

# PhoneBox Final Report

## Introduction

### Problem Description

Phones are an indispensable part of our lives now, but sometimes too indispensable. Sometimes when we need to focus and really keep away from our phones, we find the “procrastination monkey” of our unconscious telling ourselves to pick up our phones and read the latest post or put a few minutes into a game. This horrible affliction is one of the causes to late papers and anguished late nights.

### Design Concept

With this product, our team aims to minimize our phone itches by securing our phone in a box that only grants access to our phones when we truly need them. The user will insert his/her phone into the PhoneBox when he/she wants to focus on his/her task at hand. The PhoneBox will roll the phone inside a slit so that the user is not able to retrieve the phone with his/her hands. Only when the phone receives a notification or call will the PhoneBox roll the phone out for the user to use.

### Background Research

There are many apps in the iOS App store and the Android PlayStore that allow users to disable specified apps from being used or limit their use time. However these apps are not a foolproof way to prevent the user from being distracted. The user can ultimately disable these apps to access the apps that he/she wants to. What distinguishes our product from these apps is we separate the phone from the addicted user completely.

## Components Analysis

### Design Considerations

There were two main design considerations that our team debated over.

One was whether we would use a stepper motor or a high-torque DC motor. Ultimately we chose the DC motor because either way, we would have to utilize another touch-sensor to stop the motors from rotating when the phone hits the bottom of the box. The DC motor was cheaper so we went with it. To roll the phone out of the box, we would use a timer of 10 seconds, which is more than enough time to rotate most phones out of the box. The phones will not tilt over the box and fall because we designed the box to only have 50% of the phone emerge.

Another was whether we would use a microphone or photodiode to detect if the phone receives a notification. At first, it was intuitive for us to implement the microphone because when we think of phone calls, we think of their ringtones. But after some thought we realized that implementing a microphone is impractical. All phones have their speakers placed at different locations so we would not have an ideal spot to place our microphone. Furthermore, sometimes phones are

placed on silent mode, so this would invalidate our microphone. Hence, we went with the photodiode as most phones have their screens lit up when they receive a notification.

### Sensor Characterization

To detect if a phone was being inserted, we decided to use an infrared proximity sensor. This analog sensor emits infrared signal, and measures how much is reflected back. It worked very effectively, and did not give us any problems whatsoever.

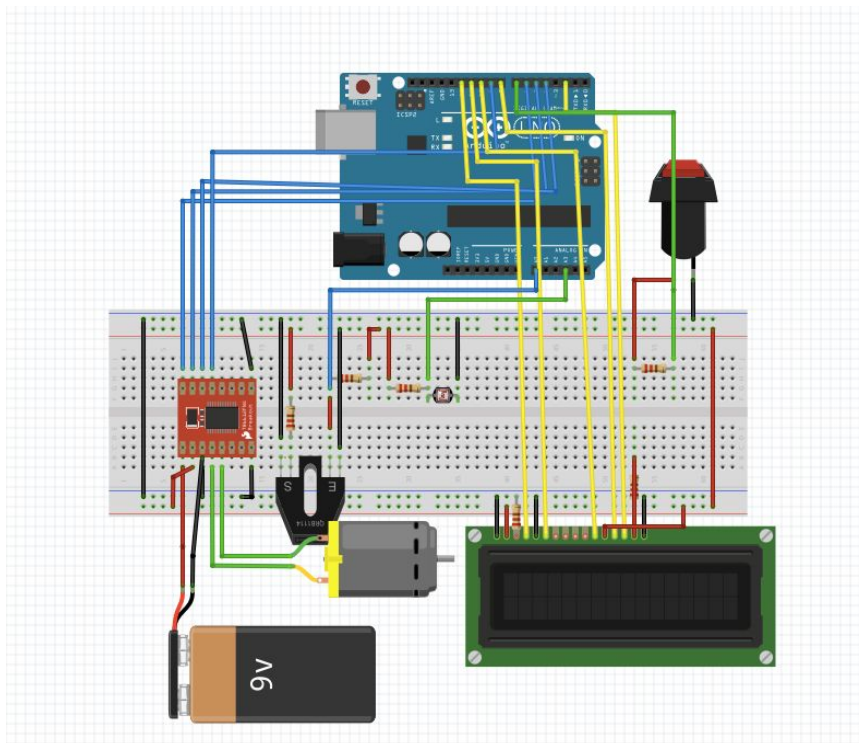
To detect when to stop the motor from spinning the phone in any further, we used a simple pushbutton. This approach was somewhat unreliable, as sometimes the rubber wheels would not gain enough traction against the screen. This prevented the button from being fully pressed. A force sensor would have been a better idea, but we were only aware of this option after ordering our parts.

