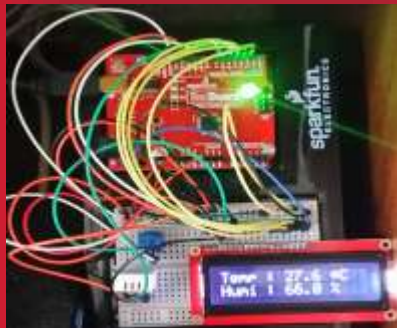




Arduino-basic

[wk03]

LCD



Learn how to code Arduino from scratch

Comsi, INJE University

2nd semester, 2018

Email : chaos21c@gmail.com



My ID (ARnn)

성명	ID
백동진	AR01
김도훈	AR02
김희찬	AR03
류재현	AR04
문민규	AR05
박진석	AR06
이승현	AR07
이승협	AR08
이후정	AR09
최민구	AR10

김다영	AR11
공진영	AR12
김해인	AR13
류성현	AR14
류재환	AR15
박상현	AR16
박해주	AR17
백지혜	AR18
송원식	AR19
신송주	AR20
윤지훈	AR21
정은성	AR22



[Review]

◆ [wk02]

- Serial comm.
- Complete your project
- Submit file : ARnn_Rpt01.zip

◆ [Target of this week]

- Complete your works
- Save your outcomes and compress 4 figures

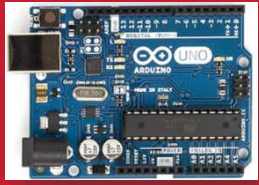
제출파일명 : **ARnn_Rpt01.zip**

- 압축할 파일들

- ① **ARnn_blink.png**
- ② **ARnn_sawtooth.png**
- ③ **ARnn_loop_escape.png**
- ④ **ARnn_sum100.png**

Email : chaos21c@gmail.com

[제목 : id, 이름 (수정)]



Hello

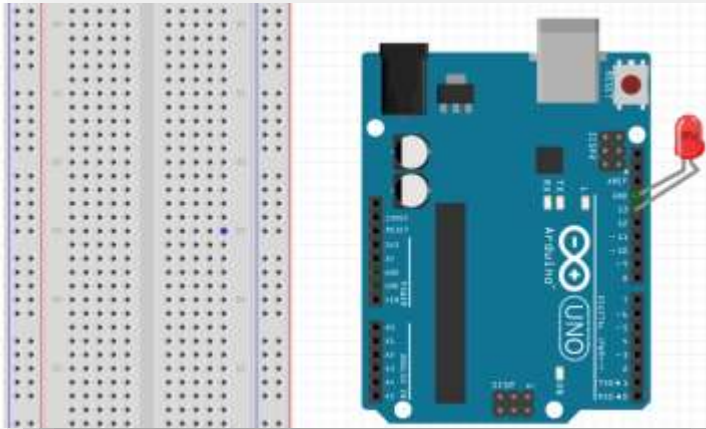


Blink

a LED



Blink a LED!



```
Blink$
1 /*
2  Blink
3  Turns on an LED on for one second, then off for one second, repeatedly.
4  */
5
6
7 // the setup function runs once when you press reset or power the board
8 void setup() {
9   // initialize digital pin 13 as an output.
10  pinMode(13, OUTPUT);
11 }
12
13 // the loop function runs over and over again forever
14 void loop() {
15   digitalWrite(13, HIGH);  // turn the LED on (HIGH is the voltage level)
16   delay(1000);             // wait for a second
17   digitalWrite(13, LOW);   // turn the LED off by making the voltage LOW
18   delay(1000);             // wait for a second
19 }
```



2. Serial comm. monitor & plotter

2. Serial comm.

시리얼 통신

- 2.1 Arduino에서 컴퓨터로 데이터 전송하기
- 2.2 변수 유형별로 컴퓨터에 전송하기
- 2.3 Arduino에서 시리얼 통신을 이용하여
 데이터 수신하기

2.1.3 Arduino에서 컴퓨터로 데이터 전송하기

ex_2_1 | 아두이노 1.8.2

↔

□

×

파일 편집 스케치 툴 도움말

✓

→

📄

⬆

⬇

ex_2_1

```

1 /*
2  예제 2.1
3  Arduino에서 컴퓨터로 변수와 문자열 전송하기
4  */
5
6  int number = 0;           // -32768~32767 범위의 변수 number 설정, 초기값은 0
7
8  void setup() {
9    Serial.begin(9600);     // 9600bps로 시리얼 통신 설정
10
11
12  void loop() {
13    Serial.print(number);   // number 변수값 출력
14    Serial.println(" sec"); // " sec"를 출력 후 줄 바꿈
15    delay(1000);           // 1초동안 지연시킨다.
16    number++;              // number 변수값을 하나 증가시킨다.
17  }
18

```

업로드 완료.

스케치는 프로그램 저장 공간 1862 바이트(5%)를 사용, 최대 32256 바이트.

전역 변수는 동적 메모리 192바이트(9%)를 사용, 1856바이트의 지역변수가 남음. 최대는 2048 바이트.

COM3 (Arduino/Genuino Uno)

36 sec

37 sec

38 sec

39 sec

40 sec

41 sec

42 sec

43 sec

44 sec

45 sec

46 sec

47 sec

48 sec

49 sec

50 sec

51 sec

52 sec

53 sec

54 sec

55 sec

56 sec

57 sec

58 sec

59 sec

60 sec

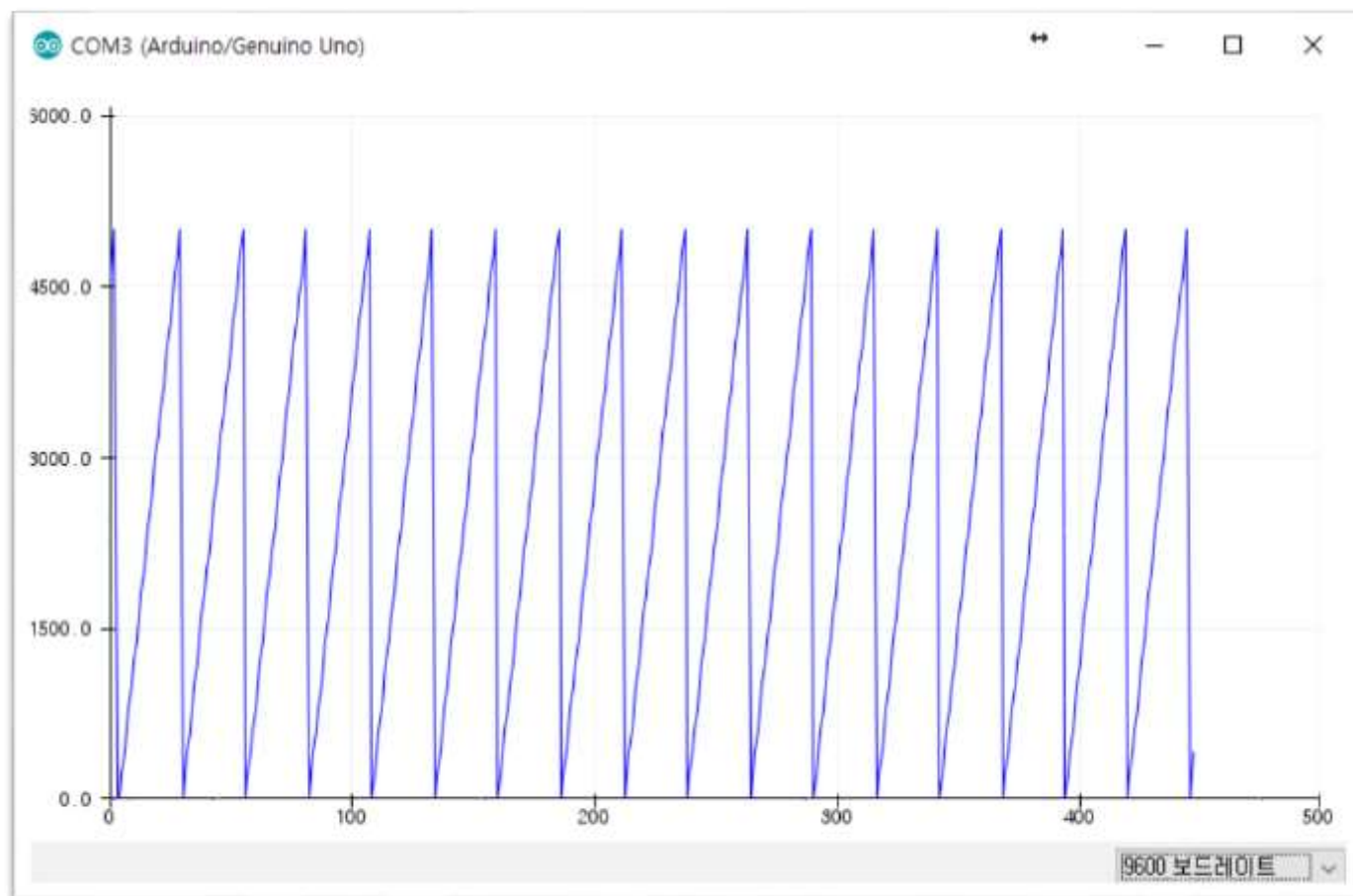
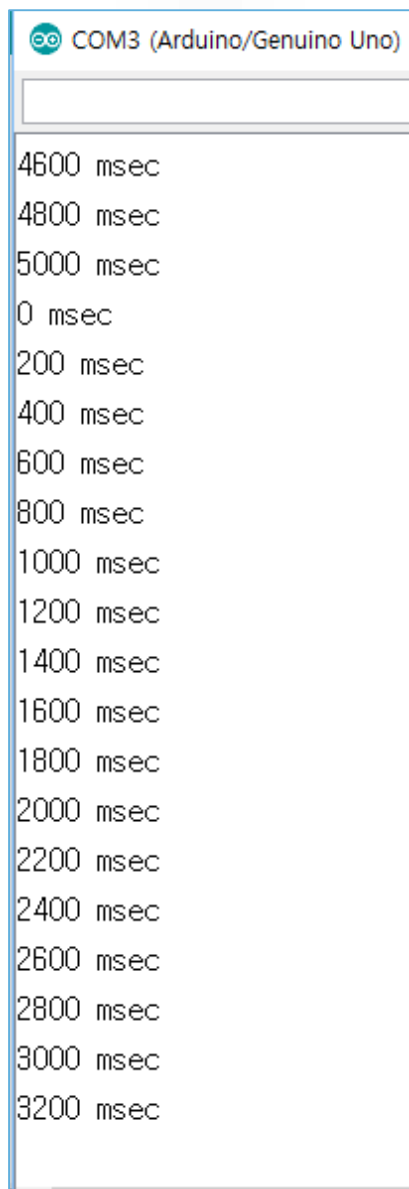
☒ 자동 스크롤

