

Buzzer Control

- Buzzer Control
- 후방 감지기 만들기

Buzzer Control

1. Buzzer의 구동 원리

- ❖ 피에조(Piezo)란 외부의 압력 변화에 전압이 발생하고, 반대로 전압이 걸리면 물리 변형이 발생하는 것을 말한다.
- ❖ 전자를 압전 효과, 후자를 역압전 효과라고 하며, 이런 성질을 활용하여 피에조 부저(Buzzer)는 전기신호를 받아 음파(진동)을 발생시켜 소리를 내도록 하는 장치로 사용된다. 이 부저는 얼마만큼의 주파수를 전달해 주는가에 따라 출력되는 소리가 달라지게 되며, 부저의 종류에 따라 주파수에 따른 소리의 높낮이를 가진다.

계이름	주파수(Hz)	계이름	주파수(Hz)	계이름	주파수(Hz)
도	261,625	도	523,25	도	1046,5
도#	277,18	도#	554,36	도#	1108,72
레	293,665	레	587,33	레	1174,66
레#	311,125	레#	622,25	레#	1244,5
미	329,63	미	659,26	미	1318,52
파	349,23	파	698,46	파	1396,92
파#	369,995	파#	739,99	파#	1479,98
솔	392	솔	784	솔	1568
솔#	415,3	솔#	830,6	솔#	1661,2
라	440	라	880	라	1760
라#	466,1	라#	932,2	라#	1864,4
시	493,8	시	987,6	시	1975,2
				도	2093

1. Buzzer의 구동 원리 (계속)

- ❖ 앞의 주파수에 맞게 피에조에 출력하여야만 원하는 소리를 낼 수 있다.
- ❖ 아두이노 보드의 디지털입출력 핀을 사용하여 주파수 신호를 만들어 내려면, 주파수와 시간과의 개념을 이해하고 신호를 발생시켜야 한다.
- ❖ 주파수와 시간과의 관계는 아래와 같다.

