

## 예제 1)



```
example1 | 아두이노 1.8.12 (Windows Store 1.8.33.0)
파일 편집 스케치 툴 도움말

example1

#define BUZZER 5

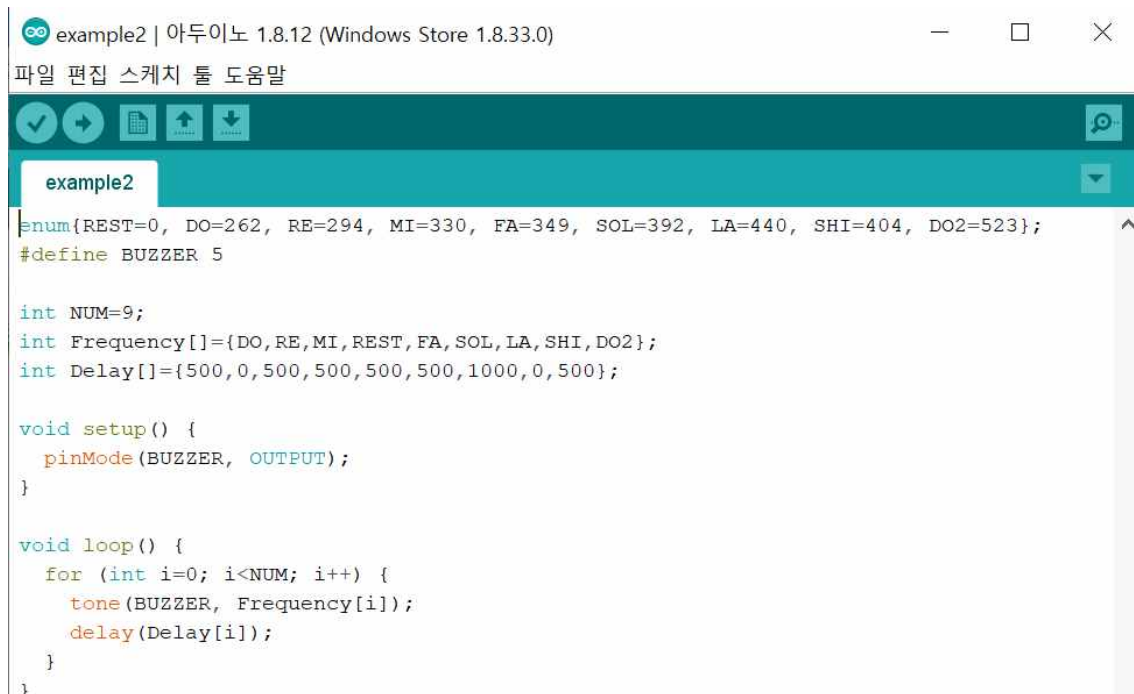
void setup() {
  pinMode(BUZZER, OUTPUT);
}

void loop() {
  tone(BUZZER, 262);
  delay(500);
  tone(BUZZER, 294);
  delay(500);
  tone(BUZZER, 330);
  delay(500);
  tone(BUZZER, 349);
  delay(500);
  tone(BUZZER, 392, 500);
  delay(500);
  tone(BUZZER, 440, 500);
  delay(500);
  tone(BUZZER, 494, 500);
  delay(500);
  tone(BUZZER, 523);
  delay(500);
  noTone(BUZZER);
  delay(500);
}
```

주파수 별로 게이름이 있고. BUZZER모듈을 통해 출력되는 주파수를 게이름으로 맞추어 도레미파솔라시도를 출력하는 매우 간단한 예제이다. 도부터 파 까지는 TONE함수의 변수를 2개만 사용하였고, 솔부터 시까지는 3개의 변수를 사용하였다.

도~도 까지 출력하는 예제에서는 침표가 없기에 noTone의 선언이 필요없는 두 번째 방식과 첫 번째 방식의 차이가 느껴지지 않았다. 맨 마지막 도를 출력하고 나서는 약간의 침을 주기 위해서 500m초의 딜레이를 발생시켰다.