8

Arduino/Genuino Uno on COM3



DIY-1. sawtooth signal



Save ARnn_sawtooth.png



2. 시리얼 통신 (Serial comm.)

2.2

변수 유형별로 컴퓨터에 전송하기

```
*** Hello Arduino ***
```

```
*** char Value ***
```

```
Binary:1000001
```

```
Decimal:65
```

```
Hexadecimal:41
```

```
ASCII:A
```

```
*** int Value ***
```

```
int Value:65
```

```
char(intValue):A
```

```
*** float Value ***
```

```
float Value:65.00
```

DIY-2. Escape from loop()

응용 문제 [DIY-2] 0~15까지 10진수를 2진수와 16진수로 출력하는 스케치를 작성해보자

```
COM3 (Arduino/Genuino Uno)

Number = 0, Binary:0, Hexadecimal:0
Number = 1, Binary:1, Hexadecimal:1
Number = 2, Binary:10, Hexadecimal:2
Number = 3, Binary:11, Hexadecimal:3
Number = 4, Binary:100, Hexadecimal:4
Number = 5, Binary:101, Hexadecimal:5
Number = 6, Binary:110, Hexadecimal:6
Number = 7, Binary:111, Hexadecimal:7
Number = 8, Binary:1000, Hexadecimal:8
Number = 9, Binary:1001, Hexadecimal:9
Number = 10, Binary:1010, Hexadecimal:A
Number = 11, Binary:1011, Hexadecimal:B
Number = 12, Binary:1100, Hexadecimal:C
Number = 13, Binary:1101, Hexadecimal:D
Number = 14, Binary:1110, Hexadecimal:E
Number = 15, Binary:1111, Hexadecimal:F
Mission completed!
```

[Hint]

1. `int number = 0; // starting number`
2. `loop()`에서 1초 간격으로 `number`를 1씩 증가
3. 옆의 방식으로 결과 출력
4. `number`가 15를 초과하면 `loop()` 탈출
`exit(0); // loop 탈출 함수`

ARnn_loop_escape.png



DIY-2. Escape from loop() – code

응용 문제 [DIY-2] 0~15까지 10진수를 2진수와 16진수로 출력하는 스케치를 작성해보자

AR00_loop_escape

```
1 /*
2  DIY-2
3  */
4
5 // start number
6 int number = 0;
7
8 // 문자열 세가지를 설정한다.
9 String stringValue[]{"Binary:", "Hexadecimal:"};
10
11 void setup() {
12   // 9600bps로 시리얼 통신 설정
13   Serial.begin(9600);
14 }
```

```
16 void loop() {
17
18   // 'char Value'를 출력하고 문자열과 숫자를 변수 유형별로 출력한다.
19   Serial.print("Number = ");
20   Serial.print(number);
21   Serial.print(", ");
22   Serial.print(stringValue[0]); // stringValue 중 첫 번째 문자열 출력
23   Serial.print(number, BIN); // 2진수 형태로 출력
24   Serial.print(", ");
25   Serial.print(stringValue[1]); // stringValue 중 첫 번째 문자열 출력
26   Serial.print(number, HEX); // 16진수 형태로 출력
27   // 줄바꿈
28   Serial.println();
29
30   number++; // number 1 증가
31
32   if(number > 15) {
33     Serial.print("Mission completed!");
34     delay(1000);
35     exit(0);
36   }
37
38   delay(1000); // 1초동안 지연시킨다.
39 }
```



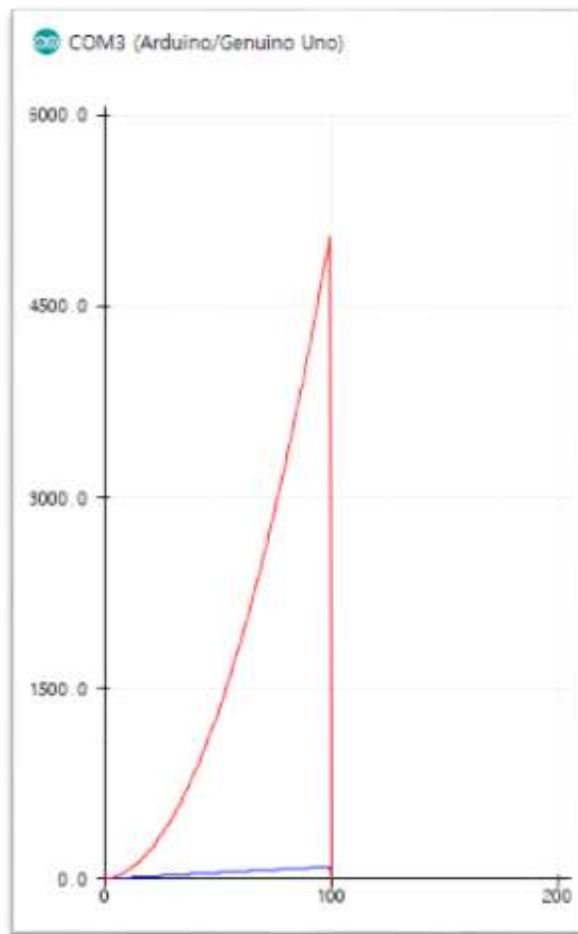
DIY-3. sum from 1 to 100

응용 문제 [DIY-3] Results on serial monitor and plotter

COM10 (Arduino/Genuino Uno)

Number = 86,	Sum = 3741
Number = 87,	Sum = 3828
Number = 88,	Sum = 3916
Number = 89,	Sum = 4005
Number = 90,	Sum = 4095
Number = 91,	Sum = 4186
Number = 92,	Sum = 4278
Number = 93,	Sum = 4371
Number = 94,	Sum = 4465
Number = 95,	Sum = 4560
Number = 96,	Sum = 4656
Number = 97,	Sum = 4753
Number = 98,	Sum = 4851
Number = 99,	Sum = 4950
Number = 100,	Sum = 5050

ARnn: $1 + 2 + \dots + 100 = 5050$



ARnn_sum100.png



DIY-3. sum from 1 to 100 - code

응용 문제 [DIY-3] Results on serial monitor and plotter

```
AR00_sum100
1 /*
2  DIY-3
3  */
4
5 // start number
6 int number = 0;
7 int sum = 0;
8
9 void setup() {
10  // 9600bps로 시리얼 통신 설정
11  Serial.begin(9600);
12 }
```

```
9 void setup() {
10  // 9600bps로 시리얼 통신 설정
11  Serial.begin(9600);
12 }
13
14 void loop() {
15
16  number++;
17  sum += number;
18  Serial.print("Number = ");
19  Serial.print(number);
20  Serial.print(", Sum = ");
21  Serial.println(sum);
22
23  if(number == 100) {
24    Serial.println();
25    Serial.print("ARnn: 1 + 2 + ... + 100 =");
26    Serial.println(sum);
27    delay(1000);
28    exit(0);
29  }
30
31  delay(100); // 0.1초동안 지연시킨다.
32 }
```



2.3 Serial monitor & plotter

2.3

시리얼 통신을 이용하여 데이터 수신하기



EX 2.3

변수 유형별 Arduino에서 컴퓨터로 전송하기 (1/3)

- 실습목표**
1. 컴퓨터에서 Arduino로 0~9의 숫자를 전송한다.
 2. Arduino에서는 전송 받은 숫자만큼 Arduino 보드의 LED를 점멸시킨다.

