

예제 1)

Led와 Buzzer를 동시에 제어하는 실습입니다. lab1-2_1과 lab1-4_2를 동시에 수행함으로써 멀티태스킹이 잘 이루어지는 지 확인을 해볼 수 있었습니다. 아두이노의 5번 핀을 LED 1번에, 9번 핀을 Buzzer모듈에 연결하였고 함수 작성 이전에 LED번호, 음계를 Define 하였습니다. 그리고 FreeRTOS Delay 함수에서 사용하기 위한 매크로 함수로 ms를 tick으로 변환하기 위해 MS2TICKS(ms)를 새로 정의하였습니다.

LedTask함수는 lab1-2_1과 동일하지만 Delay(500)이 아니라 vTaskDelay를 사용함으로써 스케줄러에 상대 시간동안 Delay를 요청할 수 있게 하였습니다. BuzzerTask함수도 마찬가지로 lab1-4_2와 동일하지만 음을 내는 시간동안 vTaskDelay를 사용하였습니다. 이후 Setup()에서 LED와 Buzzer 핀번호를 출력으로 바꾸고 xTaskCreate를 통해 태스크를 우선순위에 따라 형성하였습니다. 그리고 vTaskStartScheduler 함수를 통해 스케줄링을 통해 태스크를 실행시켰습니다.

실행결과 Buzzer에서 음을 내는 것과 Led가 0.5초마다 점등되는 것이 동시에 실행되는 것을 확인할 수 있었습니다.

```
void LedTask(void* arg) {
    while(1) {
        digitalWrite(LED, HIGH);
        vTaskDelay(MS2TICKS(500));

        digitalWrite(LED, LOW);
        vTaskDelay(MS2TICKS(500));
    }
}

void BuzzerTask(void* arg) {
    int i;
    while(1) {
        for(i=0; i<Num; i++) {
            tone(BUZZER, Frequency[i]);
            vTaskDelay(MS2TICKS(Delay[i]));
        }
    }
}

void setup() {
    // put your setup code here, to run once:
    pinMode(LED, OUTPUT);
    pinMode(BUZZER, OUTPUT);

    xTaskCreate(LedTask, NULL, 200, NULL, 1, NULL);
    xTaskCreate(BuzzerTask, NULL, 200, NULL, 2, NULL);

    vTaskStartScheduler();
}
```

<Lab3-2_1 코드>

예제 2)

Led1번과 2번을 점등하는 것을 각각 다른 태스크로 수행해서 멀티태스킹이 잘 되는지 확인하는 실습입니다. 아두이노의 5, 6번 핀을 각각 Led모듈의 1, 2번에 연결하였습니다. 이후 LedTask 함수에서는 param변수를 하나 받을 수 있게 하였습니다. 1을 받을 시 1번 Led가 작동되고 2를 받을 시 2번 Led가 작동되게 됩니다. led 변수에 어떤 불을 켤지 입력을 받으면 while문 안에서 해당 LED를 점등하게 됩니다. 예제 1과 똑같이 Delay가 아니라 태스크의 상대시간 동안 Delay를 시키는 vTaskDelay를 사용하였고 해당 불이 on이면 off시키고 off면 on시킬 수 있게 led를 제어하였습니다.

이후 Setup에서 LED1, 2번호를 출력으로 설정하였고 xTaskCreate를 통해 태스크를 생성하였습니다. 이 때 1번을 param배열을 생성하여 1번이 될 시 첫 번째 태스크, 2번이 될 시 두 번째 태스크를 실행할 수 있게 하였습니다.

실행 결과 LedTask안에서의 led변수가 각자 다른 함수(태스크)에 있기 때문에 독자적으로 1번, 2번의 Led를 점등시키는 것을 확인할 수 있었습니다.

```
void LedTask(int* pParam) {
    int led, turn, param = *pParam;
    if(param==1) {
        led=LED1; turn=HIGH;
    }
    else {
        led=LED2; turn=LOW;
    }

    while(1) {
        digitalWrite(led, turn);
        vTaskDelay(MS2TICKS(500));
        turn=(turn==HIGH)? LOW:HIGH;
    }
}

void setup() {
    // put your setup code here, to run once:
    int param[2]={1,2};
    pinMode(LED1, OUTPUT);
    pinMode(LED2, OUTPUT);

    xTaskCreate(LedTask, NULL, 200, &param[0], 1, NULL);
    xTaskCreate(LedTask, NULL, 200, &param[1], 2, NULL);

    vTaskStartScheduler();
}
```

