



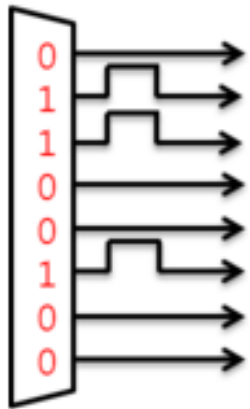
Arduino 시리얼 출력과 입력

▶ Arduino 시리얼 통신

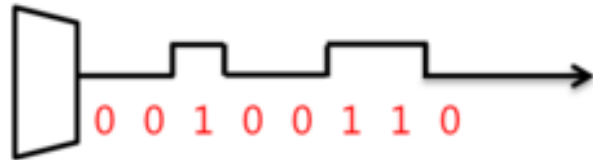
시리얼 통신(Serial Bus)이란?

- 한 개 전송 라인을 사용하여 직렬 방식으로 데이터를 송수신하는 통신 방법
- 직렬 통신 이라고도 함
- 아두이노에서는 시리얼 통신을 통해 데이터 입/출력을 할 수 있음

100(10진수) → 0x64(16진수) → 01100100(2진수)



병렬 통신



직렬 통신

▶ Arduino 시리얼 통신

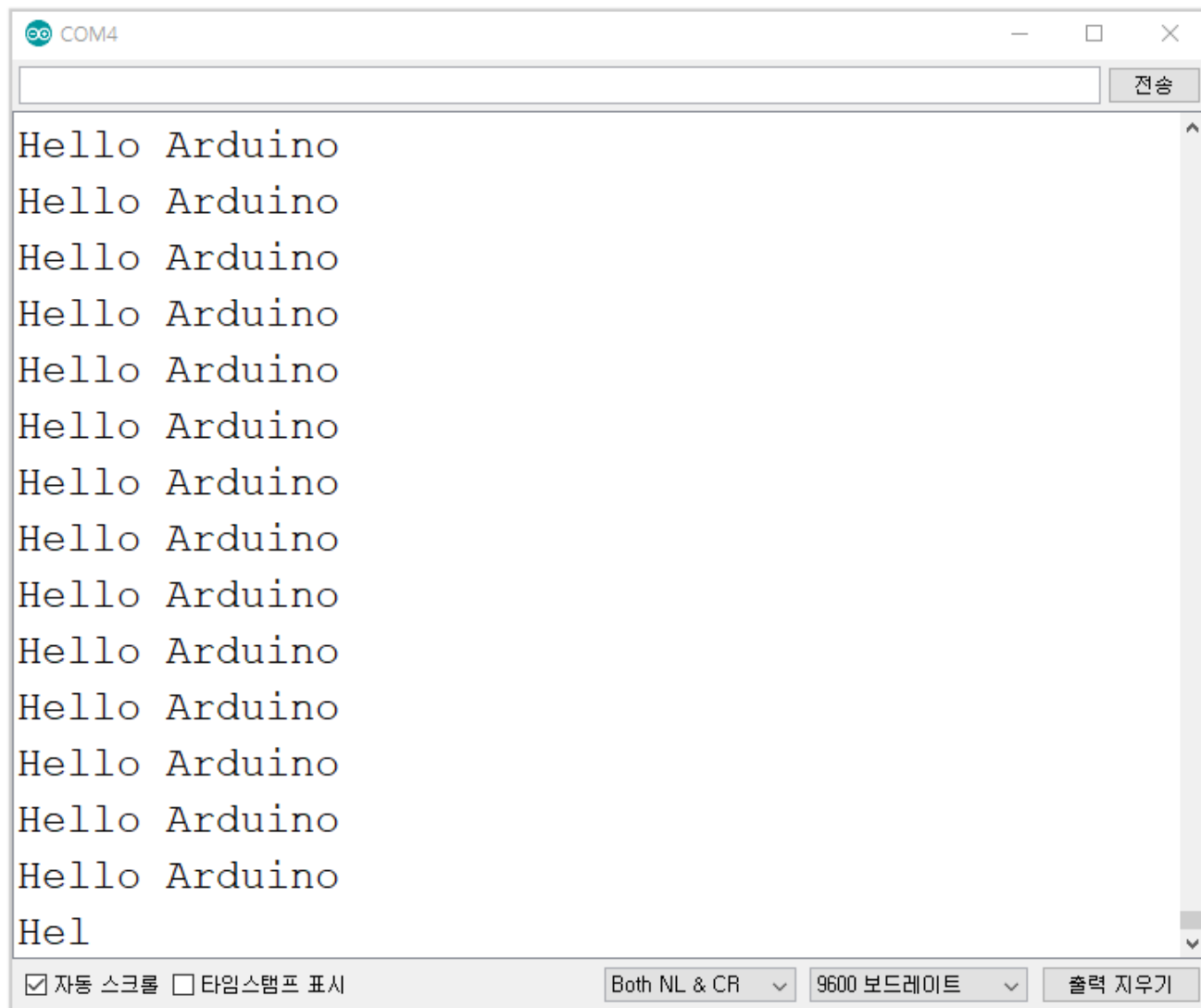
아두이노 에서도 PC와 연결된 시리얼을 통해 메시지 출력이 가능함

```
Serial.print("내용");//이어서 출력  
Serial.println("내용");//한 줄마다 출력
```

