



아주대학교

코딩 교육 봉사 동아리 SWeat x 00고등학교

5주차

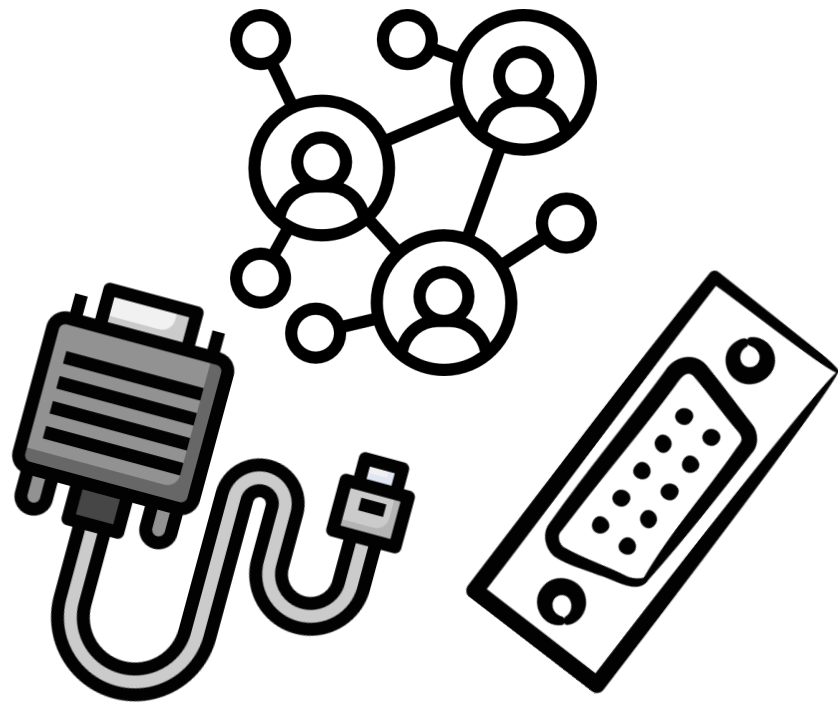
LCD,

LED + 7segment 제어

발표자 | 000 튜터, 000 튜터

4주차에 배운 내용 복습하기

Review



시리얼 통신

:직렬로 정보를 주고받는 통신 방법

※연속적인 정보가 채널이나 버스를 타고 이동해 한 번에 하나씩 정보를 전송하는 방법



스케치에서 이 돋보기를 클릭하면 시리얼 모니터를 실행할 수 있습니다.

4주차에 배운 내용 복습하기

Review

```
Serial.begin(9600);
```

: 9600의 기본 보드 속도로
시리얼 통신을 시작하겠다는 코드

```
Serial.available()
```

: Serial 포트에서 읽을 수 있는
바이트 수를 반환하는 코드

```
Serial.end();
```

: 시리얼 통신을 종료하는 코드

```
Serial.print("serial");
```

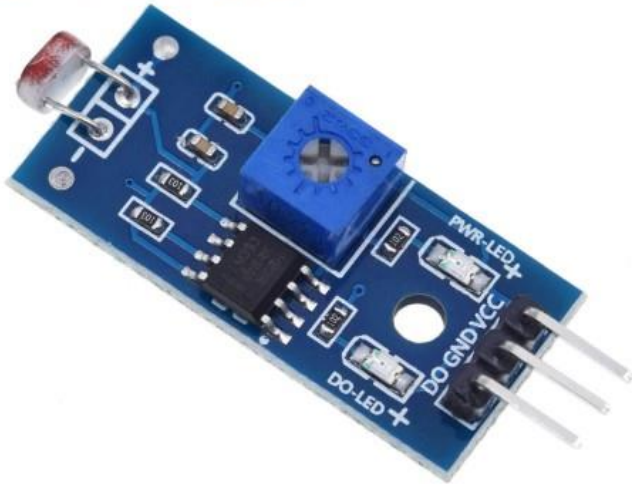
: 시리얼 모니터에 출력할 문구를 정해주는 코드

```
Serial.read();
```

: 수신한 데이터를 한 바이트 씩 읽어오는 코드

4주차에 배운 내용 복습하기

Review



적외선 센서

: 적외선을 이용해 온도, 압력, 방사선의 세기 등의 물리량이나 화학량을 감지해 신호처리가 가능한 전기량으로 변환하는 장치

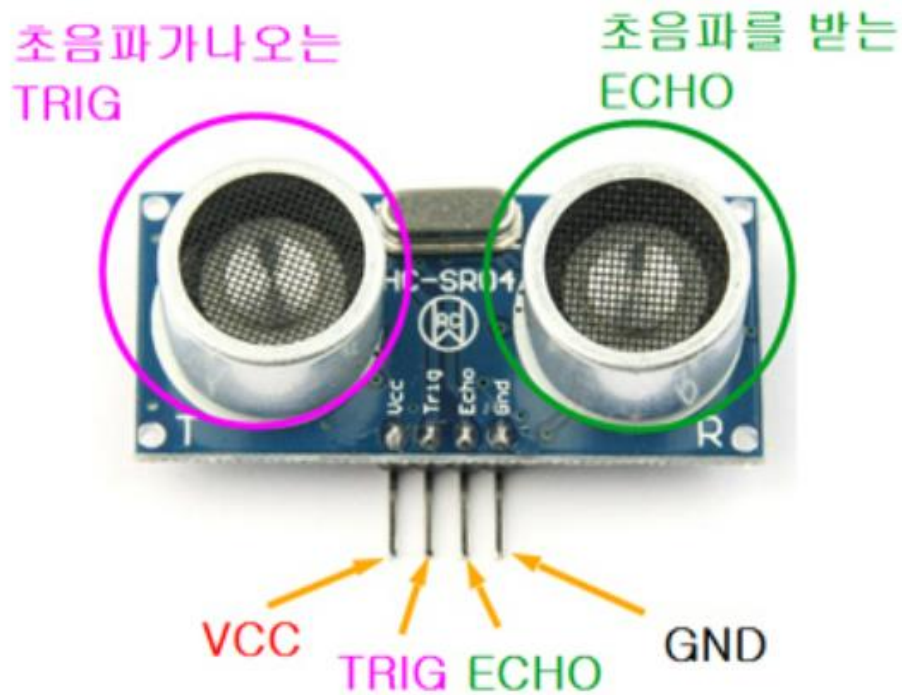
※ 적외선 센서는 거리에 따라서 전압이 변화
가변저항을 이용하여 센서의 감도를 조절

(반시계 방향-감지 거리 ↓, 시계 방향-감지 거리 ↑)

Last Class. 적외선 감지, 적외선 센서로 LED 제어 실습 진행

4주차에 배운 내용 복습하기

Review



초음파 센서

: 초음파를 이용하여 물체와의 거리를
측정하는 센서

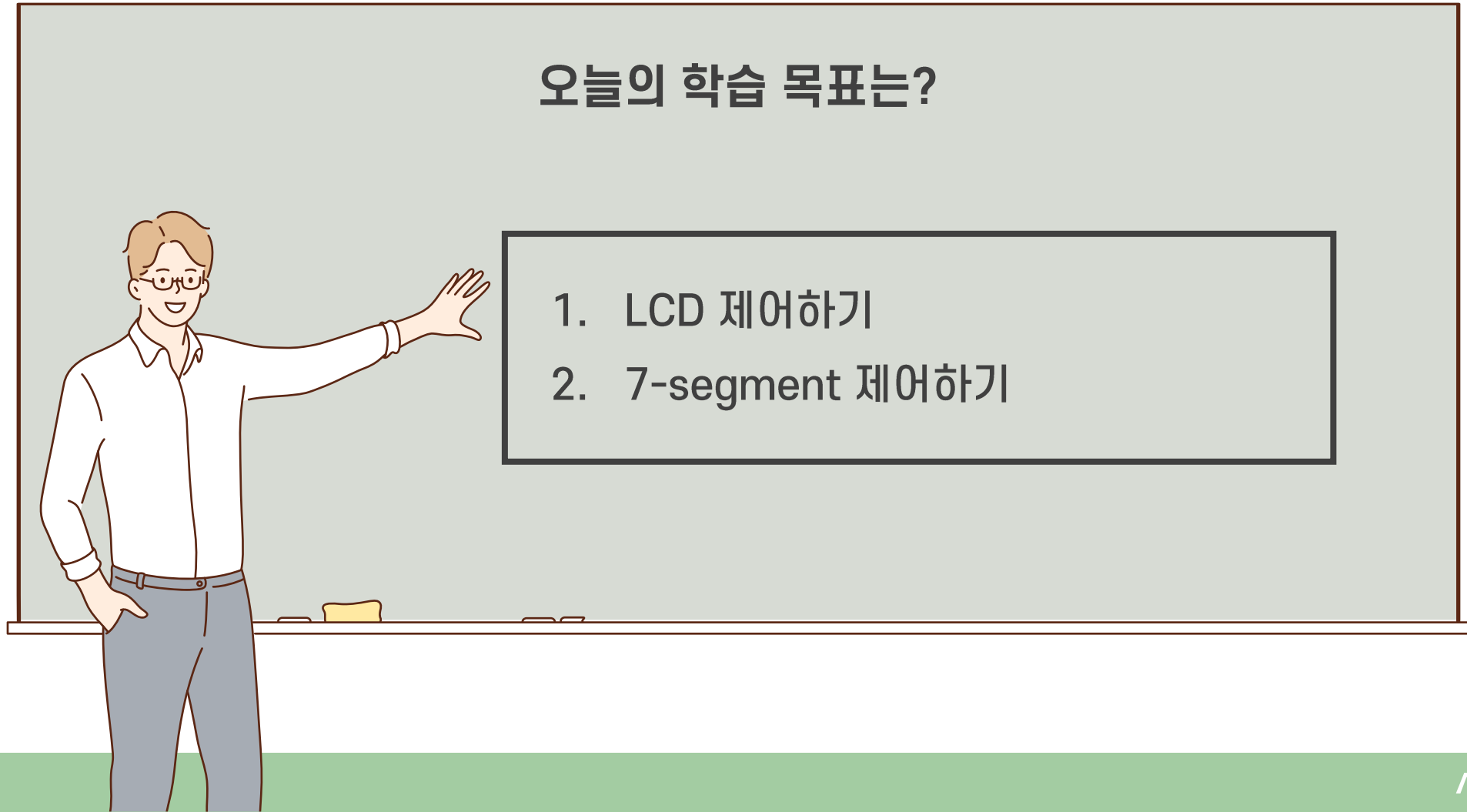
※ 초음파가 장애물에 반사되어 되돌아오면서
PWM 신호를 전달

초음파 모듈과 물체와의 거리는 $(\text{왕복시간} \mu\text{s} / 2) / 29$

Last Class. 물체 감지, LED 제어 실습 진행

학습 목표

Learning Objectives



“ LCD를 제어해 봅시다! ”

LCD 라이브러리 설치

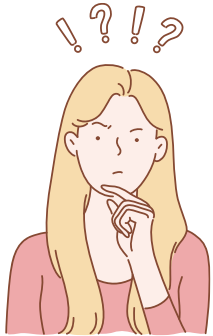
LCD 제어하기

I2C LCD를 제어하기 위한 라이브러리를 설치해요.



