Spring 2017

Packmule

Jared M. Alexander
jma126@zips.uakron.edu

Jared J. Ford
jjf51@zips.uakron.edu

Timothy J. Griffiths
tjg40@zips.uakron.edu

Andray Pennington
ap116@zips.uakron.edu

Packmule

Jared Alexander
Jared Ford
Tim Griffiths

Department of Electrical and Computer Engineering

Honors Research Project

Submitted to

The Honors College

Approved:

[illegible] Date 5/2/2017
Honors Project Sponsor (signed)

Yilmaz Sozer
Honors Project Sponsor (printed)

[illegible] Date 5/3/2017
Reader (signed)

Gregory A. Lewis
Reader (printed)

[illegible] Date 3 May 2017
Reader (signed)

ROBERT J. VEILLETTE
Reader (printed)

Accepted:

Joan Carletta Date 5/4/17
Department Head (signed)

Joan Carletta
Department Head (printed)

[illegible] Date 3 May 2017
Honors Faculty Advisor (signed)

ROBERT J. VEILLETTE
Honors Faculty Advisor (printed)

______ Date ______
Dean, Honors College

Jared Alexander

The project our team chose to design and build for our senior project is the Packmule. It is a load-bearing, following robot with the capabilities of moving autonomously or manually. An electric wheelchair was dismanlted and the base and motors were implemented into our new design along with a motor controller and microprocesser that are used to drive the motors. Manual drive of the Packmule is done using an app that was created by our team while autonomous movement is achieved through the use of ultrasonic signals. The implementation of the ultrasonic sensors was my greatest contribution to the project.

As I stated, the Packmule has the capability to track and follow its user using ultrasonic sensing. To achieve this, an ultrasonic transmitter must be worn on the belt of the user while ultrasonic sensors located around the base of the Packmule receive the transmitted signal. The strenghth of this received signal at each sensor is compared and the direction of movement is determined by a path-planning algorithm to keep the Packmule moving towards the user. For my contribution, I designed a transmitter circuit that produces an ultrasonic pulse with a center frequency of 40kHz that is strong enough to communicate with the Packmule over a distance of about 5 meters. Throuoghout the course of this project, several different methods were implemented to produce the desired signal. One way was to use an Arduino microcontoller with an ultrasonic transducer. This signal is strong with a 5V peak-to-peak (V_{p-p}) amplitude and is suitable to communicate over the desired distance. I also designed a timer circuit that implements an integrated circuit, the 555 Timer, with calculated resistance and capacitance values to produce a 40kHz pulse with a $5V_{p-p}$ amplitude able to transmit the desired distance. This signal is then projected into the air by an ultrasonic transducer. This circuit requires a supply voltage of 5V, so a linear voltage regulator circuit was designed to step down 9V from the battery down to 5V. The voltage regulator allows for a clean and precise supply voltage. To complete this task, the circuit was soldered and secured in a wearable box along with (2) 9V batteries connected in parallel, and power was separated from the circuit with a switch.

Another task I was responsible for was designing the ultrasonic reciever circuits. This circuit consists of an ultrasonic transducer that sends its received signal to a two stage BJT amplifier that is needed to amplify the attenuated ultrasonic signal to a value that is easily readable by the Arduino microcontroller. A potentiometer is also used so that each sensor can be tuned to produce equal readings. The last stage of the receiving circuit is an envelope detector that converts the signal from an oscillatory pulse to a straight DC voltage. This makes it easier to compare the received signal at each sensor. Each circuit requires a supply voltage of 5V that is taken from the 5V output of the Arduino. There are a total of 5 receiving sensors to allow for a detection area of about 180 degrees around the front of the Packmule. Each sensor was soldered and secured in project boxes before being mounted.

Other tasks that I performed along with my teammates include sensor testing and mechanical construction of the Packmule’s load-bearing bin and electrical equipment level. Sensor testing involved reading received values at different distances and angles to determine the optimal mounting locations of the receivers and acquiring data for the software lead to use in production of the algorithms and code. Mechanical construction involved mounting the electrical equipment on a wooden level supported by steel that had to be attached to the frame. The load-bearing bin was secured so that it was separate and would not interfere with the electrical equipment.

Overall, this was a satisfying and successful project. I believe my contributions were vital to its success, and I was also fortunate to have the opportunity to work with a smart and hardworking team. Taking part in the design and construction of the Packmule has strengthened my skills as an engineer, and it has allowed me to expand my abilities to work in a team setting.

Jared Ford
For my team's senior design project, Packmule, I was primarily responsible for the software design. The project, which consists of multiple ultrasonic sensors, IR proximity detectors as well as motor and a drive system needed a way to make sense of all this data and translate that into movement. This goal was completed through the use of multiple software routines that I wrote.

First, I focused my efforts on creating a way to control the system using a phone with Bluetooth. Once I was able to establish a concrete connection with a Bluetooth module we purchased, I created the mobile app with a pseudo- analog joystick interface. The app is able to process touches on the phone and translate that data into commands the motor drive is able to parse. I created the app first for Android, then I rewrote the entire application for the iPhone, both of which I made available on their respective app stores to allow for better demonstrations on presentation day. After completing the app, one could simply move the Packmule with their phone.

Next, I worked on getting reliable system movement based on detected user location data from the ultrasonic transmitter/ receivers. In order to successfully follow the user, one needs to know where the user is. This, in turn, required the system to have multiple ultrasonic receivers to be able to detect the user successfully. Each sensor would report how close the user was to it by transmitting voltage based on relative distance to the microcontroller. I then took this data and compared it to all the other data reported by the other sensors in order to make an educated decision as to what direction the Packmule should move and how fast to make it move. This algorithm was one of the more complex ones that I had to write as many of the sensors did not provide consistent results all the time. In order to overcome this difficulty, I wrote a secondary algorithm that was able to detect false readings and eliminate them from the end calculation. After completing this function, the system was able to follow a user.

Once the manual control modes and following modes were finished, I wrote another algorithm to process the IR sensor readings and avoid obstacles if they were in the way. To avoid the obstacles, based on data reported by the sensors, I created a function to pause the following routine and move around the obstacle. Avoiding the obstacles was not as simple as just backing up the device, but required the system to make real-time decisions based on the direction the user was travelling in. For instance, if the user was walking around a tree, we want the robot to not only avoid the obstacle, but to also take the most efficient path possible to save on overall power consumption and time. Depending on where the obstacle was detected, I had to write several different routines to ensure there was never a case when the robot would run into an object. Additionally, I aided with the wiring up the system and making sure everything was structurally sound for the purposes of this project.

Tim Griffiths

The purpose of the Packmule is to function as an autonomous load bearing robot, capable of following a user in either an automatic or manual mode.

My first major contribution to this project was the understanding of the existing electrical system. This involved major disassembling of the existing wheelchair unit in an effort to understand what the current system used, and what exactly our design team would need to implement or update. With an understanding of the current functionality of the robot, we were able to utilize some existing electrical components and connections in order to bring down the overall cost of our system.

Next, I was in charge of the power and drives system of the robot. This two part position first required me to do some motor testing to determine the capabilities and energy requirements our dual 24V, DC motors would require. Once these calculations were complete, I was able to determine the amount of energy the system would require by inferring the expected operation modes of the robot. From here we were able to order the correct power supplies.

Once the ultrasonic and infrared sensor circuits were assembled, it was my job to determine the proper orientation necessary to deliver upon stated engineering requirements. Using technical information gathered from data sheets and trigonometry, I was able to determine appropriate positions and mounting angles of the sensors which would provide the expected results. This was necessary in order to determine the “dead zones” in which the Packmule would not identify the user’s location as well as eliminate “overlap zones” where user’s information would be determined by multiple sensors. In addition to determining the position of the sensors, I was in charge of scheduling to the position of equipment on the sensor level to make sure there was enough room for all our equipment and nothing overlapped.

In addition to my individual contributions, I participated in team-wide tasks that required the cooperation and communication of all members to accomplish. These tasks included the testing of sensors to ensure they were performing as expected when received from the manufacturer, calibration of these sensors with our system layout, and finally the construction of the load bearing bucket.

My specific role in this project was the team leader. In addition to my engineering responsibilities, I also had several organization, communication, and time management tasks in order to keep the project up-to-date and on schedule. Some of these additional responsibilities included acting as the primary point of contact between our design team and faculty members, keeping a running budget of the cost of the project, submitting parts request forms, and implementing and updating Gantt charts in order to forecast our group’s completion of the project.

Overall, I am grateful for the opportunity to work with capable, driven honors students. Each individual’s contributions were pivotal to the success and execution of this project. This experience has helped further prepare me for a career in industry.

Packmule

Project Design Report

Design Team 02

Jared Alexander

Jared Ford

Tim Griffiths

Andray Pennington

Faculty Advisor: Dr. Yilmaz Sozer

Date Submitted: 1 May 2017

Table of Contents

List of Figures

List of Tables

Introduction

Abstract

People face demands of hauling equipment and belongings with them every day, whether it be for work or leisure. This design report discusses and details a product that would allow people to overcome the struggles of this. The Packmule is an autonomous following robot that has the capability of carrying a load up to 30 pounds. The design involves two independently controlled motors operating two drive wheels so that the Packmule will be flexible in the directions it can move. There are also two more steering wheels for support of the base and the load inside. The way in which the Packmule follows the user consists of a wireless transmitter that will be worn on the user's belt. This transmitter will emit an ultrasound signal that will be received by five ultrasonic position sensor receivers mounted on the Packmule. The signals will be filtered and amplified in order to communicate with microcontroller. The strength or amplitude from each received signal will be compared in order to determine the location of the user. Since the Packmule is autonomous, it will need to be able to detect objects in its path. Several infrared sensors will be mounted to the front of the Packmule for this reason. It will track the distance an object is away from the Packmule by measuring the amount of light . Algorithms programmed into the Arduino Mega 2560 microcontroller will take the data from the IR sensors and the ultrasonic position sensor receivers and calculate a path to safely and efficiently follow the user.

Problem Statement

Need

Every day, people struggle to tow along their weighted belongings. Whether one is taking golf clubs around the course or a toolbox to a worksite, it is exhausting and distracting to move these loads singlehandedly. The Packmule would be a solution to lighten the strain of those carrying moderate loads from place to place. The convenience of an autonomous, self-navigating robot would lighten the load of the leader allowing for less exhausting and distraction free transportation of heavy items.

Objective

The objective of the Packmule is to provide a convenient way to transport a load across an area. The design will utilize a location tracking system comprised of a ultrasonic transmitter and receivers, a wireless application for switching between automatic-follow mode and manual mode, a proximity sensor for avoiding obstacles, 2 DC motors each driving an independent wheel, and a controller. The controller will take the information from the tracking system, wireless application, and proximity sensor to control the movement of the device. Additionally, the system will have two modes: follow mode and manual mode. In follow mode, the Packmule will track and follow the user across a path while maintaining a distance of about 5 feet. When the system is in follow mode, an IR sensor will check for obstacles in front of it and allow the system to find a route to the user with no obstructions. If obstructed, the device will self-navigate around an obstacle and resume following the user. In manual mode, the user will control the Packmule's movement via a wireless application with directional control. The user will be able to park the system to prevent it from traversing the water or entering hazards. The end result will be an easy and convenient way to move one's luggage from one point to another.

Background

Autonomous robots are being used in many different fields ranging from space exploration to personal hobby projects, as well as being used to complete many different tasks in these fields. An autonomous robot will normally consist of sensors, motors, microcontrollers, and algorithms. Comparing the mentioned components to the human anatomy would go as follows:

- Sensors - Eyes and Ears
- Motors - Arms and Legs
- Microcontroller - The Body Connecting the Components
- Algorithms - The Brain Controlling all the Parts

The ability for a robot to track a user while carrying a load has been implemented in various ways, but hasn't been seen commercially performing on rougher terrain types.

Position Detection Methods

Time of Arrival

The time of arrival method uses the precise measurement of the time it takes for a transmitted signal to be received. By calculating the travel time (t) of the signal, and then multiplying it by the velocity (v) at which the signal propagates, the distance (d) between the transmitter and receiver can be found.

$$d\ [m] = v\ [m/s] * t[s]$$

Equation 1: Time of Arrival Distance [4]

In order for this method to be accurate, a minimum of three receivers must be used to pin point location using trilateration. The distance calculated will then be used as a radius around each receiver. The point at which the three circles interest is the location of the transmitter. The main drawback of this method is the need for precise time synchronization at all transmitting and receiving stations of the system. Small discrepancies in time can cause very large errors. This method shows great challenges in systems where noise and interference will exist [4].

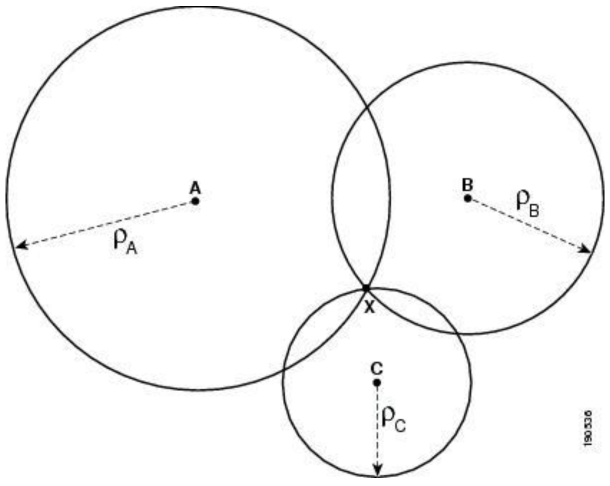


Figure 1: Time of Arrival Method [4]

Time Difference of Arrival

The method of time difference of arrival is another technique that requires a minimum of three receivers. However, it is unnecessary for precise synchronization of time between the transmitter and receivers because it uses relative time instead of absolute time for its calculations. With this method, only the receivers need time synchronization. Hyperbolic lateration is used to pinpoint the location of the transmitter based on the difference of arrival times found by at least three receivers [4].

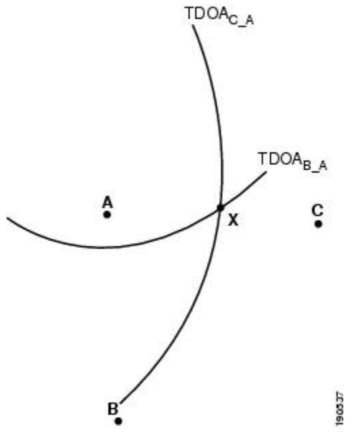


Figure 2: Time Difference of Arrival Method [4]

Amplitude of Signal

Another method of determining position of the transmitter is to measure the amplitude of the transmitted signal that is received by different ultrasonic receivers. Ultrasonic transducers will excite a voltage when receiving an ultrasonic pulse that can be manipulated into a signal that is easily readable by a microcontroller. These signals from several at least three receivers can be compared to triangulate the position of the transmitter by determining which receivers are nearest to the transmitter based on the strength of those signals. The more receivers that are used the more accurate the determined location will be. This method is beneficial in that time synchronization is required.

Obstacle Avoidance Methods

Infrared

Infrared sensors determine distance by emitting a pulse of light and waiting for the reflection. The infrared emitter consists of light emitting diodes, while the receiver is a photodetector containing a p–n junction that converts light photons into current. If the object is close, the reflected light will be stronger than if the object is further away.

Collision

Using bumper switches/sensors, Figure 3, can allow the Packmule to adjust its position after it has encountered an object in its path. It does this by maintaining a HIGH signal on its sensor port, which means the bumper switch is turned off when not pressed. While the bumper switch is pressed, the signal changes to LOW, which tells the robot that the switch has been triggered on. The bumper switch remains turned on until the button is longer being pressed.

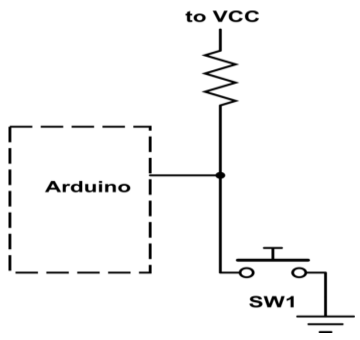


Figure 3: Bumper Switch

Active Sonar

Active sonar creates a pulse of sound, often called a "ping", and then listens for reflections (echo) of the pulse (See Figure 4). The sound waves are emitted and received through transducers. The Packmule will use a monostatic configuration in which the emitting transducer is located next to the receiving transducer. The frequencies of sonars can range from infrasonic to ultrasonic. Acoustic frequencies above the upper audible limit of human hearing (20 kHz) can be considered ultrasound, while infrasound is acoustic frequencies below 20 kHz. The Packmule will operate at an ultrasonic frequency of 40 kHz. To measure the distance to an object, the time from transmission of a pulse to reception is measured and converted into a range by knowing the speed of sound. The method of *Active Sonar* will be used for the Packmule due to its outside performance as well as its ability to detect objects before collision.

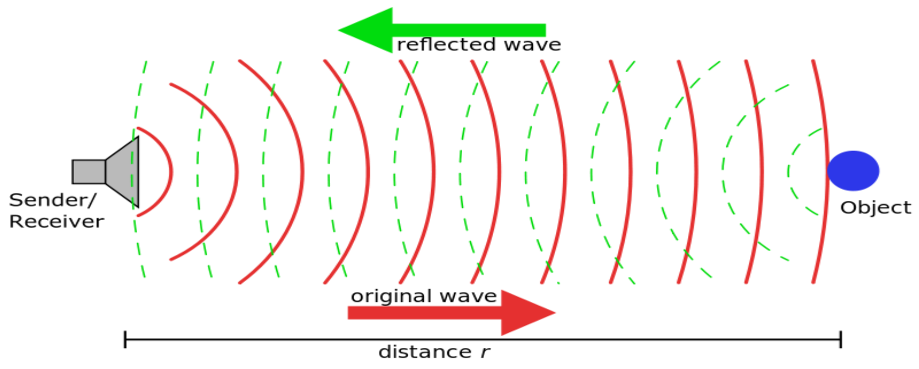


Figure 4: Active Sonar

Single-Board Microcontroller

Microcontroller systems provide multiple forms of input and output signals to allow application software to control an external "real-world" system. The Packmule will need several sensors, motor drives, and a wireless communication system in order to avoid obstacles while following the user. Table 1 shows several microcontroller board options as well as specifications that relates to the Packmule. The Arduino Uno is very cost effective, but doesn’t have and Raspberry Pi 3

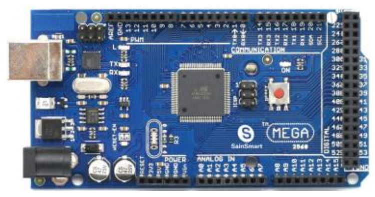
Image			
Make/Model	Arduino Mega 2560	Arduino Uno	Raspberry PI 3
List Price	$45.95	$24.95	$29.99
Analog Pins	16	6	-
Digital Pins	54 (15 PWM)	6 (No PWM)	40
DC Current per I/O Pin	40mA	20mA	16mA
On-Board BT	No	No	Yes

Table 1: Microcontroller Comparison Table

Communication Methods

Bluetooth

Bluetooth is a wireless communication standard which allows electronic devices to connect and interact with each other. Data is transmitted via low-power radio waves at a frequency of 2.45 gigahertz. The power of the transmitter governs the range over which a Bluetooth device can operate and, generally, devices are said to fall into one of three classes: class 1 are the most powerful and can operate up to 100m (330ft), class 2 (the most common kind) operate up to 10m (33ft), and class 3 are the least powerful and don't go much beyond 1m (3.3ft). The Packmule will use a mobile phone to issue manual commands, and mobile phones are currently using class 2 [5].

Marketing Requirements

1. The robot should be able to able to carry a weighted load.
2. The robot must be able to move at a standard walking speed.
3. The robot should include safety mechanisms.
4. The robot should move autonomously.
5. The robot should be able to follow the path of the user.
6. The robot should be able to navigate around obstacles in its path.
7. The robot must have a long enough battery life to complete the job.
8. The robot should be able to be manually controlled by user inputs.
9. The robot must be able to traverse outdoor terrain.
10. The robot must stay within a certain following range of user.

Objective Tree

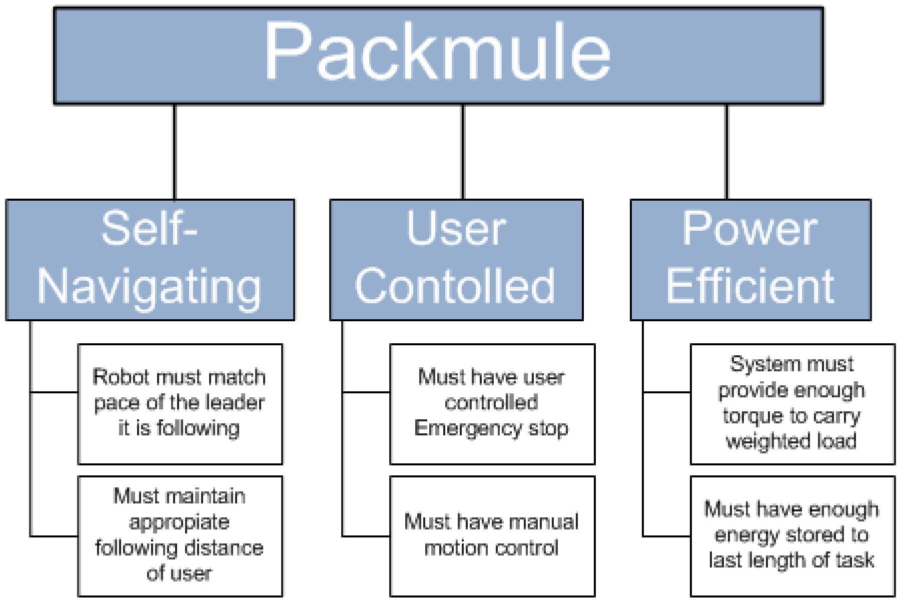

Packmule
Self-Navigating
User Contolled
Power Efficient
Robot must match pace of the leader it is following
Must maintain appropiate following distance of user
Must have user controlled Emergency stop
Must have manual motion control
System must provide enough torque to carry weighted load
Must have enough energy stored to last length of task

Figure 5: Packmule Objective Tree

Design Requirements Specification

Marketing Requirements	Engineering Requirements	Justification
7	Power Supply must have enough energy to run Packmule for one hour.	Packmule must be able to follower user to distant destinations.
1,9	Packmule must carry load up to 30 lbs in addition to weight of robot.	Packmule must handle moderate loads to be useful.
4,9	Packmule must transverse surfaces with coefficient of friction up to μ=.8	Outdoor surface of grass coefficient of friction is .8.
4,9	Packmule must be able to traverse slopes up to 10 degrees.	Must be able to handle casual outdoor slopes.
3,4,6	Packmule must be able to identify obstacles within 40 cm of path.	The robot must be able to avoid collision with objects in its path.
5	Packmule will use multiple sensors to identify position	Accurate position approximation takes at least 3 signals to compute.

3,8	Packmule must respond to user inputs via wireless communication.	The user must have limited control of robots movement.
2,4,10	Packmule must be able to travel at speeds up to 4 mph.	Should exceed average walking speed of 3 miles per hour
5,10	Packmule must be able to detect position of user within a 5m radius.	Packmule must be able to detect user within a certain range.
3,4	Packmule must perform emergency stop if user's position is undetected.	Robot should be positive of user location to ensure safe operation.

