SPI IN MBED-OS

Handong university

Jong-won Lee

2018.09.18

SPI

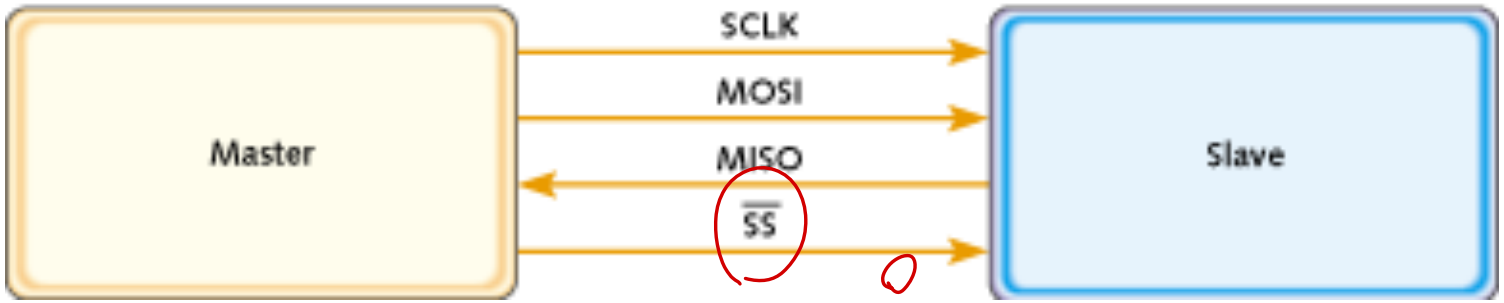
SPI

full duplex

- Serial Peripheral Interface
  - Defined by Motorola (Freescale)
  - Synchronous serial communication operating at a full duplex
  - Between Master-slave devices
  - Generally faster than I2C, capable of several Mbps
  - The SPI defines no protocol for data exchange.
    - Does not define the structure of the data stream

SPI

I²C에서는 slave의 주소로 구별

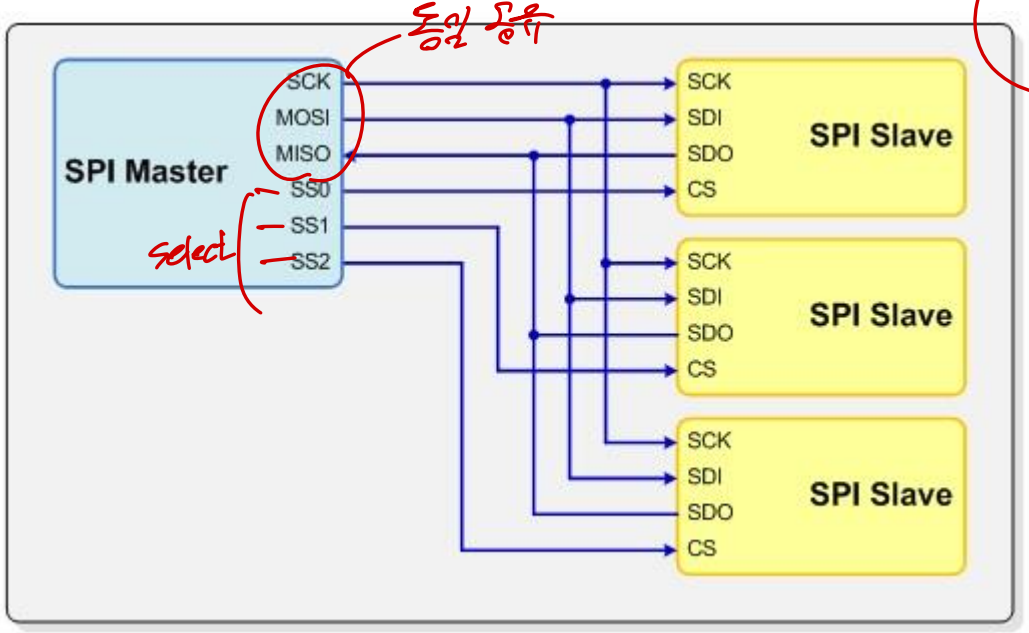


- 2 control signals
  - SCLK: clock
    - Provided by the master
  - /SS: slave select (구별)
- 2 data signals
  - MOSI: master data output, slave data input
  - MISO: master data input, slave data output

L negative값 또는 SS가 0일 때 slave를 선택

SPI

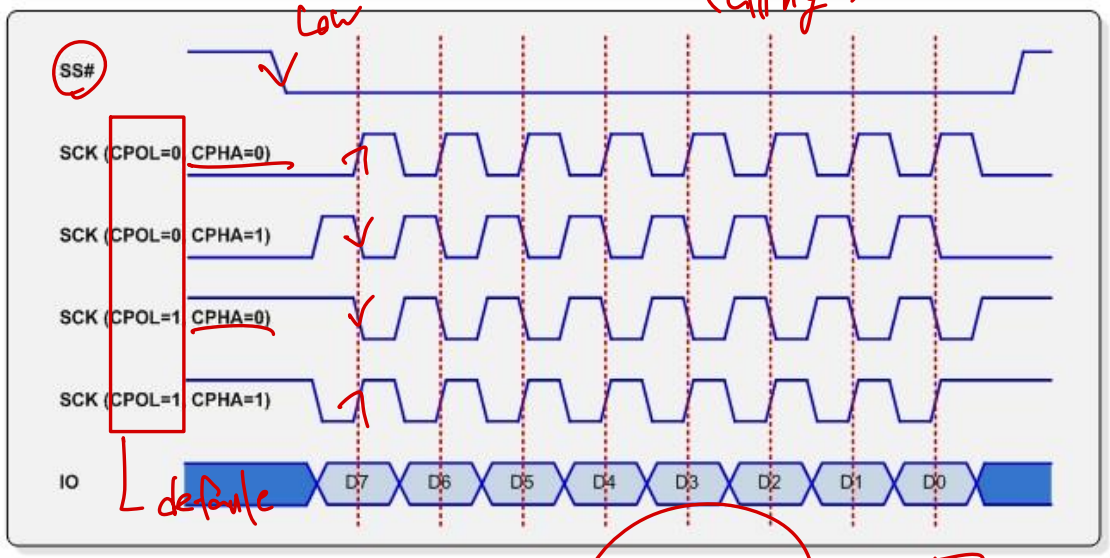
- Master-slave communication
  - For one slave or
  - For multiple slaves
    - Each slave needs separate slave select signal



SPI

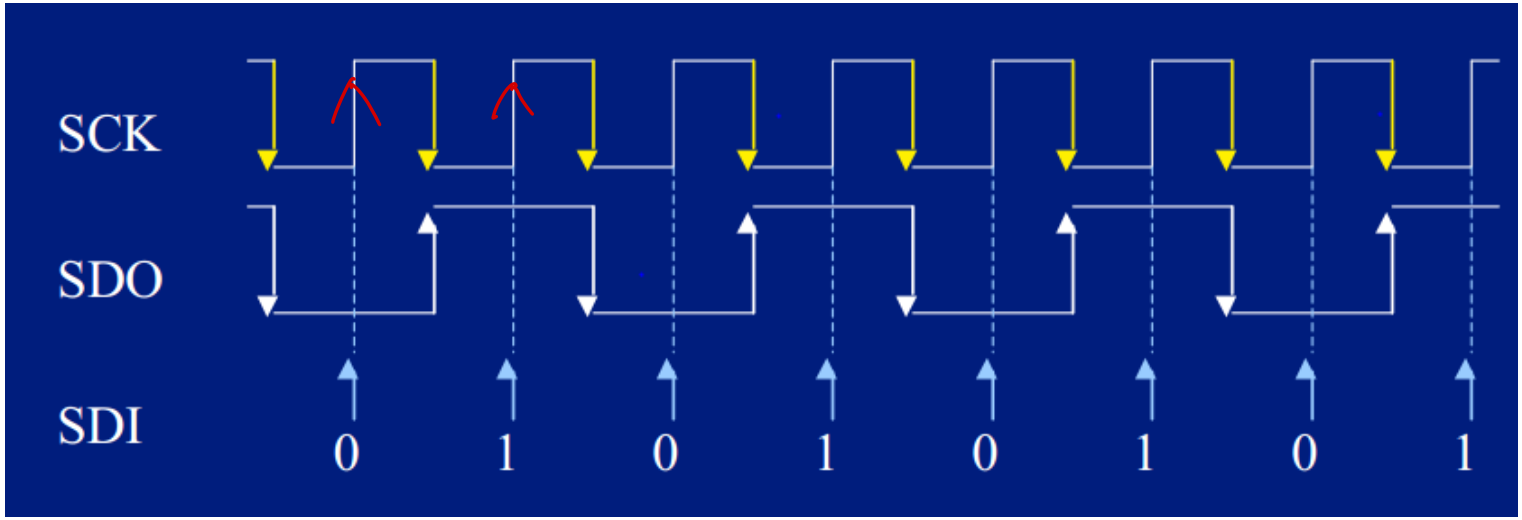
- Clock polarity (CPOL) and clock phase (CPHA) determine the active edge of the clock
- Master and slave must agree on the parameters to communicate.

