HOME AUTOMATION DEVICE - IR REMOTE CONTROLLED USING ARDUINO AND VS1838 IR RECEIVER

Final Project in CPE 305
FEEDBACK AND CONTROL SYSTEMS

Submitted by:

CATACUTAN, Zian Jolo Manalo

Submitted to:

ENGR. EDWARD RAYMUNDO
Instructor

This page wad intentionally left blank.

Concept

Home Automation

This concept focusses on making an old home into a smart home, by making most of the things automated or controlled by a device. When things are automated, they are managed by a programmable device that has a sensor, that reads external conditions and adjust connected devices based on readings.

On this project, I use the Infrared Remote-Control devices as a controlling device which is cheap and can be found in the local market. The unique codes transmitted by the IR Remote will enter the VS1838 IR receiver and will be decoded. Then, analog voltages will enter the Arduino – a programable device responding in programmed circumstances.

Layout

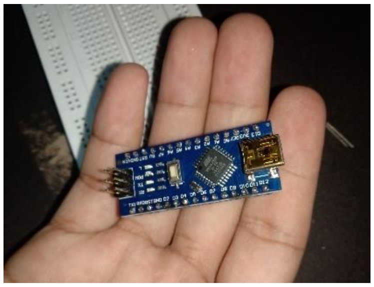
From the IR Remote Control, button codes will be decoded at first. Then a program will instruct the Arduino board based on the codes. The VS1838 will send the receive code frequency to the Arduino Board and then the Arduino board will respond by sending logic level voltages 0v-LOW 5v-HIGH to a Relay that will act as mechanical switch to close the path for AC voltage that will power up connected devices

Planning

My objectives for the final looks and feel of the device must be small, light and portable. So, I must choose all the components based on requirements.

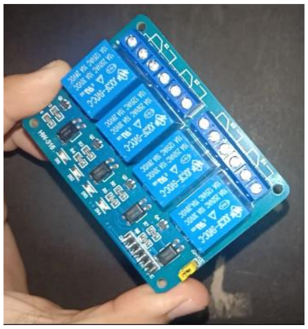
Materials

Arduino NANO



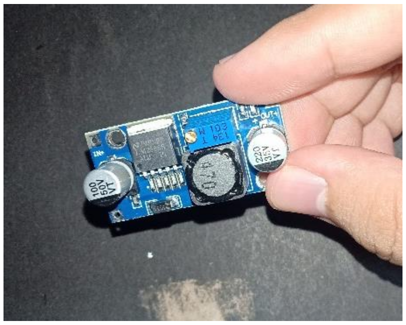
VS1838 IR Receiver

4 Channel (CH) Relay



IR Remote Control

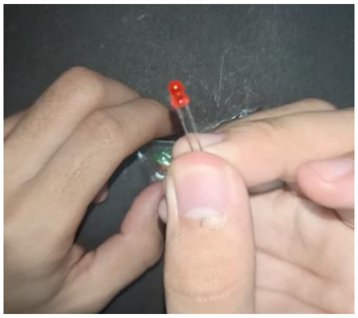
LM2596S DC-DC Buck Converter



12v Switching Power Supply

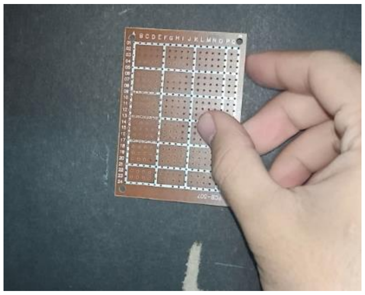
NPN 8050 Transistor

3mm LED Red, Green



330 Ohm Resistors

Perf Board



Double Pole Single Throw (DPST) Switch

Male Pin Headers (optional)

Fuse Holder

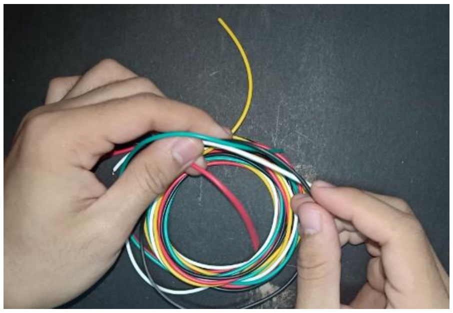
10 Amp rated Glass Fuse

AC Plug

AC Ports

Zip Ties (optional)

Thin Wires



Thick Wires 250v rated

Carton Box

Soldering Leads

Glue Sticks

Masking Tape
& Electrical Tape

Shrinkable Tubes

Tools

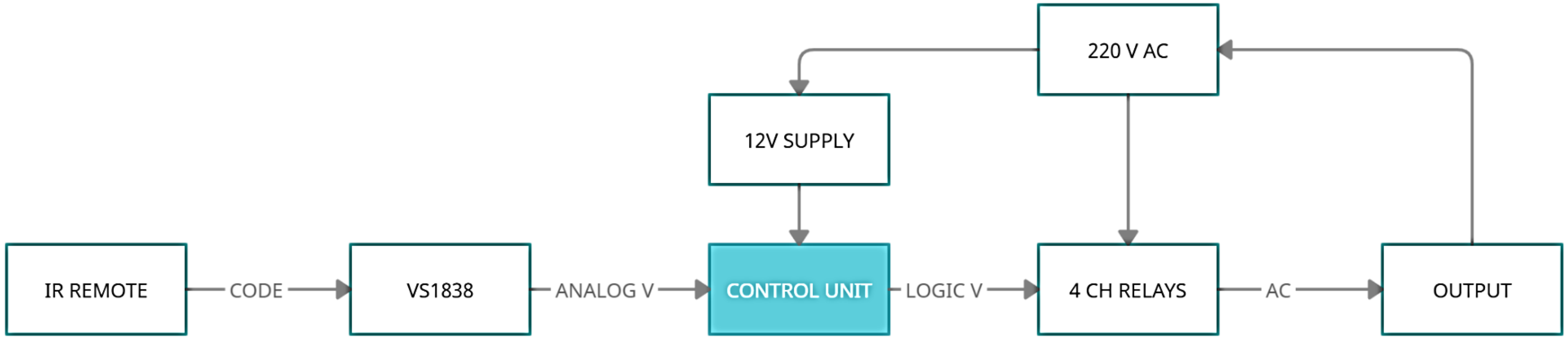
Soldering Iron
Soldering Helping Hands
Multimeter
Glue Gun
Screw Driver
Wire Stripper
Cutter
Ruler
Lab Bench power Supply (optional)
12v Drill Bit (optional)

Total Project Cost: 1,500 Php

Technical Details

Prototyping

Before jumping to soldering I first build the entire control unit to a single breadboard. I check the voltage requirements and current consumption on runtime and standby mode.