▶ Arduino 시리얼 통신 예제

```
void setup() {  
    Serial.begin(9600);  
}  
  
void loop() {  
    Serial.println("Hello Arduino");  
    delay(1000);  
}
```

툴 메뉴의 **시리얼 모니터**를 선택하거나 **Ctrl+Shift+M**을 눌러 시리얼 모니터를 실행한뒤 출력 결과를 확인한다.



▶ Arduino 시리얼 통신 코드 설명

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

데이터 전송 속도를 설정한다.

전송속도는 초당 전송되는 비트(bit)의 크기이며, 일반적으로 9600이라는 값을 많이 사용한다.

연결된 장비의 전송과 수신 속도가 동일해야 한다는 특징을 가진다.

9600 비트로 설정하면 1초당 1.2 킬로바이트(KB)의 데이터 전송을 수행한다.

```
Serial.println("Hello Arduino");
```

연결된 시리얼 포트에 ASCII Text를 `\r`, `\n`과 함께 전송한다.

`\r`은 carriage return을 수행하며 커서를 줄의 처음으로 이동시키고, `\n`은 new line의 작업을 수행하며 커서를 다음 줄로 이동시키므로 결국 엔터키를 누른 것과 같은 효과가 발생한다.

위의 명령은 단순히 Hello Arduino 라는 글자를 출력하며, 숫자의 경우 진법을 바꾸어 출력이 가능하다.

```
Serial.println(100);//100
```

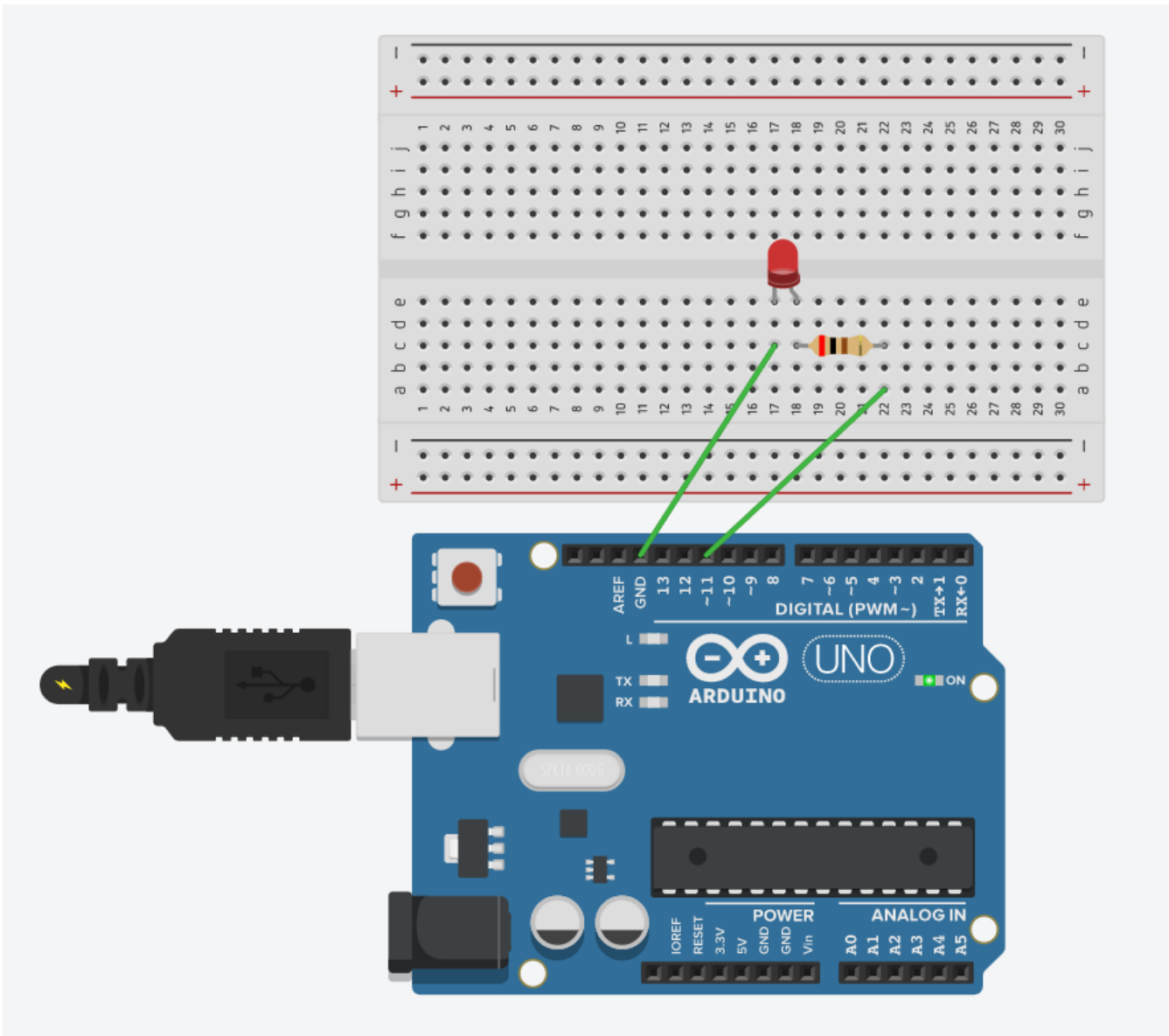
```
Serial.println(100, DEC);//100
```

```
Serial.println(100, HEX);//64
```

```
Serial.println(100, OCT);//144
```

```
Serial.println(100, BIN);//1100100
```

