

# 안드로이드\_온습도 센서



# 온/습도 센서

- 온도와 습도 등을 측정하는 센서

- 온/습도 센서에서 받아들이는 정보는 하드웨어를 통하여 디지털화된 결과를 도출

- SHT11(온/습도) 센서

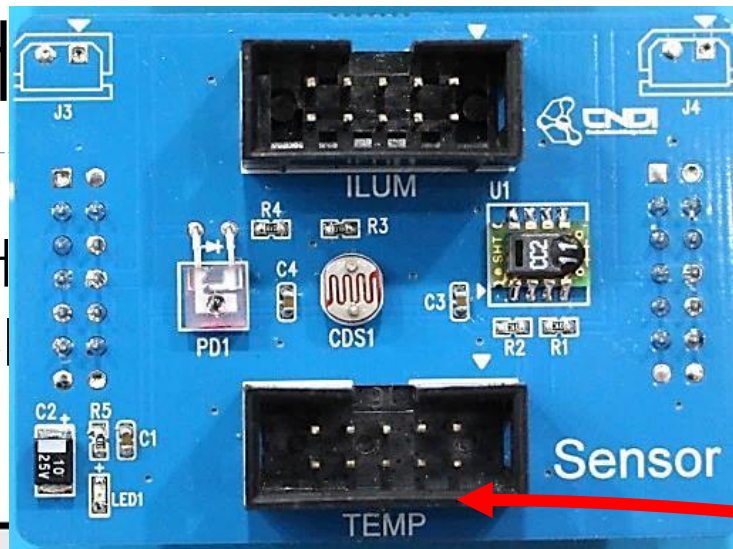
- SENSIRION사의 SHT1x 시리즈의 SHT11 Humidity & Temperature Sensor를 사용

- 디지털 출력 온/습도 센서인 SHT11은 하나의 칩에 두 개의 센서를 집적화 한 제품으로 특징은 다음과 같음

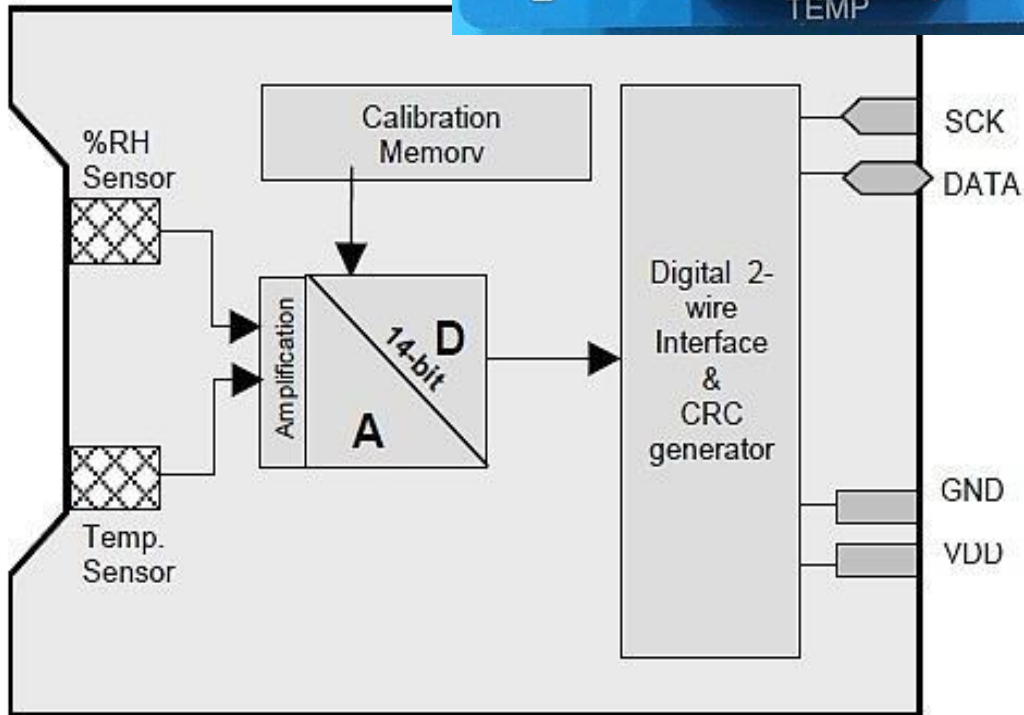
- 측정 범위 : 0~100% RH
    - 상대 습도 정밀도 +/- 3%
    - 온도 정밀도: +/- 0.4°C @ 25 °C
    - 보정된 디지털 출력 (2 와이어 인터페이스)
    - 빠른 반응 속도 < 4 sec.
    - 저전력 소비 (typ. 30 µW)
    - 낮은 가격
    - 대량 생산용으로 디자인 됨, 가격이 민감한 응용 제품용
    - 뛰어난 장기 안정성
    - 보정과 디지털 2 와이어 인터페이스 기능으로 사용이 쉬움

