

강의명: 임베디드 시스템

숙제 번호: 5

숙제 제목: Mbed application shield(Mbed 응용 실드)

학생 이름: 한규현(팀장), 전세환

1. 프로그램 app-shield-lcd

1.1 프로그램 코드 쓰기

```
#include "mbed.h"
#include "C12832.h"

C12832 lcd (D11 , D13 , D12 , D7 , D10 ); // lcd = (MOSI, SCK, RESET, A0, nCS

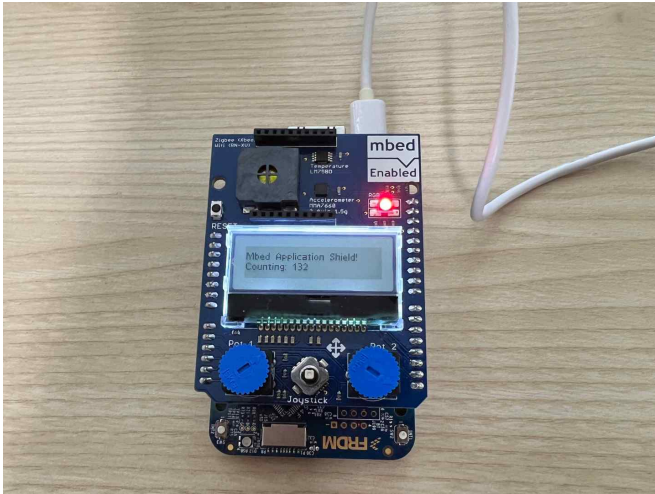
int main ()
{
    int count =0 ;

    lcd .cls ();
    lcd .locate (0 , 6 );// goto (0,
    lcd .printf ("Mbed Application Shield!"); // print message
    while (true ) {
        lcd .locate (0 , 16 );// goto (0,
        lcd .printf ("Counting: %d", count ++); // print count
        thread_sleep_for (1000 );
    }
}
```

1.2 프로그램 작성 아이디어 혹은 이유 설명 쓰기

C12832 lcd(D11,D13,D12,D7,D10)를 사용해 D11,D13,D12,D7,D10핀에 연결해준다. lcd.cls()를 사용하여 스크린을 초기화 해준다. lcd.locate()를 사용하여 0,6으로 커서를 이동시키고 lcd.printf를 사용하여 Mbed Application Shield!를 출력한다. 그후 반반복문을 사용하여 카운팅을 진행하고 0,16으로 커서를 이동시켜 Counting: 카운트를 보여주고 thread_sleep_for(1000)을 사용하여 1초간 기다린다.

