Group 7:

Từ Ánh Minh

Nguyễn Minh Trường

Dư Chấn Nguyên

Trần Minh Triết

Nguyễn Xuân Trường

Bluetooth-controlled traffic light

Task

Arduino, bluetooth module, LEDs.
- In automatic mode, set LED duration by smart phone via Bluetooth.
- In manual mode, turn on/off individual LED manually.

Components

Hardware:

- Arduino UNO
- HC-05 Bluetooth Module
- 5 mm LED: Red
- 5 mm LED: Blue
- 5 mm LED: Green
- Breadboard
- Jumper wires

Software:

- Arduino IDE
- Bluetooth Electronic

Introduction

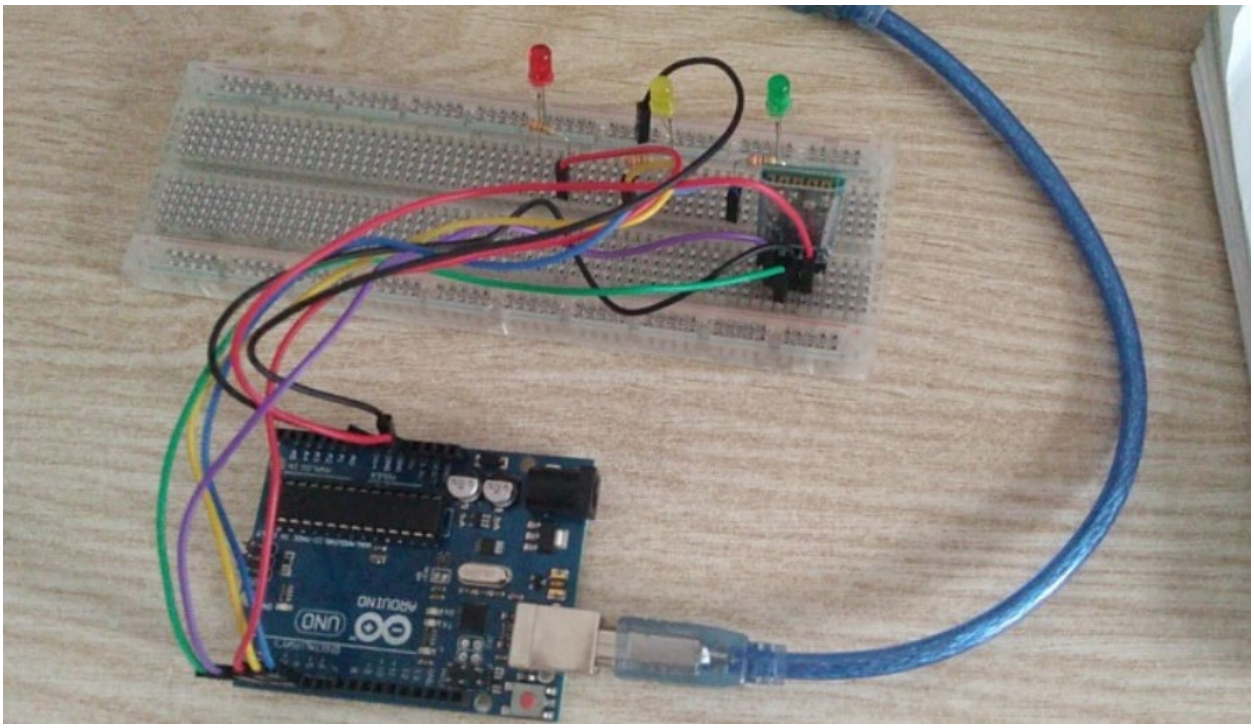
In this project I am going to demonstrate how to control 3 Led using your smart phone with the help of Bluetooth connectivity.

This circuit have two modes 1 for manual control, 2 for automation blinking.

I set manual mode is default, which mean you can control the Led manually at first. By switching the button from the app known as **Bluetooth Electronics** from the mobile phone we will be able to see that the desire led which is code for button pressed will be turned on.

On the other side, when we enter the auto mode by pressing the “Auto” button, we can enter auto mode. When the auto mode is on, we can send a period of time that the automation last for.

