

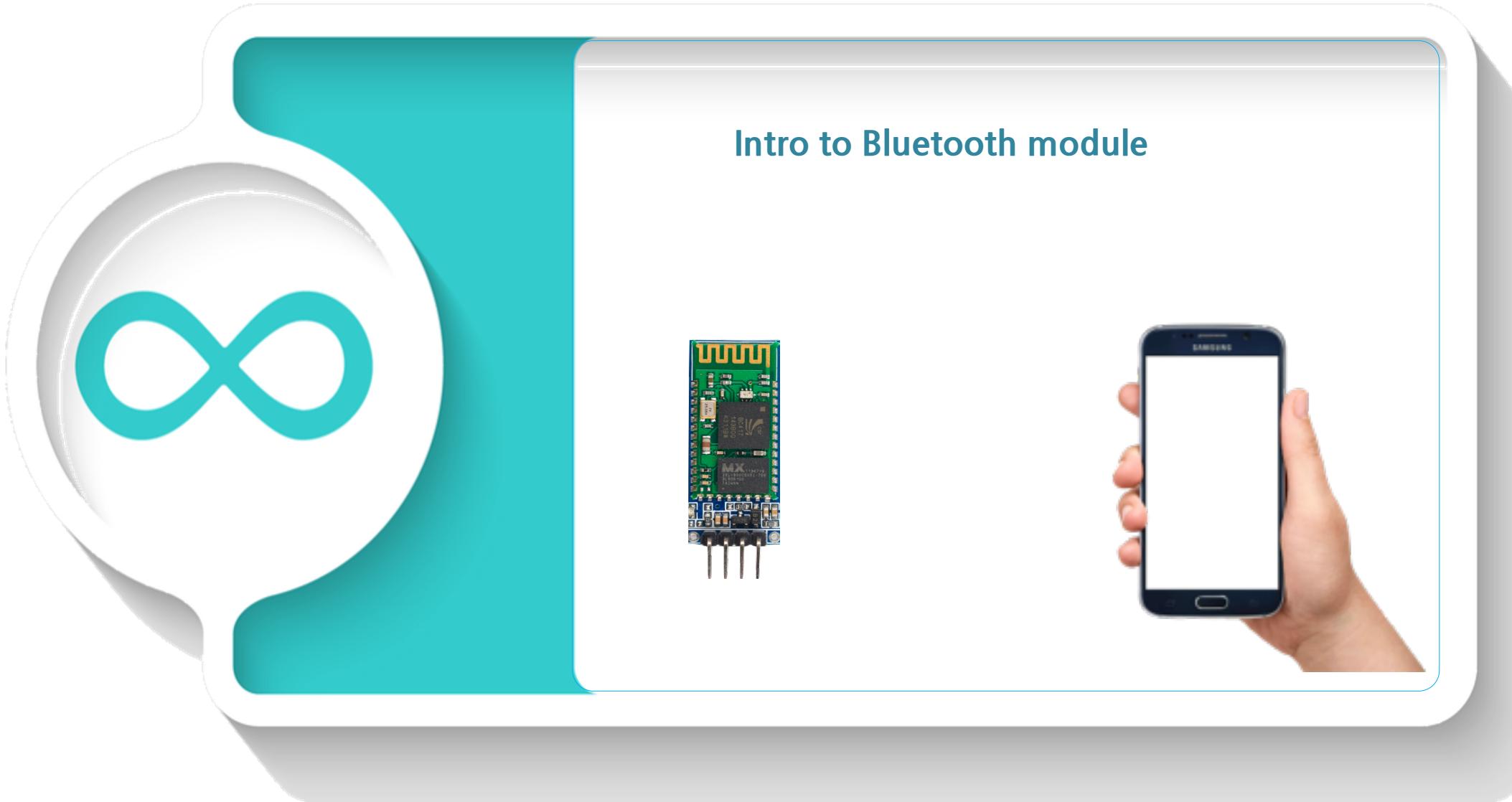
# Ch. 9 Bluetooth

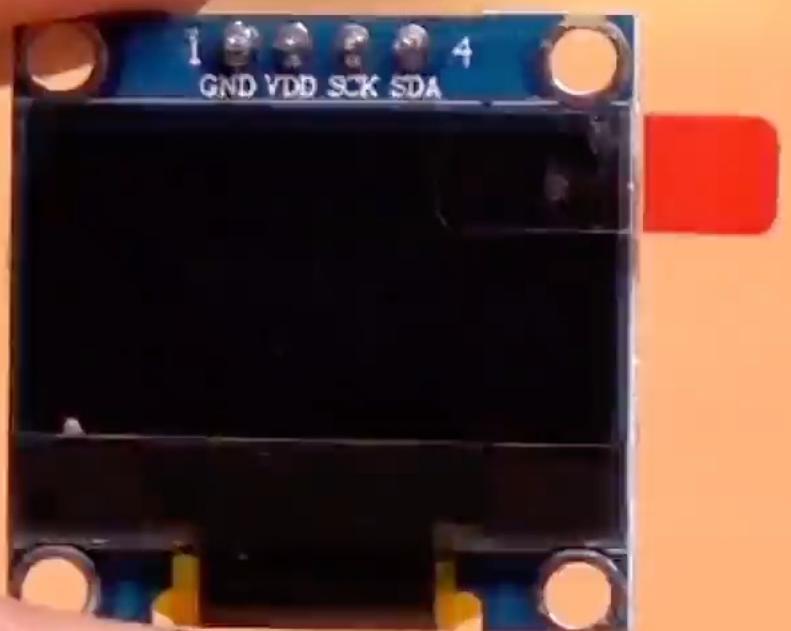
양 세훈

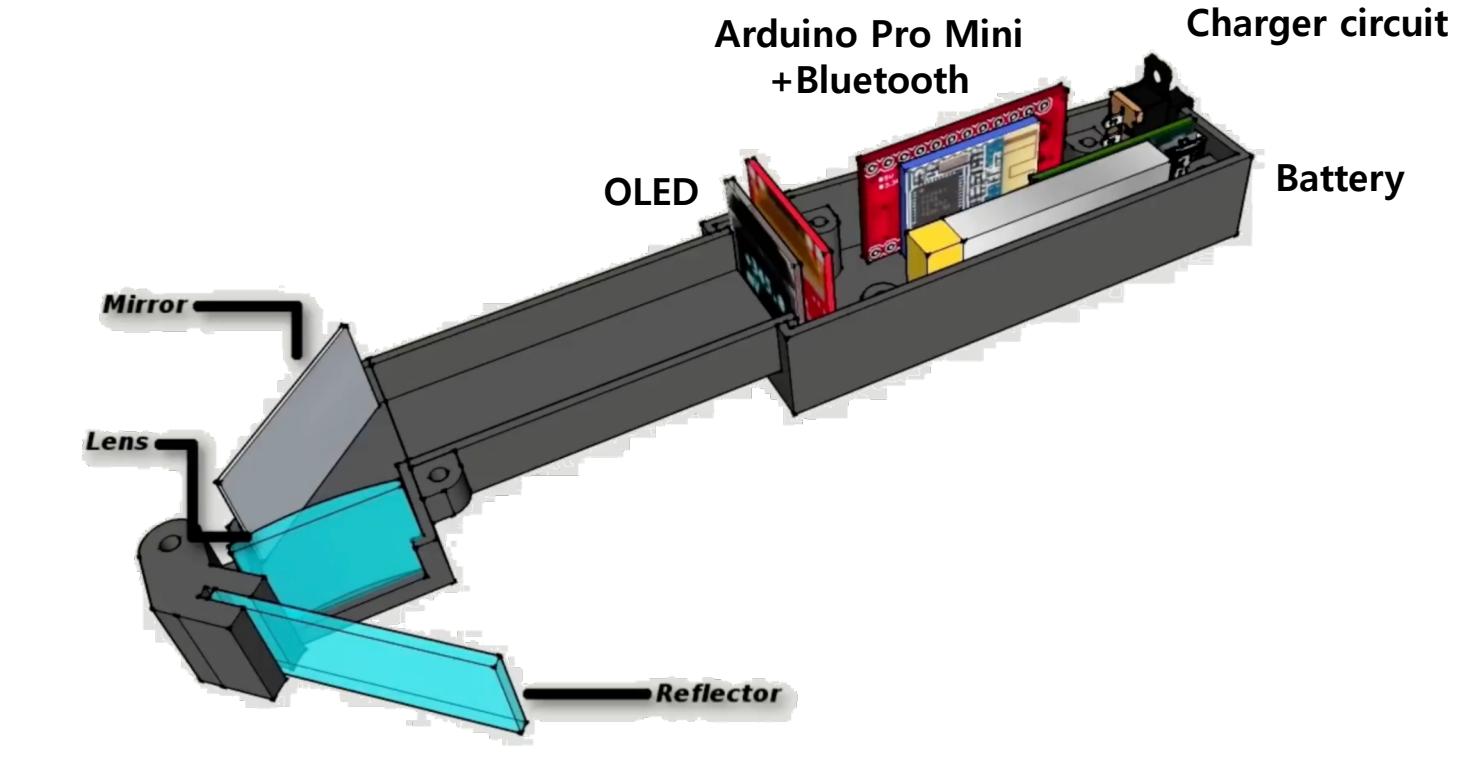


## Ch 9. Bluetooth

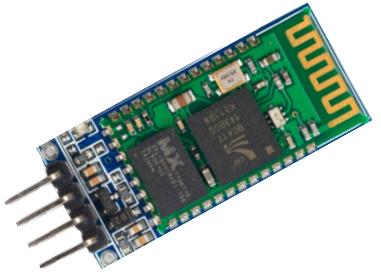
- 3) Bluetooth module naming
- 4) LED control with given App
- 5) Software Serial
- 6) Bluetooth analog data read



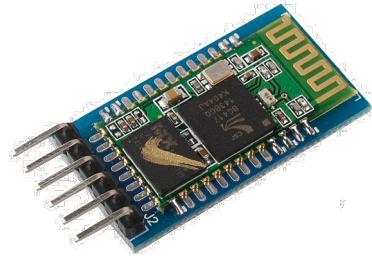




# 1) Intro to Bluetooth module



HC-06



HC-05



BLE Mini



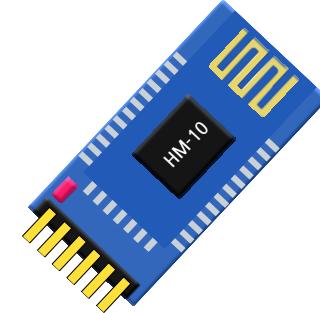
BlueSMiRF



Bluefruit



RN42



and many more

## 1) Intro to Bluetooth module



**Arduino Android IOS HM-10 BLE Bluetooth 4.0  
CC2540 CC2541 Serial Wireless Module**

HM-10  
OctagonStar BLE Bluetooth 4.0  
Universal for Android

HM-02  
HM-03  
HM-04  
05~11

## 1) Intro to Bluetooth module

ISM band : 2.4~2.485 GHz

Normal range : 10 m but able up to 100 m

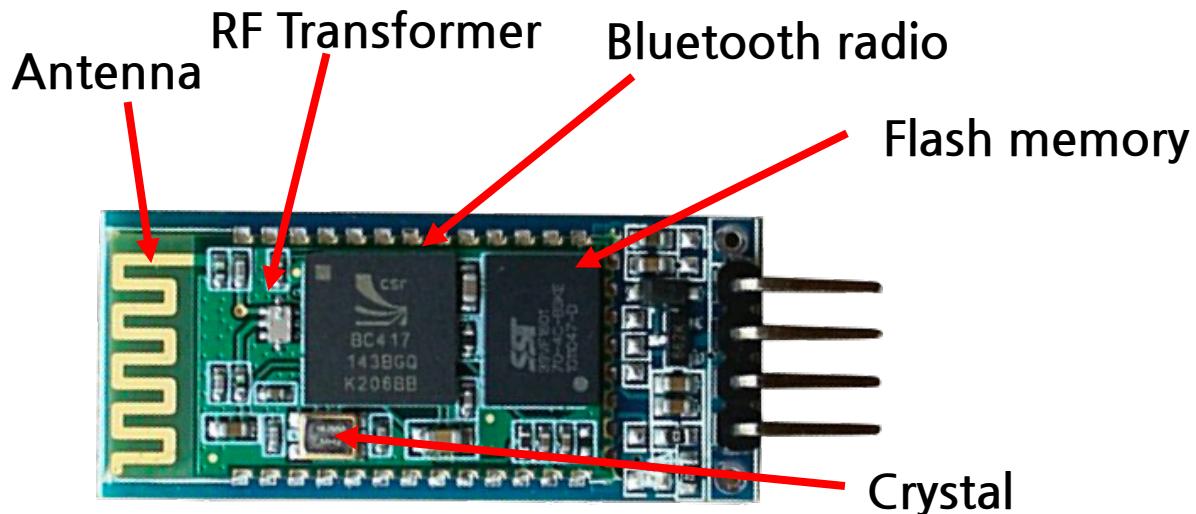
Bluetooth 5 : 2 Mbit/s Samsung Galaxy S8, iPhone 8, 8+

Member : over 30,000 industries

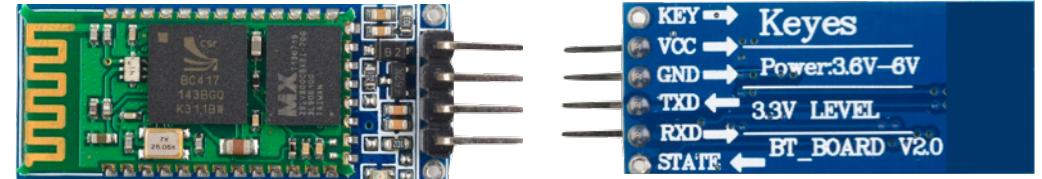
Invented : Ericson in 1994

	BLE 4.0	Wi-Fi
Frequency(GHz)	2.4	2.4, 5
Throughput(Mbps)	1	7(b),25(g),150(n)
Range(m)	50	100~200
Power consumption	Low	High

# 1) Intro to Bluetooth module

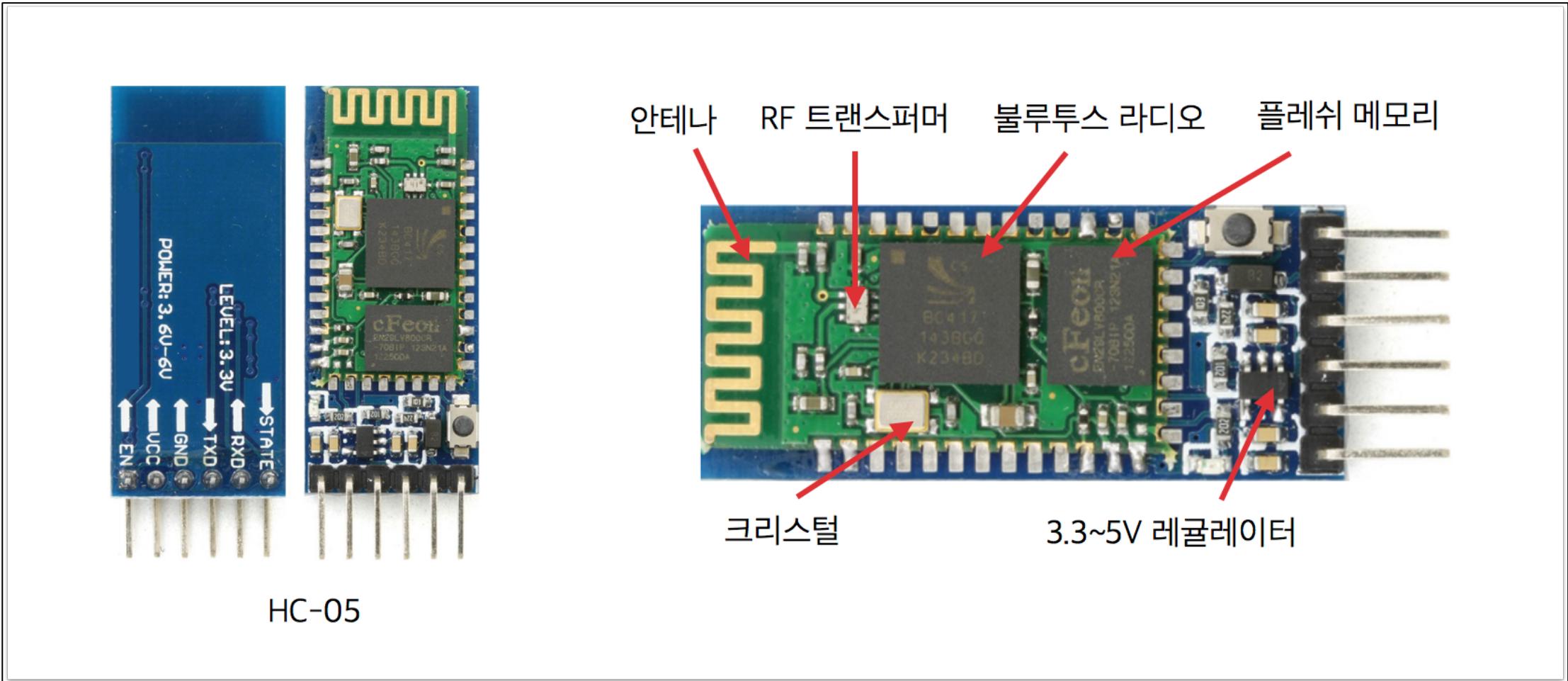


Bluetooth HC-06 Module



# 1) Intro to Bluetooth module

## Bluetooth HC-05 Module



### 3) Bluetooth naming



## Bluetooth Module Naming

목적 : 사용할 모듈에 이름 부여하기

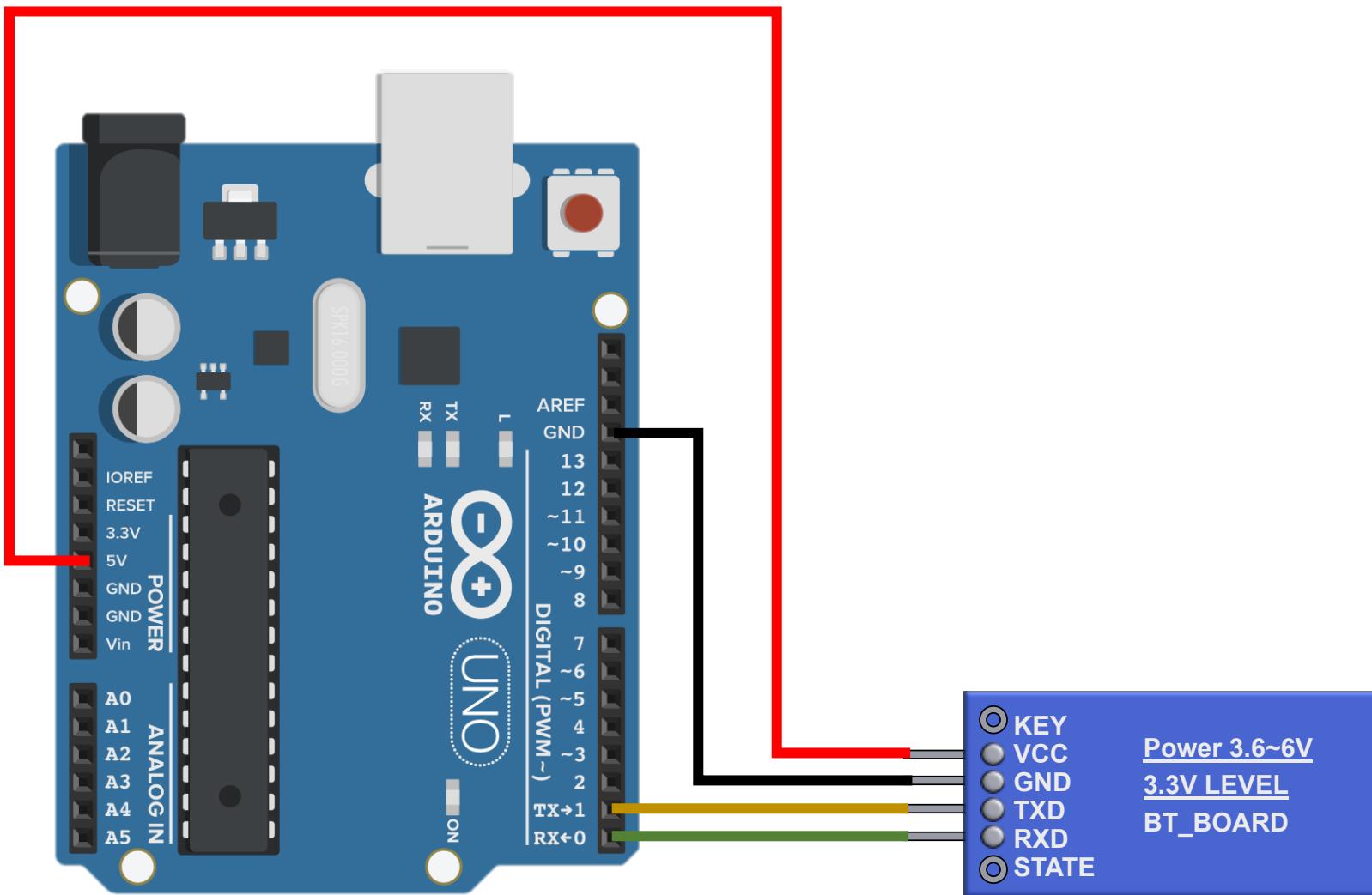


HC-06 를 대체할  
이름을 작명해 주세요 !!



