

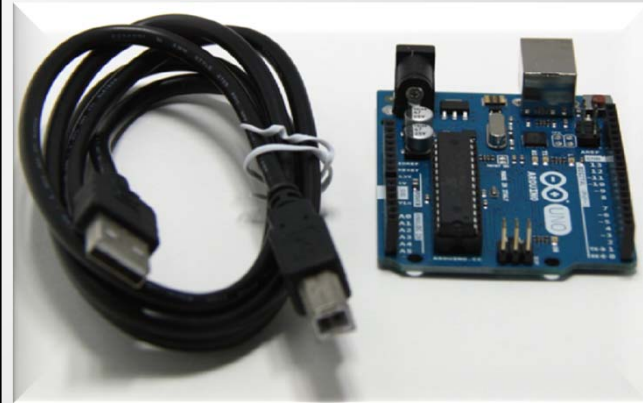
3장 예제2

예제

- 아두이노 보드와 PC와의 직렬통신

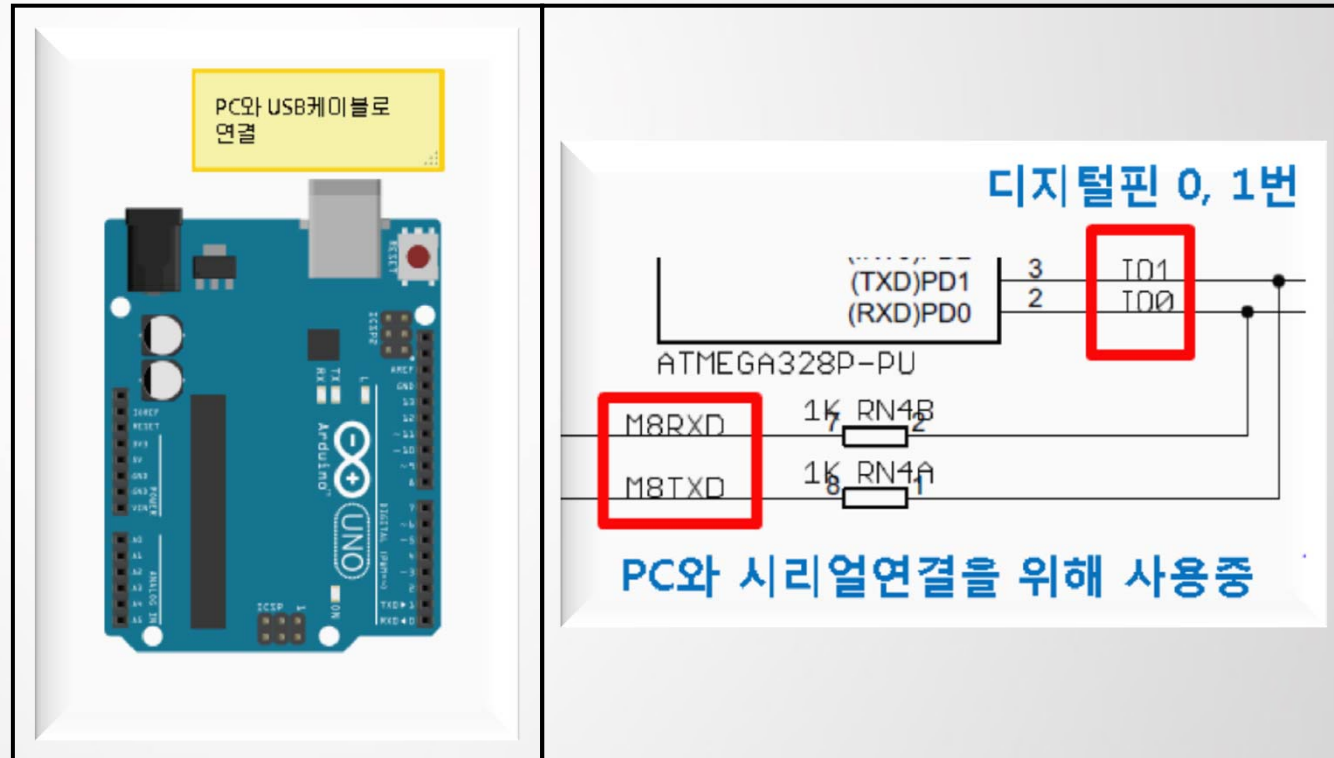
준비물

- 아두이노 우노보드
- USB케이블



3장 예제2

회로구성



3장 예제2

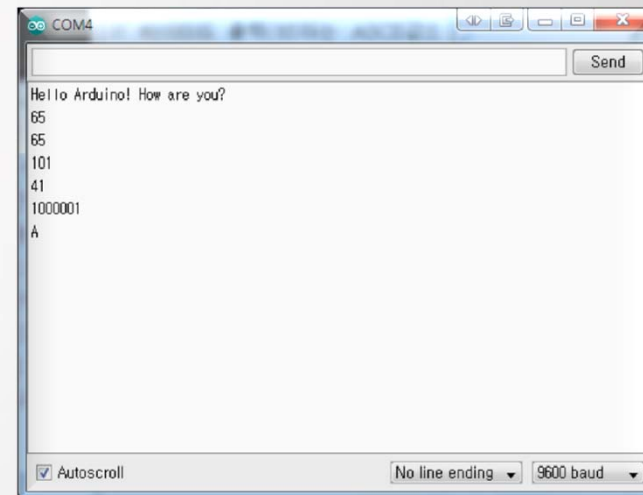
스케치

❖ 아두이노 보드와 PC사이의 시리얼통신 코드

```
int num = 65;           // 출력할 정수형 변수
void setup(){           // 초기화 구문
  Serial.begin(9600);    // 시리얼 통신 속도 설정, 시리얼모니터에서도 같게 설정
}
void loop(){            // 반복실행 구문
  Serial.print("Hello Arduino! "); // PC에 "Hello Arduino!"라고 보냄.
  Serial.println("How are you?");  // PC에 "How are you? "라고 보낸 후 캐리지 리
  // 턴, 줄 바꿈
  Serial.println(num);      // num의 값을 PC로 보냄, 받는 형태는 문자형
  Serial.println(num, DEC); // num값을 10진수형태로 보냄, 받는 형태는 문자형
  Serial.println(num, OCT); // num값을 8진수형태로 보냄, 받는 형태는 문자형
  Serial.println(num, HEX); // num값을 16진수형태로 보냄, 받는 형태는 문자형
  Serial.println(num, BIN); // num값을 2진수형태로 보냄, 받는 형태는 문자형
  Serial.write(num);        // num값을 바이너리 값으로 보냄
  // 받으면 ASCII값에 해당하는 문자가 출력됨
  Serial.end();            // 시리얼통신을 종료
}
```

3장 예제2