Prototype Image's



Hookup

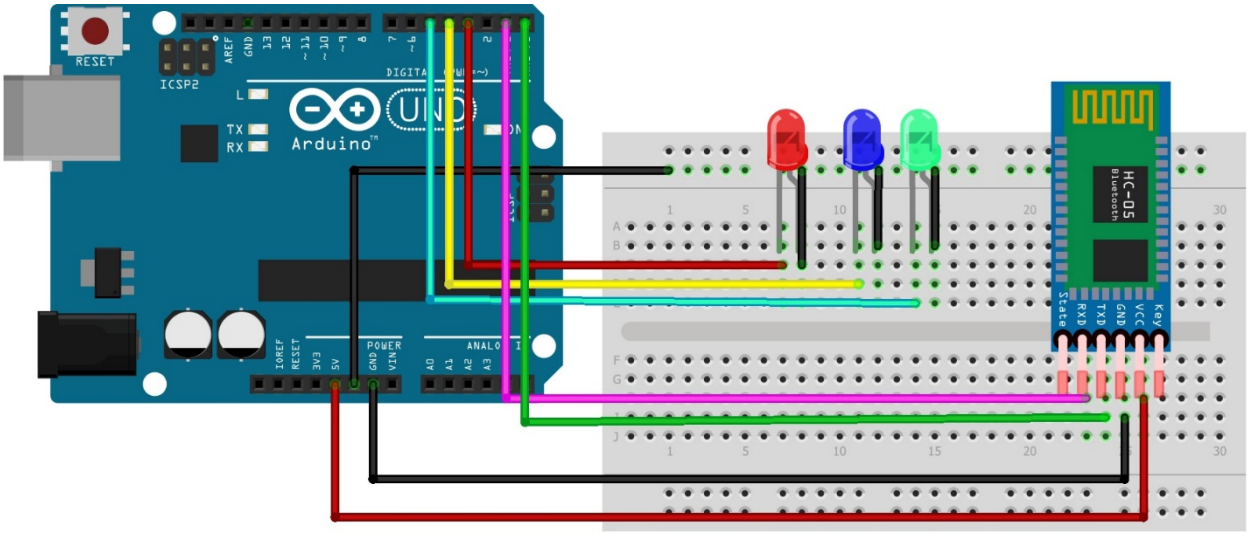
Connection of HC-05

- Hook the **GND pin (Negative Pin)** of HC-05 to **Pin GND** of Arduino.
- Connect Red **VCC Pin (Positive Pin)** of HC-05 to **VCC** of Arduino.
- Connect **TXpin (DataTransferPin)** of HC-05 to **RX** pin of Arduino.
- Connect **RXPin of HC-05** to **TXPin** of Arduino.

Connection of LED

- Hook the **GND Pin(NegativePin)** of all led to **Pin GND** of Arduino.
- Connect **VCC Pin (Positive Pin)** of Red LED to **Pin 3** of Arduino.
- Connect **VCC Pin (Positive Pin)** of Blue LED to **Pin 4** of Arduino.
- Connect **VCC Pin (Positive Pin)** of Green LED to **Pin 5** of Arduino.

Schematics



Codes

```
unsigned long count;

void setup() {
  Serial.begin(9600);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
}

void loop() {

 if(Serial.available()>0)
   {
      char data= Serial.read(); // reading the data received from the bluetooth
module
      switch(data)
      {
        case '1':
        {
          digitalWrite(3, HIGH);
          break;
        }
        case '2':
        {
          digitalWrite(4, HIGH);
```

