

아두이노 고급 프로그래밍

Professor H.J. Park, Dept. of Mechanical System Design, Seoul National University of Science and Technology.

An Unmanned aerial vehicle (UAV) is a Unmanned Aerial Vehicle. UAVs include both autonomous (means they can do it alone) drones and remotely piloted vehicles (RPVs).





CONTENTS

01 객체지향 프로그래밍

객체지향 프로그래밍에 대하여 알아보자.

02 아두이노와 Interrupt

Interrupt를 이용한 아두이노 프로그래밍

03 Serial 통신 고급주제

Serial 통신에서 고려되어야 할 사항들에 대하여 살펴본다.

04 Interrupt 이용 Serial 통신

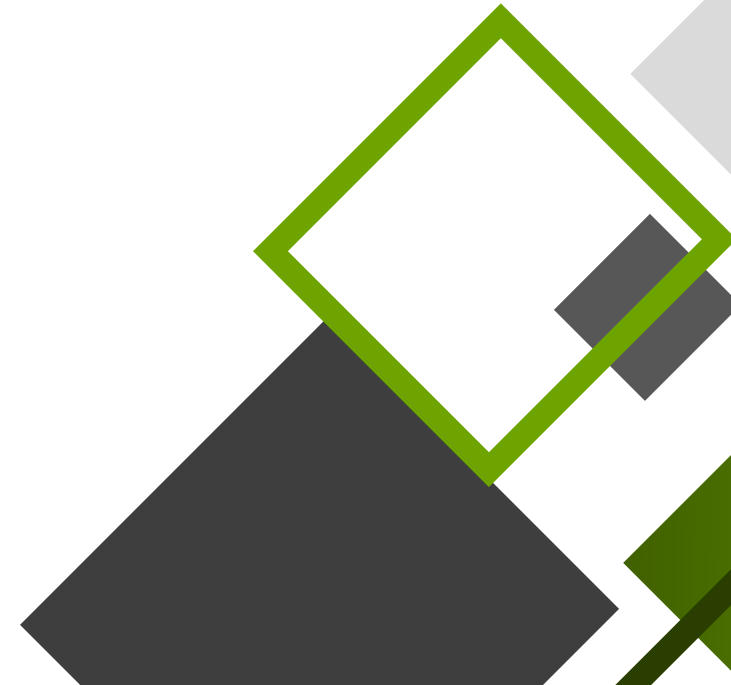
NeoHWSerial 라이브러리 사용 통신에 대하여 살펴본다.

05 Binary 전송

Binary 전송 방법에 대하여 살펴본다.



객체지향 프로그래밍



C++ 개요

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

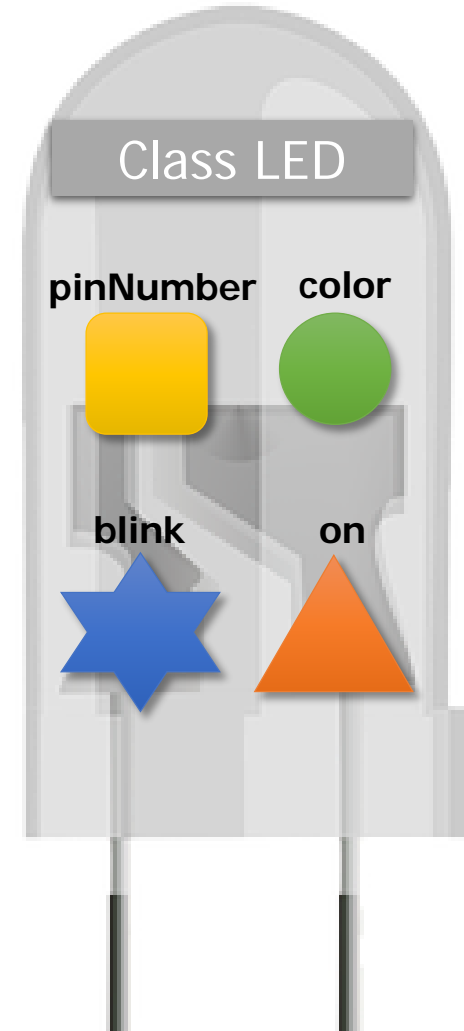
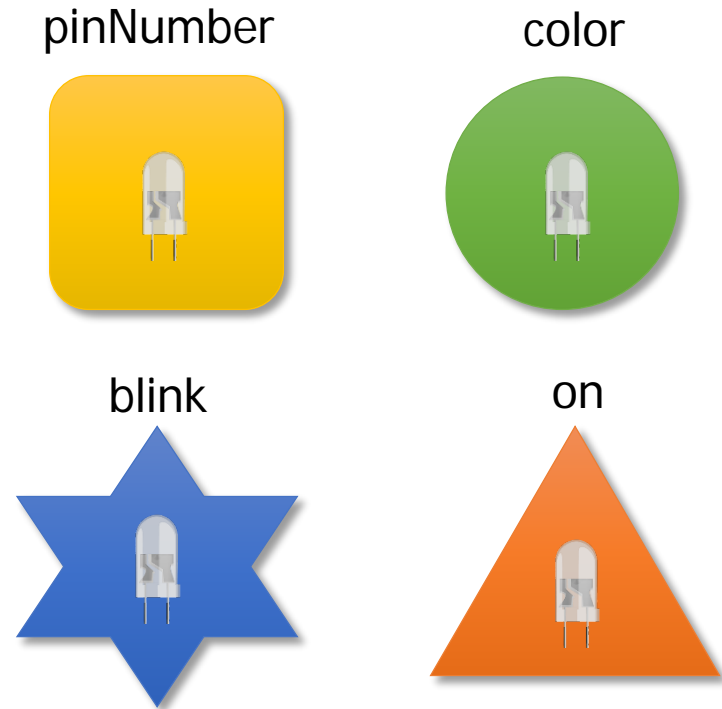
- C++ 은 Arduino, 스마트폰, PC에 등에 널리 사용되는 언어
 - C 언어를 포함.
- 객체지향적 (OOP: Object Oriented Programming) 특징
 - Encapsulation:
 - 데이터와 함수를 클래스에 포함시킨다.
 - Data hiding:
 - 함수를 통하여 데이터에 접근 (private, public ...)
 - Inheritance:
 - 기존의 클래스의 데이터나 함수를 상속받아 더욱 쉽게 기능 확장
 - Polymorphism:
 - 상속에서 연산자/함수의 overloading 을 통하여 더욱 다양한 기능이 가능하도록 한다.

객체지향 클래스

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

■ 물체(객체: Object)가 중심!

- 복잡하게 따로 있던 것들을 ➔ 객체로



Class의 개념

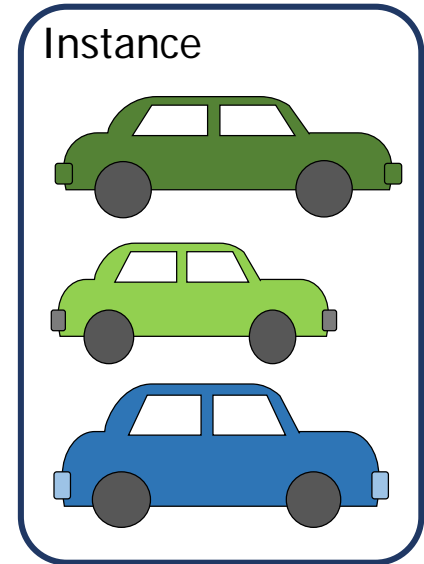
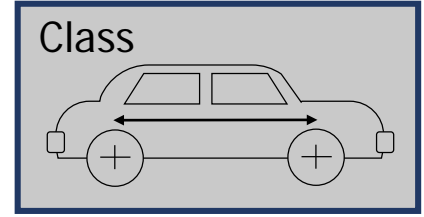
Dept. of Mechanical System Design, Seoul National University of Science and Technology.

■ Class 란?

- Class는 설계도, 금형의 의미
- Object(Instance)는 상품이다.
- Class 는 하나지만 Instance 는 많을 수 있다.

■ Class 의 구성

- 데이터(property, 속성)
 - 데이터와 함수를 클래스에 포함시킨다.
- 함수(method):
 - 함수를 통하여 데이터에 접근 (private, public ...)



▶ Arduino LED Class 만들기

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

■ LED의 기본 클래스 LEDbasic 정의

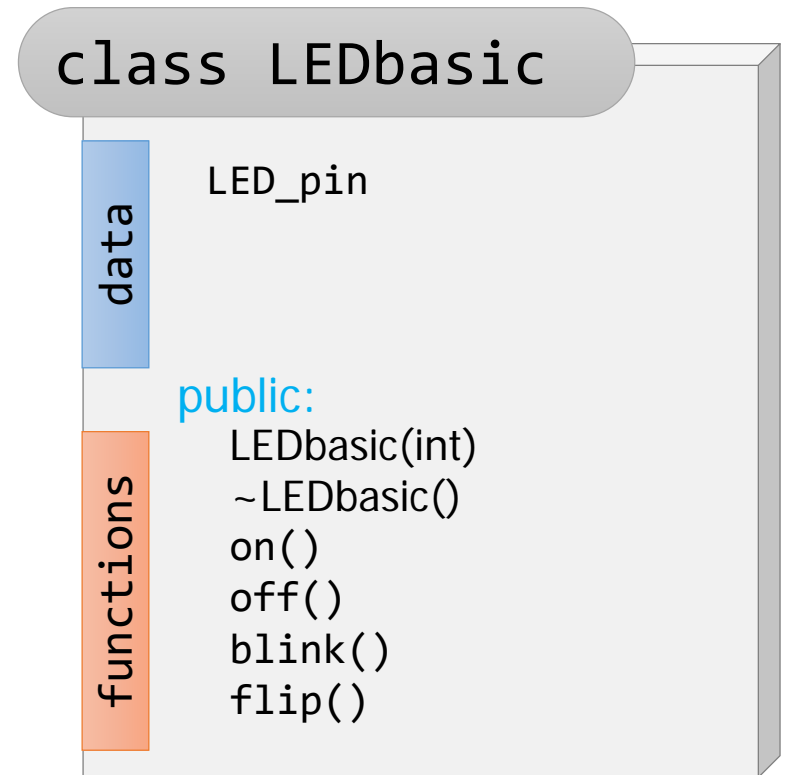
■ 멤버

▶ Property

- ▶ LED_pin: Arduino pin number of LED

▶ Method

- ▶ LEDbasic(int) : 생성자
- ▶ ~LEDbasic(): 소멸자
- ▶ on(): 켜기
- ▶ off(): 끄기
- ▶ flip(): 상태 반전
- ▶ void blink(int time): time 기간 마다 blinking

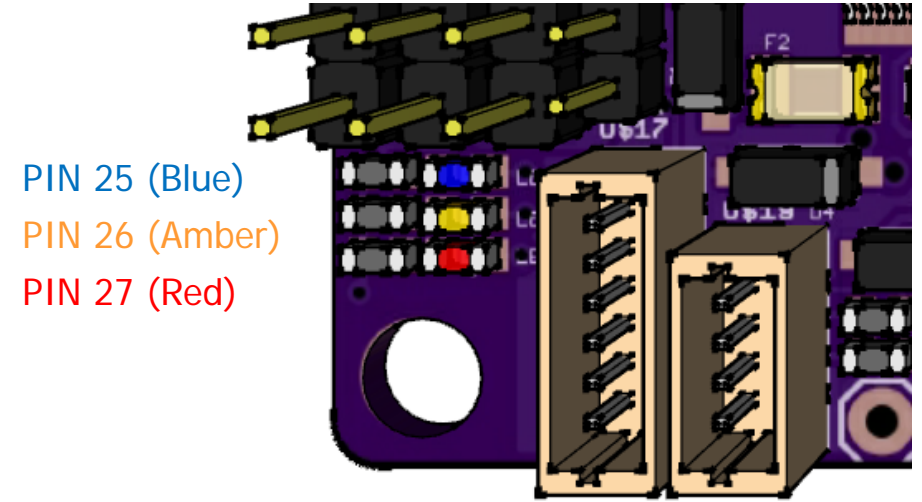


APM의 LED

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

■ APM의 LED

- 청, 오렌지, 적색 3개의 LED 가 있다.
- 각각 25, 26, 27 번 핀에 연결되어 있다.
- 단, LOW 이면 켜지고 HIGH 이면 꺼진다.



