
I2C LCD

I2C LCD

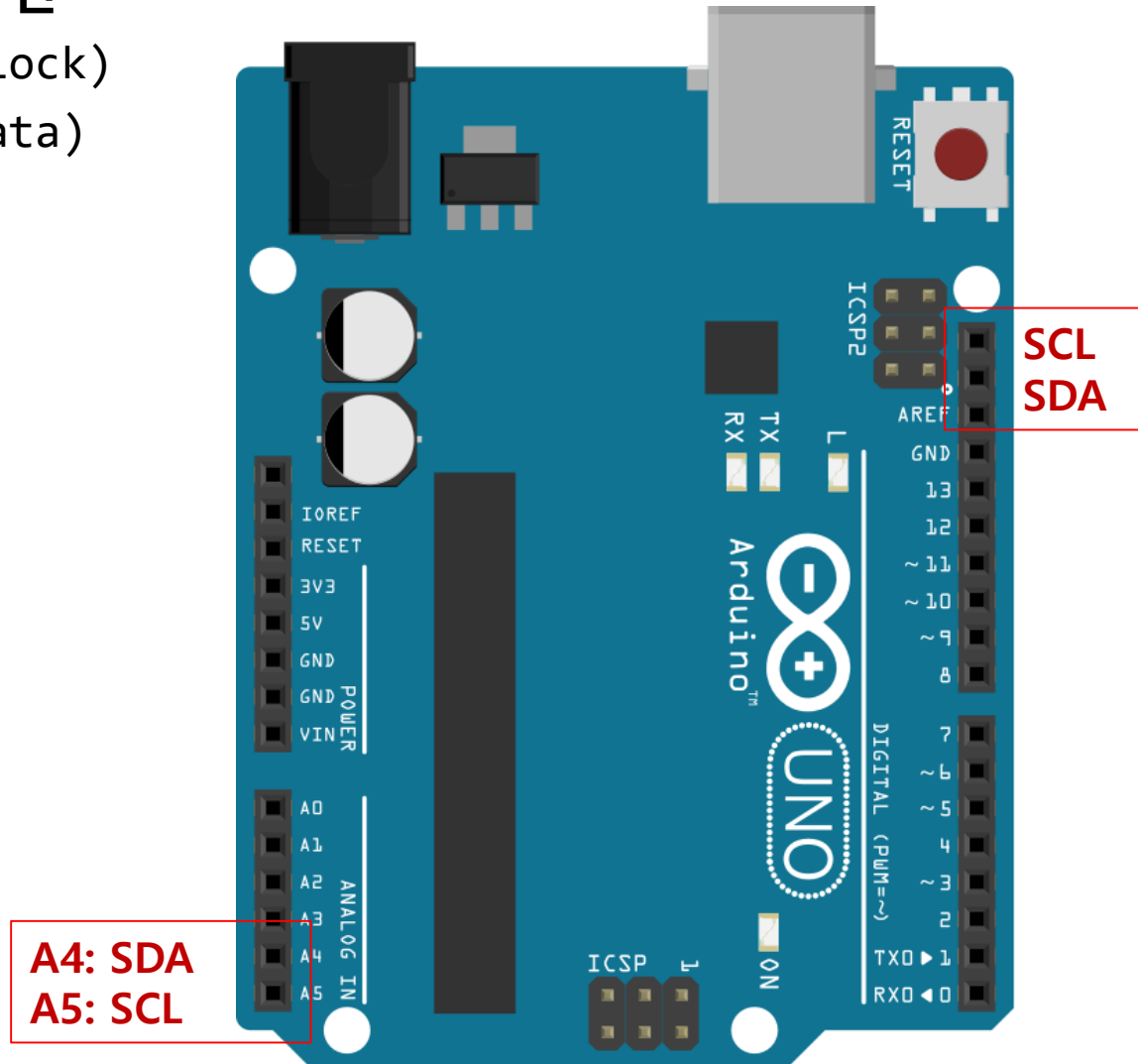
❖ I2C LCD



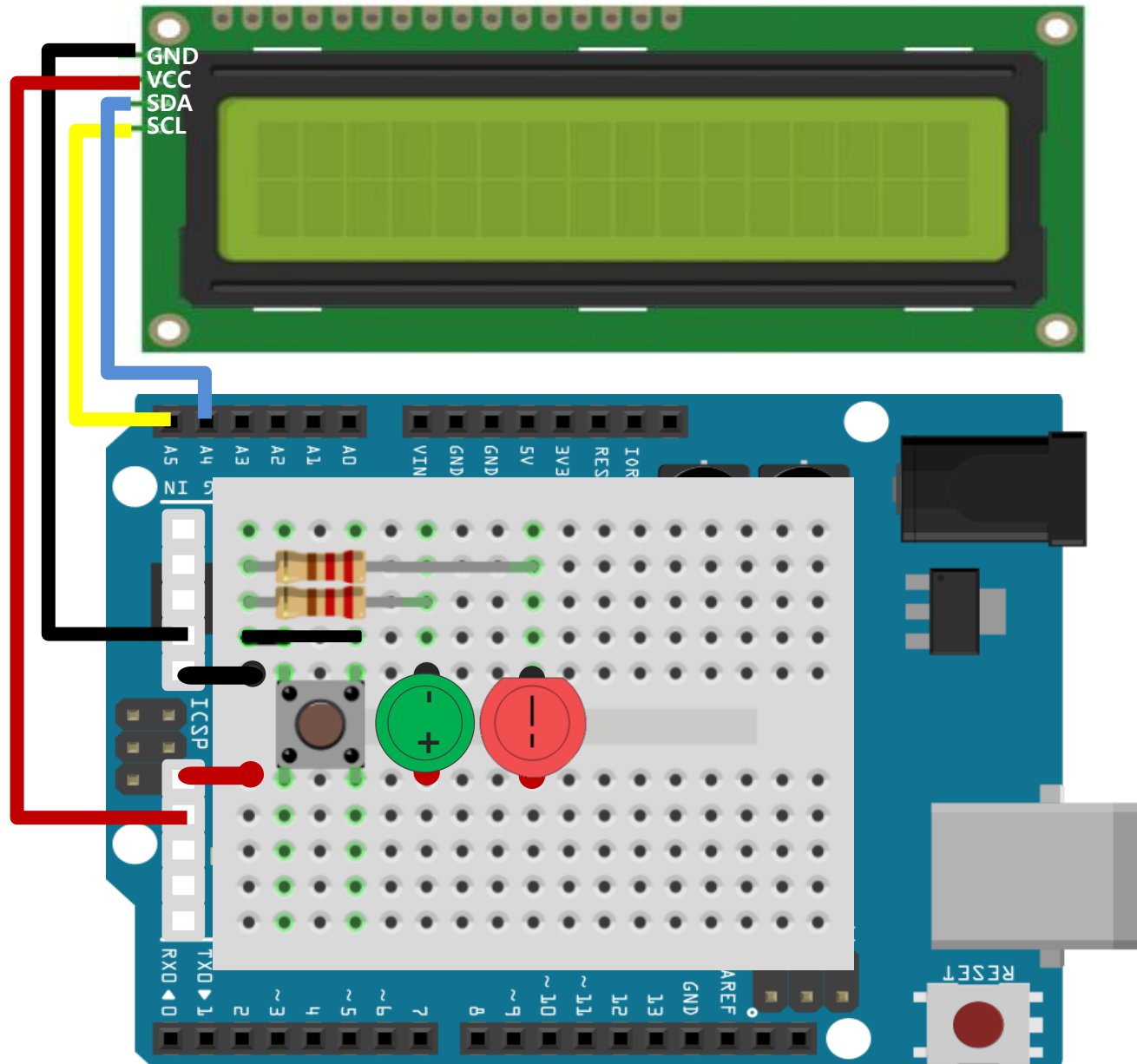
I2C LCD

❖ I2C 관련 핀

- SCL(Clock)
- SDA(Data)
- VCC
- GND



I2C LCD



GND
VCC
SDA -- A4
SCL -- A5

I2C LCD

❖ I2C LCD 디바이스 어드레스 설정

상위 어드레스 핀(A7-A3) : 고정					A2, A1, A0핀 위/아래 연결관계			Address
A7	A6	A5	A4	A3	A2	A1	A0	
0	0	1	0	0	1(연결x)	1(연결x)	1(연결x)	0x27
0	0	1	0	0	1(연결x)	1(연결x)	0(연결)	0x26
0	0	1	0	0	1(연결x)	0(연결)	1(연결x)	0x25
0	0	1	0	0	1(연결x)	0(연결)	0(연결)	0x24
0	0	1	0	0	0(연결)	1(연결x)	1(연결x)	0x23
0	0	1	0	0	0(연결)	1(연결x)	0(연결)	0x22
0	0	1	0	0	0(연결)	0(연결)	1(연결x)	0x21
0	0	1	0	0	0(연결)	0(연결)	0(연결)	0x20