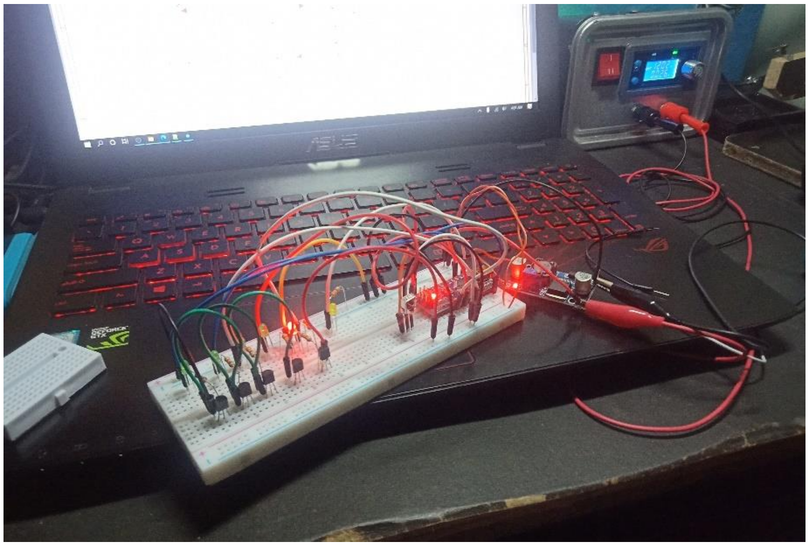


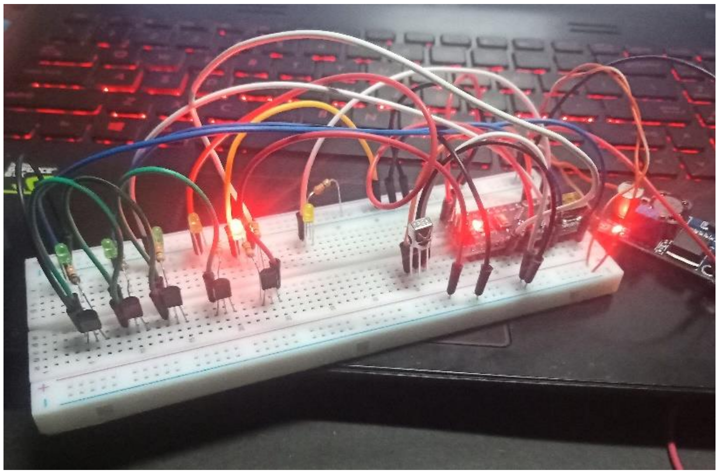
Here is the block diagram on how the whole system was connected.

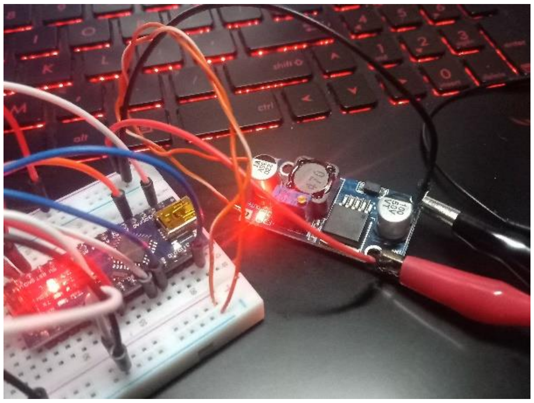
Control Unit

I first draw on the paper the control unit schematic and then build it on a breadboard.

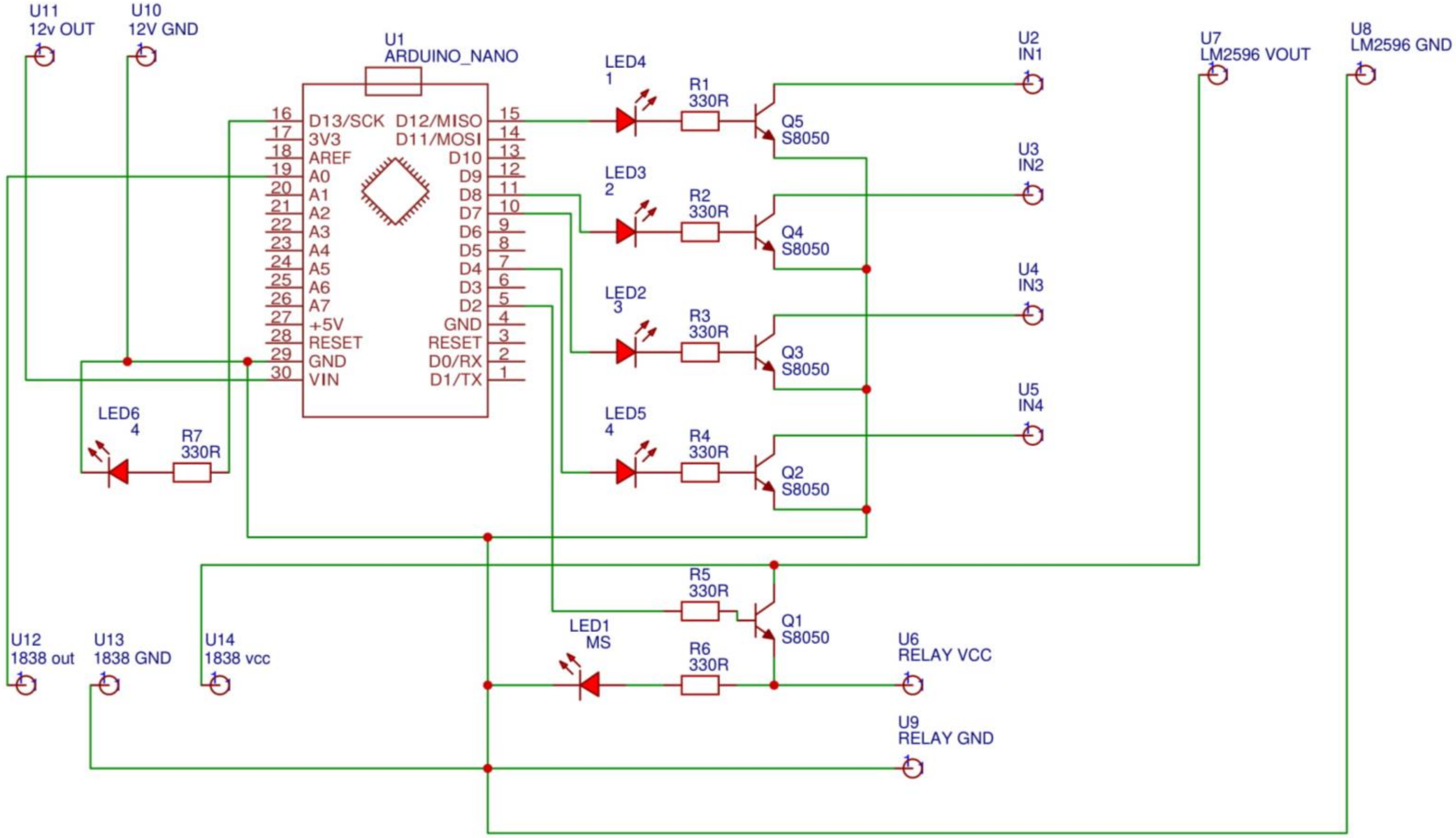
Breadboard prototype



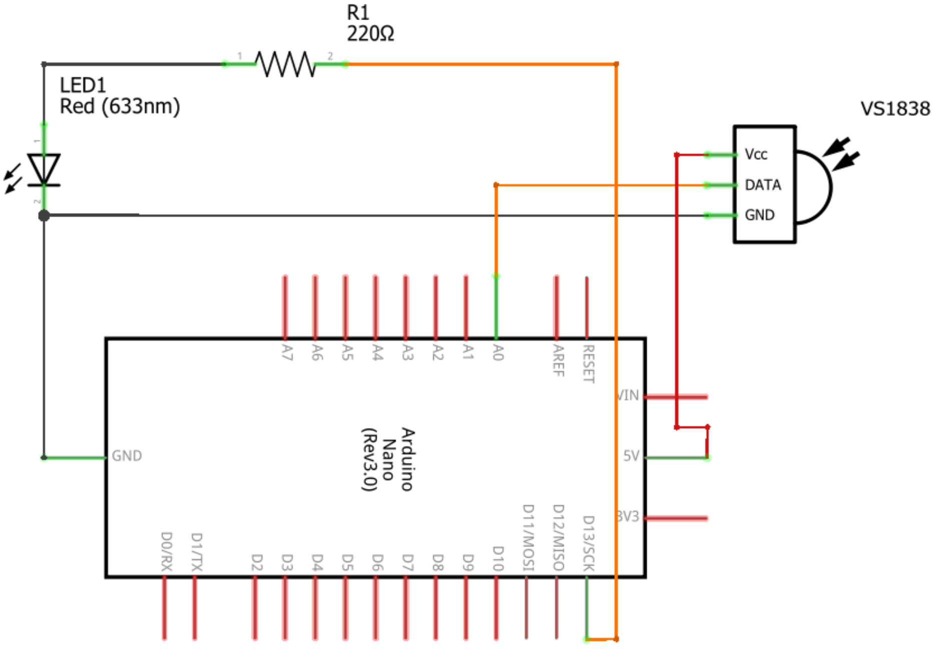




After checking the Control Unit prototype, I then draw the schematic using EASY EDA software.



