



# Multitasking (LED & Buzzer)

조진성

경희대학교 컴퓨터공학과

Mobile & Embedded System Lab.



# Task Creation & Kernel Control

## ❖ Task creation in FreeRTOS

- BaseType\_t xTaskCreate(TaskFunction\_t pvTaskCode,  
const char \* const pcName,  
unsigned short usStackDepth,  
void \*pvParameters,  
UBaseType\_t uxPriority,  
TaskHandle\_t \*pxCreatedTask);

## ❖ Kernel control in FreeRTOS

- void vTaskStartScheduler(void);
  - 멀티태스킹을 시작. 즉, 생성된 태스크의 스케줄링을 시작

# Task Control



## ❖ Delay function in FreeRTOS

- `void vTaskDelay(portTickType xTicksToDelay);`
  - 스케줄러에 상대(relative) 시간 동안 delay를 요청 (waiting state)
  - 함수가 호출된 시점부터 xTicksToDelay 동안 태스크 실행을 블록
  
- `void vTaskDelayUntil(portTickType *pxPreviousWakeTime,  
portTickType xTimeIncrement);`
  - 스케줄러에 절대(absolute) 시간 동안 delay를 요청 (waiting state)
  - 특정 시점인 pxPreviousWakeTime부터 xTimeIncrement 만큼 더한 시간까지 태스크 실행을 블록 (pxPreviousWakeTime은 실제로 xTimeIncrement만큼 더해짐)

# Task Control



## ❖ Delay function in FreeRTOS

- 다음과 같은 두 개의 태스크가 있는 경우를 비교 분석
- foo() 함수는 실행되는데 0.5초의 시간이 걸린다고 가정

```
void task1(...)
{
    while (1) {
        foo();
        vTaskDelay(1000/portTICK_PERIOD_MS);
    }
}

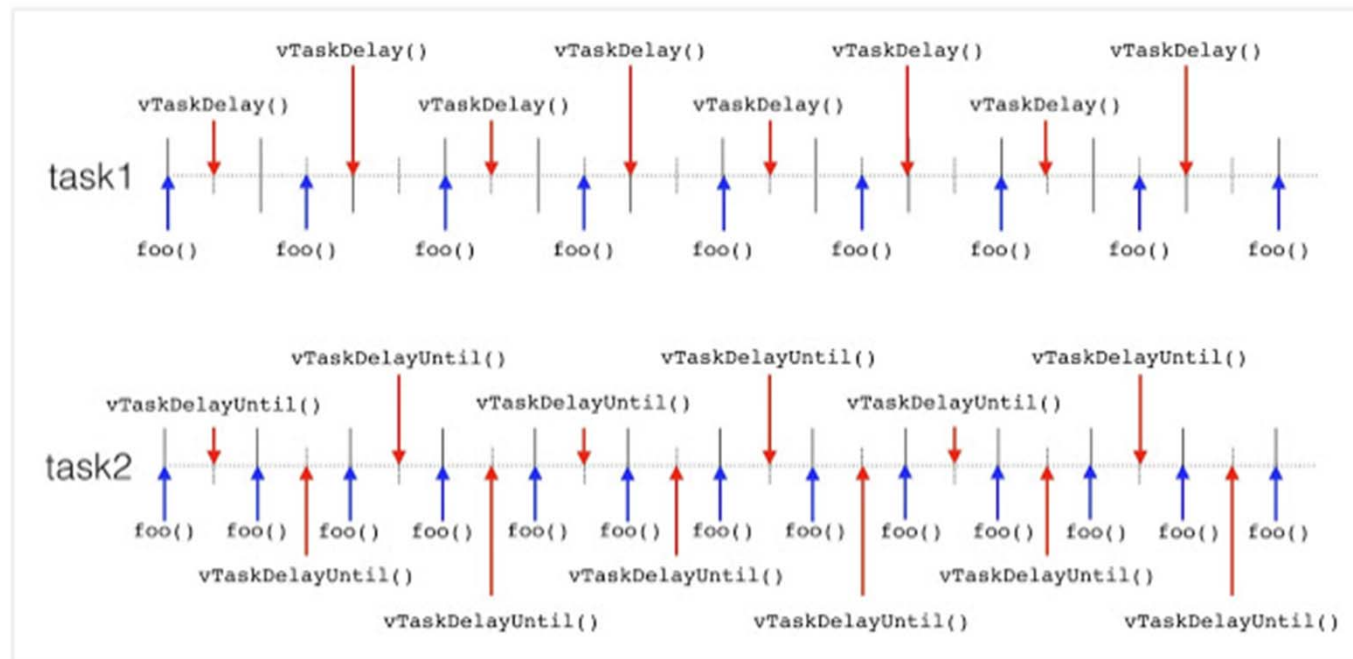
void task2(...)
{
    portTICK_TYPE xLastWakeTime = xTaskGetTickCount();

    while (1) {
        foo();
        vTaskDelayUntil(&xLastWakeTime, 1000/portTICK_PERIOD_MS);
    }
}
```

