

# MULTIWII DRONE MANUAL



학년	3
학번	201601110
작성자	권준석

## 목차

부품 설명.....	4
품목명.....	4
드론 비행시간 계산.....	5
BLDC MOTOR 스펙.....	5
프레임 결합.....	6
프레임 구조도.....	6
1-1) 프레임 전원 분배보드 작업.....	6
1-2) 프레임 상판 조립 및 프레임 암에 모터결합.....	7
1-3) 아두이노 우노보드 결합.....	7
1-4)프레임에 우노보드 장착.....	8
1-5) ESC 와 모터.....	9
ESC 와 BEC.....	11
수신기 연결 및 바인딩.....	12
FS-i6 결선도.....	13
GY-86 연결.....	13
Multiwii 펌웨어 설치.....	14
Multiwii 지원 목록.....	14
MultiWii 아두이노 파일 실행.....	16
멀티위 설정.....	16
DEF.H.....	18
펌웨어 업로드.....	18
MultiWiiconf 실행.....	19
MultiWiiConf 초기 실행화면.....	20
MultiWiiConf 와 아두이노 연동.....	20
드론초기설정.....	21

<b>1.ESC Calibration.....</b>	<b>21</b>
멀티위 코드를 이용한 ESC 캘리브레이션 방법 .....	22
조종기를 이용한 ESC 캘리브레이션 .....	22
아두이노 코드를 이용한 ESC 캘리브레이션 .....	23
<b>2. 모터 회전 테스트 .....</b>	<b>24</b>
모터 방향 확인 .....	24
<b>3. Accel Calibration.....</b>	<b>25</b>
<b>4. MAG Calibration .....</b>	<b>25</b>
송신기 설정 .....	26
시동.....	26
시동켜기.....	26
시동 끄기.....	26
FS-i6 AUX 설정 .....	27
비행 .....	28
AUX 포트 설정 .....	28
기본 비행 모드 .....	28
테스트 비행.....	29
PID 설정 .....	30
PID 설정법 .....	30

## 부품 설명

### 품목명

품목명	개수
F450 프레임	1
T2212 920kv	4
MR-X3 ESC for Multicopters (30A/OPTO/COB/6S	4
PT-B2200N-SP35	1
아두이노 UNO	1
GY-86	1
FS-iA6 2.4G 6CH AFHDS 수신기	1
Flysky FS-i6 (송신기)	1
드론 1045R CW/CCW Prop set	4
브레드보드	1
U-BEC 레귤레이터 (입력 5V ~ 23V / 출력 5V 3A)	1

## 드론 비행시간 계산

무게를 입력하세요.

	무게(g)	수량(개)
페이로드 (전체)	150	
모터 (1개)	53	4
변속기 (1개)	23	4
배터리 (1개)	192	1
프레임 (전체)	282	
총 중량	928 g (0.9 kg)	
스로를 50%~70% 수준에서 출력이 232 g 에 근접한 출력이 나오는 모터를 선정하십시오.		

모터 스펙을 입력하세요.

	호버링	최대출력
1개당 출력 (g)	320	860
전체 출력 (g)	1,280	3,440
1개당 소모전류 (A)	3.04	13.4
전체 소모전류 (A)	12.16	53.6
여유 추력 (%)	63 %	
<p>여유 추력은 최대출력과 호버링출력의 차이입니다. 40%~60% 일 때 최적의 비행성능이 나오며, 30% 이하 또는 70% 이상일 경우 안전한 비행이 불가능합니다.</p> <p>※ 여유추력 30% 이하시 안전한 비행이 불가능합니다. ※ 16.08A 이상의 변속기를 사용하십시오.</p>		

배터리 용량별 비행가능 시간

전체 배터리 용량(mAh)	안전 최대시간(80%)	최대시간(분)
12,160	48	60
6,080	24	30
3,040	12	15
1,520	6	8
760	3	4
380	1	2
실제 용량 : 2200	8.7	10.9

호버링 출력(g)에 대한 소모전류를 계산합니다.

출력 (g)	전류 (A)
모터출력 (小) 290 g	소모전류 (小) 2.6 A
모터출력 (大) 460 g	소모전류 (大) 5.1 A
찾는 모터 출력 320 g	계산된 소모전류 A
출력이 g 일 때 A 의 전류를 소모하고 출력이 g 일 때 A 의 전류를 소모하므로, 출력이 g 일 때는 A 의 전류를 소모합니다. (※ 예상값)	

<https://www.falconshop.co.kr/?NaPm=ct%3Dk9b52k6n%7Cci%3Dcheckout%7Ctr%3Dds%7Ctrx%3D%7Chk%3D2b16049a6ad7f787e32ef2beb88592919a3d7940>

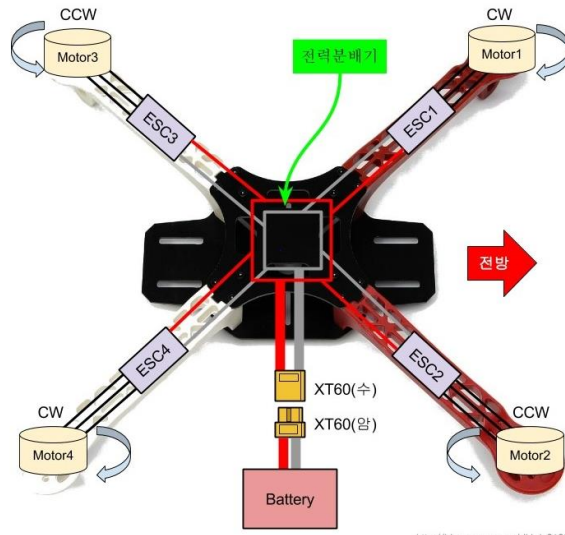
## BLDC MOTOR 스펙

ML2212 MOTOR								
Item NO.	Volts (V)	Prop	Throttle	Amps (A)	Watts (W)	Thrust (g)	Efficiency (g/W)	Operating temperature(℃)
ML2212 920KV	11.1V	DJI9.4*4.3	50%	1.8	20.0	230	11.5	37℃
			65%	2.8	31.1	310	10.0	
			75%	3.9	43.3	410	9.5	
			85%	5.5	61.1	480	7.9	
			100%	7.6	84.4	610	7.2	
		APC10*4.5	50%	2.6	28.9	290	10.0	55℃
			65%	5.1	56.6	460	8.1	
			75%	7.4	82.1	590	7.2	
			85%	10.1	112.1	730	6.5	
			100%	13.4	148.7	860	5.8	
	14.8V	DJI9.4*4.3	50%	2.7	40.0	350	8.8	52℃
			65%	4.4	65.1	490	7.5	
			75%	6.3	93.2	640	6.9	
			85%	8.3	122.8	790	6.4	
			100%	11.5	170.2	990	5.8	

Notes: The test condition of temperature is motor surface temperature in 100% throttle while the motor run 10 min.environment temperature 24℃

