THE MECHATRONIC MULE

EN 487 Engineering Design Project

Team Members:

- Allan Quillen
- Anthony Olvera
- Jonathan Widjaja
- Lucas Mendicello

MAY 5, 2017
COLORADO STATE UNIVERSITY PUEBLO

Table of Contents

List of figures

List of tables

List of Definitions

- **3D Printing:** *a process for making a physical object from a three-dimensional digital model, typically by laying down many successive thin layers of a material.*
- **AGM (Absorbent Glass Matt) Battery:** *lead acid battery which that the electrolyte is held in the glass mats, as opposed to freely flooding the plates.*
- **Arduino:** *refers to an open-source electronics platform or board and the software used to program it.*
- **Automated Guided Vehicle (AGV):** *a mobile robot that follows markers or wires in the floor, or uses vision, magnets, or lasers for navigation.*
- **Automation:** *the technique, method, or system of operating or controlling a process by highly automatic means, as by electronic devices, reducing human intervention to a minimum.*
- **Average Human Walking Speed:** *the average human walking speed is about 5.0 kilometers per hour (km/h), or about 3.1 miles per hour (mph).*
- **Bluetooth:** *a standard for the short-range wireless interconnection of cellular phones, computers, and other electronic devices.*
- **Breadboard:** *a board for making an experimental model of an electric circuit.*
- **C Rating:** *a measure of the rate at which a battery is discharged relative to its maximum capacity.*
- **CAD:** *computer aided design.*
- **COSGC (Colorado Space Grant Consortium):** *a state-wide organization funded by NASA, involving independent study at 17 colleges, universities and institutions around Colorado.*
- **Current:** *The flow or electric charge.*
- **DAQ (Data Acquisition):** *is the process of measuring an electrical or physical phenomenon such as voltage, current, temperature, pressure, or sound with a computer.*
- **DC (Direct Current***): the unidirectional flow or movement of electric charge carriers (which are usually electrons).*
- **DC Motor:** *any of a class of electrical machine that converts direct current electrical power into mechanical power.*

- **Dynamic:** *Changing.*
- **Eclipsing:** *an obscuring of the light from one celestial body by the passage of another between it and the observer or between it and its source of illumination.*
- **Feedback Control System:** *Takes the system output into consideration, which enables the system to adjust its performance to meet a desired output response.*
- **Gantt Chart:** *a chart in which a series of horizontal lines shows the amount of work done or production completed in certain periods of time in relation to the amount planned for those periods.*
- **GPS (Global Positioning System):** *a radio navigation system that allows land, sea, and airborne users to determine their exact location, velocity, and time 24 hours a day, in all weather conditions, anywhere in the world.*
- **High Density Polyethylene (HDPE):** *or polyethylene high-density (PEHD) is a polyethylene thermoplastic made from petroleum.*
- **Image Processing:** *the analysis and manipulation of a digitized image, especially in order to improve its quality.*
- **IMU (Inertial Measurement Unit):** *an electronic device that measures and reports a body's specific force, angular rate, and sometimes the magnetic field surrounding the body, using a combination of accelerometers and gyroscopes, sometimes also magnetometers.*
- **IR (Infrared***): Refers to a proximity sensor which uses emissions of infrared radiation to measure distances.*
- **LIDAR (Light Imaging Detection and Ranging):** *is a surveying method that measures distance to a target by illuminating that target with an infrared laser light.*
- **Locomotion:** *movement or the ability to move from one place to another.*
- **Mechanics:** *the branch of applied mathematics dealing with motion and forces producing motion.*
- **Mechatronics:** *technology combining mechanical and electrical engineering.*
- **Microcontroller / Microprocessor:** *an integrated circuit that contains all the functions of a central processing unit of a computer.*
- **Obstacle Detection:** *(OD) is one of the main components of the control system of autonomous vehicles which allows them to navigate around obstacles.*
- **Parameter:** *a numerical or other measurable factor forming one of a set that defines a system or sets the conditions of its operation.*

- **PCB:** *Printed Circuit Board*
- **Power:** *The rate, per unit time, at which electrical energy is transferred by an electric circuit.*
- **Processing Speed:** *one of the measures of cognitive efficiency or cognitive proficiency.*
- **Proportional Control:** *a type of linear feedback control system.*
- **Proportional Derivative Control**: *looks at the rate of change in the error. If the error is rapidly approaching zero, the output of the derivative calculation attempts to slow things down to avoid overshooting the setpoint.*
- **Robot:** *a machine capable of carrying out a complex series of actions automatically, especially one programmable by a computer.*
- **Signature:** *something (as a tune, style, or logo) that serves to set apart or identify; also: a characteristic mark*
- **Static:** *unchanging*
- **Tolerance:** *the permissible limit or limits of variation.*
- **Trim:** *make (something) neat or of the required size or form by cutting away irregular or unwanted parts.*

1. Introductory Materials

Executive Summary

As engineers, we use our scientific knowledge to solve real world problems. Engineering is the profession where math, science and common sense are applied to create or design some solution to real world problems. As engineers, we are innovating a technology that will make human beings lives easier while also keeping the earth healthy.

Carrying a heavy load of gear, items and/or equipment can be a painful process for many, especially those such as the young, the elderly, the disabled and those of us with back problems and or health complications. It can also be a time-consuming process that takes away the enjoyment of recreational activities or the efficiency of work related projects.

For a better future and to help make daily activities easier for the public, we decided to build the mechatronic mule as our senior design project using all our knowledge from the beginning of our studies at this university until now. The Mechatronic Mule is able to simplify our daily activities without producing negative effects towards the environment.

The Mechatronic Mule is a robotic wagon with autonomous following capability. It is capable of carrying loads of up to 10 cubic feet and 200 lbs. without having the hassle of manually pushing or pulling it. It automatically follows behind the designated user at asset distance as their personal cargo caddy.

The Mechatronic Mule is made of lightweight aluminum framing. So that it is lightweight but still ridged and strong. It is paneled with a high-density polyethylene plastic that is tough enough to handle moderate impacts, normal wear and tear and also is water resistant.

The Mechatronic Mule is powered by lead acid gel batteries and high powered electric DC Motors. The batteries contain enough capacity to allow for hours of continuous runtime while the motors are powerful enough to move the wagon up a 20-degree slope at full capacity.

The Mechatronic Mule is a smart machine. It uses image processing technology in parallel with a microcontroller to recognize certain characteristics such as color, combinations of color, shapes and even infrared light, so it is sure to only recognize its intended user and effectively trach and follow their path. This machine has many applications and is easy to operate by anyone.

Acknowledgement

We would like take this opportunity to express our profound gratitude and deep regards to Dr. Jaksic for his exemplary guidance, monitoring and constant encouragement throughout the course. The blessing, help and guidance given by him time to time shall carry us a long way in the journey of life on which we are about to embark.

We would also take this opportunity to express a deep sense of gratitude to Paul Wallace for his cordial support, valuable information and guidance, which helped us in completing this project through various stages. Without him letting us use the machine shop, helping us to construct the mechatronic mule and guide us when we have problems constructing, this project would be impossible to get done.

Problem Statement

General problem statement

Many people have gotten injured from carrying heavy objects, In this project we decided to make an automated guided vehicle which we call the mechatronic mule to help users carry heavy equipment from one location to another location. Some people with disabilities or those who are not strong enough to pull a cart or wagon will have an opportunity to go on an adventure. Carrying objects that are too heavy could lead to back problems and other health issues, we know medical expenses can be expensive and investing in a mechatronic mule will help prevent health issues related to carrying heavy objects.

General solution approach

The design of the mechatronic mule looks like a cart, it's battery powered and electric motor driven. The mechatronic mule will follow the user who is locked onto by the pixy to track and follow them without having to pull the mechatronic mule around. The pixy will follow the user who wears a specific color shirt or vest. The sensors also will be used to keep a set distance from the user to prevent a lost connection from being too far or too close. The microprocessor inside the mechatronic mule will determine to speed up or slow down depending on the reading of the sensors.

Operating environment

Since we are targeting the outdoor enthusiast the automated guided vehicle operating environment should be able to work in areas like on a mountain with a slope of 20 degrees or less. However, for our project we will simplify its design to operate as a proof of concept on a dirt road only. It also will be able to operate in rainy weather and in lightly snowy conditions. The bed of the cart is made of aluminum sheet which is light and strong enough to hold cargo up to 200lbs. The Mechatronic Mule should be able to operate on lightly rough conditions.

Intended user(s) and intended use(s)

User(s)

Our main target for this product is mostly outdoor enthusiasts, to help them carry their gear or equipment for camping or other outdoor activities. Many people cannot enjoy outdoor activities because they are not able to carry heavy gear due to their health condition. Some people like kids or elderly people who might have trouble in carrying their stuff, now they can use the mechatronic mule to carry their stuff around. The mechatronic mule also can be used by golfers as their caddy to carry their golf club bag around the golf course.

Use(s)

Many people only participate in outdoor activities once in their lifetime because they feel it is not worth it as the gear is too heavy to carry in one trip which requires them to go back and forth while only enjoying the activities for a short time. The automated guided vehicle will not only be useful for outdoor enthusiasts but also can be used on construction sites, industrial sites and many more areas depending on which size and trim of the automated guided vehicle is needed.

The mechatronic mule is created to help people move their gear or belongings in less trips than carrying it by themselves alone so they can enjoy more outdoor activities and inspire more interest in others in outdoor activities. The mechatronic mule will follow you by itself so you don't have to pull it which requires less energy as you only need to carry gear by loading and unloading it from the mechatronic mule. It can be operated by anyone, even people with back pain that cannot carry heavy weight.

Assumptions and Limitations

Assumption(s)

We believe that the automated guided vehicle will work on a mountain with up to a 20° slope without a problem. It will be able to move as fast as the average human walking speed and keep up without getting lost. It will be able to operate in rainy and lightly snowy weather conditions. The lifetime for the automated guided vehicle should be good to run for at least 5 years with regular maintenance.

Limitation(s)

The limitation of the automated guided vehicle is on a full charge it will be able to run for up to 2 hours and carry a load of up to 200 lbs. of gear, going uphill of slope 20° or less. Because this end product will be a working prototype, the limitation of the mechatronic mule is it might not be running properly on rough terrain as there is no suspension installed and we decided to use free floating caster wheels on the back for easier maneuvering. Since the mechatronic mule itself weighs about 150lbs. it will be hard to lift up into a truck, therefore it will need a ramp to load it to the back of a truck.

Expected end product and other deliverables

The end product is a working prototype. The project will be done by the end of April 2017. The following function works pretty well both outdoor and indoor. With a mass production we can reduce the cost and weight of the mechatronic mule. From a working prototype. We got a rough idea of how the mechatronic mule should look for the end consumer deliverable product.

2. Project Records and Activities

Functional Requirements

This product consists of a four wheeled, mobile robotic cart. Its inside has an acceptable surplus of empty volume for its user's cargo. The structural integrity of this product is durable enough to handle normal wear and tear, endure accidental impacts and allow for transportation of heavy objects without accumulating any major damage. This vehicle can be operated by anyone of eight years of age or older and does not require extensive training on how it operates. There is one main power switch and one signature teach push button. The user must simply power the vehicle and teach their signature then the mechatronic mule will follow that signature ant the distance it was set. This machine is also safe for its users; there remains no uninsulated wires or sharp edges. This product is able to track and follow its user while bearing the weight of its user's cargo (up to 200 lbs.). It can move at any range of speeds similar to the average human walking speed (up to four miles per hour) while adjusting to its user specific speed preference. It can move across easy to moderate terrain e.g. slopes of 20 degrees or less on a paved or dirt road. As a safety feature it can detect if its path is blocked which ensures it never collides with a person. Also, there is a wireless emergency, remote start/stop switch that the user can carry to start and stop the vehicle as desired. This product is not designed as a mobility device for people and should never bear passengers, also the device shall not be used to transport hazardous substances or uncontained liquids as these actions can lead to injury.

Constraint Considerations

This product has constraints on the following capabilities: energy, power, mobility, quality and environmentally invoked functionality. The vehicle contains a limited amount of electrical energy to power its locomotion. Thus, depending on the cargo, load and terrain the battery life will vary and its user needs to consider how much cargo the vehicle can carry, also where and how far they can take it. Next the vehicle has a maximum load limit of 200lbs. If its user attempts to load the vehicle past its weight capacity the vehicle can fail to move and could incur damages. The mobility of the vehicle is limited by its size and physical characteristics, it has a limit on its ability to move around tight corners, under short clearings, over large objects, through deep water, and over slick surfaces. There is also a constraint on its quality. Needless to say, it is not indestructible and it can be damaged if it were to take heavy impacts in certain areas as well as allowing sensitive electronics to overheat and or get wet. Finally, as we've discovered during development environmental lighting conditions do have an impact on the vehicles performance. Such conditions include but are not limited to at night time and facing sunrise or sunset. Future work focus on eliminating these constraints.

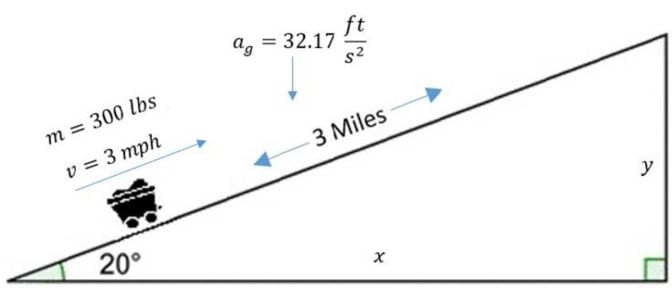
Considered Approaches

Technology Considerations

To allow for a clean yet quiet and lightweight vehicle, electric DC motors are our choice for powering the vehicles locomotion. The motors are powered by gel lead acid batteries as they allow for higher current draw than conventional lead acid batteries. This is important as we have calculated a maximum current requirement of approximately 25 amps to each motor. Various sensors are used for navigation as opposed to GPS due to resolution accuracy constraints. These various sensors consist of ultrasonic sensors to detect obstacles in the vehicles path, a photo resistor to provide data to the microprocessor about the environmental lighting conditions which in turn can be used to illuminate light onto the target to aid an eclipsing effect or night time conditions. Lastly and most critically the image processing camera provides all data on signature image size, color and location which is used to appropriately control the motors which in turn defines the vehicles navigation. All remaining capability is provided via C coded algorithms uploaded to an Arduino microcontroller which is integrated to acquire data from the sensors and execute desired actions based on that data.

