DiWheel

Kari Dennis

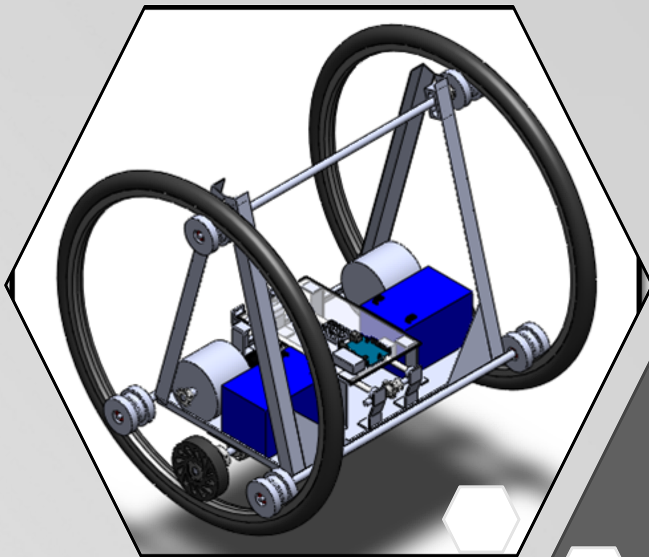
Kevin McLaughlin

Christopher Vanjoff

Christopher Parisi

Joshua Dean

The diwheel project will be a two-wheeled, durable, mobile, radio controlled vehicle with the ability to attach hardware to its chassis.



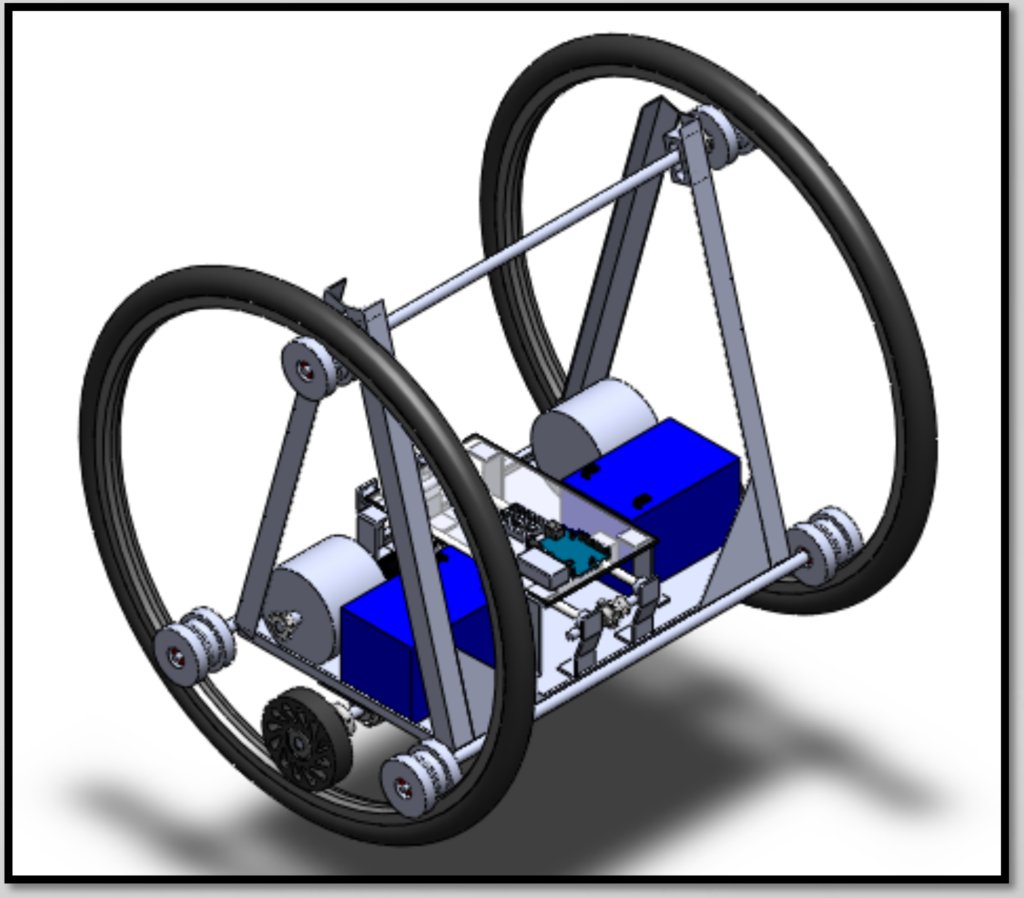
Radio Controlled

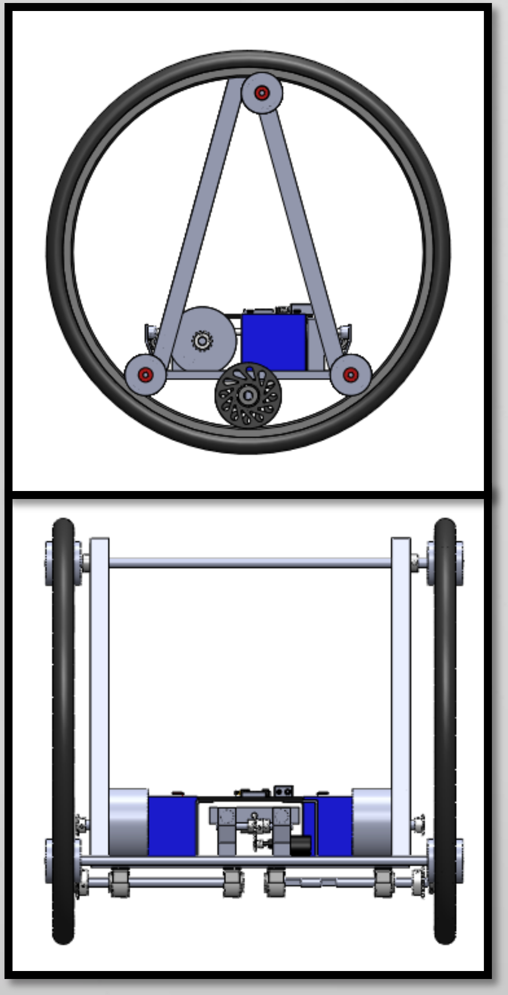
Two-Wheeled

Problem Definition:

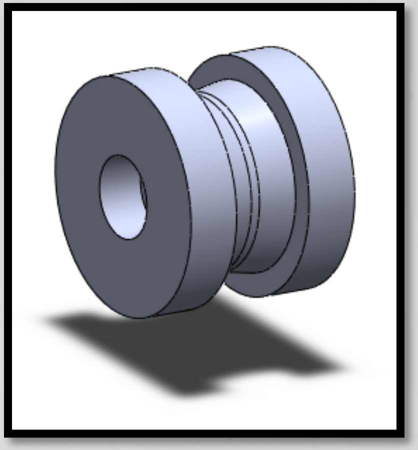
The problem that we have been given to solve is to design a **two-wheeled, mobile, durable** vehicle on which different **hardware can be mounted** and kept level. Diwheel mobility should include the ability to move forward, backward, and in circles using some form of radio control. For vehicles with two wheels in parallel side by side, the problem comes in keeping the center chassis level. Our project should incorporate a unique way **to keep the chassis level while in motion**. Because the only connection between the chassis and the wheels is through the drive train, adding any torsional correction for stability will take away from the power being delivered to the wheels. Our team must design a way to incorporate stability without taking away from the power.

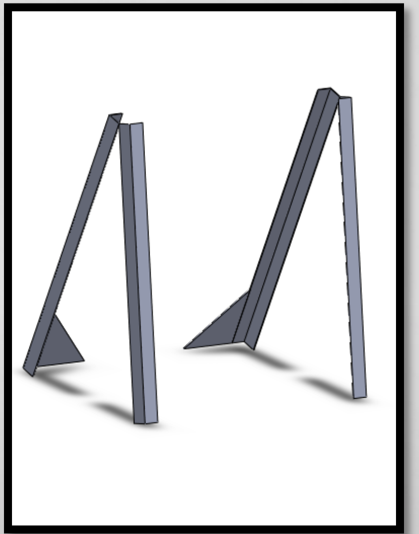
The detailed design model was created in CAD.

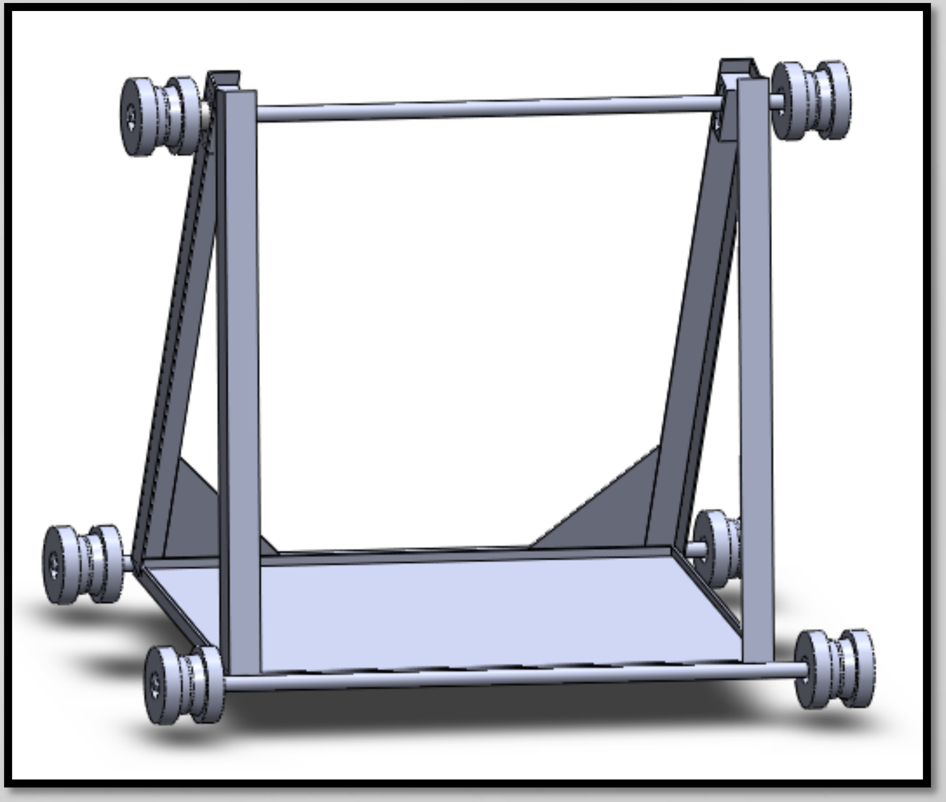


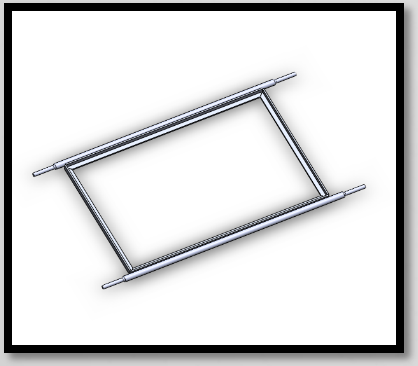


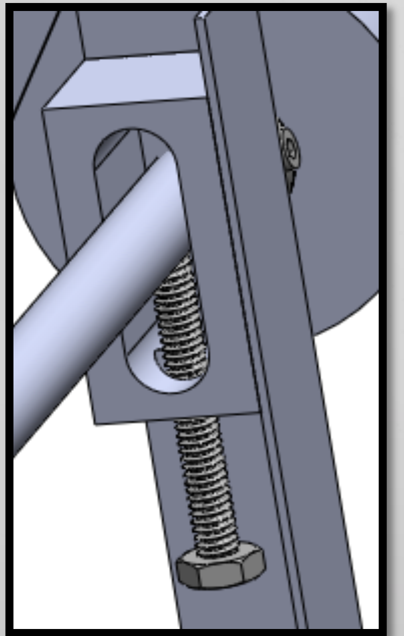
The chassis is where all the components are mounted.



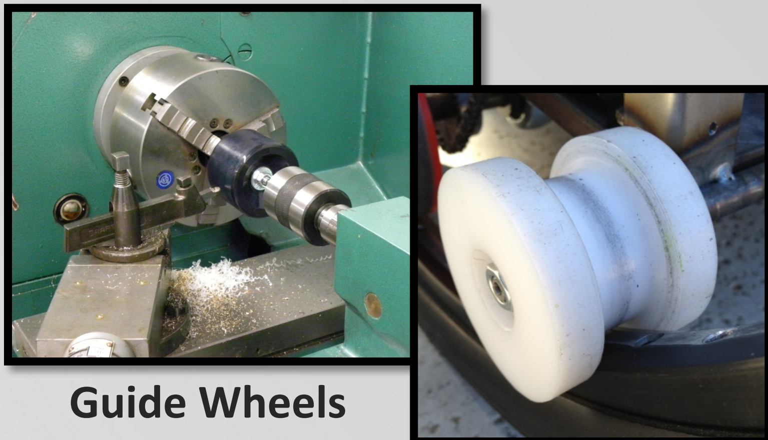








A complete chassis consists of the frame, plate, A-frame, guide wheels, and tensioner.



Guide Wheels

Frame

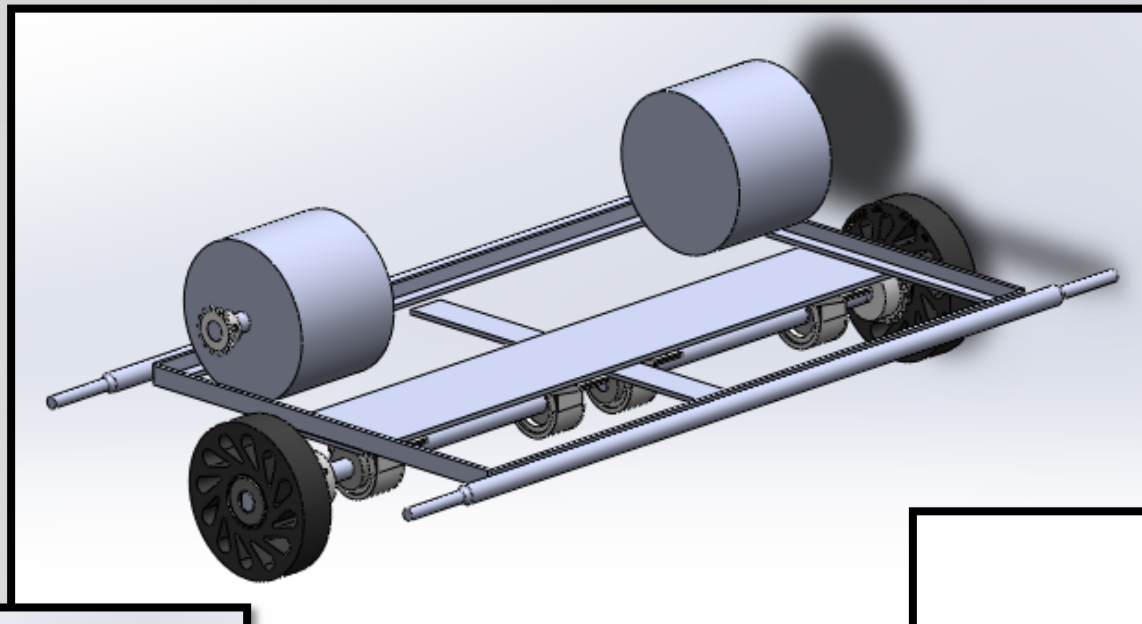
Tensioner

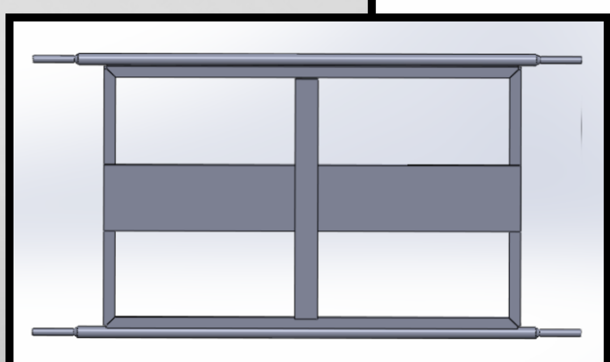
A-Frame

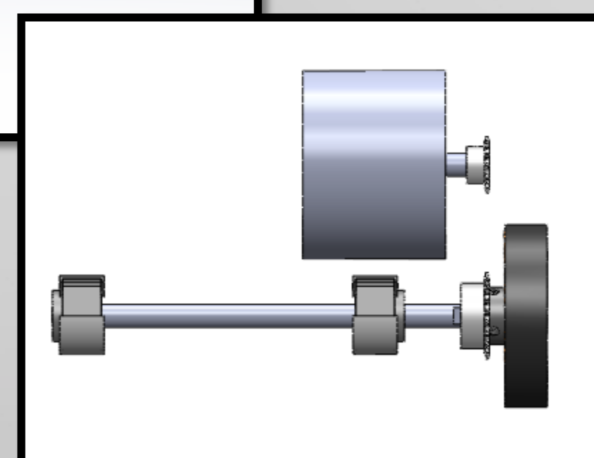
Plate

New guide wheels were made to be the same size and shape but from a low friction plastic.

The drive train provides power transfer from the motors to the drive wheels which then spin the bike wheels.



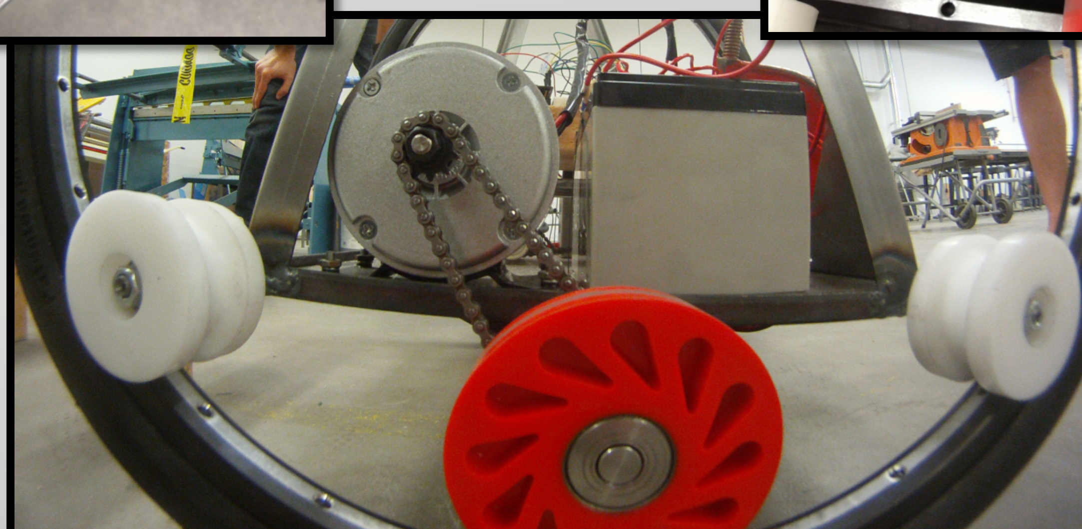




A complete drive train consists of a drive axel, bearings, and wheel along with supporting straps.

Axels, Bearings & Supporting Straps

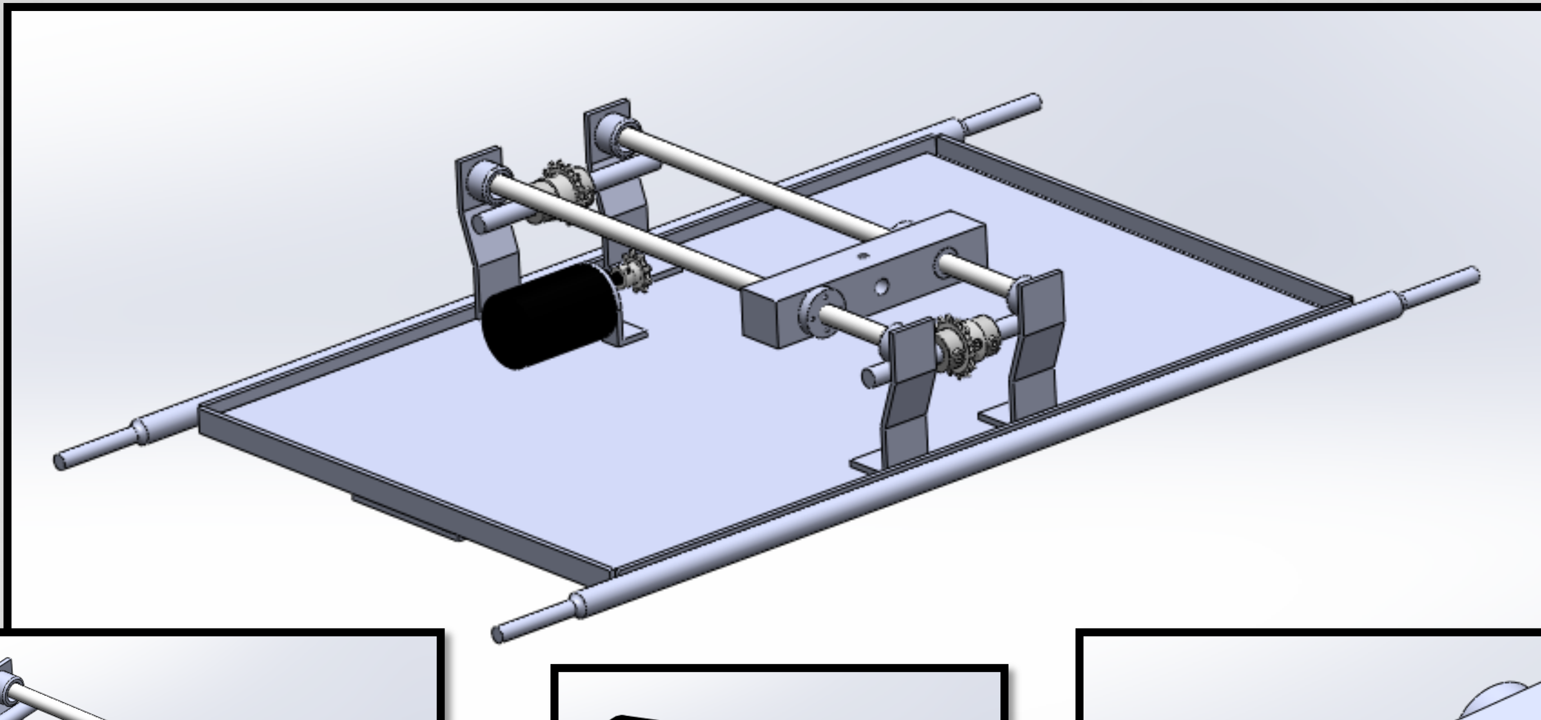
Drive Wheel, Motor & Chain

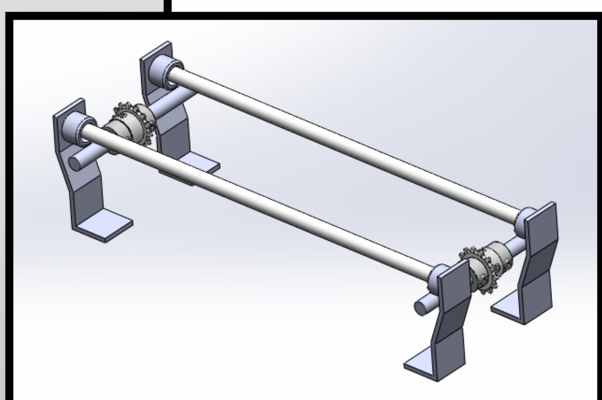


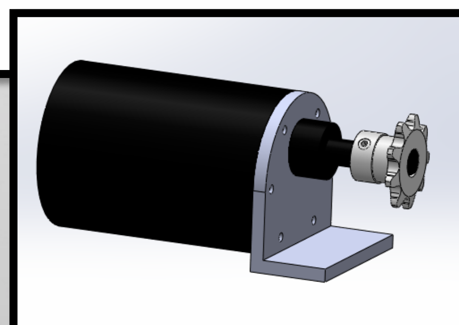
Drive Train System

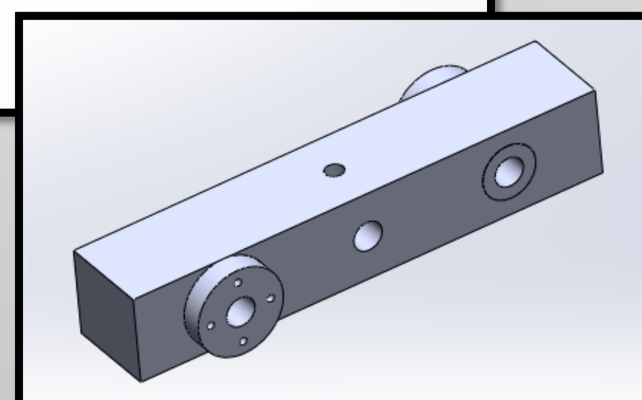
A support strap was added to maintain secure contact between the drive wheel and the rim of the bike wheel.

The leveling system provides chassis leveling while the diwheel is in motion.



