

MDSF40

Low Voltage Ceiling Fan

Application Note

Sensorless Based High-Speed PMSM Solution

目錄

目錄	2
1. 基本規格參數配置	3
1.1 電機參數配置	3
1.2 MOC 及硬體參數配置	4
1.3 保護參數配置	6
2. 低壓吊扇參數調適流程	10
2.1 啟動參數調適	10
2.2 周邊控制功能	14
2.3 順/逆風啟動參數調適	27
2.4 保護參數調適	29
2.5 其它參數配置	30
3. 程序架構	33
3.1 程式流程	33
4. 參考電路設計	34
4.1 電源輸入電路	34
4.2 核心單元及周邊電路	34
4.3 三相全橋變頻器	35
4.4 兩相採樣電路	35
4.5 母線電流採樣電路	35
4.6 BEMF 回授電路	36
4.7 外部端口	36
4.8 其他周邊	36

1. 基本規格參數配置

1.1 電機參數配置

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

參數	數值範圍/選項	說明	預設值
Motor Pole	0~30(2的倍數)	馬達極數	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電流誤差判斷值	32767

調適方向建議：

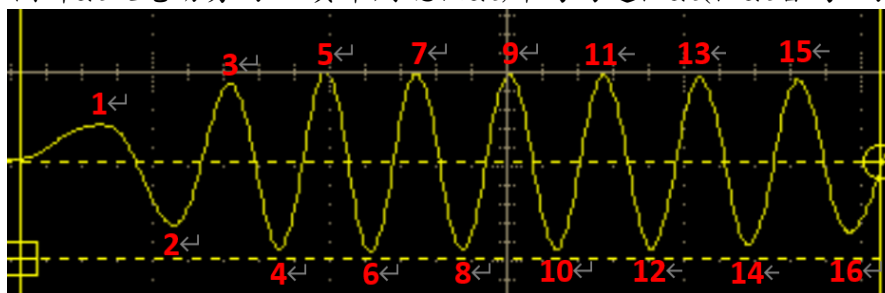
極數量測：

在無法確切得知馬達的極數規格時,可以採用量測反電動勢來取得極數,方法如下:

Step1. 示波器探棒隨意勾取馬達任一兩相之間(ex: U對V or V對W...),馬達無需接在控制板上

Step2. 利用手轉或是外力讓馬達旋轉一圈

Step3. 依照下圖計數反電動勢的正/負半周總個數,即為馬達極數(極數皆為2的倍數)



SMO:

Sliding-Mode Current Observer主要用於估測電機轉子位置，對參數變動干擾有一定程度的強健性，因此SMO_G、SMO_F參數無需透過R、L參數計算而得。當電機操作至額定功率期間，若出現電流異常發散導致停機保護，可以試誤法方式調適dSMO參數 (**G必需小於F**)。

SMO_Kslf為滑模電流觀測器於估測轉子位置時的濾波器係數，該參數與電機電氣頻率呈線性關係，也間接影響實際轉子位置與估測轉子位置間的相位關係。當電機操作至額定功率期間，若出現電流異常發散導致停機保護，可以試誤法方式調適SMO_Kslf參數。

$SMO_Gain = Kslide1 / MaxSmcError$ ，Kslide1為SMO線性增益，可適當調整SMO_Gain來間接改變Kslide1。

1.2 MOC及硬體參數配置

■ 配置文件：Pwm.h <Configuration Wizard>

■ Set MPWMDATA

參數	數值範圍/選項	說明	預設值	單位
Set PWM Frequency	12000~40000(10的倍數)	載波頻率	15000	Hz

■ Set MPWMDDB

參數	數值範圍/選項	說明	預設值	單位
Set Deadband Time	1us/1.5us/2us/3us/4us/5us	設定死區時間	1us	--

調適方向建議：

PWM輸出頻率：

PWM頻率設置越高，雖會增加MOSFET的切換損與電機發熱，進而影響系統效率，但可有效抑制電機操作於高速應用下的電流/功率/速度漣波、振噪，提升整體系統性能表現；而頻率設置低人耳會容易聽到(低於16K)，但能提高馬達最高轉速，建議感量越低的馬達設置的載波頻率要越大。

死區設定：

根據實際量測上下臂輸出波形判斷所需要的死區範圍，死區過大會影響馬達的速度特性，可能會無法提升到額定最高速。

■ 配置文件：Motor.h <Configuration Wizard>

■ Set Rshunt and OPA Gain

參數	數值範圍/選項	說明	預設值	單位
Rshunt	--	設定採樣相電阻阻值	100	0.1mR
OPA GAIN	1 Gain /2.5 Gain /5 Gain /10 Gain	內部OPA放大倍率	5 Gain	--

調適方向建議：

MDSFxx內部運算放大器增益有1、2.5、5、10四種放大增益值可供選擇，其選配原則取決於電流採樣解析度與電機額定相電流。

ShuntR * 馬達最大電流(I_{Peak}) < 0.5V

差動兩端會限制在0.5V以下，是因為內部OPA兩端最大限度電壓為0.5V。

OPA Gain * ShuntR * 馬達最大電流(I_{Peak}) < 2.5V

會限制2.5V是因為，內部A/D往上Offset了2.5V相電流有正/負，0~2.5V為負電流，2.5V~5V為正電流。

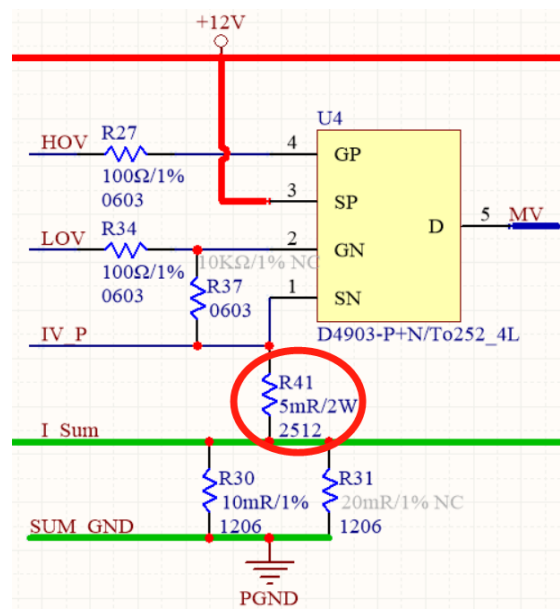
採樣電阻選配建議：

一般電流採樣 A/D 建議最大值對應馬達 $I_{peak(max)}$ 的兩倍，例如：

DC24V 馬達，在最高額定轉速下相電流 $I_{peak} = 2.7A$ ，取3A，則兩倍的採樣最大電流為6A

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

採樣電阻阻值越大，則採樣的精度也越高，但也需要考慮到採樣電阻功耗導致的溫升。在保證電阻溫升符合要求的情況下，電阻值應盡量提升。



- 配置文件：Pwm.h <Configuration Wizard>
- 開級驅動輸出模式

參數	數值範圍/選項	說明	預設值
Set MPWM_SWAP	MDSF05/ MDSF40 MDSFA0	Gate Driver輸出模式	MDSF05/MDSF40

調適方向建議：

當使用MDSF05(MCU：需外部配置Gate-Driver、MOSFET or IPM)與MDSF40 (MCM：內建P/N-type Gate-Driver，需外部配置MOSFET)時，需選定” **MDSF05/MDSF40**”，反之，當使用MDSFA0(MCM：內建N/N-type Gate-Driver，需外部配置MOSFET)時，則選定” **MDSFA0**”。(Note: 此選項千萬不能選錯,否則可能會造成MOSFET毀壞)

- 配置文件：Moc.h <Configuration Wizard>
- 空間向量脈寬調變(SVPWM)配置
- Set MOTOR_CONT2

參數	數值範圍/選項	說明	預設值
SVPWMMODE	7-SVPWM 5-SVPWM	SVPWM模式	5-SVPWM

調適方向建議：

(a). 當產品/系統以效率為優先指標時，可選定” **5-SVPWM**”，則SVPWM會以5段式切換輸出，降低MOSFET切換損與發熱，反之，當產品/系統以靜音、震動及干擾為訴求與關鍵指標時，則選定---” **7-SVPWM**”。

1.3 保護參數配置

- 配置文件：Motor.h <Configuration Wizard>
- 保護/誤動作狀態辨識 (ErrorStatus)
- Error code(MotorErrorState)

+	Error code(MotorErrorState)
+	Motor state(MotorState)
Error code(MotorErrorState)	
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)

