F.E.T.C.H
Fire Evacuation for Total Critter Happiness

Authors:
Cyrus Neary 35108133
Ian Thompson 31192131
Charles McGrath 40465130
Akshiv Bansal 35072131

Engineering Physics 253
University of British Columbia

June 18, 2015

Executive Summary

This is a design proposal for the Engineering Physics 253 Pet-Rescue Bots competition. The competition involves building an autonomous robot that can navigate an obstacle course and successfully extract six "pets" (stuffed animals).

We are the F.E.T.C.H. team, and we will be taking part in this year's Enph 253 robot competition. We intend to have the fastest course completion time of any team taking part in this competition. This is the underlying principal guiding our entire approach to this problem. The key features of our strategy are going to be getting pets 4 and 5 before pets 1, 2, and 3, thereby negating the wall of fire, coupled with completely ignoring pet 6. These two simplifications allow us to focus on making our robot as fast and reliable as possible

In the mechanical design of the robot we are aiming to make the robot as light as possible. To do this we have gone for a chassis shape and design that is made out of light materials and can be fabricated quickly. We wanted to have fast fabrication to be able to make adjustments rapidly and spend as little time as possible in between testing. The drivetrain is built to go as fast as we think is possible, while maintaining reliability in controlling the robot. This informs our decisions on gears sizes, motors, and wheel size.

Our mechanism for picking up the animals focuses on simplicity. As a result, we omitted the pursuit of pet six, which presents its own set of challenges and added complexity. Instead we want to be able to reliably pick up animals 1,2,3, and 5 with our main arm and animal 4 with out supplemental arm.

F.E.T.C.H.'s electronic systems are designed to be simple and modular. This will allow for a system that is easily built, easily serviceable, and easy to troubleshoot.

The main outcomes from our design process are: a focus on simplicity for individual parts, building to go as fast as possible given our constraints, and fabricating in a way that makes it possible to replace individual parts with out holding up the rest of the building process.

Table of Contents, List of Figures and Tables

Table of Contents

List of Figures

Preface

The authors of this report are Akshiv Bansal, Charlie McGrath, Cyrus Neary and Ian Thompson. They are all second year students in the Engineering Physics program at UBC. Collectively they have experience working on various student design projects, namely Formula UBC SAE, UBC Solar, and UBC E-Racing. This experience translates well to the rapid design and prototyping challenges presented by this competition. This design proposal has been written under the guidance of Dr. Andre Marziali, Mr. Bernhard Zender, Dr. Jon Nakane, and Ms. Pamela Rogalski. We would also like to thank Kelvin Poon, Scott Lawson, and Zendai Kashino for their assistance throughout the design process. The robot design concepts laid out in the following report were developed through close collaboration between the members of the group. We wish the contents of this report, including any details of our approach and strategy, be kept confidential from the other teams until the end of the competition. The sections of the report were written individually by members of the group, and these tasks were divided based on familiarity with the systems being discussed. The task breakup for this design proposal is as follows.

<u>Division of Labour</u>

<u>Writing</u>	<u>Design</u>
Letter of Transmittal – Cyrus and Ian	Electrical Circuits Drawings – Cyrus
Executive Summary - Akshiv	Algorithms - Akshiv and Ian
Preface – Ian	Drivetrain – Akshiv
Overview of Basic Strategy - Cyrus	Front Supplementary Arm – Ian
Chassis - Charlie	Electronics Bracket – Cyrus
Drive and Actuator System - Akshiv	Chassis- Charlie
Electrical Design - Cyrus	Main Arm – Charlie
Strategy, Algorithms and Software - Ian	Back mounts - Charlie
Risk Assessment and Contingency Planning - Ian	Assembly and Tolerance- Charlie
Task List, Major Milestones, Team Responsibilities – Akshiv, Cyrus, Ian	Electrical Circuits Drawings – Cyrus

Akshiv Bansal Charles McGrath Cyrus Neary Ian Thompson

1 Overview of Basic Strategy

1.1 Introduction

This report outlines a proposal for the design of a robot meant to compete in the "Fire at the SPCA" competition. It includes details on our team's design and strategy choices, as well details on the robot's various subsystems.

In the "Fire at the SPCA" competition, teams build robots that autonomously navigate a 2 level stage (connected by a sloped section), rescue 6 stuffed animals from an imaginary fire, and safely return them to the starting area. From the start location, there is black tape that the robot is able to follow partway across the stage. Three pets are placed at various points along the path, 8 inches to the right of the tape. The 4th animal is at the end of the black tape, sitting on the path. Animal 5 is placed 6 inches above the ground, 8 inches to the right of an imaginary line that connects the end of the tape, and an IR beacon at the end of the competition stage. Animal 6 is buried in a bucket of foam that is also in line with the end of the tape, and the IR beacon.
Adjacent to the bucket, there is a zip line that returns to the starting area.
There is also a second IR beacon directly opposite the zip line, that allows robots to re-find the tape. However, there is a wall of fire, that emerges from the stage 60 seconds after the robot begins its run, effectively blocking the black tape path after this point in time.

Our team identified 3 possible strategy choices.

1. Collect all of the animals in order (i.e. 1,2,3,4,5,6) then use the zip line to return the animals to safety.

2. Collect all of the animals in order, then return to the wall of fire, and use a spring-loaded trebuchet to throw the animals back to the starting area.

3. Collect the animals in a different order (4,5,3,2,1), excluding animal 6, and exit the area blocked off by the wall of fire before it is raised after 60 seconds.

1.2 Competition Strategy

Our team has decided to build a very simple robot that is able to accomplish its tasks quickly, and consistently.

In order to cut out any complexity that is not absolutely necessary for the completion of the course, we have chosen to avoid the zip-line, or any throwing mechanisms. Instead we will make a lightweight robot with fewer parts, that is able to navigate the course quickly enough to return past the wall of fire before the 60 second mark.

This simpler robot will also be relatively easy to construct, allowing for extra time for troubleshooting after it is built, and for modifications in strategy if things don't work the way we anticipate.

Our robot, named F.E.T.C.H (Fire Evacuation for Total Critter Happiness), will begin the competition by tape following past animals 1,2 and 3 without picking them up. It will pick up animal 4, using a small arm that reaches out in front of the robot, and then rotates away and up, clearing the animal from the driving path of the robot. This pick-up will trigger the algorithm used to find and pick up animal 5, using the main arm of the robot, rotated to the proper height. After collecting animal 5, the robot will perform a 180-degree turn, and use the IR exit beacon to re-find the black tape. Once the black tape has been found, the robot will line follow back to the entrance, and pick up animals 3, 2, and 1 on its left side.

By skipping the first 3 animals on the way up the course, we allow for more time to pick up animals 4, and 5 before the wall is raised. We also plan on incorporating a time threshold in our code, at which point the robot will abandon whatever it is doing past the wall of fire, in order to escape on time and rescue at least animals 1,2, and 3.

While we have decided to avoid the collection of animal 6 for the time being, we have not entirely left it out of consideration. At this early stage in design, our team has decided that the collection of animal 6 risks complicating the design of our robot to the point that we are not able to consistently collect all of the other animals. We have decided to focus on collecting the other animals, and building a robot that is able to move quick enough to beat the wall of fire. We plan on leaving room on the front of our robot to potentially add a pickup mechanism if we are able to finish our first build quickly, and we see that our robot is capable of collecting the first 5 animals consistently.

2 Chassis

2.1 Chassis Design Goals

Our choice of a strategy that depends on speed directed our chassis design. This primarily involved minimizing weight and moment of inertia about the point of rotation between the drive wheels, while still allowing for a stable, maneuverable platform for the arm.

2.2 Main Chassis Frame

The primary frame is designed for these criteria, thus the shape of the chassis base is an octagon that has been stretched. For strength and stiffness, edge flanges are bent and spot welded together in order to make a box-like structure. The chassis will be waterjet cut out of 0.6mm mild steel, and then bent as accurately as possible to create the structure seen in Figure 1. In order to reduce weight without compromising strength, material is cut out during machining of the top surface of the frame. These cut outs are flexible in their positioning and size, allowing for various mounting point additions.

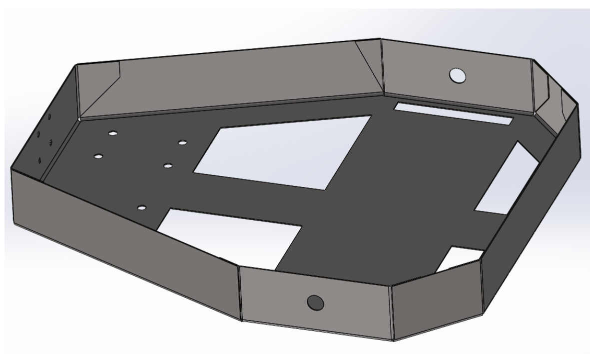


Figure 1 – Chassis Frame

2.3 Drive Motor Mounts

The motors are mounted using simple bent sheet metal boxes, which are fastened to the bottom face of the frame using setscrews and nuts. These are positioned behind the wheels on either side in order to help balance the robot around the rear axles, as the robot is mostly in front of the axles. Each axle will have a rotary encoder, which will be coupled to the end of the shaft, and this encoder will be mounted to the underside of the frame. These encoders will perform a dual function; both encoding wheel position and providing a structural mount for the other end of the axles to prevent gear slip and wobble.

2.4 Arm Support Structure

The arm will be supported by a simple sheet metal support that will be waterjet cut out of 0.6mm mild steel and bent to shape. Edge flanges will be spot welded together to create a box-like structure. This will be bolted onto the top surface of the frame in order to support the arm assembly. Similar to the frame, excess material that is not providing any strength will be cut out.

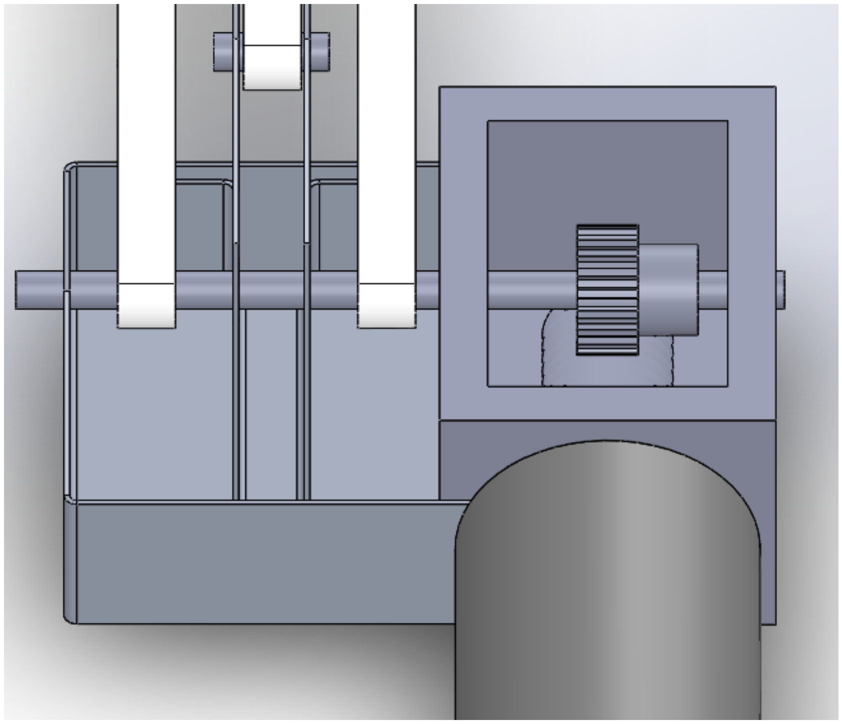


Figure 2 – Arm Supports

2.5 Electronics Housing

The electronics and TINAH board housing sits at the rear of the robot, and will be waterjet cut out of 0.8mm aluminum to save weight, as this is a non-structural part. Similar to previous parts, the part will be bent and tabs will be fastened together with rivets or set screws. The rear area of the housing will be a large door with the TINAH mounted to the outside, allowing both easy access to the TINAH from the back of the robot, as well as access to any electronics (battery, circuit boards, wiring) inside the housing.

2.6 Front Swivel Wheels

Due to the need for quick maneuvers, we chose to use two small swivel wheels mounted near the front of the robot. These will be bolted to the underside of the frame, which allows the frame to sit relatively low to the ground, creating more space for the arm and animal bucket. We currently have two options for wheels, one from McMaster Carr which is what the robot is currently designed for, and the wheels in the project lab which can be set to the correct height using a 3D printed riser.

2.7 Sensor Brackets

All small sensors will be attached to the robot using simple bent brackets that will be waterjet cut out of 0.8mm aluminum sheet. This allows for sufficient strength and very light weight. Any sensors, such as switches and pots, that experience physical loads will have brackets with tabs to form boxes for strength.

2.8 Design Adaptability

Due to the large amount of space on the frame, changes in mounting of sensors and brackets is as simple as changing hole positions. The drive wheels are mounted outside the chassis, allowing for easy modification.
The mass of the Chassis alone is expected to be 711 grams, which still includes room for refinement as parts are optimized for low weight and sufficient strength.