Hardware Arduino와 PC를 USB 케이블로 연결한다.

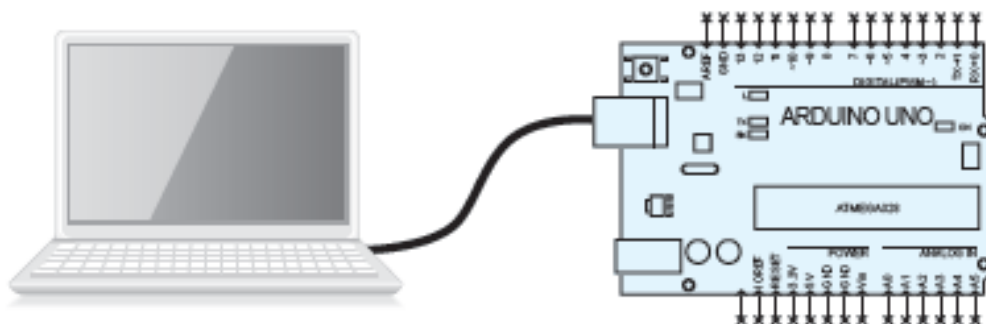


그림 2.1 Arduino와 PC와의 연결



3.3.2 시리얼 통신을 이용하여 데이터 수신하기

EX 2.3

변수 유형별 Arduino에서 컴퓨터로 전송하기 (2/3)

Commands • Serial.[available\(\)](#)

시리얼 통신에 수신된 데이터가 있는지 확인한다. 있을 경우 참(true)의 값을 갖는다.

• Serial.[read\(\)](#)

시리얼 통신을 통하여 수신된 값을 읽는다.

• [isDigit](#)(변수)

변수의 값이 ASCII 코드의 0~9의 숫자 범위에 있는지 여부를 판단. 범위에 있을 경우 참(true)의 값을 갖는다.

• [pinMode](#)(핀번호, 설정)

핀의 입출력 모드를 설정한다. '핀번호'에는 설정하고자 하는 핀의 번호와 '설정'에는 입력으로 사용하기 위해선 'INPUT', 출력으로 사용하기 위해선 'OUTPUT', 입력이며 풀업 사용시 'INPUT_PULLUP'을 적는다.

• [digitalWrite](#)(핀번호, 값)

핀에 디지털 출력(High or Low)을 한다. '핀번호'에는 출력하고자 하는 핀의 번호를, '값'에는 'HIGH' 혹은 'LOW'를 설정하여 High 혹은 Low 출력을 한다.



2.3.3 시리얼 통신을 이용하여 데이터 수신하기

EX 2.3

변수 유형별 Arduino에서 컴퓨터로 전송하기 (3/3)

- Sketch 구성**
1. 13번 핀에 연결된 내장 LED를 이용한다.
 2. 시리얼 통신 상태를 감시한 후 시리얼 통신으로 입력되는 데이터가 있을 때 이를 저장한다.
 3. 전송된 값은 ASCII 코드값이므로 이를 숫자로 변경한다.
 4. 숫자만큼 LED를 0.2초 간격으로 점멸시킨다.

실습 결과 IDE의 시리얼 모니터를 실행시켜 전송란에 0~9의 값을 입력한 후 Arduino의 LED가 입력한 값 만큼 점멸하는지를 확인해 본다..

- 응용 문제**
1. 0~9의 입력 값에 따라 점멸 주기가 변화하는 스케치를 작성해 보자.
 2. 0~9의 숫자를 전송하면 전송된 수의 2진수와 16진수를 컴퓨터로 전송하는 스케치를 만들어보자. (hint: 예제 2.2를 참고하자)



2.3.3 시리얼 통신을 이용하여 데이터 수신하기

EX 2.3

변수 유형별 Arduino에서 컴퓨터로 전송하기 (code)

ex_2_3_final

```
1 /*
2  예제 2.3
3  컴퓨터로부터 시리얼 통신을 통하여 데이터 수신하기
4 */
5
6 // LED 출력을 할 핀 번호 설정
7 const int ledPin = 13;
8
9 // 점멸횟수 변수 설정
10 int blinkNumber = 0;
11
12 void setup() {
13   // 9600bps로 시리얼 통신 설정
14   Serial.begin(9600);
15   // 13번 핀을 출력으로 설정
16   pinMode(ledPin, OUTPUT);
17 }
18
```

```
19 void loop() {
20   // 시리얼 통신으로 입력 받은 데이터가 있는지를 검사하여
21   // 데이터가 있을 경우에 if문 안의 명령어를 실행
22   if (Serial.available()) {
23     // val 변수에 시리얼 통신값 읽어오기
24     char val = Serial.read();
25     // 입력된 값이 0~9의 숫자인지를 판단
26     if (isDigit(val)) {
27       // val은 ASCII 코드값이므로 숫자로 바꿔주기 위하여
28       // '0'의 아스키 코드값을 빼줌
29       // blinkNumber에는 실제 숫자가 저장된다.
30       blinkNumber = (val - '0');
31     }
32     Serial.print("입력한 수:");
33     Serial.println(blinkNumber);
34     // Serial.println();
35     delay(2000);
36
37     // blinkNumber 만큼 LED의 켜짐상태를 길게 유지.
38     for (char i = 0; i < blinkNumber; i++) {
39       digitalWrite(ledPin, HIGH);
40       delay(100);
41       digitalWrite(ledPin, LOW);
42       delay(100);
43     }
44   }
45   // 점멸 횟수를 리셋함
46   blinkNumber = 0;
47 }
```



DIY-4. 점멸 주기가 변화

응용 문제 [DIY-4] 0~9의 입력 값에 따라 점멸 주기가 변화하는 스케치를 작성해 보자.

- 시리얼모니터에 입력한 수를 표시
- 입력한 수에 비례해서 LED 켜 상태를 길게 유지.



완성된 스케치 code를

ARnn_period.ino

로 저장해서 제출.



DIY-5. 입력된 수를 변환하여 출력

- 응용 문제 [DIY-5] 0~9의 숫자를 전송하면 전송된 수의 2진수와 8진수를 컴퓨터로 전송하는 스케치를 만들어보자. (hint: 예제 2.2를 참고하자).
- 아래 출력 참조.

```
COM3 (Arduino/Genuino Uno)
7
입력한 수:2, Binary:10, Octal:2
입력한 수:5, Binary:101, Octal:5
입력한 수:9, Binary:1001, Octal:11
입력한 수:1, Binary:1, Octal:1
입력한 수:3, Binary:11, Octal:3
```

완성된 스케치 code를
ARnn_number.ino
로 저장해서 제출.