調適方向建議：

當系統發生馬達異常停止或是無法啟動時，可利用保護狀態辨識，得知目前Error_Code輸出代號，以釐清系統的保護狀態。可在main.c檔內(第179行)，找尋if(UartFlag == 1)此判斷式，在第195行底下的Debug_(A~F)任一參數下使其等於” **MotorErrorState**” 此變數，範例如下：

```
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

於UartSystem.h <Configuration Wizard>將UART功能Enable，並且打開” DataLogger.exe ” 軟體即可監看此變數。

配置文件：Motor.h <Configuration Wizard>

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

參數	數值範圍/選項	說明	預設值	單位
Set Vbus A/D Channel	CH0~CH7	採樣母線電壓A/D通道	CH6	--
Set Vbus rate parameter	0~65535	採樣母線電壓校正參數 = (Vbus_avg / Input voltage)*1000	27500	--
OVP Values	0~4000	過電壓值	230	0.1V
OVP recovery Values	0~4000	過電壓恢復值	200	0.1V
UVP recovery Values	0~4000	欠電壓恢復值	75	0.1V
UVP Values	0~4000	欠電壓值	60	0.1V
OVP/UVLP judgment times	0~3000	OVP/UVLP判斷次數(10ms一次)	50	times

調適方向建議：

Vbus rate parameter：

先確認採樣Vbus的AD channel，並填入V_BUS_CH內，之後把” Vbus_avg” 此變數放入UART Debug監看(如同上面所提到的方式)，上電後將得到的Vbus_avg除以目前的輸入電壓，再乘上1000，得出的數值即為**Vbus rate parameter**，範例如下：

假設在工作電壓為DC12V下，所得到的Vbus_avg為330， $330 / 12 = 27.5$
 $27.5 * 1000 = 27500$ ，**Vbus rate parameter = 27500**。

OVP /UVLP setting：

將所希望的過電壓保護值以及欠電壓保護值，分別填入**OVP Values**、**UVP Values**參數，過電壓恢復值以及欠電壓恢復值，分別填入**OVP recovery Values**、**UVP recovery Values**。

OVP/UVLP judgment times參數為判斷OVP/LVP的Debounce次數，每10ms判斷一次。

- Locked-rotor protection (LRP) (✓)

參數	數值範圍/選項	說明	預設值	單位
Motor speed abnormally high value	--	馬達速度異常過高數值	400	rpm
Motor speed abnormally low value	--	馬達速度異常過低數值	20	rpm
LRP judgment cycle	10~30000	馬達堵轉判斷週期	2000	ms

調適方向建議：

LRP setting :

根據馬達的最高/低額定轉速，再累加/減一個量，來判斷馬達為堵轉狀態，高於**Motor speed abnormally high value**，認為過速保護；低於**Motor speed abnormally low value**，認為欠速保護
LRP judgment cycle 參數為判斷LRP的週期，建議不要設定太小，避免誤判。

■ Over temperature protection (OTP) (✓)

參數	數值範圍/選項	說明	預設值	單位
Set OTP A/D Channel	CH0~CH7	採樣OTP A/D通道	CH3	--
OTP A/D Values	0~1023	過溫A/D值	900	--
OTP recovery A/D Values	0~1023	過溫恢復A/D值	800	--
OTP judgment times	0~3000	OTP判斷次數(10ms一次)	5	times

調適方向建議：

OTP setting :

先確認OTP的AD channel，並填入**Set OTP A/D Channel**。OTP A/D Values為過溫保護當下的A/D值，OTP recovery A/D Values為清除過溫保護的A/D值，若可得知NTC的R/T表，可直接使用分壓公式做推算。

OTP judgment times參數為判斷OTP的Debounce次數，每10ms判斷一次。

■ Lack-Phase Protection(LPP) (✓)

- 保留；待驗證可行性

■ Phase current protection (✓)

保留；待驗證必要性

- Phase Ia、Ib、Ic parameter (unit : mA) (700)
- PHASE_OCP_DURATION parameter (unit : ms) (50)

調適方向建議：

(a). 相電流異常保護，三相電流中，若大於所設定的**Phase Ia、Ib、Ic parameter**，認為電流異常保護，設定方向可將IaFb、IbFb放置UR觀察，在馬達額定轉速運轉下將其平均值紀錄，約抓個1倍左右填入該參數。

(b). **PHASE_OCP_DURATION parameter**：判斷相電流延遲時間

- Set Protection to retry
 - AOCP_Retry_Enable (✓)
 - POCP_Retry_Enable (✓)
 - FaultLock_Retry_Enable (✓)
 - MotorLackPhase_Run_Retry_ENABLE (✓)

參數	數值範圍/選項	說明	預設值	單位
Set the number of retries	0~255	報錯後重啟次數	5	times
Set retry delay time	0~32767	重啟延時時間	1000	ms

調適方向建議：

相對應的保護功能Retry Enable被勾起後，則當該保護發生後，會自動重啟，反之則需重新斷電。

Set the number of retries：為當錯誤產生時，馬達重啟的次數，一旦超過此值，便需要斷電重啟。

Set retry delay time：為發生錯誤後到重新啟動中間的延時時間。

- 配置文件：Ocp.h <Configuration Wizard>

- 硬體過電流保護設置

- Set AOCPCONT

參數	數值範圍/選項	說明	預設值
I_SHORT	0.15V/0.2V/0.25V/0.3V/0.35V/0.4V/0.45V/0.5V	過電流參考電壓	0.15V
AOCPEN	Enable/Disable	類比過電流功能	Enable
DOCPEN	Enable/Disable	數位過電流功能	Disable

調適方向建議：

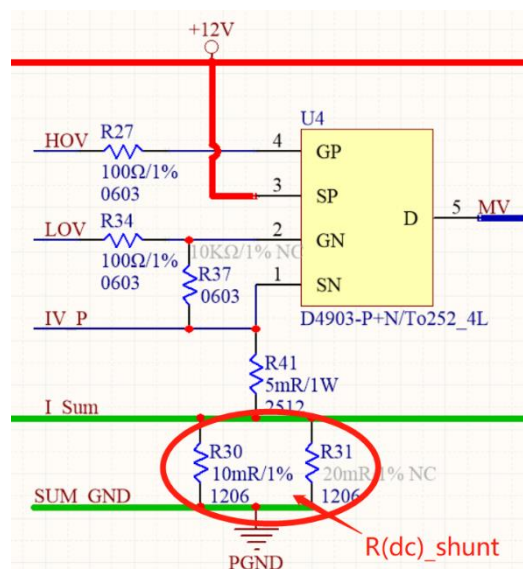
MDSF40提供8組等效過電流電壓檔位供配置，其母線電流保護點的計算為：

$$I_SHORT = I_{ocp} * R(dc_shunt)$$

AOCPEN為類比輸入式OCP，DOCPEN為數位輸入式OCP，若外掛Driver為IPM，一般皆設定DOCPEN，反之為AOCPEN。

OCP setting :

過電流值設定約在馬達最大電流(I_Peak)之1.5倍~2倍左右。



2. 低壓吊扇參數調適流程

2.1 啟動參數調適

Step1. 轉子對位(Parking)

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

參數	數值範圍/選項	說明	預設值	單位
Set DCR_V max values	0~100	DCR最終電壓	15	%
Set DCR_V min values	0~100	DCR初始電壓	5	%
Set DCR time	0~10000	DCR整體時間	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 (✓)

參數	數值範圍/選項	說明	預設值	單位
Set Startup_duty	0~32767	啟動Duty量	256	--
Set Square Parking Time	0~32767	定位時間	500	ms

調適方向建議：

當馬達啟動，會有特定位置容易造成啟動失敗現象時，可開啟DCR Start Strategy或Square Wave Parking Start Strategy二選一。在DCR策略中，根據負載扇葉不同的尺寸大小，合理調適DCR_V值，一般而言越重的負載所需的DCR_V值也越高，而DCR_time的時間越長，定位擺動的時間也相對較長，這裡不建議設定太長的時間，以免啟動時間過久。其參數可依照試誤法調適出適合馬達的啟動參數。

DCR Zone1 percentage of time 和 DCR Zone2 percentage of time兩者參數加起來需為100，一般Zone1設定上會大於Zone2。

Square Wave Parking Start Strategy：此策略暫時不建議使用。Startup duty為設定duty量，此值建議一開始先不要設定太大，避免啟動電流過大；Square Parking Time為設定整體定位時間，可依照定位情況作加減。

Step2. 啟動

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

參數	數值範圍/選項	說明	預設值	單位
Set IQ parking duration	1-32767	IQ定位持續時間	1	ms

- Set FOC LOOP Parameter

- IQ

- ✓ Set IQ Current parameter

參數	數值範圍/選項	說明	預設值	單位
Set IQ Initial current	--	IQ初始電流量	4000	mA
Set IQ Starting current	--	IQ啟動電流量	3000	mA

調適方向建議：

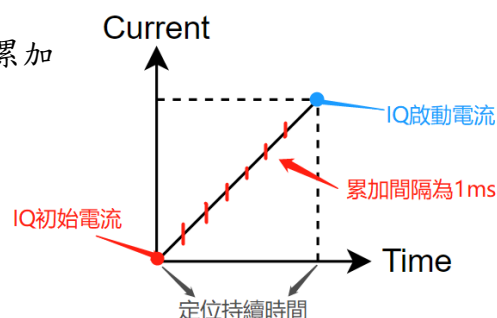
由於吊扇轉子慣量大，在啟動期間的啟動電流/力矩需大一點，依照負載大小適度調整，如此可有效完成啟動、降低震動及倒轉機率。時間適度增加可改善啟動失敗率。

對位電流的加速斜率及時間說明如下：

如右圖，定位持續時間為初始電流到啟動電流的總時間，而累加間隔時間為Timer0時間(1ms)，故每1ms的累加電流量為：

$(\text{IQ啟動電流} - \text{IQ初始電流}) / \text{定位持續時間}$

定位持續時間最小為1ms，若不需要定位可將初始電流=啟動電流



Step3. 位置開迴路運轉(Open loop)

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

參數	數值範圍/選項	說明	預設值	單位
FOC_Control_Stage	Standby/ OpenLoop/ CloseLoop	FOC控制階段	CloseLoop	--
Set SMO_RAMP acceleration slope	1~32767	PLL累積斜率	28	ms
Set PLL accumulation	1~100	PLL累加量	1	--
Set SMO_PLL initial speed	--	PLL初始轉速	5	rpm
Set SMO_PLL end speed	--	PLL結束轉速	35	rpm

調適方向建議：

FOC Control Stage :

Standby：此階段不執行任何FOC功能，可用來初步校正IA及IB參數。

OpenLoop：開迴路期間為電流閉環、位置開環。調適初期針對開迴路調適時，建議可先將FOC_Control_Stage調為OpenLoop，如此一來馬達便只會運轉在開迴路狀態，當馬達運轉中若出現不可逆的現象時，也可利用FOC_Control_Stage加以分析是開迴路出現的問題，還是閉迴路的問題。

CloseLoop：可選擇速度外環，或是功率外環。

PLL：

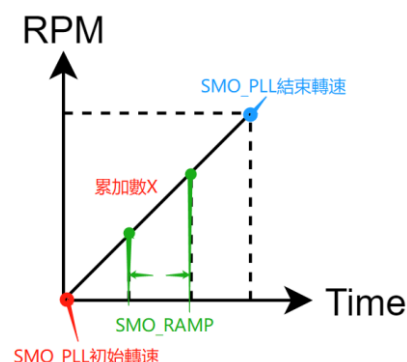
SMO_RAMP減少可有效提高位置觀測器收斂速度與減少開迴路啟動時間，但若太小有可能會導致啟動太快而失敗，需注意此變數值也包含順/逆風下的PLL加速斜率。開迴路加速度斜率及時間說明如下：

如右圖，為SMO_PLL初始轉速到SMO_PLL結束轉速斜率，
每SMO_RAMP的時間去累加一個PLL量(為圖中的累加數X)

故PLL開環的總時間計算為：

$$[(\text{SMO_PLL結束轉速} - \text{SMO_PLL初始轉速}) / (\text{PLL累加量} / \text{POLE PAIRS})] * \text{SMO_RAMP}$$

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



Step4. 開迴路銜接至閉迴路

■ 配置文件：Motor.h <Configuration Wizard>

■ Set the motor tuning process

參數	數值範圍/選項	說明	預設值	單位
Set SMO_DELAY Delay time	1-32767	開環進閉環延遲時間	10	ms

SMO_DELAY為內環準備要進閉環的延遲時間，可適度調整此參數。參數預留，暫不使用。

■ Set FOC LOOP Parameter

● IQ>

✓ Set IQ Current parameter

參數	數值範圍/選項	說明	預設值	單位
Set IQ End current	--	閉環電流初始值	2500	mA

調適方向建議：

在開環路結束後的電流即為IQ結束電流，此值在進入閉環後，會直接填入閉環的積分值以及輸出值，當位置開迴路(電流閉迴路)切入閉迴路時，由於初始外環PI控制器的積分器為零，為了避免銜接過程中轉速異常漣波，需提供外環PI控制器的積分器一個初始值。

Step5. 閉迴路運轉(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(✓)

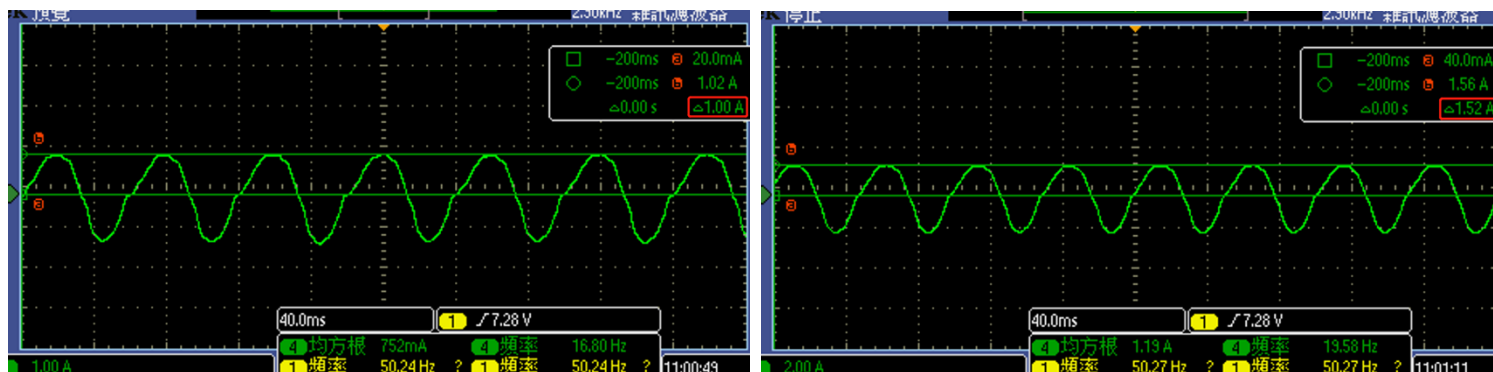
閉環控制共有三種控制模式，分別為**相電流控制(Phase Current control)**、**速度控制(Speed control)**、**功率控制(Power control)**，以下針對這三種模式分別做說明。

相電流控制(Phase Current control)

參數	數值範圍/選項	說明	預設值	單位
Set the rated output phase current (max)	0~30000	額定輸出相電流(max)	4000	mA
Set PHASE CURRENT_RAMP slope	1~32767	相電流控制周期	20	ms
Set PHASE CURRENT_CMD accelerate RAMP	0~32767	相電流累加量	5	--
Set PHASE CURRENT_CMD decelerate RAMP	0~32767	相電流遞減量	10	--

調適方向建議：

根據馬達的最大額定相電流填入設定值，這裡所控制的電流為相電流峰值，例如下圖所示：



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

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

電流控制斜率可依據 **CURRENT_RAMP**、**CURRENT_CMD accelerate RAMP** 以及 **CURRENT_CMD decelerate RAMP** 做調整，方向為每 **CURRENT_RAMP** 的時間，去累加/減 **CURRENT_CMD** 的量。

累加減相電流量 和 實際累加/減相電流的換算公式為：

$$\text{實際累加/減相電流} = (\text{相電流累加/減量}) / [(R_SHUNT * OPA_GAIN) * (1023/5) * 64]$$

速度控制(Speed control)

參數	數值範圍/選項	說明	預設值	單位
Set the rated output speed (max)	0~(32767/ Motor Pole/2)	額定速度(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	速度遞減量	5	--

調適方向建議：

根據馬達的最高額定轉速填入設定值，建議可預留一些餘量。馬達的轉速最高命令控制可利用 Set the rated output speed (max) 來設定。

速度控制斜率可依據 SPEED_RAMP、SPEED_CMD accelerate RAMP 以及 SPEED_CMD decelerate RAMP 做調整，方向為每 SPEED_RAMP 的時間，去累加/減 SPEED_CMD 的量，這裡的加減量除上 Pole pairs 即為 rpm。

因內部角速度輸出最大值為32767，所以再填入額定轉速的最大值的上限為32767/極對數，假設馬達的額定轉速超過此上限的話，請至Pwm.h <Configuration Wizard>，將Set Base Angular Velocity此參數填入 = 額定最大轉速 * Pole Pairs即可。

功率控制(Power control)

參數	數值範圍/選項	說明	預設值	單位
Set the rated output power (max)	0~10000	額定功率(max)	5100	0.01W
Set power magnification parameters	--	功率校正值	1840	--
Set I_BUS A/D Channel	CH0-CH7	採樣Ibus A/D通道	CH2	--

■ Set Ibus Limit

參數	數值範圍/選項	說明	預設值	單位
Ibus limit values	0~100	限制母線電流值	35	0.1A
Ibus OPA GAIN	0~100	內部OPA增益	11	--
Ibus Correction factor	0~1000	母線電流校正值	140	--

調適方向建議：

根據馬達的額定功率填入設定值。馬達的功率命令控制可利用Set the rated output power (max) 來設定。

如果實際功率值和填入的功率輸出有落差的話，可利用功率校正值做微調，微調方向為：

- 如果目標功率比實際功率還低 >> 增加功率校正值
- 如果目標功率比實際功率還高 >> 減少功率校正值

母線電流限制：

會需要設定此項目是在寬電壓條件下才需要額外做設定，假設工作電壓為8v~20v，額定功率需要控制在51W，則在8v和20v條件下的51W電流落差很大(8v/6.375A；20v/2.55A)，前提是馬達在低壓的情況下還能輸出這麼高的電流的話，此時會另外增加限制母線電流的功能。

和功率校正一樣，若限制母線電流和實際有落差，則微調母線電流校正值，其校正手法可以先把限制母線電流調低，在確定已經觸發限制電流的功能條件下，再去做電流校正微調，這裡要注意一件事，無論是做功率微調還是電流微調，writer需使用隔離型的，或是燒錄後暫時移除writer，接著再做校正觀測。

2.2 周邊控制功能

遙控器控制

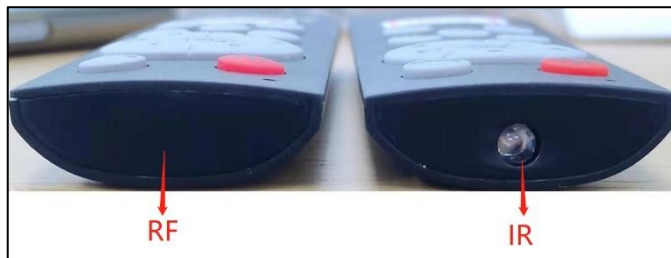
■ 配置文件： Motor.h <Configuration Wizard>

- Set motor control program
 - ✓ Set Peripheral Control
 - Remote control commands Enable/Disable(✓)

參數	數值範圍/選項	說明	預設值	單位
Remote control type	RF/IR	遙控器類別	IR	--

調適方向建議：

根據遙控器類別做選擇，若無法辨別遙控器是RF或是IR的話，可觀看遙控器頭部發射端是否有顆類似LED的元件，如果有則為IR，反之為RF。(如右圖)



- Power Level

參數	數值範圍/選項	說明	預設值	單位
Set IRF Button Power1	0~6000	功率1	500	mW
Set IRF Button Power2	0~6000	功率2	1000	mW
...以此類推	0~6000	...以此類推	2000	mW

- Speed Level

參數	數值範圍/選項	說明	預設值	單位
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
...以此類推	0~ Set the rated output speed (max)	...以此類推	250	rpm

- ✓ Observe Speed_UI parameters Enable(✓)

參數	數值範圍/選項	說明	預設值	單位
Set IRF Button SPEED_UI1	0~32767	速度積分1	500	--
Set IRF Button SPEED_UI2	0~32767	速度積分2	700	--
...以此類推	0~32767	...以此類推	900	--

- Current Level

參數	數值範圍/選項	說明	預設值	單位
Set IRF Button Current1	0~10000	電流1	800	mA
Set IRF Button Current2	0~10000	電流2	1000	mA
...以此類推	0~10000	...以此類推	2000	mA

根據Set the main control loop外環控制模式設定Power Level 或 Speed Level 或 Current Level，每個Level皆對應遙控器按鈕上的數字功能，若不需要那麼多功能，則可在RF_Decode.h/IR_Decode.h將它關閉(後面遙控解碼會另外說明)，而Level不夠可自行增加，如下範例：

依照控制模式分別增加以下紅框內代碼(如下圖)：

■ 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 </>
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

