

Motorized Ikea Skarsta stand-up desk project

Peter Eriksson, 29.4.2016

Project goal:

The goal of this project is to convert a manually operated Ikea desk (Skarsta) into a motor driven desk. The main functionality of the desk is controlling it, up and down, with a press of a button and the possibility of setting an up and down height which the desk raises or lowers to with a press of a button.

Parts used:

1. Arduino Uno R3
2. 220V AC - 9V DC power supply
3. Cordless Screwdriver (1/4")
4. 90-degree screwdriver accessory
5. 16x2 LCD Display (Optional)
6. 2x Relay for Arduino
7. Ultrasonic sensor
8. AC 85~265V to DC 12V 100W 8.5A Switching Power Supply
9. DC 5-30V to DC 1.25-30V Adjustable Step-up/Step-down Constant Voltage Constant Current Module
10. Resistors (These are the ones I used, other values can be used)
 - a. 1K Ω
 - b. 10K Ω
 - c. 22K Ω
 - d. 47K Ω
 - e. 68K Ω
 - f. 100K Ω
11. 10K Ω Potentiometer
12. 5x Buttons
13. Fuse holder and 10A fuse (Optional)
14. On/Off switch (Optional, recommended)
15. 3D Printed mounting plate (Optional)
16. Perfboard 5x7cm (Possibly optional)
17. Wire

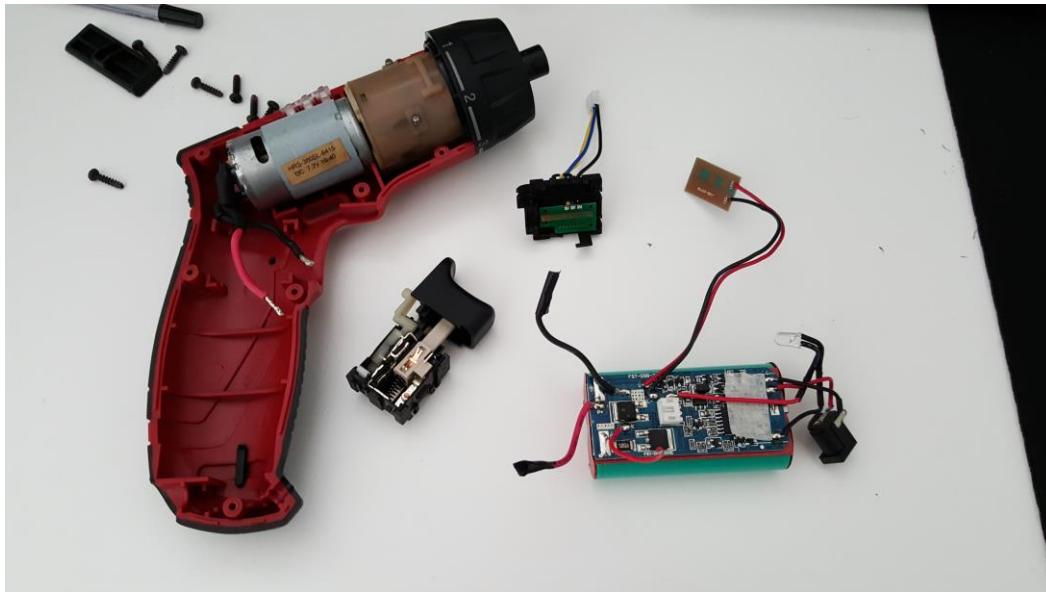
Initial testing:

To start off, we will test that all our main parts work. This is done by connecting the ultrasonic sensor, relay board and the display to the Arduino Uno. Here we will need to use a perfboard of something similar to distribute the 5V power and ground to the different parts. To see that the different parts work, one can hear when the relays switch on and off, as well as see the LED turn on and off. The display will light up and depending on the test code used could show the reading from the ultrasonic sensor, which would confirm that the sensor works.

To test the buttons, because we will be using the analog input pin on the Arduino, the resistors that will be used have to be connected. They can either be soldered in line with the buttons or in my case I used the perfboard. To get the analog value for each button we will use a simple program that displays the value on the LCD. Make note of the values that each button gives, we will need the values in our main program.

