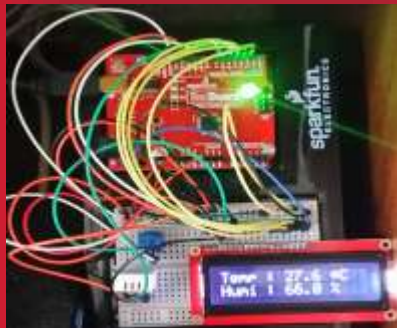




# Arduino-basic

## [wk11]

## Analog Input II.



Learn how to code Arduino from scratch

Comsi, INJE University

1<sup>st</sup> semester, 2023

Email : [chaos21c@gmail.com](mailto:chaos21c@gmail.com)



# My ID (ARnn, github repo)

AR01	강동하
AR02	정재윤
AR03	유석진
AR04	정창민
AR05	정희서
AR06	유동기
AR07	장세진
AR08	정호기

위의 **id**를 이용해서 **github**에 **repo**를 만드시오.



# [Practice]

## ◆ [wk10]

- **Arduino : Analog input I.**
- **Complete your project**
- **Submit folder : Arnn\_Rpt09**

# wk10 : Practice-09 : ARnn\_Rpt09

## ◆ [Target of this week]

- Complete your works
- Save your outcomes
- Upload all in github.

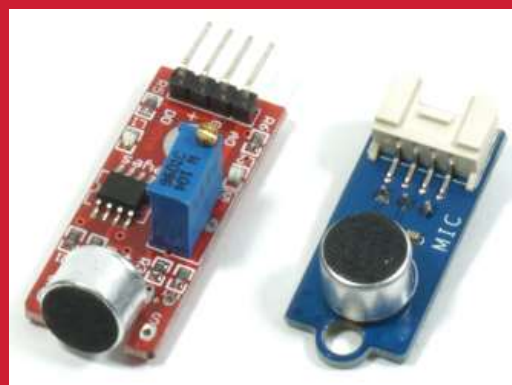
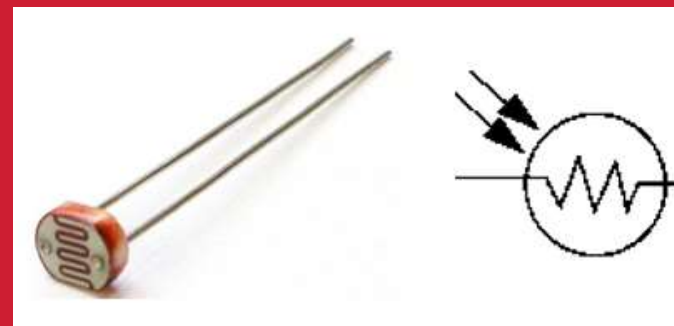
제출폴더명 : **ARnn\_Rpt09**

### 제출할 파일들

- ① **ARnn\_pwm.png**
- ② **ARnn\_pwm.ino**
- ③ **ARnn\_cds.png**
- ④ **ARnn\_cds\_project.ino**
- ⑤ **ARnn\_cds\_lm35.ino**
- ⑥ **ARnn\_cds\_lm35.png**
- ⑦ **\*.ino**



# 6. Analog input



## 6. 아날로그 신호 입력

6.1     포텐쇼미터 입력 (가변저항기)

6.2     빛 입력 (CdS, LDR)

6.3     온도 측정 (LM35, TMP36)

6.4     수위 측정

6.5     아날로그 조이스틱

6.6     소리 입력

[DIY] TMP36



# 6.4

## 수위센서



## 6.4 수위 센서 (수위 측정)



그림 6.5 디지털 신호 수위센서와(a) 실험에 사용할 아날로그 신호 수위센서(b)

- ✓ 디지털 센서는 만수를 감지
- ✓ 아날로그 센서는 수위를 측정
- ✓ 디지털 입력핀 혹은 아날로그 입력핀을 이용하여 측정



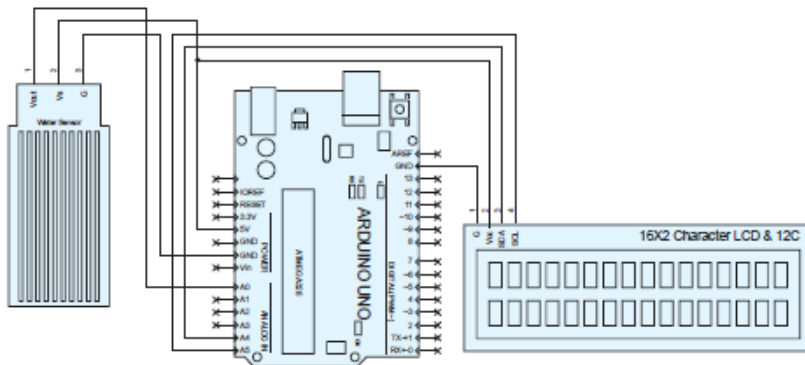
# 6.4.1 수위 센서 (수위 측정)

## EX 6.4

## 수위 측정 (1/2)

- 실습목표**
1. 수위 센서로부터 컵 안의 물의 수위를 측정한다.
  2. 아날로그 입력값과 % 값을 함께 출력한다.
  3. 센서마다 만수 시 출력값이 틀릴 수 있다. ADC 값을 확인 한 후 상수를 변경하자.

- Hardware**
1. 수위센서의 Vs(+)와 G(-) 핀을 Arduino의 5V와 GND에 연결한다.
  2. 수위센서의 Vout을 아날로그입력핀 A0에 연결한다.
  3. I2C LCD 모듈의 Vcc, GND를 Arduino의 5V, GND에 연결한다.
  4. I2C LCD 모듈의 SDA는 A4에 SCL은 A5에 연결한다.



## 6.4.2 수위 센서 (수위 측정)

### EX 6.4

### 수위 측정 (2/2)

- Sketch 구성**
1. I2C LCD 모듈을 설정한다.
  2. Water sensor로부터 ADC 값을 읽는다.
  3. ADC 값을 만수일 때 ADC 값과 비교하여 %로 수위를 계산한다.
  4. LCD에 ADC 값과 수위를 표시한다.

**실험 결과** ADC 값과 Water level 값이 출력된다.

```
ADC: 600
Water level : 100 %
```



## 6.4.3 수위 센서 (수위 측정) : code-1

```
ex_6.4
1 /*
2  예제 6.4
3  수위 센서 입력
4  */
5
6 // I2C 통신 라이브러리 설정
7 #include <Wire.h>
8 // I2C LCD 라이브러리 설정
9 #include <LiquidCrystal_I2C.h>
10
11 // LCD I2C address 설정 PCF8574:0x27, PCF8574A:0x3F
12 LiquidCrystal_I2C lcd(0x27,16,2); // LCD address:0x27, 16X2 LCD
13
14 // 0번 아날로그핀을 WaterLevel Sensor 입력으로 설정한다.
15 const int waterLevelPin = 0;
16
17 // 센서마다 만수시 ADC 값이 틀리므로 만수 시 ADC 값을 참고하여 설정한다.
18 const int waterFullAdcValue = 798;
```

```
20 void setup() {
21
22   lcd.init(); // LCD 설정
23   lcd.backlight(); // 백라이트를 켜다.
24
25   // 메시지를 표시한다.
26   lcd.print("ex 6.4");
27   lcd.setCursor(0,1);
28   lcd.print("Water Level");
29   // 3초동안 메시지를 표시한다.
30   delay(3000);
31
32   // 모든 메시지를 삭제한 뒤
33   // 숫자를 제외한 부분들을 미리 출력시킨다.
34   lcd.clear();
35   lcd.setCursor(0,0);
36   lcd.print("ADC : ");
37   lcd.setCursor(0,1);
38   lcd.print("Water Level:");
39   lcd.setCursor(15,1);
40   lcd.print("%");
41 }
```

## 6.4.3 수위 센서 (수위 측정) : code-2

```

43 void loop(){
44
45   int adcValue; // 실제 센서로부터 읽은 값 (0~1023)
46   int waterLevel; // 수위 0~100%
47
48   // 수위 센서를 통하여 입력되는 전압을 읽는다.
49   adcValue = analogRead(waterLevelPin);
50   // 아날로그 입력 값을 0~100의 범위로 변경한다.
51
52   waterLevel = map(adcValue, 0, waterFullAdcValue, 0, 100);
53
54   // 전에 표시했던 내용을 지우고
55   // LCD에 ADC 값과 수위를 출력한다.
56   // 지우지 않으면 이전에 표시했던 값이 남게 된다.
57
58   // 전에 표시했던 내용을 지운다.
59   lcd.setCursor(9,0);
60   lcd.print(" ");
61   // ADC 값을 표시한다
62   lcd.setCursor(9,0);
63   lcd.print(adcValue);

