EE 471W Final Report

VMI ECE 2018 Senior Capstone

ECE Senior Design Team

EE-471-W

Date Submitted: 04 MAY 2018

Introduction **13**

Design Methodology **14**
Functional Decomposition 15
Preliminary Design Reviews 17

References / In-Depth Documentation / Additional Information **37**
Critical Design Reviews 40

System Overview **40**
Purpose: 40
Checks: 40
Functional Decomposition 40
Requirements 43
Functionality Test 44
Main Diagram/Schematics 44

1.1 Treasure Chest Mechanism **44**

1.2 Chassis **47**

1.4 IR Receiver **51**

Dependencies [Graphic] **52**

Final System Documentation **64**
Purpose: 78
Functional Decomposition 78
Requirements 80
Functionality Test 81
Main Diagram/Schematics 81

1.1 Treasure Chest Raise Mechanism **81**

1.2 Chassis **82**

Functionality Test **82**

1.4 IR Receiver **84**

1.6 Stage A/C Button Press Mechanism **85**

1.10 Flag Raise Mechanism **85**

1.11 Power Supply **86**

Functional Decomposition **86**

1.12 and 1.13 Navigation and Main Function 88

Schedule Management Plan 93
Schedule Model Development 93
Level of Accuracy 93
Units of Measure 93
Project Schedule Model Maintenance 93
Control Thresholds 93
Reporting Formats 93

Cost Management Plan 94
Cost Planning 94
Structure and Accounts 94
Controlling Cost 94
Project Accounts 96

Purchasing Plan 97
Purchasing Metrics 98
Globalization Analysis Reports 100

Quality Management Plan 103
Quality Definition 104
Quality Management 104
Quality Assurance 104
Quality Control 104

Human Resource Management Plan 105
Roles and Responsibilities 105
Leadership 105
Teams 105
External Resources 105
Project Organization Chart 105
Staffing Management Plan 106
Individual Development Plan Template 106
Professional Goals / Motivations 106
Top Three Strengths 106
Top Three Areas of Improvement 106
Development Plan 107

Risk Management Plan 107
Methodology 107
Roles and Responsibilities 107

Budgeting 107
Timing 107
Risk Categories 108
Risk Probability and Impact 108
Reporting Formats 108
Tracking 108
Risk Register

Stakeholder Management Plan 113
Identification 114
Planning 114
Manage Engagement 114
Control Engagement 116
Alternate Stakeholder Analysis 116

Product Improvement Plan 119

Introduction

In order to fulfill the capstone requirement for the VMI ECE B.S. program as outlined in the Systems Design I (EE 321X) and Systems Design Validation (EE 471 W) courses[1], ECE students projected to graduate on May, 2018 will participate in the IEEE Southeastcon 2018 Hardware Competition[2]. To achieve this objective, the VMI 2018 Southeastcon Team composed of these students will work under the guidance of COL Thomas J. McCormick (the ECE Professor for EE 321X and EE 471 W), the general oversight of COL Addington (VMI ECE Department Head), and all other VMI ECE Department faculty and staff.

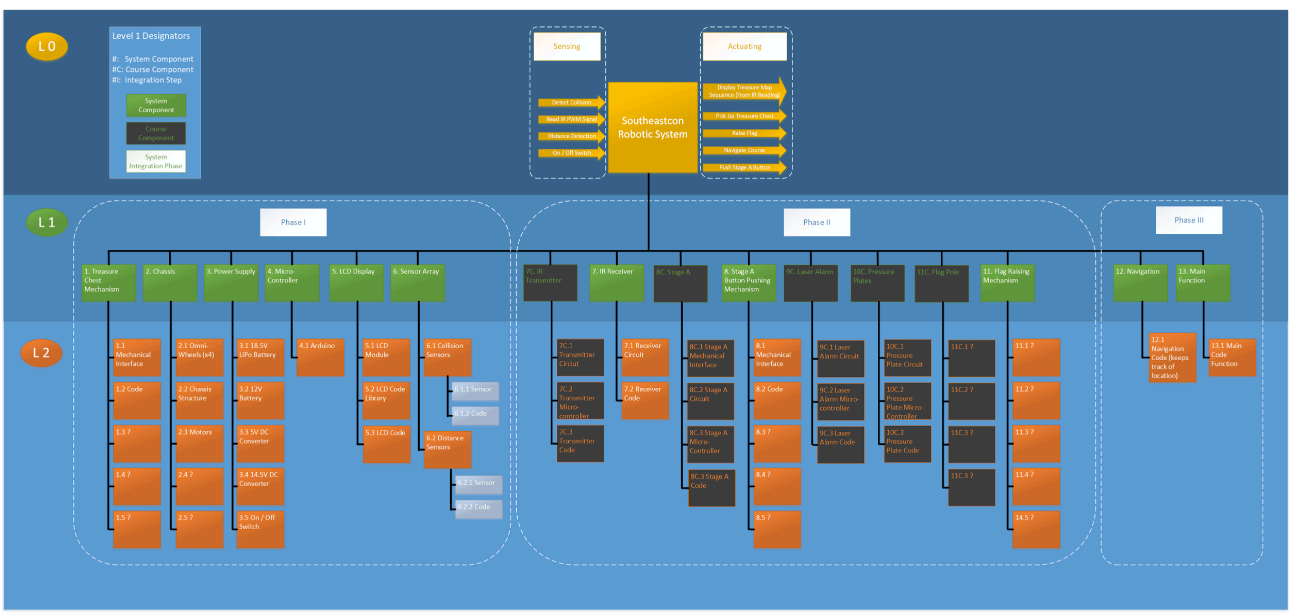
[1] http://catalog.vmi.edu/preview_entity.php?catoid=22&ent_oid=494&returnto=676
[2] http://www.ieee.org/conferences_events/conferences/conferencedetails/index.html?Conf_ID=40040

The 2018 VMI Southeastcon Team will design, construct, and test an autonomous robotic system. The team will compete in the 2018 Southeastcon Hardware Competition. A working prototype will be completed by the end of the 2017 Fall Semester. This prototype must be built within the constraints set forth by the competition coordinators as defined on the official Southeastcon 2018 Slack page and designed to achieve a maximum number of points during the competition.

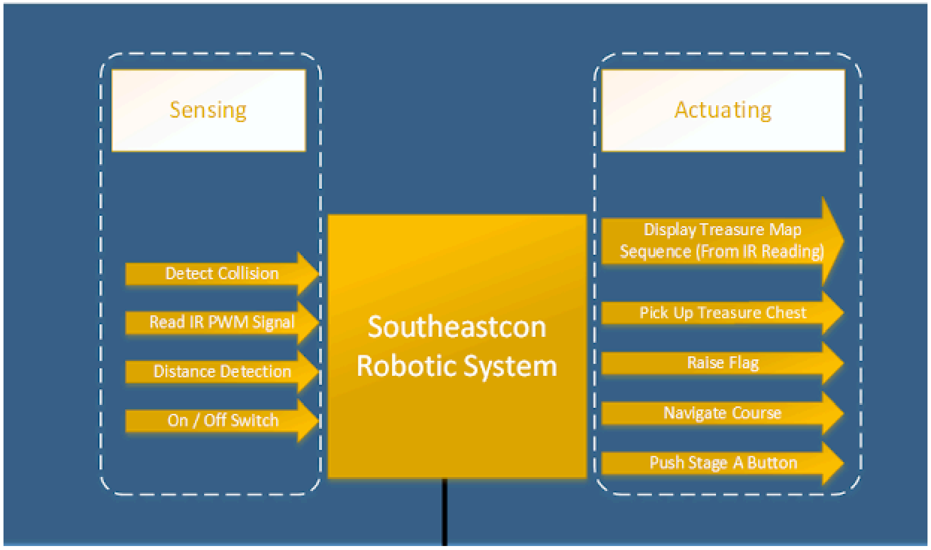
Design Methodology

The Capstone design was created through a series of preliminary design reviews (PDRs) and critical design reviews (CDRs) for each subsystem and a final CDR for the entire system. This section will address the design reviews and consequent designs which were developed by the capstone team.

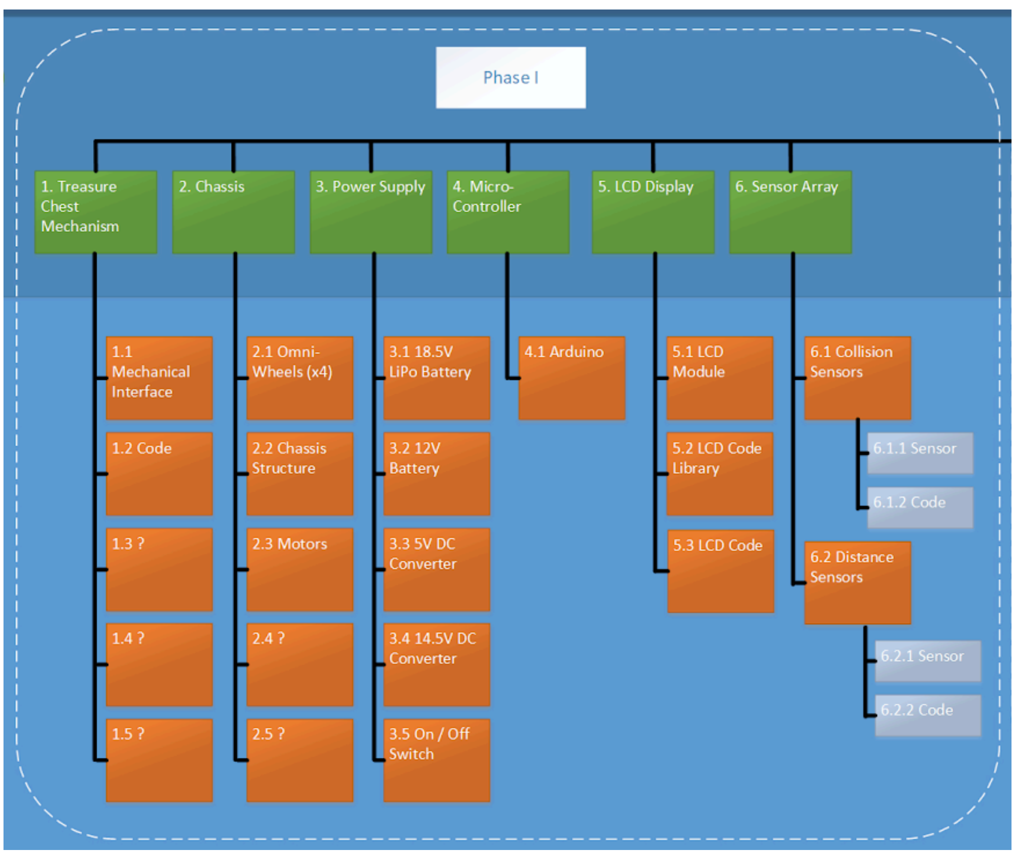
Functional Decomposition



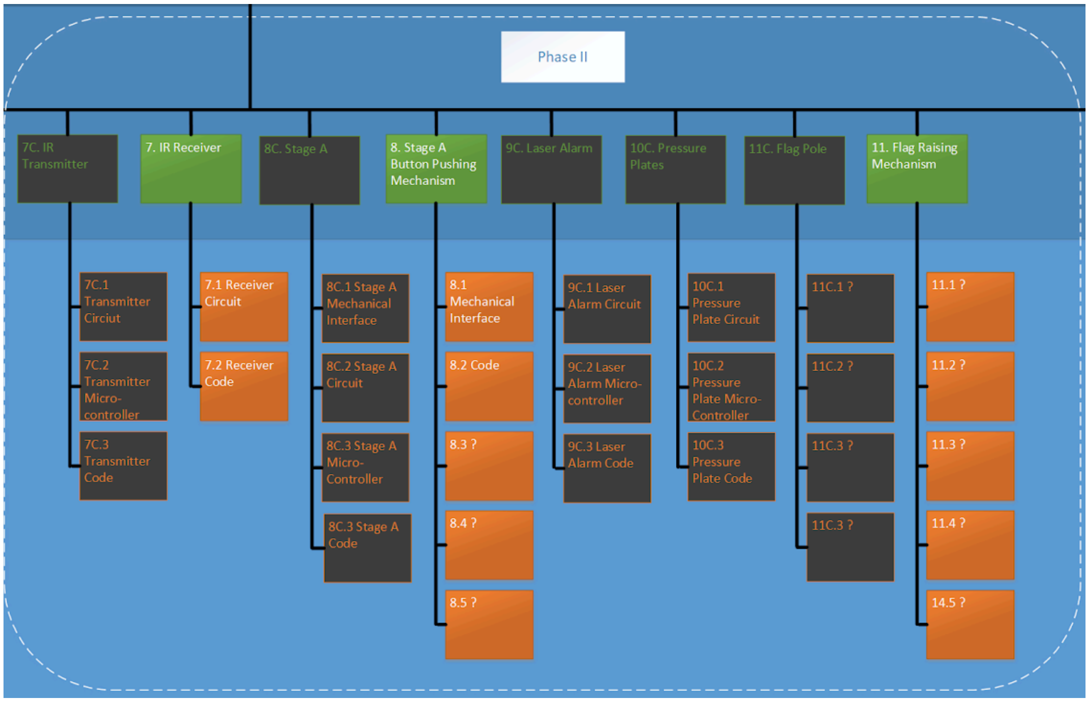
Level 0



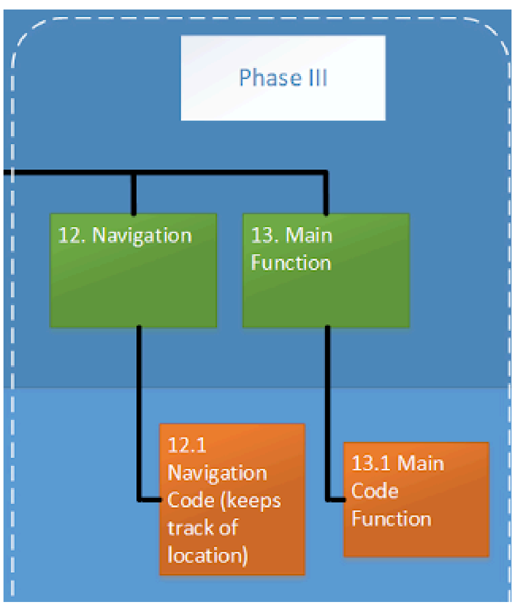
Phase I



Phase II



Phase III



Preliminary Design Reviews

Treasure Chest Mechanism PDR

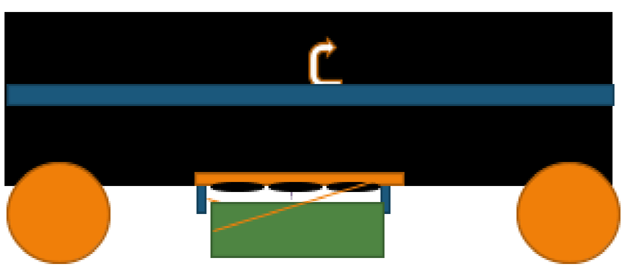
Checks

[X] Tech Team

[X] Software

[X] Hardware

Functional Decomposition [Diagram]



Operational Description

The robot will first drive over the treasure chest so the mechanism is centered on the chest. The AX-12 (orange square) with rotate, spooling out the wire (purple line) and lowering the platform (thin orange rectangle) with the magnets (black ovals) until the magnets attach securely to the chest. Then, the motor will rotate in the opposite direction, lifting the chest/platform off the ground to a height that will safely clear the ramp. The guide rods (thin pair of vertical blue rectangles) are attached through the platform and will ensure the platform does not sway back and forth. The robot will then resume normal operation.

Dependencies

Omni-Wheels

Material List

AX-12 Motor

Polymer platform

Plastic guide rods

Wire for spooling/lowering and raising platform

Neodymium magnets

Axle/axle hub

Functionality Test

This design must properly lower until secured to chest, then raise the chest/platform back up to a height that will allow the robot to clear the ramp.

Chassis PDR

Checks

[x] Tech Team
[x] Software
[x] Hardware

Functional Decomposition [Diagram]

The Chassis from the tech team standpoint has a few aspects that are the most important. Also there are some aspects that can be assumed as givens.

Level 0: Chassis Design

Level 1: a) Wheel cut outs
b) Platform for lifting box
c) Total dimensions of Robot
d) Power Consumption/ power allocation
e) Sensor arrays used
f) Micro-Controller used

Operational Description

The electronics manipulate the hardware to complete the tasks we need, such as navigating the course, and basically everything else. They are also supposed to be efficient and effective at completing the task.

Dependencies [Graphic]

1. Power Consumption (tech)
2. Size of chassis (hard)
3. Treasure Chest Mechanism (hard)
4. Budget

Material List

1. Teensy 3.6
2. Breadboards
3. 10K ohm
4. 3 Voltage dividers
5. 4 bump sensors
6. 2-4 TOF sensors
7. ~2 IR sensors
8. Voltage Regulator
9. 5V Breadboard bridge
10. Male/Female plugs for the LiPo

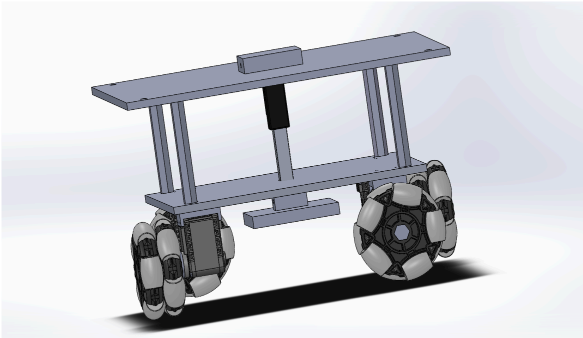
Functionality Test

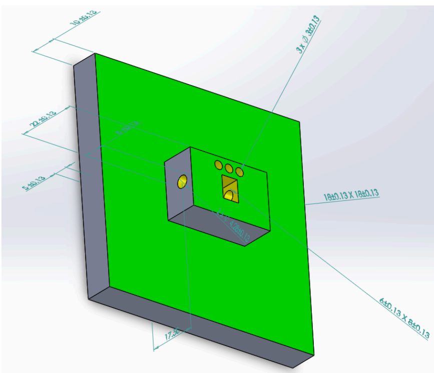
Every Component has been tested individually and functions 100% properly. Except for IR but we do not have those in yet and they have been modeled with regular photodiodes.

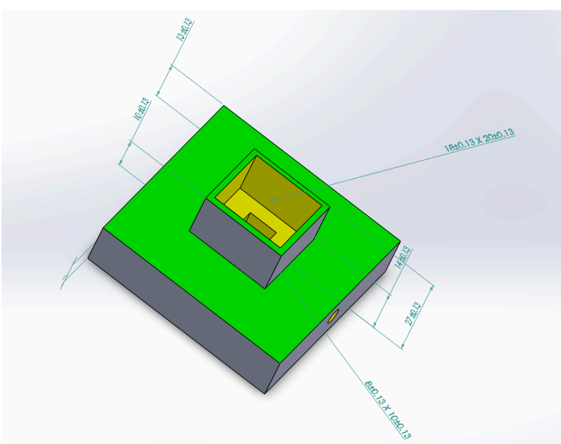
- TOF sensors Work
- Bumps Work
- Dynamixel works.
- Laser trip Wire works
- LED (acting as IR Tx) works with interrupts in a functional loop
- LCD Screen HAS NOT come in yet so we still need to test and functions check it

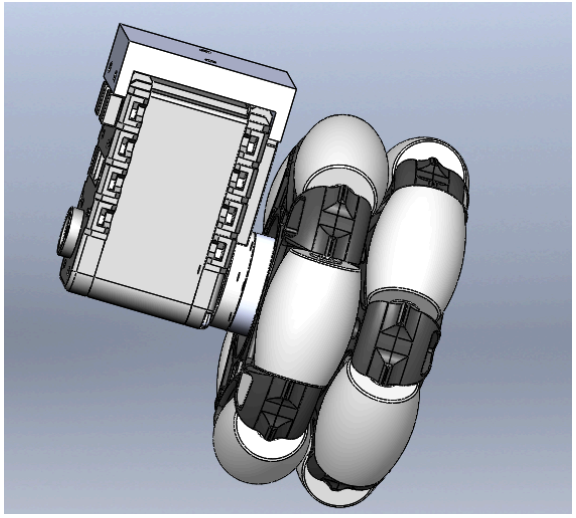
For the CDR ***It will be assumed that everything works together*** This will be tested.

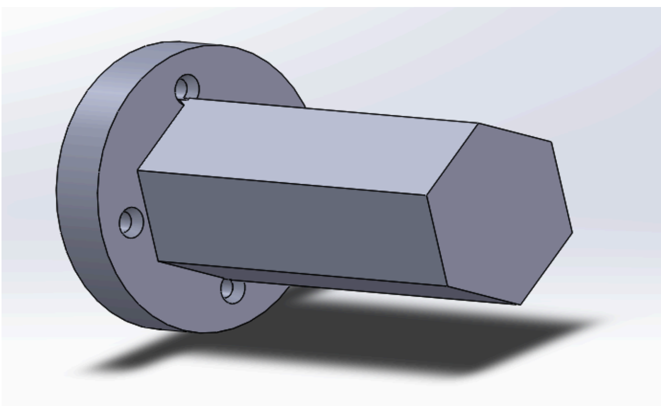
References / In-Depth Documentation / Additional Information




10±0.13
3x ⌀ 3±0.13
22±0.13
5±0.13
18±0.13 X 18±0.13
2x ⌀ 4.28±0.13
6±0.13 X 8±0.13
17.50


13±0.13
10±0.13
18±0.13 X 20±0.13
14±0.13
27±0.13
8±0.13 X 10±0.13





```
//--------------------------------------------------
//--------------------------------------------------
//--------------------------------------------------
//--------------------------------------------------
/////////////TOF-BUMP CODE/////////////
//--------------------------------------------------
//--------------------------------------------------
//--------------------------------------------------
//--------------------------------------------------

#include <arduino.h>
#include "Adafruit_VL53L0X.h"

int Bump1_pin = A0;
int Bump2_pin = A1;
int LaserLEDRx_pin = A2;
```

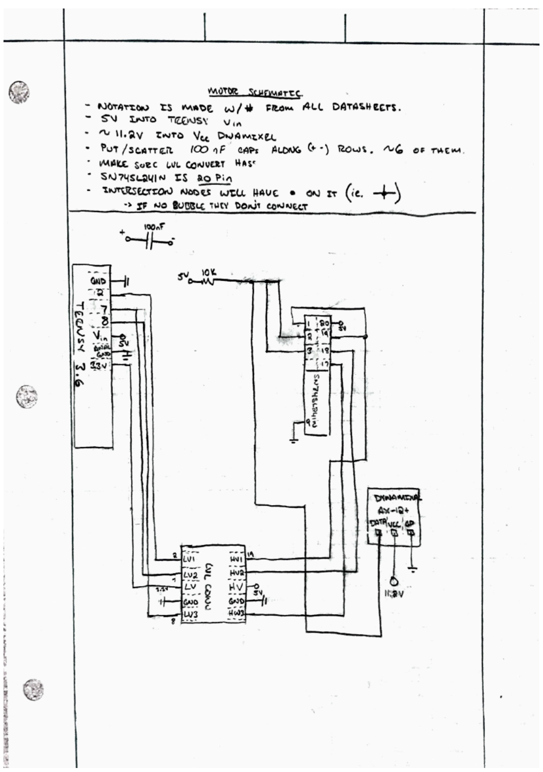
```
int LED_Bump1 = 53;
int LED_Bump2 = 52;
int LED_TOFno = 51;
int LED_TOFshort = 50;
int LED_TOFlong = 49;
int LED_Laser = 48;

Adafruit_VL53L0X lox = Adafruit_VL53L0X();

void setup() {
  // put your setup code here, to run once:
Serial.begin(115200);
//------------------------------------------------------------
//LEDS

pinMode(LED_Bump1, OUTPUT);
pinMode(LED_Bump2, OUTPUT);
pinMode(LED_TOFno, OUTPUT);
pinMode(LED_TOFshort, OUTPUT);
pinMode(LED_TOFlong, OUTPUT);
pinMode(LED_Laser , OUTPUT);

//------------------------------------------------------------

//------------------------------------------------------------
//TOF code
  // wait until serial port opens for native USB devices
  while (! Serial) {
    delay(1);
  }

  Serial.println("Adafruit VL53L0X test");
  if (!lox.begin()) {
    Serial.println(F("Failed to boot VL53L0X"));
    while(1);
  }
  // power
  Serial.println(F("VL53L0X API Simple Ranging example\n\n"));
//------------------------------------------------------------
}

void loop() {
  VL53L0X_RangingMeasurementData_t measure;
```

```
  lox.rangingTest(&measure, false); // pass in 'true' to get debug
data printout!

  if (measure.RangeStatus != 4) {  // phase failures have incorrect
data
     Serial.print(measure.RangeMilliMeter);
  } else {
    Serial.print("-1");
  }

int Bump1_val = analogRead(Bump1_pin);
int Bump2_val = analogRead(Bump2_pin);
int LaserLEDRx_val = analogRead(LaserLEDRx_pin);

Serial.print("   ");
Serial.print(Bump1_val);
Serial.print("   ");
Serial.print(Bump2_val);
Serial.print("   ");
Serial.print(LaserLEDRx_val);

Serial.print("\n");

Bump1_lightup(Bump1_val);
Bump2_lightup(Bump2_val);
int l = TOF_check();
TOF_lightup(l);
Laser_lightup(LaserLEDRx_val);

}
//------------------------------------------------
//FUCNTIONS
//------------------------------------------------
void Bump1_lightup(int x)
{
  digitalWrite(LED_Bump1, LOW);
  if(x > 100)
  {
     digitalWrite(LED_Bump1, HIGH);
  }
}
void Bump2_lightup(int x)
```

```
{
  digitalWrite(LED_Bump2, LOW);
  if(x > 100)
  {
     digitalWrite(LED_Bump2, HIGH);
  }
}
int TOF_check()
{
  VL53L0X_RangingMeasurementData_t measure;

  lox.rangingTest(&measure, false); // pass in 'true' to get debug
data printout!

  if (measure.RangeStatus != 4) {  // phase failures have incorrect
data

     return measure.RangeMilliMeter;
  } else {
    return -1;
  }
}
void TOF_lightup(int x)
{
  digitalWrite(LED_TOFlong, LOW);
  digitalWrite(LED_TOFshort, LOW);
  digitalWrite(LED_TOFno, LOW);

  if (x > 400 && x < 1000)
  {
    digitalWrite(LED_TOFlong, HIGH);
  }
  else if (x < 400 && x > 0)
  {
    digitalWrite(LED_TOFshort, HIGH);
  }
  else if (x == -1)
  {
    digitalWrite(LED_TOFno, HIGH);
  }
}
void Laser_lightup(int x)
{
```

```
  digitalWrite(LED_Laser, LOW);
  if(x > 120)
  {
     digitalWrite(LED_Laser, HIGH);
  }
}
//-------------------------------------------------
//-------------------------------------------------
//-------------------------------------------------
//-------------------------------------------------
/////////////MOTOR CODE + INTERRUPTS////////////
//-------------------------------------------------
//-------------------------------------------------
//-------------------------------------------------
//-------------------------------------------------
```

SCHEMATIC:




```
//-------------------------------------------------
//-------------------------------------------------
//-------------------------------------------------
//-------------------------------------------------
```

```
#include "DynamixelControl.h"
#include "arduino.h"
#include <avr/io.h>
#include <avr/interrupt.h>
DynamixelControl3 DmC;
int Count1 = 0; //used for first interupt
IntervalTimer IRTx;


byte MotorLeft = 13;  //CCW for forward, CW for backwards
byte MotorRight = 4;  //CW for foward, CCW for backwards1
const int Period = 1000000; //value is in microseconds
const int IRTx_pin = 32;

void setup()
{

  IRTx.begin(interruptIRTx, Period); // begin the interrupt
interruptIRTx at Period seconds
  pinMode(13, OUTPUT); //LED output
  digitalWrite(13, HIGH); //LED turn on
  Serial.begin(250000); //Serial begin at 250000
  pinMode(IRTx_pin, OUTPUT); //LED output

  DmC.begin(200000, 2); //begin speaking to the motors at 200k baud
through pin 2
  delay(10);
  motorstop(); //just as precaution have the motors just stop
  delay(10);


  DmC.freeRotation(MotorLeft, true);
  delay(10);
  DmC.freeRotation(MotorRight, true);
  delay(10);
  //next 2 commands sets motors to max speed forward
  DmC.turnVelocity(MotorLeft, CW, 0x03FF);
  delay(10);
  DmC.turnVelocity(MotorRight, CW, 0x03FF);
}
//start loop
void loop()
{
```

```
  motorEndless();

}

//-----------------------------------------------------------------
//FUNCTIONS
//-----------------------------------------------------------------

void interruptIRTx()
{
  digitalWrite(IRTx_pin,HIGH);
  delay(200);
  digitalWrite(IRTx_pin,LOW);
}
void motorEndless() {
  //just set motors forward at max speed
  //this function is better than motor forward because you can
interupt this
  //and have sensors track things and such
  //setting free rotation on
  DmC.freeRotation(MotorLeft, true);
  delay(20);
  DmC.freeRotation(MotorRight, true);
  delay(20);
  //setting motor speed forward to full speed for both motors
  DmC.turnVelocity(MotorLeft, CCW, 0x03FF);
  delay(20);
  DmC.turnVelocity(MotorRight,  CW, 0x03FF);
}
void motorstop() {
  //stops the motors
  DmC.freeRotation(MotorLeft, true);
  delay(10);
  DmC.freeRotation(MotorRight, true);
  delay(10);
  //next 3 lines set motor speed to 0
  DmC.turnVelocity(MotorLeft, CW, 0x0000);
  delay(10);
  DmC.turnVelocity(MotorRight, CCW, 0x0000);
}
```