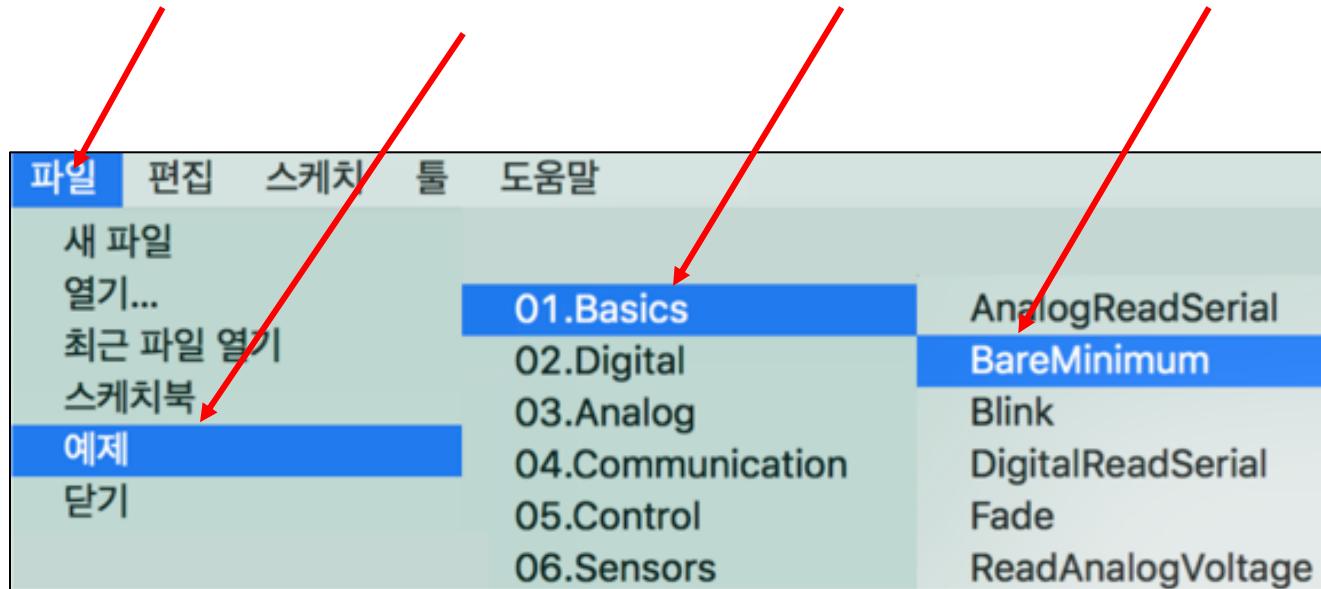
### 3) Bluetooth naming

HC-06 모듈에 이름 부여 하기 circuit



### 3) Bluetooth naming

Upload Bare minimum Sketch

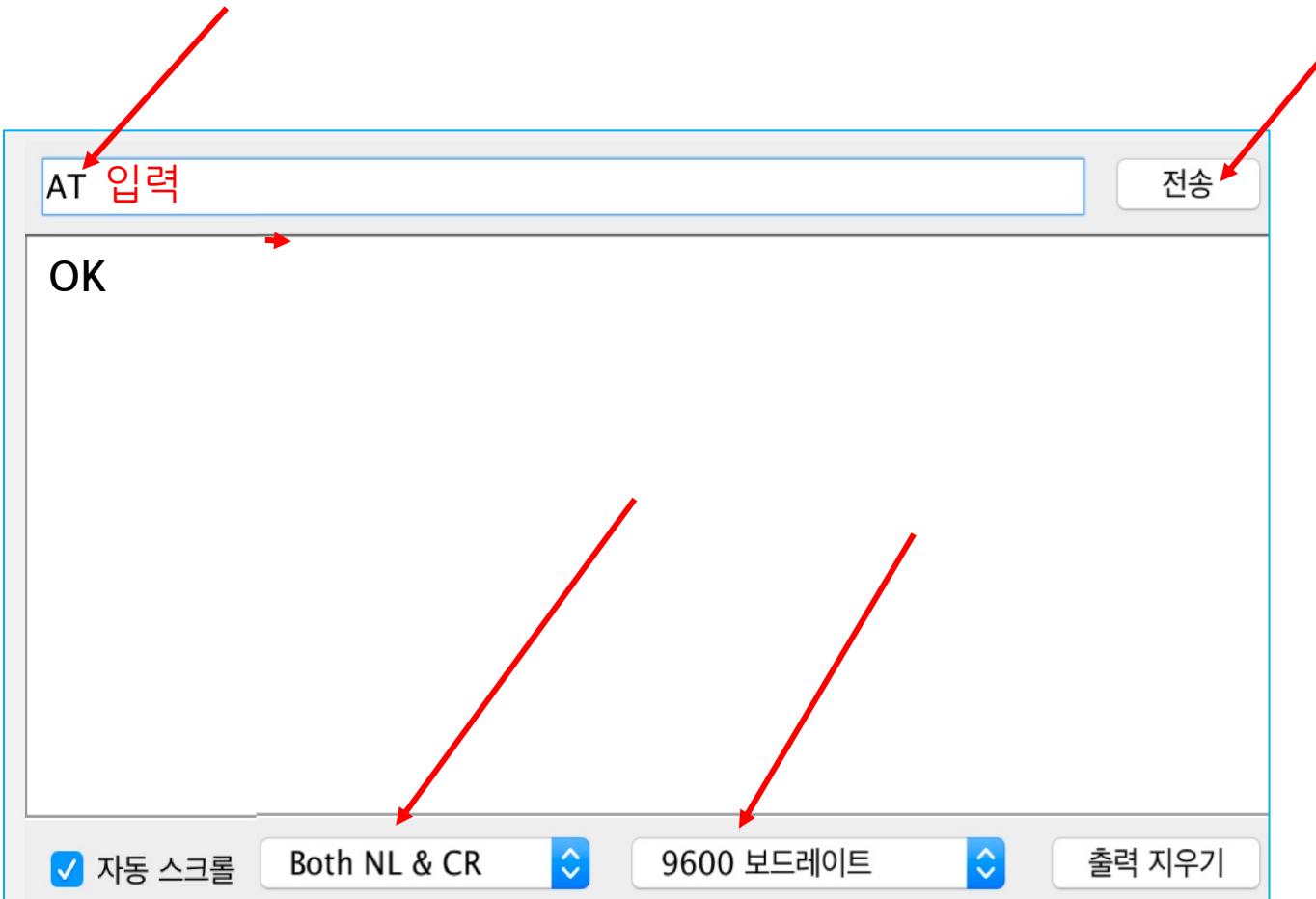


A screenshot of the Arduino IDE code editor. The title bar shows the sketch name as 'sketch\_apr07a §'. The code itself is minimal, consisting of the standard setup and loop functions:

```
1 void setup() {  
2 }  
3  
4 }  
5  
6 void loop() {  
7 }  
8 }
```

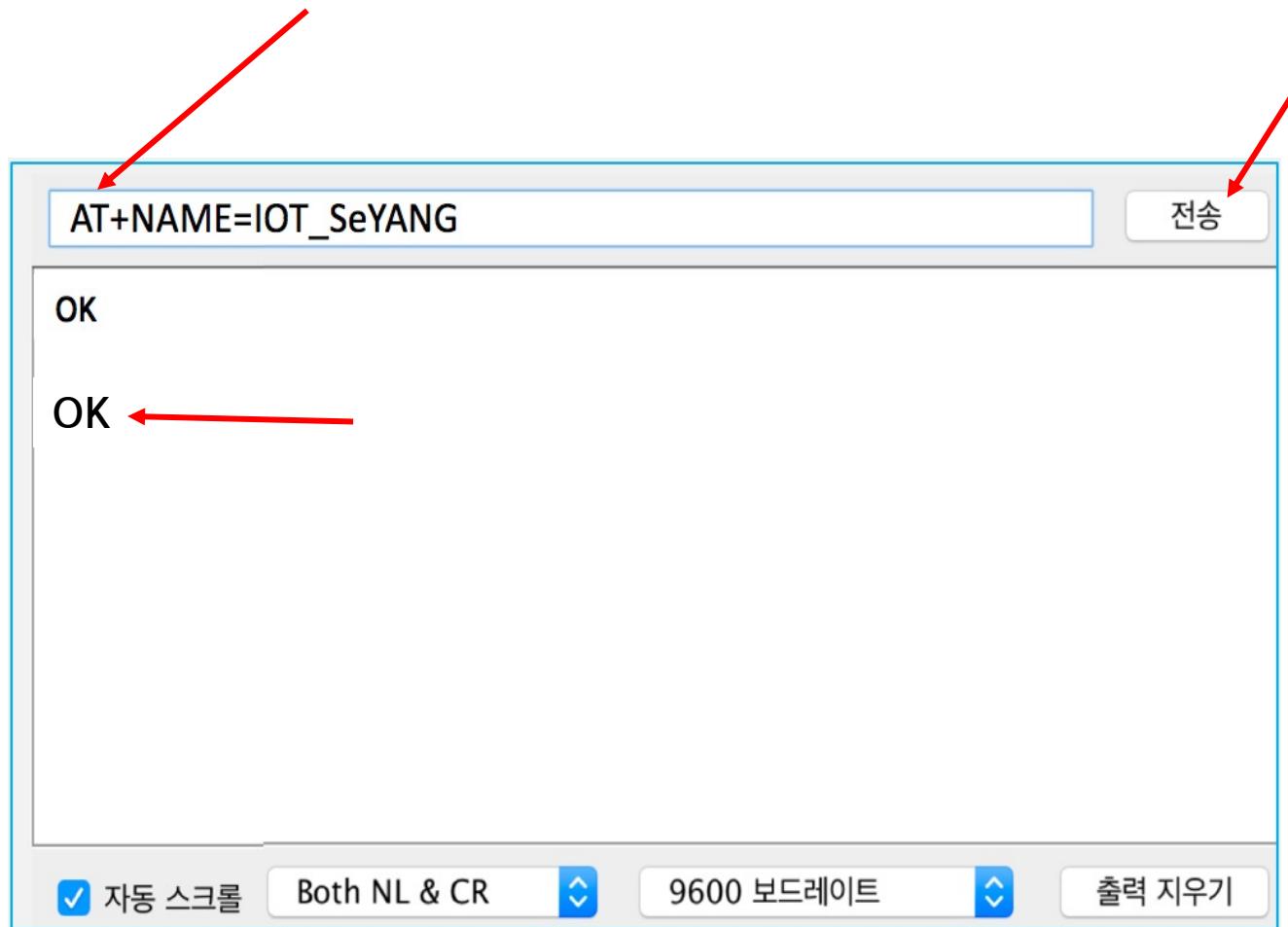
### 3) Bluetooth naming

#### AT command



### 3) Bluetooth naming

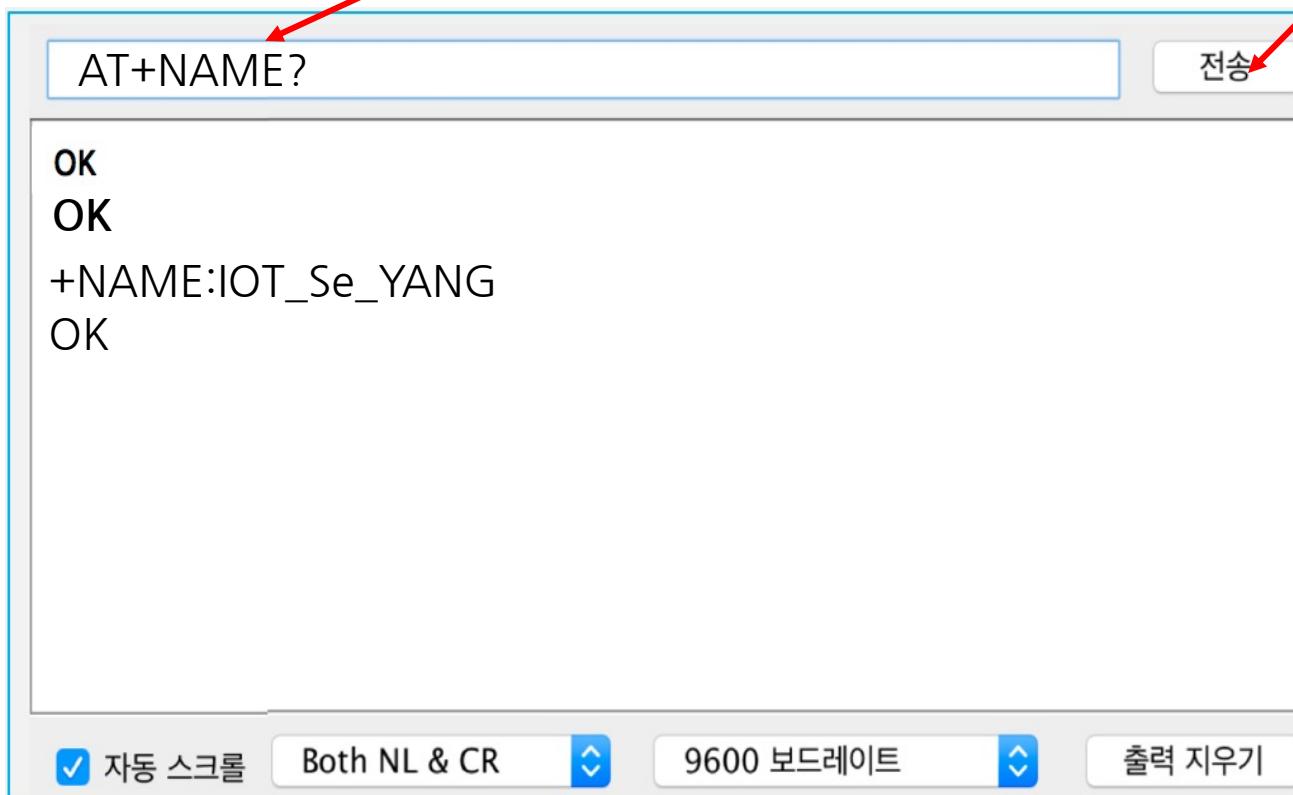
AT command to give NAME



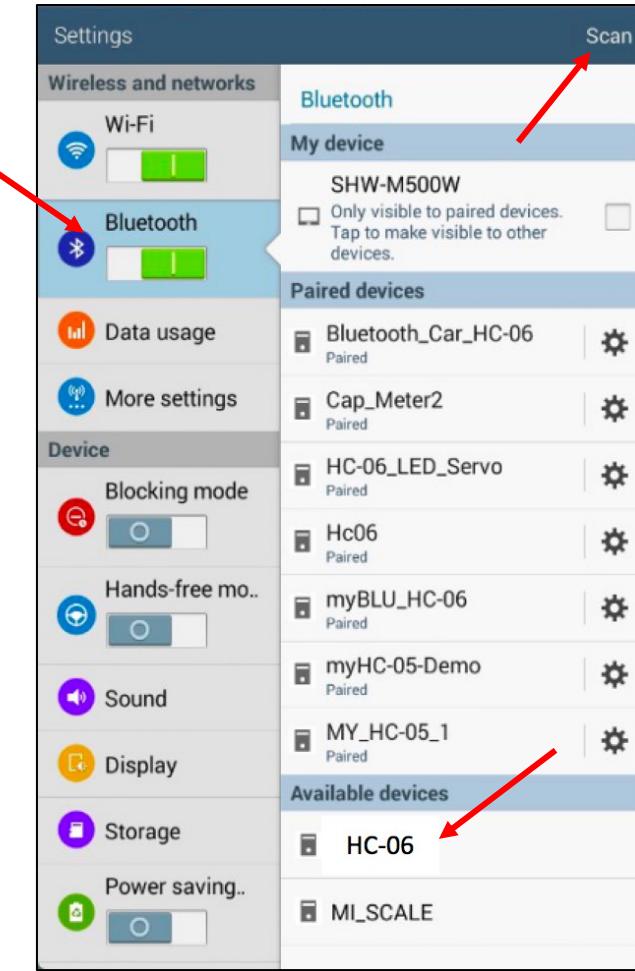
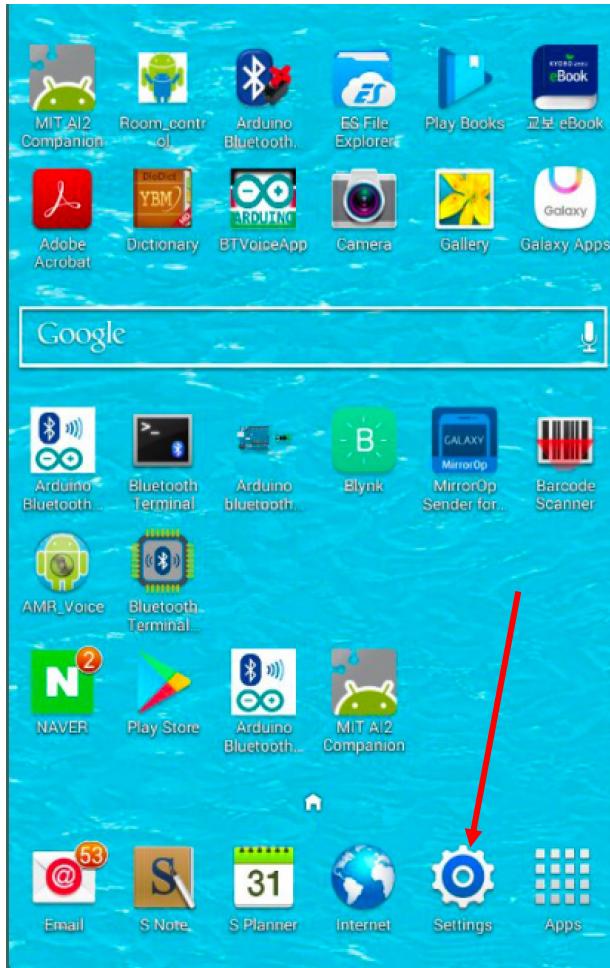
OR  
AT+NAME=IOT\_SeYANG