Power and Energy

In order to determine the appropriate battery and motors for the mechatronic mule power an energy calculations are required. Below is a simplified calculation using the principles of Newtonian mechanics and our desired specification parameters. This calculation gives us a fairly accurate measure of how powerful the machine's motors need to be and how much capacity the machine's batteries need to contain. By calculating the mechanical power and energy required to move the mechatronic mule up a twenty-degree slope for one hour we can select appropriate batteries and motors to convert enough electrical power and energy into that mechanical power and energy.



$$y = 3\ miles\ \times \sin(20\,°) = 1.026\ miles \times \frac{5280\ ft}{mile} \times \frac{0.3048\ meters}{ft} = 1651.28\ meters$$

$$PE = ma_g h = 300\ lbs \times \frac{0.454\ kg}{lb} \times 9.81 \frac{m}{s^2} \times 1651.28 = 2.204 \times 10^6\ J \approx 2.204\ MJ$$

Note that this calculation uses a total mass of 300 lbs, which is the sum of the mass of the vehicle (100 lbs) and the maximum load capacity (200lbs).

$$1\ amp \times hour \times volt = 3600\ J,\ \ using\ a\ 24\ volt\ battery,\ \ 24\ amp \times hour \times volts = 86.4\ kJ$$

$$So\ the\ battery's\ capacity\ C = \frac{2.204 \times 10^6 J /_{24\ volts}}{86.4 \times 10^3 J /_{24\ amp \times hour \times volts}} = 25.505\ amp \times hours$$

$$Power\ required = \frac{Work}{Time} = 2.204\ MJ/\left[(3\ Miles/3\ Miles/hr) \times \left({}^{3600\ sec}/_{hr}\right)\right] = 612.108\ watts$$

$$With\ 2\ powered\ motors\ we\ need\ 306.054\ watts\ per\ motor$$

$$Maximum\ current\ draw\ I = \frac{P}{V} = \frac{612.108\ watts}{24\ volts} = 25.505\ amps$$

Because the battery's capacity is equal to its discharge current over one hour it must have a C rating of at least 1, also of the calculations above neglect losses due to friction therefore we will increase power requirement by 10%

$$Power\ required = 1.1 \times 306.054 = 336.67\ watts\ per\ motor$$

$$Energy\ required = 3600\ sec \times 2 \times 336.67\ watts = 2.424\ MJ$$

$$Battery's\ capacity = \frac{2.424 \times 10^6 J/_{24\ volts}}{86.4 \times 10^3 J/_{24\ amp \times hour \times volts}} = 28.055\ amp \times hour$$

Chassis and Frame Construction

At the beginning of this project we proposed to using bar stock and welding it together. Instead we ended up using aluminum T slot bar for the chassis and frame construction. We decided so because Dr. Jaksic recommend us to use the T slot bar for easier assembly and disassembly in case something wrong has happened. Even though the T slotted aluminum bars are more expensive than a bar stock, it is easier to assemble. For the base of the frame we decided on using an aluminum plate as it is a light and strong material. While for the sides we decided on the plywood to keep it under the budget.

At the beginning of the chassis and frame construction we decided on 50 inches long x 35 inches tall x 28 inches wide dimension. However in the middle of construction we realize that size is not a standard size and it is difficult to get aluminum sheet and plywood with the size we needed, and Paul advised us to cut down the size to standard size. So we cut down the dimension size and ended up with 48 inches long with 24 inches tall and 28 inches wide so it will fit through the door. We ended up using 30mm hollow t-slotted aluminum bar as not only is it strong but it is also light weight; keeping the mechatronic mule weight as light as possible.

For the chassis, we build it from a couple t-slotted aluminum bars that were cut down to size and assembled it with the wheel-well notch. For the chassis under the cargo area we put down a couple t-slotted aluminum bars across the chassis for extra support underneath the aluminum sheet so it will be able to hold up to 200 lbs. of cargo without bending the aluminum sheet.

For the frame we used t-slotted aluminum bar as well and we also added some 45° cross supports for added stability to the frame and reinforced it so it can withstand heavy weight objects. Before we added the 45° cross supports we could feel that it is not strong enough when we move it around during the assembly process. After we added the cross support, it made the frame a lot more sturdy and made it stay in place better when we moved it around during the assembly process.

Locomotion

To accomplish the maneuverability required to track a human user we have chosen to use two motors connected to sprockets mounted to right and left axels to power the front wheels. The rear wheels are connected to free-floating caster wheels in the back of the assembly to achieve optimal performance. We had looked at many different options for the drive-style and steering style, independently powered wheels in front, back, as well on all four tires, but decided on front-wheel drive as it suited our design requirements best. The front wheels will operate in a skid-style steering similar to a tank, connected via a sleeve that includes needle-style bearings to reduce friction when turning. For the MechaMule to achieve the required speed we chose 24 Volt 350 Watt MY1016Z3 Gear Reduction Electric Motor with 9 Tooth 1/8" Bicycle Chain Sprocket) paired with a 48-tooth sprocket to ensure proper variable-walking speed can be maintained. The front wheels are 15", but are mounted in the middle tire and the rear caster-wheels are 8" to ensure the device is level. The motors chosen would provide required RPM for a 20° slope at full capacity is, however the calculations are assumed for a maintained trail without additional obstacles that would require additional power. The calculation of required max and minimum RPM was crucial in selecting motors to achieve the range required for the typical user, while still achieving the necessary power to function at full capacity as well. The average walking speed of an adult is listed between 3 to 4 mph which converts to an rpm of 89.64. Which is why the 48-tooth sprocket was chosen to reduce rpms. The motor gear and sprocket fit a single speed bicycle chain to ensure easy replacement when maintenance is needed. The chosen components should provide proper functionality for the MechaMule.

Control System and Code Design

The mechatronic mule uses the following two control systems; *object tracking* based on the Pixy camera and the pan and tilt mount that its attached to and *object following* using the motors and skid steering mechanism. Together these systems allow the vehicle to track and follow its user. Object tracking is implemented in the "TrackBlocks" function of the code. The Pixy camera handles the object detection based on the signature we've previously set in the Pixymon software. The camera will sweep back and forth on its panning mechanism until it finds its pre-programed signature then it will lock onto that image and report image size and color data to the Arduino micro controller. Using this data, the program controls the pan and tilt servos to keep the camera pointed towards the object its tracking. In the code this is done using two instances of a “ServoLoop” class, one for pan and one for tilt. The function in the code named “ServoLoop” is a feedback control loop that uses proportional derivative or PD control. The measurements are x and y for pan and tilt respectively. The set-points are the x and y coordinate for the direct center of the cameras field of view. And the outputs are the servo’s positions. During each iteration, the code calculates the difference between the measurement and the set-point as the pan and tilt errors. Then the “ServoLoop” control algorithm is invoked to calculate the outputs.

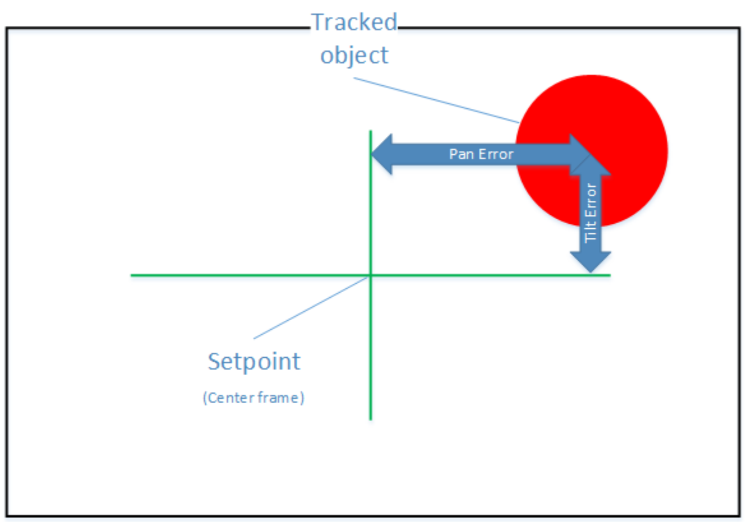


FIGURE 1 - OBJECT TRACKING

The next control system used in the code handles *object following*. In the code this is implemented in the “FollowBlocks” function which uses proportional or P control. In this function the two measurements are image size and pan position (note that tilt is not requires as the vehicle is constrained to the ground, a 2D surface) and the outputs are the speeds of the left-hand and right-hand motors. The image size is the area of the image and it gives us an idea of how far the vehicle is from its target. This area is then used to calculate the value assigned to the variable “forwardSpeed”. This causes the Mechatronic Mule to slowdown as it approaches its target. If the block area is larger than the set-point value “forwardspeed” will become negative and the robot will back up in reverse. The measurement of pan position tells how far to the left or right the target is from the set-point (directly ahead). We then use this value to adjust the speed differential between the left and right motor which causes the mule to steer towards its target.

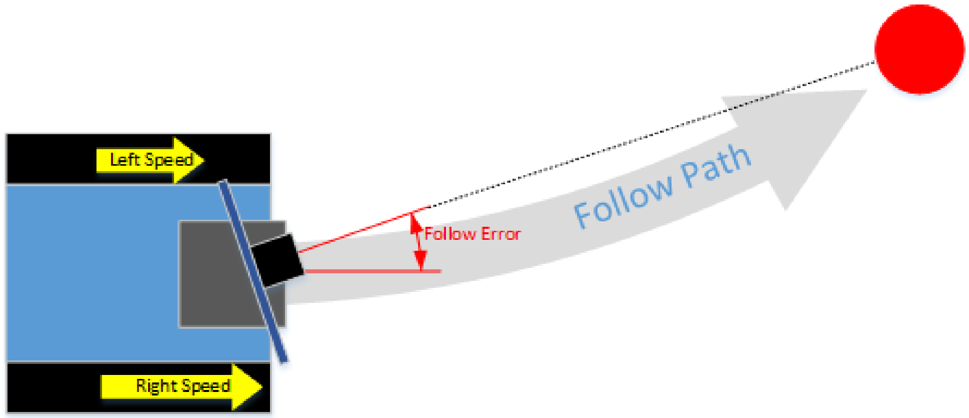


FIGURE 2 - OBJECT FOLLOWING

Electronics and Sensors

The main electronic components used in this project are an Arduino Due microcontroller for the main processor, the Pixy camera for image processing, the Sabertooth 2X25 motor driver and two Xbee radio modules for the wireless start stop system. The Arduino Due uses the Atmel SAM3X8E ARM Cortex-M3 CPU a 32-bit ARM core microcontroller, we chose this board because of it superior processing power and it also has an on board digital to analog converter that we plan to use for vehicle head lights for dark conditions or in aiding the eclipsing effect.

Technical Specifications

- Microcontroller: AT91SAM3X8E
- Operating Voltage: 3.3V
- Input Voltage (recommended): 7-12V
- Input Voltage (limits): 6-16V
- Digital I/O Pins: 54 (of which 12 provide PWM output)
- Analog Input Pins: 12
- Analog Output Pins: 2 (DAC)
- Total DC Output Current on all I/O lines: 130 mA
- DC Current for 3.3V Pin: 800 mA
- DC Current for 5V Pin: 800 mA
- Flash Memory: 512 KB all available for the user applications
- SRAM: 96 KB (two banks: 64KB and 32KB)

- Clock Speed: 84 MHz
- Length: 101.52 mm
- Width: 53.3 mm
- Weight : 36 g

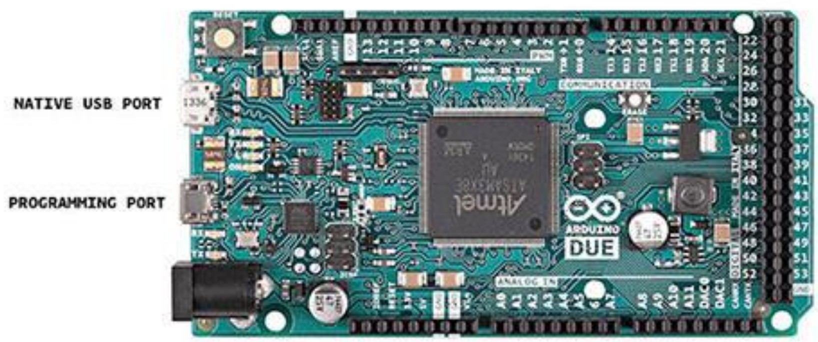


FIGURE 3 - ARDUINO DUE

This microcontroller is used in parallel with the Pixy (CMUcam5). Pixy (CMUcam5) is a partnership between the Carnegie Mellon Robotics Institute and Charmed Labs. Pixy comes from a long line of CMUcams. Vision (image) sensors are useful because they are so flexible. With the right algorithm, an image sensor can sense or detect practically anything. But there are two drawbacks with image sensors: 1) they output lots of data, dozens of megabytes per second, and 2) processing this amount of data can overwhelm many processors. And if the processor can keep up with the data, much of its processing power won't be available for other tasks.

Pixy addresses these problems by pairing a powerful dedicated processor with the image sensor. Pixy processes images from the image sensor and only sends the useful information (e.g. Blue block detected at x=54, y=103) to our microcontroller. And it does this at frame rate (50 Hz). The information is available through one of several interfaces: UART serial, SPI, I2C, USB, or digital/analog output. So our Arduino or other microcontroller can talk easily with Pixy and still have plenty of CPU available for other tasks.

Pixy uses a color-based filtering algorithm to detect objects. Color-based filtering methods are popular because they are fast, efficient, and relatively robust. Pixy calculates the color (hue) and saturation of each red, green, and blue pixel from the image sensor and uses these as the primary filtering parameters. The hue of an object remains largely unchanged with changes in lighting and exposure. Changes in lighting and exposure can have a frustrating effect on color filtering algorithms, causing them to break. Pixy's filtering algorithm is robust when it comes to lighting and exposure changes. However, the drastic change in lighting conditions inside to outside do have an effect. After much frustration, we came upon an Infrared locking firmware for the pixy, courtesy from a website called irlock.com. We use the same exact steps but just for infrared. In order to do this, we needed a filtering plate which filters out all visible light except infrared, this came from the website irlock.com. We also needed a specific camera lens for this as well (also from irlock website)

Pixy comes with a GUI (graphical user interface) called PixyMon. PixyMon is an application that runs on Windows, MacOs and Linux. It allows you to see what Pixy sees, either as raw or processed video. It also allows you to configure your Pixy, set the output port and manage color signatures. PixyMon communicates with Pixy over a standard mini USB cable. PixyMon is great for debugging your application. You can plug a USB cable into the back of Pixy and run PixyMon and then see what Pixy sees while it is hooked to your Arduino or other microcontroller -- no need to unplug anything. PixyMon is open source

Teaching pixy an object is relatively easy. In the PixyMon software simply go to "action" then set signature 1. This will freeze the frame of view on the pixy and then simply click the mouse and drag a square across the desired signature and that's it! Signature is now set. With infrared, however there is only one signature and that is infrared of course. To mitigate false detections, simply adjust certain values till you have mitigated the false detections from the sun, or the sun itself. To get around the sun simply set a max block size bigger than the sun, however this means your signature needs to be bigger than the sun, which is bigger than the sun from the pixies point of view, not the user's point of view. There is another way around this and that is in the code and is explained in the comments within the code itself.

