

임베디드 기반
근거리 블루투스 무선 통신
응용 실습

제주대학교

김도현

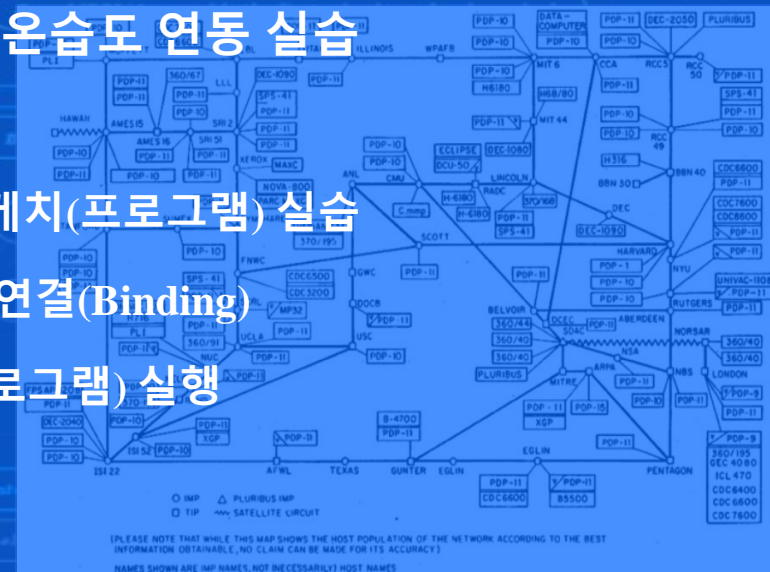
주요 학습 내용

– 아두이노 블루투스 통신 기반 문자열 전달 및 분석 실습

- 블루투스 통신 기반 문자열 전달 환경
- 아두이노 문자열(String) 관련 아두이노 함수
- 블루투스 통신 기반 문자열 전달 및 분석 스케치(프로그램) 실습
- 블루투스 Master와 Slave 설정(Command) 및 연결(Binding)
- 블루투스 통신 기반 문자열 전달 및 분석 스케치(프로그램) 실행

– 아두이노 기반 근거리 블루투스 무선 통신과 온습도 연동 실습

- 블루투스 무선 통신과 온습도 연동 환경
- 근거리 블루투스 무선 통신과 온습도 연동 스케치(프로그램) 실습
- 블루투스 Master와 Slave 설정(Command) 및 연결(Binding)
- 블루투스 무선 통신과 온습도 연동 스케치(프로그램) 실행

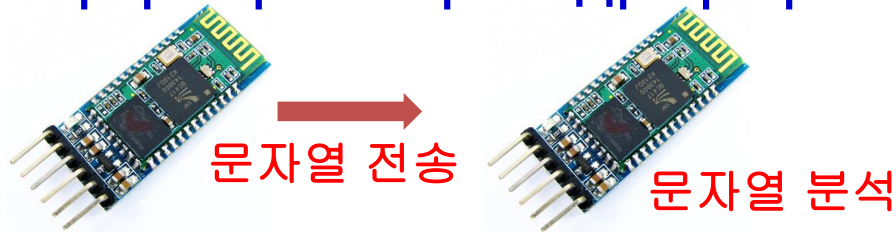


아두이노 기반 근거리 블루투스 무선 통신 기반 문자열 전달 및 분석

- 블루투스 master를 통해 명령을 slave에 전달하여 송신한 메시지를 분석하는 실습
 1. 블루투스 마스터에서 메시지를 Slave로 전달.
 2. 블루투스 Slave에서 문자열(string) 형태의 데이터를 읽기
 3. 블루투스 Slave에서 문자열(string) 읽어 드린 화면에 출력
 4. 블루투스 Slave에서 문자열(string) 읽어 드린 값의 문자열 길이를 화면에 출력
 5. 블루투스 Slave에서 문자열(string) 읽어 드린 값의 첫 번째 값을 화면에 출력
 6. 블루투스 Slave에서 문자열(string) 문자열의 앞뒤 공백 문자 제거

근거리 블루투스 무선 통신을 위한 아두이노 구성

- 아두이노 하드웨어와 소프트웨어



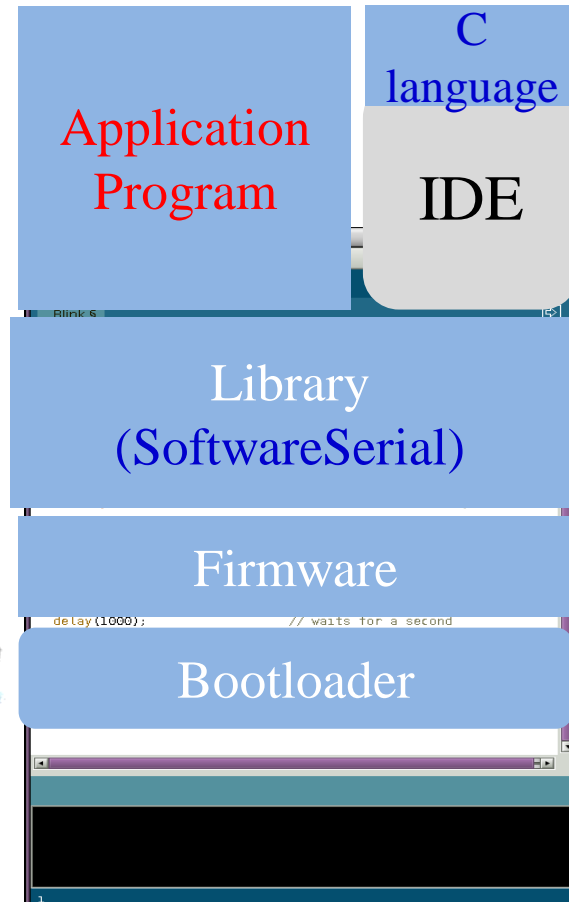
Bluetooth (HC-05)

Digital I/O (2,3 pin)

Microcontroller

Hardware Board

아두이노 하드웨어



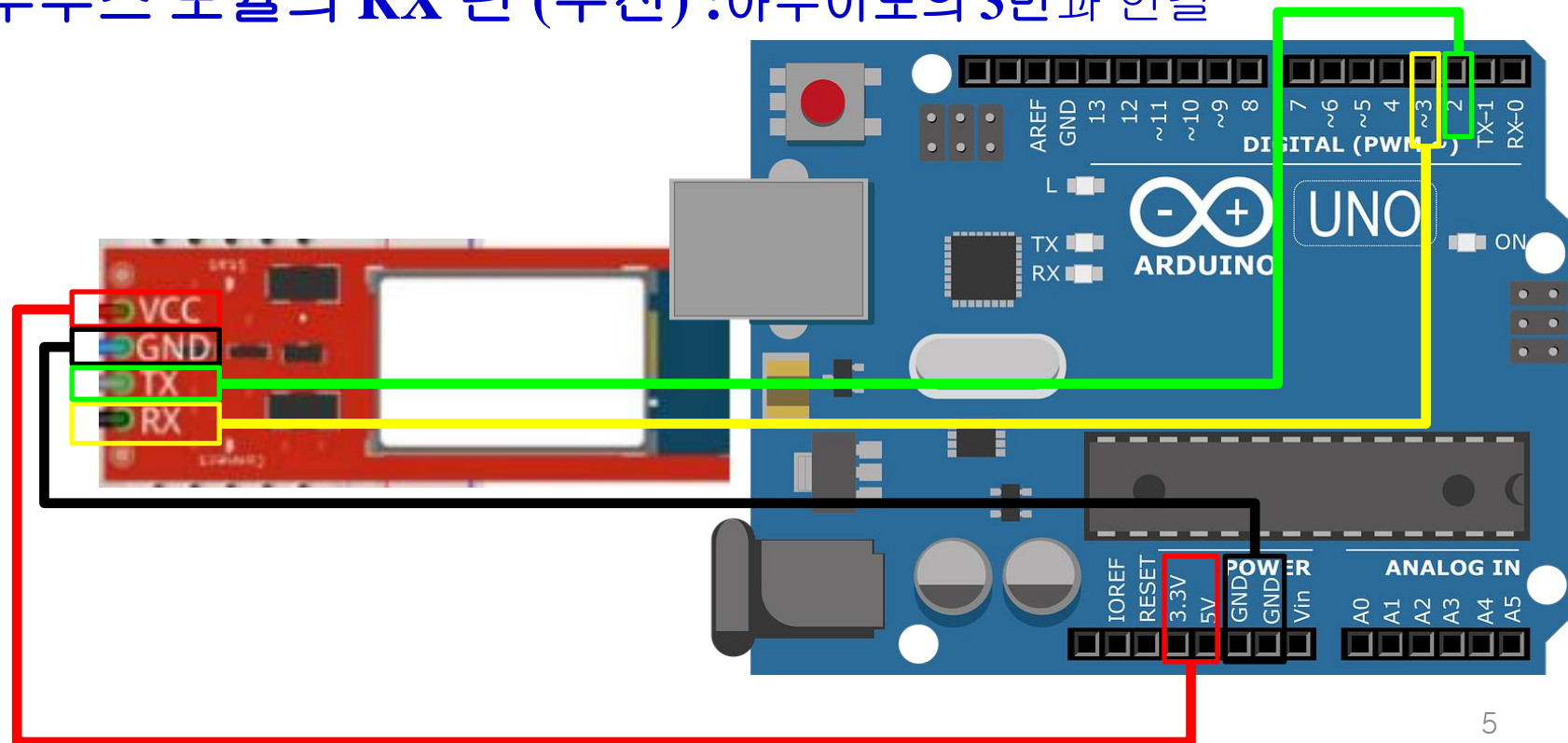
아두이노 소프트웨어

(IDE : Integrated Development Environment)

근거리 블루투스 무선 통신을 위한 블루투스 회로 구성

- **VCC(+)**부분 : 아두이노의 (HC-05은 5V)
- **GND(-)**부분 : 아두이노의 **GND**
- 블루투스 모듈의 **TX** 핀 (전송) : 아두이노의 2번과 연결
- 블루투스 모듈의 **RX** 핀 (수신) : 아두이노의 3번과 연결

구분	기능	핀
T	전송(transfer)	TX핀(2번)
R	수신(receiver)	RX핀(3번)



아두이노 기반 블루투스 통신 기반 문자열 전달 및 분석 실습 환경

1. 아두이노 하드웨어간의 블루투스 통신 환경

아두이노 (Arduino)

컴퓨터(Host)

1 Word



시리얼 통신(USB)

마스터 (master)

블루투스 (문자열 전송)

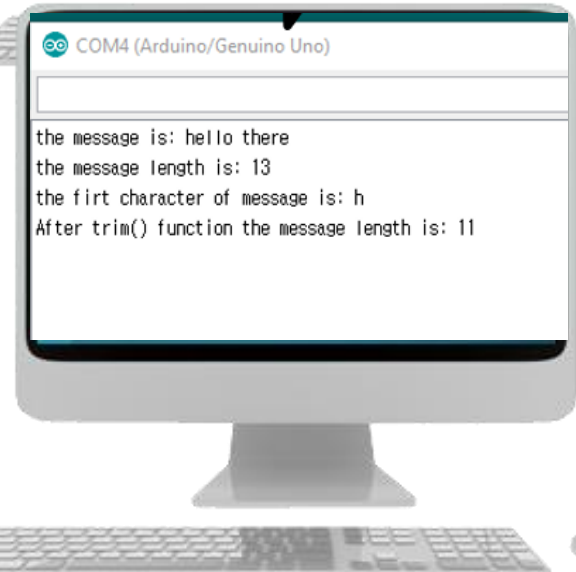
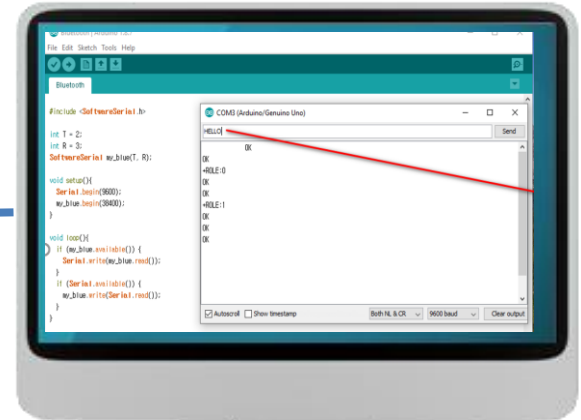
시리얼 통신(USB)

1 Word



슬레이브(Slave)

문자열 분석



아두이노 문자열(String) 관련 아두이노 함수

- `String.trim()`
 - 앞뒤 공백 문자 제거.
- `String.toLowerCase()`
 - 알파벳 문자를 찾아 모두 소문자로 변환.
- `String.toInt()`
 - 숫자 문자열이면 실제 숫자 값으로 변환.
- `String.substring(start, end)`
 - `String` 객체에서 원하는 자라의 문자열을 복사하여 새로운 `String` 객체로 반환.
- `String.length()`
 - 객체의 문자열 길이 반환.
- `String.concat(value)`
 - 객체의 뒤부분에 `value` 변수의 값 첨부.
- `String.equals(value)`
 - 객체와 `value` 변수의 값과 비교를 하여 같으면 1 다르면 0 반환.
- `String[index]`
 - 객체에 저장된 문자열 중 `index` 번째의 문자 반환.

아두이노 기반 근거리 블루투스 무선 통신과 문자열 분석 스케치(프로그램) 실습 (마스터 소스 프로그램)

```
#include <SoftwareSerial.h>
// 블루투스 통신을 사용하기 위해 불러오는 라이브러리

int T = 2;
int R = 3;
// 블루투스 모듈의 TXD 핀과 연결된 2번을 T 라는 변수로,
// 블루투스 모듈의 RXD 핀과 연결된 3번을 R 이라는 변수로 선언

SoftwareSerial my_blue(T, R);
// my_blue 라는 이름의 블루투스 객체를 선언
// 이 블루투스 객체는 T 와 R 에 의해 제어
void setup(){
  Serial.begin(9600);
  // 시리얼 통신을 9600 의 통신속도로 시작한다.
  my_blue.begin(38400);
  // 블루투스 통신을 시작한다.
}

void loop(){
  if (my_blue.available()) { //블루투스를 사용할 수 있다면
    Serial.write(my_blue.read()); //값을 읽어서 시리얼에 쓰기
  }
  if (Serial.available()) { //시리얼을 사용할 수 있다면
    my_blue.write(Serial.read()); //값을 읽어 블루투스에 쓰기
  }
}
```


아두이노 기반 근거리 블루투스 무선 통신과 문자열 분석 스케치(프로그램) 실습 (Slave 소스 프로그램)

```
#include <SoftwareSerial.h>

int t = 2;

int r = 3;

SoftwareSerial ble(t,r);

void setup() {

    Serial.begin(9600);

    ble.begin(38400);

}

void loop() {

    if(Serial.available()){

        ble.write(Serial.read());

    }

    if(ble.available()){

        String message = ble.readString(); //string 형태의 데이터를 읽어 드림

        Serial.print("the message is: ");

        Serial.print(message); //읽어 드린 값 출력

        Serial.print("the message length is: ");

        Serial.println(message.length()); //읽어 드린 값의 문자열 길이를 출력

        Serial.print("the first character of message is: ");

        Serial.println(message[0]); //읽어 드린 값의 첫 번째 값 출력

        Serial.print("After trim() function the message length is: ");

        message.trim(); //문자열의 앞뒤 공백 문자 제거

        Serial.println(message.length());

    }

}
```

근거리 블루투스 무선 통신을 위한 Master와 Slave 설정(Command)과 연결(Binding)

1. AT Master와 Slave 설정(Command) 모드(mode)로 변환 및 Serial monitor 입력 형태 변환
2. 블루투스 초기화
3. 블루투스 Master와 Slave 역할 설정
4. 아두이노 Master에서 Slave Address 확인
5. 아두이노 Master에서 블루투스 Master와 Slave 연결(Binding) 및 확인
6. 블루투스 설정(Command) 모드에서 연결(Binding) 모드 변환 완료

아두이노에서 블루투스 Master와 Slave 연결(Binding) 및 확인

- Master에서 연결한 주소 확인

- AT+BIND?

- Master에서 연결할 Master 주소 지정

- AT+BIND= 98d3,21,fc81ce

- 주소를 “:” 부호에서 “,”로 바꿈.

- Master와 Slave 연결(Binding) 확인

- 블루투스의 불빛이 동시에 깜빡이면 연결이 된 것이다.

COM3 (Arduino/Genuino Uno)

```
AT+BIND?  
OK  
+ROLE:0  
OK  
OK  
+ROLE:1  
OK  
+ADDR:98d3:61:fd5644  
OK  
+BIND:0:0:0  
OK
```

바인딩 된 주소

Slave Address

Master Address

COM3 (Arduino/Genuino Uno)

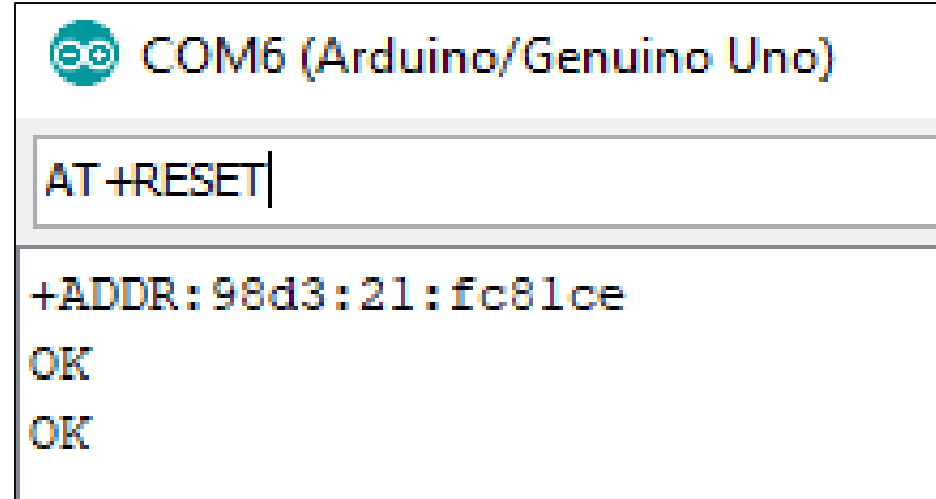
```
AT+BIND=98d3,21,fc81ce  
OK  
+ROLE:0  
OK  
OK  
+ROLE:1  
OK  
+ADDR:98d3:61:fd5644  
OK  
+BIND:0:0:0  
OK  
OK
```

바인딩 할 주소



블루투스 설정(Command) 모드에서 연결(Binding) 모드 변환 완료

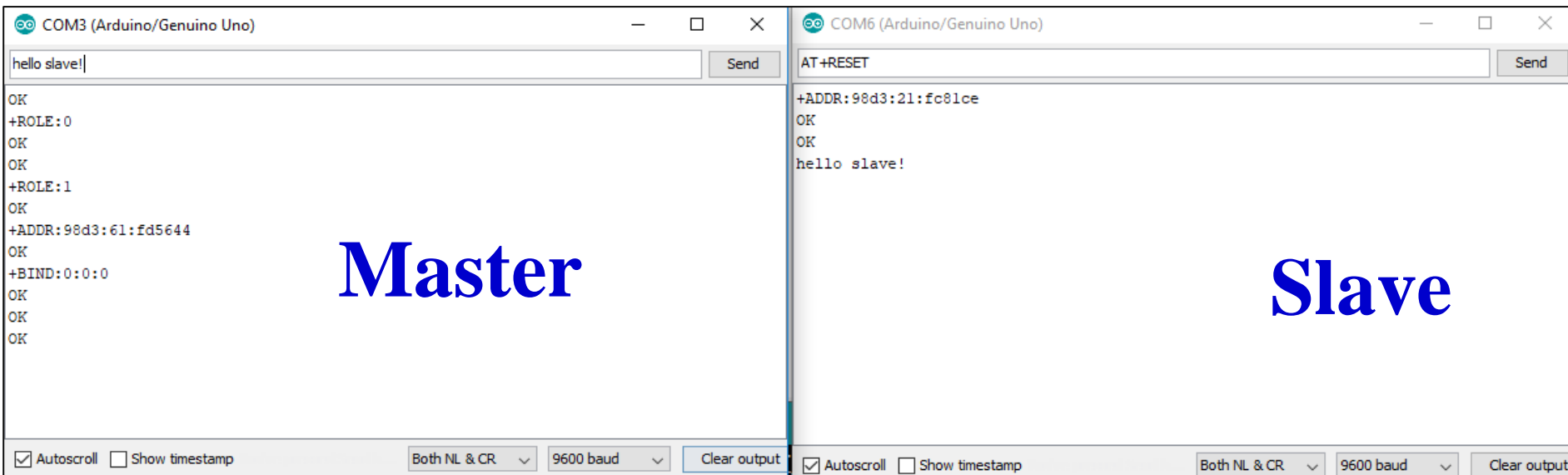
- 아두이노를 AT 명령어 모드
에서 연결 모드로 변환함.
 - AT+RESET



```
COM6 (Arduino/Genuino Uno)

AT+RESET

+ADDR:98d3:21:fc81ce
OK
OK
```



COM3 (Arduino/Genuino Uno)

hello slave!

Send

OK
+ROLE:0
OK
OK
+ROLE:1
OK
+ADDR:98d3:61:fd5644
OK
+BIND:0:0:0
OK
OK
OK

Master

☒ Autoscroll ☐ Show timestamp Both NL & CR 9600 baud Clear output

COM6 (Arduino/Genuino Uno)

AT+RESET

Send

+ADDR:98d3:21:fc81ce
OK
OK
hello slave!

Slave

☒ Autoscroll ☐ Show timestamp Both NL & CR 9600 baud Clear output

아두이노 기반 근거리 블루투스 무선 통신과 문자열 분석 스케치(프로그램) 실습 결과

Slave

마스터

Bluetooth-Training1-S | Arduino 1.8.7
File Edit Sketch Tools Help

```
Bluetooth-Training1-S
#include <SoftwareSerial.h>
int t = 2;
int r = 3;

SoftwareSerial ble(t,r);

void setup() {
  Serial.begin(9600);
  ble.begin(38400);
}

void loop() {
  if(Serial.available()){
    ble.write(Serial.read());
  }

  if(ble.available()){
    String message = ble.readString();
    Serial.print("the message is: ");
    Serial.print(message);
    Serial.print("the message length is: ");
    Serial.println(message.length());
    Serial.print("the first character of message is: ");
    Serial.println(message[0]);
    Serial.print("After trim() function the message length is: ");
    message.trim();
    Serial.println(message.length());
  }
}
```

COM4 (Arduino/Genuino Uno)

```
the message is: hello there
the message length is: 13
the first character of message is: h
After trim() function the message length is: 11
```

☒ Autoscroll ☐ Show timestamp Both NL & CR 9600 baud Clear output

```
Bluetooth-another
#include <SoftwareSerial.h>

int T = 2;
int R = 3;
SoftwareSerial my_blue(T, R);

void setup(){
  Serial.begin(9600);
  my_blue.begin(38400);
}

void loop(){
  if (my_blue.available()) {
    Serial.write(my_blue.read());
  }
  if (Serial.available()) {
    my_blue.write(Serial.read());
  }
}
```

COM3 (Arduino/Genuino Uno)

☒ Autoscroll ☐ Show timestamp Both NL & CR 9600 baud Clear output

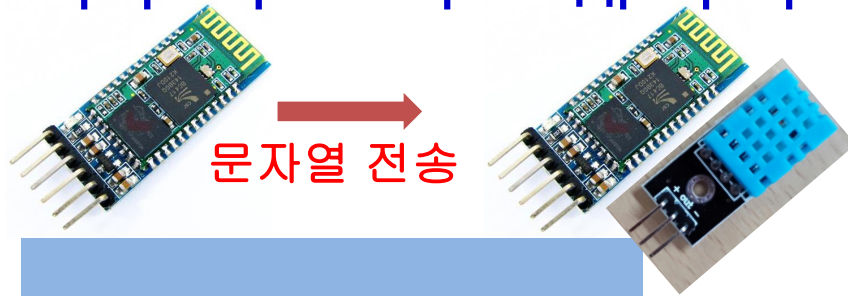
Sketch uses 5270 bytes (16%) of program storage space. Maximum is 32256 bytes.
Global variables use 414 bytes (20%) of dynamic memory, leaving 1634 bytes for local variables. Maximum is 2048 bytes.