Modifications made to some parts:

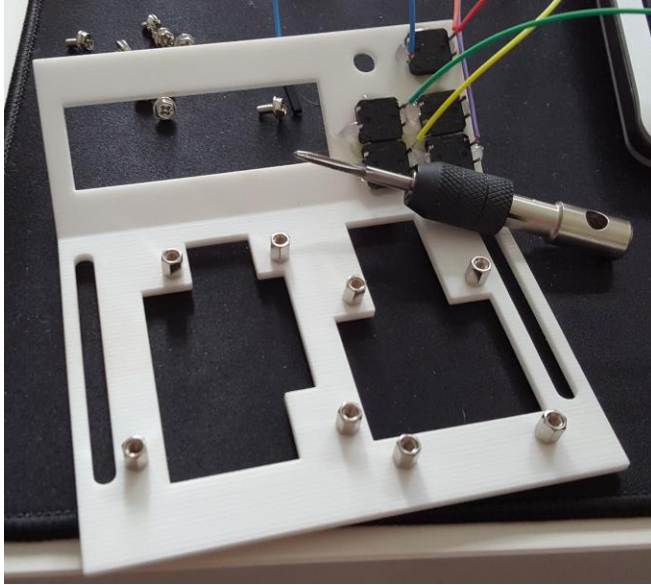
Some of our parts will need to be modified in this project. The cordless powered screwdriver needs to be opened and all but the motor removed from the housing.



Next the power cables have to be extended to reach the relay board, depending on where the relay board is placed. In my case they were extended 60 cm.



In my project I made a 3D printed mounting plate which was later modified with standoff screws to mount the larger parts to the mounting plate. Picture on next page.

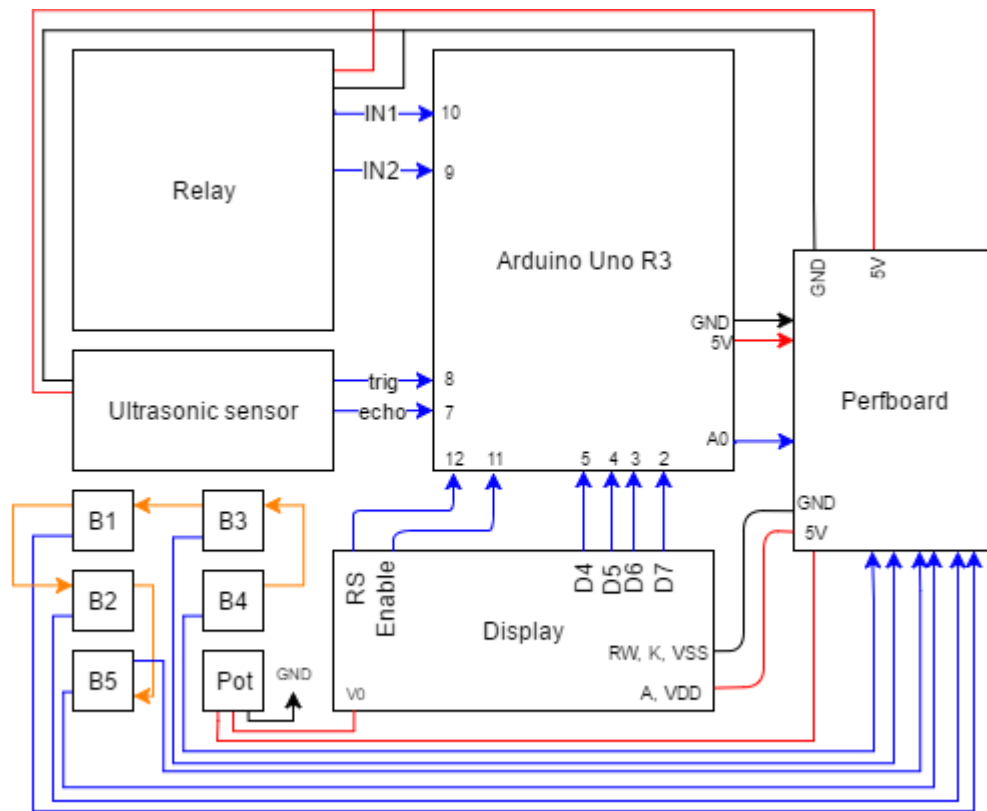


First mark where the mounting standoff screws are supposed to be. Drill holes, in my case I used a 2.5mm drill bit. Next either tap the holes with a m3 tap or try to screw the standoffs in without tapping the holes. I used a tap.

