
Chapter 13. 그래픽 LCD

텍스트 LCD가 고정된 위치에 문자 단위로만 출력이 가능한 반면 그래픽 LCD(GLCD)는 픽셀 단위로 제어가 가능하므로 문자는 물론 도형, 이미지 등도 출력할 수 있다. GLCD 모듈은 크기와 해상도가 다양할 뿐만이 아니라 핀 배열이 표준화되어 있지 않아 GLCD에 따라 연결 방법이 서로 다르므로 반드시 데이터시트를 참고하여야 한다. 학습보드에 포함된 그래픽 LCD는 128×64 해상도를 가지며 KS0108 LCD 드라이버 칩을 사용한 그래픽 LCD이다.

1. glcd 라이브러리

텍스트 LCD와 달리 그래픽 LCD 제어를 위한 라이브러리는 아두이노의 기본 라이브러리로 포함되어 있지 않다. 따라서 GLCD 제어를 위해서는 먼저 openGLCD 라이브러리를 다운 받아¹⁾ 설치하여야 한다. 라이브러리가 설치되었으면 GLCD를 연결해 보자. GLCD는 20개의 핀을 가지고 있다. 20개의 핀 중 13개는 데이터와 제어 신호를 위해 사용되는 핀이며, 6개는 전원 관련 핀, 그리고 나머지 하나는 리셋 핀이다. openGLCD 라이브러리가 설치된 디렉터리 아래 'config\ks0108' 디렉터리에 'PinConfig_ks0108-Mega.h' 파일에 아두이노 메가2560과 GLCD를 연결하기 위한 디폴트 핀들이 정의되어 있다.