```

以下根據IR控制或是RF控制分別增加紅框內代碼:

■ 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解碼

■ 配置文件：RF_Decode.h <Configuration Wizard>

● Customize

✓ Set RF_DEC_CTRL

參數	數值範圍/選項	說明	預設值	單位
RFIN_SEL	P0.5(IR1) / P3.3(IR2)	選擇接收碼輸入腳	P3.3(IR2)	--

✓ Set RF_DEC_SET

參數	數值範圍/選項	說明	預設值	單位
CLK_DIV_SEL	24MHz/16MHz/8MHz/ 6MHz/3MHz/2MHz	數據解碼時鐘頻率	2MHz	--
DIN_DB_SEL	0ns /250ns /500ns 1000ns	數據彈跳時間	500ns	--
DIN_TYP	Data Type1/Data Type2	數據類別	Data Type1	--
DOUT_REV	First input data is LSB/ First input data is MSB	解碼輸出數據反向	First input data is LSB	--
HEADER_EN	No HEADER / With HEADER	頭碼選擇	No HEADER	--

調適方向建議：

首先先確認RF DATA連接腳位，並在RFIN_SEL選擇該輸入腳位(如右圖)。

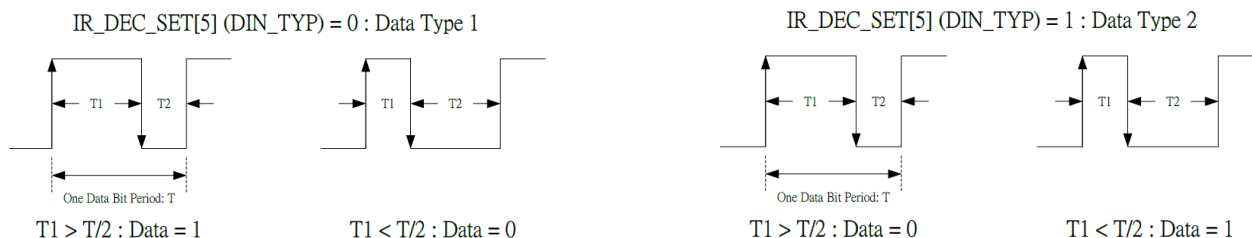
CLK_DIV_SEL設定：內部Counter上限為32767，再不超過此值的前提下設定該時鐘頻率，其公式如下：

