



# KeyPad 디바이스 제어

조진성

경희대학교 컴퓨터공학과

Mobile & Embedded System Lab.



Computer Engineering in KyungHee University

**Mobile & Embedded System Lab.**

# Interrupt



## ❖ 클럭 (Clock)

- 클럭 신호는 시간의 흐름에 따른 High와 Low 상태의 변화를 나타냄
- 클럭 신호의 종류
  - 상승 엣지(Rising Edge): 클럭이 Low -> High로 변경
  - 하강 엣지(Falling Edge): 클럭이 High -> Low로 변경
- 주기(Period)와 주파수(Frequency)
  - 주기: 상승(또는 하강) 엣지 발생 후 다음으로 상승(또는 하강) 엣지가 발생할 때 까지의 시간
  - 주파수: 1초당 발생하는 주기의 횟수
  - 주기와 주파수는 역수 관계( $\text{주기} = \frac{1}{\text{주파수}}$ ,  $\text{주파수} = \frac{1}{\text{주기}}$ )

## ❖ 카운터(Counter)

- 클럭 신호의 횟수를 측정하여 처리하는 장치
- 클럭 신호가 하나 발생할 때마다 카운터/타이머가 1씩 증가



# Interrupt



## ❖ Interrupt in Arduino

- Arduino는 인터럽트 제어를 위한 함수를 제공
  - `attachInterrupt(intNo, ISR, mode)`: ISR 등록
  - `detachInterrupt(intNo)`: ISR 해제
  - `Interrupts()`: 전체 인터럽트 활성화
  - `noInterrupts()`: 전체 인터럽트 비활성화
- `attachInterrupt(intNo, ISR, mode)`
  - intNo: 인터럽트 번호
    - Arduino는 인터럽트 번호를 가지고 있으며 이를 통해 인터럽트를 제어
    - Arduino MEGA ADK의 인터럽트 번호 및 대응 핀 번호
      - Arduino MEGA ADK에서 0번 인터럽트를 사용하기 위해서는 2번 핀을 통해서만 제어 가능
      - Arduino MEGA ADK에서 3번 인터럽트를 사용하기 위해서는 20번 핀을 통해서만 제어 가능
      - 참고) Arduino UNO는 2개의 인터럽트가 존재

인터럽트 번호	0	1	2	3	4	5
Arduino MEGA ADK 핀 번호	2	3	21	20	19	18
Arduino UNO 핀 번호	2	3	X	X	X	X

# Interrupt

## ❖ Interrupt in Arduino

### ▪ attachInterrupt(intNo, ISR, mode)

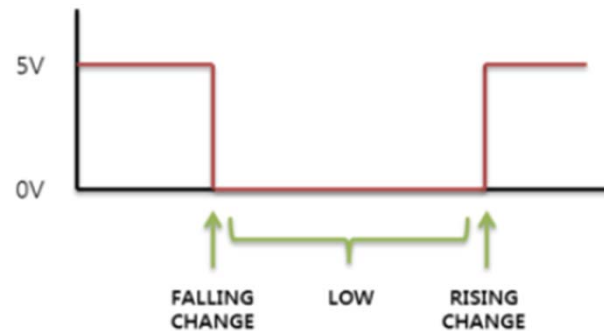
#### • ISR: Interrupt Service Routine (Interrupt Handler)

- ISR에는 인터럽트 발생 시 실행될 함수의 이름을 기재
- ISR에 사용될 함수 작성 시 유의사항

- delay()가 작동하지 않음
- 매개변수 사용이 불가
- 반환 값 없음
- 최대한 짧고 빠르게 처리하도록 구현

#### • mode: 인터럽트 발생 조건

- 인터럽트는 FALLING / RISING / CHANGE / LOW의 4가지 모드 중 하나를 통해 발생
  - FALLING: 인터럽트 핀의 신호가 HIGH -> LOW로 변경될 때 인터럽트 발생
  - RISING: 인터럽트 핀의 신호가 LOW -> HIGH로 변경될 때 인터럽트 발생
  - CHANGE: 인터럽트 핀의 신호가 HIGH -> LOW 또는 LOW -> HIGH로 변경될 때 인터럽트 발생
  - LOW: 인터럽트 핀의 신호가 LOW일 때 인터럽트 발생(LOW인 동안 반복 호출)

