# INVERTED PENDULUM ON A CART SENIOR DESIGN PROJECT REPORT

An Inverted Pendulum on a Cart Useful for

Educating Future Students.

Submitted to

Dr. Robert Weissbach

Professor Elizabeth Freije

Mr. Craig Stephens

Engineering Technology Department

by

Sergio Vergara

Hawra Aljishi

Dylan Logan

May 3, 2023

# EXECUTIVE SUMMARY

The purpose of this project is to design and build an inverted pendulum system from an existing nonworking inverted pendulum system for the Engineering and Technology Department. Once completed, this system will be used for educational purposes for future students to observe code, understand the design, and study the physics of the system. We have tested our system numerous times with a variety of different device configurations so as to perfect the system. We learned a great deal from the redesigning and rebuilding of this entire system, so know it will meet the sponsor's needs of educating future students. We have clearly labeled and documented everything, which will help when students are working on the system in the future. Future students can improve upon the system by adding limit switches that will allow them to know when they have reached the end limits of the pendulum with the cart. They could also secure the cart better, so it tracks straight and has no forward and back movement on the track. This document contains but is not limited to the information for our scope, design decisions, schematics, design blocks, bill of materials, and code. This was designed to show the finalized overview of our systems and what we presented at the end of the semester.

# TABLE OF CONTENTS

**EXECUTIVE SUMMARY** ........ **2**
**REVISION HISTORY** ........ **4**
**SCOPE & MARKETING REQUIREMENTS** ........ **5**
Features ........ 5
In Scope ........ 5
Out of Scope ........ 5
**SPECIFICATION REQUIREMENTS** ........ **5**
System ........ 5
Pendulum ........ 5
**SYSTEM DESIGN** ........ **6**
Original System Overview ........ 6
Referenced Documents ........ 6
System-Wide Decisions ........ 6
System Architectural Design ........ 9
System Components ........ 12
Concept of Execution ........ 13
**TEST METHODOLOGY & RESULTS** ........ **14**
Supply List ........ 14
Personnel Required ........ 14
Equipment Description ........ 14
Tests ........ 15
**CONCLUSION & NOTES** ........ **20**
**APPENDICES** ........ **21**
Final Track ........ 21
Final Enclosure ........ 22
Fritzing of System: Diagram ........ 23
Fritzing of System: Schematic ........ 24
Software Flow Chart ........ 25
Final Main Code ........ 29
Datasheets ........ 36
Gant Chart ........ 36
Bill of Materials ........ 37

**REVISION HISTORY**

| Version | Date | Revised by | Description |
|---|---|---|---|
| 1.0 | 03/31/2023 | Sergio Vergara, Hawra Aljishi, Dylan Logan | Initial Report Draft |
| 1.1 | 03/31/2023 | Sergio Vergara | Added updated documentation from LLD and code. |
| 1.2 | 03/31/2023 | Hawra Aljishi | Added updated test mythology, test result, requirements, datasheets, and Gantt chart. |
| 1.3 | 3/31/2023 | Dylan Logan | Added Executive Summary, High level design documentation, Summary, and notes. |
| 1.3 | 5/2/2023 | Dylan Logan, Sergio Vergara, Hawra Aljishi | Updated Diagrams, code, and other sections of paper |

## SCOPE & MARKETING REQUIREMENTS

***Features***

Our system has many features to allow for a user experience that is consistent in execution and simple in terms of user input. Physical controls are installed on the system's enclosure to allow manual movement, automatic operation, and system power control. The system relies on a belt-driven tracking system to reduce motion noise and to depend less on gears that break at a faster rate. A new power supply is included to power the system, avoiding older components that cannot provide the same consistent power to run the system. The entire system runs on code that was created to automate the cart's movement and provide the desired result in a few minutes.

***In Scope***

- Change the existing microcontroller.
- Change the existing Power supply.
- Add casing for our existing controller.
- Change the existing motors from DC to stepper.
- Change the existing tracking system for the cart.
- Update the existing code for the cart.
- Update the existing cabling system.
- Change the existing circuit board.
- Update the existing cart.

***Out of Scope***

- Having a wireless controller connection.
- Using LabView to control the cart.
- Using Simulink to control the cart.
- Using PLC.

## SPECIFICATION REQUIREMENTS

***System***

- All electronics for the system must be run using 120VAC or less.
- The emergency stop button needs to shut down the whole system.
- The start switch starts and turns off the automatic operation of the cart movements to swing the pendulum.
- The left and right buttons must move the cart left and right for manual adjustment.

***Pendulum***

- Start from non-inverted position (down position).
- Must balance in the upright position.

## SYSTEM DESIGN

### *Original System Overview*

- There needs to be an updated system controller implemented in the project.
- New stepper motors will be replacing old DC motors.
- The cabling system will need to be changed.
- The power supply will be replaced.

### *Referenced Documents*

| Title | Source | Comment |
|---|---|---|
| NFPA National Electric Code | https://www.nfpa.org/Codes-and-Standards/All-Codes-and-Standards/List-of-Codes-and-Standards | The latest NEC international electric code handles topics such as wiring methods and grounding protection. |
| IEEE Draft Guide for AC Motor Protection | "IEEE Draft Guide for AC Motor Protection," in IEEE PC37.96/D11, August 2012 , vol., no., pp.1-141, 23 Aug. 2012. | Generally accepted methods of protection for ac motors are provided for certain sizes and applications. |

### *System-Wide Decisions*

Criteria considered in this Section:

- Troubleshooting Time – Time taken for testing, designing, and replacing components.
- Reliable – End of life for component (how has it affected the system?)
- Existing Software–How much software related documentation is correct and available?
- Cost – How much would it cost to keep the existing system and components and repair it or cost to replace the entire system or individual components?
- Portable – Is the system able to be removed and installed elsewhere?
- Visual Appeal – Is the system and its components well packaged and updated or does it look dated, with loose wiring and exposed components?
- Control – Motor's response time to given commands and manual controls.
- Precision – How precise can we get the movements from the motor?
- Speed – How fast can the motor physically be?
- Interrupt Pins – The system requires at least 4 interrupt pins.
- Input Voltage – The system requires at least a 3VDC input voltage (however >=5 is preferred).
- Footprint – How big will it be? (Smaller is preferred to better fit 16”x16”x8” enclosure.)
- Documentation – Is the documentation correct and updated?
- Existing Hardware - Is the hardware existing, and in a useable state? (Without needing replacements?)
- Redesign Time – Time taken to complete tasks like balancing pendulum, correcting weight, re-designing current track, replacing dc motors with stepper of motors, and re-designing code for new motors, encoder signal logic and pendulum track.

## Base/Track

| Worst Option | Mid Option | Best Option |
|---|---|---|
| 1 | 2 | 3 |

| Higher # is more desirable for weights | Troubleshooting Time | Reliable | Existing Software | Cost | Portable | Visual Appeal | Total |
|---|---|---|---|---|---|---|---|
| Weight | 6 | 5 | 4 | 3 | 2 | 1 | # |
| Old Base | 1 | 2 | 3 | 3 | 2 | 1 | 42 |
| Base Given to us | 2 | 3 | 1 | 3 | 3 | 3 | 49 |
| Custom Base | 2 | 3 | 1 | 1 | 3 | 3 | 43 |

## Motor

| Worst Option | Mid Option | Best Option |
|---|---|---|
| 1 | 2 | 3 |

| Higher # is more desirable for weights | Control | Precision | Speed | Existing Software | Cost | Reliable | Total |
|---|---|---|---|---|---|---|---|
| Weight | 6 | 5 | 4 | 3 | 2 | 1 | # |
| DC Motor | 1 | 1 | 2 | 3 | 3 | 2 | 36 |
| Stepper Motor | 3 | 3 | 3 | 1 | 3 | 3 | 57 |
| Servo Motor | 1 | 2 | 3 | 1 | 1 | 3 | 36 |

## Microcontroller

| Worst Option | Mid Option | Best Option |
|---|---|---|
| 1 | 2 | 3 |

| Higher # is more desirable for weights | Interrupt Pins | Input Voltage | Existing Software | Cost | Footprint | Total |
|---|---|---|---|---|---|---|
| Weight | 5 | 4 | 3 | 2 | 1 | # |
| Arduino Mega | 2 | 3 | 3 | 3 | 1 | 38 |
| Arduino Uno | 1 | 3 | 3 | 2 | 2 | 32 |
| ESP32 | 3 | 1 | 1 | 3 | 3 | 31 |

## Power Supply

| | Worst Option | Mid Option | Best Option |
|---|---|---|---|
| | 1 | 2 | 3 |

| Higher # is more desirable for weights | Documentation | Reliable | Footprint | Portable | Cost | Total |
|---|---|---|---|---|---|---|
| Weight | 5 | 4 | 3 | 2 | 1 | # |
| Old P.S. | 2 | 1 | 3 | 1 | 3 | 28 |
| Purchased P.S. | 3 | 3 | 2 | 3 | 1 | 40 |
| Lab P.S. | 3 | 3 | 1 | 1 | 3 | 35 |