# 온/습도 센서

- 다음 블록 다이어그램에 표시된 온/습도 센서라고 표시

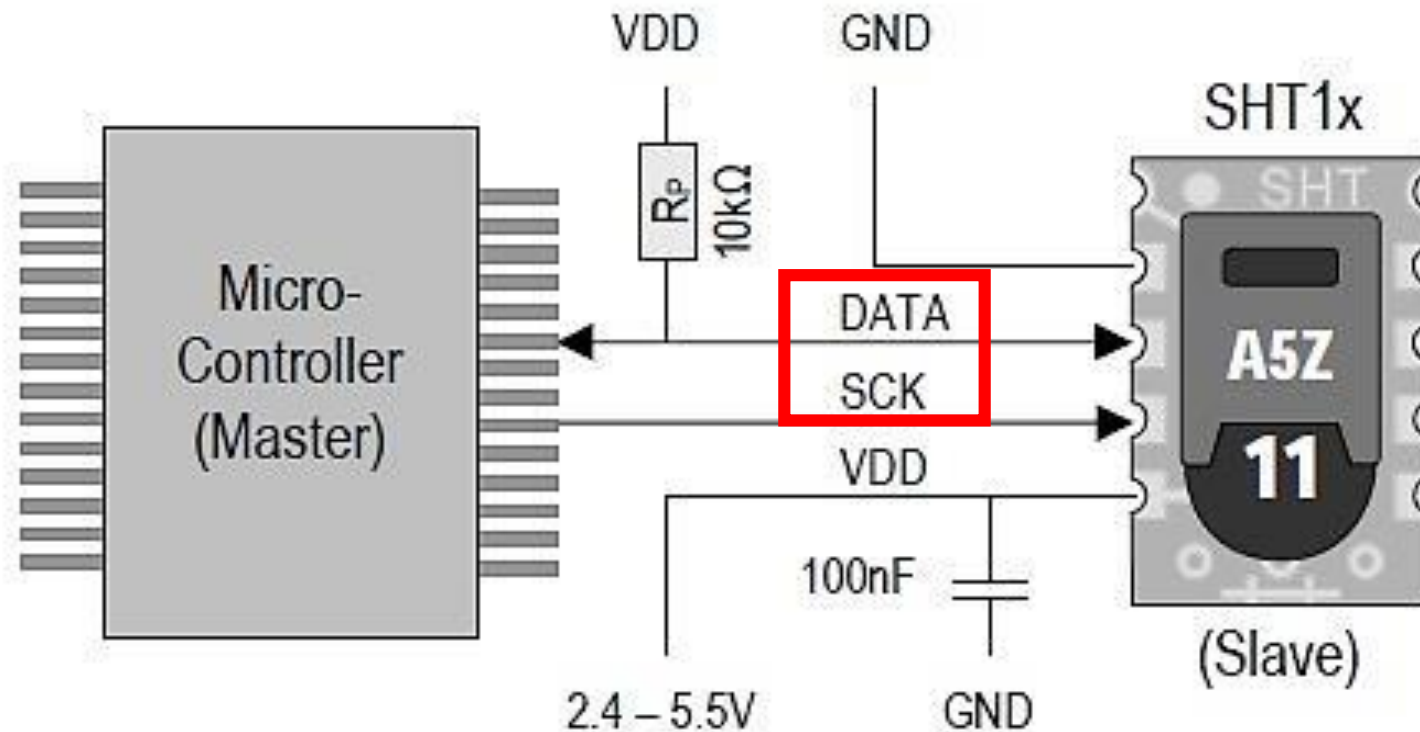


되어 있는 부분과 Temp  
습도의 정도를 센싱(Sensing)



# 온/습도 센서(내부의 A/D 변환기)와 라즈베리 파이의 통신

- 센싱된 온습도의 아날로그 수치는 14비트 A/D 변환기를 통해 디지털로 수치화하고 I2C를 통하여 정보를 처리할 MCU(라즈베리파이)로 전달
- 일반적인 회로구성 예



