

C h a p t e r

08



LCD 모듈

Liquid Crystal Device Module

1

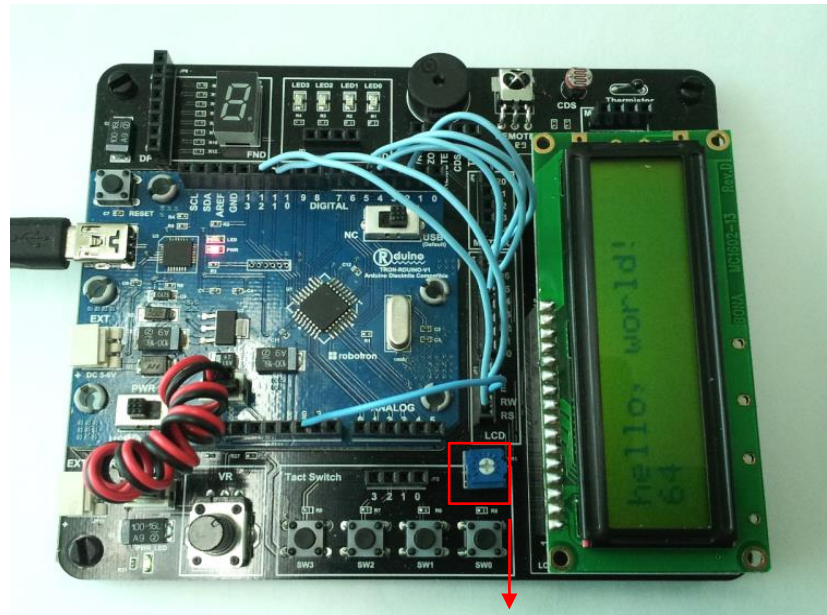
LCD 모듈

문자/숫자를 출력하기 위해 사용되는 LCD 모듈의 제어하는 방법에 대해서 학습합니다.

■ LCD 모듈

LCD 모듈은 글자(영어), 숫자 등을 출력하는데 많이 사용되는 장치입니다. 보통 LCD 모듈은 16칸 2줄의 문자 LCD 모듈 (“16x2 캐릭터 LCD” 라고 부름)을 사용하며, 아래 그림과 같이 베이스 보드의 오른쪽 LCD 소켓에 LCD 모듈을 끼워 사용합니다.

LCD 모듈 핀	기 능	설 명
1	Vss	GND
2	VDD	VCC
3	VL	LCD 구동 전압
4	RS	데이터/명령어 선택
5	RW	데이터 읽기/쓰기 결정
6	E	LCD 사용 (Enable)
7-14	DATA 0-7	데이터 라인



LCD 모듈에 글자가 보이지 않거나
약하게 보이면, 서서히 돌려 조절합니다.

보통, 문자 LCD 모듈은 16개의 핀으로 구성되어 있고, 15번과 16번 핀은 LCD 모듈의 백라이트 기능을 수행합니다.
각 핀의 기능은 표와 같습니다.

LCD 모듈에 문자열 출력

베이스 보드에 LCD 모듈을 연결하고, LCD 화면에 문자열을 출력하는 프로그램을 실습합니다.

하드웨어 연결

1. 공두이노 보드와 베이스 보드의 LCD 모듈 연결핀을 연결 선으로 아래와 같이 연결합니다.

공두이노 보드	연결방향	베이스 보드 LCD 모듈
2번 PIN	→	D7
3번 PIN	→	D6
4번 PIN	→	D5
5번 PIN	→	D4
11번 PIN	→	E
12번 PIN	→	RS
GND PIN	→	RW

- ✓ LCD 모듈을 사용하기 위해서는 프로그램의 맨 처음 부분에

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
```

를 사용합니다.

베이스 보드의 LCD 모듈에 “Hello, World!” 문자를 출력합니다.