### 3) Bluetooth naming

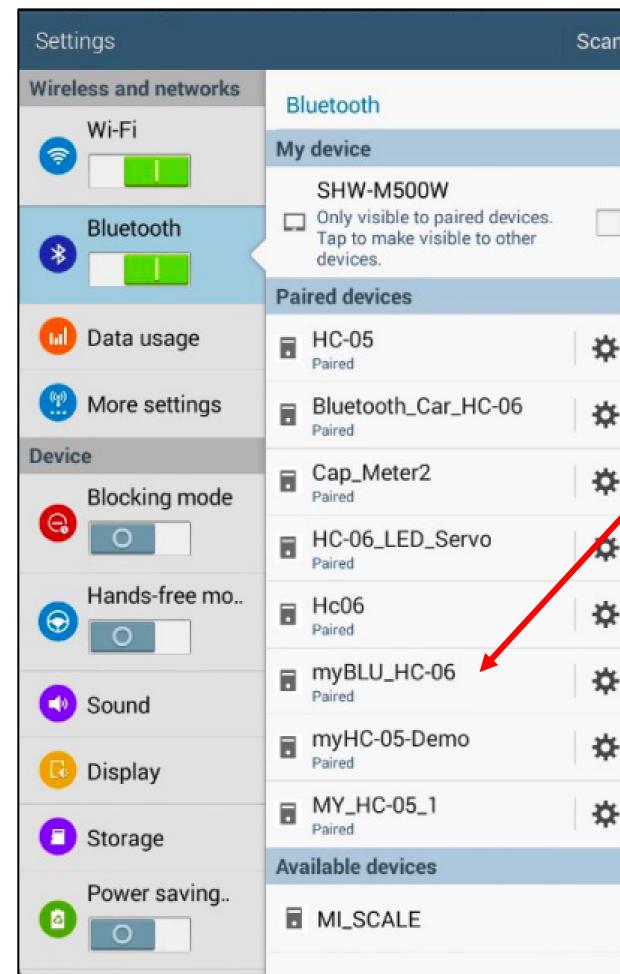
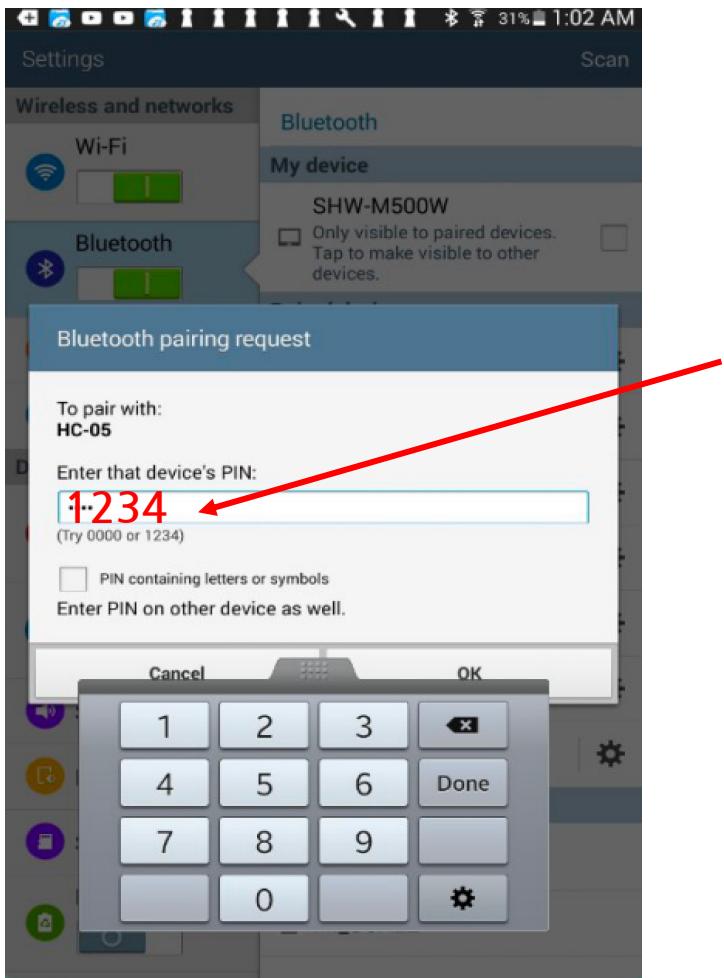
#### Ask module NAME



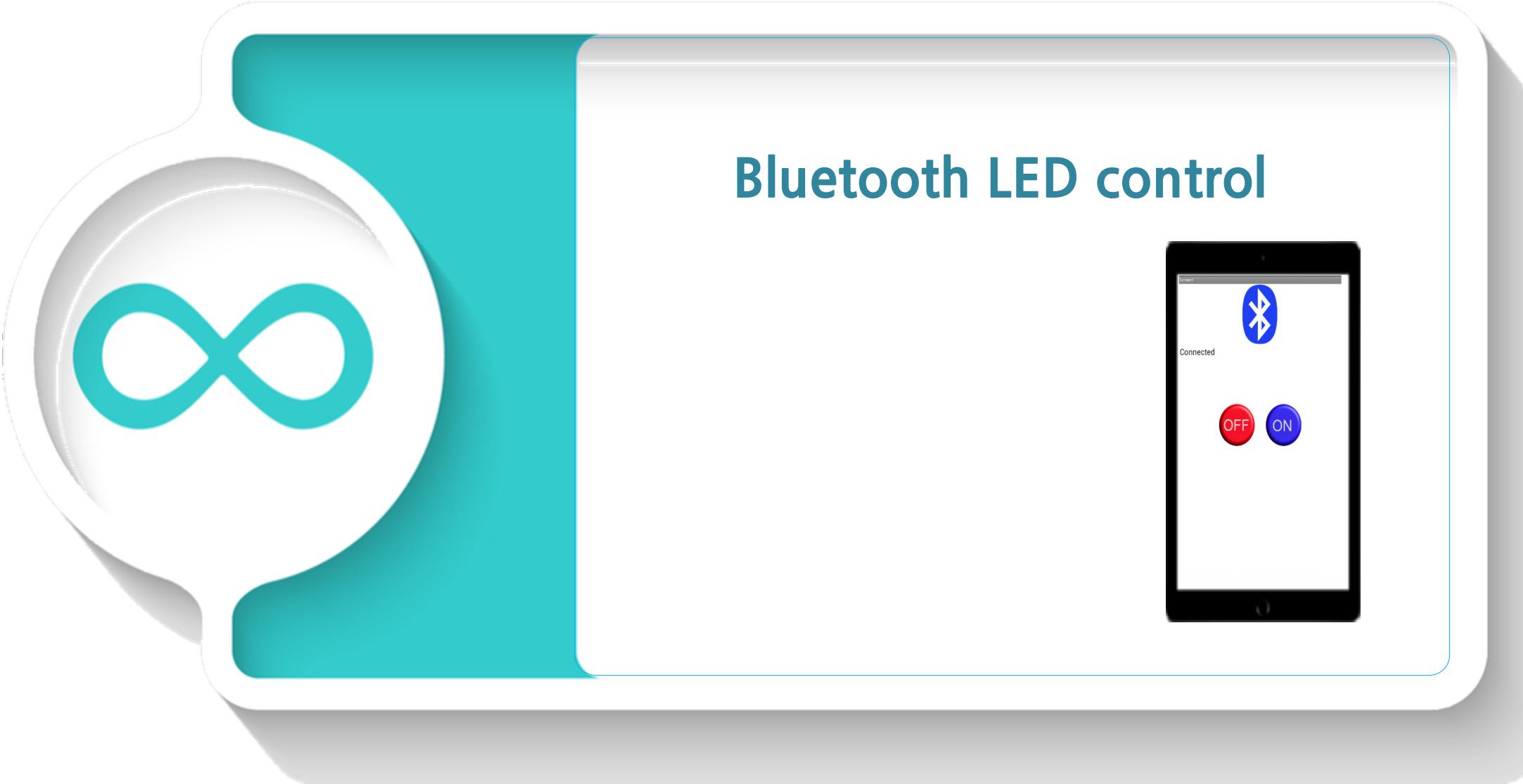
## Bluetooth Pairing



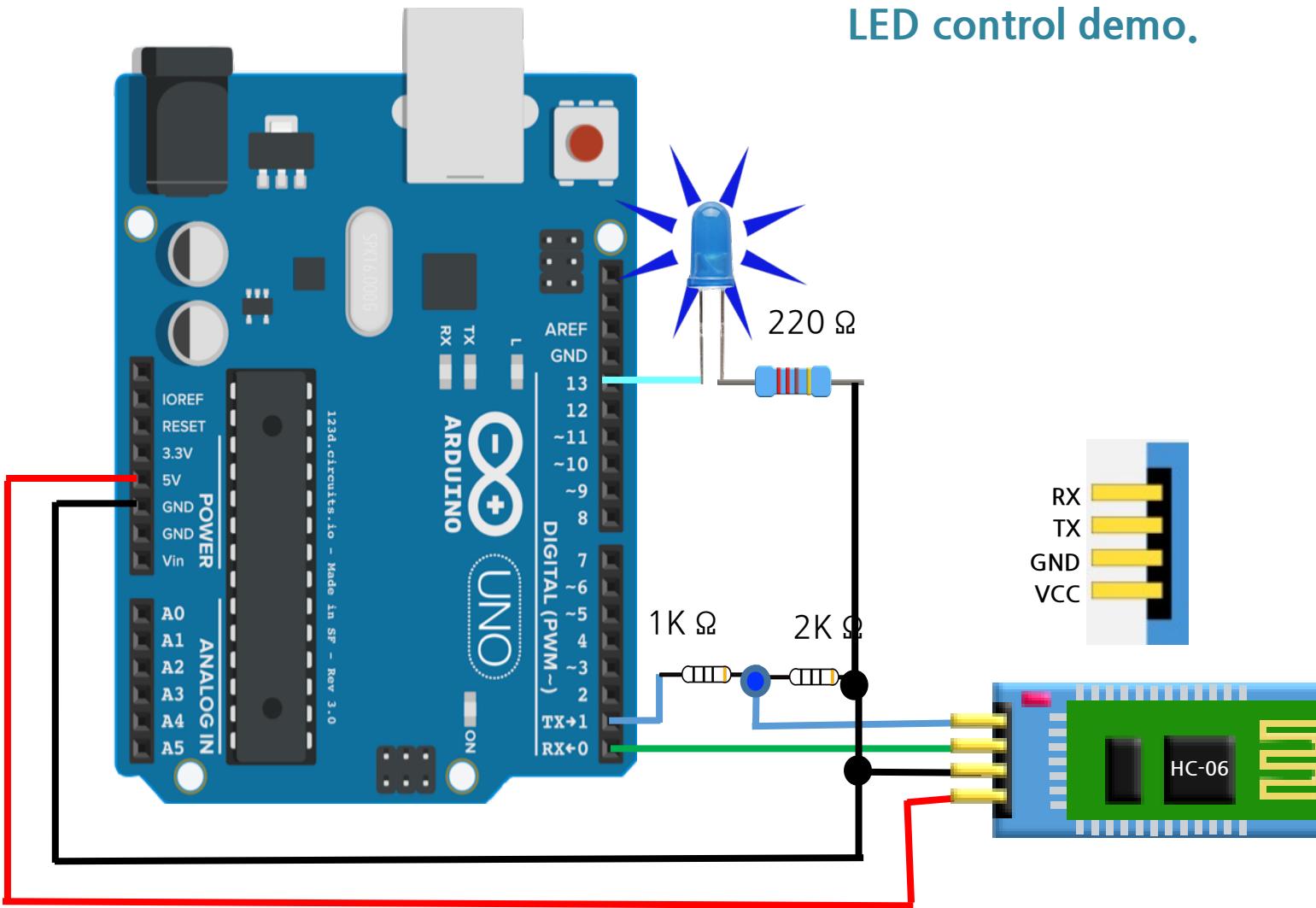
## Bluetooth Device PIN number



#### 4) Bluetooth LED control with given app

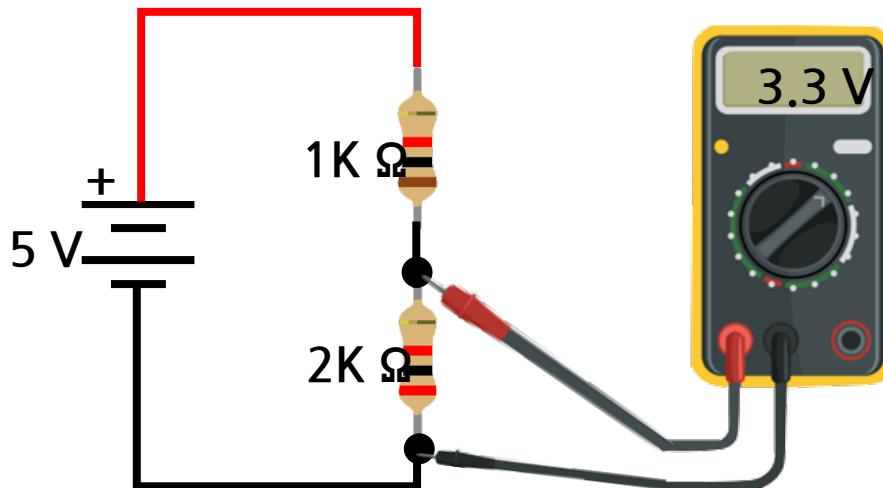


#### 4) Bluetooth LED control with given app



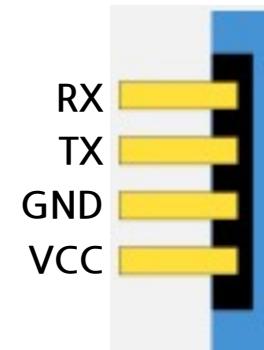
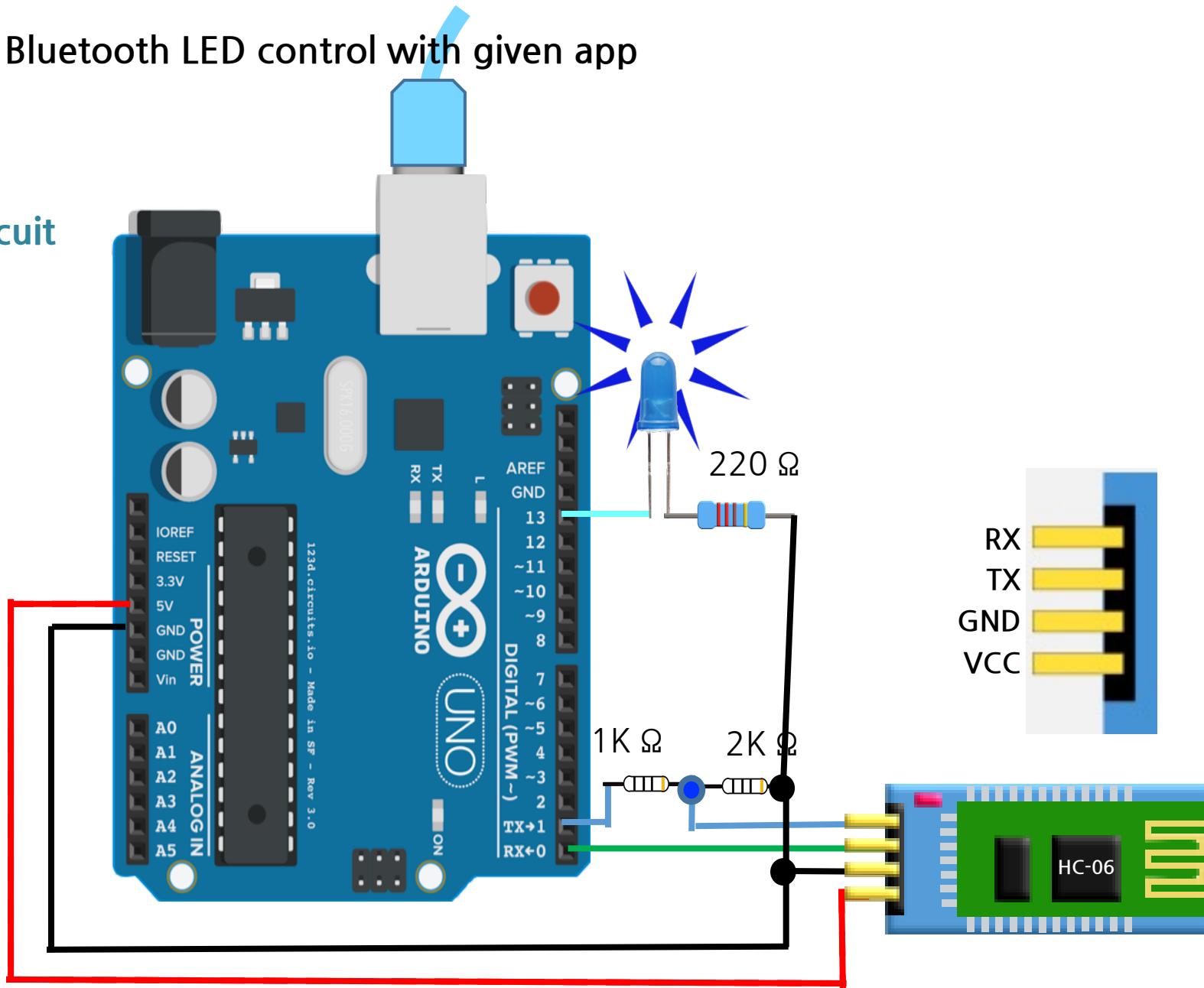
#### 4) Bluetooth LED control with given app

