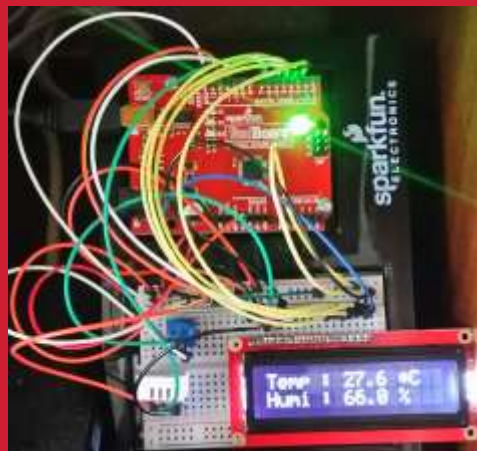




Arduino-IOT

[wk05]

Arduino sensors



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

Comsi, INJE University

2nd semester, 2018

Email : chaos21c@gmail.com





My ID

진영빈	AA01
김태은	AA02
도한솔	AA03
박지수	AA04
신성	AA05
박현승	AA06
이석주	AA07
전규은	AA08
정영관	AA09
정의석	AA10

이근재

AA11



[Review]

◆ [wk04]

- **Arduino basic circuits**
- **Complete your project**
- **Submit file : AAnn_Rpt03.zip**

◆ [Target of this week]

- Complete your works
- Save your outcomes and compress 3 figures

제출파일명 : **AAnn_Rpt03.zip**

- 압축할 파일들

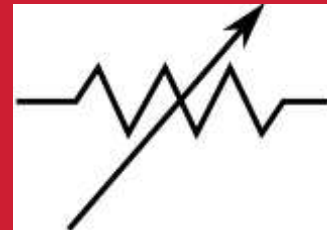
- ① **AAnn_Monitoring.png**
- ② **AAnn_multi_Monitoring.png**
- ③ **AAnn_multi_Signals.png**

Email : chaos21c@gmail.com

[제목 : id, 이름 (수정)]

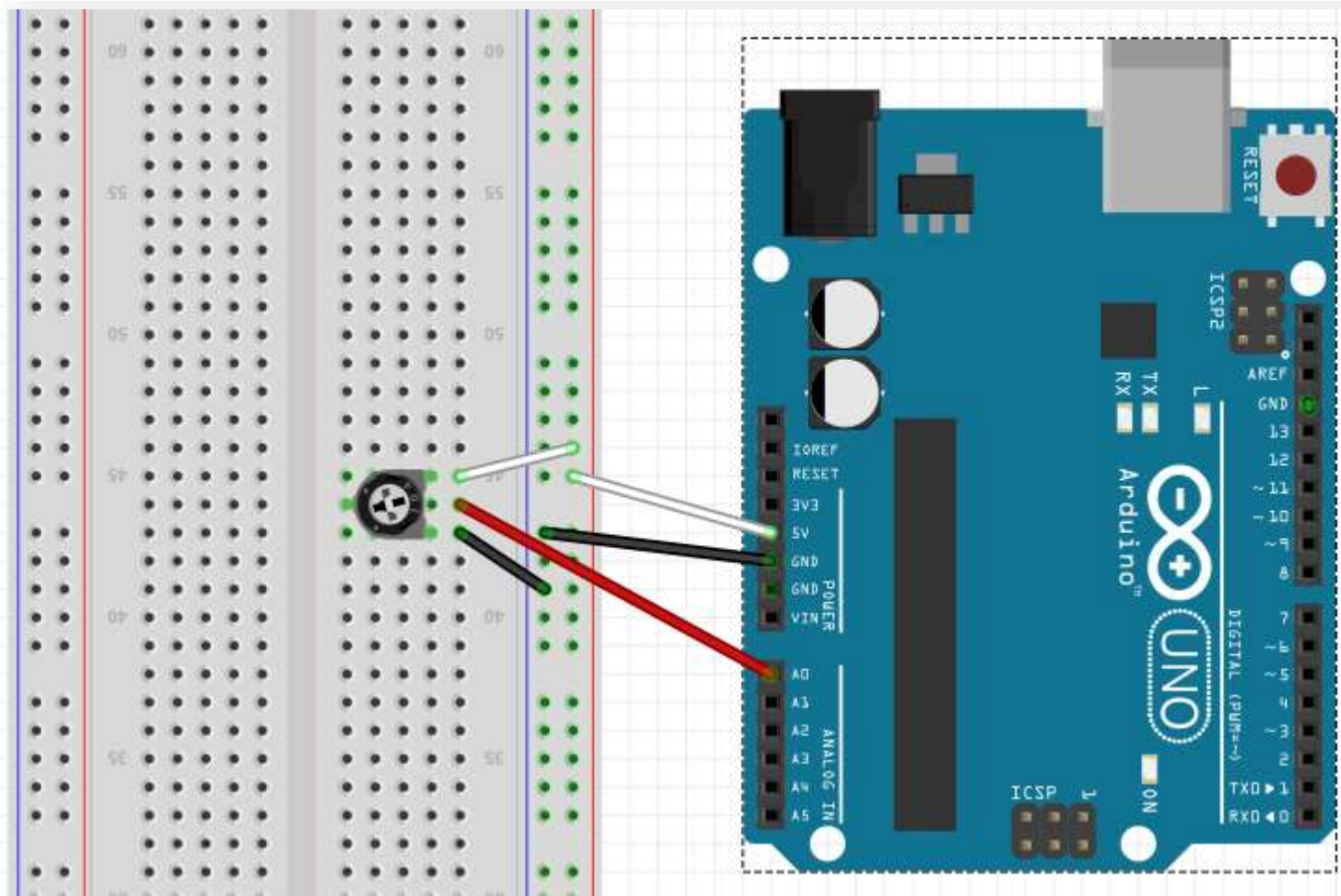


Analog Signal



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.

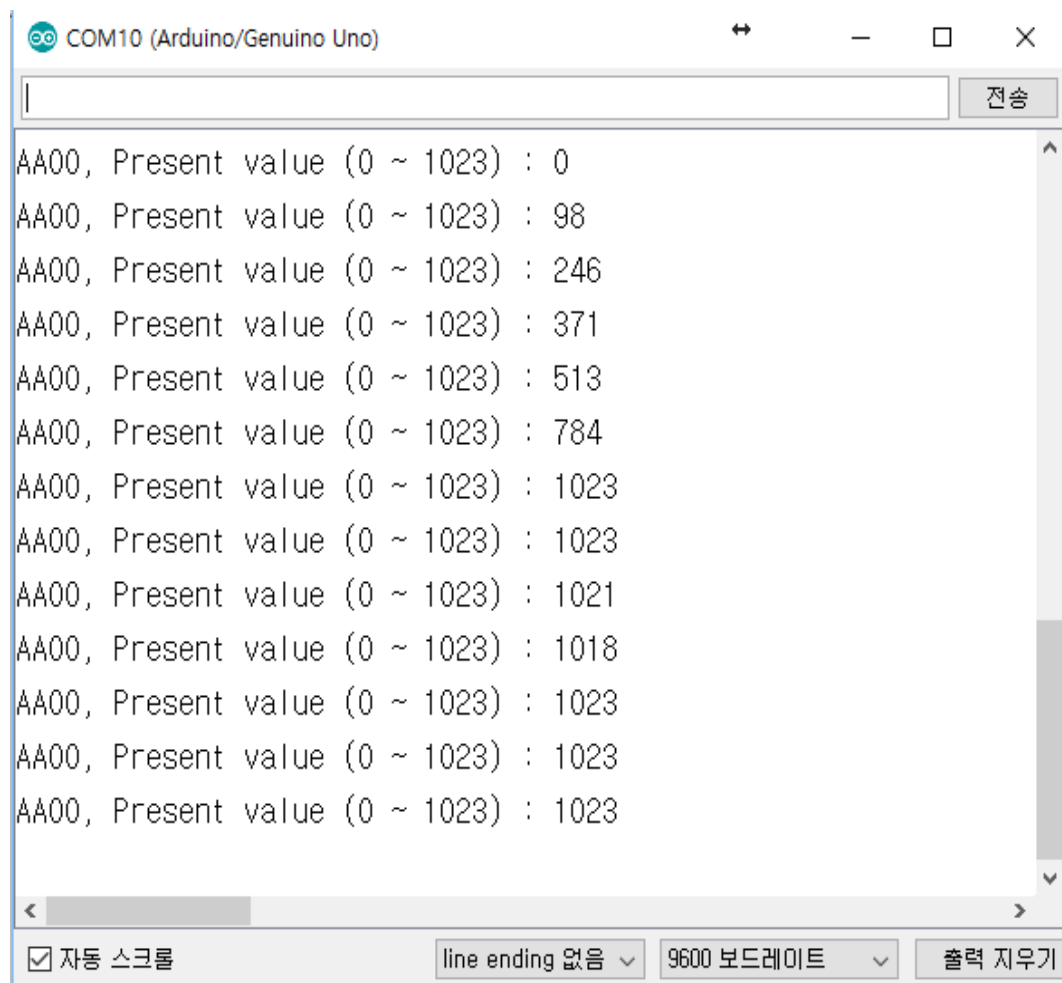
▶ 아두이노 코드 : sketch06_analog_read.ino

```
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 : $0 < \text{value} < 1023$



아날로그 값을 저항 및 전압으로 변환

▶ 저항 또는 전압 환산

$$1. \text{저항} = 10.0 * A0 / 1023 \text{ (k}\Omega\text{)}$$

$$2. \text{전압} = 5.0 * A0 / 1023 \text{ (V)}$$

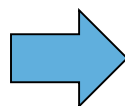
A0: 아날로그 핀 A0에서의 측정값 (0 ~ 1023)

A2.5.5 Analog value to Resistance

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

```
COM10 (Arduino/Genuino Uno)
전송

AA00, Present value (0 ~ 1023) : 0
AA00, Present value (0 ~ 1023) : 98
AA00, Present value (0 ~ 1023) : 246
AA00, Present value (0 ~ 1023) : 371
AA00, Present value (0 ~ 1023) : 513
AA00, Present value (0 ~ 1023) : 784
AA00, Present value (0 ~ 1023) : 1023
AA00, Present value (0 ~ 1023) : 1023
AA00, Present value (0 ~ 1023) : 1021
AA00, Present value (0 ~ 1023) : 1018
AA00, Present value (0 ~ 1023) : 1023
AA00, Present value (0 ~ 1023) : 1023
AA00, Present value (0 ~ 1023) : 1023
```



```
COM10 (Arduino/Genuino Uno)
전송

AA00, Present R (0 ~ 10.0) : 0.00
AA00, Present R (0 ~ 10.0) : 0.12
AA00, Present R (0 ~ 10.0) : 2.68
AA00, Present R (0 ~ 10.0) : 3.45
AA00, Present R (0 ~ 10.0) : 4.15
AA00, Present R (0 ~ 10.0) : 5.34
AA00, Present R (0 ~ 10.0) : 6.68
AA00, Present R (0 ~ 10.0) : 7.50
AA00, Present R (0 ~ 10.0) : 8.43
AA00, Present R (0 ~ 10.0) : 10.00
AA00, Present R (0 ~ 10.0) : 9.98
AA00, Present R (0 ~ 10.0) : 9.96
AA00, Present R (0 ~ 10.0) : 10.00
```

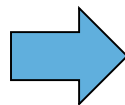
```
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  // print out the value you read:
  Serial.print("AA00, Present R (0 ~ 10.0) : ");
  float resistance = sensorValue*(10.0/1023.0); // kΩ
  Serial.println(resistance);
  delay(500);    // 2 Hz sampling
}
```

A2.5.6 Analog value to Voltage

Serial monitor : Voltage ($0 < V < 5 \text{ V}$)

```
COM10 (Arduino/Genuino Uno)
전송

AA00, Present value (0 ~ 1023) : 0
AA00, Present value (0 ~ 1023) : 98
AA00, Present value (0 ~ 1023) : 246
AA00, Present value (0 ~ 1023) : 371
AA00, Present value (0 ~ 1023) : 513
AA00, Present value (0 ~ 1023) : 784
AA00, Present value (0 ~ 1023) : 1023
AA00, Present value (0 ~ 1023) : 1023
AA00, Present value (0 ~ 1023) : 1021
AA00, Present value (0 ~ 1023) : 1018
AA00, Present value (0 ~ 1023) : 1023
AA00, Present value (0 ~ 1023) : 1023
AA00, Present value (0 ~ 1023) : 1023
```