잠깐! 설치 전 Q&A

질문이 있어요!



라이브러리란 무엇일까요?

#include를 통해
해당 파일을 포함시켜
기능들을 사용해요!

Q

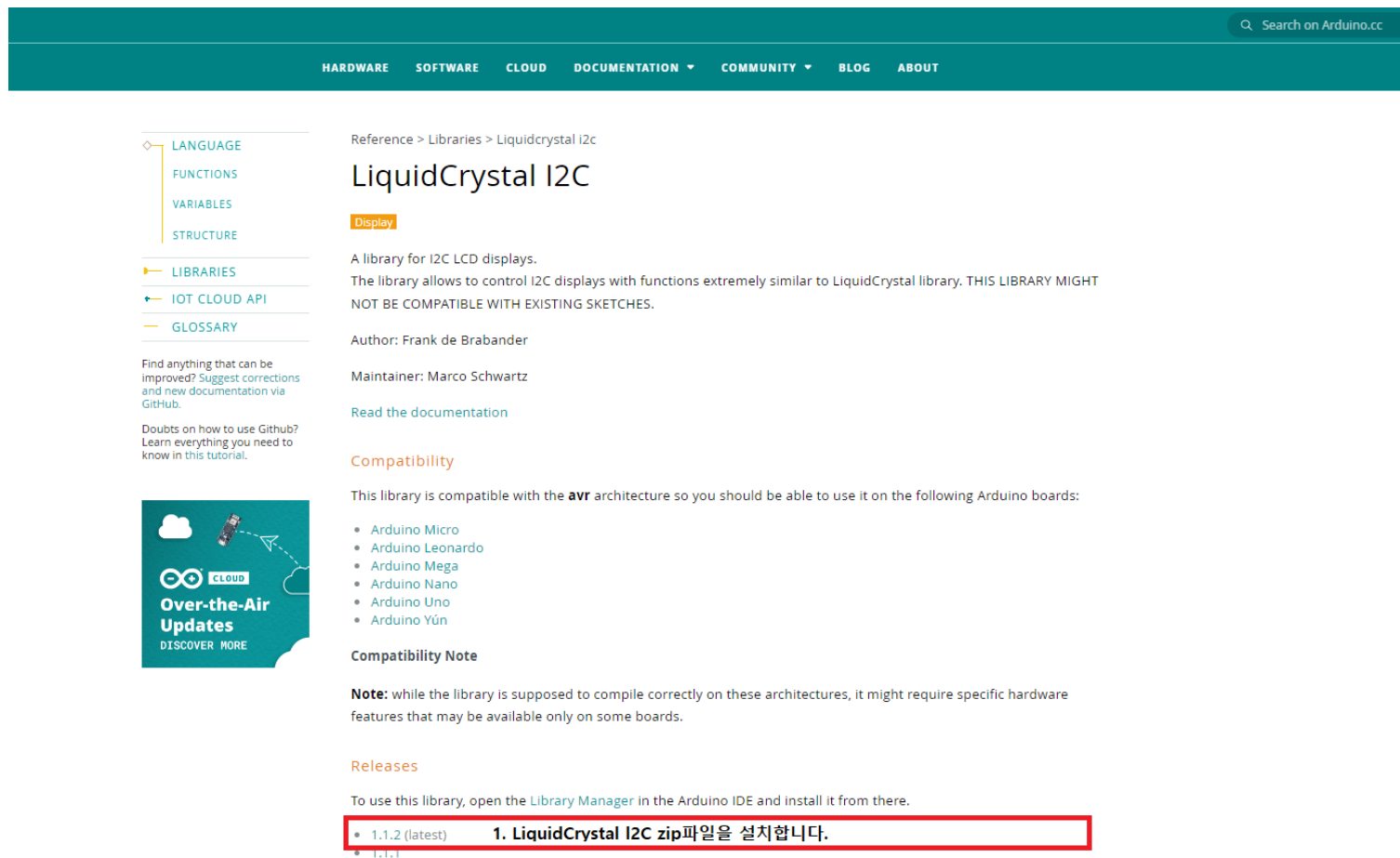
A

반복적으로 사용되거나 필요한 기능 혹은 값 등을
미리 정의하여 사용할 수 있도록 제공하는 것입니다.



LCD 라이브러리 설치

LCD 제어하기



Reference > Libraries > LiquidCrystal I2C

LiquidCrystal I2C

Display

A library for I2C LCD displays.
The library allows to control I2C displays with functions extremely similar to LiquidCrystal library. THIS LIBRARY MIGHT NOT BE COMPATIBLE WITH EXISTING SKETCHES.

Author: Frank de Brabander
Maintainer: Marco Schwartz
[Read the documentation](#)

Compatibility

This library is compatible with the **avr** architecture so you should be able to use it on the following Arduino boards:

- Arduino Micro
- Arduino Leonardo
- Arduino Mega
- Arduino Nano
- Arduino Uno
- Arduino Yún

Compatibility Note

Note: while the library is supposed to compile correctly on these architectures, it might require specific hardware features that may be available only on some boards.

Releases

To use this library, open the [Library Manager](#) in the Arduino IDE and install it from there.

- 1.1.2 (latest) **1. LiquidCrystal I2C zip파일을 설치합니다.**
- 1.1.1

링크([LiquidCrystal I2C - Arduino Reference](#))에 접속하여
[LiquidCrystal I2C 라이브러리 zip] 파일을 다운로드합니다.

LCD 라이브러리 설치 – Arduino 라이브러리 매니저 설치

LCD 제어하기

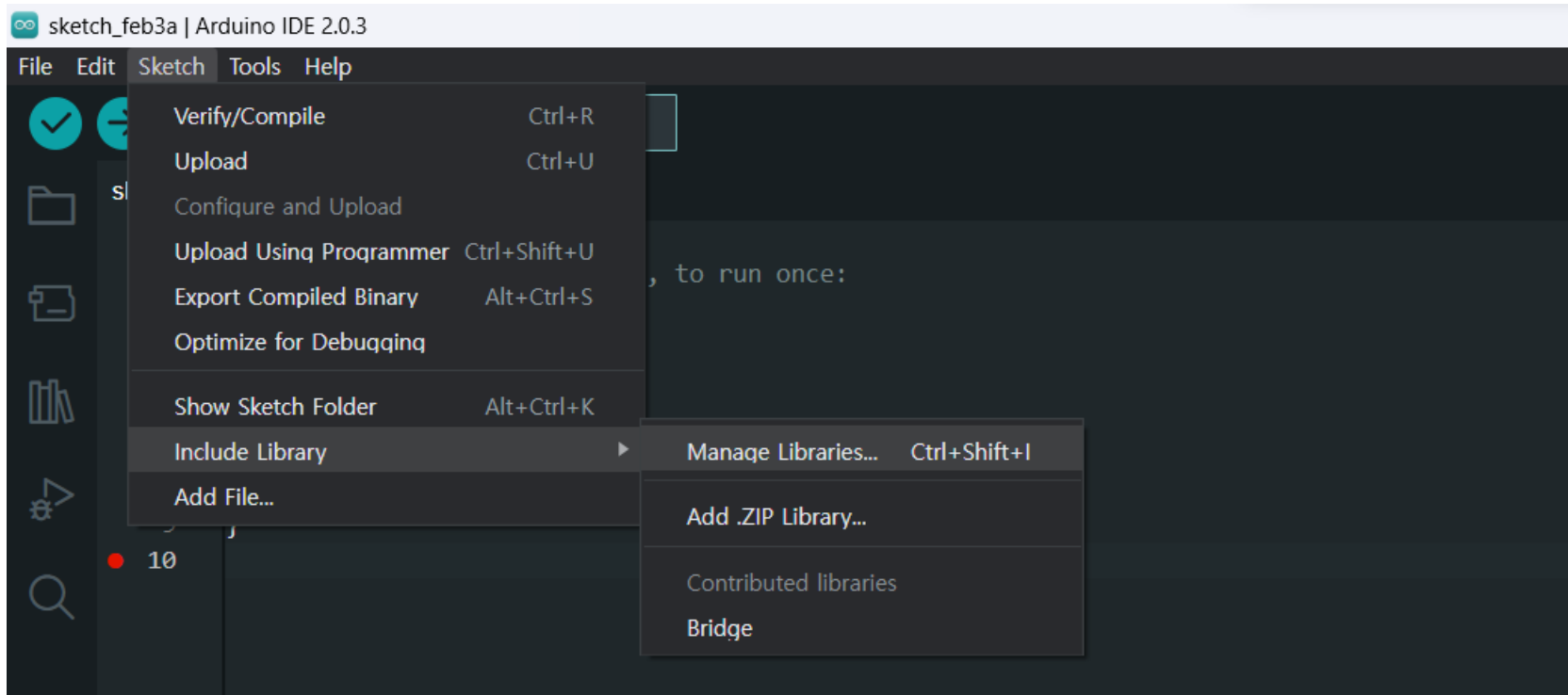
다운받은 라이브러리를 사용하기 위한
[Bridge for Arduino library manager] 를 설치해요.



Arduino IDE가 컴퓨터의 저장공간에 접근하도록 도와줘요!