Voltage divider



#### 4) Bluetooth LED control with given app

Circuit



#### 4) Bluetooth LED control with given app



The screenshot shows the Arduino IDE interface with the title bar "Bluetooth\_LED\_App". The code area contains the following sketch:

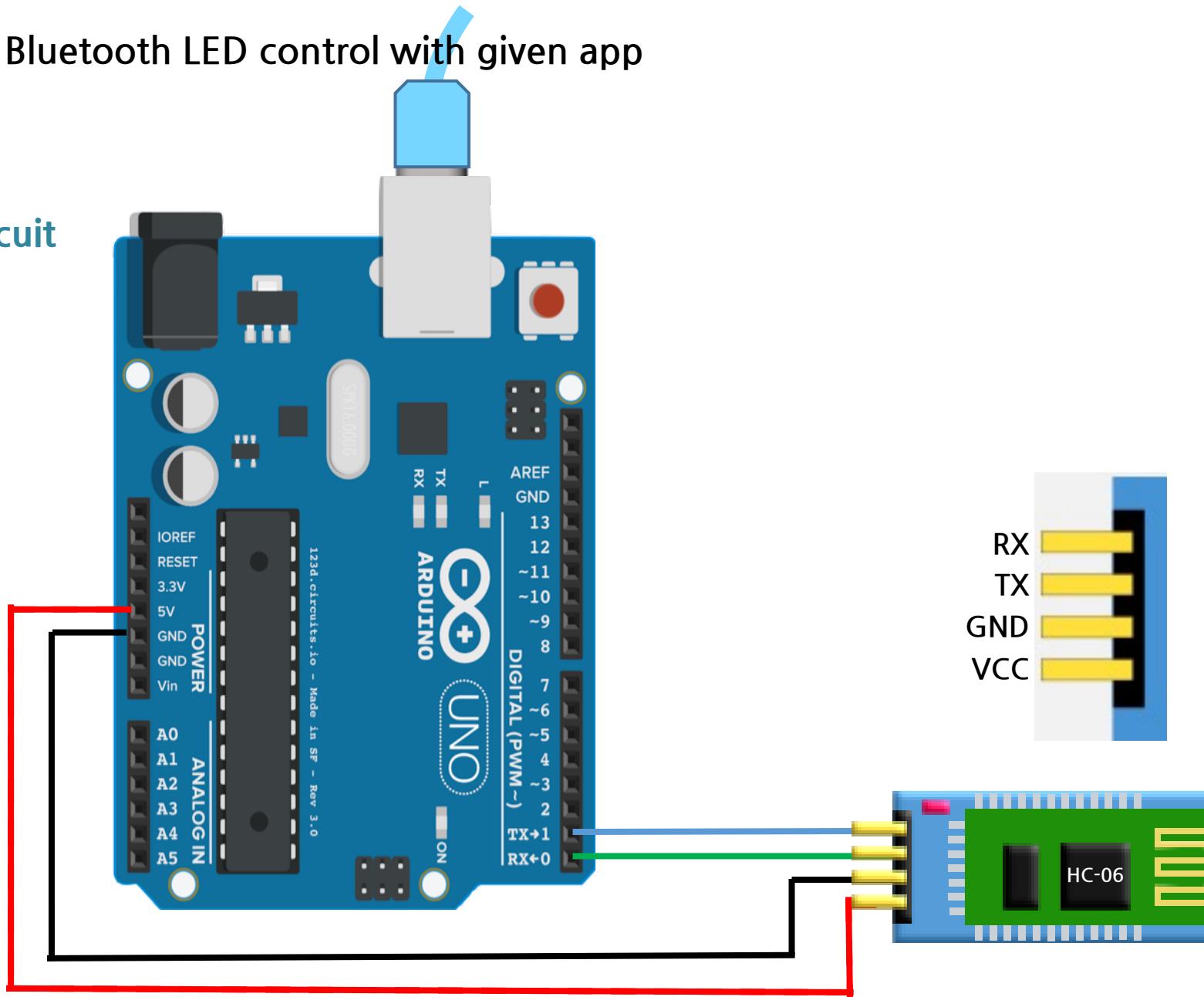
```
1 String message; // Store incoming command data
2 int led =13;
3
4 void setup() {
5   Serial.begin(9600);
6   pinMode(led,OUTPUT);
7   digitalWrite(led,LOW);
8 }
9
10 void loop() {
11   while (Serial.available() ) { // while there is data
12     delay(50);
13     char c = Serial.read();
14     message = message + c;    //Command such as "on"
15   }
16
17   if (message.length() > 0) {
18     Serial.println(message) ;
19     if (message == "ON") {
20       digitalWrite(13, HIGH) ;
21     }
22     if (message == "OFF") {
23       digitalWrite(13, LOW) ;
24     }
25   }
26   message = "" ; // clear the memory
27 }
28 }
```

#### 4) Bluetooth LED control with given app

스마트폰에 앱 설치    **LED control APP**

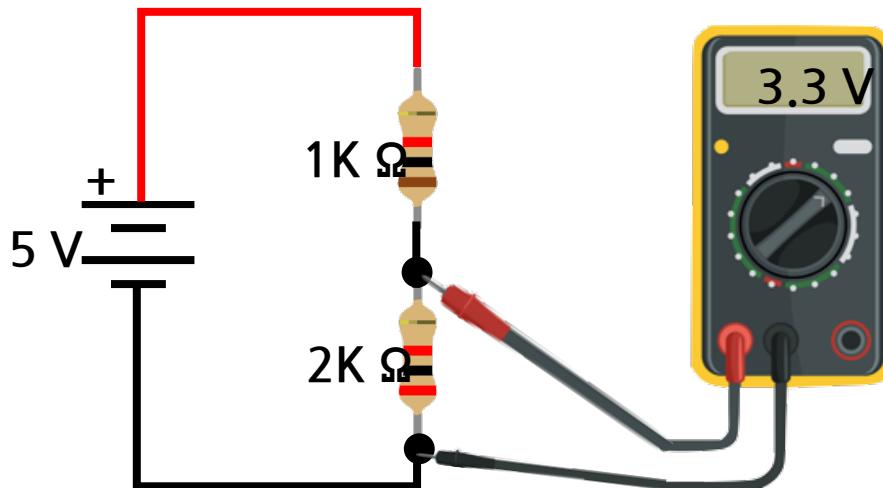
#### 4) Bluetooth LED control with given app

Circuit



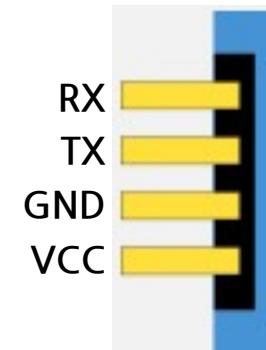
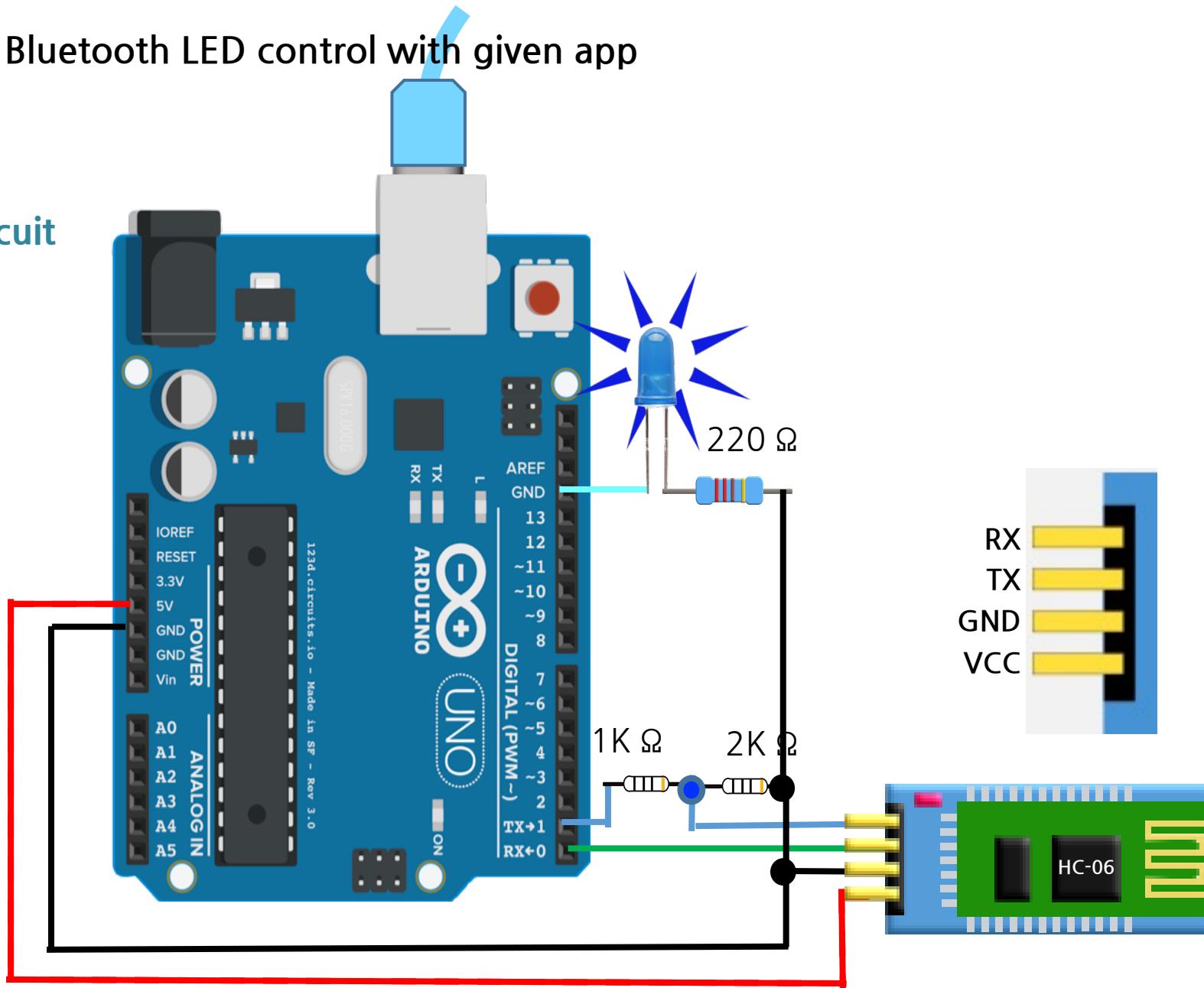
#### 4) Bluetooth LED control with given app

Voltage divider



#### 4) Bluetooth LED control with given app

Circuit

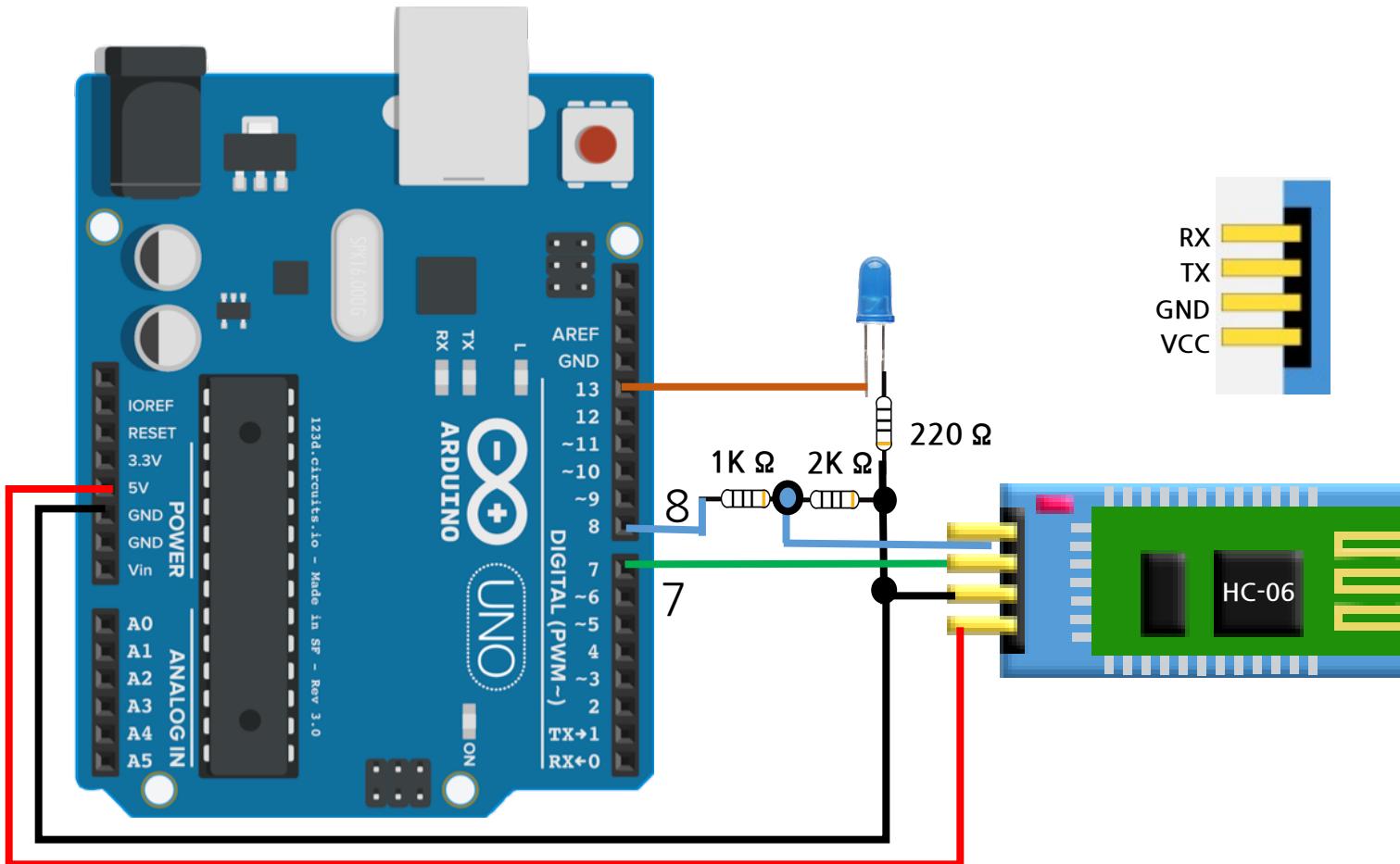


The Arduino logo is a white speech bubble containing a teal infinity symbol, set against a teal background.

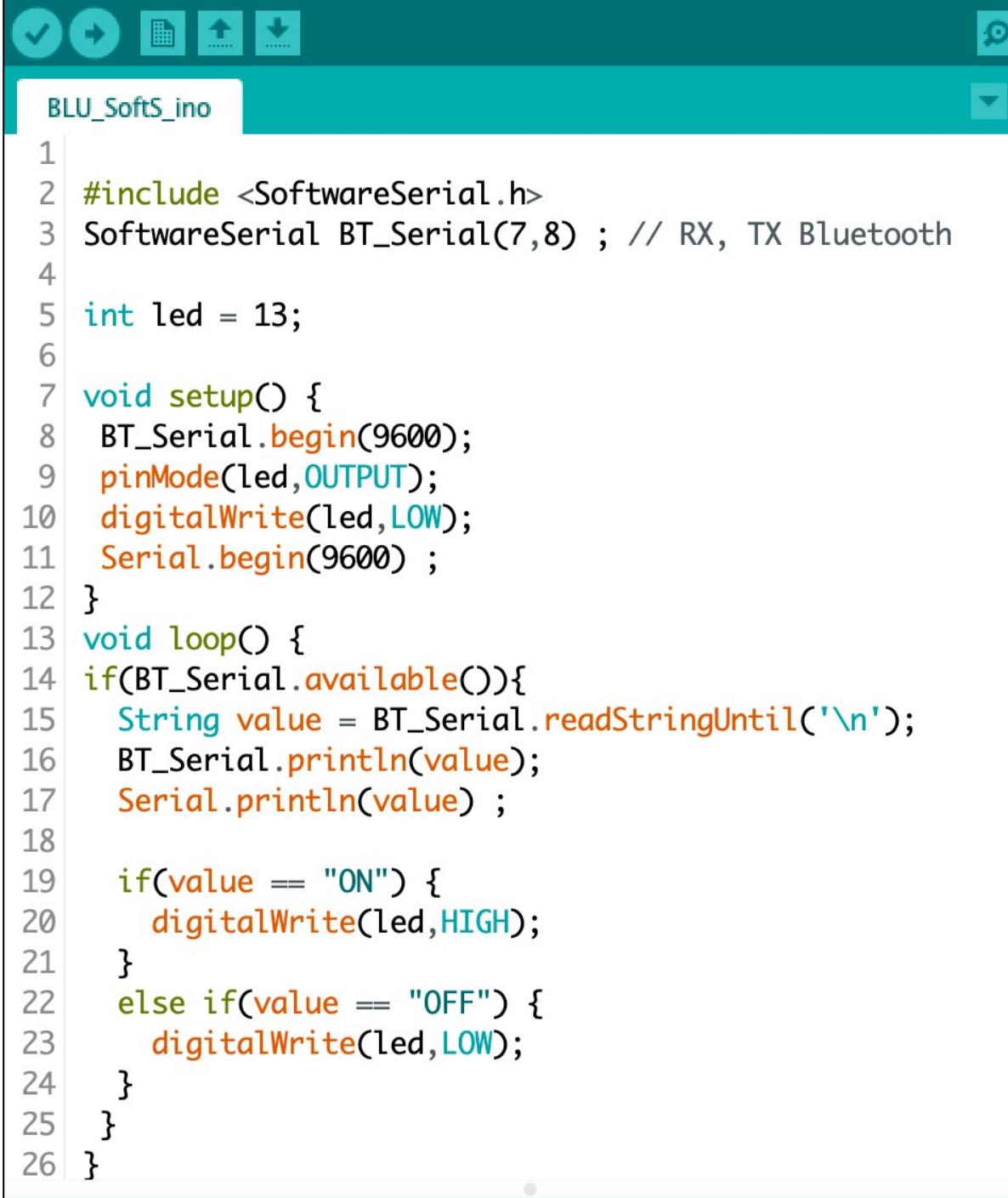
## 5) Bluetooth SoftwareSerial

## 5) Software Serial

Bluetooth SoftwareSerial circuit



## 5) Software Serial



The image shows the Arduino IDE interface with a sketch titled "BLU\_SoftS.ino". The code uses SoftwareSerial to handle Bluetooth communication and controls a LED on pin 13 based on received commands.

```
1 #include <SoftwareSerial.h>
2 SoftwareSerial BT_Serial(7,8) ; // RX, TX Bluetooth
3
4 int led = 13;
5
6 void setup() {
7     BT_Serial.begin(9600);
8     pinMode(led,OUTPUT);
9     digitalWrite(led,LOW);
10    Serial.begin(9600) ;
11 }
12
13 void loop() {
14     if(BT_Serial.available()){
15         String value = BT_Serial.readStringUntil('\n');
16         BT_Serial.println(value);
17         Serial.println(value) ;
18
19         if(value == "ON") {
20             digitalWrite(led,HIGH);
21         }
22         else if(value == "OFF") {
23             digitalWrite(led,LOW);
24         }
25     }
26 }
```