<u>**IR Transmitter & Receiver PDR**</u>

Checks

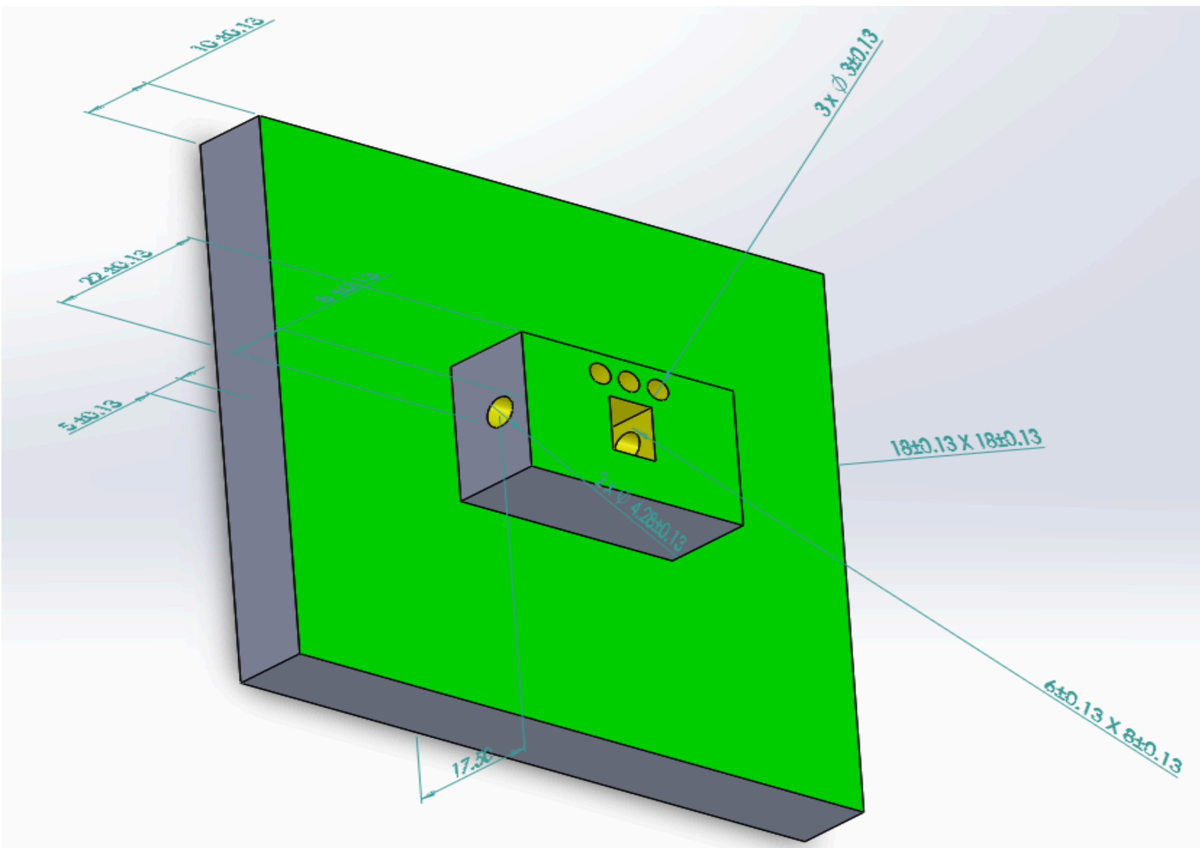
[X] Tech Team

[X] Software

[X] Hardware

Functional Decomposition [Diagram]

Actuator to Platform



Operational Description

IR Receiver: This device will descend, attached to the bottom of the actuator platform, to a maximum range of no more than 2 inches (maximum effective range). From there, it will receive a synch bit followed by a binary 3-bit signal (to be clarified by further instructions) at a specified duty cycle and frequency and communicate with the Teensy to write to the LCD Display. This will continue for one minute, with a specified interval in between signal emission.

IR Emitter: This part is not part of the robot; it is a stage element. It will send out the signal at a desired frequency and duty cycle. This will be adjusted by the referees.

Dependencies [Graphic]

IR Transmitter

Material List

Standard LCD 16x2

IR Receiver Sensor

Teensy 3.6

Functionality Test

This test must prove the IR Transmitter can properly output a three bit code and the IR Receiver must properly read the three bit code. Once the code is read by the IR Receiver it must display the proper code on the LCD screen.

Diagrams/Schematics

IR Receiver

```
void setup() {

//ADJUST THESE

int pinIRin= 13; //pin that the IR is getting information from

int freq= 440; //this is the frequency that the IRxmitter is giving
back to this unit

int sequence[]= {0, 0, 0};

int sync= 1000; //how long after the sync bit the signal is written

 //synchronization bit
```

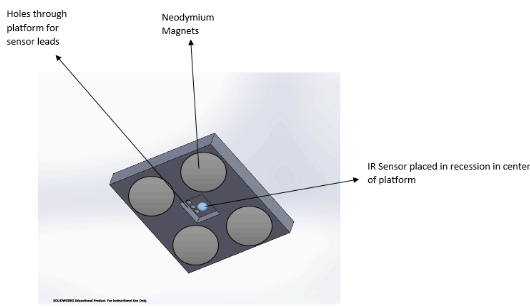
```
 while(digitalRead(pinIRin)== LOW) //do nothing while the the Xmitter
isn't sending)
 {
 ;
 }
 while(digitalRead(pinIRin) == HIGH) //now it's high, still do
nothing
 {
 ;
 }
 delay(sync);
 //store the bits into the 3 bit binary sequence
 for(int i=0; i<3; i++){
  sequence[i]=digitalRead(pinIRin);
  delay(1/freq); //sample every period
  }
}
void loop() {
}
```

IRXmitter

```
  //constants (ADJUST THESE)
  int DC = .5; //duty cycle, given in percentage
  int freq = 440; //frequency of the signal
  int pinIRin = 14; //pin that the IR is being written to
  int sequence[] = {0, 0, 0};
  int sync = 1000; //how long after the sync bit the signal is
written


  //WILL NEED TO USE TIMER INTERRUPTS
```

```
  int interval= 5; //how often the pattern repeats in seconds
  //timer for the 1 minute

  //these things help with the duty cycle(DO NOT TOUCH)
  float period = 1 / freq;
  int hightide = period * DC;
  int lowtide = period * (1 - DC);
void setup() {
}
void loop() {
  //binary code
  for(int i=0; i<3; i++)
   {
    syncbit();
    dutycycles(i);
   }
}
void dutycycles(int i){
    if(sequence[i]==1){
    digitalWrite(pinIRin, HIGH);
    delay(hightide);
    digitalWrite(pinIRin, LOW);
    delay(lowtide);
    }
    else{
    digitalWrite(pinIRin, LOW);
    delay(hightide);
    digitalWrite(pinIRin, HIGH);
```

```
    delay(lowtide);
    }
}
void syncbit(){
  digitalWrite(pinIRin, HIGH);
  delay(10000); //wait 10 seconds)
  digitalWrite(pinIRin, LOW);
  delay(sync); //now the two should be synched
}
//SECTION FOR THE TIMERS
//  noInterrupts();           // disable all interrupts
//  TCCR1A = 0;
//  TCCR1B = 0;
//  TCNT1  = 0;
//
//  OCR1A = 31250;            // compare match register 16MHz/256/2Hz
//  TCCR1B |= (1 << WGM12);   // CTC mode
//  TCCR1B |= (1 << CS12);    // 256 prescaler
//  TIMSK1 |= (1 << OCIE1A);  // enable timer compare interrupt
//  interrupts();             // enable all interrupts
//}
//
//ISR(TIMER1_COMPA_vect)          // timer compare interrupt service
routine
//{
//  digitalWrite(ledPin, digitalRead(ledPin) ^ 1);   // toggle LED
pin
//}
```



Flag Raise Mechanism PDR

Checks

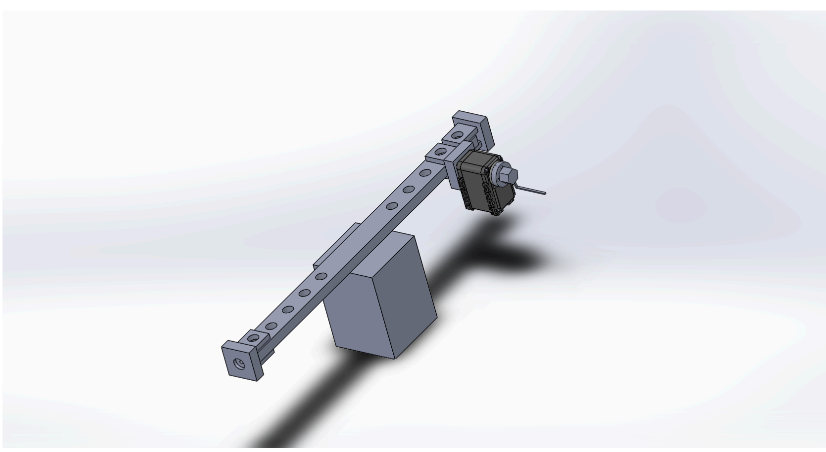
[X] Tech Team

[X] Software

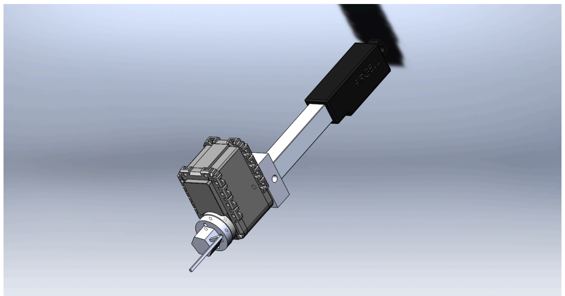
[X] Hardware

Functional Decomposition

Horizontal Actuator



Vertical Actuator



Operational Description

To raise the flag for either design option, we will first need to extend the actuator until it is outside the chassis. Then, the robot will drive forward until the rotating arm is centered in between the 'Captain's Wheel'. Once in place, the AX-12 will begin rotating until the flag is raised. If the 'Captain's Wheel' is different from the image above, the arm will be re-designed so that a proper coupling will be made with the wheel.

Dependencies [Graphic]

Chassis

AX-12

Material List

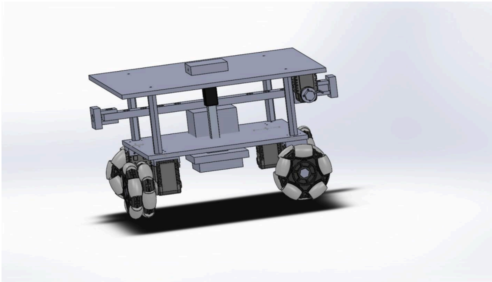
- AX-12
- L-16 Actuator or RobotShop Actuator
- 3D printed AX-12 Mount
- 3D printed arm to rotate wheel

Functionality Test

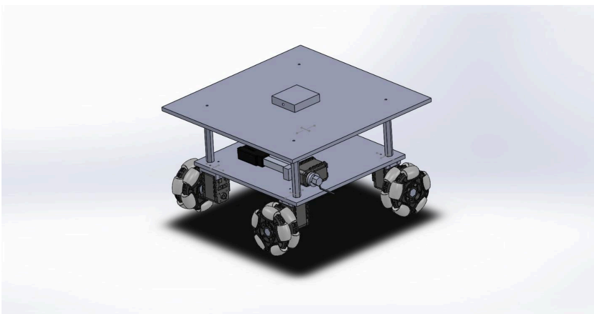
The mechanism will be tested so that it is at an appropriate height off the ground to interface with the Captain's Wheel and will spin the wheel reliably.

References / In-Depth Documentation / Additional Information

Horizontal Actuator

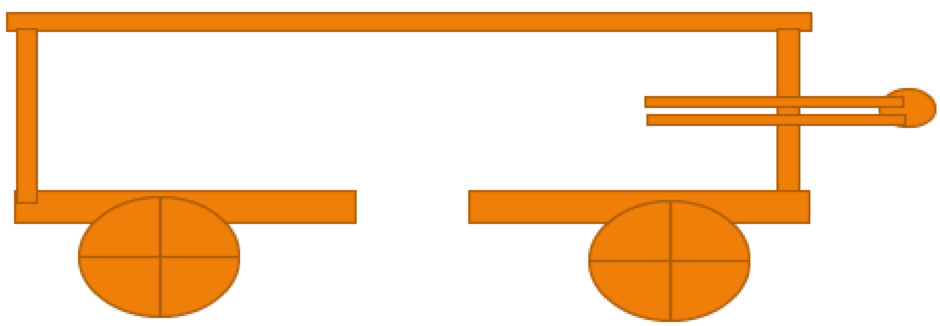


Vertical Actuator



Stage A Push Button PDR

Functional Decomposition [Diagram], Basic Model



Operational Description

Gear rack kit w/785 servo is mounted to extrude to the front or rear (to be later determined), and push button after machine receives the appropriate binary sequence and maneuvers to the desired location using sensors.

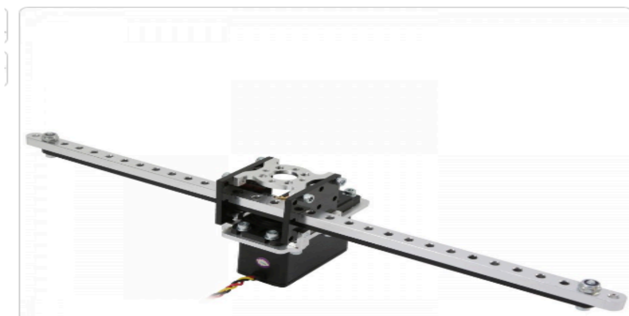
However avoiding motors would be better

Cutting arrays of rods and mounting them to either the front or rear and then ramming into the wall to press the button. Another being to mount flexible rods, and then have the robot come close in a set approximate using the sensors, then move horizontal to the wall in a sweeping motion. Additional information being, these arrays would be laid to form a bushel which incorporates rods extruding at various angles.

Dependencies

Primary: Push button

- Gear Rack Kit w/785 Servo



- Sensors, i.e time of flight, and light sensor
- Height of button
- Recession depth of button
- Force required to trigger button

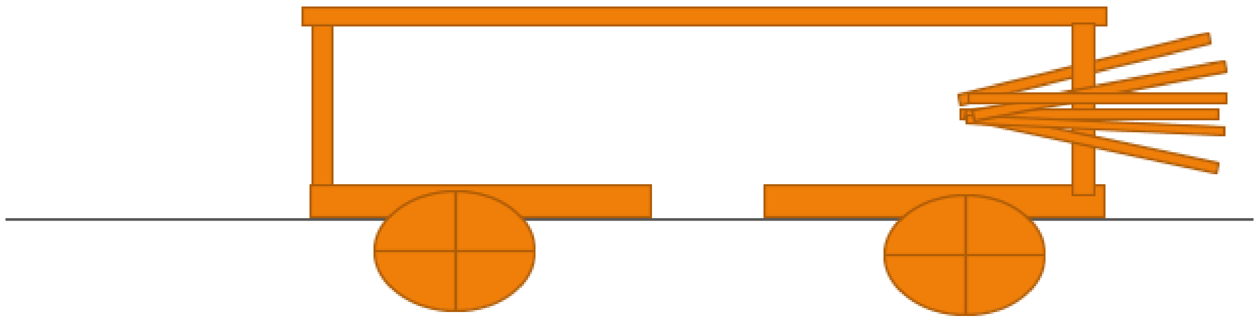
Material List

- Springs
- Silicon Brush
- Gear Rack Kit w/785 Servo
- Rods, i.e metal or wooden
- Foam balls, and strips as seen with noodles.

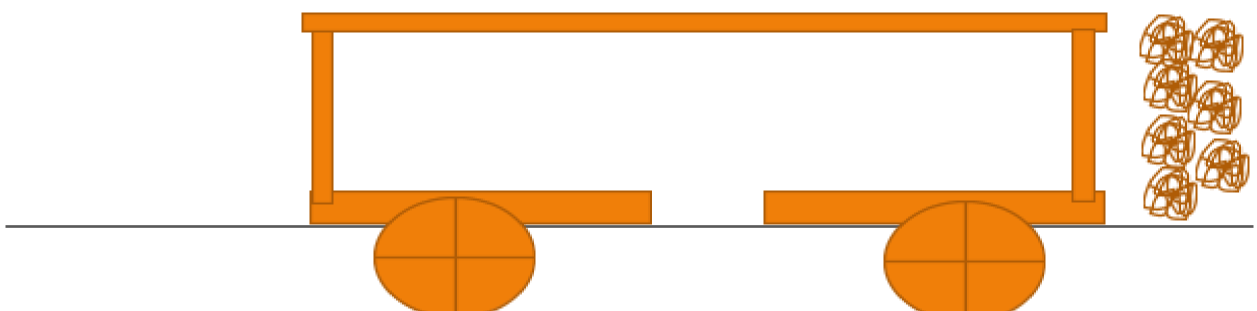
Functionality Test

We will test the functionality of the mechanism by conducting a series of tests with different buttons and depth of recessions in the Stage A wood planks. If the mechanism can depress the button from an inch error from the center of the button in any direction, the design will be successful.

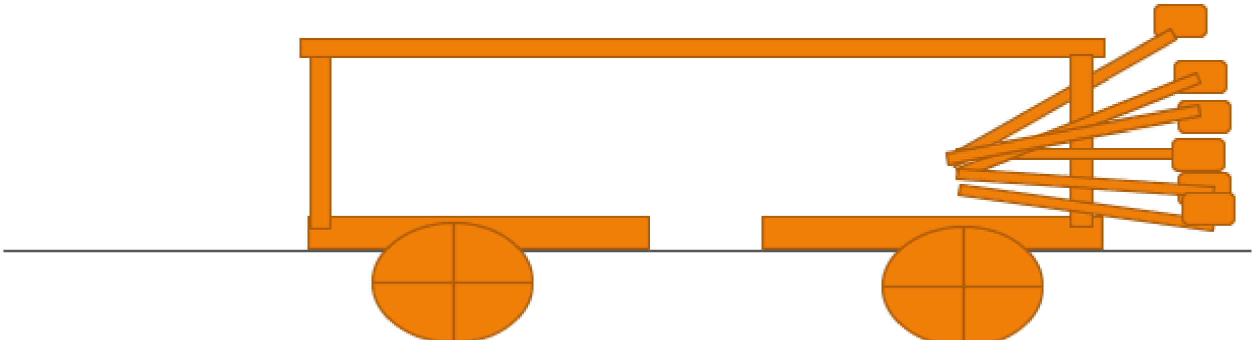
SILICON BRUSH-Designed to go in a sweeping motion across the wall in hopes that a brush falls into the button recession.



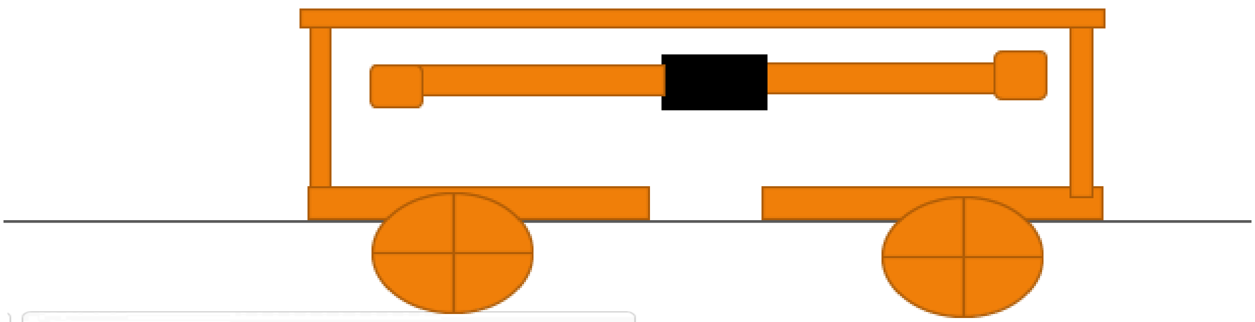
SPRING PAD- Presses into the wall. One spring will fall into the chassis as the others compresses unto wall. This also helps alleviate potential force upon the mechanism from driving into the wall.

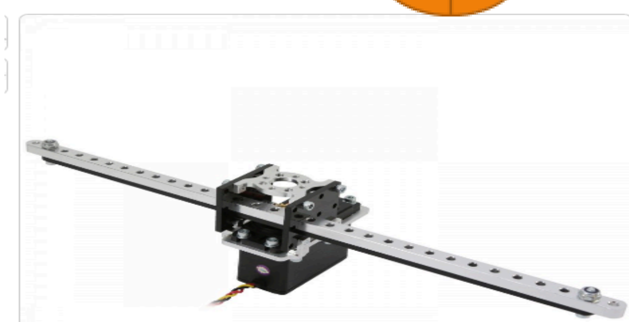


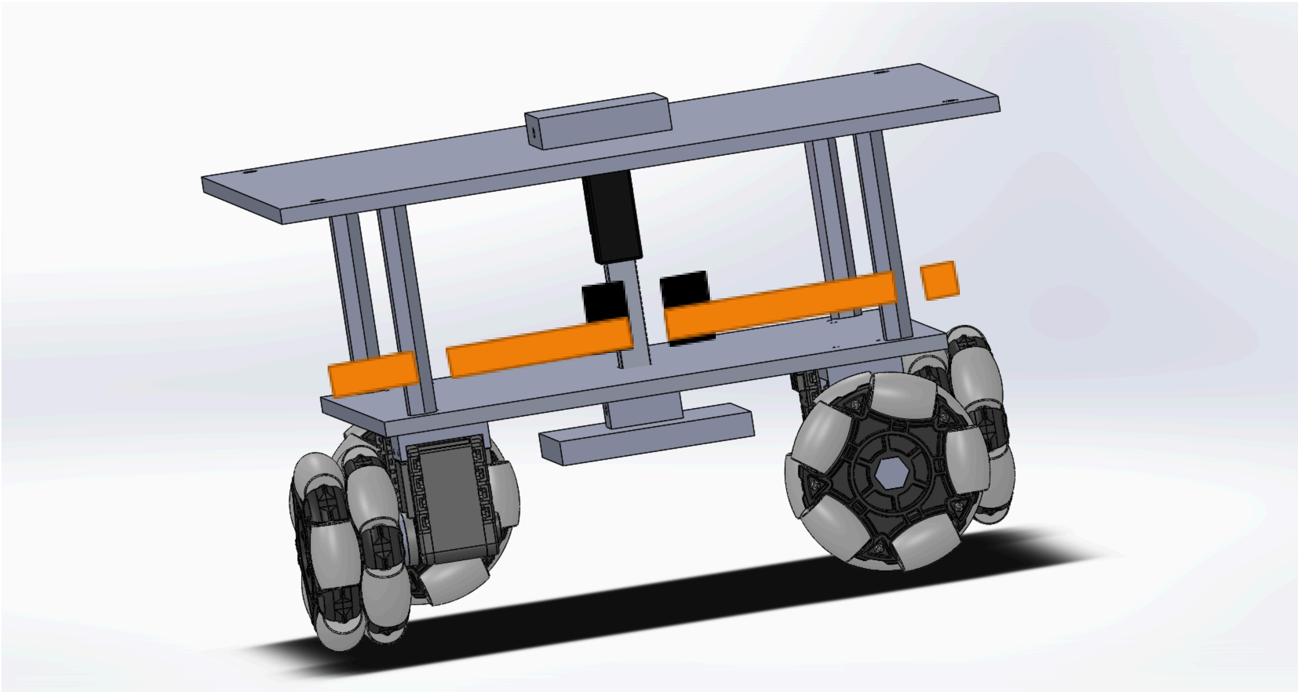
Steel/wooden Rods with foam tips/rods- This is a hybrid of the spring, and basic singular prod design. The foam covers a wider area, but



Gear Rack Kit w/785 Servo- Designed to extrude from within the chassis, and using the time of flight sensors to position the system in front of the button. The end of the system can be interchanged for foam and spring.







Critical Design Reviews

The approved CDRs were compiled and added to a system CDR. This document defines how the system will be integrated into a final product.

SYSTEM CDR

2018 ECE Capstone

System Overview

1. Purpose:

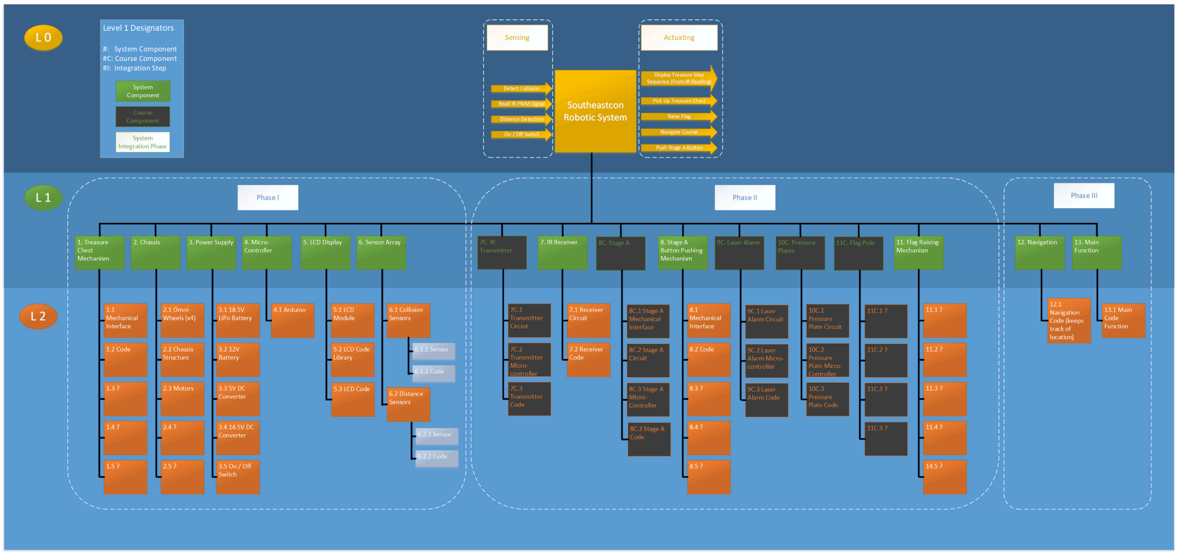
The main purpose of this system CDR is to have a comprehensive document that will allow our team to construct/repair/replace any and all components in a timely manner. In addition, our plan is to have multiple working robots in case of one breaking and this document will allow us to make an exact replica of the first. It will also be used as a template for our final presentation for in addition to our final product we can look at this document and have all of the calculations, specifications, and schematics that went into designing and building the robot.