```

```

65   // 전에 표시했던 내용을 지운다.
66   lcd.setCursor(13,1);
67   lcd.print(" ");
68   // 수위를 표시한다
69   lcd.setCursor(12,1);
70   lcd.print(waterLevel);
71
72   delay(1000);
73 }

```

DIY

응용 문제

1. 사전에 설정한 수위에서 LED를 점등 시키는 회로를 구성하고 동작시키는 스케치를 작성해 보자. (사전 설정 수위 = 80 %)

→ 점등 사진은 **ARnn\_waterlevel.png**

→ 아두이노 코드는 **ARnn\_waterlevel.ino** 로 저장하고 제출



# 6.5

## 아날로그 조이스틱



## 아날로그 조이스틱



그림 6.6 실험에 사용할 아날로그 조이스틱 모듈

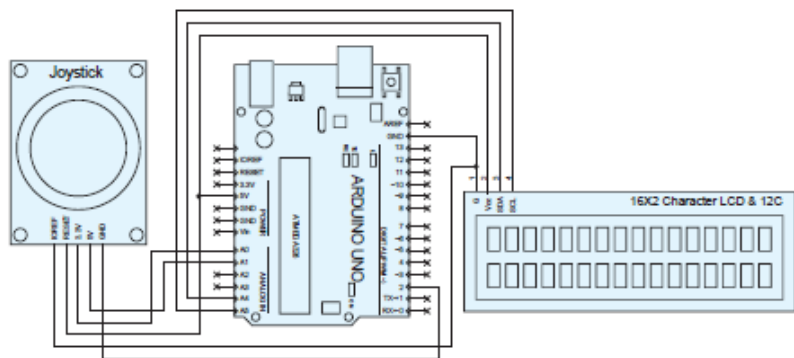
- ✓ X, Y 축의 움직임은 포텐쇼미터로 감지
- ✓ Z 축으로의 움직임은 디지털 스위치 입력

## EX 6.5

## 아날로그 조이스틱 (1/3)

- 실습목표**
1. 아날로그 조이스틱을 이용하여 X, Y 축으로 변하는 아날로그 값을 입력 받아 LCD에 출력한다.
  2. Z 축 입력에 대해서는 백라이트를 점멸시킨다.

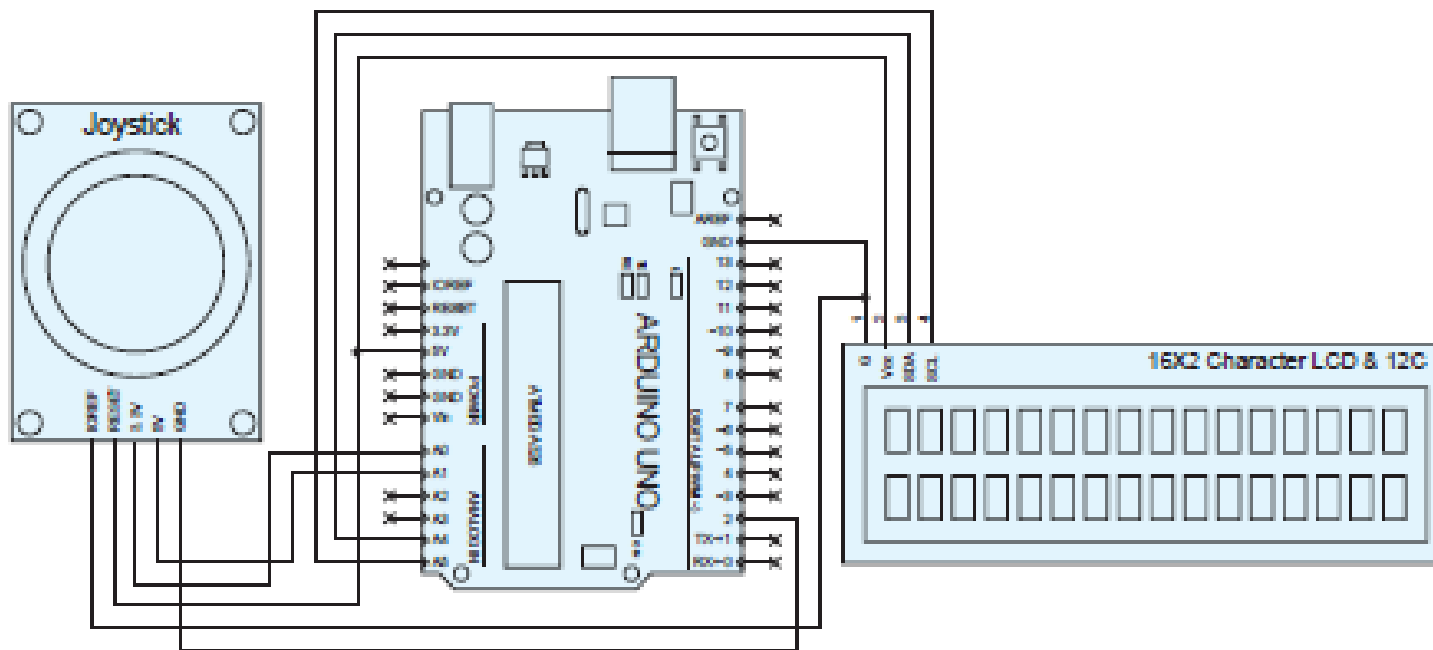
- Hardware**
1. 조이스틱의 5V와 G 핀을 Arduino의 5V와 GND에 연결한다.
  2. X축 각도인 VRX는 아날로그입력 A0핀에, Y축 각도인 VRY는 아날로그입력 A1핀에 연결한다.
  3. Z축 입력인 SW는 디지털입출력핀 2번에 연결한다.
  4. I2C LCD 모듈의 Vcc, GND를 Arduino의 5V, GND에 연결한다.
  5. I2C LCD 모듈의 SDA는 A4에 SCL은 A5에 연결한다.
  6. A0핀으로부터 X축의 아날로그 변위와 A1핀으로부터 Y축의 아날로그변위를 ADC로 입력받는다.
  7. Z축 입력은 스위치 입력으로 디지털입출력핀 2번은 반드시 풀업 설정을 해줘야 한다.





## EX 6.5

## 아날로그 조이스틱 (2/3)



X축 각도인 VRX는 아날로그입력 A0핀에, Y축 각도인 VRY는 아날로그입력 A1핀에 연결.

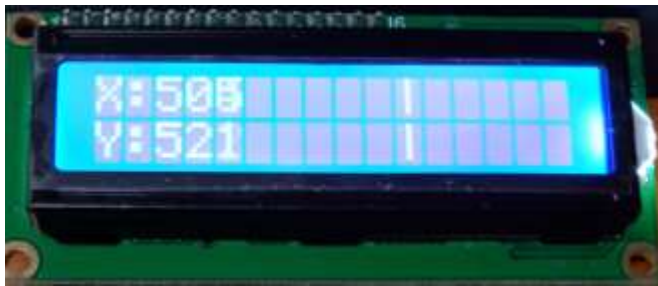
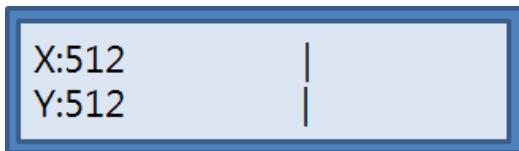
Z축 입력인 SW는 디지털입출력핀 2번에 연결 (INPUT\_PULLUP)

## EX 6.5

## 아날로그 조이스틱 (3/3)

- Sketch 구성
1. X, Y축의 움직임에 대하여 아날로그 입력핀 0번과 1번에서 **아날로그 입력**을 받는다.
  2. 디지털입력핀 2번에서 Z축으로의 **디지털 입력**을 받는다.
  3. X, Y축의 움직임을 LCD에 그래프로 나타내고 ADC 값도 함께 나타내 준다.
  4. Z축 디지털 입력이 발생했을 경우 **백라이트를 점멸**시킨다.

