
초음파 센서

초음파 센서

❖ 초음파 센서

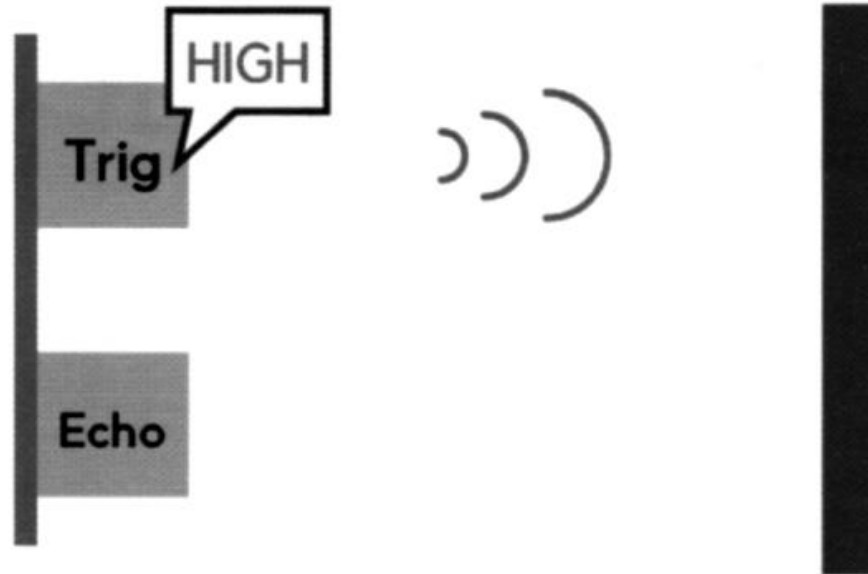
- 초음파를 이용하여 벽이나 사물의 거리를 측정하는 센서
- HC-SR04 모듈
 - Vcc: 전원
 - Trig : 트리거 (초음파 송출 제어)
 - Echo : 에코(초음파 수신)
 - GND: 접지
 - 최대거리: 4~5m
 - 인식각도: 15도