2. Checks:

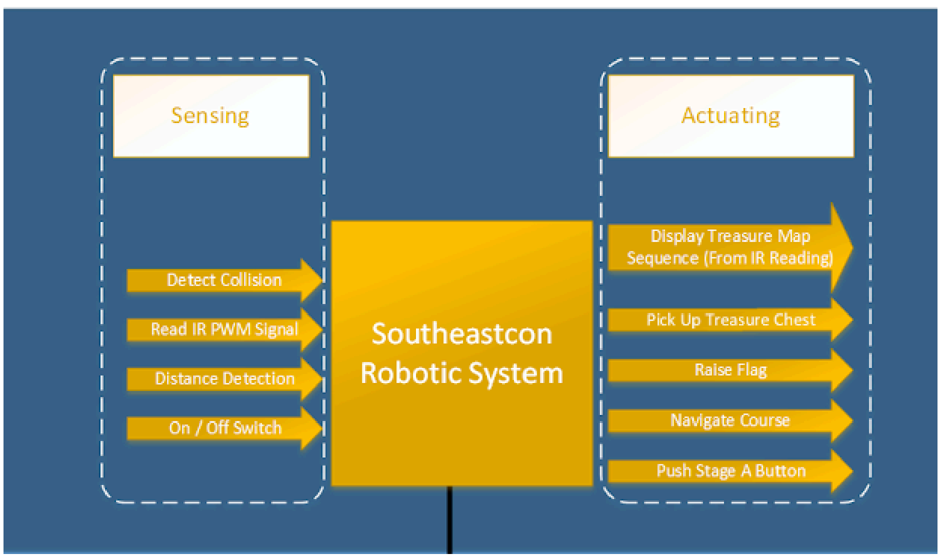
[ ] Tech
[ ] Software
[ ] Hardware

3. Functional Decomposition

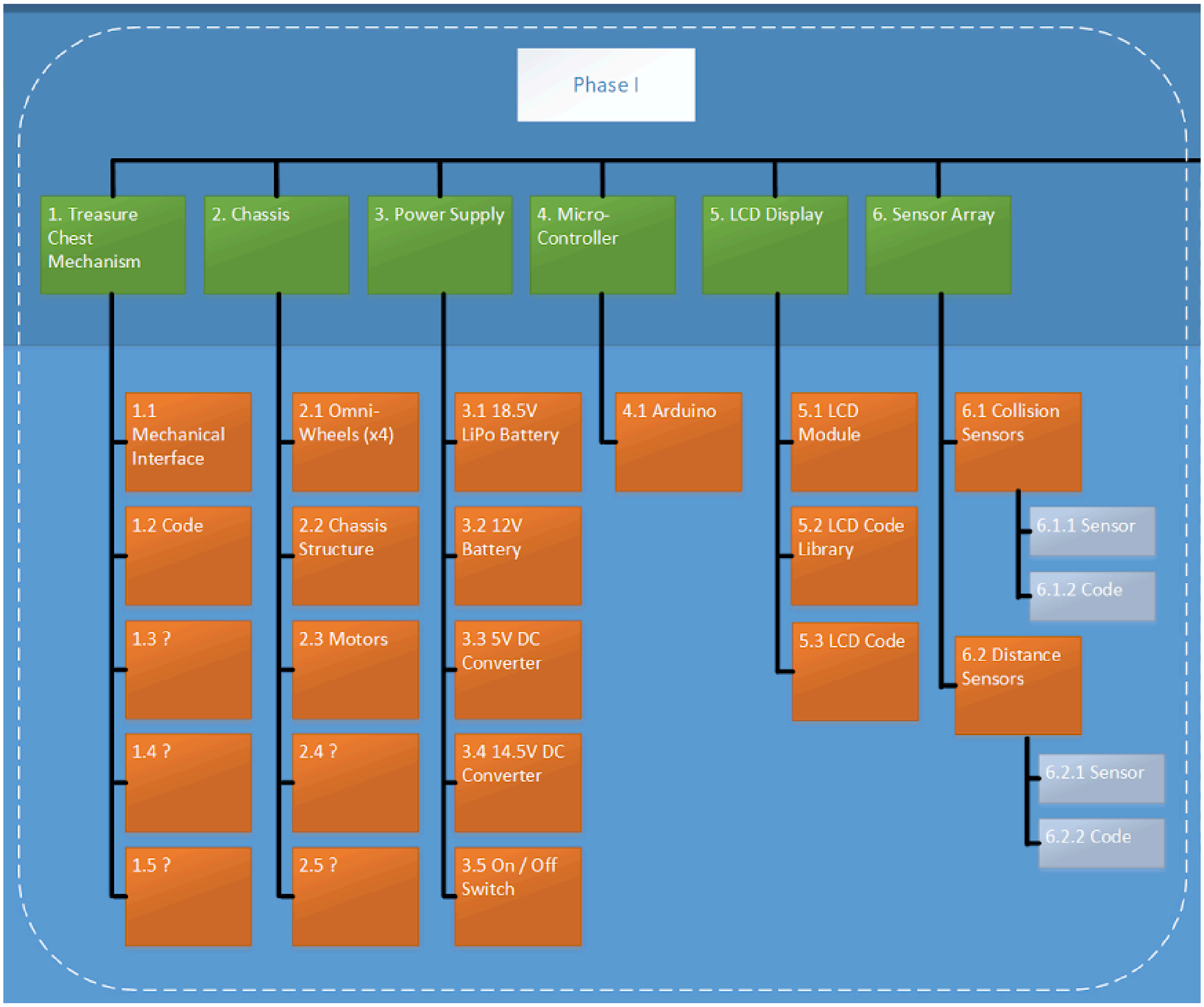
Functional Decomposition



Level 0

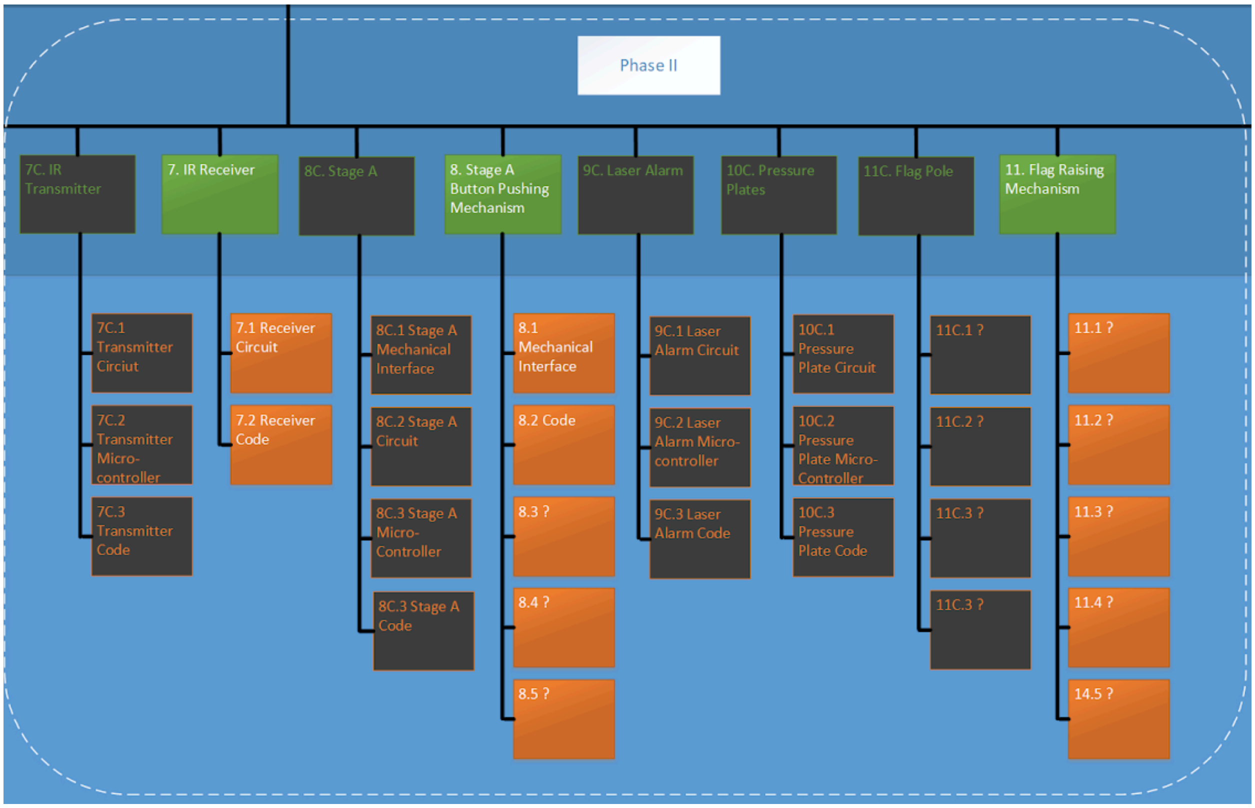


Phase I

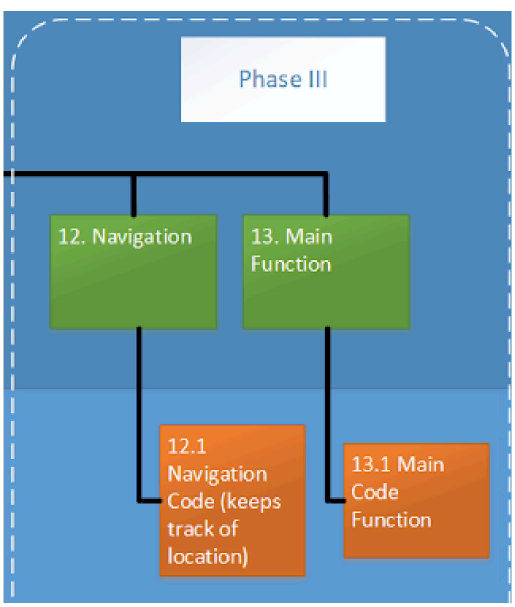


Phase I
1. Treasure Chest Mechanism
2. Chassis
3. Power Supply
4. Micro-Controller
5. LCD Display
6. Sensor Array
1.1 Mechanical Interface
1.2 Code
1.3 ?
1.4 ?
1.5 ?
2.1 Omni-Wheels (x4)
2.2 Chassis Structure
2.3 Motors
2.4 ?
2.5 ?
3.1 18.5V LiPo Battery
3.2 12V Battery
3.3 5V DC Converter
3.4 14.5V DC Converter
3.5 On / Off Switch
4.1 Arduino
5.1 LCD Module
5.2 LCD Code Library
5.3 LCD Code
6.1 Collision Sensors
6.1.1 Sensor
6.1.2 Code
6.2 Distance Sensors
6.2.1 Sensor
6.2.2 Code

Phase II



Phase III



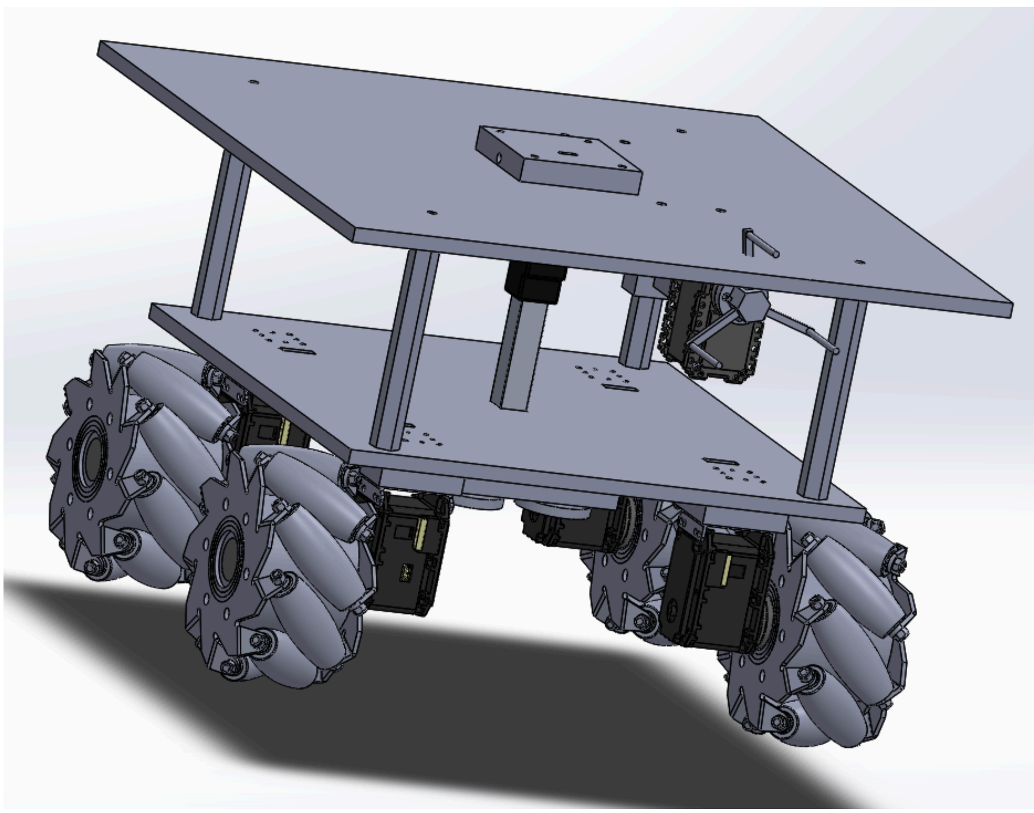
4. Requirements

The system must meet all the requirements set forth in the Engineering Requirements Document and the most current version of the competition rules.

5. Functionality Test

The system will be run through the course 100 times. Out of these runs, it may fail at any stage of the competition once. This confirms a reliability of 99% which is in accordance with the 98.5% reliability specified in the Engineering Requirements

6. Main Diagram/Schematics

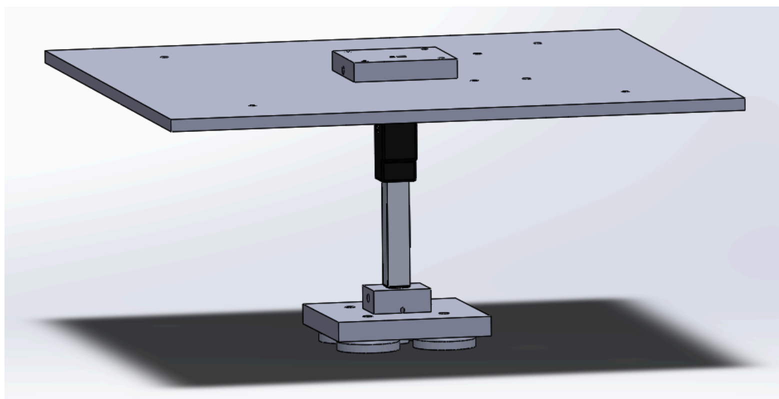


1.1 Treasure Chest Mechanism

Checks

[x] Tech Team
[x] Software
[x] Hardware

Visuals



Operational Description

First, the robot will move over top of the treasure chest. Once the gap in the bottom of the robot is centered with the treasure chest, the Actuonix L16-R linear actuator (black) will extend from its stowed position, lowering the platform (orange) with neodymium magnets (grey) until it is magnetically secured to the treasure chest (green). After the chest is attached, the L16-R will retract the chest until it is off the ground and centralized inside the robot chassis. After the chest is retracted, the robot will continue the course.

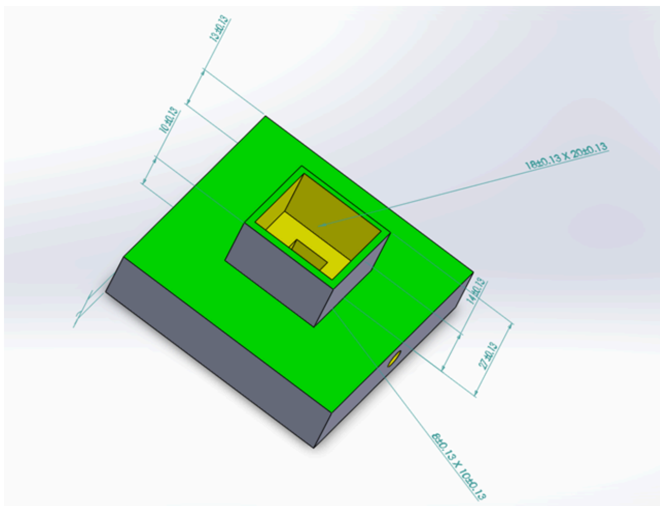
Material List

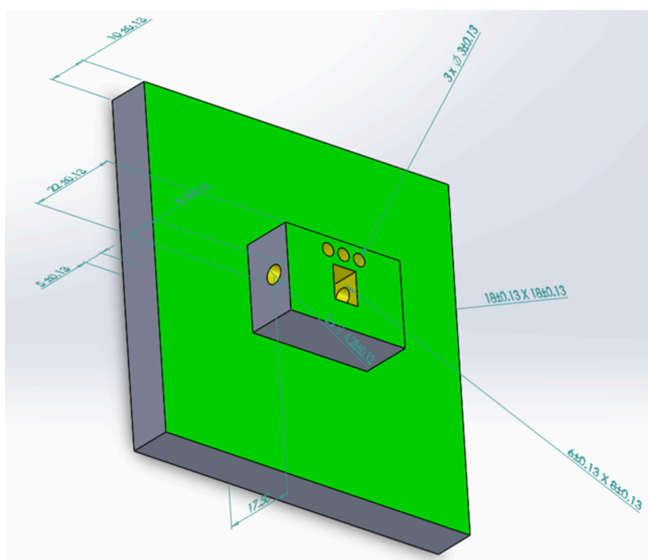
Magnet platform (to be 3D printed)
Neodymium magnets (to be secured to magnet platform)
J-B Weld (to secure magnets to magnet platform)
L16-R Linear Actuator (to raise/lower magnet platform)

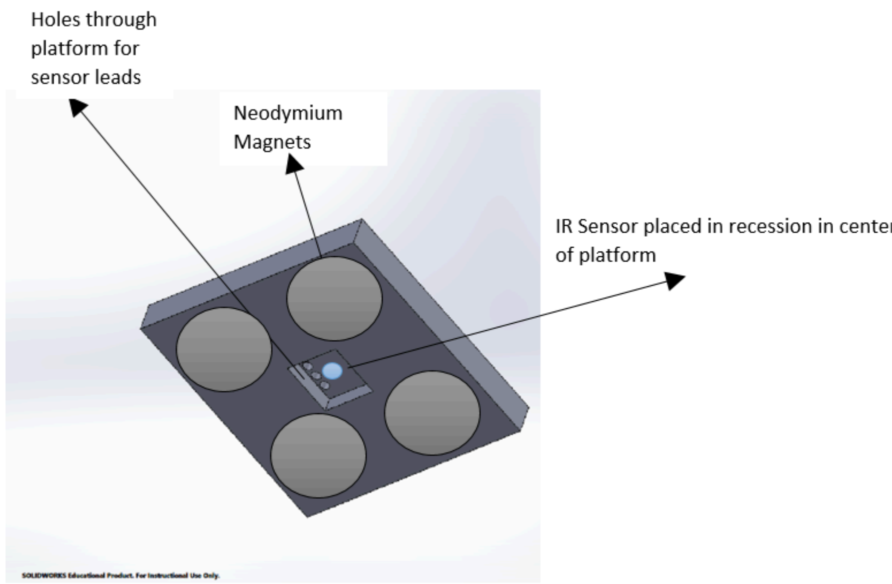
Functionality Test

This mechanism must properly lower magnet platform, then raise the platform/secured treasure chest to the appropriate height within the robot chassis.

Additional Aids







Actuonix L16 “Miniature Linear Actuators”
Below is some theoretical code:

```
//attach the servo to a pin
#include Servo.h
Servo LHS; (Left hand sided servo)
LHS.attach(4);
Write(80); //arbitrary angle
Delay(1000);
Write(0); //return to start
```

1.2 Chassis

Checks

[x] Tech Team
[x] Software
[x] Hardware

Operational Description

The Chassis is the backbone of the robot which all of our sensors, actuators, and motors will be attached to. It needs to be strong enough to support the weight and also firm enough so that our navigation will not be affected. The chassis will also have 2 levels so that there will be enough space for the circuits but also for the battery and all of our sensors, etc.

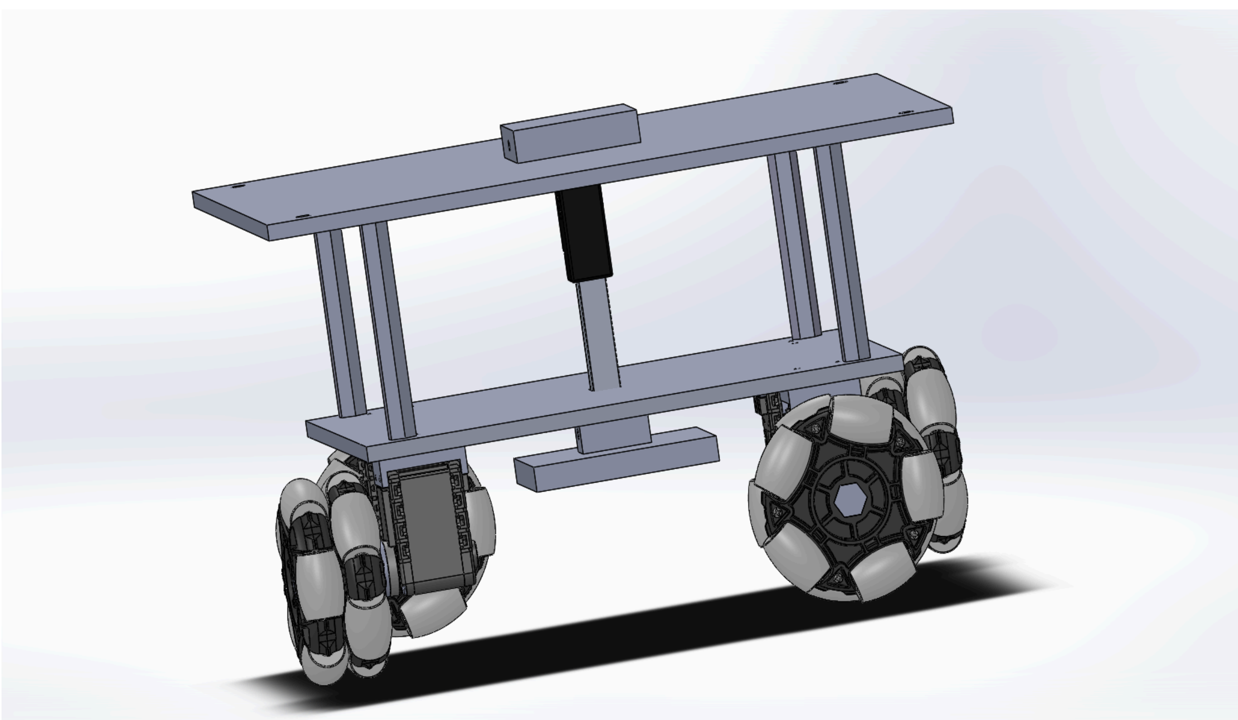
Material List

Teensy 3.6
RX-28
AX-12A
Breadboards/PCB
4 bump sensors
4 TOF sensors
Voltage Regulators
Male/Female plugs for the Battery
IR Receiver
2 Linear Actuators

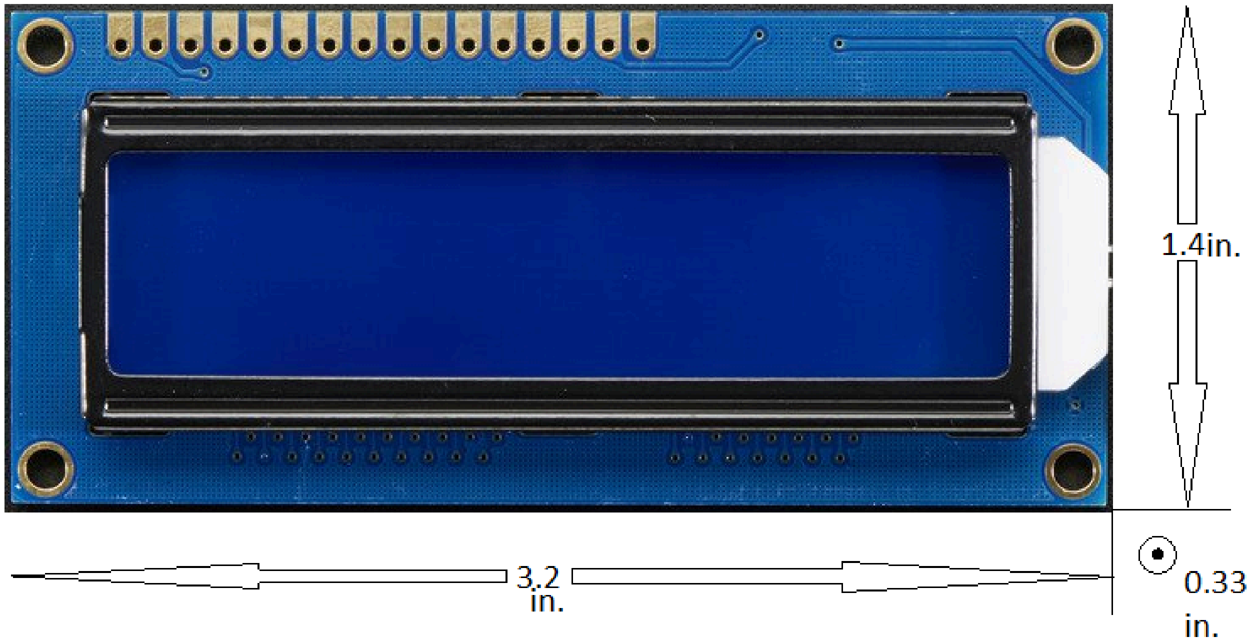
Functionality Test

Every Component has been tested individually and functions 100% properly. Except for IR but we do not have those in yet and they have been modeled with regular photodiodes.

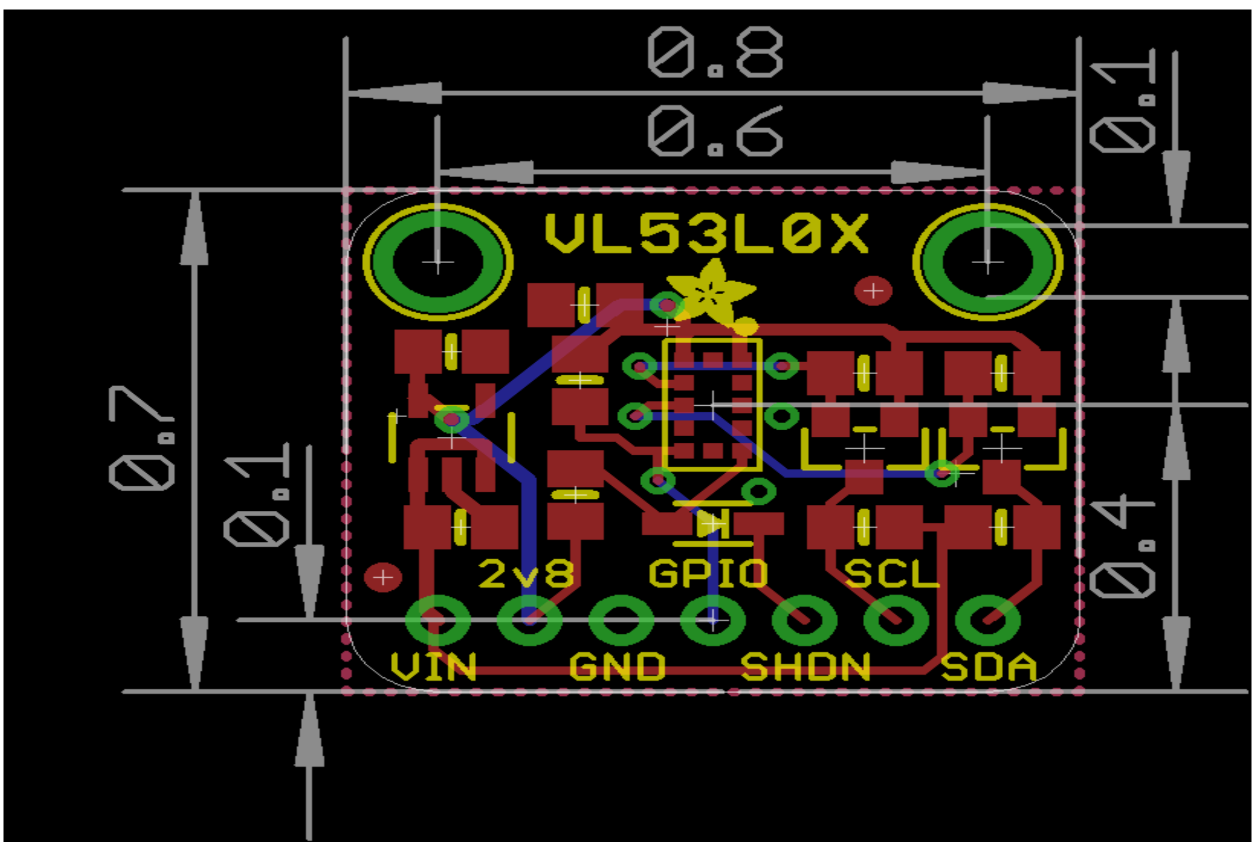
TOF sensors Work
Bumps Work
Movement (Strafing, Forward, Back)
IR RX Reading Signal
LCD Displaying



Chassis 3D Model



LCD Screen



Time of Flight Dimensions

PURPOSE	LEFT SIDE	RIGHT SIDE	PURPOSE
GND	**GND**	**Vin (3.6-6.0 V)**	5V IN
	0	**Analog GND**	GND
	1	**3.3V (250mA max)**	Power Supply for TOF
Direction Pin RX-28	**2**	**23**	
Direction Pin AX-12A	**3**	**22**	
	4	**21**	
	5	**20**	
	6	**19**	(SCL0) T.O.F.
RX-28 RX3	**7**	**18**	(SDA0) T.O.F.
RX-28 TX3	**8**	**17**	
AX-12A RX2	**9**	**16**	
AX-12A TX2	**10**	**15**	(A1) BUMP 2 IN
	11	**14**	(A0) BUMP 1 IN
	12	**13**	
	3.3V	**GND**	GND
LCD RS	**24**	**A22**	
LCD EN	**25**	**A21**	
LCD DB4	**26**	**39**	
LCD DB5	**27**	**38**	Shutdown TOF
LCD DB6	**28**	**37**	Shutdown TOF
LCD DB7	**29**	**36**	Shutdown TOF
	30	**35**	Shutdown TOF
IR Rx	**31**	**34**	
	32	**33**	

Teensy 3.6 Pinout

1.4 IR Receiver

Checks

[X] Tech Team
[X] Software
[X] Hardware

Functional Decomposition [Diagram]

Actuator to Platform

Operational Description

IR Emitter: This part is not part of the robot; it is a stage element. The emitter will send out a synch bit start burst signal for 9 ms, then a 4.5 ms space, followed by 8 bits that contain the message, and then an end burst. The first 5 bits of the message will be sent as Logical 0s, and then the last 3 bits will define the coordinates for the route that needs to be followed. Logical 0s have a length of 1.12 ms and Logical 1s have a length of 2.25 ms. The message will be sent every 5 seconds.