아두이노 기반 근거리 블루투스 무선 통신과 온습도 연동 실습

- 블루투스 master를 통해 명령을 slave에 전달하여 온도/습도 센서로 데이터를 읽는 실습
 1. 블루투스 마스터에서 명령어를 Slave로 전달.
 2. 명령어 't'를 전달하면 Slave에서 온도 값을 반환.
 3. 명령어 'h'를 전달하면 Slave에서 습도 값을 반환.

아두이노 기반 근거리 블루투스 무선 통신과 온습도 연동 구성

• 아두이노 하드웨어와 소프트웨어



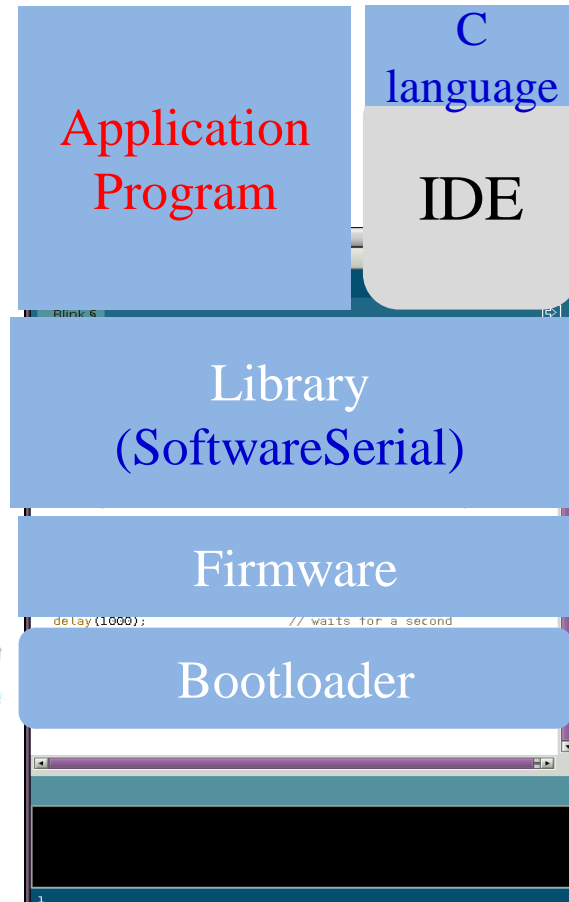
Bluetooth (HC-05)

Digital I/O (2,3 pin)

Microcontroller

Hardware Board

아두이노 하드웨어



아두이노 소프트웨어

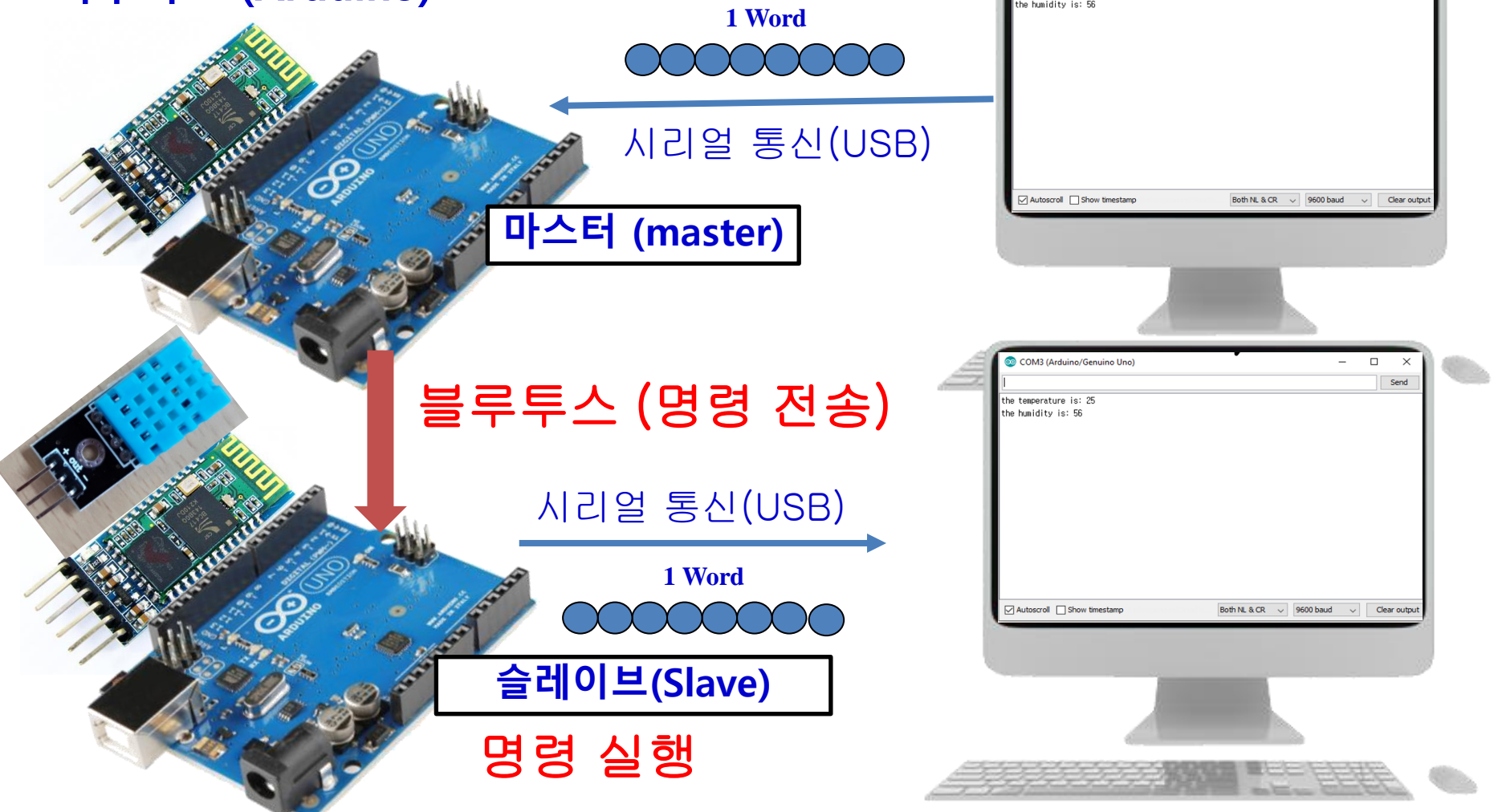
(IDE : Integrated Development Environment)

아두이노 간 블루투스 무선 통신 실습 환경

컴퓨터(Host)

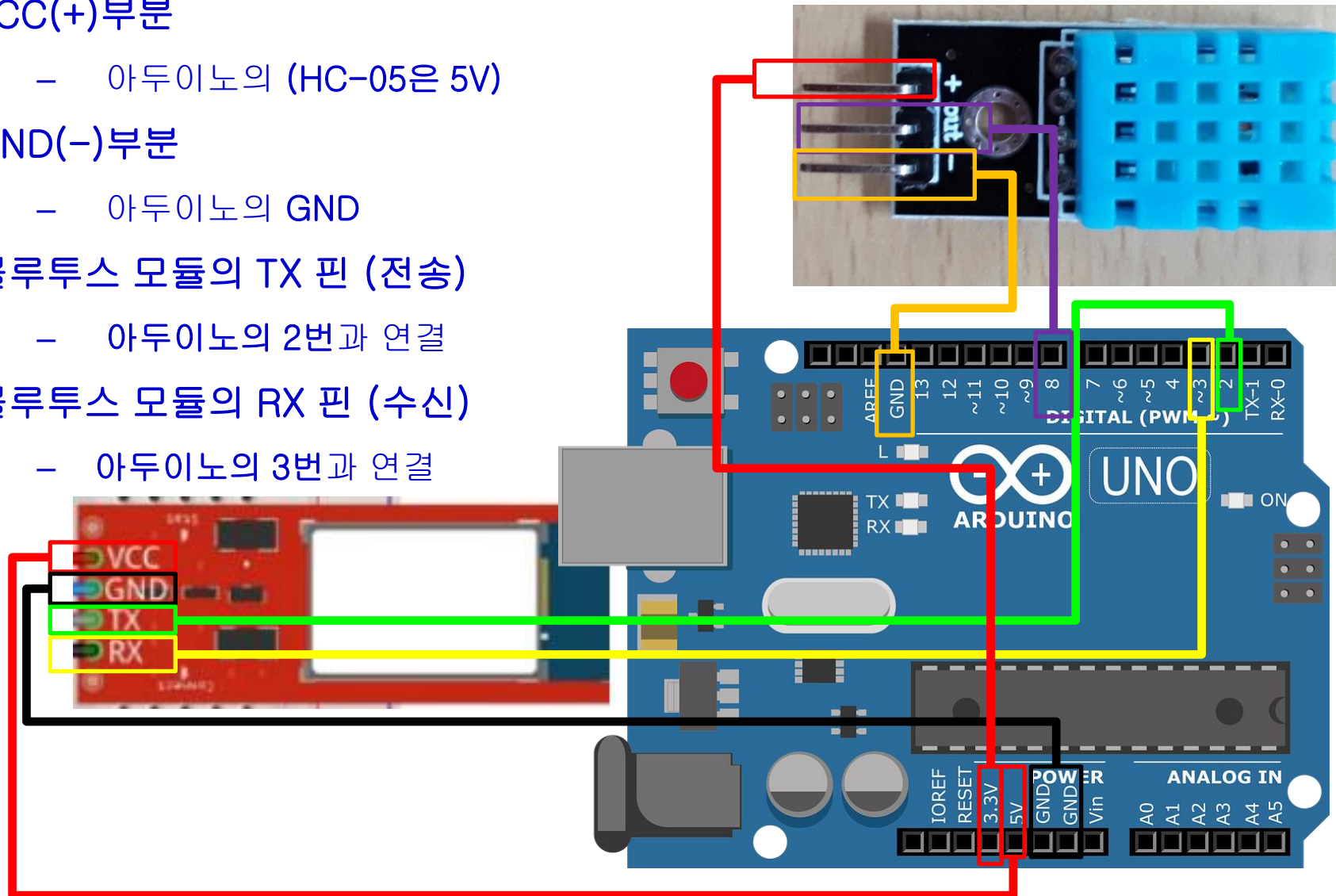
1. 아두이노 하드웨어 간의 블루투스 통신 환경

아두이노 (Arduino)



블루투스 무선 통신과 온습도 연동을 위한 Slave 블루투스 회로 구성

- VCC(+)부분
 - 아두이노의 (HC-05은 5V)
- GND(-)부분
 - 아두이노의 GND
- 블루투스 모듈의 TX 핀 (전송)
 - 아두이노의 2번과 연결
- 블루투스 모듈의 RX 핀 (수신)
 - 아두이노의 3번과 연결



아두이노 기반 근거리 블루투스 무선 통신과 온습도 연동 실습 (마스터 소스 프로그램)

```
#include <SoftwareSerial.h>
// 블루투스 통신을 사용하기 위해 불러오는 라이브러리

int T = 2;
int R = 3;
// 블루투스 모듈의 TXD 핀과 연결된 2번을 T 라는 변수로,
// 블루투스 모듈의 RXD 핀과 연결된 3번을 R 이라는 변수로 선언

SoftwareSerial my_blue(T, R);
// my_blue 라는 이름의 블루투스 객체를 선언
// 이 블루투스 객체는 T 와 R 에 의해 제어
void setup(){
  Serial.begin(9600);
  // 시리얼 통신을 9600 의 통신속도로 시작한다.
  my_blue.begin(38400);
  // 블루투스 통신을 시작한다.
}

void loop(){
  if (my_blue.available()) { //블루투스를 사용할 수 있다면
    Serial.write(my_blue.read()); //값을 읽어서 시리얼에 쓰기
  }
  if (Serial.available()) { //시리얼을 사용할 수 있다면
    my_blue.write(Serial.read()); //값을 읽어 블루투스에 쓰기
  }
}
```

아두이노 기반 근거리 블루투스 무선 통신과 온습도 연동 실습 (Slave 소스 프로그램)

```
#include <SoftwareSerial.h>

#include <DHT.h>

int t = 2;

int r = 3;

int pin = 8;

DHT sensor(pin, DHT11);

SoftwareSerial ble(t,r);

void setup() {

    Serial.begin(9600);

    ble.begin(38400);

    sensor.begin();

}

void loop() {

    if(Serial.available()){
        ble.write(Serial.read());
    }

    if(ble.available()){
        char command = ble.read(); //char 형태의 값을 읽어 드림
        if(command == 't'){ //char t와 비교하여 같으면
            int temp = sensor.readTemperature(); //센서로 부터 온도 값 읽어 드림
            String data = "the temperature is: "; //문자열 생성
            data.concat(temp); //온도값을 문자열에 추가
            data.concat("\n"); //문자열의 끝을 알림
            ble.write(data.c_str()); //블루투스를 통하여 마스터로 전송
            Serial.println(data); //시리얼로 전송
        }

        if(command == 'h'){ //char h와 비교하여 같으면
            int humi = sensor.readHumidity(); //센서로 부터 습도 값 읽어 드림
            String data = "the humidity is: ";
            data.concat(humi);
            data.concat("\n");
            ble.write(data.c_str());
            Serial.println(data);
        }

    }

}
```

근거리 블루투스 무선 통신을 위한 Master와 Slave 설정(Command)과 연결(Binding)

1. AT Master와 Slave 설정(Command) 모드(mode)로 변환 및 Serial monitor 입력 형태 변환
2. 블루투스 초기화
3. 블루투스 Master와 Slave 역할 설정
4. 아두이노 Master에서 Slave Address 확인
5. 아두이노 Master에서 블루투스 Master와 Slave 연결(Binding) 및 확인
6. 블루투스 설정(Command) 모드에서 연결(Binding) 모드 변환 완료

아두이노에서 블루투스 Master와 Slave 연결(Binding) 및 확인

- Master에서 연결한 주소 확인

- AT+BIND?

- Master에서 연결할 Master 주소 지정

- AT+BIND= 98d3,21,fc81ce

- 주소를 “:” 부호에서 “,”로 바꿈.

- Master와 Slave 연결(Binding) 확인

- 블루투스의 불빛이 동시에 깜빡이면 연결이 된 것이다.

COM3 (Arduino/Genuino Uno)

```
AT+BIND?  
OK  
+ROLE:0  
OK  
OK  
+ROLE:1  
OK  
+ADDR:98d3:61:fd5644  
OK  
+BIND:0:0:0  
OK
```

바인딩 된 주소

Slave Address

Master Address

COM3 (Arduino/Genuino Uno)

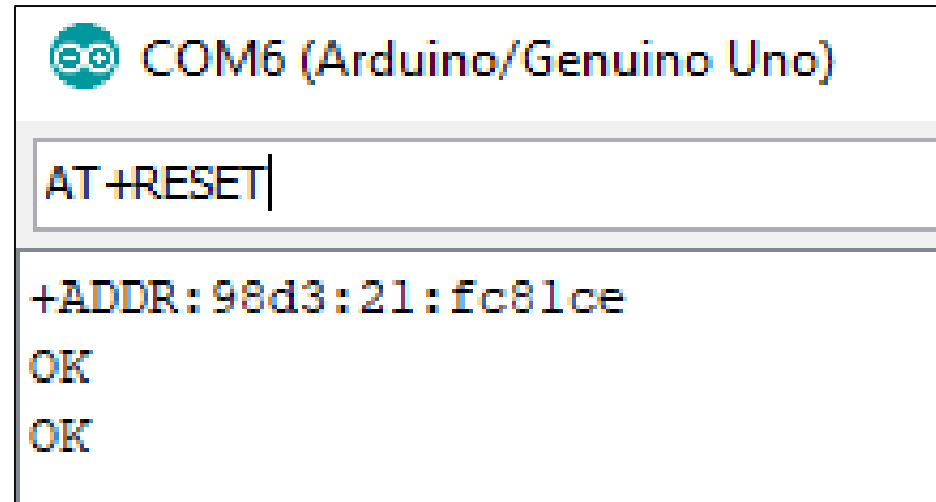
```
AT+BIND=98d3,21,fc81ce  
OK  
+ROLE:0  
OK  
OK  
+ROLE:1  
OK  
+ADDR:98d3:61:fd5644  
OK  
+BIND:0:0:0  
OK  
OK
```

바인딩 할 주소



블루투스 설정(Command) 모드에서 연결(Binding) 모드 변환 완료

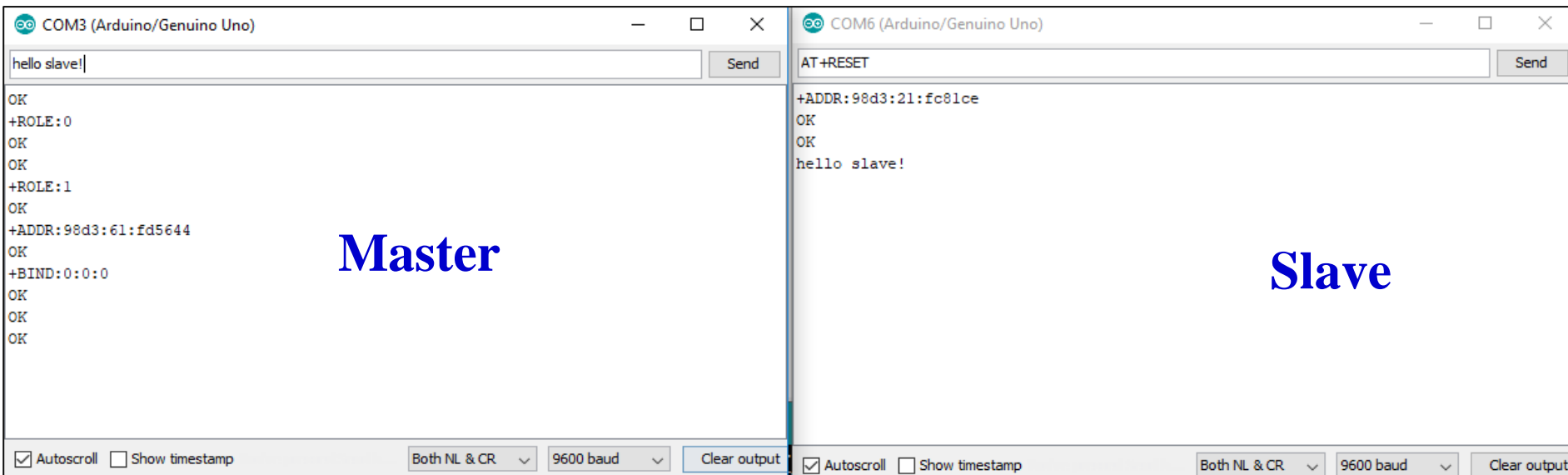
- 아두이노를 AT 명령어 모드
에서 연결 모드로 변환함.
 - AT+RESET



```
COM6 (Arduino/Genuino Uno)

AT+RESET

+ADDR:98d3:21:fc81ce
OK
OK
```



COM3 (Arduino/Genuino Uno)

hello slave!

Send

OK
+ROLE:0
OK
OK
+ROLE:1
OK
+ADDR:98d3:61:fd5644
OK
+BIND:0:0:0
OK
OK
OK

Master

☒ Autoscroll ☐ Show timestamp Both NL & CR 9600 baud Clear output

COM6 (Arduino/Genuino Uno)

AT+RESET

Send

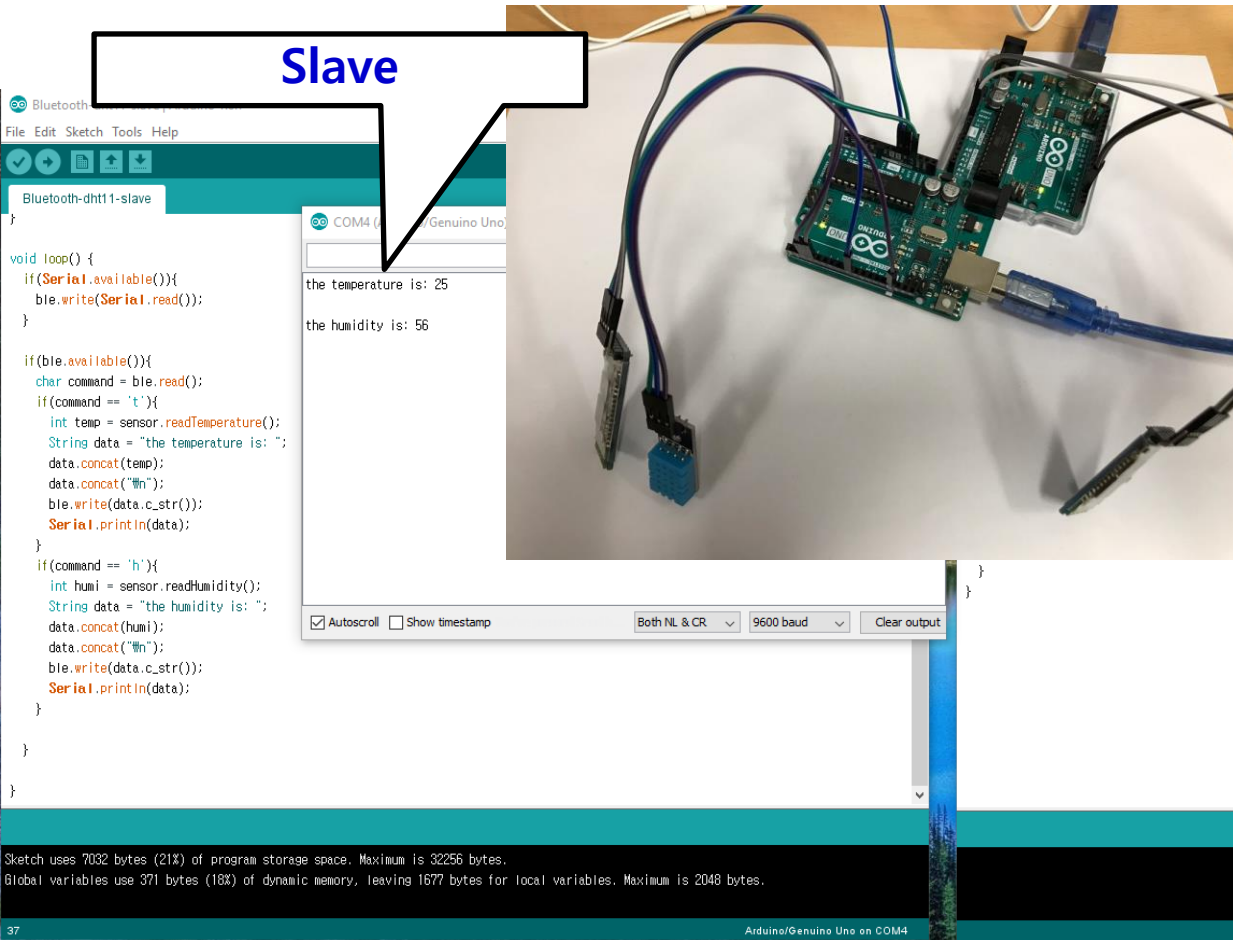
+ADDR:98d3:21:fc81ce
OK
OK
hello slave!

