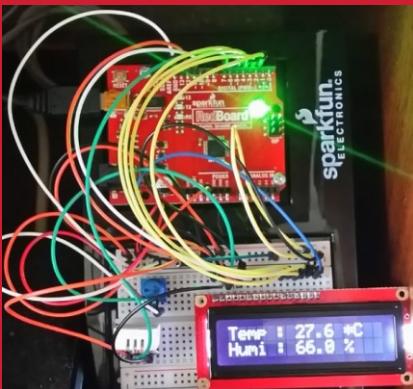


HW Programming

wk14 :

Arduino Coding VI.

Analog input project



Basic HW coding using Arduino and EV3 (RP3)

HIT, INJE University

1st semester, 2017

Email : yish@inje.ac.kr

Weekly schedule of HP– 1st semester, 2017



- **wk01 :**
- **wk02 :**
- **wk03 :**
- **wk04 :**
- **wk05 :**
- **wk06 : Introduction to class and enrollment in cyber class, and installing SW**
- **wk07 : Basic HW : Arduino I. – circuit**
- **wk08 : Mid-term exam. → Practice of various circuits**
- **wk09 : Basic HW : Arduino II. – coding by Arduino IDE**
- **wk10 : Arduino Coding II. – Serial monitor**
- **wk11 : Arduino Coding III. – LED**
- **wk12 : Arduino Coding IV. – LED & Digital input**
- **wk13 : Arduino Coding V. – Analog input & LCD**
- **wk14 : Arduino Coding VI. – Analog input & final project**
- **wk15 : Final exam.**



Arduino SW

fritzing.org Fritzing Fritzing

fritzing

electronics
made easy

Projects Parts Download Learning Services Contribute FORUM FAB

fritzing APP

Download the free Fritzing App and start building immediately!

Fritzing is an open-source hardware initiative that makes electronics accessible as a creative material for anyone. We offer a software tool, a community website and services in the spirit of Processing and Arduino, fostering a creative ecosystem that allows users to document their prototypes, share them with others, teach electronics in a classroom, and layout and manufacture professional pcbs.

Download and Start

Download our latest version 0.9.3b released on June 2, 2016 and start right away.

Produce your own board

With Fritzing Fab you can easily and inexpensively turn your circuit into a real, custom-made PCB. Try it out now!

Participate

Fritzing can only act as a creative platform if many

<http://fritzing.org/home/>



Arduino

Starter Kit



Arduino Starter Kit





Arduino Starter Kit : contents

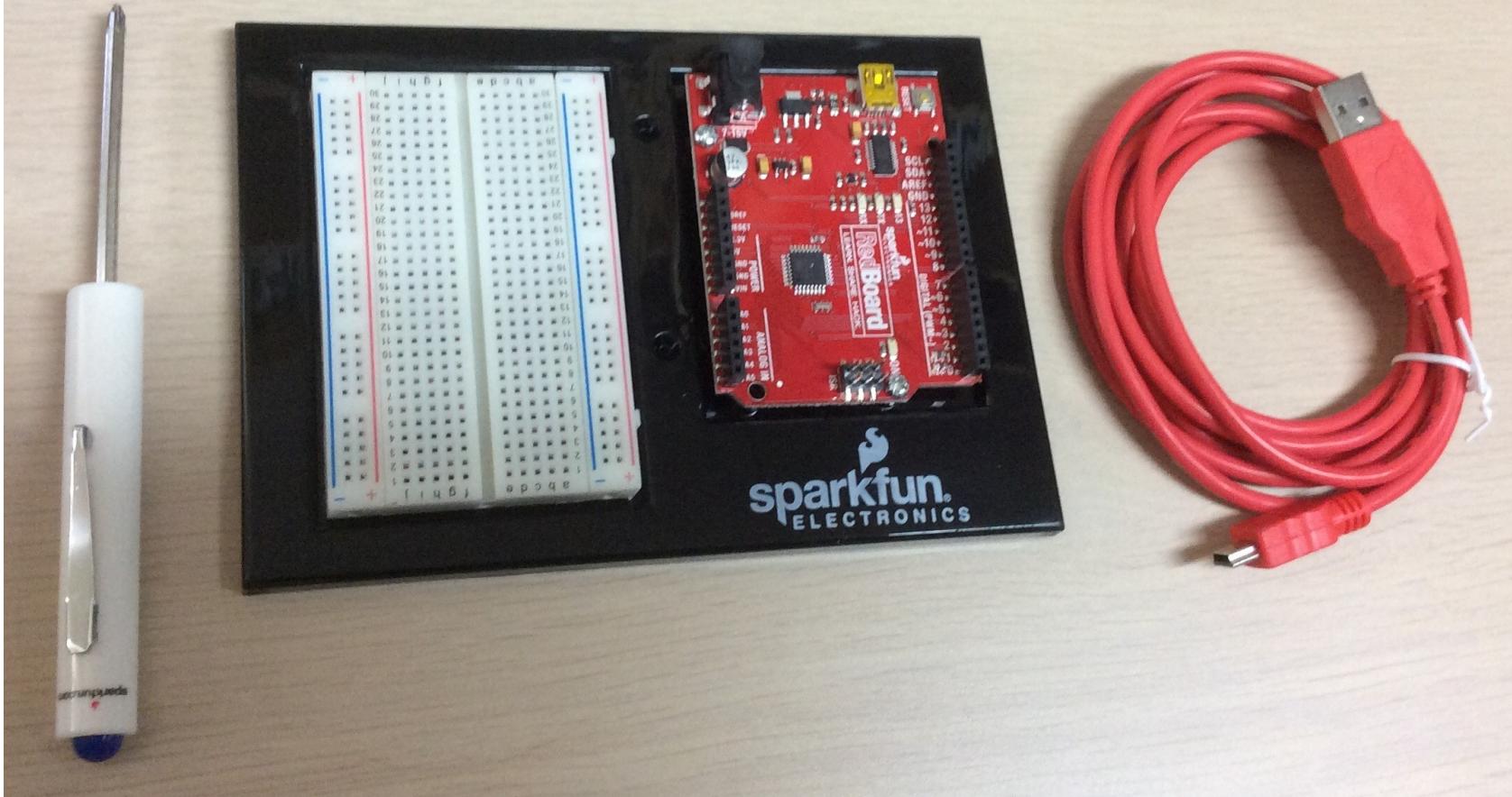
Arduino set

- **SparkFun 레드보드(아두이노 uno 호환)**
- 아두이노, 레드보드 고정판
- 한글판 가이드 북
- **400옴 브레드보드**
- 부품케이스
- **16x2 White on Black LCD**
- **74HC595** 쉬프트레지스터
- **2N2222** 트랜지스터
- **1N4148** 다이오드 (2)
- **기어달린 DC 모터**
- 서보모터
- **SPDT 5V 릴레이**
- **TMP36 온도센서**

- **플렉스 센서(Flex sensor)**
- **소프트 포텐시오미터(Softpot)**
- **SparkFun USB Cable**
- 점퍼케이블 (65)
- **Photocell**
- **RGB LED**
- **적색, 황색, 청색, 초록색 LED (5)**
- **10K 포텐시오미터 (가변저항, Trimpot)?**
- 피에조 부저
- **풀사이클 푸시버튼 (4)**
- **330, 10K 저항 (20)**



Arduino Starter Kit (Board setting)





1. Arduino SW: IDE



The screenshot shows the top navigation bar of the Arduino website. On the left is the Arduino logo. To its right are several menu items: HOME, BUY, SOFTWARE (which is highlighted in a darker teal color), PRODUCTS, LEARNING, FORUM, SUPPORT, and BLOG. The background of the header is a teal gradient.

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



1.5 Arduino IDE

sketch_may01a | 아두이노 1.8.2

파일 편집 스케치 툴 도움말

- 새 파일 Ctrl+N
- 열기... Ctrl+O
- 최근 파일 열기 >
- 스케치북 >
- 예제 >
- 닫기 Ctrl+W
- 저장 Ctrl+S
- 다른 이름으로 저장... Ctrl+Shift+S
- 페이지 설정 Ctrl+Shift+P
- 인쇄 Ctrl+P
- 환경설정 Ctrl+Comma
- 종료 Ctrl+Q

환경설정

설정 네트워크

스케치북 위치: C:\Users\yish-HC\Documents\Arduino

에디터 언어: 시스템 기본설정 (마두이노를 재시작해야 함)

에디터 글꼴 크기: 14

Interface scale: 자동 100 % (마두이노를 재시작해야 함)

다음 동작중 자세한 출력 보기: 컴파일 업로드

컴파일러 경고: None

줄 번호 표시

코드 풀딩 사용하기

업로드 후 코드 확인하기

외부 에디터 사용

Aggressively cache compiled core

시작시 업데이트 확인

스케치 파일을 저장할 때 새로운 확장자(.pde -> .ino)로 업데이트

검증 또는 업로드 할 때 저장하기

추가적인 보드 매니저 URLs

추가적인 환경 설정은 파일에서 직접 편집할 수 있습니다
C:\Users\yish-HC\AppData\Local\Arduino15\preferences.txt
(마두이노가 실행되지 않는 경우에만 수정 가능)

확인 취소

Arduino coding VI.



1. **Arduino SW : IDE**
2. **Blink a LED**
3. **Serial monitor**
4. **LED Control**
5. **Digital input: Switch**
6. **Analog input: Sensors**
7. **Project : Sensors & Motor**



5. Digital input



5. 디지털 신호 입력

디지털 신호

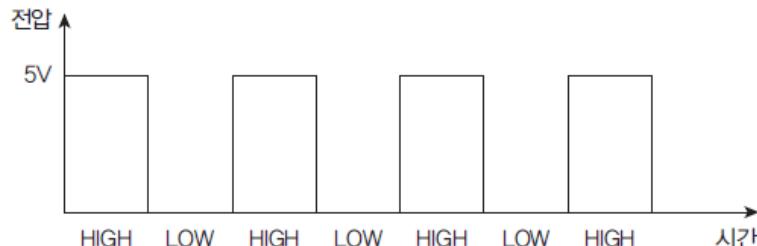


그림 5.1 HIGH LOW 신호가 반복되는 디지털신호

풀업 / 풀다운

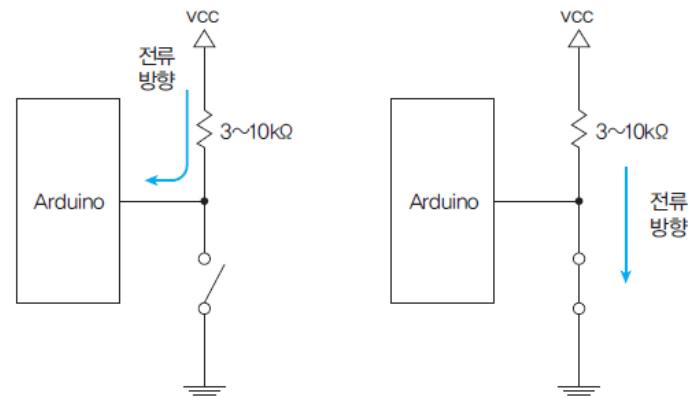


그림 5.2 풀업저항과 스위치 동작. Arduino는 스위치가 오프상태인 (a)에서는 HIGH 신호로, 스위치가 온상태인 (b)에서는 LOW 신호로 인식한다.

- ✓ 0과 1 혹은 High Low 두 가지 값으로 표현되는 신호
- ✓ 잡음에 강하고 데이터의 저장 및 처리가 용이

- ✓ 디지털 신호 입력핀에 아무것도 연결되지 않았을 때의 상태를 High 혹은 Low 신호로 만들어 안정시킴



5.2 디지털 신호 입력 – 안정된 Switch

DIY 8. 현재의 LED 상태 (LOW or HIGH)와 스위치 누른 횟수를 아래와 같이 출력하는 두 code를 비교하시오.