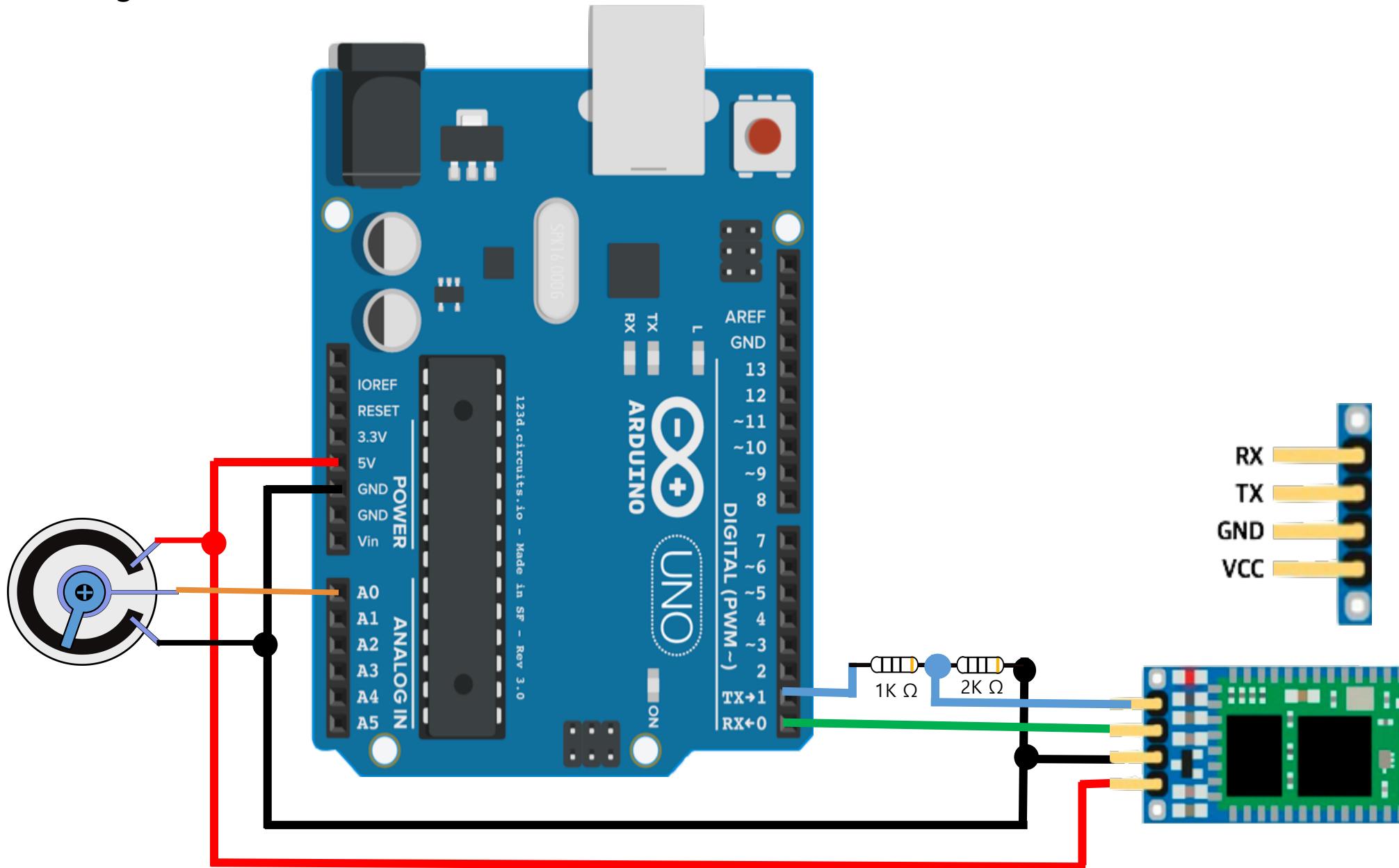


## 6) Bluetooth: Analog Data Read

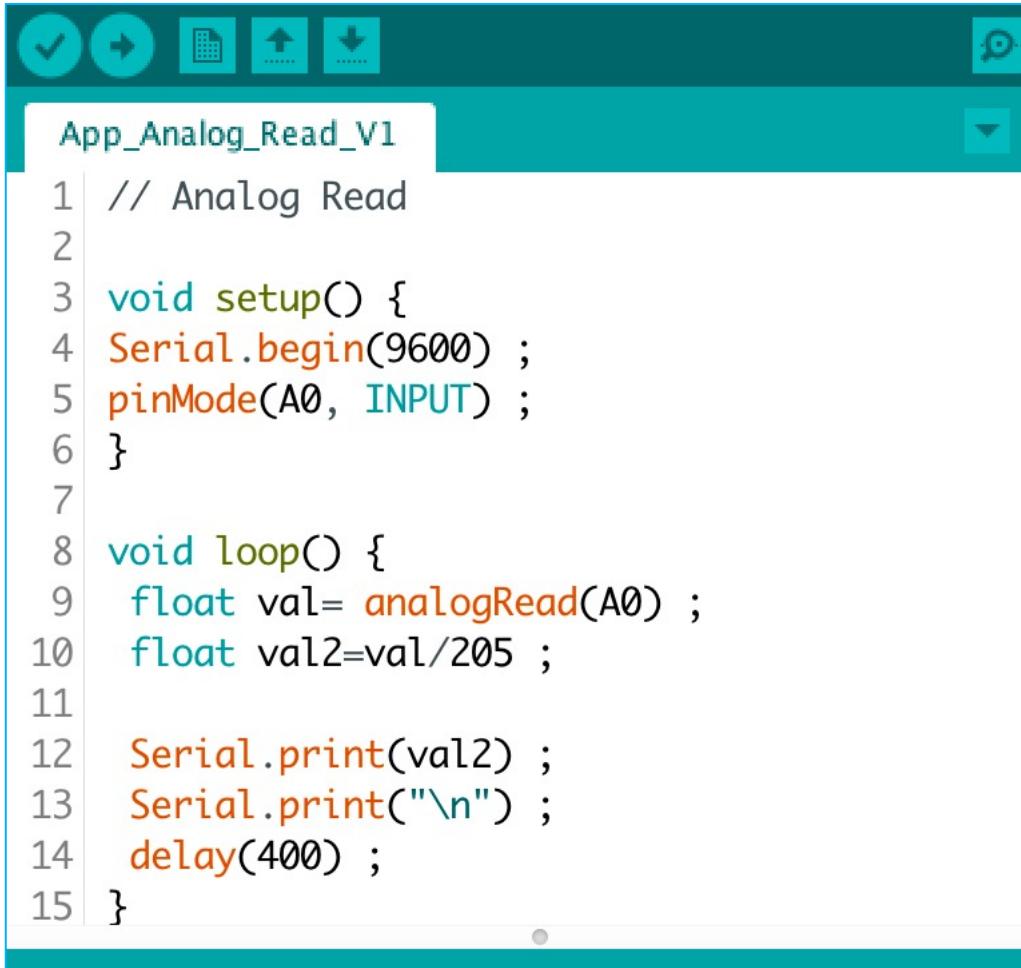
# Bluetooth Analog data read



## Bluetooth Analog data read



## Bluetooth Analog data read



The screenshot shows a microcontroller development environment with a teal-themed interface. At the top, there are several icons: a checkmark, a circular arrow, a file folder, an upward arrow, a downward arrow, and a magnifying glass. The title bar displays the project name "App\_Analog\_Read\_V1". The main area contains the following C++ code:

```
1 // Analog Read
2
3 void setup() {
4     Serial.begin(9600) ;
5     pinMode(A0, INPUT) ;
6 }
7
8 void loop() {
9     float val= analogRead(A0) ;
10    float val2=val/205 ;
11
12    Serial.print(val2) ;
13    Serial.print("\n") ;
14    delay(400) ;
15 }
```

## 6) Bluetooth Analog data read

Bluetooth Analog APP

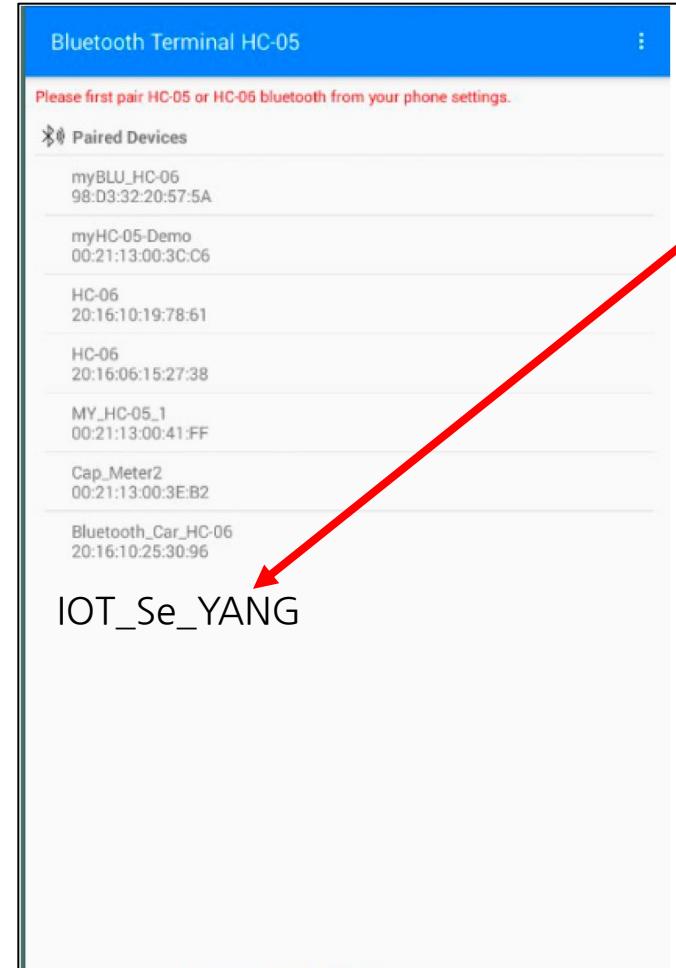
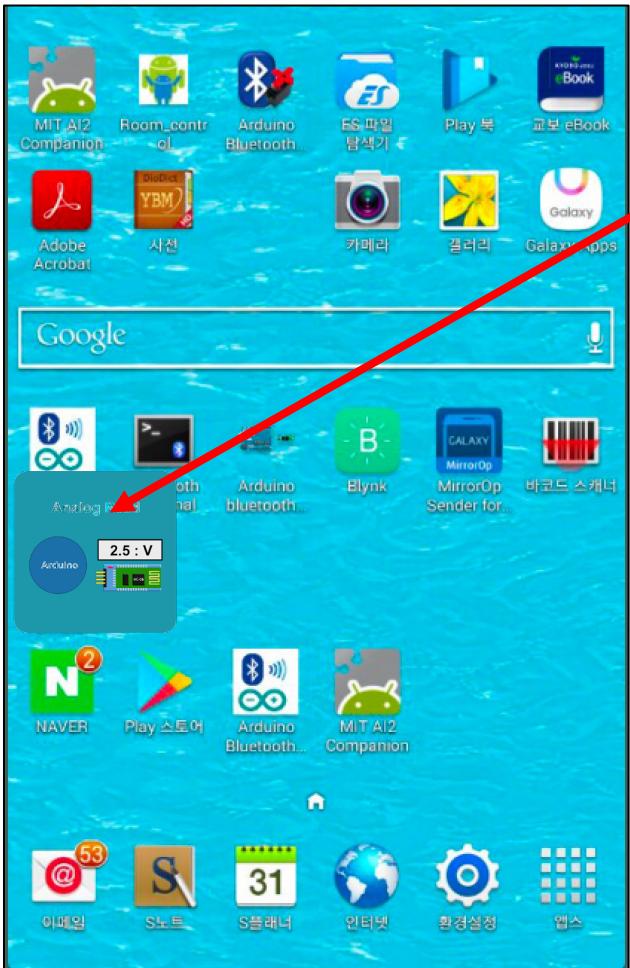
Smartphone



PC

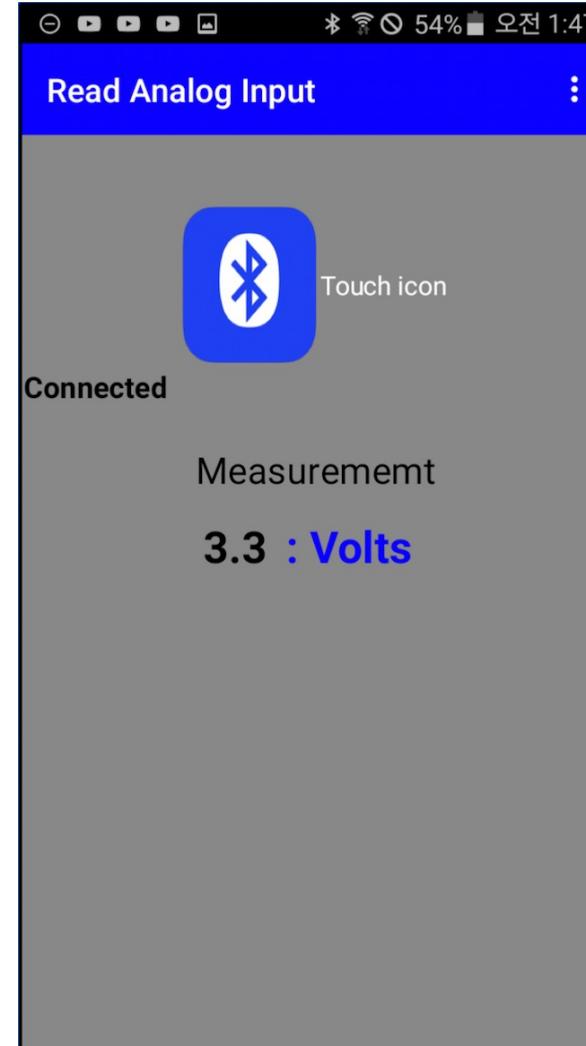
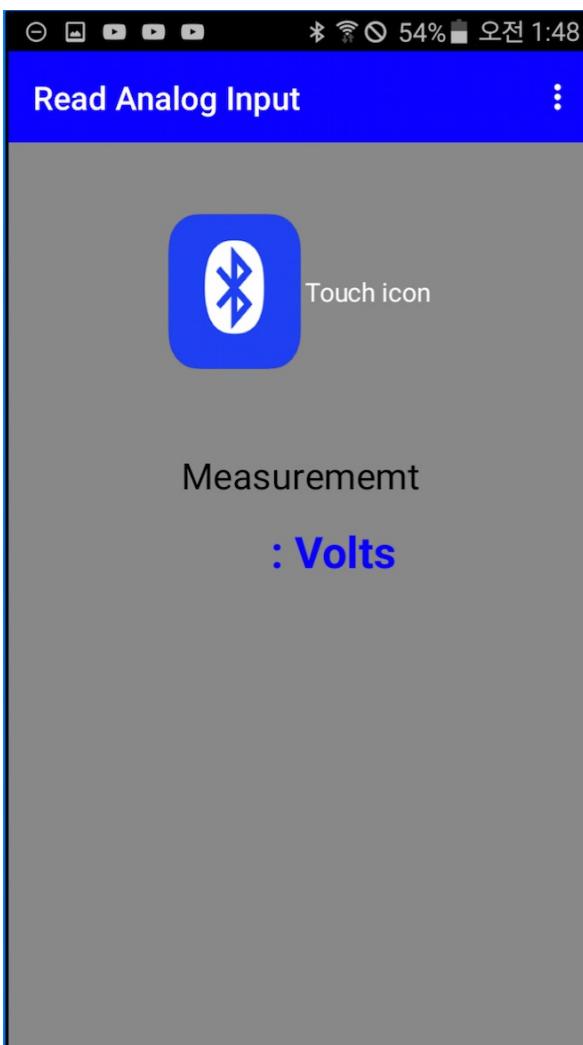
## 6) Bluetooth Analog data read