# Task Control

## ❖ Delay function in FreeRTOS

- task1의 경우 0초에 foo() 함수를 호출하고 0.5초에 vTaskDelay() 함수를 호출
- 이후 1초를 블럭한 뒤 다시 1.5초에 foo()를 호출
  - 즉 foo() 함수는 1.5초에 한번씩 호출
- task2의 경우 foo() 함수의 실행 시간에 관계없이 정확히 1초에 한번씩 foo() 함수를 호출
  - foo() 함수의 실행시간은 delay 시간(여기서는 1초)보다는 짧아야 함



# 하드웨어 구성 (1)

## ❖ LED와 Buzzer를 동시에 제어

- lab1-2\_1.ino와 lab1-4\_2.ino를 동시에 수행
- Lab. 1-2) 실습 예제(1)과 Lab. 1-4) 실습 예제(2)의 하드웨어 구성을 동시에 연결





# 하드웨어 구성 (1)

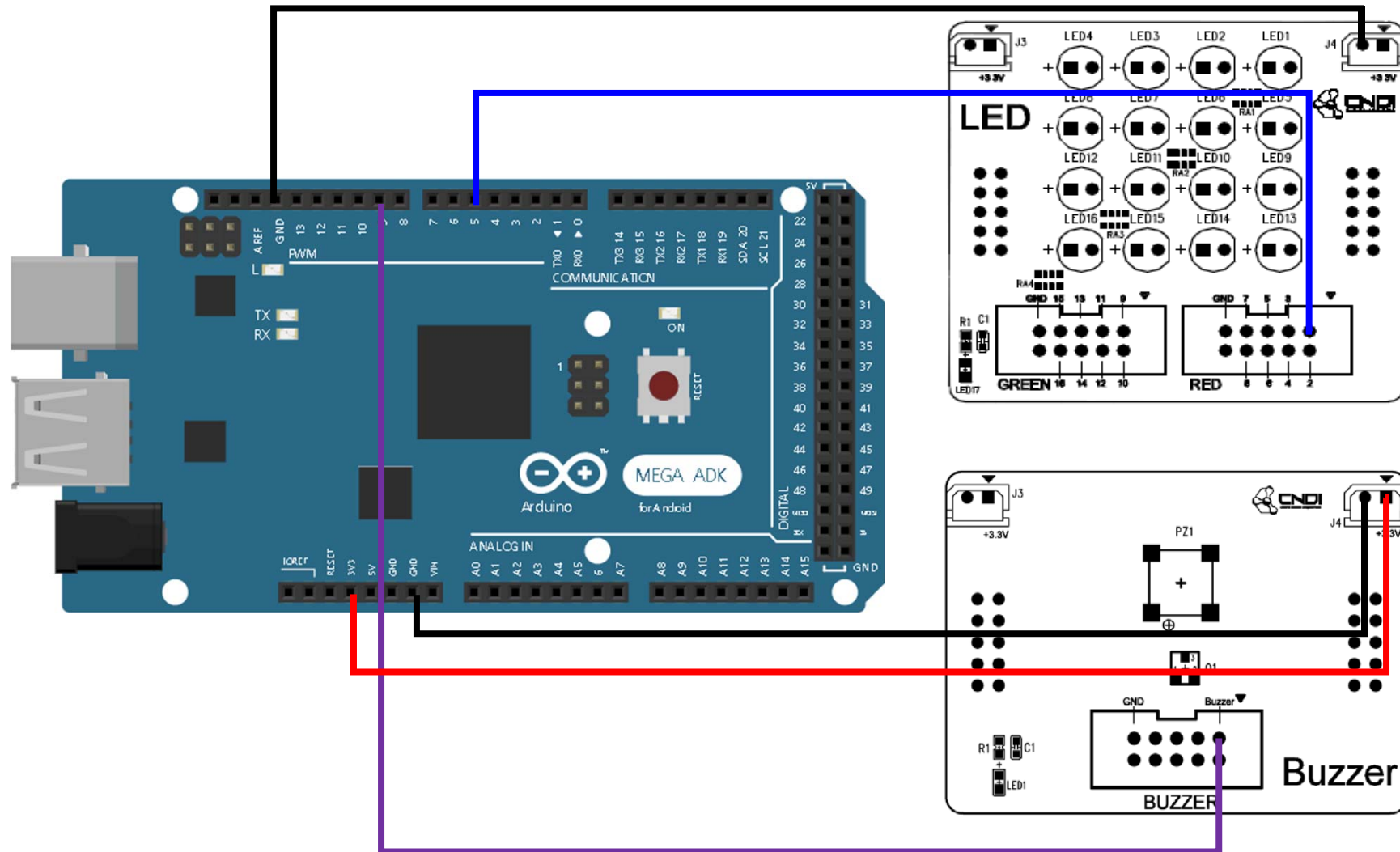


## ❖ LED와 Buzzer를 동시에 제어

- Arduino MEGA ADK의 DIGITAL 5와 LED 모듈의 RED 포트의 핀 1을 연결
- Arduino MEGA ADK의 GND(0V)와 LED 모듈의 J4 포트의 좌측에 연결
- Arduino MEGA ADK의 DIGITAL 9와 Buzzer 모듈의 BUZZER 포트의 핀 Buzzer를 연결
- Arduino MEGA ADK의 3.3V와 Buzzer 모듈의 J4 포트의 +3.3V와 연결
- Arduino MEGA ADK의 GND(0V)와 Buzzer 모듈의 J4 포트의 좌측에 연결

# 하드웨어 구성 (1)

## ❖ LED와 Buzzer를 동시에 제어





# 실습 예제 (1)



## ❖ LED와 Buzzer를 동시에 제어

▪ lab3-2\_1.ino

```
// FreeRTOS 라이브러리 사용을 위한 헤더파일 선언
#include <FreeRTOS_AVR.h>

// FreeRTOS Delay 함수에서 사용하기 위한 매크로 함수: ms -> ticks 변환
#define MS2TICKS(ms) (ms / portTICK_PERIOD_MS )

// Arduino Pin 정의
#define LED      5
#define BUZZER   9

// 음계 정의
enum { REST=0, DO=262, RE=294, MI=330, FA=349, SOL=392, LA=440, SHI=494, DO2=523 };

int Num = 9;
int Frequency[] = { DO, RE, MI, REST, FA, SOL, LA, SHI, DO2 }; // 음계 정의
int Delay[] = { 500, 0, 500, 500, 500, 500, 1000, 0, 500 }; // 유지시간 정의
```