LCD 라이브러리 설치 – Arduino 라이브러리 매니저 설치

LCD 제어하기



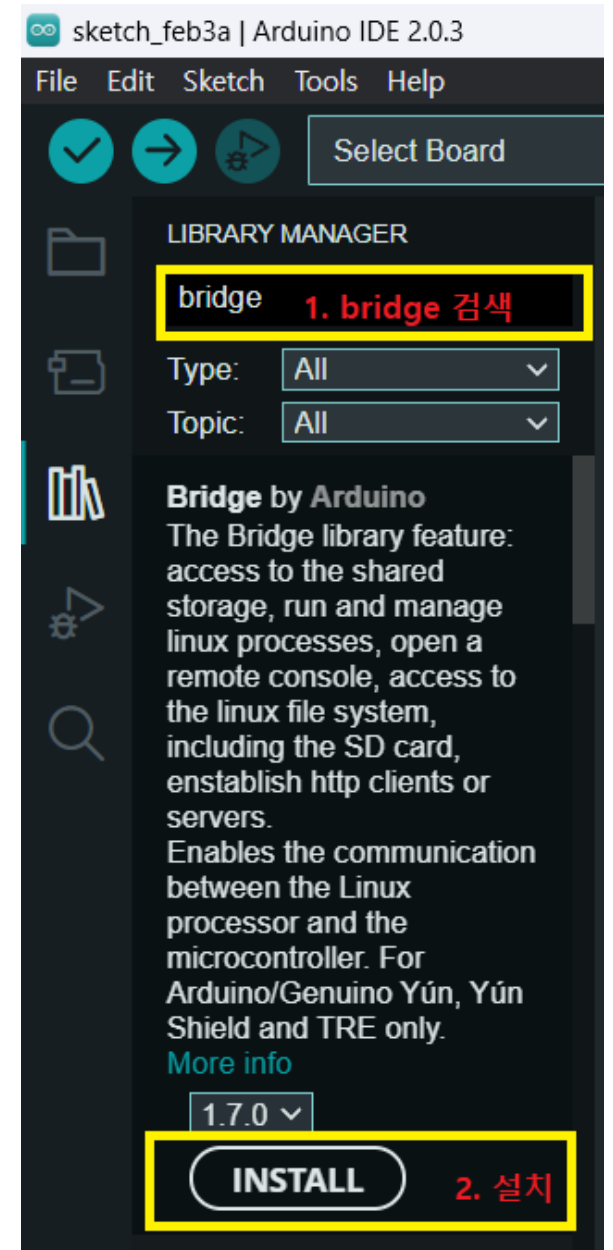
Arduino IDE에서

[Sketch]-[Include Library]-[Manager Libraries]를 클릭합니다.

LCD 라이브러리 설치 – Arduino 라이브러리 매니저 설치

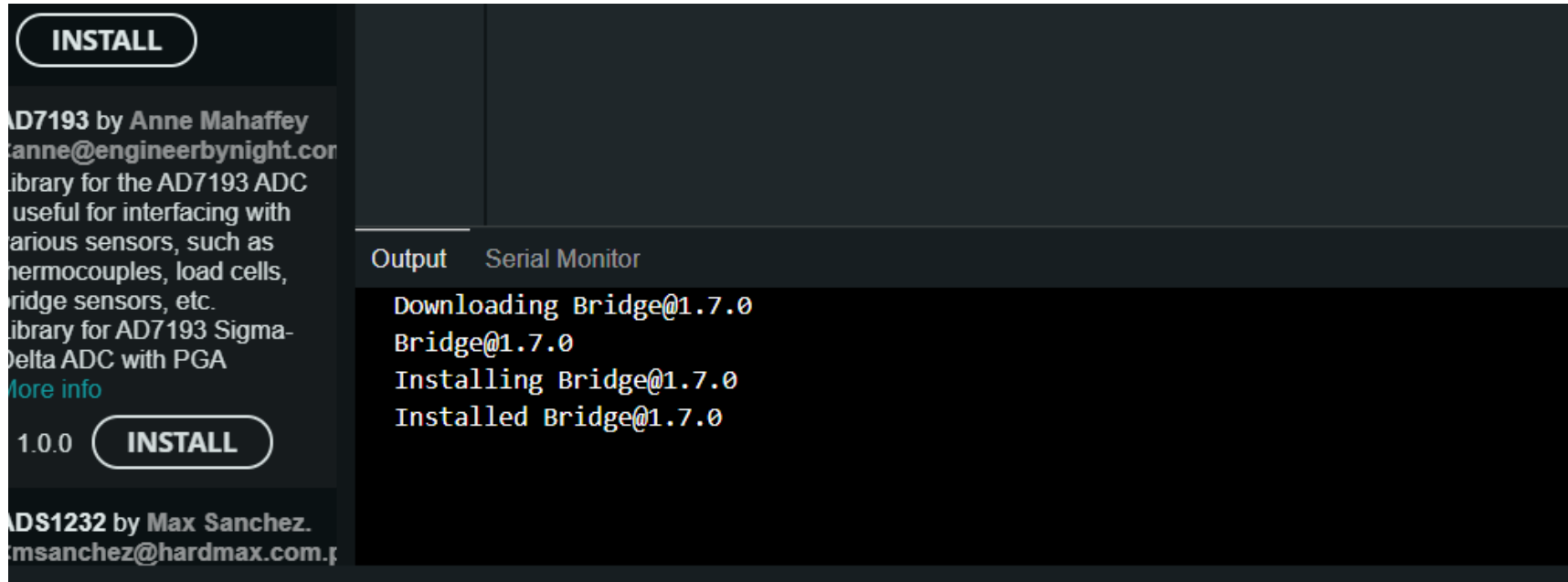
LCD 제어하기

bridge를 검색하여 Bridge by Arduino를 설치합니다.



LCD 라이브러리 설치 – Arduino 라이브러리 매니저 설치

LCD 제어하기

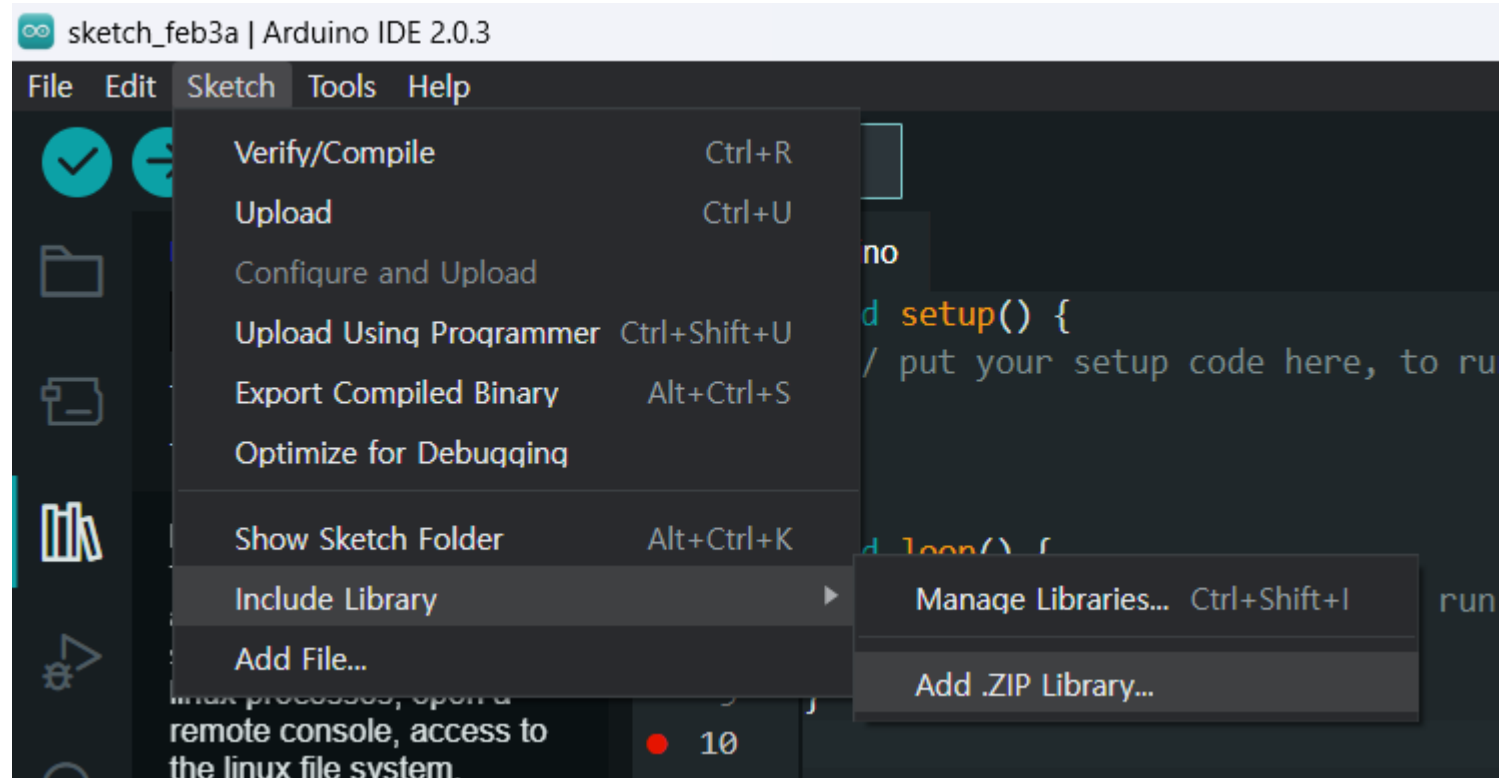


성공적으로 설치된 화면입니다.

