



IoT Device! mbed로 쉽게 제어하자

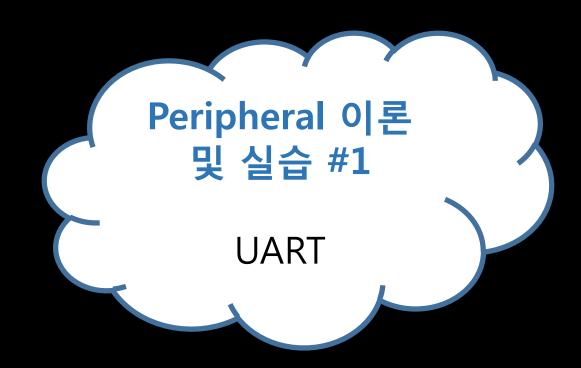




목차

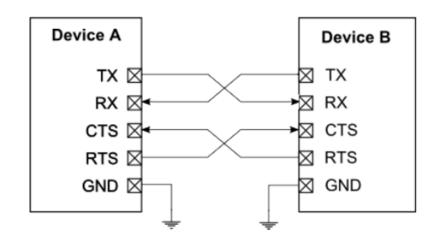


- UART
- I2C
- SPI
- ADC
- 응용 실습



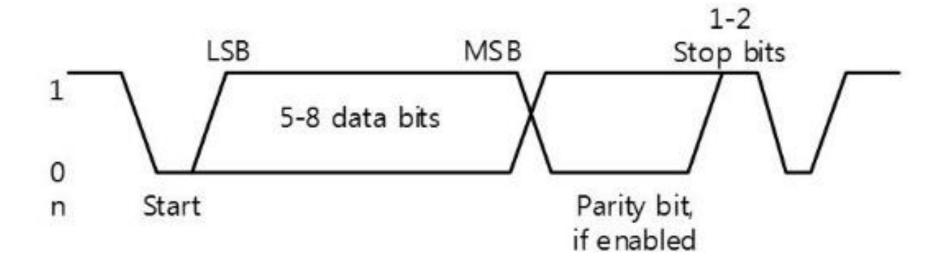


- UART(Universal Asynchronous Receiver/Transmitter)
- 1) UART는 병렬 데이터의 형태를 직렬 방식으로 전환하여 데이터를 전송한다.
- 2) 비동기식 통신이므로 sender(데이터를 보내는 기기)와 receiver(데이터를 받는 기기)의 Baud rate는 같아야 한다. (9600bps = 9600bit/sec)





UART(Universal Asynchronous Receiver/Transmitter)





Serial UART Functions in mbed Library

Constructor & Destructor Documentation

```
Serial ( PinName tx,
PinName rx,
const char * name = NULL
)

Create a Serial port, connected to the specified transmit and receive pins.

Parameters:
tx Transmit pin
rx Receive pin

Note:
Either tx or rx may be specified as NC if unused
```

void baud (int baudrate) [inherited]

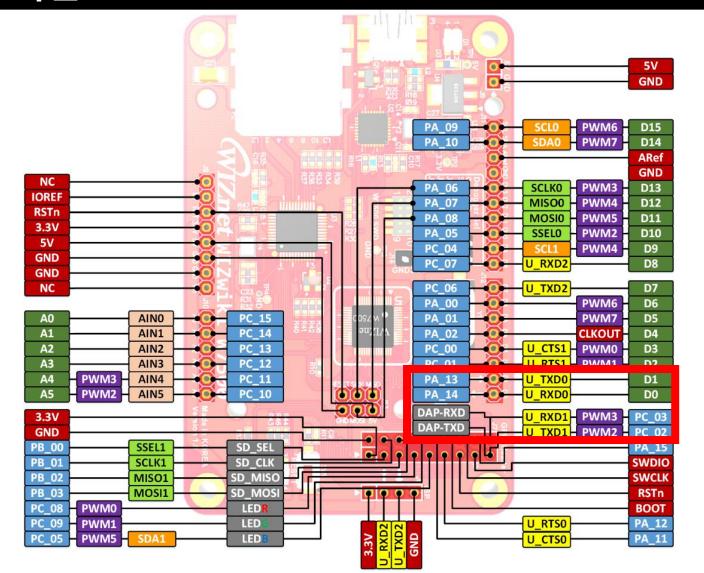
Set the baud rate of the serial port.

Parameters:

baudrate The baudrate of the serial port (default = 9600).

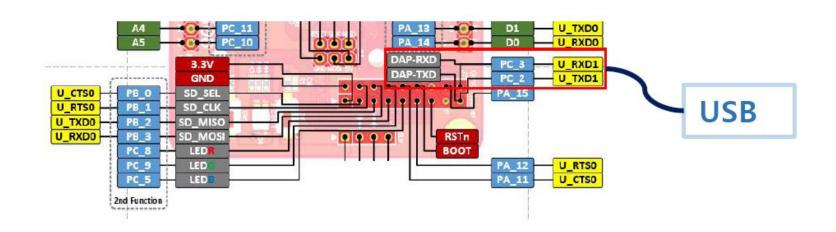








- UART(Universal Asynchronous Receiver/Transmitter)
- 3) W7500은 UARTO, UART1, UART2를 제공한다.
- 4) UART1은 W7500의 USB 케이블로 입출력이 가능하다.
- 5) Sender와 receiver의 RX와 TX가 교차하여서 연결 된다는 것을 유의한다.