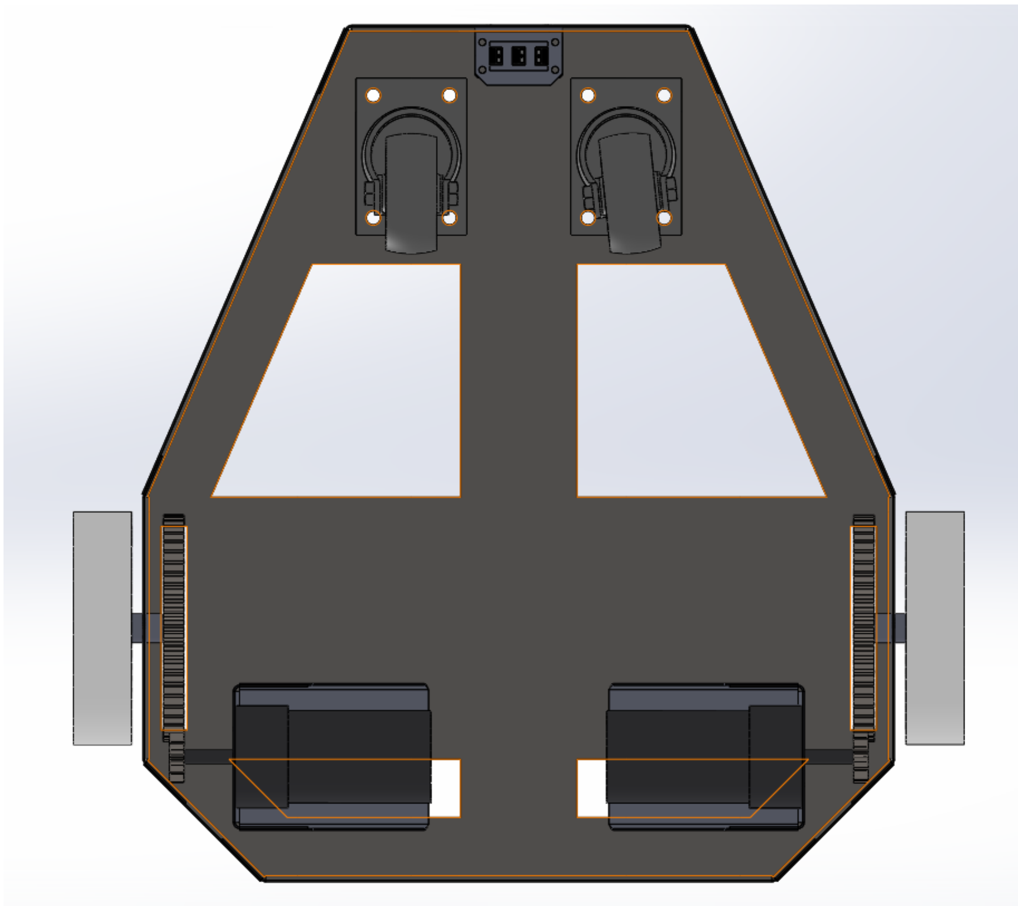


Figure 3 – Chassis Motor Mounts

2. 9 Animal Containment

Animal containment will be accomplished by mounting bend rods to the front of the robot, which go out to a height of around 6 inches above the chassis surface. These rods will have string or wire fastened to them to create a wire cage that surrounds the front area of robot. This acts as a large, flexible and lightweight animal container. The rods will be positioned to maximize the contained volume while still allowing clearance for both the front/rear arms and animals. Where needed, aluminum sheet metal guards will be implemented to prevent the animals from contacting moving parts such as gears.

3 Drive and Actuator Systems

3.1 Requirements for Drivetrain

Our strategy centers around robot speed. In order to deal with the challenges of the competition, namely the wall of fire, we have picked a system that focuses on reducing the time needed to navigate the course. Our goal is to go fast, so we are using high power motors, with relatively large wheels (3 inches). This forces us to be more precise in machining our gears, and spend more time in the optimization of the drivetrain. These trade-offs allow us to greatly reduce the overall complexity of our robot, by bypassing the major design challenge of the wall of fire.

3.2 Drive mechanism and transmission

We are driving the two back wheels of the robot through independent motors and varying the power input to them as a method for turning the robot. This will allow us to execute much faster turns, which is necessitated by our strategy. This placement of our drive mechanism also allows us to be more symmetric in the placement of the loads on the robot. Symmetric load placement will facilitate control of the robot's steering, as the control loop will be more stable.

We intend to use the ungeared Barber Coleman motors to power the rear wheel drivetrain. These then go through a gearbox with a 20:1 reduction to get to the wheels. It is possible that this is an overestimate of the gearing we need, but it seems reasonable given our factors of safety (see transmission calculations appendix 2). These motor shafts feed in to laser cut acrylic gears that are press fit onto the shaft. If it proves to be too difficult to create a reliable mate with acrylic, we will switch to waterjet cutting polycarbonate. The outer gear is then connected to our wheel axles.

3.3 Transmission calculations

Instead of explicitly using a factor of safety, we make two generous assumptions. First we assume the mass of the robot is 5.5 kg, which is a large over estimate, and then we say we are powering the motors with 12V DC, instead of the 15 V we expect to get from the LiPo battery. This will compensate for frictional losses in the transmission and the driving.

3.4 Motors

We need three motors in order to have an operational robot, as well as one servo. Two ungeared Barber Coleman motors are needed to power the drivetrain of the robot. One ungeared Barber Coleman motor is needed to for the rotation of the arm. Lastly, a servo is needed for the supplemental pickup of animal four.

All three DC motors require external H-Bridge circuits, powered by 15 volts from our lithium polymer battery packs. This is to allow them to draw sufficient current for reliable function. In order to get a sense of the motor behaviour, we constructed a simplified motor model based on Shane Colton's guide to DC motors in the MIT 2.007 course (appendix 1).

For the driving motors we expect to run motors at max power up the ramp and closer to maximum efficiency for the rest of the course. See appendix 1 for power calculations of motors.

The arm is powered by an ungeared Barber Coleman motor, and is going to be run at max power exclusively. This is because we are only running the motor for very short intervals, and we need this operation to be very consistent. We are taking the mid-point in the simplified motor model to approximate our power consumption.

Lastly we are using a servo as a pickup for the fourth animal. This is a very small operation, as the motor arm does not need to sweep very far. To accomplish this task we will be using a Hobby King Servo Motor.

3.5 Wheel Selection

We choose wheel size based on the constraints of our robot. We need to choose wheels that allow us to meet our required timing, make it possible to climb the hill, and allow us to reliably follow tape with our control algorithms.

Based on geometric constraints of the robot we have a hard limit on the width of the wheels. The wheels can be no more than 1 inch (2.54 cm) wide. We have no such limit on the radius of the wheel, here we look to our other constraints to choose a size.

Our aim is to traverse the course quickly, but still have the power to make it up the hill. Our torque delivery is limited both by the power of the motors, and by the friction of our wheels. Another major constraint on our wheel choice is ease of fabrication. We want to select a wheel of a size that is relatively easy to purchase. While prototyping we will use wheels that we build, but for our final robot we will use wheels manufactured to a reliable quality.

Another important design decision is the material that we build the wheel out of. In prototyping we are going to laser cut our wheels out of acrylic. Given that plastics are generally quite soft, this will probably not match our friction needs. To compensate for this we will be adding rubber tires to increase the friction between the surface of the track and the wheels. Our final wheels will be polyurethane wheels and rubber tires.

Keeping these constraints in mind, we have decided to go with a wheel diameter of 3 inches. This ensures that we will be able to switch to a manufactured wheel eventually, because that size is a standard part for many RC car kits and is readily available online and in several local hobby stores.

Using our gear train calculator we see that we can reliably get up the ramp with this setup of motors, gears, and wheels. It remains to be seen if our control systems will reliably follow tape in this configuration, but since the drivetrain is largely decoupled from the rest of the robot we can easily swap components until we have reliable tape following. Due to our manufacturing process, we can test multiple wheel radii, and iterate quickly to find the best one.

Estimated Max Speed

In our current setup the limiting factor for speed is how quickly the motors can spin given their loading. The only forces applied to the robot along its direction of motion during normal operation (not including the ramp), are the force of the motors at the wheels and the sum of the resistance forces, namely: friction between the ground and the wheels and the drag forces caused by the geometry of the robot. Given the speeds that our robot moves at, we do not expect the air drag forces to play a big role. We want to run the motors at about 70% of their top speed, based on our MatLab simulation of our drivetrain.

3.6 Actuator systems

The robot design has a main arm that is used to accomplish the primary task of animal retrieval. The main arm is designed to be able to pickup animals 1,2,3, and 5. We have a second supplemental arm for animal four.

Main Arm

The main arm consists of three functions:

1. Detection of the animals
2. Picking up and moving the animals
3. Dropping the animals into the bucket

Please see appendix 3 for the power transfer calculations showing the viability of this arm design.

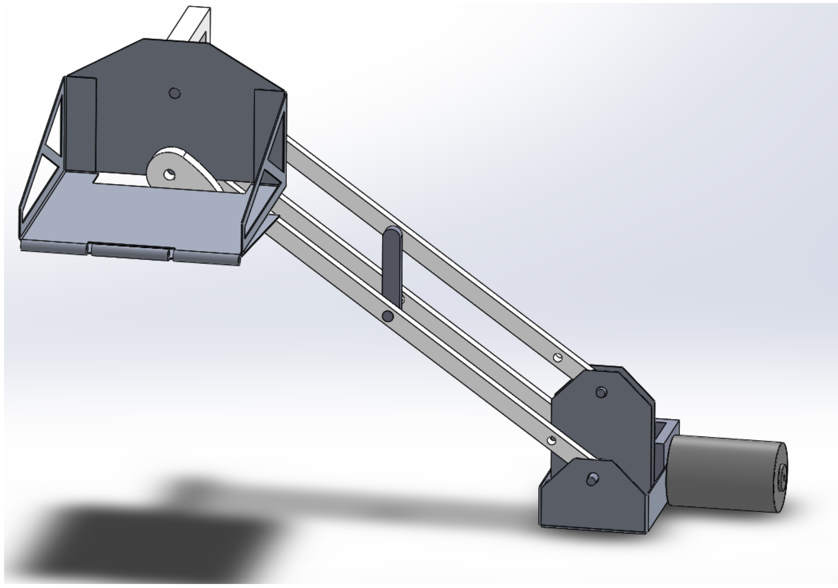


Figure 4 – Main Arm

Animal Detection

In order to detect the animals reliably, we are using a flexible non-magnetic (aluminum) plate, in front of a magnetic plate (steel plate). Once the animal is attached magnetically through the aluminum plate to the steel plate, the aluminum plate depresses to close a contact switch, alerting software that we have picked up an animal. This plate is positioned at an angle to allow for multiple contact geometries.

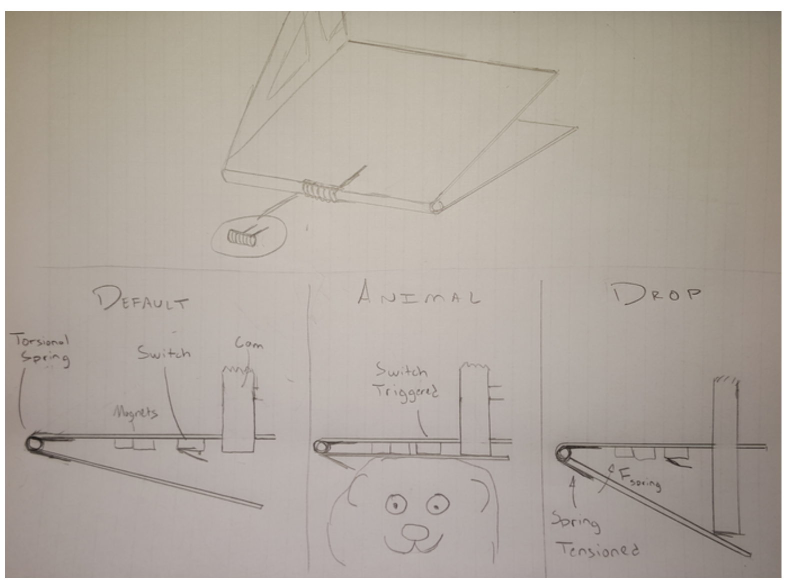


Figure 5 – Main Contact Plate

Picking up and moving the animals

Once F.E.T.C.H is certain that it has an animal, it will move the arm over the bucket to drop the animal in. Linkages connected to an ungeared Barber Coleman motor accomplish this motion. The trajectory of the plate on the end of the is a circle. The linkage is composed of three bars allowing for approximately 190 degrees of rotation, in its current configuration. We have designed the geometry such that at the height of 6 inches, and at the height of 12 inches, the horizontal distance of the plate from the centerline of the robot is the same. This means that there are two different plate heights that give the plate the same horizontal distance of 8 inches from the black tape. See appendix 4 for the main arm's geometry.

