



HY16F 系列

產品應用注意說明書

HY16F Series Application Note

Table of Contents

1.	前言	5
2.	HY16F 系列 IP 使用注意事項	6
2.1.	WDT	6
2.1.1.	WDT Reset-使用注意事項	6
2.1.2.	WDT Reset-範例程式	6
2.2.	GPIO	7
2.2.1.	GPIO 複用引腳 PT3.6/PT3.7-使用注意事項	7
2.2.2.	GPIO 複用引腳 PT3.6/PT3.7-範例程式	7
2.3.	類比電源 VDDA, VDD18, VPP	8
2.3.1.	VDDA -使用注意事項	8
2.3.2.	VDD18-使用注意事項	8
2.3.3.	VPP-使用注意事項	10
2.4.	ADC.....	11
2.4.1.	ADC 通道切換-使用注意事項.....	11
2.4.2.	ADC 通道切換-範例使用說明.....	11
2.4.3.	ADC 啟動時間穩定問題	12
2.4.4.	ADC 線性度問題	12
2.4.5.	ADC-IA 使用說明	13
2.5.	Power	14
2.5.1.	Active Mode --> Idle Mode/Sleep Mode-使用注意事項	14
2.5.2.	Active Mode --> Idle Mode/Sleep Mode-範例程式	14
2.5.3.	Idle Mode/Sleep Mode --> Active mode-使用注意事項	15
2.5.4.	Idle Mode/Sleep Mode --> Active mode-範例程式	15
2.5.5.	Idle Mode/Sleep Mode --> Active mode- GPIO 喚醒使用注意事項	16
2.6.	內部高頻 HAO & 內部低頻 LPO	17
2.6.1.	內部高頻 DrvCLOCK_CalibrateHAO -使用注意事項.....	17
2.6.2.	內部高頻 DrvCLOCK_CalibrateHAO -範例程式	17
2.6.3.	內部高頻 HAO & 內部低頻 LPO –使用注意事項.....	17
2.6.4.	內部高頻 HAO =16MHz –使用注意事項	18
2.7.	Flash	19
2.7.1.	Flash Read/Write –使用注意事項	19

2.7.2.	Flash 資料變異 –建議與改善方針	19
2.8.	LCD	21
2.8.1.	DrvLCD_VLCDTrim–使用注意事項	21
2.8.2.	DrvLCD_VLCDTrim–範例程式.....	21
2.8.3.	LCD 頻率 LCK –使用注意事項.....	22
2.9.	RTC	23
2.9.1.	RTC –使用注意事項.....	23
2.10.	Timer	24
2.10.1.	TimerA –使用注意事項.....	24
2.11.	DAC.....	25
2.12.	R2ROP	26
2.12.1.	R2ROP 當 unit gain buffer –使用注意事項	26
2.13.	ECK/EDIO	26
2.13.1.	ECK/EDIO 開發通訊口引腳-使用注意事項	26
3.	HY16F 系列開發工具應用說明	27
3.1.1.	AndeSightRDS V2.0.1 轉換到 AndeSightRDS V2.1.1	27
3.1.2.	Stack 佔多少 SRAM 空間問題	31
4.	HY16F 系列加密&解密功能說明.....	32
5.	HY16F 系列硬體設計注意事項	33
5.1.1.	電路設計注意事項.....	33
5.1.2.	PCB Layout 注意事項	34
5.1.3.	RS 防護設計注意事項.....	42
5.1.4.	ESD 防護設計注意事項	45
5.1.5.	COB 打線注意事項	46
6.	HY16F 系列軟體設計注意事項	47
6.1.	HY16F 的 VDD3V 上電穩壓判斷	47
7.	REVISION HISTORY	49

注意：

- 1、本說明書中的內容，隨著產品的改進，有可能不經過預告而更改。請客戶及時到本公司網站下載更新 <http://www.hycontek.com>。
- 2、本規格書中的圖形、應用電路等，因第三方工業所有權引發的問題，本公司不承擔其責任。
- 3、本產品在單獨應用的情況下，本公司保證它的性能、典型應用和功能符合說明書中的條件。當使用在客戶的產品或設備中，以上條件我們不作保證，建議客戶做充分的評估和測試。
- 4、請注意輸入電壓、輸出電壓、負載電流的使用條件，使 IC 內的功耗不超過封裝的容許功耗。對於客戶在超出說明書中規定額定值使用產品，即使是瞬間的使用，由此所造成的損失，本公司不承擔任何責任。
- 5、本產品雖內置防靜電保護電路，但請不要施加超過保護電路性能的過大靜電。
- 6、本規格書中的產品，未經書面許可，不可使用在要求高可靠性的電路中。例如健康醫療器械、防災器械、車輛器械、車載器械及航空器械等對人體產生影響的器械或裝置，不得作為其部件使用。
- 7、本公司一直致力於提高產品的品質和可靠度，但所有的半導體產品都有一定的失效概率，這些失效概率可能會導致一些人身事故、火災事故等。當設計產品時，請充分留意冗餘設計並採用安全指標，這樣可以避免事故的發生。
- 8、本規格書中內容，未經本公司許可，嚴禁用於其他目的之轉載或複製。

1. 前言

本文, 產品應用注意說明適用於以下紘康 HY16F 系列產品 :

- HY16F18x Series : HY16F184, HY16F187, HY16F188.
- HY16F19xB Series : HY16F196B, HY16F197B, HY16F198B.
- HY16F39 Series : HY16F3981.

2. HY16F 系列 IP 使用注意事項

在此章節特別針對 HY16F 系列在特定 IP 使用上，說明使用注意事項。如果使用該 IP 做應用，但沒注意到 IP 的使用限制，則將可能發生不可預期的結果。使用者在使用該 IP 之前，應花時間特別研讀過後，再設計使用，可減少許多開發上的冤枉之路。

2.1. WDT

看門狗 WDT 為顧名思義的為晶片的看守者，主要用於產生喚醒事件，或晶片發生不可預期當機後，能進行基礎自動重置功能。在此特別加強說明 WDT Reset 功能使用注意事項。

2.1.1. WDT Reset-使用注意事項

當 WDT Reset Mode 功能開啟之後，就無法再切換回 WDT Timer Mode，即設置 WDNMI 0x40108[6]=1b 之後，就無法再設置回 WDNMI 0x40108[6]=0b，使用者在設置與硬用此功能之前，需要先知道 WDT Reset Mode 有此應用限制。此外，程式設計者也需注意到 WDT 功能開啟之後，WDT Count, WDTO 0x40108[30:16] 就會開始計數，這也是無法關閉的，即設置 ENWDT 0x40108[4]=1b，就無法再設置回 ENWDT 0x40108[4]=0b。所以如果開啟設置 WDNMI 0x40108[6]=1b 與 ENWDT 0x40108[4]=1b 之後，記得要在預期規劃時間內做 WDT 計數器歸零動作，即設置 0x40108[5]=1b，以避免程式開發過程忘記清除 WDT 計數器，造成 WDT Reset 事件產生動作。注意：當 WDT Reset 發生之後，各 IP 的暫存器狀態會維持原狀，即 WDT Reset 前與 WDT Reset 之後 IP 暫存器狀態一樣。WDT Reset 發生之後，程式只會先跳回到 Reset 向量位置，之後再往下執行 Startup code 與 CPU 初始化等相關動作，但 IP 暫存器狀態不會重新初始化。這與實際對 HY16F 硬體 Reset pin 做 Reset，結果是有差異的。

