
임베디드 시스템

LCD 모듈 제어

In-Hyeok Kang

M23522@hallym.ac.kr

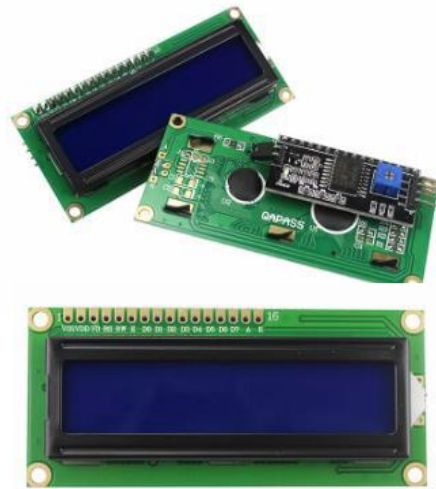
연구실: 공학관 1321호

Contents

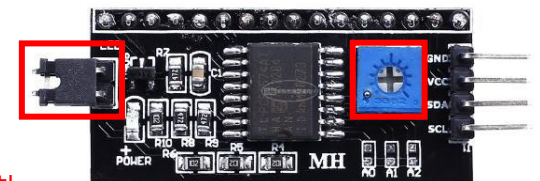
1. LCD 모듈
2. Raspberry Pi I2C 인터페이스 활성화
3. LCD 모듈 연결
4. LCD 모듈 제어

LCD 모듈

- LCD(Liquid Crystal Display) 모듈
 - I2C(Inter-Integrated Circuit) 인터페이스 16×2 LCD 모듈
 - 온보드 대비 제어 조정, 백 라이트 및 I2C 통신 인터페이스가 있으며 2-line 16-character 지원
 - LCD piggy-back board를 사용하여 I2C 버스를 통해 원하는 데이터를 LCD에 표시할 수 있음



1602 LCD 모듈



가변저항

- 문자 선명도 조절
- 시계방향으로 돌릴수록 밝아짐

점퍼 스위치

- 접지: Backlight On
- 비접지: Backlight Off



I2C I/O Expander(PCF8574T)

LCD 모듈

- LCD 모듈 – 1602 LCD Datasheet 참고
 - Compatible with Arduino Board or other controller board with I2C bus.
 - Display Type: Negative white on Blue backlight.
 - I2C Address: ~~0x38-0x3F (0x3F default)~~ (0x27 default)
 - Supply voltage: 5V
 - Interface: I2C to 4bits LCD data and control lines.
 - Contrast Adjustment: built-in Potentiometer.
 - Backlight Control: Firmware or jumper wire.
 - Board Size: 80x36 mm.

LCD 모듈

- LCD 모듈 – 1602 LCD Datasheet 참고

Hitachi's HD44780 based character LCD are very cheap and widely available, and is an essential part for any project that displays information. Using the LCD piggy-back board, desired data can be displayed on the LCD through the I2C bus. In principle, such backpacks are built around PCF8574 (from NXP) which is a general purpose bidirectional 8 bit I/O port expander that uses the I2C protocol. The PCF8574 is a silicon CMOS circuit provides general purpose remote I/O expansion (an 8-bit quasi-bidirectional) for most microcontroller families via the two-line bidirectional bus (I2C-bus). Note that most piggy-back modules are centered around PCF8574T (SO16 package of PCF8574 in DIP16 package) with a default slave address of 0x27. If your piggy-back board holds a PCF8574AT chip, then the default slave address will change to 0x3F. In short, if the piggy-back board is based on PCF8574T and the address connections (A0-A1-A2) are not bridged with solder it will have the slave address 0x27.

- I2C를 사용하는 범용 양방향 8-bit I/O port expander인 PCF8574(NXP)를 기반으로 구축됨
- PCF8574는 실리콘 CMOS 회로로, 2-line bidirectional bus(I2C bus)를 통해 대부분의 microcontroller 제품군에 범용 원격 I/O 확장(8-bit quasi-bidirectional)을 제공
- **Piggy-back board가 PCF8574T를 기반으로 하고, address connections(A0-A1-A2)가 납땜으로 연결되어 있지 않을 경우, slave address는 0x27임**



LCD 모듈

- LCD 모듈 – 1602 LCD Datasheet 참고

11.5 CGROM (Character Generator ROM)

CGROM has a 5 x 8 dots 204 characters pattern and a 5 x 10 dots 32 characters pattern. CGROM has 204 character patterns of 5 x 8 dots.