실행 결과 X, Y 축의 아날로그값과 그래프가 출력된다. 조이스틱을 누르면 백라이트가 점멸한다.



ex\_6\_5

```
1 /*
2  예제 6.5
3  조이스틱 입력
4  */
5
6  // I2C 통신 라이브러리 설정
7  #include <Wire.h>
8  // I2C LCD 라이브러리 설정
9  #include <LiquidCrystal_I2C.h>
10
11 // LCD I2C address 설정 PCF8574:0x27, PCF8574A:0x3F
12 LiquidCrystal_I2C lcd(0x27,16,2); // LCD address:0x27, 16X2 LCD
13
14 // 0번 아날로그핀을 X 축 입력으로 설정
15 const int xAxisPin = 0;
16 // 1번 아날로그핀을 Y 축 입력으로 설정
17 const int yAxisPin = 1;
18 // 2번 디지털 입력 핀을 Z 축 입력으로 설정
19 const int zAxisPin = 2;
20
```

```
21 void setup() {
22
23     // Z 축 입력은 디지털 입력으로 설정한다.
24     pinMode(zAxisPin, INPUT_PULLUP);
25
26     lcd.init(); // LCD 설정
27     lcd.backlight(); // 백라이트를 켜다.
28
29     // 메시지를 표시한다.
30     lcd.print("ex 6.5");
31     lcd.setCursor(0,1);
32     lcd.print("Joystick");
33     // 3초동안 메시지를 표시한다.
34     delay(3000);
35
36     // 모든 메시지를 삭제한 뒤
37     // X축 Y축 문자를 출력한다.
38     lcd.clear();
39     lcd.setCursor(0,0);
40     lcd.print("X:");
41     lcd.setCursor(0,1);
42     lcd.print("Y:");
43     lcd.setCursor(15,1);
44 }
```

## 6.5.3 아날로그 조이스틱: code-2

```

46 void loop(){
47
48   // X, Y, Z 축 값을 읽는다.
49   int xValue = analogRead(xAxisPin);
50   int yValue = analogRead(yAxisPin);
51   int zValue = digitalRead(zAxisPin);
52
53   // 그래프를 그리기 위해서 X, Y 값을 조절한다.
54   int xDisplay = map(xValue,0,1023,6,15);
55   int yDisplay = map(yValue,0,1023,6,15);
56
57   // 첫 째 줄에 전에 표시했던 내용을 지운다.
58   lcd.setCursor(2,0);
59   lcd.print("          "); // 14칸 공백
60   // X 축의 ADC 값을 출력한다.
61   lcd.setCursor(2,0);
62   lcd.print(xValue);
63   // 조이스틱의 X 값에 따라 그래프를 출력한다
64   lcd.setCursor(xDisplay,0);
65   lcd.print("|");

```

```

67   // 둘째 줄에 전에 표시했던 내용을 지운다.
68   lcd.setCursor(2,1);
69   lcd.print("          "); // 14칸 공백
70   // Y 축의 ADC 값을 출력한다.
71   lcd.setCursor(2,1);
72   lcd.print(yValue);
73   // 조이스틱의 Y 값에 따라 그래프를 출력한다
74   lcd.setCursor(yDisplay,1);
75   lcd.print("|");
76
77   // Z 방향으로 눌렀을 때 백라이트를 점멸한다.
78   if(zValue == LOW){
79     lcd.noBacklight();
80     delay(300);
81     lcd.backlight();
82   }
83   delay(100);
84 }

```

## 6.5.4 아날로그 조이스틱 - DIY (참고)

### DIY

1. 5개의 LED를 브레드보드에 '+' 모양으로 배치시킨다.

### 응용 문제

2. 조이스틱의 방향에 따라 해당하는 LED를 점등시키는 스케치를 작성해 보자.



## DIY

1. 5개의 LED를 브레드보드에 '+' 모양으로 배치시킨다. (white → green)

## 응용 문제

2. 조이스틱의 방향에 따라 해당하는 LED를 점등시키는 스케치를 작성해 보자.

→ **ARnn\_joystick.ino** 로 저장하고 제출

```
21 // LED
22 const int wShoot = 3; // white Shoot
23 const int rStop = 5; // red Stop
24 const int bGo = 6; // blue Go
25 const int yRight = 9; // yellow Right
26 const int yLeft = 10; // yellow Left
```

```
32 // LED
33 pinMode(wShoot, OUTPUT);
34 pinMode(rStop, OUTPUT);
35 pinMode(bGo, OUTPUT);
36 pinMode(yRight, OUTPUT);
37 pinMode(yLeft, OUTPUT);
```

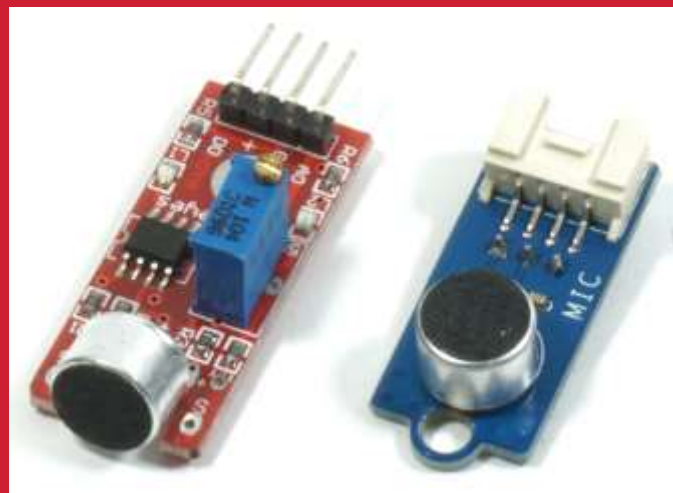
동작 중 사진을

**ARnn\_joystick.png** 로 저장...



6.6

# 마이크로폰 모듈



## 마이크론 모듈



그림 6.7 마이크론 모듈

- ✓ 입력되는 소리 신호와 비례하여 아날로그 신호 출력
- ✓ 디지털 출력으로 사용할 때는 내장된 포텐쇼미터로 임계값 조절

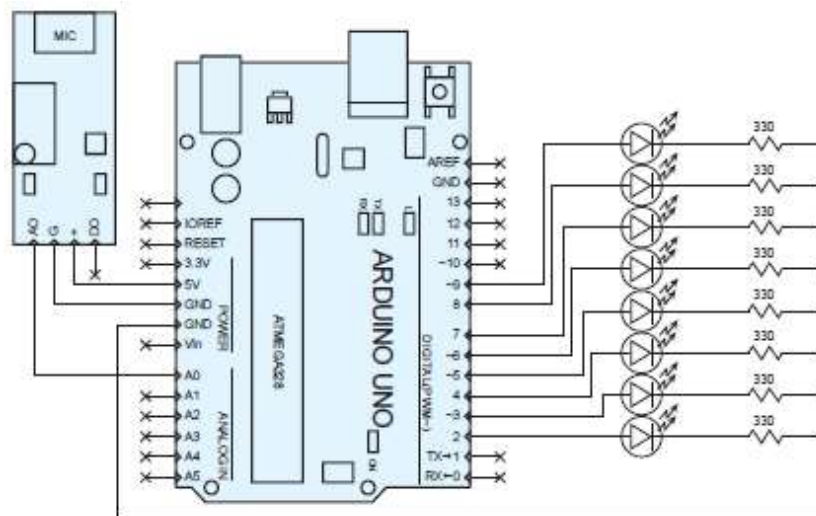


## EX 6.6

## 소리 입력 (1/3)

- 실습목표
1. 마이크론 모듈을 이용하여 소리를 아날로그 신호로 입력 받는다.
  2. 소리의 크기에 따라 8개의 LED로 그래프 바를 만들어 보자.