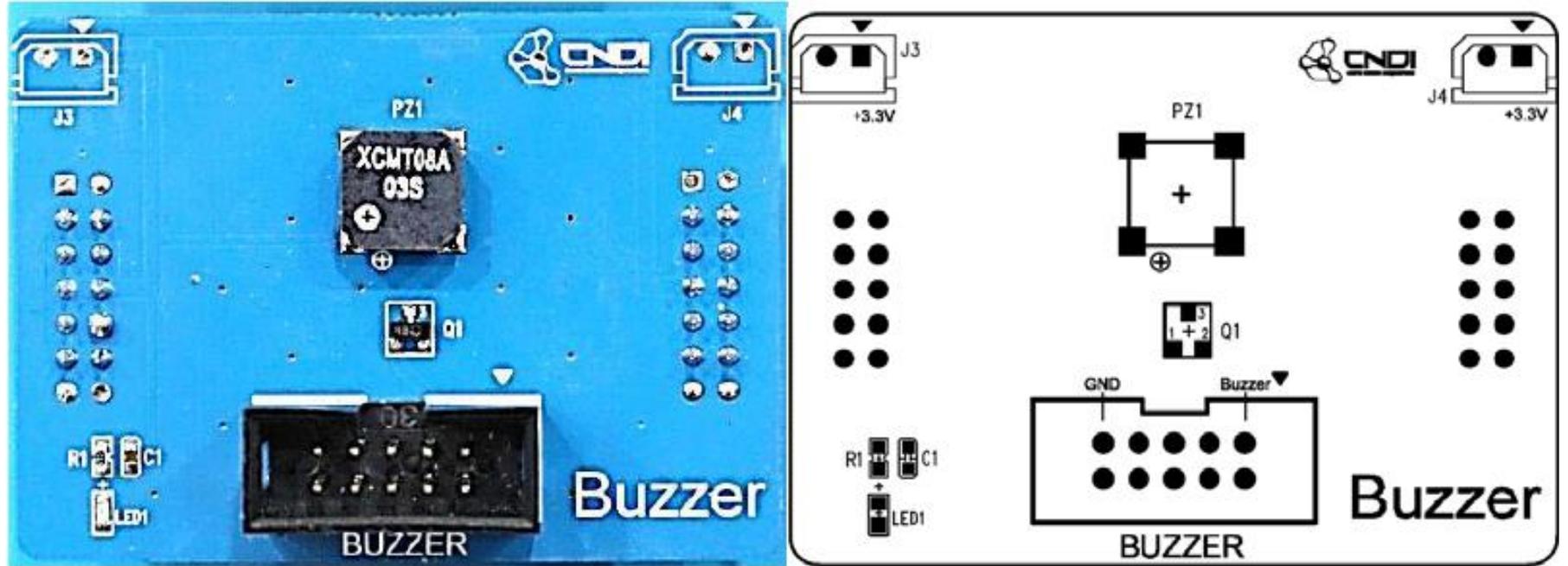
$$T = \frac{1}{f}$$

- ❖ 위 식에서 T 는 시간이면 f 는 주파수이다. 예를 들어 위의 표에서 4옥타브 솔 음은 390Hz 이다. 이 주파수를 위 공식에 대입해보면

$$T = \frac{1}{390Hz} = 2564.10[\mu s]$$

- ❖ 위와 같이 2564.10 us 의 주기를 가지게 되는 것이다. 이 주기 마다 파형을 한 번씩 출력하여야만 4옥타브 솔 음을 피에조 부저를 통해 낼 수 있다.

1.1 피에조 Buzzer 모듈



1.2 Arduino에서 Buzzer Control 함수: tone() 함수

❖ tone() 함수

- 😊 핀에 특정 주파수의 사각파(50% 듀티 사이클)를 생성하는 함수
- 😊 noTone() 함수를 호출하기 전까지 생성된 톤이 계속 유지된다.
- 😊 피에조 부저 또는 톤을 재생하는 다른 스피커를 제어하는데 사용될 수 있다.
- 😊 tone() 함수는 오직 하나의 핀에서 동작한다.
- 😊 만약 tone함수를 미리 다른 핀에서 재생하고 있는 경우, tone함수를 호출하여도 아무런 동작을 하지 않는다.
- 😊 같은 핀에서 다시 호출되는 경우 호출 주파수를 변경한다.
- 😊 tone() 함수는 31Hz 이하의 주파수는 생성할 수 없다.
- 😊 tone() 함수의 인자는 2개 또는 3개이다.
 - tone(pin, frequency)
 - tone(pin, frequency, duration)
 - pin : pin number
 - frequency : 주파수
 - duration : 재생 시간 (msec)

1.3 Arduino에서 Buzzer Control 함수: noTone 함수

❖ noTone(pin)

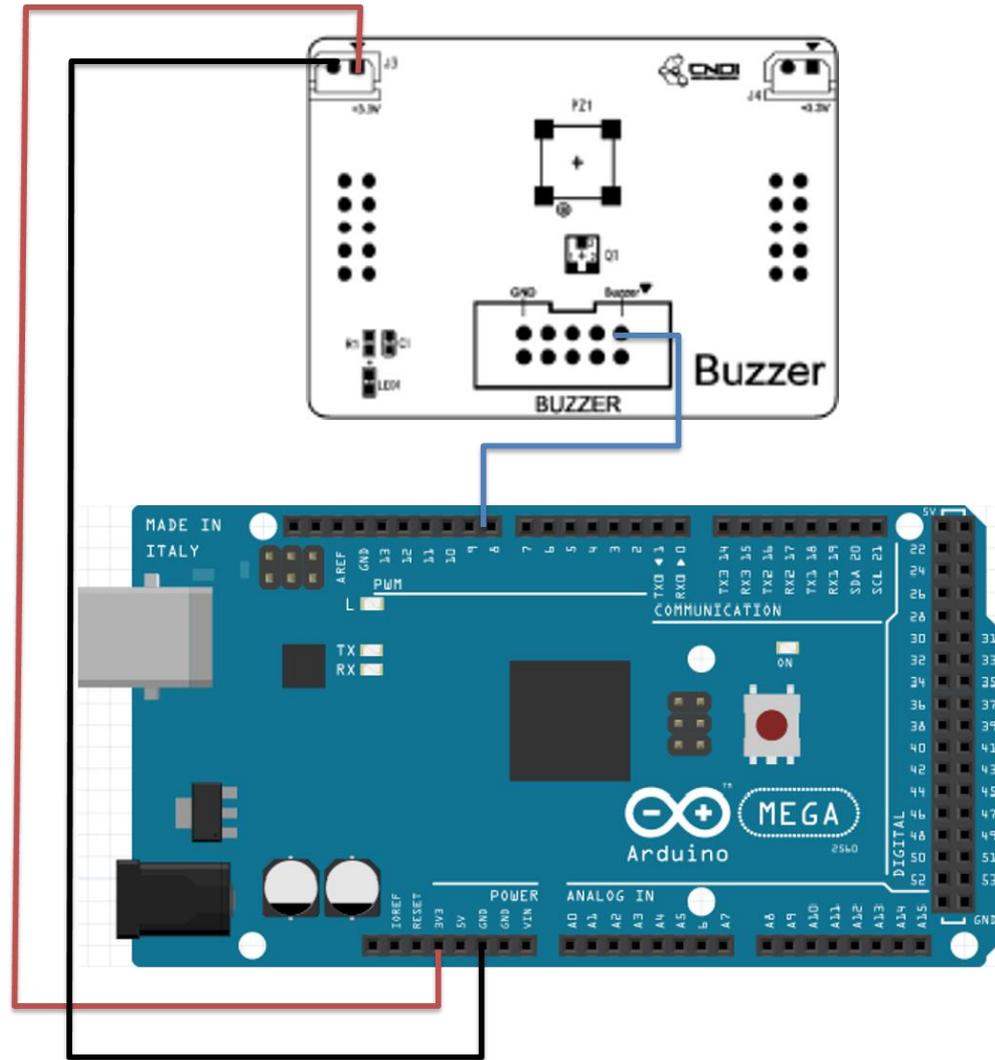
- 😊 tone() 함수에 의해 트리거된 사각파를 중지하는데 사용되는 함수 이다.
- 😊 tone이 생성중인 경우에는 영향을 주지 않는다. 만약 여러 개의 핀에 다른 피치를 재생하려는 경우 다른 핀에 tone() 함수를 호출하기 전에 noTone()을 호출해야 한다.