구현사진



3장 예제2

함수

❖ Serialbegin(통신 속도, 통신 설정)

설명 : 시리얼통신의 속도를 설정하고 시리얼통신을 시작합니다. 통신 속도는 baud rate입니다. 통신 설정은 데이터셋, 패리티와 stop bits로 이루어진 옵션을 사용합니다. 예를 들어 데이터셋은 8bit에 패리티는 없고(NO) stop bits는 1bit라고 한다면 통신 설정에 SERIAL_8N1을 넣으면 됩니다. 하지만 통신 설정을 빼고 명령을 줘도 기본으로 SERIAL_8N1이 설정됩니다.

❖ Serial.print(데이터) 또는 Serial.print(데이터, 형식)

형식 : HEX(16진수), DEC(10진수), OCT(8진수), BIN(2진수), 숫자(표시할 소숫점 자리)

설명 : 시리얼 통신을 통해 ASCII문자 형태로 데이터를 보냅니다. 사용할 수 있는 방법은 많이 있습니다만. 데이터에 숫자를 넣는다면 받는 쪽은 숫자를 그대로 표시한 문자 형태로 받습니다. 따옴표 안에 메시지를 적어 보내면 메시지가 그대로 보내지기도 합니다. 16진수부터 2진수 형으로 보낼 수도 있습니다.

3장 예제2

함수

❖ Serial.println(데이터) 또는 Serial.println(데이터, 형식)

형식 : HEX(16진수), DEC(10진수), OCT(8진수), BIN(2진수), 숫자(표시할 소숫점 자리)

설명 : 앞서 설명한 Serial.println()과 사용법이 동일합니다만 데이터를 보낸 후 마지막에 캐리지리턴과 줄 바꿈을 같이 보내 받는 쪽에서 줄이 넘어가는 효과를 볼 수 있습니다.

❖ Serial.write(값), Serial.write(값들의 집합), Serial.write(버퍼, 버퍼길이)

설명 : 값을 바이너리 형태로 보냅니다. 버퍼를 보낼 때에는 버퍼의 길이를 넣어주면 됩니다. 받는 쪽에서는 이 바이너리 형태가 옵션에 따라서 값에 대응하는 ASCII 문자형으로 표시되거나 HEX형으로 표시됩니다. ASCII 문자표는 부록을 참고하시길 바랍니다.