Control Unit Schematic Diagram



Schematic Decoding IR Remote Button Codes

Schematic for Decoding IR Remote Button Codes

Arduino Nano Code to Decode IR Remote Button Codes

```
#include <IRremote.h>

#define IR_RECEIVE_PIN A0

void setup() {
  Serial.begin(115200);
  IrReceiver.begin(IR_RECEIVE_PIN, ENABLE_LED_FEEDBACK);

}

void loop() {
  // put your main code here, to run repeatedly:
  if(IrReceiver.decode()){
    Serial.println(IrReceiver.decodedIRData.decodedRawData, HEX);
    IrReceiver.resume();
  }
}
```

Arduino Nano Code for the Whole System

```
#include <IRremote.h>
/* Button Codes
  CH- 0xBA45FF00
  CH 0xB946FF00
  CH+ 0xB847FF00
  PREV 0xBB44FF00
  NEXT 0xBF40FF00
  PLAY/PAUSE 0xBC43FF00
  VOL- 0xF807FF00
  VOL+ EA15FF00
  EQ 0xF609FF00

  0 0xE916FF00
  1 0xF30CFF00
  2 0xE718FF00
  3 0xA15EFF00
  4 0xF708FF00
  5 0xE31CFF00
  6 0xA55AFF00
  7 0xBD42FF00
  8 0xAD52FF00
  9 0xB54AFF00
*/
//codes
#define main_switch 0xE916FF00
#define channel_1 0xF30CFF00
#define channel_2 0xE718FF00
#define channel_3 0xA15EFF00
#define channel_4 0xF708FF00
#define turn_off_all 0xE31CFF00

#define IR_RECEIVE_PIN A0

//signal pin
const int main_switch_pin = 2;
const int channel1_pin = 12;
const int channel2_pin = 8;
const int channel3_pin = 7;
const int channel4_pin = 4;

int on_channels = 0;

int pin_data[] = {0, 0, 0, 0, 0, 0};
int pins[] = {main_switch_pin, channel1_pin, channel2_pin, channel3_pin,
channel4_pin};

void setup() {
  //Serial.begin(9600);
  IrReceiver.begin(IR_RECEIVE_PIN, ENABLE_LED_FEEDBACK);
  for (int i = 0; i < 5; i++) {
    pinMode(pins[i], OUTPUT);
  }
  digitalWrite(pins[0], updatePinData(0)); //start the main switch HIGH
}
```

```
void modify_all_channels(int mode) {
  //reset all channels but stay the power
  if (pin_data[0] != 0) {
    //is ms is on
    if (mode == 0) {
      pin_data[5] = 1;
    } else {
      pin_data[5] = 0;
    }
    for (int i = 1; i < 5; i++) {
      digitalWrite(pins[i], mode);
      pin_data[i] = mode;
    }
  }
}

//will update toggle status of each output pin
int updatePinData(int index) {
  if (index != 0 && pin_data[0] == 0) {
    //if other pins is requesting but the main switch is off do not turn on
    return 0;
  }
  if (pin_data[index] == 0) {
    pin_data[index] = 1;
    return 1;
  }
  pin_data[index] = 0;
  return 0;
}

//cut the power to led indicators if main switch is off
//but do not reset pin status
boolean cutPower() {
  int main_status = pin_data[0];
  if (!main_status) {
    //LOW turned off
    for (int i = 1; i < 5; i++) {
      digitalWrite(pins[i], main_status);
    }
    return true;
  }
  //  //bring back all pin status
  for (int i = 1; i < 5; i++) {
    digitalWrite(pins[i], pin_data[i]);
  }
  return false;
}

void loop() {
  if (IrReceiver.decode()) {
    switch (IrReceiver.decodedIRData.decodedRawData) {

      case main_switch: //main switch
        //Serial.println(IrReceiver.decodedIRData.decodedRawData);
        digitalWrite(pins[0], updatePinData(0));
        cutPower();
        break;
      case channel_1: //channel 1
        //Serial.println(IrReceiver.decodedIRData.decodedRawData);
        digitalWrite(channel1_pin, updatePinData(1));
        break;
      case channel_2: //channel 2
        //Serial.println(IrReceiver.decodedIRData.decodedRawData);
        digitalWrite(pins[2], updatePinData(2));
        break;
      case channel_3: //channel 3
        //Serial.println(IrReceiver.decodedIRData.decodedRawData);
        digitalWrite(pins[3], updatePinData(3));
        break;
      case channel_4: //channel 4
        //Serial.println(IrReceiver.decodedIRData.decodedRawData);
        digitalWrite(pins[4], updatePinData(4));
        break;
      case turn_off_all:
        //Serial.println(IrReceiver.decodedIRData.decodedRawData);
        on_channels = 0;
        for (int i = 1; i < 5; i++) {
          if (pin_data[i] == 1) {
            on_channels++;
          }
        }
        if (on_channels > 0) {
          //if one is detected turn off all
          modify_all_channels(0); //off all
        } else {
          modify_all_channels(1); //on all
        }
        break;
      default:
        break;
    }
    IrReceiver.resume();
  }
}
```

IR Remote Control Button Functions

0 – Turn ON/OFF Device.

1 – Turn ON/OFF Channel 1 AC PORT.

2 – Turn ON/OFF Channel 2 AC PORT.

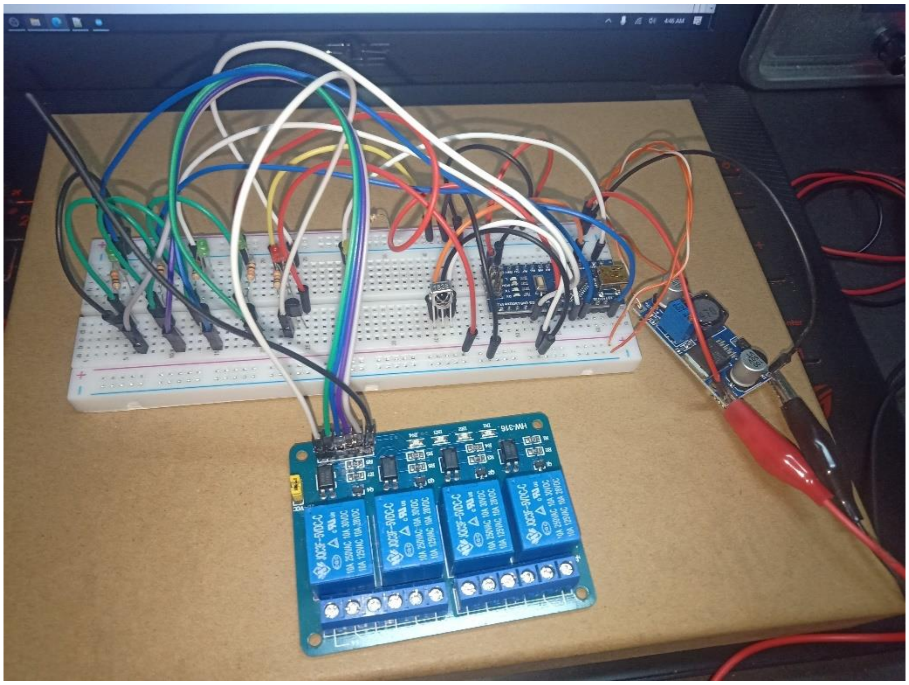
3 – Turn ON/OFF Channel 3 AC PORT.

