

# 아두이노 기초 및 실험

[기계공학실험1]

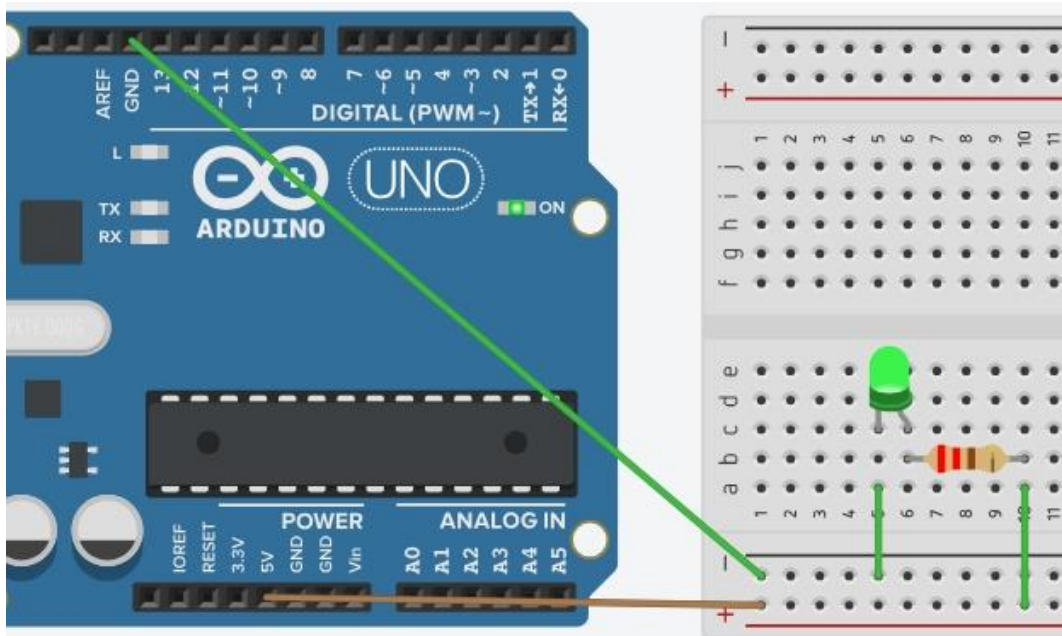
1분반(금요일 3조)

학번: 20191820

이름: 김형준

## 1. LED에 저항을 연결하는 이유 설명

LED 소자가 버틸 수 있는 최대 전압은 2V 정도 인데, 아두이노의 +극(5V)과 LED의 +극을 저항 없이 직접 연결하게 되면 과전압이 흐르게 되어 LED소자와 아두이노 보드, 최악의 경우에는 전원을 공급하는 컴퓨터까지도 손상을 입을 수 있기 때문에 적절한 전압을 공급하기 위해 아래 그림처럼 LED의 +극에 저항을 연결해야만 한다.



위와 같이 저항을 연결하면, 아두이노에서 공급하는 5V의 전압이 저항을 지나면서 소모되어, LED에 걸리는 전압의 크기가 2V이하의 적절한 전압이 된다.

이때 LED에 연결할 저항의 저항값은 LED에 흐를 수 있는 최대 전류의 크기가 20mA이므로 옴의 법칙에 의해  $\frac{3V}{20mA} = 150\Omega$  이상이 되어야 한다. 따라서 아두이노 키트에서 220 $\Omega$ 의 저항을 LED에 연결하면 적절한 전류를 LED에 흐르게 할 수 있다.

