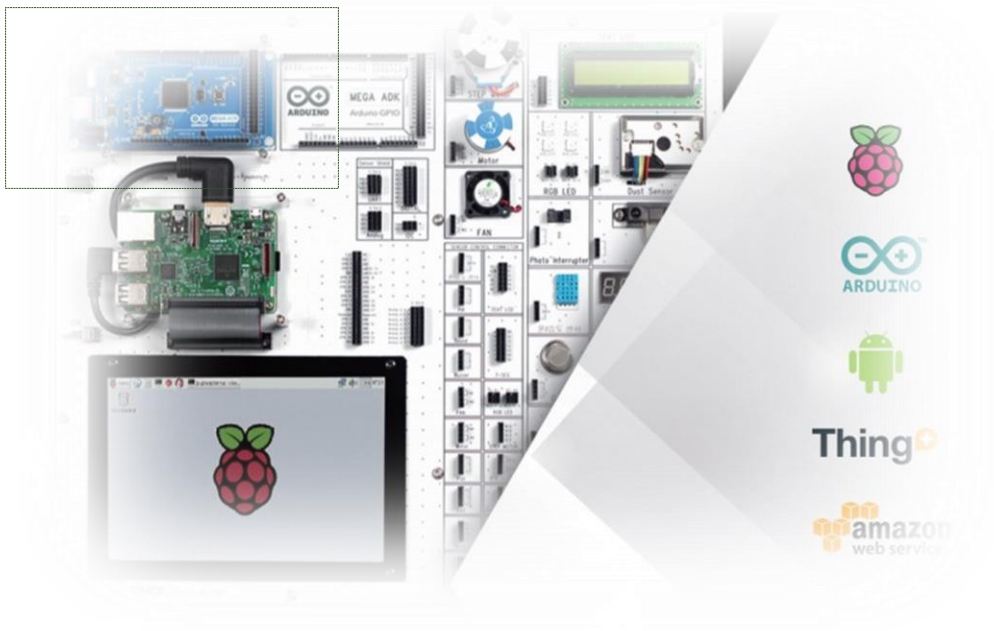
마이크로프로세서
(강의자료 #10)



교과목명 : 마이크로프로세서 (01)

담당교수 : 이 수 형

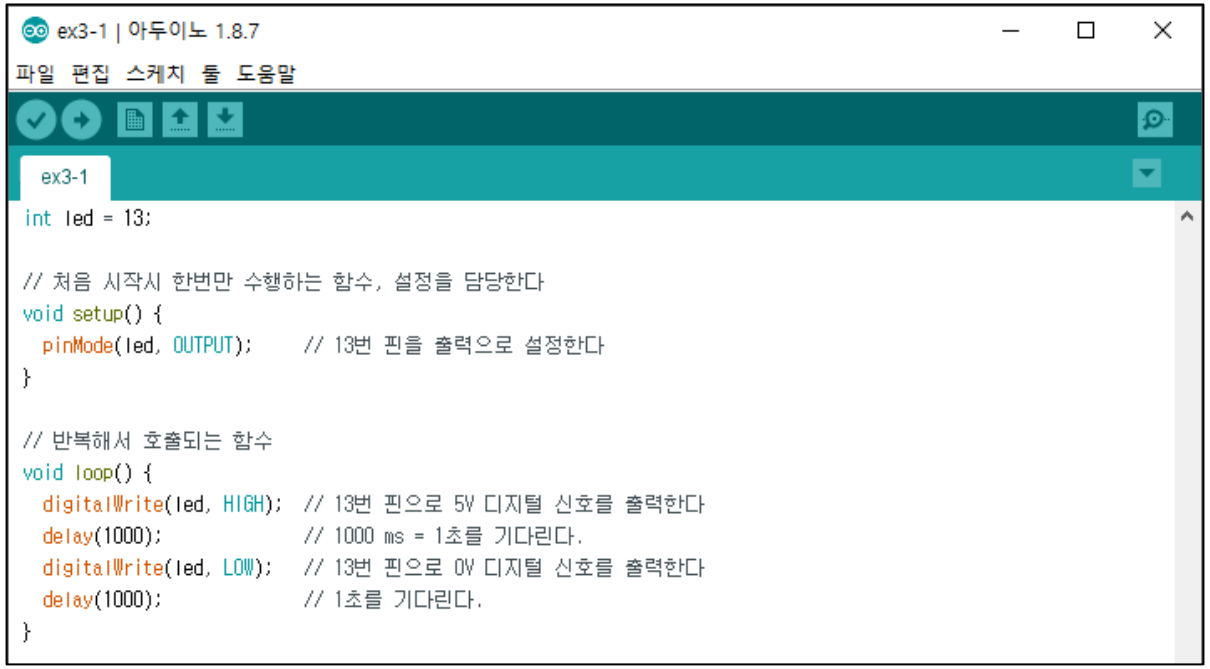
E-mail : soohyong@uu.ac.kr

교재명 : 스마트기기 개발을 위한 아두이노, 이수형. (LINC+ 사업단 배포)

수강생 안내

- 본 교과목은 기본적으로 “대면”수업입니다.
- ‘코로나바이러스감염증-19’ 등의 이유로 대면수업을 받지 못하는 학생들을 위해서 ‘비대면 실시간 수업’을 병행합니다. 기본적으로 대면 수업이므로 대면 수업에 참석한 학생들은 교실에서 출석을 체크하면 되며, 참석하지 못하고 ‘비대면 수업’을 듣는 학생들은 ‘실시간 온라인 강의’에 참석하면 출석이 반영됩니다.
- 수강생 여러분들은 강의 자료를 온라인 배포 및 게재 등의 행위를 할 시 저작권 문제가 발생할 수 있사오니 이에 유념하여 강의 자료를 공유하는 행위는 삼가 바랍니다.
- 교재는 “스마트기기 개발을 위한 아두이노”이며, 나누어준 키트는 각자 잘 관리하면서 실습실(대면) 및 집(비대면)에서 실험/실습을 수행할 수 있도록 진행합니다.

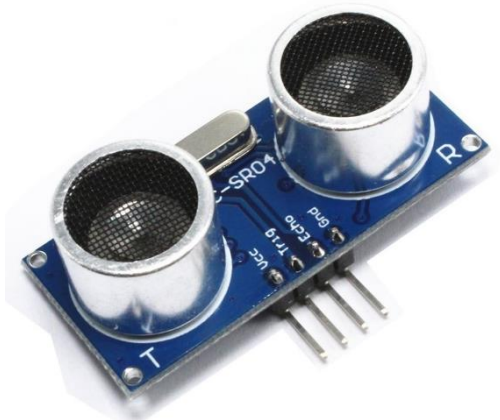
8. 센서 활용


ex3-1 | 아두이노 1.8.7
파일 편집 스케치 툴 도움말
ex3-1
int led = 13;
// 처음 시작시 한번만 수행하는 함수, 설정을 담당한다
void setup() {
pinMode(led, OUTPUT); // 13번 핀을 출력으로 설정한다
}
// 반복해서 호출되는 함수
void loop() {
digitalWrite(led, HIGH); // 13번 핀으로 5V 디지털 신호를 출력한다
delay(1000); // 1000 ms = 1초를 기다린다.
digitalWrite(led, LOW); // 13번 핀으로 0V 디지털 신호를 출력한다
delay(1000); // 1초를 기다린다.
}

