

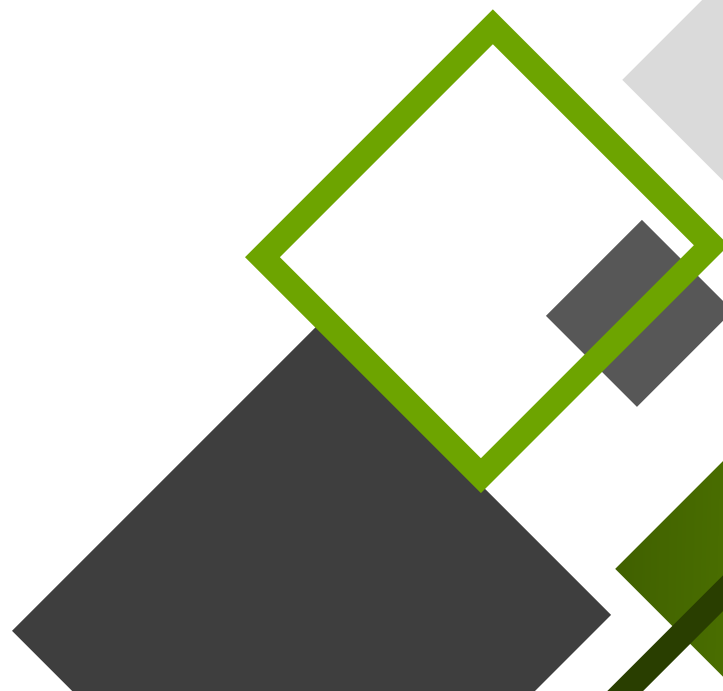
모터제어와 RC

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). A UAV is capable of controlled, sustained level flight and is powered by a jet, reciprocating, or electric engine.





PWM 이란?

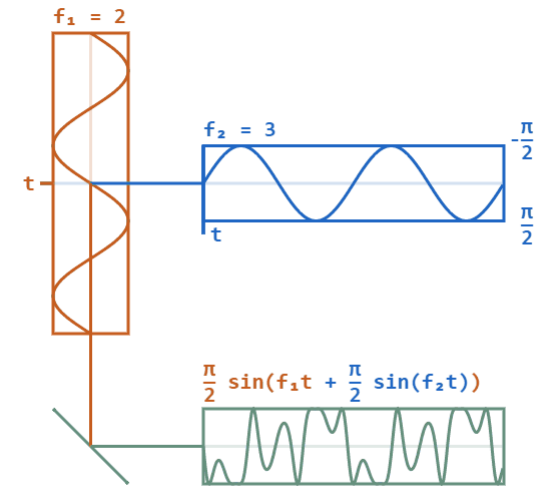
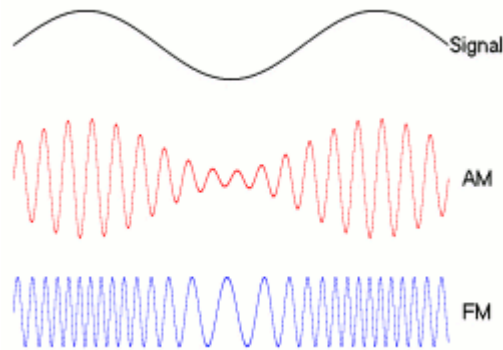


PWM (pulse width modulation)

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

■ 변조 (modulation)

- 정보를 신호에 담는 기법을 의미
 - 아날로그 변조:
 - **AM**(Amplitude), **FM**(Frequency), **PM**(Phase)



By Berserkerus - Own work, CC BY-SA 2.5,
<https://commons.wikimedia.org/w/index.php?curid=5071748>

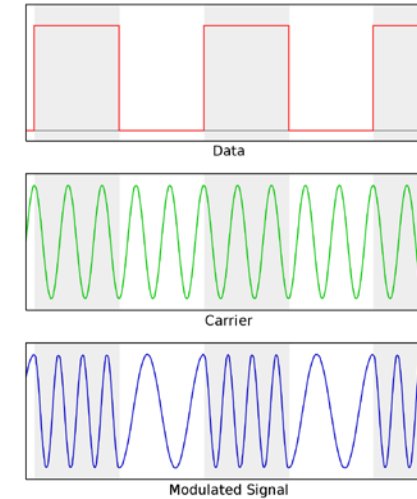
https://commons.wikimedia.org/wiki/File:Phase_modulation.gif

PWM (pulse width modulation)

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

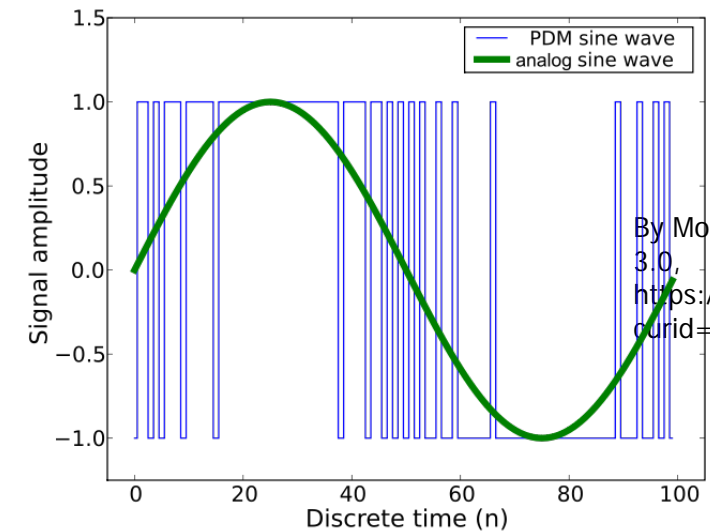
- 디지털변조:
 - **FSK** (Frequency Shift key),
 - **ASK**(Amplitude Shift key)
- 펄스변조:
 - **PWM**(Pulse Width),
 - **PPM**(Pulse Position) ,
 - **PDM**(Pulse Density)

FSK



<https://commons.wikimedia.org>

PDM



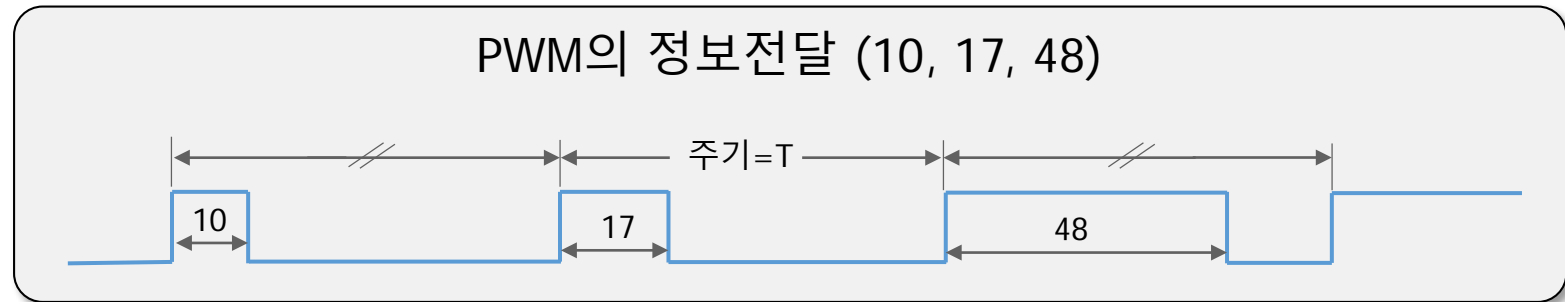
By Moxfyre at English W
3.0,
<https://commons.wikime>
curid=3239064

PWM 이란?

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

■ 펄스폭변조 (Pulse width modulation)

- PWM은 일정 주기 간격으로 나타나는 펄스의 폭(On-duty)으로 정보를 전달
- 예) 다음과 같은 파형으로 10 - 17 - 48 순서로 정보전달



■ 주요 용도

- 정보전달
 - RC control 에 사용, RC servo motor
- Power control
 - 모터 제어, 조명제어, 히터 제어

일반적 PWM 규격

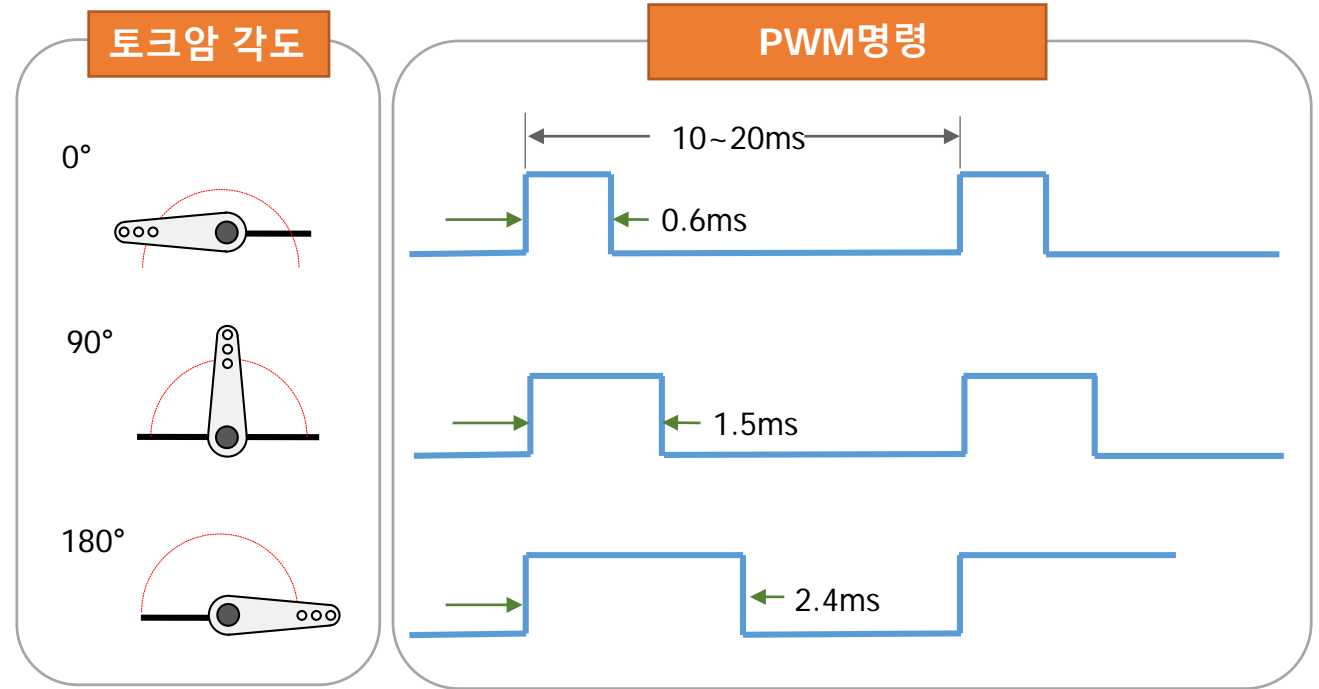
Dept. of Mechanical System Design, Seoul National University of Science and Technology.