<Lab3-2_2 코드>

과제 1)

예제 1번을 멀티태스킹을 사용하지 않고 구현하는 실습입니다. 멀티태스킹을 사용하지 않고 Delay만을 사용해서 하려니 쉽지 않았고 멀티태스킹의 필요성을 확인할 수 있었습니다.

효율적으로 코드를 짜기 위해 Frequency배열을 바꿨습니다. 원래는 도레미파솔라시도가 있었다면 실제로 출력되는 음인 도미파솔라도만 배열에 넣어주고 라는 0.5초가 아니라 1초동안 출력이 되기 때문에 두 개를 넣어주었습니다. 예제 1번을 잘 보면 Led가 0.5초마다 점등되고, 음도 0.5초마다 바뀌기 때문에 일정 sync에 맞춰서 태스크가 실행되는 것을 알 수 있습니다. 그렇기 때문에 for문안에 1가 8번 돌아가면서 1가 짝수일 땐 불을 켜주고, 불을 꺼주었습니다. 그리고 그 시간에 맞추어 나와야 하는 음이 Frequency에 저장되어 있어서 해당 음을 출력하였습니다.

실행 결과 멀티태스킹은 쓰이지 않았지만 Delay되는 시간에 맞추어 두 가지 모듈이 동시 동작하는 것을 확인할 수 있었습니다.

```
ass3-2_1
enum{REST=0, DO=262, RE=294, MI=330, FA=349, SOL=392, LA=440, SHI=404, DO2=523};
#define BUZZER 9
#define LED 5
int NUM=9;
int Frequency[]={DO,MI,REST,FA,SOL,LA,LA,DO2};

void setup() {
    pinMode(BUZZER, OUTPUT);
    pinMode(LED, OUTPUT);
}

void loop() {
    for(int i=0; i<8; i++) {
        tone(BUZZER, Frequency[i]);
        if(i%2==0) digitalWrite(LED, HIGH);
        else digitalWrite(LED, LOW);
        delay(500);
    }
}
```

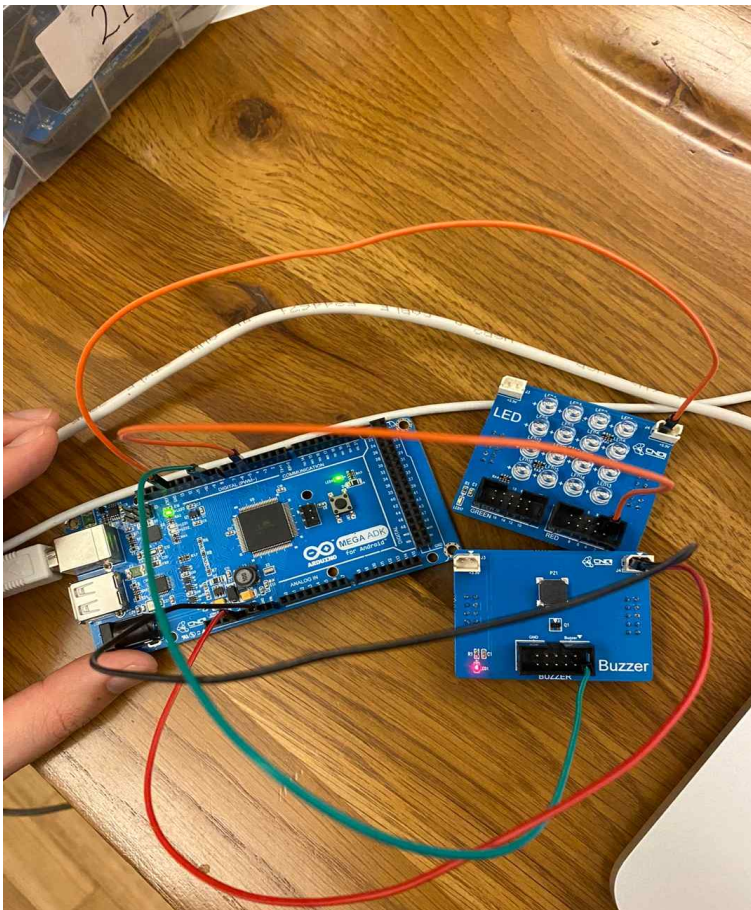
<ass3-2_1 코드>

과제 2)