8.1 초음파 센서

- 거리 측정 센서
  - 초음파, 레이저, 적외선 센서를 사용
  - 초음파 : 상대적으로 저렴하게 제작 가능
- 초음파 센서
  - 40kHz의 음파를 발산하여 되돌아오는 신호를 탐지하여 음파의 전달 시간을 측정하여 거리를 계산

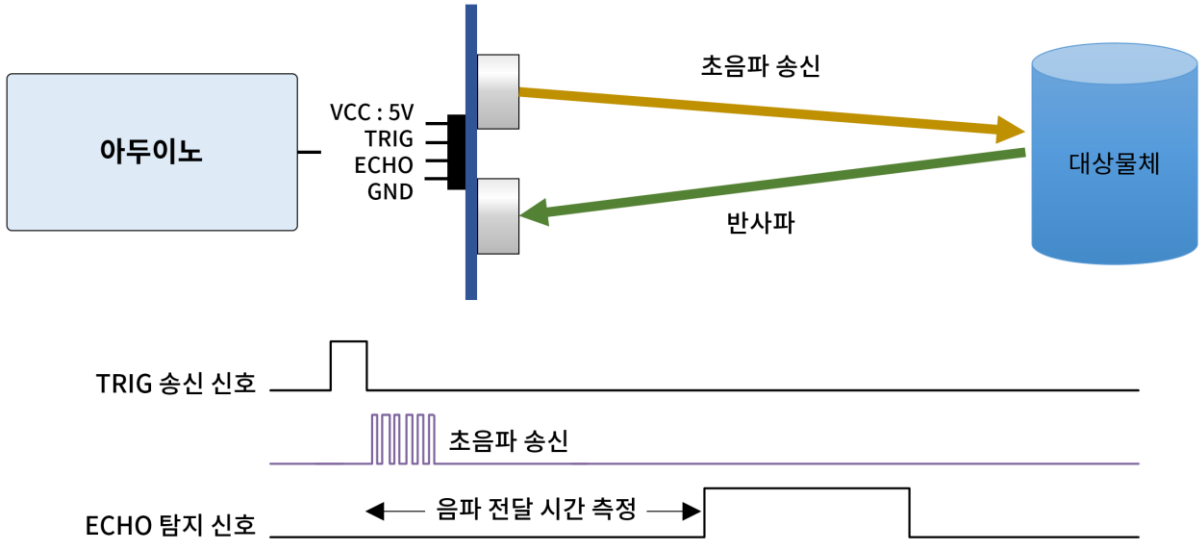
Parallax.com 사의 초음파 센서



HC-SR04 초음파 센서

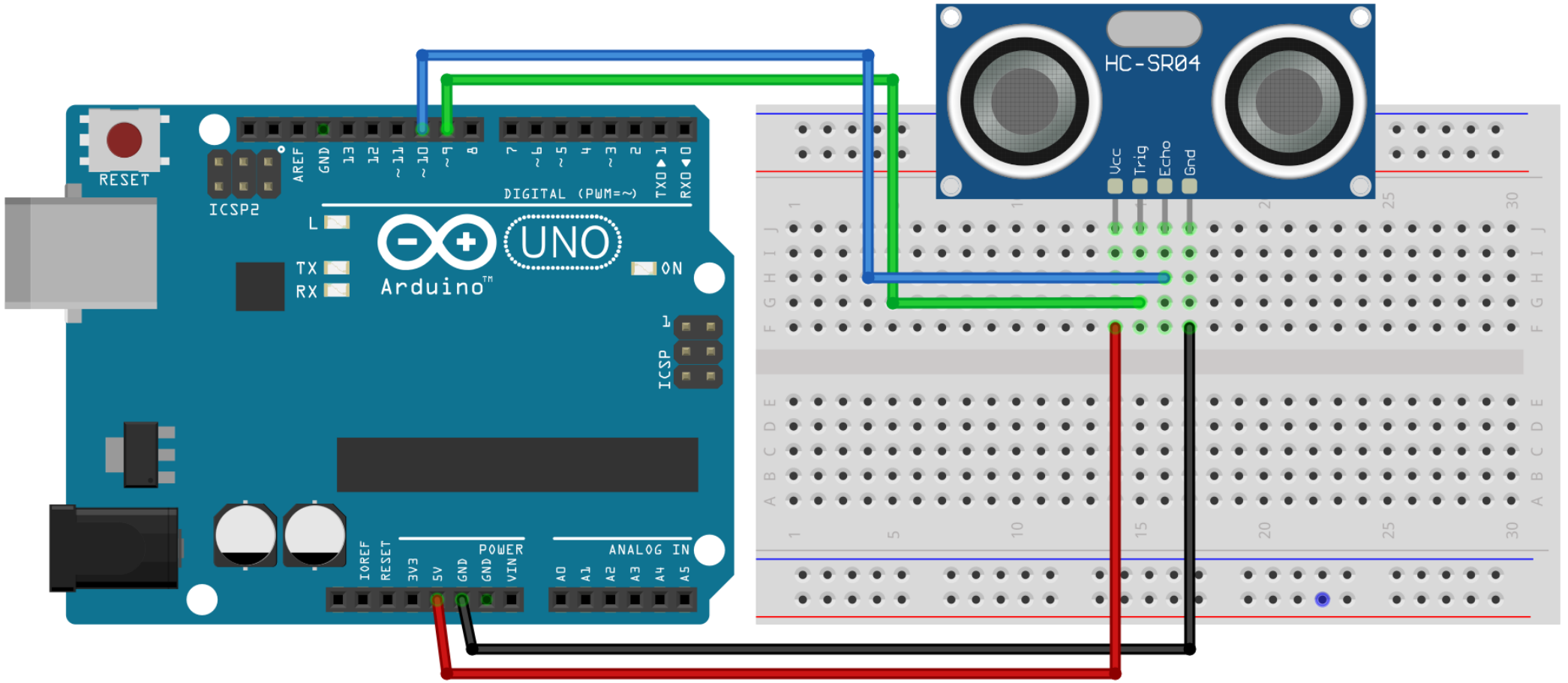
- 초음파 센서의 거리 측정
  - TRIG : 10us의 펄스 송신 → 물체에 반사되어 되돌아옴
  - ECHO : 반사파가 도달하면 HIGH 신호가 입력됨
  - 도달 시간 (음속 v = 340 [m/s])

$$v = \frac{2L}{t} \Rightarrow L = \frac{v \cdot t}{2} = \frac{340[\mathrm{m/s}] \cdot t[\mathrm{s}]}{2} = \frac{0.034[\mathrm{cm}/\mu\mathrm{s}] \cdot t[\mu\mathrm{s}]}{2} = 0.017 \cdot t\ [\mathrm{cm}]$$