### Selection



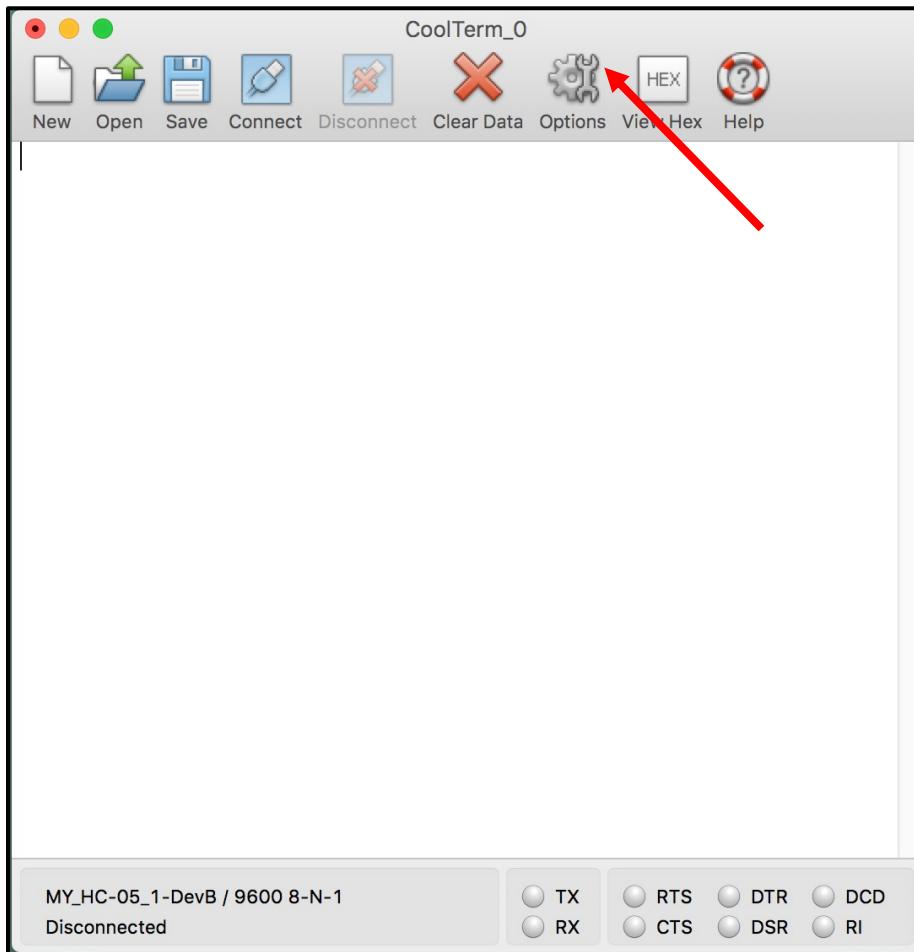
## 6) Bluetooth Analog data read

Analog read value

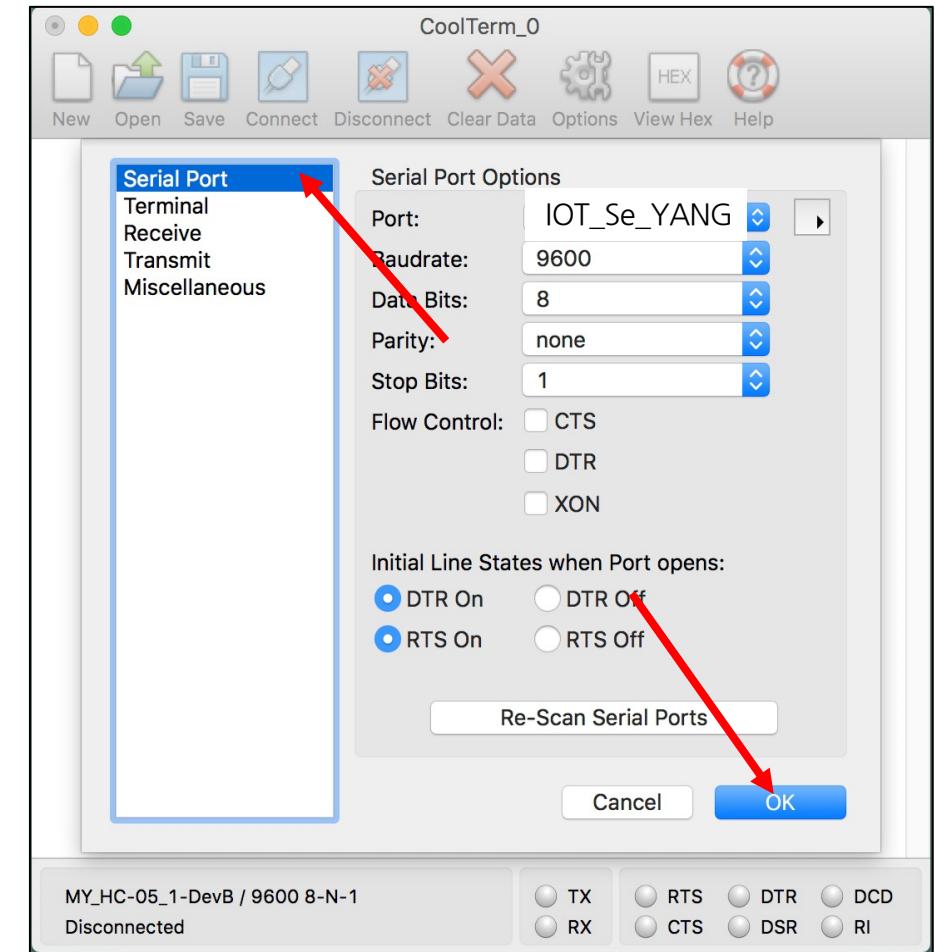


## 6) Bluetooth Analog data read

### Read Analog value from Coolterm

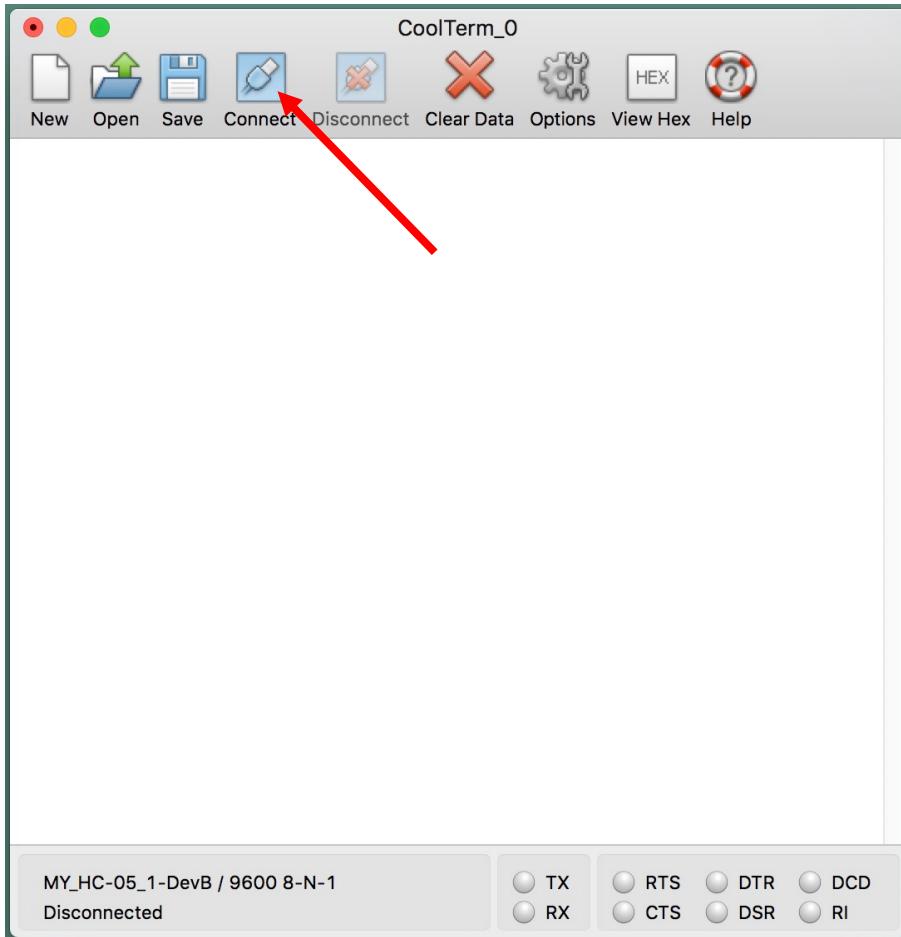


### Analog Read from PC



## 6) Bluetooth Analog data read

### Coolterm connect

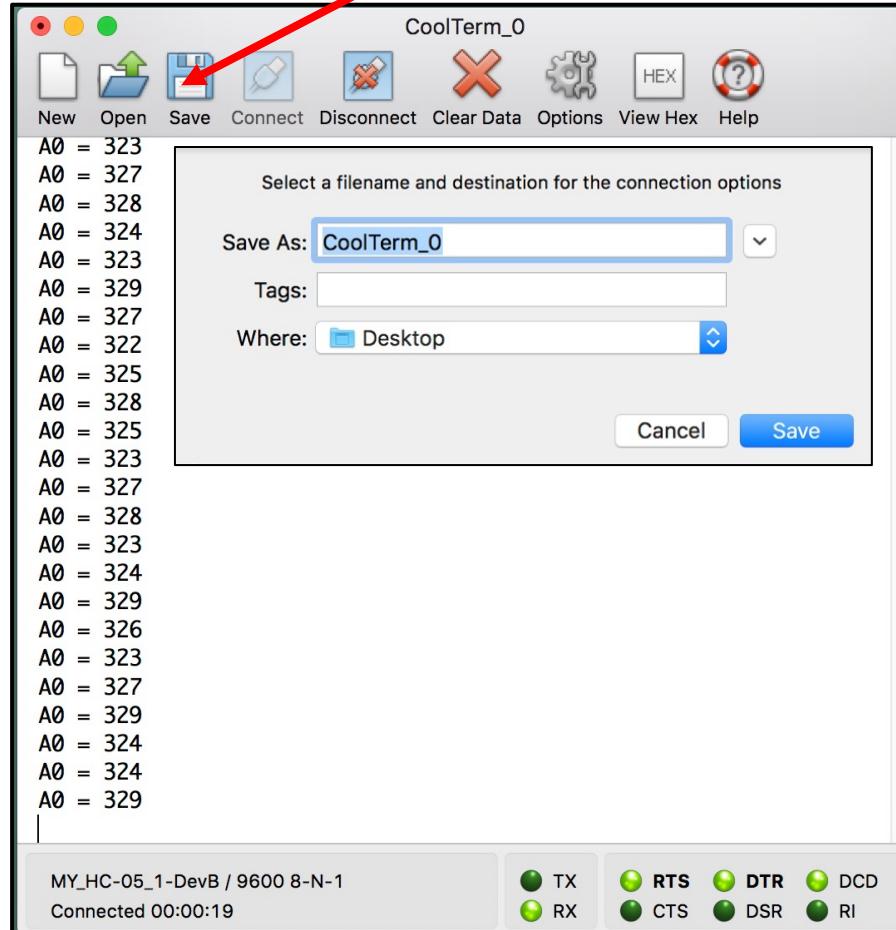


The screenshot shows the CoolTerm software interface after connection. The title bar reads "CoolTerm\_0". The menu bar and toolbar are identical to the first screenshot. The main window displays a series of analog data readings: "A0 = 323", "A0 = 327", "A0 = 328", "A0 = 324", "A0 = 323", "A0 = 329", "A0 = 327", "A0 = 322", "A0 = 325", "A0 = 328", "A0 = 325", "A0 = 323", "A0 = 327", "A0 = 328", "A0 = 323", "A0 = 324", "A0 = 329", "A0 = 326", "A0 = 323", "A0 = 327", "A0 = 329", "A0 = 324", "A0 = 324", "A0 = 329". The status bar at the bottom shows "Connected 00:00:19". The legend at the bottom right indicates that green dots represent TX, RX, RTS, CTS, DTR, SRS, DCD, and RI.

## 6) Bluetooth Analog data read

Analog value save

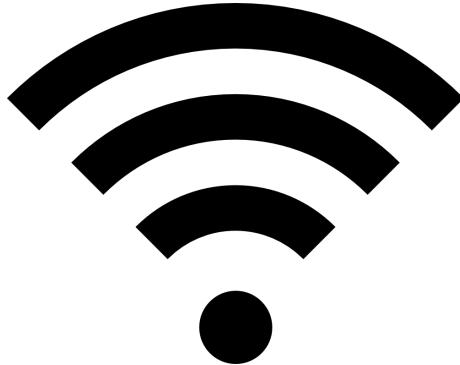
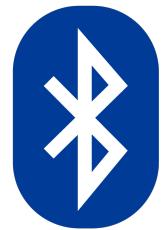
Bluetooth data saving to PC