Mode	CPOL	CPHA
0	0	0
1	0	1
2	1	0
3	1	1



SPI

- Data is only output during the rising or falling edge of SCK
- The opposite edge is used to ensure data is valid at the time of reading

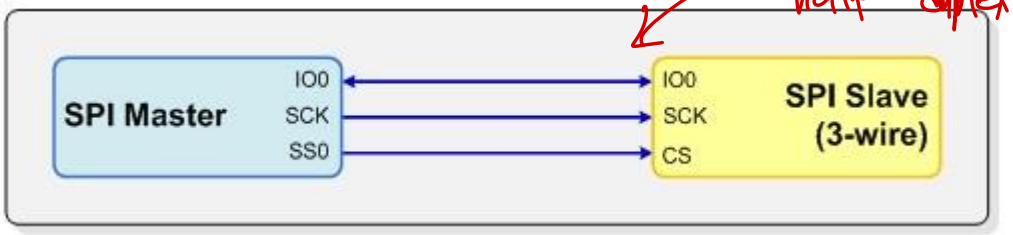


MOSI

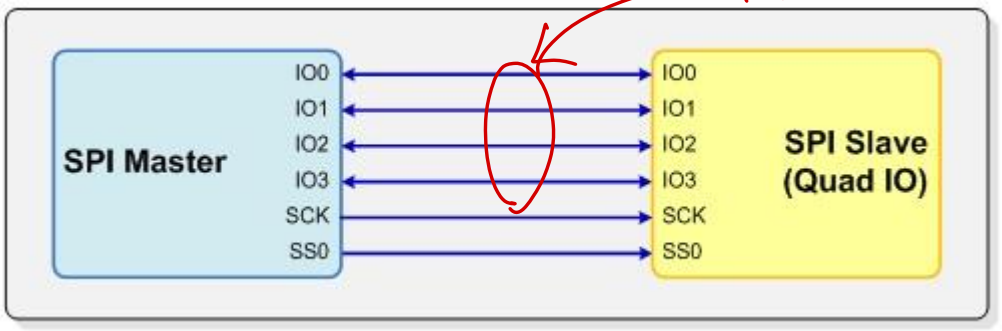
MISO

SPI

- 3-Wire SPI configuration



- Quad IO SPI configuration (multi-IO configuration)



SPI in Nucleo-F411RE

SPI in Nucleo-F411RE

- 5 SPI in STM32F411
  - SPI1, SPI2, SPI3, SPI4, SPI5
    - SPI1, SPI4, SPI5: Up to 50Mbps
    - SPI2, SPI3: Up to 25MHz
- CS/SCL/MOSI/MISO pins for SPI

50Mbps
25MHz
slave는 다 고 rate가 있음.
둘 다 만족하는거 써

SPI in Nucleo-F411RE

- SPI Pins in Mbed-OS

SPI 신호	핀 번호	외부 확장 커넥터	SPI 신호	핀 번호	외부 확장 커넥터
SPI1_SCK	PA_5 PB_3	D13 D3 (SW0)	SPI1_NSS	PA_4, PA_15	A2, CN7-17
SPI1_MOSI	PA_7 PB_5	D11, D4	SPI1_MISO	PA_6 PB_4	D12, D5
SPI2_SCK	PB_10 PB_13 PC_7	D6 CN10-30 D9	SPI2_NSS	PB_9 PB_12	D14 CN10-16
SPI2_MOSI	PB_15 PC_3	CN10-26 CN7-37	SPI2_MISO	PB_14 PC_2	CN10-28 CN7-35
SPI3_SCK	PB_3_ALT0 PC_10 PB_12	D3 (SW0) CN7-1 CN10-16	SPI3_NSS	PA_4_ALT0, PA_15-ALT0	A2, CN7-17
SPI3_MOSI	PB_5_ALT0 PC_12	D4 CN7-3	SPI3_MISO	PB_4_ALT0 PC_11	D5 CN7-2
SPI4_SCK	PB_13_ALT0	CN10-30	SPI4_NSS	PA_12_ALT0	CN10-12
SPI4_MOSI	PA_1	A1	SPI4_MISO	PA_11	CN10-14
SPI5_SCK	PB_0	A3	SPI5_NSS	PB_1	CN10-24
SPI5_MOSI	PA_10 PB_8	D2 D15	SPI5_MISO	PA_12	CN10-12

SPI in Nucleo-F411RE

- SPI Pins in Mbed-OS

SPI 신호	핀 번호	외부 확장 커넥터	SPI 신호	핀 번호	외부 확장 커넥터
SPI1_SCK	PA_5 PB_3	D13 D3 (SW0)	SPI1_NSS	PA_4, PA_15	A2, CN7-17
SPI1_MOSI	PA_7 PB_5	D11, D4	SPI1_MISO	PA_6 PB_4	D12, D5
SPI2_SCK	PB_10 PB_13 PC_7	D6 CN10-30 D9	SPI2_NSS	PB_9 PB_12	D14 CN10-16
SPI2_MOSI	PB_15 PC_3	CN10-26 CN7-37	SPI2_MISO	PB_14 PC_2	CN10-28 CN7-35

SPI in Nucleo-F411RE

- SPI Pins in Mbed-OS

SPI 신호	핀 번호	외부 확장 커넥터	SPI 신호	핀 번호	외부 확장 커넥터
SPI3_SCK	PB_3_ALT0 PC_10 PB_12	D3 (SW0) CN7-1 CN10-16	SPI3_NSS	PA_4_ALT0, PA_15_ALT0	A2, CN7-17
SPI3_MOSI	PB_5_ALT0 PC_12	D4 CN7-3	SPI3_MISO	PB_4_ALT0 PC_11	D5 CN7-2
SPI4_SCK	PB_13_ALT0	CN10-30	SPI4_NSS	PA_12_ALT0	CN10-12
SPI4_MOSI	PA_1	A1	SPI4_MISO	PA_11	CN10-14
SPI5_SCK	PB_0	A3	SPI5_NSS	PB_1	CN10-24
SPI5_MOSI	PA_10 PB_8	D2 D15	SPI5_MISO	PA_12	CN10-12

Mbed-OS SPI API

SPI class

생성자	SPI (PinName mosi, PinName miso, PinName sclk, PinName ssel=NC)
동작	SPI 통신으로 사용할 MOSI, MISO, SCLK 핀을 주어야 한다. 만약 단방향 통신만이 사용될 경우, mosi 혹은 miso를 NC로 설정할 수 있다. ssel 인자는 SPI_NSS 핀으로 사용되는 핀을 정의하는 것이며, 해당 핀에 대한 값을 정해주지 않으면 /SS 핀을 별도로 제어한다는 의미이다. (하지만 SPI API 함수 내에서 직접 /SS 핀을 제어하지 않기 때문에 별도로 프로그램에 의해서 제어되어야 한다.) 해당 SPI는 모드 0, 8비트 포맷, 전송 속도는 1Mbps로 설정된다. 그리고 디폴트 전송 데이터 값은 0xFF이다.
함수	void format (int bits, int mode = 0)
동작	시리얼 통신의 포맷을 설정한다. 인자: bits = SPI 프레임 당 비트 수를 결정한다. 허용 값은 8또는 16이다. mode = SPI의 동작 모드를 결정한다. 가능한 값은 0/1/2/3이며, 디폴트 모드는 0이다.

Mbed-OS SPI API

- SPI class

함수	int write (int value)
동작	슬레이브로 설정된 포맷에 따라 value 값을 전송한다. 반환 인자: 슬레이브로부터의 데이터이다.
함수	int write (const char *tx_buffer, int tx_length, char *rx_buffer, int rx_length)
동작	주어진 슬레이브로 데이터를 전송하고, 동시에 데이터를 수신하여 저장한다. 전송하는 데이터 수는 max(tx_length, rxlength)가 된다. 만약 tx_length 〈 rx_length이면 전송되는 데이터는 디폴트 전송 값이 된다. (이 값은 set_ default_ wrrite_value() 에 의해서 결정된다.) 인자: tx_buffer = 전송할 데이터가 저장된 버퍼 포인터, tx_length = 전송할 데이터 개수, rx_buffer = 슬레이브로부터 수신한 데이터 저장할 버퍼 포인터, rx_length = 수신할 데이터 개수. 반환 인자: 슬레이브에 전송한 데이터 수.

Mbed-OS SPI API

- SPI class

함수	void frequency (int hz = 1000000)
동작	시리얼 통신에 사용할 SCLK 클럭의 속도를 hz 파라미터로 설정한다. 디폴트 속도는 1 Mbps이다.
함수	void set_default_write_value (char data)
동작	슬레이브로부터 데이터를 읽을 경우, 특별히 전송할 값을 주지 않은 경우에 전송에 사용되는 데이터 값이다.

Mbed-OS SPI API

- SPI class: Non-blocking function

int	**transfer** (const Type *tx_buff, int tx_length, Type *rx_buff, int rx_length, const event_callback_t &callback, int event = SPI_EVENT_ COMPLETE);
동작	비동기식 방식으로 데이터를 전송하고 읽는다. tx_buff: 전송할 데이터를 저장한 버퍼 포인터. tx_length: 전송할 데이터 바이트 수. rx_buff: 읽은 데이터 저장할 버퍼 포인터, rx_length: 읽을 데이터 바이트 수 callback: 이벤트가 발생하였을 때, 불리는 콜백 함수이다. event: 콜백 함수를 부르는 이벤트의 종류이다. (SPI_EVENT_ERROR, SPI_EVENT_COMPLETE, SPI_EVENT_RX_OVERFLOW, SPI_EVENT_ALL) 반환값은 '0'이면 성공, SPI 주변기기가 다른 일을 하고 있으면 -1이다.