COM6

```
LED: LOW, count: 1
LED: HIGH, count: 2
LED: LOW, count: 3
LED: HIGH, count: 4
LED: LOW, count: 5
LED: HIGH, count: 6
LED: LOW, count: 7
LED: HIGH, count: 8
LED: LOW, count: 9
LED: HIGH, count: 10
```

```
25 void loop(){
26     int swInput = digitalRead(inputPin); // 스위치 입력을 받는다
27     int ledOutput = digitalRead(ledPin); // LED의 출력 상태를 확인한다
28
29     if (swInput == LOW){
30         delay(400);
31         if (ledOutput) {
32             Serial.print("LED: ");
33             Serial.print(led$[ledOutput]); // LED 점등 표시
34             delay(100);
35             digitalWrite(ledPin, LOW); // LED가 점등되어 있으면 소등
36         }
37         else {
38             Serial.print("LED: ");
39             Serial.print(led$[ledOutput]); // LED 소등 표시
40             delay(100);
41             digitalWrite(ledPin, HIGH); // LED가 소등되어 있으면 점등
42         }
43         ++count;
44         // 스위치 입력 횟수를シリ얼 통신으로 전송한다.
45         Serial.print(", count: ");
46         Serial.println(count);
47     }
48 }
```

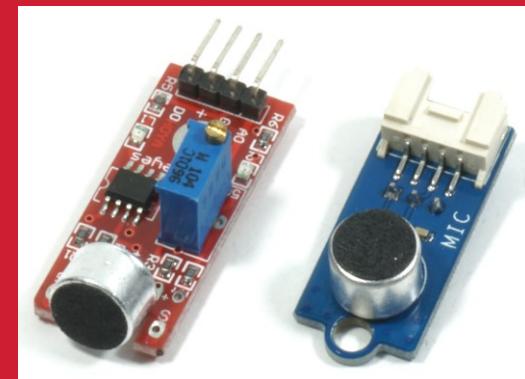
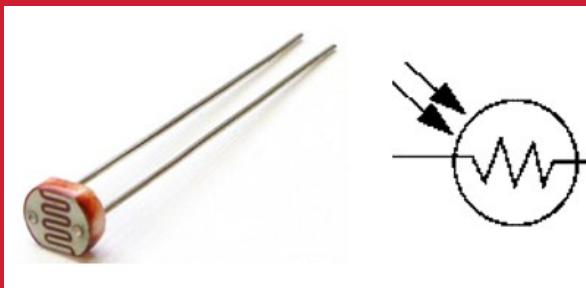
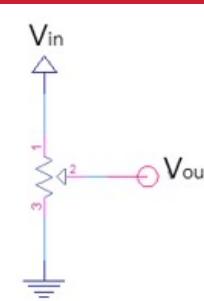
```
// 2번 핀을 스위치 입력으로 설정
const int inputPin = 2;
int count = 0;
String stringValue[] = {"LOW", "HIGH"};
void setup() {
    // Arduino 13번 핀에 내장된 LED를 출력으로 설정한다.
    pinMode(ledPin, OUTPUT);
    // 스위치 입력을 위하여 2번핀을 입력으로 설정하고 풀업시킨다.
    pinMode(inputPin, INPUT_PULLUP);
    Serial.begin(9600);
}

void loop() {
    // 스위치 입력을 받는다
    int swInput = digitalRead(inputPin);
    int ledOutput = digitalRead(ledPin);

    // 스위치가 눌렸을 때
    if (swInput == LOW) {
        delay(200);
        if (ledOutput) {
            digitalWrite(ledPin, LOW);
        }
        else {
            digitalWrite(ledPin, HIGH);
        }
        count++;
        Serial.print("LED : ");
        Serial.print(stringValue[ledOutput]);
        Serial.print(", Count: ");
        Serial.println(count);
    }
}
```



6. Analog input





6. 아날로그 신호 입력

6.1 포텐시미터 입력 (가변저항기)

* LCD 회로

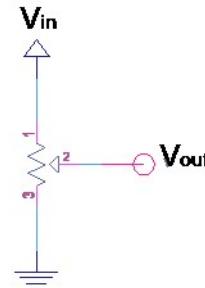
6.2 빛 입력

6.3 모터 제어

* Final Project

6.1 포텐쇼미터 (가변저항 조절)

포텐쇼미터 (Potentiometer)



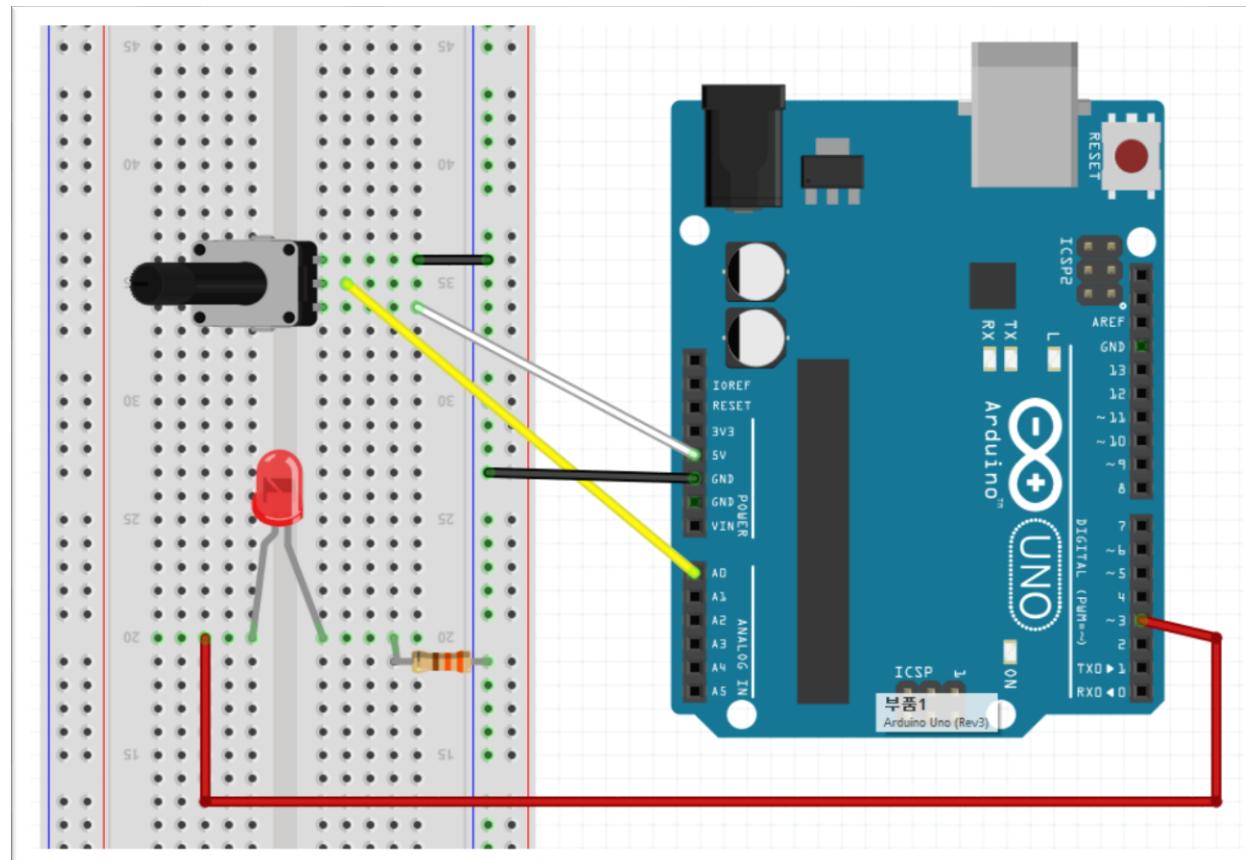
- ✓ 회전, 직선 변위를 감지하는 센서
- ✓ 위치에 따라 저항 값이 변화함. (가변저항기)
- ✓ ADC를 이용하여 변화된 저항에 전압을 인가하여 전압의 변화를 감지



6.1 포텐시미터 (가변저항 조절)

응용 문제 1. delay함수를 사용하지 말고 4.2절의 예제를 참고하여 PWM 단자를 이용하여 LED의 밝기를 조절해 보자.

- PWM을 지원하는 디지털 3 번 핀에 단색 LED를 연결. (330 ohm 저항 연결)





6.1 포텐쇼미터 (code-2, pwm)

```

6 // 0번 아날로그핀을 포텐쇼미터 입력으로 설정한다.
7 const int potentiometerPin = 0;
8
9 // 13번 핀에 연결되어 있는 내장 LED를 출력으로 사용한다.
10 const int ledPin = 13;
11
12 // #3 pin is defined to PWM output pin
13 const int pwmOutputPin = 3;
14
15 void setup() {
16 // 13번 핀을 출력으로 설정한다.
17 pinMode(ledPin, OUTPUT);
18 // 시리얼 통신을 설정한다.
19 Serial.begin(9600);
20 }

```

COM6

ADC Value: 1023. Duty cycle: 100%. pwm: 255
 ADC Value: 1022. Duty cycle: 99%. pwm: 254
 ADC Value: 1023. Duty cycle: 100%. pwm: 255

결과 화면 캡처: HPnn_pwm.png 로 저장...

```

22 void loop(){
23     int adcValue; // 실제 센서로부터 읽은 값 (0~1023)
24     int duty; // LED 점멸 주기 (0~100%)
25     int pwm; // pwm 출력용
26
27 // 포텐쇼미터 값을 읽는다.
28     adcValue = analogRead(potentiometerPin);
29 // 포텐쇼미터 값을 0~100의 범위로 변경한다.
30     duty = map(adcValue, 0, 1023, 0, 100);
31
32 // LED를 duty ms 만큼 점등한다.
33     digitalWrite(ledPin, HIGH);
34     delay(duty);
35 // 나머지 시간에는 소등시킨다.
36     digitalWrite(ledPin, LOW);
37     delay(100-duty);
38
39 // pwmOutputPin Led ON
40     pwm = map(adcValue, 0, 1023, 0, 255);
41     analogWrite(pwmOutputPin, pwm);
42 // 시리얼 통신으로 ADC 값과 Duty를 출력한다.
43     Serial.print("ADC Value: ");
44     Serial.print(adcValue);
45     Serial.print(". Duty cycle: ");
46     Serial.print(duty);
47     Serial.println("%");
48 }