Slave

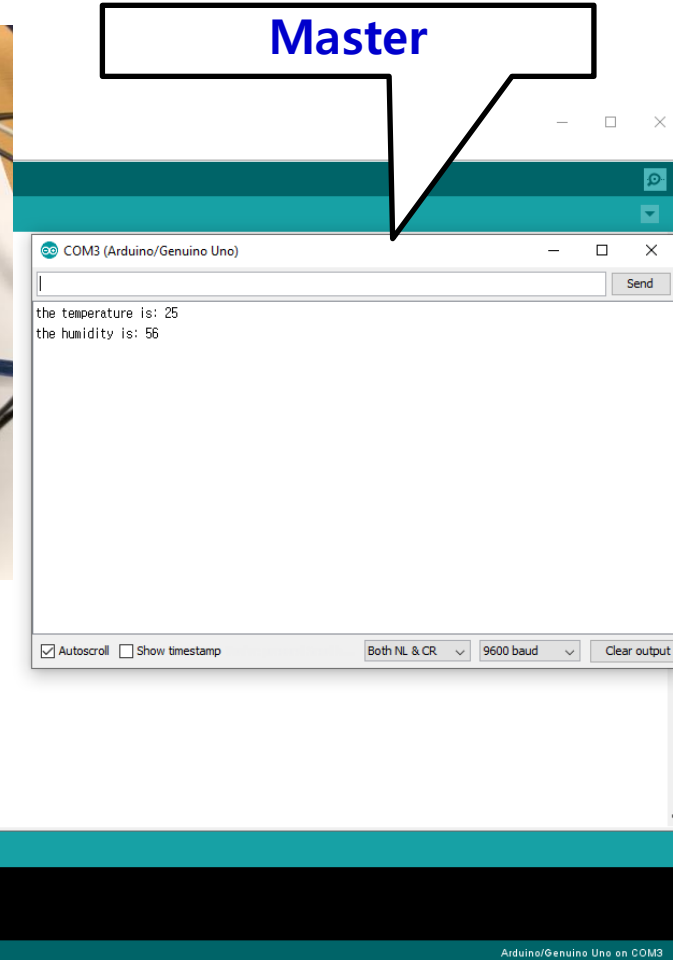
☒ Autoscroll ☐ Show timestamp Both NL & CR 9600 baud Clear output

아두이노 기반 근거리 블루투스 무선 통신과 온습도 연동 실습 결과

Slave



Master



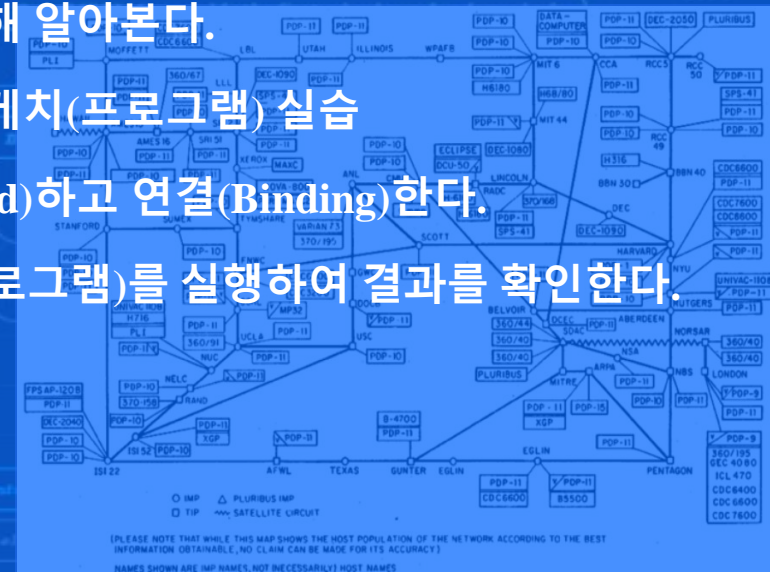
학습 내용 정리

– 아두이노 블루투스 통신 기반 문자열 전달 및 분석 실습을 진행한다.

- 블루투스 통신 기반 문자열 전달 환경에 대해 알아본다.
- 아두이노 문자열(String) 관련 아두이노 함수를 살펴본다.
- 블루투스 통신 기반 문자열 전달 및 분석 스케치(프로그램)을 실습한다.
- 블루투스 Master와 Slave 설정(Command)하고 연결(Binding)한다.
- 블루투스 통신 기반 문자열 전달 및 분석 스케치(프로그램)를 실행한다.

– 아두이노 기반 근거리 블루투스 무선 통신과 온습도 연동 실습을 진행한다.

- 블루투스 무선 통신과 온습도 연동 환경에 대해 알아본다.
- 근거리 블루투스 무선 통신과 온습도 연동 스케치(프로그램) 실습
- 블루투스 Master와 Slave 설정(Command)하고 연결(Binding)한다.
- 블루투스 무선 통신과 온습도 연동 스케치(프로그램)를 실행하여 결과를 확인한다.





수업을 마치겠습니다.

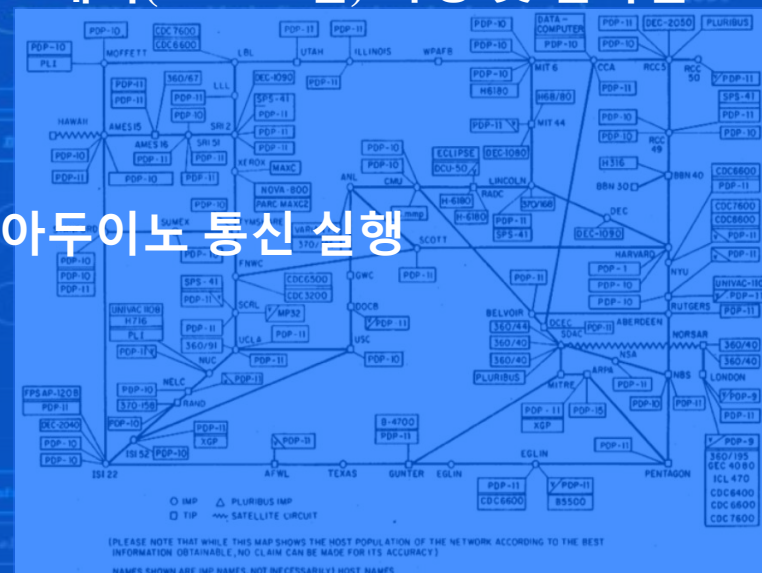
근거리 블루투스 기반
아두이노와 컴퓨터 통신
환경 구축 및 실습

제주대학교

김도현

주요 학습 내용

- 근거리 블루투스 무선 통신을 위한 아두이노 하드웨어 구성
- 근거리 블루투스 기반 아두이노와 컴퓨터 통신 실습 환경
- 아두이노 블루투스 Slave 설정(Command)
- 아두이노 블루투스 Slave 연결(Binding)
- 근거리 블루투스 무선 통신을 위한 아두이노 스케치(프로그램) 작성 및 컴파일
- 컴퓨터 블루투스 Master 연결(Binding)
- 근거리 블루투스 무선 통신을 위한 컴퓨터와 아두이노 통신 실행



근거리 블루투스 기반 아두이노 환경 구성

- 아두이노 하드웨어와 소프트웨어

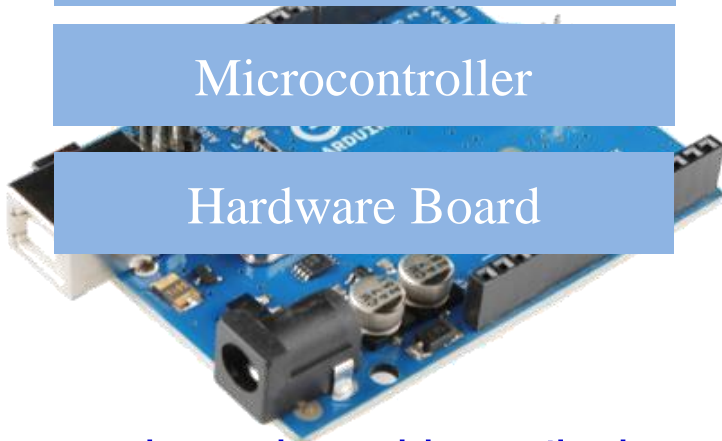


Bluetooth (HC-05)

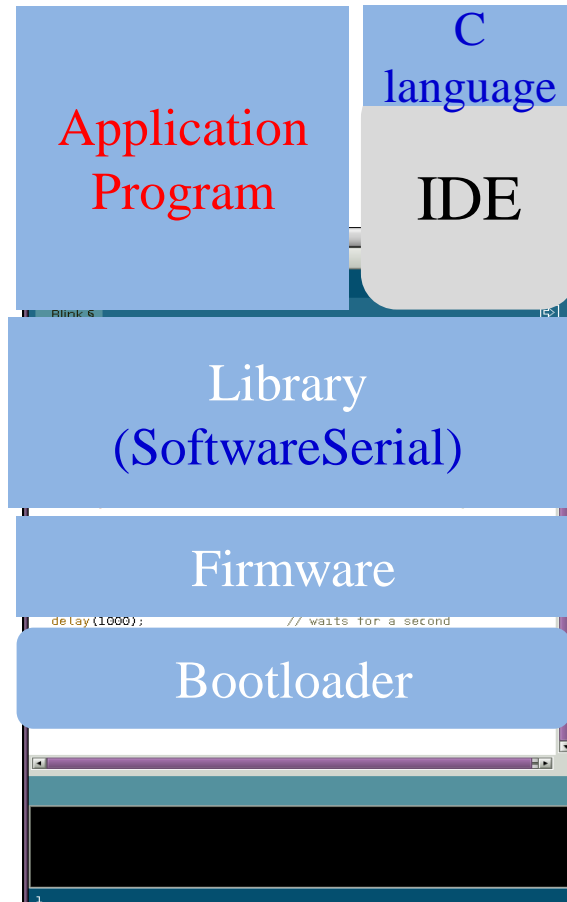
Digital I/O (2,3 pin)

Microcontroller

Hardware Board



아두이노 하드웨어



아두이노 소프트웨어

(IDE : Integrated Development Environment)

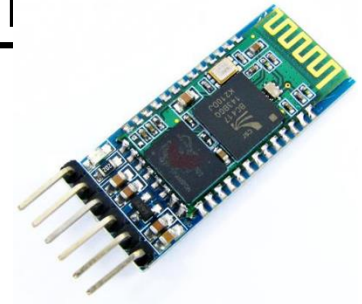
근거리 블루투스 무선 통신 기반 아두이노 하드웨어 구성 요소

- Layout of Arduino UNO

1. Digital Ground
3. Digital pins 2, 3
4. Digital pins Seral In/Out
5. Reset Button
6. In-circuit Serial Programmer
7. 8. Power and Ground pins
9. External Power Supply In
10. USB port

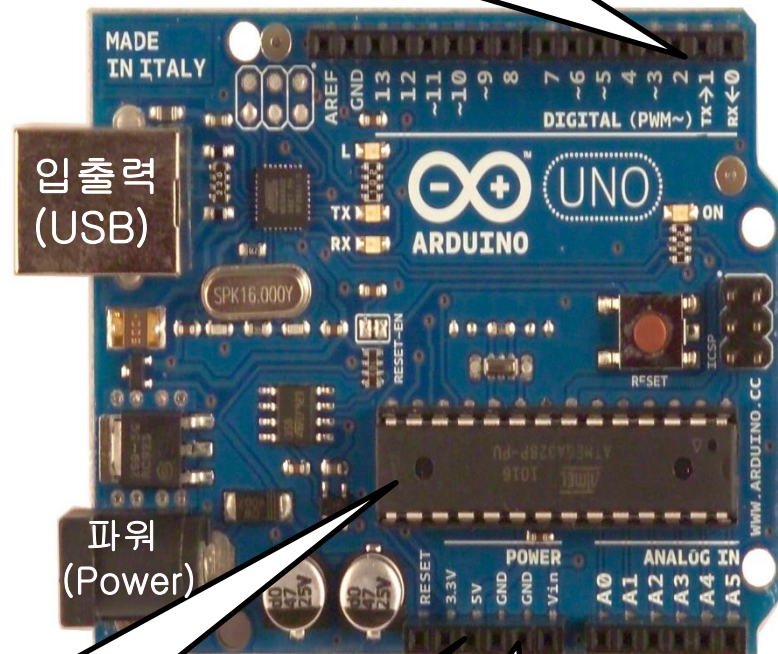


Jumper Wire
(Various Colors)



Bluetooth Module
(HC-05)

디지털 입/출력 PIN 2,3



입출력
(USB)

파워
(Power)

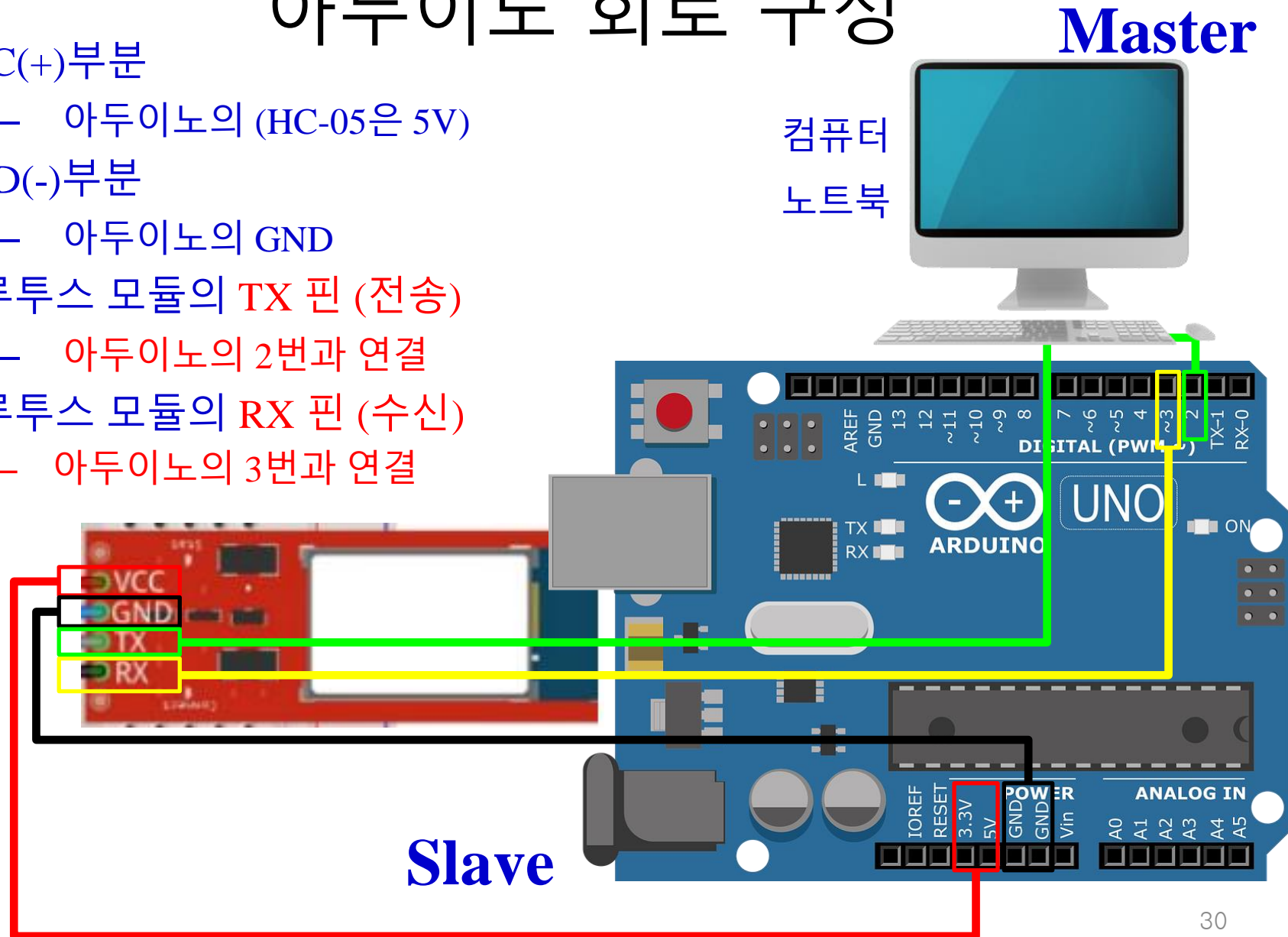
마이크로 컨트롤러
ATMega328

5V

GND

근거리 블루투스 무선 통신을 위한 아두이노 회로 구성

- VCC(+)부분
 - 아두이노의 (HC-05은 5V)
- GND(-)부분
 - 아두이노의 GND
- 블루투스 모듈의 TX 핀 (전송)
 - 아두이노의 2번과 연결
- 블루투스 모듈의 RX 핀 (수신)
 - 아두이노의 3번과 연결



근거리 블루투스 기반 아두이노와 컴퓨터 통신 실습 환경

컴퓨터

- 아두이노 하드웨어와 컴퓨터간의 블루투스 통신 환경

(블루투스 통신 가능)

아두이노 (Arduino)

1 Word

블루투스 (문자열 전송)

1 Word

시리얼 (문자열 전송)

Slave

Master

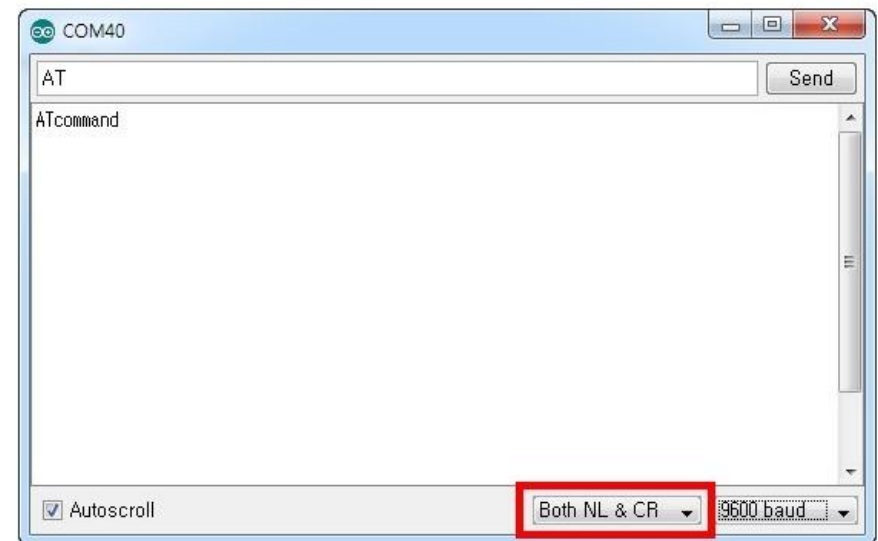
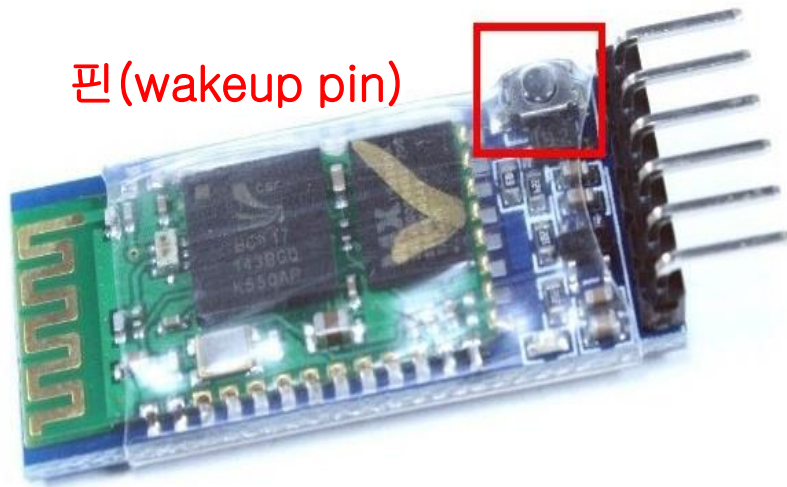
Slave 화면

근거리 블루투스 기반 컴퓨터 통신을 위한 아두이노 Slave 설정 (Command)과 연결(Binding)

1. Slave 디바이스 설정(Command) 모드(mode)로 변환 및
Serial monitor 입력 형태 변환
2. 아두이노 블루투스 초기화
3. 아두이노 블루투스 Slave 비밀번호 확인
4. 아두이노 블루투스 Slave 이름 설정
5. 아두이노 블루투스 Slave 속도 설정
6. 아두이노 블루투스 설정(Command) 모드에서 연결(Binding) 모드
변환 완료

근거리 블루투스 기반 컴퓨터 통신을 위한 아두이노 Slave 설정 1

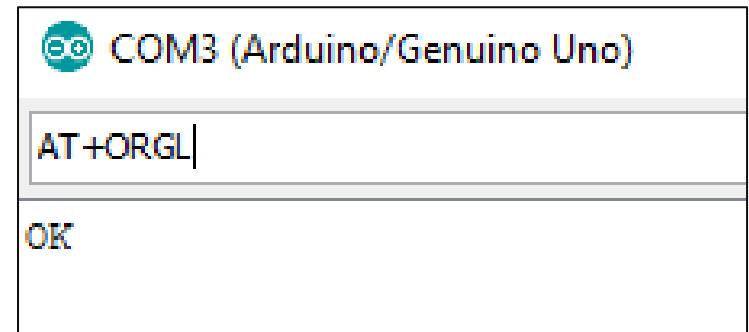
- 블루투스를 AT Master와 Slave 설정(Command) 모드(mode)로 변환
 - 아래의 핀(wakeup pin)을 눌러 준 상태에서 전원을 뽑았다 다시 꽂아 줌.
 - 정상적으로 AT Master와 Slave 설정 변환이 될 경우
 - 블루투스 모듈의 LED등의 빈도가 늦어 짐.
- 블루투스를 Serial monitor 입력 형태로 변환.



근거리 블루투스 기반 컴퓨터 통신을 위한 아두이노 Slave 설정 2

블루투스 초기화

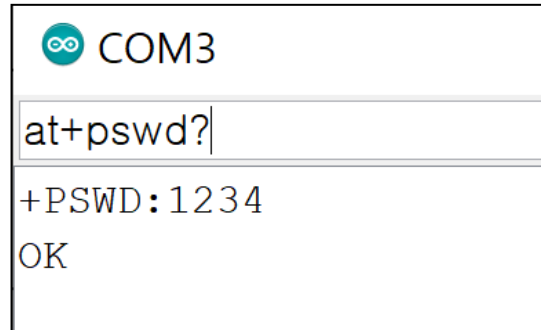
- 디폴트 상태로 설정 하기 위해 시리얼 모니터에서 AT+ORGL 명령어를 입력한다.
- Module work mode: Slave Mode
- Connection mode : connect to the specified device
- Serial parameter : Baud rate = 38400,0,0
- Specified device : 0:0:0



근거리 블루투스 기반 컴퓨터 통신을 위한 아두이노 Slave 설정 3

- 아두이노 블루투스 Slave 비밀번호 확인

- 비밀번호 확인
- at+pswd?



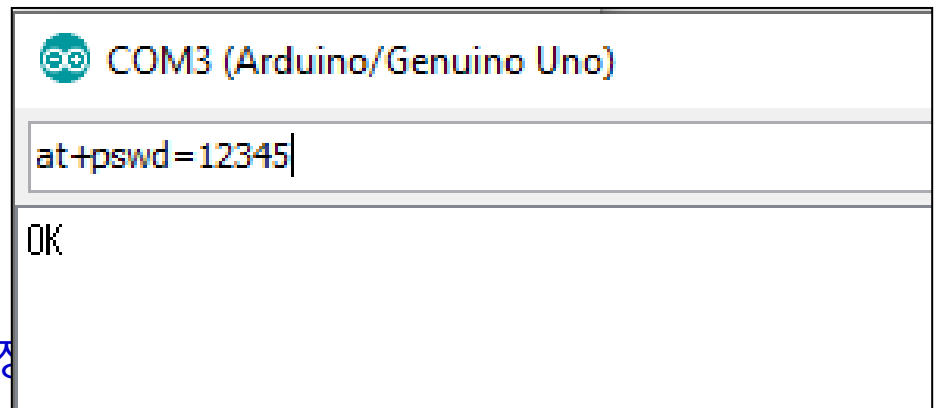
COM3

```
at+pswd?
```

+PSWD:1234
OK

- 비밀번호 설정

- at+pswd=1234



COM3 (Arduino/Genuino Uno)

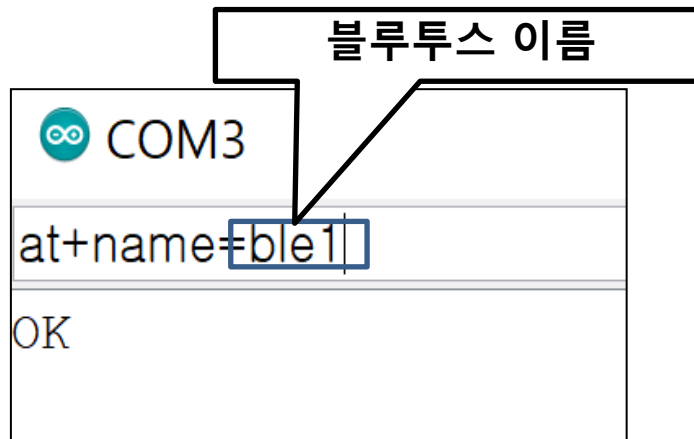
```
at+pswd=12345
```

OK

** 디폴트 상태로 비밀번호가 설정

근거리 블루투스 기반 컴퓨터 통신을 위한 아두이노 Slave 설정 4

- 아두이노 블루투스 Slave 이름 설정
 - 블루투스 이름 설정: 컴퓨터에서 연결할 아두이노 블루투스 디바이스 이름 설정
 - `at+name=ble1`




근거리 블루투스 기반 컴퓨터 통신을 위한 아두이노 Slave 설정 5

- 아두이노 블루투스 Slave 속도 설정

- 전송 속도 설정: 폰에서 사용하는 전송 속도와 맞추는 작업을 진행.


