

# Arduino Light Controller

Department of Mechanical Engineering, University of California, Berkeley

December 6, 2020

Eric Kim (3032245503)

## Purpose of this project

Generally, we turn the lights on and off by hand. Sometimes when I leave home, I forget to turn off the light in my room. Currently, the concept of the Internet of Things has emerged as technology has gradually developed. The Internet of Things is a technology that collects, stores, and analyzes data using sensors and communications to various devices. Through this technology, many smart home appliances have been developed and sold. Using smart home appliances, we can adjust the temperature, lighting, and appliances of our homes outside. Currently, smart LED light bulbs are on the market for about \$20, and smart light bulbs can be turned on and off through applications anywhere. I wanted to make my own smart light controller using servo motor and Bluetooth sensor. Through this project, it was a great opportunity to learn the use of Arduino Uno controller, Bluetooth sensor, and servo motors.



Figure 1: Common house light switch



Figure 2: light switch with Arduino controller

For a product explanation and testing video, link: <https://youtu.be/huq4bqdQtDE>

## Design and details

**Location selection of Servomotor:** There have been many trials and errors on where to install the servo motor to turn the lights on and off. After testing whether to install it in the housing or next to the light switch, I realized that fixing it next to the light switch (Figure 3) increases the accuracy of the servomotor to operate the switch. It is also easy to find the required angle to press the switch.

**Servo Motor:** The servo motor on the left is for turning off the lights and the motor on the right is for turning on the lights. When the left motor receives the signal '1' via the Bluetooth sensor, it rotates from 180 to 60 degrees and presses the switch. Conversely, the right motor rotates from 0 to 120 degrees when it receives a signal of 2 and presses the light switch.

**Control Box:** The control box contains three components that Arduino Uno Controller, Breadboard that Bluetooth sensor is installed, Portable battery (power supply). The battery is used to power the Uno controller, and Bluetooth and servo motors are supplied with 5V power from the controller. Breadboard and battery are secured to the housing to prevent movement.

The design of the final box was made by hand, producing boxes of various models several times. I tried to make the housing size as small and light as possible. In the last homework #5, we combined the breadboard with the wooden plate, so the length of the box had to be longer (Fig 2: a black part) to use the board we had. If I make and trim an appearance with a 3D printer or laser cutting machine, it may give you a feeling like a real product.