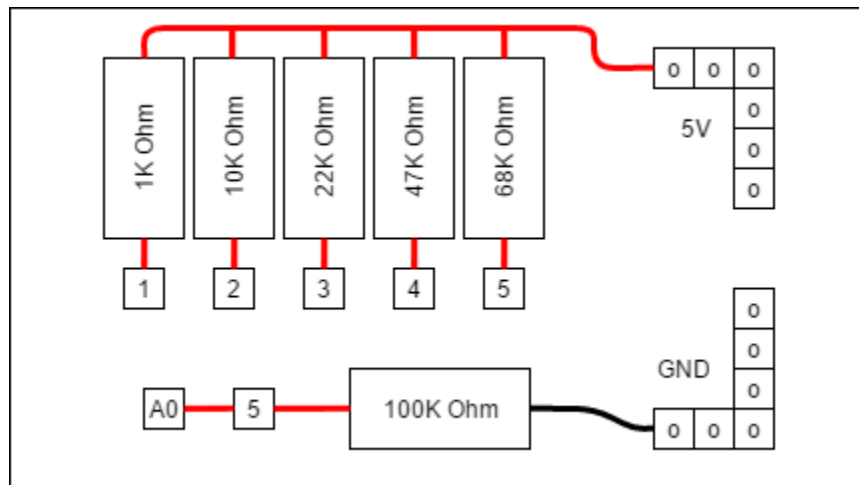
The holes for the buttons might need to be enlarged, depending on the buttons that one uses.

The display is a bit of a tight fit, but because of this tight fit it does not need to be attached with screws to the mounting plate.

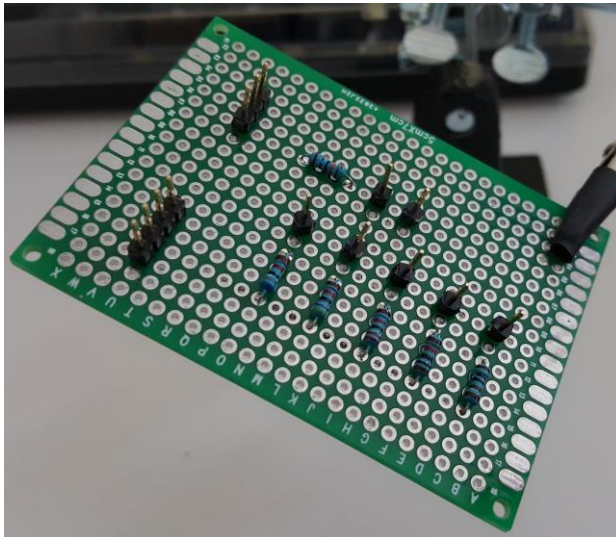
Connecting electronics / simple wiring diagram:



Basically how everything is wired to the Arduino Uno.

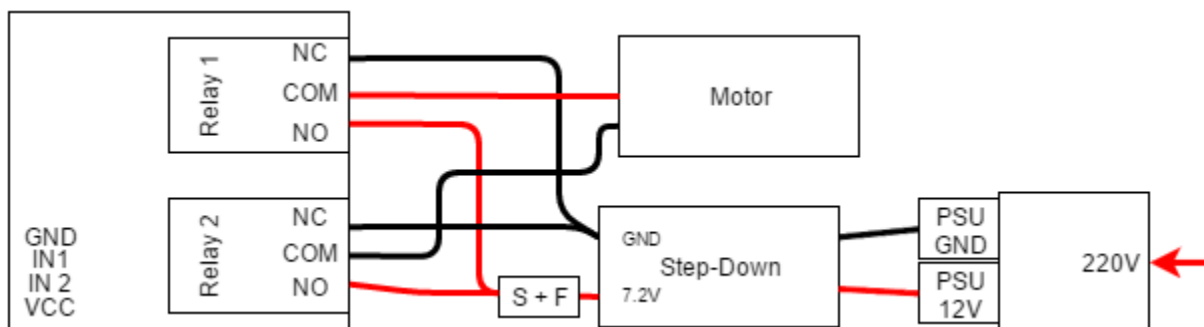


The Perfboard, number 1-5 go to the buttons, 5V and Ground go to parts and the Arduino Uno.



