

# **MDSF40**

## **Low Voltage Ceiling Fan**

### **Application Note**

Sensorless Based High-Speed PMSM Solution

目录

目錄 ..... 錯誤! 尚未定義書籤。

1. 基本規格參數配置 ..... 3

    1.1 電機參數配置 ..... 錯誤! 尚未定義書籤。

    1.2 MOC 及硬體參數配置 ..... 錯誤! 尚未定義書籤。

    1.3 保護參數配置 ..... 錯誤! 尚未定義書籤。

2. 低壓吊扇參數調適流程 ..... 10

    2.1 啟動參數調適 ..... 錯誤! 尚未定義書籤。

    2.2 周邊控制功能 ..... 錯誤! 尚未定義書籤。

    2.3 順/逆風啟動參數調適 ..... 錯誤! 尚未定義書籤。

    2.4 保護參數調適 ..... 錯誤! 尚未定義書籤。

    2.5 其它參數配置 ..... 錯誤! 尚未定義書籤。

3. 程序架構 ..... 33

    3.1 程式流程 ..... 錯誤! 尚未定義書籤。

4. 參考電路設計 ..... 34

    4.1 電源輸入電路 ..... 錯誤! 尚未定義書籤。

    4.2 核心單元及周邊電路 ..... 錯誤! 尚未定義書籤。

    4.3 三相全橋變頻器 ..... 錯誤! 尚未定義書籤。

    4.4 兩相採樣電路 ..... 錯誤! 尚未定義書籤。

    4.5 母線電流採樣電路 ..... 錯誤! 尚未定義書籤。

    4.6 BEMF 回授電路 ..... 錯誤! 尚未定義書籤。

    4.7 外部端口 ..... 錯誤! 尚未定義書籤。

    4.8 其他周邊 ..... 錯誤! 尚未定義書籤。

## 1. 基本规格参数配置

### 1.1 电机参数配置

- 配置文件：MOTOR.h <Configuration Wizard>
- Set motor parameters

参数	数值范围/选项	说明	预设值
Motor Pole	0~30(2的倍数)	马达极数	14
Motor SMO_G	0~32767	$G = Ts / Ls/2$	15000
Motor SMO_F	0~32767	$F = 1 - (G * Rs/2)$	23000
Motor SMO_Kslf	0~32767	Low Pass Filter Gain	8000
Motor SMO_Gain	0~32767	$= Kslide1 / MaxSmcError$	32767
MaxSmcError	0~32767	SMO电流误差判断值	32767

调适方向建议：

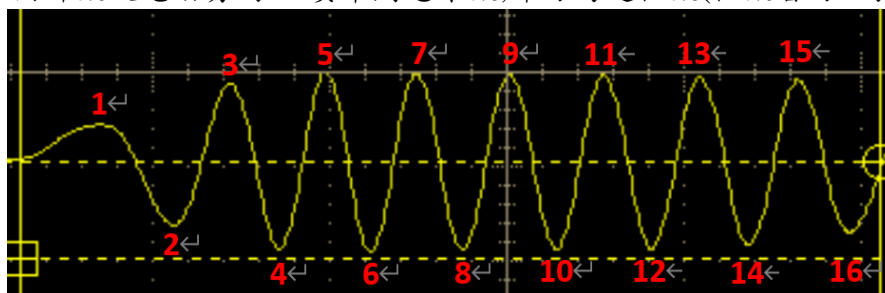
极数量测：

在无法确切得知马达的极数规格时,可以采用量测反电动势来取得极数,方法如下:

Step1. 示波器探棒随意勾取马达任一两相之间(ex: U对V or V对W...),马达无需接在控制板上

Step2. 利用手转或是外力让马达旋转一圈

Step3. 依照下图计数反电动势的正/负半周总个数,即为马达极数(极数皆为2的倍数)



SMO:

Sliding-Mode Current Observer主要用于估测电机转子位置，对参数变动干扰有一定程度的强健性，因此SMO\_G、SMO\_F参数无需透过R、L参数计算而得。当电机操作至额定功率期间，若出现电流异常发散导致停机保护，可以试误法方式调适dSMO参数 (**G必需小于F**)。

SMO\_Kslf为滑模电流观测器于估测转子位置时的滤波器系数，该参数与电机电气频率呈线性关系，也间接影响实际转子位置与估测转子位置间的相位关系。当电机操作至额定功率期间，若出现电流异常发散导致停机保护，可以试误法方式调适SMO\_Kslf参数。

$SMO\_Gain = Kslide1 / MaxSmcError$ ，Kslide1为SMO线性增益，可适当调整SMO\_Gain来间接改变Kslide1。

## 1.2 MOC及硬件参数配置

### ■ 配置文件：Pwm.h <Configuration Wizard>

#### ■ Set MPWMDATA

参数	数值范围/选项	说明	预设值	单位
Set PWM Frequency	12000~40000(10的倍数)	载波频率	15000	Hz

#### ■ Set MPWMDDB

参数	数值范围/选项	说明	预设值	单位
Set Deadband Time	1us/1.5us/2us/3us/4us/5us	设定死区时间	1us	--

调适方向建议：

PWM输出频率：

PWM频率设置越高，虽会增加MOSFET的切换损与电机发热，进而影响系统效率，但可有效抑制电机操作于高速应用下的电流/功率/速度涟波、振噪，提升整体系统性能表现；而频率设置低人耳会容易听到(低于16K)，但能提高马达最高转速，建议感量越低的马达设置的载波频率要越大。

死区设定：

根据实际量测上下臂输出波形判断所需要的死区范围，死区过大会影响马达的速度特性，可能会无法提升到额定最高速。

### ■ 配置文件：Motor.h <Configuration Wizard>

#### ■ Set Rshunt and OPA Gain

参数	数值范围/选项	说明	预设值	单位
Rshunt	--	设定采样相电阻阻值	100	0.1mR
OPA GAIN	1 Gain /2.5 Gain /5 Gain /10 Gain	内部OPA放大倍率	5 Gain	--

调适方向建议：

MDSFxx内部运算放大器增益有1、2.5、5、10四种放大增益值可供选择，其选配原则取决于电流采样解析度与电机额定相电流。

**ShuntR \* 马达最大电流( $I_{Peak}$ ) < 0.5V**

差动两端会限制在0.5V以下，是因为内部OPA两端最大限度电压为0.5V。

**OPA Gain \* ShuntR \* 马达最大电流( $I_{Peak}$ ) < 2.5V**

会限制2.5V是因为，内部A/D往上Offset了2.5V相电流有正/负，0~2.5V为负电流，2.5V~5V为正电流。

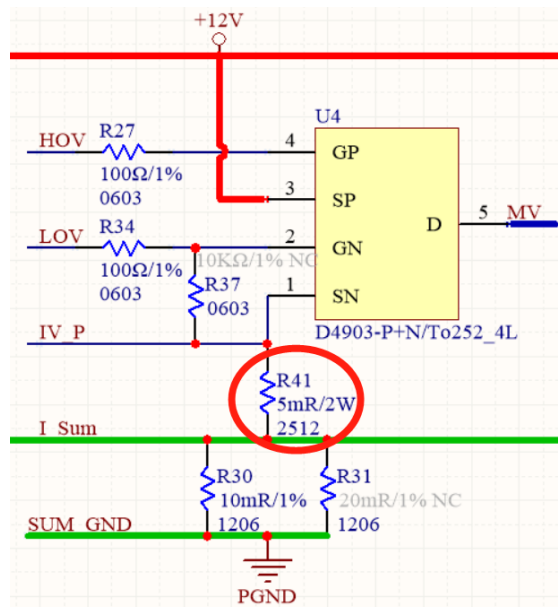
采样电阻选配建议：

一般电流采样 A/D 建议最大值对应马达  $I_{peak(max)}$  的两倍，例如：

DC24V 马达，在最高额定转速下相电流  $I_{peak} = 2.7A$ ，取3A，则两倍的采样最大电流为 6A

$R_{Shunt} = 2.5V / Gain / 6A$ ，Gain 先取 5，则  $R_{shunt} = 83mR$ ，取 0.1R 为用。

采样电阻阻值越大，则采样的精度也越高，但也需要考虑到采样电阻功耗导致的温升。在保证电阻温升符合要求的情况下，电阻值应尽量提升。



- 配置文件：Pwm.h <Configuration Wizard>
- 闸级驱动输出模式

参数	数值范围/选项	说明	预设值
Set MPWM SWAP	MDSF05/ MDSF40 MDSFA0	Gate Driver输出模式	<b>MDSF05/MDSF40</b>

调适方向建议：

当使用MDSF05(MCU：需外部配置Gate-Driver、MOSFET or IPM)与MDSF40 (MCM：内建P/N-type Gate-Driver，需外部配置MOSFET)时，需选定” **MDSF05/MDSF40**”，反之，当使用MDSFA0(MCM：内建N/N-type Gate-Driver，需外部配置MOSFET)时，则选定” **MDSFA0**”。( **Note:** 此选项千万不能选错,否则可能会造成MOSFET毁坏)

- 配置文件：Moc.h <Configuration Wizard>
- 空间向量脉宽调变(SVPWM)配置
- Set MOTOR\_CONT2

参数	数值范围/选项	说明	预设值
SVPWMMODE	7-SVPWM 5-SVPWM	SVPWM模式	<b>5-SVPWM</b>

调适方向建议：

(a). 当产品/系统以效率为优先指标时，可选定” **5-SVPWM**”，则SVPWM会以5段式切换输出，降低MOSFET切换损与发热，反之，当产品/系统以静音、震动及干扰为诉求与关键指标时，则选定---” **7-SVPWM**”。

### 1.3 保护参数配置

- 配置文件：Motor.h <Configuration Wizard>
- 保护/误动作状态辨识 (ErrorStatus)
- Error code(MotorErrorState)

+	Error code(MotorErrorState)
+	Motor state(MotorState)
<b>Error code(MotorErrorState)</b>	
Clear =	0(無錯誤)
OverVbus =	1(過電壓)
UnderVbus =	2(欠電壓)
OverTemperature =	4(過溫)
FaultLock =	8(堵轉)
AOCP =	16(過電流)
POCP =	32(相電流異常)

调适方向建议：

当系统发生马达异常停止或是无法启动时，可利用保护状态辨识，得知目前Error\_Code输出代号，以厘清系统的保护状态。可在main.c档内(第179行)，找寻if(UartFlag == 1)此判断式，在第195行底下的Debug\_(A~F)任一参数下使其等于” **MotorErrorState**” 此变数，范例如下：

```
if(UartFlag == 1){  
    SFR_PAGE = 0; Debug_A = eprom_data;  
    SFR_PAGE = 0; Debug_B = PI_OUT;  
    SFR_PAGE = 1; Debug_C = PI_OUT;  
    SFR_PAGE = 2; Debug_D = PI_OUT;  
    SFR_PAGE = 2; Debug_E = PI_UI;  
    SFR_PAGE = 3; Debug_F = MotorErrorState;  
  
    Uart_Package_Build();  
    UartFlag = 0;  
}
```

加入此變數

于UartSystem.h <Configuration Wizard>将UART功能Enable，并且打开” DataLogger.exe ” 软体即可监看此变数。

### 配置文件：Motor.h <Configuration Wizard>

- Set motor protection function
- Overvoltage/Undervoltage protection (OVP/UVLP) (✓)

参数	数值范围/选项	说明	预设值	单位
Set Vbus A/D Channel	CH0~CH7	采样母线电压A/D通道	CH6	--
Set Vbus rate parameter	0~65535	采样母线电压校正参数 = (Vbus_avg / Input voltage)*1000	27500	--
OVP Values	0~4000	过电压值	230	0.1V
OVP recovery Values	0~4000	过电压恢复值	200	0.1V
UVP recovery Values	0~4000	欠电压恢复值	75	0.1V
UVP Values	0~4000	欠电压值	60	0.1V
OVP/UVLP judgment times	0~3000	OVP/UVLP判断次数(10ms一次)	50	times