Table 2: Engineering requirements

Accepted Technical Design

Theory of Operation

The user will wear a device that transmits a wireless ultrasonic signal at 40 kHz. Five ultrasonic sensors mounted on the body of the Packmule will receive the wireless ultrasonic signal. This signal will be filtered and amplified into a signal that can easily be read by the microcontroller. The amplitude, or strength, of the signal will be analyzed by the Position Detection Routine algorithms programmed into the microcontroller to determine the position of the transmitted signal. In order for the autonomous Packmule to avoid hitting obstacles that may come in the path between the user and the Packmule, three infrared sensors that transmit and receive a beam of infrared light will be mounted to the front of the body. If an object is in the sensor's path, the transmitted beam will reflect off it and return to the receiver of the sensor. The strength of the reflected light will be stronger the closer the object is to the Packmule. This signal strength will be read by the microcontroller and analyzed by the Obstacle Detection Routine to determine whether an object is present within range of the Packmule. The Path Planning Algorithm will take inputs from the Obstacle Detection Routine and Position Detection Routine to determine the motor operation needed to accurately follow the user. An Android and iOS app will be built to allow for manual override of the Packmule motor operations.

Level 0 Block Diagrams

Level 0 Software Block Diagram

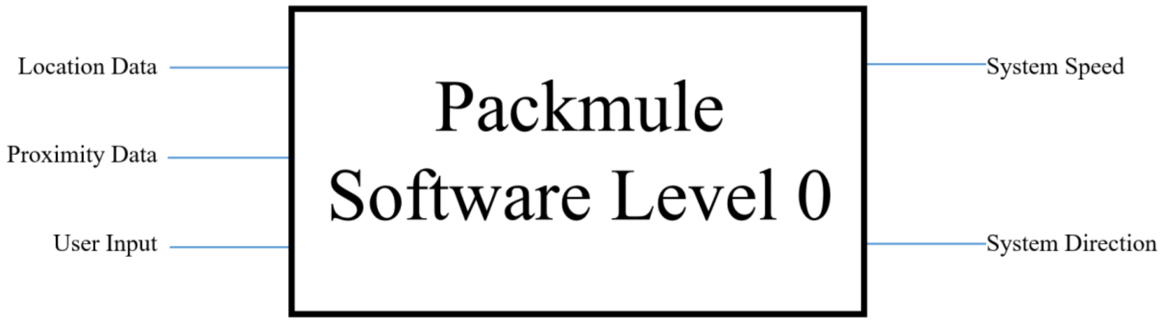


Figure 6: Level 0 Software Block Diagram

Module	Level 0 Software Block Diagram
Designed by	Jared Ford
Inputs	User Location Data, Obstacle Proximity Data, User Directional Inputs
Outputs	Speed and Location Data
Description	The system will take in data from sensors and user input to determine what path it will take. The calculated information will be passed along to the motor controller.

Table 3: Level 0 Software Specifications

Level 0 Hardware Block Diagram

s

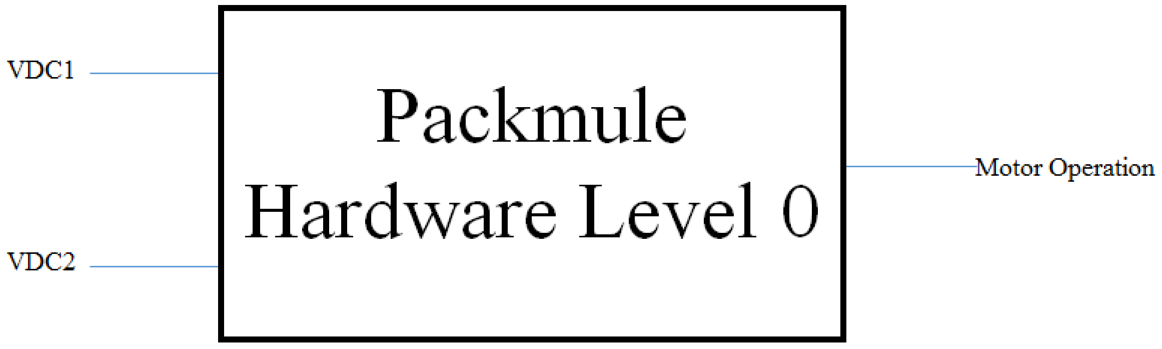


Figure 7: Level 0 Hardware Block Diagram

Module	Level 0 Hardware Block Diagram
Designed by	Jared Alexander / Andray Pennington
Inputs	VDC1: +9VDC used to power components in the Packmule. VDC2: +24VDC used to power components in the Packmule.
Outputs	Motor Operation
Description	The Packmule takes in DC voltages to power the components of the system that will send information to the microcontroller. Using this data, the microcontroller will determine the position of the user through algorithms. Based on this information, signals will be sent to the motor driver to output accurate motor operation.

Table 4: Level 0 Hardware Specifications

Level 1 Block Diagrams

Level 1 Packmule Software Diagram

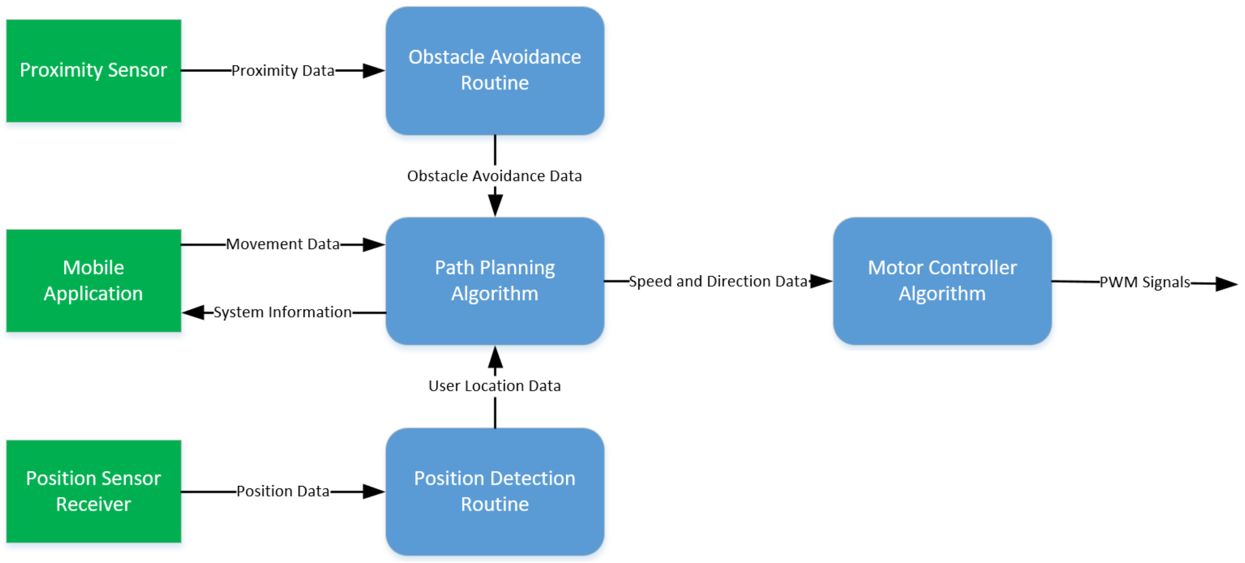


Figure 8: Level 1 Software Diagram

Module	Obstacle Avoidance Routine
Designed by	Jared Ford
Inputs	Proximity Sensor Data
Outputs	Obstacle Avoidance Data
Description	The routine will check if an obstacle is within a specified distance from the Packmule. If such an obstacle exists, this information will be passed along to the path planning algorithm. This, in effect, will trigger an interrupt sequence that temporarily overrides the user following routine.

Table 5: Obstacle Avoidance Routine

Module	Path Planning Algorithm
Designed by	Jared Ford
Inputs	Obstacle Avoidance Data, Directional Signals, and Position Data
Outputs	Speed and Direction Data
Description	The system will either operate in one of two modes. In manual mode which purely relies on directional signals from user input to generate speed and direction data. In autonomous mode, the system will locate the user and plan a path to him or her. If an obstacle is encountered while in autonomous mode, the algorithm will find a new path to the user.

Table 6: Path Planning Algorithm

Module	Position Detection Routine
Designed by	Jared Ford
Inputs	Position Data
Outputs	User Location Data
Description	An array of sensor data will be sent to the routine. From these signals, it will be able to calculate the distance each sensor is from the user and pass this information to the path planning algorithm.

Table 7: Position Detection Routine

Module	Motor Controller Algorithm
Designed by	Jared Ford
Inputs	Speed and Direction Data
Outputs	PWM Signals
Description	Speed and direction data is read in from the path planning algorithm. This data is then converted into serial data that is passed along to the Sabertooth motor driver. The library is able to parse integer values in the range of -127 to 127 for both speed and direction signals. These numbers are then converted, into PWM signals that the motor will recognize and respond to.

Table 8: Motor Controller Algorithm

Module	Mobile Application
Designed by	Jared Ford
Inputs	System Information
Outputs	Directional control signals
Description	System information is received wirelessly from the Packmule system and is displayed on the mobile phone application. Directional control signals are generated within the application and sent to the Packmule. The mobile application features an emergency stop function that can prevent movement at any time when the Packmule is in autonomous following mode

Table 9: Mobile Application

Level 1 Hardware Block Diagram

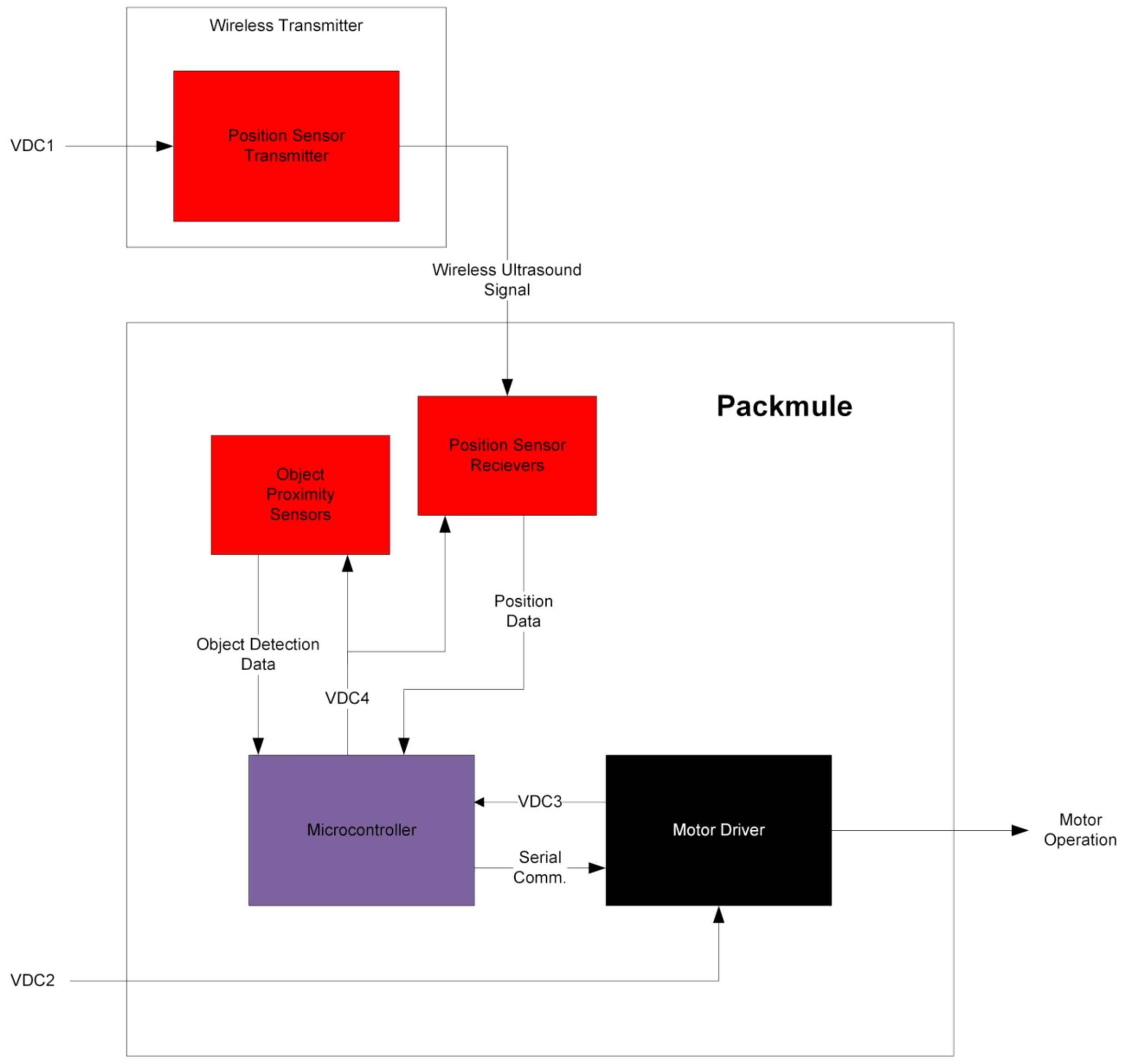


Figure 9: Level 1 Hardware Block Diagram

Module	Level 1, Hardware Design
Designed by	Jared Alexander / Andray Pennington
Inputs	VDC1: +9VDC used to power components in the wireless ultrasonic transmitter worn by the user. VDC2: +24VDC used to power the motor driver. VDC3: +5VDC used to power the Microcontroller. VDC4: +5VDC used to power the Position Sensor Receivers and the Object Proximity Sensors.
Outputs	Signal to left and right motor to ensue movement.
Description	The wireless Position Sensor Transmitter transmits an ultrasonic pulse that is received by each Position Sensor Receiver. The different amplitudes of the pulse received at each Position Sensor Receiver are sent to the Microcontroller. At the same time, the Object Proximity Sensors transmit infrared light in the forward direction of the Packmule to detect objects in its path and send this information to the Microcontroller. With this information, the Microcontroller determines the appropriate motor operation and sends a Serial Communication signal to control the Motor Driver, which in turn drives each motor.

Table 10: Level 1 Hardware Specifications

Level 2 Block Diagrams

Level 2 Software Path Planning

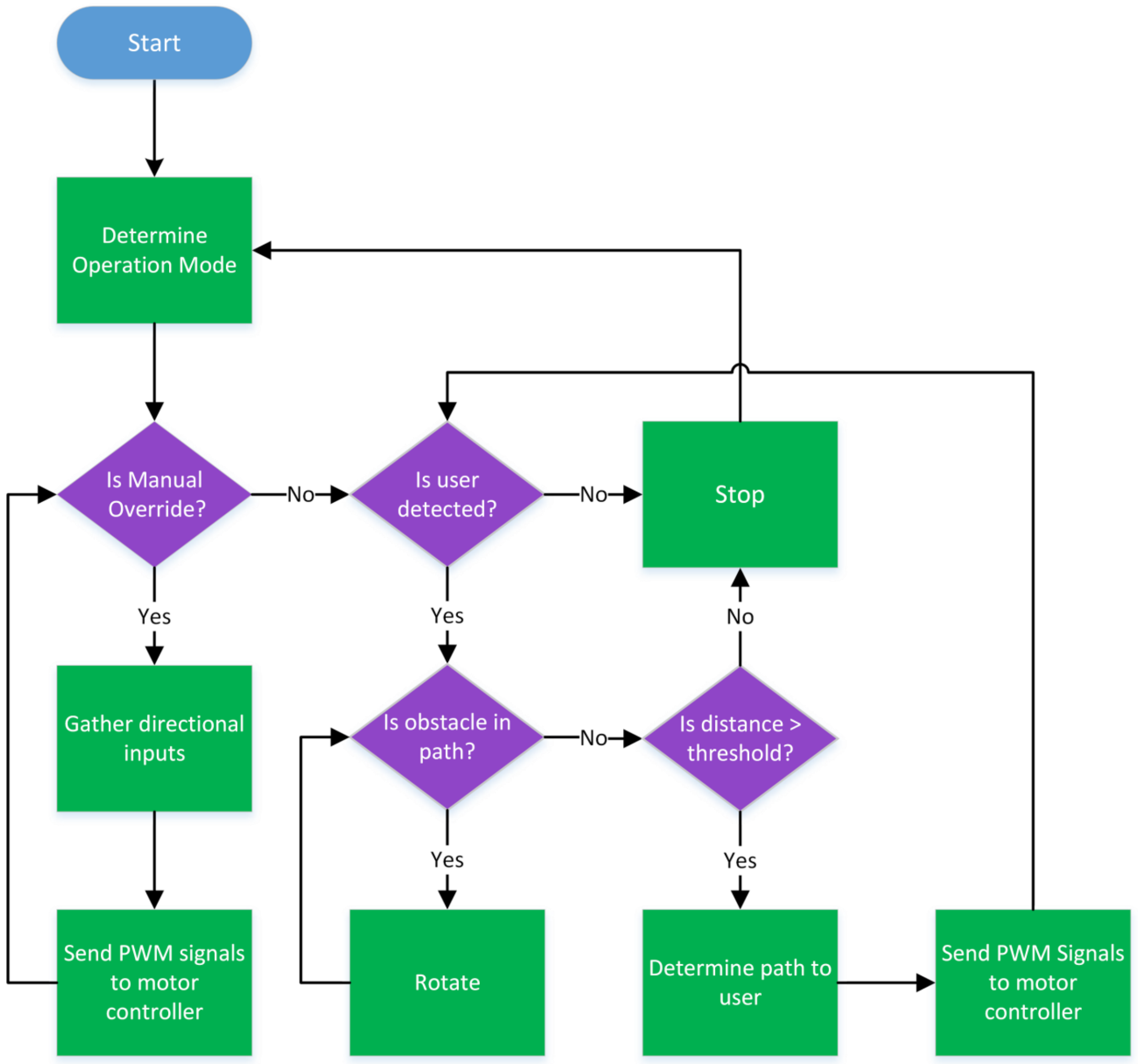


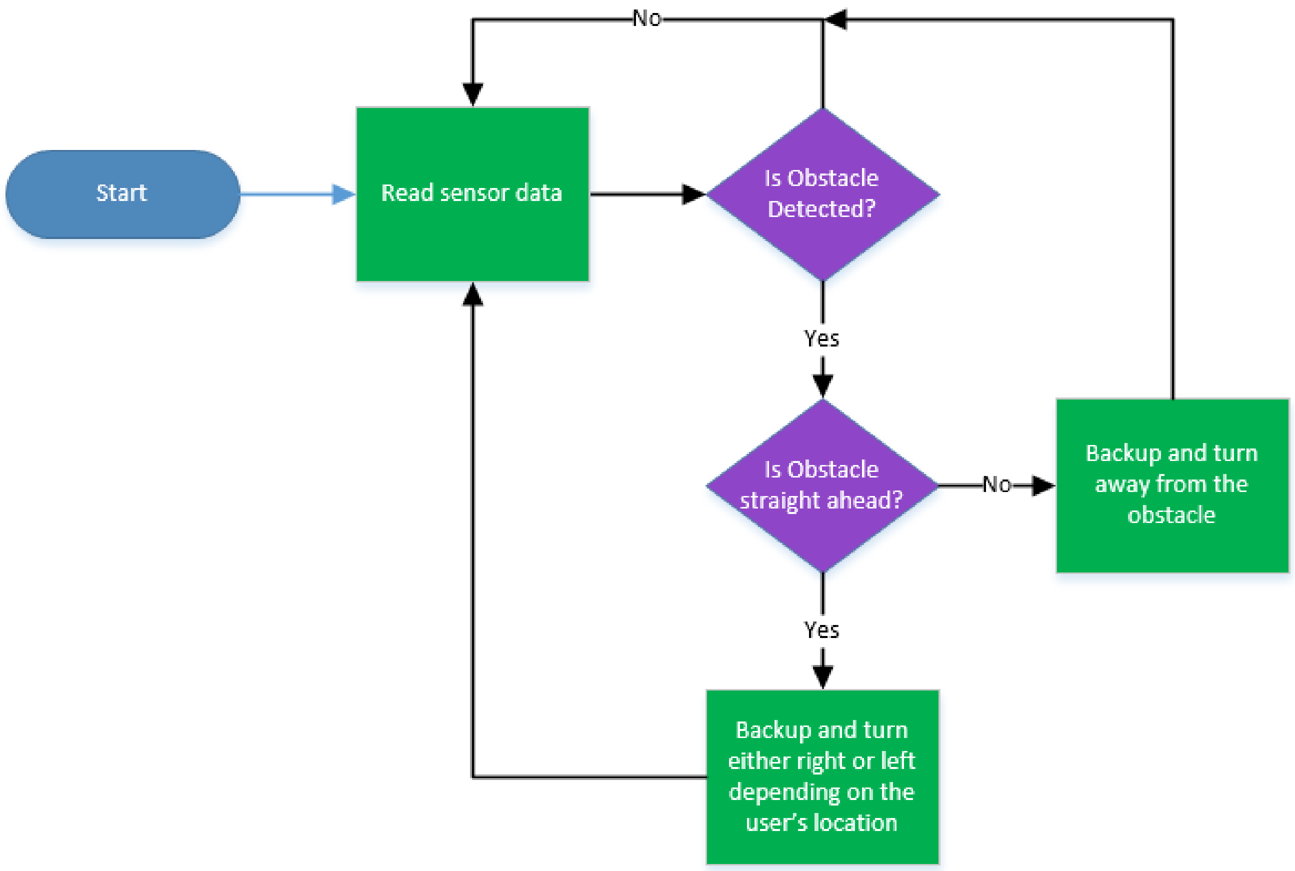
Figure 10: Level 2 Path Planning Software Diagram

```
void followUser(){
  // Variables to hold sensor readings
  int l, fl, fc, fr, r;
  // Read from all the sensors
  l = analogRead(RECEIVER_L);
  fl = analogRead(RECEIVER_FL);
  fc = analogRead(RECEIVER_FC);
  fr = analogRead(RECEIVER_FR);
  r = analogRead(RECEIVER_R);
  int dir = getDirection(l, fl, fc, fr, r);

 switch(dir){
    case DIRECTION_LEFT:
      ST.turn(-30);
      ST.drive(0);
      Serial2.print("Hard Left");
      //Serial.println("hard left");
      break;
    case DIRECTION_SLIGHT_LEFT:
      ST.turn(-15);
      ST.drive(20);
      Serial2.print("Left");
      //Serial.println("left");
      break;
    case DIRECTION_FORWARD:
      ST.turn(0);
      ST.drive(40);
      Serial2.print("Forward");
      //Serial.println("forward");
      break;
    case DIRECTION_SLIGHT_RIGHT:
      ST.turn(15);
      ST.drive(20);
      Serial2.print("Right");
      //Serial.println("right");
      break;
    case DIRECTION_RIGHT:
      ST.turn(30);
      ST.drive(0);
      Serial2.print("Hard Right");
      //Serial.println("hard right");
      break;
    case -1: // We only want to fall into this case if several consecutive errors have occurred
      ST.turn(0);
      ST.drive(0);
      if(errorCleared){
        Serial2.print("User Lost");
        errorCleared = false;
      }
      Serial.println("error");
      break;
     default:
        successCount++;
        //Serial.println("Adding Success");
        break;
  }
  Serial.print(l);
  Serial.print(" ");
  Serial.print(fl);
  Serial.print(" ");
  Serial.print(fc);
  Serial.print(" ");
  Serial.print(fr);
  Serial.print(" ");
  Serial.print(r);
  Serial.println();
}

void clearErrors(){
  errors = 0;
  if(!errorCleared){
        Serial2.print("");
        errorCleared = true;
    }
}
```

Table 11: Path Planning Pseudocode

The path planning algorithm takes in a speed as a floating point value and a directional input as a string. The algorithm first checks if the system is in manual mode or not. If it is, then the speed and direction values from the manual user input are passed along to the motor controller. Otherwise, it looks for the user. By making a call to the position detection routine, the algorithm will know whether or not the user has been found. If the user has been found, a check is performed to see if there is an obstacle in the path. The system will move away from any obstacles and proceed to follow the user by determining their location and sending an appropriate speed and direction value to the motor controller algorithm maintain the proper distance from the user. Finally, if the user cannot be found, a signal is sent out alerting the user that he/ or she is out of range of the system. The rest of the code can be found at https://github.com/jrodbossman.