RC 서보 모터

- 20ms주기(50Hz)내에 **On-duty**를 0.6~2.4ms 사이로 주면 모터 각도가 지정됨.

ESC

- PWM 신호에 비례하여 BLDC모터 속도제어
- On-duty** 1000 ~ 2000 μ s 범위

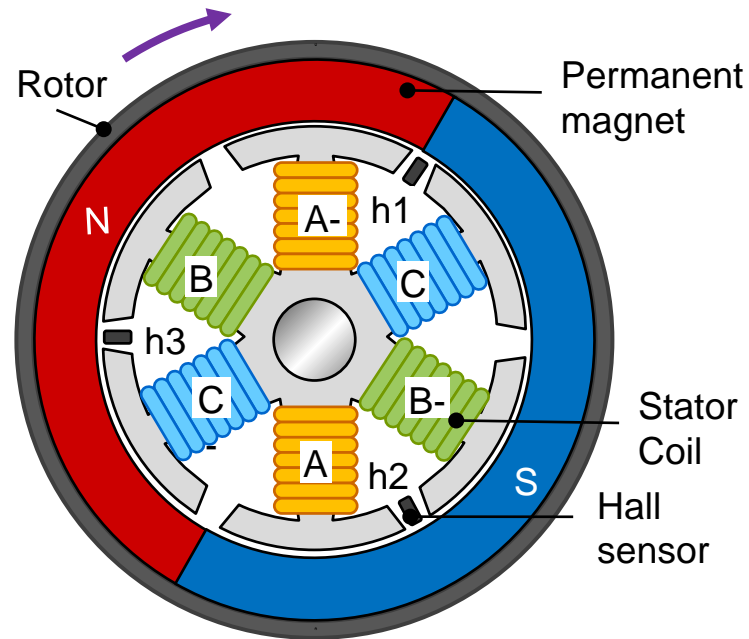


BLDC motor

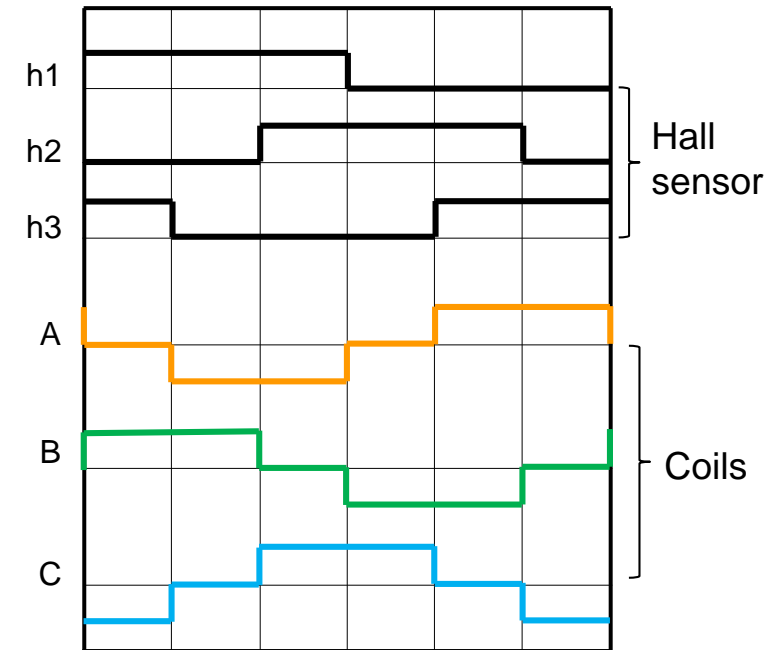
Dept. of Mechanical System Design, Seoul National University of Science and Technology.

■ BLDC 모터의 내부 구조

- A,B,C 3개의 코일
- h1, h2, h3개의 hall 센서
- Rotor에 영구자석



(a)



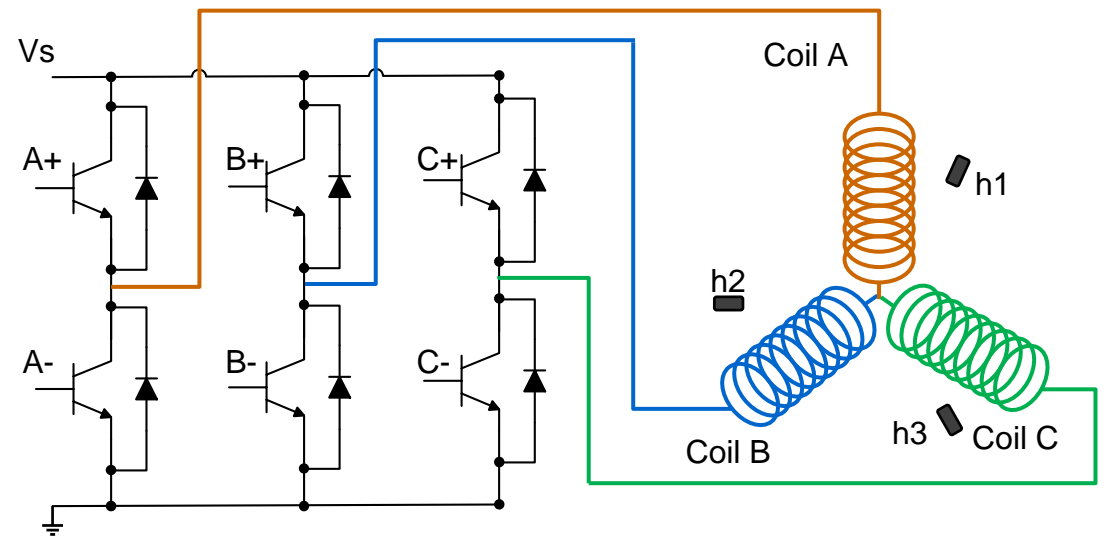
(b)

BLDC 모터의 구동

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

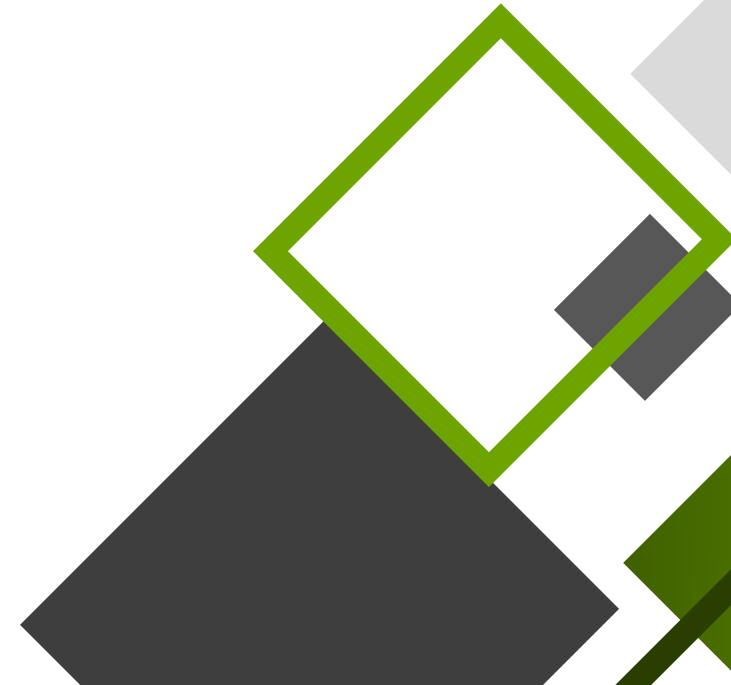
■ ESC의 회로

- ABC 코일을 순차적으로 전류를 흘려 구동
- Sensor-less BLDC motor control
 - hall센서를 사용하지 않는
 - Sensor-less BLDC





ATmega2560의 타이머



ATmega 2560의 타이머

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) 6개가 있음.
- 내부 또는 외부 펄스를 count함.
- 용도에 맞게 pre-scale을 이용하여 속도조절

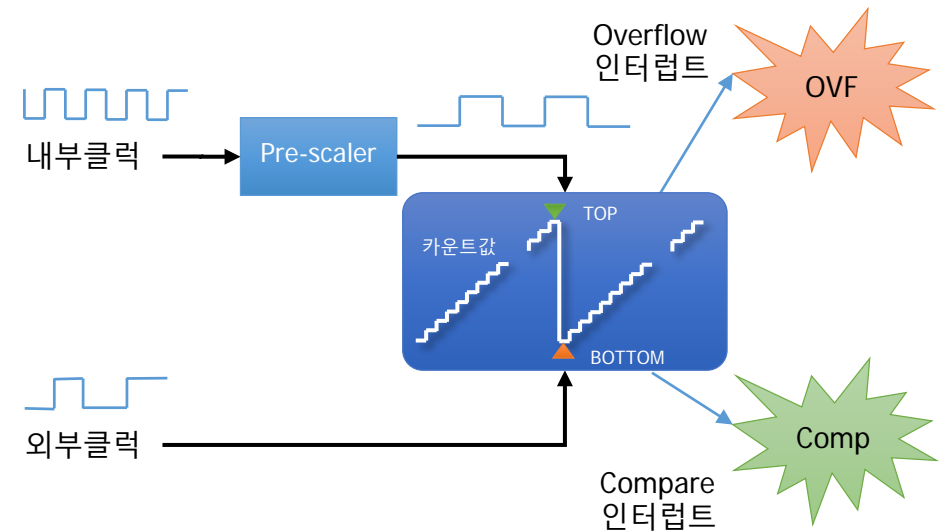
■ 지원 모드

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

■ 인터럽트

- Overflow Interrupt
- Compare Interrupt
- Input Capture Interrupt

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

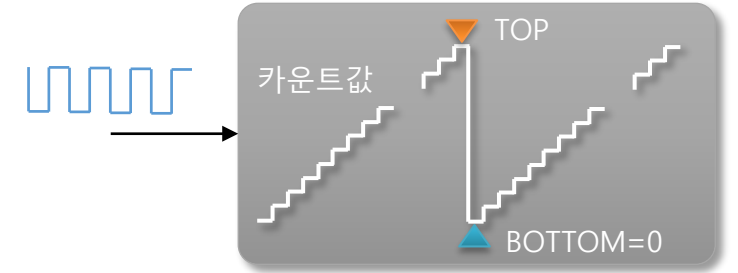


16Bit Timer in ATmega2560

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

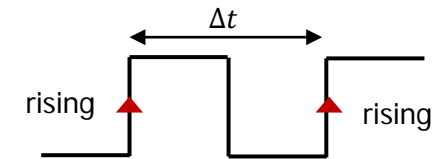
■ 주요 모드

- **일반모드:** 일반 계수 또는 타이머용
 - 최대 65535 까지 카운트 (Top=0xFFFF)
- **CTC모드:** 가변 주파수를 출력할 때 사용
 - Top = OCRnA 또는 ICRn
- **고속 PWM:** 고속의 PWM을 발생
- **위상정정 PWM:** 위상을 유지하는 위상정정 PWM 발생
- **위상-주파수 정정 PWM:** 가변 주파수인 위상정정 PWM