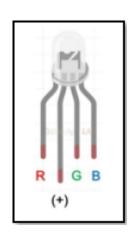


▶▶ 실습 개요

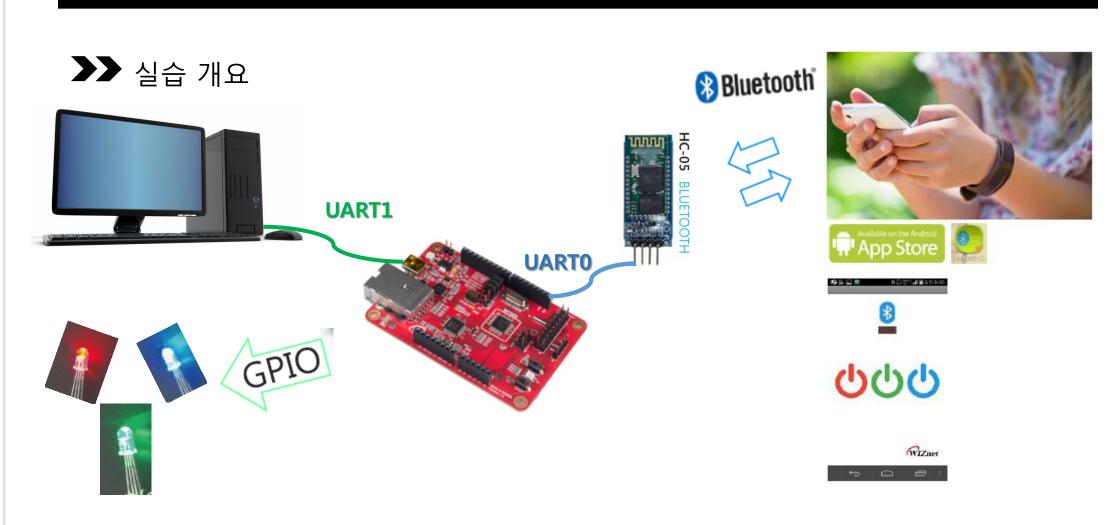
• 안드로이드 App을 이용한 Serial to Bluetooth(UART 이용 디바이스)로 3 colors LED를 제어한다.



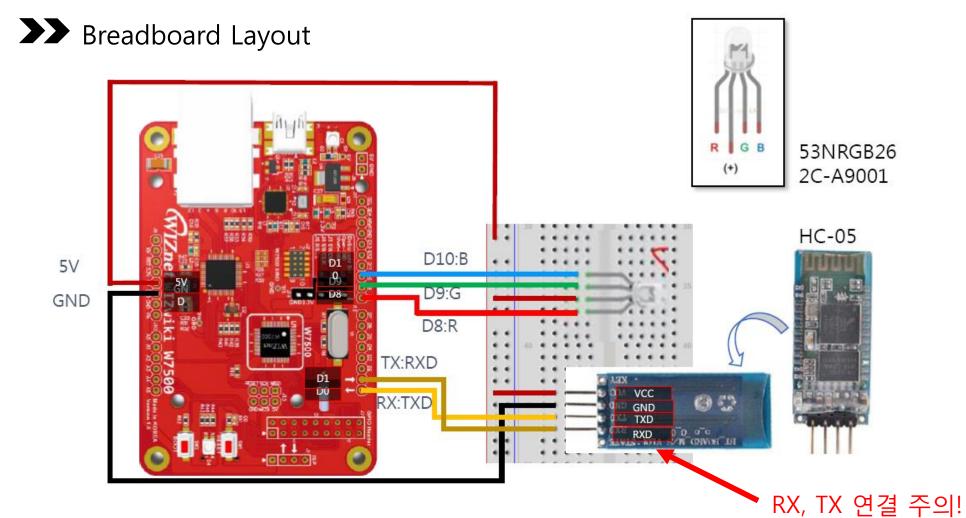














Bluetooth Setting

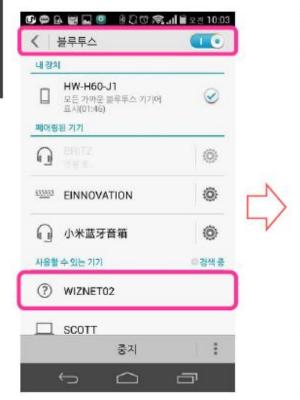
- https://developer.mbed.org/users/joon874/code/Bluetooth_HC-05_ATmode_DEMO/
- AT 커맨드를 통해서 Bluetooth name과 password를 세팅 (at+name=yourname, at+pswd=yourpsword, at+reset)

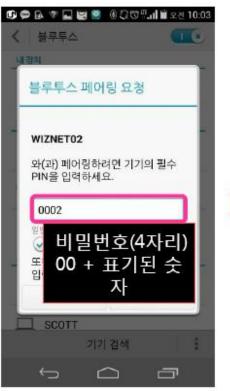




▲ 스마트폰 Bluetooth 설정

안드로이드 폰에서 블루투스를 찾아 연결



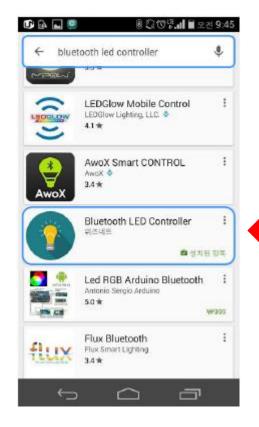






▶▶ 스마트폰 어플리케이션 다운로드 (Android)

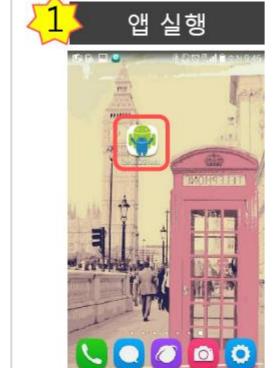
1) Play 스토어에서 "Bluetooth LED Controller" 검색

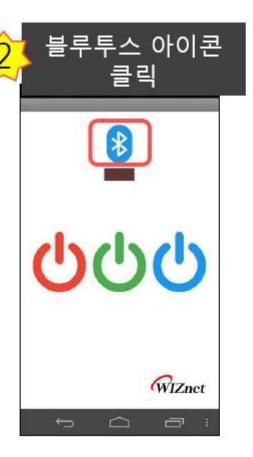


2) 다운로드 및 설치

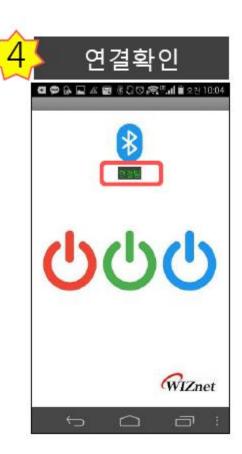


▶▶ 어플리케이션 실행 및 Bluetooth 연결











▶ 실습

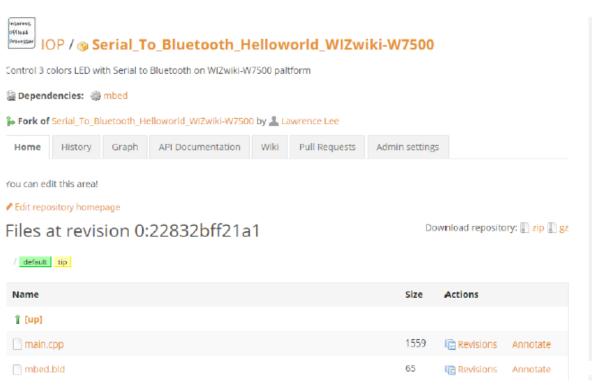








https://developer.mbed.org/teams/IOP/code/Serial_To_Bluetooth_Helloworld_WIZwiki-W/