1.4 Buzzer_Control1 예제 핀 배열

- ❖ tone() 함수와 noTone() 함수를 이용하여 도~시 까지 반복하는 프로그램

Device	MegaADK	Buzzer
PIN NO	8	Buzzer

1.5 Buzzer_Control1 예제 핀 결선



1.6 Buzzer_Control1 예제

```
int BuzzerPin = 8; //Buzzer와 연결된 핀 정의
void setup(){
    pinMode(BuzzerPin,OUTPUT); //Buzzer 핀 출력으로 설정
}

void loop(){
    tone(BuzzerPin,262);
    delay(500);

    noTone(BuzzerPin);
    delay(500);

    tone(BuzzerPin,294);
    delay(500);
```

1.6 Buzzer_Control1 예제 (계속)

```
noTone(BuzzerPin);  
delay(500);  
tone(BuzzerPin,330);  
delay(500);  
noTone(BuzzerPin);  
delay(500);  
tone(BuzzerPin,349);  
delay(500);  
noTone(BuzzerPin);  
delay(500);  
tone(BuzzerPin,392);  
delay(500);  
noTone(BuzzerPin);  
delay(500);
```

1.6 Buzzer_Control1 예제 (계속)

```
tone(BuzzerPin,440);  
delay(500);  
noTone(BuzzerPin);  
delay(500);  
tone(BuzzerPin,494);  
delay(500);  
noTone(BuzzerPin);  
delay(500);  
tone(BuzzerPin,523);  
delay(500);  
noTone(BuzzerPin);  
delay(500);  
}
```

1.7 Buzzer_Control1 예제 설명

- ❖ Buzzer와 연결된 아두이노 핀을 출력으로 설정
- ❖ loop() 함수에서 tone() 함수를 통해 4옥타브 도부터 5옥타브 도까지의 음을 500ms 동안 출력하고 500ms 지연하는 동작을 반복

2. Buzzer로 학교종 연주하기

1. 학 교 종 이 땡 땡 땡 슬 슬 미 미 레
2. 학 교 종 이 땡 땡 땡 어 서 모 이 자

The first line of music is written on a single staff in 4/4 time. It consists of 11 notes: G4, A4, B4, A4, G4, F4, E4, D4, C4, B3, A3. The notes are grouped into four measures: the first measure has G4, A4, B4, A4; the second has G4, F4, E4, D4; the third has D4, C4, B3, A3; and the fourth has a whole note A3. The lyrics are written below the staff, with two lines of text: '1. 학 교 종 이 땡 땡 땡' and '2. 학 교 종 이 땡 땡 땡' for the first six notes, and '어 서 모 이 자' for the last five notes. The notes '슬', '슬', '미', '미', '레' are written in pink above the staff.