3. LCD

Liquid Crystal Display



핀에 직접 연결
7 핀



I²C 통신
2 핀

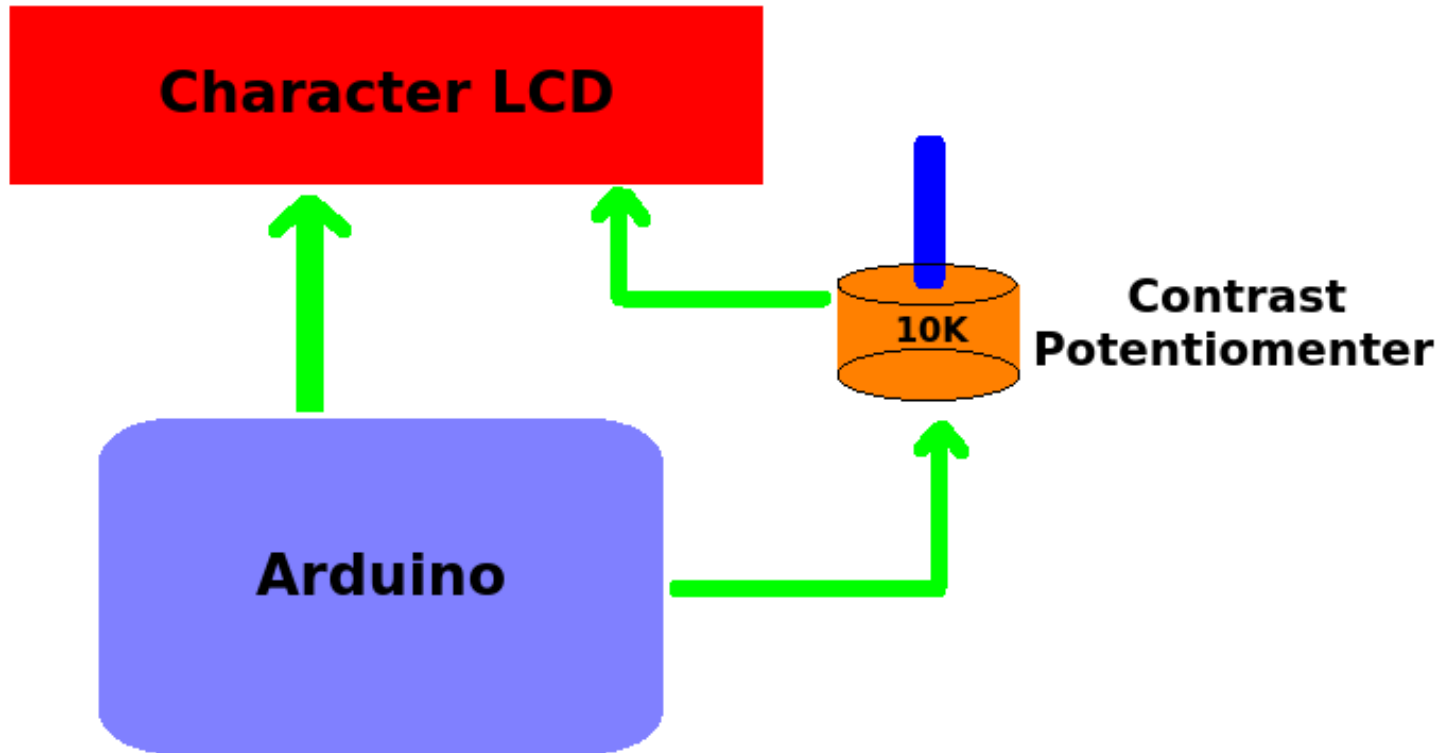


얇은 액정판 아래 조명을 비추는 장치로서 액정판의 전류 흐름을 제어하여 문자나 그림을 표시

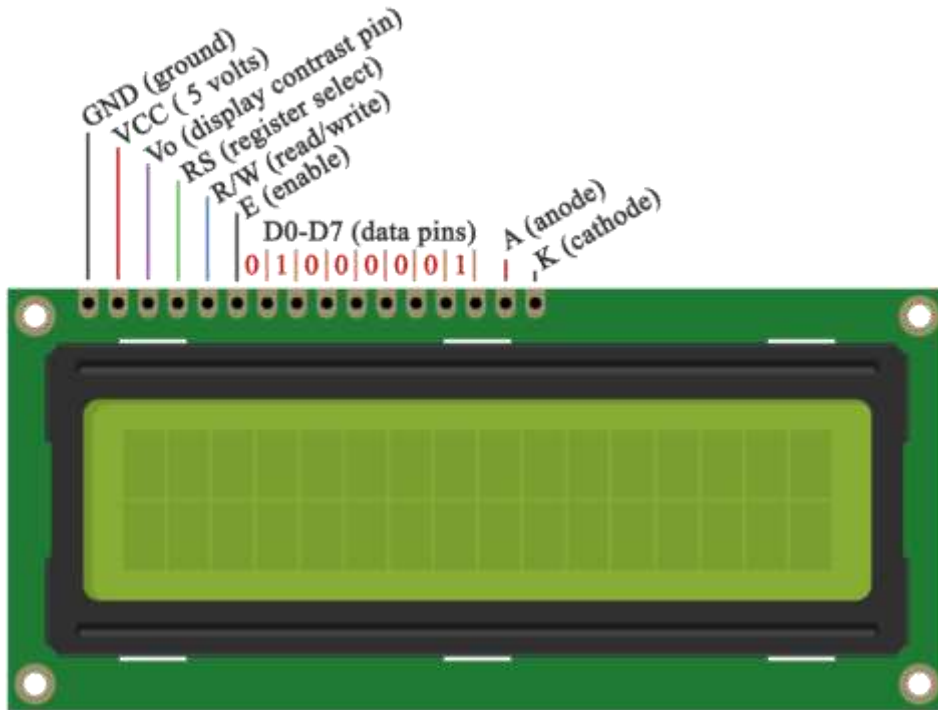
Liquid crystal display

- 3.1 입출력 핀을 이용하여 LCD 모듈에 표시하기
- 3.2 I²C를 이용한 LCD 출력

3.1.1 Introduction to LCD Module



LCD (Liquid Crystal Display, 16 X 2)

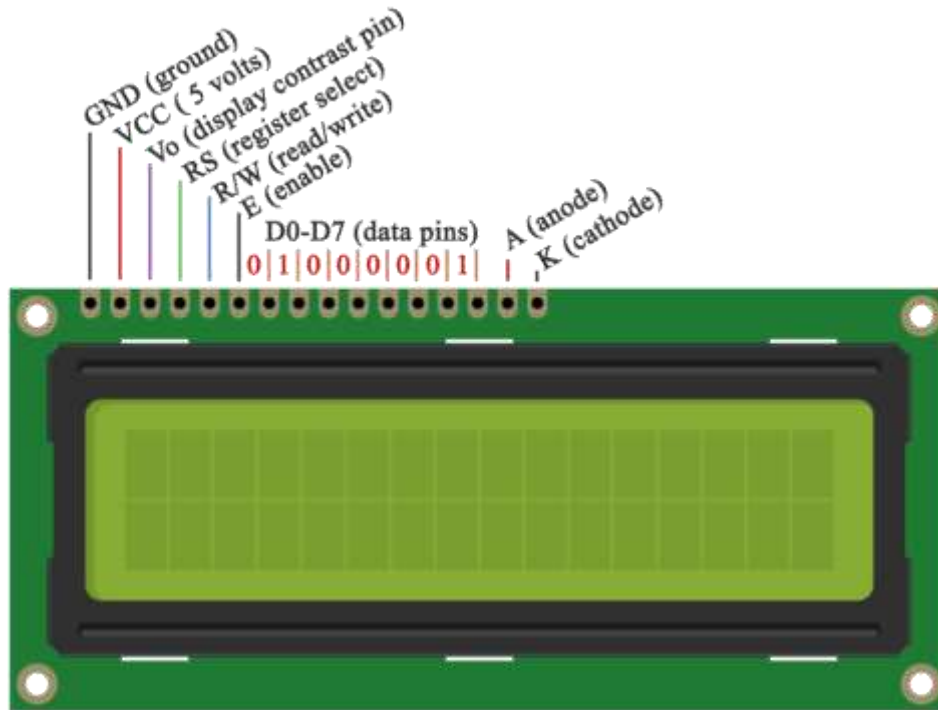


(1,2, ... 15,16)

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-)

3.1.3 Introduction to LCD Module

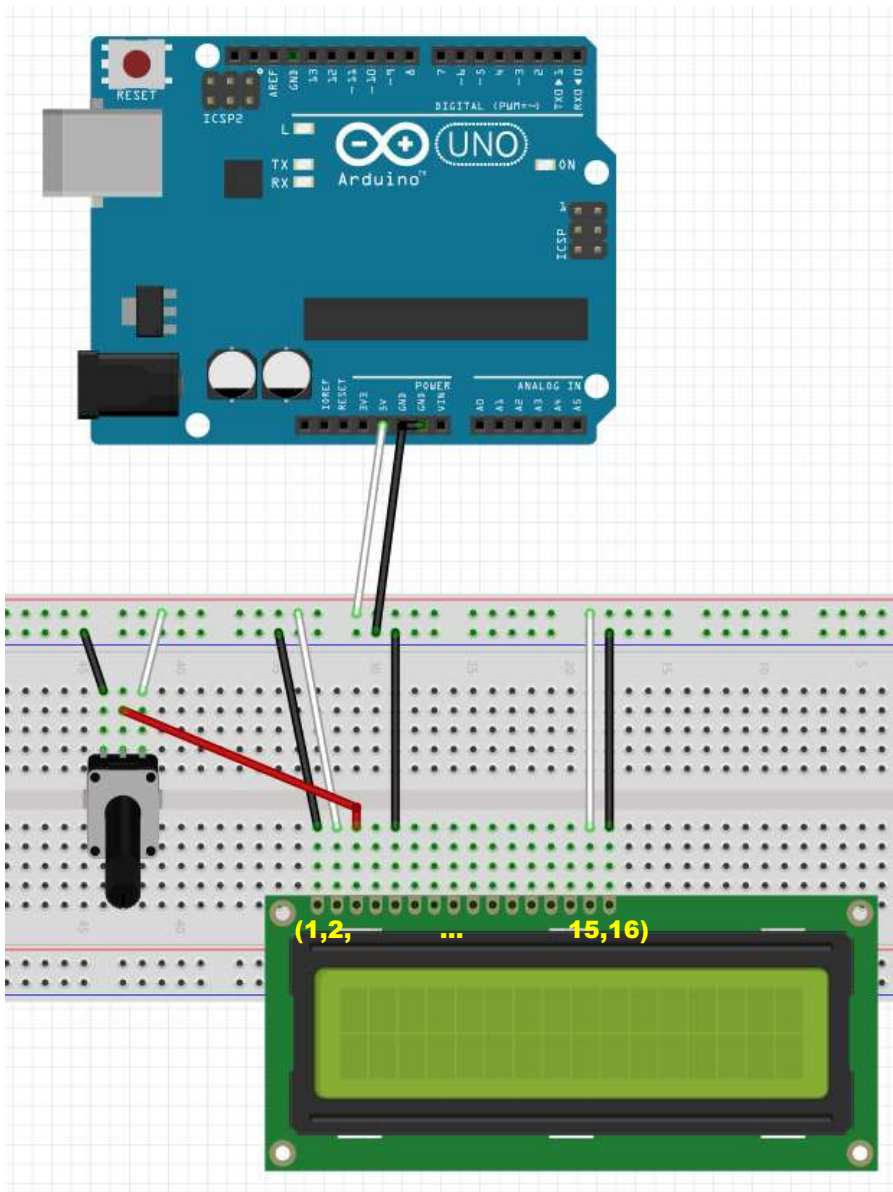
LCD (Liquid Crystal Display, 16 X 2)



