

Chapter 12. 텍스트 LCD

액정(liquid crystal)은 액체이면서 고체의 성질을 띠는 물질로 1800년대 처음 발견되어 1960년대 후반에 이르러 디스플레이로 처음 만들어졌다. 텍스트 LCD는 고정된 위치에 문자 단위로 영문자와 숫자 등을 표시할 수 있어 간단한 메시지 출력에 흔히 사용된다. 텍스트 LCD 사용을 위해 아두이노에는 기본 라이브러리 중 하나로 LiquidCrystal 라이브러리가 포함되어 있다. 텍스트 LCD는 표시할 수 있는 문자의 수에 따라 여러 종류가 존재하지만 16글자씩 2줄, 총 32 글자를 표시할 수 있는 텍스트 LCD를 흔히 볼 수 있으며 학습보드에도 동일한 텍스트 LCD가 포함되어 있다.

1. 텍스트 LCD

텍스트 LCD는 16개의 연결 핀을 가지고 있으며 이 중 3개는 제어 신호 핀, 8개는 데이터 신호 핀이다. 나머지 5개는 전원 및 백라이트 제어를 위해 사용되는 핀이다.

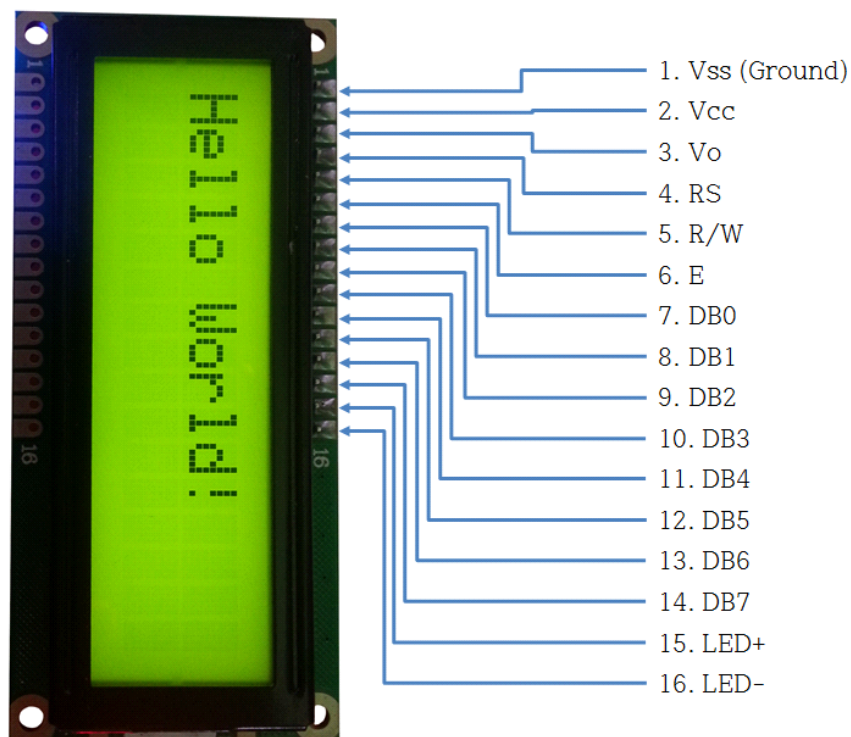


그림 1. 텍스트 LCD 핀 맵

핀 번호	이름	설명
1	V _{SS}	그라운드 (GND)
2	V _{DD}	5V 동작 전원
3	V _O	LED 전원으로 가변저항을 통해 0 ~ 5V 사이 입력
4	RS	레지스터 선택 (Register Select)
5	R/W	읽기 쓰기 (Read/Write)
6	E	활성화 (Enable)
7~14	DB0~DB7	데이터 신호 핀
15	LED+	백라이트 전원
16	LED0	

표 1. 텍스트 LCD의 핀

3개의 제어 신호는 표 2와 같다.