```
COM10 (Arduino/Genuino Uno)
전송

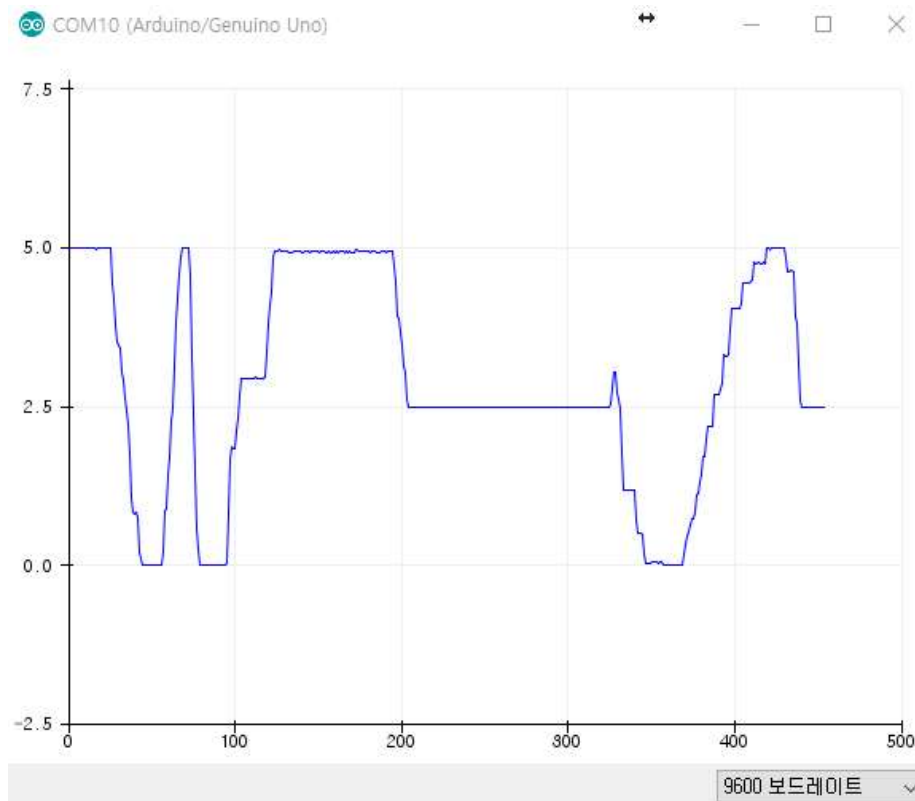
AA00, Present V (0 ~ 5.0) : 0.00
AA00, Present V (0 ~ 5.0) : 0.25
AA00, Present V (0 ~ 5.0) : 0.75
AA00, Present V (0 ~ 5.0) : 1.73
AA00, Present V (0 ~ 5.0) : 2.26
AA00, Present V (0 ~ 5.0) : 2.61
AA00, Present V (0 ~ 5.0) : 3.37
AA00, Present V (0 ~ 5.0) : 4.20
AA00, Present V (0 ~ 5.0) : 4.81
AA00, Present V (0 ~ 5.0) : 5.00
AA00, Present V (0 ~ 5.0) : 4.99
AA00, Present V (0 ~ 5.0) : 5.00
AA00, Present V (0 ~ 5.0) : 5.00
```

```
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  // print out the value you read:
  Serial.print("AA00, Present V (0 ~ 5.0) : ");
  float voltage= sensorValue*(5.0/1023.0); // V
  Serial.println(voltage);
  delay(500);    // 2 Hz sampling
}
```

A2.5.7 ReadAnalogVoltage

Result

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



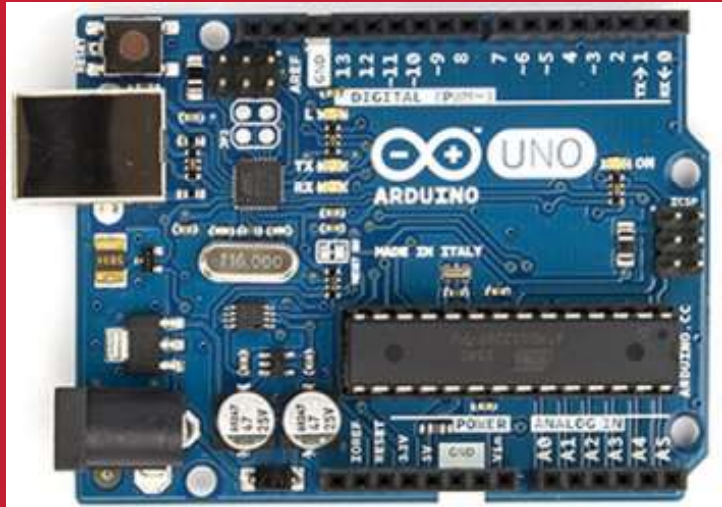
Save as
AAnn_AnalogVoltage.png



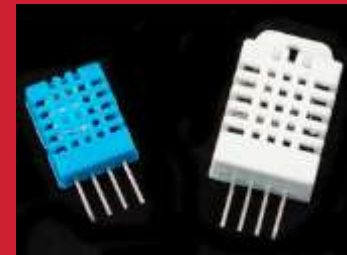
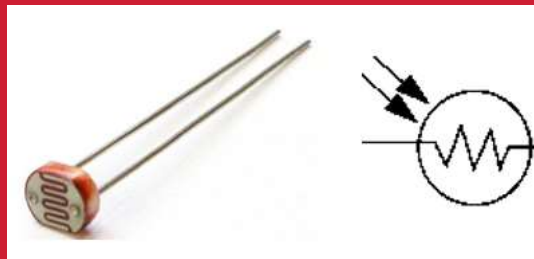
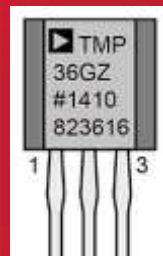
A2.5.8 ReadAnalogVoltage using f_map()

Hint code : f_map() instead of map()

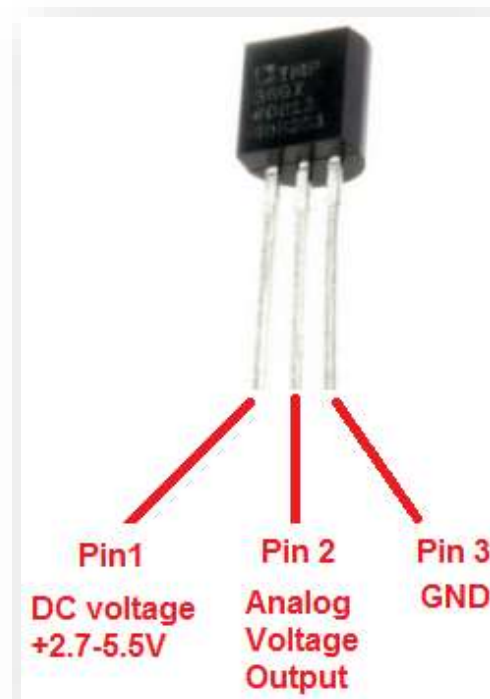
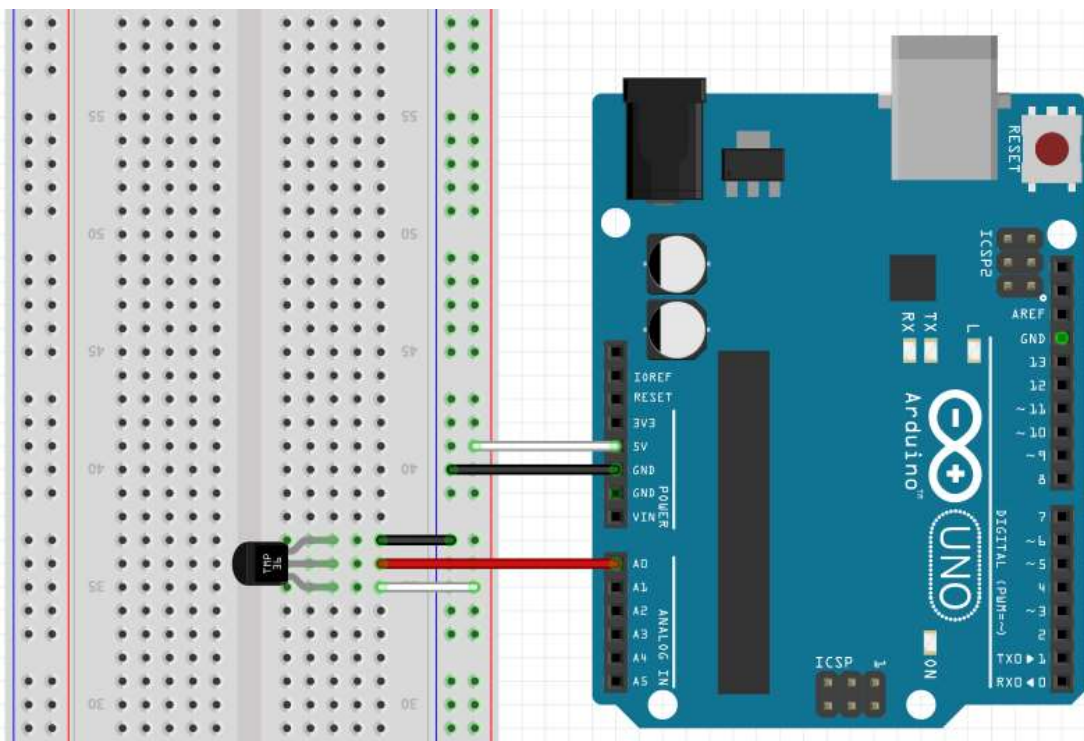
```
AAnn_AnalogRead_fmap $
9 // the setup routine runs once when you press reset:
10 void setup() {
11   // initialize serial communication at 9600 bits per second:
12   Serial.begin(9600);
13 }
14
15 // the loop routine runs over and over again forever:
16 void loop() {
17   // read the input on analog pin 0:
18   int sensorValue = analogRead(A0);
19   //float voltage = map(sensorValue, 0, 1023, 0.0, 5.0); // map 0~1023 to 0~5
20   // float voltage = sensorValue*(5.0/1023.0);
21   float voltage = f_map(sensorValue, 0, 1023, 0.0, 5.0); // map 0~1023 to 0~5
22   // print out the value you read:
23   Serial.print("AA00, Present voltage (0.0 ~ 5.0) : ");
24   Serial.println(voltage);
25   delay(500);          // delay in between reads for stability
26 }
27
28 float f_map(long x, long in_min, long in_max, float out_min, float out_max)
29 {
30   return (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min;
31 }
```



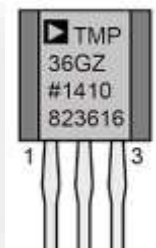
Arduino Sensors



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

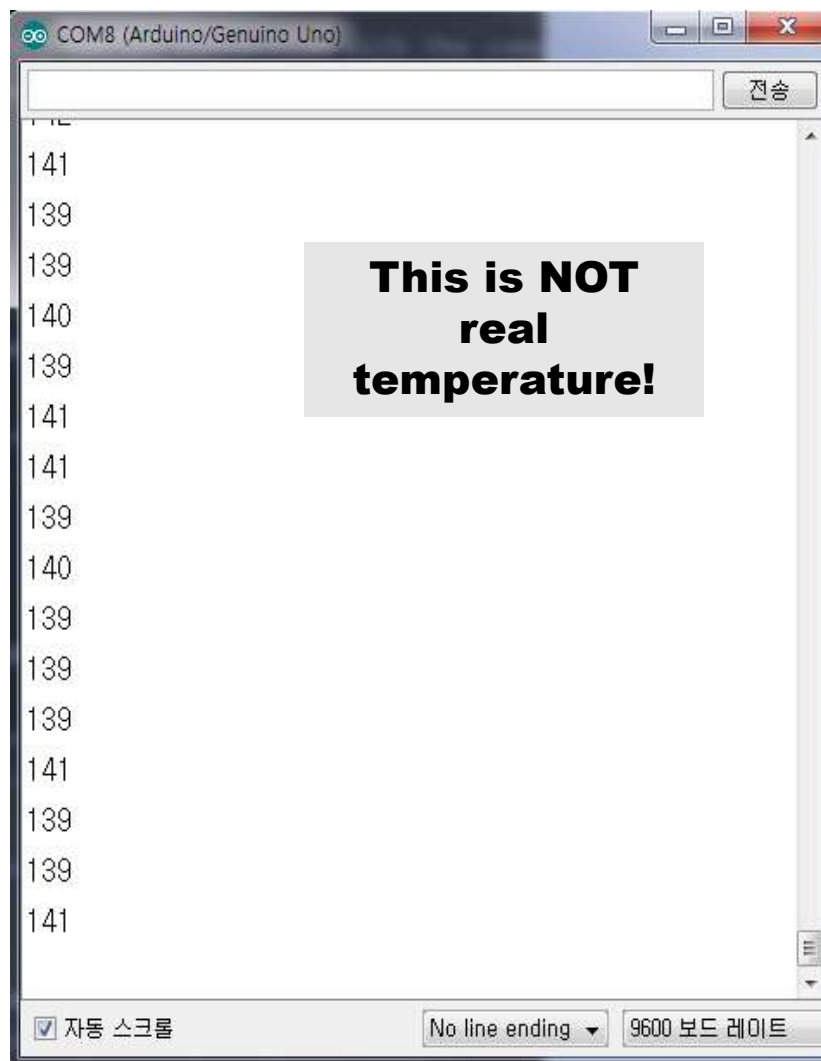
A3.1.2 Temperature sensor [TMP36]

Simple code

