ME102B – Fall 2020
Final Project

Study Helper

By Jung Won Shin

Description of the product:

Smartphones have been around us for more than a decade now, and I believe that no other electronic device has changed our lives this much. It reshaped the way we communicate with each other, provided us various platforms where we can watch movies, listen to music, and so on. Despite countless advantages that made our lives more convenient than ever before, there are concerns related to the negative consequences of using smartphones. Among these concerns, one of the biggest problems I have is that I am constantly getting distracted even when I really need to study or to get an important task done. One of the best ways to prevent this is to be able to have a product in which I can put my smartphone inside and lock it for a certain amount of time. I designed a lock box with servo motors and switch. Instruction is as follows.

1. Place your smartphone inside the box
2. Press switch to close the box.
3. Press switch to open the box after your set time has passed

Be sure to cancel all the alarms scheduled to go off since there is no way to open the box until the set time is reached.

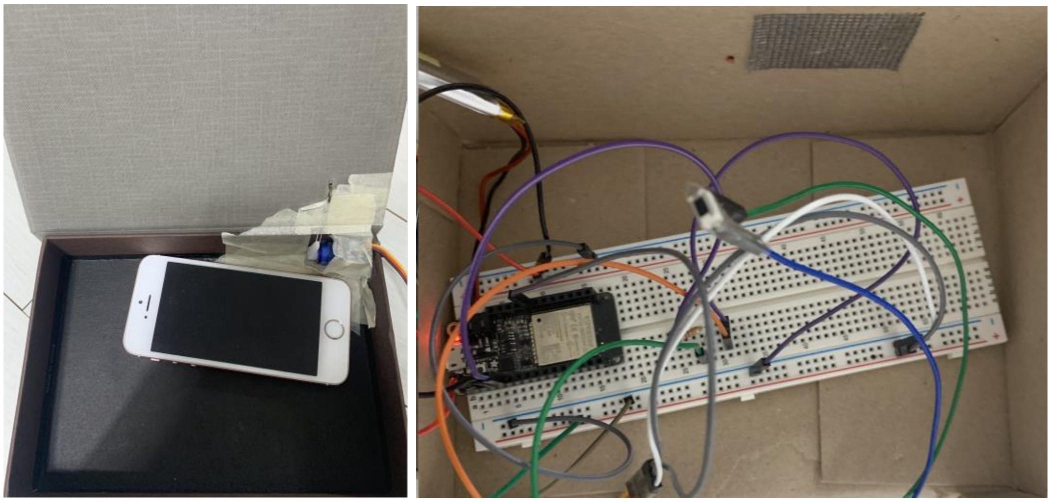
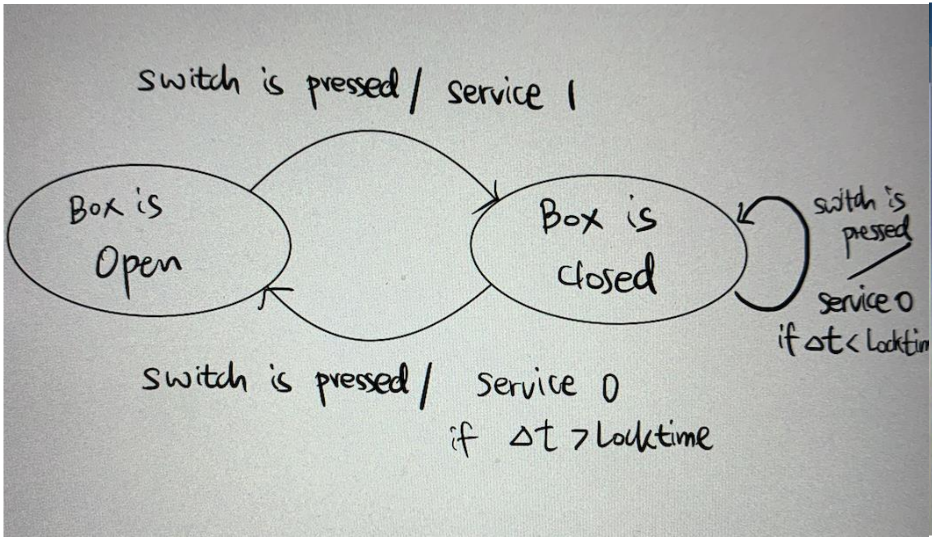
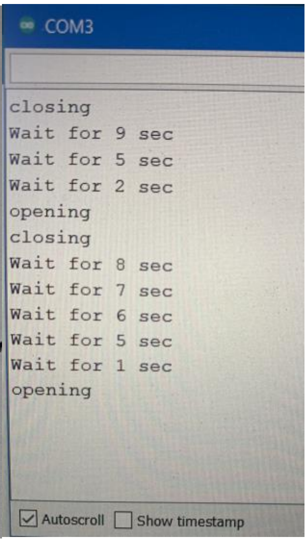