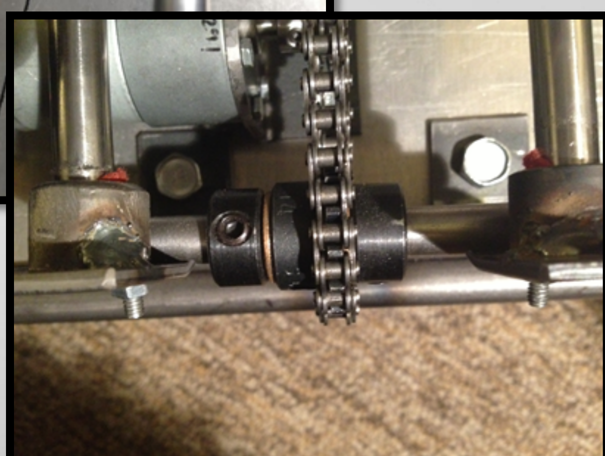




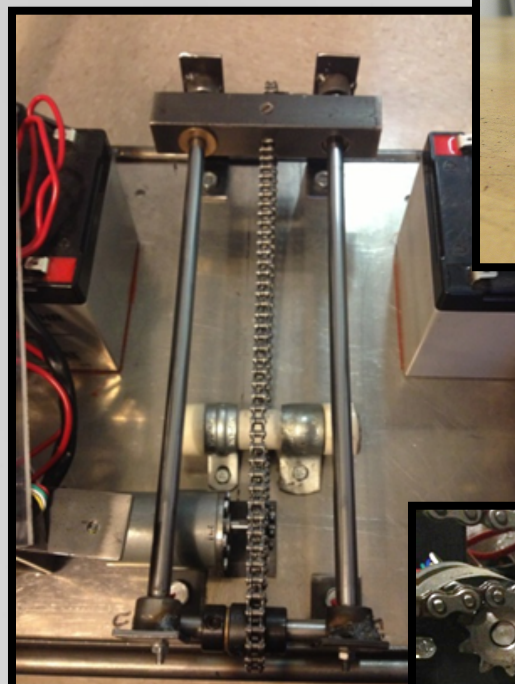


The leveling system consists of a weight, rods with mounts and axel, and a motor.

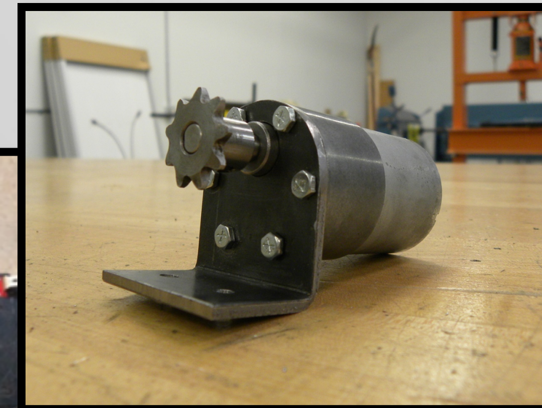
Rod & Mounts



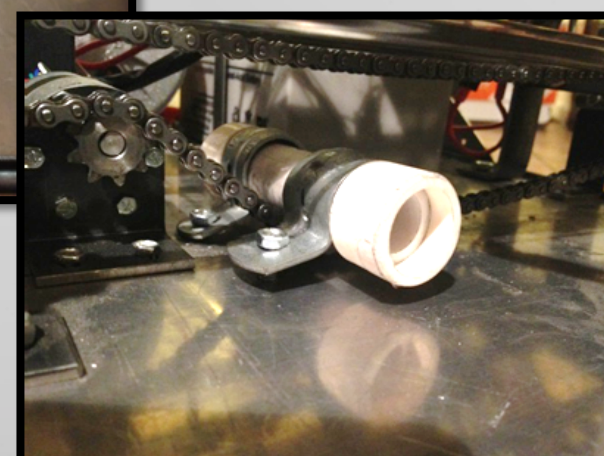
Axel



System

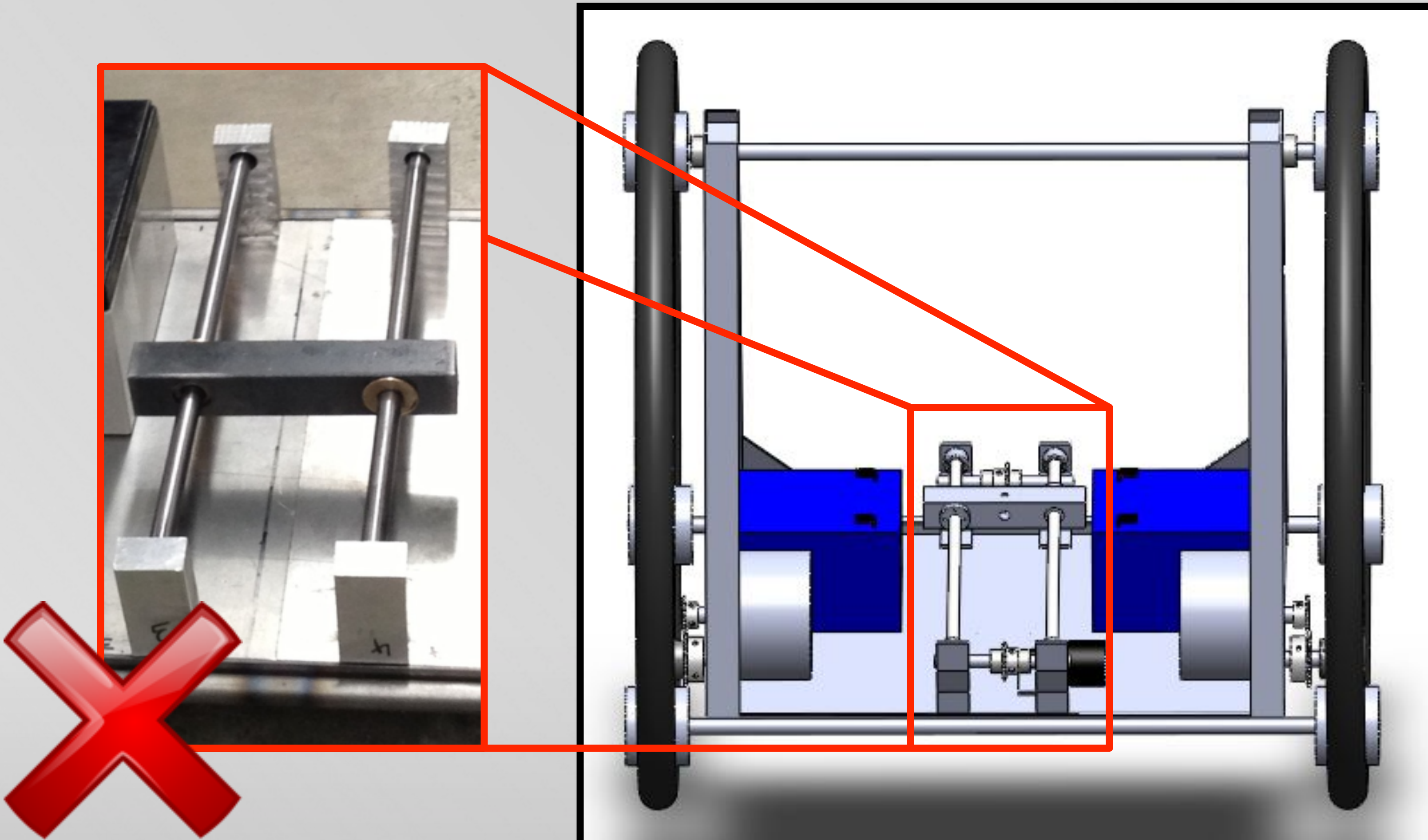


Motor

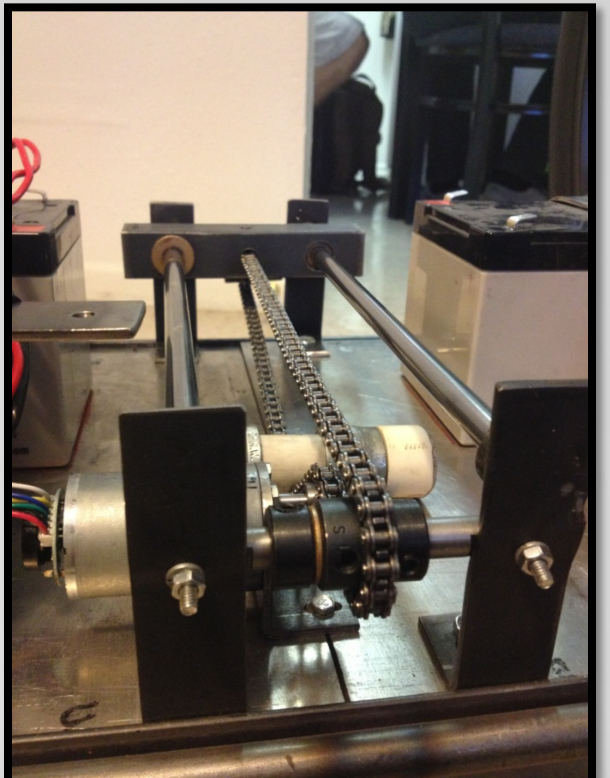


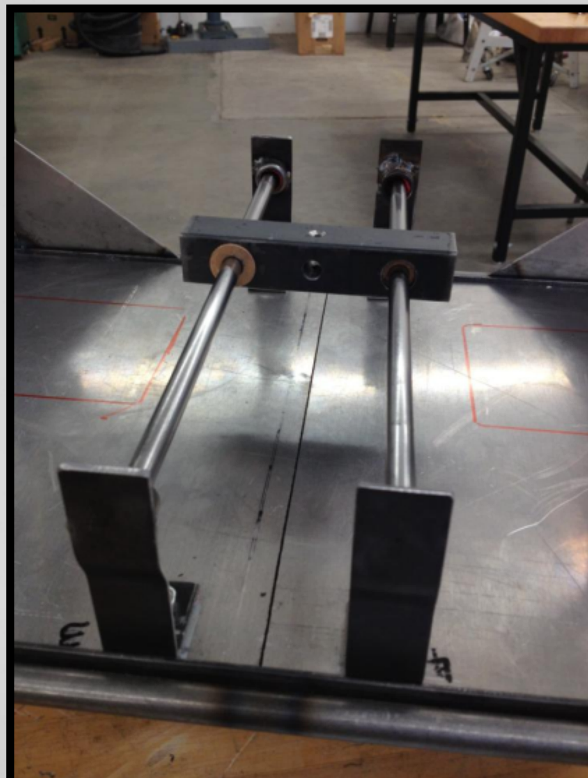
Motor, Chain & Tensioner

The leveling system initially failed the test for friction, placement and strength.

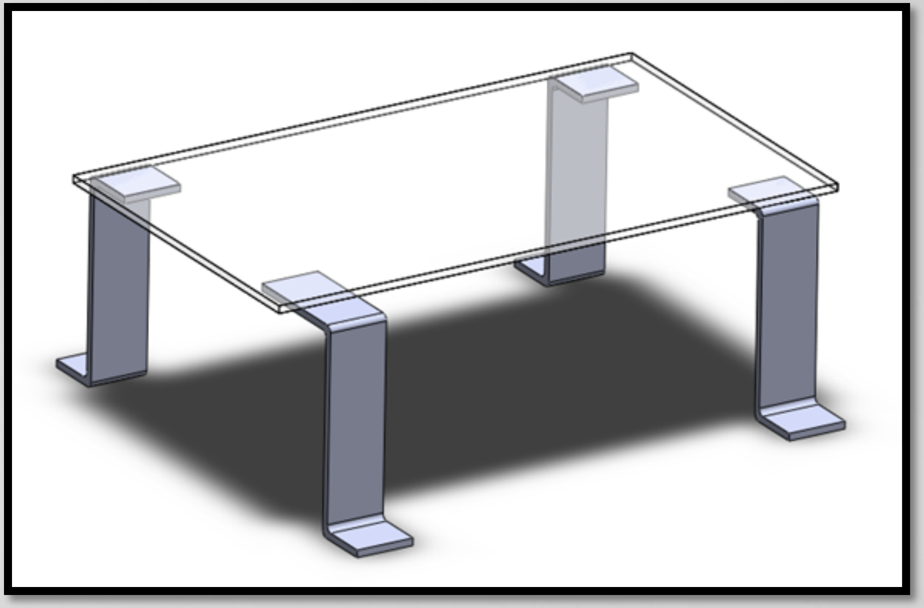


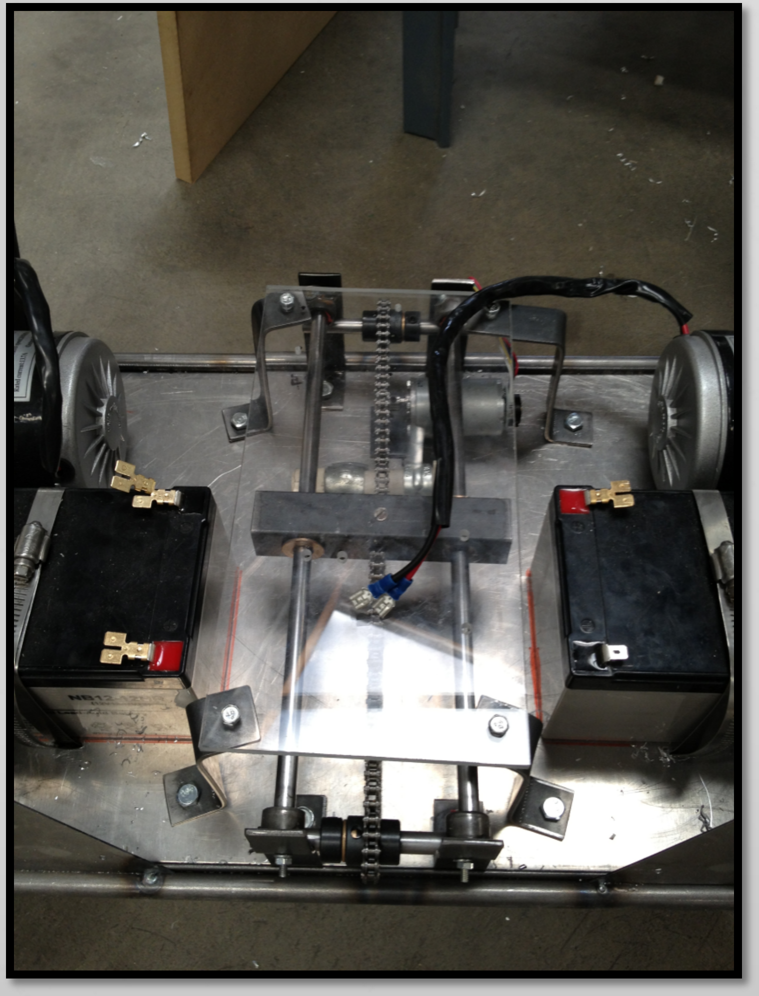
After redesign, the leveling system passed the friction, placement and strength tests.



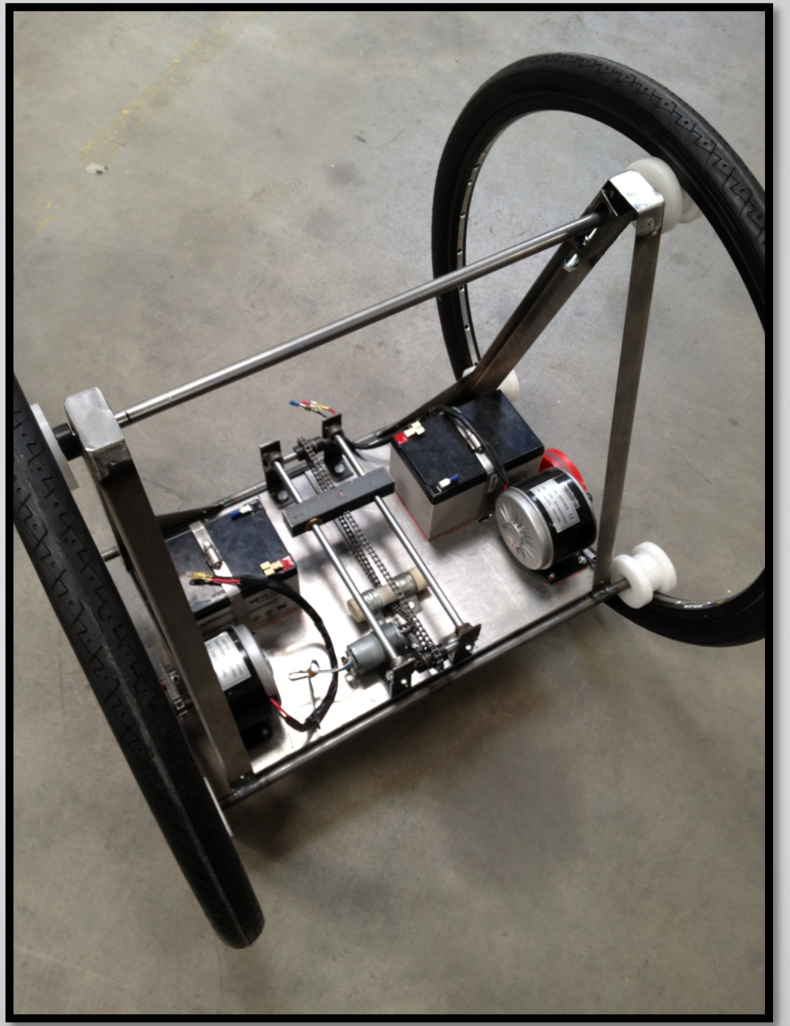


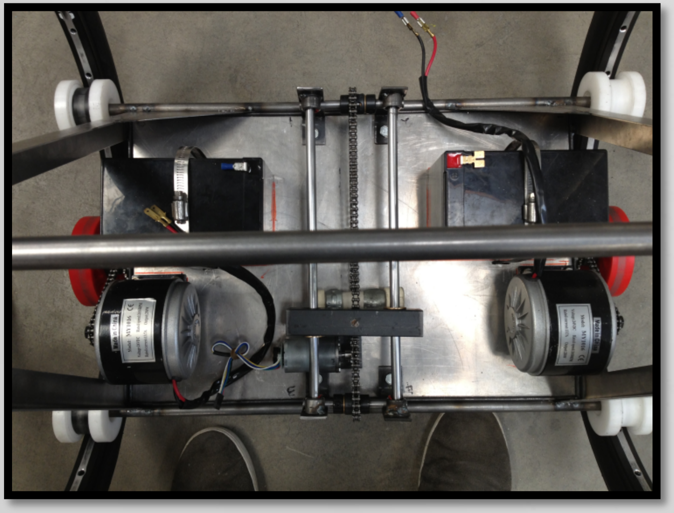
The hardware table provides a surface to mount the electrical components of the diwheel.

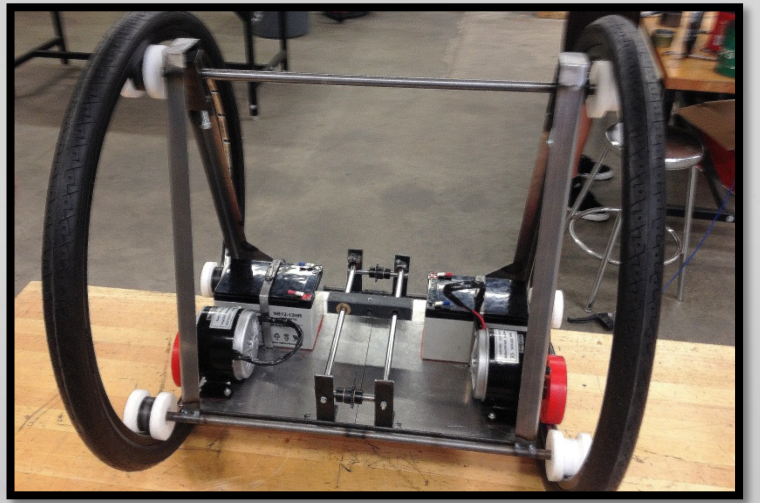




All mechanical subsystems were combined to create the complete mechanical prototype of the diwheel.





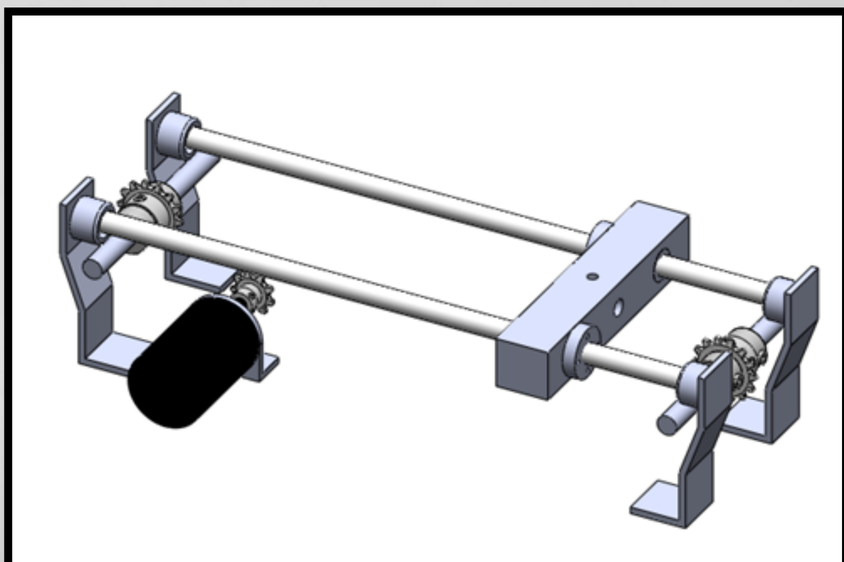


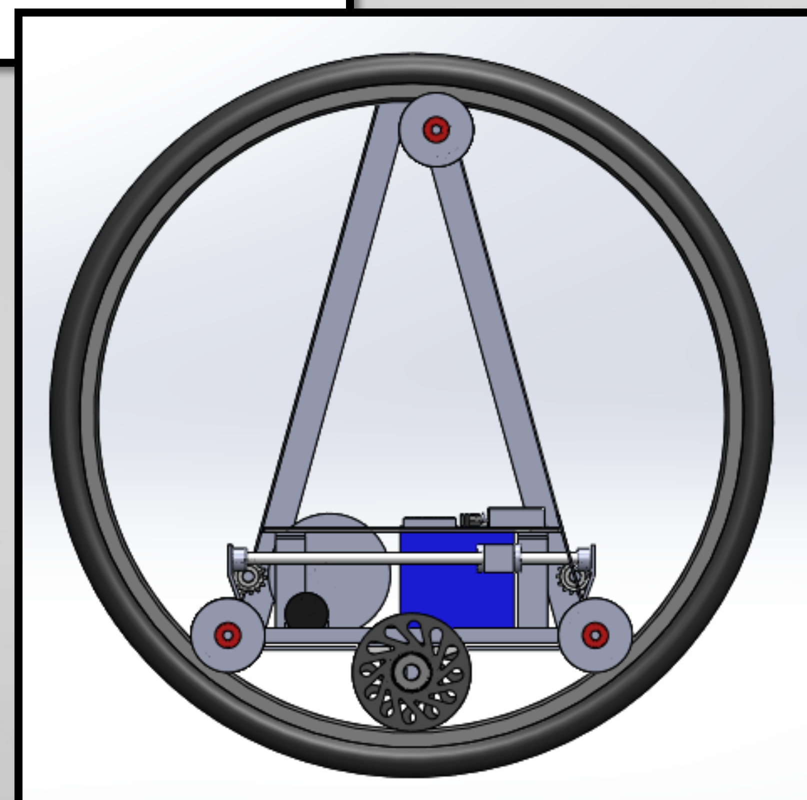
The diwheel is governed by three dynamic equations that will be used to develop the control systems.

$$[M_b e^2 + J_b + m(r^2 + l^2)]\ddot{\theta} - M_b g e\theta + mg(l - r\theta) - (M_b Re - mRr)\ddot{\phi} + T + mr\ddot{l} = 0 \quad (1)$$

$$[M_w R^2 + J_w - M_b(R - e)R - m(R - r)R]\ddot{\phi} + [J_b + M_b(R - e)R + m(R - r)R]\ddot{\theta} - M_b g e\theta + mg(l - r\theta) - m(R - r)\ddot{l} = 0 \quad (2)$$

$$m(\ddot{l} + R\ddot{\phi} - r\ddot{\theta}) + F + mg\theta = 0 \quad (3)$$

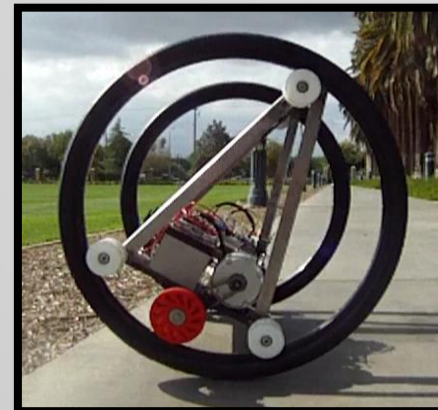




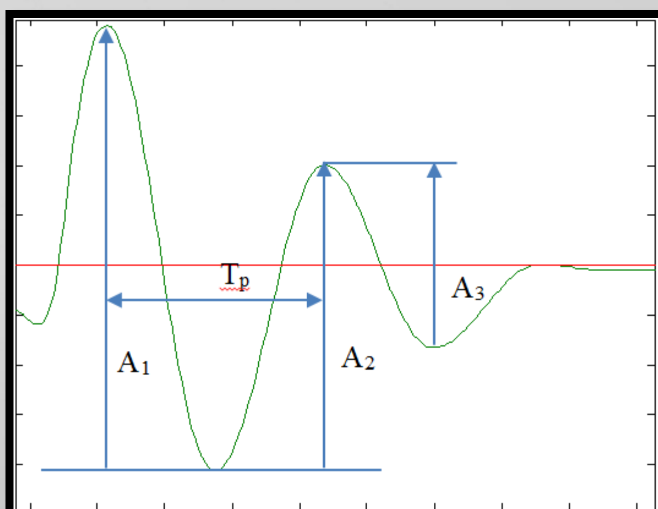
The damping ratio of the diwheel was found by applying a disturbance and measuring the successive amplitudes.