Arm Gear Box

The motor connects to the arm through a worm gear. As the motor spins it advances the worm gear and that drives the gear forward or back to move the linkage. The reason for using a worm gear in this system is to be able to lock the arm in place while the robot translates. We use the known height of the animals as the method for figuring out where they are, and as a result we need the arm to remain in the position we set. The rotating arm allows for a faster and more reliable pickup, as compared to linear actuation. This is because we use rotational motion to translate in two dimensions simultaneously rather than translating in one dimension at a time. In addition to this we require less precision in our movement because we only ever move between three set positions and should never to stop in intermediate states.

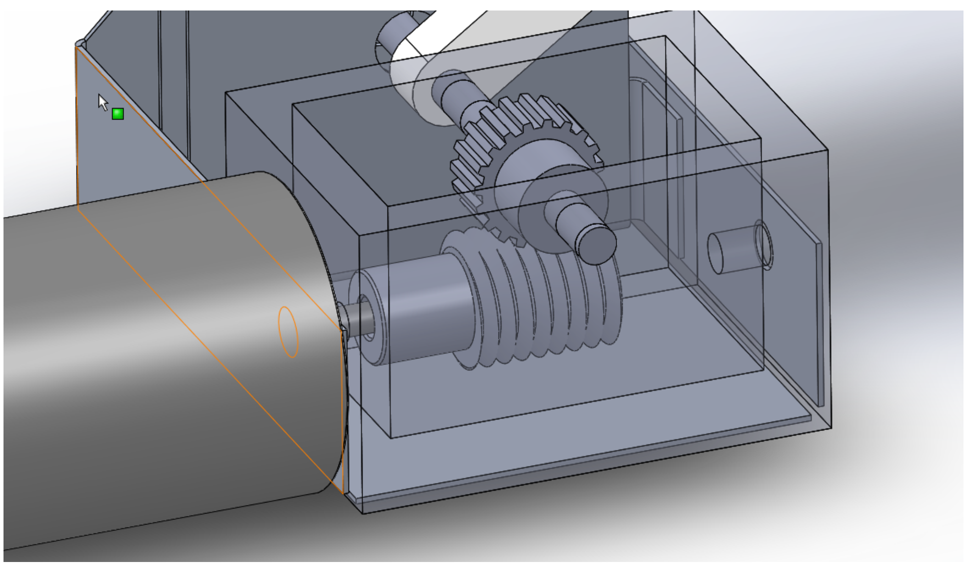


Figure 6 – Worm Gear Box

Dropping the Animals into the Bucket

Now that we can detect animals and move them from side to side, we still need a method of dropping them into a bucket. To accomplish this we us a cam attached to a linkage bar. This bar extends down to the central rotating axle. The cam is located in between the aluminum plate and the steel plate. This cam extends when the arm is directly overhead, by way of its connection to the rotating axle. The cam forces the aluminum plate and the steel apart breaking the magnetic link between the animal and the steel. We position the bucket directly underneath the plate at this point in the arms trajectory to catch the falling animals. The primary advantage of using a cam system instead hitting the animal against a support beam, is the savings on the weight of the center beam.

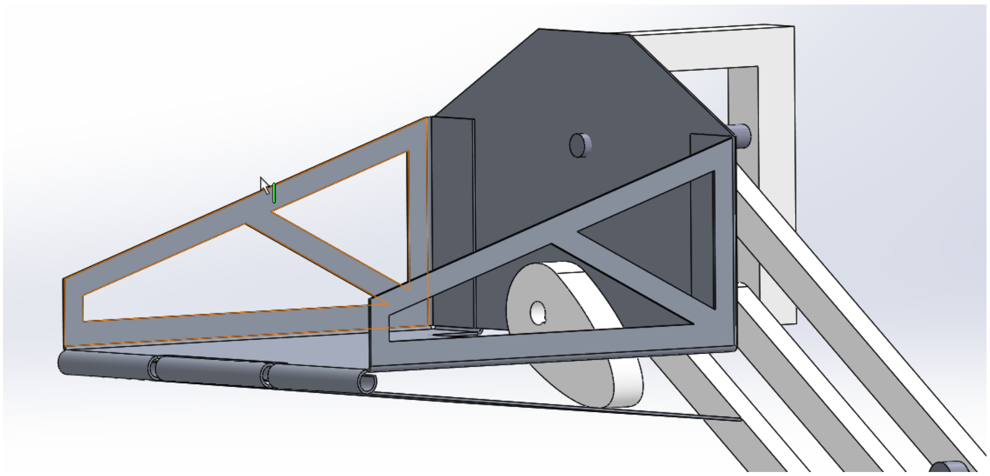


Figure 7 – Cam Assembly

Secondary Arm

The secondary arm is the same sort of plate attached to a rigid member being driven by a servo. This arm contacts animal four and upon contact triggers the servo motor to rotate the arm. We rotate the arm just enough so that the animal is no longer touching the ground. This way we do not need to worry about any complex geometry, or any high power mechanics.

See Appendix 3 for calculations for arm transmission and power.

4 Electronics and Sensor Systems

4.1 Introduction

The following section outlines the power supplies used by the F.E.T.C.H. robot, as well as its sensor and actuator systems. All electronic systems on the robot are designed to be simple and modular, in order to allow for a system that is easily built, easily serviceable, and easy to troubleshoot.

Necessary electronic components for the completion of our competition strategy are: 2 contact switches, 2 rotary encoders, 2 IR sensors, 3 reflective sensors, a rotary potentiometer, a servo motor output, and 3 DC motor outputs. All of these sensors and actuators, will be controlled by the TINAH microprocessor board, and powered either by +5V from the TINAH, +15V from the LiPo battery, or $\pm$9V from the onboard 9V batteries.

Please see appendix 5 for an overview schematic of the entire system.

4.2 TINAH Board Resource Allocations

TINAH Board Resources	
Digital Pins (16)	
0	Arm Animal Pickup Switch
1	Animal 4 Pickup Switch
2	Left Wheel Rotary Encoder A
3	Right Wheel Rotary Encoder A
4 to 16	Not Connected
Analog In (8)	
0	Pot for servo
1	IR 1 - Front
2	IR 2 - Front
3	QRD 1
4	QRD 2
5	QRD 3
6	Knob
7	Knob
PWM Out	
0	(Motor Out)
1	(Motor Out)
2	
3	
4	(Motor Out)
5	(Motor Out)
Motor Outputs and Indicators	
0	Left Drive Wheel
1	Right Drive Wheel
2	Servo Arm
3	
Servo Out	
0	4 Passive Pickup
1	
2	
Knobs	
6	Tuning Control (PD Control/ QRD Threshold for competition day)
7	Tuning Control (PD Control)
Power Rails	
0	PCB 4 (5V, GND)
1	PCB 5 (5V, GND)
2	Contact Switches (5V, GND)

Figure 8 – TINAH Resource Allocations

4.3 Wire Bundles

Wire Bundles				
Bundle #	Start	End	Wires	Notes
0	TINAH (Analog in)	PCB 5	5V (w/ Resistor), GND, QRD Signal 1-3	100 Ohm Resistor on end of 5V line
1	TINAH (Analog in)	Arm Pot	5V, GND, Signal	Analog Signal to give feedback on main arm servo motor position.
2	15V Battery	Left H-Bridge	15V, GND	15V power to H-Bridge
3	15V Battery	Right H-Bridge	15V, GND	15V power to H-Bridge
4	15V Battery	Arm H-Bridge	15V, GND	15V power to H-Bridge
5	Motor Out 0	Left H-Bridge	Signal +, Signal -	
6	Motor Out 1	Right H-Bridge	Signal +, Signal -	
7	Motor Out 2	Arm H-Bridge	Signal +, Signal -	
8	TINAH (Digital in)	Main Arm Pickup Switch	GND, Switch Signal	Switch used to sense animal pickup on main arm.
9	TINAH (Digital in)	Front Arm Pickup Switch	GND, Switch Signal	Switch used to sense animal 4 pickup in front of robot.
10	TINAH (Digital in)	Left Rotary Encoder	GND, Signal A	Rotary encoder to count turns of left drive wheel.
11	TINAH (Digital in)	Right Rotary Encoder	GND, Signal A	Rotary encoder to count turns of right drive wheel.
12	Servo Out 0	Front Pickup Arm	Signal, Power, GND	Servo to lift front arm after detecting animal 4.
13	9V Batteries	IR Filter Circuit	+9V, -9V, GND	Batteries connected in series. Ground in middle. Outer voltages (+9V, -9V) used to power Op-Amps in circuit.
14	15V Battery	TINAH DC Input	Provided Battery Connection	This is our boards main power

Figure 9 – Wire Bundles

4.4 PCB List

PCB List		
PCB Number	Contents	Size (cm by cm)
0	Front Left IR Filter	8 x 5
1	Front Right IR Filter	8 x 5
2	Left Drive Wheel H-Bridge	6 x 4
3	Right Drive Wheel H-Bridge	6 x 4
4	Arm H-Bridge & "Servo Feedback"	6 x 4
5	3 QRDs	1.5 x 3

Figure 10 – List of PCBs

4.5 Modified H-Bridge Motor Driver Circuit

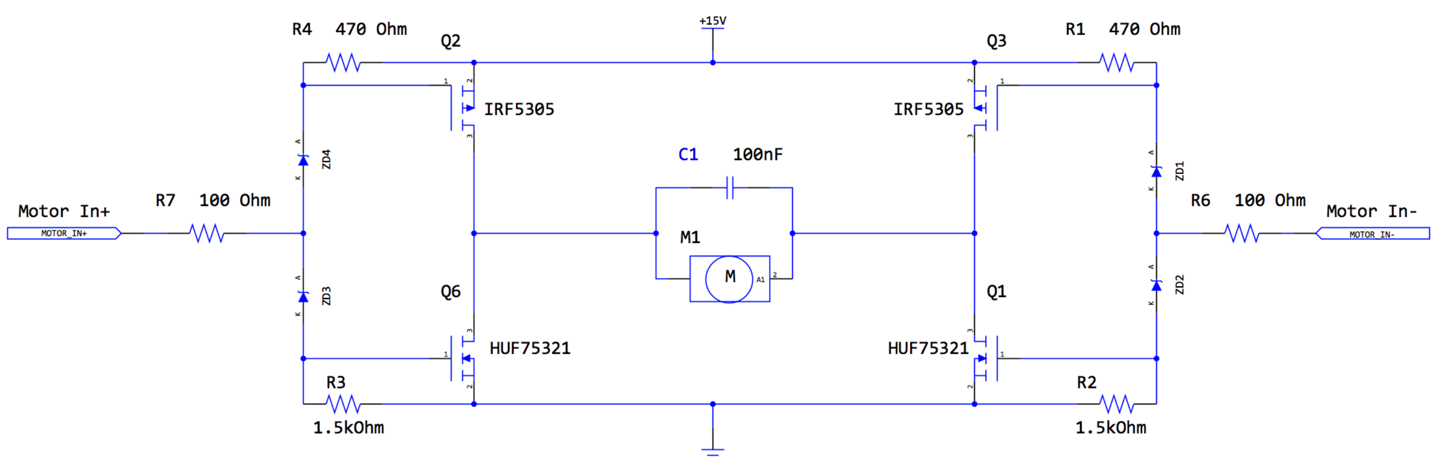


Figure 11 – H-Bridge Circuits

4.6 Infrared Sensor Circuit

In this circuit, we will use 3 band pass filters in series to steepen the slope of the bode plot on either side of the 10kHz frequency mark, allowing us to detect only these signals. We have chosen component values such that each filter has a gain of 2 and the signal strength is not diminished by the inclusion of the extra band-pass filters. Please refer to figure 12 below.

