

# 위기상황에서 드론의 비상착륙 시스템 제작

국방디지털융합학과 공수교대  
(김경은, 박준혁, 원유겸, 이준혁, 허주영)

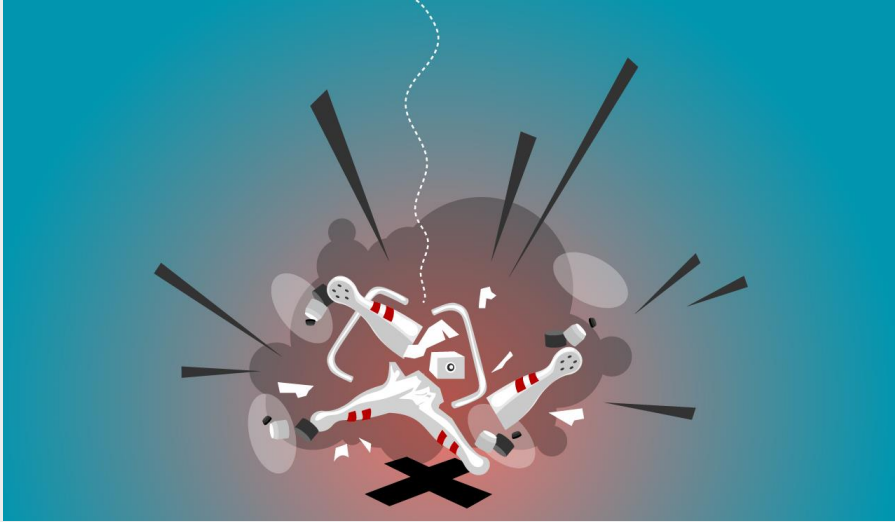
# 목차

## 01 주제 소개

연구 주제 및 목적  
선행연구 및 드론 소개

## 02 진행 상황

## 03 느낀점 및 향후 계획



기상상황, 배터리, 주파수 혼선 등의 문제로  
드론이 추락하는 경우 발생

→ 인명, 재산피해 및 동체 파손의 위험

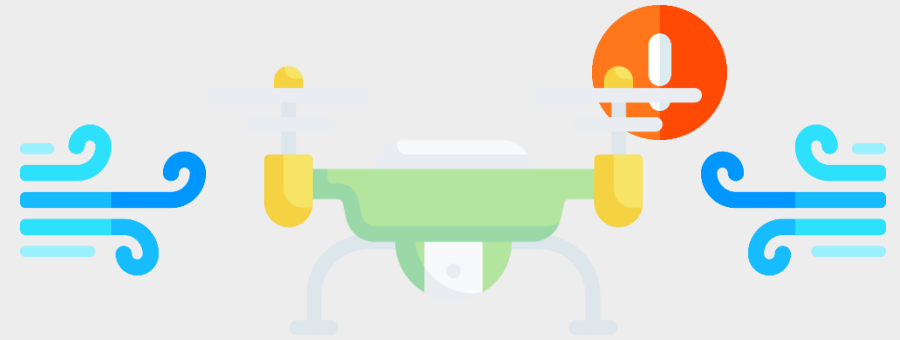
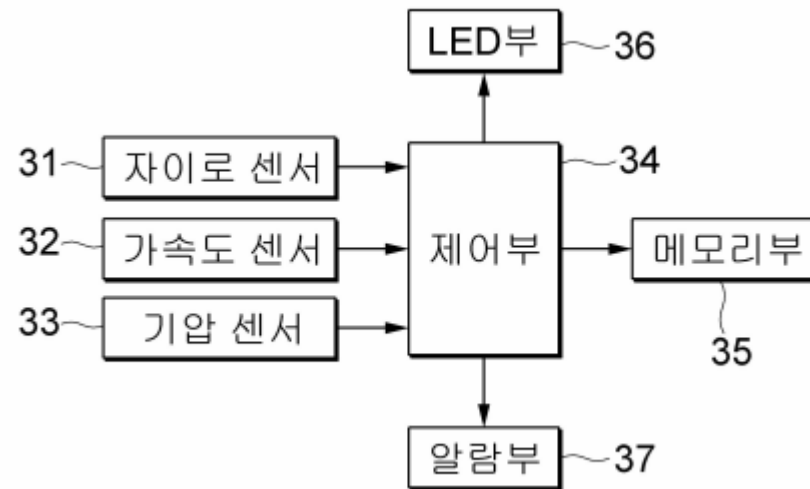
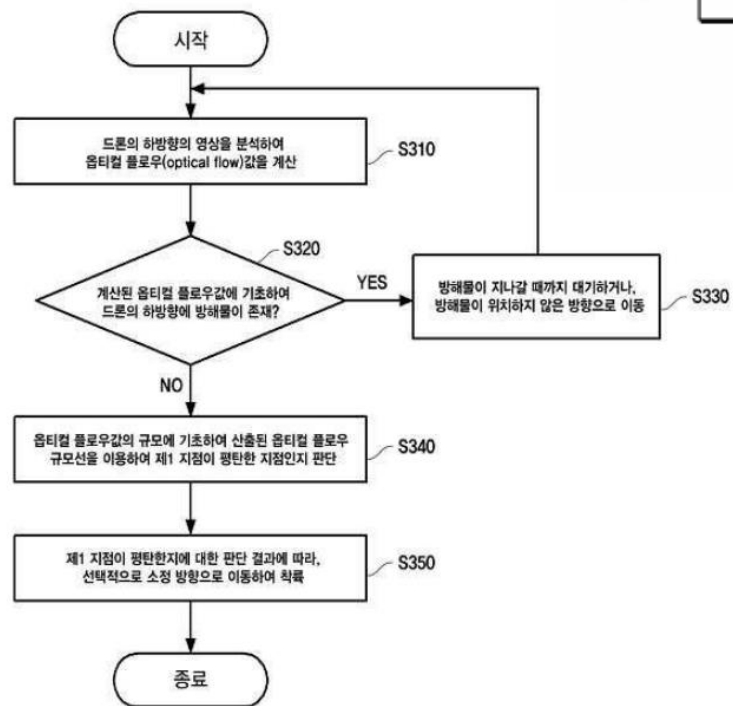


