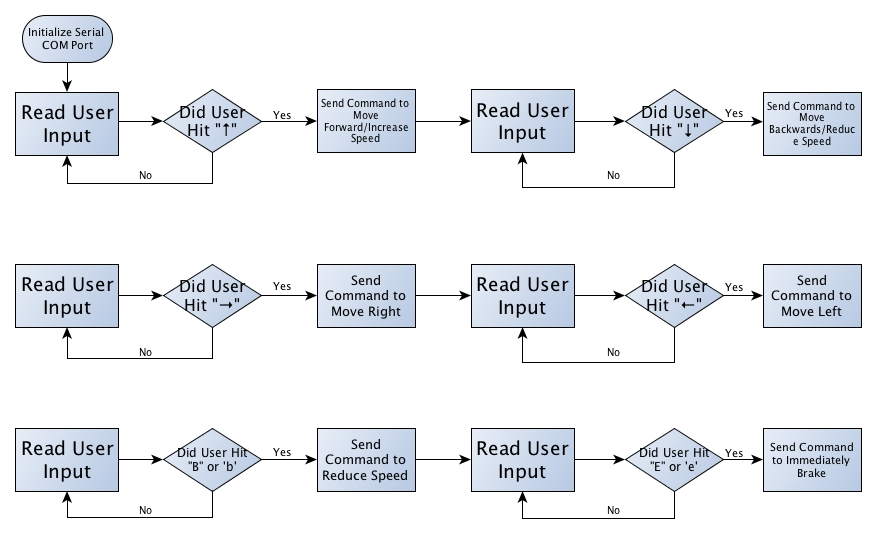


Figure 23: Robot Control Flow Chart (Created by Group)

**4.5 Voltage Regulators**

Because the power supply that we are going to be using is actually much higher than the rated voltages for a majority of the parts we will be using for our robot, it is certainly obvious that we will need to do something to maintain the voltage in our circuits at a manageable level for the components. This can be achieved in a variety of ways. For instance, we could use voltage divider networks of resistors to do this. However, due to the amount of components for our robot for which this type of voltage regulation would need to be done, attempting to regulate the voltage in this way would not be very feasible, as this would result in the need for many more resistors than would likely fit on our PCB. It therefore became necessary for us to investigate the use of voltage regulators as our means of limiting the voltage flowing into our devices.

Due to the nearly limitless variety of voltage regulators on the market, we therefore needed to conduct further research to see which would be the best for our needs. Would we go with Texas Instruments parts? Would we rather go with components from Linear Technology? Would we go with switching regulators, or would it be best to stick to the linear voltage regulators with which we’ve become familiar from use in the lab? These are all questions that would need answering before we could even begin to design a finalized circuit schematic for our PCB.

In order to simplify the process a little, we have constructed a table of the parts which are of the most concern to us in maintaining proper regulation, as these are all very much mission-critical components. Failure to meet these requirements could result in a catastrophic loss for our robot, and it is therefore imperative that we ensure that we stay within the voltage ranges set forth by these components’ datasheets.

|  |  |
| --- | --- |
| Part | Operating Voltage Range |
| PIC Microcontroller | 3.3 V – 5.5 V |
| Pressure Sensor | 2.375 V – 5.5 V |
| Temperature/Humidity Sensor | 4.75 V – 5 V |
| Ultrasonic Range Finder | 2.5 V – 5 V |
| Data Acquisition Transmitter | 3 V – 12 V |
| Data Acquisition Receiver | 3 V – 12 V |
| XBee Transmitter | 2.1 V – 3.6 V |
| XBee Receiver | 2.1 V – 3.6 V |
| MAX232 Chip | 3 V – 5.5 V |

Table 8: Parts to Keep in Regulation

From this table, it is fairly clear that we should be fine to keep all our supply voltage regulated to about 5v. With this in mind, we can now begin looking into possibilities for voltage regulators. Fortunately, we already have experience with some regulators from Texas Instruments, and we were able to begin our investigation there. However, before we began looking into specific parts, it was necessary for us to investigate whether we would be using switching regulators, as opposed to linear voltage regulators.

**4.5.1 Linear Regulators vs. Switching Regulators**

Before we could decide what type of regulators we’d be using for our circuits, we would first need to familiarize ourselves with the differences between the various types of voltage regulators on the market. We could go with the standard linear regulators with which we are familiar, but we could also utilize something like a switching regulator. Linear voltage regulators work in a fairly simple manner. The difference between the input voltage and the rated output regulated voltage is dissipated across the semiconductor material of the regulator as heat. While this makes for very simple circuitry to work with, and while these types of regulators are much cheaper, it is very important to note that the way in which these regulators actually regulate the voltage is quite inefficient. The equation for power dissipated across one of these types of regulators is the following: P­reg  = (Vin - Vout) \* Iload, where Preg is the power dissipated through the regulator, (Vin - Vout) is the difference between the input voltage and regulated output voltage, and Iload is the current through whatever device may be connected to the output of the regulator. As an example, if a 12-volt power supply were to be connected to a 3.3-volt regulator, which was in turn connected to the power input pin of a microcontroller that sinks 5 mA of current, this would mean that 43.5 mW of power would be dissipated across the voltage regulator. While this is not an issue, as this is a rather low power dissipation and well within the operating limits of most linear regulators, this can become quite a problem when dealing with components that sink much more current, as the power dissipation would grow rather rapidly. Depending on the thermal resistivity of the semiconductor material within the regulator, as well as a few other factors, a very small increase in power could lead to a very rapid rise in temperature across the regulator. This in turn could cause the part to cease its normal functions, which could then lead to major problems in the circuitry to which it is attached.

With this in mind, however, we did note that a vast majority of our components are low-powered enough that these types of issues would not arise, and in the event that they did, we would be able to resolve them fairly easily by adding a simple heat sink to ensure that our parts do not get destroyed. Armed with this knowledge of linear voltage regulators, we were ready to begin looking at components which could be useful in our design. Before doing so, however, we realized that it would also be necessary to investigate all possibilities for voltage regulation for our device, as this is the only way we would truly be able to know which option would be the best to suit our needs.

In addition to linear voltage regulators, we also investigated into the properties of switching regulators, or switch-mode regulators or DC/DC converters, as they are commonly called. These devices work much in the same manner as their linear counterparts, but in many cases their efficiency is nearly double that of linear voltage regulators. Through our research, we had noticed some regulators with an efficiency as high as 87 percent. This type of efficiency would work wonders for our robot, as these types of regulators are typically used in applications such as robotics. Additionally, these types of regulators are also great in applications where a great deal of heat would cause issues, as these regulators are designed to generate very little thermal energy while in operation. They are commonly used as a means for voltage regulation in applications where battery power is the main source of power for the circuit, and are popular for such applications as a result of their amazing capacity for efficiency.

Switching voltage regulators rely very heavily on the concept of pulse width modulation. There are many various types of switching regulators, such as step-down (Buck) regulators, step-up (boost) regulators, and even inverting regulators, which simply invert the voltage placed on the input pin. While the last two types of regulators have countless uses in modern electronic technology, for the purposes of our own project, we would need to stick with a Buck regulator, which takes an input voltage and regulates it down to a lower voltage. Figure 24 below is a common setup for one of these types of regulators.

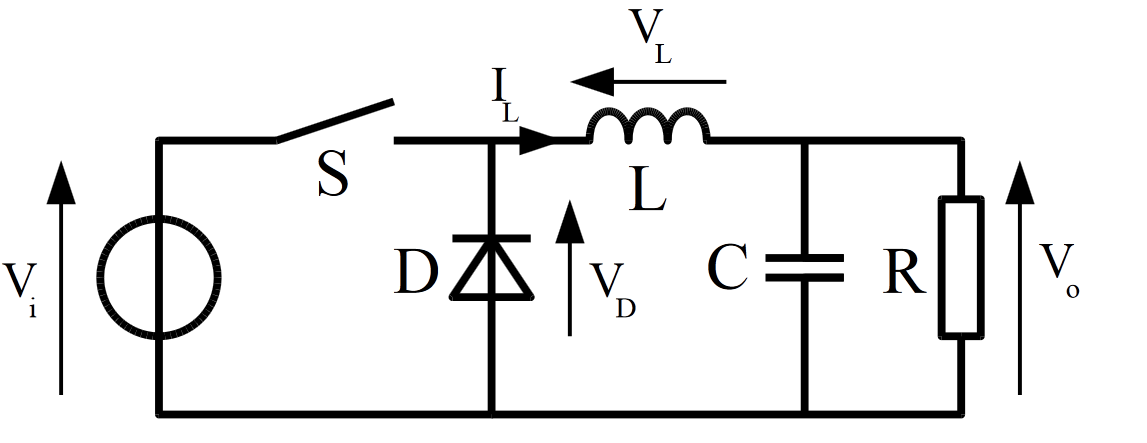


Figure 24: Buck Regulator (Reprinted with Permission of Wikipedia)

The way this works, is that there is a bipolar junction transistor connected to a PWM controller, which causes the transistor to act as a switch. The latter half of the circuit includes an RLC network in which the inductor is then connected directly to the input. When the switch closes, the inductor then dumps current into the capacitor and the load, causing the capacitor to charge up. Once the PWM controller causes the switch to open once again, the current through the inductor cannot change instantaneously. Therefore, the voltage across the inductor must change to compensate for this to hold the current at a constant value. This continuous cycle is what causes the output voltage to remain steady at a fixed value, depending on the values for inductance, input voltage, and various other electrical characteristics for the circuit.

Because all this is typically included in a single integrated circuit, implementation into our design would be relatively simple to accomplish, and is therefore another option that we have considered. Armed with the knowledge and insight into the operation of linear and switching voltage regulators we have gained, we were then prepared to finally begin our investigation of components for use in our robot. We began with an investigation into various linear voltage regulators.

**4.5.2 Linear Regulator Options**

Based on our experiences taking the Electronics II class at the University of Central Florida, we knew that looking into National Semiconductor’s LM7800 series of voltage regulators would be a great place to start, as our group members have had experience in creating regulating circuits with these components. With a dropout voltage of 2 volts, this regulator should actually be more than able to suit our needs, as we’re going to be using a 7.2 volt supply. Unfortunately, we discovered that we would really only be able to purchase this component in a through-hole package. The major disadvantage to this is that it would take up more board real-estate, and would likely be a cause for concern later on in the prototyping phase. While we would still be able to use this in the event we were unable to settle on another device, we strongly believed that this would not be the greatest idea. We did decide to keep the 7805 regulator on our list of contingencies, as this would certainly suffice for our needs. This particular regulator can be purchased for $1.95 from SparkFun. Figure 25 below demonstrates how we would have this particular regulator connected to some of our components.



Figure 25: LM7805 Regulator connected to some of our components

In addition to seeking a surface mount component for our project, we also decided that we would look for something more efficient as well, as efficiency is one of the major constraints with which we are working on this project. The less power dissipated by all our parts, the more efficient our final design will be. With this in mind, we expanded our search to include the LM2936 from National Semiconductor (now a part of Texas Instruments). This regulator offered a number of advantages for us.

One of the largest advantages of this regulator is that it is offered in a surface mount Small Outline Integrated Circuit (SOIC) package, featuring eight pins, of which two are used, four are ground pins, and the remaining two pins are no connect pins. Based on researching this component’s datasheet, it is clear that this part would be more than able to do the job for us, and it would likely be a great option for us to go with. It has a very low quiescent current value, which is perfect because this enables the regulator to behave more like an ideal regulator, in which the quiescent current is zero. This regulator would maintain regulation at 5 V for our parts, and even includes reverse battery protection, likely in the form of a diode connected between the input and output pins in order to prevent current from flowing back through and causing damage to our power supply. This in and of itself, is a huge advantage as this would simplify out design a little bit in that we would no longer have to account for this reverse current in our design. Additionally, it ensures that the life of our power supply would be preserved as much as possible.

The major drawback to this voltage regulator is the price. At $2.44 per regulator from Digi-Key, these components prove to be a little expensive, as we would need to purchase a few of these to ensure that all our parts are receiving voltages within their range of operation. Also, as with the 7800-series regulators, one major issue we would have with this regulator is its lack of efficiency. Because of the fact that this is also a linear voltage regulator, it operates by the same principles as the 7800 series regulators, meaning that even with just a very small amount of voltage regulation, the power dissipation can be incredibly high, causing a wide variety of problems for us. These types of issues could range from desoldering to the part simply not working and not providing the regulation we need it to provide, which in turn could severely damage the other components on our PCB. With this in mind, we continued our search for a voltage regulation solution. For a rough idea of how this would look if implemented in our circuit, Figure 26 shows how this regulator would be connected to some of our components.

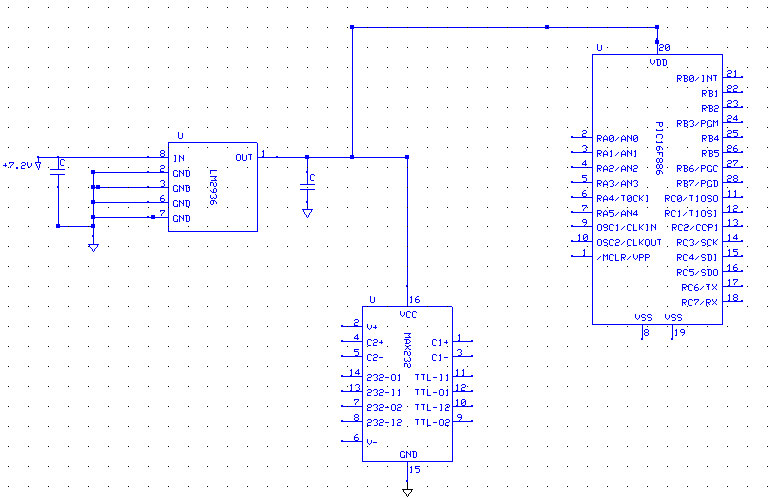


Figure 26: Example of the LM2936 connected to some of our components

As seen in Figure 26 above, the regulator is connected to the input voltage source, and its output is connected to the other devices requiring voltage regulation. The capacitors to ground are in place to ensure that the signal is kept clean, and the other five pins are connected directly to ground. Were this an adjustable regulator, other resistors and capacitors would be implemented in the circuit as well.

Another part that we had considered is the LT1121-5 from Linear Technology. Being a very widely-known manufacturer of IC components such as these, we decided to search Linear Technology’s extensive list of parts to see whether they had something that could suit our needs. We ended up investigating the 1121-5, which is another linear 5 volt regulator that we could use in our design. One of the reasons that we looked into this particular product is that Marcial had some experience in seeing the adjustable version of this regulator in action from a project that he had been working on with his employer. Because of this experience, we found that this regulator would likely be the best one for our needs between all the linear regulators we investigated.

This device has a very low dropout voltage between .13 V and .42 V, depending on the input current. Again, this ensures that we’ll be able to maintain the necessary regulation for our parts. Additionally, and perhaps the largest advantage of this, is that this component is available as both a through-hole part and in a surface-mount part, which means that we’ll be able to use this regulator for both prototyping, as well as our final design. This regulator also has a very low quiescent current value, which means that this also will act nearly ideal, and will ensure that our parts are well- protected. The only source of concern for us for this part, really, is cost. This part is available for $3.92 from Digi-Key, which is slightly expensive, but is in fact, less expensive than the LM2936 regulator. Despite this, our familiarity with this part would likely be well worth the additional cost.

Despite the fact that we had linear voltage regulators so thoroughly, we also realized that we would need to look into what our options would be for regulation using a switch-mode regulator. From what our research had told us, this would be a better option for us, given the power-saving properties of switching regulators.

**4.5.3 Switching Regulators**

The first switching regulator we investigated was the LTC3632 regulator from Linear Technology. This regulator is a Buck regulator designed to regulate down to 5 V. The part’s datasheet actually gives an example of how this particular part should be connected, which actually has proven to be quite useful. This component comes in a convenient 8-pin Mini Small Outline Package (MSOP), which fits our needs for size. Additionally, we would be able to utilize the example circuitry from the datasheet in order to ensure this component regulates to 5 volts, so this eliminates the need for guesswork and additional calculations on our part, which simplifies the process for us even more. The largest advantage that this would give us is the fact that Buck regulators are known for their power-saving capabilities. This would allow us to extend the life of our robot’s battery and enable it to run for a longer amount of time. The regulator would be connected to our circuit in a similar manner to what is seen in figure 27 below.

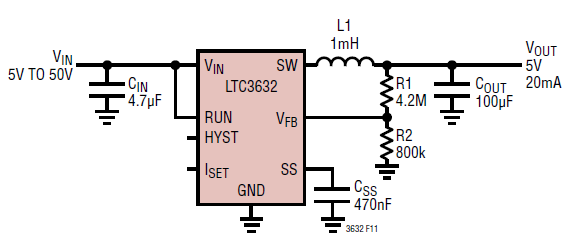


Figure 27: Sample implementation of the LTC3632

(Pending permission from Linear Technology)

Here, the LC filter that was discussed previously can be seen, and these are the components that do a majority of the work in this regulator. This external circuitry is what will maintain the voltage at 5 volts.

After further investigation, we soon realized that switch mode regulation, while certainly more efficient, would be slightly more difficult to implement. We have therefore decided that we will stick with linear voltage regulators to accomplish this task for our robot. Ultimately, we believe that sticking with the LT1121 regulator will give us the least amount of difficulty, allow us to accomplish what we are setting out to accomplish, and allow us to do so as efficiently and inexpensively as possible.

**5.0 Power and Communication**

**5.1 Chassis**

Since there are plenty of pre-made remote control cars as well as those we could put together ourselves available to us, we decided to breakdown the advantages and disadvantages of each to come up with the best solution for our robots main body. Not only will the design have to be effective, but the cost has to be right as well.The first option we looked into was to build a simple chassis ourselves. One model we found was the Boe Bot robot chassis sold by Parallax.