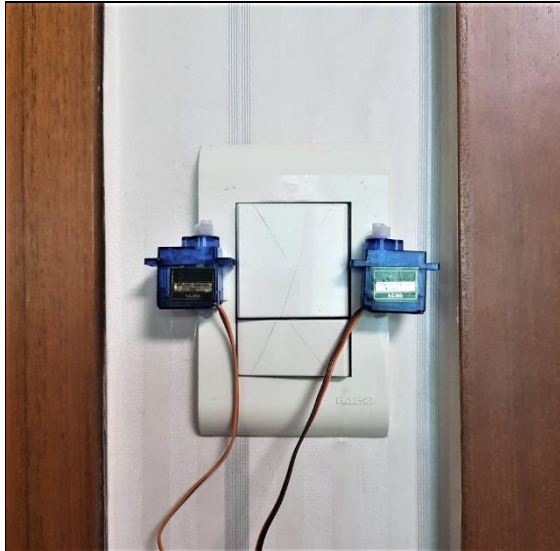


Figure 3: Location of Servo motor

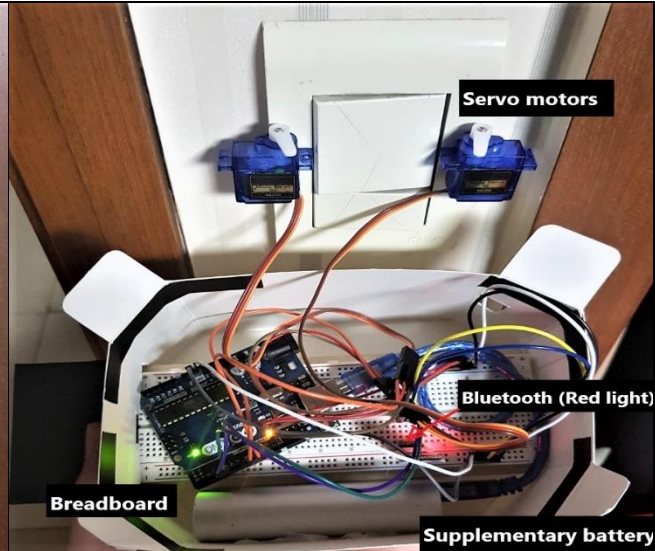


Figure 4: Control Box internal appearance

**Application on phone:** Bluetooth sensor and my phone can be connected through application. Connect the HC-06 Bluetooth module that I used in a machine as shown in Fig 5-1. Then, I choose the switch mode which can switch on and off (Fig 5-2). When the mode is selected, button images in green and red are shown on the screen (Fig 5-3). When I press a green button, the app sends a number signal of 2 to the controller as I set it up and red button sends a number signal of 1 to the controller.

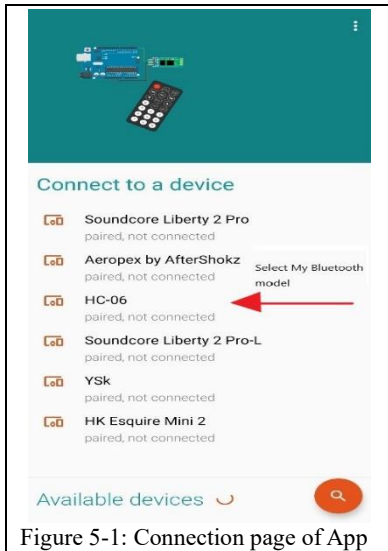


Figure 5-1: Connection page of App

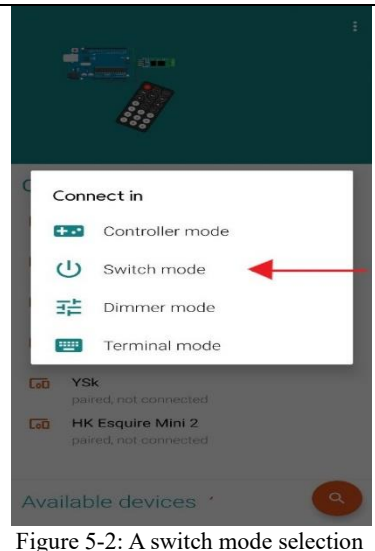


Figure 5-2: A switch mode selection

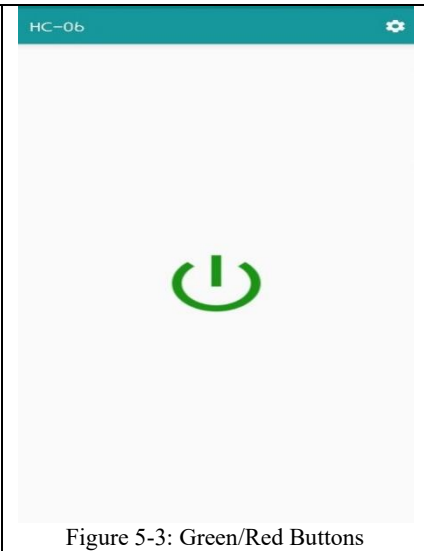


Figure 5-3: Green/Red Buttons

## Circuit Diagram

In this project, I could try new components that we could not use in the class. It is a challenge to have a chance to produce new product with simple sensor and motors. I used three main components: Two servo motors, Bluetooth sensor, and Arduino Uno board.