调适方向建议：

#### Vbus rate parameter：

先确认采样Vbus的AD channel，并填入V\_BUS\_CH内，之后把” Vbus\_avg” 此变数放入UART Debug监看(如同上面所提到的方式)，上电后将得到的Vbus\_avg除以目前的输入电压，再乘上1000，得出的数值即为Vbus rate parameter，范例如下：

假设在工作电压为DC12V下，所得到的Vbus\_avg为330， $330 / 12 = 27.5$   
 $27.5 * 1000 = 27500$ ，Vbus rate parameter = 27500。

#### OVP /UVLP setting：

将所希望的过电压保护值以及欠电压保护值，分别填入OVP Values、UVP Values参数，过电压恢复值以及欠电压恢复值，分别填入OVP recovery Values、UVP recovery Values。

OVP/UVLP judgment times参数为判断OVP/LVP的Debounce次数，每10ms判断一次。

- Locked-rotor protection (LRP) (✓)

参数	数值范围/选项	说明	预设值	单位
Motor speed abnormally high value	--	马达速度异常过高数值	400	rpm
Motor speed abnormally low value	--	马达速度异常过低数值	20	rpm
LRP judgment cycle	10~30000	马达堵转判断周期	2000	ms



调适方向建议：

#### LRP setting :

根据马达的最高/低额定转速，再累加/减一个量，来判断马达为堵转状态，高于**Motor speed abnormally high value**，认为过速保护；低于**Motor speed abnormally low value**，认为欠速保护

**LRP judgment cycle** 参数为判断LRP的周期，建议不要设定太小，避免误判。

#### ■ Over temperature protection (OTP) (✓)

参数	数值范围/选项	说明	预设值	单位
Set OTP A/D Channel	CH0~CH7	采样OTP A/D通道	CH3	--
OTP A/D Values	0~1023	过温A/D值	900	--
OTP recovery A/D Values	0~1023	过温恢复A/D值	800	--
OTP judgment times	0~3000	OTP判断次数(10ms一次)	5	times

调适方向建议：

#### OTP setting :

先确认OTP的AD channel，并填入**Set OTP A/D Channel**。OTP A/D Values为过温保护当下的A/D值，OTP recovery A/D Values为清除过温保护的A/D值，若可得知NTC的R/T表，可直接使用分压公式做推算。

**OTP judgment times**参数为判断OTP的Debounce次数，每10ms判断一次。

#### ■ Lack-Phase Protection(LPP) (✓)

- 保留；待验证可行性

#### ■ Phase current protection (✓)

保留；待验证必要性

- Phase Ia、Ib、Ic parameter (unit : mA) (700)
- PHASE\_OCP\_DURATION parameter (unit : ms) (50)

调适方向建议：

(a). 相电流异常保护，三相电流中,若大于所设定的**Phase Ia、Ib、Ic parameter**,认为电流异常保护,设定方向可将IaFb, IbFb放置UR观察,在马达额定转速运转下将其平均值纪录,约抓个1倍左右填入该参数.

(b). **PHASE\_OCP\_DURATION parameter** : 判断相电流延迟时间



- Set Protection to retry
  - AOCP\_Retry\_Enable (✓)
  - POCP\_Retry\_Enable (✓)
  - FaultLock\_Retry\_Enable (✓)
  - MotorLackPhase\_Run\_Retry\_ENABLE (✓)

参数	数值范围/选项	说明	预设值	单位
Set the number of retries	0~255	报错后重启次数	5	times
Set retry delay time	0~32767	重启延时时间	1000	ms

调适方向建议：

相对应的保护功能Retry Enable被勾起后，则当该保护发生后，会自动重启，反之则需重新断电。

**Set the number of retries**：为当错误产生时，马达重启的次数，一旦超过此值，便需要断电重启。

**Set retry delay time**：为发生错误后到重新启动中间的延时时间。

- 配置文件：Ocp.h <Configuration Wizard>

- 硬件过电流保护设置

- Set AOCPCONT

参数	数值范围/选项	说明	预设值
I_SHORT	0.15V/0.2V/0.25V/0.3V/0.35V/0.4V/0.45V/0.5V	过电流参考电压	0.15V
AOCPEN	Enable/Disable	类比过电流功能	Enable
DOCPEN	Enable/Disable	数位过电流功能	Disable

调适方向建议：

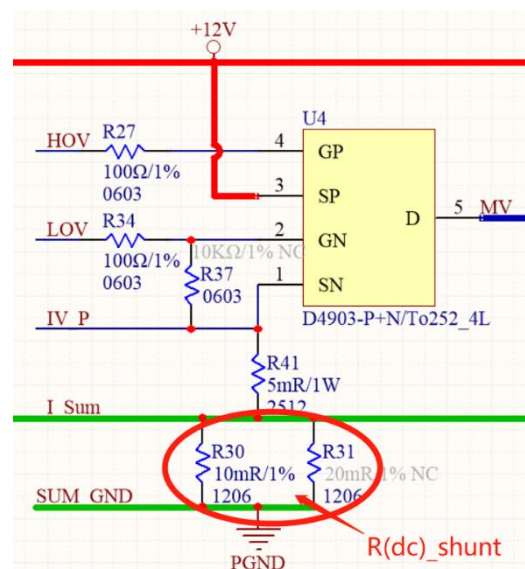
MDSF40提供8组等效过电流电压档位供配置，其母线电流保护点的计算为：

$$I\_SHORT = I_{ocp} * R(dc\_shunt)$$

AOCPEN为类比输入式OCP，DOCPEN为数位输入式OCP，若外挂Driver为IPM，一般皆设定DOCPEN，反之为AOCPEN。

#### OCP setting :

过电流值设定约在**马达最大电流(I\_Peak)之1.5倍~2倍左右**。



## 2. 低压吊扇参数调适流程

### 2.1 启动参数调适

#### Step1. 转子对位(Parking)

- 配置文件：Motor.h <Configuration Wizard>
- Set the motor tuning process
  - DCR Start Strategy (✓)

参数	数值范围/选项	说明	预设值	单位
Set DCR_V max values	0~100	DCR最终电压	15	%
Set DCR_V min values	0~100	DCR初始电压	5	%
Set DCR time	0~10000	DCR整体时间	1000	ms
Set DCR Zone1 percentage of time	0~100	0~180 占比	80	%
Set DCR Zone2 percentage of time	0~100	181~359 占比	20	%

- Square Wave Parking Start Strategy (✓)

参数	数值范围/选项	说明	预设值	单位
Set Startup_duty	0~32767	启动Duty量	256	--
Set Square Parking Time	0~32767	定位时间	500	ms

调适方向建议：

当马达启动，会有特定位置容易造成启动失败现象时，可开启DCR Start Strategy或Square Wave Parking Start Strategy二选一。在DCR策略中，根据负载扇叶不同的尺寸大小，合理调适DCR\_V值，一般而言越重的负载所需的DCR\_V值也越高，而DCR\_time的时间越长，定位摆动的时间也相对较长，这里不建议设定太长的时间，以免启动时间过久。其参数可依照试误法调适出适合马达的启动参数。

DCR Zone1 percentage of time 和 DCR Zone2 percentage of time两者参数加起来需为100，一般Zone1设定上会大于Zone2。

Square Wave Parking Start Strategy：此策略暂时不建议使用。Startup duty为设定duty量，此值建议一开始先不要设定太大，避免启动电流过大；Square Parking Time为设定整体定位时间，可依照定位情况作加减。

## Step2. 启动

- 配置文件：Motor.h <Configuration Wizard>
- Set the motor tuning process

参数	数值范围/选项	说明	预设值	单位
Set IQ parking duration	1-32767	IQ定位持续时间	1	ms

- Set FOC LOOP Parameter

- IQ

- ✓ Set IQ Current parameter

参数	数值范围/选项	说明	预设值	单位
Set IQ Initial current	--	IQ初始电流量	4000	mA
Set IQ Starting current	--	IQ启动电流量	3000	mA

调适方向建议：

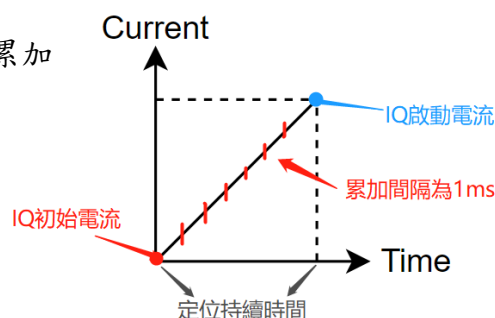
由于吊扇转子惯量大，在启动期间的启动电流/力矩需大一点，依照负载大小适度调整，如此可有效完成启动、降低震动及倒转机率。时间适度增加可改善启动失败率。

对位电流的加速斜率及时间说明如下：

如右图，定位持续时间为初始电流到启动电流的总时间，而累加间隔时间为Timer0时间(1ms)，故每1ms的累积电流量为：

$(\text{IQ启动电流} - \text{IQ初始电流}) / \text{定位持续时间}$

定位持续时间最小为1ms，若不需要定位可将初始电流=启动电流



## Step3. 位置开回路运转(Open loop)

- 配置文件：Motor.h <Configuration Wizard>
- Set the motor tuning process

参数	数值范围/选项	说明	预设值	单位
FOC_Control_Stage	Standby/ OpenLoop/ CloseLoop	FOC控制阶段	CloseLoop	--
Set SMO_RAMP acceleration slope	1~32767	PLL累积斜率	28	ms
Set PLL accumulation	1~100	PLL累加量	1	--
Set SMO_PLL initial speed	--	PLL初始转速	5	rpm
Set SMO_PLL end speed	--	PLL结束转速	35	rpm

调适方向建议：

**FOC Control Stage :**

Standby：此阶段不执行任何FOC功能，可用来初步校正IA及IB参数。

OpenLoop：开回路期间为电流闭环、位置开环。调适初期针对开回路调适时，建议可先将**FOC\_Control\_Stage**调为**OpenLoop**，如此一来马达便只会运转在开回路状态，当马达运转中若出现不可逆的现象时，也可利用**FOC\_Control\_Stage**加以分析是开回路出现的问题，还是闭回路的问题。

CloseLoop：可选择速度外环，或是功率外环。

**PLL :**

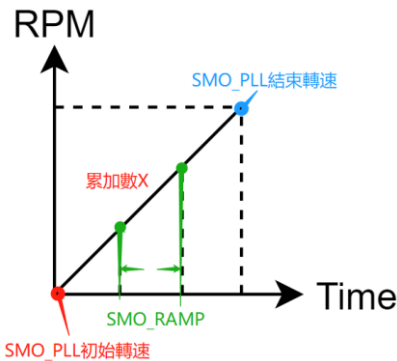
**SMO\_RAMP**减少可有效提高位置观测器收敛速度与减少开回路启动时间，但若太小有可能会启动太快而失败，需注意此变数值也包含顺/逆风下的PLL加速斜率。开回路加速度斜率及时间说明如下：

如右图，为SMO\_PLL初始转速到SMO\_PLL结束转速斜率，  
每SMO\_RAMP的时间去累加一个PLL量(为图中的累加数X)

故**PLL开环的总时间计算为：**

$$\frac{[(\text{SMO\_PLL结束转速} - \text{SMO\_PLL初始转速}) / (\text{PLL累加量} / \text{POLE PAIRS})] * \text{SMO\_RAMP}}$$

Ex :  $[(30-5)/(1/7)] * 15\text{ms} = 2625\text{ms}$



**Step4. 开回路衔接至闭回路**

- 配置文件：Motor.h **<Configuration Wizard>**
- Set the motor tuning process

参数	数值范围/选项	说明	预设值	单位
Set SMO_DELAY Delay time	1-32767	开环进闭环延迟时间	10	ms

SMO\_DELAY为内环准备要进闭环的延迟时间，可适度调整此参数。参数预留，暂不使用。

- Set FOC LOOP Parameter
  - IQ>
    - ✓ Set IQ Current parameter

参数	数值范围/选项	说明	预设值	单位
Set IQ End current	--	闭环电流初始值	2500	mA

调适方向建议：

在开环路结束后的电流即为IQ结束电流，此值在进入闭环后，会直接填入闭环的积分值以及输出值，当位置开回路(电流闭回路)切入闭回路时，由于初始外环PI控制器的积分器为零，为了避免衔接过程中转速异常涟波，需提供外环PI控制器的积分器一个初始值。