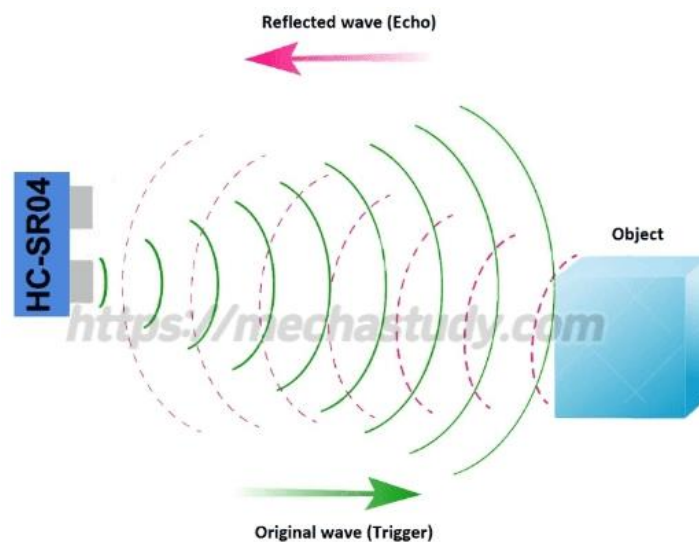
하지만, LED 구동시 필요한 최소 전압과 최소 전류도 있기 때문에, 너무 큰 저항(10k $\Omega$ )을 연결하는 것은 바람직하지 않다.

## 2. 초음파 센서 및 부저의 원리 설명

### 2-1. 초음파 센서의 원리

초음파는 인간의 가청주파수인 20000Hz를 초과하는 고주파 음파로서, 압전소자에 고주파 전류를 흘려서 빠르게 진동시키거나 자성물체에 대해 자기장을 변화시켜 진동시키는 방식으로 생성한다.

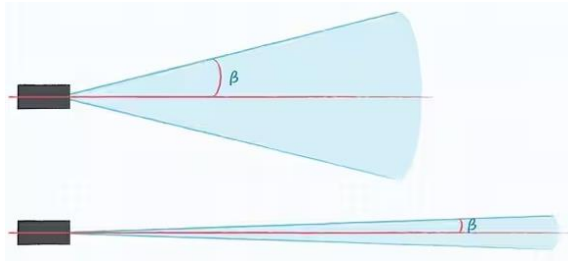
초음파 센서의 감지 방식은 박쥐가 비행중에 길을 찾기위해 반향 정위를 사용하는 방법과 유사하다. 초음파 센서는 초음파 발신기와 초음파 수신기, 제어 회로로 구성되어 있는데, 주변의 물체에 대한 거리를 계산하기 위해 발신기에서 초음파를 일정한 주기로 생성하여 공기 중으로 방출한다. 그 다음, 방출된 초음파가 주변의 물체에 반사되고, 이 반사된 초음파를 수신기에서 감지하여 방출된 초음파가 다시 반사되어 돌아오는데 까지 걸린 시간을 정확하게 측정한다. 제어 회로에서는 초음파 센서의 작동을 제어하고 데이터를 처리하는 역할을 하는데, 이 회로는 센서의 정확도와 신뢰성에 중요한 영향을 미친다.



아두이노에서 사용되는 초음파 센서인 HC-SR04는 Trig부와 Echo부에서 각각 초음파를 발신하고 감지하는데, 아두이노의 디지털 핀을 통해 Trig부에 전압을 가하면 Trig핀에 연결된 압전소자가 초음파 신호를 발생시키고, Echo부에서 물체에서 반사된 초음파를 감지해 전압을 발생시켜 Input핀을 통해 아두이노가 초음파가 돌아올 때까지 걸린 시간을 측정할 수 있게 한다.

물체까지의 거리(d)는 다음과 같은 식으로 구할 수 있다.

$$d = \frac{t \cdot c}{2} \quad (t : \text{초음파 송신부터 수신까지의 소요시간}, c : \text{초음파의 속도 (340m/s)})$$



왼쪽 그림과 같이 초음파 센서에서 발생시키는 초음파는 멀리 갈수록 모든 방향으로 넓게 퍼지므로 적절한 음파의 빔 각도를 가지는 초음파 센서를 선택해야 한다. 빔 각도가 넓은 센서의 경우 넓은 영

역에서 물체를 감지할 수 있지만, 정확도가 떨어진다는 단점이 있고, 빔 각도가 좁은 센서의 경우 정확도가 높지만 물체를 감지할 수 있는 영역이 좁다는 단점이 있다.

## 2-2. 부저의 원리

부저는 전압을 통해 내부의 막을 진동시켜 소리를 내는 장치이다.

