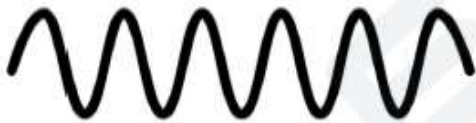


아날로그 - 출력

앞의 입출력 예제들은 모두 HIGH / LOW , 1 / 0 , ON / OFF 와 같이 신호가 두 가지로 이루어진 것들이었다. 하지만 빛의 세기, 온도, 거리 등을 감지하거나 소리 크기 조절, 빛의 세기 조절 등의 많은 값들은 연속되는 아날로그 신호들로 이루어져있다.



디지털 방식으로 작동하는 아두이노는 PWM (Pulse Width Modulation : 펄스 폭 변조)를 이용해 디지털 방식으로 아날로그 신호를 출력한다. 디지털 신호 ON (5V) OFF (0V) 신호가 지속되는 시간 부분을 변경 (펄스 폭 변경)하고, 이 패턴을 눈이 인식하지 못하는 속도로 반복하면, 0~5V 사이의 전압인 것처럼 보인다. 이는 마치 1초에 24프레임의 애니메이션이 연속된 동작을 보여주는 것처럼 느껴지는 이치라고 보면 된다.

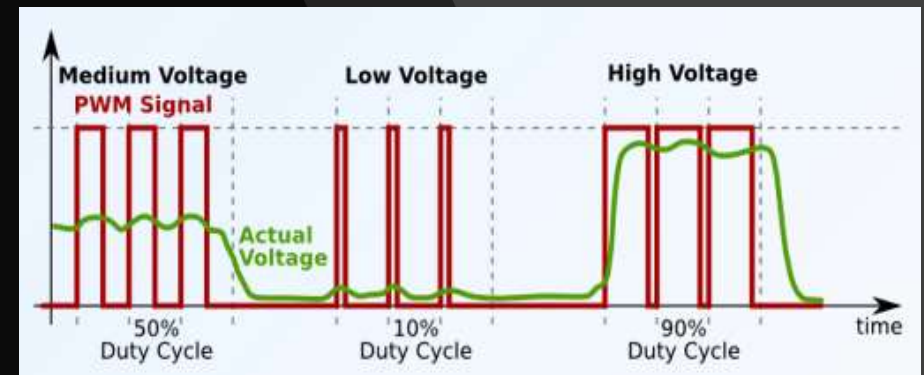
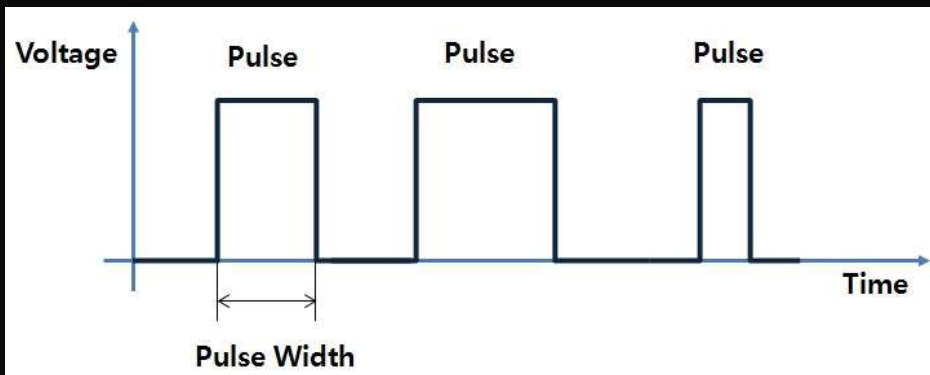
PWM 밝기 측정은 듀티 사이클 (duty cycle, 5V 전압의 지속 시간으로 생각하자.)이라는 용어로 설명된다. 듀티 사이클은 전체 실행 시간 대비 회로가 켜져 있는 시간의 백분율로 100%는 최대 밝기이고, 낮은 백분율은 낮은 광 출력을 나타낸다.