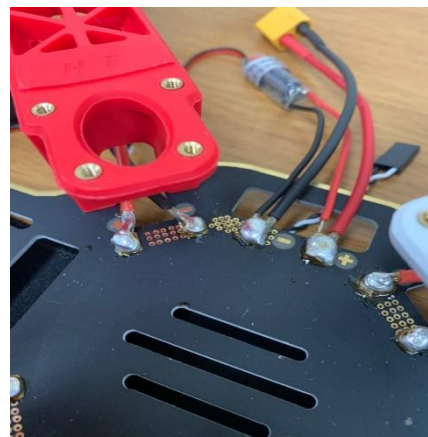
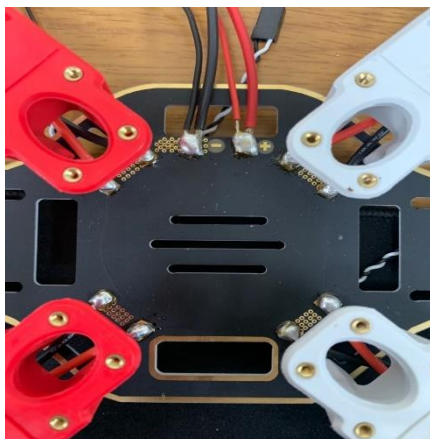
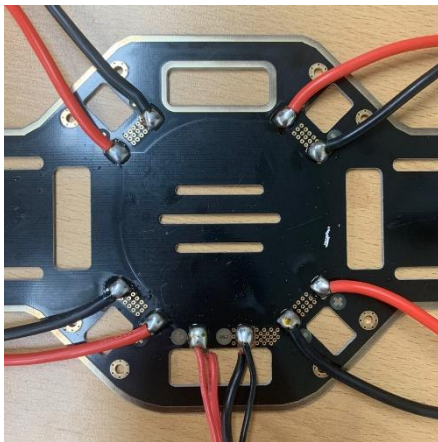
## 프레임 결합

### 프레임 구조도

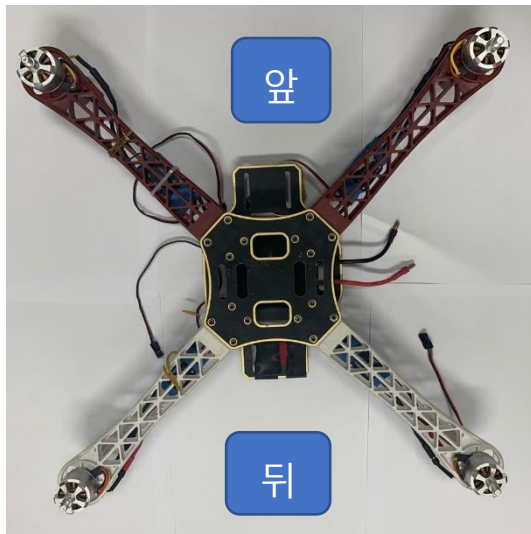


### 1-1) 프레임 전원 분배보드 작업

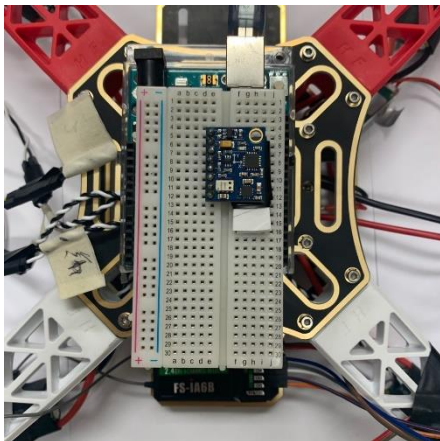
프레임 전원 분배판에 ESC를 극성에 맞춰 납땜해주세요.



## 1-2) 프레임 상판 조립 및 프레임 암에 모터결합

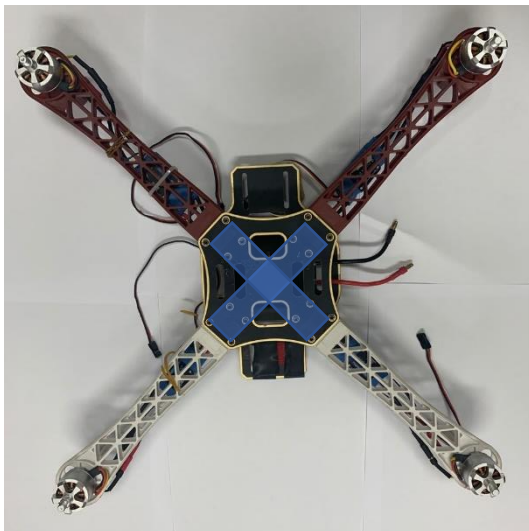
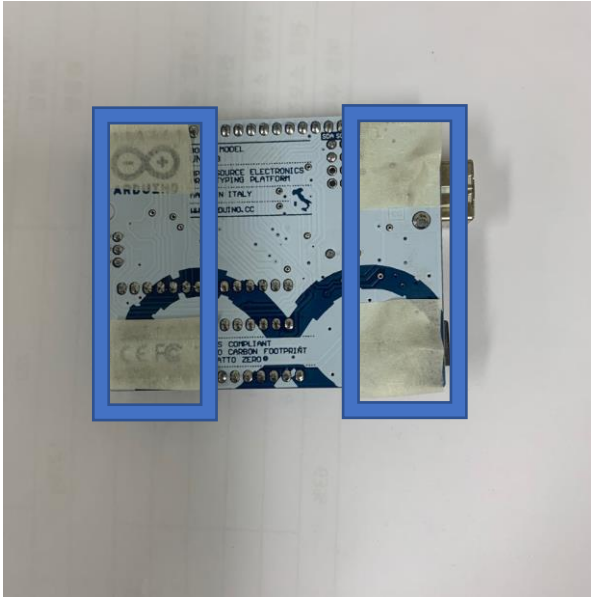


## 1-3) 아두이노 우노보드 결합



아두이노 우노보드위에 스펀지테이프를 붙여 브레드보드와 결합한다.