Amplitude 1

Amplitude 2

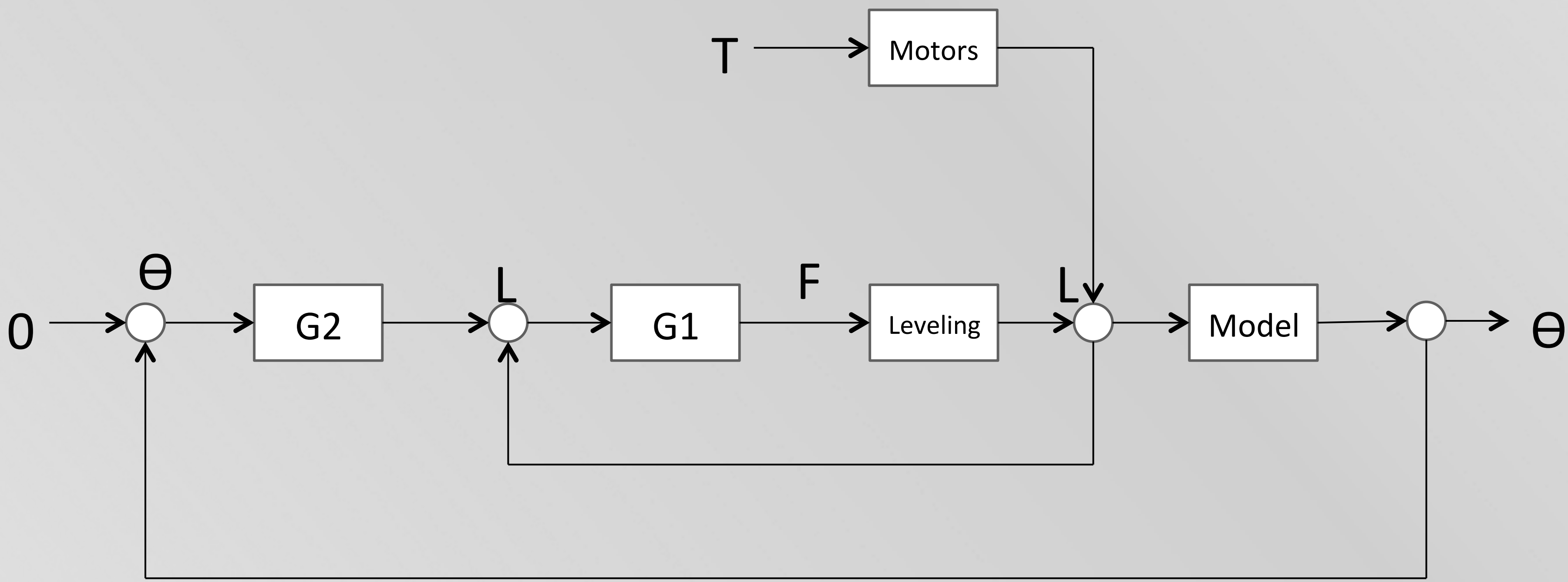


Amplitude 3

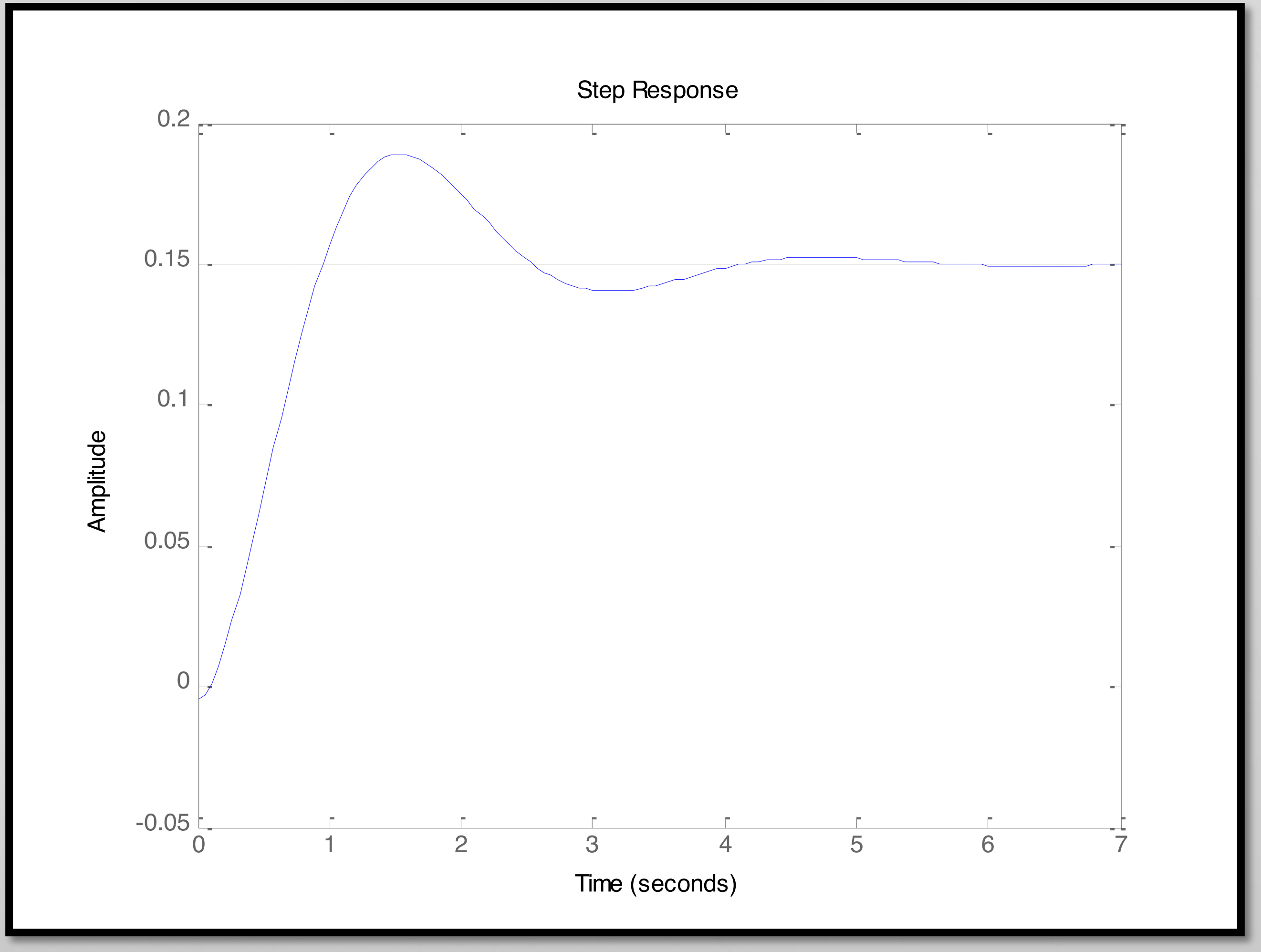


The damping ratio of the diwheel was found by applying a disturbance and measuring the successive amplitudes.

A block diagram was created to model the control system for the diwheel.



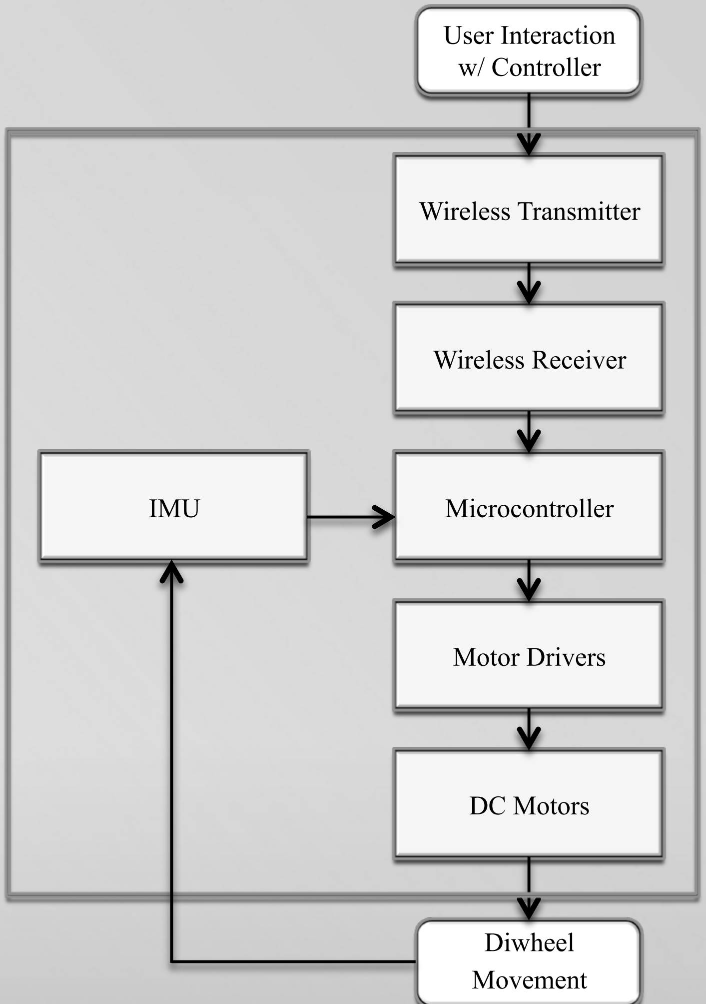
MATLAB was used to design a controller which quickly brings the system to steady state.



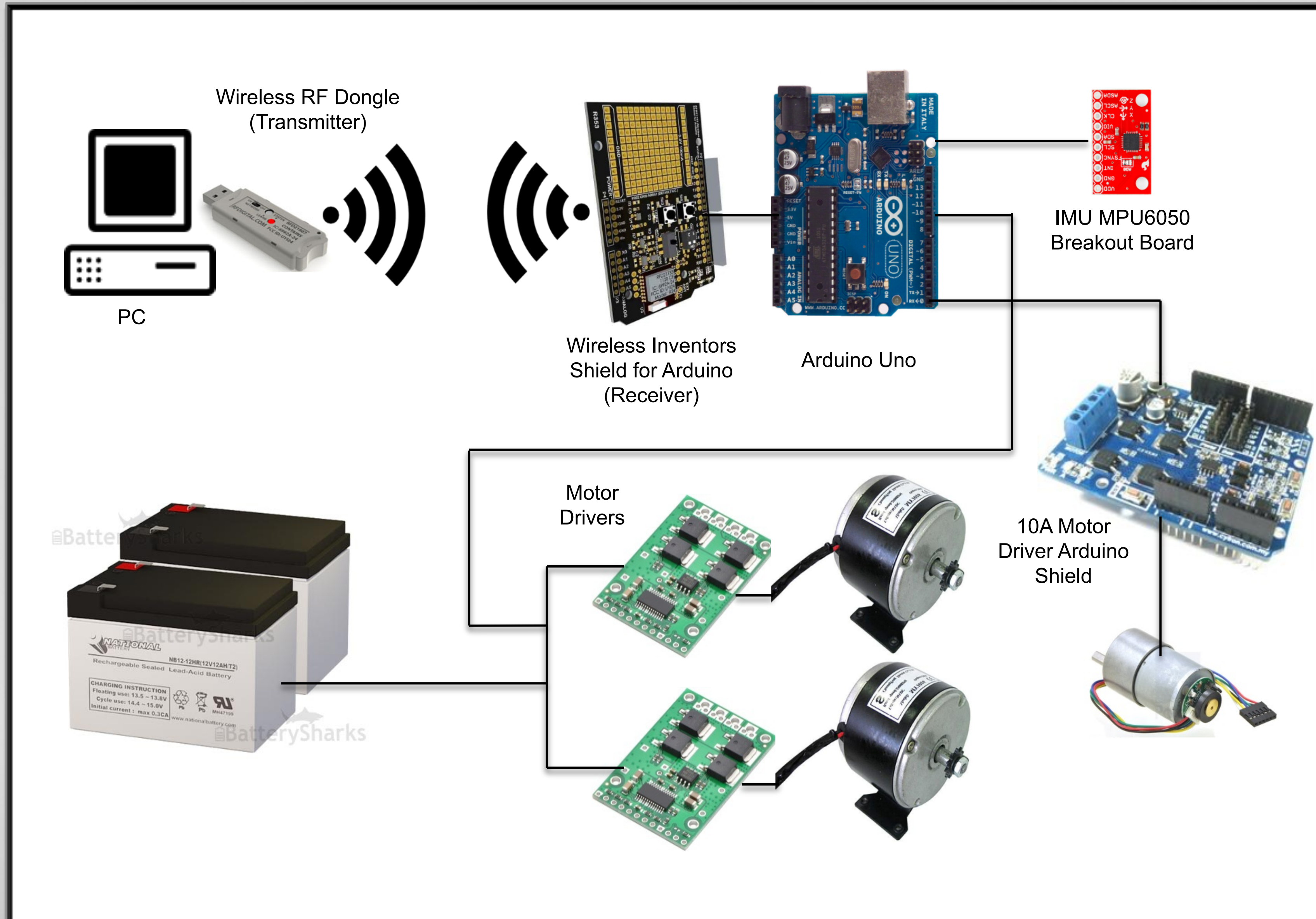
Level 0 of the electrical design architecture shows the overall picture of the functionality of the hardware.

Module	Hardware
Inputs	-Directional data from user's controller (forward, backward, left, right) -Current inertial data from diwheel (leveling)
Outputs	A voltage to control direction of DC motors
Functionality	Using the data received from the user's controller and the current status of the diwheel's inertia, the control system will output a positive or negative voltage to each DC motor in order to make the diwheel move as the user instructs.

Level 1 of the electrical design architecture describes the functionality of the hardware and its modules.



This level shows the flow of data and how the user input will be used to control the diwheel.

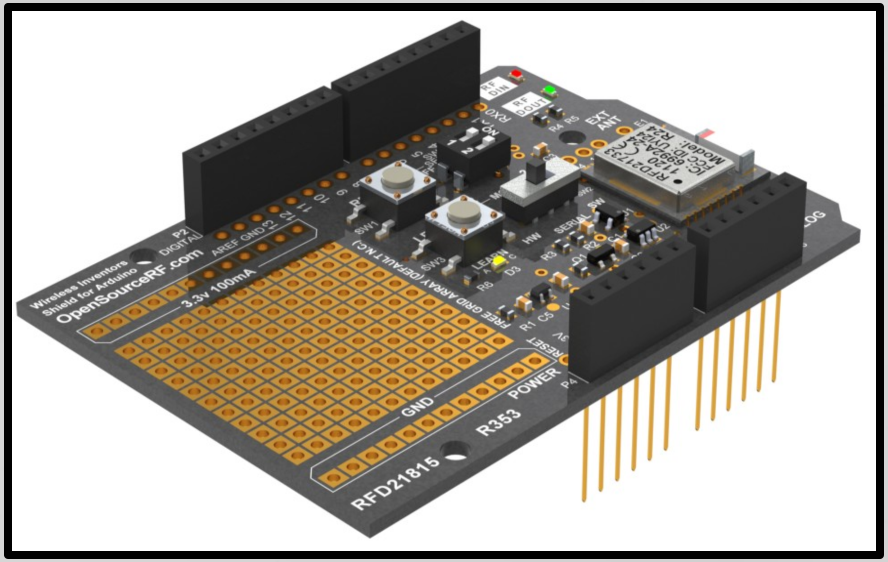

Wireless RF Dongle
(Transmitter)
PC
Wireless Inventors
Shield for Arduino
(Receiver)
Arduino Uno
IMU MPU6050
Breakout Board
10A Motor
Driver Arduino
Shield
Motor
Drivers

Wireless communication is utilized using the Wireless Inventors Shield for Arduino and the Wireless RF USB Dongle.

RFD21807

http://www.rfdigital.com

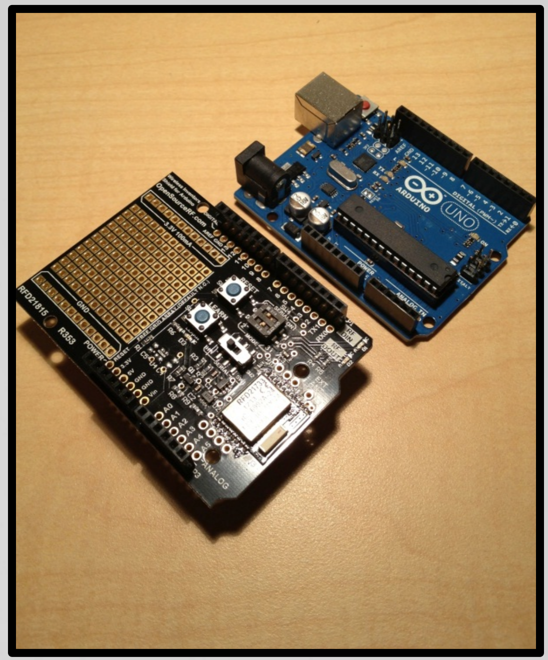
RFD21815

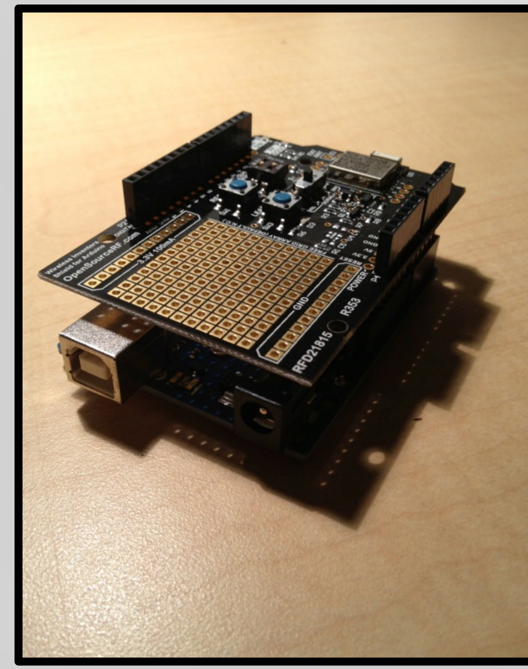


http://www.kickstarter.com

These components operate at 2.4GHz and have a range of up to 500ft making them ideal for our design. The frequency is within the IEEE standard for communication

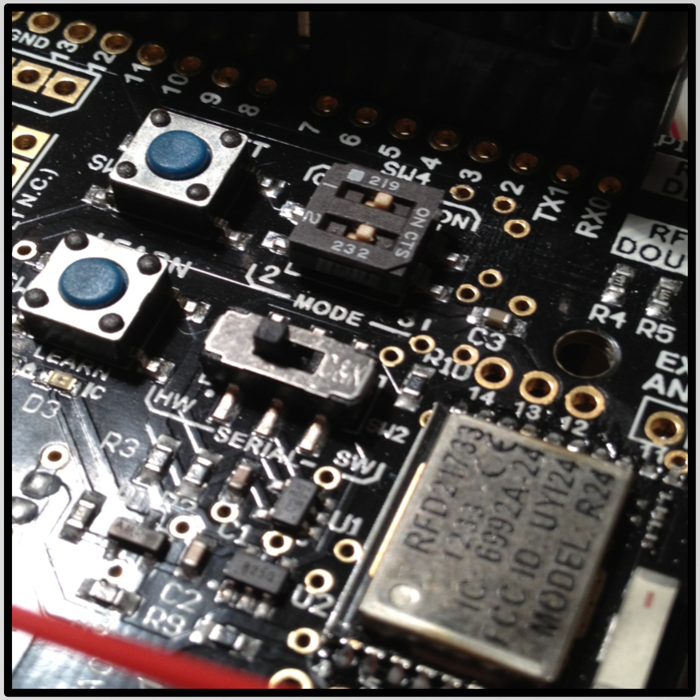
Prototyping of the wireless communication components was a quick and simple process.

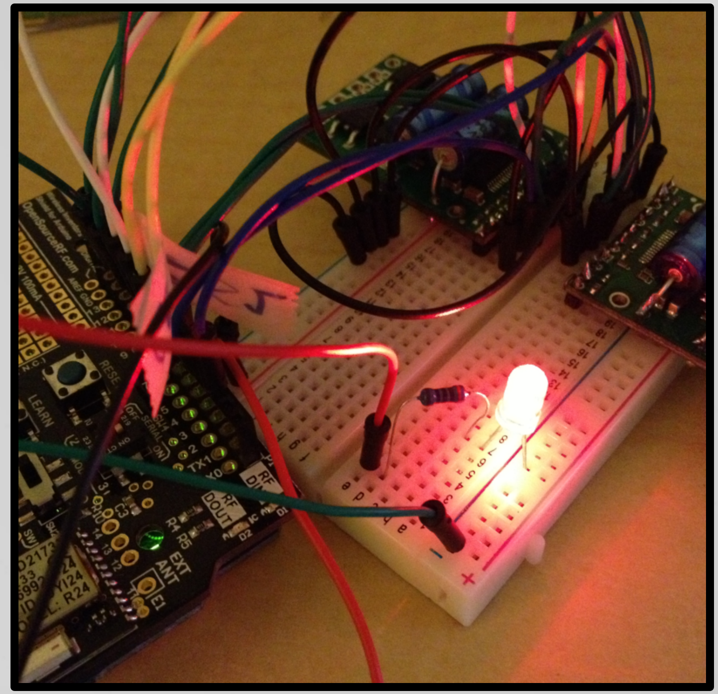




The shield was connected to the Arduino and the receiver and transmitter were synced together.

Testing involved a simple program which turned on an LED when a specific command was received.





Through testing, we learned that the wireless receiver needed to be on the hardware serial pin setting.

The range was tested by toggling the LED at distances up to 500 feet.

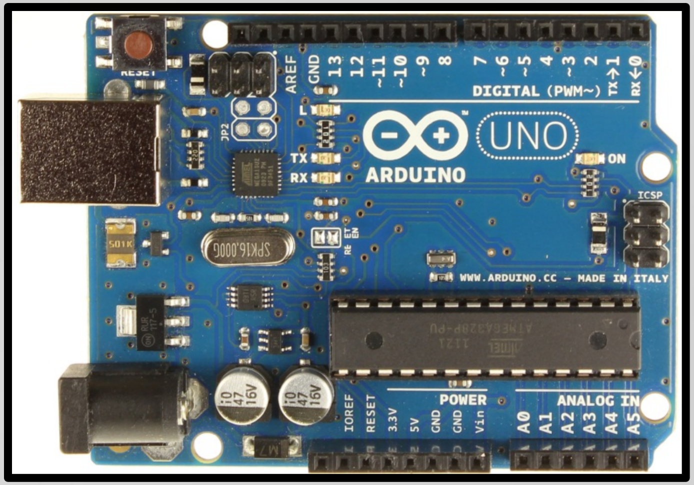
http://www.techgadgetscenter.com/

http://ledshoppe.com/

500 ft.

The movement and control of the diwheel is done through the Arduino Uno microcontroller.

Inputs



Outputs

http://www.arduino.cc

Module	Microcontroller
Inputs	User input from wireless receiver Inertia data from IMU sensor
Outputs	Voltage to control Motor Drivers
Functionality	The microcontroller will gather the input from the user as well as the feedback from the IMU sensor and determine what the needed voltage for the DC motors will be.

The Arduino will handle all inputs, outputs, and leveling control.