실습 – UART Example Code 설명

```
1 #include "mbed.h"
                                                             LED 컨트롤을 위해 D8,9,10 핀을 Digital Output 으
                                                              로 선언. 초기값은 '1'
3 /* Digital Out Pin Configuration */
4 DigitalOut RED(D8,1);
5 DigitalOut GREEN(D9,1);
6 DigitalOut BLUE(D10,1);
                                                             UARTO 와 UART1 선언
8 /* UART Pin Configuration */
9 Serial pc(USBTX, USBRX);
                                                             UARTO 과 UART1 의 baudrate
10 Serial bt(D1,D0);
11
                                                             를 설정
12
13 int main (void)
14 {
15
     /* baudrate configuration */
                                                             가장 처음 Serial Terminal에 찍히는 문자열
16
     pc.baud(115200);
17
     bt.baud(9600);
     pc.printf("WIZwiki-W7500 BT\n\r");
                                                             'ch' 는 블루투스로 부터 받아오는 문자 저장 변수
21
22
     char ch;
                                                             'msg' 는 PC 에서 블루투스로 보내는 문자열 저장 버퍼
     char msg[256];
```



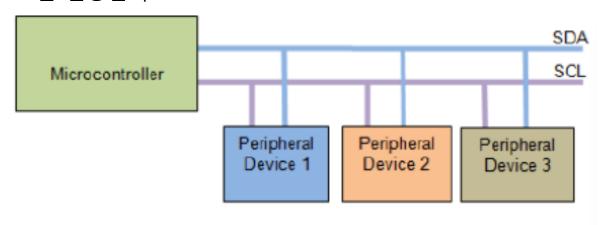
실습 – UART Example Code 설명

```
24
      while(1)
25
26
         /* WIZwiki-W7500 to Bluetooth */
         if(pc.readable())
                                                                    UART1에서 입력 받은 문자열을 UART0 로 출력
29
            pc.scanf("%s", &msg);
30
            bt.printf("%s",msg);
31
32
                                                                    UARTO에서 문자를 읽어와서 UART1 로 출력
33
         /* Bluetooth to WIZwiki-W7500 */
34
         if(bt.readable())
35
36
            ch = bt.getc();
37
            pc.putc(ch);
38
                                                                    UART0에서 읽어온 문자가 'r' 인 경우
            /* Control 3 colors LED */
                                                                    RED LED ON/OFF
            if(ch == 'r'){
                RED = !RED;
                /* Notice RED LED condition to Bluethooth */
                                                                    UART0에서 읽어온 문자가 'g' 인 경우
                if(RED == 0) bt.printf("RED ON");
                else
                              bt.printf("RED OFF");
            }else if(ch == 'g'){
                                                                     GREEN LED ON/OFF
46
                GREEN = !GREEN;
47
                /* Notice GREEN LED condition to Bluethooth *.
48
                if(GREEN == 0) bt.printf("GREEN ON");
                                                                    UART0에서 읽어온 문자가 'b' 인 경우
                               bt.printf("GREEN OFF"):
49
                else
50
            }else if(ch == 'b'){
51
                BLUE = !BLUE;
                                                                     BLUE LED ON/OFF
52
                /* Notice BLUE LED condition to Bluethooth */
53
                if(BLUE == 0)
                               bt.printf("BLUE ON");
54
55
56
```





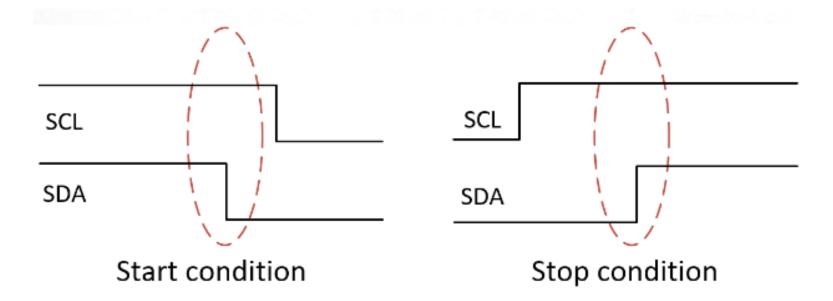
- I²C(Inter Integrated Circuit)
- 1) I²C는 필립스에서 개발한 직렬 컴퓨터 버스이며, 마더보드, 임베디드 시스템, 휴대전화 등에서 저속의 주변 기기를 연결하기 위해 사용된다.
- 2) I²C 는 master와 slave 간의 동기식 시리얼 데이터 동작 프로토콜이다.
- 3) I²C 는 SCL(Serial Clock)과 SDA(Serial Data), 2 개의 시리얼 버스를 사용한다.
- 4) 8-bit 마다 ACK/NACK 를 전송한다.