Step5. 闭回路运转(Close loop)

- 配置文件：Motor.h <Configuration Wizard>
- Set motor control program
  - Set the main control loop
    - ✓ Constant Phase Current control Enable/Disable(✓)
    - ✓ Constant Speed control Enable/Disable(✓)
    - ✓ Constant Power control Enable/Disable(✓)

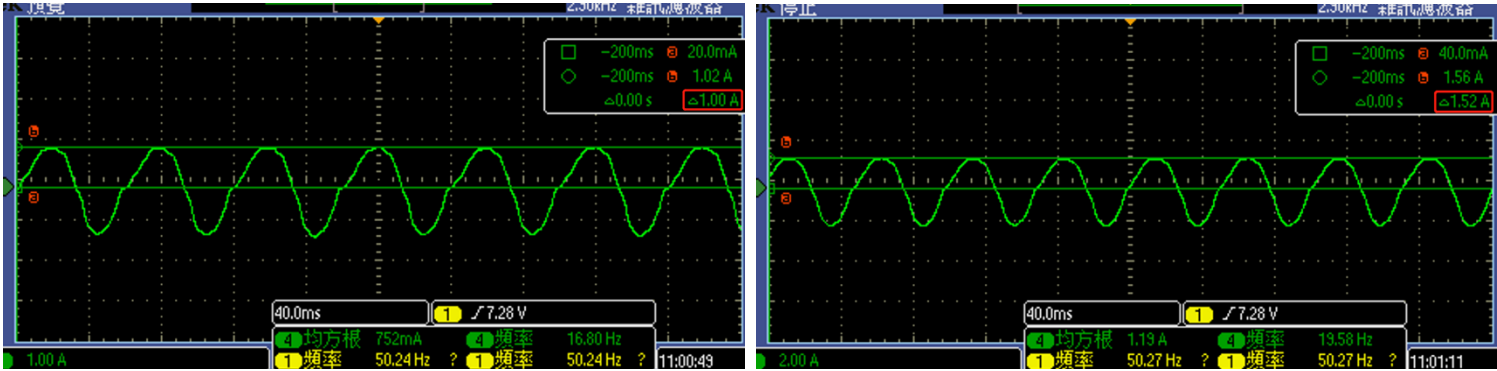
闭环控制共有三种控制模式，分别为相电流控制(Phase Current control)、速度控制(Speed control)、功率控制(Power control)，以下针对这三种模式分别做说明。

相电流控制(Phase Current control)

参数	数值范围/选项	说明	预设值	单位
Set the rated output phase current (max)	0~30000	额定输出相电流(max)	4000	mA
Set PHASE CURRENT_RAMP slope	1~32767	相电流控制周期	20	ms
Set PHASE CURRENT_CMD accelerate RAMP	0~32767	相电流累加量	5	--
Set PHASE CURRENT_CMD decelerate RAMP	0~32767	相电流递减量	10	--

调适方向建议：

根据马达的最大额定相电流填入设定值，这里所控制的电流为相电流峰值，例如下图所示：



Set the rated output phase current (max) = 1A      Set the rated output phase current (max) = 1.5A

电流控制斜率可依据 CURRENT\_RAMP、CURRENT\_CMD accelerate RAMP 以及 CURRENT\_CMD decelerate RAMP 做调整，方向为每 CURRENT\_RAMP 的时间，去累加/减 CURRENT\_CMD 的量。

累加减相电流量 和 实际累加/减相电流的换算公式为：

实际累加/减相电流 = (相电流累加/减量) / [(R\_SHUNT \* OPA\_GAIN) \* (1023/5) \* 64]



## 速度控制(Speed control)

参数	数值范围/选项	说明	预设值	单位
Set the rated output speed (max)	0~(32767/ Motor Pole/2)	额定速度(max)	380	rpm
Set SPEED_RAMP slope	1~32767	速度控制周期	20	ms
Set SPEED_CMD accelerate RAMP	0~32767	速度累加量	15	--
Set SPEED_CMD decelerate RAMP	0~32767	速度递减量	5	--

调适方向建议：

根据马达的最高额定转速填入设定值，建议可预留一些余量。 马达的转速最高命令控制可利用 Set the rated output speed (max) 来设定。

速度控制斜率可依据 SPEED\_RAMP、SPEED\_CMD accelerate RAMP 以及 SPEED\_CMD decelerate RAMP 做调整，方向为每 SPEED\_RAMP 的时间，去累加/减 SPEED\_CMD 的量，这里的加减量除上 Pole pairs 即为 rpm。

因内部角速度输出最大值为32767，所以再填入额定转速的最大值的上限为32767/极对数，假设马达的额定转速超过此上限的话，请至Pwm.h <Configuration Wizard>，将Set Base Angular Velocity此参数填入 = 额定最大转速 \* Pole Pairs即可。

## 功率控制(Power control)

参数	数值范围/选项	说明	预设值	单位
Set the rated output power (max)	0~10000	额定功率(max)	5100	0.01W
Set power magnification parameters	--	功率校正	1840	--
Set I_BUS A/D Channel	CH0-CH7	采样Ibus A/D通道	CH2	--

### ■ Set Ibus Limit

参数	数值范围/选项	说明	预设值	单位
Ibus limit values	0~100	限制母线电流值	35	0.1A
Ibus OPA GAIN	0~100	内部OPA增益	11	--
Ibus Correction factor	0~1000	母线电流校正	140	--

调适方向建议：

根据马达的额定功率填入设定值。 马达的功率命令控制可利用Set the rated output power (max) 来设定。

如果实际功率值和填入的功率输出有落差的话，可利用功率校正做微调，微调方向为：

- 如果目标功率比实际功率还低 >> 增加功率校正
- 如果目标功率比实际功率还高 >> 减少功率校正

母线电流限制：

会需要设定此项目是在宽电压条件下才需要额外做设定，假设工作电压为8v~20v，额定功率需要控制在51W，则在8v和20v条件下的51W电流落差很大(8v/6.375A；20v/2.55A)，前提是马达在低压的情况下还能输出这么高的电流的话，此时会另外增加限制母线电流的功能。

和功率校正一样，若限制母线电流和实际有落差，则微调母线电流校正，其校正手法可以先把限制母线电流调低，在确定已经触发限制电流的功能条件下，再去做电流校正微调，这里要注意一件事，无论是做功率微调还是电流微调，writer需使用隔离型的，或是烧录后暂时移除writer，接着再做校正观测。

## 2.2 周边控制功能

## 遥控器控制

- 配置文件： Motor.h <Configuration Wizard>
  - Set motor control program
    - ✓ Set Peripheral Control
      - Remote control commands Enable/Disable(✓)

参数	数值范围/选项	说明	预设值	单位
Remote control type	RF/IR	遥控器类别	IR	--

调适方向建议：

根据遥控器类别做选择，若无法辨别遥控器是RF或是IR的话，可观看遥控器头部发射端是否有颗类似LED的元件，如果有则为IR，反之为RF。(如右图)



- Power Level

参数	数值范围/选项	说明	预设值	单位
Set IRF Button Power1	0~6000	功率1	500	mW
Set IRF Button Power2	0~6000	功率2	1000	mW
...以此类推	0~6000	...以此类推	2000	mW

- Speed Level

参数	数值范围/选项	说明	预设值	单位
Set IRF Button SPEED1	0~ Set the rated output speed (max)	速度1	100	rpm
Set IRF Button SPEED2	0~ Set the rated output speed (max)	速度2	150	rpm
...以此类推	0~ Set the rated output speed (max)	...以此类推	250	rpm

- ✓ Observe Speed\_UI parameters Enable(✓)

参数	数值范围/选项	说明	预设值	单位
Set IRF Button SPEED_UI1	0~32767	速度积分1	500	--
Set IRF Button SPEED_UI2	0~32767	速度积分2	700	--
...以此类推	0~32767	...以此类推	900	--

- Current Level

参数	数值范围/选项	说明	预设值	单位
Set IRF Button Current1	0~10000	电流1	800	mA
Set IRF Button Current2	0~10000	电流2	1000	mA
...以此类推	0~10000	...以此类推	2000	mA



根据Set the main control loop外环控制模式设定Power Level 或 Speed Level 或 Current Level，每个Level皆对应遥控器按钮上的数字功能，若不需要那么多功能，则可在RF\_Decode.h/IR\_Decode.h将它关闭(后面遥控解码会另外说明)，而Level不够可自行增加，如下范例：

依照控制模式分别增加以下红框内代码(如下图):

#### ■ 配置文件：Motor.h <Configuration Wizard>

```

378 // <o> Set IRF Button Power7 (unit : mW)<0-6000>
379 #define IRF_Power_7 5100
380
381 // <o> Set IRF Button Power8 (unit : mW)<0-6000>
382 #define IRF_Power_8 5100
383 </h>

431 // <o> Set IRF Button SPEED_UI7 <0-32767>
432 #define IRF_SPEED_UI_7 1400
433
434 // <o> Set IRF Button SPEED_UI8 <0-32767>
435 #define IRF_SPEED_UI_8 1400
436 /h>

404 // <o> Set IRF Button SPEED7 (unit : rpm)
405 #define IRF_SPEED_7 (short)((float) 380 * 32767 / (BASE_RPM))
406
407 // <o> Set IRF Button SPEED8 (unit : rpm)
408 #define IRF_SPEED_8 (short)((float) 380 * 32767 / (BASE_RPM))

463 // <o> Set IRF Button Current7 (unit : mA)<0-10000>
464 #define IRF_Current_7 (float)4500/1000
465 #define IRF_Current_7_VALUE (int16)((float) IRF_Current_7 * I_AMPLIFIER) // unit : A * Gain
466
467 // <o> Set IRF Button Current8 (unit : mA)<0-10000>
468 #define IRF_Current_8 (float)4500/1000
469 #define IRF_Current_8_VALUE (int16)((float) IRF_Current_8 * I_AMPLIFIER) // unit : A * Gain
470 </h>

1309 enum Remote_Control_Speed_Level{
1310     R_1_SPEED = 1,
1311     R_2_SPEED = 2,
1312     R_3_SPEED = 3,
1313     R_4_SPEED = 4,
1314     R_5_SPEED = 5,
1315     R_6_SPEED = 6,
1316     R_7_SPEED = 7,
1317     R_8_SPEED = 8
1318 };

```

#### ■ 配置文件：Motor.c

```

178 #if (IRF_TRI == 1)
179     typedef struct{
180         int tableData[9]; ← 填入Level數+1
181         int tableDataUI[9];
182     }ConstIRF;
183
184 #if (POWER_CONTROL == 1)
185     code const ConstIRF rtConstPwr = {
186         {0, IRF_Power_1, IRF_Power_2, IRF_Power_3, IRF_Power_4, IRF_Power_5, IRF_Power_6, IRF_Power_7, IRF_Power_8}
187     };
188 #endif
189 #endif
190
191 #if (SPEED_CONTROL == 1)
192     code const ConstIRF rtConstSpd = {
193         {0, IRF_SPEED_1, IRF_SPEED_2, IRF_SPEED_3, IRF_SPEED_4, IRF_SPEED_5, IRF_SPEED_6, IRF_SPEED_7, IRF_SPEED_8},
194         {Spd_MinLimit+1, IRF_SPEED_UI_1, IRF_SPEED_UI_2, IRF_SPEED_UI_3, IRF_SPEED_UI_4, IRF_SPEED_UI_5, IRF_SPEED_UI_6, IRF_SPEED_UI_7, IRF_SPEED_UI_8}
195     };
196 #endif
197
198 #if (POWER_CONTROL == 1)
199     code const ConstIRF rtConstPwr = {
200         {0, IRF_Power_1, IRF_Power_2, IRF_Power_3, IRF_Power_4, IRF_Power_5, IRF_Power_6, IRF_Power_7, IRF_Power_8}
201     };
202 #endif