IR Receiver: This device will descend, attached to the bottom of the actuator platform, to a maximum range of no more than 2 inches (maximum effective range). We will use code to read the times of each pulse and by reading the pulse we will know where in the message the signal is and whether it is logical 1 or 0. Once we find out the last 3 bits, we will store it in an array and from there, the receiver will communicate with the Teensy to write the array to the LCD Display.

Dependencies [Graphic]

IR Transmitter

Material List

Standard LCD 16x2

IR Receiver Sensor

Teensy 3.6

Functionality Test

This test must prove that the IR Receiver can properly read in the signal message sent and chop it into a three bit code. Once the code is read by the IR Receiver it must display the proper code on the LCD screen.

Diagrams/Schematics

```
const byte ledPin = 13;
const byte interruptPin = 2;
volatile byte state = LOW;
int sigArray [10000][2];
int indexSigArray;
int indexRead;
int tNow;
int tLast;
int logic;
int Map [3];

void setup() {
  Serial.begin(9600);
  tNow = 0;
  tLast = 0;
  indexSigArray = 0;
  pinMode(ledPin, OUTPUT);
  pinMode(interruptPin, INPUT);
  attachInterrupt(digitalPinToInterrupt(interruptPin), blink, CHANGE);
}

void loop() {
  digitalWrite(ledPin, state);
  delay(50);
  process();
}

void blink()
{
  tNow = micros();

  if(digitalRead(interruptPin) == HIGH)
  {
          sigArray[indexSigArray][0] = 1;
  }
```

```
 else
 {
        sigArray[indexSigArray][0] = 0;
 }

 sigArray[indexSigArray][1] = tNow - tLast;
 tLast = tNow;
 indexSigArray++;
}

void process()
{
 for(;indexRead < indexSigArray - 17; indexRead++)
 {
        Serial.print("Time in Micro-Seconds: ");
        Serial.print(sigArray[indexRead][1]);
        Serial.print(", ");
        Serial.print(sigArray[indexRead][0]);
        Serial.print("\n");

        if(sigArray[indexRead][0] == 1 &&
        sigArray[indexRead][1] > 8000 &&
        sigArray[indexRead][1] < 10000)
        {
        if(sigArray[indexRead + 1][0] == 0 &&
        sigArray[indexRead + 1][1] > 4000 &&
        sigArray[indexRead + 1][1] < 5000 &&

        sigArray[indexRead + 2][0] == 1 &&
        sigArray[indexRead + 2][1] > 300 &&
        sigArray[indexRead + 2][1] < 800 &&

        sigArray[indexRead + 3][0] == 0 &&
        sigArray[indexRead + 3][1] > 300 &&
        sigArray[indexRead + 3][1] < 800 &&

        sigArray[indexRead + 4][0] == 1 &&
        sigArray[indexRead + 4][1] > 300 &&
        sigArray[indexRead + 4][1] < 800 &&

        sigArray[indexRead + 5][0] == 0 &&
        sigArray[indexRead + 5][1] > 300 &&
        sigArray[indexRead + 5][1] < 800 &&

        sigArray[indexRead + 6][0] == 1 &&
        sigArray[indexRead + 6][1] > 300 &&
        sigArray[indexRead + 6][1] < 800 &&
```

```
        sigArray[indexRead + 7][0] == 0 &&
        sigArray[indexRead + 7][1] > 300 &&
        sigArray[indexRead + 7][1] < 800 &&

        sigArray[indexRead + 8][0] == 1 &&
        sigArray[indexRead + 8][1] > 300 &&
        sigArray[indexRead + 8][1] < 800 &&

        sigArray[indexRead + 9][0] == 0 &&
        sigArray[indexRead + 9][1] > 300 &&
        sigArray[indexRead + 9][1] < 800 &&

        sigArray[indexRead + 10][0] == 1 &&
        sigArray[indexRead + 10][1] > 300 &&
        sigArray[indexRead + 10][1] < 800 &&

        sigArray[indexRead + 11][0] == 0 &&
        sigArray[indexRead + 11][1] > 300 &&
        sigArray[indexRead + 11][1] < 800)
        {
        digitalWrite(ledPin, HIGH);
        Serial.print("Positive ID\n");
        if(sigArray[indexRead + 13][1] > 300 &&
        sigArray[indexRead + 13][1] < 800)
        {
          Map[0] = 0;
        }
        else
        {
        Map[0] = 1;
        }

        if(sigArray[indexRead + 15][1] > 300 &&
        sigArray[indexRead + 15][1] < 800)
        {
        Map[1] = 0;
        }
        else
            {
        Map[1] = 1;
        }

        if(sigArray[indexRead + 17][1] > 300 &&
        sigArray[indexRead + 17][1] < 800)
        {
        Map[2] = 0;
        }
        else
```

```
        {
        Map[2] = 1;
        }

        Serial.print("Map: ");
        Serial.print(Map[0]);
        Serial.print(", ");
        Serial.print(Map[1]);
        Serial.print(", ");
        Serial.print(Map[2]);
        Serial.print("\n");
        }
        }
 }
```

1.6 Stage A Button Mechanism

Checks

[x] Tech Team
[x] Software
[x] Hardware

Operational Description

The Stage A button press will simply be an acrylic sheet mounted on both sides of the robot. We will mount a limit switch to these platforms which will activate when the Stage A button is pressed. This will ensure we have proper feedback for the robot so it will know to move on to the next stage of competition.

Dependencies

Chassis
Time of Flight Sensors
Bump Sensors

Material List

Acrylic Sheet
Acrylic Cement
Bump Sensors

Functionality Test

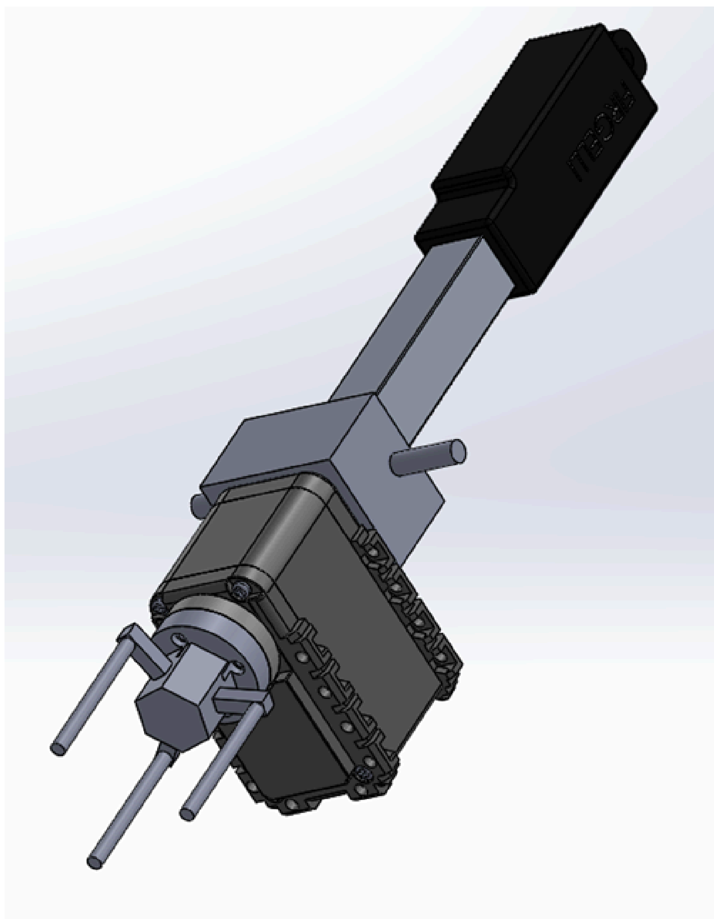
We will test the functionality of the mechanism by conducting a series of tests with different buttons and depth of recessions in the Stage A wood planks. If the mechanism can depress the button from an inch error from the center of the button in any direction, the design will be successful.

1.10 Flag Raise Mechanism

Checks

[x] Tech Team
[x] Software
[x] Hardware

Visuals



Operational Description

Extend the actuator until it is outside the chassis. Then, the robot will drive forward until the rotating arm is centered in front of the ‘Captain’s Wheel’. Once in place, the AX-12 will begin rotating until the flag is raised.

Dependencies

AX-12 properly extended from the robot using an actuator, and linearly stable.

3 prongs extended from the AX-12. Within the prong design, it is a two part. One is the extrusion perpendicular to the AX-12 for optimum circumference match to the wheel. The next is a perpendicular extrusion to pierce the gaps of the wheel.
Flag Destination: The flag will be raised via a 3D printed pirate ship wheel attached to a rotary encoder. The encoder is located on a 10”x15” of ¾” plywood. Which is centered 24” from the Northern edge of the field, mounted on the outside of the 2x4 with 10.875” of the board above the Eastern 2x4. The rotary encoder is centered 7” from the top of the 2x4 and on the centerline of the short side of the 10”x15” plywood piece.

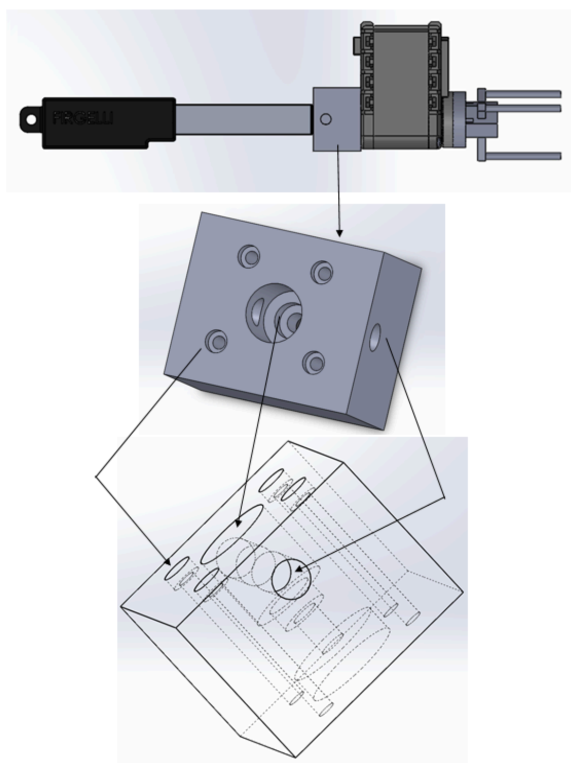
Material List

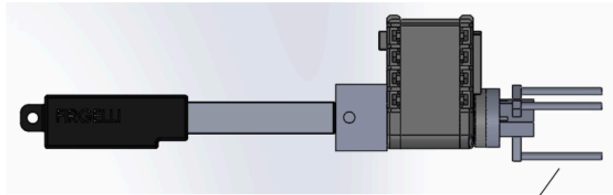
AX-12
L-16 Actuator 3D printed AX-12 Mount
3D printed arm to rotate wheel

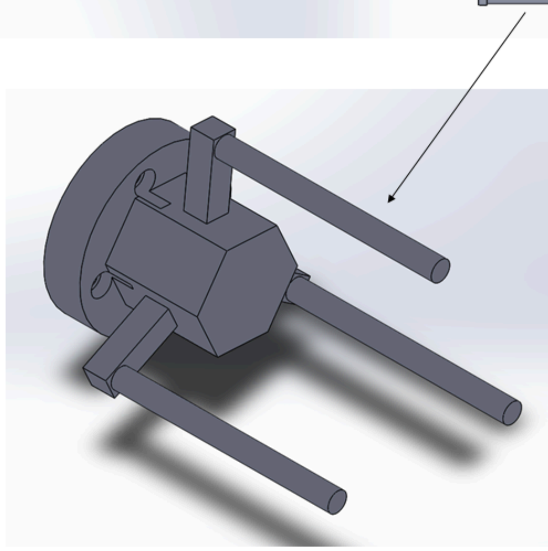
Functionality Test

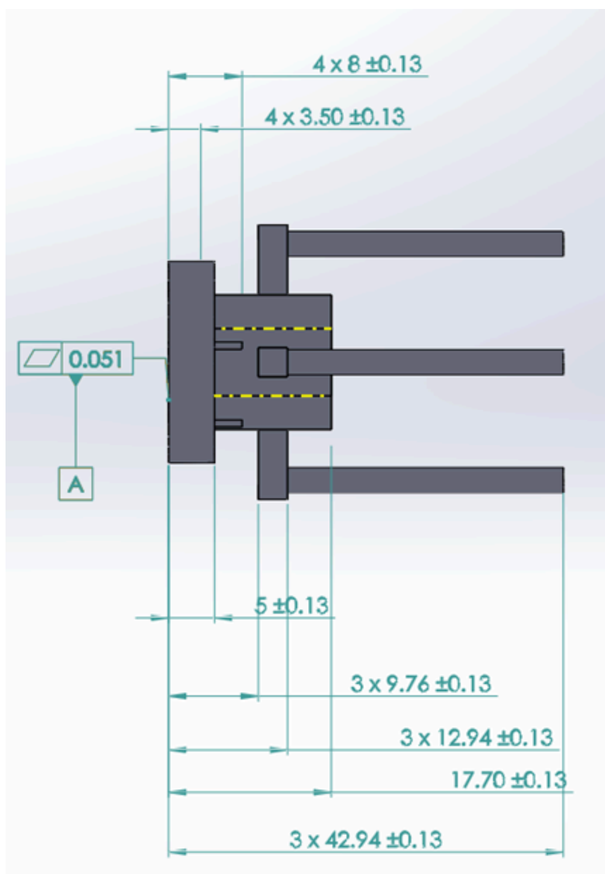
Test the height off the ground to interface with the Captain’s Wheel.
Test the approach of the robot to the wheel.
Test the amount of trials the mechanism can complete without fail.

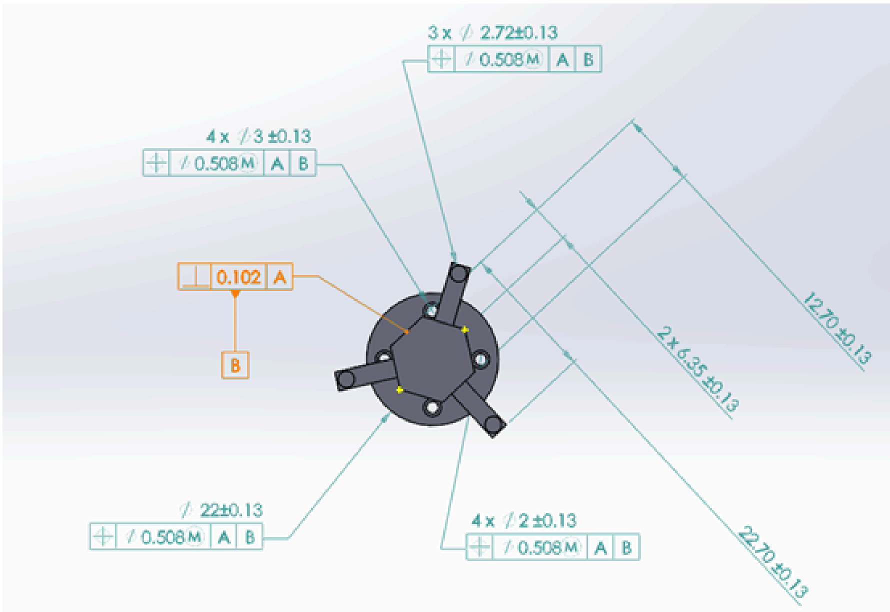
Additional Aids








4 x 8 ±0.13
4 x 3.50 ±0.13
0.051
A
5 ±0.13
3 x 9.76 ±0.13
3 x 12.94 ±0.13
17.70 ±0.13
3 x 42.94 ±0.13



1.11 Power Supply

Checks

[x] Tech Team
[x] Software
[x] Hardware

Functional Decomposition

The Chassis from the tech team standpoint has a few aspects that are the most important. Also there are some aspects that can be assumed as givens.

Level 0: Chassis Design
Level 1: a) Size of Chassis Layers
b) Dynamixel Current draw
c) Component Current tolerance

Operational Description

The Purpose of this power breakdown is to ensure that at any moment during the life of our project that we have a sufficient amount of power supplied to our system. This will ensure consistent and low noise which is crucial to testing and development of our system. We have decided to work with more power than needed and regulate down because that will ensure consistency and low noise is delivered to our system.

Dependencies

- Budget
- Physical Size / weight

Material List

- 3.3,5,6,10/12 Voltage regulators
- 7.2 V/ 22.2 V LiPo Battery

Functionality Test

- Ensure proper power is delivered to the appropriate power bus
- Ensure the consistency of system

References / In-Depth Documentation / Additional Information

Qty	**Description**	**Operating Voltage (V)**	**Com Voltage Max (V)**	**Avg Current Draw (mA)**	**Max Current Draw (mA)**	AVG Power(mW)	Max Power (mW)
Sensors							
5	Dynamixel Motor	11.20	5.00	250.00	900.00	2800	10080
2	TOF VL53L0X	3.30		19.00	40.00	62.7	132
2	TOF VL6180	3.30		1.70	250.00	5.61	825
1	Teesny 3.6	3.30	3.30	NA	300.00		990

1	Adafruit LCD	5.00	5.20	35.00	40.00	175	200
2	Bump Sensors	NA	NA	NA	NA		
2	Linear Actuator	6.00	5.00	150.00	500.00	900	3000
1	IR Rx	NA	NA	NA	NA		
Statistics:		**Power (mW)**					
	Total Power	15,227.00					
	Avg Power	7,500.00					

Regulators:

3.3 V → https://www.adafruit.com/product/2236

5 V → https://www.jameco.com/z/L7806CV-STMicroelectronics-Standard-Regulator-6-Volt-1-5-Amp-3-Pin-3-Tab-TO-220-Tube_924668.html

6 V → https://www.adafruit.com/product/2166

10 V → https://www.jameco.com/z/L7912CV-STMicroelectronics-Standard-Regulator-12-Volt-1-5-Amp-3-Pin-3-Tab-TO-220_889524.html

12 V → https://www.jameco.com/z/LM2940T-10-0-NOPB-Texas-Instruments-Low-Dropout-Regulator-10-Volt-1A-3-Pin-3-Tab-TO-220-Rail_837441.html

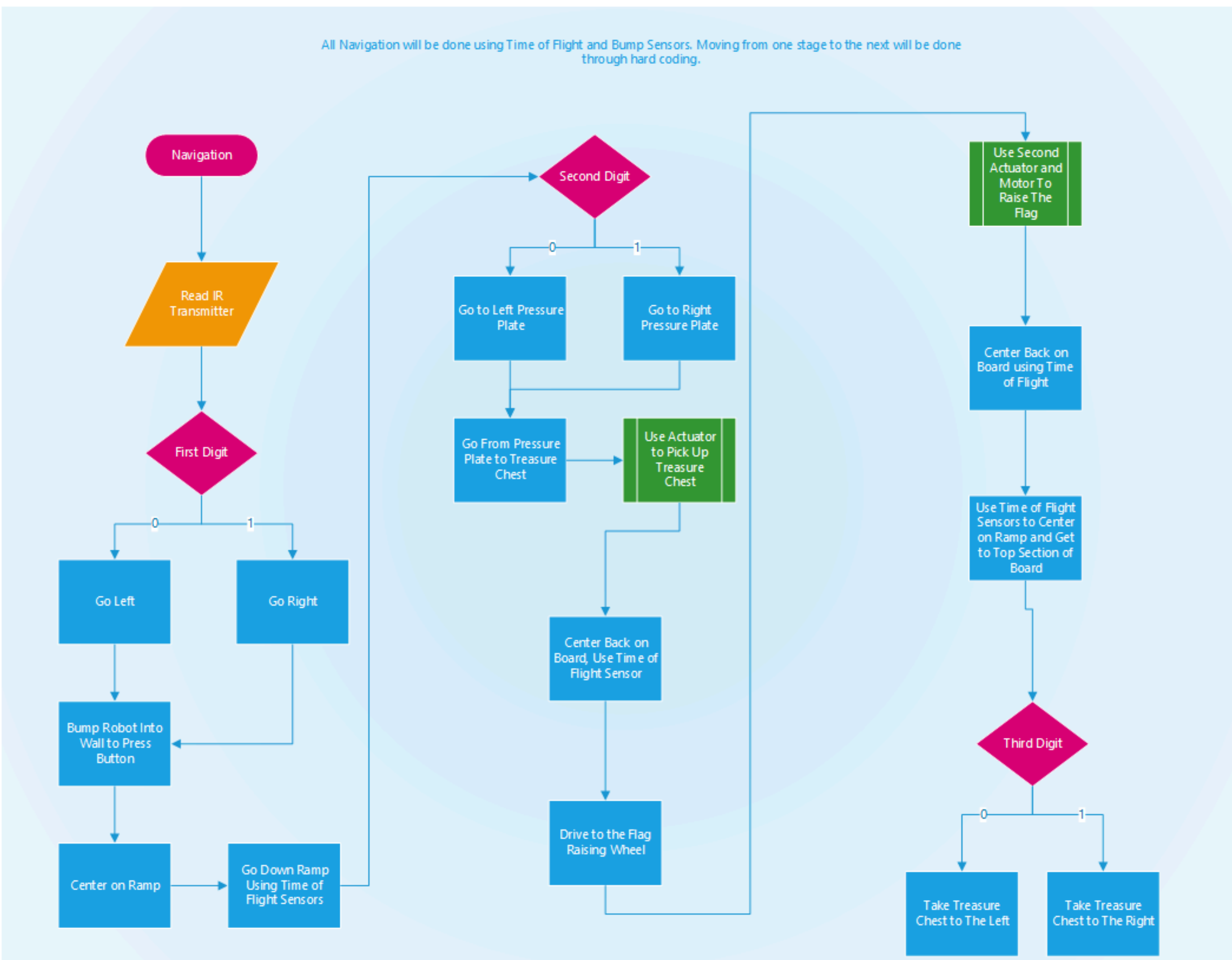
Power Analytics Link:
https://docs.google.com/spreadsheets/d/1MRJOkO2pHgWeYnhSWhsD2hVa_Do2689yxK7091xumYs/edit#gid=0

1.12 and 1.13 Navigation and Main Function

Checks

[x] Tech Team
[x] Software
[x] Hardware

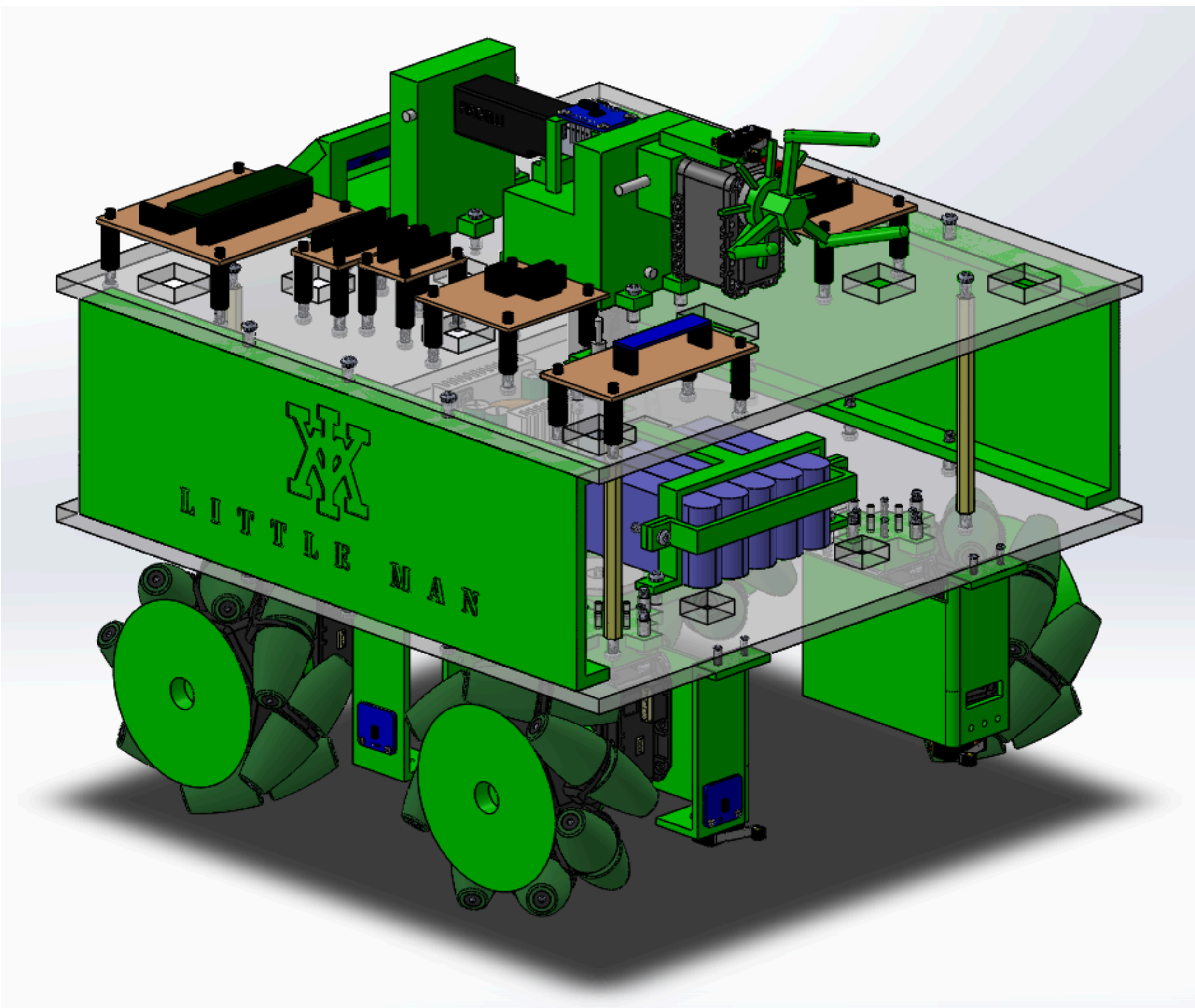
Flowchart



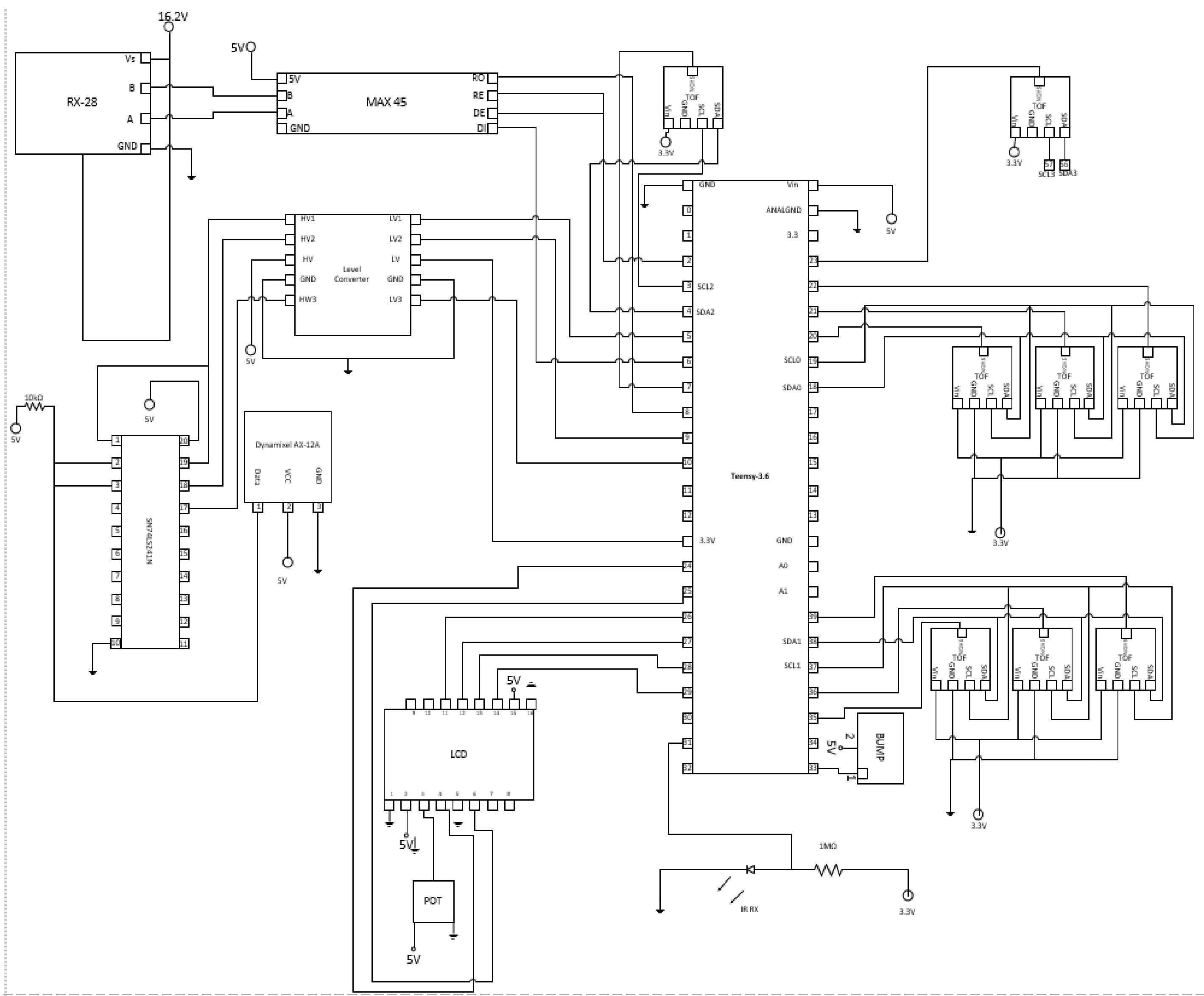
Final System Documentation

After months of refinement, we transitioned from PDRs to CDRs, to our final designs in this section of the document.

