

# ESP32 초음파센서 & 부저 제어 실습지시서

## 준비물

- ESP32 개발 보드 (예: ESP32 Dev Module)
- 부저 1개
- 초음파센서 1개
- 브레드보드와 점퍼 와이어
- USB 케이블 (ESP32와 PC 연결용)

## 공통 회로 연결법

- 초음파센서 TRIG 신호 → ESP32 GPIO 핀 (실습별 지정)
- 초음파센서 ECHO 신호 → ESP32 GPIO 핀 (실습별 지정)
- 초음파센서 접지 → GND
- 초음파센서 전원 → VCC

## 프로젝트 A: 초음파 거리 감지 경고 부저 시스템

1. 준비
  - 사용 부품: 초음파센서 1개, 부저 1개, 점퍼선, ESP32
2. 배선
  - 초음파센서(TRIG): 12번, 초음파센서(ECHO): 14번, 부저: 2번
3. 업로드
  - 코드 스케레톤

```
#include <Arduino.h>
const int TRIG = 12;    // 초음파 센서의 Trig 핀
const int ECHO = 14;    // 초음파 센서의 Echo 핀
const int BUZ = 2;      // 부저 핀
void setup() {
  Serial.begin(115200);  // 시리얼 모니터 시작 (통신속도 115200bps)
  pinMode(TRIG, OUTPUT); // Trig 핀 출력 설정
  pinMode(ECHO, INPUT);  // Echo 핀 입력 설정
  pinMode(BUZ, OUTPUT);  // 부저 핀 출력 설정
}
void loop() {
  // --- 초음파 센서 신호 보내기 ---
  digitalWrite(TRIG, LOW); delayMicroseconds(2);
  digitalWrite(TRIG, HIGH); delayMicroseconds(10);
  digitalWrite(TRIG, LOW);
  // --- Echo 핀에서 반사 시간 측정 ---
  float distance = pulseIn(ECHO, HIGH) * 0.034 / 2; // 거리(cm) 계산
  Serial.print("Distance: "); Serial.print(distance); Serial.println(" cm");
  // --- 10cm 이하일 때 경고음 ---
  if (distance < 10) tone(BUZ, 1000); // 1kHz 소리 발생
  else noTone(BUZ);                 // 부저 끄
  delay(300);
}
```

4. 관찰
  - 초음파 센서(HC-SR04)를 사용해 물체와의 거리를 측정하고, 그 거리가 **10cm 이하로 가까워졌을 때** 부저에서 **경고음(삐—)** 을 내는 간단한 감지 시스템
  - **Trig** 핀으로 초음파 신호를 짧게(10μs) 보냅니다.
  - **Echo** 핀은 반사되어 돌아온 초음파 신호의 시간을 측정합니다.
  - 그 시간을 이용해 거리를 계산합니다.
  - 계산된 거리가 **10cm 미만**이면 부저에서 1kHz의 경고음을 냅니다.
  - 10cm 이상이면 부저를 끄고 다시 거리 측정을 반복합니다.

## 프로젝트 B: 거리-주파수 변조형 초음파 부저 시스템

1. 준비
  - 사용 부품: 초음파센서 1개, 부저 1개, 점퍼선, ESP32
2. 배선
  - 초음파센서(TRIG): 12번, 초음파센서(ECHO): 14번, 부저: 2번
3. 업로드
  - 코드 스케레톤

```
#include <Arduino.h>
const int TRIG = 12; // 초음파 센서의 Trig 핀
const int ECHO = 14; // 초음파 센서의 Echo 핀
const int BUZ = 2; // 부저 핀
void setup() {
  Serial.begin(115200); // 시리얼 모니터 시작 (통신속도 115200bps)
  pinMode(TRIG, OUTPUT); // Trig 핀 출력 설정
  pinMode(ECHO, INPUT); // Echo 핀 입력 설정
  pinMode(BUZ, OUTPUT); // 부저 핀 출력 설정
}
void loop() {
  // --- 초음파 센서 신호 보내기 ---
  digitalWrite(TRIG, LOW); delayMicroseconds(2);
  digitalWrite(TRIG, HIGH); delayMicroseconds(10);
  digitalWrite(TRIG, LOW);
  // --- Echo 핀에서 반사 시간 측정 ---
  float distance = pulseIn(ECHO, HIGH) * 0.034 / 2; // 거리(cm) 계산
  // --- 거리값을 주파수로 변환 (5~50cm → 2000~200Hz) ---
  int freq = map(distance, 5, 50, 2000, 200);
  freq = constrain(freq, 200, 2000);
  tone(BUZ, freq, 100); // --- 주파수에 따라 소리 발생 ---
  Serial.print("Distance: "); Serial.print(distance);
  Serial.print(" cm | Tone: "); Serial.println(freq);
  delay(200);
}
```