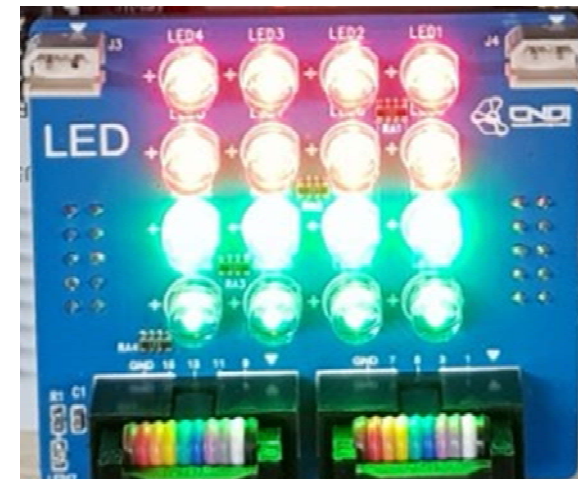
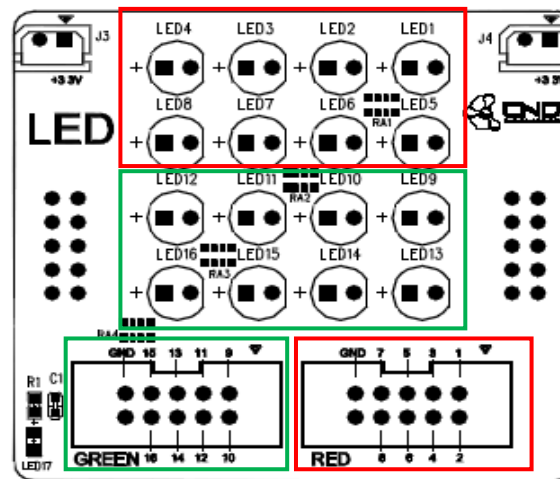


# 하드웨어 구성

## ❖ Arduino MEGA ADK



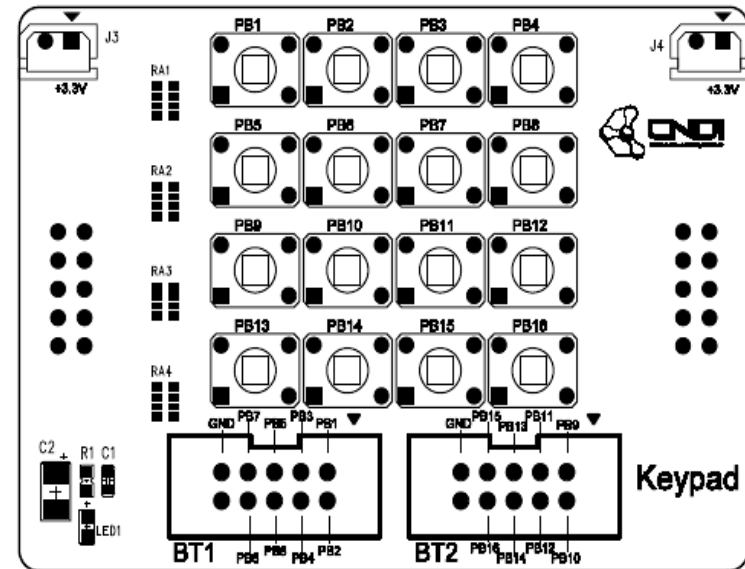
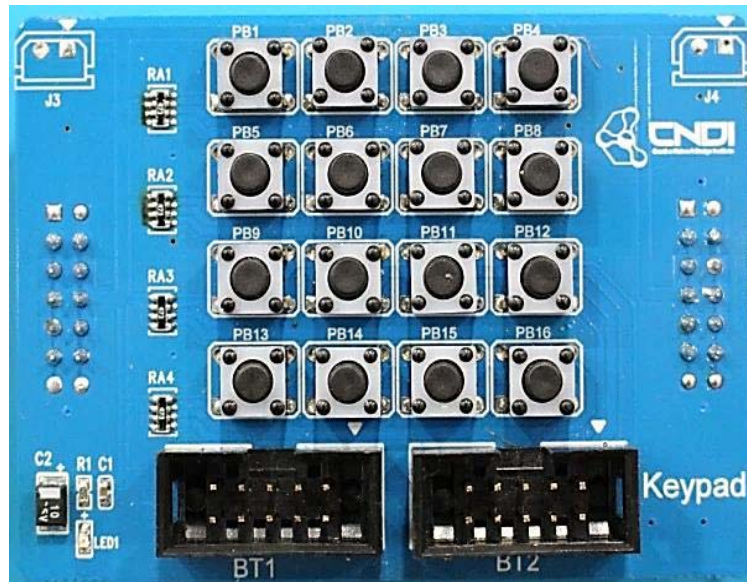
## ❖ LED 모듈