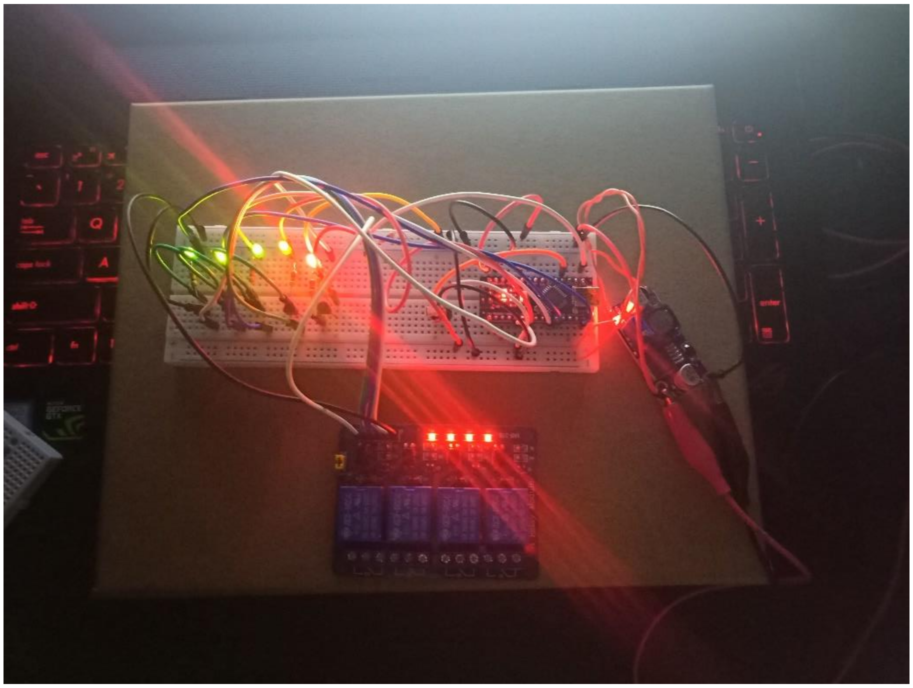
4 – Turn ON/OFF Channel 4 AC PORT.

5 – Inverse of All – if there is an active relay TURN OFF ALL, else TURN ON ALL.

The Whole System Prototype

After testing the control unit circuit, I then connect everything to test the whole system. Using a Lab bench power supply, I temporarily connected a 12v source to be the voltage source for the whole system. This 12v source will be replace later on with the switching power supply (see the block diagram). Good thing everything works fine!





Findings

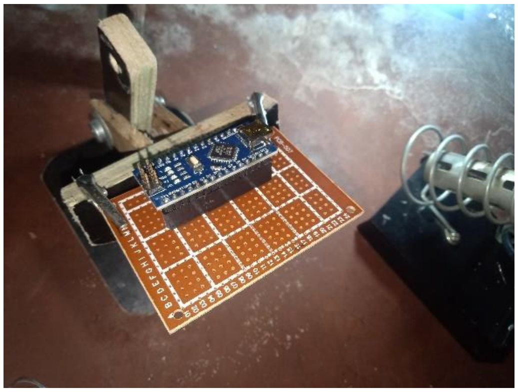
I found out that in standby mode (no relays are on), the whole system draws, 21 milliamps, which is a good sign since it is low.

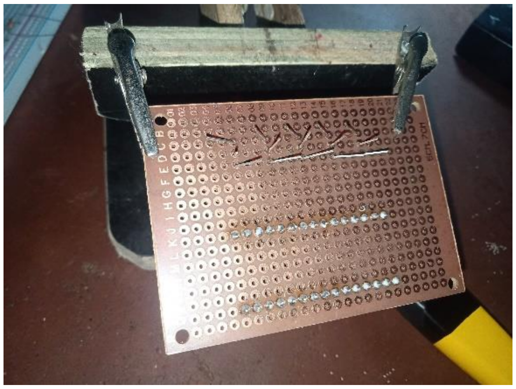
I found out that in operating mode (all relays are on), the whole system draws, 153 milliamps, which is a good sign since it is not too high.

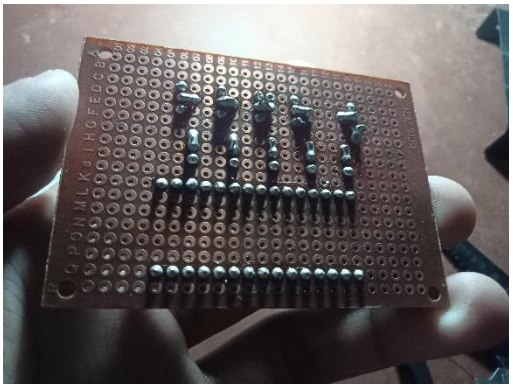
This implies that my 12v 2A switching power supply can power up the whole system without any problems.

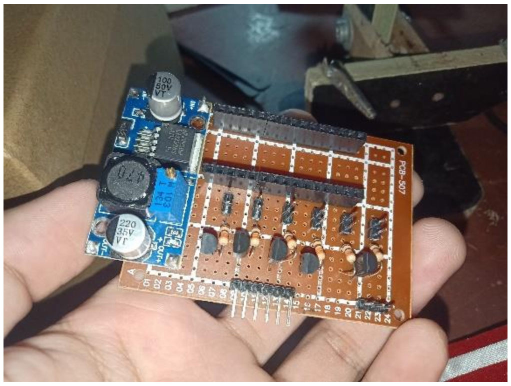
Soldering

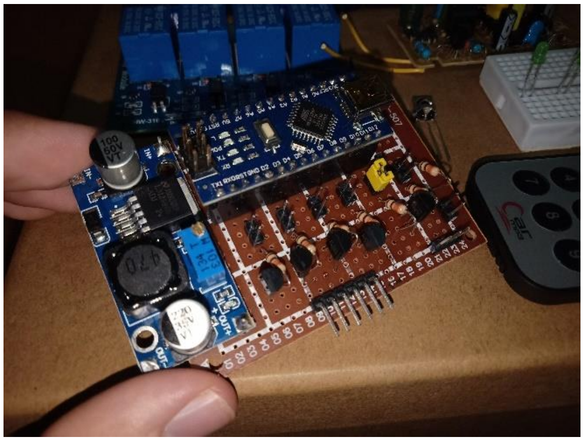
Using the perf board, I soldered the control unit part as close as possible. The reason is efficiency. The closer the components the lesser travel for electricity, the efficient the system will be.