실습1: I2C 주소 스캔 - I2C 디바이스 찾기

❖ F1 > Arduino Examples

- Examples for Arduino Uno > Wire > i2c_scanner
- 예제 코드 ex01/app.ino로 복사

실습1: I2C 주소 스캔 - I2C 디바이스 찾기

❖ ex01/app.ino(i2c_scanner)

```
#include <Wire.h>

void setup()
{
  Wire.begin();
  Serial.begin(9600);
  while (!Serial);           // Leonardo: wait for serial monitor
  Serial.println("\nI2C Scanner");
}

void loop()
{
  byte error, address;
  int nDevices;

  Serial.println("Scanning...");

  nDevices = 0;
```

실습1: I2C 주소 스캔 - I2C 디바이스 찾기

❖ ex01/app.ino(i2c_scanner)

```
for(address = 1; address < 127; address++ ){
  Wire.beginTransaction(address);
  error = Wire.endTransmission();

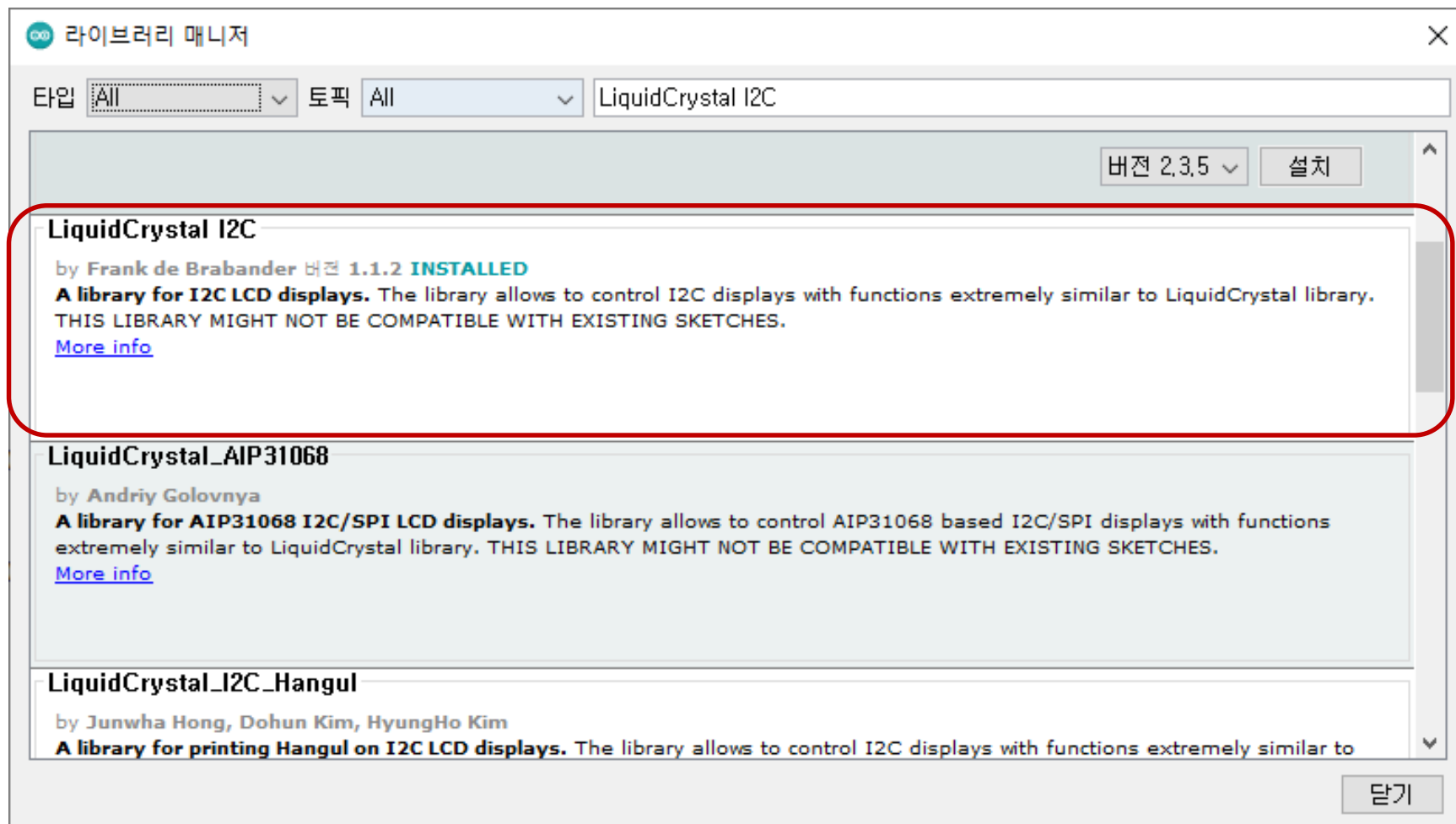
  if (error == 0){
    Serial.print("I2C device found at address 0x");
    if (address<16) Serial.print("0");
    Serial.print(address,HEX);
    Serial.println("  !");
    nDevices++;
  }
  else if (error==4){
    Serial.print("Unknown error at address 0x");
    if (address<16) Serial.print("0");
    Serial.println(address,HEX);
  }
}
if (nDevices == 0) Serial.println("No I2C devices found\n");
else Serial.println("done\n");

delay(5000);          // wait 5 seconds for next scan
}
```


