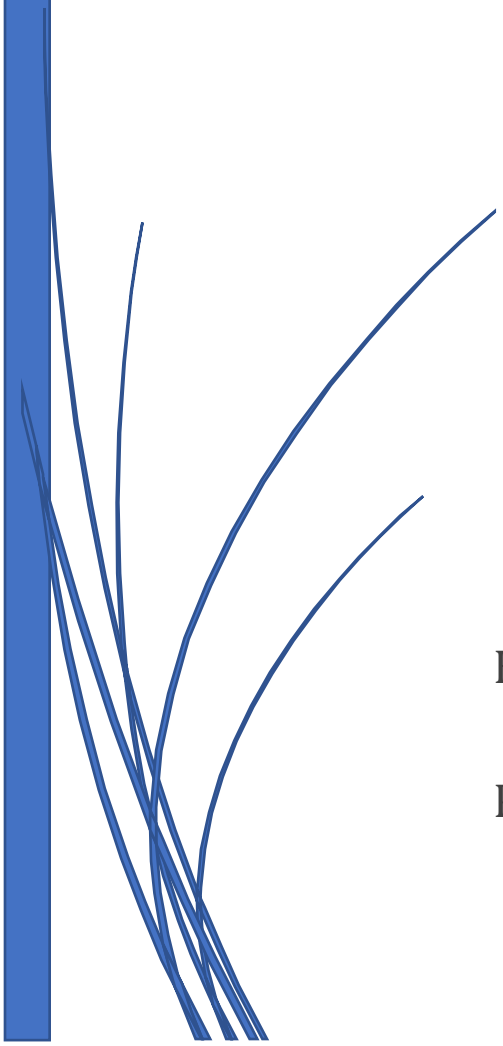
May 08, 2018

Architectural Robotics

ECE 4320, University of Missouri

Fall Detection & Assistance System



Hotrabhavananda, Benjarit

Professor: Dr. Marjorie Skubic

Table of Contents

Introduction

Fall is very common as people age and they find it difficult to stand up after a fall. The idea of this project is to detect when an older adult fall on the floor and provide assistance to help them get up from the floor. Our system is an easy-to-use assistive device to aid them to stand-up. Our project is aiming to help elders to live independently and safely.

Implementation

We have planted pressure sensors on floor to detect whether a person is walking or had a fall. When a person is walking on the floor, the sensors on the floor will be activated. If an elder had a fall, the number of activated sensors increases, which, in turn, activates the falling assistant.

When an elder fall on the floor, the system will calculate where the elder is falling. After the system has the location where the elder falls, the actuator will move the ring handle to that location. Once the ring arrives at the location, another actuator will work to stretch a rope that the ring is tight with down from the top of the house structure to the person. The person then can grab the ring and press a button that will pull his/her body up to the position that he/she feels confrontable. Once the elder gets up and stand, he/she could push another button that will reset the system back to its original position.

Mechanism

Motors

There are three stepper motors that are used in our project:

(1) x-axis motor: Used for moving the handle ring horizontally
(2) y-axis motor: Used for moving the handle ring towards inside or outside
(3) z-axis motor: Used for moving the handle ring vertically

Lights

There are six lights that are attached at the ceiling of the room. All six lights will brighten up when the system detects someone falls. Under normal condition, the lights will remain at the brightness of users' preference.

Buttons

There are two buttons and they are programmed for 2 purposes.

(1) Button 1: Reset the system

(2) Button 2: Pull up/down the ring hanger

Button use descriptions:

(1) Button 1: Reset the system

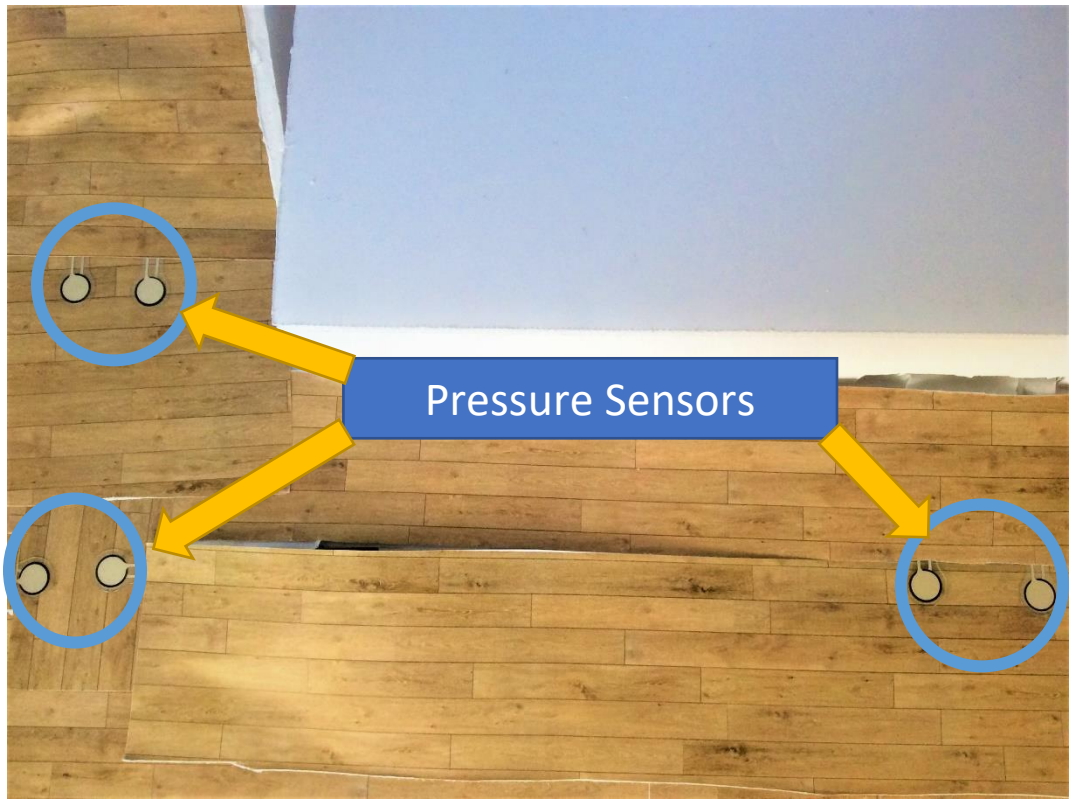
At any moment this button is pressed, the system will stop and reset itself to its original position. This feature is implemented to protect the system from mistakenly detect the fall when there is really nothing happens. Such scenario could be his/her grandchildren playing on the floor and two or more pressure sensors were triggered, leading up to activate the actuator. Another use of reset button is when the it finishes its job of helping an elder to get up.