LCD 라이브러리 설치 – Arduino 라이브러리 매니저

LCD 제어하기

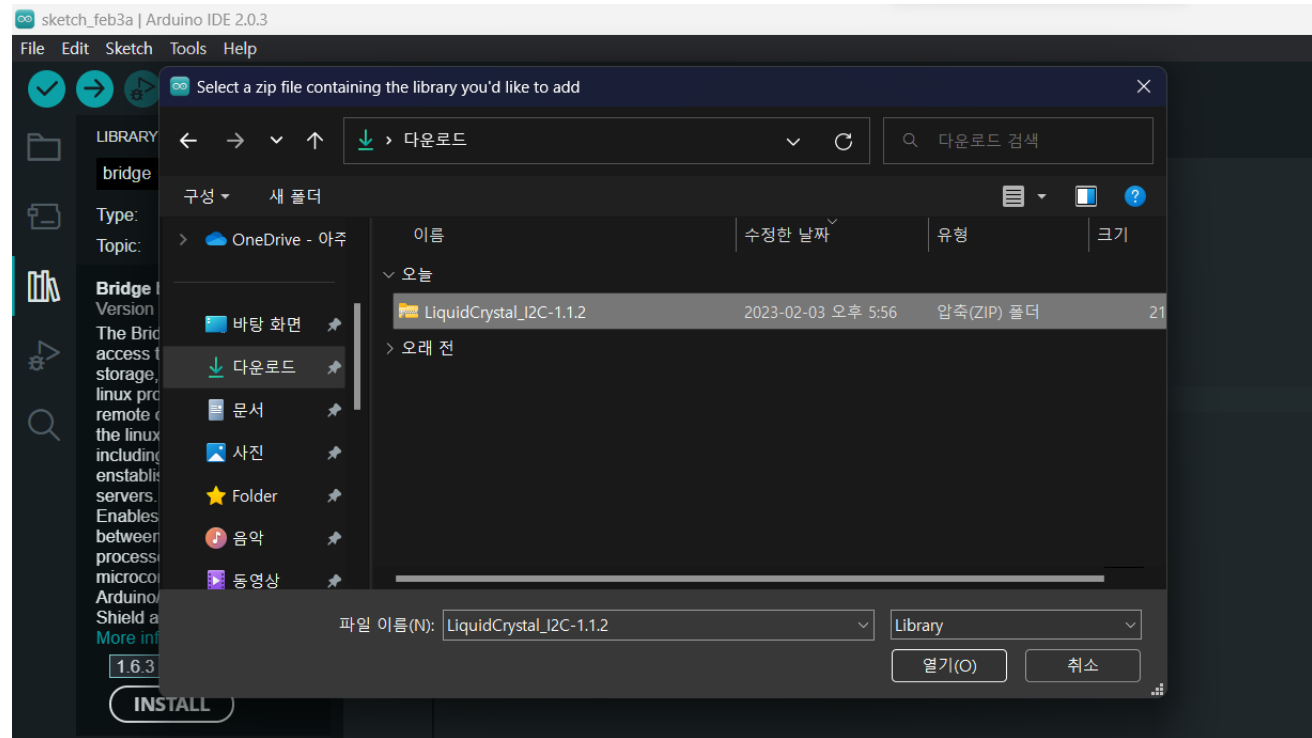
LiquidCrystal I2C
라이브러리를
사용할 수 있도록
zip 파일을 찾아야 합니다.



[Sketch]-[Include Library]-[Add.ZIP Library]를 클릭합니다.

LCD 라이브러리 설치 – Arduino 라이브러리 매니저

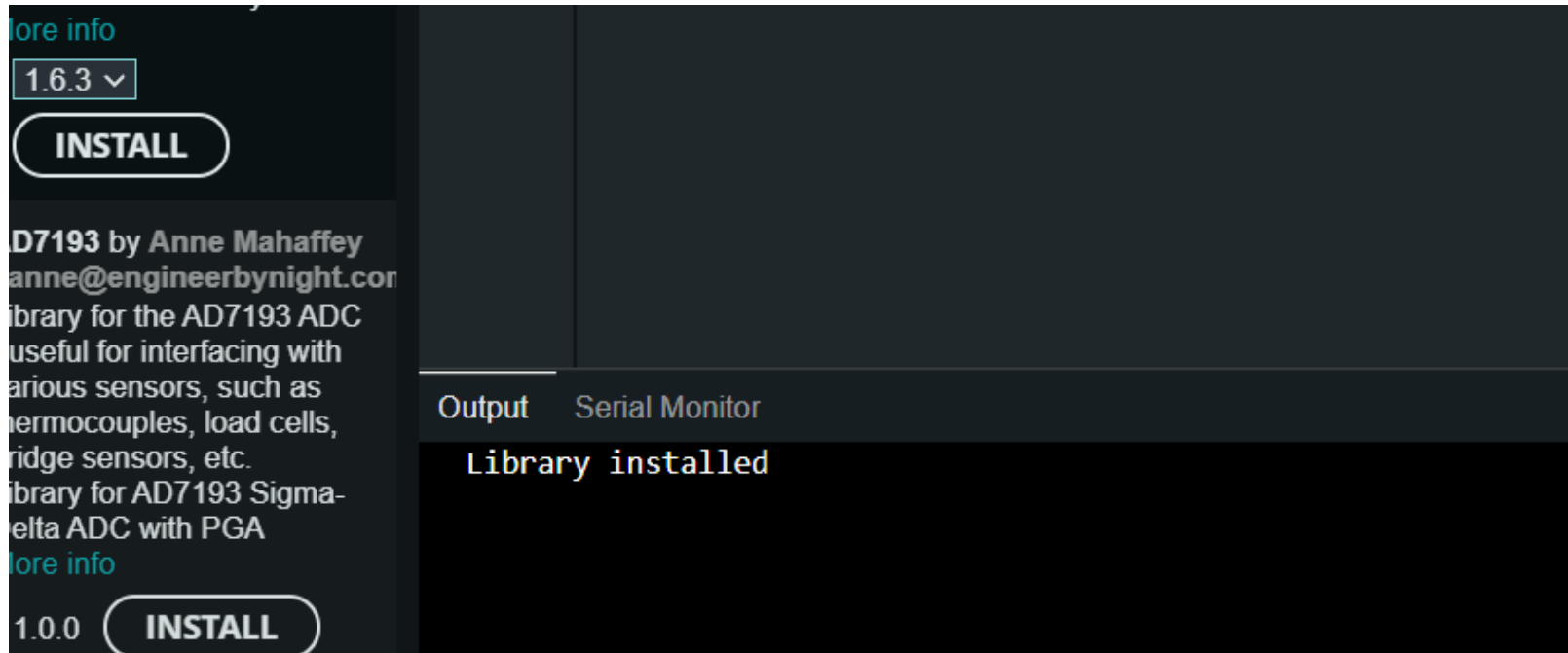
LCD 제어하기



다운로드한 zip파일을 선택한 후 [열기]를 클릭하여 완료합니다.

LCD 라이브러리 설치 – Arduino 라이브러리 매니저

LCD 제어하기



성공적으로 라이브러리 설치가 완료된 화면입니다!

“

본격적으로 LCD를 제어해 볼까요?

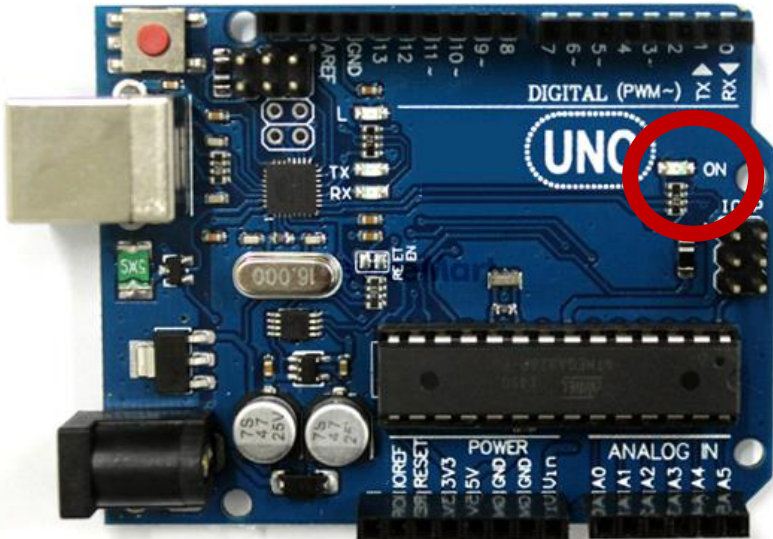
”

실습을 준비해볼까요?

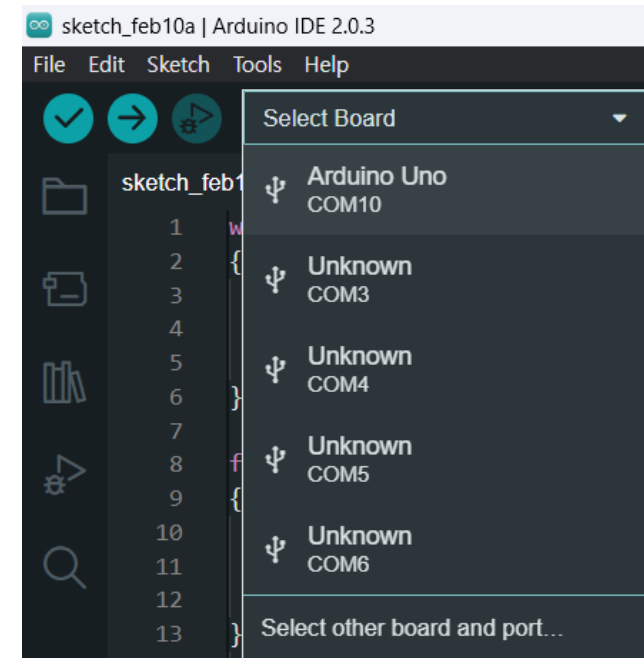
LCD 제어하기

매번 실습 때 해주어야 해요 ~

컴퓨터에 아두이노 보드를 연결하고, 아두이노 IDE에서 연결된 Arduino Uno 보드와 포트번호를 선택합니다.



ON에 불이 들어와야 해요.



보드
포트번호

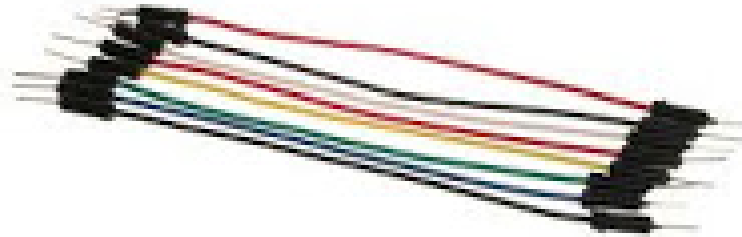
<실습1> LCD 제어하기

LCD 제어하기

실습 준비물



LCD x 1



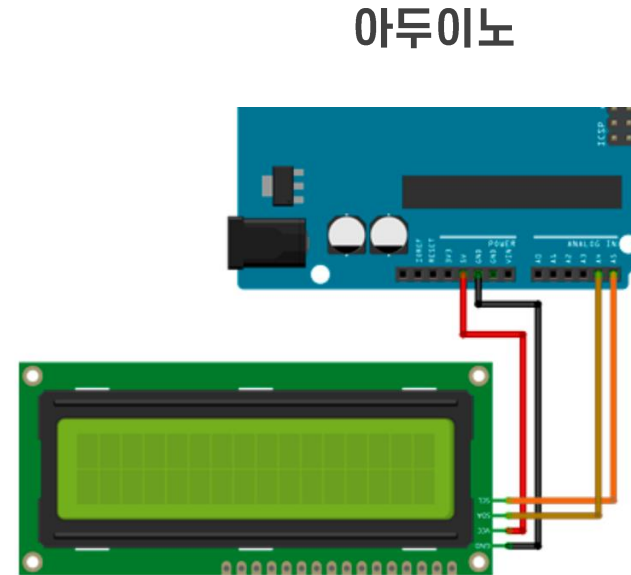
점퍼 케이블 x 4

<실습1> LCD 제어하기: 아두이노에 LCD 1개를 연결하고 화면에 글자를 출력해봅시다.

