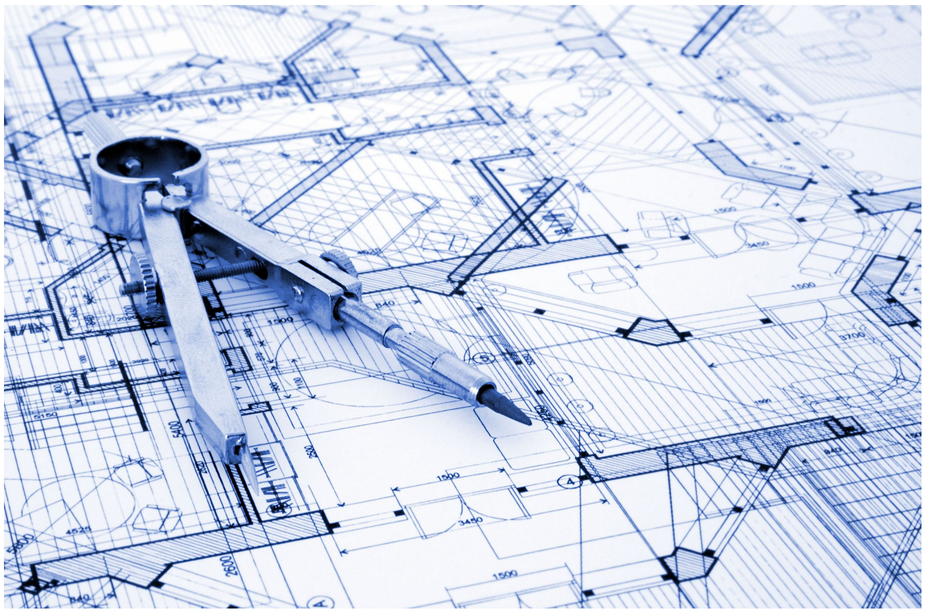


Engineering Notebook

Jacob M. Kiggins

01/25/2016 — 05/26/2016

Contents

1 Problem Analysis

Problem Statement In this years competition our team will design and build an autonomous robot that can deposit jewelry (rings) to one dozen boxes along the parade route on a specified track. Our team must design and build an autonomous robot(s) that can deposit one ring in each of the 12 boxes located along the parade route on the track. Our robot(s) will have a maximum time of 90 seconds in each of our four allotted trials. The robots must begin within an 8 X 12 X 10 high size limit but may expand to any size during a trial.

1.1 Inital Reactions

The track (Figure: 1) has a particular topology which the sensor system and robot may take advantage of.

- The walls enclose the route
- The black line follows near the boxes
- Overall, the track is symmetrical
- The boxes are all red
- The boxes are all in corners

Navigation of the course would be much easier to do if the wooden walls did not impede travel. The start and size constraint ”boxes” are large enough to accommodate multiple robots. Multiple robots will most likely have an advantage over a single robot. We will analyze this idea as we progress.

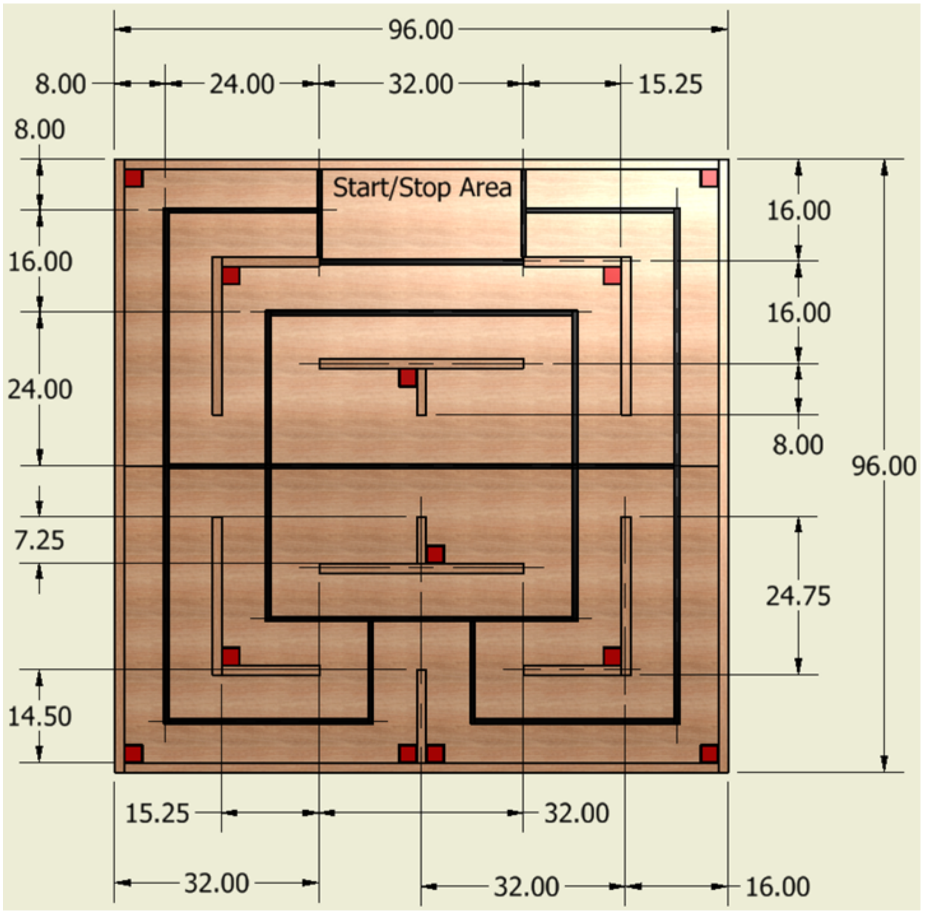


Figure 1: Parade Route Track Specs

1.2 Rules and Scoring

Allowable Energy Sources Any energy source is allowed as long as it is completely contained within the robot and does not create or emit any gaseous, liquid, or solid emissions. Energy sources must not present any safety hazards to participants or spectators.

Structure The robot must fit inside a box with vertical sides having inside dimensions of 8.0 X 12.0 and have a maximum height of 10.0. The robot may expand to any size after the start of a trial.

Components Team members using materials which are commonly available to the general public must perform all fabrication. Use of commercially available vehicles, robots, or entire kits such as RC cars, Legos, K-nex, Fischer-Technics, Parallax or erector sets may not be used. The use of Lego Mindstorm micro-controller bricks are prohibited. Individual components from these cars, robots, or kits (except the Mindstrorm Brick) may be integrated into a teams robot as long as the majority of the robots components are not from the same car, robot, or kit source. The cost of purchasing all components must not exceed $400.