The Arduino itself did not require prototyping but was involved in the prototyping of most other components.

software_serial_loop | Arduino 1.0

```
//Wireless Inventors Shield Test (software)

#include <SoftwareSerial.h>

SoftwareSerial mySerial(10,11);

int incomingByte = 0;

void setup()
{
  mySerial.begin(9600);
}

void loop()
{
  if (mySerial.available() > 0)
  {
    incomingByte = mySerial.read();
    mySerial.print("I received: ");
    mySerial.write(incomingByte);
    //mySerial.print("/r/n");
  }
}
```

16 Arduino Uno on

Motor_Test | Arduino 1.0.1

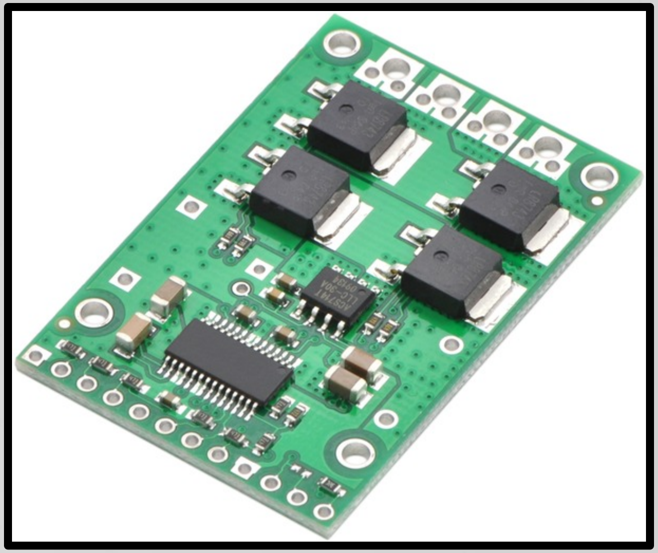
```
const int motor1Pin1 = 3; // H-bridge leg 1 (pin 1A)
const int motor1Pin2 = 4; // H-bridge leg 2 (pin 2A)
const int motor2Pin1 = 5; // H-bridge leg 3 (pin 3A)
const int motor2Pin2 = 6; // H-bridge leg 4 (pin 4A)
const int enable1Pin = 8; // H-bridge enable pin
const int enable2Pin = 9; // H-bridge enable pin
int incomingByte = 0;

void setup() {
  Serial.begin(9600);
  pinMode(motor1Pin1, OUTPUT);
  pinMode(motor1Pin2, OUTPUT);
  pinMode(enable1Pin, OUTPUT);
  pinMode(motor2Pin1, OUTPUT);
  pinMode(motor2Pin2, OUTPUT);
  pinMode(enable2Pin, OUTPUT);

  digitalWrite(enable1Pin, LOW);
  digitalWrite(enable2Pin, LOW);
}

  void loop() {
     if (Serial.available() > 0)
     {
        incomingByte = Serial.read();
        switch(incomingByte) {

          case 'w': //forwards
             digitalWrite(enable1Pin, HIGH);
             digitalWrite(motor1Pin1, HIGH);
             digitalWrite(motor1Pin2, LOW);

             digitalWrite(enable2Pin, HIGH);
             digitalWrite(motor2Pin1, HIGH);
             digitalWrite(motor2Pin2, LOW);
             break;

          case 's': //backwards
             digitalWrite(enable1Pin, HIGH);
             digitalWrite(motor1Pin1, LOW);
             digitalWrite(motor1Pin2, HIGH);
```

hardware_serial_loop | Arduino 1.0.1

```
int incomingByte = 0;
int brightLED = 0;
int outputValue = 0;
const int analogOutPin = 2;

void setup()
{
  Serial.begin(9600);
  pinMode(8, OUTPUT);
  pinMode(7, OUTPUT);
}

void loop()
{
  if (Serial.available() > 0)
  {
    incomingByte = Serial.read();
    switch(incomingByte) {
      case 'w':
        digitalWrite(7, HIGH);
        digitalWrite(8, HIGH);
        break;
      case 's':
        digitalWrite(7, HIGH);
        digitalWrite(8, HIGH);
        break;
      case 'a':
        digitalWrite(7, LOW);
        digitalWrite(8, HIGH);
        break;
      case 'd':
        digitalWrite(7, HIGH);
        digitalWrite(8, LOW);
        break;
      case 'x':
        digitalWrite(7, LOW);
        digitalWrite(8, LOW);
      default:
        break;
    }
  }
}
```

19 Arduino Uno on /dev/cu.usbmodem621

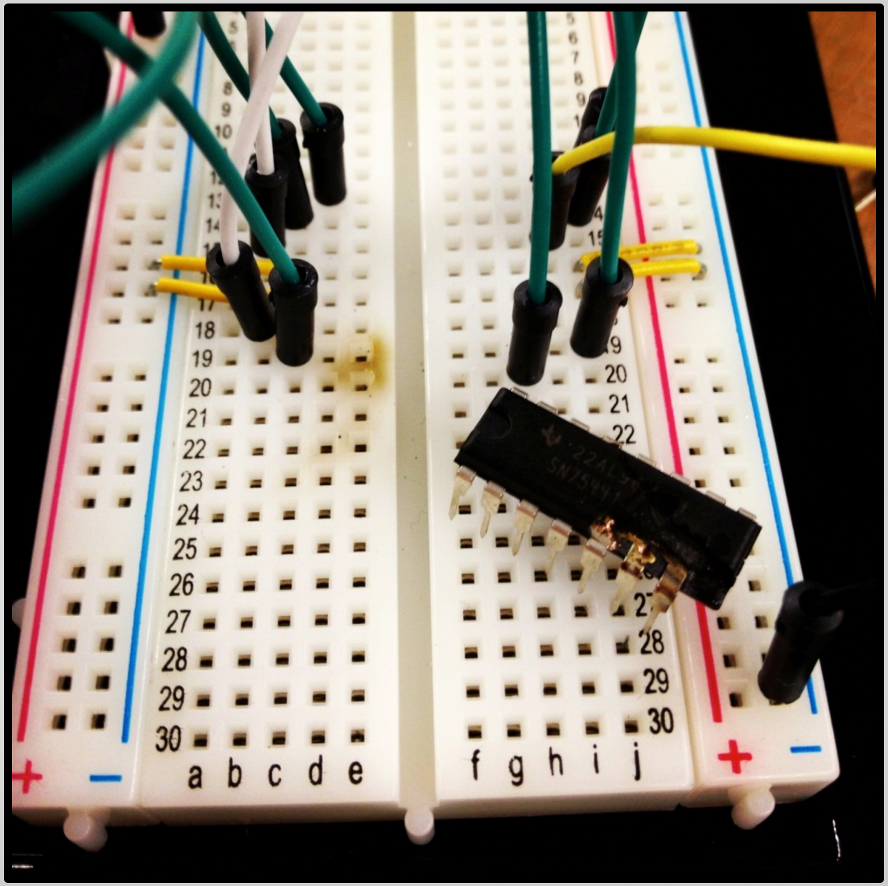
In order to control the speed and direction of each motor, high power motor driver circuits are used.

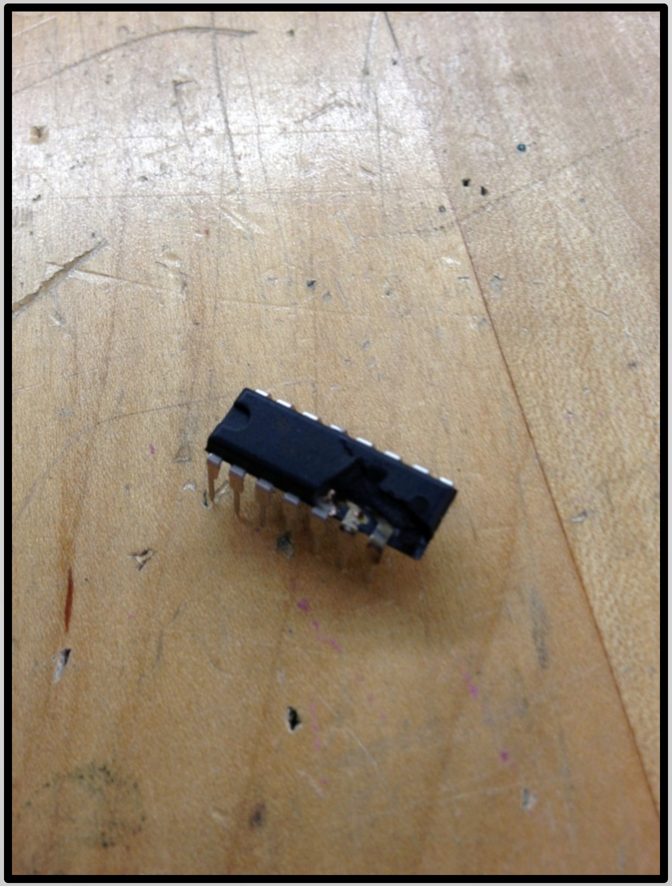


http://www.pololu.com/catalog/product/1456

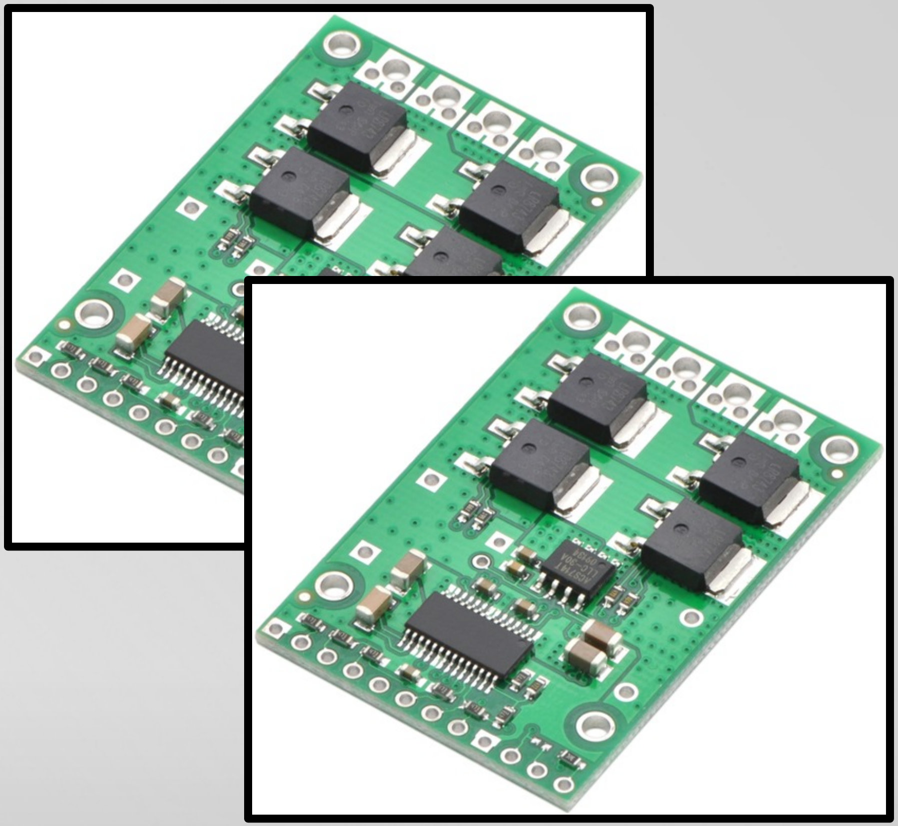
Module	Motor Driver (250W Motors)
Inputs	PWM signal from Microcontroller Direction signal from Microcontroller 24V From Batteries
Outputs	Voltage to control speed and direction of DC motors
Functionality	The motor driver will take in the output voltage of the microcontroller and output the correct polarity to the DC motors. Controls the DC motors to go forwards or backwards. The PWM signal controls the average voltage allowed to pass through to the motors which controls the speed.

Our initial design used an H-Bridge IC chip to control the motors but was not suited for the high-power motors.



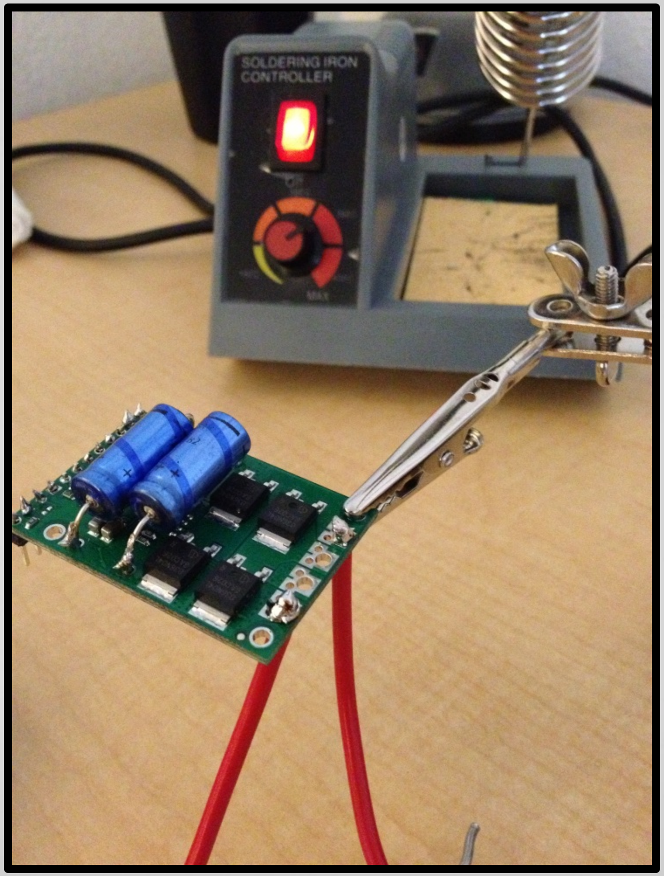


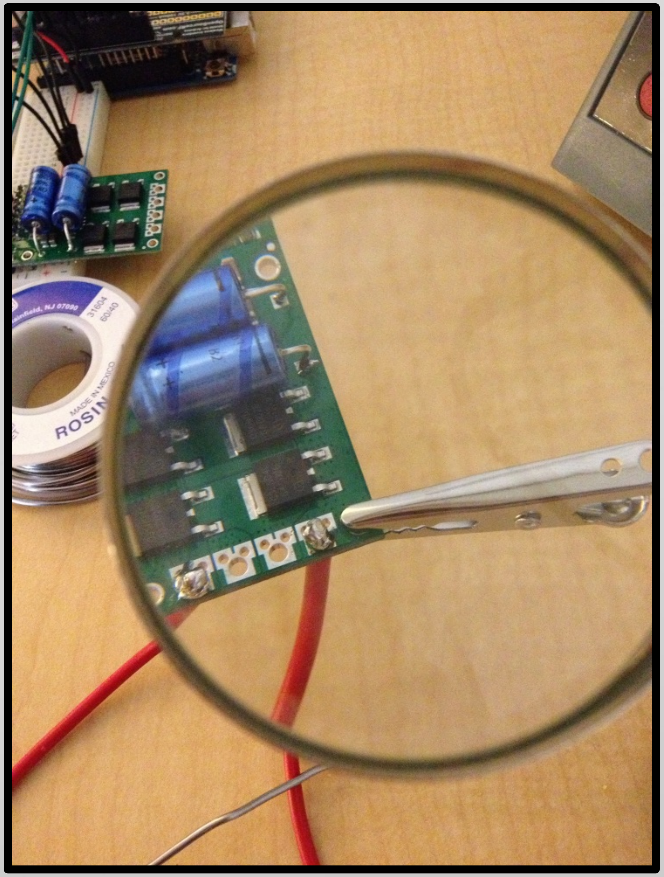
The design was updated to use two high-powered motor drivers that could handle the current draw of the motors.



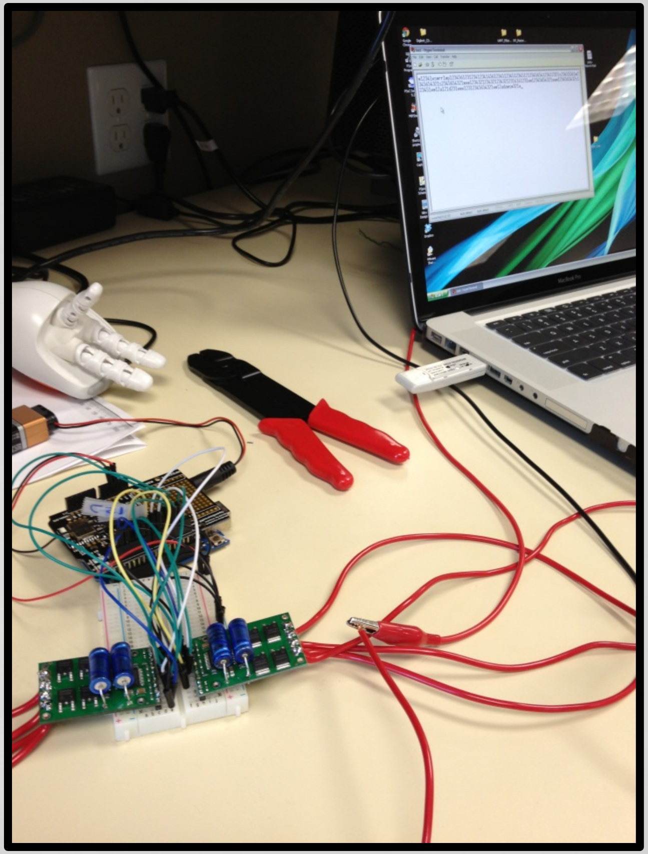
The Pololu High-Powered motor driver is capable of handling current up to 23A as well as Voltages from 5.5V to 40V

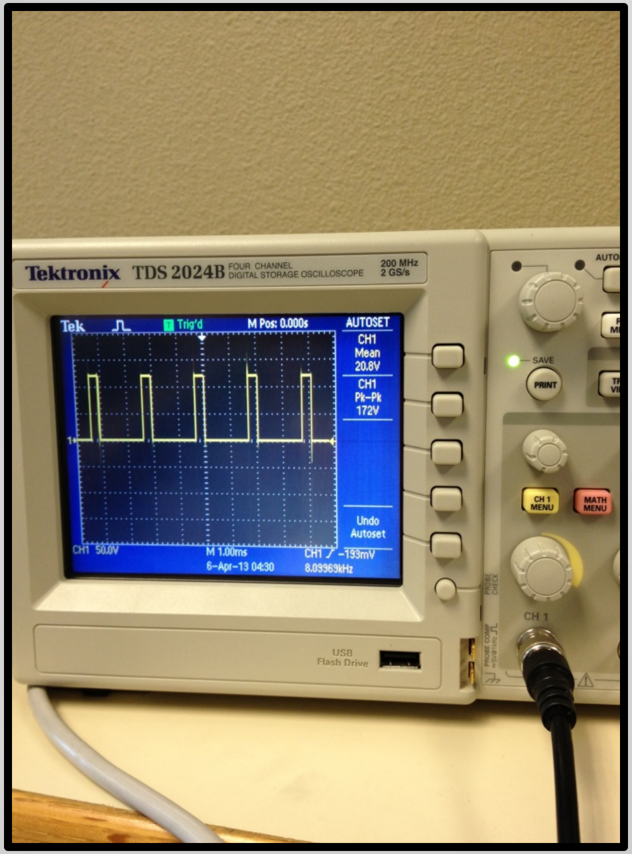
Prototyping of the motor drivers involved soldering the headers and wiring the connections.





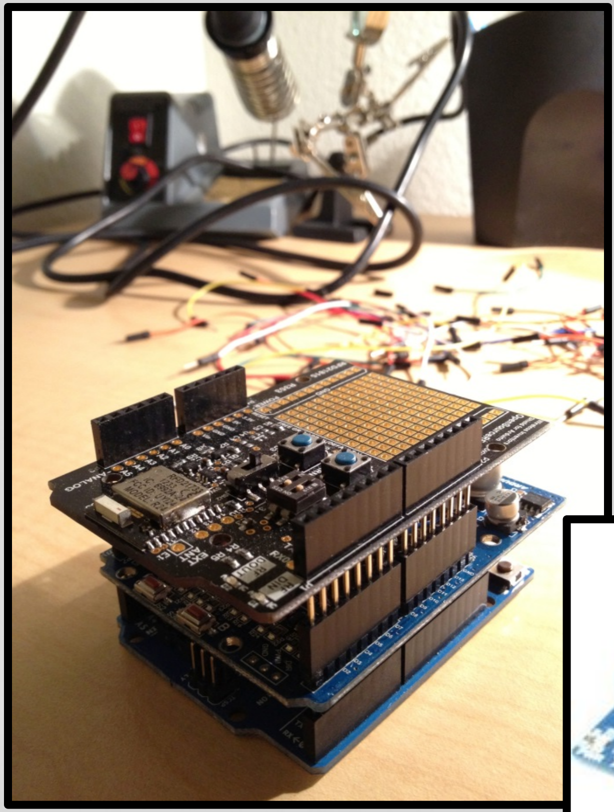
Before connecting the motor drivers to the batteries and motors, a “no-load” test was done using oscilloscopes.

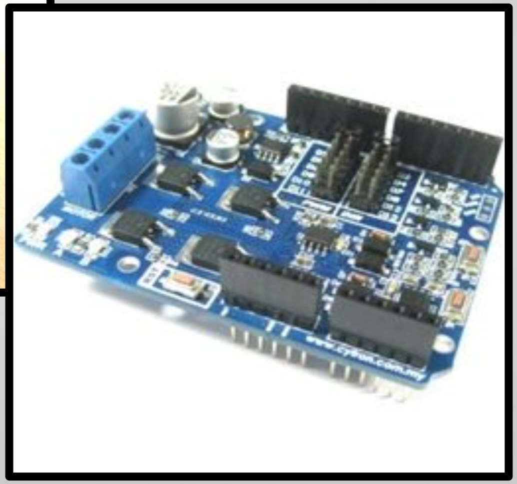




The oscilloscope showed the correct PWM signal
for the value given

A third motor driver circuit is used to control the leveling system motor.

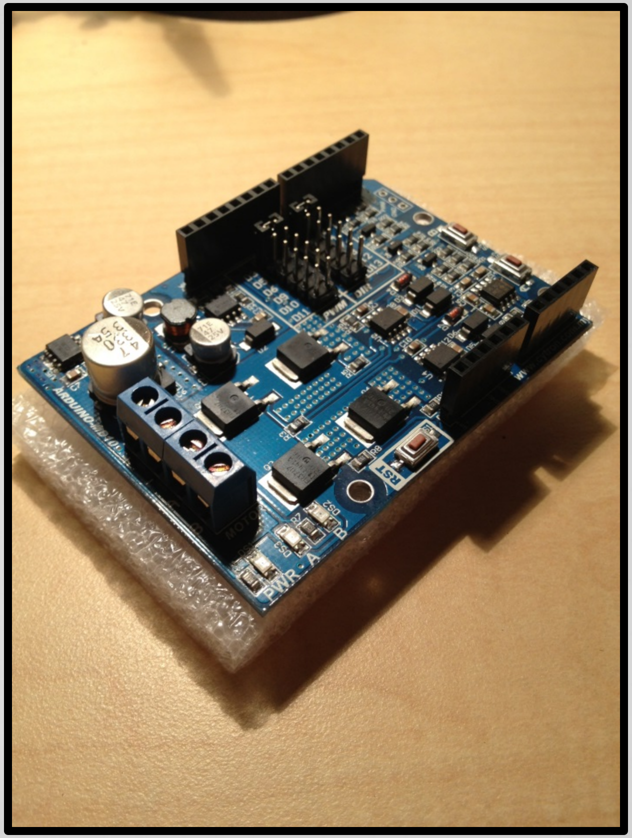


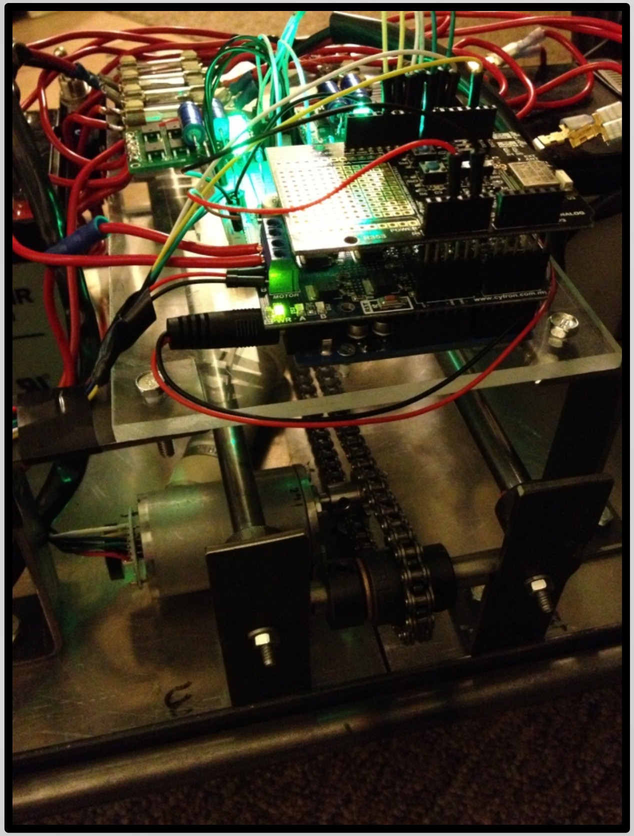


http://www.robotshop.com/

- The 10A DC Motor Arduino Shield was chosen to save space and wiring
- This creates a triple stack for the Arduino

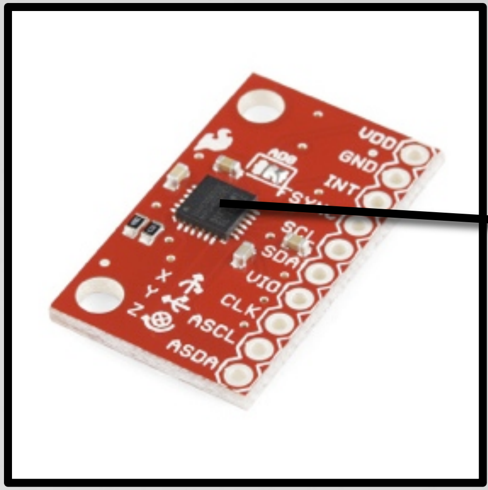
The motor driver has on board test buttons which helped speed up testing.



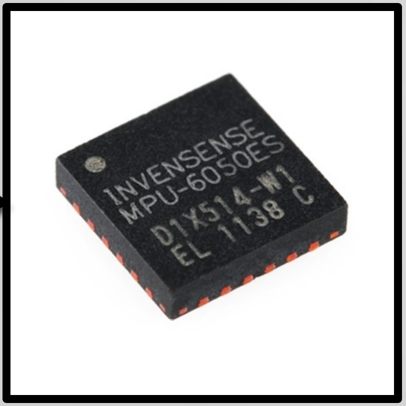


Testing was successfully done after the linear weight was mounted

The Inertia Measurement Unit (IMU) is used to gather data from the movement of the diwheel using the built in Accelerometers and Gyroscopes.



www.sparkfun.com

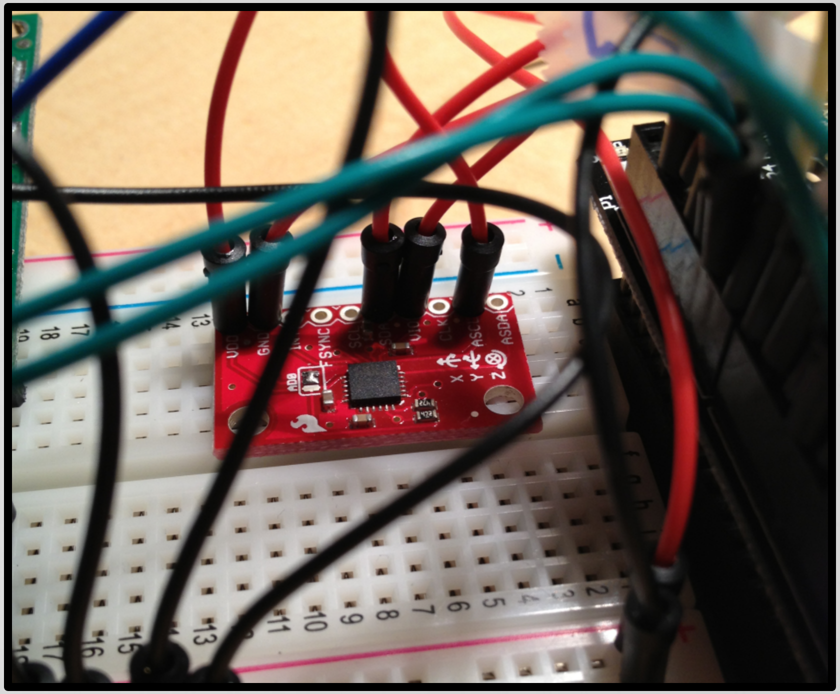


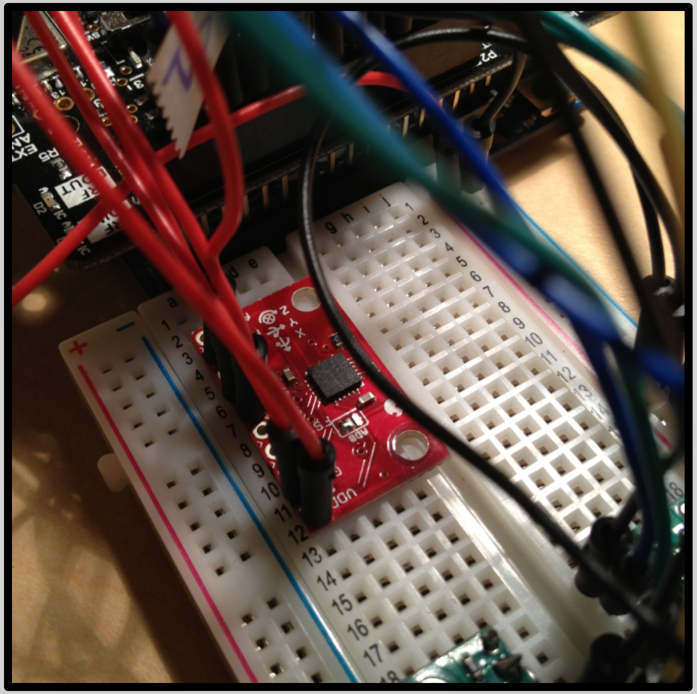
www.sparkfun.com

Module	Inertial Measurement Unit
Inputs	Inertial forces acting on device
Outputs	Data describing current status of the diwheel's inertia
Functionality	The accelerometers and gyroscopes inside the IMU will provide data describing the angle, pitch, and momentum of the device.