5. 슬 슬 라 라 슬 슬 미 슬 미 레 미 도
선 생 님 이 우 리 를 기 다 리 신 다.
사 생 이 님 게 오 리 는 도 공 부 잘 하 자.

The second line of music is written on a single staff in 4/4 time. It consists of 11 notes: G4, A4, B4, A4, G4, F4, E4, D4, C4, B3, A3. The notes are grouped into four measures: the first measure has G4, A4, B4, A4; the second has G4, F4, E4, D4; the third has D4, C4, B3, A3; and the fourth has a whole note A3. The lyrics are written below the staff: '선 생 님 이 우 리 를 기 다 리 신 다.' and '사 생 이 님 게 오 리 는 도 공 부 잘 하 자.' The notes '슬', '슬', '라', '라', '슬', '슬', '미', '슬', '미', '레', '미', '도' are written in pink above the staff.

위 악보는 4옥타브의 음계를 사용하며 이 음계들의 주파수를 정의하고 배열로 정리하여 tone() 함수로 넘겨주어 연주하게 된다.

2.1 Buzzer_Control2 예제

```
int piezo = 8;

long Do = 262, Re = 294, mi=330, pa=349, sol=392, ra=440, si=494;

long music_data[24]={sol, sol, ra, ra, sol, sol, mi, sol, sol, mi, mi, Re,
sol, sol, ra, ra, sol, sol, mi, sol, mi, Re, mi, Do};

long DelayTime[24]={400, 400, 400, 400, 400, 400, 600, 400, 400, 400, 400,
800, 400, 400, 400, 400, 400, 400, 600, 400, 400, 400, 400, 600};

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

2.2 Buzzer_Control2 예제