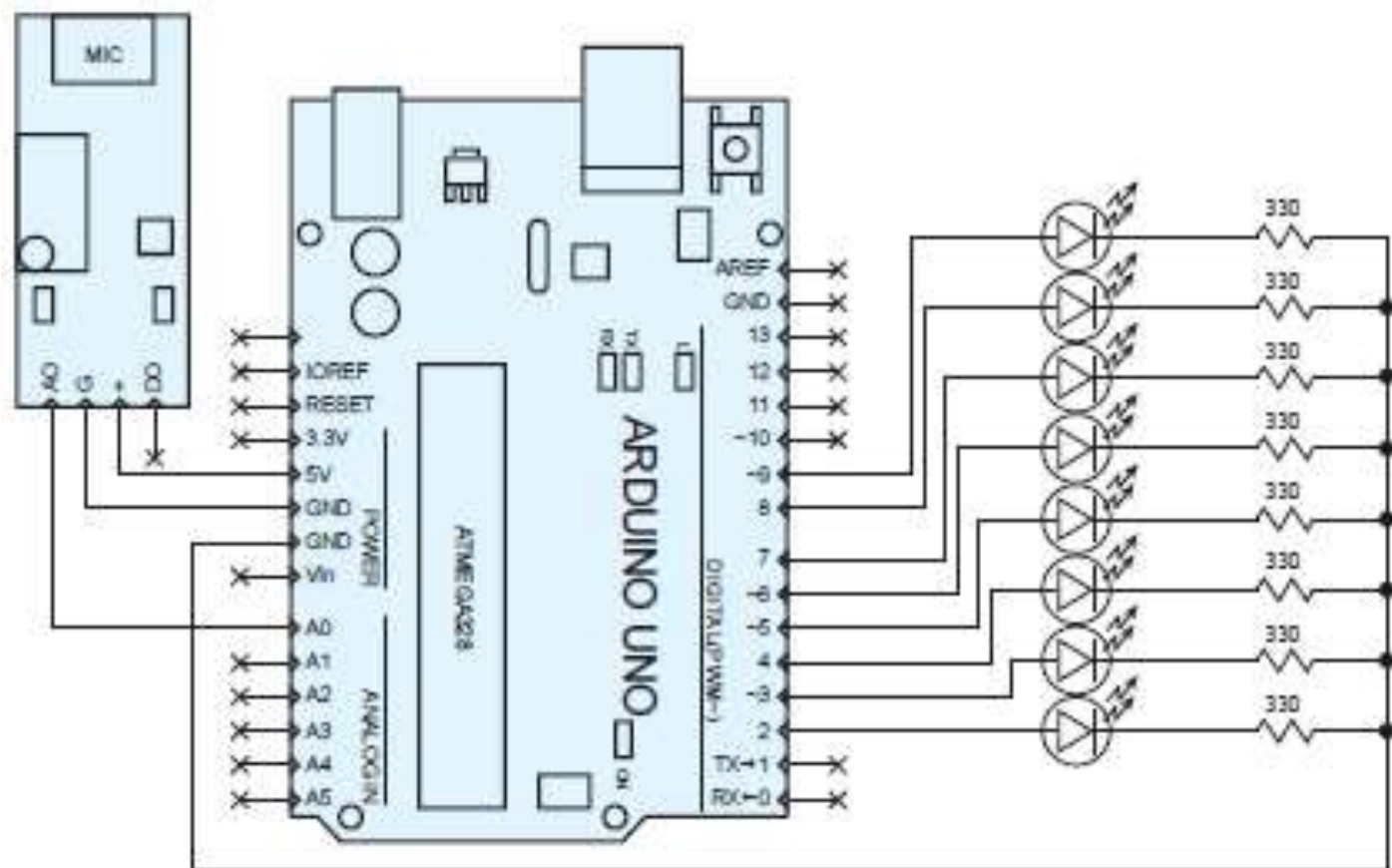
- Hardware
1. LED 바를 만들기위해 2~9번핀에 8개의 LED를 연결한다. Anode를 Arduino의 핀에 연결하고 Cathode에 220 $\Omega$  저항을 연결하여 GND에 연결한다.
  2. 마이크론 모듈의 +와 G를 Arduino의 5V와 GND에 연결한다.
  3. 마이크론 모듈의 AO핀을 Arduino의 아날로그 입력핀 A0에 연결한다.
  4. 입력되는 소리의 크기를 ADC로 읽어 LED바를 동작하는데 참고한다.



# 6.6.1 마이크론 모듈

EX 6.6

소리 입력



## EX 6.6

## 소리 입력 (2/3)

### Commands

- `analogRead(아날로그 핀번호)`

아날로그핀에서 아날로그 값을 읽는다. 0~5V사이의 전압을 0~1023사이의 값으로 표현한다.

- `map(변수명, 범위1 최소값, 범위1 최대값, 범위2 최소값, 범위2 최대값)`

변수명의 변수의 범위1의 범위와 범위2의 범위에 매칭시킨다. 즉 변수가 0~100의 범위를 갖고 이를 50~200의 범위로 매칭하려면 'map(변수명, 0, 100, 50, 200)'의 명령어로 매칭시킬 수 있다.

- `for(변수=시작 값 ; 조건 ; 변수의 증분){ }`

변수의 시작 값부터 조건이 만족하는 경우 '{ }' 내의 명령을 수행한다. '변수의 증분'에서는 1회 명령이 수행될 때 마다 변수를 증가 혹은 감소시킨다.

### Sketch 구성

1. MIC 모듈에서 출력되는 아날로그 신호를 아날로그 입력핀 0번에서 입력받는다.
2. 디지털 출력 2~9번핀에 LED를 입력된 아날로그 값과 대응하여 단계별로 출력한다.
3. 아날로그 신호가 클수록 많은 수의 LED를 켜다.

## 6.6.2 마이크로폰 모듈 – code

ex\_6\_6

```

1  /*
2  예제 6.6
3  소리 입력
4  */
5
6  // A0번 핀에서 사운드 입력을 받는다
7  char soundInputPin = 0;
8
9  // 그래프 바 LED 출력핀을 level 변수에 저장
10 char ledLevel[8] = {2, 3, 4, 5, 6, 7, 8, 9};
11
12 void setup() {
13   // 그래프 바 LED 핀을 출력으로 설정
14   for(int i=0 ; i<=7 ; i++){
15     pinMode(ledLevel[i], OUTPUT);
16   }
17 }

```

```

19 void loop(){
20
21   // A0번 핀에서 사운드 입력을 받는다
22   int soundInput = analogRead(soundInputPin);
23   // 노이즈부분을 제외한 50~900의 범위로 입력받은 사운드 크기를
24   // 0~7단계로 변경한다.
25   int soundLevel = map(soundInput, 50, 900, 0, 7);
26
27
28   // 전체 LED를 소등한다.
29   for(int i = 0 ; i <= 7 ; i++){
30     digitalWrite(ledLevel[i], LOW);
31   }
32
33   // 0~7 단계 중 입력보다 작은 레벨의 LED는 점등한다.
34   for(int i = 0 ; i <= soundLevel ; i++){
35     digitalWrite(ledLevel[i], HIGH);
36   }

```

EX 6.6

소리 입력 (3/3)

- Sketch 구성**
1. MIC 모듈에서 출력되는 아날로그 신호를 아날로그 입력핀 0번에서 입력받는다.
  2. 디지털 출력 2~9번핀에 LED를 입력된 아날로그 값과 대응하여 단계별로 출력한다.
  3. 아날로그 신호가 클수록 많은 수의 LED를 켜다.

**실습 결과** 소리의 크기에 따라서 LED 바가 점등된다.

사진을 **ARnn\_sound\_bar.png**  
로 저장...

DIY

응용 문제

시리얼 통신을 통하여 소리의 크기를 PC 모니터에 출력해 보자.

```
COM11 (Arduino/Genuino Uno)
Sound input: 34, Sound level: 0
Sound input: 267, Sound level: 1
Sound input: 1021, Sound level: 7
Sound input: 1021, Sound level: 7
```

직렬모니터 출력화면을  
**ARnn\_sound\_monitor.png**  
로 저장...



