ME102B -- Fall 2020
Final Project

The Automatic Nail Filer

By Wonsik Chung

Description of the product:

I used to have a very terrible nail cutting skill, so whenever I cut my nails, the edge of the nails was always very sharp. This always bothers me, so I often grinded the sharp edge of the nails by using a nail filer. Also, my nails used to grow really fast, so I have to cut them once in a week. The problem of the nail filer is that you must have a good technique to have a nice smooth surface cut, and also, it takes time. The primary motivation of this project arose due to my desire of having a convenient nail filer machine, which will help me to arrange my nails in a short time without having a nice cutting technique. While I was learning the fundamental knowledge of mechatronics design, I have learned various theories related to DC motors and its speed control. I do not believe that this project perfectly reflects my understanding of this class and the various materials that are covered in this class, but yet, it is a great opportunity for me to actually conduct some research and do some hands-on experience based on the knowledge that I gained from this class.

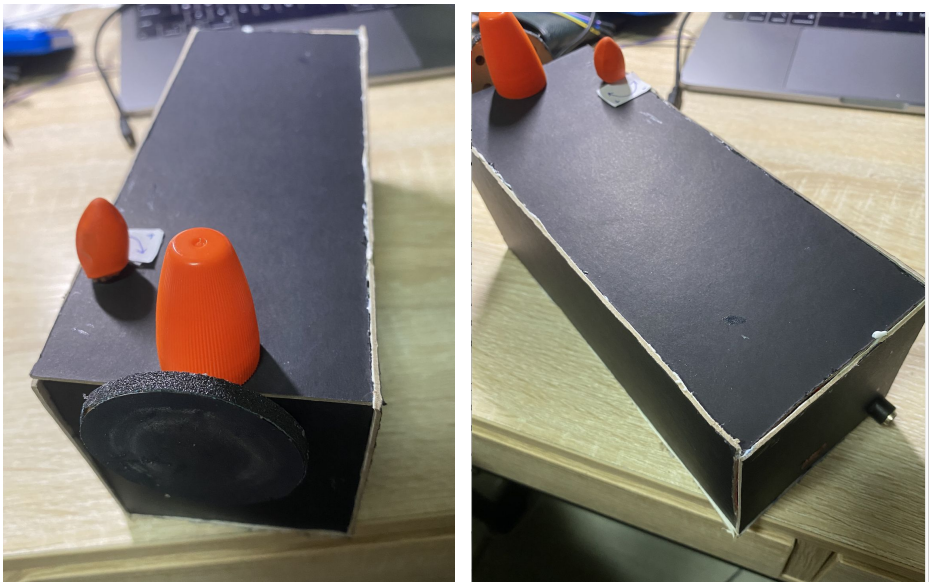


Figure 1: Prototyped the Automatic Nail Filer Front(left) and Back(right)

For the device's operation and its validity, please visit: https://youtu.be/2C8GqKiAacY

Description of Parts:

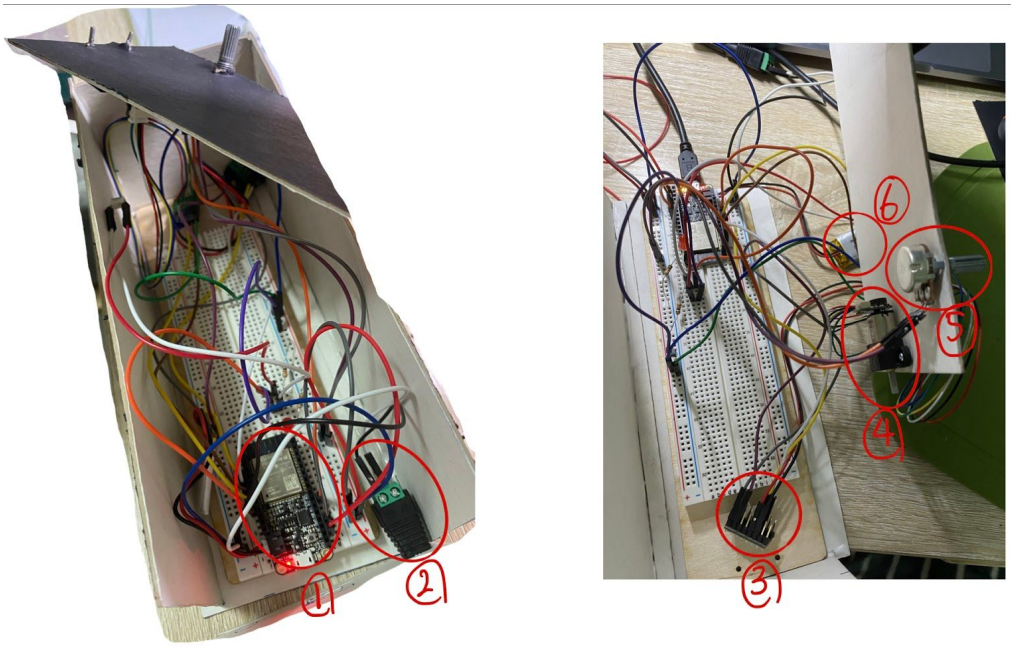


Figure 2: Part assemblies with its circuitry final (left) and initial (right)

Each part indicated by numbers in order represents a ESP32-WROOM, a 5V DC Power supply, a motor driver, a DC motor connected to a rotary encoder, a potentiometer, and a lithium battery. Each component is assembled with a breadboard. The potentiometer is a major component that controls the speed of the DC motor. The motor driver will allow the motor to turn either clockwise or counterclockwise depending on the implemented control system. The lithium battery is installed in order to operate the model without connecting ESP32 device to computer. DC power supply cable is mounted to the back side of the wall of the body, therefore it can be connected to the power supply source at any place. The detail circuit sketch is provided below at circuit description.

Circuit Description:

Controlling the speed of the DC motor was one of the primary goals of this project. The three main components: potentiometer, linear encoder, and motor drive are the major components that are required to successfully complete the project.

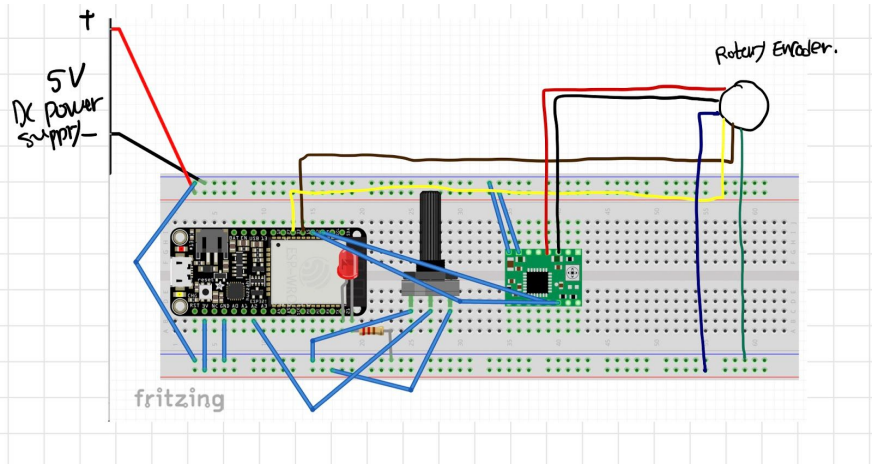
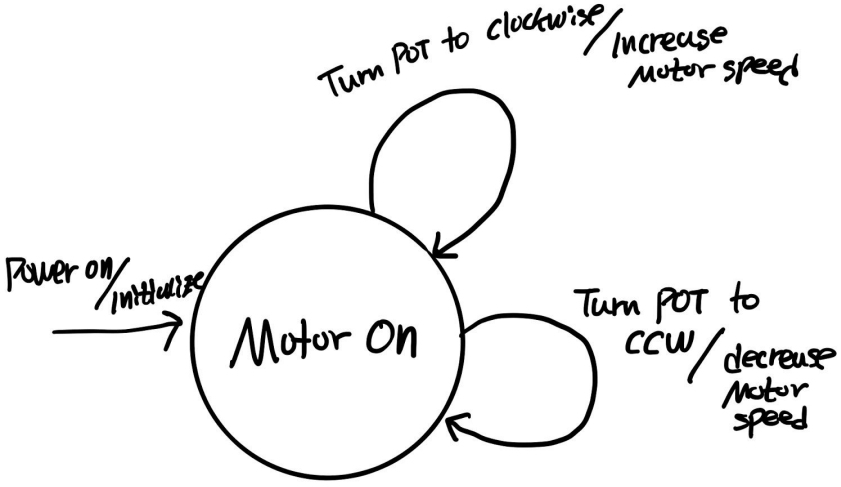