LCD 제어하기

이렇게 해봐요!

1. LCD의 GND핀과 아두이노 보드의 전원 영역의 GND핀을 점퍼선으로 연결합니다.
2. LCD의 VCC핀과 아두이노 보드의 5V핀을 점퍼선으로 연결합니다..
3. LCD의 SDA핀과 아두이노 보드의 A4핀을 점퍼선으로 연결합니다.
4. LCD의 SCL핀과 아두이노 보드의 A5핀을 점퍼선으로 연결합니다.



아두이노 우노보드	LCD 모듈
GND	GND
5V	VCC
A4	SDA
A5	SCL

회로도

결선표

잠깐! Sketch 작성 전에 확인해요 - LCD 주소 확인하기

LCD 제어하기

코드 작성에 사용할 I2C LCD의 **주소**를 확인해요!



잠깐! Sketch 작성 전에 확인해요 - LCD 주소 확인하기

LCD 제어하기

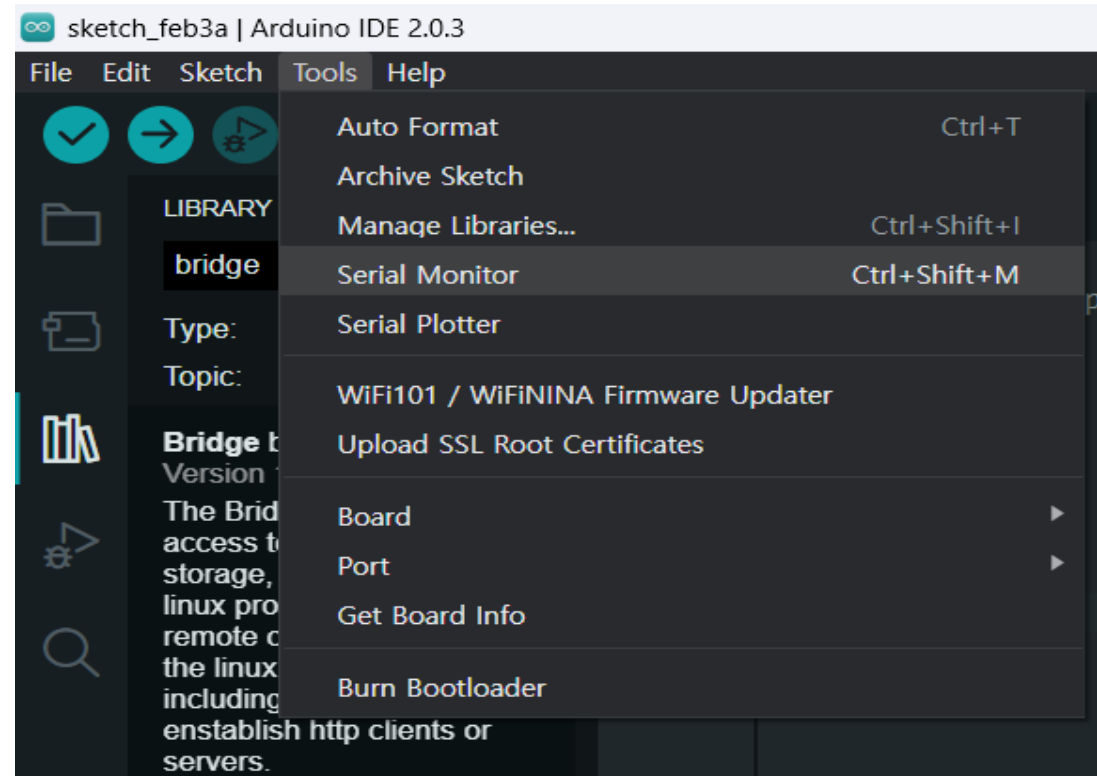
```
1  #include <Wire.h>
2
3
4  void setup()
5  {
6      Wire.begin();
7
8      Serial.begin(9600);
9      while (!Serial);
10     Serial.println("\nI2C Scanner");
11 }
12
13
14 void loop()
15 {
16     byte error, address;
17     int nDevices;
18
19     Serial.println("Scanning...");
20
21     nDevices = 0;
22     for(address = 1; address < 127; address++)
23     {
24         Wire.beginTransmission(address);
25         error = Wire.endTransmission();
26
```

```
26
27     if (error == 0)
28     {
29         Serial.print("I2C device found at address 0x");
30         if (address<16)
31             Serial.print("0");
32         Serial.print(address,HEX);
33         Serial.println(" !");
34
35         nDevices++;
36     }
37     else if (error==4)
38     {
39         Serial.print("Unknown error at address 0x");
40         if (address<16)
41             Serial.print("0");
42         Serial.println(address,HEX);
43     }
44 }
45 if (nDevices == 0)
46     Serial.println("No I2C devices found\n");
47 else
48     Serial.println("done\n");
49
50 delay(5000);
51 }
52
```

Sketch에서 LCD 주소 Scanner 코드를 실행합니다. <https://playground.arduino.cc/Main/I2cScanner/>

🔔 잠깐! Sketch 작성 전에 확인해요 - LCD 주소 확인하기

LCD 제어하기



[tools]-[serial monitor] 클릭 후 하단의 Serial monitor에서 주소를 확인합니다.

잠깐! Sketch 작성 전에 확인해요 - LCD 규격 확인하기

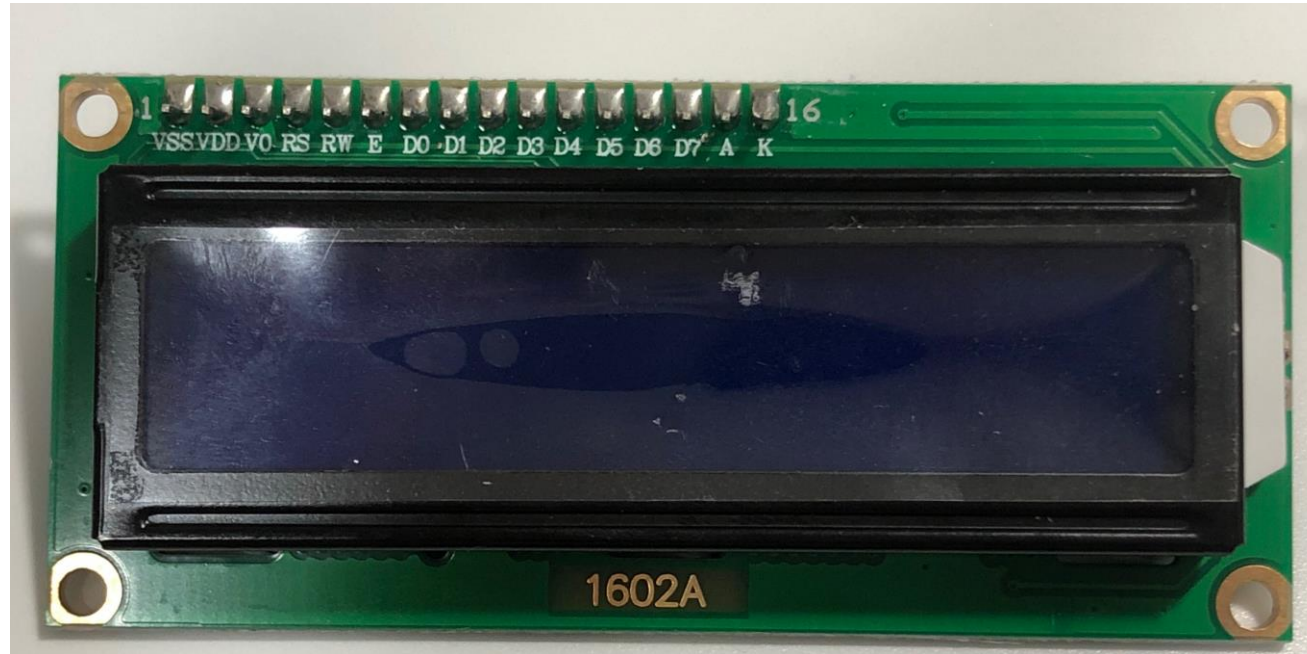
LCD 제어하기

LCD 모니터의 열과 행 수를 부품명을 통해 확인해요.



실습 전 설정하기

LCD 모니터의 열과 행 수 확인하기



1602A → 2행 16열

스케치 작성

Sketch

1. 필요한 헤더파일들을 불러옵니다.
2. LCD의 기본정보를 입력합니다.
-> ICD(접근주소, lcd 열의 수, lcd 행의 수)
3. LCD 모니터에 출력할 글자를 입력합니다.

사용된 함수

lcd.init(); LCD 초기화

lcd.backlight(); LCD의 백라이트를 켜

lcd.setCursor(열, 행); 커서 또는 출력의 위치를
지정된 좌표로 이동

lcd.print(); LCD 화면에 값을 출력

```
1  #include <Wire.h>
2  #include <LiquidCrystal_I2C.h>
3
4  LiquidCrystal_I2C lcd(0x27,16,2);
5
6  void setup()
7  {
8      lcd.init();
9      lcd.backlight();
10     lcd.setCursor(0,0);
11     lcd.print("Hello,world!");
12     lcd.setCursor(3,1);
13     lcd.print("Sweat!");
14 }
15
16 void loop()
17 {
18 }
```

Review: **헤더파일** < .h>

헤더 파일을 불러와야만

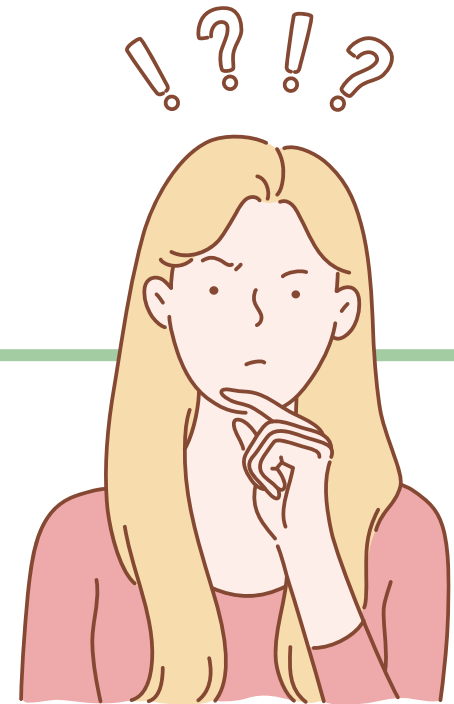
관련 함수들을 사용할 수 있습니다

글자가 출력되지 않아요!