Technical Specifications

- Processor: NXP LPC4330, 204 MHz, dual core
- Image sensor: Omnivision OV9715, 1/4", 1280x800
- Lens field-of-view: 75 degrees horizontal, 47 degrees vertical
- Lens type: standard M12 (several different types available)
- Power consumption: 140 mA typical
- Power input: USB input (5V) or unregulated input (6V to 10V)
- RAM: 264K bytes
- Flash: 1M bytes
- Available data outputs: UART serial, SPI, I2C, USB, digital, analog

- Dimensions: 2.1" x 2.0" x 1.4
- Weight: 27 grams

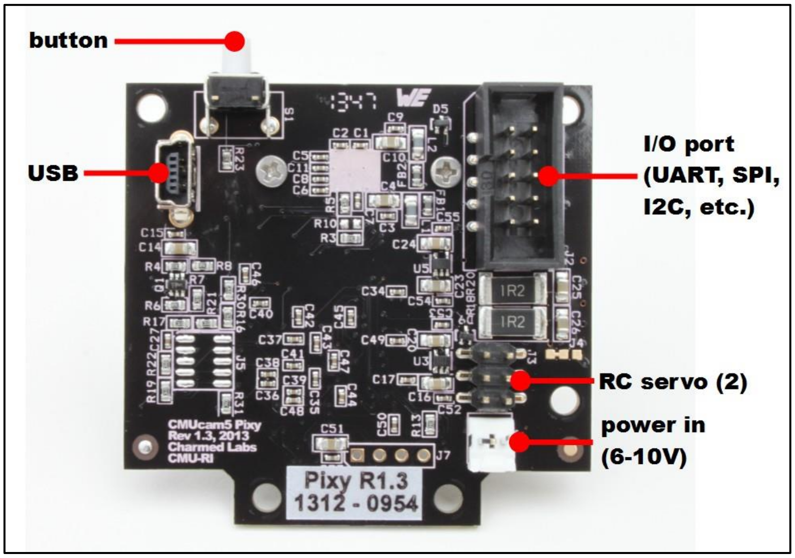


Figure 4 - Pixy

Another critical electrical component required for this project is a motor driver we've selected the Sabertooth 2x25 motor controller made by dimension engineering due to its power output capabilities as our motors require a maximum current draw of 19 amps each. It is suitable for large robots of up to 300 lbs. It has built in overcurrent and thermal protection which is a redundant safety feature which protects the driver from damage in the event of an over demanding power situation. The onboard dip switches allow for a voltage cutoff which protects the battery from over drainage and further damage. The sabretooth 2x25 is also a regenerative motor driver, meaning when the AGV moves downhill or is manually pushed the driver will provide a charge back to the batteries. This will save energy and ultimately increase the vehicles run time. In our project, we control the speed of the motors with packetized serial data which is possible using this motor controller. The data is produced by the Arduino and interpreted and converted to a motor speed by this motor driver.

Technical Specifications

- 25A continuous, 50A peak per channel
- 6-30V nominal, 33.6V absolute maximum
- Synchronous regenerative drive
- Ultra-sonic switching frequency

- Thermal and overcurrent protection
- Lithium protection mode
- Input modes: Analog, R/C, simplified serial, packetized serial
- Size: 2.6” x 3.2” x .82” 65 x 80 x 21 mm

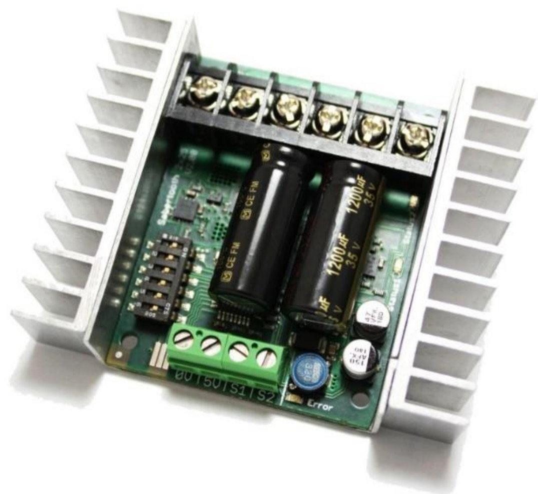


FIGURE 5 - SABERTOOTH 2X25

Xbee s1 radio modules by Digi are radio modules that communicate on the 2.4 GHZ radio band similar to Wi-Fi and Bluetooth these modules are capable of sending data packets wirelessly for a variety of purposes. However, our use for them is much simpler. We use two Xbee s1 radio modules (one as a receiver and one as a transmitter) in the mechatronic mule’s remote start/stop system they are programed in the XCTU software. The radios are assigned the same address and the same communication rate to ensure communication between the two radios. Then IO line passing is enabled which causes the receiving Xbee to mimic the pin voltage on the transmitting Xbee. Using an SPDT push button on the remote the D0 pin on the transmitting Xbee is connected to 3.3v this causes the receiving Xbee’s D0 pin to go high and in turn trigger a relay which cuts the speed signal to the motor. The button is locking which keeps the relay energized and the vehicle stopped until it is pressed again which then re-connects the speed signal to the motor driver and allows the vehicle to move once again.

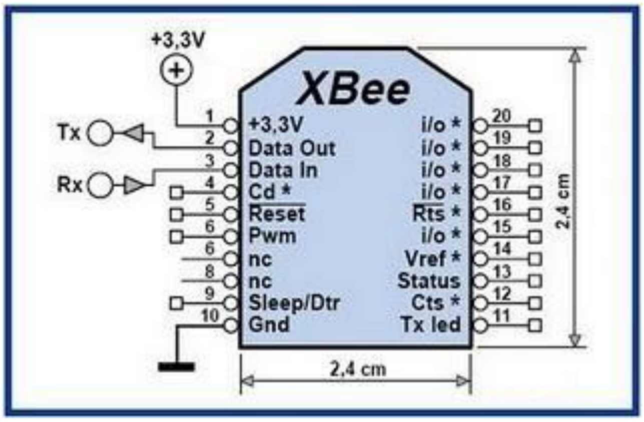


FIGURE 6 - XBEE S1

Technical Specifications

- 3.3V at 50mA
- 250kbps Max data rate
- 1mW output (+0dBm)
- 300ft (100m) range
- Fully FCC certified
- 6 10-bit ADC input pins
- 8 digital IO pins
- 128-bit encryption
- Local or over-air configuration
- AT or API command set
- Trace Antenna

Technical Approach Considerations

The most effective method in our technical approach consisted of prototyping on a scale model. We took advantage of a four-wheeled robotic machine that we built for independent study in the 2015 COSGC robotics challenge. The robot is already fitted with batteries, motors, a microcontroller and all necessary sensors to prototype the main concept and the final product. The entire robotic platform serves as a functional scale model for our design. We used the Solid Works 3D modeling CAD software to construct a final design, assembly and drawings, also to obtain critical data points on mass and mechanical functionality. In the meanwhile, we did all analyses, computations and calculations by hand or using the Matlab software to ensure all systems met specifications. We used software packages such as pixymon to configure the Pixy camera and XCTU to configure the Xbee radio modules. Lastly we utilized the CSU-Pueblo machine shop for all metallic parts fabrication and assembly.

Security Considerations

At this point we have minimal security considerations, and or proprietary information. However, all information is only accessible to the members of our team through our engineering logbook or drop box account, thus our data, designs, ideas etc. are secure.

Detailed Design

Drawings

Frame and Chassis

The following drawings are of the fully assembled design, as well as an exploded view showing various components. The assembly was created by our group, however the T-slot extrusion, wheels, tires, bearings, motors and batteries where all components found on Grabcad created by various users. The majority of components required customization for our design as lengths had to modified and individual parts were constructed into assemblies.

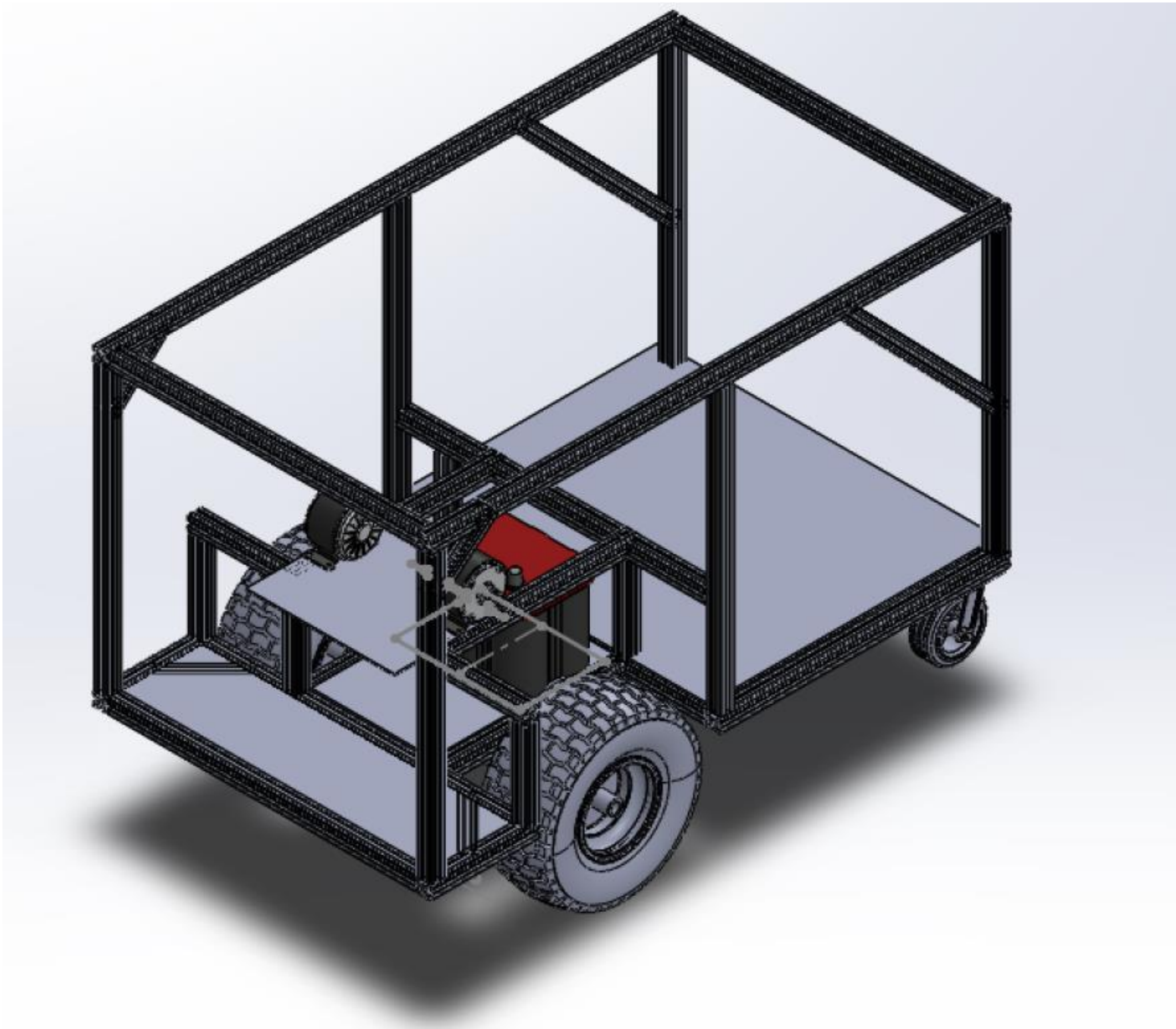


FIGURE 7 - CHASSIS & FRAME 3D

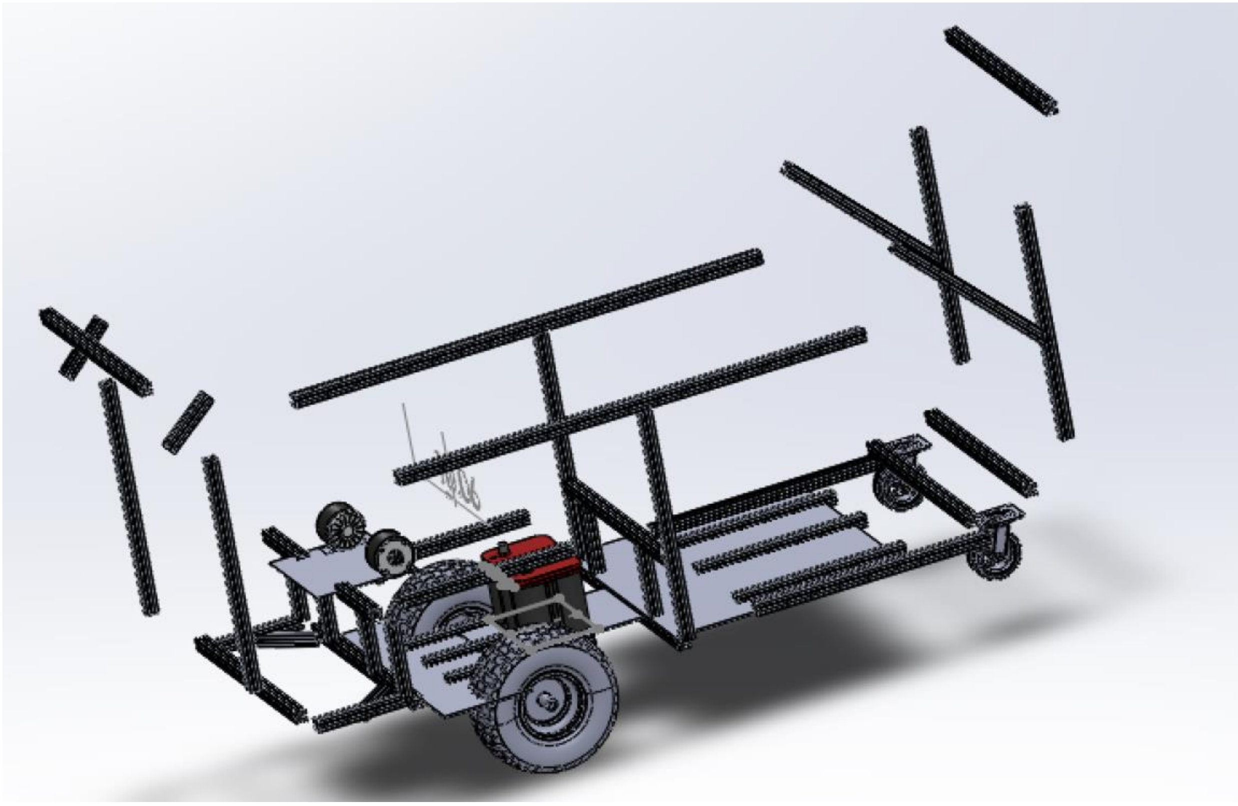


FIGURE 8 - FRAME & CHASSIS EXPLODED VIEW

Pixy Enclosure Case

This first design was from AntonioJose8 from thingiverse.com. we modeled the design from looking at all the pictures and measured certain dimensions on the pixy. It was printed on Colorado State University Pueblo's Engineering Department 3D printers in ABS Plastic. However, the pictures you see below are from the website www.thingiverse.com.