1. Servo Motor: When I first started the project, I tried to use the DC motor that was included in the kit, but I decided to use the servo motor because I thought the DC motor simply stops when the electricity is cut off in the form of an electric shaft rotating, which was judged difficult to specify the exact stop position by inertia. I bought two servo motor (SG-90, ~\$1.00). SG-90 motors are standard servo motors and can be used up to 0 to 180 degrees. When I used it, I could see that it actually worked 10 to 170 degrees. The red line corresponds to Vcc (5V), the brown (black in the figure) to GND and the orange(yellow) to the data pin.
2. Bluetooth Sensor: It is a Bluetooth module (HC-06, ~\$3.50) that operates in slave mode. This Bluetooth module requires a total of four pins to be connected. Two of these are Vcc and GND and the other two are data pins. The other two pins correspond to RX and TX respectively and are for communication between Bluetooth and microcontroller. Like Serial communication, Bluetooth's RX, TX and Arduino's RX and TX intersected and connected. In other words, it is TX because Bluetooth sends data, and it is RX because Arduino receives this data.
3. Arduino Uno: I used Uno board (~\$4) that I had before. The reason I used the Uno board instead of the ESP32 controller included in the kit was because I wanted to challenge myself. I found that Pin 0 and pin 1 on the Arduino Uno board consist of Rx and Tx. These two pins are pins that are responsible for communication between the Arduino board and the computer, which are not used as much as possible. So, I added a new library function while using Uno Controller. The library is called SoftwareSerial. SoftwareSerial is a feature that helps ordinary I/O data pins operate as RX and TX pins. Thought this library, I was able to complete the connection between the Bluetooth and the controller.

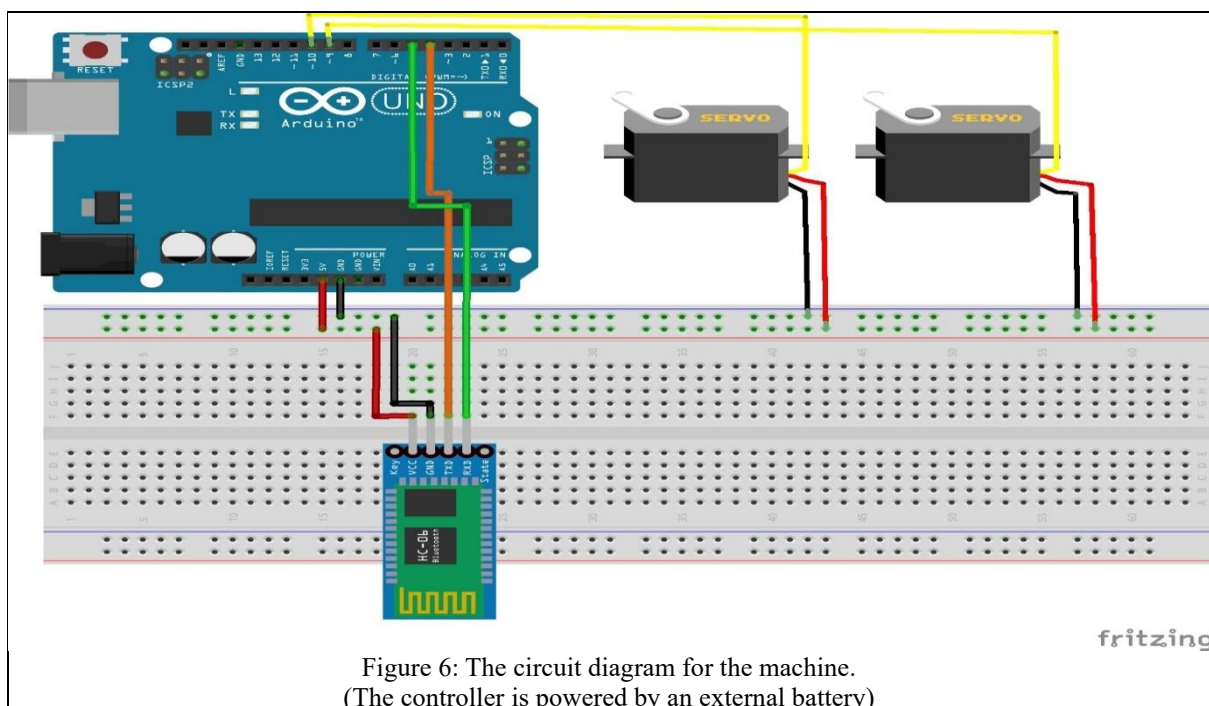
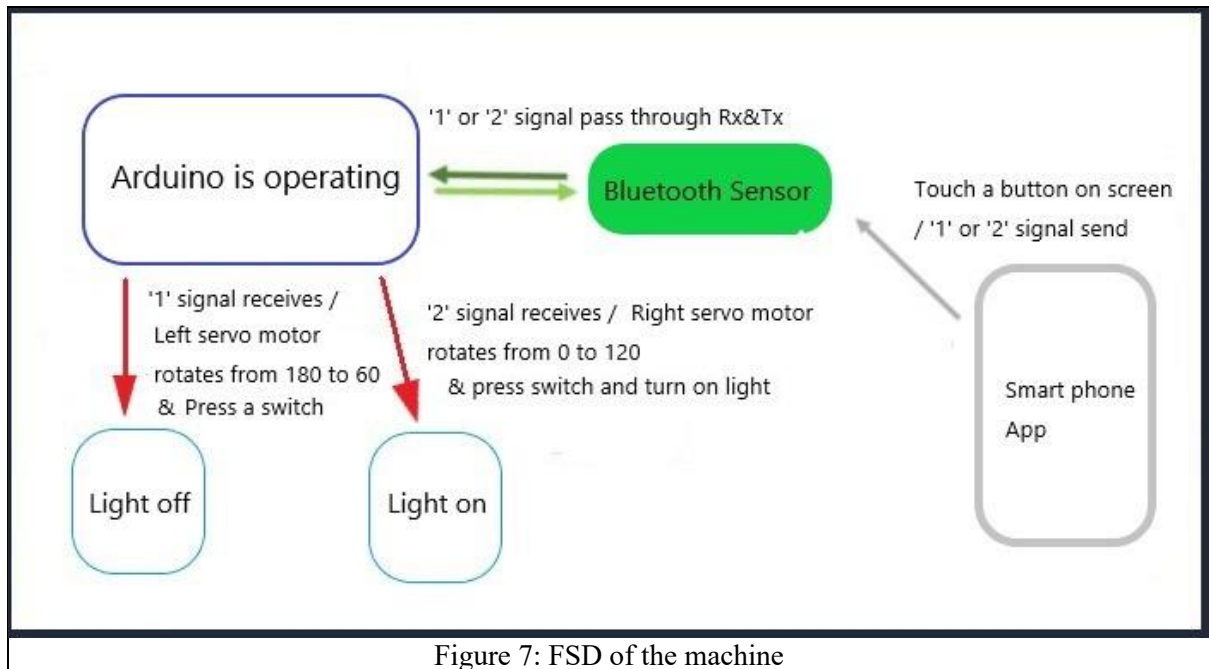