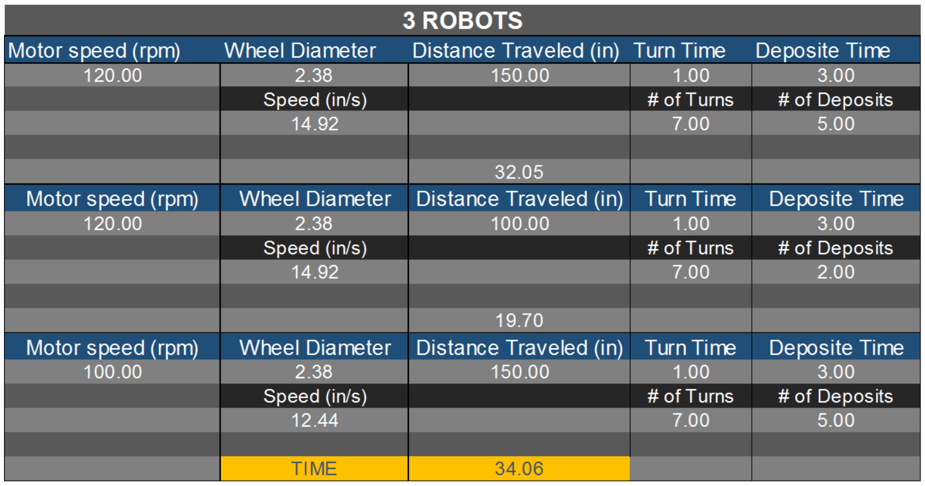
General Scoring and Rules

- Five points will be awarded for each ring deposited in a box (one ring per box)
- A twenty point bonus will be awarded for a robot that deposits a ring in each of six boxes and returns to the start/stop area
- If a robot completes a perfect run in under ninety seconds, the remaining time will be added to that teams score.
- The trial is ended if ninety seconds has passed or...
- The robot stops moving and shows no signs of continuing or...
- The robot deposits twelve rings and returns to the start/stop area
- A robot must deposit exactly one ring in a box to receive credit for that box

There will be four trials and a exhibit poster session. A teams score will be the sum of their four trials and their poster session (maximum of 120 points)

1.3 Multiple Robot Analysis

Assumptions This analysis assumes a robot that may navigate it's way around the track by following the lines and/or walls. There are also a few assumed values about robot maneuverability which are outlined in figure 2



3 ROBOTS				
Motor speed (rpm)	Wheel Diameter	Distance Traveled (in)	Turn Time	Deposite Time
120.00	2.38	150.00	1.00	3.00
	Speed (in/s)		# of Turns	# of Deposits
	14.92		7.00	5.00
		32.05		
Motor speed (rpm)	Wheel Diameter	Distance Traveled (in)	Turn Time	Deposite Time
120.00	2.38	100.00	1.00	3.00
	Speed (in/s)		# of Turns	# of Deposits
	14.92		7.00	2.00
		19.70		
Motor speed (rpm)	Wheel Diameter	Distance Traveled (in)	Turn Time	Deposite Time
100.00	2.38	150.00	1.00	3.00
	Speed (in/s)		# of Turns	# of Deposits
	12.44		7.00	5.00
	TIME	34.06		

Figure 2: Three Robot Analysis

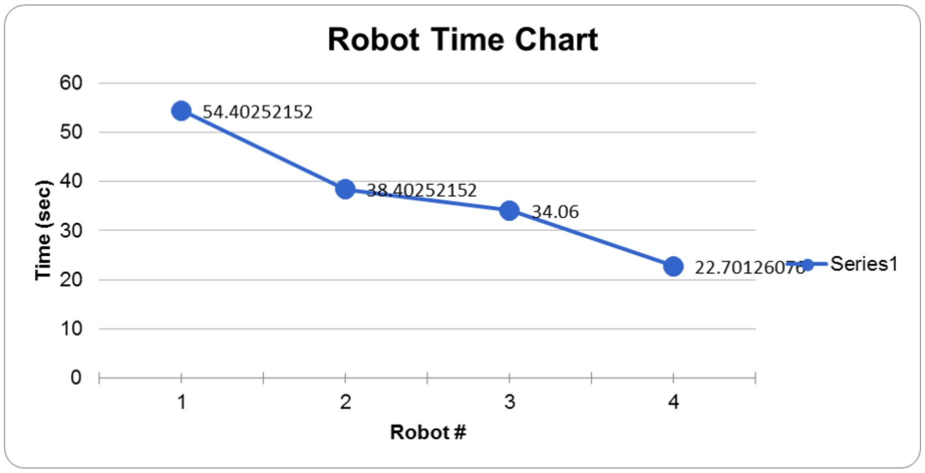


Figure 3: Multiple Robot Timing Graph

SIGNATURE: *Jacob M. Kiggins* DATE: 2/10/2016

2 Game Strategy

Overview Our basic strategy involves the use of a line sensor mounted on the bottom of a tank style drive robot. We will use the line to navigate the track and deposit rings from off the side of the robot. We will also integrate a distance sensor. This when coupled with our quadrature encoders should allow us accurate off-the-line navigation allowing us to deposit rings over the wall as well as make complex maneuvers.

2.1 Mechanical

Requirements

- Quick assembly/dis-assembly
- Short manufacture time
- Highly durable under when subject to impact
- Secure when assembled
- Reliable ring drop mechanism with large margin of error in terms of robot location with respect to the box

Solution We will use a 3D printed chassis with pre-made mounting holes and a .25" thickness. Our drive-train will consists of two Lego motors and an omni-wheel. Our chassis will be designed such that all drive-train components snap into place easily.