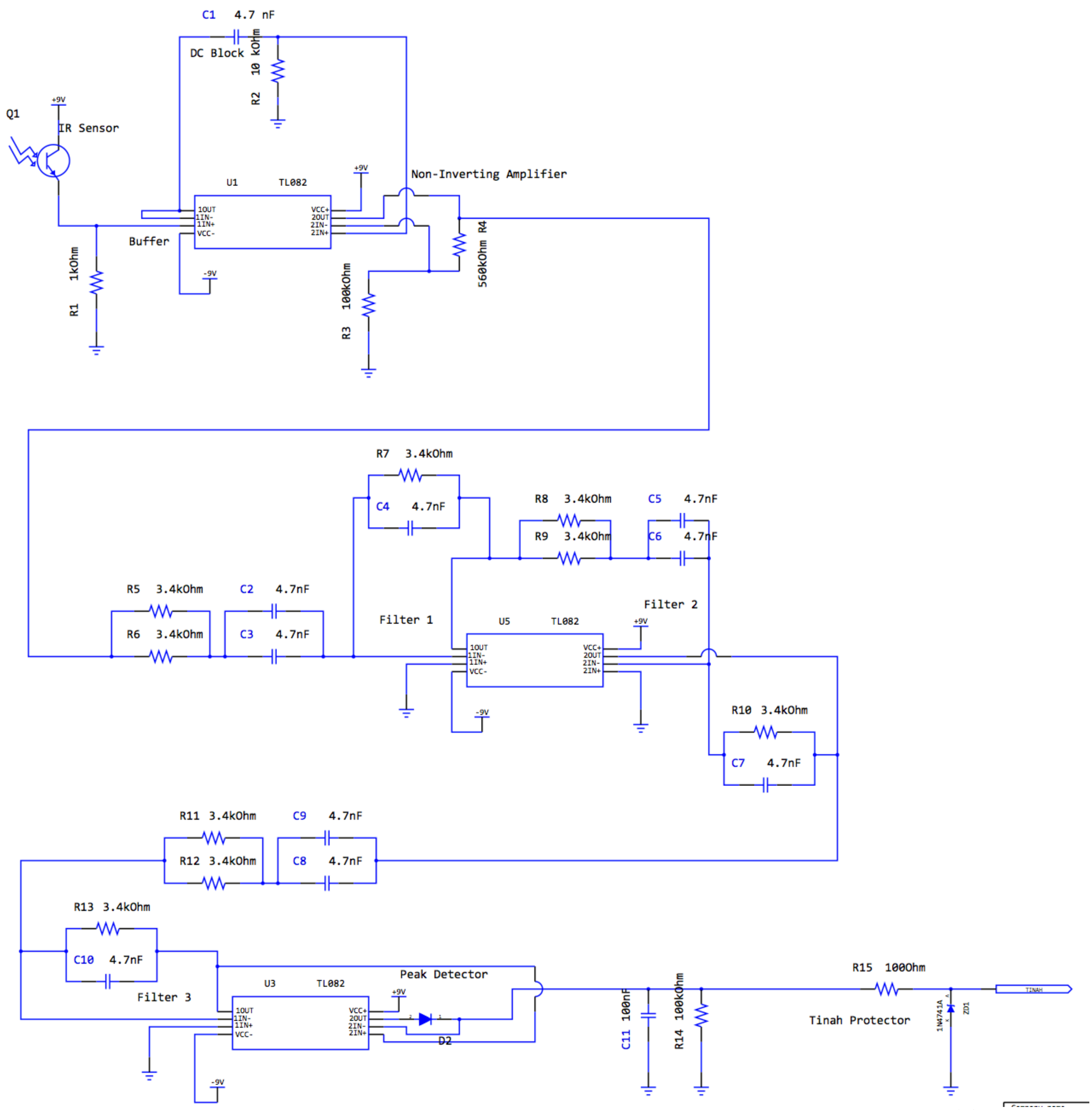


Figure 12 – IR Sensor Circuits

4.7 Contact Switches

In order to detect when an animal has been picked up, F.E.T.C.H will have contact switches that trigger when an animal becomes magnetically attached to either the main arm, or the front arm.
Please see appendix 5 for reference on how the switches are connected to the TINAH.

4.8 Rotary Encoders

In order to collect animal 5 past the end of the black tape, F.E.T.C.H will be required to have some input that gives it feedback on its position and orientation. F.E.T.C.H will make use of 2 rotary encoders, one per wheel, in order to count the number of turns that each of its wheels make, giving the robot feedback as to its orientation on the competition stage.

The rotary encoder is electrically connected to ground, and to one digital input pin on the TINAH. The encoder provides a connection between ground and this digital pin 24 times per revolution, providing a digital signal that the TINAH is able to count and process, in order to track wheel rotation.

Please refer to appendix 5 for the proposed wiring of the encoders.
Note that a pull-up resistor is not necessary between the digital input pin of the TINAH and the pin of the rotary encoder because the TINAH has built in pull-up resistors on all of its digital inputs.

4.9 Main Arm Servo Motor

The main arm that will be collecting animals 1,2,3, and 5 requires a servo system that is able to rotate the arm to a precise position, in order to be at the proper height to magnetically attach to the animals.
We will construct such a system using a DC motor, a rotary potentiometer and control software on the TINAH board.
Please refer to figure 11 for the H-Bridge circuit that will drive the motor, and to appendix 5 for the proposed wiring of the rotary potentiometer.

4.10 Connectors

In order to achieve a modular electronics system, we will use MTA 100 wire to board connectors to connect our PCB boards to the TINAH board.

4.11 Power Supplies and Grounding

The H-Bridge motor driver circuits will be powered by +15V directly from the lithium polymer battery. The TINAH board will also be powered by the +15V lithium polymer battery. The TL082 op-amps in the IR sensor circuit will be powered by ±9V.

All circuit grounds throughout the robot are electrically connected to the battery ground. This is to ensure that there are no floating voltages, and no confusion in our electronics that could arise from separate circuits referencing their voltages to separate grounds.

Please see appendix 5 for battery and grounding reference.

5 Strategy, Software, and Algorithms:

5.1 Overview of Robot Control:

Our approach to controlling our robot throughout the course relies heavily on three main control loops: tape following, IR following, and wheel encoding. With these 3 main modes of control our robot will be able to navigate the course. During this navigation, we will make use of the TINAH board's interrupt routines to halt our navigation for animal pickups. We will have error handling algorithms ready to be triggered when the robot's state becomes no longer manageable with our three main control loops.

5.2 Description of Control Loop Algorithms:

5.2.1 Tape Following:

To perform tape following with our robot, we will be implementing the same PID control algorithm that was implemented in lab 5. Because tape following quickly is essential to our strategy, we will be using 3 QRDs instead of 2 to give us more position resolution around the tape and improve our tape following accuracy and speed. Three QRDs will allow us to have 7 possible positions with respect to the tape instead of the 5 obtained with 2 QRDs. Once our QRDs are placed on our chassis, we will determine the exact displacements that these different QRD states correspond to and use these positions as the discrete states in our code.

5.2.2 Wheel Encoding:

For sections of the course that require precise maneuvering, for example driving in a straight line to reach pet 5, then turning around 180 degrees, we will employ wheel encoding to precisely control our robot's position. Using the interrupt routines of the TINAH board, we can drive the motors in given directions while counting the exact number of pulses given by rotary encoders attached to each wheel. To do this, we will set up the interrupt routine to increment a counter each time it sees a low to high state change

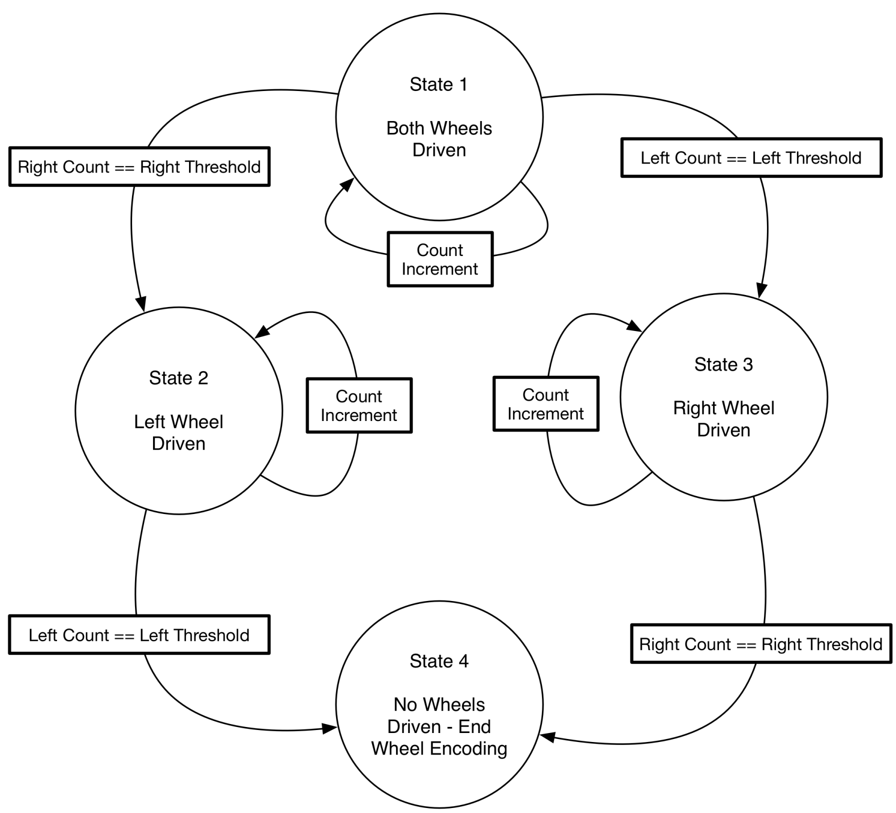


Figure 13 – Wheel Encoding Finite State Machine

of the encoder. Once the count of these pulses reaches exactly the desired value, we will stop all wheel rotation, knowing that the robot has travelled the exact desired distance or undergone the exact desired rotation.

5.2.3 IR Following:

Our IR algorithm is very similar to the PID control algorithm that was implemented for tape following in lab 5. We will read voltages from two IR sensors mounted a few inches apart on the front of the robot. By driving our robots wheels independently, we will minimize the difference between the two signals, placing us on a straight path towards the IR beacon. We will have a set voltage threshold corresponding to a distance at which we should have found tape again, or have passed the 5th pet. Crossing this threshold will trigger error handling algorithms for re-finding tape or the pet. The PID algorithm will be almost the same as for tape following, except we will be dealing with continuous differences, rather than a few discrete states. Because of this, computing the D term will no longer need the "creative" slope approximation we used in lab 5.

5.2.4 Animal Pickup:

We rely on the interrupt routines of the TINAH board to allow us to trigger our arm for animal pickup. When either of the contact switches on the arms are closed, the digital pins observe a high to low

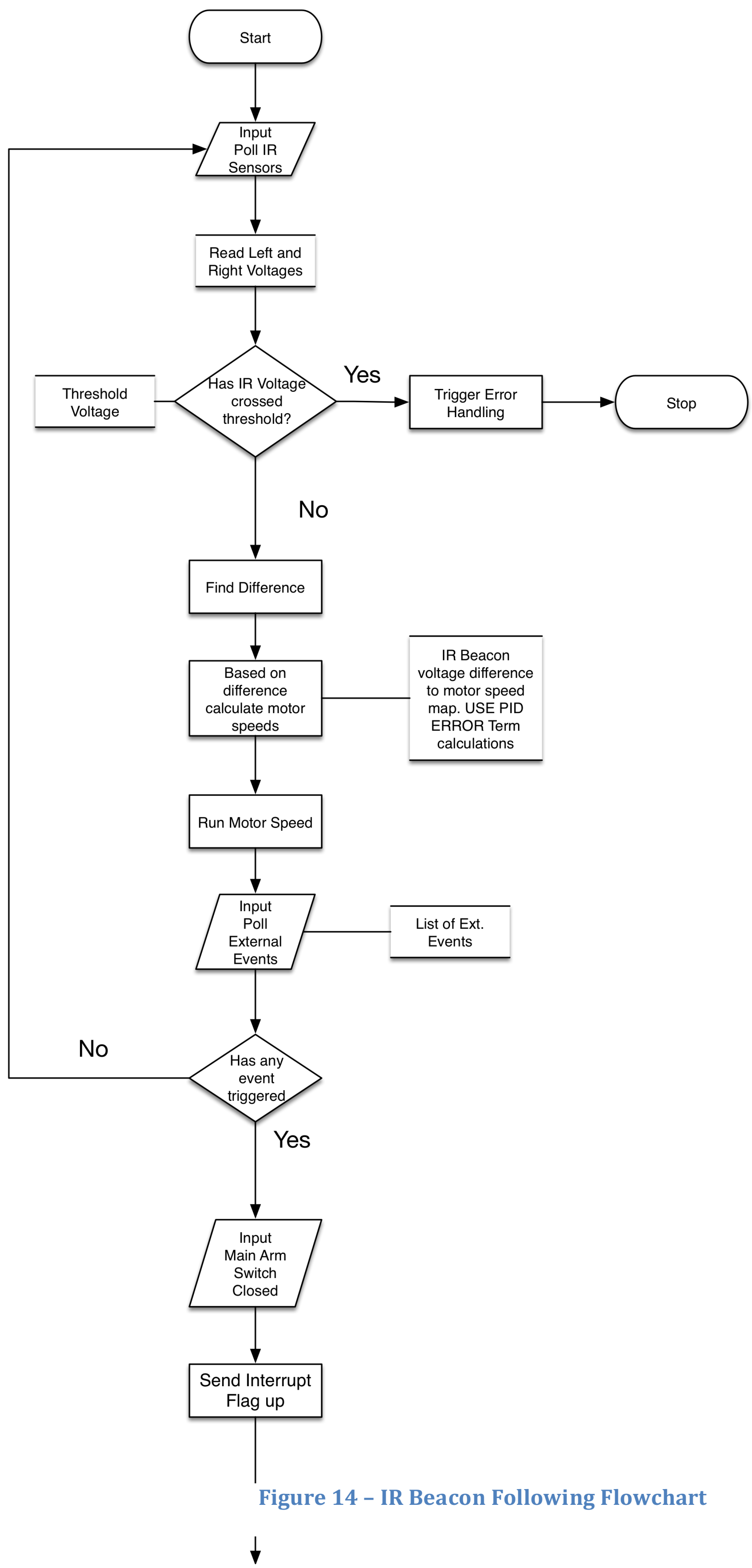


Figure 14 – IR Beacon Following Flowchart

transition in the signal. We can then lift the arm off its point of contact. If the arm switch continues to be closed, we know we have collected an animal, and can proceed to stow it away. If we have had a false closure, the switch will not remain closed, and we can then decide to either re-attempt a pickup, or continue our run, depending on the robot's current task.