```



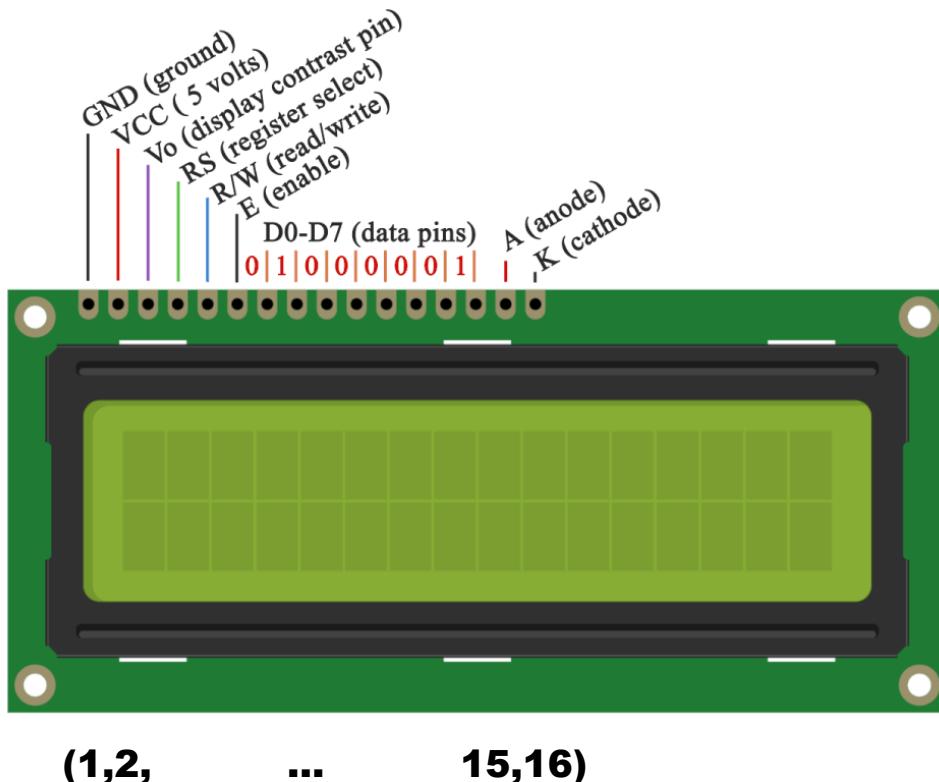
Introduction to LCD





Introduction to LCD

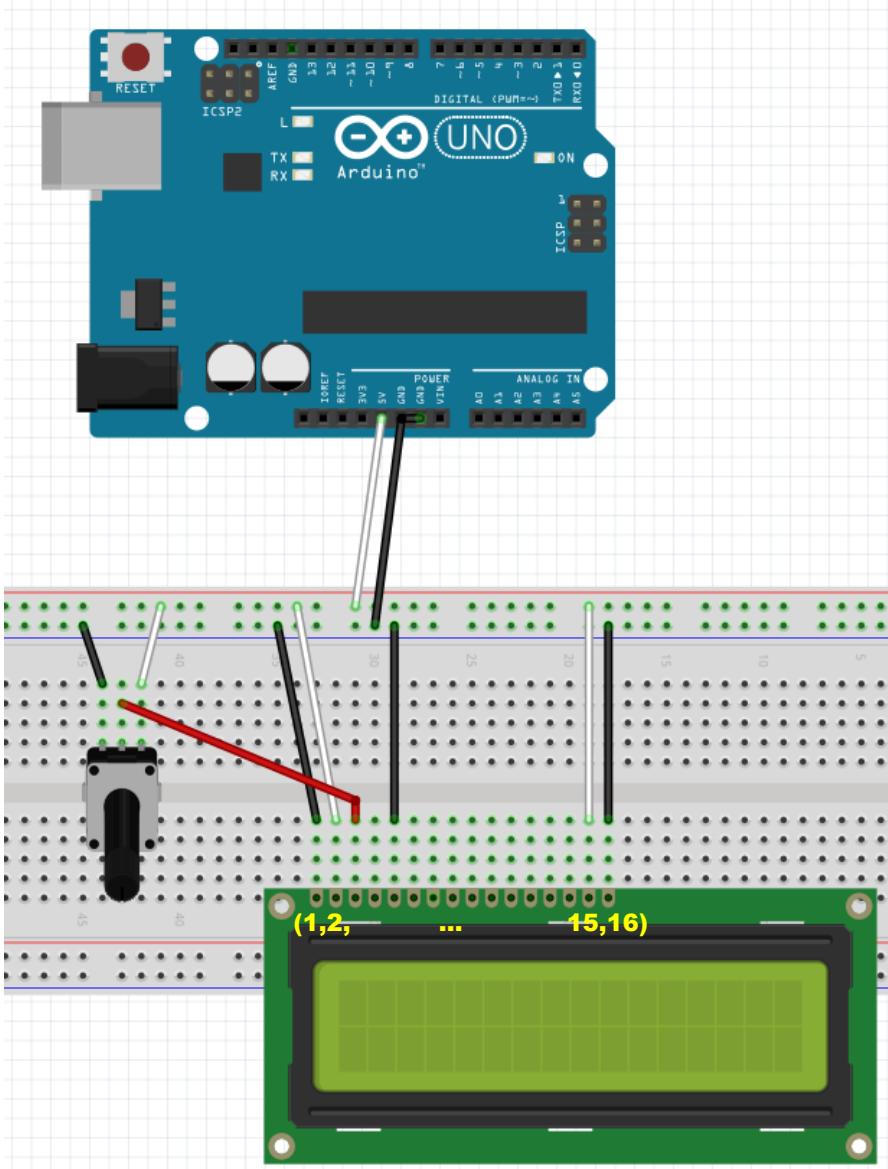
LCD (Liquid Crystal Display, 16 X 2)



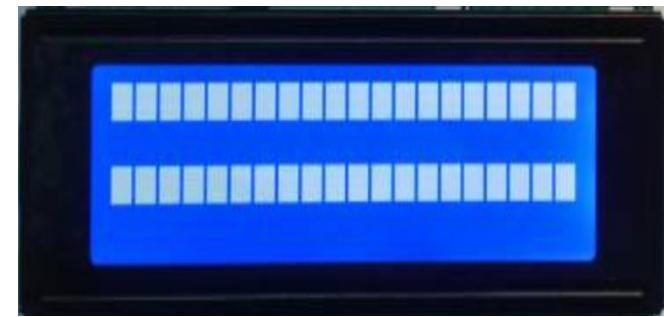
- Pin 1 to Arduino GND
- Pin 2 to Arduino 5V
- Pin 3 to wiper
- Pin 4 to Arduino pin 12
- Pin 5 to Arduino GND
- Pin 6 to Arduino pin 11
- Pin 11 to Arduino pin 5
- Pin 12 to Arduino pin 4
- Pin 13 to Arduino pin 3
- Pin 14 to Arduino pin 2
- Pin 15 to +5V (with 220 or 330 Ω)
- Pin 16 to GND



Introduction to LCD (LCD 초기화)

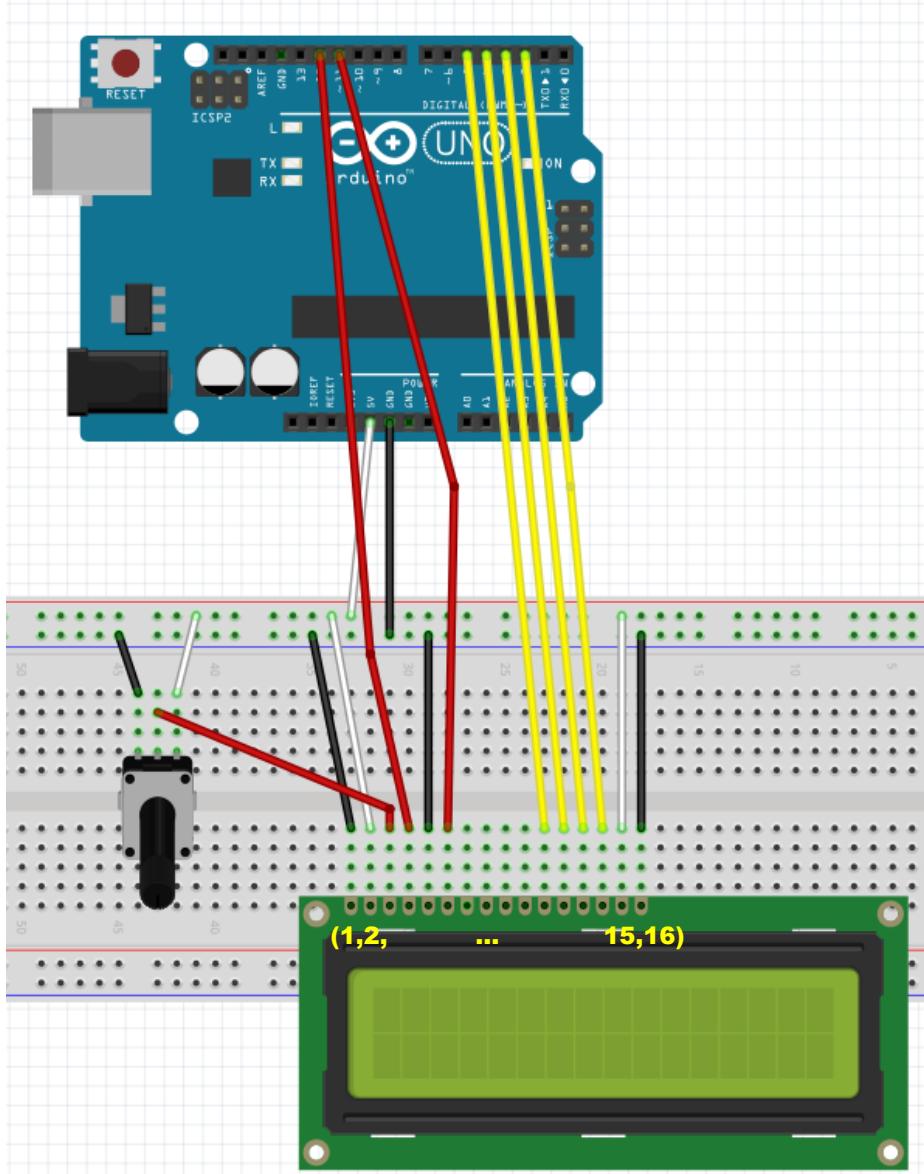


전원 연결 후
LCD 초기화





Introduction to LCD (데이터 입력)