SPI LAB

Handong university

Jong-won Lee

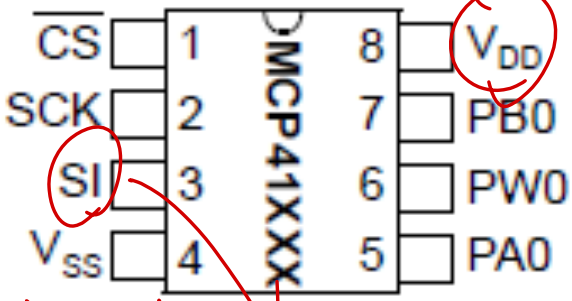
2021.05.18

MCP41010:

Digital Potentiometer (가변 저항)

MCP41010

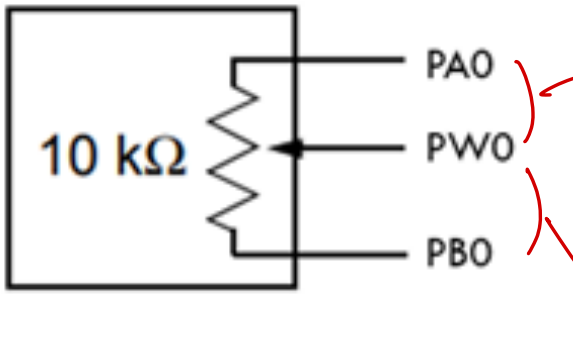
- MCP 41010
  - Single supply operation: 2.7V ~ 5.5V
  - Potentiometer of 10 k, and 256 taps
  - SPI interface: support mode 0,0 or 1,1
  - Max. clock rate of SPI: ~ 5.8MHz
  - 8 pin interface



핀 이름	기능	핀 이름	기능
CS	SPI의 /SS	VDD	전원 (2.7V ~ 5.5V)
SCK	SPI의 SCLK	PB0	가변저항의 양단 터미널
SI	SPI의 MOSI	PW0	가변저항의 중간 터미널
VSS	GND	PA0	가변저항의 양단 터미널

MCP41010

- MCP 41010: Operation



$$R_{WA} = \frac{R_{AB} \times (256 - Val)}{256} + R_W$$

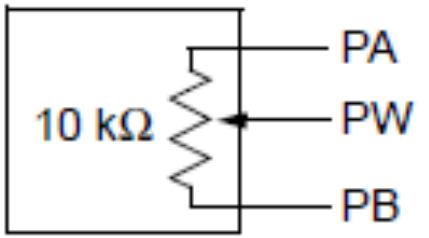
$$R_{WB} = \frac{R_{AB} \times Val}{256} + R_W$$

R_W: wiper resistance (52Ω)

MCP41010

val = 0xC0 = 192

- MCP 41010: Operation example



Example:
R = 10 kΩ
Code = C0h = 192d

$$R_{WA}(D_n) = \frac{(R_{AB})(256 - D_n)}{256} + R_W$$

$$R_{WA}(C0h) = \frac{(10k\Omega)(256 - 192)}{256} + 52\Omega$$

$$R_{WA}(C0h) = 2552\Omega$$

$$R_{WB}(D_n) = \frac{(R_{AB})(D_n)}{256} + R_W$$

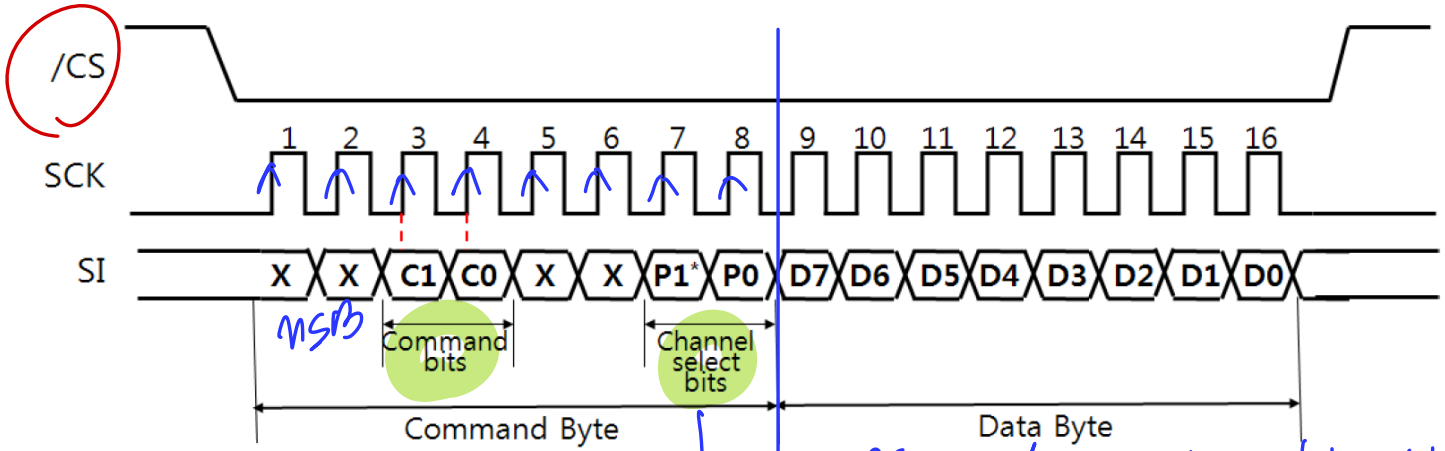
$$R_{WB}(C0h) = \frac{(10k\Omega)(192)}{256} + 52\Omega$$

$$R_{WB}(C0h) = 7552\Omega$$

R_W: wiper resistance

MCP41010

- SPI commun. format



C1 C0	동작	P1 P0	동작
0 0	No command	0 0	어떤 가변저항도 선택되지 않음.
0 1	가변저항 제어 데이터 전송	0 1	가변 저항 0 선택
1 0	셧다운	1 0	가변 저항 1 선택
1 1	No command	1 1	가변 저항 1 & 2 선택

P1: don't care for MCP41xxx

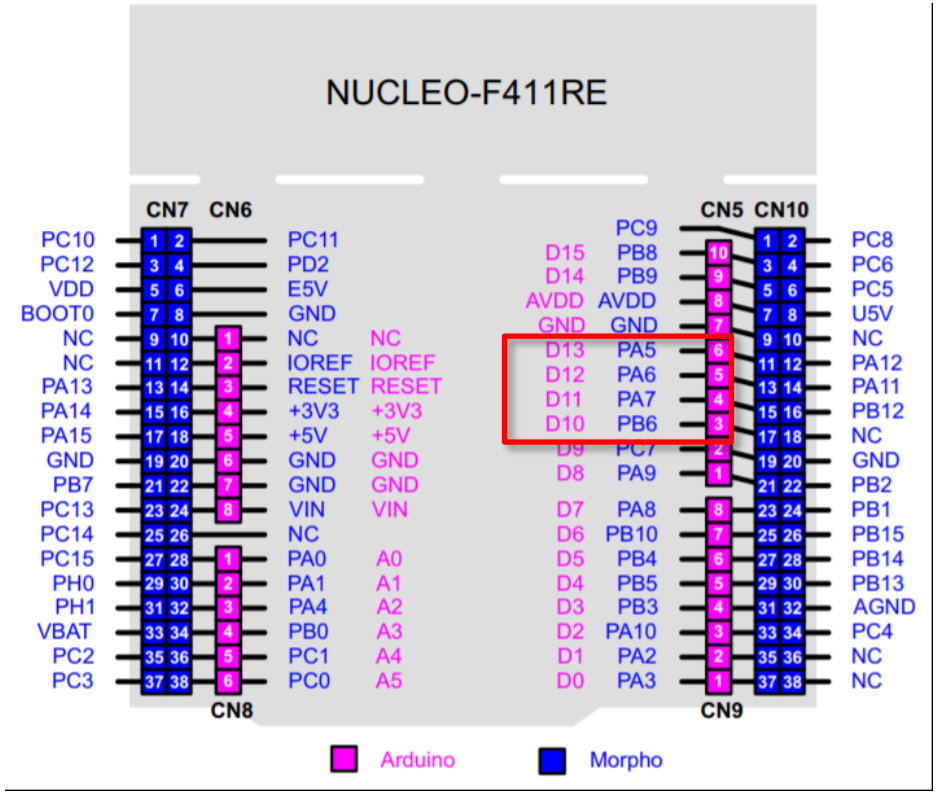
Lab12-1

0 ~ 255 입력

- 실습 목적
  - SPI 인터페이스가 어떻게 동작하는지 이해할 수 있다.
  - MCP41010 디지털 가변저항의 특성을 이해하고 목적에 맞게 사용할 수 있다.
- 실습 시나리오
  - 터미널부터 디지털 가변저항 값을 입력 받아 디지털 가변 저항 값을 조절한다.
  - 변화된 가변 저항 값에 의해서 선형적으로 변하는 전압 값을 ADC를 통하여 읽음으로써, SPI를 사용해서 정상적으로 MCP41010을 제어하고 있는 가를 확인한다.
  - SPI의 데이터 전송 포맷은 8 bit 모드를 이용한다.