예제 8-1

- 회로구성



아두이노 우노 : 1개, 초음파 센서 : HC-SR04, 점퍼선 여러 개

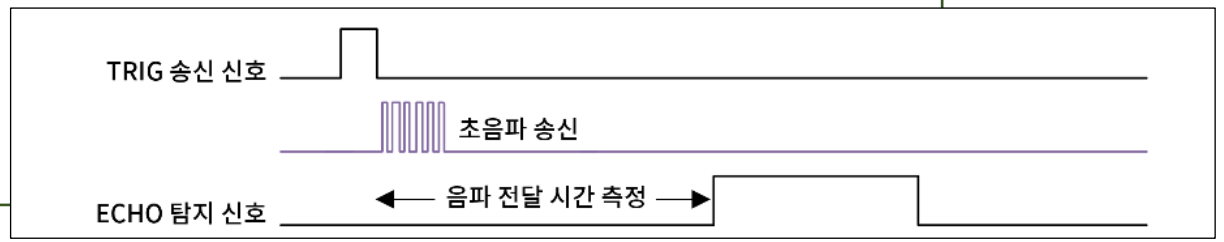
예제 8-1: 스케치

예제 8-1. 초음파 센서의 사용

```
int trig = 9;    // TRIG
int echo = 10;   // ECHO

void setup()
{
  // TRIG는 출력용으로 설정
  pinMode(trig, OUTPUT);
  // ECHO는 입력용으로 설정
  pinMode(echo, INPUT);
  // 시리얼 통신 초기화
  Serial.begin(9600);
}
```

예제 8-1: 스케치

```
void loop()
{
  // TRIG 신호 만들기
  // - 2us 동안 LOW 신호 출력 : 전에 HIGH로 되어있을 경우를 방지
  digitalWrite(trig, LOW);
  delayMicroseconds(2);
  digitalWrite(trig, HIGH); // - 10us 동안 HIGH 신호 출력
  delayMicroseconds(10);
  digitalWrite(trig, LOW);

  // ECHO 신호 입력받기, echo핀에 HIGH가 될때까지 시간 측정
  long duration = pulseIn(echo, HIGH);
  // 시간을 거리로 환산
  double distance = duration * 0.017;
  // 시리얼 포트를 통해서 전송
  Serial.println(distance);
  // 잠시 대기하기
  delay(10);
}
```



예제 8-1

- 시간 측정용 함수
  - pulseIn() 함수 : 지정된 신호가 들어올 때까지의 시간 측정
    - long duration = pulseIn(echo, HIGH);
    - echo핀에 HIGH가 들어올 때까지 기다리면서 시간 측정 (단위 us)

- 3핀의 초음파 센서 사용
  - 3핀 센서 : TRIG핀과 ECHO핀이 동일한 핀을 사용함
  - 연결된 핀을 OUPUT모드로 변경 후 펄스 송신, 그리고 INPUT모드로 변경 후 pulseIn()함수를 이용하여 측정

예제 8-1: 수정

3핀 초음파 센서의 사용시 (9번핀에 연결된 경우)

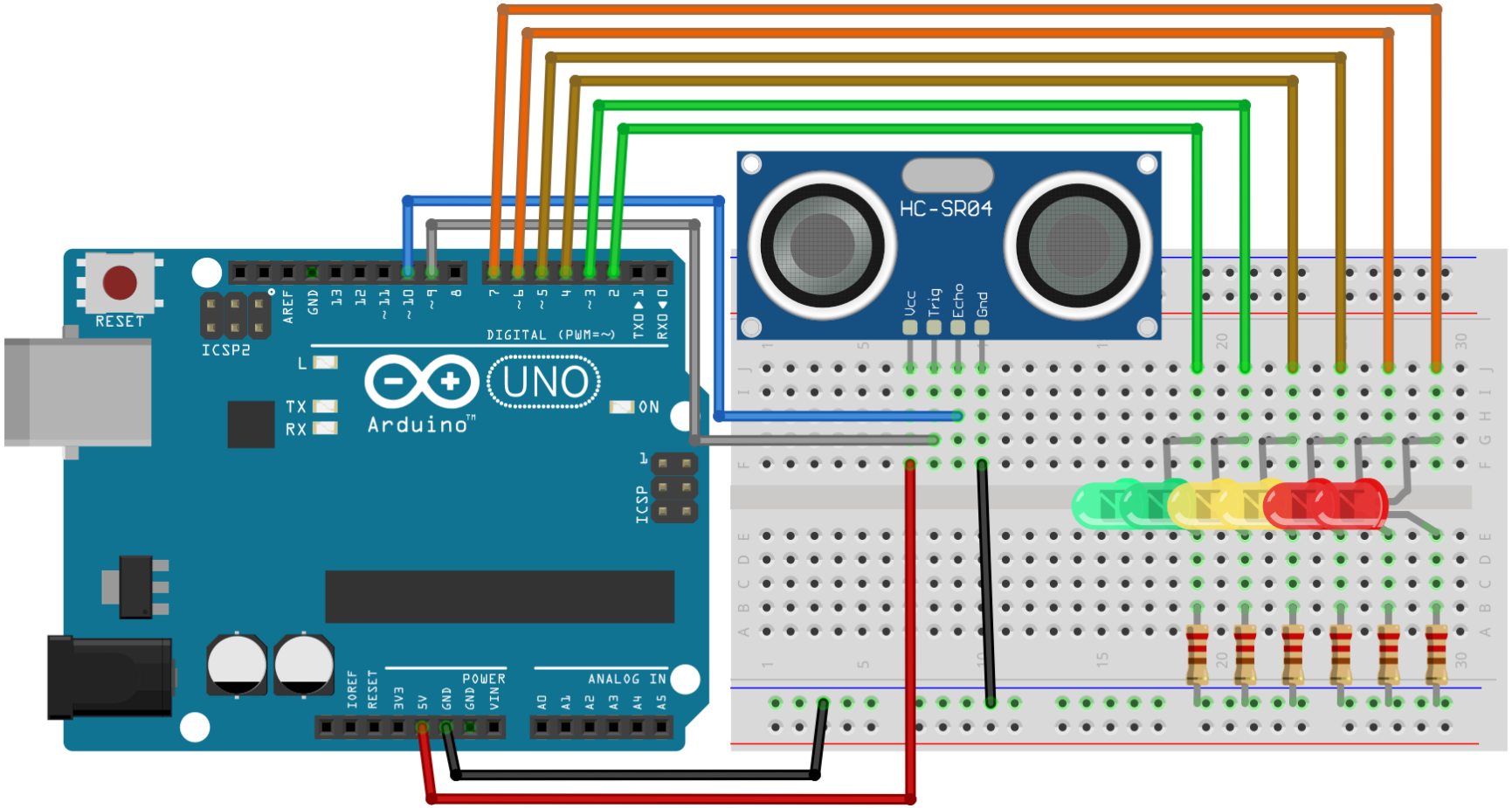
```
void loop()
{
  // TRIG 신호 만들기
  // 9번핀을 출력용으로 설정
  pinMode(9, OUTPUT);
  // - 마이크로초 동안 LOW 신호 출력
  digitalWrite(9, LOW);
  delayMicroseconds(2);
  // - 10 마이크로초 동안 HIGH 신호 출력
  digitalWrite(9, HIGH);
  delayMicroseconds(10);
  digitalWrite(9, LOW);

  // 9번핀을 입력용으로 설정
  pinMode(9, INPUT);
  // ECHO 신호 입력받기, echo핀에 HIGH가 될때까지 시간 측정
  long duration = pulseIn(9, HIGH);

  …
```

연습

- 6장의 캐릭터 LCD를 사용하여 측정된 거리를 아두이노에서 바로 표시할 수 있는 회로를 구성하고 스케치를 작성하라.