1) <https://bitbucket.org/bperrybap/openglcd/wiki/Home>

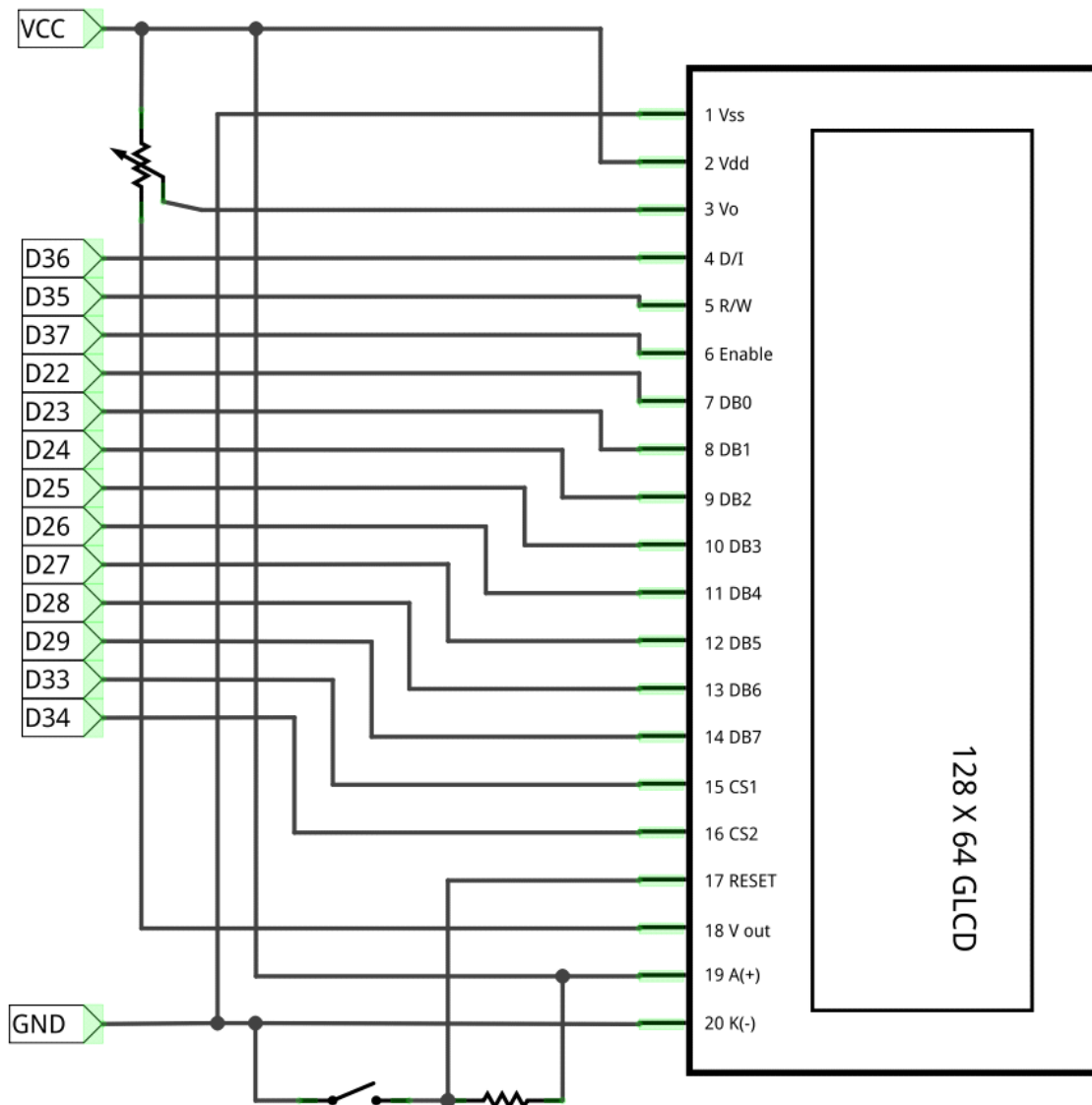


그림 1. GLCD 연결 회로도

GLCD는 8비트 데이터를 전송하므로 DB0부터 DB7까지 데이터 전송을 위한 핀이 정의되어 있고, CS1(Chip Select)과 CS2는 GLCD 모듈에서 화면을 둘로 나누어 제어하기 때문에 각 영역을 선택하기 위해 사용된다. R/W (Read/Write), D/I (Data/Instruction), Enable은 제어 신호에 해당한다. 나머지 6개 중 2쌍은 GLCD와 백라이트를 위한 전원에 해당한다. 나머지 2개는 콘트라스트 조절을 위한 용도로 사용된다. GLCD와 메가2560 연결을 정리하면 표 1과 같다.

GLCD 핀	아두이노 핀	기능	GLCD 핀	아두이노 핀	기능
1	GND		11	26	Data 4
2	VCC		12	27	Data 5
3	-	Contrast In	13	28	Data 6
4	36	D/I (Data/Instruction)	14	29	Data 7
5	35	R/W (Read Write)	15	33	Chip 1 Select
6	37	Enable	16	34	Chip 2 Select
7	22	Data 0	17	-	Reset
8	23	Data 1	18	-	Contrast Out
9	24	Data 2	19	VCC	Backlight +
10	25	Data 3	20	GND	Backlight -

표 1. GLCD와 아두이노 메가2560 연결

학습보드에는 그래픽 LCD와 콘트라스트 조절을 위한 가변저항이 포함되어 있다.

