

Be Artists of Robotics &  
Advanced Micro Intelligence

바라미

Since 1994



# 바라미 아두이노 세미나

2-1. tone함수와 시간제어함수

김정현

kimdictor@gmail.com

박제윤

jeyunp@hanyang.ac.kr

# 목차

- 2차시 리마인드
- tone() 함수
- delay() 함수
- millis() 함수

# 2차시 리마인드

단순 상태

펄스

직렬 신호

입력 (I)

digitalRead()

pulseIn()

shiftIn()

출력 (O)

digitalWrite()

tone()

shiftOut()

# tone() 함수 : 펄스 출력 시작

## 정의

- `tone(pin, frequency)`
- `tone(pin, frequency, duration)`

## 매개변수

- `int pin` : 핀 번호
- `unsigned int frequency` : 주파수 (Hz)
- `unsigned long duration` : 지속시간 (ms)

## 세부사항

AVR에서 31~65535Hz, Due 구현 없음  
AVR에서 PWM D3, D15 간섭

# noTone() 함수 : 펄스 출력 중단

정의 ● noTone(pin)

매개변수 ● `int pin` : 핀 번호

세부사항

# Due에서의 대안

단순히 반복문과 digitalWrite 조합

```
bool enable = true;
unsigned int freq = 100;
while(enable) {
    digitalWrite(11, HIGH);
    delayMicroseconds(1000000 / freq * 2);
    digitalWrite(11, LOW);
    delayMicroseconds(1000000 / freq * 2);
}
```

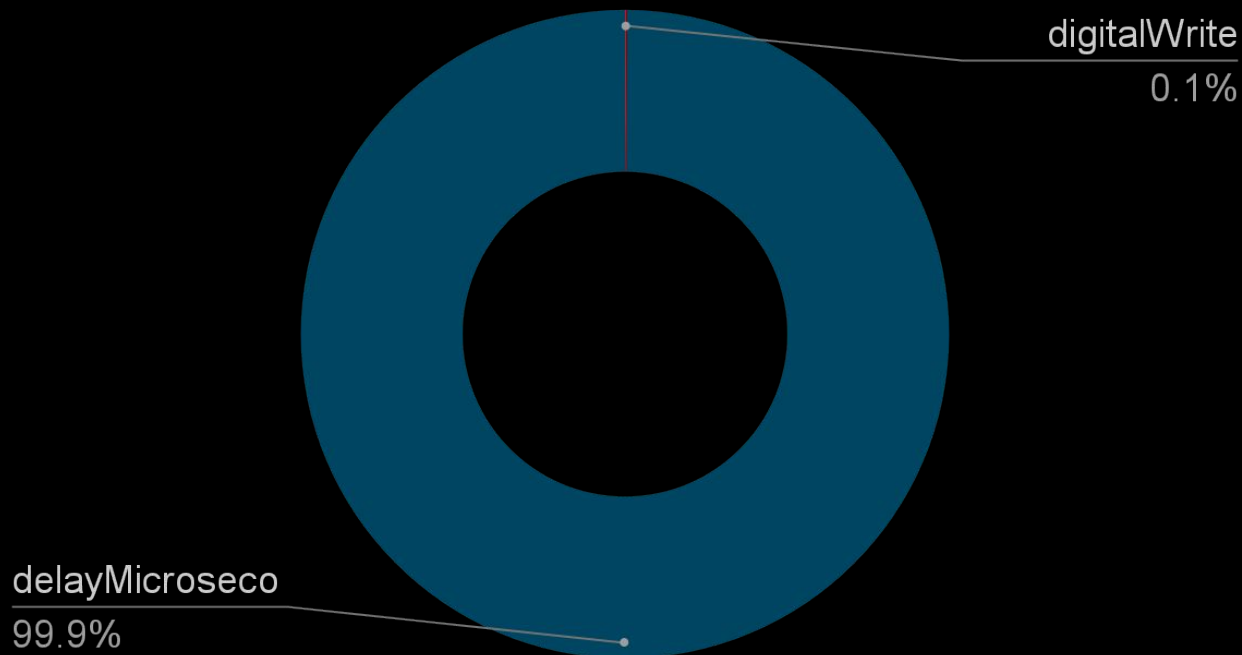
탈출 방법?

# Due에서의 대안

1초동안 명령어의 CPU 점유 비율

1초 = 1000000마이크로초  
평균 소요시간 digitalWrite = 2.0us

점유 시간 비율



99.9%는

노는 CPU

# Due에서의 대안 2

millis()와 조건문 사용

```
now = micros();  
if (now - last > 5000) {  
    status = status == HIGH ? LOW : HIGH  
    digitalWrite(11, status);  
    last = now;  
} else {  
    // other task
```



남는 시간에 다른 작업!



# Due에서의 대안 2

## 문제점

- 프로그램 구조 복잡성과 boilerplate 증가
- 정확성을 보증할 수 없음 (다른 ISR에 의한)
- 경험적으로 테스트가 필요한 시간 값

# Due에서의 대안 3

OS에게 맞기거나

타이머 인터럽트한테 맡기자!

6차시에 봐요~

# delay() 함수 : 정의

## 정의

- `delay(ms)`
- `delayMicroseconds(us)`

## 매개변수

- `unsigned long` `ms` : 멈출 밀리초 시간
- `unsigned int` `us` : 멈출 마이크로초 시간

## 세부사항

$3 < us < 16383$ 에서 정확하게 동작

# delay() 함수 : 구현

```
void delay( unsigned long ms )
{
    if (ms == 0)
        return;
    uint32_t start = GetTickCount();
    do {
        yield();
    } while (GetTickCount() - start < ms);
}
```

```
extern uint32_t GetTickCount( void )
{
    return _dwTickCount ;
}
```

In-circuit Emulator	
Cortex-M3 Processor f <sub>max</sub> 84 MHz	24-bit SysTick Counter
MPU	N V I C

# millis() 함수 : 런타임 측정

## 정의

- `time = millis()`
- `time = micros()`

## 반환 값

- `unsigned long time` : 프로그램 시작 후 경과 시간

## 세부사항

반환 값을 할당한 변수 형에 주의,  
오버플로우가 발생할 수 있음

# millis() 함수 : 구현

```
unsigned long millis( void )
{
// todo: ensure no interrupt
    return GetTickCount() ;
}
```

# 질문은 완전 환영이에요 : Q&A

- 카카오톡 (아두이노 스터디 단톡방 통해 연락주세요!)
- E-mail (다음 슬라이드에 주소 있습니다!)
- 바라미 디스코드 (아무 바라미 회원한테나 물어보시면 됩니다.)
- Zoom (강의자한테 연락 주세요~)
- 실습날에 물어보셔도 됩니다!!

# 감사합니다!!

## 강의자

김정현

[kimdicator@gmail.com](mailto:kimdicator@gmail.com)

박제윤

[jeyunp@hanyang.ac.kr](mailto:jeyunp@hanyang.ac.kr)

이 발표자료는 2021년 한양공대 학술소모임 “바라미” 아두이노 세미나 발표자료이며 강의자가 저작권을 소유합니다.

더 많은 정보는 바라미 유튜브 채널과 바라미 홈페이지(<https://ibarami.com>)를 참고하세요.

이 발표자료는 CC BY-SA (저작자표시-동일조건변경허락)에 따라 이용할 수 있으며 포함된 저작물은 각 저작권자의 소유입니다.