2.1.2. WDT Reset-範例程式

```
DrvWDT_Open(E_IRQ, E_PRE_SCALER_D32); //設置看門狗(WDT)為 IRQ mode 及 CLK / 32  
DrvWDT_ResetEnable(); // WDT 中斷工作模式選擇為 Reset Mode
```

2.2. GPIO

GPIO 口都具有多個複用功能，但是一般同一時間只能設置某個功能有效，所以在不需用到複用功能時，請記得關閉複用功能。PT3.6/PT3.7 這兩根複用 GPIO 在使用上需特別注意，如果同時開啟類比與數位功能的時候，將可能會發生 VDDA 電源倒灌並且造成 ADC 量測異常等問題。

2.2.1. GPIO 複用引腳 PT3.6/PT3.7-使用注意事項

PT3.7/OPO 複用引腳:

- 設計為 OPO output ; 控制暫存器 OPOE 0x41900[1]=1b, PT3PU7 =PT3OE7=PT3IE7=0b
- 設計為 PT3.7 GPIO input ; 控制暫存器 ENRFO 0x40400[1]=0b, PT3IE7=1b
- 設計為 PT3.7 GPIO output ; 控制暫存器 ENRFO 0x40400[1]=0b, PT3IE7=1b(即使不作為輸入也必須強制設定), PT3OE7=1b

PT3.6/REFO 複用引腳 :

- 設計為 REFO output ; 控制暫存器 ENRFO 0x40400[1]=1b, PT3PU6=PT3OE6=PT3IE6=0b
- 設計為 REFO input ; 控制暫存器 ENRFO 0x40400[1]=0b, PT3PU6=PT3OE6=PT3IE6=0b
- 設計為 PT3.6 GPIO input ; 控制暫存器 ENRFO 0x40400[1]=0b, PT3IE6=1b
- 設計為 PT3.6 GPIO output ; 控制暫存器 ENRFO 0x40400[1]=0b, PT3IE6=1b(即使不作為輸入也必須強制設定), PT3OE6=1b

2.2.2. GPIO 複用引腳 PT3.6/PT3.7-範例程式

使用者設計 PT3.6 與 PT3.7 當做 GPIO Output 功能時候，則 PT3.6 與 PT3.7 程式的設置初始化流程如下：

```
DrvPMU_REFO_Disable(); //setting ENRFO 0x40400[1]=0b
DrvGPIO_Open(E_PT3,0xC0,E_IO_INPUT); // setting PT3IE6=1b, PT3IE7=1b
DrvGPIO_Open(E_PT3,0xC0,E_IO_OUTPUT); // setting PT3OE6=1b, PT3OE7=1b
```

2.3. 類比電源 VDDA, VDD18, VPP

使用者在設計 HY16F 系列線路圖的時候，應該要特別注意到 VDDA, VDD18, VPP 這三隻腳位的連接。

2.3.1. VDDA -使用注意事項

VDDA 的電容可連接範圍在 1uF~10uF, 但使用時候需注意到 VDDA 開啟時候需要有等待穩定時間, 電容越大, VDDA 需要等待的穩定時間也越久, VDDA 的外掛電容大小也與負載和電流驅動能力有關。

當 VDDA 接入一個 1uF 對地電容時, VDDA 的穩定時間至少需要大於 0.5ms, 當 VDDA 接入一個 10uF 對地電容時, VDDA 的穩定時間至少需要大於 1ms. 若在 VDDA 未穩定情況下啟動 ADC 功能, 將會造成錯誤的 ADC 資料擷取。

VDDA 外掛電容的大小主要與驅動電流的能力有關·依 HY16F 系列晶片的設計要求如下：

VDDA 驅動能力要求為 1mA 時： 外接的對地電容要為 1uF ·

VDDA 驅動能力要求為 10mA 時： 外接的對地電容要為 10uF

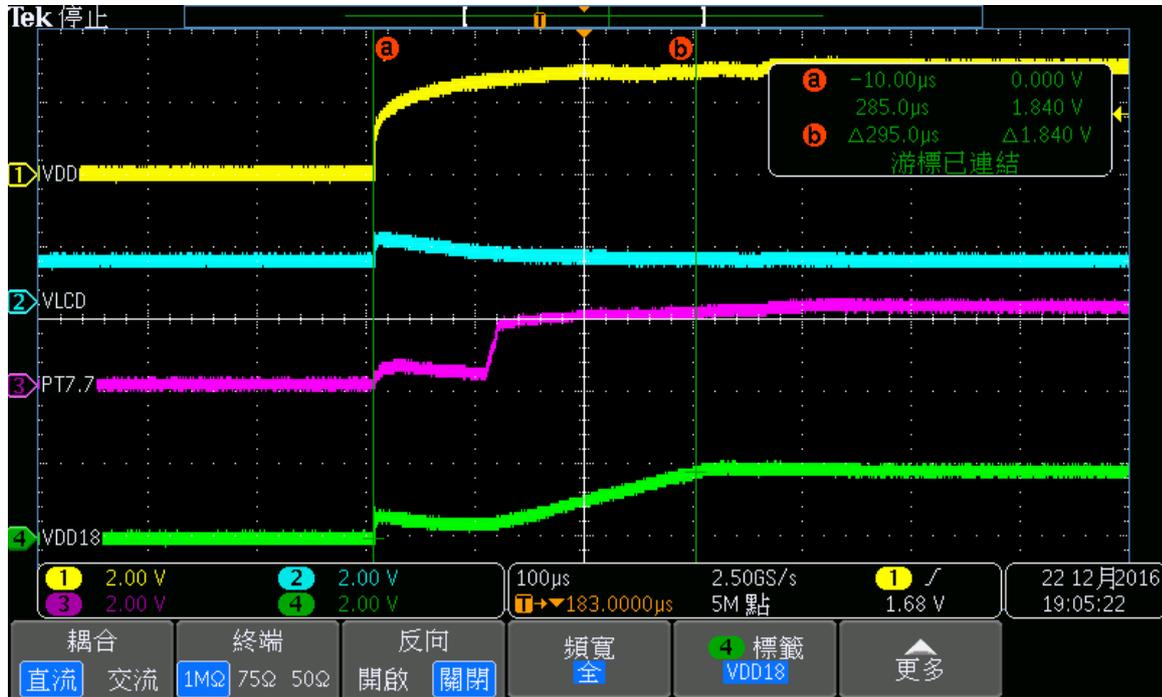
建議使用者在 VDDA 對地電容統一為 10uF, 以增強驅動能力及電壓的穩定度

注意：因 VDDA 電容大小與驅動能力及電壓的穩定度有關，建議電容的選擇為 10uF-Y5R or 10uF-X5R 規格，電容溫度與電容量變化在 15%以內的誤差規格