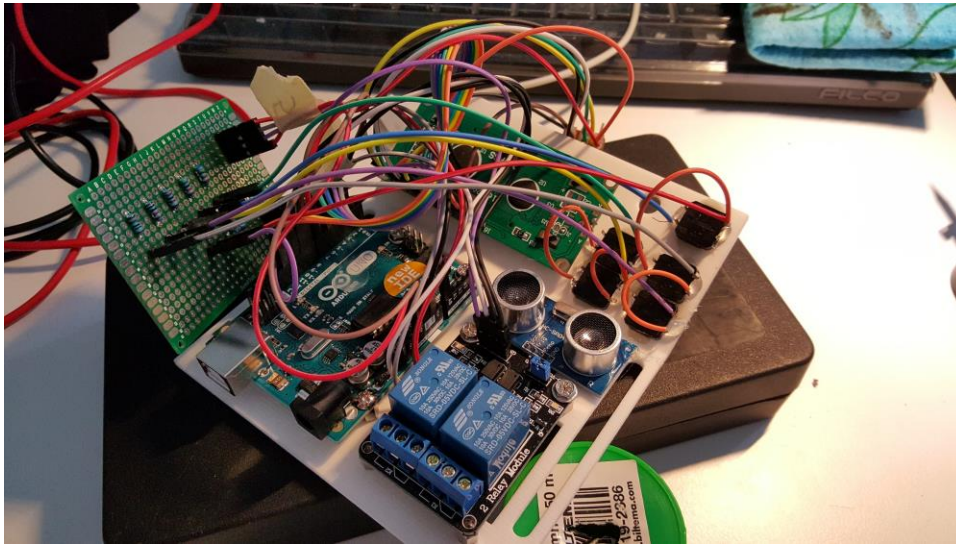
The perfboard with resistors and pins.

The main power for the motor is connected in the following way:



Note: S + F stands for Switch and Fuse, which are connected after each other.

Assembly of parts:



The final assembly of the main parts on the mounting plate. The cables can be cleaned up a bit by routing them underneath the Arduino.

The cordless powered screwdriver was mounted to the underside of the desk with a 90-degree attachment which was needed due to the size of the powered screwdriver. Simple mounting method using metal strips.



The power supply as well as the mounting plate with all the components were also mounted to the underside of the desk, using normal screws.



Code:

The following code is just the start of what can be done with this project, more code will be written to add functionality.

```
//Desk project
//First define all pins
//Then Set everything up
//Then loop
#include <LiquidCrystal.h>
#include <NewPing.h>

#define RELAY1 9           //Up
#define RELAY2 10          //Down
#define TRIG 8             //Ultrasonic sensor trig
#define ECHO 7             //Ultrasonic sensor echo
#define MAX_DIST 200       //Set a MAX distance for the Ultrasonic sensor
#define RS 12              //LCD RS pin
#define E 11               //LCD Enable pin
#define D4 5               //LCD D4 pin
#define D5 4               //LCD D5 pin
#define D6 3               //LCD D6 pin
#define D7 2               //LCD D7 pin
#define AnalogPin A0       //Analog pin that will be used for buttons
```

//Measured values of the different button and resistor combinations, both a LOW and HIGH is needed

```
#define Button1_LOW 605
#define Button1_HIGH 615
#define Button2_LOW 695
#define Button2_HIGH 700
#define Button3_LOW 830
#define Button3_HIGH 845
#define Button4_LOW 925
#define Button4_HIGH 935
#define Button5_LOW 1005
#define Button5_HIGH 1015
```

//Max and Min height of the desk

```
#define MAX_UP 150
#define MIN_DOWN 80
```

```
LiquidCrystal lcd(RS, E, D4, D5, D6, D7);    //Set up the LCD display
NewPing sonar(TRIG, ECHO, MAX_DIST);        //Set up the Ultrasonic sensor
```

```
long distance = 0;           //Height value
long pre_set_up = 100;       //initial preset up height = 150
long pre_set_down = 80;      //initial preset down height = 100
long buttonVal = 0;          //Value of analogRead / Button
```

```
void setup() {
  lcd.begin(16, 2);          //Start the LCD display
  lcd.noAutoscroll();        //Disable auto scroll
  pinMode(RELAY1, OUTPUT);    //Set pin for relay 1 to output
  pinMode(RELAY2, OUTPUT);    //Set pin for relay 2 to output
  digitalWrite(RELAY1, HIGH); //Turn relay 1 off
  digitalWrite(RELAY2, HIGH); //Turn relay 2 off
}
```

```
distance = sonar.ping_cm();    //Get an initial height reading
```

```
String text1 = "UP:" + pre_set_up; //Text to be initially written to the display
String text2 = "DOWN:" + pre_set_down;
String final_text = text1 + text2;
lcd.display();                  //making sure the display is on
lcd.clear();                   //Clear the screen
lcd.print(text1);               //Print the text on the screen
lcd.setCursor(0,1);            //Move cursor to next row
lcd.print(distance);            //Print height
}
```

```
long read_Button() {           //Function used to get the button value
  long temp_button = 0;
  int j;
  for ( j = 0; j < 5; j++) {
    temp_button += analogRead(AnalogPin);
    delay(10);
  }
}
```

```

temp_button = temp_button / 5; //Take five readings and then use the average to determine which button is
pressed
return temp_button;
}