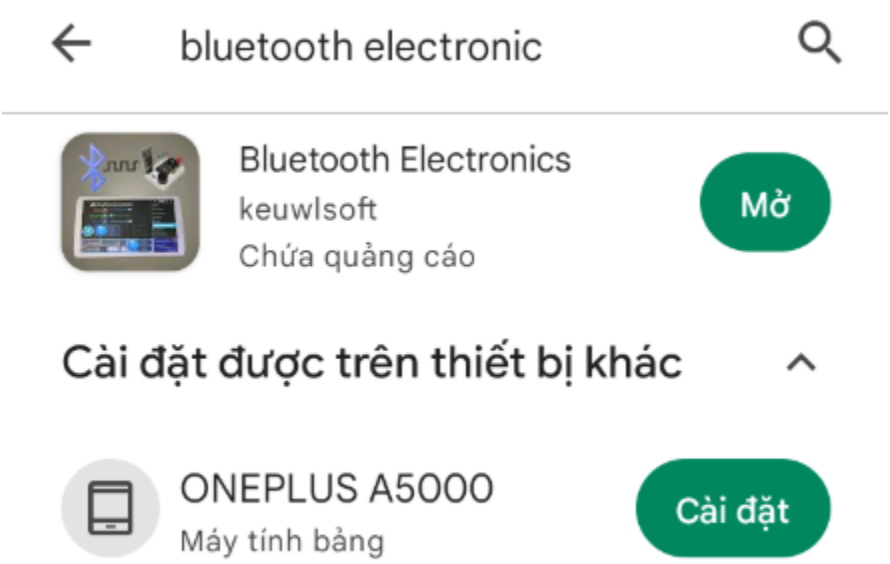
```
  break;
}
case '3':
{
  digitalWrite(5, HIGH);
  break;
}


case '4':
{
  digitalWrite(3, LOW);
  break;
}
case '5':
{
  digitalWrite(4, LOW);
  break;
}
case '6':
{
  digitalWrite(5, LOW);
  break;
}
case 'O':
{
  digitalWrite(3, LOW);
  digitalWrite(4, LOW);
  digitalWrite(5, LOW);
  while (Serial.available()==0)
  {

  } // wait for userinput
  float time = Serial.parseFloat();
  Serial.println(time);
  count = millis();
  while(millis() - count <= time*500*2)
  {
    Serial.println(millis() - count);
    digitalWrite(3, HIGH);
    delay(200);
    digitalWrite(3, LOW);
    digitalWrite(4, HIGH);
    delay(200);
    digitalWrite(4, LOW);
```

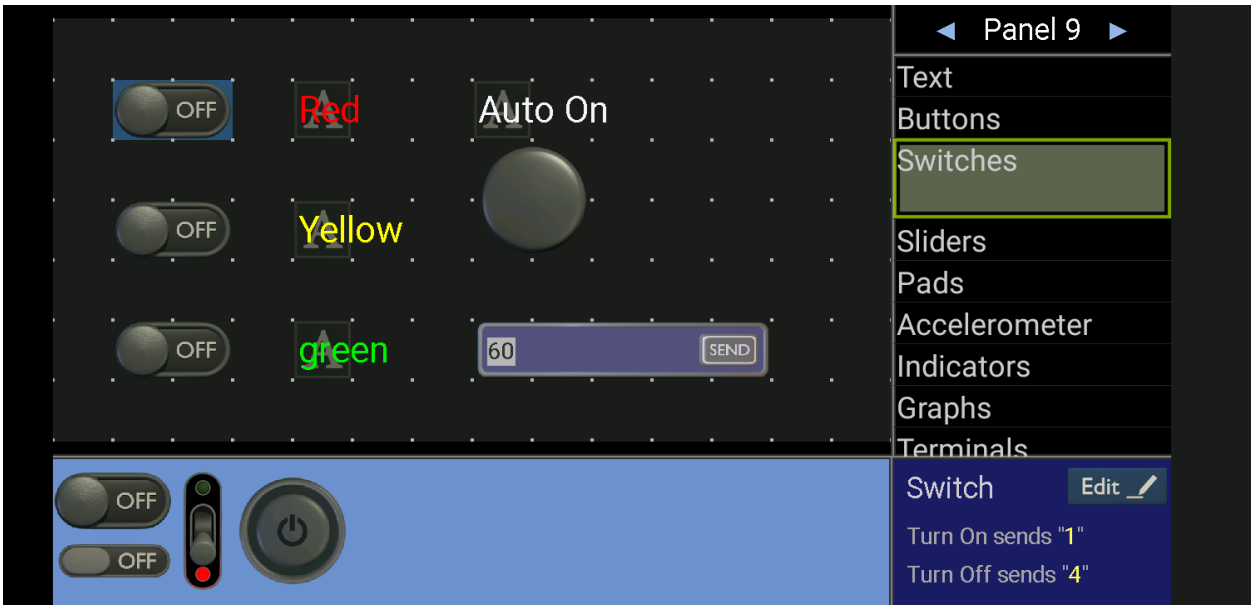
```
            digitalWrite(5, HIGH);
            delay(200);
            digitalWrite(5, LOW);
          }
          break;
        }
        default:
        {
          break;
        }
      }
      Serial.println(data);
   }
   delay(50);
}
```