(2) Button 2: Pull up/down Button

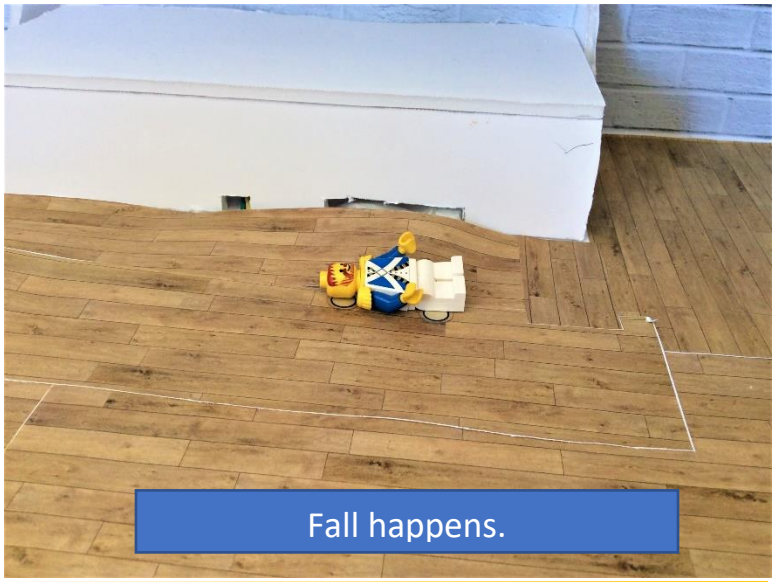
The ring hanger that comes down from the ceiling has a button that attached to it. An elder can use this button to help pull his/herself up. This feature is used when that person's legs get weak and unable to stand by his/herself.