제어 신호	설명
RS (Register Select)	텍스트 LCD를 제어하기 위해서는 제어 레지스터와 데이터 레지스터의 두 개 레지스터가 사용되며, RS 신호는 명령을 담고 있는 레지스터(RS = LOW)와 데이터를 담고 있는 레지스터(RS = HIGH) 중 하나를 선택하기 위해 사용한다.
R/W (Read/Write)	읽기 (R/W = HIGH) 및 쓰기 (R/W = LOW) 모드 선택을 위해 사용한다. 일반적으로 LCD는 데이터를 쓰기위한 용도로만 사용하므로 일반적으로 R/W 신호는 GND에 연결한다.
E (Enable)	하강 에지(falling edge)에서 LCD 드라이버가 처리를 시작하도록 지시하기 위한 신호로 사용된다.

표 2. 텍스트 LCD 제어 신호

DB0에서 DB7까지의 8개 핀은 데이터 핀으로 8비트 모드로 동작하는 경우에는 8개가 모두 사용되지만 4비트 모드로 동작하는 경우에는 4개만 사용된다. 마이크로컨트롤러와 사용하는 경우에는 연결 핀의 수를 줄이기 위해 4비트 모드를 사용하는 경우가 일반적이며 학습보드에서도 4비트 모드를 사용한다. 텍스트 LCD의 경우 일반적으로 데이터를 읽는 용도로는 사용하지 않으므로 R/W는 GND에 연결하여 사용한다.

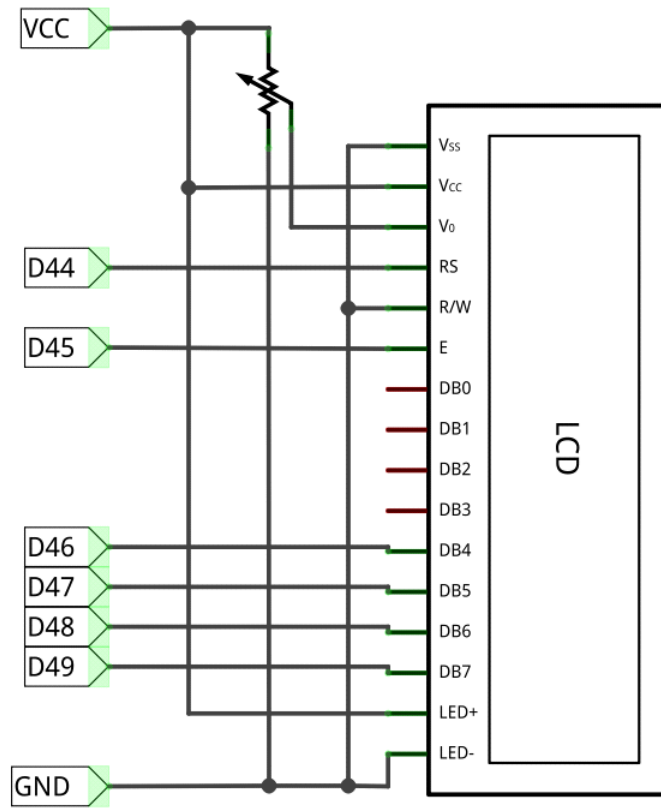


그림 2. 텍스트 LCD 연결 회로도

학습보드에는 텍스트 LCD와 콘트라스트 조절을 위한 가변저항이 포함되어 있다.



그림 3. 학습보드의 텍스트 LCD와 콘트라스트 조절용 가변저항

2. LiquidCrystal 라이브러리

아두이노의 LiquidCrystal 라이브러리는 4비트 모드와 8비트 모드를 지원하지만 이 장에서는 4비트 모드로 사용한다. 먼저 ‘스케치 → Include Library → LiquidCrystal’ 메뉴를 선택하여 LiquidCrystal.h 파일을 포함시키고, LiquidCrystal 클래스의 객체를 생성한다. 객체를 생성할 때에는 텍스트 LCD와 연결된 핀 번호를 지정해 주어야 한다.

```
// 핀 번호 (RS, E, DB4, DB5, DB6, DB7)
LiquidCrystal lcd(44, 45, 46, 47, 48, 49);
```

begin 함수로 텍스트 LCD 객체를 초기화 한다. 초기화 함수에서는 행과 열의 개수를 지정한다. 폰트 크기도 지정할 수 있지만 2행인 경우에는 5x8 크기의 폰트만 사용할 수 있다.

```
void begin(uint8_t cols, uint8_t rows, uint8_t charsize = LCD_5x8DOTS);
```

