# 2. 아두이노 타이머 인터럽트 : MsTimer2

0.5초 마다 센서의 값을 시리얼 통신으로 송신, 0.1초 마다 현재 센서 값을 저장하는 등, 주기적인 시간마다 수행하는 코드는 타이머 인터럽트를 사용하면 쉽게 구현이 가능합니다. 아두이노는 타이머 인터럽트를 위한 MsTimer2 라이브러리를 지원합니다. MsTimer2 라이브러리는 마이크로프로세서 전문 지식이 없이도 간단한 방법으로 특정 시간 간격으로 코드를 실행 할 수 있는 함수를 제공합니다. 본 포스트에서는

## 라이브러리 설치 방법, MsTimer2 라이브러리 사용 방법과 예제 에 대해 작성하였습니다.

### 아두이노 타이머

Arduino Uno Board에 사용되는 메인 컨트롤러는 Atmega328입니다. Atmega328를 3개의 타이머를 가지고 있습니다. 아두이노는 이 타이머 3개를 가공하여, PWM 출력 및 시간 관련 함수들을 제공합니다.

* Timer0 - 8Bit : PWM 5/6핀, 시간 관련 함수 (delay(), millis() 등)
* Timer1 - 16Bit : PWM 9/10핀 👉 외부 라이브러리 Servo.h 와 함께 사용 불가
* Timer2 - 8Bit : PWM 3/11핀 👉 외부 라이브러리 MsTimer2.h 와 함께 사용 불가

### 타이머 인터럽트 사용 방법

아두이노 라이브러리 매니저에서 제공되는 MsTimer2 라이브러리를 사용하면, 손쉽게 타이머 인터럽트를 구성 할 수 있습니다. MsTimer2 라이브러리에서 Timer Interrupt 와 관련된 함수는 아래와 같습니다.

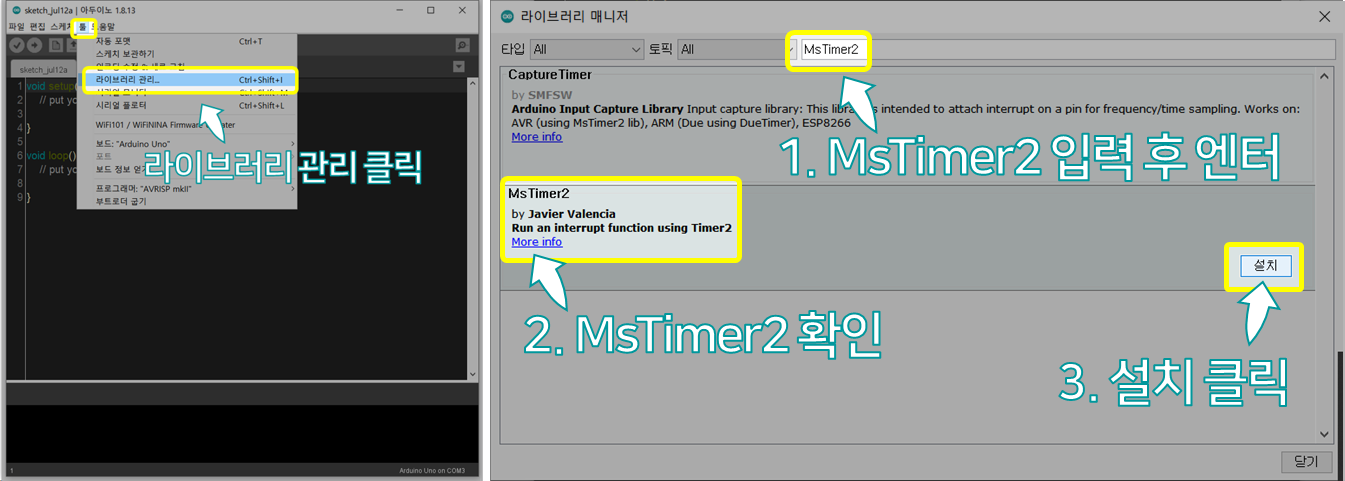
* MsTimer2::set() 타이머 인터럽트 발생 시간 설정 - 최대 약 49.7日
* MsTimer2::start() 타이머 인터럽트 활성화
* MsTimer3::stop() 타이머 인터럽트 비활성화

❗ 주의사항

MsTimer2는 타이머 인터럽트를 구성하는 경우 3번핀, 9번핀은 PWM 출력 기능으로 사용하지 못합니다.

#### **가. MsTimer2 라이브러리 설치 방법**

MsTiemr2 라이브러리는 Arduino IDE 의 라이브러리 매니저를 사용하면 쉽게 설치가 가능합니다. 설치 방법은 아래의 그림을 참고해주세요.



#### 

#### **나. MsTimer2::set()**

MsTimer2::set() 함수는 타이머 인터럽트 발생 시간과, 타이머 인터럽트 발생시 실행 될 함수명을 설정합니다.