글자 출력 오류 해결

Q

화면만 반짝이거나, 네모칸만 출력되는 등
글자가 출력되지 않으면 어떻게 해야할까요?



🔔 글자가 출력되지 않아요!

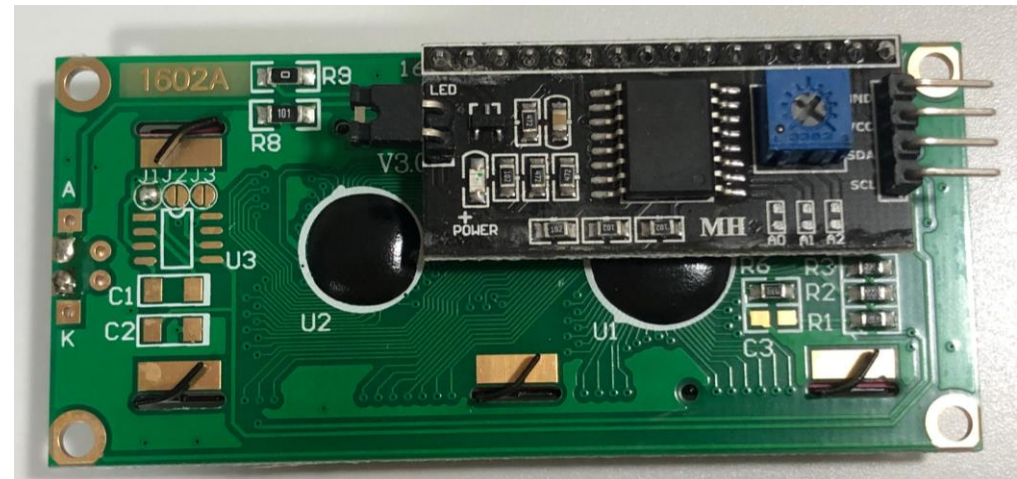
글자 출력 오류 해결

A

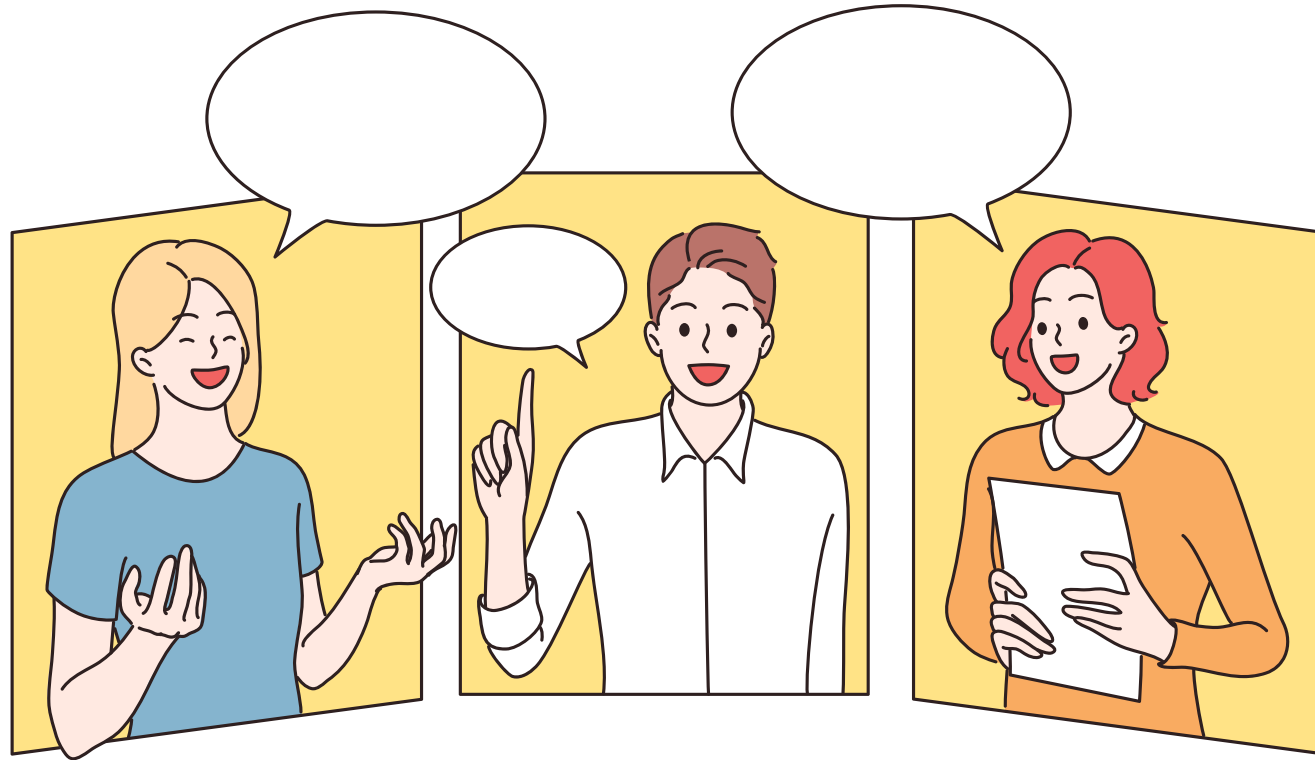
1. I2C LCD의 주소를 다시 확인합니다.
2. 주소가 맞지만 출력되지 않을 경우,
저항값의 세기를 조절합니다.



LCD 모니터를 뒤집어 파란색 블록 안의
십자 나사를 세심하게 조금씩 돌려가며
저항값을 조절하고, 변화가 보이면
➡ 아두이노 재연결



잠깐 쉬는 시간 !



“

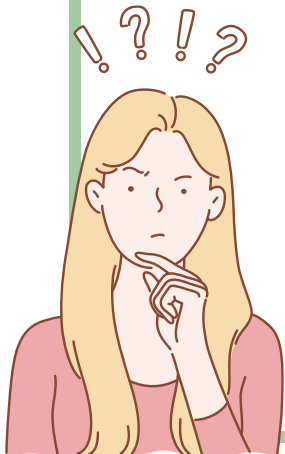
7-segment를 제어해 봅시다!

”

7-segment가 무엇일까요?

7-segment LED 제어하기

7-segment가
무엇인가요?



Q

엘리베이터, 디지털 시계 등에서
볼 수 있는 숫자 모양을 아시나요?



이 하나의 숫자 LED는 총 7개의
선분 디스플레이로 구성되어 있어
7-segment라고 불립니다.

A

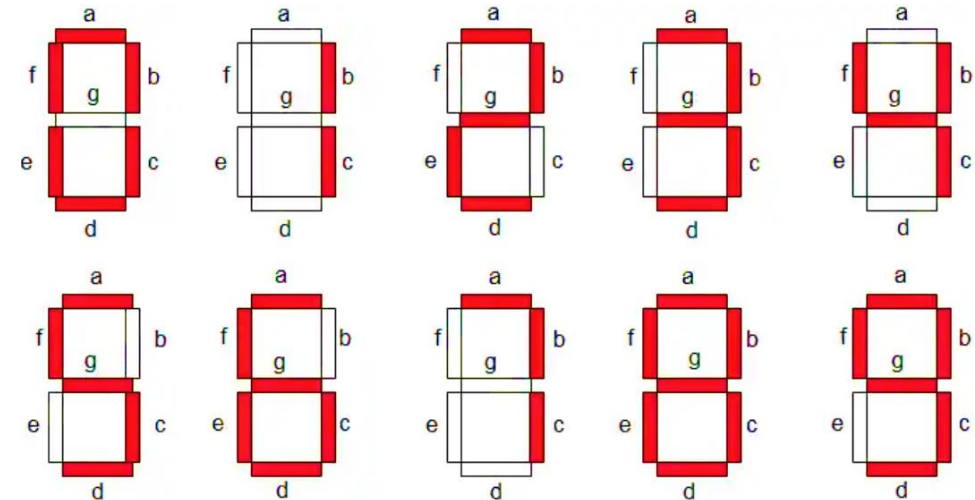
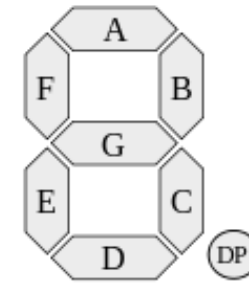
다음 페이지에서 자세히 살펴봅시다 ~:D



7-segment가 무엇일까요?