FIGURE 10 - FRONT HALF OF CASE

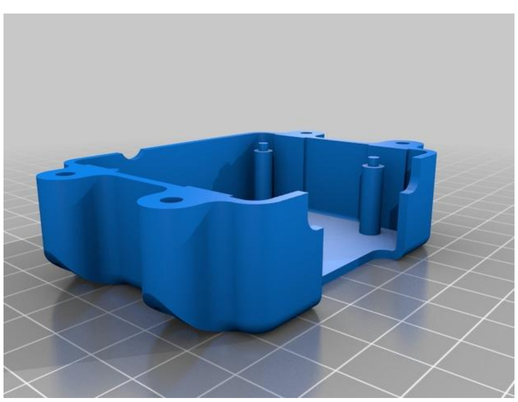


FIGURE 9 - BACK HALF OF CASE

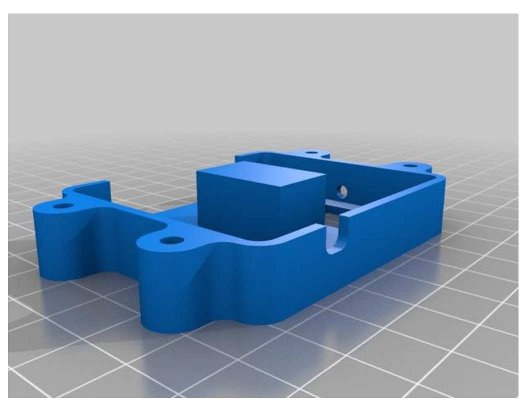


Figure 12 - Base Plate of Case

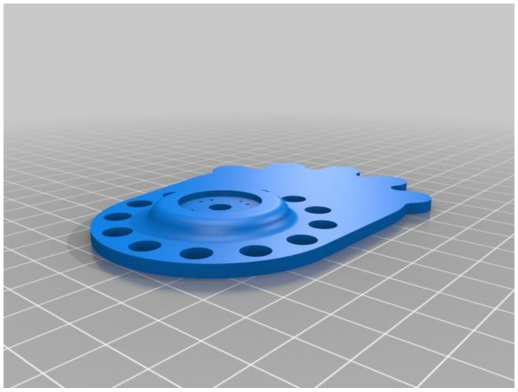


Figure 11 - Servo Base Attachment

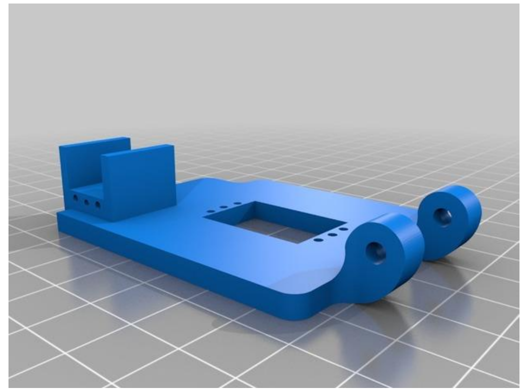


Figure 14 - Hinge to Servo Base Attachment

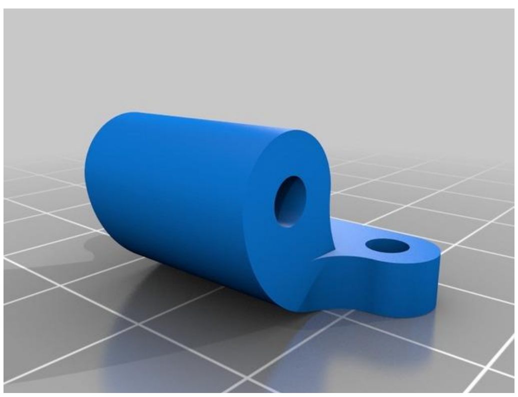


Figure 13 - Pivot Arm

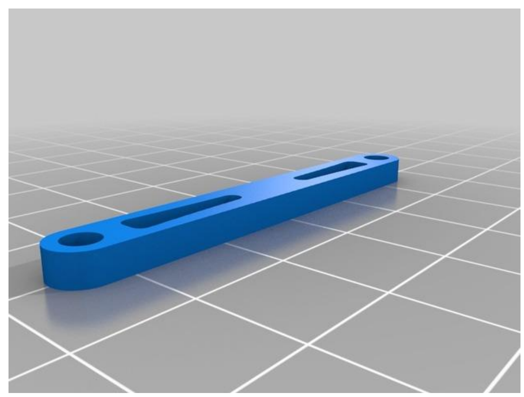


Figure 16 - Servo Attachment to Swing Arm

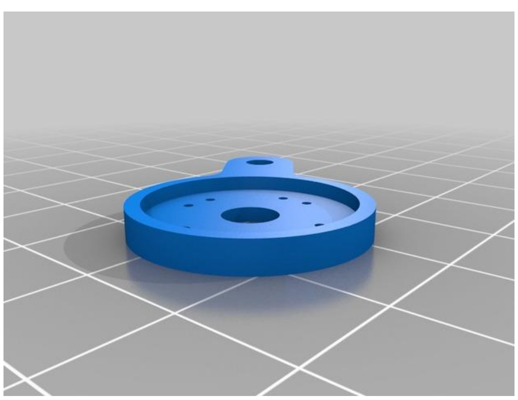
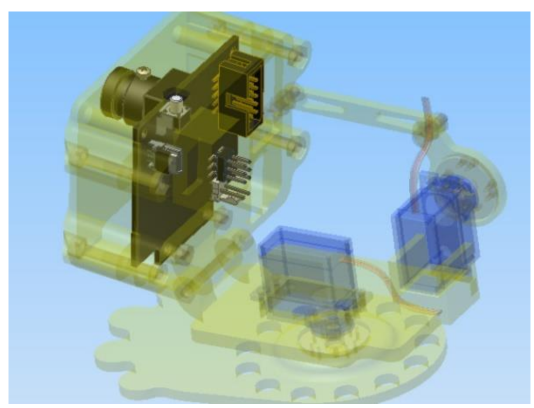


Figure 15 - Pixy's Case Full Assembly



Wireless Start/Stop Remote

Below is the 3D cad model of the enclosure for the wireless start/stop remote. The model was created in Solidworks and 3D printed on Colorado State University Pueblo's Engineering Department 3D printers in PLA Plastic.

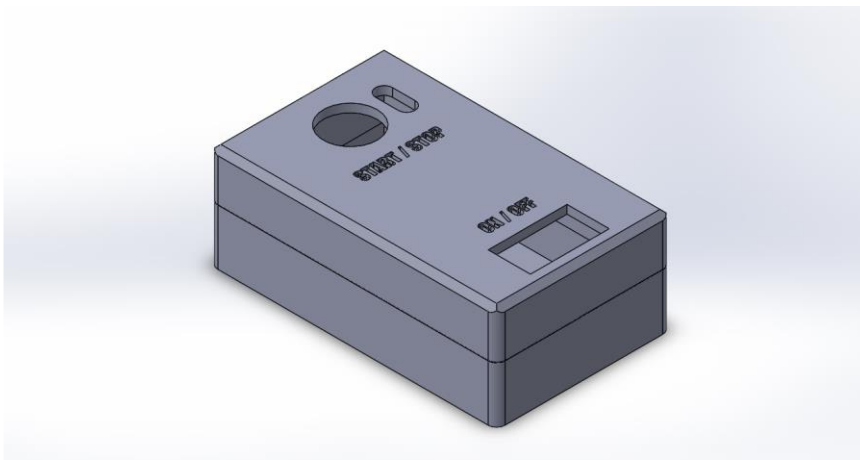


FIGURE 17 - 3D DESIGN OF KILL SWITCH

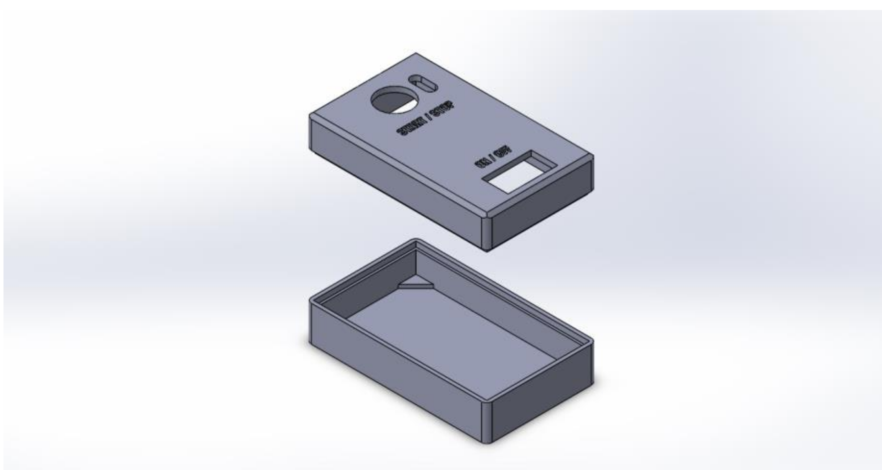


FIGURE 18 - EXPLODED VIEW OF THE KILL SWITCH BOX

Circuit Design

The electronic circuit was originally created on a breadboard and in breadboard view in the fritzing software which then automatically generates a circuit schematic. All that's left to do is clean up the wiring.

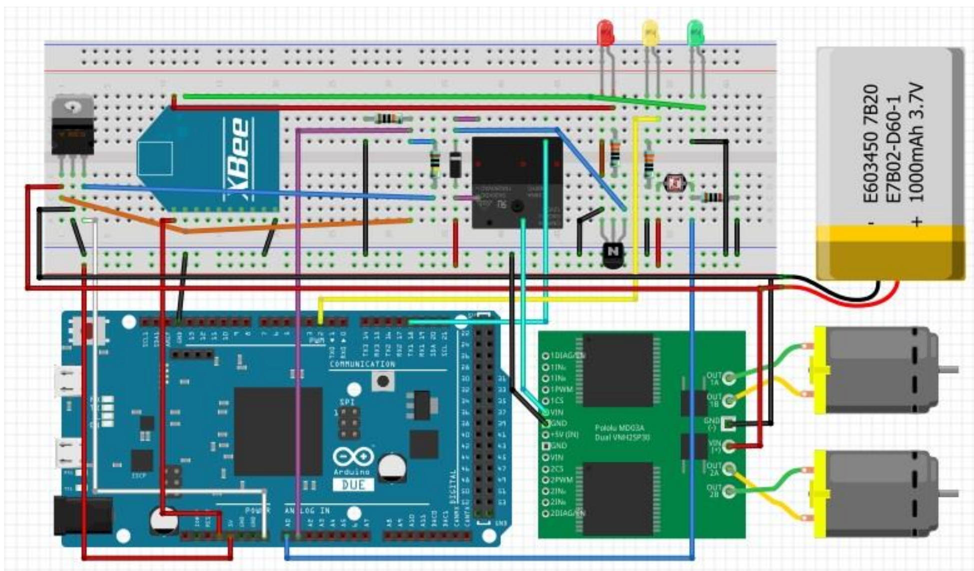


FIGURE 19 - BREADBOARD CIRCUIT

PCB Design

From this circuit the PCB layout shown below was developed by means of photofabrication. First this design is printed onto a sheet of transparency paper. A photoresist coated copper clad board is covered with this design and exposed under a florescent light for ten minutes. Next the board is treated with a developer and all un-exposed resist is dissolved. The board is again chemically treated with ferric chloride which dissolves all exposed copper all remaining copper is the designed circuit.