5.3 Flow of Control for the Course:

The run will begin with the activation of our robot using an on/off switch, after which time, the robot will be completely autonomous. It will begin by line following and will follow the control flow shown in flowcharts Animal 4 → Animal 5→ Return to 3 → Pet 1-3 (See Appendix 6a to 6d).

5.4 Error Handling:

Tape Not Found:

If we fail to re-find tape after collection of Animal 5 we trigger an error handling algorithm. The current algorithm uses wheel encoding to rotate while searching for the tape. If this does not appear to work, we may also add movement to the algorithm so the robot not only rotates, but also drives side to side in an attempt to re-find tape.
See Appendix 6e for flowchart.

Lost IR Beacon:

In the case that we lose the IR beacon's signal while attempting to follow it, we will implement an algorithm, which rotates the robot side to side, searching for increases in the voltage due to IR signal. The robot alternates rotating either direction looking for the signal, then after some time with no success, shuts down its functions to preserve battery life.
See Appendix 6f for flowchart.

6 Risks and Contingency Planning

Our simple, performance focused approach to the challenge of "Fire at the SPCA" has allowed us to create a robot that has very few complex parts, reducing our risk of failure. The simplicity of our design leaves a lot of freedom to make adjustments to our robot in response to problems that arise, as well as some freedom to adjust our timeline in response to these problems. However, the performance of our robot's individual components is still essential to our team's success, and the following problems, if they present themselves, will require us to take significant actions to assure the success of our build process.

Risk Condition	Probability of Occurrence	Impact to Project	Change to Work Plan	Expected Date of Risk Decision
Robot not fast enough to escape wall of flame	Low/Moderate	High	Add front throwing mechanism to get animals out from the wall of flame, or re-gear our drive train and move to larger wheels to gain more speed.	Week 2/3
Robot unable to reliably re-find tape after IR following.	Moderate	Moderate	Use wheel encoders to assist IR following by encoding the turnaround and forcing straight line driving back to the tape.	Week 4
Unable to make size limitations because of machining tolerances.	High	Low/Moderate	Reduce wheel size and create lower profile arm top to reduce height.	Week 2
Unable to reliably find pet 5 with wheel encoders	Low	Moderate	Add QRD proximity sensor on arm to allow us to detect pet 5 for better pickup.	Week 2.5
Unable to fabricate arm to function precisely because of tolerances.	Low/Moderate	Moderate	Change arm materials to be fabricated with only the best tolerance machines (laser cut). Possibly 3D print the most precise pieces. Change to stiffer materials at the sacrifice of some weight or add lateral stabilization if "wobbling" is an issue	Week 1.5
Drive train gearings aren't properly proportioned to allow for enough speed or power	Moderate	Moderate	Because there are so many variables to change within the drive train, the best way to combat these problems would be to slightly over compensate in the safe direction (i.e. Basic function over speed)	Week 1/2
Unable to keep all pets within robot	Moderate	Low/Moderate	Redesign bucket to be larger to take all of the animals, possibly elongate chassis to accommodate bucket	Week 2

Other groups are successfully getting all pets or look like they are on course to do so, rendering us uncompetitive	Moderate	High	First discuss if we need to get pet 6 to meet our competition goals. If so, design and implement a simple front mounted pickup that can allow us to get pet 6 by driving towards the beacon then driving pickup into bucket. Use wheel encoding and IR following to assure safe return to the tape. Possibly change the pickup for pet 4 to be integrated into this design. Quickly produce and test this mechanism (under a week) so that we can rapidly decide if we will continue to pursue it.	Week 4
Group member goes out of contact/loses motivation/has some outside circumstance limiting their participation	Low/Moderate	Moderate/High	First have a discussion with the group member to see if it can be resolved, and possibly come to the course leadership for help with the resolution. If we have to lose some of their contributions, hold a rescheduling meeting and re-create our timeline for the project.	Any Time
Group Becomes Dysfunctional: Disorganization, confrontation or apathy	Low	High	Call for outside Mediation	Any Time

7 Task List

For a detailed Gantt Chart and task list, see Appendix 7.

8 Major Milestones and Testing Criteria

- Get working line follower
  The line follower is considered to be working if it can navigate the course in one direction in under 20 seconds
- Get a working arm
  The arm is considered working if it can trace the intended trajectory at least 10 times in succession, by the use of motor power only
- Adjust Drivetrain for testing
  A drivetrain that is reliable enough is one that makes it through the course under full load, even if it does not match the time constraints or is inconsistent. We still need a basic level of consistency so the threshold is 3 successful runs in a row.
- Reliable Wheel Encoding
  Wheel Encoding is considered reliable if the robot can execute the following loop successfully. Travel forward by 2 feet, turn 180˚, return to starting position, repeat 5 times.
- Reliable IR Beacon following
  IR Beacon following is considered reliable if the robot can follow the IR beacon in both directions at least 20 times. This is very critical to robots expected function and as a result the testing threshold is much higher
- Reliable Tape Following
  The tape following is considered reliable when the robot, with all of the competition loads added can navigate both the test track and the competition track, 30 times without failure. In addition to this there is a timing constraint on the navigation of competition track to meet the wall of fire challenge.
- Reliable Animal Pickup
  Reliable animal pickup constitutes the ability to pickup animals 1-5 at least 5 times in a row. We test this individually for each arm first before we eventually bring them together and try a continuous run.
- Complete course under time constraints
  Course completion is finish the course with all five animals safely stored at least 10 times in succession.
- Optimize drivetrain
  The optimization requirement will only be required if the previous goal is not achieved. This will be revisited based on new information that becomes available through course of the building of the robot.

Appendices

Appendix 1 – Motor Calculations

We can model a motor as a simple resistor connected in series with a voltage. As a result we can generate three motor models from the data provided on the Project Lab website for the Barber Coleman motors.

Model 1: Stall Torque and Stall Current

$$Internal\ Resistance\ = \frac{Input\ Voltage}{Stall\ Current}$$

$$K_t = \frac{Stall\ Torque}{Stall\ Current}$$

$$Torque = Stall\ Torque - \frac{{K_t}^2}{Internal\ Resistance} \times Angular\ Velocity$$

Model 2: Maximum Angular Velocity and No Load Current

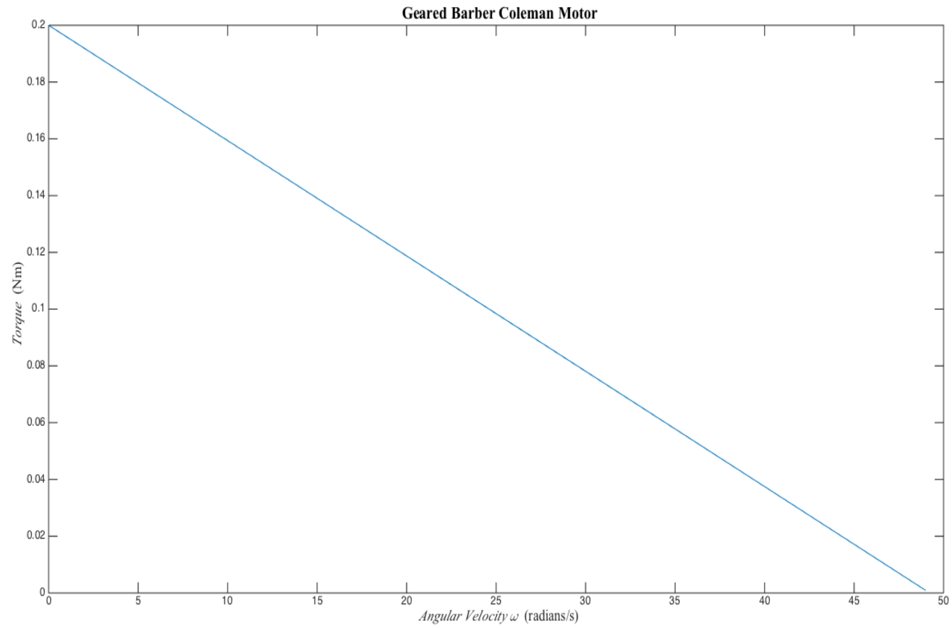
$$Internal\ Resistance\ = \frac{Input\ Voltage}{Stall\ Current}$$

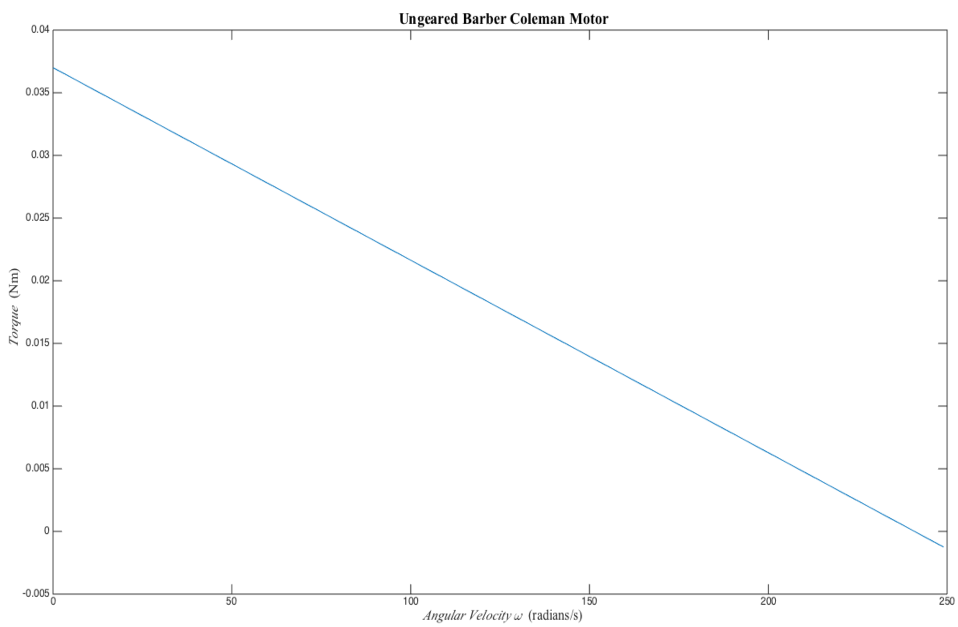
$$K_t = \frac{(Input\ Voltage - Internal\ Resistance \times No\ Load\ Current)}{Max\ Angular\ Frequency}$$

$$Torque = K_t \times \frac{InputVoltage}{InternalResistance} - \frac{{K_t}^2}{Internal\ Resistance} \times Angular\ Velocity$$

Model 3: Linear Fit: Stall Current and Maximum Angular Velocity

$$Torque = \frac{-Stall\ Torque}{Maximum\ Angular\ Velocity} \times Angular\ Velocity + Stall\ Torque$$


Geared Barber Coleman Motor
Torque (Nm)
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
0 5 10 15 20 25 30 35 40 45 50
Angular Velocity ω (radians/s)

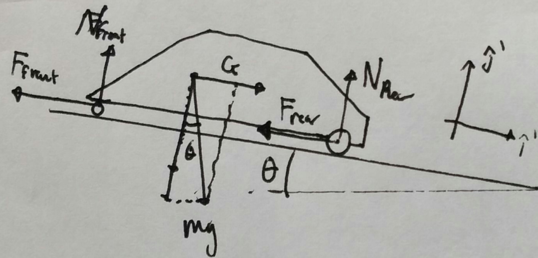

Ungeared Barber Coleman Motor
Torque (Nm)
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
-0.005
0 50 100 150 200 250
Angular Velocity ω (radians/s)

Appendix 2 – Drive Calculations

For Robot Moving Uphill:

Assume Front wheels undriven so $F_{front} = 0$

$N_{front} + N_{rear} = mg\cos\theta$ as there is no vertical acceleration



The relative magnitude of N_{rear} & N_{front} depend on centre of gravity. From solidworks & preliminary estimates, assume rear takes 80% of load, so:

$$N_{rear} = 0.8\,mg\cos\theta \implies F_{rear} = \mu_s\, 0.8\,mg\cos\theta$$

assume static (ie. RWS) for max friction.