■ Input Capture 기능

- **ICPn** 핀에 입력되는 펄스의 상승 하강 edge 검출
- 그 때의 카운트 값을 저장 → edge 간의 시간간격 측정



16Bit Timer in ATmega2560

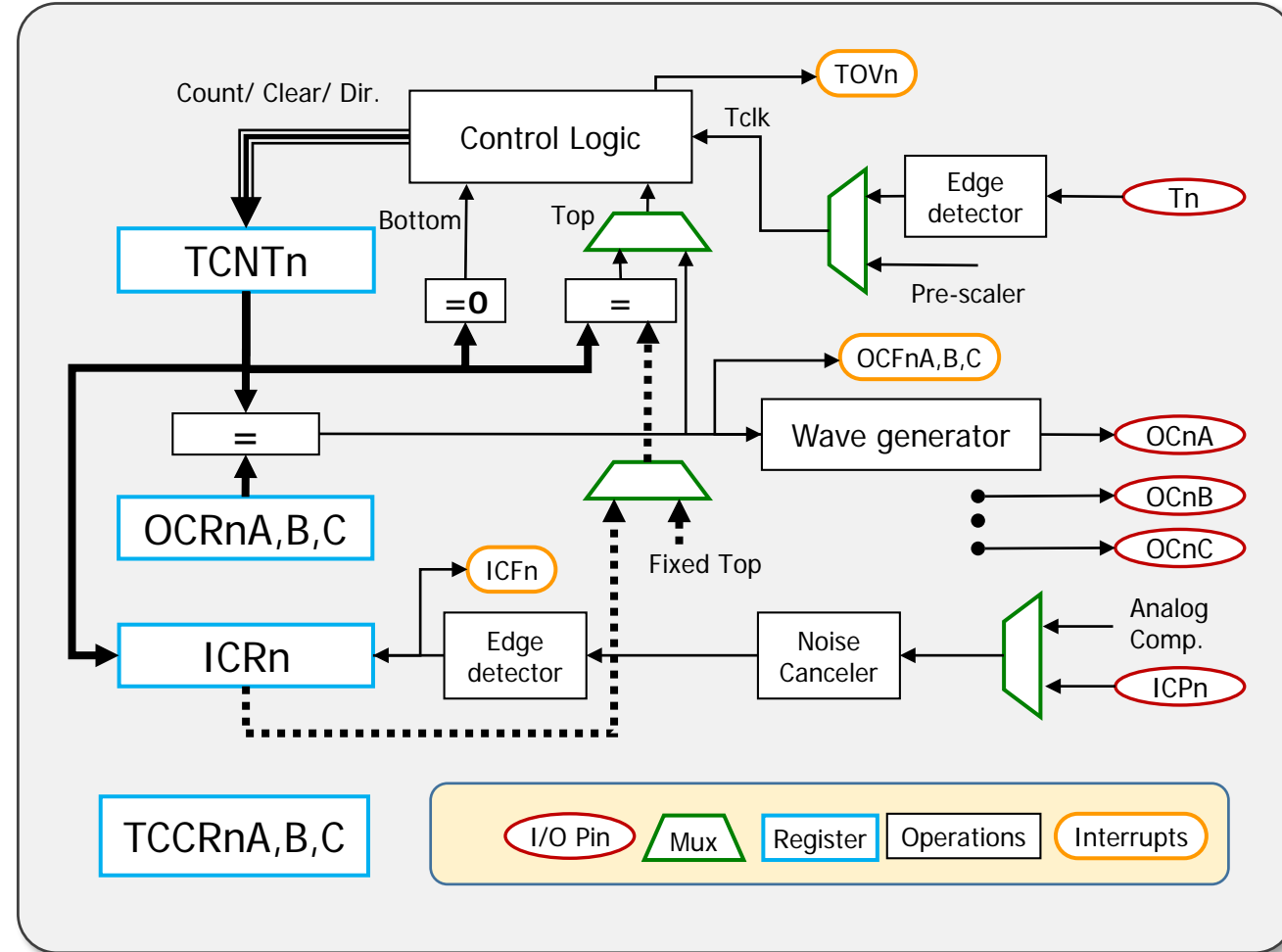
Dept. of Mechanical System Design, Seoul National University of Science and Technology.

Registers

- **TCNTn:**
 - 카운트값 저장
- **OCRnA, B, C:**
 - 비교값 저장, TOP값 지정
- **ICRn:**
 - Input capture값 저장
 - TOP값 지정
- **TCCRnA, B, C:** 타이머 설정

입출력 핀

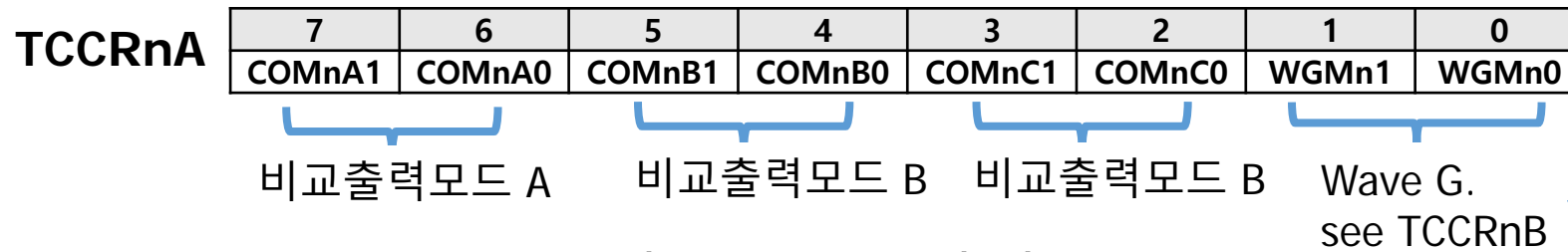
- Tn: 외부 펄스 입력핀
- OCnA,B,C: 비교 일치 출력핀
- ICP: Input Capture 핀



ATmega2560 Timer n의 레지스터

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

TCCRnA -Control register A



Mode	WGMn0:3
Normal	0
CTC	4,C
Fast PWM	5,6,7,E,F
PC PWM	1,2,3,A,B
F-PC PWM	8,9

- Wave generation 비트: 모드 결정
- 비교출력모드 비트: 모드에 따라 출력핀 OCnX 의 동작 규정

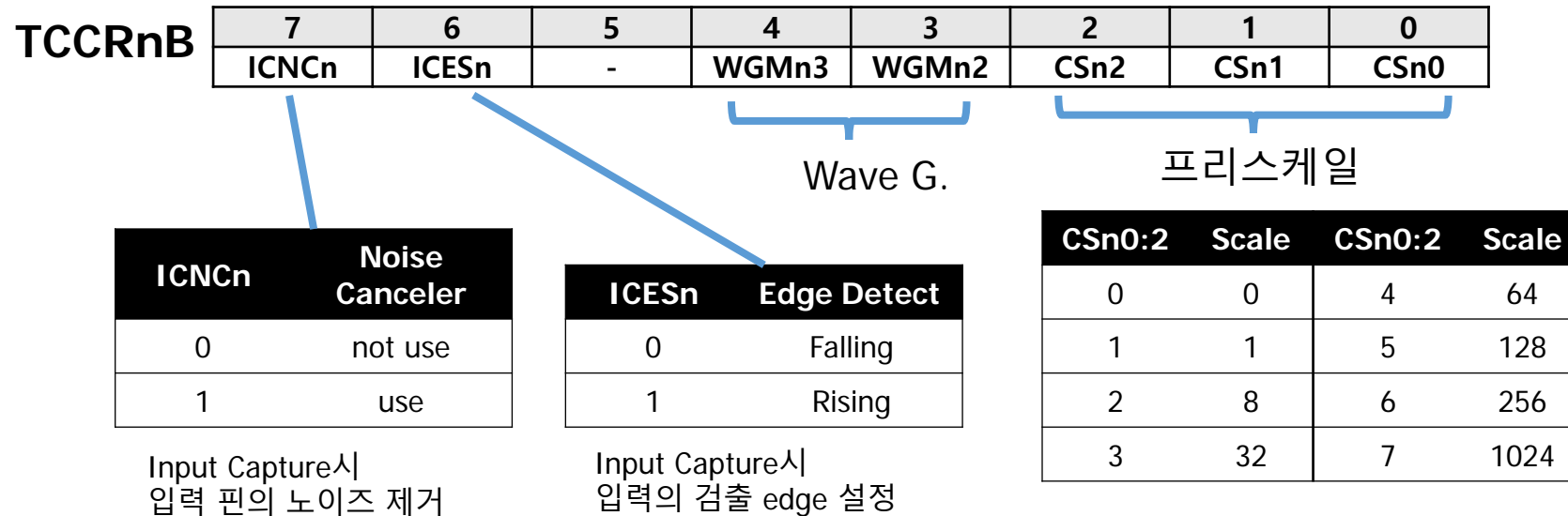
Normal, CTC mode	
COMnX0:1	OCnX 동작
0	Normal port
1	OCRnX 일치 → 반전
2	OCRnX 일치 → 0
3	OCRnX 일치 → 1

Fast PWM mode	
COMnX0:1	OCnX 동작
0	Normal port
1	OCRnX 일치 → 반전 (WGMn=F,E인 경우만)
2	OCRnX 일치 → 0, TOP → 1
3	OCRnX 일치 → 1, TOP → 0

ATmega2560 Timer n의 레지스터

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

■ TCCRnB -Control register B



ATmega2560 Timer n의 레지스터

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

■ TCCRnC -Control register C

TCCRnC	7	6	5	4	3	2	1	0
	FOCnA	FOCnB	FOCnC	-	-	-	-	-

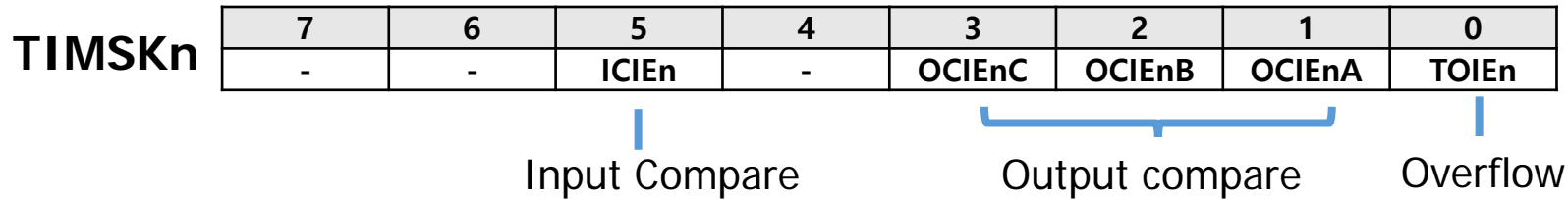
Force Output Compare:

- non-PWM에서 동작
- 1로 만들면 즉각적 compare 실행
- COMnx1:0에 따라
OCnA/OCnB/OCnC가 변화

ATmega2560 Timer n의 레지스터

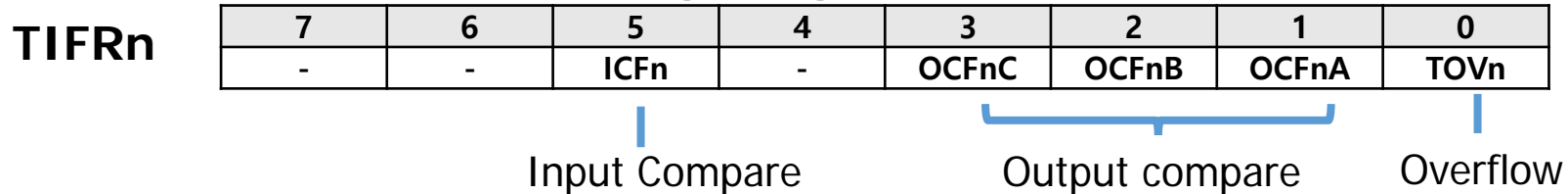
Dept. of Mechanical System Design, Seoul National University of Science and Technology.

TIMSKn - Interrupt mask



- 해당 bit에 1을 쓰면 인터럽트 활성화

TIFRn - Interrupt flag register



- 해당 Event가 발생하면 1이 됨
- 플래그에 1을 써주면 지워지고, ISR이 호출되면 자동으로 지워짐.

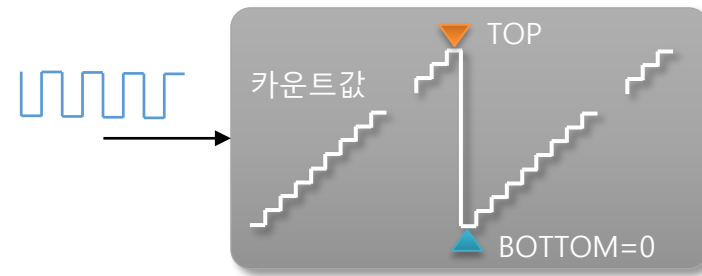
ATmega2560 Timer2의 레지스터

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

■ Fast PWM 모드인 경우 TOP의 설정

- TOP 값은 WGMn0:3 으로 지정

Mode	WGMn0:3	TOP
Fast PWM	5	FF (8bit)
	6	1FF (9bit)
	7	3FF (11bit)
	E	ICRn
	F	OCRnA



Timer1 PWM 출력 설정

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

■ Timer1을 사용

- OC1B, OC1A 해당 핀 출력으로 할당

```
pinMode(12, OUTPUT); // OUT1 (PB6/OC1B)
pinMode(11, OUTPUT); // OUT2 (PB5/OC1A)
```

- TCCR1A, TCCR1B

- Fast PWM모드에서 TOP을 ICR1으로 하려면 WGM → 0x0E (1110)
- Pre-scale CS1=(010) → 8배, $8/16,000,000 = 0.5 \mu s$

```
TCCR1A = ((1 << WGM11));
TCCR1B = (1 << WGM13) | (1 << WGM12) | (1 << CS11);
```

- ICR1: 50Hz를 만들려면 $TOP = 20ms / 0.5 \mu s = 40000$

```
ICR1 = 40000; // 0.5us tick => 50hz freq.
OCR1A = 0xFFFF; OCR1B = 0xFFFF; // disable
```

APM핀	Timer	OCX#	AVR핀
OUT1	1	OCB1	PB6(12)
OUT2	1	OCA1	PB5(11)
OUT3	4	OCC4	PH5(8)
OUT4	4	OCB4	PH4(7)
OUT5	4	OCA4	PH3(6)
OUT6	3	OCC3	PE5(3)
OUT7	3	OCB3	PE4(2)
OUT8	3	OCA3	PE3(5)

Timer3 PWM 출력 설정

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

■ Timer3을 사용(Timer4도 동일)

- OC3C, OC3B, OC1A 해당 핀 출력으로 할당

```
pinMode(3,OUTPUT);pinMode(2,OUTPUT);pinMode(5,OUTPUT);
```

• TCCR3A, TCCR3B

- Fast PWM모드에서 TOP을 ICR3으로 하려면 WGM→ 0x0E (1110)
- Pre-scale CS3=(010) → 8배, $8/16,000,000 = 0.5 \mu s$

```
TCCR3A = ((1<<WGM31));  
TCCR3B = (1<<WGM33) | (1<<WGM32) | (1<<CS31);
```

• ICR3: 50Hz를 만들려면 $TOP=20ms / 0.5 \mu s = 40000$

```
ICR3 = 40000; // 0.5us tick => 50hz freq.  
OCR3A = 0xFFFF; OCR3B = 0xFFFF; OCR3C = 0xFFFF;
```

Timer5 PWM, Input Capture 설정

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

■ Timer5 사용

- OC5B, OC5C 핀 출력, ICP5 핀 입력

```
pinMode(48, INPUT); // PPM Input (PL1/ICP5)
pinMode(45, OUTPUT); // OUT10 (PL4/OC5B)
pinMode(44, OUTPUT); // OUT11 (PL5/OC5C)
```

- TCCR5A, TCCR5B

- Fast PWM모드, TOP을 OCR5A로 하려면 WGM → 0x0F (1111)
- Pre-scale CS5=(010) → 8배, $8/16,000,000 = 0.5 \mu s$
- Input Capture 시 Rising Edge검출 ICES5=1

```
TCCR5A = ((1<<WGM50) | (1<<WGM51));
TCCR5B = ((1<<WGM53) | (1<<WGM52) | (1<<CS51) | (1<<ICES5));
```

Timer5 PWM, Input Capture 설정

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

- TOP 설정
 - TOP= OCR5A
 - 50Hz를 만들려면 $OCR5A = 20ms / 0.5\mu s = 40000$

```
OCR5A = 40000; // 0.5us tick => 50hz freq. TOP is OCR5A
```

- TIMSK5
 - Input Capture 인터럽트 활성화

```
TIMSK5 |= (1<<ICIE5);
```

PWM Test Code

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

- Output1과 Output1를 각각 1000, 2000 으로 PWM 출력
 - PWM 주기 20ms,
 - Output1: on-duty 1ms, Output2: on-duty 2ms

timerTest.ino

```
#include "timerSetup.h"
void setup() {
  Serial.begin(115200);
  initTimers();
  enableOut(0); enableOut(1);
  outputCh(0, 1000);
  outputCh(1, 2000);
}
void loop() {
}
```

timerSetup.h

```
#ifndef __TIMERSETUP_H__
#define __TIMERSETUP_H__
#define NUM_CHANNELS 8
#define MIN_PULSEWIDTH 900
#define MAX_PULSEWIDTH 2100
#include <Arduino.h>
void outputCh(unsigned char ch, uint16_t pwm);
void enableOut(uint8_t ch);
void disableOut(uint8_t ch);
void initTimers();
#endif
```

PWM Test Code

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

code

timerSetup.cpp[1/2]

```
#include "timerSetup.h"
void initTimers(){
    //TIMER1: WGM 1110, TOP=ICR1, CS=2, OUT1,OUT2
    pinMode(12,OUTPUT); // OUT1 (PB6/OC1B)
    pinMode(11,OUTPUT); // OUT2 (PB5/OC1A)
    TCCR1A = ((1<<WGM11));
    TCCR1B = (1<<WGM13) | (1<<WGM12) | (1<<CS11);
    ICR1 = 40000; // 0.5us tick => 50hz freq
    OCR1A = 0xFFFF; OCR1B = 0xFFFF; //disable
}
void outputCh(unsigned char ch, uint16_t pwm){
    pwm=constrain(pwm,MIN_PULSEWIDTH,MAX_PULSEWIDTH);
    pwm<=1; // pwm*2; Register 2000 ~ 4000
    switch(ch) {
        case 0: OCR1B=pwm; break; // out1
        case 1: OCR1A=pwm; break; // out2
    }
}
```

PWM Test Code

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

code

timerSetup.cpp[1/2]

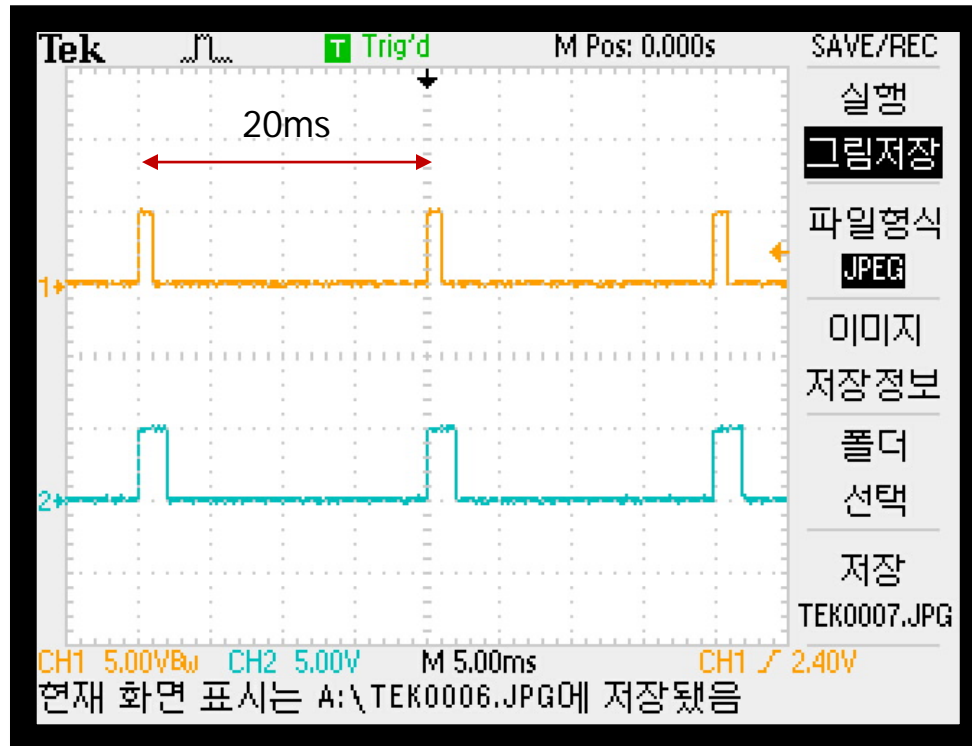
```
void enableOut(uint8_t ch){
    switch(ch) {
        case 0: TCCR1A |= (1<<COM1B1); break; // CH_1 : OC1B
        case 1: TCCR1A |= (1<<COM1A1); break; // CH_2 : OC1A
    }
}

void disableOut(uint8_t ch) {
    switch(ch) {
        case 0: TCCR1A &= ~(1<<COM1B1); break; // CH_1 : OC1B
        case 1: TCCR1A &= ~(1<<COM1A1); break; // CH_2 : OC1A
    }
}
```