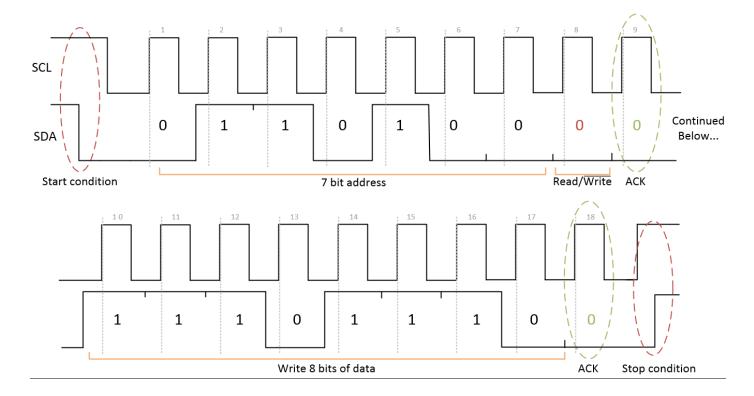
I²C(Inter Integrated Circuit)

start condition : SCL이 High 일때 SDA High-> Low stop condition : SCL이 High 일때 SDA Low -> High





- I²C(Inter Integrated Circuit)
- ex) addres가 0110100인 slave에게 11101110 data를 쓸때





>> mbed library I2C

```
int write (int
                           address,
            const char * data,
            int
                           length,
                           repeated = false
             bool
Write to an I2C slave.
Performs a complete write transaction. The bottom bit of the address is forced to 0 to indicate a write.
Parameters:
        address 8-bit I2C slave address [ addr | 0 ]
        data
                  Pointer to the byte-array data to send
        length Number of bytes to send
        repeated Repeated start, true - do not send stop at end
Returns:
       0 on success (ack), non-0 on failure (nack)
Definition at line 54 of file I2C.cpp.
```

void start (void)

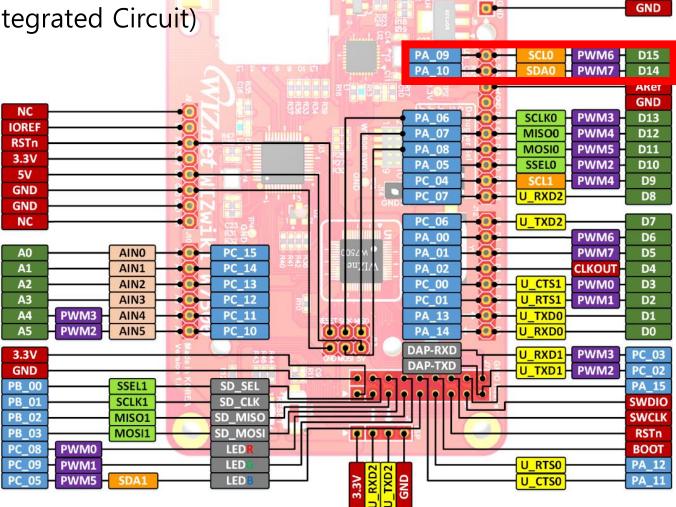
Creates a start condition on the **I2C** bus. Definition at line 85 of file <u>I2C.cpp</u>.

void stop (void)

Creates a stop condition on the **I2C** bus. Definition at line 89 of file <u>I2C.cpp</u>.



I²C(Inter Integrated Circuit)





실습 개요

- I^2C 를 이용하여 OLED에 원하는 문자열과 1초마다 증가하는 변수 값을 출력한다.
- OLED: mbed에 등록된 LCD components 라이브러리 참조 (https://developer.mbed.org/components/Adafruit-OLED-128x32/)







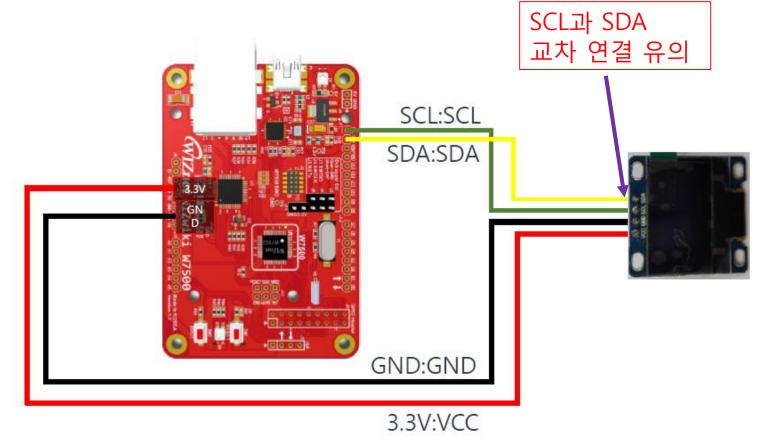








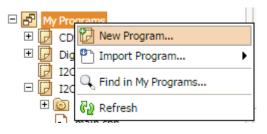
>> Breadboard Layout

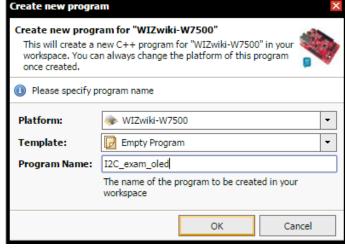


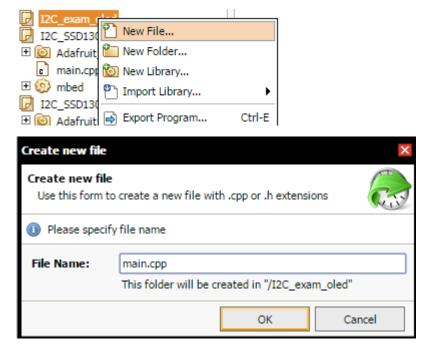


프로젝트 생성

- 현재 디렉토리 (우클릭) -> New Program.. -> Empty Program 체크 -> 프로젝트 이름 설정 -> OK 클릭
- 현재 프로젝트 (우클릭) -> New File... -> main.cpp 생성



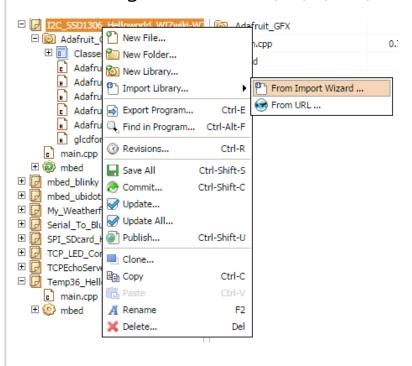


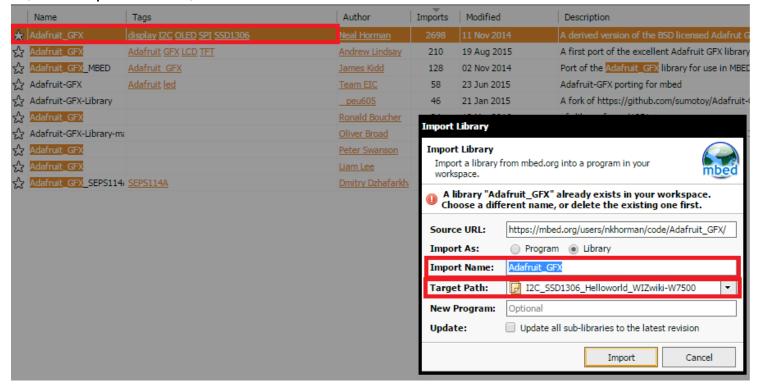




Library Import

- 현재 디렉토리(폴더) 우클릭 -> Import Library... -> From Import Wizard ... 클릭
- Target Path를 현재 디렉토리로 설정하고 "Import" 클릭









```
#include = 사용해 mbed.h, adafruit_SSD1306.h"

#include = 사용해 mbed.h, adafruit_SSD1306.h 를 링크

property of the p
```