```

以下根据IR控制或是RF控制分别增加红框内代码:

■ 配置文件 RF\_Decode.c

```
141  #if(e_RF_SPEED7 == Enable)
142      case RF_DATA_SPEED7: IRF_Channel = R_SPEED; IRF_SPEED_Level = R_7_SPEED; break;
143  #endif
144
145  #if(e_RF_SPEED8 == Enable)
146      case RF_DATA_SPEED8: IRF_Channel = R_SPEED; IRF_SPEED_Level = R_8_SPEED; break;
147  #endif
```

■ 配置文件：RF\_Decode.h <Text Editor>

```
253  //<e> Enable Speed7 Button
254      #define e_RF_SPEED7 0
255  #if(e_RF_SPEED7 == 1)
256      //<o> Button speed 7<0-65535>
257      #define RF_DATA_SPEED7 65533
258  #endif
259  //</e>
260
261  //<e> Enable Speed8 Button
262      #define e_RF_SPEED8 0
263  #if(e_RF_SPEED8 == 1)
264      //<o> Button speed 8<0-65535>
265      #define RF_DATA_SPEED8 65533
266  #endif
267  //</e>
```

■ 配置文件 IR\_Decode.c

```
86  #if(e_IR_SPEED7 == 1)
87      case IR_DATA_SPEED7: IRF_Channel = R_SPEED; IRF_SPEED_Level = R_7_SPEED; break;
88  #endif
89
90  #if(e_IR_SPEED8 == 1)
91      case IR_DATA_SPEED8: IRF_Channel = R_SPEED; IRF_SPEED_Level = R_8_SPEED; break;
92  #endif
```

■ 配置文件：IR\_Decode.h <Text Editor>

```
217  //<e> Enable Speed7 Button
218      #define e_IR_SPEED7 0
219  #if(e_IR_SPEED7 == 1)
220      //<o> Button speed 7<0-65535>
221      #define IR_DATA_SPEED7 51255
222  #endif
223  //</e>
224
225  //<e> Enable Speed8 Button
226      #define e_IR_SPEED8 1
227  #if(e_IR_SPEED8 == 1)
228      //<o> Button speed 8<0-65535>
229      #define IR_DATA_SPEED8 51255
230  #endif
231  //</e>
```

## RF解码

### ■ 配置文件：RF\_Decode.h <Configuration Wizard>

#### ● Customize

#### ✓ Set RF\_DEC\_CTRL

参数	数值范围/选项	说明	预设值	单位
RFIN_SEL	P0.5(IR1) / P3.3(IR2)	选择接收码输入脚	P3.3(IR2)	--

#### ✓ Set RF\_DEC\_SET

参数	数值范围/选项	说明	预设值	单位
CLK_DIV_SEL	24MHz/16MHz/8MHz/ 6MHz/3MHz/2MHz	数据解码时钟频率	2MHz	--
DIN_DB_SEL	0ns /250ns /500ns 1000ns	数据弹跳时间	500ns	--
DIN_TYP	Data Type1/Data Type2	数据类别	Data Type1	--
DOUT_REV	First input data is LSB/ First input data is MSB	解码输出数据反向	First input data is LSB	--
HEADER_EN	No HEADER / With HEADER	头码选择	No HEADER	--

调适方向建议：

首先先确认RF DATA连接脚位，并在RFIN\_SEL选择该输入脚位(如右图)。

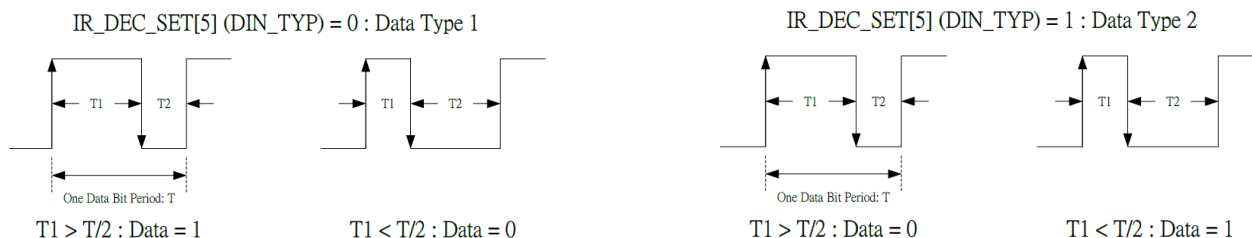
CLK\_DIV\_SEL设定：内部Counter上限为32767，再不超过此值的前提下设定该时钟频率，其公式如下：

$CLK\_DIV\_SEL \leq 32767 / (\text{数据时间宽度})$ 。例如某RF资料宽度最长为Stop(10ms)，则 $CLK\_DIV\_SEL \leq 32767 / 10\text{ms}$

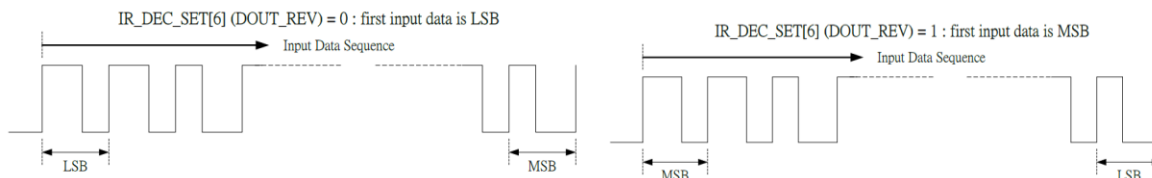
>> CLK\_DIV\_SEL <= 3.2MHz

IR1/CH3/P0.5	33
CAP1/CH4/P0.6	32
CAP2/CH5/P0.7	31
INT0/CH6/P3.2	30
IR2/INT1/CH7/P3.3	29

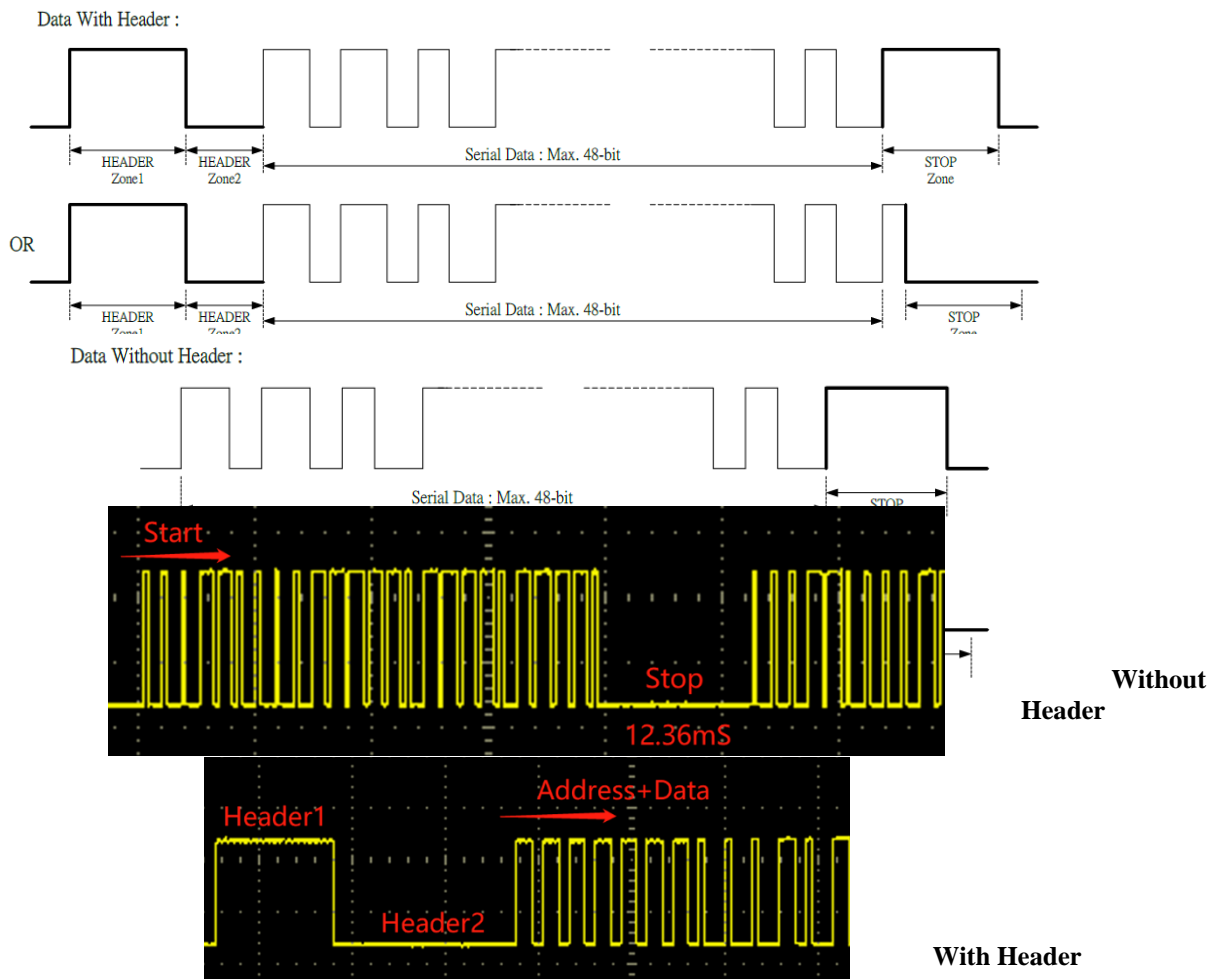
DIN\_TYP 数据类别：资料0和1的判断格式如下：



DOUT\_REV 设定：定义第一笔Bit为MSB或是LSB。



HEADER\_EN 设定：选择需不需要头码判断。这里要注意，若选择With Header，Header的Zone1和Zone2必需同时存在，无法设定0避免掉。



● Remote control code set

✓ Head code set

参数	数值范围/选项	说明	预设值	单位
Set Header Zone1 time width	0~32767	头码1(High)	5000	us
Set Header Zone2 time width	0~32767	头码2(Low)	7800	us

✓ Stop code set

参数	数值范围/选项	说明	预设值	单位
Set Stop time width	0~32767	结束码	5000	us

✓ RF decoding to UR Enable(✓)

✓ Learning function (✓)

参数	数值范围/选项	说明	预设值	单位
Learning data values set	0~65535	学习键数值	243	--
Set learn time	0~20	学习时间	10	sec
Judge the times of successful learning	0~20	判定学习成功次数	1	10times

参数	数值范围/选项	说明	预设值	单位
RF address value set	0~65535	地址码数值	54399	--

✓ RF data values set

● Enable OFF Button(✓)

参数	数值范围/选项	说明	预设值	单位
Button OFF	0~65535	Off键数值	13503	--

● Enable Speed1 Button(✓)

参数	数值范围/选项	说明	预设值	单位
Button speed 1	0~65535	速度1键数值	15903	--

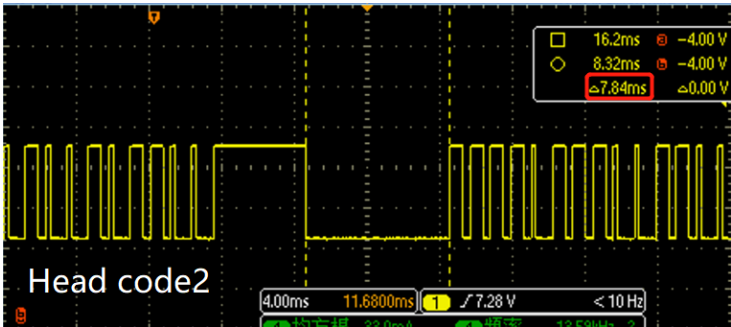
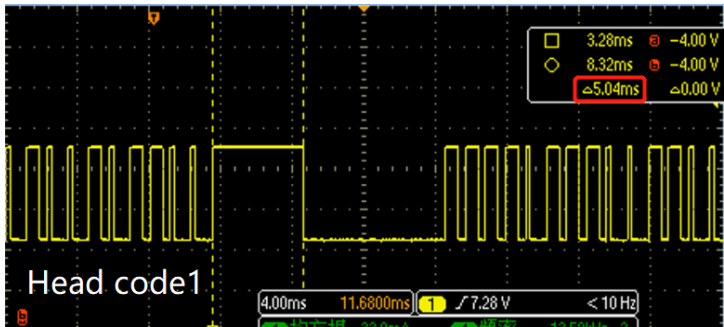
● Enable Speed2 Button(✓)

参数	数值范围/选项	说明	预设值	单位
Button speed 2	0~65535	速度2键数值	13983	--

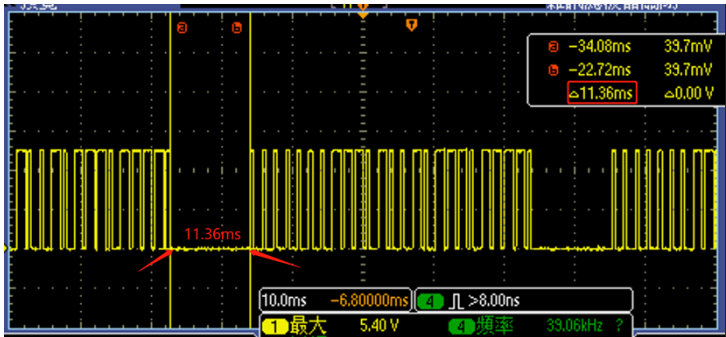
以此类推...