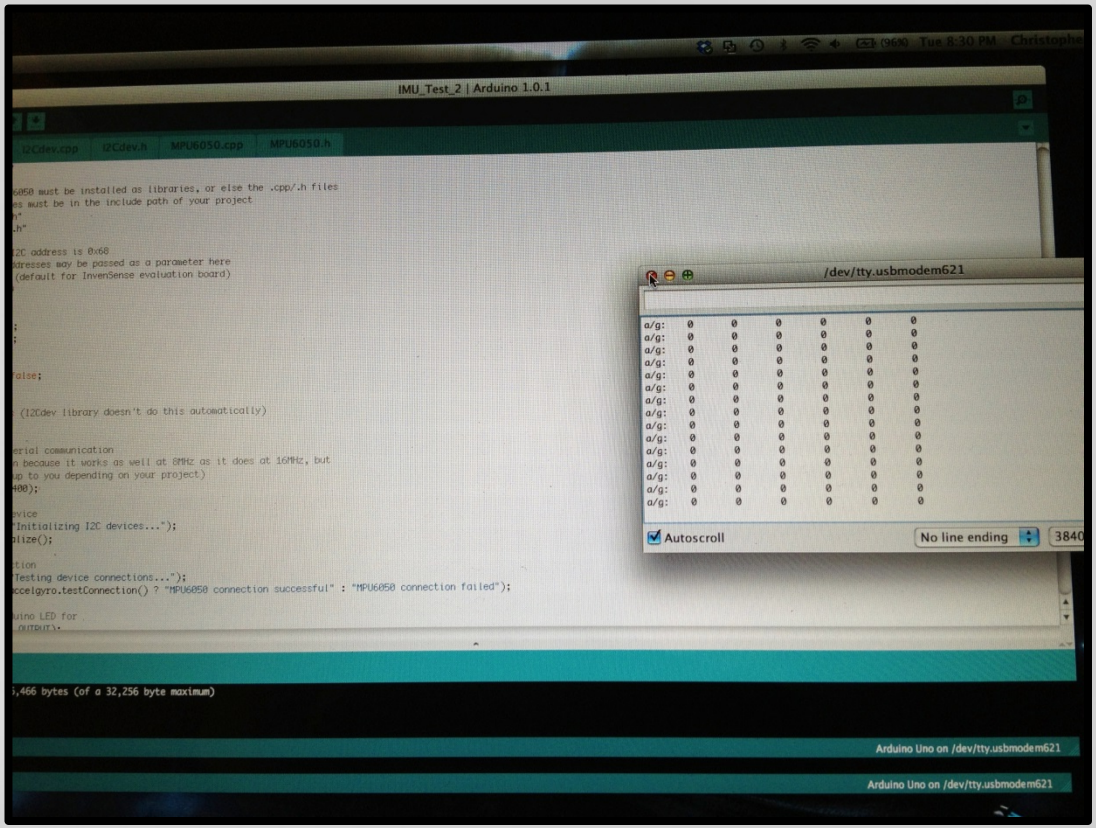
Our design uses the Triple Axis Accelerometer & Gyro breakout board MPU-650 for our IMU

Prototyping involved wiring the breakout board to the breadboard and making the connections to the Arduino.



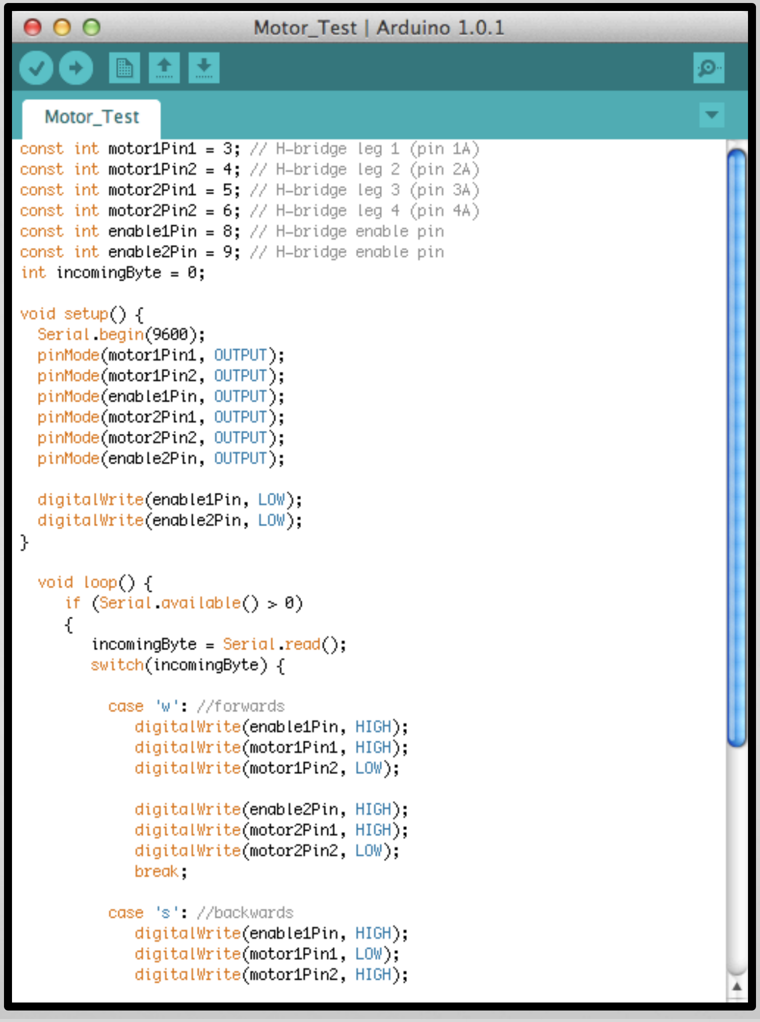


Initial testing was successful but a later attempt yielded no results.



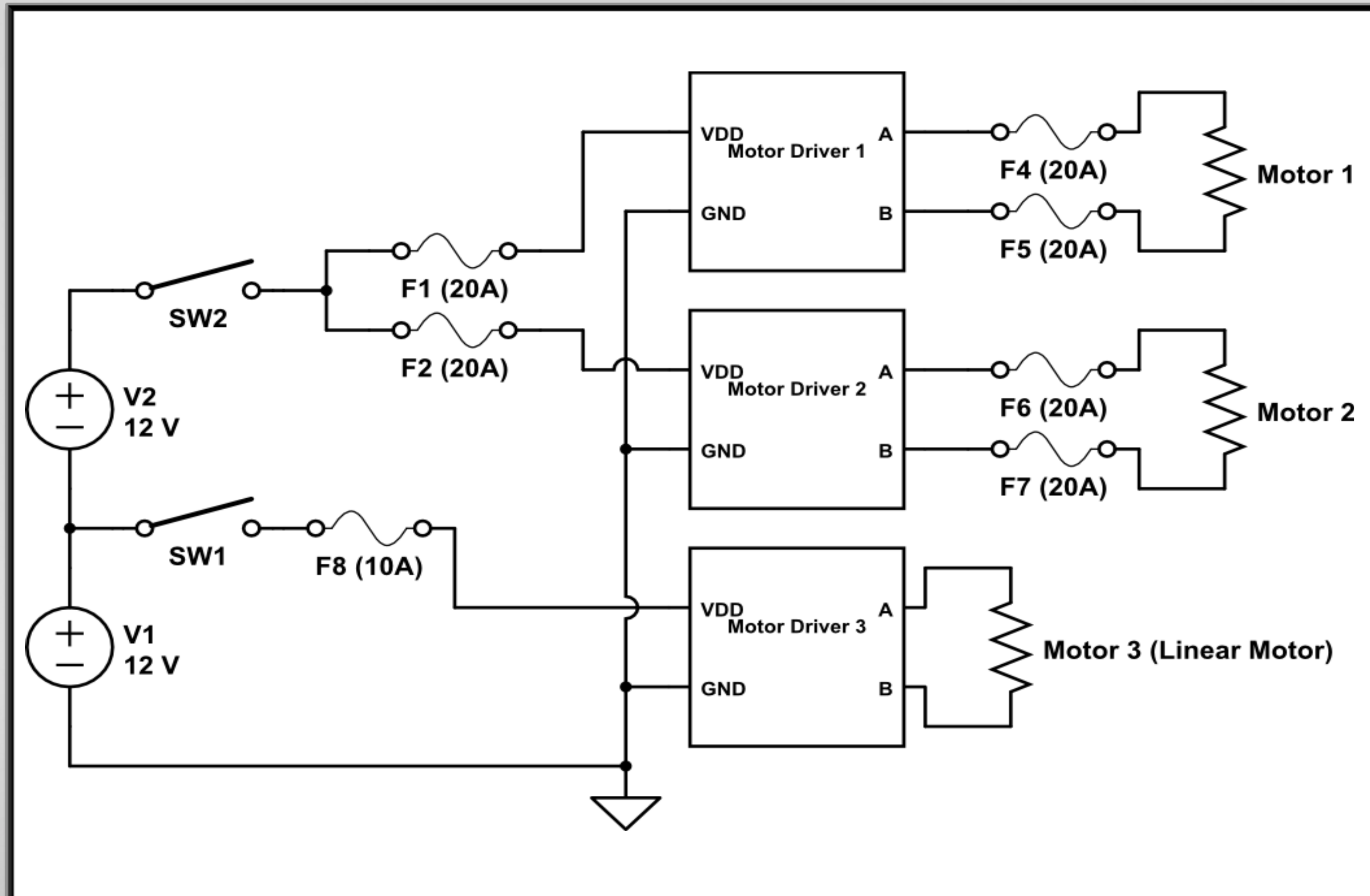
The same test was repeated but data was no longer being outputted.

Software prototyping was done alongside the hardware prototyping and involved writing the Arduino code.

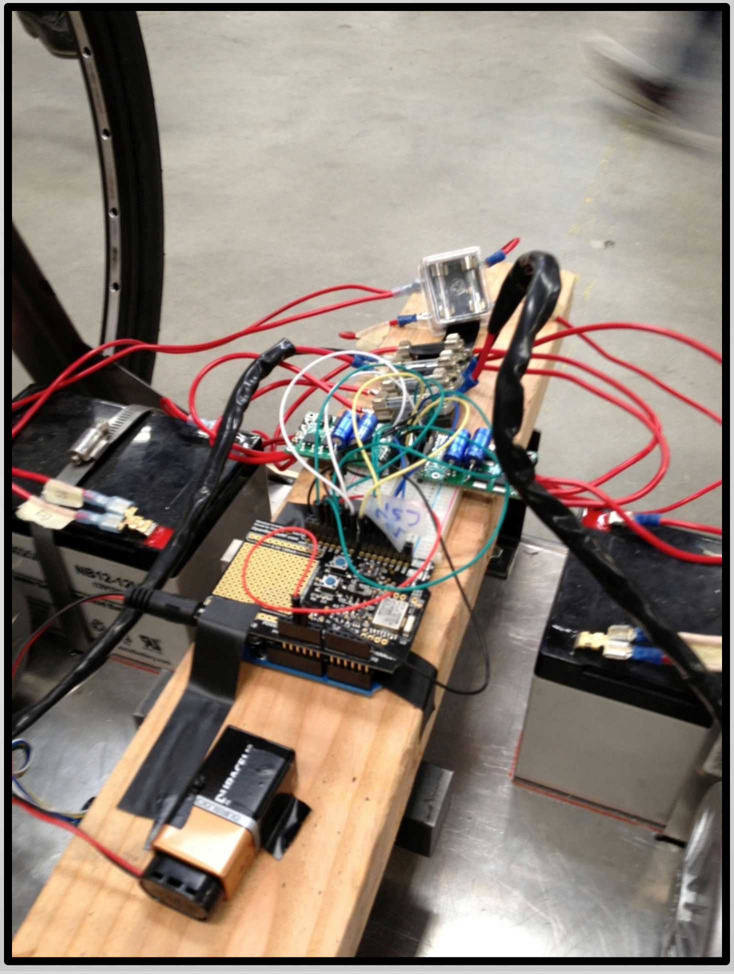


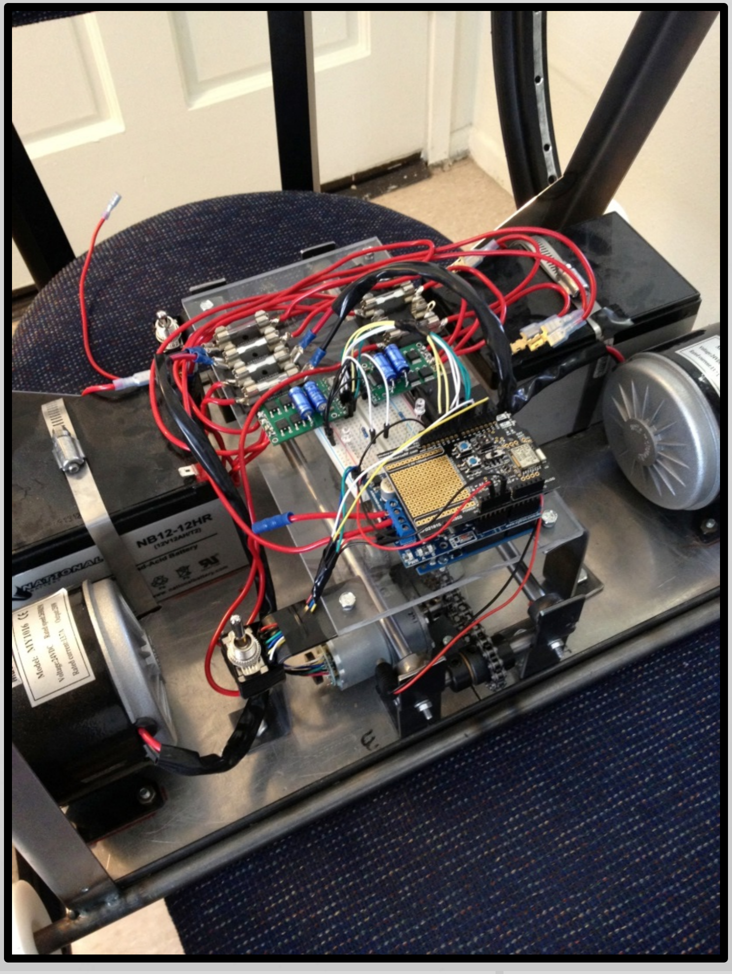

```
const int motor1Pin1 = 3; // H-bridge leg 1 (pin 1A)
const int motor1Pin2 = 4; // H-bridge leg 2 (pin 2A)
const int motor2Pin1 = 5; // H-bridge leg 3 (pin 3A)
const int motor2Pin2 = 6; // H-bridge leg 4 (pin 4A)
const int enable1Pin = 8; // H-bridge enable pin
const int enable2Pin = 9; // H-bridge enable pin
int incomingByte = 0;

void setup() {
  Serial.begin(9600);
  pinMode(motor1Pin1, OUTPUT);
  pinMode(motor1Pin2, OUTPUT);
  pinMode(enable1Pin, OUTPUT);
  pinMode(motor2Pin1, OUTPUT);
  pinMode(motor2Pin2, OUTPUT);
  pinMode(enable2Pin, OUTPUT);

  digitalWrite(enable1Pin, LOW);
  digitalWrite(enable2Pin, LOW);
}

  void loop() {
     if (Serial.available() > 0)
     {
        incomingByte = Serial.read();
        switch(incomingByte) {

          case 'w': //forwards
             digitalWrite(enable1Pin, HIGH);
             digitalWrite(motor1Pin1, HIGH);
             digitalWrite(motor1Pin2, LOW);

             digitalWrite(enable2Pin, HIGH);
             digitalWrite(motor2Pin1, HIGH);
             digitalWrite(motor2Pin2, LOW);
             break;

          case 's': //backwards
             digitalWrite(enable1Pin, HIGH);
             digitalWrite(motor1Pin1, LOW);
             digitalWrite(motor1Pin2, HIGH);
```

The first step of the electrical system prototyping was to finalize the circuit and connections.

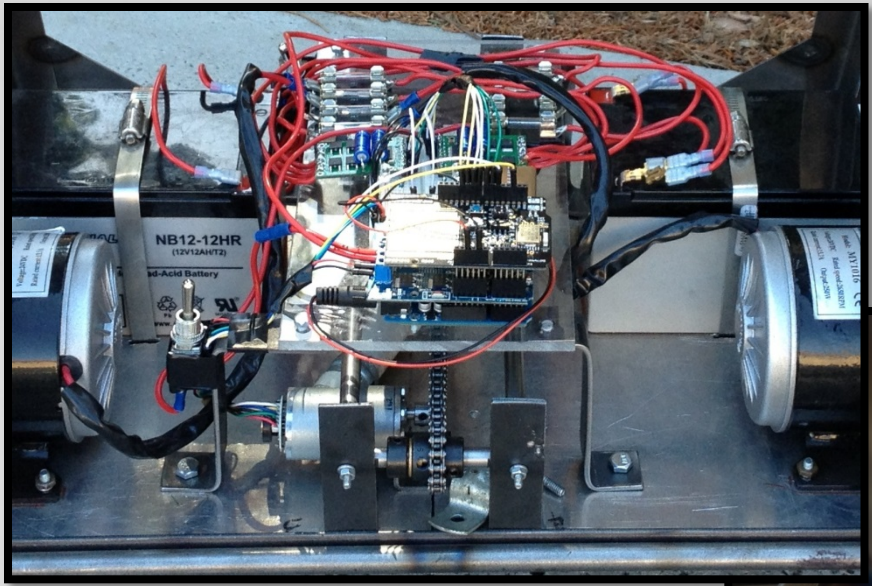


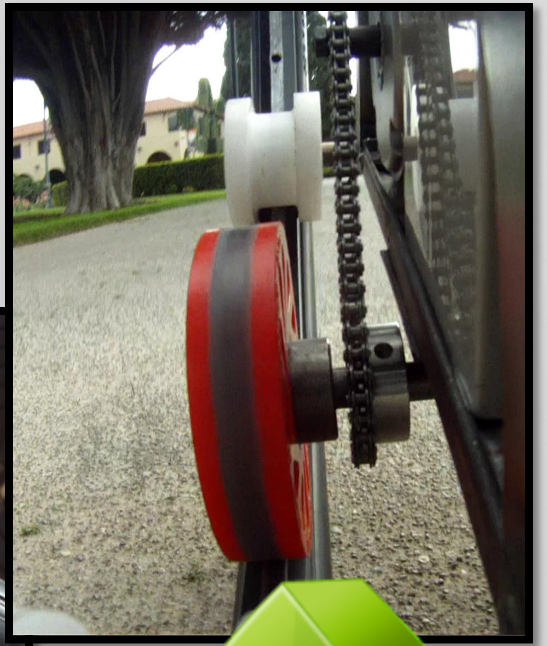
Once the connections were mapped out, the physical wiring could be completed and system testing could begin.





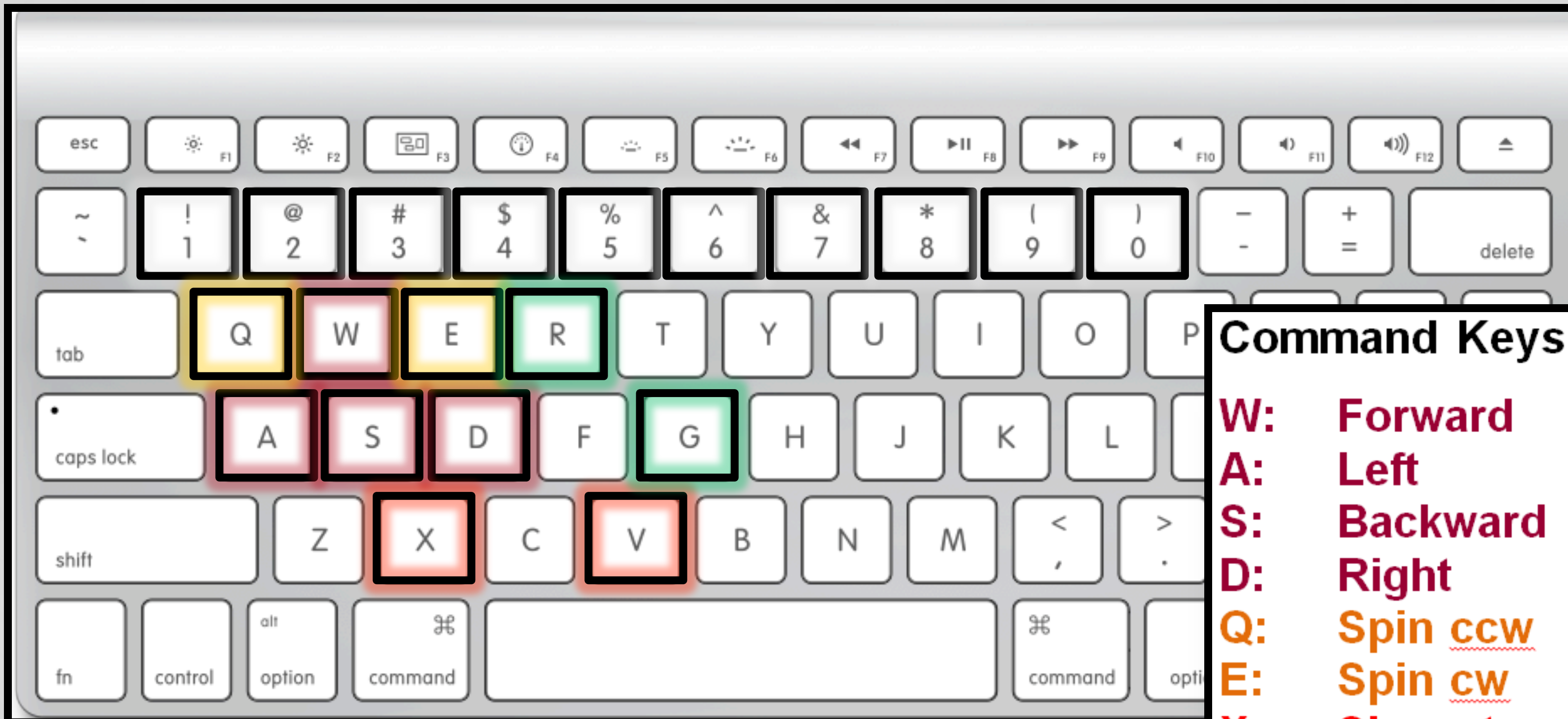
The mechanical and electrical systems were successfully integrated together.





The two systems worked together and the diwheel could be tested for drivability.