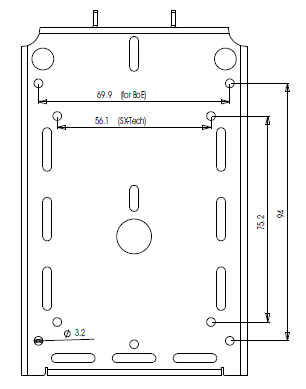


Figure 28: Boe Bot Chassis Dimensions Bottom View (Reprinted with permission from Parallax)

As can be seen from figure 28, there is a lot of room and plenty of premade holes for mounting wheels, motors and other components and at only $24.99, this seems like it would act as a nice base for our chassis. The total length of this chassis is 127 cm and it is 87 cm wide. This provides us with plenty of real estate to not only assemble our electronic parts on, but also the motors with which our robot would move. The chassis is also squarely punched in order to be compatible with Parallax and Futaba style servomotors. This may work to our advantage so no additional drilling will be needed to mount our motors. We did find other base chassis; however, most of them were way out of our price range. They ranged anywhere from $60 to well over $100. Since this conflicts with our goal of low cost, we have to be careful when it comes time to choose our parts, and this includes the chassis.

Despite the fact that the design of the above chassis supports servo motors, we decided to look into other types as well. We found that we could either use continuous DC motors, stepper motors or the aforementioned servo motors. We did some research and found advantages and disadvantages to both which are outlined in Table 9.

|  |  |  |
| --- | --- | --- |
| Motor | Pros | Cons |
| Cont.  DC | * Wide selection available, both new and used. * Very easy to control via computer with relays or electronic switches. | * Requires gear reduction to provide torques needed for most robotic applications. * Poor standards in size and mounting arrangements. |
| Stepper | * Does not require gear reduction to power at low speeds. * Low cost when purchased on the surplus market. * Dynamic braking effect achieved by leaving coils of stepper motor energized (motor will not turn, but will lock in place). | * Poor performance under varying loads. Not great for robot locomotion over uneven surfaces. * Consumes high current. * Needs special driving circuit to provide stepping rotation. |
| RC  Servo | * Least expensive non-surplus source for gear motors. * Can be used for precise angular control, or for continuous rotation (the latter requires modification). * Available in several standard sizes, with standard mounting holes. | * Requires special driving circuit. * Though more powerful servos are available, practical weight limit for powering a robot is about 10 pounds. |

Table 9: Various Motor Pros and Cons

**5.2 Motors**

**5.2.1 Continuous DC**

The first type of motor we looked at was the continuous DC motor. We liked the fact that they were easy to find and relatively easy to implement with a computer. Since none of us had any kind of experience with small motors, we looked into some components that would make our DC motor work for us. The first main component needed is an H-bridge. What this allows us to do is to get both a forward and reverse response from the motor without having to rewire our circuit. This circuit can be set up with either mechanical switches or electronic switches, but since our background revolves around electronic components, it would be in our best interest to use electronic based switches.

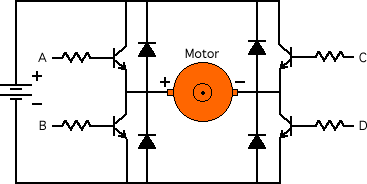


Figure 29: Typical H-bridge using transistors as switches (Reprinted with permission from Micromouse)

In the above circuit, the transistors replace the mechanical switches. In this circuit, the winding in the motor acts as an inductor and when the current through the motor is shut off, the current can’t immediately go to zero, and a voltage is produced that opposes this current, therefore a diode is placed in parallel with each transistor to provide a safe pathway for current to go in order to not blow the transistors. Closing switches A and D will result in forward motion and closing C and B will result in reverse. Closing switches A and C will short the battery which will allow us brake in a sense.

Since our design would require 2 DC motors in the rear of the robot to power each wheel, we felt it would be advantageous to find an H-bridge that could handle such a task. We found such a part made by Texas Instruments called the DRV8835 Dual Low Voltage H-bridge IC. This is a 12 pin device and has the pin layout described in Table 10.

|  |  |
| --- | --- |
| Pin | Function |
| 1-VM | Motor Supply Voltage |
| 2-AOUT1 | Bridge A Output 1 |
| 3-AOUT2 | Bridge A Output 2 |
| 4-BOUT1 | Bridge B Output 1 |
| 5-BOUT2 | Bridge B Output 2 |
| 6-GND | Ground |
| 7-BIN2/ BENBL | Bridge B Input 2/Enable Input |
| 8-BIN1/ BPHASE | Bridge B Input 1/Phase Input |
| 9-AIN2/ AENBL | Bridge A Input2/Enable Input |
| 10-AIN1/ APHASE | Bridge A Input 1/Phase Input |
| 11-MODE | Input Select Mode |
| 12-VCC | Device Supply Voltage |

Table 10: DRV8835 Dual Low Voltage H-bridge IC Specification table

This device supports a 2-11 V motor operating voltage and a device supply voltage of 2-7 V. This range is perfect for us as it allows for a lot of flexibility with the power supply that we choose. We found out there are two modes for this device, In/In mode and Phase/Enable mode. Upon reviewing the data sheet, In/In mode sets the MODE pin to 0 and allows for forward, reverse, brake and coasting modes dependent on the four possibilities for both In pins. The Phase/Enable mode sets the MODE pin high and only allows for forward, reverse and braking features. Either mode of this H bridge will work for us if we choose to go with a DC motor setup. The AOUT and BOUT pins are simply connected to the motor.

Another common part associated with using DC motors is a gearbox. However, since our robot will be light weight and not have to carry a heavy load, high torque will not be necessary, however, there are high rpm motors out there that we could use that would suit our needs that come with a good enough gear ratio for low speeds. If we choose to use DC motors in our design, the gears will be taken care of in the initial design of the motor, so we do not need to worry about any extra gear reduction.

We did find a DC motor that would be perfect for our design. The MT-WSM500 is a high torque mini DC gear motor rated from 3-12 V. This model costs $12.95 per unit and is shown in Figure 30. The small size of the motor allows it to be easily concealable and will take up very little room on our chassis should we choose to use it.

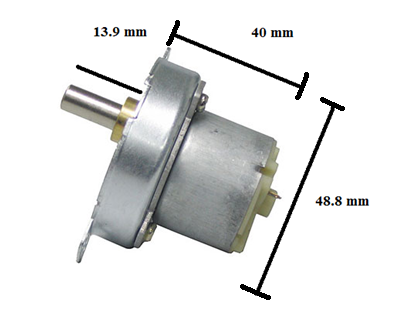


Figure 30: MT-WSM500 DC Motor dimensions (Reprinted with permission from BatterySpace)

If we were to use the above H-bridge, the connections would be easy enough. We would need two for our design to power the rear wheels and all that would be left to do would be to decide which mode (In/In or Phase/Enable) we wanted to use. Table 11 below contains a list of characteristics that made this motor desirable to us.

|  |  |
| --- | --- |
| Characteristic | Specification |
| Gear Reduce Ratio | 1:108 |
| Voltage & Current | 3-12 V DC < 25 mA with no load |
| Torque | 1000 g-cm @ 6V |
| RPM | 5 RPM @ 3V, 25 RPM @12 V |
| Reversibility | Yes |
| Weight Per Motor | 4 oz |

Table 11: Continuous DC motor specifications

Since the terrain our robot will be going through may not be very even, this high torque motor may be what we need after all. The fact that each motor is only 4 oz is also really great, since we would like to keep the robot light and mobile. Since the H bridge supports up to an 11 V input, we would not run these motors to their full potential, but we are able to get pretty close to it if we chose to use the 11 V that the part is rated at.

**5.2.2 Stepper Motor**

The next option we had for a motor was the stepper motor. While the disadvantages seemed to outweigh the advantages of using this type of motor described in Figure 30, it was still worth seeing what this kind of set up would do for our overall design.

The addition of a stepper motor would provide some extra precision that simply cannot be reached with a DC motor. Since the stepper motors have a certain amount of steps that can be implemented per 360 degrees, it would be very easy to control how many steps are taken by the microcontroller. If the degree per step is known as well as the gear ratio and radius of the tires used, it would be very easy to make the robot as precise as needed by making the proper calculations. The main problem lies with the terrain our robot will see. It will be really hard to predict the terrain of the robot until it is actually in the disaster area. Since the stepper motors use a permanent magnet rotating shaft, the magnets will constantly be shifting around, providing us with the strong possibility of having our motors not work to our standards due to offsetting the magnets. Also, as mentioned before, these precise movements are for more finesse applications and since we already decided we are going to use a video camera to gauge our distance, this extra precision will not be needed. Since the camera will not be affected by the conditions the robot sees, we decided that stepper motors will be unnecessary for this design.

**5.2.3 RC Servo Motors**

This section will talk about servo motors and why they would be useful in our design. These motors are common in radio applications, which is very beneficial to us since we would like to use this type of communication. In the application of RC cars, these motors are often used to control the throttle, brakes and steering, however with modification, the motors can be used to control the remote control cars wheels.

Most servo motors have a range of 0 to 180 degrees, but this would not benefit us in making our servo motor a drive motor. Instead, some modifications must be made to allow for a full 360 degrees of rotation. We could employ the brute force method and open up the motor, remote the electronics and bring out the power lines and either remove or modify the potentiometer to get it to rotate the full 360 degrees. The only thing we would have to do then is to mount pieces for the wheels to go on and make sure that we have a motor driver circuit to make the motor functional.

The easier way to modify the motor is to break the feedback loop that is internal in the motors circuitry. This is most easily done by removing the potentiometer (which is basically a variable resistor) from the circuit and replacing it with a voltage divider. This will be advantageous to us because it will allow us to keep the motor in the neutral position. We want this to occur because keeping it in the neutral position will allow for the motor to turn at the same rate when it goes either forwards or backwards. This technique will work because the potentiometer will now not be able to limit the servo motors rotations. Thankfully none of these methods need to be implemented because servo motors now come with a full 360 degree range.

The RC servo motors are controlled by Pulse Code Modulation (PCM). The parameters for this pulse are that there must be a pulse with a minimum width, a maximum width and another pulse for repetition rate. The neutral pulse, which will allow for the potential of rotation in both clockwise and counterclockwise motions to operate the same is around 1500 micro seconds. Providing pulses above and below this rate will allow for our forward and reverse motion. In general, the minimum pulse rate is around 1000 micro seconds and the maximum around 2000 micro seconds. However, these values are dependent on the type and brand of the servo motor used.

Servo motors generally consist of a motor, feedback device and a control board. The feedback device is usually a potentiometer. The motor is designed to turn the output shaft and the potentiometer at the same time, however, our design would replace the variable resistor with a voltage divider, hence not allowing for any change in the resistance. The servo will take in a pulse and the control board will decipher this data. Since the motor would now be able to move 360 degrees without any modification, gears will not be limited. The control board would then provide the correct action that the motor will take and this will allow our robot to move either forwards or backwards, depending on the length of our pulse.

Now that we understood how these motors work, it was time to start finding different parts and comparing their specifications. It delighted us to know that the tedious task of modifying the servo may not have to take place after all, since there are models out there that have 360 degree rotation available at the time of purchase. The first motor we found was sold by SparkFun made by SpringRC.

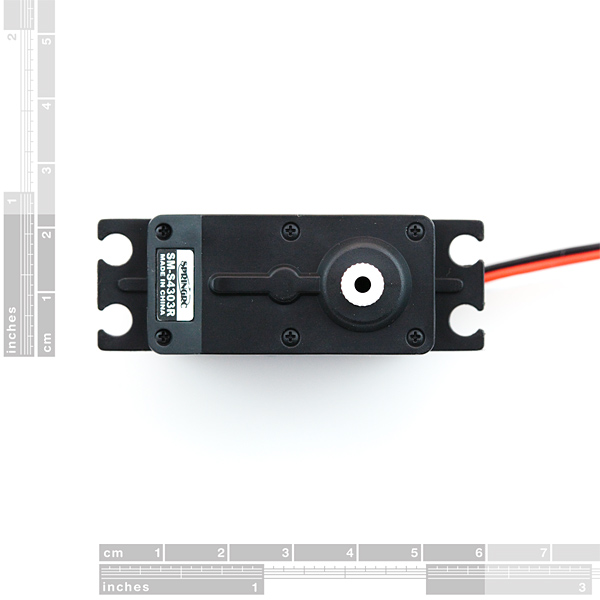


Figure 31: SpringRC 43R Servo motor dimensions (Reprinted with permission from SparkFun)

As mentioned before, this model already has a full 360 degrees of motion making our lives easier since we would not have to make modifications to it. Its operating voltage is anywhere from 4.3-6 V DC. We were not looking for a heavy duty motor with tons of torque, so the maximum voltage allowed would give us a torque of 4.8 kg-cm. Even at 4.3 V, we would have plenty of torque since it is rated at 3.3 kg-cm at this voltage level. The hook up of this model is relatively easy since like most servo motors, there are 3 wires. In this model, the red wire is set to Vcc, the black wire is set to ground and the white wire would be used to read the incoming signal.

This models intermediate impulse width is 1.5 milliseconds, which matched up with what we expected when we researched servo motors. Any pulse width above this value will result in clockwise rotations of the motor and vice versa. We also appreciated that the motors are relatively light at only 44 g. The price is also very competitive to the DC motor we were looking at since this model costs $13.95 per unit.

The next model we looked at was the SM-S4303R servo made by SpringRC and sold by Phidgets Inc. This particular model has an operating voltage of 5 V and also produces 3.3 kg-cm of torque at 4.8 V just like the motor found on SparkFun. This motor weighs only 41 g, making it just a few grams smaller than the first model we looked at. One disadvantage of this motor compared to the first one is that we can’t gain more rotations out of the motor. It has a set operating voltage whereas the model on SparkFun allows for a little better range. Despite this minor disadvantage, this motor still has some appeal since it is only $10.70 per unit.

**5.2.4 Price Comparison**

Now that we have searched for a chassis and researched different parts that would propel it, it was time to see if not only the components were functional and what we were looking for, but also if the prices were right. It quickly became apparent that building and mounting everything ourselves would be quite expensive, and with all of the other components still left to buy, this seemed highly unpractical. If we were to use the Boe Bot chassis and either the DC or servo motors, this would already bring our cost above $50. This price does not account for the four wheels and tires needed or equipment that would assist us in steering the front wheels. Also, the fact that we would most likely have to make additional modifications to the chassis for mounting purposes did not make this method of doing things very desirable.

This left us with the idea of buying a pre-assembled remote control car and simply modifying it. This process may not be too easy since there is a possibility that the remote control car will not come with schematic diagrams that show how the internal circuitry works, hence making it difficult for us to tap into the controls for the motors. Despite this minor setback, further research has shown that this method has been implemented many times before with many cases being greatly successful. Also, since we have a smaller group of only 3 members, we decided it was not in our best interest to build our own anyway since none of us have any prior experience building such things and we wanted to make this project stand out for the technology implemented in it and not for its homemade look. Therefore, we will buy a preassembled model and focus on the electronics.

We did some searching online and compared a lot of different models. We wanted something that was rather compact, but also something that had some decent speed. Our plan is to gut the model car of most of its components, leaving the chassis top and motors intact. We then want to modify the car in a way such that it looks like we did little to nothing at all to it. This would mean that what we found would have to allow for enough room for us to be able to hook up our electronics and then be able to hide them underneath the body of the chassis. Doing this would also test us in our ability to make our design compact. We have seen many cool search and rescue projects in the past, but some looked rather chunky and did not use the space available to them in a way that would give the appearance of a sleek design. It is quite common to see these smaller designs done on a more professional level since the designers in charge have so much more experience than we do. However, with proper attention to detail and careful planning, we feel like we could make this happen. Having a limited space would challenge us to use every inch of what’s available to us to the fullest.

We found the perfect candidate for our remote control car at Target. Not only did it look the part of being a rugged search and rescue vehicle, but it also appeared to have enough space underneath the body for us to work with. Figure 32 below is a stock image of the vehicle.

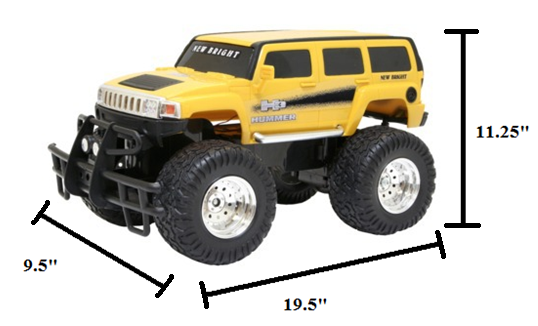


Figure 32: Remote Control Car dimensions (Reprinting pending permission from Target)

This model relies on RF communication to make it move, but since we plan on gutting the circuitry, this would allow us to explore other options for communication. This model also comes with a 6 V rechargeable battery and charger. The fact that this remote control car is already carefully designed will allow us to include the power source on the robot as is without having to make any further modifications, thus allowing us to keep going with the idea of keeping the stock look of the vehicle. The current set up of the car allows for a range of 100 feet with the remote control that it comes with, but further research will be done to see if this distance can be improved upon with components that we will be purchasing separately.

One drawback of this car is that its maximum running time is only 20 minutes if we were to use the 6 V battery it came with. However, we are not planning on running this robot full speed at all times since the environments it will be used in will not allow it to. Therefore, this is really not a big issue. We did like the fact that it does have the potential for decent speed ratings. If our robot is in an environment such that there is a clear path and we need it to make a quick escape, it can reach speeds of up to 7 mph.

Another great feature of this car is its weight and decent ground clearance. At only 4.56 pounds, this will allow for easy transportation from one place to another. Also, the lighter our final robot is, the less power it will need to consume in order to move, so this initial weight is a great starting point. We would like the weight of our final design to weigh as little as possible, keeping the amount of torque needed to propel it to a minimum. A lot of the other remote control cars were simply too low to the ground and would not thrive in an environment with debris in its way. One concern we did have is the fact that the tires are plastic. They simply may not be able to get the traction necessary to overcome certain debris, but the advantages that this remote control car has to offer to us simply cannot be ignored.

The last specification that sealed the deal on this car was the price. It will only cost us $29.99. For only $30, we are able to get a chassis, motors to propel the car, a 6 V rechargeable battery and a charger. For the price, this simply cannot be beat when compared with getting all of the initial components ourselves. At this price, it would also be relatively cheap for us to buy a whole new unit if necessary in case we damage some of the components. This is obviously not ideal, but it will save us lots of money if we do need to do this. The fact that we only need to go to Target to buy this model instead of having to order one online and not know when we will receive it will also work in our favor.