- 블루투스 속도 확인

- at+uart?

 COM3
at+uart?
ERROR: (0) +UART: 9600, 0, 0 OK

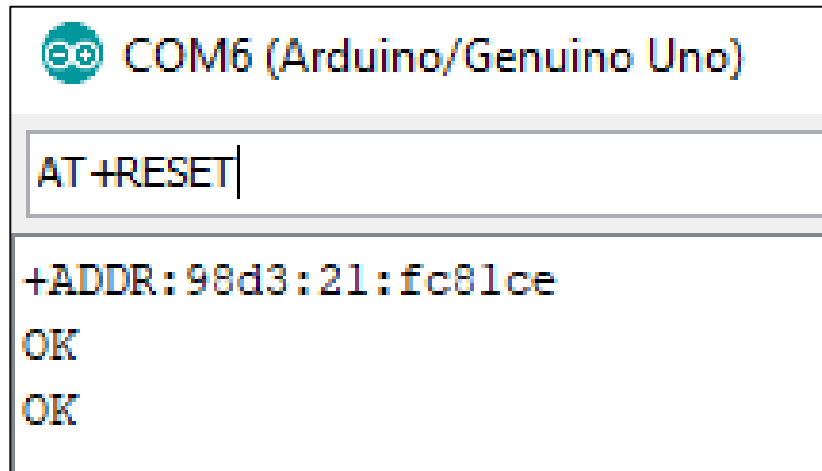
- 블루투스 속도 설정

- at+uart=9600,0,0

 COM3
at+uart=9600,0,0
ERROR: (0) +UART: 9600, 0, 0 OK OK

근거리 블루투스 기반 컴퓨터 통신을 위한 아두이노 Slave 연결 모드 6

- 아두이노 블루투스 설정(Command) 모드에서 연결(Binding) 모드 변환 완료
 - 아두이노를 AT 명령어 모드에서 연결 모드로 변환
 - 컴퓨터에서 아두이노와 블루투스 연결 진행.
 - at+reset



```
COM6 (Arduino/Genuino Uno)
AT+RESET
+ADDR:98d3:21:fc81ce
OK
OK
```

근거리 블루투스 무선 통신을 위한 아두이노 Slaver 스케치(프로그램)

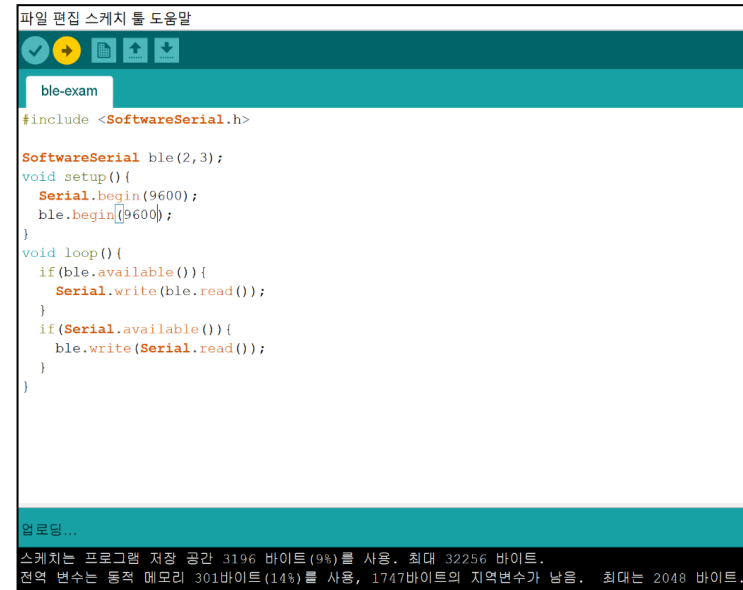
- 컴퓨터와 블루투스 실습을 위해 데이터 전송 속도를 38400에서 9600으로 수정 권고

```
#include <SoftwareSerial.h>           // 블루투스 통신을 사용하기 위해 불러오는 라이브러리
int T = 2;                             // 블루투스 모듈의 TXD 핀과 연결된 2번을 T 라는 변수 선언
int R = 3;                             // 블루투스 모듈의 RXD 핀과 연결된 3번을 R 이라는 변수 선언
```

```
SoftwareSerial my_blue(T, R);          // my_blue 라는 이름의 블루투스 객체 선언
// 이 블루투스 객체는 T 와 R 에 의해 제어
```

```
void setup(){
  Serial.begin(9600);                  // 시리얼 통신을 9600 의 통신속도로 시작한다.
  my_blue.begin(9600);                 // 블루투스 통신을 시작한다.
}
```

```
void loop(){
  if (my_blue.available()) {           //블루투스를 사용할 수 있다면
    Serial.write(my_blue.read());      //값을 읽어서 시리얼에 쓰기
  }
  if (Serial.available()) {             //시리얼을 사용할 수 있다면
    my_blue.write(Serial.read());      //값을 읽어 블루투스에 쓰기
  }
}
```



근거리 블루투스 무선 통신을 위한 아두이노 Slaver 스케치(프로그램) 실행

1. 새로운 스케치 생성

2. 컴파일

3. 업로드

4. 스케치(프로그램) 작성

파일 편집 스케치 툴 도움말



새 파일

2 3 1

파일 편집 스케치 툴 도움말



ble-exam

```
#include <SoftwareSerial.h>

SoftwareSerial ble(2,3);
void setup() {
  Serial.begin(9600);
  ble.begin(9600);
}
void loop() {
  if(ble.available()){
    Serial.write(ble.read());
  }
  if(Serial.available()){
    ble.write(Serial.read());
  }
}
```

4

업로딩...

스케치는 프로그램 저장 공간 3196 바이트(9%)를 사용. 최대 32256 바이트.
전역 변수는 동적 메모리 301바이트(14%)를 사용, 1747바이트의 지역변수가 남음. 최대는 2048 바이트.

근거리 블루투스 무선 통신을 위한 컴퓨터 블루투스 Master 연결(Binding)

- 컴퓨터 블루투스 연결(블루투스 지원하는 컴퓨터에서).
 - 컴퓨터 설정에서 블루투스 및 기타 아두이노 검색 및 확인
 - 컴퓨터 블루투스 활성화(1)
 - 아두이노 디바이스 추가(2)
 - 아두이노 디바이스 비밀번호 입력.

기타 디바이스



ble1
페어링됨

1

Bluetooth 및 기타 디바이스



Bluetooth 또는 기타 디바이스 추가

Bluetooth



컴

이제 "DESKTOP-PMIC41G"(으)로 검색될 수 있습니다.

디바이스 추가

2

추가할 디바이스 유형을 선택하세요.



Bluetooth

마우스, 키보드, 펜, 오디오 및 기타 유형의 **Bluetooth** 디바이스



무선 디스플레이 또는 도크

Miracast 또는 무선 도크를 사용하는 무선 모니터, TV 또는 PC

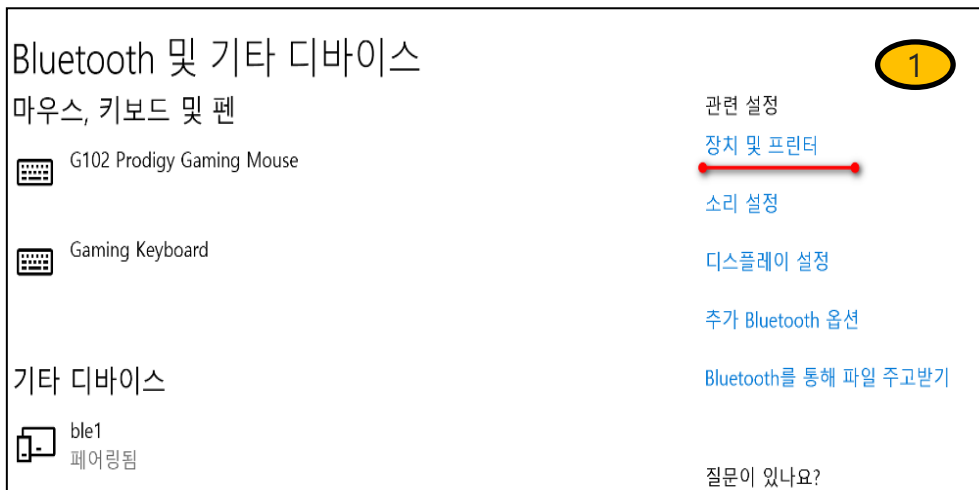
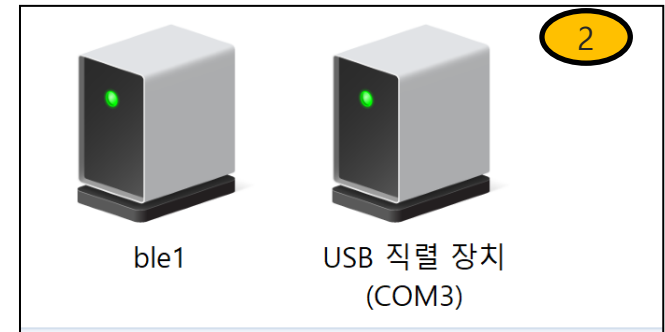


기타 모든 디바이스

무선 어댑터, **DLNA** 등의 **Xbox** 컨트롤러

근거리 블루투스 무선 통신을 위한 컴퓨터 블루투스 Master 연결(Binding)

- 컴퓨터 블루투스 연결(블루투스 지원하는 컴퓨터에서).
 - 블루투스 및 기타 디바이스에서 아두이노 디바이스에서(1)
 - 포트 번호 확인.
 - 장치 및 프린터 실행.
 - 블루투스 장치에서 마우스 오른쪽 선택(2)
 - 속성 선택.
 - 포트 번호 확인(3)



근거리 블루투스 무선 통신을 위한 컴퓨터 블루투스 Master 연결(Binding)

- 컴퓨터 블루투스 Master 연결(Binding) 공개 소프트웨어
 - Hercules SETUP utility 다운로드(<https://www.hw-group.com/software/hercules-setup-utility>)
 - Hercules SETUP utility 실행 및 테스트

The screenshot shows the HW group website's page for the Hercules SETUP utility. The page includes a navigation menu with 'Products', 'Solutions', 'Support', and 'Contact'. Below this, there's a breadcrumb trail: 'Home > Products > Software > HW group > Hercules SETUP utility'. The main heading is 'Hercules SETUP utility'. The description states: 'Hercules SETUP utility is useful serial port terminal (RS-485 or RS-232 terminal), UDP/IP terminal and TCP/IP Client Server terminal. It was created for HW group internal use only, but today it's includes many functions in one utility and it's Freeware! With our original devices (Serial/Ethernet Converter, RS-232/Ethernet Buffer or I/O Controller) it can be used for the UDP Config.' The 'Licence type' is 'Freeware' and the 'SW version' is 'Hercules'. Under 'Features:', it lists: 'Complete support for Windows 7, 8, 8.1 and 10' and 'All basic TCP and UDP utilities in one file, no installation required [just'. On the right, there's an image of the software interface. At the bottom right, a blue button with a download icon and the text 'Download latest version of Hercules ver. 3.2.8' is highlighted with a red border.

hw-group.com/software/hercules-setup-utility

HW group

EN | CS

Products Solutions Support Contact

Devices Sensors Accessories Software Canceled

Home > Products > Software > HW group > Hercules SETUP utility

Hercules SETUP utility

Hercules SETUP utility is useful serial port terminal (RS-485 or RS-232 terminal), UDP/IP terminal and TCP/IP Client Server terminal. It was created for HW group internal use only, but today it's includes many functions in one utility and it's Freeware! With our original devices (Serial/Ethernet Converter, RS-232/Ethernet Buffer or I/O Controller) it can be used for the UDP Config.

Licence type: Freeware

SW version: Hercules

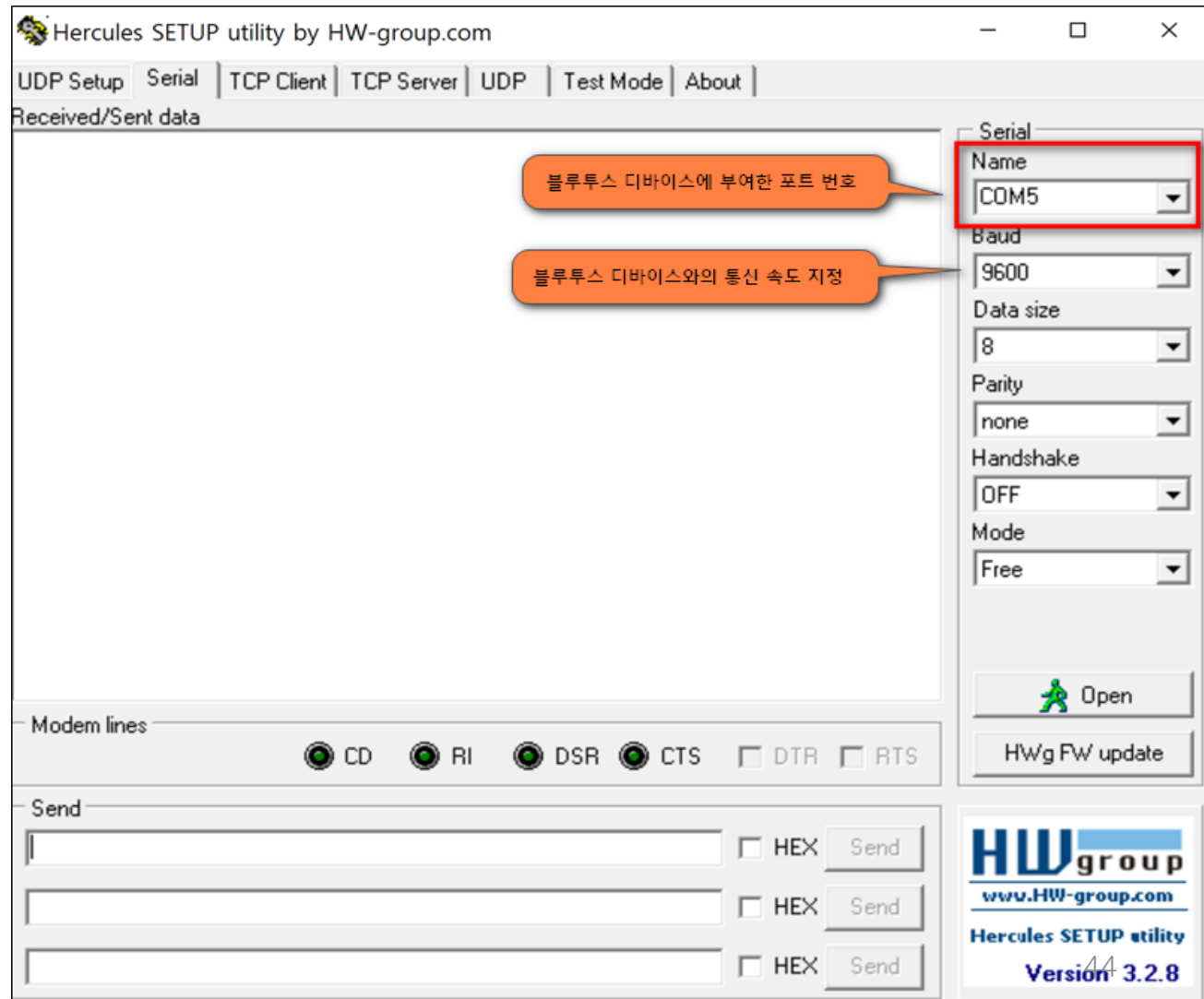
Features:

- Complete support for Windows 7, 8, 8.1 and 10
- All basic TCP and UDP utilities in one file, no installation required [just

Download latest version of Hercules ver. 3.2.8

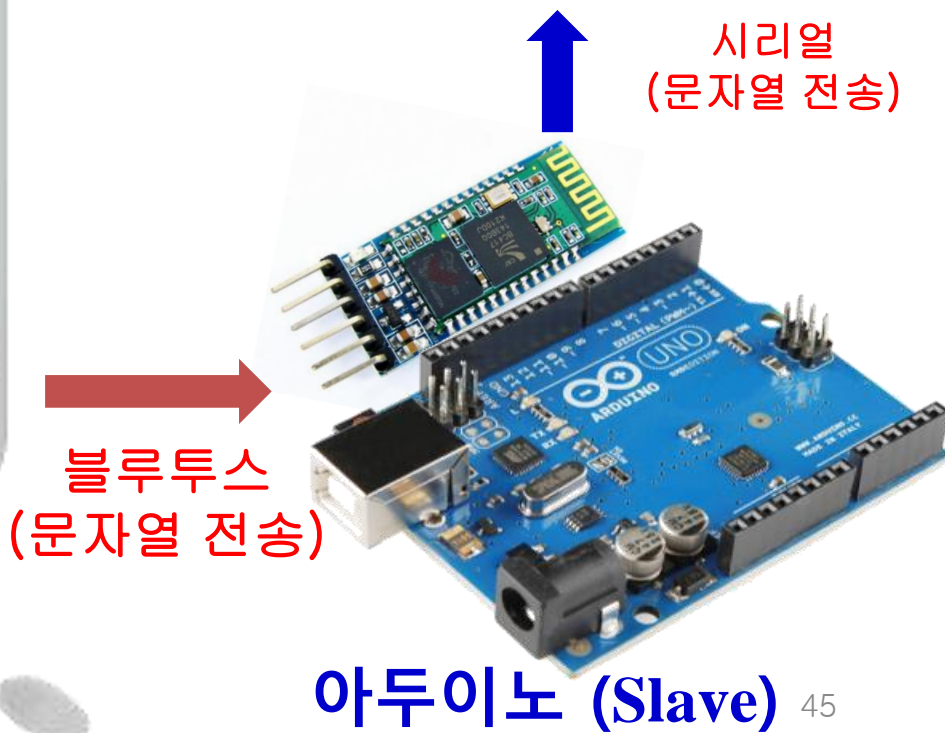
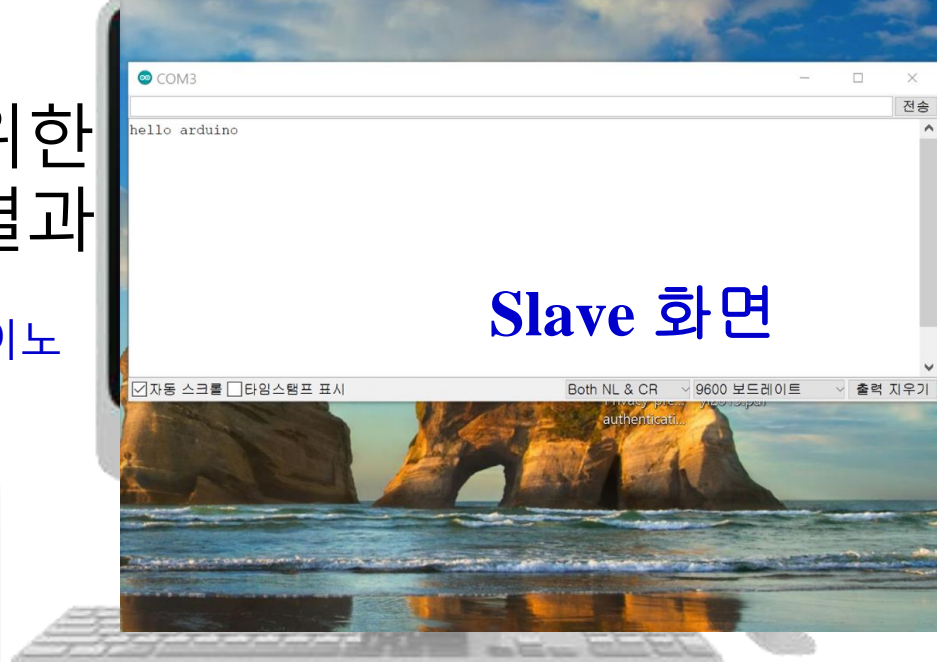
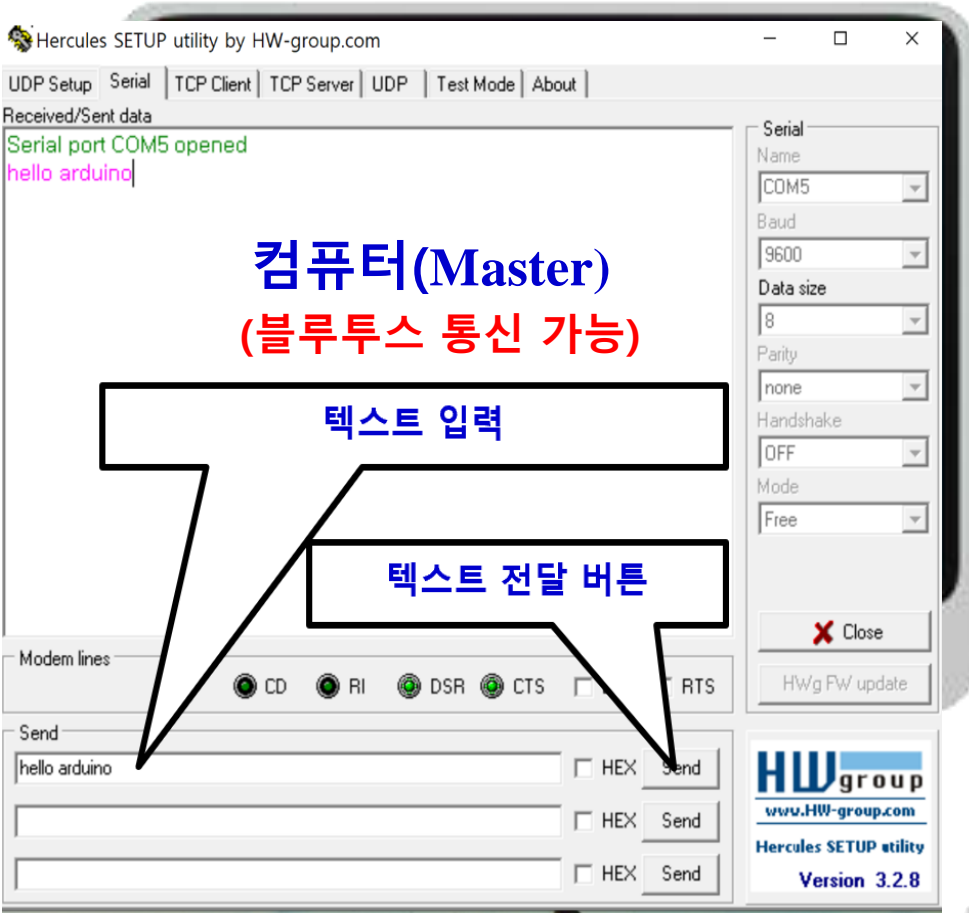
근거리 블루투스 무선 통신을 위한 컴퓨터 블루투스 Master 연결(Binding)

