







Arduino-IOT [wk05]

Arduino Sensors

Visualization of Signals using Arduino, Node.js & storing signals in MongoDB & mining data using Python

Drone-IoT-Comsi, INJE University

2nd semester, 2020

Email: chaos21c@gmail.com

No DE ARDUINO

My ID

1분반-목요일 (2학년)

- AA1-01: 강서현
- AA1-02: 강태민
- AA1-03: 김세은
- AA1-04: 여수민
- AA1-05: 정영훈
- AA1-06: 차혁준
- AA1-07: 하태헌
- AA1-08: 김경욱
- AA1-09: 김민욱
- AA1-10: 김민성

- AA1-11: 김민준
- AA1-12: 김인수
- AA1-13: 김현식
- AA1-14: 장성운
- AA1-15: 전승진
- AA1-16: 정희철
- AA1-17: 조동현
- AA1-18: 전동빈
- AA1-19: 신종원

2분반-수요일 (3학년)

- AA2-01: 강민수
- AA2-11: 이정문
- AA2-02: 구병준
- AA2-12: 이주원
- AA2-03: 김종민
- AA2-13: 정재영
- AA2-04: 박성철
- AA2-14: 하태성
- AA2-05: 이승현
- AA2-15: 김경미
- AA2-06: 이창호
- AA2-16: 김규년
- AA2-07: 손성빈
- AA2-17: 김유빈
- AA2-08: 안예찬
- AA2-18: 송다은
- AA2-09: 유종인
- AA2-19: 정주은
- AA2-10: 이석민
- AA2-20: 권준표





[Practice]

- ◆ [wk04]
- > Arduino basic circuits
- Complete your project
- Upload folder: aax-nn-rpt04
- Use repo "aax-nn" in github

wk04: Practice-04: AAnn_Rpt04



- [Target of this week]
 - Complete your works
 - Save your outcomes and upload 3 figures in github

Upload folder: aax-nn_rpt04

- 제출할 파일들

- ① AAnn_Monitoring.png
- 2 AAnn_multi_Monitoring.png
- 3 AAnn_multi_Signals.png
- 4 AAnn_AnalogVoltage.png
- 5 All *.ino





Ardulho



<u> https://www.arduino.cc/</u>



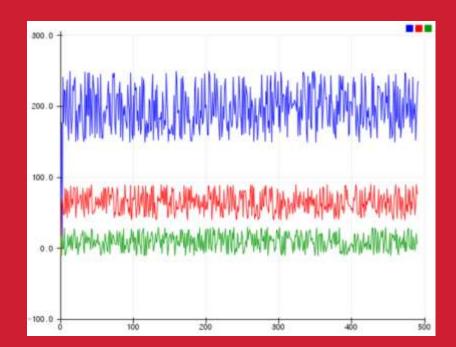
Arduino

circuits



[DIY] Multi-signals

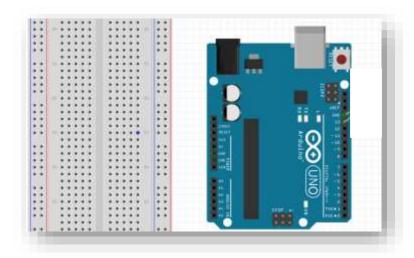
다중신호 시뮬레이션 및 모니터링







DIY - 스케치



아두이노에서 LED와 저항을 모두 제거하고 USB만 컴퓨터와 연결한다.

전자 소자 연결 없이 마구잡이 수 생성 함수를 이용해서 조도, 습도, 온도에 해당되는 3개의 신호를 만든다.

온도는 값의 범위를 -10 ~ 30, 습도는 40 ~ 90, 그리고 조도는 150 ~ 250 으로 가상적 으로 설정한다.

직렬통신 모니터링을 이용해서 세 개의 신호의 변화를 모니터링 하는 코드를 만들어 결과를 확인한다.

▶ 스케치 구성

- 1.3 개의 신호를 담을 변수를 초기화한다.
- 2. setup()에서 직렬 통신 속도를 9600 bps 로 설정하고 컴퓨터와 연결한다.
- 3. loop()에서 마구잡이 수를 세 개 발생시켜서 직렬 통신으로 3 개의 pwm 값을 각각 컴퓨터로 전송한다.





DIY - code

```
sketch05_multi_signals

1 /*

2 Multi Signals

3 Simulation of multiple random signals

4 */

5 // signals

6 int humi=0;

7 int temp=0;

8 int lux=0;
```

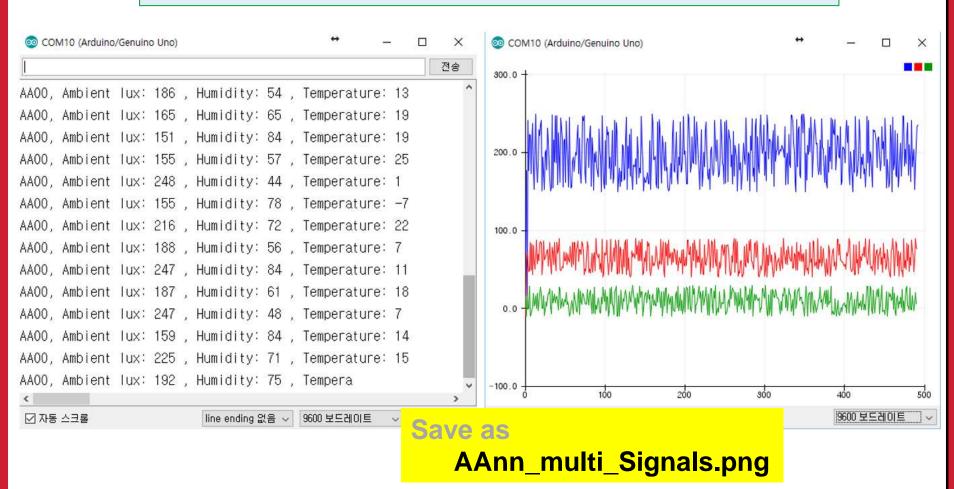
```
10 // the setup routine runs once when you press reset:
11 void setup() {
    // initialize serial communication at 9600 bits per second:
13
    Serial begin (9600);
14 }
15
16 // the loop routine runs over and over again forever:
17 void loop() {
18 // Multi signals
19 humi = random(40,90);
20 temp = random(-10, 30);
21 lux = random(150,250);
22 Serial.print("AAOO, Ambient lux: ");
    Serial.print(lux);
    Serial.print(" , Humidity: ");
    Serial.print(humi);
    Serial.print(" , Temperature: ");
    Serial println(temp);
    delay(500); // delay in between reads for stability
29 }
```



DIY - result

DIY 결과

가상적인 세 개의 센서신호 시뮬레이션:조도(위), 습도(중간), 온도(아래).



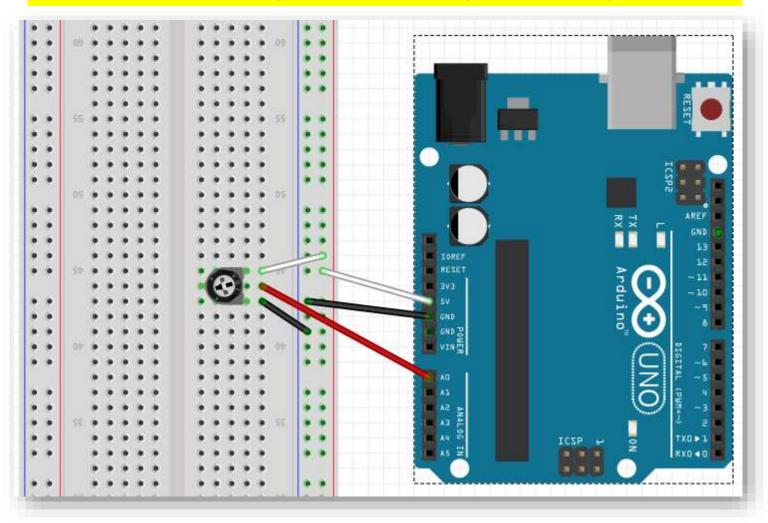


Analog Signal



A2.5.1 AnalogReadSerial (circuit)

Standard potentiometer (가변 저항기)





A2.5.2 AnalogReadSerial (code)

▶ 스케치 구성 (코드 4-1)