```
#include <LiquidCrystal.h>

// LCD 연결 설정
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
    lcd.begin(16, 2);           // 16칸 2줄 LCD 초기화
    lcd.print("Hello, World!"); // 문자열 출력
}

void loop() {
}
```

■ LCD 관련 함수

1. LiquidCrystal() - LCD 제어 선의 연결 핀을 설정합니다.

LiquidCrystal(RS, E, D4, D5, D6, D7) 로 간단히 사용이 가능합니다.

2. begin() - LCD의 종류를 설정합니다. lcd.begin(cols, rows) 을 사용하여 연결된 LCD의 가로(칸)과 세로(줄)을 입력합니다.

3. clear() - LCD 화면을 모두 삭제하고, 커서의 위치를 첫 위치(0,0)로 이동합니다.

4. home() - 커서의 위치를 첫 위치(0,0)로 이동합니다.

5. setCursor() - 커서의 위치를 설정한 위치로 이동합니다.

lcd.setCursor(x, y) 로, 가로(x)와 세로(y) 위치를 설정합니다.

6. write() - LCD에 글자를 출력합니다. lcd.write(data) 를 사용합니다.

7. print() - LCD에 숫자나 문자열을 출력합니다.

8. cursor() - LCD 화면에 커서(_)를 보입니다.

9. noCursor() - LCD 화면에 커서를 감춥니다.

10. blink() - LCD 화면의 커서를 깜빡입니다.

11. noBlink() - LCD 화면의 커서를 깜빡이지 않습니다..

12. display() - LCD 화면의 내용을 나타나게 합니다.

13. noDisplay() - LCD 화면의 내용을 사라지게 합니다.

14. scrollDisplayLeft() - LCD 화면의 내용을 왼쪽으로 한 칸 이동합니다.

15. scrollDisplayRight() - LCD 화면의 내용을 오른쪽으로 한 칸 이동합니다.

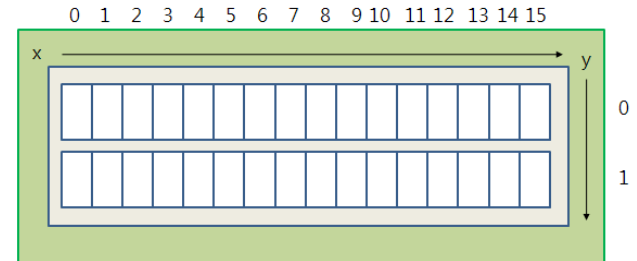
16. autoscroll() - LCD 화면이 자동으로 스크롤(이동)되도록 합니다.

17. noAutoscroll() - LCD 화면이 자동으로 스크롤(이동)되지 않도록 합니다.

18. leftToRight() - LCD 화면에 표시되는 글자가 왼쪽에서 오른쪽 방향으로 출력되도록 합니다.

19. rightToLeft() - LCD 화면에 표시되는 글자가 오른쪽에서 왼쪽 방향으로 출력되도록 합니다.

20. createChar() - LCD 모듈의 사용자 정의 문자를 생성합니다.



LCD 모듈에 함수를 이용하여 다양하게 출력하는 연습을 합니다.

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
    lcd.begin(16, 2);
}

void loop() {
    lcd.print("Cursor ON-Blink");
    lcd.cursor();
    lcd.blink();
    delay(3000);
    lcd.clear();
    lcd.print("Cursor OFF");
    lcd.noBlink();
    lcd.noCursor();
    delay(3000);
    lcd.clear();
    ...
}
```

커서 보이고 깜빡임

커서 감추고 깜박임 없음

```
for (int i=0; i<=50; i++){
    lcd.home();
    lcd.print("No : ");
    lcd.print(i);
    delay(300);
}

for (int i=0; i<3; i++){
    lcd.noDisplay();
    delay(500);
    lcd.display();
    delay(500);
}

...
}
```

숫자를 0 부터 50까지 출력

LCD 모듈 표시된 글자를 3번 깜빡임

```
for (int i=0; i<3; i++){  
    lcd.scrollDisplayRight();  
    delay(500);  
}
```



LCD 모듈에 표시된 글자를 3번 오른쪽으로 이동

```
for (int i=0; i<3; i++){  
    lcd.scrollDisplayLeft();  
    delay(500);  
}
```



