# 멜로디 제어 응용

박 정욱

## 배울 내용(Contents)

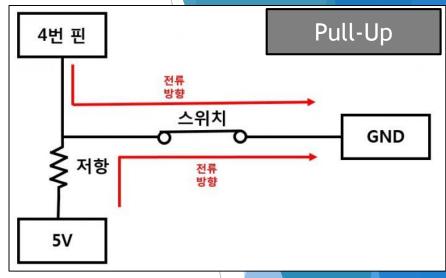
- 버튼 기본과 응용 내용 복습
- 멜로디 내용 복습과 프로젝트 진행
  - 배열 내용 이해
- 함수의 사용과 이해
- 함수 사용 응용과 멜로디 제어에 접합

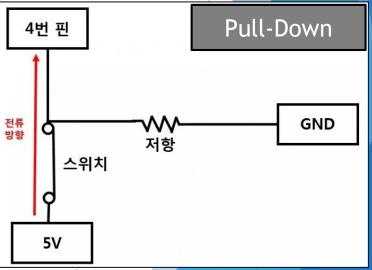
#### Switch(버튼 복습)

- 버튼은 기본적으로 Floating(플로팅) 상태에서 각각 회로 구성에 따라 Pull-Up과 Pull-Down 버튼으로 사용한다.
  - Pull-Up: 왼쪽 상단의 그림처럼 저항을 VCC에 연결한 상태
  - Pull-Down: 왼쪽 하단의 그림처럼 저항을 GND에 연결한 상태
- 기본적으로

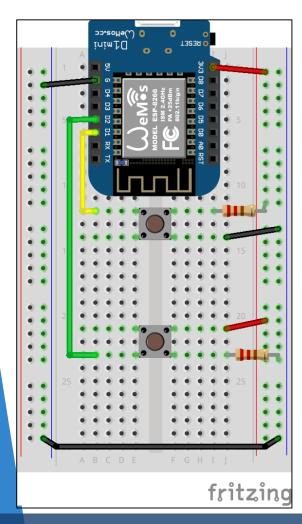
Pull-Up은 눌렸을 때 O(Low), 가만히 있을 때 1(High)를 출 력하며,

Pull-Down은 눌렸을 때 1(High), 가만히 있을 때 0(Low)를 출력 한다.





#### Switch 기초 복습



■ 각각 Pull-Up(위), Pull-Down(아래) 회로를 둘다 구성하고 스위치 동작의 차이점을 구분한다.

- 회로 구성
  - VCC = 3.3v
  - 디지털핀 = D1, D2
  - GND = GND

• 소스를 코딩하고 동작 상태를 확인.

# Switch 기초 복습

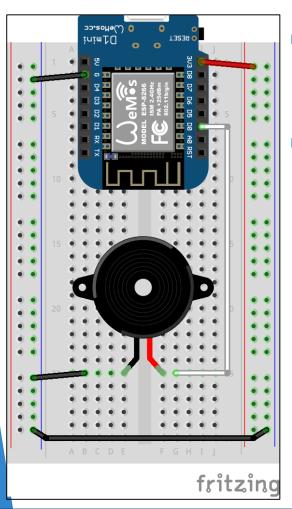
```
22_Lab-02§
1 const int btn1 = D1;
2 const int btn2 = D2;
4 void setup() {
    Serial.begin(115200):
    pinMode(btn1, INPUT);
    pinMode(btn2, INPUT);
9 }
11 void loop() {
    Serial.print("Button1 : ");
    Serial.println(digitalRead(btn1));
14 Serial.print("Button2 : ");
15 Serial.println(digitalRead(btn2));
    Serial.println("-----");
    delay(500);
18 }
```

#### 멜로디 기본 내용



- 피에조 부저(스피커)로 간단한 스 피커 제작을 진행 한다.
- -삑 소리(비프음)만 사용 가능하며, 음높이를 사용하여 도~시 까지의 음을 표현 가능 하다.

#### 멜로디 실습 기초



■ 스피커 1개를 연결하여 멜로디 실습을 진행

- 회로 구성
  - 스피커의 -극에 GND
  - 스피커의 +극에 D0를 연결

### 멜로디 실습 기초

```
Test
 1 int buzzer = DO:
2
3 void setup() {
     pinMode(buzzer, OUTPUT);
5|}
7 void loop() {
     digitalWrite(buzzer, HIGH);
     delay(1000);
     digitalWrite(buzzer, LOW);
     delay(1000);
12 }
```

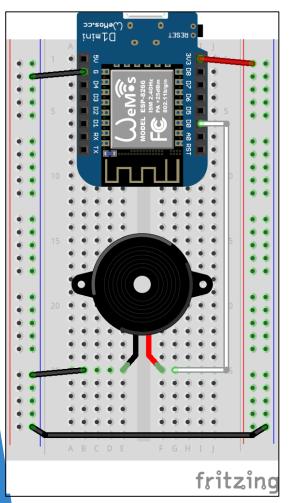
#### 멜로디 실습 기초(2)