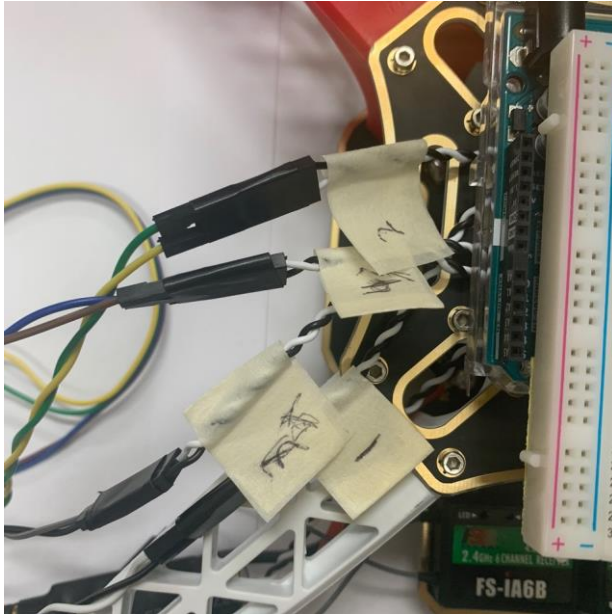
#### 1-4)프레임에 우노보드 장착



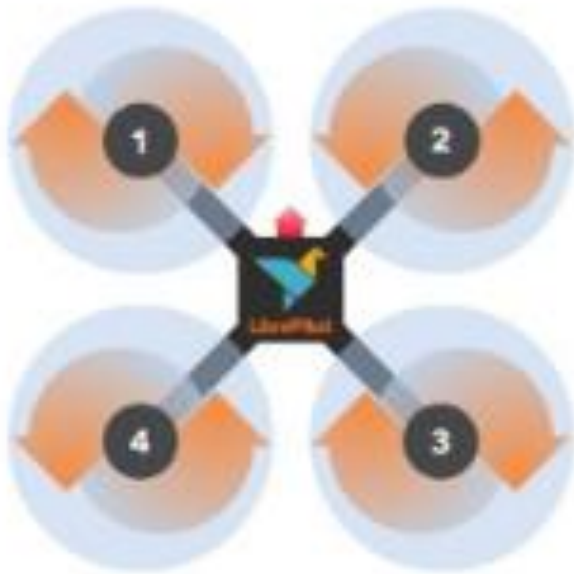
기체의 중심 부위와 자이로의 중심이 일치하여야 드론은 더욱 더 정밀하게 자세 제어가 가능합니다. 위치가 일정하지 않을 경우 기체가 자세 제어를 할 때 현위치를 이탈하여 자세를 제어할 수 있습니다. 장착하려는 판위에 x자 표시로 하여금 가능한 기체의 중심에서 벗어나지 않게 위치하도록 한다. 아두이노 밑판에 스펀지 테이프를 붙여 기체에 장착한다.



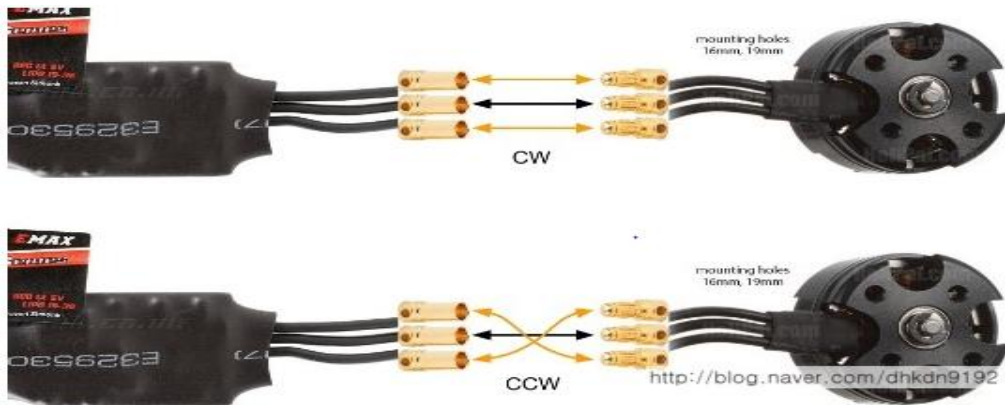
### 1-5) ESC와 모터



ESC선에는 몇번 모터와 연결되어있는지 표기해두는 것이 좋다.



쿼드 콥터의 기본적인 모터 번호는 위와 같다.



모터 회전 방향을 결정하는 것은 esc와 모터의 선 연결도에 따라 달라집니다.

모터나 모터와 연결되는 ESC에 바나나 커넥터가 없는 경우 커넥터를 납땜하고 수축튜브를 이용하여 절연해야 한다.

※커넥터 없이 ESC와 모터를 납땜을 이용하여 연결할 수 있지만 모터 방향이 잘못되어 있으면 납을 제거하고 다시 연결해야 하는 일이 발생할 수 있다.

4개의 변속기 신호선은 우노 보드의 DIGITAL핀 3,10,9,11번에 연결된다.

변속기	우노보드
1 번	3 번
2 번	10 번
3 번	9 번
4 번	11 번

## ESC와 BEC

### ABOUT ESC(Electric Speed Controller)& BEC(Battery Eliminator Circuit)

전자 변속기는 모터의 회전속도를 변경해주는 장치이다. ESC가 나오기 전까지는 기계적인 방식으로 모터의 회전속도를 변경해주는 수동식 변속기(Manual Speed Controller)가 있었다. 수동식 변속기를 사용하면 전압 차이로 인하여 동력용 배터리와 수신기용 배터리까지 필요하기 때문에 비효율적이었다. 2000년대부터는 BEC가 내장된 ESC가 나왔다. ESC에 BEC(Battery Eliminator Circuit)를 내장하여 동력용 배터리 하나만으로 수신기에도 전원을 공급할 수 있다.

ESC는 크게 두가지 종류가 있는데 BEC(Battery Eliminator Circuit)내장된 ESC와, OPTO ESC가 있다.

BEC는 Step Down Regulator와 같이 높은 전압을 낮은 전압으로 낮춰주는 장치이다. BEC는 Linear 방식과 Switching 방식 두가지로 분류할 수 있다.

#### 1) Linear 방식= 선형방식

전압을 낮추기 위해 열 손실 방식을 사용한다. 예로들면 12v에서 5v로 바꾸려고 하면 7v는 열로 바뀌 버리고 5v로 전압을 낮추는 방식이다. 선형방식의 단점은 발열이 많고, 효율이 낮다는 점이다. 그렇기 때문에 선형방식을 사용하려면 입력과 출력의 전압차이가 적은 것이 좋다.

#### 2)Switching 방식

작은 FET 트랜지스터를 이용하여 전압을 초당 150,000번 정도 빠르게 on-off하며 전압을 조정한다. 이렇게 나온 출력은 인덕터와 캐패시터를 통해 필터링 되어 리플을 부드럽게 하여 안정적인 DC전압을 제공하게 된다. 각 스위치 사이클 동안 트랜지스터가 켜져 있는 시간을 변화시킴으로써 장치는 부하없이 일정한 전압을 유지하기 위해 스스로 제어한다. 스위칭 방식은 선형방식과는 다르게 과도한 전압을 폐열로 버리지 않고 각 사이클에 충분한 전력을 공급받으면 스위칭방식의 BEC가 꺼지게된다. 스위칭 트랜지스터나, 컨트롤러 IC 및 회로의 인덕터에서는 소량의 열이 손실되지만, 이는 일반적으로 총 에너지 양의 4~6%에 불과하다. 이 때문에 스위칭 방식의 전체효율은 입력 전압에 관계없이 약 95%이다. 하지만 단점은 노이즈가 발생한다.

2000년대 말까지도 RC카, 고정익, 헬기는 27MHz, 40MHz, 72MHz의 아날로그 전파로 송수신이 되었다. 스위칭 방식 BEC에서 발생하는 노이즈가 아날로그 전파에 간섭을 일으키곤 하였다.

그 이후로 두가지의 ESC방식을 사용하게 되었다. Linear 방식의 BEC와 OPTO BEC를 내장시키는 것이다.

#### 1) Linear BEC 내장 ESC

위 글에 내용과같이 선형방식의 BEC는 발열이 발생하지만 노이즈가 발생하지는 않는다.

#### 2) OPTO방식 ESC

OPTO방식의 BEC는 ESC 제조사들이 ESC의 신뢰성을 높이기 위해 사용하여 OPTO 방식을 채택하였는데 OPTO는 Opto-isolator 또는 Opto-coupler의 약자이다. 수신기에서 ESC로 신호를 송신할 때 PWM방식으로 전달이 된다. ESC의 고전압 전력선과 상대적으로 낮은 수신기의 5v PWM신호선이 하나의 회로에 있음으로 신호에 왜곡이 발생할 수 있다. 왜곡된 신호는 노이즈가 발생할 수 있다. 이 문제를 해결하기 위해 OPTO방식을 이용하여 OPTO는 수신기에서 ESC로 보내지는 신호는 PWM 신호로 전송된다. 전기 신호는 A까지만 연결이 되어있고 A는 B에게 전달받은 신호를 광학방식으로 전달해준다. 그렇게 되면 아래의 '그림1'과 같이 A 와 B는 전기적으로 분리 되어있기 때문에 노이즈가 발생되지 않는다.