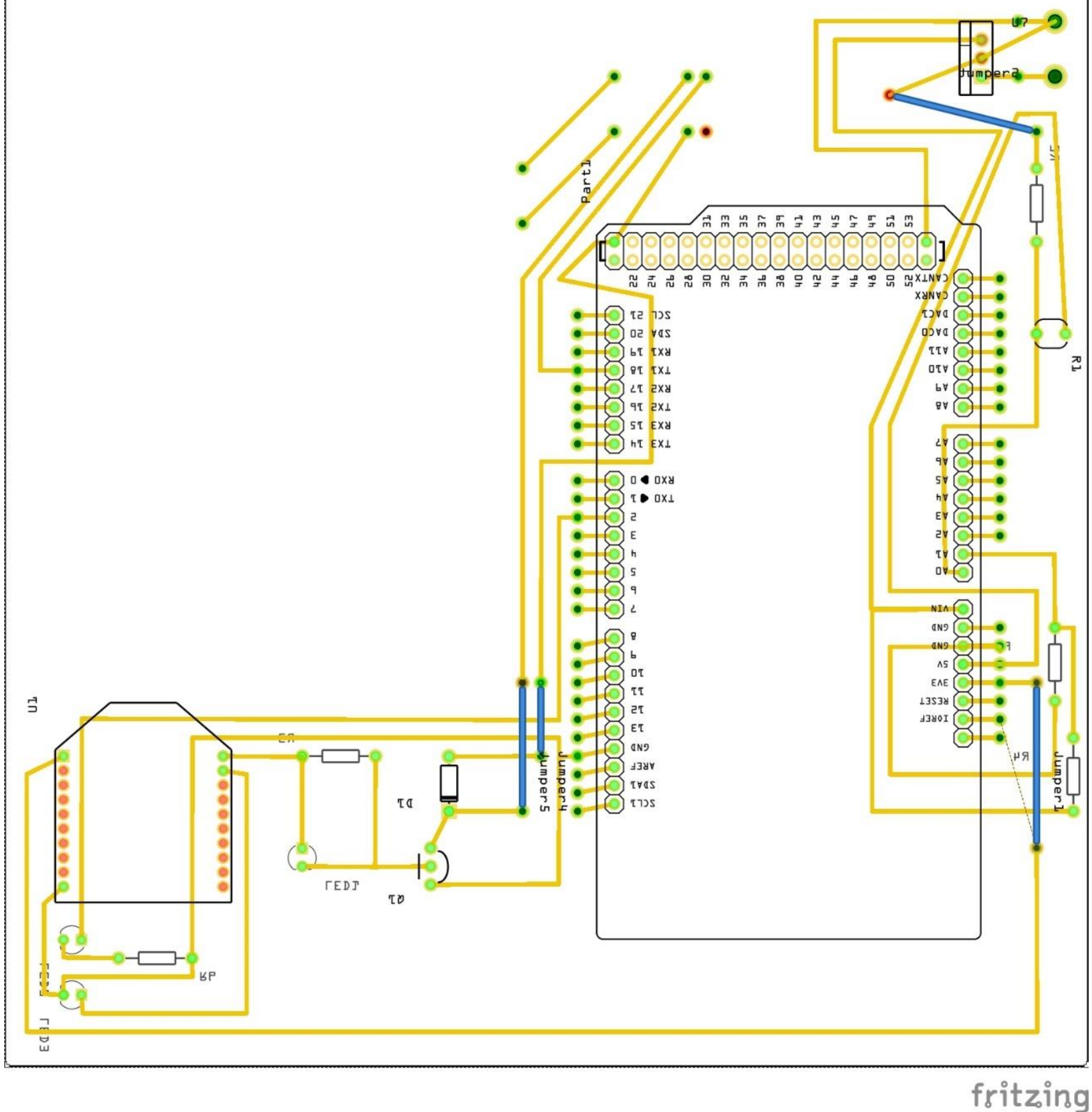


FIGURE 20 - PCB LAYOUT

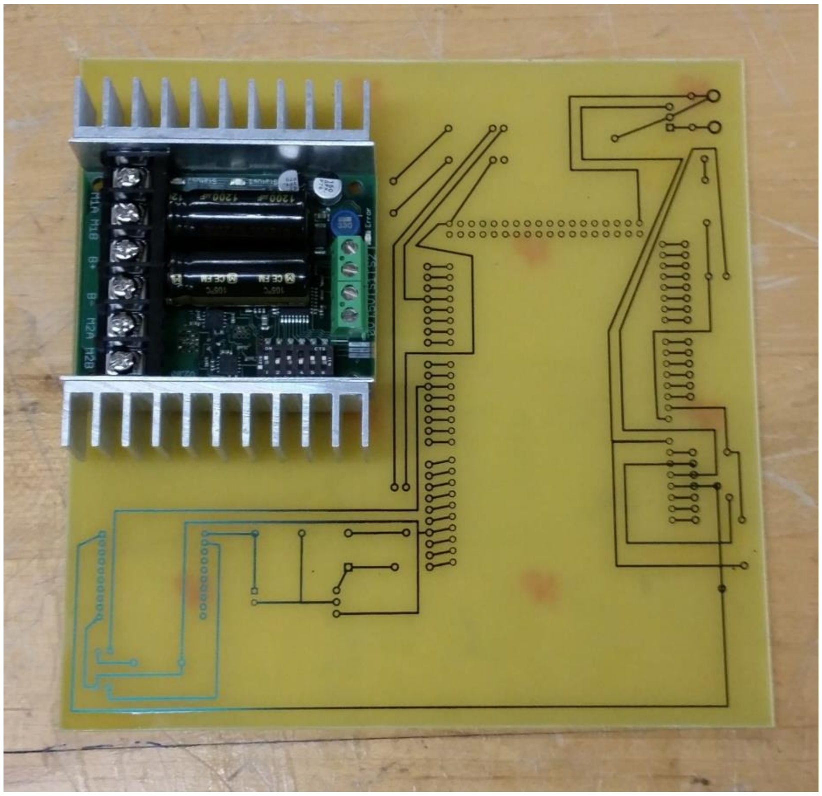


FIGURE 21 - ETCHED PCB

Arduino and Xbee

Component	Input	Output
Input Voltage	Vin	
Photo Resistor	A0	
Voltage Divider	A1	
Low Voltage LED		Digital 2
Motor Driver		TX1

TABLE 1 - ARDUINO DUE PINS USED

Component	Input	Output
Input Voltage	Pin 1	
SPDT Relay		D0
Green LED		D1

TABLE 2 - XBEE S1 PINS USED

End Product Testing and Results

Testing

We are ensuring the project is on track and on schedule by conducting routine testing. We have scheduled certain times to test systems such as battery life, machine power and navigation over a dirt road with a slope of 20 degrees. This testing will be done at the Pueblo Motorsports Park which is open to the public. Our measures of performance will be gauged by comparison of the current parameters to those of previous tests after modifications are made. In doing so we expect to optimize the performance of systems near or within or desired specifications. These parameters are variable criteria such as battery life, instantaneous power, and data acquisition, processing speed, reliability and weight. During the testing procedure we will collect numerical data describing these characteristics. Afterwards we will devise actions steps to take in order to improve or optimize those traits which do not meet desired performance. We will make these modifications and expect to see improvement when testing is conducted again.

Results

The results to our testing were somewhat satisfying. The machine was very responsive as we had hoped for. However, the motion was slightly violent and the vehicle would hastily accelerate and abruptly stop when the signature was detected and lost. We plan on fixing this in the software. First we dialed the speed back to around 70 percent of its maximum also we plan on creating a function for acceleration and deceleration before and after a signature is detected. The robot also succeeded in carrying over 200 lbs up a 20 degree slope we tested this by standing on it while it drove up the wheelchair ramp and the machine worked flawlessly. The robot also succeeded in our final test and that was battery life. It ran for well over an hour on varying terrains such as concrete, dirt and grass. One thing that should be mentioned is that the robot did lose performance as time went on. However, this was as expected as the batteries lose their ability to provide high currents as time passes. For future improvement, we plan to switch to lithium batteries instead.

3. Required Resources

Personal Effort

Estimated

1. Mechanical Team (Jonathan and Lucas) -

   The mechanical team was given the task of creating a frame with a general design in mind. The basic design was agreed upon in the previous semester when the project was proposed with a moderate sized chassis (60” x 32” x 36”). The additional tasks given to the mechanical team were selecting and ordering all mechanical materials, any necessary fabrication of custom components and assembly of all mechanical components to construct the unit. The mechanical team was to calculate the necessary power, slope, and weight to be used in the design, however, Anthony provided many of the calculations necessary in selecting the required components to ensure a functional design. The calculations were verified numerous times to reduce any design errors. Mechanical components included, but were not limited to the bar stock, fasteners, screen netting, bed liner, tires, bearings, wheel assemblies, sheet metal, as well as, any additional hardware needed. The mechanical team divided responsibilities among the two members (Johnathan and Lucas) which included machining, fabricating, and assembly, as well as any additional mechanical work that would be required. After assembly, has been completed the mechanical design would be tested to verify the requirements have been met. Both teams have been given the responsibility and authority to make necessary changes to the design to ensure success. After mechanical responsibilities are finished, the team will evaluate the design in the testing phase as a whole group, rather than individual systems.

2. Electronics Team (Allan and Anthony) –

   The electronics team was tasked with creating and integrating the necessary programming via an Arduino Due to operate the Pixy Camera, which controls the Sabretooth motor driver to obtain optimal performance via independently power motors. The program necessary to track the user and detect obstacles was to be created in a joint effort by Allan and Anthony. They planned on creating a circuit by integrating a Pixy IMU, Lidar sensors, Sabretooth motor driver, Arduino Due to accomplish the given requirements. The code necessary would track a user via color and shape to ensure the desired target is followed using the IMU Pixy, as well as avoiding obstacles. The obstacle detection incorporated in the program would be using lidar sensors to detect obstacles in between the MechaMule and the registered user to ensure safety. The hardware required for the circuit, such as resistors, transistors, and LEDs will also be the responsibility of the electronics team as it is part of the circuit design. Accountability for the responsibilities and authority to make design changes to the electrical components were also agreed upon by the group as a whole.

3. Combined Effort (Allan, Anthony, Jonathan, and Lucas) –

The group is scheduled to meet two to three times week to report current progress and brainstorm about current challenges throughout the semester. The product testing phase will be conducted by the entire group, as all individual and team activities will have been completed. Any potential challenges will be analyzed and potentially solved as a group as well. The final report and presentation materials will also be the responsibility of the entire group.

Actual

1. Mechanical Team (Jonathan and Lucas)

The estimation for the personnel requirements were fairly accurate to the actual work performed in general by the mechanical team. After consulting both Dr. Jaksic and Paul Wallace about various construction methods with various materials, the chosen material to construct the chassis was 30mm aluminum T-Slot extrusion in the size of 10 foot sections; instead of 1 inch aluminum square tubing that would have been attached by welds that would not have been adjustable like the T-Slot was. However due to shipping costs Paul Wallace downsized to 6' sections to eliminate freight shipping costs which were substantially higher than standard shipping which is limited to lengths of 6 feet or less. In creating the MechaMule Paul Wallace's assistance and knowledge has invaluable in fabricating custom parts, and oversights in our design. The chassis dimensions had to be altered due to dynamic constraints when moving through standard doorways and given our power and capacity constraints. The final proposed dimensions of the MechaMule were to be 48"in length x 28" in width x 26" in height. However, due to blade width during machining operations, and various measurement errors the actual dimensions of the completed assembly is 47.75" in length x 27.75"in width x 25" in height. The final dimensions are sufficient in cargo space and can also accomplish the required maneuverability to navigate through doors, trails. Etc. The unit could be loaded in a truck by two averaged sized men as the design does not exceed space requirements of a standard truck bed, and is 150lb when empty. The robot may also be loaded in a more automated fashion via ramps onto a trailer or truck; granted the ramps don't exceed 20°slope when loaded. Other responsibilities that had changed included the mechanical team being responsible for the motors and batteries, as the individual responsible for electronics components being ordered through the university form deemed the fore mentioned components mechanical. Johnathan and Lucas completed an order form with the necessary materials and submitted it to Paul Wallace after being approved by Dr. Jaksic. After the order was placed the mechanical team had to meet various times with

Paul to add the additional components rejected by the individual in charge of ordering electronics, as well as, to change certain suppliers that could take additional shipping time that we had not planned for. Paul addressed and solved all challenges encountered when ordering from remaining suppliers. After receiving all of the mechanical supplies, it was determined that additional 30 mm Extrusion had to be ordered due to an error in communication. Paul ordered five segments of six-foot 30mm T-Slot extrusion, when our order called for 5 segments of ten-foot 30 mm T-Slot extrusion as a result of confusion about our chassis design. The additional order was for two segments of the six-foot 30mm extrusion as well as and additional twenty-five fasteners. The additional extrusion was used to create a rigid frame with vertical supports and a wheel well. After the initial frame was completed, Paul Wallace fabricated various custom components including a keyed drive axel and custom sprocket mounts. The mechanical team then had sheet metal cut down to six by a locals sheet metal shop, which was further machined by Paul Wallace to achieve the desired dimensions. After all existing parts had been installed the decision to reinforce the frame was made and additional supplies were needed. The supplies needed were two more six-foot segments of the extrusion, thirty additional fasteners, sixty t-nuts and four additional corner supports. The mechanical team completed assembly with the additional parts with the exception of mounting the motors. The motor mount dimensions were determined with the assistance of Paul, which were later also fabricated by Paul as well. The mechanical assembly was complete after attaching the motor and chain assembly; in which the chain had to be altered to the correct size. The side panels and top panel were later attached and the design was introduced to the testing phase. Throughout the semester both Johnathan and Lucas worked on CAD drawing in Solidworks. Later in the semester Lucas was tasked with creating a full assembly of the MechaMule; including the majority of components with the exception of chains side paneling, electronics box, and pixy enclosure. Jonathan was the primary designer behind the poster with some assistance in layout from Lucas.

2. Electrical Team (Allan and Anthony) –

Much like the mechanical team the general responsibilities of the team followed the estimated expectations fairly well. The electrical team invested a large amount of time initially into understanding the capabilities and limits of the IMU Pixy. After gaining a general understanding of the various functions and capabilities the team dug deeper into the ability of the pixy to be used in for desired requirements for the MechaMule. After analyzing various methods, an example from a previously cited source was used to base the code used to track objects via color and shape. The code was altered and added to by the electrical team which was then tested on a robot previously created by a space grant team which Anthony participated in. The prototype functioned as desired with minimal modifications in an indoor environment, but encountered issues when changing lighting conditions.

The electrical team placed the order with Dr. Unglaub for all the electrical components, but had to update the order multiple times; due to the restrictions placed upon them which limited the suppliers available for use. After the order had been placed, Allan worked primarily on the IMU to ensure proper functionality; which was also the same model that had been ordered. The pixy he owned personally, had been purchased over winter break to work with before the supplies purchased through the school had been ordered, to obtain additional time if needed. While working with the prototype the electrical team also created an emergency kill-switch controlled with the wireless network that the x-bee transceiver/receiver operates on. The kill-switch's circuit and functionality was verified on the prototype; the necessary code and design was created by Anthony. However, both Allan and Anthony worked twenty plus hours troubleshooting the circuit on the PCB board that Anthony created for the final assembly.

In the experimentation on the prototype, a crucial part of creating the signature was data collected via the IMU to adapt to various lighting conditions. The lighting conditions which were adjusted using a program provided by Charm Labs, aptly named pixymon, functioned much better inside, due to the wash-out and/or eclipse effect caused by the sun which overpowered the target signature in extremely bright/sunny conditions. The signature would become lost or washed-out when in bright light; the detected colors under additional light would differ from the stored signature due to the difference in brightness. After many trials and attempts to mitigate outside light, the electrical team chose to work with an infrared signature which in theory should have fewer false reading and/or errors in comparison to a color signature. The infrared signature method was investigated initially by Anthony with further analysis and experiments conducted by Allan. After initial experiments and analysis, it was determined a standard infrared led was not strong enough to mitigate the signature created by the sun. The solution created by the electrical team could be implemented in two ways; a higher-powered infrared led target attached or carried by the intended user or an infrared led capable of producing the necessary power that a signature at desired range could be reflected back to the IMU using a light-weight reflective object attached/worn by target user. The two-proposed solution still encountered errors under certain lighting conditions (i.e. dusk and dawn), so additional analysis was done to find a better solution. Allan was able to adjust various settings in the program pixymon after additional research, (pixymon which store signatures of the IMU) and was able to achieve satisfactory performance under various lighting conditions. Anthony created a PCB (printed Circuit Board) for the required circuit and etched it himself. Additionally, he created the enclosure to house the electronic components (i.e. motor driver, microcontroller, circuitry, etc.) while Allan created a 3d printed enclosure for the pixy assembly that Lucas assisted in printing as he already worked in the 3d printing lab. After all troubleshooting was completed, the electronic components were mounted to test full functionality.

3. Combined Effort (Allan, Anthony, Jonathan, and Lucas) –

After assembly was completed of the MechaMule moved forward to the testing phase. Any challenges encountered, were analyzed and solved in a timely manner by the entire team. In this phase, we had to verify all proposed requirements were fulfilled and functioned in a satisfactory manner. After the device functionality was confirmed we focused on reporting our results. The responsibilities of the presentation and report were equally divided among team members and were frequently discussed to ensure the team was on the same page. Team members also assisted other team members frequently throughout the design and manufacturing process to provide support and additional resources to ensure the team's success.