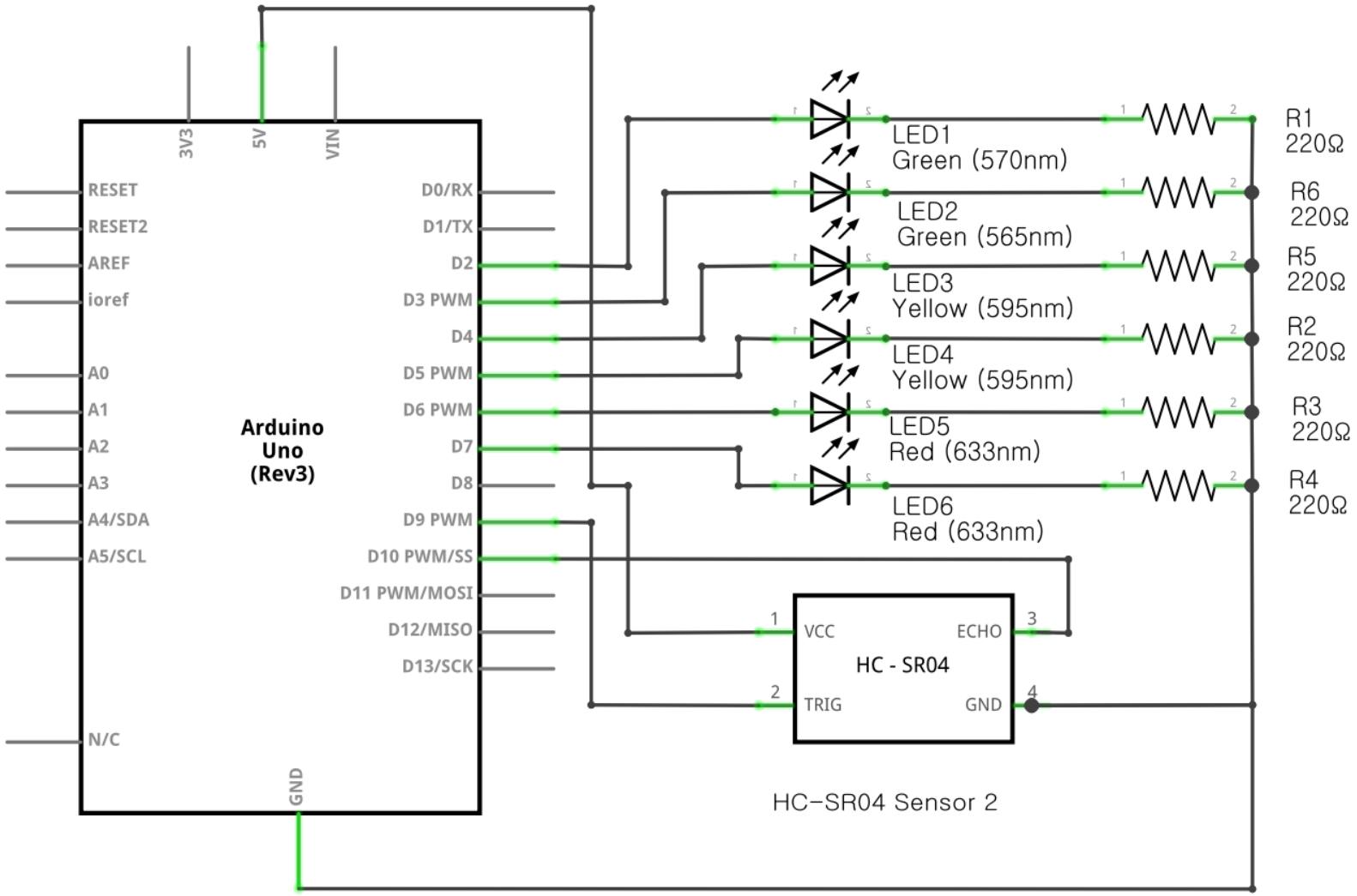
예제 8-2 : 후방 감지기

- 회로구성



아두이노 우노 : 1개, 초음파 센서 : HC-SR04, LED : 색상별 2개씩, 저항 : 220옴 6개, 점퍼선 여러 개

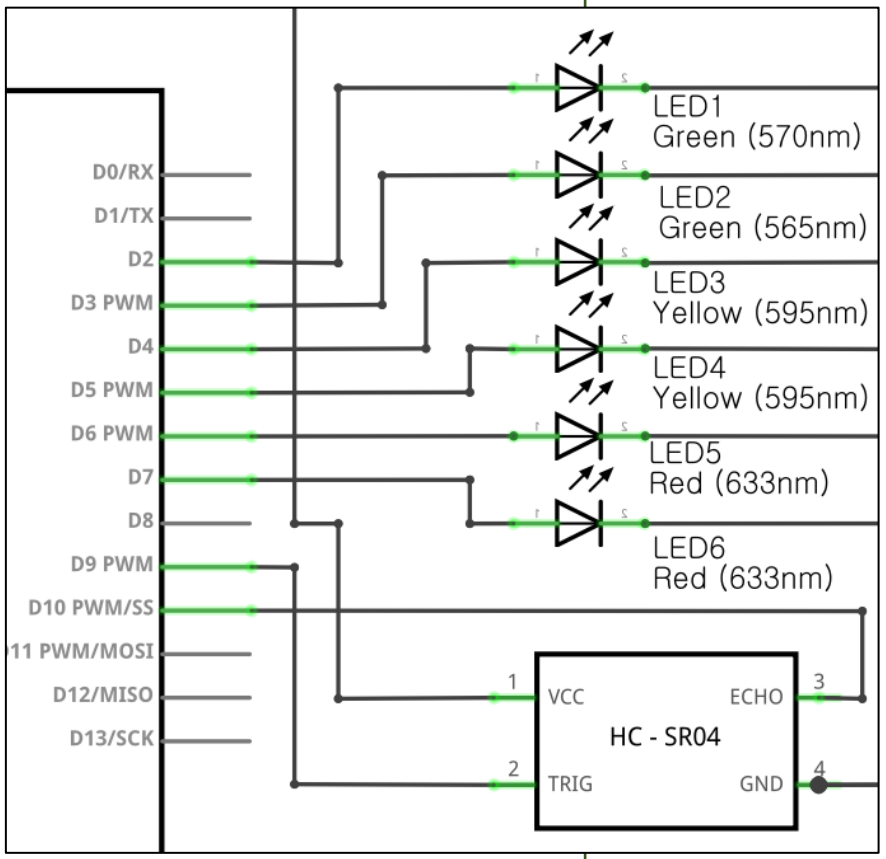
예제 8-2


3V3
5V
VIN
RESET
RESET2
AREF
ioref
A0
A1
A2
A3
A4/SDA
A5/SCL
N/C
GND
D0/RX
D1/TX
D2
D3 PWM
D4
D5 PWM
D6 PWM
D7
D8
D9 PWM
D10 PWM/SS
D11 PWM/MOSI
D12/MISO
D13/SCK
Arduino
Uno
(Rev3)
LED1
Green (570nm)
LED2
Green (565nm)
LED3
Yellow (595nm)
LED4
Yellow (595nm)
LED5
Red (633nm)
LED6
Red (633nm)
R1
220Ω
R6
220Ω
R5
220Ω
R2
220Ω
R3
220Ω
R4
220Ω
VCC
ECHO
HC - SR04
TRIG
GND
HC-SR04 Sensor 2