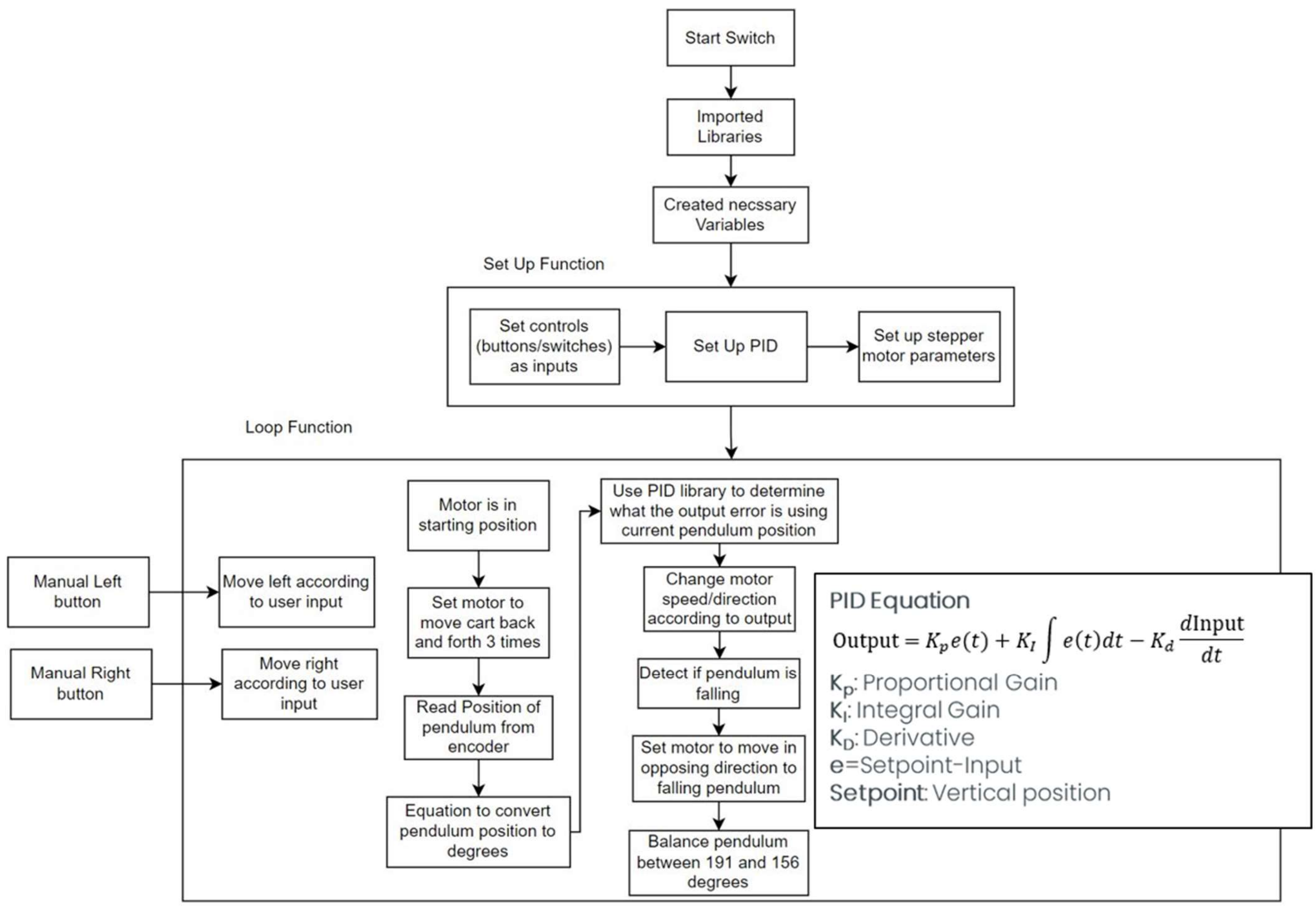
## Pendulum System

| | Worst Option | Mid Option | Best Option |
|---|---|---|---|
| | 1 | 2 | 3 |

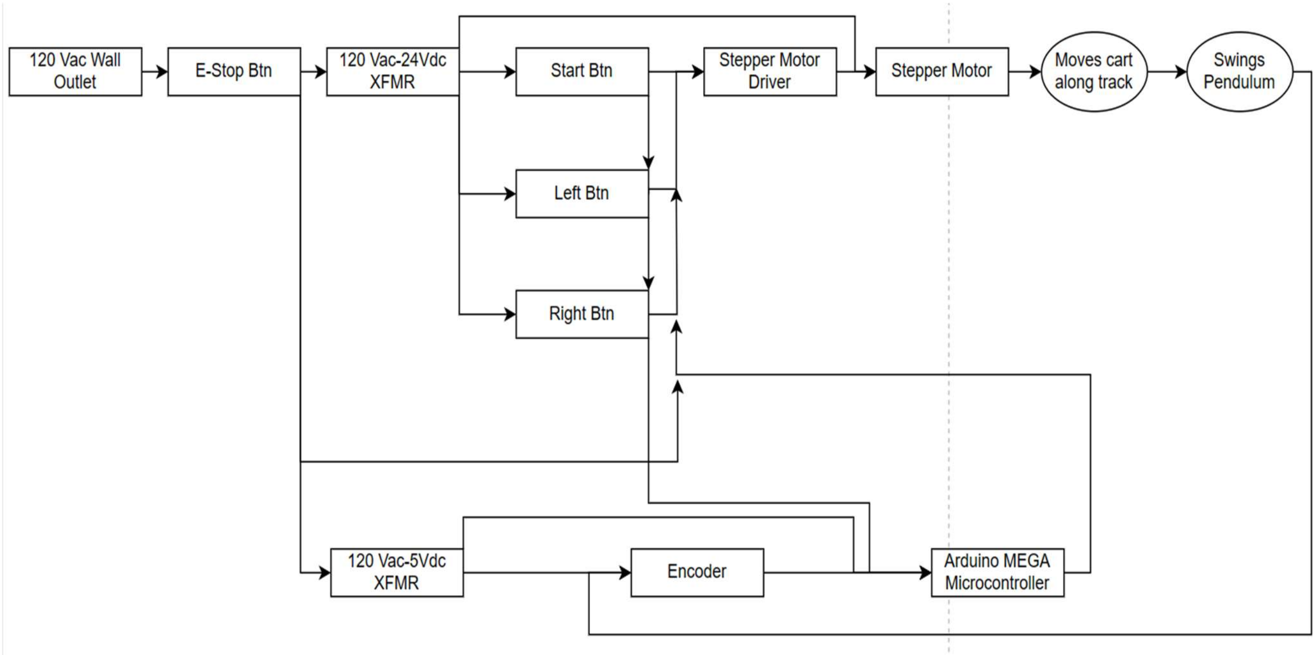
| Higher # is more desirable for weights | Existing Hardware | Redesign Time | Reliable | Cost | Visual Appeal | Total |
|---|---|---|---|---|---|---|
| Weight | 5 | 4 | 3 | 2 | 1 | # |
| Old Pendulum Components | 3 | 2 | 2 | 3 | 2 | 37 |
| New Pendulum Components | 1 | 1 | 3 | 2 | 3 | 25 |