# 실습 예제 (1)

## ❖ LED와 Buzzer를 동시에 제어

### ▪ lab3-2\_1.ino

```
void LedTask(void* arg) {
    while (1) {
        // LED ON
        digitalWrite(LED, HIGH);
        // 500ms 지연
        vTaskDelay(MS2TICKS(500));
        // LED OFF
        digitalWrite(LED, LOW);
        // 500ms 지연
        vTaskDelay(MS2TICKS(500));
    }
}

void BuzzerTask(void* arg) {
    int i;

    while (1) {
        for (i = 0; i < Num; i++) {
            // 9번 PIN을 통해 출력
            tone(BUZZER, Frequency[i]);
            // 유지 시간
            vTaskDelay(MS2TICKS(Delay[i]));
        }
    }
}
```

```
void setup() {
    // LED, Buzzer Pin 출력으로 설정
    pinMode(LED, OUTPUT);
    pinMode(BUZZER, OUTPUT);

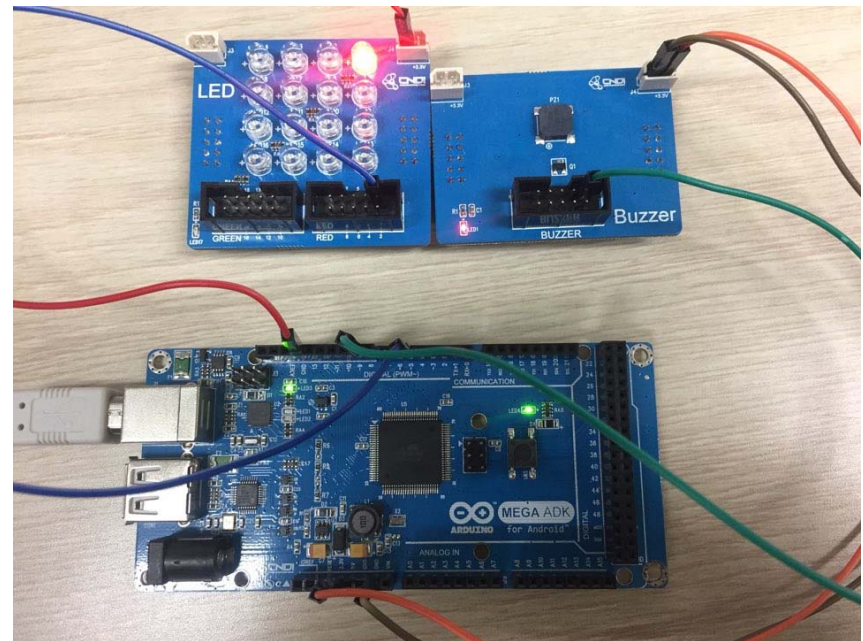
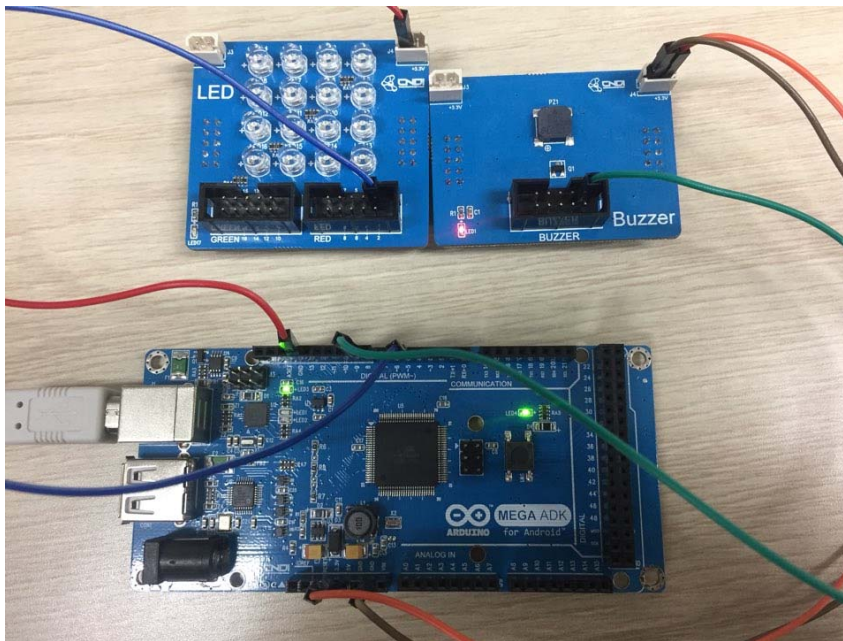
    // Task 생성
    xTaskCreate(LedTask, NULL, 200, NULL, 1, NULL);
    xTaskCreate(BuzzerTask, NULL, 200, NULL, 2, NULL);

    // scheduler 시작
    vTaskStartScheduler();
}

void loop() {
}
```