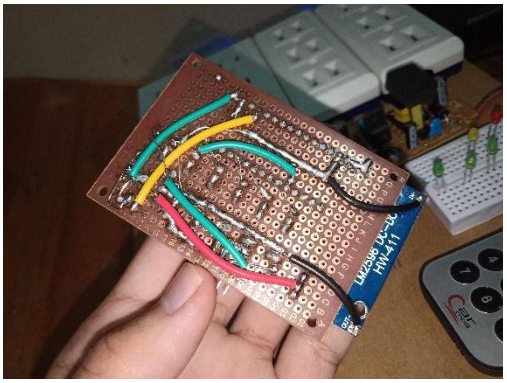




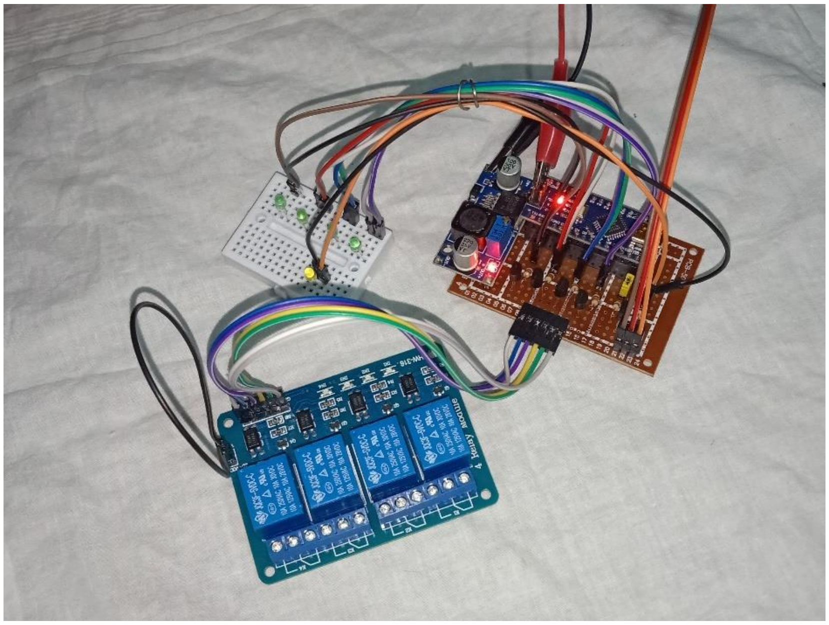




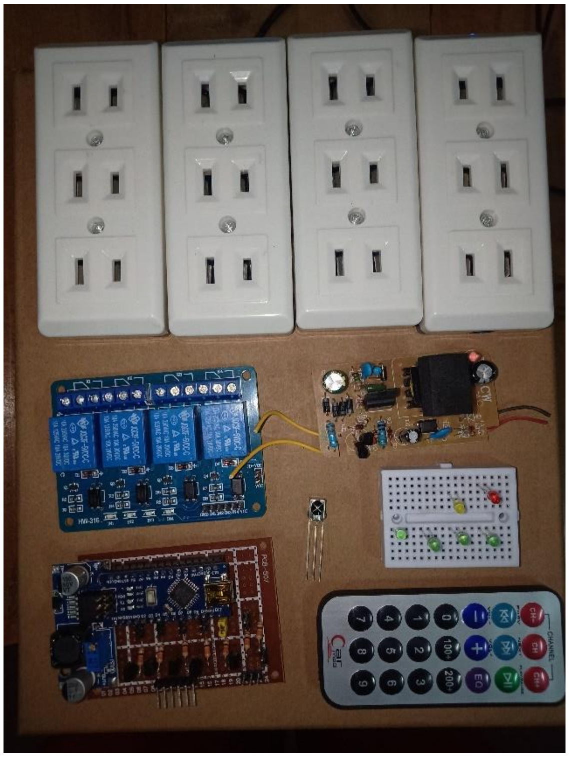




After soldering the control unit, I tested it by temporarily connecting the relay and 12v supply. And everything works fine!



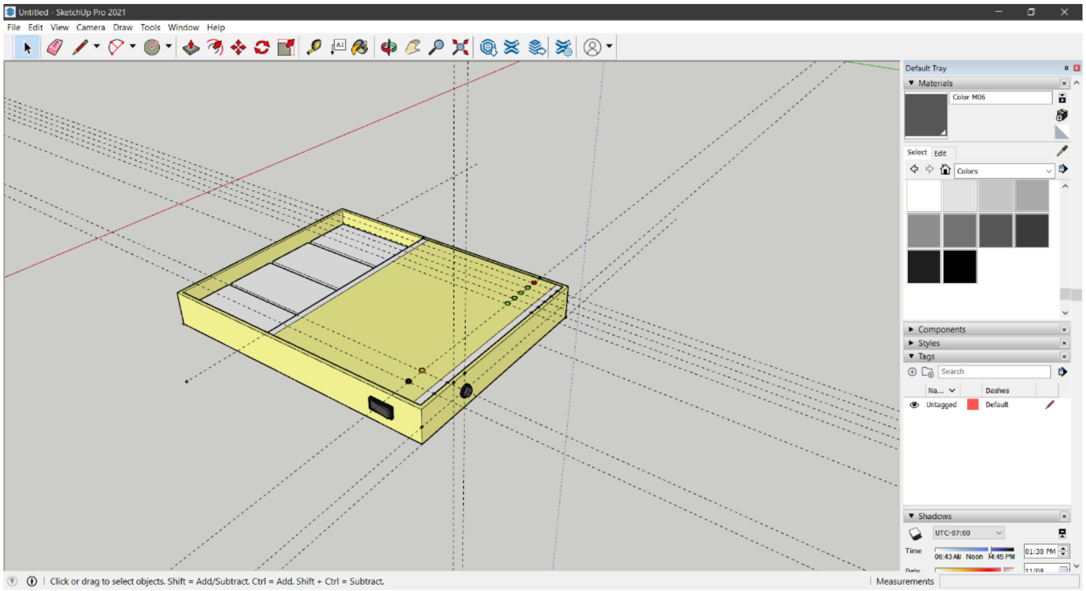
At this point, I completed almost everything, and its time to place it on the casing.