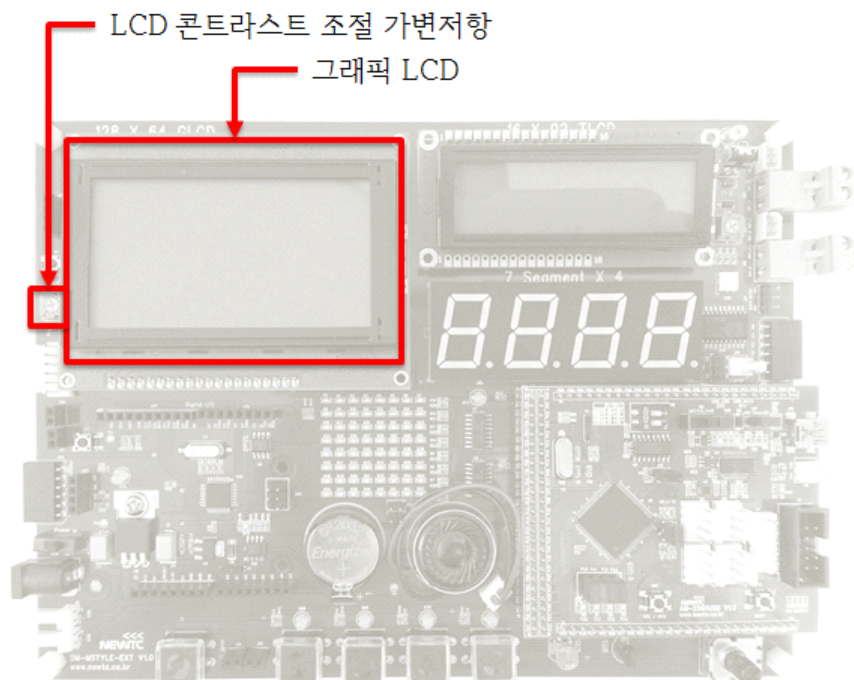


그림 2. 학습보드의 그래픽 LCD와 콘트라스트 조절용 가변저항

GLCD에 'hello world!'를 출력해보자. '스케치 → Include Library → openGLCD' 메뉴 항목을 선택하면 필요한 헤더 파일들이 자동으로 포함된다.

```
#include <openGLCD.h>
#include <openGLCD_Buildinfo.h>
#include <openGLCD_Config.h>
```

GLCD 제어를 위한 클래스의 이름은 glcd이며 glcd의 객체인 'GLCD'를 통해 GLCD 장치를 제어할 수 있다. GLCD 객체는 라이브러리에 정의되어 있으므로 별도의 객체 생성은 필요하지 않다. GLCD의 연결이 번거롭고 귀찮은 작업인 반면 연결만 제대로 되었다면 사용 방법은 간단하다. 초기화 함수 init과 출력 함수 print 만으로 'hello, world!'를 출력할 수 있다.

스케치 1. GLCD 테스트

```
#include <openGLCD.h>
#include <openGLCD_Buildinfo.h>
#include <openGLCD_Config.h>

void setup()
{
    GLCD.Init();                // GLCD 초기화

    GLCD.SelectFont(System5x7); // 출력 폰트 선택

    GLCD.print("hello, world!"); // 문자열 출력
}

void loop()
{
}
```

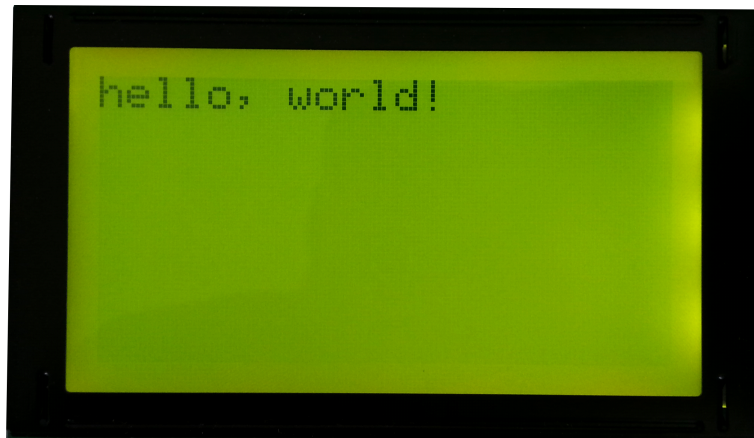


그림 3. 스케치 1 출력 결과

GLCD에서 원하는 위치에 문자를 출력하기 위해서는 커서 함수를 사용할 수 있다. GLCD는 픽셀 별 위치 지정이 가능하지만 문자열 출력은 지정한 폰트에 따라 문자 단위로 위치를 지정할 수 있다. CursorTo 함수를 사용하면 행과 열을 지정하여 문자를 출력할 수 있으며, CursorToXY 함수를 사용하면 픽셀 단위로 위치를 지정하여 문자를 출력할 수 있다. 두 함수 모두에서 좌상단 점이 (0, 0)으로 기준이 된다. 스케치 2는 CursorTo 함수와 CursorToXY 함수를 사용하여 임의의 위치를 지정하여 문자를 출력하는 스케치의 예이다.