예제2)



```
enum{REST=0, DO=262, RE=294, MI=330, FA=349, SOL=392, LA=440, SHI=404, DO2=523};
#define BUZZER 5

int NUM=9;
int Frequency[]={DO, RE, MI, REST, FA, SOL, LA, SHI, DO2};
int Delay[]={500, 0, 500, 500, 500, 500, 1000, 0, 500};

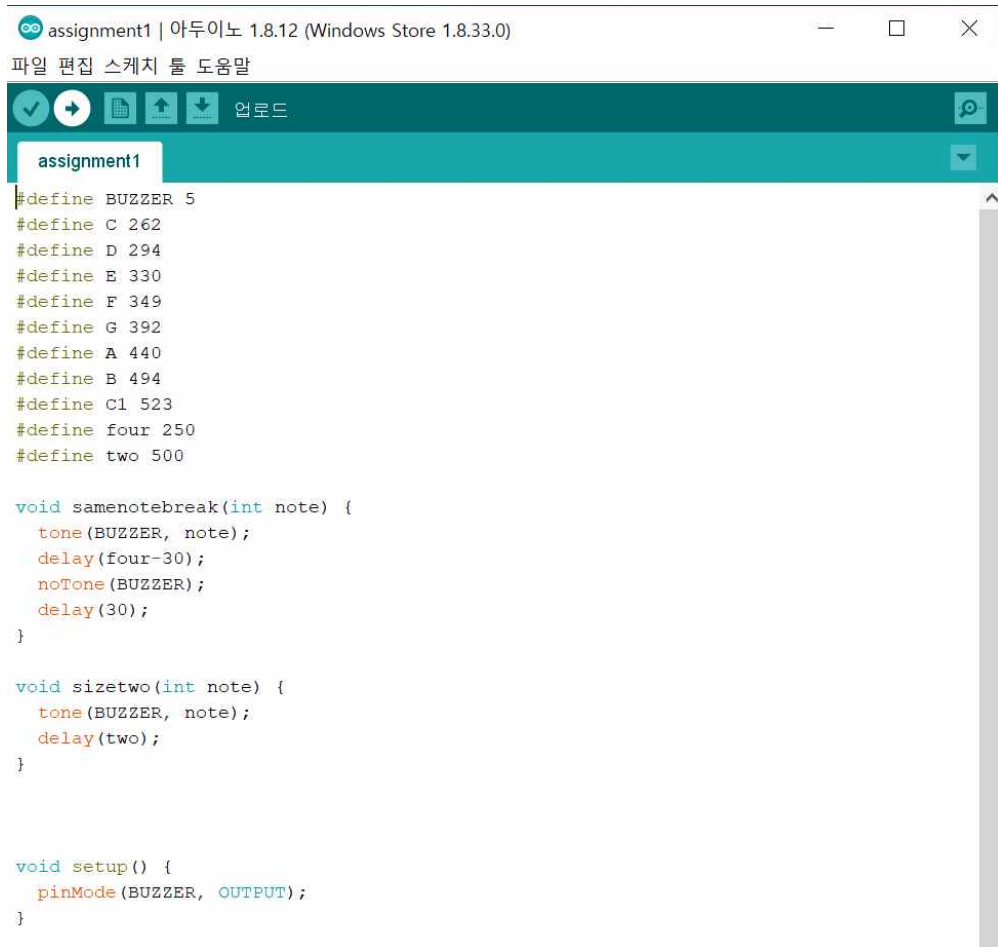
void setup() {
  pinMode(BUZZER, OUTPUT);
}

void loop() {
  for (int i=0; i<NUM; i++) {
    tone(BUZZER, Frequency[i]);
    delay(Delay[i]);
  }
}
```

일일이 루프에서 음 하나하나를 출력하던 예제 1번과는 달리 도~도를 출력하기 위해 주파수마다 대응되는 게이름 값을 정의를 해주었다. 또한 도~도 까지 출력되기 위한 8가지 음을 효과적으로 출력하기 위해 배열을 통해 그 음이 유지되는 시간과 출력주파수를 만들었다.

따라서 레, 시는 delay가 0이므로 출력되지 않는다. 또한 미와 파 사이에 있는 REST는 주파수가 0 이므로 500m초 동안 지속은 되지만 아무 소리가 나지 않는다. 또한 라는 다른 음보다 두 배의 delay가 설정돼있으므로 소리가 두 배 더 길게 나는 것을 확인할 수 있었다.

## 실습과제1)



```
#define BUZZER 5
#define C 262
#define D 294
#define E 330
#define F 349
#define G 392
#define A 440
#define B 494
#define C1 523
#define four 250
#define two 500

void samenotebreak(int note) {
    tone(BUZZER, note);
    delay(four-30);
    noTone(BUZZER);
    delay(30);
}

void sizetwo(int note) {
    tone(BUZZER, note);
    delay(two);
}

void setup() {
    pinMode(BUZZER, OUTPUT);
}
```

도레미파솔라시도 까지의 음을 define을 통해 대응되는 주파수를 간편하게 알파벳으로 정의하였다. 그리고 학교종 음악에서 음표의 길이가 4분음표, 2분음표 두 개가 나오는데 이때의 길이 또한 define을 통해 적당한 속도로 정의하였다.