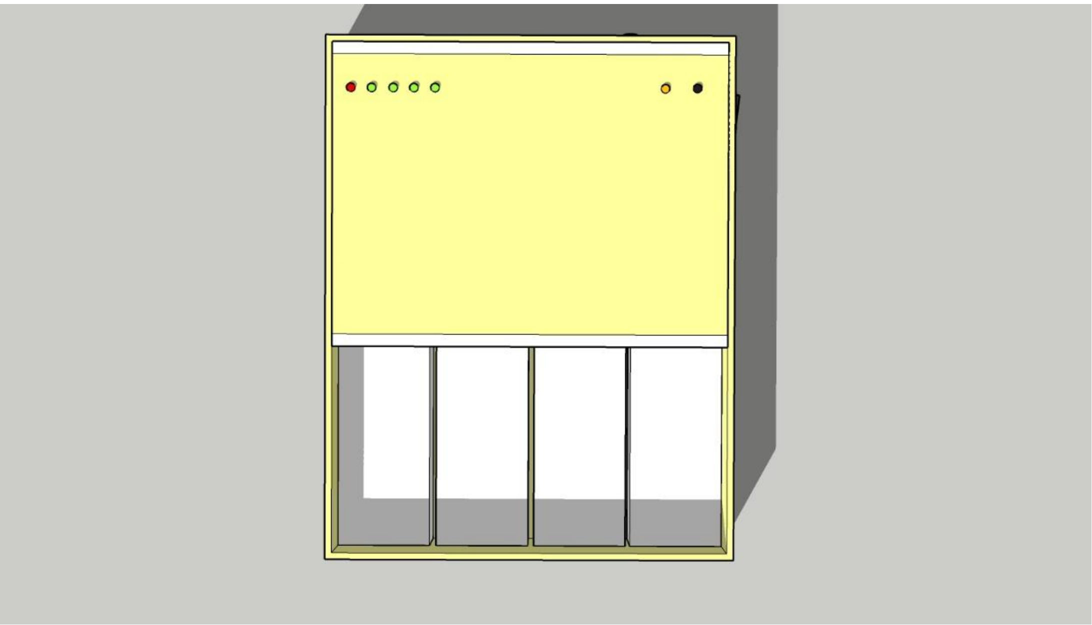
Combining Everything

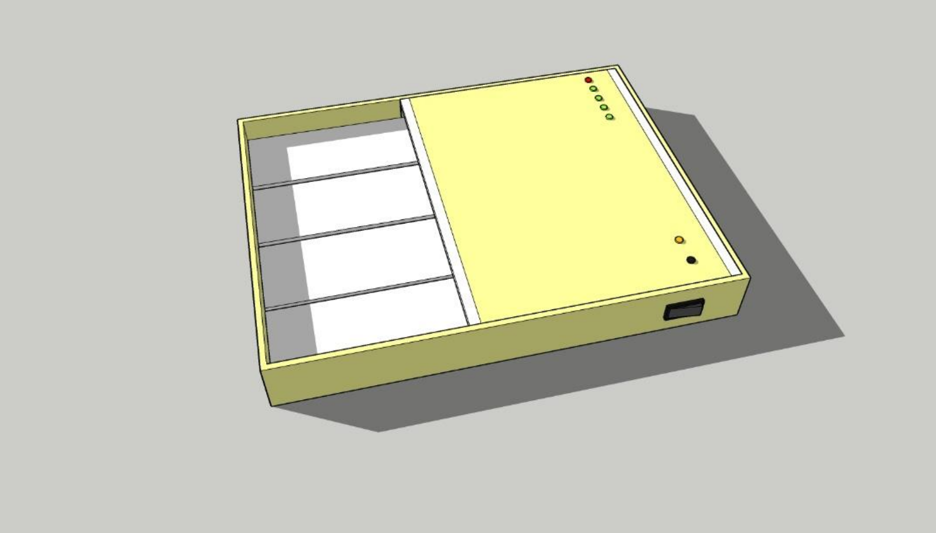
Frame Design

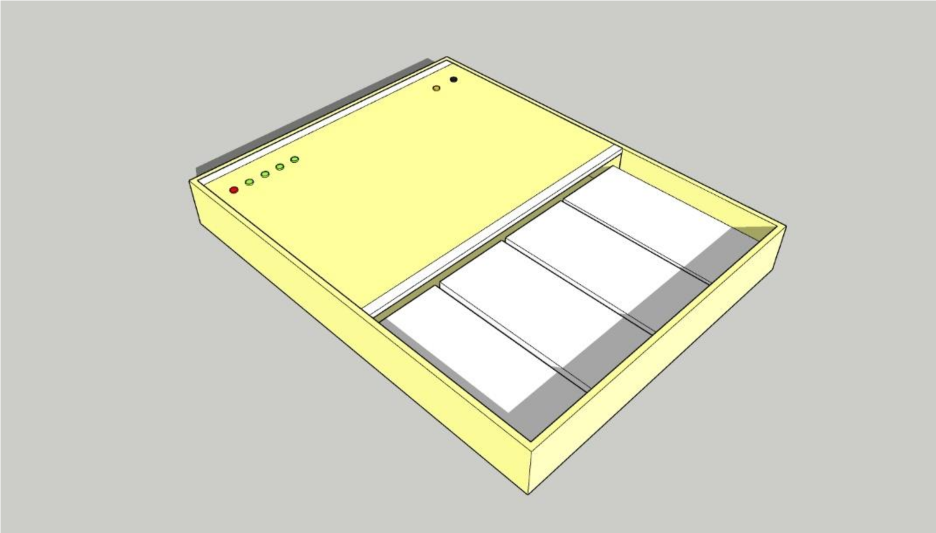
Before building the frame I first visualize it using a 3d modelling software, specifically Sketchup 2021.

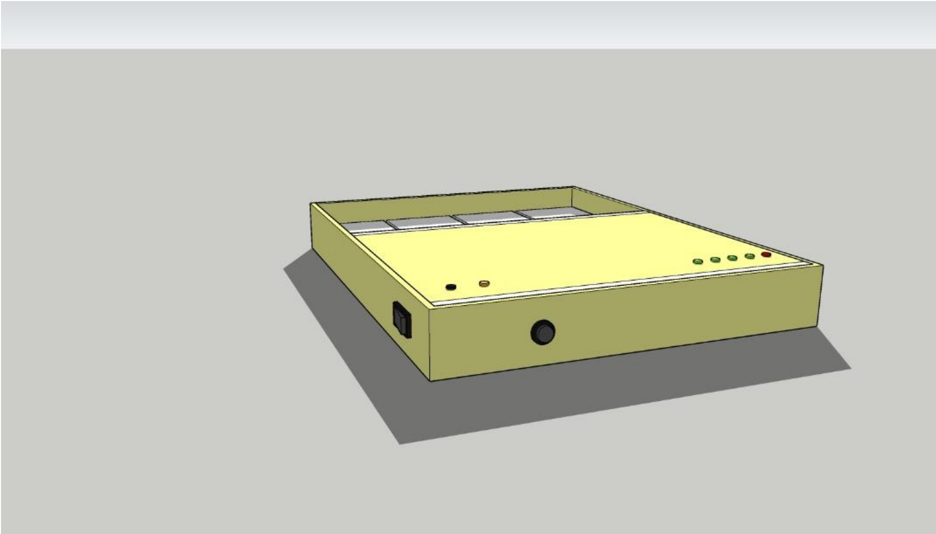


Click or drag to select objects. Shift = Add/Subtract. Ctrl = Add. Shift + Ctrl = Subtract.

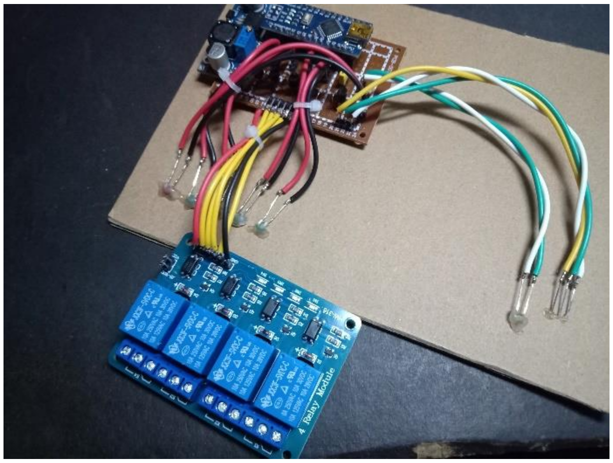


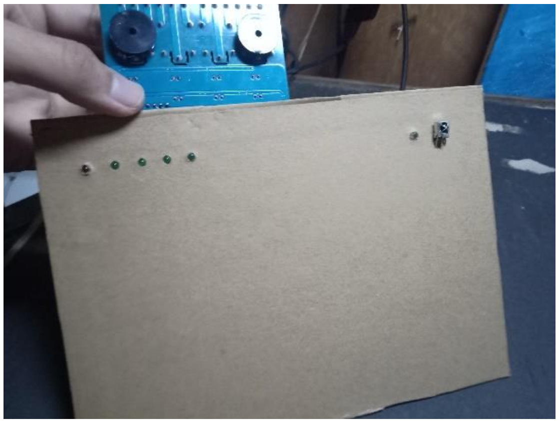


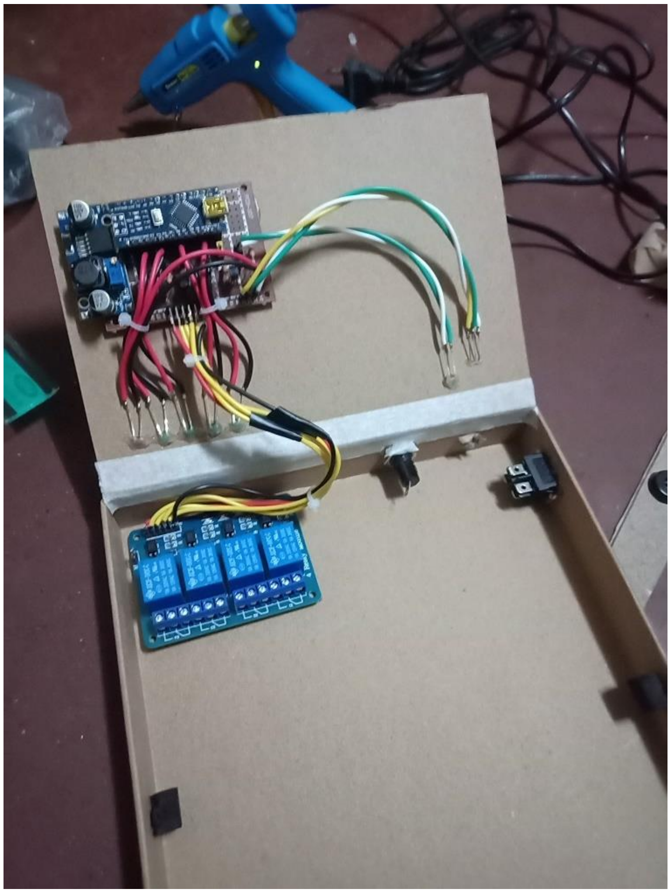


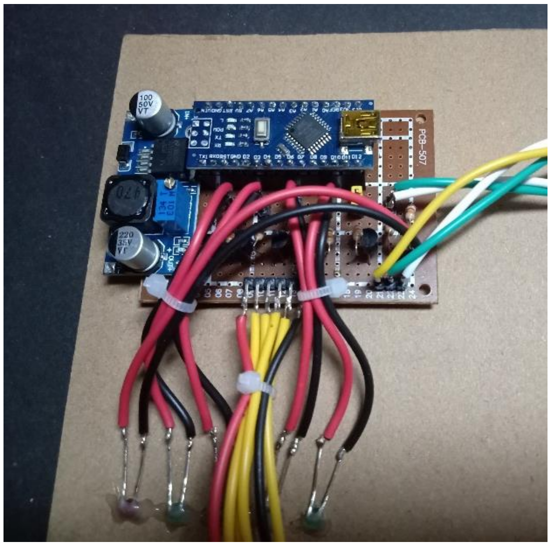


Using the carton box, I cut everything and form the frame based on the 3d model. Then I place first the control unit circuit board and the relay and connecting all the led indicators and the VS1838, with thin wires. I also placed the DPST Switch and the Fuse Holder.