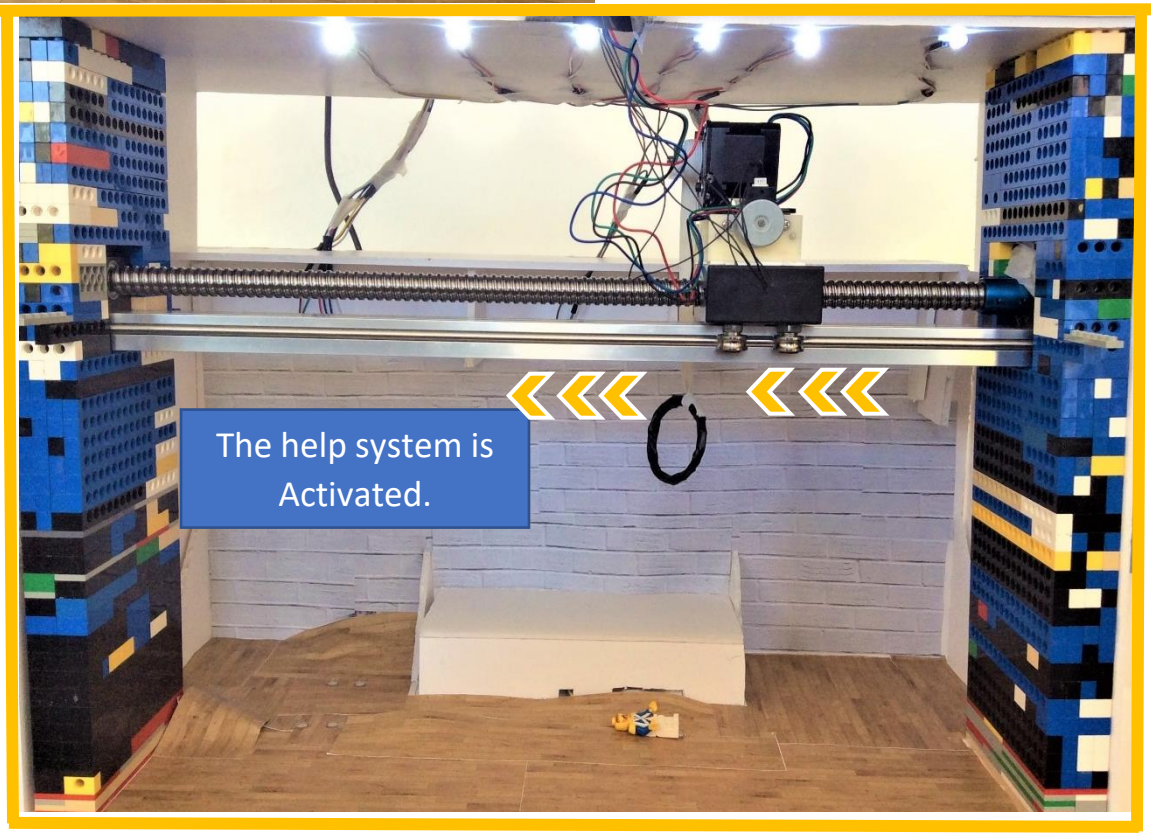
Photos

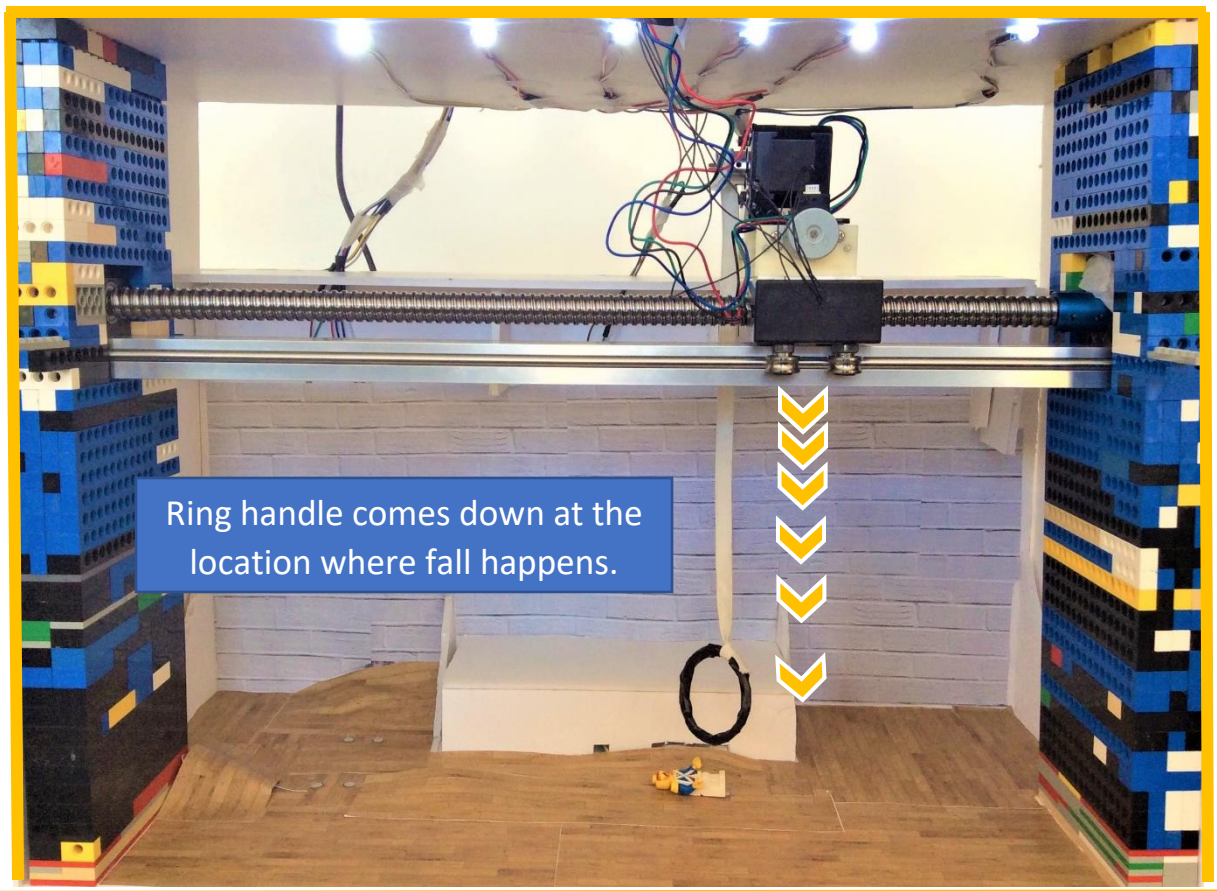

Fall Detection
& Assistance
System
Overall look of our project

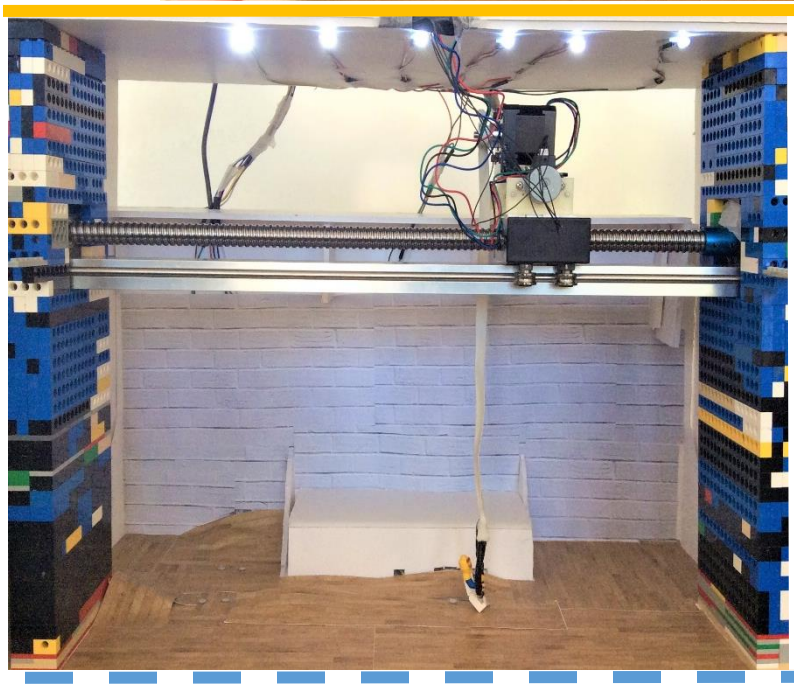

Pressure Sensors


Normal Condition.

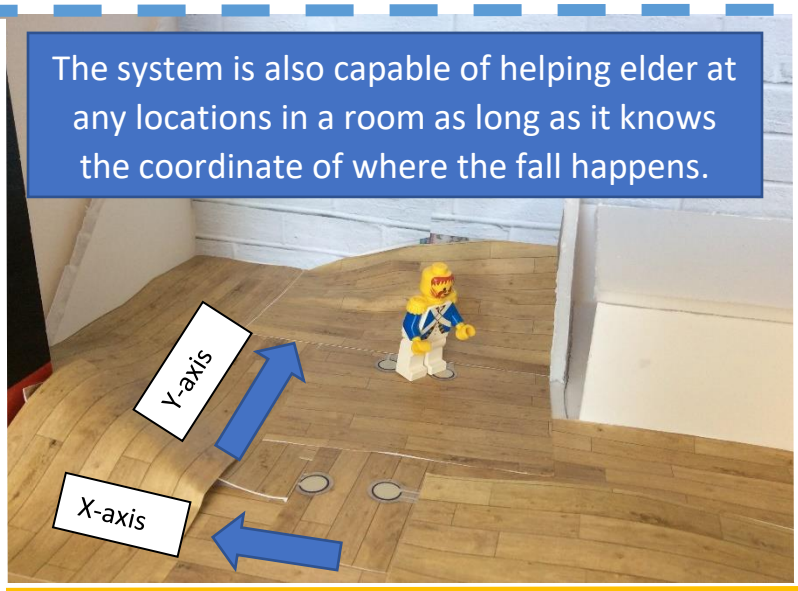

Fall happens.


The help system is Activated.