학교종이 땡땡땡을 출력할 때의 음은 솔솔라라 솔솔미 인데 솔솔, 라라처럼 붙어있는 음이 있을 시에는 약간의 쉼표를 주지 않으면 음이 가령 500msec만큼 두 번 출력하는 경우더라도 1000msec만큼 하나의 음처럼 출력되는 것을 확인하였다. 따라서 samenotebreak라는 함수를 통해 4분음표의 음이 두 번 연속으로 나온다면 약간의 noTone시간이 발생하여서 음이 끊어지는 것을 느껴지게 할 수 있게 하였다.

또한 sizetwo 함수는 2분음표의 노트를 출력하는 함수이다.

```

void loop() {
  samenotebreak(G);
  samenotebreak(G);
  samenotebreak(A);
  samenotebreak(A);
  samenotebreak(G);
  samenotebreak(G);
  sizetwo(E);
  samenotebreak(G);
  samenotebreak(G);
  samenotebreak(E);
  samenotebreak(E);
  sizetwo(D);
  tone(BUZZER, D);
  delay(four);
  noTone(BUZZER);
  delay(four);

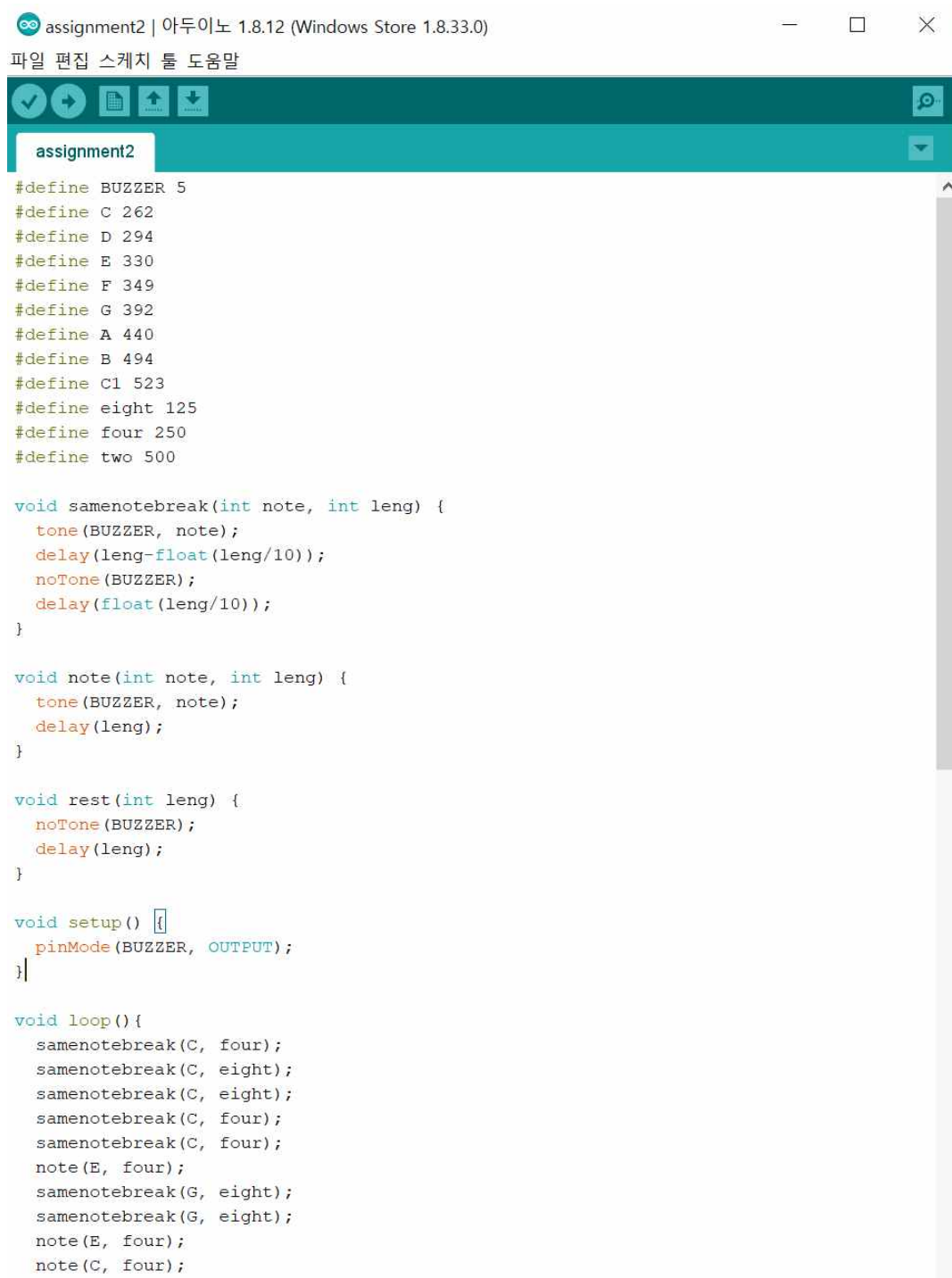
  samenotebreak(G);
  samenotebreak(G);
  samenotebreak(A);
  samenotebreak(A);
  samenotebreak(G);
  samenotebreak(G);
  sizetwo(E);
  tone(BUZZER, G);
  delay(four);
  tone(BUZZER, E);
  delay(four);
  tone(BUZZER, D);
  delay(four);
  tone(BUZZER, E);
  delay(four);

  sizetwo(C);
  tone(BUZZER, C);
  delay(four);
  noTone(BUZZER);
  delay(four);
  delay(two);
}