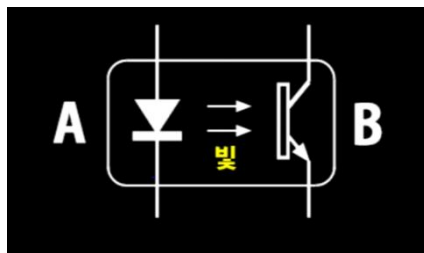
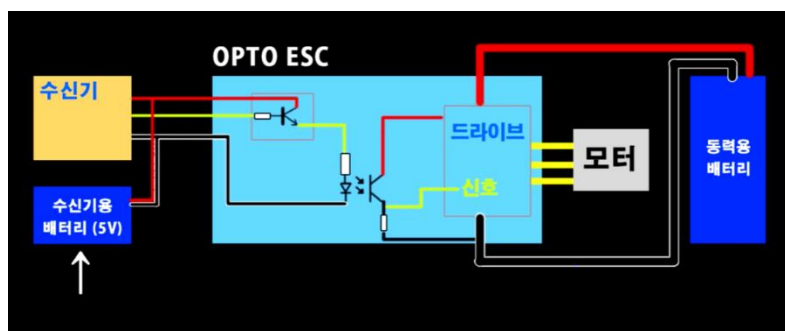


그림1



하지만 OPTO 방식의 단점은 동력용 배터리와 수신기와 연결이 되어있지 않기 때문에 수신기에 별도의 배터리를 장착해주어야 한다. 원리적으로 BEC를 내장할 수 없다. 그렇기 때문에 현대에 와서 OPTO ESC는 편의상 BEC가 없는 변속기를 편의상 OPTO 변속기라고 부르고 있다.

이렇게 OPTO- isolator 가없는 ESC는 다시 노이즈에 취약하게 된다. 하지만 현재 송수신기는 아날로그 방식이 아닌 2.4GHz의 디지털 송수신기를 사용하고 있기 때문에 ESC에서 발생하는 노이즈가 영향을 미치지 않는다고 한다.

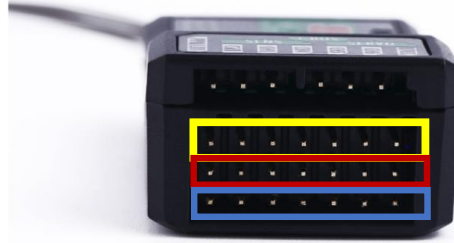
그렇기 때문에 이 글에서는 ESC는 변속기가 없는 OPTO-ESC(no BEC)를 채택하였다. BEC가 없기 때문에 U-BEC를 이용하여 보드에 전원을 공급한다.

## 수신기 연결 및 바인딩

바인딩이란 송신기와 수신기간의 주파수를 일치시켜 다른 수신기와 연결되지 않고 설정된 수신기로만 연결되도록 하는 것이다. AM/ FM 아날로그 주파수를 사용했을 당시에는 크리스탈을

변경하여 주파수를 바꾸어 중복되지 않게 하였다. 현재는 2.5GHz 디지털 주파수 대역을 사용하고 있고 조종기마다 바인딩 방법은 다르다. 이 글에서는 수신기 FS-ia6b를 사용하고 있습니다.

## FS-i6 결선도



  = 신호

  = 5V

  = gnd

수신기(신호)	아두이노 UNO
CH1(AILE)	Digital pin 4
CH2(ELEV)	Digital pin 5
CH3(THRO)	Digital pin 2
CH4(RUDD)	Digital pin 6
CH5(AUX1)	Digital pin 7
5V	5V
GND	GND

## GY-86 연결

GY-86	아두이노 UNO
GND	GND
VCC_IN	5V
SCL	A5
SDA	A4

## Multiwii 펌웨어 설치

펌웨어 다운로드 사이트: <https://code.google.com/archive/p/multiwii/>

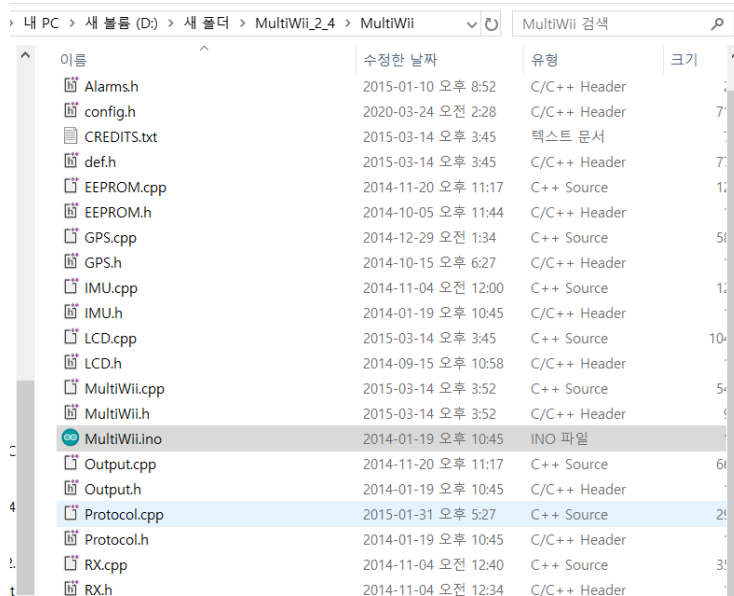
### Multiwii 지원 목록

- multi Rotor type
  - BICOPTER
  - TRICOPTER
  - QUAD +
  - QUAD X
  - HEX Y6
  - HEX FLAT +
  - HEX FLAT X
  - OCTO X8
  - OCTO FLAT +
  - OCTO FLAT X
- Gimbal
  - when associated with an accelerometer, MultiWii is able to drive 2 servos for PITCH and ROLL gimbal system adjustment
  - the gimbal can also be adjusted via 2 RC channels
- Camera trigger output
  - a servo output can be dedicated to trigger a camera button. A servo travel pattern is defined in this case
- GUI:
  - Coded with processing, java core: Linux/MAC/PC compatible, USB connection
  - exhaustive parameter configuration
  - graphical visualization of sensors, motors and RC signal
- Flight mode:
  - one of the following mode
    - angle velocity driven (accro mode)
    - absolute angle driven (level mode)
  - optional mode, compatible with the 2 previous one
    - altitude assisted mode (baro mode, compatible with the 2 previous mode)
    - head lock assisted mode (magneto mode)
- Hardware compatibility
  - receiver
    - any standard receiver with a minimum of 4 RC channels
    - any PPM sum receiver
  - servo
    - up to 4 any standard 50Hz servos can be used
  - motor ESCs
    - up to 8 standard ESC can be used, boosted with a 488Hz refresh rate
  - sensors
    - 3 MEMS Gyro
    - 2x IDG-650, 1x ISZ-650 (genuine Wii Motion Plus)
    - ITG3200
    - 3 MEMS Acc (optional)
    - LIS3L02AL (genuine Nunchuk)
    - BMA020
    - BMA180
    - 3 MEMS magnetometer (optional)

- HMC5883
  - HMC5843
  - 1 MEMS barometer (optional)
  - BMP085
- LCD for configuration of every parameters on the field
  - any Sparkfun serial 9600 baud LCD 2x16 characters
  - TEXTSTAR LCD 2x16 characters, with its 4 buttons supported

## MultiWii 아두이노 파일 실행

MultiWii2\_4/MultiWii/MultiWii.ino 로 들어가면 아두이노 파일이 실행된다.