```

TMP36 $
1 //
2 //  AA00, 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)



A3.1.3 Temperature sensor [TMP36]

Sensor property

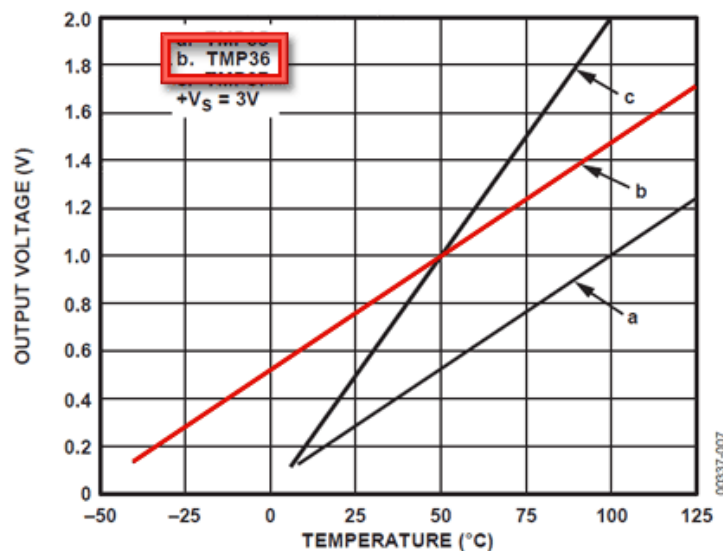


Figure 6. Output Voltage vs. Temperature

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)

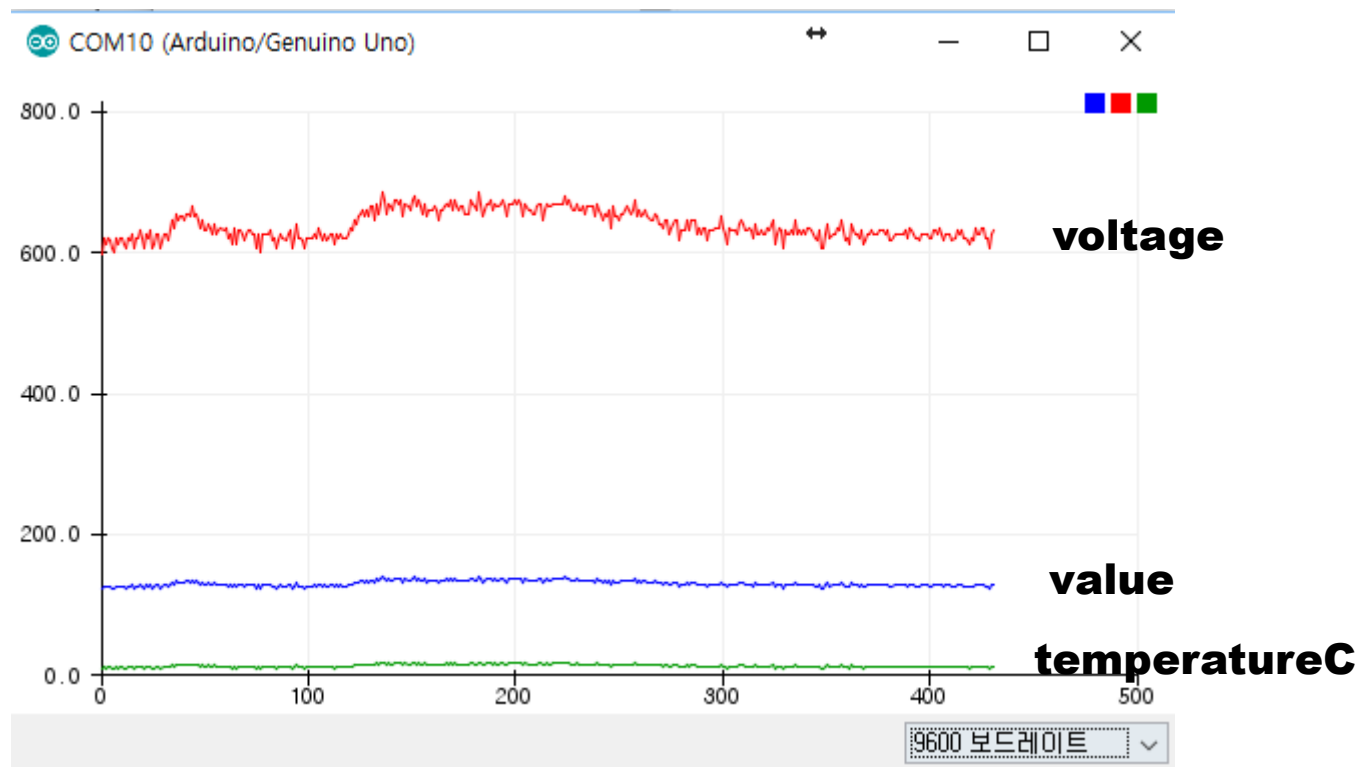
```

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

```

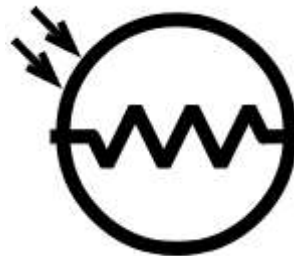
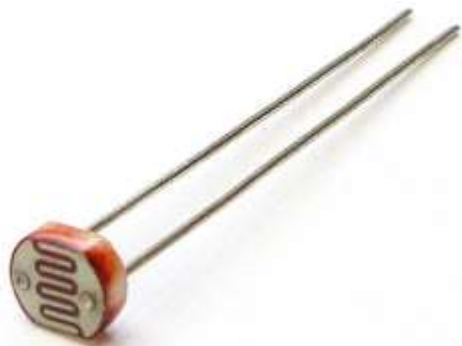


A3.1.5 Temperature sensor [TMP36]

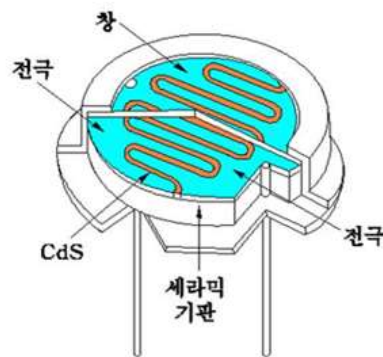


Save as
AAnn_TMP36.png

CdS 센서- photoresistor

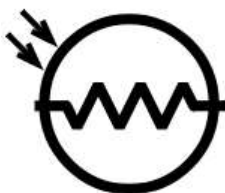
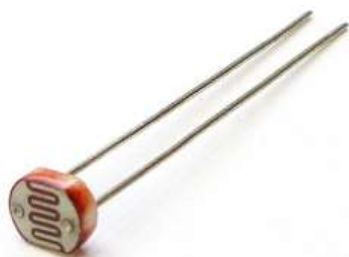


CDS특성



1. 감도
- 빛의 파장에 따라 감도가 다름
2. 허용손실
- 비교적 큰 전류를 흘릴 수 있음
3. 암 전류
- 빛이 없어도 약간의 전류가 흐름
4. 명 전류
- 빛을 비추면 흐르는 전류
5. 응답특성
- 응답 시간 지연
- 빛의 세기에 따라 응답시간 다름
6. 가변저항
- 빛에 따른 가변저항

CdS 센서 - photoresistor



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

럭스

다른 뜻에 대해서는 [Lux](#) 문서를 참조하십시오.

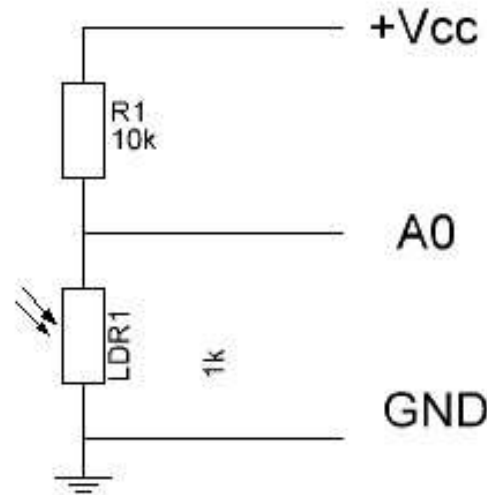
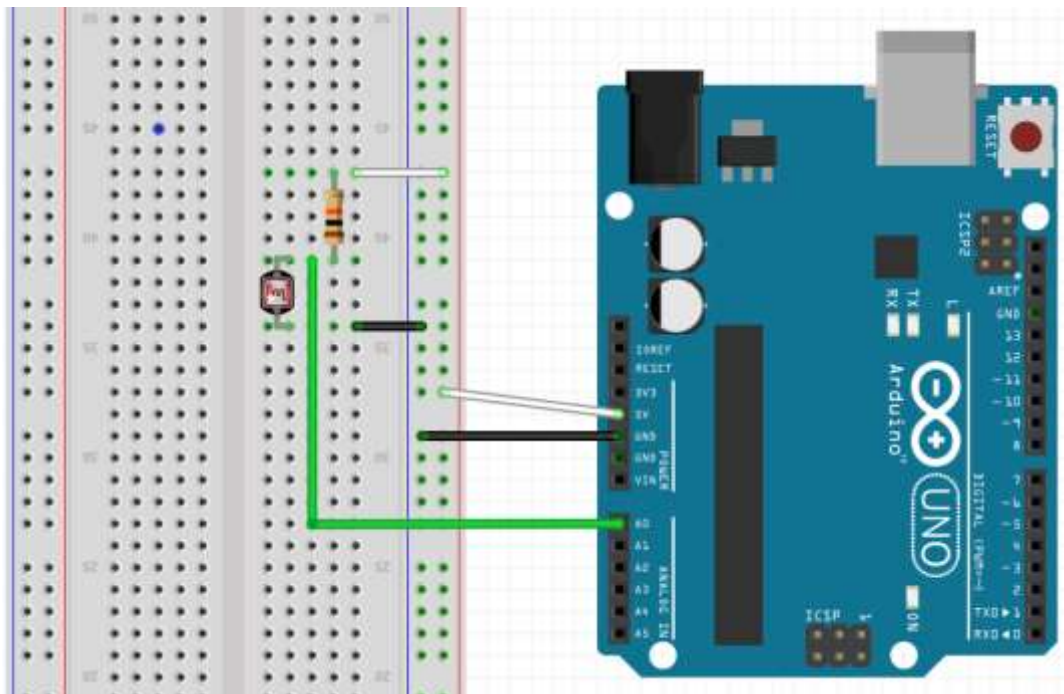
럭스(lux, 기호 **lx**)는 빛의 **조명도**를 나타내는 **SI 단위**이다. 럭스는 **루멘**에서 유도

$$1 \text{ lx} = 1 \text{ lm/m}^2 = 1 \text{ cd}\cdot\text{sr}\cdot\text{m}^{-2}$$

럭스의 예 [\[편집\]](#)

I 밝기차	예
10 ⁻⁵ lux	가장 밝은 별(시리우스)의 빛 ^[1]
10 ⁻⁴ lux	하늘을 덮은 완전한 별빛 ^[1]
0.002 lux	대기광이 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 해뜰이 뜨는 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

CdS 센서 회로



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

광센서에서의 전압 강하 값을 **A0**로 측정



▶ 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.



CdS 센서 회로 - 측정 1.

CdS_start

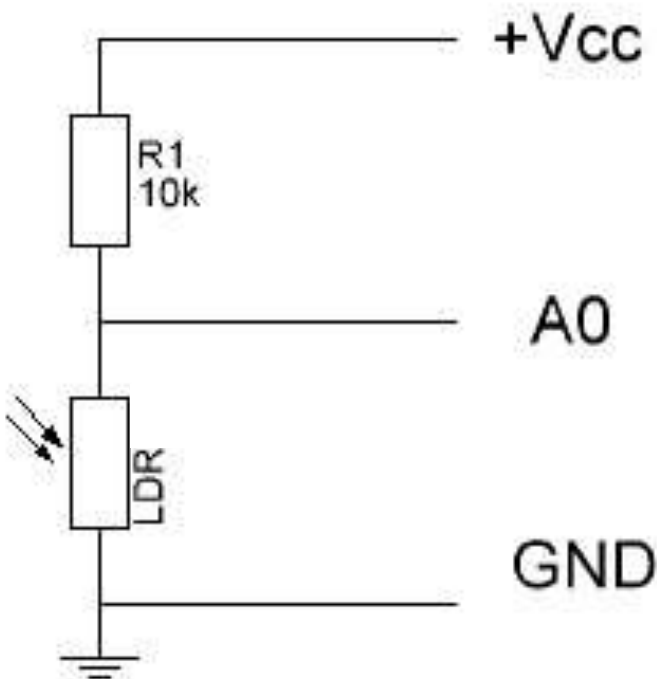
```
1 #define CDS_INPUT 0
2
3 void setup() {
4   Serial.begin(9600);
5 }
6
7 void loop() {
8
9   int value = analogRead(CDS_INPUT);
10  Serial.println(value);
11
12  delay(1000);
13 }
14
```

COM11 (Arduino/Genuino Uno)

672		어
672		두
671		올
669		날
209		때
205		때
207		밤
207		을
205		날
207		
62		
59		때
53		

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.