# 온/습도 센서 동작 방식

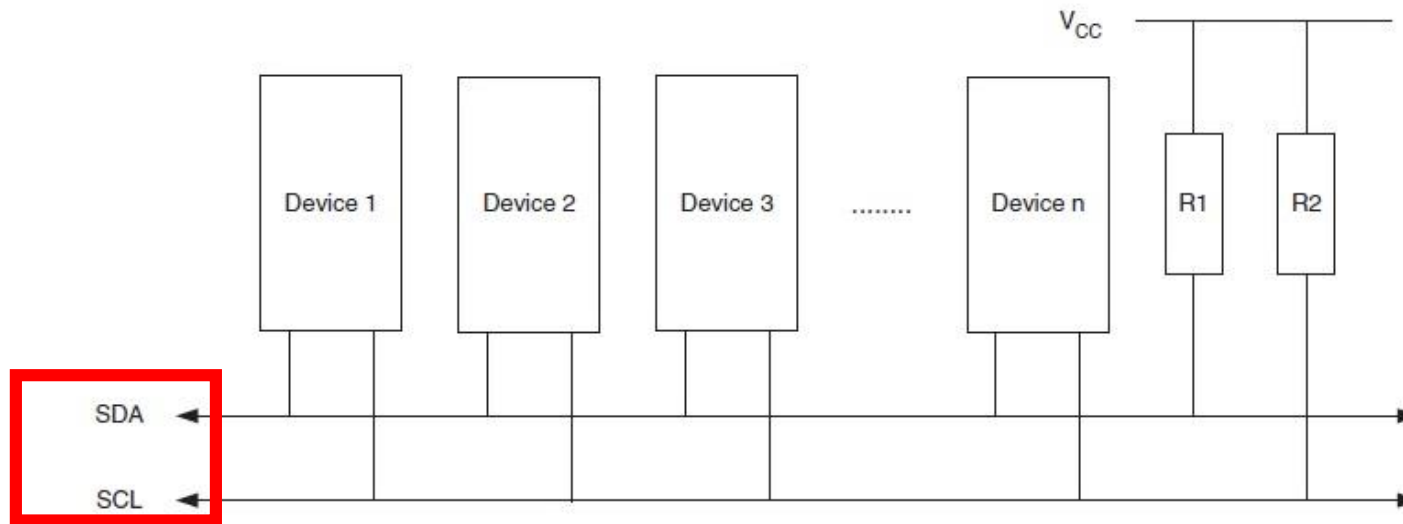
---

- 센서가 전원을 입력 받으면 온/습도를 센싱하는 하드웨어를 통해 아날로그 정보를 측정
- 측정된 정보는 센서 자체가 가진 A/D 변환 기능을 통해 디지털 수치로 변환
- 변환된 수치는 직렬 버스를 통해 CPU로 전달

## 2-Wire 직렬 인터페이스(Two-wire Serial Interface, TWI)

- TWI 프로토콜은 단지 2개의 양방향 버스 라인, 클럭(SCL)을 위한 하나의 버스와 데이터(SDA)를 위한 또 하나의 버스를 이용한 128개 이상 다른 장치의 상호연결을 위한 시스템 디자인을 허용

- 디바이스를 구분하기 위해 주소 개념 사용



# TWI(I2C) 특징

---

- 단순하지만 파워풀하고 유연한 통신 인터페이스, 단지 2개 버스 라인 필요
- 마스터와 슬레이브 동작을 지원
- 장치는 전송자와 수신자로서 동작할 수 있음
- 클럭 신호는 단방향이며, 마스터에 의해 발생되고 데이터 신호는 양방향으로 데이터를 송수신하기 위해 사용

# 라즈베리파이\_동작 센서(temp.c) sensor.h /sensor.c

```
1. #include "sensor.h"
2. volatile float temp;
3. volatile float humi;
4. int main(void)
5. {
6.     if(wiringPiSetupGpio() == -1)
7.         return 1;
8.     SHT11_Init();
9.     while(1)
10.    {
11.        Transmission_start();
12.        temp = get_SHT11_data(TEMP);
13.        delay(100);
14.        Transmission_start();
15.        humi = get_SHT11_data(HUMI);
16.        delay(100);
17.        printf("Temp = %5.2f[C], Humi = %5.2f[%]Wn", temp, humi);
18.    }
19.    return 0;
20.}
```

## • 수행 tip

1. 온습도센서 모듈, FND모듈을 라즈베리파이의 GPIO와 연결
2. C언어로 소스코드 작성
3. 소스코드 컴파일하기  
`gcc -c sensor.c`  
`gcc -o 실행파일이름 파일이름.c 센서.o -lwiringPi`
4. 코드 실행하기  
`sudo ./파일이름`



# Fan 개념 및 제어방법

- 전력(전기적 에너지)을 회전운동의 힘(기계적 에너지)으로 바꾸어 바람, 즉 공기의 흐름을 만드는 기계로써, **전력을 공급하면 회전 날개를 회전**
  - 선풍기에 FAN이 포함되고, 컴퓨터도 칩의 열을 식혀주기 위해 작은 FAN이 들어있으며, 내부의 공기를 환기하는 목적의 FAN도 있음
- FAN의 동작은 ON/OFF 두 가지
- **Fan 모듈의 FAN 포트 각 핀에 입력되는 신호에 따른 동작**

동작	MT_P 핀	MT_N 핀
ON(회전)	High	Low
OFF(정지)	Low	Low

# 라즈베리파이\_FAN

```
1.  #include <stdio.h>
2.  #include <wiringPi.h>
3.  #define P_PIN  6
4.  #define N_PIN  12

5.  void FanOn(void){ // Fan 회전
6.      digitalWrite(P_PIN, HIGH);
7.      digitalWrite(N_PIN, LOW);
8.  }

9.  void FanOff(void){ // Fan 회전 중지
10.     digitalWrite(P_PIN, LOW);
11.     digitalWrite(N_PIN, LOW);
12. }

13. int main(void){

14.     if(wiringPiSetupGpio() == -1)
15.         return 1;

16.     pinMode(P_PIN, OUTPUT);
17.     pinMode(N_PIN, OUTPUT);

18.     while(1){
19.         FanOn();
20.         delay(2000);
21.         FanOff();
22.         delay(2000);
23.     }
24.     return 0;
25. }
```

# 동작 센서\_클라이언트 (Temp\_client)