예제 8-2: 스케치

예제 8-2. 초음파 센서를 활용한 후방감지기

```
// 초음파 센서의 핀과 연결할 아두이노 핀 정의
int trig = 9;   // TRIG
int echo = 10;  // ECHO

// 출력할 LED의 핀들을 정의
int leds[6] = { 2, 3, 4, 5, 6, 7 };

void setup()
{
  // TRIG는 출력용으로 설정
  pinMode(trig, OUTPUT);
  // ECHO는 입력용으로 설정
  pinMode(echo, INPUT);
  // LED 핀들을 출력용으로 설정
  for(int i=0; i<6; i++)
    pinMode(leds[i], OUTPUT);
  // 시리얼 통신 초기화
  Serial.begin(9600);
}
```



예제 8-2: 스케치

예제 8-2. 초음파 센서를 활용한 후방감지기

```
void loop()
{
  // TRIG 신호 만들기
  // - 2us 동안 LOW 신호 출력
  //   이 전에 HIGH로 되어있을 경우를 방지
  digitalWrite(trig, LOW);
  delayMicroseconds(2);
  // - 10us 동안 HIGH 신호 출력
  digitalWrite(trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig, LOW);

  // ECHO 신호 입력받기, echo핀에 HIGH가 될때까지 시간 측정
  long duration = pulseIn(echo, HIGH);
  // 시간을 거리로 환산
  double distance = duration * 0.017;
  Serial.println(distance);

  ……
```

예제 8-1과 동일한 코드

예제 8-2: 스케치

예제 8-2. 초음파 센서를 활용한 후방감지기

```
  ……

  // LED를 출력하기 위하여 값의 범위를 변환
  // 10 cm ~ 100 cm 를 6 ~ 0개의 숫자 (점등할 LED 수)로 조절
  int ledno = map(distance, 10, 150, 6, 0);
  for(int i=0; i<6; i++) {
    if(i < ledno) {
      digitalWrite(leds[i], HIGH);
    }
    else {
      digitalWrite(leds[i], LOW);
    }
  }
  // 잠시 대기하기
  delay(100);
}
```

예제 8-2 : 설명

- map() 함수
  - 한 변수의 범위를 다른 범위로 계산하여 주는 함수
    - int ledno = map(distance, 10, 150, 6, 0);
    - 변수 distance의 값이 10~150사이로 변하는 경우 이 값들을 6~0 사이의 값으로 선형보간하여 변환
    - distance == 0 → ledno = 6