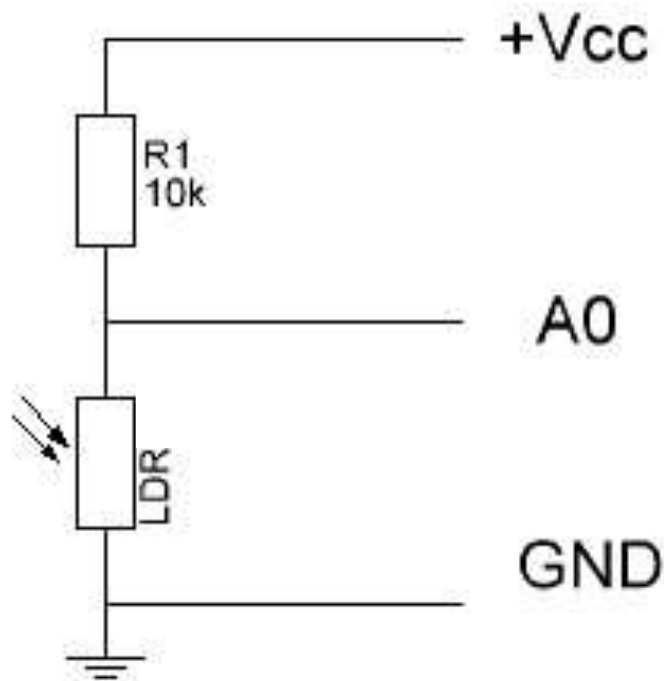
If you would use the LDR as the lower part of a voltage divider, then in darkness there would be a high voltage over the LDR, while in bright light, there would be a low voltage over that resistor.

어두우면 측정 값이 작아지고 밝을수록 값이 커져야 된다.
그리고 측정 값은 **lux**로 표현된다.

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

A0에서 측정되는 LDR
양단의 전압 = V_{out}

CdS 센서 회로 분석 (2/2)



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

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

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

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

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

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

A0에서 측정되는 **LDR**
양단의 전압 = **V_{out}**

A3.2.5 Luminosity sensor [sketch-2]

▶ 스케치 구성

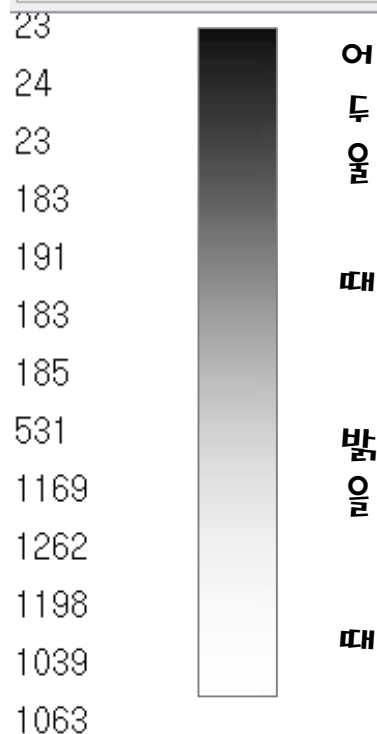
1. A0 핀을 CdS 조도 센서의 입력으로 설정한다.
2. `setup()`에서 직렬 통신 속도를 9600 bps 로 설정하고 컴퓨터와 연결한다.
3. `loop()`에서 **`analogRead()`** 함수로 A0 핀에서 측정되는 값을 읽어 들인다.
4. A0 측정값 (0 ~ 1023)을 전압 (0 ~ 5 V)으로 환산한다.
5. 전압 (V)을 온도 (°C)로 환산한 후, A0 측정값, 환산 전압, 환산 조도를 한 줄로 1 초마다 컴퓨터로 전송한다.

CdS 센서 회로 - 측정 2.

```

sketch08_CdS2
1 // lux
2 #define CDS_INPUT 0
3
4 void setup() {
5   Serial.begin(9600);
6 }
7 void loop() {
8   int value = analogRead(CDS_INPUT);
9   Serial.println(int(luminosity(value)));
10  delay(1000);
11 }
12
13 //Voltage to Lux
14 double luminosity (int RawADC0){
15   double Yout=RawADC0*5.0/1023; // 5/1023 (Vin = 5 V)
16   double lux=(2500/Yout-500)/10;
17   // lux = 500 / Rldr, Yout = Ildr*Rldr = (5/(10 + Rldr))*Rldr
18   return lux;
19 }
  
```

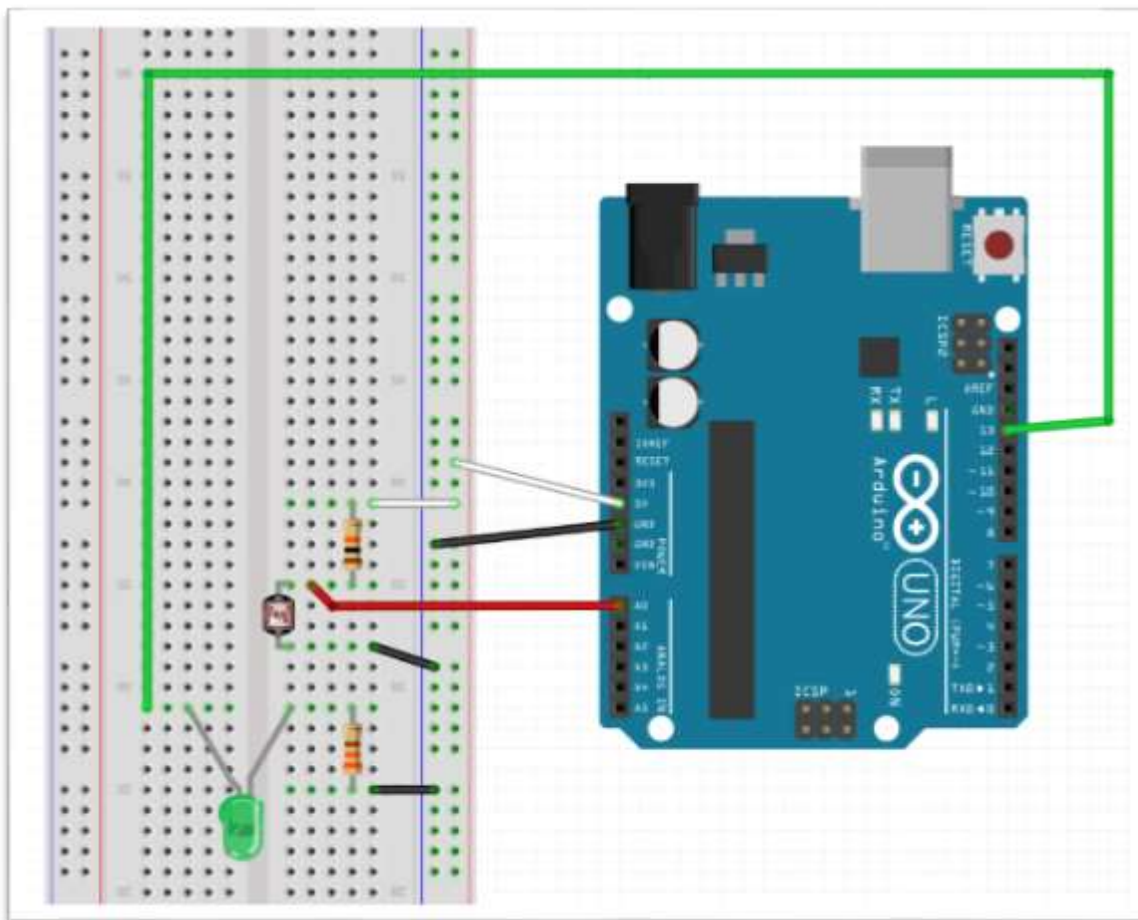
COM11 (Arduino/Genuino Uno)



밝을수록 측정 값이 커지고
어두울수록 값이 작아진다 !!!

DIY 조도 값에 따라 LED를 켜고 끄는 코드를 만드시오.

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





DIY 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 Code

```

Cds_LED
1 // lux
2 #define CDS_INPUT 0
3 // LED pin
4 const int ledPin = 13;
5
6 int threshold = 70;
7
8 void setup() {
9   pinMode(ledPin, OUTPUT);
10  Serial.begin(9600);
11 }

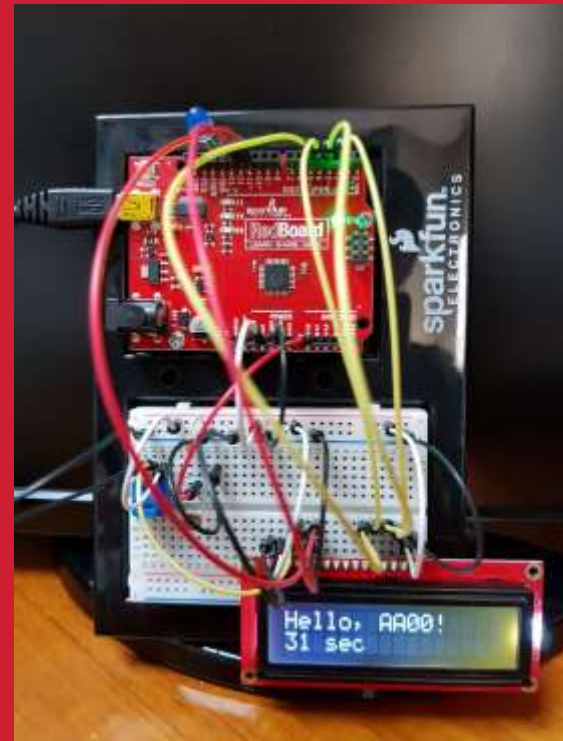
13 void loop() {
14   int value = analogRead(CDS_INPUT);
15   int lux = int(luminosity(value))
16   Serial.println(lux);
17
18   // If lux is lower than a threshold, LED is set ON.
19   if(lux >= threshold)
20     digitalWrite(ledPin, LOW);
21   else
22     digitalWrite(ledPin, HIGH);
23
24   delay(1000);
25 }
26
27 //Voltage to LuxLux
28 double luminosity (int RawADC0){
29   double Vout=RawADC0*0.0048828125; // 5/1024 (Vin = 5 V)
30   int lux=(2500/Vout-500)/10; // lux = 500 / Rldr, Vout = Ildr*Rldr = (5/(10 + Rldr))*Rldr
31   return lux;
32 }