Financial Requirements

Estimated

The following table was created last semester based on the proposed design. The Table includes both mechanical and electrical components. This estimate was fairly accurate to the actual required materials with the exception of few items that were removed as well as, additional extrusion and hardware that was required.

Parts	Base Price
Aluminum Sheet	$52.26
Aluminum Rod	$15.00
Plastic Sheet	$50.00
Mesh Screen	$30.00
(4) Tires	$168.00
(2) Motors	$149.98
Battery	$300.00
#35 Chain	$20.63
Pixy with Pan/Tilt Mount	$120.00
Lidar Sensor	$150.00
(2) Ultrasonic Sensors	$60.00
(2) Infrared Sensors	$50.00
Microprocessor	$40.99
Bed Liner Coating	$400.00
Water Sealing Material	$11.94
(2) 58T Sprockets	$51.50
Raw Aluminum	$250.00
Motor Driver	$119.99
Hardware and Wires	$250.00
IMU	$14.95
3.3 V Regulator	$0.95
Wires, Headers	$10.00
Shipping/Tax/Miscellaneous Charges	$300.00
Total	$2616.19

TABLE 3 - ESTIMATED COST

Actual

The following table includes all parts used in the assembly of the MechaMule. Unused sensors, mesh screen, and bed liner from the previous table were removed. This was due to design changes after the construction of the unit began. Also, we had to order additional T-slot extrusion and hardware due to fore mentioned confusion. The actual cost of the unit when the unnecessary components were removed dropped substantially and was below our initially proposed budget.

Parts	Elec./Mech.	Quantity	Base Price
T slotted aluminum extrusion (6 ft segments)	M	9	$216.00
5058 Aluminum sheet	M	1	$50.00
Rear wheels	M	2	$39.98
Front wheels	M	2	$94.76
Motors	E	2	$149.98
Battery	E	2	$200.00
Bicycle chain	M	1	$22.00
Arduino Uno	E	1	$24.95
Arduino Due	E	1	$40.99
Water Sealing Material	M	1	$18.79
Pixy camera	E	1	$69.00
Pan tilt mount	E	1	$39.00
Bearings	M	2	$35.60
48T Sprocket	M	2	$49.90
Motor Driver	E	1	$119.99
Jumper Wires	E	1	$1.95
IMU	E	1	$14.95
3.3 V regulator	E	1	$1.95
5v regulator	E	1	$0.87
12v regulator	E	1	$1.50
Hardware (includes fasteners and supports)	M	1	$450
1/16" Plywood	M	1	$12
IR LEDs	E	1	$25
Circuit Etching Kit	E	1	$50
Transparency Paper	M	1	$47.25
9V Barrel Jack	E	2	$6.99
Hinges	M	1	$1.98

Magnets	M	1	$2.47
Acrylic Sheet	M	1	$10.28
Electrical Wire	E	1	$7.37
Colored T-shirts	M	1	$10.79
Total			$1,816.27

TABLE 4 - ACTUAL COST

Schedules

The project we are working on is going to be divided into two areas of focus, mechanical and electrical. Allan Quillen and Anthony Olvera are going to work primarily on the electrical side, doing programming, sensor integration, and control systems. While Jonathan Widjaja and Lucas Mendicello will primarily work on the mechanical design, which includes choosing the materials, fabricating, and installation on our prototype for proof of concept. Individuals will be responsible for certain tasks within their subgroup, such as 3D-printing tasks, machining parts of the frame, creating program codes, etc. However, all decisions will be made as a group even though subdivisions have been made. This will aid us in the design as each team member will be able to put forward their own personal experience. All potential ideas or design will be recorded in either the hardcopy of our logbook or the digital copy depending on the material included. The project has been divided into essentially 4 tasks.

Project Planning (Task One) – Evaluating problems and solutions, Potential design, and Gathering information regarding materials, parts and processes (i.e. prices and calculation requirements).

Implementing the potential design (Task Two) - fabricate or machine parts and verify programming is sufficient to meet the design requirements and assemble the prototype as proof of concept.

Testing the potential design (Task Three) - Test various parameters of design confinements, prove that the basic concepts of the design are fulfilled, and analyze any errors or “bugs” found. Edit or improvise any issues to ensure functional design. Run finalized prototype through simulation to verify functionality.

Pitch and/or Propose Design (Task Four) – Solve or improvise any problems, then create the proposal including all working features to demonstrate the capabilities of the design. Create potential counter-arguments for any issues encountered.

Schedule

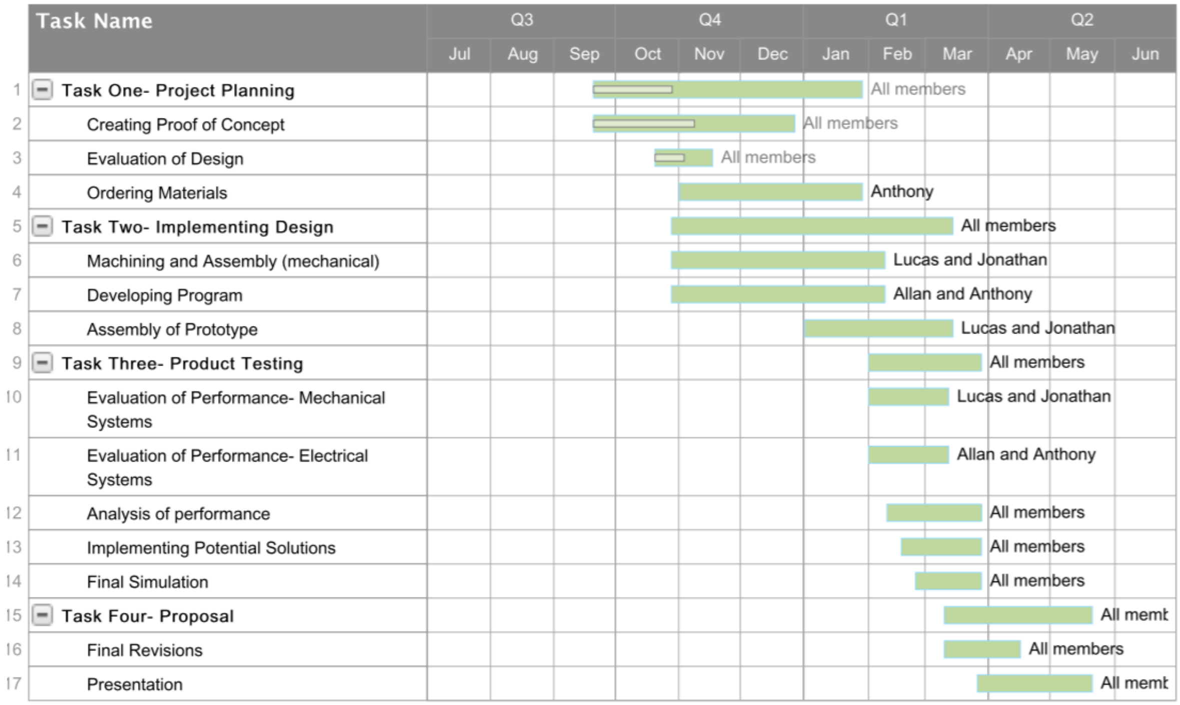


Figure 22 - Gantt chart

Task	Start Date	Duration	End Date
Part 1- Project planning	Aug-16	4 months	Nov-16
Creating Proof of concept	Sep-16	2 months	Nov-16
(i.e. programming smaller robot,			
verify code, potential issues/errors)			
Ordering materials	Oct-16	1 month	30-Oct-16
Part 2 - Implementing	Nov-16	2 months	Jan-17
Potential Design			
Constructing Prototype	1-Dec-16	1 month	Feb-17
(I.e. machining, assembly, etc.)			
Part 3 – Testing	Feb-17	1 month	Mar-17
evaluate and analyze potential	1-Feb-17	1 month	1-Mar-17
problems and solutions			
Simulation	1-Mar-17	1 month	31-Mar-17
Part 4 - Final Proposal and	1-Apr-17	1 month	1-May-17
Presentation of Design			

TABLE 5 - ESTIMATED SCHEDULE

Project Tracking Procedures

As described earlier we will track our progress by keeping a Gantt chart available to all members of the group. This way we will be able to organize project activity in the most effective and time efficient manner while keeping the ability to see our progress and if we are on, ahead or behind schedule. By keeping such a chart available we can make critical decisions before it is too late, keep activities on track and to produce a finished product on or before our deadline. Each member of the group will be assigned to certain activities so we can hold each other accountable to staying on or ahead of schedule. The Gantt chart will be organized such that critical actions are conducted first and so one late but important action cannot and will not postpone the entire project.

4. Closure Materials

Life Long Learning

During this project, we encountered various problems in which we needed to obtain knowledge we did not gain in any of our required course work. Among these problems was programming. We did learn a very limited amount of Arduino programming in the EN 362 class, however this project required coding with a P and PD control algorithm to control robot speed, which ties control systems to programming. Programming at this level was challenging and required much time and research. We learned about different data types, pointers and arrays from forums and on the Arduino website. Through trial and error, we were able to make our code functional for our purposes. Another problem which required extra knowledge and outside research was work with image processing and a vision system. Our project requires a robot to track and follow objects regardless of changing lighting conditions so that it works indoors and outdoors, on a cloudy day or a sunny day. So, we needed to obtain knowledge and skill with the pixy programming software. The software is called pixymon, which we have never used before. Experimenting with this software allowed us to learn much about optics and how cameras actually work. Also, how to fine tune parameters such as exposure, white balance, and saturation levels such that the camera can recognize certain signatures in the most effective manner in these changing lighting conditions. we obtained our knowledge through online research and through email conversation to the Pixy engineers at Charm Labs. They provided us with documentation and after reading and experimenting we were able to tune the camera to work for our purposes. Next we learned how to design and develop printed circuit boards which is a unique process that we have never had experience with. This was needed to be done as we wanted to develop a permanent circuit so that wires could not come loose and cause the robot to malfunction. Knowledge on designing a circuit that can fit on a single layer of copper, transferring the image and following the photo fabrication etching process were all things we had learned through online research and following instruction manuals that came with the etching kit. Using this instruction, we developed a functional PCB. And lastly we learned much about mechanical structure and design. Our project required us to design and develop a sturdy mechanical frame, an axle, and a drive mechanism. During the process of creating this project. We received a lot of helpful advice from Paul Wallace about how certain parts can be machined to develop the mechanisms previously mentioned. We then applied wat we learned and successfully developed a functional locomotive robot. By seeking and obtaining this information, this project was a learning experience for all of us. We believe that this experience was beneficial and will help us engage in a lifetime of learning from now throughout our future careers as engineers.

Impacts of Proposed Solution

We believe the mechatronic mule could impact the world in many ways. In a global context. It is a product that could be used anywhere in the world for a variety of purposes. First we believe it has military applications which could assist not only the United States military but also allied forces around the world. A larger and more ruggedly engineered version of this machine could assist soldiers in carrying their gear for them reducing the burden for them to perform in critical situations, also enhance man power and military efficiency. If the mechatronic mule were developed for allied militaries it could enhance foreign relations all while helping to keep the world safe from enemy threats. Another global impact of the mechatronic mule is an astronaut assisted robot. The machine would be much more advanced but could follow the same principle so that it could assist astronauts in developing a habitat on mars one day, furthering mankind in space exploration. In an economic context, custom engineered versions of the mechatronic mule could save money for companies all around the world. Examples include but are not limited to automated food or package delivery, worker assisted transportation vehicles in factories or warehousing and construction robots which more easily transport materials. The mechatronic mule could replace certain human workers which would eliminate human errors and thus create safer working environments. Such a replacement could also reduce the risk of asset losses in injury related lawsuits. The mechatronic mule is also beneficial to the environment. Because it uses electric DC motors instead of fossil fuels it does not emit any carbon. Perhaps the mechatronic mule will replace cars in the future and thus drastically reduce pollution. Lastly in a societal context the mechatronic mule should be available to consumers as a personal caddy. The mule could be engineered in many different trims to accommodate golfers, hikers, shoppers or anyone who has the need to move their gear more easily for whatever their intended purpose.

Sustainability

As we head into the future humanity sees the need to automate more and more devices and surely people have the need to transport items and will continue to have that need. We see this project as the beginning of an emerging technology that will be with us indefinitely. Transportation has been an essential need for humanity since antiquity. As time goes on we find better solutions and ways of transporting ourselves as well as our items. For example the automobile was invented in 1885 and has been with us ever since. This invention proved to be one of the most sustainable products in its design and has only evolved more and more over the years. So we believe that the mechatronic mule is similar to the automobile in a certain sense. The idea and initial design are in its infancy, however the idea and the concept will be sustained for many years as we head into the future. In a physical sense, as of now the mechatronic mule is expected to have a lifetime of around five years, however given more time, and improved use of resources the mechatronic mule could become a machine that lasts decades or more. Just like the automobile with regular care and service it could last longer than we would ever expect.

Project Evaluation

Milestone Accomplishments

The first project milestone was the development of a functional code which enables object tracking and following on our prototype model. To evaluate this milestone effectively we gauged the robot's abilities. The evaluation criteria was as follows. Can the robot stop, go and follow a person at a set distance? Can the robot turn corners? Can the robot match the speed of its user? Can it carry the desired weight for the desired run time? Our next milestone was the development of a 3D CAD model of the final full sized vehicle. Criteria for evaluation was determined on how well the model matches or sketches in dimensions and in shape. The next milestone was ordering parts and equipment. We were posed with the questions of, will the parts arrive in a timely fashion? Are we within budget? Next was machined parts and fabrication. We were posed with another set of questions do parts meet size, tolerance and weight requirements? The answer was yes after much hard work and rework as well. Lastly was the final assembly and testing.

Commercialization

Intellectual Property Considerations

The actual idea of a vehicle following you and carrying a payload came from Allan Quillen. The design of the frame was done by Lucas Mendicello, Anthony Olvera, and Jonathan Widjaja. Portions of the code used in our project was courtesy of https://learn.adafruit.com/pixy-pet-robot-color-vision-follower-using-pixycam/overview . This code included the basic math for proportional and derivative control and also functions that call to pixy. Anthony Olvera replaced the functions that call to a different motor control to fit our motor controller. Other modifications to the code are included in the comments of the code. The Idea to use infrared to track our object due to variable lighting conditions is credited to Anthony Olvera, however the actual infrared tracking algorithm is credited to irlock.com. The actual device and every item included on our purchase is owned by Colorado State University of Pueblo because they funded the project.

Commercialization Considerations

We do not have any Commercialization Considerations for this device because it is not our device to commercialize.

Recommendations and Improvement

For any future group considering on improving or designing a similar device to our mechatronic mule please read the following suggestions.