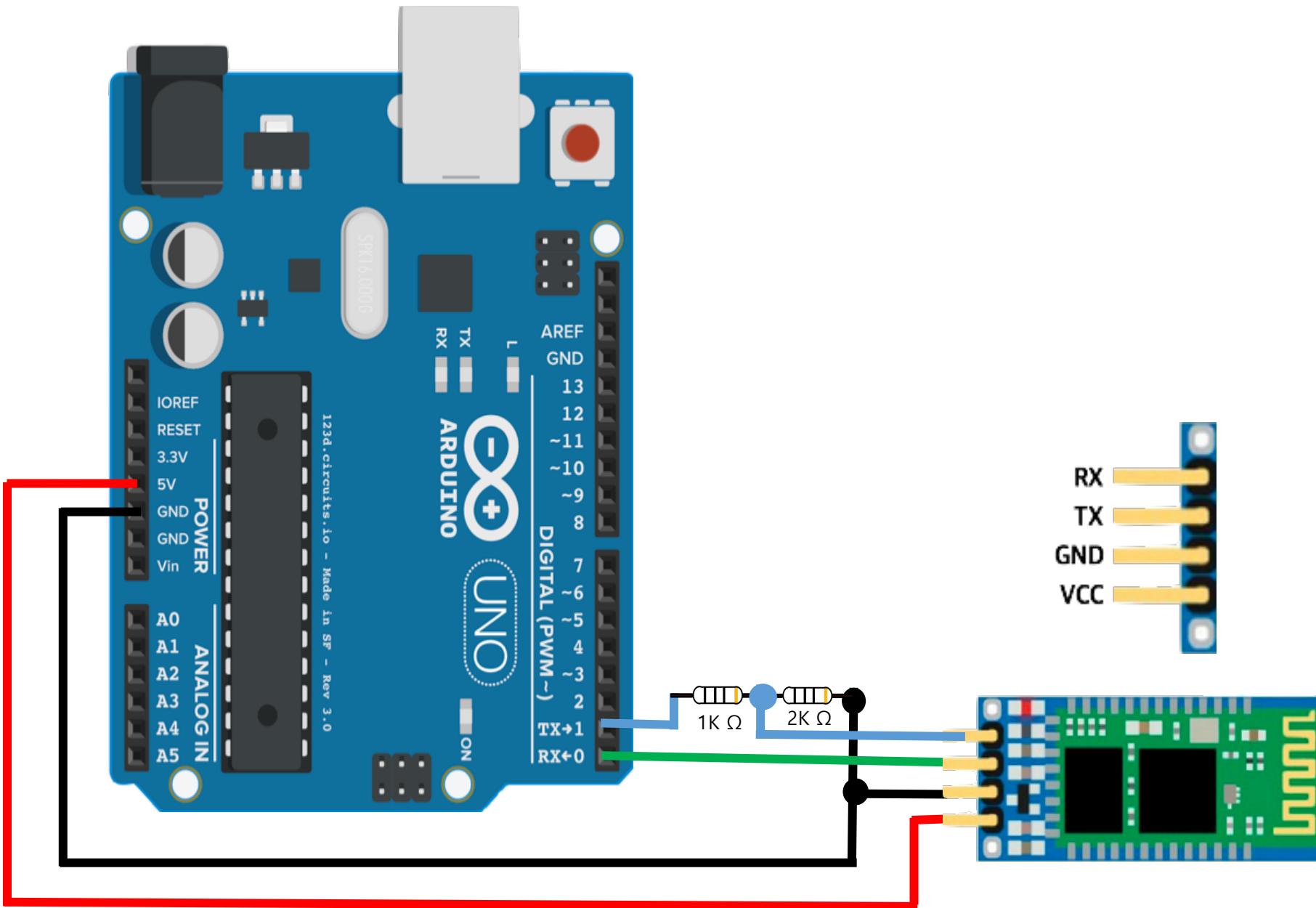


## 7) Bluetooth: Servo motor control

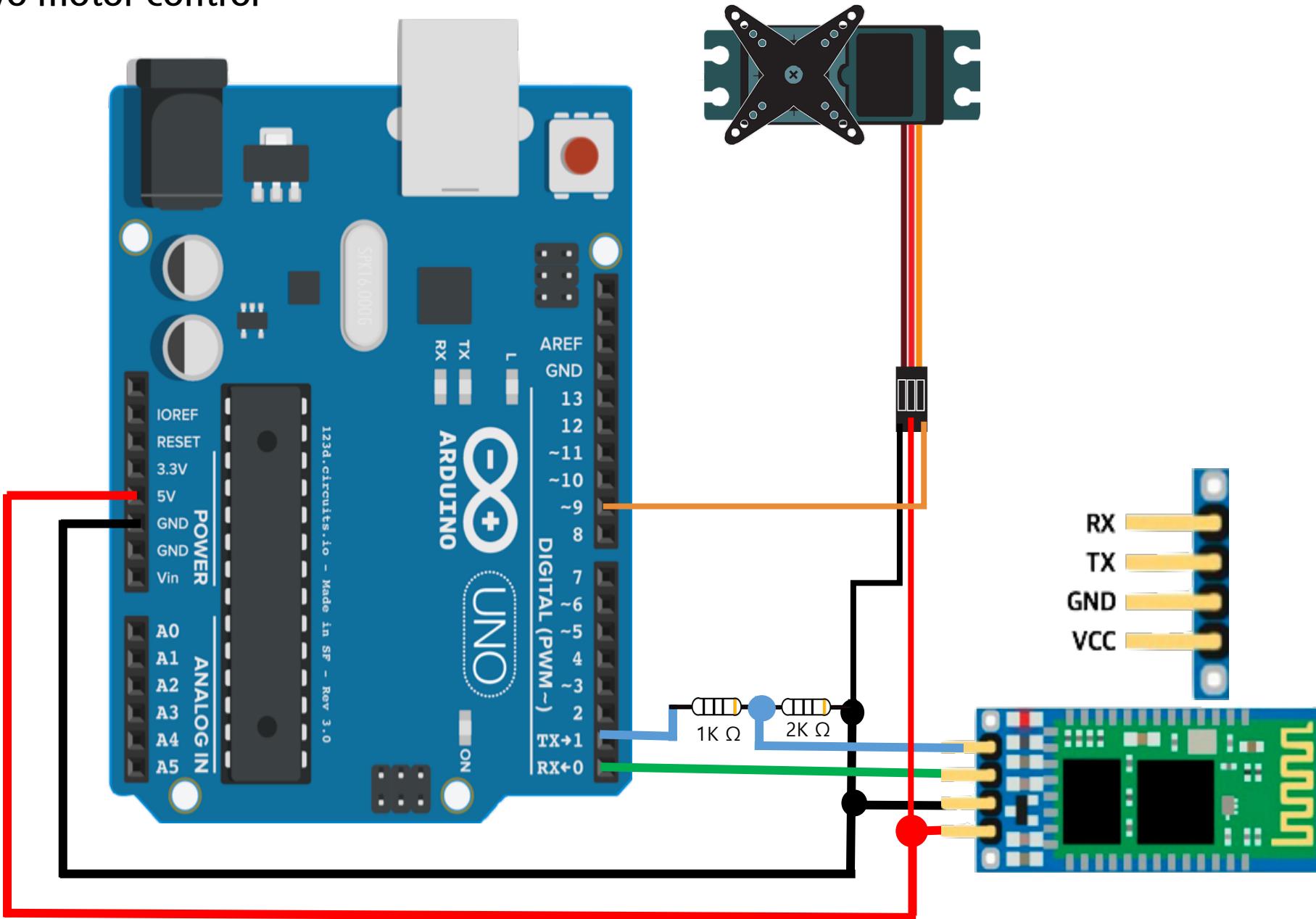
## Bluetooth Servo motor control



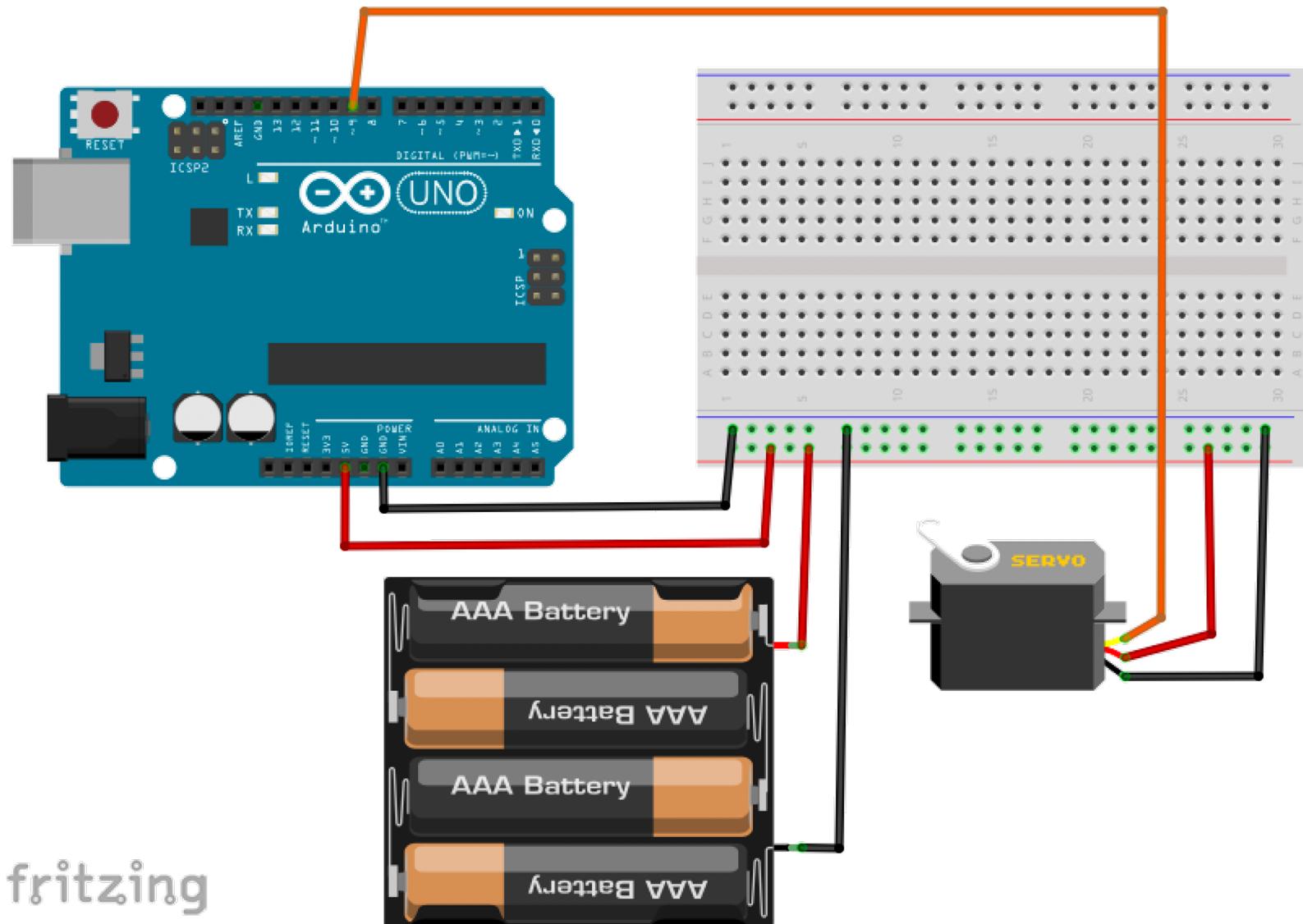
## Bluetooth Servo motor control



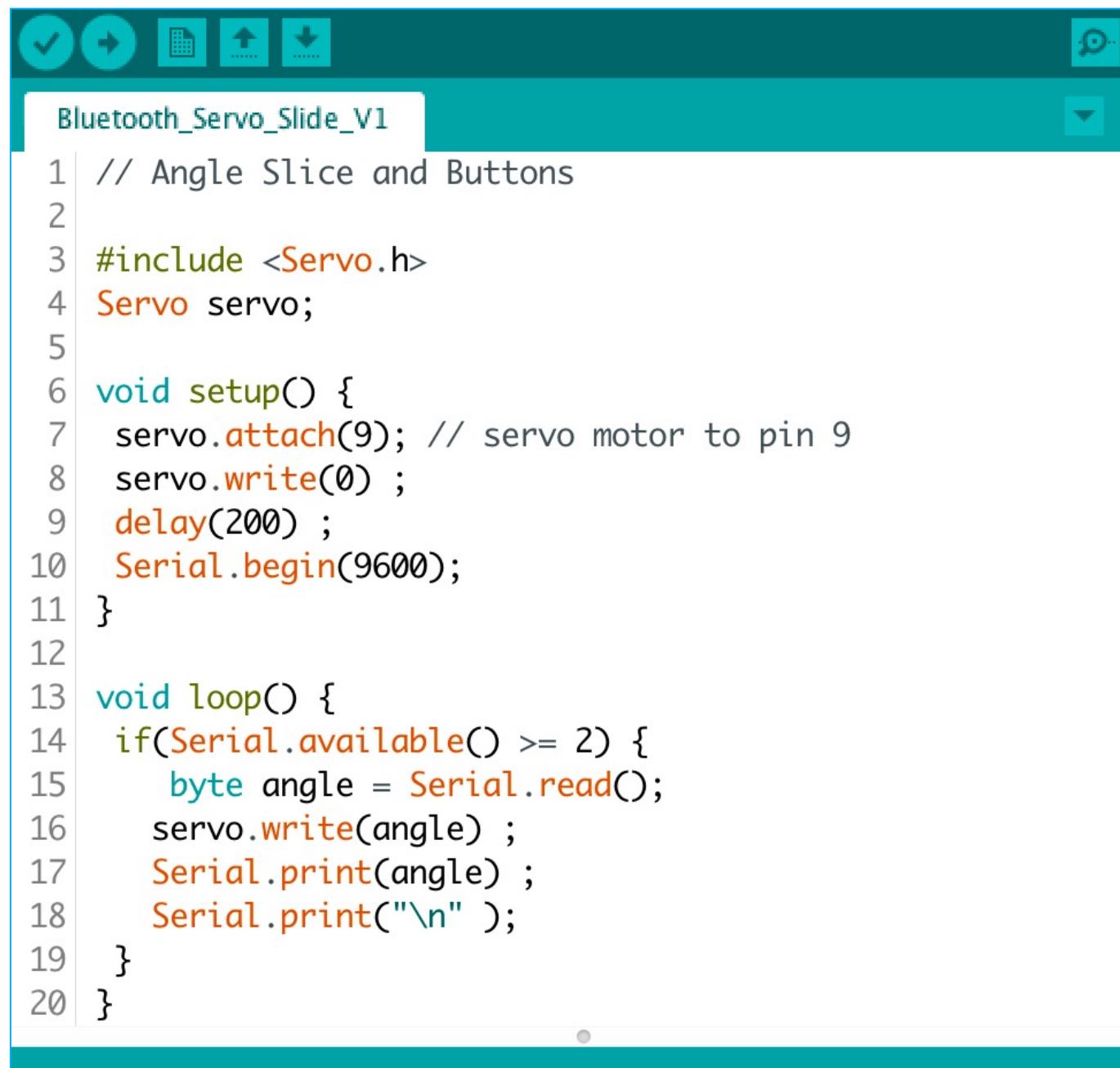
## Bluetooth Servo motor control



## 7) Bluetooth Servo motor control



## Bluetooth Servo motor control



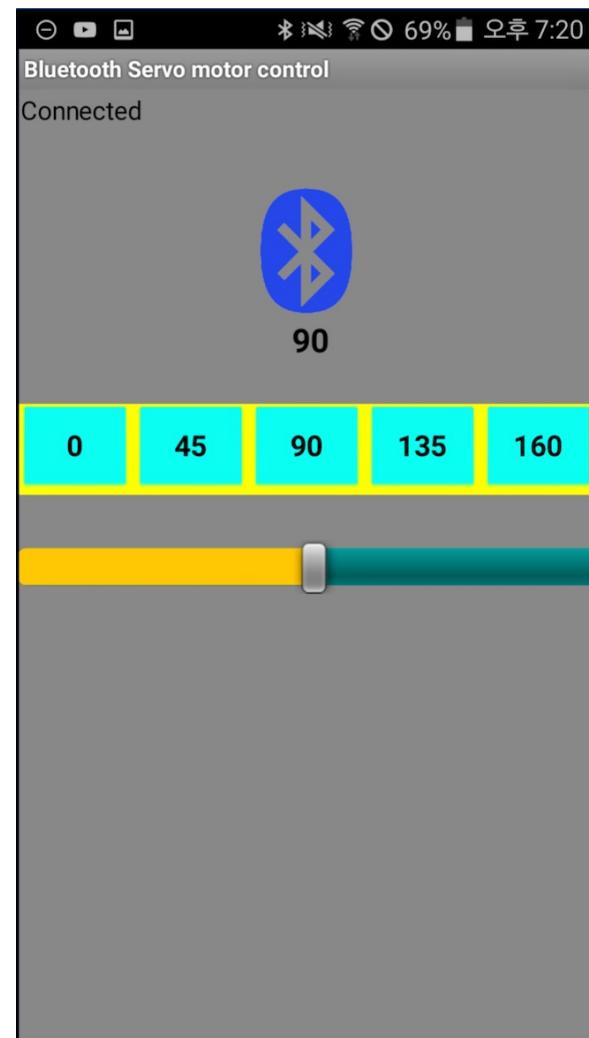
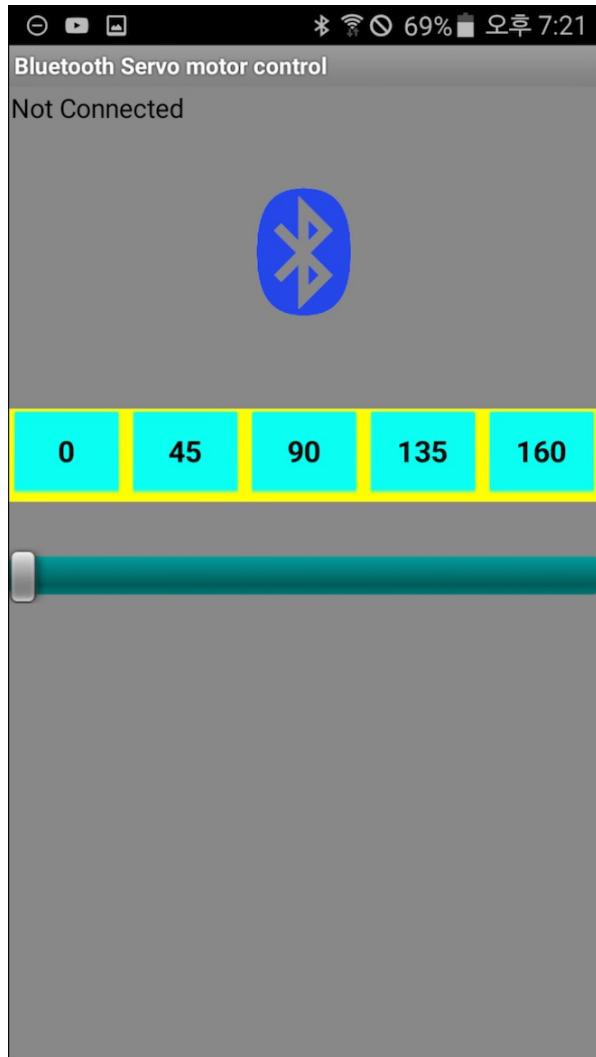
The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** The title "Bluetooth\_Servo\_Slide\_V1" is displayed in the center of the header bar.
- Tool Buttons:** Standard Arduino IDE tool icons are visible in the top-left corner.
- Code Area:** The main area contains the following C++ code for controlling a servo motor via Bluetooth serial communication:

```
1 // Angle Slice and Buttons
2
3 #include <Servo.h>
4 Servo servo;
5
6 void setup() {
7   servo.attach(9); // servo motor to pin 9
8   servo.write(0) ;
9   delay(200) ;
10  Serial.begin(9600);
11 }
12
13 void loop() {
14   if(Serial.available() >= 2) {
15     byte angle = Serial.read();
16     servo.write(angle) ;
17     Serial.print(angle) ;
18     Serial.print("\n" );
19   }
20 }
```

## Bluetooth Servo motor control

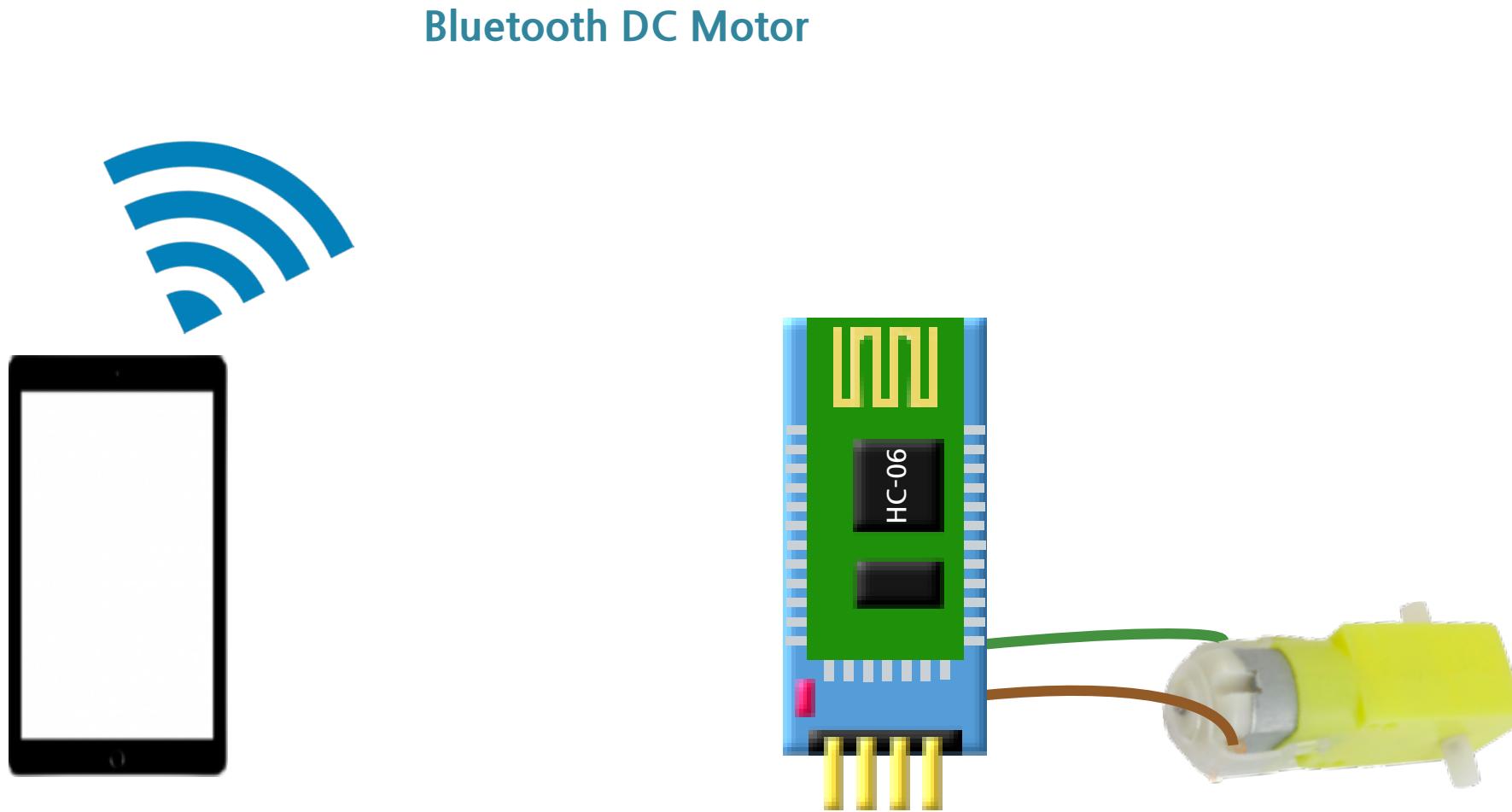
App to your phone





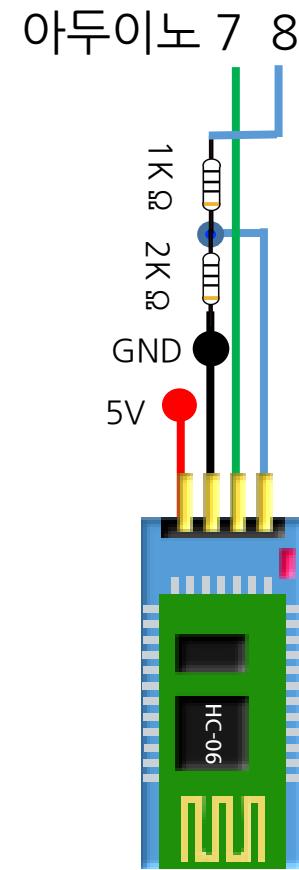
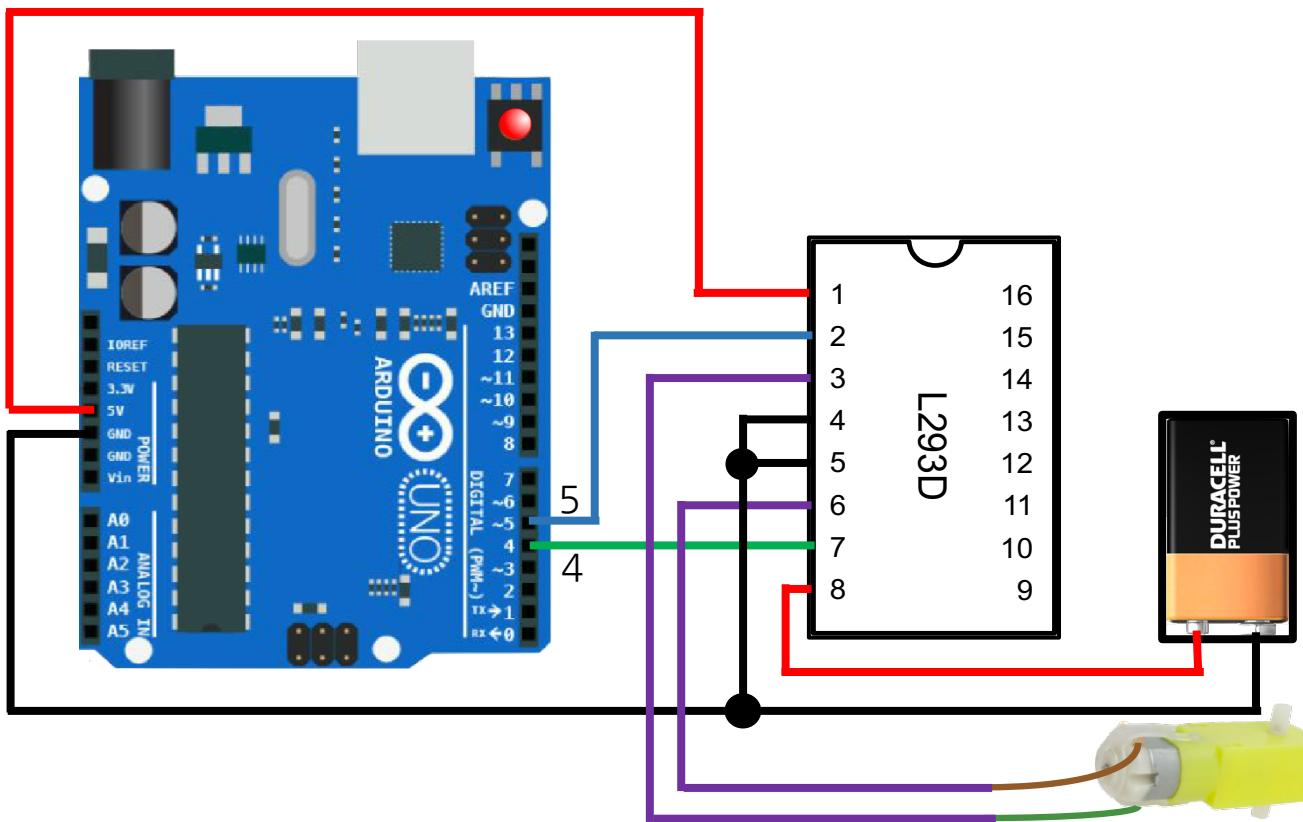
## 8) Bluetooth: DC motor control