Arduino LED Class 만들기

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

■ LEDbasic Class 코드

- .h 파일
- .cpp 파일

LEDbasic.h

```
#ifndef LEDBASIC_H    // Include guard
#define LEDBASIC_H
#include <Arduino.h>  // for C++
class LEDbasic {
    int LED_pin;
public:
    LEDbasic(int);
    ~LEDbasic();
    void on();
    void off();
    void blink(int);
    void flip();
};
#endif
```

LEDbasic.cpp

```
#include "LEDBasic.h"
LEDbasic::LEDbasic(int pin){    //constructor
    LED_pin=pin;
    pinMode(LED_pin, OUTPUT);
}
LEDbasic::~LEDbasic(){/*nothing*/} //destructor
void LEDbasic::on(){
    digitalWrite(LED_pin, LOW);
}
void LEDbasic::off(){//turn the LED off
    digitalWrite(LED_pin, HIGH);
}
void LEDbasic::flip(){//turn the LED off
    digitalWrite(LED_pin,!digitalRead(LED_pin));
}
void LEDbasic::blink(int time){//blink the LED
    on();
    delay(time/2);    //wait half of period
    off();
    delay(time/2);
}
```

▶ Arduino LED Class 만들기

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

■ 아두이노 코드

- ▶ 객체생성
 - ▶ LEDbasic obj_name(pin)
- ▶ 함수 실행
 - ▶ obj_name.blink(time)

LED_object.ino

```
#include "LEDBasic.h"
LEDbasic LEDred(27), LEDyellow(26), LEDblue(25);
void setup() {}
void loop() {
    LEDblue.blink(100);
}
```

■ 고려사항

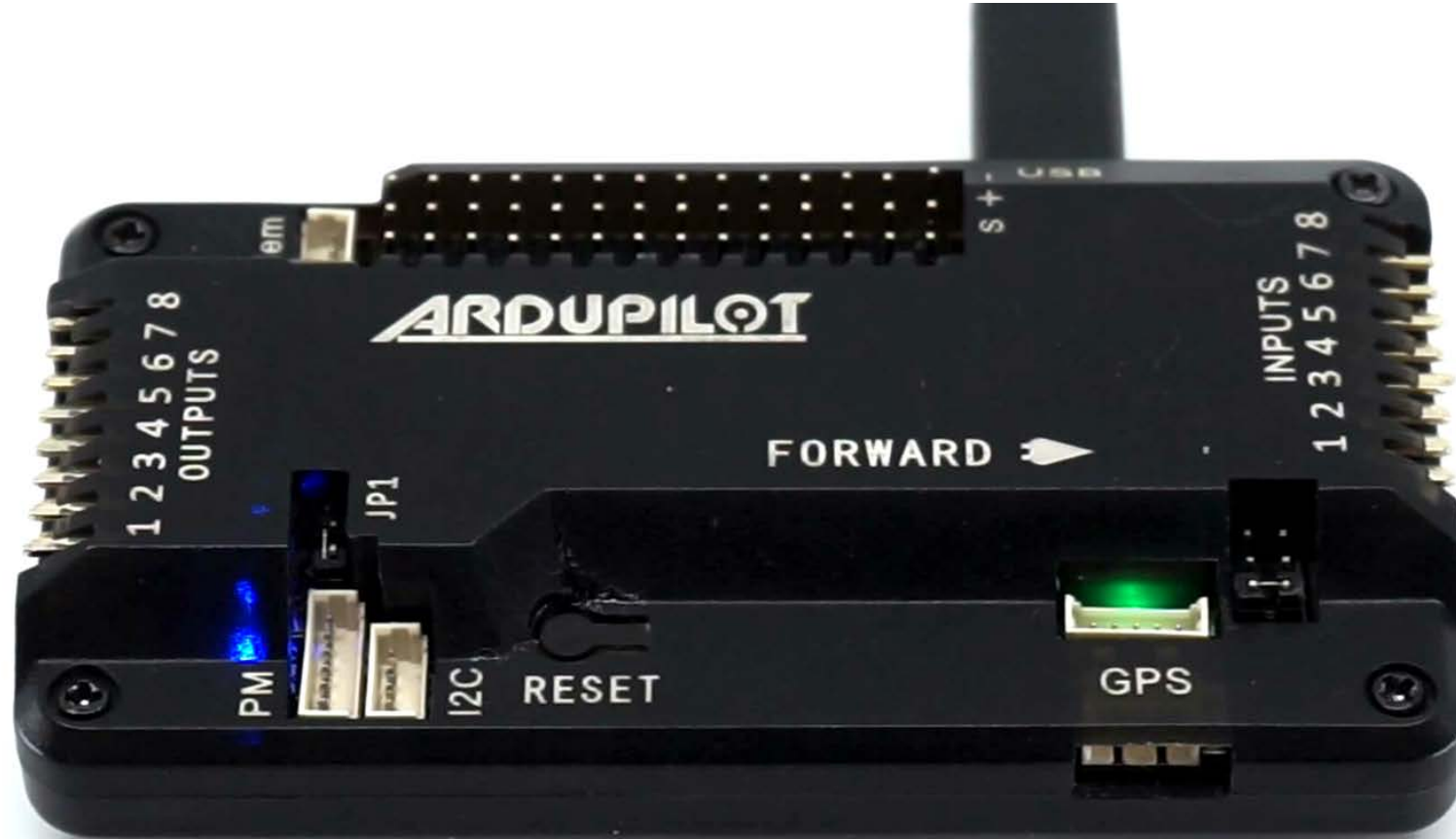
- ▶ cpp, h 파일은 같은 디렉토리에 있어야 한다.
- ▶ ino 파일 안에 클래스 선언이 가능 (가능하지만 규모가 커지면 복잡)

■ 문제점

- ▶ 동시에 3개의 LED를 blinking 하면 동시 실행이 안됨.

결과 동영상

Dept. of Mechanical System Design, Seoul National University of Science and Technology.



LED Flicker Class

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

■ 독립적으로 flickering 하는 LED 클래스

- millis() : 부팅 후 지금까지의 ms 시간

■ 멤버

▶ Property

- ▶ LED_pin: LED의 핀번호
- ▶ intval: 반전주기
- ▶ prevMillis: ms 단위의 이전 시간

▶ Method

- ▶ Flicker(int pinN,long intv) : 생성자
- ▶ update(): 상태 업데이트 동작

class Flicker

data

```
int LED_pin  
int intval  
uint32_t prevMillis
```

functions

public:

```
Flicker(int pinN,long intv)  
void update()
```

LED Flicker Class

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

클래스 코드

- LEDblue 100 ms
- LEDred 1000ms

Flicker_object.ino -2

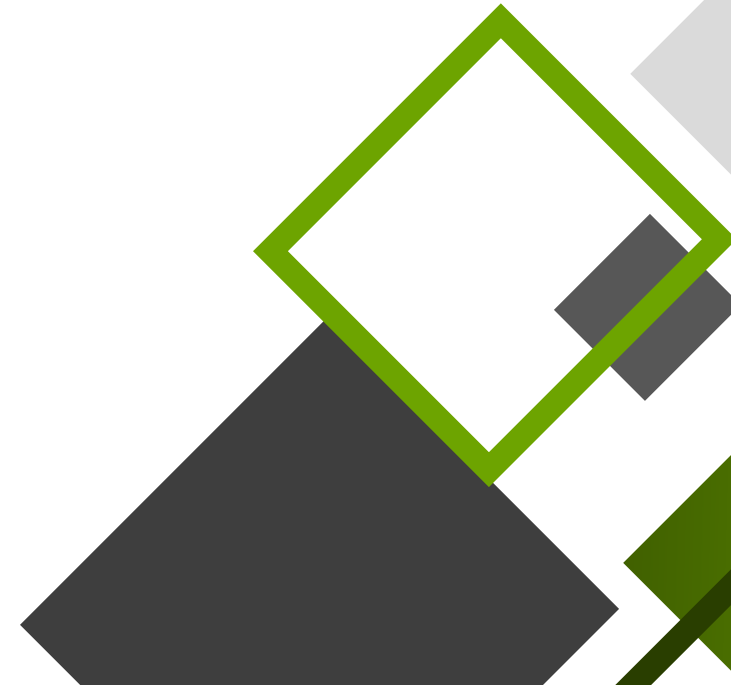
```
Flicker LEDblue(25,100),LEDred(27,1000);  
void setup() { }  
void loop() {  
    LEDblue.update();  
    LEDred.update();  
}
```

Flicker_object.ino -1

```
class Flicker{  
    int LED_pin, intval;  
    uint32_t prevMillis;  
public:  
    Flicker(int pinN,long intv){  
        LED_pin=pinN;intval=intv;  
        pinMode(LED_pin, OUTPUT);  
        prevMillis=millis();  
    }  
    void update() {  
        uint32_t curMillis = millis();  
        if( curMillis-prevMillis> intval ) {  
            prevMillis = curMillis;  
            digitalWrite(LED_pin,!digitalRead(LED_pin));  
        }  
    }  
};
```