아날로그 – PWM

- **PWM(Pulse Width Modulation)**

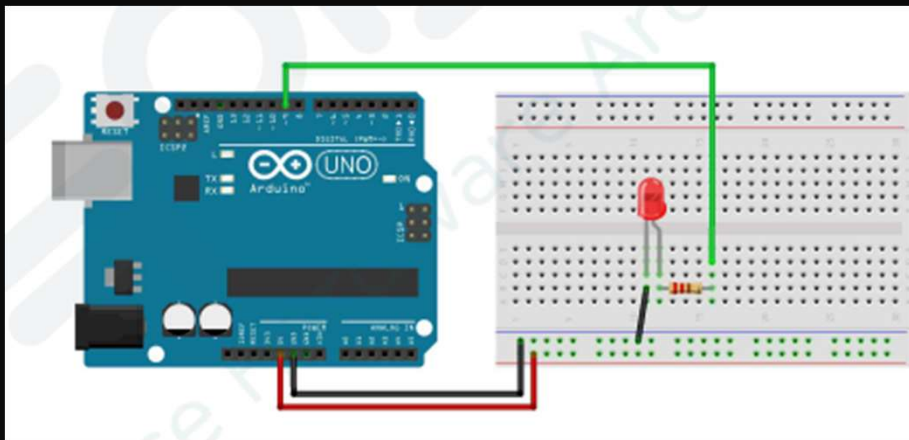
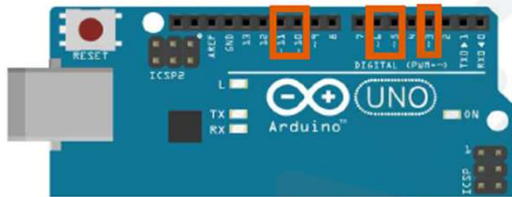
- PWM(Pulse Width Modulation)이란 펄스 폭 변조라고 하며, 명칭에서 알려주듯이 펄스 전압의 High/Low를 유지하는 폭을 조절함으로써 해당 시간에 전압이 어느 정도 걸리는지를 제어하는 방식
 - 기준에서 진폭이 신속하고 과도하게 변화하는 것을 펄스라 함



아날로그 – PWM

아두이노의 0~13번 디지털 핀에는 틸트 (~) 표시가 붙은 핀들이 있는데, 이 핀들이 아날로그 출력 (PWM) 핀이다.

~3, ~5, ~6, ~9, ~10, ~11 아날로그 핀들은 `pinMode()` 설정이 없다. 틸트 (~) 표시의 핀이지만 잘 확인해서 사용하면 된다.



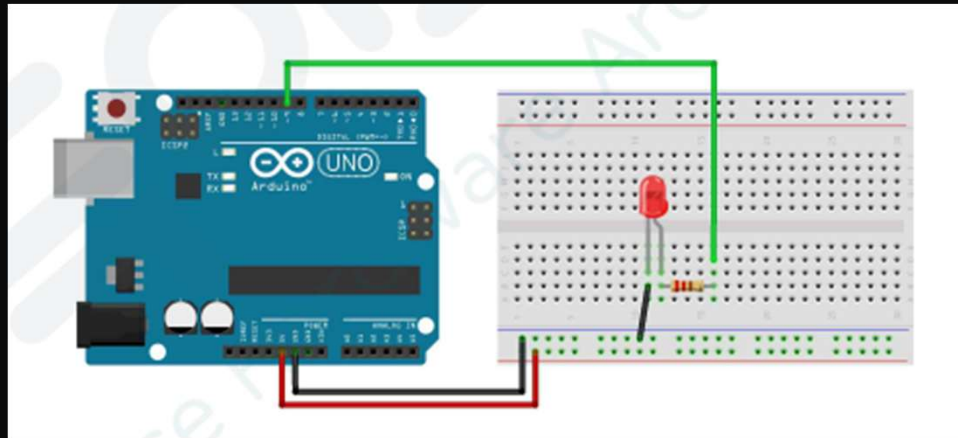
Step1 변수와 핀 모드 설정

```
1 int led = 9;           // 아날로그 출력 (PWM) LED 핀을 9 번에 설정
2 int brightness = 0;    // LED 밝기를 저장하는 변수
3 int fadeAmount = 5;    // fade 저장 변수
4
5 // the setup routine runs once when you press reset:
6 void setup() {
7
8   pinMode(led,OUTPUT);  // 9 번 핀을 출력으로 선언
9                          // analogWrite 구동되는 핀은 출력 선언하지 않아도 된다.
```