Level 2 Software Obstacle Avoidance



Start
Read sensor data
Is Obstacle Detected?
No
Yes
Is Obstacle straight ahead?
No
Backup and turn away from the obstacle
Yes
Backup and turn either right or left depending on the user's location

Figure 11: Level 2 Obstacle Avoidance Software Diagram

```
void checkIR() {
  if(analogRead(IR_LEFT) > IR_THRESHOLD) {
    if(leftObstacle >= IR_SUCCESS_THRESHOLD) {
      ST.drive(REVERSE);
      ST.turn(RIGHT);
      delay(1000);
      //Serial.println("left");
      ST.drive(FORWARD);
      ST.turn(0);
      delay(1500);
      ST.drive(0);
      ST.turn(0);
      leftObstacle = -1;
    }
    leftObstacle++;
  }
  else {
    leftObstacle = 0;
  }
  if(analogRead(IR_CENTER) > IR_THRESHOLD) {
    if(centerObstacle >= IR_SUCCESS_THRESHOLD) {
      ST.drive(REVERSE);
      ST.turn(0);
      delay(1000);
      recoverFromCenterObstacle();
      //Serial.println("center");
      ST.drive(0);
      ST.turn(0);
      centerObstacle = -1;
    }
    centerObstacle++;
  }
  else {
    centerObstacle = 0;
  }
  if(analogRead(IR_RIGHT) > IR_THRESHOLD) {
    if(rightObstacle >= IR_SUCCESS_THRESHOLD) {
      ST.drive(REVERSE);
      ST.turn(LEFT);
      delay(1000);
      //Serial.println("right");
      ST.drive(FORWARD);
      ST.turn(0);
      delay(1500);
      ST.drive(0);
      ST.turn(0);
      rightObstacle = -1;
    }
    rightObstacle++;
  }
  else {
    rightObstacle = 0;
  }

  Serial.print(analogRead(IR_LEFT));
  Serial.print(" ");
  Serial.print(analogRead(IR_CENTER));
  Serial.print(" ");
  Serial.print(analogRead(IR_RIGHT));
  Serial.println();
}
```

```
void recoverFromCenterObstacle() {
  if(analogRead(RECEIVER_FL) >= analogRead(RECEIVER_FR)) {
    ST.drive(0);
    ST.turn(LEFT);
  }
  else {
    ST.drive(0);
    ST.turn(RIGHT);
  }
  delay(1000);
  ST.drive(FORWARD_SLOW);
  ST.turn(0);
  delay(500);
}
```

Table 12: Obstacle Avoidance Pseudocode

The obstacle algorithm first issues commands to all of the IR sensors to broadcast beams of light and observe the amount of light reflected. The intensity of the reflected light is then calculated by the algorithm and the relative distance from the obstacle (if one is detected) is compared on a sensor by sensor basis to a minimum threshold value that specifies how close the system can get to an obstacle. If any sensor is less than the threshold, the algorithm will determine in which direction to move the system and relay this information back to the path planning algorithm. The rest of the code can be found at https://github.com/jrodbossman.

Level 2 Software Position Detection

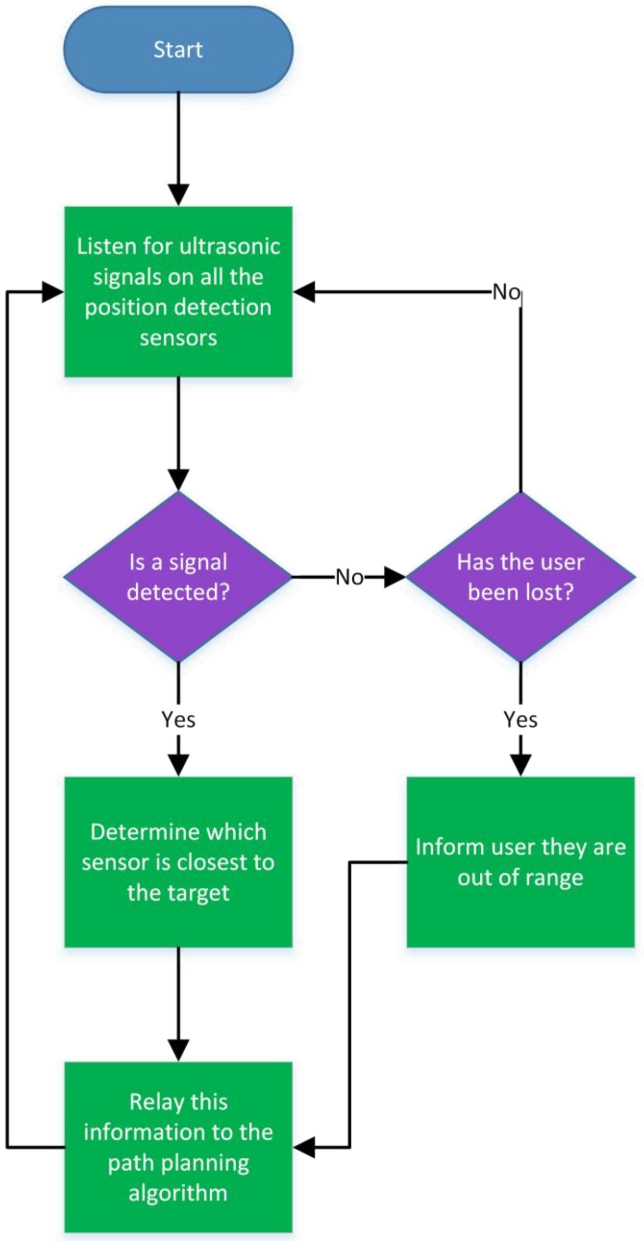

Start
Listen for ultrasonic signals on all the position detection sensors
No
Is a signal detected?
No
Has the user been lost?
Yes
Yes
Determine which sensor is closest to the target
Inform user they are out of range
Relay this information to the path planning algorithm

Figure 12: Level 2 Position Detection Software Diagram

```
// Function to determine where the user is
int getDirection(int l, int fl, int fc, int fr, int r) {
  int max = 0, direction = DIRECTION_LEFT;
  if(l > max){
    max = l;
  }
  if(fl > max) {
    max = fl;
    direction = DIRECTION_SLIGHT_LEFT;
  }
  if(fc > max){
    max = fc;
    direction = DIRECTION_FORWARD;
  }
  if (fr > max){
    max = fr;
    direction = DIRECTION_SLIGHT_RIGHT;
  }
  if(r > max) {
    max = r;
    direction = DIRECTION_RIGHT;
  }
  if (max < 100){
    errors++;
    successCount = 0;
    if(errors > 5){
      return -1;
    }
    return previousDir;
  }
  clearErrors();
  if(successCount > 10){
    successCount = 11;// Preventing integer from getting too large
    Serial2.print(" ");
    previousDir = direction;
    return direction;
  }
  return 's';
}
```

Table 13: Position Detection Pseudocode

The position detection routine iterates through all of the ultrasonic transducers and compares the amplitudes. The largest amplitude received corresponds to the sensor that is closest to the user. The system will also store the next closest sensor's data. Once these two numbers are known, the algorithm will compare their difference to a threshold value that will determine if they are just about the same distance from the user and store this in a boolean variable to later be used by the path planning algorithm. Once the algorithm completes its execution, the path planning algorithm will parse this information and plan a route to the user. The rest of the code can be found at https://github.com/jrodbossman.

Level 2 Software Motor Controller

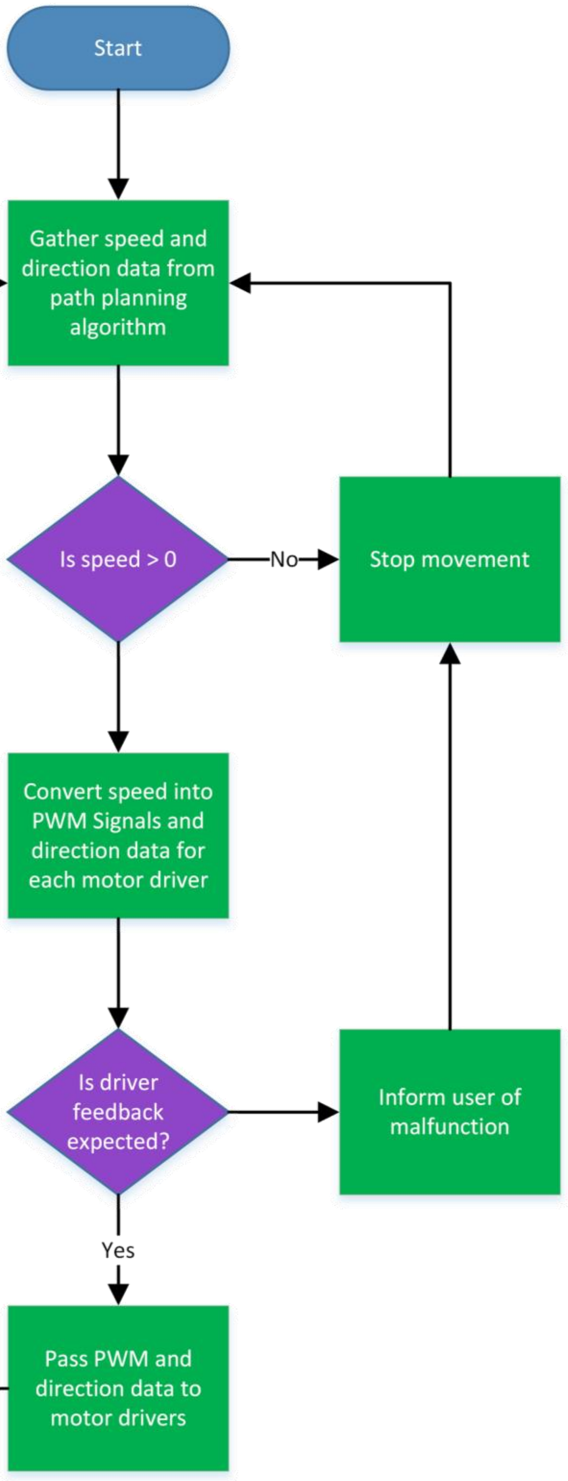

Start
Gather speed and direction data from path planning algorithm
Is speed > 0
No
Stop movement
Convert speed into PWM Signals and direction data for each motor driver
Is driver feedback expected?
Inform user of malfunction
Yes
Pass PWM and direction data to motor drivers

Figure 13: Level 2 Motor Controller Software Diagram

```
// We are able to utilize a library that makes controlling the motor very straight forward

// To turn, we employ ST.turn(int value) where value can range from -127 to 127
// To drive, we employ ST.drive(int value)w here value can range from -127 to 127

void motorControllerExample(Direction directionData){
  ST.drive(directionData.speed);
  ST.turn(directionData.direction);
  // To stop
  ST.drive(0);
  ST.turn(0);
}
```

Table 14: Motor Controller Pseudocode

The job of the motor controller algorithm consists of taking in speed, direction, and degree of turning if applicable and converting the speed into PWM values for the left and right motor as well as a direction for each motor to travel in. First, the algorithm assumes that both motors will be traveling at the speed the system takes in by performing the speed to PWM routine on the given value. Next, it assumes that both motors will be traveling in the forward direction. Finally, the algorithm uses the provided direction string to alter the initial values. In the reverse case, both motors will move at full speed, but in the reverse direction. In the forward case, none of the initial values will change. If the system needs to turn right, then the left motor will move about twice as fast as the right motor and if a hard turn is needed, then the right motor will spin in the reverse direction. For the turn left case, the values will be the opposite of the case in which the system turns right. For the purposes of the Packmule system, a tank style turning algorithm was implemented. The rest of the code can be found at https://github.com/jrodbossman.

Level 2 Software Mobile Application

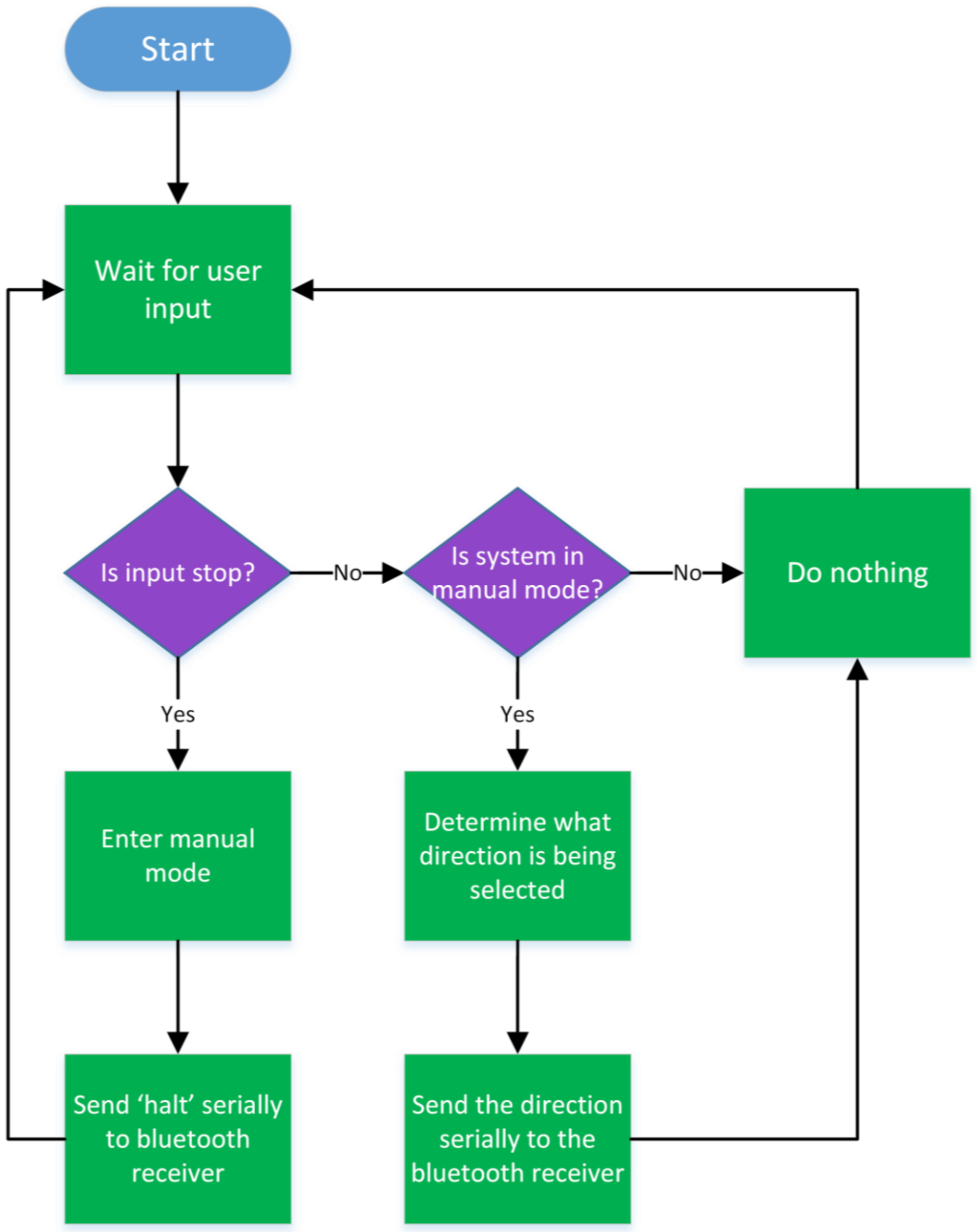

Start
Wait for user input
Is input stop?
No
Is system in manual mode?
No
Do nothing
Yes
Yes
Enter manual mode
Determine what direction is being selected
Send 'halt' serially to bluetooth receiver
Send the direction serially to the bluetooth receiver

Figure 14: Level 2 Mobile Application Software Diagram

```
void onJoyStickTouch(){
                int direction = GetJoystickDirection();
                if(isManualMode){
                  if(direction == JoyStick.STICK_UP) {
                                bluetooth.sendSerial(JoyStick.Magnitude + "," + DIRECTION_FORWARD);
                   }
                            if(direction == JoyStick.STICK_DOWN) {
                                        bluetooth.sendSerial(JoyStick.Magnitude + "," + DIRECTION_REVERSE);
                            }
                            if(direction == JoyStick.STICK_RIGHT) {
                               bluetooth.sendSerial(JoyStick.Magnitude + ","+ DIRECTION_RIGHT);
                            }
                            if(direction == JoyStick.STICK_LEFT) {
                               bluetooth.sendSerial(JoyStick.Magnitude + "," + DIRECTION_LEFT);
                   }
                            else {
                                        bluetooth.sendSerial("0,0");
                            }
                }
   else if(direction == JoyStick.STOP){
                            manualMode = true;
                            bluetooth.sendSerial("0,0");
                }
}

void onModeSwitchChanged(){
    bluetooth.sendSerial(switch.getCurrentState());
}

void onDataReceived(string  receivedMessage){
    dataTextBox.setText(receivedMessage);
}
```

Table 15: Mobile Application Pseudocode

The mobile application communicates serially over Bluetooth with the path planning algorithm via a Bluetooth interrupt command. A joystick will listen for touch events and send the corresponding direction and speed based on the location of the touch. Furthermore, the application will allow for the switching between manual and follower modes via a button press which also employs the use of a touch event listener. Finally, any pertinent system data originating from the Packmule system will be transmitted to the mobile application and displayed in a message area for the user to see. This can be anything ranging from a low battery signal to a user not found alert. The rest of the code can be found at https://github.com/jrodbossman.

Level 2 Hardware: Object Detection

The sharp ir sensor uses triangulation and a small linear CCD array to compute the distance and/or presence of objects in the field of view. A pulse of IR light is emitted by the emitter, and travels out into the field of view. In the case of no object, the light is never reflected, and the reading shows no object. If the light reflects off an object, it returns to the detector and creates a triangle between the point of reflection, the emitter and the detector. The incident angle of the reflected light varies based on the distance to the object. The receiver portion of the IR sensor is a precision lens that transmits reflected light onto various portions of the enclosed linear CCD array based on the incident angle of the reflected light. The CCD array can then determine the incident angle, and thus calculate the distance to the object. [7]

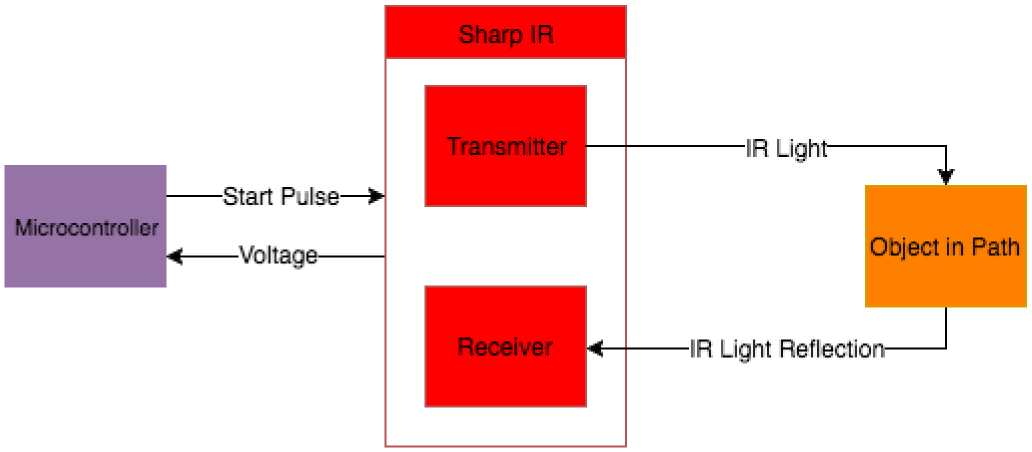


Figure 15: Object Detection

Module	Level 2 Hardware: Object Detection
Designed by	Andray Pennington
Output	Voltage Range: 3.1V at 10cm to 0.4V at 80cm
Description	Detects objects in the direct path of the Packmule by using an infrared transmitter to send a light pulse, and an infrared receiver to read the reflection. The ir sensor will then output a voltage ranging from 3.1V at 10cm to 0.4V at 80cm to the microcontroller.

Table 16: Object Detection Specifications

Level 2 Hardware: Wireless Transmitter

The user will wear a wireless transmitter on their belt that will allow the Packmule to locate and create a path to follow the user. A power supply consisting of two 9V batteries connected in parallel will feed the Transmitter Voltage Regulator that will step down the voltage to 5VDC to feed the Arduino Uno or the Timer circuit. A switch allows the batteries to be disconnected from the rest of the circuit to prevent drainage on the batteries. This schematic can be found in Figure 20. The Arduino Uno and the Timer circuit output an oscillatory pulse train centered at 40 kHz. Values of R1, R2, and C can be calculated using Equation 2.

$$fc = \frac{1}{0.693C(R1 + 2R2)}$$

Equation 2: Timer Frequency [3]

With the Arduino Uno setup, the regulated 5V will input into Vin port. The Ultrasonic Transducer will connect to the selected Arduino analog output pin to excite it with a 40 kHz wireless ultrasonic signal. Similarly the Timer Circuit will input 5V from the Voltage Regulator. The 555 Timer along with its calculated resistor and capacitance values will produce a 40kHz pulse that will excite the ultrasonic transducer. This schematic can be found in Figure 21.