- Implement side and back end detection (Infrared or Ultrasonic)
- Improved Battery Efficiency (Lithium Polymer)
- Added suspension for more rough terrain
- Upgraded image processing or tracking
- Increased carrying capacity
- Smaller, durable, and sleeker design.
- Four-wheel motorization

Project Team Information

Names: Allan Quillen, Anthony Olvera, Jonathan Widjaja, Lucas Mendicello

Faculty Advisors: Dr. Jaksic, Dr. Depalma, and Paul Wallace.

Closing Summary

Our main goal was to develop an automated vehicle that tracks you within about a one-meter distance, avoid static and dynamic obstructions, and be able to carry a payload of up to 200 lbs. In order to track the user, we used an infrared detection system that allows the robot to follow its user wearing the appropriate infrared The project was built on a budget of $2500 which should be plenty assuming our cost estimation did not exceed $2000. The ability to carry a relatively small payload of whatever the user can think of was intended to help them transport more than what is humanly possible. This is a “small” scale prototype of what could be conceived as an automated guided vehicle that could carry over 1000lbs for more industrial or military means.

Works Cited

References

UNFOLD YOUR NEXT JOURNEY. (n.d.). Retrieved October 13, 2016, from http://gethover.com/

Dent, S. (n.d.). How I turned my Xbox's Kinect into a wondrous motion-capture device. Retrieved October 18, 2016, from https://www.engadget.com/2015/03/08/using-the-kinect-for-motion-capture/

How does Microsoft's Kinect work from a technology ... (n.d.). Retrieved October 20, 2016, from https://www.quora.com/How-does-Microsofts-Kinect-work-from-a-technology-standpoint

MacCormick, J. (n.d.). How does the Kinect work? Retrieved October 31, 2016, from http://users.dickinson.edu/~jmac/selected-talks/kinect.pdf

Infrared detection. https://irlock.com/

Image processing aka Pixy. http://www.cmucam.org/

Pixy pet code and resources. https://learn.adafruit.com/pixy-pet-robot-color-vision-follower-using-pixycam?view=all

Motor controller specs and libraries. https://www.dimensionengineering.com/products/sabertooth2x25

Pixy and Arduino Code

https://learn.adafruit.com/pixy-pet-robot-color-vision-follower-using-pixycam/overview

