
타이머, 카운터

실습1: 500ms초마다 LED 토글

❖ ex01/app.ino

```
#include <Led.h>

Led led(8);
boolean led_val = LOW;

void setup() {
}

void loop() {
    delay(500);           // 500ms 지연
    led_val = !led_val;   // LED 출력 변수 반전
    led.setValue(led_val);
}
```

타이머, 카운터

아두이노 보드				ATmega328P MCU 핀 번호 및 포트명	관련 타이머/카운터
아두이노 핀 명	디지털 입출력 핀	PWM 출력 핀	PWM 파형 주파수		
오른쪽	13	13		19. PB5/SS	
	12	12		18. PB4/MISO	
	~11	11	~11 490Hz	17. PB3/MOSI/OC2A	Timer2
	~10	10	~10 490Hz	16. PB2/SS/OC1B	Timer1
	~9	9	~9 490Hz	15. PB1/OC1A	Timer1
	8	8		14. PB0	
	7	7		13. PD7	
	~6	6	~6 980Hz	12. PD6/OC0A	Timer0
	~5	5	~5 980Hz	11. PD5/OC0B	Timer0
	4	4		6. PD4	
	~3	3	~3 490Hz	5. PD3/INT1/OC2B	Timer2

타이머, 카운터

❖ 타이머

- Timer0
 - 8비트 타이머
 - delay(), millis(), micros()에서 사용

- Timer1
 - 16비트 타이머
 - 우노보드의 경우 서보모터 라이브러리(Servo) 함수에서 사용
 - Timer1 인터럽트 사용 시에는 서보 모터 관련 함수 사용 불가
 - D9핀과 D10핀의 PWM 출력 기능도 사용 불가

- Timer2
 - 8비트 타이머
 - tone() 함수 등에 사용
 - Timer2 인터럽트 사용 시에는 tone() 함수 사용불가
 - D3핀과 D11핀의 PWM 출력 기능 사용 불가

타이머, 카운터

❖ Timer2 라이브러리

○ 라이브러리 매니저로 MsTimer2 검색 후 설치

MsTimer2::set(unsigned long ms, void (*f)())		
기능	타이머2의 인터럽트 주기를 ms 단위로 설정하며, 설정된 시간이 경과되면 함수 f가 실행된다. 따라서, 설정된 시간마다 계속 함수 f가 실행된다.	
매개변수	ms	ms 단위의 주기 값(unsigned long)
	f	호출되는 인터럽트 함수
리턴 값	없음	

MsTimer2::start()		
기능	타이머2 인터럽트 동작 시작	
매개변수	없음	
리턴 값	없음	

MsTimer2::stop()		
기능	타이머2 인터럽트 동작 금지	
매개변수	없음	
리턴 값	없음	

실습2: 타이머2 인터럽트 이용 LED 1초마다 토글

❖ ex02/app.ino

```
#include <Led.h>
#include <MsTimer2.h>

boolean led_val = LOW;
Led led(8);

void setup() {
    // 타이머2 인터럽트 설정 및 동작 시작
    MsTimer2::set(500, flash); // 500ms 인터럽트 주기 설정, flash() 함수 실행
    MsTimer2::start();         // 타이머2 동작 시작
}

void loop() {
}

// 타이머2 인터럽트 함수, 500ms 마다 자동 실행
void flash()
{
    led_val = !led_val;
    led.setValue(led_val);
}
```

타이머, 카운터

❖ 타이머1 인터럽트 라이브러리

- 라이브러리 매니저에서 TimerOne 검색 후 설치
 - <https://playground.arduino.cc/Code/Timer1/>

void Timer1.initialize(unsigned long period)		
기능	타이머1을 초기화하며, 타이머1 라이브러리에서 제공하는 함수들을 사용하기 위해서는 이 함수를 우선 실행하여야 한다. us 단위로 타이머의 인터럽트 주기를 지정할 수 있다. (PWM 출력핀 ~9, ~10의 PWM 출력기능은 사용할 수 없다)	
매개변수	period	us 단위의 주기 값(unsigned long)
리턴 값	없음	
void Timer1.start(void)		
기능	타이머1 동작 시작	
매개변수	없음	
리턴 값	없음	
void Timer1.stop(void)		
기능	타이머1 동작 정지	
매개변수	없음	
리턴 값	없음	