Setup mobile app

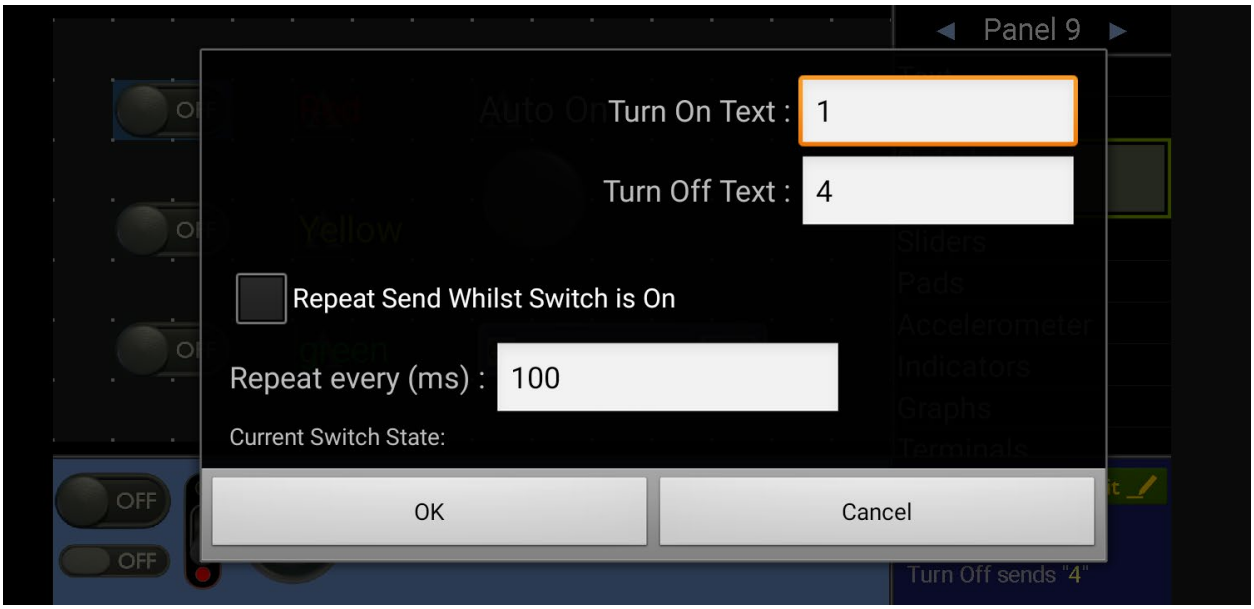


Install **Bluetooth Electronics** on ch play/ apps store

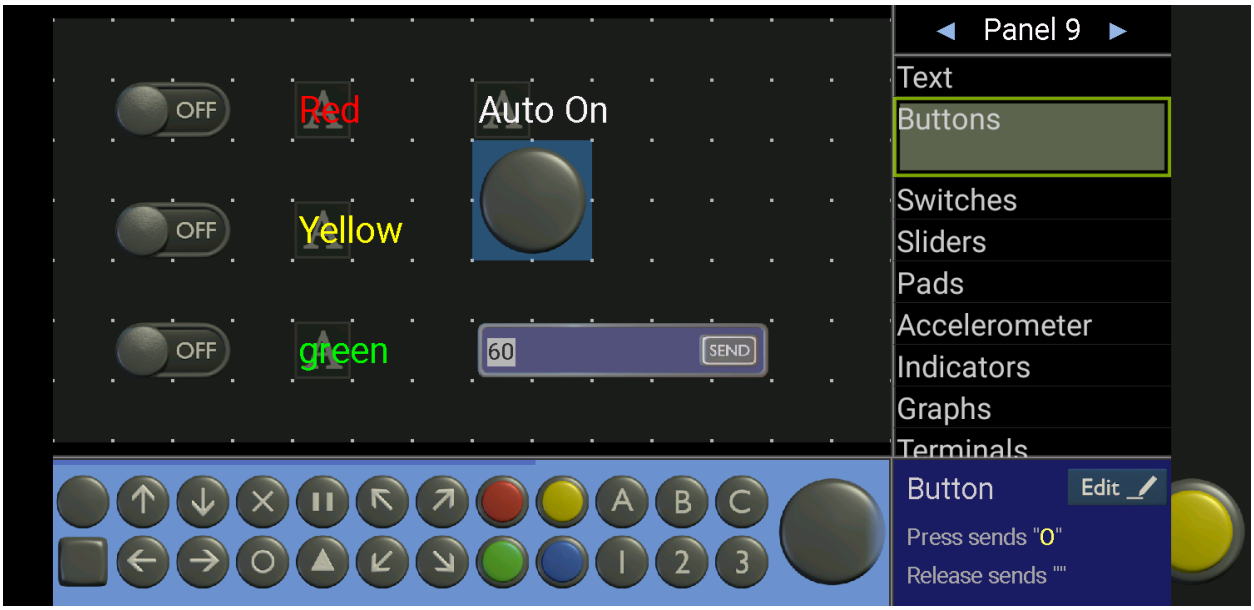
Overview my controller



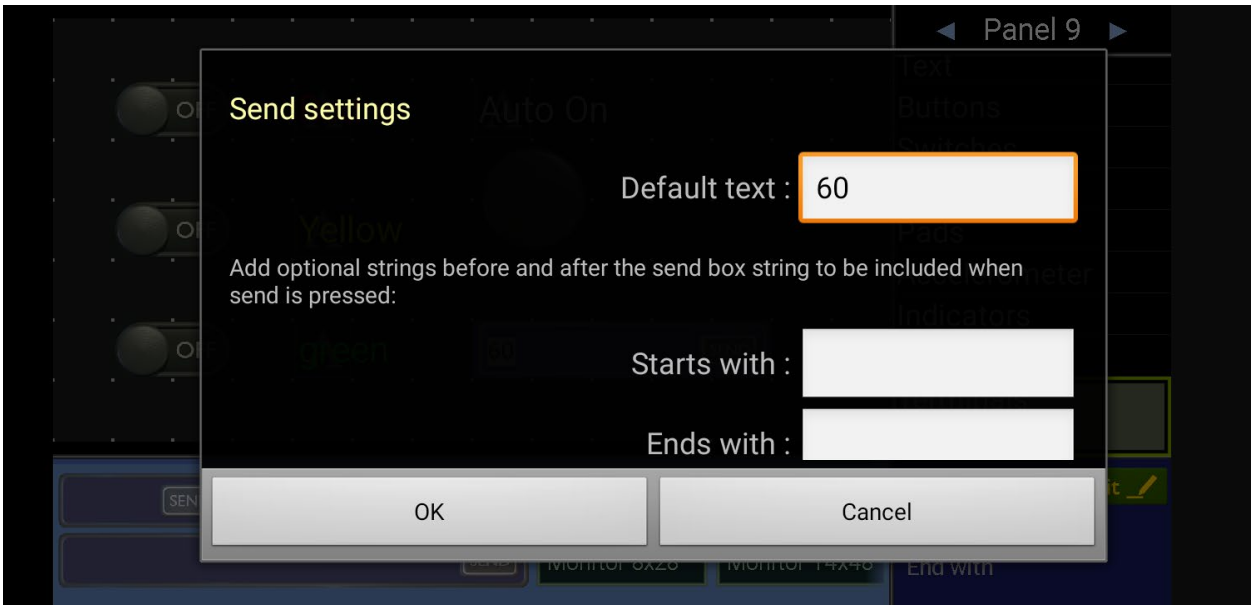
3 switches to control 3 leds



For example in the switch control red led, due to the code, I will send text “1” when the switch is on and “4” when the led is off.



Setting with button to turn enter the auto mode.



A terminal for sending number of second duration of the leds.