2.2 Electrical

Requirements

- Interface with a QTR-8 line sensor(Figure 4)
- Interface with a ZX IR distance sensor (Figure 6)
- Circuitry to interface with quadrature encoders for off-the-line navagation
- Interface with servo, used to drop rings

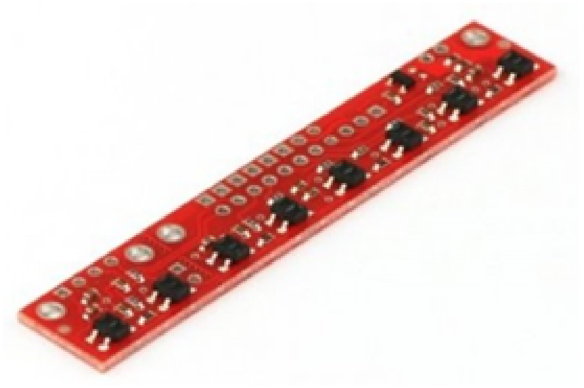


Figure 4: QTR-8 sensor

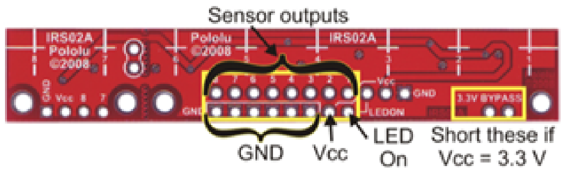


Figure 5: QTR-8 Bottom View

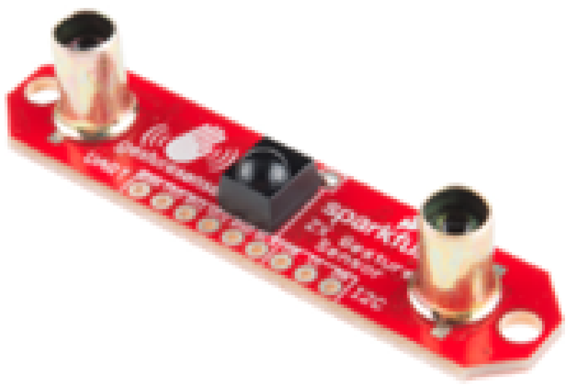


Figure 6: Sparkfun ZX distance sensor

Soultion We will design an engraved circuit board to route all connections around. Header pins will be extensively used. Components should be replaceable without de-soldering. High voltage rails will not be close to low voltage rails to minimize the likelihood of chip damage.

2.3 Programming

Requirements

- Program should be divided into two independent components
  - Physical actions the robot can perform
  - Sensor feedback which can control the flow of these actions
- No interrupts except those required for communications (I2C) and the quadrature encoders
- The route should be mapped out with a series of blocking function calls
- The program needs to be flexible to changes in physical configuration

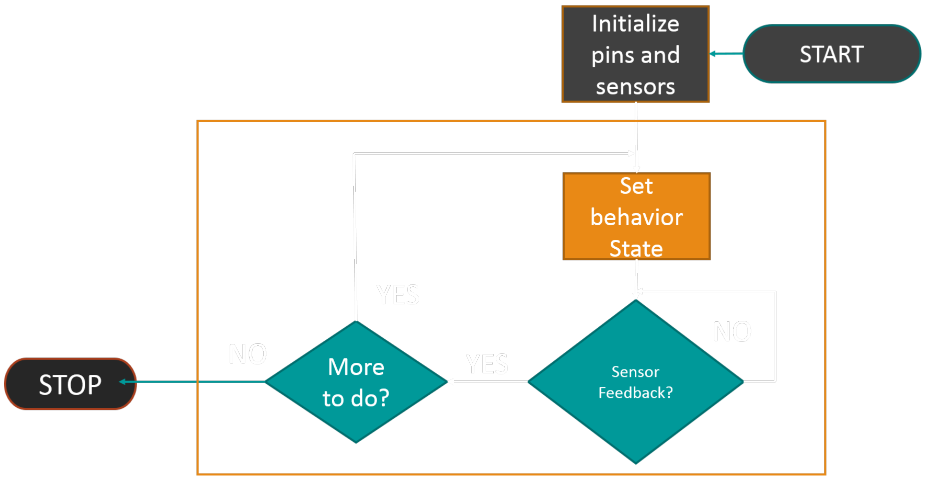


Figure 7: Basic program flow chart

3 Encoder Navigation

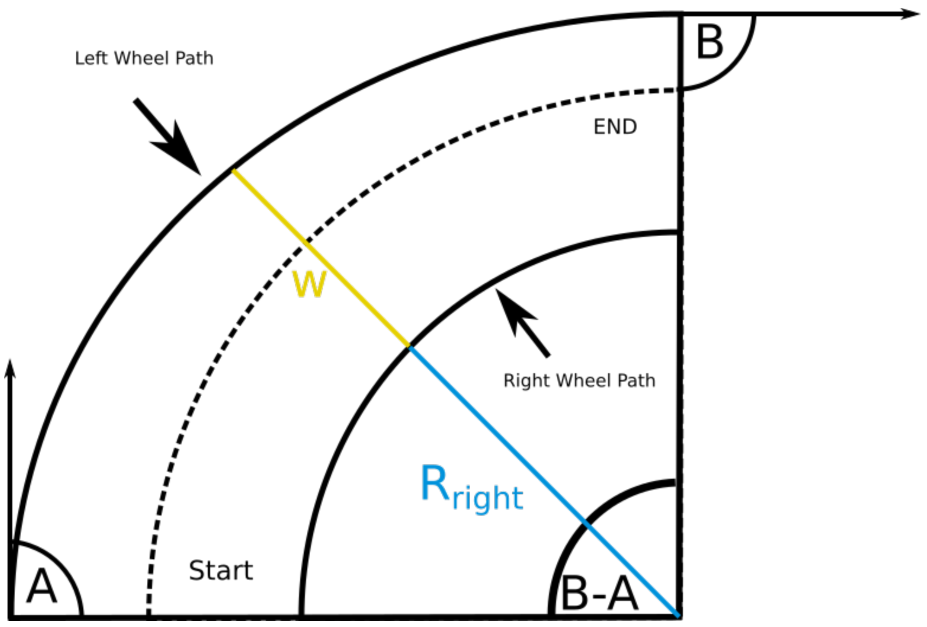


Figure 8: Navigation Diagram

Motivation Since this robot is driven with two independent wheels on either side of it's frame a certain theoretical model must be created to describe it's motion from available inputs. Of these available inputs the quadrature encoders on each wheel will be the most effective in determining movement. The characteristics we are most interested in are the change in distance and the change in heading. This model will be developed below

Let W = wheel base in cm, $R_{left} = W + R_{right}$, A be the start angle (usually zero) and B be the ending angle, We also define $R = R_{right} + W/2$

Let T_{left} be the encoder tick count on the left wheel, T_{right} be the encoder tick count on the right wheel, both since the position was last updated

3.1 Radius Development

$$\frac{R_{left}}{R_{right}} = \frac{T_{left}}{T_{right}} = \frac{R + W/2}{R - W/2}$$

$$T_{left}R - T_{left}W/2 = T_{right}R + T_{right}W/2 \tag{1}$$

$$R(T_{left} - T_{right}) = \frac{W}{2}(T_{left} + T_{right})$$
$$R = \frac{W}{2} * \frac{T_{left} + T_{right}}{T_{left} - T_{right}} \tag{2}$$

3.2 Angle Development

$$\begin{aligned} da &= B - A \\ &= \frac{S}{R} = \frac{T_{left} * CMPT}{R + W/2} = \frac{T_{right} * CMPT}{R - W/2} \\ &= \frac{T_{average}}{R} \end{aligned} \tag{3}$$

Where $CMPT$ is the centimeters per encoder tick ,and S is arc length

3.3 Conclusions

With this mathematical model, given accurate encoder counting and minimal wheel slip, should provide reliable off-the-line navigation. Also, this model can provide feedback to the robot when it is moving in an arc-like path or simply rotating.

