강의명: 임베디드 시스템
숙제 번호: 4
숙제 제목: Serial communication(직렬 통신)
학생 이름: 한규현

1. 프로그램 uart-tx/uart-rx

1.1 프로그램 코드 쓰기
uart-tx

```
#include "mbed.h"
Serial uart_tx(D1, D0); // uart_tx = (D1=TX, D0=RX)
DigitalIn sw(D2, PullDown); // sw = D2(PullDown)
int main()
{
    while(true) {
        if(sw ==1) { // if sw is on
            uart_tx.putc('1'); // send '1'
            thread_sleep_for(200); // wait 200 ms
        }
    }
}
```

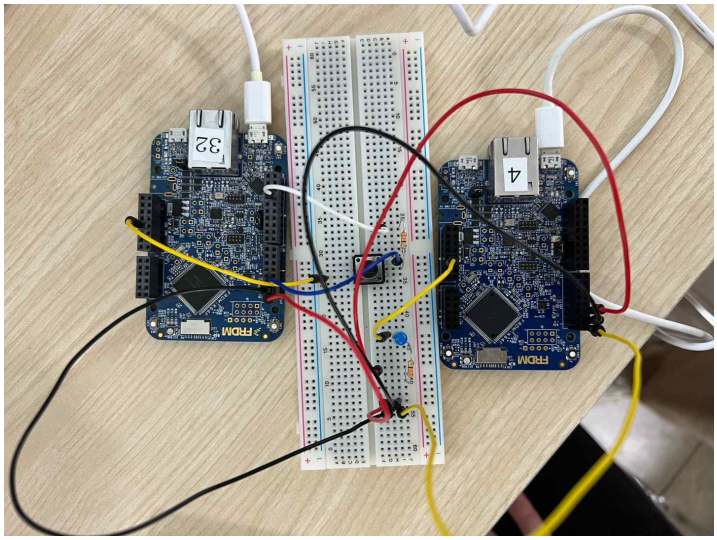
urat-rx

```
#include "mbed.h"
Serial uart_rx(D1, D0); // uart_rx = (D1=TX, D0=RX)
DigitalOut led(D2); // led = D2
int main()
{
    while(true) {
        if(uart_rx.getc() =='1') { // if getc() is '1'
            led =1; // on LED
            thread_sleep_for(200); // wait 200 ms
            led =0; // off LED
        }
    }
}
```

1.2 프로그램 작성 아이디어 혹은 이유 설명 쓰기
D0, D1 pin을 UART RX 및 TX로 연결하기 위해 Serial uart_tx(D1, D0);, Serial

uart_rx(D1, D);로 코드를 작성한다. 32번 보드의 D2 pin에는 버튼 스위치를, 4번 보드의 D2 pin에는 LED를 연결했다. 이를 32번 보드의 버튼 스위치를 누르면 4번 보드의 LED가 빛나도록 각각 DigitalIn sw(D2, PullDown);, DigitalOut led(D2); 작성한다. tx부분에서 스위치가 눌리는 경우 if(sw==1)로 처리하고, rx에 1을 보내도록 uart_tx.putc('1');로 작성했다. 이후 잠깐에 시간을 두기 위해 200ms의 wait를 주었다. rx부분에서는 urat_rx.getc() == '1'이라는 tx에서 보낸 1을 받으면 led = 1로 하여 200ms 후에 멈추도록 했다.

1.3 하드웨어 구성 사진 첨부하기



1.4 프로그램 수행 사진/동영상(Youtube 링크) 첨부하기
https://youtu.be/0v3SfoLvAxg

2. 프로그램 spi-master/spi-slave

2.1 프로그램 코드 쓰기
spi-master

```
#include "mbed.h"
SPI spi(PTD2, PTD3, PTD1); // spi = (MOSI, MISO, SCLK)
DigitalOut ss(PTD0, 0x01); // ss = PTD0
DigitalIn sw(D0, PullDown); // sw = D0(PullDown)
int main()
{
    spi.format(8, 3); // 8-bit data, mode 3
    spi.frequency(1000000); // frequency 1,000,000 Hz
    while(true) {
        if(sw ==1) { // if sw is on
        ss =0; // enable slave
```

```
        spi.write('1'); // send '1' and receive
        ss =1; // disable slave
        thread_sleep_for(200); // wait 200 ms
        }
    }
}
```

spi-slave

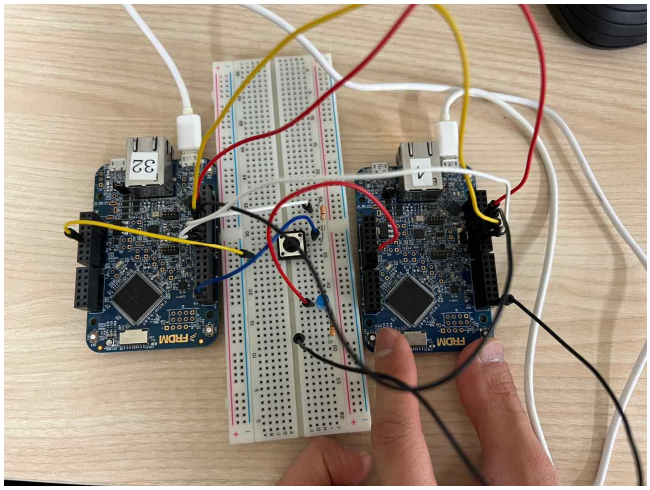
```
#include "mbed.h"
SPISlave spi_slave(PTD2, PTD3, PTD1, PTD0); // spi_slave = (MOSI, MISO,
SCLK, SS)
DigitalOut led(D0); // led = D0
int main()
{
    unsigned char r;
    spi_slave.format(8, 3); // 8-bit data, mode 3
    spi_slave.frequency(1000000); // frequency 1,000,000 Hz
    led =0;
    while(true) {
        if(spi_slave.receive()) { // if spi data is ready
            r =spi_slave.read(); // r = receive data
            spi_slave.reply(r); // and send r
            if(r =='1') { // if r == '1'
                led =1; // on LED
                thread_sleep_for(200); // wait for 200 ms
                led =0; // off LED
            }
        }
    }
}
```