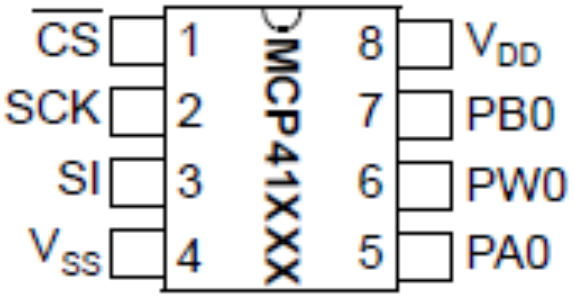
SPI in Nucleo-F411

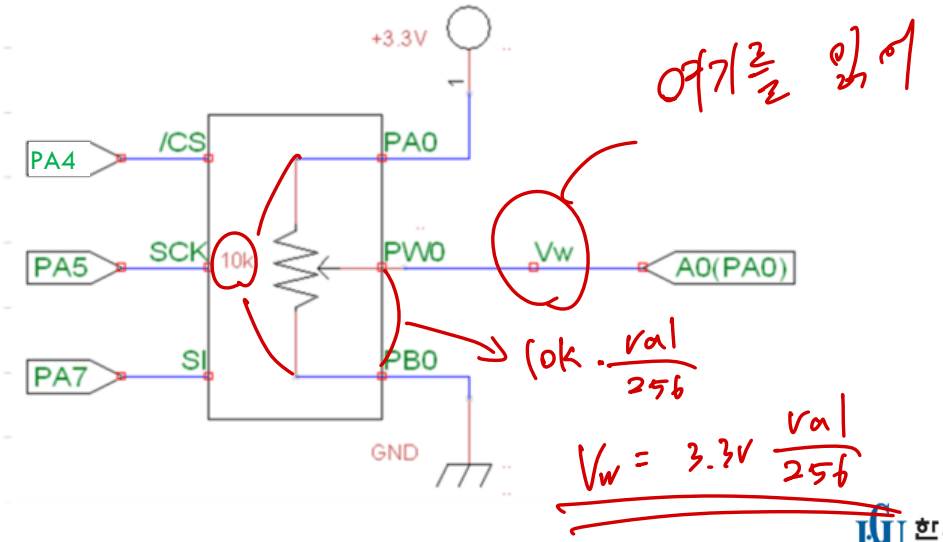
- SPI Pins in Mbed
  - MOSI: ADUINO_UNO_D11
  - MISO: ARDUINO_UNO_D12
  - SCK: ARDUINO_UNO_D13
  - /SS: ARDUINO_UNO_A2 (PA_4)



Lab12-1

- 회로 구성
  - VSS and PB0 of MCP41010 (pin 4, 7) ⇔ CN6-6 (GND)
  - VDD and PA0 of MCP41010 (pin 5, 8) ⇔ CN6-4 (+3.3V)
  - /CS pin of MCP41010 (pin 1) ⇔ A2 (PA_4)
  - SI pin of MCP41010 (pin 3) ⇔ D11 (PA_7)
  - SCK pin of MCP41010 (pin 2) ⇔ D13 (PA_5)
  - PW0 pin of MCP41010 (pin 6) ⇔ A0 (PA_0)





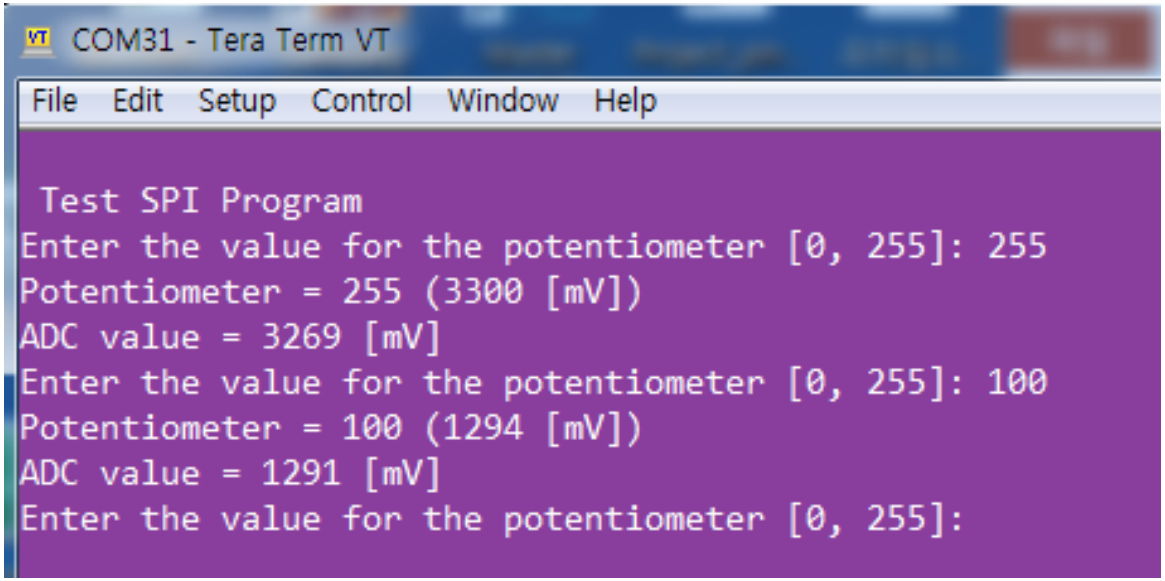
Lab12-1

- A sample code

```
#include "mbed.h"

SPI spi(ARDUINO_UNO_D11, ARDUINO_UNO_D12, ARDUINO_UNO_D13);
DigitalOut cs(PA_4);

UnbufferedSerial pc(CONSOLE_TX, CONSOLE_RX, 115200);
AnalogIn ain0(ARDUINO_UNO_A0);
```

Lab12-1

- Result



128X64 OLED & SSD1306

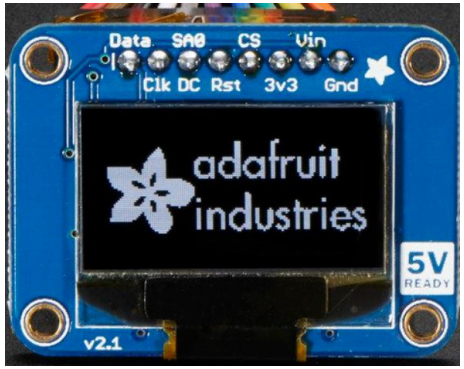
Handong university

Jong-won Lee

2022.05.10

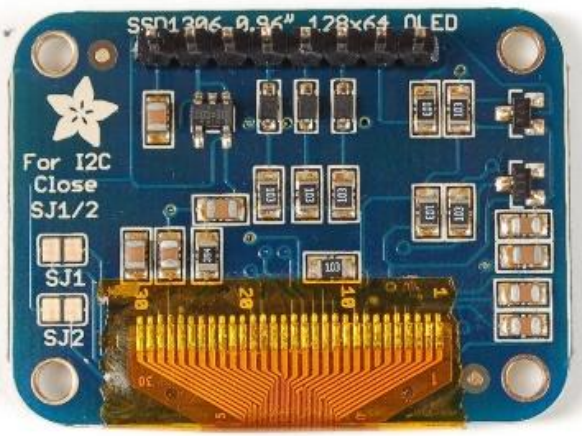
Adafruit Monochrome 0.96" 128x64 OLED

- Monochrome 0.96" 128x64 OLED
- OLED driver IC: SSD1306
- Interface
  - **SPI (most popular)**, I2C, and 8-bit parallel
- Library for Arduino
  - Adafruit GFX library: low level library
    - https://github.com/adafruit/Adafruit-GFX-Library
  - Adafruit SSD1306 library: SSD 1306 driver library
    - requires GFX library. (for defined I2C or SPI interface)
    - https://github.com/adafruit/Adafruit_SSD1306



Adafruit Monochrome 0.96" 128x64 OLED

- Connections for SPI (default mode)
  - SJ1 and SJ2 are open.
  - Connect pins
    - GND
    - Vin: 5V
    - Data: (MOSI)
    - Clk: (CLK)
    - D/C#: Data/Command#
    - RST: Reset
    - CS: (CS)



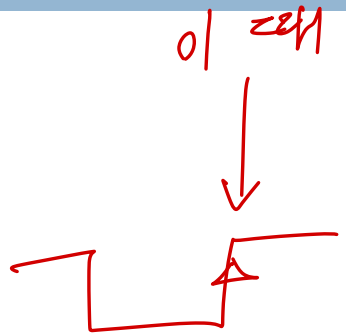
Adafruit Monochrome 0.96" 128x64 OLED

- Connections for I2C
  - SJ1 and SJ2 are closed.

  - Connect pins
    - GND
    - Vin: 5V
    - Data: (SDA): (has pull-up resistor)
    - Clk: (SCL): (has pull-up resistor)
    - D/C#(SA0): (has pull-up resistor)
      - If this pin is open, the 7-bit address of I2C is 0x3D.
      - If the pin is connected to GND, the 7-bit slave address is 0x3C.
    - RST: Reset (has pull-up resistor)

Adafruit Monochrome 0.96" 128x64 OLED

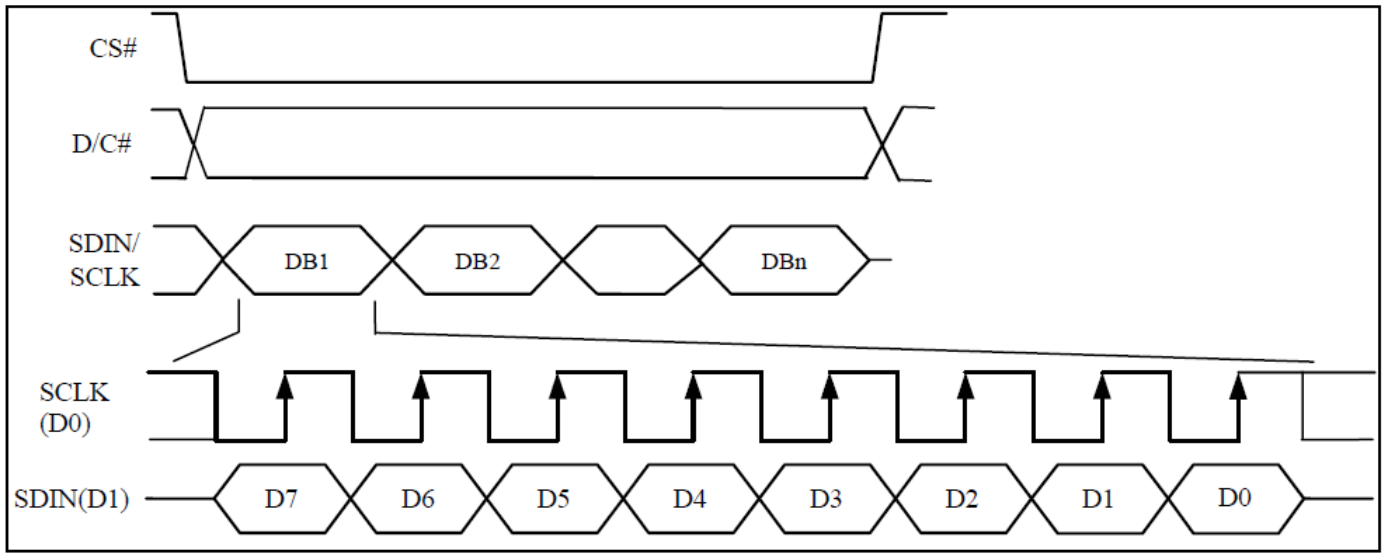
- 4-wire SPI format
  - SCLK, MOSI (data), CS#, D/C#
  - Support Mode: 3 (1,1)
  - Max. clock: 10 MHz
  - Read data on rising edge of SCLK
    - MSB first transmitted
  - D/C# is sampled on every eighth clock (D0) and the data byte in the shift register is written to the Graphic Display Data RAM (GDDRAM) or command register in the same clock.