## 멀티위 설정





멀티위 소스코드는 CONFIG.H 에서 대부분 수정 가능하다.

```
--
34  /***** The type of multicopter *****/
35  //define GIMBAL
36  //define BI
37  //define TRI
38  //define QUAD
39  #define QUADX
40  //define Y4
41  //define Y6
42  //define HEX6
43  //define HEX6X
44  //define HEX6H // New Model
45  //define OCTOX8
46  //define OCTOFLATP
47  //define OCTOFLATX
48  //define FLYING_WING
49  //define VTAIL4
50  //define AIRPLANE
51  //define SINGLECOPTER
52  //define DUALCOPTER
53  //define HELI_120_CCFM
54  //define HELI_90_DEG
55
56  /***** Motor minthrottle *****/
57  /* Set the minimum throttle command sent to the ESC (Electronic Speed Controller)
58   * this is the minimum value that allows motors to run at a idle speed */
59
```

사용할 기체의 형태에 맞게 주석(#define QUADX)을 해제한다. 현재 기체의 프레임은 QUAD X 이다.

## IMU 센서 모듈 선택

```
108 //define AEROQUADSHIELDv2
109 //define ATAVRSBIN1 // Atmel 9DOF (Contribution by EOSBandi). requires 3.3V power.
110 //define SIRIUS // Sirius Navigator IMU <- confirmed by Alex
111 //define SIRIUSGPS // Sirius Navigator IMU using external MAG on GPS board <- confirmed by Alex
112 //define SIRIUS600 // Sirius Navigator IMU using the WMP for the gyro
113 //define SIRIUS_AIR // Sirius Navigator IMU 6050 32U4 from MultiWiiCopter.com <- confirmed by Alex
114 //define SIRIUS_AIR_GPS // Sirius Navigator IMU 6050 32U4 from MultiWiiCopter.com with GPS/MAG remote located
115 //define SIRIUS_MEGAv5_OSD // Paris_Sirius™ ITG3050, BMA280, MS5611, HMC5883, uBlox http://www.MultiwiiCopter.com <-
116 //define MINIWI // Jussi's MiniWii Flight Controller <- confirmed by Alex
117 //define MICROWII // MicroWii 10DOF with ATmega32u4, MPU6050, HMC5883L, MS561101BA from http://flyduino.net/
118 //define CITRUSv2_1 // CITRUS from qcrd.ca
119 //define CHERRY6DOFv1_0
120 //define DROTEK_10DOF // Drotek 10DOF with ITG3200, BMA180, HMC5883, BMP085, w or w/o LLC
121 //define DROTEK_10DOF_MS // Drotek 10DOF with ITG3200, BMA180, HMC5883, MS5611, LLC
122 //define DROTEK_6DOFv2 // Drotek 6DOF v2
123 //define DROTEK_6DOF_MPU // Drotek 6DOF with MPU6050
124 //define DROTEK_10DOF_MPU//
125 //define MONGOOSE1_0 // mongoose 1.0 http://store.ckdevices.com/
126 //define CRIUS_LITE // Crius MultiWii Lite
127 //define CRIUS_SE // Crius MultiWii SE
128 //define CRIUS_SE_v2_0 // Crius MultiWii SE 2.0 with MPU6050, HMC5883 and BMP085
129 //define OPENLRSv2MULTI // OpenLRS v2 Multi Rc Receiver board including ITG3205 and ADXL345
130 //define BOARD_PROTO_1 // with MPU6050 + HMC5883L + MS baro
131 //define BOARD_PROTO_2 // with MPU6050 + slave MAG3110 + MS baro
132 //define GY_80 // Chinese 10 DOF with L3G4200D ADXL345 HMC5883L BMP085, LLC
133 //define GY_86 // Chinese 9 DOF with ITG3205 ADXL345 HMC5883L LLC
134 #define GY_86 // Chinese 10 DOF with MPU6050 HMC5883L MS5611, LLC
135 //define GY_90 // Chinese 10 DOF with MPU6050 HMC5883L BMP085, LLC
136 //define GY_521 // Chinese 6 DOF with MPU6050, LLC
137 //define INNOVWORKS_10DOF // with ITG3200, BMA180, HMC5883, BMP085 available here http://www.diywiiCopter.com
138 //define INNOVWORKS_6DOF // with ITG3200, BMA180 available here http://www.diywiiCopter.com
139 //define MultiWiiMega // MEGA + MPU6050+HMC5883L+MS5611 available here http://www.diywiiCopter.com
140 //define PROTO_DIY // 10DOF mega board
141 //define IOI_MINI_MULTIWII// www.bambucopter.com
142 //define Bobs_6DOF_V1 // BobsQuads 6DOF V1 with ITG3200 & BMA180
143 //define Bobs_9DOF_V1 // BobsQuads 9DOF V1 with ITG3200, BMA180 & HMC5883L
144 //define Bobs_10DOF_BMP_V1 // BobsQuads 10DOF V1 with ITG3200, BMA180, HMC5883L & BMP180 - BMP180 is software compat
145 //define FLYDUINO_MPU // MPU6050 Break Out onboard 3.3V reg
146 //define CRIUS_AIO_PRO
147 //define DESQUARED6DOFV2GO // DEsquared V2 with ITG3200 only
148 //define DESQUARED6DOFV4 // DEsquared V4 with MPU6050
149 //define LADYBIRD
150 //define MEGAWAP_V2_STD // available here: http://www.multircshop.com <- confirmed by Alex
151 //define MEGAWAP_V2_ADV
152 //define HK_MultiWii_SE_V2 // Hobbyking board with MPU6050 + HMC5883L + BMP085
153 //define HK_MultiWii_328P // Also labeled "Hobbyvbro" on the back. ITG3205 + BMA180 + BMP085 + HMC5883L + DSM2 Cor
```

GY-86 센서를 사용하기 때문에 134line 을 주석 해제한다.

GY-86 은 가속도,자이로, 전자 나침반, 고도센서가 내장되어있고 MPU6050(가속도 3 축, 자이로 3 축), HMC5883(전자 나침반, 지자계 3 축), MS5611(고도계, 압력센서) 통합센서 모듈이다.

## DEF.H

```
1295 #if defined(GY_86)
1296 #define MPU6050
1297 #define HMC5883
1298 #define MS561101BA
1299 #define ACC_ORIENTATION(X, Y, Z) {imu.accADC[ROLL] = -X; imu.accADC[PITCH] = -Y; imu.accADC[YAW] = Z;}
1300 #define GYRO_ORIENTATION(X, Y, Z) {imu.gyroADC[ROLL] = Y; imu.gyroADC[PITCH] = -X; imu.gyroADC[YAW] = -Z;}
1301 #define MAG_ORIENTATION(X, Y, Z) {imu.magADC[ROLL] = X; imu.magADC[PITCH] = Y; imu.magADC[YAW] = -Z;}
1302 #define MPU6050_I2C_AUX_MASTER // MAG connected to the AUX I2C bus of MPU6050
1303 #undef INTERNAL_I2C_PULLUPS
1304 #endif
// ...
```

CONFIG.H 에서 GY\_86 이 정의되면 DEF.H 에서 GY-86 에 대한 센서나 정보들이 정의된다. GY-86 은 자체적으로 풀업 상태이므로 I<sup>2</sup>C 의 풀업을 사용하지 않는다.