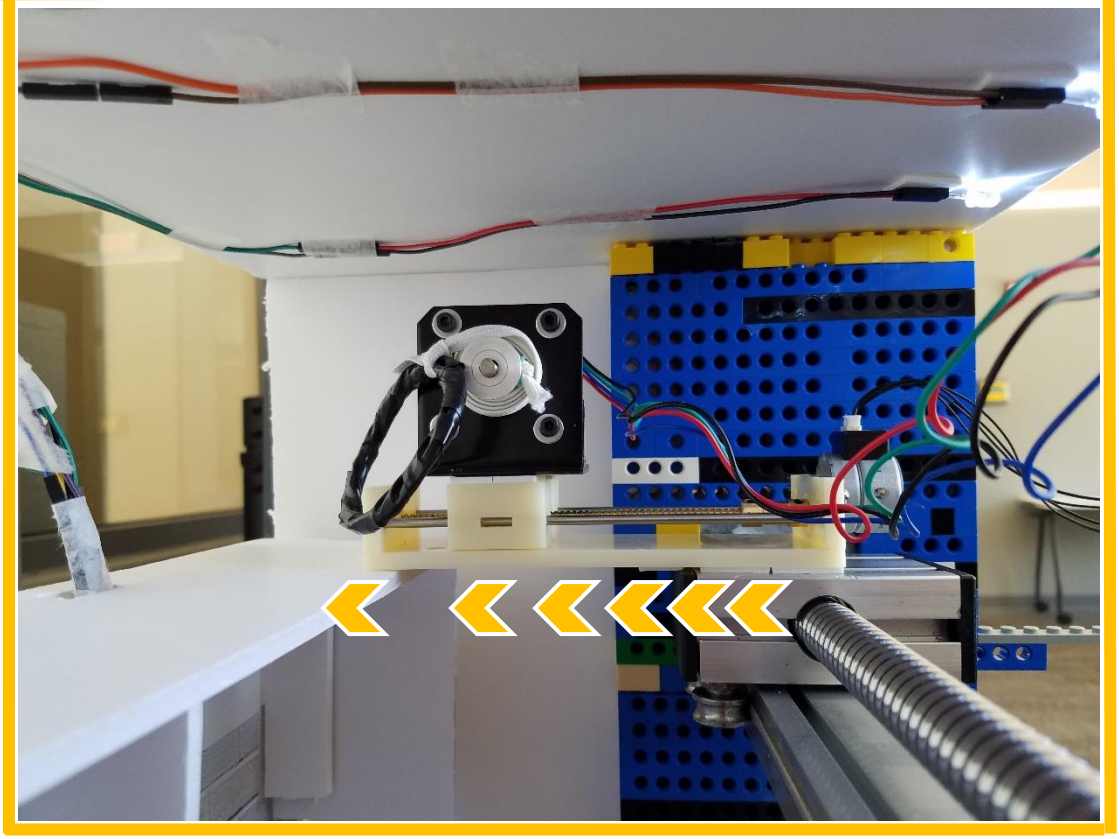

Ring handle comes down at the location where fall happens.

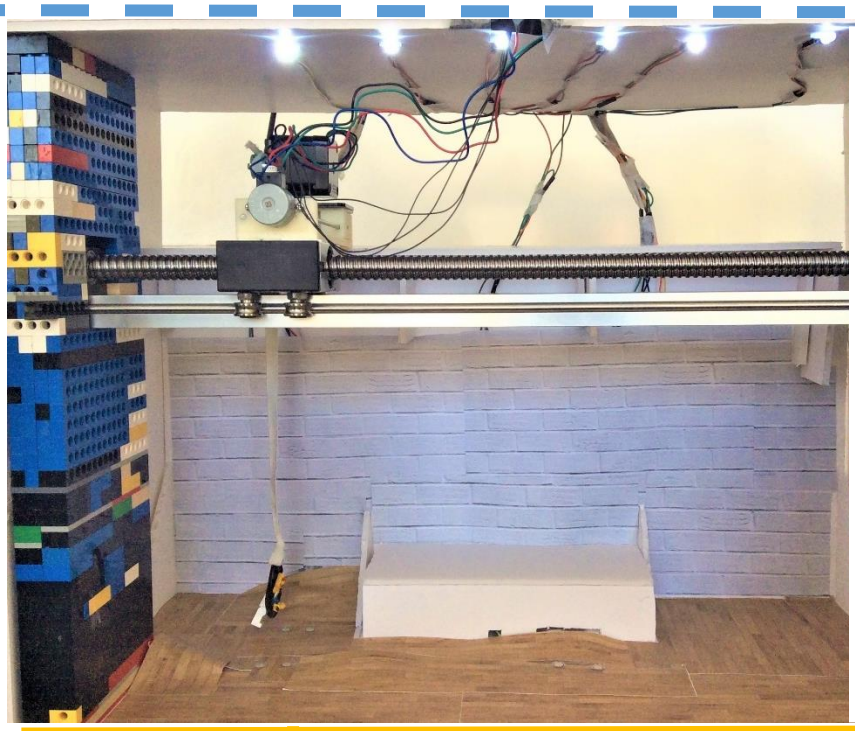


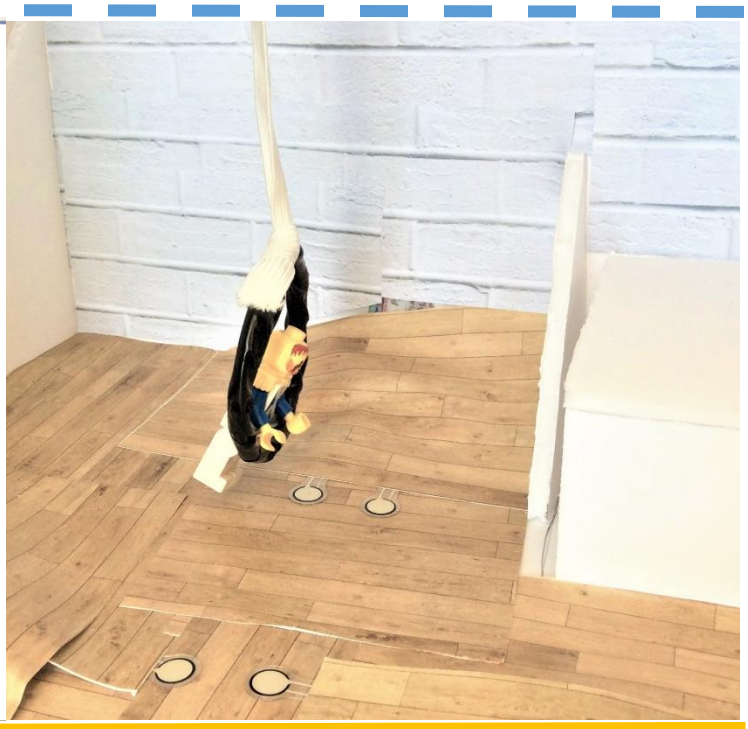

The elder is helped to get back up.