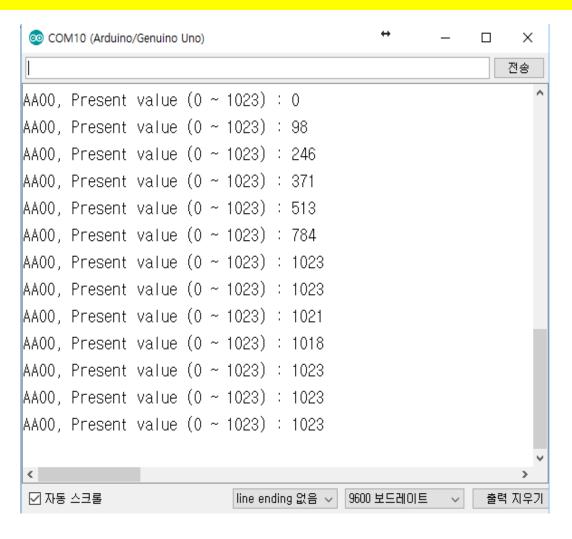
- 1. setup()에서 직렬 통신 속도를 9600 bps 로 설정하고 컴퓨터와 연결한다.
- 2. loop()에서 analogRead() 함수로 A0 핀에서 측정되는 값을 읽어 들인다.
- 3. 직렬 통신으로 A0 측정값을 한 줄로 0.5 초 마다 컴퓨터로 전송한다.
- ▶ 아두이노 코드 : sketch06_analog_read.ino

```
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  Serial.print("AA00, Present value (0 ~ 1023):");
  Serial.println(sensorValue);
  delay(500);  // 2 Hz sampling
}
```



A2.5.3 ReadAnalogValue

Serial monitor: 0 < value < 1023







A2.5.4 Analog value to Resistance or Voltage

아날로그 값을 저항 및 전압으로 변환

▶ 저항 또는 전압 환산

- 1. 저항 = 10.0 * A0 / 1023 (kΩ)
- 2. 전압 = 5.0 * A0 / 1023 (V)

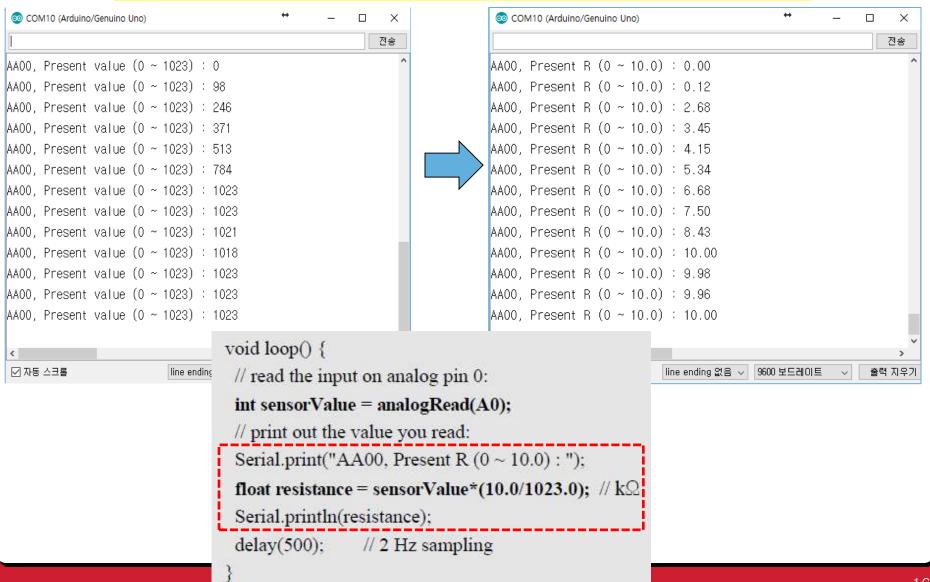
A0: 아날로그 핀 A0에서의 측정값 (0~1023)





A2.5.5 Analog value to Resistance

Serial monitor : Resistance ($0 < R < 10 \text{ k}\Omega$)

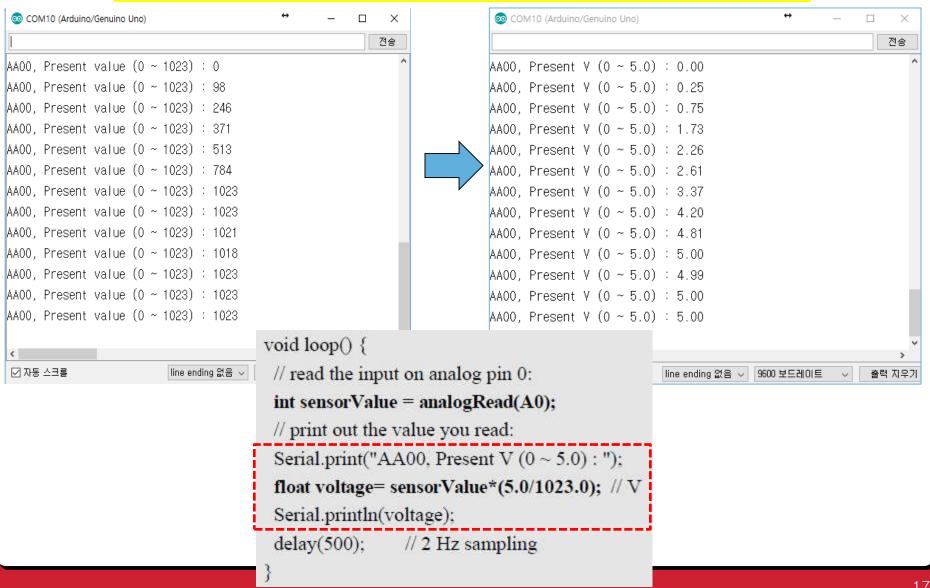






A2.5.6 Analog value to Voltage

Serial monitor : Voltage (0 < V < 5 V)

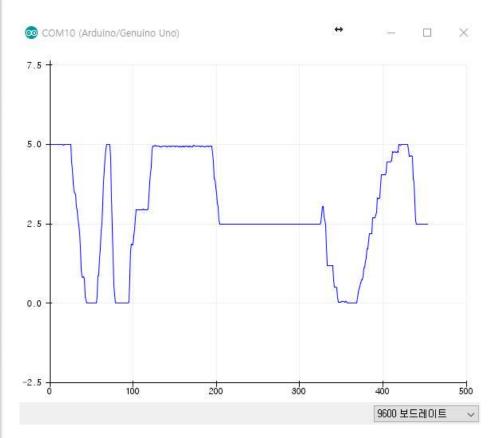




A2.5.7 ReadAnalogVoltage

Result

```
COM4
4A00, Present voltage (0.0 ~ 5.0) : 5.00
4A00, Present voltage (0.0 ~ 5.0): 3.68
4A00, Present voltage (0.0 ~ 5.0): 2.42
4A00, Present voltage (0.0 ~ 5.0) : 1.37
AA00, Present voltage (0.0 ~ 5.0): 0.00
4A00, Present voltage (0.0 ~ 5.0) : 0.00
4A00, Present voltage (0.0 ~ 5.0): 0.00
AA00, Present voltage (0.0 ~ 5.0) : 0.88
4A00, Present voltage (0.0 ~ 5.0) : 1.47
AA00, Present voltage (0.0 ~ 5.0) : 2.11
4A00, Present voltage (0.0 ~ 5.0): 2.79
4A00, Present voltage (0.0 ~ 5.0) : 3.38
4A00, Present voltage (0.0 ~ 5.0) : 3.99
AA00, Present voltage (0.0 ~ 5.0): 4.91
4A00, Present voltage (0.0 ~ 5.0) : 5.00
4A00, Present voltage (0.0 ~ 5.0): 5.00
4A00, Present voltage (0.0 - 5.0) : 4.68
4A00, Present voltage (0.0 ~ 5.0) : 3.88
4A00, Present voltage (0.0 ~ 5.0) : 3.35
```



Save as

AAnn_AnalogVoltage.png

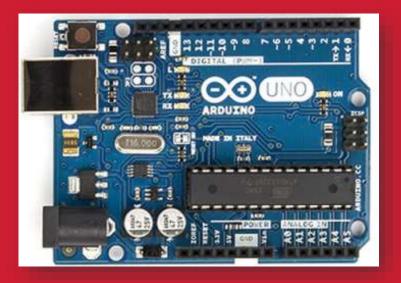


A2.5.8 ReadAnalogVoltage using f_map()