调适方向建议：

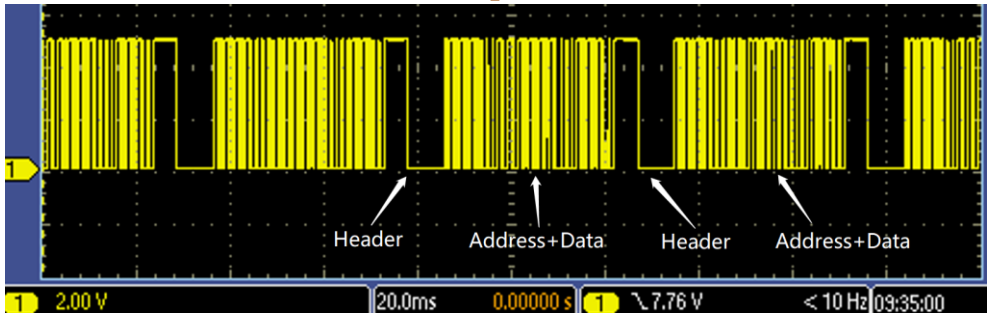
头码设定：请根据示波器上量测RF TX丢出来的波形，讯号开头宽度较宽的部分为头码，High的宽度为**Header Zone1 time width**; Low的宽度为**Header Zone2 time width**，例图如下：



结束码设定：请根据示波器上量测RF TX丢出来的波形，讯号结尾宽度较宽的结束码部分为**Stop time width**，例图如下：



假设遇到没有结束码的遥控器(如下图)，则可将**Stop time width**设定和**Header Zone1 time width**一样。



地址/数据码解码：当头码和结束码设定完成后，首先要先判断该遥控器的地址码以及数据码长度，**目前代码得预设值为16bits address + 16bits data**，若和预设值不同，需自行在代码上做修改，位置如右图(RF\_Decode.c)，每个暂存器为8bits，请根据遥控器该bits数做位移。

地址码/数据码bits数判断：在相同遥控器下，分别按下不同按键并且记录，前几个相同的bits为地址码，后几个不同bits为数据码，例如：(以下0和1依照Page18 **DIN\_TYP** 数据类别做转换)

```

83  IR_DEC_CTRL = IR_DEC_CTRL_REGS;
84  }
85  #endif
86  //***** RF_Decode_Read *****
87  //*****
88  //*****
89  //*****
90  #if (Remote_control_type == dRF_Control)
91  void RFDecode_Read(void){
92  if((IR_DEC_CTRL & 0x10) == 0x10){
93  IR_DEC_CTRL |= 0x01;
94
95  RF_Decode.RF_address = IR_DOUT0;
96  RF_Decode.RF_address += (IR_DOUT1 * 256);
97  RF_Decode.RF_data = IR_DOUT2;
98  RF_Decode.RF_data += (IR_DOUT3 * 256);
99
100  #if (RF_decoding_UR_Enable == 1)
101  RF_Decode.RF_address_ur = RF_Decode.RF_address;
102  RF_Decode.RF_data_ur = RF_Decode.RF_data;
103  #endif
104
105  #if (Remote_control_learning_function == Enable)
106  if(FLAG_RF_Learning_sw==0)
107  RFLearning_mode(); //遥控器学习模式
108  if(RF_Decode.RF_address==RF_Decode.RF_Learning_Address[0]+(RF_Decode.RF_
109  #else

```



OFF按键 : 1111 1110 0010 1011 1111 1101 0010 1100

Speed1按键 : 1111 1110 0010 1011 1111 1000 0111 1100

Speed2按键 : 1111 1110 0010 1011 1111 1001 0110 1100

蓝色部分为地址码，剩余不同部份为数据码，所以便是16bits address + 16bits data，虽然20bits address+ 12bits data在这例子也通用，但为了资料处理好撰写，建议以8的倍数做区分。

Note：在确认地址码和数据码bits数时，建议记录3个按键以上，避免误判。

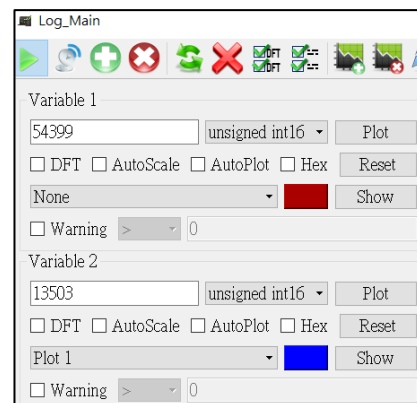
勾选**RF decoding to UR Enable**，并开启Log\_Main观测解码hex值，  
(Note:做这动作前请记得至UartSystem.h <Configuration Wizard>将UART Enable勾选起来，否则UART功能不会作动)。

如右图，按下任一按键观察Variable1和Variable2数值，1的为**RF address value set**，2的为该按键功能的**data values**。

Note：若开启**Learning function**，**RF address value set**可以不用填写。

学习功能(对码)：开启**Learning function**，依照前面叙述的Uart观测数值方式，将对码数值填入**Learning data values set**，**Set learn time**为学习时间，从MCU work那一刻开始算起，过了该时间便不再学习；**Judge the times of successful learning**为判断学习成功次数，每10次为一单位，意思为当MCU判断到该学习按键x次数，认定为学习成功便将该遥控器的地址码，学习储存至EEPROM。解码完毕记得将**RF decoding to UR Enable**关闭。

Note：若需要开启学习功能，请记得至Eeprom.h <Configuration Wizard>将EEPROM Enable勾选起来。





## IR解码

### ■ 配置文件：IR\_Decode.h <Configuration Wizard>

#### ● Customize

#### ✓ Decoding with Capture(✓)

#### ● IR decoding to UR Enable(✓)

参数	数值范围/选项	说明	预设值	单位
Set the number of data to observe	1~50	观测第几个Bit数值	1	--

参数	数值范围/选项	说明	预设值	单位
Set IR High Head counter_min	0~65535	头码最小值	11100	--
Set IR High Head counter_MAX	0~65535	头码最大值	15000	--
Set IR High Head counter_error	0~10000	头码误差量	2000	--

#### ✓ Set read IR data format

参数	数值范围/选项	说明	预设值	单位
IR_data_format	Low_level/ High_level	IR数值判断准位	High_level	--

参数	数值范围/选项	说明	预设值	单位
Set IR 0 data counter	0~65535	数据0计数	1681	--
Set IR 1 data counter	0~65535	数据1计数	4955	--
Set IR data counter_error	0~10000	数据计数误差量	400	--
Set the number of bits of IR data	0~16	数据码bit数	16	Bits
Set the number of bits of the IR address	0~16	地址码bit数	16	Bits
Set IR address value	0~32767	地址码数值	7038	--

#### ✓ IR data values set

#### ● Enable OFF Button(✓)

参数	数值范围/选项	说明	预设值	单位
Button OFF	0~65535	Off键数值	4335	--

#### ● Enable Speed1 Button(✓)

参数	数值范围/选项	说明	预设值	单位
Button speed 1	0~65535	速度1键数值	32895	--

#### ● Enable Speed2 Button(✓)

参数	数值范围/选项	说明	预设值	单位
Button speed 2	0~65535	速度2键数值	16575	--

以此类推...

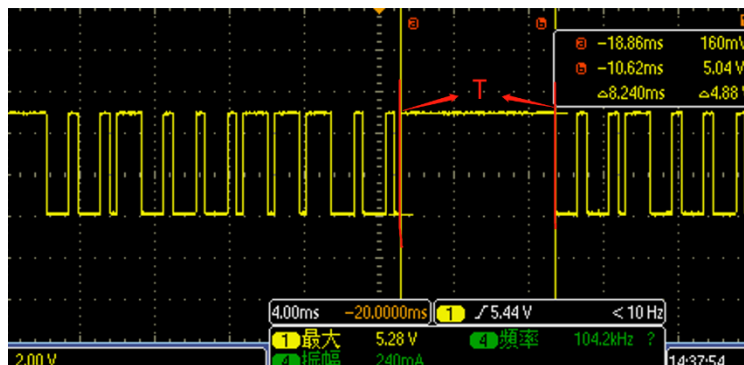
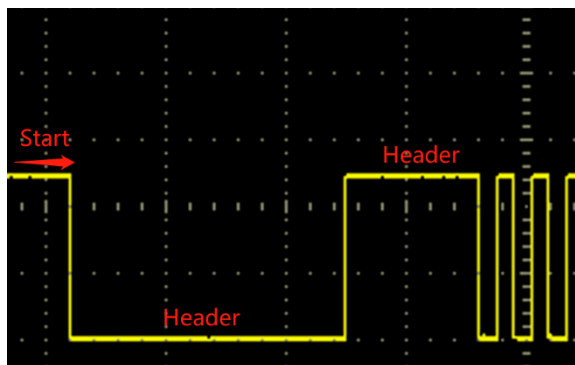
调适方向建议：

若以 Capture 作为解码，请先至 **Capture.h <Configuration Wizard>** 并将 External Capture Enable/Disable 勾选起来，以及至 **Interrupt.h <Configuration Wizard>** 将 CAPTURE Interrupt 勾选起来，确认 Capture 使用的脚位并且设定在 CAPPINSEL，最后将 **Decoding with Capture** 勾选。

解头码：量测任一按键的讯号，并在示波器上记录头码的时间宽度(如右下图)，**因内部 Capture 只能采用正缘触发，故以下的头码宽度都是以正半周讯号为主**，但若头码同时有负半周讯号和正半周讯号(如左下图)，有两种解法：1. 负半周讯号直接跳过，只判断正半周讯号；2. 改采用 TIMER+INT 解码。

**IR High Head counter 数值为：**量测到的时间宽度 \* E\_CAPCKS(从 Capture.h 设定)

例如右上图量测到为 8.24ms，CAPCKS 设定为 48MHZ/16



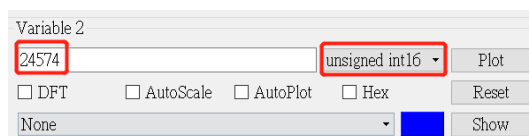
则 IR High Head counter = 8240us \* 3MHz = 24720.