Pin 1 to Arduino GND

Pin 2 to Arduino 5V

Pin 3 to wiper

Pin 4 to Arduino pin 12

Pin 5 to Arduino GND

Pin 6 to Arduino pin 11

Pin 11 to Arduino pin 5

Pin 12 to Arduino pin 4

Pin 13 to Arduino pin 3

Pin 14 to Arduino pin 2

Pin 15 to +5V

Pin 16 to GND

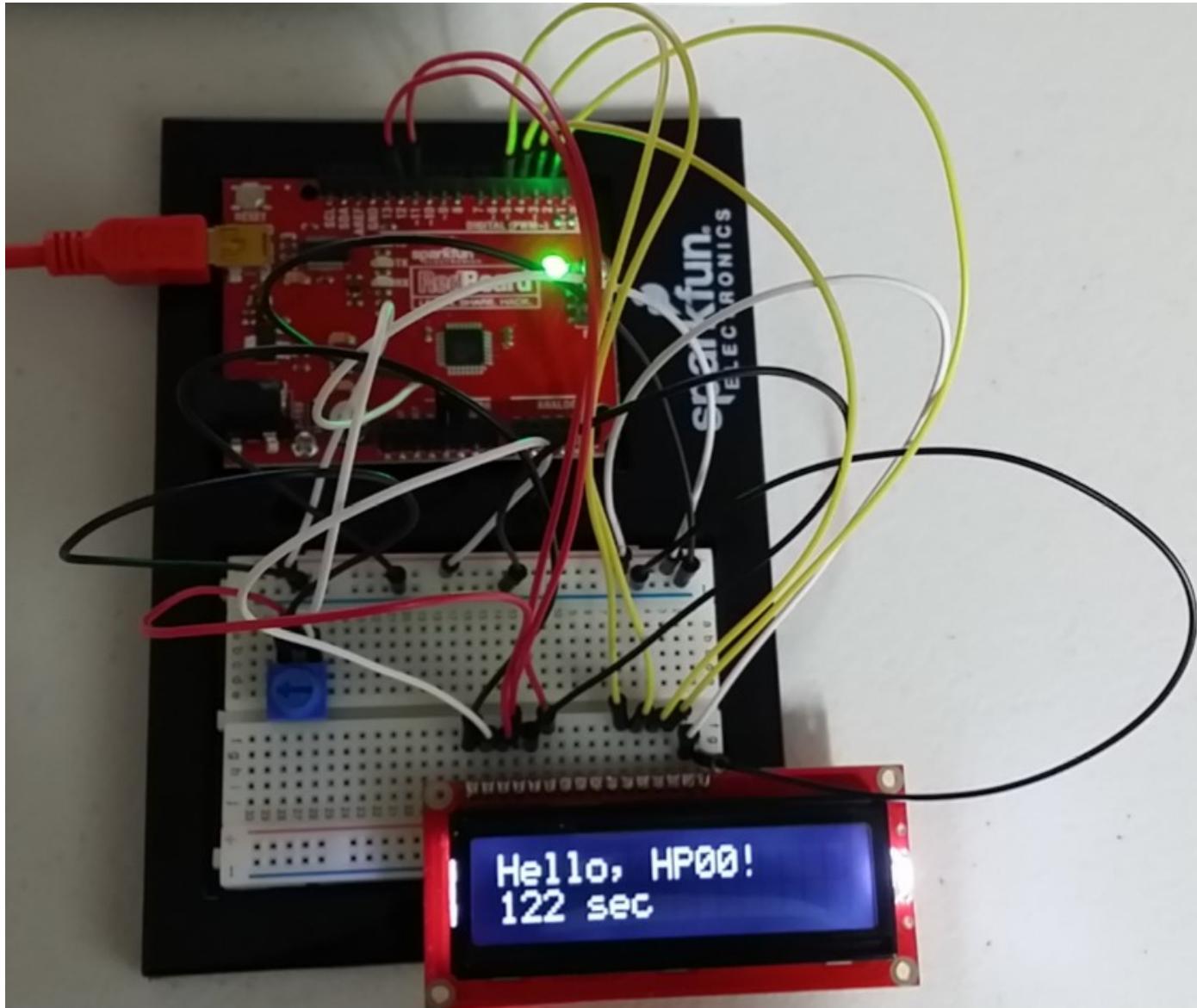


Introduction to LCD – code “Hello HP”

```
40 // include the library code:  
41 #include <LiquidCrystal.h>  
42  
43 // initialize the library with the numbers of the interface pins  
44 LiquidCrystal lcd(12, 11, 5, 4, 3, 2);  
45  
46 void setup() {  
47     // set up the LCD's number of columns and rows:  
48     lcd.begin(16, 2);  
49     // Print a message to the LCD.  
50     lcd.print("Hello, HP00!");  
51 }  
52  
53 void loop() {  
54     // set the cursor to column 0, line 1  
55     // (note: line 1 is the second row, since counting begins with 0):  
56     lcd.setCursor(0, 1);  
57     // print the number of seconds since reset:  
58     lcd.print(millis() / 1000);  
59 }
```



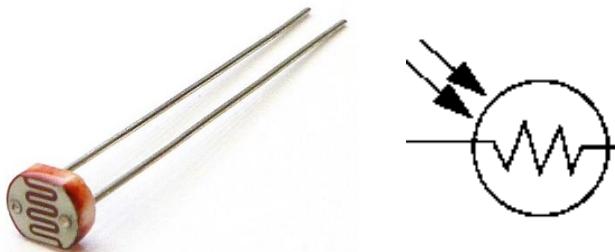
Introduction to LCD



결과 화면 촬영: **HPnn_Hello.png** 로 저장...

6.2 조도 센서 (빛의 양 측정)

CdS 센서



- ✓ 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 문서](#)를 참조하십시오.

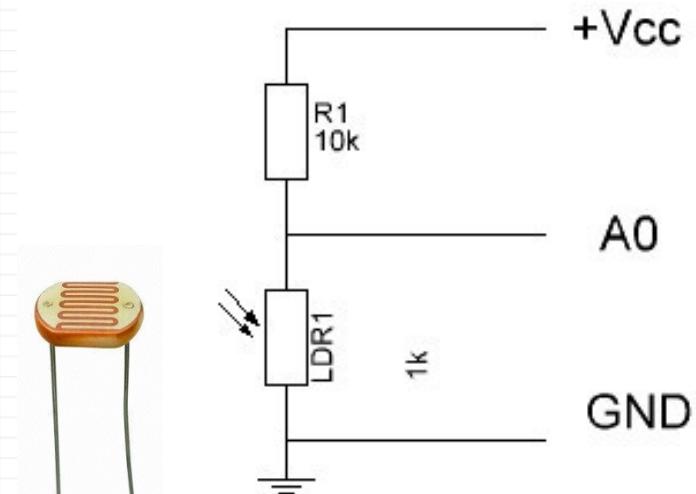
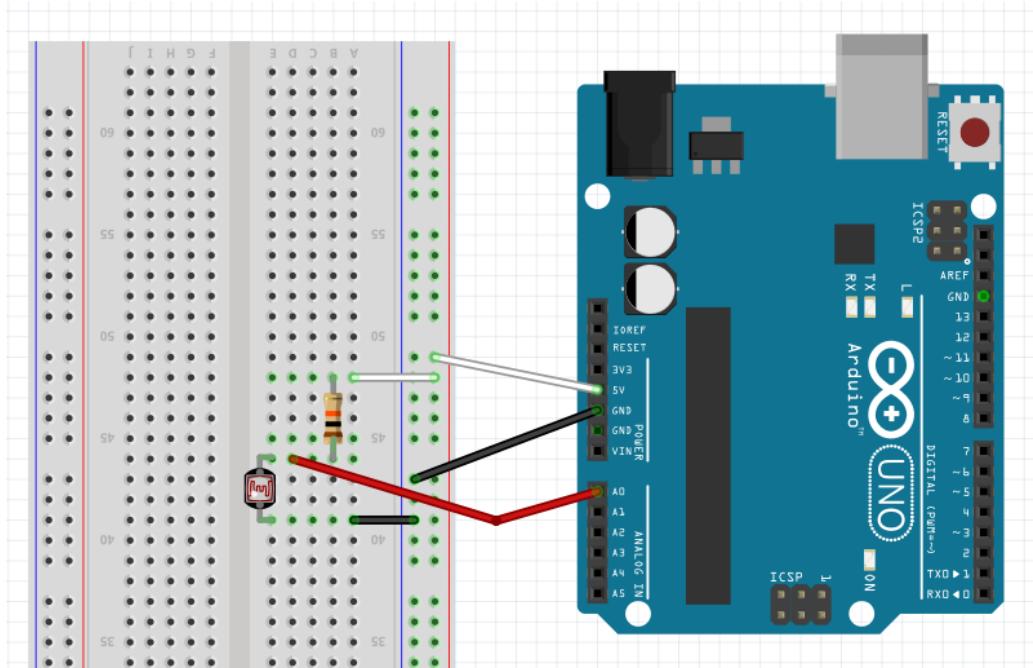
럭스(lux, 기호 lx)는 빛의 조명도를 나타내는 SI 단위이다. 럭스는 루멘에서 유도
 $1 \text{ lx} = 1 \text{ lm/m}^2 = 1 \text{ cd}\cdot\text{sr}\cdot\text{m}^{-2}$

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	대기광이 있는 달 없는 맑은 밤 하늘 ^[1]
0.01 lux	초승달
0.27 lux	맑은 밤의 보름달 ^{[1][2]}
1 lux	절대 위도를 덮은 보름달 ^[3]
3.4 lux	맑은 하늘 아래의 어두운 황혼 ^[4]
50 lux	거실 ^[5]
80 lux	북도/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

6.2 조도 센서 (빛의 양 측정)

CdS 센서 회로



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

저항과 광센서 사이에서 전압 값을 A0로 측정



조도 센서 (빛의 양 측정)

EX 6.2

빛 입력 (1/3)

실습목표 CdS 셀을 이용하여 조도를 측정해 보자.

1. CdS 셀로 측정된 조도를 아날로그 핀을 통하여 0~1023 범위로 읽는다.
2. ADC 값을 시리얼모니터로 0~100%의 범위로 출력한다. (빛의 밝기가 아니고 단지 밝기 비율)
 $(0 \sim 1023) \rightarrow (100 \sim 0)\%$
3. ADC 값을 시리얼모니터로 lux 값으로 출력한다.

Hardware

1. CdS셀과 10k Ω 저항을 연결한 뒤 저항의 한쪽 끝은 5V에 CdS셀의 한쪽 끝은 GND에 연결한다.
2. 저항과 CdS셀 사이를 아날로그입력핀 A0에 연결한다.



6.2 조도 센서 (빛의 양 측정)

CdS 센서 회로 – 측정 1.

Light_start.ino §

```
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 }
```

COM4

233

234

235

237

235

235

236

241

386

975

965

964

964

967

어두우면 측정 값이 커지고 밝을수록 값이 작아진다 ???



6.2 조도 센서 (빛의 양 측정)

CdS 센서 회로 – 측정 2.

```
Light_final $  
1 // lux  
2 #define CDS_INPUT 0  
3 void setup() {  
4 Serial.begin(9600);  
5 }  
6 void loop() {  
7 int value = analogRead(CDS_INPUT);  
8 Serial.println(int(luminosity(value)));  
9 delay(1000);  
10 }  
11  
12 //Voltage to LuxLux  
13 double luminosity (int RawADC0){  
14 double Vout=RawADC0*0.0048828125; // 5/1024 (Vin = 5 V)  
15 int lux=(2500/Vout-500)/10; // lux = 500 / Rldr, Vout = Ildr*Rldr = (5/(10 + Rldr))*Rldr  
16 return lux;  
17 }
```

COM4

어	두	율
1		
1		
2		
1		
1		
0		
72		
166		
167		
168		
167		
167		
166		
166		

밝
을

때

밝을수록 측정 값이 커지고 어두울수록 값이 작아진다 !!!

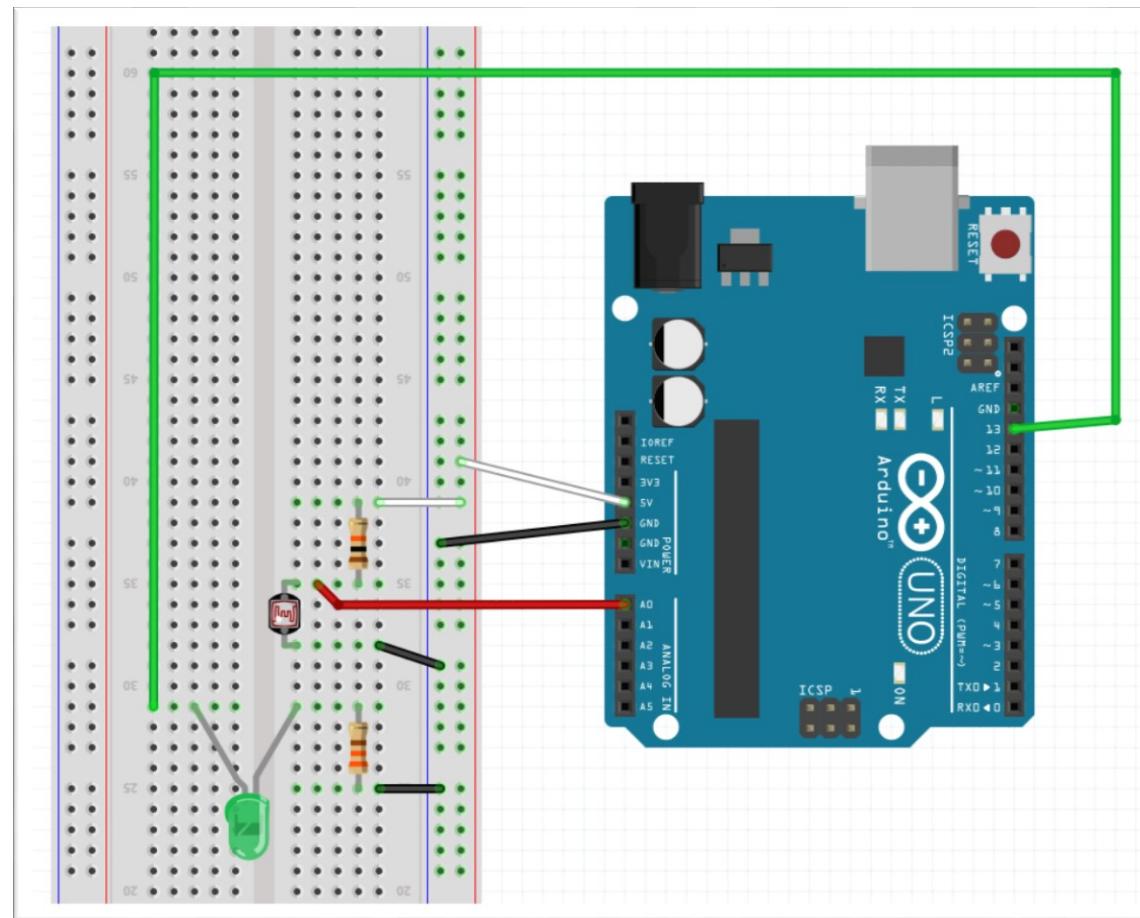


6.2 조도 센서 (빛의 양 측정 응용)

DIY

조도 값에 따라 LED를 켜고 끄는 코드를 만드시오.