- 소리 = 공기 주파수의 진동이기때문에, 주파수에 따라 음높이를 변환 가능.
- tone함수 : tone(핀번호 혹은 이름, 음의 높낮이, 음의 지속시간)

( 단위 : Hz )

옥타브 음계	1	2	3	4	5	6	7	8
C(도)	32.7032	65.4064	130.8128	261.6256	523.2511	1046.502	2093.005	4186.009
C#	34.6478	69.2957	138.5913	277.1826	554.3653	1108.731	2217.461	4434.922
D(레)	36.7081	73.4162	146.8324	293.6648	587.3295	1174.659	2349.318	4698.636
D#	38.8909	77.7817	155.5635	311.1270	622.2540	1244.508	2489.016	4978.032
E(n])	41.2034	82.4069	164.8138	329.6276	659.2551	1318.510	2637.020	5274.041
F(과)	43.6535	87.3071	174.6141	349.2282	698.4565	1396.913	2793.826	5587.652
F#	46.2493	92.4986	184.9972	369.9944	739.9888	1479.978	2959.955	5919.911
G(솔)	48.9994	97.9989	195.9977	391.9954	783.9909	1567.982	3135.963	6271.927
G#	51.9130	103.8262	207.6523	415.3047	830.6094	1661.219	3322.438	6644.875
A(라)	55.0000	110.0000	220.0000	440.0000	880.0000	1760.000	3520.000	7040.000
A#	58.2705	116.5409	233.0819	466.1638	932.3275	1864.655	3729.310	7458.620
B(시)	61.7354	123.4708	246.9417	493.8833	987.7666	1975.533	3951.066	7902.133

#### 멜로디 실습 기초(2)



- 스피커 1개를 연결하여 멜로디 실습을 진행
- 회로 구성
  - 스피커의 -극에 GND
  - 스피커의 +극에 D0를 연결

# 멜로디 실습 기초(2)

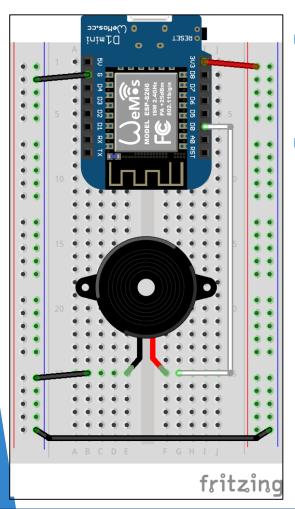
```
Test§
 1 int buzzer = DO;
  void setup() {
     pinMode(buzzer, OUTPUT);
 5
7 void lood() {
     tone(buzzer, 262, 500);
     delay(500);
     tone(buzzer, 294, 500);
     delay(500);
     tone(buzzer, 330, 500);
     delay(500);
     tone(buzzer, 349, 500);
     delay(500);
     tone(buzzer, 392, 500);
     delay(500);
     tone(buzzer, 440, 500);
     delay(500);
     tone(buzzer, 494, 500);
     delay(500);
     tone(buzzer, 523, 500);
     delay(500);
24 }
```

#### 멜로디 응용

주파수에 따라 음높이를 변화 할 수 있다는 점을 고려하면, 음높이의 차이에 따른 "음"을 낼 수 있고 따라서 노래를 비슷하게 제작 할 수 있음.

■ 각각 음과 음 사이에 WeMos 특징인 "Delay"를 발생시켜서 음의 간격을 조정하여 노래를 만들도록 한다.

# 멜로디 응용



■ 스피커 1개를 연결하여 멜로디 실습을 진행

- 회로 구성
  - 스피커의 -극에 GND
  - 스피커의 +극에 D0를 연결

멜로디 응용

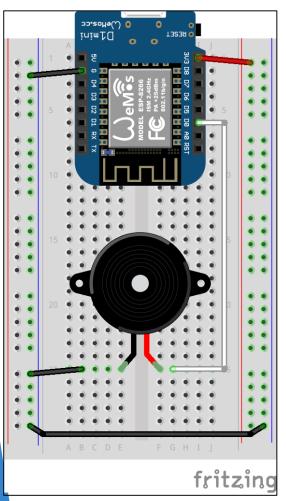
```
22_Lab-06
 1 #define C 262 // 도
2 #define D 294 // 레
3 #define E 330 / □
4 #define F 349 // Ⅲ
5 #define G 392 // 含
6 #define A 440 // 라
7 #define B 494 // 시
9 const int piezoPin = DO; // 부저의 ⊕극
11 int tempo = 200; // duration 옵션 값 설정
12 int notes[25] = { G, G, A, A, G, G, E, G, G, E, E, D, G, G, A, A, G, G, E, G, E, D, E, C };
14 void setup() {
    pinMode (piezoPin, OUTPUT);
16|}
18 void loop() {
19 for (int i = 0; i < 12; i++) {
      tone (piezoPin, notes[ i ], tempo);
      delay (300);
    delay(100); // 멜로디 중간에 짧게 멈추는 용도
    for (int i = 12; i < 25; i++) {
      tone (piezoPin, notes[ i ], tempo);
      delay(300);
28
29 }
```

#### 멜로디 응용(2) - 심화

시리얼 값을 넘겨서, 각각 실행에 대한 멜로디를 실행시키는 방식으로 응용 프로젝트를 제작.

