

따라 하면서 배우는

아두이노

아두이노 메가2560으로 마이크로컨트롤러 시작하기

주기적인 데이터 처리

주기적인 처리를 위한 함수

void delay(unsigned long ms)

- 매개변수
 - ms : 밀리초 단위의 지연 시간
- 반환값 : 없음

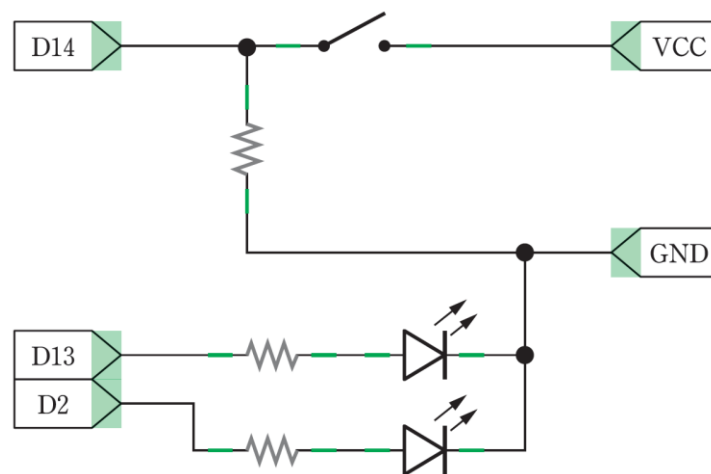
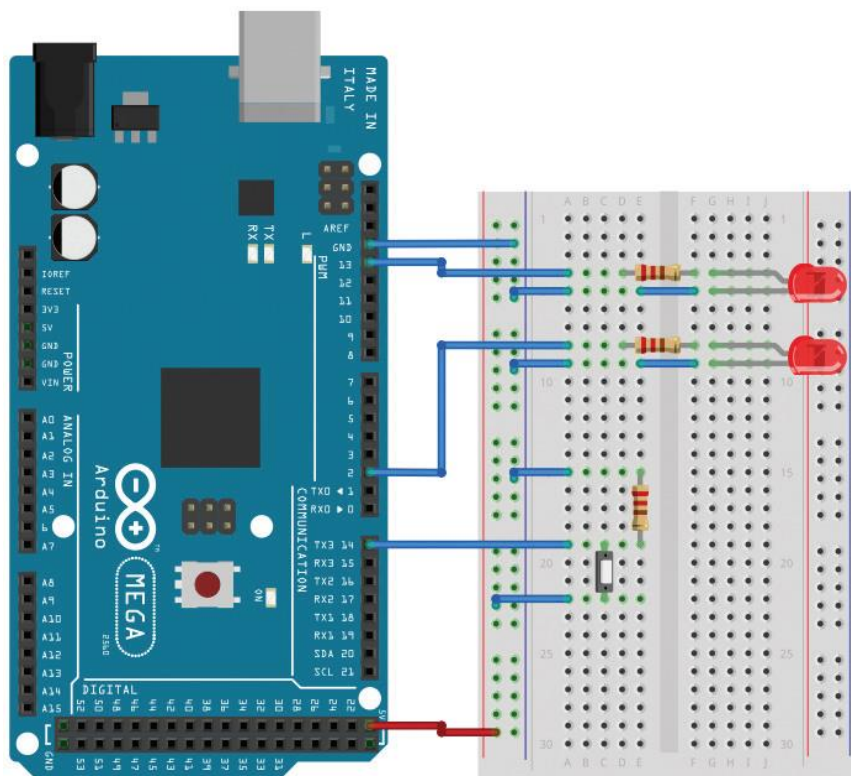
void delayMicroseconds(unsigned int us)

- us : 마이크로초 단위의 지연 시간
- 반환값 : 없음

unsigned long millis(void)

- 매개변수 : 없음
- 반환값 : 프로그램이 시작된 이후의 밀리초(millisecond) 단위 경과 시간

버튼과 LED 연결



스케치 8-1 : 주기적인 LED 제어 - delay

```
void loop() {  
    digitalWrite(pin_LED1, LED_state1);  
    delay(1000); // 1초 대기  
    LED_state1 = !LED_state1; // 13번 LED 반전  
  
    if(digitalRead(pin_button)){ // 버튼이 눌려진 경우  
        LED_state2 = !LED_state2; // 2번 LED 반전  
        digitalWrite(pin_LED2, LED_state2);  
    }  
}
```