***System Architectural Design***

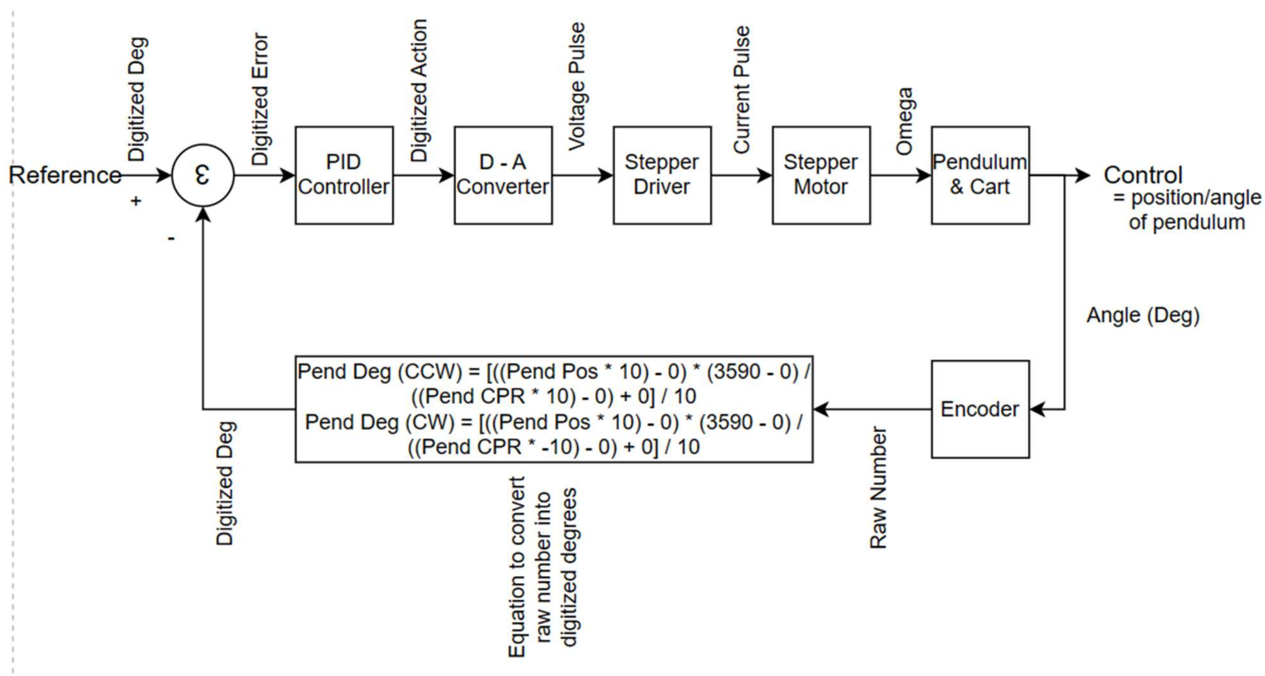
**<u>Software Block Diagram</u>**



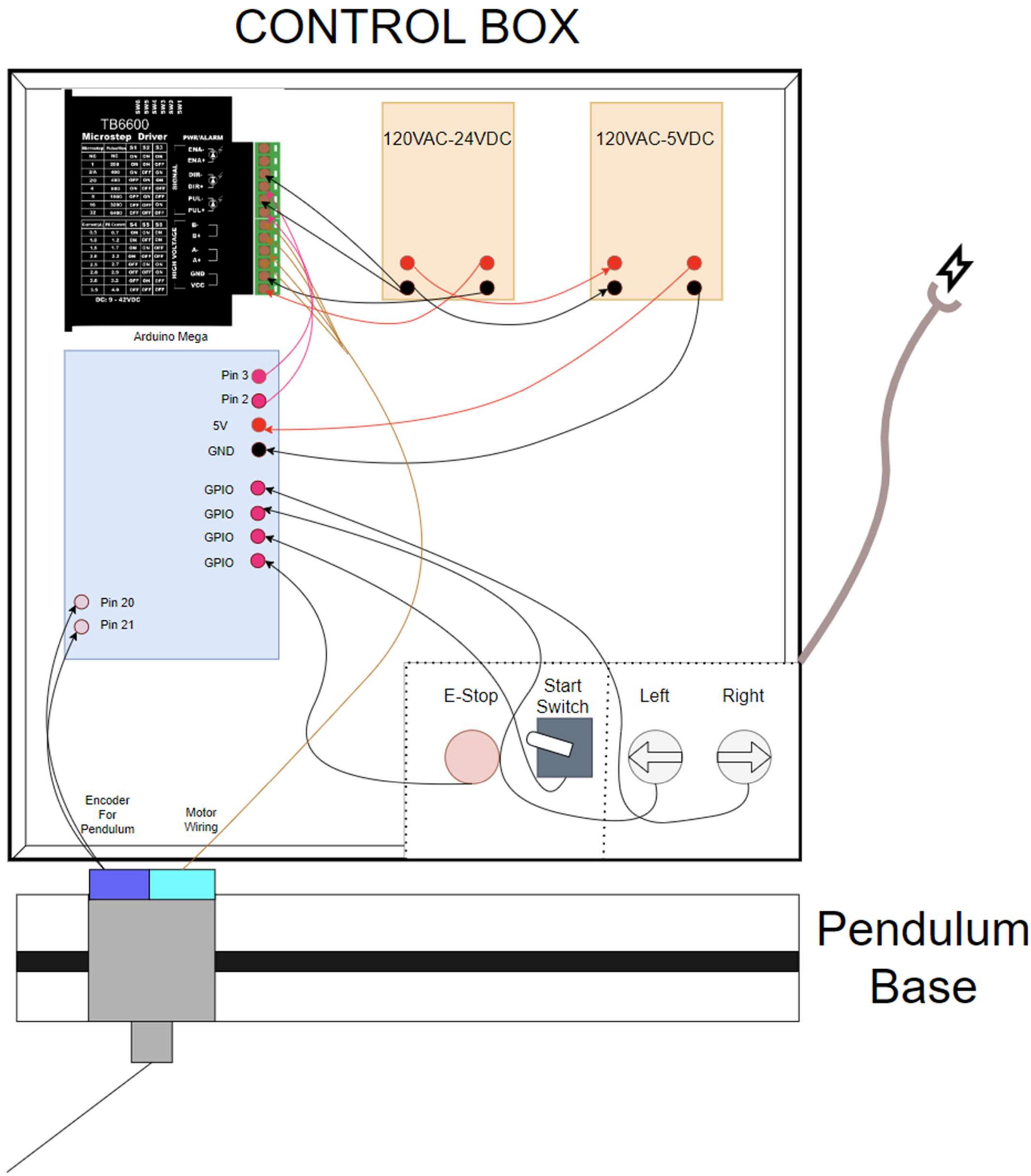
## Hardware Block Diagram




## Control Diagram



**User Interface Design**



***System Components***

- On Control Box
  - **E-Stop Button** – We chose to use an E-Stop Button at the beginning of our diagram so that if there are any problems down the line with power or the pendulum is about to hit something for whatever reason, all an operator must do is push the E-Stop button in and everything will be de-energized.
  - **On/Off Switch** – We chose to use an On/Off switch so we can power our pendulum to start the program automatically when the switch is flipped on. With this, we can also stop the automatic operation of our program by setting the switch to the off position.
  - **Left/Right Buttons** - We chose to use manual buttons as opposed to the original circuit board with buttons on it as we saw that the board would be more delicate and run the risk of breaking the board while in use. We decided that the manual buttons will look much better and be more rigid on our control box. We chose to use a Left Button so we can manually swing the pendulum by forcing the stepper motor to move the cart left on the track without the program running automatically. We chose to use a Right Button so we can manually swing the pendulum by forcing the stepper motor to move the cart right on the track without the program running automatically.
- Inside Control Box
  - **120 Vac – 24 Vdc Transformer / 120 Vac– 5 Vdc Transformer** – One reason we chose to make use of a purchased transformer is because power supplies of this size have gone way down in price. Another reason is because the power supply that was on the given board on the nonfunctional pendulum was designed and built in 1999 and struggled to sufficiently power the system. This will give us ease of being able to power the new stepper motor adequately as the old power supply only supplied ~17 V, whereas the new stepper motor requires 24 Vdc. The second transformer will also be able to efficiently power the Arduino MEGA microcontroller.
  - **Stepper Motor Driver** – We chose to use a stepper motor driver as it is a way for the microcontroller to communicate with the stepper motor and tell it how much to move and when. The driver will drive the stepper motor, which will swing the pendulum, the pendulum's position will be read by the pendulum's encoder, which will report its readings to the microcontroller, which will in turn communicate back to the driver to tell the stepper motor how to move to swing the pendulum.

- Outside Control Box
    - **Stepper Motor** – We chose to use a stepper motor due to being able to read its position without the use of an encoder. The stepper motor should be able to change directions quick enough to enable the pendulum to swing up into a position to where the stepper motor can control the cart and keep the pendulum upright.
    - **Pendulum** – We chose to use the original pendulum due to wanting to prove that we can make the original pendulum work by only changing some details, but not reengineering a new pendulum. We determined the existing pendulum, and its original encoder could continue to provide the operational performance and signaling needed to control the system.
    - **Encoder** - We chose to use the original encoder for the pendulum due to wanting to prove that we can make the given pendulum work by only changing some details, but not reengineering a whole new pendulum with another encoder. We wanted to make use of the encoder we already have mounted on the pendulum itself.
    - **Power Cable –** We have incorporated a power cable that will plug into a wall outlet to provide 120Vac for the system.

***Concept of Execution***

1. Plug the power cord into a 120Vac wall outlet, and electricity will flow to the transformer.
2. Use the ON/OFF switch to power the pendulum system (when the switch is set to "on" the program starts automatically, when the switch is set to the off position the automatic operation of the program stops).
3. Once the switch is set to on, the power supply will power up both the cart motor and the logic board.
4. The encoder of the pendulum will start to take measurements of its position.
5. The stepper motor will move left and right depending on the pendulum's position (shown in degrees).
6. The motion of the stepper motor's cart will begin to invert the pendulum. The encoder of the pendulum will keep measuring and it will send the position data to the microcontroller.
7. Once in between in its upright position, the pendulum will stay balanced. It will still need to send/receive data commands to "hold" the pendulum in an upright position.

# TEST METHODOLOGY & RESULTS

The purpose of this part of the document is to provide instructions and information on how to test if the built inverted pendulum on a cart is considered complete and functional. Several pieces of the project need to be evaluated to consider it a successful creation that a consumer may use. For example, the correct voltages and currents need to be inserted in each of the different components (microcontroller, stepper motor, motor driver, etc.) in order to have a fully functioning product. This document also goes into details on how to determine if the cart is working as intended in terms of its movements as well as if the microcontroller is getting the signals it needs (such as from an encoder).

### *Supply List*

- Digital multimeter
    - Needs to be able to read currents and voltages
- Multimeter probes
- Arduino Mega Microcontroller
- Transformer (120VAC – 24VDC)
- Transformer (24VDC – 5VDC)
- Product Stand
- Pendulum Track (with cart built in)
- Monitor

### *Personnel Required*

- Operations Manager(s)
- Computer Engineer(s)
- Electrical Engineer(s)

### *Equipment Description*

A digital multimeter will be used in many, but not all, of the upcoming tests and procedures. The multimeter will be provided by the Electrical Engineer and will allow the engineers to visualize and document the current and voltage readings gathered from our standard probing techniques. Several other processes will be tested and observed through code, in order to confirm that a signal is being grabbed, a string output will be utilized. For example, an if statement will be used in order to test for a signal connection, if one is not received then a string will be shown on a monitor to confirm.

***Tests***

**Test 1 - System Power**

| Test Description | Verify the components are receiving correct voltages | | |
|---|---|---|---|
| Location | Circuitry | | |
| Step | Actions | Expected Results | Pass (✓) or Fail (X) |
| 1. | Turn on system power. | Verify System power. | Pass |
| 2. | Probe transformer 1 input with multimeter | Verify 120VAC from wall outlet is coming into the transformer. | Pass |
| 3. | Probe transformer 1 output with multimeter | Verify 120VAC is converted to 24VDC. | Pass |
| 4. | Probe stepper motor driver input with multimeter | Verify driver gets 24VDC. | Pass |
| 5. | Probe transformer 2 output with multimeter | Verify 120VAC is stepped down to 5VDC. | Pass |
| 6. | Probe microcontroller input with multimeter | Verify microcontroller and encoder are getting 5VDC. | Pass |
| 7. | Turn off system power. | Verify process fully shuts down entire system. | Pass |

**Test 2 – Start Switch**

| Test Description | Verify the start switch enables the pendulum system | | |
| --- | --- | --- | --- |
| Location | Circuitry & Code | | |
| Step | Actions | Expected Results | Pass (✓) or Fail (X) |
| 1. | Turn on system power. | Verify System power. | Pass |
| 2. | Probe transformer 1 input with multimeter | 120VAC from wall outlet is coming into the transformer. | Pass |
| 3. | Flip Start Switch to the "On" position | Verify that the motor starts moving the cart and attempting to balance the pendulum automatically. | Pass |
| 4. | Flip Start Switch to the "Off" position | Verify System the motor stops moving and the automatic operation of the system halts. | Pass |
| 5. | Turn off system power. | Verify process fully shuts down automatic operation of system. | Pass |

