

강의명: 임베디드 시스템

숙제 번호: 3

숙제 제목: Pulse-width-modulation(펄스 폭 변조)

학생 이름: 한규현

1. 프로그램 pwm-led

1.1 프로그램 코드 쓰기

```
#include "mbed.h"

PwmOut led (PTA1 ); // led = PTA1

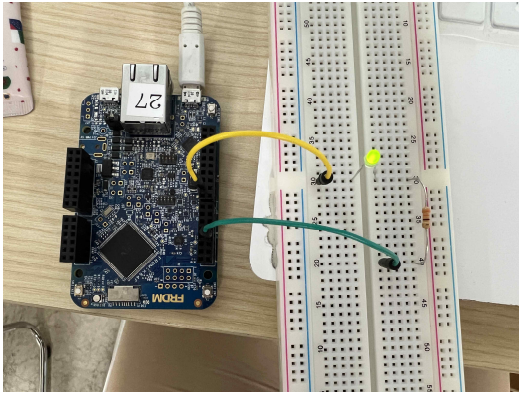
int main ()
{
    float duty;
    led .period (10.0 /1000 );
    while (true ) {
        for (duty =0.0 ; duty <1.1 ; duty = duty +0.1 ) {
            led .write (duty);
            thread_sleep_for (1000 );
        }
    }
}
```

1.2 프로그램 작성 아이디어 혹은 이유 설명 쓰기

PWMOut pin인 PTA1에 green LED를 연결하여 LED 출력하고자 PwmOut led(PTA1)과 같이 코드를 작성했다.

period는 10.0/1000으로 설정했고, duty cycle의 범위는 0.0-1.0이므로 for문을 위의 코드와 같이 설정했다. 이후 duty 변수의 값에 따라 led.write(duty);로 코드를 작성해 led가 출력된다. 마지막에는 LED의 밝기가 서서히 조작되도록 해서 사람의 눈으로 판단할 수 있도록 thread_sleep_for(1000);을 작성하여 텀을 두었다.

1.3 하드웨어 구성 사진 첨부하기



1.4 프로그램 수행 사진/동영상(Youtube 링크) 첨부하기

<https://youtu.be/CFW74O6-7qg>

2. 프로그램 pwm-music

2.1 프로그램 코드 쓰기

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

PwmOut buzzer (PTA1 );      // buzzer = PTA1

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 , 9 , 9 , 9 , 12 , 12 , 12 ,
    12 , 12 , 12 , 12 ,
    12 , 12 , 12 , 12 , 12 ,
};

int main ()
{
    while (true ) {
        for (int i =0 ; i <= length; i ++ ) {
            if (frequency [i] ==0 ) {
                buzzer =0.0 ;
            } else {
                buzzer .period (1.0 /frequency [i]); // period = (1.0 /
frequency)
                buzzer =0.5 ; // duty cycle = 50%
            }
            thread_sleep_for (2500.0 /beat [i]); // duration = (C / beat) ms
        }
    }
}

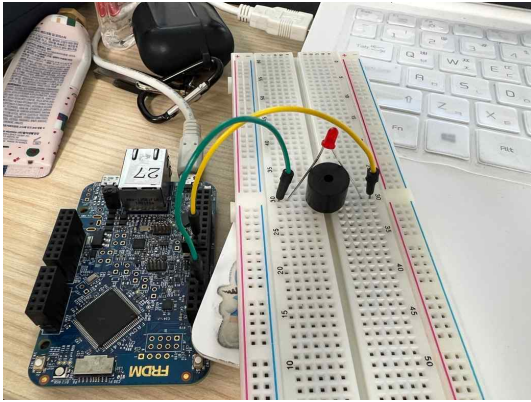
```

2.2 프로그램 작성 아이디어 혹은 이유 설명 쓰기

PWMOut pin인 PTA1에 부저를 연결하고자 PwmOut buzzer(PTA1);으로 작성하였다. 슬라이드6-14를 참고하여 하나의 음악을 음의 주파수 및 길이의 배열로 표현했고 상단 코드에 float frequency[], float beat[] 부분과 같이 작성했다. frequency가 0인 경우, 음악이 울리지 않는 상황이다.

길이 75만큼 음악이 재생되도록 for문을 작성했다. frequency가 0이면 부저에 0 값을 쓰게 했고, 그렇지 않으면 주파수의 역순으로 주기를 구하고, duty cycle을 50%로 설정한다. 이후 thread_sleep_for(2500.0/beat[i]);로 작성하여 음악 사이사이 시간(duration)을 2500/beat[i]로 적당한 시간에 텀을 주었다.

2.3 하드웨어 구성 사진 첨부하기



2.4 프로그램 수행 사진/동영상(Youtube 링크) 첨부하기

<https://youtu.be/RpYc4RG0TXo>

3. 프로그램 pwm-sg90-servo

3.1 프로그램 코드 쓰기

```
#include "mbed.h"

PwmOut servo (PTA1 ); // servo = PTA1

int main ()
{
    float w;
    servo .period (20.0/1000.0 ); // period = 20 ms for SG90 Servo

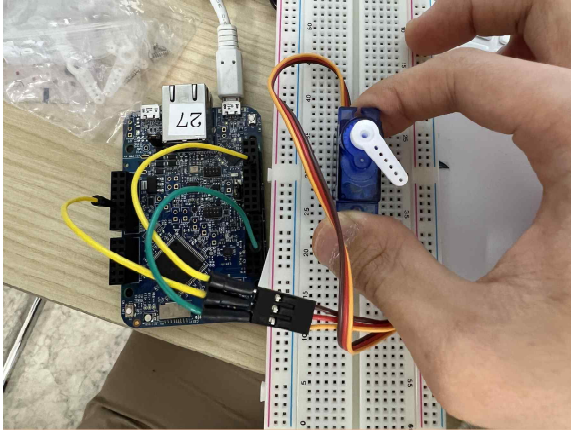
    while (true ) {
        // 500 to 2,500 us for degree 0 to 180
        for (w =500 ; w <=2500 ; w = w +100 ) {
            servo .write (w / (20.0 *1000.0 ));
            thread_sleep_for (1000 );
        }
    }
}
```

3.2 프로그램 작성 아이디어 혹은 이유 설명 쓰기

PWMOut pin인 PTA1에 SG90 서브 모터를 PwmOut servo (PTA1);와 같이 연결했다. 각도 (degree)를 조절하기 위해 0도(w=500) 180도(w=2500) 사이 w가 100씩 증가하면서 각도 조절이 가능하게 했다. w값 변화에 따라 servo.write(w/20.0*1000.0);으로 servo 모터가 작동

하게 된다. 서브모터가 너무 빠르게 동작하지 않도록 `thread_sleep_for(1000);`으로 텀을 두었다.

3.3 하드웨어 구성 사진 첨부하기



3.4 프로그램 수행 사진/동영상(Youtube 링크) 첨부하기

<https://youtu.be/CFW7406-7qg>

끝.