3D Model



Circuit Schematic



Microcontroller Pinout

PURPOSE	LEFT SIDE	RIGHT SIDE	PURPOSE
GND	**GND**	**Vin (3.6-6.0 V)**	5V IN
	0	**Analog GND**	GND
	1	**3.3V (250mA max)**	Power Supply for TOF
Direction Pin RX-28	**2**	**23**	
Direction Pin AX-12A	**3**	**22**	
	4	**21**	
	5	**20**	
	6	**19**	(SCL0) T.O.F.
RX-28 RX3	**7**	**18**	(SDA0) T.O.F.
RX-28 TX3	**8**	**17**	
AX-12A RX2	**9**	**16**	
AX-12A TX2	**10**	**15**	(A1) BUMP 2 IN
	11	**14**	(A0) BUMP 1 IN
	12	**13**	
	3.3V	**GND**	GND
LCD RS	**24**	**A22**	
LCD EN	**25**	**A21**	
LCD DB4	**26**	**39**	
LCD DB5	**27**	**38**	Shutdown TOF
LCD DB6	**28**	**37**	Shutdown TOF
LCD DB7	**29**	**36**	Shutdown TOF
	30	**35**	Shutdown TOF
IR Rx	**31**	**34**	
	32	**33**	

Navigation and Motor Analysis

Time Requirements

The minimum time in which a robot can finish the competition after stage A is 45 Seconds. Since the individual tasks on the contest course are rudimentary, it is feasible that other teams will complete all tasks within the allotted time frame of four minutes. The only way to realistically outperform other teams is to complete the course in the minimum time possible. Since the Stage A button becomes the activation button for stage C after 45 seconds, that is the minimum time in which the course may be completed after the Stage A Button is pushed.

Competition Task	Time (s)
Prior to Stage A	
IR	0[3]
Stage A	1
Following Stage A	
Travel (Includes Stage C)	15
Stage B	10
Treasure Chest	7.5
Flag Raising	7.5
Buffer	5
TOTAL	**46**

Motor Power Calculations

The force opposing wheel movement for the robotic system was calculated for no added weight on a level surface, with the added weight of the treasure chest on a level

[3] The IR Stage will not be given time since it is not feasible to start the competition until the data package is transmitted (the data is also transmitted in under 30 milliseconds, which is insignificant compared to the order of magnitude of the other time calculations).

surface, and with the added weight of the treasure chest on a 20° incline. The rotational frictional coefficient was generalized to be .089[4]. The frictional coefficient of the AX-12 model dynamixel motors was used as an approximation for the RX-28 Dynamixel motors since a measurement for the RX-28 Model could not be found. The coulomb frictional coefficient for the AX-12 motors was found to be 0.29. The total frictional coefficient acting against motor rotations based on the rotational force of the motors is .089 + .29 for a total frictional coefficient of .38.

Force on a Level Surface without Treasure Chest

$$F_{wL} = F_N \cdot \mu_r$$
$$F_{wL} = m \cdot g \cdot \mu_r$$
$$F_{wL} = 2.8kg \cdot 9.8\, m/s^2 \cdot 0.38$$
$$F_{wL} = 10.4N$$

Force on a Level Surface with Treasure Chest

$$F_{wLC} = F_N \cdot \mu_r$$
$$F_{wLC} = m \cdot g \cdot \mu_r$$
$$F_{wLC} = 4.1kg \cdot 9.8\, m/s^2 \cdot 0.38$$
$$F_{wLC} = 15.3N$$

Force on an Inclined Surface with Treasure Chest

$$F_{wIC} = F_N \cdot \mu_r + F_r$$
$$F_{wIC} = m \cdot g \cdot cos(\theta) \cdot \mu_r + m \cdot g \cdot sin(\theta)$$
$$F_{wIC} = m \cdot g\left[cos(\theta) \cdot \mu_r + sin(\theta)\right]$$
$$F_{wIC} = 4.1kg \cdot 9.8\, m/s^2 [cos(20°) \cdot 0.38 + sin(20°)]$$
$$F_{wIC} = 28.1N$$

Torque on Each Motor

$$\tau_m = F_w r_w$$
$$r_w = 0.03m,\ 0.05m,\ 0.076m$$

60 mm Mecanum Wheels

Force	F_w	Motor Torque (for all motors) (N·m)	F_w / 4	Motor Torque (for each motor) (N·m)

[4] https://arxiv.org/pdf/1211.2323.pdf, Page 12.

F_{wL}	10.4	0.31	2.6	0.078
F_{wLC}	15.3	0.46	3.8	0.11
F_{wIC}	28.1	0.84	7.0	0.21

100 mm Mecanum Wheels

Force	F_w	Motor Torque (for all motors) (N·m)	$F_w / 4$	Motor Torque (for each motor) (N·m)
F_{wL}	10.4	0.52	2.6	0.13
F_{wLC}	15.3	0.77	3.8	0.19
F_{wIC}	28.1	1.4	7.0	0.35

152 mm Mecanum Wheels

Force	F_w	Motor Torque (for all motors) (N·m)	$F_w / 4$	Motor Torque (for each motor) (N·m)
F_{wL}	10.4	0.79	2.6	0.20
F_{wLC}	15.3	1.2	3.8	0.29
F_{wIC}	28.1	2.1	7.0	0.53

Motor Speed Requirements

In order for the Robotic System to navigate the course with 15 seconds of travel time, it must move with a minimum speed of .4 m/s (see equation 1).

$$V = X_T / Time_T$$

$$V = 6.2\, m / 15\, s = 0.4\, m/s \quad \text{(Equation 1)}$$

RX-28 Dynamixel motors were tested at 18.5V and were observed to move at 1.5 revolutions per second. The motors have a stall torque (μ_r)of 3.7 N·m. Since torque and angular velocity are linearly dependant, the speed may easily be obtained for any torque given the above information (see equation 2).

$$RPM = -\frac{90}{3.7}\tau + 90$$

$$RPM = -24.3\tau + 90 \qquad \text{(Equation 2)}$$

$$V = 2 \cdot \pi \cdot r_w \cdot RPM / 60$$

60 mm Mecanum Wheels

Force	Motor Torque (for each motor) (N·m)	Max RPM	Max Speed (m/s)
F_{wL}	0.078	88	0.28
F_{wLC}	0.11	87	0.27
F_{wIC}	0.21	85	0.27

Force	Max Speed (m/s)	Distance (m)	Time (s)
F_{wL}	0.28	2.8	10
F_{wLC}	0.27	2.6	9.6
F_{wIC}	0.27	0.31	1.6

TOTAL TIME (S)	27
REQUIRED TIME (S)	15
BUFFER TIME (S)	-12

100 mm Mecanum Wheels

Force	Motor Torque (for each motor) (N·m)	Max RPM	Max Speed (m/s)
F_{wL}	0.13	87	0.46
F_{wLC}	0.19	85	0.45
F_{wIC}	0.35	81	0.42

Force	Max Speed (m/s)	Distance (m)	Time (s)
F_{wL}	0.46	2.8	6.1
F_{wLC}	0.45	2.6	5.8
F_{wIC}	0.42	0.31	0.74

TOTAL TIME (S)	12.6
REQUIRED TIME (S)	15
BUFFER TIME (S)	2.4

Projected Time moving to Stage A

Force	Max Speed (m/s)	Distance (m)	Time (s)
F_{wL}	0.46	.46	1.0

152 mm Mecanum Wheels

Force	Motor Torque (for each motor) (N·m)	Max RPM	Max Speed (m/s)
F_{wL}	0.20	85	0.68
F_{wLC}	0.29	83	0.66
F_{wIC}	0.53	77	0.61

Force	Max Speed (m/s)	Distance (m)	Time (s)
F_{wL}	0.68	2.8	4.1
F_{wLC}	0.66	2.6	3.9
F_{wIC}	0.61	0.31	0.51

TOTAL TIME (S)	8.5

REQUIRED TIME (S)	15
BUFFER TIME (S)	6.5

Projected Time moving to Stage A

Force	Max Speed (m/s)	Distance (m)	Time (s)
F_{wL}	0.46	.46	1.0

Power Usage

$$Time = \frac{Distance}{V_{Max}}$$

$$E_{Mech\ Out} = F_w \cdot Distance \cdot \frac{1000\ mWh}{3600\ J}$$

$$P_{Mech\ Out} = \frac{F_w \cdot Distance}{Time}$$

Force	F_w	Distance (m)	Time (s)	$E_{Mech\ Out}$ (All Motors) (mWh)	$P_{Mech\ Out}$ (All Motors) (W)
F_{wL}	10.4	2.8	6.1	8.1	4.8
F_{wLC}	15.3	2.6	6.2	11	6.4
F_{wIC}	28.1	0.31	0.8	2.4	11

TOTAL POWER CONSUMPTION (mWh)	**22**

Power Draw from Battery

The motors used in the RX-28 Dynamixel Model have a maximum efficiency of 85%[5]. All Calculations are done given a voltage of 18.5V.

Avg Power Draw:

$$E_{Avg}(W) = \frac{(4.8\ W)(6.1\ s) + (6.4\ W)(6.2\ s) + (11\ W)(0.31\ s)}{(0.85)} \cdot \frac{(1000\ m)\ Hours}{3600\ s} = 20\ mWh$$

Max Power Draw:

$$E_{Max}(W) = \frac{(11\ W)(0.31\ s)}{(.85)} \cdot \frac{(1000\ m)\ Hours}{3600\ s} = 3.6\ mWh$$

[5] https://www.maxonmotorusa.com/maxon/view/product/motor/dcmotor/remax/remax17/214897

Avg Power Draw:

$$P_{Avg}(W) = \frac{(4.8\ W)\ (6.1\ s) + (6.4\ W)(6.2\ s) + (11\ W)(0.31\ s)}{(12.6\ s)(0.85)} = 5.8\ W$$

Max Power Draw:

$$P_{Max}(W) = \frac{11\ W}{(.85)} \cdot 1000 = 13\ W$$

Avg Current Draw:

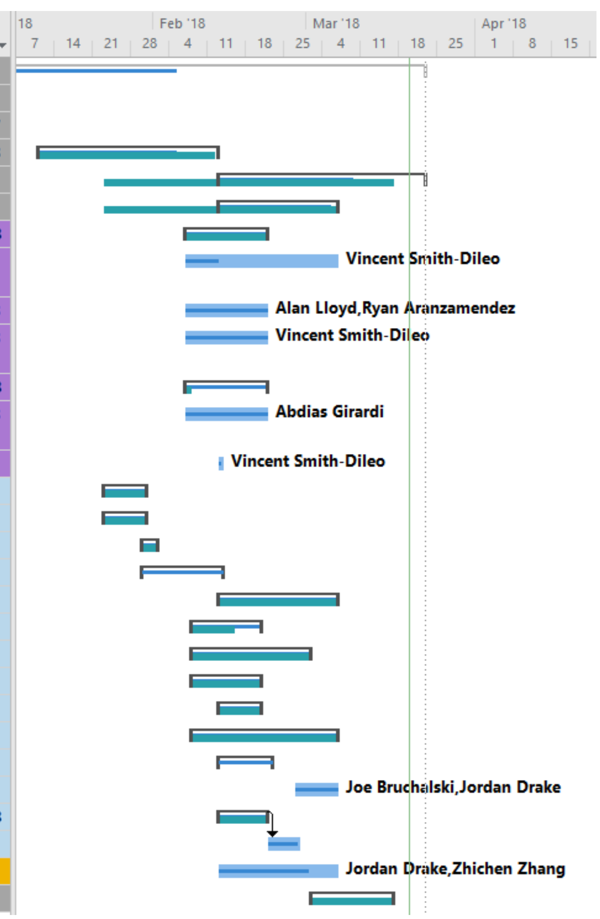
$$I_{Avg}(mA) = \frac{(4.8\ W)\ (6.1\ s) + (6.4\ W)(6.2\ s) + (11\ W)(0.31\ s)}{(12.6\ s)(0.85)(18.5\ V)} \cdot 1000 = 310\ mA$$

Max Current Draw:

$$I_{Max}(mA) = \frac{11\ W}{(18.5\ V)(.85)} \cdot 1000 = 700\ mA$$

Refinement Phase Timeline

Outline Numbe	Task Name	% Comp	Duration	Start	Finish
0	**Capstone**	**76%**	**146 days?**	**Mon 9/4/17**	**Thu 3/22/18**
1	**Design**	**100%**	**25 days**	**Mon 9/4/17**	**Thu 10/5/17**
2	**Construction**	**72%**	**43 days**	**Thu 10/5/17**	**Mon 12/4/17**
3	**V3 System Integration**	**74%**	**23 days**	**Thu 1/11/18**	**Mon 2/12/18**
4	**V3 System Refinement**	**65%**	**29 days**	**Tue 2/13/18**	**Thu 3/22/18**
4.1	**Refinement Part 1**	**93%**	**16 days**	**Tue 2/13/18**	**Tue 3/6/18**
4.1.1	**Part Refinement**	**98%**	**11 days**	**Wed 2/7/18**	**Wed 2/21/18**
4.1.2	Print Finalized Parts and Replacements	20%	20 days?	Wed 2/7/18	Tue 3/6/18
4.1.3	Keep Drawings Up to Date	100%	11 days?	Wed 2/7/18	Wed 2/21/18
4.1.4	Keep Exploded View/ Animation Up to Date	100%	11 days?	Wed 2/7/18	Wed 2/21/18
4.1.5	**Model Parts in Solidworks**	**100%**	**11 days**	**Wed 2/7/18**	**Wed 2/21/18**
4.1.6	Structural testing/reliability on Solidworks	100%	11 days?	Wed 2/7/18	Wed 2/21/18
4.1.7	Print Additional Parts	20%	1 day?	Tue 2/13/18	Tue 2/13/18
4.1.8	**Code Optimization**	**100%**	**6 days**	**Tue 1/23/18**	**Tue 1/30/18**
4.1.9	**Debugging**	**100%**	**6 days**	**Tue 1/23/18**	**Tue 1/30/18**
4.1.10	**Code Mirroring**	**100%**	**3 days**	**Tue 1/30/18**	**Thu 2/1/18**
4.1.11	**Full Code Integration**	**100%**	**11 days?**	**Tue 1/30/18**	**Tue 2/13/18**
4.1.12	**Navigation**	**100%**	**16 days**	**Tue 2/13/18**	**Tue 3/6/18**
4.1.13	**SCL/SDA**	**100%**	**9 days**	**Thu 2/8/18**	**Tue 2/20/18**
4.1.14	**TOF reset**	**100%**	**16 days**	**Thu 2/8/18**	**Thu 3/1/18**
4.1.15	**Board**	**100%**	**9 days**	**Thu 2/8/18**	**Tue 2/20/18**
4.1.16	**Wheel Verification**	**100%**	**6 days**	**Tue 2/13/18**	**Tue 2/20/18**
4.1.17	**IR Generator**	**100%**	**19 days**	**Thu 2/8/18**	**Tue 3/6/18**
4.1.18	**Gyro**	**100%**	**8 days?**	**Tue 2/13/18**	**Thu 2/22/18**
4.1.19	Wire Decluttering	100%	6 days?	Tue 2/27/18	Tue 3/6/18
4.1.20	**PCB Integration**	**100%**	**7 days**	**Tue 2/13/18**	**Wed 2/21/18**
4.1.21	Verify V3 Code Functionality	80%	4 days?	Thu 2/22/18	Tue 2/27/18
4.1.22	Refine PCBs	80%	16 days	Tue 2/13/18	Tue 3/6/18
4.2	**Refinement Part 2**	**0%**	**11 days**	**Fri 3/2/18**	**Fri 3/16/18**



ENGINEERING REQUIREMENTS

Requirements for Points Earned Outside of Round: All members must understand how points are earned.

1. Team Logo should be displayed on the team flag, shirts, and robot.
2. Hardware team must wear team shirts throughout competition.

Robot Specific Requirements: The robot must be designed to meet each of these in order to qualify to compete.

1. The robot must be a single unit.
2. The robot's size must not be greater than 12"x12"x12" at the beginning of each round.
3. The robot's extended size during the competition must not be greater than 20"x20"x20".
4. The robot cannot include any form of wireless communication and cannot fly.
5. The robot must be autonomous and use IR sensors to receive given coordinate locations.
6. The robot's logo must be identifiable on robot and must contain school colors.

Playing Field Requirements: The robot in general must do this to begin, and to complete the course.

1. The robot must be at the starting sequester location 15 minutes before the round starts.
2. The robot must be able to complete the course and objectives in under 4 minutes
3. The robot must be able to compete in 2 rounds before the final round with 95% (TBR) reliability.
4. The robot must be able to read and maneuver around the playing field.
5. The robot must not be in contact with water or will result in the round ending.
6. The robot must raise a laminated paper or fabric flag no larger than 8"x12" that is attached to a flagpole no longer than 2 feet and no wider than 1" diameter.

Modules: The robot must be able to complete each objective at each destination in chronological order and return to starting point for maximum points.

1. The robot must display correct binary code on LCD screen with 99.79% (TBR) reliability.
2. The robot must be able to press a limit switch at destination A with 99.79% (TBR) reliability.
3. The robot must cross bridge from ship with 95% (TBR) reliability.
4. The robot must be able to press a limit switch at destination B with 99.79% (TBR) reliability.
5. The robot must be able to pick up the treasure chest off the playing field using designed mechanism at destination B with 99.79% (TBR) reliability.
6. The robot must move the treasure chest with without dropping it for the rest

of the round with 99.79% (TBR) reliability.

7. The robot must be able to use a mechanism to turn the wheel and raise the flag at destination B with 99.79% (TBR) reliability.

8. The robot must return to the destination A (destination C) after completing all objectives with the treasure chest with 99.79% (TBR) reliability.

Reliability of the Robot:

Our goal for reliably completing all 3 rounds of competition is 95% ($R_{Competition}$). To ensure this level of reliability, we must derive the reliability required for 1 round of competition (R_{System}), as well as the reliability of each module required (R_{Module}). If we start with the reliability of completing all 3 rounds of competition successfully, we may work backwards to find the reliability of each module that would be required to ensure 95% reliability for completing 3 rounds of competition.

Derivations of Reliability:

If we want the robot to successfully complete 3 rounds of competition with a 95% reliability ($R_{Competition}$), the reliability of the robot for one round required (R_{System}) is:

$$R_{Competition} = R_{System}^{3}$$
$$95\% = R_{System}^{3}$$
$$\sqrt[3]{95\%} = R_{System}$$
$$R_{System} = 98.3\%$$

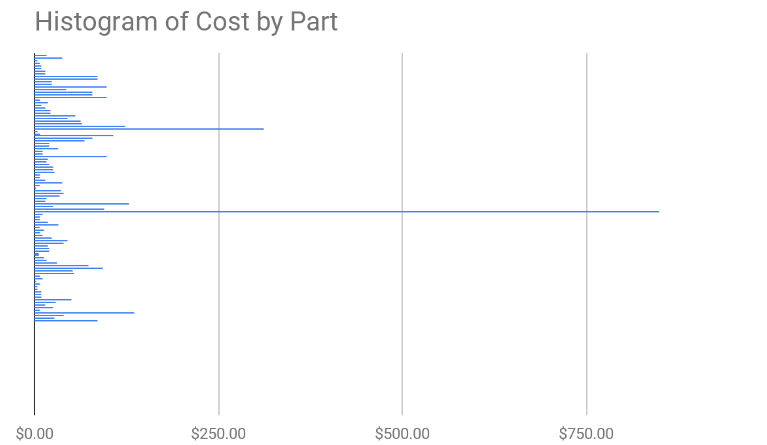
This calculation implies that the robot must be able to complete 1 round of competition with 98.3% reliability to complete all 3 rounds with 95% reliability.

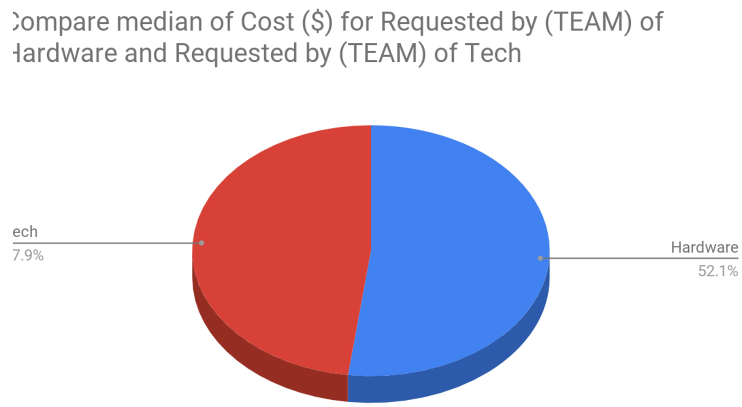
Now that we have derived the reliability for the robot to complete 1 round of competition (R_{System}), now we may derive the modular reliability (R_{Module}) with 8 modules to acquire a competition reliability ($R_{Competition}$) of 95%.

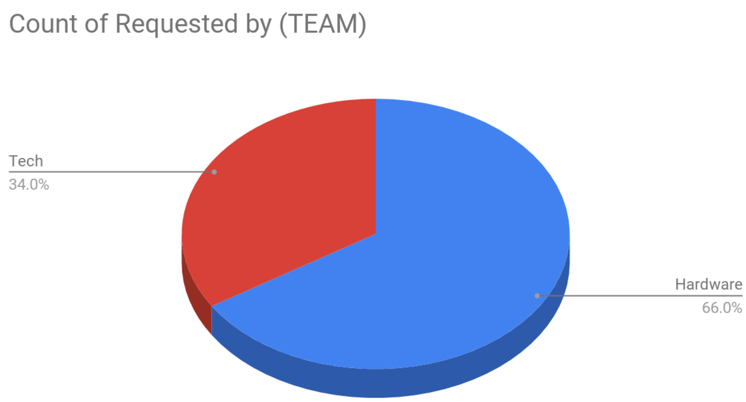
$$R_{System} = R_{Module}^{8}$$
$$98.3\% = R_{Module}^{8}$$
$$\sqrt[8]{98.3\%} = R_{Module}$$
$$R_{Module} = 99.79\%$$

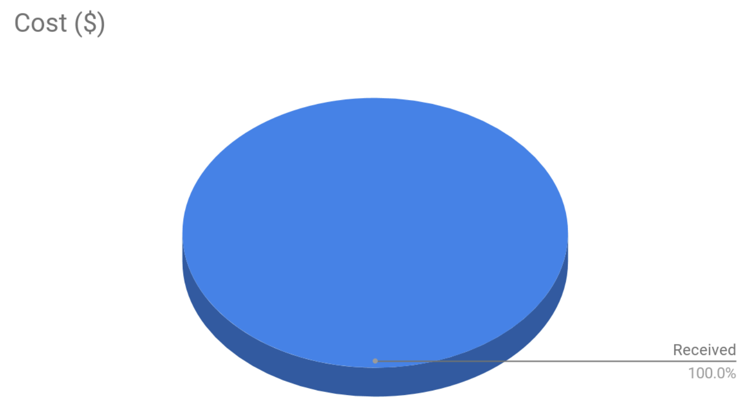
From this calculation, we have concluded that each of the 8 modules must have a reliability of 99.79% to ensure that we complete all 3 rounds of competition with 95% reliability. Our strategy for ensuring this reliability is to run the robot through the entire course as many times as possible, recording any failures that occur. We will identify modules which need improvement to bring our competition reliability ($R_{Competition}$) up to the desired 95%.

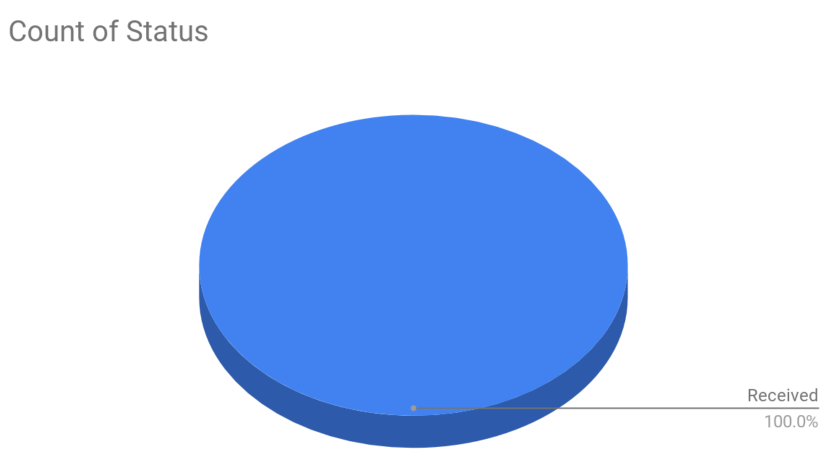
Procurement and Finance











Project Accounts and Budget

Initial Values		Current Accounts		Credited
Account	Value (US$)	Account	Value (US$)	Value (US$)
WBS Section 1 HW	1,000.00	WBS Section 1 HW	0.00	1,000.00
WBS Section 1 Tech	1,000.00	WBS Section 1 Tech	0.00	1,000.00
WBS Section 2 HW	500.00	WBS Section 2 HW	0.00	500.00
WBS Section 2 Tech	500.00	WBS Section 2 Tech	0.00	500.00
WBS Section 1 HW Reserve	500.00	WBS Section 1 HW Reserve	0.00	500.00
WBS Section 1 Tech Reserve	500.00	WBS Section 1 Tech Reserve	0.00	500.00
WBS Section 2 HW Reserve	250.00	WBS Section 2 HW Reserve	201.03	48.97
WBS Section 2 Tech Reserve	250.00	WBS Section 2 Tech Reserve	0.00	250.00
Work Package Cost	4,500.00	***Work Package Cost***	201.03	4,298.97
Contingency Reserve	500.00	Contingency Reserve	260.25	239.75
Control Cost Baseline	5,000.00	***Control Cost Baseline***	461.28	4,538.72
Management Reserve	1,018.18	Management Reserve	1,018.18	0.00
Project Budget	**6,018.18**	***Project Budget***	**1,479.46**	**4,538.72**
Credited Per Team by Phase				
Team	**Phase**	**Value (US$)**		
HW	1	1564.93		
Tech	1	1000.85		
HW	2	548.97		
Tech	2	1423.97		
Total Amount Credited		**4538.72**		

FINAL SYSTEM CDR

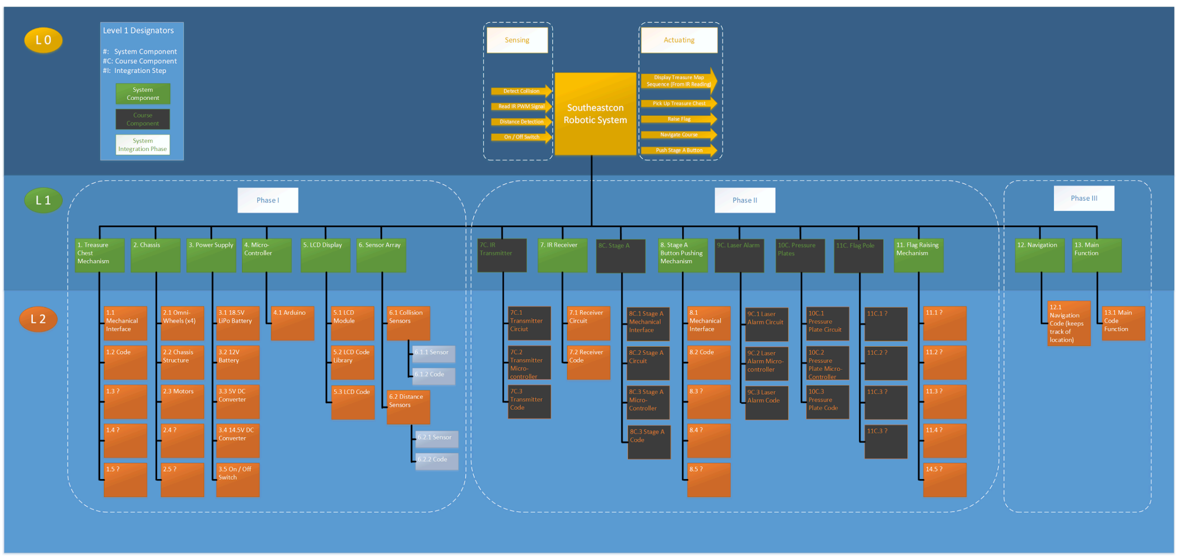
2018 ECE Capstone

System Overview

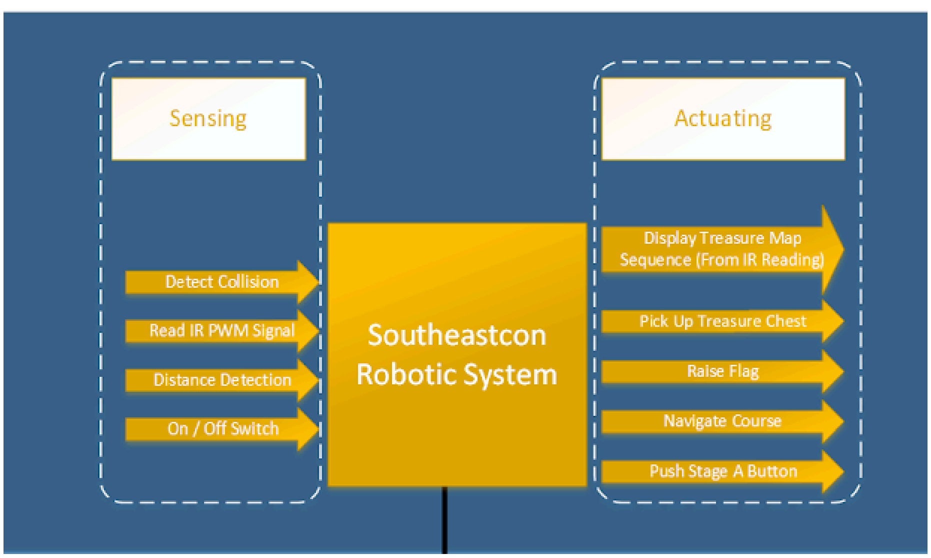
1. Purpose:

The purpose of the System CDR is to have a comprehensive document that will allow our team to construct/repair/replace any and all components in a timely manner. It will also be used as a template for our final presentation and a reference for all of the calculations, specifications, and schematics that went into designing and building the robot.