1.3 하드웨어 구성 사진 첨부하기



1.4 프로그램 수행 사진/동영상(Youtube 링크) 첨부하기

https://youtu.be/3f9_up5VTyA

2. 프로그램 app-shield-rgb-led

2.1 프로그램 코드 쓰기

```
#include "mbed.h"
#include "C12832.h"

C12832 lcd (D11 , D13 , D12 , D7 , D10 ); // lcd = (MOSI, SCK, RESET, A0, nCS)
PwmOut led_r (D5 ); // led_r = (Red LED)
PwmOut led_g (D9 ); // led_g = (Red LED)
PwmOut led_b (D8 ); // led_b = (Red LED)

int main ()
{
    lcd .cls ();
    lcd .locate (0 , 6 );
    lcd .printf ("RGB LED!");
    while (true ) {
        for (float f =0.0 ; f <1.0 ; f +=0.05 ) {
            led_r =1.0 - f; // on LED Red
            led_g =1.0 ; // off LED Green
            led_b =1.0 ; // off LED Blue
            lcd .locate (0 , 16 );
```

```

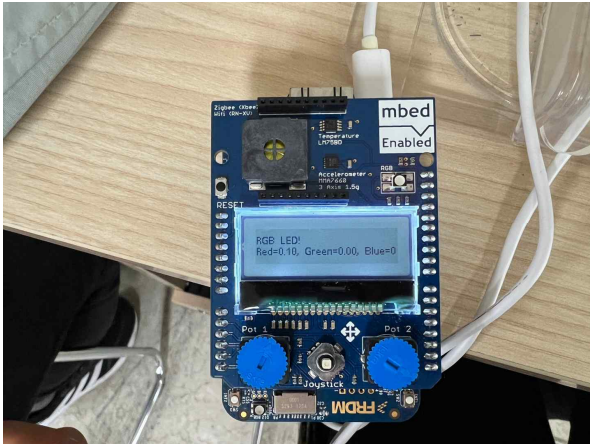
    lcd .printf ("Red=%.2f, Green=%.2f, Blue=%.2f", f, 0.0 , 0.0 );
    thread_sleep_for (10 );
}
for (float f =0.0 ; f <1.0 ; f +=0.05 ) {
    led_r =1.0 ; // off LED Red
    led_g =1.0 - f; // on LED Green
    led_b =1.0 ; // off LED Blue
    lcd .locate (0 , 16 );
    lcd .printf ("Red=%.2f, Green=%.2f, Blue=%.2f", 0.0 , f, 0.0 );
    thread_sleep_for (10 );
}
for (float f =0.0 ; f <1.0 ; f +=0.05 ) {
    led_r =1.0 ; // off LED Red
    led_g =1.0 ; // off LED Green
    led_b =1.0 - f; // on LED Blue
    lcd .locate (0 , 16 );
    lcd .printf ("Red=%.2f, Green=%.2f, Blue=%.2f", 0.0 , 0.0 , f);
    thread_sleep_for (10 );
}
}
}
}

```

2.2 프로그램 작성 아이디어 혹은 이유 설명 쓰기

PwmOut led_r(D5)를 사용하여 D5를 연결하고 lcd.cls()를 사용하여 lcd 화면을 초기화 해준다. 커서를 locate를 사용하여 0,6으로 이동 시켜주고 RGB LED!를 화면에 출력해 준다. 그 후 반복문을 사용하여 RGB의 값을 하나씩 변화 시켜준다. Shield에서 b가 작동하지 않아 r와 g값만 변화 시켜주었다. R을 0.05씩 줄여주고 g 값을 0.05씩 줄여준다. 줄여준 값을 화면에 출력해준다. 이때 led는 common anode이므로 1.0에서 0.05씩 빼서 값을 led에 넣어주고 출력은 뺀 값인 f를 출력했다.

2.3 하드웨어 구성 사진 첨부하기



2.4 프로그램 수행 사진/동영상(Youtube 링크) 첨부하기

<https://youtu.be/UV1x13Y3H58>

3. 프로그램 app-shield-joystick

3.1 프로그램 코드 쓰기

```
#include "mbed.h"
#include "C12832.h"

C12832 lcd (D11 , D13 , D12 , D7 , D10 ); // lcd = (MOSI, SCK, RESET, A0, nCS)

DigitalIn up (A2 ); // up = A1
DigitalIn down (A3 ); // down = A3
DigitalIn left (A4 ); // left = A4
DigitalIn rite (A5 ); // rite = A5
DigitalIn center (D4 ); // center = D4

int main ()
{
    lcd .cls ();
    lcd .locate (0 , 6 );
    lcd .printf ("Move Joystick!");
    while (true ) {
        if (up) { // if up
            lcd .locate (0 , 16 );
            lcd .printf ("Up ");
        }
    }
}
```

```

    if (down) { // if down
        lcd .locate (0 , 16 );
        lcd .printf ("Down ");
    }
    if (left) { // if left
        lcd .locate (0 , 16 );
        lcd .printf ("Left ");
    }
    if (rite) { // if right
        lcd .locate (0 , 16 );
        lcd .printf ("Right ");
    }
    if (center) { // if center
        lcd .locate (0 , 16 );
        lcd .printf ("Center ");
    }
    thread_sleep_for (200 );
}
}

```

3.2 프로그램 작성 아이디어 혹은 이유 설명 쓰기

DigitalIn 을 사용하여 5 way joystick을 A2,A3,A4,A5,에 넣어 주었다. 클릭은 D4에 넣어 주었다. 마찬가지로 화면을 lcd.cls로 초기화 시켜주고 0.6으로 커서를 이동 시켜서 Move JoyStick!을 시켜준다. 조건문을 사용하여 만약 위 방향을 입력하면 0,16에 커서를 이동시켜서 UP을 출력시킨다. 다른 방향이 나왔을 경우 방향에 맞는 문자를 출력하고 thread_sleep_for(200)을 통해 0.2초간 기다렸다가 다음 입력 값을 받아서 수행한다.