4 Prototyping

Lego Robot Though the robot is mostly finished in solidworks (1st rev. at least) We need a test robot to move along the program development. To accomplish this we designed and built a lego version (10) of our robot which is almost dimensional-ly equivalent to the final bot.

The main purpose of this robot is to interface with the encoders and show that they can be a reliable control point. We have also created a breadboard to prototype our circuit design. Throughout discussion we have considered the idea of resin-ing the breadboard to create a more permanent solution.

Cylinder Deposit System We have been able to 3D print our first iteration of the Cylinder deposit system (9). It is designed to hold the rings in a stacked fashion and Our custom "Sploosher" arm will eject the rings left or right.

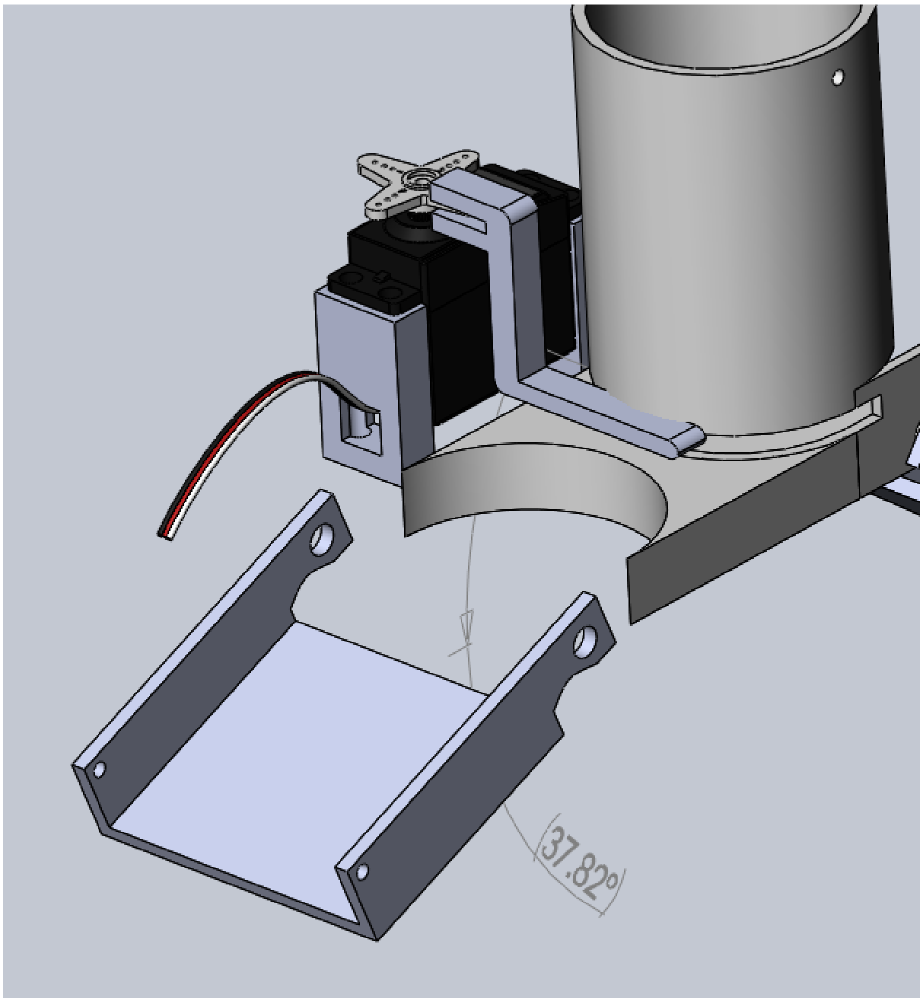


Figure 9: Cylinder Ring Deposit System

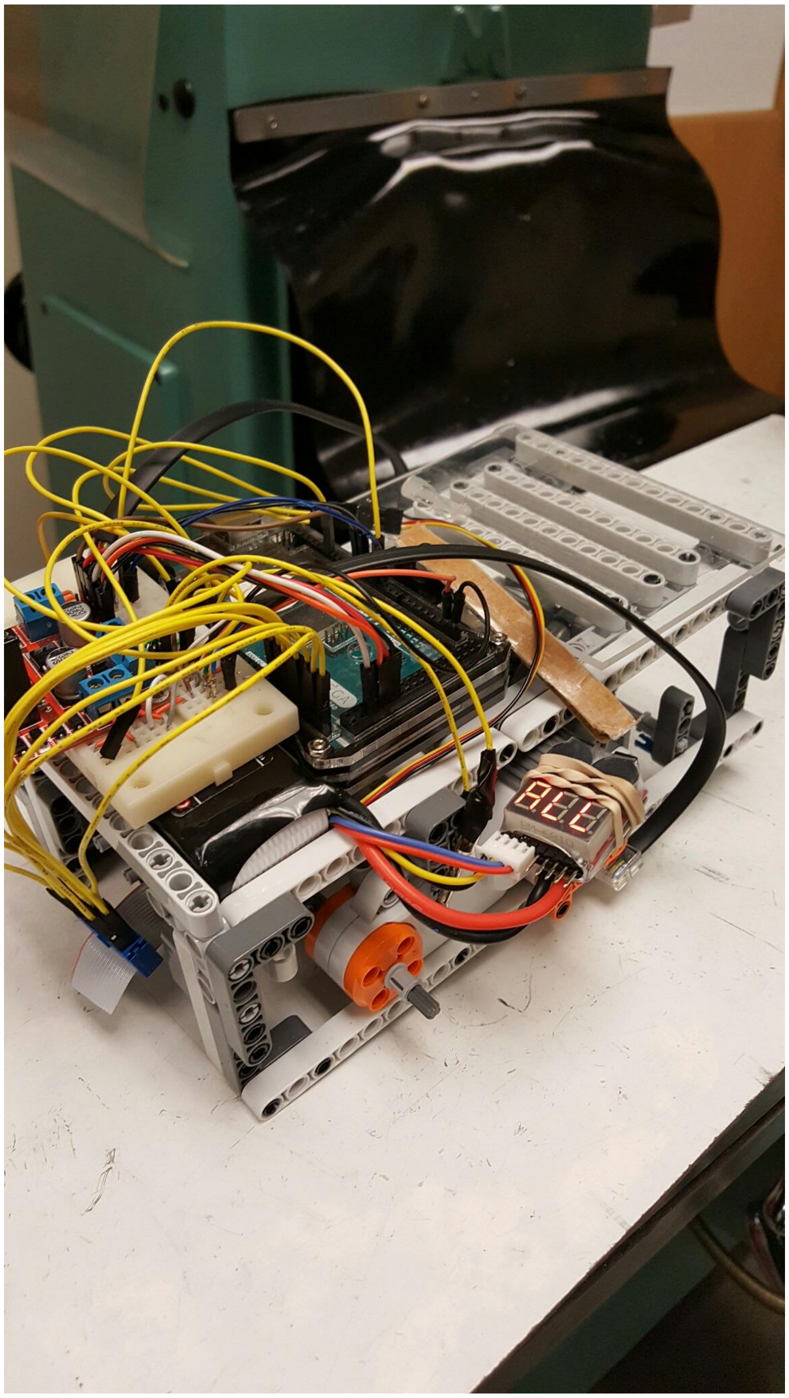


Figure 10: Lego Version of Robot

SIGNATURE: *Jacob M. Kiggins* DATE: 3/16/2016

4.1 Prototype Results

The Good... The Lego motors have proved dependable and the overall drive system is solid. The QTR-8 sensor (4) is very capable and the robot itself shields the sensor from ambient light. Our ring deposit system is very solid, though it shoots rings out the top occasionally. This is most likely due to the servo "Kicking" when the power is initially applied, this is of course an electrical problem.

The Bad... The encoder navigation is less reliable than we had hoped. Some possible causes are the inability for the Arduino Mega to keep up with the 720 ticks/s x 2 motors. Another is the small-angle-error which plagues the method we use to calculate the robot's "heading". The H-Bridge we are using does not have a break mode so there are a lot of sketchy time delays where the motors have opposite power applied.