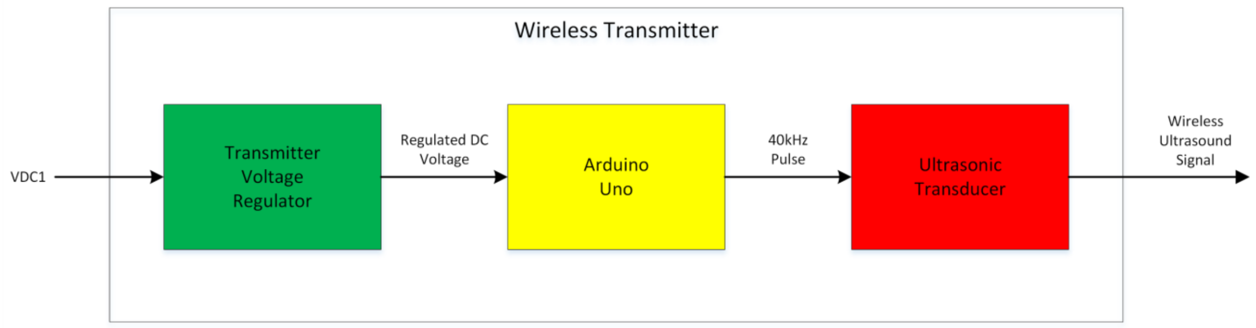


Figure 16: Wireless Transmitter using Arduino Uno Level 2 Block Diagram

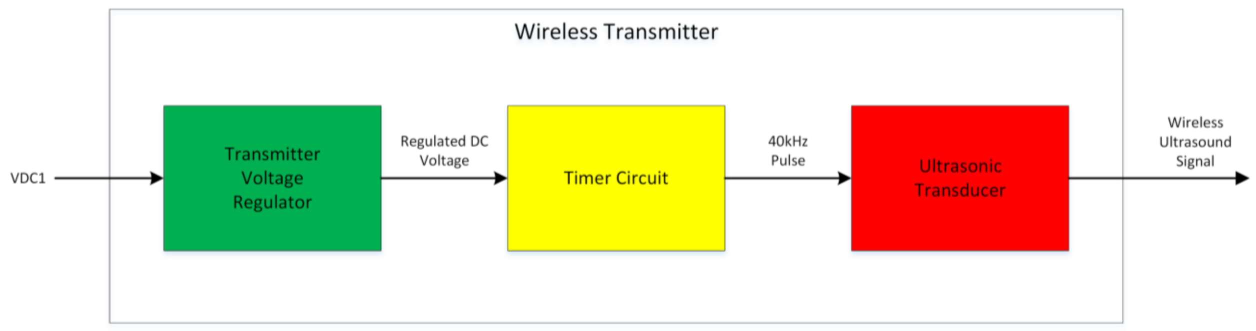


Figure 17: Wireless Transmitter using Timer Circuit Level 2 Block Diagram

Module	Wireless Transmitter, Hardware Design
Designed by	Jared Alexander
Inputs	VDC1: +9VDC used to feed the Transmitter Voltage Regulator.
Outputs	The ultrasonic transducer produces a wireless ultrasound signal.
Description	The wireless transmitter contains its own power supply, batteries, that feeds the Transmitter Voltage Regulator. The Transmitter Voltage Regulator produces a steady +5VDC to power the rest of the Wireless Transmitter, the Arduino Uno or the Timer Circuit. The Arduino Uno or the Timer Circuit creates a pulse signal that then excites the Ultrasonic Transducer and emits a wireless ultrasound signal.

Table 17: Wireless Transmitter Level 2 Specifications

Level 2 Hardware: Position Sensor

There will be a total of five position sensors mounted on the Packmule to receive the Wireless Ultrasound signal from the transmitter worn by the user. Each sensor will contain an Ultrasonic Transducer to receive the Wireless Ultrasound signal. The signal will cause the transducer to excite a voltage. Because the Wireless Ultrasound signal will be attenuated after traveling through the air, it is required that it be amplified prior to being read by the Arduino Mega. It must be strong enough to be easily read by this component and so that it is stronger than potential noise in its surroundings. To achieve this, a two stage BJT amplifier circuit is used. The received Wireless Ultrasonic signal will be passed through two n-type BJT transistors that will amplify the attenuated signal. This circuit uses a 2 MΩ potentiometer that gives each circuit the ability to be tuned to read equal values. Each circuit was tested and tuned to be equal so that they would report equal data to the Arduino Mega. The final stage before the voltage signal is sent to the microcontroller is the Envelope Detector stage. This stage uses the Envelope Detector circuit. This circuit causes sampling of the sinusoidal signal at the rising edges of the sinusoid. Positive voltage values that are easier for the microcontroller to read will be the output. The configuration is a 1N4148 diode with a parallel connecting capacitor, C, and resistor, R. Equation 7 shows that the time constant, RC, must be an order of magnitude greater than center frequency period, 1/fc. Through testing of the circuit, it was found that a 0.1uF capacitor and 1MΩ resistor allowed for the best reading of the Wireless Ultrasound signal. The center frequency of this signal once again is 40 kHz.

$$\frac{1}{fc} << RC$$

Equation 2: Envelope Detector Time Constant

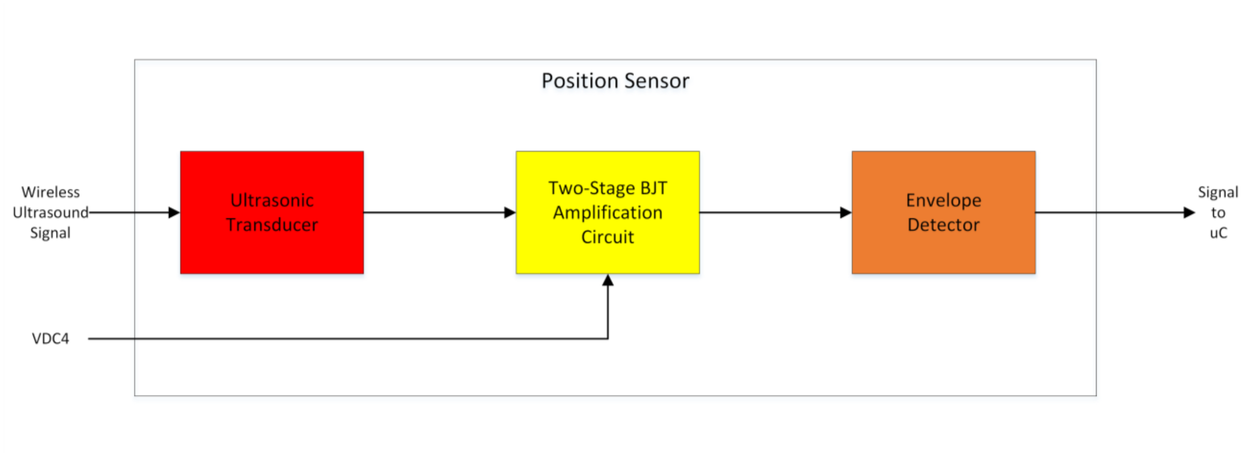


Figure 18: Position Sensor Level 2 Block Diagram

Module	Position Sensor, Hardware Design
Designed by	Jared Alexander
Inputs	VDC4: +5VDC from the Arduino Mega 2560 used to power the BJT Amplification Circuit. Wireless Ultrasound Signal sent from the Wireless Transmitter.
Outputs	Voltage signal sent to the microcontroller.
Description	VDC4 feeds the Sensor Voltage Regulator with +5VDC. The Wireless Ultrasound Signal is received by the Ultrasonic Receiver. This signal is the sent through the Two Stage BJT Amplification Circuit to increase the signal that is received by the Ultrasonic Transducer. This signal is then sent to an Envelope Detector circuit. The output voltage of this circuit will be sent to the microcontroller.

Table 18: Position Sensor Level 2 Specifications

Level 2 Hardware: Motor Driver Block

To reduce the cost of the project, the design team accepted a donated electric wheelchair base. This existing system also placed physical limitations upon the design. The initial system utilized 10 in (0.42 ft) diameter wheels, had an approximated weight of 70 lbs (with batteries).

Electrically, the system operated 2, 24V DC motors. With these limitations, the specifications of the Packmule drive subsystem were able to be calculated. The Packmule is required to carry an additional 30 lbs (Engineering requirement 2)

To meet the 4 mile per hour requirement (engineering requirement 8) the required wheel velocity needed to be calculated. This was done in the following manner:

$$\omega = \frac{Velocity}{Radius\ of\ wheel} = \frac{5.86fps}{.42f} = 14.0808\frac{rad}{s} * \frac{1}{2\pi}\frac{rev}{rad} = 2.24\frac{rev}{s}$$

Equation 3: Required Wheel Velocity

With the required speed of the wheels determined, the torque needed to be calculated. Prior to the torque calculation, the amount of force required to move Packmule needed to be calculated. This is done in several different situations. The first situation is continuous motion on a flat surface. When in continuous motion, the Packmule has to overcome the force from rolling friction, which is much lower than that of static friction. This requires much less torque from the Packmule. From research [6] the average rolling resistance found between grass and tire is 0.007 with a standard deviation of .002 . A rolling resistance value of 0.015 is used to cover most situations. The force required by the motors to overcome the rolling frictional forces were calculated as follows:

$$Force = Coefficent\ of\ Friction * Normal\ Force = 0.015 * 100lbs = 1.5lbs$$

Equation 4: Situation 1 Force equation

Next the torque required by the motor was calculated:

$$Torque = radius\ of\ wheel * Force = .42ft * 1.5lbs = .625ft\ lbs$$

Equation 5: Situation 1 Torque Equation

This value is the total required amount of torque. With 2 motors, the required torque per motor is half of the total amount. Each motor is required to fulfill 0.3125 ft lbs of torque in this situation.

Next the power of the motor was calculated:

$$Power = Torque * rotational\ velocity = .3125ft\ lbs * 14.08\frac{rad}{s} = 4.4003ft\frac{lbs}{s}$$

Equation 6: Situation 1 Power Equation

Converting to Watts:

$$4.4003ft\frac{lbs}{s} * \frac{745\,Ws}{550\,ft\,lbs} = 5.9603\,W\ per\ motor$$

Equation 7: Situation 1 Watts Conversion

With 2 motors, this leads to 11.921 W of power in this situation.

The second situation is in high friction environments. Through online research [1] it has been determined that the greatest coefficient of static friction encountered by tire on wheel contact is approximately μ=.8. While most coefficients of static friction between rubber and grass were closer to a μ between 0.2 and 0.3, the Packmule must be able to begin motion in the harshest environment. Calculating the motor requirements as before:

$$Force = Coefficient\ of\ Friction * Normal\ Force = 0.8 * 100lbs = 80lbs$$

Equation 8: Situation 2 Force Equation

$$Torque = radius\ of\ wheel * Force = \ .042ft * 80lbs = 33.33ft\ lbs = 16.67ft\ lbs/motor$$

Equation 9: Situation 2 Torque Equation

$$Power = Torque * rotational\ velocity = \ 16.67\ ft\ lbs/motor * 14.08\frac{rad}{s}$$
$$= 234.68\frac{lbs}{s}\ per\ motor$$

Equation 10: Situation 2 Power Equation

$$234.68ft\frac{lbs}{s} * \frac{745\,Ws}{550\,ft\,lbs} = 317.88\,W\ per\ motor = 635.77\ combined\ W$$

Equation 11: Situation 2 Watt Conversion Equation

This situation will likely lead to momentarily overdrawing the motors. With motor drivers rated up to 60A peak motor current, there should not be a problem overcoming this situation. It is very unlikely the Packmule will experience a scenario with this much friction in its expected outdoor usage.

The third situation is moving up an incline of 10 degrees. In this situation, the force calculation is altered to include the amount of force required to overcome the vertical translation in addition to the horizontal translation. In this example the

$$Force = Weight * sin\theta + Weight * Coefficient\ of\ Friction * cos\theta$$
$$Force = 100lbs * sin\ 10 + 100lbs * \ .015 * cos10 = 18.84lbs$$

Equation 12: Situation 3 Force Equation

$$Torque = radius\ of\ wheel * Force = \ .042\ ft * \ 18.84 lbs = 7.91\ ft\ lbs = \ 3.95 ft \frac{lbs}{motor}$$

Equation 13: Situation 3 Torque Equation

$$Power = Torque * rotational\ velocity = 3.95\ \ ft \frac{lbs}{motor} * 14.08 \frac{rad}{s} = 55.71 \frac{lbs}{s} per\ motor$$

Equation 14: Situation 3 Power Equation

$$55.71 ft \frac{lbs}{s} * \frac{745\ W\ s}{550\ ft\ lbs} = \ 75.46 W\ per\ motor = 150.92\ combined\ W$$

Equation 15: Situation 3 Watt Conversion Equation

Once the drive system has been proven to handle each of the three situations, it became necessary to determine the amount of energy the system would use. The Packmule is not expected to operate in the harshest environment for the entire duration of the required 1 hour of continuous operation (Engineering Requirement 1). Based on the three given situations, the Packmule is expected to spend 60% of its operation in situation 1 (36 minutes), 30% in situation 2 (18 minutes), and 10% in situation 3 (6 minutes). Calculating the energy required based on the assumed operation, the expected energy required for one hour is calculated in the following manner.

$$\begin{gathered} Total\ Energy = Power(situation\ 1) * Time(situation\ 1) + Power(situation\ 2) \\ * Time(situation\ 2) + Power(situation\ 3) * Time(situation\ 3) \\ Total\ Energy = (5.96W * 2160s) + (635.77W * 1080s) + (\ 150.92W\ \ * 360s) = 753839.70J \end{gathered}$$

Equation 16: Required Energy Calculation

With the required energy calculated, batteries must be sized to provide the proper battery capacity. Two, 12 V, 33AH batteries running for 1 hour continuously can provide:

$$12V * 33Ah * 3600s * 2\ (batteries) = 2851200J\ available\ energy\ for\ 1\ hour$$

Equation 17: Available Energy Calculation

As this amount is greater than the energy required to run the Packmule for one hour, the will be sufficient power supplies. The batteries provided with the wheelchair were not capable of holding a charge for operation of the wheelchair. Replacement batteries were purchased to the same ratings.

The motor drivers will take in serial signals from microcontroller. Each motor driver independently drives 1 wheel. This allows the Packmule to take turns while maintaining some sense of forward motion. It also allows for smaller radius turns. If both motors were controlled by a single driver the Packmule would only be able to have forward and reverse directions. There would need to be some system in place to use the pivot wheels to steer the Packmule. From no-load laboratory testing Packmule will draw approximately 10A to move at the desired speed. This amount is expected to increase to 14A when the load is added. The selected Sabertooth 2x60 Motor Driver is capable of handling 60 A continuous draw, which is more than satisfactory for the Packmule.

Module	Motor Driver
Designed by	Tim Griffiths
Inputs	VDC2: +24VDC Serial signals containing direction and speed information from microcontroller.
Outputs	Voltage to motor 5V supply to Arduino microcontroller
Description	Once the microcontroller determines the proper speed and direction for each motor, it sends that information via serial communication to the motor drivers. The motor drivers then convert that information

Table 19: Level 2 Motor Driver Block

The motors are Electrocraft 660-204-044's. Based on the peak torque and motor voltage constants provided by the manufacturer, a basic Speed vs. Torque curve can be created. This plot is not entirely representative of the motor characteristics. It was created by extrapolating between these two extreme values. The provided gearboxes allow a 1:2 ratio, which trades motor speed for torque. This will allow our motors to meet the required torques while maintaining the required speed specified in previous calculations.

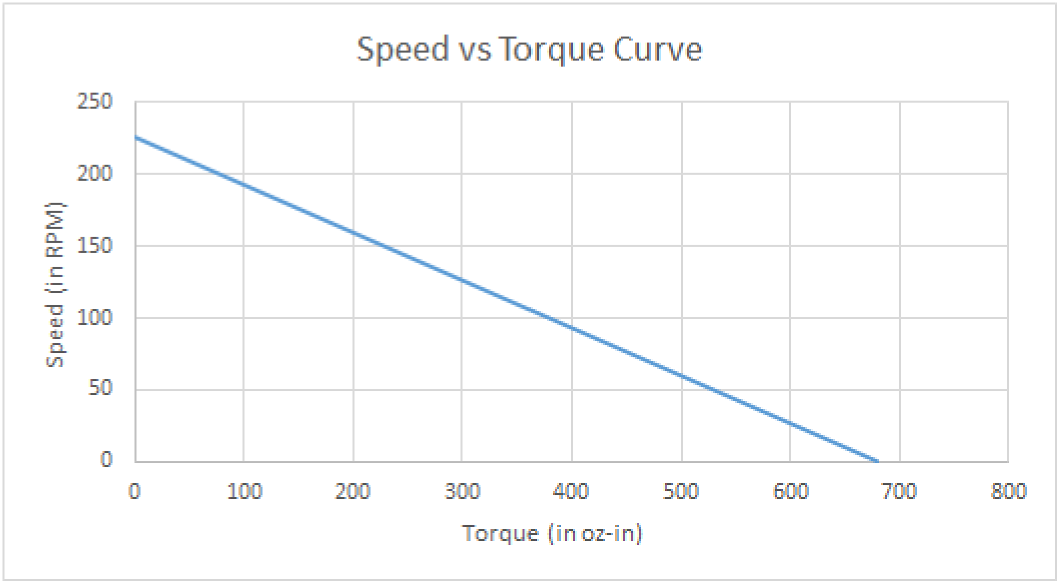


Figure 19: Speed v Torque Curve

Module	Motors
Designed by	Tim Griffiths
Inputs	VDC1: +24 VDC from the 2 series connected 12V batteries. Motor driver enable signal
Outputs	Rotational Motion
Description	The motor receives voltages from the motor drives and responds by turning based on the voltage received. There is not feedback from the motor, instead the Packmule will rely on constant updates from sensors to determine the speed and distance it has travelled.

Table 20: Level 2 Motor Block Diagram

Hardware Schematics

The implemented ultrasonic sensor circuit (Figure 20) will regulate a 9v battery to 5v using a LM7805 voltage regulator. The 5v will then be supplied to an Arduino Uno microcontroller. An ultrasonic transducer is connected to a ground and a digital pin on the microcontroller.

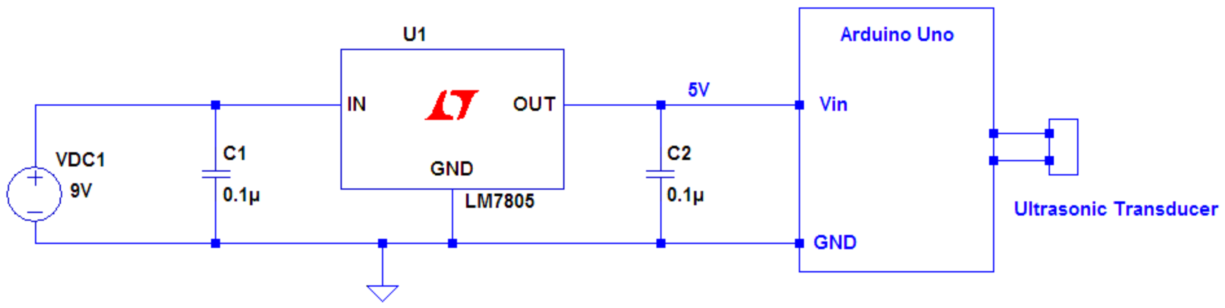


Figure 20: Voltage Regulator and Implemented Ultrasonic Transmitter Circuit

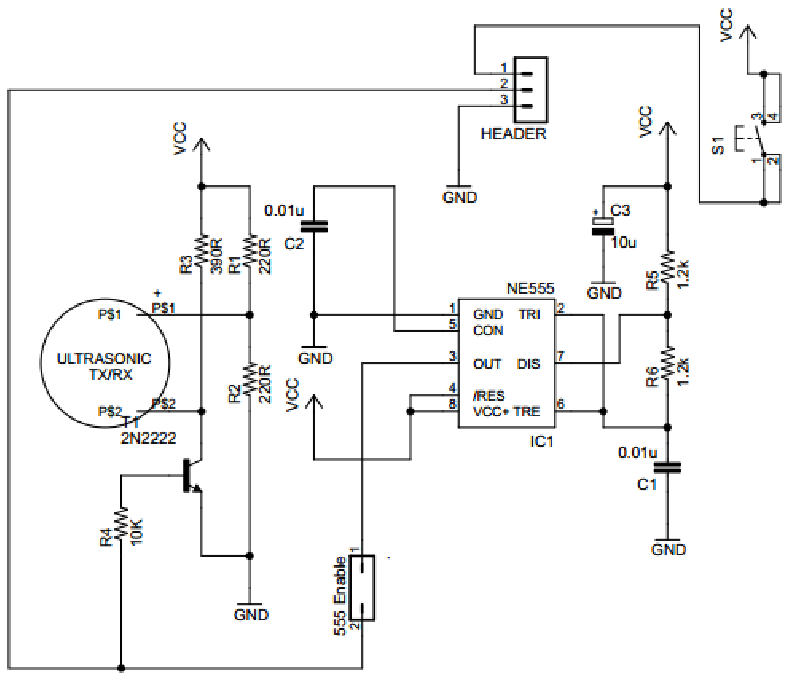


Figure 21: Alternative Ultrasonic Transmitter Timer Circuit

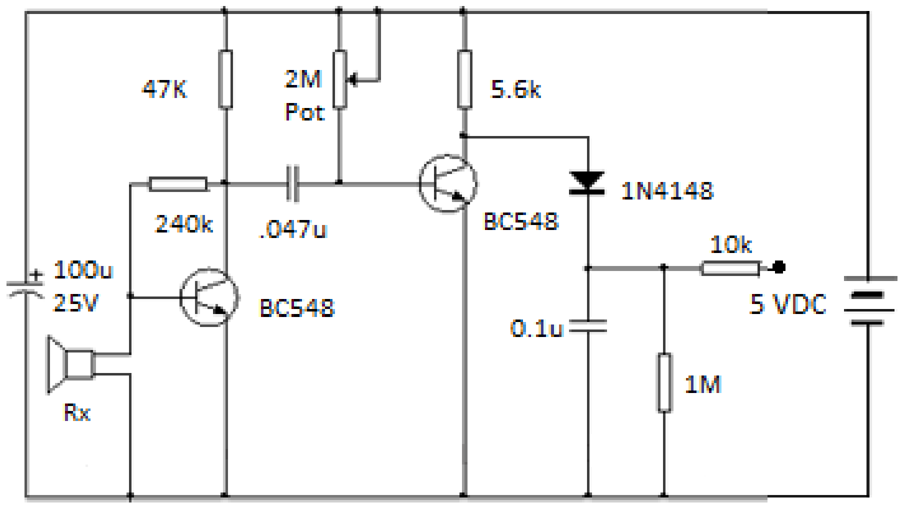


Figure 22: Ultrasonic Receiver Circuit

Operation, Maintenance, and Repair Instructions

Operation

- Items required for operation
  - The Packmule robot
  - An ultrasonic transmitter
  - A mobile phone with the Packmule application
- Turning on the robot
  - To power up the Packmule system, flip the switch located at the rear to the on position.
- Making the robot move
  - Manual Mode
    - This mode of operation is engaged by default. In order to start moving the robot around, you must first have installed the latest version of the Packmule application from the App Store or Google Play Store onto your phone and connect via bluetooth to the device.
    - To start moving the robot around, place your finger on the joystick and drag in the direction you wish to see the robot move.
    - To adjust the speed, press the hamburger icon in the top left of the and enter the speed you wish to have the system follow your as a percentage.
  - Following Mode
    - This mode not enabled by default. In order to engage this mode, first install the app onto your device and connect to the Packmule system. Then, open up the side-menu by tapping on the hamburger icon in the top left. From there, you should see a switch that, when tapped, will engage

following mode.

            - Make sure that the ultrasonic transmitter is powered, so that the Packmule can start following immediately.
            - If for some reason you need to stop the Packmule, hit the big green button in the app. It should now turn red. This indicates that the Packmule is has paused following. To resume following, simply hit the red button
        - Honking the horn
            - To honk the horn, tap the horn icon on the app when it is connected to the Packmule.
- Other key features
    - The Packmule system comes equipped with proximity sensors to avoid obstacles. These are enabled only when the Packmule is following the user.