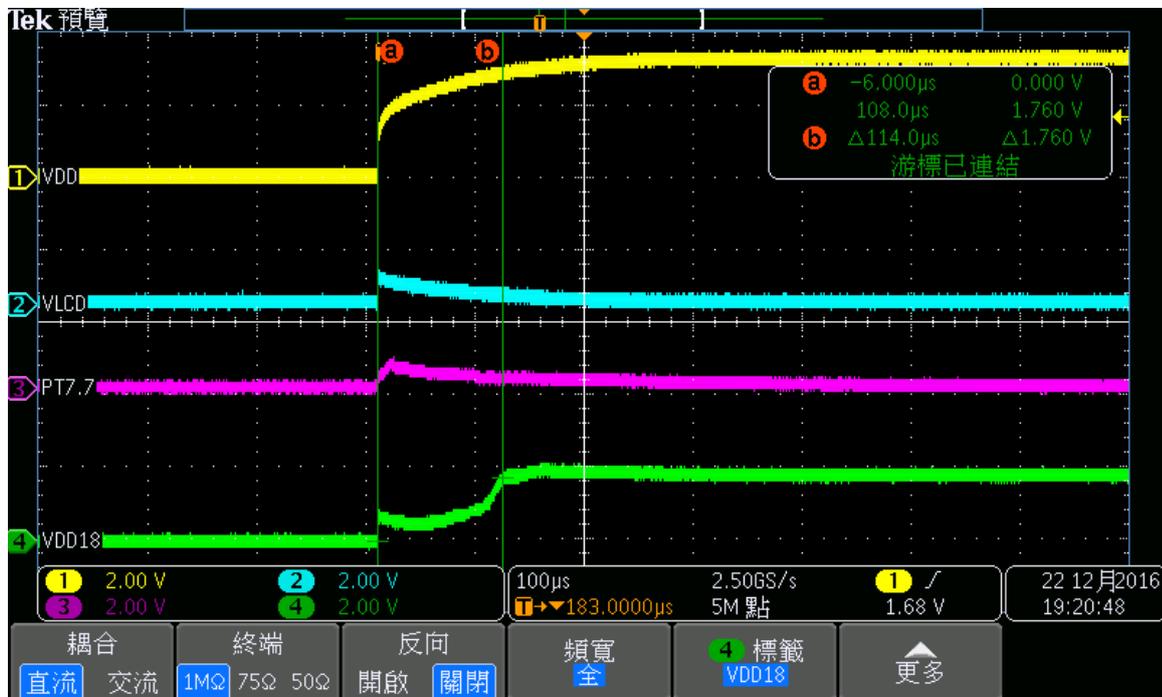
2.3.2. VDD18-使用注意事項

VDD18 的電容只能連接 1uF. 如果 VDD18 的電容連接太大, 則可能會造成上電時候產生 LCD 殘影問題。

如下圖為例：VDD18 使用 10uF 電容圖形，因為 VDD18 從 power on 到穩定時間較長，約為 295us，觀察 SEG 腳位 PT7.7，因為控制邏輯尚未穩定，PT7.7 已經被異常拉起，導致 LCD 殘影現象發生。



如下圖為例：VDD18 使用 1uF 電容圖形，因為 VDD18 從 power on 到穩定時間約為 114us，觀察 SEG 腳位 PT7.7，因為控制邏輯已經穩定，PT7.7 沒有被異常拉起，不會有 LCD 殘影現象發生。



2.3.3. VPP-使用注意事項

當 VPP 引腳連接 0.1uF 的時候，在 Andesight IDE 做線上燒錄或使用 HY16F WRITER 燒錄器做燒錄，都會顯示燒錄失敗訊息。所以 VPP 引腳不能連接任何電容，也不能做任何負載電壓輸出應用，VPP 引腳應該維持保留 NC 狀態(保持空接狀態)。

2.4. ADC

ADC 做通道切換的時候(EX:從 AIO0-AIO1 切換到 AIO2-AIO3),應該要注意到,需要做 ADC Combfilter(梳狀濾波器)設置,不然可能會造成 ADC Offset 停在不同的狀態而發生量測結果漂移的現象

2.4.1. ADC 通道切換-使用注意事項

當使用者做 ADC 通道切換時,因量測到的訊號不同,所以前兩筆的 ADC 訊號是無效的
使用者可利用復位(CFRST=1b)控制來達到自動丟棄前兩筆,第三筆才會產生ADC中斷來讀取ADC訊號的動作
如果實際應用中,沒有切換 ADC input channel 的需求,量測通道都是一樣的,就可不需做復位 CFRST=1b 動作.

2.4.2. ADC 通道切換-範例使用說明

Change ADINP,ADINN to A (EX : AIO0-AIO1)

comb filter reset,

sample A0, 硬體自行丟棄

sample A1, 硬體自行丟棄

sample A2, 從硬體看到的第一筆中斷輸出值

Change ADINP,ADINN to B (EX : AIO2-AIO3)

comb filter reset,

sample B0, 硬體自行丟棄

sample B1, 硬體自行丟棄

sample B2, 從硬體看到的第一筆中斷輸出值

Change ADINP,ADINN to A (EX : AIO0-AIO1)

comb filter reset,

sample A3, 硬體自行丟棄

sample A4, 硬體自行丟棄

sample A5, 從硬體看到的第一筆中斷輸出值

Change ADINP,ADINN to B (EX : AIO2-AIO3)

comb filter reset,

sample B3, 硬體自行丟棄

sample B4, 硬體自行丟棄
sample B5, 從硬體看到的第一筆中斷輸出值

Change ADINP,ADINN to A (EX : AIO0-AIO1)
comb filter reset,
sample A6, 硬體自行丟棄
sample A7, 硬體自行丟棄
sample A8, 從硬體看到的第一筆中斷輸出值

Change ADINP,ADINN to B (EX : AIO2-AIO3)
comb filter reset,
sample B6, 硬體自行丟棄
sample B7, 硬體自行丟棄
sample B8, 從硬體看到的第一筆中斷輸出值

當 ADC OSR=32768, ADC Clock=327Khz, 則 ADC Output rate 約為 10sps=0.1sec
因為 ADC 3nd comb filter 架構, 所以一開始測試時, 需要丟棄前兩筆數值, 第三筆數值才是穩定狀態,
也就是說每次切換通道之後的第一筆 ADC 穩定數值輸出時間為 $3/10=0.3\text{sec}$. 但如果 ADC 通道沒有做切換的
時候, 因為不需再做 CFRST=1b 動作, 所以第二筆, 第三筆直到第 N 筆資料的 ADC 穩定輸出時間都為 0.1sec.

2.4.3. ADC 啟動時間穩定問題

每一次的 VDDA 做 on/off 切換都需要等待穩定時間. 當 VDDA 沒有任何負載的時候, 並且在 ADC 是 external input short 情況下, 不會有 ADC 啟動穩定時間的問題. 而當 VDDA 有負載與 ADC 有輸入阻抗的時候, 會觀察到 ADC 前面幾筆資料不穩定的問題. 因此, 在實際應用中, 如果有做 VDDA on/off 切換, 當 VDDA on 的時候, 需要先等待 100ms 的穩定時間, 再做 ADC 資料的擷取, 可以避免 ADC 前面幾筆資料不穩定問題發生.

2.4.4. ADC 線性度問題

HY16F系列中, ADC的最佳可量測電壓範圍是會隨著參考電壓的設置有關, 使用者在使用時候應注意, 避免ADC線性誤差問題發生.