      …

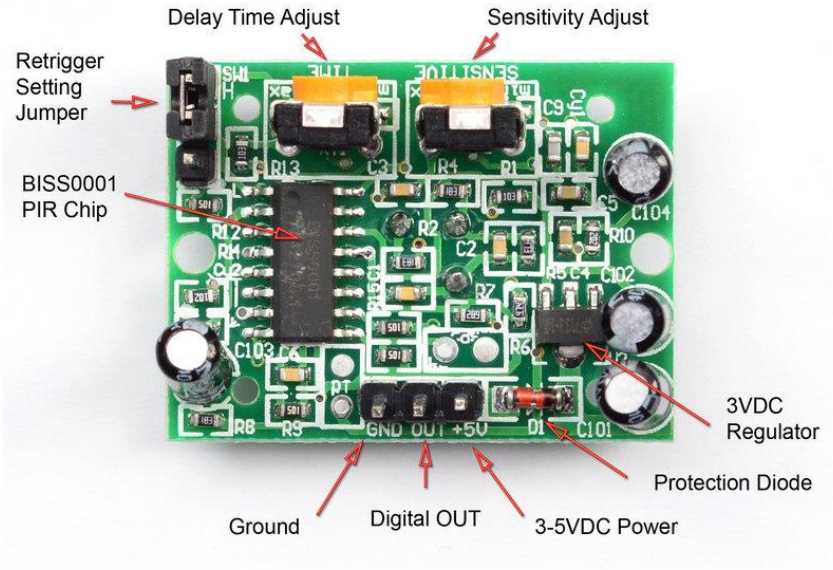
      distance == 75 → ledno = 3

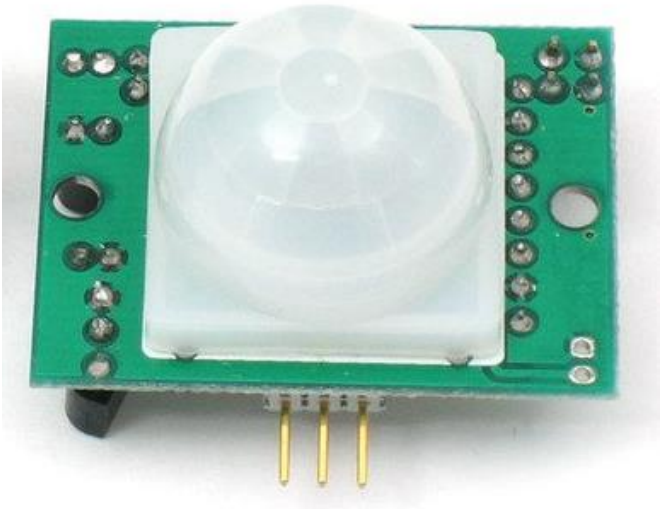
      …

      distance == 150 → ledno = 0

8.3 적외선 인체감지 센서

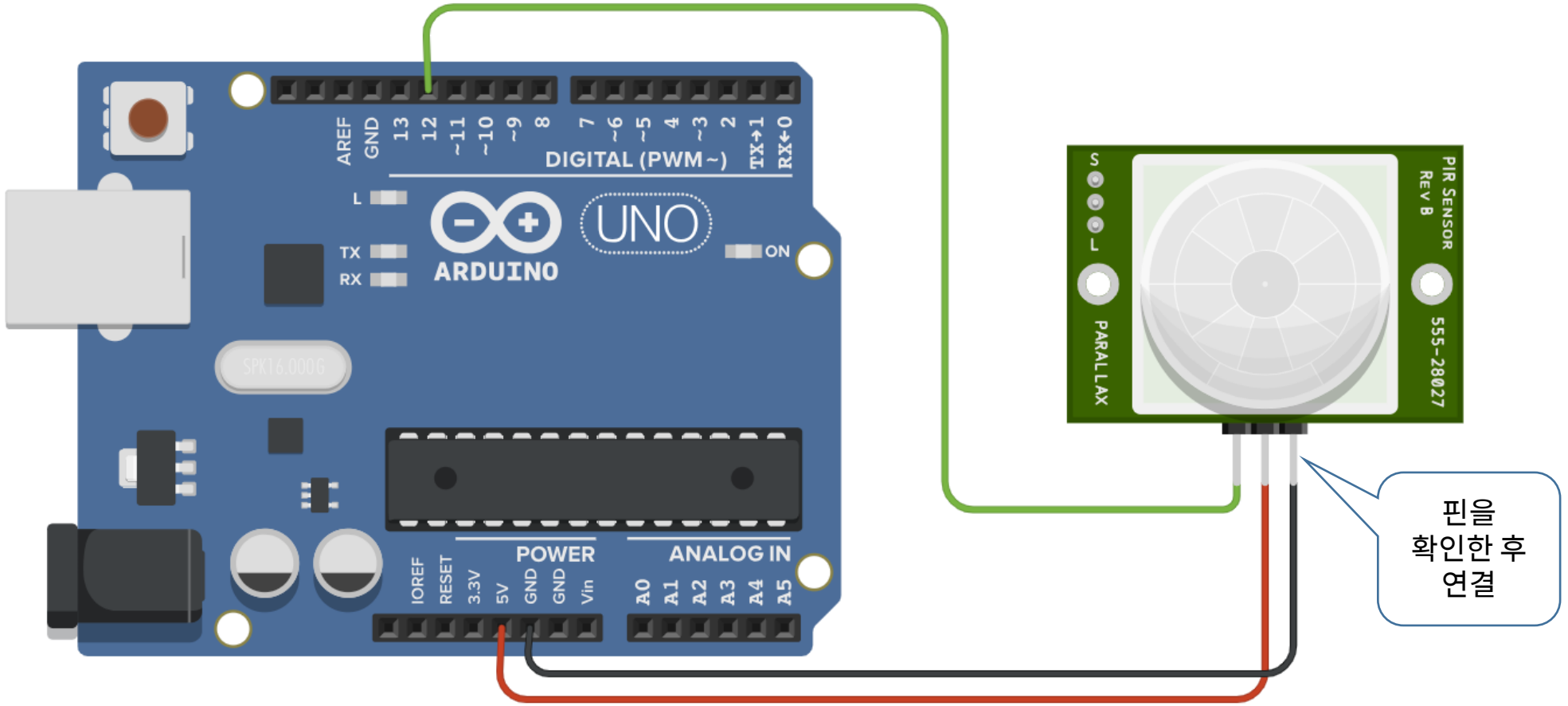
- 적외선 인체감지 센서 모듈 : PIR(Passive Infra-red)
  - 적외선 센서가 내장되어 있어 주위에 인체가 발산하는 적외선의 변화가 생기는 경우 일정 시간 동안 HIGH 신호를 출력으로 보내는 센서
  - 앞면 : dome 형태의 편광 필터가 있으며 외부에 노출
  - 뒷면 : 2개의 가변저항으로 감도/HIGH신호 유지 시간 조절





예제 8-3

- 회로구성



아두이노 우노 : 1개, 적외선 인체감지 센서 : 1개, 점퍼선 여러 개

예제 8-3: 스케치

예제 8-3. 적외선 인체감지 센서

```
// PIR 센서를 사용할 핀 정의
int pir = 12;
// 아두이노 보드의 LED 출력
int led = 13;

void setup()
{
  // PIR센서가 연결된 핀을 입력용으로 설정
  pinMode(pir, INPUT);
  pinMode(led, OUTPUT); // LED를 출력용으로 설정
  Serial.begin(9600);
}

void loop()
{
  int value = digitalRead(pir); // 센서를 통해서 값을 읽어들임
  // 센서값에 따라 LED를 켬
  if(value) digitalWrite(led, HIGH);
  else digitalWrite(led, LOW);
  Serial.println(value);
  delay(100);
}
```

8.4 온습도 센서

- 온도/습도 센서
  - DHT11 : 아두이노에서 사용가능한 대표적인 온도/습도 센서
    - ➢ 온도사양 (Temperature Specification)
      - ✓ 분해능(Resolution) : 1 ℃
      - ✓ 정확도(Accuracy) : ± 2 ℃
      - ✓ 측정범위(Measuring Range) : 0~50 ℃
    - ➢ 습도사양 (Humidity Specification)
      - ✓ 분해능(Resolution) : 1% RH
      - ✓ 정확도(Accuracy) : ± 5% RH (0~50°C)
      - ✓ 측정범위(Measuring Range) : 20~90% RH (25°C)
    - ➢ 사양 (Specification) :
      - ✓ 동작전압(Operating Voltage) : 3.3 ~ 5V
      - ✓ 권장 보관 조건(Recommended storage conditions) : 온도 10~40 ℃ / 습도 60% RH 이하

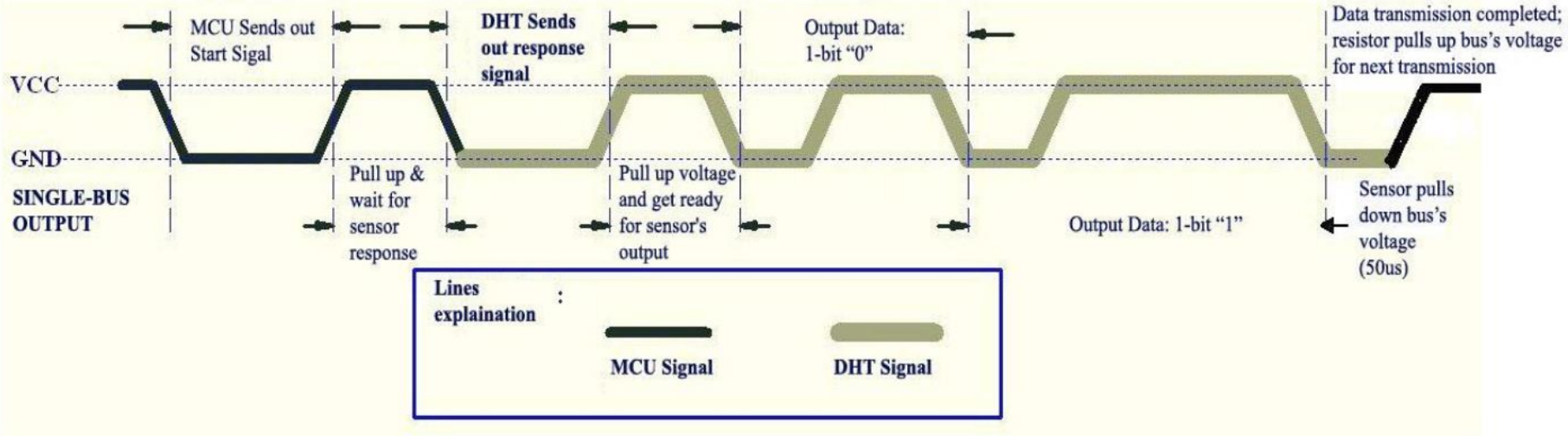
8.4 온습도 센서

- AM2302센서 (DHT22)
  - DHT11에 비해서 고성능의 센서
    - ➢ 온도사양 (Temperature Specification) :
      - ✓ 분해능(Resolution) : 0.1 ℃
      - ✓ 정확도(Accuracy) : ± 0.5 ℃
        - 측정범위(Measuring Range) : -40~80 ℃
    - ➢ 습도사양 (Humidity Specification) :
      - 분해능(Resolution) : 0.1% RH
      - 정확도(Accuracy) : ± 2% RH
    - ➢ 단점 : 측정시간이 더 필요함