```

```

void manual_up() { //Basic function that turns on the relays so that the table goes up
distance = sonar.ping_cm();
if(distance >= MAX_UP) {
lcd.clear();
lcd.setCursor(0,1);
lcd.print("MAX HIGHT!");
digitalWrite(RELAY1, HIGH); //Turn relay 1 off
digitalWrite(RELAY2, HIGH); //Turn relay 2 off
}
else {
digitalWrite(RELAY2, HIGH); //Turn relay 2 off, making sure
digitalWrite(RELAY1, LOW); //Turn relay 1 on, table goes up
}
}

```

```

void manual_down() { //Basic function that turns on the relays so that the table goes down
distance = sonar.ping_cm();
if(distance <= MIN_DOWN) {
lcd.clear();
lcd.setCursor(0,1);
lcd.print("MINIMUM HIGHT!");
digitalWrite(RELAY1, HIGH); //Turn relay 1 off
digitalWrite(RELAY2, HIGH); //Turn relay 2 off
}
else {
digitalWrite(RELAY1, HIGH); //Turn relay 1 off, just making sure
digitalWrite(RELAY2, LOW); //Turn relay 2 on, table goes down
}
}

```

```

long get_height() { //Function to get an accurate height reading
long temp_echo_time;
long temp_distance;
temp_echo_time = sonar.ping_median(10); //Take 10 readings and return the average
temp_distance = sonar.convert_cm(temp_echo_time); //Convert echo time to cm
return temp_distance;
}

```

```

int going_up(){ //Function to have the desk go up to a set height
long TmpButtonVal = 0;
int returnVal = 0;
long temp_echo_time;
temp_echo_time = sonar.ping_median(5);
distance = sonar.convert_cm(temp_echo_time); //Get the current height in cm
TmpButtonVal = read_Button(); //Check if a button is pressed

if (distance < (pre_set_up + 2) && distance > (pre_set_up - 2)){
digitalWrite(RELAY1, HIGH); //Turn relay 1 off
}

```



```

digitalWrite(RELAY2, HIGH);    //Turn relay 2 off
returnVal = 1;
return returnVal;
}
else if (distance < pre_set_up){ //Under the set height, we go up
digitalWrite(RELAY2, HIGH);    //Turn relay 2 off, making sure
digitalWrite(RELAY1, LOW);     //Turn relay 1 on, table goes up
delay(1000);                   //Let table go up for 3 seconds
lcd.clear();
lcd.setCursor(0, 1);
lcd.print("Height:" + distance);
returnVal = 0;
return returnVal;             //Still not at set height, return 0
}
else if (distance > pre_set_up){ //Over the set height, we go down
digitalWrite(RELAY1, HIGH);    //Turn relay 1 off, just making sure
digitalWrite(RELAY2, LOW);     //Turn relay 2 on, table goes down
delay(1000);                   //Let table go up for 3 seconds
lcd.clear();
lcd.setCursor(0, 1);
lcd.print("Height:" + distance);
returnVal = 0;
return returnVal;             //Still not at set height, return 0
}
else if (distance == pre_set_up){ //When we reach the set height
digitalWrite(RELAY1, HIGH);    //Turn relay 1 off
digitalWrite(RELAY2, HIGH);    //Turn relay 2 off
returnVal = 1;
return returnVal;
}
else if (TmpButtonVal > 600){ //If a button is pressed the loop is stopped
digitalWrite(RELAY1, HIGH);    //Turn relay 1 off
digitalWrite(RELAY2, HIGH);    //Turn relay 2 off
returnVal = 1;
return returnVal;             //Return 1 to get out of loop, due to button press
}
else {
lcd.print("ERROR IN SYSTEM");
digitalWrite(RELAY1, HIGH);    //Turn relay 1 off
digitalWrite(RELAY2, HIGH);    //Turn relay 2 off
returnVal = 1;
return returnVal;             //Return 1 to get out of loop, due to error in system
}
returnVal = 0;
return returnVal;
}

int going_down() { //Function to have the desk go down to a set height
long TmpButtonVal = 0;
int returnVal = 0;
long temp_echo_time;
TmpButtonVal = read_Button(); //Check if a button is pressed
temp_echo_time = sonar.ping_median(5);