```
void loop(){
    for(int j=0; j<24; j++)
    {
        tone(piezo,music_data[j]);
        delay(300);
        noTone(piezo);
        delay(DelayTime[j]);
    }
    delay(2000);
}
```

후방감지기 만들기

1. Rear Sensor by PSD 예제 핀 배열

- ❖ PSD 센서의 입력 신호를 A/D 컨버팅 하여 그 거리 값에 따라 거리 가까워 질수록 부저를 울리는 주기를 빠르게 하여 실제 후방감지기의 동작과 유사한 동작을 하는 스케치 프로그램
- ❖ PSD 모듈과 buzzer 모듈을 이용하여 구현

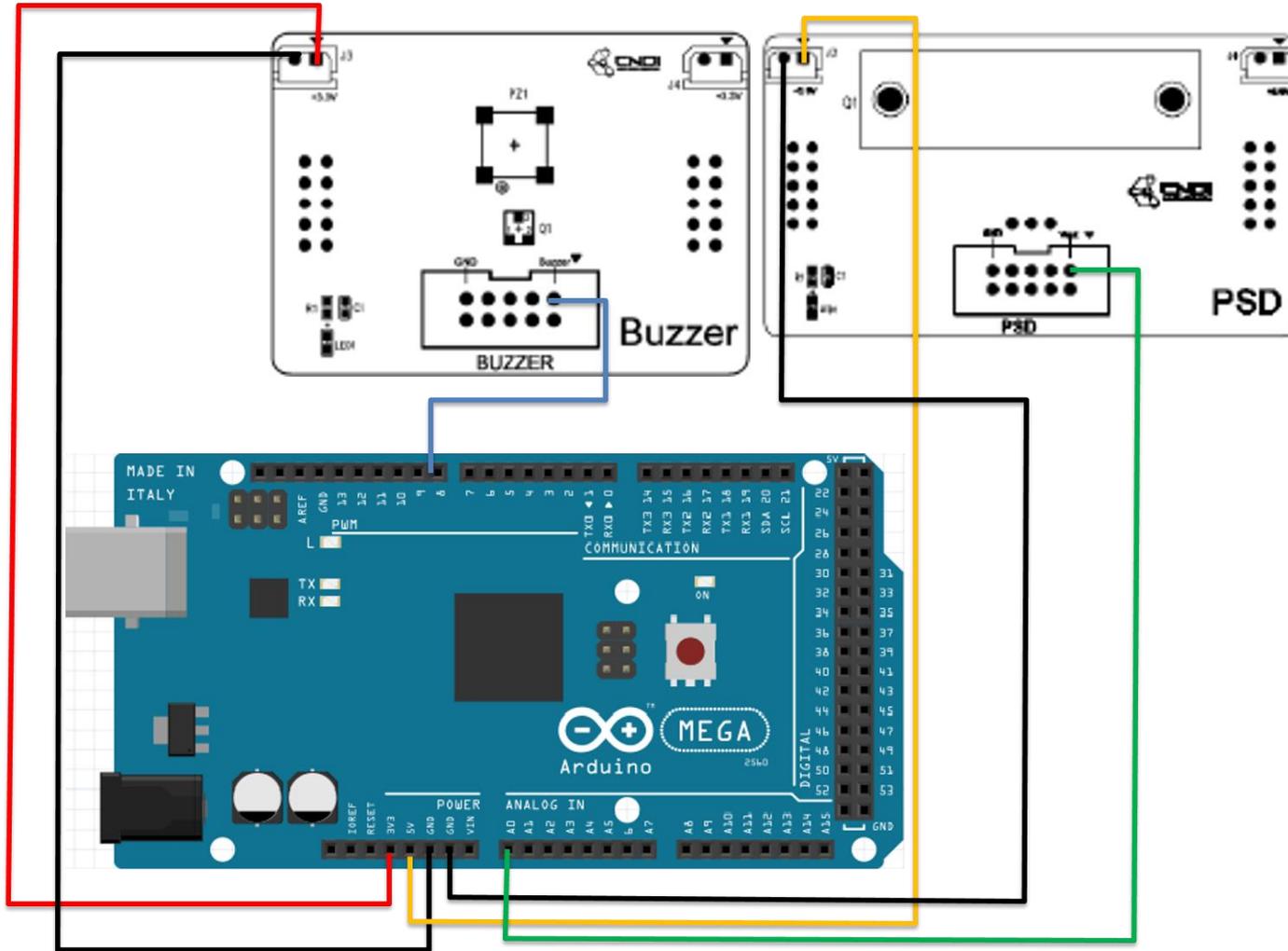
Buzzer 핀 결선

Device	MegaADK	Buzzer
PIN NO	8	Buzzer

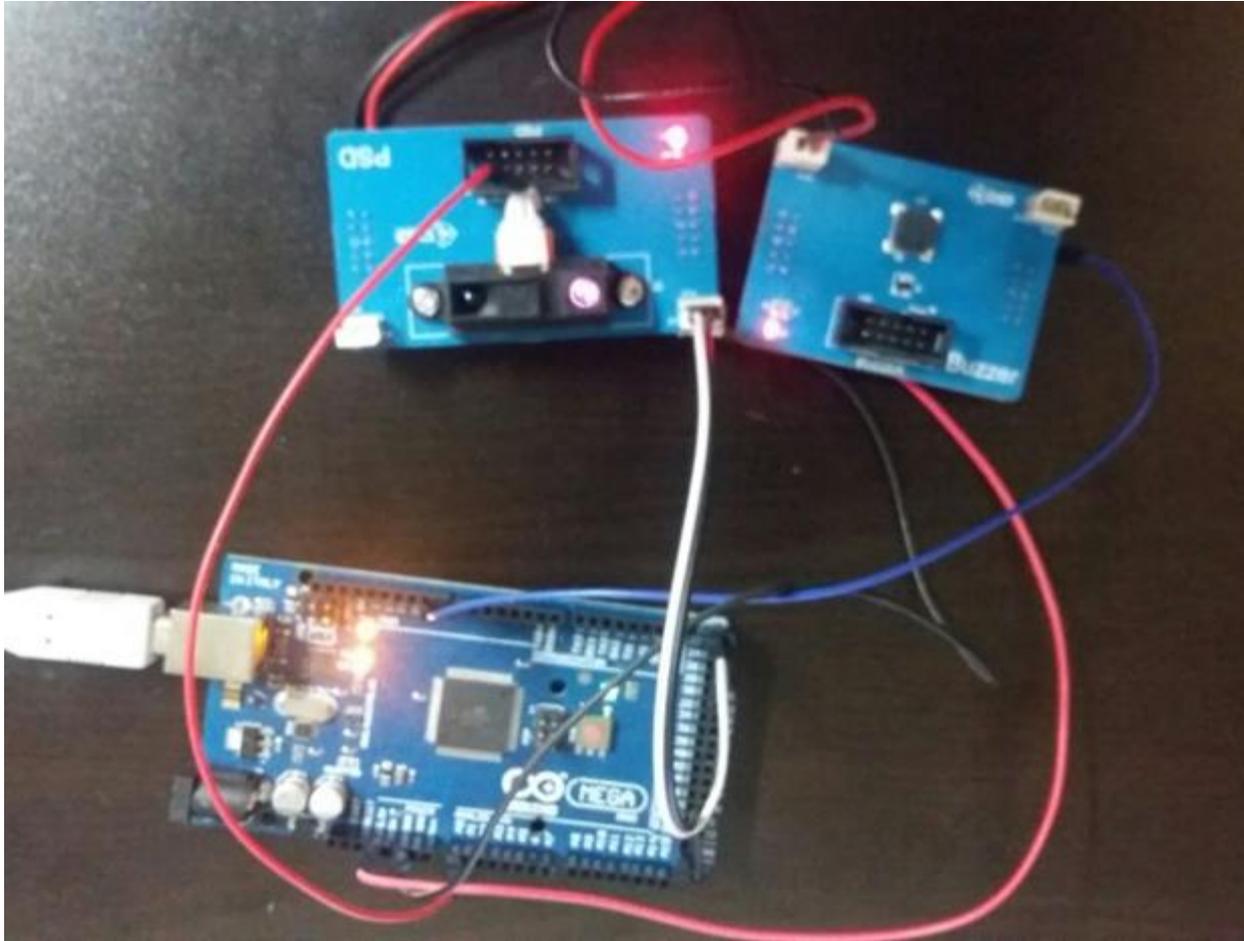
PSD 핀 결선

Device	MegaADK	PSD
PIN NO	A0	PSD Vout

1.1 Rear Sensor by PSD 예제 핀 결선



1.2 Rear Sensor by PSD 예제 실제 핀 결선



1.3 RearSensorbyPSD 예제