4. 관찰
  - 초음파 센서(HC-SR04) 로 측정한 물체와의 거리에 따라 부저 소리의 높낮이(주파수)가 자동으로 변하는 프로젝트
  - Trig 핀에서 초음파를 짧게 발사
  - Echo 핀에서 반사된 신호의 시간을 측정
  - 거리(cm) 를 계산한 뒤, 그 값을 map() 함수를 이용해 주파수(Hz) 로 변환
    - 가까울수록 높은 음 (최대 2000Hz)
    - 멀수록 낮은 음 (최소 200Hz)
  - tone() 함수를 이용해 변환된 주파수로 부저가 소리를 냄
  - 시리얼 모니터에는 거리(cm) 와 현재 발생 중인 주파수(Hz) 가 함께 출력

## 프로젝트 C: 거리 단계별 비프음 경고 시스템

1. 준비
  - 사용 부품: 초음파센서 1개, 부저 1개, 점퍼선, ESP32
2. 배선
  - 초음파센서(TRIG): 12번, 초음파센서(ECHO): 14번, 부저: 2번
3. 업로드
  - 코드 스케레톤

```
#include <Arduino.h>
const int TRIG = 12; // 초음파 센서의 Trig 핀
const int ECHO = 14; // 초음파 센서의 Echo 핀
const int BUZ = 2; // 부저 핀
void setup() {
  Serial.begin(115200); // 시리얼 모니터 시작 (통신속도 115200bps)
  pinMode(TRIG, OUTPUT); // Trig 핀 출력 설정
  pinMode(ECHO, INPUT); // Echo 핀 입력 설정
  pinMode(BUZ, OUTPUT); // 부저 핀 출력 설정
}
void loop() {
  // --- 초음파 센서 신호 보내기 ---
  digitalWrite(TRIG, LOW); delayMicroseconds(2);
  digitalWrite(TRIG, HIGH); delayMicroseconds(10);
  digitalWrite(TRIG, LOW);
  // --- Echo 핀에서 반사 시간 측정 ---
  float distance = pulseIn(ECHO, HIGH) * 0.034 / 2; // 거리(cm) 계산
  Serial.print("Distance: "); Serial.print(distance);
  // --- 거리 단계별로 비프 속도 조절 ---
  if (distance < 10) { // 매우 가까움
    tone(BUZ, 1000, 100); delay(100); }
  else if (distance < 20) { // 약간 가까움
    tone(BUZ, 800, 100); delay(300); }
  else if (distance < 40) { // 중간 거리
    tone(BUZ, 600, 100); delay(600); }
  else { // 멀리 있음
    noTone(BUZ); delay(500); }
}
```

4. 관찰
  - 초음파 센서(HC-SR04) 로 물체까지의 거리를 측정하고, 그 거리에 따라 부저의 비프음(삐삐삐) 속도와 음 높이를 단계별로 조절하는 시스템
  - 시간으로부터 거리(cm) 를 계산
  - 계산된 거리값에 따라 부저의 주파수(음 높이) 와 반복 속도(비프 간격) 를 설정
  - 매우 가까움(10cm 미만): 빠른 비프음 + 높은 음 (1000Hz)
  - 약간 가까움(10~20cm): 중간 속도 비프 + 중간 음 (800Hz)
  - 중간 거리(20~40cm): 느린 비프 + 낮은 음 (600Hz) /
  - 멀리 있음(40cm 이상): 부저 꺼짐 (무음)