**5.3 Communications**

One major question we had upon building our robot was how we were going to get out robot to move and operate with the best results. We decided to look into both infrared and radio frequency technologies and see what the advantages and disadvantages were for both. Not only did the technology have to be suitable, but it also had to fit our budget. Therefore, it was necessary to take a good look at both kinds of technologies and decide what was best for us.

**5.3.1 Infrared Communication**

In this day and age, the popularity of infrared communication is picking up steam. We decided that it would be worth our time to see if infrared would provide us with the best results for our robots mobility. We knew that this technology is very prevalent in the remote controls of many household items such as DVD players and TVs, but we were curious how infrared would work in the claustrophobic environment which our robot would be in.

It quickly became evident that infrared provides many advantages. The fact that it requires low power to operate greatly intrigued us since we are interested in building a low power robot. The circuitry also happens to cost very little which also got us piqued our interest. Another interesting thing we found out was that since the whole system requires excellent line of sight, better security is enabled. Any infiltration of our data would require that other parties be able to access great line of sight as well, but since we are not concerned who retrieves our data, since the basis of our robot is to inform as many people of certain conditions as possible anyways, we felt that this property did not matter to us.

Infrared provides a much shorter wavelength which would allow us to gain readings from our robot at a much higher rate than anything at a lower frequency, such as radio waves. This would only serve to make the operation of our robot more accurate, since the response of our robot would occur much quicker.

**5.3.2 RF (Radio Wave) Communication**

After doing research, we decided that this was the ideal type of communication we were going to implement for our project. The biggest problem with the infrared technology was simply line-of-sight. Since our robot is going to be designed to go into a disaster area, the chances that objects will obstruct our sight with the robot will be extremely high, thus making infrared technology useless to us. Another advantage of using RF communication was that the range is simply greater. With an RF setup, it is possible to have a range of 300 feet from the remote (which in our case will be a laptop) to the base unit. Figure 33 below is a simplified overview of how we want our communications to behave.



Figure 33: Module showing how we want our robot to receive instructions

Basically, the laptop will act as the base for the transmission of the direction we want our robot to go in, which will be picked up by a receiver on our chassis. We have decided to forego the use of the remote control that came with our car and use the directional pad on a laptop as the primary means of making our robot go in the desired direction. This will allow us to use the laptop screen as a means of driving the robot while also using the same screen to act as the “eyes” of our robot. This would eliminate the original remote control completely, thus making the number of components of our overall design smaller.

Since RF technology has been used in many applications over the years, the amount of parts available to implement the technology is rather high. Not only is the technology readily available, but it can be implemented in many ways. This part of the paper will look at a few different RF receivers and transmitters and will weigh the factors of each so that we can pick the part that best suits our needs.

**5.3.4 Wireless Adapters**

We first decided to look at different RF transmitters that were currently on the market. The first one we encountered was an RF Link Transmitter rated at 433 MHz found at Sparkfun. The following figure shows how big it is.

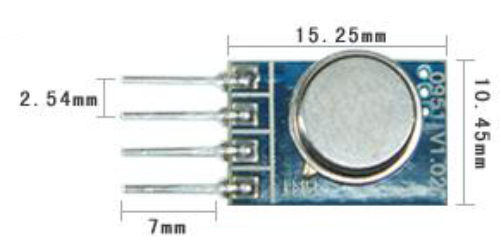


Figure 34: Wenshing Hi Power Transmitting Module (Reprinted with permission from SparkFun)

The following diagram shows the pin layout for the device.

|  |  |
| --- | --- |
| Pin | Signal |
| 1 | GND |
| 2 | Data In |
| 3 | Vcc |
| 4 | ANT |

Table 12: Pin Layout of Wenshing Transmitter

We already knew what the first 3 pins represented but after a little research we figured out that to properly transmit instructions to our robot, an antenna was needed. An in depth look at antennas will be discussed a little later on.

The following table shows some key specifications of this transmitter.

|  |  |
| --- | --- |
| Characteristic | Unit |
| Operating Frequency | 434 MHz |
| Data Rate | 8 kbps |
| Current Consumption | 8 mA |
| Output Power | 32 mW |
| Operating Voltage | 3-12 V |
| Operating Ambient Temp | -20 to +85 degrees C |

Table 13: Highlighted specs of Wenshing transmitter

This transmitter did have a couple of nice features about it. We liked the fact that the operating voltage was so broad. This would enable us to find a power source that will fit our budget rather than being limited to a certain voltage. Also, since applications of this transmitter involve being used in car keys fobs, wireless game pads and wireless toys, this transmitter may be exactly what we are looking for. We also liked that fact that the output power was low, but we also had to consider that the antenna will draw power as well.

The next transmitter we looked at again came from SparkFun. It is an RF Link Transmitter rated at 315 MHz. The pin assignment for this device was just like that of the 434 MHz component. Just like the previous transmitter, this device requires an antenna. Without this antenna, the range we get from it will be useless in comparison to what we need for our project. Since using RF technology allows us to cover great distances, we want to optimize the distance at which our robot will be able to operate. If we are able to design or acquire a good antenna, this transmitter will allow us to control our robot from up to 500 feet away. Despite the great range, none of us have experience designing RF antennas or laying one out on a board to achieve optimum performance, so we know we will not be able to gain this good of range. Despite this fact, we are confident we can at least have successful transmission from at least 200 feet away. The specs for this transmitter were also exactly the same as the previous transmitter we looked at.

The characteristics of this transmitter are relatively the same, the only difference being the operating frequency. This brought up the question of which frequency would be best for us to use. It turns out that 315 MHz transmitters are primarily used for remote keyless entry and garage door openers. Since these applications are used quite often, there is a great chance for interference. Since human life may be at stake, we want as few possible chances for interference as possible. Further research showed that 433 MHz may not be a good option either. The transmitting distance is great, at around 1000 feet, however, in the United States, 433 MHz is a popular band for amateur radio stations. Since it is impossible to predict where the bases of these amateur radio stations reside, we will still be taking a risk using this technology.

Operating at a less crowded frequency of 418 MHz would be an option, but on top of finding an antenna and receiver combination to support this frequency, we would also have to develop a whole board with a microcontroller purchased separately to implement it. Since both the 315 and 433 MHz option provide problems, we looked into other options.

We found that the best option available to us was the use an all-inclusive kit that offered a different frequency. We ended up finding an Xbee Wireless Communication Starter Kit that operated around 2.4 GHz that has been successfully implemented in past projects for only $79.99. Included with this kit are 2 Xbee 1 mW communication modules (transceivers), an XBee Explorer USB, an XBee Explorer Regulated, and a Mini B USB cable.

We liked this module a lot for many reasons. For one, the modules operate on 3.3 V at 50 mA. This would not only allow us to keep our power output low for this piece of circuitry, but to also use a small voltage source to keep our design compact. Another highlight of this system was that it allows up to 250 kbps max data rate. Since the data we are interested in is not overtly complex, thus not taking up nearly that much capacity, this gave us more than enough to work with. The range this kit offered was also very nice. With ideal conditions, it is supposed to operate within a 300 ft range. Since we decided that 100 ft of range for our robot would be sufficient, this gives us plenty of extra space, allowing us to go deeper into whichever kind of disaster area we send our robot into. The key feature enabling this great range is the built in antenna. With the parts we considered before, we would have had to implement our own antenna, and since none of us have any experience with dealing with all of the factors that generate a successfully working antenna, we felt that this was a great feature to this kit.

The Explorer USB which is a USB serial unit that will enable us to program the pins on the modules and provide us a way to interface our laptop with the robot via X- CTU software. As mentioned before, the kit also includes a mini USB that will connect the explorer to our laptop. The following figure shows all of the pins on this device. There are multiple pins in which we can input data. We will then be able to connect the data out pin to our receiver and be one step closer to interfacing with our robot. The USB will provide a simple connection from the Explorer to our laptop and will provide great portability and easy connections no matter where our robot is bound to go.

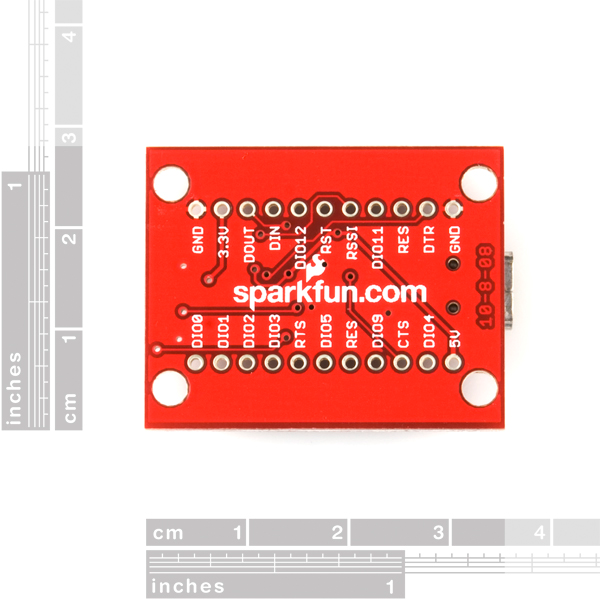


Figure 35: Xbee Explorer USB (Reprinted with permission from SparkFun)

The next part of the kit comes in the form of a voltage regulator. It will allow us to hook up a voltage from 3.3-5 V and have it regulated down to 3.3 V, which is the operating voltage that these parts operate at. Any more voltage than that will not guarantee that the part will work and anything less will simply leave us with some fancy pieces of hardware that don’t do anything. On top of just being a regulator, this component will also assist in signal conditioning. It also is equipped with a few LEDs that indicate to the user that the power is on as well as if the data in and data out pins are functioning the way they should be. This will help us with the initial troubleshoot that will be done once it is hooked up. To use this part, we simply have to take our Xbee receiver and plug it into the connectors on the regulator which can be seen in Figure 36.

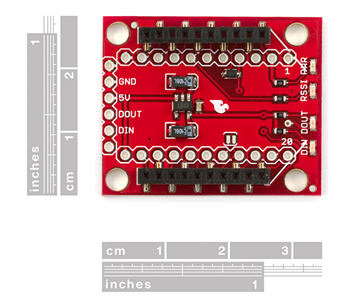


Figure 36: Xbee Explorer Regulated (Reprinted with permission from SparkFun)

The last components that come with the kit are two 1 mW communication modules. One will plug directly into our regulator pictured in Figure 36 and the other will be attached to our motor control on the robot itself. For the price, the receiver sensitivity is amazing. It is able to pick up a wide range of signals, the smallest of which would be at -92 dB. This is great because we need to be able to know that our robot will still pick up signals and operate accordingly even if the environment it is not able to allow for maximum strength from the signal. The best features of these modules in our opinion are the built in antennas. They are whip antennas, but are very small due to the high frequency at which they operate at. This is very convenient for us because we will not have to set aside too much room to accommodate the antennas. Figure 37 shows this module with the built in antenna.

Since this kit is very popular, there are plenty of guides and sample code we can use to troubleshoot our devices. The LEDs are also a nice touch since we would not have to go far into the configuration of our parts to realize that something is wrong. Any help we can get would be much appreciated since this will be the first time we will use wireless technology for our own application.

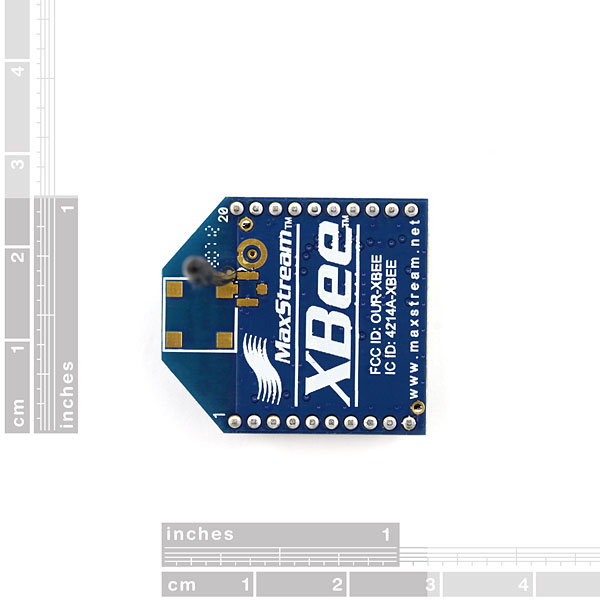


Figure 37: Xbee 1 mW Wire Antenna Transceiver (Reprinted with permission from SparkFun)

**5.4 Power Supply**

**5.4.1 Electronics Power Supply**

It was previously decided that we would keep the battery that the remote control car came with to power the vehicle in order to save time and money modifying it ourselves. However, one of the biggest obstacles we had to overcome was how we were going to power our robot’s electronics. Since the amount of real estate we have to work with is rather small, we knew that some form of battery would be effective. The problem with this is that there are just so many variations that we could use. This section will be used to highlight different types of batteries to figure out which one would work best for us.

One thing we will consider about the battery is its size. Ideally, we would like the batteries to rest underneath the chassis. However, mounting them to the back or the roof of the chassis would not be an issue if this simply will not work. The next thing we will consider is whether we would like them to be rechargeable or not. Depending on how quality the batteries are, this rechargeable feature will help save us money down the line. Not only does the size have to right, but the efficiency of the battery must be great as well. We’d rather spend the money on an efficient battery we’d have to throw away than a rechargeable one that simply doesn’t last. Further research will lead us to the right battery that will work for us.

There are currently a lot of different types of batteries available to us that can be used for robotics projects. They include, but are not limited to alkaline, lithium (both of which are disposable), lead acid, nickel cadmium (NiCad), and nickel metal hydride (NiMH) among many others. Each battery has its advantages and disadvantages and once each one is looked at carefully we will have the right one that works for us.

While looking into these types of batteries, we learned more about the rating of them. While we knew that putting batteries in series would result in more voltage, it was not apparent to us that putting batteries in parallel with each other will produce more current. This is an important realization since some of our components will need more current than others to work to their full potential. It is also important that if we are to wire our batteries in a parallel connection that we avoid self-discharging. This occurs when one battery charges another in parallel, which results in high inefficiency and could make for permanent damage done to the battery. To avoid doing this, we will have to avoid putting batteries in parallel that do not have equal charges, as one will charge the other, resulting in self discharge. It is also very important to make sure that the batteries are the same type, which will also help avoid this problem. It is also not wise to include older batteries with newer batteries, as this can also create problems.

Another new term we were introduced to was the mAh (milli Amp hour). This is the rating of how much power the particular battery can give. If a battery is rated at 1000 mAh, this would mean we could get 1 A of current for a whole hour, 2 A of current for 30 minutes or 4 A for 15 minutes. The amount of current is proportional to the amount of time the battery will last.

**5.4.2 Alkaline**

Now that we got some background information common to all batteries, it was time to start our research by looking into alkaline, or otherwise known as disposable batteries. One great thing about these is that they are very widely available and come in so many shapes and sizes. The typical voltage level for the common alkaline battery is around 1.5 V per cell, however, sizes up to 9 V are available. We decided to take a look at a typical AA battery made by Energizer.

Each unit supplies 1.5 V of power. Since our minimum required voltage on most of our components is around 5 V, at least 4 will be needed to provide the proper power as well as some form of regulator to produce the right voltage. As we learned earlier, not only is the voltage important, but the current it is able to produce is important as well.

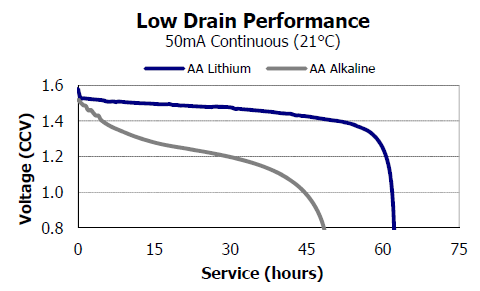


Figure 38: Alkaline and Lithium AA Battery Performance (Pending Permission from Energizer)

Since power is an important factor in our design, we wanted to keep the current consumption relatively low. This graph shows the performance of a AA battery over time. If we were to use these batteries to get up to 50 mA out of them, we would expect to get around 30 hours of use, since we would not want to run the batteries once they have discharged enough to only be at 1.2 V. This does bring up the issue that once these batteries do discharge to this level, we will only be at a total level of 4.8 V, which would not supply sufficient enough power. Therefore, it would probably be a better idea to use them for only 4-5 hours, since this would assure us that our batteries each supplied 1.4 V, leaving us with a total of 5.6 V available to us. Since the battery supplying main power to the motors of our robot will not last nearly as long as this, this is not an issue. The weight of the batteries is also very nice. Each battery typically weighs only 0.5 ounces. Therefore, this would add very little added weight to our design.

Another great thing about this type of battery is its great shelf life. We could leave these batteries dormant for years and they would still be just as efficient as if we bought them yesterday. They also work very well in a wide range of temperatures which would allow us to bring the DZERV to wide range of environments. They also provide a slow self discharge rate, which will allow us to power our electronics for a decent amount of time.

One thing we did not like about these batteries is that the cost will be relatively high simply because we cannot recharge these batteries. This would not pose a problem in the beginning, but over time, with continuous use, these would have to be replaced. Another issue that comes along with this would be disposing of the batteries. Simply throwing them in the garbage is not an option, since these batteries do contain harmful chemicals that have the potential to cause problems for the environment once they start to break down. Extra care and consideration will be needed to avoid this.

**5.4.3 Lithium**

There is another type of disposable battery out there available to us. It is the lithium battery. They are very similar to the above mentioned alkaline batteries, but provide some significant improvements. These batteries come in a variety of shapes in sizes, such as button cells, AA, AAA and 9 V sizes. However, we are only interested in looking at the commonly used sizes that are found in most consumer electronics today. One advantage this battery holds over regular alkaline batteries is that each cell can produce anywhere from 1.5-3.7 V per cell, which is over twice the voltage of the alkaline.

Not only is the voltage provided a little higher, but the information in Figure 38 shows just how much more efficient they are. If we were to run these batteries to achieve a maximum current of 50 mA, we could run these batteries for a little over 60 hours. However, since we must maintain a voltage level of at least 5 V, we must limit this value a little bit and only run them around 50 hours, since this value will allow us to still have 1.4 V from each battery available to us. When compared to rechargeable batteries, they simply last a lot longer. This will be something that we will have to consider when we make our final decision.