Maintenance

- Charging the batteries
    - The Packmule system is equipped with two 12V car batteries. To charge them, first disconnect each of the batteries from the robot's base by unclipping the wire harness.
    - Next, connect each battery separately to a 14V power supply. It is important that positive and negative connections do not get flipped.
    - The batteries must charge for several hours and the length of charging should depend on the amount of voltage remaining. A full charge is in the range of 12.6-12.7V, whereas a low charge is around 12.4V.
- Replacing the transmitter's batteries
    - The transmitter is powered by two 9V batteries.
    - To Replace them, unscrew the 4 Phillip's head screws, remove the casing, detach the old batteries and replace them with two new 9V batteries.
    - Do not mix new and old batteries.
- Regular inspection
    - Check bolts and nut to make sure they are all tight.
    - Make sure the tires are inflated and there are no flats
        - A bike pump can be used to inflate the tires.
    - If the system is not used for a while (a few months), be sure to recharge the batteries. Failure to do so may contribute to premature battery failure and will impede the overall performance of system

Repair instructions (Items appear in proper testing order)

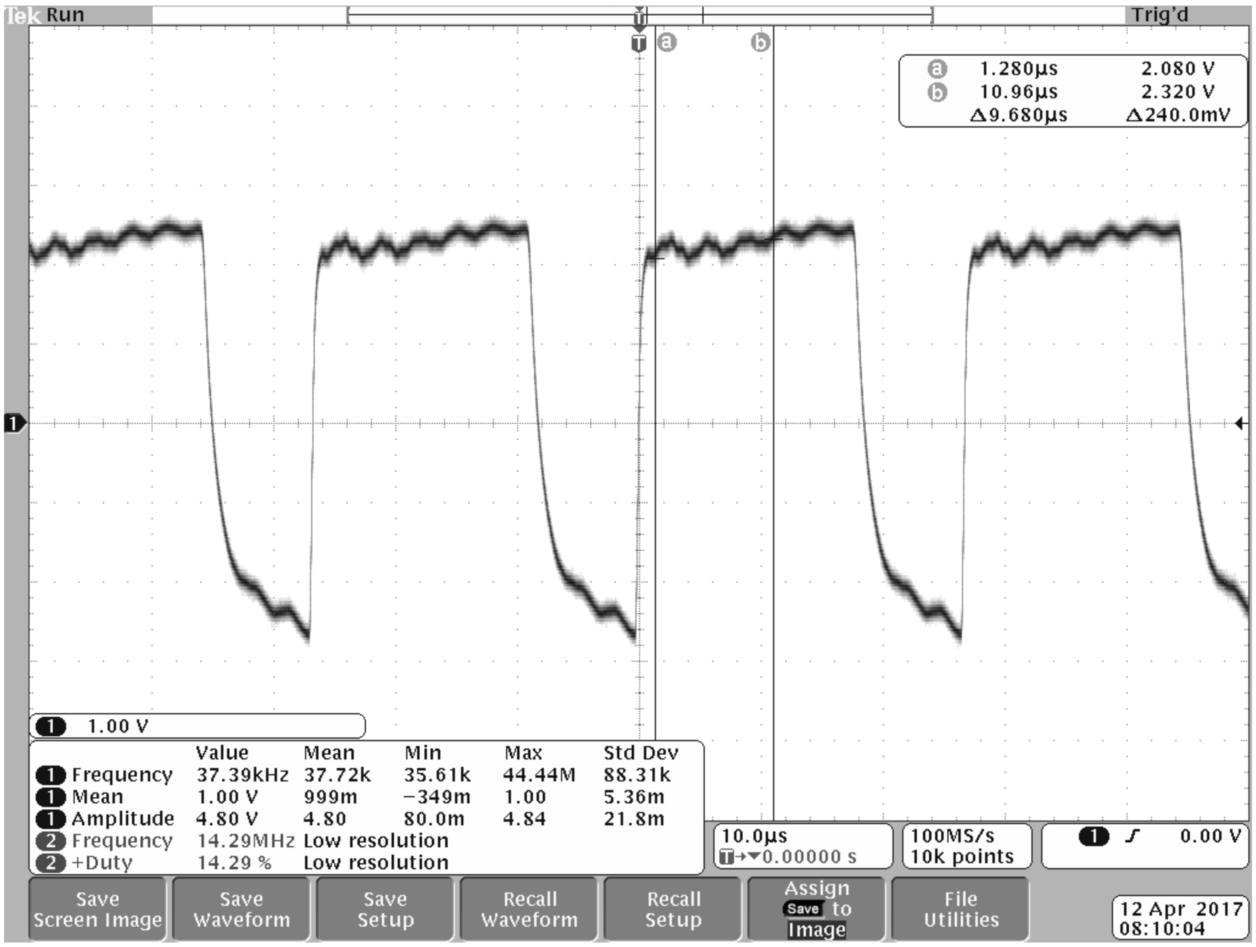
- Issue: Packmule is not following properly
    - Cause: Sensor misalignment
        - Point the transmitter directly at the system when attempting to follow
        - Realign sensors to all face up about 15 degrees

        - Cause: Transmitter not broadcasting
            - Replace the batteries
            - Check all solder joints for continuity
            - Consider replacing the 40kHz transducer module
        - Cause: Receiver is not working properly
            - Ensure the power and ground connections are solid.
            - Check all solder joints for continuity,
            - Consider replacing the 40kHz transducer module
        - Cause: Receiver needs recalibrated
            - Open up the receiver block, identify the trimpot, and hook the sensor up to an oscilloscope using the signal wire output and ground to make your connections
            - If the signal is coming across weak, turn the trimpot clockwise, otherwise turn it counter-clockwise
            - Repeat this for all receivers to ensure a consistent reading
    - Issue: Packmule is running into obstacles
        - Cause: Infrared Sensor is not aimed properly
            - Make sure the IR Sensors are all level and pointed at -40degrees, 0 degrees, and 40 degrees from left to right respectively.
        - Cause: Infrared Sensor is disconnected
            - Make sure all wires are securely fastened on both ends of the IR sensor.

Testing Procedures

Transmitter Circuit

- Ultrasonic signal generated using a 555 timer and 1% resistors and capacitors (Non-ideal as we were over 2.5kHz off our target frequency on average)



```
a   1.280µs    2.080 V
b   10.96µs    2.320 V
    Δ9.680µs   Δ240.0mV
```

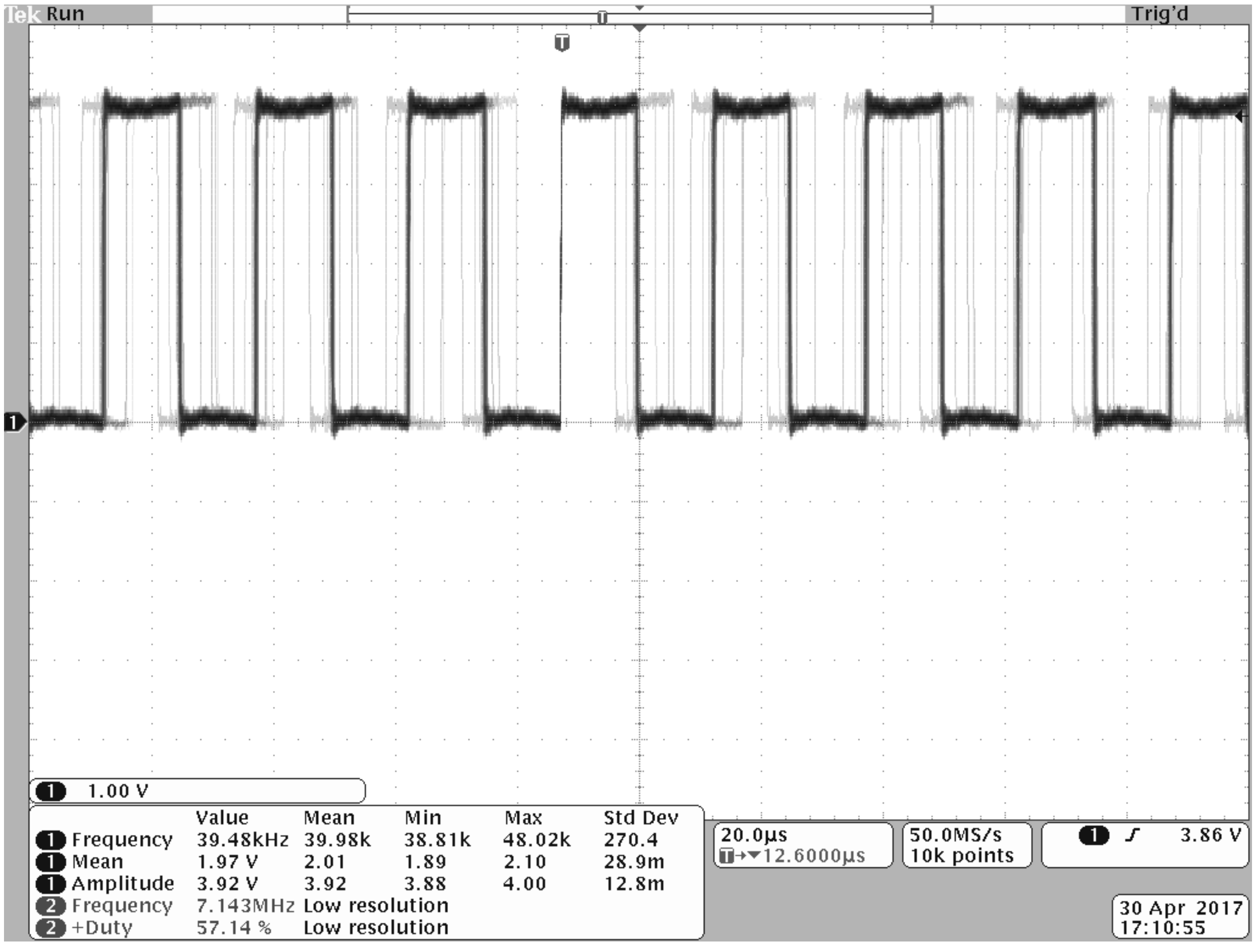
```
1  1.00 V
             Value      Mean            Min      Max      Std Dev
1 Frequency  37.39kHz   37.72k          35.61k   44.44M   88.31k
1 Mean       1.00 V     999m            −349m    1.00     5.36m
1 Amplitude  4.80 V     4.80            80.0m    4.84     21.8m
2 Frequency  14.29MHz   Low resolution
2 +Duty      14.29 %    Low resolution
```

```
10.0µs               100MS/s       1 ∫ 0.00 V
T→▼0.00000 s         10k points
```

```
12 Apr 2017
08:10:04
```

Figure 23: Transmitter output of 555 timer circuit

- ○ Ultrasonic signal generated using the arduino (Ideal as we were less than 500Hz off our target frequency on average)



	Value	Mean	Min	Max	Std Dev
1 Frequency	39.48kHz	39.98k	38.81k	48.02k	270.4
1 Mean	1.97 V	2.01	1.89	2.10	28.9m
1 Amplitude	3.92 V	3.92	3.88	4.00	12.8m
2 Frequency	7.143MHz	Low resolution			
2 +Duty	57.14 %	Low resolution			

Figure 24: Transmitter output of Arduino circuit

Infrared Proximity Sensors

- ○ The Infrared proximity sensors were tested by putting various obstacles of different heights and colors in front of the Packmule. Each gave a slightly different reading, but a threshold of 2V seemed to be the typical reading the system would yield when approaching an obstacle.
- ○ This was more of a trial and error form of testing as lighting plays a huge part in the sensitivity of IR sensors. In order to overcome this, all the IR sensors are hidden beneath the base of the system to provide a more consistent environment.

Receiver circuit

- ○ Throughout the process of developing the Packmule system, the receiver circuits were rigorously tested and calibrated to ensure proper following.
- ○ Each of the circuits was hooked up to the analog pin an Arduino and was tested at a

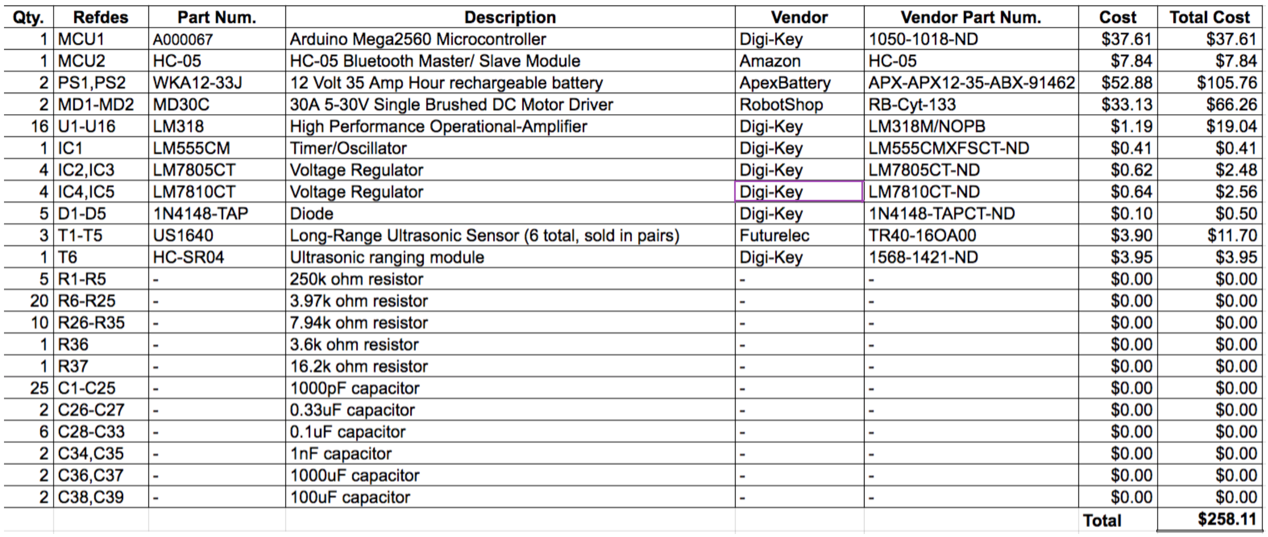
distance of a meter from the transmitter in line of sight. Each transmitter was adjusted using an onboard trimpot that was installed just for this purpose.

After proper calibration, each sensor would output a value of 900/1023 which translates to about 4.4V at a meter away from the signal source.

Motor Controller

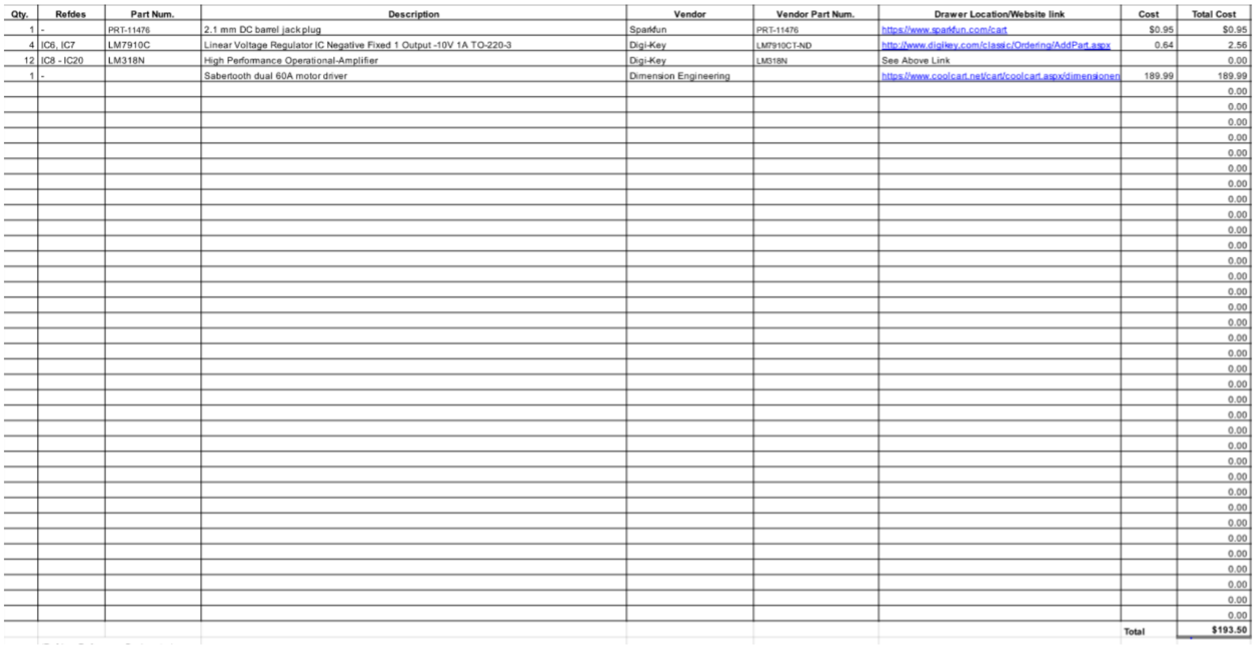
- Throughout the process of developing the Packmule system, the motor controller was rigorously tested.
- From the beginning, the motors were known to be good and well capable of handling a small 30lb load in addition to the weight of the frame. However, the same could not be said of the motor controller.
- Several tests were done, none of which could really be numerically computed as they were so fast. At first, there was no guarantee that the motor controllers would be able to handle precise movement. In order to test this, the robot was subjected to traversing the narrow halls of ASEC.
- In order to test the speed and responsiveness of the system, the robot would be controlled at 100% power forward and then would go into full reverse. After passing these tests, several incline and offroad tests were performed with weighted loads.

Parts List



Qty.	Refdes	Part Num.	Description	Vendor	Vendor Part Num.	Cost	Total Cost
1	MCU1	A000067	Arduino Mega2560 Microcontroller	Digi-Key	1050-1018-ND	$37.61	$37.61
1	MCU2	HC-05	HC-05 Bluetooth Master/ Slave Module	Amazon	HC-05	$7.84	$7.84
2	PS1,PS2	WKA12-33J	12 Volt 35 Amp Hour rechargeable battery	ApexBattery	APX-APX12-35-ABX-91462	$52.88	$105.76
2	MD1-MD2	MD30C	30A 5-30V Single Brushed DC Motor Driver	RobotShop	RB-Cyt-133	$33.13	$66.26
16	U1-U16	LM318	High Performance Operational-Amplifier	Digi-Key	LM318M/NOPB	$1.19	$19.04
1	IC1	LM555CM	Timer/Oscillator	Digi-Key	LM555CMXFSCT-ND	$0.41	$0.41
4	IC2,IC3	LM7805CT	Voltage Regulator	Digi-Key	LM7805CT-ND	$0.62	$2.48
4	IC4,IC5	LM7810CT	Voltage Regulator	Digi-Key	LM7810CT-ND	$0.64	$2.56
5	D1-D5	1N4148-TAP	Diode	Digi-Key	1N4148-TAPCT-ND	$0.10	$0.50
3	T1-T5	US1640	Long-Range Ultrasonic Sensor (6 total, sold in pairs)	Futurelec	TR40-16OA00	$3.90	$11.70
1	T6	HC-SR04	Ultrasonic ranging module	Digi-Key	1568-1421-ND	$3.95	$3.95
5	R1-R5	-	250k ohm resistor	-	-	$0.00	$0.00
20	R6-R25	-	3.97k ohm resistor	-	-	$0.00	$0.00
10	R26-R35	-	7.94k ohm resistor	-	-	$0.00	$0.00
1	R36	-	3.6k ohm resistor	-	-	$0.00	$0.00
1	R37	-	16.2k ohm resistor	-	-	$0.00	$0.00
25	C1-C25	-	1000pF capacitor	-	-	$0.00	$0.00
2	C26-C27	-	0.33uF capacitor	-	-	$0.00	$0.00
6	C28-C33	-	0.1uF capacitor	-	-	$0.00	$0.00
2	C34,C35	-	1nF capacitor	-	-	$0.00	$0.00
2	C36,C37	-	1000uF capacitor	-	-	$0.00	$0.00
2	C38,C39	-	100uF capacitor	-	-	$0.00	$0.00
						Total	**$258.11**

Table 21: Parts List 1

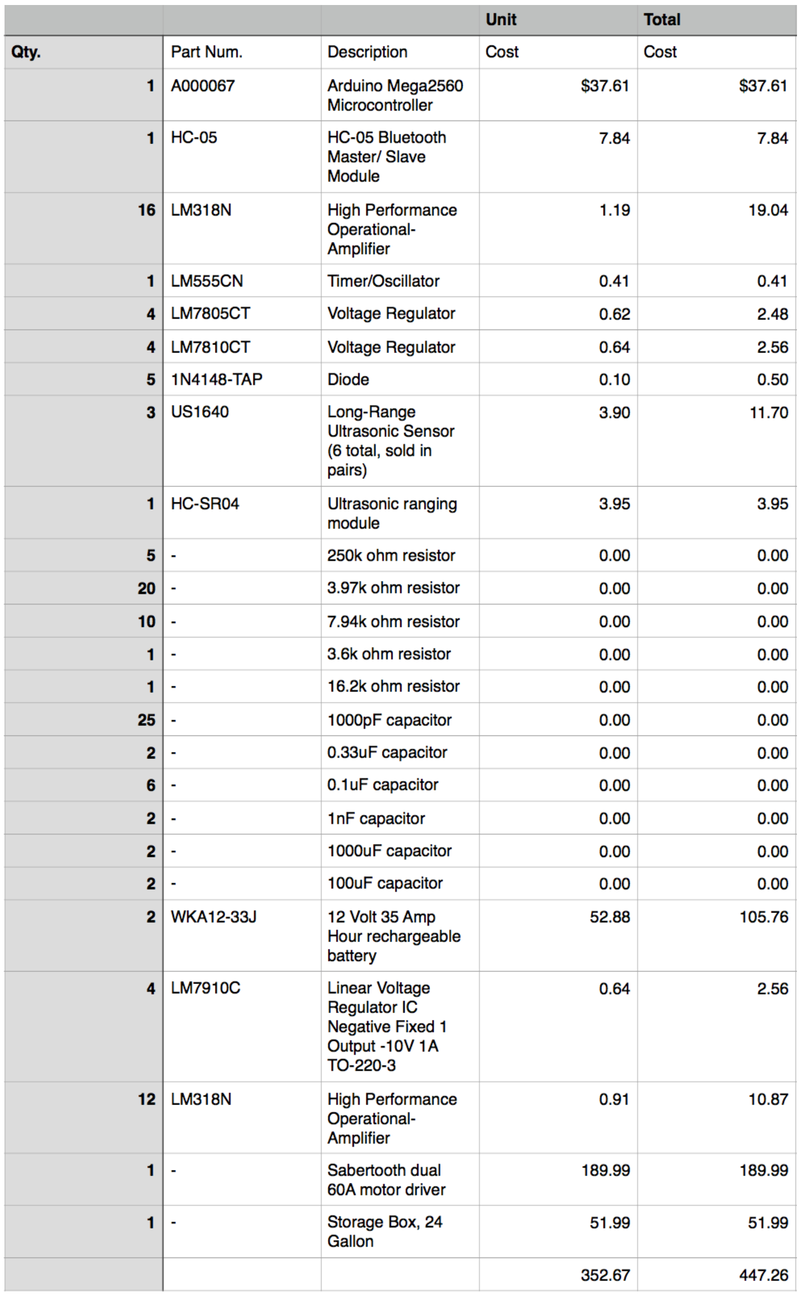


Qty.	Refdes	Part Num.	Description	Vendor	Vendor Part Num.	Drawer Location/Website link	Cost	Total Cost
1	-	PRT-11476	2.1 mm DC barrel jack plug	Sparkfun	PRT-11476	https://www.sparkfun.com/cart	$0.95	$0.95
4	IC6, IC7	LM7910C	Linear Voltage Regulator IC Negative Fixed 1 Output -10V 1A TO-220-3	Digi-Key	LM7910CT-ND	http://www.digikey.com/classic/Ordering/AddPart.aspx	0.64	2.56
12	IC8 - IC20	LM318N	High Performance Operational-Amplifier	Digi-Key	LM318N	See Above Link		0.00
1	-		Sabertooth dual 60A motor driver	Dimension Engineering		https://www.coolcart.net/cart/coolcart.aspx/dimensionen	189.99	189.99
								0.00
								0.00
								0.00
								0.00
								0.00
								0.00
								0.00
								0.00
								0.00
								0.00
								0.00
								0.00
								0.00
								0.00
								0.00
								0.00
								0.00
								0.00
								0.00
								0.00
								0.00
								0.00
								0.00
								0.00
								0.00
								0.00
								0.00
								0.00
								0.00
								0.00
								0.00
								0.00
								0.00
								0.00
								0.00
							Total	$193.50

Table 22: Parts List 2

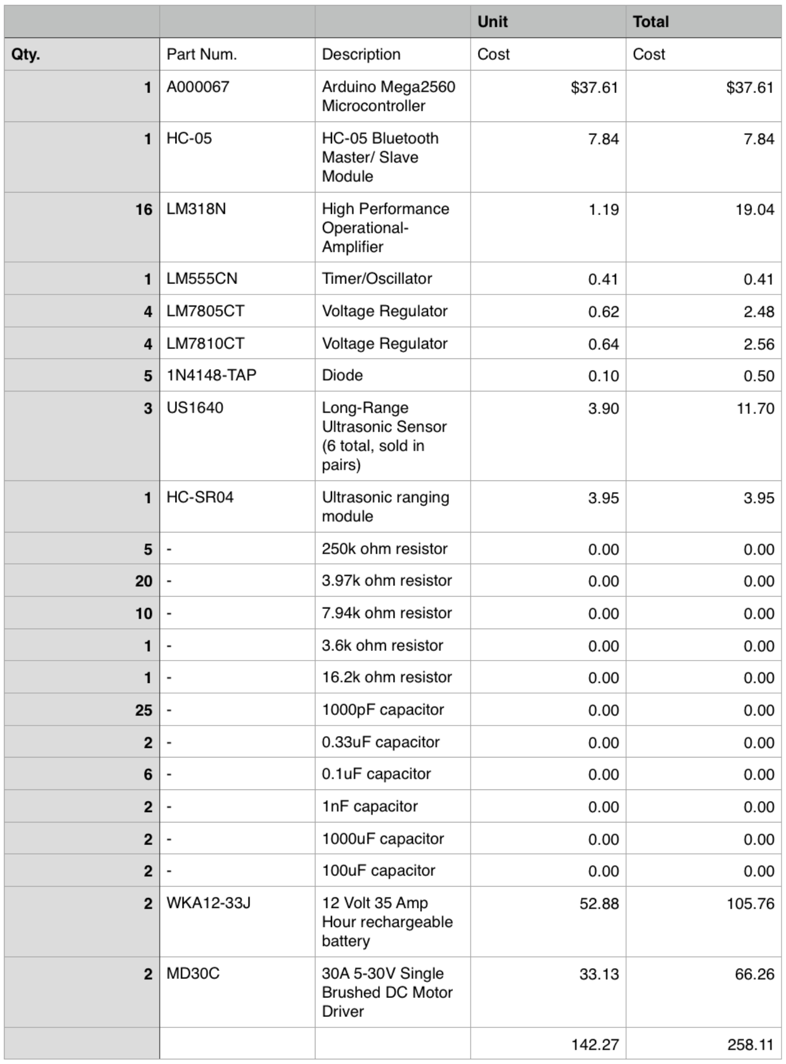
Parts list 1 was the order placed before winter break. This list includes the materials for the initial design of the Packmule system. Parts List 2 Design Team 02's was allocated $400 for the completion of this project and received an additional $100 from corporate sponsorship. With the total cost of this project coming to $385.35, the design stayed within budget.