Table 1. Initial Condition of Trajectory

설정 변수	설정 변수 설명
$h_e$	플래어 제적(예: 지수함수)의 터치다운 이후의 최종 저지점
$x_{TD}$	
$\dot{h}_{TD}$	



조사 결과 대부분 GPS를 이용하여 특정 지점에 착륙하도록 하거나, 미리 착륙 범위를 정해놓은 경우가 많음. 카메라를 이용하여 착륙 지점을 결정하는 경우도 있다! → 우선 카메라나 GPS를 이용하지 않고, 센서 값을 이용하여 비상상황을 인지하고 착륙하려고 계획을 세움



## STM32CubeIDE

STMicroelectronics에서 배포  
microcontroller unit 통합 개발 환경 프로그램  
이클립스 프레임워크 기반, C/C++ 개발 플랫폼  
보드와는 ST-Link라는 디버거를 이용해 연결,



## 01

## 드론 소개

&lt; Top &gt;

ST-LINK Downloader

iA6B Receiver

Barometer

Main MCU  
STM32F405R

메인 코어

3 x Debug LED

디버그 상태 표시

6 DOF Sensor

자유도 6의 각속도 센서

+5V Power input  
+VBAT Battery ADCGPS Ublox M8N  
(5pin 2mm pitch)

GPS 연결

PWM 6CH

모터 제어

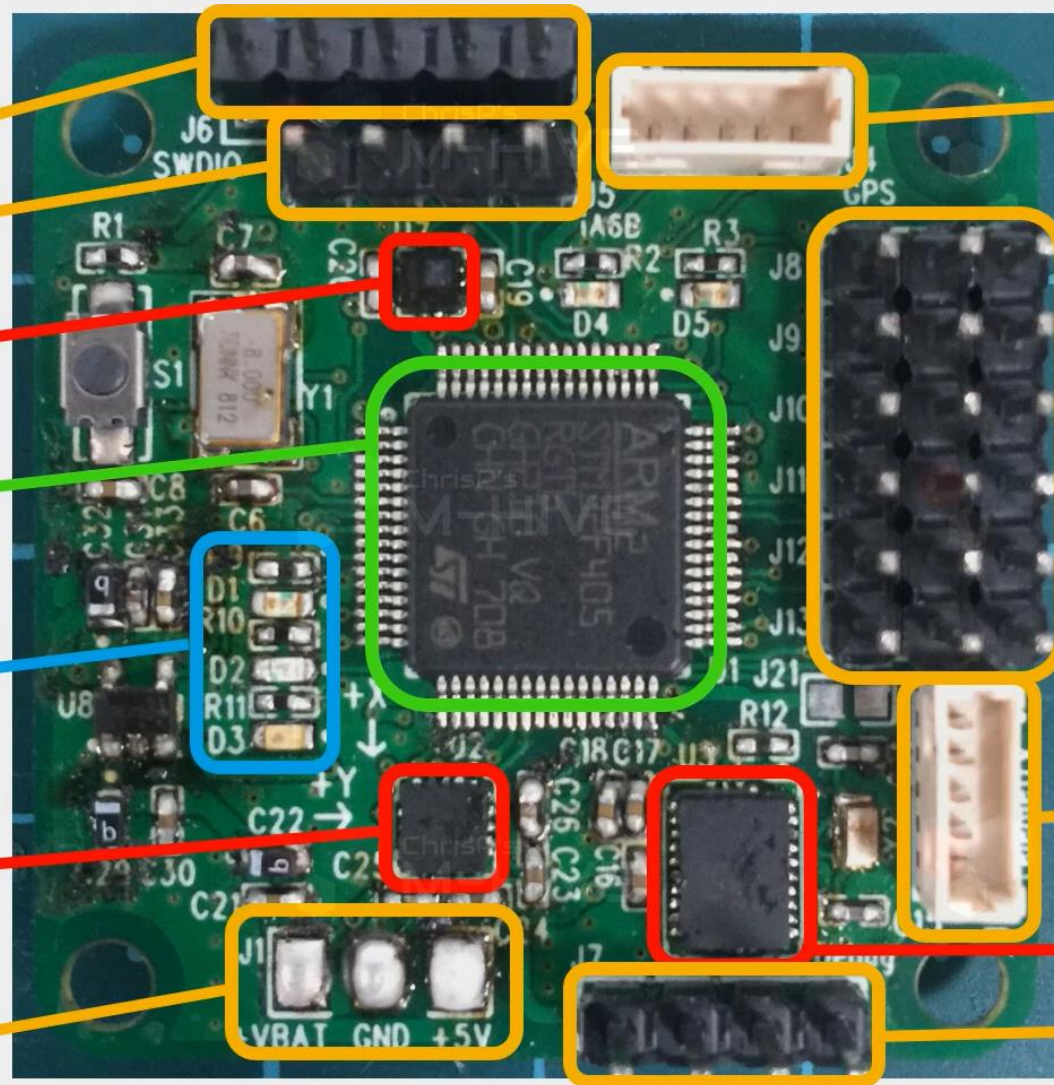
3DR Telemetry  
(5pin 2mm pitch)