- 컴퓨터에서 Hercules
SETUP utility 실행
 - 포트 번호 설정
 - 아두이노 블루투스
Slave와 연결속도 설정



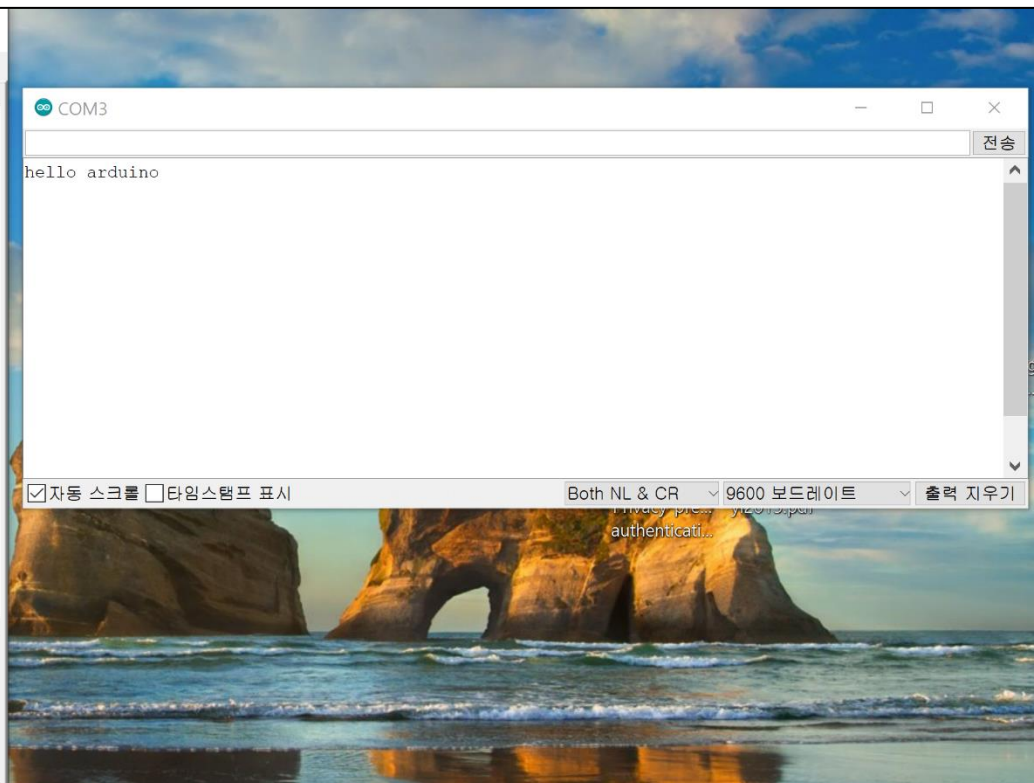
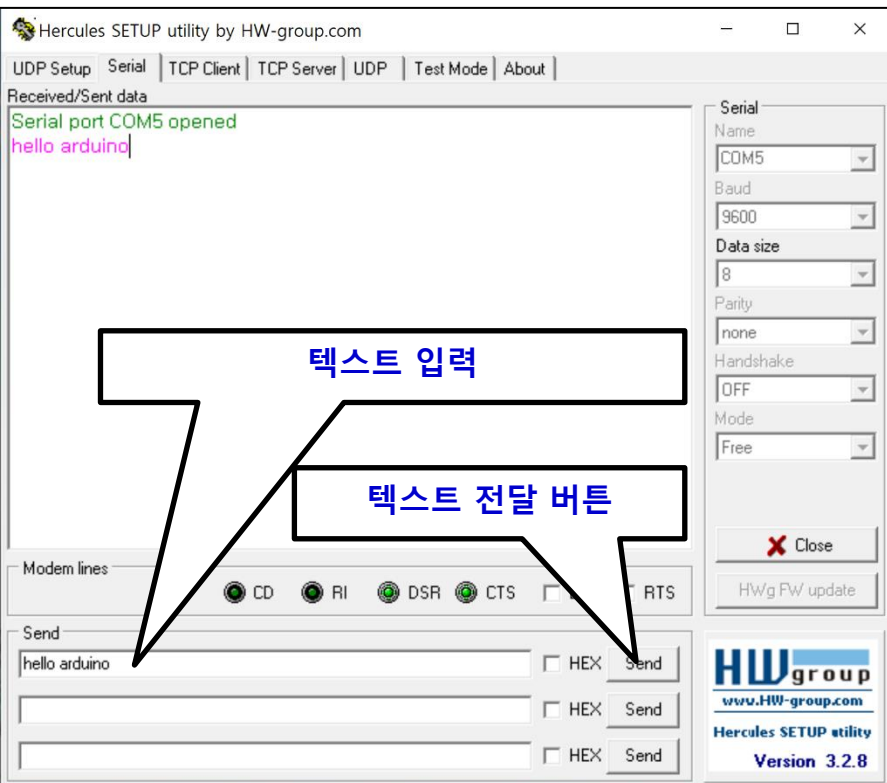
근거리 블루투스 무선 통신을 위한 컴퓨터와 아두이노 통신 실습 결과

- 블루투스 무선 통신 기반 컴퓨터(Master)와 아두이노(Slave) 연결 및 메시지 전달 결과



근거리 블루투스 무선 통신을 위한 컴퓨터와 아두이노 통신 실습 결과

- 블루투스 무선 통신 기반 컴퓨터(Master)와 아두이노(Slave) 연결 및 메시지 전달 결과

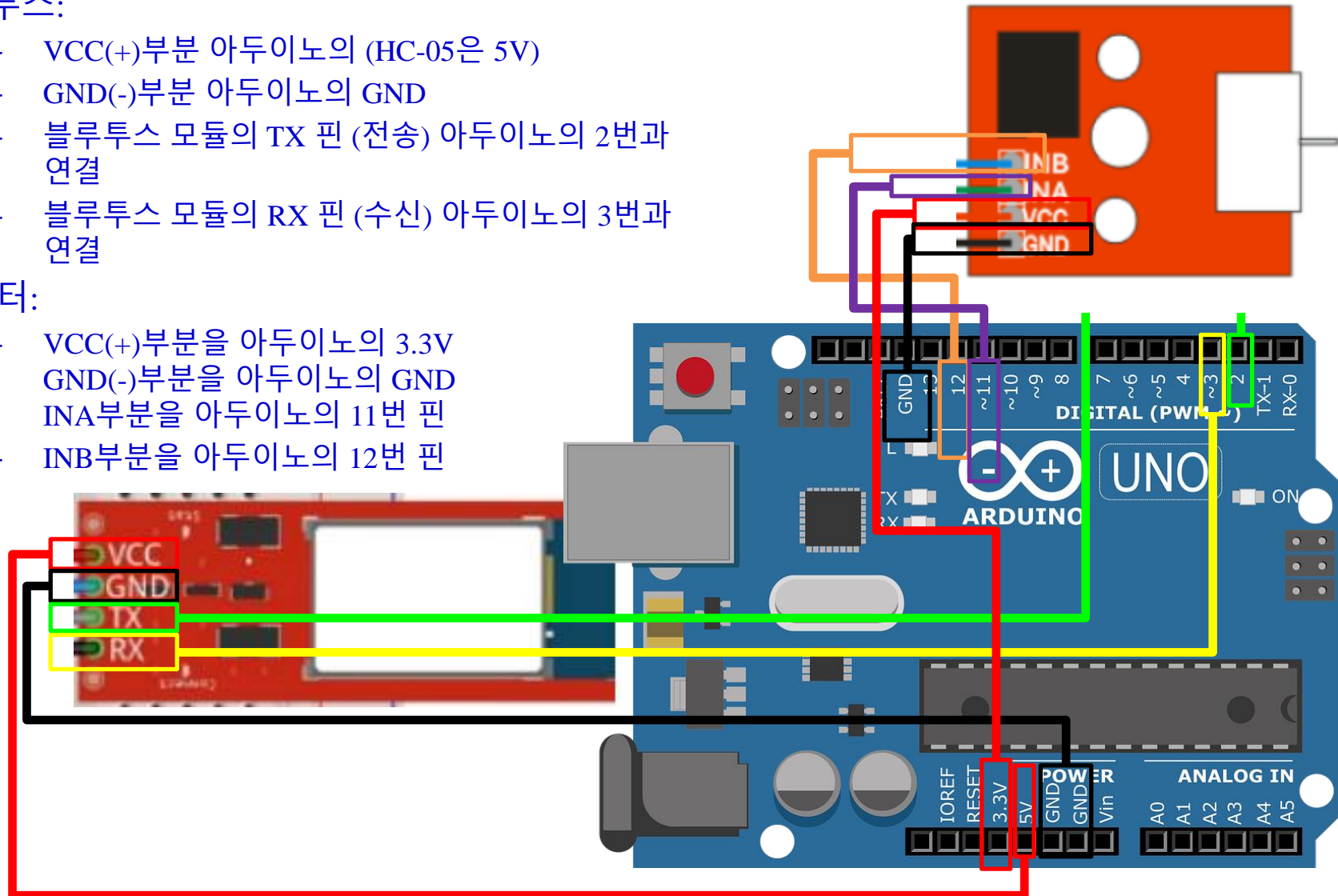


근거리 블루투스 무선 통신 기반 컴퓨터와 아두이노 DC 모터 팬 제어 추가 실습

- **컴퓨터와 아두이노 DC 모터 팬 제어 하드웨어 구성**
 - 컴퓨터 :블루투스 마스터
 - 아두이노 : 블루투스 Slave, DC 모터 팬
- **아두이노 DC 모터 팬 제어 실습 내용**
 - 컴퓨터 블루투스 마스터에서 명령어를 Slave로 전달.
 - 컴퓨터 블루투스 마스터에서 명령어 ‘o’를 전달하면 Slave에서 팬 실행.
 - 컴퓨터 블루투스 마스터에서 명령어 ‘s’를 전달하면 Slave에서 팬 정지.

근거리 블루투스 무선 통신 기반 아두이노 Slave DC 모터 팬과 블루투스 회로 구성

- 블루투스:
 - VCC(+)부분 아두이노의 (HC-05은 5V)
 - GND(-)부분 아두이노의 GND
 - 블루투스 모듈의 TX 핀 (전송) 아두이노의 2번과 연결
 - 블루투스 모듈의 RX 핀 (수신) 아두이노의 3번과 연결
- 팬 모터:
 - VCC(+)부분을 아두이노의 3.3V
 - GND(-)부분을 아두이노의 GND
 - INA부분을 아두이노의 11번 핀
 - INB부분을 아두이노의 12번 핀



근거리 블루투스 무선 통신 기반 아두이노 DC 모터 팬 제어 (Slave 소스 프로그램)

```
#include <SoftwareSerial.h>
int t = 2;
int r = 3;
int ina = 11;
int inb = 12;

SoftwareSerial ble(t,r);

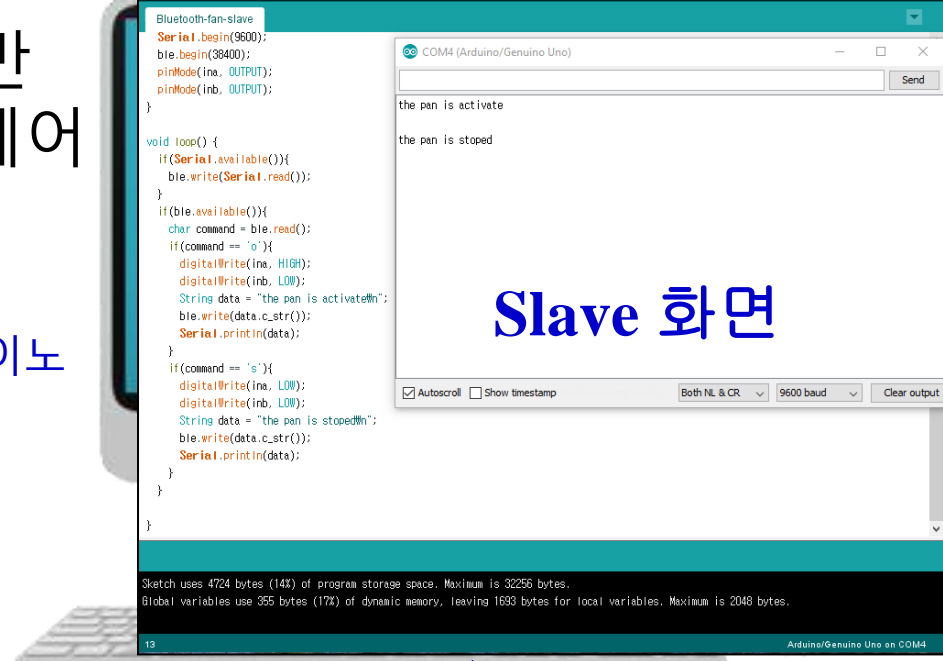
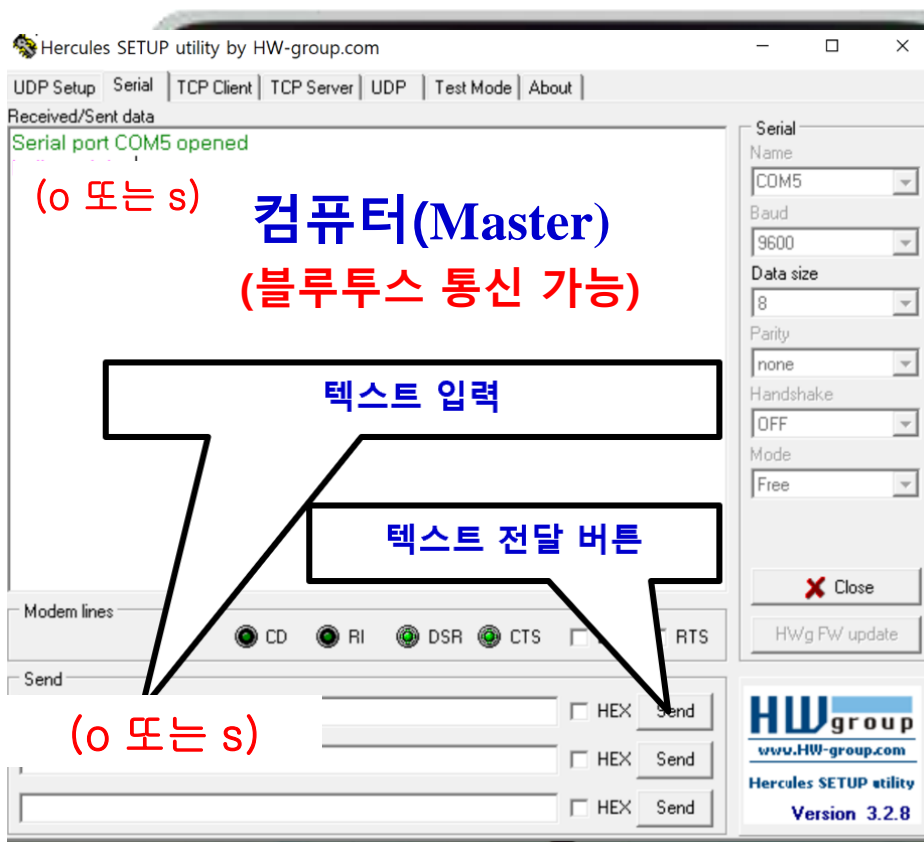
void setup() {
  Serial.begin(9600);
  ble.begin(38400);
  pinMode(ina, OUTPUT);
  pinMode(inb, OUTPUT);
}

void loop() {
  if(Serial.available()){
    ble.write(Serial.read());
  }
  if(ble.available()){
    char command = ble.read(); //블루투스에서 전달된 값을 읽어 드림
    if(command == 'o'){ // char 값이 o 이면
      digitalWrite(ina, HIGH); //팬을 작동하도록 하나의 pin에는 high 하나의 pin에는 low로 출력 함
      digitalWrite(inb, LOW);
      String data = "the pan is activate\n"; // 문자열 생성
      ble.write(data.c_str()); //블루투스를 통하여 마스터로 문자열 전달
      Serial.println(data); //시리얼로 문자열 전달
    }
    if(command == 's'){ // char 값이 s 이면
      digitalWrite(ina, LOW); //팬을 작동 하지 않도록 모든 pin에서 LOW값을 출력
      digitalWrite(inb, LOW);
      String data = "the pan is stoped\n";
      ble.write(data.c_str());
      Serial.println(data);
    }
  }
}
```

- 제대로 작도이 되지 않으면 수동으로 9600으로 변환 권장
 - ble.begin(38400);

근거리 블루투스 무선 통신 기반 컴퓨터와 아두이노 DC 모터 팬 제어 실습 결과

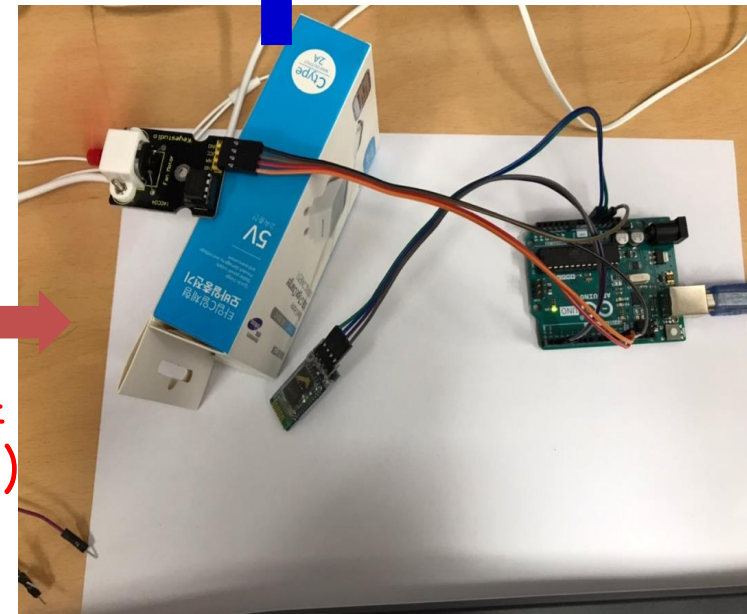
- 블루투스 무선 통신 기반 컴퓨터(Master)와 아두이노 (Slave) 연결 및 메시지 전달 결과



Slave 화면

시리얼(문자열 전송)

블루투스 (o 또는 s)

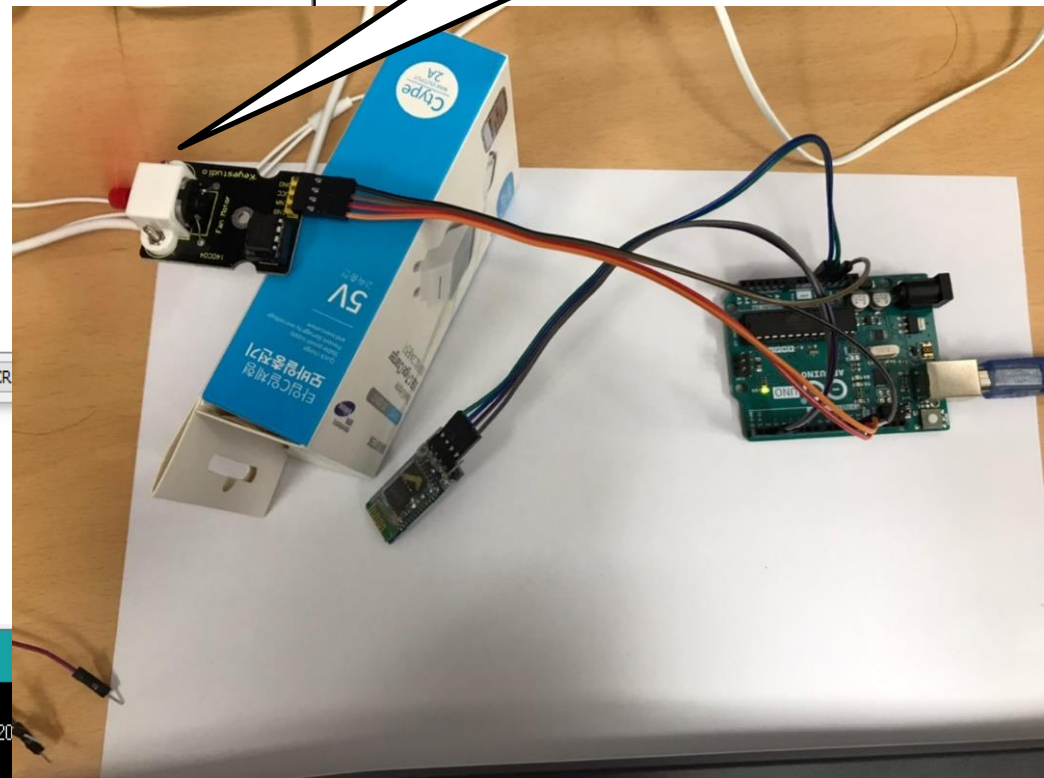
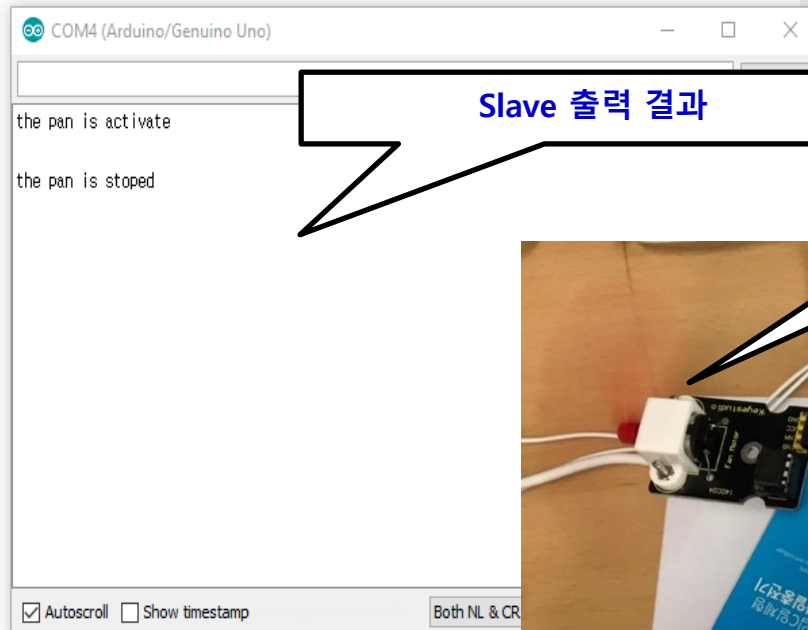


아두이노 (Slave)¹⁰

근거리 블루투스 무선 통신 기반 컴퓨터와 아두이노 DC 모터 팬 제어 실습 결과

```
Bluetooth-fan-slave
Serial.begin(9600);
ble.begin(38400);
pinMode(ina, OUTPUT);
pinMode(inb, OUTPUT);
}

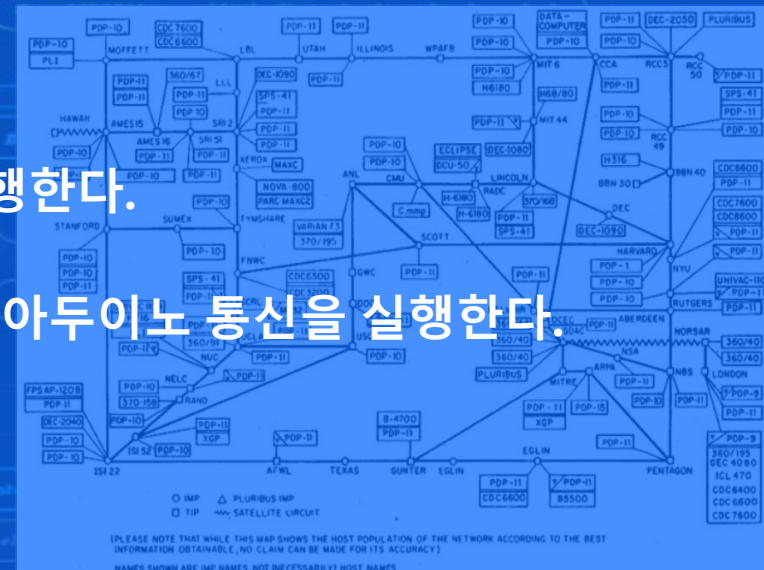
void loop() {
  if(Serial.available()){
    ble.write(Serial.read());
  }
  if(ble.available()){
    char command = ble.read();
    if(command == 'o'){
      digitalWrite(ina, HIGH);
      digitalWrite(inb, LOW);
      String data = "the pan is activate#n";
      ble.write(data.c_str());
      Serial.println(data);
    }
    if(command == 's'){
      digitalWrite(ina, LOW);
      digitalWrite(inb, LOW);
      String data = "the pan is stoped#n";
      ble.write(data.c_str());
      Serial.println(data);
    }
  }
}
```



Sketch uses 4724 bytes (14%) of program storage space. Maximum is 32256 bytes.
Global variables use 355 bytes (17%) of dynamic memory, leaving 1693 bytes for local variables. Maximum is 2048 bytes.