**5.4.3 Lead Acid**

The next potential candidate for a battery for our electronics was a lead acid battery. We were quick to realize that this would not be a good idea simply for its bulkiness. These are the batteries that are used for motor vehicles. With bulkiness comes added weight, and since our initial robot weighs less than 5 pounds, our battery would clearly weigh more than the vehicle itself. Even if we were able to use this battery, the added torque needed to get our vehicle to move would drain our motor battery relatively quickly. Another disadvantage of this battery is its high current output. Since this battery is used to achieve this, the components for our sensors and camera would simply burnout rather quickly.

**5.4.4 Nickel Cadmium (NiCd)**

Our quest for a potential power source has led us to explore the realm of rechargeable batteries as well. One such battery is the nickel cadmium battery. This type of battery is a great contender for us since it has a lot of advantages and very few disadvantages. These batteries are very difficult to damage since they tolerate deep discharge for long periods of time. Since these batteries are able to be recharged, it was important to find out just how many cycles these batteries could withstand in terms of charge and discharge cycles. It turns out that the NiCd model has the best performance in this area when compared to other rechargeable batteries. The energy density level is also quite high. This simply means that per unit of volume, more energy is able to be stored, which would help us run our electronics over a greater period of time with smaller batteries. The nickel cadmium batteries also have a smaller self-discharge rate when compared to other rechargeable ones, at only 20% per month. One of the best attributes to this battery is that it has very high current output, which would be very beneficial in powering a small to moderate sized robot such as ours.

Despite the good things to say about the NiCd, there are some disadvantages associated with it. They do tend to cost more than other rechargeable batteries. The metal itself (cadmium) is also highly toxic to most life forms. However, technology has made it so they can be used without worrying about doing harm. One of the biggest problems associated with this battery is its memory effect. These batteries store less and less charge after each recharge. This can pose a problem to us after a while since we would not be able to rely on the ratings of our batteries. The best scenario for recharging would be to wait until the battery is fully discharged, but this would be very inconvenient for us since the electronic components would cease to work a long time before this would occur. However, if we can get enough uses out of the battery despite its memory effects, this type of battery would still be a good candidate.

**5.4.5 Nickel Metal Hydride (NiMH)**

The last potential battery we looked at was the nickel metal hydride (NiMH) variety. It too is rechargeable and comes in many different forms, including the common AA size. These batteries are similar to the NiCd batteries, but instead of using cadmium as the main material used as the anode, hydrogen is used instead. Each cell for one of these batteries will provide 1.2 V.

The NiMH model has made a lot of improvements over the NiCd batteries and provides a lot of advantages for using them. For one, they have a very wide operating temperature anywhere from 30 to +75 degrees C. This will provide a great range of environments for our robot to be in where we can expect solid performance from the battery. These batteries also possess a very long lifetime at around 3000 cycles. Recharging the battery can be as fast as only 1 hour, but typically it is done over a longer period of time and since we don’t plan on having to do any emergency recharging, this will not be an issue. These batteries also have better results when it comes to memory effects. Unlike the Ni-Cd batteries, the NiMH batteries do not require full discharge every time you wish to recharge the battery. This is a great feature since we will not have to worry about the battery failing since we can recharge it daily. The only condition with this is that we let the battery fully discharge once a month. The last nice feature about these is that they are a lot friendlier to the environment then the Ni-Cd batteries since there is no lead, cadmium or mercury inside of them.

There are some disadvantages to using this battery as well. For one, they have a high discharge rate, which means they will not last as long as the NiCd batteries per charge. Long time storage of these batteries is also discouraged since they do deteriorate over time but reconditioning them (charging and discharging them several times before reuse) will help counteract this problem. These batteries are also susceptible to overcharging, but with a good quality charger, this problem will be taken care of. Also, at only 1.2 V per cell, it provides less voltage than disposable alkaline batteries, thus requiring us to have more batteries for similar voltage levels.

**5.5 Choosing a Battery**

Upon further discussing it between our group members, we decided to go with a rechargeable battery for our electronics. We know we wanted either a NiCd or NiMH battery so we browsed for parts.

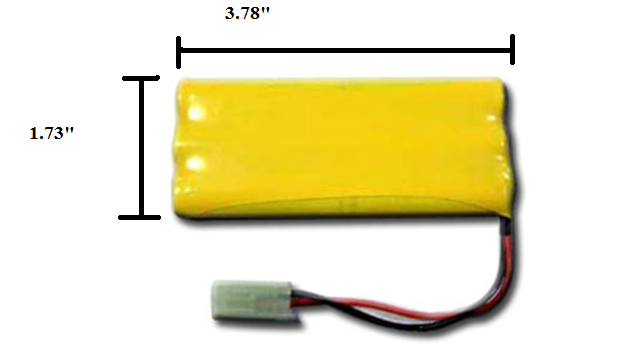


Figure 39: 7.2 V, 700 mAH NiCd battery (Reprinted with with permission from BatterySpace)

We found a 7.2 V NiCd battery from a website called BatterySpace. This battery has a very high capacity, providing 700 mAH. Since the current needed for most of our electronics is relatively low, we expect that we will get plenty of use out of this battery until it is time to recharge. The weight of this battery is only 4.2 oz, which is great. If we need to mount this battery on the rear of our chassis, it will not put too much weight on the back of the vehicle. With the right charger, this battery can be up and running at full capacity in only 1.5 hours. This is beneficial because if we were to buy 2 of these batteries, one will be finished charging by the time the other battery runs low. The price of this battery allows for us to do this, since they are only $6.95 each. This model is also appealing to us since it will match our chassis color. This would allow us to put the battery pack on the back of the vehicle if it simply won’t fit with everything else underneath the chassis and still blend in with the initial chassis color.

Even though this seemed like a great candidate for our battery, we decided to explore NiMH batteries as well, since they do offer some advantages over the NiCd variety. From All-Battery.com we found this 7.2 V NiMH battery made by Tenergy.

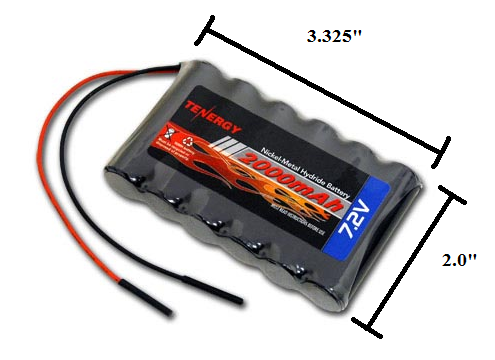


Figure 40: Tenergy 7.2 V 2000 mAh Side-by-Side NiMH Battery Pack (Reprinted with permission from All-Battery)

This particular model has 6 AA batteries at 1.2 V per cell that will provide a total of 7.2 V. The side-by-side configuration seen in Figure 40 enables us to get more current out of the battery. The greatest advantage we would have using this battery is that these cells are not affected by any memory effect. This would allow us to get the full use out of our battery without having to worry if the last charge we did on it might be our last. This battery will also give us 2000 mAh, but since we are not looking to use a lot of current, we hope to get lots of use out of this battery due to the small current levels we need for out electronics. Like the NiCd battery we considered, it also weighs 4 oz and because of this it would also be fine to attach it to the back of the chassis if it simply won’t fit underneath it. Charging the battery would not take long either since the average time needed to do so is only 2 hours. The price of this battery is a little more than the NiCd battery, but this was to be expected since the overall performance is better. This model is only $3 more, but should cause us less trouble than the NiCd battery, so we decided to go with this model.

**5.5.1 Battery Charger**

The next step for us once we found out battery was to find a sufficient charger that was both efficient and affordable. This led us to a simple charger that would only charge a 7.2 V NiCd or NiMH battery. We found such a thing at BatteryJunction.com.

We initially decided to go with this charger, but further research decided us to look at other models. For one, this charger limited us to just one battery size. This would not be a big deal, but if we wanted to upgrade the battery that would power our car motors, we might not be able to use this charger. Also, overcharging NiMH batteries can result in problems and this particular model didn’t have good enough features available to prevent this in a way that we were comfortable with. This charger only costs $4.18 which made us question the overall quality of the product. Our group collectively made the decision that spending a little more on our charger to provide us with added range and better features would be the wiser decision.

When researching NiCd batteries, we were pleased to see that BatterySpace also recommended a charger for our NiMH model battery that provided us with the range and features that we were looking for.

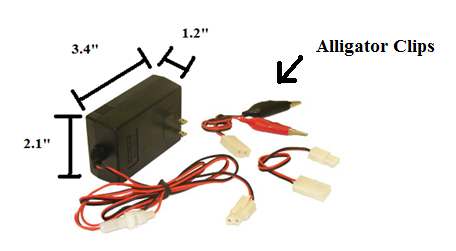


Figure 41: Charger for 6-12 V NiCd or NiMH Batteries (Reprinted with permission from BatterySpace)

The above figure shows the dimensions of this charger. It is quite compact and even comes with alligator clips that would allow us to charge any battery that fits the chargers specifications. This charger will support any model of NiCd or NiMH between 6 and 12 V. This offers us a lot of range and flexibility for any batteries we may use, including the 6 V rechargeable battery that comes with our remote control car. One great feature of this charger is that the charging process is automatically cut off by negative delta V when the battery is fully charged. This is really nice because the NiMH batteries we looked at can be damaged if they are overcharged. This charger also offers protection in case we accidentally attach the battery to the wrong terminals on the charger. This charger also uses pulse and negative pulse technology which allows for protection against overheating. Doing this also increases the battery’s life cycle, and since memory effect can be an issue, this is very beneficial. The fact that it only costs $19.95 also makes this the best candidate for our charger.

**5.6 Antenna**

Since we found a micro transmitter and receiver it was now time to find an antenna that would assist in the transmission of data that our microcontroller received from the sensors to our laptop. There were many factors to consider while doing so. The only solid thing we knew about the antennas in question was that we would like them to be surface mountable. This would require that the antenna be really small, and since our transmission frequency is quite high at 916 MHz, this would make the required antenna size rather small. We found out that there are many options available to us that will allow us to achieve a surface mounted design. However, since we have no previous experience with antennas, we decided it was best to research each one to avoid disappointment in the future.

**5.6.1 Ceramic Antenna**

The first antenna we stumbled upon was a 916 MHz ceramic surface mount antenna.

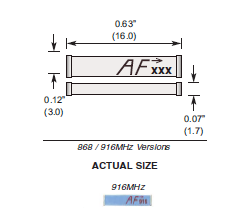


Figure 42: 916 MHz Ceramic Chip Antenna by Antenna Factor (Permission to reprint pending from Digi-Key)

This figure represents a blown up picture of the ceramic antenna. It is one of the world’s smallest high performance chip antennas and works with many applications including Bluetooth, 802.11, home RF, Zigbee along with others and only costs $1.99. This antenna uses Low Temperature Cofired Ceramic (LTCC) technology, which embeds the antenna in a ceramic substrate. This technology enables the operating temperature to be fairly good. It will operate anywhere from -40 to +85 degrees C. It also has a 10 MHz bandwidth which gives us a decent range. It also has the impedance level required for our micro transmitter, which is 50 ohms. The antenna does have 2 pins, one which would attach to the transmitter or receiver and one pin is to be soldered down only to add extra

support. No electrical connection would be needed. Despite the advantages, there were some concerns with using this model. For one, the antenna itself is really small. It is only 16 mm long and 3 mm wide. Mounting this on our PCB board may prove to be too daunting. Its small size also raises the question of how fragile it is. We need our antenna to be able to be durable enough to withstand some uneven terrain and this may not be what we are looking for. There are a lot of layout considerations that we must follow as well which also make it undesirable. The area underneath the antenna should be free of components, traces and planes in order for this antenna to work properly and even then, it requires the use of a very small microstrip trace on the PCB.

**5.6.2 Planar Antenna**

The next model we considered was a 916 MHz planar antenna called the Splatch made by Linx Technologies.

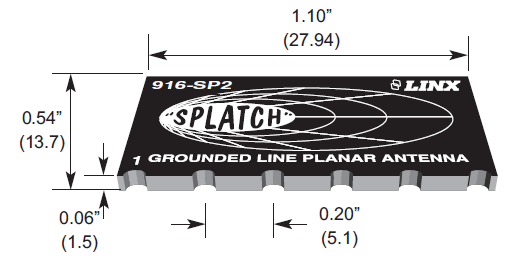


Figure 43: 916 MHz Planar Surface Mountable Antenna (Permission to reprint pending from AntennaFactor)

Pictured above in Figure 43 is what the 916 MHz Antenna looks like. It is surface mountable, just like we wanted and provides some desirable features. The Splatch is able to achieve good performance by using a grounded-line technique. It also matches the required 50 ohm impedance needed for our micro transmitter and receiver. The bandwidth of this antenna is 30 MHz, which is more than that of the ceramic antenna, which makes this one more desirable in case we are unable to tune exactly to 916 MHz. However, if we were able to mount this antenna in a way to gain optimum performance, we are looking at a VSWR of around 1.9. The VSWR rating of an antenna is a very important number. It stands for Voltage Standing Wave Ratio and is a function of the reflection coefficient which describes the power reflected from the antenna. Ideally, we would like this number to be really small and tuning our antenna to the center frequency of 916 MHz would provide us with the best VSWR.

Using the equation VSWR = where Γ is the reflection coefficient, and the VSRW is around 1.9, we calculate that a little less than 11.1% of the power of the antenna will be reflected. This is important to us since this power will not be able to be transmitted to our receiver and thus will be no use to us. Research has shown us that a VSRW of less than 2 represents a great antenna match and thus is very desirable.

The cost of this antenna is also great. It would only cost us $2.08. The mounting for this antenna on our PCB is also simplified. Like the ceramic antenna, no ground plane or traces are allowed underneath the antenna. This shouldn’t be a problem because the antenna is rather small, but not too small. It is 1.10” long and 0.54” wide and would not take up a lot of space on our board. It also offers plenty of vias that would go to the ground plane.

**5.6.3 Whip Antenna**

The last type of antenna we considered was the whip antenna. This antenna consists of a single straight, flexible wire or rod that is usually mounted above the ground plane. These antennas are found in a lot of everyday items such as walkie talkies, cordless phones and are also used for car radios. Since what we are trying to do definitely relates to these applications, we decided that looking into this antenna would benefit us.

We found a 916 MHz whip antenna but it was not surface mountable. It connects to the receiver or transmitter by an RP-SMA connector. This definitely posed a problem to us, but further research showed that there is a part available to us that will allow for an initial RP-SMA mount that will relay to a surface mountable part.

The part that we found that will be used to mount the RP-SM A based antenna to our PCB board is rather small, which is great since all space on our board needs to be optimized. It supports frequencies up to 12.4 GHz and provides us with 50 ohm impedance. We know how durable whip antennas are, since we all have one on our vehicles and they are able to endure harsh conditions, which is crucial if we are to take our vehicle into disaster areas.

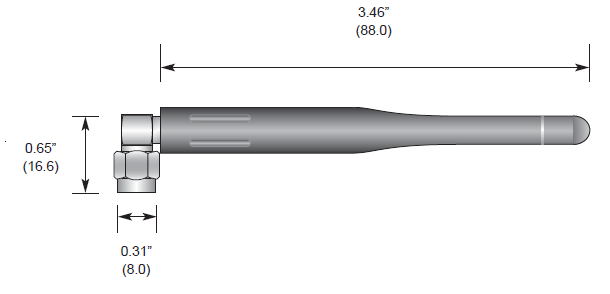


Figure 44: 916 MHz ¼ Wave Whip Antenna (Permission to reprint pending from AntennaFactor)

This is a picture of the whip antenna we are considering. From the figure, it is clear that we cannot simply mount the antenna directly to the PCB board, so the connector above is necessary. With this connector in place, we would have a couple of different options for the direction our antenna points in since it has a right-angle mount. This flexibility will allow us to find what angle will give our antenna the best gain, hence increasing the range that our robot will be able to travel while still relaying readings to our laptop. If the antenna does need to be perpendicular to our PCB board, a simple modification to the chassis will allow it to protrude outside of the body of our remote control car.

Although the center frequency for this antenna is at 916 MHz, there is a bandwidth of 150 MHz. This will allow us to test out a few different frequencies and adjust if there is a lot of noise with one we choose. However, we still need to stay near this center frequency for the best VSWR rating. At the center frequency, we have a typical rating of less than 1.7 on the VSWR scale, which makes this the best antenna we have currently looked at. The length of the antenna is only 3.46” long which makes it much easier to see and deal with than the much smaller chip antennas. In case we do need more space between the connector on the PCB and the antenna itself, we can always get some wire that will also give us 50 ohms of resistance so we don’t have to worry about how and where the PCB is laid out. This antenna costs $7.39, which is considerably more than the others. However, the added benefit of being able to move the antenna where we see fit is very appealing to us. Also, the broader bandwidth this model offers gives us even more flexibility that the chip antennas cannot. The biggest problem involving this antenna is the fact that all of these extra components will only add noise to our signal, thus making these great specs meaningless; therefore, we will not go with the whip antenna for our design. We have decided to go with the Splatch planar antennas since it is a little bigger than the ceramic one and despite its small size, these things are supposed to work great. Proper prototyping and testing will have to be done to confirm just how well it will perform, but for now, this is the model we are going with.

**5.7 Motor Controller**

A key component to our design will be our motor controller. This will be the part that will enable us to drive our motors. This section of the paper will have us looking at a motor controller that we felt would do a very sufficient job at guiding our robot through various disaster zones. There are so many important functions that this component must be able to perform so this part must be good quality. The functions that this part provides goes beyond just making the motors move. It also involves adjusting the speed at which our robot is moving, selecting the direction of our vehicle (forwards or reverse), regulating the torque or the motor as well as having safeguards to prevent our motors becoming overloaded and therefore burning out.

The remote control car that we are using for the main body of our robot has two motors. One is used to steer the car and the other is used to steer it. In order to tap into the pre-existing system and make these motors function wirelessly, we needed a controller that could handle two separate motors. This will take some work and a little finesse but we feel that we will be able to do it. This method will save us a lot of money and the hassle of assembling our own chassis and other components so we are definitely willing to make it work. We were able to find such a motor controller made by Polulu. Figure 45 shows just how small the component is.