Financial Budget



			Unit	Total
Qty.	Part Num.	Description	Cost	Cost
1	A000067	Arduino Mega2560 Microcontroller	$37.61	$37.61
1	HC-05	HC-05 Bluetooth Master/ Slave Module	7.84	7.84
16	LM318N	High Performance Operational-Amplifier	1.19	19.04
1	LM555CN	Timer/Oscillator	0.41	0.41
4	LM7805CT	Voltage Regulator	0.62	2.48
4	LM7810CT	Voltage Regulator	0.64	2.56
5	1N4148-TAP	Diode	0.10	0.50
3	US1640	Long-Range Ultrasonic Sensor (6 total, sold in pairs)	3.90	11.70
1	HC-SR04	Ultrasonic ranging module	3.95	3.95
5	-	250k ohm resistor	0.00	0.00
20	-	3.97k ohm resistor	0.00	0.00
10	-	7.94k ohm resistor	0.00	0.00
1	-	3.6k ohm resistor	0.00	0.00
1	-	16.2k ohm resistor	0.00	0.00
25	-	1000pF capacitor	0.00	0.00
2	-	0.33uF capacitor	0.00	0.00
6	-	0.1uF capacitor	0.00	0.00
2	-	1nF capacitor	0.00	0.00
2	-	1000uF capacitor	0.00	0.00
2	-	100uF capacitor	0.00	0.00
2	WKA12-33J	12 Volt 35 Amp Hour rechargeable battery	52.88	105.76
4	LM7910C	Linear Voltage Regulator IC Negative Fixed 1 Output -10V 1A TO-220-3	0.64	2.56
12	LM318N	High Performance Operational-Amplifier	0.91	10.87
1	-	Sabertooth dual 60A motor driver	189.99	189.99
1	-	Storage Box, 24 Gallon	51.99	51.99
			352.67	447.26

Table 23: Actual Budget

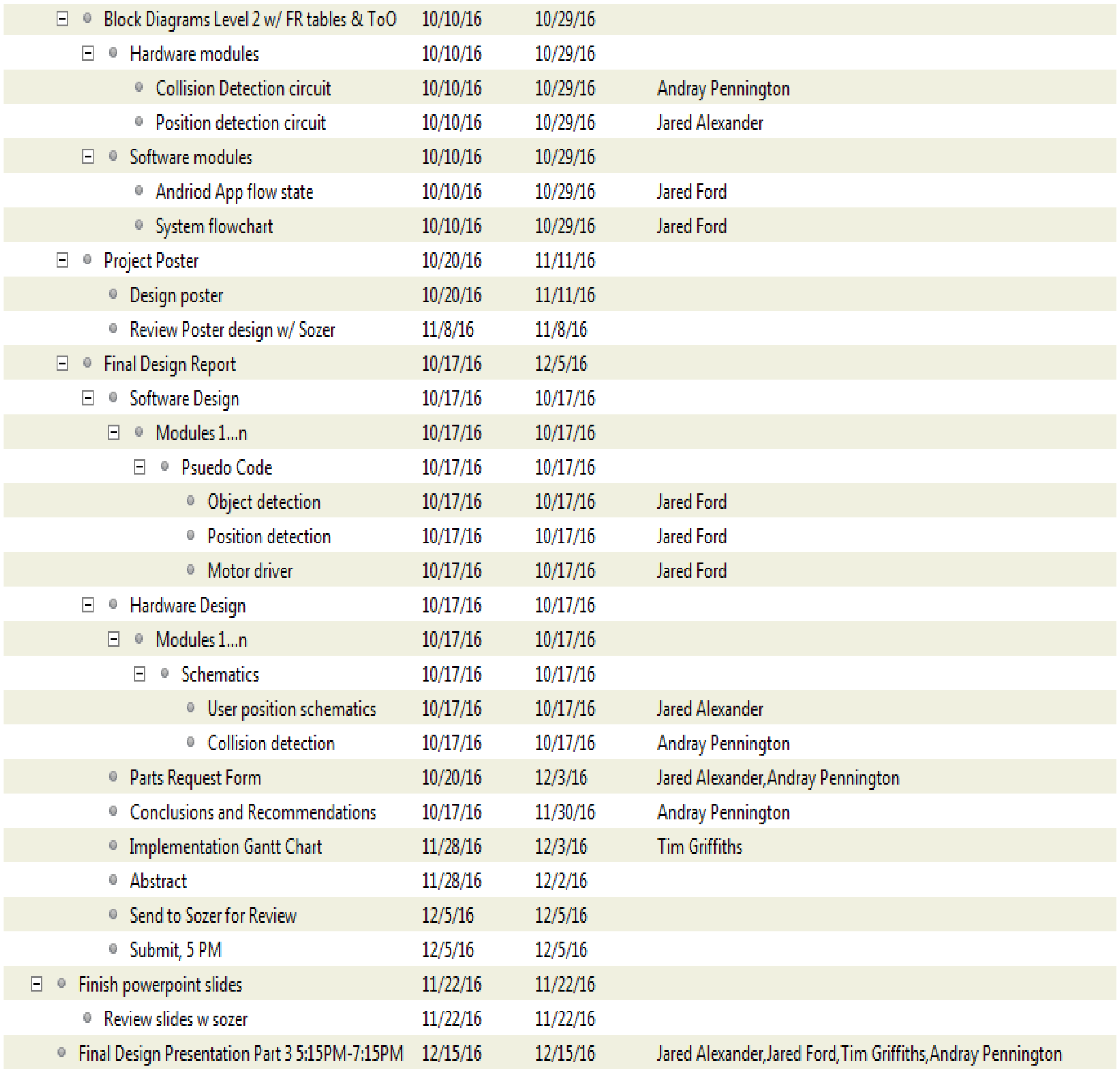


			Unit	Total
Qty.	Part Num.	Description	Cost	Cost
1	A000067	Arduino Mega2560 Microcontroller	$37.61	$37.61
1	HC-05	HC-05 Bluetooth Master/ Slave Module	7.84	7.84
16	LM318N	High Performance Operational-Amplifier	1.19	19.04
1	LM555CN	Timer/Oscillator	0.41	0.41
4	LM7805CT	Voltage Regulator	0.62	2.48
4	LM7810CT	Voltage Regulator	0.64	2.56
5	1N4148-TAP	Diode	0.10	0.50
3	US1640	Long-Range Ultrasonic Sensor (6 total, sold in pairs)	3.90	11.70
1	HC-SR04	Ultrasonic ranging module	3.95	3.95
5	-	250k ohm resistor	0.00	0.00
20	-	3.97k ohm resistor	0.00	0.00
10	-	7.94k ohm resistor	0.00	0.00
1	-	3.6k ohm resistor	0.00	0.00
1	-	16.2k ohm resistor	0.00	0.00
25	-	1000pF capacitor	0.00	0.00
2	-	0.33uF capacitor	0.00	0.00
6	-	0.1uF capacitor	0.00	0.00
2	-	1nF capacitor	0.00	0.00
2	-	1000uF capacitor	0.00	0.00
2	-	100uF capacitor	0.00	0.00
2	WKA12-33J	12 Volt 35 Amp Hour rechargeable battery	52.88	105.76
2	MD30C	30A 5-30V Single Brushed DC Motor Driver	33.13	66.26
			142.27	258.11

Table 24: Original Budget

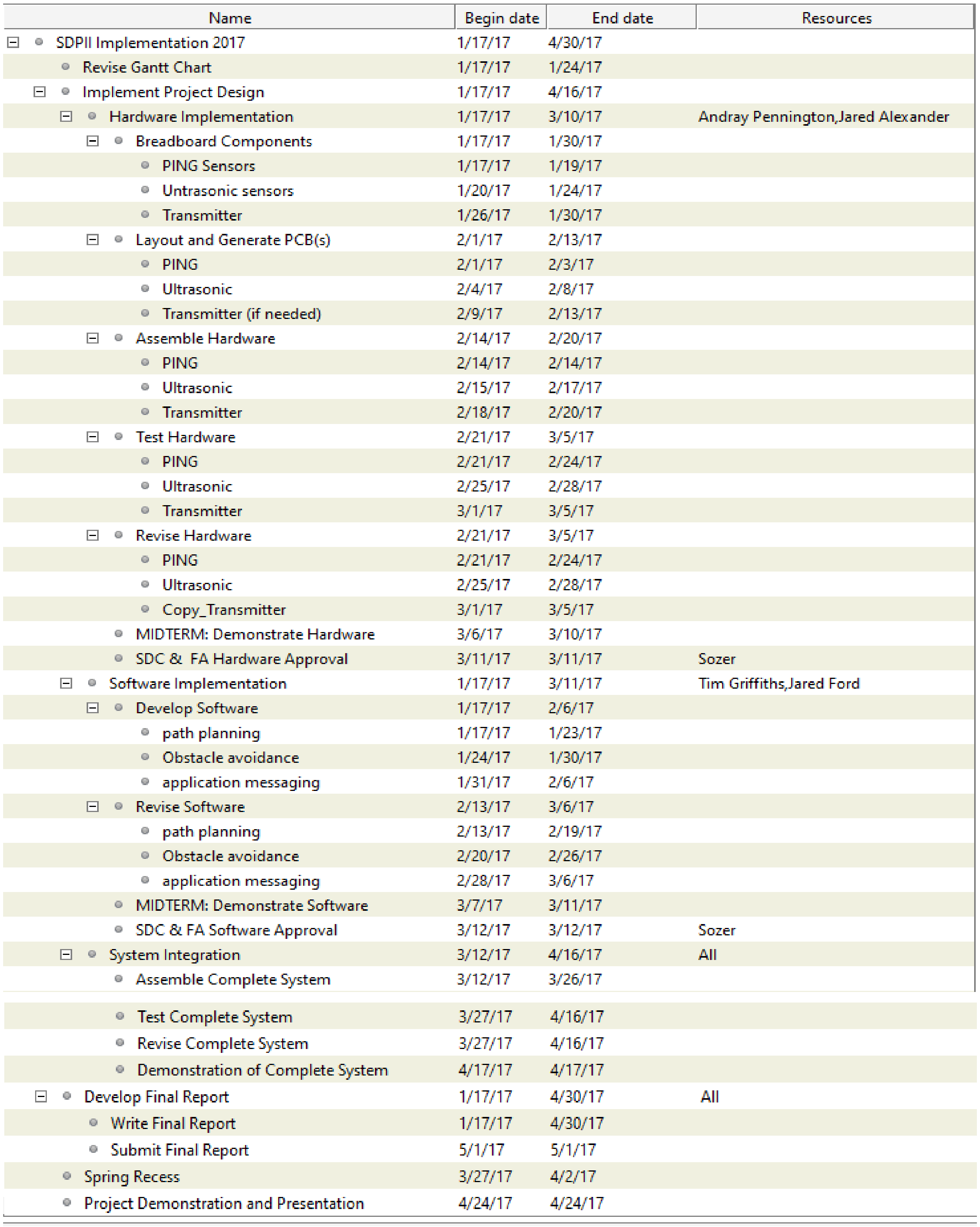
Design Team 02 was allocated $400 for the completion of this project and received an additional $100 from corporate sponsorship. The Actual budget (Table 23) showed a final total of $447.26, and the original budget (Table 24) showed a final total of $258.11. The difference between the two budgets is $189.15. This is largely due to the “upgraded” motor driver and the original budget not taking into account a storage container.

Gantt Charts



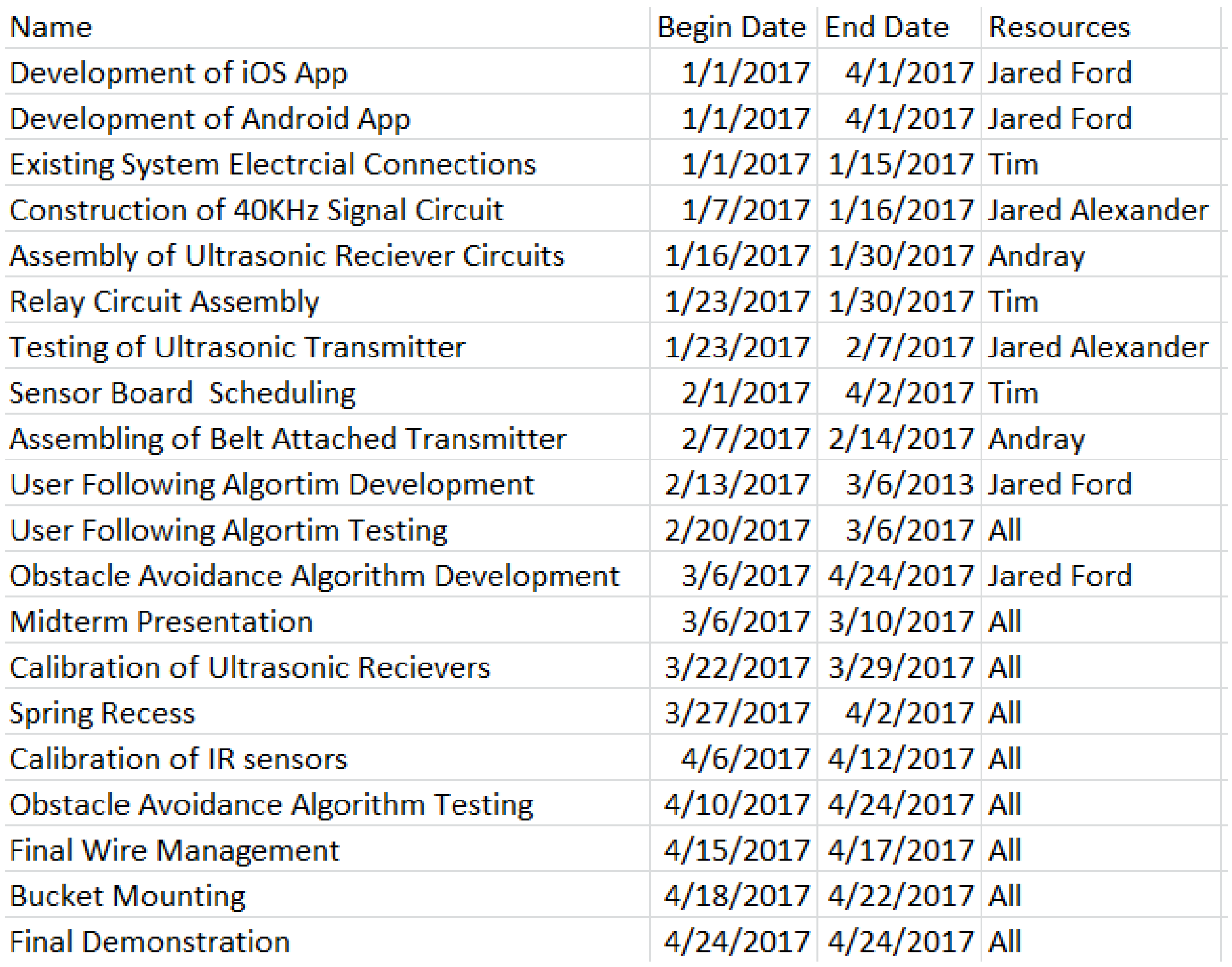
Block Diagrams Level 2 w/ FR tables & ToO	10/10/16	10/29/16	
Hardware modules	10/10/16	10/29/16	
Collision Detection circuit	10/10/16	10/29/16	Andray Pennington
Position detection circuit	10/10/16	10/29/16	Jared Alexander
Software modules	10/10/16	10/29/16	
Andriod App flow state	10/10/16	10/29/16	Jared Ford
System flowchart	10/10/16	10/29/16	Jared Ford
Project Poster	10/20/16	11/11/16	
Design poster	10/20/16	11/11/16	
Review Poster design w/ Sozer	11/8/16	11/8/16	
Final Design Report	10/17/16	12/5/16	
Software Design	10/17/16	10/17/16	
Modules 1...n	10/17/16	10/17/16	
Psuedo Code	10/17/16	10/17/16	
Object detection	10/17/16	10/17/16	Jared Ford
Position detection	10/17/16	10/17/16	Jared Ford
Motor driver	10/17/16	10/17/16	Jared Ford
Hardware Design	10/17/16	10/17/16	
Modules 1...n	10/17/16	10/17/16	
Schematics	10/17/16	10/17/16	
User position schematics	10/17/16	10/17/16	Jared Alexander
Collision detection	10/17/16	10/17/16	Andray Pennington
Parts Request Form	10/20/16	12/3/16	Jared Alexander,Andray Pennington
Conclusions and Recommendations	10/17/16	11/30/16	Andray Pennington
Implementation Gantt Chart	11/28/16	12/3/16	Tim Griffiths
Abstract	11/28/16	12/2/16	
Send to Sozer for Review	12/5/16	12/5/16	
Submit, 5 PM	12/5/16	12/5/16	
Finish powerpoint slides	11/22/16	11/22/16	
Review slides w sozer	11/22/16	11/22/16	
Final Design Presentation Part 3 5:15PM-7:15PM	12/15/16	12/15/16	Jared Alexander,Jared Ford,Tim Griffiths,Andray Pennington

Table 25: Midterm Gantt Chart



Name	Begin date	End date	Resources
SDPII Implementation 2017	1/17/17	4/30/17	
Revise Gantt Chart	1/17/17	1/24/17	
Implement Project Design	1/17/17	4/16/17	
Hardware Implementation	1/17/17	3/10/17	Andray Pennington,Jared Alexander
Breadboard Components	1/17/17	1/30/17	
PING Sensors	1/17/17	1/19/17	
Untrasonic sensors	1/20/17	1/24/17	
Transmitter	1/26/17	1/30/17	
Layout and Generate PCB(s)	2/1/17	2/13/17	
PING	2/1/17	2/3/17	
Ultrasonic	2/4/17	2/8/17	
Transmitter (if needed)	2/9/17	2/13/17	
Assemble Hardware	2/14/17	2/20/17	
PING	2/14/17	2/14/17	
Ultrasonic	2/15/17	2/17/17	
Transmitter	2/18/17	2/20/17	
Test Hardware	2/21/17	3/5/17	
PING	2/21/17	2/24/17	
Ultrasonic	2/25/17	2/28/17	
Transmitter	3/1/17	3/5/17	
Revise Hardware	2/21/17	3/5/17	
PING	2/21/17	2/24/17	
Ultrasonic	2/25/17	2/28/17	
Copy_Transmitter	3/1/17	3/5/17	
MIDTERM: Demonstrate Hardware	3/6/17	3/10/17	
SDC & FA Hardware Approval	3/11/17	3/11/17	Sozer
Software Implementation	1/17/17	3/11/17	Tim Griffiths,Jared Ford
Develop Software	1/17/17	2/6/17	
path planning	1/17/17	1/23/17	
Obstacle avoidance	1/24/17	1/30/17	
application messaging	1/31/17	2/6/17	
Revise Software	2/13/17	3/6/17	
path planning	2/13/17	2/19/17	
Obstacle avoidance	2/20/17	2/26/17	
application messaging	2/28/17	3/6/17	
MIDTERM: Demonstrate Software	3/7/17	3/11/17	
SDC & FA Software Approval	3/12/17	3/12/17	Sozer
System Integration	3/12/17	4/16/17	All
Assemble Complete System	3/12/17	3/26/17	
Test Complete System	3/27/17	4/16/17	
Revise Complete System	3/27/17	4/16/17	
Demonstration of Complete System	4/17/17	4/17/17	
Develop Final Report	1/17/17	4/30/17	All
Write Final Report	1/17/17	4/30/17	
Submit Final Report	5/1/17	5/1/17	
Spring Recess	3/27/17	4/2/17	
Project Demonstration and Presentation	4/24/17	4/24/17	

Table 26: Implementation Gantt Chart



Name	Begin Date	End Date	Resources
Development of iOS App	1/1/2017	4/1/2017	Jared Ford
Development of Android App	1/1/2017	4/1/2017	Jared Ford
Existing System Electrcial Connections	1/1/2017	1/15/2017	Tim
Construction of 40KHz Signal Circuit	1/7/2017	1/16/2017	Jared Alexander
Assembly of Ultrasonic Reciever Circuits	1/16/2017	1/30/2017	Andray
Relay Circuit Assembly	1/23/2017	1/30/2017	Tim
Testing of Ultrasonic Transmitter	1/23/2017	2/7/2017	Jared Alexander
Sensor Board Scheduling	2/1/2017	4/2/2017	Tim
Assembling of Belt Attached Transmitter	2/7/2017	2/14/2017	Andray
User Following Algortim Development	2/13/2017	3/6/2013	Jared Ford
User Following Algortim Testing	2/20/2017	3/6/2017	All
Obstacle Avoidance Algorithm Development	3/6/2017	4/24/2017	Jared Ford
Midterm Presentation	3/6/2017	3/10/2017	All
Calibration of Ultrasonic Recievers	3/22/2017	3/29/2017	All
Spring Recess	3/27/2017	4/2/2017	All
Calibration of IR sensors	4/6/2017	4/12/2017	All
Obstacle Avoidance Algorithm Testing	4/10/2017	4/24/2017	All
Final Wire Management	4/15/2017	4/17/2017	All
Bucket Mounting	4/18/2017	4/22/2017	All
Final Demonstration	4/24/2017	4/24/2017	All

Table 27: Actual Gantt Chart

While the goal of a project is to stick to the predetermined schedule, unforeseen problems can result in a change of plans. During some of the ultrasonic and PING sensor tesing in early march, a system incompatibility was found. This resulted in the team being forced to go back and change the design to function with infrared sensors instead. As a result of this unforeseen technology switch, the actual implementation was pushed back more than intended. Additionally, group roles were abandoned in hopes that an all hands on deck approach would allow the project to be completed on time. Fortunately, this approach allowed the team to stay on the same page as the project was completed.