9 DOF Sensor

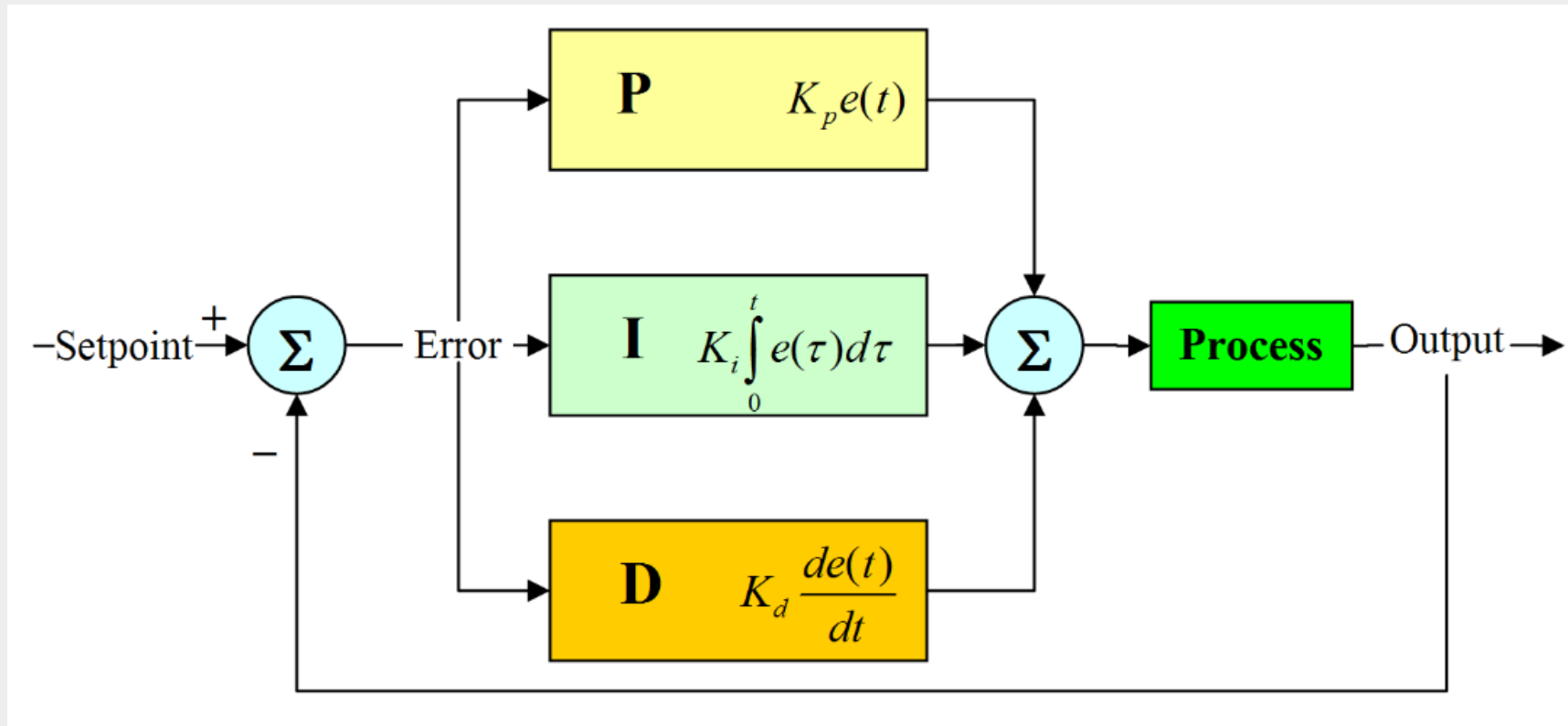
자유도 9의 각도 센서

Debug UART

터미널 통신



# 01 기초 지식

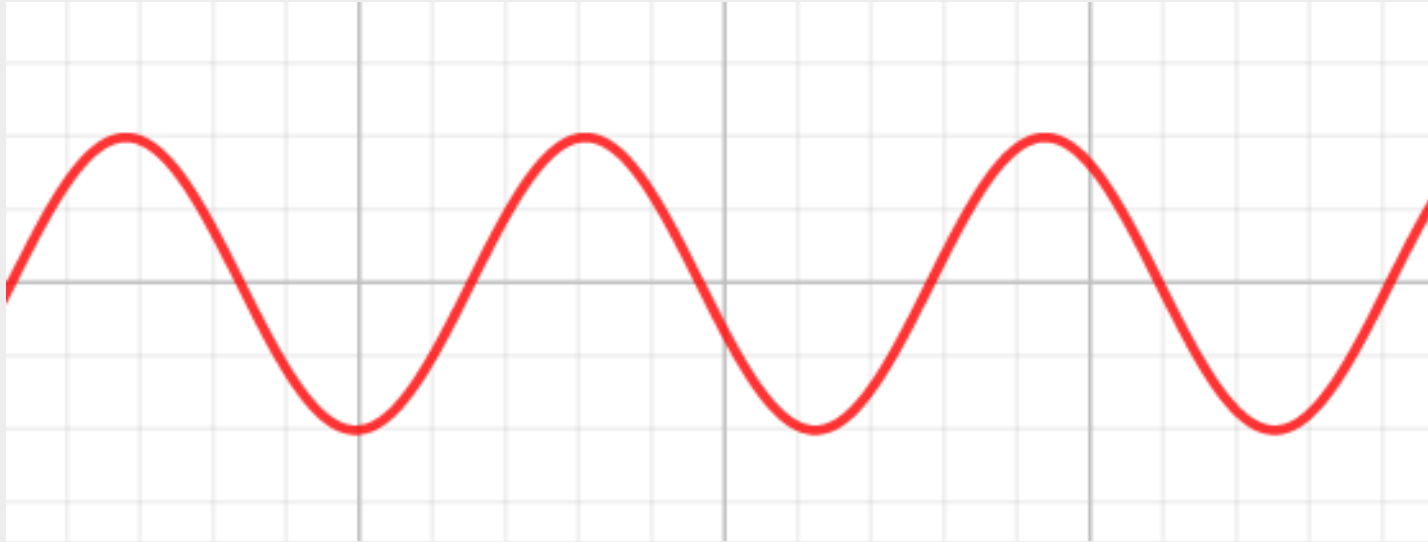


P - 비례항: 현재 상태에서의 오차 값의 크기에 비례한 제어작용을 한다.

I - 적분항: 정상상태 오차를 없애는 작용을 한다.

D - 미분항: 출력값의 급격한 변화에 제동을 걸어 오버슈트를 줄이고 안정성을 향상시킨다.