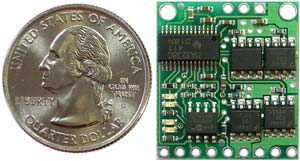


Figure 45: Pololu Low Voltage Dual-Serial Motor Controller

This motor controller would set us back $31.95, but the benefits of using this component would be well worth that. The baud rate that the controller functions at is between 1200 and 19200, which we can set up while using software from our wireless component. We will be controlling the chip with Xbee wireless products that will then relay the desired function to our controller which will then activate the motors. The Polulu model has nine pins and they are as follows.

|  |  |
| --- | --- |
| Pin | Function |
| 1 | Motor Supply (0-7 V) |
| 2 | Ground |
| 3 | Logic supply (3.0-5.5 V) |
| 4 | Serial Control Input |
| 5 | Reset |
| 6 | Motor 1, Positive Input |
| 7 | Motor 1, Negative Output |
| 8 | Motor 0, Positive Input |
| 9 | Motor 0, Negative Output |

Table 14: Motor Controller Pin Layout

Table 14 shows that this controller has the capability to control two motors. At this point in time we are still going to use the 6 V supply that comes with our remote control car therefore this fits in the allowable range of motor supply voltage that this part can handle. The only other real concern with this part was the logic supply voltage. This is the voltage that will be seen by our microcontroller. The operating range for our PIC processor is 2-5.5 V and since most of our sensors run at 5 V, the same supply for the sensors can be connected to the logic supply pin without any worry of damaging the part. The serial input pin will be connected to our Xbee receiver. This way a binary instruction can be converted to a voltage level that will then go on to power our motors.

Figure 46 shows the rough schematic that we will follow in order to get our robot to move. First we will give a command on our keyboard to move in a certain direction. This information will be transmitted from our Xbee module to its receiver located on our robot. This signal will then be sent to the motor controller where the data will be processed in such a way that it will then travel to our motors, allowing for the initial instruction to be implemented. If all goes well, the correct action by the robot will be taken and we can then provide further instructions to guide our robot wherever we would like it to go.

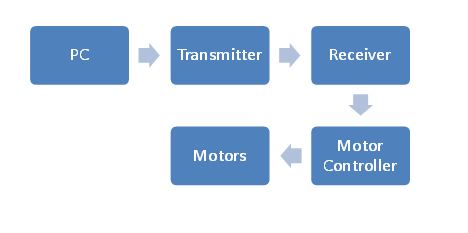


Figure 46: Order of Operation for Wireless Control

**6.0 Project Prototype Construction and Coding**

Of course, the real work for our project actually comes in actually physically constructing and testing our robot. This is the part of the process where we finally stop designing and start building. This is the most important stage of the process because it is where we will find out whether what we have designed will work the way we designed it. It is also the part where all the learning that we have done throughout our undergraduate careers as engineering students culminates. It is this portion of the Senior Design process that separates the students from the engineers. It is this portion of the process that we’ve been waiting for, and it is here that we will be able to put our abilities to the test in order to proceed with building something that we had designed, and end up with a fully-functional prototype to demonstrate before our Senior Design committee.

**6.1 Strategic Locations**

Obviously, one of the major things we would need to look into before settling down to begin construction on our project is where we would actually meet up to work on this robot. Clearly, a centralized location would be the best option for the three of us, so neither one of us would need to commute too great of a distance to work on the project. Secondly, we would need a place that had the tools we would need to put our project together. This includes soldering irons/equipment, power supplies, multimeters, and oscilloscopes. While the absence of these tools would not necessarily hinder us from working on the robot, they would certainly be a welcome addition to our working environment. Therefore we settled in a few places where we could work.

**6.1.1 EECS Senior Design Laboratory**

One of the first places that we considered doing a majority of our construction work is the Senior Design lab for electrical and computer engineering students located on the second floor of the Engineering I building at UCF. The reason for this is that the lab contains many workstations for senior design groups, and all of the equipment we could possibly need for building our robot. This also satisfied our location requirement, as we all spend a great deal of time on campus at UCF as it stands. This in turn would provide an extra layer of convenience for our needs. In addition, working on our project on campus in the presence of other engineers would give us the opportunity to ask for the opinions of other electrical engineering students, as well as seek advice in troubleshooting issues with our project. This could prove to be quite invaluable as we go through the process of building and testing our robot. While we were certain that we would be spending most of our time in the Senior Design Lab, we had also taken into consideration that we wouldn’t possibly be able to spend all of our time there. It then became necessary to investigate other possibilities for build locations.

**6.1.2 Group Members’ Homes**

We therefore began looking into what we would be able to do as far as working from the home of one of our group members was concerned. After a few minutes of discussion, we learned that Michael lives in the Sterling apartments right across the street from campus. This would be an ideal place for us to get work done, as it is certainly a convenient location for Michael, and is a place that Robert and Marcial would be able to reach easily from almost anywhere. The only issue is that because this is an apartment, space for the build may be rather limited, so we decided that in the event we weren’t able to work in the senior design campus, this would most likely not be the wisest choice. It was clear that we would probably need to find another place to work.

We then considered Marcial’s house. He lives in a house in the Lake Nona area. While this is not quite a centralized location for us, he did mention that he has access to a wide variety of tools and also has a garage. Having a garage in which to work could prove to be quite helpful, as this would give us a great deal of space in which to work, and having access to the tools that we’d need, with the exception of an oscilloscope, would also make things easy for us. After a few more minutes of debate, we decided that this certainly would be a great option, but the distance would be too great for the other two members to travel. We therefore concluded that we wouldn’t be able to work on our project from Marcial’s home.

We then looked into seeing whether we would be able to work from Robert’s home. It was discovered that Robert lived in the most centralized location out of the three of us. Like Marcial, he also lives in a house with a garage, and therefore, we’d be able to utilize this space for our build. The major advantage to working at Robert’s home is that even though he did not have all the tools Marcial has, he lives in more of a centralized location for all of us to meet up, if required. Marcial would be able to bring the tools from his home to complete work there if necessary. We then decided that we would work from Robert’s home in the event that we wouldn’t be able to get something done in the Senior Design lab on campus.

**6.2 Parts Acquisition**

**6.2.1 Vendor Selection**

Another one of the more important decisions in the overall design process is determining where parts will be purchased. Where will we purchase all of our parts form? Will we buy them in a store, or online? What kind of lead time will there be on the parts that we do need? Will we be able to ensure that we’re able to have all our parts in a timely manner so that we can have time to physically build the robot and test it out before our project is due? These are all questions to which we now have answers, as a great deal of research was done into what would be the best way to go about purchasing and acquiring our supplies. For starters, we had an incredibly wide variety of vendors from which to choose. We could purchase parts directly from the manufacturers, such as Microchip for our PIC microcontroller, or Texas Instruments for various other parts such as voltage regulators. While this may have resulted in slightly cheaper pricing on the components we are going to be purchasing, we decided that it would be more simple and much less of a hassle to order as many parts as possible from a distributor such as SparkFun or Digi-Key. This would allow us to lump many components into a single order and allow us to have the components all shipped and/or delivered at or near the same time. Additionally, due to the fact that we would be consolidating many separate orders from many different vendor into a single or very few orders from one or two vendors, this would likely save us a great deal of time and money with the overall building process for this project. We did some further investigation into the companies we would likely be purchasing parts from, and after a great deal of discussion, we decided we would need to look to a few different vendors to purchase all the components that we’ll need. Our reasoning behind this is very simple. SparkFun is a very highly-trusted vendor for electrical components such as microprocessors and LEDs, among other things, including modeling kits for amateurs. This alone, combined with many word-of-mouth referrals, and some of our members’ own personal experience with SparkFun all led to the decision that we ultimately made in purchasing our components. Digi-Key is also another vendor from which we will be purchasing components. Because SparkFun and Digi-Key are such well-trusted sources, we will also be able to ensure that our parts will arrive quickly as needed, which will save our group from an incredible amount of unnecessary stress the last few weeks of the construction phase, as opposed to the possibility with some unknown vendor that we may not receive any of our components in time to actually physically put the components together on our Printed Circuit Board. Also, based on personal experience, it has been noted that SparkFun’s pricing is actually quite competitive, and therefore will hopefully save us a great deal of money in the long run, as we are going to be working with rather limited resources and a fairly tight budget. However, we will not be able to purchase all of our components from distributors like SparkFun and Digi-Key, and we will need to turn to the actual manufacturers for these components. One such component is the camera that we will be using. Because SparkFun does not carry the camera that we will be using, it will be necessary to go ahead and purchase this directly from X10. We will also be doing this with our microcontroller as well. We are going to be purchasing the PIC microcontroller and a programming kit for it directly from MicroChip, as this is the most direct way to have both of these parts shipped together.

**6.3 Printed Circuit Board Vendors**

Another major component for our project is the printed circuit board on which we will be placing our components. This was a significant task for our group to accomplish, as none of us had any previous experience in PCB design, and so we needed to do a great deal of research on companies to make our board, as well as the software packages available on the market to aid in laying the board out.

**6.3.1 ExpressPCB**

One of the first companies that we came across in our research was ExpressPCB. This company offers quite a few features that would be useful for us in designing our board. For instance, their website features a section containing resources on getting started with PCB design. In addition, ExpressPCB offers their own line of software tools for designing circuit schematics, as well as software for actually laying out the PCB itself. Additionally, these software tools are completely free of charge, and the ExpressPCB software even has a built-in ordering functionality so that users can actually use the software to order their board after laying it out.

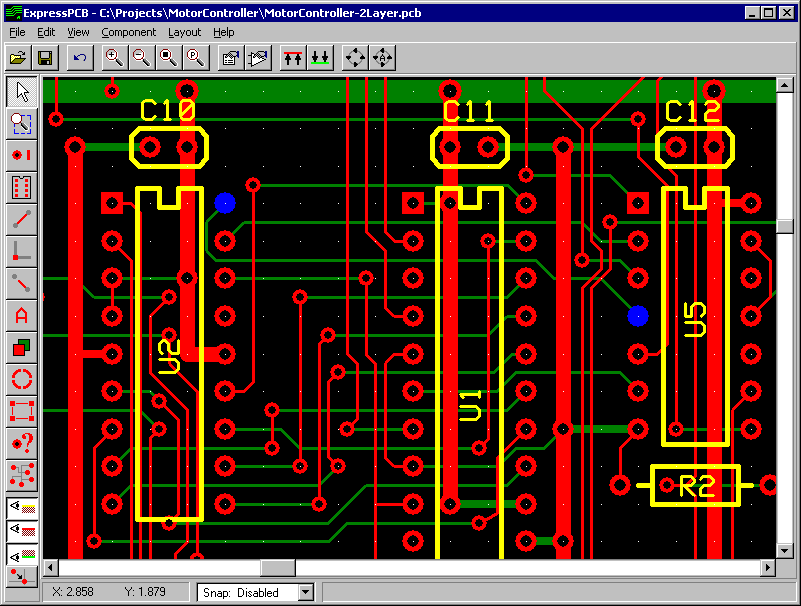


Figure 47: PCB Layout Software (Reprinted with Permission of ExpressPCB)

The figure above demonstrates some of the functionalities of the ExpressPCB software. There are built-in functions that allow the user to choose components and place them on the board. Here, the user can also add traces, and specify how wide they need to be, among other things. This software also offers countless component libraries in order to facilitate the board layout process.

Additionally, ExpressPCB also offers quite a fairly wide range of solutions that can fit almost anyone’s PCB needs. They offer their MiniBoard line, offers two varieties: the MiniBoardPro, and the Standard MiniBoard. They both come standard with two layers and plated through holes. The major difference between the Standard MiniBoard and the MiniBoardPro is that the MiniBoardPro includes solder mask layers and a white silkscreen layer, which gives the boards a more professional look. There is also a sizeable difference in pricing, as the two-layer MiniBoardPro comes with a set of three identical boards for $75, and the Standard MiniBord comes with a set of three boards for a fixed price of $51. ExpressPCB additionally offers a four- layer version of the MiniBoardPro with layers for a ground plane and power plane. This could prove useful, as it would enable us to better isolate components that needed it. The additional planes in the four-layer board do cause a fairly significant increase in price. A set of three of these boards would cost $98. The major disadvantage to these MiniBoard models is that the size must be limited to 3.8 by 2.5 inches, which means that we would need to ensure that all our parts would fit on such a small space. An alternative to this would be to go with one of their standard boards, which have much fewer limitations. Their Standard boards are exactly like the Standard MiniBoards, but these are scalable, with a pricing structure based on the physical area of the board. This gives us a little more room to work with as far as the size of our boards is concerned.

One of the biggest factors about choosing a PCB vendor is the estimated lead time. For most PCB models, they offer to have the boards shipped the following business day, which is actually a considerably fast shipping time.

**6.3.2 Sunstone Circuits**

Another PCB manufacturer that we investigated is Sunstone Circuits. We first stumbled across this manufacturer by accident as we began our search, but this particular manufacturer offers a great deal of support for their customers’ design needs. Similar to ExpressPCB, they also offer a design tool called PCB123 that enables users to lay out their own PCB and then order using the designs they’ve created. It even includes a built-in tool that allows customers to calculate the total cost of the board before they order. The remarkable thing about this software is that is completely free to use, and includes much more functionality than many of the other leading PCB CAD software suites available on the market such as CadSoft’s EAGLE software. This would definitely prove useful in laying out our board, as we are looking to reduce costs wherever we possibly can, and having free CAD software would be helpful for doing this. The software also is very robust, featuring a Design Rule Check tool that checks to ensure that the board being laid out meets many essential design criteria that are crucial to PCB design.

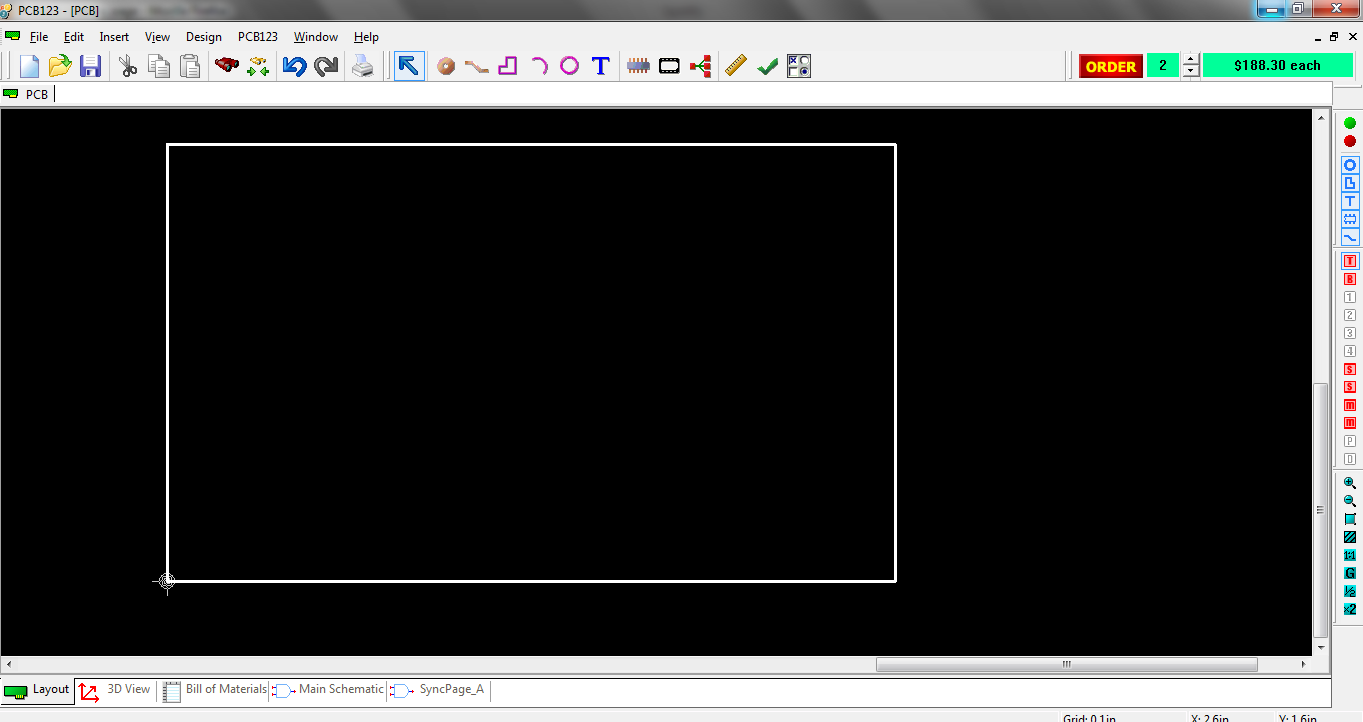


Figure 48: Screen Capture of the PCB123 Software

The above image is a screenshot of the PCB123 environment, showing all the various design features, including the Design Rule Check tool, the various CAD tools, and even the per-unit cost for such a board. Unfortunately, as can be seen in the screenshot, pricing with Sunstone is a bit on the expensive side. While the board we’ll be ordering will be much smaller than the one depicted in the image, we have found through experimentation, that we’ll still be spending roughly $180 per board with a two board minimum via this software’s ordering tool. This proves to be far too expensive for our design, and as a result, we collectively decided that we will no longer be taking Sunstone Circuits into consideration for our PCB design.

**6.3.3 PCBFabExpress**

The final PCB manufacturer we investigated was PCBFabExpress. Based on the experiences of students who have utilized this manufacturer before, we have decided that that this could possibly be a viable option for us. In addition to the word of mouth referrals that we have received for this manufacturer, we have also discovered that there is a discount available for students. Being the thrifty Senior Design group that we are, we decided that this was worth a further investigation, and found that we could actually receive up to $150 off the total cost of our board just by purchasing with them. This alone is an incredible reason for us to go with PCBFabExpress. While they do not offer their own free PCB CAD software like the other manufacturers we’ve looked into, they do offer assembly service, either via customer- supplied components, or as a turnkey service, meaning that they would supply the parts. While this would be convenient for us, we feel that this would not be necessary for a few reasons. First, this would likely drive up the costs for us drastically. Additionally, having someone else assemble the printed circuit board for us would almost entirely defeat the purpose of the Senior Design process, as assembling a circuit board is a skill that we as electrical engineering students should acquire throughout the course of this process.