## 8) Bluetooth DC motor control

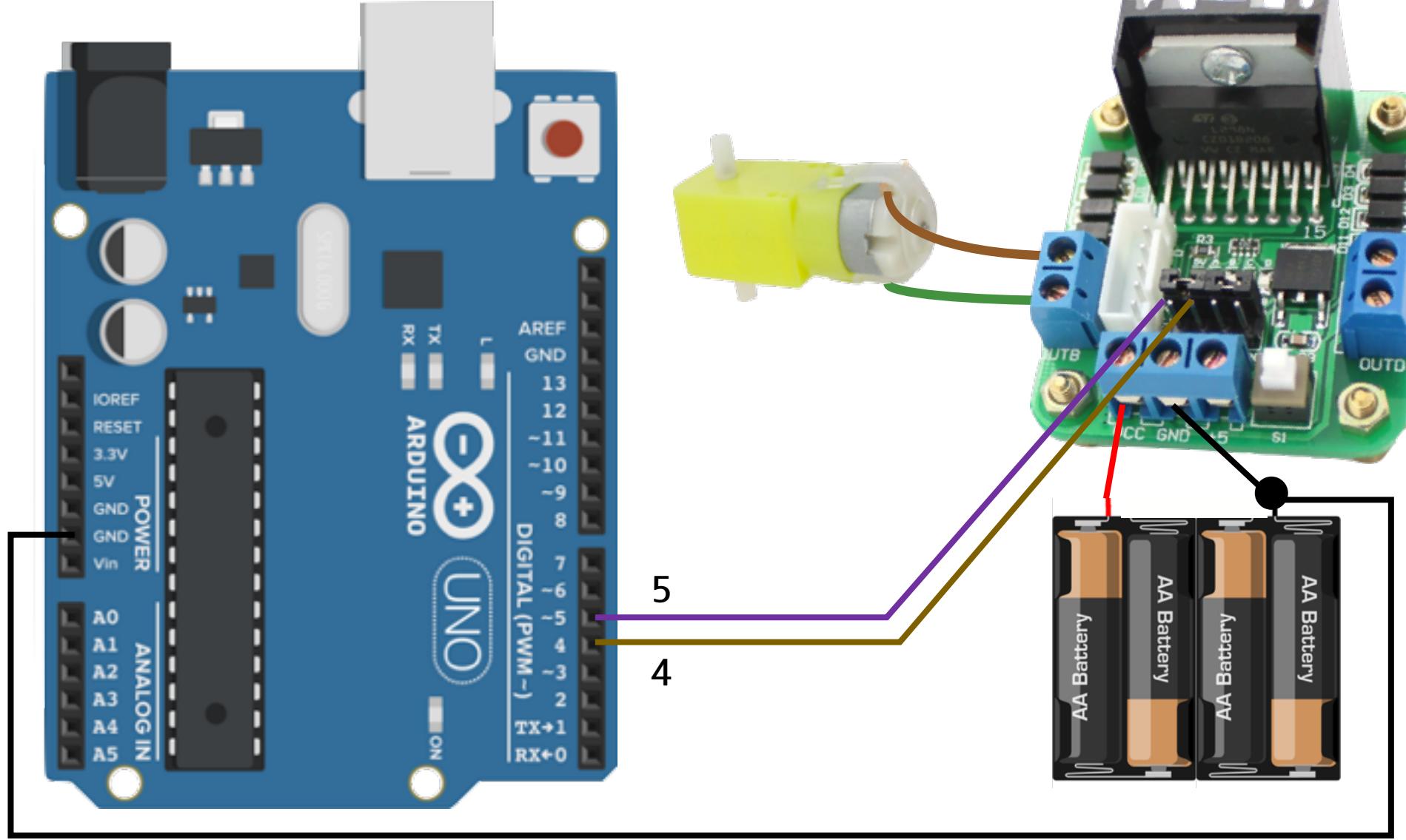


## 8) Bluetooth DC motor control

Circuit : DC motor direction control

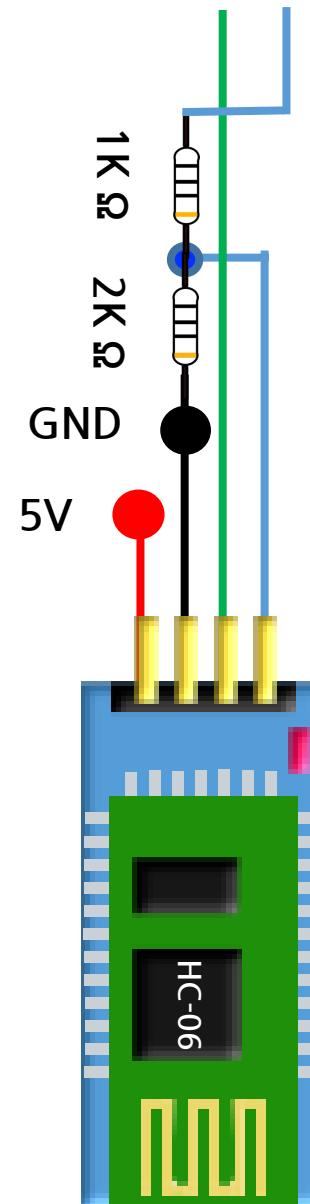
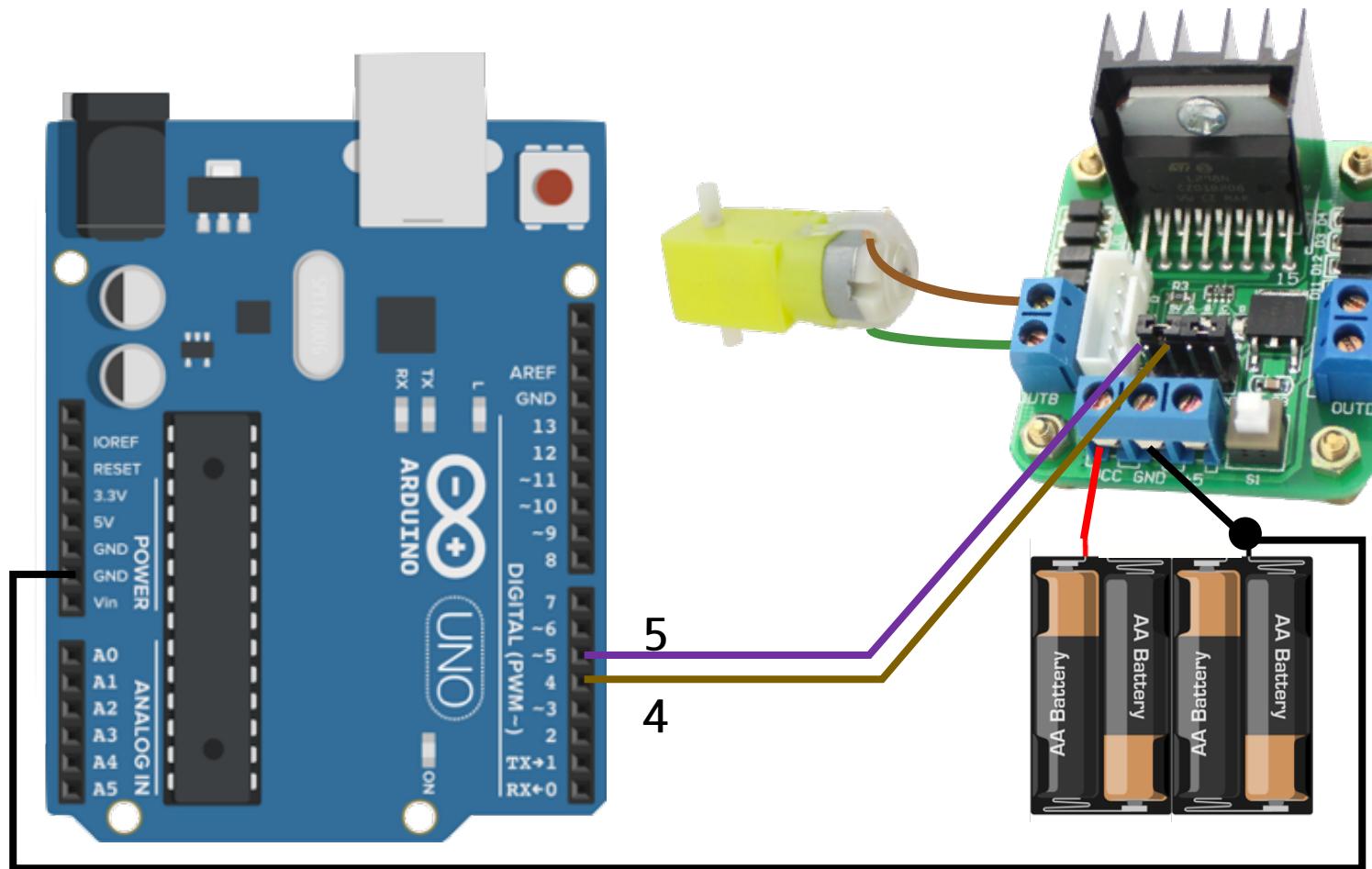


## L298N 모터 드라이버 회로



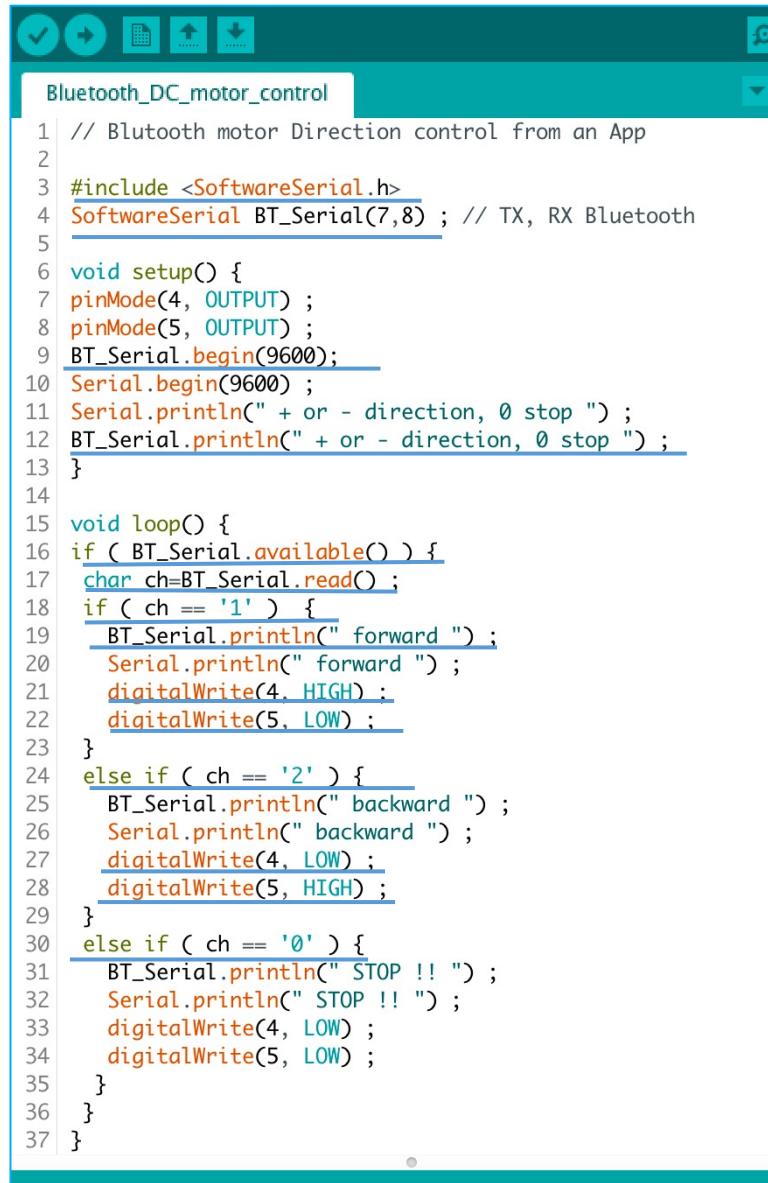
# 블루투스 L298N 모터 드라이버 회로

아두이노 7 8



## 8) Bluetooth DC motor control

### Bluetooth DC motor control Sketch



The screenshot shows the Arduino IDE interface with the title bar "Bluetooth\_DC\_motor\_control". The code editor contains the following C++ code:

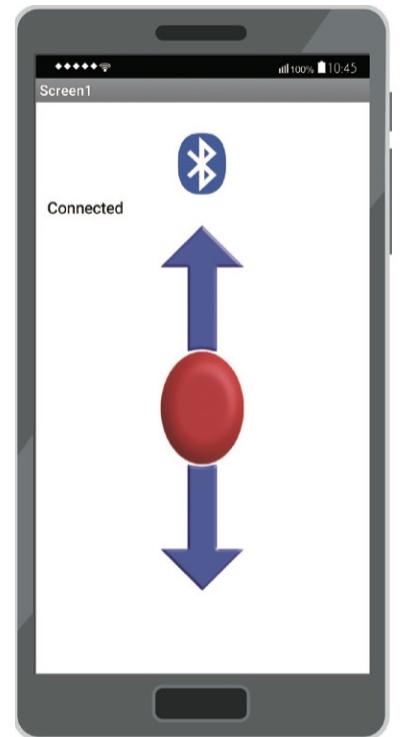
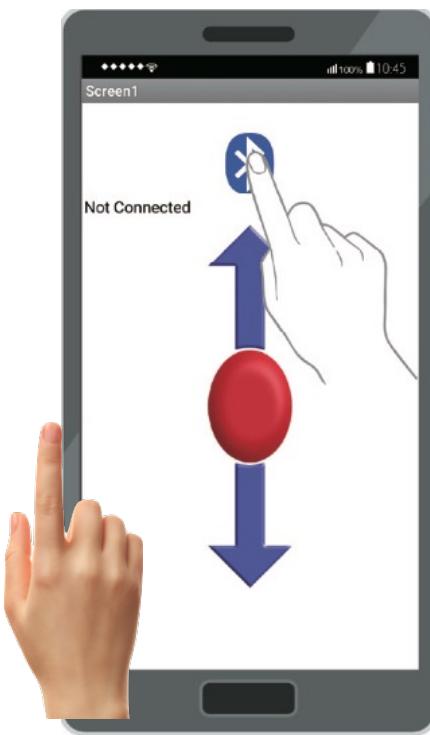
```
// Bluetooth motor Direction control from an App
#include <SoftwareSerial.h>
SoftwareSerial BT_Serial(7,8); // TX, RX Bluetooth

void setup() {
pinMode(4, OUTPUT);
pinMode(5, OUTPUT);
BT_Serial.begin(9600);
Serial.begin(9600);
Serial.println(" + or - direction, 0 stop ");
BT_Serial.println(" + or - direction, 0 stop ");
}

void loop() {
if ( BT_Serial.available() ) {
char ch=BT_Serial.read();
if ( ch == '1' ) {
BT_Serial.println(" forward ");
Serial.println(" forward ");
digitalWrite(4, HIGH);
digitalWrite(5, LOW);
}
else if ( ch == '2' ) {
BT_Serial.println(" backward ");
Serial.println(" backward ");
digitalWrite(4, LOW);
digitalWrite(5, HIGH);
}
else if ( ch == '0' ) {
BT_Serial.println(" STOP !! ");
Serial.println(" STOP !! ");
digitalWrite(4, LOW);
digitalWrite(5, LOW);
}
}
}
```

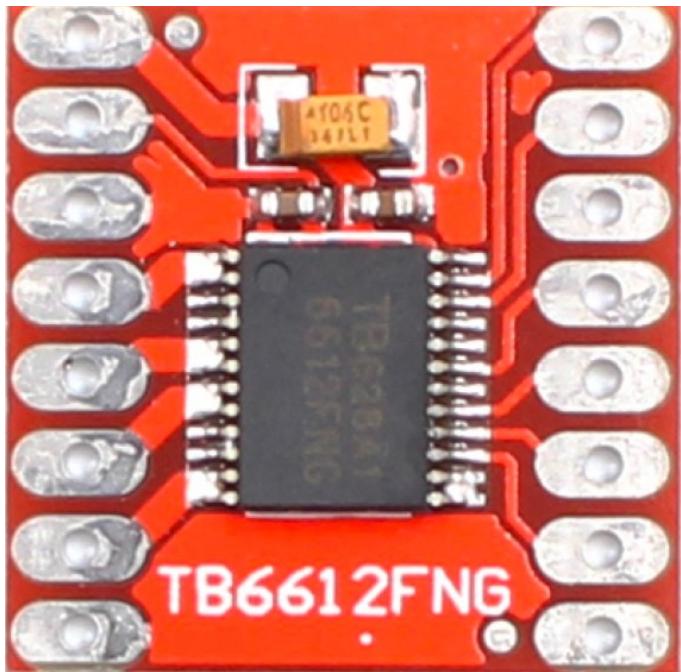
## 8) Bluetooth DC motor control

### App install and control

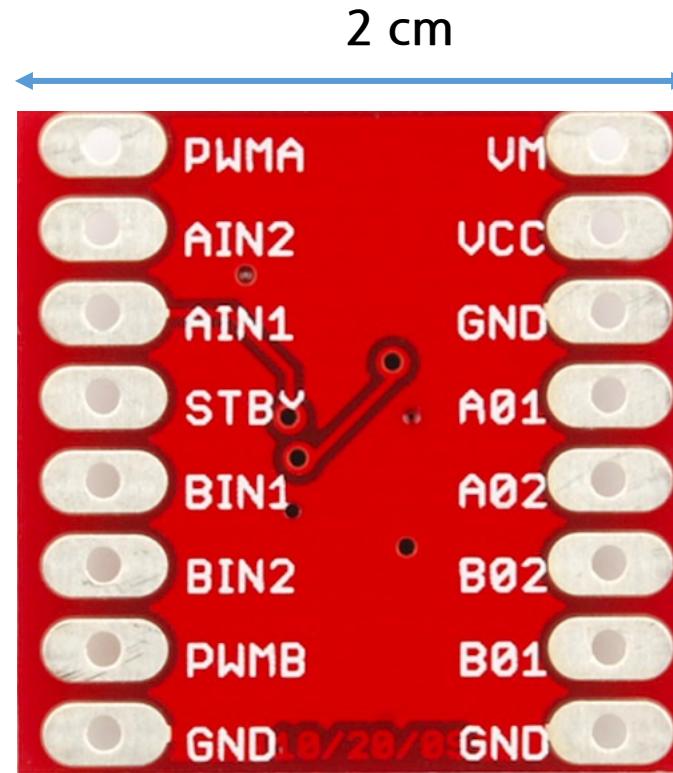


Project : TB6612 모터 드라이버 사용 블루투스 모터 컨트롤 회로와 코딩 작성

## TB6612FNG 모터 드라이버



앞면



뒷면