Hint code : f_map() instead of map()

```
AAnn_AnalogRead_fmap §
977 the setup routine runs once when you press reset:
10 void setup() {
    // initialize serial communication at 9600 bits per second:
    Serial.begin(9600);
13|}
14
15 // the loop routine runs over and over again forever:
16 void loop() {
    // read the input on analog pin 0:
   int sensorValue = analogRead(A0);
   //float voltage = map(sensorValue, 0, 1023, 0.0, 5.0); // map 0~1023 to 0~5
20/// float voltage = sensorValue*(5.0/1023.0);
   !float voltage = f_map(sensorValue, 0, 1023, 0.0, 5.0); // map 0~1023 to 0~5
    // print out the value you read:
    Serial.print("AA00, Present voltage (0.0 ~ 5.0) : ");
    Serial.println(voltage);
241
    delay(500);
                 // delay in between reads for stability
26|}
28 float f_map(long x, long in_min, long in_max, float out_min, float out_max)
29|{
    return (x - in min) * (out max - out min) / (in max - in min) + out min;
```

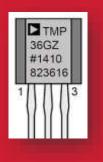


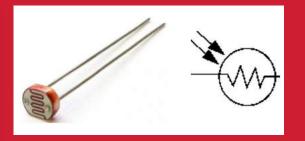


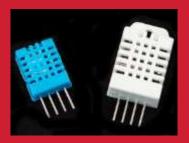
Arduino

Sensors





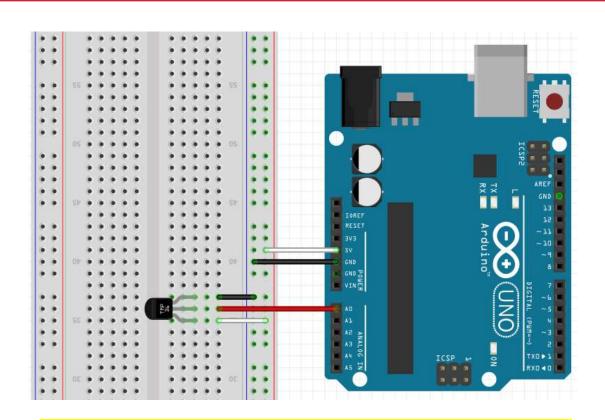


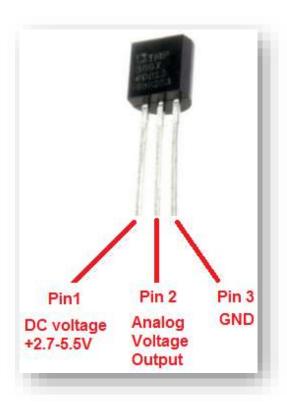




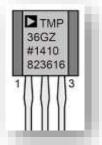


A3.1.1 Temperature sensor [TMP36]





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



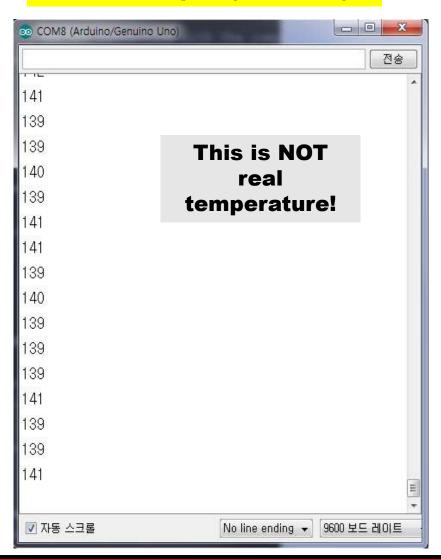


A3.1.2 Temperature sensor [TMP36]

Simple code

```
TMP36§
       AA00, TMP36 sensor
3 1 / /
5 #define TEMP_INPUT 0
6// or int TEMP_INPUT = 0;
8 void setup() {
    Serial.begin(9600);
10 }
11
12 void loop() {
13
    int value = analogRead(TEMP INPUT);
14
    Serial.println(value);
16
    delay(1000);
18 }
```

Serial output (0 ~ 1023)







A3.1.3 Temperature sensor [TMP36]

Sensor property

2.0 1.8 1.6 1.4 1.2 1.0 0.8 0.4 0.2 0.4 0.2 0.50 -50 -25 0 25 TEMPERATURE (°C)

Figure 6. Output Voltage vs. Temperature

Output Voltage (mV) vs. Temperature (°C)			
V	0	500	1000
Т	-50	0	50

https://github.com/Redwoods/Arduino/blob/ master/ar-iot/py-ml/tmp36 LR.ipynb

Temperature conversion

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



```
// 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

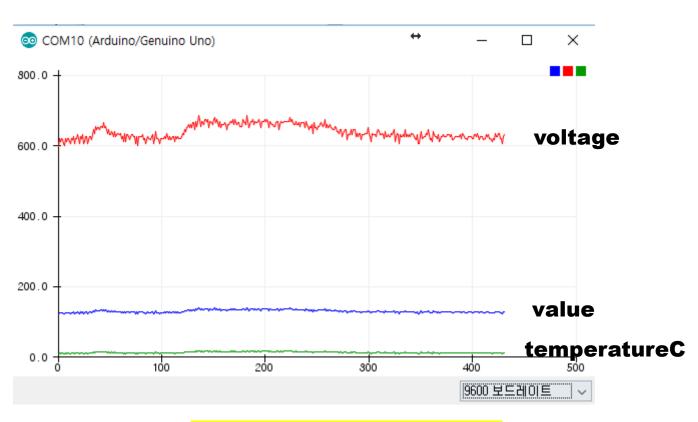
Serial output (°C)