아두이노와 Interrupt

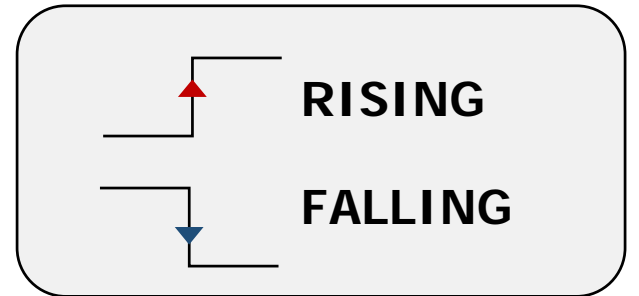


외부 인터럽트 사용

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

■ 외부인터럽트(External Interrupt):

- 특정 핀에 미리 지정한 **이벤트**가 발생하면 인터럽트!
- 다음과 같은 이벤트 중 하나를 선택 가능.
 - **LOW/HIGH** : 핀이 low 또는 high 상태가 되면 발생
 - **CHANGE**: 핀의 상태가 바뀌면 발생
 - **RISING** : 핀의 상태가 low에서 high로 바뀌면 발생
 - **FALLING** : 핀의 상태가 high에서 low로 바뀌면 발생



■ 아두이노 Mega2560의 외부인터럽트 지원 핀

- 2, 3, 18, 19, 20, 21 핀

외부 인터럽트 사용

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

■ 외부인터럽트관련 아두이노 지원함수

intNo = digitalPinToInterrupt(intPinNum)

- *intPinNum*: 외부 인터럽트 핀번호
- *intNo*: interrupt number

attachInterrupt(intNo, ISRname, mode)

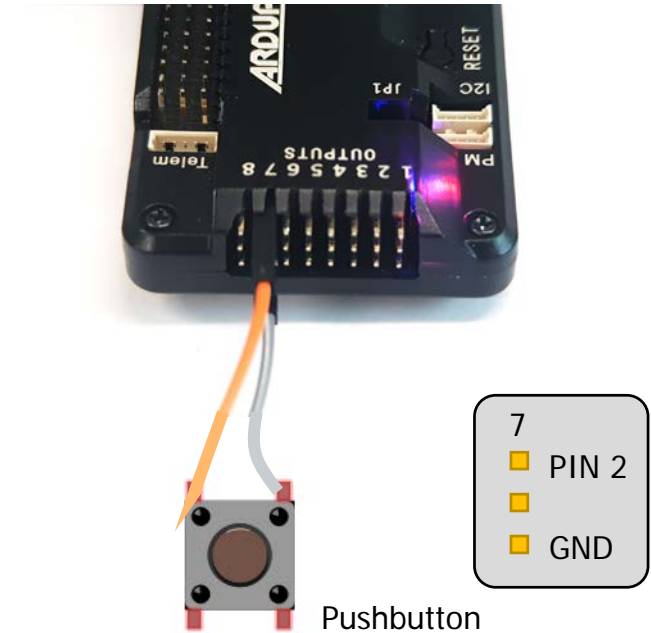
- *intNo*: interrupt number
- *ISRname*: interrupt service routine Name
- *mode*: FALLING/ RISING/ CHANGE/ HIGH/ LOW
 - FALLING/RISING: 상승/하강 엣지
 - CHANGE: 상승, 하강 모두 반응
 - HIGH/LOW: level high/low

외부 인터럽트 사용

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

■ 실험 회로

- 그림과 같이 APM의 OUTPUT 7번 이용
- 버튼 스위치 회로구성
- OUTPUT 7번은 아두이노 2번핀



외부 인터럽트 사용

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

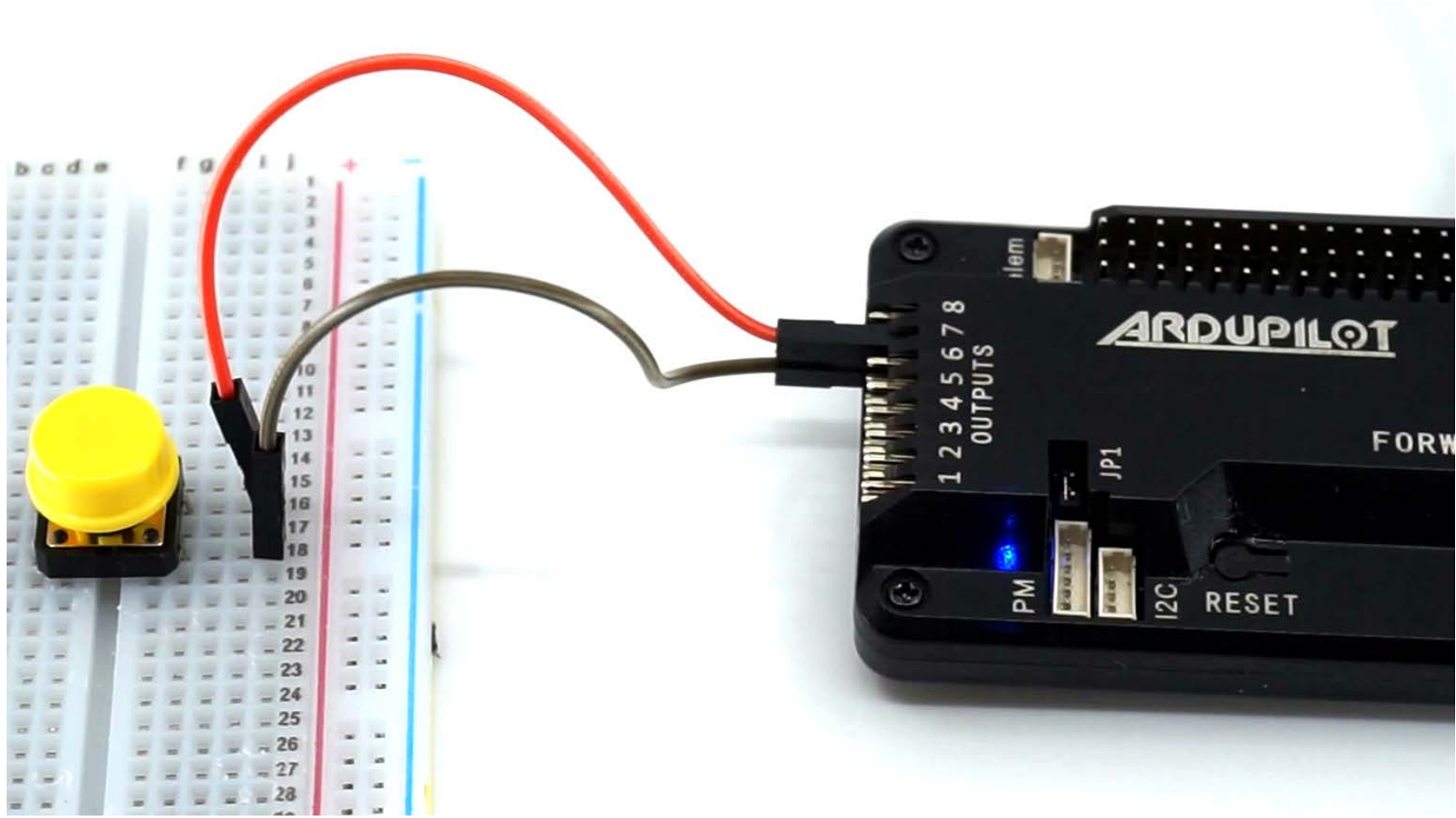
■ 코드

EI_interrupt.ino

```
void setup() {  
  Serial.begin(9600);  
  pinMode(25, OUTPUT); pinMode(26, OUTPUT);  
  pinMode(2, INPUT_PULLUP);           // Input Pull-up  
  digitalWrite(2, HIGH);               // apply Pull-up voltage  
  attachInterrupt(digitalPinToInterrupt(2), flip, FALLING);  
}  
  
void loop() {  
  digitalWrite(26, LOW);  delay(1000);  
  digitalWrite(26, HIGH); delay(1000);  
}  
  
void flip(){           // interrupt service routine  
  digitalWrite(25, !digitalRead(25));  
}
```

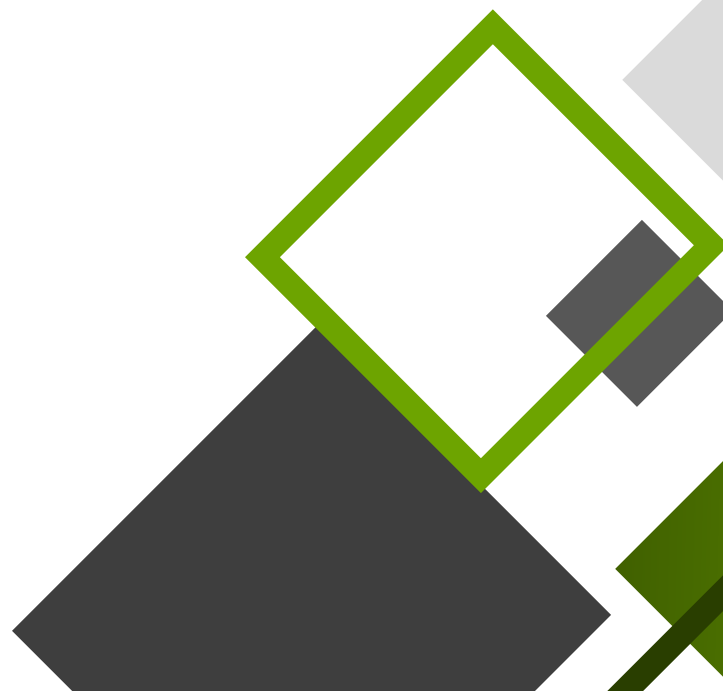
결과 동영상

Dept. of Mechanical System Design, Seoul National University of Science and Technology.