예제 2번에서 각각 LedTask와 BuzzerTask에 있는 vTaskDelay를 Delay로 변경한 다음 실행결과가 어떻게 변화하는 지 관찰하는 실습입니다.

1) LedTask와 BuzzerTask 모두 vTaskDelay를 Delay로 변경:

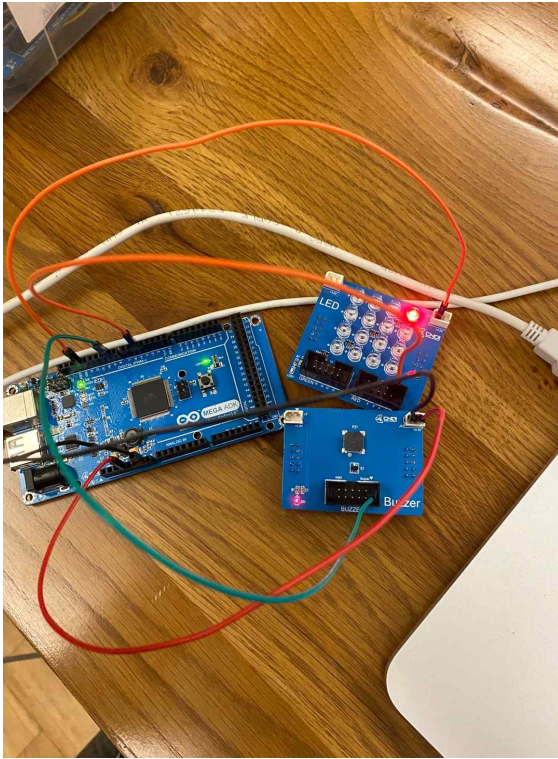
아무일도 일어나지 않습니다. vTaskDelay() 함수를 통해서 입력 tick동안 Sleep을 해줘야하는데 두 함수 다 Delay를 사용해서 태스크 별로 상대시간 동안의 Delay가 발생하지 않게 됩니다. 따라서 입력 tick이 없으므로 상대시간 0동안의 출력을 계속하기 때문에 아무일이 일어나지 않습니다.



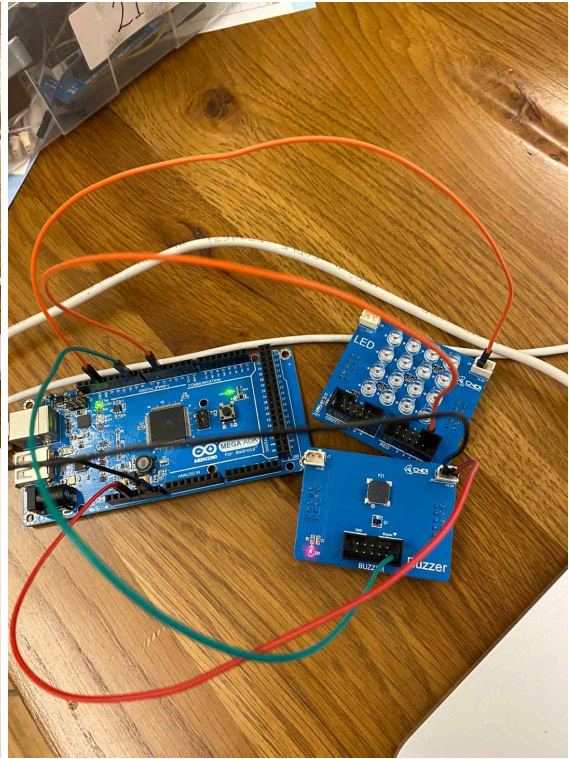
<ass3-2_2_1.c 실행결과>

2) LedTask만 vTaskDelay를 Delay로 변경:

LedTask만 작동하게 됩니다. vTaskStartScheduler로 태스크가 생성되면서 각 태스크들의 상대시간만큼 지연이 되는게 아니라 LedTask는 Delay를 통해 절대적인 시간만큼의 작동을 하게 됩니다. While문에서 계속 반복돼서 태스크가 수행되면서 우선순위가 2인 BuzzerTask는 실행되지 않은 채로 계속 LedTask만 수행되는 것입니다.