# 01 기초 지식



PID X

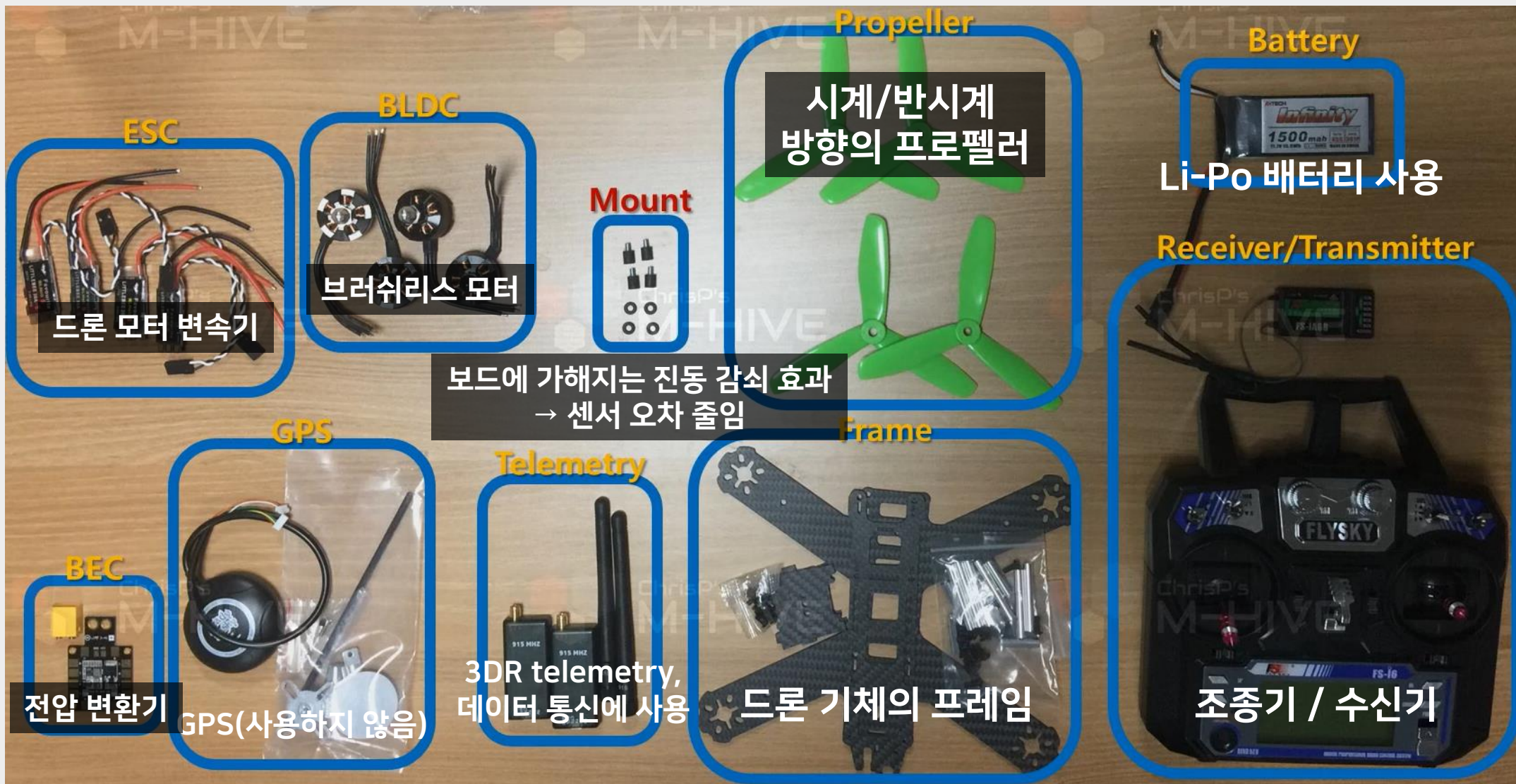


PID O



## 01

## 드론 소개



```

#define DT 0.001f
#define OUTER_DERIV_FILT_ENABLE 1
#define INNER_DERIV_FILT_ENABLE 1

void Double_Roll_Pitch_PID_Calculation(PIDDouble* axis, float set_point_angle, float angle/*BNO080 Rotation Angle*/, float rate/*ICM-20602 Angular Rate*/)
{
    /***** Double PID Outer Begin (Roll and Pitch Angular Position Control) *****/
    axis->out.reference = set_point_angle; //Set point of outer PID control
    axis->out.meas_value = angle;          //BNO080 rotation angle

    axis->out.error = axis->out.reference - axis->out.meas_value; //Define error of outer loop
    axis->out.p_result = axis->out.error * axis->out.kp;           //Calculate P result of outer loop

    axis->out.error_sum = axis->out.error_sum + axis->out.error * DT; //Define summation of outer loop
#define OUT_ERR_SUM_MAX 500
#define OUT_I_ERR_MIN -OUT_ERR_SUM_MAX
    if(axis->out.error_sum > OUT_ERR_SUM_MAX) axis->out.error_sum = OUT_ERR_SUM_MAX;
    else if(axis->out.error_sum < OUT_I_ERR_MIN) axis->out.error_sum = OUT_I_ERR_MIN;
    axis->out.i_result = axis->out.error_sum * axis->out.ki;         //Calculate I result of outer loop

    axis->out.error_deriv = -rate;                                //Define derivative of outer loop (rate = ICM-20602 Angular Rate)

    #if !OUTER_DERIV_FILT_ENABLE
    axis->out.d_result = axis->out.error_deriv * axis->out.kd;       //Calculate D result of outer loop
    #else
    axis->out.error_deriv_filt = axis->out.error_deriv_filt * 0.4f + axis->out.error_deriv * 0.6f; //filter for derivative
    axis->out.d_result = axis->out.error_deriv_filt * axis->out.kd; //Calculate D result of inner loop
    #endif

    axis->out.pid_result = axis->out.p_result + axis->out.i_result + axis->out.d_result; //Calculate PID result of outer loop
    /***** Double PID Inner Begin (Roll and Pitch Angular Rate Control) *****/
    axis->in.reference = axis->out.pid_result; //Set point of inner PID control is the PID result of outer loop (for double PID control)
    axis->in.meas_value = rate;              //ICM-20602 angular rate

    axis->in.error = axis->in.reference - axis->in.meas_value; //Define error of inner loop