**Test 3 – E-stop**

| Test Description | Verify the e-stop button cuts power | | |
|---|---|---|---|
| Location | Circuitry & Code | | |
| Step | Actions | Expected Results | Pass (✓) or Fail (X) |
| 1. | Turn on system power. | Verify System power. | Pass |
| 2. | Probe transformer 1 input with multimeter | 120VAC from wall outlet is coming into the transformer. | Pass |
| 3. | Flip Start Switch to the "On" position | Verify that automatic pendulum operation starts process. | Pass |
| 4. | Press E-Stop button | Verify System has no power. | Pass |
| 5. | Probe microcontroller input with multimeter | Verify microcontroller and encoder are not getting 5VDC. | Pass |
| 6. | Turn off system power. | Verify process fully shuts down automatic operation of system. | Pass |

**<u>Test 4 – Pendulum</u>**

| Test Description | Verify the cart can balance the pendulum | | |
|---|---|---|---|
| Location | Circuitry & Code | | |
| Step | Actions | Expected Results | Pass (✓) or Fail (X) |
| 1. | Turn on system power. | Verify System power. | Pass |
| 2. | Flip Start Switch to the "On" position | Verify that pendulum system starts process. | Pass |
| 3. | Wait until Pendulum gets to upright position | Verify that the pendulum can hold its upright position. | Fail |
| 4. | Lightly push the pendulum in a single direction (no more than ~5 degrees from upright position) | Verify that the cart moves along the tracks to keep pendulum upright. | Fail |
| 5 | Flip Start Switch to the "Off" position | Verify System the motor stops moving and shuts down automatic operation of system. | Pass |
| 6. | Turn off system power. | Verify process fully shuts down automatic operation of system. | Pass |

**<u>Test 5 – Manual Movement</u>**

| | | | |
|---|---|---|---|
| Test Description | Verify the cart can be moved left and right (manually) | | |
| Location | Circuitry & Code | | |
| Step | Actions | Expected Results | Pass (✓) or Fail (X) |
| 1. | Turn on system power. | Verify System power. | Pass |
| 2. | Press the right button on the control panel. | Verify that the motor moves the cart to the right. | Pass |
| 3. | Press the left button on the control panel | Verify that the motor moves the cart to the left. | Pass |
| 4. | Turn off system power. | Verify process fully shuts down automatic operation of system. | Pass |

## CONCLUSION

In conclusion, we redesigned and rebuilt an inverted pendulum system from an existing nonworking inverted pendulum system for the Engineering and Technology Department. This system will now be used for educational purposes for future students to observe code, understand the design, and study the physics of the system. We have tested our system numerous times with a variety of different device configurations so as to perfect the system. We learned a great deal from the redesigning and rebuilding of this entire system, so we know it meets the sponsor's needs of educating future students. We have clearly labeled and documented everything, which will help when students are working on the system in the future. Some possible future improvements include adding limit switches that will allow the microcontroller to know when the cart has reached the end limits of the pendulum. Students could also secure the cart better, so it tracks straight and has no forward and back movement on the track. In this document, we outlined the information for our scope, design decisions, schematics, design blocks, bill of materials, and code. We constructed this report to show the finalized overview of our systems and what we will present by the end of the semester.

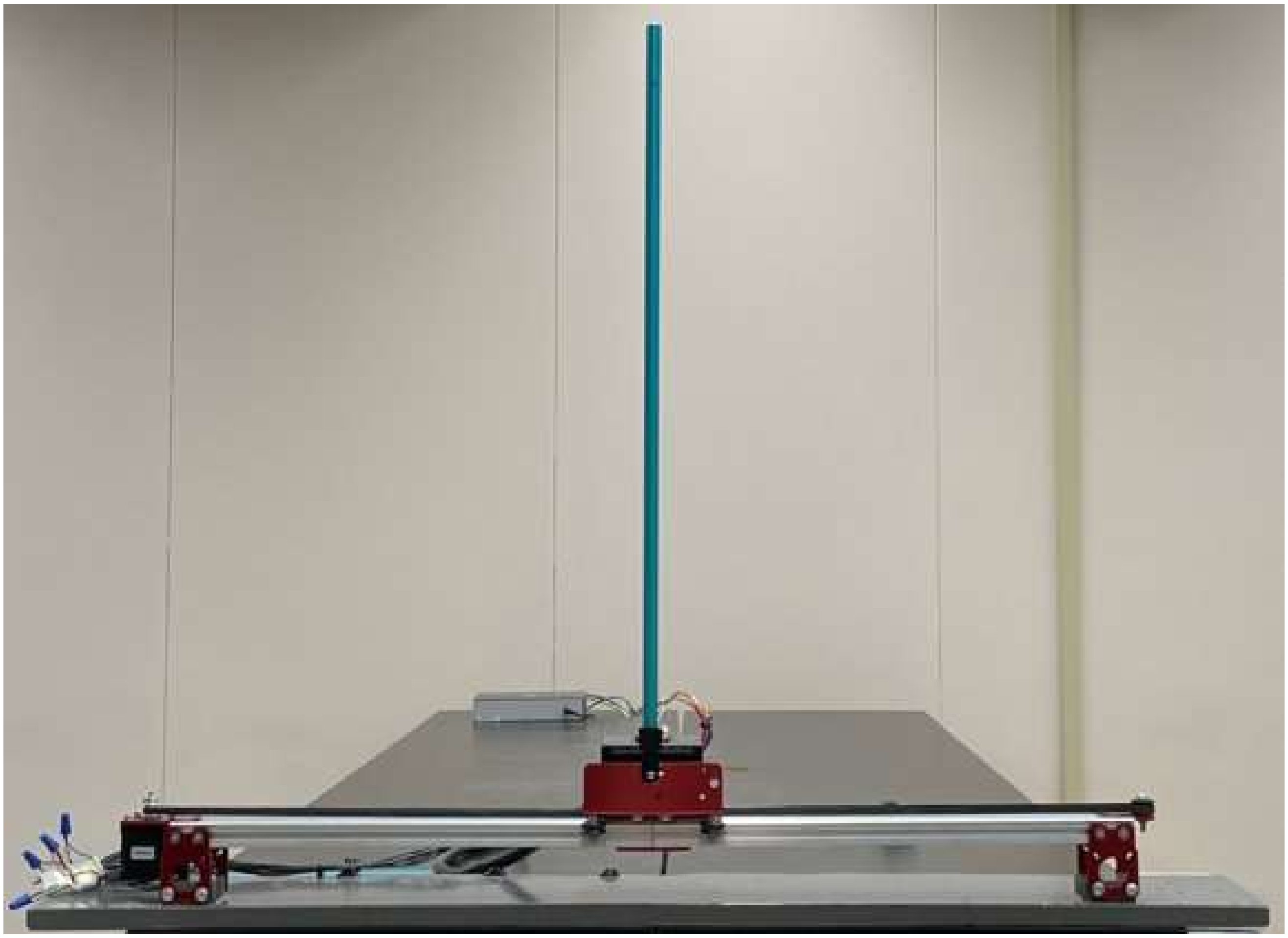
## NOTES

Nikolas Mahaffey and Charles Mashek deserve special recognition for their tremendous effort in helping us throughout the entirety of this project. They helped the student that was assigned this project last year, so they had parts and ideas on how to improve upon the system and shared everything they possibly could have with us. We cannot thank them enough for everything. Professor Phil Pash also deserves special recognition for his efforts in working with Professor Freije and us to perfect our system throughout the entire semester. He went above and beyond for us even though he was not one of our sponsors, so we sincerely want to thank Professor Pash for his efforts.

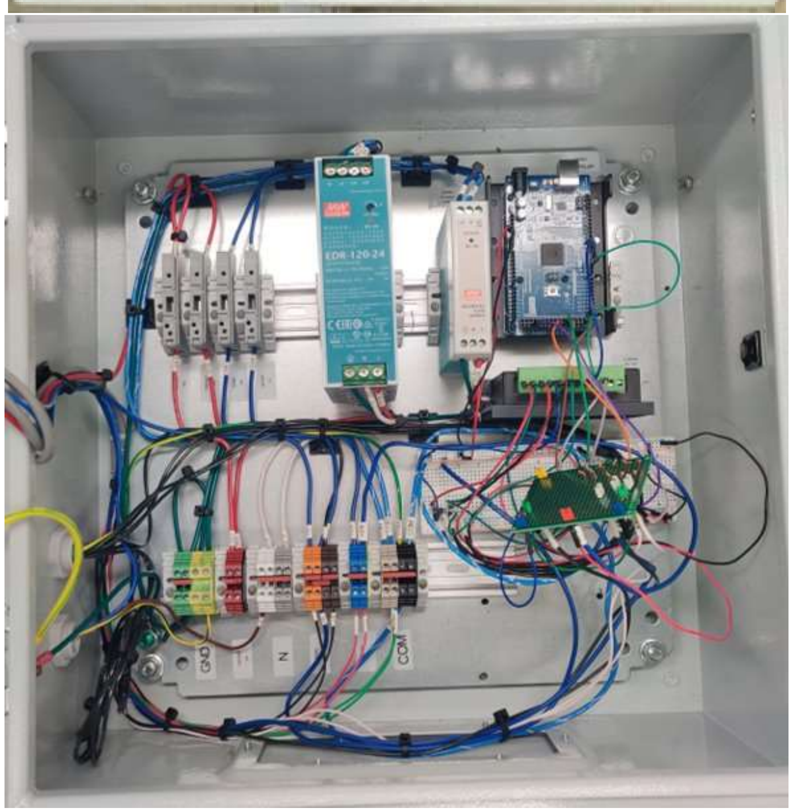
There were several sections of this project that could be improved to better the overall system and make it more consistent than it is in its current state. One example would be to replace the wheels to make them wider in order for them to have a snugger fit on the track and ride smoother. Another would be to upgrade to a larger panel so that future students can easily work inside of this box. Incorporating limit switches would also help the system so that it does not keep trying to move past the edge of the borders. We would advise that whoever works on this in the future attempts to take a more calculated approach with transfer functions for every step in our control diagram.