Function	E(RD#)	R/W#(WR#)	CS#	D/C#	D0
Write command	Tie LOW	Tie LOW	L	L	↑
Write data	Tie LOW	Tie LOW	L	H	↑

Adafruit Monochrome 0.96" 128x64 OLED

- 4-wire SPI format
  - SPI format

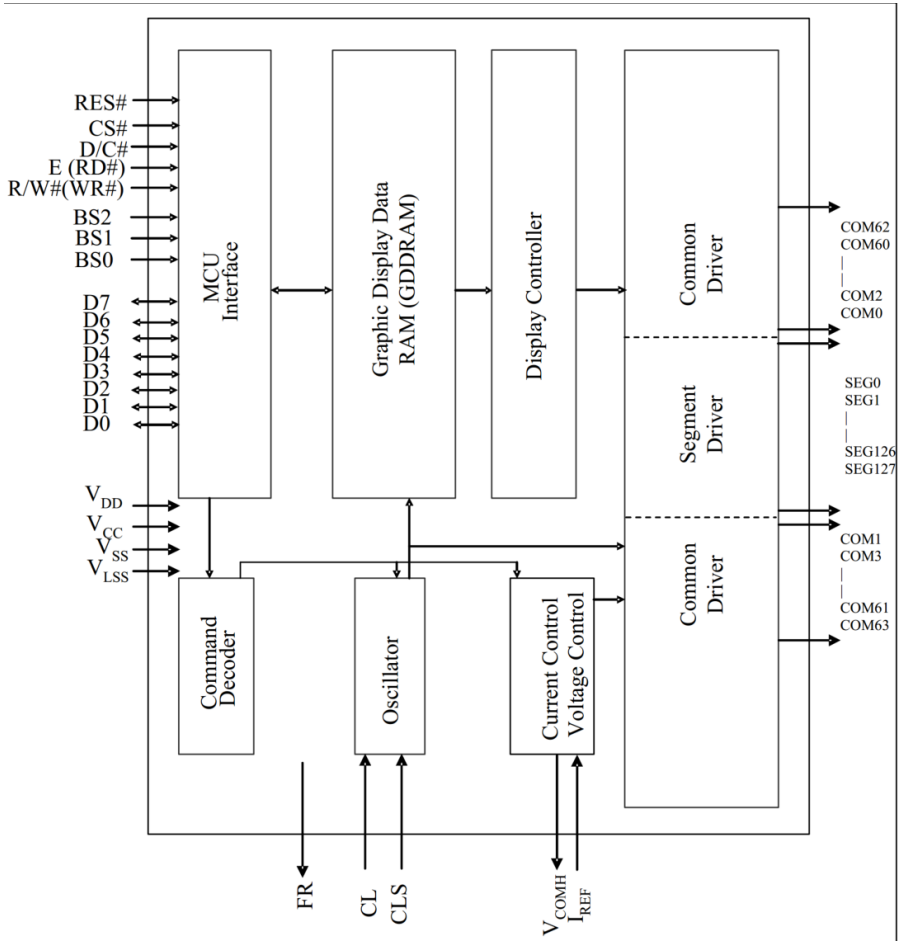


SSD1306

SSD1306

- SSD1306 features
  - Embedded 128 x 64 bit SRAM display buffer
  - Screen saving continuous scrolling function in both horizontal and vertical direction
  - Programmable Frame Rate and Multiplexing Ratio
  - Row Re-mapping and Column Re-mapping

- SSD1306 Block Diagram



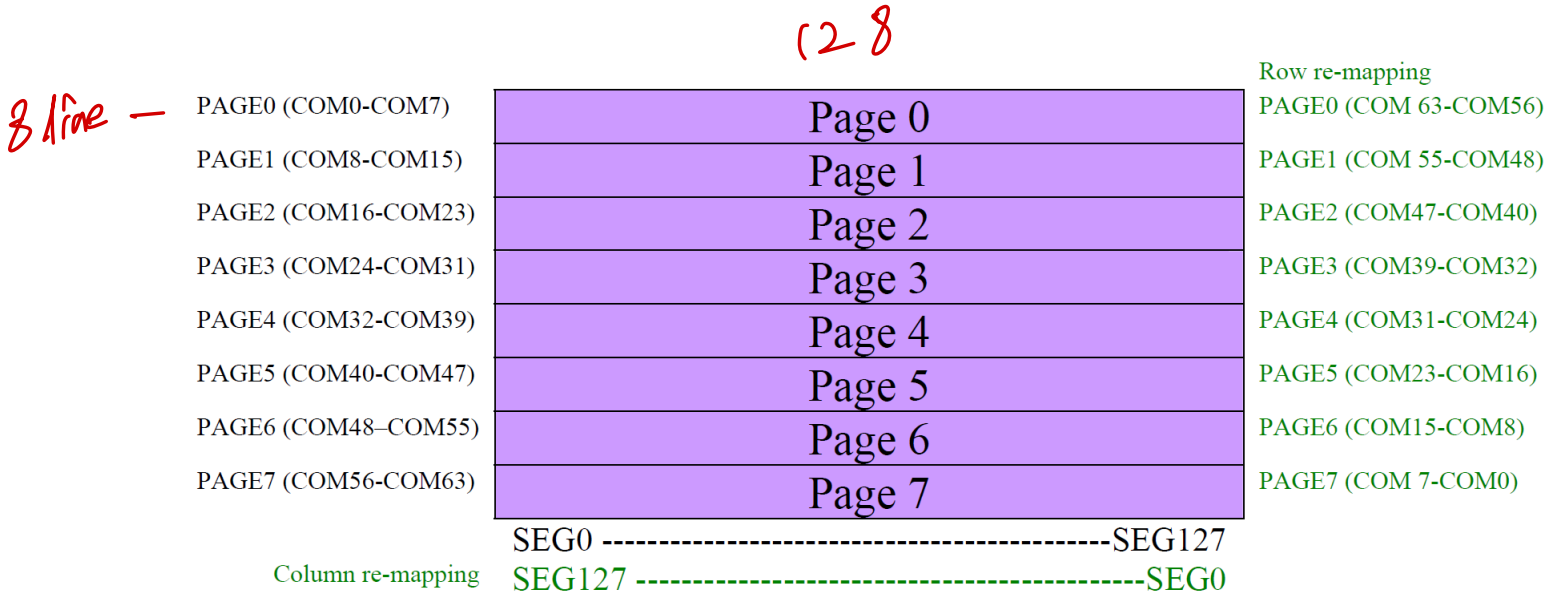
SSD1306

- Controller chip: SSD1306
  - Interface mode & pins
    - Mode is determined by BS[2:0]

SSD1306 Pin Name	I^2C Interface	6800-parallel interface (8 bit)	8080-parallel interface (8 bit)	4-wire Serial interface	3-wire Serial interface
BS0	0	0	0	0	1
BS1	1	0	1	0	0
BS2	0	1	1	0	0

SSD1306: GDDRAM Structure

- The size of the RAM is 128 x 64 bits and the RAM is divided into eight pages, from PAGE0 to PAGE7