7-segment LED 제어하기

**A~G까지 7개의
선분 디스플레이가 있습니다.**
**오른쪽 아래의 DP(Dot Point)까지
총 8개의 디스플레이를
제어해 봅시다.**

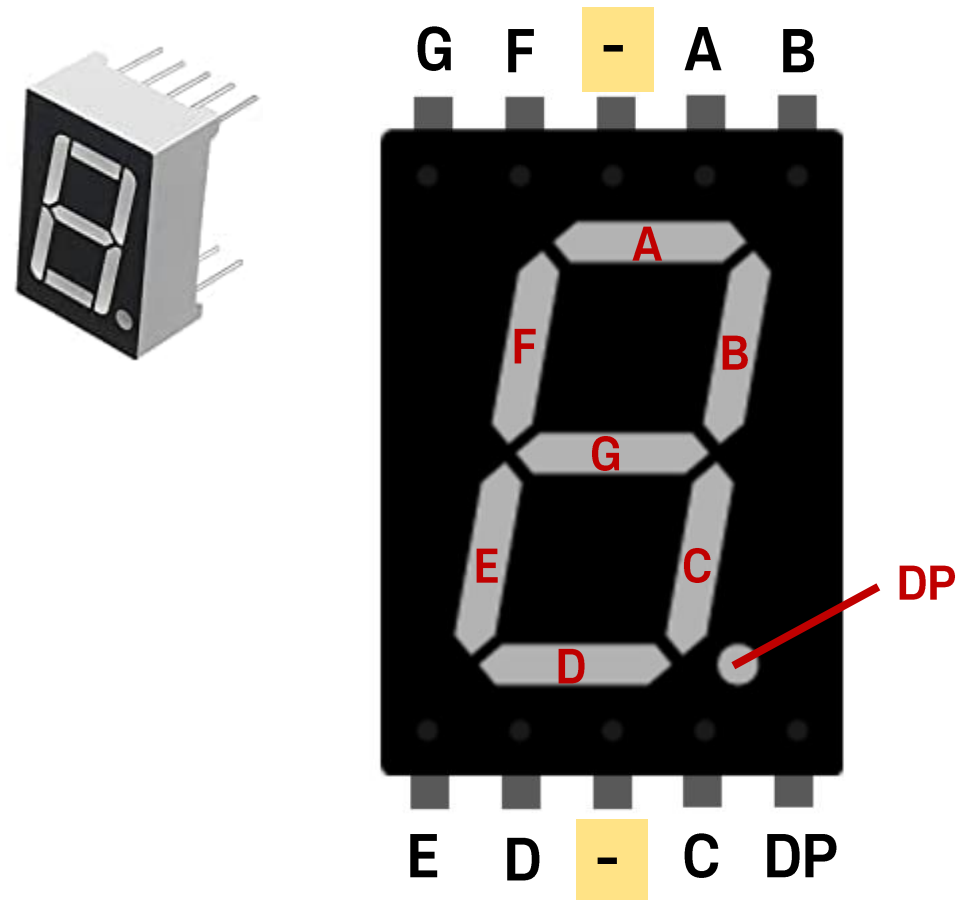


7-segment가 무엇일까요?

7-segment LED 제어하기

실습 때 사용할 7-segment는
음극(-, GND)에 연결해야 하는
'캐소드 타입'입니다.

※ 다른 타입으로는 '아노드 타입'이 있습니다. 아노드 타입은
양극(+, 5V)에 연결해야 합니다. 부품에 따라 다르니 모델명을
확인해야 합니다.



“

실습을 해 볼까요?

”

<실습2> 7-segment 제어하기

7-segment 제어하기

실습 준비물

7-segment x 1

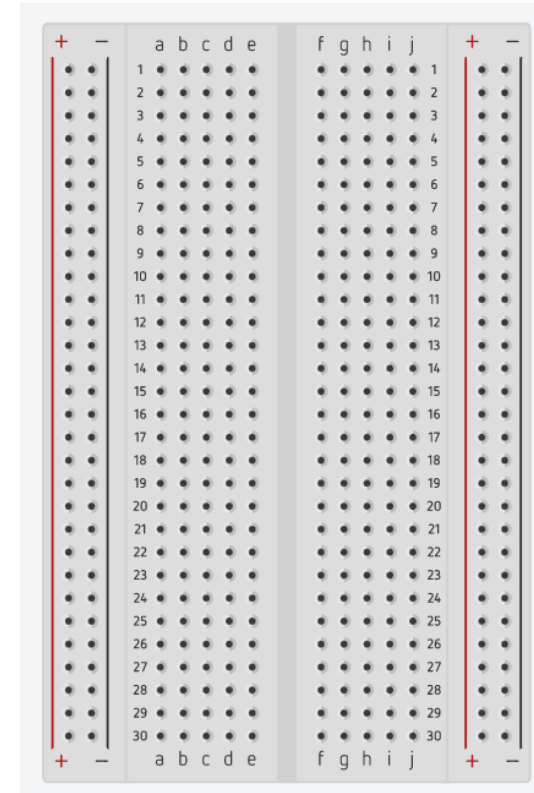


브레드보드 x 1

저항(220k옴) x 2



점퍼선 x 12

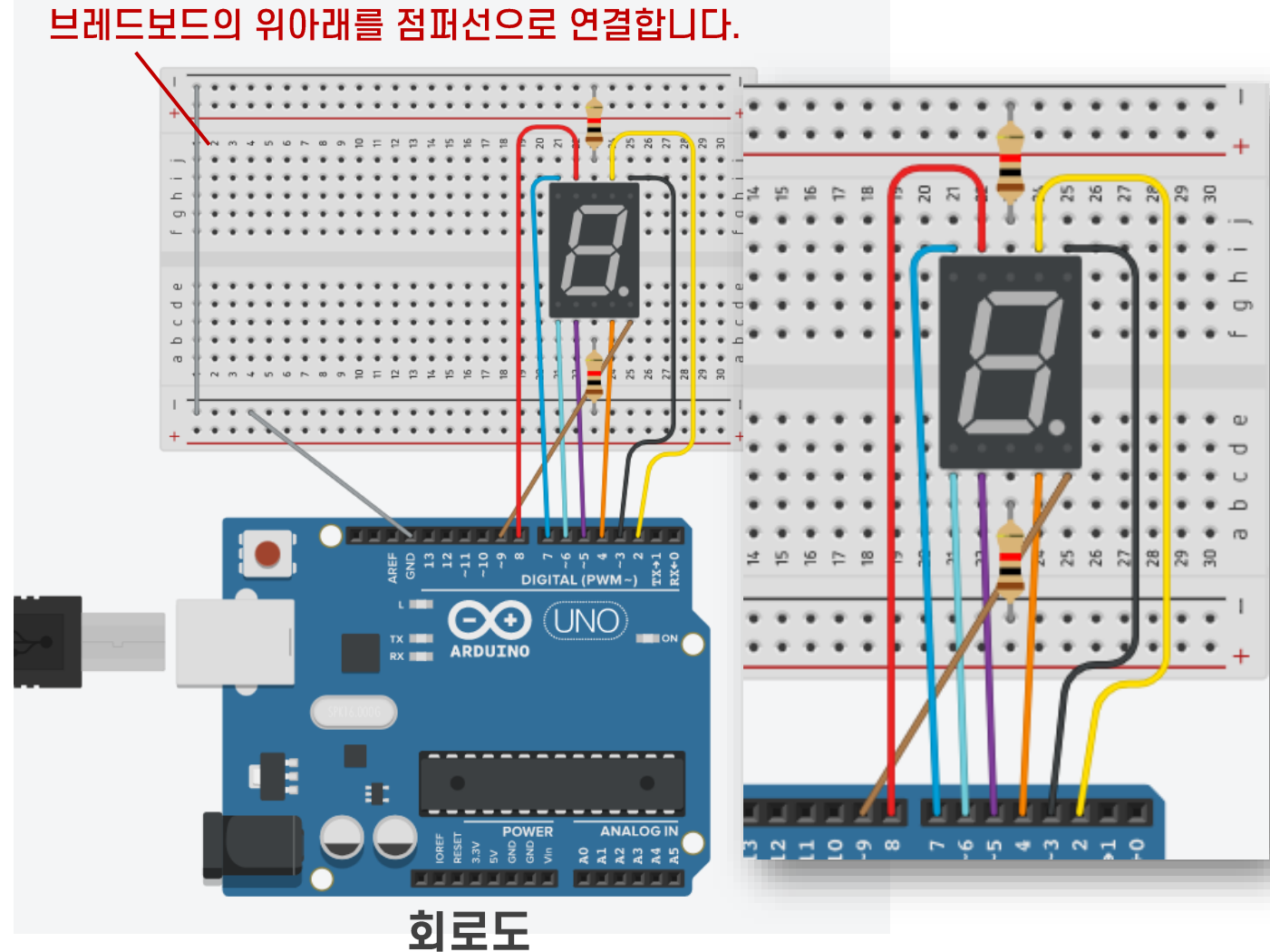


<실습2> 7-segment 제어하기: 7-segment에 0 ~ 9 까지의 숫자를 출력해 봅시다.

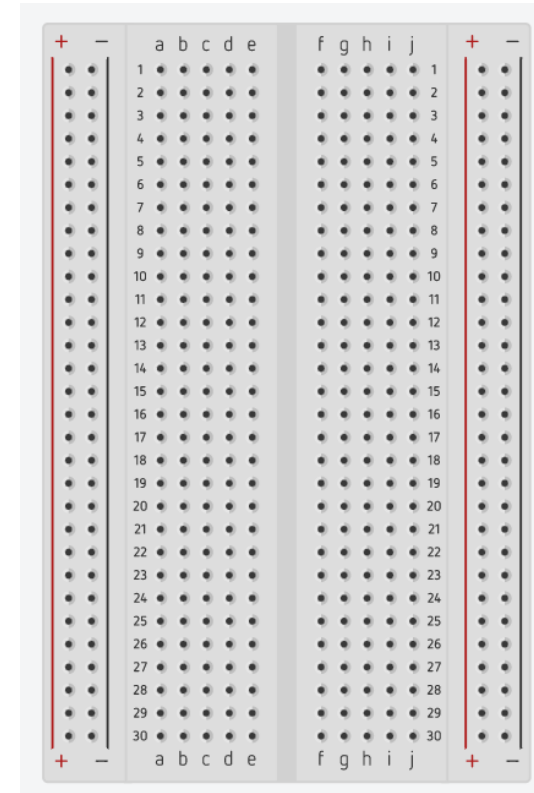
7-segment 제어하기

이렇게 해봐요!

1. 브레드보드를 가로로 두고 중심선을 기준으로 위 아래에 7-segment를 꽂아 연결합니다.
2. 저항을 이용해 7-segment의 가운데 핀들을 각각 위아래의 음극으로 연결합니다.
3. 나머지 핀들은 디지털영역에 연결합니다.
4. 브레드보드의 위아래 음극을 점퍼선으로 연결하고 아두이노의 GND에 연결합니다.



7-segment 제어하기



스케치 작성

Sketch

1. `segmentLEDs[]`: 핀 번호를 저장합니다.
`segmentLEDsNum`: 핀의 개수를 저장합니다.
`digitForNum[10][8]`: 10개의 숫자를 8핀을 이용하여 출력할 수 있도록 2차원 배열로 저장합니다.
1: HIGH(켜짐), 0:LOW(꺼짐)
2. 8개의 핀을 출력핀으로 설정합니다.
3. 0~9까지 숫자를 1초씩 출력하고 모두 출력한 후 1초간 대기합니다.

사용된 함수

`pinMode`(핀 번호, 모드)

: 해당 핀을 출력(OUTPUT) 또는 입력(INPUT) 모드로 설정한다.