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

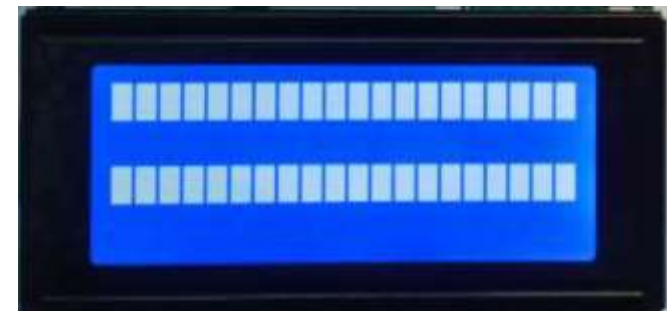
Pin 1 to Arduino GND
 Pin 2 to Arduino +5V
 Pin 3 to wiper
 Pin 4 to Arduino pin D12
 Pin 5 to Arduino GND
 Pin 6 to Arduino pin D11
 Pin 11 to Arduino pin D5
 Pin 12 to Arduino pin D4
 Pin 13 to Arduino pin D3
 Pin 14 to Arduino pin D2
 Pin 15 to +5V (with 220 or 330 Ω)
 Pin 16 to GND

3.1.4 LCD 초기화 (pin-1, 2, 3, 5, 15,16)

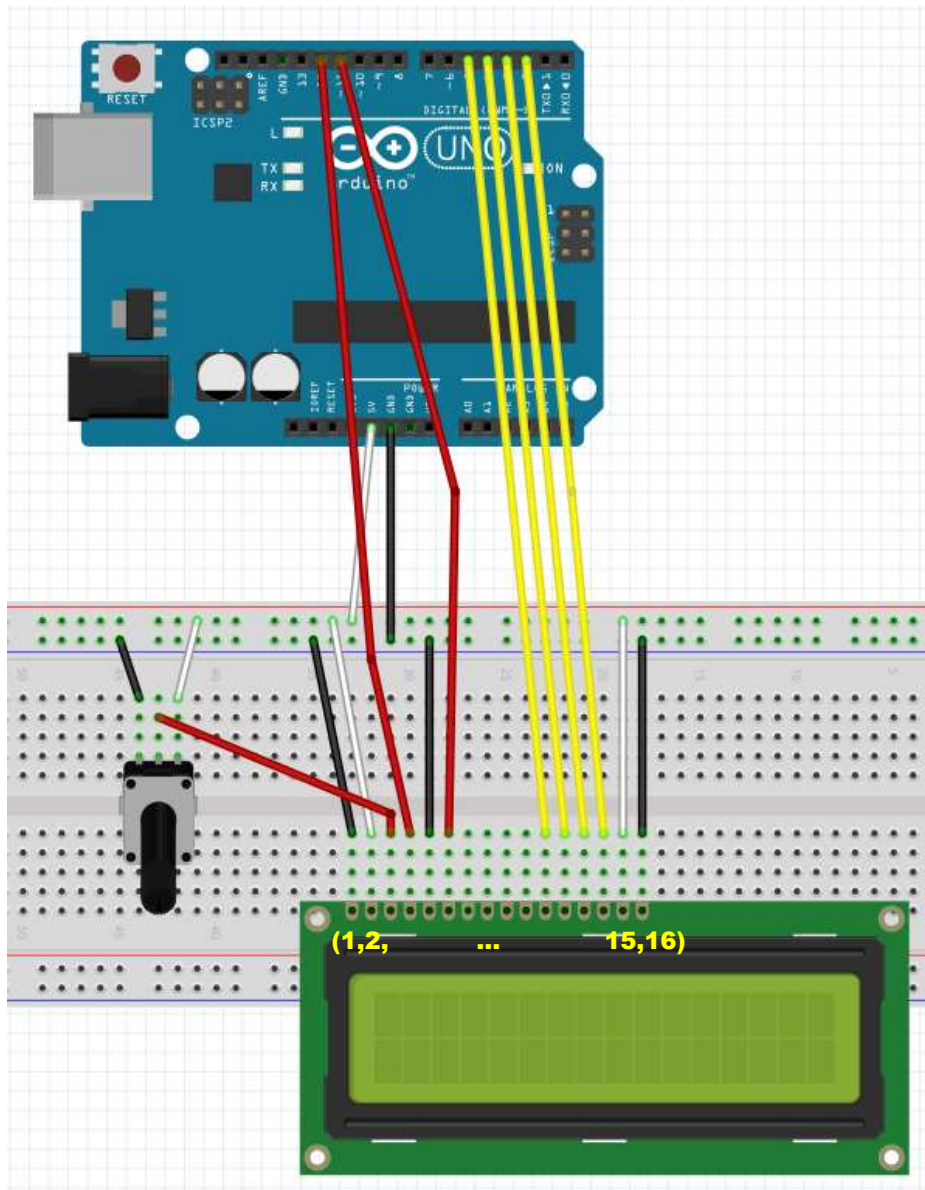


Pin 1 to Arduino GND
Pin 2 to Arduino +5V
Pin 3 to wiper (potentiometer)
Pin 5 to Arduino GND
Pin 15 to +5V
Pin 16 to GND

전원 연결 후
LCD 초기화



3.1.5 데이터 입력 초기화 (pin-4, 6, 11,12,13,14)



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

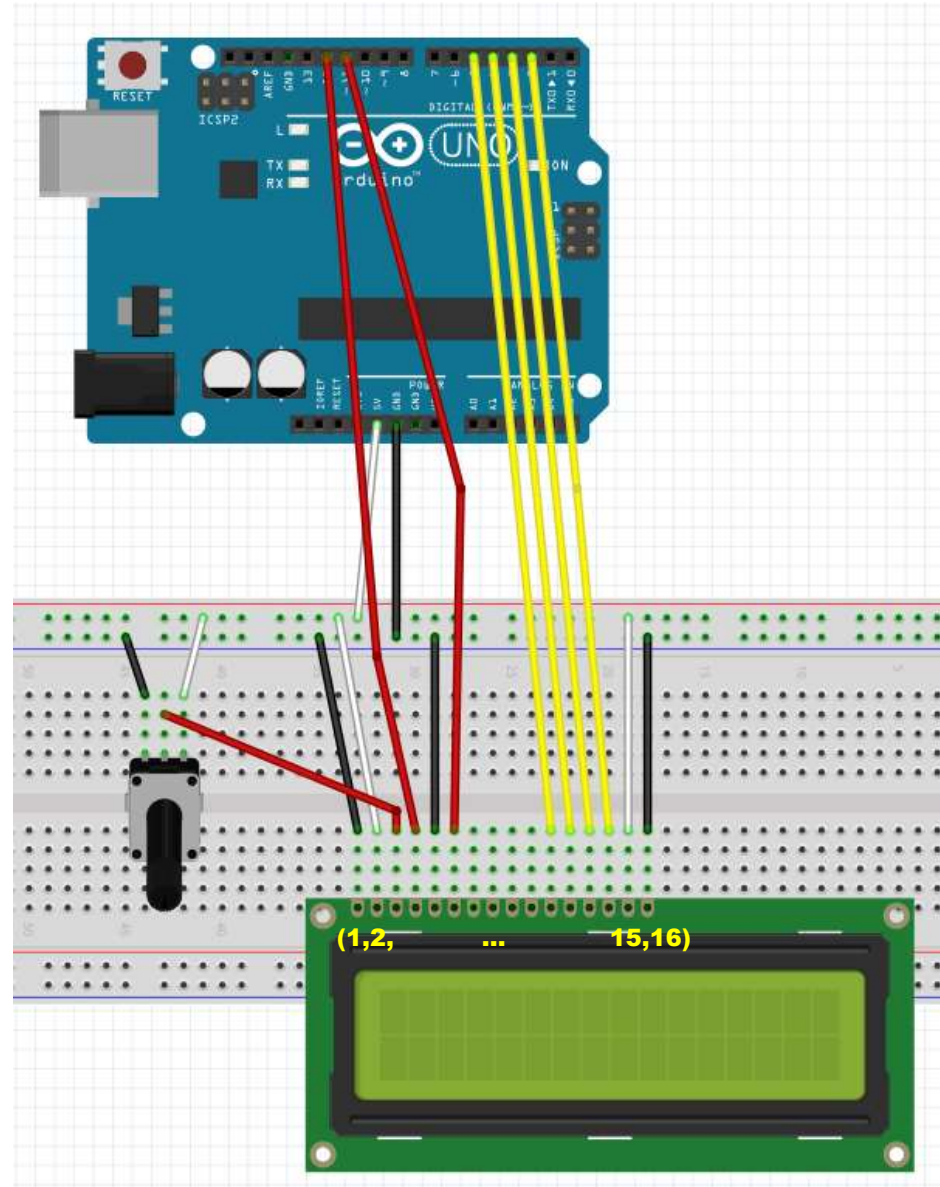
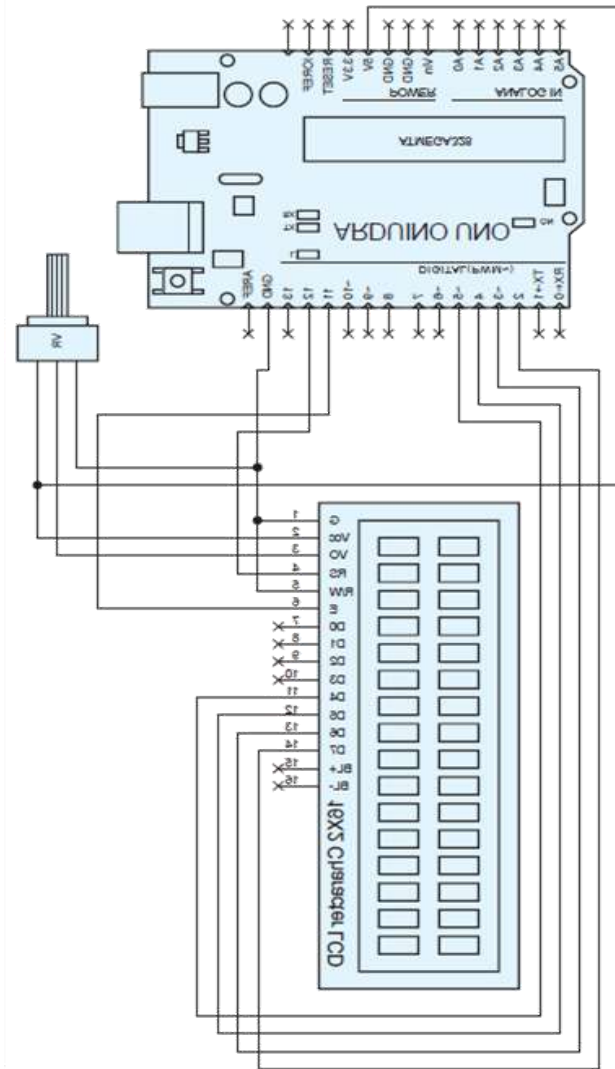
Pin 13 to Arduino pin D3