3.3 하드웨어 구성 사진 첨부하기



3.4 프로그램 수행 사진/동영상(Youtube 링크) 첨부하기

<https://youtu.be/CTzqYYYi5tM>

4. 프로그램 app-shield-pot

4.1 프로그램 코드 쓰기

```
#include "mbed.h"
#include "C12832.h"

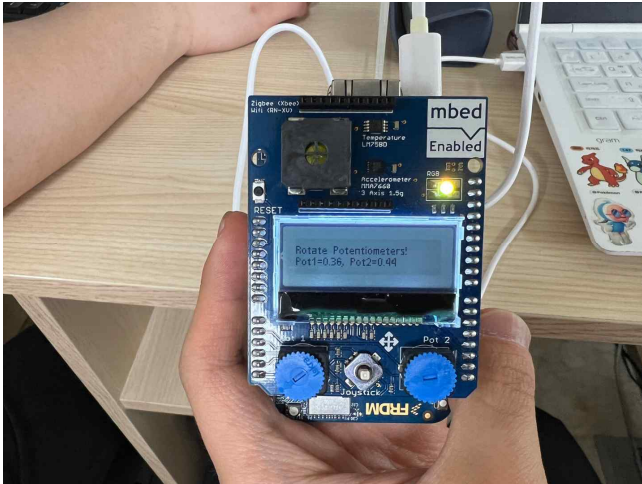
C12832 lcd (D11 , D13 , D12 , D7 , D10 ); // lcd = (MOSI, SCK, RESET, A0, nCS)
AnalogIn pot1 (A0 ); // pot1 = A0
AnalogIn pot2 (A1 ); // pot2 = A1
PwmOut led_r (D5 ); // led_r = (Red LED)
PwmOut led_g (D9 ); // led_g = (Green LED)

int main ()
{
    lcd .cls ();
    lcd .locate (0 , 6 );
    lcd .printf ("Rotate Potentiometers!");
    while (true ) {
        lcd .locate (0 , 16 );
        lcd .printf ("Pot1=%4.2f, Pot2=%4.2f", pot1 .read (), pot2 .read ());
        led_r =1.0 -pot1 .read (); // pot1 -> Red LED
        led_g =1.0 -pot2 .read (); // pot2 -> Green LED
        thread_sleep_for (200 );
    }
}
```

4.2 프로그램 작성 아이디어 혹은 이유 설명 쓰기

AnalogIn 을 통해 potentiometer1과 2를 각각 A0와 A1에 연결해준다. 가변저항의 변화를 led로 보여주기 위해 PwmOut을 사용해 led R과G를 각각 D5와 D9에 연결해준다. 화면을 cls를 사용해 초기화 해주고 0.6으로 커서를 이동시켜서 Rotate Potentiometers!를 출력한다. 0,16으로 커서를 이동시켜서 printf로 pot1과 pot2가 얼마만큼 돌아갔는지 pot1.read()로 값을 읽어 와서 출력해준다. 읽어온 값만큼 1에서 빼주어 led를 출력한다. Thread_sleep_for(200)을 사용해 0.2초간 기다렸다가 다음 명령을 수행해준다.