以下為使用注意事項 :

當REFP-REFN(VREF) = 1*VDDA時,

則VINP-VINN的差動輸入信號不能大於 $1/2*VDDA$, 否則會有線性度問題

當REFP-REFN(VREF) = $1/2*VDDA$ 時,

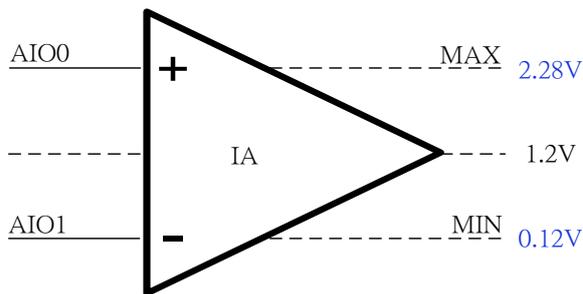
則 VINP-VINN 的差動輸入信號不能大於 $0.9 \cdot V_{REF}$ ，否則會有線性度問題

舉例：設置 $REFP-REFN=VDDA-VSSA$ ，並且 $VDDA=2.4V$ ，此時設置 $FRb=0b$ ，滿量程參考電壓輸入，即 $V_{REF} \cdot 1$ 這時候 ADC 最大可量測的電壓 V_{in} 範圍是 $1/2 \cdot VDDA=1.2V$ ，並非是 $0.9 \cdot V_{REF}=2.16V$ 。因此，在此設置條件下，如果量測電壓 V_{in} 是超出 $1.2V$ 以上，就會有 ADC 線性誤差問題發生。

2.4.5. ADC-IA 使用說明

HY16F 系列中，只有 HY16F3981 在 ADC 前端具有 IA(儀表放大器功能)，在使用儀表放大器需要注意到 IA 的量測輸入訊號是有 IA Input range 使用限制，輸入範圍為 $BIAS=1.2V$ ，input signal= $\pm 1.08V$ 。在實際使用上，可這樣設計使用，ADC 量測通道設置為 AIO0-AIO1，在 ADC 的輸入負端(AIO1)輸需要有一個 BIAS 參考電壓 (EX:可用 HY16F3981 的 REFO 輸出 $1.2V$ 當做 BIAS 參考電壓)，如果 IA 輸入端沒有連接一個 BIAS 參考電壓，在量測上會發生問題，使用者需特別注意到此 IA 應用限制。

HY16F3981 IA Input Range : $BIAS=1.2V$ ，input signal= $\pm 1.08V$



HY16F3981 的 IA 雖然具有高輸入阻抗特性，但是在連接具有高輸入阻抗特性的 Sensor 時，此時如果程式又設置為 IA Chopper On 的模式(即 $0x41600[5:4]=11b$)，當 Sensor 輸入阻抗大於 $10k$ 歐姆的時候，會有整體的 ADC 精度衰減的情況出現，隨著輸入阻抗越大，ADC 精度衰減越多。建議使用者在連接高輸入阻抗 Sensor 訊號的時候，程式設置為 IA Chopper Off(即 $0x41600[5:4]=00b$)，利用軟體 SW Chopper 的方式來做 ADC offset 雜訊扣除的方式。此方式不會受高輸入阻抗而影響到 ADC 的精度。

2.5. Power

HY16F 在省電模式下 Idle Mode 可以達到 5uA. Sleep Mode 可以達到 3.5uA, 當在 Sleep Mode 進入 Sleep Mode 之前把 VDD18 LDO 做 OFF 動作, 可以達到 2.5uA. 使用者在進入 Idle 或 Sleep Mode 之前, 必須要確保相關類比電源與內部高頻晶振都是關閉情況下, 才可達到與規格書一樣的省電功耗. 此外, 當從省電模式到喚醒之後, 使用者也需要注意需要重新初始化與開啟相關應用 IP. 才可避免發生進入省電模式之前 IP 工作正常, 但喚醒之後 IP 工作異常的問題發生. 此應用問題, 時常發生在程式開發階段, 程式設計者應該特別注意.

2.5.1. Active Mode --> Idle Mode/Sleep Mode-使用注意事項

在進入省電模式之前, 可按照順序做以下設置 :

1. 把 CPU 的工作頻率先切換到 LPO 低頻之後, 再把 HAO 高頻關掉.
2. 如有開啟相關類比電源輸出, 則也需要做相對應的類比電源關閉動作.
3. GPIO 的 Pull High 或 Pull Low 設計, 則是依照外部線路設計不同來做相對應的設置, 避免進入 Sleep/Idle 時候發生漏電或電壓倒灌問題.
4. 同時確保在進入省電模式之前, 必須也要確保相對應的喚醒中斷功能是已經開啟, 這樣才能達到與規格書一樣的省電規格.

2.5.2. Active Mode --> Idle Mode/Sleep Mode-範例程式

以下範例, 程式設計有應用到 ADC 與 LCD IP, 當要進入 Sleep Mode 之前, 做相對應關閉動作 :

```
//Sleep Mode setting, Closed VLCD+Clock
DrvLCD_VLCDMode(0); //LCD VLCD
while((inw(0x41B00)&(1<<IDF))==0); //Wait LCD Idle, IDF=20
DrvCLOCK_SelectMCUClock(E_LS_CK,0); //SET CPUCKL=LPO/1
DrvCLOCK_CloseIHOSC(); //Close HAO

// Sleep Mode setting, Closed VDDA+ADC
DrvPMU_VDDA_LDO_Ctrl(E_PullDown); //Closed VDDA
DrvPMU_BandgapDisable(); //Closed Bandgap
DrvADC_Disable(); //Closed ADC

//Enter sleep Mode
DrvPMU_LDO_LowPower(1); //Closed VDD18 LDO
SYS_LowPower(0); //Enter Sleep Mode
```

說明：該範例程式，在進入 Sleep Mode 之前，先做關閉 VLCD 與 Clock 動做，之後再做 VDDA 與 ADC 關閉動做，確認相關 IP 都完全關閉之後，再進入 Sleep Mode 之前做最後的 VDD18 LDO 關閉動做，進入 Sleep Mode 之後可以量測到 2.5uA 的功耗。

2.5.3. Idle Mode/Sleep Mode --> Active mode-使用注意事項

要從 Idle 或 Sleep Mode 喚醒之後，可按照順序做以下設置：

1. 首先可把 HAO 高頻震盪打開，HAO 高頻震盪打開之後，也需要先等待一段穩定時間。
2. HAO 穩定才做後續的其它類比電源重新開啟與 IP 設置。

舉一個的應用實例說明：PWM 頻率源選擇為 HAO，進入 Sleep Mode 之後會把 HAO 高頻震盪關閉，當晶片從 Sleep Mode 喚醒之後，重新開啟 PWM 做輸出，會發現 PWM 沒有動作，這時候是因為 HAO 沒有先做重新開啟動做，PWM 的頻率源 HAO 沒有震盪。諸如此類的應用，使用者在開發程式的時候，都需要特別注意到。

2.5.4. Idle Mode/Sleep Mode --> Active mode-範例程式

以下範例程式為，當從 Sleep Mode 喚醒之後，做相對應 ADC 與 LCD IP 重新初始化動作

```
//VDD18 LDO On
DrvPMU_LDO_LowPower(0); //Enable VDD18 LDO

//Clock start to work
DrvCLOCK_EnableHighOSC(E_INTERNAL,10); //Enable HAO
DrvCLOCK_SelectMCUClock(E_HS_CK,0); //SET CPUCKL=LPO/1
Delay(5000); //Delay time for HAO stable

// LCD start to work
DisplayInit(); // re-initial LCD

//ADC start to work
DrvPMU_VDDA_LDO_Ctrl(E_LDO); //Enable VDDA
DrvADC_ClearIntFlag();
DrvADC_Enable(); //Enable ADC
DrvADC_CombFilter(ENABLE); //Enable comb filter
DrvADC_EnableInt();
```