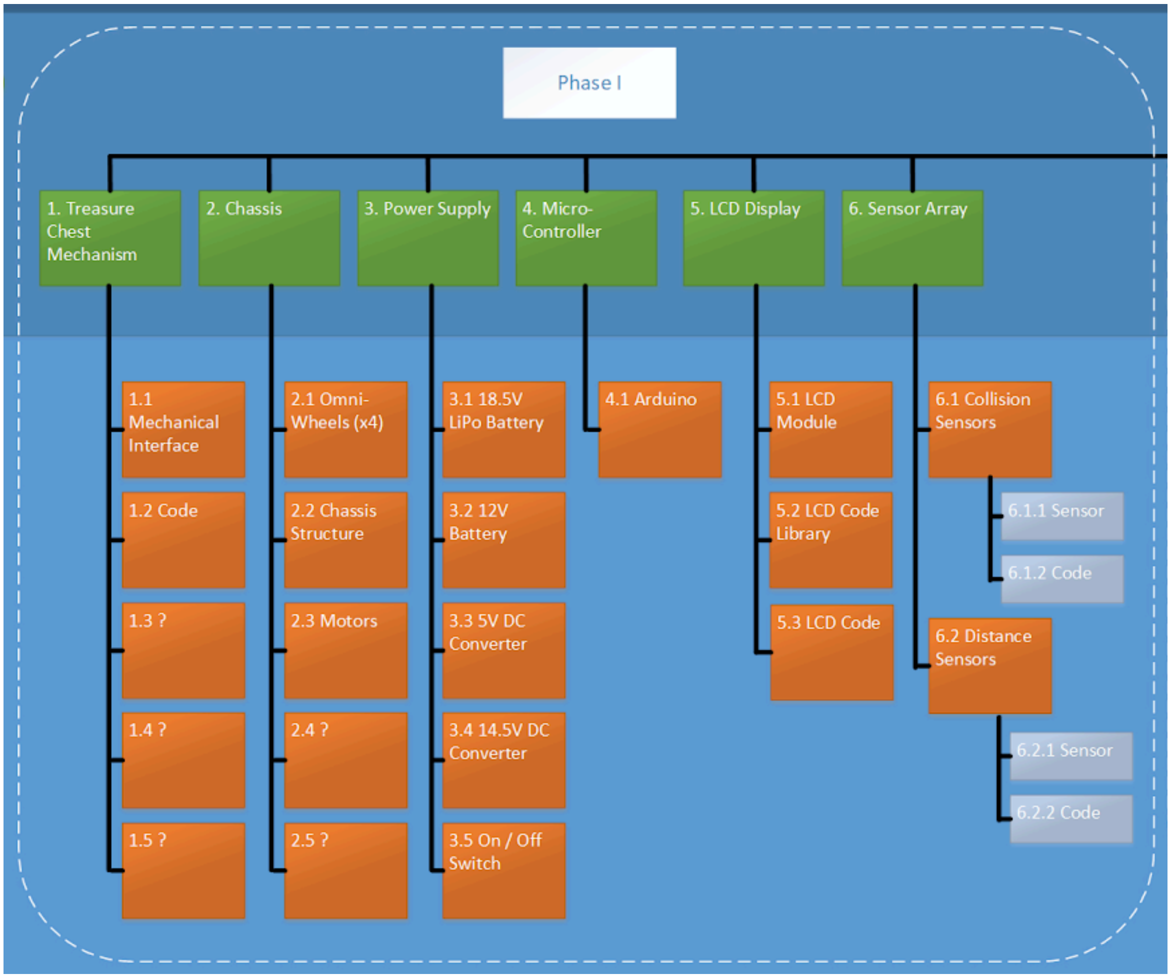
2. Functional Decomposition



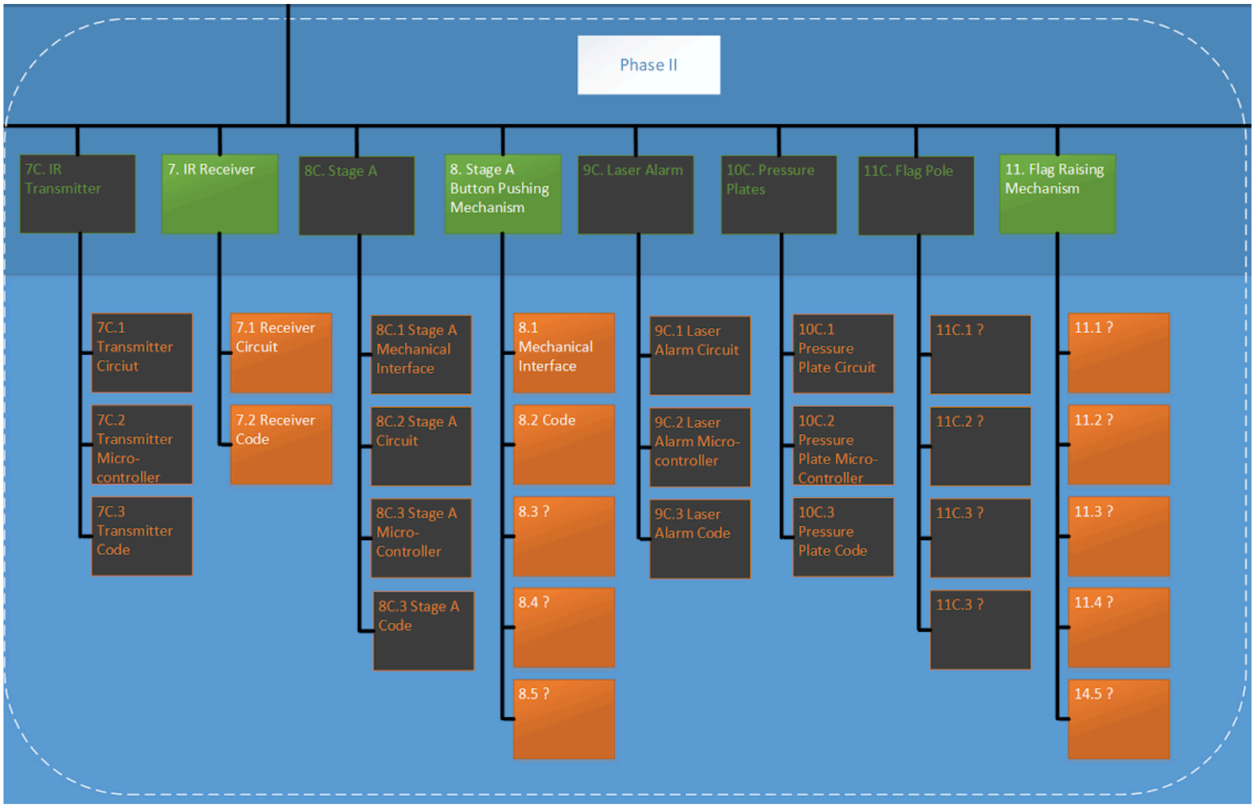
Level 0



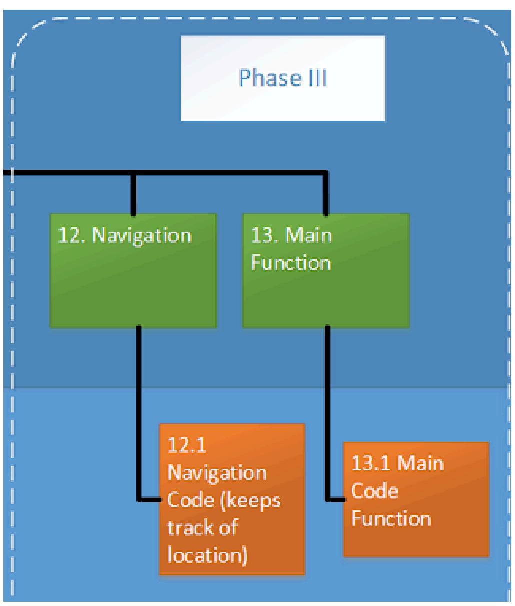
Phase I


Phase I
1. Treasure Chest Mechanism
2. Chassis
3. Power Supply
4. Micro-Controller
5. LCD Display
6. Sensor Array
1.1 Mechanical Interface
1.2 Code
1.3 ?
1.4 ?
1.5 ?
2.1 Omni-Wheels (x4)
2.2 Chassis Structure
2.3 Motors
2.4 ?
2.5 ?
3.1 18.5V LiPo Battery
3.2 12V Battery
3.3 5V DC Converter
3.4 14.5V DC Converter
3.5 On / Off Switch
4.1 Arduino
5.1 LCD Module
5.2 LCD Code Library
5.3 LCD Code
6.1 Collision Sensors
6.1.1 Sensor
6.1.2 Code
6.2 Distance Sensors
6.2.1 Sensor
6.2.2 Code

Phase II



Phase III



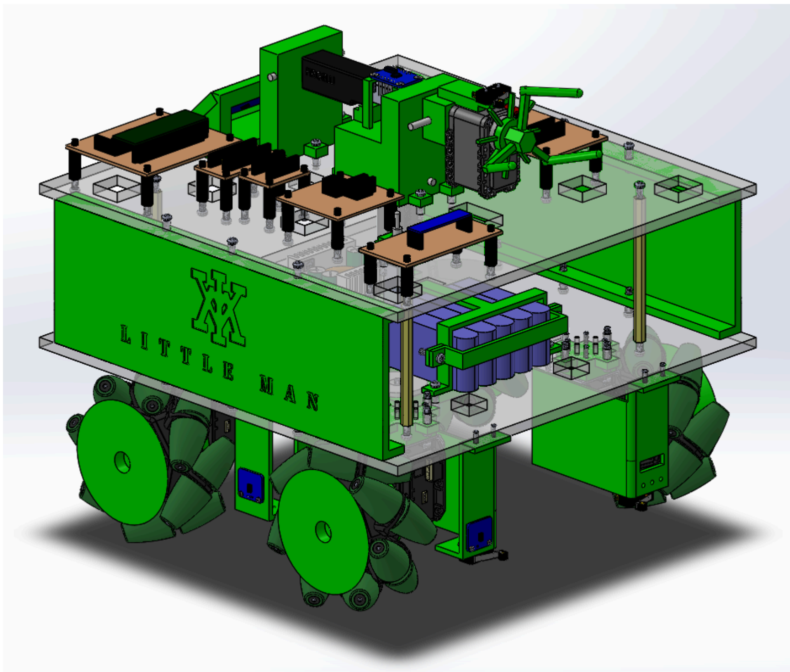
3. Requirements

The system must meet all the requirements set forth in the Engineering Requirements Document and the most current version of the IEEE competition rules.

4. Functionality Test

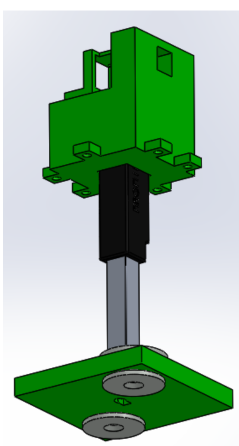
The system will be run through the course 100 times. Out of these runs, it may fail at any stage of the competition once. This confirms a reliability of 99% which is in accordance with the 98.5% reliability specified in the Engineering Requirements

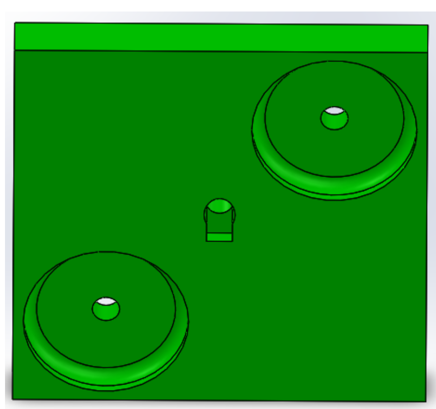
5. Main Diagram/Schematics

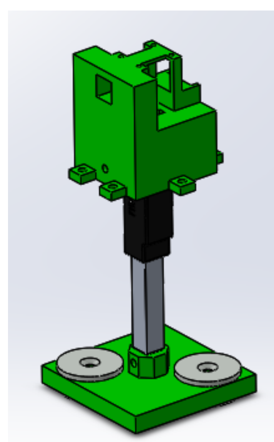


1.1 Treasure Chest Raise Mechanism

Visuals







Operational Description

First, the robot will move over top of the treasure chest. Once the gap in the bottom of the robot is centered with the treasure chest, the Actuonix L16-R linear actuator will extend from its stowed position, lowering the platform with neodymium magnets until it is magnetically secured to the treasure chest. After the chest is attached, the actuator will retract the chest until it is off the ground and centralized inside the robot chassis. After the chest is retracted, the robot will continue the course.

Material List

- Top Chassis to Actuator mount (3D printed)
- Magnet platform (3D printed)
- Neodymium magnets
- L16-R Linear Actuator

Functionality Test

This mechanism must reliably lower the magnet platform until the neodymium magnets are secured to the top of the treasure chest. After the treasure chest is attached, the platform/secured treasure chest will be retracted to the appropriate height within the robot chassis.

1.2 Chassis

Operational Description

The Chassis is the physical backbone of the robot which all of our sensors, actuators, and motors will be mounted on. It must be rugged enough to support the weight of the equipment, but designed precisely enough that the navigation system will be reliable. The chassis has two levels which accommodate the treasure chest and flag raise mechanisms, sensors, circuitry, LCD, and other components.

Material List

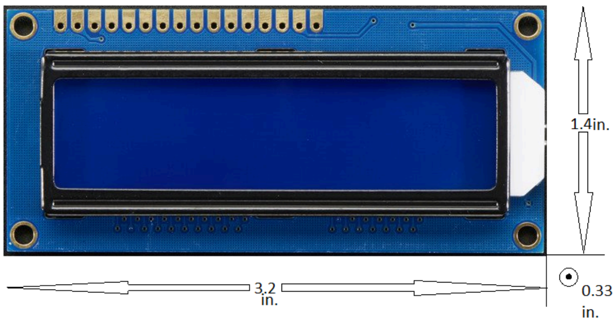
- Teensy 3.6
- RX-28
- AX-12A
- Breadboards/PCB
- 4 bump sensors
- 4 TOF sensors
- Voltage Regulators
- Male/Female plugs for the Battery
- IR Receiver
- 2 Linear Actuators

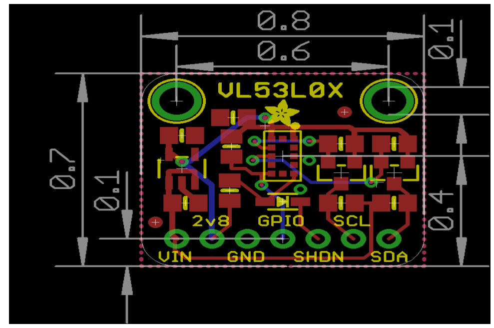
Functionality Test

Every Component has been tested individually and functions 100% properly. Except for IR but we do not have those in yet and they have been modeled with regular photodiodes.

- TOF sensors Work
- Bumps Work
- Movement (Strafing, Forward, Back)

- IR RX Reading Signal
- LCD Displaying





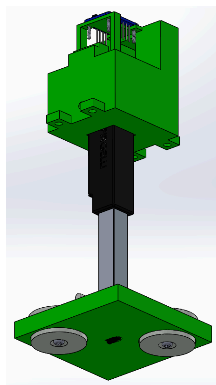
PURPOSE	LEFT SIDE	RIGHT SIDE	PURPOSE
GND	**GND**	**Vin (3.6-6.0 V)**	5V IN
	0	**Analog GND**	GND
	1	**3.3V (250mA max)**	Power Supply for TOF
Direction Pin RX-28	**2**	**23**	
Direction Pin AX-12A	**3**	**22**	
	4	**21**	
	5	**20**	
	6	**19**	(SCL0) T.O.F.
RX-28 RX3	**7**	**18**	(SDA0) T.O.F.
RX-28 TX3	**8**	**17**	
AX-12A RX2	**9**	**16**	
AX-12A TX2	**10**	**15**	(A1) BUMP 2 IN
	11	**14**	(A0) BUMP 1 IN
	12	**13**	
	3.3V	**GND**	GND
LCD RS	**24**	**A22**	
LCD EN	**25**	**A21**	
LCD DB4	**26**	**39**	
LCD DB5	**27**	**38**	Shutdown TOF
LCD DB6	**28**	**37**	Shutdown TOF
LCD DB7	**29**	**36**	Shutdown TOF

	30	**35**	Shutdown TOF
IR Rx	**31**	**34**	
	32	**33**	

Teensy 3.6 Pinout

1.4 IR Receiver

Visual



Operational Description

IR Emitter: This part is not part of the robot; it is a stage element. The emitter will send out a sync bit start burst signal for 9 ms, then a 4.5 ms space, followed by 8 bits that contain the message, and then an end burst. The first 5 bits of the message will be sent as Logical 0s, and then the last 3 bits will define the coordinates for the route that needs to be followed. Logical 0s have a length of 1.12 ms and Logical 1s have a length of 2.25 ms. The message will be sent every 5 seconds.

IR Receiver: This device will descend, attached to the bottom of the actuator platform, to a maximum range of no more than 2 inches (maximum effective range). Our software reads the times of each pulse and by reading the pulse we will know where in the message the signal is and whether it is logical 1 or 0. Once we find out the last 3 bits, we will store it in an array and from there, the receiver will communicate with the Teensy to write the array to the LCD Display.

Dependencies

- Treasure Chest Raise Mechanism
- IR Transmitter

Material List

- Standard LCD 16x2
- IR Receiver Sensor
- Teensy 3.6

Functionality Test

The IR Receiver must properly read the signal message sent and divide it into a three bit code. Once the code is read by the IR Receiver it must display the proper code on the LCD screen.

1.6 Stage A/C Button Press Mechanism

Operational Description

The stage A/C button press mechanism consists of two 3D printed panels mounted on both sides of the robot. The robot will strafe left or right until the Time of Flight sensors measure a predetermined distance from the walls of the board which will ensure the button has been pressed. In future updates, a verification system will be implemented which will utilize a photodiode to ensure that the confirmation LED has illuminated on the side of the contest board.

Dependencies

- Chassis
- Time of Flight Sensors

Material List

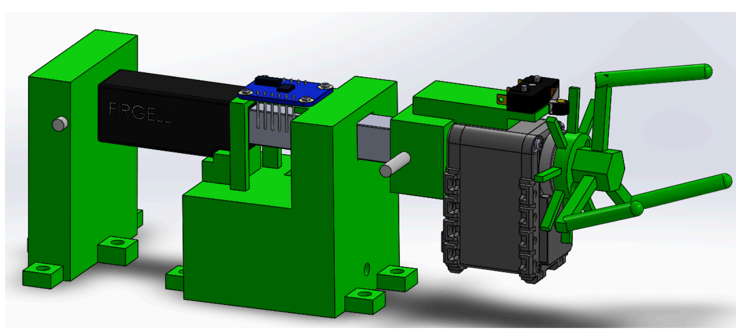
- Stage A/C Press Panels (3D Printed)

Functionality Test

We will test the functionality of the mechanism by repeatedly navigating the robot from the center of the board to either of the press buttons. The panels and mounting hardware will be observed for their integrity and security throughout the testing process.

1.10 Flag Raise Mechanism

Visuals



Operational Description

The Flag Raise Mechanism utilizes the same linear actuator as the treasure chest raise mechanism. An AX-12 motor and 3D printed, three pronged arm are attached to the end of the actuator. When the robot is properly aligned in front of the Captain's Wheel, the linear actuator will extend until the three pronged arm is outside the chassis and in between the spokes of the Captain's Wheel. The AX-12 / three pronged arm will then rotate until the flag is raised.

Dependencies

- Chassis
- Navigation

- Linear actuator / AX-12 Software

Material List

- AX-12
- L-16 Actuator
- L-16 Actuator to Chassis mounts (3D printed)
- Three pronged arm (3D printed)

Functionality Test

The robot must be able to properly center itself on the contest board to align with the "Captain's Wheel" which will raise the flag. After the robot is properly centered, the arm must extend and the AX-12 motor must turn the specified number of rotations reliably. The three pronged arm must make proper contact with the Captain's Wheel in order to turn it reliably.

1.11 Power Supply

Functional Decomposition

The total power needed for the robot can be derived from the power needed for each subsystem during operation. The power supply must be designed to comply with the size and weight constraints of the robot chassis.

- Level 0: Chassis Design
- Level 1: a) Size of Chassis Top and Bottom "Decks"
  b) Dynamixel Current draw
  c) Component Current tolerance

Operational Description

The purpose of the power breakdown is to ensure that at any moment during the competition the robot have a sufficient amount of power for operation. This will ensure consistent and low noise which is crucial to testing and development of our system. We have decided to work with more power than required and regulate down to the appropriate level to ensure consistency and low noise within the system.

Dependencies

- Chassis Size / Weight Limitations
- Sensors, wheels, electronic components

Material List

- 3.3,5,6,10/12 Voltage regulators
- 7.2 V/ 22.2 V LiPo Battery

Functionality Test

- Ensure proper power is delivered to the appropriate power bus
- Ensure the consistency of power delivered throughout system

Additional Information

Qty	Description	Operating Voltage (V)	Com Voltage Max (V)	Avg Current Draw (mA)	Max Current Draw (mA)	AVG Power(mW)	Max Power (mW)
Sensors							
5	Dynamixel Motor	18.50	5.00	313.05	699.52	5791.4	12941.2
2	TOF VL53L0X	3.30		19.00	40.00	62.7	132.0
2	TOF VL6180	3.30		1.70	250.00	5.6	825.0
1	Teesny 3.6	3.30	3.30	91.00	300.00	300.3	990.0
1	Adafruit LCD	5.00	5.20	35.00	40.00	175.0	200.0
2	Bump Sensors	NA	NA	NA	NA		
2	Linear Actuator	6.00	5.00	150.00	500.00	900.0	3000.0
1	IR Rx	3.30	3.30	0.00	0.00	0.0	0.0
Statistics:		**Power (mW)**					
	Max Power	18,088.18					
	Avg Power	7,235.0					

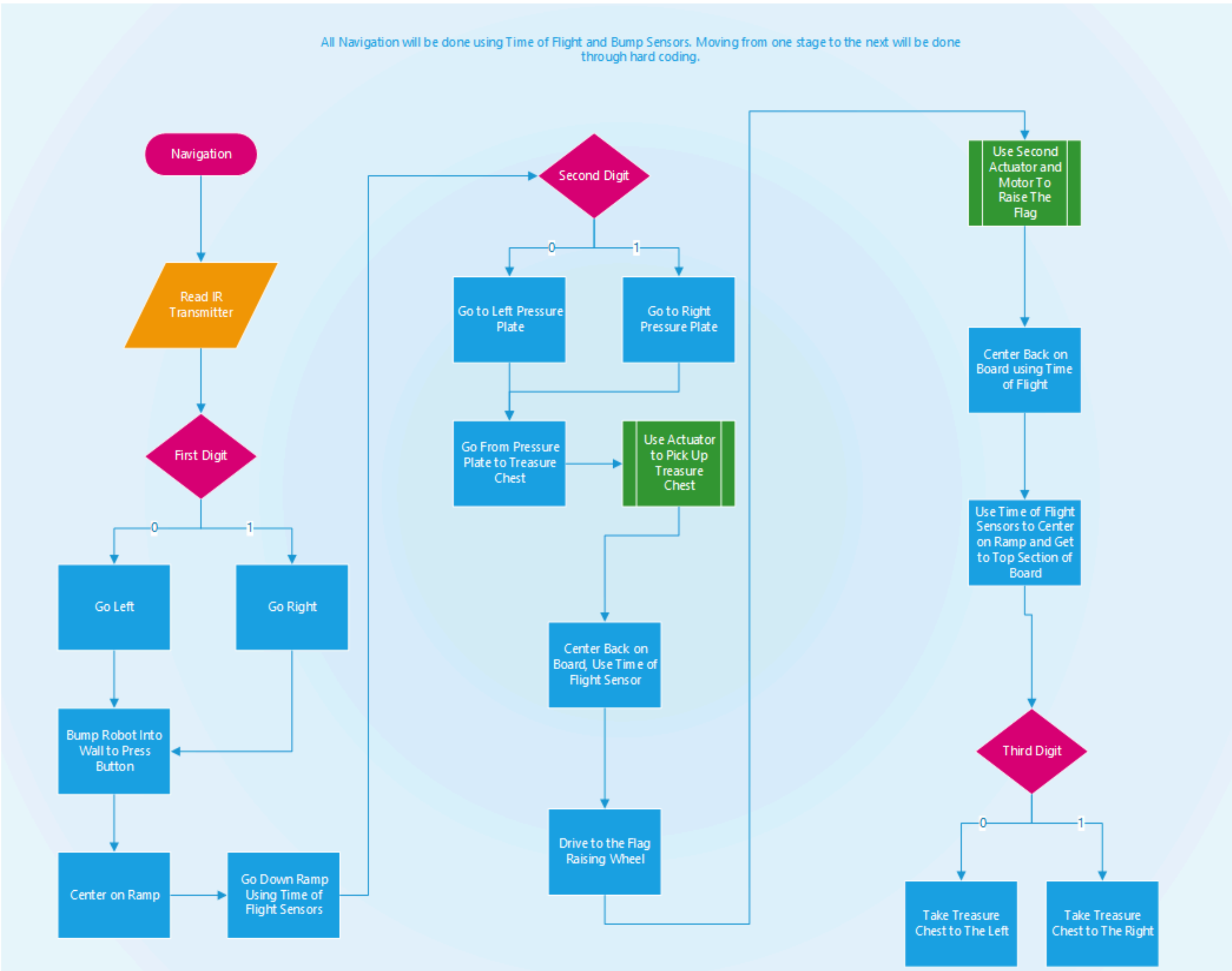
Regulators:

3.3 V: https://www.adafruit.com/product/2236
5 V:
https://www.jameco.com/z/L7806CV-STMicroelectronics-Standard-Regulator-6-Volt-1-5-Amp-3-Pin-3-Tab-TO-220-Tube_924668.html
6 V: https://www.adafruit.com/product/2166
10V:
https://www.jameco.com/z/L7912CV-STMicroelectronics-Standard-Regulator-12-Volt-1-5-Amp-3-Pin-3-Tab-TO-220_889524.html
12 V:
https://www.jameco.com/z/LM2940T-10-0-NOPB-Texas-Instruments-Low-Dropout-Regulator-10-Volt-1A-3-Pin-3-Tab-TO-220-Rail_837441.html

Power Analytics Link:
https://docs.google.com/spreadsheets/d/1MRJOkO2pHgWeYnhSWhsD2hVa_Do2689yxK7091xumYs/edit#gid=0

1.12 and 1.13 Navigation and Main Function

Flowchart



2018 CAPSTONE API

Black means we can proceed with the current version of it
Red means we need to make it adaptable (be able to use the map to tell you which way to go)
Blue means that overloaded functions exist to incorporate the int Map[3] from the IR

//--
//--
NAMING CONVENTION FOR ALL OF TOF READINGS

- **int R = 0;**
- **int L = 1;**
- **int BR = 2;**
- **int BL = 3;**
- **int FR = 4;**
- **int FL = 5;**
- **int TFR = 6;**
- **int TFL = 7;**

//--
//--

int CenterUPTREASURE(int x)

- Centers the robot Orthogonal to the wall so that you can move onto the Chest
- Input is irrelevant
- **Returns 0** if the function is not complete.
- **Returns 1** if function is complete.

int MoveCenterB()

- Centers the Robot in the center of the board by moving left from push button in Stage B
- The **TOF[R] < 435 is up for some tweaking** but it seems to work for now.
- Needs to be setup for dual use.
- **Returns -1** if the function is not complete.
- **Returns 1** if function is complete
- Overloaded function is capable of reading in Map[] values and selecting which direction to move if necessary

int APPROACHB()

- Once finished going down the ramp the robot will then proceed straight to the correct position to hit stage B push buttons.
- **No input**
- **Returns -1** if the function is not complete.
- **Returns 1** if function is complete

int SLAMB(int x)

- Completes the stage B push button by slamming into the correct side
- **Input 0 for right side**.
- **Input 1 for left side**.
- **TOFVal[R/L] > 85 may need some tweaking.** Seems to work for now.
- **Returns -1** if the function is not complete.
- **Returns 1** if function is complete

void MOTORSETUP()

- **Begins Serial3 communication with Teensy**
- Also sets all of the motors to continuous mode.

void FORWARD(int x)

- Moves the Robot Forward
- **Input == SPEED**

- **Speed values can range from 0 < speed < 1023 (IF THE VALUE IS NOT IN THIS RANGE THE MOTORS WILL NOT RESPOND)**
- Nothing to Return

void BACKWARD(int x)

- Moves the Robot BACKWARD
- **Input == SPEED**
- **Speed values can range from 0 < speed < 1023 (IF THE VALUE IS NOT IN THIS RANGE THE MOTORS WILL NOT RESPOND)**
- Nothing to Return

int pushStageA(int x)

- Reads the TOFs every iteration.
- **Input 0 to push left button**
- **Input 1 to push right button**
- **Returns -1** if the function is not complete.
- **Returns 1** if function is complete

void STOP()

- Stops Robot
- **No Input**
- **No Output**

int CenterUPB(int x)

- Centers the robot Orthogonal to the wall so that you can move into the center of the board after you hit.
- **Returns -1** if the function is not complete.
- **Returns 1** if function is complete
- Overloaded function exists in alternative build

int CenterUPA()

- Centers for ramp success after moving to the center of the board
- No Input
- **Returns 0** if the function is not complete.
- **Returns 1** if function is complete

void TURNLEFT(int x)

- Turns Robot Left
- **Input == SPEED**
- **Speed values can range from 0 < speed < 1023 (IF THE VALUE IS NOT IN THIS RANGE THE MOTORS WILL NOT RESPOND)**
- Nothing to Return

void TURNRIGHT(int x)

- Turns Robot Right
- **Input == SPEED**
- **Speed values can range from 0 < speed < 1023 (IF THE VALUE IS NOT IN THIS RANGE THE MOTORS WILL NOT RESPOND)**
- Nothing to Return

int Center(int x)

- **Uses Control system**
- Centers after the robot has pushed the button
- **Input 0** Centers from right side to center
- **Input 1** Centers from Left side to center
- **Returns 0** if the function is not complete.
- **Returns 1** if function is complete

void STRAIFLC(int x, int D)

- **Control system** for moving left and right during stage A
- **Input X** == Speed of movement
- **Input D** == Distance away from the back wall
- **No Return**

void STRAIFRC(int x, int d)

- **Control system** for moving left and right during stage A
- **Input X** == Speed of movement
- **Input D** == Distance away from the back wall
- **No Return**

void Sensors_setup()

- Sets up all of the sensors
- **DO NOT EDIT!! PLEASE FOR THE LOVE OF GOD**

void ReadTOFB()

- **If you aren't getting readings that are correct make sure you are calling proper funciton**
- Reads 4 TOF Sensors:
  - **Abs right**
  - **Abs left**
  - **Back Right**
  - **Back Left**
- Assigns them to the appropriate array value
- Use naming convention to call correct sensor

void ReadTOFF()

- **If you aren't getting readings that are correct make sure you are calling proper funciton**
- Reads 4 TOF Sensors:
  - **Abs right**
  - **Abs left**
  - **Front Bottom Right**
  - **Front Bottom Left**
- Assigns them to the appropriate array value
- Use naming convention to call correct sensor

void ReadTOFFT()

- **If you aren't getting readings that are correct make sure you are calling proper funciton**
- Reads 4 TOF Sensors:
  - **Abs right**
  - **Abs left**
  - **Front Top Right**
  - **Front Top Left**
- Assigns them to the appropriate array value
- Use naming convention to call correct sensor

int LeftTOFCHECK()
int RightTOFCHECK()
int BackRTOFCHECK()
int BackLTOFCHECK()
int FrontRTOFCHECK()
int FrontLTOFCHECK()
int TopFrontRTOFCHECK()
int TopFrontLTOFCHECK()

- All of these functions acquire the input from the respective TOF void TOFSETUP()

- **DO NOT EDIT!! PLEASE FOR THE LOVE OF GOD**
- TOF SETUP
- This is where the pinmode is controlled along with address of each TOF
- Also Timeouts are set at the end of this function
- **THE CODE FOR THE VLOX680 IS LOCATED AT THE END AS WELL**

void TOE(int x)

- Currently has lcd print stuff on it so you can see the outputs of the TOFs on the lcd
- Throw away code that will not be used later
- Debugging only

float range(float x)

- Used for actuators.
- **Not exactly sure what this does**

void clearMem()

- Sets all the memory of the control system back to 0
- Everything that needs to implemented with this function has been done

void STRAIFL(int x)

- Just strafes to the left. **Doesn't have any sort of control system**
- **Input == speed**
- **No Output**

void STRAIFR(int x)

- Just strafes to the Right. **Doesn't have any sort of control system**
- **Input == speed**
- **No Output**

int SwitchDet()

- Makes sure that the treasure chest is on the middle for magnets to pick it up. Once in the middle it will pick up the chest.
- **No input**
- **Returns 1 or -1**

int MoveCenterRAMP();

-Center Robot before moving up ramp for stage C

int APPROACHC()

-Has robot move across board towards ramp after raising flag

Int APPROACHConRAMP()

-Has robot move to center of stage C after moving onto the deck

Int DEBUG(String Thing1, String Thing2)

-Prints out current stage, current action it should be performing, along with all 6 TOF values to see what values it's reading in)

-Is mostly for loops, since single use functions would only call it once.

-MIGHT BE MODIFIED TO INCLUDE ENCODER VALUES FROM DYNAMIXEL WHEEL TURN.

Void currentArray()

-Prints out how much each

Schedule Management Plan

Vincent Smith-DiLeo, Jake Banigan, Ryan Aranzamendez
Help Received: None

Schedule Model Development

The schedule model is based on the Critical Path Method (CPM) and is constructed and updated using the Microsoft Projects software suite. The schedule will be modified only by the project manager (PM) and a scheduler designated by the PM. The schedule and Work Breakdown Structure (WBS) will be developed, updated, and published in the form of a Gantt chart on the team Trello page.

Level of Accuracy

The duration estimates of scheduled tasks will be given, at minimum, a 10% estimated tolerance for schedule delays.

Units of Measure

Project objectives will be scheduled in days as the lowest unit of time. The lowest unit of cost will be measured in cents.

Project Schedule Model Maintenance

Scheduling will be conducted by the PM, scheduler, and team leads. Scheduling will be conducted prior to team meetings at a minimum of once per week.

Control Thresholds

The sum of schedule delays must not cause the second phase of the project to extend beyond 13 DEC 17. Individual tasks must not exceed their allotted time constraint by more than 50 percent their original allotted time.

Reporting Formats

Schedule Reports will be given at the start of each meeting. In depth schedule reports will be presented as part of the monthly Project Management Review (PMR).

Cost Management Plan

Alan Lloyd and Yousef Malik
Help Received: None

Cost Planning

A budget will be assigned to the hardware and tech teams during the completion of tasks 1 and 2 of the WBS (also referred to as Phases 1 and 2). The hardware and tech teams will each be assigned an initial budget of $1,000 during the completion of Phase 1 and a budget of $500 during the completion of Phase 2. These budgets will be reviewed and may be changed by the PM based on the immediate needs of the project. In the case that the current budget is insufficient to produce deliverables of a quality consistent with project goals, the PM will request additional funds, which will be supplied (as specified under section VII of the Project Charter).

Structure and Accounts

Units of Measurement: Cost will be tracked in US Dollars (US$). Work hours will not be taken into consideration when calculating cost
Level of Precision: Cost will be tracked to the accuracy of the cent (US$0.01)
Level of Accuracy: Cost estimates will have a ± 10% accepted variance
Control Thresholds: If any team is projected to go over budget, the team lead will work with the PM to assess whether the design should be modified to stay within budget or if additional funds should be requested for that team's budget. If additional reserve funds are unused after the completion of the WBS section one, they will be split equally among the contingent funds for WBS section two budgets (see Table 1 and Figure 1)
Reporting Formats: Costs will be tracked in the Accounts page of the BOM & Finance google sheets document. A cost report will be submitted after the completion of sections one and two of the WBS. Cost reports will also be covered in the monthly PMR meetings

Controlling Cost

Procurement will be tracked in the project BOM & Finance document. This electronic, cloud based spreadsheet will track purchases and calculate the remaining project budget.

Table 1. Project Accounts

Account	Value (US$)
WBS Section 1 HW Cost Estimate:	$1,000
WBS Section 1 Tech Cost Estimate	$1,000
WBS Section 2 HW Cost Estimate	$500
WBS Section 2 Tech Cost Estimate	$500
WBS Section 1 HW Reserve Estimate	$500
WBS Section 1 Tech Reserve Estimate	$500
WBS Section 2 HW Reserve Estimate	$250
WBS Section 2 Tech Reserve Estimate	$250
Work Package Cost Estimate	$4,500
Contingency Reserve	$500
Control Cost Baseline	$5,000
Management Reserve	$1,018
Project Budget	$6,018

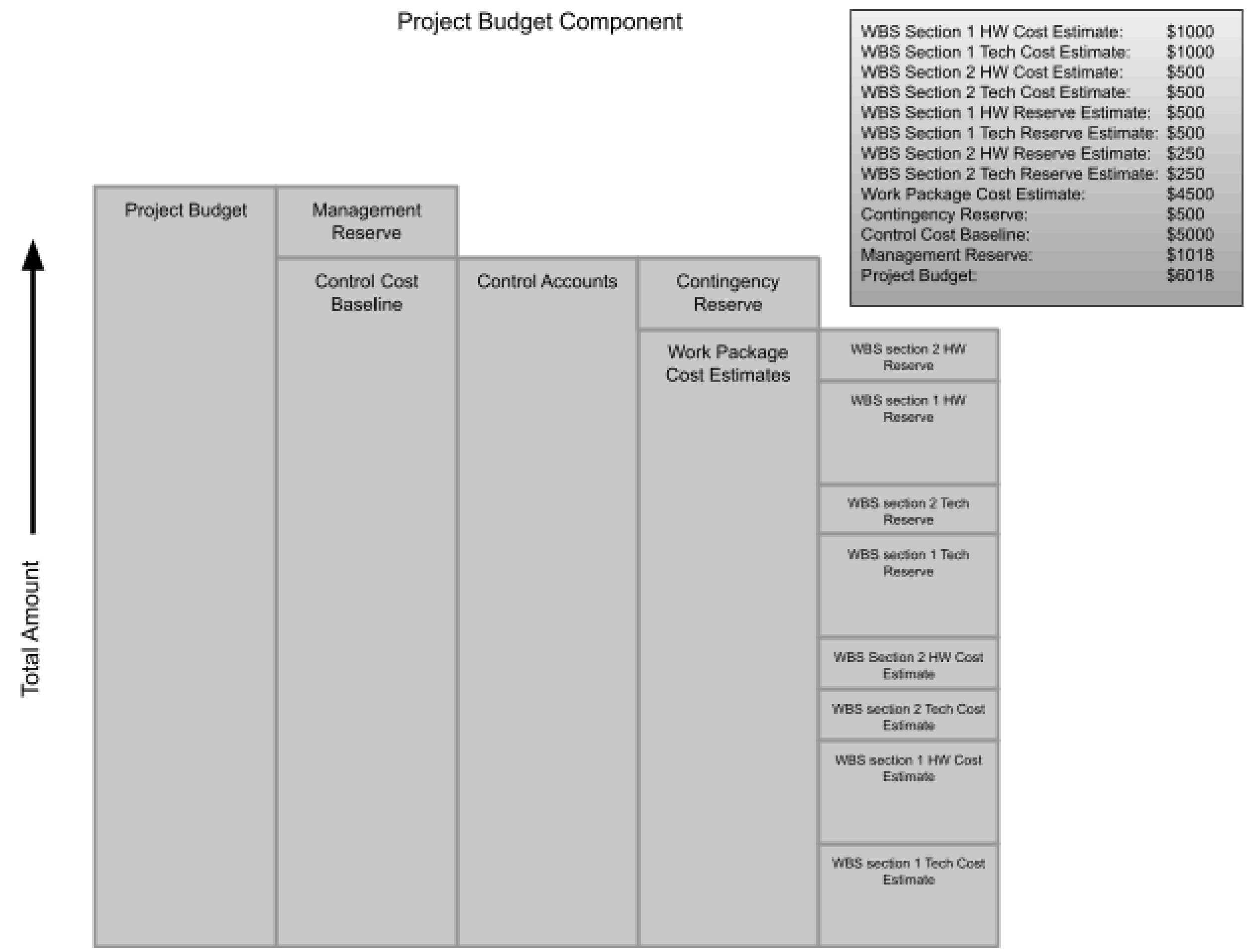


Figure 1. Project Accounts Graph

Project Accounts

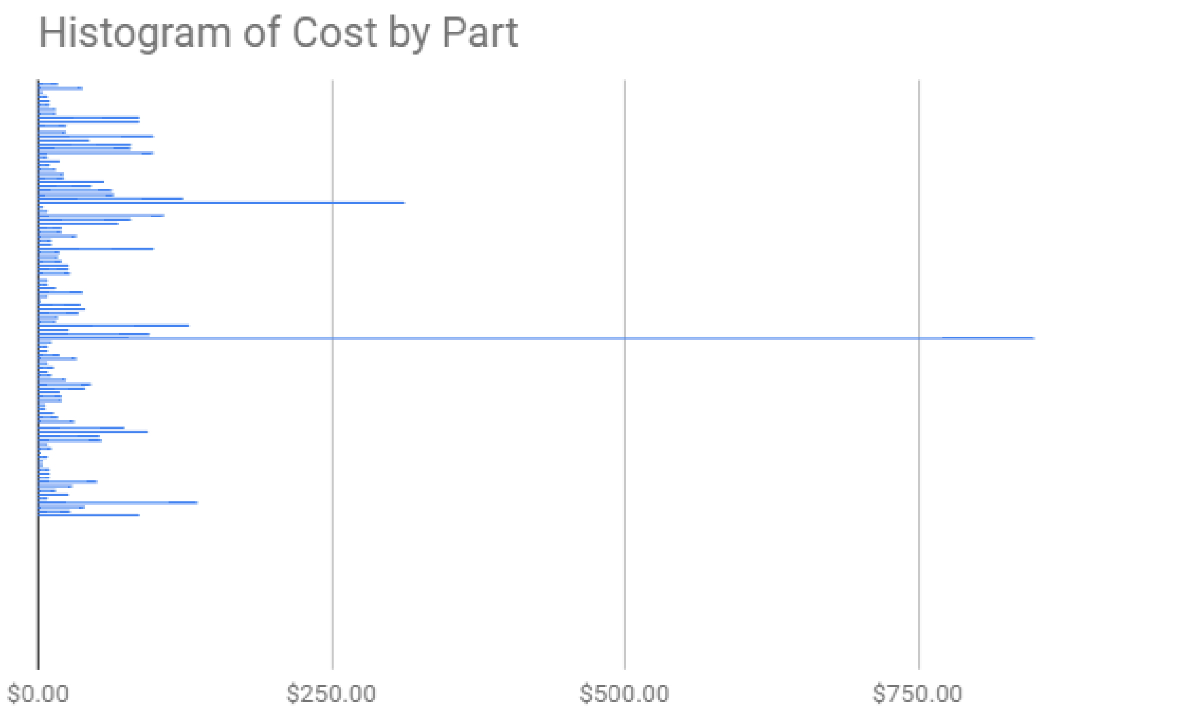
Initial Values		**Current Accounts**		**Credited**
Account	**Value (US$)**	**Account**	**Value (US$)**	**Value (US$)**
WBS Section 1 HW	1,000.00	WBS Section 1 HW	0.00	1,000.00
WBS Section 1 Tech	1,000.00	WBS Section 1 Tech	0.00	1,000.00
WBS Section 2 HW	500.00	WBS Section 2 HW	0.00	500.00
WBS Section 2 Tech	500.00	WBS Section 2 Tech	0.00	500.00

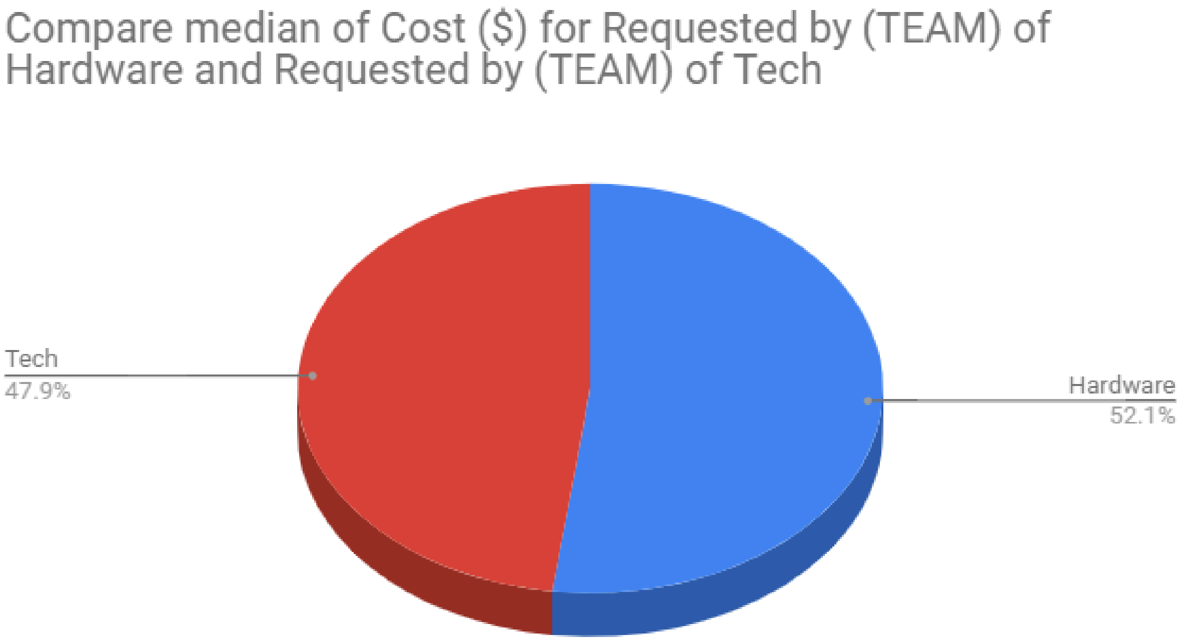
WBS Section 1 HW Reserve	500.00	WBS Section 1 HW Reserve	0.00	500.00
WBS Section 1 Tech Reserve	500.00	WBS Section 1 Tech Reserve	0.00	500.00
WBS Section 2 HW Reserve	250.00	WBS Section 2 HW Reserve	201.03	48.97
WBS Section 2 Tech Reserve	250.00	WBS Section 2 Tech Reserve	0.00	250.00
Work Package Cost	4,500.00	***Work Package Cost***	201.03	4,298.97
Contingency Reserve	500.00	Contingency Reserve	260.25	239.75
Control Cost Baseline	5,000.00	***Control Cost Baseline***	461.28	4,538.72
Management Reserve	1,018.18	Management Reserve	1,018.18	0.00
Project Budget	**6,018.18**	***Project Budget***	**1,479.46**	**4,538.72**
Credited Per Team by Phase				
Team	**Phase**	**Value (US$)**		
HW	1	1564.93		
Tech	1	1000.85		
HW	2	548.97		
Tech	2	1423.97		
Total Amount Credited		**4538.72**		

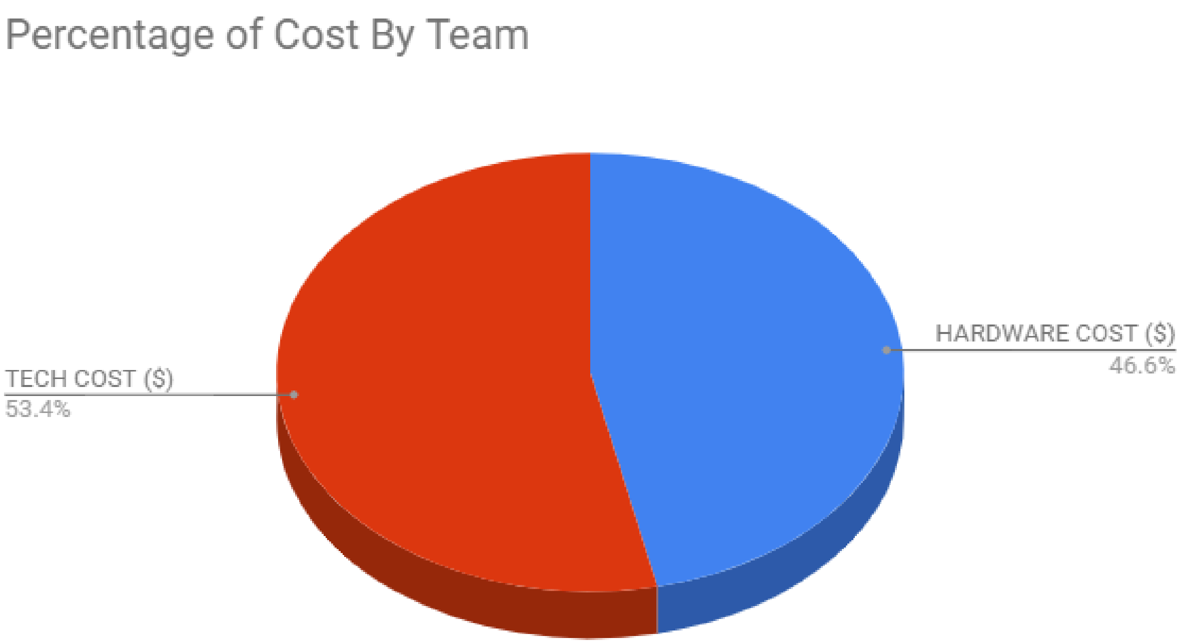
Purchasing Plan

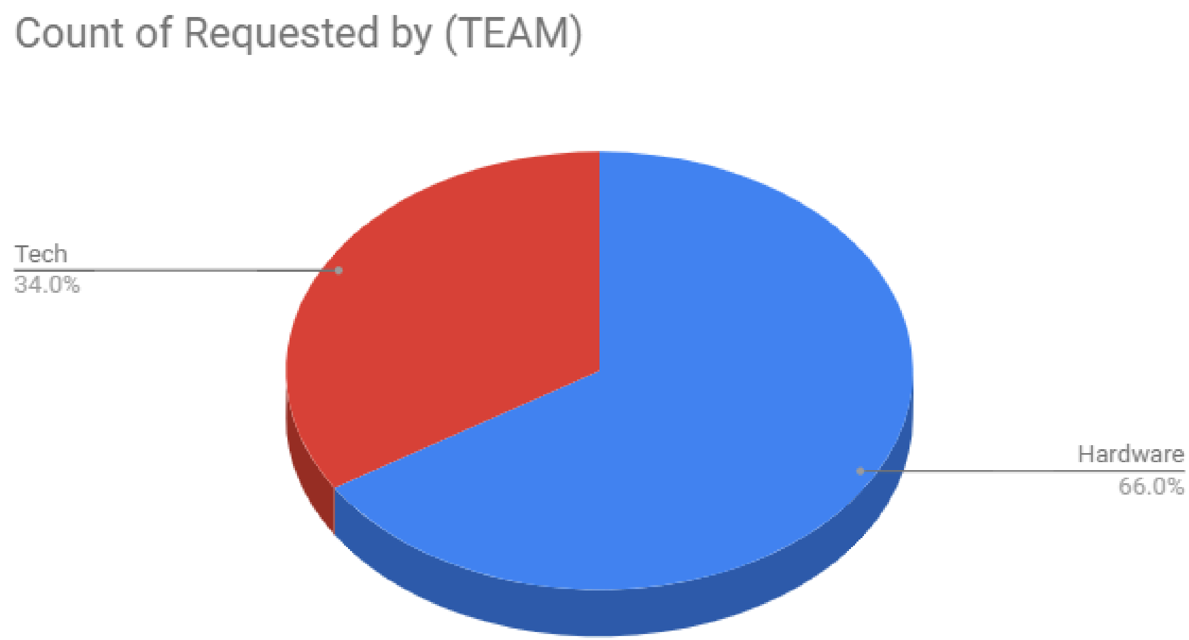
All purchases were added to the bill of Materials. Charts were used to analyze costs by team, the relative costs of purchased item, among other aspects procurement. Teams members were also made aware of the global implications of purchasing project materials.

Purchasing Metrics




Compare median of Cost ($) for Requested by (TEAM) of Hardware and Requested by (TEAM) of Tech
Tech
47.9%
Hardware
52.1%


Percentage of Cost By Team
TECH COST ($)
53.4%
HARDWARE COST ($)
46.6%



Globalization Analysis Reports

The capstone project used parts from a number of vendors and international manufacturers. Each member of the capstone team wrote a personal reflection on how globalization has made this project possible and how it continues impact the decision making process when designing and integrating complex systems.

Yousef Malik, Ryan Arazamendez, Buka Anwah
There are engineers and there is engineering all over the world. Engineering had made revolutions and advancements in this lifetime where people from different areas of the world have experienced. Engineers have developed to create a system or subsystems for a system from engineers, products, and ideas all over the world, to make an invention or innovation to revolutionize the world. This system that the senior capstone team constructed was not a revolutionary system, but there were aspects of how there were impacts of globalization on engineering.

Solidworks played a huge part in the design for the senior capstone project. The impact Solidworks had of the globalization on engineering is that Solidworks is developed by Dassault Systèmes, a European software company headquartered in France. Using this software, the senior capstone design team were able to 3D model subsystems of the robot to display a virtual assembly and even 3D print several components for the robot using 3D modeling on Solidworks. The Dynamixel Motors that were used on the robot were imported from South Korea from a company called Robotis. These motors, especially the AX-12, from Robotis are used all over the world for robotics.

Not only is the senior capstone project have an impact of globalization of engineering, the IEEE Southeast Conference had many engineers, scholars, professionals, professors, etc. from all over the world. There were different ideas and different studies from various areas of the world which show how much of an impact there is of globalization on engineering

Wyatt Raymond, Edward Olbrych, Joseph Brucahlski

The Globalization of Engineering has allowed designers to rapidly create and prototype ideas in ways and speeds that would not have thought to be possible only decades ago. With the advent of the internet, Engineers now have almost any possible resource available to them granted they have the required budget. Engineers no longer must focus on reinventing the wheel for if the solution is out there to purchase then you might as well buy instead of designing it. Because so many of these low-level components are readily available, engineers can now spend much more time focusing on larger more complicated and big picture-oriented projects.

