

```
// -----
// Z-WING2025 PIC12F1572 (C) 2025 inakakoubouKANAI
// -----
#include <xc.h>
#define XTAL_FREQ 16000000 //delay マクロ用 16MHz
//CONFIG
#pragma config FOSC = INTOSC // 内蔵クロック使用
#pragma config WDTE = OFF // Watchdog Timer を無効
#pragma config PWRTE = ON // Power-up Timer を有効
#pragma config MCLRE = OFF // MCLR は、デジタル入力 内部で VDD に接続
#pragma config CP = OFF // コードプロテクトは無効
#pragma config BOREN = ON // ブラウンアウトは、有効
unsigned int GYRSENTER; //GP5 ジャイロのニュートラル値
unsigned int GYR; //GP5 ジャイロからの入力信号数
unsigned int SRV; //GP4 受信機からのサーボ信号
unsigned int GS; //サーボ出力
unsigned int j; //ワーク
unsigned int GYR0;
unsigned int SRV0;
unsigned int GYRMAX;
unsigned int GYRMIN;
//-----合成してサーボに出力-----
void out_steering()
{
```

//ピッチ、ベクター制御用

```
// - , =
if (GYR > GYRMAX) ※PIC [ - ] , [ = ] のとき有効にし、それ以外は //を入れる。
{
    GYR = GYRMAX; //ジャイロで無駄に舵角を上げない
}
if (GYR < GYRMIN)
{
    GYR = GYRMIN; //ジャイロで無駄に舵角を下げない
}
```

```
// - , .
GS = SRV - GYR + GYRSENTER; ※PIC [ - ] , [ . ] のとき有効にし、それ以外は //を入れる。
```

```
// = , :
GS = SRV + GYR - GYRSENTER; ※PIC [ = ] , [ : ] のとき有効にし、それ以外は //を入れる。
```

//ヨー制御用

```
// . . , . : , : . , : :
GS = SRV; ※PIC [ . . ] , [ . : ] , [ : . ] , [ : : ] のとき有効にし、それ以外は //を入れる。
```

```
// . .
if (GYR <= GYRSENTER) {
    GS = SRV + (GYR - GYRSENTER) * 2;
} ※PIC [ . . ] のとき有効にし、それ以外は //を入れる。
```

```
// . :
if (GYR <= GYRSENTER) {
    GS = SRV + (GYRSENTER - GYR) * 2;
} ※PIC [ . : ] のとき有効にし、それ以外は //を入れる。
```

```
// : .
if (GYR >= GYRSENTER) {
    GS = SRV + (GYRSENTER - GYR) * 2;
} ※PIC [ : . ] のとき有効にし、それ以外は //を入れる。
```

```
// : :
if (GYR >= GYRSENTER) {
    GS = SRV + (GYR - GYRSENTER) * 2;
} ※PIC [ : : ] のとき有効にし、それ以外は //を入れる。
```

```
//-----
}
```

```

j=0;
GS=(unsigned int) (GS*1.05+0.5); //ニュートラル補正
RA0=1;
while(j<GS)
{
j++;
delay_us(4); //出力感度
}
RA0=0;
}

//ジャイロ初期値
void read_G()
{
GYR0=0;
while (RA4==1); //ジャイロ信号切れるまで待つ
while (RA4==0); //ジャイロ信号あるまで待つ
while (RA4==1)
{
GYR0++;
delay_us(6);
}
GYRSENTER=GYR0;
GYRMIN=GYR0*0.95; //ジャイロで効果がある最小舵角
GYRMAX=GYR0*1.05; //ジャイロで効果がある最大舵角
}

//ジャイロの信号を読む
void read_2()
{
GYR0=0;
while (RA4==1); //信号切れるまで待つ
while (RA4==0); //信号あるまで待つ
while (RA4==1)
{
GYR0++;
delay_us(6);
}
GYR=GYR0;
}

//サーボの信号を読む
void read_1()
{
SRV0=0;
while (RA5==1); //信号切れるまで待つ
while (RA5==0); //信号あるまで待つ
while (RA5==1)
{
SRV0++;
delay_us(6);
}
SRV=SRV0;
}

void mix_2to1()
{
delay_ms(10000); //正しいジャイロ信号が来るのを待つ
read_G();
read_G();
GYRSENTER=(unsigned int) ((GYRSENTER+GYR0)/2+0.5);
while(1)
{
//1 回目の出力
read_1();
out_steering();
//2 回目の出力
read_2();
out_steering();
}
}

```

```
}  
}
```

```
void main(void)
```

```
{
```

```
//レジスタの設定
```

```
OSCCON=0b01111000; //クロック周波数を 16MHz に設定
```

```
TMR0=0x00; //初期値 0
```

```
ANSELA = 0b00000000; //全てのピンをデジタルモードに設定
```

```
OPTION_REGbits.nWPUEN = 0; // 入力ピンのプルアップ無効化を解除
```

```
OPTION_REG = 0b00000010;
```

```
WPUA=0x30; //GP5,4 にプルアップ抵抗を設定
```

```
TRISA = 0b00110000; //右から RA0,RA1,RA2,X,RA4,RA5 4,5 を入力
```

```
INTCON=0x00; //割り込み不可
```

```
PORTA = 0b00000000; // 出力ピンの初期化(全て LOW にする)
```

```
mix_2to1();
```

```
while(1) {}
```

```
}
```