 노래를 본인이 제작 할 수 있다면 충분히 많은 양의 기능들을 제작 할 수 있다.

#### 멜로디 응용(2) - 심화



- 스피커 1개를 연결하여 멜로디 실습을 진행
- 회로 구성
  - 스피커의 -극에 GND
  - 스피커의 +극에 D0를 연결

```
Test
 1 #include "pitches.h"
3 int speakerpin = DO;
 4 int note[] = {2093, 2349, 2637, 2793, 3136, 3520, 3951, 4186};
6 void schoolplay() {
     int melody[] = {NOTE_G7, NOTE_G7, NOTE_A7, NOTE_A7, NOTE_G7,
                    NOTE_G7, NOTE_E7, NOTE_G7, NOTE_G7, NOTE_E7,
                    NOTE_E7, NOTE_D7, NOTE_G7, NOTE_G7, NOTE_A7,
                    NOTE_A7, NOTE_G7, NOTE_G7, NOTE_E7, NOTE_G7,
                    NOTE_E7, NOTE_D7, NOTE_E7, NOTE_C7
                   };
     int noteDurations[] = {4, 4, 4, 4, 4, 4, 2, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 2, 4, 4, 4, 1};
    for (int thisNote = 0; thisNote < 24; thisNote++)</pre>
15
16
       int noteDuration = 1000 / noteDurations[thisNote];
17
       tone(speakerpin, melody[thisNote], noteDuration); //소리를 낸다.
18
       int pauseBetweenNotes = noteDuration * 1.30;
                                                        -//delay 계산식
19
       delay(pauseBetweenNotes);
                                                        //delav
20
      noTone(speakerpin);
                                                        //대상핀 출력 중단
21
22 }
```

```
24 void superplay() {
   int melody2[] = {
     NOTE_E7, NOTE_E7, 1, NOTE_E7, 1, NOTE_C7, NOTE_E7, 1, NOTE_G7, 1, 1, 1, NOTE_G6, 1, 1, 1,
     NOTE_C7, 1, 1, NOTE_G6, 1, 1, NOTE_E6, 1, 1, NOTE_A6, 1, NOTE_B6, 1, NOTE_AS6, NOTE_A6, 1,
     NOTE_GG, NOTE_E7, NOTE_G7, NOTE_A7, 1, NOTE_F7, NOTE_G7, 1, NOTE_F7, 1, NOTE_C7, NOTE_D7, NOTE_B6, 1, 1,
     NOTE_C7, 1, 1, NOTE_G6, 1, 1, NOTE_E6, 1, 1, NOTE_A6, 1, NOTE_B6, 1, NOTE_AS6, NOTE_A6, 1,
30
     NOTE_G6, NOTE_E7, NOTE_G7, NOTE_A7, 1, NOTE_F7, NOTE_G7, 1, NOTE_E7, 1, NOTE_C7, NOTE_D7, NOTE_B6, 1, 1
31
   int noteDurations2[] = {
32
33
     38
   int size = sizeof(melody2) / sizeof(int);
   for (int thisNote = 0; thisNote < size; thisNote++) {</pre>
     int noteDuration = 1000 / noteDurations2[thisNote];
     tone(speakerpin, melody2[thisNote], noteDuration); //소리를 낸다.
     int pauseBetweenNotes = noteDuration * 1.30;
     delay(pauseBetweenNotes);
     noTone(speakerpin);
45
                                          //대상핀 출력 중단
46
47 }
```

# 멜로디 응용 - 심화

```
49 void setup() {
    Serial.begin(9600):
    pinMode(speakerpin, OUTPUT);
52 }
54 void loop() {
    if (Serial.available() > 0) {
      char inputData = Serial.read();
      if (inputData == 's') {
        schoolplay();
59
60
       else if (inputData == 'p') {
        superplay();
62
63
       else if (inputData == 'a') {
64
        int elementCount = sizeof(note) / sizeof(int);
65
        for (int i = 0; i < elementCount; i++) //note를 play
66
          tone(speakerpin, note[i], 500);
          delay(750);
69
70
       else if (inputData == 'n') {
        char temp = Serial.read();
         int sel_note;
74
         if (temp == 'C') {
          sel_note = 0;
```

```
else if (temp == 'D') {
           sel_note = 1;
         else if (temp == 'E') {
           sel_note = 2;
82
         else if (temp == 'F') {
           sel_note = 3;
         else if (temp == 'G') {
           sel_note = 4;
88
         else if (temp == 'A') {
           sel_note = 5;
91
         else {
           sel_note = 6;
94
         tone(speakerpin, note[sel_note], 500);
96
97
98 }
```

# 응용 과제

■ 시리얼 모니터가 아닌 버튼을 사용해서 노래를 제어하도록 응용 가능.

■ 개인이 만들고 싶은 멜로디를 제작 가능.

• Q & A