부저에서 소리가 나는 원리는 압전효과를 역으로 적용하는 것인데, 압전물질에 전압을 가하면 기계적인 변형이 발생하는 현상을 이용한다.

부저 내부의 막에 전압이 가해지면 압전효과에 의해 진동하게 되어 소리가 발생하게 된다.

이때, 전압을 가해주는 주기를 디지털 신호로 조절하여 막의 진동수를 조절할 수 있는데, 이를 통해 부저로 원하는 주파수의 소리를 발생시킬 수 있다.

가해주는 전압의 진동수가 클수록 부저에서 고주파의 소리가 발생하며, 가해주는 전압의 진동수가 작을수록 부저에서 저주파의 소리가 발생하게 된다.

부저는 크게 능동 부저와 수동 부저 두가지 종류로 나뉘는데, 능동 부저는 내장된 회로가 있어 전원이 공급되면 소리가 출력되고, 공급되지 않으면 소리가 출력되지 않는 부저이다.

그러나 능동 부저는 미리 정해진 한 가지의 주파수의 소리만 낼 수 있다는 단점이 있다.

수동 부저는 내장된 회로가 없기 때문에 전원만 공급하면 소리가 나지 않고, 전원을 공급하면서 주파수를 디지털 신호로 만들어 전류를 흘려주면 그 주파수에 해당하는 소리를 내는 부저이다. 따라서 수동 부저는 능동 부저와 다르게 다양한 주파수의 소리를 낼 수 있다.

### 3. 아두이노 코드 및 설명

#### 3-1. 조건문을 활용하여, 시리얼 모니터로 특정 값 입력 시 그에 대응하는 LED 발광

```
1  int LED_R = 11;
2  int LED_Y = 10;
3  int LED_G = 9;
4  char input;
5
6  void setup() {
7      Serial.begin(9600);
8      Serial.flush();
9
10     pinMode(LED_R, OUTPUT);
11     pinMode(LED_Y, OUTPUT);
12     pinMode(LED_G, OUTPUT);
13     digitalWrite(LED_R, LOW);
14     digitalWrite(LED_Y, LOW);
15     digitalWrite(LED_G, LOW);
16
17     Serial.println("Input Color (R : Red, Y : Yellow, G : Green)");
18 }
19
20 void loop() {
21     if (Serial.available() > 0)
22     {
23         input = Serial.read();
24         Serial.print(input);
25
26         if (input == 'R')
27         {
28             digitalWrite(LED_R, HIGH);
29             digitalWrite(LED_Y, LOW);
30             digitalWrite(LED_G, LOW);
31         }
32         else if (input == 'Y')
33         {
34             digitalWrite(LED_R, LOW);
35             digitalWrite(LED_Y, HIGH);
36             digitalWrite(LED_G, LOW);
37         }
38         else if (input == 'G')
39         {
40             digitalWrite(LED_R, LOW);
41             digitalWrite(LED_Y, LOW);
42             digitalWrite(LED_G, HIGH);
43         }
44     }
45 }
```

빨강, 노랑, 초록색 LED에 연결한 digital pin을 int형 전역변수를 선언해 값을 저장한다.

실험에서는 빨강, 노랑, 초록색 각각을 11번, 10번, 9번으로 연결하였다.

그 다음 시리얼 모니터에서 입력한 문자를 저장할 char형 전역변수 input를 선언했다.

void setup()에서 아래와 같은 초기화 코드를 넣는다. (시작시 한번만 실행)

9600 보드레이트(초당 9600bit 전송)로 통신을 맞추기 위해 Serial.begin(9600)을 사용하고, 시리얼 수신부의 버퍼를 비우기 위해 Serial.flush()를 사용했다.

각각의 LED는 모두 출력으로 사용되므로 pinMode(LED pin, OUTPUT)으로 설정하고,

각각의 LED의 초기상태를 모두 꺼져있게 설정하기 위해 digitalWrite(LED pin, LOW)로 설정했다.

마지막으로 시리얼 모니터에 사용법을 메시지로 출력한다(Serial.println).

void loop()에서 초기화 후 계속 반복할 코드를 넣는다.