Figure 1 : Lock box(left) and control box(right). Control box sits next to the lock box.

For video demonstration of the product, visit https://youtu.be/FDCCFRyRpCM

Finite state machine:


switch is pressed / service 1
Box is Open
Box is closed
switch is pressed / service 0 if Δt< Locktim
switch is pressed / service 0 if Δt > Locktime

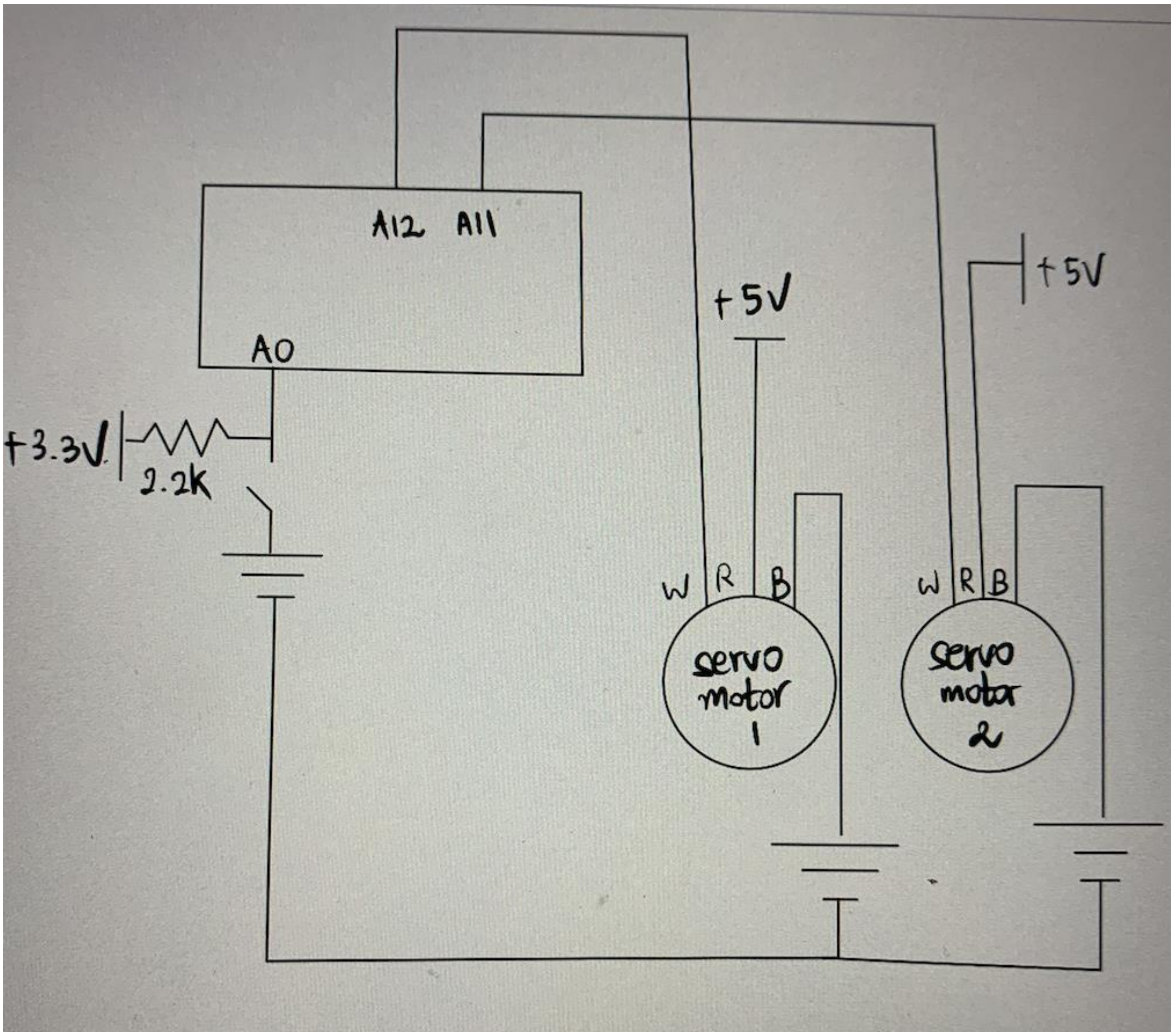


```
closing
Wait for 9 sec
Wait for 5 sec
Wait for 2 sec
opening
closing
Wait for 8 sec
Wait for 7 sec
Wait for 6 sec
Wait for 5 sec
Wait for 1 sec
opening
```