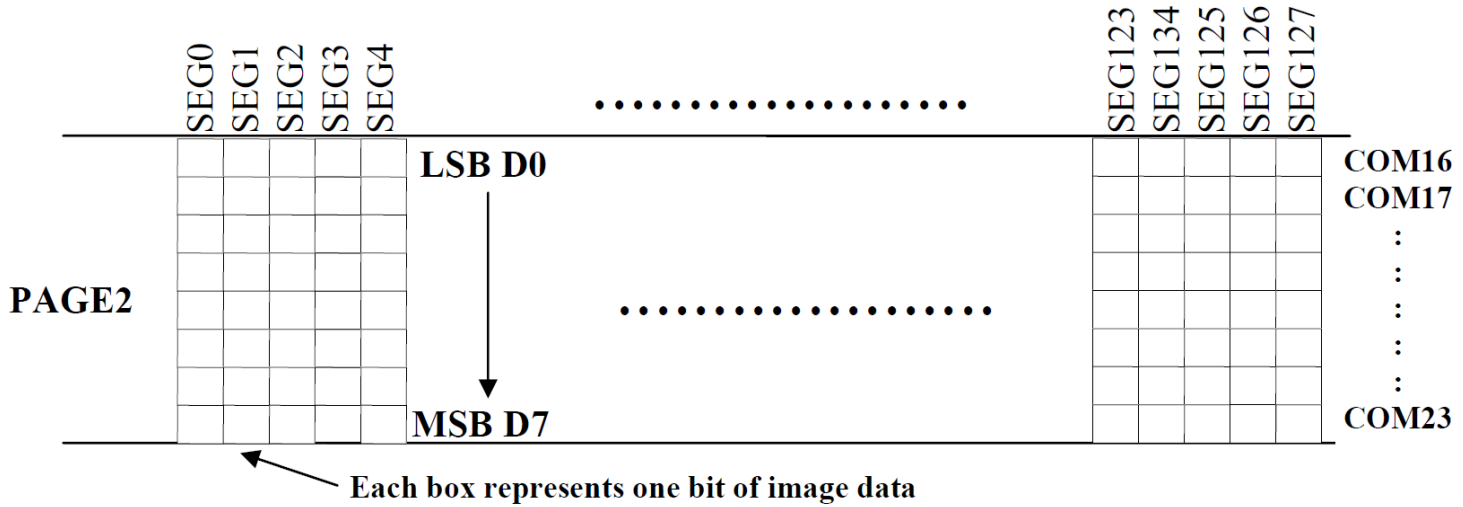


SSD1306: GDDRAM Structure

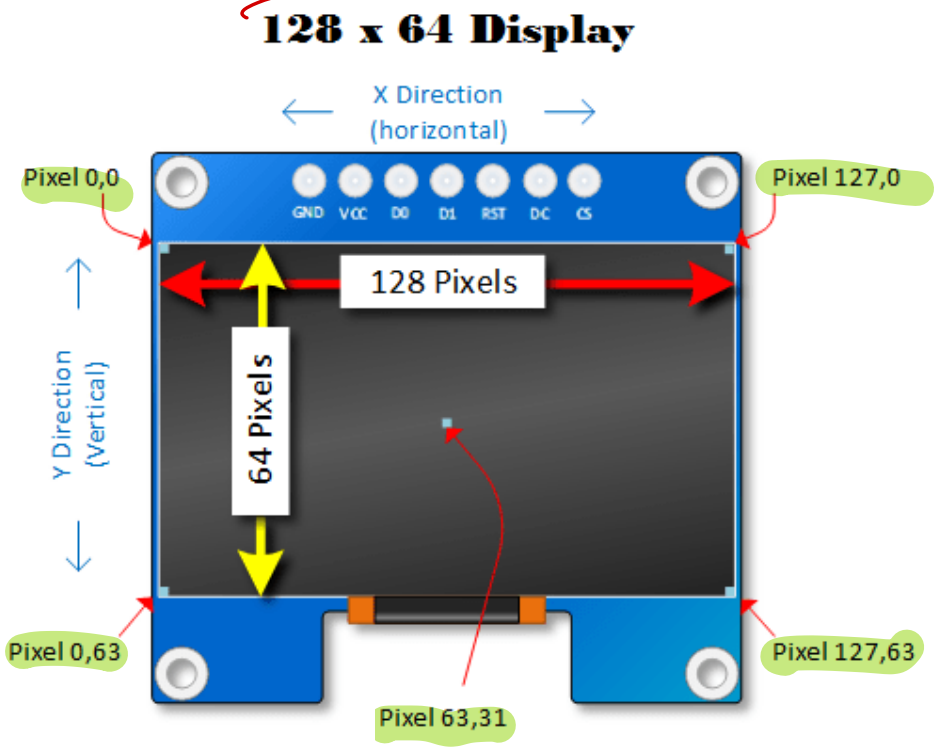
- When one data byte is written into GDDRAM, all the rows image data of the same page of the current column are filled (i.e. the whole column (8 bits) pointed by the column address pointer is filled.).
- Data bit D0 is written into the top row, while data bit D7 is written into bottom row

matching

Figure 8-14 : Enlargement of GDDRAM (No row re-mapping and column-remapping)



The Adafruit GFX Coordinate System



Common Referencing Format

(X Position , Y Position)

Pixel Number from left to right Where Zero is left most.

Pixel Number from up to top Where Zero is top most.

SSD1306: Commands

- SSD1306: command table

1. Fundamental Command Table											
D/C#	**Hex**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**	**Command**	**Description**
0 0	81 A[7:0]	1 A_7	0 A_6	0 A_5	0 A_4	0 A_3	0 A_2	0 A_1	1 A_0	Set Contrast Control	Double byte command to select 1 out of 256 contrast steps. Contrast increases as the value increases. (RESET = 7Fh)
0	A4/A5	1	0	1	0	0	1	0	X_0	Entire Display ON	A4h, X_0=0b: Resume to RAM content display (RESET) Output follows RAM content A5h, X_0=1b: Entire display ON Output ignores RAM content
0	A6/A7	1	0	1	0	0	1	1	X_0	Set Normal/Inverse Display	A6h, X[0]=0b: Normal display (RESET) 0 in RAM: OFF in display panel 1 in RAM: ON in display panel A7h, X[0]=1b: Inverse display 0 in RAM: ON in display panel 1 in RAM: OFF in display panel
0	AE AF	1	0	1	0	1	1	1	X_0	Set Display ON/OFF	AEh, X[0]=0b:Display OFF (sleep mode) (RESET) AFh X[0]=1b:Display ON in normal mode

SSD1306: Commands

- SSD1306: command table

2. Scrolling Command Table

D/C#	Hex	D7	D6	D5	D4	D3	D2	D1	D0	Command	Description
0	26/27	0	0	1	0	0	1	1	X_0	Continuous Horizontal Scroll Setup	26h, X[0]=0, Right Horizontal Scroll 27h, X[0]=1, Left Horizontal Scroll (Horizontal scroll by 1 column) A[7:0] : Dummy byte (Set as 00h) B[2:0] : Define start page address 000b – PAGE0 | 011b – PAGE3 | 110b – PAGE6 001b – PAGE1 | 100b – PAGE4 | 111b – PAGE7 010b – PAGE2 | 101b – PAGE5 C[2:0] : Set time interval between each scroll step in terms of frame frequency 000b – 5 frames | 100b – 3 frames 001b – 64 frames | 101b – 4 frames 010b – 128 frames | 110b – 25 frame 011b – 256 frames | 111b – 2 frame D[2:0] : Define end page address 000b – PAGE0 | 011b – PAGE3 | 110b – PAGE6 001b – PAGE1 | 100b – PAGE4 | 111b – PAGE7 010b – PAGE2 | 101b – PAGE5 The value of D[2:0] must be larger or equal to B[2:0] E[7:0] : Dummy byte (Set as 00h) F[7:0] : Dummy byte (Set as FFh)
0	A[7:0]	0	0	0	0	0	0	0	0		
0	B[2:0]	*	*	*	*	*	B_2	B_1	B_0		
0	C[2:0]	*	*	*	*	*	C_2	C_1	C_0		
0	D[2:0]	*	*	*	*	*	D_2	D_1	D_0		
0	E[7:0]	0	0	0	0	0	0	0	0		
0	F[7:0]	1	1	1	1	1	1	1	1		

SSD1306: Commands

- SSD1306: command table

2. Scrolling Command Table

D/C#	Hex	D7	D6	D5	D4	D3	D2	D1	D0	Command	Description
0	29/2A	0	0	1	0	1	0	X_1	X_0	Continuous Vertical and Horizontal Scroll Setup	29h, X_1X_0=01b : Vertical and Right Horizontal Scroll 2Ah, X_1X_0=10b : Vertical and Left Horizontal Scroll (Horizontal scroll by 1 column) A[7:0] : Dummy byte B[2:0] : Define start page address 000b – PAGE0 | 011b – PAGE3 | 110b – PAGE6 001b – PAGE1 | 100b – PAGE4 | 111b – PAGE7 010b – PAGE2 | 101b – PAGE5 | C[2:0] : Set time interval between each scroll step in terms of frame frequency 000b – 5 frames | 100b – 3 frames 001b – 64 frames | 101b – 4 frames 010b – 128 frames | 110b – 25 frame 011b – 256 frames | 111b – 2 frame D[2:0] : Define end page address 000b – PAGE0 | 011b – PAGE3 | 110b – PAGE6 001b – PAGE1 | 100b – PAGE4 | 111b – PAGE7 010b – PAGE2 | 101b – PAGE5 | The value of D[2:0] must be larger or equal to B[2:0] E[5:0] : Vertical scrolling offset e.g. E[5:0]= 01h refer to offset =1 row E[5:0] =3Fh refer to offset =63 rows **Note** (1) No continuous vertical scrolling is available.
0	A[2:0]	0	0	0	0	0	0	0	0		
0	B[2:0]	*	*	*	*	*	B_2	B_1	B_0		
0	C[2:0]	*	*	*	*	*	C_2	C_1	C_0		
0	D[2:0]	*	*	*	*	*	D_2	D_1	D_0		
0	E[5:0]	*	*	E_5	E_4	E_3	E_2	E_1	E_0		

SSD1306: Commands

- SSD1306: command table (scroll)

2. Scrolling Command Table											
D/C#	**Hex**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**	**Command**	**Description**
0	2E	0	0	1	0	1	1	1	0	Deactivate scroll	Stop scrolling that is configured by command 26h/27h/29h/2Ah. **Note** (1) After sending 2Eh command to deactivate the scrolling action, the ram data needs to be rewritten.
0	2F	0	0	1	0	1	1	1	1	Activate scroll	Start scrolling that is configured by the scrolling setup commands :26h/27h/29h/2Ah with the following valid sequences: Valid command sequence 1: 26h ;2Fh. Valid command sequence 2: 27h ;2Fh. Valid command sequence 3: 29h ;2Fh. Valid command sequence 4: 2Ah ;2Fh. For example, if "26h; 2Ah; 2Fh." commands are issued, the setting in the last scrolling setup command, i.e. 2Ah in this case, will be executed. In other words, setting in the last scrolling setup command overwrites the setting in the previous scrolling setup commands.

