

✓ 아두이노기반 블루투스 통신 및 제어

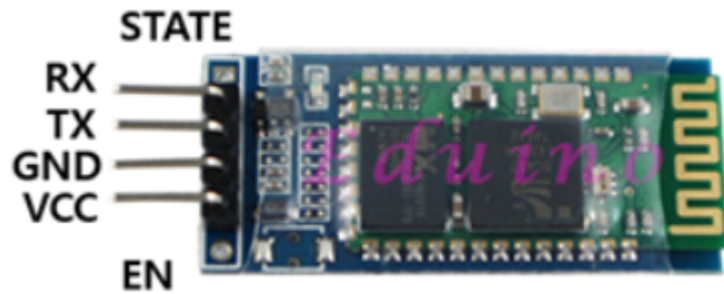
✓ 준비물

- 아두이노 보드
 - 블루투스(HC-06)
 - 아두이노 IDE
-

✓ 특징

- 디지털 통신 기기를 위한 개인 근거리 무선 통신 산업 표준
 - (Personal Area Network)의 산업 표준중 하나인 블루투스는 ISM(Industrial Scientific and Medical) 대역에 포함되는 2.4~2.485GHz의 단파 UHF 전파를 이용하여 전자 장비 간의 짧은 거리의 데이터 통신 방식을 규정
 - 블루투스는 10m 정도의 근거리 범위에서 사용 가능(그 이상은 보장할 수 없음)
 - 저전력을 강조한 통신
 - 비교적 낮은 속도로 디지털 정보를 무선 통신을 통해 주고 받는 용도로 채용되고 있음
-

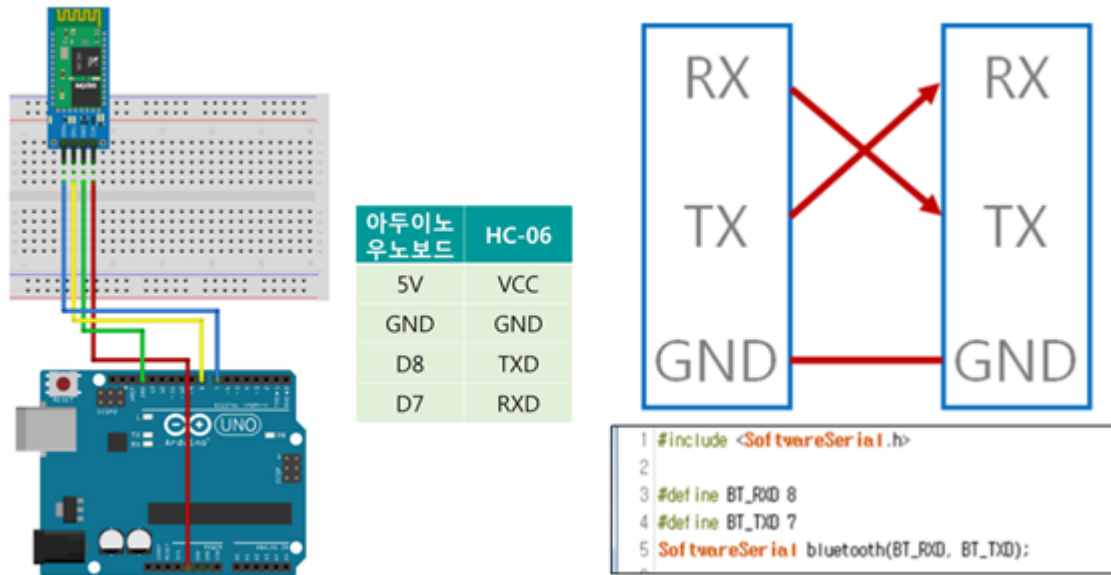
✓ 블루투스 HC-06



- HC-06 - 아두이노에서 시리얼통신을 이용하여 데이터 값을 주고 받을 수 있는 모듈이며, 10미터 정도에서 무선으로 데이터를 보내거나 받을 수 있음
-

✓ 아두이노 블루투스 모듈(HC-06)로 무선 통신 하기

✓ 1) 아두이노와 HC-06를 연결



✓ 2) 코딩

```

1 #Bluetooth_1
2 #include <SoftwareSerial.h>
3 #define BT_RXD 8
4 #define BT_TXD 7
5 SoftwareSerial bluetooth(BT_RXD, BT_TXD);
6
7 void setup(){
8   Serial.begin(9600);
9   bluetooth.begin(9600);
10 }
11
12 void loop(){
13   if (bluetooth.available()) {

```

```

14 Serial.write(blueetooth.read());
15 }
16 if (Serial.available()) {
17   bluetooth.write(Serial.read());
18 }
19 }