타이머 인터럽트 사용



타이머 인터럽트: Timer Counter 소개

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

■ ATmega2560 Timer Counter

- 8bit timer (0,2)와 16bit (1,3,4,5) 총 6개 내장.
- 내부 또는 외부 펄스를 count함.
- 용도에 맞게 pre-scale을 이용하여 속도조절

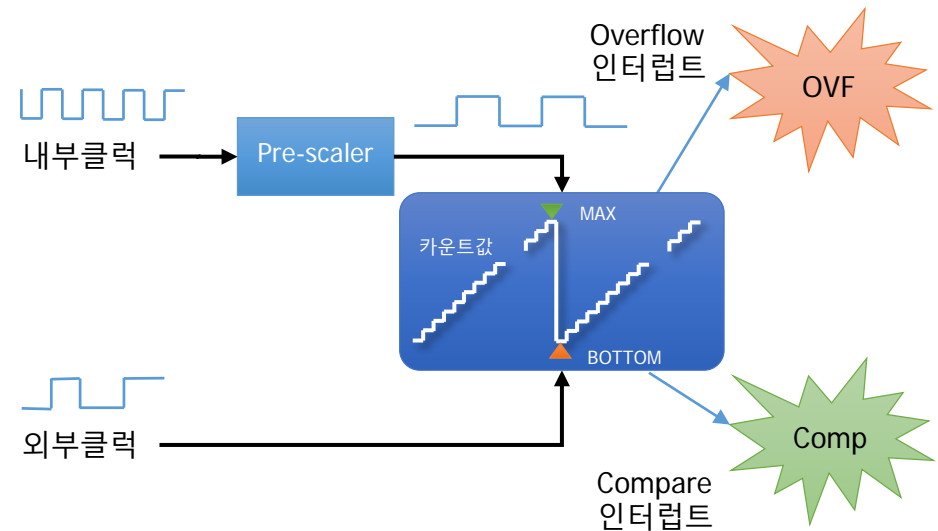
■ 지원 모드

- Normal: 일반 모드
- CTC(Clear Timer on Compare) 모드,
- 3가지 pwm mode 지원

■ 지원 인터럽트

- Overflow Interrupt
- Compare Interrupt
- Input Capture Interrupt

ATmega2560 타이머카운터					
8비트			16비트		
T/C0	T/C2	T/C1	T/C3	T/C4	TC/5



타이머 인터럽트: Timer2의 레지스터

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

■ TCCR2A -Control register A

7	6	5	4	3	2	1	0
COM2A1	COM2A0	COM2B1	COM2B0	-	-	WGM21	WGM20

비교출력모드 A 비교출력모드 B 웨이브 제너레이션

Mode	WGMn0:3
Normal	0
CTC	2
Fast PWM	3,7
PC PWM	1,5

■ TCCR2B -Control register B

7	6	5	4	3	2	1	0
FOC2A	FOC2B	-	-	WGM22	CS22	CS21	CS20

Force Output Compare: 즉각적 비교 프리스케일

CS22	CS21	CS20	pre-scale
0	0	0	0
0	0	1	1
0	1	0	8
0	1	1	32
1	0	0	128
1	0	1	256
1	1	0	1024

타이머 인터럽트: Timer2의 레지스터

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

TIMSK2 -Interrupt mask

TIMSK2	7	6	5	4	3	2	1	0
	-	-	-	-	-	OCIE2B	OCIE2A	TOIE2

Output compare Overflow

- 해당 bit에 1을 쓰면 인터럽트 활성화

TIFR2 - Interrupt flag register

TIFR2	7	6	5	4	3	2	1	0
	-	-	-	-	-	OCF2B	OCF2A	TOV2

Output compare Interrupt flag Overflow Interrupt flag

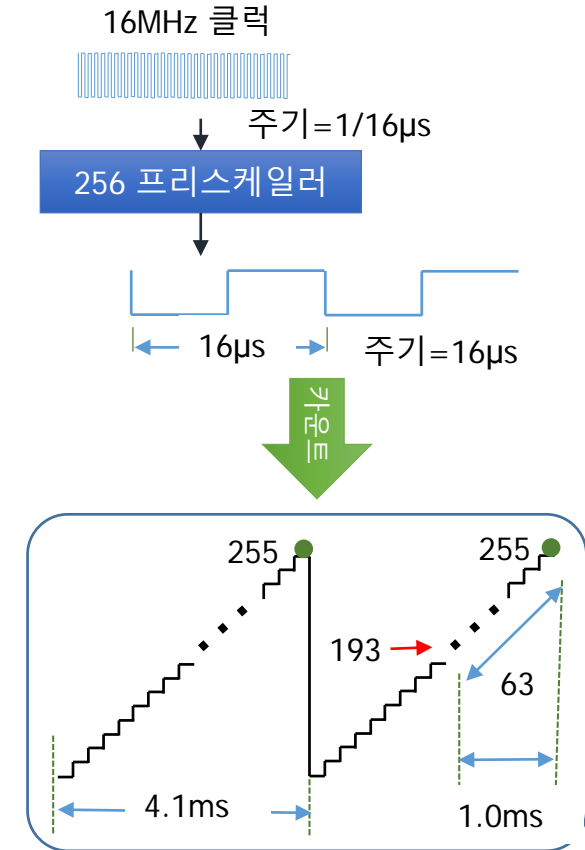
- 해당 Event가 발생하면 1이 됨
- 1을 플래그에 써주면 지워지고, interrupt를 사용하면 자동으로 지워짐

타이머 인터럽트: 1ms Timer 만들기

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

■ Timer2 가 0~255 까지 256개

- 기본 clock 이 16MHz
- pre-scale을 256 으로 하면
 - (CS22 CS21 CS20) = (101)
 - 1 펄스 주기 = $256/16000000 = 16 \mu s$
- Timer overflow time
 - overflow시 매번 TCNT2=193 로 초기화
 - $16 * (256 - 193) = 1,008 \mu s \approx 1 \text{ ms}$



타이머 인터럽트: 1ms Timer 만들기

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

■ Arduino code

- 25번: 500ms
- 27번: 5ms

```
uint32_t period=50000,prevTime;
void setup() {
  Serial.begin(115200);
  pinMode(27,OUTPUT); pinMode(25,OUTPUT);
  prevTime=micros();
  initTimer();
}
void loop() {
  digitalWrite(25,LOW);delay(500);
  digitalWrite(25,HIGH);delay(500);
}
void periodic(){
  uint32_t curTime=micros(); TCNT2 = 193;
  if (curTime-prevTime<period) return;
  prevTime=curTime;
  Serial.println("1");
  digitalWrite(27, !digitalRead(27));
}
ISR(TIMER2_OVF_vect) {
  periodic();
}
```

타이머 인터럽트: 1ms Timer 만들기

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

■ Arduino code

```
void initTimer(){
    TIMSK2 = 0;           // To prevent interrupts during setup
    TCCR2A = 0;           // normal mode
    TCCR2B = _BV(CS21) | _BV(CS22); // Set prescaler (110) -> clk/256 -> 16usec
    TCNT2 = 193;          // Timer overflow time = 16*(256-193) = 1008us
    TIFR2 = _BV(TOV2);    // clear previous interrupts flags
    TIMSK2 = _BV(TOIE2);  // enable the OV interrupt
}
```

■ 참고

ISR : interrupt service routine

_BV(bit) :

AVR macro로서 <avr/io.h> 에서 정의되어 있음

#define _BV(bit) (1<<(bit))

타이머 인터럽트: Timer2 Class 만들기

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

■ Timer2 Class 정의

- init(...): 초기화 및 함수지정
- _proc : 실행하고자하는 함수

■ 함수 포인터 선언

- 함수를 지칭하는 포인터 형을 선언함
 - ex) typedef int (*fun)(int,int)
- fun myfun;

```
typedef return_type (*fuction_pointer)(type_param, ...);
```

TimerTwo.h

```
#ifndef __TIMERTWO_H__
#define __TIMERTWO_H__
#include <stdint.h>
typedef void (*_procedure)(void );
class Timer2{
public:
    void init(_procedure);
    static _procedure _proc;
};
#endif
```

타이머 인터럽트: Timer2 Class 만들기

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

■ Timer2 Class 코드

```
TimerTwo.cpp

#include "TimerTwo.h"
#include "avr/interrupt.h"
_procedure Timer2::_proc;
void Timer2::init(_procedure proc){
    TIMSK2 = 0;
    TCCR2A = 0;
    TCCR2B = _BV(CS21) | _BV(CS22); //-> 16usec
    TCNT2 = 193;
    TIFR2 = _BV(TOV2);
    TIMSK2 = _BV(TOIE2);
    _proc=proc;
}
ISR(TIMER2_OVF_vect) {
    Timer2::_proc (); //excute _proc() function
    TCNT2 = 193;
}
```

타이머 인터럽트: Timer2 Class 만들기

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

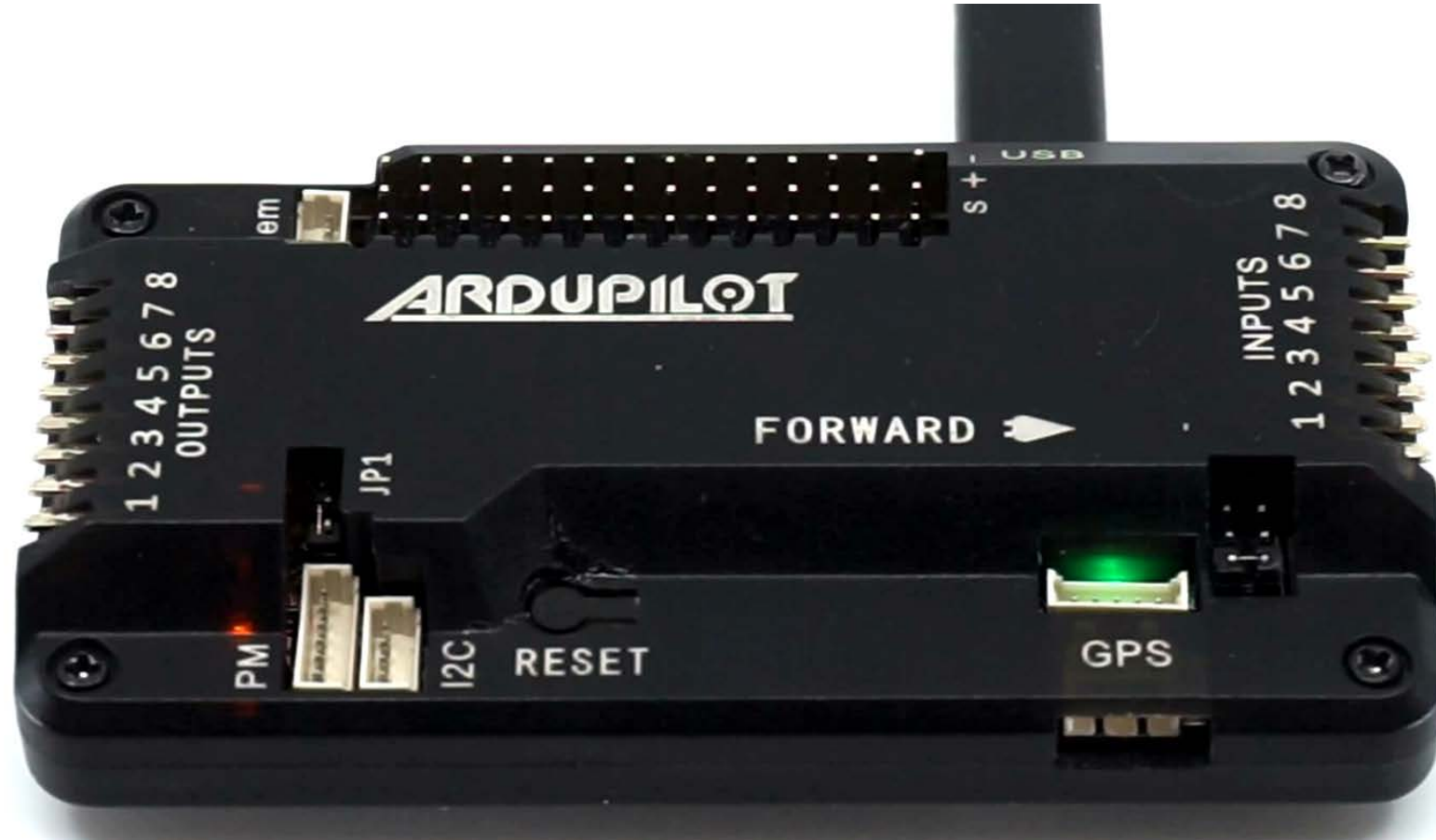
■ Arduino 코드

simpleLED.ino

```
#include "TimerTwo.h"
bool state=false;
uint32_t period=100000,prevTime;
Timer2 back;
void setup() {
    pinMode(26,OUTPUT); pinMode(25,OUTPUT);
    prevTime=micros();
    back.init(periodic);
}
void loop() {
    digitalWrite(25,1); delay(1000);
    digitalWrite(25,0); delay(1000);
}
void periodic(){
    uint32_t curTime=micros();
    if (curTime-prevTime < period) return;
    prevTime=curTime;
    digitalWrite(26, state);
    state=!state;
}
```