Pin 14 to Arduino pin D2

Pin 15 to +5V

Pin 16 to GND

DIY-6. LCD module circuit



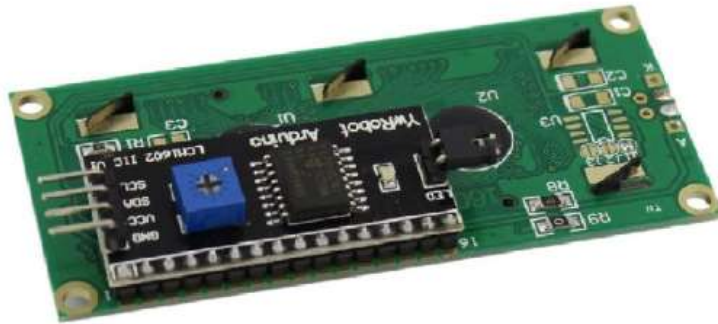
3.1.6 Introduction to LCD - code “Hello ARnn”

- `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.

3.1.7 Introduction to LCD - code “Hello ARnn”

```
hello_LCD
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, ARnn!");
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 }
```


3.2 I²C를 이용한 LCD 출력



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

I²C 는 풀업 저항이 연결된 직렬 데이터(**SDA**)와 직렬 클럭(**SCL**)이라는 두 개의 양 방향 오픈 컬렉터 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/>



3.2.1 I²C를 이용한 LCD 출력

특수문자나 기호는
LCD 모듈 문자표를 참고하여 출력

표 3.1 LCD 모듈 문자표

Upper 4 Bits	Lower 4 Bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)			00P`P										一タミαp			
xxxx0001	(2)			!1AQa4										。アチムäq			
xxxx0010	(3)			"2BRbr										「イツメβθ			
xxxx0011	(4)			#3CScs										」ウデモe∞			
xxxx0100	(5)			\$4DTdt										、エトヤμΩ			
xxxx0101	(6)			%5EUeu										・オナユcsÜ			
xxxx0110	(7)			&6FVfv										ヲカニヨρΣ			
xxxx0111	(8)			'7GWgw										フキヌラgπ			
xxxx1000	(1)			<8HXhx										イクネリJX			
xxxx1001	(2)			>9IYiy										ウケノル-y			
xxxx1010	(3)			*:JZjz										エコハレjチ			
xxxx1011	(4)			+;K[k{										オサヒロ*万			
xxxx1100	(5)			,<L¥ll										ヤシフワφ円			
xxxx1101	(6)			-=M]m}										ユスヘンも÷			
xxxx1110	(7)			.>N^n÷										ヨセホ"ん			
xxxx1111	(8)			/?O_o+										ッソマ°ö			

LCD의 원리(VA)

① Back Light Unit

② Polarizer

③ TFT

④ Liquid Crystal(액정)

⑤ Color Filter

⑥ Polarizer

배열(On)

배열(Off)

<http://blog.samsungdisplay.com>

3.2.2 I²C를 이용한 LCD 출력

I²C (Inter Integrated Circuit)

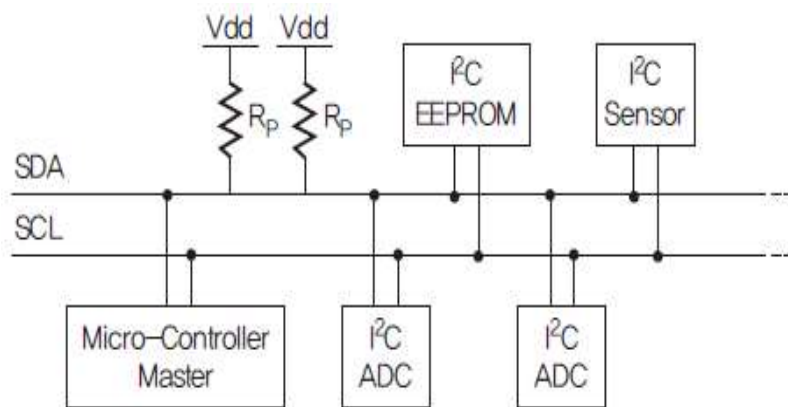


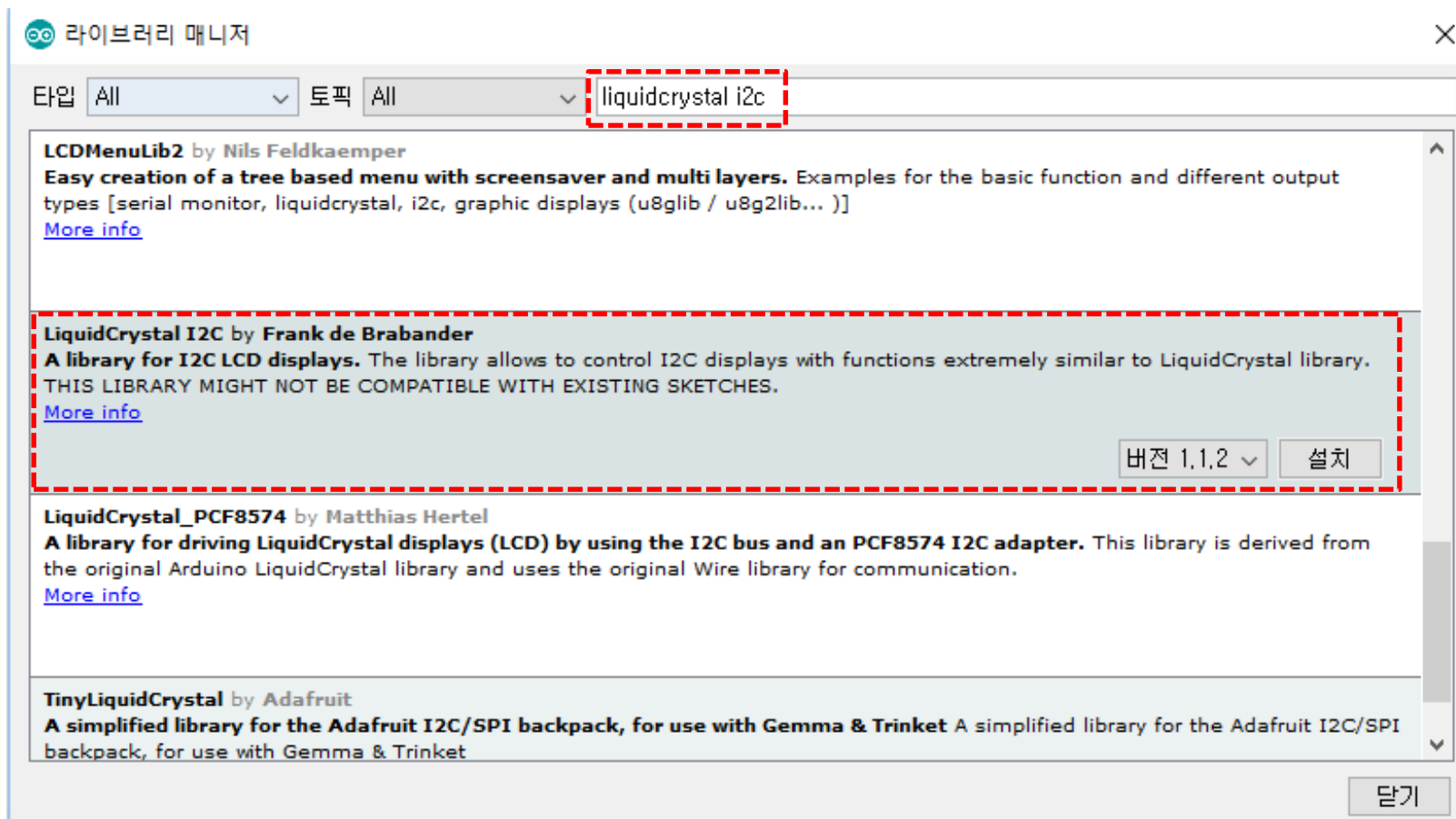
그림 3.2 I²C를 이용한 네트워크

- ✓ 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 위해선 PCF8574 IC를 사용
 - ✓ SDA, SCL 두 개의 입출력 핀만 필요