```

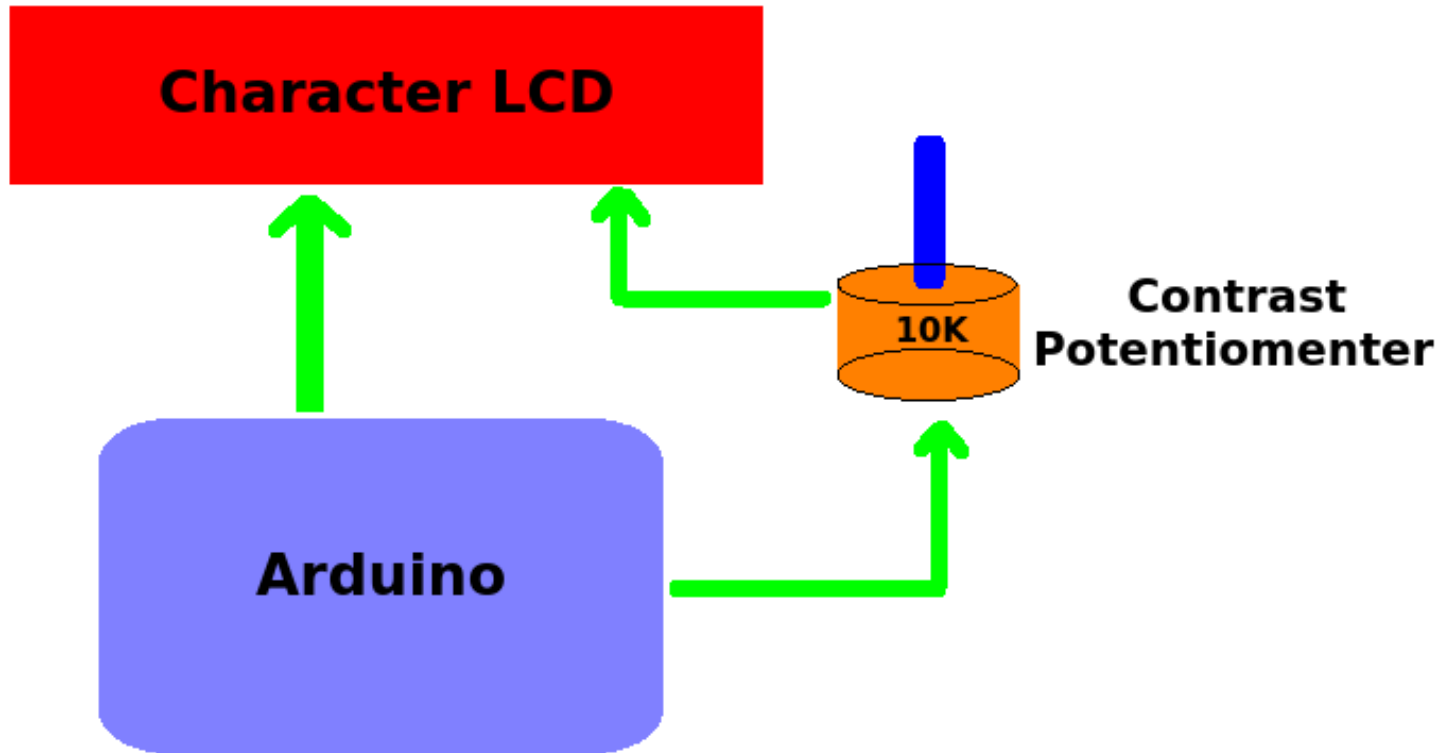
AAnn_CdS_LED.ino



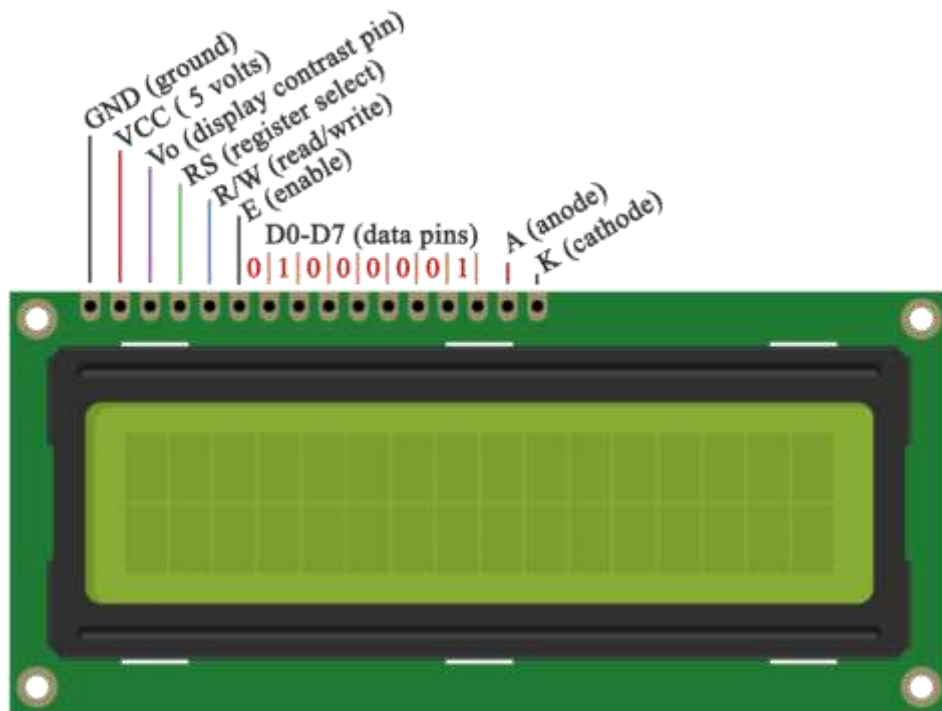
Signal Monitoring via LCD



Introduction to LCD



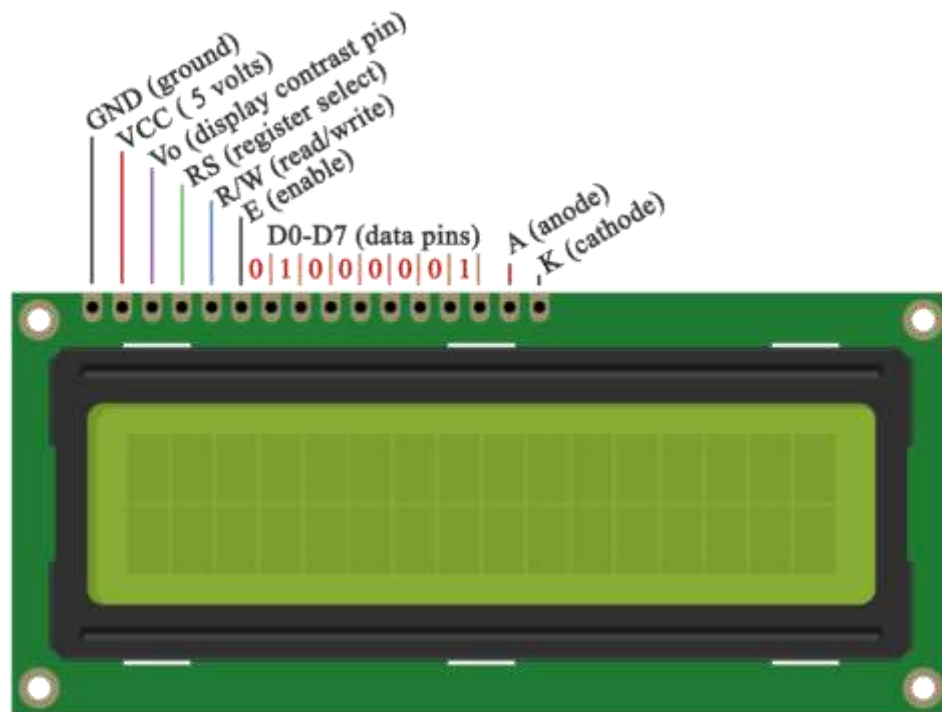
LCD (Liquid Crystal Display, 16 X 2)



(1,2, ... 15,16)

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 330 Ω)
- K (16, Backlight-)

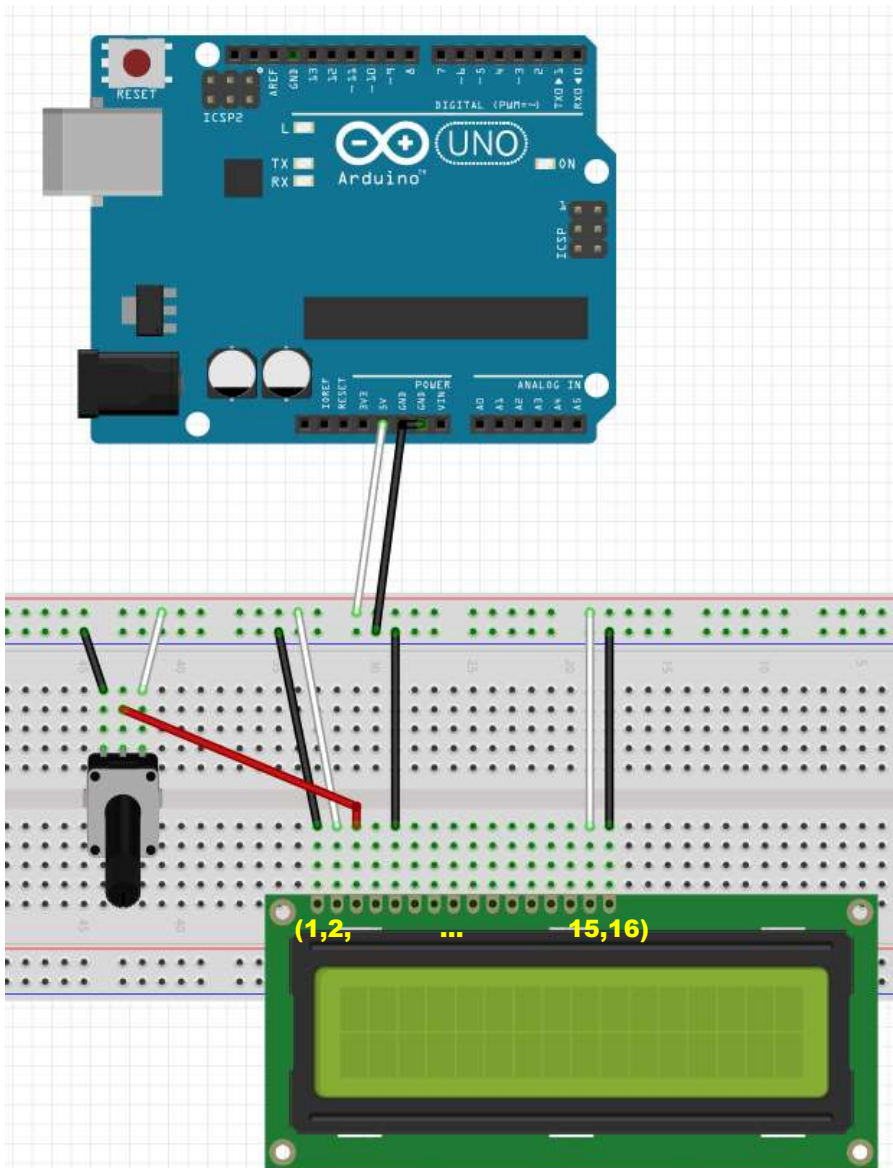
LCD (Liquid Crystal Display, 16 X 2)



(1,2, ... 15,16)

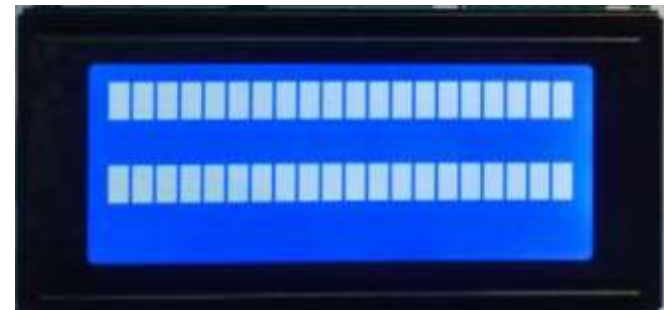
Pin 1 to Arduino GND
 Pin 2 to Arduino +5V
 Pin 3 to wiper
 Pin 4 to Arduino pin D12
 Pin 5 to Arduino GND
 Pin 6 to Arduino pin D11
 Pin 11 to Arduino pin D5
 Pin 12 to Arduino pin D4
 Pin 13 to Arduino pin D3
 Pin 14 to Arduino pin D2
 Pin 15 to +5V (with 220 or 330 Ω)
 Pin 16 to GND