Figure 1: Left is a finite state diagram and right is the demonstration of the product. Set time can be easily modified to the desired period of time. Desired period of time was set to 10 seconds for the demonstration.

Circuit:
Switch: I used an existing switch from the kit. I programmed a microcontroller (ESP32) in such a way that the switch opens the box only after some period of time (desired time period) has passed. Switch will not active otherwise (wait until set time is reached). Pullup resistor is added to prevent floating.

Servo motor: I purchased two servo motors (SG-90, ~$3.00) that opens and closes the box. I calibrated parameters that determine the pulse width for minimum and maximum position of the servo so that both motors are rotating the same degrees. Motors open and close the box at 0 and 180 degrees respectively. (I used only one servomotor in the video because my other servomotor was broken.)



I added an Arduino code in Appendix I for complete understanding.

Appendix I : Arduino code

```
#include <ESP32Servo.h>

//output
#define servoPin1 A12
#define servoPin2 A11

//input
#define BUTTON_PIN A0

//threshold value
#define LOCKTIME 10000 // 1000 = 1s, set it to your desired locktime
#define DEBOUNCE 200

Servo myservo1;
Servo myservo2;

//timer variable
volatile unsigned long myTime;
volatile int myTime_last = 0;
volatile unsigned long myTime_lock = 0;

//state machine variable
boolean state = 0;
volatile boolean buttonPressEvent = false;

//servo position
int closebox = 180;
int openbox = 0;

void setup() {
  pinMode(BUTTON_PIN, INPUT_PULLUP);
  ESP32PWM::allocateTimer(0);
  ESP32PWM::allocateTimer(1);
  ESP32PWM::allocateTimer(2);
  ESP32PWM::allocateTimer(3);
  myservo1.setPeriodHertz(50);
  myservo2.setPeriodHertz(50);
  myservo1.attach(servoPin1, 650, 2400);
  myservo2.attach(servoPin2, 450, 2200);
  Serial.begin(9600);
  attachInterrupt(digitalPinToInterrupt(BUTTON_PIN), buttonIsPressed, HIGH);
}

void loop() {
  switch(state){
    case 0:
        if (buttonPressEvent == true){
          state = 1;
          Service1();
          buttonPressEvent = false;
    }
    break;

    case 1:
         if (buttonPressEvent == true){
           Service0();
           buttonPressEvent = false;
    }
    break;

    default:
    break;

    }
  }

void Service1(){
  Serial.println("Closing");
  myTime_lock = millis();
  myservo1.write(closebox);
  myservo2.write(closebox);
}

void Service0(){
  myTime = millis();
  if (myTime - myTime_lock >  LOCKTIME){
      myservo1.write(openbox);
      myservo2.write(openbox);
      myTime_lock = myTime;
      state = 0;
      Serial.println("Opening");
}
  else{
  Serial.print(“Wait for ”);
  Serial.print(LOCKTIME –(myTime – myTime_lock));
  Serial.println(“ sec”)
  }

  }

void buttonIsPressed(){
  myTime = millis();
  if (myTime - myTime_last > DEBOUNCE){
    buttonPressEvent = true;
    myTime_last = myTime;
}
}
```