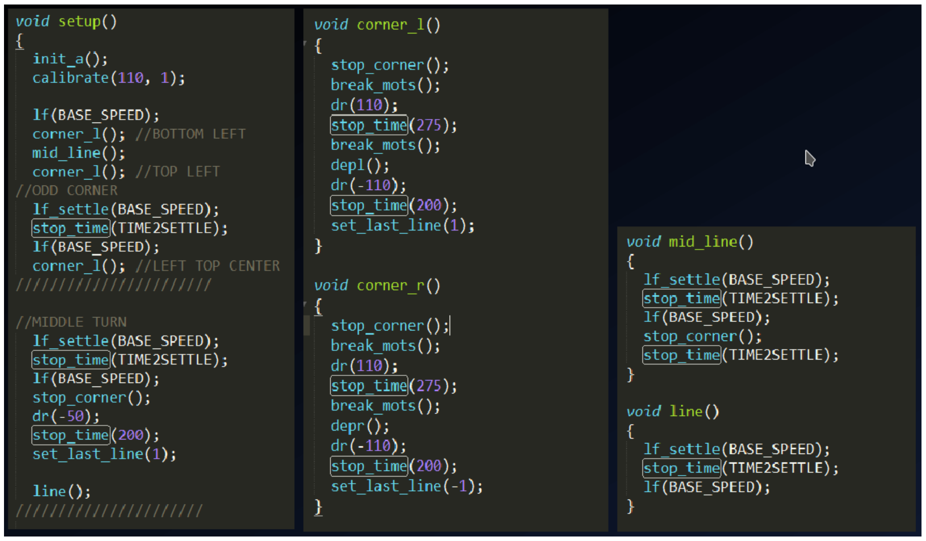
4.2 Prototype conclusions

Going Forward... We Are going switch to the teensy 3.2 micro-controller instead of an arduino mega. The teensy has a clock speed of 97MHz (as opposed to 16 MHz) and built in hardware quadrature encoder counters. This should give us a much more accuracy in all aspects of the robot. We also plan to switch to a polulu H-Bridge that is much smaller and has a break mode. To combat the navigation errors a software band-pass filter may be used on the heading, only allowing "reasonable" changes to occor on the heading.

5 Program Structure

5.1 Mapping the Route through the Track

Motivation Due to differences in each team member's experience with programming. We developed a system in which every action or decision the robot makes can be reduced to a series of function calls. This above all else allows each member of the team to work on the program if nessisary. Also with this model the path we take around the track can be changed very quickly. This model can be seen in Figure 11.



```
void setup()
{
  init_a();
  calibrate(110, 1);

  lf(BASE_SPEED);
  corner_l(); //BOTTOM LEFT
  mid_line();
  corner_l(); //TOP LEFT
//ODD CORNER
  lf_settle(BASE_SPEED);
  stop_time(TIME2SETTLE);
  lf(BASE_SPEED);
  corner_l(); //LEFT TOP CENTER
///////////////////////

//MIDDLE TURN
  lf_settle(BASE_SPEED);
  stop_time(TIME2SETTLE);
  lf(BASE_SPEED);
  stop_corner();
  dr(-50);
  stop_time(200);
  set_last_line(1);

  line();
//////////////////////
```

```
void corner_l()
{
  stop_corner();
  break_mots();
  dr(110);
  stop_time(275);
  break_mots();
  depl();
  dr(-110);
  stop_time(200);
  set_last_line(1);
}

void corner_r()
{
  stop_corner();
  break_mots();
  dr(110);
  stop_time(275);
  break_mots();
  depr();
  dr(-110);
  stop_time(200);
  set_last_line(-1);
}
```