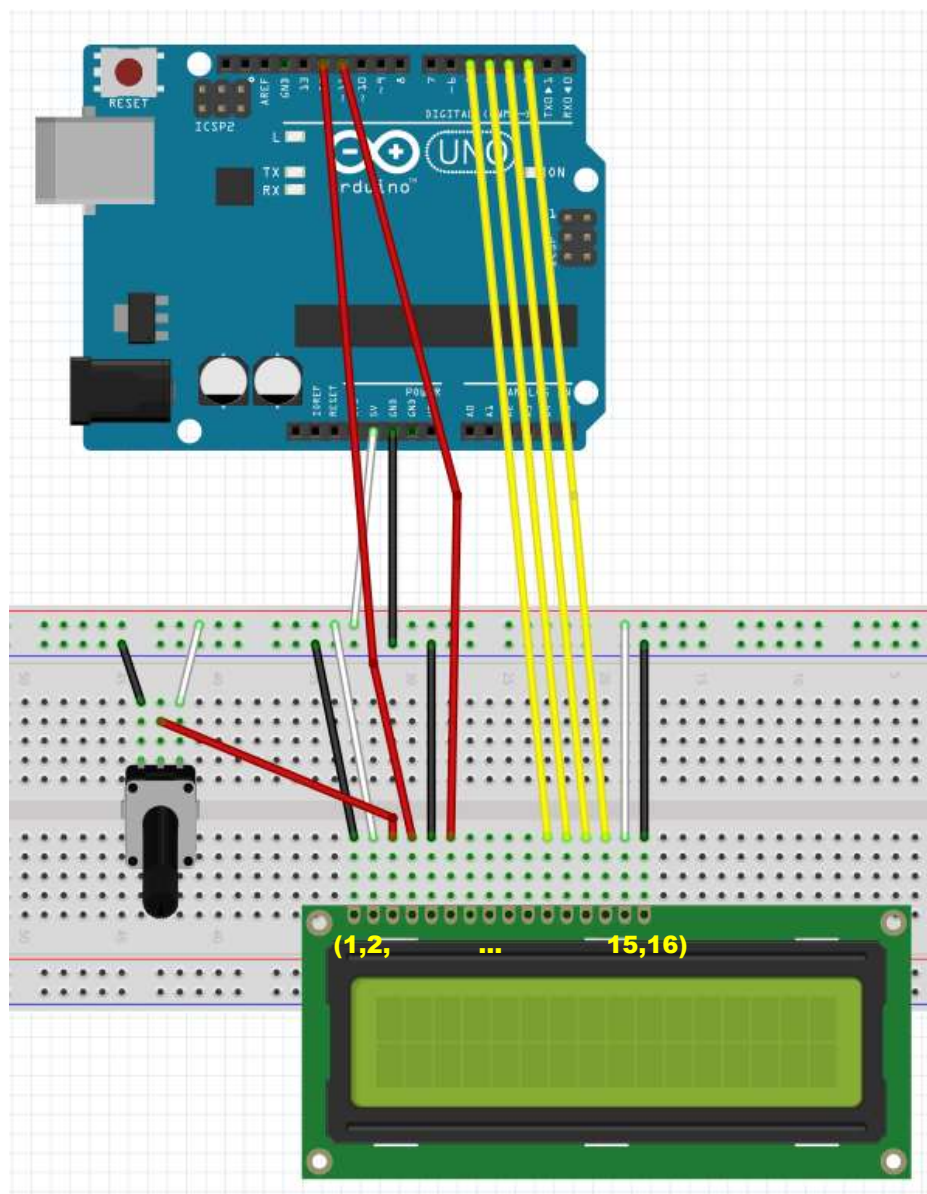
LCD 초기화 (pin-1, 2, 3, 5, 15,16)



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

전원 연결 후
LCD 초기화





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



Introduction to LCD - code “Hello AAnn”

- `LiquidCrystal lcd(rs, en, d4, d5, d6, d7)`
lcd란 이름으로 I2C에 연결된 LCD 모듈 객체.
- `lcd.begin(행, 열)`
lcd란 이름의 LCD 모듈의 크기를 정의한다.
- `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.

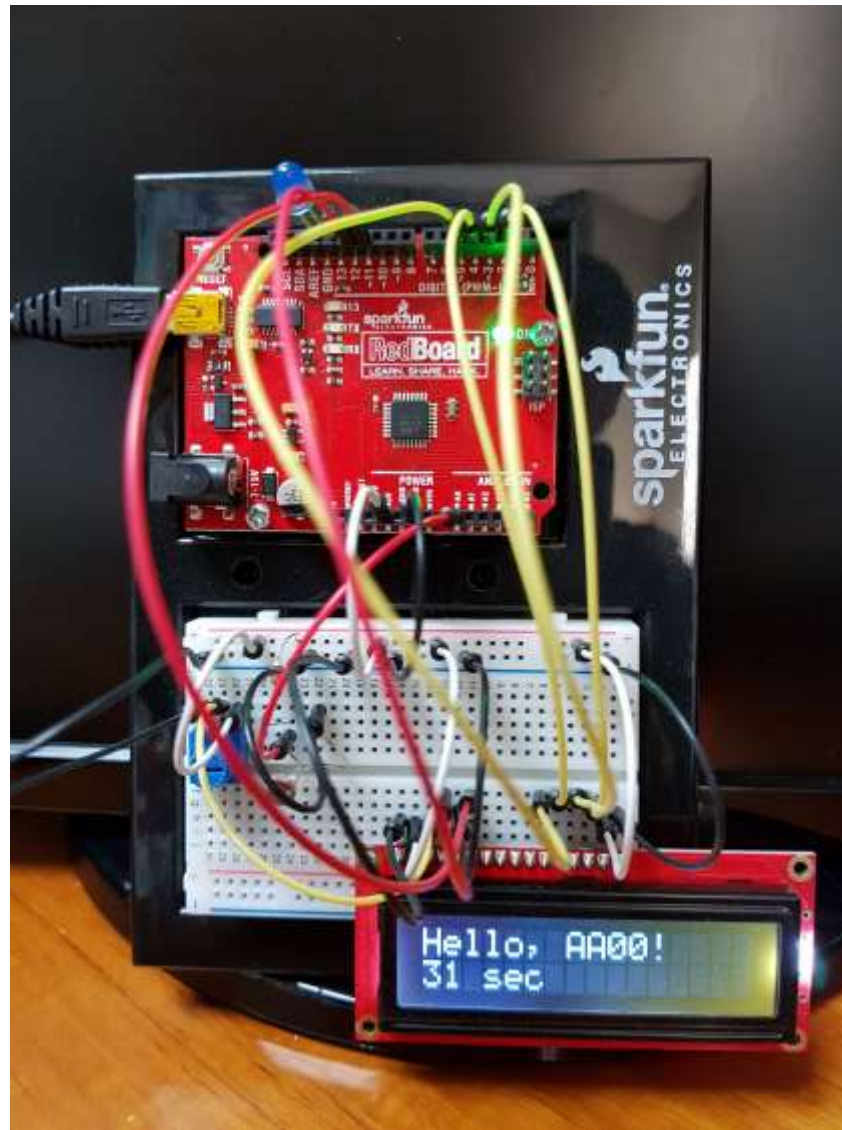


Introduction to LCD - code "Hello AAnn"

AAnn_hello_LCD

```
6
7 // include the library code:
8 #include <LiquidCrystal.h>
9
10 // initialize the library with the numbers of the interface pins
11 LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
12
13 void setup() {
14   // set up the LCD's number of columns and rows:
15   lcd.begin(16, 2);
16   // Print a message to the LCD.
17   lcd.print("Hello, AAnn!");
18 }
19
20 void loop() {
21   // set the cursor to column 0, line 1
22   lcd.setCursor(0, 1); // second line, first column
23   // print the number of seconds since reset:
24   lcd.print(millis() / 1000);
25   lcd.print(" sec");
26 }
```


Introduction to LCD - “Hello AAnn”

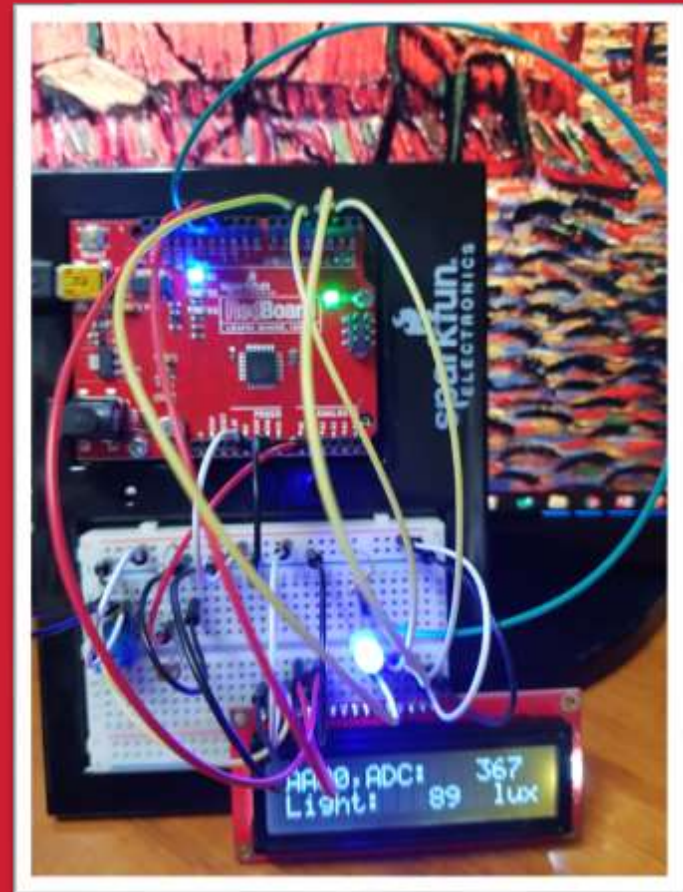


결과 화면 촬영: [AAnn_Hello_LCD.png](#) 로 저장...

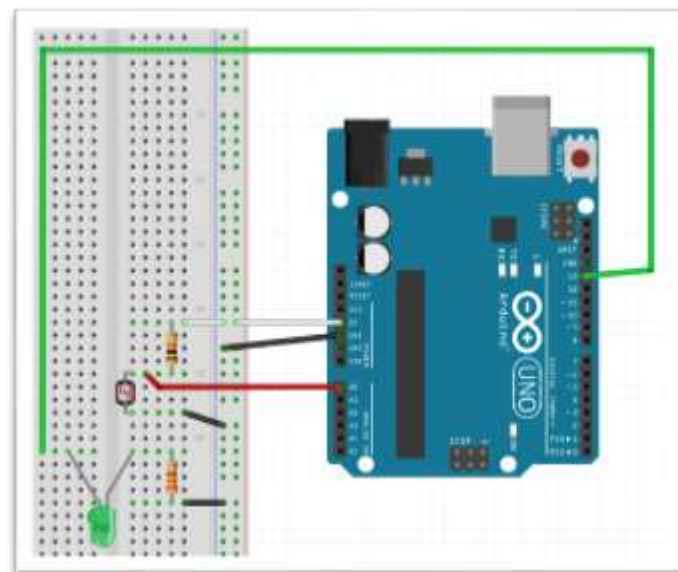
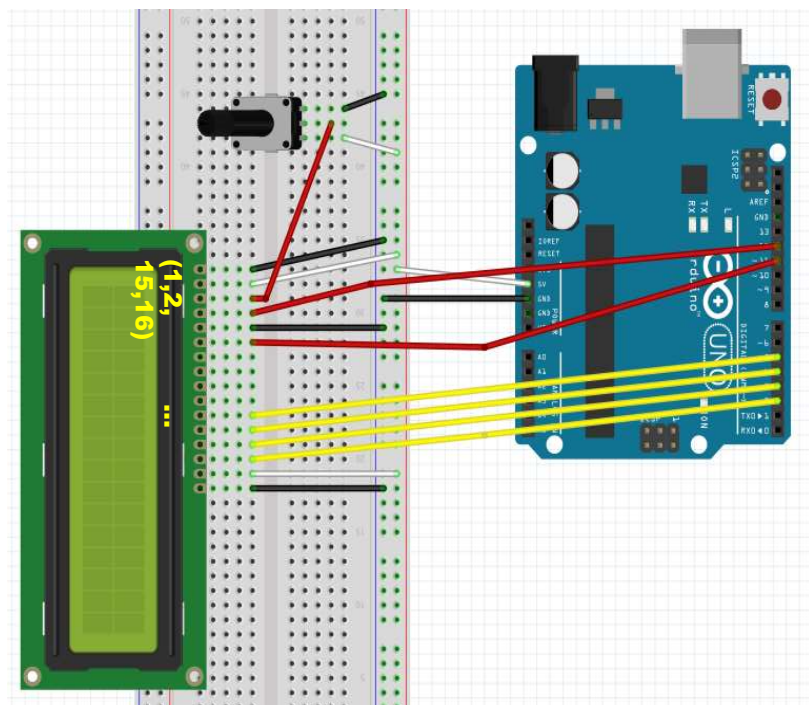