11.6 CGRAM (Character Generator RAM)

CGRAM has up to 5 . 8 dot, 8 characters. By writing font data to CGRAM, user defined characters can be used.

- 2가지 종류의 character 생성 방식
 - ✓ CGROM: 기존에 정의된 character pattern 생성
 - ✓ CGRAM: User defined characters 사용 가능

| Character Code (DDRAM Data) | | | | | | | | | CGRAM Address | | | | | | Character Patterns (CGRAM Data) | | | | | | | | |
|--------------------------------|----|----|----|----|----|----|----|----|------------------|----|----|----|----|----|------------------------------------|----|----|----|----|----|----|----|---|
| b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 | b5 | b4 | b3 | b2 | b1 | b0 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 | |
| 0 | 0 | 0 | 0 | 0 | - | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | - | - | - | 1 | 1 | 1 | 1 | 1 | |
| | | | | | | 0 | 0 | 0 | | | | 0 | 0 | 1 | | | | 0 | 0 | 0 | | | |
| | | | | | | 0 | 0 | 0 | | | | 0 | 1 | 0 | | | | 0 | 0 | 1 | 0 | 0 | |
| | | | | | | 0 | 0 | 0 | | | | 0 | 0 | 1 | | | | 1 | 0 | 0 | 1 | 0 | 0 |
| | | | | | | 0 | 0 | 0 | | | | 0 | 1 | 0 | | | | 0 | 0 | 1 | 0 | 0 | 0 |
| | | | | | | 0 | 0 | 0 | | | | 0 | 1 | 0 | | | | 1 | 0 | 0 | 1 | 0 | 0 |
| | | | | | | 0 | 0 | 0 | | | | 0 | 1 | 1 | | | | 0 | 0 | 1 | 0 | 0 | 0 |
| | | | | | | 0 | 0 | 0 | | | | 0 | 1 | 1 | | | | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | - | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | - | - | - | 1 | 1 | 1 | 1 | 0 | |
| | | | | | | 0 | 0 | 1 | | | | 0 | 0 | 1 | | | | 1 | 0 | 0 | 0 | 1 | |
| | | | | | | 0 | 0 | 1 | | | | 0 | 1 | 0 | | | | 1 | 0 | 0 | 0 | 1 | |
| | | | | | | 0 | 0 | 1 | | | | 0 | 1 | 1 | | | | 1 | 1 | 1 | 1 | 0 | |
| | | | | | | 0 | 0 | 1 | | | | 1 | 0 | 0 | | | | 1 | 0 | 1 | 0 | 0 | |
| | | | | | | 0 | 0 | 1 | | | | 1 | 0 | 1 | | | | 1 | 0 | 1 | 0 | 0 | |
| | | | | | | 0 | 0 | 1 | | | | 1 | 1 | 0 | | | | 1 | 0 | 0 | 0 | 1 | |
| | | | | | | 0 | 0 | 1 | | | | 1 | 1 | 1 | | | | 0 | 0 | 0 | 0 | 0 | |

LCD 모듈

- LCD 모듈 – 1602 LCD Datasheet 참고
 - LCD 모듈 제어를 위한 명령어 테이블
 - 각 명령어는 8-bit로 표현되며,
LCD 모듈 제어 라이브러리에 함수로 제공됨