$CLK_DIV_SEL \leq 32767 / (\text{數據時間寬度})$ 。例如某RF資料寬度最長為Stop(10ms)，則 $CLK_DIV_SEL \leq 32767 / 10\text{ms}$

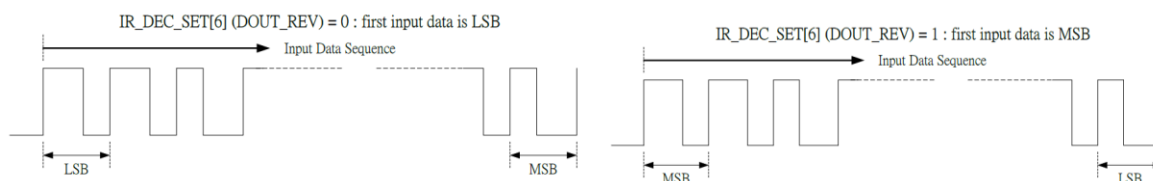
$>> CLK_DIV_SEL \leq 3.2\text{MHz}$

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

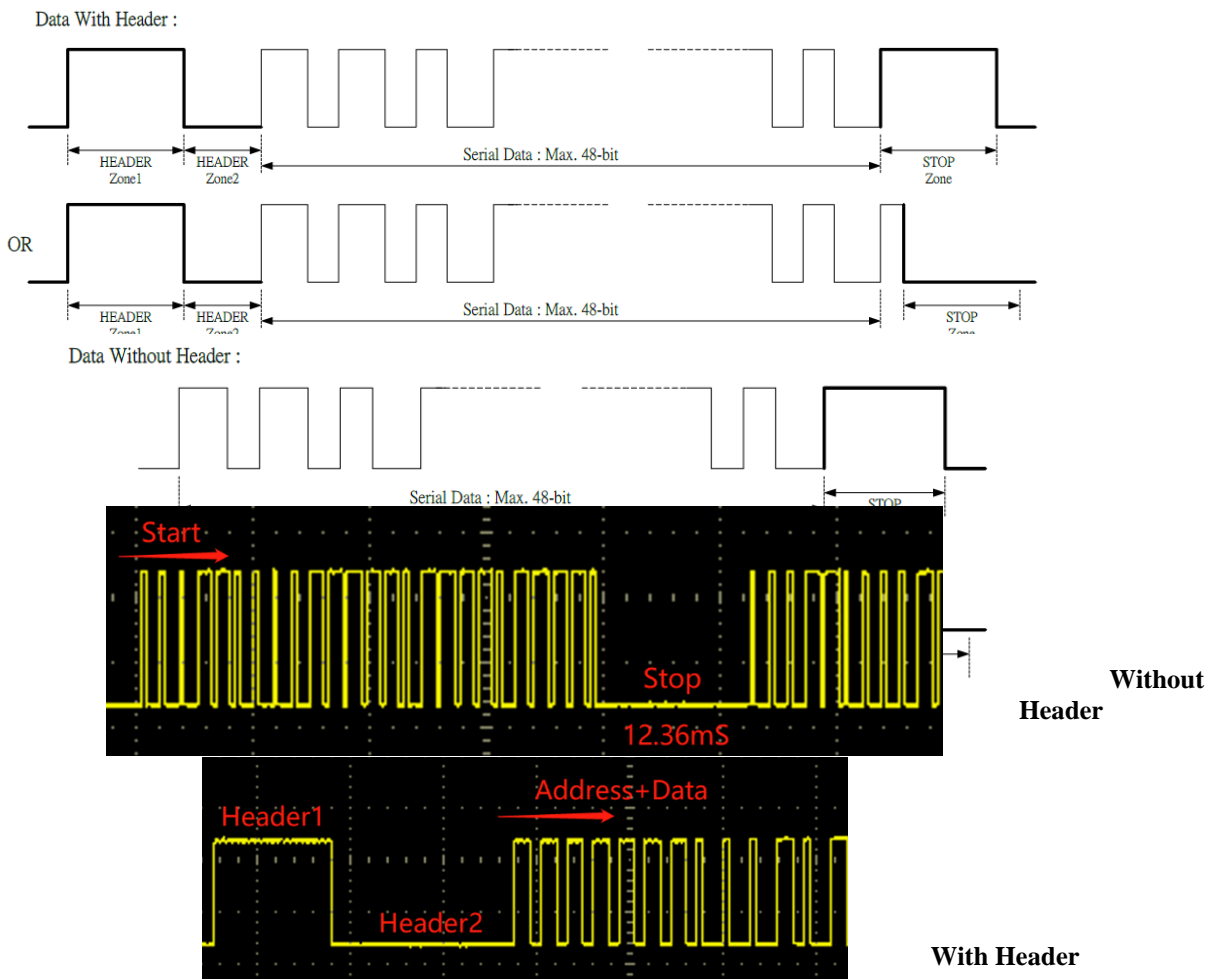
DIN_TYP 數據類別：資料0和1的判斷格式如下：



DOUT_REV 設定：定義第一筆Bit為MSB或是LSB。



HEADER_EN 設定：選擇需不需要頭碼判斷。這裡要注意，若選擇With Header，Header的Zone1和Zone2必需同時存在，無法設定0避免掉。



- Remote control code set
 - ✓ Head code set

參數	數值範圍/選項	說明	預設值	單位
Set Header Zone1 time width	0~32767	頭碼1(High)	5000	us
Set Header Zone2 time width	0~32767	頭碼2(Low)	7800	us

- ✓ Stop code set

參數	數值範圍/選項	說明	預設值	單位
Set Stop time width	0~32767	結束碼	5000	us

- ✓ RF decoding to UR Enable(✓)
 - ✓ Learning function (✓)

參數	數值範圍/選項	說明	預設值	單位
Learning data values set	0~65535	學習鍵數值	243	--
Set learn time	0~20	學習時間	10	sec
Judge the times of successful learning	0~20	判定學習成功次數	1	10times

參數	數值範圍/選項	說明	預設值	單位
RF address value set	0~65535	地址碼數值	54399	--

- ✓ RF data values set
 - Enable OFF Button(✓)

參數	數值範圍/選項	說明	預設值	單位
Button OFF	0~65535	Off鍵數值	13503	--

● Enable Speed1 Button(✓)

參數	數值範圍/選項	說明	預設值	單位
Button speed 1	0~65535	速度1鍵數值	15903	--

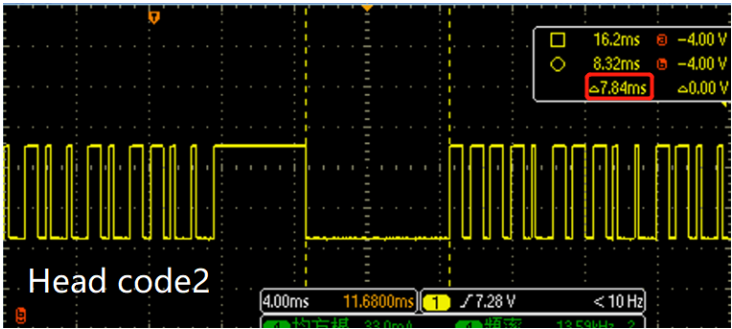
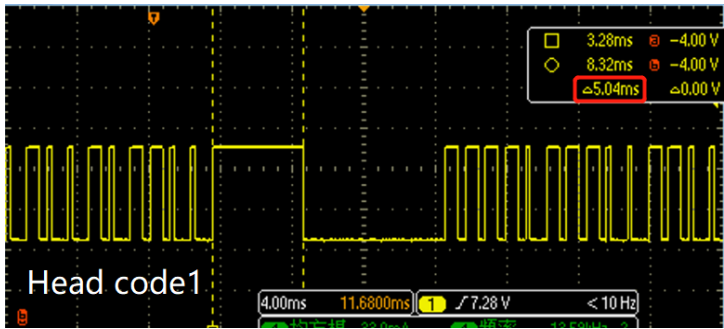
● Enable Speed2 Button(✓)

參數	數值範圍/選項	說明	預設值	單位
Button speed 2	0~65535	速度2鍵數值	13983	--

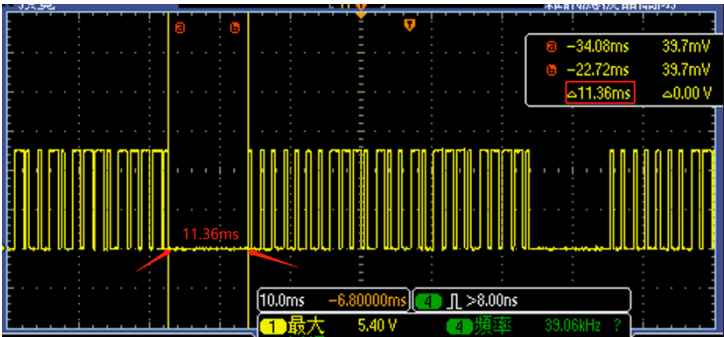
以此類推...