## 펌웨어 업로드

아두이노와 컴퓨터를 연결 시켜주고 아두이노 툴-> 보드:'Arduino Uno' ,  
포트:"COM7"(포트번호는 PC 의 환경에 따라 다르다.) 연결이 확인되면 업로드를 눌러 MultiWii  
업로드를 시켜준다.

## MultiWiiconf 실행

MultiWiiconf 는 JAVA 를 설치해야만 실행 가능하다.

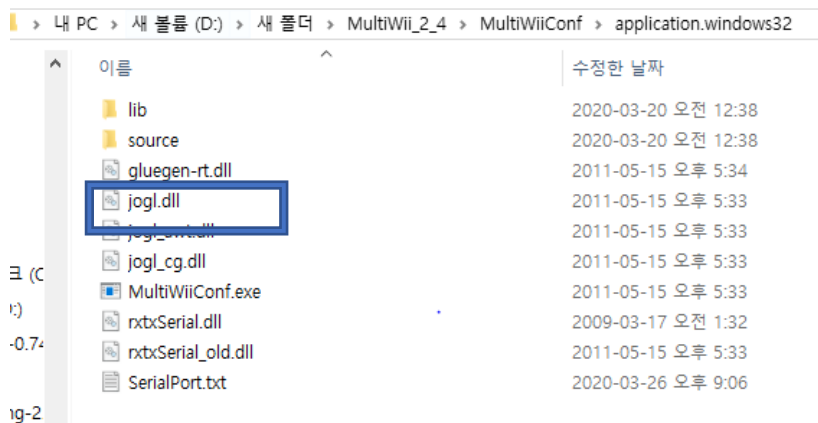
자바홈페이지: <https://java.com/ko/download/>

application.linux32	2020-03-20 오전 12:38	파일 폴더
application.linux64	2020-03-20 오전 12:38	파일 폴더
application.macosx	2020-03-20 오전 12:38	파일 폴더
application.windows32	2020-03-24 오전 2:40	파일 폴더
application.windows64	2020-03-20 오전 12:38	파일 폴더
defaults.mwi	2013-09-09 오후 11:42	MWI 파일
MultiWiiConf.pde	2014-12-31 오후 12:21	Processing Sc

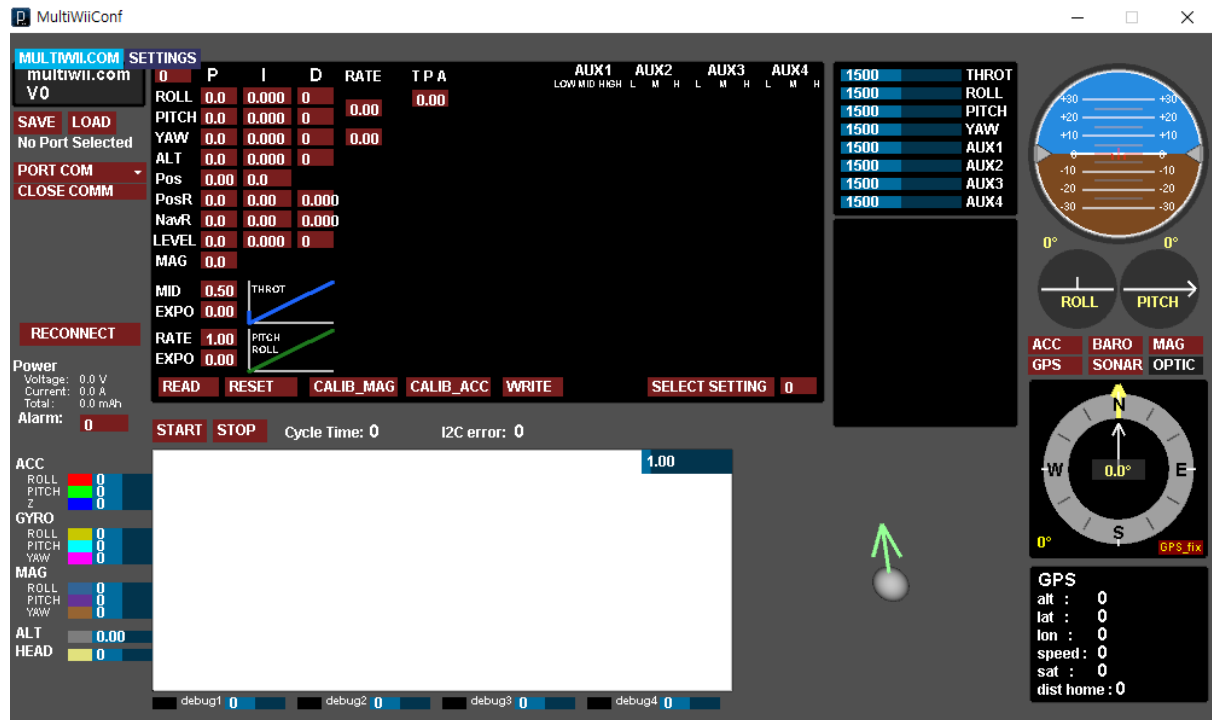
파일 위치:MultiWii2\_4/MultiWiiConf/

위의 위치로 들어가면 자신에게 맞는 os 를 선택할 수 있다.

윈도우는 64bit 가 아니라 32bit 로 들어가야지 GUI 창이 실행 가능하다.