# 하드웨어 구성

## ❖ Keypad 모듈



# 하드웨어 구성



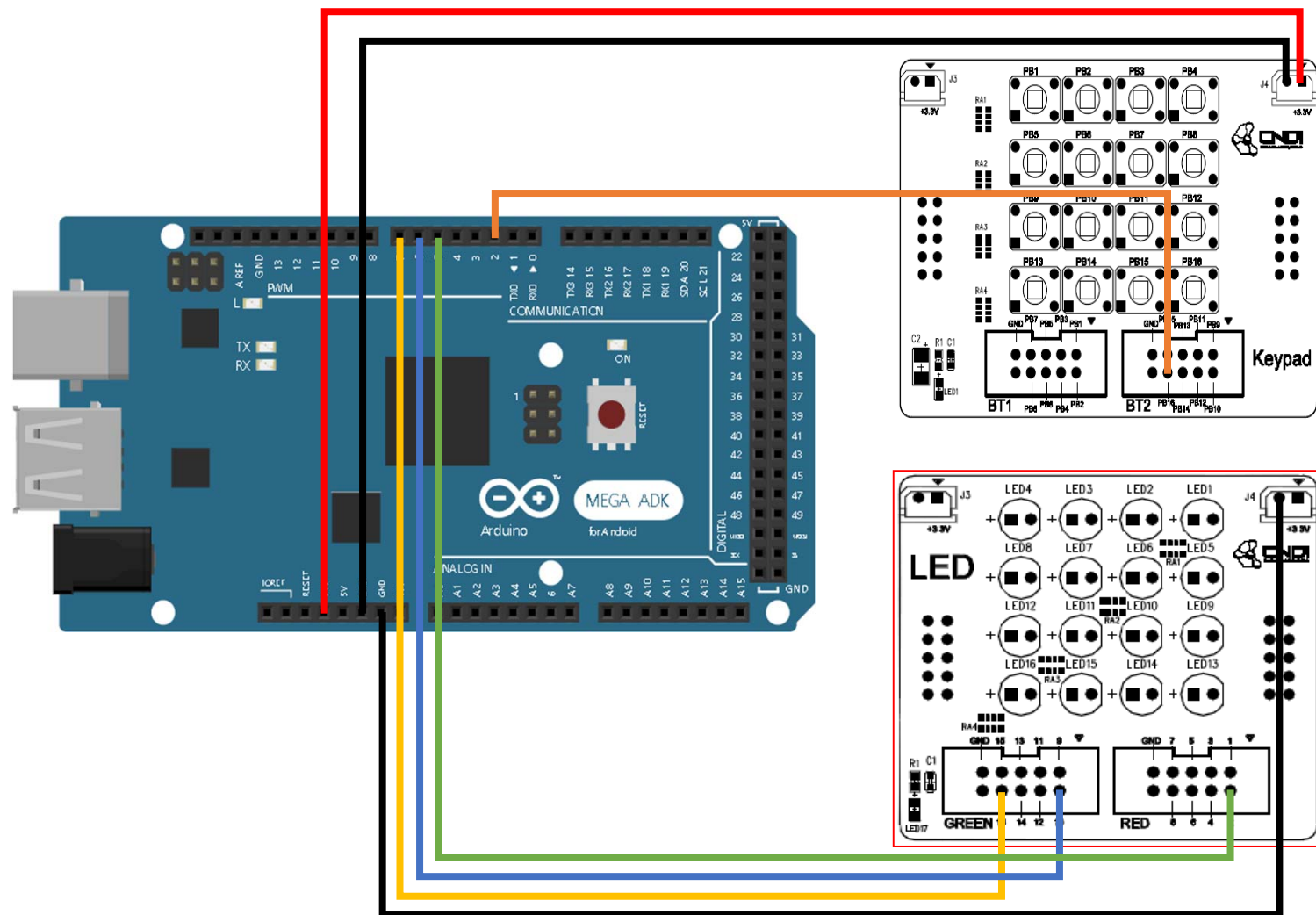
## ❖ Keypad 및 Interrupt를 활용한 LED 제어

- Keypad의 버튼을 눌렀을 때 LED가 ON/OFF
  - 디바이스는 타이머에 의해 1초 간격으로 LED 1, 9가 순차적으로 ON/OFF
  - 버튼 16을 한 번 누르면 LED 16이 ON, 다시 한 번 누르면 OFF
- 실습을 위한 Arduino MEGA ADK와 Keypad 및 LED 결선 방법
  - Arduino MEGA ADK의 DIGITAL 2와 Keypad 모듈의 BT2 포트의 핀 PB16을 연결
  - Arduino MEGA ADK의 3.3V와 Keypad 모듈의 J4 포트의 +3.3V와 연결
  - Arduino MEGA ADK의 GND(0V)와 Keypad 모듈의 J4 포트의 좌측에 연결
  - Arduino MEGA ADK의 DIGITAL 5와 LED 모듈의 RED 포트의 핀 1을 연결
  - Arduino MEGA ADK의 DIGITAL 6과 LED 모듈의 GREEN 포트의 핀 9를 연결
  - Arduino MEGA ADK의 DIGITAL 7과 LED 모듈의 GREEN 포트의 핀 16을 연결
  - Arduino MEGA ADK의 GND(0V)와 LED 모듈의 J4 포트의 좌측에 연결