說明：該範例程式，再從 Sleep Mode 喚醒之後，先做重新開啟 Clock 動作，之後才做 LCD 與 VDDA 與 ADC 等相關 IP 初始化動作。

2.5.5. Idle Mode/Sleep Mode --> Active mode- GPIO 喚醒使用注意事項

HY16F 的 GPIO 時常被拿來設計做為 Sleep or Idle Mode 的喚醒按鍵設計，在程式的設置上需要特別注意增加各 GPIO 的 IDF 中斷條件旗標判斷，這可以避免發生 GPIO 喚醒 Sleep or Idle Mode 會死機的問題發生。舉例：如果設置使用 PT1.7 做為 Idle or Sleep Mode 的喚醒按鍵，則晶片在進入 Idle or Sleep Mode 之前，必須要先判斷 PT17IDF 中斷條件旗標是否為 1，中斷條件旗標為 1 才可進入 Idle or Sleep Mode。

GPIO Base Address + 0X0C (0X4080C)						
PT1CR3 (PT1 Control Register 3)						
Bit	[31:24]	[23:21]	[21:18]	[17:16]		
名稱	PT17IDF~ PT10IDF	PT17ITT	PT16ITT	PT15ITT		
RW	R-0	RW-0				
Bit	[15]	[14:12]	[11:9]	[8:6]	[5:3]	[2:0]
名稱	PT15ITT	PT14ITT	PT13ITT	PT12ITT	PT11ITT	PT10ITT
RW	RW-0					

位元	名稱	描述
		PT1.7 中斷條件旗標 (舉例：可在進入 Sleep Mode 之前判斷該 bit 為 0b 或 1b。 如果為 1b 代表中斷條件可成立，則進入 Sleep Mode 之後， 可以透過 PT1.7 喚醒，如果為 0b，則無法透過 PT1.7 喚醒)

2.6. 內部高頻 HAO & 內部低頻 LPO

CPU 開啟切換內部高頻振盪器使用時，應該都需要注意等待穩定時間。不該在振盪器還沒穩定的時候，就急著往下做其它事情產生當機問題。此外，使用者常常發生，批量生產時候，部分晶片會發生 UART 傳輸資料錯誤等問題，該問題也與 HAO 相關設置有密切關係。

2.6.1. 內部高頻 DrvCLOCK_CalibrateHAO -使用注意事項

HY16F19x 系列，各頻段的 HAO 有提供校正值可供使用者做使用。當把校正值填入 HATOR 暫存器之後，HAO 頻率誤差可以縮小在 2%以內。使用者在使用 UART 功能的時候，如果又使用 HAO 當做 UART 主頻率，那一定要使用 DrvCLOCK_CalibrateHAO 此函數，可以把 HAO 頻率誤差落在 2%以內。可以解決批量生產時候，部分晶片發生 UART 傳輸資料錯誤等問題。

2.6.2. 內部高頻 DrvCLOCK_CalibrateHAO -範例程式

範例程式如下：

```
DrvCLOCK_EnableHighOSC(E_INTERNAL,10); //Select HSRC
DrvCLOCK_SelectIHOSC(1); //Select internal 4MHZ
DrvCLOCK_SelectMCUClock(0,0); //CPU CLOCK IS 'hs_ck/1'
DrvCLOCK_CalibrateHAO(1); //Calibration HAO_4MHz=4.147MHz
DrvUART_ClkEnable(1,0); //choose the internal HAO as UART clock source
DrvUART_Open(4147,B9600,DRVUART_PARITY_NONE,DRVUART_DATABITS_8,DRVUART_STOPBITS_1,2);
```

說明：該範例程式，預先設置 HAO=4MHz，這時候 HAO 的頻率誤差為 4MHz +/- 10%，當執行函式 DrvCLOCK_CalibrateHAO(1);之後，把 HAO=4MHz 的效正值填入在 HATOR 內，可以讓 HAO 的頻率為 4.147MHz +/- 2%誤差。當 HAO 頻率設定完成之後，UART 的頻率源設置選項也需要輸入正確的數值。這樣可以有效解決批量生產時候，部分晶片發生 UART 傳輸資料錯誤等問題。並且需注意到，不同段的 HAO 頻率有不同的 Trim 值，所以當設置 HAO=4MHz 的時候，只能選擇 DrvCLOCK_CalibrateHAO(1)，同理，當設置 HAO=2MHz 的時候，只能選擇 DrvCLOCK_CalibrateHAO(0)，依此類推。

2.6.3. 內部高頻 HAO & 內部低頻 LPO –使用注意事項

使用內部晶振需要注意事項：

- 使用 HAO 16MHz 晶振時，晶片工作電壓須限制在高壓動作。
- 設置寄存器 HAOTR 0x40304[7:0]的值，可調整 HAO 輸出頻率值。

舉例:當設定 HAO 工作在 2MHz 時，如果實際輸出只有 1.99MHz，那麼即可藉由調整控制位 HAOTR[7:0]來調整頻率輸出，HAOTR 預設值為 0x80，往上調整即可增加 HAO 實際工作頻率。

- 晶片在默認啟動的震盪器為內部 HAO 2MHz，用戶可進行設置暫存器 0x40300[4:3]來配置為其它的 HAO 輸出頻率值。
- HAO 配置為 4MHz 起震的穩定時間約為 0.5ms。
- 執行休眠(SLEEP)指令後，HAO 震盪器皆會停止震盪進入休眠狀態。
- 從休眠模式(Sleep Mode)到喚醒的時間約 $1024*HAO+2048*LPO \leq 64ms$ 。

注意:如果要滿足上述描述喚醒時間，應該在進入 Sleep Mode 之前，CPU 頻率源選擇為 HAO。

- 執行待機(IDLE)指令，並不會停止 HAO 震盪器，但 CPU 會進入待機狀態。
- 從 Idle Mode 到喚醒時間約 500 個指令週期，當 CPU Clock 使用預設 HAO=2MHz 時，喚醒時間約 250us。
- LPO 為晶片內部低速 RC 震盪器，輸出頻率 35KHz，具有低功耗電流特性，且在晶片上電或喚醒後即起振，同時也是無法關閉，在整個晶片工作過程 LPO 都是運行的。
- LPO 起振穩定時間約為 510us 且是看門狗(WDT)的唯一工作時脈源。
- 執行休眠指令(SLEEP)後，LPO 震盪器皆會停止震盪。
- 執行待機指令(IDLE)後，LPO 不會停止震盪，但 CPU 會進入待機狀態。
- 從 Idle Mode 到喚醒時間約 500 個指令週期，當 CPU Clock 使用預設 LPO=35kHz 時，喚醒時間約 14ms。

2.6.4. 內部高頻 HAO =16MHz –使用注意事項

HY16F 晶片在 HAO=16MHz，並且晶片工作電壓低於 2.2V~1.8V 的情況下，會有 PC counter 機率性發生跳飛的問題發生，導致程式誤動作。因此在設置 HAO=16MHz 的情況下，建議設置 CPU 的主工作頻率為 HAO=16MHz/2 而不要使用 HAO=16MHz/1。這可以降低 PC counter 跳飛的問題。如果應用上必須要工作在 HAO=16MHz/1，則可外部增加 Reset IC，當 HY16F 晶片工作電壓 VDD3V 低於 2.2V 的時候，立刻把電壓拉到 0V，避免發生晶片工作電壓出現在 1.8V~2.2V 的區間..