- 연결
  - DHT11/DHT22 : 4개의 핀이 있으며 추가 회로구성이 필요함
  - 센서 모듈을 사용함으로 포트에 직접 연결이 가능해짐

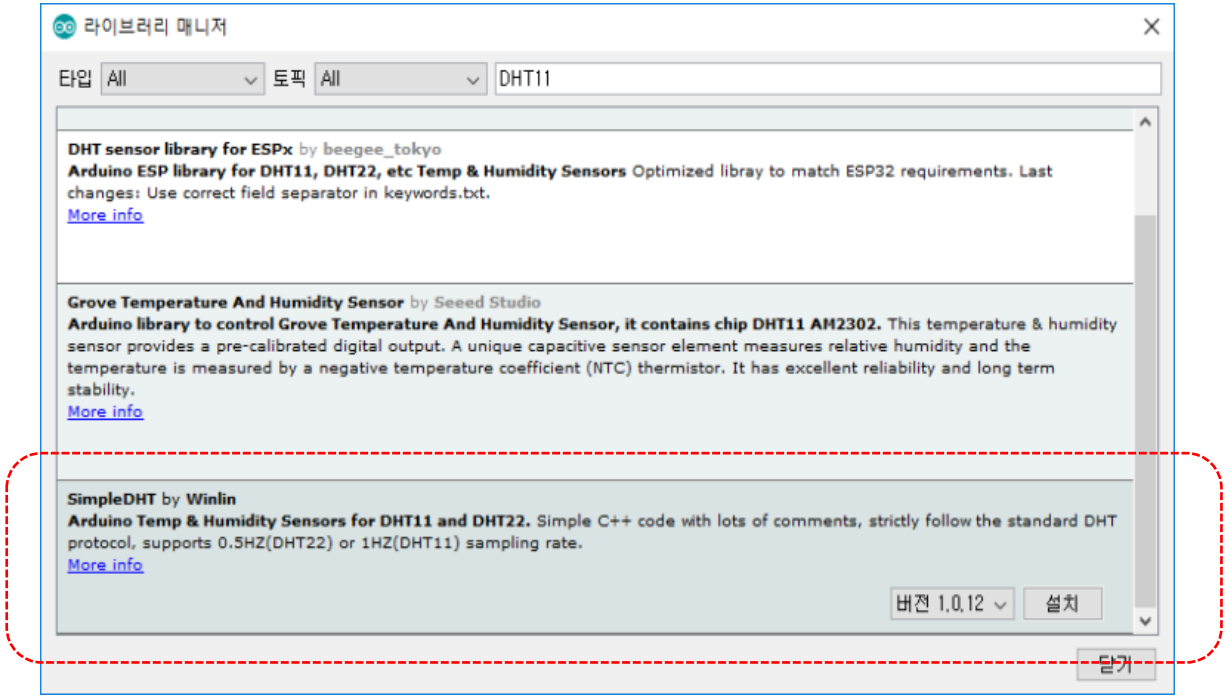
8.4 온습도 센서

- 데이터 전송 방법
  - 1개의 데이터 선을 통해서 아두이노와 데이터를 주고 받는 방식 (Single Wire Two Way)
  - 아두이노에서 시작 신호를 전송한 후 대기 → 센서에서 온도와 습도 정보를 전송 (응답신호 + 40비트 이진 신호 전송)
  - 복잡함 ⇒ [SimpleDHT by Winlin] 라이브러리 사용