11.9 Instruction Table

| Instruction | Instruction code | | | | | | | | | | Description | Execution time (fosc= 270 KHZ) |
|----------------------------|------------------|-----|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|--|--------------------------------|
| | RS | R/W | DB ₇ | DB ₆ | DB ₅ | DB ₄ | DB ₃ | DB ₂ | DB ₁ | DB ₀ | | |
| Clear Display | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | Write "20H" to DDRA and set DDRAM address to "00H" from AC | 1.53ms |
| Return Home | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | - | Set DDRAM address to "00H" From AC and return cursor to its original position if shifted. The contents of DDRAM are not changed. | 1.53ms |
| Entry mode Set | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | I/D | SH | Assign cursor moving direction And blinking of entire display | 39us |
| Display ON/OFF control | 0 | 0 | 0 | 0 | 0 | 0 | 1 | D | C | B | Set display (D), cursor (C), and Blinking of cursor (B) on/off Control bit. | |
| Cursor or Display shift | 0 | 0 | 0 | 0 | 0 | 1 | S/C | R/L | - | - | Set cursor moving and display Shift control bit, and the Direction, without changing of DDRAM data. | 39us |
| Function set | 0 | 0 | 0 | 0 | 1 | DL | N | F | - | - | Set interface data length (DL: 8-Bit/4-bit), numbers of display Line (N: =2-line/1-line) and, Display font type (F: 5x11/5x8) | 39us |
| Set CGRAM Address | 0 | 0 | 0 | 1 | AC ₅ | AC ₄ | AC ₃ | AC ₂ | AC ₁ | AC ₀ | Set CGRAM address in address Counter. | 39us |
| Set DDRAM Address | 0 | 0 | 1 | AC ₆ | AC ₅ | AC ₄ | AC ₃ | AC ₂ | AC ₁ | AC ₀ | Set DDRAM address in address Counter. | 39us |
| Read busy Flag and Address | 0 | 1 | BF | AC ₆ | AC ₅ | AC ₄ | AC ₃ | AC ₂ | AC ₁ | AC ₀ | Whether during internal Operation or not can be known By reading BF. The contents of Address counter can also be read. | 0us |
| Write data to Address | 1 | 0 | D ₇ | D ₆ | D ₅ | D ₄ | D ₃ | D ₂ | D ₁ | D ₀ | Write data into internal RAM (DDRAM/CGRAM). | 43us |
| Read data From RAM | 1 | 1 | D ₇ | D ₆ | D ₅ | D ₄ | D ₃ | D ₂ | D ₁ | D ₀ | Read data from internal RAM (DDRAM/CGRAM). | 43us |

INTRODUCTION

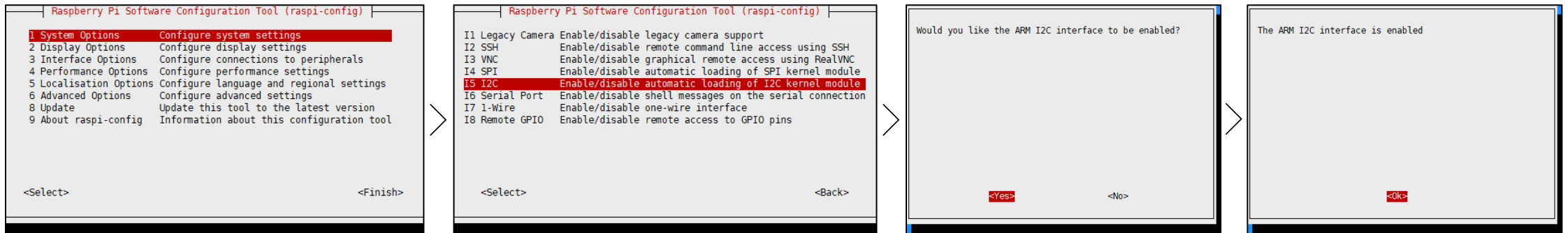
- LCD 모듈 – 1602 LCD Datasheet 참고
 - CGROM 표현 방식
 - 영어와 숫자의 경우, 8-bit 데이터는 ASCII 코드와 동일함
 - 따라서 프로그래밍 시, char 형식 데이터를 bit 형식으로 바꿔서 Display해주면 됨