4.3 하드웨어 구성 사진 첨부하기



4.4 프로그램 수행 사진/동영상(Youtube 링크) 첨부하기

<https://youtu.be/Owc87uRmN1w>

5. 프로그램 app-shield-speaker

5.1 프로그램 코드 쓰기

```
#include "mbed.h"
#include "C12832.h"
#include "pitches.h"

C12832 lcd (D11 , D13 , D12 , D7 , D10 ); // lcd = (MOSI, SCK, RESET, A0, nCS)
PwmOut speaker (D6 ); // speaker = D6

int length =75 ;
float frequency[] = {
    NOTE_E6, NOTE_E6, 0 , NOTE_E6, 0 , NOTE_C6, NOTE_E6, 0 , NOTE_G6, 0 , 0 ,
    0 , NOTE_G5, 0 , 0 , 0 , NOTE_C6, 0 , 0 , NOTE_G5, 0 , 0 , NOTE_E5, 0 ,
    0 ,
    NOTE_A5, 0 , NOTE_B5, 0 , NOTE_AS5, NOTE_A5, 0 , NOTE_G5, NOTE_E6,
    NOTE_G6,
    NOTE_A6, 0 , NOTE_F6, NOTE_G6, 0 , NOTE_E6, 0 , NOTE_C6, NOTE_D6,
    NOTE_B5,
    0 , 0 , NOTE_C6, 0 , 0 , NOTE_G5, 0 , 0 , NOTE_E5, 0 , 0 , NOTE_A5, 0 ,
    NOTE_B5,
```

```

    0 , NOTE_AS5, NOTE_A5, 0 , NOTE_G5, NOTE_E6, NOTE_G6, NOTE_A6, 0 ,
NOTE_F6,
    NOTE_G6, 0 , NOTE_E6, 0 , NOTE_C6, NOTE_D6, NOTE_B5, 0 , 0
};

float beat[] = {
    12 , 12 , 12 , 12 , 12 , 12 , 12 , 12 , 12 , 12 , 12 , 12 , 12 , 12 , 12 , 12 ,
, 12 , 12 ,
    12 , 12 , 12 , 12 , 12 , 12 , 12 , 12 , 12 , 12 , 12 , 12 , 12 , 12 , 12 , 12 ,
, 9 , 9 , 9 , 12 ,
    12 , 12 , 12 , 12 , 12 , 12 , 12 , 12 , 12 , 12 , 12 , 12 , 12 , 12 , 12 , 12 ,
, 12 , 12 , 12 ,
    12 , 12 , 12 , 12 , 12 , 12 , 12 , 12 , 12 , 12 , 9 , 9 , 9 , 12 , 12 , 12 ,
12 , 12 , 12 , 12 ,
    12 , 12 , 12 , 12 , 12 ,
};

int main ()
{
    lcd .cls (); lcd .locate (0 , 6 ); lcd .printf ("Listen to speaker!");
    lcd .locate (0 , 16 ); lcd .printf ("Mario~~~");
    while (true ) {
        for (int i =0 ; i <= length; i ++ ) {
            if (frequency [i] ==0 ) {
                speaker =0.0 ;
            } else {
                speaker .period (1.0 /frequency [i]); // period = (1.0 /
frequency)
                speaker =0.5 ; // duty cycle = 50%
            }
            thread_sleep_for (2500.0 /beat [i]); // duration = (C / beat) ms
        }
    }
}

```

5.2 프로그램 작성 아이디어 혹은 이유 설명 쓰기

Pitches.h를 import하여 주파수 매크로를 받아오고 마리오 악보를 frequency와 beat에 저장한다. 화면에는 0.6으로 커서를 이동시킨 후에 Listen to speaker!를 출력하고 0,16에는 Mario~~~를 출력한다. 그후에 스피커에서 마리오 음악을 출력한다.