2.7. Flash

HY16F 系列為 Flash 產品，可以自行規劃與指定 Flash 區塊位置為 Data Flash 做儲存效正參數使用。在使用 Flash 做資料(Write/Read)存取之前，也應該注意到一些使用限制。

2.7.1. Flash Read/Write –使用注意事項

1. 在執行 Flash 燒錄與讀取程式指令之前，必須先執行 SYS_DisableGIE(); 關閉全域使能中斷，這可以避免程式運行異常的行為發生。
2. Flash 有寫入次數限制，Flash 有保證至少 2 萬次的寫入壽命。再進行 Flash Write 寫入之前，一定要有 Erase 動作，這樣寫入才會正常。Flash 壽命是依照 Erase 次數來計算，非寫入次數。Erase all+ N* Page Write，仍然只算是損失一次使用壽命。同一個 Page 使用 N*page Erase，就代表已使用了 N 次壽命。
3. Flash Write 與 Read 都有寫入時間，使用者可以直接使用 HY16F C 函式庫的 Flash 相關控制函數做 Write 與 Read。使用 ROM_BurnWord 或 ROM_BurnPage 約花費 30ms 時間，使用 PageErase 或 SectorErase 約花費 25ms 時間。以上函式已經包含 Erase 功能。
4. 執行 Flash 燒錄指令，必須確保晶片工作電壓 VDD3V 高於 2.7V，如果晶片電壓 VDD3V 低於 2.7V，則可能會發生燒錄錯誤行為。

以下說明 Flash Read/Write 函式庫指令執行時間

1. DrvFlash_Burn_Word(unsigned int addr,unsigned int DelayTime,unsigned int data): 執行時間約 30ms, Delay time 填入時間為 0x2000 (當 HAO=2M 時候填入 0x2000, HAO=4M 時候填入 0x4000 依此類推)
2. ROM_BurnPage(unsigned int addr,unsigned int DelayTime,int* data); 執行時間約 30ms, Delay time 填入時間為 0x2000 (當 HAO=2M 時候填入 0x2000, HAO=4M 時候填入 0x4000 依此類推)
3. PageErase(unsigned int addr,unsigned int DelayTime); 執行時間約 25ms , Delay time 填入時間為 0x2000 (當 HAO=2M 時候填入 0x2000, HAO=4M 時候填入 0x4000 依此類推)
4. SectorErase(unsigned int addr,unsigned int DelayTime); 執行時間約 25ms , Delay time 填入時間為 0x2000 (當 HAO=2M 時候填入 0x2000, HAO=4M 時候填入 0x4000 依此類推)

2.7.2. Flash 資料變異 –建議與改善方針

HY16F 處理客訴的經驗，有發生過 Flash 資料變異的經驗，Flash 資料變異的狀況可能是 bit 由 0 變 1，也可能是 bit 由 1 變 0，並且 Flash 異常的區域可能是 APP.bin 區域，也可能是 Data.bin 區域。所以當發生問題之後，唯一修復的辦法就是重新燒錄一次完整的量產.bin 檔案，即讓 Flash 資料回歸出廠時的狀態。為了避免發生類似的 Flash 異常問題與有效的降低不良情況，在此提出以下建議與改善方針，可有效改善與降低 Flash 異常的發生機率，以下是建議事項說明：

建議 1：應用上, HY16F 晶片如果是選擇工作在 HAO=16MHz/1. 要注意當晶片的工作頻率是高頻情況下 (HAO=16MHz/1)並且晶片工作在低電壓低於 2.2V~1.8V 的情況下, 會有 PC counter 機率性發生跳飛的問題發生, 這樣可能會導致程式誤動作. 如果不幸的又誤動作執行到程式碼 Flash 自我燒錄函式(EX: ROM_BurnPage), 那就有可能會產生 Flash 誤動作燒錄的異常現象. 因此在設置 HAO=16MHz/1 的情況下, 建議設置 CPU 的主工作頻率為 HAO=16MHz/2 而不要使用 HAO=16MHz/1. 這可以降低 PC counter 跳飛的問題. 如果應用上必須只能工作在 HAO=16MHz/1, 則可考慮外部增加 Reset IC, 當 HY16F 晶片工作電壓 VDD3V 低於 2.2V 的時候, 立刻把電壓拉到 0V, 避免發生晶片工作電壓出現在 1.8V~2.2V 的區間. 如果不外加 Reset IC, 當晶片斷電與上電的時候, 也要想辦法減少與縮短電壓 VDD3V 處在 1.8V~2.2V 的這個上升與下降時間

建議 2：執行 Flash 燒錄與讀取程式指令之前, 必須先執行 SYS_DisableGIE(); 關閉全域使能中斷, 這可以避免程式運行異常的行為發生.

建議 3：執行 Flash 燒錄指令, 必須確保晶片工作電壓 VDD3V 高於 2.7V, 如果晶片電壓 VDD3V 低於 2.7V, 則可能會發生燒錄錯誤行為.

建議 4：為了避免在執行 Flash 自我燒錄之前有突波的雜訊干擾電壓造成燒入異常. 所以新增檢查 VDD3V 電壓的 function. 以 HY16F198B 為例子說明：利用 198B 內部的 CMP 來比較 VDD3V 電壓是否大於 2.75V, VDD3V 大於 2.75V 才執行 Flash 自我燒錄. 詳細操作說明可以參考章節 6.1 HY16F 的 VDD3V 上電穩壓判斷

建議 5：在執行 Flash 自我燒錄&讀取之前, 強制執行切換到 HAO=2M/2, 並且增加 Delay 時間, 當 Flash 燒錄或讀取完之後, 再把 HAO 切換為原本預期的工作頻率. 以下舉例說明 HAO=16M/2 的應用範例.

範例 1(Flash 自我燒錄)：當遇到要執行 ROM_BurnPage 這個燒錄指令, 請按照以下方式操作.

```

SYS_DisableGIE();           //關閉 GIE
outw(CLOCK_BASE+8, 0x0202); //設定 MCU Clock/2
outw(CLOCK_BASE, 0x1901);   //把頻率切換為 HAO=2MHz, 所以 HAO=2M/2
Delay(2);                   //增加一小段 Delay 時間
ROM_BurnPage(uiMemPageAddr,0x8000, pucWriteBufTmp); //執行 Flash 自我燒錄
Delay(2);                   //增加一小段 Delay 時間
outw(CLOCK_BASE, 0x1919);   //把頻率切換為預設使用的頻率, 此指令是切換回 HAO=16MHz/2
SYS_EnableGIE(4, 0x1FF);    //重新再開啟 GIE

```

範例 2(Flash 自我讀取)：當遇到要執行 ReadPage 這個讀取指令, 請按照以下方式操作

```

SYS_DisableGIE();           //關閉 GIE
outw(CLOCK_BASE+8, 0x0202); //設定 MCU Clock/2
outw(CLOCK_BASE, 0x1901);   //把頻率切換為 HAO=2MHz, 所以 HAO=2M/2
Delay(2);                   //增加一小段 Delay 時間
ReadPage(0x8000,BufferRx);  //執行 Flash 自我讀取
Delay(2);                   //增加一小段 Delay 時間
outw(CLOCK_BASE, 0x1919);   //把頻率切換為預設使用的頻率, 此指令是切換回 HAO=16MHz/2
SYS_EnableGIE(4, 0x1FF);    //重新再開啟 GIE