5. Appendices

Appendix A – Drawings

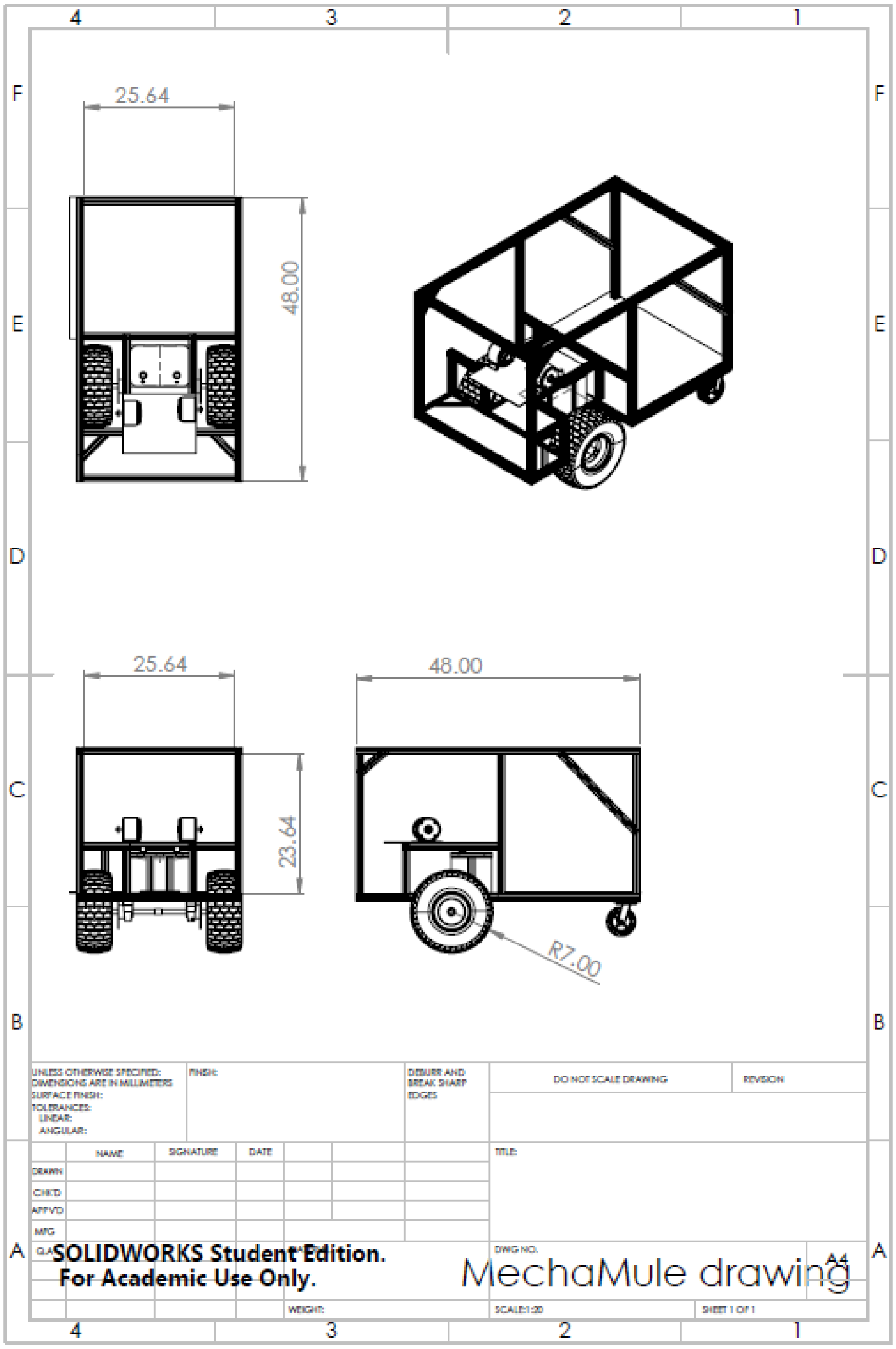


FIGURE 23 - FRAME AND CHASSIS

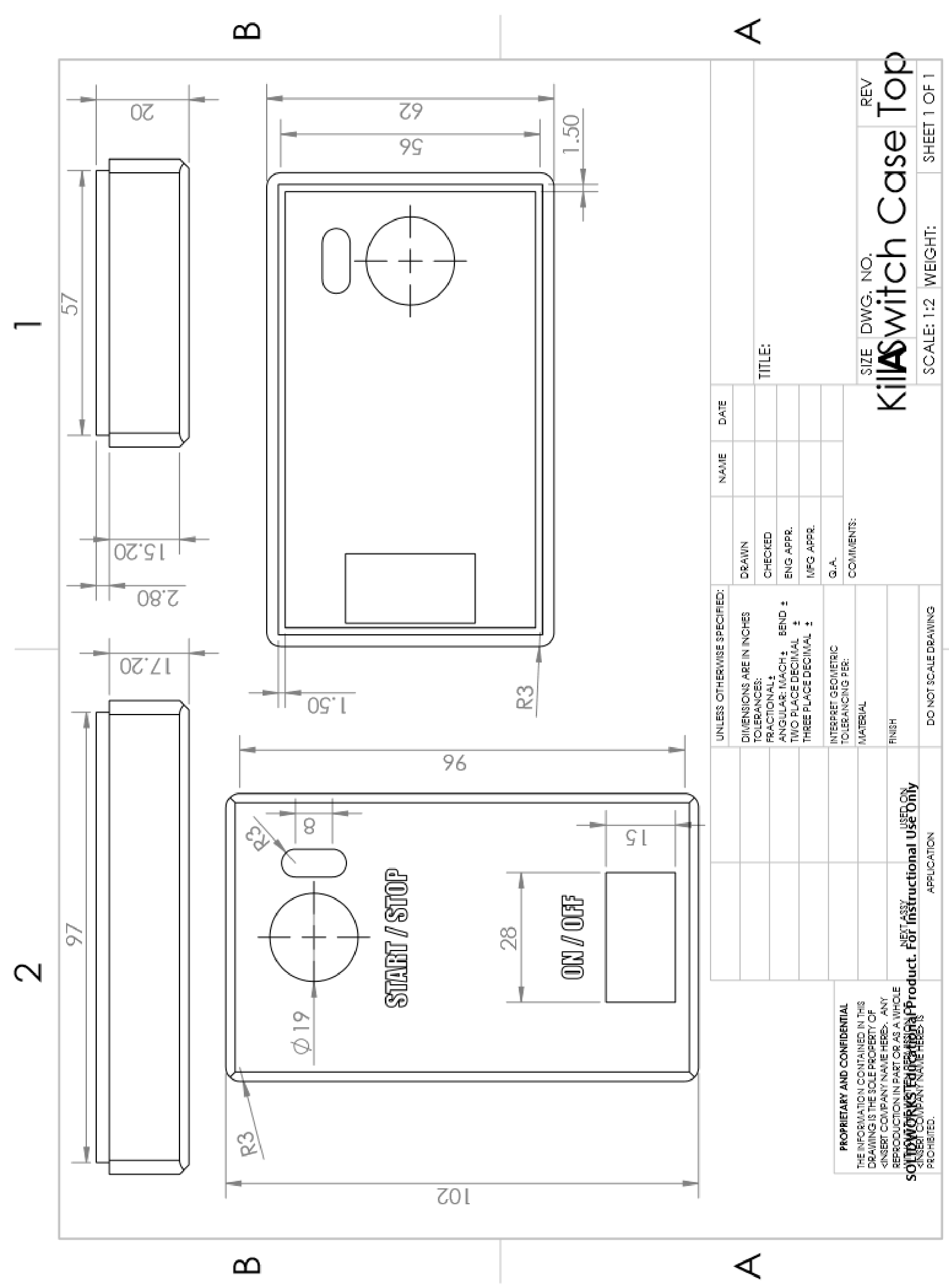


FIGURE 24 - START/STOP REMOTE TOP

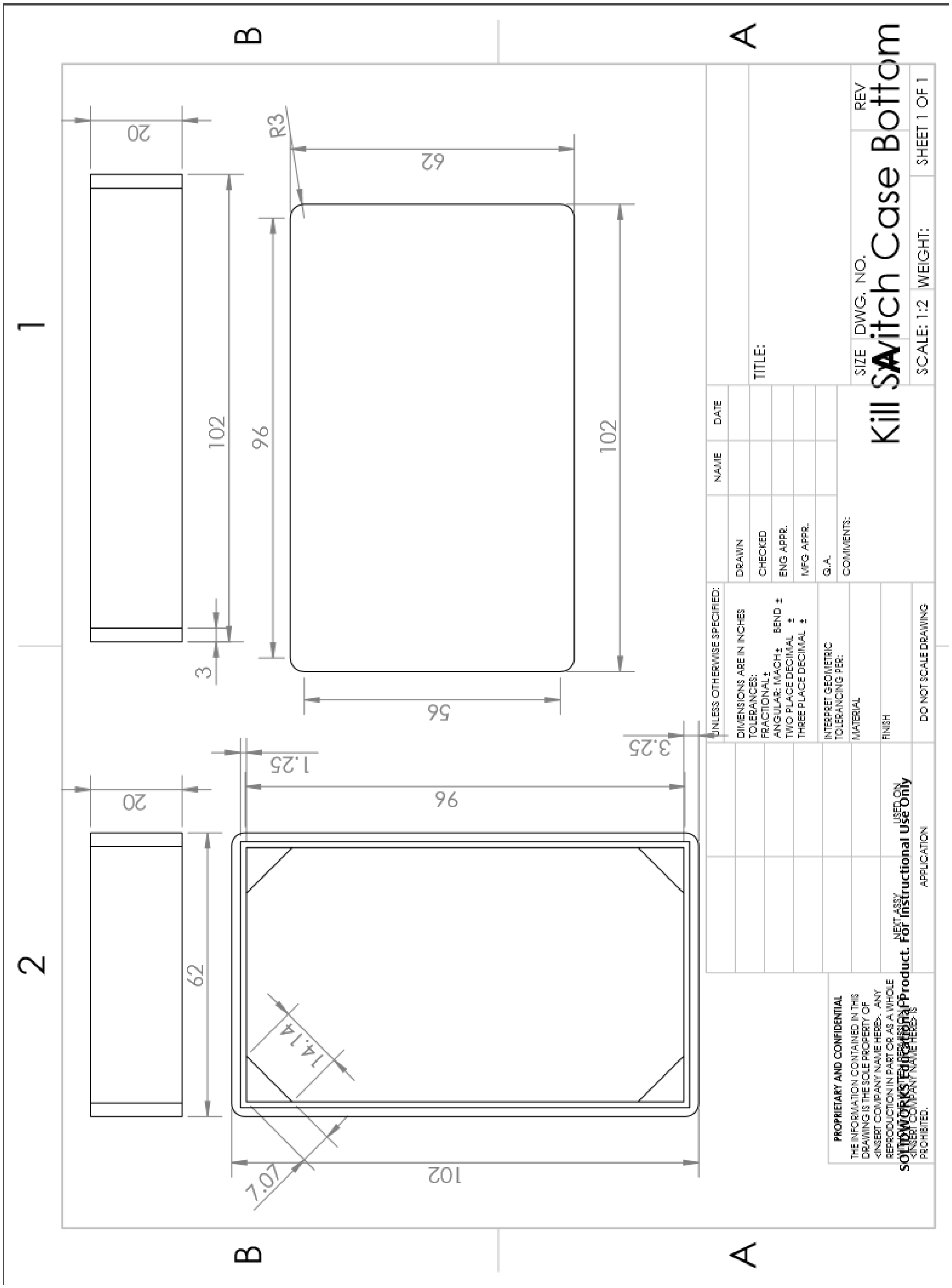


FIGURE 25 - START/STOP REMOTE BOTTOM

Appendix B – Schematic

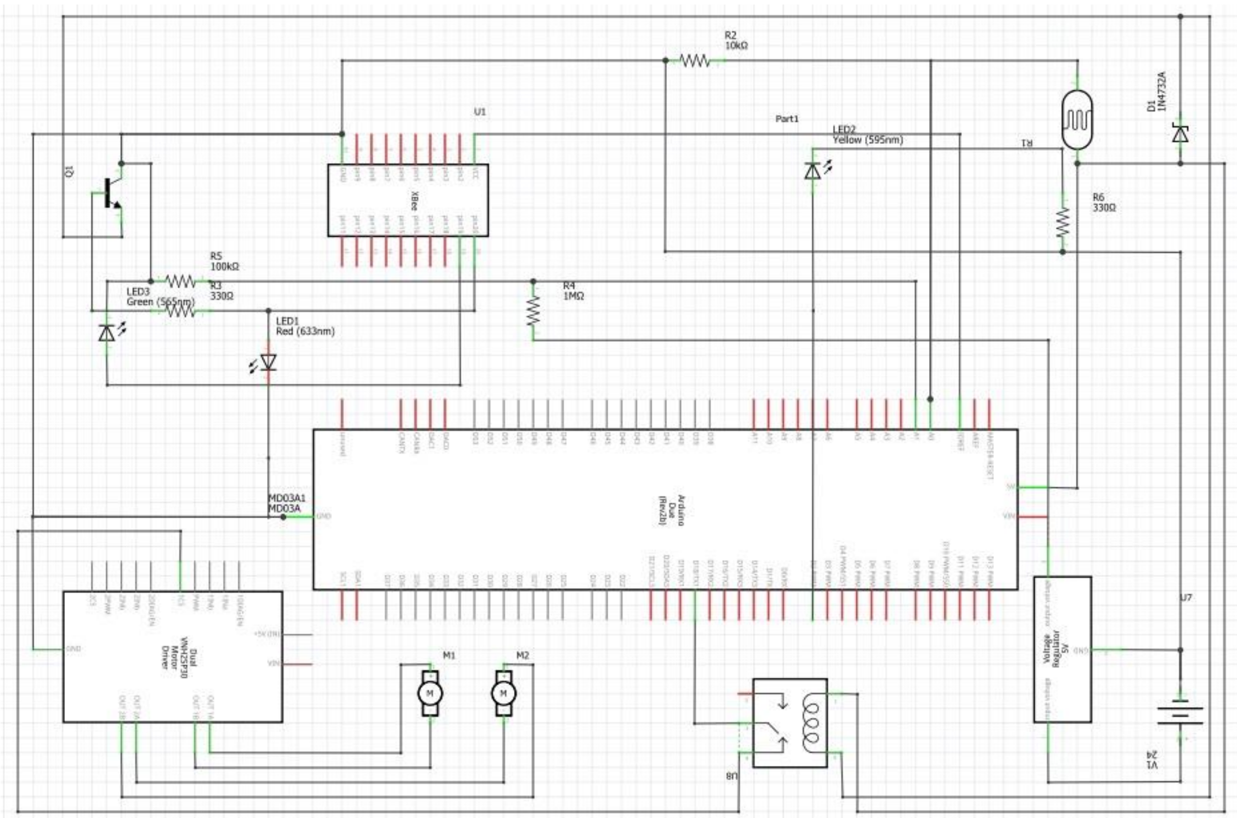


FIGURE 26 - SCHEMATIC

Appendix C – Code

```
//==========================================================================
//
//  Pixy Pet Robot
//
//  Mechatronic Mule
//
//  Adafruit invests time and resources providing this open source code,
//  please support Adafruit and open-source hardware by purchasing
//  products from Adafruit!
//
// Written by: Bill Earl for Adafruit Industries
// Modified by: Anthony Olvera for Colorado State University-Pueblo
//
//==========================================================================
// begin license header
//
// All Pixy Pet source code is provided under the terms of the
// GNU General Public License v2 (http://www.gnu.org/licenses/gpl-2.0.html).
//
// end license header
//
//==========================================================================
//
// Portions of this code are derived from the Pixy CMUcam5 pantilt example code.
//
//==========================================================================
#include <SPI.h>
#include <Pixy.h>
```

```
#include <PixySPI_SS.h>//modified
#include <stdint.h>
//#include <unistd.h>
//#include <pixydefs.h>
#include <SabertoothSimplified.h> // modified
SabertoothSimplified ST; //modified
//#include <ZumoMotors.h> modified

#define X_CENTER    160L
#define Y_CENTER    100L
#define RCS_MIN_POS     0L
#define RCS_MAX_POS     1000L
#define RCS_CENTER_POS  ((RCS_MAX_POS-RCS_MIN_POS)/2)
//--------------------------------------
// Servo Loop Class
// A Proportional/Derivative feedback
// loop for pan/tilt servo tracking of
// blocks.
// (Based on Pixy CMUcam5 example code)
//--------------------------------------
int lowVPin = 2;
class ServoLoop
{
public:
  ServoLoop(int32_t proportionalGain, int32_t derivativeGain);
  void update(int32_t error);
  int32_t m_pos;
  int32_t m_prevError;
  int32_t m_proportionalGain;
  int32_t m_derivativeGain;
```

```
};
// ServoLoop Constructor
ServoLoop::ServoLoop(int32_t proportionalGain, int32_t derivativeGain)
{
  m_pos = RCS_CENTER_POS;
  m_proportionalGain = proportionalGain;
  m_derivativeGain = derivativeGain;
  m_prevError = 0x80000000L;
}

// ServoLoop Update
// Calculates new output based on the measured
// error and the current state.
void ServoLoop::update(int32_t error)
{
  long int velocity;
  char buf[32];
  if (m_prevError!=0x80000000)
  {
    velocity = (error*m_proportionalGain + (error - m_prevError)*m_derivativeGain)>>10;
    m_pos += velocity;
    if (m_pos>RCS_MAX_POS)
    {
      m_pos = RCS_MAX_POS;
    }
    else if (m_pos<RCS_MIN_POS)
    {
      m_pos = RCS_MIN_POS;
    }
  }
```

```
  m_prevError = error;
}
// End Servo Loop Class
//--------------------------------------
Pixy pixy;  // Declare the camera object
ServoLoop panLoop(200, 200);  // Servo loop for pan
ServoLoop tiltLoop(150, 200); // Servo loop for tilt
//--------------------------------------
// Setup - runs once at startup
//--------------------------------------
void setup()
{
  pinMode(lowVPin, OUTPUT);          // Declare low voltage indicator pin as an output
  SabertoothTXPinSerial.begin(9600); // modified for sabertooth motor driver
  Serial.begin(9600);
  Serial.print("Starting...\n");
  pixy.init();
}
uint32_t lastBlockTime = 0;
//--------------------------------------
// Main loop - runs continuously after setup
//--------------------------------------
void loop()
{
  int sensorValue = analogRead(A0); // Read signal from analog input
  float voltage = 11.3636363636*(sensorValue*(5.0/1023.0)); // Convert Signal to voltage
  if (voltage < 10) // Iluminate low voltage LED if voltage is below desired value
  {
  digitalWrite (lowVPin, HIGH);
  }
```

```
  else
  {
    digitalWrite (lowVPin, LOW);
  }
  uint16_t blocks;
  blocks = pixy.getBlocks();
  // If we have blocks in sight, track and follow them
  if (blocks)
  {
    int trackedBlock = TrackBlock(blocks);
    FollowBlock(trackedBlock);
    lastBlockTime = millis();
  }
  else if (millis() - lastBlockTime > 100)
  {
    //motors.setLeftSpeed(0);
    //motors.setRightSpeed(0);
    ST.motor(1, 0); //modified for sabertooth
    ST.motor(2, 0); //modified for sabertooth
    ScanForBlocks();
  }
}
int oldX, oldY, oldSignature;
//---------------------------------------
// Track blocks via the Pixy pan/tilt mech
// (based in part on Pixy CMUcam5 pantilt example)
//---------------------------------------
int TrackBlock(int blockCount)
{
  int trackedBlock = 0;
```

```
  long maxSize = 0;
  Serial.print("blocks =");
  Serial.println(blockCount);
  for (int i = 0; i < blockCount; i++)
  {
    if ((oldSignature == 0) || (pixy.blocks[i].signature == oldSignature))
    {
      long newSize = pixy.blocks[i].height * pixy.blocks[i].width;
      if (newSize > maxSize)
      {
        trackedBlock = i;
        maxSize = newSize;
      }
    }
  }
  int32_t panError = X_CENTER - pixy.blocks[trackedBlock].x;
  int32_t tiltError = pixy.blocks[trackedBlock].y - Y_CENTER;
  panLoop.update(panError);
  tiltLoop.update(tiltError);
  pixy.setServos(panLoop.m_pos, tiltLoop.m_pos);
  oldX = pixy.blocks[trackedBlock].x;
  oldY = pixy.blocks[trackedBlock].y;
  oldSignature = pixy.blocks[trackedBlock].signature;
  return trackedBlock;
}
//--------------------------------------
// Follow blocks via the Sabertooth 2X25 driver
//
// This code makes the robot base turn
// and move to follow the pan/tilt tracking
```

```
// of the head.
//--------------------------------------
int32_t size = 400;
void FollowBlock(int trackedBlock)
{
  int32_t followError = RCS_CENTER_POS - panLoop.m_pos;  // How far off-center are we
looking now?
  // Size is the area of the object.
  // We keep a running average of the last 8.
  size += pixy.blocks[trackedBlock].width * pixy.blocks[trackedBlock].height;
  size -= size >> 3;
  // Forward speed decreases as we approach the object (size is larger)
  int forwardSpeed = constrain(127 - (size/256), -100, 127); //modified
  // Steering differential is proportional to the error times the forward speed
  int32_t differential = (followError + (followError * forwardSpeed))>>8;
  // Adjust the left and right speeds by the steering differential.
  int leftSpeed = constrain(forwardSpeed + differential, -127, 127); //modified
  int rightSpeed = constrain(forwardSpeed - differential, -127, 127); //modified
  // And set the motor speeds
  ST.motor(1, leftSpeed);//modified
  ST.motor(2, rightSpeed);//modified
  //motors.setLeftSpeed(leftSpeed);
  //motors.setRightSpeed(rightSpeed);
}
//--------------------------------------
// Random search for blocks
//
// This code pans back and forth at random
// until a block is detected
//--------------------------------------
```

```
int scanIncrement = (RCS_MAX_POS - RCS_MIN_POS) / 150;
uint32_t lastMove = 0;
void ScanForBlocks()
{
  if (millis() - lastMove > 20)
  {
    lastMove = millis();
    panLoop.m_pos += scanIncrement;
    if ((panLoop.m_pos >= RCS_MAX_POS)||(panLoop.m_pos <= RCS_MIN_POS))
    {
      tiltLoop.m_pos = random(RCS_MAX_POS * 0.6, RCS_MAX_POS);
      scanIncrement = -scanIncrement;
      if (scanIncrement < 0)
      {
        ST.motor(1,-79); //modified
        ST.motor(2,79);  //modified
        //motors.setLeftSpeed(-250);
        //motors.setRightSpeed(250);
      }
      else
      {
        ST.motor(1, +57); //modified
        ST.motor(2,-57);  //modified
        //motors.setLeftSpeed(+180);
        //motors.setRightSpeed(-180);
      }
      delay(random(250, 500));
    }
    pixy.setServos(panLoop.m_pos, tiltLoop.m_pos);
  }
```

Appendix D – Individual Reports

Anthony Olvera – Electronics and Programing

Task	Description	Time Spent (Hours)
Creating prototype	Installed necessary components onto space-grant robot	2
Programming	Getting necessary libraries, modifying code, writing code and fine tuning	40
Fabricating connectors	Power Jack, SPI Cable, etc.	4
Ordering parts	Selecting parts and filling out appropriate order forms	5
Develop kill switch	Configuring Xbee radios, designing circuit, soldering, design of enclosure and 3D printing	30
Develop relay cut off circuit	Configuring Xbee radios, designing circuit and soldering	25
Sensor integration	Ultrasonic and photo resistor	3
Developing circuit schematic	Making circuit in fritzing software and routing all connections	5
Assisting in assembly	Helping Jonathan and Lucas in cutting aluminum, and assembling the frame	4
Voltage divider design	Creating a voltage divider circuit to measure low voltage	3
Shopping	Getting additional parts from various stores	2
PCB Development	Design, etching and soldering	8
Electronics Box	Cutting, drilling and gluing acrylic.	9
Troubleshooting	Self-explanatory	10
Testing	Experimenting, collecting data, seeing what needs to be improved	3
Improving	Code for dampened acceleration, creating Circuit and wearable IR light	5
Typing Report	Self-explanatory	5
Creating Presentation	Self-explanatory	2
TOTAL		165

Allan Quillen – Electronics and Programming

Task	Description	Time Spent (Hours)
Pixy Solid Works	Re develop the pixy camera from previous drawings to fit our specifications	30
Pixy Camera	Learn and tune correct color and IR signatures to match overall size to distance ratio	50
Assisting in assembly	Assisted in small assembly and input in design	10
Modification to Programming	Modify programming control signature size to distance ratio as well as determining desired signature size to speed ratio	25
Report	Adding in pictures and technical specs and procedures to final draft	10
Research	Researching of sabertooth specs, pixy specs, Arduino due specs, motor specs, Infrared circuit designs	20
Infrared circuit design	Research and test IR led driver circuit for optimal brightness of signature saturation	10

Jonathan Widjaja – Mechanical and Assembly

Task	Description	Time Spent
Solidworks	Designing 3D design of the chassis and get visual idea of the end product	8 Hours
Determine chassis and frame material	Researching on the materials and parts for the mechanical parts, looking parts that we need from certain website and putting it on the order form	7 Hours
Tire and wheel hunting	Tire and wheel hunting by visiting locals shop to get close to see which dimension of tire and wheel we needed as well as what type we should be using	3 Hours
Machining	Cutting down the t slot aluminum bar to the size we needed on the band saw and grinding the sharp edges and make it smooth	27 Hours
Assembling	Constructing the t slot aluminum bar to make the chassis and frame. As well as mounting the pillow block bearings, front tire and wheel to the axle, caster wheel, needle bearings, metal sheet, plastic the mechatronic mule. Mounting the motors, chains, battery and the electronic box on to the mechatronic mule	60 Hours
Testing the mechanical movement	Testing the mechanical movement by pushing it around manually to make sure the smooth movement of the mechanical parts.	4 Hours
Testing the full assembly	Testing the fully assembly both indoor and outdoor to make sure it working perfectly	15 Hours
Reports	Writing weekly reports and writing my part of final report	18 Hours
Final presentation	Preparing for final presentation as well as making the poster and printing the poster	8 Hours
Consultation	Consulting with Paul Wallace about the design and figuring which is the best way to mount some of the parts	5 Hours
Total		155 Hours

Lucas Mendicello – Mechanical Design-Implementation

Task	Description	Time
Determine supplies	Finding the proper mechanical supplies need to construct the design.	15 hrs
Solidworks	Creating assembly withal components.	10 hrs
3D printing	Assisted Anthony and Allan in printing the required housing, mounts, etc.	5 hrs
Machining	Making components the necessary length by cutting, milling, etc.	30 hrs
Construction	Assembling the components into the MechaMule.	50 hrs
Reports	Weekly reports as well as a section of the final report.	15 hrs
Consultation	Consulted Paul Wallace frequently about design and cutom components that were manufactured.	15 hrs
Testing and Troubleshooting	Testing design, analysis, and troubshooting.	10 hrs
Total Hours	Combined for semester	155 hrs