```
#include <MsTimer2.h> //타이머관련 라이브러리 선언
int BuzzerPin = 8; //Buzzer 와 연결된 핀 변수 선언 및 초기화
unsigned int g_unSensorValue = 0; //PSD 센서값 관련 변수 선언 및 초기화
void setup(){
    pinMode(BuzzerPin,OUTPUT); //디지털 입출력 A8 번 핀 출력으로 설정
    MsTimer2::set(50,SensorRead); //타이머 설정,50ms 주기로 SensorRead()
                                호출
    MsTimer2::start();
}
void SensorRead(){
    g_unSensorValue = analogRead(A0); //A0 핀을 A/D 컨버팅하여 변수에 저장
}
```

1.3 RearSensorbyPSD 예제 (계속)

```
void BuzzerContol(int Freq, int PlayTime){
    tone(BuzzerPin,Freq);
    delay(PlayTime);
}

void BuzzerStop(){
    noTone(BuzzerPin); //부저 정지
}

void Buzzer_sequence(int PlayTime){
    BuzzerContol(523,PlayTime); //부저 제어 함수 호출
    delay(PlayTime); //PlayTime 만큼 지연
    BuzzerStop(); //부저 정지 함수 호출
}
```

1.3 RearSensorbyPSD 예제 (계속)

//PSD 센서 값을 비교 하여 부저를 제어하는 함수에 올리는 주기를 달리할 수 있도록 변수로 넘겨줌

```
void RearSensorbyPSD_sequence (void){  
    if (g_unSensorValue < 600 && g_unSensorValue >= 460) {Buzzer_sequence(60);}  
    else if (g_unSensorValue < 460 && g_unSensorValue >= 350) {Buzzer_sequence(100);}  
    else if (g_unSensorValue < 350 && g_unSensorValue >= 230) {Buzzer_sequence(200);}  
    else if (g_unSensorValue < 230 && g_unSensorValue >= 180) {Buzzer_sequence(300);}  
    else if (g_unSensorValue < 180 && g_unSensorValue >= 150) {Buzzer_sequence(400);}  
    else {}  
}  
  
void loop(){  
    RearSensorbyPSD_sequence(); //부저 주기 검출 함수 호출  
}
```