```

2.8. LCD

HY16F19xB 系列產品有 LCD IP 功能，使用 LCD 需要注意到 VLCD 工作電壓，也需要注意到 LCD 工作頻率 LCK 設置。如果電壓設置與 LCD 面板電壓不符合則會產生，LCD 顯示看起來亮度不足或著是 LCD 沒有點亮的區塊看起來像是點亮的顯示問題(從可視斜角看會很明顯)。如果 LCD 工作頻率設置過慢，則是會從 LCD 顯示上看起來有閃頻的顯示異常問題。所以 LCD 工作電壓 VLCD 與 LCD 工作頻率 LCK 設置上都需要特別注意

2.8.1. DrvLCD_VLCDTrim–使用注意事項

各 VLCD 電壓有提供校正值可以使用，當使用函式 DrvLCD_VLCDTrim 可以把 VLCD 校正值帶入，使得 VLCD 電壓可以控制在 5%的誤差範圍內，如果沒有使用函式 DrvLCD_VLCDTrim，則 VLCD 電壓誤差則在 10% 範圍內，使用者常常發生批量生產時，發現部分晶片的 LCD 顯示過亮或過暗，這是因為 VLCD 電壓批量生產 10%誤差導致，如果在使用時，有使用函式 DrvLCD_VLCDTrim 則可以把批量生產所產生的 VLCD 電壓問題有效解決，可以控制批量生產 VLCD 電壓誤差範圍在 5%以內。

2.8.2. DrvLCD_VLCDTrim–範例程式

```
//Case5, VLCD=2.55V, +/- 5% :  
DrvLCD_VLCDMode(E_VLCD24);  
DrvLCD_VLCDTrim(5);
```

```
//Case4, VLCD=2.73V, +/- 5%  
DrvLCD_VLCDMode(E_VLCD27);  
DrvLCD_VLCDTrim(4);
```

```
//Case3, VLCD=2.93V, +/- 5%:  
DrvLCD_VLCDMode(E_VLCD30);  
DrvLCD_VLCDTrim(3);
```

```
//Case2, VLCD=3.16V, +/- 5%:  
DrvLCD_VLCDMode(E_VLCD30);  
DrvLCD_VLCDTrim(2);
```

```
//Case1, VLCD=3.43V, +/- 5%: :  
DrvLCD_VLCDMode(E_VLCD33);
```

DrvLCD_VLCDTrim(1);

說明：該範例程式，說明如何設定使用效正值在 VLCD 電壓，使用 DrvLCD_VLCDMode 與 DrvLCD_VLCDTrim 這兩個函式搭配使用。並且需注意到，因為不同的 VLCD 電壓段有各自的 VLCD Trim 值。所以正確設置使用方式應該是要按照範例程式中的 Case1~Case5 則一來使用即可

2.8.3. LCD 頻率 LCK –使用注意事項

LCD 頻率(LCK)設置問題，當 LCD 設置 1/4duty · 外部電壓 R-TYPE，選擇 LS_CK/8/32=136HZ=LCK 的掃描頻率 · LCD 顯示會閃屏，以 4com 來說，如果要眼睛看 LCD 不閃爍，則 LCD frame Freq. 要接近 60Hz. 故 LCD 掃描頻率至少要 $4com \times 60 = 240Hz.$ ，而 LCK 的時間是以半周期(1/2T)來算的，所以 LCK 要設置成 $240Hz \times 2 = 480Hz$ ，所以應該要改成 LS_CK/8/8=544Hz，才可避免接上 4com 有顯示閃爍的問題發生 LCD 規格上會都會寫 LCD frame Freq. (一般為 50~60Hz).

2.9. RTC

HY16F 系列產品, RTC 為常使用的 IP 功能. 在此特別針對 RTC 的 IP 功能做注意事項介紹.

2.9.1. RTC –使用注意事項

1. 程式設計者, 如打算進入省電模式(Sleep/Idle)的時候還是持續讓 RTC 做計數工作, 需要注意, RTC 可以在 Idle Mode 工作, 但是 RTC 在 Sleep Mode 是無法正常工作. 以下為 RTC+Idle Mode 的參考工作電流 :

- (1)RTC + idle without LCD, 8uA
- (2)RTC + Idle + LCD pump, 18uA
- (3)RTC + Idle + LCD pump + HYCON LCD panel, 25uA

2. RTC 的頻率源選擇可以是內部低頻 LPO 或著是外部低頻 LSXT(32768Hz), 如果選擇的是外部低頻 LSXT, 則需要考量到 32768 起振穩定問題, 起振穩定問題牽涉到 PCB Layout 走線和晶振品牌規格都會有關係. HY16F 系列在連接 32768 晶振的外部引腳, 均可選擇調整電容值大小(0~20pF)與示情況並連電阻(10M ohm), 來達到 32768 穩定起振.

3. RTC IP 的暫存器 0x41A04[22:16]可以提供 RTC 補償使用, 當外部振盪器 LSXT 的震盪頻率偏離 32768Hz, 可以透過補償的方式, 讓 RTC 計數頻率趨近於 32768Hz, 可補償的頻率範圍為 : 最大的輸入頻率為 32772Hz · 最小的輸入頻率為 32763Hz. 例如 : 外部振盪器 LSXT 的震盪頻率為 32764Hz, 這時候如果在 0x41A04[22:16] 暫存器填入 0111111b, 則補償值為+126PPM, 可讓 RTC 計數頻率達到 32768Hz.

4. 在實際電路中, 可以通過示波器觀察振盪波形來判斷振盪器是否工作在最佳狀態, 示波器在觀察振盪波形時, 觀察 OSCO 管腳(Oscillator output), 應選擇 100MHz 頻寬以上的示波器探頭, 這種探頭的輸入阻抗高, 容抗小, 對振盪波形相對影響小. 由於探頭上一般存在 10~20pF 的電容, 所以觀測時, 適當減小在 OSCO 管腳的電容可以獲得更接近實際的振盪波形. 工作良好的振盪波形應該是一個漂亮的正弦波, 峰峰值應該大於電源電壓的 70%. 若峰峰值小於 70%, 可適當減小 OSCI 及 OSCO 管腳上的外接負載電容. 反之, 若峰峰值接近電源電壓且振盪波形發生畸變, 則可適當增加負載電容。

5. 在振盪線路中, 很重要的一個特性是整個線路上的負載電容(Load Capacitance;CL), 它是由 gate 端(OSCI) 的頻率調整電容 CG, drain 端 OSCO 的頻率調整電容 CD 及雜散電容 CS 等三個參數共同組成. 其中負載電容及兩個頻率調整電容是已知的, 可以透過下列公式求得雜散電容值:

$$CL = (CG // CD) + CS \quad \text{--->>} \quad CL = [(CG \times CD) / (CG + CD)] + CS$$

通常廠家建議的 32768Hz 外接負載電容(CL)為 7pF、9pF、及 12.5pF (通常以 12.5pF 的規格居多), 而一般雜散電容 (CS) = PCB 雜散電容 + 晶片電路接面電容, 一般約在 3~5pF, 如果暫不考慮 CS, 則 $CL = (CG // CD, 12.5pF = (CG // CD, 得 CG, CD 各用 25pF.$

2.10. Timer

HY16F 系列產品, Timer 為常使用的 IP 功能. 在此特別針對 Timer 的 IP 功能做注意事項介紹.