Step2 반복 실행문

```
9
10 void loop() {
11
12   analogWrite(led,brightness);    // 9 번 핀의 밝기를 설정
13
14   brightness = brightness + fadeAmount; // 루프를 돌때마다 fadeAmount 만큼 밝기 변화
15
16   if (brightness <= 0 || brightness >= 255) { // 밝기변화의 양 끝이면
17     fadeAmount = -fadeAmount;                // 방향을 바꾼다
18   }
19
20   delay(30);                                // 30 밀리초 기다린다.
21 }
```

아날로그 – PWM

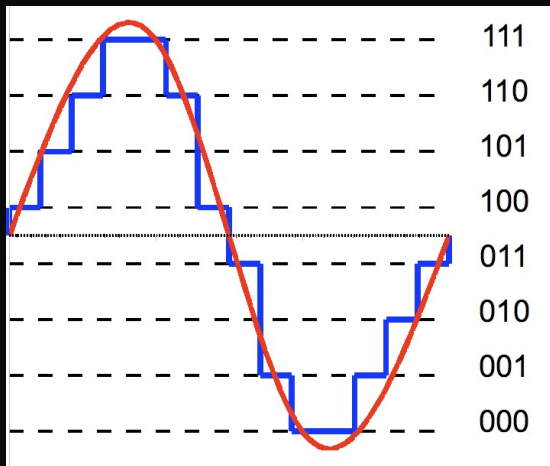
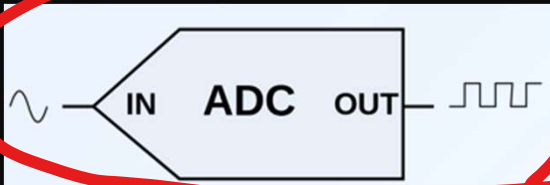


실습 : PWM LED 제어
1) 밝아졌다 어두워지는 동작을 반복하는 프로그램

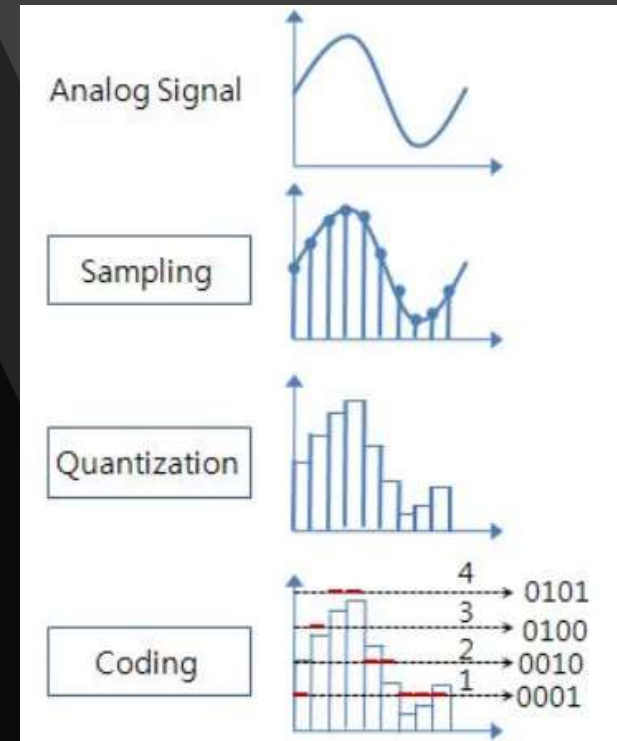
아날로그 - 입력

• ADC(Analog to Digital Converter)