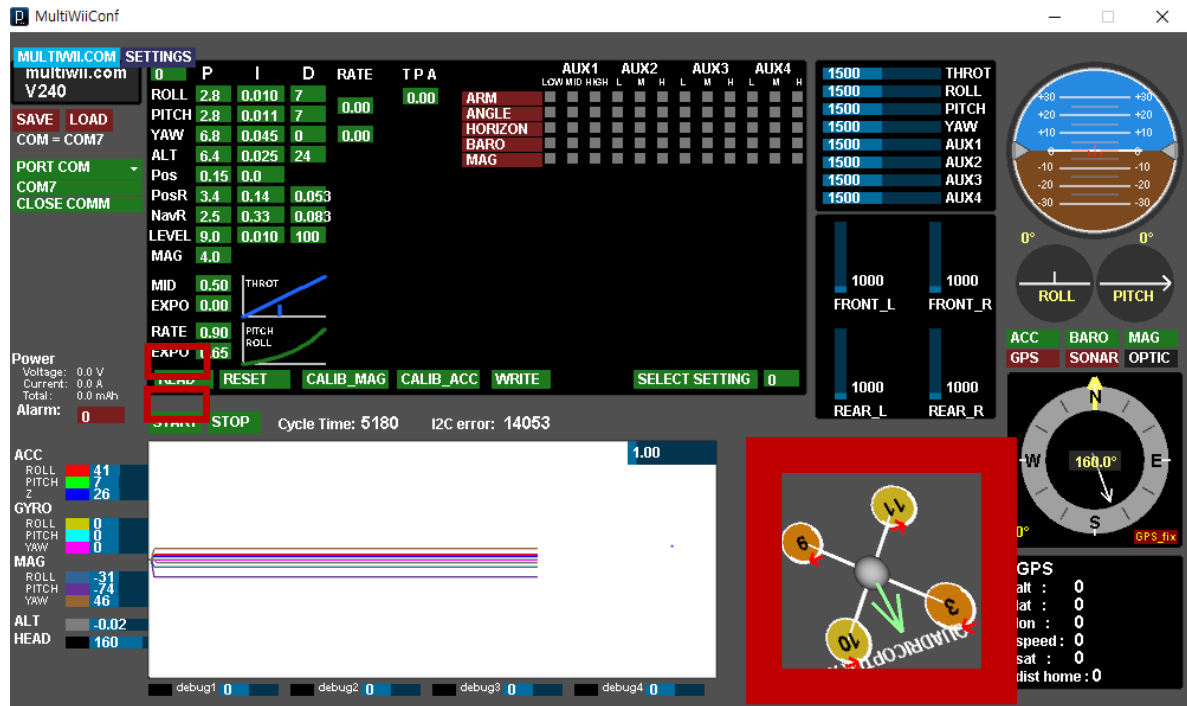


## MultiWiiConf 초기 실행화면



연결이 되어있지 않아 여러 공백이 나타난다.

## MultiWiiConf와 아두이노 연동



1. 통신포트 선택



아두이노 보드와 PC 가 연결된 상태에서 CONF 를 실행시켜야 통신포트 번호가 보인다.  
활성화된 포트 번호를 눌러 설정한다.

2. READ 버튼 클릭
3. START 버튼 클릭

위의 그림과같이 정상 작동이 되어진다. 비행 기체 이미지 표시는 지자계가 포함된 GY-80, GY-86 등의 센서를 사용하는 경우에는 정방향을 향하고 있지 않을 수 있다.

## 드론초기설정

### 1.ESC Calibration

ESC 캘리브레이션이란 ESC 가 응답해야 하는 스로틀 입력의 범위를 초기에 가르쳐주어야한다. 스로틀 채널의 PWM 값이 모터 꺼짐 명령과 일치하는지, PWM 값이 최대 스로틀과 일치하는지 알아야한다. 캘리브레이션 된 값들은 ESC 내부에 저장된다.

참조: <https://ardupilot.org/plane/docs/guide-esc-calibration.html>

ESC 캘리브레이션 방법에는 여러가지가 있다. 조종기로 인한 캘리브레이션, FC 펌웨어 내부 ESC 캘리 기능을 통한 캘리브레이션, 그리고 아두이노 소스코드를 통한 캘리브레이션이 있다. 추천하는 방법은 FC 펌웨어 내부 ESC 캘리 기능을 이용하여 캘리브레이션을 하는 것을 추천한다.

#### ※ESC 캘리시 주의사항

무조건 프로펠러를 먼저 제거해주고 캘리브레이션을 시작하여야한다. 모터가 최대 속력으로 회전하기 때문에 프로펠러가 장착되어있으면 큰 부상이 일어날 수도 있다.

## 멀티위 코드를 이용한 ESC 캘리브레이션 방법

1. 기체의 연결된 배터리를 제거합니다.
2. CONFIG.H 의 1146 line " #define ESC\_CALIB\_CANNOT\_FLY // uncomment to activate " 의 주석을 해제시켜준다.

```
1136  /******  
1137  /******   ESCs calibration   *****  
1138  /******  
1139  
1140  /* to calibrate all ESCs connected to MWii at the same time (useful to avoid unplugging/re-plugging each ESC)  
1141  Warning: this creates a special version of MultiWii Code  
1142  You cannot fly with this special version. It is only to be used for calibrating ESCs  
1143  Read How To at http://code.google.com/p/multiwii/wiki/ESCsCalibration */  
1144  #define ESC_CALIB_LOW  MINCOMMAND  
1145  #define ESC_CALIB_HIGH 2000  
1146  // #define ESC_CALIB_CANNOT_FLY // uncomment to activate  
1147
```

3. 저장을 한 후 우노보드에 업로드시켜주고 usb 연결을 해제한다.
4. 배터리를 연결시켜준다.
5. 연결시키고 몇초뒤에 최대 속도로 회전하고 멈추게 된다.
6. 20 초 정도 대기후에 배터리 연결을 해제시켜준다.
7. 우노와 pc 연결후 1146line 을 다시 주석처리 해주고 보드에 업로드 시켜준다.

## 조종기를 이용한 ESC 캘리브레이션

ESC 하나하나씩 캘리브레이션을 해준다.

1. 수신기에 전원을 입력해준다.
2. CH3 THRO 에 해당 ESC 신호선을 꽂아준다. GND 선도 접지를 해주어야한다.
3. 송신기에 전원을 키고 스로틀을 최대로 올려준다.
4. 배터리를 esc 에 인가시켜준다.
5. Esc 에 배터리 전원 인가가 되면 High 비프음이 들린다.
6. 송신기 스로틀 레버를 끝까지 내려주면 LOW 비프음이 들리면서 캘리브레이션이 된다.
7. 송신기의 스로틀 레버를 올리게되면 모터가 돌아가는 것으로 캘리가 되었는지 확인 할 수 있다.

## 아두이노 코드를 이용한 ESC 캘리브레이션

esc\_cali | 아두이노 1.8.12 (Windows Store 1.8.33.0)

파일 편집 스케치 들 도움말

```
esc_cali
1 #include <Servo.h>
2 #define MAX 2000
3 #define MIN 700
4 #define PIN 3 //핀설정
5
6 Servo motor;
7 void setup() {
8   Serial.begin(9600);
9   Serial.println("begin");
10  motor.attach(PIN);
11  Serial.println("calibrating~~")
12  Serial.println("모터가 돌아가고 2초뒤에 아무키나 입력")
13  motor.writeMicroseconds(MAX);
14
15  while(!Serial.available());
16  Serial.read();
17  Serial.println("min speed");
18  motor.writeMicroseconds(MIN);
19  Serial.println("calibrate done");
20 }
21
22 void loop() {
23 }
```

1. 소스코드를 보드에 업로드
2. 보드와 pc 가 연결된 상태에서 리포 배터리 연결
3. 시리얼창을 키고 모터가 최대속도로 회전하고 2~3 초 이후에 아무키 입력
4. 캘리브레이션 끝

## 2. 모터 회전 테스트

모터의 방향 및 구동이 되는 것을 확인하기 위해 모터 테스트를 실시 해준다.

☺ moter\_test | 아두이노 1.8.12 (Windows Store 1.8.33.0)

파일 편집 스케치 둘 도움말

```
moter_test$
1 #include<Servo.h>
2 Servo myservo;
3 int val;
4 void setup()
5 {
6   Serial.begin(9600);
7   myservo.attach(9);
8   delay(100);
9 }
10 void loop()
11 {
12   while(Serial.available()>0)
13   {
14     val=Serial.parseInt();
15     myservo.writeMicroseconds(val);
16     Serial.print(val);
17     delay(3000);
18   }
19 }
20 }
```

시리얼 창에 1000~ 2000 사이 값을 넣어준다. 모터 회전 방향을 알기 위해서는 속도가 낮아야 하므로 1050 을 시리얼 창에 입력하면 된다.