초음파센서 사용하기

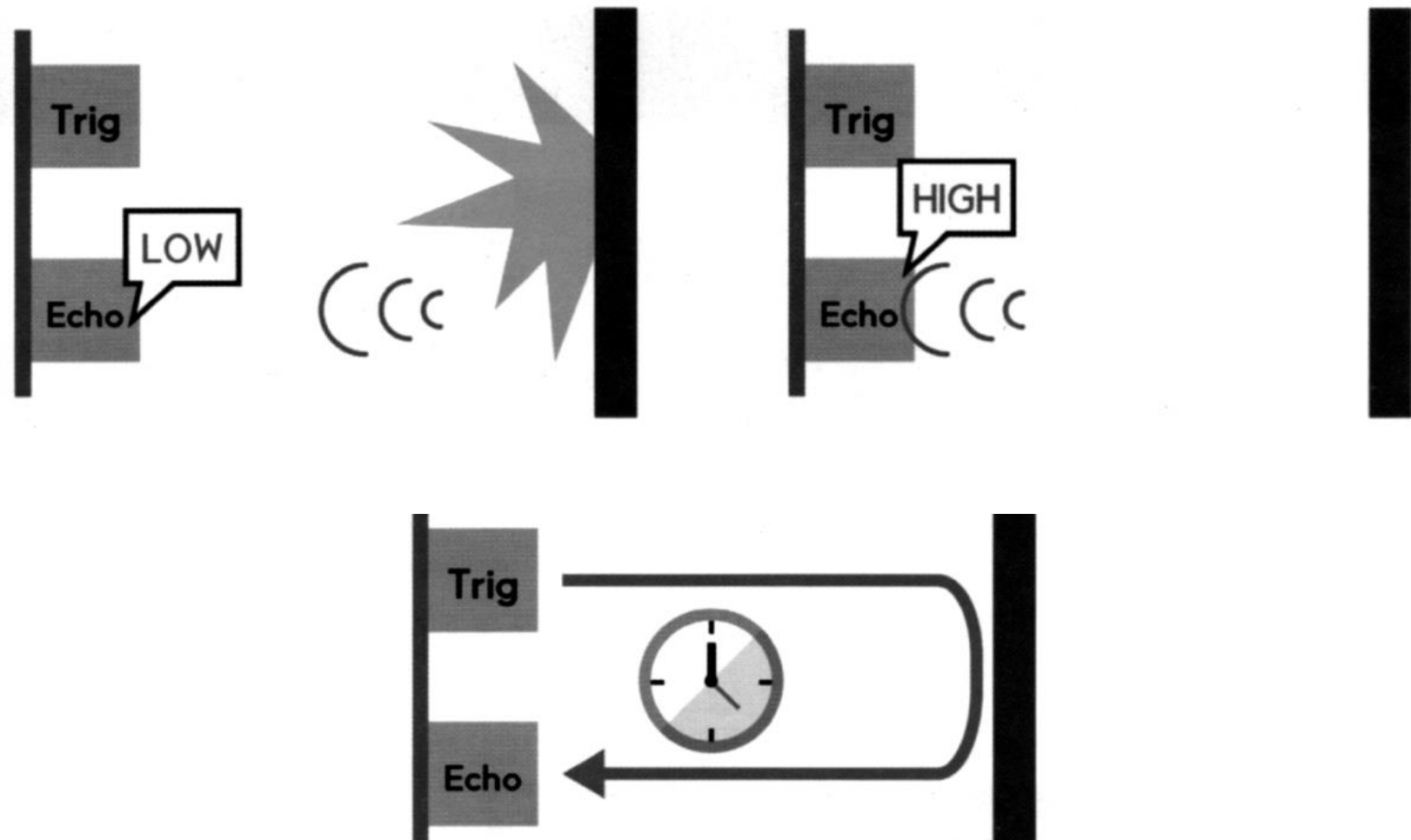
❖ 초음파 센서 동작 원리

- Trig에 HIGH 전압 인가시 초음파 발사
- Echo에서 초음파 수신(처음에는 LOW)
- 시간 차이를 측정하여 거리 계산



초음파센서

❖ 초음파 센서 동작 원리



초음파센서

❖ 거리 계산

- 음속 : 340 m/s
- 비례식
 - $1,000,000 \text{ us} : 34,000 \text{ cm} = x \text{ us} : y \text{ cm}$
 - $y = 34,000 * x / 1,000,000$
 $= 0.034 * x \text{ cm}$
- 왕복 반영
 - $y = 0.034 * x / 2 = 0.017 * x \text{ cm}$

초음파센서

❖ delayMicroseconds()

- 아두이노 보드를 일정 시간 멈추는 함수입니다.
- delayMicroseconds(멈출 시간)
- 매개변수
 - 멈출 시간: 단위는 마이크로초로 (0.000001초)
- 반환 값
 - 없음.
- 사용 예
 - `delayMicroseconds(2);` // 아두이노 보드를 0.002 밀리초(0.000002초) 멈춥니다.