To detect light emitted from the phone's screen, we used a very simple photocell. This sensor gave us mild difficulty. We often had to change the threshold value in our code for reasons unknown to us.

Fortunately, we were able to get all of these sensors working before presentation time.

## **Design**

### Circuit Schematic



### Physical Construction

The box was constructed by the ECEB machine shop. It has a hard plastic exterior, with a slit for phone insertion. Behind the slit, there are 2 sets of 4 rubber wheels used to guide the phone in. One set guides one side of the phone in. Two of the wheels, attached to the same drive shaft are driven by the motor. The distance between the two sets of wheels can change due to a set of springs, to accommodate phones of different thickness.

Due to our massive amount of circuitry, we chose to run the box with the insides completely exposed, somewhat defeating the purpose of the device.

### Code/Execution

Our code started out as simply a system to test our various analog sensors. We added more and more to the code to implement the other parts of the box, and it soon became complete.

The code is fairly simple. It first waits for a reading above a certain threshold from the proximity sensor. Essentially, the code forms an infinite loop, and breaks out of the infinite loop once the phone is detected. Once the infinite loop ends, the code spins the motor in the direction that brings the phone inwards. The motor continues to run until the phone presses the button. Again, this is accomplished with an infinite loop, and the button breaks out of the infinite loop. The code then waits for a reading above a certain threshold from the photocell. We obtained this threshold from the light emitted from the phone screen once the phone receives a call. When the threshold is reached, the motor is spun in the opposite direction, spitting the phone back out. The motor spins for 10 seconds, more than enough time for any phone to exit. At this point, the code returns to its original state, waiting for the proximity sensor to reach the threshold value.

### **Conclusion**

With our final product, the box successfully recognized the insertion of a phone through the optical detector and correctly spun the phone into the box until it hit the button. Once the phone reaches the bottom, the LCD displays the amount of time the phone has been stored away for the user, informing the user for how long he has focused. Finally, when we had another phone call the phone inside the box, the photodiode correctly recognized a lit up screen and spun the phone out for the user to accept the call.

This project was required us to characterize many different sensors. However, the logic of our project was very straightforward. Most of our difficulties laid in correctly characterizing the sensors. We would find a threshold value for our photodiode only to discover that the value is only right when the environment around it is bright. When we have our phone inserted into the box, the photodiode detects a dark surrounding. This showed us the importance of envisioning the product in use when developing it.

## Appendix - Code

```
#include <SparkFun_TB6612.h>
/* PINOUT
 * Proximity: A0
 * Light : A3
 * Button : 7
 */
int in1 = 4;
int in2 = 5;
int PWM = 6;
int STBY = 9;
Motor motor1 = Motor(in1, in2, PWM, 1, STBY);
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  Serial.print("Begin");
  pinMode(A0, INPUT);
  pinMode(7, INPUT);
  digitalWrite(7, HIGH);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(PWM, OUTPUT);
  pinMode(STBY, OUTPUT);
  pinMode(2, INPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  delay(100);
  int proximity = analogRead(A0);
  float proximityAdjusted = (float)proximity*5.0/1023.0;
  //Serial.println(proximity);
  if (proximity<=800){
    Serial.print("Turn Motor on");
    motor1.drive(-255);
    while(true){
      //Serial.print(digitalRead(7));
```

```
if (!digitalRead(7)){
  Serial.print("Motr off");
  motor1.brake();
  break;
}
}
while(true){
int light = analogRead(A3);
Serial.println(light);
if ( light >=200){
  Serial.print("phone coming out");
  motor1.drive(255);
  delay(12000);
  motor1.brake();
  break;

}
}
}

delay(10000);
}
```