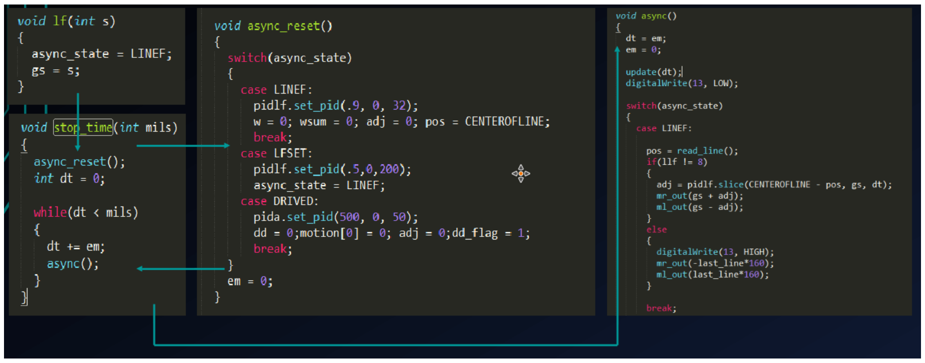
```
void mid_line()
{
  lf_settle(BASE_SPEED);
  stop_time(TIME2SETTLE);
  lf(BASE_SPEED);
  stop_corner();
  stop_time(TIME2SETTLE);
}

void line()
{
  lf_settle(BASE_SPEED);
  stop_time(TIME2SETTLE);
  lf(BASE_SPEED);
}
```

Figure 11: Track Navigation

5.2 Separation of Actions and Sensor Feedback

Motivation Throughout the process of developing this code it became exceedingly clear that flexibility was one of the most important features. We needed a system where adding a sensor and allowing its feedback to control the robot was as simple as writing a new method or two. We also needed it to be possible to add "Actions" to the robot with similar ease. The best way to do this it to separate the control structures completely. Actions need to be influenced by sensor feedback obviously, but the code should be such that if every sensor suddenly blew up the robot could still be commanded to move. Such an implimentaion is outlined in Figure 12



```
void lf(int s)
{
  async_state = LINEF;
  gs = s;
}
```

```
void stop_time(int mils)
{
  async_reset();
  int dt = 0;

  while(dt < mils)
  {
    dt += em;
    async();
  }
}
```

```
void async_reset()
{
  switch(async_state)
  {
    case LINEF:
      pidlf.set_pid(.9, 0, 32);
      w = 0; wsum = 0; adj = 0; pos = CENTEROFLINE;
      break;
    case LFSET:
      pidlf.set_pid(.5,0,200);
      async_state = LINEF;
    case DRIVED:
      pida.set_pid(500, 0, 50);
      dd = 0;motion[0] = 0; adj = 0;dd_flag = 1;
      break;
  }
  em = 0;
}
```

```
void async()
{
  dt = em;
  em = 0;

  update(dt);
  digitalWrite(13, LOW);

  switch(async_state)
  {
    case LINEF:

      pos = read_line();
      if(llf != 8)
      {
        adj = pidlf.slice(CENTEROFLINE - pos, gs, dt);
        mr_out(gs + adj);
        ml_out(gs - adj);
      }
      else
      {
        digitalWrite(13, HIGH);
        mr_out(-last_line*160);
        ml_out(last_line*160);
      }

      break;
```

Figure 12: Independent System Outline

6 3D Printing Components, and Assembling

Overview With TYESA approaching we have sped up our progress on the mechanical side. The robot has had the addition of folding ramps on either side. Our chassis has been printed with all mounting holes ”pre-drilled” and only the removal of support material is necessary. We have also perfected our snap together design. The entire robot comes together with nothing more than 3 Lego cross pieces and two hex screws.

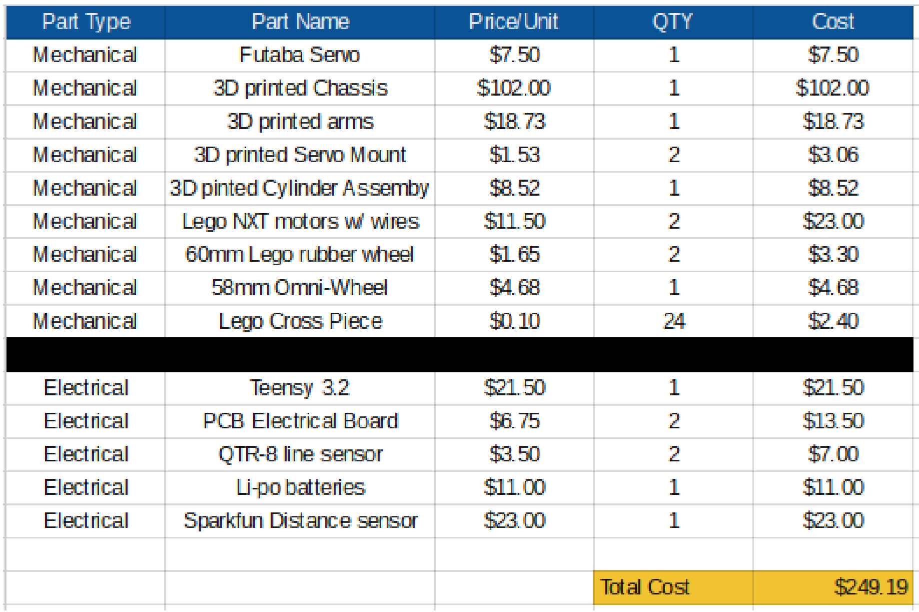
6.1 Manufacture Timing

Different combinations of parts printing together have been tested and an optimal solution has been found. The part groupings are shown below.

1. Top and bottom chassis plates - 10 hours
2. Cylinder Assembly, Servo Mounts - 5 hours
3. Ramps and Splooshner - 5 hours

In the end it takes a mere twenty hours to create our robot, start to finish, mechanically.

6.2 Bill of Materials



Part Type	Part Name	Price/Unit	QTY	Cost
Mechanical	Futaba Servo	$7.50	1	$7.50
Mechanical	3D printed Chassis	$102.00	1	$102.00
Mechanical	3D printed arms	$18.73	1	$18.73
Mechanical	3D printed Servo Mount	$1.53	2	$3.06
Mechanical	3D pinted Cylinder Assemby	$8.52	1	$8.52
Mechanical	Lego NXT motors w/ wires	$11.50	2	$23.00
Mechanical	60mm Lego rubber wheel	$1.65	2	$3.30
Mechanical	58mm Omni-Wheel	$4.68	1	$4.68
Mechanical	Lego Cross Piece	$0.10	24	$2.40
Electrical	Teensy 3.2	$21.50	1	$21.50
Electrical	PCB Electrical Board	$6.75	2	$13.50
Electrical	QTR-8 line sensor	$3.50	2	$7.00
Electrical	Li-po batteries	$11.00	1	$11.00
Electrical	Sparkfun Distance sensor	$23.00	1	$23.00
			Total Cost	$249.19