(这里要注意一点，如果 Counter 出来的值过小(比如都是小于 1000)，那代表 E\_CAPCKS 要再往上设定，一般建议头码的 Counter 应大于 10000 以上，如果头码的 Counter 过小的话，在后面数值的 Counter 判断，很容易会有误判的情况发生)

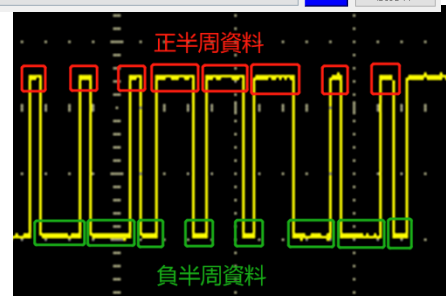
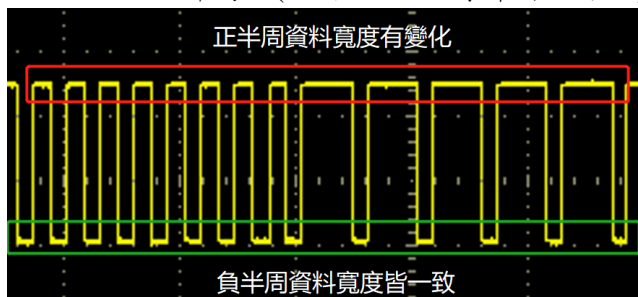
而 counter min/MAX 的差别在于，有些遥控器的头码宽度，每个按键都会有些微差异性，把量测到最小的宽度填入 min，最大填入 MAX，而如果每个按键的头码宽度差异性不大的话，两者值填入一样。

**IR High Head counter\_error** 一般填入 IR High Head counter \* 10% 的值(若非整数，可往上提升至整数)。例如 IR High Head counter = 24720; 24720 \* 10% = 2472，则 IR High Head counter\_error 填入 3000(多估抓一些误差值)

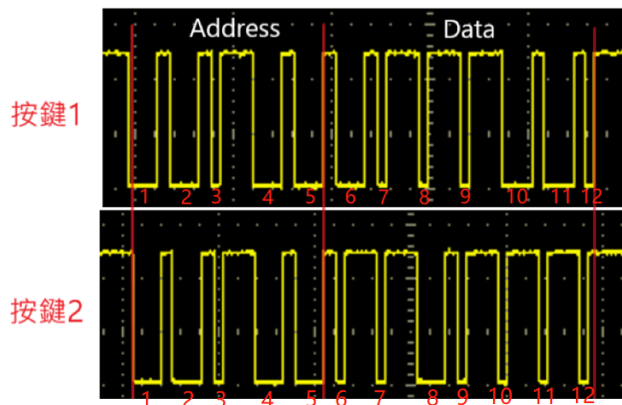
将 **IR decoding to UR Enable** 开启，当按下任一按键，若 Variable1 有数值的跳动，且和填入的 IR High Head counter 差不多时，代表头码有正确解码到。



**设定 IR\_data\_format:** 此设定为抓取资料(包含地址)的方式，Low\_level 为抓取负半周资料，High\_level 为抓取正半周资料(如右图)当出现如下图的资料码的话，建议设定为 High\_Level，或是改用 TIMER+INT 解码。(已可以明显判断 0 和 1 为主)



量测数据波形确认地址 bit 数以及数据 bit 数(相同遥控器，不同按键下，一样的码为地址码，不一样的码为数据码)，范例如下：



左图为两个按键的地址+数据码

可以观测出,从第6个bit开始便不一样(IR\_data\_format 设定Low\_Level), 所以Address bit数为5, Data bit数为7  
将7填入Set the number of bits of IR data

将5填入Set the number of bits of the IR address

(建议在区分判断Address和Data时, 请分别观测3个按键以上, 避免只判断2个按键容易会有误区)

设定data0和1的定义, 一般会将时间宽度短的定义为0; 长的定义为1。

如上图, Address即为11011(设定为Low\_Level, 只看负半周宽度), 11011为十进制” 27” , 将27填入Set IR address value.

**Set IR 0 data counter、Set IR 1 data counter** : 判断Data0/1的Counter数, 开启**IR decoding to UR Enable**, Set the number of data to observe此参数需搭配UR中的Variable2数值, 以上图为例: 比如说我已知第1个bit为1; 第3个bit为0, 首先判断Data=1的Counter, 则在Set the number of data to observe填入1, 并且按下该按键观测Variable2数值, 将其数值填入**Set IR 1 data counter**, 反之Data=0的Counter则是在Set the number of data to observe填入3, 观测Variable2数值, 将其数值填入**Set IR 0 data counter**。(注意: 判断顺序必需由第1个bit开始, 并且填入相对应的Counter, 再由左至右找寻另一个data的位置, 如果bit1=0的话, 再找bit=1的位置, 反之亦然, 但如果一开始直接跳bit的话, 程式内部可能会直接跳Error)。当设定完成后, 观测UR的Variable3数值, 如果此数值和IR\_address\_value一样的话, 代表到目前为止解码OK.

**IR data values set** : 最后将UR的Variable4填入相对应按键名称内, 例如我按下Off功能按键, 且Variable4为198, 则在Button OFF填入198, 以此类推...

## VSP控制

### ■ 配置文件: Motor.h <Configuration Wizard>

#### ● Set motor control program

#### ✓ Set Peripheral Control

#### ● VSP control commands Enable/Disable (✓)

参数	数值范围/选项	说明	预设值	单位
Set VSP_CH	CH0~CH7	VSP A/D	CH4	--

### ■ Slope control

参数	数值范围/选项	说明	预设值	单位
Set VSP start voltage	0~50	VSP启动电压	5	0.1V
Set VSP max voltage	0~50	VSP最大电压	45	0.1V
Set VSP stop voltage	0~50	VSP停止电压	3	0.1V

### ■ Stage control

参数	数值范围/选项	说明	预设值	单位
Set VSP SPEED1 speed	--	VSP第一段速度	100	rpm
Set VSP SPEED2 speed	--	VSP第二段速度	150	rpm

以此类推...

◆ Observe Speed\_UI parameters Enable(√)

参数	数值范围/选项	说明	预设值	单位
Set VSP_SPEED_UI_1	--	VSP第一段速度	350	--
Set VSP_SPEED_UI_2	--	VSP第二段速度	800	--

以此类推...

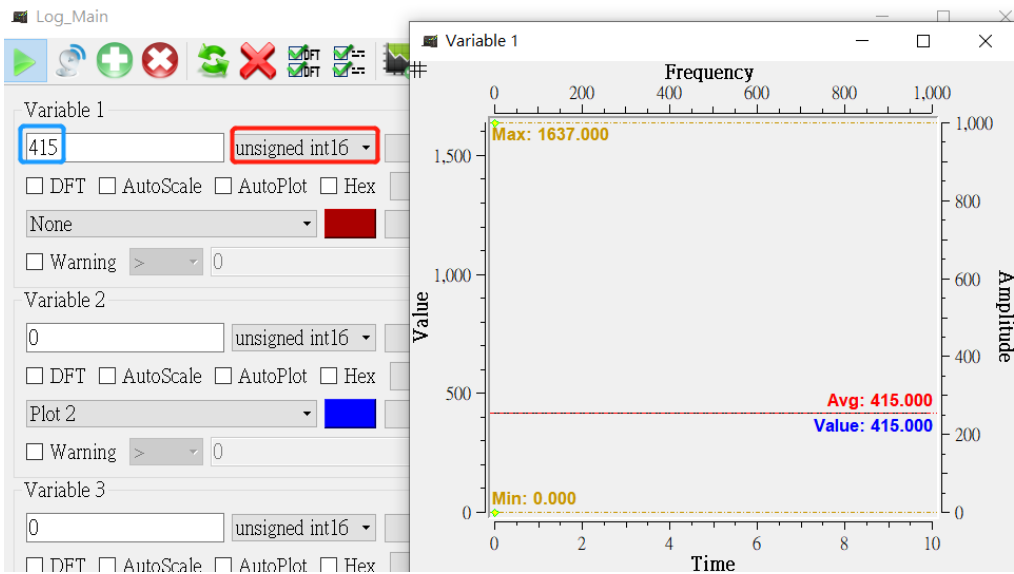
调适方向建议：

VSP外部控制功能，首先确认VSP的A/D Channel，并填入VSP\_CH。VSP共有两种控制模式，分别为Slope control、Stage control，目前VSP只供测试用，Stage control目前只开放控制速度功能，后续若有案子需求，再额外更新。

补充：速度 UI

在速度外环下，无论是遥控器控制或是VSP控制，都会有个UI设定参数可做使用，此值的设定是为了加/减转速快速反应，其调适说明如下：

将所需的各段转速值填入之后，开启 Observe Speed\_UI parameters Enable，在吊扇各段位速度底下观察 Variable1 参数，如下图所示：



假设按下速度1按键(SPEED1 speed)，当吊扇速度稳态后，此时Variable1值为416，则将416填入 Button1\_SPEED\_UI，SPEED2\_speed对应Button2\_SPEED\_UI，以此类推...。若实际测出来有过冲的现象，可向下微调此参数。需注意此功能只限定转速控制下设定。

Speed\_UI调适完毕记得将Observe Speed\_UI parameters Enable 关闭

2.3 顺/逆风启动参数调适

- 配置文件：Motor.h <Configuration Wizard>
- Set Fairwind and Headwind judgment function
  - BEMF Fairwind/Headwind judgment (resistance) Enable/Disable (✓)

参数	数值范围/选项	说明	预设值	单位
BEMF Fairwind/Headwind adjustment process	LEVEL_1/LEVEL2	顺逆风判断步骤	LEVEL_2	--

✓ Set BEMF A/D Channel

参数	数值范围/选项	说明	预设值	单位
Set BEMF_V_CH	CH0~CH7	BEMF_V A/D通道	CH3	--
Set BEMF_W_CH	CH0~CH7	BEMF_W A/D通道	CH5	--

✓ Set Speed for judging Fairwind/Headwind

参数	数值范围/选项	说明	预设值	单位
Set the minimum connection speed for judging Fairwind	--	判断顺风最小转速	23	rpm
Set the maximum connection speed for judging Fairwind	--	判断顺风最大转速	180	rpm
Set the maximum connection speed to judge the headwind	--	判断逆风最大转速	100	rpm
Set the duration of Fairwind start judgment	--	判断顺逆风周期	100	ms
Set Headwind_high_speed_start_Strategy	PLL/ABS	高速逆风启动策略	ABS	--

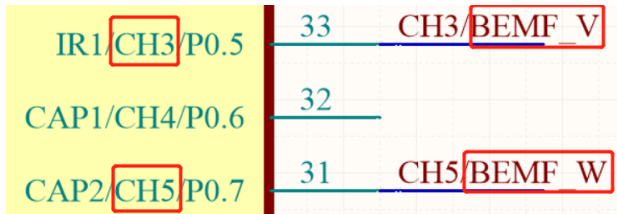
调适方向建议：

先确认两相的分压电阻连接的AD Channel，分别填入 Set BEMF\_V\_CH、Set BEMF\_W\_CH。顺逆风调适初期，可先将BEMF调机流程选择LEVEL\_1，此时可在马达静止状态，用手拨扇叶使马达旋转，测试顺逆风启动的状态。

判断衔接顺风转速范围：衔接顺风的条件机制，在最小转速到最大转速之间，皆可衔接至顺风，可根据实际吊扇衔接状况做微调，会限制最小转速是因为，当转速过小时，会无法判断到BEMF，所以无法得知马达位置，故限制一个下限值。

判断衔接逆风转速：在此设定的转速值以下(Set the maximum connection speed to judge the headwind)，皆可满足衔接逆风机制的条件，若大于此转速则Free Run，直到马达低于此转速，可依照实际马达状况做微调。

Headwind\_high\_speed\_start\_Strategy，逆风衔接策略，分别有PLL策略以及ABS策略，目前建议使用ABS策略即可。





参数	数值范围/选项	说明	预设值	单位
Set Fairwind/Headwind angle(CW_CCW=0)	0~383	CW_CCW=0 顺逆风初始角度	0	--
Set Fairwind/Headwind angle(CW_CCW=1)	0~383	CW_CCW=1 顺逆风初始角度	64	--
Set IQ Fairwind/Headwind starting current(IQ_CMD)	--	顺逆风启动Iq电流	2000	mA
Set ID Fairwind/Headwind starting current(ID_CMD)	--	顺逆风启动Id电流	0	mA
Set the initial value of Fairwind/Headwind current(IQ_UI)	0~32767	顺逆风启动初始内环输出量	8000	--
Set Headwind to Fairwind target speed	--	逆风衔接进入外环速度	45	rpm
Set the judgment counter for the static BEMF value of the motor	0~10000	判断马达静止状态延迟时间	300	ms