調適方向建議：

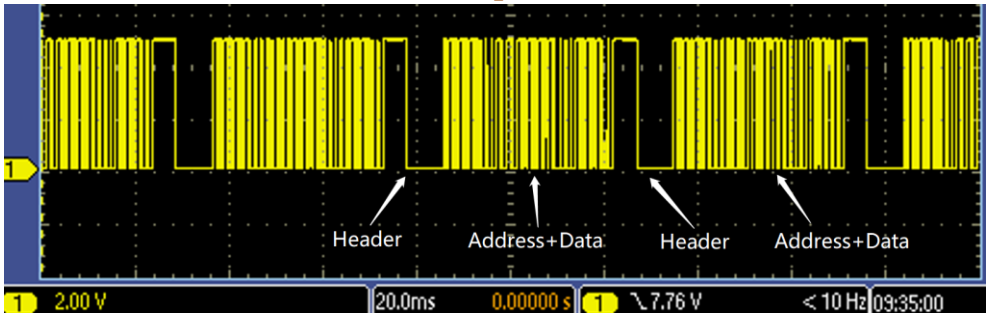
頭碼設定：請根據示波器上量測RF TX丟出來的波形，訊號開頭寬度較寬的部分為頭碼，High的寬度為**Header Zone1 time width**; Low的寬度為**Header Zone2 time width**，例圖如下：



結束碼設定：請根據示波器上量測RF TX丟出來的波形，訊號結尾寬度較寬的結束碼部分為**Stop time width**，例圖如下：



假設遇到沒有結束碼的遙控器(如下圖)，則可將**Stop time width**設定和**Header Zone1 time width**一樣。



地址/數據碼解碼：當頭碼和結束碼設定完成後，首先要先判斷該遙控器的地址碼以及數據碼長度，目前代碼得預設值為**16bits address + 16bits data**，若和預設值不同，需自行在代碼上做修改，位置如右圖(RF_Decode.c)，每個暫存器為8bits，請根據遙控器該bits數做位移。