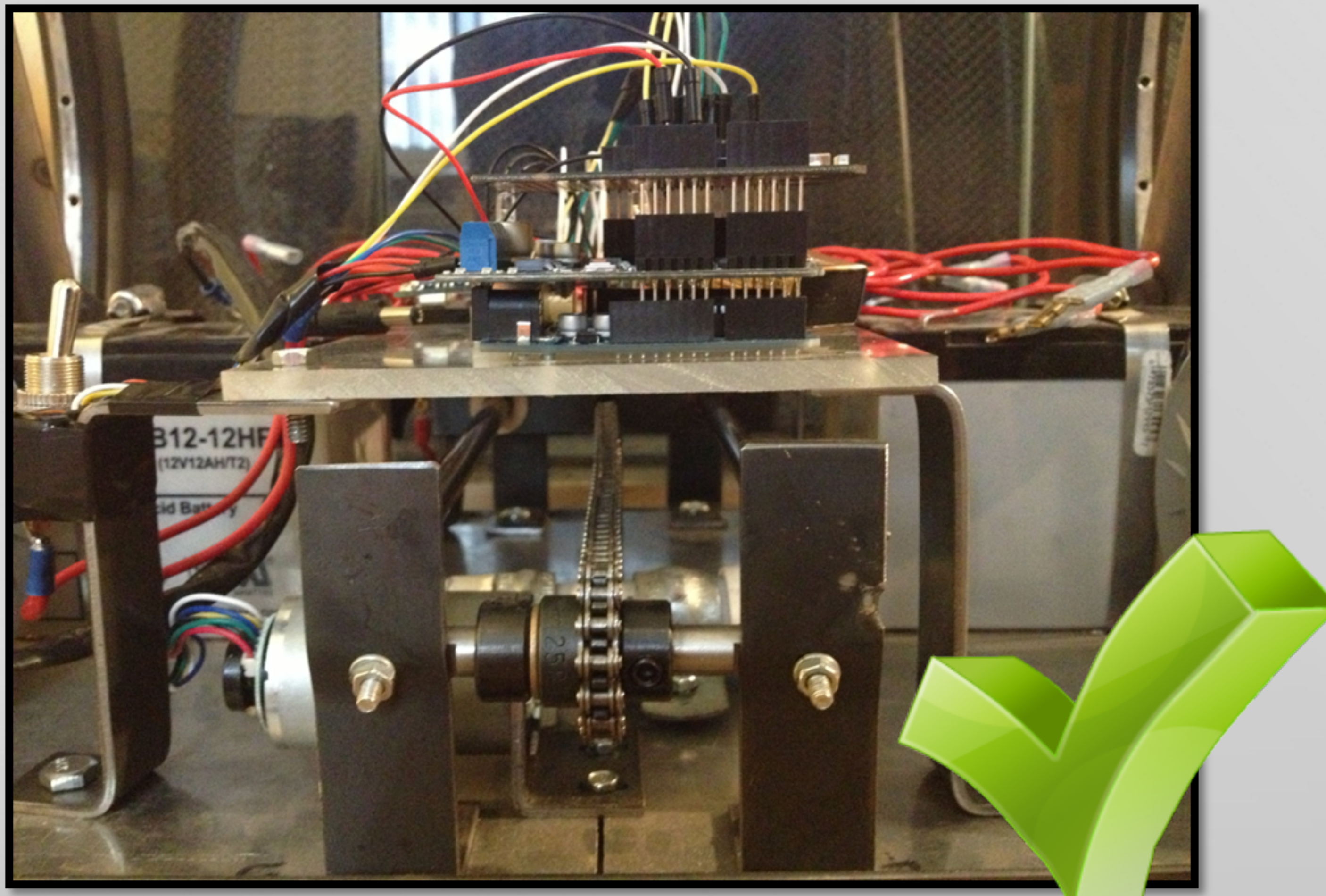
The controls were programmed to make operation of the diwheel easy and user friendly.



Command Keys:

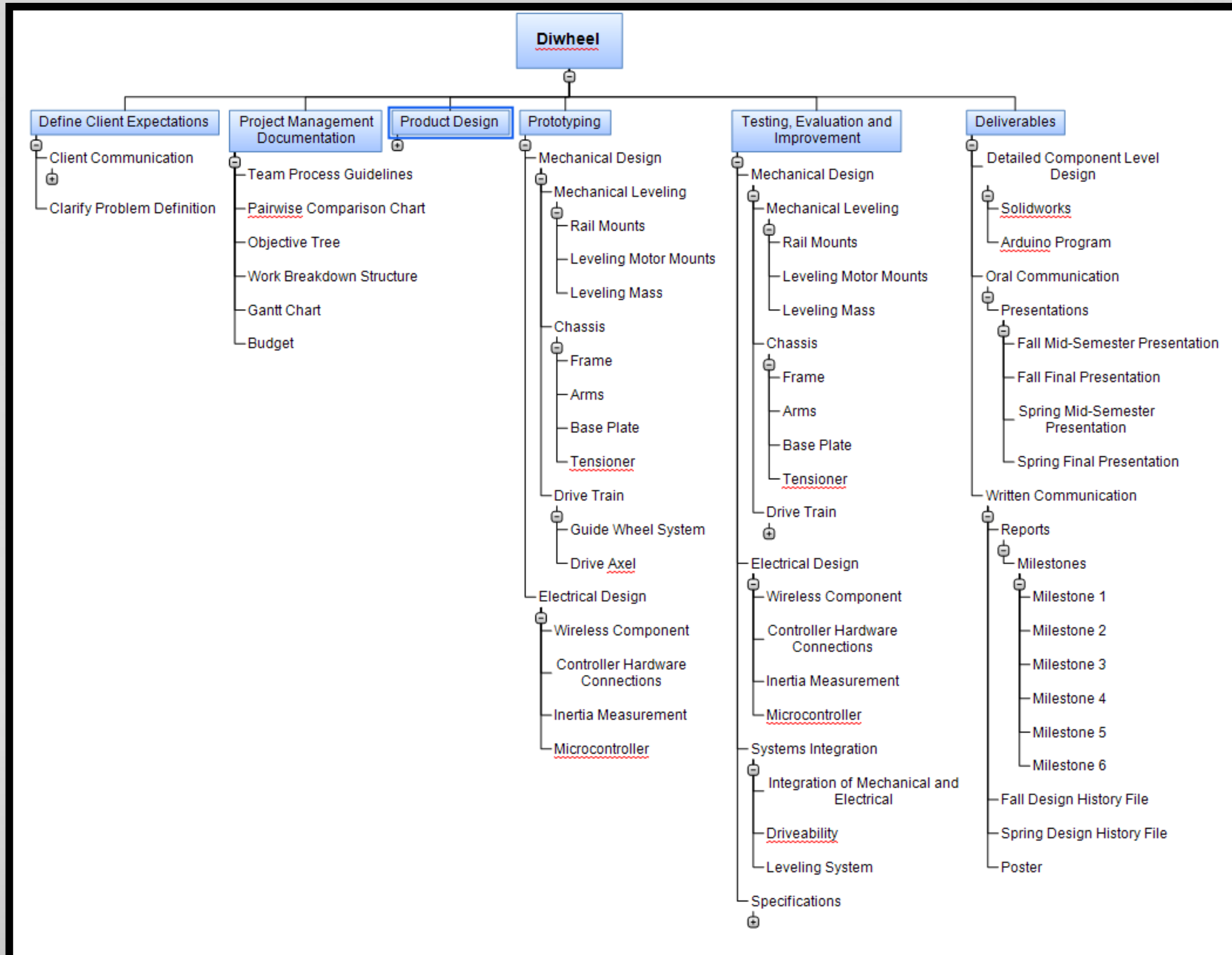
W: Forward
A: Left
S: Backward
D: Right
Q: Spin ccw
E: Spin cw
X: Slow stop
V: Emergency stop
G: Clear system to go
R: Rest system
1-0: Low to high speeds

Once it was determined that the diwheel could function properly the leveling motor could be tested for functionality.



The diwheel was then tested to see if the design specifications were met.

The Work Breakdown Structure (WBS) illustrates the steps needed to accomplish the project.



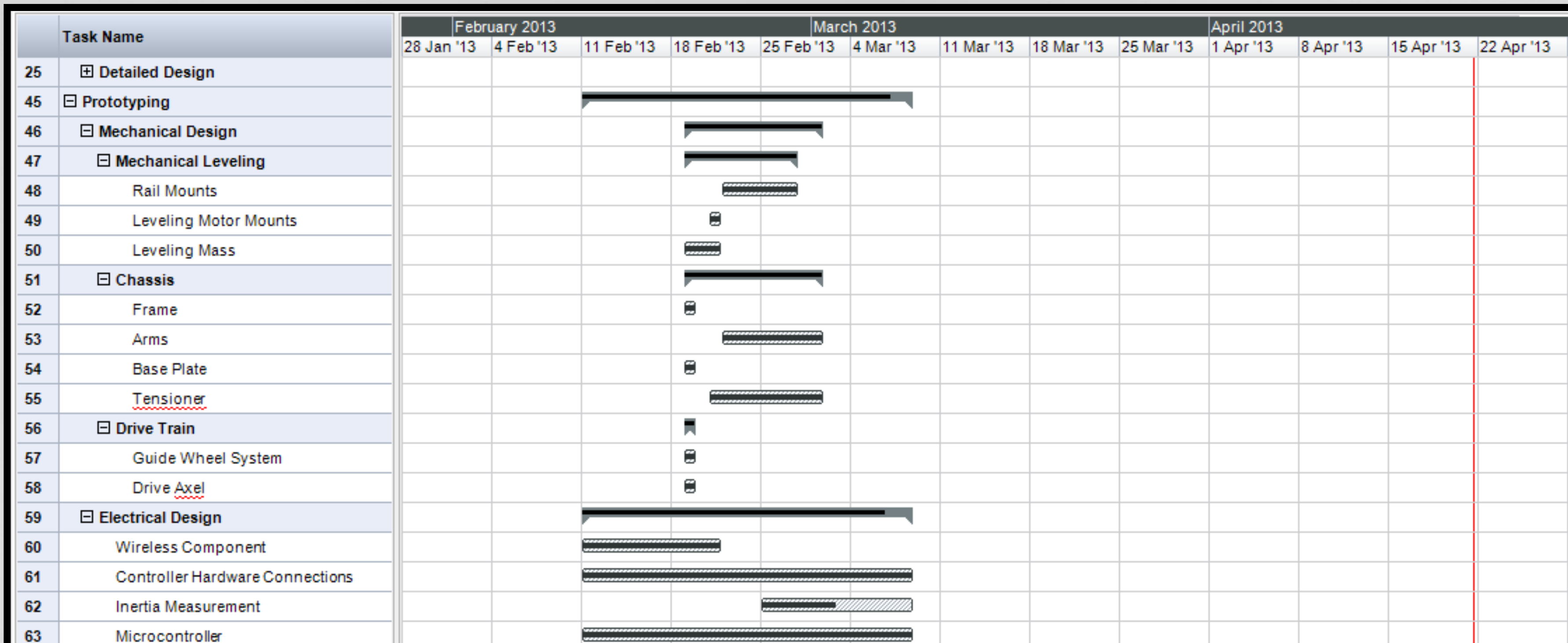
The Work Breakdown Structure (WBS) illustrates the steps needed to accomplish the project.

- Prototyping
  - Mechanical Design
    - Mechanical Leveling
      - Rail Mounts
      - Leveling Motor Mounts
      - Leveling Mass
    - Chassis
      - Frame
      - Arms
      - Base Plate
      - Tensioner
    - Drive Train
      - Guide Wheel System
      - Drive Axel
  - Electrical Design
    - Wireless Component
    - Controller Hardware Connections
    - Inertia Measurement
    - Microcontroller

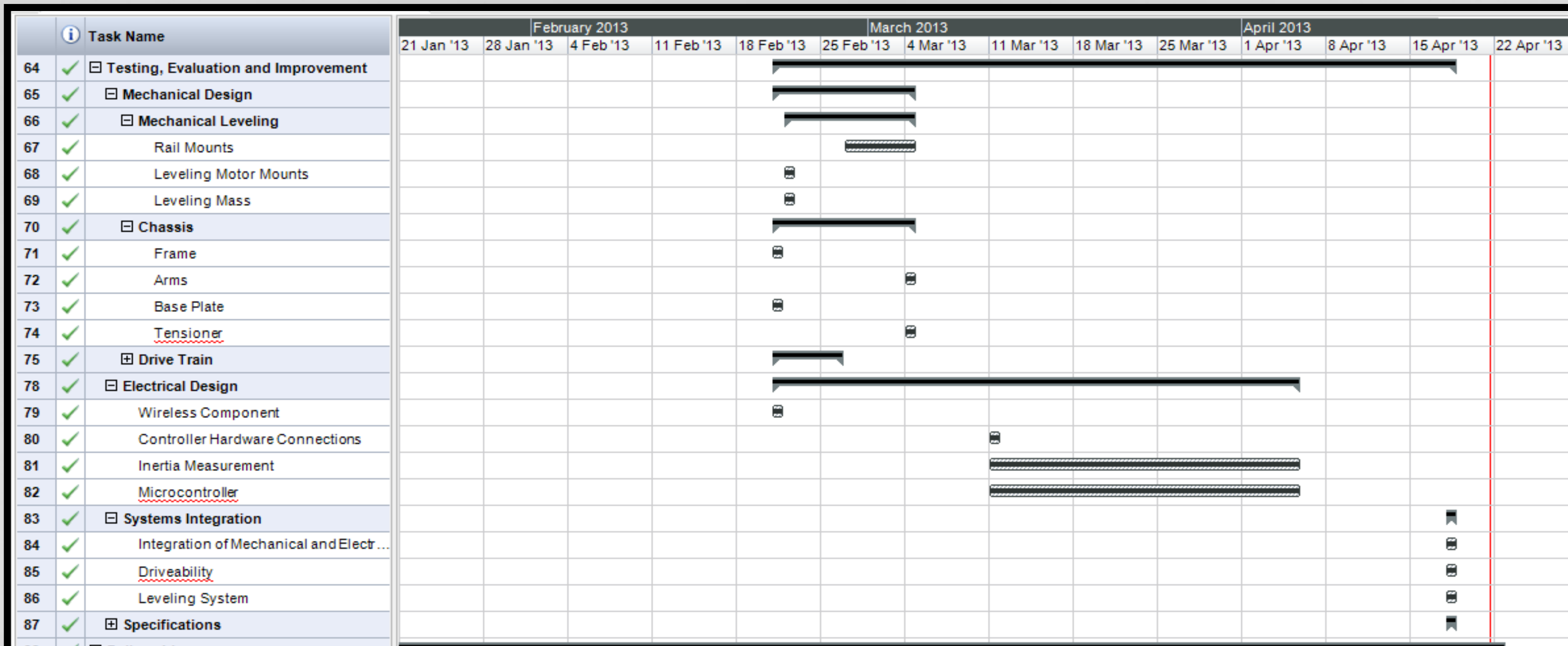
- Testing, Evaluation and Improvement
  - Mechanical Design
    - Mechanical Leveling
      - Rail Mounts
      - Leveling Motor Mounts
      - Leveling Mass
    - Chassis
      - Frame
      - Arms
      - Base Plate
      - Tensioner
    - Drive Train
  - Electrical Design
    - Wireless Component
    - Controller Hardware Connections
    - Inertia Measurement
    - Microcontroller
  - Systems Integration
    - Integration of Mechanical and Electrical
    - Driveability
    - Leveling System
  - Specifications

- Deliverables
  - Detailed Component Level Design
    - Solidworks
    - Arduino Program
  - Oral Communication
    - Presentations
      - Fall Mid-Semester Presentation
      - Fall Final Presentation
      - Spring Mid-Semester Presentation
      - Spring Final Presentation
  - Written Communication
    - Reports
      - Milestones
        - Milestone 1
        - Milestone 2
        - Milestone 3
        - Milestone 4
        - Milestone 5
        - Milestone 6
    - Fall Design History File
    - Spring Design History File
    - Poster

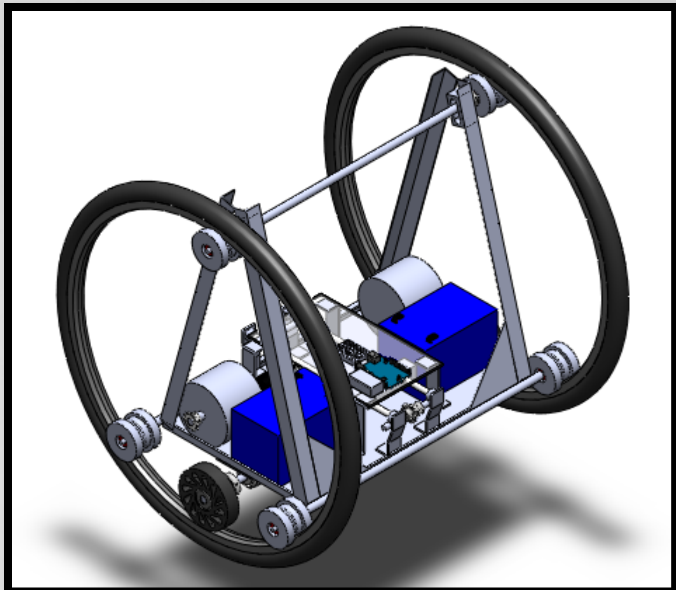
The Gantt Chart shows the duration and deadline of each step of the WBS.



The Gantt Chart shows the duration and deadline of each step of the WBS.



Our project has been to design a diwheel with a mounted hardware on its chassis that can be kept level while in motion.

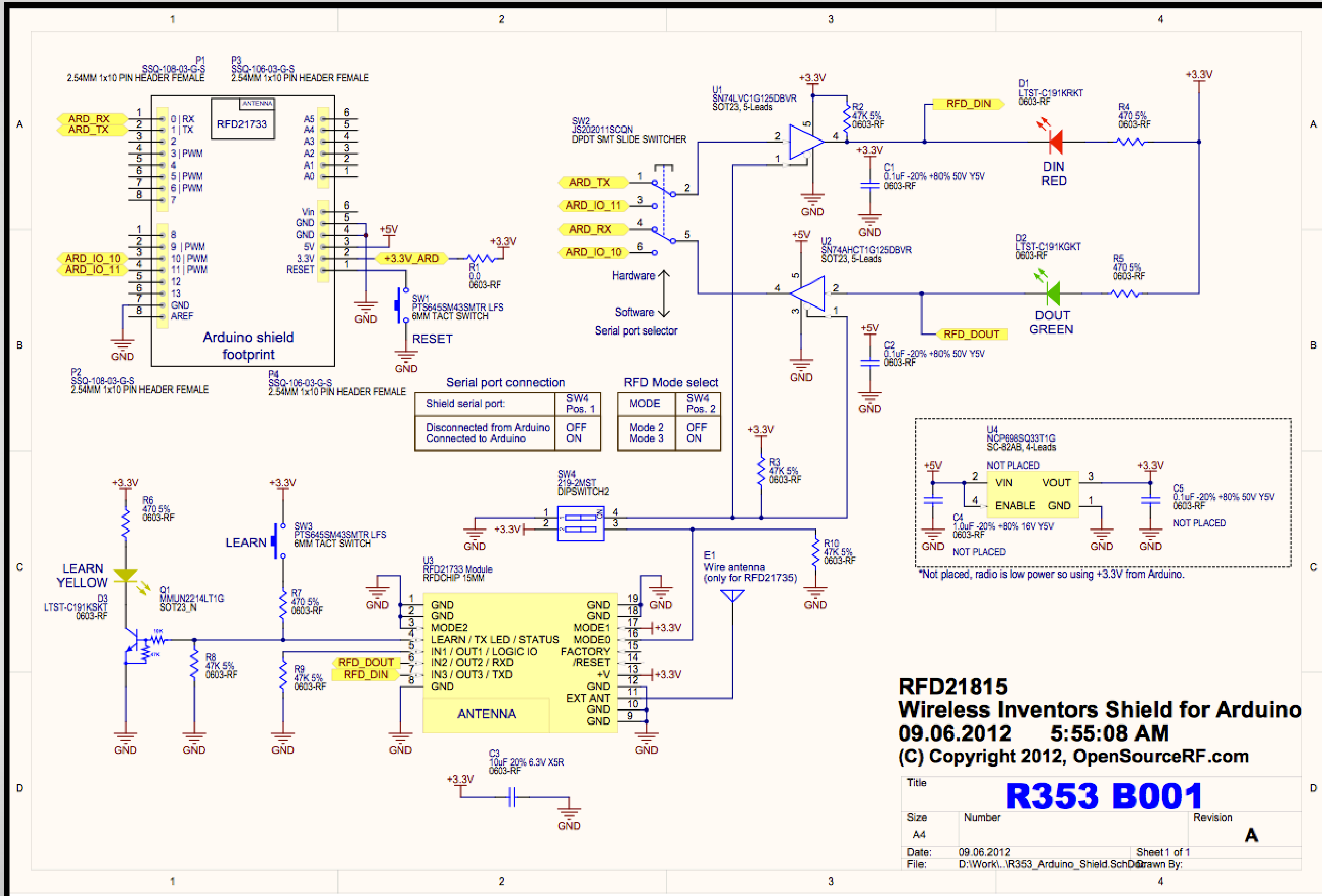


http://www.google.com

http://www.google.com

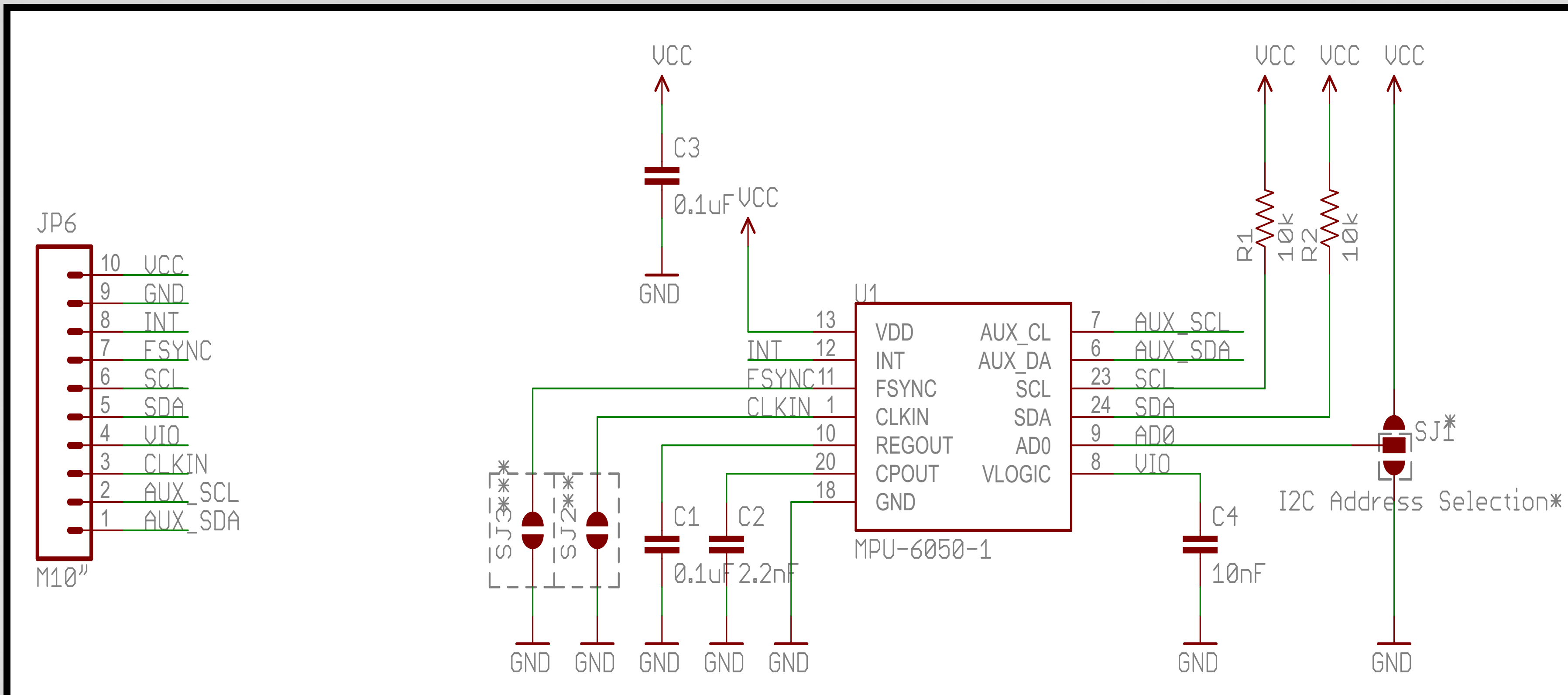
Backup Slides

Term	Variable	Value	Units
Mass of body	m_b	18.000	kg
Radius of Wheels	R	0.305	m
Radius of Mass	e	0.127	m
Max Pitch Allowed	ϴ	10.000	deg
Max Pitch Allowed	ϴ	0.175	rad
Max Torque	T_m	7.480	Nm
Max Displacement of	l_max	0.110	m
Mass of Slider	m_s	1.000	kg
Max Linear Acceleration	a	1.217	m/s^2
Max Angular Acceleration	$\ddot{\phi}$	3.992	rad/s^2
Gravitational constant	a_g = m_b*e*g	22.426	
Cross Coupling term	a_x = m_b*e*R	0.697	
Displacement of slider at max acceleration, 1.2166 m/s^2	l_acc	0.071	m
Displacement of slider at constant velocity, 7.3 m/s	l_vel	0.045	m