▶▶ 실습

```
11 int main()
12 {
     //i2c.frequency(100000);
13
      //i2c.start();
14
15
16
      uint16_t x=0;
17
18
      myOled.begin();
      myOled.printf("%ux%u \nHellow World\r\n", myOled.width(), myOled.height()); ---
20
      myOled.display();
21
22
23
24
25
26
27
28
29
```

- begin() 함수로 OLED 초기 세팅
- Serial을 통해 문자열을 OLED 로 전 송
- display() 함수로 OLED 출력



▶▶ 실습

30 }

```
11 int main()
12 {
      //i2c.frequency(100000);
13
14
      //i2c.start();
15
16
      uint16 t x=0;
17
18
      myOled.begin();
      myOled.printf("%ux%u \nHellow World\r\n", myOled.width(), myOled.height());
20
       myOled.display();
21
22
       while (1)
23
24
           myled_R = !myled_R;
25
26
                       Insert code
27
28
          wait(1.0);
29
```

- begin() 함수로 OLED 초기 세팅
- Serial을 통해 문자열을 OLED 로 전 송
- display() 함수로 OLED 출력
- 동작확인을 위한 while()문
- 1초마다 RED LED 가 점멸, 'x' 값이 증가하며 OLED 에 출력





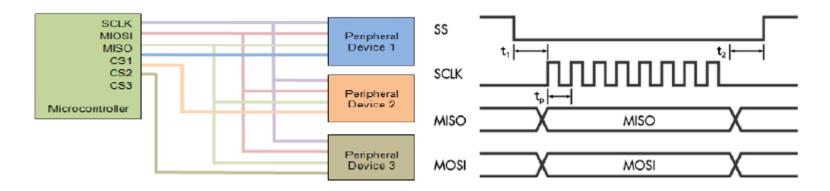
SPI 기초 이론

- SPI(Serial Peripheral Interface)
- 1) SPI는 고속 동기식 직렬 데이터 동작 프로토콜이다.
- 2) I^2 C와 마찬가지로 Master Slave 방식으로 동작하며, Master가 동기를 위한 클럭을 출력한다.
- 3) Master에서 Slave를 선택하고 양방향 또는 단방향 전송을 한다.



SPI 기초 이론

- SPI(Serial Peripheral Interface Bus)
- 4) SPI는 4개의 시리얼 버스를 사용한다.
 - -SCLK(clock)
 - -CS(slave Chip Select)
 - -MOSI(Master Output data Slave Input data)
 - -MISO(Master Input data Slave Output data)
- 5) 1 vs N 통신이 가능하며, 한 개 이상의 slave 디바이스를 사용하는 경우, Master는 CS 신호를 이용하여 디바이스와 통신한다.





SPI 기초 이론

SPI(Serial Peripheral Interface Bus)

Constructor & Destructor Documentation

```
SPI ( PinName mosi,
PinName miso,
PinName sclk,
PinName ssel = NC
)

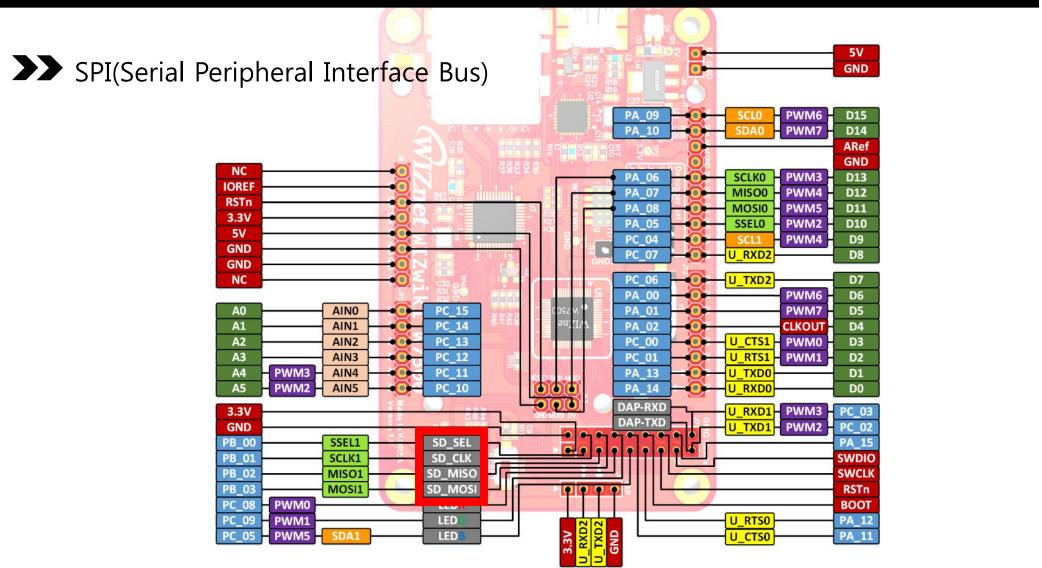
Create a SPI master connected to the specified pins.
mosi or miso can be specfied as NC if not used

Parameters:
mosi SPI Master Out, Slave In pin
miso SPI Master In, Slave Out pin
sclk SPI Clock pin
ssel SPI chip select pin
```

```
1 // Send a byte to a SPI slave, and record the response
 3 #include "mbed.h"
 5 // hardware ssel (where applicable)
 6 //SPI device(p5, p6, p7, p8); // mosi, miso, sclk, ssel
 8 // software ssel
9 <u>SPI</u> device(p5, p6, p7); // mosi, miso, sclk
10 DigitalOut cs(p8); // ssel
11
12 int main() {
      // hardware ssel (where applicable)
      //int response = device.write(0xFF);
16
      // software ssel
      cs = 0;
       int response = device.write(0xFF);
19
       cs = 1:
20 }
```