## APPENDICES

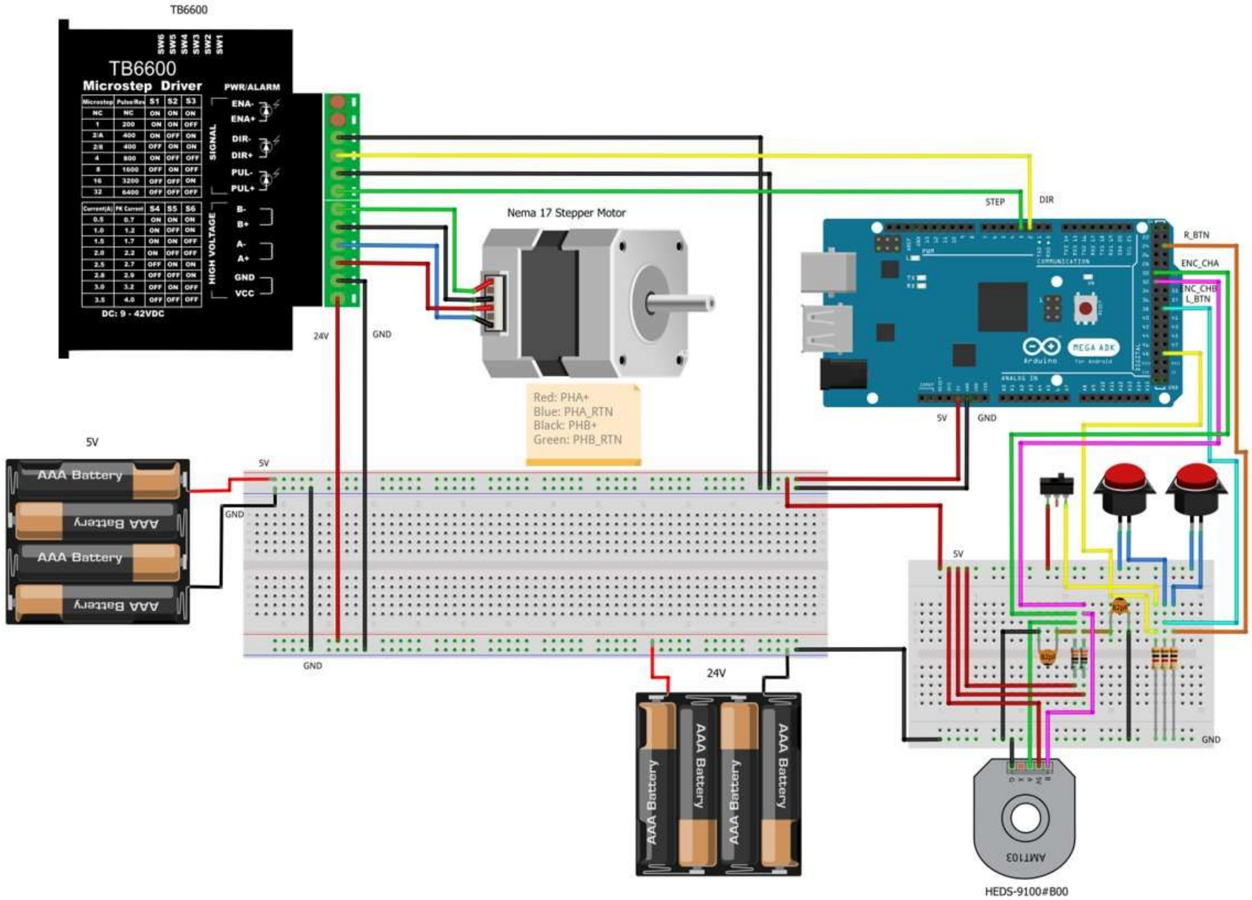
### *Final Track*



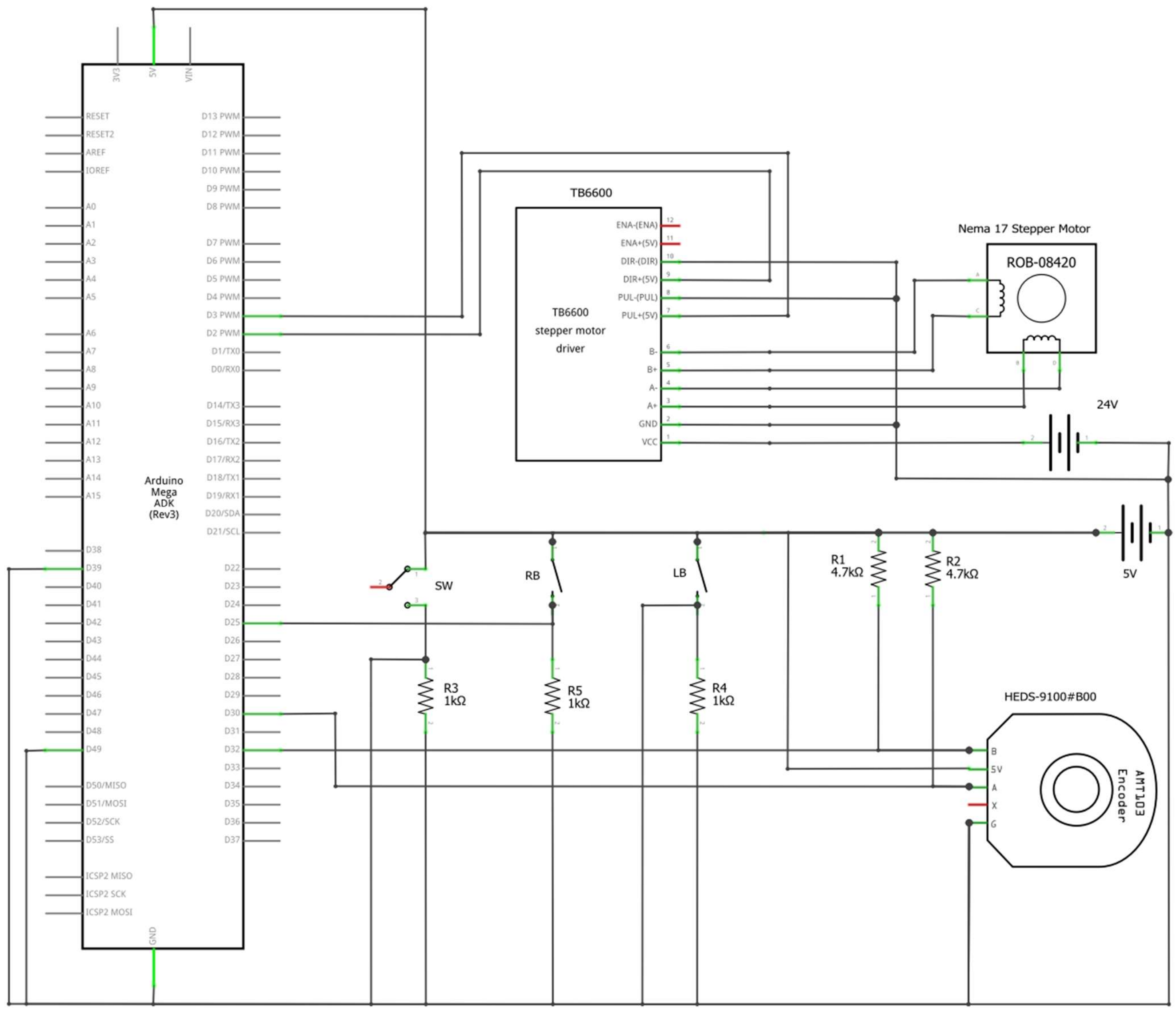
***Final Enclosure***

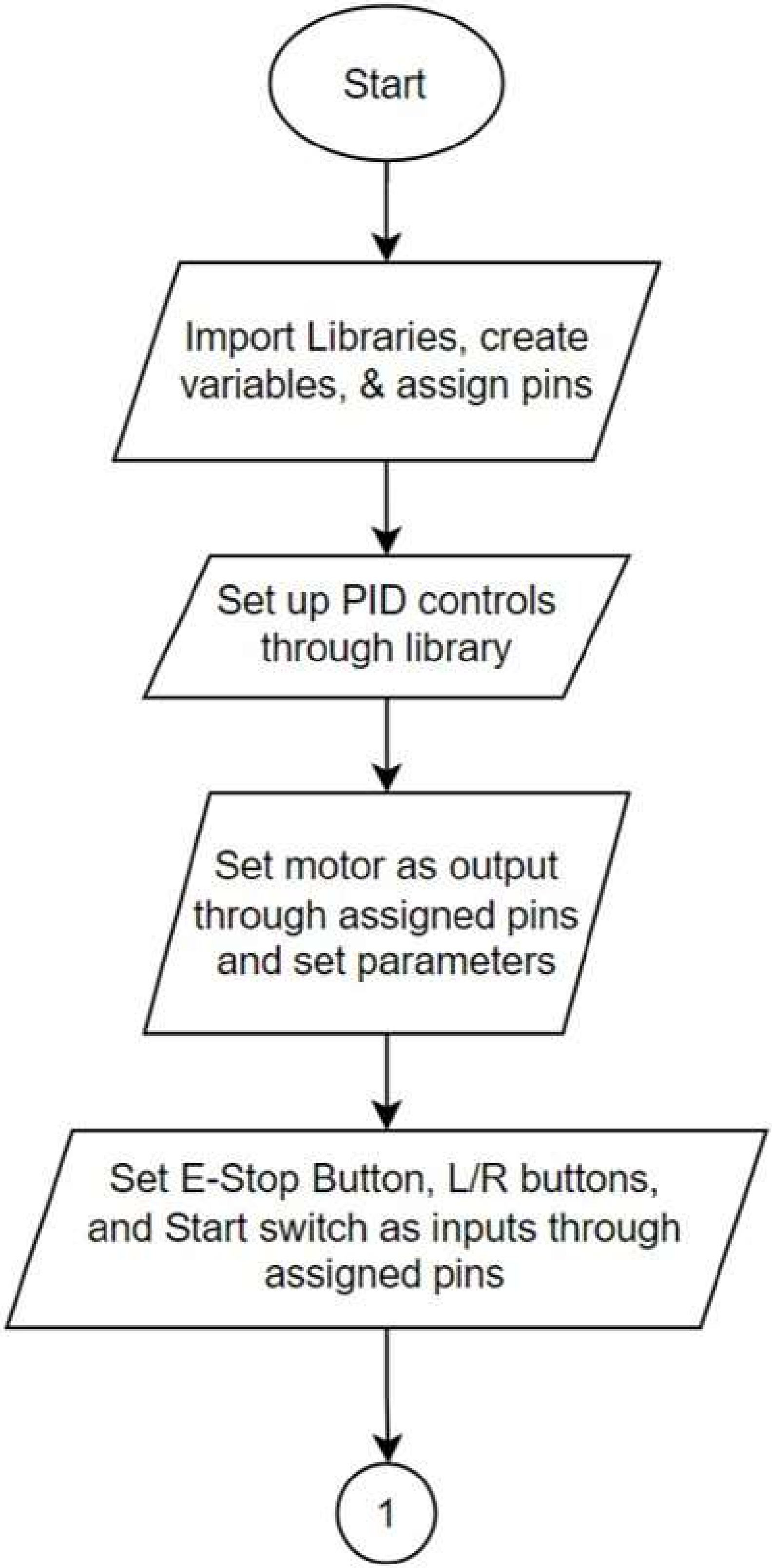
### *Fritzing of System: Diagram*

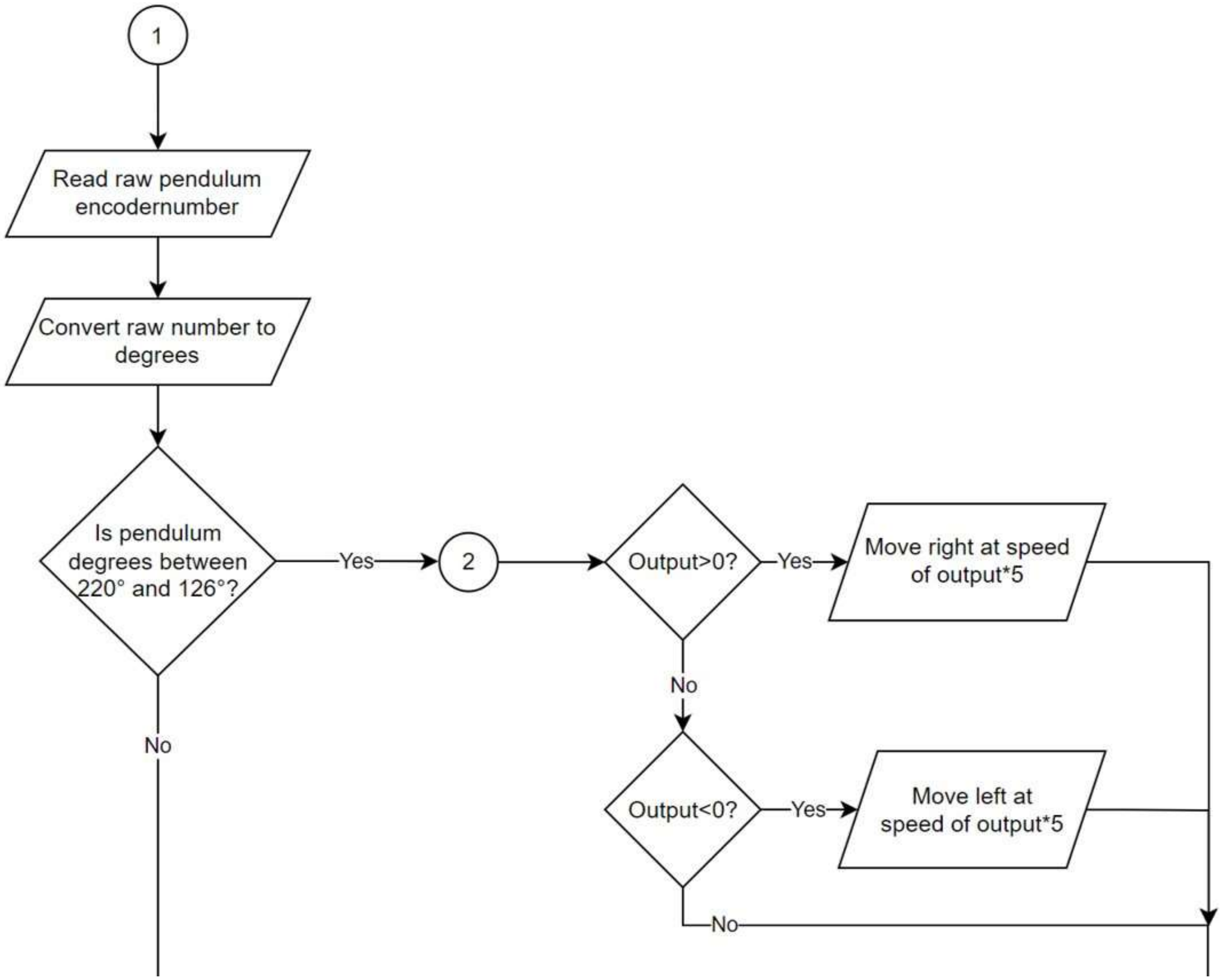


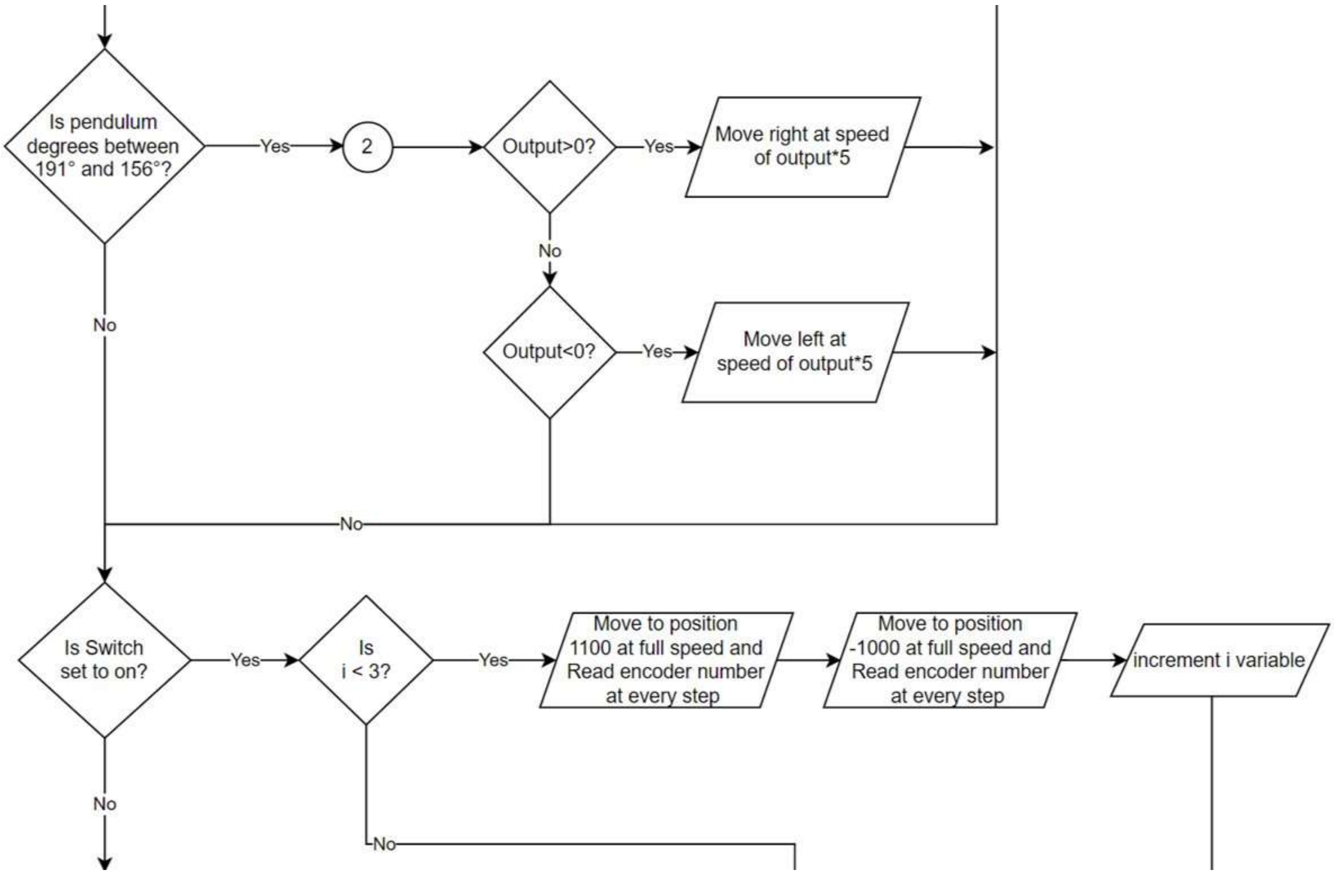
## *Fritzing of System: Schematic*

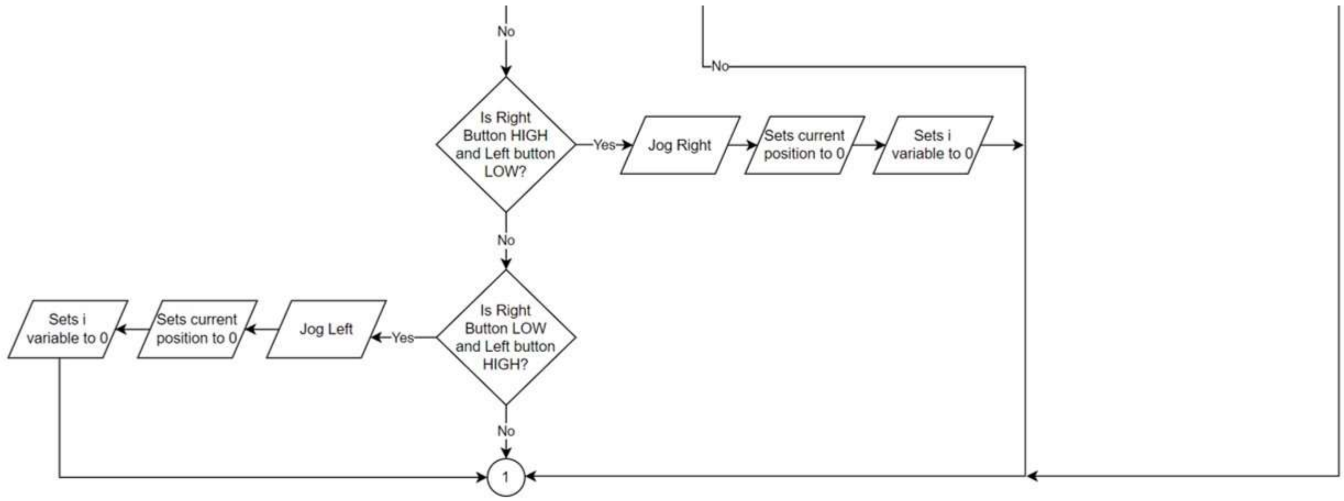


***Software Flow Chart***

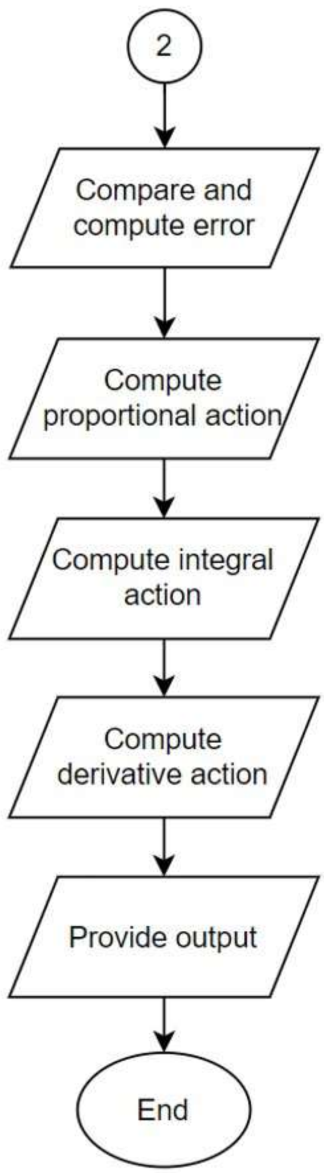



1
Read raw pendulum encodernumber
Convert raw number to degrees
Is pendulum degrees between 220° and 126°?
Yes
No
2
Output>0?
Yes
Move right at speed of output*5
No
Output<0?
Yes
Move left at speed of output*5
No


Is pendulum degrees between 191° and 156°?
Yes
2
Output>0?
Yes
Move right at speed of output*5
No
Output<0?
Yes
Move left at speed of output*5
No
No
Is Switch set to on?
Yes
Is i < 3?
Yes
Move to position 1100 at full speed and Read encoder number at every step
Move to position -1000 at full speed and Read encoder number at every step
increment i variable
No
No



No
No
Is Right Button HIGH and Left button LOW?
Yes
Jog Right
Sets current position to 0
Sets i variable to 0
No
Is Right Button LOW and Left button HIGH?
Yes
Jog Left
Sets current position to 0
Sets i variable to 0
No
1

2
Compare and compute error
Compute proportional action
Compute integral action
Compute derivative action
Provide output
End

***Final Main Code***

```
#include <Arduino.h>

#include <Encoder.h>      // Found in the Arduino Library Installer Version 1.4.2 by Paul Stoffregen

#include <PID_v1.h>       // Found in the Arduino Library Installer Version  1.2.1 by Brett Beauregard

#include <AccelStepper.h> // Found in the Arduino Library Installer Version  1.64 by Mike McCauley


/*

   Inverted Pendulum Senior Project Spring 2023

   Programmer(s): Sergio Vergara, Hawra Aljishi

   Initial Code Referenced by: Charles "Jake" Masheck, Nikolas Mahaffey


   Goal: The purpose of this project is to design and build an inverted pendulum system from an existing
nonworking

   inverted pendulum system for the Engineering and Technology Department.

   Once completed, this system will be used for educational purposes in order for future students to

   observe code, understand the design, and study the physics of the system.

*/


// Pendulum/Encoder variables

Encoder Pendulum(30, 32);  // (A, B)

const long pendulumCPR = 4095;  // Encoder counter per revolution, 4095


// define variables for running the motor