12. Standard character pattern

| Upper 4bit Lower 4bit | LLLL | LLH | LLHL | LLHH | LHL | LHLH | LHHL | LHHH | HLLL | HLLH | HLHL | HLHH | HLLH | HHLH | HHHL | HHHH |
|--------------------------------|------------------|-----|------|-------|-----|------|------|------|------|------|--------|-------|------|------|------|------|
| LLLL | CG RAM (1) | | | 00P`P | | | | | | | | 一ヲΞxp | | | | |
| LLH | (2) | | ! | 1A0a9 | | | | | | | 。アチ△≡9 | | | | | |
| LLHL | (3) | | " | 2BRbr | | | | | | | 「イウ×pθ | | | | | |
| LLHH | (4) | | # | 3C5cs | | | | | | | 」ウてEε∞ | | | | | |
| LHL | (5) | | \$ | 4DTdt | | | | | | | 、エトtμΩ | | | | | |
| LHLH | (6) | | % | 5EUeu | | | | | | | ・オオIεO | | | | | |
| LHHL | (7) | | & | 6FVfv | | | | | | | ヲカニヨpΣ | | | | | |
| LHHH | (8) | | ' | 7GWgw | | | | | | | アキヌヲgπ | | | | | |
| HLLL | (1) | | (| 8HXhx | | | | | | | イウネリ、又 | | | | | |
| HLLH | (2) | |) | 9IYiy | | | | | | | ウツル、y | | | | | |
| HLHL | (3) | | * | JZjz | | | | | | | エコハレJ≠ | | | | | |
| HLHH | (4) | | + | KLk< | | | | | | | オヲヒロ×% | | | | | |
| HHL | (5) | | , | <L¥ll | | | | | | | ヤシフワΦm | | | | | |
| HHLH | (6) | | — | =MIm} | | | | | | | ユズへント÷ | | | | | |
| HHHL | (7) | | . | >N^n÷ | | | | | | | ヨセホ、ん | | | | | |
| HHHH | (8) | | / | ?0_0+ | | | | | | | ウリマ°6 | | | | | |





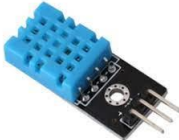
Raspberry Pi I2C 인터페이스 활성화

- I2C 인터페이스 활성화
 - XSHELL을 통해 Raspberry Pi에 원격 접속 후, 아래 명령어 입력
 - ✓ **sudo raspi-config**
 - **[3 Interface Options]** 선택
 - **[I5 I2C Enable/disable automatic loading of I2C kernel module]** 선택
 - **[Yes]** 선택
 - **[Finish]** 선택



LCD 모듈 연결

- 1. 구성품 준비

| 번호 | 구성요소 | 사진 |
|----|-----------------|--|
| 1 | Raspberry Pi 본체 | <div> <Raspberry Pi 3 Model B+></div> <div> <Raspberry Pi 4 Model B></div> |
| 2 | 점프 와이어(F/F 7개) |  |
| 3 | LCD 모듈 |  |
| 4 | 온습도 센서 모듈 |  |

LCD 모듈 연결

• 2. 구성품 연결

- [LCD 모듈] (LCD 1602 - 라즈베리 파이)

✓ [점프 와이어(F/F)]로 연결

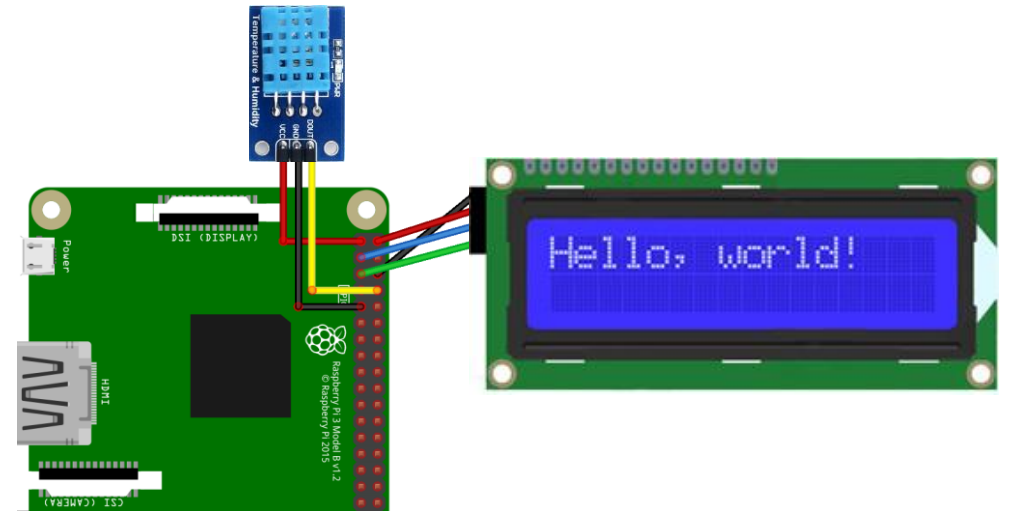
| LED 센서 모듈 | GPIO Pins |
|-----------|------------------|
| SDA | 8 (Serial Data) |
| SCL | 9 (Serial Clock) |
| VCC | 5.0 VDC |
| GND | Ground |