```

```

while (1)
{
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
    if(tim7_1ms_flag==1){
        tim7_1ms_flag=0;
        //PID 제어가 1ms마다 반복
        ccr1 = 10500+500+(iBus.LV-1000)*10 - (iBus.RV-1500) * 5 + (iBus.RH - 1500) * 5 - (iBus.LH-1500) * 5;
        ccr2 = 10500+500+(iBus.LV-1000)*10 + (iBus.RV-1500) * 5 + (iBus.RH - 1500) * 5 + (iBus.LH-1500) * 5;
        ccr3 = 10500+500+(iBus.LV-1000)*10 + (iBus.RV-1500) * 5 - (iBus.RH - 1500) * 5 - (iBus.LH-1500) * 5;
        ccr4 = 10500+500+(iBus.LV-1000)*10 - (iBus.RV-1500) * 5 - (iBus.RH - 1500) * 5 + (iBus.LH-1500) * 5;
        //printf("%f %f\n", BNO080_Pitch, ICM20602.gyro_x);
        //printf("%f %f\n", BNO080_Pitch, ICM20602.gyro_y);
        //printf("%f %f\n", BNO080_Pitch, ICM20602.gyro_z);
    }

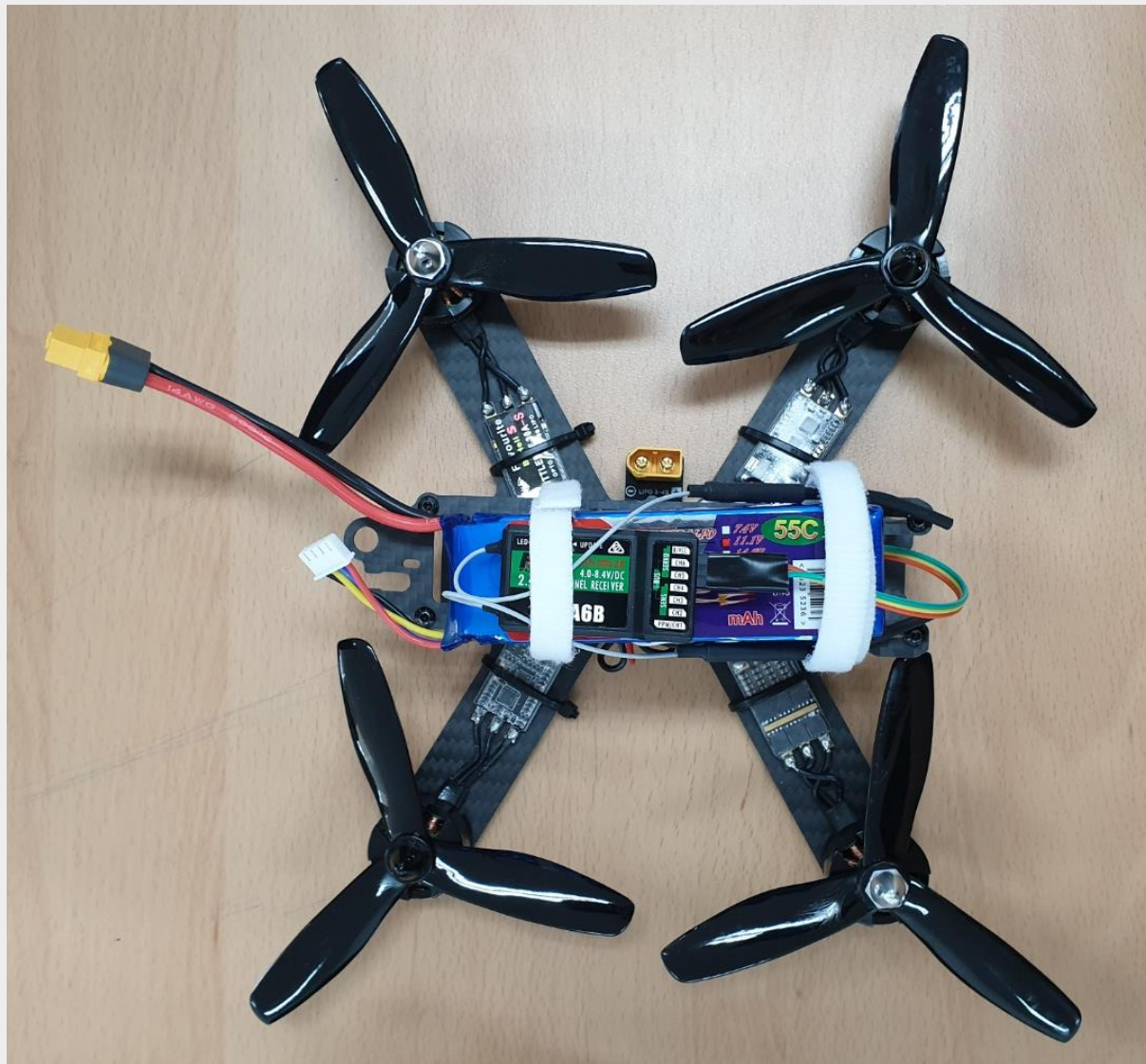
    if(iBus.SwA==2000&& iBus_SwA_Prev!=2000)
    {
        if(iBus.LV<1010)
        {
            motor_armining_flag=1;
        }
        else
        {
            while(Is_iBus_Throttle_Min()==0 || iBus.SwA==2000){
                LL_TIM_CC_EnableChannel(TIM3, LL_TIM_CHANNEL_CH4);

                TIM3->PSC = 1000;
                HAL_Delay(70);
                LL_TIM_CC_DisableChannel(TIM3, LL_TIM_CHANNEL_CH4);
                HAL_Delay(70);
            }
        }
    }
    iBus_SwA_Prev=iBus.SwA;
}

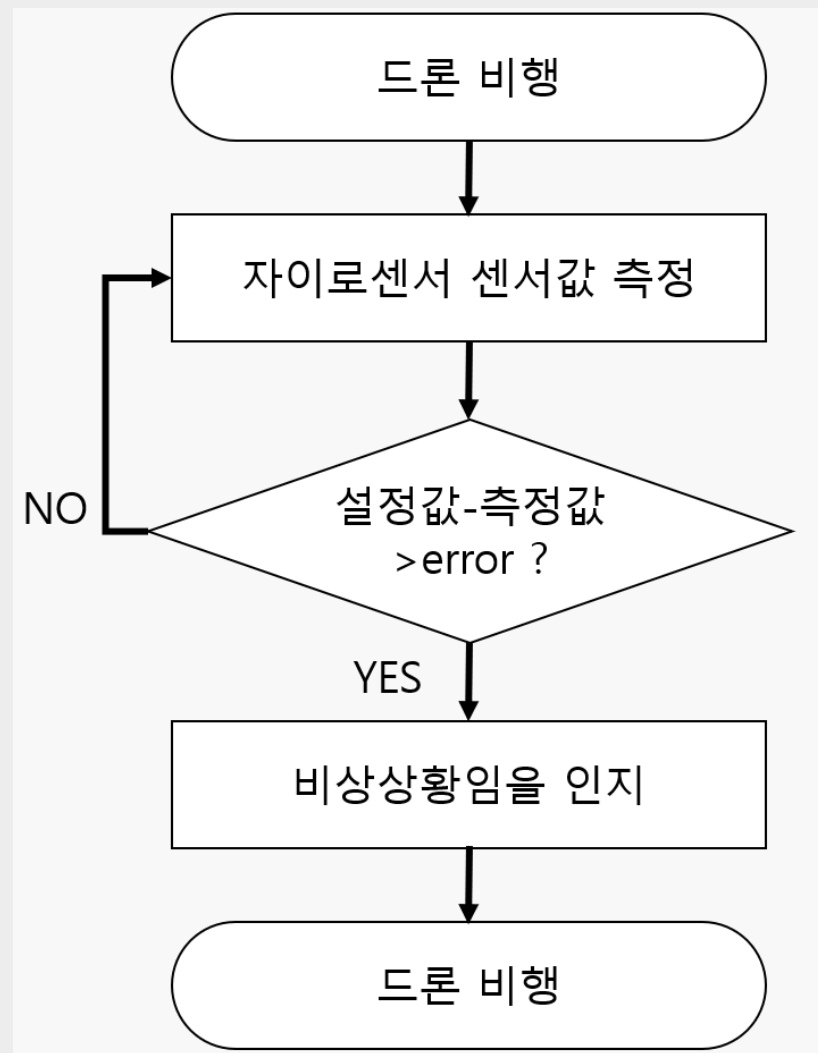
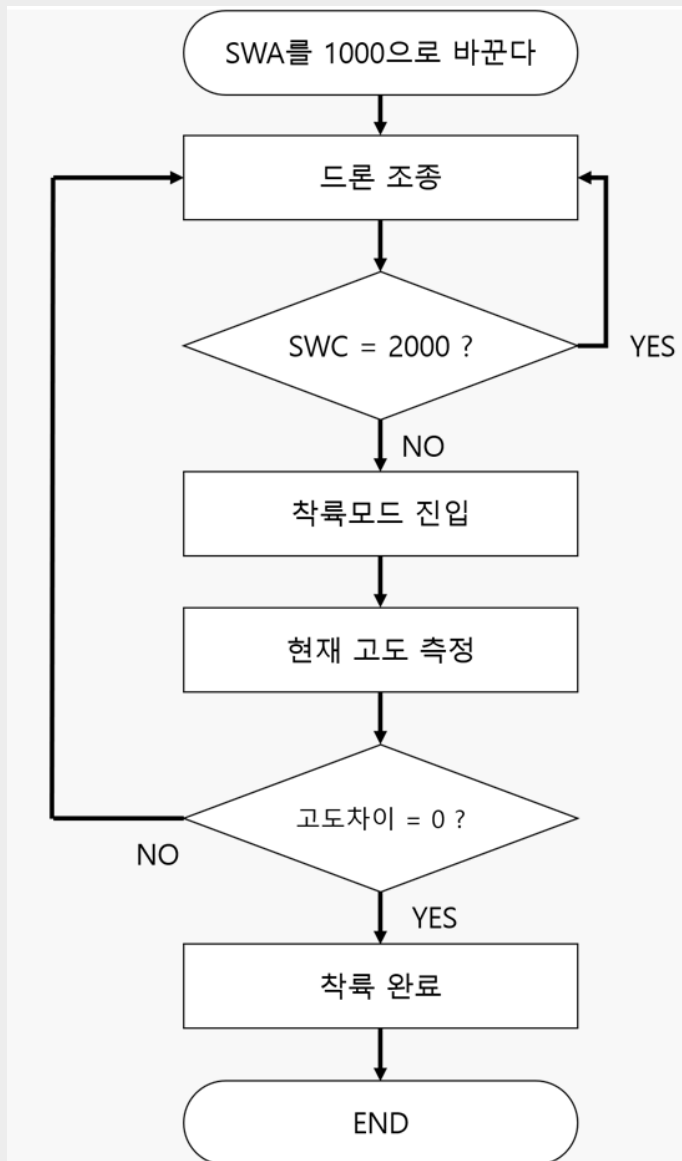
```