```
TMP36
10|}
                                                                  11
12 void loop() {
                                                                  AA00, value = 131 : 640.27 mV, 14.03 degrees C
     //getting the voltage reading from the temperature sensor
                                                                  AA00, value = 130 : 635.39 mV, 13.54 degrees C
    int value = analogRead(TEMP_INPUT);
                                                                  AA00, value = 132 : 645.16 mV, 14.52 degrees C
15 Serial.print("AA00, value = ");
                                                                  AA00, value = 128 : 625.61 mV, 12.56 degrees C
16
    Serial.print(value);
                                                                  AA00, value = 129 : 630.50 mV, 13.05 degrees C
    Serial.print(" : ");
                                                                  AA00, value = 128 : 625.61 mV, 12.56 degrees C
18
                                                                  AA00, value = 128 : 625.61 mV, 12.56 degrees C
19
     // converting that reading to voltage
                                                                  AA00, value = 128 : 625.61 mV, 12.56 degrees C
20
     float voltage = value * 5.0 * 1000; // in mV
                                                                  AA00, value = 128 : 625.61 mV, 12.56 degrees C
21
     voltage /= 1023.0;
                                                                  AA00, value = 128 : 625.61 mV, 12.56 degrees C
                                                                  AA00, value = 128 : 625.61 mV, 12.56 degrees C
23
     // print out the voltage
                                                                  AA00, value = 130 : 635.39 mV, 13.54 degrees C
     Serial.print(voltage);
24
                                                                  AA00, value = 128 : 625.61 mV, 12.56 degrees C
25
     Serial.print(" mV, ");
                                                                  AA00, value = 128 : 625.61 mV, 12.56 degrees C
26
                                                                  AA00, value = 132 : 645.16 mV, 14.52 degrees C
     // now print out the temperature
27
                                                                  AA00, value = 129 : 630.50 mV, 13.05 degrees C
     float temperatureC = (voltage - 500) / 10;
28
                                                                  AAOO, value = 132 : 645.16 mV, 14.52 degrees C
     Serial.print(temperatureC);
29 i
                                                                  AA00, value = 129 : 630.50 mV, 13.05 degrees C
30
     Serial.println(" degrees C");
                                                                  AAOO. value = 130 : 635.39 mV, 13.54 degrees C
                                                                  AA00, value = 128 : 625.61 mV, 12.56 degrees C
    delay(1000);
                                                                  AA00, value = 128 : 625.61 mV, 12.56 degrees C
33 }
```



A3.1.5 Temperature sensor [TMP36]

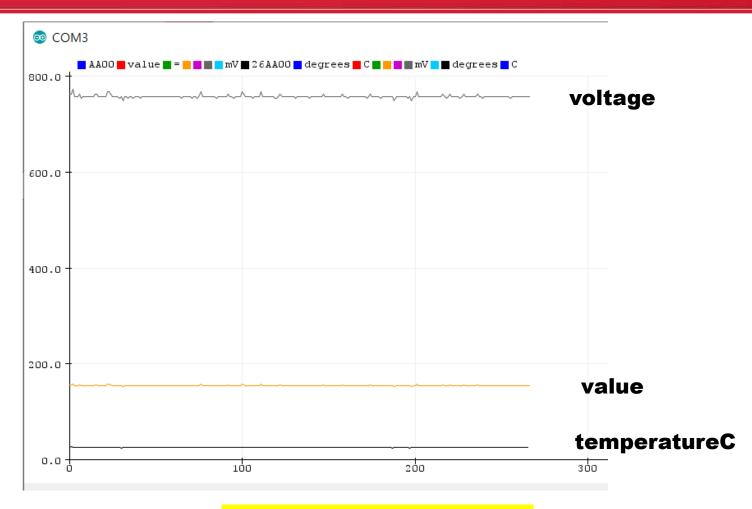




Save as AAnn_TMP36.png



A3.1.5 Temperature sensor [TMP36]



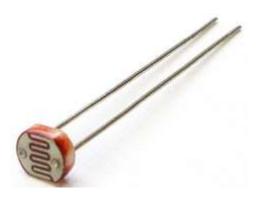
Save as AAnn_TMP36.png

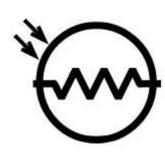




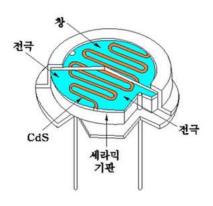
A3.2 Luminosity sensor [Photocell LDR]

CdS 센서- photoresistor





CDS특성



- 1. 감도
 - -빛의 파장에 따라 감도가 다름
- 2. 허용손실
 - -비교적 큰 전류를 흘릴 수 있음
- 3. 암 전류
 - -빛이 없어도 약간의 전류가 흐름
- - 빛을 비추면 흐르는 전류
- 응답 시간 지연 <mark>빛의 세기</mark>에 따라 응답시간 다름
- 6. 가변저항
 - -빛에 따른 가변저항





A3.2.1 Luminosity sensor [Photocell LDR]

CdS 센서 - photoresistor





- ✓ CdS 분말을 세라믹 기판 위에 압축하여 제작
- ✓ 빛이 강할 수록 저항 값이 감소
- ✓ ADC를 이용하여 변화된 저항에 전압을 인가하여전압의 변화를 감지
- ✓ 자동 조명장치, 조도 측정 등에 사용

럭스

🚅 다른 뜻에 대해서는 Lux 문서를 참조하십시오

럭스(lux, 기호 1x)는 빛의 조명도를 나타내는 SI 단위이다. 럭스는 루멘에서 유도 $1 | x = 1 | m/m^2 = 1 \text{ cd·sr·m}^{-2}$

럭스의 예 [편집]

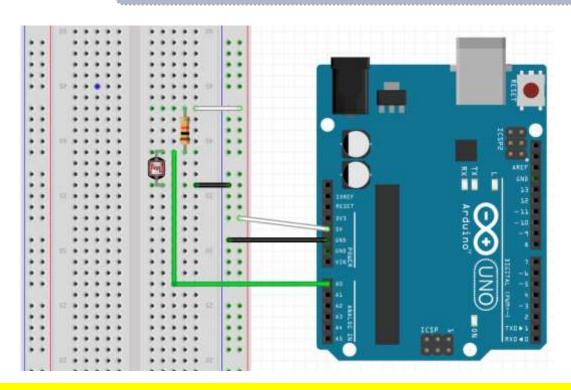
I밝기차	예
10 ⁻⁵ lux	가장 밝은 별(시리우스)의 빛 ^[1]
10 ⁻⁴ lux	하늘을 덮은 완전한 별빛[1]
0.002 lux	<mark>대기광</mark> 이 있는 달 없는 맑은 밤 하늘 ^[1]
0.01 lux	초승달
0.27 lux	맑은 밤의 보름달 ^{[1][2]}
1 lux	열대 위도를 덮은 보름달 ^[3]
3.4 lux	맑은 하늘 아래의 어두운 황혼 ^[4]
50 lux	거실 ^[5]
80 lux	복도/화장실 ^[6]
100 lux	매우 어두운 낮 ^[1]
320 lux	권장 오피스 조명 (오스트레일리아) ^[7]
400 lux	맑은 날의 해돋이 또는 해넘이
1000 lux	인공 조명 $^{[1]}$; 일반적인 TV 스튜디오 조명
10,000-25,000 lux	낮 (직사광선이 없을 때) ^[1]
32,000–130,000 lux	직사광선

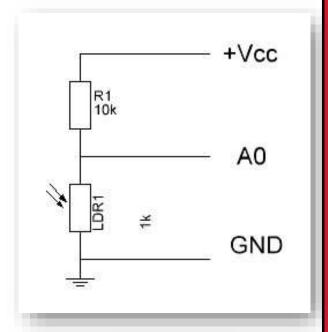




A3.2.2 Luminosity sensor [Photocell LDR]

CdS 센서 회로





Parts: 20 mm photocell LDR, R (10 k Ω X 1)

광센서에서의 전압 강하 값을 A0로 측정







A3.2.3 Luminosity sensor [sketch-1]

▶ 스케치 구성

- 1. A0 핀을 CdS 조도 센서의 입력으로 설정한다.
- 2. setup()에서 직렬 통신 속도를 9600 bps 로 설정하고 컴퓨터와 연결한다.
- 3. loop()에서 analogRead() 함수로 A0 핀에서 측정되는 값을 읽어 들인다.

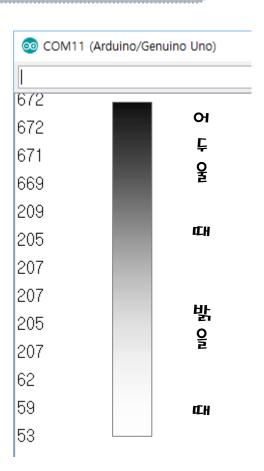




A3.2.4 Luminosity sensor [Photocell LDR]

CdS 센서 회로 - 측정 1.

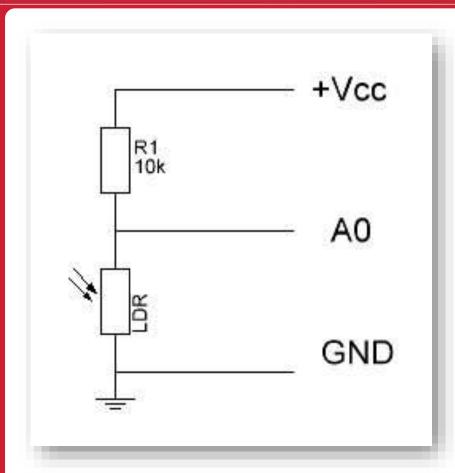
```
CdS_start
 1 #define CDS_INPUT 0
 3 void setup() {
     Serial.begin(9600);
 5 }
 7 void loop() {
 8
     int value = analogRead(CDS_INPUT);
10
    Serial.println(value);
11
    delay(1000);
12
13 }
14
```



어두우면 측정 값이 커지고 밝을수록 값이 작아진다 ???



CdS 센서 회로 분석 (1/2)



LDR's (Light dependent resistors) have a low resistance in bright light and a high resistance in the darkness.

If you would us the LDR as the lower part of a voltage divider, then in darkness there would be a high voltage over the LDR, while in bright light, there would be a low voltage over that resistor.

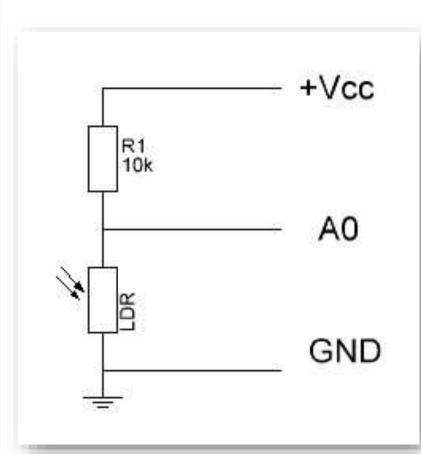
어두우면 측정 값이 작아지고 밝을수록 값이 커져야 된다. 그리고 측정 값은 lux로 표현된다.

$$V_{out} = rac{R_{ldr}}{R_1 + R_{ldr}} * V_{cc}$$

A0에서 측정되는 **LDR** 양단의 전압 = **V**_{out}



CdS 센서 회로 분석 (2/2)



$$(a) \ V_{out} = rac{R_{ldr}}{(R_1 + R_{ldr})} * V_{CC} \; ,$$

(b)
$$R_{ldr} = \frac{10 * V_{out}}{(5 - V_{out})} (k\Omega)$$
,

(c)
$$V_{out} = value * V_{CC}/1023$$
,

$$(d) \ Lux = \frac{500}{R_{ldr}} \ ,$$

$$(e) \;\; Lux = (rac{2500}{V_{out}} - 500)/10 \; (lux).$$

$$V_{out} = \frac{R_{ldr}}{R_1 + R_{ldr}} * V_{cc}$$

A0에서 측정되는 **LDR** 양단의 전압 **= V**out





A3.2.5 Luminosity sensor [sketch-2]

▶ 스케치 구성

- 1. A0 핀을 CdS 조도 센서의 입력으로 설정한다.
- 2. setup()에서 직렬 통신 속도를 9600 bps 로 설정하고 컴퓨터와 연결한다.
- 3. loop()에서 analogRead() 함수로 A0 핀에서 측정되는 값을 읽어 들인다.
- 4. A0 측정값 (0~1023)을 전압 (0~5 V)으로 환산한다.
- 5. 전압 (V)을 온도 (°C)로 환산한 후, A0 측정값, 환산 전압, 환산 조도를 한 줄로 1 초 마다 컴퓨터로 전송한다.

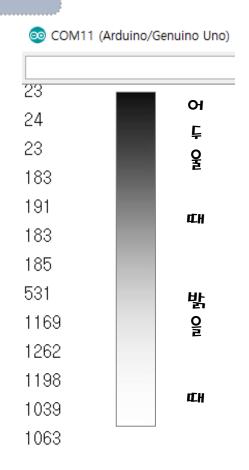




A3.2.6 Luminosity sensor [Photocell LDR]

CdS 센서 회로 - 측정 2.