### ✓ Observe BEMF\_V\_W Parameters (✓)

调适方向建议：

**顺逆风衔接判断角度设定**：此参数(Fairwind/Headwind angle(CW\_CCW=0/1))建议不做更改。

**顺逆风衔接IQ设定**：根据衔接进顺/逆风的情况做微调，若衔接后马达会有很大的撞击声，则可将衔接IQ电流往下做微调的动作。

**顺逆风衔接IQ\_UI初始值设定**：设定此参数是让衔接进顺逆风时，IQ\_OUT有个初始值在，不会从0开始。

**Headwind to Fairwind target speed**：逆风启动后，转正转衔接close loop的转速值，此值为决定在逆风启动下，衔接至close loop的转速值，太小可能会造成衔接上的失败。

**Set the judgment counter for the static BEMF value of the motor**：判断马达静止时间，因马达速度过低时，无法依靠BEMF来判断马达目前处于低速运转还是静止不动，故当BEMF\_V-W低于某值以下(代码预设为5，可至Motor.c 第765行做修改)，直接判定为马达静止，而设定的判断时间便是避免杂讯产生误判的延迟时间。

## 2.4 保护参数调适

- 配置文件：**Motor.h** <Configuration Wizard>
- **Set motor protection function**
- **过/欠电压保护**

- 参数配置如1.3节(保护参数配置)所示。

调适验证方向：

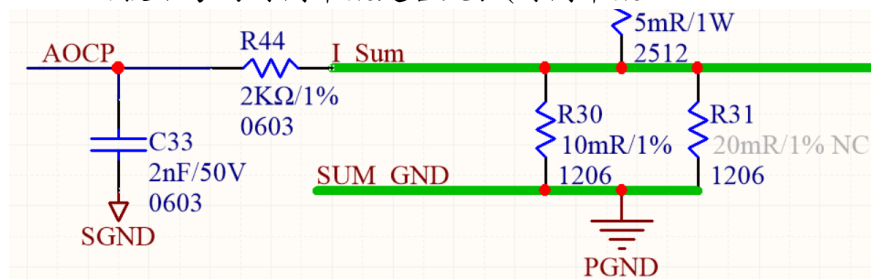
1. 依据客户提供的过、欠电压与恢复电压条件进行设定。
2. 过电压及恢复测试：启动吊扇后，将电源供应器的电压升至过电压准位，确认吊扇是否进入停机保护，接着将电源供应器的电压降至过电压恢复准位，判断吊扇是否正常启动。
3. 欠电压及恢复测试：启动吊扇后，将电源供应器的电压降至欠电压准位，确认吊扇是否进入停机保护，接着将电源供应器的电压升至欠电压恢复准位，判断吊扇是否正常启动。

- **过电流保护-测试硬件(重点测试)**

- 参数配置如1.3节(保护参数配置)所示。

调适验证方向：

1. 验证硬件，先将AOCP此点和5V短路，马达可不接，利用UART监看是否有跳OCP，确认没问题再做以下短路测试。
2. 启动吊扇后，任意将两相(UV or UW or VW)输出进行短路，以判断吊扇是否进入停机保护，接着重新上电，确认吊扇是否能正常工作，并多次重复该步骤，以确保MOSFET能充分保护。
3. 若短路实验后吊扇无法重启，则检查MOSFET的规格是否满足、AOCP回授讯号在布线路径上是否有干扰、AOCP回授讯号的时间常数是否太长(时间常数： $RC \leq 2k * 2nf = 4us$ )。



- **堵转保护**

- 参数配置如1.3节(保护参数配置)所示。

调适方向建议：

可利用手拉住扇叶，确认吊扇进入失步/高频状态，堵转保护是否正常作动。



## 2.5 其它参数配置

- 配置文件：**Motor.h <Configuration Wizard>**
- Set motor control program
  - Set other functions

参数	数值范围/选项	说明	预设值	单位
Set CW/CCW steering	CW/CCW	转子方向设定	<b>CW</b>	--
Set Stop_Fun stop speed	--	PWM停止输出转速	<b>60</b>	rpm

调适方向建议：

设定CW/CCW转向依据所需的扇叶方向做修改。

Stop\_Fun 停止转速在这里指的是，当遥控器按下停止键时，马达转速低于此参数时，PWM才会停止输出，大于此参数的转速值时，PWM皆还在输出当中，这是为了顺风衔接顺畅而设定的。其值可设定在低速判断不到顺风转速的衔接点。

### ✓ CW/CCW test Enable/Disable(✓)

参数	数值范围/选项	说明	预设值	单位
Running time after changing direction	--	换方向启动后运转时间	<b>10000</b>	ms
Stop this time before catching the wind	--	停止此时间后再接顺风	<b>2000</b>	ms
Operating time after connecting downwind	--	衔接顺风后运转时间	<b>5000</b>	ms

调适方向建议：

此功能为测试用。其测试流程为：马达启动运转(初始依照Set CW/CCW steering设定方向)，运转时间为**Running time after changing direction**，接着马达停止运转，时间为**Stop this time before catching the wind**，接着衔接顺风启动，运转时间为**Operating time after connecting downwind**，最后换方向启动，时间为**Running time after changing direction**，不断循环...

### ✓ IPD Enable/Disable(✓)

- Automatic cycle test of initial position Enable/Disable(✓)

调适方向建议：

此IPD功能需搭配配置文件 **IPD.h <Configuration Wizard>**做调适，**Automatic cycle test of initial position**此功能可自动在马达静止状态，不断循环打IPD，用来做IPD功能准确性，可将IPD Pattern此参数拉到UART观测，在马达转子不同角度下，固定位置的Pattern是否皆一样。

- 配置文件 **IPD.h <Configuration Wizard>**
- Set IPD LEVEL

参数	数值范围/选项	说明	预设值	单位
I_SHORT	0.15V/0.2V/0.25V 0.3V/0.35V/0.4V 0.45V/0.5V	IPD OCP LEVEL	<b>0.15V</b>	--
AOC PEN	Disable/Enable	类比OCP功能	<b>Enable</b>	--
DOC PEN	Disable/Enable	数位OCP功能	<b>Disable</b>	--
IPD Path Select	IPD Current Compare from AOC Path IPD Current Compare from OPA Path	IPD OCP判断路径	AOC Path	--

- Set IPD IAECYC

参数	数值范围/选项	说明	预设值	单位
IAECYC	48MHz/24MHz/12MHz/6MHz	IPD Counter Frequency	<b>24MHz</b>	--

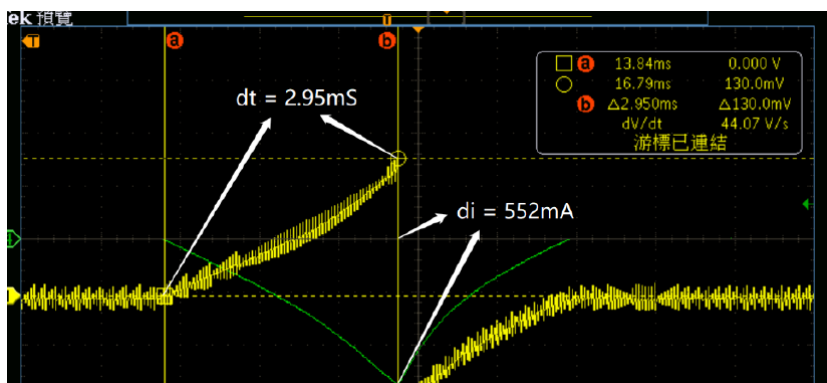
调适方向建议：

IPD所产生的声音大小，和I\_SHORT成正比，若设定至0.15V时声音还是很大，只能增加Shunt R的阻值(需注意电阻的瓦数)。

**AOCPEN、DOCPEN、IPD Path Select**：不建议更动。**IAECYC**：此参数为IPD重要参数，用来计数OCP dt数值，转子在不同位置下所执行的IPD，取最小Counter为该位置的Pattern，其Counter最大值为65535，可依照量测的OCP dt宽度，去得知所需要的IAECYC，不同的IAECYC可判定到的最大dt宽度，计算如下：

$$\begin{aligned} 48\text{MHz} &: (1/48\text{MHz}) * 65535 \approx 1.3\text{ms} \\ 24\text{MHz} &: (1/24\text{MHz}) * 65535 \approx 2.6\text{ms} \\ 12\text{MHz} &: (1/12\text{MHz}) * 65535 \approx 5.2\text{ms} \\ 6\text{MHz} &: (1/6\text{MHz}) * 65535 \approx 10.4\text{ms} \end{aligned}$$

以右图为例，黄色波形为OCP讯号，dt为2.95ms，假设此时IAECYC选择48MHz或24MHz，因为dt皆大于Counter的最大值产生溢位，你会发现马达在启动时马达依然会倒转，所以只dt只适合选择12MHz。



✓ **Brake control Enable/Disable(✓)**

参数	数值范围/选项	说明	预设值	单位
Set braking force	0~100	刹车Duty量	99	%

调适方向建议：

刹车力道，顺逆风Diode架构或是内部估测器用，暂时没用到。

- 配置文件：**Ipwm.h <Configuration Wizard>**
- **IPWM Enable/Disable(✓)**

参数	数值范围/选项	说明	预设值	单位
IPWM Frequency SET	--	IPWM频率	4000	Hz
IPWM MODE SET	Force Low/Force High/Active Low/Active High	IPWM模式	Active High	--

● **Set buzzer working time**

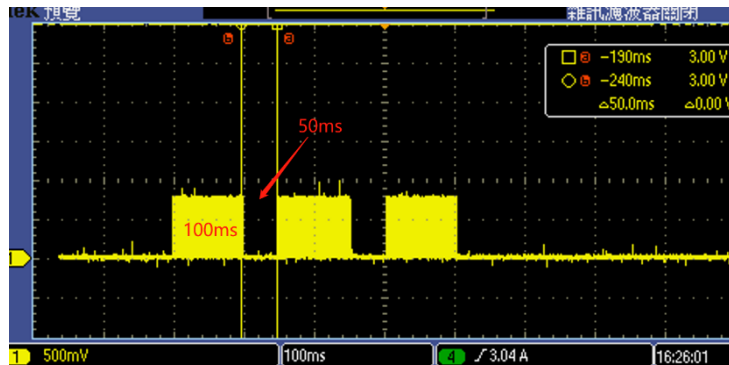
参数	数值范围/选项	说明	预设值	单位
IPWM Times_Init	1~255	IPWM执行次数	1	times
IPWM Enable time_Init	1~10000	IPWM执行时间	100	ms
IPWM Disable time_Init	1~10000	IPWM间隔时间	50	ms

以此类推...

调适方向建议：

此功能需搭配Buzzer做使用，并且接到MCU内部IPWM脚位。将Buzzer工作频率填入IPWM Frequency。

IPWM MODE：若Buzzer为外激式(无源)的，建议设定为Active High，若为自激式(有源)的Buzzer，建议设定为Force High。底下的Buzzer working区分为开机设定、关机设定、错误设定，可依照客户所需要的buzzer鸣叫的次数、时间长短自行做调适，其中IPWM Times为次数，Enable time为鸣叫时间，Disable time为关闭时间，例如Times=3; Enable time = 100ms ; Disable time = 50ms，则IPWM波形如下：



## 3. 程序架构

### 3.1 程式流程

#### ■ 配置文件：main.c

#### ■ Main Function

- 系统初始化设定
- while(1){
  - ✓ RFDcode\_Read(); // RF 解码读取
  - ✓ IRDcode\_Read(); // IR 解码读取
  - ✓ EEP\_Storage(); // EEPROM 储存
  - ✓ 怠速启动电压零交越点判断与转速估测
  - ✓ UART丢数据
- }

#### ■ 配置文件：Interrupt.c

#### ■ Timer 0 ISR(Time Based : 1ms)

- Correction\_Current\_AD\_offset(); //功率控制-母线电流A/D初始校正
- Motor\_Control(); //马达的错误判断、正反转控制、马达启动
- Buzzer\_Fun(); //Buzzer.