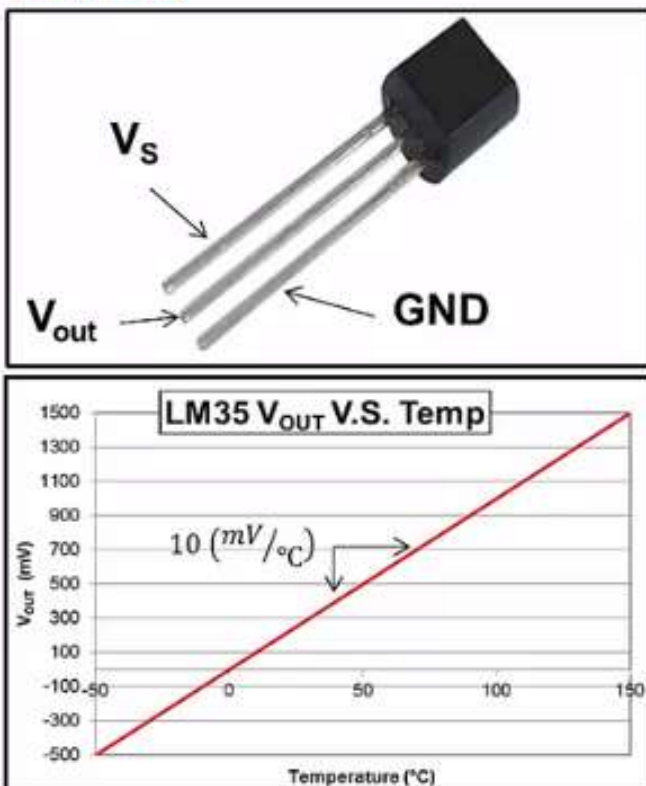
# 온도센서



## 6.3 온도 센서 (주변 온도측정)

### LM35 온도-전압 특성

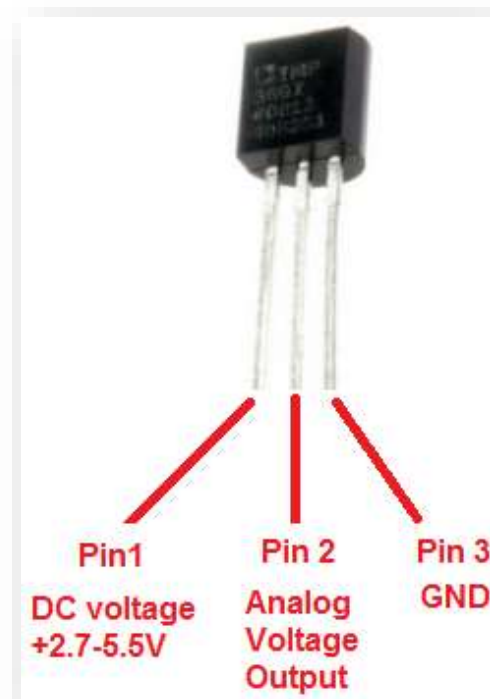
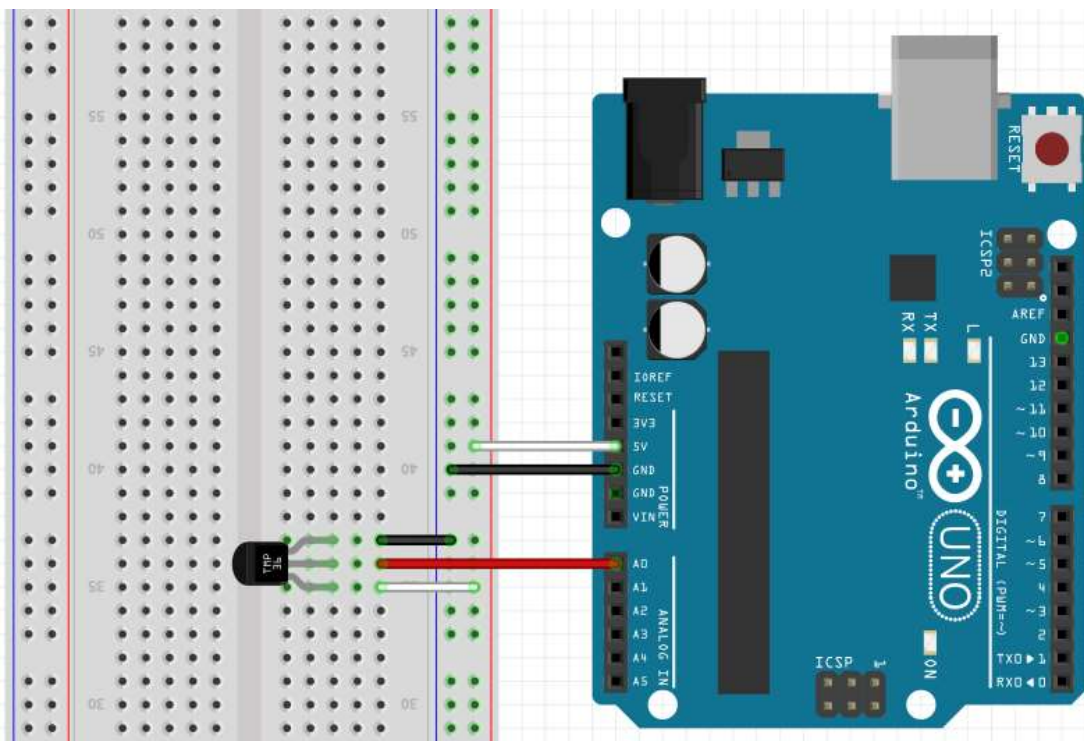
#### LM35



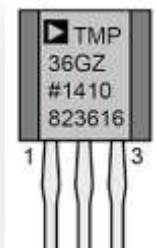
- Three-Pin
  - TO-92 Package
  - Easy to Use
  - 4V-20V Operating Range
  - 60 $\mu$ A Max Current Draw
- Analog Output
  - 0.5 $^{\circ}C$  Accuracy at 25 $^{\circ}C$
  - Easily read by Arduino
  - Highly Linear Transfer Function
  - 10 ( $mV/^{\circ}C$ ) Slope

✓ 전원과 접지를 연결하면  $V_{out}$ 에 0~500  $^{\circ}C$  까지 0.01V 단위로 전압 출력(0~5000mV)이 발생

# A3.1.1 Temperature sensor [ TMP36]



## Parts : TMP36



- **Size:** TO-92 package (about 0.2" x 0.2" x 0.2") with three leads
- **Price:** \$2.00 at the [Adafruit shop](#)
- **Temperature range:** -40°C to 150°C / -40°F to 302°F
- **Output range:** 0.1V (-40°C) to 2.0V (150°C) but accuracy decreases after 125°C
- **Power supply:** 2.7V to 5.5V only, 0.05 mA current draw

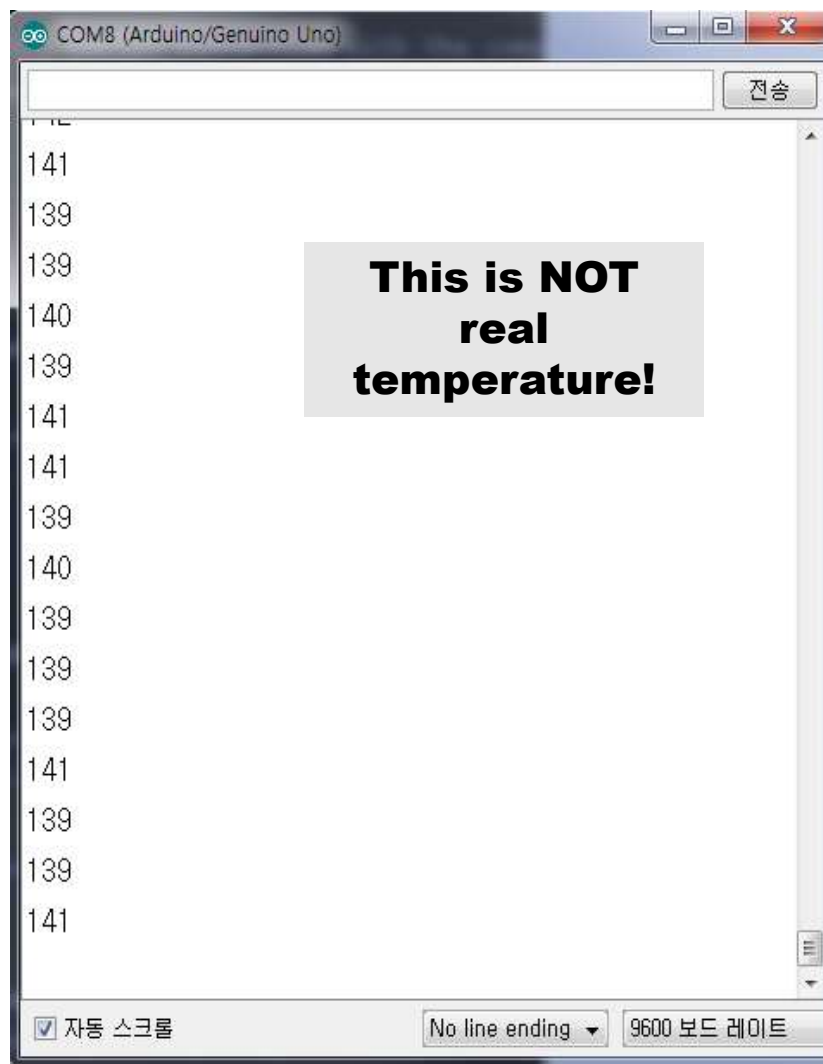


## Simple code

```

TMP36 $
1 //
2 //  AR00, TMP36 sensor
3 //
4
5 #define TEMP_INPUT 0
6 // or  int TEMP_INPUT = 0;
7
8 void setup() {
9   Serial.begin(9600);
10 }
11
12 void loop() {
13
14   int value = analogRead(TEMP_INPUT);
15   Serial.println(value);
16
17   delay(1000);
18 }
    
```