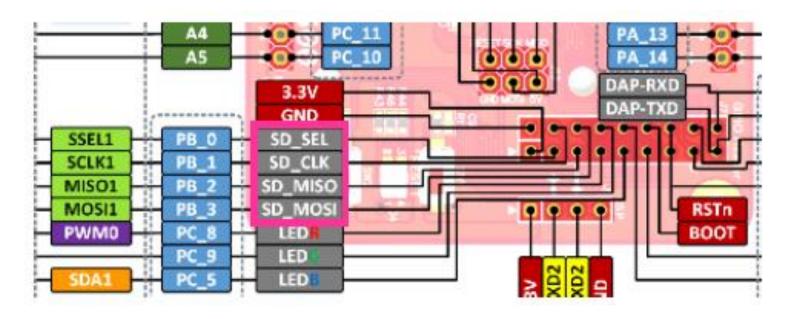
SPI 기초 이론





SPI 기초 이론

SPI(Serial Peripheral Interface Bus)



SPI Pinout

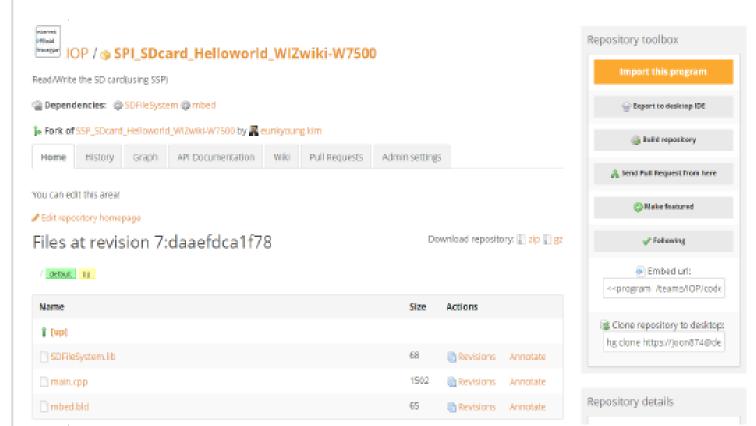


▶▶ 실습 개요

- 범용 SDcard library를 이용한다.
- 모니터를 통해서 문자열을 입력 받고, 입력 받은 문자열을 SPI 통신을 이용해 SDcard에 저장한다.
- 저장된 문자열을 모니터에 출력한다.



- Import SPI SDcard Example
- https://developer.mbed.org/teams/IOP/code/SPI_SDcard_Helloworld_WIZwiki-W7500/



Import this program 클릭!!





- #include 를 이용해 mbed.h 링크
- SDFileSystem.h 링크

타겟보드가 WIZwiki-W7500 일때, PB_3, PB_2, PB_1, PB_0 을 각각 MOSI, MISO, CLK, SEL 으로 선언

Cool Components workshop board

위에 정의한 SPI 핀들로 파라미터를 채운다



SPI SDcard Example Code 설명

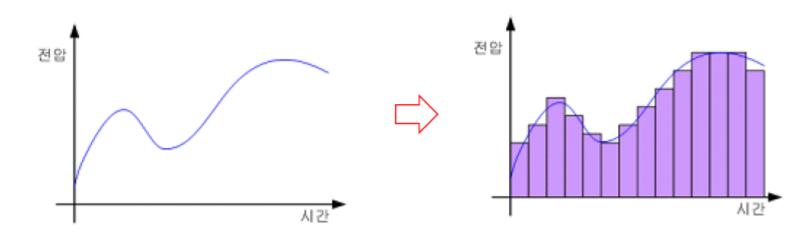
```
4 int main() {
                                                                             SDcard 안에 mydir 폴더 생성
         char str[MAX SIZE];
         mkdir("/sd/mydir", 0777);
         //"w" : {Write} - Create an empty file for output operations. If a file with the same name already exists, its c
         //"a+" : (append/update) - open (or create if the file does not exist) for update at the end of the file
         FILE *fp = fopen("/sd/mydir/sdtest.txt", "w");
         if(fp == NULL) {
             error("Could not open file for write\n");
                                                                                폴더에 sdtest.txt 파일을 write 모드로 생성
         //fprintf(fp, "Hello fun SD Card World!");
                                                                                "w" 일 경우, 지정된 경로에 새로운 파일을 생성
         printf("Insert data:");
         scanf("%s",str);
                                                                                "a+" 일 경우, 기존의 파일에 추가로 입력
         printf("\r\n");
                                                                                파일생성에 실패하면 Err 메시지 출력
         fflush(stdin);
         fprintf(fp, "%s", str);
         fclose(fp);
         printf("Reading from SD card...\r\n");
         fp = fopen("/sd/mydir/sdtest.txt", "r");
                                                                             입력되는 문자들을 문자열로 파일에 쓴다
         if (fp != NULL) {
              fgets(str, MAX_SIZE, fp);
             printf("%s", str);
             printf("\r\n");
            fclose(fp);
                                                                              SDcard에 만들어진 sdtest.txt 에서 정의한 string 사이
즈만큼의 문자열을 모니터로 출력하고 파일을 닫는다.
         } else {
             printf("failed!\n");
```