const int jogSpeed = 1300;   // Set the speed that the cart moves in manual mode, was 1300

const long acceleration = 16000; // was 16000


// define pins for motor

const int stepPin = 3;  // Breadboard - 7 or System - 2

const int dirPin = 2;  // Breadboard - 6 or System - 5

const int right = 25;

const int left = 39;
```

```
const int swBtn = 49;

// Variable for running
int shouldRun = 0;
int toggleDir = 0;
int i = 0;

AccelStepper stepper(1, stepPin, dirPin); // (Typeof driver: with 2 pins, STEP, DIR)

// PID Configuration
double setpoint = 177.40; //This is the vertical position of the pendulum., was 179.40,176.40


// Project documentation on circuitdigest.com to learn how to set these values
double Kp = 950;    // Proportional   Tune 1st
double Ki = 1.25;   // Integral       Tune 3rd
double Kd = .5;     // Derivative     Tune 2nd


double input, output;
PID pid(&input, &output, &setpoint, Kp, Ki, Kd, DIRECT);  //setup the PID control IOs


void setup()
{
  // Motor Parameters and pins
  stepper.setMaxSpeed(4000);
  stepper.setAcceleration(acceleration);
  stepper.setSpeed(jogSpeed);
  //stepper.moveTo(800);   // initial target position
  pinMode(stepPin,OUTPUT);
  pinMode(dirPin,OUTPUT);
```

```
  // Set up the manual buttons
  pinMode(right, INPUT);
  pinMode(left, INPUT);
  pinMode(swBtn,INPUT);

  // PID parameters
  pid.SetMode(AUTOMATIC);
  pid.SetSampleTime(10);
  pid.SetOutputLimits(-255, 255);

  // Serial monitor for debugging
  // Serial.begin(115200);
}

long positionPend  = 100;  // set to 100 allows for the program to detect change/initial position

void loop()
{
  // Encoder Readings //
  static float pendDegrees = 0;
  static long newPend;
  newPend = Pendulum.read();


  // Read pendulum position and convert to degrees //
  if (newPend != positionPend)
  {
    positionPend = newPend;

    if (positionPend > pendulumCPR)       // Reset if the pendulum goes all the way around in the positive
direction
```

```
    {
      Pendulum.write(0);
    }
    if (positionPend < (-1*pendulumCPR))  // Reset if the pendulum goes all the way around in the negative direction
    {
      Pendulum.write(0);
    }
    if (positionPend > 0)                  // Count up and map the values to degrees in the CC direction
    {
      pendDegrees = map(positionPend * 10, 0, (pendulumCPR *10), 0, 3590);
    }
    if (positionPend < 0)                  // Count down and map the values to degrees in the CW direction
    {
      pendDegrees = map(positionPend * 10, 0, (pendulumCPR *-10), 3590, 0);
    }