계산된 PID값을 이용하여 모터 제어









## 허주영

스스로 다른 조원들보다 특출나서 조장이 된 것이 아니라고 생각했기에, 1년간 진행될 프로젝트의 조장을 맡는 것은 상당히 큰 부담으로 다가왔다. 조원 모두가 드론에 대해 잘 아는 상황이 아니었기 때문에 진행 과정에서 발생한 크고작은 오류에 대처하는 것도 상당히 어려운 일이었다. 그럼에도 불구하고 그 문제를 해결했을 때의 성취감과, 어려운 상황에서 리더의 위치에서 함께 문제를 해결하고자 노력한 흔적은 앞으로 나에게 큰 도움이 될 것이라고 생각한다. 남은 2학기도 더 열심히 공부해서, 그 때는 지금보다 성공적인 마무리를 짓고 싶다.

## 김경은

학교 생활 중에 이러한 팀 프로젝트를 하는 것은 처음이었다. 대학 생활을 하면서 한 번 팀 프로젝트를 경험해보고 싶었다. 학과에서 파란학기라는 콘텐츠를 제안해 시작하게 되었는데 처음 시작할 때 다들 대학교 첫 팀플이다 보니 주제 선정, 계획 작성 등 많은 어려움을 겪었다. 하지만 조원들끼리 의견 내보고 합리적인 의사결정을 통해 극복할 수 있었다. 연구 도중에도 처음 연구 주제의 목표에 달성하지 못하고 하는 도중 드론을 변경하여 처음부터 다시 시작하게 되는 상황이 발생했었다. 하지만 조장을 중심으로 조원들이 잘 극복해나가 현재 까지 드론 연구가 잘 될수 있었다. 이번 프로젝트를 통해 여러 난관을 겪을 수 있었고 이를 통해 많은 것들을 배울 수 있었다.

## 박준혁

이번 파란학기 주제를 위기상황에서의 비상착륙 알고리즘으로 선정했을 때, 주변 선배들이나 교수님들이 드론 제작과 알고리즘 설계 2가지를 모두 이루는데 있어서 많은 어려움이 있을거라고 이야기하셨다. 지금 현재 드론 제작까지 진행이 된 상태인데 스스로 공부하고 공부한 내용을 바탕으로 드론을 제작하는 과정은 어려움이 많았다. 초반에 공부하였던 라즈베리 파이 파트에서는 실질적인 결과물은 완성하지 못하였지만 드론 제작에 대한 기본 지식을 학습할 수 있었던 좋은 활동이 되었다. 이러한 학습한 내용들을 기반으로 활동을 이어나가 프로젝트를 성공시키겠다.

## 이준혁

기초적인 내용을 하나도 모르는 상태에서 드론을 처음부터 제작한다는 것이 가장 어려웠다. 특히 오류가 발생했을 때 오류를 고치기 보단 이것저것 해보다 어찌다보니 해결이 되어버리는 경우가 많아 아쉬움이 많이 남는 활동이었다.

## 원유겸

파란학기를 하면서, 우리가 직접 팀을 꾸리고 주제를 잡아 새로운 도전을 한다는 것에 가슴을 뛰었습니다. 물론 시행 도중에 계획과 다른 일들에 어려움도 있었습니다. 하지만 문제를 극복하고 드론이 비행했을때, 우리가 이렇게 해낼 수 있다는 성취감을 배우는 기회였습니다.

- ✓ 2학기에는 비상착륙, 비상상황 인지 알고리즘이 실제로 동작하게 구현할 것
- ✓ 여러 번 실험을 통해 드론이 안정적으로 비행하고 착륙하는 능력 가지게 할 것



**감사합니다**