3.2.3 I²C를 이용한 LCD 출력

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

스케치 > 라이브러리 포함하기 > 라이브러리 관리



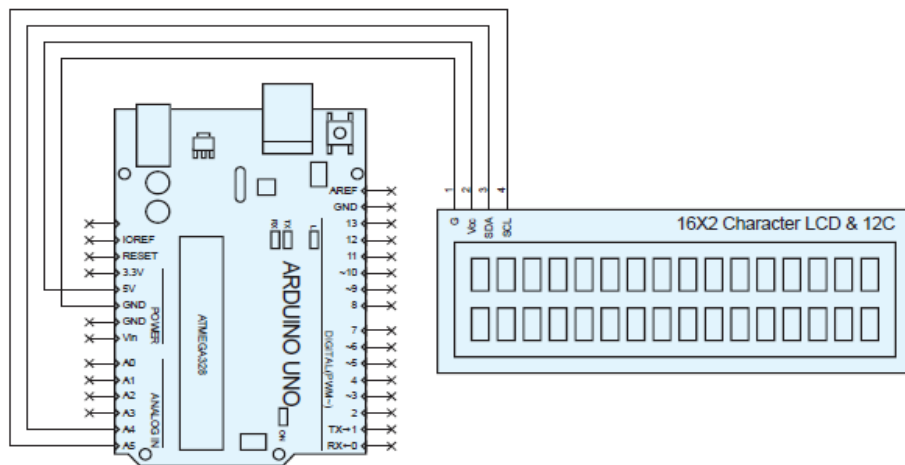
3.2.4 I²C를 이용한 LCD 출력

EX 3.2

I²C를 이용한 LCD 출력 (1/3)

- 실습목표**
1. 16X2 도트매트릭스 LCD를 I²C를 이용하여 제어한다.
 2. 'Welcome' 메시지와 함께 백라이트를 점멸시킨다.
 3. 시리얼 포트에 입력 받은 값을 LCD에 출력한다.

- 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.



3.2.5 I²C를 이용한 LCD 출력

EX 3.2 I²C를 이용한 LCD 출력 (2/3)

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.

Sketch 구성

1. I²C 방식의 LCD 모듈을 사용하기 위해 앞서 다운받은 라이브러리를 추가해 준다.
2. 라이브러리의 함수를 이용하여 LCD를 설정해 준다.
3. `setup()`에서 'Welcome'메시지와 백라이트를 점멸시킨다.
4. 시리얼 통신으로 데이터를 입력받기위해서 시리얼 통신 설정을 해 준다.
5. 데이터 입력이 있을 때 이를 LCD에 출력해 준다.

EX 3.2 I²C를 이용한 LCD 출력 (code)

```
ex_3_2_final
6 // I2C 통신 라이브러리 설정
7 #include <Wire.h>
8 // I2C LCD 라이브러리 설정
9 #include <LiquidCrystal_I2C.h>
10
11 // LCD I2C address 설정 PCF8574:0x27, PCF8574A:0x3F
12 LiquidCrystal_I2C lcd(0x3F,16,2); // LCD address:0x27,
```



```
14 void setup()
15 {
16   // 9600 bps로 시리얼 통신 설정
17   Serial.begin(9600);
18   lcd.init(); // LCD 설정
19   lcd.clear(); // LCD를 모두 지운다.
20   lcd.backlight(); // 백라이트를 켜다.
21   // Arduino LCD, Welcome 표시
22   lcd.setCursor(0,0);
23   lcd.print("Arduino LCD");
24   delay(3000);
25   lcd.setCursor(0,1);
26   lcd.print("Welcome");
27   delay(250);
28
29   // LCD 백라이트를 두 번 점멸
30   lcd.noBacklight();
31   delay(250);
32   lcd.backlight();
33   delay(250);
34   lcd.noBacklight();
35   delay(250);
36   lcd.backlight();
37   delay(3000);
38
39   // Open Serial Monitor, Type to display 표시
40   lcd.clear();
41   lcd.setCursor(0,0); //Start at character 0 on line 0
42   lcd.print("Open Serial Mntr");
43   lcd.setCursor(0,1);
44   lcd.print("Type to display");
45 }
```


3.2.6 I²C를 이용한 LCD 출력 (code-2)

EX 3.2 I²C를 이용한 LCD 출력 (code)



```

47 void loop()
48 {
49     // 시리얼 통신 수신 값이 있을 때
50     if (Serial.available()) {
51         delay(100);
52         // 모두 삭제
53         lcd.clear();
54         // 커서를 좌측 상단으로
55         lcd.setCursor(0,0);
56         // "Message from PC" 출력
57         lcd.print("Message from PC");
58         // 커서를 두 번째 줄로
59         lcd.setCursor(0,1);
60
61         // LCD에 PC에서 전송된 데이터를 출력
62         while (Serial.available() > 0) {
63             lcd.write(Serial.read());
64         }
65     }
66 }

```


EX 3.2 I²C를 이용한 LCD 출력 (3/3)

- 실행 결과
1. Arduino LCD 표시 후 백라이트가 2회 점멸한다.
 2. 시리얼 모니터를 실행 시킨 후 메시지를 입력하여 보자. → “Hello ARnn”
 3. 메시지가 LCD에 출력되는지를 확인해 보자.

Take a photo of LCD screen.

**Save photo as
ARnn_LCD_hello.png**



DIY-7

시리얼 통신으로 입력 받은 1~9의 숫자에 대하여 LCD의 백라이트가 입력된 숫자만큼 점멸하고 점멸 횟수를 표시하는 스케치를 작성해 보자.

(hint: 예제 2.3을 참고하자)

Save ARnn_LCD.ino



[Practice]

◆ [wk03]

- **Arduino LCD**
- **Complete your project**
- **Submit file : ARnn_Rpt02.zip**

wk03 : Practice-01 : ARnn_Rpt02.zip

◆ [Target of this week]

- Complete your works
- Save your outcomes and compress all.

제출파일명 : **ARnn_Rpt02.zip**

- 압축할 파일들

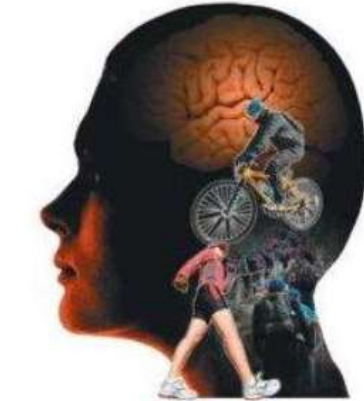
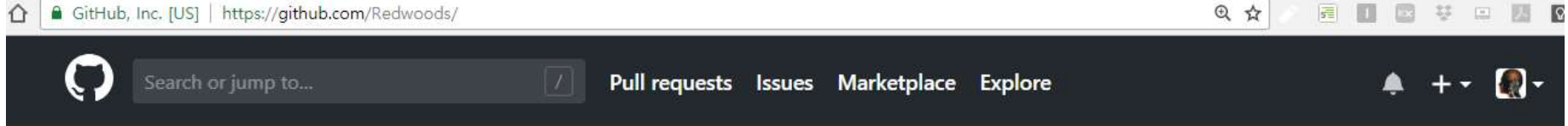
- ① **ARnn_period.ino**
- ② **ARnn_number.ino**
- ③ **ARnn_LCD.fzz**
- ④ **ARnn_LCD_hello.png**
- ⑤ **Arnn_LCD.ino**

Email : chaos21c@gmail.com

[제목 : id, 이름 (수정)]

● References & good sites

- ✓ <http://www.arduino.cc> Arduino Homepage
- ✓ <http://www.github.com> GitHub
- ✓ <http://www.google.com> Googling
- ✓ <https://www.youtube.com> Youtube



Redwoods Yi

Redwoods

Add a bio

GimHae, Republic of Korea

chaos21c@gmail.com

Overview

Repositories 7

Stars 2

Followers 1

Following 0

Pinned repositories

Customize your pinned repositories

Py

Lectures on coding python from scratch to the advanced level.