CdS LCD Project

LCD에 조도 값을
표시하면서
조도에 따라 **LED**를
ON/OFF



CdS-LCD project





CdS-LCD project

Set CdS-LCD project

Project

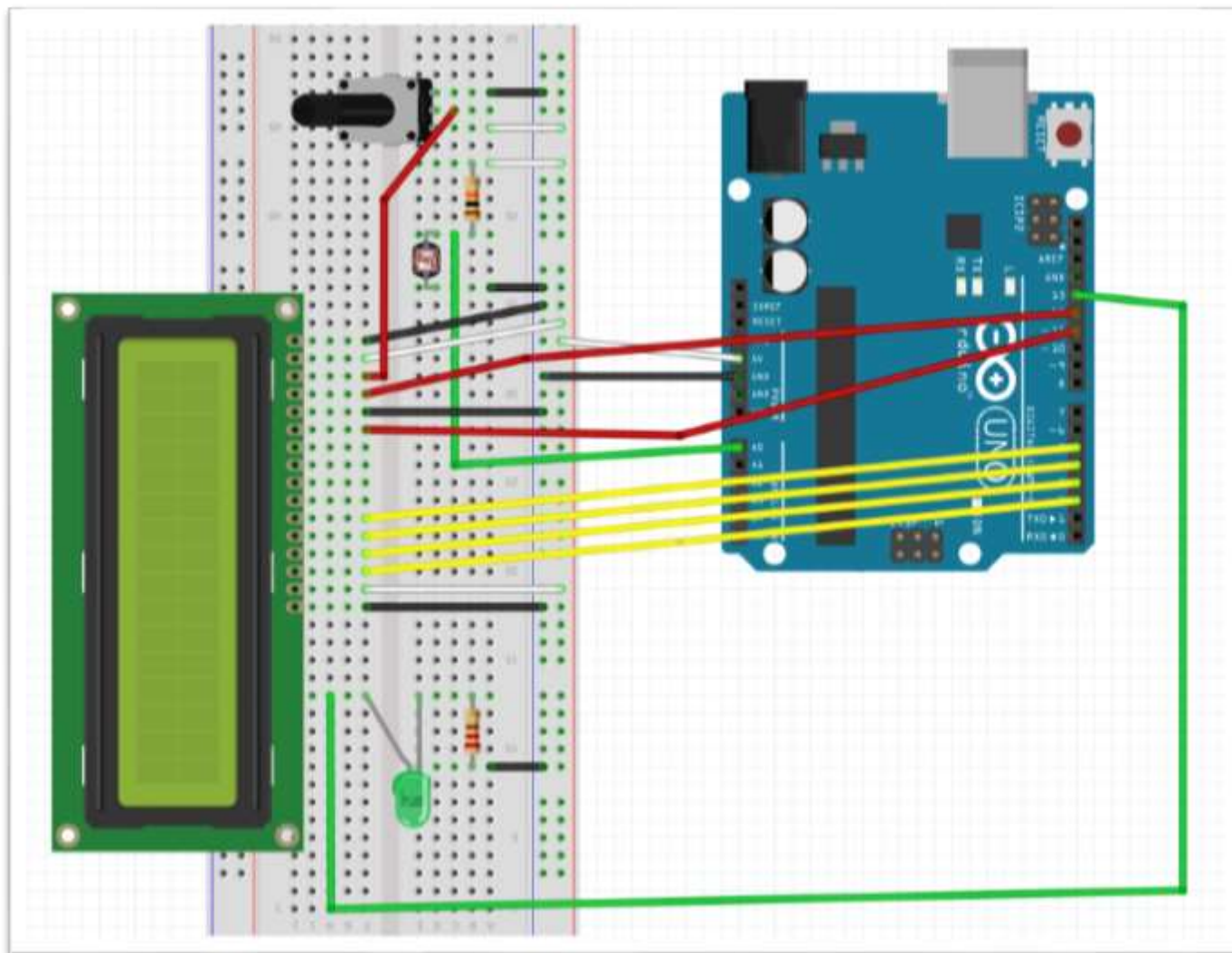
CdS 셀을 이용하여 조도를 측정해 보자.

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

Hardware

1. LCD를 연결한다.
2. CdS셀과 10k Ω 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_LED.fzz





CdS-LCD project : new code

CdS 센서 LCD 회로 - code: AAnn_LCD_lux.ino

CdS_LCD_lux_start

```

1  /*
2  빛 입력 LCD 모니터링 및 제어
3  */
4
5  // LCD 라리브러리 설정
6  #include <LiquidCrystal.h>
7  // LCD 설정
8  LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // rs,en,d4,d5,d6,d7
9  // 0번 아날로그핀을 CdS 셀 입력으로 설정한다.
10 const int CdSPin = 0;    // CdS => A0
11 const int ledPin = 13;   // LED pin => D13
12
13 // LED OFF above threshold lux
14
15 void setup() {
16   pinMode(ledPin, OUTPUT);
17   // 16X2 LCD 모듈 설정하고 백라이트를 켜다.
18   lcd.begin(16,2);
19   // 모든 메시지를 삭제한 뒤
20   // 숫자를 제외한 부분들을 미리 출력시킨다.
21   lcd.clear();
22   lcd.setCursor(0,0);
23   lcd.print("AA00,ADC: ");
24   lcd.setCursor(0,1);
25   lcd.print("Light: ");
26   lcd.setCursor(13,1);
27   lcd.print("lux"); //
28 }

```

```

30 void loop(){
31   int adcValue; // 실제 센서로부터 읽은 값 (0~1023)
32   int illuminance; // 현재의 밝기. 0~100%
33   int lux;        // 현재의 밝기. lux
34
35   // CdS cell을 통하여 입력되는 전압을 읽는다.
36   adcValue = analogRead(CdSPin);
37   // luminosity() 함수를 이용해서 Lux 를 계산한다.
38   lux = int(luminosity(adcValue));
39
40   // 전에 표시했던 내용을 지운다.
41   lcd.setCursor(12,0);
42   lcd.print(" ");
43   // ADC 값을 표시한다
44   lcd.setCursor(12,0);
45   lcd.print(adcValue);
46   // 전에 표시했던 내용을 지운다.
47   lcd.setCursor(9,1);
48   lcd.print(" ");
49   // 밝기를 표시한다
50   lcd.setCursor(9,1);
51   lcd.print(lux);
52
53   // On/Off LED by threshold
54
55   delay(1000);
56
57 }

```

LED ON/OFF

기능을 추가해서

Code를 완성 후,

AAnn_LCD_lux.

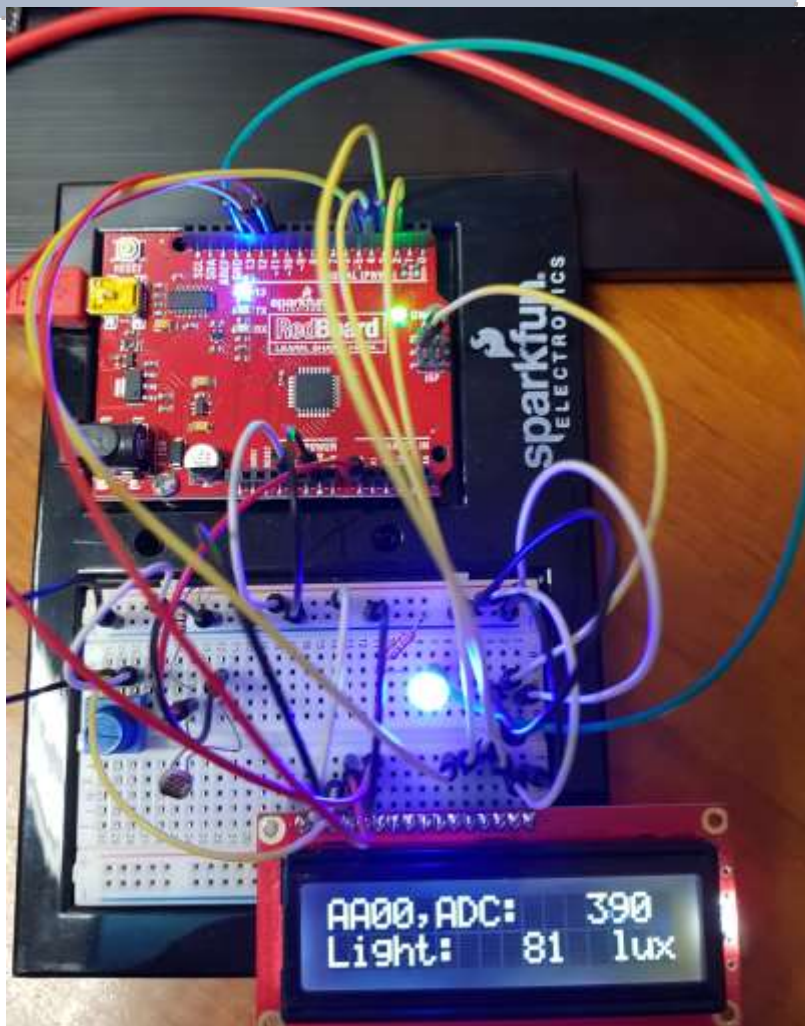
ino

로 저장...

CdS 센서 LCD 회로 - 측정 결과

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

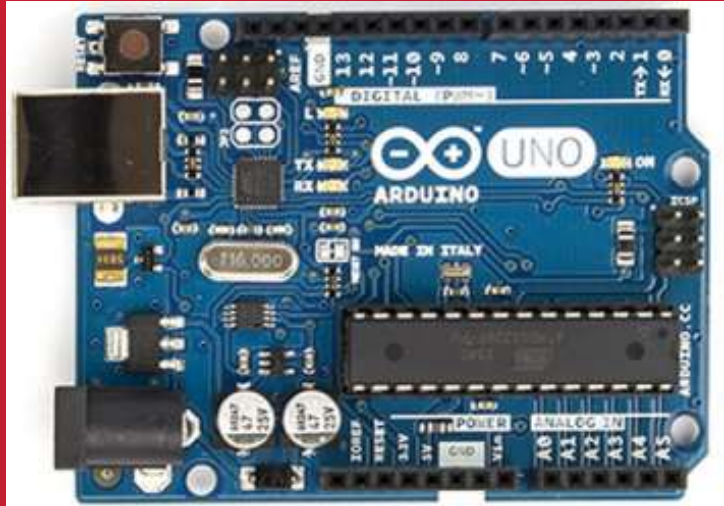
LED가 켜진
화면을 폰으로
촬영해서 그림을
제출하시오.



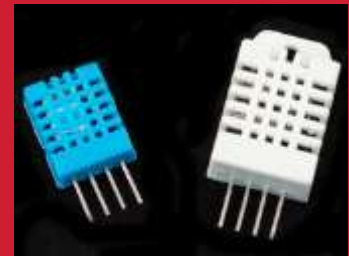
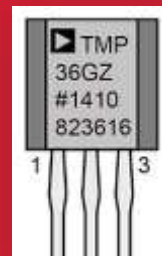
조도에 따라 **LED**가 **ON/OFF** 되는 것을 확인 받고
결과 화면 촬영: **AAnn_LCD_lux.png** 로 저장...