**6.3.4 Final Decision**

After a great deal of research and consideration, we had narrowed down our choices to either ExpressPCB or PCBFabExpress, as these were the two manufacturers that seemed to provide the most promise to fulfill our needs. In the end, we decided that cost was our most important deciding factor for the PCB manufacturer, as it had been for a vast majority of other things for our project.

While we are aware that lower cost may mean that the quality of the work done may not be as high as if we spend a little more money to go with a more expensive vendor, we realized that the quality of the ExpresspCB boards do seem that they will suffice for our purposes, and we are therefore certain, that this will fit our needs. Based on the experiences of other senior design groups, we feel that their work, as well as their pricing will be perfect for our robot. More specifically, we will be going with the MiniBoardPro model, as this will give us the look that we are going for in terms of having something professionally done. Additionally, the pricing for these boards fits well within our budget, and we feel that this will result in the perfect board for our needs.

**7.0 Testing**

This part of our document will be set aside to discuss the various testing methods and procedures used to test every individual component of our design. This will include checking the durability, quality and overall performance of each piece. Each individual component must be checked first before it is to be integrated into something else. If a problem is found in this early stage, we will be able to troubleshoot so much easier and either debug any coding issues or trace any hardware issues. A series of tests will be done to ensure that our parts are in working order. Charts will be set up to document the progress of what we should see compared with what we do see and if standards are not met, further troubleshooting will be done until it has been decided that it is not the wiring or coding that is incorrect, but simply due to the parts limitations.

**7.1 Chassis**

The testing of the chassis will prove to be very important. We know that the initial chassis is stable, since it was manufactured to be that way, but during the building process, it may be compromised. Until we receive all of our parts and start assembling them, there is no guarantee that the space underneath the chassis will be enough to house all of our components. In order to make everything fit, we may have to make additional modifications that will lead us to make sure that the body of our vehicle will not fall off and expose our circuitry. Modifications may include raising the chassis up several inches or allowing for additional space surrounding the chassis. The former will make our robot more top heavy and extra precautions will have to be taken to account for that. The latter would pose the problem of making our robot wider, thus limiting certain areas our robot can go.

First, we will see if the robot can support its own weight by picking it up and moving it around. This will provide a quick check to see if any components come lose or if there is any violent shifting of components underneath the chassis. Once this is complete, we need to see how steep of an incline our robot can handle. This includes going up a hill and driving on an incline. Although we will not be completely certain of the threshold of incline our robot can handle while it is in a disaster area, at least we can make a better educated guess before we send our robot once we can see the projected area on our laptop screen from the camera. Next, we will check to see how much weight our robot can support and still function to our standards. Debris is always a factor in a disaster area, so again, we can use our camera to see if there will be falling debris that our robot can’t handle.

The following data in the table shows our testing procedure as well as our expected

results.

|  |  |  |
| --- | --- | --- |
| Test | Description | Expected Result |
| Carry and Shake | Pick up vehicle and  shake to see if  components are loose | No parts will move  any considerable  amount |
| Forward Motion  Incline | See what kind of  incline our robot can  overcome | Successful climb  resulting in further  exploration of disaster  area |
| Sideways Incline | See what kind of angle  our robot can handle  while driving on steep  slope | Continue motion  regularly unless robot  topples due to too  much incline |
| Weight Threshold | See what kind of  weight our robot can  support while still  being able to  function | Either the robot will  continue without a  problem or debris will  obstruct our motion |

Table 15: Chassis Testing

**7.2 Speed Test**

There may come a time when our robot is investigating an area where we may need to get out in a hurry. Other times, we may need to go very slow in order to overcome obstacles or dodge them altogether if the space allotted is low. Our initial remote control car is rated at 7 mph at full speed, however, added weight will be added to the initial design which will either allow us to run at 7 mph for a shorter period of time due to draining the battery faster or it will run below this speed. The terrain the robot is working in will also have an effect on the overall speed of our robot; therefore, it must be tested in different environments. Determining the speed will take a little effort on our parts. First, we will need to measure a set distance somewhere between 30 and 50 feet and mark it. Then we will be able to place our robot at the start and measure the time it takes for it to cross the finish line. We will then be able to take this average speed and convert this number into mph. This test will be done on flat surface, dirt, grass and gravel, since these are all possible environments our robot will be able to endure.

Shown below in the table is our testing procedure we will use to determine speed in

different environments for different situations.

|  |  |  |
| --- | --- | --- |
| Test | Description | Expected Result |
| Speed (Flat Surface) | Max and min speeds  will be determined  on flat surface | Max speed: 5-6 mph  Min speed: 1 mph |
| Speed (Grass) | Max and min speeds  will be determined  in grass | Max speed: 4-5 mph  Min speed: 1 mph |
| Speed (Dirt) | Max and min speeds  will be determined  in dirt | Max speed: 4-6 mph  Min speed: 1 mph |
| Speed (Gravel/Rock) | Max and min speeds  will be determined  in gravel or rock | Max speed: 4-5 mph  Min speed: 1 mph |

Table 16: Speed Testing

**7.3 Battery Life**

It is important that we test the lifespan of both the battery that will power the motors of our robot and the battery for our electronics. To test the battery for the motor we will simply drain the battery three separate times at varying speeds, one being full speed, the next being minimum speed and the last being somewhere in between. The results determined from this test will be the basis for how long our robot will run and at what speed. We will also have to test the battery for our electronics. We will do this by draining the battery completely for this as well. We will know our battery is no longer supplying the necessary power when our sensors fail to gather and/or deliver results to us. Since the batteries will be independent of each other, there will be no need to do any kind of hybrid testing between the two. The table shows the test procedure we will use to rate our battery life.

|  |  |  |
| --- | --- | --- |
| Test | Description | Expected Results |
| Battery Life: Full  Speed (Motors) | Constant motion  until battery is  drained | 30-45 minutes |
| Battery Life: Low  Speed (Motors) | Constant motion  until battery is  drained | 1-1.5 hrs |
| Battery Life:  Moderate Speed  (Motors) | Constant motion  until battery is  drained | 45 minutes to 1 hr |

Table 17: Battery Testing

**7.4 Battery Charging**

Not only will it be important to know how long our batteries will last, but it will also be important to find out exactly how long it takes for our batteries to recharge. For our 7.2 V NiMH battery rated at 2000 mAh, the expected charging time should be right around 3.5 hours. The remote control car that we purchased does not give a specified time that it takes for the battery to charge, so additional testing will be required to find out this information. There is a status LED on our 6-12 V charger so we will simply let the battery die and start charging it, recording the time it takes to fully charge. We are currently unsure of what the motor battery charger has in terms of alerting us of its charging status, so we will record these results when they can be determined. If we find that the motor battery is compatible with our 6-12 NiCd or NiMH charger we will check to see which one is more sufficient.

|  |  |  |
| --- | --- | --- |
| Test | Description | Expected Results |
| 6 V Motor Battery | Measure exact time  it takes to recharge  battery. | Unknown. |
| 7.2 V Electronics  Battery | Measure exact time  it takes to recharge  battery. | 3.5 hours |

Table 18: Battery Charger Testing

**7.5 Video Camera**

The video camera will be our sole way of determining the next direction we will send our robot in, so checking to see if it is accurate will be crucial. We need to have a data rate good enough in order to allow for closest real-time stream as possible. We will be able to test the quality of our stream by placing in our robot in an environment where we do not need the camera to see where the robot is going. Doing this will enable us to see how much lag we are experiencing as well as observing the overall quality of video our camera is able to give us. We will also need to make sure that the range of motion of our camera is accurate. To do this, we will set up a simple circuit in which we will ask the microcontroller to move our camera from side to side with a few simple commands. The following information in Table 18 outlines the tests that will be taken to test certain specs of our camera.

|  |  |  |
| --- | --- | --- |
| Test | Description | Expected Results |
| Data Rate | We must figure out  quality of our video  stream to laptop | Stream will be good  enough to make real  time decisions. |
| Visibility | We must make sure  that the environment  we are in can be  properly represented. | Environment will be  recognizable. |
| Camera Angle | We must see if  proper angling of  camera can be  achieved. | Camera will have a  viewing angle of 60  degrees. |

Table 19: Video Camera Testing

**7.6 PCB Testing**

This may very well be the most crucial part of our testing. If our PCB board is not in working order, all of our electronic components may not work in unison, which would result in lackluster performance in our robot. The testing of our board can take place once all the components are soldered in their correct place on the board. The easiest way to check if we have valid connections is to use a voltmeter and/or ammeter to see if there are readings. Once we receive these readings, we will use our best judgment to see if these readings make sense for the individual component that is being tested. If there happens to be a problem with any connection, we will attempt to salvage it as best we can. This is a last ditch effort however as individual parts will be tested with one another during the prototype phase.

If we do have a problem with our initial PCB, actions must be taken in order to fix it. To save money, schematics will be retested first. This will help us trace our connections on our PCB board. If we did miss a connection, we will have to order a new board, which is not desirable. Therefore, the utmost care will be taken before we decide to solder all of the components to our board. If we determine that our PCB layout is correct we will proceed to the proper testing. The information in Table 20 shows what will be done in order to test our PCB.

|  |  |  |
| --- | --- | --- |
| Test | Description | Expected Results |
| Check for loose  components | Make sure that no  component on the  board shifts or falls  off | Parts should remain  soldered to PCB board |
| Test for input and  output voltages and/or  currents | Voltmeter or ammeter  will be used to make  sure parts are working | Meters should pick up  readings that make  sense based on the  part |

Table 20: PCB Testing

**7.7 Wireless Control Testing**

Our XBEE transmitter and receivers must both be configured in a proper manner to ensure successful communication with each other. The following tests are tests which we will perform in order to ensure our wireless modules are properly configured, communicating, and working.

|  |  |  |
| --- | --- | --- |
| **Test** | **Description** | **Expected Result** |
| Base Computer Communication | Send simple code to transmitter connected to base computer via USB | Code should be sent to receiver and seen in the robot |
| Motor Controller Communication | Send forward command to receiver | Wheels should move forward |
| Module Communication | Test the communication between the TX and RX on separate computers | Data is properly sent and received |
| Range | Find the maximum range of communication between TX and RX | 200 feet |
| Climate Control | Run the modules in various temperatures and environments | Both the RX and TX should still properly transmit information |
| Temperature Control | Run the modules in various environments and temperatures | Measured voltage and current should still be as expected |

Table 21: Wireless Control Testing

**7.8 Sensor Testing**

In order to test our sensors, we will be utilizing an Arduino mega1280 development board, which one of the group members had acquired from a previous project. All of the sensors but one come in a through-hole part. The only sensor, which comes only in surface mount form, is our pressure sensor. We will eventually purchase a small breakout board in order to test this sensor. Meanwhile, we will utilize a breadboard and the Arduino to test the other sensors. Once we connect the sensors onto our breadboard and with the Arduino, we will write a series of simple code modules in order to test our sensors. Some of these tests can be seen in the table above.

Some of the sensors do not even require a breadboard for testing purposes, rather, they can be purchased on a breakout board. Aside of this, our group could further our soldering and PCB layout knowledge, and create our own simple breakout boards using simple Vector boards rather than complicated PCBs which involve traces and footprints. As previously mentioned, for simple and basic testing purposes, we will use an Arduino to test the sensors and make sure that the sensor itself properly works. We decided to use an Arduino due to the fact that it had fifty four digital I/O pins. However, we will not utilize the Arduino for further testing than this due to the fact that it varies too much from our PIC microcontroller.

|  |  |  |
| --- | --- | --- |
| **Test** | **Description** | **Expected Result** |
| Temperature Sensor Heat | Hold flame near the sensor | We should see a spike in the temperature output |
| Temperature Sensor Cold | Place sensor in freezer for 5 – 8 seconds | We should see a dramatic decrease in the temperature output |
| Humidity Sensor High | Place sensor in bathroom after taking a warm shower | We should see a high humidity output |
| Humidity Sensor Low | Place the sensor near a running dehumidifier | We should see a low humidity output |
| Range Finder Distance | Hold object 5 inches away from range finder | We should see a distance output of about 5 inches |
| Range Finder Distance | Hold object 10, 15, 20, 25, 30 inches away from range finder | We should see a distance output of about 10, 15, 20, 25, and 30 inches |

Table 22: Sensor Testing

**7.9 Microcontroller Testing**

Since the microcontroller is essentially the brain of our robot, we must ensure that it is properly working through various tests. In order to test, we will utilize various tools and equipment. We will be using the PICKit 2 debug express, MPLAB IDE, a development board, and a breadboard. When we conduct testing, we will load code onto our microcontroller connected to our development board via the PICKit 2 programmer which we purchased from Microchip. The PICKit 2 also features a debugging tutorial to debug our circuit from within the MPLAB IDE. The figure below displays the manner in which the testing code will flow.

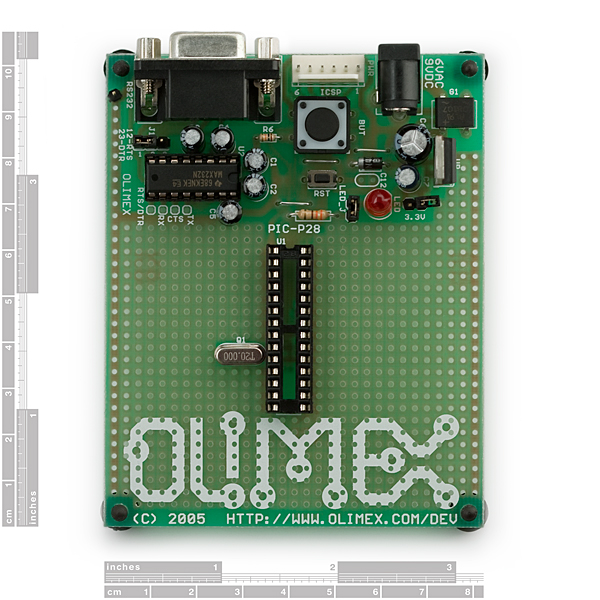


Figure 49: Code Testing Flow (Create by Group, Images Reprinted with Permission of Sparkfun)

The table which follows depicts some of the tests which we will execute on our microcontroller before we permanently have it soldered onto our PCB. This table is not fully explicit, and more tests will be conducted once we are further along in either the prototyping or construction phase.

|  |  |  |
| --- | --- | --- |
| **Test** | **Description** | **Expected Result** |
| Power On | Once we made our connections, power it on | Measured voltage and current should not exceed expected calculations |
| Simple Command | Send a command to power on a LED | The LED should light |
| RX and TX Connection | Is our base computer recognizing the transmitter attached to the MCU | We should see the receiver listed under a port on the base computer |
| Climate Control | Run the MCU in various environments and temperatures | Measured voltage and current should still be as expected |
| Temperature Control | Run the MCU with a constant current and voltage for 8-10 minutes | Voltage, current, and temperature should not exceed datasheet specs |

Table 23: Microcontroller Testing

**8.0 Bill of Materials and Responsibilities**

**8.1 Bill of Materials**

Due to the fact that our project is currently not being sponsored, the cost of the robot will be split among the three group members. However, as a result of our project lasting Spring and Fall versus Spring and Summer, we may have time to still find sponsorship. Since we will not begin building for a while, but rather, have plans to create prototypes on paper and schematics during the Summer, we still have the chance to present our project to organizations in hopes of sponsorship. Two of the group members are currently partaking in the College Work Experience Program at Lockheed Martin, and may seek sponsorship from them. Receiving sponsorship would only bring upon a positive impact on our project. Logic tells us that the more money we have, the better of quality components we will be able to purchase for our robot.

As previously mentioned, one of our main goals was to consolidate companies from which we would order parts from. Luckily, we were able to narrow down or component ordering to six online stores from which we would order and one store which we would physically visit to buy components: Sparkfun, Microchip Direct, Digi-Key, x10, Amazon, All-Battery, and RadioShack. The following table depicts the parts that we will acquire for our project, as well as individual and final cost.

|  |  | Unit Number | Unit Cost | Quantity | Source | Cost |
| --- | --- | --- | --- | --- | --- | --- |
| **Materials** |  |  |  |  |  |  |
| Microcontroller |  | PIC 16F886 | $5.99 | 2 | Microchip Direct | $11.98 |
| μC Dev. Board |  | Dev-00018 | $15.95 | 1 | Sparkfun | $15.95 |
| μC Programmer |  | PICKit 2 Debug Exp. | $49.99 | 1 | Microchip Direct | $49.99 |
| μC Transmitter |  | TXM-916-ES | $13.84 | 1 | Digi-Key | $13.84 |
| μC Receiver |  | RXM-916-ES | $17.12 | 1 | Digi-Key | $17.12 |
| Video Camera |  | AirSight IP (XX34A) | $99.99 | 1 | x10 | $99.99 |
| Camera Battery Back |  | XCam2 | $19.99 | 1 | x10 | $19.99 |
| Ultrasonic Range Finder |  | LV-EZ1 | $25.95 | 1 | Sparkfun | $25.95 |
| Pressure Sensor |  | MPL115A1 | $11.95 | 1 | Sparkfun | $11.95 |
| Temperature/Humidity Sensor |  | HTM1735LF | $25.95 | 1 | Digi-Key | $25.95 |
| Qik Dual Series Motor Controller |  | ROB-09106 | $64.95 | 1 | Sparkfun | $64.95 |
| XBee Wireless Modules |  | 2mW Series 2.5, 2.4 GHz | $21.95 | 2 | Sparkfun | $43.90 |
| XBee Board |  | Explorer Regulated | $9.95 | 1 | Sparkfun | $9.95 |
| XBee Board |  | Explorer USB | $29.95 | 1 | Sparkfun | $29.95 |
| PCB |  |  |  | 1 |  | $100 |
| Chassis |  | 1:14 R/C FF 6v Hummer | $29.99 | 1 | Target | $29.99 |
| LEDs |  | 10mm White Superbright (NTE-30071) | $3.51 | 3 | RadioShack | $10.53 |
| Phototransistor |  | LTR-4206E | $0.43 | 1 | RadioShack | $0.43 |
| Transistor (NPN) |  | 2N3904 | $0.99 | 2 | RadioShack | $1.98 |
| Crystal Oscillator |  | COM-00534 | $0.95 | 1 | Sparkfun | $0.95 |
| Power Supply |  | Tenergy 7.2V 2000mAh | $9.95 | 1 | All-Battery | $9.95 |
| Power Supply Charger |  | Smart Charger for 6-12V NiMH | $19.99 | 1 | All-Battery | $19.95 |
| Antenna |  | 916 MHz Planar | $2.08 | 2 | Digi-Key | $4.16 |
|  |  |  |  |  |  | **$619.40** |
| **Miscellaneous** |  |  |  |  |  |  |
| Batteries |  | AA |  | 8 | RadioShack | $4.99 |
| Battery Holder |  |  | $1.99 | 1 | RadioShack | $1.99 |
| Cable |  | USB to RS-232 Serial | $1.00 | 1 | Amazon | $1.00 |
| Cable |  | USB to microUSB | $1.00 | 1 | Amazon | $1.00 |
| Wires |  |  |  |  |  | $10.00 |
|  |  |  |  |  |  | **$18.98** |
|  |  |  |  |  |  |  |
| **Tools** |  |  |  |  |  |  |
| Soldering Iron |  |  |  | 1 | LAB | $0.00 |
| Solder (spool) |  |  |  | 1 | LAB | $0.00 |
| Wire Stripper/Cutter |  |  |  | 1 | LAB | $0.00 |
| Screwdriver |  |  |  | 1 | LAB | $0.00 |
| Workbench Tools (sander, drill press, etc.) |  |  |  | 1 | LAB | $0.00 |
| Hammer |  |  |  | 1 | LAB | $0.00 |
|  |  |  |  |  |  |  |
| **Total** |  |  |  |  |  | **$638.38** |

Table 24: Project Cost Breakdown

As we can see from the table above, our project will average about 600 dollars. From the estimated cost, each group member will contribute about 200 dollars into building our robot. As previously mentioned, we will still strive to seek sponsorship throughout the Summer months to help alleviate some of the costs. Aside of helping with costs, sponsorship would allow us for flexibility with parts such as buying more precise sensors, or building a better PCB.