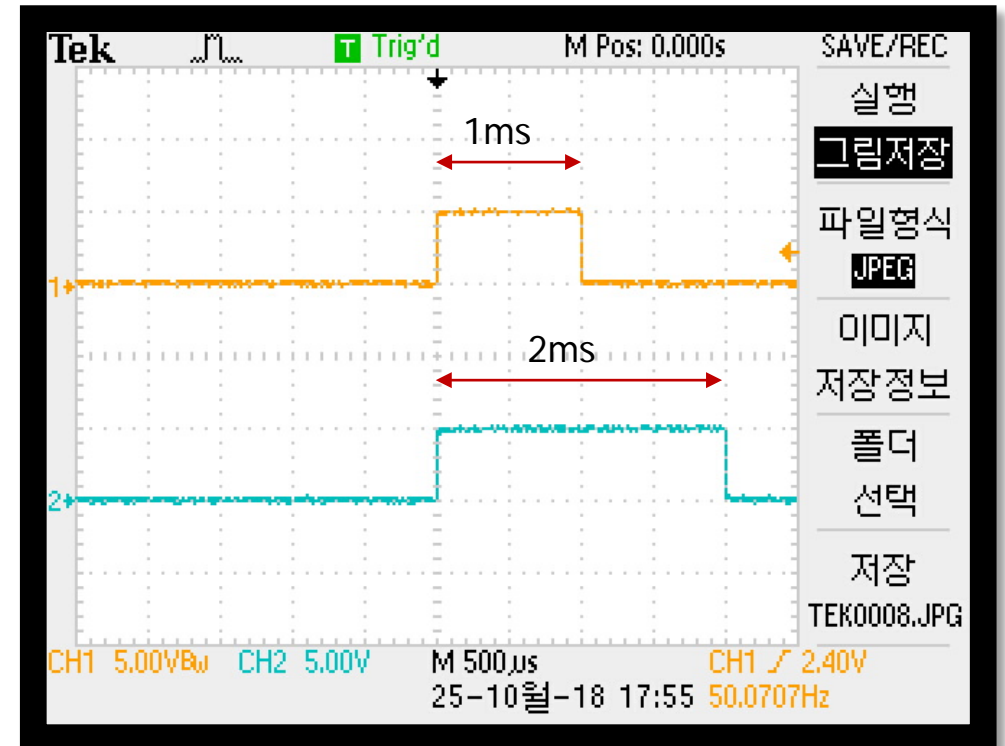

PWM Test Code

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

PWM 측정



PWM 주파수: 50Hz



CH1: PWM = 1000, CH1: PWM = 2000,

입력 Ch 설정

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

■ RC 출력 Ch배정

■ PWM 의 최소 최대값 배정

- us 단위로 지정

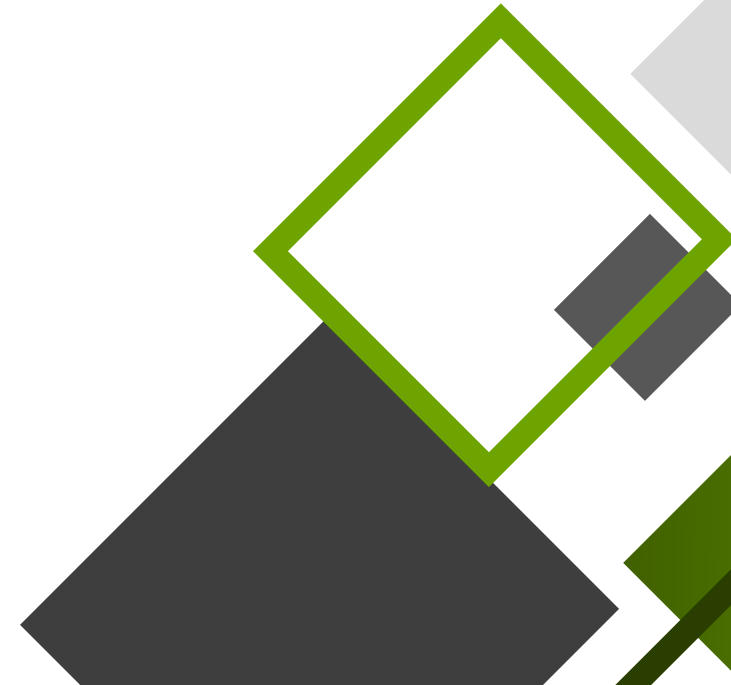
```
#define MIN_PULSEWIDTH 900
```

```
#define MAX_PULSEWIDTH 2100
```

No.	Channel #	Timer #	OCnX: X
0	1	Timer1	B
1	2	Timer1	A
2	3	Timer4	C
3	4	Timer4	B
4	5	Timer4	A
5	6	Timer3	C
6	7	Timer3	B
7	8	Timer3	A
9	10	Timer5	B
10	11	Timer5	C



RC와 PPM 입력

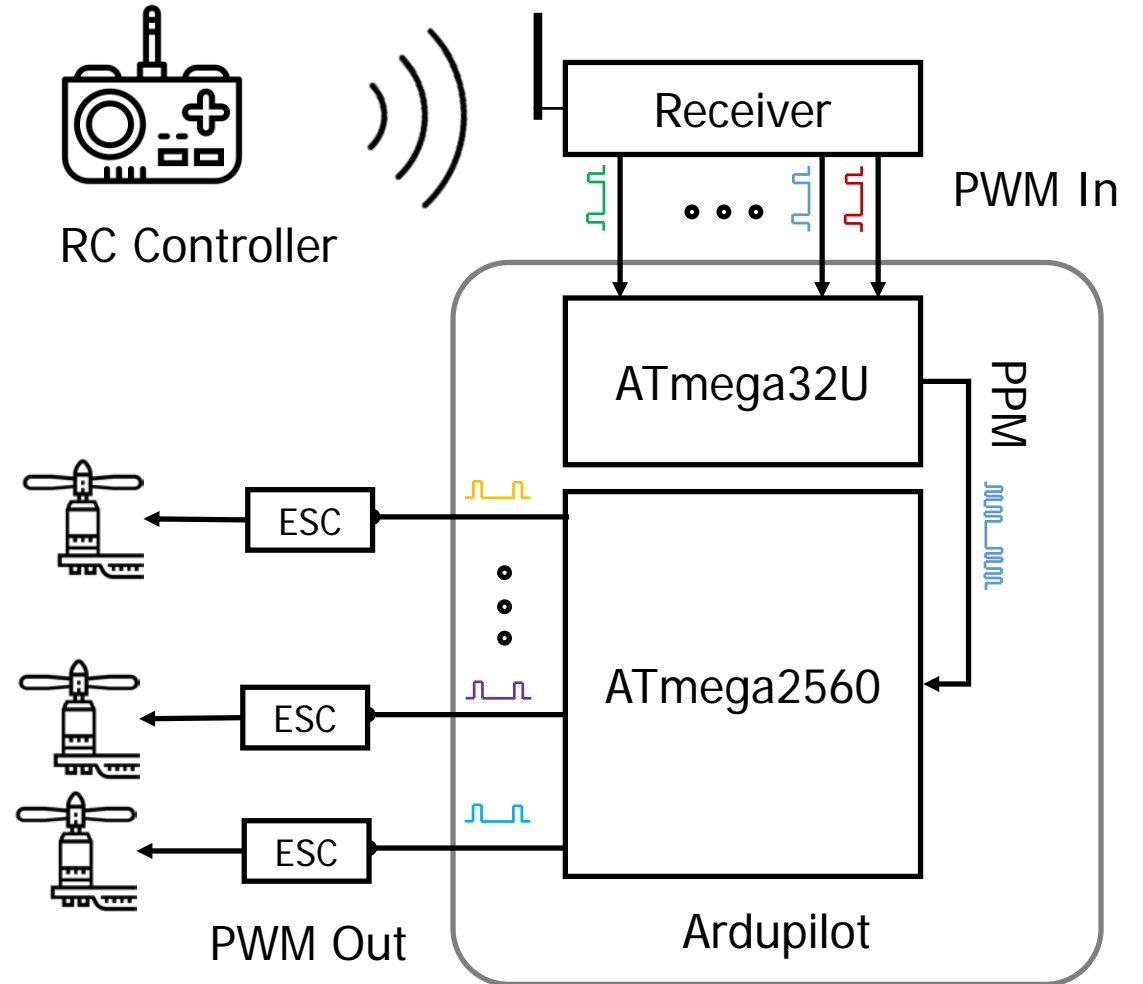


RC Channel 구성

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

■ RC 전송 순서

- RC 조정기는 n 채널의 정보를 포함하는 전파를 발생
- 전파를 수신한 수신기는 복조를 통하여 채널 별로 PWM 발생하여 Ardupilot에 입력
- 32U가 PWM을 PPM으로 변환하여 2560에 전달
- 2560이 PPM을 받아 적절한 제어와 변환을 한 후 다시 채널 별로 PWM 발생
- 각 모터의 ESC는 PWM에 따라 속도제어