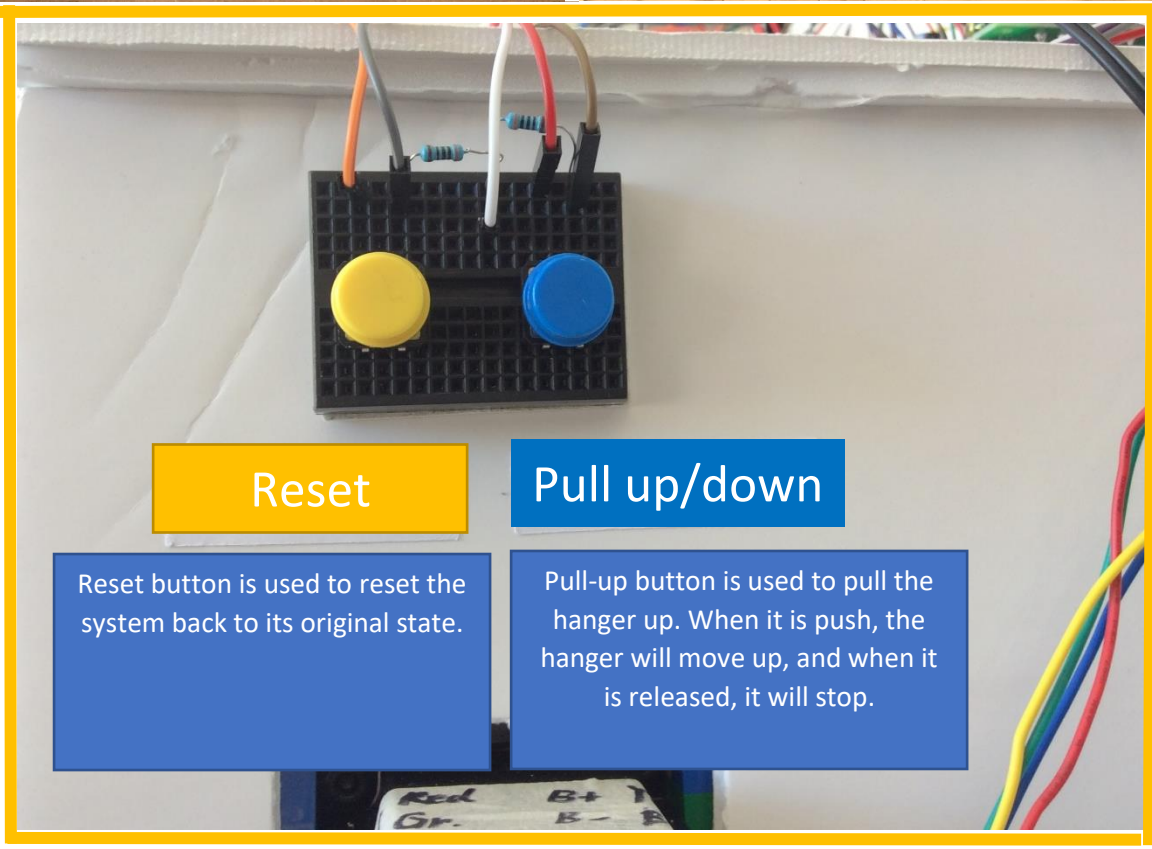

The system is also capable of helping elder at any locations in a room as long as it knows the coordinate of where the fall happens.
Y-axis
X-axis

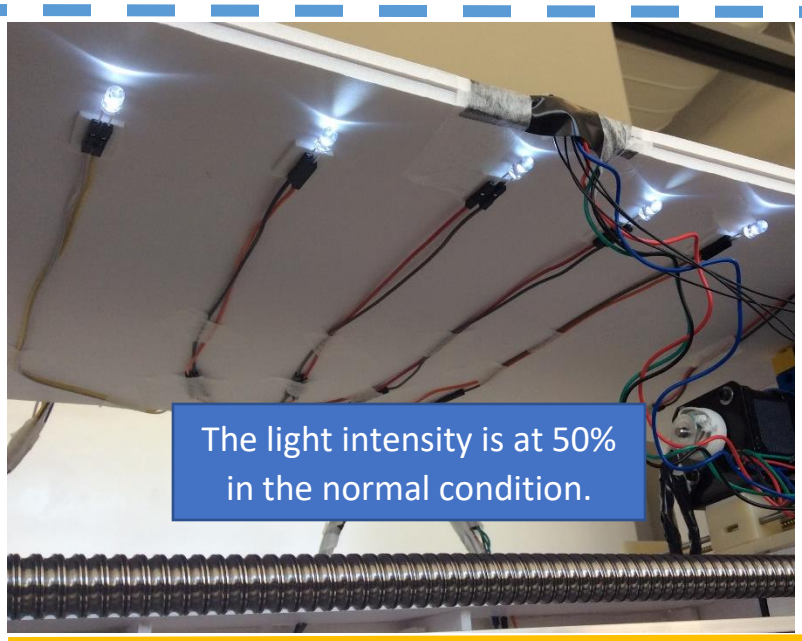

Somebody falls at the inner part of the room.