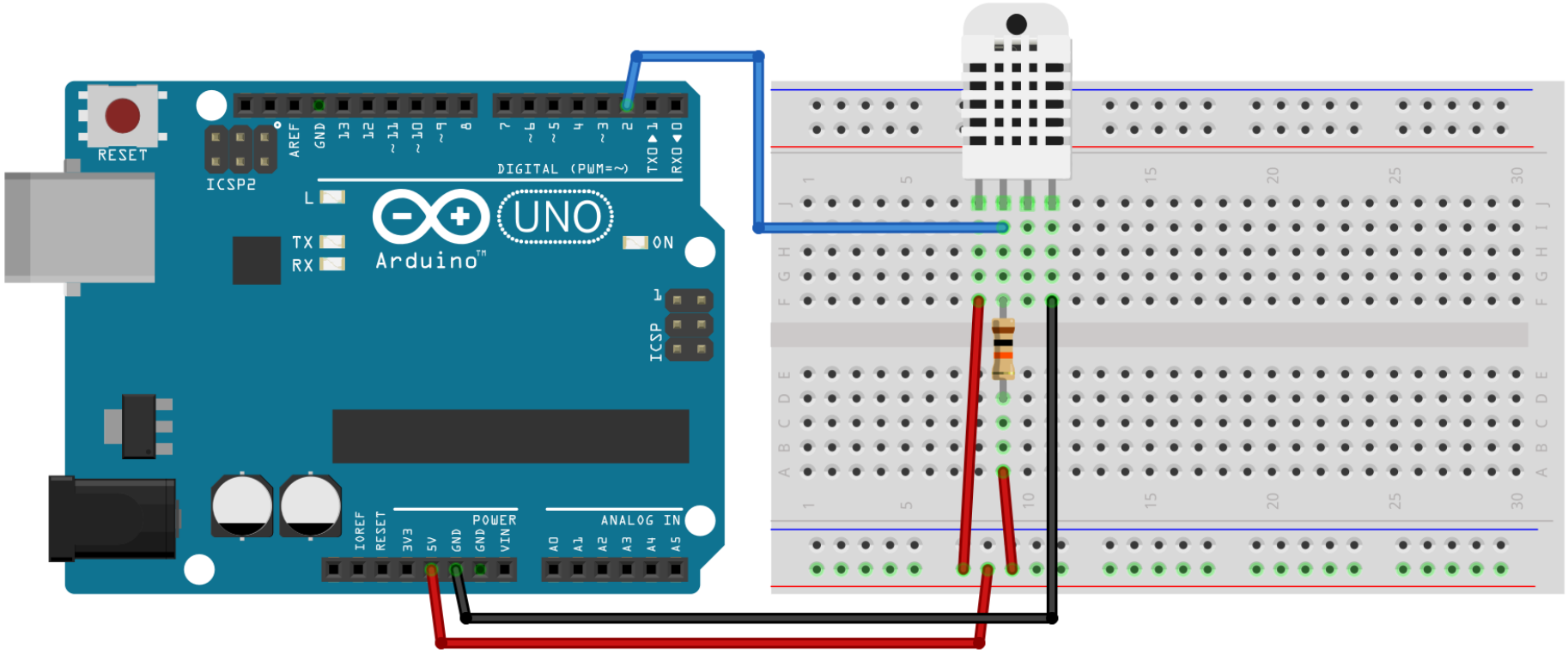
8.4 온습도 센서

- DHT11/DHT22 라이브러리
  - 라이브러리 매니저를 이용하여 SimpleDHT 라이브러리 설치



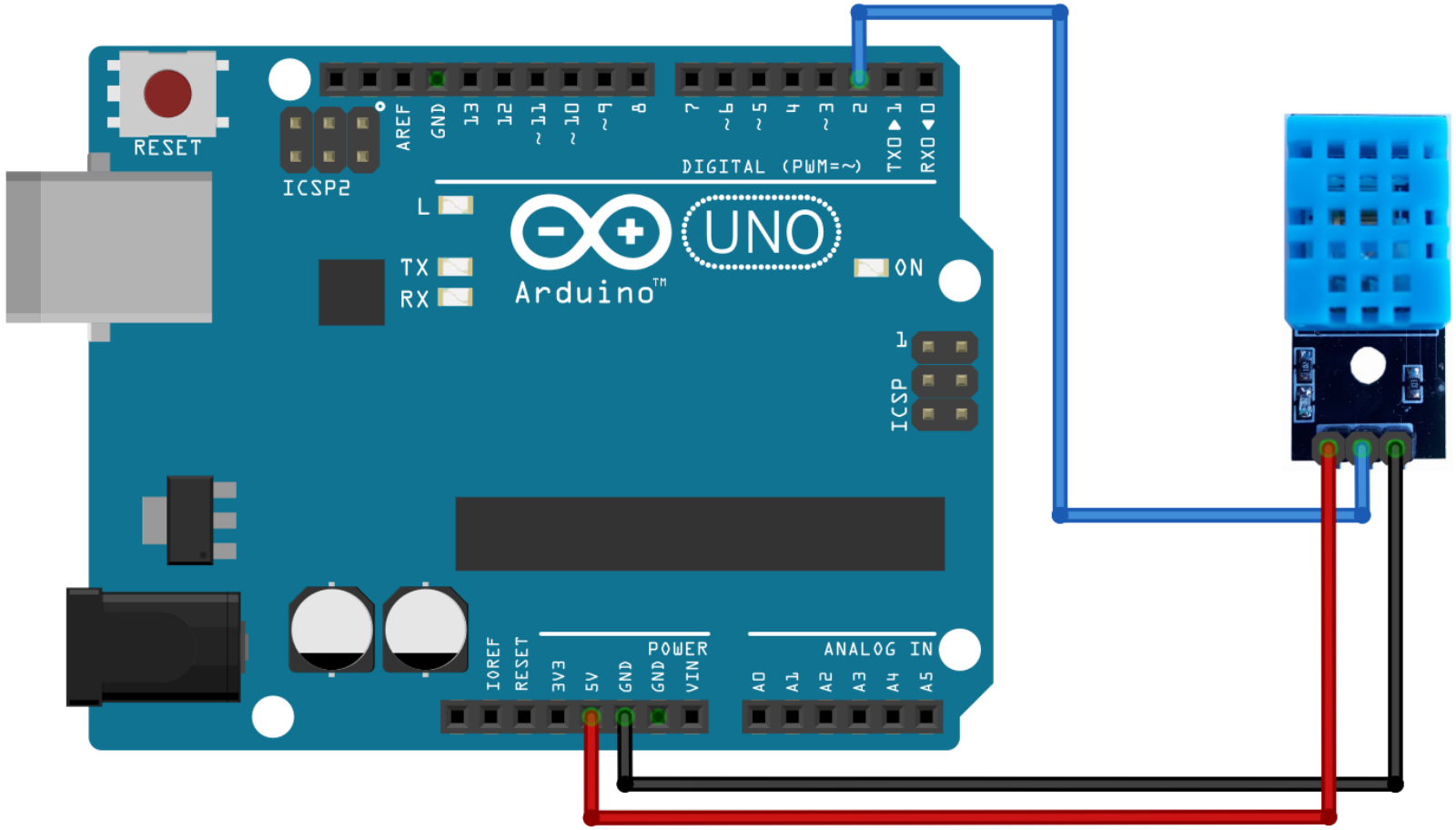
예제 8-4

- 회로구성


RESET
ICSP2
AREF
GND
13
12
~ 11
~ 10
~ 9
8
7
~ 6
~ 5
4
~ 3
2
TX→1
RX←0
DIGITAL (PWM=~)
L
TX
RX
UNO
Arduino™
ON
ICSP
POWER
ANALOG IN
IOREF
RESET
3V3
5V
GND
GND
VIN
A0
A1
A2
A3
A4
A5

예제 8-4

- 회로구성 : 모듈 사용시



예제 8-4: 스케치

예제 8-4. 온습도 센서 DHT11 구동

```
// DHT11을 구동시키기 위한 헤더파일
#include <SimpleDHT.h>

// DHT11 센서모듈이 연결된 데이터 선
int pin = 2;

// DHT11 센서모듈을 사용하기 위한 객체 정의, 생성자에 핀 번호 사용
SimpleDHT11 dht11(pin);

void setup() {
  // 시리얼 통신 초기화
  Serial.begin(9600);
}
```

예제 8-4: 스케치

예제 8-4. 온습도 센서 DHT11 구동

```
void loop() {
  // 측정 시작
  Serial.println("Start DHT11 sensing...");

  // 측정용 변수
  byte temperature = 0; // 온도
  byte humidity = 0;    // 습도

  // 온도 측정 결과의 성공 여부를 판단하기 위한 변수
  int err = SimpleDHTErrSuccess;

  // 측정 결과를 읽어오면서, 오류 여부를 검사하기
  if((err = dht11.read(&temperature, &humidity, NULL)) != SimpleDHTErrSuccess) {
    // 오류 발생시 처리할 내용
    Serial.print("Read DHT11 failed, error code =");
    Serial.println(err);
    delay(1000);
    return;
  }
  ……
```

예제 8-4: 스케치

예제 8-4. 온습도 센서 DHT11 구동

```
  ……

  // 측정 결과를 출력하기
  // 온도 측정 결과 출력
  Serial.print("Temperature : ");
  Serial.print(temperature);
  Serial.println(" °C");
  // 습도 측정 결과 출력
  Serial.print("Humidity : ");
  Serial.print(humidity);
  Serial.println(" %");
  // 빈 줄을 하나 더 출력
  Serial.println("");

  // DHT11 센서의 경우 1초에 1번 측정 가능하므로 1초동안 대기
  delay(1500);
}
```

- SimpleDHT11 라이브러리 사용
  - 객체 생성 : 연결된 핀 번호를 생성자에 전달
    - SimpleDHT11 dht11(9);
  - 온도 측정 : 온도/습도를 저장할 변수를 인수로 전달
    - dht11.read(&temperature, &humidity, NULL));
  - DHT22를 사용하는 경우

```
// 전연변수
SimpleDHT22 dht22(9); // dht11과 동일한 형식으로 정의

... // 생략
loop() {
  ... // 생략
  // 측정용 변수 정의
  float temperature = 0; // 온도
  float humidity = 0;    // 습도
  // 측정 결과를 읽어오면서, 오류 여부를 검사하기
  dht22.read(&temperature, &humidity, NULL);
  ... // 생략
  delay(2500); // 2초 이상 대기해야 함
}
```