In addition to the project cost breakdown table above, the following pie chart depicts the price breakdown for our project. It depicts the percentage distribution from the table above. Every part and component that will be integrated into our project is represented as a percentage in order to depict a distribution of the cost.

Figure 50: Cost Breakdown

**8.2 Responsibilities**

Because each group member has his own strengths and weaknesses, we will attempt to equally divide and distribute portions of the project to work on according to strengths. Overall, our goal is to have all the group members to contribute in all aspects of the project, whether it be coding the microcontroller, writing the GUI for the user, wiring a testing development board, or soldering the final PCB. The goal is that we attain as much knowledge as possible in the fields in which we might lack in. The following chart depicts an approximation of the amount of work and responsibilities which will be divided among the group.

Figure 51: Group Responsibility Chart (Created by Group)

**8.3 Project Milestones**

In order to properly and efficient complete this project in a timely and organized manner, our group must devise some sort of milestone chart. Creating such a chart will give act as a reference to keep track of all the tasks and responsibilities which we have to accomplish over the course of the upcoming months. We attempted to make this approximation as accurate as possible in order to avoid any unforeseen pitfalls or shortcomings. The milestones tables contains all aspects of the project including designing, building, testing, and writing of all the documentation.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Project Name** | **Days** | **Start** | **End** |
| **1.0** | **Senior Design Plan** | 210 | Jan. 9 | Dec. 3 |
|  |  |  |  |  |
| **2.0** | **Writing phase** |  |  |  |
| **2.1** | Initial Document | 8 | Feb. 7 | Feb. 14 |
| **2.2** | Half Final Document | 28 | Feb. 14 | Mar. 12 |
| **2.3** | Final EEL 4914 Doc | 30 | Mar. 12 | Apr. 20 |
| **2.4** | Final Document | 105 | Aug. 20 | Dec. 3 |
| **2.5** | Final Presentation | 105 | Aug. 20 | Dec. 3 |
| **3.0** | **Research Phase** |  |  |  |
| **3.1** | Hardware | 76 | Feb. 7 | Apr. 23 |
| **3.2** | Software | 76 | Feb. 7 | Apr. 23 |
| **4.0** | **Design Phase** |  |  |  |
| **4.1** | Sensors/Electronics | 69 | Feb. 14 | Apr. 23 |
| **4.2** | PCB Layout | 91 | Apr.23 | Jul. 23 |
| **4.3** | Software(Comm.) | 69 | Feb. 14 | Apr. 23 |
| **4.4** | Ordering/Receiving Parts | 91 | Apr. 23 | Jul 23 |
|  |  |  |  |  |
| **5.0** | **Assembly Phase** |  |  |  |
| **5.1** | Electronics Mounting/Soldering | 25 | Aug. 20 | Sep. 14 |
| **5.2** | Electrical Connections | 25 | Aug. 20 | Sep. 14 |
| **5.3** | Server/Software | 50 | Aug. 20 | Oct. 9 |
| **5.4** | GUI | 20 | Sep. 20 | Oct. 10 |
|  |  |  |  |  |
| **6.0** | **Testing Phase** |  |  |  |
| **6.1** | Microcontroller | 52 | Oct. 9 | Nov. 30 |
| **6.2** | Camera | 52 | Oct. 9 | Nov. 30 |
| **6.3** | Sensors | 52 | Oct. 9 | Nov. 30 |
| **6.4** | Navigation | 52 | Oct. 9 | Nov. 30 |
| **6.5** | Software | 52 | Oct. 9 | Nov. 30 |
| **6.6** | Whole Unit | 52 | Oct. 9 | Nov. 30 |

Table 25: Project Milestones

**Appendix A: Bibliography**

“An Introduction to Infrared Technology:Applications in the Home, Classroom, Workplace, and Beyond ...”

<http://trace.wisc.edu/docs/ir_intro/ir_intro.htm>

“IR Vs. RF: Which is the best for my needs?”

<http://www.aclasstechnology.com/downloads/IRvsRF.pdf>

# Which is the best frequency to use for my application: 315MHz, 418MHz, 433.92MHz, or 900MHz?

<http://www.linxtechnologies.com/support/knowledgebase/rf-modules/which-is-the-best-frequency-for-my-application/>

“Understanding Gear Reduction”

<http://www.teamdavinci.com/understanding_gear_reduction.htm>

“XBee Wireless Communication Starter Kit”

<http://www.trossenrobotics.com/p/Xbee-Communication-Starter-Kit.aspx?feed=Froogle>

“Choosing a Motor: DC, Stepper, or Servo – Free How-to Robot Construction Article”

<http://www.robotoid.com/howto/choosing-a-motor-type.html>

“DC Motor: High Torque Mini DC Gear Motor 3-12V, 5-25 rpm for Hobby / Robots”

[http://www.batteryspace.com/dcmotorhightorqueminidcgearmotor3- 12v1300rpmforhobbyrobots.aspx?gclid=CI39kerToa8CFQrCtgodp3JlaA](http://www.batteryspace.com/dcmotorhightorqueminidcgearmotor3-12v1300rpmforhobbyrobots.aspx?gclid=CI39kerToa8CFQrCtgodp3JlaA)

“Stepper Motor Control – AVR Tutorial”

<http://extremeelectronics.co.in/avr-tutorials/stepper-motor-control-avr-tutorial/>

“How to Build a Robot Tutorial”

<http://www.societyofrobots.com/batteries.shtml>

“Alkaline Battery: Advantages and Disadvantages”

<http://www.doityourself.com/stry/alkaline-battery-advantages-and-disadvantages>

“Lithium Battery”

<http://en.wikipedia.org/wiki/Lithium_battery>

“Lead Acid Batteries”

<http://www.mpoweruk.com/leadacid.htm>

“Nickel-Cadmium Battery”

[http://en.wikipedia.org/wiki/Nickel%E2%80%93cadmium\_battery#Comparison\_w](http://en.wikipedia.org/wiki/Nickel%E2%80%93cadmium_battery#Comparison_w_other_batteries)

[\_other\_batteries](http://en.wikipedia.org/wiki/Nickel%E2%80%93cadmium_battery#Comparison_w_other_batteries)

“NiCd Battery Pack: 7.2V 700mAH for RC Cars: Nikko / Radioshack etc”

[www.batteryspace.com/nicdbatterypackone72v700mahforrccarsnikkoradioshacketc.aspx](http://www.batteryspace.com/nicdbatterypackone72v700mahforrccarsnikkoradioshacketc.aspx)

“Nickel Metal Hydride NiMH Batteries”

<http://www.mpoweruk.com/nimh.htm>

ANT-916-CHP-T Datasheet

<http://www.antennafactor.com/resources/data-guides/ant-xxx-chp-x.pdf>

“VSWR”

<http://www.antenna-theory.com/definitions/vswr.php>

Energizer L91 Datasheet

<http://data.energizer.com/PDFs/l91.pdf>

## “RF Link Transmitter - 434MHz”

## <http://www.sparkfun.com/products/10534>

Stevens Institute of Technology Search and Rescue Robot Senior Design Project

<http://tiger.ece.stevens-tech.edu/08-09/grp16/Other%20Documents/EE423_Group16_FallFinalReport1.pdf>

“Robot to the rescue: University of Warwick students gear up for European championship”

<http://www.sciencebusiness.net/news/75691/Robot-to-the-rescue-University-of-Warwick-students-gear-up-for-European-championship>

“Microsoft Kinekt-powered robot to aid earthquake rescue-”

<http://www.bbc.com/news/technology-12559231>

“How WiFi Works”

<http://computer.howstuffworks.com/wireless-network1.htm>

“Bluetooth Basics”

<http://www.baracoda.com/baracoda/technology/why-bluetooth.html>

“How Bluetooth Works”

<http://electronics.howstuffworks.com/bluetooth.htm>

“A beginner’s guide to switching regulators”

<http://www.dimensionengineering.com/info/switching-regulators>

“Switching Regulators”

<http://www.national.com/assets/en/appnotes/f5.pdf>

LM2936 Datasheet

<http://www.ti.com/lit/ds/symlink/lm2936.pdf>

LT1121 Datasheet

<http://cds.linear.com/docs/Datasheet/1121ff.pdf>

MAX-232 Datasheet

<http://datasheets.maxim-ic.com/en/ds/MAX220-MAX249.pdf>

“Control a Car Via PC Using XBee”

<http://sites.google.com/site/electronicprojectsprogramming/control-an-rc-car-via-pc-using-xbee>

“RS-232 Cables, Wiring, and Pinouts”

<http://www.zytrax.com/tech/layer_1/cables/tech_rs232.htm>

“TTL to RS232 Converter”

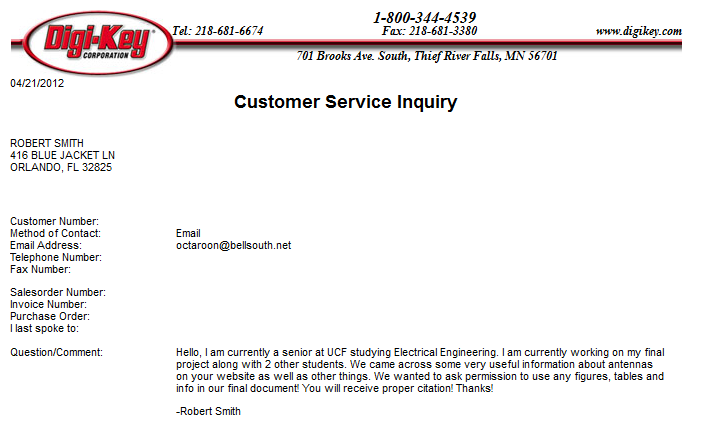
<http://www.coolcircuit.com/gadgets/2008/07/15/ttl-to-rs232-converter/>

PIC16F886 Datasheet

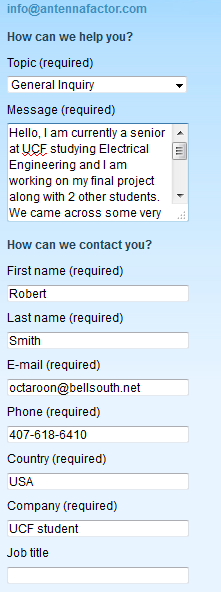
[ww1.microchip.com/downloads/en/DeviceDoc/41291F.pdf](file:///C:\Users\TJ\Dropbox\Senior%20Design\ww1.microchip.com\downloads\en\DeviceDoc\41291F.pdf)

**Appendix B: Permissions**

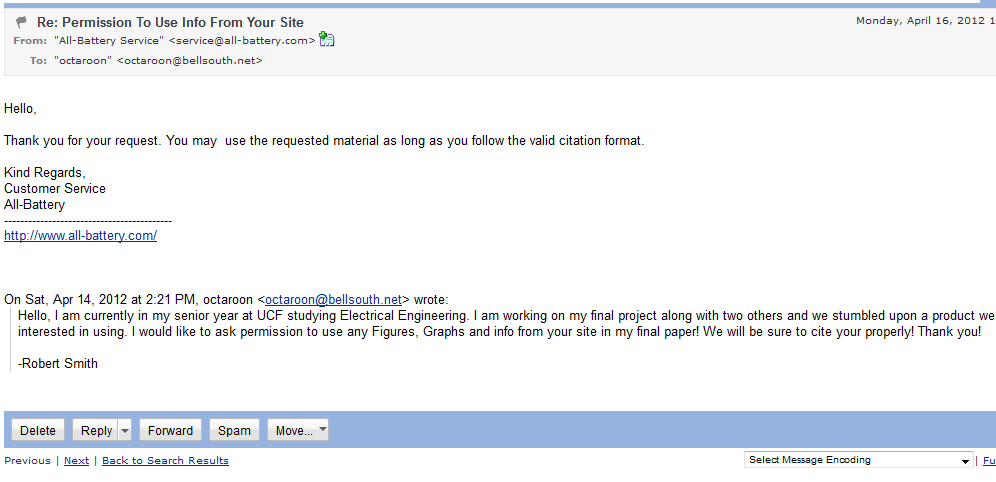
**Digi-Key (PENDING)**



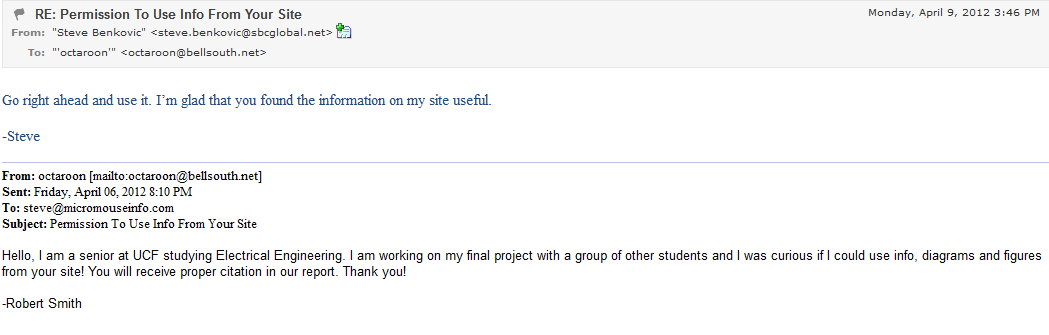
**Antenna Factor (PENDING)**



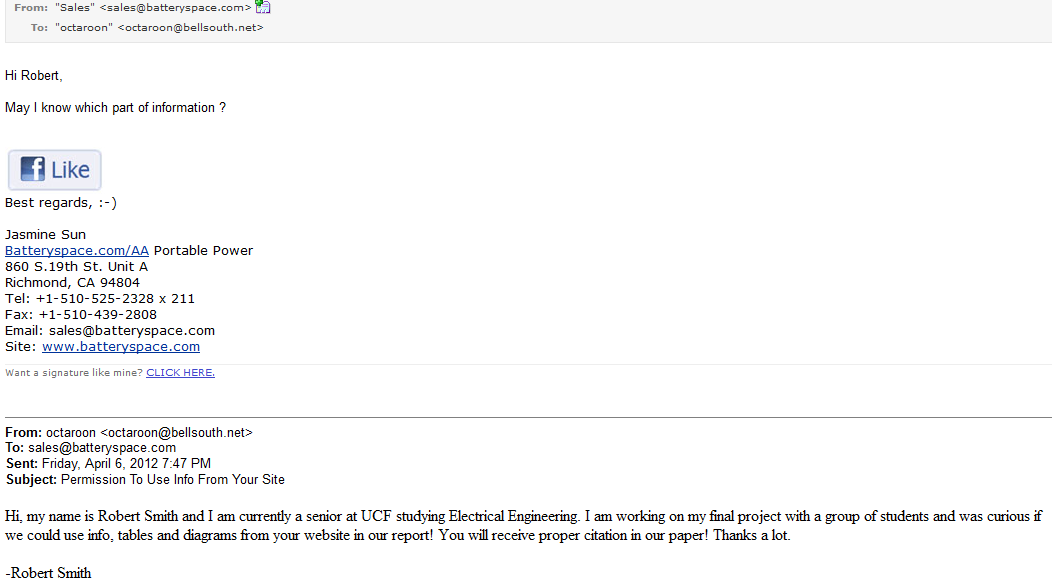
All-Battery Service



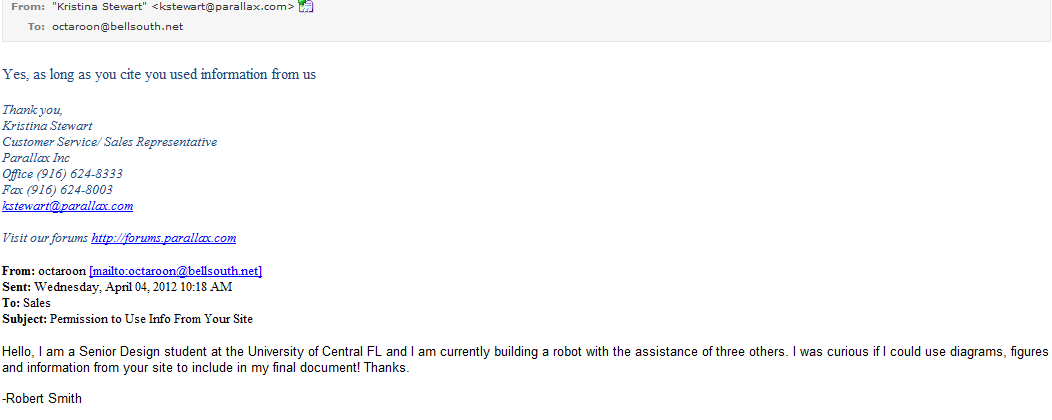
Micro Mouse



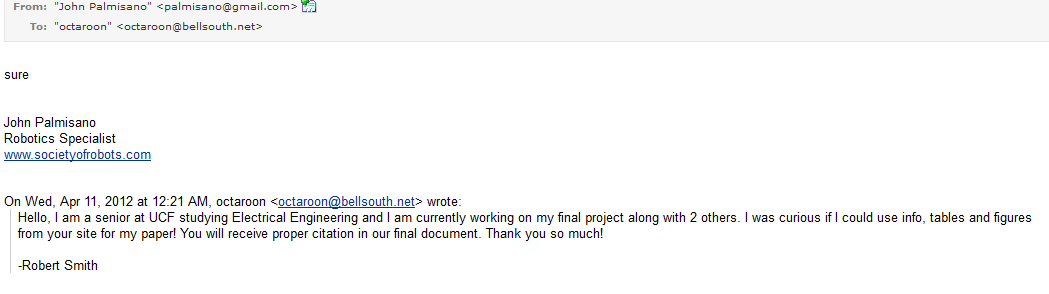
Battery Space (PENDING)