Figure 13: Cost analysis PER robot

6.3 Figures

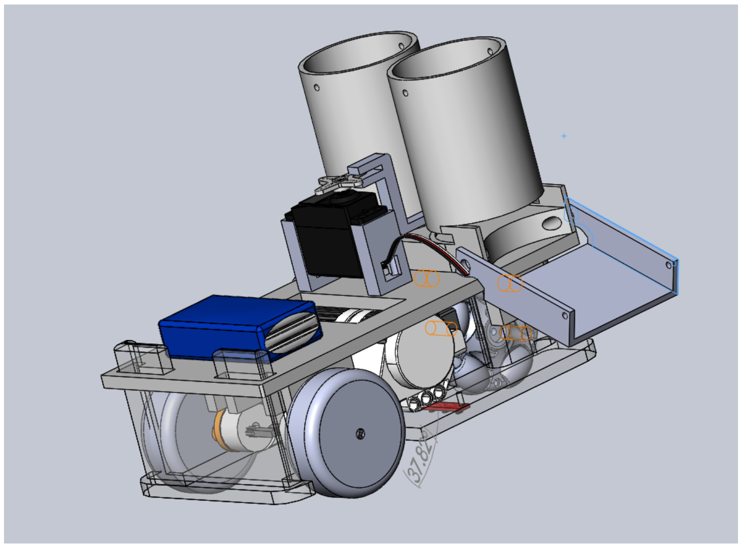


Figure 14: Full Robot in CAD

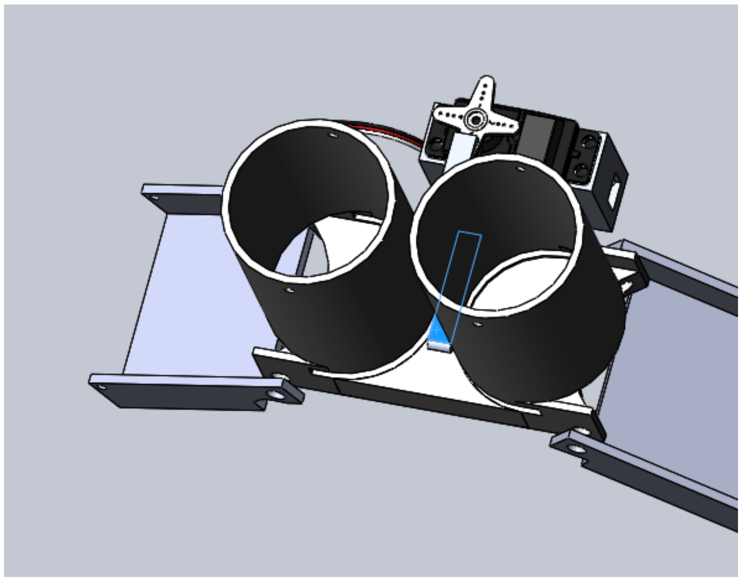


Figure 15: Full Cylinder Deposit System

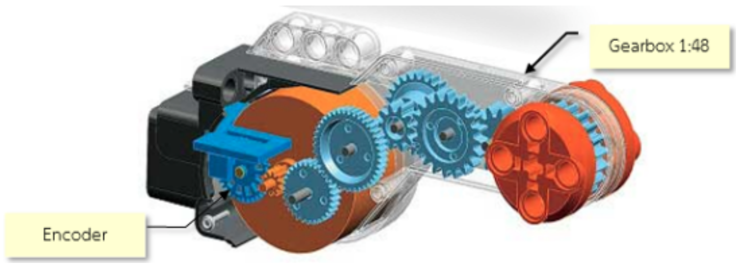


Figure 16: Exploded view of Lego Motor

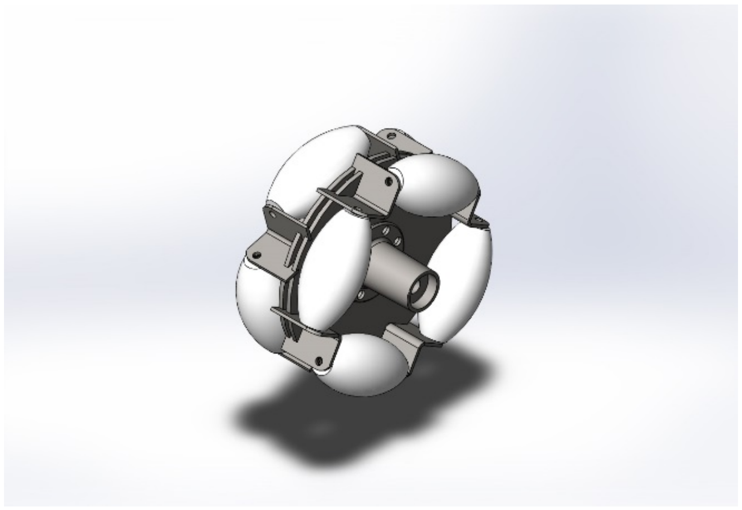


Figure 17: Front Omni-Wheel

7 Electrical Specifications

Overview During out discussion on the electrical system we have decided not to create a conventional wiring diagram. Instead we will use excel to create a pin map which will show how the teensy 3.2 (18) is connected to the rest of the components. The reason for this is we aren't sure whether our final board will end up on a solder proto-board or an engraved board. This type of specification gives us flexibility in final design. As you can see in Figure 19 Each set of pins is color coded to their connection with their connection to the teensy. The pins on the actual device being interfaced with are also listed as to create a full pin map.