```

3) 센서 테스트

- 시리얼 모니터에서 'AT' 입력 시 'OK' 라고 올라오는 것을 확인



AT COMMAND LISTING

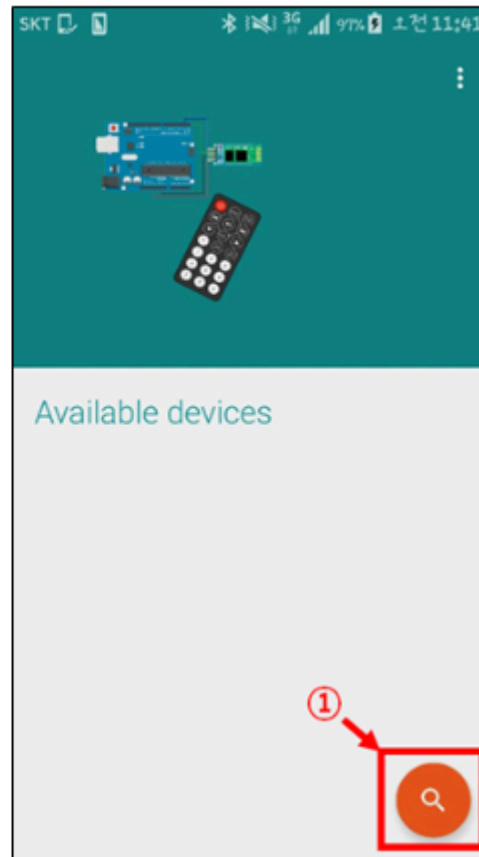
	COMMAND	FUNCTION
1	AT	Test UART Connection
2	AT+RESET	Reset Device
3	AT+VERSION	Query firmware version
4	AT+ORGL	Restore settings to Factory Defaults
5	AT+ADDR	Query Device Bluetooth Address
6	AT+NAME	Query/Set Device Name
7	AT+RNAME	Query Remote Bluetooth Device's Name
8	AT+ROLE	Query/Set Device Role
9	AT+CLASS	Query/Set Class of Device CoD
10	AT+IAC	Query/Set Inquire Access Code
11	AT+INQM	Query/Set Inquire Access Mode
12	AT+PSWD	Query/Set Pairing Passkey
13	AT+UART	Query/Set UART parameter
14	AT+CMODE	Query/Set Connection Mode
15	AT+BIND	Query/Set Binding Bluetooth Address
16	AT+POLAR	Query/Set LED Output Polarity
17	AT+PIO	Set/Reset a User I/O pin
18	AT+MPIO	Set/Reset multiple User I/O pin
19	AT+MPIO?	Query User I/O pin
20	AT+PSCAN	Query/Set Scanning Parameters
21	AT+SNIFF	Query/Set SNIFF Energy Savings Parameters

ERROR CODES

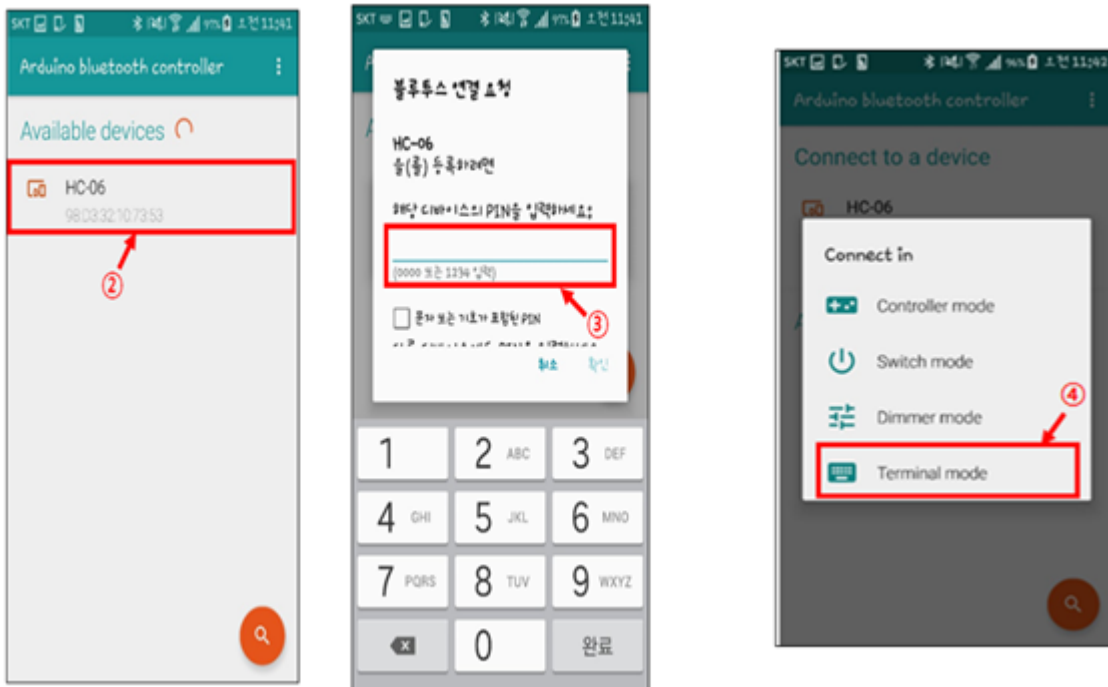
ERROR CODE	VERBOSE
0	Command Error/Invalid Command
1	Results in default value
2	PSKEY write error
3	Device name is too long (>32 characters)
4	No device name specified (0 length)
5	Bluetooth address NAP is too long
6	Bluetooth address UAP is too long
7	Bluetooth address LAP is too long
8	PIO map not specified (0 length)
9	Invalid PIO port Number entered
A	Device Class not specified (0 length)
B	Device Class too long
C	Inquire Access Code not Specified (0 length)
D	Inquire Access Code too long
E	Invalid Inquire Access Code entered
F	Pairing Password not specified (0 length)
10	Pairing Password too long (> 16 characters)
11	Invalid Role entered
12	Invalid Baud Rate entered
13	Invalid Stop Bit entered
14	Invalid Parity Bit entered
15	No device in the Pairing List