# 하드웨어 구성

## ❖ Keypad 및 Interrupt를 활용한 LED 제어

### ■ 결선 그림





# 실습 예제

## ❖ Keypad 및 Interrupt를 활용한 LED 제어

- 스케치 작성
  - lab1-3.ino

```
#include <MsTimer2.h>

int Button = 2; // Arduino Button PIN 정의(인터럽트 0번 사용)
int Led [3] = {5, 6, 7}; // Arduino LED PIN 정의

// Timer function
void LedControl() {
    static int counter = 0; // ISR에서 사용할 배열 인덱스
    static byte output = HIGH; // LED(1, 9) ON/OFF

    // LED(1, 9) ON/OFF 수행
    digitalWrite(Led[counter++], output);

    if (counter > 1) {
        counter = 0; // 배열 인덱스 초기화
        output = ! output; // 현재와 반대 상태로 전환
    }
}

// ISR
void ButtonLedControl() {
    static byte output = HIGH; // LED(16) ON/OFF를 결정할 변수

    digitalWrite(Led[2], output); // LED(16) ON/OFF 수행
    output = ! output; // 현재와 반대 상태로 전환
}
```

```
void setup() {
    int i;

    // Button PIN을 입력으로 설정
    pinMode(Button, INPUT_PULLUP);

    // LED PIN을 출력으로 설정
    for(i=0; i<3; i++)
        pinMode(Led[i], OUTPUT);

    // 인터럽트 설정
    attachInterrupt(0, ButtonLedControl, RISING);

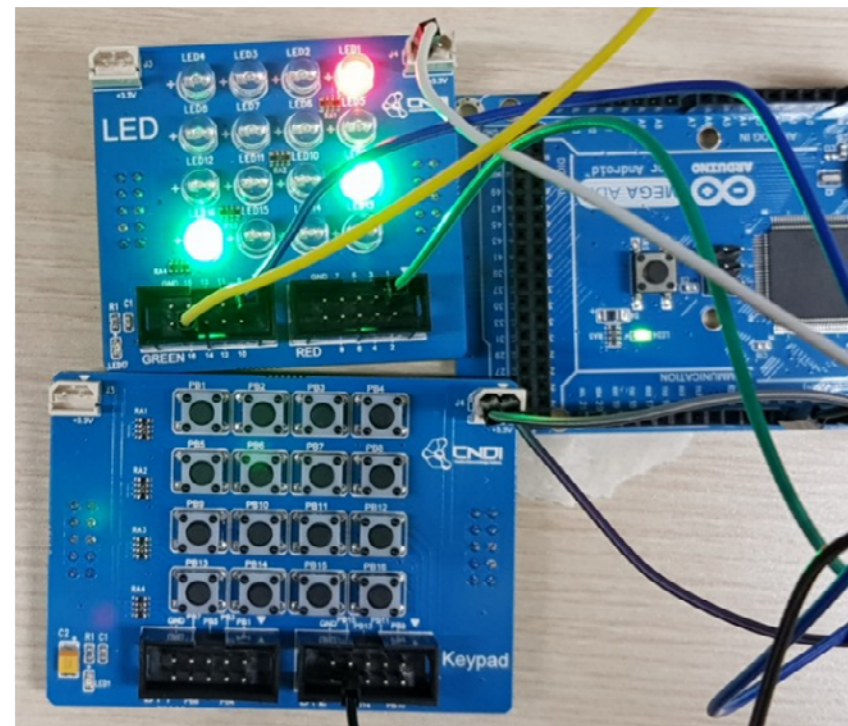
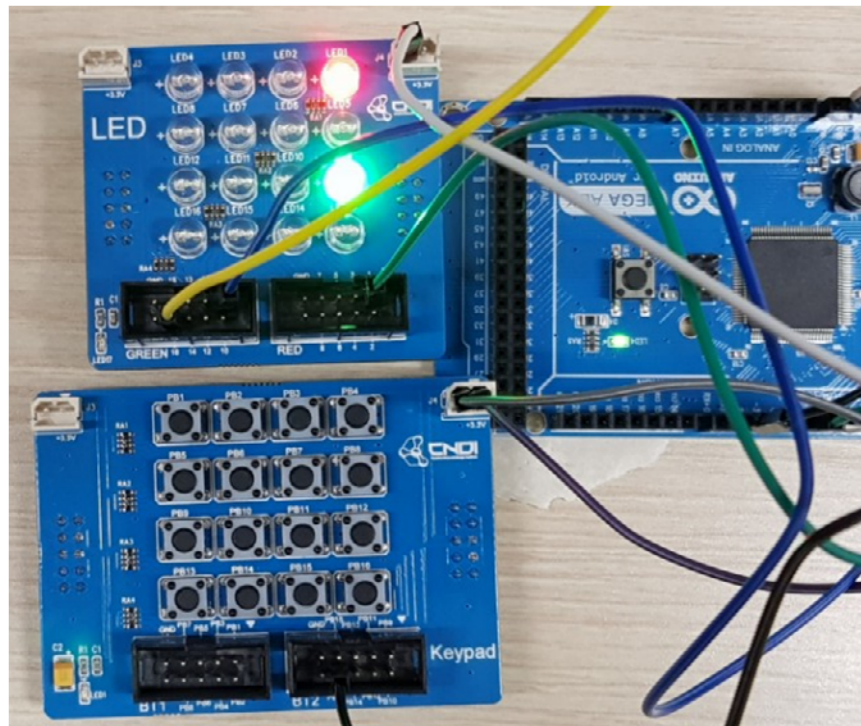
    // Timer 설정
    MsTimer2::set(500, LedControl);

    // Timer 시작
    MsTimer2::start();
}

void loop() {
}
```

# 실습 예제

- ❖ Keypad 및 Interrupt를 활용한 LED 제어
  - 동작 화면(일부)



# 실습 과제



## ❖ 1) 버튼 2개를 사용한 인터럽트 사용

- 실습 예제는 버튼 1개만 사용
- 이에 버튼 2개를 사용하여 하나는 LED 16을 ON(falling edge), 나머지 하나는 LED 16을 OFF(rising edge)하도록 코드 작성
  - ON/OFF 각각 인터럽트를 통해 동작하도록 구성

## ❖ 2) 실습과제 1)을 polling을 사용하여 다시 구현

## ❖ 참고자료) Debounce

- <https://www.arduino.cc/en/Tutorial/Debounce>



# Q & A



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