Figure 3: Sketched Circuit Diagram of the Model

When the power supply is connected, the motor is initiated. The potentiometer is the main key to adjust the amount of voltage going through the circuit and finally the motor driver handles the amount of power to drive the motor.

Finite State Machines:



As it described above, the behavior of the automatic nail filer is very simple. When the power supplies, the motor will turn on and stay on. When we move the potentiometer knob clockwise, the motor speed will increase and vice versa. Simplicity is one of the main concerns of this project and the model fully reflects simplicity in order to fulfil my needs.

Appendix

ARDUINO IDE CODES (Button pin is implemented just in case of future development)

```
#include <ESP32Encoder.h>
const int BUTTON_PIN = 13;
const int POT_PIN = A2;
const int ENCODE_A = 27;
const int ENCODE_B = 33;
const int MOTOR_PIN0 = 15;
const int MOTOR_PIN1 = 32;
const int LED_PIN = 21;
const int LED_CHANNEL0 = 0;
const int LED_CHANNEL1 = 1;
int FREQ = 20000;
int RESOLUTION = 8;
int pwmChannel = 0;

int ButtonState = 0;
volatile boolean MotorOn = false;

void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(BUTTON_PIN, INPUT);
  pinMode(POT_PIN, INPUT);
  pinMode(MOTOR_PIN0, OUTPUT);
  pinMode(MOTOR_PIN1, OUTPUT);
  pinMode(LED_PIN, OUTPUT);

  attachInterrupt(digitalPinToInterrupt(BUTTON_PIN), ButtonPressed, RISING);
  ledcSetup(LED_CHANNEL0, FREQ, RESOLUTION);
  ledcSetup(LED_CHANNEL1, FREQ, RESOLUTION);

  ledcAttachPin(MOTOR_PIN0, LED_CHANNEL0);
  ledcAttachPin(MOTOR_PIN1, LED_CHANNEL1);

  ledcWrite(LED_CHANNEL0, 0);
  ledcWrite(LED_CHANNEL1, 0);



  //digitalWrite(LED_BUILTIN, LOW);
  //digitalWrite(LED_PIN, LOW);
  //ledcWrite(MOTOR_PIN0, LOW);
```

```
  attachInterrupt(digitalPinToInterrupt(BUTTON_PIN), ButtonPressed, RISING);
  ledcSetup(LED_CHANNEL0, FREQ, RESOLUTION);
  ledcSetup(LED_CHANNEL1, FREQ, RESOLUTION);

  ledcAttachPin(MOTOR_PIN0, LED_CHANNEL0);
  ledcAttachPin(MOTOR_PIN1, LED_CHANNEL1);

  ledcWrite(LED_CHANNEL0, 0);
  ledcWrite(LED_CHANNEL1, 0);



  //digitalWrite(LED_BUILTIN, LOW);
  //digitalWrite(LED_PIN, LOW);
  //ledcWrite(MOTOR_PIN0, LOW);
  //ledcWrite(MOTOR_PIN1, LOW);

  Serial.begin(115200);

}

void loop() {
  ledcWrite(0,analogRead(POT_PIN));

}
```