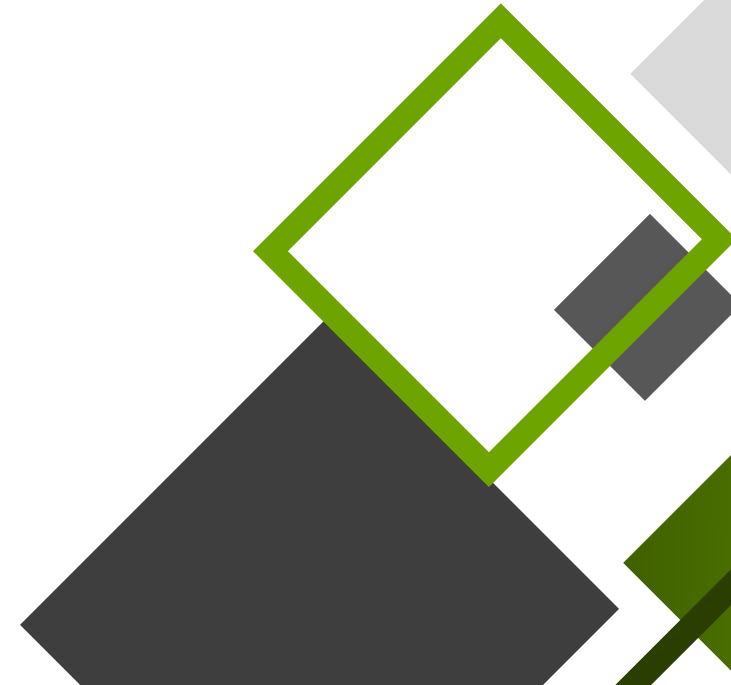
결과 동영상

Dept. of Mechanical System Design, Seoul National University of Science and Technology.





Serial 통신 고급주제



Serial 통신의 고려사항

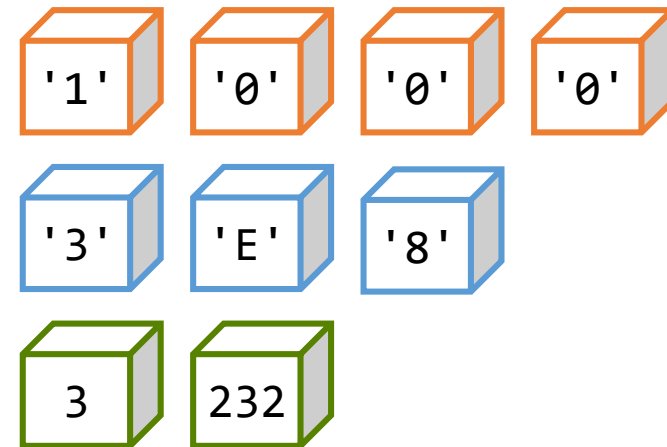
Dept. of Mechanical System Design, Seoul National University of Science and Technology.

■ 시리얼 통신의 수신 고려사항

- 다른 작업 중 데이터 수신을 할 수 없음.
- **parseInt()**나 **parseFloat()** 는 1초 정도의 시간 지연 문제
- **SerialEvent()**를 사용하면 **loop()** 끝에서 처리.
 - **loop()**에 **delay()**가 있으면 문제
- ISR을 사용하는 **NeoHwSerial** 라이브러리를 이용하면 언제나 수신 가능.

■ 시리얼 통신의 송신 고려사항

- 1000을 보내는 방법
 - ① 10진문자로: '1' - '0' - '0' - '0'
 - ② 16진문자로: '3' - 'E' - '8'
 - ③ binary전송: '\0x03' - '\0xE8'



Serial Event: loop-back example

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

- LED를 점멸하는 중에 시리얼 모니터에서 오는 데이터를 재전송

```
void setup() {  
  Serial.begin(9600);  
  pinMode(27,OUTPUT);  
}  
void loop() {  
  digitalWrite(27,HIGH);delay(2000);  
  digitalWrite(27,LOW);delay(2000);  
}  
void serialEvent(){  
  while (Serial.available()){  
    uint8_t val=Serial.read();  
    Serial.write(val);  
  }  
}
```

loop에 delay()함수가 있어 통신이
원활하지 못하다.

결과 동영상

Dept. of Mechanical System Design, Seoul National University of Science and Technology.



String class in arduino

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

■ String class: 문자열 클래스

- 선언: `String s = "Arduino";`
- 연산 : `c = a + b;` (붙이기)
- 메소드
 - `.reserve(n)` : n개 확보 메모리
 - `.toFloat(), .toInt()` : 형 변환 `inString+=c;`
 - `.endsWith(s)`
 - `.startsWith()`
 - `.substring(from[, to])`
- 한글자씩 붙이기 위해서는 `char` 형이 되어야한다.
 - `inString+= c;`

Serial Event: float read example

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

- LED를 점멸하는 중에 시리얼 모니터에서 오는 데이터의 제공근을 구하여 전송

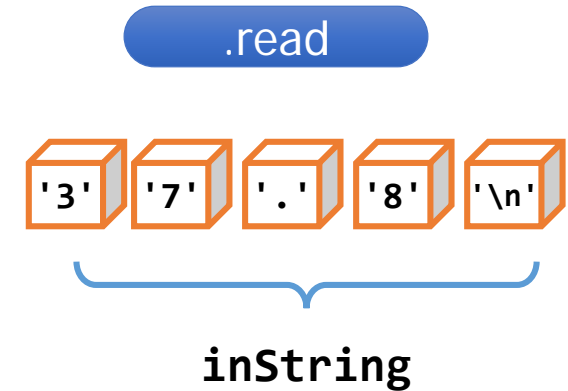
```
String inString = "";  
void setup() {  
  Serial.begin(9600);  pinMode(27,OUTPUT);  
  inString.reserve(50);  
}  
void loop() {  
  digitalWrite(27,HIGH);delay(2000);  
  digitalWrite(27,LOW);delay(2000);  
}
```

Serial Event: float read example

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

code

```
void serialEvent(){
  while (Serial.available()){
    char c= (char)Serial.read();
    if (c=='\n'){
      float f= inString.toFloat();
      Serial.println(sqrt(f),5);
      inString="";
    }
    else
      inString+=c;
  }
}
```



loop에 delay()함수가 있으면 통신이 원활하지 못하다.

결과 동영상

Dept. of Mechanical System Design, Seoul National University of Science and Technology.



Receive Comma Separated Values

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

■ CSV(comma separated values)예제

- 시리얼모니터 창에 직각삼각형의 두 변 3.0, 4.0 를 입력하면 빗변 5.0을 수신

```
String inString = "";  
float x,y;  
void setup() {  
  Serial.begin(9600);  
  pinMode(25,OUTPUT);  
  inString.reserve(200);  
}  
void loop() {  
  digitalWrite(25,1); delay(2000);  
  digitalWrite(25,0); delay(2000);  
}
```

Receive Comma Separated Values

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

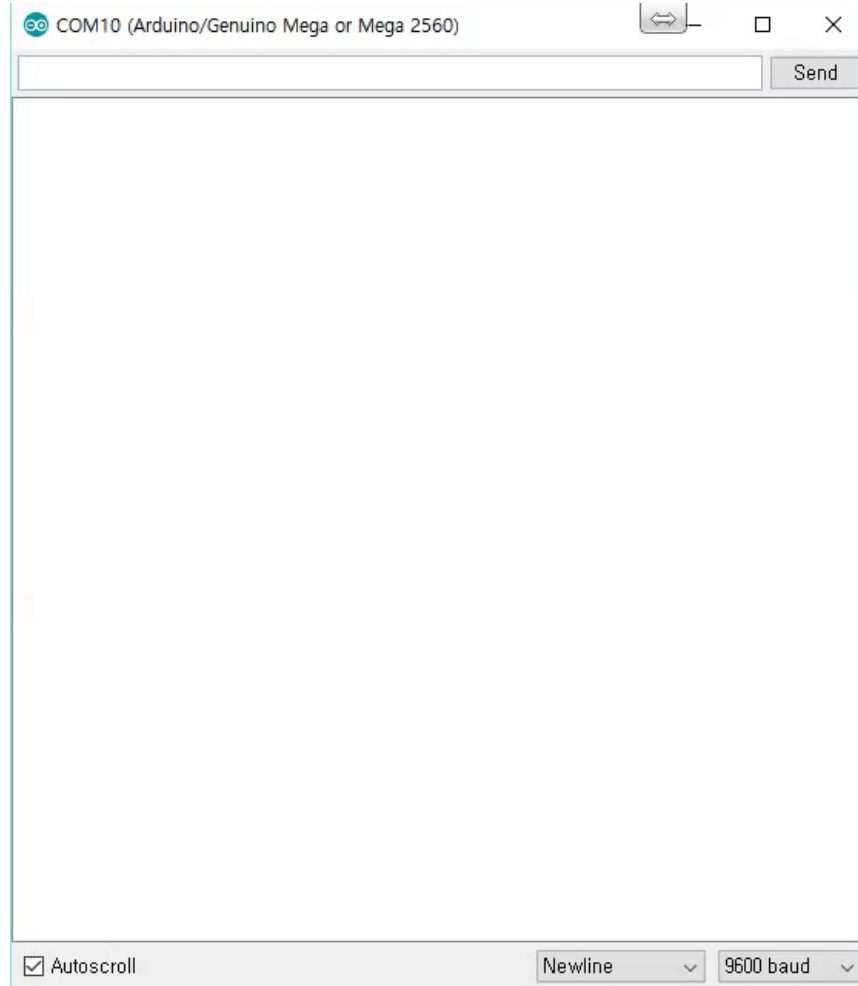
■ 코드

```
void serialEvent(){
  while (Serial.available()){
    char c= (char)Serial.read();
    if (c=='\n'){
      y= inString.toFloat();
      Serial.println(sqrt(x*x+y*y),5);
      inString="";
    }
    else if (c==','){
      x= inString.toFloat();
      inString="";
    }
    else
      inString+=c;
  }
}
```

3개의 수를 입력 받으려는 경우는
어떻게 바뀌어야 할까?