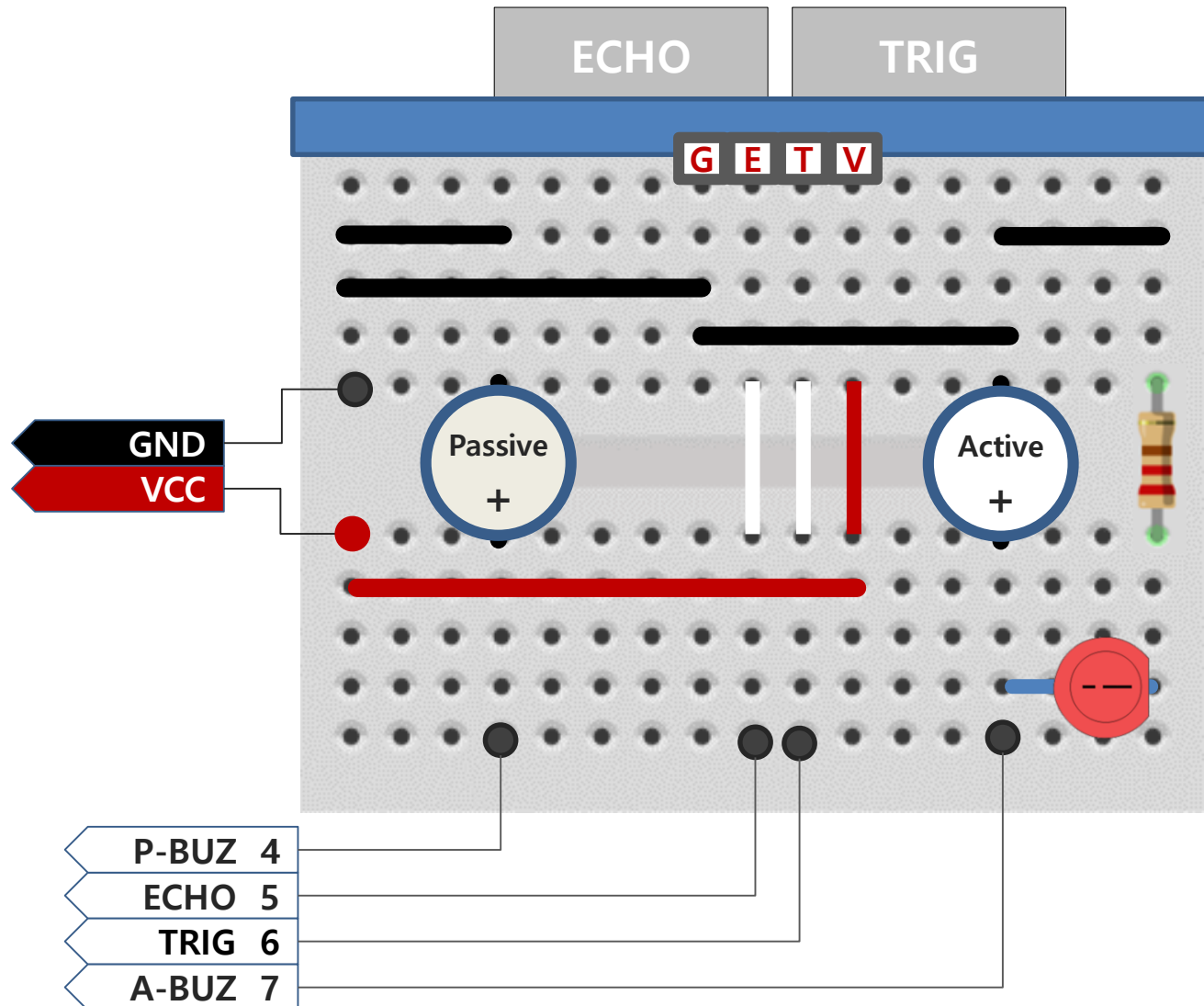
초음파센서

❖ pulseIn()

- 디지털 입력 핀의 전압이 LOW 또는 HIGH가 될 때까지 걸린 시간 측정
- 핀은 꼭 pinMode 함수를 이용해 입력 모드로 설정
- pulseIn(핀 번호, 전압)
 - 매개변수
 - 핀 번호: 전압이 바뀌는 시간을 재고자 하는 핀 번호
 - 전압: 바뀌는 전압. HIGH - LOW에서 HIGH가 될 때
- 반환값
 - 전압이 바뀌는 데 걸린 시간을 마이크로초 기준으로 반환
 - 만약 1초 동안 기다렸는데도 전압이 바뀌지 않으면 0을 반환
- 사용 예
 - `long duration = pulseIn(3, HIGH);`
//3번 핀의 전압이 HIGH가 될 때까지 걸린 시간