Design Team Information

The following list consists of the design team and their role for the project:

- Jared Alexander, Hardware Manager, EE
- Jared Ford, Software Manager, CpE
- Tim Griffiths, Team Leader, EE
- Andray Pennington, Archivist, EE

Conclusions and Recommendations

The Packmule will be able to fulfill its role of simplifying the transport of moderate loads over a distance. Further development of this project would lead to more models of the Packmule, which would allow the robot to carry heavier loads or operate at faster speeds. This project provided an introduction to the concepts of leader follower robots and autonomous obstacle avoiding navigation, both of which are being implemented more and more in today's integrated society. The level of research and design in this project was suitable as a senior capstone project. For those interested in taking an in depth look at the source code for this project, it is located on GitHub at https://github.com/jrodbossman.

The completed project met all required specifications and went as far as exceeding many of them. A few expectations that were exceeded included making, both android and iOS applications, making the design aesthetically pleasing, having the ability to travel well above a standard walking pace, etc. All in all, the project can be considered a success. The Packmule system is very reliable and can go days on a single charge. The limiting factor may be the battery life of the phone controlling the system.

The team dynamics were as follows:

- Jared Alexander
  - Responsible primarily for hardware implementation. Jared played an important role in getting the system wired up, drawing schematics, troubleshooting the hardware, and coming up with new and unique ways to solve problems.
- Jared Ford
  - Responsible primarily for software implantation. Jared was a key player in getting all the code for the Arduino, Android and iOS applications running. He was also responsible for bridging the gap between the hardware and software interfaces. Jared was a driving force in the implementation of the Packmule system. In his efforts, Jared was able to implement all the software routines rapidly enough to come and help out with the hardware design and troubleshooting.
- Tim Griffiths
  - Tim was the team leader for this project, but that was not his only role. In fact, Tim was responsible for ensuring that the motors had proper signals and power being fed to them. Additionally, he was a responsible for dissecting original circuitry from the wheelchair and, from that dissection, proposed ways to integrate the existing hardware in this new design. Tim also made sure to keep everyone updated with regular meetings and deadline updates.

- Andray Pennington
    - Andray was also primarily responsible for hardware, but he also helped keep the group on task and organized. He had the critical role of archivist and made sure everything from the design to the end implementation was well documented. Andray worked closely with Jared Alexander to get the transmitter/ receiver circuits up and running which was one of the most crucial aspects of the project.

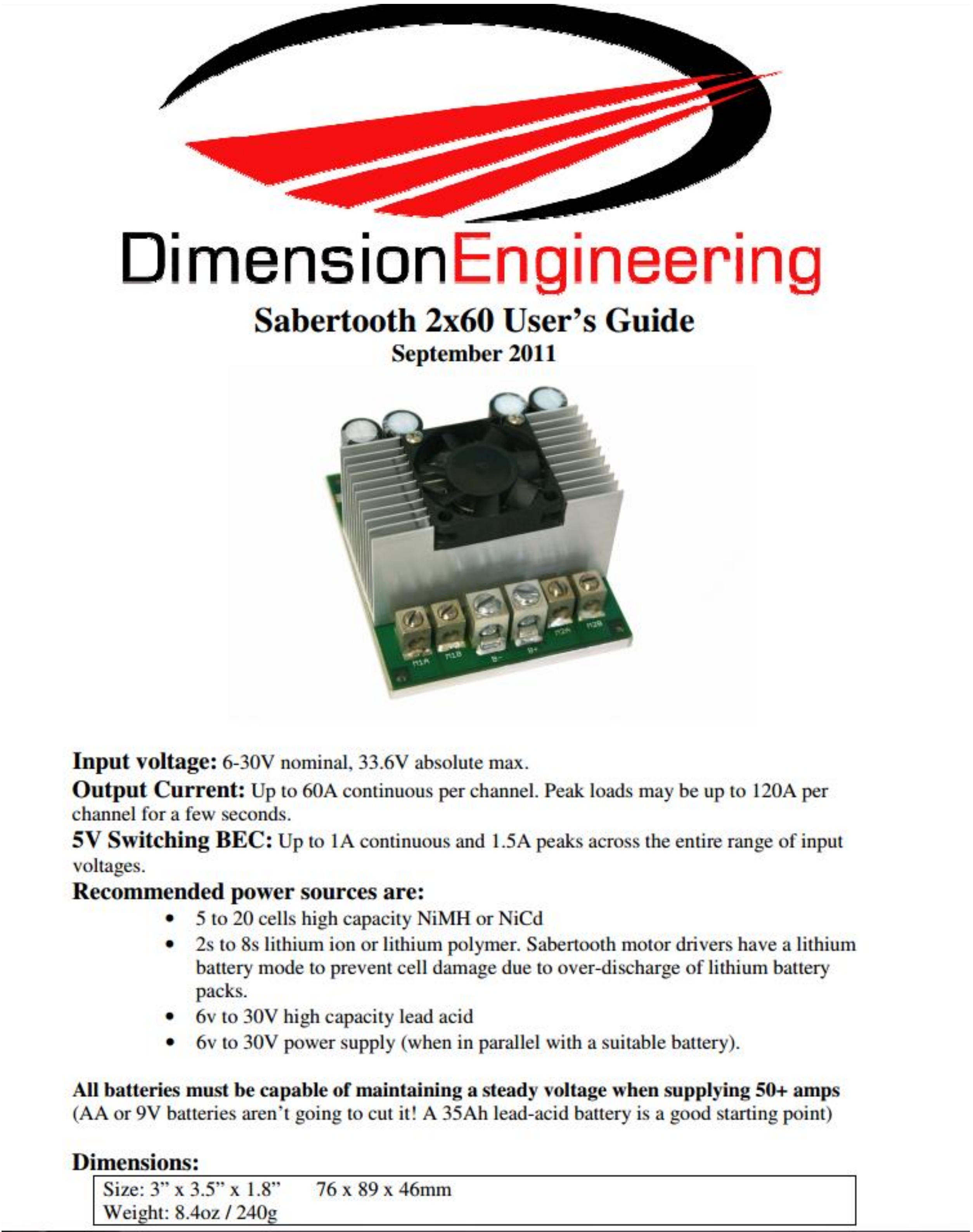
There are a few recommendations for future students who may wish to tackle a similar that should be considered. One of the most important things to consider is how much you are paying for sensors. Make sure you don’t get cheap sensors. In this case you get what you pay for. In the design of Packmule, we used many varieties of ultrasonic transducers and can safely say that the more expensive and more reputable brands performed significantly better. Additionally, one should also consider doing rigorous schematic simulation before deciding on a part to order. As a rule, simulations should be taken for what they’re worth, but oftentimes there are unforeseen factors that come into play. Make sure all noise, signals, and constraints are taken into account when doing system simulation.

References

[1] Cenek, Peter D., Neil J. Jamieson, and Maurice WMcLarin. "Frictional Characteristics of Roadside Grass Types." (n.d.): n. pag. Opus Internation Consultants. Web. 25 Aug. 2016.
[2] H. Zumbahlen, Sallen-Key Filters, 1st ed. Norwood, MA: Analog Devices, Inc., 2016. <http://www.analog.com/media/en/training-seminars/tutorials/MT-222.pdf>
[3] "The 555 timer", Electronics.dit.ie, 2016. Web. 12 Nov. 2016. <http://www.electronics.dit.ie/staff/mtully/555%20folder/555%20timer.htm>
[4] Wi-Fi Location-Based Services 4.1 Design Guide. Publication no. OL-11612-01. Cisco Systems, Inc., 20 May 2008. Web. 10 Oct. 2016. <http://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Mobility/WiFiLBS-DG/wifich2.html>.
[5] "Bluetooth Technology Website". Bluetooth.com. N.p., 2016. Web. 7 Nov. 2016. <https://www.bluetooth.com/>
[6] W. J. Steyn and J. Warnich, “COMPARISON OF TYRE ROLLING RESISTANCE FOR DIFFERENT MOUNTAIN BIKE TYRE DIAMETERS AND SURFACE CONDITIONS,” South African Journal for Research in Sport, Physical Education and Recreation, pp. 179–193, 2014.
[7] "Sharp Infrared Ranger Comparison." Sharp Infrared Ranger Comparison | Acroname. N.p., n.d. Web. 29 Apr. 2017.

Appendices

Part Number	**Description**	**Data Sheet**
MD30C	Cytron 30A Motor Driver Shield	http://www.robotshop.com/media/files/images3/md30cusersmanual_1_.pdf
HM-10	Bluetooth to Serial Port Module	ftp://imall.iteadstudio.com/Modules/IM130614001_Serial_Port_BLE_Module_Master_Slave_HM-10/DS_IM130614001_Serial_Port_BLE_Module_Master_Slave_HM-10.pdf
Arduino Mega 2560	Microcontroller with Atmega128 microchip	https://www.arduino.cc/en/Main/ArduinoBoardMega2560
HC-SR04	Ping Sensor	https://www.sparkfun.com/products/13959
US1640	Long range ultrasonic sensor, 40 Khz	http://www.futurlec.com/Ultrasonic_Sensors.shtml
LM555CM	555 timer/ oscillator	https://www.fairchildsemi.com/datasheets/LM/LM555.pdf
LM7805CT	Voltage Regulator IC	https://www.fairchildsemi.com/datasheets/LM/LM7805.pdf
1N4148-TAP	Diode	http://www.vishay.com/docs/81857/1n4148.pdf


DimensionEngineering

Sabertooth 2x60 User's Guide

September 2011

Input voltage: 6-30V nominal, 33.6V absolute max.

Output Current: Up to 60A continuous per channel. Peak loads may be up to 120A per channel for a few seconds.

5V Switching BEC: Up to 1A continuous and 1.5A peaks across the entire range of input voltages.

Recommended power sources are:

- 5 to 20 cells high capacity NiMH or NiCd
- 2s to 8s lithium ion or lithium polymer. Sabertooth motor drivers have a lithium battery mode to prevent cell damage due to over-discharge of lithium battery packs.
- 6v to 30V high capacity lead acid
- 6v to 30V power supply (when in parallel with a suitable battery).

All batteries must be capable of maintaining a steady voltage when supplying 50+ amps
(AA or 9V batteries aren't going to cut it! A 35Ah lead-acid battery is a good starting point)

Dimensions:

Size: 3" x 3.5" x 1.8"	76 x 89 x 46mm
Weight: 8.4oz / 240g	

Figure 25: Sabertooth 2x60 Motor Driver Datasheet Overview

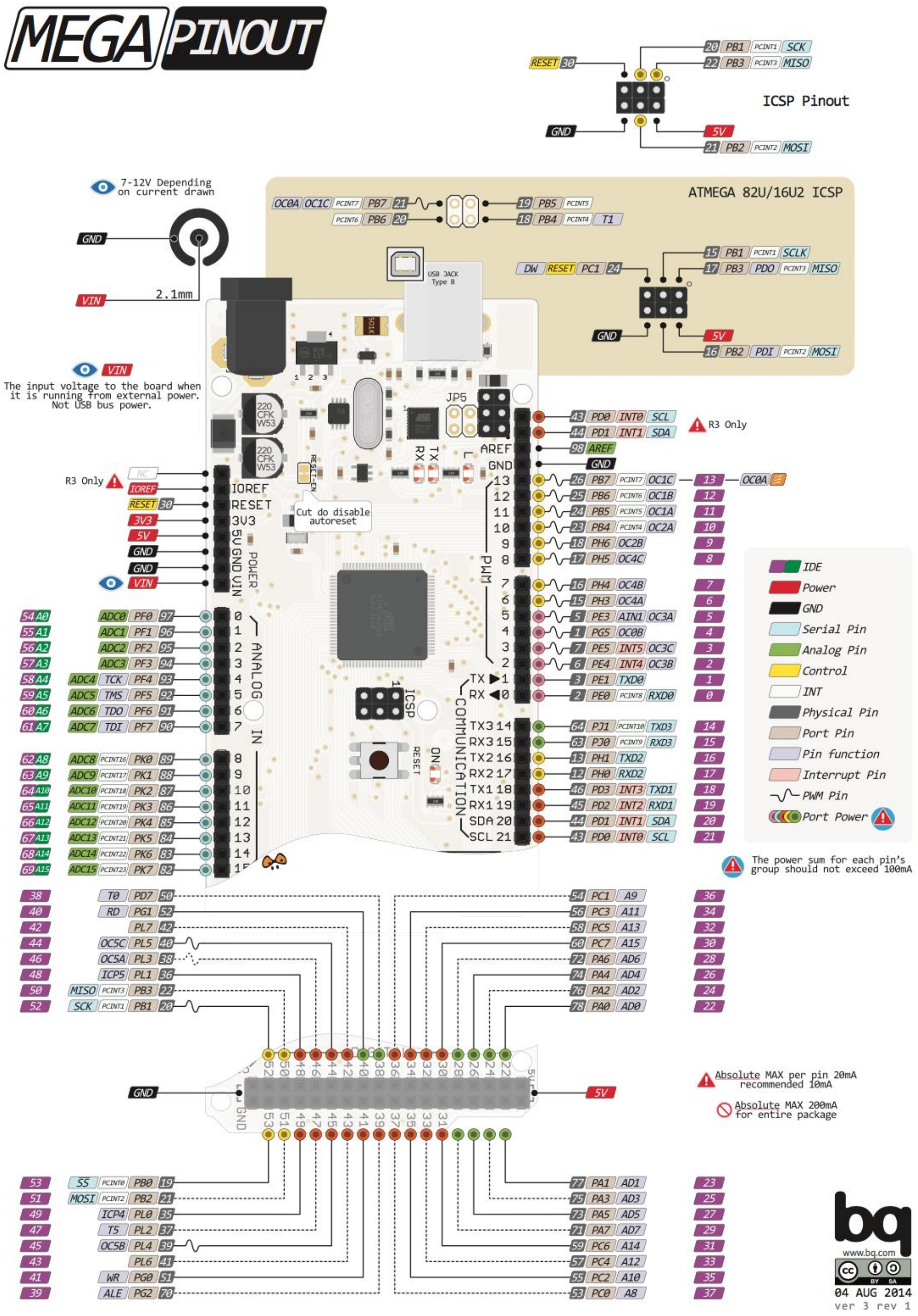
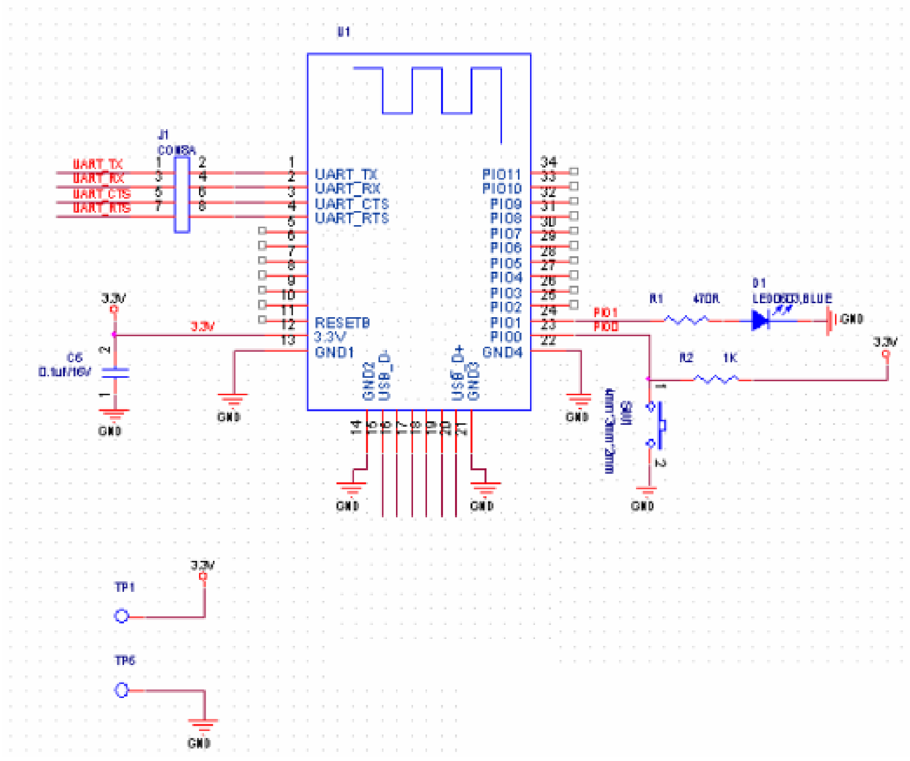

MEGA PINOUT
ICSP Pinout
ATMEGA 82U/16U2 ICSP
7-12V Depending on current drawn
2.1mm
The input voltage to the board when it is running from external power. Not USB bus power.
R3 Only
Cut do disable autoreset
IDE
Power
GND
Serial Pin
Analog Pin
Control
INT
Physical Pin
Port Pin
Pin function
Interrupt Pin
PWM Pin
Port Power
The power sum for each pin's group should not exceed 100mA
Absolute MAX per pin 20mA recommended 10mA
Absolute MAX 200mA for entire package
www.bq.com
04 AUG 2014
ver 3 rev 1

Figure 26: Arduino Mega 2560 Datasheet Overview

1. Product technical specifications

1.1 HM-10 Schematic



1.2 HM-10 Size

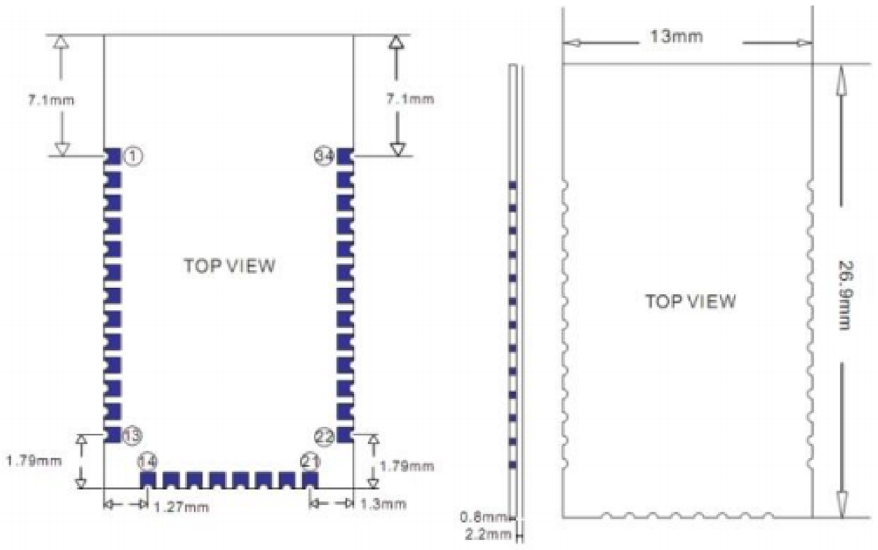
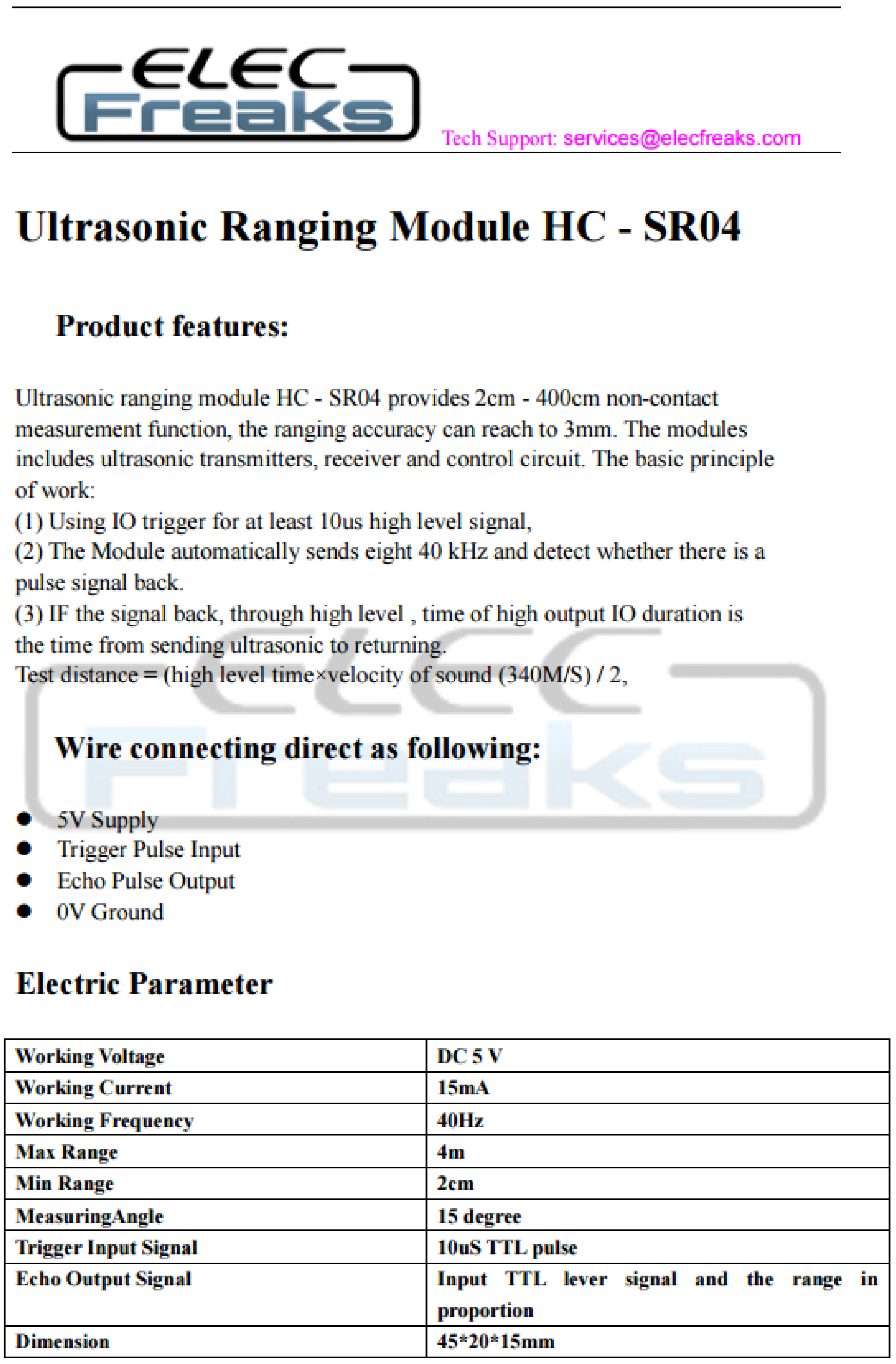