타이머, 카운터

❖ 타이머1 인터럽트 라이브러리

void Timer1.restart(void)	
기능	타이머1 다시 시작(타이머 값은 리셋)
매개변수	없음
리턴 값	없음
void Timer1.resume(void)	
기능	타이머1 정지된 동작 다시 시작(타이머 값은 그대로)
매개변수	없음
리턴 값	없음
unsigned long Timer1.read(void)	
기능	us 단위 시간 타이머1 값 읽기
매개변수	없음
리턴 값	us 단위의 시간 값

타이머, 카운터

❖ 타이머1 인터럽트 라이브러리

void Timer1.setPeriod(long period)		
기능	us 단위의 새로운 인터럽트 주기 설정(1us~8.3s)	
매개변수	period	us 단위의 주기 값(unsigned long)
리턴 값	없음	

void Timer1.attachInterrupt(void(*ISR)()) void Timer1.attachInterrupt(void(*ISR)(), unsigned long period)		
기능	us 단위로 지정된 시간마다 인터럽트 함수(ISR) 호출	
매개변수	ISR	호출되는 인터럽트 함수
	period	us 단위의 주기 값(unsigned long)
리턴 값	없음	

void Timer1.detachInterrupt()		
기능	Timer1 인터럽트 해제	
매개변수	없음	
리턴 값	없음	

실습3: LED 1초마다 토글

❖ ex03/app.ino

```
// 타이머1 인터럽트 이용
#include <Led.h>
#include <TimerOne.h>

boolean led_val = LOW;
Led led(8);

void setup() {
    // 타이머1 인터럽트 설정 및 동작 시작
    Timer1.initialize(500000); // 500ms 주기 설정
    Timer1.attachInterrupt(flash); // 타이머1 인터럽트 함수 지정
}

void loop() {
}

// 타이머1 인터럽트 함수, 500ms 마다 자동 실행
void flash()
{
    led_val = !led_val;    // LED 출력 변수 반전
    led.setValue(led_val);
}
```

SimpleTimer

SimpleTimer

❖ SimpleTimer

- 라이브러리 매니저에서 SimpleTimer 검색 후 설치
- millis()를 기반으로 동작 → 1mS 해상도를 가짐
- polling 방식 사용
 - loop() 함수에서 run() 메서드를 호출
 - 콜백함수가 실행되었을 때 정확한 동작을 보장하지 않음
 - SimpleTimer 한번 실행 시 5mS가 소요
 - 인터벌을 2mS로 지정하면 제대로 작동되지 않음
 - 인터벌 타이머로 사용시에는 최소 5mS 이상의 인터벌 설정이 필요

SimpleTimer

❖ 주요 메서드

- SimpleTimer()
 - 생성자
- int setInterval(long d, timer_callback f)
 - d 밀리초 마다 함수 f를 호출하도록 등록
 - f의 signature: void f()
 - Timer ID 리턴
- int setTimeout(long d, timer_callback f)
 - d 밀리초 후에 함수 f 호출
 - 한 번만 호출
- int setTimer(long d, timer_callback f, int n)
 - d 밀리초 마다 함수 f를 호출하되, n번 호출 이후 삭제됨
- void enable(int timerId)/boolean isEnabled(int timerId)
- void disable(int timerId)
- void toggle(int timerId)
- void restartTimer(int timerId)
- void deleteTimer(int timerId)
- int getNumTimers()

SimpleTimer

❖ ex04/app.ino

```
#include <SimpleTimer.h>

SimpleTimer timer;

// callback 함수
void repeatMe() {
    Serial.print("Uptime (s): ");
    Serial.println(millis() / 1000);
}

void setup() {
    Serial.begin(9600);
    timer.setInterval(1000, repeatMe);
}

void loop() {
    timer.run();
}
```