Syntax

* MsTimer2::set(unsinged long ms, void (\*f)())

Parameters

* ms : 타이머 인터럽트를 발생 시킬 ms 단위의 시간 (Datatype : unsigned long)
* void (\*f)() : ms 시간 간격으로 실행 될 함수 명 - 반환 값이 없는 함수

Returns : 없음

Example

* MsTimer2::set(100, timerIRQ) : 100ms 마다 void timerIRQ() 함수를 실행하도록 설정

#### **다. MsTimer2::start()**

MsTimer2::start() 함수는 타이머 인터럽트를 활성화 합니다. 이 코드가 작성된 시간일 기준으로 시간 측정을 진행 합니다.

Syntax

* MsTimer2:start()

#### **라. MsTimer2::stop()**

MsTimer2::stop() 함수는 타이머 인터럽트를 비활성화 합니다.

#### **마. MsTimer2::stop()**

MsTimer2::stop() 함수는 타이머 인터럽트를 비활성화 합니다.

Syntax

* MsTimer2::stop()

### 타이머 인터럽트 써먹기

#### 예제 : 아두이노 내장 보드 점멸

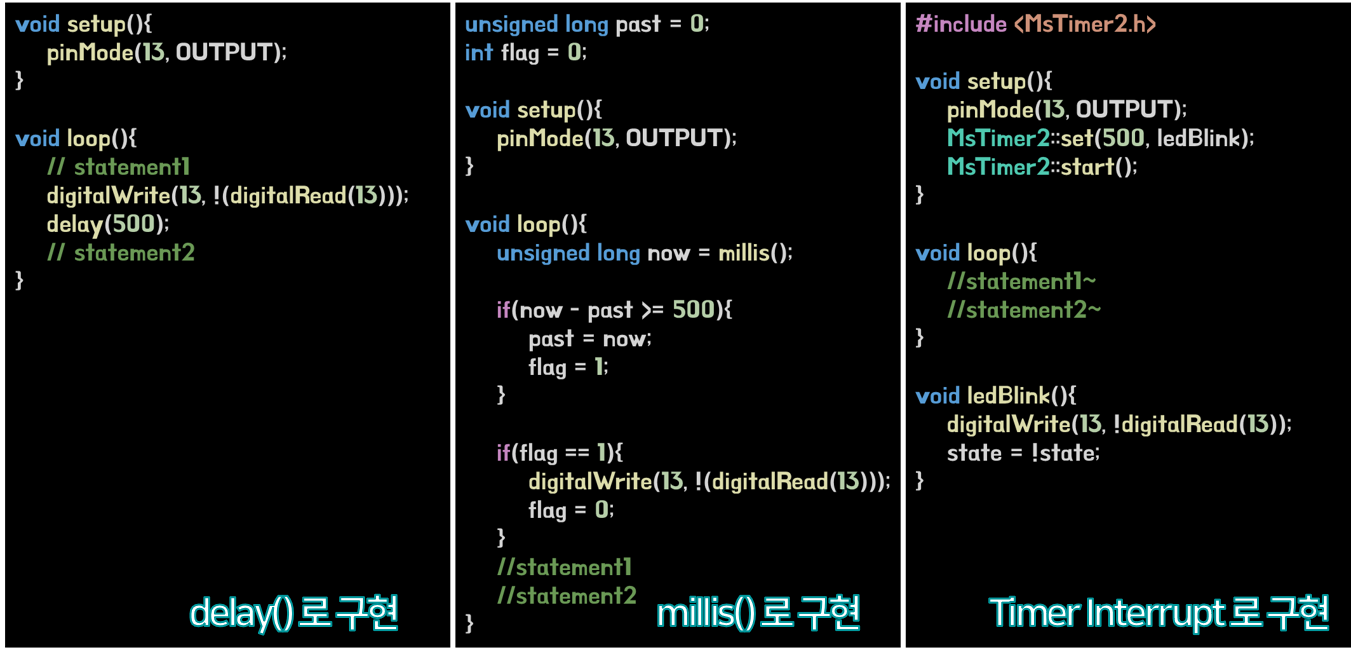
아래는 500ms 간격으로 Arduino Uno 보드 내장 LED 를 점멸 하는 코드와 실행 결과 입니다.