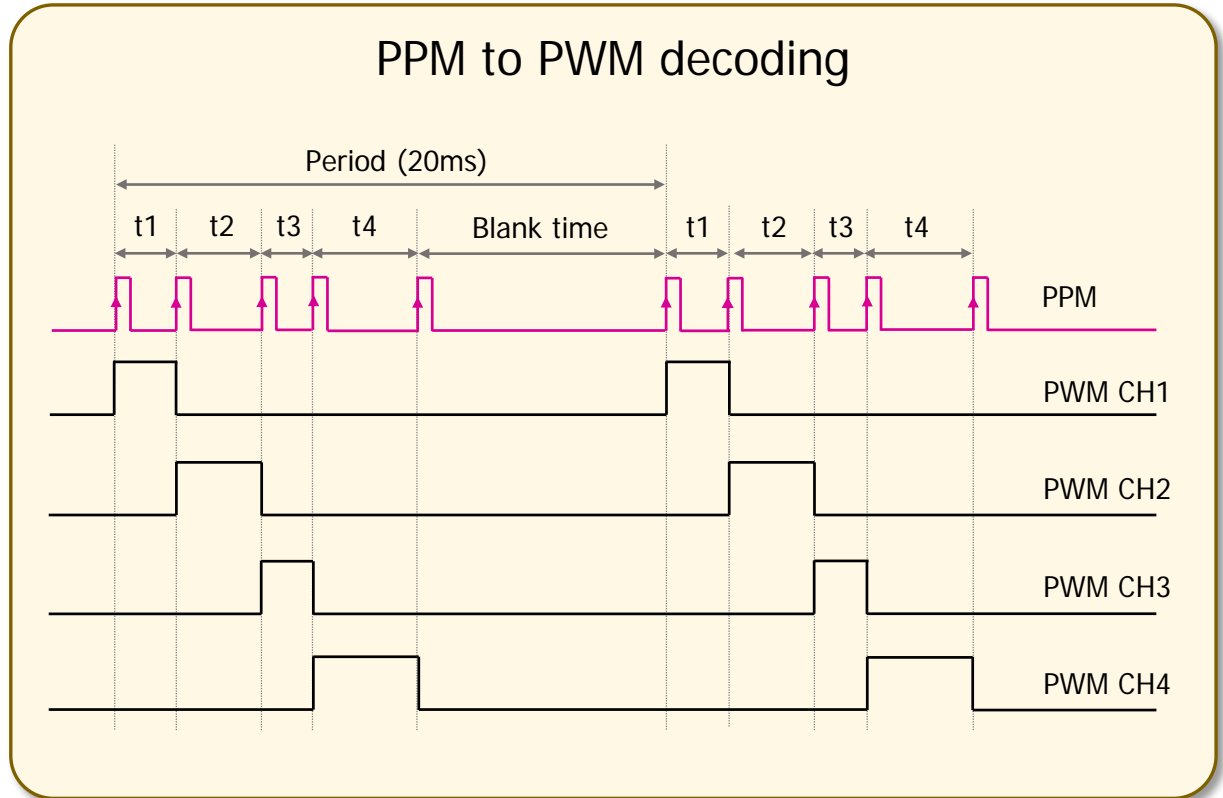


PPM과 PWM의 변환관계

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

■ PPM(Pulse Position Modulation)

- n 개의 PWM의 On-duty 정보들을 $n+1$ 개의 펄스의 위치로 표현
- 총 주기가 20ms
- 그림에서 보는 바와 같이 $n+1$ 개의 펄스들의 간격이 t_1, t_2, t_3, t_4
- 따라서,
 - $\text{On duty of CH } n = t_n$



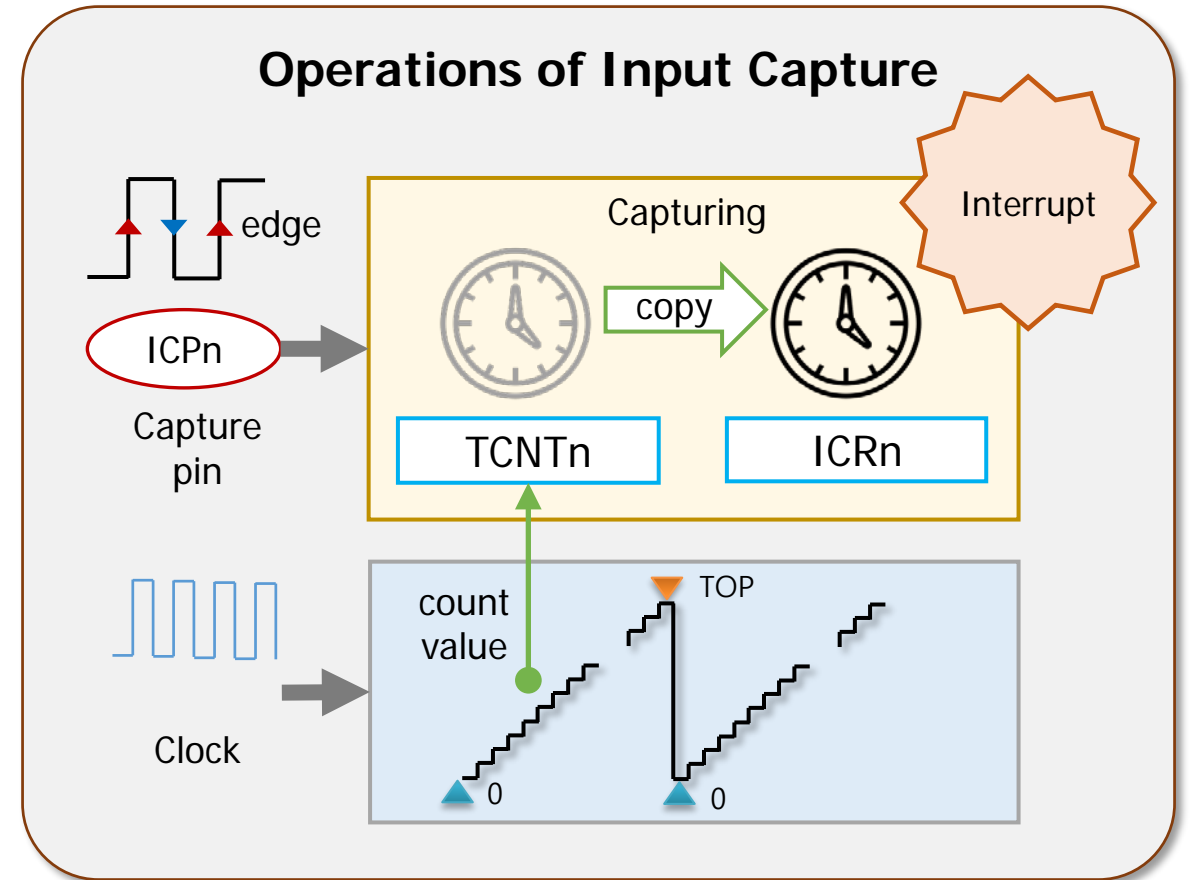
Input Capture 동작

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

■ 동작 과정

- 타이머에 Input capture 설정 후 다음 동작이 실행됨.
- Capture pin에서 edge 검출 됨.
- 현재 카운트 값 TCNTn이 ICRn에 복사됨
- 이 때 캡처 인터럽트가 기동되며 ICRn값을 읽어 별도 변수에 보관
- 시간 간격은 다음과 같음

$$\text{time interval} = \text{Time}_{\text{current}} - \text{Time}_{\text{previous}}$$



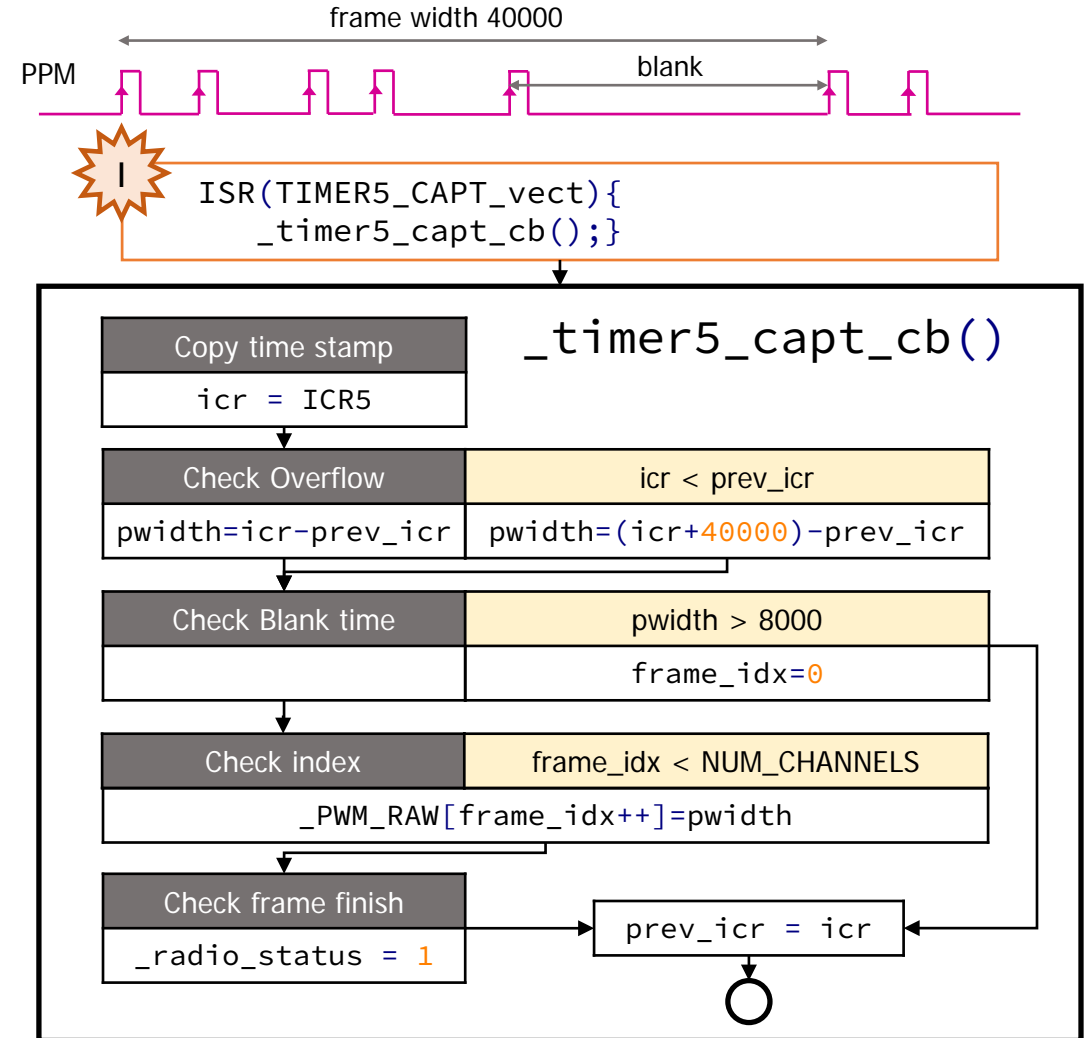
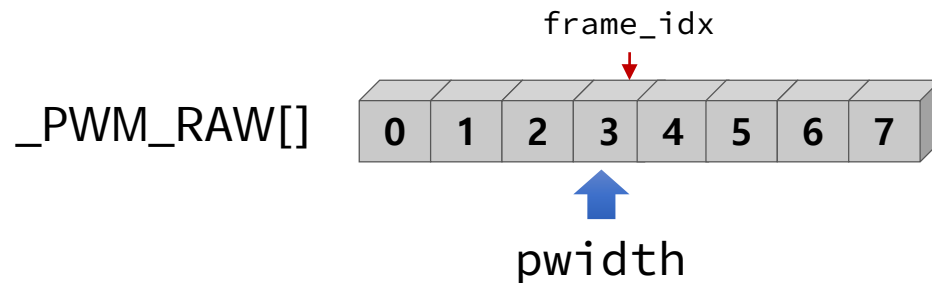
Capture interrupt 처리

