**✅ 8.1주차: 센서 사용법 기초**

**1. 버저 (Buzzer)**

* 디지털 출력으로 단순 소리 생성
* **tone() 함수** 사용 → 주파수 기반 소리 출력

**🎯 예제 코드**

cpp

복사편집

#include "pitches.h"

int melody[] = {

NOTE\_C4, NOTE\_D4, NOTE\_E4, NOTE\_F4,

NOTE\_G4, NOTE\_A4, NOTE\_B4, NOTE\_C5

};

int durations[] = {8, 8, 6, 6, 4, 4, 4, 4};

void setup() {

for (int i = 0; i < 8; i++) {

int duration = 1000 / durations[i];

tone(2, melody[i], duration);

delay(duration + 30);

}

}

void loop() {}

**2. 모션 감지 센서 (PIR)**

* 움직임 감지 시 HIGH 출력
* 디지털 입력 사용

**🎯 예제 코드**

cpp

복사편집

int ledPin = 13;

int pirPin = 2;

void setup() {

pinMode(ledPin, OUTPUT);

pinMode(pirPin, INPUT);

}

void loop() {

int motion = digitalRead(pirPin);

digitalWrite(ledPin, motion);

delay(10);

}

**3. 초음파 센서 (HC-SR04)**

* 거리 측정: Trig → 초음파 발신 / Echo → 수신
* 거리(cm) = 시간 / 29 / 2

**🎯 예제 코드**

cpp

복사편집

void setup() {

Serial.begin(9600);

pinMode(2, OUTPUT); // Trig

pinMode(3, INPUT); // Echo

}

void loop() {

digitalWrite(2, HIGH); delayMicroseconds(10); digitalWrite(2, LOW);

int duration = pulseIn(3, HIGH);

int cm = duration / 29 / 2;

Serial.print(cm); Serial.println("cm");

delay(300);

}

**✅ 9주차: 아두이노 통신 기초**

**● 통신 개념**

* **Parallel vs Serial 통신**
  + Parallel: 여러 라인을 동시에 사용 (고속, 복잡)
  + Serial: 한 라인에 순차적으로 전송 (저속, 간단)
* **동기식(Synchronous) vs 비동기식(Asynchronous)**
  + 동기식: Clock 사용 (I2C, SPI 등)
  + 비동기식: Clock 없이 데이터 전송 (UART/Serial 통신)

**● 아두이노 기본 통신 방식**

* 디지털 핀 사용한 1:1 통신 가능
* 모르스 기호 같은 간단한 시리얼 구조

**● 온습도 센서 (DHT11)**

* 핀 구성: VCC, DATA, NC, GND
* 온도: 090% RH ±5%
* 아두이노와 디지털 통신으로 주기적 전송
* **🎯 예제 코드**
* cpp
* 복사편집
* #include <DHT11.h>
* DHT11 dht(2);
* void setup() {
* Serial.begin(9600);
* }
* void loop() {
* float temp, humi;
* if (dht.read(humi, temp) == 0) {
* Serial.print("Temp: "); Serial.print(temp);
* Serial.print(" Humi: "); Serial.println(humi);
* } else {
* Serial.println("Sensor error");
* }
* delay(1000);
* }

**✅ 10주차: Serial(UART) 통신 상세**

**● Hardware Serial (UART)**

* 비동기식 직렬 통신
* **RX↔TX 크로스 연결** 필요
* 클럭 불일치 문제를 해결하기 위해 **Start/Stop 비트** 사용
* 예: 9600 8N1 = 9600bps, 8bit, No parity, 1 stop bit

**● Software Serial**

* 기본 RX/TX 외 디지털 핀으로 시리얼 통신 구현
* **주의사항**:
  + 일부 라이브러리와 충돌 가능성 있음
  + RAM 사용량: 약 150~200byte
* 일반적으로 외부 장치와의 통신에 사용 (블루투스 등)

**● 시리얼 테스트 예제**

* 시리얼 통신으로 PC에서 LED 제어
* Serial.available() → 입력 감지 후 LED on/off

**🎯 시리얼 LED 제어 예제**

cpp

복사편집

void setup() {

Serial.begin(9600);

pinMode(4, OUTPUT);

}

void loop() {

if (Serial.available()) {

char c = Serial.read();

if (c == '1') {

digitalWrite(4, !digitalRead(4));

Serial.println("LED TOGGLED");

}

}

}

**✅ 11주차: 통신 모듈 응용**

**● 유선 통신**

* **USB**, **Ethernet**
  + 안정적이며 구성 간단
  + Ethernet은 TCP/IP, HTTP 서버 구현 가능

**● 무선 통신 모듈**

1. **적외선(IR)**
   * 간단한 회로, 근거리 전용, 장애물에 취약
2. **RF (ex. HC-11)**
   * 200MHz~2.4GHz, N:N 가능, 간단한 구조
3. **ZigBee**
   * 고속 Mesh 네트워크 구성 가능, 난이도 높음
4. **LoRa**
   * 장거리(10~20km), 저전력 통신 가능
5. **Bluetooth (HC-06)**
   * Classic BT (2.0), 모바일 연동 최적
   * SoftwareSerial 사용, AT 명령어로 설정 가능
6. **Wi-Fi (ESP8266 등)**
   * AP/Client 모두 가능, 인터넷 연결 지원

**● 블루투스 채팅 예제**

* 아두이노 ↔ 모바일 간 실시간 채팅
* PC Serial ↔ 아두이노 ↔ Bluetooth ↔ 안드로이드

**● RF 채팅 예제**

* HC-11 이용한 다중 사용자 채팅
* Echo 예제 활용, PC 간 메시지 전송
* **● 블루투스 예제: Echo 채팅**
* cpp
* 복사편집
* #include <SoftwareSerial.h>
* SoftwareSerial BTSerial(2, 3);
* void setup() {
* Serial.begin(9600);
* BTSerial.begin(9600);
* }
* void loop() {
* if (BTSerial.available())
* Serial.write(BTSerial.read());
* if (Serial.available())
* BTSerial.write(Serial.read());
* }