▶ Arduino 시리얼 통신 입력 예제



```
int LED = 11;
void setup()
{
    Serial.begin(9600);
    pinMode(redLED, OUTPUT);
}

void loop()
{
    if(Serial.available()){
        int input = Serial.parseInt();
        analogWrite(LED, input);
    }
}
```

직렬 모니터를 통해 전송 값을 입력 받아
전구의 밝기는 제어할 수 있음

▶ Arduino 시리얼 통신 입력 예제 설명

```
void setup()  
{  
    Serial.begin(9600);  
    pinMode(redLED, OUTPUT);  
}
```

setup 함수에서 Serial의 전송속도를 9600bit로 설정한다.
시리얼 모니터의 보드레이트와 일치시켜야 원하는 결과를 받을 수 있다.
pinMode는 현재 pinMode를 OUTPUT으로 설정

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

loop 함수는 지속적으로 반복되기 때문에 입력값이 들어온 경우만 원하는 코드를 수행하기 어려우므로 `Serial.available` 함수를 사용하여 입력값이 발생하였는지 여부를 파악한다.

`Serial.available` 함수는 시리얼을 통해 입력된 데이터의 바이트 수를 돌려준다. 따라서 0이 아니면 조건을 충족하게 되므로 if 내부의 코드가 실행된다.
(참고 : C언어 계열에서는 0이 거짓, 나머지 숫자는 참으로 인식한다)

▶ Arduino 시리얼 통신 입력 예제 설명

```
int input = Serial.parseInt();  
analogWrite(redLED, input);
```

`Serial.parseInt` 함수는 시리얼 입력값을 int 형태로 변환하여 입력하는 명령이다. 이외에도 여러 가지 입력 명령들이 존재한다.

- [Serial.read\(\)](#)
- [Serial.parseInt\(\)](#)
- [Serial.parseFloat\(\)](#)
- [Serial.readBytes\(\)](#)
- [Serial.readBytesUntil\(\)](#)
- [Serial.readString\(\)](#)
- [Serial.readStringUntil\(\)](#)

`analogWrite` 함수를 사용하여 입력값을 PWM 출력핀에 설정한다.

▶ Arduino 시리얼 통신 입력 실습

입력한 pin번호에 해당하는 LED 전구 불 켜기

초록LED – 3번 PIN / 빨강LED – 4번 PIN / 파랑LED – 5번 PIN / 노랑LED – 6번 PIN

1. 해당되는 PIN 번호를 시리얼 통신을 통해 입력하면 해당 LED의 불이 켜지고, 나머지는 불이 모두 꺼져야 함
2. 1번을 입력하면 모든 전구의 불이 켜져야 함
3. 0번을 입력하면 모든 전구의 불이 꺼져야 함

불이 켜질 때에는 시리얼 모니터를 통해 어떤 불이 켜지는지 출력되어야 함

Ex) 3번을 입력하면 "GREEN LED ON"

ex) 1번을 입력하면 "ALL LED ON"