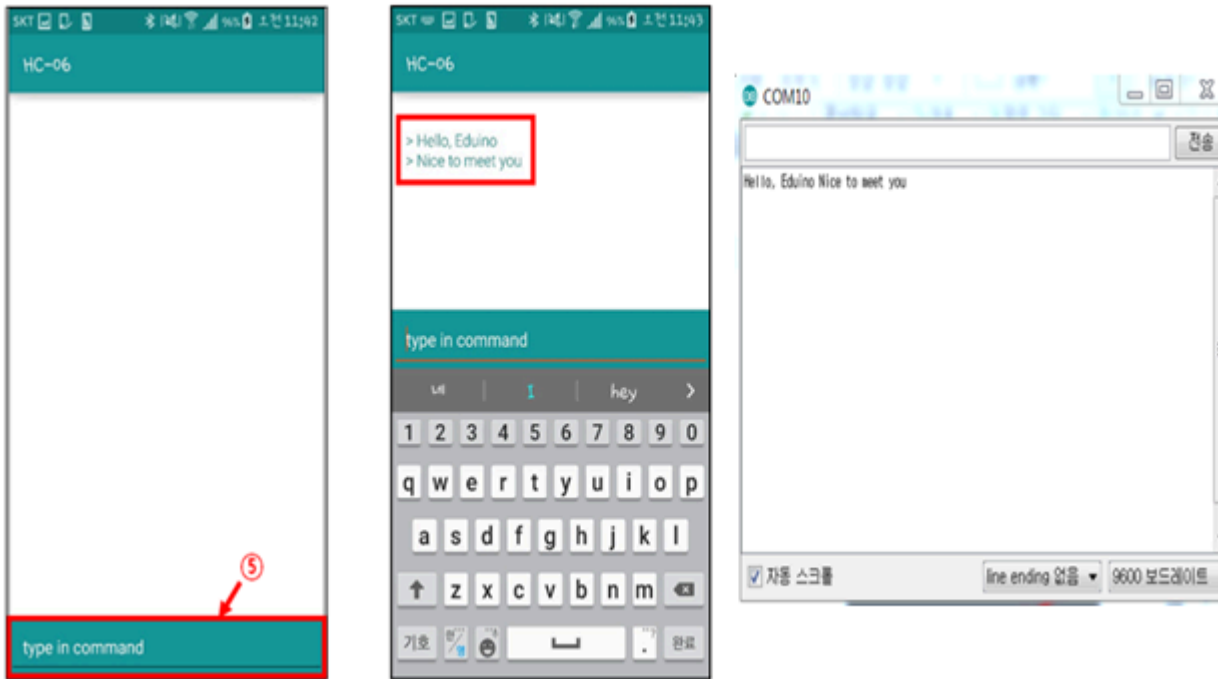
✓ 4. 스마트폰 어플과 블루투스 통신하기

- 스마트폰의 어플스토어에서 'Arduino Bluetooth Controller' 이름을 가진 어플리케이션을 다운로드 및 설치
- 스마트폰 주변의 블루투스 장치 검색을 시작
- 초기 PIN번호 1234를 입력