학습 내용 정리

- 근거리 블루투스 무선 통신을 위한 아두이노 하드웨어 구성에 대해 알아본다.
- 근거리 블루투스 기반 아두이노와 컴퓨터 통신 실습 환경에 대해 살펴본다.
- 아두이노 블루투스 Slave 설정(Command)을 수행한다.
- 아두이노 블루투스 Slave 연결(Binding)을 수행한다.
- 근거리 블루투스 무선 통신을 위한 아두이노 스케치(프로그램) 작성하고 컴파일한다.
- 컴퓨터 블루투스 Master 연결(Binding)을 수행한다.
- 근거리 블루투스 무선 통신을 위한 컴퓨터와 아두이노 통신을 실행한다.





Thank you for listening

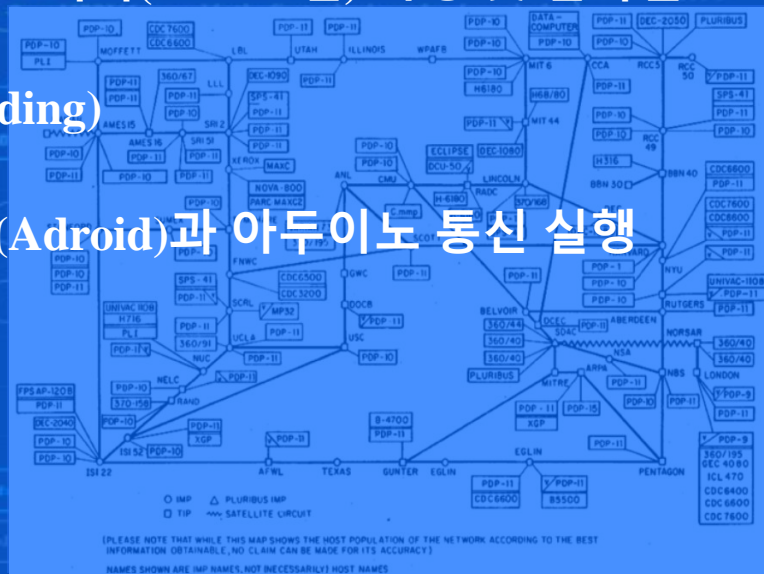
근거리 블루투스 무선 통신 기반 아두이노와 스마트 폰 통신 환경 구축 및 실습

제주대학교

김도현

주요 학습 내용

- 근거리 블루투스 무선 통신을 위한 아두이노 하드웨어 구성
- 근거리 블루투스 기반 아두이노와 스마트 폰(Android) 통신 실습 환경
- 아두이노 블루투스 Slave 설정(Command)
- 아두이노 블루투스 Slave 연결(Binding)
- 근거리 블루투스 무선 통신을 위한 아두이노 스케치(프로그램) 작성 및 컴파일
- 스마트 폰(Android) 블루투스 Master 연결(Binding)
- 근거리 블루투스 무선 통신을 위한 스마트 폰(Android)과 아두이노 통신 실행



근거리 블루투스 무선 통신을 위한 아두이노 구성

- 아두이노 하드웨어와 소프트웨어

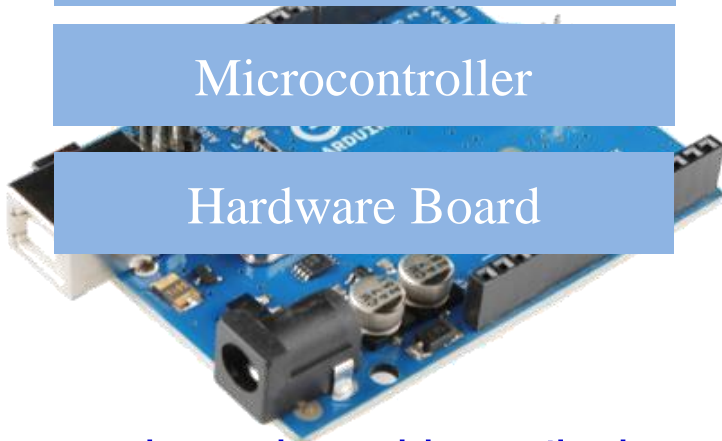


Bluetooth (HC-05)

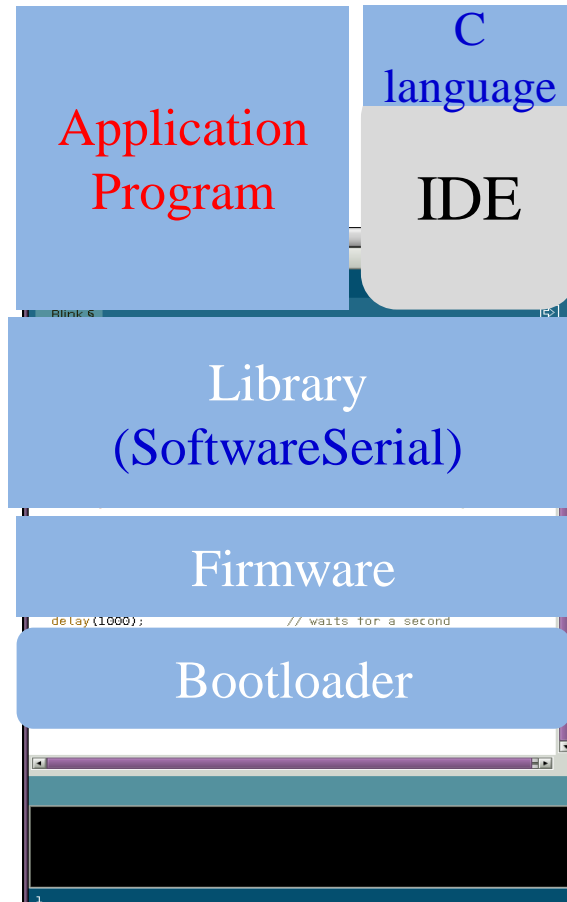
Digital I/O (2,3 pin)

Microcontroller

Hardware Board



아두이노 하드웨어



아두이노 소프트웨어

(IDE : Integrated Development Environment)

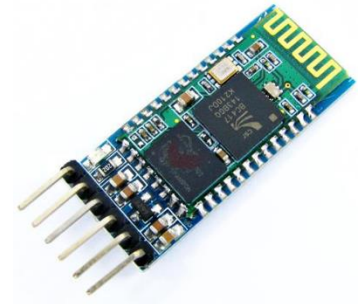
근거리 블루투스 기반 아두이노 환경 구성

- Layout of Arduino UNO

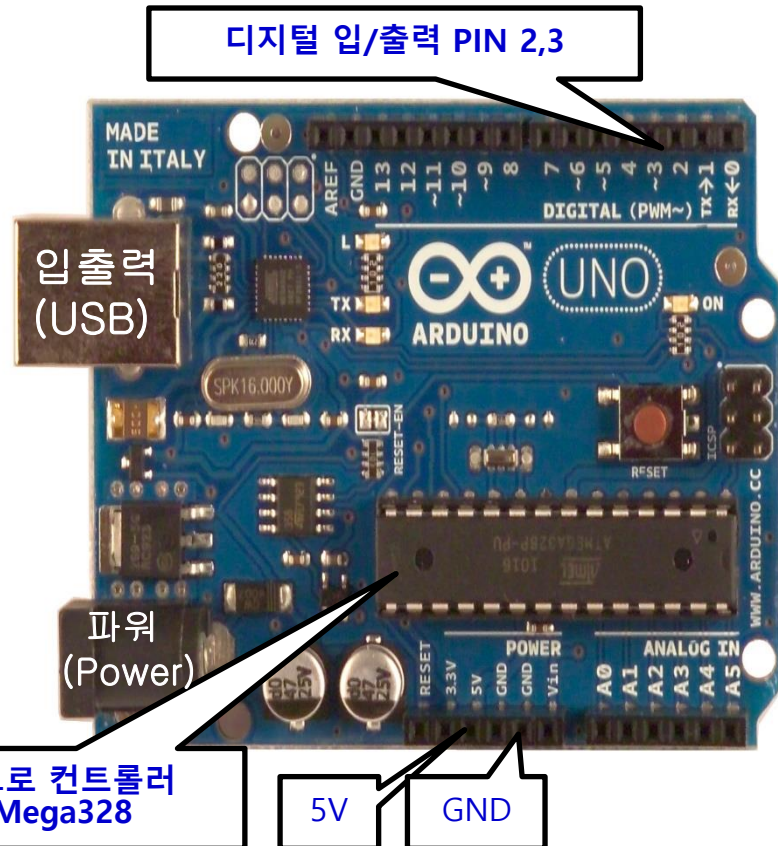
1. Digital Ground
3. Digital pins 2, 3
4. Digital pins Serial In/Out
5. Reset Button
6. In-circuit Serial Programmer
7. 8. Power and Ground pins
9. External Power Supply In
10. USB port



Jumper Wire
(Various Colors)



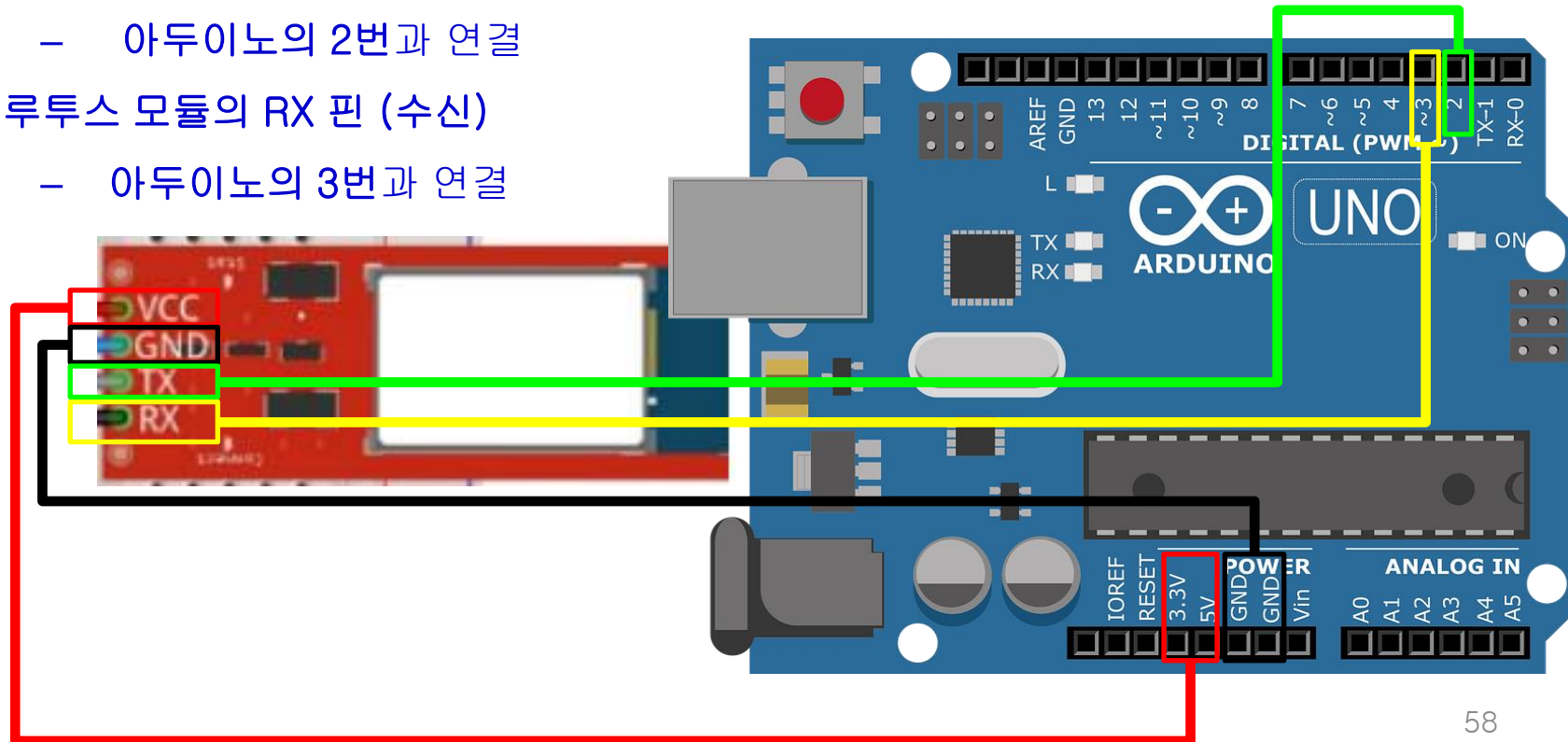
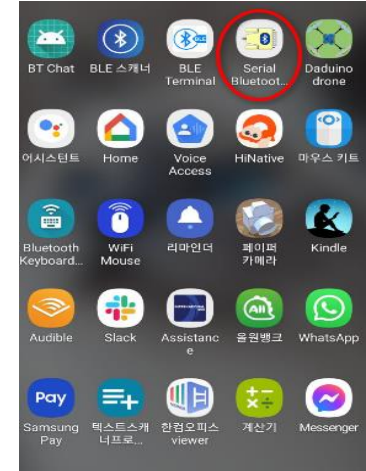
Bluetooth Module
(HC-05)



근거리 블루투스 무선 통신을 위한 아두이노 회로 구성

- VCC(+)부분
 - 아두이노의 (HC-05은 5V)
- GND(-)부분
 - 아두이노의 GND
- 블루투스 모듈의 TX 핀 (전송)
 - 아두이노의 2번과 연결
- 블루투스 모듈의 RX 핀 (수신)
 - 아두이노의 3번과 연결

스마트 폰
(Adroid)



근거리 블루투스 기반 아두이노와 스마트폰 통신 실습 환경

스마트 폰(Android)

- 아두이노 하드웨어와 스마트폰(Android)간의 블루투스

통신 환경

아두이노 (Arduino)

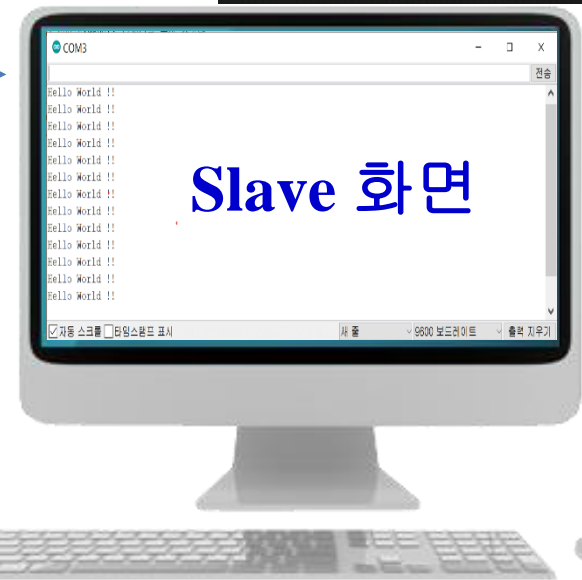
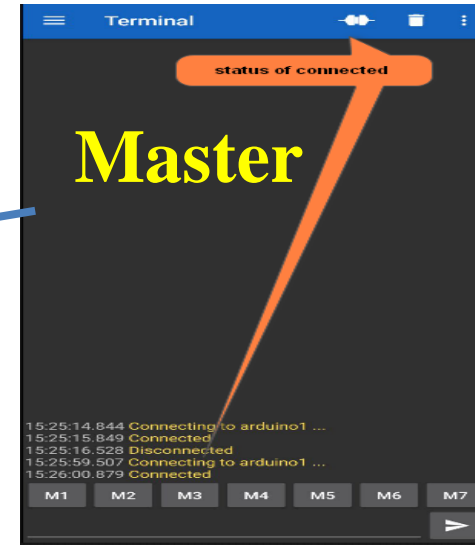
1 Word

블루투스 (문자열 전송)

1 Word

시리얼 (문자열 전송)

Slave

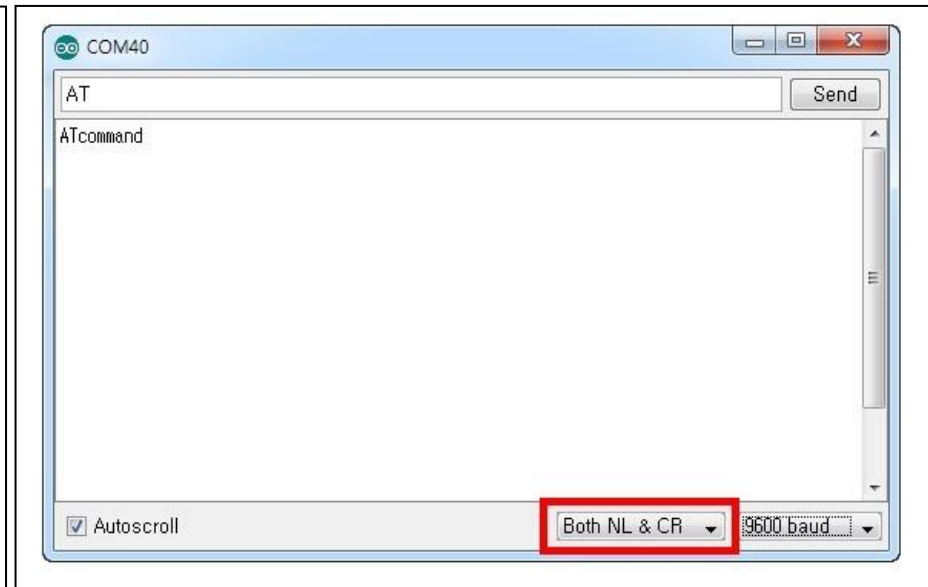
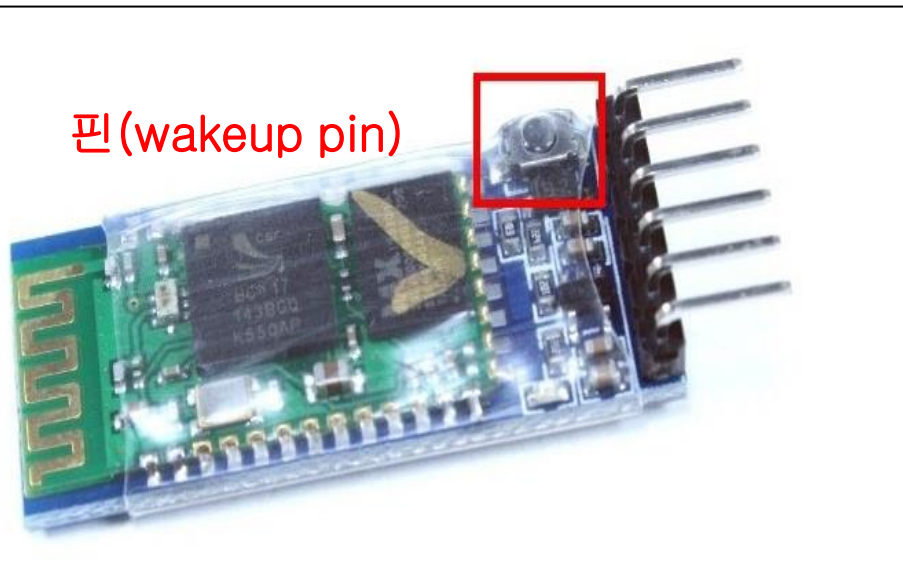


근거리 블루투스 기반 스마트 폰(Android) 통신을 위한 아두이노 Slave 설정(Command)과 연결(Binding)

1. Slave 디바이스 설정(Command) 모드(mode)로 변환 및
Serial monitor 입력 형태 변환
2. 아두이노 블루투스 초기화
3. 아두이노 블루투스 Slave 비밀번호 확인
4. 아두이노 블루투스 Slave 이름 설정
5. 아두이노 블루투스 Slave 속도 설정
6. 아두이노 블루투스 설정(Command) 모드에서 연결(Binding) 모드
변환 완료

근거리 블루투스 기반 스마트 폰(Android) 통신을 위한 아두이노 Slave 설정 1

- 블루투스를 AT Master와 Slave 설정(Command) 모드(mode)로 변환
 - 아래의 핀(wakeup pin)을 눌러 준 상태에서 전원을 뽑았다 다시 꽂아 줌.
 - 정상적으로 AT Master와 Slave 설정 변환이 될 경우
 - 블루투스 모듈의 LED등의 빈도가 늦어 짐.
- 블루투스를 Serial monitor 입력 형태로 변환.

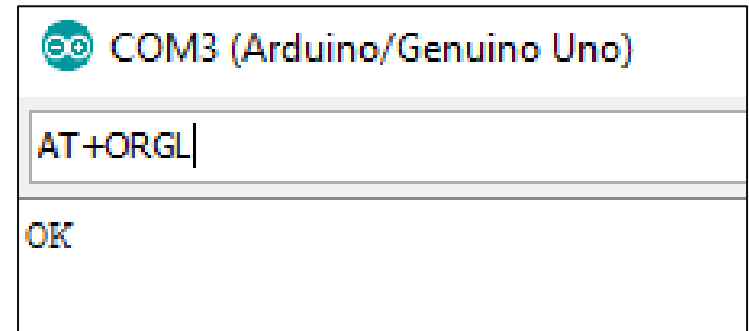


근거리 블루투스 기반 스마트 폰(Android) 통신을 위한 아두이노 Slave 설정 2

블루투스 설정

2. 블루투스 초기화

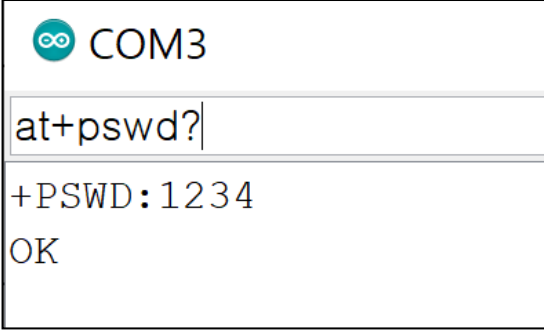
- 디폴트 상태로 설정 하기 위해 시리얼 모니터에서 AT+ORGL 명령어를 입력한다.
- Module work mode: Slave Mode
- Connection mode : connect to the specified device
- Serial parameter : Baud rate = 38400,0,0
- Specified device : 0:0:0



근거리 블루투스 기반 스마트 폰(Android) 통신을 위한 아두이노 Slave 설정 3

- 아두이노 블루투스 Slave 비밀번호 확인

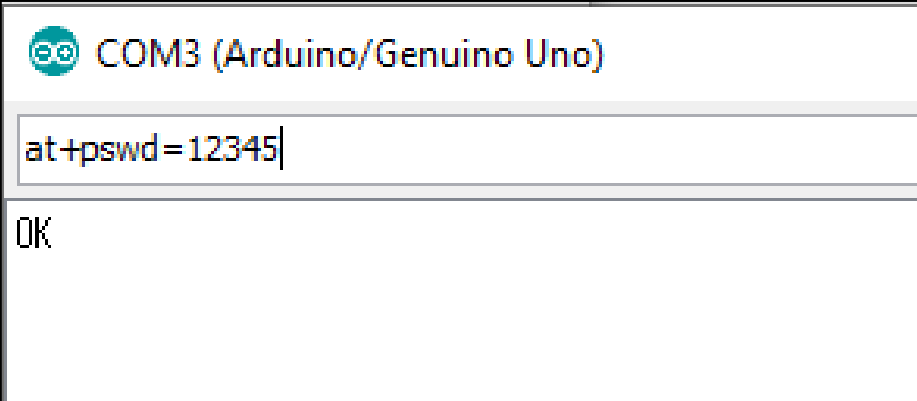
- 비밀번호 확인
- at+pswd?