地址碼/數據碼bits數判斷：在相同遙控器下，分別按下不同按鍵並且記錄，前幾個相同的bits為地址碼，後幾個不同bits為數據碼，例如：(以下0和1依照Page18 **DIN_TYP** 數據類別做轉換)

```

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(); //遙控器學習模式
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

藍色部分為地址碼，剩餘不同部份為數據碼，所以便是16bits address + 16bits data，雖然20bits address+ 12bits data在這例子也通用，但為了資料處理好撰寫，建議以8的倍數做區分。

Note：在確認地址碼和數據碼bits數時，建議記錄3個按鍵以上，避免誤判。

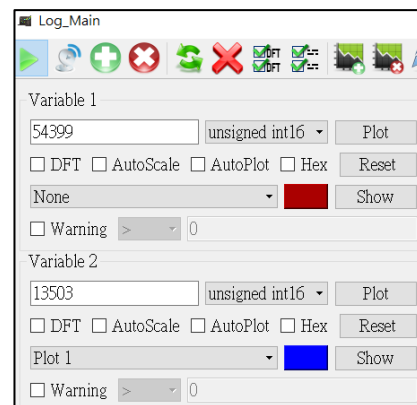
勾選**RF decoding to UR Enable**，並開啟Log_Main觀測解碼hex值，
(Note:做這動作前請記得至UartSystem.h <Configuration Wizard>將UART Enable勾選起來，否則UART功能不會作動)。

如右圖，按下任一按鍵觀察Variable1和Variable2數值，1的為**RF address value set**，2的為該按鍵功能的**data values**。

Note：若開啟**Learning function**，**RF address value set**可以不用填寫。

學習功能(對碼)：開啟**Learning function**，依照前面敘述的Uart觀測數值方式，將對碼數值填入**Learning data values set**，**Set learn time**為學習時間，從MCU work那一刻開始算起，過了該時間便不再學習；**Judge the times of successful learning**為判斷學習成功次數，每10次為一單位，意思為當MCU判斷到該學習按鍵x次數，認為學習成功便將該遙控器的地址碼，學習儲存至EEPROM。解碼完畢記得將**RF decoding to UR Enable**關閉。

Note：若需要開啟學習功能，請記得到Eeprom.h <Configuration Wizard>將EEPROM Enable勾選起來。



IR解碼

■ 配置文件： IR_Decode.h <Configuration Wizard>

● Customize

✓ Decoding with Capture(✓)

● IR decoding to UR Enable(✓)

參數	數值範圍/選項	說明	預設值	單位
Set the number of data to observe	1~50	觀測第幾個Bit數值	1	--

參數	數值範圍/選項	說明	預設值	單位
Set IR High Head counter_min	0~65535	頭碼最小值	11100	--
Set IR High Head counter_MAX	0~65535	頭碼最大值	15000	--
Set IR High Head counter_error	0~10000	頭碼誤差量	2000	--

✓ Set read IR data format

參數	數值範圍/選項	說明	預設值	單位
IR_data_format	Low_level/ High_level	IR數值判斷準位	High_level	--

參數	數值範圍/選項	說明	預設值	單位
Set IR 0 data counter	0~65535	數據0計數	1681	--
Set IR 1 data counter	0~65535	數據1計數	4955	--
Set IR data counter_error	0~10000	數據計數誤差量	400	--
Set the number of bits of IR data	0~16	數據碼bit數	16	Bits
Set the number of bits of the IR address	0~16	地址碼bit數	16	Bits
Set IR address value	0~32767	地址碼數值	7038	--

✓ IR data values set

● Enable OFF Button(✓)

參數	數值範圍/選項	說明	預設值	單位
Button OFF	0~65535	Off鍵數值	4335	--

● Enable Speed1 Button(✓)

參數	數值範圍/選項	說明	預設值	單位
Button speed 1	0~65535	速度1鍵數值	32895	--

● Enable Speed2 Button(✓)

參數	數值範圍/選項	說明	預設值	單位
Button speed 2	0~65535	速度2鍵數值	16575	--

以此類推...

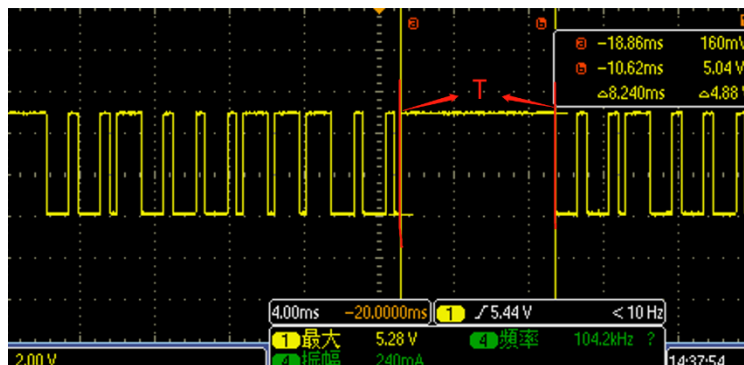
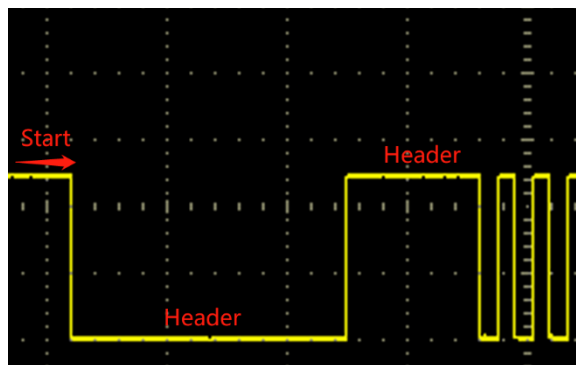
調適方向建議：

若以 Capture 作為解碼，請先至 **Capture.h <Configuration Wizard>** 並將 External Capture Enable/Disable 勾選起來，以及至 **Interrupt.h <Configuration Wizard>** 將 CAPTURE Interrupt 勾選起來，確認 Capture 使用的腳位並且設定在 CAPPINSEL，最後將 **Decoding with Capture** 勾選。

解頭碼：量測任一按鍵的訊號，並在示波器上記錄頭碼的時間寬度(如右下圖)，**因內部 Capture 只能採用正緣觸發，故以下的頭碼寬度都是以正半周訊號為主**，但若頭碼同時有負半周訊號和正半周訊號(如左下圖)，有兩種解法：1. 負半周訊號直接跳過，只判斷正半周訊號；2. 改採用 TIMER+INT 解碼。

IR High Head counter 數值為：量測到的時間寬度 * E_CAPCKS(從 Capture.h 設定)

例如右上圖量測到為 8.24ms，CAPCKS 設定為 48MHZ/16



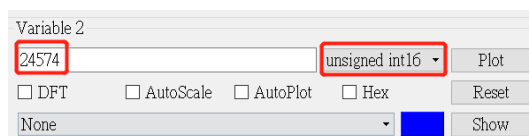
則 IR High Head counter = 8240us * 3MHz = 24720.

(這裡要注意一點，如果 Counter 出來的值過小(比如都是小於 1000)，那代表 E_CAPCKS 要再往上設定，一般建議頭碼的 Counter 應大於 10000 以上，如果頭碼的 Counter 過小的話，在後面數值的 Counter 判斷，很容易會有誤判的情況發生)

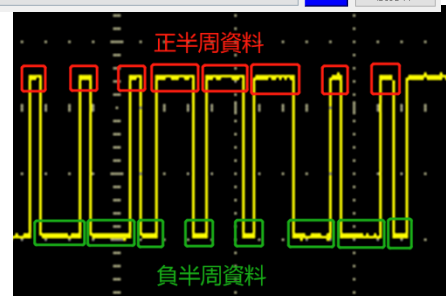
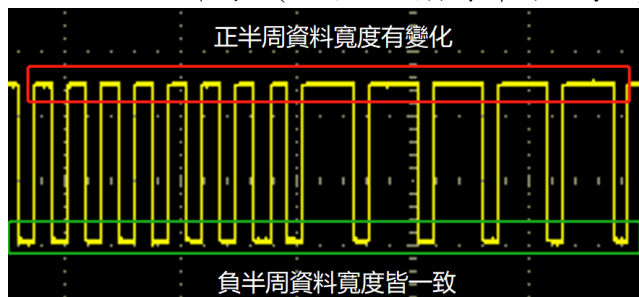
而 counter min/MAX 的差別在於，有些遙控器的頭碼寬度，每個按鍵都會有些微差異性，把量測到最小的寬度填入 min，最大填入 MAX，而如果每個按鍵的頭碼寬度差異性不大的話，兩者值填入一樣。

IR High Head counter_error 一般填入 IR High Head counter * 10% 的值(若非整數，可往上提升至整數)。例如 IR High Head counter = 24720; 24720*10% = 2472，則 IR High Head counter_error 填入 3000(多估抓一些誤差值)

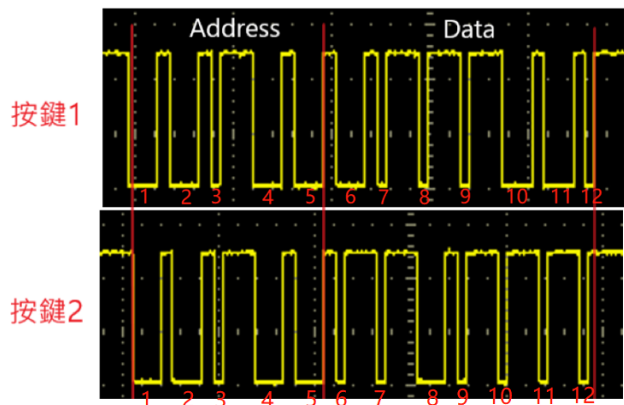
將 **IR decoding to UR Enable** 開啟，當按下任一按鍵，若 Variable1 有數值的跳動，且和填入的 IR High Head counter 差不多時，代表頭碼有正確解碼到。



設定 IR_data_format: 此設定為抓取資料(包含地址)的方式，Low_level 為抓取負半周資料，High_level 為抓取正半周資料(如右圖)當出現如下圖的資料碼的話，建議設定為 High_Level，或是改用 TIMER+INT 解碼。(已可以明顯判斷 0 和 1 為主)



量測數據波形確認地址 bit 數以及數據 bit 數(相同遙控器，不同按鍵下，一樣的碼為地址碼，不一樣的碼為數據碼)，範例如下：



左圖為兩個按鍵的地址+數據碼

可以觀測出,從第6個bit開始便不一樣(IR_data_format 設定Low_Level), 所以Address bit數為5, Data bit數為7
將7填入Set the number of bits of IR data

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

(建議在區分判斷Address和Data時, 請分別觀測3個按鍵以上, 避免只判斷2個按鍵容易會有誤區)

設定data0和1的定義, 一般會將時間寬度短的定義為0; 長的定義為1。

如上圖, Address即為11011(設定為Low_Level, 只看負半周寬度), 11011為十進制” 27” , 將27填入Set IR address value.