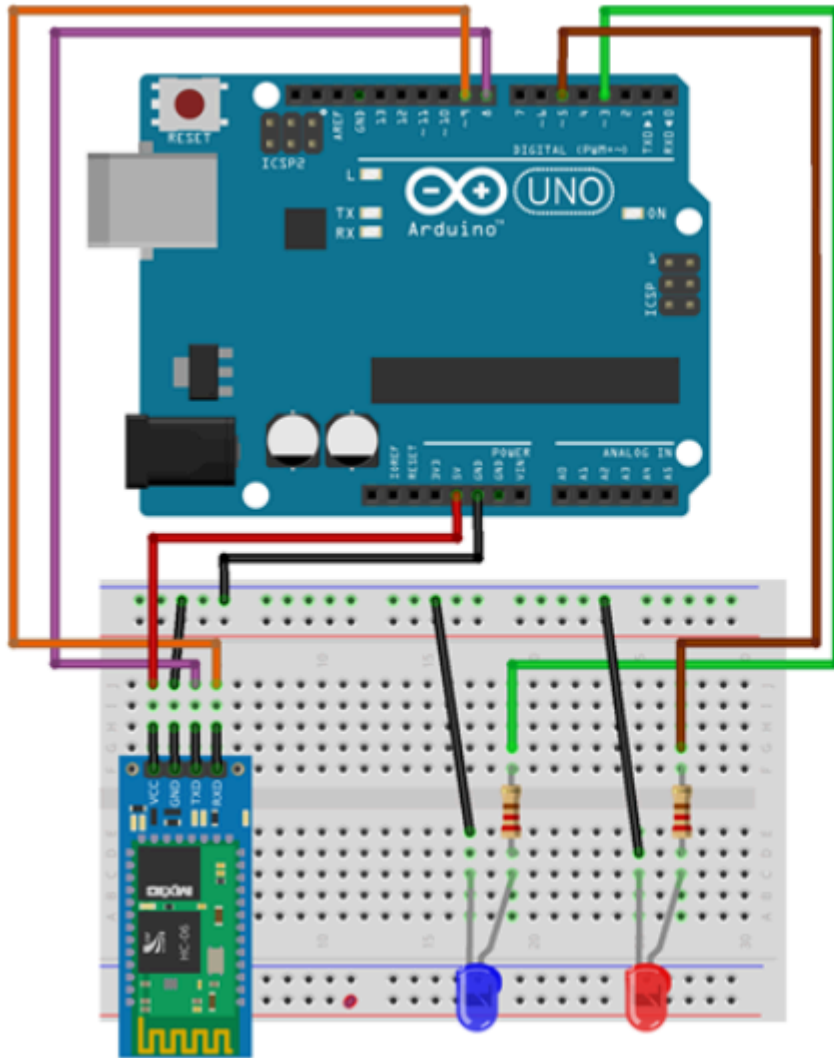
- 연결 모드를 선택하는 화면에서 모드는 총 4가지 선택이 가능하며, 여기서는 '[Terminal mode](#)'를 선택하여 스마트폰과 블루투스 모듈 간의 문자 송수신을 테스트





✓ 블루투스(어플)를 이용하여 LED 제어하기

- 1) 아두이노와 연결하기



우노 보드	LED
D3	파랑 +
D5	빨강 +
GND	GND

우노 보드	HC-06
VCC	VCC
GND	GND
D8	TX
D9	RX

- 2) 코딩


```
1 #Bluetooth_LED
2 #include <SoftwareSerial.h>
3 SoftwareSerial bluetooth(8, 9); // 8번 TX, 9번 RX
4 void setup()
5 {
6   Serial.begin(9600);
7   bluetooth.begin(9600);
8   pinMode(3, OUTPUT); // LED OUTPUT
9   pinMode(5, OUTPUT); // LED OUTPUT
10 }
11
12 void loop()
13 {
14   char val = bluetooth.read();
15   if (bluetooth.available()){
16     Serial.write(bluetooth.read());
17   }
18   if(val == 'a'){
19     digitalWrite(3,HIGH);
20   }
21   if(val == 'b'){
22     digitalWrite(5, HIGH);
23   }
24   else if(val == 'c'){
25     digitalWrite(3, LOW);
26     digitalWrite(5, LOW);
27   }
28 }
```

- 3) 센서 테스트

- 'Terminal mode' 에서 'a'를 입력하면 위 사진처럼 파란색 LED 점등 되는 것을 확인
- 'Terminal mode' 에서 'b'를 입력하면 위 사진처럼 붉은색 LED 점등 되는 것을 확인

- 'Terminal mode' 에서 'c'를 입력하면 위 사진처럼 모든 LED를 소등