5.3 하드웨어 구성 사진 첨부하기



5.4 프로그램 수행 사진/동영상(Youtube 링크) 첨부하기

<https://youtu.be/Jz8n1PQ1vKM>

6. 프로그램 app-shield-3-axis

6.1 프로그램 코드 쓰기

```
#include "mbed.h"
#include "C12832.h"
#include "MMA7660.h"

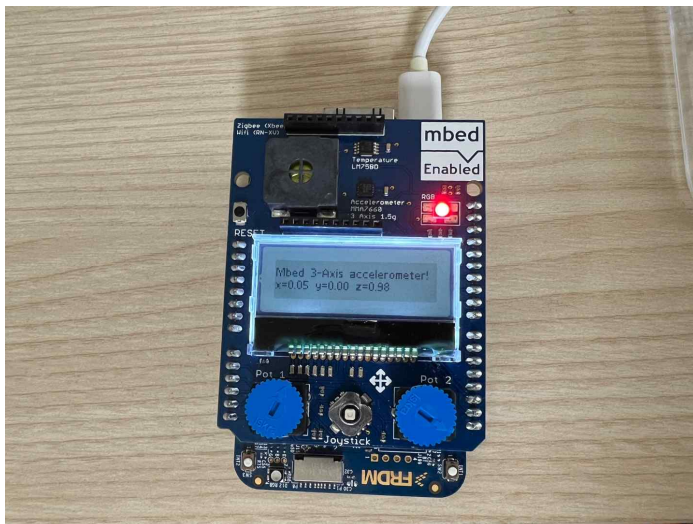
C12832 lcd (D11 , D13 , D12 , D7 , D10 ); // lcd = (MOSI, SCK, RESET, A0, nCS)
MMA7660 MMA (D14 , D15 ); // MMA = (I2C_SDA, I2C_SCL)

int main ()
{
    lcd .cls ();
    lcd .locate (0 , 6 );
    lcd .printf ("Mbed 3-Axis accelerometer!");
    while (true ) {
        lcd .locate (0 , 16 );
        lcd .printf ("x=%.2f y=%.2f z=%.2f", MMA .x (), MMA .y (), MMA .z ());
        thread_sleep_for (100 );
    }
}
```

6.2 프로그램 작성 아이디어 혹은 이유 설명 쓰기

MMA7660.h를 import하여 3축 가속도계를 사용한다. MMA7660의 MMA를 사용해 D14,D15 핀에 연결한다. cls를 사용해 화면을 초기화 해주고 0,6dp Mbed 3-Axis accelerometer!를 출력한다. 0,16으로 이동해 x,y,z가 얼마나 변화하는지 출력한다. 값은 MMA.x(), MMA.y() MMA.z()로 값을 받아와서 화면에 출력해 준다. Thread_sleep_for(100)으로 0.1초간 기다렸다가 다음 명령을 수행한다.

6.3 하드웨어 구성 사진 첨부하기



6.4 프로그램 수행 사진/동영상(Youtube 링크) 첨부하기

<https://youtu.be/jgpvMs03kEo>

7. 프로그램 app-shield-temp

7.1 프로그램 코드 쓰기

```
#include "mbed.h"
#include "C12832.h"
#include "LM75B.h"

C12832 lcd (D11 , D13 , D12 , D7 , D10 ); // lcd = (MOSI, SCK, RESET, A0, nCS)
LM75B temp (D14 , D15 ); // temp = (SDA, SCL)

int main ()
{
    lcd .cls ();
```

```

lcd .locate (0 , 6 );
lcd .printf ("Current temperature!");
while (true ) {
    lcd .locate (0 , 16 );
    lcd .printf ("%5.2f degree Celsius", temp .read ());
    thread_sleep_for (1000 );
}
}

```

7.2 프로그램 작성 아이디어 혹은 이유 설명 쓰기

LM75B temp를 사용하여 온도 센서를 D14와 D15에 연결해준다. cls를 하여 화면을초기화 해주고 0,6으로 커서를 이동시킨 후에 Current temperature!를 출력해준다.

커서를 0,16으로 이동시킨후에 현재 온도 degree Celsius를 출력해준다. 온도는 temp.read()로 읽어와서 출력해준다. Thread_sleep_for(1000)을 활용하여 1초간 기다린후 다음 명령을 수행한다.

7.3 하드웨어 구성 사진 첨부하기



7.4 프로그램 수행 사진/동영상(Youtube 링크) 첨부하기

<https://youtu.be/7K6hjaFY5BM>

끝.