```

출력할 때 2분음표의 .이 붙어있는 것은 3박자만큼 음을 출력하는 것인데 이때는 2분음표를 출력하고 그 다음 또 4분음표를 출력하는 식으로 설정하였다.

## 실습과제2)



```
#define BUZZER 5
#define C 262
#define D 294
#define E 330
#define F 349
#define G 392
#define A 440
#define B 494
#define C1 523
#define eight 125
#define four 250
#define two 500

void samenotebreak(int note, int leng) {
    tone(BUZZER, note);
    delay(leng-float(leng/10));
    noTone(BUZZER);
    delay(float(leng/10));
}

void note(int note, int leng) {
    tone(BUZZER, note);
    delay(leng);
}

void rest(int leng) {
    noTone(BUZZER);
    delay(leng);
}

void setup() {
    pinMode(BUZZER, OUTPUT);
}

void loop(){
    samenotebreak(C, four);
    samenotebreak(C, eight);
    samenotebreak(C, eight);
    samenotebreak(C, four);
    samenotebreak(C, four);
    note(E, four);
    samenotebreak(G, eight);
    samenotebreak(G, eight);
    note(E, four);
    note(C, four);
}
```

실습과제 1과 크게 다르지 않지만 함수를 몇 개 추가하였다. samenotebreak에서는 leng이라는 변수를 추가로 받아서 해당 음표가 굳이 4분음표가 아닐 시에도 해당 음 길이의 10분의 1만큼은 쉴 수 있게 설정을 하였다.

또한 음을 출력하는 note 함수와 쉴 때 아무것도 출력하지 않는 rest 함수를 통해서 loop에서의 코드길이가 조금은 더 짧아질 수 있게 하였다.

```

samenotebreak(G, eight);
samenotebreak(G, eight);
note(E, four);
samenotebreak(G, eight);
samenotebreak(G, eight);
note(E, four);
samenotebreak(C, four);
samenotebreak(C, four);
samenotebreak(C, two);

samenotebreak(G, four);
samenotebreak(G, four);
note(E, four);
note(C, four);
samenotebreak(G, four);
samenotebreak(G, four);
samenotebreak(G, two);

samenotebreak(G, four);
samenotebreak(G, four);
note(E, four);
note(C, four);
samenotebreak(G, four);
samenotebreak(G, four);
samenotebreak(G, two);

samenotebreak(G, four);
samenotebreak(G, four);
note(E, four);
note(C, four);
samenotebreak(G, eight);
samenotebreak(G, eight);
note(G, eight);
note(A, eight);
note(G, two);

note(C1, four);
note(G, four);
note(C1, four);
note(G, four);
note(E, four);
note(D, four);
note(C, two);
rest(1000);
}

```

마지막에 rest(1000)을 넣은 것은 곡이 끝났다는 것을 심표를 길게 출력해서 표현한 것이다.  
 각 예제별 동영상은 폴더안에 따로 첨부하였습니다!