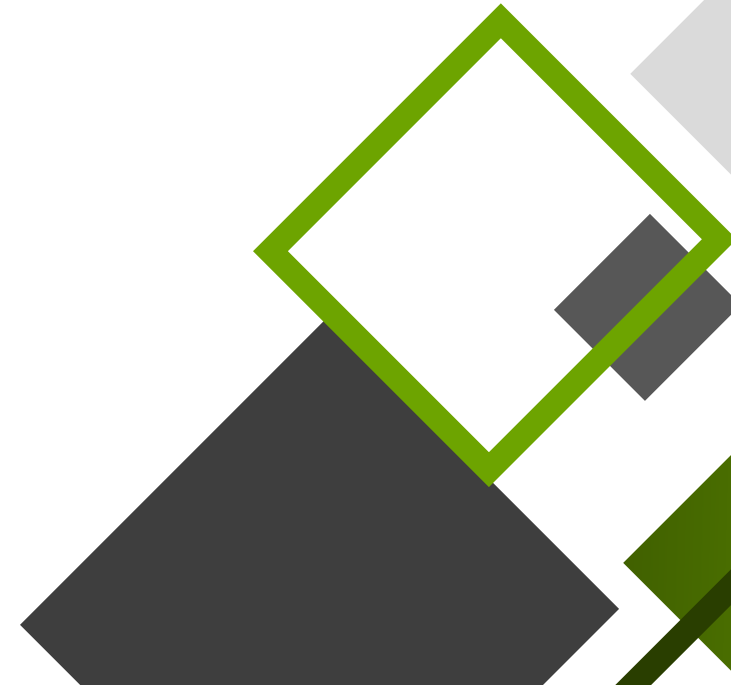
결과 동영상

Dept. of Mechanical System Design, Seoul National University of Science and Technology.





Interrupt 이용 Serial 통신



UART interrupt 사용

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

■ NeoHWSerial 라이브러리 활용

- 라이브러리 : <https://github.com/SlashDevin/NeoHWSerial>
- download zip file
- 설치방법:
 - Documents\ Arduino\ libraries\ NeoHWSerial 를 만든다.
 - 다음 파일 복사

이름	수정한 날짜	유형	크기
NeoHWSerial.cpp	2017-04-04 오전...	C++ Source File	7KB
NeoHWSerial.h	2017-04-04 오전...	C Header File	5KB
NeoHWSerial_private.h	2017-04-04 오전...	C Header File	5KB
NeoHWSerial0.cpp	2017-04-04 오전...	C++ Source File	3KB
NeoHWSerial1.cpp	2017-04-04 오전...	C++ Source File	3KB
NeoHWSerial2.cpp	2017-04-04 오전...	C++ Source File	2KB
NeoHWSerial3.cpp	2017-04-04 오전...	C++ Source File	2KB

UART interrupt 사용

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

■ NeoHWSerial 라이브러리 사용

- 라이브러리 사용

```
#include <NeoHWSerial.h>
```

- 라이브러리 include

- 시작

```
NeoSerial.begin( baudrate );
```

- baudrate* : 통신속도 설정 (9600, 19200, ... , 115200, ..., 250000)

- ISR 함수지정

```
NeoSerial.attachInterrupt( isr_func );
```

- sr_func* : 데이터가 도착했을 때 실행해야 할 함수명

- 수신

```
void isr_func( uint8_t r ){ ... }
```

- r* : 수신 8bit 데이터

- 전송

```
NeoSerial.write( c );
```

- c* : 전송할 데이터

NeoHWSerial: loop-back example

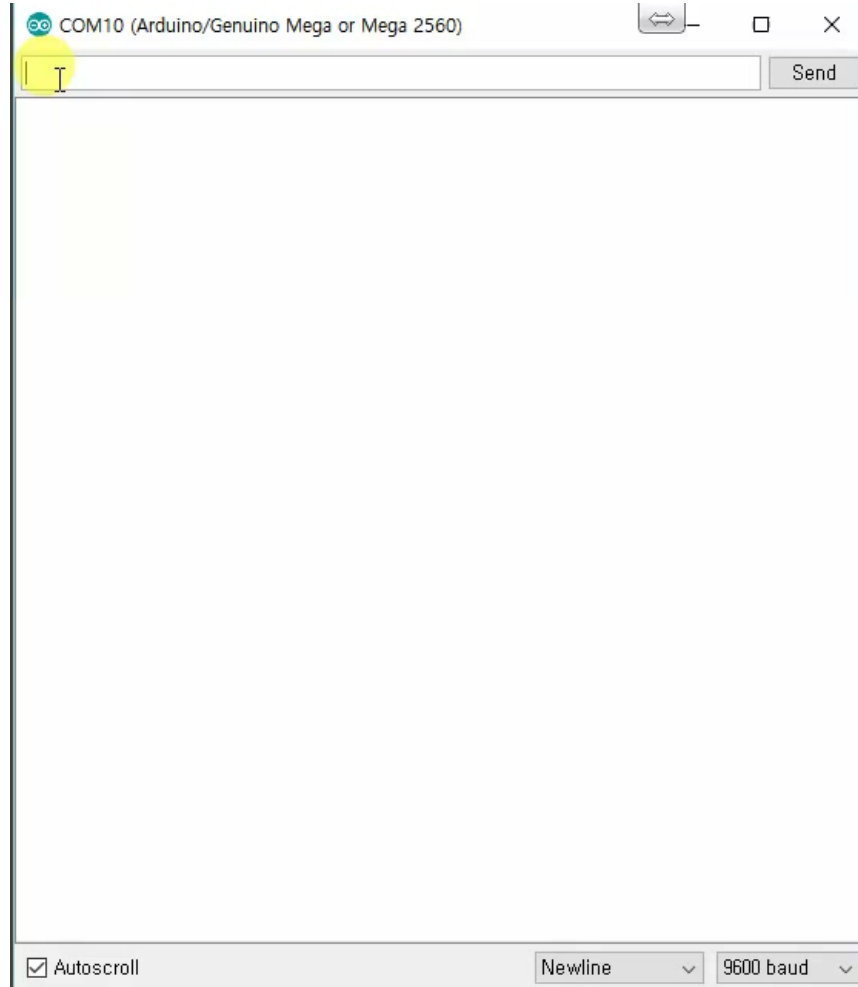
Dept. of Mechanical System Design, Seoul National University of Science and Technology.

- LED를 점멸하는 중에 시리얼 모니터에서 오는 데이터를 재전송

```
#include <NeoHWSerial.h>
uint8_t c;
void setup() {
  pinMode(27,OUTPUT);
  NeoSerial.begin( 9600 );
  NeoSerial.attachInterrupt( recvData );
}
void loop() {
  digitalWrite(27,HIGH); delay(2000); digitalWrite(27,LOW); delay(2000);
}
void recvData( uint8_t c ) {
  NeoSerial.write( c );
}
```

결과 동영상

Dept. of Mechanical System Design, Seoul National University of Science and Technology.



NeoHWSerial : float read example

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

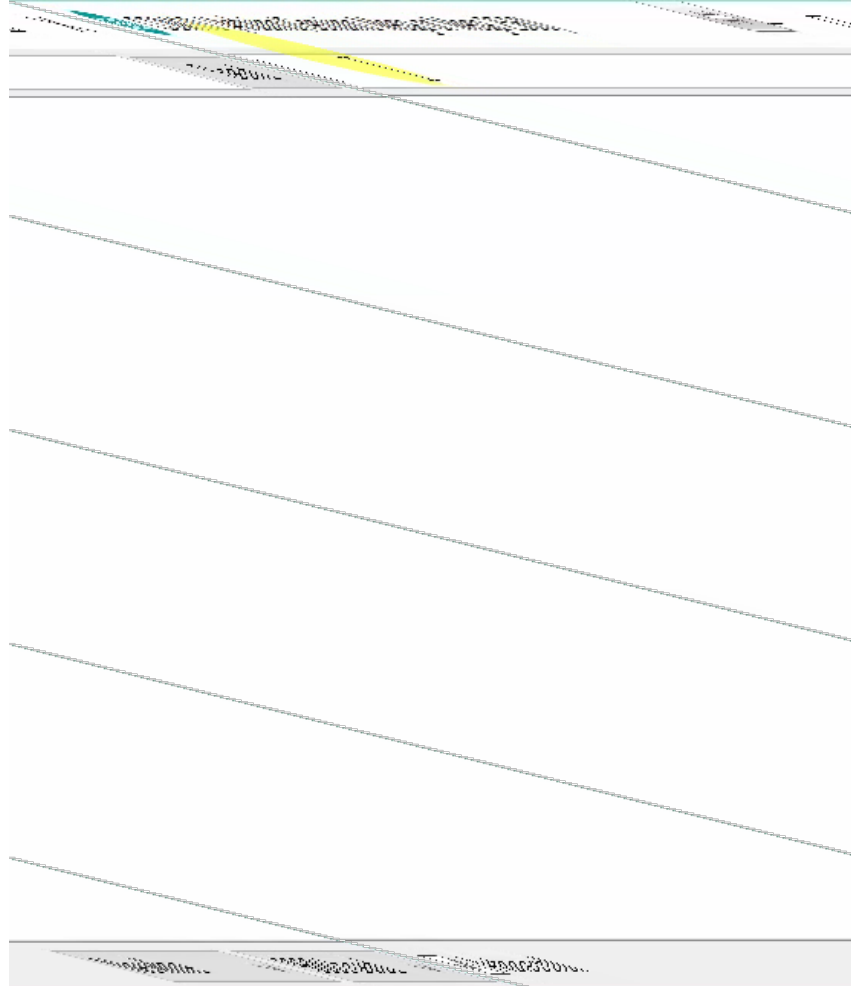
■ LED를 점멸하는 중에 제곱근을 구하여 전송

```
#include <NeoHWSerial.h>
String inString = "";
void setup() {
  pinMode(27,OUTPUT);
  NeoSerial.begin(9600);
  NeoSerial.attachInterrupt( recvData );
  inString.reserve(50);
}
void loop() {
  digitalWrite(27,HIGH); delay(1000); digitalWrite(27,LOW); delay(1000);
}
void recvData( uint8_t c) {
  if (c=='\n'){
    float f= inString.toFloat();
    NeoSerial.println(sqrt(f),5);
    inString="";
  }
  else
    inString+=(char)c;    // char로 변환이 중요
}
```

loop에 delay()함수가 있어도 통신이 원활하다.

결과 동영상

Dept. of Mechanical System Design, Seoul National University of Science and Technology.