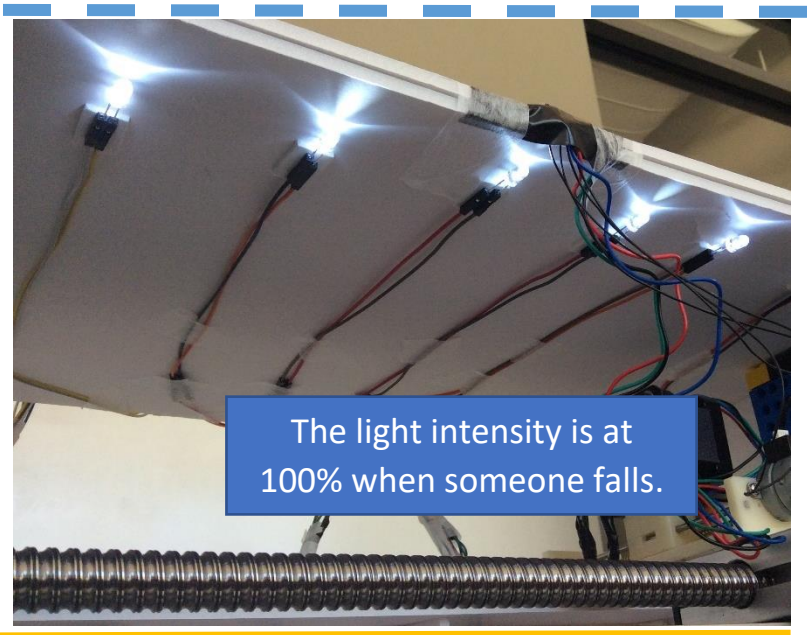


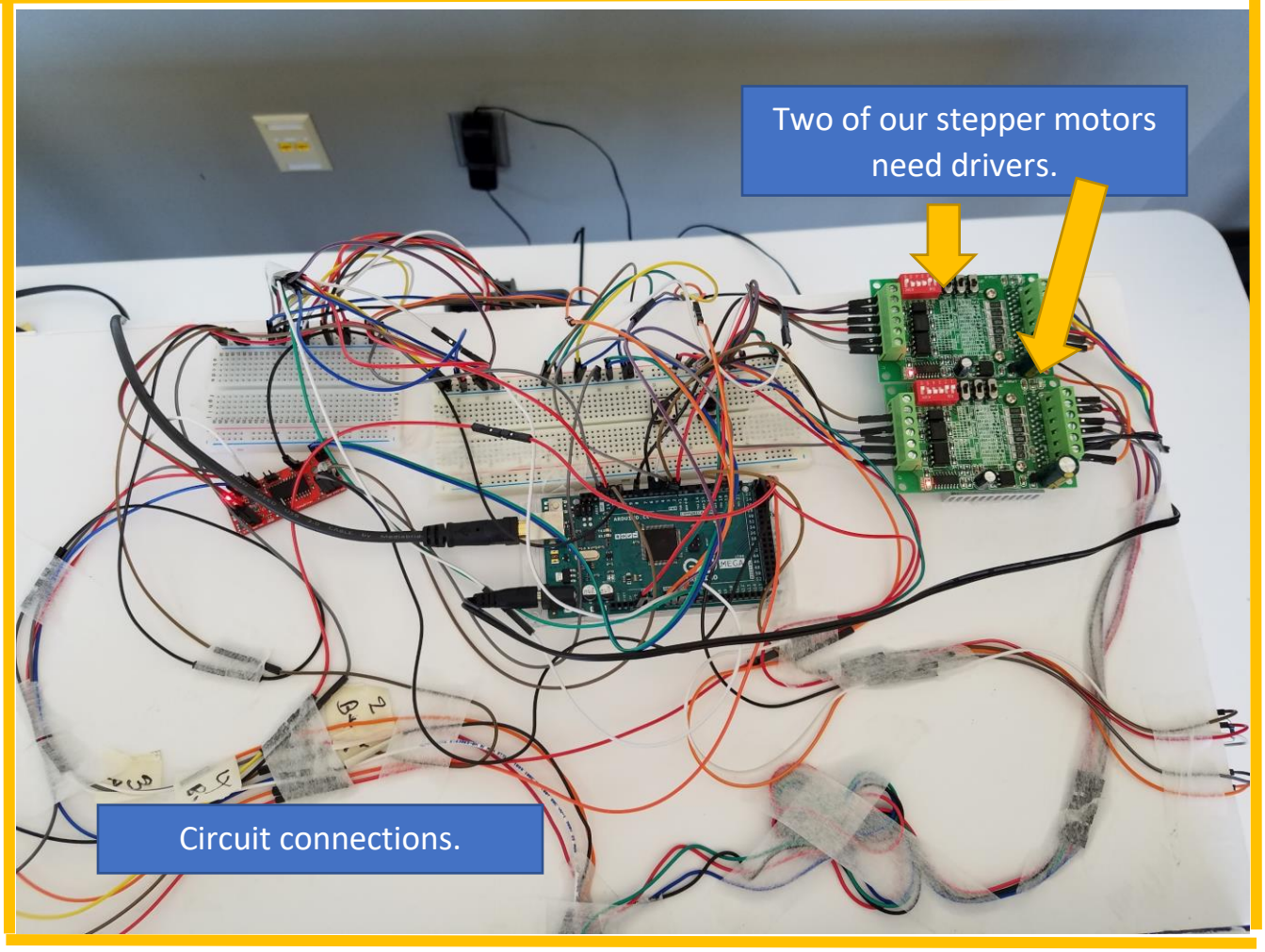





Reset
Reset button is used to reset the system back to its original state.
Pull up/down
Pull-up button is used to pull the hanger up. When it is push, the hanger will move up, and when it is released, it will stop.


The light intensity is at 50% in the normal condition.


The light intensity is at 100% when someone falls.


Two of our stepper motors need drivers.
Circuit connections.

Design progression

Detail on the final design

List of Equipment:

(1) 1 Arduino Mega

(2) 2 Arduino UNO

(3) 3 Bread Board

(4) Jumper wires

(5) 2 Buttons

(6) Pressure sensors

(7) LED Lights

(8) Form boards

(9) Legos

(10) Colored Printouts of walls

(11) Motor

(12) Motors' drivers

(13) Linear actuators

The entire home is built with foam board. The top of the room has the Arduino mega, 2 Arduino UNO, jumper wires, bread board etc. There are two linear actuators, one to move in x-axis and another to move in y-axis, this way we can cover the entire room in bigger scale. In our project we have demonstrated 3 floor locations with pressure sensors. Based on the fall location, the linear actuator will go to the target location. There are LED lights that indicate which direction the hanger is moving at that moment. Once the hanger reaches the (x,y) location on the ceiling using the linear actuators, the hanger finally comes down from the ceiling towards the ground in z direction. In larger scale the button can be placed on the hanger, so that it's convenient for the person to press the button as necessary.

How design meets the intended functionality

1. Our project has sensors: pressure sensor and actuators: 2 linear actuators, a motor and LED lights.
2. This project's target audience is the aging adults (above 65 years of age). As the older adults above 65 years tend to fall more and its very difficult for them to stand up on their own. It becomes more painful when the older adults stay in an independent house (this system can be installed pervasively). So, our project aims to help those older adults to stand up when they do not have anyone to help them stand up. Our system detects when someone had a fall and an assistive hanger moves to the target location. The older adult can help themselves stand up by holding the hanger.
3. Our project could be adjustable to be used with or without disabilities.

Solution tried and rationale for the chosen solution

Solution tried 1: We thought that there might be a scenario when kids might fall or just playing on the floor, and our system might get activated.

Chosen Solution 1: So, we have programmed a reset button. Anytime the reset button is pressed, our system will get back to its original state.