Next week



Arduino & Node.js





IOT: HSC

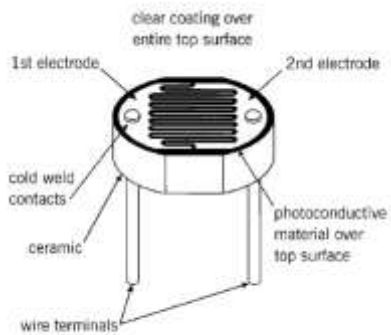
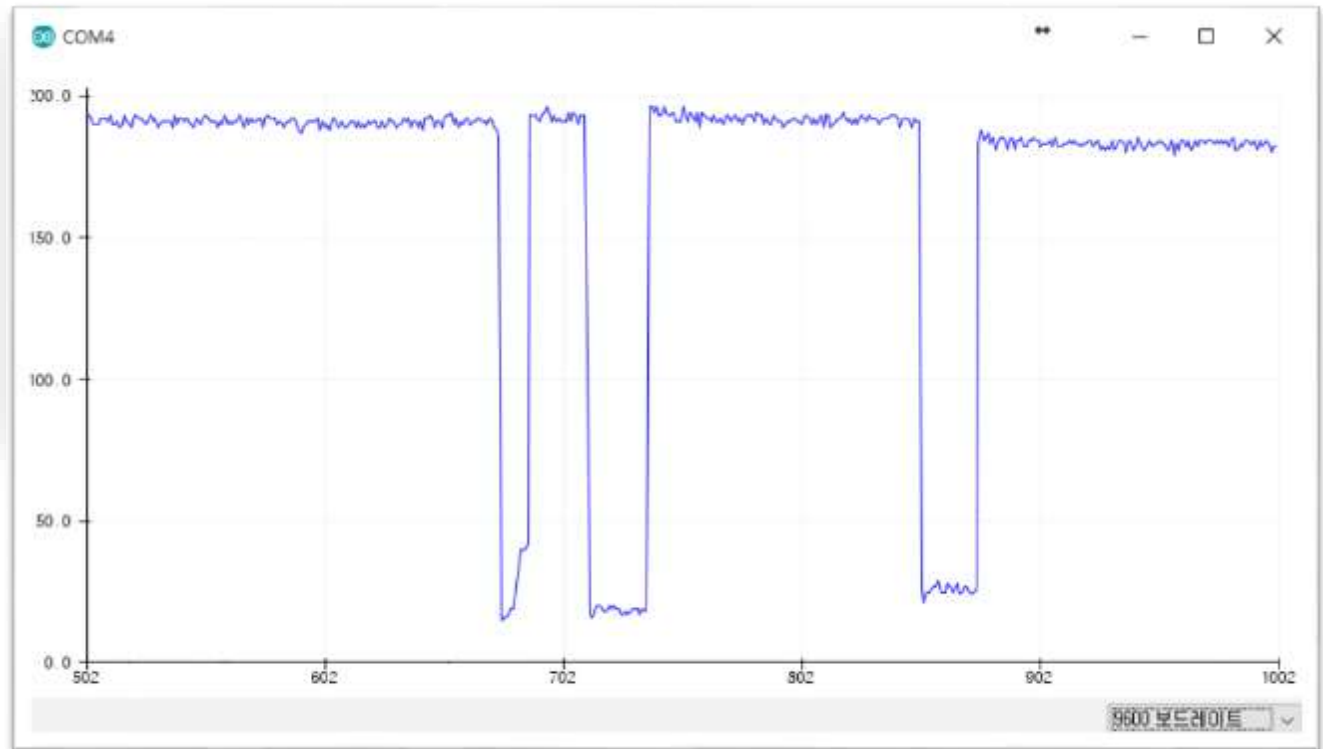
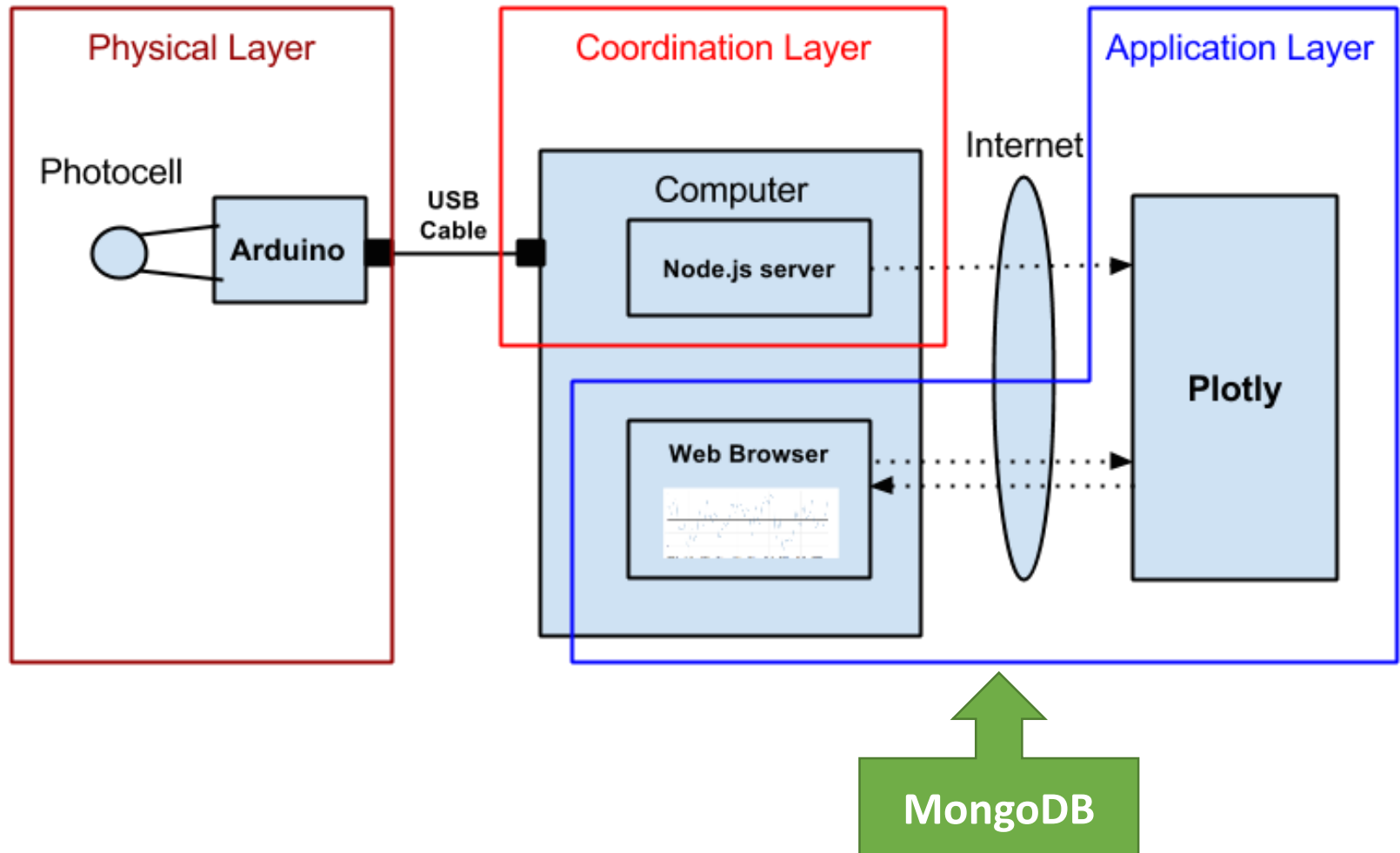


Figure 3
Typical Construction of a Plastic Coated Photocell



Layout [H S C]



Arduino data + plotly

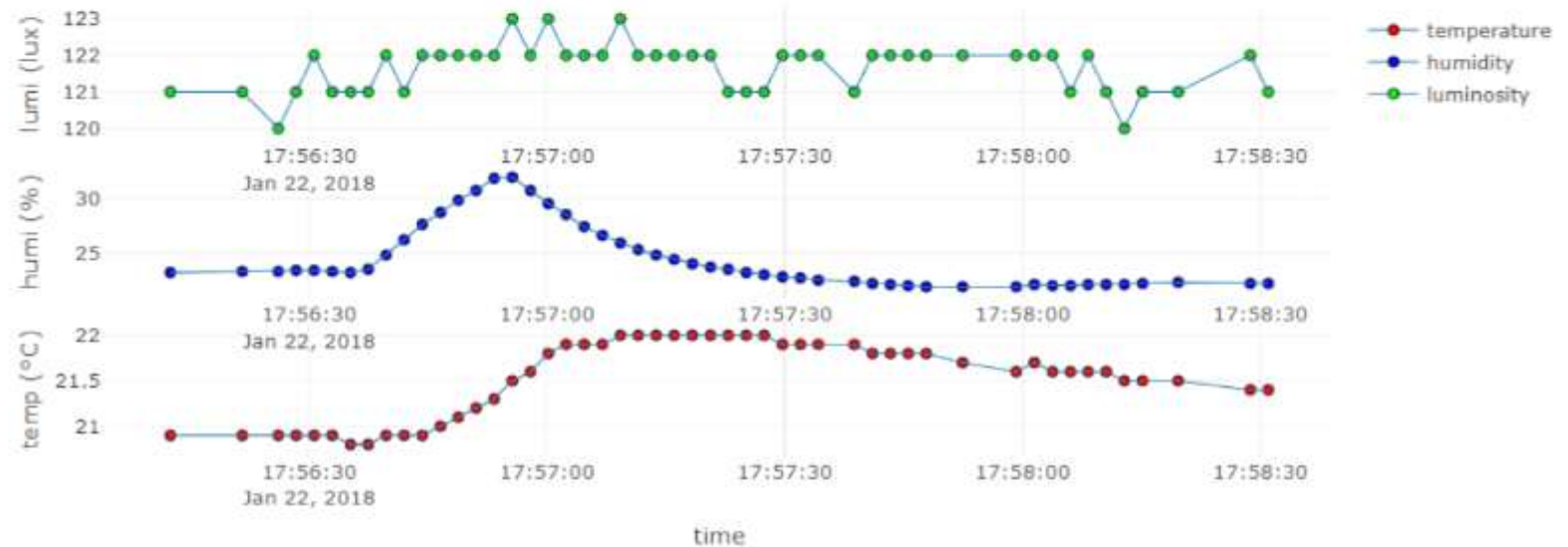
Time series by AA00



Real-time Weather Station from sensors



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





[Practice]

◆ [wk05]

- **Arduino sensors**
- **Complete your project**
- **Submit file : AAnn_Rpt04.zip**

◆ [Target of this week]

- Complete your works
- Save your outcomes and compress 6 outputs

제출파일명 : **AAnn_Rpt04.zip**

- 압축할 파일들

- ① **AAnn_AnalogVoltage.png**
- ② **AAnn_TMP36.png**
- ③ **AAnn_CdS_LED.ino**
- ④ **AAnn_Hello_LCD.png**
- ⑤ **AAnn_LCD_lux.ino**
- ⑥ **AAnn_LCD_lux.png**

[제목 : **id, 이름 (수정)**]

● 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
- ✓ <http://www.github.com> GitHub



주교재 및 참고도서

아두이노와 Node.js에 기반한 IOT 신호 시각화

| 저자 이 상 훈 |

인제대학교 출판부

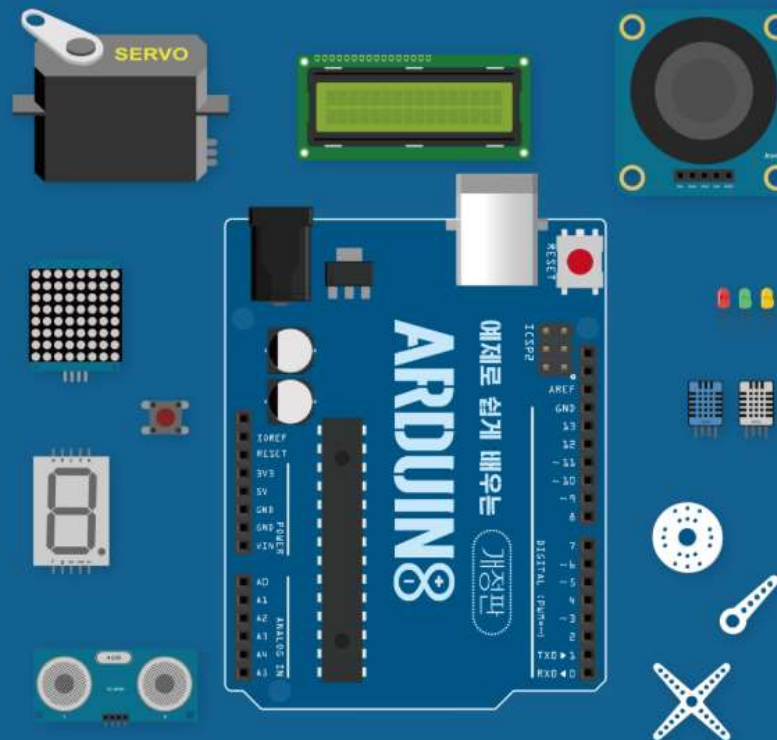
아두이노와 Node.js에 기반한

IOT 신호 시각화

| 저자 이 상 훈 |



인제대학교 출판부



예제로 쉽게 배우는

아두이노

개정판

장성용 · 김진환 지음

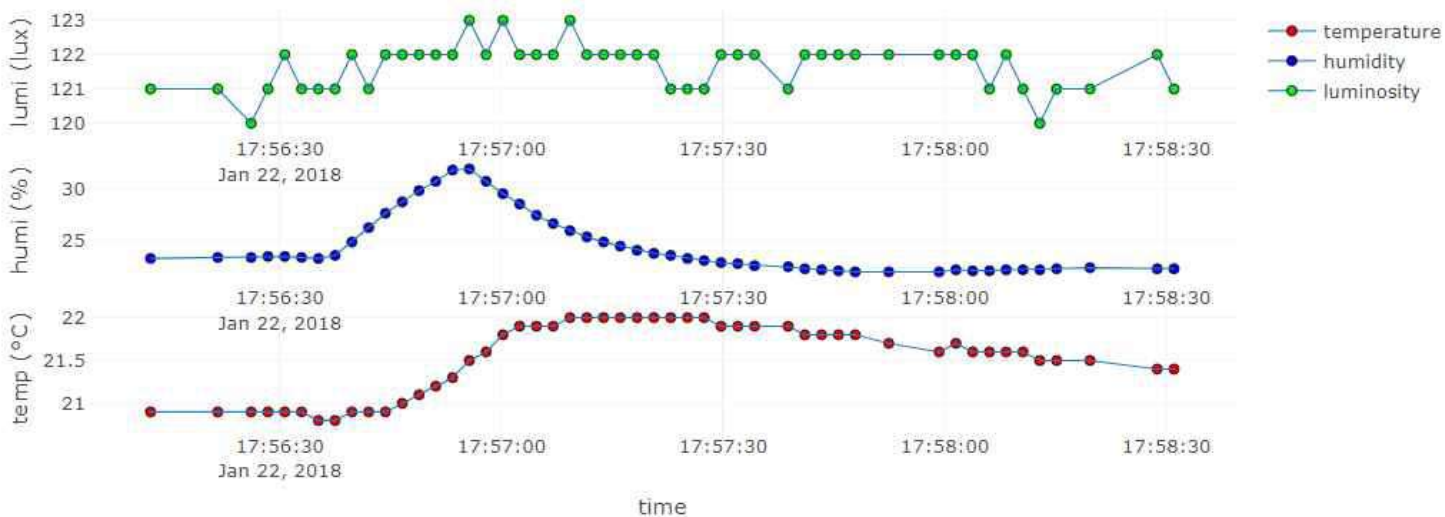
인제대학교 출판부

Target of this class

Real-time Weather Station from sensors



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



Another target of this class

PPG with rangeslider