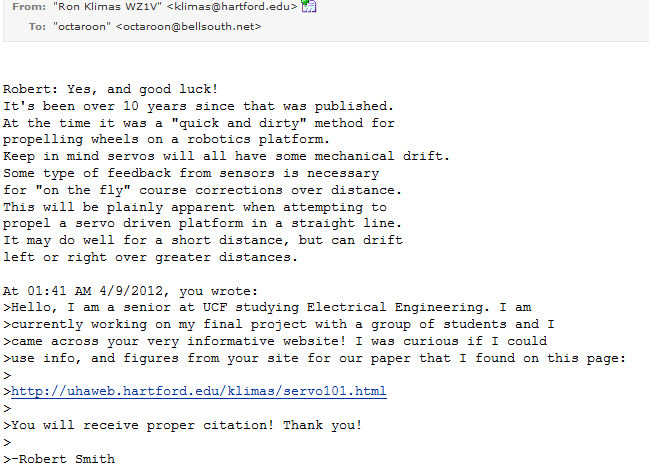
Parallax



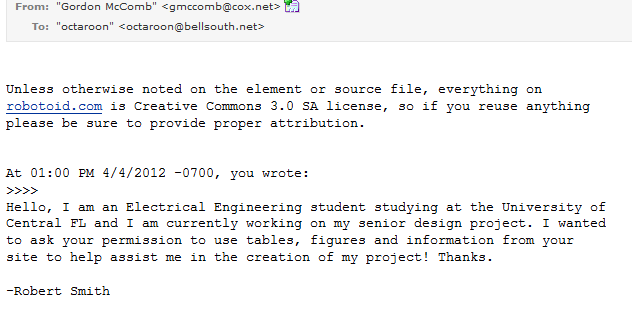
Society of Robots



University of Hartford



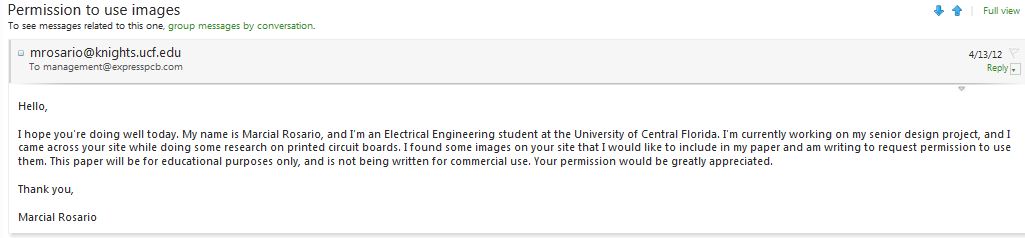
Robotoid



Trossen Robotics

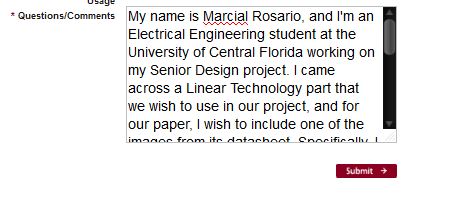


ExpressPCB

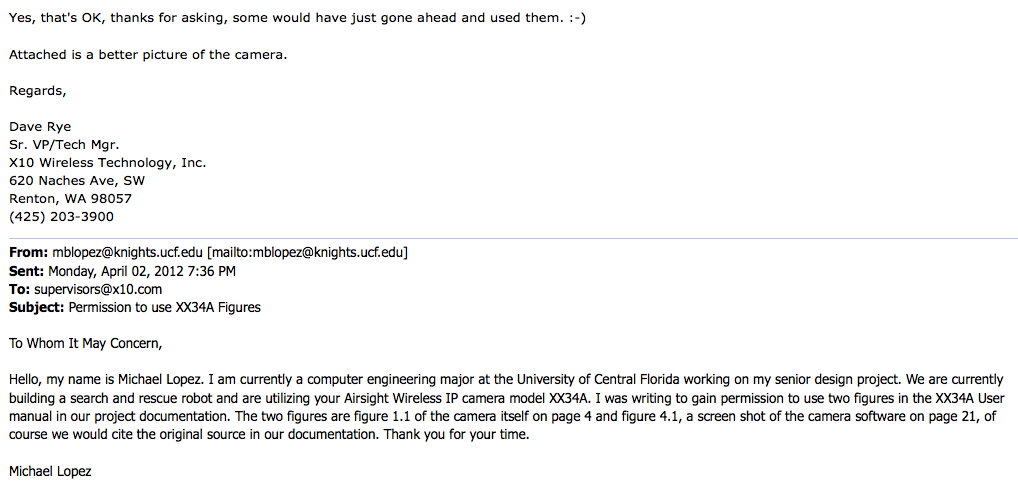




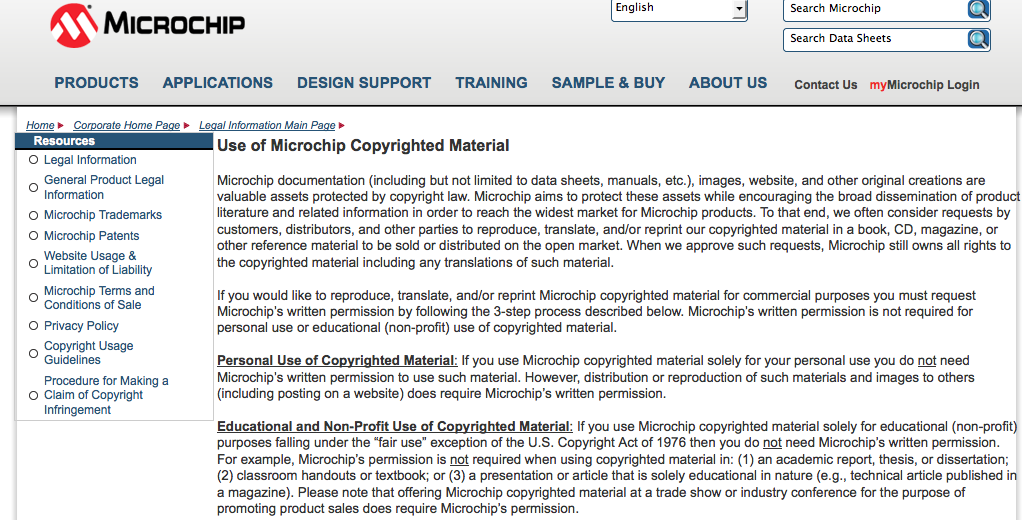
Linear Technology (Pending Permission)



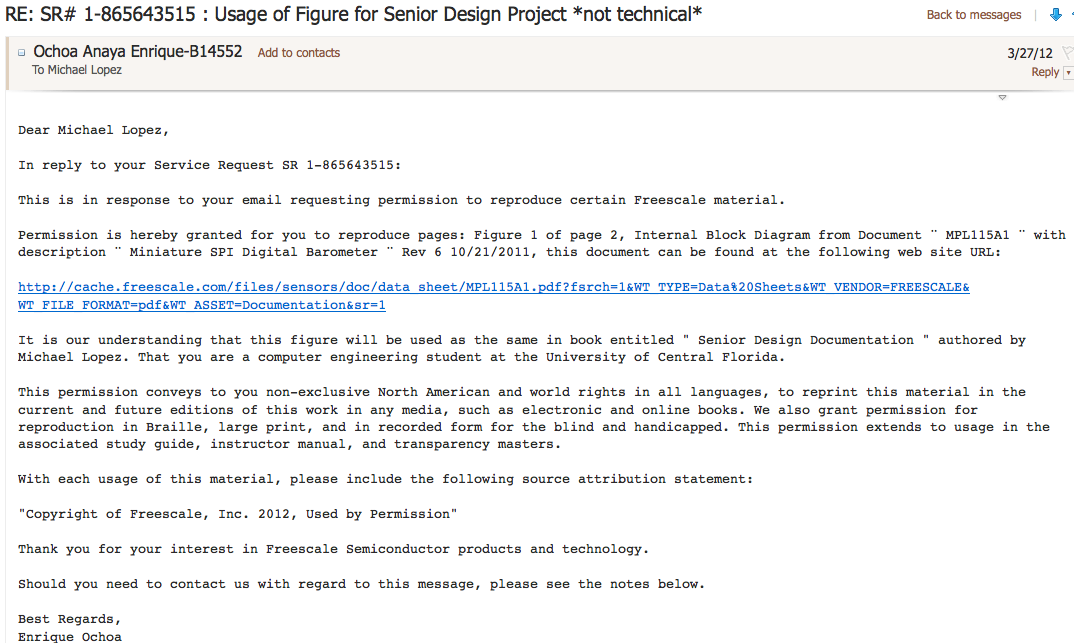
X10



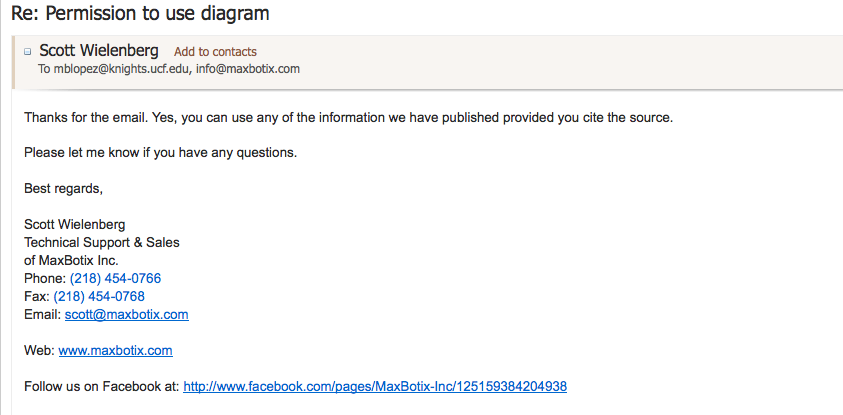
MicroChip



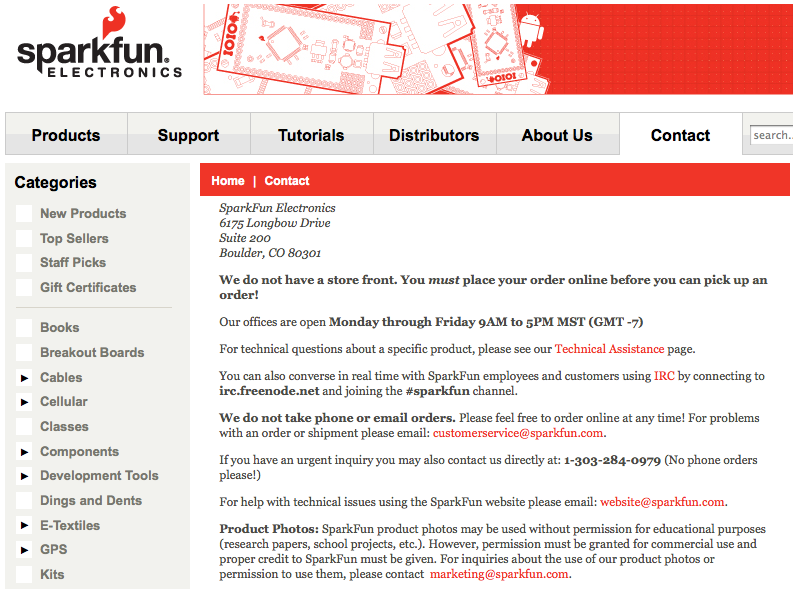
Freescale Semiconductors



Maxbotix



Sparkfun



Additional Pending Sources:

Target

Energizer

**Appendix C: Index of Figures**

Figure 1 – MQ-9 Reaper………...……………………………………………………….10

Figure 2 – uC Block Diagram……………………………………………………………21

Figure 3 – Microcontroller Schematic…………………………………………………...24

Figure 4 – MPLAB IDE……………………………………………………….………....26

Figure 5 – Microcontroller Flow Chart………………………………………......……....27

Figure 6 – PICKit Connection to MCU……………………………………………….....28

Figure 7 – PIC Development Board……………………………………………………...29

Figure 8 – MPL115A1 Connection to MCU…………………………………………….30

Figure 9 – Pressure Sensor Internal Block Diagram……………………………………..31

Figure 10 – Voltage vs. Humidity Graph of Temp/Humidity Sensor……………………32

Figure 11 – Temp/Humidity Sensor Connection to MCU……………………………….33

Figure 12 – Range Finder Beam Pattern…………………………………………………34

Figure 13 – Range Finder Connections to MCU………………………………………...35

Figure 14 – Dark-Detecting LED Circuit………………………………………………..36

Figure 15 – Airsight IP Camera………………………………………………………….38

Figure 16 – Camera Software……………………………………………………………39

Figure 17 – Sensor Data Flow…………………………………………………...………40

Figure 18 – RF RX to MAX232 Wiring…………………………………………………41

Figure 19 – Software Flowchart…………………………………………………………41

Figure 20 – Log Data…………………………………………………………………….42

Figure 21 – Display File GUI……………………………………………………………43

Figure 22 – Wireless Control…………………………………………………………….44

Figure 23 – Robot Control Flow Chart…………………………………………………..45

Figure 24 – Buck Regulator……………………………………………………………...47

Figure 25 – LM7805 Regulator Connected to Components…………………….……….49

Figure 26 – Example of LM2936 Connected to Components……………………...……50

Figure 27 – Sample Implementation of LTC3632……………………………………….51

Figure 28 – Boebot Chassis Dimensions (Bottom View)………………………………..52

Figure 29 – Typical H-bridge Using Transistors as Switches…………………………...54

Figure 30 – MT-WSM500 DC Motor Dimensions………………………………………56

Figure 31 – SpringRC 43R Servo Motor Dimensions…………………………………...58

Figure 32 – Remote Control Car Dimensions……………………………………………60

Figure 33 – Module Showing How Our Robot Will Receive Instructions………………62

Figure 34 – Wenshing Hi Power Transmitting Module……………………....………….63

Figure 35 – Xbee Exploer USB………………………………………………………….65

Figure 36 – Xbee Explorer Regulated…………………………………………………...66

Figure 37 – Xbee 1 mW Wire Antenna Transceiver…………………………………….67

Figure 38 – Alkaline and Lithium AA Battery Performance…………………….………68

Figure 39 – 7.2 V, 700 mAh NiCd Battery………………………………………………71

Figure 40 – Tenergy 7.2 V 2000 mAh Side-by-Side NiMH Battery Pack………………72

Figure 41 – Compact Smart Charger (0.6 A) for NiCd or NiMH Batteries……………..73

Figure 42 – 916 MHz Ceramic Chip Antenna by Antenna Factor………………………74

Figure 43 – 916 MHz Planar Surface Mountable Antenn………………………….……75

Figure 44 – 916 MHz ¼ Wave Whip Antenna……………………………....…………..76

Figure 45 – Polulu Low Voltage Dual-Serial Motor Controller…………………………78

Figure 46 – Order of Operation For Wireless Control…………………………….....…..79

Figure 47 – PCB Layout Software……………………………………………………….82

Figure 48 – Screen Capture of PCB123 Software……………………………………….84

Figure 49 – Code Testing Flow………………………………………………………….91

Figure 50 – Cost Breakdown………………………....………………………………….95

Figure 51 – Group Responsibility Chart…....……………………………………………96

**Appendix D: Index of Tables**

Table 1 – Table of Requirements……………………………………….............................5

Table 2 – Pros and Cons of 802.11………………………………………………………13

Table 3 – Possible PIC MCUs…………………………………………………………...23

Table 4 – Microcontroller Sensor Output………………………………………………..25

Table 5 – Humidity to Voltage…………………………………………………………..32

Table 6 – RS232 Outputs………………………………………………………………...40

Table 7 – Robot Control…………………………………………………………………44

Table 8 – Parts to Keep in Regulation…………………………………………………...46

Table 9 – Various Motor Pros and Cons…………………………………………………53

Table 10 – DRV8835 Dual Low Voltage H-bridge Spec Table…………………………54

Table 11 – Continuous DC Motor Specifications…………………………………….….56

Table 12 – Pin Layout of Wenshing Transmitter………………………………….……..63

Table 13 – Hightlighted Specs of Wenshing Transmitter………………………….…….63

Table 14 – Motor Controller Pin Layout…………………………………………...........78

Table 15 – Chassis Testing………………………………………………………………86

Table 16 – Speed Testing………………………………………………………………...87

Table 17 – Battery Testing……………………………………………………………….87

Table 18 – Battery Charger Testing……………………………………………………...88

Table 19 – Video Camera Testing………………........………………………………….89

Table 20 – PCB Testing………………………………………………………………….89

Table 21 – Wireless Control Testing…………………………………………………….90

Table 22 – Sensor Testing…………………….………………………………………….91

Table 23 – Microcontroller Testing……………………………………………………...92

Table 24 – Project Cost Breakdown……………………………………………………..93

Table 25 – Project Milestones……………………………………………………………96