    pendDegrees = pendDegrees / 10.0;      // Divide by 10 to get degrees From 0 to 360 with decimal resolution
  }
  //Serial.println(pendDegrees);

  // BALANCE CHECK
  if (pendDegrees <= 220 && pendDegrees >= 126)  // If the pendulum is +- degrees from the setpoint, was 185-175, 192-162, 220-130
  {
    input = pendDegrees; //set the input of the PID to the pendulum degrees
    pid.Compute(); //Compute the PID output control values

    if (output > 0) //If the PID output is positive
    {
```

```
    // move the motor to the right
    stepper.setSpeed(output*5);
    stepper.runSpeed();
    stepper.setCurrentPosition(0);
  }
  else if (output < 0)  //If the PID output is negative
  {
    // move the motor to the left
    stepper.setSpeed(output*5);
    stepper.runSpeed();
    stepper.setCurrentPosition(0);
  }
 }
 if (pendDegrees <= 191 && pendDegrees >= 156)  // If the pendulum is +- degrees from the setpoint, was
185-175, 192-162, 220-130
 {
  input = pendDegrees; //set the input of the PID to the pendulum degrees
  pid.Compute(); //Compute the PID output control values

  if (output > 0) //If the PID output is positive
  {
    // move the motor to the right
    stepper.setSpeed(output*5);
    stepper.runSpeed();
    stepper.setCurrentPosition(0);
  }
  else if (output < 0)  //If the PID output is negative
  {
    // move the motor to the left
    stepper.setSpeed(output*5);
```

```
      stepper.runSpeed();
      stepper.setCurrentPosition(0);
    }
  }

  if(digitalRead(swBtn))
  {
    if( i < 3)
    {
      // Move motor to the right
      stepper.moveTo(1100);
      while (stepper.currentPosition() != 1050) // Full speed up to that point, was 1050
      {
        stepper.run();
        newPend = Pendulum.read();
      }
      stepper.runToPosition();
      newPend = Pendulum.read();


      // Now go to the left
      stepper.moveTo(-1000);
      while (stepper.currentPosition() != 0) // Full speed basck to 0
      {
        stepper.run();
        newPend = Pendulum.read();
      }
      stepper.runToPosition();
      stepper.stop(); // Stop as fast as possible: sets new target
```

```
      newPend = Pendulum.read();
      i=i+1;
    }
  }
  else if((digitalRead(right)==1) && (digitalRead(left)==0))
  {
    stepper.setSpeed(jogSpeed);
    stepper.runSpeed();
    stepper.setCurrentPosition(0);
    i = 0;
  }
  else if((digitalRead(right)==0) && (digitalRead(left)==1))
  {
    stepper.setSpeed(-jogSpeed);
    stepper.runSpeed();
    stepper.setCurrentPosition(0);
    i = 0;
  }
}
```