COM3
at+pswd?
+PSWD:1234
OK

- 비밀번호 설정

- at+pswd=1234



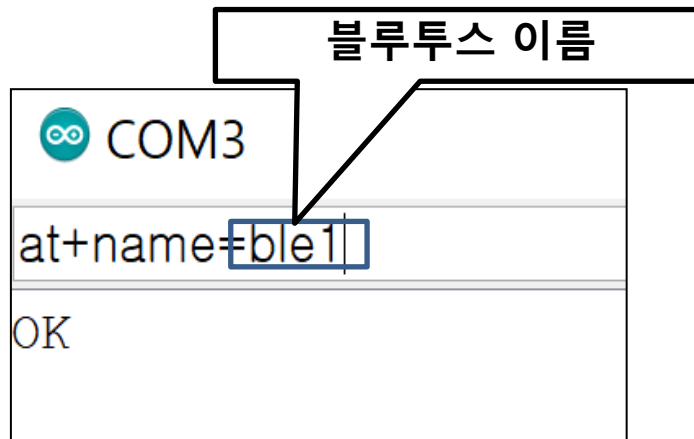
COM3 (Arduino/Genuino Uno)
at+pswd=12345
OK

** 디폴트 상태로 비밀번호가 설정

근거리 블루투스 기반 스마트 폰(Android) 통신을 위한 아두이노 Slave 설정 4

- 아두이노 블루투스 Slave 이름 설정

- 블루투스 이름 설정: 컴퓨터에서 연결할 아두이노 블루투스 디바이스 이름 설정
- `at+name=ble1`



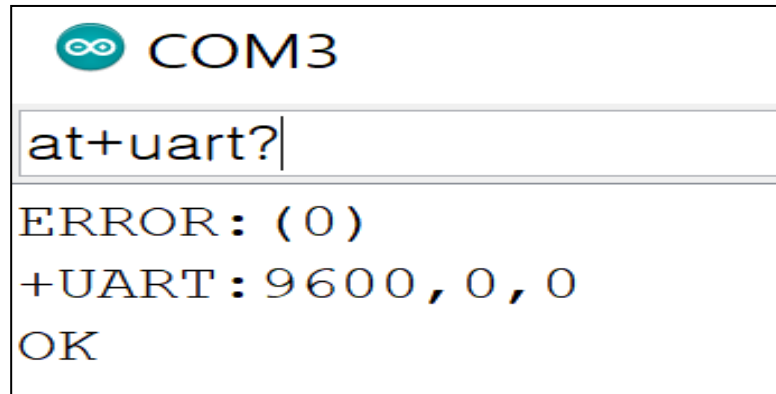
근거리 블루투스 기반 스마트 폰(Android) 통신을 위한 아두이노 Slave 설정 5

- 아두이노 블루투스 Slave 속도 설정

- 전송 속도 설정: 폰에서 사용하는 전송 속도와 맞추는 작업을 진행.

- 블루투스 속도 확인

- at+uart?

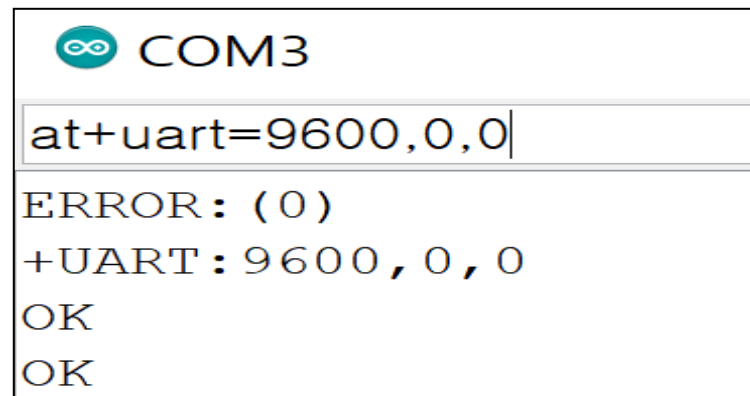


COM3

```
at+uart?  
ERROR: (0)  
+UART: 9600, 0, 0  
OK
```

- 블루투스 속도 설정

- at+uart=9600,0,0

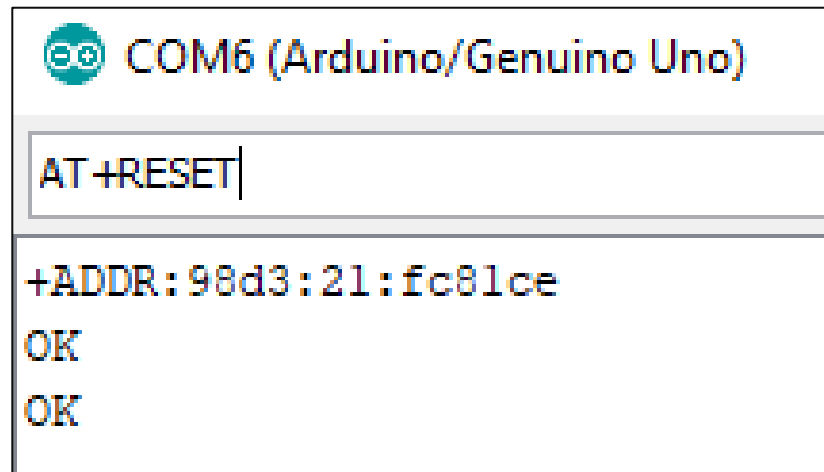


COM3

```
at+uart=9600,0,0  
ERROR: (0)  
+UART: 9600, 0, 0  
OK  
OK
```

근거리 블루투스 기반 스마트 폰(Android) 통신을 위한 아두이노 Slave 연 결 모드 6

- 아두이노 블루투스 설정(Command) 모드에서 연결(Binding) 모드 변환 완료
 - 아두이노를 AT 명령어 모드에서 연결 모드로 변환
 - 컴퓨터에서 아두이노와 블루투스 연결 진행.
 - at+reset



```
COM6 (Arduino/Genuino Uno)

AT+RESET|

+ADDR:98d3:21:fc81ce
OK
OK
```

근거리 블루투스 무선 통신을 위한 아두이노 Slaver 스케치(프로그램)

- 스마트 폰(Adroid)과 블루투스 실습을 위해 데이터 전송 속도를 38400에서 9600으로 수정 권고

```
#include <SoftwareSerial.h>
```

```
int T = 2;
```

```
int R = 3;
```

// 블루투스 통신을 사용하기 위해 불러오는 라이브러리

// 블루투스 모듈의 TXD 핀과 연결된 2번을 T 라는 변수 선언

// 블루투스 모듈의 RXD 핀과 연결된 3번을 R 이라는 변수 선언

```
SoftwareSerial my_blue(T, R);
```

// my_blue 라는 이름의 블루투스 객체 선언

// 이 블루투스 객체는 T 와 R 에 의해 제어

```
void setup(){
```

```
  Serial.begin(9600);
```

```
  my_blue.begin(9600);
```

```
}
```

// 시리얼 통신을 9600 의 통신속도로 시작한다.

// 블루투스 통신을 시작한다.

```
void loop(){
```

```
  if (my_blue.available()) {           //블루투스를 사용할 수 있다면
```

```
    Serial.write(my_blue.read()); //값을 읽어서 시리얼에 쓰기
```

```
  }
```

```
  if (Serial.available()) {           //시리얼을 사용할 수 있다면
```

```
    my_blue.write(Serial.read()); //값을 읽어 블루투스에 쓰기
```

```
  }
```

```
}
```

파일 편집 스케치 툴 도움말



ble-exam

```
#include <SoftwareSerial.h>
```

```
SoftwareSerial ble(2,3);
```

```
void setup() {
```

```
  Serial.begin(9600);
```

```
  ble.begin(9600);
```

```
}
```

```
void loop() {
```

```
  if (ble.available()) {
```

```
    Serial.write(ble.read());
```

```
  }
```

```
  if (Serial.available()) {
```

```
    ble.write(Serial.read());
```

```
  }
```

```
}
```

업로딩...

스케치는 프로그램 저장 공간 3196 바이트(9%)를 사용, 최대 32256 바이트.

전역 변수는 동적 메모리 301바이트(14%)를 사용, 1747바이트의 지역변수가 남음. 최대는 2048 바이트.

근거리 블루투스 무선 통신을 위한 아두이노 Slaver 스케치(프로그램) 실행

1. 새로운 스케치 생성

2. 컴파일

3. 업로드

4. 스케치(프로그램) 작성

파일 편집 스케치 툴 도움말



새 파일

2 3 1

파일 편집 스케치 툴 도움말



ble-exam

```
#include <SoftwareSerial.h>

SoftwareSerial ble(2,3);
void setup() {
  Serial.begin(9600);
  ble.begin(9600);
}
void loop() {
  if(ble.available()){
    Serial.write(ble.read());
  }
  if(Serial.available()){
    ble.write(Serial.read());
  }
}
```

4

업로딩...

스케치는 프로그램 저장 공간 3196 바이트(9%)를 사용. 최대 32256 바이트.
전역 변수는 동적 메모리 301바이트(14%)를 사용, 1747바이트의 지역변수가 남음. 최대는 2048 바이트.

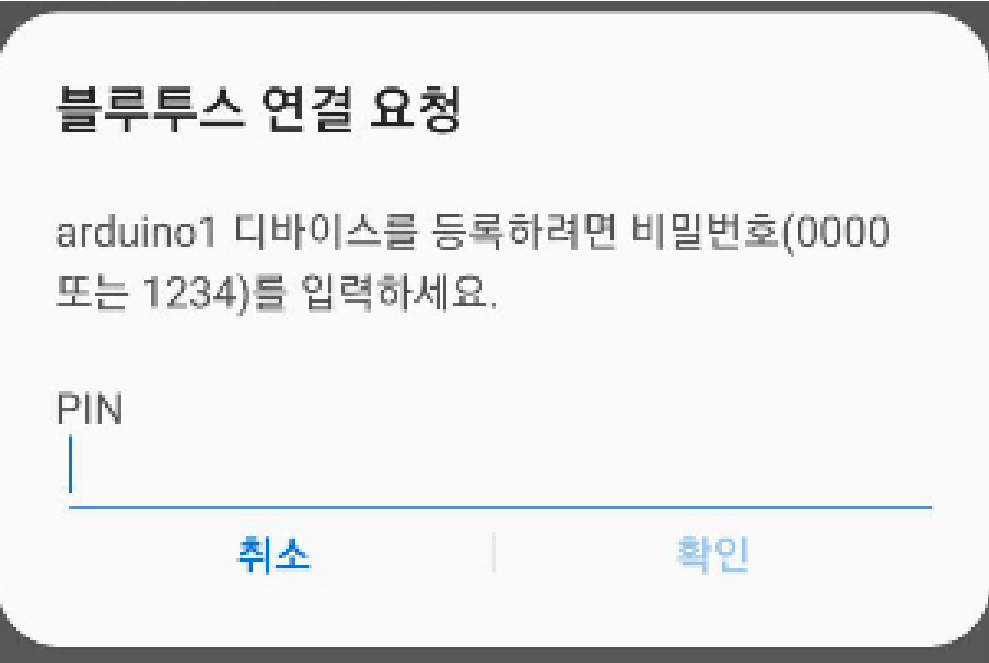
근거리 블루투스 무선 통신을 위한 스마트 폰(Android) 블루투스 Master 연결(Binding)

- 스마트 폰(Android) 블루투스 연결(블루투스 지원하는
스마트 폰(Android)에서).
 - 스마트 폰(Android) 블루투스 아두이노 검색 및 확인



근거리 블루투스 무선 통신을 위한 스마트 폰(Android) 블루투스 Master 연결(Binding)

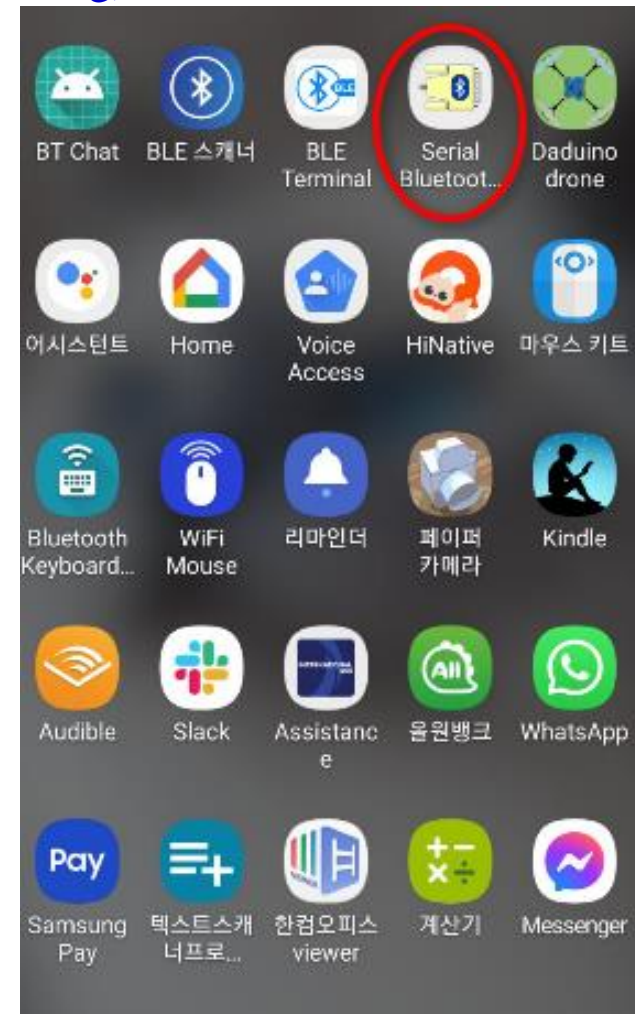
- 스마트 폰(Android) 블루투스 연결(블루투스 지원하는 스마트 폰(Android)에서).
 - 스마트 폰(Android)에 블루투스 아두이노 등록
 - 스마트 폰(Android) 선택한 아두이노 블루투스 디바이스 비밀번호 입력.



A screenshot of a Bluetooth connection request dialog box on an Android device. The dialog has a white background with rounded corners and a dark border. At the top, the title "블루투스 연결 요청" (Bluetooth Connection Request) is displayed in bold black text. Below the title, a message in black text reads: "arduino1 디바이스를 등록하려면 비밀번호(0000 또는 1234)를 입력하세요." (To register the arduino1 device, enter the password (0000 or 1234)). Underneath the message, the label "PIN" is positioned above a text input field. The input field contains a vertical cursor line. At the bottom of the dialog, there are two buttons: "취소" (Cancel) on the left and "확인" (Confirm) on the right, both in blue text.

근거리 블루투스 무선 통신을 위한 스마트 폰(Android) 블루투스 Master 연결(Binding)

- 스마트 폰(Android) 블루투스 Master 연결(Binding) 공개 소프트웨어
 - Serial Bluetooth Terminal 앱 실행 및 테스트



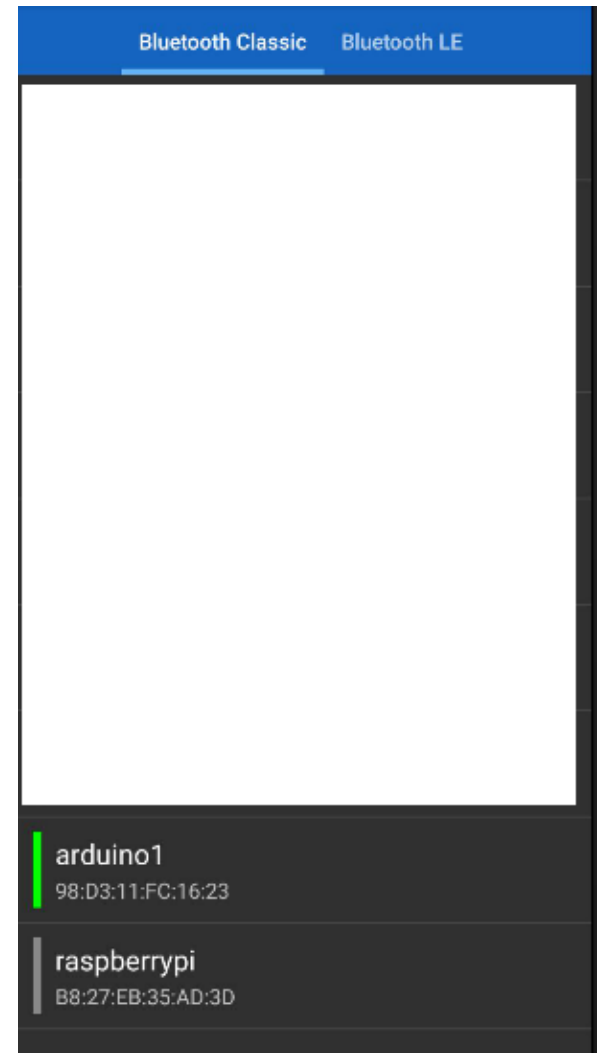
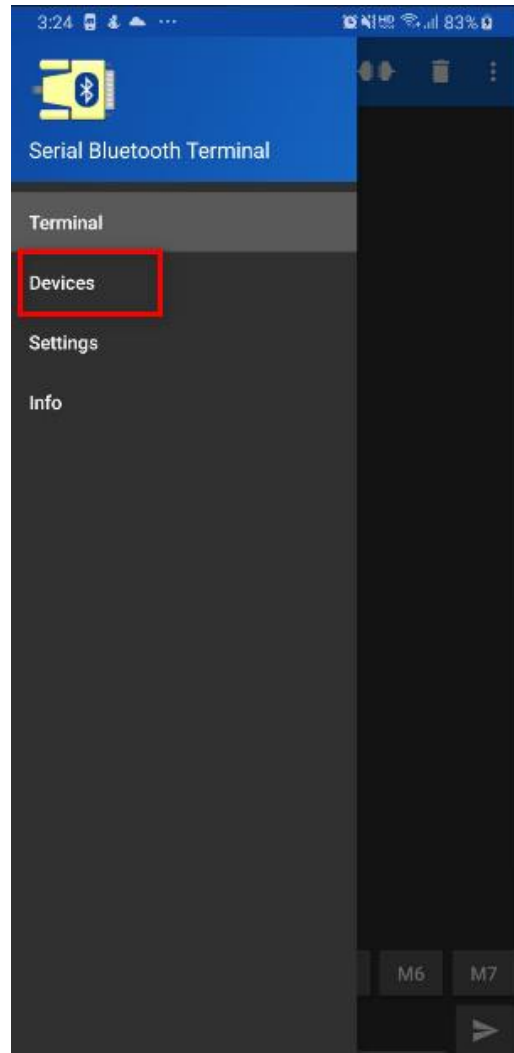
근거리 블루투스 무선 통신을 위한 스마트 폰(Android) 블루투스 Master 연결

- 스마트 폰(Android)에서

Serial Bluetooth

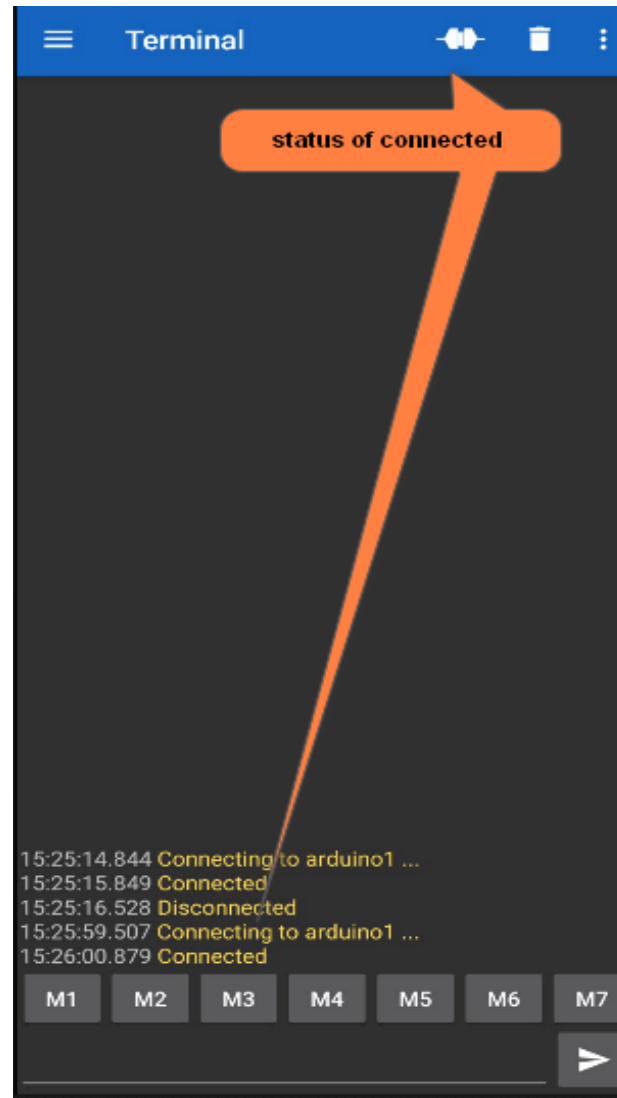
Terminal 실행

- 앱 실행화면에서 디바이스(Devices) 리스트 확인
- Device 리스트에서 블루투스 아두이노 선택



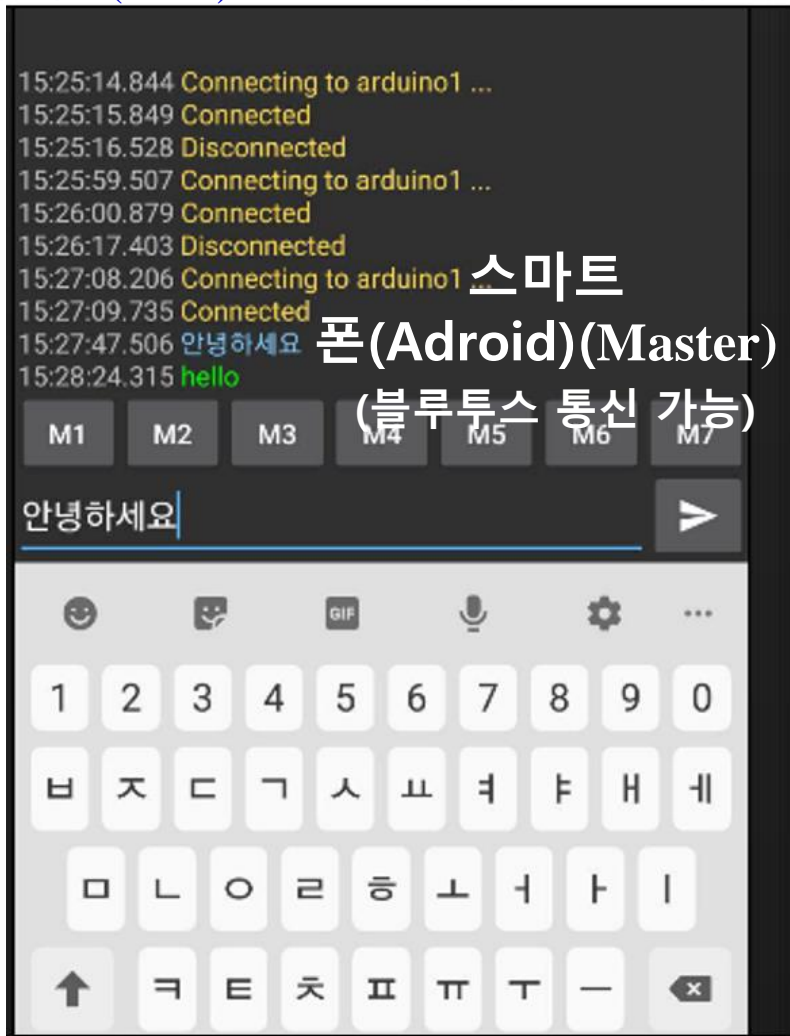
근거리 블루투스 무선 통신을 위한 스마트 폰(Android) 블루투스 Master 연결