SSD1306: Commands

- SSD1306: command table (advanced Graphic commands)

6. Advance Graphic Command Table

D/C#	Hex	D7	D6	D5	D4	D3	D2	D1	D0	Command	Description
0 0	23 A[6:0]	0 *	0 *	1 A5	0 A4	0 A3	0 A2	1 A1	1 A0	Set Fade Out and Blinking	A[5:4] = 00b Disable Fade Out / Blinking Mode[RESET] A[5:4] = 10b Enable Fade Out mode. Once Fade Mode is enabled, contrast decrease gradually to all pixels OFF. Output follows RAM content when Fade mode is disabled. A[5:4] = 11b Enable Blinking mode. Once Blinking Mode is enabled, contrast decrease gradually to all pixels OFF and than contrast increase gradually to normal display. This process loop continuously until the Blinking mode is disabled. A[3:0] : Set time interval for each fade step A[3:0] | Time interval for each fade step 0000b | 8 Frames 0001b | 16 Frames 0010b | 24 Frames : | : 1111b | 128 Frames Note (1) Refer to section 10.3.1 for details.
0 0	D6 A[0]	1 0	1 0	0 0	1 0	0 0	1 0	1 0	0 A0	Set Zoom In	A[0] = 0b Disable Zoom in Mode[RESET] A[0] = 1b Enable Zoom in Mode Note (1) The panel must be in alternative COM pin configuration (command DAh A[4] =1) (2) Refer to section 10.3.2 for details.

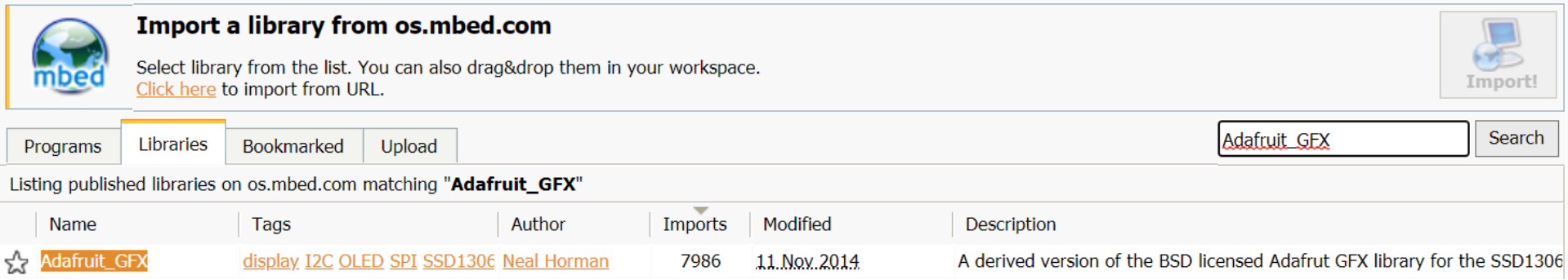
Adafruit GFX & SSD1306 Driver Libraries

Adafruit_GFX Library

- Library for Arduino
  - Adafruit GFX library: low level library
    - https://github.com/adafruit/Adafruit-GFX-Library
  - Adafruit SSD1306 library: SSD 1306 driver library
    - requires GFX library. (for defined I2C or SPI interface)
    - https://github.com/adafruit/Adafruit_SSD1306

Adafruit_GFX Library

- Import Adafruit_GFX library by Neal Horman



Adafruit_GFX Library

- In Adafruit_GFX_Conifg.h

```
#ifndef _ADAFRUIT_GFX_CONFIG_H_
#define _ADAFRUIT_GFX_CONFIG_H_

// Uncomment this to turn off the builtin splash
// #define NO_SPLASH_ADAFRUIT

// Uncomment this to enable all functionality
#define GFX_WANT_ABSTRACTS

// Uncomment this to enable only runtime font scaling, without all
the rest of the Abstracts
#define GFX_SIZEABLE_TEXT


#endif
```

Adafruit_GFX Library

- In Adafruit_SSD1306.cpp
  - Replace **wait_ms()** to **ThisThread::sleep_for()**

Adafruit GFX & SSD1306 Driver Libraries

- Reference library:
  - https://os.mbed.com/users/nkhorman/code/Adafruit_GFX/
- Class inheritance

```
class Adafruit_GFX : public Stream
{
…
```

```
class Adafruit_SSD1306 : public Adafruit_GFX
{
…
```

```
class Adafruit_SSD1306_SPI : public Adafruit_SSD1306
{
…
```

```
class Adafruit_SSD1306_I2C : public Adafruit_SSD1306
{
…
```

Adafruit_SSD1306_SPI Class

- Constructor
  - Create a SSD1306 SPI transport display driver instance with the specified DC, RST, and CS pins, as well as the display dimensions.

```
Adafruit_SSD1306_Spi ( SPI &spi, PinName DC, PinName RST, PinName CS,
                       uint8_t rawHeight = 32, uint8_t rawWidth = 128
)
```

DC: data/command pin (Data/Command#)
RST: reset pin (active low)
rawHight: the vertical number of pixels for the display, defaults to 32
rawWidth: the horizontal number of pixels for the display, defaults to 128

Adafruit_SSD1306_I2C Class

- Constructor
  - Create a SSD1306 I2C transport display driver instance with the specified RST pin, the I2C address, as well as the display dimensions.

```
Adafruit_SSD1306_I2c (I2C &i2c,  PinName RST,
                      uint8_t i2cAddress = SSD_I2C_ADDRESS,
                      uint8_t rawHeight = 32, uint8_t rawWidth = 128)
)
```

I2C slave address: **SSD_I2C_ADDRESS**
RST: reset pin (active low)
rawHight: the vertical number of pixels for the display, defaults to 32
rawWidth: the horizontal number of pixels for the display, defaults to 128

Member Functions

void **clearDisplay** (void)

Clear the display buffer.

void **display** ()

Cause the display to be updated with the buffer content.

void **splash** ()

Fill the buffer with the AdaFruit splash screen.

Member Functions

virtual void **drawPixel** (int16_t x, int16_t y, uint16_t color)

Paint one **BLACK** or **WHITE** pixel in the display buffer.

virtual void **drawFastHLine** (int16_t x, int16_t y, int16_t w, uint16_t color)

Draw a Horizontal Line.
Note: GFX_WANT_ABSTRACTS must be defined in Adafruit_GFX_config.h

virtual void **drawFastVLine** (int16_t x, int16_t y, int16_t h, uint16_t color)

Draw a vertical line.
Note: GFX_WANT_ABSTRACTS or GFX_SIZEABLE_TEXT must be defined in Adafruit_GFX_config.h

virtual void **drawLine** (int16_t x0, int16_t y0, int16_t x1, int16_t y1, uint16_t color)

Draw a line.
Note: GFX_WANT_ABSTRACTS or GFX_SIZEABLE_TEXT must be defined in Adafruit_GFX_config.h

Member Functions

void **drawRect** (int16_t x, int16_t y, int16_t w, int16_t h, uint16_t color)

Draw a rectangle.
Note: GFX_WANT_ABSTRACTS must be defined in Adafruit_GFX_config.h

void **fillRect** (int16_t x, int16_t y, int16_t w, int16_t h, uint16_t color)

Draw and fill a rectangle.
Note: GFX_WANT_ABSTRACTS or GFX_SIZEABLE_TEXT must be defined in Adafruit_GFX_config.h

void **drawRoundRect** (int16_t x0, int16_t y0, int16_t w, int16_t h, int16_t radius, uint16_t color)

Draw a rounded rectangle.
Note: GFX_WANT_ABSTRACTS must be defined in Adafruit_GFX_config.h

void **fillRoundRect** (int16_t x0, int16_t y0, int16_t w, int16_t h, int16_t radius, uint16_t color)

Draw and fill a rounded rectangle.
Note: GFX_WANT_ABSTRACTS must be defined in Adafruit_GFX_config.h

Member Functions

void **drawTriangle** (int16_t x0, int16_t y0, int16_t x1, int16_t y1,
int16_t x2, int16_t y2, uint16_t color)

Draw a triangle.
Note: GFX_WANT_ABSTRACTS must be defined in Adafruit_GFX_config.h

void **fillTriangle** (int16_t x0, int16_t y0, int16_t x1, int16_t y1,
int16_t x2, int16_t y2, uint16_t color)

Draw and fill a triangle.
Note: GFX_WANT_ABSTRACTS must be defined in Adafruit_GFX_config.h

void **drawCircle** (int16_t x0, int16_t y0, int16_t r, uint16_t color)

Draw a circle.
Note: GFX_WANT_ABSTRACTS must be defined in Adafruit_GFX_config.h

void **fillCircle** (int16_t x0, int16_t y0, int16_t r, uint16_t color)

Draw and fill a circle.
Note: GFX_WANT_ABSTRACTS must be defined in Adafruit_GFX_config.h

Member Functions

void **drawBitmap** (int16_t x, int16_t y, const uint8_t *bitmap,
int16_t w, int16_t h, uint16_t color)

Draw a bitmap.
Note: GFX_WANT_ABSTRACTS must be defined in Adafruit_GFX_config.h

void **drawChar** (int16_t x, int16_t y, unsigned char c, uint16_t color,
uint16_t bg, uint8_t size)

Draw a text character at a specified pixel location.

size_t **writeChar** (uint8_t c)

Draw a text character at the text cursor location.

void **setTextCursor** (int16_t x, int16_t y)

Set the text cursor location, based on the size of the text.

Member Functions

int16_t **height** (void)

Get the height of the display in pixels.

int16_t **width** (void)

Get the width of the display in pixels.

void **fillScreen** (uint16_t color)

Fill the entire display.