초음파센서

❖ 회로도



초음파 센서

❖ ex01/app.ino 초음파 센서 거리 측정

```
#include <MiniCom.h>

MiniCom com;

int echoPin = 5;
int triggerPin = 6;

void check() {
    // trigger 핀으로 10us의 펄스를 발생
    digitalWrite(triggerPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(triggerPin, LOW);

    // echo 핀 입력으로부터 거리르 cm 단위로 계산
    int distance = pulseIn(echoPin, HIGH) / 58;
    com.print(1, "Dist.(cm)=", distance);
}
```

초음파 센서

❖ ex01/app.ino 초음파 센서 거리 측정

```
void setup() {  
    com.init();  
    com.setInterval(1000, check);  
    com.print(0, "[[Distance]]");  
    pinMode(echoPin, INPUT);  
    pinMode(triggerPin, OUTPUT);  
}
```

```
void loop() {  
    com.run();  
}
```

Ultra 클래스

❖ Ultra.h

```
#pragma once

#include <Arduino.h>

class Ultra {
protected:
    int echo;
    int trig;

public:
    Ultra(int echo, int trig);
    int getDistance();
};
```

Ultra 클래스

❖ Ultra.cpp

```
#include "Ultra.h"

Ultra::Ultra(int echo, int trig) : echo(echo), trig(trig) {
    pinMode(echo, INPUT);
    pinMode(trig, OUTPUT);
}

int Ultra::getDistance() {
    digitalWrite(trig, HIGH);
    delayMicroseconds(10);
    digitalWrite(trig, LOW);

    // echo 핀 입력으로부터 거리르 cm 단위로 계산
    //  $y = 0.034 * x / 2 = 0.017 * x$  cm --> 실수 연산
    int distance = pulseIn(echo, HIGH) / 58;
    return distance;
}
```

초음파 센서

❖ ex02/app.ino

```
#include <MiniCom.h>
#include <Ultra.h>

MiniCom com;
Ultra ultra(5, 6);

void check() {
    int distance = ultra.getDistance();
    com.print(1, "Dist.(cm)=", distance);
}

void setup() {
    com.init();
    com.setInterval(1000, check);
    com.print(0, "[[Distance]]");
}

void loop() {
    com.run();
}
```

초음파 센서

❖ Buzzer.h 자동차 후방 감지기 만들기

```
#pragma once
#include <Arduino.h>

class Buzzer {
protected:
    int pin;          // 핀번호
    int state;        // 소리 출력 상태
    int on_time;      // 소리를 내야하는 시간 간격
    int off_time;     // 꺼져야 할 시간 간격
    unsigned long old_time; // 이전 상태 변화 시간
    bool bstop;

public:
    Buzzer(int pin);
    void beep();
    void stop() {
        bstop = true;
        digitalWrite(pin, LOW);
    }
    void setOffTime(int off_time);
    void run();
};
```

초음파 센서

❖ Buzzer.cpp 자동차 후방 감지기 만들기

```
#include "Buzzer.h"

Buzzer::Buzzer(int pin): pin(pin) {
    state = LOW;
    off_time = 0;
    on_time = 100; // 100ms
    old_time = millis();
    pinMode(pin, OUTPUT);
    bstop = true;
}

void Buzzer::beep() {
    digitalWrite(pin, HIGH);
    delay(50);
    digitalWrite(pin, LOW);
}

void Buzzer::setOffTime(int off_time) {
    this->off_time = off_time;
    bstop = false;
}
```

초음파 센서

❖ Buzzer.cpp 자동차 후방 감지기 만들기

```
void Buzzer::run() {
    unsigned long cur, diff;

    if(bstop) return;

    cur = millis();          // 현재 시간
    diff = cur - old_time;    // 이전 상태변화 시간에서의 시간차

    if(state) { // 소리를 출력하고 있는 상태
        if(diff > on_time) { // 소리 출력 시간이 끝났음
            digitalWrite(pin, LOW); // 소리 끄기
            state = LOW;            // 꺼짐 상태
            old_time = cur;         // 상태 변이 시간 갱신
        }
    } else { // 소리가 꺼져 있는 상태
        if(diff > off_time ) { // 소리 꺼진 시간이 끝났음
            digitalWrite(pin, HIGH); // 소리 켜기
            state = HIGH;            // 켜짐 상태
            old_time = cur;         // 상태 변이 시간 갱신
        }
    }
}
```