- [DHT11 온습도 센서] (DHT11 - 라즈베리 파이)

✓ [점프 와이어(F/F)]로 연결

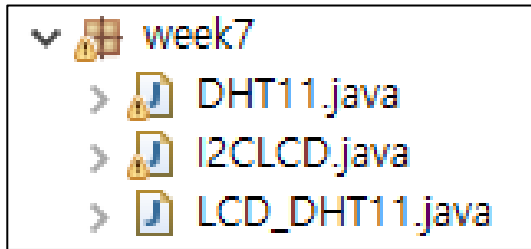
| DHT11 온습도 센서 | GPIO Pins |
|--------------|-----------|
| DOUT | 15 |
| GND | Ground |
| VCC | 3.3 VDC |

| GPIO# | NAME | | NAME | GPIO# |
|-------|-------------------|----|---------------------|-------|
| | 3.3 VDC Power | 1 | 5.0 VDC Power | 2 |
| 8 | GPIO 8 SDA1 (I2C) | 3 | 5.0 VDC Power | 4 |
| 9 | GPIO 9 SCL1 (I2C) | 5 | Ground | 6 |
| 7 | GPIO 7 GPCLK0 | 7 | GPIO 15 TxD (UART) | 15 |
| | Ground | 9 | GPIO 16 RxD (UART) | 16 |
| 0 | GPIO 0 | 11 | GPIO 1 PCM_CLK/PWM0 | 1 |
| 2 | GPIO 2 | 13 | Ground | 14 |
| 3 | GPIO 3 | 15 | GPIO 4 | 4 |



LCD 모듈 제어

- 1. LCD_DHT11.java: 라즈베리 파이에서 DHT11로부터 온습도 데이터를 읽어와, 해당 값을 I2C 인터페이스를 통해 LCD 디스플레이에 표시
 - (a): 무한 루프문
 - ✓ 온습도 데이터를 주기적으로 읽어오고, 이를 LCD 디스플레이에 출력함



```
import com.pi4j.io.i2c.I2CBus;
import com.pi4j.io.i2c.I2CDevice;
import com.pi4j.io.i2c.I2CFactory;

public class LCD_DHT11 {

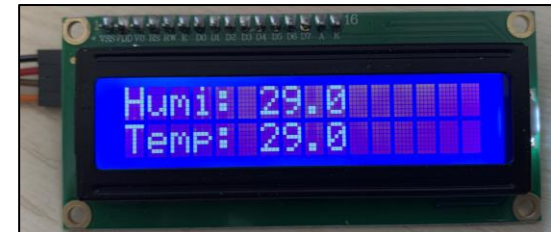
    public static void main(String[] args) {
        try {
            DHT11 dht = new DHT11(); // DHT11 Class 객체 생성
            I2CBus bus = I2CFactory.getInstance(I2CBus.BUS_1); // I2C Bus 객체 생성
            I2CDevice dev = bus.getDevice(0x27); // I2C Device 객체 생성
            I2CLCD lcd = new I2CLCD(dev); // I2CLCD Class 객체 생성, 생성한 I2C Device 객체를 인자값으로 넣어줌
            lcd.init(); // 초기화
            lcd.backlight(true); // Back light on

            Thread.sleep(2000);
        }
        catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

LCD 모듈 제어

- 2. JAR 파일 생성 후 XFTP를 통해 Raspberry Pi로 전송
- 3. Raspberry Pi에서 JAR 파일 실행
 - `sudo java -jar lcd_dht11.jar`
- 4. 결과
 - LCD 모듈에 습도/온도가 출력되는 것을 알 수 있음
 - Error 발생 시, I2C 인터페이스 활성화 여부를 했는지 다시 확인할 것
 - Connection Error 발생할 경우, Raspberry Pi 재부팅 후, JAR 파일 재실행

```
pi@raspberrypi:~/ES_proj $ sudo -s java -jar lcd_dht11.jar
Checksum Error
Checksum Error
Humidity = 88.0% Temperature = 27.0°C | 80.6°F)
Humidity = 90.0% Temperature = 27.0°C | 80.6°F)
```





감사합니다

Thank You