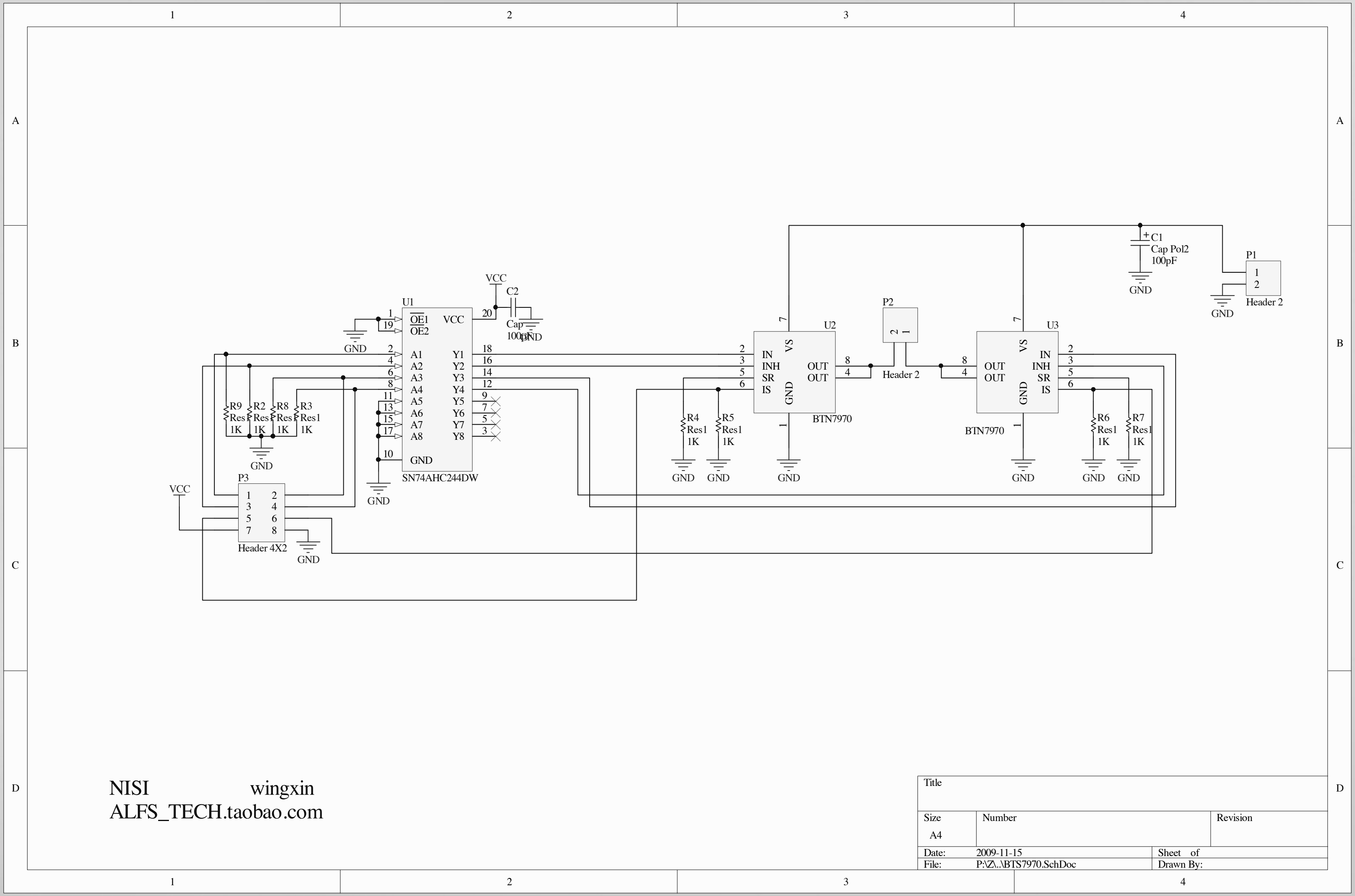


P1
SSQ-108-03-G-S
2.54MM 1x10 PIN HEADER FEMALE
P3
SSQ-106-03-G-S
2.54MM 1x10 PIN HEADER FEMALE
ARD_RX
ARD_TX
0 | RX
1 | TX
2
3 | PWM
4
5 | PWM
6 | PWM
7
ANTENNA
RFD21733
A5
A4
A3
A2
A1
A0
8
9 | PWM
10 | PWM
11 | PWM
12
13
GND
AREF
ARD_IO_10
ARD_IO_11
Vin
GND
GND
5V
3.3V
RESET
+5V
+3.3V_ARD
+3.3V
R1
0.0
0603-RF
SW1
PTS645SM43SMTR LFS
6MM TACT SWITCH
RESET
Arduino shield
footprint
GND
P2
SSQ-108-03-G-S
2.54MM 1x10 PIN HEADER FEMALE
P4
SSQ-106-03-G-S
2.54MM 1x10 PIN HEADER FEMALE
SW2
JS202011SCQN
DPDT SMT SLIDE SWITCHER
ARD_TX
ARD_IO_11
ARD_RX
ARD_IO_10
Hardware
Software
Serial port selector
U1
SN74LVC1G125DBVR
SOT23, 5-Leads
R2
47K 5%
0603-RF
C1
0.1uF -20% +80% 50V Y5V
0603-RF
RFD_DIN
D1
LTST-C191KRKT
0603-RF
DIN
RED
R4
470 5%
0603-RF
U2
SN74AHCT1G125DBVR
SOT23, 5-Leads
D2
LTST-C191KGKT
0603-RF
R5
470 5%
0603-RF
DOUT
GREEN
RFD_DOUT
C2
0.1uF -20% +80% 50V Y5V
0603-RF
Serial port connection
Shield serial port: | SW4 Pos. 1
Disconnected from Arduino | OFF
Connected to Arduino | ON
RFD Mode select
MODE | SW4 Pos. 2
Mode 2 | OFF
Mode 3 | ON
R3
47K 5%
0603-RF
U4
NCP698SQ33T1G
SC-82AB, 4-Leads
NOT PLACED
VIN
VOUT
ENABLE
GND
C4
1.0uF -20% +80% 16V Y5V
0603-RF
NOT PLACED
C5
0.1uF -20% +80% 50V Y5V
0603-RF
NOT PLACED
*Not placed, radio is low power so using +3.3V from Arduino.
R6
470 5%
0603-RF
LEARN
YELLOW
D3
LTST-C191KSKT
0603-RF
Q1
MMUN2214LT1G
SOT23_N
SW3
PTS645SM43SMTR LFS
6MM TACT SWITCH
LEARN
R7
470 5%
0603-RF
R8
47K 5%
0603-RF
R9
47K 5%
0603-RF
SW4
219-2MST
DIPSWITCH2
E1
Wire antenna
(only for RFD21735)
R10
47K 5%
0603-RF
U3
RFD21733 Module
RFDCHIP 15MM
GND
GND
MODE2
LEARN / TX LED / STATUS
IN1 / OUT1 / LOGIC IO
IN2 / OUT2 / RXD
IN3 / OUT3 / TXD
GND
ANTENNA
GND
GND
MODE1
MODE0
FACTORY
/RESET
+V
GND
EXT ANT
GND
GND
RFD_DOUT
RFD_DIN
C3
10uF 20% 6.3V X5R
0603-RF
RFD21815
Wireless Inventors Shield for Arduino
09.06.2012 5:55:08 AM
(C) Copyright 2012, OpenSourceRF.com
Title
R353 B001
Size
A4
Number
Revision
A
Date: 09.06.2012
Sheet 1 of 1
File: D:\Work\..\R353_Arduino_Shield.SchDoc
Drawn By:

MPU-6050 Breakout Board Circuit Diagram.

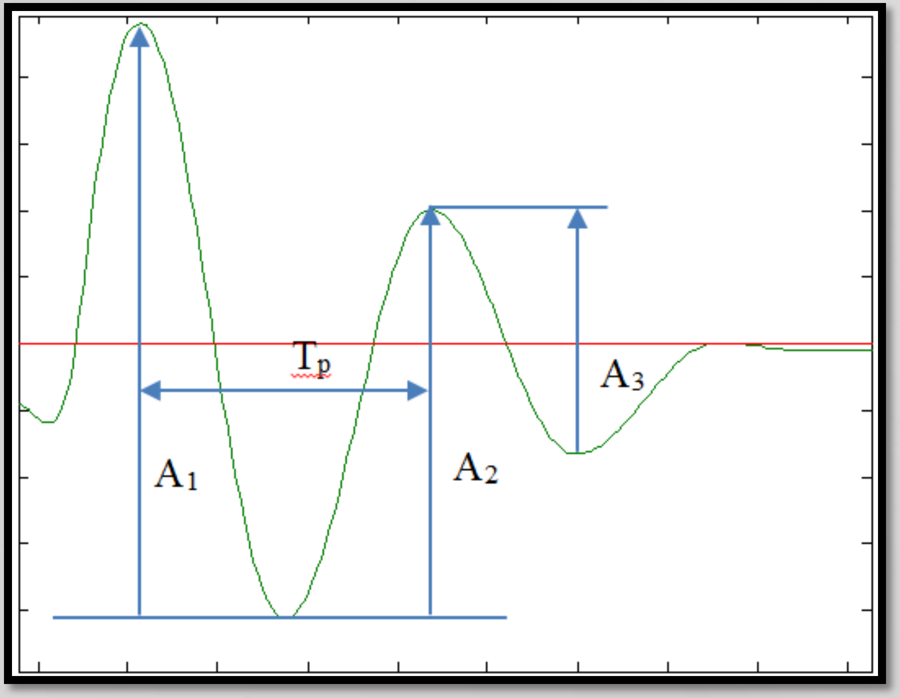


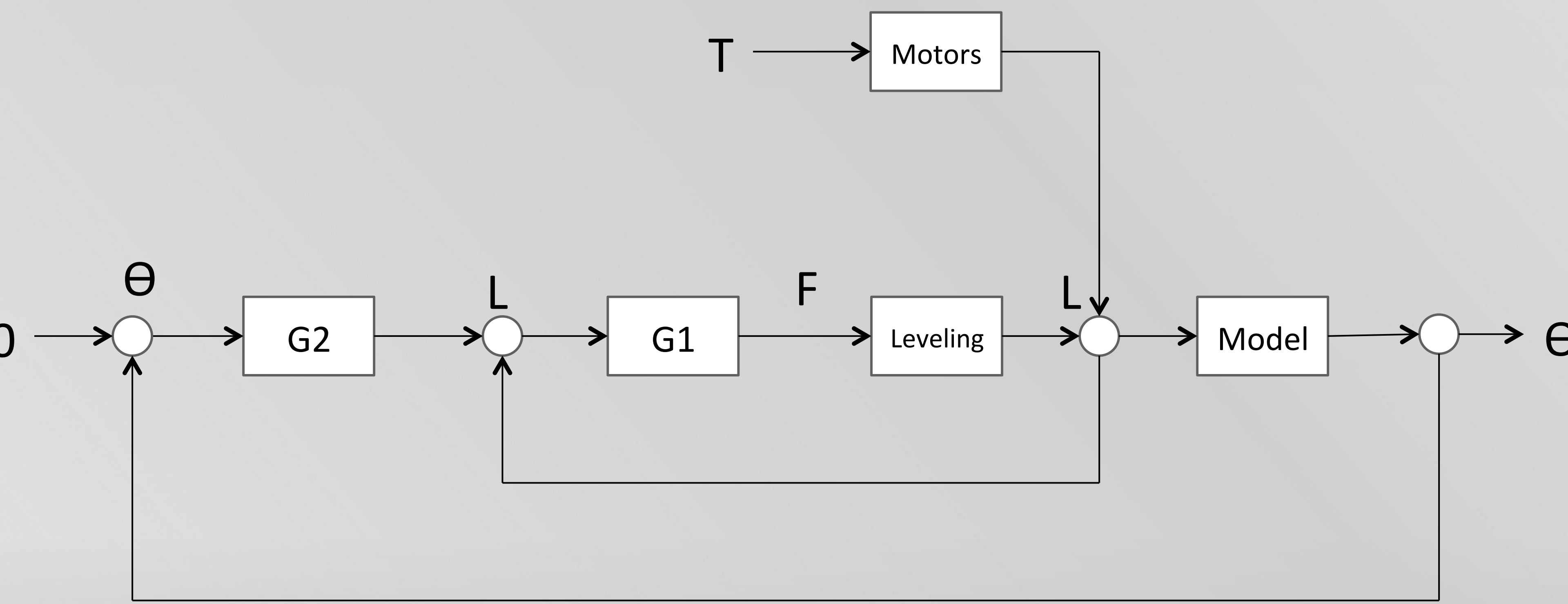
Motor Driver Circuit Schematic

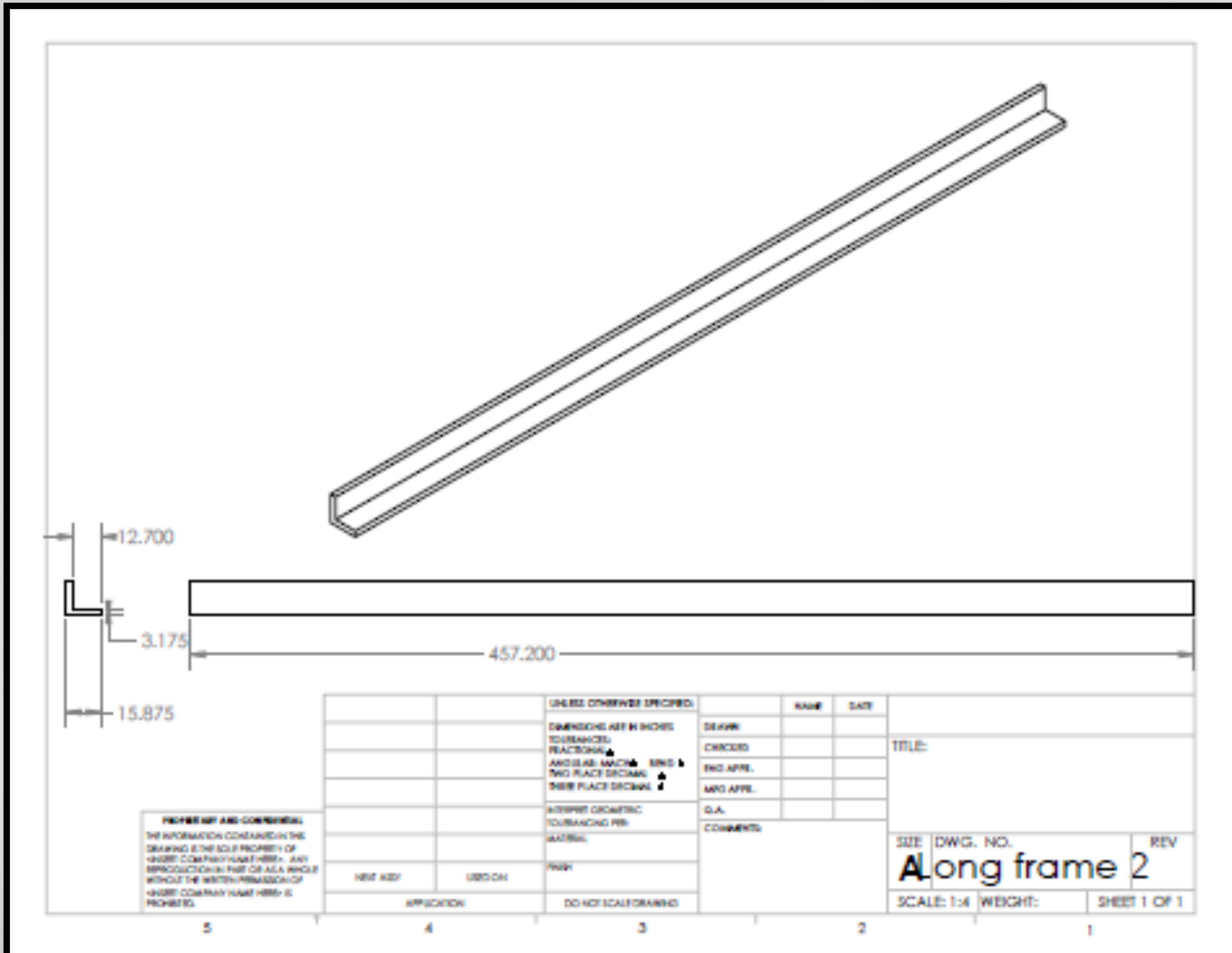


	Time	Amplitude (°)
A1	6.080	98.500
A2	6.480	-56.200
A3	7.210	27.300
A4	8.100	-12.200

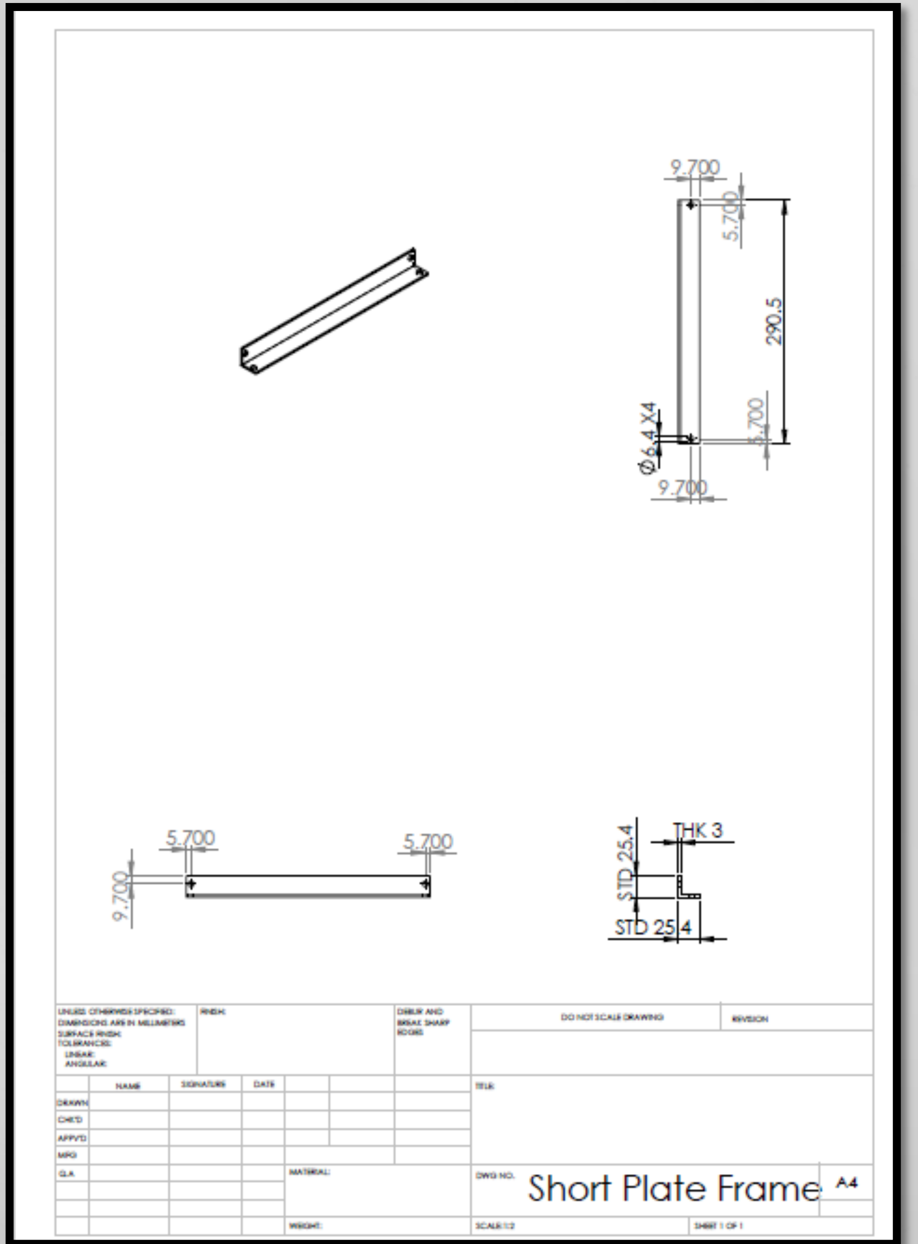
$$\zeta = \frac{1}{\sqrt{1 + \frac{\pi}{\ln\left(\frac{A_1 - A_2}{A_2 - A_3}\right)}}}$$