Set IR 0 data counter、Set IR 1 data counter : 判斷Data0/1的Counter數, 開啟IR decoding to UR Enable, Set the number of data to observe此參數需搭配UR中的Variable2數值, 以上圖為例: 比如說我已知第1個bit為1; 第3個bit為0, 首先判斷Data=1的Counter, 則在Set the number of data to observe填入1, 並且按下該按鍵觀測Variable2數值, 將其數值填入Set IR 1 data counter, 反之Data=0的Counter則是在Set the number of data to observe填入3, 觀測Variable2數值, 將其數值填入Set IR 0 data counter。(注意: 判斷順序必需由第1個bit開始, 並且填入相對應的Counter, 再由左至右找尋另一個data的位置, 如果bit1=0的話, 再找bit=1的位置, 反之亦然, 但如果一開始直接跳bit的話, 程式內部可能會直接跳Error)。當設定完成後, 觀測UR的Variable3數值, 如果此數值和IR_address_value一樣的話, 代表到目前為止解碼OK。

IR data values set : 最後將UR的Variable4填入相對應按鍵名稱內, 例如我按下Off功能按鍵, 且Variable4為198, 則在Button OFF填入198, 以此類推...

VSP控制

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

● Set motor control program

✓ Set Peripheral Control

● VSP control commands Enable/Disable (✓)

參數	數值範圍/選項	說明	預設值	單位
Set VSP_CH	CH0~CH7	VSP A/D	CH4	--

■ Slope control

參數	數值範圍/選項	說明	預設值	單位
Set VSP start voltage	0~50	VSP啟動電壓	5	0.1V
Set VSP max voltage	0~50	VSP最大電壓	45	0.1V
Set VSP stop voltage	0~50	VSP停止電壓	3	0.1V

■ Stage control

參數	數值範圍/選項	說明	預設值	單位
Set VSP SPEED1 speed	--	VSP第一段速度	100	rpm
Set VSP SPEED2 speed	--	VSP第二段速度	150	rpm

以此類推...

◆ **Observe Speed_UI parameters Enable(✓)**

參數	數值範圍/選項	說明	預設值	單位
Set VSP_SPEED_UI_1	--	VSP第一段速度	350	--
Set VSP_SPEED_UI_2	--	VSP第二段速度	800	--

以此類推...

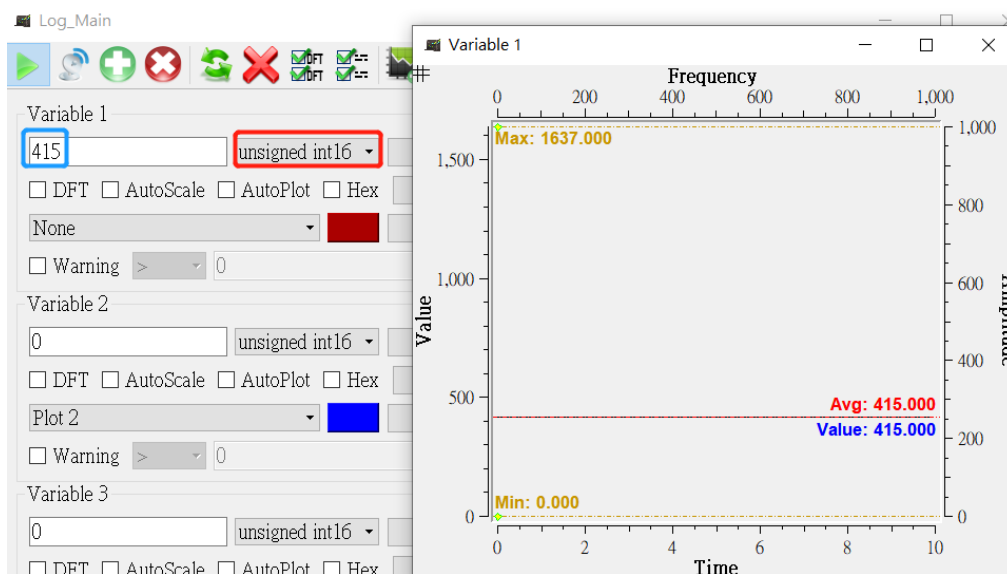
調適方向建議：

VSP外部控制功能，首先確認VSP的A/D Channel，並填入VSP_CH。VSP共有兩種控制模式，分別為Slope control、Stage control，目前VSP只供測試用，Stage control目前只開放控制速度功能，後續若有案子需求，再額外更新。

補充：速度 UI

在速度外環下，無論是遙控器控制或是VSP控制，都會有個UI設定參數可做使用，此值的設定是為了加/減轉速快速反應，其調適說明如下：

將所需的各段轉速值填入之後，開啟 **Observe Speed_UI parameters Enable**，在吊扇各段位速度底下觀察 **Variable1** 參數，如下圖所示：



假設按下速度1按鈕(SPEED1 speed)，當吊扇速度穩態後，此時Variable1值為416，則將416填入 **Button1_SPEED_UI**，SPEED2_speed對應Button2_SPEED_UI，以此類推...。若實際測出來有過衝的現象，可向下微調此參數。需注意此功能只限定轉速控制下設定。

Speed_UI調適完畢記得將**Observe Speed_UI parameters Enable** 關閉

2.3 順/逆風啟動參數調適

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

參數	數值範圍/選項	說明	預設值	單位
BEMF Fairwind/Headwind adjustment process	LEVEL_1/LEVEL2	順逆風判斷步驟	LEVEL_2	--

✓ Set BEMF A/D Channel

參數	數值範圍/選項	說明	預設值	單位
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

參數	數值範圍/選項	說明	預設值	單位
Set the minimum connection speed for judging Fairwind	--	判斷順風最小轉速	23	rpm
Set the maximum connection speed for judging Fairwind	--	判斷順風最大轉速	180	rpm
Set the maximum connection speed to judge the headwind	--	判斷逆風最大轉速	100	rpm
Set the duration of Fairwind start judgment	--	判斷順逆風週期	100	ms
Set Headwind_high_speed_start_Strategy	PLL/ABS	高速逆風啟動策略	ABS	--

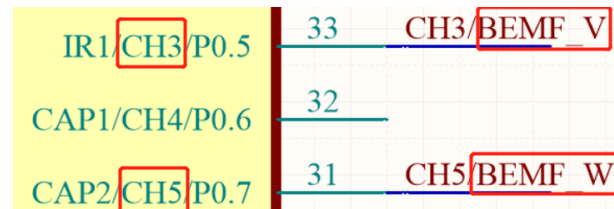
調適方向建議：

先確認兩相的分壓電阻連接的AD Channel，分別填入 Set BEMF_V_CH、Set BEMF_W_CH。順逆風調適初期，可先將BEMF調機流程選擇LEVEL_1，此時可在馬達靜止狀態，用手撥扇葉使馬達旋轉，測試順逆風啟動的狀態。

判斷銜接順風轉速範圍：銜接順風的條件機制，在最小轉速到最大轉速之間，皆可銜接至順風，可根據實際吊扇銜接狀況做微調，會限制最小轉速是因為，當轉速過小時，會無法判斷到BEMF，所以無法得知馬達位置，故限制一個下限值。

判斷銜接逆風轉速：在此設定的轉速值以下(Set the maximum connection speed to judge the headwind)，皆可滿足銜接逆風機制的條件，若大於此轉速則Free Run，直到馬達低於此轉速，可依照實際馬達狀況做微調。

Headwind_high_speed_start_Strategy，逆風銜接策略，分別有PLL策略以及ABS策略，目前建議使用ABS策略即可。



參數	數值範圍/選項	說明	預設值	單位
Set Fairwind/Headwind angle(CW_CCW=0)	0~383	CW_CCW=0 順逆風初始角度	0	--
Set Fairwind/Headwind angle(CW_CCW=1)	0~383	CW_CCW=1 順逆風初始角度	64	--
Set IQ Fairwind/Headwind starting current(IQ_CMD)	--	順逆風啟動Iq電流	2000	mA
Set ID Fairwind/Headwind starting current(ID_CMD)	--	順逆風啟動Id電流	0	mA
Set the initial value of Fairwind/Headwind current(IQ_UI)	0~32767	順逆風啟動初始內環輸出量	8000	--
Set Headwind to Fairwind target speed	--	逆風銜接進入外環速度	45	rpm
Set the judgment counter for the static BEMF value of the motor	0~10000	判斷馬達靜止狀態延遲時間	300	ms

✓ Observe BEMF_V_W Parameters (✓)

調適方向建議：

順逆風銜接判斷角度設定：此參數(Fairwind/Headwind angle(CW_CCW=0/1))建議不做更改。

順逆風銜接IQ設定：根據銜接進順/逆風的狀況做微調，若銜接後馬達會有很大的撞擊聲，則可將銜接IQ電流往下做微調的動作。

順逆風銜接IQ_UI初始值設定：設定此參數是讓銜接進順逆風時，IQ_OUT有個初始值在，不會從0開始。

Headwind to Fairwind target speed：逆風啟動後，轉正轉銜接close loop的轉速值，此值為決定在逆風啟動下，銜接至close loop的轉速值，太小可能會造成銜接上的失敗。

Set the judgment counter for the static BEMF value of the motor：判斷馬達靜止時間，因馬達速度過低時，無法依靠BEMF來判斷馬達目前處於低速運轉還是靜止不動，故當BEMF_V-W低於某值以下(代碼預設為5，可至Motor.c 第765行做修改)，直接判定為馬達靜止，而設定的判斷時間便是避免雜訊產生誤判的延遲時間。

2.4 保護參數調適

- 配置文件：**Motor.h** <Configuration Wizard>
- **Set motor protection function**
- **過/欠電壓保護**
 - 參數配置如1.3節(保護參數配置)所示。

調適驗證方向：

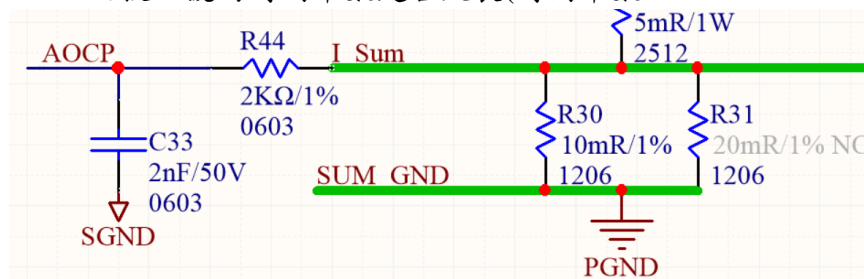
1. 依據客戶提供的過、欠電壓與恢復電壓條件進行設定。
2. 過電壓及恢復測試：啟動吊扇後，將電源供應器的電壓升至過電壓準位，確認吊扇是否進入停機保護，接著將電源供應器的電壓降至過電壓恢復準位，判斷吊扇是否正常啟動。
3. 欠電壓及恢復測試：啟動吊扇後，將電源供應器的電壓降至欠電壓準位，確認吊扇是否進入停機保護，接著將電源供應器的電壓升至欠電壓恢復準位，判斷吊扇是否正常啟動。

- **過電流保護-測試硬體(重點測試)**

- 參數配置如1.3節(保護參數配置)所示。

調適驗證方向：

1. 驗證硬體，先將AOCP此點和5V短路，馬達可不接，利用UART監看是否有跳OCP，確認沒問題再做以下短路測試。