Figure 6: The circuit diagram for the machine.  
(The controller is powered by an external battery)

### Finite state diagram of Machine

In fact, the behavior pattern of the machine lies in controlling the servo motor according to the data and button conditions that come from the smartphone app. For example, if I send 1 or 2 signals to the Bluetooth module through a mobile phone app, the Bluetooth sensor sends data to the controller and the controller sends the output to the servo motor to activate left servo motor or right servo motor. For the final Arduino code, see Appendix.



## Appendix: Arduino Code

```
1 #include<SoftwareSerial.h> //library for Arduino Uno controller and bluetooth
2 #include<Servo.h>
3 //-----servo motor
4 int servoPin1 = 9;
5 int servoPin2 = 10;
6 Servo servol;
7 Servo servo2;
8
9 //-----bluetooth
10 int blue_in = 4;
11 int blue_out = 5;
12 SoftwareSerial bluetoothSerial(blue_in, blue_out);
13 char receiver;
14 int angle;
15
16 void setup() {
17   Serial.begin(9600);
18   bluetoothSerial.begin(9600);
19   servol.attach(servoPin1);
20   servo2.attach(servoPin2);
21   servol.write(180);
22   servo2.write(0);
23 }
24
25 void loop() {
26   if(bluetoothSerial.available() > 0){
27     receiver = bluetoothSerial.read();
28     Serial.write(receiver);
29
30     if(receiver == '1'){ // turn off a light
31       for(angle = 180; angle > 60; angle--){
32         servol.write(angle);
33         delay(2);
34       }
35       delay(1000);
36       servol.write(180); // return to original position
37     }
38     if(receiver == '2'){ // turn on a light
39       for(angle = 0; angle < 120; angle++){
40         servo2.write(angle);
41         delay(2);
42       }
43       delay(1000);
44       servo2.write(0); // return to original position
45     }
46   }
47 }
```