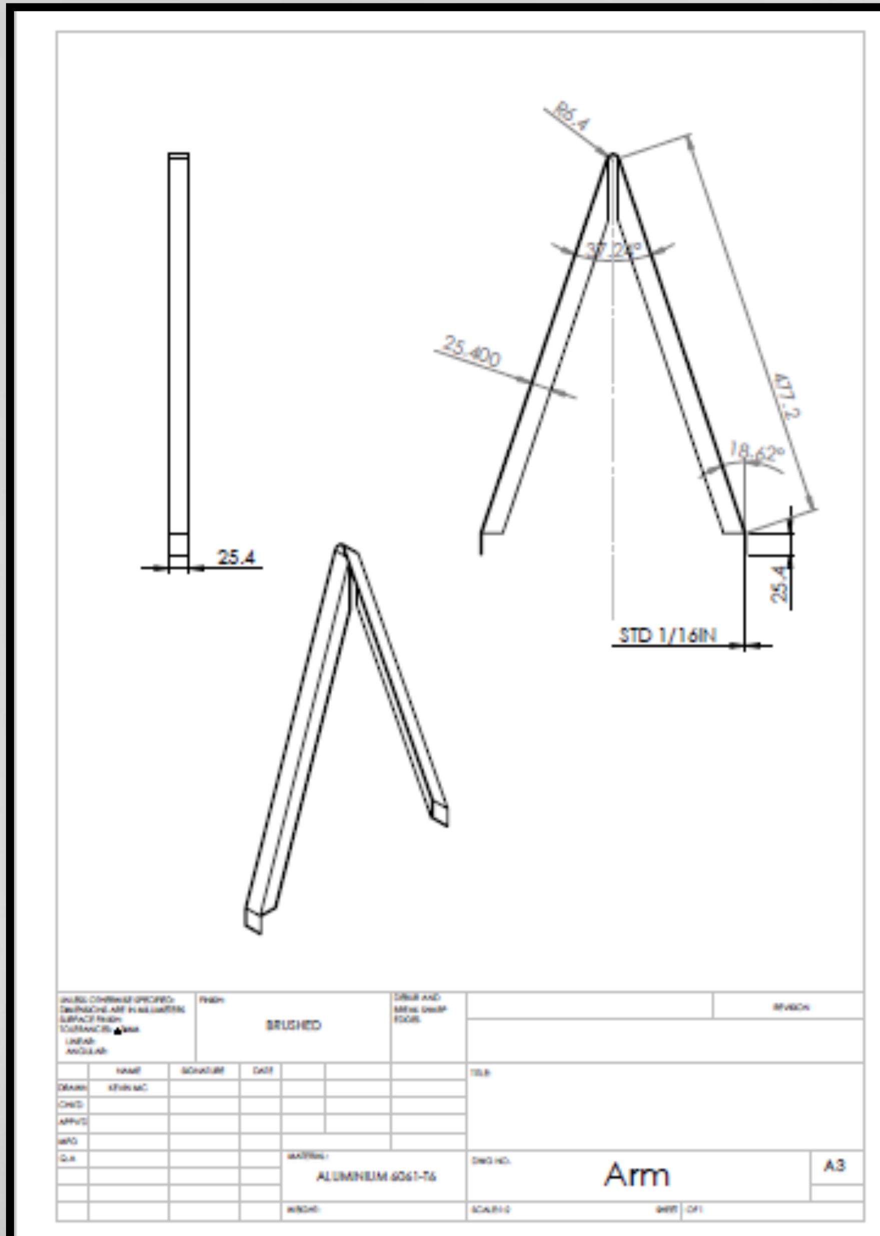


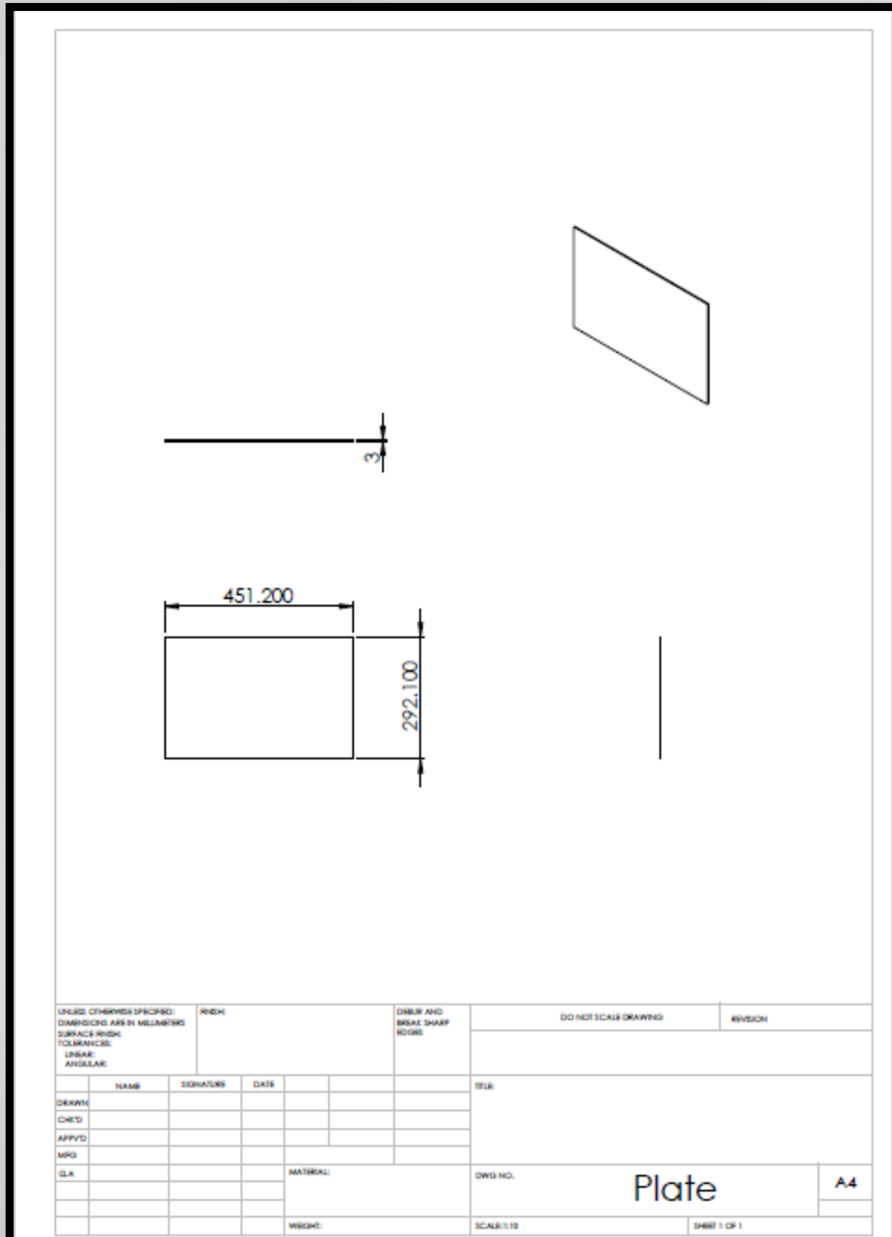

T
Motors
0
θ
G2
L
G1
F
Leveling
L
Model
θ



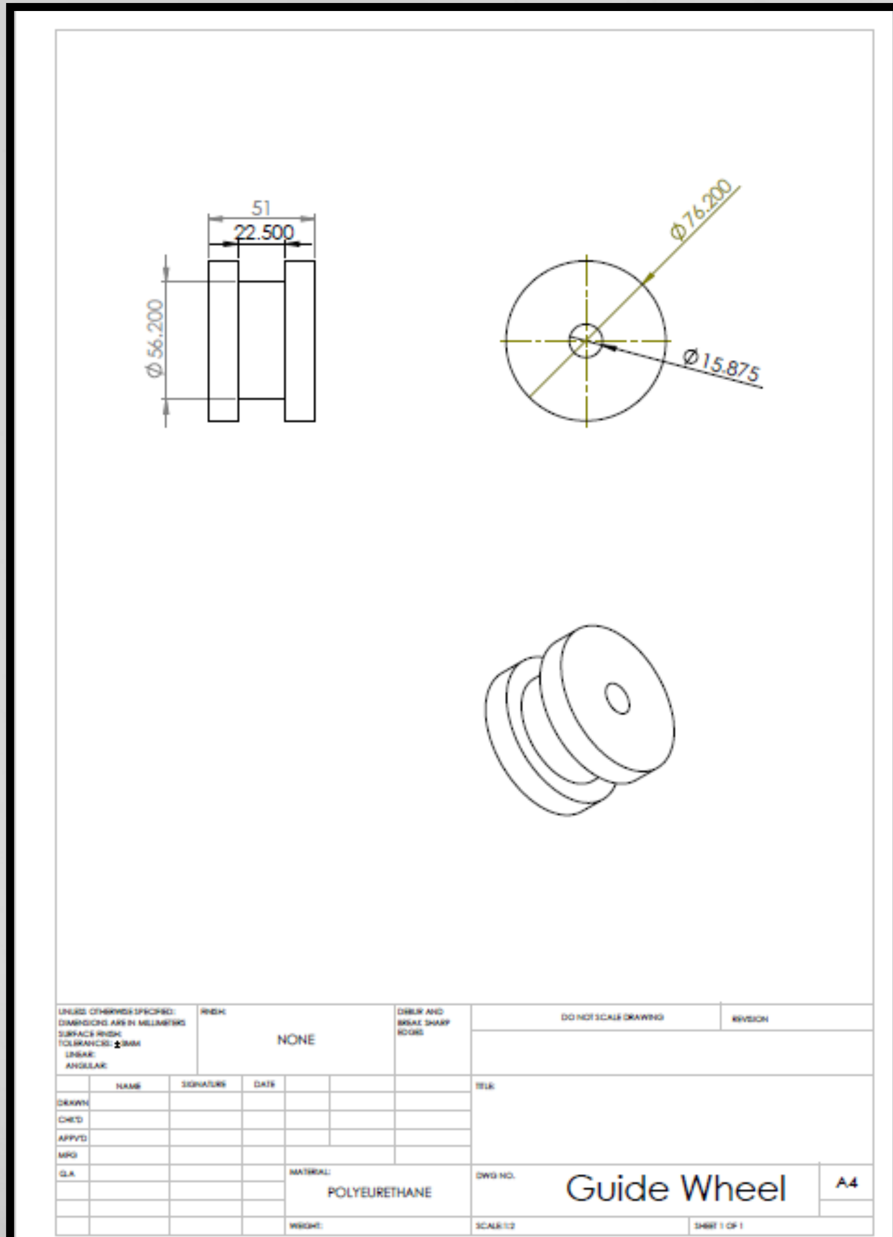
12.700
3.175
457.200
15.875
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL
ANGULAR: MACH BEND
TWO PLACE DECIMAL
THREE PLACE DECIMAL
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
FINISH
DO NOT SCALE DRAWING
NAME
DATE
DRAWN
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS:
TITLE:
NEXT ASSY
USED ON
APPLICATION
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.
SIZE
DWG. NO.
REV
A
Long frame 2
SCALE: 1:4
WEIGHT:
SHEET 1 OF 1
5
4
3
2
1


9.700
5.700
290.5
Ø6.4 X4
9.700
5.700
5.700
9.700
THK 3
STD 25.4
STD 25.4
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBUR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
CHK'D
APPV'D
MFG
Q.A
MATERIAL:
WEIGHT:
TITLE:
DWG NO.
Short Plate Frame
A4
SCALE:1:2
SHEET 1 OF 1

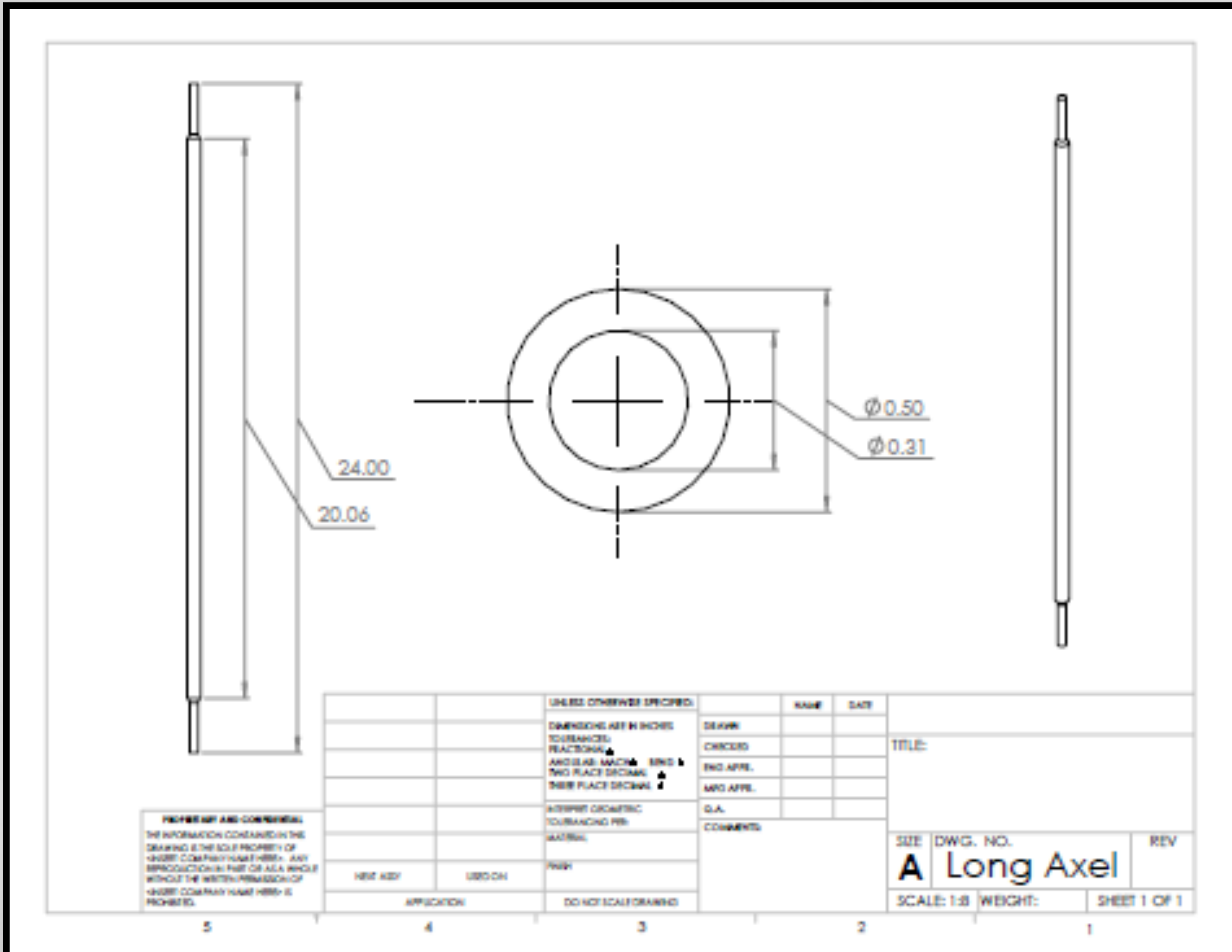

R6.4
37.24°
25.400
477.2
18.62°
25.4
STD 1/16IN
25.4
BRUSHED
ALUMINIUM 6061-T6
Arm
A3

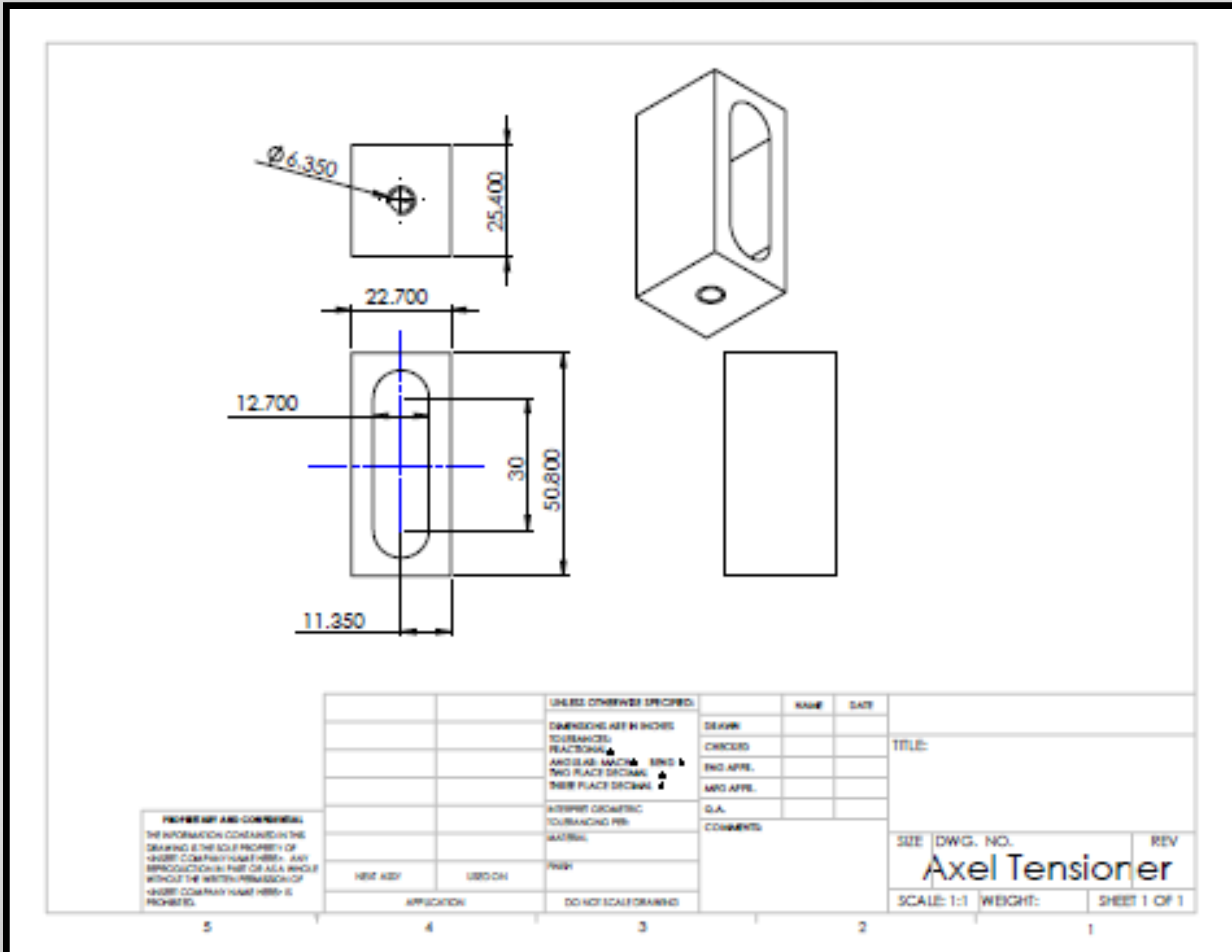


3
451.200
292.100
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBUR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
CHK'D
APPV'D
MFG
Q.A
MATERIAL:
TITLE:
DWG NO.
Plate
A4
WEIGHT:
SCALE:1:10
SHEET 1 OF 1

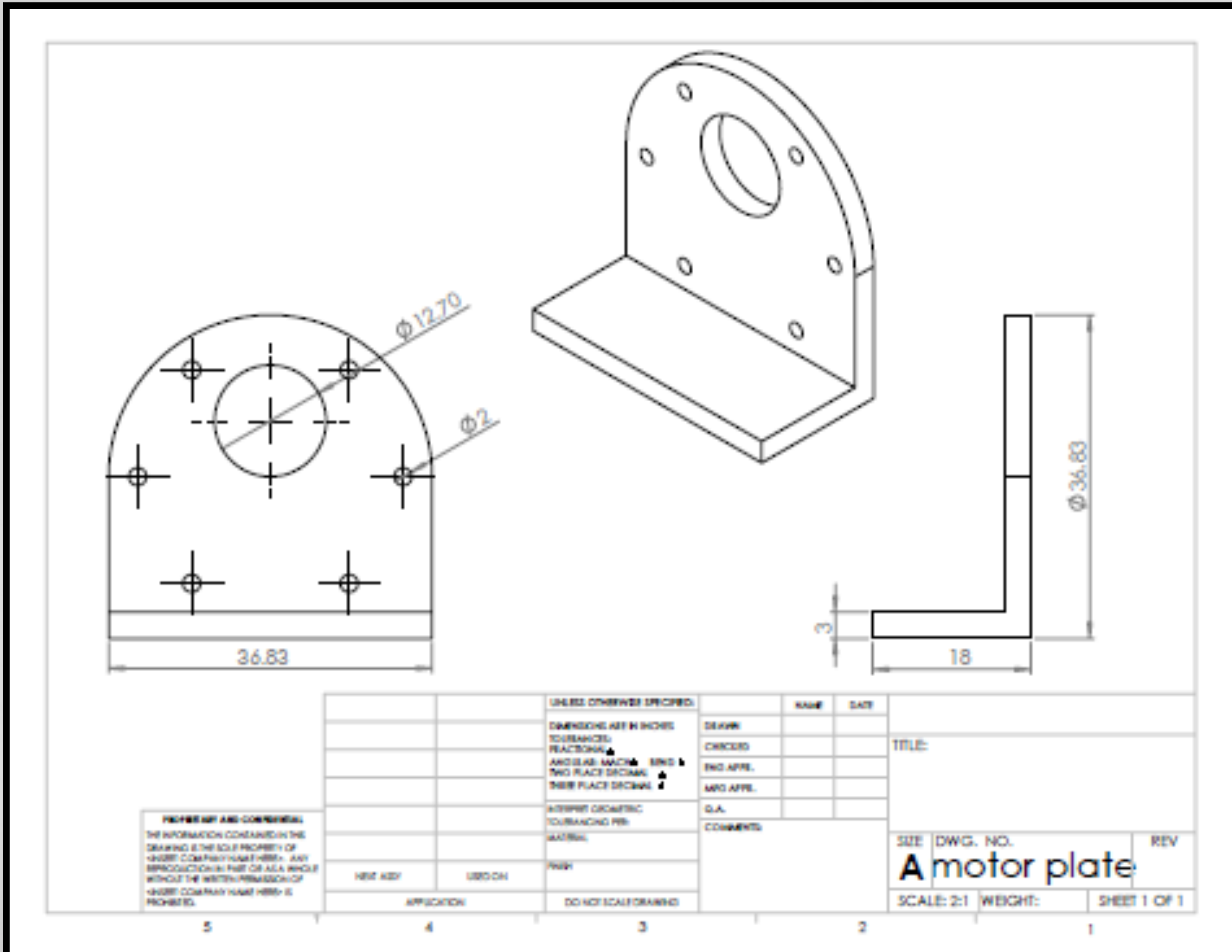


51
22.500
Ø 56.200
Ø 76.200
Ø 15.875
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
NONE
DEBUR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
CHK'D
APPV'D
MFG
Q.A
TITLE:
MATERIAL:
POLYEURETHANE
DWG NO.
Guide Wheel
A4
WEIGHT:
SCALE:1:2
SHEET 1 OF 1

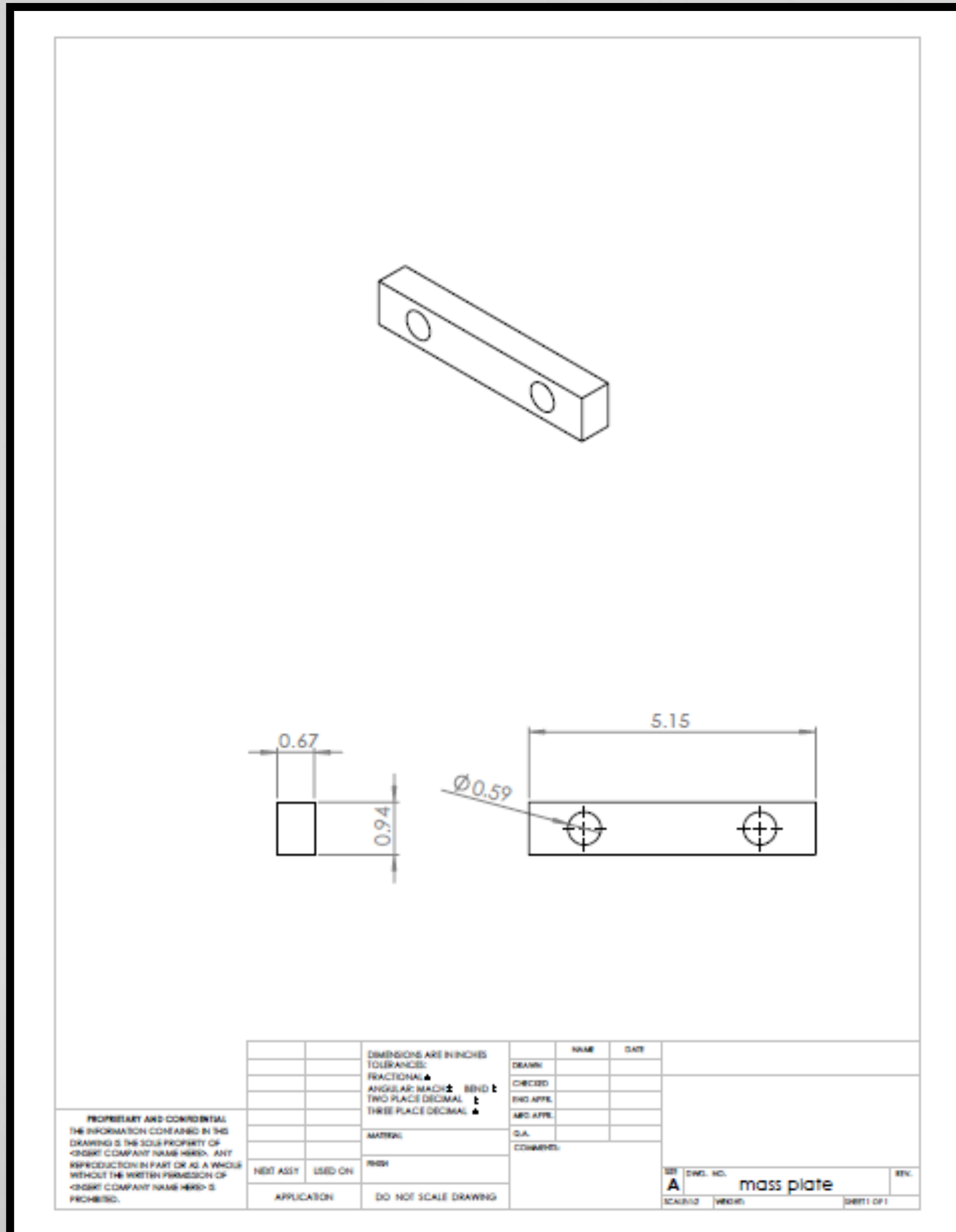

24.00
20.06
Ø0.50
Ø0.31
TITLE:
SIZE
DWG. NO.
REV
A
Long Axel
SCALE: 1:8
WEIGHT:
SHEET 1 OF 1
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
NAME
DATE
DRAWN
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS:
NEXT ASSY
USED ON
APPLICATION
DO NOT SCALE DRAWING
PROPRIETARY AND CONFIDENTIAL
5
4
3
2
1

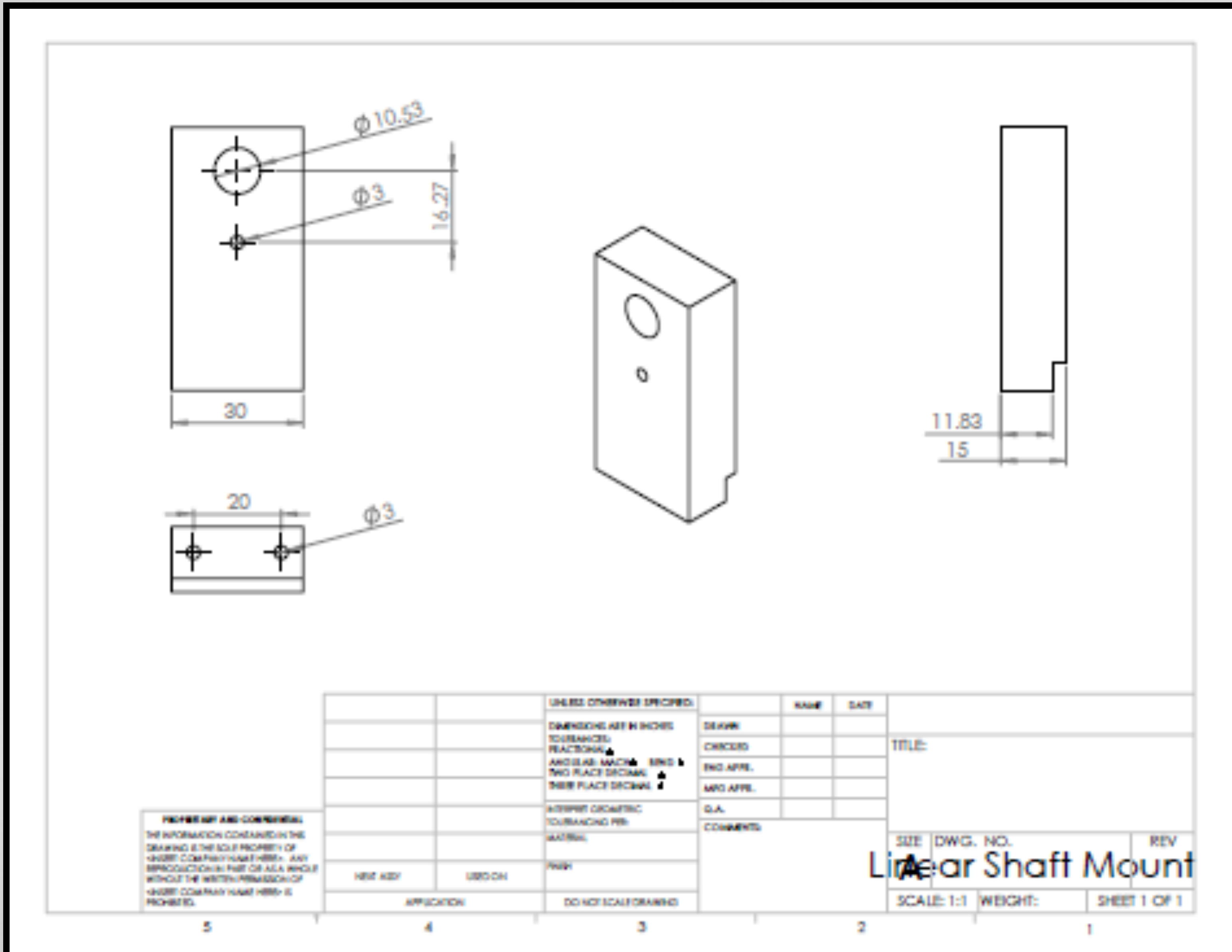


Ø6.350
25.400
22.700
12.700
30
50.800
11.350
UNLESS OTHERWISE SPECIFIED:
NAME
DATE
DRAWN
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS:
TITLE:
PROPRIETARY AND CONFIDENTIAL
NEXT ASSY
USED ON
APPLICATION
DO NOT SCALE DRAWING
SIZE
DWG. NO.
REV
Axel Tensioner
SCALE: 1:1
WEIGHT:
SHEET 1 OF 1
5
4
3
2
1



Ø12.70
Ø2
36.83
Ø36.83
3
18
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL
ANGULAR: MACH BEND
TWO PLACE DECIMAL
THREE PLACE DECIMAL
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
FINISH
NEXT ASSY
USED ON
APPLICATION
DO NOT SCALE DRAWING
NAME
DATE
DRAWN
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS:
TITLE:
SIZE
DWG. NO.
REV
A
motor plate
SCALE: 2:1
WEIGHT:
SHEET 1 OF 1
5
4
3
2
1


0.67
0.94
Ø0.59
5.15
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL
ANGULAR: MACH BEND
TWO PLACE DECIMAL
THREE PLACE DECIMAL
MATERIAL
FINISH
NAME
DATE
DRAWN
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS:
PROPRIETARY AND CONFIDENTIAL
NEXT ASSY
USED ON
APPLICATION
DO NOT SCALE DRAWING
A
DWG. NO.
mass plate
REV.



ϕ10.53
ϕ3
16.27
30
20
ϕ3
11.83
15
UNLESS OTHERWISE SPECIFIED:
NAME
DATE
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL
ANGULAR: MACH BEND
TWO PLACE DECIMAL
THREE PLACE DECIMAL
DRAWN
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS:
TITLE:
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
FINISH
PROPRIETARY AND CONFIDENTIAL
NEXT ASSY
USED ON
APPLICATION
DO NOT SCALE DRAWING
SIZE
DWG. NO.
REV
A
Linear Shaft Mount
SCALE: 1:1
WEIGHT:
SHEET 1 OF 1
5
4
3
2
1