초음파 센서

❖ ex03/app.ino 자동차 후방 감지기 만들기

```
#include <MiniCom.h>
#include <Ultra.h>
#include <Buzzer.h>

MiniCom com;
Ultra ultra(5, 6);
Buzzer buzzer(7);

void check() {
    int distance = ultra.getDistance();
    com.print(1, "Dist.(cm)=", distance);

    if(distance < 50 && distance >= 30) {
        buzzer.setOffTime(1000);
    } else if(distance < 30 && distance >= 20) {
        buzzer.setOffTime(500);
    } else if(distance < 20 && distance >= 10) {
        buzzer.setOffTime(200);
    } else if(distance < 10) {
        buzzer.setOffTime(100);
    } else {
        buzzer.stop();
    }
}
```

초음파 센서

❖ ex03/app.ino 자동차 후방 감지기 만들기

```
void setup() {  
    com.init();  
    com.setInterval(100, check);  
    com.print(0, "[[Distance]]");  
    buzzer.stop();  
}  
  
void loop() {  
    com.run();  
    buzzer.run();  
}
```

초음파 센서

❖ 물체 감지기

- 물체가 지정 범위 내로 진입하면 지정한 함수 호출
- 초음파 센서가 일종의 스위치 역할을 함
 - 범위 내 진입 : 스위치 on
 - 범위 밖으로 나감: 스위치 off

초음파 센서

❖ Ultra.h 수정

```
#pragma once

#include <Arduino.h>

typedef void (*ultra_callback_t)();

class Ultra {
protected:
    int echo;
    int trig;
    int threshold;           // 기준 거리
    ultra_callback_t on_f;   // 기준 거리 내로 진입한 경우 호출 할 함수
    ultra_callback_t off_f;  // 기준 거리를 벗어난 경우 호출 할 함수
    int state;               // 벗어난 경우 false, 진입한 경우 true

public:
    Ultra(int echo, int trig);
    int getDistance();
    void setThreshold(int t, ultra_callback_t on_f, ultra_callback_t off_f);
    int run();
};
```

초음파 센서

❖ Ultra.cpp 수정

```
#include "Ultra.h"

Ultra::Ultra(int echo, int trig) : echo(echo), trig(trig) {
    pinMode(echo, INPUT);
    pinMode(trig, OUTPUT);
    on_f = NULL;
    off_f = NULL;
    state = false;
}

:

void Ultra::setThreshold(int t, ultra_callback_t on_f, ultra_callback_t off_f)
{
    threshold = t;
    this->on_f = on_f;
    this->off_f = off_f;
}
```

초음파 센서

❖ Ultra.cpp 수정

```
int Ultra::run() {
    int distance = getDistance();
    if(!state && distance <= threshold) {        // 기준 거리내로 진입한 경우
        state = true;
        if(on_f != NULL) {
            on_f();
        }
        // 기준 거리를 벗어난 경우
    } else if(state && (distance == 0 || distance > (threshold+2))){
        state = false;
        if(off_f != NULL) {
            off_f();
        }
    }

    if(distance == 0 || distance > threshold ) {
        return 0;
    }

    return distance;
}
```

초음파 센서

❖ ex04/app.ino] 물체 감지 경보기

```
#include <MiniCom.h>
#include <Ultra.h>

MiniCom com;
Ultra ultra(5, 6);

void notice_in() {
    com.print(1, "NOTICE IN");
}

void notice_out() {
    com.print(1, "NOTICE OUT");
}

void setup() {
    com.init();
    com.print(0, "[[Distance]]");
    ultra.setThreshold(40, notice_in, notice_out);
}

void loop() {
    com.run();
    ultra.run();
}
```