## Serial output (0 ~ 1023)



## Sensor property

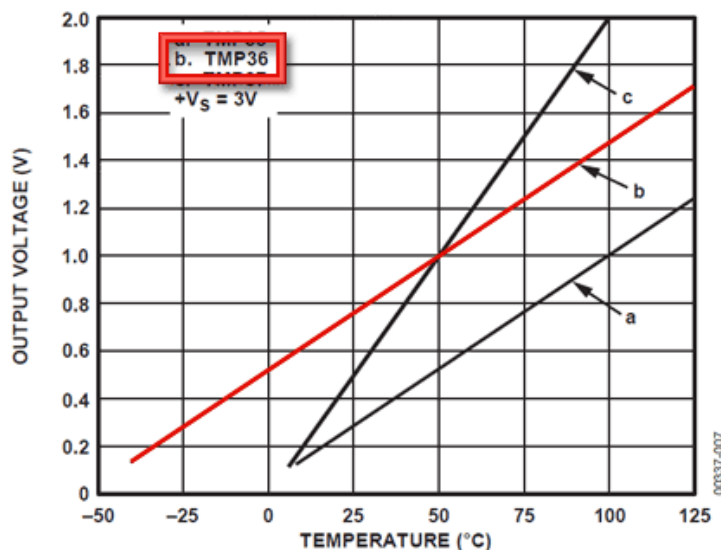


Figure 6. Output Voltage vs. Temperature

### Output Voltage (mV) vs. Temperature (°C)

V	0	500	1000
T	-50	0	50

[https://github.com/Redwoods/Arduino/blob/master/ar-iot/py-ml/tmp36\\_LR.ipynb](https://github.com/Redwoods/Arduino/blob/master/ar-iot/py-ml/tmp36_LR.ipynb)

## Temperature conversion

$$\text{Temp (}^{\circ}\text{C)} = (\text{Vout} - 500) / 10$$

$$\text{Vout (mV)} = \text{value} * (5000 / 1023)$$

$$(0 \leq \text{value} \leq 1023)$$



```
// converting that reading to voltage
float voltage = value * 5.0 * 1000; // in mV
voltage /= 1023.0;
float temperatureC = (voltage - 500) / 10 ;
```



## A3.1.4 Temperature sensor [ TMP36 ]

### Working code

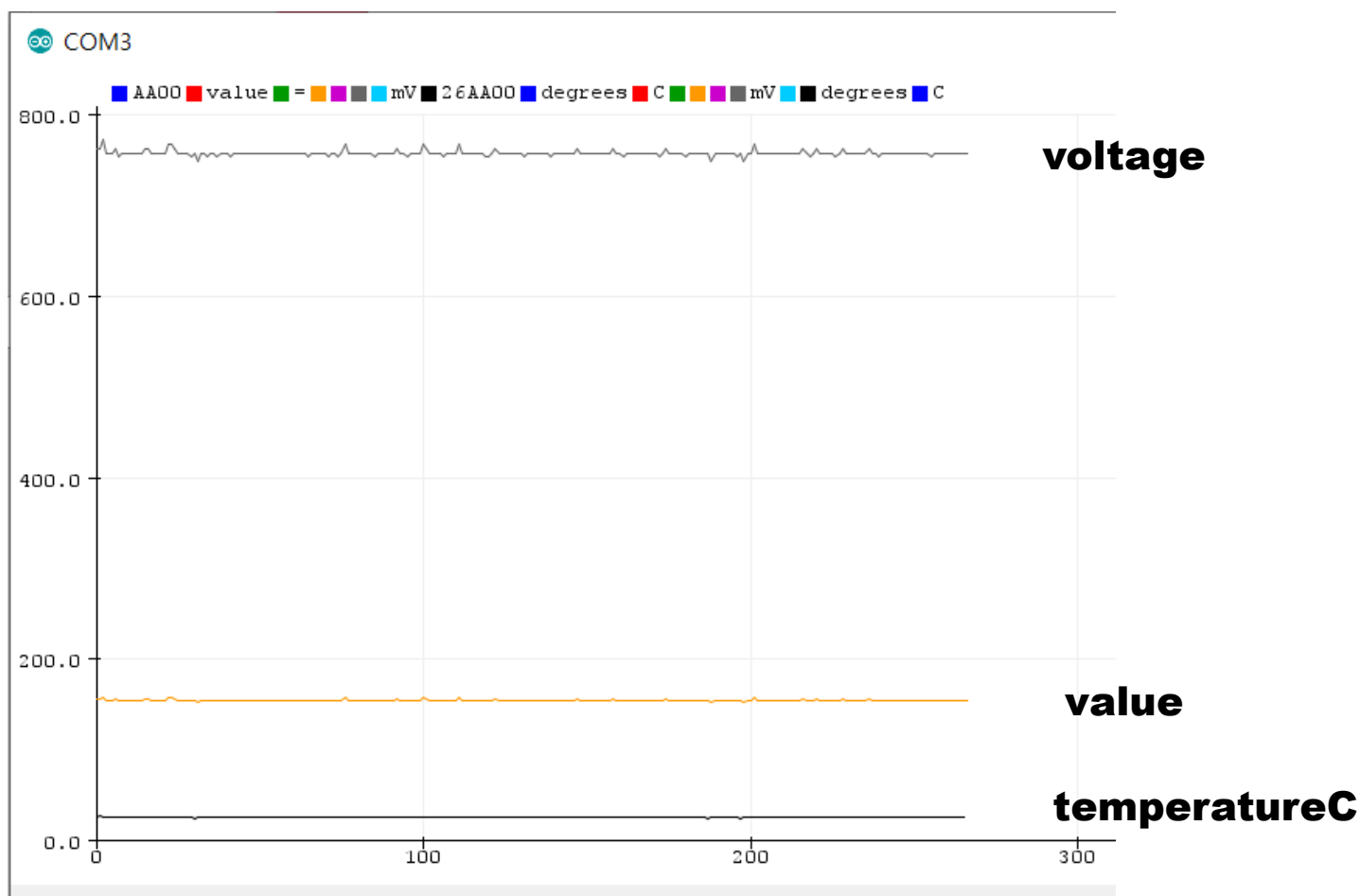
```
TMP36
10 }
11
12 void loop() {
13     //getting the voltage reading from the temperature sensor
14     int value = analogRead(TEMP_INPUT);
15     Serial.print("AA00, value = ");
16     Serial.print(value);
17     Serial.print(" : ");
18
19     // converting that reading to voltage
20     float voltage = value * 5.0 * 1000; // in mV
21     voltage /= 1023.0;
22
23     // print out the voltage
24     Serial.print(voltage);
25     Serial.print(" mV, ");
26
27     // now print out the temperature
28     float temperatureC = (voltage - 500) / 10 ;
29     Serial.print(temperatureC);
30     Serial.println(" degrees C");
31
32     delay(1000);
33 }
```

### Serial output ( °C)

```
COM4
AA00, value = 131 : 640.27 mV, 14.03 degrees C
AA00, value = 130 : 635.39 mV, 13.54 degrees C
AA00, value = 132 : 645.16 mV, 14.52 degrees C
AA00, value = 128 : 625.61 mV, 12.56 degrees C
AA00, value = 129 : 630.50 mV, 13.05 degrees C
AA00, value = 128 : 625.61 mV, 12.56 degrees C
AA00, value = 128 : 625.61 mV, 12.56 degrees C
AA00, value = 128 : 625.61 mV, 12.56 degrees C
AA00, value = 128 : 625.61 mV, 12.56 degrees C
AA00, value = 128 : 625.61 mV, 12.56 degrees C
AA00, value = 130 : 635.39 mV, 13.54 degrees C
AA00, value = 128 : 625.61 mV, 12.56 degrees C
AA00, value = 128 : 625.61 mV, 12.56 degrees C
AA00, value = 132 : 645.16 mV, 14.52 degrees C
AA00, value = 129 : 630.50 mV, 13.05 degrees C
AA00, value = 132 : 645.16 mV, 14.52 degrees C
AA00, value = 129 : 630.50 mV, 13.05 degrees C
AA00, value = 130 : 635.39 mV, 13.54 degrees C
AA00, value = 128 : 625.61 mV, 12.56 degrees C
AA00, value = 128 : 625.61 mV, 12.56 degrees C
```



# A3.1.5 Temperature sensor [ TMP36]



Save as  
ARnn\_TMP36.png



# [Practice]

## ◆ [wk11]

- **Arduino : Analog input II**
- **Complete your project**
- **Submit folder : ARnn\_Rpt10**

# wk11 : Practice-10 : ARnn\_Rpt10

## ◆ [Target of this week]

- Complete your works
- Save your outcomes
- Upload all in github.