# 실습 예제 (1)

- ❖ LED와 Buzzer를 동시에 제어
  - 동작 화면



## 하드웨어 구성 (2)

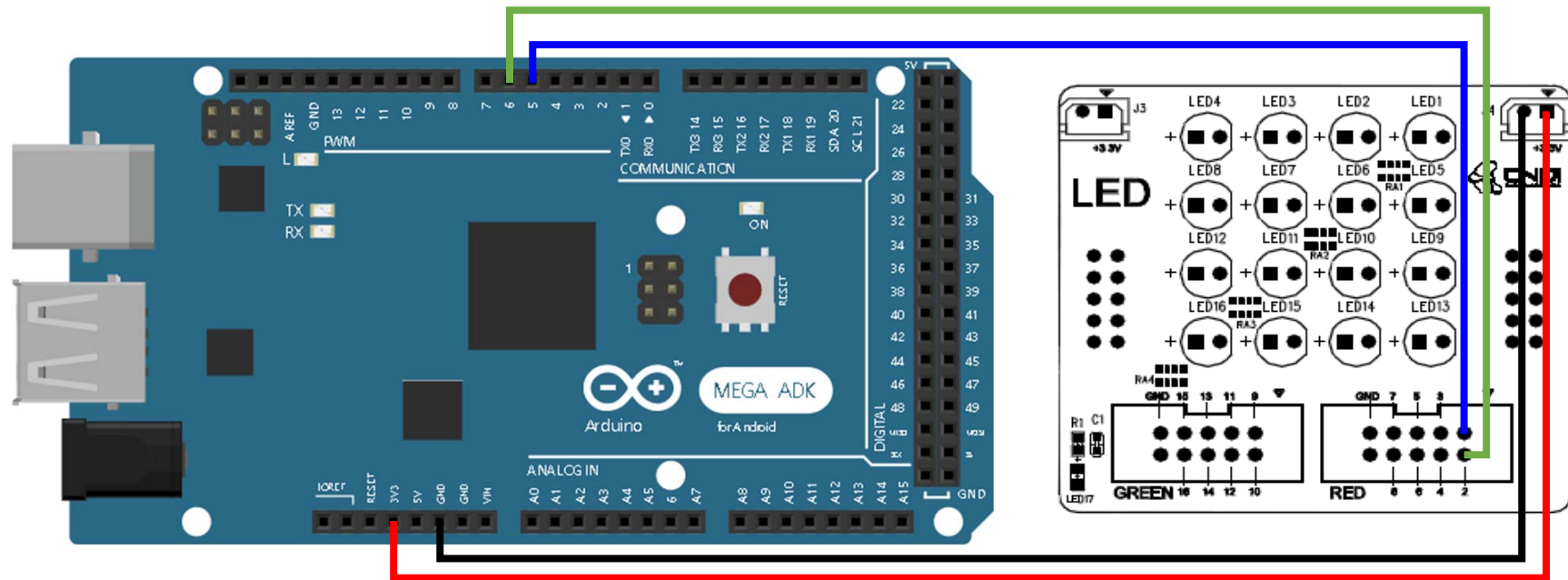


### ❖ LED 1개씩 ON/OFF task 2개 동작

- 500ms 간격으로 ON/OFF
- 실습을 위한 Arduino MEGA ADK와 LED 결선 방법
  - Arduino MEGA ADK의 DIGITAL 5와 LED 모듈의 RED 포트의 핀 1을 연결
  - Arduino MEGA ADK의 DIGITAL 6과 LED 모듈의 RED 포트의 핀 2를 연결
  - Arduino MEGA ADK의 3.3V와 LED 모듈의 J4 포트의 +3.3V와 연결
  - Arduino MEGA ADK의 GND(0V)와 LED 모듈의 J4 포트의 좌측에 연결

## 하드웨어 구성 (2)

- ❖ LED 1개씩 ON/OFF task 2개 동작
  - 결선 그림



## 실습 예제 (2)



### ❖ LED 1개씩 ON/OFF task 2개 동작

#### ▪ lab3-2\_2.ino

```
#include <FreeRTOS_AVR.h>

#define MS2TICKS(ms) (ms / portTICK_PERIOD_MS )

#define LED1 5
#define LED2 6

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);
        // 500ms 지연
        vTaskDelay(MS2TICKS(500));
        turn = (turn == HIGH) ? LOW : HIGH;
    }
}
```

```
void setup() {
    // LED, Buzzer Pin 출력으로 설정
    int param[2] = { 1, 2 };

    pinMode(LED1, OUTPUT);
    pinMode(LED2, OUTPUT);

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

    // scheduler 시작
    vTaskStartScheduler();
}

void loop() {
}
```



# 실습 과제



## ❖ 1) 실습 예제를 FreeRTOS 없이 작성

- lab1-2\_1.ino와 lab1-4\_2.ino를 동시에 수행하는 아두이노 프로그램 작성
- Multitasking의 필요성 확인

## ❖ 2) lab3-2\_1.ino에서

- [2-1] LedTask/BuzzerTask에서 모든 vTaskDelay( )를 delay( )으로 변경 후 동작을 확인하고, 원인을 분석하여 제시
- [2-2] LedTask에서만 vTaskDelay( ) 대신 delay( )으로 변경 후 동작을 확인하고, 원인을 분석하여 제시
- [2-3] LedTask에서만 vTaskDelay( ) 대신 delay( )으로 변경하고, LedTask의 우선순위를 2로, BuzzerTask의 우선순위를 1로 변경 후 동작을 확인하고, 원인을 분석하여 제시



# Q & A



<http://mesl.khu.ac.kr>