I2C LCD

❖ 라이브러리 설치

- Library Manager > LiquidCrystal I2C 검색
 - Frank de Brabander 제작



❖ 객체 생성

```
#include <LiquidCrystal_I2C.h>
```

생성자:

```
LiquidCrystal_I2C(주소, 컬럼 수, 행 수);
```

```
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

❖ 주요 메서드

- `init();` // 초기화
- `setCursor(uint8_t, uint8_t);` // (컬럼번호, 행번호), 0부터 시작
- `print(...);` // 문자열 출력
- `clear();`
- `home();`
- `backlight();` `noBacklight();`
- `cursor();` `noCursor();`
- `display();` `noDisplay();`
- `blink();` `noBlink();`
- `printLeft();` `printRight();`
- `leftToRight();` `rightToLeft();`
- `shiftIncrement();` `shiftDecrement();`
- `autoscroll();` `noAutoscroll();`
- `scrollDisplayLeft();` `scrollDisplayRight();`
- `createChar(uint8_t, uint8_t[]);`
- `write(uint8_t);`

실습2: 간단한 문자 출력

❖ ex02/app.ino

```
#include <LiquidCrystal_I2C.h>

// set the LCD address to 0x27 for a 16 chars and 2 line display
LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {
    lcd.init(); // initialize the lcd
    // Print a message to the LCD.
    lcd.backlight();
    lcd.setCursor(3,0);
    lcd.print("Hello, world!");
    lcd.setCursor(2,1);
    lcd.print("Ywrobot Arduino!");
    lcd.setCursor(0,2);
    lcd.print("Arduino LCM IIC 2004");
    lcd.setCursor(2,3);
    lcd.print("Power By Ec-yuan!");
}

void loop() {
}
```

실습3: 문자열 스크롤

❖ ex03/app.ino

```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup()
{
    // put your setup code here, to run once:
    lcd.init();

    lcd.setCursor(0, 0);    // 커서 홈(0,0)
    lcd.print("1+++++++23-----45*****67#####8"); // 40문자 저장

    lcd.setCursor(0, 1);    // 커서 2행 1열(0, 1)
    lcd.print("1+++++++23-----45*****67#####8"); // 40문자 저장
}
```

실습3: 문자열 스크롤

❖ ex03/app.ino

```
void loop()
{
    // put your main code here, to run repeatedly:
    int    n;