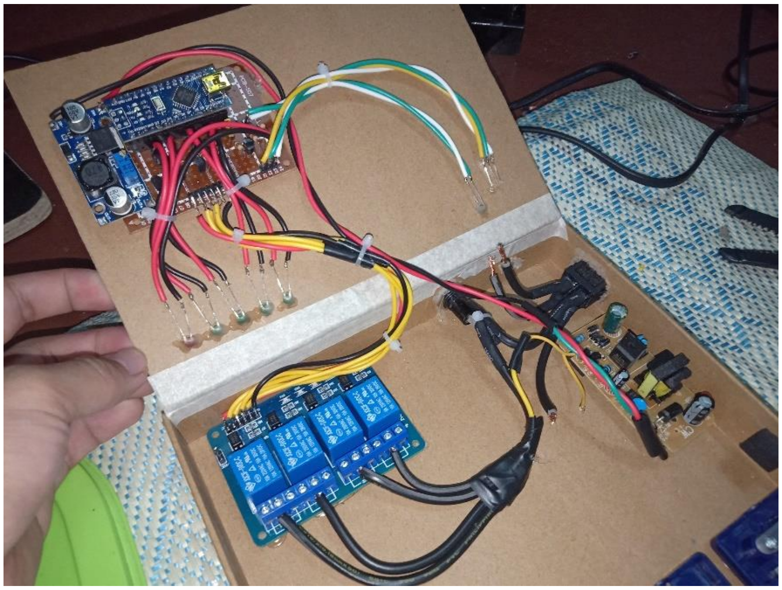


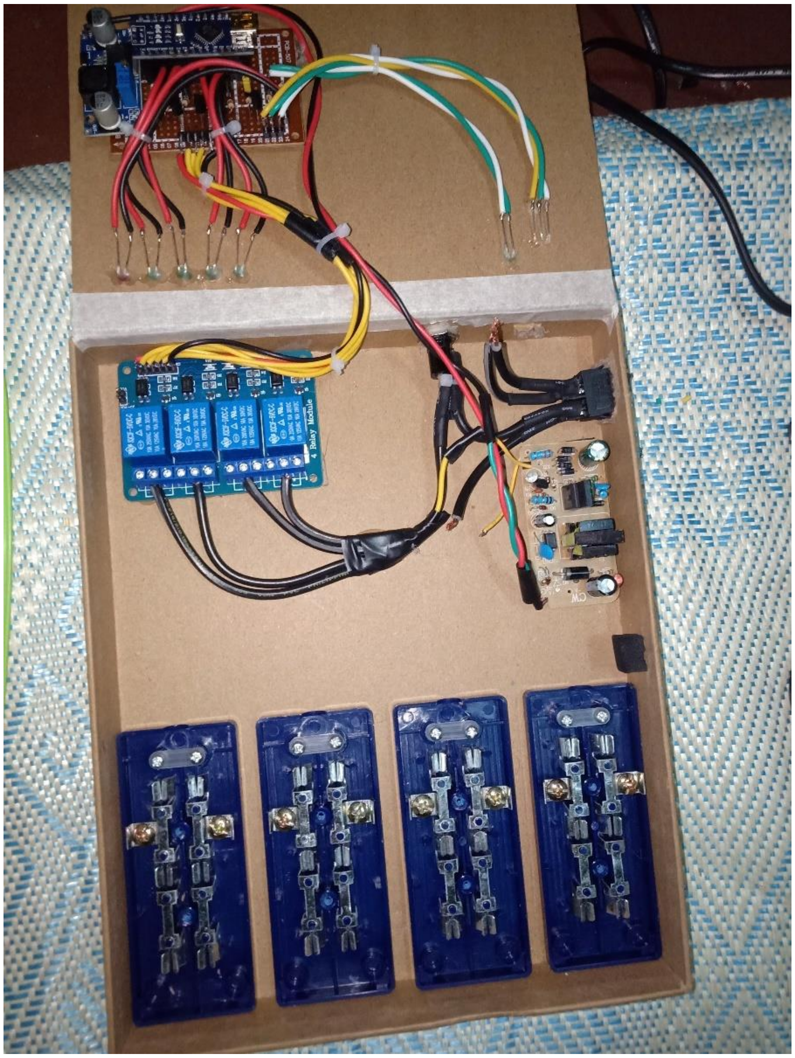


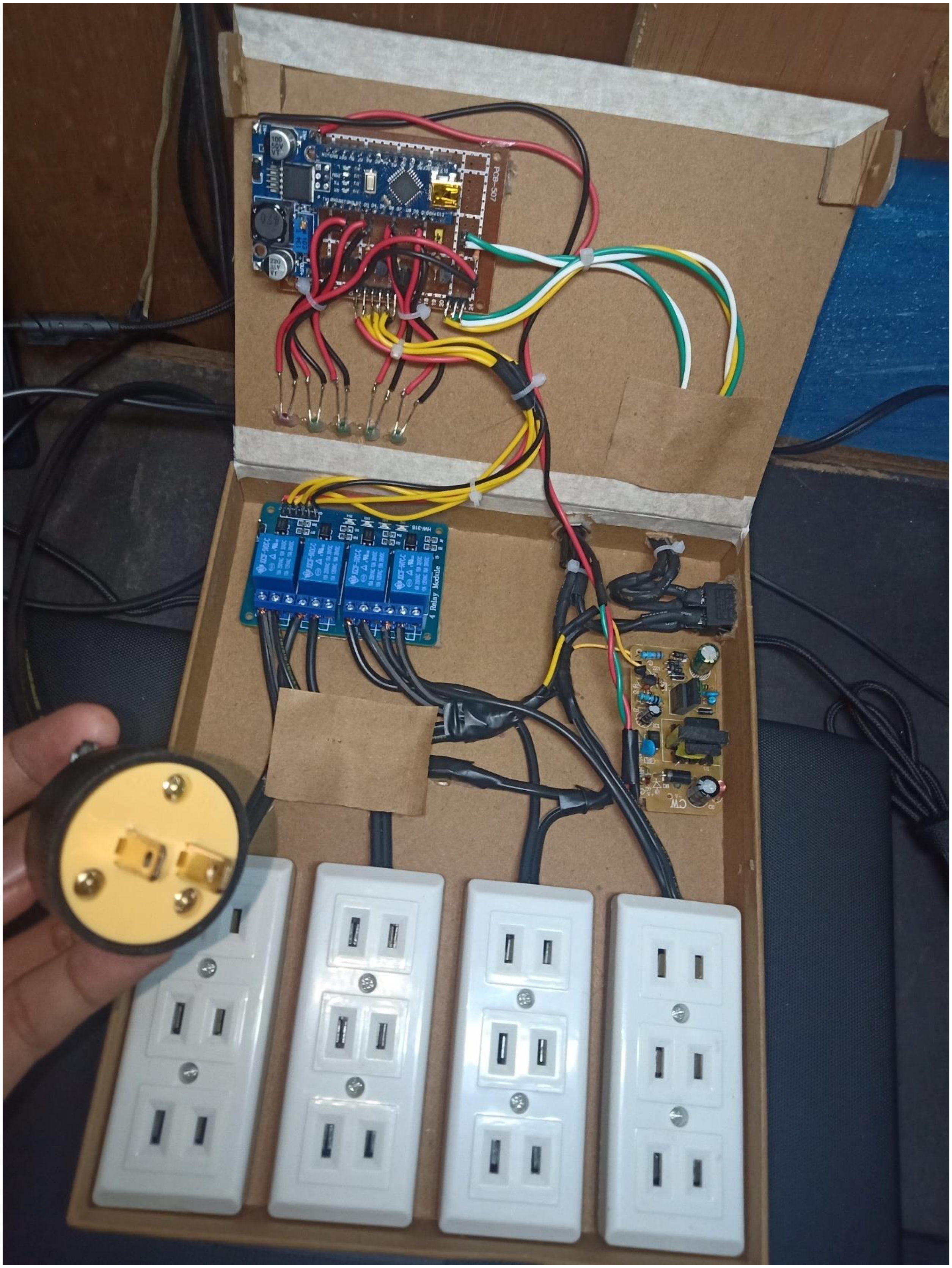


I then test the circuit before wiring the AC Voltage source and 12v supply. I temporarily hook up 12v power source from my lab bench power supply. Good thing it works fine.