### 모터 방향 확인



종이테이프 등을 붙여 테스트 할 때 모터 방향을 확인해준다.



### 3. Accel Calibration

가속도센서 보정은 기체를 평평한 곳에 위치하여야한다. 경사진 곳이나 흔들림이 있는곳 이면 보정이 잘못되어 이륙을 못할 수 도 있다.

CALIB\_ACC 버튼을 눌러 보정을 시작한 후 15 초정도 대기한다. 피치, 롤, 수평계가 일치하게 된다. 캘리를 했는데도 수평계가 맞지 않는 다면 UNO 보드의 리셋버튼을 누르고 재부팅을 하고 다시 시도한다.

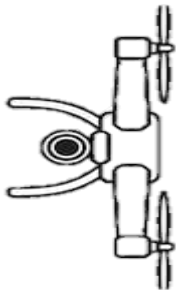
### 4. MAG Calibration

지자계 센서 보정을 하는것이다.

보정은 roll->pitch->yaw 순서로 각각 360 도씩 돌려주면 된다.



0도



90도



180도

지자계 보정을 할 때 평형 상태에서 시작을 하고 30 초 이내로 기체를 3 방향으로 돌려주면 된다.

## 송신기 설정

### 시동

```
234 | /***** ARM/DISARM *****/
235 | /* optionally disable stick combinations to arm/disarm the motors.
236 |  * In most cases one of the two options to arm/disarm via TX stick is sufficient */
237 | #define ALLOW_ARM_DISARM_VIA_TX_YAW|
238 | // #define ALLOW_ARM_DISARM_VIA_TX_ROLL
```

ARM: 스로틀 최소, 요 최대

DISARM: 스톱을 최소, 요 최소

### 시동켜기

모드 1



모드 2



### 시동 끄기



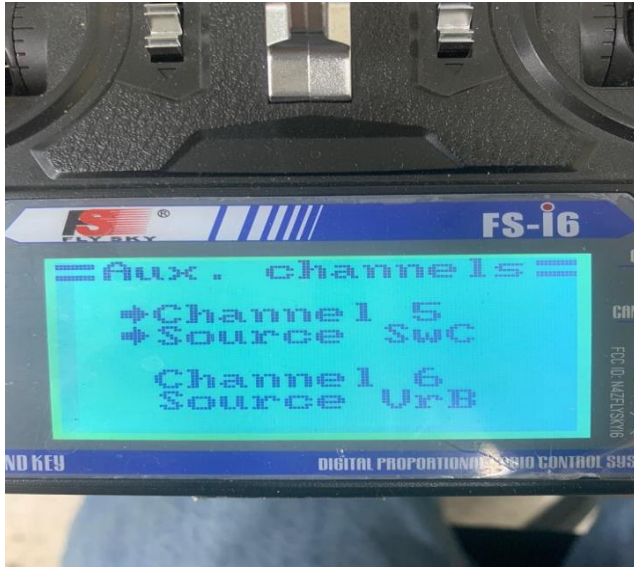
## FS-i6 AUX 설정



1. Ok 버튼을 꺾 눌러 메뉴로 이동하여 Functions setup으로 들어간다



2. Aux, channels로 진입한다.



3. 원하는 포트로 설정한다.
4. CANCEL을 꺾 눌러 저장한다.

## 비행

### AUX 포트 설정

AUX1,2 번의 입력되는 신호에 따라 비행모드가 변경된다.

	AUX1			AUX2			AUX3			AUX4		
	LOW	MID	HIGH	L	M	H	L	M	H	L	M	H
ARM												
ANGLE												
HORIZON												
BARO												
MAG												

### 기본 비행 모드

**ACRO MODE:** 기본 디폴트 모드로 자이로만을 이요하며 비행한다. 계속 보정하며 수평을 유지해야한다,

**ANGLE:** 수평을 유지하는 모드로 자이로와 가속도계를 사용한다.

**HORIZON:** ANGL 모드와 같게 수평을 유지하는 모드로 격한 움직임이있을 때 ACRO 모드와 유사하게 움직인다. 자이로와 가속도계를 사용한다.

**BARO:** 대기압센서가 부착된 경우 선택이 가능하다. 송시기에서 다른 입력이 올때 까지 현재 고도를 유지한다.

## 테스트 비행

기본적인 하드웨어와 소프트웨어 설정이 끝났다.

첫 비행은 지면 효과를 받는 높이 (약 1.2M)이상에서 비행을 실시한다.

Horizon 이나 ANGLE 모드로 비행하는 것을 추천한다.

송신기 트림은 중립이어야한다.

드론에 부착되어있는 여러 부품들은 단단하게 고정시키고 점퍼선등 탈부착이 쉬운 것들은 빠지지않게 고정시킨다. 점퍼선 같은 경우는 날개에 걸려 끊어져 비행중 위험한 상황을 유발할 수 있기 때문에 날개와 접촉하지 않도록 잘 고정시킨다

초기 비행이기 때문에 조종 미숙이나 드론 초기 설정 오류로 인해 사고가 발생 할 수 있다. 그렇기 때문에 드론을 고정한 상태에서 비행을 하는 것을 추천한다. 쿼드콥터의 4개의 지지대중 대각선 2개만 끈으로 묶어 바닥에 고정시킨다.

끈이 모터에 걸리지 않도록 조심히 비행을 할 것을 권한다.

## PID 설정

### P(proportional): 비례제어

비례제어란 현재 값과 목표값의 차이를 기준으로 제어한다.

P 가 너무 크면 복원력이 커지게 된다. 너무 낮으면 안정적으로 목표값에 도달하나 반응성이 느려지고 크게하면 반응성이 좋아지지만 조종성이 약해진다.

### I(Integral): 적분제어

적분제어란 지금까지의 오차를 누적하여 오차를 줄여 나가는 방식이다. 목표값과 현재값이 차이가 많이 나면 overshoot 이 발생된다.

### D(Derivative): 미분제어

기체 상태가 변할 때 빠르게 보정하는 역할을 함으로 D 값이 크면 조종성 및 반응도가 좋아진다.

## PID 설정법

$P=0$ ,  $I=0.010$ ,  $D=0$  으로 시작 설정한다. PI 제어가 가장 중요하기 때문에 안정된 PI 값을 찾아야한다. D 값은 기본 설정에 놔두어도 괜찮다.

$I=0.010$  에서 조금씩 증가시켜 원하는 값을 찾는다.

호버링 상태에서 기체가 일정 방향으로 흐르는 것은 I 값이 낮아서 오차를 빠르게 보정하지 못하는 경우 일수 도 있다. 조정해도 문제가 계속 발생하면 ACC cal 를 다시 하거나 송수신기 값을 초기화 시켜준다.

기체가 흔들리는 것은 I 값이나 P 값이 높아서 흔들리는 것으로 값을 줄인다.

P 값이 커지게 되면 자세유지에 대한 복원성이나 반응성이 높아지고 작아지게되면 반대로 복원성이나 반응성이 낮아진다.

스로틀 조작으로 호버링을 유지하기 힘들면 P 값을 낮춘다.

기체가 안정되지 못하고 흐르면 I 값을 키운다.

기체가 중심을 못잡고 여러 방향으로 흔들리면 I 값을 줄이거나 P 값을 증가 시켜야한다.