- 매개변수
 - cols : 열의 개수
 - rows : 행의 개수
 - charsize : 글자 크기
- 반환값 : 없음

텍스트 LCD에 문자를 출력하기 위해서는 바이트 단위의 출력함수인 `write`나 문자열 단위의 출력함수인 `print`를 사용할 수 있다. 특히 `print` 함수는 문자열, 숫자 등을 출력하기 위해 `Serial` 클래스의 `print` 함수와 동일한 방법으로 사용할 수 있다.

```
size_t write(uint8_t ch)
```

- 매개변수
 - ch : 출력할 문자
- 반환값 : 출력한 바이트 수

```
size_t print(value, format)
```

- 매개변수
 - value : 출력값 (char, char 배열, String, 정수, 실수 등)
 - format : 출력 형식
- 반환값 : 출력된 바이트 수

화면을 지우기 위해서는 `clear` 함수를, 커서를 임의의 위치로 옮기기 위해서는 `setCursor` 함수를 사용할 수 있다.

```
void clear(void)
```

- 매개변수 : 없음
- 반환값 : 없음

```
void setCursor(uint8_t col, uint8_t row)
```

- 매개변수

col : 열 위치 (0 ~ 15)

row : 행 위치 (0 ~ 1)

- 반환값 : 없음

이외에도 LiquidCrystal 클래스에는 유용한 멤버 함수들이 정의되어 있으므로 자세한 내용은 클래스가 정의된 헤더 파일을 참고하면 된다. 스케치 1은 LiquidCrystal 라이브러리를 사용하여 텍스트 LCD에 'Hello World!'를 표시하는 스케치의 예이다.

스케치 1. Hello World!

```
#include <LiquidCrystal.h>
```

```
// 핀 번호 (RS, E, DB4, DB5, DB6, DB7)
```

```
LiquidCrystal lcd(44, 45, 46, 47, 48, 49); // LCD 연결
```

```
void setup(){
```

```
  lcd.begin(16, 2); // LCD 초기화
```

```
  lcd.print("Hello World!"); // 문자열 출력
```

```
}
```

```
void loop(){
```

```
}
```

텍스트 LCD에 문자열이 표시되지 않는다면 가변저항으로 콘트라스트를 조절하면 된다.



그림 4. 스케치 1 실행 결과

setCursor 함수를 사용하여 임의의 위치에 문자를 표시해 보자. 텍스트 LCD에서 문자는 현재 커서의 위치에 표시되며 커서의 위치는 좌상단이 (0, 0)으로 기준점이 된다. 스케치 2는 setCursor 함수를 사용하여 숫자를 출력하는 스케치의 예이다.

스케치 2. 임의의 위치에 문자 나타내기

```
#include <LiquidCrystal.h>

// 핀 번호 (RS, E, DB4, DB5, DB6, DB7)
LiquidCrystal lcd(44, 45, 46, 47, 48, 49); // LCD 연결

void setup(){
    lcd.begin(16, 2);                                // LCD 초기화

    lcd.setCursor(0, 0);                             // 0행 0열로 이동
    lcd.write('1');                                   // 문자 단위 출력
    lcd.setCursor(5, 0);
    lcd.write('2');
    lcd.setCursor(0, 1);
    lcd.write('3');
    lcd.setCursor(5, 1);
    lcd.write('3');
}

void loop(){
}
```

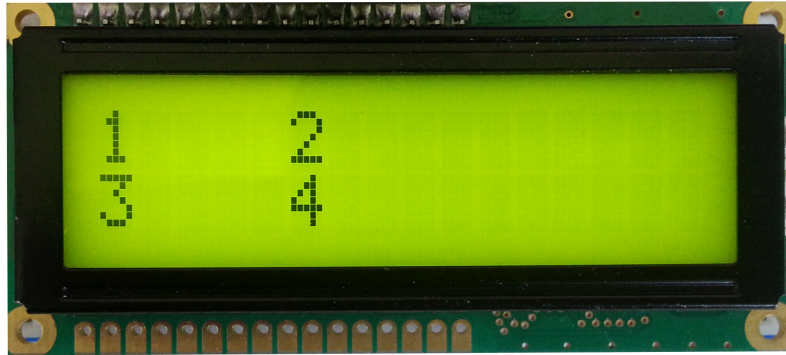


그림 6. 스케치 2 실행 결과