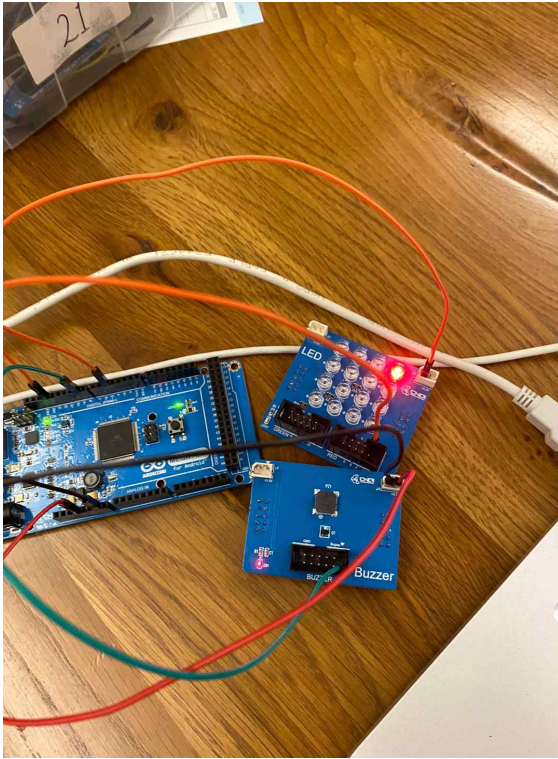


<ass3-2_2_1.c ON>

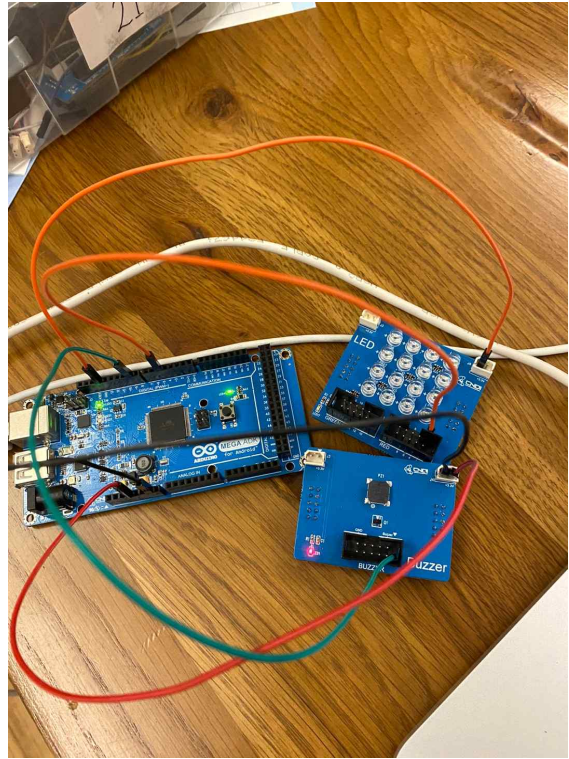


<ass3-2_2_1.c OFF>

3) LedTask만 vTaskDelay를 Delay로 변경 후 우선순위를 서로 바꾸기
 LedTask만 작동하게 됩니다. vTaskStartScheduler로 태스크가 생성되면서 각 태스크들의 상대시간만큼 지연이 되는게 아니라 LedTask는 Delay를 통해 절대적인 시간만큼의 작동을 하게 됩니다. 설령 BuzzerTask의 우선순위가 높더라도 Tick이 발생하지 않고 바로 끝나버리므로 LedTask만 실행되는 것처럼 보이는 것입니다.



<ass3-2_3_1.c ON>



<ass3-2_3_1.c OFF>

사진이 첨부되지 않은 실습은 동영상으로 폴더에 첨부하였습니다.