스케치 2. 위치를 지정한 문자 출력

```
#include <openGLCD.h>
#include <openGLCD_Buildinfo.h>
#include <openGLCD_Config.h>

void setup(){
    GLCD.Init();                // GLCD 초기화

    GLCD.SelectFont(System5x7); // 문자 출력을 위한 폰트 선택

    for(int i = 0; i < 8; i++){
        GLCD.CursorTo(i, i);
        GLCD.print(i);
    }
}
```

```
for(int i = 0; i < 6; i++){  
    GLCD.CursorToXY(100, i * 10);  
    GLCD.print((char)('A' + i));  
}  
}  
  
void loop() {  
}
```

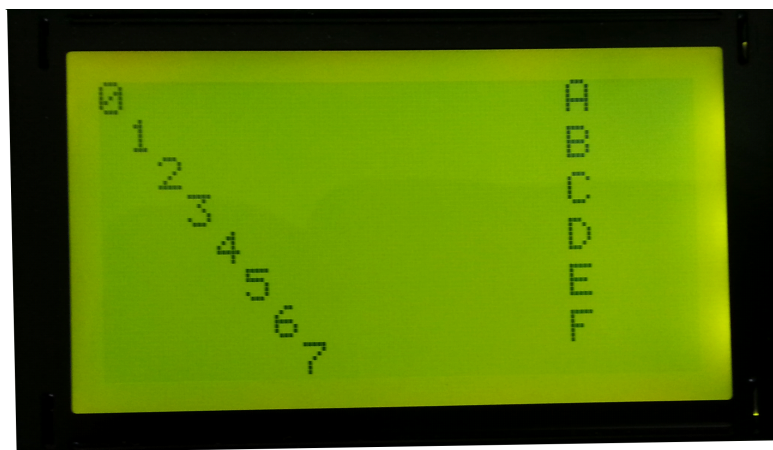


그림 4. 스케치 2 출력 결과

사용한 폰트가 5×7 크기의 폰트이므로 여백을 포함하여 한 글자는 6×8 도트를 필요로 한다. 행과 열의 개수는 사용하는 폰트 크기에 따라 자동으로 조절된다.

2. 도형 그리기

openGLCD 라이브러리에는 문자뿐만이 아니라 다양한 도형을 출력할 수 있는 함수들이 준비되어 있다. 자세한 내용은 openGLCD 라이브러리와 함께 배포되는 문서를 참고하면 된다. 스케치 3은 GLCD에 도형을 그리는 스케치의 예이다. 이 장에서 사용하는 GLCD는 단색 표현만 가능하므로 도형을 그리는 함수의 매개변수 중 하나인 color는 사용하지 않는다.

스케치 3. 도형 그리기

```
#include <openGLCD.h>
#include <openGLCD_Buildinfo.h>
#include <openGLCD_Config.h>

void setup(){
    GLCD.Init();                // GLCD 초기화

    // 직선 (x1, y1, x2, y2)
    GLCD.DrawLine(0, 0, 127, 63);
    GLCD.DrawLine(127, 0, 0, 63);

    // 사각형 (x, y, width, height)
    GLCD.DrawRect(0, 0, 128, 64);

    // 모서리 둥근 사각형 (x, y, width, height, radius)
    GLCD.DrawRoundRect(10, 10, 108, 44, 10);

    // 원 (center_x, center_y, radius)
    GLCD.DrawCircle(64, 32, 20);

    // 타원 (center_x, center_y, radius_x, radius_y)
    GLCD.DrawEllipse(64, 32, 50, 10);
}

void loop() {
}
```

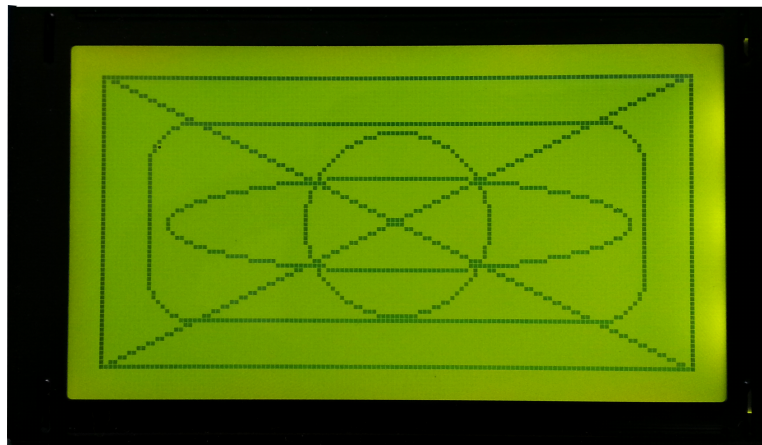


그림 6. 스케치 3 실행 결과