```
sketch08_CdS2
 1 // lux
2 #define CDS_INPUT 0
4 void setup() {
5 Serial begin(9600);
6.}
7 void loop() {
   int value = analogRead(CDS_INPUT);
   Serial.println(int(luminosity(value)));
   delay(1000):
10
11 }
13 //Yoltage to Lux
14 double luminosity (int RawADCO){
    double Vout=RawADC0*5.0/1023; // 5/1023 (Vin = 5 V)
    double lux=(2500/Yout-500)/10;
    // lux = 500 / Rldr, Yout = Ildr*Rldr = (5/(10 + Rldr))*Rldr
    return lux;
```



밝을수록 측정 값이 커지고 어두을수록 값이 작아진다 !!!

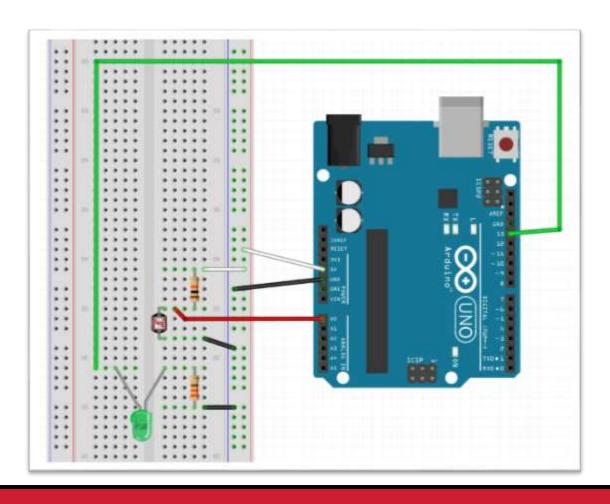


[DIY] Luminosity sensor [Photocell LDR]

DIY

조도 값에 따라 LED를 켜고 끄는 코드를 만드시오.

- 단색 LED의 anode를 D13번, cathode를 330 Ω 저항에 연결 후 GND에 연결하시오.
- 조도 값이 문턱 값 이상이면 LED를 OFF, 그렇지 않으면 ON.







[DIY] Luminosity sensor [Photocell CdS LDR]

Code

Write down your code here to complete the task that turns on LED when luminosity of ambient light becomes lower than a threshold.

조도 값이 문턱 값 이상이면 LED를 OFF, 그렇지 않으면 ON.





[DIY] Luminosity sensor [Photocell CdS LDR]

Code