## 프로젝트 D: 시간 기반 거리 기록 및 시각화 초음파 시스템

1. 준비
  - 사용 부품: 초음파센서 1개, 부저 1개, 점퍼선, ESP32
2. 배선
  - 초음파센서(TRIG): 12번, 초음파센서(ECHO): 14번, 부저: 2번
3. 업로드 - 코드 스켈레톤

```
#include <Arduino.h>
const int TRIG = 12;    // 초음파 센서의 Trig 핀
const int ECHO = 14;    // 초음파 센서의 Echo 핀
const int BUZ  = 2;     // 부저 핀
unsigned long prevTime = 0;

void setup() {
  Serial.begin(115200);    // 시리얼 모니터 시작 (통신속도 115200bps)
  pinMode(TRIG, OUTPUT);  // Trig 핀 출력 설정
  pinMode(ECHO, INPUT);   // Echo 핀 입력 설정
  pinMode(BUZ, OUTPUT);   // 부저 핀 출력 설정
}

void loop() {
  if(millis() - prevTime >= 1000) {
    prevTime = millis();
    // --- 초음파 센서 신호 보내기 ---
    digitalWrite(TRIG, LOW); delayMicroseconds(2);
    digitalWrite(TRIG, HIGH); delayMicroseconds(10);
    digitalWrite(TRIG, LOW);
    // --- Echo 핀에서 반사 시간 측정 ---
    float distance = pulseIn(ECHO, HIGH) * 0.034 / 2; // 거리(cm) 계산
    // --- 거리 출력 ---
    Serial.print("Time: ");    Serial.print(prevTime / 1000);
    Serial.print("s | Distance: ");    Serial.print(distance);
    Serial.print(" cm | ");
    // --- 거리값에 따라 # 개수로 시각화 (1cm당 1개, 최대 50개) ---
    int barLength = map(distance, 0, 50, 0, 50);
    for (int i = 0; i < barLength; i++) Serial.print('#');
    Serial.println();
    // --- 부저 경고 (10cm 이하일 때 짧게 뻤) ---
    if (distance < 10) tone(BUZ, 1200, 200);
    else noTone(BUZ);
  }
}
```

4. 관찰- millis() 타이머를 이용하여 1초마다 자동으로 거리(cm)를 측정하고, 그 결과를 시리얼 모니터에 수치 + 막대 그래프(###) 형태로 거리 기록형 초음파 시스템
  - 프로그램이 실행되면 millis()로 경과 시간을 계속 측정
  - 1초가 지날 때마다 초음파 센서로 거리를 측정
  - 측정된 거리값을 숫자와 함께 # 문자의 개수로 시각화해 시리얼에 표시
  - 예를 들어 30cm이면 ##### 와 같이 표시
  - 거리가 10cm 이하일 때는 부저가 짧게 뻤- 소리를 냄

## 프로젝트 E: 시리얼 명령 제어형 초음파 감지기

1. 준비  
-사용 부품: 초음파센서 1개, 부저 1개, 점퍼선, ESP32
2. 배선  
-- 초음파센서(TRIG): 12번, 초음파센서(ECHO): 14번, 부저: 4번
3. 업로드  
- 코드 스켈레톤

```
#include <Arduino.h>
const int TRIG = 12; // 초음파 센서의 Trig 핀
const int ECHO = 14; // 초음파 센서의 Echo 핀
const int BUZ = 2; // 부저 핀
bool active = false; // on/off 상태 저장 변수
void setup() {
  Serial.begin(115200); // 시리얼 모니터 시작 (통신속도 115200bps)
  pinMode(TRIG, OUTPUT); // Trig 핀 출력 설정
  pinMode(ECHO, INPUT); // Echo 핀 입력 설정
  pinMode(BUZ, OUTPUT); // 부저 핀 출력 설정
}
void loop() {
  // --- 시리얼 명령 읽기 ---
  if (Serial.available()) {
    String cmd = Serial.readStringUntil('\n');
    cmd.trim(); // 개행 제거
    if (cmd == "on") {
      active = true; Serial.println("▶ Measuring Started");
    }
    else if (cmd == "off") {
      active = false; noTone(BUZ); Serial.println("■ Measuring Stopped");
    }
  }
  if (active) { // --- on 상태일 때만 측정 ---
    digitalWrite(TRIG, LOW); delayMicroseconds(2);
    digitalWrite(TRIG, HIGH); delayMicroseconds(10);
    digitalWrite(TRIG, LOW);
    float distance = pulseIn(ECHO, HIGH) * 0.034 / 2;
    Serial.print("Distance: "); Serial.print(distance); Serial.println(" cm");
    if (distance < 10) tone(BUZ, 1000); else noTone(BUZ);
    delay(300);
  }
}
```

4. 관찰 - 시리얼 통신(Serial Monitor) 을 이용해 사용자가 직접 “on/off” 명령을 입력하여 초음파 센서의 거리 측정을 시작하거나 중지할 수 있는 제어형 시스템  
- 시리얼 모니터에서 on 입력 → 초음파 거리 측정 시작  
- 시리얼 모니터에서 off 입력 → 측정 중단 + 부저 끄  
- on 상태에서는 초음파 센서가 지속적으로 거리를 계산하고 시리얼에 출력  
- 거리가 10cm 이하로 접근하면 부저가 뽐— 소리를 냄  
- 측정이 중단(off)되면 모든 동작이 멈춤