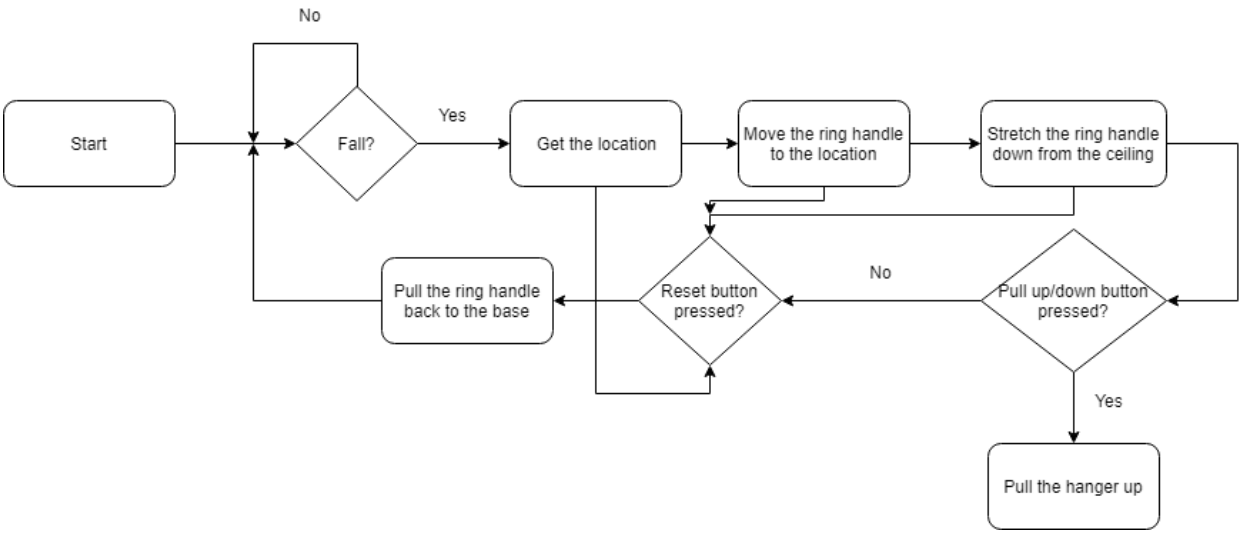
Solution tried - 2: Initially we had planned to place the pressure sensor in a straight line.

Chosen Solution - 2: Then we placed the pressure sensor in 3 different location, each location having 2 pressure sensors so that we could demonstrate the 2-dimensional movement of the linear actuator.

Solution tried - 3: At first, we thought of pulling the hanger automatically towards the ceiling once the person holds the hanger. But we thought it might be risky because there may be a case where the person might not have hold the hanger properly and the hanger will pull them up.

Chosen Solution - 3: So, we did a manual button for the hanger to move up. The person can press the button to pull the hanger towards the ceiling and when they release the button, the hanger stops moving. This way they can take their own time to get up as their convenience.

Code logic



We had programmed our system to archive the functionalities as describe above in the flow chart diagram.