```
Cds_LED
                                     13 void loop() {
                                     14 int value = analogRead(CDS_INPUT);
 1 // Tux
                                     15 int lux = int(luminosity(value))
2 #define CDS_INPUT 0
                                     16 Serial.println(lux);
3 // LED pin
 4 const int ledPin = 13;
                                        // If lux is lower than a threshold, LED is set ON.
                                     19 ! if(lux >= threshold)
                                          digitalWrite(ledPin, LOW);
 6 int threshold = 70;
                                          digitalWrite(ledPin, HIGH);
8 void setup() {
    pinMode(ledPin, OUTPUT);
                                         delay(1000);
     Serial.begin(9600);
                                     25|}
                                        //Voltage to Lux
11|}
                                     27 double luminosity (int RawADCO){
                                          double Vout=RawADC0*5.0/1023; // 5/1023 (Vin = 5 V
                                        double Tux=(2500/Yout-500)/10;
                                          // lux = 500 / Ridr, Yout = IIdr*Ridr = (5/(10 + Ridr))*Ridr
                                          return lux;
```

AAnn_CdS_LED.ino



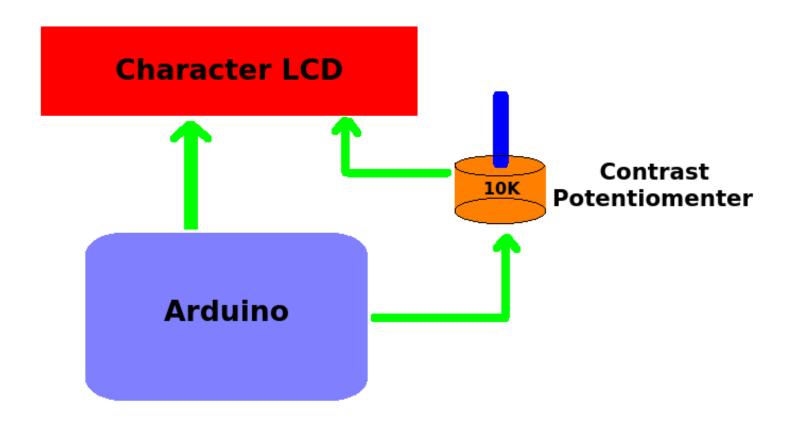
Signal Monitoring via LCD







Introduction to LCD





Liquid crystal display

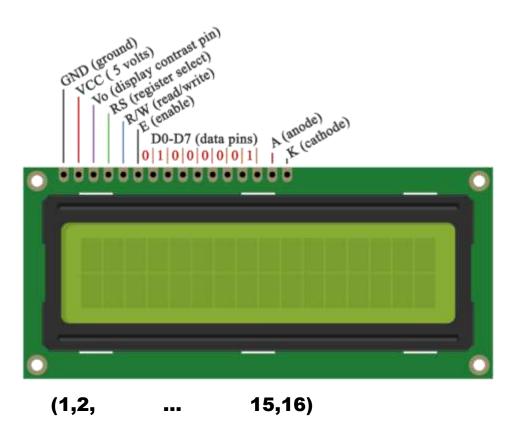
- 1 입출력 핀을 이용하여 LCD 모듈에 표시하기
- 2 12를 이용한 LCD 출력





1. Introduction to LCD

LCD (Liquid Crystal Display, 16 X 2)



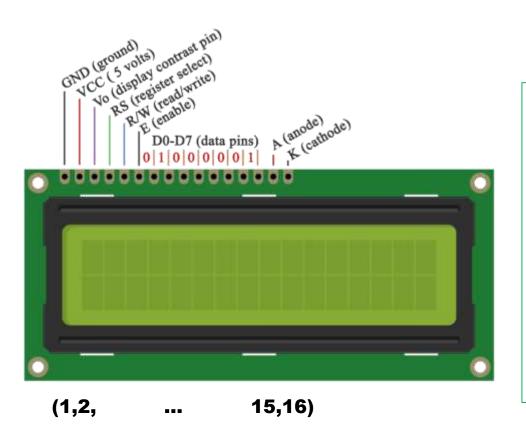
- 1. GND
- 2. VCC (+5V)
- 3. Vo (contrast, 가변저항기 연결)
- 4. RS
- 5. R/W
- 6. E
- > D0 ~ D7 (data, 7~14)
- A (15, Backlight+, 220 or 330Ω)
- K (16, Backlight-)





Introduction to LCD

LCD (Liquid Crystal Display, 16 X 2)



Pin 1 to Arduino GND Pin 2 to Arduino +5V

Pin 3 to wiper

Pin 4 to Arduino pin D12

Pin 5 to Arduino GND

Pin 6 to Arduino pin D11

Pin 11 to Arduino pin D5

Pin 12 to Arduino pin D4

Pin 13 to Arduino pin D3

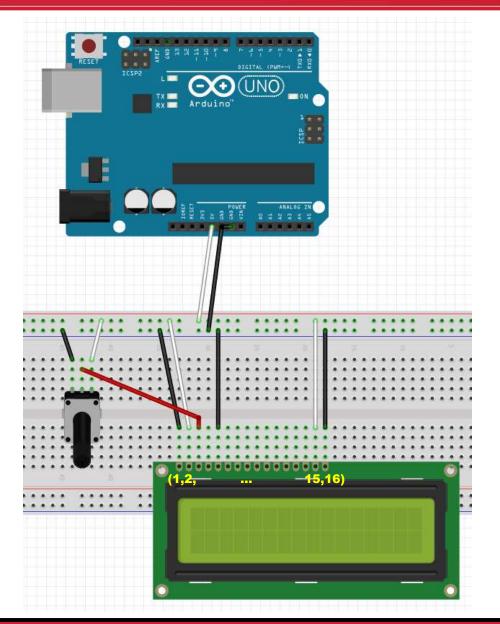
Pin 14 to Arduino pin D2

Pin 15 to +5V (with 220 or 330 Ω)

Pin 16 to GND

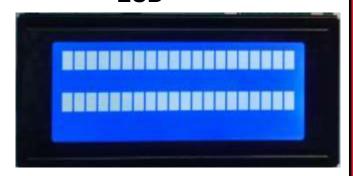


LCD 초기화 (pin-1, 2, 3, 5, 15,16)



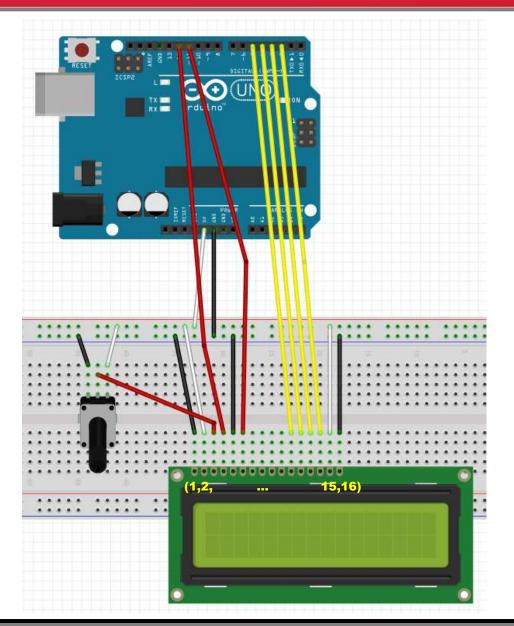
Pin 1 to Arduino GND
Pin 2 to Arduino +5V
Pin 3 to wiper (potentiometer)
Pin 5 to Arduino GND
Pin 15 to +5V
Pin 16 to GND

전**원 연결 후** LCD ^{초기화}





데이터 입력 초기화 (pin-4, 6,



Pin 1 to Arduino GND

Pin 2 to Arduino 5V

Pin 3 to wiper

Pin 4 to Arduino pin D12

Pin 5 to Arduino GND

Pin 6 to Arduino pin D11

Pin 11 to Arduino pin D5

Pin 12 to Arduino pin D4