- 아날로그-디지털 컨버터(변환회로) 혹은 D/A컨버터 등으로 불리며, 디지털 전기 신호를 아날로그 전기 신호로 변환하는 전자회로



신호 샘플링

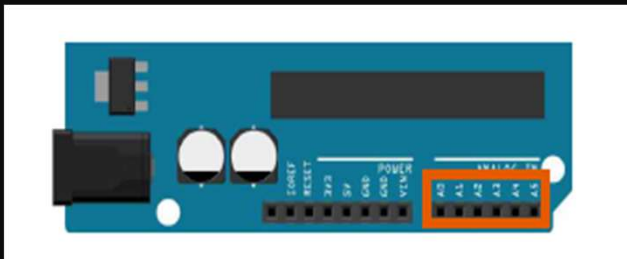


아날로그 – 입력

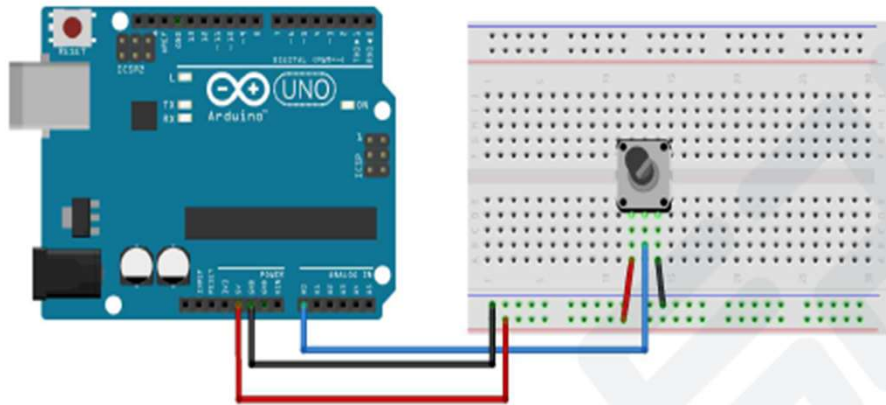
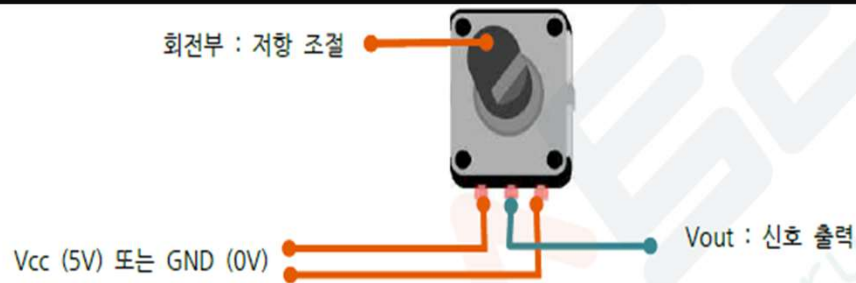
아두이노는 6채널 아날로그- 디지털 변환기 (Analog Digital Converter : ADC)가 포함되어있다.

변환기가 3비트 분해 능력을 가진다면, 입력 전압 범위를 2^3 개의 단계로 구분한다는 뜻이다. 해상도가 높을수록 훨씬 높은 정확도를 가지고 측정할 수 있다.

아두이노의 아날로그- 디지털 변환기는 아날로그 신호를 10비트 분해 능력 ($2^{10} = 1024$)을 가진다. 즉, 0~5V의 입력 전압을 디지털 신호로 변환하여 0~1023 경수로 반환하며, 2.4mV 단위로 전압을 구분할 수 있다.



아날로그 - 입력



Step1 변수와 핀 모드 설정

```
1 void setup() {
2
3   Serial.begin(9600); // 9600 속도로 시리얼 통신 시작
4                       // 아날로그 입출력 핀은 선언하지 않아도 된다.
5 }
```

Step2 반복 실행문

```
5   :
6 void loop() {
7
8   // 가변저항의 가운데 핀을 A0 핀에 연결
9   int sensorValue = analogRead(A0); // 가변저항 값을 읽어 sensorValue 에 저장
10                                     // sensorValue 에는 0~1023 범위의 정수 값이 저장
11
12   float voltage = sensorValue * (5.0 / 1023.0); // 아날로그 판독 값을 전압 0~5V 로 변환
13   // 연산 결과 값이 실수형이므로 float 변수에 저장
14   Serial.println(voltage); // 전압값을 읽어 시리얼 창에 한 줄씩 프린트
15 }
16
```

아날로그 - 입력

실습 : Analog LED 제어
1) 가변 저항으로 LED 밝기 제어