Appendices

Project3.ino

This code is our main code for this project.

```
// defines pins numbers
const int stepPin = 5; //CLK+
const int dirPin = 2; //CW+ Directin of the motor
const int enPin = 8;

// rope motor
const int stepPin2 = 6; //CLK+
const int dirPin2 = 4; //CW+ Directin of the motor
const int enPin2 = 3;

const int ledPin = 13;

const int blue = 23;
const int green = 25;
const int yellow = 27;

const int interruptPin_CW = 18;
```

```
const int interruptPin_CCW = 19;
long X = 0;
long Y = 0;
long Z = 0;
bool is_reset = 0;
int position_var;
long target_step = 0;
long vertical_step = 0;


//small linear actuator
//Declare pin functions on Redboard
#define stp 22
#define dir 12
#define EN  11

// states variables that will be used in the program
long numberOfStep = 0;
int flag = 0;

//for pressure sensor
int fsrPin = 0;     // the FSR and 10K pulldown are connected to a0
int fsrPin1 = 1;
int fsrPin2 = 2;
int fsrPin3 = 3;
int fsrPin4 = 4;
int fsrPin5 = 5;

int fsrReading;     // the analog reading from the FSR resistor divider
int fsrReading1;
int fsrReading2;
int fsrReading3;
int fsrReading4;
int fsrReading5;



int Led_internsity = 64;
int reset2 = 0;
int reset3 = 0;
void setup() {
  //position_var = 1;
  Serial.begin(2000000);
  //LEDS
  pinMode(ledPin,OUTPUT);
  pinMode(blue,OUTPUT);
  pinMode(green,OUTPUT);
  pinMode(yellow,OUTPUT);
```

```
  // Sets the two pins as Outputs
  pinMode(stepPin,OUTPUT);
  pinMode(dirPin,OUTPUT);
  pinMode(enPin,OUTPUT);

  pinMode(stepPin2,OUTPUT);
  pinMode(dirPin2,OUTPUT);
  pinMode(enPin2,OUTPUT);


  pinMode(interruptPin_CW, INPUT);
  pinMode(interruptPin_CCW, INPUT);
  digitalWrite(enPin,LOW);
  digitalWrite(enPin2,LOW);
  //attachInterrupt(digitalPinToInterrupt(interruptPin_CW), VMotor_CW, RISING);
  //attachInterrupt(digitalPinToInterrupt(interruptPin_CCW), VMotor_CCW, RISING);

  //small linear actuator
  pinMode(stp, OUTPUT);
  pinMode(dir, OUTPUT);
  pinMode(EN, OUTPUT);
  resetEDPins(); //Set step, direction, microstep and enable pins to default states
  position_var = 1;
}
void loop(){

  //for pressure sensor//
  fsrReading = analogRead(fsrPin);
  fsrReading1 = analogRead(fsrPin1);
  fsrReading2 = analogRead(fsrPin2);
  fsrReading3 = analogRead(fsrPin3);
  fsrReading4 = analogRead(fsrPin4);
  fsrReading5 = analogRead(fsrPin5);
//
//  // LED = HIGH
    analogWrite(ledPin, Led_internsity);


    digitalWrite(EN, LOW); //Pull enable pin low to allow motor contro
      //StepForwardDefault();
      //VMotor_CW1(100000);
      //vertical_step = 1800;
      //reverse_vertical_motor();
      //if(position_var = 1) {


   if(fsrReading > 1 && fsrReading1 > 1) {
    reset2 = 0;
```

```
  reset3 = 0;
  vertical_step = 5900;
  Led_internsity = 255;
  analogWrite(ledPin, 255);
  Serial.println("Someone fell at location 0 & 1");
  target_step = 30667;
  Serial.print("target step = ");
  Serial.println(target_step);
  VMotor_CW1(target_step);
  position_var = 0;
  }

 else if(fsrReading2 > 1 && fsrReading3 > 1) {
  reset2 = 0;
  reset3 = 0;
  vertical_step = 5900;
  Led_internsity = 255;
  analogWrite(ledPin, 255);
  Serial.println("Someone fell at location 2 & 3");
  target_step = 105333;
  Serial.print("target step = ");
  Serial.println(target_step);
  VMotor_CW1(target_step);
  position_var = 0;
  }

 else if(fsrReading4 > 1 && fsrReading5 > 1) {
  reset2 = 0;
  reset3 = 0;
  vertical_step = 5900;
  Led_internsity = 255;
  analogWrite(ledPin, 255);
  Serial.println("Someone fell at location 4 & 5");
  target_step = 105333;
  Serial.print("target step = ");
  Serial.println(target_step);
  StepForwardDefault();
    if(reset3 != 1){
      VMotor_CW1(target_step);
      position_var = 0;
    }
  }

 else if(digitalRead(interruptPin_CCW) == 1 ){ // reset
  reset2 = 0;
  reset();
 }
 while(digitalRead(interruptPin_CW) == 1){ // reset
```

```
    reset2 = 0;
    digitalWrite(dirPin2,LOW); // MOVE TO THE LEFT SIDE.
      if(vertical_step <= 0 ){
        break;
      }
      Serial.print("Vertical steps ******** = ");
      Serial.println(vertical_step);
      digitalWrite(stepPin2,HIGH);
      delayMicroseconds(1500);
      digitalWrite(stepPin2,LOW);
      delayMicroseconds(1500);
      //Serial.print("current steps = ");
      //Serial.println(vertical_step);
      vertical_step--;
  }
  resetEDPins();
  delay (1000);
}



/*----------Function Calling----------*/


void VMotor_CW1(long target_step)
{
  digitalWrite(green, HIGH);
  Serial.print("target step = ");
  Serial.println(target_step);
  digitalWrite(dirPin,LOW ); // MOVE TO THE LEFT SIDE.
  // Makes 200 pulses for making one full cycle rotation
  for (X = X; X <= target_step; X++)
  {
    digitalWrite(stepPin,HIGH);
    delayMicroseconds(1);
    digitalWrite(stepPin,LOW);
    delayMicroseconds(1);
    Serial.print("current steps = ");
    Serial.println(X);

    if(digitalRead(interruptPin_CCW) == 1){
    reset2 = 1;
    reset();
    break;
   }
  }
  digitalWrite(green, LOW);
  if(reset2 != 1){
```

```
    vertical_motor();
  }
}

void reset() // to go to
{
  if(reset2 != 1){
     reverse_vertical_motor();
  }
  digitalWrite(dirPin,HIGH); //MOVE TO THE RIGHT SIDE.
  // Makes 400 pulses for making two full cycle rotation
  digitalWrite(green, HIGH);
  for (X = X-1; X >= 0; X--)
  {
    digitalWrite(stepPin,HIGH);
    delayMicroseconds(1);
    digitalWrite(stepPin,LOW);
    delayMicroseconds(1);
    Serial.println(X);
  }
  digitalWrite(green, LOW);
  Led_internsity = 64;
  X = 0;

  ReverseStepDefault();
}

//small linear actuator
void resetEDPins()
{
  digitalWrite(stp, LOW);
  digitalWrite(dir, LOW);
  digitalWrite(EN, HIGH);
}

//Default microstep mode function
void StepForwardDefault()
{

  Serial.println("Moving forward at default step mode.");
  digitalWrite(dir, LOW); //Pull direction pin low to move "forward"
  while(numberOfStep < 78000)  //Loop the forward stepping enough times for motion to be visible
  {
    digitalWrite(blue, HIGH);
    digitalWrite(stp,HIGH); //Trigger one step forward
    delayMicroseconds(350);
    digitalWrite(stp,LOW); //Pull step pin low so it can be triggered again
    delayMicroseconds(350);
```

```
   Serial.print("Invalid option entered: ");
   Serial.println(numberOfStep);
    numberOfStep++;
    if(digitalRead(interruptPin_CCW) == 1){
      reset3 = 1;
      ReverseStepDefault();
      break;
    }
  }
   digitalWrite(blue, LOW);
}
//Reverse default microstep mode function
void ReverseStepDefault()
{
  Serial.println("Moving in reverse at default step mode.");
//Reset Easy Driver pins to default states
  digitalWrite(dir, HIGH); //Pull direction pin high to move in "reverse"
  while(numberOfStep > 0)//Loop the stepping enough times for motion to be visible
  {
    digitalWrite(blue, HIGH);
    digitalWrite(stp,HIGH); //Trigger one step
    delayMicroseconds(350);
    digitalWrite(stp,LOW); //Pull step pin low so it can be triggered again
    delayMicroseconds(350);
    numberOfStep--;
  }
    digitalWrite(blue, LOW);
}


void vertical_motor()
{

  Serial.print("Vertical step = ");
  Serial.println(vertical_step);
  digitalWrite(dirPin2,HIGH); // MOVE TO THE LEFT SIDE.
  // Makes 200 pulses for making one full cycle rotation
  for (int x = 0; x <= vertical_step; x++)
  {
    digitalWrite(yellow, HIGH);
    digitalWrite(stepPin2,HIGH);
    delayMicroseconds(500);
    digitalWrite(stepPin2,LOW);
    delayMicroseconds(500);
    Serial.print("Vertical steps = ");
    Serial.println(x);
    if(digitalRead(interruptPin_CCW) == 1){
      vertical_step = x;
```

```
      reset();
      break;
    }
  }
  digitalWrite(yellow, LOW);
}
void reverse_vertical_motor()
{

  Serial.print("Vertical step = ");
  Serial.println(vertical_step);
  digitalWrite(dirPin2,LOW); // MOVE TO THE LEFT SIDE.
  // Makes 200 pulses for making one full cycle rotation
  for (int x = vertical_step; x > 0; x--)
  {
    digitalWrite(yellow, HIGH);
    digitalWrite(stepPin2,HIGH);
    delayMicroseconds(500);
    digitalWrite(stepPin2,LOW);
    delayMicroseconds(500);
    Serial.print("Vertical steps = ");
    Serial.println(x);
    if(digitalRead(interruptPin_CCW) == 1){
      reset();
      break;
   }
  }
   digitalWrite(yellow, LOW);
}
```