***Datasheets***

Stepper Motor: https://hobbycomponents.com/motors-and-servos/701-nema-17-stepper-motor

https://media.digikey.com/pdf/Data%20Sheets/Seeed%20Technology/316020003_Web.pdf

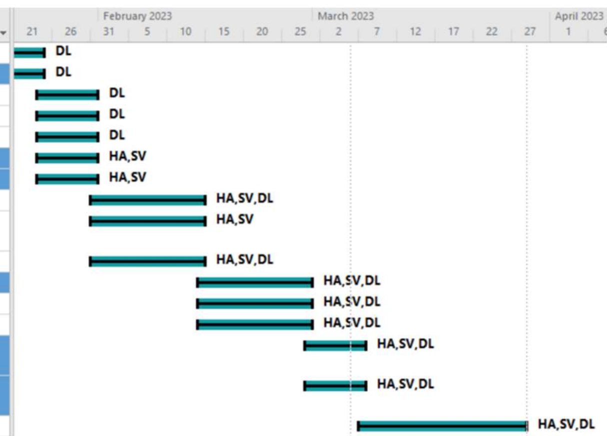
Stepper Motor Driver: https://www.makerguides.com/wp-content/uploads/2019/10/TB6600-Manual.pdf

Encoder: https://docs.broadcom.com/doc/AV02-1867EN

Microcontroller: https://docs.arduino.cc/static/c83705fd057244880b8eec3a692146ac/A000067-datasheet.pdf

***Gant Chart***

| Task Name | Duration | Start | Finish | % Complete | Plan to Finish |
|---|---|---|---|---|---|
| Tore down old pendulum and rebuilt new | 6 days | Tue 1/17/23 | Tue 1/24/23 | 100% | Wed 4/12/23 |
| Work on Power Supply connection | 6 days | Tue 1/17/23 | Tue 1/24/23 | 100% | NA |
| Work on Motor connection | 6 days | Tue 1/24/23 | Tue 1/31/23 | 100% | Tue 3/14/23 |
| Work on Encoder connection | 6 days | Tue 1/24/23 | Tue 1/31/23 | 100% | Tue 3/14/23 |
| Work on attaching the Cart to the system | 6 days | Tue 1/24/23 | Tue 1/31/23 | 100% | Wed 4/12/23 |
| Work on attaching the Pendulum to the cart | 6 days | Tue 1/24/23 | Tue 1/31/23 | 100% | NA |
| Work on Microcontroller connection | 6 days | Tue 1/24/23 | Tue 1/31/23 | 100% | NA |
| Program development cycle worksheet | 11 days | Tue 1/31/23 | Tue 2/14/23 | 100% | Wed 4/12/23 |
| Update FlowCharts, Block Diagrams, and any other documents for hardaware and software | 11 days | Tue 1/31/23 | Tue 2/14/23 | 100% | Tue 3/14/23 |
| Complete Microcontroller Code | 11 days | Tue 1/31/23 | Tue 2/14/23 | 100% | Sat 4/15/23 |
| Verify System powered from wall outlet (120VAC) | 11 days | Tue 2/14/23 | Tue 2/28/23 | 100% | NA |
| Verify that the pendulum can hold its upright position | 11 days | Tue 2/14/23 | Tue 2/28/23 | 100% | Sat 4/15/23 |
| Verify Manual Movement (Right & Left movments) | 11 days | Tue 2/14/23 | Tue 2/28/23 | 100% | Tue 3/14/23 |
| Verify process fully shuts down automatic operation of system using E-Stop Switch | 6 days | Tue 2/28/23 | Tue 3/7/23 | 100% | NA |
| Verify that the motor starts moving the cart using Start Switch | 6 days | Tue 2/28/23 | Tue 3/7/23 | 100% | NA |
| Improvement time | 16 days | Tue 3/7/23 | Tue 3/28/23 | 100% | Thu 4/20/23 |



*Bill of Materials*

**Purchased Supplies**

| Inverted Pendulum On a Cart BOM (Purchased Supplies) | | | | | | |
|---|---|---|---|---|---|---|
| QTY | Item | Part # | Brand | Vendor | Cost (Each) | Cost (Total) |
| 1 | 120 Vac - 24 Vdc Power Supply | EDR-120-24 | Mean Well | Amazon | $35.67 | $35.67 |
| 1 | 120 Vac - 5 Vdc Power Supply | MDR-10-5 | Mean Well | Amazon | $18.49 | $18.49 |
| 1 | Terminal Block Kit | 737123111329 | Dinkle | Amazon | $28.99 | $28.99 |
| 1 | Din Rail | D357A11-305(2) | International Connector | Amazon | $9.99 | $9.99 |
| 1 | Self Tapping Screws | TRUS-ZIN850-100 | Strong-Point | Amazon | $7.00 | $7.00 |
| 1 | Enclosure w/ Back Plate | 16"x16"x8" SPT | Vevor | Amazon | $69.52 | $69.52 |
| 1 | Zip Tie Mounts w/ Zip Ties | B08HGKVMSQ | Wahsure | Amazon | $10.99 | $10.99 |
| 1 | Fuse Holders | A17031000ux0614 | Uxcell | Amazon | $9.99 | $9.99 |
| 1 | E Stop Button | a12082000ux0339 | Uxcell | Amazon | $9.62 | $9.62 |
| 1 | Start Button | JH54023-R | Joyho | Amazon | $10.99 | $10.99 |
| 1 | Relay | 741416011418 | Mxuteuk | Amazon | $12.98 | $12.98 |
| 1 | Manual Buttons | 2SMF24E-22B | DMWD | Amazon | $13.99 | $13.99 |
| 1 | Terminal End Blocks | RTBESEW35-20 | FMHXG | Amazon | $8.59 | $8.59 |
| 1 | 120mm Grab Handles | AAB-PULL-K120-2 | PZRT | Amazon | $12.99 | $12.99 |
| 1 | 120 Vac Limit Switches | B07CM6Y3J8 | TWTADE | Amazon | $9.99 | $9.99 |
| 1 | Velcro Strips | 90975W | VELCRO | Amazon | $7.25 | $7.25 |
| 1 | Terminal End Covers | DK2.5N-BK | Dinkle | Amazon | $7.99 | $7.99 |
| 1 | Terminal Jumpers | DSS2.5N-10P | Dinkle | Amazon | $26.02 | $26.02 |
| 1 | Terminal Blocks | DK2.5N-BL | Dinkle | Amazon | $14.99 | $14.99 |
| 1 | Rotary Encoder | HEDS-9100# J00 | Broadcom Limited | Digi-Key | $21.40 | $21.40 |
| 4 | Grounding Bracelet | B08CXQN86W | Arctic Eagle | Amazon | $6.99 | $27.96 |
| 1 | Stepper Motor Driver | TB6600 | Usongshine | Amazon | $9.98 | $9.98 |
| | | | | | | $0.00 |
| | | | | Total Price | | $385.38 |

**Pre-Owned Supplies**

| Inverted Pendulum On a Cart BOM (Pre-Owned Supplies) | | | | | | |
|---|---|---|---|---|---|---|
| QTY | Item | Part # | Brand | Vendor | Cost (Each) | Cost (Total) |
| 1 | Arduino MEGA Microcontroller | 2560 R3 | Arduino | eBay | $32.00 | $32.00 |
| 1 | Wire | 16AWG50-6OFC | GS Power | Amazon | $28.99 | $28.99 |
| 1 | Stepper Motor | 42BYGH40-401A | Seeedstudio | Seeedstudio | $14.50 | $14.50 |
| 1 | Stepper Motor Driver | A4988 | Unahtinr | WalMart | $7.94 | $7.94 |
| 1 | Motor Encoder | Unknown | Unknown | eBay | $11.00 | $11.00 |
| 1 | Pendulum | Unknown | Quanser | Quanser | $500.00 | $500.00 |
| 1 | Pendulum Base | Unknown | Unknown | Unknown | $50.00 | $50.00 |
| 1 | 5A Fuse | GMA-5-R | Bussmann | Amazon | $4.29 | $4.29 |
| 1 | 2A Fuse | GMA-2-R | Bussmann | Amazon | $5.99 | $5.99 |
| 1 | 4-Wire Shielded Cable | 3244 | West Penn Wire | Amazon | $16.99 | $16.99 |
| 1 | Charcoal Gray Spray Paint | 7784830 | Rust-Oleum | Amazon | $6.98 | $6.98 |
| 1 | Labels | NA | Panduit | Kirby Risk | $25.00 | $25.00 |
| 1 | 1/2" Cord Grip Connectors | 1/2-2. | Otdorpatio | Amazon | $9.99 | $9.99 |
| 1 | 11-Conductor Cable | NA | Southwire | Kirby Risk | $20.00 | $20.00 |
| 1 | Heat Shrink | ASD-123 | Eventronic | Amazon | $6.29 | $6.29 |
| 1 | Butt Connectors | TICONN-174 | TICONN | Amazon | $11.95 | $11.95 |
| 1 | Electrical Tape | 6132-BA-10 | Scotch | Amazon | $7.33 | $7.33 |
| 1 | 1.1K and 5k-Ohm Resistors | A-0005-C10 | E-Projects | Amazon | $7.00 | $7.00 |
| 1 | Protoboard | N01Aa-UMA09 | Miuzei | Amazon | $15.99 | $15.99 |
| | | | | | | $0.00 |
| | | | | Total Price | | $782.23 |