Assume 2 back drive motors outputting $F_{motor} = \frac{\text{Torque} \times \text{Gear Ratio}}{R_{wheel}}$

Total motor power can be $2 * F_{motor}$, however if $2F_{motor} > F_{Rear}$, then the wheel slips, so the most force that will be useful is $2F_{motor} = F_{Rear}$. If $2F_{motor} < F_{Rear}$, our force is limited by $2F_{motor}$. So for any case our force that propels the robot is the ~~minimum~~ minimum of $2F_{motor}$ and F_{Rear}

Then, in our direction of travel, we see that our acceleration is:

$$a = \frac{\min\{2F_{motor}, F_{Rear}\} - mg\sin\theta}{m}$$

Appendix 3 – Arm Power Calculations

Arm power transfer calculations

Torque requirements for the arm

Constants

$$g = 9.81\ m/s^2$$

$$Motor\ torque = \frac{0.037}{2}\ Nm$$

$$Motor\ Speed = \frac{2300}{2}\ RPM$$

$$Mass\ animal = 0.1\ kg$$

$$Mass\ arm\ end = 0.1\ kg$$

$$Mass\ arm = 0.039\ kg$$

$$Length\ arm = 0.245\ m$$

Torque and Speed Calculation

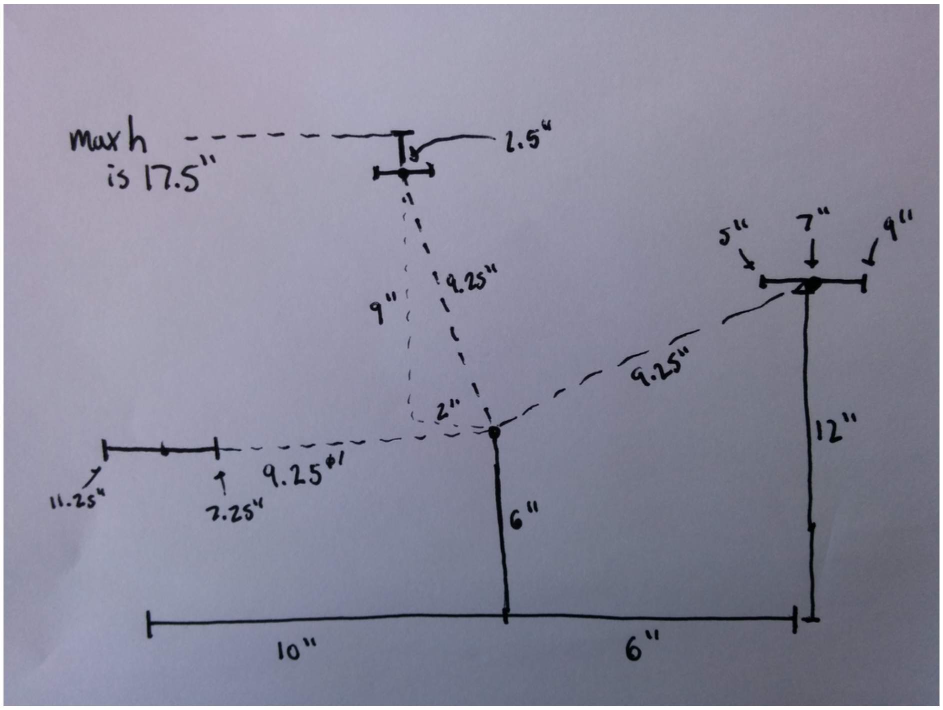
$$Torque = (Mass\ animal + Mass\ arm\ end + \frac{Mass\ arm}{2}) \times g \times Length\ arm$$

$$Torque = 74.7\ Nm$$

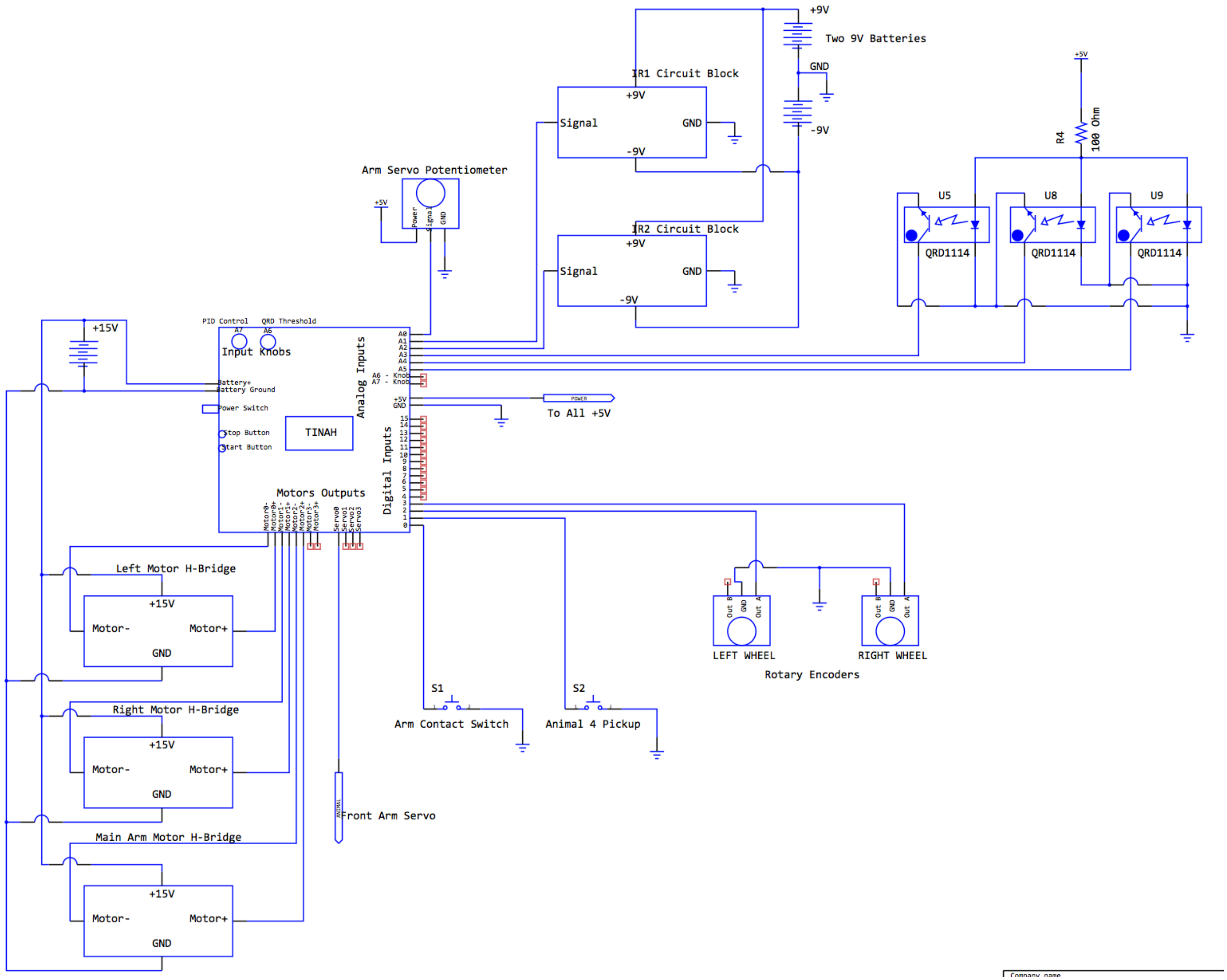
$$Needed\ Ratio = \frac{Torque}{Motor\ torque} = 29$$