#### ■ Timer 1 ISR(Time Based : 10ms)

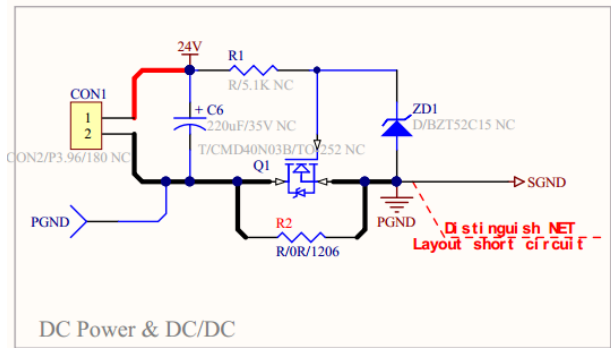
- Vsp\_Fun(); //外部VSP命令接收
- FaultLock\_Fun (); //系统堵转、过/低速保护
- Vbus\_Protect\_Fun (Vbus\_avg); //系统母线电压侦测、保护判断
- Temperture\_Protect\_Fun (Temperture\_avg); //系统过温保护
- Phase\_OCP\_Protect\_Fun() //系统相电流保护
- LACK\_OF\_PHASE\_Function(); //马达静止缺相判断
- MotorLackPhase\_Run\_Fun(); //马达运转缺相判断

#### ■ PWM MAX ISR(Time Based : Fpwm)

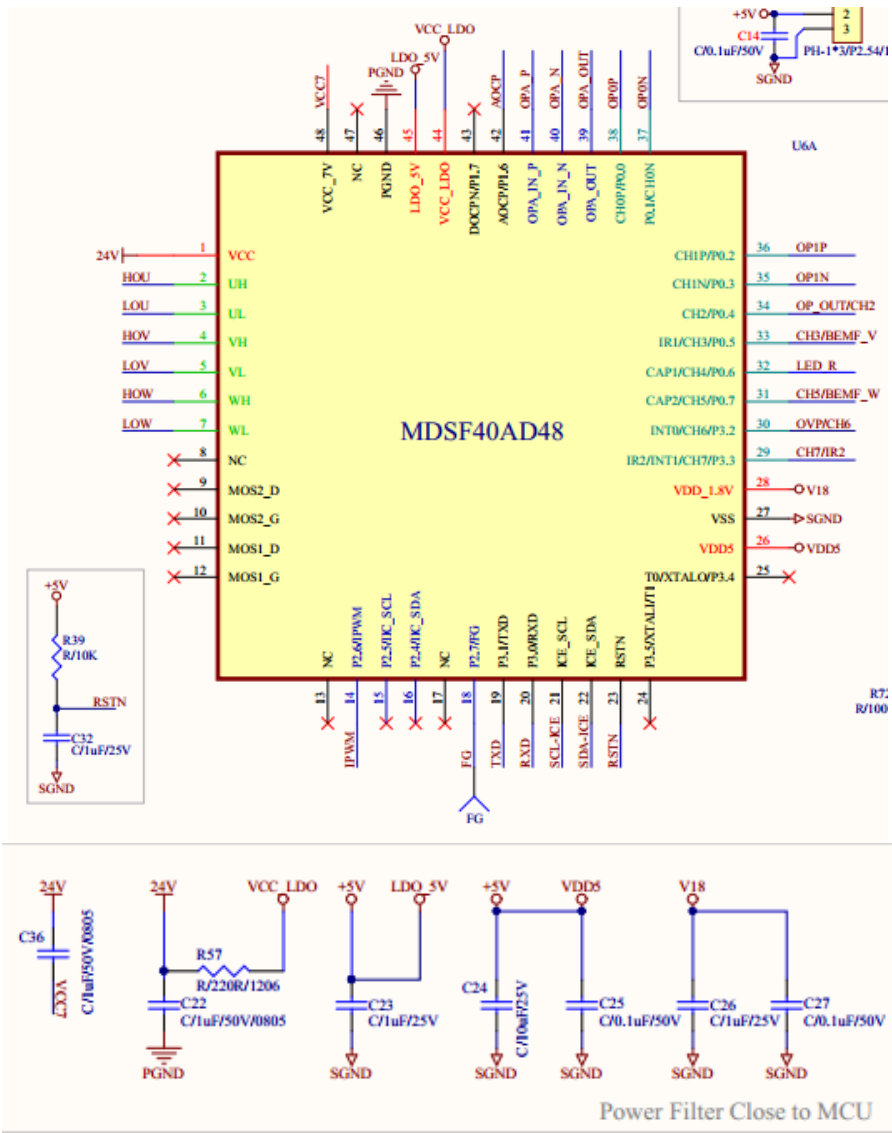
- 怠速启动电压零交越点判断与转速估测，旗标开启

## 4. 参考电路设计

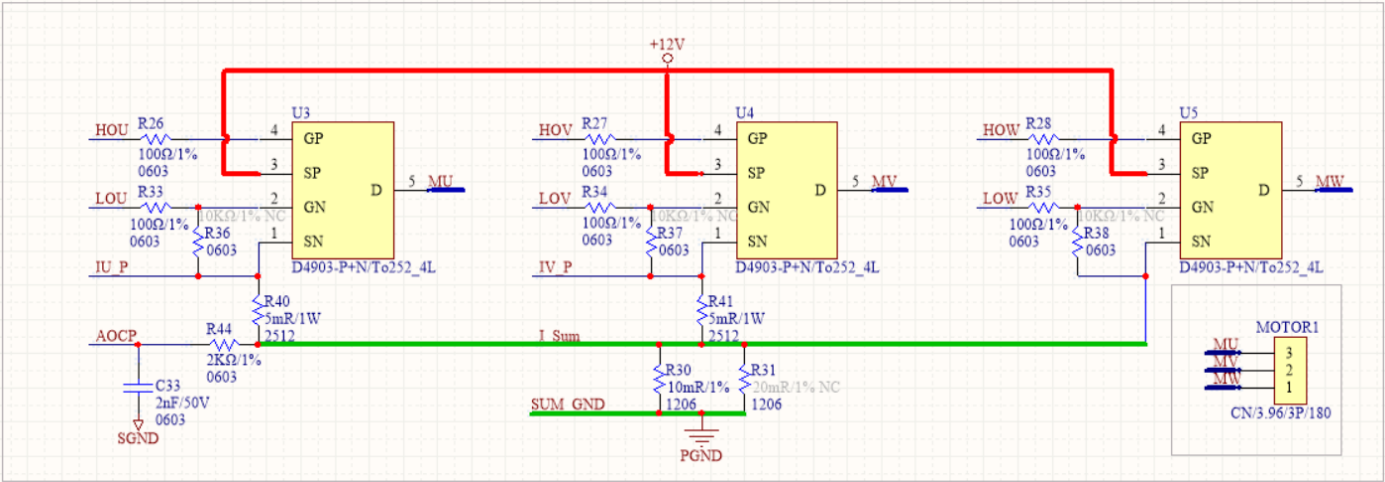
### 4.1 电源输入电路



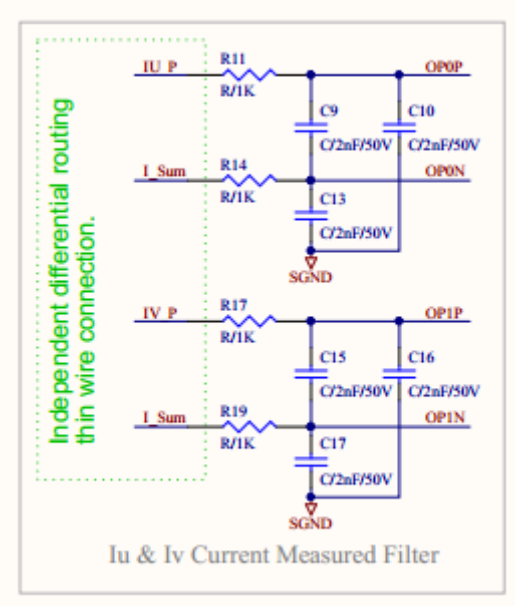
### 4.2 核心单元及周边电路



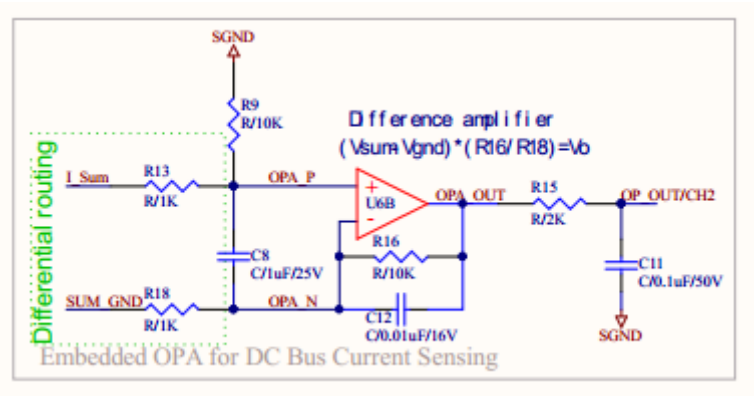
4.3 三相全桥变频器



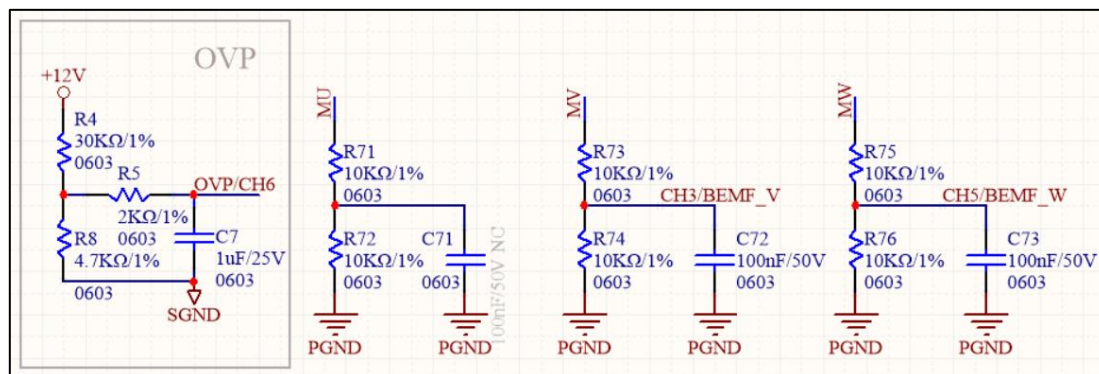
4.4 两相采样电路



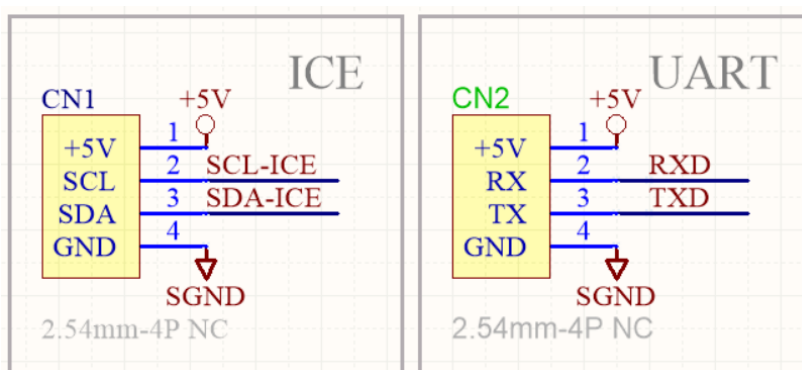
4.5 母线电流采样电路



## 4.6 BEMF回授电路



## 4.7 外部端口



## 4.8 其他周边