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





6.2 조도 센서 (빛의 양 측정 응용)

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 }
```

6.3 서보 모터 구동

서보모터 (Servo Motor)

- ✓ 기계적인 위치, 속도, 가속도 등을 제어하는 모터
- ✓ 산업용 서보모터는 로봇의 관절, 공작 기계의 위치제어 등에 사용
- ✓ RC용 서보모터는 RC 자동차나 RC 비행기에 사용



그림 7.3 산업용 서보모터(a)와 RC용 서보모터(b)



그림 7.4 실험에 사용할 소형 RC 서보모터 (SG90)

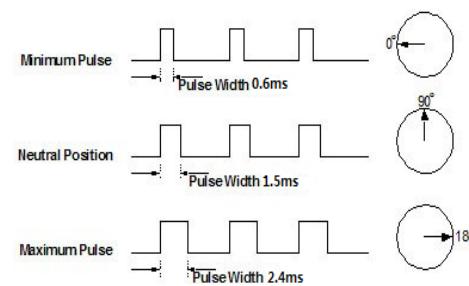


그림 7.5 PWM 신호와 RC 서보모터의 외전각

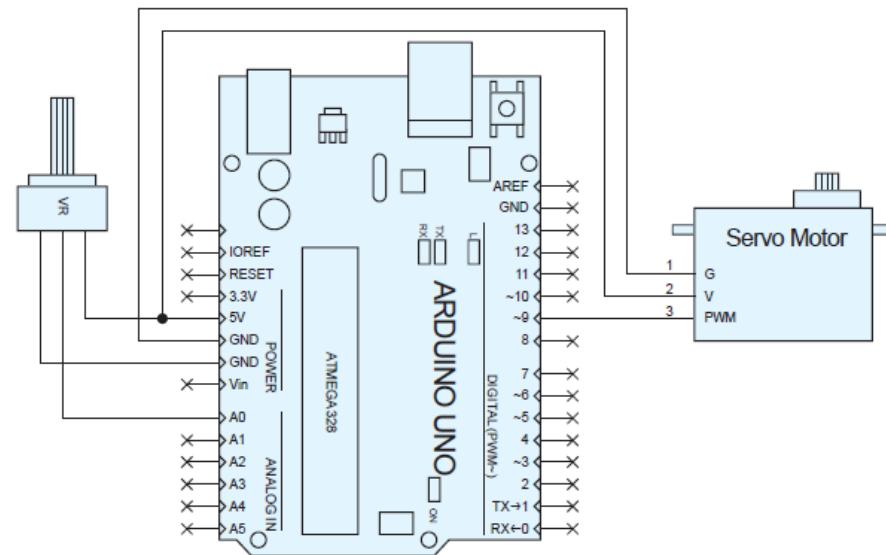
6.3 서보 모터 구동

EX 7.2

서보모터 구동 (1/3)

실습목표

1. 소형 RC용 서보모터를 구동한다.
2. 포텐시미터의 각도에 따라서 서보모터의 각도를 조절한다.
3. 현재 각도를 시리얼 통신으로 전송한다.





6.3 서보 모터 구동

EX 7.2

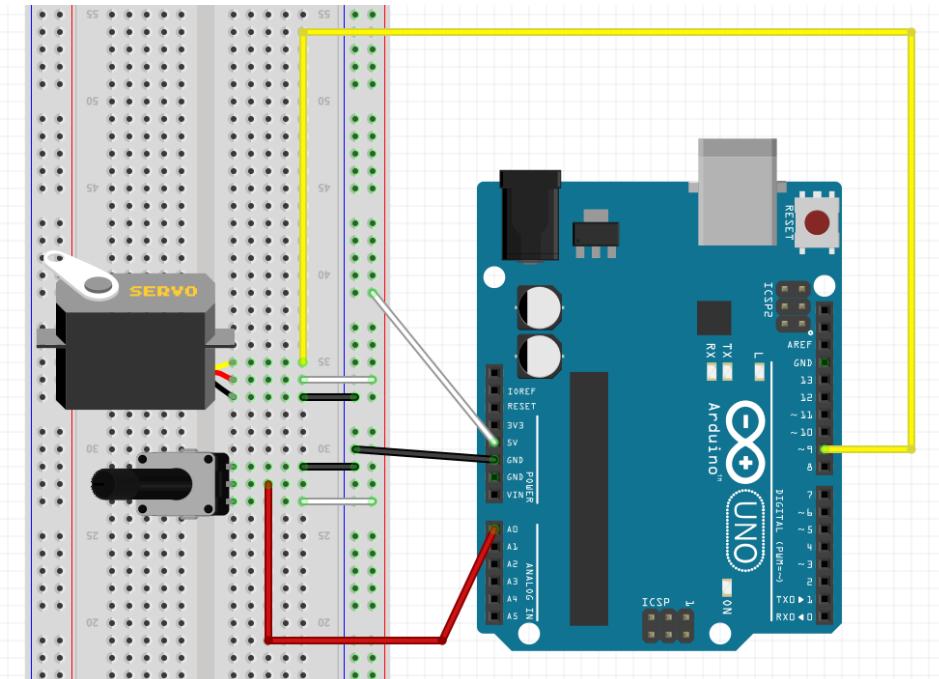
서보모터 구동 (회로)

Hardware 1. 포텐시미터의 1, 3번핀을 Arduino의 5V, GND에 연결한다.

2. 포텐시미터의 2번핀을 Arduino의 아날로그입력핀 A0에 연결한다.

3. 서보모터의 V(적색)와 GND(검정 혹은 갈색)핀을 Arduino의 5V와 GND에 연결한다.

4. Arduino의 9번핀을 서보모터의 PWM핀(노랑, 주황)과 연결한다.



PWM=Orange (노랑)
Vcc=Red (+)
Ground=Brown (-)



6.3 서보 모터 구동

EX 7.2

서보모터 구동 (2/3)

Commands

- 서보모터이름.attach(핀번호)

이름을 설정한 서보모터를 핀번호에 설정한다.

- 서보모터이름.attach(핀번호, 최소펄스, 최대펄스)

이름을 설정한 서보모터를 핀번호에 설정한다.

서보모터가 동작하는 최소 펄스와 최대 펄스를 **マイ크로세컨드(μs)** 단위로 설정한다.

- 서보모터이름.write(각도)

이름을 설정한 서보모터를 정해진 각도로 위치시킨다.



6.3 서보 모터 구동

EX 7.2

서보모터 구동 (3/3)

Sketch 구성

1. 서보모터 라이브러리를 추가한다.
2. 서보모터 이름을 설정하고 9번핀을 서보모터 출력으로 사용한다.
3. 서보모터의 회점위를 설정한다 라이브러리의 attach()함수를 사용한다.
4. 포텐쇼미터값을 아날로그 0번핀으로 입력받아 0~1023 범위의 포텐쇼미터값 범위를 0~180도로 환산하여 서보모터를 동작시킨다.

실습 결과

1. 포텐쇼미터의 회전각에 따라 서보모터가 회전한다.
2. 포텐쇼미터 값이 변화하면 시리얼 통신으로 회전각을 전송한다.



6.3 서보 모터 구동

Code & result

ex_7_2_start.ino

```

6 // 서보모터 라이브러리 불러오기
7 #include <Servo.h>
8
9 // 서보모터 이름 설정
10 Servo motor1;
11
12 // 서보 모터 신호핀 설정
13 int servoMotorPin = 9;
14
15 // 포텐시미터 핀 설정
16 int potentiometerPin = 0;
17
18 // 모터 각도 변수 설정
19 int motorAngle;
20 int motorAngleOld;
21
22 void setup() {
23
24 // 서보모터 설정. 0.6ms 부터 2.4ms 범위로 설정
25 motor1.attach(servoMotorPin,600,2400);
26 // 시리얼 통신 설정
27 Serial.begin(9600);
28 }
```