Pin 13 to Arduino pin D3

Pin 14 to Arduino pin D2

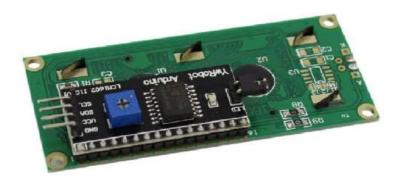
Pin 15 to +5V

Pin 16 to GND



2. I²C를 이용한 LCD 출력





I²C(^{아이스케어드시}, Inter-Integrated Circuit)는 필립스에서 개발한 직렬 버스이다. 마더보드, 임베디드 시스템, 휴대 전화 등에 저속의 주변 기기를 연결하기 위해 사용된다.

I²C 는 <u>물업 저항</u>이 연결된 직렬 데이터(SDA)와 직렬 클럭(SCL)이라는 두 개의 양 방향 <u>오픈</u> <u>컬렉터</u> 라인을 사용한다. 최대 전압은 +5 V 이며, 일반적으로 +3.3 V 시스템이 사용되지만 다른 전압도 가능하다.

https://ko.wikipedia.org/wiki/I%C2%B2C

http://www.ifuturetech.org/product/16x2-lcd-i2c-lcd/





I²C를 이용한 LCD 출력

1²C (Inter Integrated Circuit)

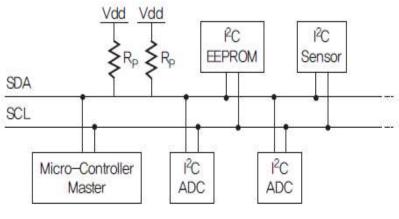


그림 3.2 fC를 이용한 네트워크

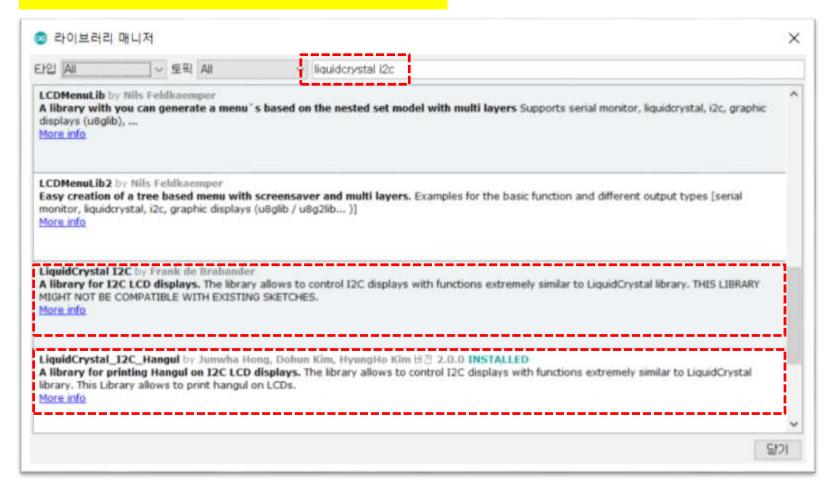
- ✓ Phillips사에서 개발된 규격이며 TWI라고도 함.
- ✓ SDA(Serial Data line), SCL(Serial Clock Line)
 - 두 선으로 통신
- ✓ Master와 Slave로 구분되어 Master에서 통신을 주관
- ✓ 최대 112개의 노드를 연결 가능하고 최고 3.4Mbps의 속도
- ✓ LCD 모듈을 I²C 통신으로 제어하기 위해선PCF8574 IC를 사용
- ✓ SDA, SCL 두 개의 입출력 핀만 필요



I²C를 이용한 LCD 출력 - 라이브러리 설

라이브러리 매니저를 이용하여 I2C LCD용 라이브러리(LiquidCrystal I2C)를 설치

스케치 > 라이브러리 포함하기 > 라이브러리 관리

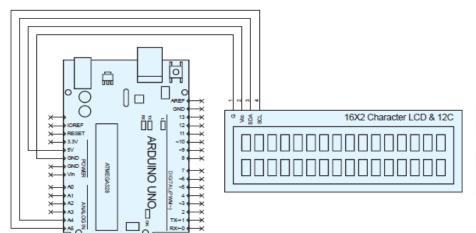




I²C를 이용한 LCD 출력 회로

Hardware

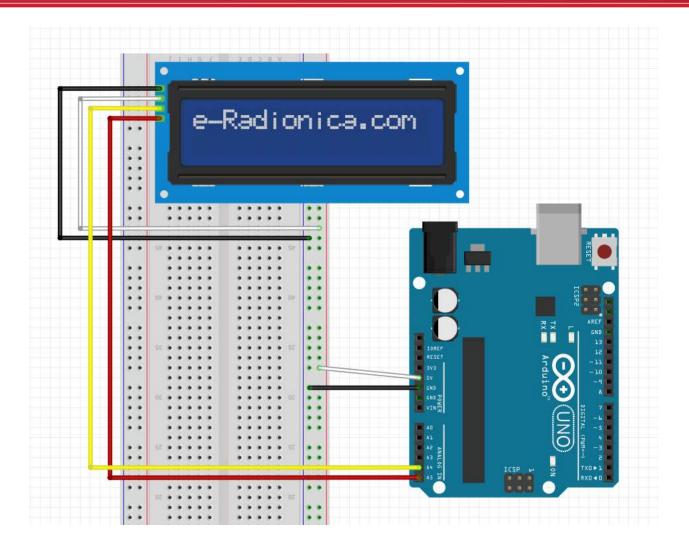
- 1. I²C LCD 모듈과 Arduino는 전원핀 Vcc, GND와 I²C 통신핀 SDA, SCL이 연결되어야 한다.
- 2. I²C LCD 모듈의 Vcc와 GND를 Arduino의 5V와 GND에 연결한다.
- 3. SDA는 A4에, SCL은 A5에 연결한다.







I²C를 이용한 LCD 출력 회로





I²C를 이용한 LCD 출력

Commands

- LiquidCrytral_I2C(I2C 주소, 가로 글자수, 세로 글자수)

 LCD 모듈이 연결된 I2C 주소와 LCD의 가로, 세로 글자수를 설정한다.
- lcd.init(); LCD 모듈을 설정한다.
- lcd.clear(): lcd란 이름의 LCD 모듈의 화면의 모든 표시를 지우고 커서를 왼쪽 위로 옮긴다.
- lcd.home(): lcd란 이름의 LCD 모듈의 커서를 왼쪽 위로 옮긴다.
- lcd.setCursor(행, 열): lcd란 이름의 LCD 모듈의 커서를 원하는 위치로 이동시킨다.
- lcd.print(데이터): lcd란 이름의 LCD 모듈에 데이터를 출력한다.
- lcd.noBacklight(): lcd란 이름의 LCD 모듈의 백라이트를 소등한다.
- lcd.backlight(); lcd란 이름의 LCD 모듈의 백라이트를 점등한다.



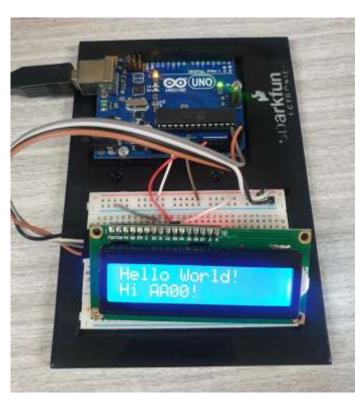
I²C를 이용한 LCD 출력

Take a photo of LCD screen.

Save photo as **AAnn_LCD_hello.png**



Save code: AAnn_LCD.ino





CdS LCD Project

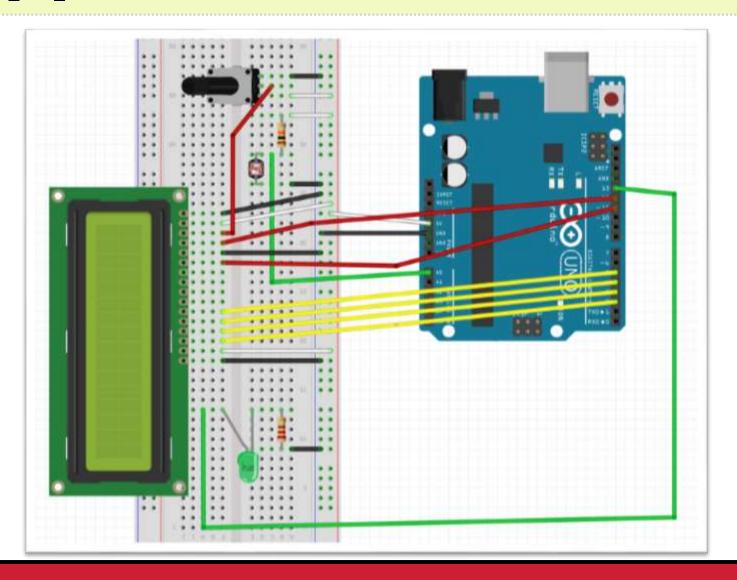
LCD에 조도 값을 표시하면서 조도에 따라 LED를 ON/OFF





CdS-LCD project: fzz circuit

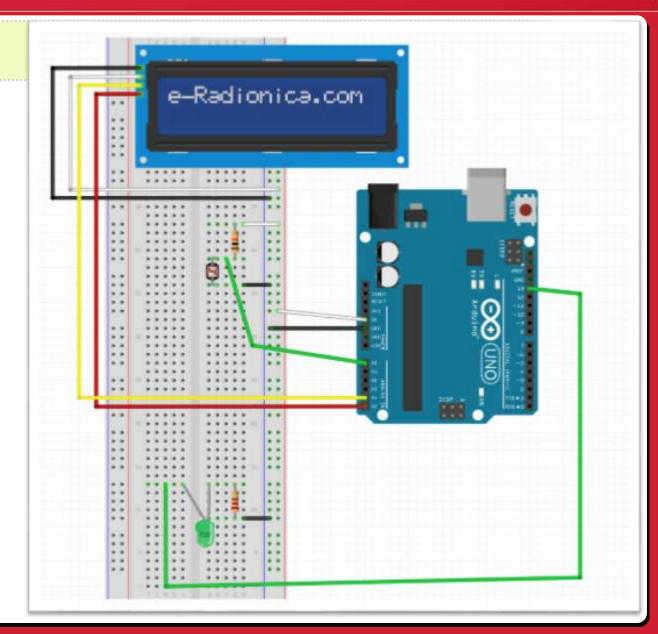
CdS_LCD_LED.fzz





CdS-I2C_LCD project: fzz circuit

CdS_I2C_LCD_LED.fzz







CdS-LCD project

Set CdS-LCD project

Project

CdS 셀을 이용하여 조도를 측정해 보자.

- 1. CdS 셀로 측정된 조도를 아날로그 핀을 통하여 0~1023 범위로 읽는다.
- 2. ADC 값을 LCD 모듈로 lux로 출력한다. (빛의 밝기)
- 3. lux 값에 따라 D13에 연결된 단색 LED의 ON/OFF를 조정한다.

Hardware

- 1. LCD를 연결한다.
- 2. $CdS셀과 10k\Omega$ 저항을 연결한 뒤 저항의 한쪽 끝은 5V에 CdS셀의 한쪽 끝은 GND에 연결한다.
- 3. 저항과 CdS셀 사이를 아날로그 입력핀 A0에 연결한다.
- 4. 단색 LED를 330 Ω 저항을 연결해서 디지털 입력핀 D13과 GND에 연결한다.





CdS-LCD project: new code

CdS 센서 LCD 회로 - code: AAnn_LCD_lux.ino