Figure 27: HM-10 Bluetooth Module Datasheet Overview



Tech Support: services@elecfreaks.com

Ultrasonic Ranging Module HC - SR04

Product features:

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:
(1) Using IO trigger for at least 10us high level signal,
(2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
(3) IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.
Test distance = (high level time×velocity of sound (340M/S) / 2,

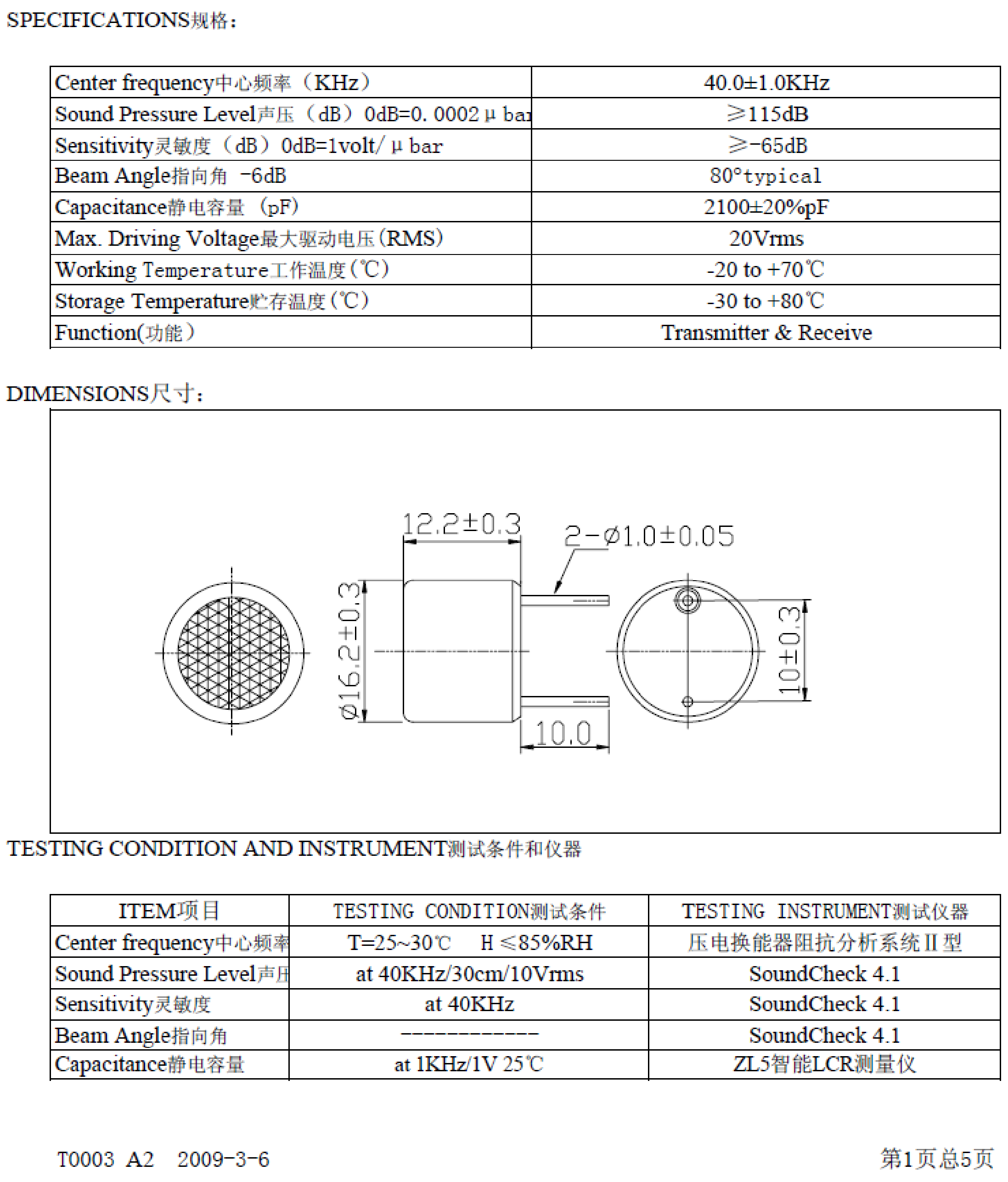
Wire connecting direct as following:

- 5V Supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

Electric Parameter

Working Voltage	**DC 5 V**
Working Current	**15mA**
Working Frequency	**40Hz**
Max Range	**4m**
Min Range	**2cm**
MeasuringAngle	**15 degree**
Trigger Input Signal	**10uS TTL pulse**
Echo Output Signal	**Input TTL lever signal and the range in proportion**
Dimension	**45*20*15mm**

Figure 28: HC-SR04 Sensor Datasheet Overview



SPECIFICATIONS规格：

Center frequency中心频率（KHz）	40.0±1.0KHz
Sound Pressure Level声压（dB）0dB=0. 0002 μ ba	≥115dB
Sensitivity灵敏度（dB）0dB=1volt/ μ bar	≥-65dB
Beam Angle指向角 -6dB	80°typical
Capacitance静电容量（pF）	2100±20%pF
Max. Driving Voltage最大驱动电压(RMS)	20Vrms
Working Temperature工作温度(℃)	-20 to +70℃
Storage Temperature贮存温度(℃)	-30 to +80℃
Function(功能)	Transmitter & Receive

DIMENSIONS尺寸：

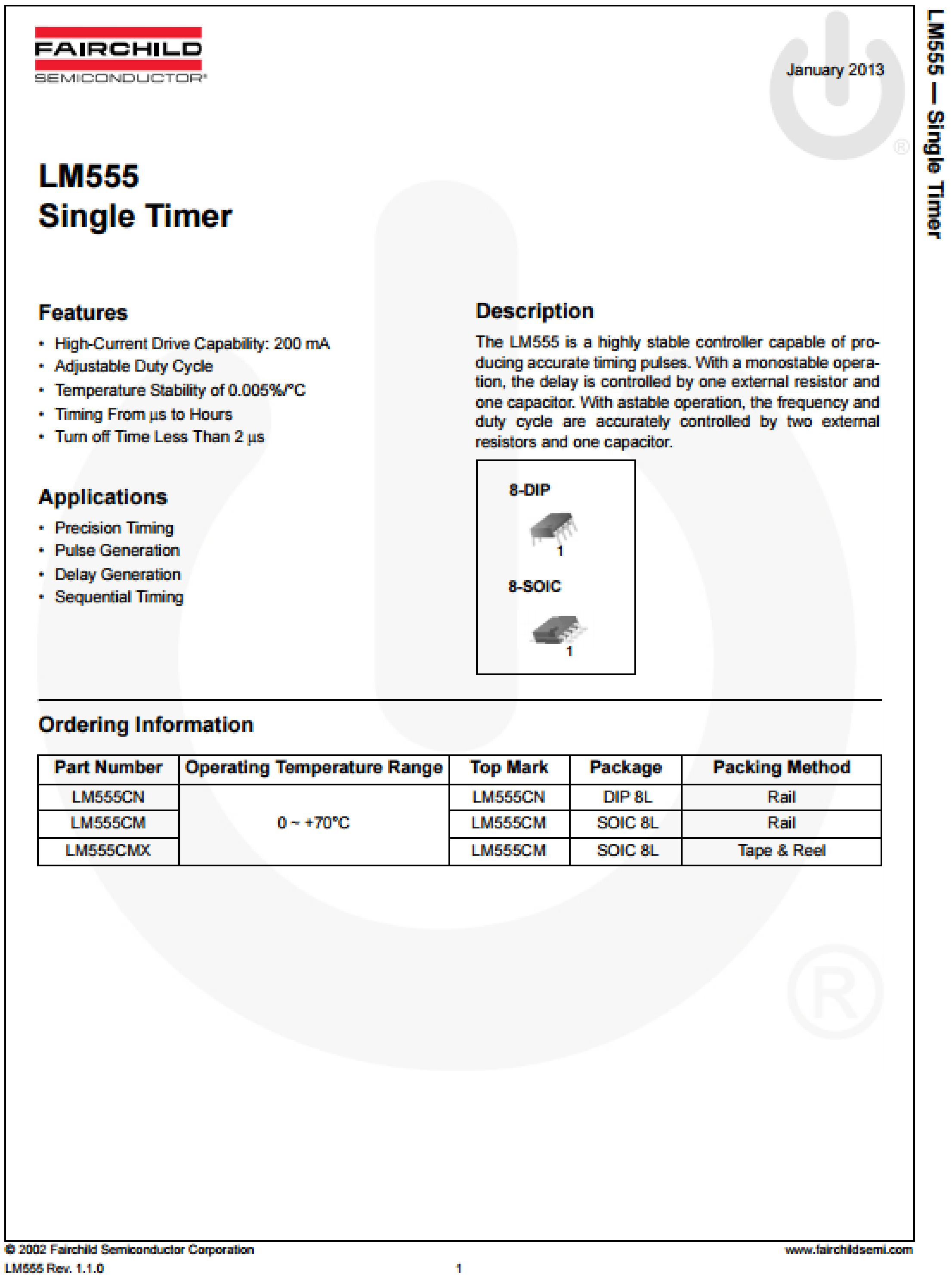
12.2±0.3
2-Ø1.0±0.05
Ø16.2±0.3
10±0.3
10.0

TESTING CONDITION AND INSTRUMENT测试条件和仪器

ITEM项目	TESTING CONDITION测试条件	TESTING INSTRUMENT测试仪器
Center frequency中心频率	T=25~30℃ H ≤85%RH	压电换能器阻抗分析系统 II 型
Sound Pressure Level声压	at 40KHz/30cm/10Vrms	SoundCheck 4.1
Sensitivity灵敏度	at 40KHz	SoundCheck 4.1
Beam Angle指向角	------------	SoundCheck 4.1
Capacitance静电容量	at 1KHz/1V 25℃	ZL5智能LCR测量仪

T0003 A2 2009-3-6 第1页总5页

Figure 29: Ultrasonic Transducer US1640 Datasheet Overview



FAIRCHILD SEMICONDUCTOR®

January 2013

LM555 — Single Timer

LM555
Single Timer

Features

- High-Current Drive Capability: 200 mA
- Adjustable Duty Cycle
- Temperature Stability of 0.005%/°C
- Timing From μs to Hours
- Turn off Time Less Than 2 μs

Applications

- Precision Timing
- Pulse Generation
- Delay Generation
- Sequential Timing

Description

The LM555 is a highly stable controller capable of producing accurate timing pulses. With a monostable operation, the delay is controlled by one external resistor and one capacitor. With astable operation, the frequency and duty cycle are accurately controlled by two external resistors and one capacitor.

8-DIP

1

8-SOIC

1

Ordering Information

Part Number	Operating Temperature Range	Top Mark	Package	Packing Method
LM555CN	0 ~ +70°C	LM555CN	DIP 8L	Rail
LM555CM		LM555CM	SOIC 8L	Rail
LM555CMX		LM555CM	SOIC 8L	Tape & Reel

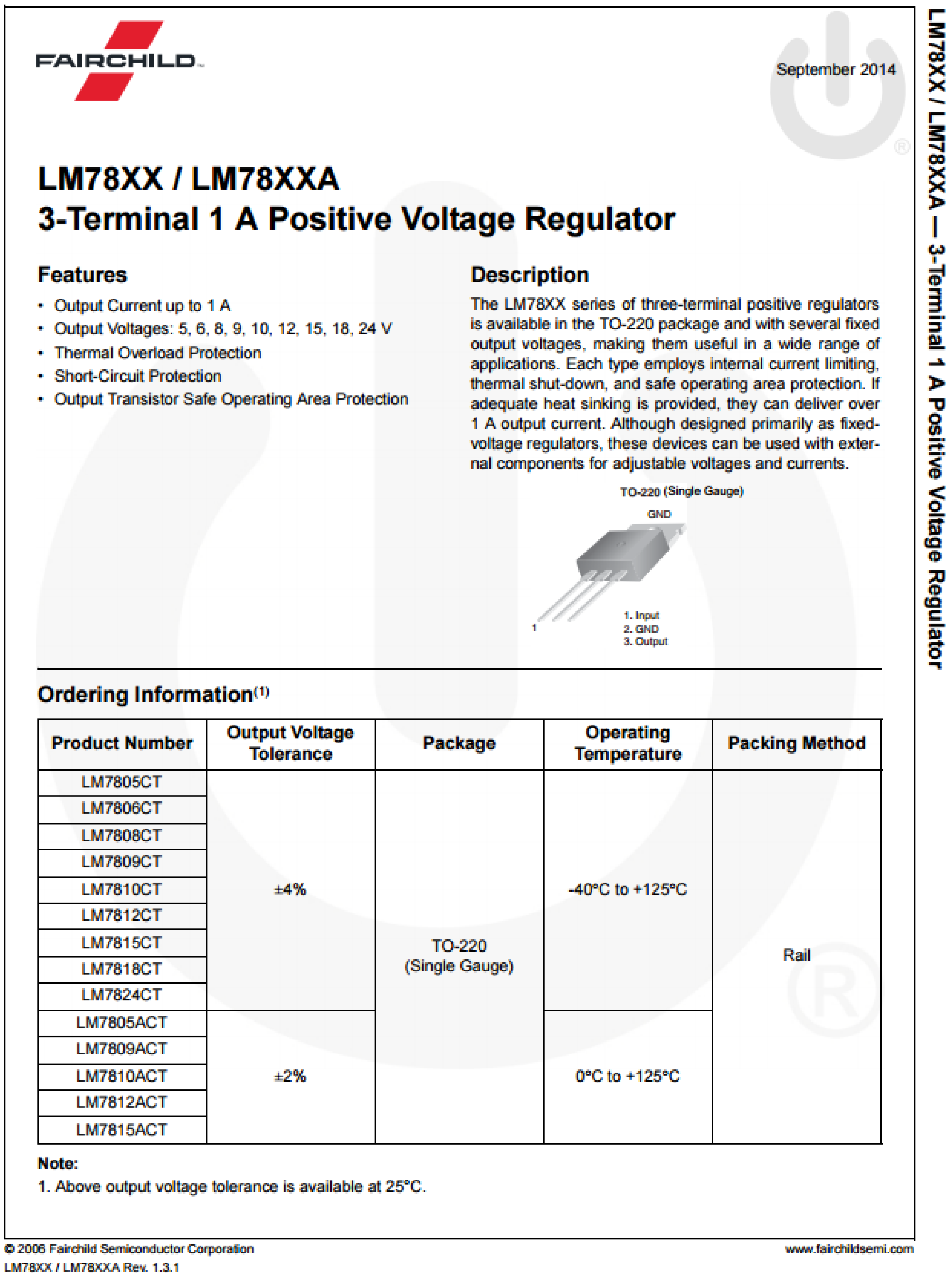
© 2002 Fairchild Semiconductor Corporation

www.fairchildsemi.com

LM555 Rev. 1.1.0

1

Figure 30: LM555 Timer Datasheet Overview



FAIRCHILD

September 2014

LM78XX / LM78XXA — 3-Terminal 1 A Positive Voltage Regulator

LM78XX / LM78XXA 3-Terminal 1 A Positive Voltage Regulator

Features

- Output Current up to 1 A
- Output Voltages: 5, 6, 8, 9, 10, 12, 15, 18, 24 V
- Thermal Overload Protection
- Short-Circuit Protection
- Output Transistor Safe Operating Area Protection

Description

The LM78XX series of three-terminal positive regulators is available in the TO-220 package and with several fixed output voltages, making them useful in a wide range of applications. Each type employs internal current limiting, thermal shut-down, and safe operating area protection. If adequate heat sinking is provided, they can deliver over 1 A output current. Although designed primarily as fixed-voltage regulators, these devices can be used with external components for adjustable voltages and currents.

TO-220 (Single Gauge)

GND

1

1. Input
2. GND
3. Output

Ordering Information[1]

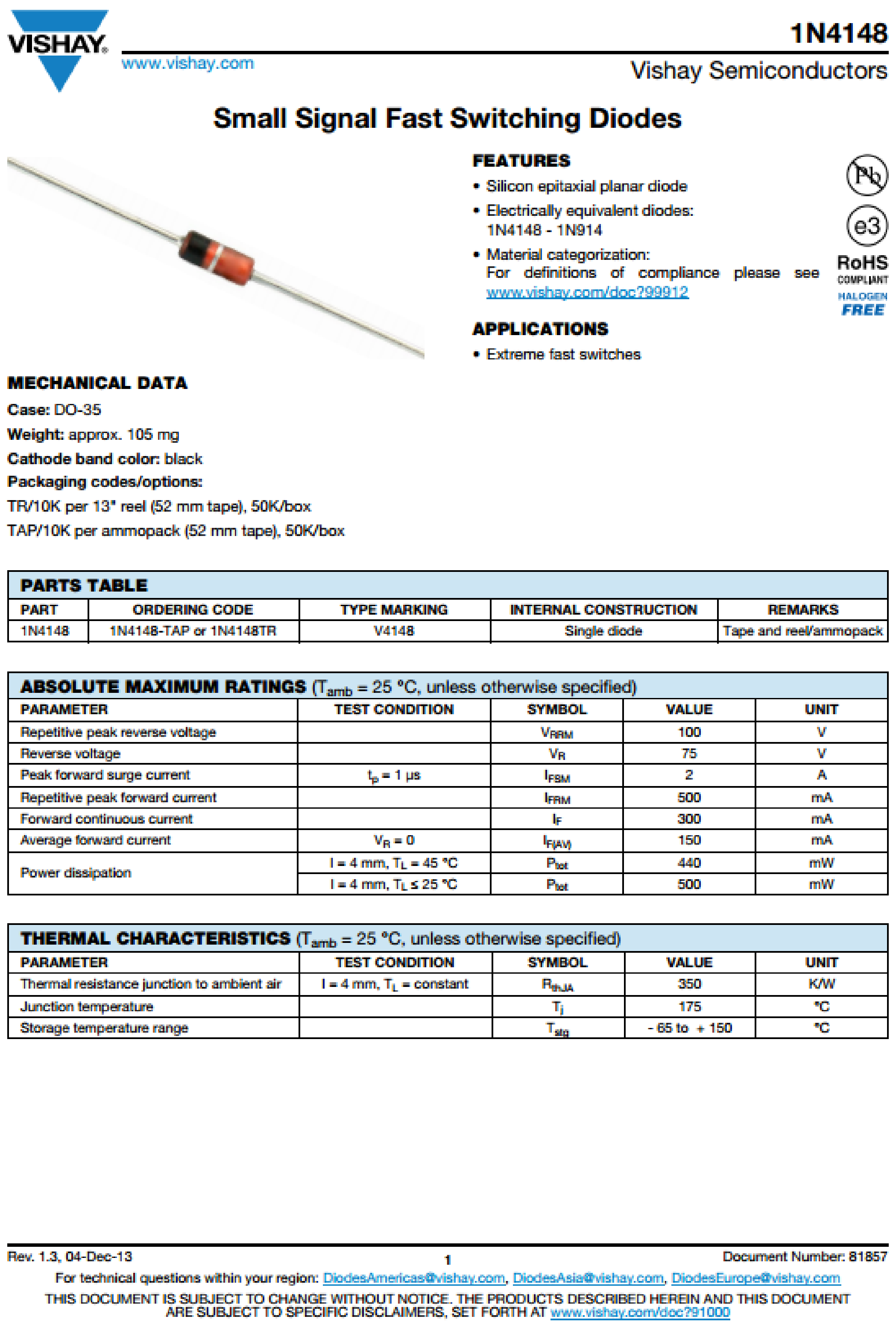
Product Number	Output Voltage Tolerance	Package	Operating Temperature	Packing Method
LM7805CT	±4%	TO-220 (Single Gauge)	-40°C to +125°C	Rail
LM7806CT				
LM7808CT				
LM7809CT				
LM7810CT				
LM7812CT				
LM7815CT				
LM7818CT				
LM7824CT				
LM7805ACT	±2%		0°C to +125°C	
LM7809ACT				
LM7810ACT				
LM7812ACT				
LM7815ACT				

Note:
1. Above output voltage tolerance is available at 25°C.

© 2006 Fairchild Semiconductor Corporation
LM78XX / LM78XXA Rev. 1.3.1

www.fairchildsemi.com

Figure 31: Voltage Regulators LM7805 Datasheet Overview



VISHAY.

www.vishay.com

1N4148

Vishay Semiconductors

Small Signal Fast Switching Diodes

FEATURES

- Silicon epitaxial planar diode
- Electrically equivalent diodes: 1N4148 - 1N914
- Material categorization: For definitions of compliance please see www.vishay.com/doc?99912

e3 RoHS COMPLIANT HALOGEN FREE

APPLICATIONS

- Extreme fast switches

MECHANICAL DATA

Case: DO-35

Weight: approx. 105 mg

Cathode band color: black

Packaging codes/options:

TR/10K per 13" reel (52 mm tape), 50K/box

TAP/10K per ammopack (52 mm tape), 50K/box

PARTS TABLE

PART	ORDERING CODE	TYPE MARKING	INTERNAL CONSTRUCTION	REMARKS
1N4148	1N4148-TAP or 1N4148TR	V4148	Single diode	Tape and reel/ammopack

ABSOLUTE MAXIMUM RATINGS (T_{amb} = 25 °C, unless otherwise specified)

PARAMETER	TEST CONDITION	SYMBOL	VALUE	UNIT
Repetitive peak reverse voltage		V_{RRM}	100	V
Reverse voltage		V_R	75	V
Peak forward surge current	t_p = 1 µs	I_{FSM}	2	A
Repetitive peak forward current		I_{FRM}	500	mA
Forward continuous current		I_F	300	mA
Average forward current	V_R = 0	$I_{F(AV)}$	150	mA
Power dissipation	l = 4 mm, T_L = 45 °C	P_{tot}	440	mW
	l = 4 mm, $T_L \leq$ 25 °C	P_{tot}	500	mW

THERMAL CHARACTERISTICS (T_{amb} = 25 °C, unless otherwise specified)

PARAMETER	TEST CONDITION	SYMBOL	VALUE	UNIT
Thermal resistance junction to ambient air	l = 4 mm, T_L = constant	R_{thJA}	350	K/W
Junction temperature		T_j	175	°C
Storage temperature range		T_{stg}	- 65 to + 150	°C

Rev. 1.3, 04-Dec-13 1 Document Number: 81857

For technical questions within your region: DiodesAmericas@vishay.com, DiodesAsia@vishay.com, DiodesEurope@vishay.com

THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE. THE PRODUCTS DESCRIBED HEREIN AND THIS DOCUMENT ARE SUBJECT TO SPECIFIC DISCLAIMERS, SET FORTH AT www.vishay.com/doc?91000

Figure 32: 1N4148 Diode Datasheet Overview