$$Speed = \frac{Motor\ Speed}{Needed\ Ratio} = 40\ RPM$$

Appendix 4 – Robot Starting Geometry

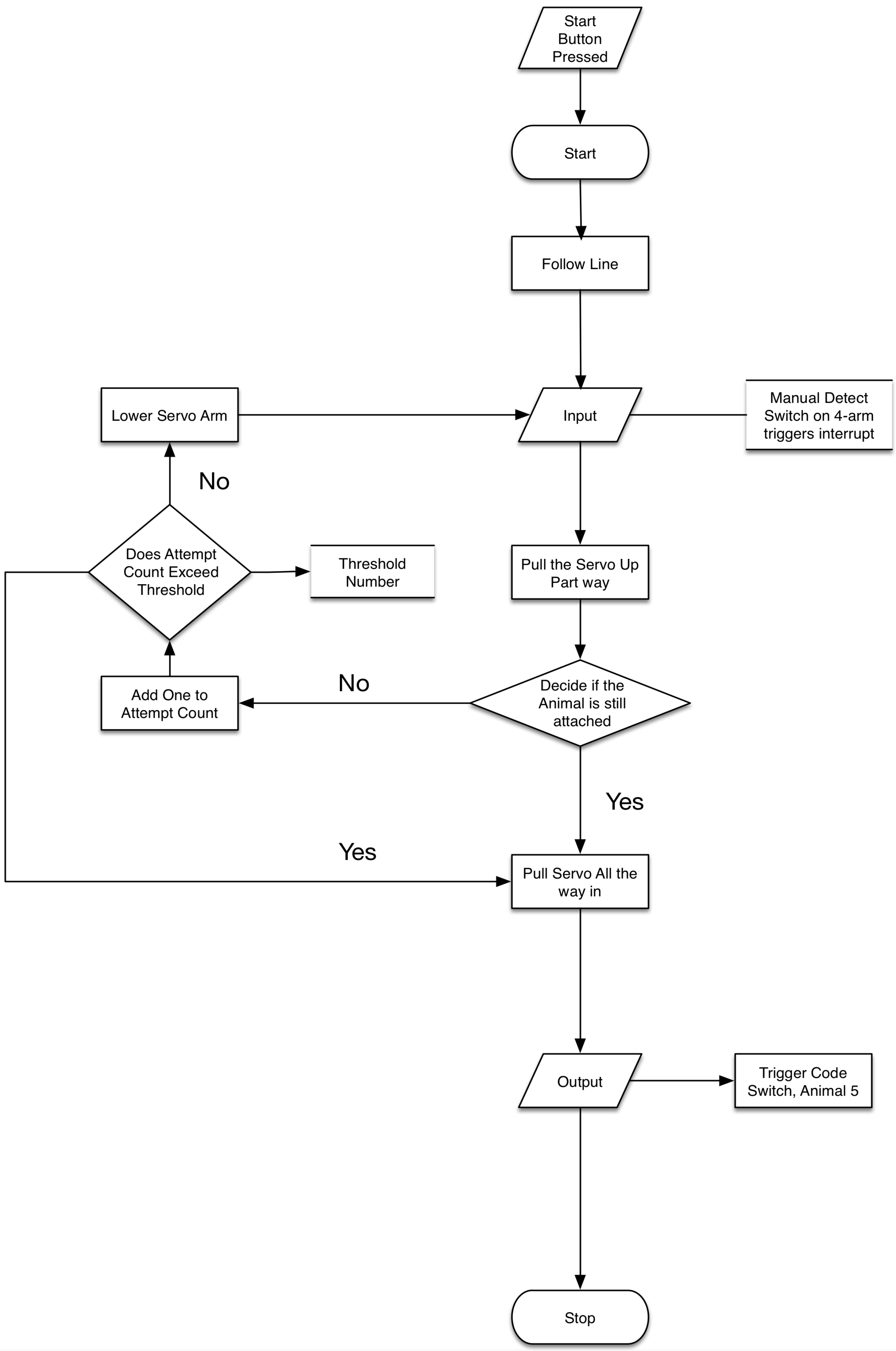


Appendix 5 – Circuit Overview

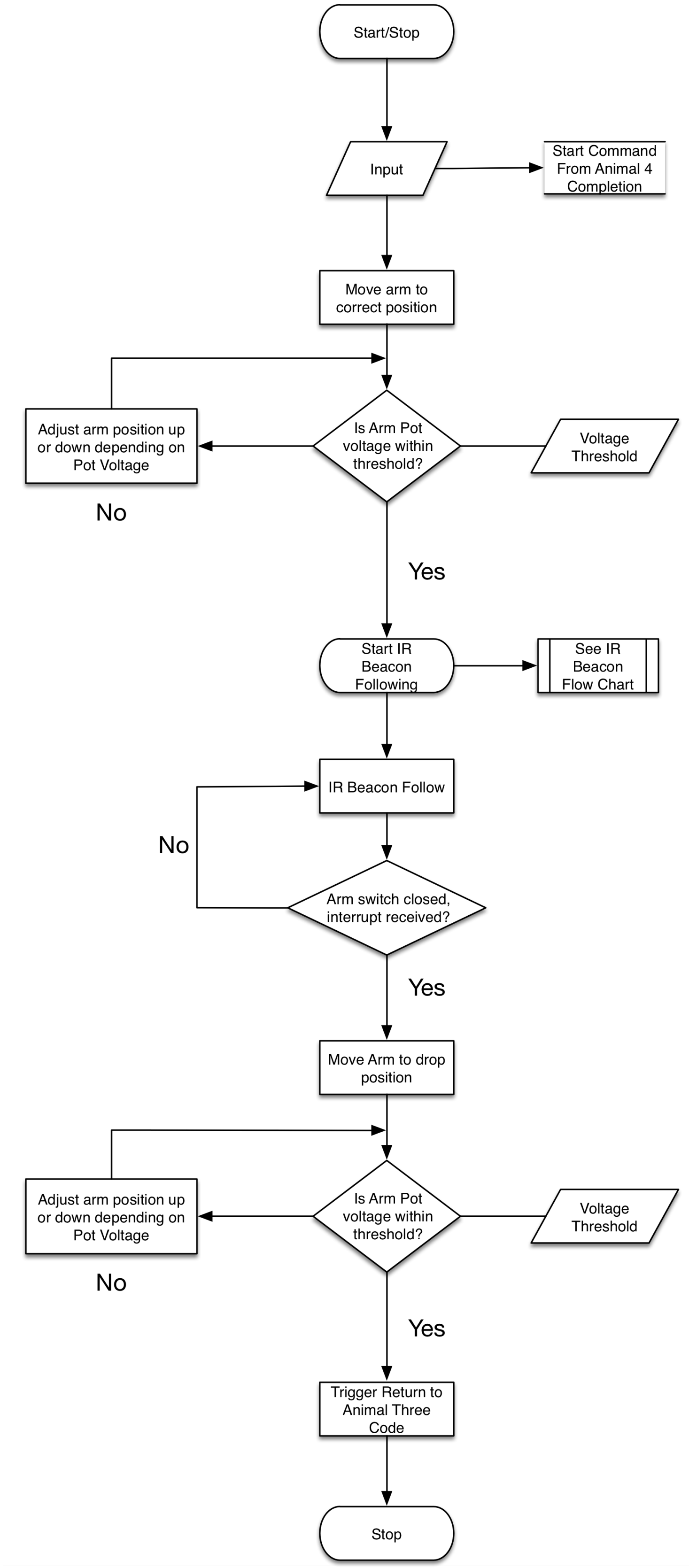


Appendix 6 - Flowcharts

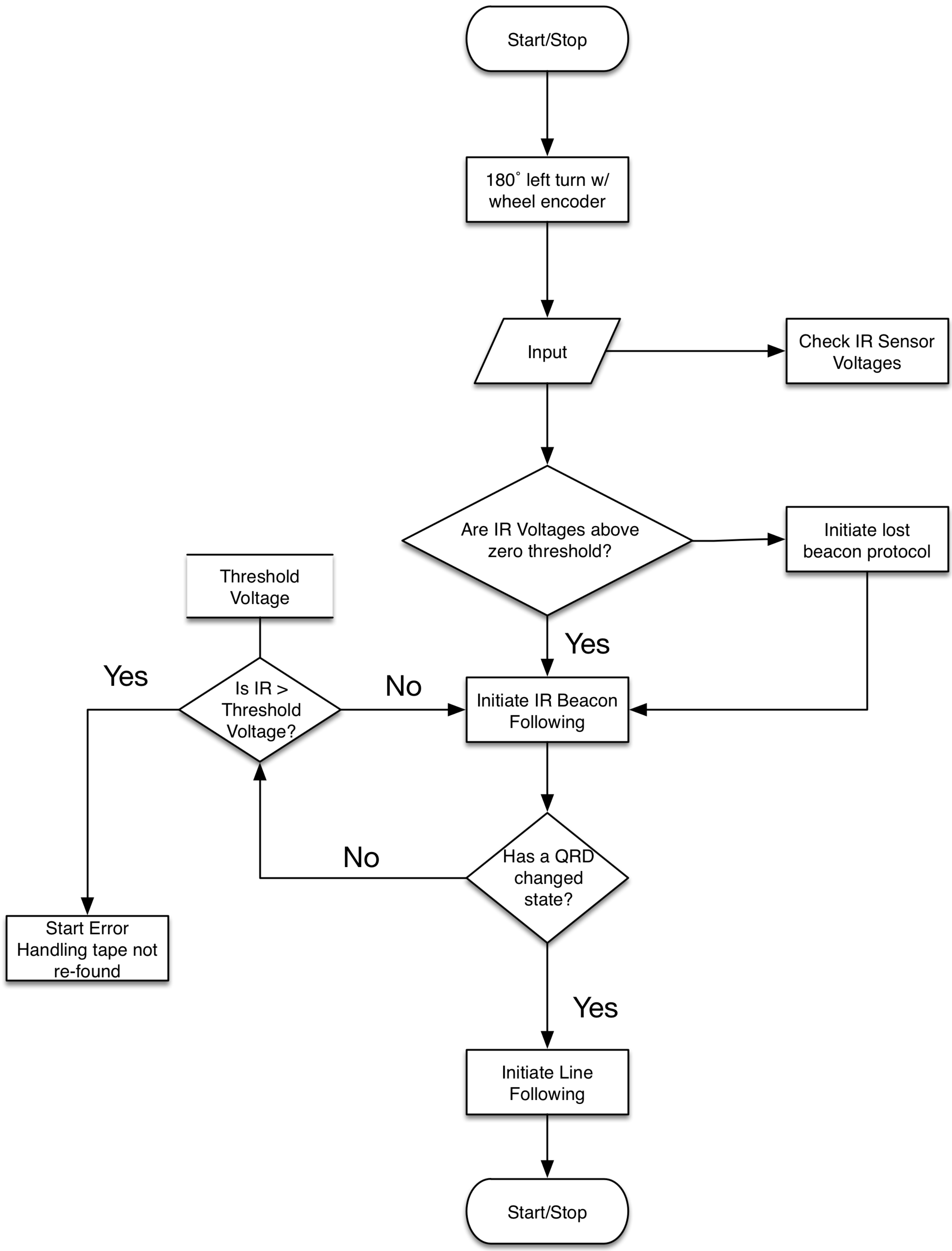
6a) Animal 4 Pickup



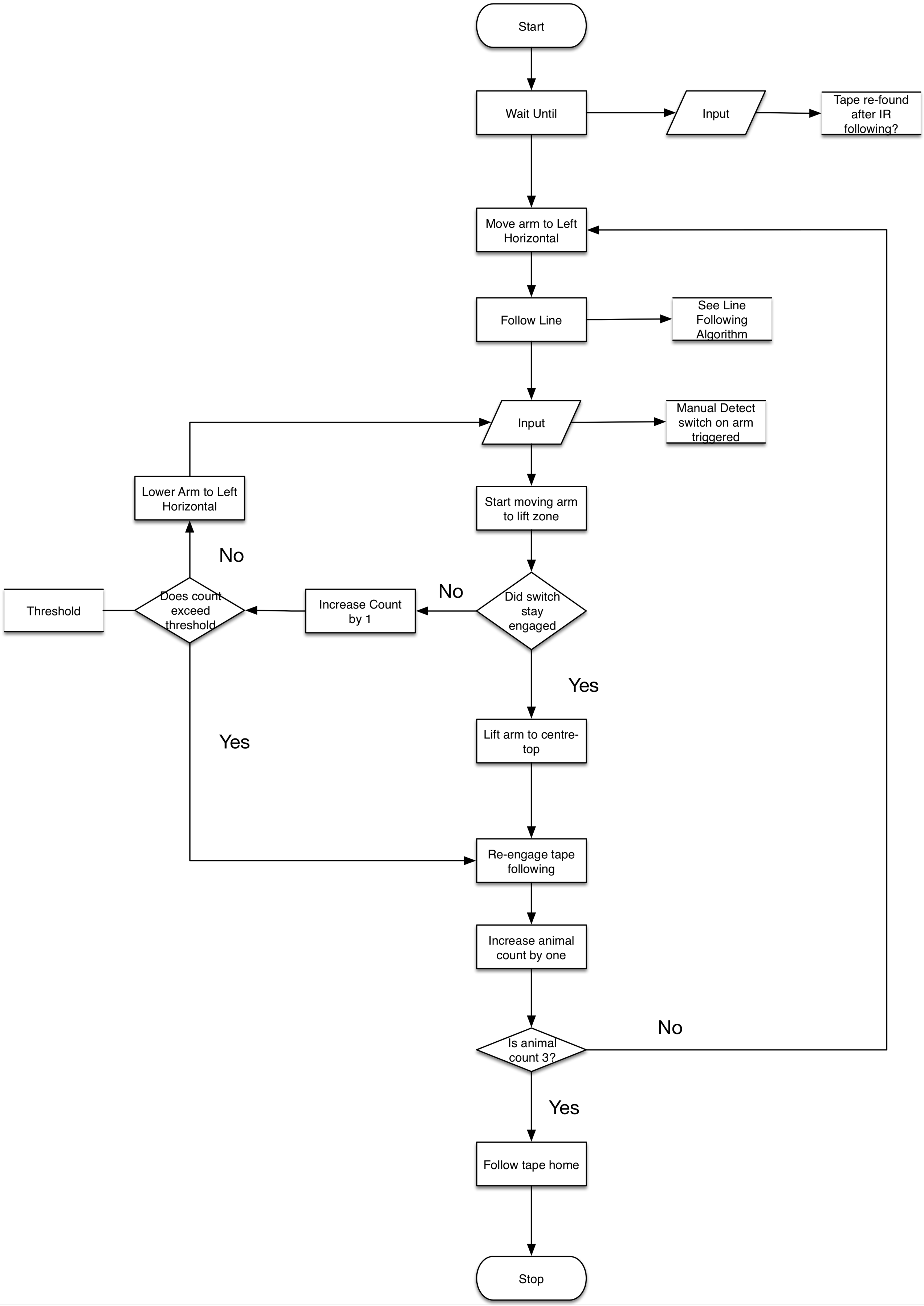
6b) Animal 5 Pickup



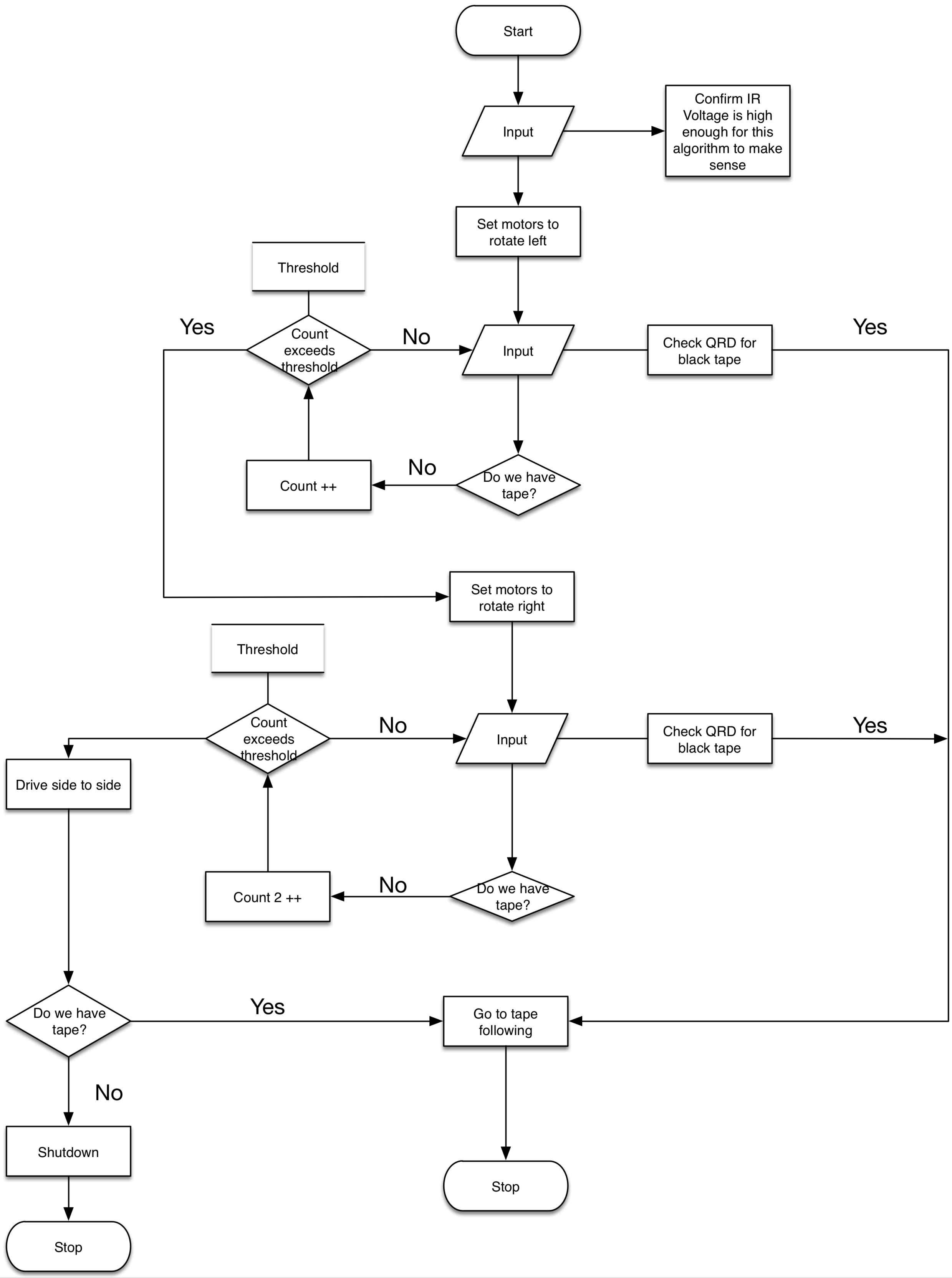
6c) Return to Animal 3



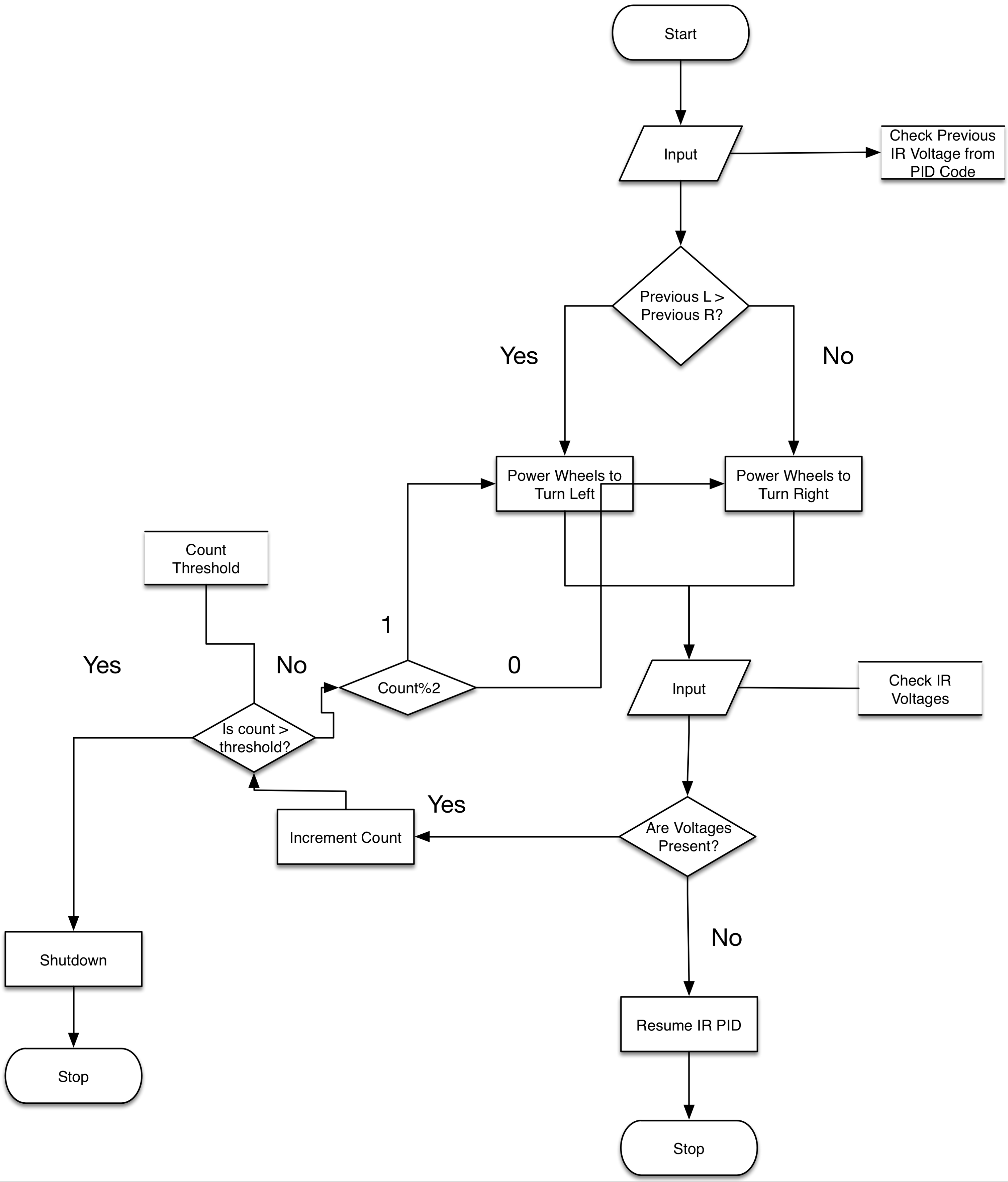
6d) Animal 3, 2, 1 Algorithm



6e) Error Handling: Tape Not Found

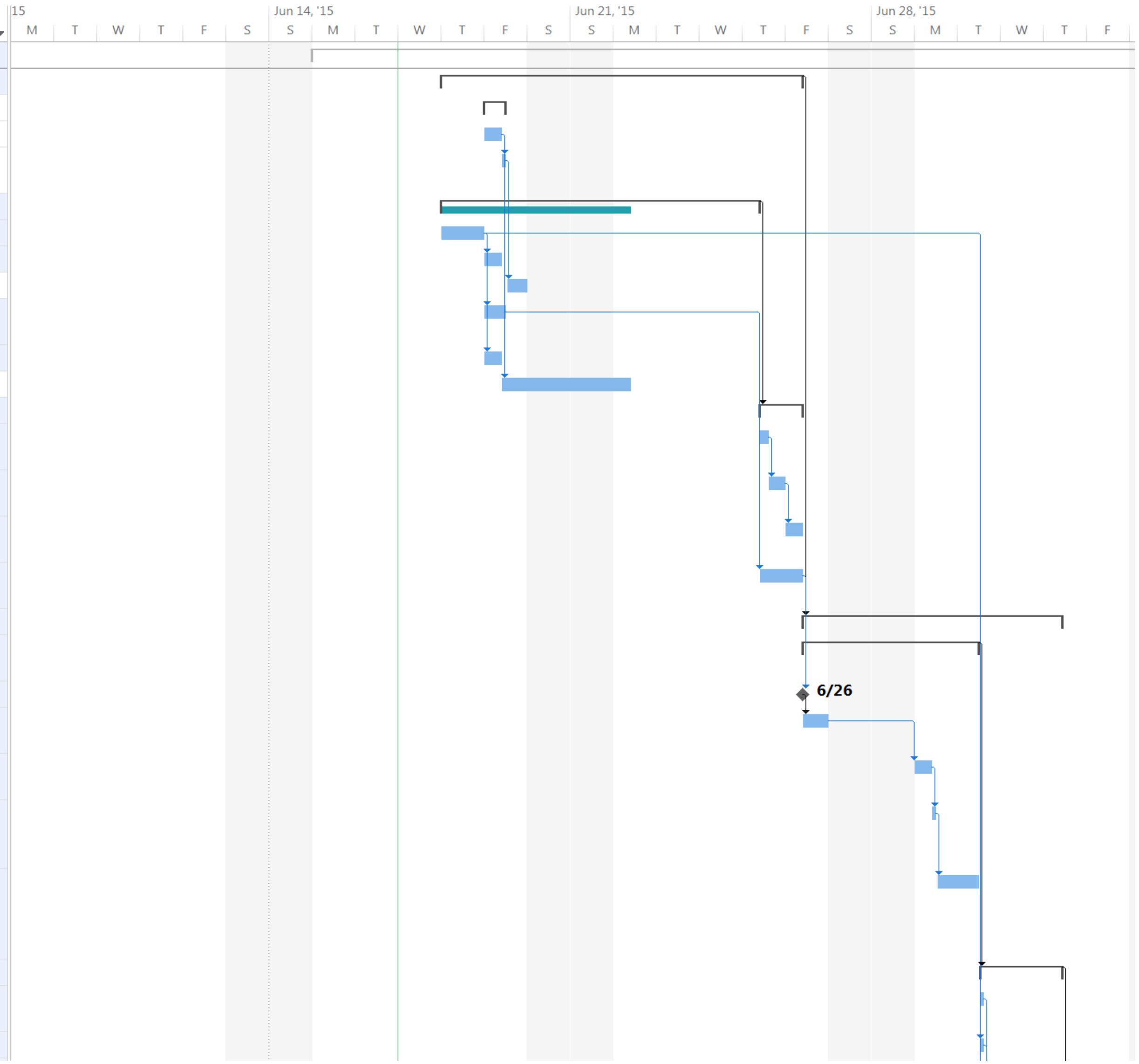


6f) Error Handling: Lost Beacon



Appendix 7 – Gantt Chart and Task List

	Task Name	Duration	Start	Finish
0	RobotGanttChart	23.6 days?	Mon 6/15/15	Thu 7/16/15
1	Building Prototype Chasis	6.25 days	Thu 6/18/15	Fri 6/26/15
2	Admin Task	0.5 days	Fri 6/19/15	Fri 6/19/15
3	Buy/source rear wheels	0.25 days	Fri 6/19/15	Fri 6/19/15
4	Buy/source front swivel wheels	0.25 days	Fri 6/19/15	Fri 6/19/15
5	Mechanical Tasks	5.25 days	Thu 6/18/15	Thu 6/25/15
6	Cut, bend aluminum	1 day	Thu 6/18/15	Thu 6/18/15
7	Make the QRD bracket	0.25 days	Fri 6/19/15	Fri 6/19/15
8	Laser Cut Gears	0.5 days	Fri 6/19/15	Fri 6/19/15
9	Make motor mounts for drive wheels	0.5 days	Fri 6/19/15	Fri 6/19/15
10	Make basic TINAH mount	0.25 days	Fri 6/19/15	Fri 6/19/15
11	Machine wheel axles	1 day	Fri 6/19/15	Mon 6/22/15
12	Electrical Tasks	1 day	Thu 6/25/15	Fri 6/26/15
13	Make and test 6 QRDs (2 sets of 3 on PCB)	0.5 days	Thu 6/25/15	Thu 6/25/15
14	Record the thresholds of each of the QRD	0.25 days	Thu 6/25/15	Thu 6/25/15
15	Create wiring bundles and QRD holding PCBs	0.25 days	Fri 6/26/15	Fri 6/26/15
16	Create and test H-Bridge Circuits for wheel motors	1 day	Thu 6/25/15	Fri 6/26/15
17	Building Arm	4.1 days?	Fri 6/26/15	Thu 7/2/15
18	Constructing Servoed Motor	2.25 days?	Fri 6/26/15	Tue 6/30/15
19	Construct H-Bridge Circuit	0 days	Fri 6/26/15	Fri 6/26/15
20	Create pulley to connect to potentiometer	0.75 days	Fri 6/26/15	Fri 6/26/15
21	Build potentiometer feedback to TINAH	0.25 days	Mon 6/29/15	Mon 6/29/15
22	Find potentiometer values for known arm positions	0.25 days	Mon 6/29/15	Mon 6/29/15