```
AAnn_LCD_lux_start §
 2 빛 입력 LCD 모니터링 및 제어
 4 // LCD 라리브러리 설정
 5 #include <LiquidCrystal 12C.h>
 6 #include<Wire.h>
 7 // LCD 설정
 8 LiquidCrystal 12C 1cd(0x27,16,2): // 0x3F
 9 // 0번 아날로그핀을 CdS 셀 입력으로 설정한다.
10 const int CdSPin = 0: // CdS => A0
11 const int ledPin = 13; // LED pin => D13
13 // LED OFF above threshold lux
15 void setup() {
16 pinMode(ledPin, OUTPUT);
17 // 16X2 LCD 모듈 설정하고 백라이트를 켠다.
18 lcd.init():
19 lcd.backlight();
20 // 모든 메세지를 삭체한 뒤
21 // 숫자를 제외한 부분들을 미리 출력시킨다.
    lcd.clear():
   lcd.setCursor(0,0):
24 | lcd.print("AAOO, ADC: "):
    lcd.setCursor(0.1);
   lcd.print("Light: ");
26
    lcd.setCursor(13.1):
27
28
    lcd.print("lux"): //
29 }
```

```
30 void loop(){
   int adcValue; // 실제 센서로부터 읽은 값 (0~1023)
   int illuminance; // 현재의 밝기. 0~100%
   int lux;
           // 현재의 밝기. Tux
34
35
   // CdS cell을 통하여 입력되는 전압을 읽는다.
   adcValue = analogRead(CdSPin);
36
   // luminosity() 함수를 이용해서 Lux 를 계산한다.
   lux = int(luminosity(adcValue));
   // 전에 표시했던 내용을 지운다.
40
   lcd.setCursor(12,0);
   lcd.print(" ");
   // ADC 값을 표시한다
   lcd.setCursor(12,0);
   lcd.print(adcValue);
45
                                   LED ON/OFF
   // 전에 표시했던 내용을 지운다.
46
   lcd.setCursor(9.1);
                                   기능을 추가해서
   lcd.print(" ");
   // 밝기를 표시한다
                                  Code를 완성 후,
   lcd.setCursor(9,1);
50
   lcd.print(lux);
51
                                 AAnn_LCD_lux.
52
  // On/Off LED by threshold
54
                                      로 저장...
   delay(1000);
57 }
```

ino

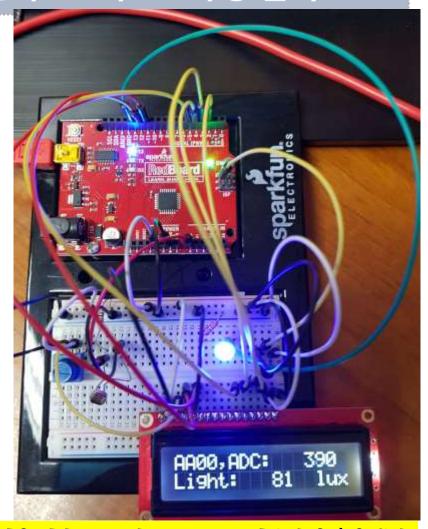


CdS-LCD project: result

CdS 센서 LCD 회로 - 측정 결과

주변의 조도에 따라 어두우면 LED가 켜지고, 밝으면 LED가 꺼지도록 코드를 수정하시오.

LED가 켜진 화면을 폰으로 촬영해서 그림을 제출하시오.



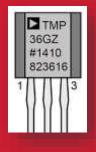
조도에 따라 LED가 ON/OFF 되는 것을 확인 받고 결과 화면 촬영: AAnn_LCD_lux.png 로 저장...



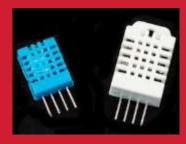


Arduino

& Node.js

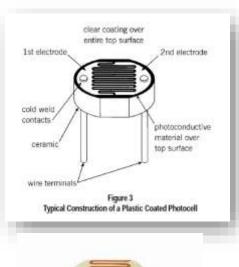




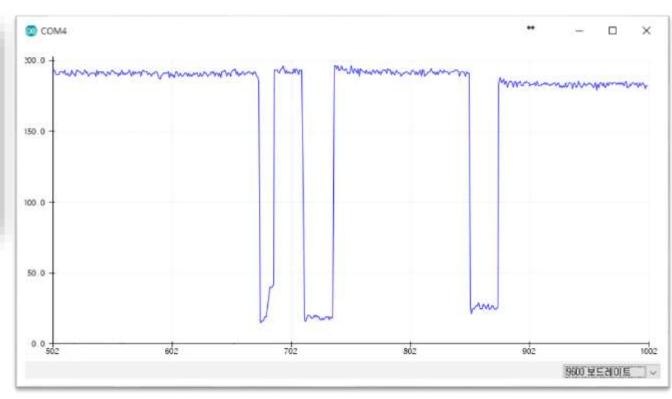




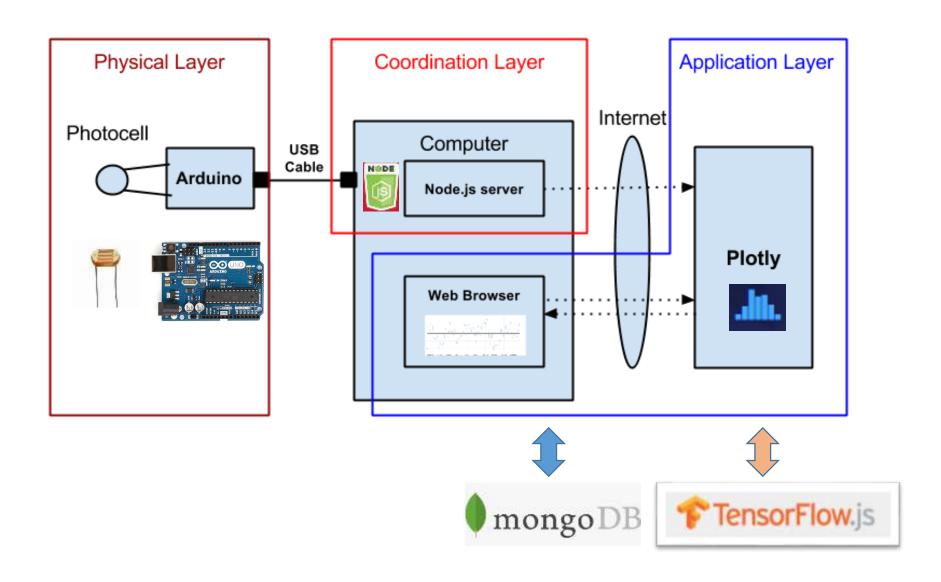
IOT: HSC







Layout [H S C]



Arduino data + plotly



Real-time Weather Station from sensors

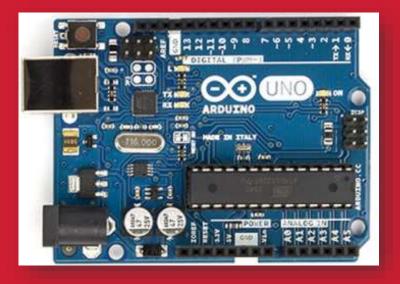


on Time: 2018-01-22 17:58:31.012



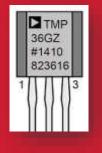


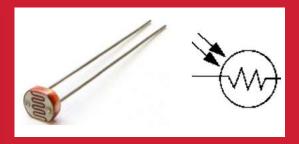
Arduino

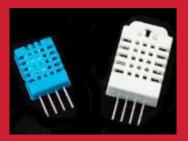


Sensors

+ Node.js











[Practice]

- **♦** [wk05]
- Arduino sensors
- Complete your project
- Upload folder: aax-nn-rpt05
- Use repo "aax-nn" in github

wk05: Practice-05: AAnn_Rpt05



- [Target of this week]
 - Complete your works
 - Save your outcomes and upload 3 figures in github

Upload folder: aax-nn-rpt05

- 제출할 파일들
 - ① AAnn_TMP36.png
 - 2 AAnn_LCD_hello.png
 - 3 AAnn_LCD_lux.png
 - 4 All *.ino

Lecture materials



References & good sites

- ✓ http://www.arduino.cc Arduino Homepage
- http://www.nodejs.org/ko Node.js
- https://plot.ly/ plotly
- https://www.mongodb.com/ MongoDB
- ✓ http://www.w3schools.com

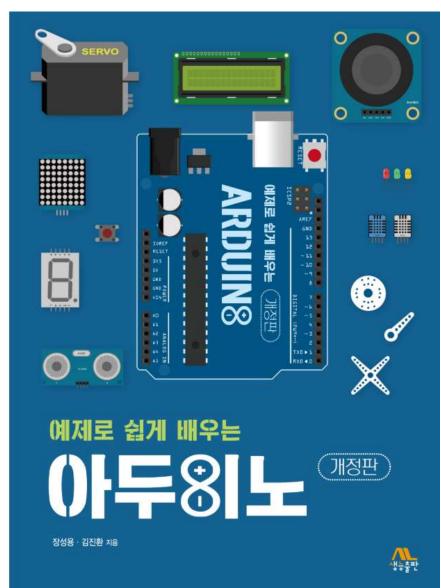
 By w3schools.com
- http://www.github.com GitHub





주교재 및 참고도서





Target of this class





Real-time Weather Station from sensors



on Time: 2018-01-22 17:58:31.012



Target of this class

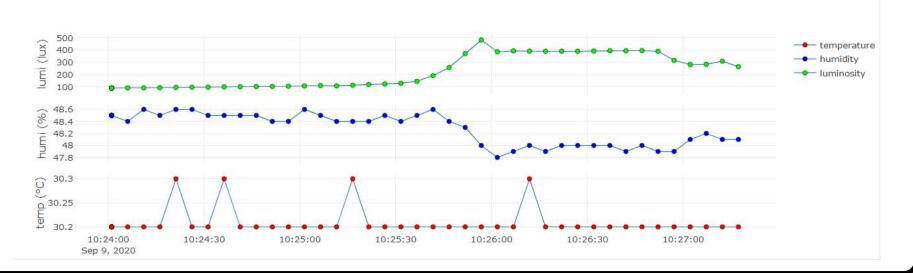




Real-time Weather Station from nano 33 BLE sensors



on Time: 2020-09-09 10:27:17.321



Another target of this class