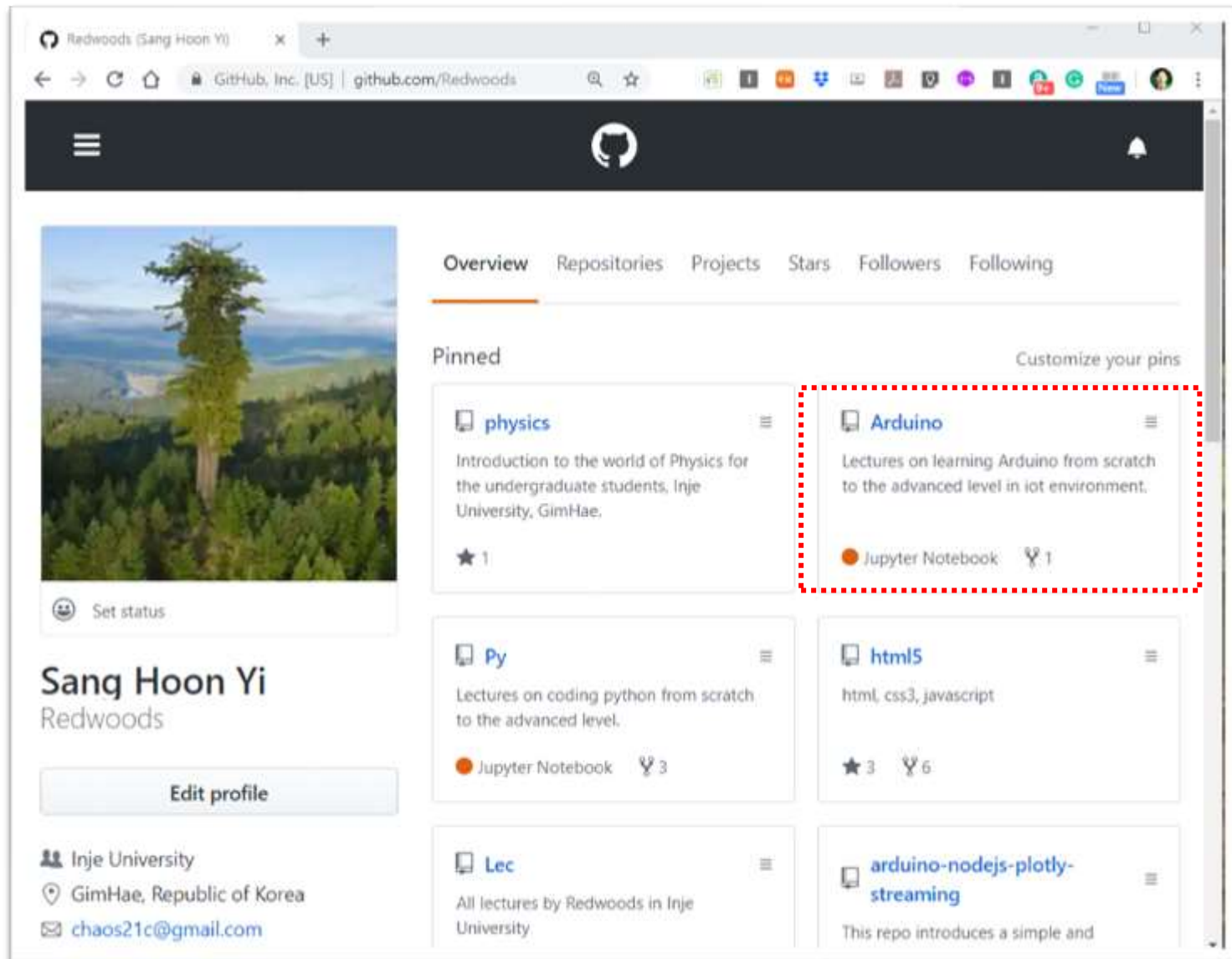
제출폴더명 : **ARnn\_Rpt10**

### 제출할 파일들

- ① **ARnn\_waterlevel.png**
- ② **ARnn\_waterlevel.ino**
- ③ **ARnn\_joystick.ino**
- ④ **ARnn\_joystick.png**
- ⑤ **ARnn\_sound\_bar.png**
- ⑥ **ARnn\_sound\_monitor.png**
- ⑦ **Arnn\_TMP36.png**
- ⑧ **\*.ino**

## ● References & good sites

- ✓ <http://www.nodejs.org/ko> Node.js
- ✓ <http://www.arduino.cc> Arduino Homepage
- ✓ <http://www.w3schools.com> By w3schools
- ✓ <http://www.github.com> GitHub
- ✓ <http://www.google.com> Googling



Redwoods (Sang Hoon Yi)

GitHub, Inc. [US] | github.com/Redwoods

Overview Repositories Projects Stars Followers Following

Pinned Customize your pins

**physics**  
Introduction to the world of Physics for the undergraduate students, Inje University, GimHae.  
★ 1

**Arduino**  
Lectures on learning Arduino from scratch to the advanced level in iot environment.  
Jupyter Notebook 🍴 1

**Py**  
Lectures on coding python from scratch to the advanced level.  
Jupyter Notebook 🍴 3

**html5**  
html, css3, javascript  
★ 3 🍴 6

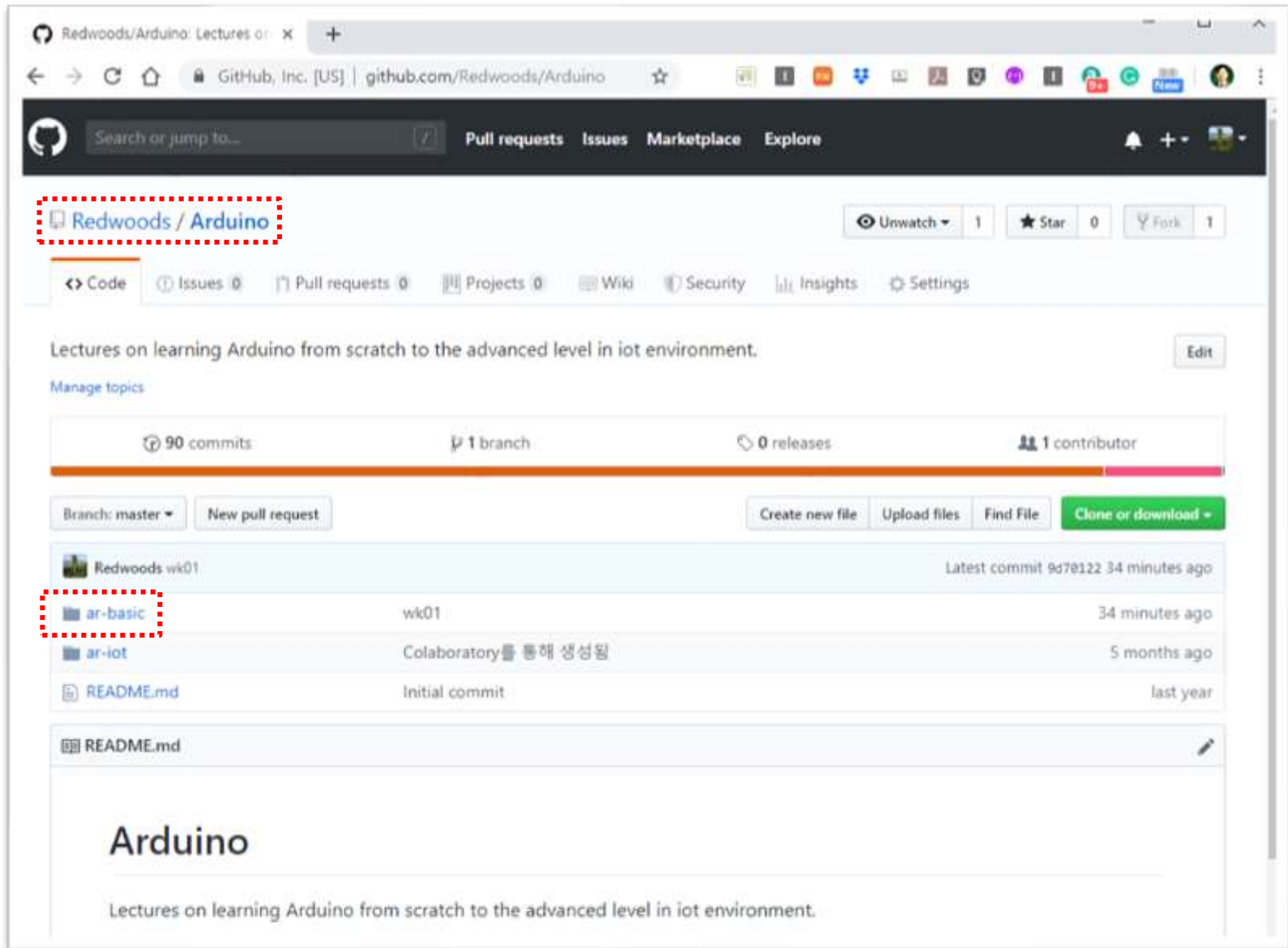
**Lec**  
All lectures by Redwoods in Inje University.