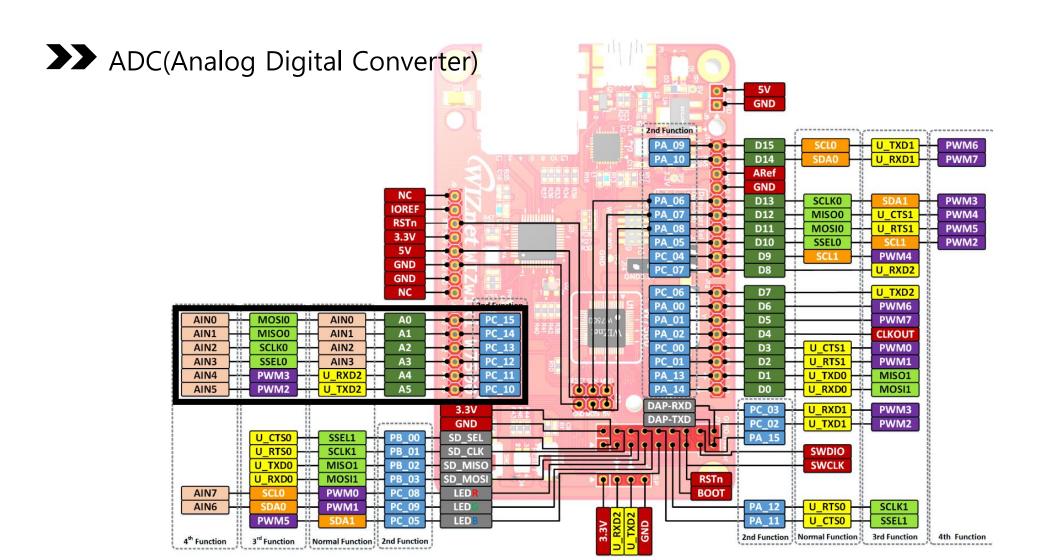
ADC 기초 이론

- ADC(Analog Digital Converter)
- 1) ADC는 아날로그 신호를 디지털 신호로 바꿔주는 변환기이다.
- 2) ADC는 입력되는 전압/전류 변화를 샘플링 하여 아날로그 신호를 측정한다.
- 3) WIZwiki-W7500은 Analog input으로 12-bit를 사용하여 0 ~ 2¹²-1 사이의 Digital data를 얻는다.
- 4) WIZwiki-W7500은 ADC Pin Out. 6개의 ADC Channel 중에서 선택해 사용할 수 있다





ADC 기초 이론





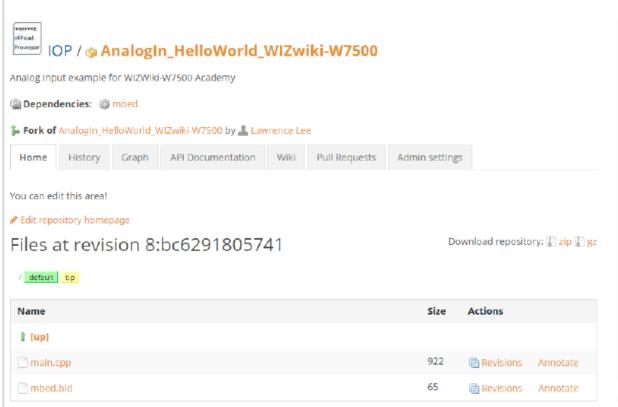
▶▶ 실습 개요

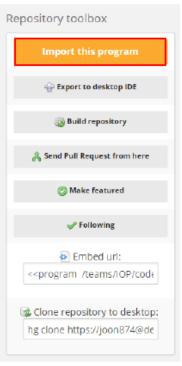
- 가변저항에 따라서 변하는 아날로그 값을 입력 받는다.
- 아날로그 값에 따라서 LED를 제어한다.
- 아날로그 값은 1초마다 시리얼 모니터로 확인 가능하다.



Import ADC Input Example

https://developer.mbed.org/teams/IOP/code/AnalogIn_HelloWorld_WIZwiki-W7500/







Analog Input Example Code 설명

```
#include 를 이용해 mbed.h 링크
 3 #include "mbed.h"
 5 // Initialize a pins to perform analog input and digital output fucntions
 6 AnalogIn ain(A0);
                                                                             A0 를 Analog Input 으로 선언
 7 DigitalOut myled_R(LED_RED);
                                      // LED RED on WIZwiki-W7500
                                                                             RED LED 를 Digital Output 으로 선언
9 int main(void)
10 {
      int ain val = 0;
12
13
     while (1) {
14
                                                                              값을 조건과 비교해서
15
         ain val = ain.read()*1000;
16
                                                                              조건보다 크면 Red LED On
         // compare with specific value
17
18
         if(ain val > 500) myled R = 1;
                                         // Red LED OFF
19
         else myled R = 0;
                                       // Red LED ON
                                                                              그렇지 않으면 Red LED OFF
         // output the voltage and analog values
21
         printf("======\r\n");
22
         printf("voltage value : %3.3f\r\n", ain.read()*3.3f);
23
                                                                             read() 함수로 아날로그 값을 읽어와서
24
         printf("analog value : %3.3f\r\n", ain.read());
                                                                              1초마다 모니터에 출력
25
         printf("analog value x1000 : %d\r\n",ain_val);
26
         wait(1.0);
27
28 }
```



▶▶ 실습 준비물



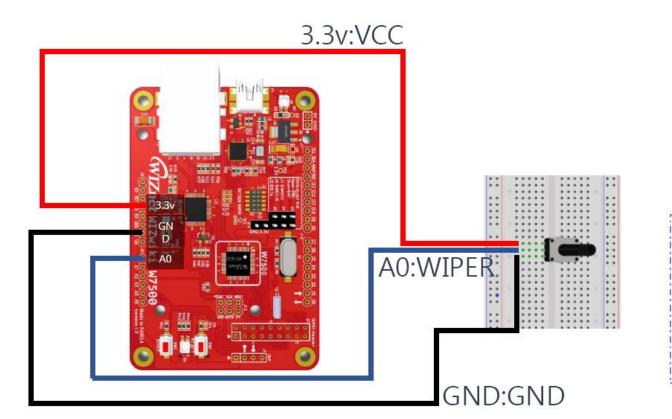




GF063PK B103



>> Breadboard Layout

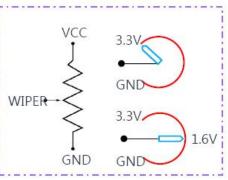


voltage value : 1.928
analog value : 0.605

voltage value : 2.024
analog value : 0.603

voltage value : 1.056
analog value : 0.281

voltage value : 0.000
analog value : 0.000







▶▶ 실습 개요

- 1) 조도센서의 아날로그 값을 읽어온다.
- 2) 밝을 때는 BLUE LED가 켜지고, 어두울 때는 RED LED가 켜진다.
- 3) 센서에서 측정한 값을 모니터에 출력한다.