At this point the only missing components are the AC Ports, the 12v switching power supply and the AC Voltage lines. This is the most dangerous part of this project because I am using AC live voltage which can cause burns and the worst case is death. The next page will show the photos of internal AC connections. I use the thick wires for connections.

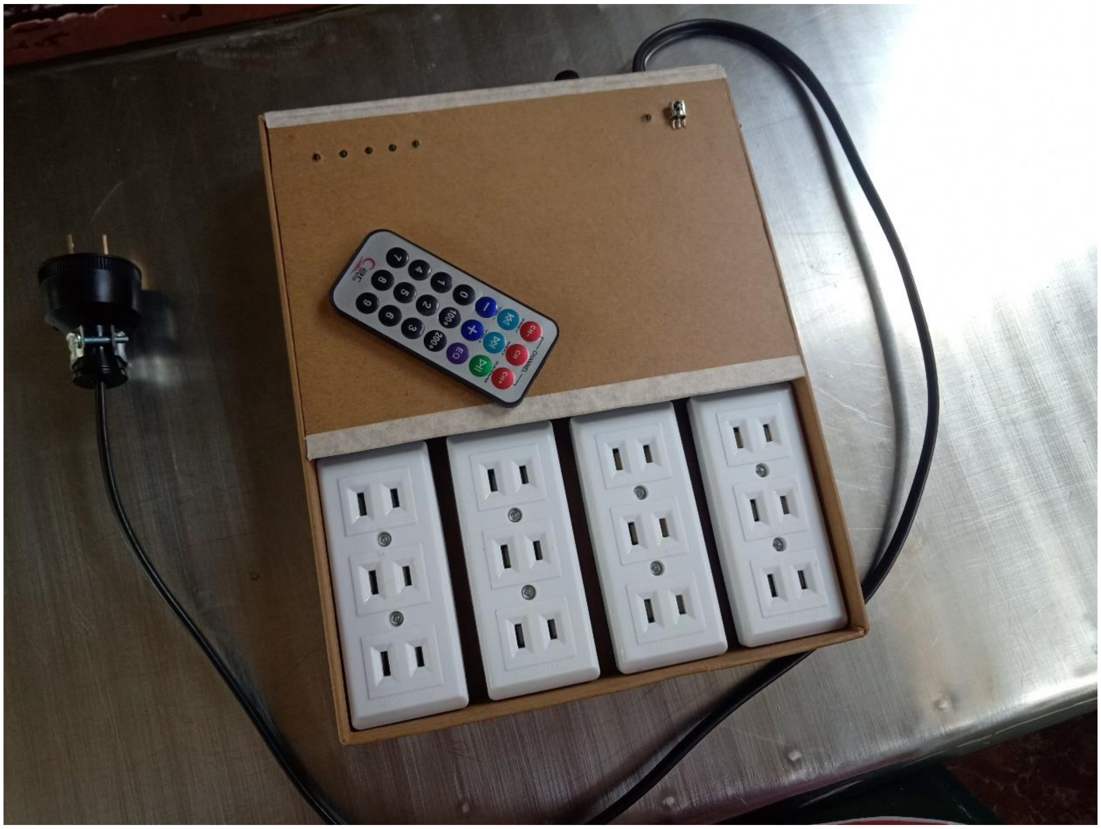


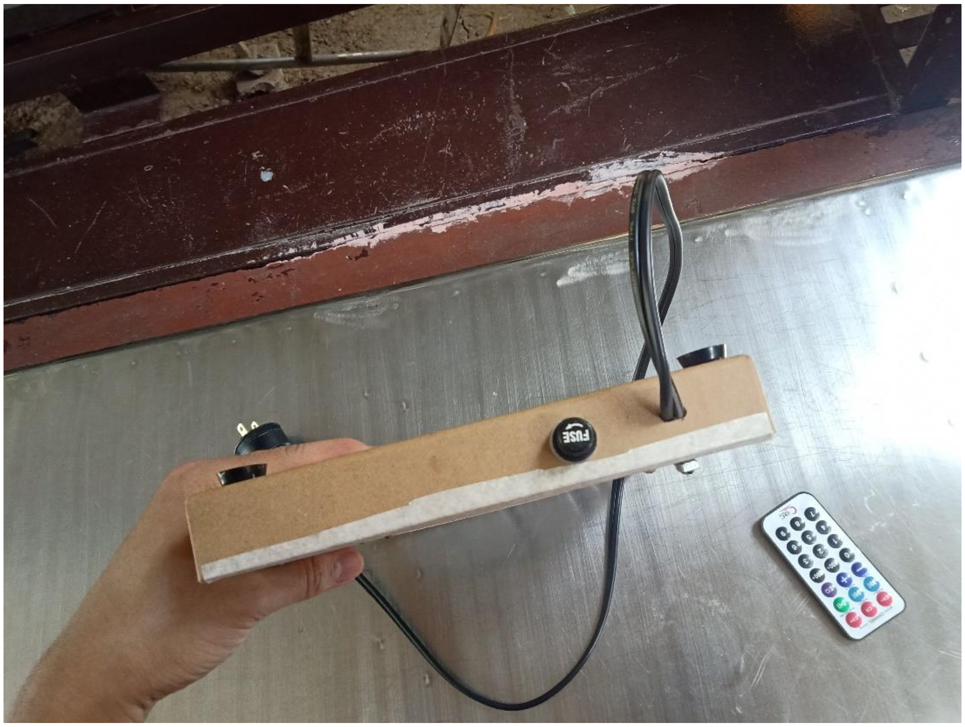

4 Relay Module

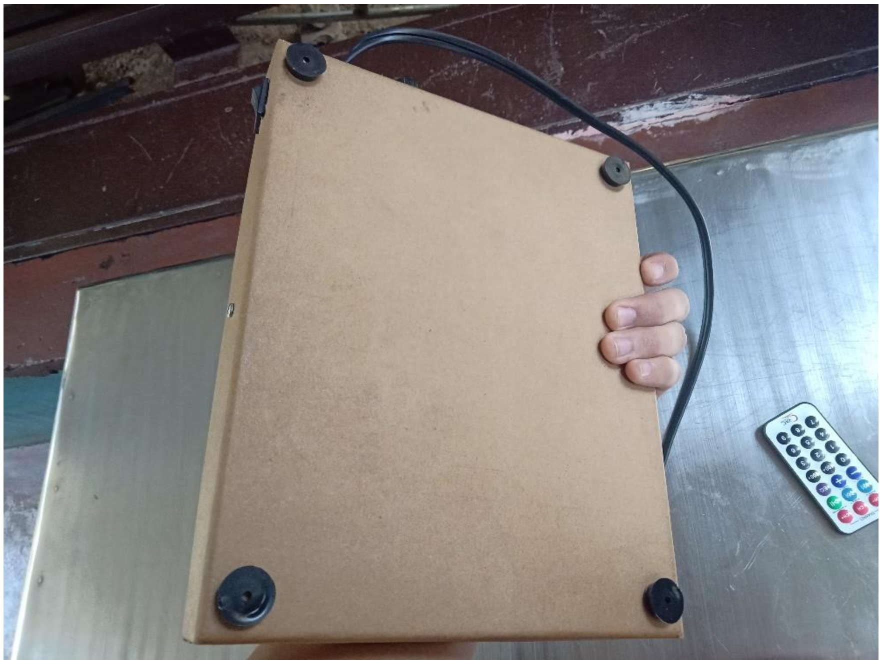

PCB-507
HW-316
4 Relay Module
CW

Final Output

Here is the final result, after a total of 2 days working.




FUSE



But the question is, is it working? Please see the video to see how it works! That's all thank you.