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

주요 동작

주요 변수

- icr: time 임시 보관 변수
- prev_icr: 이전 time 보관 변수
- pwidth: 펄스간의 간격 보관
- frame_idx: 펄스 인덱스
- _PWM_RAW[] : 측정값을 보관하는 배열
- _radio_status: 측정 frame 완성 flag



Capture interrupt 처리

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

■ 코드

- inputCh() 함수
 - PWM_RAW[ch]에 저장된 PWM 값을 us단위로 반환
 - 저장 중에 인터럽트가 동작하여 값이 바뀌지 않도록 함.
 - cli() : 중단, sei(): 재개
 - 2로 나누어 us 단위로 변환
 - constrain()함수로 값 제한
 - 읽기 완료 표시
 - _radio_status = 0

variables

```
volatile uint16_t _PWM_RAW[NUM_CHANNELS]
    = {2400,2400,2400,2400,2400,2400,2400,2400};
volatile uint8_t _radio_status=0;
```

inputCh

```
uint16_t inputCh(unsigned char ch){
    uint16_t result;
    cli();    //disable all interrupts
    result = _PWM_RAW[ch];
    sei();    //enable all interrupts again
    result >>= 1; // convert to us unit
    result = constrain(result,MIN_PULSEWIDTH,MAX_PULSEWIDTH);
    _radio_status = 0; // Radio channel read
    return result;
}
```


Capture interrupt 처리

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

■ 코드

- GetState() 함수
 - 프레임 읽기 완료면 1
 - 읽기 중에는 0
 - _radio_status 반환
- ISR(TIMER5_CAPT_vect)
 - Timer5의 Capture interrupt ISR

getState

```
unsigned char getState(void){  
    return _radio_status;  
}
```

Interrupt Service Routine

```
ISR(TIMER5_CAPT_vect) {  
    _timer5_capt_cb();  
}
```

Capture interrupt 처리

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

■ 코드

_timer5_capt_cb

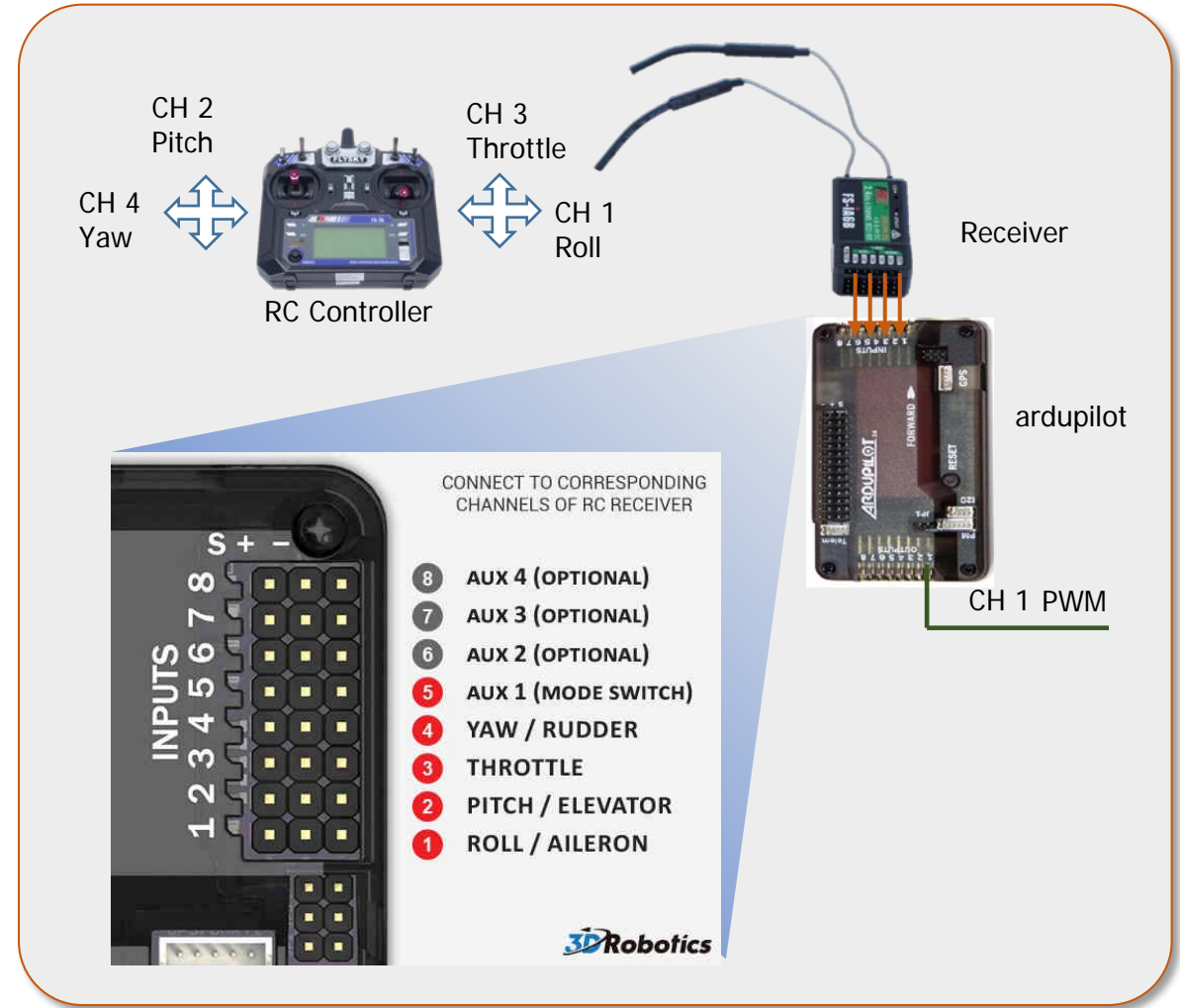
```
void _timer5_capt_cb(void) {
    static uint16_t prev_icr;
    static uint8_t frame_idx;
    uint16_t icr, pwidth;
    icr = ICR5; // copy time stamp
    if ( icr < prev_icr ) { // TOP = 40000
        pwidth = ( icr + 40000 ) - prev_icr;
    } else {
        pwidth = icr - prev_icr;
    }
    if ( pwidth > 8000 ) { // if is blank time
        frame_idx=0;
    } else {
        if ( frame_idx < NUM_CHANNELS ) {
            _PWM_RAW[ frame_idx++ ] = pwidth;
            if (frame_idx >= NUM_CHANNELS) {
                _radio_status = 1; // One frame finished
            }
        }
    }
    prev_icr = icr; // Save icr for next call
}
```

PPM decoding 실습

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

■ 실험장치 구성

- Binding 된 송수신기를 준비
- 송신기의 joystick 사용
- Receiver의 CH1~4 를 ardupilot Input 1~4로 연결
 - CH1: ROLL, CH2:PITCH,
 - CH3:YAW, CH3: THROTTLE
- 소프트웨어로 PPM 으로부터 PWM 을 구하고 OUTPUT 1~4로 다시 PWM 출력



전체코드

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

■ 코드

timerCapture1.ino

```
#include "timerSetup.h"
void setup() {
    Serial.begin(115200);
    initTimers();
    enableOut(0); enableOut(1); enableOut(2);
    enableOut(3); enableOut(4);
}
void loop() {
    if (getState()){
        Serial.print(inputCh(0));Serial.print(",");
        Serial.print(inputCh(1));Serial.print(",");
        Serial.print(inputCh(2));Serial.print(",");
        Serial.print(inputCh(3));Serial.print(",");
        Serial.print(inputCh(4));Serial.print("\n");
        outputCh(0, inputCh(0));outputCh(1, inputCh(1));
        outputCh(2, inputCh(2));outputCh(3, inputCh(3));
    }
    delay(10);
}
```

전체코드

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

■ 코드

timerSetup.h

```
#ifndef __TIMERSETUP_H__
#define __TIMERSETUP_H__
#define NUM_CHANNELS 8
#define MIN_PULSEWIDTH 900
#define MAX_PULSEWIDTH 2100
#include <Arduino.h>
void outputCh(unsigned char ch, uint16_t pwm);
void enableOut(uint8_t ch);
void disableOut(uint8_t ch);
void initTimers();
unsigned char getState(void);
uint16_t inputCh(unsigned char ch);
#endif
```

전체코드

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

코드

timerSetup.cpp [1/6]

```
#include "timerSetup.h"
volatile uint16_t _PWM_RAW[NUM_CHANNELS]
    = {2400,2400,2400,2400,2400,2400,2400,2400};
volatile uint8_t _radio_status=0;
void outputCh(unsigned char ch, uint16_t pwm){
    pwm=constrain(pwm,MIN_PULSEWIDTH,MAX_PULSEWIDTH);
    pwm<=1;    // pwm*2; Register 2000 ~ 4000
    switch(ch)  {
        case 0: OCR1B=pwm; break;    // out1
        case 1: OCR1A=pwm; break;    // out2
        case 2: OCR4C=pwm; break;    // out3
        case 3: OCR4B=pwm; break;    // out4
        case 4: OCR4A=pwm; break;    // out5
        case 5: OCR3C=pwm; break;    // out6
        case 6: OCR3B=pwm; break;    // out7
        case 7: OCR3A=pwm; break;    // out8
        case 9: OCR5B=pwm; break;    // out10 x
        case 10: OCR5C=pwm; break;   // out11 x
    }
}
```

전체코드

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

■ 코드

timerSetup.cpp [2/6]

```
void enableOut(uint8_t ch){
    switch(ch) {
        case 0: TCCR1A |= (1<<COM1B1); break; // CH_1 : OC1B
        case 1: TCCR1A |= (1<<COM1A1); break; // CH_2 : OC1A
        case 2: TCCR4A |= (1<<COM4C1); break; // CH_3 : OC4C
        case 3: TCCR4A |= (1<<COM4B1); break; // CH_4 : OC4B
        ...
    }
}

void disableOut(uint8_t ch) {
    switch(ch) {
        case 0: TCCR1A &= ~(1<<COM1B1); break; // CH_1 : OC1B
        case 1: TCCR1A &= ~(1<<COM1A1); break; // CH_2 : OC1A
        case 2: TCCR4A &= ~(1<<COM4C1); break; // CH_3 : OC4C
        case 3: TCCR4A &= ~(1<<COM4B1); break; // CH_4 : OC4B
        ...
    }
}
```