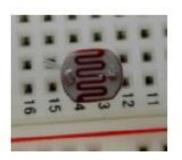
Voltage Input = analog.read() * 3.3v









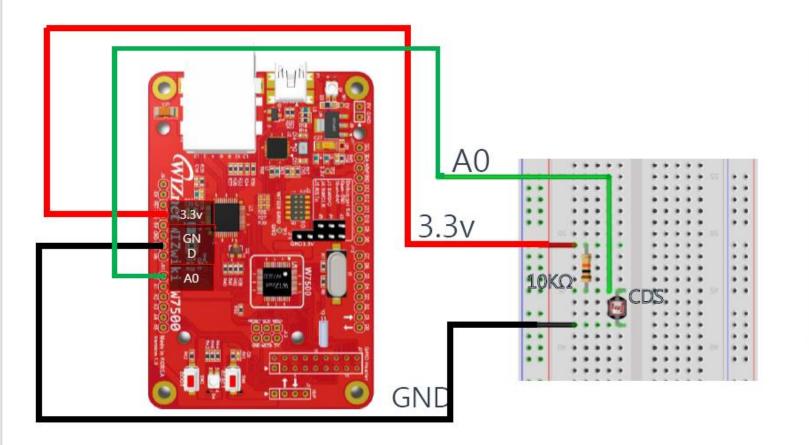


PGM5537D





Breadboard Layout



CDS Data: 343 CDS Voltage : 1.152V

CDS Data : 338

CDS Voltage : 1.077V

CDS Data: 341

CDS Voltage : 1.125V

CDS Data: 602

CDS Voltage : 2.201V

CDS Data: 696

CDS Voltage : 2.296V





실습 코드

```
3 DigitalOut myled R(LED RED);
                              // LED RED on WIZwiki-W7500
4 DigitalOut myled B(LED BLUE);
                              // LED BLUE on WIZwiki-W7500
                                                                  A0 을 Analog Input 으로 선언
5 AnalogIn CDS(A0);
                           // Analog Input Pin A0
                                                                  RED LED 를 Digital Output 으로 선언
                                                                  BLUE LED 를 Digital Output 으로 선언
 7 int main() {
     printf("Hello WizWIki-W7500!\r\n");
     printf("======\r\n");
11
     int CDS data = 0;
                                                                  CDS.read() 로 아날로그 값을 읽어온다
     double CDS vol = 0;
                                                                  *1000 은 3자리 정수로 나타내기 위함
     while(1) {
15
         CDS data = CDS.read()*1000;
                                                                  *3.3 은 0v~3.3v 사이 값으로 나타내기 위함
         CDS vol = CDS.read()*3.3;
         //CDS Seneor ADC Low Data
         printf("CDS Data : %3d\r\n",CDS data);
20
         //CDS Sensor Voltage data
                                                                  1초마다 아날로그 값을 모니터에 출력
         printf("CDS Voltage : %3.31fV\r\n",CDS vol);
22
         printf("=====\r\n");
         wait(1.0);
         //Status is Dark and RED LED ON, BLUE LED OFF
         if(CDS data < 500) {
            myled R = 1;
                                                                  아날로그 값을 정해놓은 값과 비교해 RED LED 와 BLUE
            myled B = 0;
                                                                  LED 를 켜고 끈다
         //Status is Bright and BLUE LED ON, RED LED OFF
         else {
            myled R = 0;
            myled B = 1;
```





https://developer.mbed.org/teams/IOP/code/CDS_HelloWorld_WIZwiki-W7500/



▶▶ 실습 개요

- 1) 센서의 아날로그 값을 읽어온다.
- 2) 읽어온 아날로그 값으로 섭씨와 화씨를 구해서 모니터에 출력한다.

Temp in
$$^{\circ}$$
C = [(Voltage Input) - 0.5] x 100
Temp in $^{\circ}$ F = [(Temp in $^{\circ}$ C x 9) / 5] + 32.0







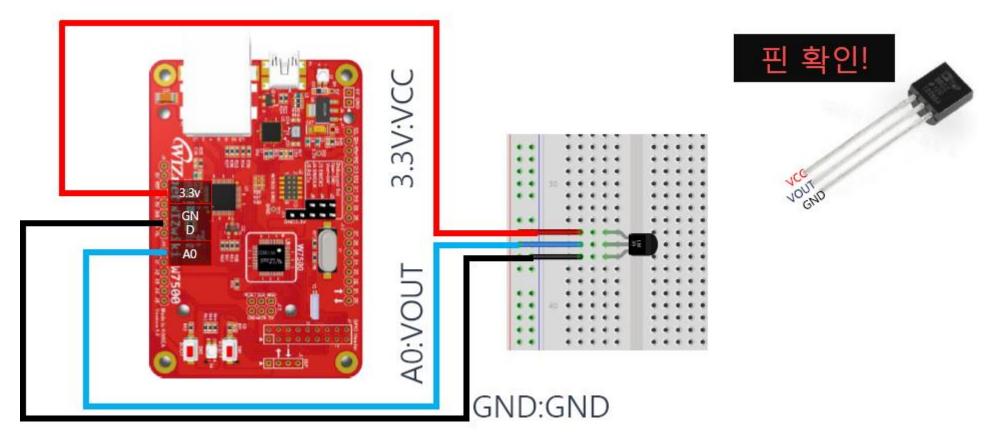




TMP36GT9



Breadboard Layout





▶ 실습

```
1 /* Analog Input "Temp36" Temperature Sensor Example Program */
 3 #include "mbed.h"
 5 // Initialize a pins to perform analog input fucntions
 6 AnalogIn ain(A0); // connect A0(WIZwiki-W7500) to Vout(Temp36)
 8 int main(void)
      while (1)
10
11
12
                                                 // connect Vs(Temp36) to 3.3V(WIZwiki-W7500)
13
                     Insert Code
14
                                                 // calculate temperature C
15
                                                 // calculate temperature F
16
17
18
          printf("tempC value : %5.2f C \r\n", tempC);
19
          printf("tempF value : %5.2f F \r\n", tempF);
20
21
          wait(1.0);
22
23 }
```



Thank you