23	Create PID code to get motor to re-establish to a given potentiometer value	1 day?	Mon 6/29/15	Tue 6/30/15
24	Constructing Physical Arm	1.85 days	Tue 6/30/15	Thu 7/2/15
25	Laser cut the supports for the arm out of acrylic	0.25 days	Tue 6/30/15	Tue 6/30/15
26	3D print cam	0.25 days	Tue 6/30/15	Tue 6/30/15



27	Machine the pivot pins out of brass	0.5 days	Tue 6/30/15	Wed 7/1/15
28	Bend sheet metal base	0.5 days	Tue 6/30/15	Tue 6/30/15
29	3D print upper arm bracket	0.1 days	Tue 6/30/15	Tue 6/30/15
30	Mount arm	0.75 days	Tue 6/30/15	Wed 7/1/15
31	Reconfigure line following	0.75 days	Wed 7/1/15	Thu 7/2/15
32	**Building IR Follower**	**1 day**	**Thu 7/2/15**	**Fri 7/3/15**
33	Build and test to IR filters	1 day	Thu 7/2/15	Fri 7/3/15
34	Tune PID control on the circuits to have fast and accurate following	1 day	Thu 7/2/15	Fri 7/3/15
35	**Build Bucket**	**0.75 days**	**Mon 6/15/15**	**Mon 6/15/15**
36	Generate flat pattern	0.5 days	Mon 6/15/15	Mon 6/15/15
37	Laser cut bucket out of wood	0.25 days	Mon 6/15/15	Mon 6/15/15
38	**Build Chasis**	**7 days**	**Fri 7/3/15**	**Tue 7/14/15**
39	**Mechanical Tasks**	**3 days**	**Fri 7/3/15**	**Wed 7/8/15**
40	Generate flat pattern	1 day	Fri 7/3/15	Mon 7/6/15
41	Water jet cut chasis out of alumnium	1 day	Mon 7/6/15	Tue 7/7/15
42	Laser cut final gears and sizes out	2 days	Fri 7/3/15	Tue 7/7/15
43	Attach final wheels	1 day	Tue 7/7/15	Wed 7/8/15
44	Attach QRD mounts	0.5 days	Tue 7/7/15	Tue 7/7/15
45	Attach motor mounts	0.5 days	Tue 7/7/15	Tue 7/7/15
46	Build final bucket out of wire mesh	1 day	Tue 7/7/15	Wed 7/8/15
47	**Electrical Tasks**	**4 days**	**Wed 7/8/15**	**Tue 7/14/15**
48	Attach all of our circuits to robot	2 days	Wed 7/8/15	Fri 7/10/15
49	Tune the circuits on the robot	2 days	Fri 7/10/15	Tue 7/14/15
50	**Closing**	**2.25 days**	**Tue 7/14/15**	**Thu 7/16/15**
51	**Cleanup prototyping**	**1.25 days**	**Tue 7/14/15**	**Wed 7/15/15**
52	Paint	0.5 days	Tue 7/14/15	Tue 7/14/15
53	Debur	0.5 days	Tue 7/14/15	Wed 7/15/15
54	Inscribe	0.25 days	Wed 7/15/15	Wed 7/15/15
55	**Celebrate**	**1 day**	**Wed 7/15/15**	**Thu 7/16/15**
56	Party	1 day	Wed 7/15/15	Thu 7/16/15