시리얼 모니터에서 값이 입력된 경우에만 LED 발광여부에 변화가 있어야 하므로 if문의 조건에 Serial.available() > 0을 넣고, if문 내부에 LED 제어코드를 넣는다.

Serial.read()를 사용해 input 변수에 입력한 문자를 저장하고, Serial.print(input)으로 입력한 문자를 시리얼 모니터에 출력한다. 그 후, if문을 사용해 input의 값이 각각 R일 때 빨강색 LED만 켜지

게(빨강색만 HIGH, 나머지는 LOW로 설정), Y일 때 노란색 LED만 켜지게, G일 때 초록색 LED만 켜지게 설정했다.

### 3-2. 조건문을 활용하여, 시리얼 모니터로 특정 값 입력 시 그에 대응하는 소리 발생

```
1  int buzzer = 6;
2  char input;
3
4  void setup() {
5      Serial.begin(9600);
6      Serial.flush();
7
8      pinMode(buzzer, OUTPUT);
9
10     Serial.println("Input tone (H : High, M : Medium, L : Low)");
11 }
12
13 void loop() {
14     if (Serial.available() > 0)
15     {
16         input = Serial.read();
17         Serial.print(input);
18
19         if (input == 'H')
20         {
21             tone(buzzer,4000,1000);
22         }
23         else if (input == 'M')
24         {
25             tone(buzzer,2000,1000);
26         }
27         else if (input == 'L')
28         {
29             tone(buzzer,500,1000);
30         }
31     }
32 }
```

부저에 연결한 digital pin을 int형 전역변수를 선언해 값을 저장한다.

실험에서는 부저를 6번으로 연결하였다.

그 다음 시리얼 모니터에서 입력한 문자를 저장할 char형 전역변수 input를 선언했다.

void setup()에서 아래와 같은 초기화 코드를 넣는다. (시작시 한번만 실행)

위 실험과 마찬가지로, 9600 보드레이트(초당 9600bit 전송)로 통신을 맞추기 위해 Serial.begin(9600)을 사용하고, 시리얼 수신부의 버퍼를 비우기 위해 Serial.flush()를 사용했다.

부저는 출력으로 사용되므로 pinMode(부저의 pin, OUTPUT)으로 설정하고,

마지막으로 시리얼 모니터에 사용법을 메시지로 출력한다(Serial.println).

void loop()에서 초기화 후 계속 반복할 코드를 넣는다.

시리얼 모니터에서 값이 입력된 경우에만 부저에 소리가 출력되야 하므로 if문의 조건에 Serial.available() > 0을 넣고, if문 내부에 부저 제어코드를 넣는다.

Serial.read()를 사용해 input 변수에 입력한 문자를 저장하고, Serial.print(input)으로 입력한 문자를 시리얼 모니터에 출력한다. 그 후, if문을 사용해 input의 값이 각각 H일 때 고주파(4000Hz)가, M일 때 중주파(2000Hz)가, L일 때 저주파(500Hz)가 1초간(1000ms) 출력되게 설정했다.

### 3-3. 초음파 센서를 활용하여, 거리에 따른 LED 발광 및 부저 소리 발생

```
1  int LED_R = 11;
2  int LED_Y = 10;
3  int LED_G = 9;
4  int buzzer = 6;
5  int Echo = 3;
6  int Trig = 2;
7
8  void setup() {
9      Serial.begin(9600);
10     Serial.flush();
11
12     pinMode(LED_R, OUTPUT);
13     pinMode(LED_Y, OUTPUT);
14     pinMode(LED_G, OUTPUT);
15     pinMode(buzzer, OUTPUT);
16     pinMode(Trig, OUTPUT);
17     pinMode(Echo, INPUT);
18 }
21 void loop() {
22     float duration, distance;
23
24     // 초음파 발사
25     digitalWrite(Trig,HIGH);
26     delay(1);
27     digitalWrite(Trig,LOW);
28
29     // 반사된 초음파의 시간을 저장
30     duration = pulseIn(Echo, HIGH);
31
32     // 거리 계산
33     distance = ((float)(duration*340)/10000)/2;
34
35     // 거리 출력
36     Serial.print("distance : ");
37     Serial.print(distance);
38     Serial.println("cm");
39
40     // 거리에 따라 LED 및 부저 출력
41     if (distance <= 10)
42     {
43         tone(buzzer,4000,1000);
44         digitalWrite(LED_R, HIGH);
45         digitalWrite(LED_Y, LOW);
46         digitalWrite(LED_G, LOW);
47     }
48     else if (distance <= 20)
49     {
50         tone(buzzer,2000,1000);
51         digitalWrite(LED_R, LOW);
52         digitalWrite(LED_Y, HIGH);
53         digitalWrite(LED_G, LOW);
54     }
55     else
56     {
57         tone(buzzer,500,1000);
58         digitalWrite(LED_R, LOW);
59         digitalWrite(LED_Y, LOW);
60         digitalWrite(LED_G, HIGH);
61     }
62 }
```