```

30 void loop(){
31 // 포텐시미터 값을 읽어옴
32 int potentiometer = analogRead(potentiometerPin);
33
34 // 포텐시미터 값을 모터 각도로 변환한다
35 motorAngle = map(potentiometer,0,1023,0,180);
36
37 // 모터에 각도값을 전달한다
38 motor1.write(motorAngle);
39
40 // 이전각도와 현재 각도가 같지 않으면 시리얼 모니터에
41 // 각도를 출력한다.
42 if(motorAngle != motorAngleOld){
43 Serial.print("Servo Motor Angle is: ");
44 Serial.println(motorAngle);
45 }
46
47 // 현재의 모터 각도를 저장한다.
48 motorAngleOld = motorAngle;
49
50 delay(20);
51 }
```

COM4

Servo Motor Angle is: 180
 Servo Motor Angle is: 179
 Servo Motor Angle is: 160
 Servo Motor Angle is: 139
 Servo Motor Angle is: 120
 Servo Motor Angle is: 109
 Servo Motor Angle is: 103
 Servo Motor Angle is: 87
 Servo Motor Angle is: 76
 Servo Motor Angle is: 37
 Servo Motor Angle is: 5
 Servo Motor Angle is: 0

결과 화면 :

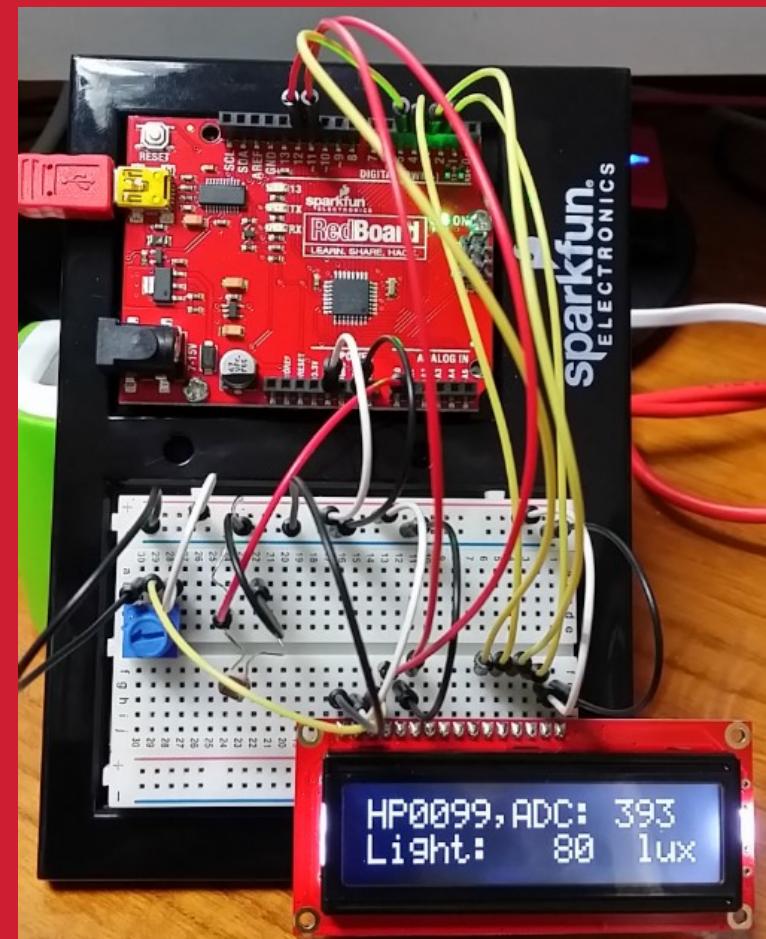
HPnn_servo.png
로 저장....

Delay를 조정해서 각도 출력



7. Project

LCD에 조도 값을
표시하면서
조도에 따라 LED를
ON/OFF





조도 센서 (빛의 양 측정)

EX 6.2

빛 입력 (1/3)

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

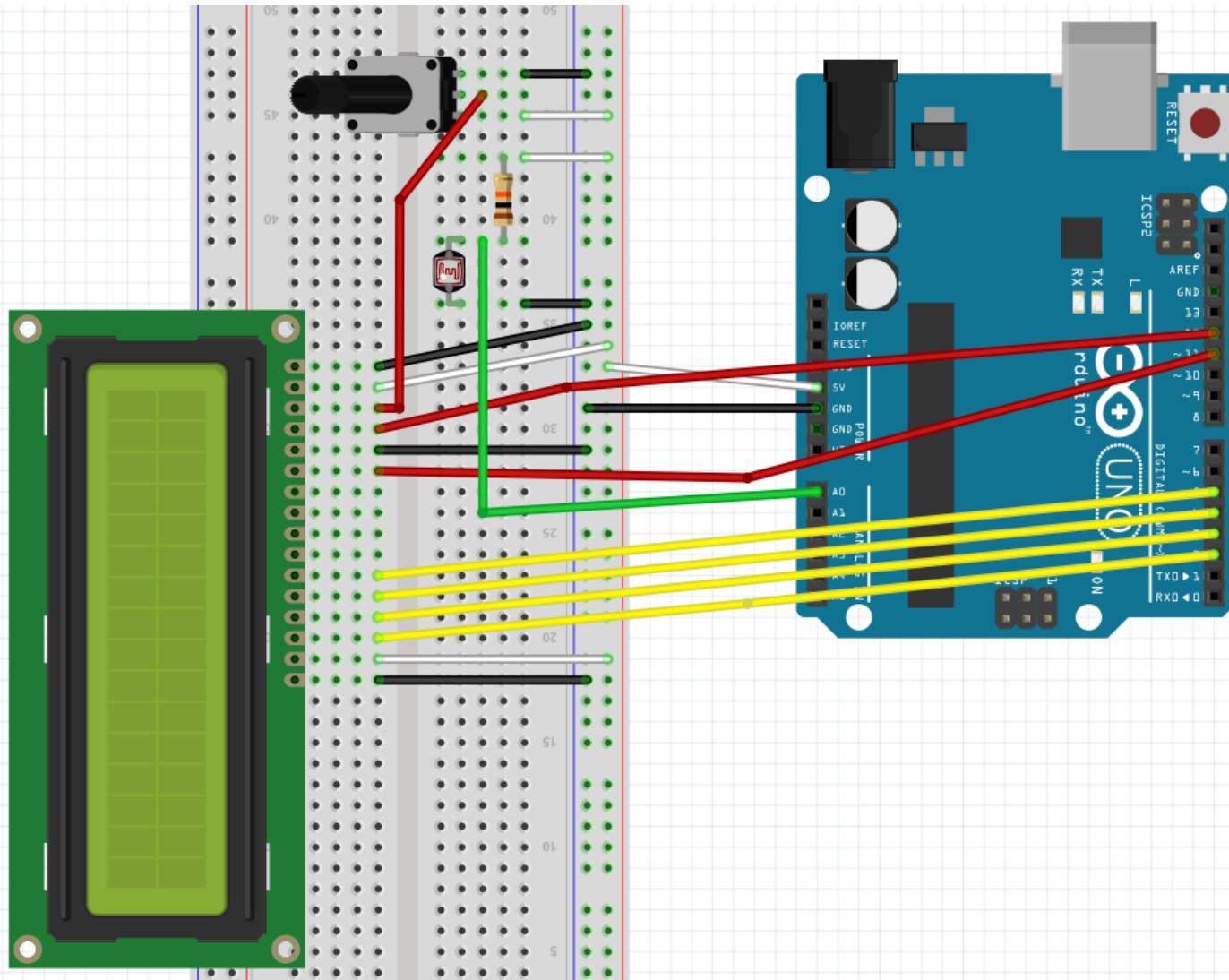
1. LCD를 연결한다.
2. CdS셀과 $10k\Omega$ 저항을 연결한 뒤 저항의 한쪽 끝은 5V에 CdS셀의 한쪽 끝은 GND에 연결한다.
3. 저항과 CdS셀 사이를 아날로그입력핀 A0에 연결한다.
4. 단색 LED를 $330\ \Omega$ 저항을 연결해서 디지털입력핀 D13과 GND에 연결한다.



조도 센서 (빛의 양 측정)

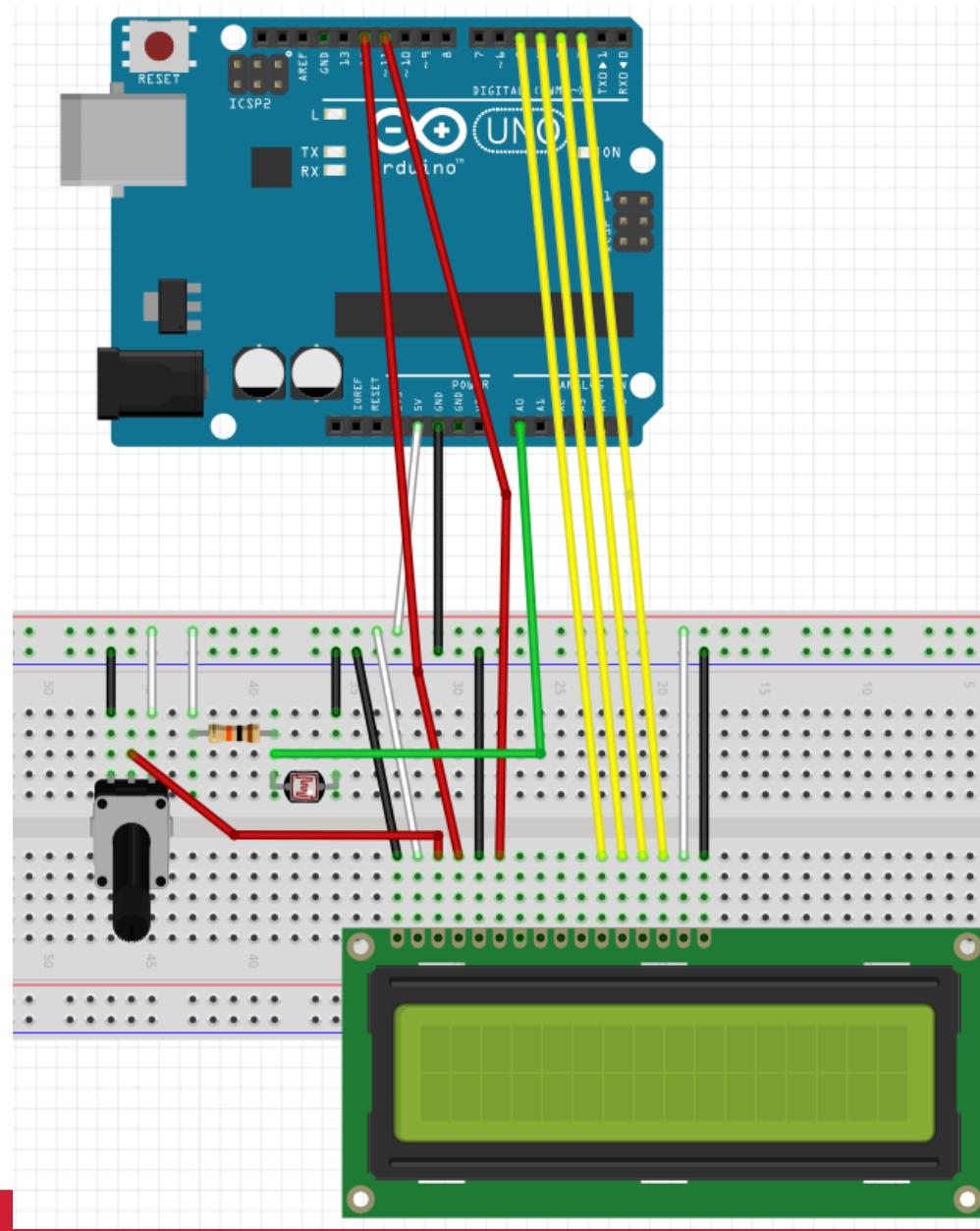
EX 6.2

Fritzing circuit





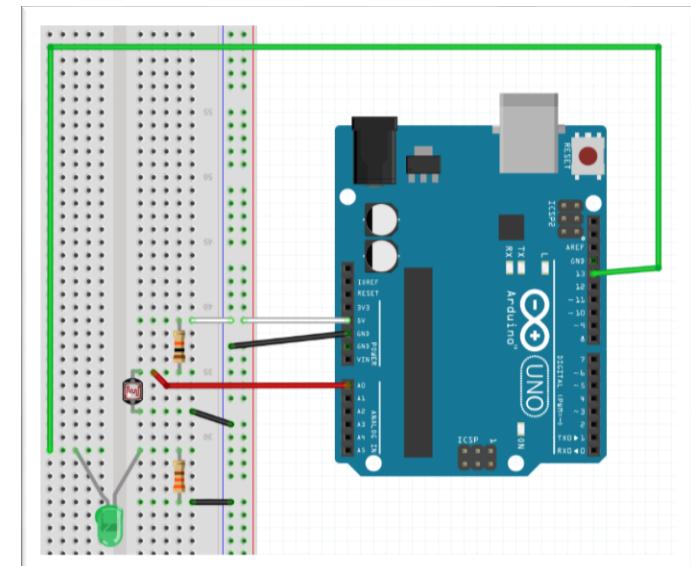
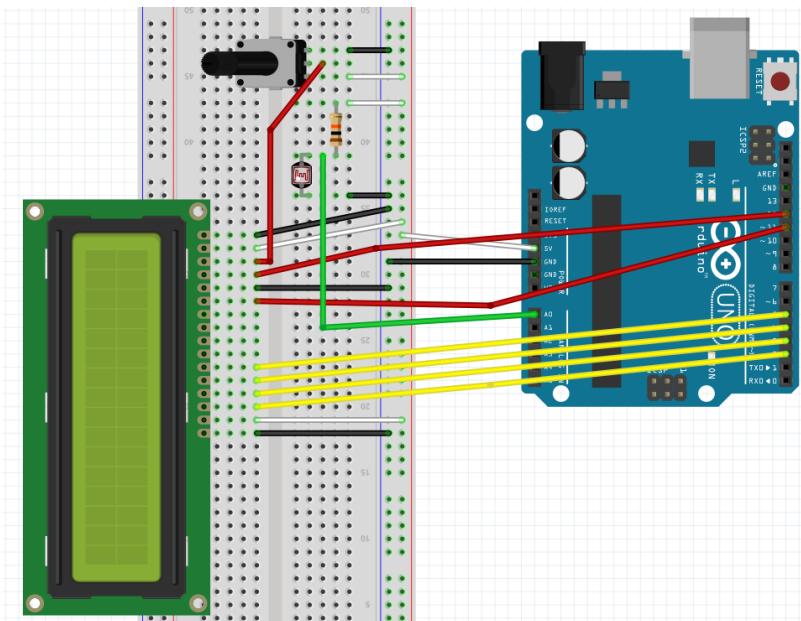
조도 센서 (빛의 양 측정)





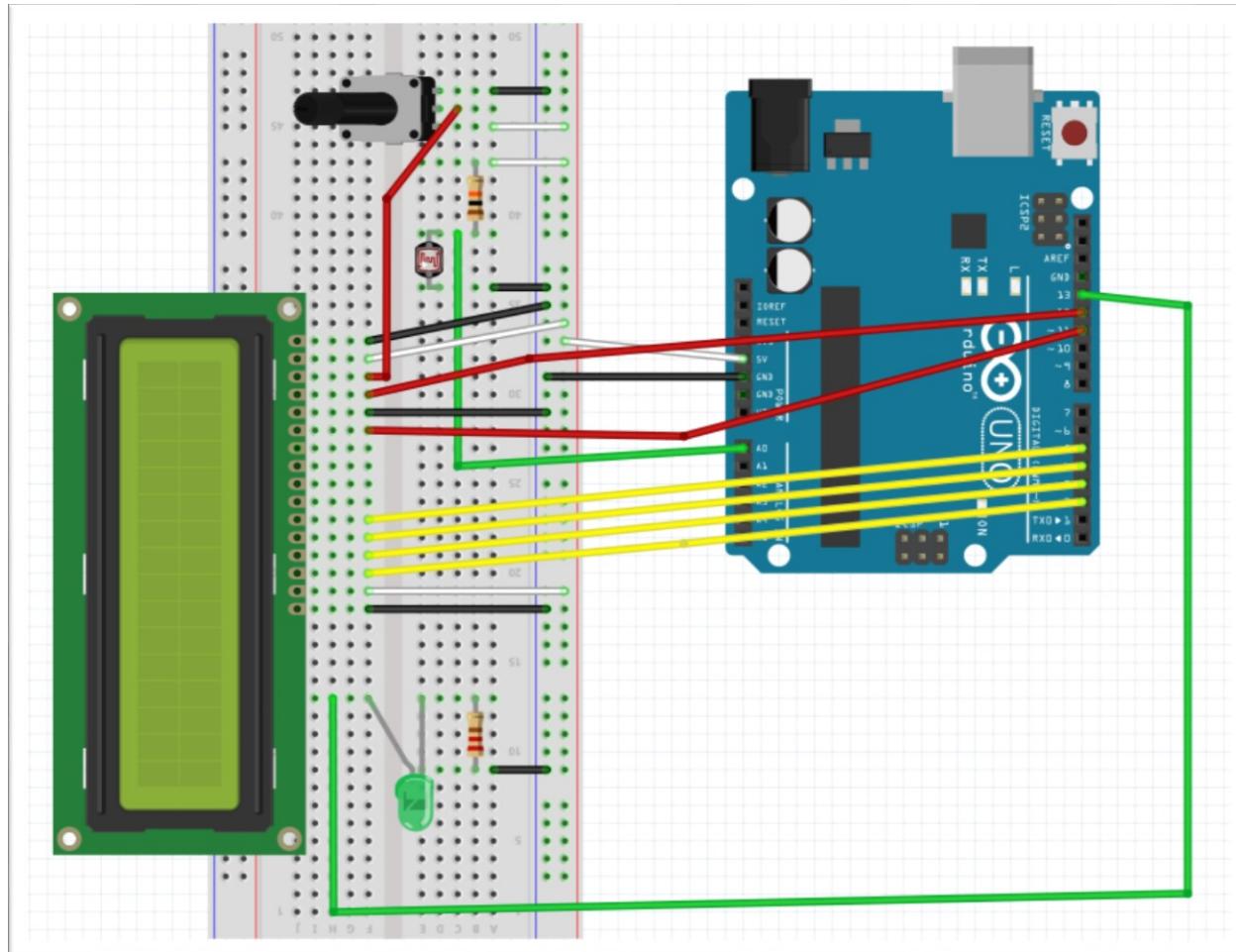
Design new circuit with Fritzing

[DIY] 아래 왼쪽 회로에 단색 LED, 330 Ω 저항을 추가하시오.
(오른쪽 회로 참조)
→ HPnn_Lux.fzz 로 저장.



Design new circuit with Fritzing

[DIY] HPnn_Lux.fzz 로 저장.



Fritzing 화로도 소스로
HPnn_lux.fzz 로 저장...



조도 센서 (빛의 양 측정)

EX 6.2

빛 입력 (2/3)

Commands

- `analogRead(아날로그 핀번호)`

아날로그 핀에서 아날로그 값을 읽는다. 0~5V사이의 전압을 0~1023 사이의 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.

- `LiquidCrytral 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.



6.2 조도 센서 (빛의 양 측정)

CdS 센서 LCD 회로 - code 1.

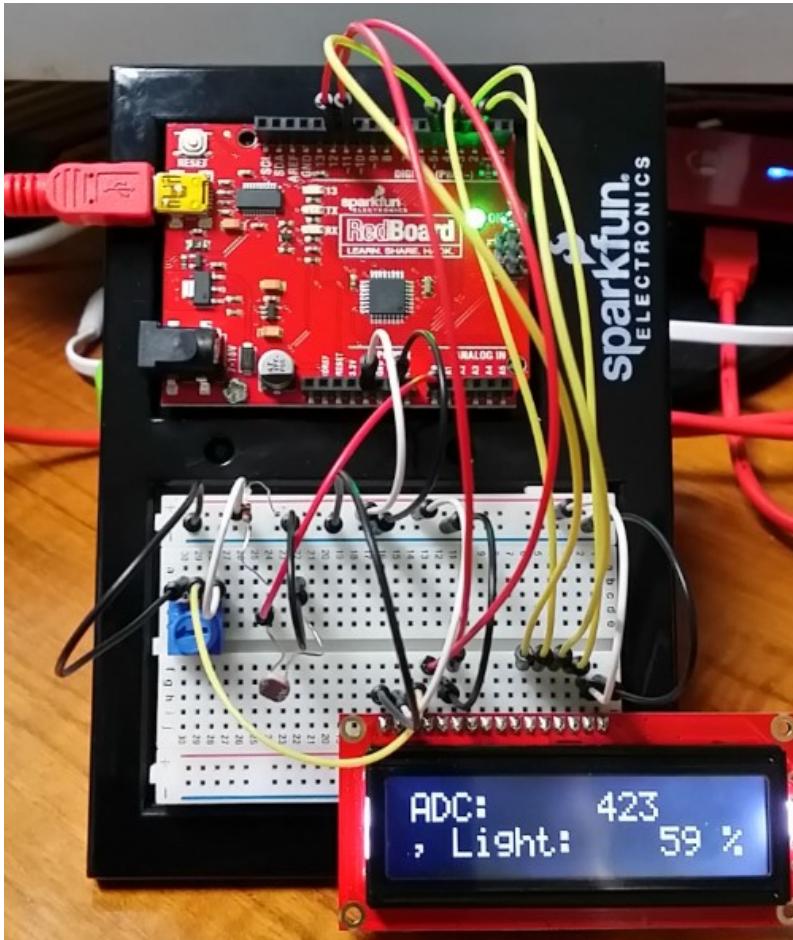
ex_6_2_start.ino