Jupyter Notebook

Arduino

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

Lec

All lectures by Redwoods in Inje University

Jupyter Notebook

hw-coding

Resource for lecture of Hardware Programming (2017, Inje university)


Arduino

171 contributions in the last year




Contribution settings

Redwoods/Arduino: Lect

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

 Search or jump to...

[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)

Redwoods / Arduino

Unwatch

1

Star

0

Fork

0

<> Code

Issues 0

Pull requests 0

Projects 0

Wiki

Insights

Settings

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

Edit

Add topics

2 commits

1 branch

0 releases

1 contributor

Branch: master


New pull request

Create new file




Upload files


Find file


Clone or download

 Redwoods 2018 start

Latest commit 38ca9e0 28 minutes ago

 ar-basic	2018 start	28 minutes ago
 ar-iot	2018 start	28 minutes ago
 README.md	Initial commit	43 minutes ago

 README.md



Arduino

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





http://arduinostory.com/goods/goods_view.php?goodsNo=1000000306

상급키트 구성품

1 1EA 아두이노 우노 R3 DIP 아두이노 우노 R3 (DIP) 호환보드 기본 메인보드입니다.	2 1EA 9V 배터리 홀더 9V 배터리를 연결하여 아두이노에 외부전원을 공급할 수 있습니다.	3 1EA 7세그먼트 4채널 7세그먼트가 4개 연결된 형태의 부품입니다. 총 12개의 핀을 사용합니다.	4 1EA 7세그먼트 1채널 공통 음극 7세그먼트 시계나 점수 등의 숫자를 표현 할 때 많이 사용됩니다.	13 1EA 수동부저 아두이노의 tone함수를 통해 소리를 내는 부저입니다.	14 6EA 택트스위치 (12x12x7) 스위치를 누르고 있을 경우만 ON됩니다.	15 각3EA 택트스위치 컵 (피랑,노랑,초록,빨강,하양) 택트스위치를 사용할 때 스위치간의 구분을 할 수 있습니다.	16 3EA 조도센서 빛을 감지하거나 빛의 밝기를 아날로그로 출력해주는 CDS 센서입니다.
5 1EA 74HC595N 기본 메인보드입니다. 74HC595N LED, 드로메트릭스, NFD 제어 IC 입니다.	6 1EA 65핀 점퍼 와이어 브레드보드에 연결할 때 사용하는 65핀 점퍼와이어 입니다.	7 1EA 무지개 점퍼선 F-M 20cm M타입과 F타입이 양쪽으로 달린 무지개 점퍼선입니다.	8 1EA 투명 부품 케이스 대,소 키트 구성품을 담을 수 있는 투명 부품 케이스입니다.	17 각5EA LED 5mm (빨강,노랑,초록,하양,파랑) 기본으로 사용되는 LED입니다. 동작전압 : 2.2~2.4V 사용전류 : 20mA 미만	18 1EA 헤더핀 1x40/2.54mm 핀 간격은 2.54mm이며 헤더핀의 길이는 약 1.15cm입니다.	19 1EA USB케이블 50cm PC와 아두이노 우노 보드를 연결하여 프로그램을 다운로드 할 때 사용합니다.	20 1EA 저항값 카드 저항값을 쉽게 확인 할 수 있는 카드입니다. 사이즈 : 60mm x 50mm
9 1EA 가변저항10K 물리변 저항값이 가능합니다. (0~10KΩ)	10 1EA 1602 I2C LCD 아두이노 16x2 I2C LCD 모듈입니다. LCD입니다.	11 각 10EA 저항 100, 220, 330, 1K, 2K, 4.7K, 10K, 47K, 100K	12 1EA 브레드 보드 830홀 브레드 보드 830홀(봉무형) 센서 테스트나, 회로 프로토타입을 작성할 때 사용됩니다.	21 1EA 능동부저 Signal 단자가 HIGH 일 때 약 2.5kHz의 음이 발생됩니다.	22 1EA 5V 1채널 릴레이 모듈 아두이노의 디지털 핀과 모듈 하단의 IN 핀들을 연결해 릴레이를 제어할 수 있는 모듈입니다.	23 1EA 8x8 도트 매트릭스 모듈 LED로 다양한 연출을 할 수 있습니다.	24 1EA 4x4 16 키패드 모듈 16개의 버튼을 사용할 수 있습니다.

<div>25</div> <div>1EA</div> <div></div> <div>무선 리모콘 키트</div> <div>핵파선을 사용해서 리모콘 기능을 구현할 수 있습니다.</div>	<div>26</div> <div>2EA</div> <div></div> <div>가열기 센서 스위치</div> <div>센서의 가열기에 따라 스위치 역할을 합니다.</div>	<div>27</div> <div>1EA</div> <div></div> <div>or</div> <div>사운드 센서 모듈</div> <div>아두이노와 호환되는 사운드센서 모듈입니다.</div>	<div>28</div> <div>1EA</div> <div></div> <div>불꽃 센서</div> <div>근거리 화재, 불꽃을 감지하는 센서입니다.</div>	<div>37</div> <div>1EA</div> <div></div> <div>DC 5V 스텝 모터</div> <div>28BYJ 28BYJ48 스텝 모터 중 저렴한 편에 속하는 모델입니다. 5개의 핀을 사용합니다.</div>	<div>38</div> <div>1EA</div> <div></div> <div>DS1302 RTC 모듈</div> <div>아두이노 등 마이크로컨트롤러에 사용이 가능합니다.</div>	<div>39</div> <div>1EA</div> <div></div> <div>아두이노 우노 프로토 쉼드</div> <div>UNO 보드에서 회로를 간단히 짜기 위해 보드 위에 얹어 사용하는 쉼드입니다.</div>	<div>40</div> <div>1EA</div> <div></div> <div>3축 가속도 센서 모듈</div> <div>가속도를 측정할수 있는 센서입니다.</div>
<div>29</div> <div>1EA</div> <div></div> <div>모터 드라이버 모듈</div> <div>L2N2003 스텝 모터 드라이버 모듈 5V ~ 12V를 사용할수합니다.</div>	<div>30</div> <div>1EA</div> <div></div> <div>LM35 온도 센서</div> <div>온도를 아날로그 값으로 출력합니다.</div>	<div>31</div> <div>1EA</div> <div></div> <div>수위 센서 모듈</div> <div>센서가 액체에 잠긴 정도를 아날로그 값으로 출력합니다.</div>	<div>32</div> <div>1EA</div> <div></div> <div>SG90 서보모터</div> <div>Vcc, GND, 신호선, 총 3개의 핀이 있습니다. 로봇팔이나 자동차, 비행기 조종에 사용합니다.</div>	<div>41</div> <div>1EA</div> <div></div> <div>5V DC모터</div> <div>5V DC모터</div>	<div>42</div> <div>1EA</div> <div></div> <div>인체 감지 센서 모듈</div> <div>핵파선을 이용해 움직임을 감지하는 센서입니다. 오션이 감지되면 HIGH 신호를 출력합니다.</div>	<div>43</div> <div>5EA</div> <div></div> <div>다이오드 1N4001</div> <div>다이오드 1N4001</div>	<div>44</div> <div>5EA</div> <div></div> <div>세라믹 캐패시터 (22pF)</div> <div>세라믹 캐패시터 (22pF)</div>
<div>33</div> <div>1EA</div> <div></div> <div>초음파 거리 센서 모듈</div> <div>5V를 사용하여 만직 거리는 2cm에서 500cm입니다.</div>	<div>34</div> <div>1EA</div> <div></div> <div>조이스틱 모듈</div> <div>기본적으로 조이스틱 모듈은 두개의 가변저항이 서로 수직으로 회전하는 형태로 되어있습니다.</div>	<div>35</div> <div>1EA</div> <div></div> <div>온습도 센서 모듈</div> <div>아두이노 온습도 센서중 가장 대중적으로 사용되는 DHT11 디지털 센서입니다.</div>	<div>36</div> <div>1EA</div> <div></div> <div>RGB LED 모듈</div> <div>RGB LED 모듈로 RGB LED 세개를 하나로 묶은 상품입니다.</div>	<div>45</div> <div>5EA</div> <div></div> <div>세라믹 캐패시터 (1uF)</div> <div>세라믹 캐패시터 (1uF)</div>	<div>46</div> <div>5EA</div> <div></div> <div>트랜지스터 2N2222</div> <div>트랜지스터 2N2222</div>	<div>47</div> <div>5EA</div> <div></div> <div>트랜지스터 BC547</div> <div>트랜지스터 BC547</div>	<div>48</div> <div>5EA</div> <div></div> <div>트랜지스터 BC557</div> <div>트랜지스터 BC557</div>
<div>49</div> <div>2EA</div> <div></div> <div>전해 캐패시터 (50V 10uF)</div> <div>전해 캐패시터 (50V 10uF)</div>	<div>50</div> <div>2EA</div> <div></div> <div>전해 캐패시터 (50V 100uF)</div> <div>전해 캐패시터 (50V 100uF)</div>	<div></div>					