전체코드

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

■ 코드

timerSetup.cpp [3/6]

```
void initTimers(){
    //TIMER1: WGM 1110, TOP=ICR1, CS=2, OUT1,OUT2
    pinMode(12,OUTPUT); // OUT1 (PB6/OC1B)
    pinMode(11,OUTPUT); // OUT2 (PB5/OC1A)
    TCCR1A = (1<<WGM11);
    TCCR1B = (1<<WGM13) | (1<<WGM12) | (1<<CS11);
    ICR1 = 40000; // 0.5us tick => 50hz freq
    OCR1A = 0xFFFF; OCR1B = 0xFFFF; //disable
    //TIMER4: WGM 1110, TOP=ICR4, CS=2, OUT3,OUT4,OUT5
    pinMode(8,OUTPUT); // OUT3 (PH5/OC4C)
    pinMode(7,OUTPUT); // OUT4 (PH4/OC4B)
    pinMode(6,OUTPUT); // OUT5 (PH3/OC4A)
    TCCR4A = (1<<WGM41);
    TCCR4B = (1<<WGM43) | (1<<WGM42) | (1<<CS41);
    OCR4A = 0xFFFF; OCR4B = 0xFFFF; OCR4C = 0xFFFF;
    ICR4 = 40000; // 0.5us tick => 50hz freq
```


전체코드

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

코드

timerSetup.cpp [4/6]

```
//TIMER3: WGM 1110, TOP=ICR3, CS=2, OUT6,OUT7,OUT8
pinMode(3,OUTPUT); // OUT6 (PE5/OC3C)
pinMode(2,OUTPUT); // OUT7 (PE4/OC3B)
pinMode(5,OUTPUT); // OUT8 (PE3/OC3A)
TCCR3A = ((1<<WGM31));
TCCR3B = (1<<WGM33) | (1<<WGM32) | (1<<CS31);
OCR3A = 0xFFFF; OCR3B = 0xFFFF; OCR3C = 0xFFFF;
ICR3 = 40000; // 0.5us tick => 50hz freq
//TIMER5: WGM 1111, TOP=OCR5A, CS=2, PPM,OUT10,OUT11
pinMode(48, INPUT); // PPM Input(PL1/ICP5)
pinMode(45, OUTPUT); // OUT10 (PL4/OC5B)
pinMode(44, OUTPUT); // OUT11 (PL5/OC5C)
TCCR5A = ((1<<WGM50) | (1<<WGM51));
TCCR5B = ((1<<WGM53) | (1<<WGM52) | (1<<CS51) | (1<<ICES5)); //Rising
OCR5A = 40000; // 0.5us tick => 50hz freq.
TIMSK5 |= (1<<ICIE5); // Enable Input Capture interrupt
}
```

전체코드

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

■ 코드

timerSetup.cpp [5/6]

```
void _timer5_capt_cb(void) {
    static uint16_t prev_icr;
    static uint8_t frame_idx;
    uint16_t icr, pwidth;
    icr = ICR5;    // copy time stamp
    if ( icr < prev_icr ) {    // TOP = 40000
        pwidth = ( icr + 40000 ) - prev_icr;
    } else {
        pwidth = icr - prev_icr;
    }
    if ( pwidth > 8000 ) { // if is blank time
        frame_idx=0;
    } else {
        if ( frame_idx < NUM_CHANNELS ) {
            _PWM_RAW[ frame_idx++ ] = pwidth;
            if (frame_idx >= NUM_CHANNELS) {
                _radio_status = 1; // One frame finished
            }
        }
    }
    prev_icr = icr; // Save icr for next call.
}
```

전체코드

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

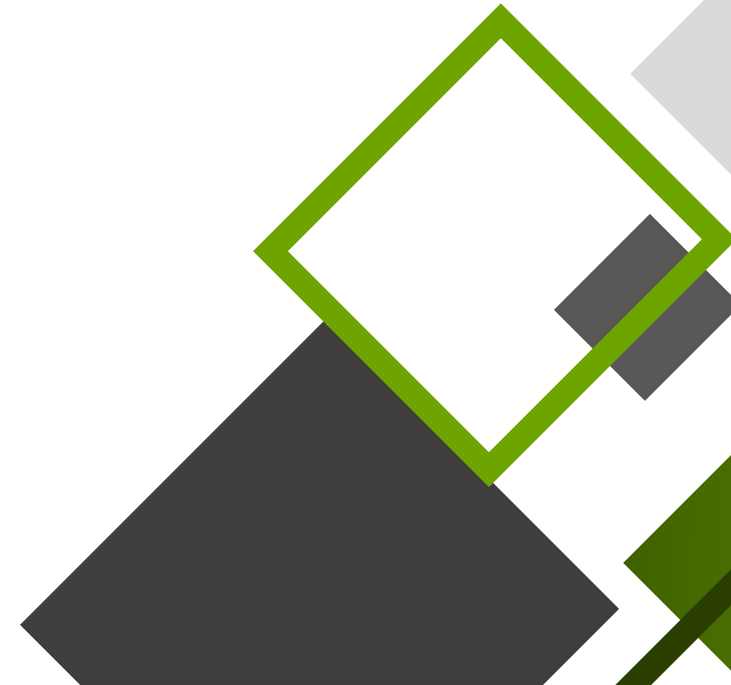
■ 코드

timerSetup.cpp [6/6]

```
unsigned char getState(void){
    return _radio_status;
}
uint16_t inputCh(unsigned char ch){
    uint16_t result;
    cli();    //disable all interrupts
    result = _PWM_RAW[ch];
    sei();    //enable all interrupts again
    result >>= 1; // convert to us unit
    result = constrain(result,MIN_PULSEWIDTH,MAX_PULSEWIDTH);
    _radio_status = 0; // Radio channel read
    return result;
}
ISR(TIMER5_CAPT_vect) {
    _timer5_capt_cb();
}
```



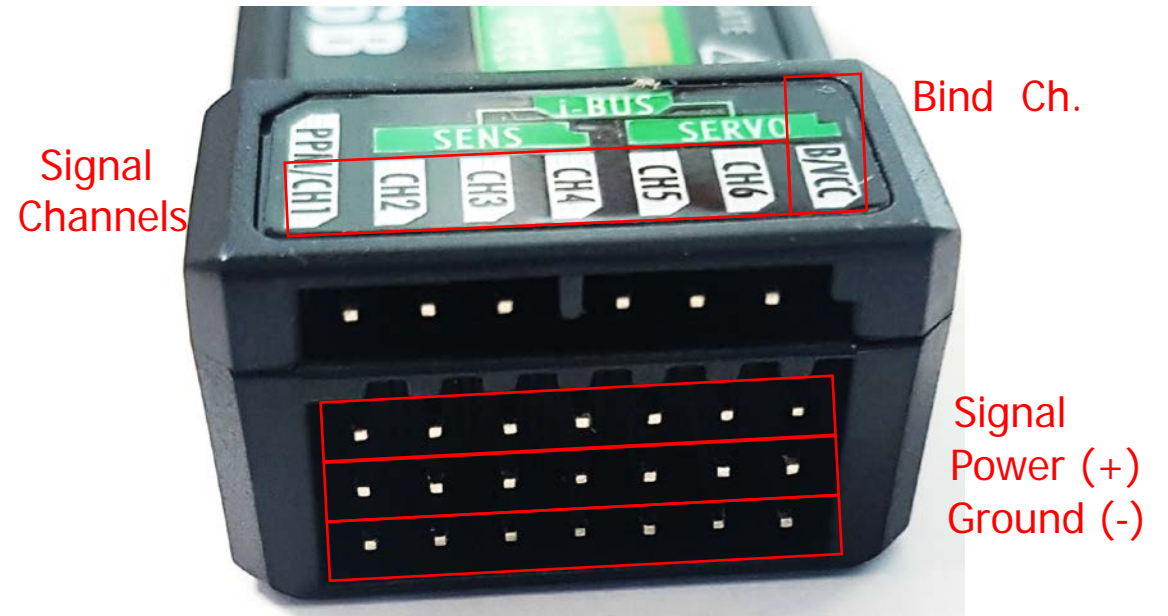
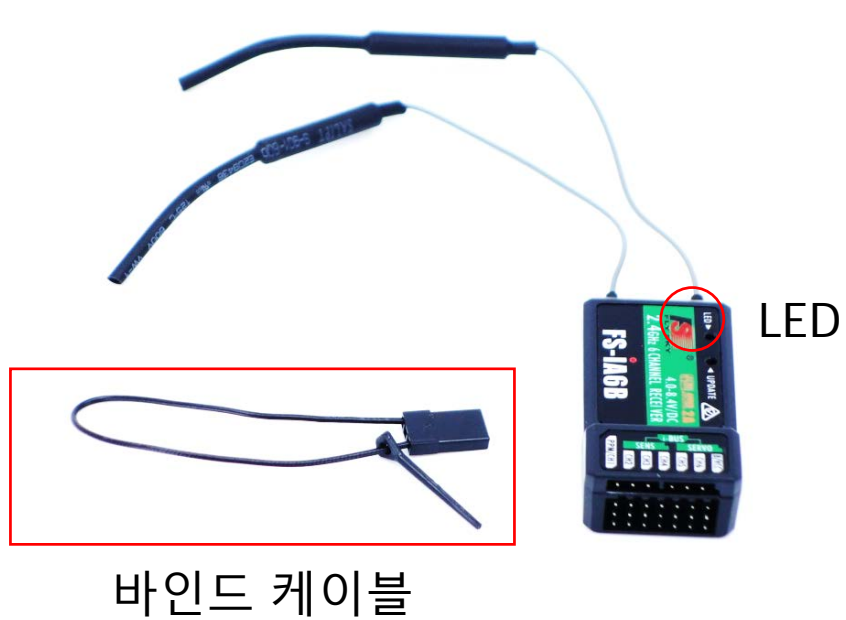
송수신기 바인딩



송수신기 바인딩

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

■ 수신기 주요 부분



송수신기 바인딩

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

■ 바인딩 절차

- ① 바인딩 케이블을 수신기의 바인드 모드 채널에 연결
- ② 수신기의 전원 ON
- ③ 수신기의 LED가 빠르게 깜빡이면, 수신기가 바인딩 모드로 작동 중임.
- ④ 조종기의 BIND KEY버튼을 누른 상태에서 조종기의 전원을 ON.
- ⑤ 수신기의 LED가 계속 켜지는 ON 상태가 되면 바인딩이 완료.
- ⑥ 조종기와 수신기의 전원을 끄고, 바인딩 케이블을 제거.
- ⑦ 이후 조종기와 수신기의 전원을 켜면 둘은 자동 연결됨.



실습결과

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



실습결과

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





THANK YOU

Powerpoint is a complete presentation graphic package it gives you everything you need to produce a professional-looking presentation