Not only are physical components readily available to designers on the internet, but also thousands of pages of software and documentation exist as well. Much like having prefabricated parts available, existing code and documentation allow designers to take these base level concepts and either quickly modify them to suit their needs or they can combine existing to concepts to create something entirely revolutionary. Innovation has never been accessible like it is today and the pace at which it is occurring is only exponentially increasing.

Zhichen Zang, Jake Banagin

Ever since the technological revolution started, the everyday life for countries with resources has been getting easier. Incorporation of technology into daily lives and communication between people at long distances has become very routine. Before the advanced technology came to life people had to travel for weeks and sometimes months to trade with other countries, to acquire their goods. Now people can go on the internet and order virtually anything from anywhere, and this is the part of globalization that the world has went through in the past few decades.

During the Capstone class the robot that was built included parts from United States, Korea, China, Italy, etc. Some of the parts were purchased from a website Adafruit.com which has many distributors worldwide, when something was ordered for the project there was a big chain of companies worldwide used to deliver the product to the doorstep. The impact of this is amazing, just with one click of a button someone sitting home in the US could support someone in China or any other country that was used in this project.

The Dynamixel RX-28 motors were made in Korea, it is widely known that Korean culture includes very hard working and on point people, which very well reflected on the product that was delivered from their country, it had a lot of support and documentation online, with a very nice system integration. Italy is not known for their high tech but they did come out with Arduino which is a learning developmental board that is made cheap for schools and such, someone form the US took that idea and made a more advanced board teensy for college or maybe even graduate level, teensy is using Arduino IDE, this is a great example globalization, people from different countries coming together to create tools for learning that are better.

Not only do parts made from different areas in the world but also ideas on building the robot came from different people from different places. For example, as an international student from China, me myself Zhichen came up with idea on robot arm mechanism which made it extremely easy for our robot to pick up the treasure chest. Furthermore, ideas on how to navigate the robot also brought versatility to our project. For example, researching for Markov decision process to find optimal route and reinforcement learning to minimize the time of completion brought diversity to the project. Even though we didn't use those methods, researching those methods gave us better understanding on how information globalization could possibly impact Engineering.

In conclusion, throughout last few decades technology has aided in the globalization of the world, making our lives very simple and helping us with day to day tasks. During the capstone project we have benefited from globalization greatly by getting parts from other countries and even using services such as documentation support written by foreign people.

Jordan Drake, Rush Earman

Engineering, by definition, is the action of working artfully to bring something about. Over the past two semesters, our capstone class has worked tirelessly to bring something (our robot) about. While there were several late nights, long days, and endless hours put into bringing this robot to fruition, we did not create every piece in house. Many of the parts, especially the individual components, had to be ordered online and shipped to us, where we would piece it on the robot. The pieces we ordered online were manufactured across the globe, which is where globalization comes into play.

Vex, the company that created our omni-wheels has their main U.S. office and distribution center in the town where they first started, Greenville, Texas. This means that having the parts shipped to us was much faster than overseas, and the company has seven different distribution plants all across the world, to help ensure customers receive their packages quickly. LPKF, the company that makes the PCB printer we used is stationed in Germany, and as we saw during the beginning of the spring semester, it can be very difficult and expensive to get replacement parts and service. Other companies, like DYNAMIXEL who created our motors, are based in South Korea, and PJRC, who created the Teensy we used, all sell items on Amazon, where they can be purchased and received very shortly.

Globalization has allowed engineers to "bring something about" using parts from all over the world within a very short period of time. This has increased productivity, and especially during our capstone, given us flexibility to adapt the robot as we saw fit. While it is still expensive, and in-person services have not improved, globalization has revolutionized engineering.

Vincent Smith-DiLeo, Niles Tate, Alan Llyod

Within the past few decades, people around the globe have become interconnected through the internet, commercial air travel, and business opportunities. In our engineering capstone project, global markets have allowed us access to a variety of electronic components, power tools, machines, and other equipment that would not have been available to a college student just ten years ago. This uninhibited access to cutting edge resources has allowed our capstone group to create a fully functional robot from the "ground-up" in our design lab.

In terms of 3D modelling and printing, the capstone team was rather fortunate since we have a 3D printer in house. When it comes to the global economy, the only dependency we had on the 3D printing was the cost of the filament that we used. For the first few months, we used a polylactic acid (PLA) before converting over to carbon fiber filament. Although we purchased these filaments from sources inside the United States, it was because of the global economy that this was possible. Countries like the United Kingdom, Sweden, and the Netherlands produce filaments, which causes US companies to vary on prices to stay at a competitive level. Even if these countries didn't produce filaments, the companies inside the US would keep the prices relative, but the international prices certainly help.

In relation to the capstone, one of the obstacles we had to complete on the course involved the usage of a Raspberry Pi, a microcomputer, to transmit a 8 bit signal via IR. The Raspberry Pi and microcomputers in general are becoming more popular due to how small, simple, and useful it can be. Additionally, these minicomputers are also fairly cheap. The reason why Raspberry Pi was developed was to be able to spread and promote the teaching of basic computer science in schools and in developing countries. In countries like Ghana where there are only a few people who have access to the Web or Cuba where there are only 5% of Cubans have access to the open Internet, companies like Raspberry Pi is trying to change that with the most expensive microcomputer from the company being $35.00, the growth of computer science and electrical engineering around the world would increase very fast as well as the technological innovations if more companies jumped on board with this movement.

The impacts of globalization on engineering can be seen in the Electrical Engineering display case on the 500 level of Nichols. The robots from Electrical Engineering capstones from the past were handmade with 'crude' materials not generally used in Electrical Engineering i.e. wood, duct tape, and other sub-optimal hardware. Robots from the past 5 years, however, take a much more modern and professional appearance. Globalization in engineering has allowed recent Electrical Engineering capstones to utilize 3D printing, laser cutting, Printed Circuit Boards (PCBs), and affordable, high-performance microcontrollers. These high-end, industry level technologies have been made available by the globalization of engineering, greatly enriching the systems designed and fabricated by the Electrical Engineering capstone at VMI.

Quality Management Plan

Wyatt Raymond, Joseph Bruchalski
HR: None

Quality Definition

Project quality will be defined within the scope of product performance during the 2018 Southeastcon Competition. Quality will be defined as the degree to which a product reliably achieves a maximum number of point during this competition. In order to be reliable, the product must have low variance of points earned per run and low probability of failure during the competition time frame. The product will achieve points as defined in the official contest rules, which are available on the official 2018 Southeastcon Slack Channel.

Quality Management

The team leads will verify that team members produce deliverables within the guidelines of the quality definition as specified under “Quality Definition” in this document. The project manager will communicate with team leads and attend key system tests and demonstrations to verify that quality is implemented in the product at all stages of development.

Quality Assurance

In order to produce a reliable product, simplistic solutions will be favored in order to reduce the failure often associated with highly complex systems. To achieve a maximum number of points, multiple solutions will be researched for each product subsystem. This will allow developers to choose the most high performing and reliable system. During the product fabrication and integration phase (Part 2 of the WBS), subsystems will be tested individually and during integration. Subsystems and the system during different phases of integration will be tested in accordance with parts of the contest relevant to the subsystem or integrated system under test. The system will also be design to take potential course variance into consideration. Once the integrated system passes initial testing, the course will be modified to reflect a variance exceeding the expected maximum variance expected to be encountered at the competition.

Quality Control

A system that fails a test or exhibits the need for continual repair will be evaluated and reviewed in order to identify the cause for failure. If it is found that the source of failure is in the fabrication, the product will be rebuilt using a fabrication process that addresses the identified point of failure. If the source of failure is found to be in the design or cannot be identified, the product will be redesign and the identified point of failure will be addressed in the new design.

All deficiencies encountered will be added as entries to the process improvement plan. The process improvement plan will be consulted when designing, fabricating, integrating, and testing deliverables.

Human Resource Management Plan

Ed Olbrych and Buka Anwah
H/R: None

Roles and Responsibilities

Leadership

Project Manager: Manages and direct project workflow. Keeps track of goals, budget, and schedule.

Team Leads: Works with team on project. Manages and Directs team towards achieving goals and meeting deadlines. Works with other Team Leads and Project Manager to meet project goals and deadlines.

Teams

Tech Team: Designs electronic hardware portion of design. Interface, and verify functionality of all electronic components (sensors, microcontrollers, power supply etc.) and final electronic portion of design.

Software Team: Develops Code for microcontroller(s) to provide functionality to working system.

Hardware Team: Develops physical mechanical portions of design. Interfaces / chooses electro-mechanical components.

External Resources

Design Contractor: Advises team and designed parts per request of Team Lead and with Authorization of Project Manager.

Project Consultant Teams: Two Consulting Teams will be formed to review the project and offer solutions both informally and during formal processes such as monthly PMR meetings.

Project Organization Chart

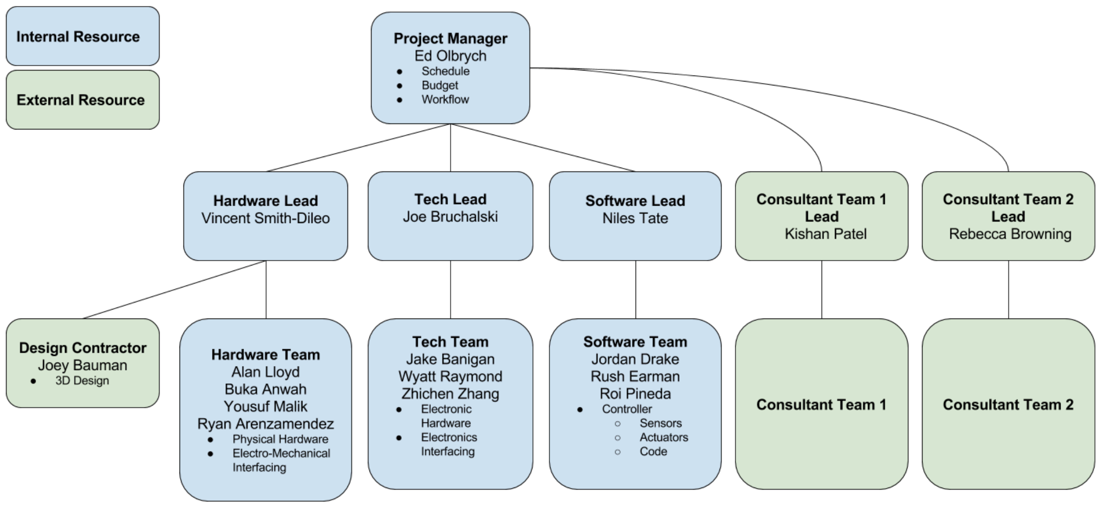


Figure 2. Project Organization for the Semester of FL17

Staffing Management Plan

Team members will present Individual Development Plans to team their respective team leads prior to the first PMR meeting. Team leads will meet with their team members on a monthly basis to track progress.

Individual Development Plan Template

Professional Goals / Motivations

Describe why you want to develop yourself over the course of the Capstone Project. What are your big-picture career goals?

Top Three Strengths

What are your top three strengths that you use to contribute to the capstone project as part of the Team.

Top Three Areas of Improvement

Specifically, what could you improve to achieve more as part of a high performing team.

Development Plan

Briefly explain how you plan to further your strengths and develop your areas of improvement.

Risk Management Plan

Roi Pineda, Jordan Drake, Rush Earman IV
Help Received: None

Methodology

A risk register will be continually updated to track risks as they are identified throughout the project. Risk mitigation and response plans will be put in place as risks are identified. PDRs, CDRs, and other project documents will be discussed to identify points of failure. Multiple design solutions for each system will be considered although fallback designs will not be documented to the CDR level. Logistics will also be considered as a point of failure pertaining to parts as well as to personnel. The impact will be taken into consideration when allocating resources to address risks. The stakeholder register will be consulted when planning for and reacting to risks. Stakeholders will be approached in a way that is within the scope of their involvement with the project and is consistent with their communication preferences (as outlined in the stakeholder register).

Roles and Responsibilities

Risk Management will be coordinated by the project manager and will incorporate items identified by the three teams. The Team Leads will facilitate risk identification, documentation, and risk mitigation and response plan generation.

Budgeting

Since the project was originally contracted with a cost plus budget, any overages which occur from risks will be covered by a request for additional funds. The project charter allows for overages so long as the total budget cost does not exceed $10,000. This allows for a cost overage of $6981.82, which is more than twice the initial project cost. Risk response plans will take this budget cap into account so that contingencies may be implemented without exceeding this budget cap.

Timing

Risk Management will be conducted during team meetings. Any risks identified during these meetings will be added to the risk register. When a risk response plan goes into effect, funds will be requested by the project manager from stakeholders in accordance with the project charter which allows for overages that would prevent a serious negative impact on the project. Risks will also be addressed at the monthly PMR meetings.

Risk Categories

Risks will be categorized according to the team or group of people to whom the risk pertains the most.

Risk Probability and Impact

Risks in the Risk Register will be given a risk level based on the Probability and Impact Matrix (see figure 1.).

Probability	**Risk Level**				
0.9	0.05	0.09	0.18	0.36	0.72
0.7	0.04	0.07	0.14	0.28	0.56
0.5	0.03	0.05	0.1	0.2	0.4
0.3	0.02	0.03	0.06	0.12	0.24
0.1	0.01	0.01	0.02	0.04	0.08
	<= 0.05 / Very Low	0.1 / Low	0.2 / Moderate	0.4 / High	0.8 / Very High

Figure 3. Probability and Impact Matrix.

Reporting Formats

Each risk entered in the risk register will contain the following fields: Risk, Description, Category, Cause, Probability (Low, Medium, High), Impact, Proposed Responses (ie. Response Plan), Owner (Individual or group who will track and manage this risk), and Status.

Tracking

Risk planning outcomes will be documented in the Risk Register. The register will be made available to all team members. Team Leads will gather risks for this document and will update risks accordingly. The Project Manager will also edit and update this document. The risk register will be consulted during team meetings, demos, and any other situation where a point of failure is eminent.

Risk Register

#	Risk	Description	Category	Cause
1	Chassis / components break at competiton	Mechanical / circuitry failures, wiring problems	Hardware / Tech	Mechanical stress and circuit burnout over time
2	Variance in Competition Boards	Slight differences in competiton board due to non uniform fabrication	Software	Host team lack of attention to detail in creating multiple competition boards
3	Manufacturer unable to deliver product / product defects	Inability for part manufacturers to deliver part for unforseen circumstances, defects in part fabrication process	Tech	faulty manufacturing equipment, natural disaster
4	Inability to access files/resources at competition	Loss of WiFi connectivity	Software / Tech	Lack of wifi caused by power outage or poor signal
5	Vehicle Breakdown	Vehicle is unusable or breaks down on the way to competition	Logistics	Age of vehicle, mechanical failure
6	Team Demotivation	Team members are reluctant to do work, are not interested in project success, do not ownership	PM	Overworked, External Stress due to other obligations, Lack of visible progress, conflict within teams

		of project		
7	**PCB/Sensor/Teensy Burnout**	**Something breaks, we can't replace/ Vendor ends support of system**	**Tech/Hardware**	**Overuse/ Inproper use/not handled properly/mistakes**
8	Sensor/Motor does not work as planned	a sensor that does not have the capability to complete the task. Motors that are not effective/ reliable enough to work.	Tech	Improper research, Overlooking of multiple things/ poor planning
9	**Robot breaking, Pieces being pulled out.**	**a piece of the robot breaks, some wires get pulled out.**	**Tech**	**Accidental, not enough care being placed in keeping robot intact. improper handling of robot.**
10	Computer problems	Laptop we decide to use for coding gets damaged or crashes	Software	Dropping, Water Damage or Computer Failure
11	**Rule changes at last minute**	**If adjustments to the rules are made due to problems**	**Software/Tech/Hardware**	**Course problems or IEEE decides to change**
12	Sensor Failure	If sensor breaks, moves or malfunctions and due to the inability to accurately measure and get information	Software/Tech	Sensor not being correctly aligned, sensor breaking or sensor malfunctions and inaccurately measures data

#	Risk	Probability	Impact	Probability (Numeric)	Imapact (Numeric)	Threat Level (Numeric)
1	Chassis / components break at competiton	Low	Catastrophic Failure	0.3	0.8	0.24
2	Variance in Competition Boards	High	Loss of Points	0.7	0.4	0.28

3	Manufacturer unable to deliver product / product defects	Low	Delay in Design / Robot fabrication schedule	0.3	0.2	0.06
4	Inability to access files/resources at competition	Very Low	Loss of Points / Inability to alter designs while at competition	0.1	0.4	0.04
5	Vehicle Breakdown	Low	Delay in Personnel / Robot Arriving on time to Competition	0.2	0.8	0.16
6	Team Demotivation	Moderate	Delay in schedule, Decrease in qulity	0.5	0.2	0.1
7	**PCB/Sensor/Teensy Burnout**	**Moderate**	**catosrophic/ Our robot will not work/ delays scheduling and anyone that depends on the system to perform**	**0.5**	**0.8**	**0.4**
8	Sensor/Motor does not work as planned	Low	Delay in schedule, Robot potentially won't work. Software will be effected heavily.	0.3	0.2	0.06
9	**Robot breaking, Pieces being pulled out.**	**moderate**	**Catostrophic/ Robot will not function**	**0.5**	**0.8**	**0.4**

10	Computer problems	Low	Loss of Points/Inability to complete course	0.3	0.8	0.24
11	**Rule changes at last minute**	**Very High**	**Loss of Points**	**0.9**	**0.4**	**0.36**
12	Sensor Failure	Low	Catostrophic/ Robot will not function properly	0.3	0.8	0.24

#	**Risk**	**Mitigation Plans**	Owner	Status (Open, Closed)
1	Chassis / components break at competiton	Bring backup robot / replacement parts and circuitry, Bring 3D printer	Vincent / Joe	Open
2	Variance in Competition Boards	Software must be flexible, create search patterns to find objectives that are slightly off from specs	Vincent	Open
3	Manufacturer unable to deliver product / product defects	Buy parts from multiple vendors	Vincent / Joe	Open
4	Inability to access files/resources at competition	Bring all necessary files on USB sticks / bring phones, tablets with cellular internet	Vincent / Joe / Niles	Open
5	Vehicle Breakdown	Identify backup drivers, ensure enough cars are taken to accomodate all personnel arrive if a vehicle breaks down	Ed	Open

6	Team Demotivation	1.Get feedback from team members, do not assign excessive work to overworked / stressed team members, 2. Identify conficts by "walking around". Move conflicting team members to different teams. 3.Team bonding activities will be conducted to facilitate good relations amoung team members.	Ed	Open
7	**PCB/Sensor/Teensy Burnout**	**Have multiple replacement sensors/ have more than enough of everything to the extent that if you break something there is a backup for the backup already**	**Joe**	**Open**
8	Sensor/Motor does not work as planned	Have a suitable replacement piece. Do research continually. Also test and make sure that the products can perform.	Joe	Open
9	**Robot breaking, Pieces being pulled out.**	**Have a proper schematic of everything that we built. In depth documentation on any pertinent pieces. Have back up Teensys/Breadboards,PCB,Batteries.**	**Joe**	**Open**
10	Computer problems	Post all code into Box.com to access from any computer	Niles	Open
11	**Rule changes at last minute**	**Bring laptop to make immediate changes and bring extra hardware and tech equipment to be able to adjust**	**Niles/Joe/Vincent**	**Open**
12	Sensor Failure	Bring extra sensors	Niles/Joe	Open

Stakeholder Management Plan

Zhichen Zhang and Niles Tate
H/R: none

Identification

Stakeholder were identified as individuals who have may have a serious impact on, and are invest in, the outcome of the project (see Table 2).

Table 2. Stakeholder Identification Matrix

Stakeholder	Investment	Impact
COL McCormick	Teaches Capstone Course	Authority, Scope
COL Addington	Department Head, Authorizes project budget from primary investor (ECE Department)	Budget
Rick Amenell	Handles all outgoing orders, Responsible for the lab and equipment used for the project	Procurement, Work Area
Dawn Cochran	Coordinates all travel and registration expenses for the competition.	Logistics
COL Livingston	Interested in project success as IEEE Student Branch Counselor	Communication

Planning

The stakeholders are listed in the stakeholder register which includes the following details for each stakeholder: name, role, communication goal, contact frequency, contact method, engagement level, details of interest, tolerance for overages, and additional notes. A Power / Interest grid was used to identify how each stakeholder should be managed (see Figure 4).

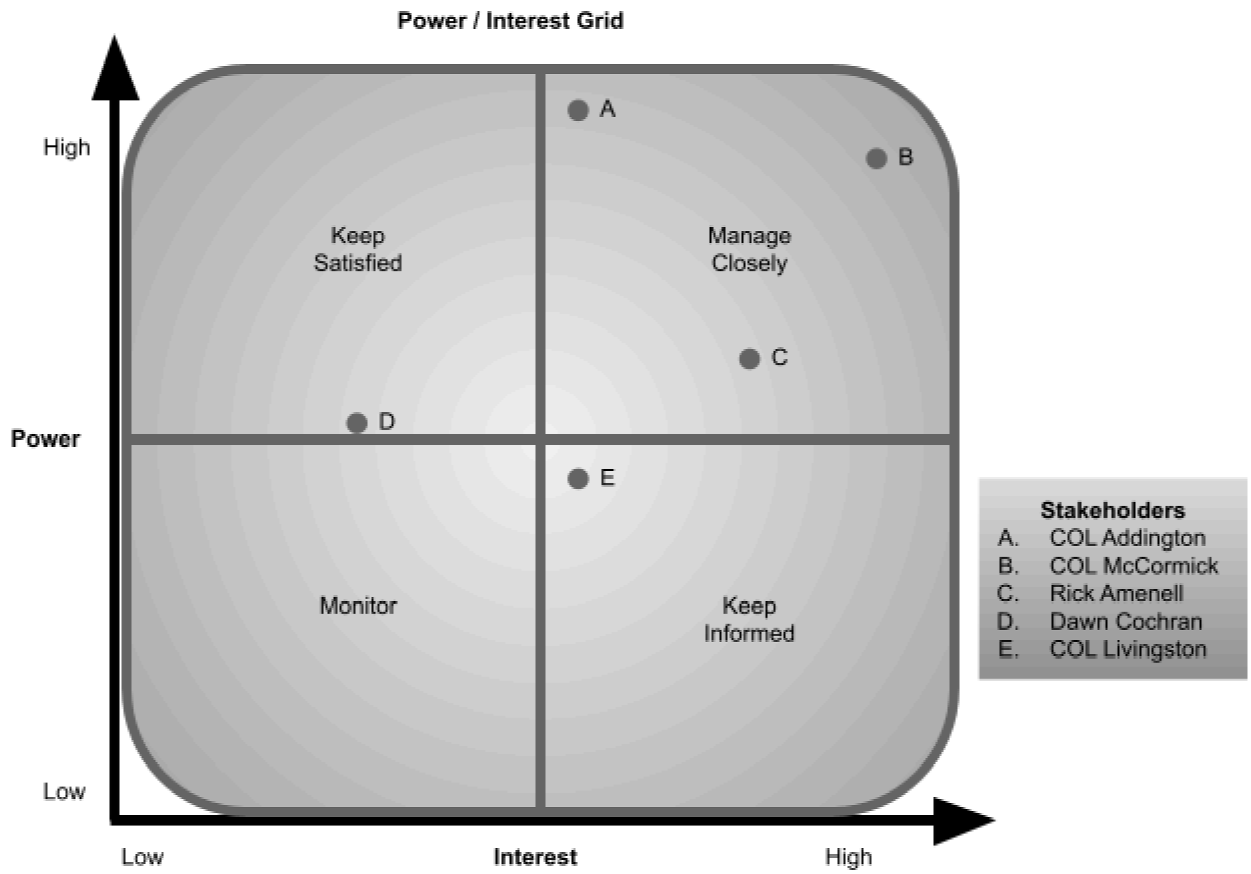


Figure 4. Power / Interest Grid.

Manage Engagement

In order to track stakeholder engagement, Initially stakeholders have have a low impedance to project objectives and have appeared to buy into the project. In order to track engagement, a Stakeholder Engagement Assessment Matrix was created (see Table 2).

Table 2. Stakeholder Engagement Assessment Matrix

Stakeholder	Unaware	Resistant	Neutral	Supportive	Leading
COL McCormick					CD
COL Addington					CD
Rick Amenell					CD
Dawn Cochran				C	D
COL Livingston				C	D

Control Engagement

Stakeholders will be invited to participate in both weekly meetings and PMRs. They will also be provided with a written progress report after each monthly PMR. Stakeholder meetings will also be scheduled at least once during the completion of part one of the WBS and once during the completion of part two of the WBS.

Alternate Stakeholder Analysis

PROGRAM STAKEHOLDER ANALYSIS			
PROGRAM NAME	VMI Southeastcon 2018	PROGRAM MANAGER	Edward Olbrych

NAME	ROLE	COMMUNICATION GOAL	CONTACT FREQUENCY	CONTACT METHOD
Sponsors, managers, users, etc.		Request specific activities or actions, recognize performance, ask for resources, influence performance or morale, encourage input, etc.	Daily, weekly, upon request, specific activities only, milestones only, etc.	Email, phone, in-person, presentation, web conference, etc.
COL Shawn Addington	VMI ECE Dept. Head	Aquire Resources, Increase Funding, Authorize / Support Project Events	Monthly	Email, in-person, presentation

COL McCormick	Professor and Project Advisor	Consulting, Recognize Performance	Weekly	Email, phone, in-person, presentation
Rick Amenell	Electronics Technichian, Aquisitions	Aquire Resources,	Weekly	Email, phone, in-person
Dawn Cochran	Administrative Assistant	Project Funding Coordination, Logistics Planning, Logistics Funding	Weekly	Email, in-person
COL Livingston	IEEE VMI Branch Counciler	Communication with 2018 Southeastcon Coordinators and Supporting Organization (IEEE), IEEE Funds (for logistics), Consultant	Weekly	Email, in-person

NAME	ENGAGEMENT LEVEL	DETAILS OF INTEREST	TOLERANCE FOR OVERAGES	ADDITIONAL NOTES
Sponsors, managers, users, etc.	Unaware, Resistant, Nutral, Supportive, Leading	Specific activities, milestones, progress, issues, etc.	Tolerance for budget overages, tolerance for schedule overages, acceptable reasons for each.	Work style, attitude, influence, experience, etc.
COL Shawn Addington	Supporting	Milestones, Student Achievement	Tolerance for budget overages: HIGH, tolerance for schedule overages, MODERATE,	All communication must be conducted in a profesional mannor.

			acceptable reasons: improved accademic success and achievment	Professionalism!!
COL McCormick	Leading	Milestones, Progress, Student Achievment.	Tolerance for budget overages: HIGH, tolerance for schedule overages, MODERATE, acceptable reasons: improved accademic success and achievment	Keep informed about progress, issues, etc. Any feedback needs to be acted on immediatly.
Rick Amenell	Leading	BOM / Orders / procurement	Tolerance for budget overages: LOW: cannot procede with orders without approval for overage, tolerance for schedule overages, HIGH	Likes things neat and organized
Dawn Cochran	Supporting	Competition Attendance and Conference Logistics Planning	Tolerance for budget overages: MODERATE: Regarding Logistics, this is not acceptable, tolerance for schedule overages, LOW, as	Likes prompt communication, likes things neat and organized

			this pretains to Conference Logistics	
COL Livingston	Supporting	Competition Success, Student Achievement	Tolerance for budget overages: HIGH: Primarily interested in project outcome, tolerance for shedule overances: MODERATE, primarily as this would impact project success.	Likes prompt communication, student involvement

Product Improvement Plan

Sensor Array

1. Add Bump Sensors to the Sensor Array for improved border detection.
2. Add sensor array to read the LEDs at stages to provide feedback. Maybe have a row of 5 or 6 sensors will a green filter.
3. Look into using IR sensors sampled at a fast rate → compare with time of flight sensors
4. Research new sensors to use with or in place of current sensors
5. Add extra teensies for each group of time of flights
6. Mechanism to verify completion of flag raising mechanism.
7. Add Bump sensors to front to detect if robot is not centered on chest
    a. Align robot to chest until front sensors are not in contact with chest.

Wheels

1. Add Larger Wheels to Improve course completion time.

System Integration

1. Have a detailed set of instructions that describes how to build the robot from scratch as well as assemble it from spare parts (pre-built modules: PCB, Power System, 3D printed parts, etc.).

Code

1. Find out if pointers take less time than global variables. If so replace global variables with pointers.