Note: GFX_WANT_ABSTRACTS must be defined in Adafruit_GFX_config.h

void **setTextSize** (uint8_t s)

Set the size of the text to be drawn.

Note: Make sure to enable either GFX_SIZEABLE_TEXT or GFX_WANT_ABSTRACTS

Member Functions

void **setTextColor** (uint16_t c)

Set the text foreground and background colors to be the same.

void **setTextColor** (uint16_t c, uint16_t b)

Set the text foreground and background colors independently.

void **setTextWrap** (bool w)

Set text wrapping mode true or false..

void **setRotation** (uint8_t r)

Set the display rotation, 0, 1, 2, or 3.

uint8_t **getRotation** (void)

Get the current rotation.

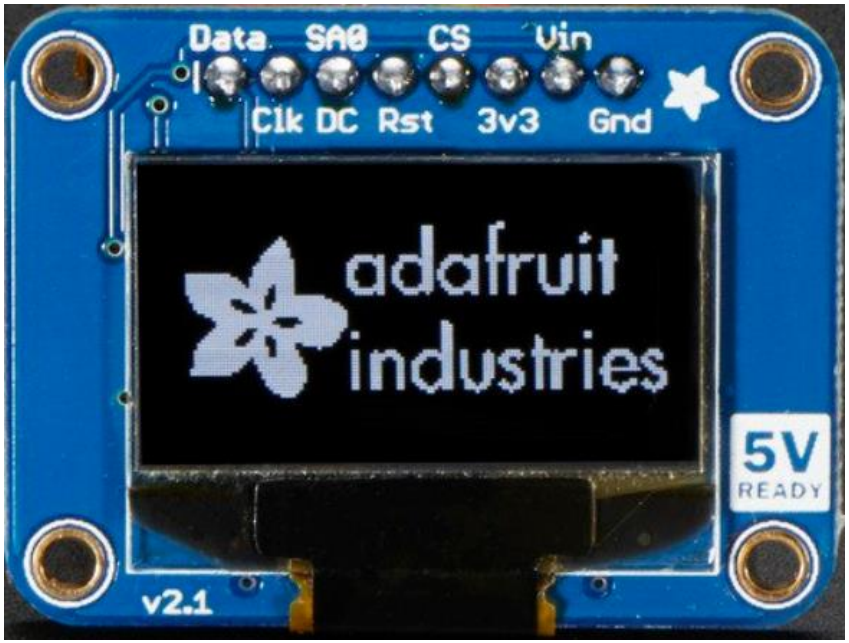
Labs

Lab12-3

- 실습 목적
  - OLED의 동작 방식을 이해할 수 있다.
  - 출력하기 원하는 데이터를 OLED 화면에 display할 수 있다.
- 실습 시나리오
  - OLED 첫째 라인에 " Handong Univ. CSEE"이란 문자를 표시한다.
  - OLED 5번째 라인에 온도를 다음과 같이 표시한다.
    - "Temperature: xx.x C"
  - OLED 7번째 라인에 습도를 다음과 같이 표시한다.
    - "Humidity: xx.x %"
  - 여기에서 온/습도는 random한 값으로 표시한다.

Lab12-3

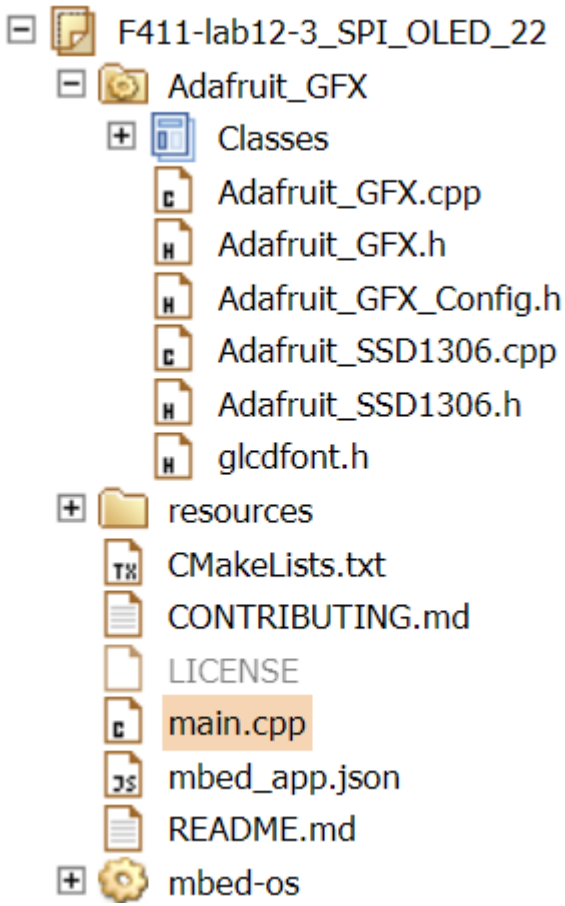
- 회로 구성



Connections for SPI		
OLED	**←→**	**Nucleo**
GND	←→	GND
Vin	←→	5V
DATA	←→	MOSI (D11)
CLK	←→	SCK (D13)
CS	←→	CS (D10)
D/C#	←→	D4
RST	←→	D7

Lab12-3

- Project structure

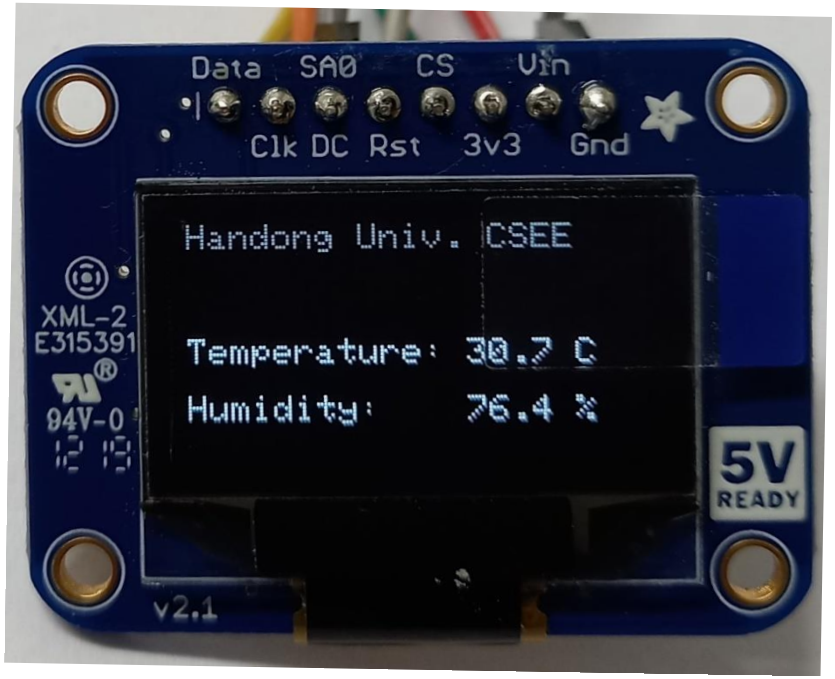


Lab12-3

```
#include "mbed.h"
#include "Adafruit_SSD1306.h"

// a class to pre-initialize an SPI
class SPIPreInit : public SPI
{
public:
    SPIPreInit(PinName mosi, PinName miso, PinName clk) : SPI(mosi,miso,clk)
    {
        format(8,3);
        frequency(2000000);
    };
};

SPIPreInit gSPI(ARDUINO_UNO_D11, NC, ARDUINO_UNO_D13);

//Adafruit_SSD1306_Spi(SPI &spi, PinName DC, PinName RST, PinName CS, uint8_t rawHieght = 32, uint8_t rawWidth = 128)
Adafruit_SSD1306_Spi gOLED(gSPI, ARDUINO_UNO_D4, ARDUINO_UNO_D7, ARDUINO_UNO_D10, 64);

int main()
{
    int tmp;

    ThisThread::sleep_for(5000ms);
    gOLED.clearDisplay();

    gOLED.printf("Handong Univ. CSEE\r\n\n\n\n");
    gOLED.printf("Temperature:\r\n\n");
    gOLED.printf("Humidity:");

    while(1)
    {
        gOLED.setTextCursor(13*6, 4*8);
        tmp = rand() % 1000;
        gOLED.printf("%4.1f C", tmp/10.0);

        gOLED.setTextCursor(13*6, 6*8);
        tmp = rand() % 1000;
        gOLED.printf("%4.1f %%", tmp/10.0);

        gOLED.display();
        ThisThread::sleep_for(5000ms);
    }
}
```

Lab12-3

- A sample result



Lab12-4

- Adafruit_SSD1306_Spi Class를 다음과 같은 형태로 동작하도록 관련 library를 수정한다.
  - Main program에서 별도의 class를 마들 필요가 없게 하기 위하여
- OLED 첫번째 라인에 자신의 이름을 표시하되, 폰트 크기를 2배가 되게 한다. (한 라인에 허용 글자 수에 따라 적절하게 표시)
- HDT21D 센서를 이용하여 3초 간격으로 측정한 온도와 습도를 표시.
- 출력 예:.

Lab12-4

- Adafruit_SSD1306_Spi Class
  - Constructor
    - Create a SSD1306 SPI transport display driver instance with the specified DC, RST, and CS pins, as well as the display dimensions.

```
Adafruit_SSD1306_Spi ( PinName MOSI, PinName CLK, PinName CS,
                       PinName DC, PinName RST,
                       uint8_t rawHeight = 32, uint8_t rawWidth = 128
)
```

Parameters of SPI: **MOSI**, **CLK**, **CS**
DC: data/command pin (Data/Command#)
RST: reset pin (active low)
rawHight: the vertical number of pixels for the display, defaults to 32
rawWidth: the horizontal number of pixels for the display, defaults to 128