```
6 // LCD 라이브러리 설정
7 #include <LiquidCrystal.h>
8
9 // LCD 설정
10 LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
11 // rs,en,d4,d5,d6,d7
12 // 0번 아날로그핀을 CdS 셀 입력으로 설정한다.
13 const int CdSPin = 0;
14
15 void setup() {
16 // 16X2 LCD 모듈 설정하고 백라이트를 켠다.
17 lcd.begin(16,2);
18
19 // 메세지를 표시한다.
20 lcd.print("ex 6.2");
21 lcd.setCursor(0,1);
22 lcd.print("CdS Cell Test");
23 // 3초동안 메세지를 표시한다.
24 delay(3000);
25
26 // 모든 메세지를 삭제한 뒤
27 // 숫자를 제외한 부분들을 미리 출력시킨다.
28 lcd.clear();
29 lcd.setCursor(0,0);
30 lcd.print("ADC: ");
31 lcd.setCursor(0,1);
32 lcd.print(", Light: ");
33 lcd.setCursor(15,1);
34 lcd.print("%");
35 }
```

```
37 void loop(){
38
39 int adcValue; // 실제 센서로부터 읽은 값 (0~1023)
40 int illuminance; // 현재의 밝기. 0~100%
41
42 // CdS cell을 통하여 입력되는 전압을 읽는다.
43 adcValue = analogRead(CdSPin);
44 // 아날로그 입력 값을 0~100의 범위로 변경한다.
45 illuminance = map(adcValue, 0, 1023, 100, 0);
46
47 // 전에 표시했던 내용을 지우고
48 // LCD에 ADC 값과 밝기를 출력한다.
49 // 지우지 않으면 이전에 표시했던 값이 남게 된다.
50
51 // 전에 표시했던 내용을 지운다.
52 lcd.setCursor(9,0);
53 lcd.print(" ");
54 // ADC 값을 표시한다
55 lcd.setCursor(9,0);
56 lcd.print(adcValue);
57
58 // 전에 표시했던 내용을 지운다.
59 lcd.setCursor(13,1);
60 lcd.print(" ");
61 // 밝기를 표시한다
62 lcd.setCursor(12,1);
63 lcd.print(illuminance);
64
65 delay(1000);
66 }
```

6.2 조도 센서 (빛의 양 측정)

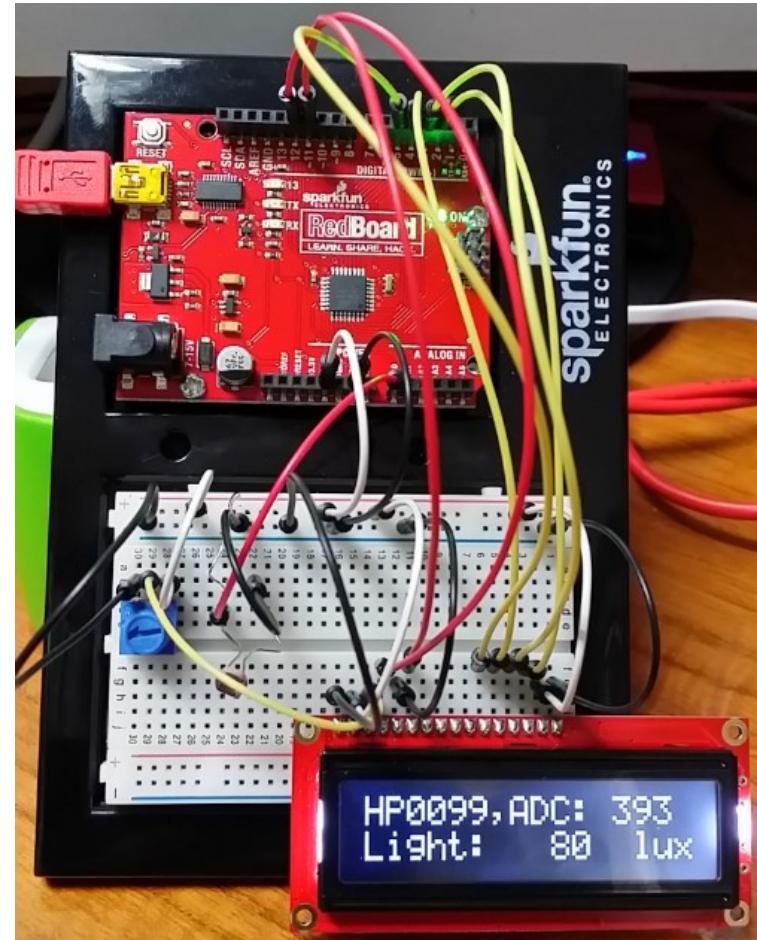
CdS 센서 LCD 회로 - 측정 1



실제 밝기값
lux로
표시되도록
코드를
수정하시오.

luminosity
함수 이용

수정된 화면을
폰으로
촬영해서
그림을
제출하시오.





6.2 조도 센서 (빛의 양 측정)

CdS 센서 LCD 회로 - code 2.

cds_LCD