빨강, 노랑, 초록색 LED와 부저, 초음파 센서의 Echo와 Trig에 연결한 digital pin들을 각각 int형 전역변수를 선언해 값을 저장한다.

실험에서는 빨강, 노랑, 초록색 각각을 11번, 10번, 9번으로, 부저는 6번, 초음파 센서의 Echo와 Trig는 각각 3번과 2번에 연결하였다.

void setup()에서 아래와 같은 초기화 코드를 넣는다. (시작시 한번만 실행)

9600 보드레이트(초당 9600bit 전송)로 통신을 맞추기 위해 Serial.begin(9600)을 사용하고, 시리얼 수신부의 버퍼를 비우기 위해 Serial.flush()를 사용했다.

각각의 LED는 모두 출력으로 사용되므로 pinMode(LED pin, OUTPUT)으로 설정하고,

부저와 초음파 센서의 Trig부는 모두 출력으로 사용되므로 pinMode(pin, OUTPUT)로 설정한다.

마지막으로, 초음파 센서의 Echo부는 입력으로 사용되므로 pinMode(pin, INPUT)으로 설정한다.

void loop()에서 초기화 후 계속 반복할 코드를 넣는다.

초음파가 송신부터 수신까지 걸리는 시간( $\mu\text{s}$ 단위)을 저장할 변수 duration과 물체의 거리(cm단위)를 저장할 변수 distance를 각각 float 타입으로 선언한다.

digitalWrite(Trig,HIGH)로 Trig부에서 초음파를 발사하고, 중복 발사를 방지하기 위해 1ms이후 digitalWrite(Trig,LOW)로 초음파 발사를 중단한다.

그 다음, pulseIn 함수를 사용하여 Echo부에 도달한 반사된 초음파의 측정된 시간을 duration에 저장한다. (저장된 시간은 마이크로초 단위임)

이후 distance에 duration의 값을 사용해 물체의 거리를 계산해서 저장한다.

이때, distance는 cm 단위이고, duration은 센서로부터 물체까지 왕복하는데 걸린 시간이므로,

$$\begin{aligned}\text{distance} &= \left(\frac{\text{duration}}{2}[\mu\text{s}]\right) \times (340[\text{m/s}]) = \left(\frac{\text{duration}}{2} \times 10^{-6}[\text{s}]\right) \times (34000[\text{cm/s}]) \\ &= (((\text{duration} \times 340) \div 10^4) \div 2)[\text{cm}]\end{aligned}$$

가 된다. 이를 코드로 나타내면 distance = ((float)(duration\*340)/10000)/2; 이다.

그 다음, 계산된 거리를 Serial.print함수를 사용해 시리얼 모니터에 출력한다.

마지막으로 if문을 사용해 거리에 따라 점등할 LED와 부저에서 출력할 소리의 주파수를 선택한다.

거리가 10cm 이하인 경우, 고주파(4000Hz)를 출력하고, 빨간색 LED만 점등한다.

거리가 10cm 초과, 20cm 이하인 경우 중주파(2000Hz)를 출력하고, 노란색 LED만 점등한다.

거리가 20cm 초과인 경우, 저주파(500Hz)를 출력하고, 초록색 LED만 점등한다.