- 스마트 폰(Android)에서 Serial Bluetooth Terminal 앱에서
 - 연결 상태 확인 및 메시지 전송

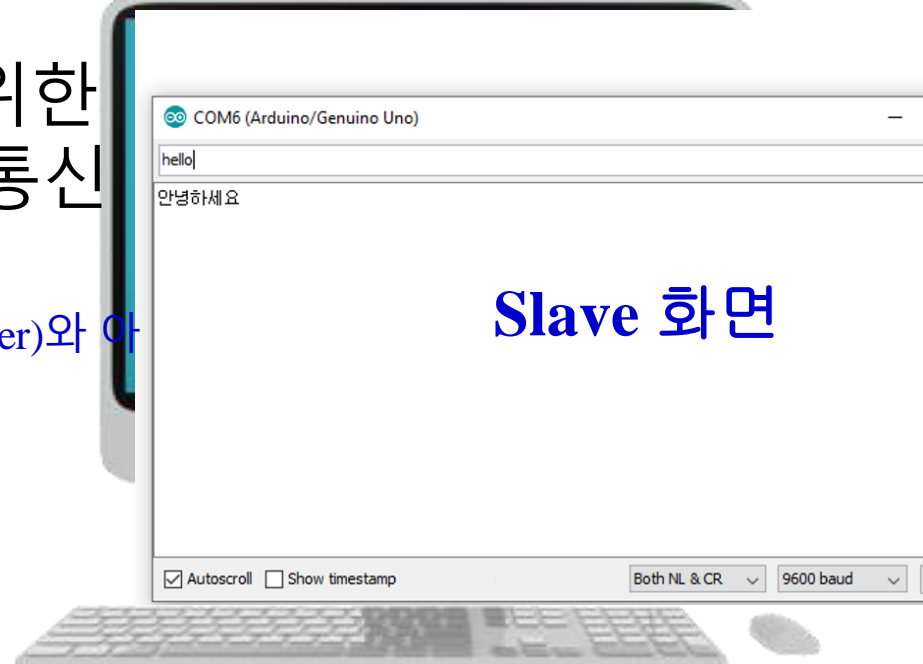


근거리 블루투스 무선 통신을 위한 스마트 폰(Android)과 아두이노 통신 실습 결과

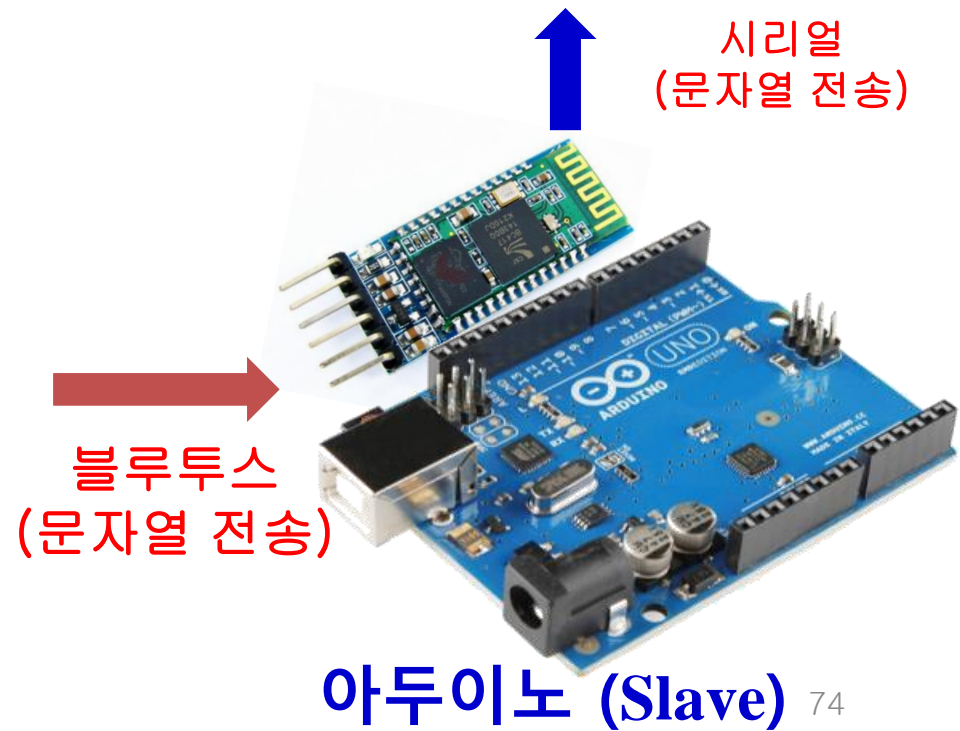
- 블루투스 무선 통신 기반 스마트 폰(Android)(Master)와 아두이노(Slave) 연결 및 메시지 전달 결과



스마트
폰(Android)(Master)
(블루투스 통신 가능)



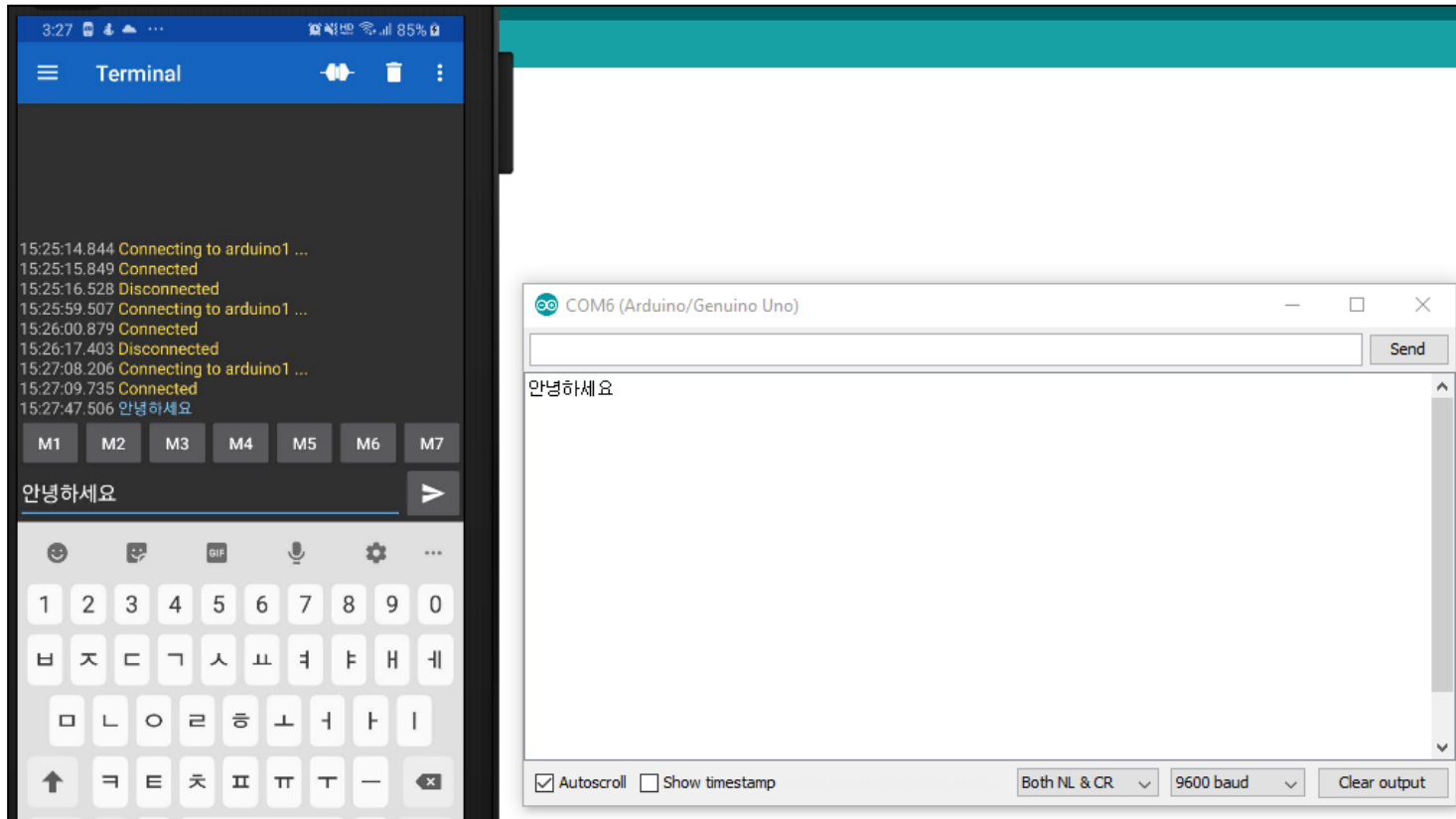
Slave 화면



아두이노 (Slave) 74

근거리 블루투스 무선 통신을 위한 스마트 폰(Android) 블루투스 Master 연결

- 블루투스 무선 통신 기반 스마트 폰(Android)(Master)와 아두이노(Slave) 연결
및 메시지 전달 결과



근거리 블루투스 무선 통신 기반 스마트 폰(Android)과 아두이노 조도 데이터 수집추가 실습

- 스마트 폰(Android)과 아두이노 조도 데이터 수집 하드웨어 구성
 - 스마트 폰(Android) :블루투스 마스터
 - 아두이노 : 블루투스 Slave, 아두이노, 조도 센서, LED(13번 핀)
- 아두이노 조도 데이터 수집 실습 내용
 - 스마트 폰(Android) 블루투스 마스터에서 명령어를 Slave로 전달.
 - 스마트 폰(Android) 블루투스 마스터에서 명령어 ‘o’를 전달하면 Slave에서 조도를 측정하여 어두우면 LED ON 반대로 밝으면 LED OFF 를 하는 자동 LED 기능 수행.
 - 스마트 폰(Android) 블루투스 마스터에서 명령어 ‘s’를 전달하면 Slave에서 자동 LED 기능 정지.

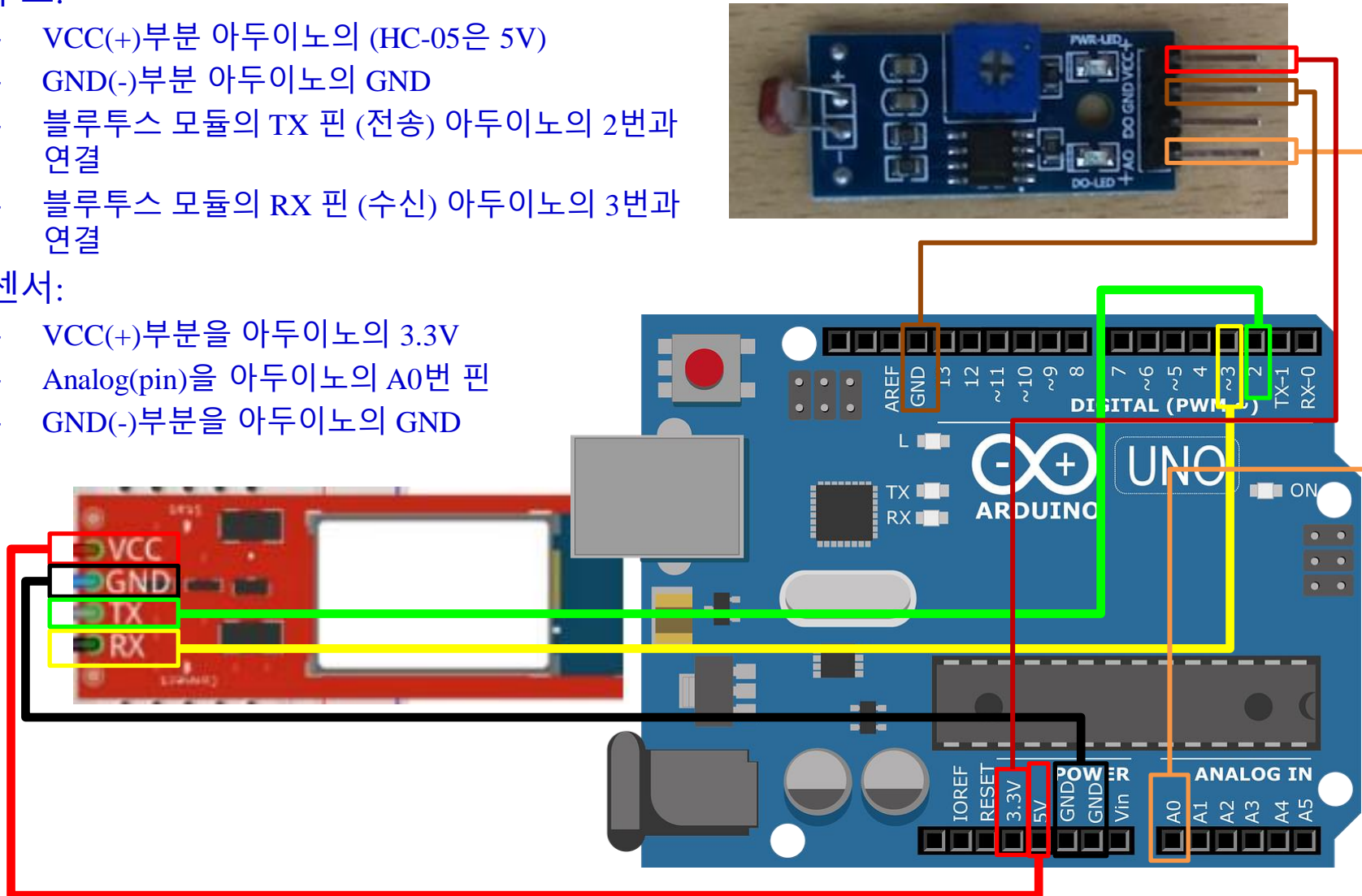
근거리 블루투스 무선 통신 기반 아두이노 Slave 조도 센서과 블루투스 회로 구성

- 블루투스:

- VCC(+)부분 아두이노의 (HC-05은 5V)
- GND(-)부분 아두이노의 GND
- 블루투스 모듈의 TX 핀 (전송) 아두이노의 2번과 연결
- 블루투스 모듈의 RX 핀 (수신) 아두이노의 3번과 연결

- 조도센서:

- VCC(+)부분을 아두이노의 3.3V
- Analog(pin)을 아두이노의 A0번 핀
- GND(-)부분을 아두이노의 GND



근거리 블루투스 무선 통신 기반 아두이노 조도 데이터 수집(Slave 소스 프로그램)

```
#include <SoftwareSerial.h>
```

```
int t = 2;
```

```
int r = 3;
```

```
int pin = 13;
```

```
int light = A0;
```

```
int mode = '0'; // 실행 상태 저장 할 변수
```

```
int threshold = 280; //밝다와 어둡다는 구분할 한계 값
```

```
SoftwareSerial ble(t,r);
```

```
void setup() {
```

```
    Serial.begin(9600);
```

```
    ble.begin(38400);
```

```
    pinMode(pin, OUTPUT);
```

```
}
```

```
ble.begin(38400);
```

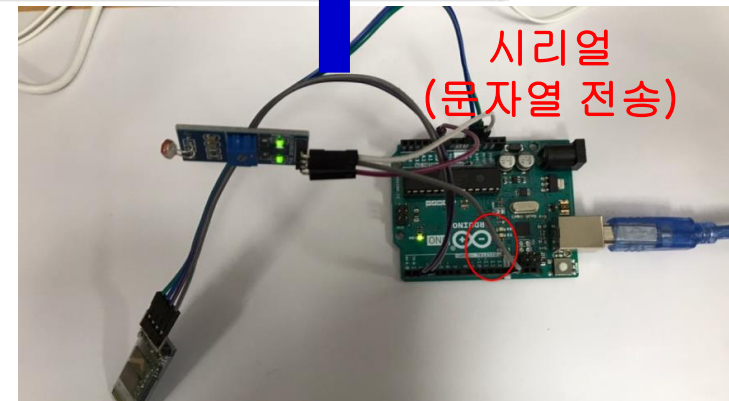
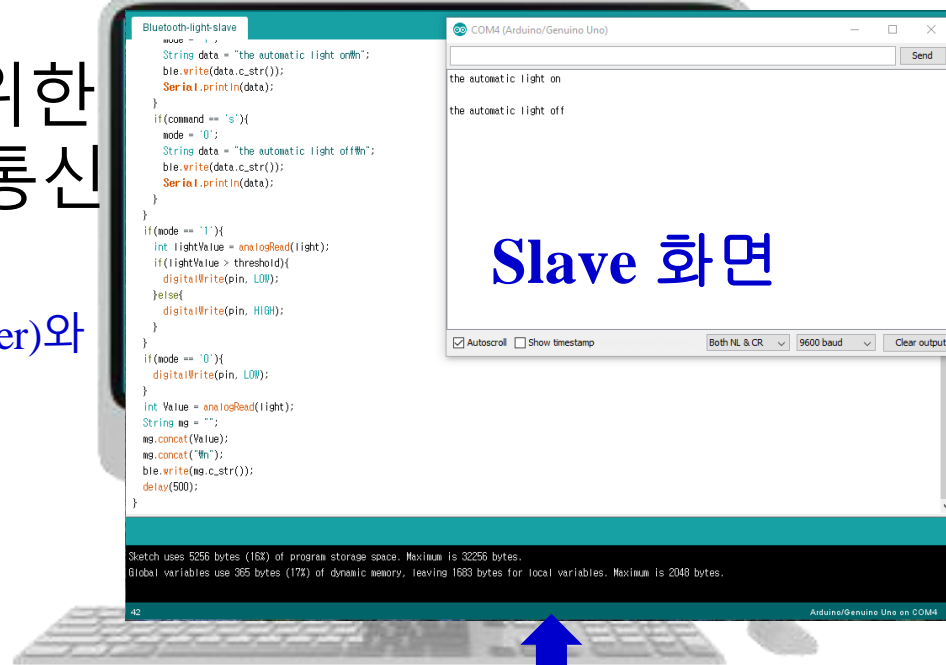
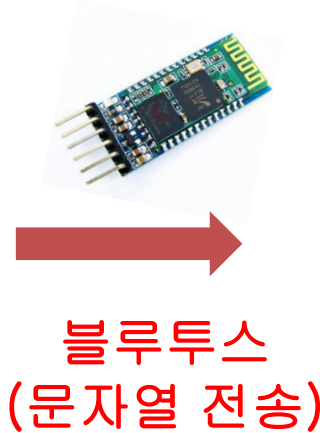
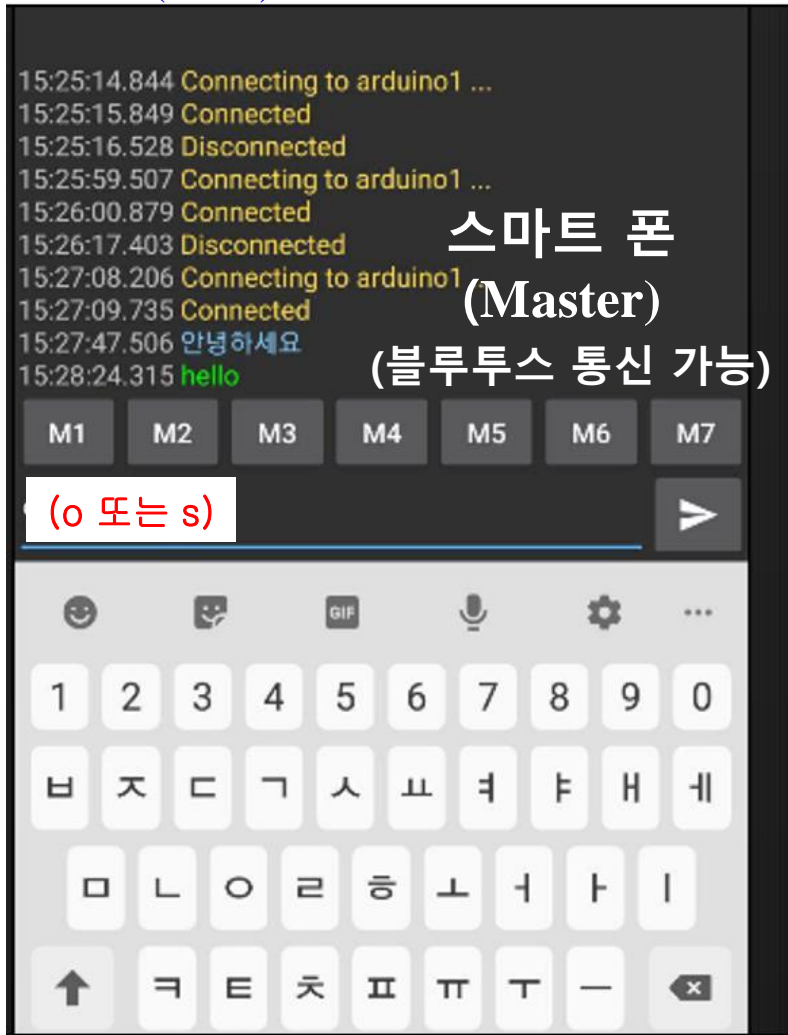
제대로 작도이 되지 않으면 수동으로

9600으로 변환 권장

```
void loop() {  
    if(Serial.available()){  
        ble.write(Serial.read());  
    }  
    if(ble.available()){  
        char command = ble.read(); //블루투스에서 전달하는 메시지 읽음  
        if(command == 'o'){          //전달된 값이 o이면  
            mode = '1'; // 한계 값에 의하여 LED를 자동으로 크고 끄도록 모드 변경  
            String data = "the automatic light onWn";  
            ble.write(data.c_str());  
            Serial.println(data);  
        }  
        if(command == 's'){          //전달된 값이 s이면  
            mode = '0';              // 자동 모드 취소  
            String data = "the automatic light offWn";  
            ble.write(data.c_str());  
            Serial.println(data);  
        }  
    }  
    if(mode == '1'){  
        int lightValue = analogRead(light); // 조도 센서로 부터 값을 읽어 드림  
        if(lightValue > threshold){          // 한계 값(280)보다 크면 LED off  
            digitalWrite(pin, LOW);  
        }else{                               // 한계 값보다 작으면 LED on  
            digitalWrite(pin, HIGH);  
        }  
    }  
    if(mode == '0'){                       // 자도 모드 취소 및 LED off  
        digitalWrite(pin, LOW);  
    }  
}
```

근거리 블루투스 무선 통신을 위한 스마트 폰(Android)과 아두이노 통신 실습 결과

- 블루투스 무선 통신 기반 스마트 폰(Android)(Master)와 아두이노(Slave) 연결 및 메시지 전달 결과



근거리 블루투스 무선 통신 기반 스마트 폰(Android)과 아두이노 조도 데이터 수집 실습 결과

스마트 폰(Android) 자동 LED 실행

스마트 폰(Android)으로 부터 o/s 명령
어 전달한 결과

```
light-slave
data = "the automatic light on\n";
ite(data.c_str());
.println(data);

nd == 's'){
'0';
data = "the automatic light off\n";
ite(data.c_str());
.println(data);

'1'){
tValue = analogRead(light);
value > threshold){
Write(pin, LOW);
Write(pin, HIGH);

'0'){
rite(pin, LOW);

= analogRead(light);
= "";
value);
"\n");
ng.c_str());
;
```

COM4 (Arduino/G

Send

the automatic light on

the automatic light off

☒ Autoscroll ☐ Show timestamp Both NL & CR 9600 baud Clear output

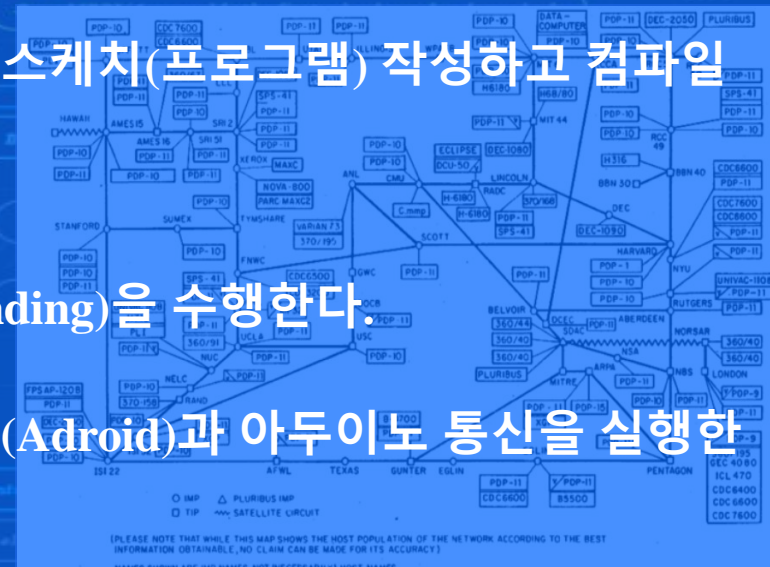
256 bytes (16%) of program storage space. Maximum is 32256 bytes.
les use 365 bytes (17%) of dynamic memory, leaving 1663 bytes for local variables. Maximum is 2048 bytes.

Arduino/Genuino Uno on COM4

스마트 폰(Android) 자동 LED 실행 취
소

학습 내용 정리

- 근거리 블루투스 무선 통신을 위한 아두이노 하드웨어 구성에 대해 알아본다.
- 근거리 블루투스 기반 아두이노와 스마트 폰(Android) 통신 실습 환경에 대해 살펴본다.
- 아두이노 블루투스 Slave 설정(Command)을 수행한다.
- 아두이노 블루투스 Slave 연결(Binding)을 수행한다.
- 근거리 블루투스 무선 통신을 위한 아두이노 스케치(프로그램) 작성하고 컴파일한다.
- 스마트 폰(Android) 블루투스 Master 연결(Binding)을 수행한다.
- 근거리 블루투스 무선 통신을 위한 스마트 폰(Android)과 아두이노 통신을 실행한다.





수업을 마치겠습니다.