NeoHWSerial: Receive CSV

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

■ CSV(comma separated values)예제

- 시리얼모니터 창에 직각삼각형의 두 변 3.0, 4.0 를 입력하면 빗변 5.0을 수신

```
#include <NeoHWSerial.h>
String inString = "";
float x,y;
bool getFrame=false;
void setup() {
  pinMode(27,OUTPUT);
  NeoSerial.begin( 9600 );
  NeoSerial.attachInterrupt( recvData );
  inString.reserve(50);
}
void loop() {
  digitalWrite(27,HIGH); delay(1000);
  digitalWrite(27,LOW); delay(1000);
}
```

NeoHWSerial: Receive CSV

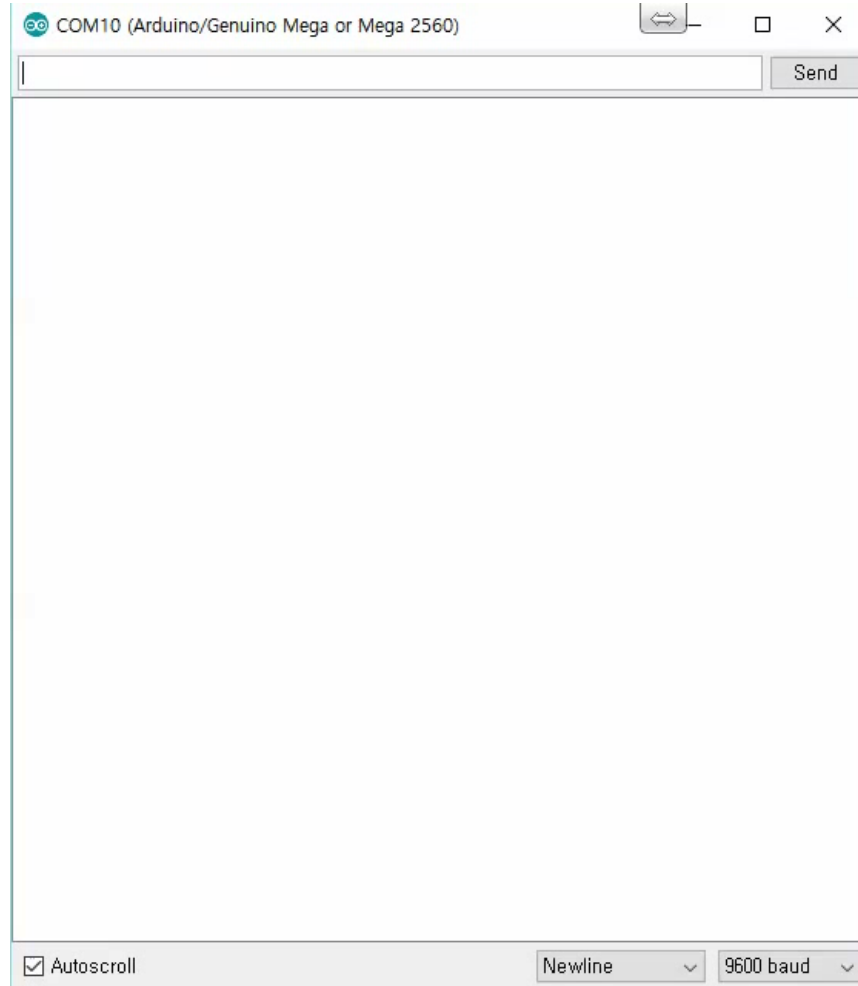
Dept. of Mechanical System Design, Seoul National University of Science and Technology.

■ 코드

```
void recvData( uint8_t c) {  
    if (c=='\n'){  
        y= inString.toFloat();  
        NeoSerial.println(sqrt(x*x+y*y),5);  
        inString="";  
    }  
    else if (c==','){  
        x= inString.toFloat();  
        inString="";  
    }  
    else  
        inString+=(char)c;    // char로 변환이 중요  
}
```

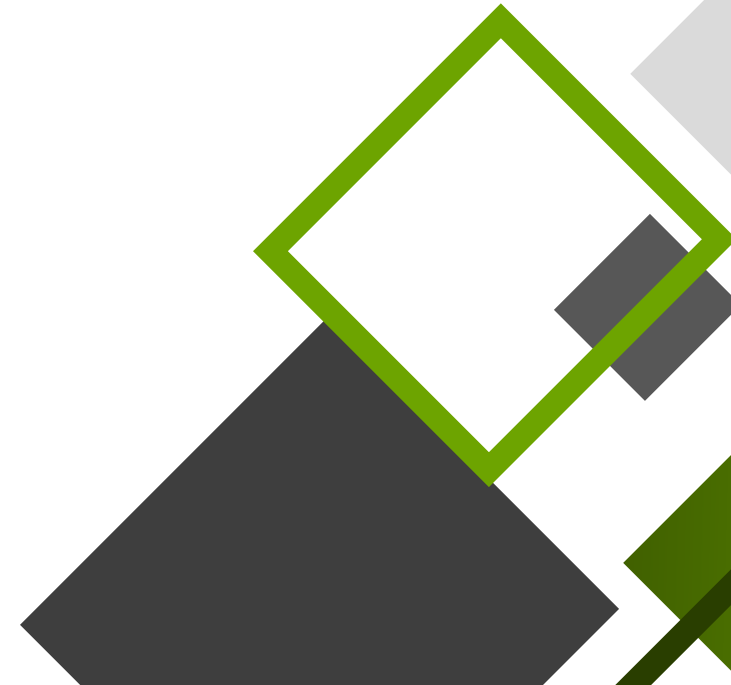
결과 동영상

Dept. of Mechanical System Design, Seoul National University of Science and Technology.





Binary 전송



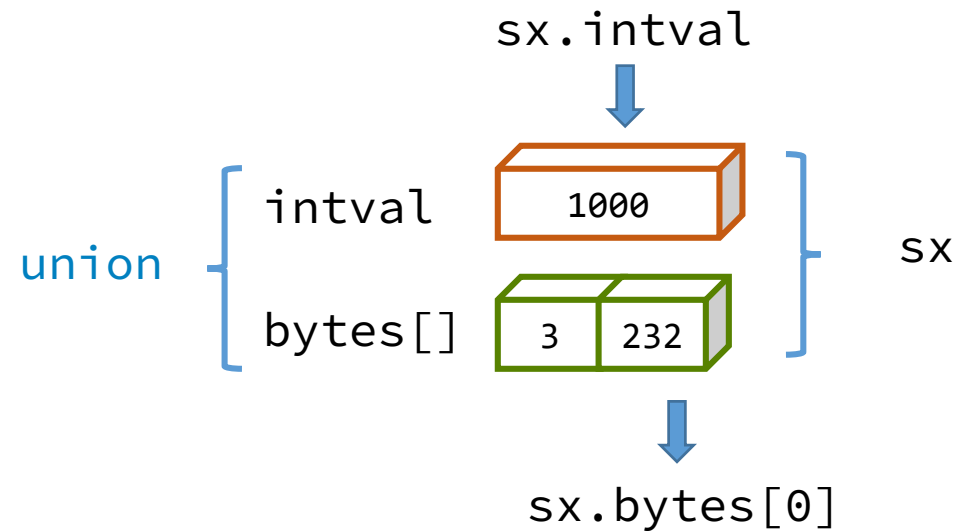
Serial 통신에서 binary로 전송

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

■ 시리얼 통신의 binary로 2 byte 보내기

- 1000을 binary로: '\0x03' - '\0xE8'

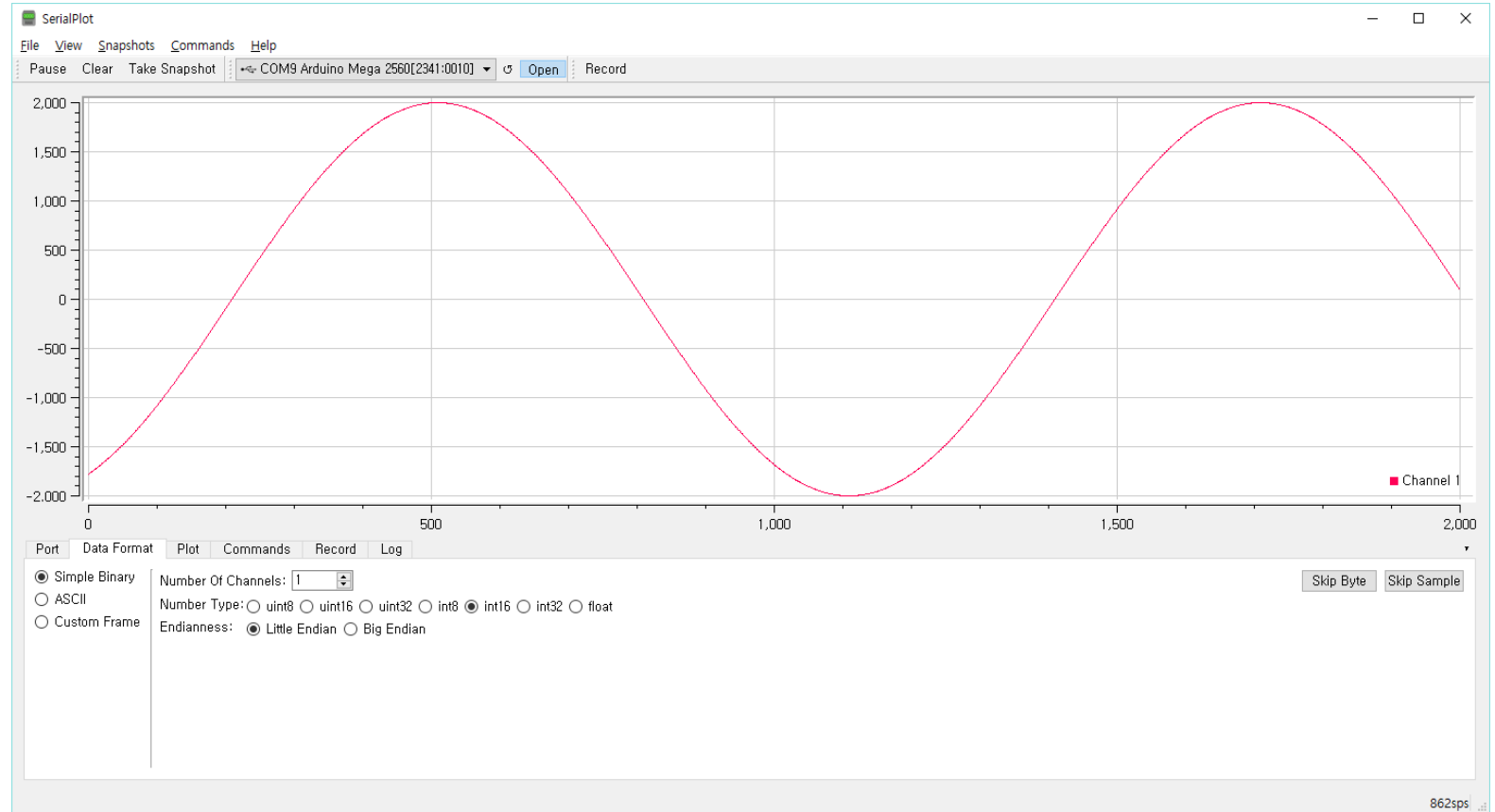
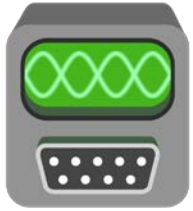
```
#define omega 0.3
int j=0;
union {int intval; byte bytes[2];} sx;
void setup() {
  Serial.begin(115200);
}
void loop() {
  float t=omega*DEG_TO_RAD*j;
  sx.intval=2000.*sin(t);
  Serial.write(sx.bytes,2);
  delay(1);
  j++;
}
```