2.2 프로그램 작성 아이디어 혹은 이유 설명 쓰기

spi-master에 SPI는 4개의 wire를 사용하고, master에는 MOSI는 PTD2, MISO는 PTD3, SCK는 PTD1을 설정했다. slave도 같은 방식으로 하되, SS에 PTD0를 설정했다. 추가적으로 Master에는 DigitalIn sw(D0,PullDown);을 설정했다. master, slave 모두 8-bit data, mdoe3, 주파수는 1,000,000으로 설정하기 위해 spi.format(8,3); spi.frequency(1000000); 프로그램 했다. 이후 while문으로 스위치 on/off 명령을 전달하도록 하는데, ss = 0이 되면 1을 전송해서 보내고, 이후 ss = 1로 바꾸고 wait 200ms로 했다. 이후 slave는 r = spi_slave.read();로 1이 보내지면 수신하고 1이 되면 200ms 후 led를 켜고 꺼지게 프로그램

을 작성했다.

2.3 하드웨어 구성 사진 첨부하기



2.4 프로그램 수행 사진/동영상(Youtube 링크) 첨부하기
https://youtu.be/Ev-zMl_oQio

3. 프로그램 i2c-master/i2c-slave

3.1 프로그램 코드 쓰기
ic2-master

```
#include "mbed.h"
SPI spi(PTD2, PTD3, PTD1); // spi = (MOSI, MISO, SCLK)
DigitalOut ss(PTD0, 0x01); // ss = PTD0
DigitalIn sw(D0, PullDown); // sw = D0(PullDown)
int main()
{
    spi.format(8, 3); // 8-bit data, mode 3
    spi.frequency(1000000); // frequency 1,000,000 Hz
    while(true) {
        if(sw ==1) { // if sw is on
        ss =0; // enable slave
        spi.write('1'); // send '1' and receive
        ss =1; // disable slave
        thread_sleep_for(200); // wait 200 ms
        }
    }
}
```

ic2-slave

```
#include "mbed.h"
I2CSlave i2c_slave(I2C_SDA, I2C_SCL); // i2c_slave = (I2C_SDA, I2C_SCL)
DigitalOut led(D0); // led = D0
const int addr =0xa0; // slave addr (even no)
int main()
{
    int i, r;
    led =0;
    i2c_slave.address(addr); // set slave address
    while(true) {
        i =i2c_slave.receive(); // check if my address is selected
        switch(i) {
        case I2CSlave::WriteAddressed: // WRITE means slave read
            i2c_slave.read(); // read address skips
            r =i2c_slave.read(); // r = read data
            if(r =='1') { // if r == '1'
                led =1; // on led
                thread_sleep_for(200); // wait 200 ms
                led =0; // off led
            }
            break;
        default:
            break;
        }
    }
}
```

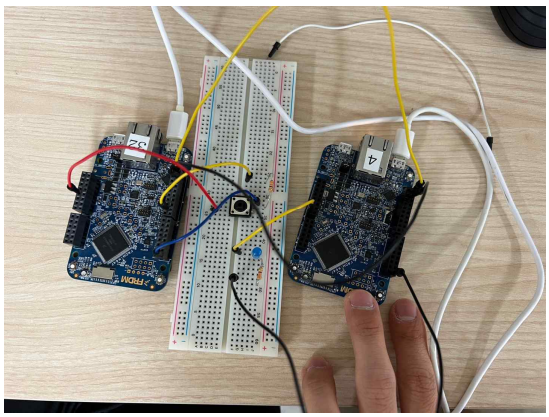
3.2 프로그램 작성 아이디어 혹은 이유 설명 쓰기

I2C 통신을 위해 master에는 I2C i2c(I2C_SDA, I2C_SCL); slave에는 I2CSlave i2c_slave(I2C_SDA, I2C_SCL); 로 프로그램 하였다. master 보드의 D0 pin에는 버튼 스위치 연결을 위해 DigitalIn sw(D0, PullDown);을 작성했고, slave 보드에는 D0 pin에 LED를 연결하여 빛이 날 수 있도록 DigitalOut led(D0);를 작성했다. 또한 const int addr = 0xa0; 로 slave의 주소를 담았다.

이후 앞선 문제들과 비슷한 방법으로 master에는 while문으로 sw == 1 즉 스위치가 켜지면 slave를 address로 지정하게 했다. 이후 1을 쓰고 작성 후 멈추고 200ms wait를 주었다. slave에는 반복문이 돌고 i2c_slave.receive();로 slave address가 맞으면 r = ic2_slave.read(); 로 스위치를 켜라는 명령인지 확인하고 맞은 경우 led = 1로 불을 키게 하

는 방법으로 프로그램을 하였다.

3.3 하드웨어 구성 사진 첨부하기



3.4 프로그램 수행 사진/동영상(Youtube 링크) 첨부하기
https://youtu.be/41rvlCPlvN4

끝.