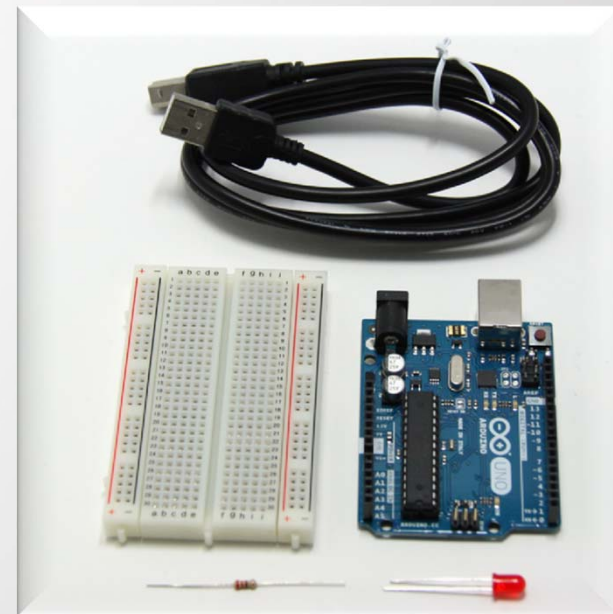
3장 예제3

예제

- 시리얼 통신으로 LED 켜고 끄기

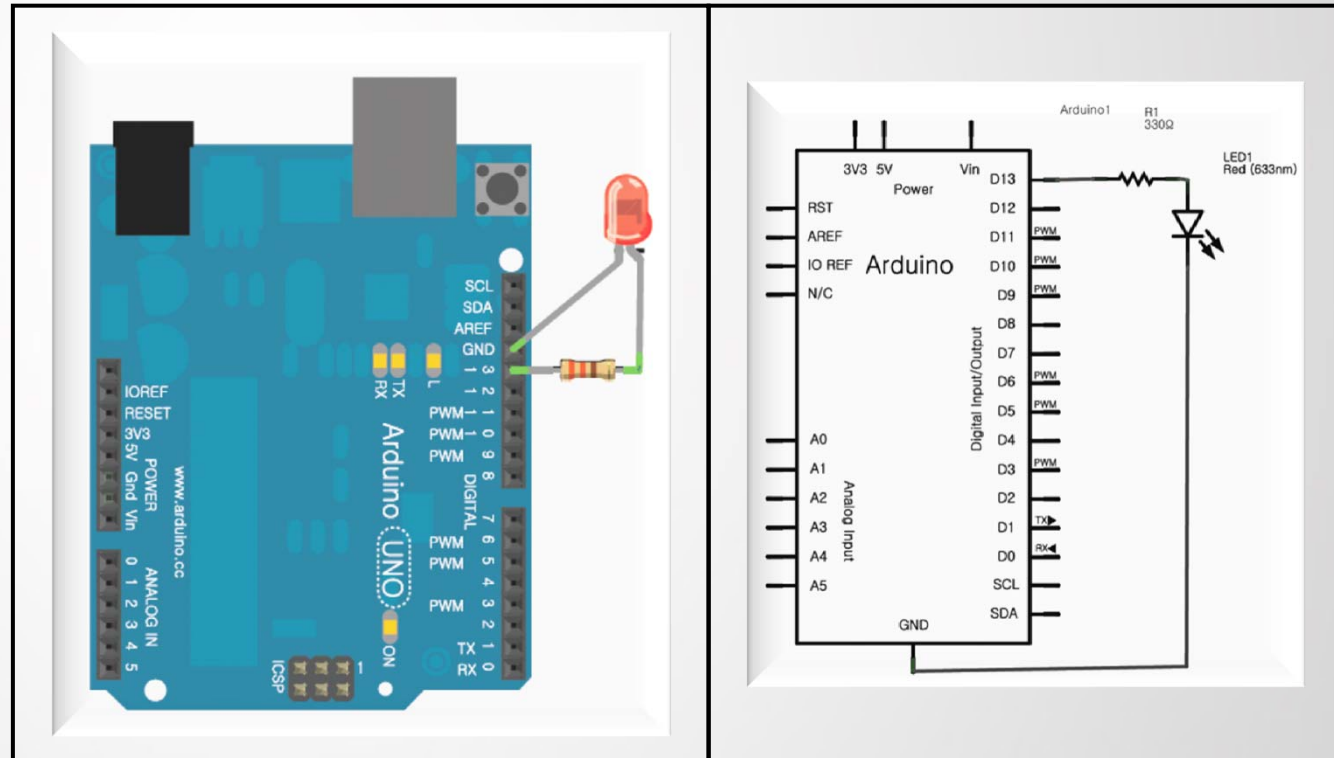
준비물

- 아두이노 우노 보드
- USB케이블
- 브레드 보드
- 저항 330Ω
- LED 1.8V 10mA



3장 예제3

회로구성



3장 예제3

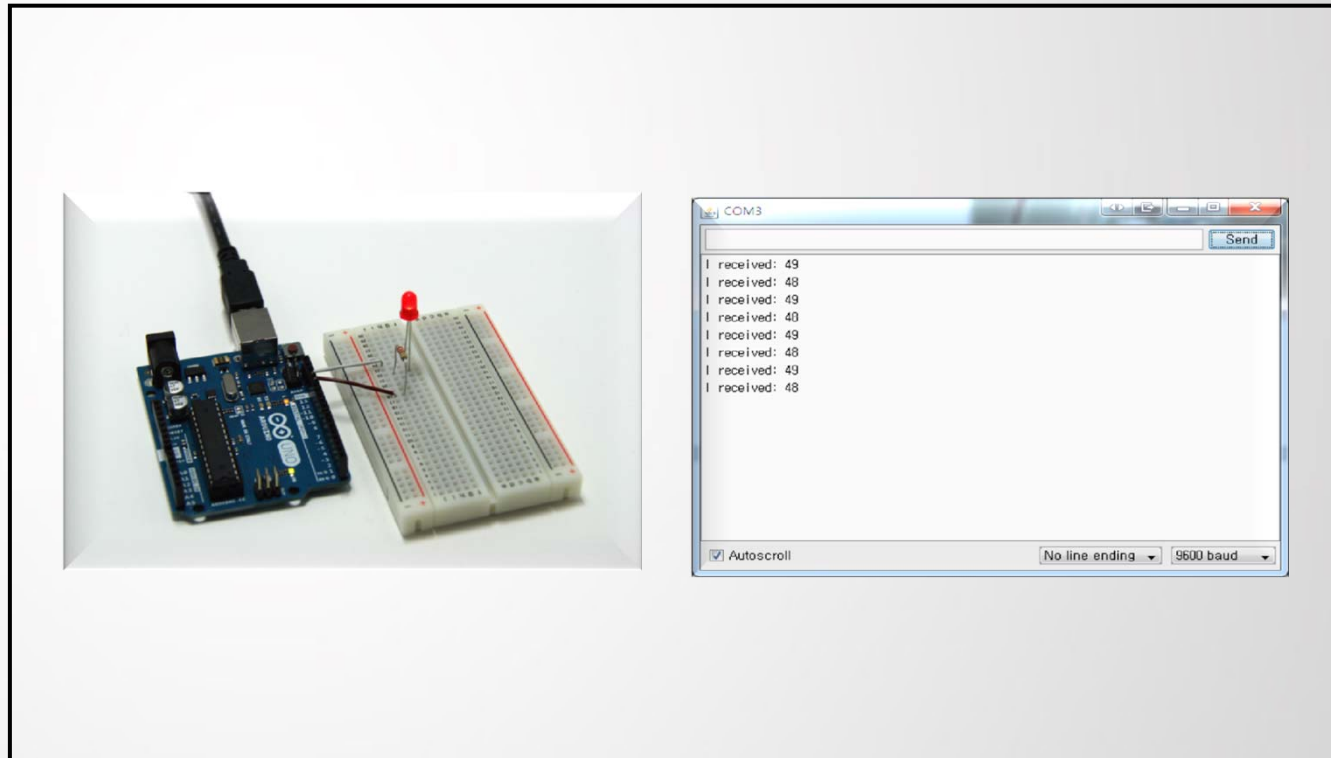
스케치

❖ 아두이노 보드와 PC사이의 시리얼통신 코드

```
const int LedPin = 13;           // LED 출력 핀을 지정하는 변수
byte CommandLed = '0';          // PC로부터 시리얼 통신을 통해 받을 명령
void setup(){                   // 초기화 구문
    Serial.begin(9600);          // 통신 속도 9,600bps로 시리얼 통신 시작
    pinMode(LedPin, OUTPUT);     // LED가 연결된 핀을 출력으로 설정
}
void loop() {                   // 반복 실행 구문
    if (Serial.available() > 0) { // 만약 시리얼 통신으로 받은 데이터가 있으면
        CommandLed = Serial.read(); // 시리얼 데이터를 읽어 CommandLed에 저장
        if (CommandLed == '1') { // 받은 데이터가 '1'이면
            digitalWrite(LedPin, HIGH); // LED를 켜기 위해 LED 핀에 HIGH를 출력
        }
        else if (CommandLed == '0') { // 그렇지 않고 받은 데이터가 '0'이면
            digitalWrite(LedPin, LOW); // LED를 끄기 위해 LED 핀에 LOW를 출력
        }
        Serial.print("I received: "); // 받은 명령어를 확인하기 위해 메시지 출력
        Serial.write(CommandLed); // 받은 명령어를 다시 PC로 전송
        Serial.println("");
    }
}
```

3장 예제3

구현사진



3장 예제3

함수

❖ Serial.available()

설명 : 시리얼통신을 통해서 수신된 데이터들은 버퍼에 저장되고 이 버퍼에 저장된 데이터의 개수(즉, 앞으로 가져올 수 있는 데이터들)를 리턴해 줍니다. 버퍼는 약 64 bytes까지 저장할 수 있습니다.

❖ Serial.read()

설명 : 시리얼 통신을 통해 받은 데이터를 읽어 옵니다.