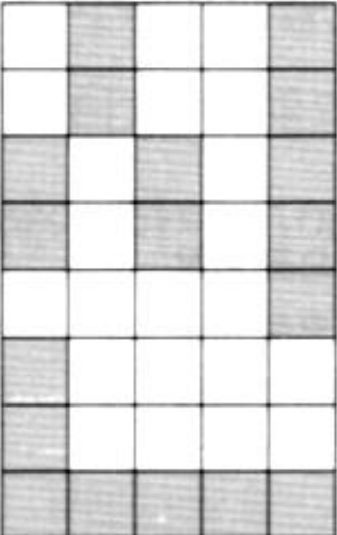
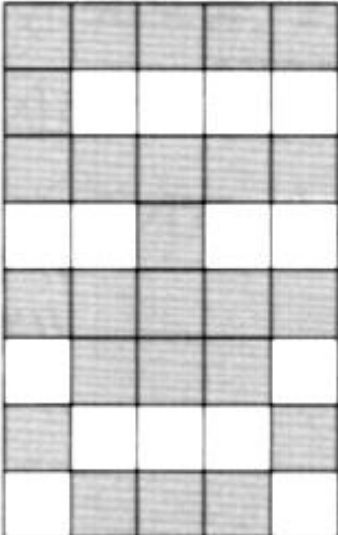
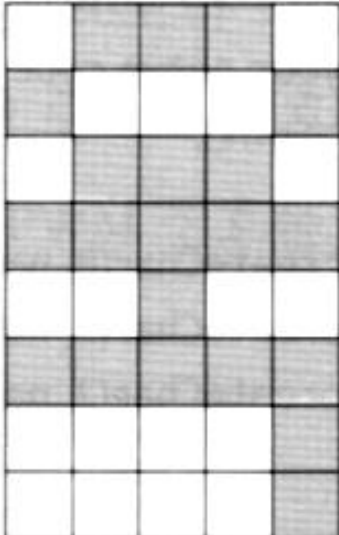
    for(n = 0;n < 80;n++){
        lcd.scrollDisplayLeft(); // 왼쪽으로 한 칸씩 이동
        delay(500);
    }
    delay(4000);                // 4초 대기

    for(n = 0;n < 80;n++){
        lcd.scrollDisplayRight(); // 오른쪽으로 한 칸씩 이동
        delay(500);
    }
    delay(4000);                // 4초 대기
}
```

실습3: 사용자 정의 패턴 출력

❖ 사용자 정의 문자 패턴

- 5x8 크기
- 8글자까지 저장 가능
- 문자 패턴 만들기
 - <https://maxpromer.github.io/LCD-Character-Creator/>

	0x09		0x1F		0x0E
	0x09		0x10		0x11
	0x15		0x1F		0x0E
	0x15		0x04		0x1F
	0x01		0x1F		0x04
	0x10		0x0E		0x1F
	0x10		0x11		0x01
	0x1F		0x0E		0x01

실습4: 사용자 정의 패턴 출력

❖ ex04/app.ino

```
#include <LiquidCrystal_I2C.h>
// '신' 패턴
uint8_t name_s[8] = {0x09, 0x09, 0x15, 0x15, 0x01, 0x10, 0x10, 0x1F};
// '동' 패턴
uint8_t name_d[8] = {0x1F, 0x10, 0x1F, 0x04, 0x1F, 0x0E, 0x11, 0x0E};
// '욱' 패턴
uint8_t name_u[8] = {0x0E, 0x11, 0x0E, 0x1F, 0x04, 0x1F, 0x01, 0x01};

LiquidCrystal_I2C lcd(0x27, 16, 2);    // lcd 객체 생성

void setup()
{
    // put your setup code here, to run once:
    lcd.init();

    lcd.createChar(0, name_s);          // '신' 패턴 코드 0으로 저장
    lcd.createChar(1, name_d);          // '동' 패턴 코드 1으로 저장
    lcd.createChar(2, name_u);          // '욱' 패턴 코드 2으로 저장
}
```

실습4: 사용자 정의 패턴 출력

❖ ex04/app.ino

```
void loop()
{
    // put your main code here, to run repeatedly:
    lcd.setCursor(0, 0);          // 커서 2행 1열(0, 1)
    lcd.print("Hello, Arduino!"); // 문자열 표시

    lcd.setCursor(0, 1);          // 커서 2행 1열(0, 1)
    lcd.print("My name is ");     // 문자열 표시

    lcd.write(0);
    lcd.write(1);
    lcd.write(2);
}
```