`digitalWrite`(핀 번호, 값)

: 해당 디지털 핀에 HIGH(전기공급) 또는 LOW(전기차단)의 값을 주는 명령어

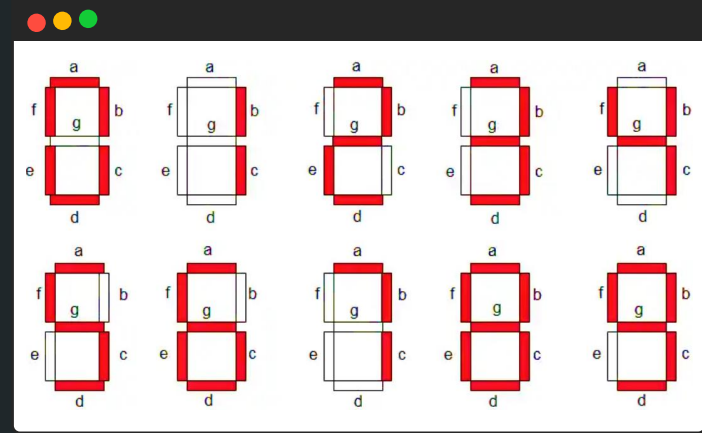
💡 Tip: 배열(1차원, 2차원)

자료형 변수명[개수] = { , , ... }

자료형 변수명[행][열] =

{ {..},{..},... }

```
1 int segmentLEDs[] = {2, 3, 4, 5, 6, 7, 8, 9};
2 int segmentLEDsNum = 8;
3
4 int digitForNum[10][8] = {
5 // {A, B, C, D, E, F, G, H}
6 {1, 1, 1, 1, 1, 0, 1, 0}, //0
7 {0, 1, 1, 0, 0, 0, 0, 0}, //1
8 {1, 1, 0, 1, 1, 1, 0, 0}, //2
9 {1, 1, 1, 1, 0, 1, 0, 0}, //3
10 {0, 1, 1, 0, 0, 1, 1, 0}, //4
11 {1, 0, 1, 1, 0, 1, 1, 0}, //5
12 {1, 0, 1, 1, 1, 1, 1, 0}, //6
13 {1, 1, 1, 0, 0, 0, 0, 0}, //7
14 {1, 1, 1, 1, 1, 1, 1, 0}, //8
15 {1, 1, 1, 1, 0, 1, 1, 0} //9
16 };
17
18
19 void setup() {
20 for (int i = 0 ; i < segmentLEDsNum ; i++) {
21     pinMode(segmentLEDs[i], OUTPUT);
22 }
23 }
24
25 void loop() {
26 for (int i = 0 ; i < 10 ; i++) {
27     for (int j = 0 ; j < segmentLEDsNum ; j++) {
28         digitalWrite(segmentLEDs[j], digitForNum[i][j]);
29     }
30     delay(1000);
31 }
32     delay(1000);
33 }
```



Review: 반복문 for ()
for(초기식; 조건식; 증감식) {
 // 조건이 참일 때 실행될 코드
}

Quiz를 맞춰 보아요!

Quiz



오늘 배운 내용을
잘 이해했는지
퀴즈를 풀어보아요 !

Quiz 1. lcd.setCursor(3,1)은 무엇을 의미할까요?

Answer.

커서 또는 ①의 위치를
지정된 ②로 이동시킨다.

즉, ③번째 열
④번째 행으로 이동시킨다.

① : , ② : , ③ : , ④ :

```
1  #include <Wire.h>
2  #include <LiquidCrystal_I2C.h>
3
4  LiquidCrystal_I2C lcd(0x27,16,2);
5
6  void setup()
7  {
8      lcd.init();
9      lcd.backlight();
10     lcd.setCursor(0,0);
11     lcd.print("Hello,world!");
12     lcd.setCursor(3,1);
13     lcd.print("Sweat!");
14 }
15
16 void loop()
17 {
18 }
```

Quiz 1. lcd.setCursor(3,1)은 무엇을 의미할까요?

Answer.

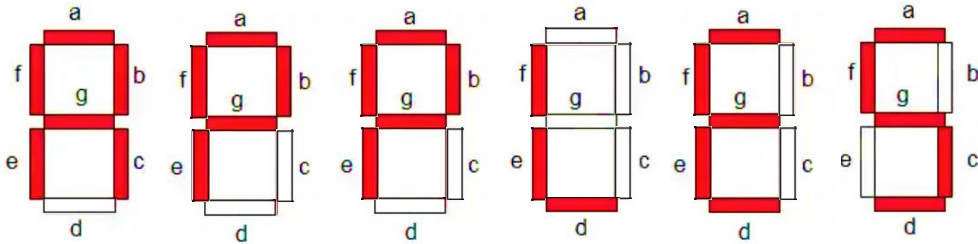
커서 또는 출력의 위치를
지정된 좌표로 이동시킨다.

즉, 3번째 열
1번째 행으로 이동시킨다.

정답: ① : 출력 ② : 좌표 ③ : 3 , ④ : 1

```
1  #include <Wire.h>
2  #include <LiquidCrystal_I2C.h>
3
4  LiquidCrystal_I2C lcd(0x27,16,2);
5
6  void setup()
7  {
8      lcd.init();
9      lcd.backlight();
10     lcd.setCursor(0,0);
11     lcd.print("Hello,world!");
12     lcd.setCursor(3,1);
13     lcd.print("Sweat!");
14 }
15
16 void loop()
17 {
18 }
```

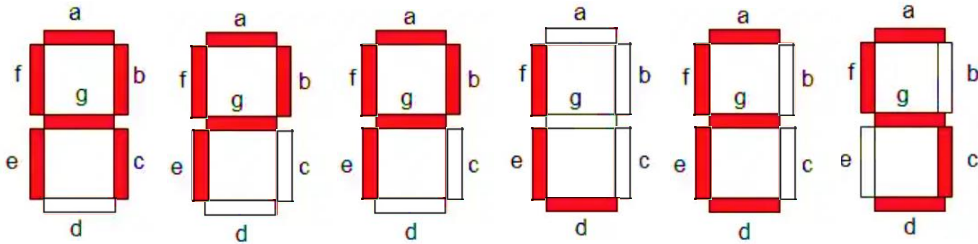
Quiz 2. 7-segment에 APPLES의 알파벳을 순서대로 출력하려 합니다. 주어진 빈칸에 들어갈 배열들을 맞춰볼까요? (단, 회로도는 실습의 회로와 같습니다.)



① :

```
int segmentLEDs[] = {2, 3, 4, 5, 6, 7, 8, 9};
int segmentLEDsNum = 8;
int digitForAlphabet[5][8] = {
// {A, B, C, D, E, F, G, DP}
//A
//E
//L
//P
//S
};
void setup() {
for (int i = 0 ; i < segmentLEDsNum ; i++) {
pinMode(segmentLEDs[i], OUTPUT);
}
}
// APPLES의 알파벳을 차례대로 출력합니다.
void loop() {
for (int j = 0 ; j < segmentLEDsNum ; j++) { //A
digitalWrite(segmentLEDs[segmentLEDsNum], digitForAlphabet[0][j]);
} delay(1000);
for (int j = 0 ; j < segmentLEDsNum ; j++) { //P(2초동안)
digitalWrite(segmentLEDs[segmentLEDsNum], digitForAlphabet[3][j]);
} delay(1000);
for (int j = 0 ; j < segmentLEDsNum ; j++) { //L
digitalWrite(segmentLEDs[segmentLEDsNum], digitForAlphabet[2][j]);
} delay(1000);
for (int j = 0 ; j < segmentLEDsNum ; j++) { //E
digitalWrite(segmentLEDs[segmentLEDsNum], digitForAlphabet[1][j]);
} delay(1000);
for (int j = 0 ; j < segmentLEDsNum ; j++) { //S
digitalWrite(segmentLEDs[segmentLEDsNum], digitForAlphabet[4][j]);
} delay(1000);
}
```

Quiz 2. 7-segment에 APPLES의 알파벳을 순서대로 출력하려 합니다. 주어진 빈칸에 들어갈 배열들은 맞혀볼까요? (단, 회로도에는 실습의 회로와 같습니다.)



정답 ① :

```
{1, 1, 1, 0, 1, 1, 1, 0}, //A
{1, 0, 0, 1, 1, 1, 1, 0}, //E
{0, 0, 0, 1, 1, 1, 0, 0}, //L
{1, 1, 0, 0, 1, 1, 1, 0}, //P
{1, 0, 1, 1, 0, 1, 1, 0}, //S
```

```
int segmentLEDs[] = {2, 3, 4, 5, 6, 7, 8, 9};
int segmentLEDsNum = 8;
int digitForAlphabet[5][8] = {
// {A, B, C, D, E, F, G, DP}
  {1, 1, 1, 0, 1, 1, 1, 0}, //A
  {1, 0, 0, 1, 1, 1, 1, 0}, //E
  {0, 0, 0, 1, 1, 1, 0, 0}, //L
  {1, 1, 0, 0, 1, 1, 1, 0}, //P
  {1, 0, 1, 1, 0, 1, 1, 0}, //S
};

void setup() {
  for (int i = 0 ; i < segmentLEDsNum ; i++) {
    pinMode(segmentLEDs[i], OUTPUT);
  }
}

// APPLES의 알파벳을 차례대로 출력합니다.
void loop() {
  for (int j = 0 ; j < segmentLEDsNum ; j++) {          //A
    digitalWrite(segmentLEDs[segmentLEDsNum], digitForAlphabet[0][j]);
  } delay(1000);
  for (int j = 0 ; j < segmentLEDsNum ; j++) {          //P(2초동안)
    digitalWrite(segmentLEDs[segmentLEDsNum], digitForAlphabet[3][j]);
  } delay(1000);
  for (int j = 0 ; j < segmentLEDsNum ; j++) {          //L
    digitalWrite(segmentLEDs[segmentLEDsNum], digitForAlphabet[2][j]);
  } delay(1000);
  for (int j = 0 ; j < segmentLEDsNum ; j++) {          //E
    digitalWrite(segmentLEDs[segmentLEDsNum], digitForAlphabet[1][j]);
  } delay(1000);
  for (int j = 0 ; j < segmentLEDsNum ; j++) {          //S
    digitalWrite(segmentLEDs[segmentLEDsNum], digitForAlphabet[4][j]);
  } delay(1000);
}
```

오늘 수업 Point!



오늘 배운 내용에서 질문 있나요?