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | #include <MsTimer2.h>    const int ledPin = 13;  volatile boolean state = true;    void setup(){  pinMode(ledPin, OUTPUT);  MsTimer2::set(500, ledBlink);  MsTimer2::start();  }    void loop(){  //statement1~  //statement2  }    void ledBlink(){  digitalWrite(ledPin, state);  state = !state;  } |  |
| --- | --- | --- |

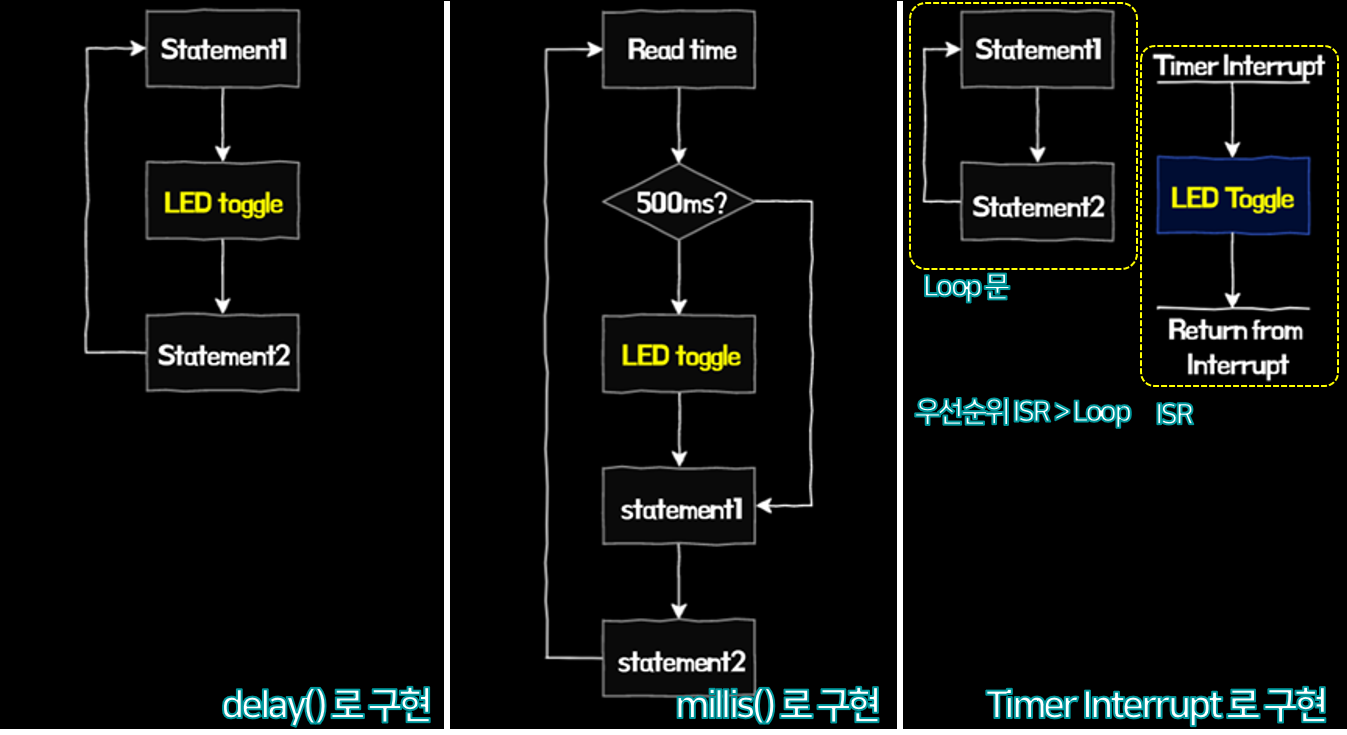
#### 

#### blink 예제 비교

아래의 3가지 코드는 500ms 마다 LED 가 점멸 하는 코드 입니다. 각각 delay(), millis(), Timer Interrupt 로 구성된 코드 입니다. LED 를 점멸하는 한 가지 기능만 있는 경우 모두 동일하게 동작하겠지만, 다른 기능들이 추가되는 경우 어떤 방식으로 구현 하는 것이 좋을지 선택이 필요합니다.



LED blink 코드 구현 방법 3가지



코드 종류별 순서도

* delay() 로 구현
  + delay() 는 시간 지연을 하는 함수로, 여러가지 기능 구현하는 코드 작성에는 지양해야 할 함수
* millis() 로 구현
  + statement1 와 statement2 의 수행 시간에 따라 LED toggle 기능 약간의 지연이 발생 할 수 있음
  + Timer2를 다른 용도로 사용해서 MsTimer2를 사용하지 못하는 경우 대안이 가능
* Timer Interrupt로 구현
  + 타이머 인터럽트 처리 문이 Loop() 문의 기능보다 우선순위가 높아 500ms 간격으로 LED toggle 가능

### 더 세분화된 타이머 설정

MsTimer2 라이브러리는 최소 ms 단위의 시간으로 타이머 인터럽트 발생이 가능합니다. ms시간 단위보다 더 세분화된 타이머 설정이 필요하면 [1](https://juahnpop.tistory.com/89#footnote_89_1)FlexiTimer2.h 라이브러리를 사용하는 방법이 있습니다. FlexiTiemr2.h 를 사용하면 최대 1/3ms 단위로 타이머 설정이 가능합니다.