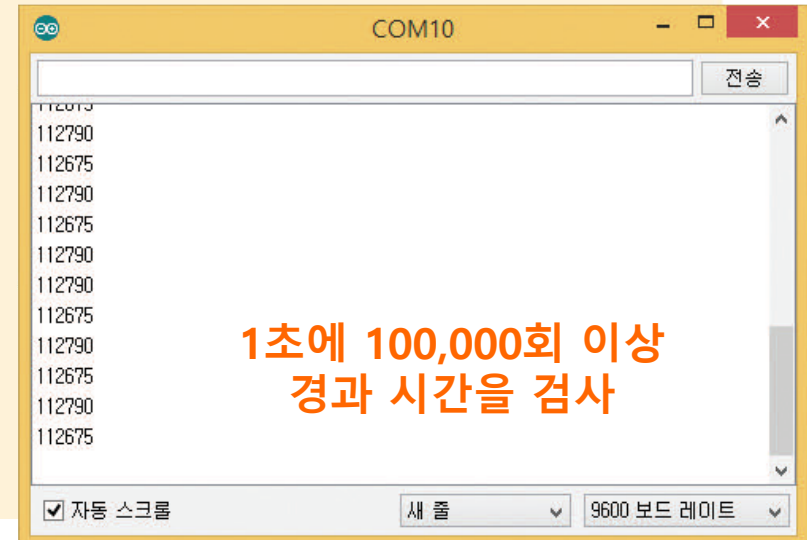
1초 대기 시간 동안 버튼의 눌림 여부를 검사하지 못하므로
2번째 LED가 버튼에 즉각적으로 반응하지 못함

스케치 8-2 : Blink

- delay 함수를 통해 LED를 반전시킨 후 1초 동안 대기
- delay 함수가 실행 중인 동안에는 대부분의 마이크로컨트롤러 동작이 중지됨
 - 스케치 8-1에서 delay 함수가 실행 중인 동안 버튼 상태를 검사하지 못함
- millis 함수는 프로그램 시작 후의 실행 시간을 반환함
 - delay 함수와 달리 실행 시간을 바로 반환하므로 다른 작업이 가능
 - 일정 시간 간격을 설정하기 위해서는 경과 시간을 계속 검사하여야 함

스케치 8-3 : delay 함수 없는 Blink

```
void loop() {  
    time_current = millis();           // 현재 시간  
    count++;                          // loop 함수 실행 횟수  
  
    // 1초 이상 시간이 경과한 경우  
    if(time_current - time_previous >= 1000){  
        time_previous = time_current; // 시작 시간 갱신  
  
        LED_state = !LED_state;  
        digitalWrite(pin_LED, LED_state);  
  
        Serial.println(count);  
        count = 0;  
    }  
}
```



스케치 8-4 : millis 함수로 스케치 8-1 수정

```
void loop() {
```

```
    time_current = millis();
```

```
    if(time_current - time_previous >= 1000){
```

```
        time_previous = time_current;
```

```
        LED_state1 = !LED_state1;
```

```
        digitalWrite(pin_LED1, LED_state1);
```

```
    }
```

첫 번째 LED의 blink

// 시작 시간 갱신

// LED 반전

```
    if(digitalRead(pin_button)){
```

```
        LED_state2 = !LED_state2;
```

```
        digitalWrite(pin_LED2, LED_state2);
```

```
        delay(100);
```

```
    }
```

```
}
```

// 버튼이 눌려진 경우

// 2번 LED 반전

스케치 8-5 : 버튼을 누르는 순간 감지

- 스케치 8-4의 경우 버튼을 누르고 있으면 계속 LED가 반전됨
- 버튼의 이전과 현재 상태를 저장하여 상태가 바뀌는 순간에만 LED를 반전

```
button_state_current = digitalRead(pin_button);  
if(button_state_current){                                     // 버튼이 눌러진 경우  
    if(button_state_previous == false){  
        button_state_previous = true;  
        LED_state2 = !LED_state2;                           // 2번 LED 반전  
        digitalWrite(pin_LED2, LED_state2);  
    }  
}  
else{  
    button_state_previous = false;  
}
```


Thank you!!