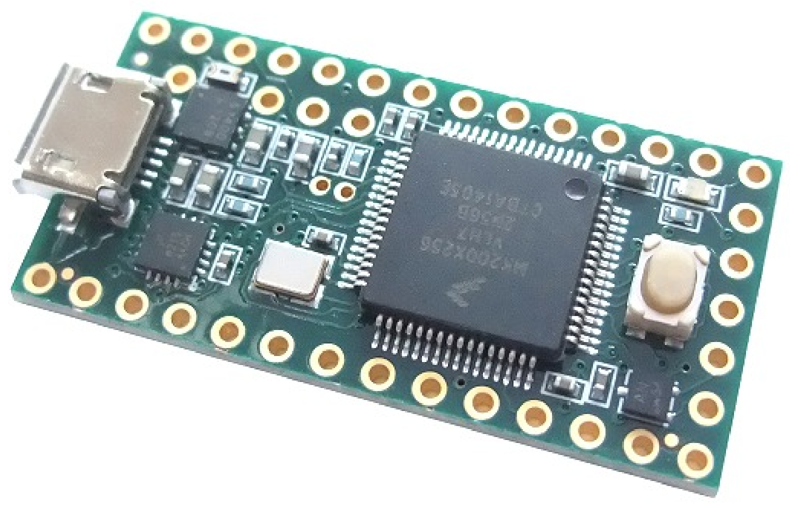
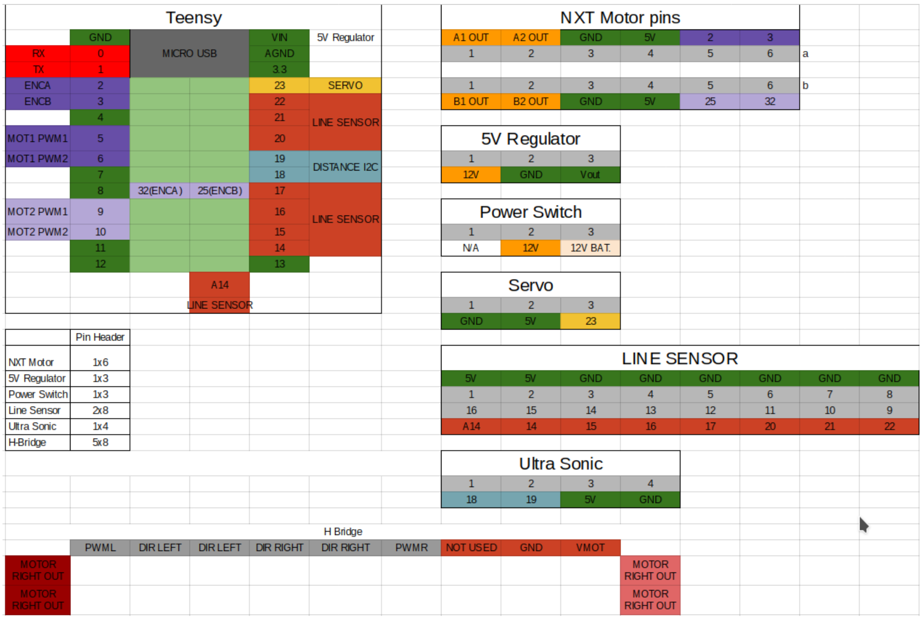


Figure 18: Teensy 3.2 Microcontroller



Teensy

	GND	MICRO USB		VIN	5V Regulator
RX	0			AGND	
TX	1			3.3	
ENCA	2			23	SERVO
ENCB	3			22	LINE SENSOR
	4			21	
MOT1 PWM1	5			20	
MOT1 PWM2	6			19	DISTANCE I2C
	7			18	
	8	32(ENCA)	25(ENCB)	17	LINE SENSOR
MOT2 PWM1	9			16	
MOT2 PWM2	10			15	
	11			14	
	12			13	
			A14		
			LINE SENSOR		

	Pin Header
NXT Motor	1x6
5V Regulator	1x3
Power Switch	1x3
Line Sensor	2x8
Ultra Sonic	1x4
H-Bridge	5x8

NXT Motor pins

A1 OUT	A2 OUT	GND	5V	2	3	
1	2	3	4	5	6	a
1	2	3	4	5	6	b
B1 OUT	B2 OUT	GND	5V	25	32	

5V Regulator

1	2	3
12V	GND	Vout

Power Switch

1	2	3
N/A	12V	12V BAT.

Servo

1	2	3
GND	5V	23

LINE SENSOR

5V	5V	GND	GND	GND	GND	GND	GND
1	2	3	4	5	6	7	8
16	15	14	13	12	11	10	9
A14	14	15	16	17	20	21	22

Ultra Sonic

1	2	3	4
18	19	5V	GND

H Bridge

	PWML	DIR LEFT	DIR LEFT	DIR RIGHT	DIR RIGHT	PWMR	NOT USED	GND	VMOT	
MOTOR RIGHT OUT										MOTOR RIGHT OUT
MOTOR RIGHT OUT										MOTOR RIGHT OUT

Figure 19: Pin Routing Specification Sheet

SIGNATURE: *Jacob M. Kiggins* DATE: 4/25/2016

8 Conclusion

Throughout this design and development process we have all learned a lot. There are many small details in developing a robot such as this. Attention to those details can make the difference between a winning robot and a losing one. We have also gained better insight into the transition between a theoretical model and a working system. As well as which Theoretical models will actually work in the real world. This as well as experience gained from the TYESA competition has prepared us to tackle ASEE.
Also, we won TYESA

8.1 Action Shot

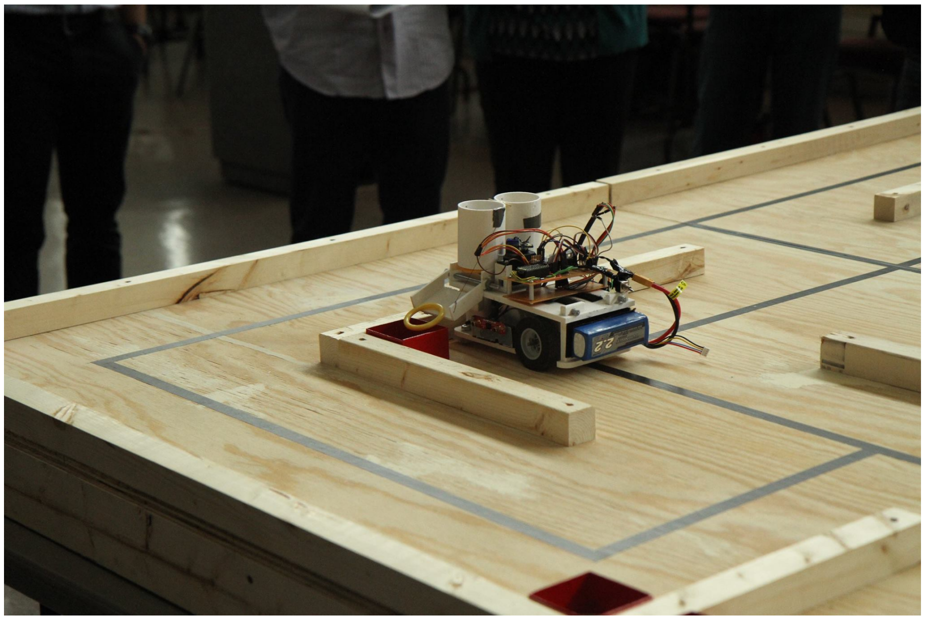


Figure 20: Robot Depositing Ring