LCD 모듈에 표시된 글자를 3번 왼쪽으로 이동

```
lcd.clear();  
}
```



- ✓ LCD 화면에 아날로그 소자인 가변저항(VR)과 광센서(CDS)의 값을 입력 받아 출력하는 프로그램을 작성합니다.

exam047

VR = 326
CDS = 220

실습 16

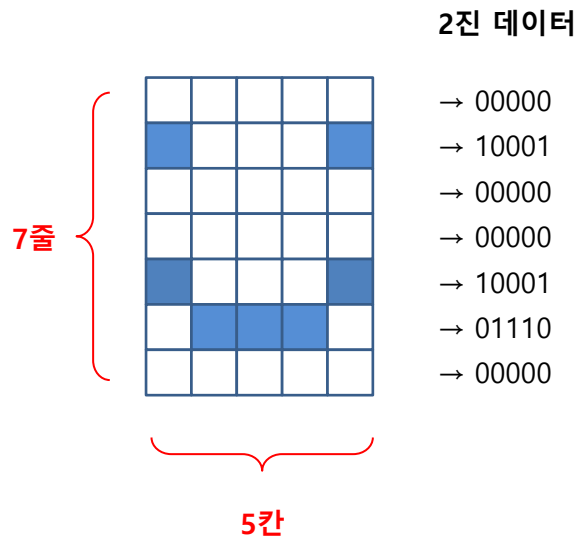
LCD 모듈에 사용자 정의 문자 출력

LCD 모듈에 임의로 정의한 문자를 만들어, LCD 화면에 출력하는 프로그램을 실습합니다.

하드웨어 연결

1. 공두이노 보드와 베이스 보드의 LCD 모듈 연결핀을 연결 선으로 아래와 같이 연결합니다.

공두이노 보드	연결방향	베이스 보드 LCD 모듈
2번 PIN	→	D7
3번 PIN	→	D6
4번 PIN	→	D5
5번 PIN	→	D4
11번 PIN	→	E
12번 PIN	→	RS
GND PIN	→	RW



베이스 보드의 LCD 모듈에 사용자 정의 문자 (앞 장에서 만든 웃는 얼굴) 를 출력합니다.

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

byte smiley[8] = {
    B00000,
    B10001,
    B00000,
    B00000,
    B10001,
    B01110,
    B00000,
};

void setup() {
    lcd.createChar(0, smiley);    // 0번 문자에 사용자 정의 문자
    lcd.begin(16, 2);
    lcd.write(byte(0));          // 0번 문자 출력
}

void loop() {
}
```

사용자 정의 문자 데이터

베이스 보드의 LCD 모듈에 사용자 정의 문자를 이용하여 아래 글씨를 출력합니다.

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

byte user0[8] = {B01111,B00001,B00001,B00001,B00001,B00001,B00001,}; // ㄱ
byte user1[8] = {B10000,B10000,B10000,B10000,B10000,B10000,B10000,}; // |
byte user2[8] = {B01111,B01000,B01000,B01111,B01000,B01000,B01111,}; // ≡
byte user3[8] = {B01001,B01001,B01001,B01111,B01001,B01001,B01001,}; // H
byte user4[8] = {B00110,B01111,B00110,B01001,B00110,B00000,B01111,}; // 호
byte user5[8] = {B00011,B00010,B00010,B00010,B00010,B00010,B00011,}; // [
byte user6[8] = {B11000,B01000,B01000,B01000,B01000,B01000,B11000,}; // ]

void setup() {
    lcd.createChar(0, user0); lcd.createChar(1, user1); lcd.createChar(2, user2); lcd.createChar(3, user3);
    lcd.createChar(4, user4); lcd.createChar(5, user5); lcd.createChar(6, user6);

    lcd.begin(16, 2);

    lcd.write(byte(0)); // ㄱ
    lcd.write(byte(1)); // |
    lcd.write(byte(2)); // ≡
    lcd.write(byte(3)); // H
    lcd.write(byte(4)); // 호
    lcd.write(byte(1)); // |
    lcd.setCursor(0,1); // 다음줄
    lcd.write(byte(5)); // [
    lcd.write(byte(6)); // ]
}

void loop() { }
```

사용자 문자 정의

사용자 문자 출력