```

```

distance = sonar.convert_cm(temp_echo_time);    //Get the current height in cm

if (distance < (pre_set_down + 2) && distance > (pre_set_down - 2)){
    digitalWrite(RELAY1, HIGH);    //Turn relay 1 off
    digitalWrite(RELAY2, HIGH);    //Turn relay 2 off
    returnVal = 1;
    return returnVal;
}
else if (distance < pre_set_down){ //Under the set height, we go up
    digitalWrite(RELAY2, HIGH);    //Turn relay 2 off, making sure
    digitalWrite(RELAY1, LOW);     //Turn relay 1 on, table goes up
    delay(1000);                   //Let table go up for 3 seconds
    lcd.clear();
    lcd.setCursor(0, 1);
    lcd.print("Height:" + distance);
    returnVal = 0;
    return returnVal;              //Still not at set height, return 0
}
else if (distance > pre_set_down){ //Over the set height, we go down
    digitalWrite(RELAY1, HIGH);    //Turn relay 1 off, just making sure
    digitalWrite(RELAY2, LOW);     //Turn relay 2 on, table goes down
    delay(1000);                   //Let table go up for 3 seconds
    lcd.clear();
    lcd.setCursor(0, 1);
    lcd.print("Height:" + distance);
    returnVal = 0;
    return returnVal;              //Still not at set height, return 0
}
else if (distance == pre_set_down){ //Set height is achieved, stop motor and end loop
    digitalWrite(RELAY1, HIGH);    //Turn relay 1 off
    digitalWrite(RELAY2, HIGH);    //Turn relay 2 off
    returnVal = 1;
    return returnVal;
}
else if (TmpButtonVal > 600){      //If a button is pressed the loop is stopped
    digitalWrite(RELAY1, HIGH);    //Turn relay 1 off
    digitalWrite(RELAY2, HIGH);    //Turn relay 2 off
    returnVal = 1;
    return returnVal;              //Return 1 to get out of loop, due to button press
}
else {
    lcd.print("ERROR IN SYSTEM");
    digitalWrite(RELAY1, HIGH);    //Turn relay 1 off
    digitalWrite(RELAY2, HIGH);    //Turn relay 2 off
    returnVal = 1;
    return returnVal;              //Return 1 to get out of loop, due to error in system
}
returnVal = 0;
return returnVal;
}

```

```

void loop() {
  buttonVal = read_Button();      //Check if button is pressed, get the value

  if ( buttonVal > 500) {
    if( buttonVal >= Button1_LOW && buttonVal <= Button1_HIGH ) {
      //Button 1
      manual_up();
      buttonVal = 0;
    }
    else if ( buttonVal >= Button2_LOW && buttonVal <= Button2_HIGH ) {
      //Button 2
      manual_down();
      buttonVal = 0;
    }
    else if ( buttonVal >= Button3_LOW && buttonVal <= Button3_HIGH ) {
      //Button 3
      while (going_up() == 0);
      buttonVal = 0;
    }
    else if ( buttonVal >= Button4_LOW && buttonVal <= Button4_HIGH ) {
      //Button 4
      while (going_down() == 0);
      buttonVal = 0;
    }
    else if ( buttonVal >= Button5_LOW && buttonVal <= Button5_HIGH ) {
      //Button 5
      lcd.setCursor(0,1);
      lcd.print("Choose up/down");
      delay(1000);
      buttonVal = read_Button();      //Read button to get which height to set
      if ( buttonVal >= Button3_LOW && buttonVal <= Button3_HIGH ) {
        lcd.clear();
        lcd.setCursor(0,1);
        lcd.print("Up - set");
        pre_set_up = get_height();
      }
      else if ( buttonVal >= Button4_LOW && buttonVal <= Button4_HIGH ) {
        lcd.clear();
        lcd.setCursor(0,1);
        lcd.print("Down - set");
        pre_set_down = get_height();
      }
      delay(1000);
      buttonVal = 0;
    }
  }
  else {
    digitalWrite(RELAY1, HIGH);      //Turn relay 1 off
    digitalWrite(RELAY2, HIGH);      //Turn relay 2 off
  }
}

```