**arduino-nodejs-plotly-streaming**  
This repo introduces a simple and

**Sang Hoon Yi**  
Redwoods  
Edit profile

Inje University  
GimHae, Republic of Korea  
chaos21c@gmail.com





The screenshot shows the GitHub repository page for **Redwoods/Arduino**. The repository description is "Lectures on learning Arduino from scratch to the advanced level in iot environment." The repository has 90 commits, 1 branch, 0 releases, and 1 contributor. The file list includes **ar-basic** (34 minutes ago), **ar-iot** (5 months ago), and **README.md** (last year). The **ar-basic** file is highlighted with a red dashed box. The repository is currently on the **master** branch.

Redwoods/Arduino

Unwatch 1 Star 0 Fork 1

Code Issues 0 Pull requests 0 Projects 0 Wiki Security Insights Settings

Lectures on learning Arduino from scratch to the advanced level in iot environment. Edit

Manage topics

90 commits 1 branch 0 releases 1 contributor

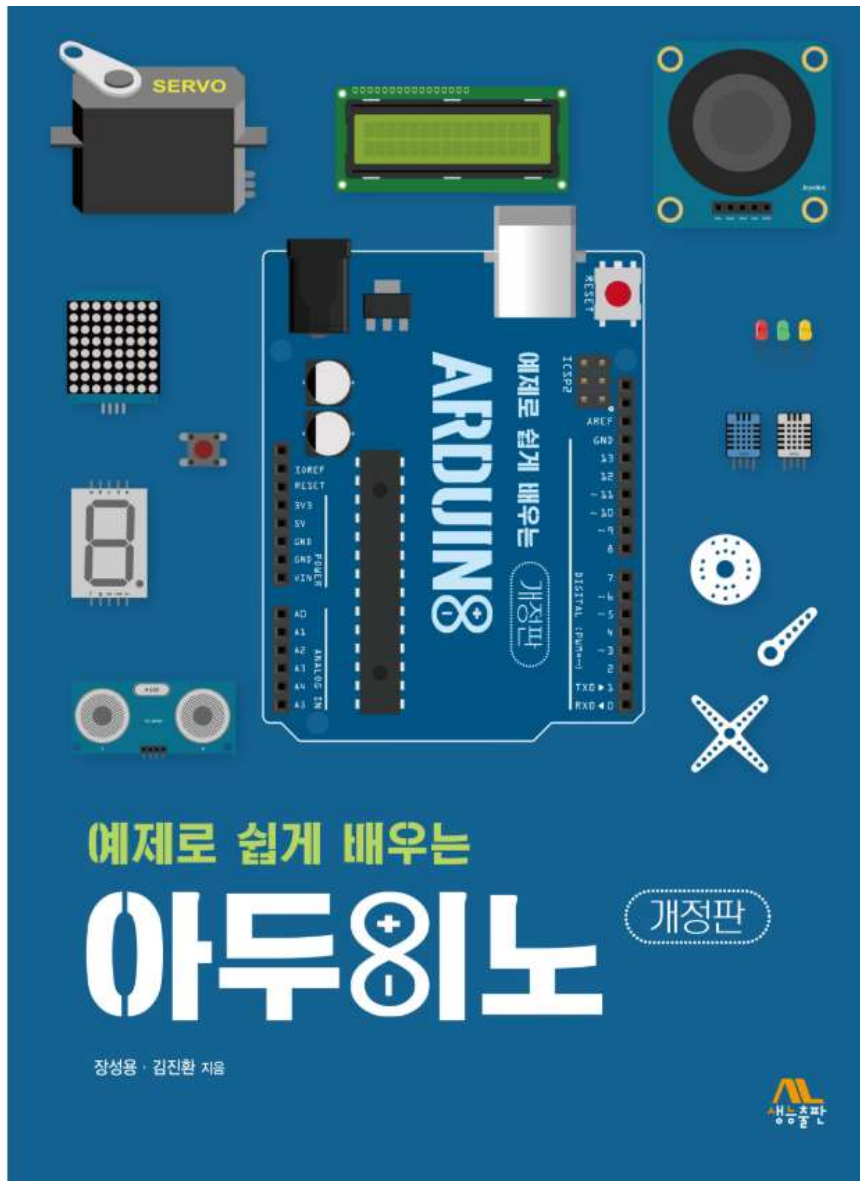
Branch: master New pull request Create new file Upload files Find File Clone or download

File	Commit	Time
Redwoods wk01	Latest commit 9d7e122	34 minutes ago
ar-basic	wk01	34 minutes ago
ar-iot	Colaboratory를 통해 생성됨	5 months ago
README.md	Initial commit	last year

README.md

## Arduino

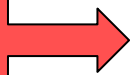
Lectures on learning Arduino from scratch to the advanced level in iot environment.



# 아두이노 키트(Kit)



web



<https://www.devicemart.co.kr/goods/view?no=12170416>

## 아두이노 레벨업키트(골드) 구성품

						
아두이노 UNO	USB 케이블	830핀 브레드보드	미니 브레드보드	점퍼와이어 세트	듀폰케이블 M/F	듀폰케이블 M/M
						
저항 220Ω	저항 1KΩ	저항 10KΩ	가변저항 10KΩ	빨강 LED	녹색 LED	파랑 LED
						
노랑 LED	RGB LED (CA)	RGB LED 모듈	1digit FND (CA)	4digit FND (CA)	8x8 도트 매트릭스	택트 스위치
						
택트 스위치 캡	볼 스위치	리드 스위치 센서	4x4키 매트릭스	5V 릴레이 모듈	조이스틱 모듈	수위 센서
						
온도센서 LM35	터미스터	습도센서	CDS 조도센서	불꽃감지센서	적외선 수신기	IR 리모컨

# 아두이노 키트(Kit) : Part-2

						
TCRT5000 적외선 센서	인체감지센서 모듈	사운드센서	능동부저	수동부저	초음파센서	I2C 1602 LCD 모듈
						
서보모터	스텝모터	스텝모터드라이버	RFID 수신 모듈	RFID 카드	RFID 태그	DS1302 RTC 모듈
						
1N4001 다이오드	2N2222 트랜지스터	74HC595	1x40 핀헤더	9V 배터리 스냅	아크릴 고정판	

■ 아두이노 UNO × 1	■ USB 케이블 × 1	■ 830핀브레드보드 × 1	■ 미니 브레드보드 × 1	■ 점퍼와이어세트 × 1
■ 듀폰케이블 × 80 (M/F,M/M)	■ 저항 × 30	■ 가변저항 × 1	■ LED × 20	■ RGB LED × 1
■ RGB LED 모듈 × 1	■ 1digit FND(CA) × 1	■ 4digit FND(CA) × 1	■ 8×8도트 매트릭스 × 1	■ 탭스위치 × 5
■ 탭스위치 캡 × 5	■ 볼스위치 × 1	■ 리드 스위치 센서 × 1	■ 4×4 키 매트릭스 × 1	■ 5V 릴레이 모듈 × 1
■ 조이스틱 모듈 × 1	■ 수위 센서 × 1	■ 온도센서 LM35 × 1	■ 써미스터 × 1	■ 온습도센서 × 1
■ CdS 조도센서 × 1	■ 불꽃감지센서 × 1	■ 적외선 수신기 × 1	■ IR 리모컨 × 1	■ TCRT5000 적외선 센서 × 1
■ 인체감지센서 모듈 × 1	■ 사운드센서 × 1	■ 능동부저 × 1	■ 수동부저 × 1	■ 초음파센서 × 1
■ I2C 1602 LCD 모듈 × 1	■ 서보모터 × 1	■ 스텝모터 × 1	■ 스텝모터드라이버 × 1	■ RFID 수신 모듈 × 1
■ RFID 카드 × 1	■ RFID 태그 × 1	■ DS1302 RTC 모듈 × 1	■ 1N4001 다이오드 × 1	■ 2N2222 트랜지스터 × 1
■ 74HC595 × 1	■ 1x40 핀헤더 × 1	■ 9V 배터리 스냅 × 1	■ 아크릴 고정판 × 1	