2. 啟動吊扇後，任意將兩相(UV or UW or VW)輸出進行短路，以判斷吊扇是否進入停機保護，接著重新上電，確認吊扇是否能正常工作，並多次重複該步驟，以確保MOSFET能充分保護。
3. 若短路實驗後吊扇無法重啟，則檢查MOSFET的規格是否滿足、AOCP回授訊號在布線路徑上是否有干擾、AOCP回授訊號的時間常數是否太長(時間常數： $RC \leq 2k * 2nf = 4us$)。



- **堵轉保護**

- 參數配置如1.3節(保護參數配置)所示。

調適方向建議：

可利用手拉住扇葉，確認吊扇進入失步/高頻狀態，堵轉保護是否正常作動。

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	轉子方向設定	CW	--
Set Stop_Fun stop speed	--	PWM停止輸出轉速	60	rpm

調適方向建議：

設定CW/CCW轉向依據所需的扇葉方向做修改。

Stop_Fun 停止轉速在這裡指的是，當遙控器按下停止鍵時，馬達轉速低於此參數時，PWM才會停止輸出，大於此參數的轉速值時，PWM皆還在輸出當中，這是為了順風銜接順暢而設定的。其值可設定在低速判斷不到順風轉速的銜接點。

✓ 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	--	換方向啟動後運轉時間	10000	ms
Stop this time before catching the wind	--	停止此時間後再接順風	2000	ms
Operating time after connecting downwind	--	銜接順風後運轉時間	5000	ms

調適方向建議：

此功能為測試用。其測試流程為：馬達啟動運轉(初始依照Set CW/CCW steering設定方向)，運轉時間為**Running time after changing direction**，接著馬達停止運轉，時間為**Stop this time before catching the wind**，接著銜接順風啟動，運轉時間為**Operating time after connecting downwind**，最後換方向啟動，時間為**Running time after changing direction**，不斷循環...

✓ IPD Enable/Disable(✓)

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

調適方向建議：

此IPD功能需搭配配置文件 **IPD.h <Configuration Wizard>**做調適，**Automatic cycle test of initial position**此功能可自動在馬達靜止狀態，不斷循環打IPD，用來做IPD功能準確性，可將IPD Pattern此參數拉到UART觀測，在馬達轉子不同角度下，固定位置的Pattern是否皆一樣。

- 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	0.15V	--
AOC PEN	Disable/Enable	類比OCP功能	Enable	--
DOC PEN	Disable/Enable	數位OCP功能	Disable	--
IPD Path Select	IPD Current Compare from AOC Path IPD Current Compare from OPA Path	IPD OCP判斷路徑	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	24MHz	--

調適方向建議：

IPD所產生的聲音大小，和I_SHORT成正比，若設定至0.15V時聲音還是很大，只能增加Shunt R的阻值(需注意電阻的瓦數)。

AOCPEN、DOCPEN、IPD Path Select：不建議更動。**IAECYC**：此參數為IPD重要參數，用來計數OCP dt數值，轉子在不同位置下所執行的IPD，取最小Counter為該位置的Pattern，其Counter最大值為65535，可依照量測的OCP dt寬度，去得知所需要的IAECYC，不同的IAECYC可判定到的最大dt寬度，計算如下：

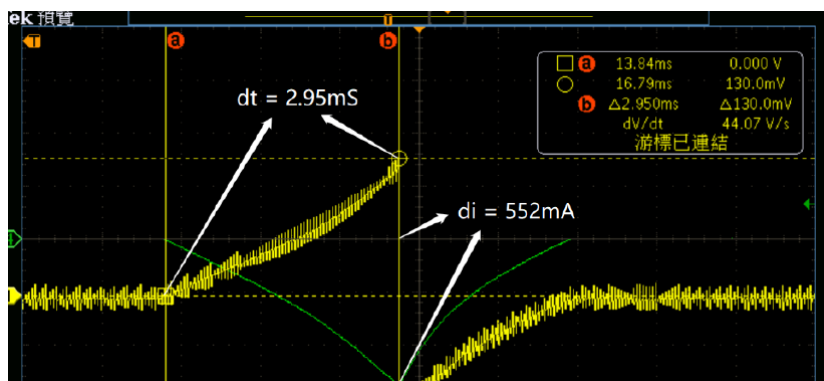
$$48\text{MHz} : (1/48\text{MHz}) * 65535 \approx 1.3\text{ms}$$

$$24\text{MHz} : (1/24\text{MHz}) * 65535 \approx 2.6\text{ms}$$

$$12\text{MHz} : (1/12\text{MHz}) * 65535 \approx 5.2\text{ms}$$

$$6\text{MHz} : (1/6\text{MHz}) * 65535 \approx 10.4\text{ms}$$

以右圖為例，黃色波形為OCP訊號，dt為2.95ms，假設此時IAECYC選擇48MHz或24MHz，因為dt皆大於Counter的最大值產生溢位，你會發現馬達在啟動時馬達依然會倒轉，所以只dt只適合選擇12MHz。



✓ **Brake control Enable/Disable(✓)**

參數	數值範圍/選項	說明	預設值	單位
Set braking force	0~100	剎車Duty量	99	%

調適方向建議：

剎車力道，順逆風Diode架構或是內部估測器用，暫時沒用到。

■ 配置文件：**Ipwm.h <Configuration Wizard>**

■ **IPWM Enable/Disable(✓)**

參數	數值範圍/選項	說明	預設值	單位
IPWM Frequency SET	--	IPWM頻率	4000	Hz
IPWM MODE SET	Force Low/Force High/Active Low/Active High	IPWM模式	Active High	--

● **Set buzzer working time**

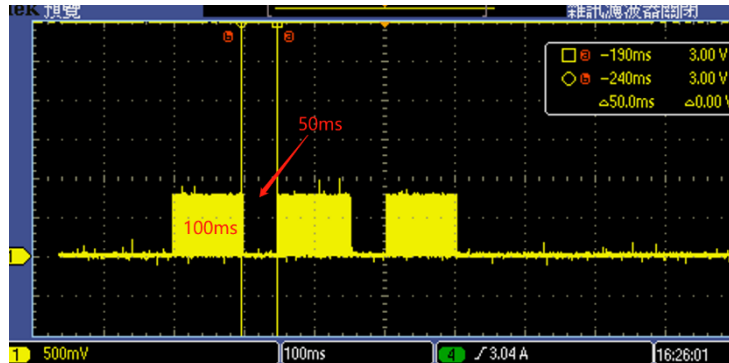
參數	數值範圍/選項	說明	預設值	單位
IPWM Times_Init	1~255	IPWM執行次數	1	times
IPWM Enable time_Init	1~10000	IPWM執行時間	100	ms
IPWM Disable time_Init	1~10000	IPWM間隔時間	50	ms

以此類推...

調適方向建議：

此功能需搭配Buzzer做使用，並且接到MCU內部IPWM腳位。將Buzzer工作頻率填入IPWM Frequency。

IPWM MODE：若Buzzer為外激式(無源)的，建議設定為Active High，若為自激式(有源)的Buzzer，建議設定為Force High。底下的Buzzer working區分為開機設定、關機設定、錯誤設定，可依照客戶所需要的buzzer鳴叫的次數、時間長短自行做調適，其中IPWM Times為次數，Enable time為鳴叫時間，Disable time為關閉時間，例如Times=3; Enable time = 100ms ; Disable time = 50ms，則IPWM波形如下：



3. 程序架構

3.1 程式流程

■ 配置文件：main.c

■ Main Function

- 系統初始化設定
- while(1){
 - ✓ RFDecode_Read(); // RF 解碼讀取
 - ✓ IRDecode_Read(); // IR 解碼讀取
 - ✓ EEP_Storage(); // EEPROM 儲存
 - ✓ 怠速啟動電壓零交越點判斷與轉速估測
 - ✓ UART丟數據
- }

■ 配置文件：Interrupt.c

■ Timer 0 ISR(Time Based : 1ms)

- Correction_Current_AD_offset(); //功率控制-母線電流A/D初始校正
- Motor_Control(); //馬達的錯誤判斷、正反轉控制、馬達啟動
- Buzzer_Fun(); //Buzzer.

■ Timer 1 ISR(Time Based : 10ms)

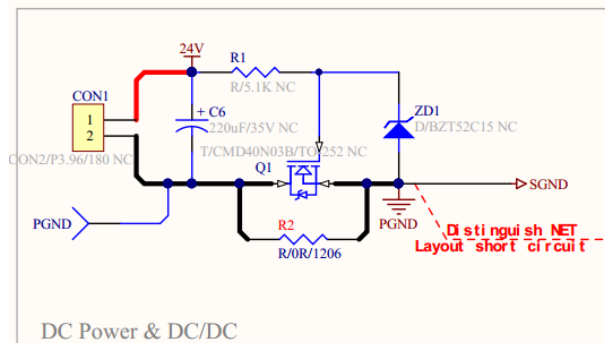
- Vsp_Fun(); //外部VSP命令接收
- FaultLock_Fun (); //系統堵轉、過/低速保護
- Vbus_Protect_Fun (Vbus_avg); //系統母線電壓偵測、保護判斷
- Temperture_Protect_Fun (Temperture_avg); //系統過溫保護
- Phase_OCP_Protect_Fun() //系統相電流保護
- LACK_OF_PHASE_Function(); //馬達靜止缺相判斷
- MotorLackPhase_Run_Fun(); //馬達運轉缺相判斷

■ PWM MAX ISR(Time Based : Fpwm)

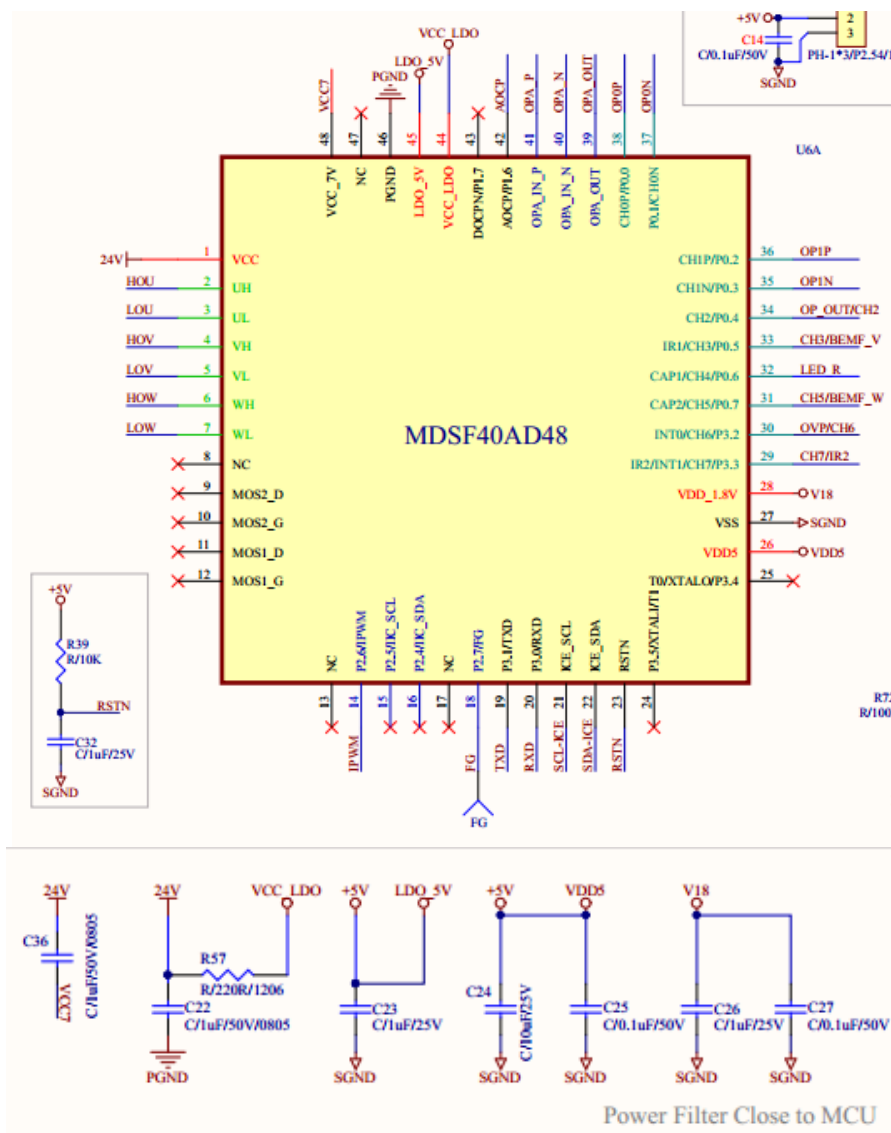
- 怠速啟動電壓零交越點判斷與轉速估測，旗標開啟

4. 參考電路設計

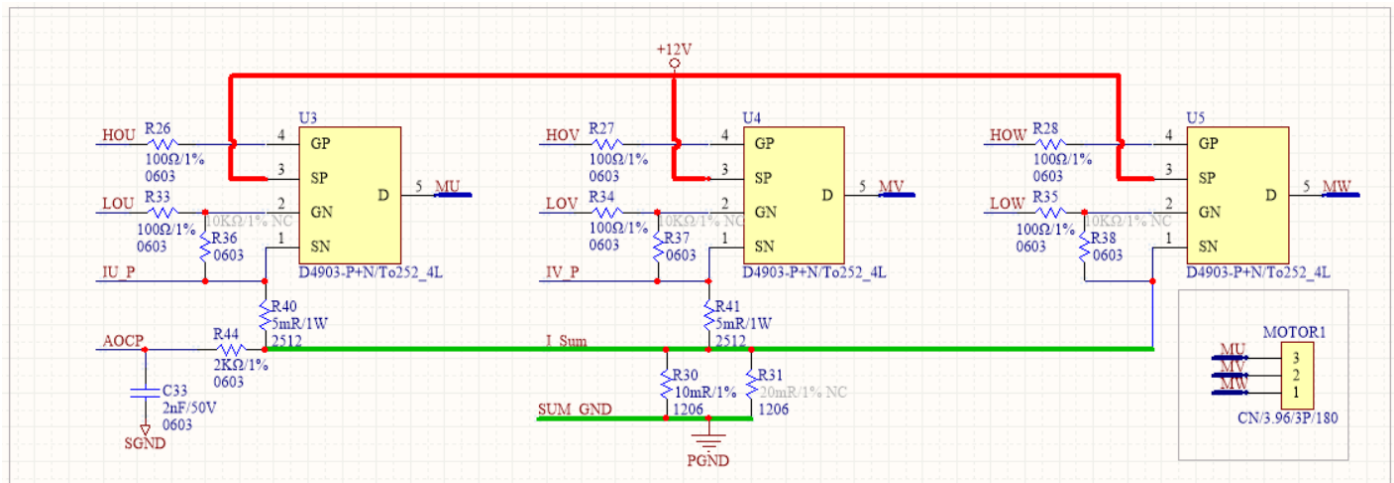
4.1 電源輸入電路



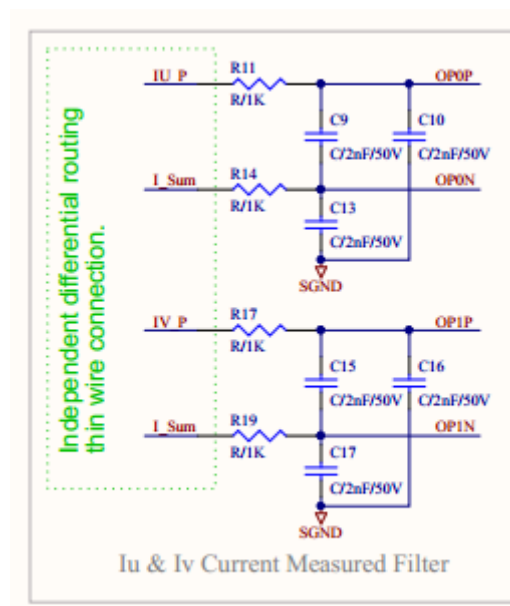
4.2 核心單元及周邊電路



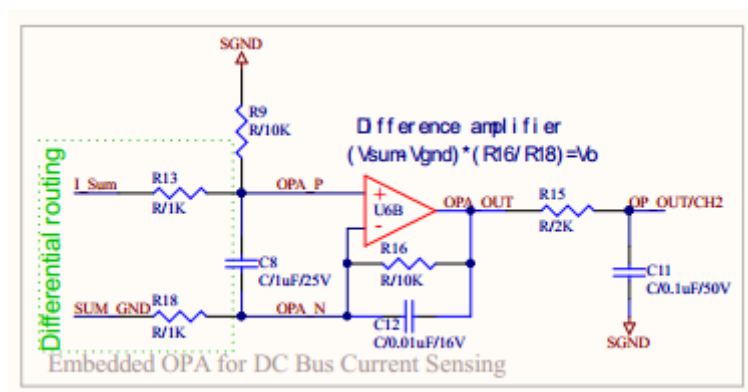
4.3 三相全橋變頻器



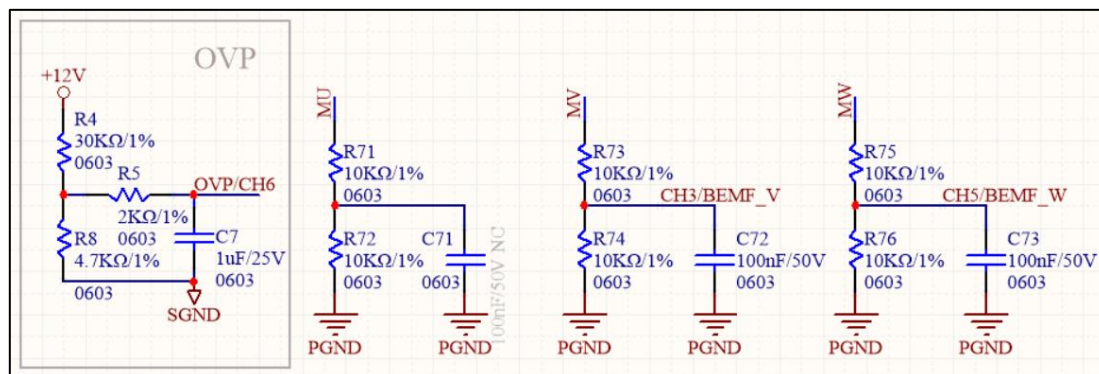
4.4 兩相採樣電路



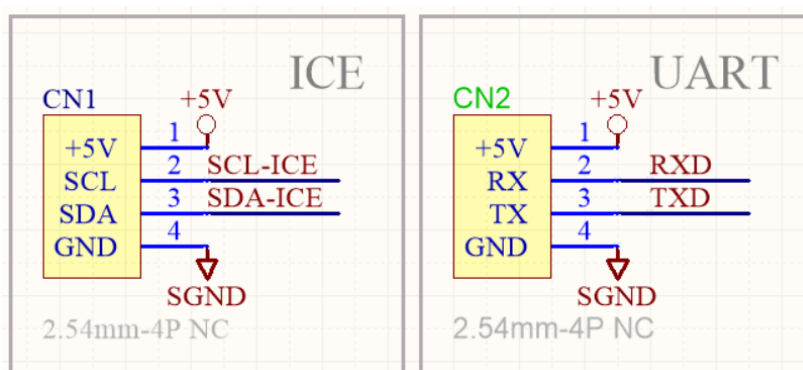
4.5 母線電流採樣電路



4.6 BEMF回授電路



4.7 外部端口



4.8 其他周邊