Results

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

SerialPlot 사용



문제점

- packet시작 불분명

Download

- <https://bitbucket.org/hyOzd/serialplot>

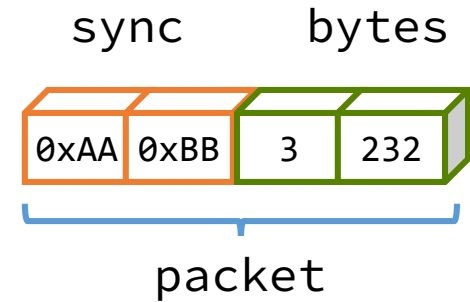
Serial 통신에서 binary로 전송

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

■ binary에 sync 도입으로 packet 구성

- sync byte 추가 { 0xAA, 0xBB }

```
#define omega 0.3
int j=0;
union {int intval; byte bytes[2];} sx;
char sync[2]={0xAA,0xBB}; //sync
void setup() {
    Serial.begin(115200);
}
void loop() {
    float t=omega*DEG_TO_RAD*j;
    sx.intval=2000.*sin(t);
    Serial.write(sync,2);
    Serial.write(sx.bytes,2);
    delay(1);
    j++;
}
```

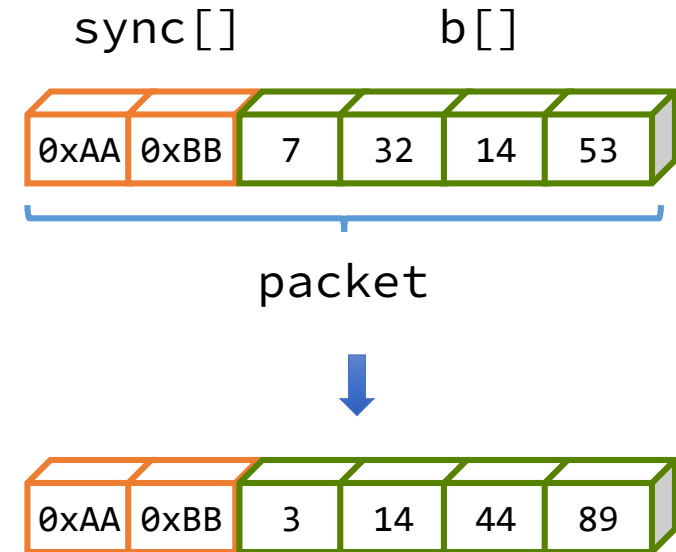


Serial 통신에서 binary로 전송

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

float type으로 packet 구성

```
#define omega 0.3
int j=0;
char sync[2]={0xAA,0xBB};
void setup() {
    Serial.begin(115200);
}
void loop() {
    float t=omega*DEG_TO_RAD*j;
    float fdata=2000.*sin(t);
    sendPacket(fdata);
    delay(1);
    j++;
}
void sendPacket(float f) {
    byte* b = (byte *) &f;
    Serial.write(sync,2);
    Serial.write(b,4);
}
```



Results

Dept. of Mechanical System Design, Seoul National University of Science and Technology.

■ frame 설정 필요

Port Data Format Plot Commands Record Log

☐ Simple Binary
☐ ASCII
☒ Custom Frame

Frame Start:

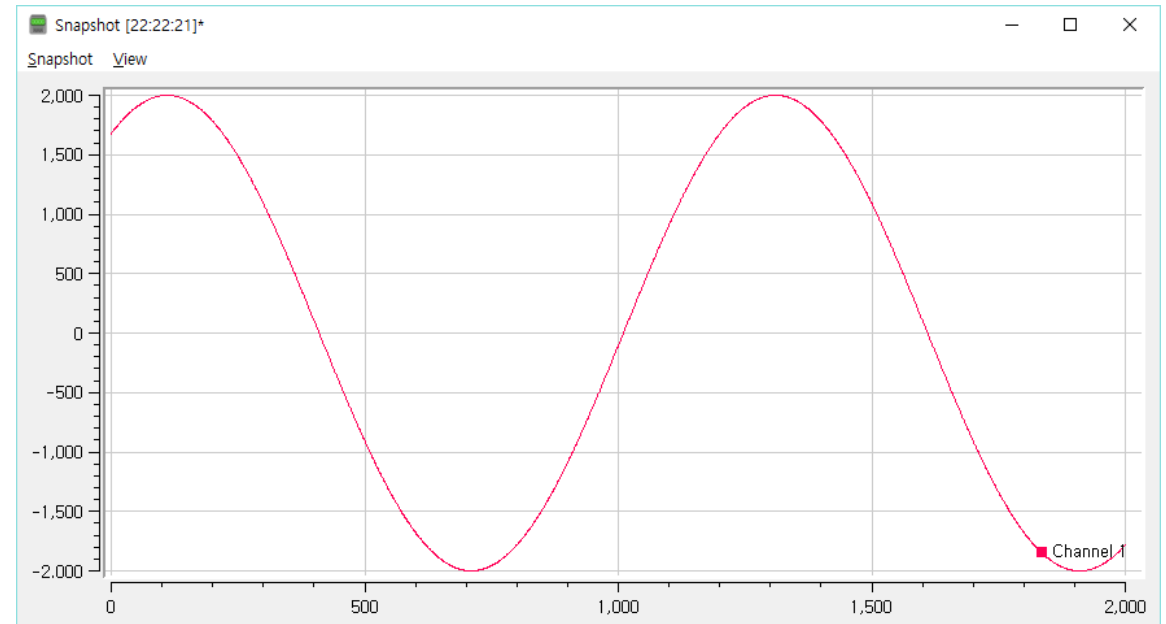
Channels:

Frame Size: ☒ Fixed Size: ☐ First byte of the payload is size

Number Type: ☐ uint8 ☐ uint16 ☐ uint32 ☐ int8 ☐ int16 ☐ int32 ☒ float

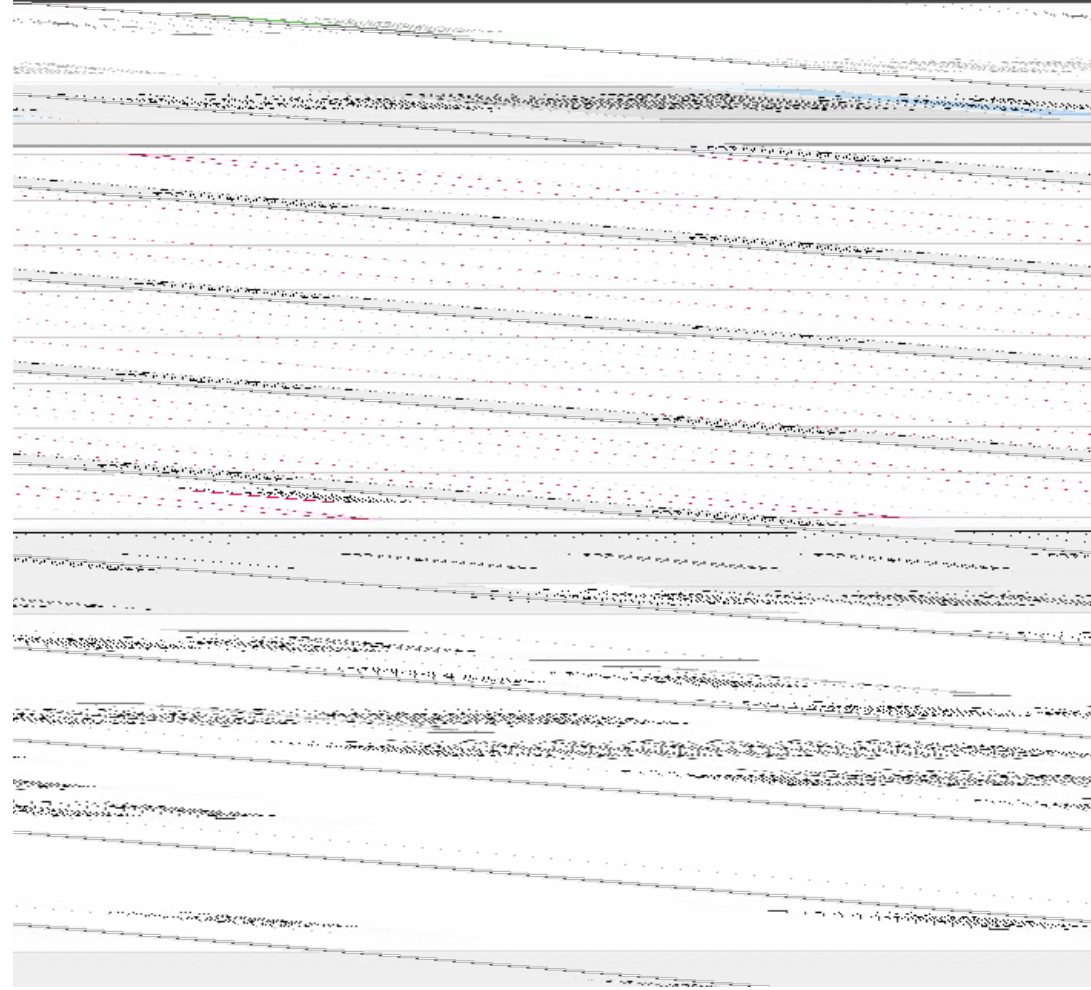
Endianness: ☒ Little Endian ☐ Big Endian

Checksum: ☐ Enabled



결과 동영상

Dept. of Mechanical System Design, Seoul National University of Science and Technology.



The background of the slide is decorated with abstract geometric shapes. On the left and right sides, there are clusters of squares and rectangles in various shades of green (from light to dark), grey, and dark grey. Some of these shapes are outlined, while others are solid. The central area is white, providing a clean space for the text.

THANK YOU

An Unmanned aerial vehicle (UAV) is a Unmanned Aerial Vehicle. UAVs include both autonomous (means they can do it alone) drones and remotely piloted vehicles (RPVs).