```

6 // LCD 라이브러리 설정
7 #include <LiquidCrystal.h>
8
9 // LCD 설정
10 LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
11 //      rs,en,d4,d5,d6,d7
12 // 0번 아날로그핀을 CdS 셀 입력으로 설정한다.
13 const int CdSPin = 0;
14
15 void setup() {
16 // 16X2 LCD 모듈 설정하고 백라이트를 켠다.
17 lcd.begin(16,2);
18
19 // 메세지를 표시한다.
20 lcd.print("ex 6.2");
21 lcd.setCursor(0,1);
22 lcd.print("CdS Cell Test");
23 // 3초동안 메세지를 표시한다.
24 delay(3000);
25
26 // 모든 메세지를 삭제한 뒤
27 // 숫자를 제외한 부분들을 미리 출력시킨다.
28 lcd.clear();
29 lcd.setCursor(0,0);
30 lcd.print("HP0099,ADC: ");
31 lcd.setCursor(0,1);
32 lcd.print("Light: ");
33 lcd.setCursor(13,1);
34 lcd.print("lux"); // 수정
35 }

```

```

37 void loop(){
38     int adcValue; // 실제 센서로부터 읽은 값 (0~1023)
39     int illuminance; // 현재의 밝기, 0~100%
40     int lux; // 현재의 밝기, lux
41
42     // CdS cell을 통하여 입력되는 전압을 읽는다.
43     adcValue = analogRead(CdSPin);
44     // luminosity() 함수를 이용해서 Lux 를 계산한다.
45     lux = int(luminosity(adcValue)); // 수정
46
47     // 전에 표시했던 내용을 지운다.
48     lcd.setCursor(12,0);
49     lcd.print("    ");
50     // ADC 값을 표시한다
51     lcd.setCursor(12,0);
52     lcd.print(adcValue);
53     // 전에 표시했던 내용을 지운다.
54     lcd.setCursor(9,1);
55     lcd.print("    ");
56     // 밝기를 표시한다
57     lcd.setCursor(9,1);
58     lcd.print(lux); // 수정
59     delay(1000);
60 }
61
62 //Voltage to LuxLux
63 double luminosity (int RawADC0){
64     double Vout=RawADC0*0.0048828125; // 5/1024 (Vin = 5 V)
65     int lux=(2500/Vout-500)/10; // lux = 500 / Rldr, Vout = Ildr*Rldr = (5/(10 + Rldr))*Rldr
66     return lux;
67 }

```

LED ON/OFF

기능을 추가해서

Code를 완성 후,

HPnn_lux.ino

로 저장...



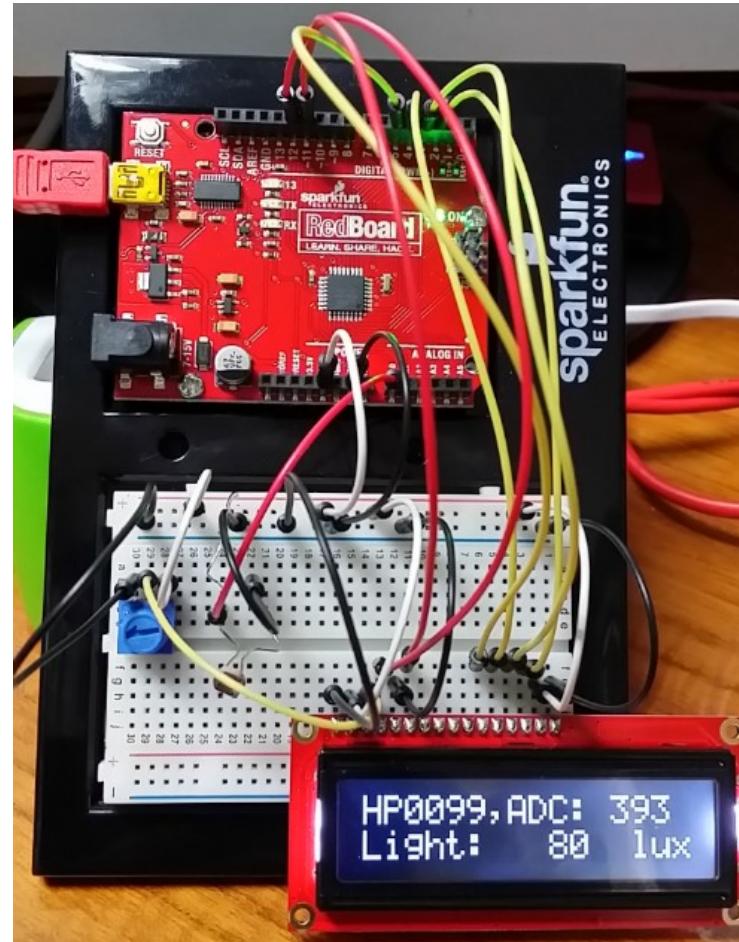
6.2 조도 센서 (빛의 양 측정)

CdS 센서 LCD 회로 - 측정 2

실제 밝기가
lux로 표시되도록
코드를 수정하시오.

luminosity
함수 이용

수정된 화면을
폰으로 촬영해서
그림을 제출하시오.



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



[Project]

- ◆ [wk14]
 - Arduino coding VI
 - Complete your project
 - Upload file name : HPnn_Rpt09.zip

wk14 : Practice-09 : HPnn_Rpt09.zip

◆ [Target of this week --- 배점:15]

- Complete your 9th project
- Confirm your outcome with LED ON/OFF.
- Save your outcome and compress all files.

제출파일명 : **HPnn_Rpt09.zip**

- 압축할 파일들 (jpg file도 가능함.)

- ① **HPnn_servo.png**
- ② **HPnn_Lux.png**
- ③ **HPnn_Lux.fzz**
- ④ **HPnn_Lux.ino**

wk15 : Final examination (written)



일시 : 2017. 6. 21 (수) 6 교시(오후 2시 ~)

장소 : E115

시험범위 : Arduino coding --- 30점 (30%)

- wk09_HP ~ wk14_HP
- 수업 중 배운 Arduino 코드에서 출제 (객관식/단답형/코드작성)
- [참고 소스] github.com/redwoods/hw-coding/

성적 평가 기준

1. 기초회로 – 30 점
2. Arduino 과제(9회, 기말 포함) – 40점 (환산 점수)
3. Arduino 필기시험 - 30점

wk15 : Final examination (codes)



github.com/redwoods/hw-coding/

Redwoods / hw-coding

Code Issues 0 Pull requests 0 Projects 0 Wiki Settings

Branch: master ▾ hw-coding / Src / src4exam /

Redwoods Upload arduino sources for final examination.

..

Cds_LCD.ino	Upload arduino sources for final examination.
HelloHPnn.ino	Upload arduino sources for final examination.
Light_final.ino	Upload arduino sources for final examination.
hp00_4_leds_A.ino	Upload arduino sources for final examination.
hp00_diy2_escape.ino	Upload arduino sources for final examination.
hp00_diy3_sum100.ino	Upload arduino sources for final examination.
hp00_led_period.ino	Upload arduino sources for final examination.
hp00_number_output.ino	Upload arduino sources for final examination.
hy00_6_1_pwm.ino	Upload arduino sources for final examination.



교재



1장 Arduino 소개 및 사용법

2장 시리얼 통신

3장 LCD 출력

4장 LED 출력 1

4장 LED 출력 2

4장 LED 출력 3

5장 디지털 신호 입력

중간평가

6장 아날로그 신호 입력 1

| 6장 아날로그 신호 입력 2

| 7장 모터 구동

| 8장 적외선 리모컨

| 9장 여러 가지 부품들

| 10장 프로젝트 1

| 10장 프로젝트 2

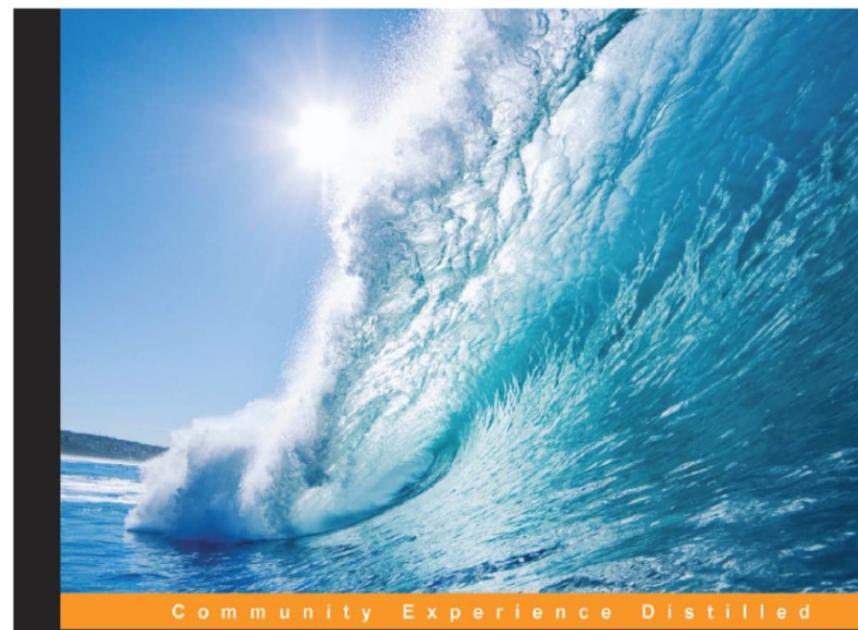
| 기말평가

생능출판사

2016



References

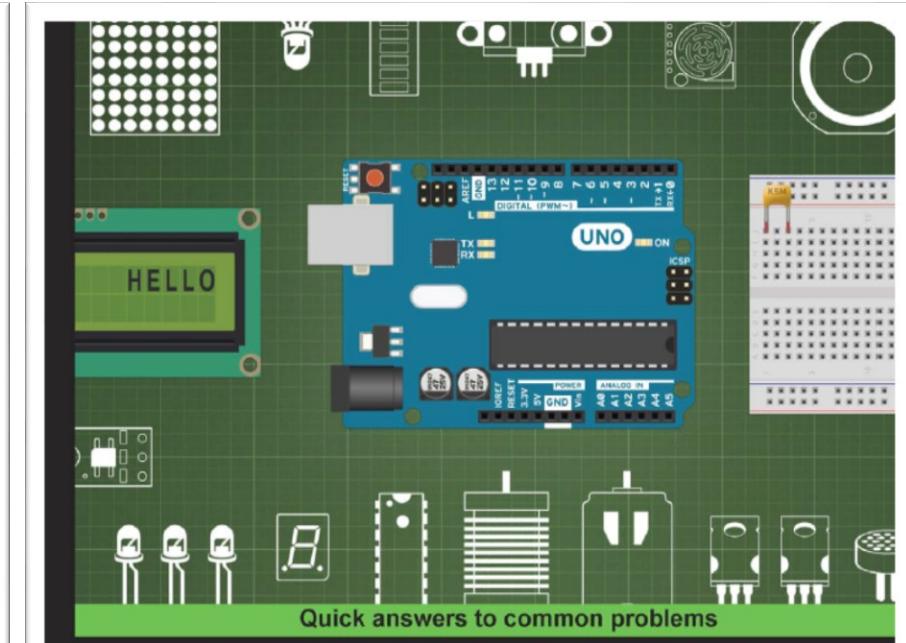


Arduino Essentials

Enter the world of Arduino and its peripherals and start creating interesting projects

Francis Perea

[PACKT]
PUBLISHING



Arduino Development Cookbook

Over 50 hands-on recipes to quickly build and understand Arduino projects, from the simplest to the most extraordinary

Cornel Amariei

[PACKT] open source*
PUBLISHING



[참고 : 저항 값 읽기]