2.10.1. TimerA –使用注意事項

在使用 TimerA 時候, 程式設定 DrvTMA_Open(0,0)與 DrvTMA_Open(1,0)做計數器功能使用, 會發現到有計數不準與不穩定的現象. HY16F 的 CPU 進出中斷消耗約 200 個機械指令週期, 如果 CPU=8MHz, 則總共消耗時間為 $(1/8000000)*200=25\mu s$. 當設定程式執行 DrvTMA_Open(0,0), 則進入一次的中斷時間為 $8000000/32(ENTAD=1b)/2(TAS=0000b)=125kHz=8\mu s$, 8us 的時間遠遠小於 25us, 真實情況會看到程式是一直卡在 TimerA 中斷而進入不到 main 執行

同理

DrvTMA_Open(1,0), 則一次進入中斷時間為 16us

DrvTMA_Open(2,0), 則一次進入中斷時間為 32us

32us>25us,可以看到穩定輸出, 因此, 使用 TimerA 做計數時候, 最快建議使用為 DrvTMA_Open(2,0).

2.11. DAC

HY16F 的 DAC 單獨, 沒什麼 Driver 能力, 大約 15uA. 如果是 DAC+OP, 那最大有 1mA(VDDA=3V 情況下), 0.5mA(VDDA=2.4V 情況下). 使用者需注意 DAC 推動負載的大小, 避免 DAC 輸出在大負載的時候, 電壓被異常拉低的情況發生.

2.12. R2ROP

HY16F 系列產品, R2ROP 為常使用的 IP 功能. 在此特別針對 R2ROP 的 IP 功能做注意事項介紹

2.12.1. R2ROP 當 unit gain buffer –使用注意事項

R2ROP 當 unit gain buffer 應用的時候, input 從 0~5mV 的時候會存在一個 offset 電壓存在, 並且在此範圍內電壓變化從輸出端是觀察不到, 直到約 6mV 電壓的時候, 電壓輸出才會開始有反應. 如果要有比較好的線性度, 則建議工作電壓是在 $VSSA+0.1V\sim VDDA-0.1V$ 範圍內. 因為此限制, 在此提供一個參考的解決方案, 把整個量測系統工作電壓墊高, 可以利用 HY16F 的 REFO 這隻接腳輸出 1.2V, 來當作 Bias 電壓. 使得參考電壓點從 0V 變成 1.2V.

2.13. ECK/EDIO

2.13.1. ECK/EDIO 開發通訊口引腳-使用注意事項

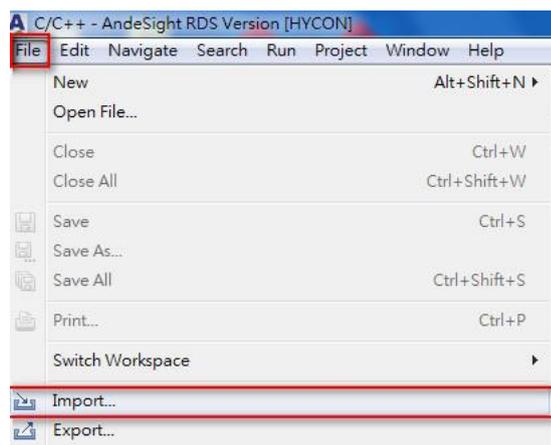
HY16F 系列產品, 程式開發燒錄引腳是透過 ECK 與 EDIO 來通訊. 這兩隻引腳即使沒有使用到, 也不可以空接 Floating. 如果保持空接可能會導致不可預期的異常狀態出現, 例如: 產生異常大電流. 所以在硬體規劃的時候, 建議這兩隻腳連接 1k 歐姆電阻到地(VSS).

3. HY16F 系列開發工具應用說明

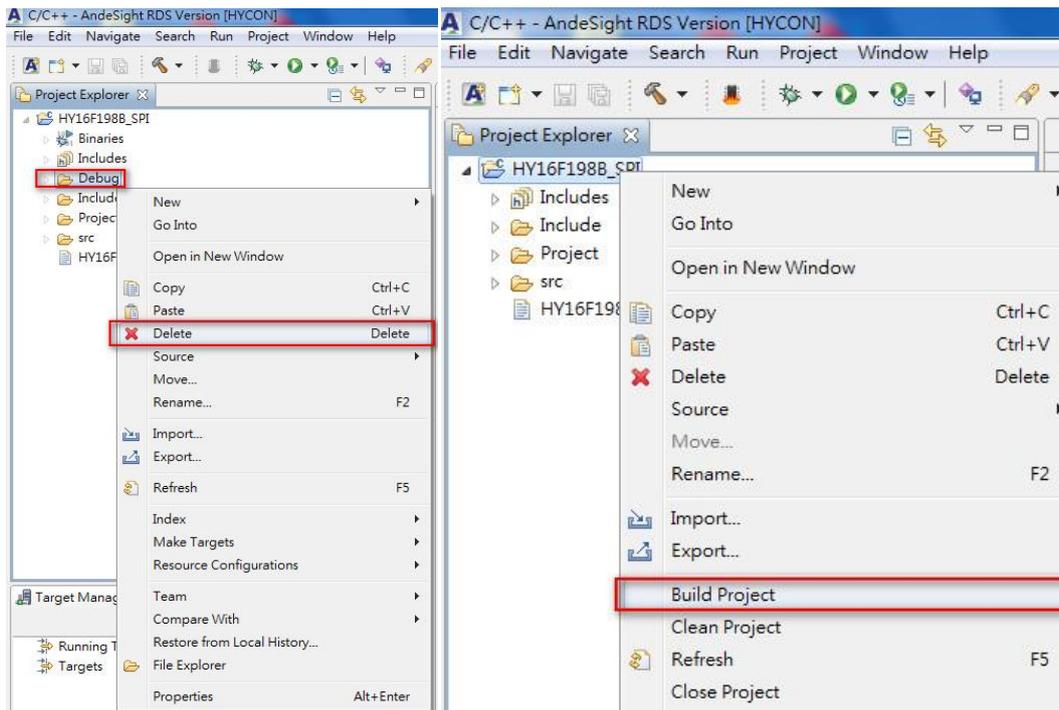
3.1.1. AndeSightRDS V2.0.1 轉換到 AndeSightRDS V2.1.1

在 2017 年以前使用的標準開發工具為 IDE:AndeSight RDS V2.0.1 搭配 HY-Protoool, 在 2017 年之後, 更新升級開發工具為 IDE:AndeSight RDS V2.1.1 搭配 HY16F Mini Link. 使用者應該要使用正確的硬體開發工具與 IDE 軟體版本, 才可正常做 HY16F 程式開發與 Debug 等動作. 此外, 也需注意, 如果舊有的 AndeSight RDS V2.0.1 開發專案包, 如果要移植到 AndeSight RDS V2.1.1 環境做沿續設計使用, 需要先做以下 Step1~Step4 等設定動作, 才可以確保舊有的 RDS V2.0.1 專案包在 RDS V2.1.1 環境轉換是沒問題的, 可以正常做 Debug 等單步執行除錯動作.

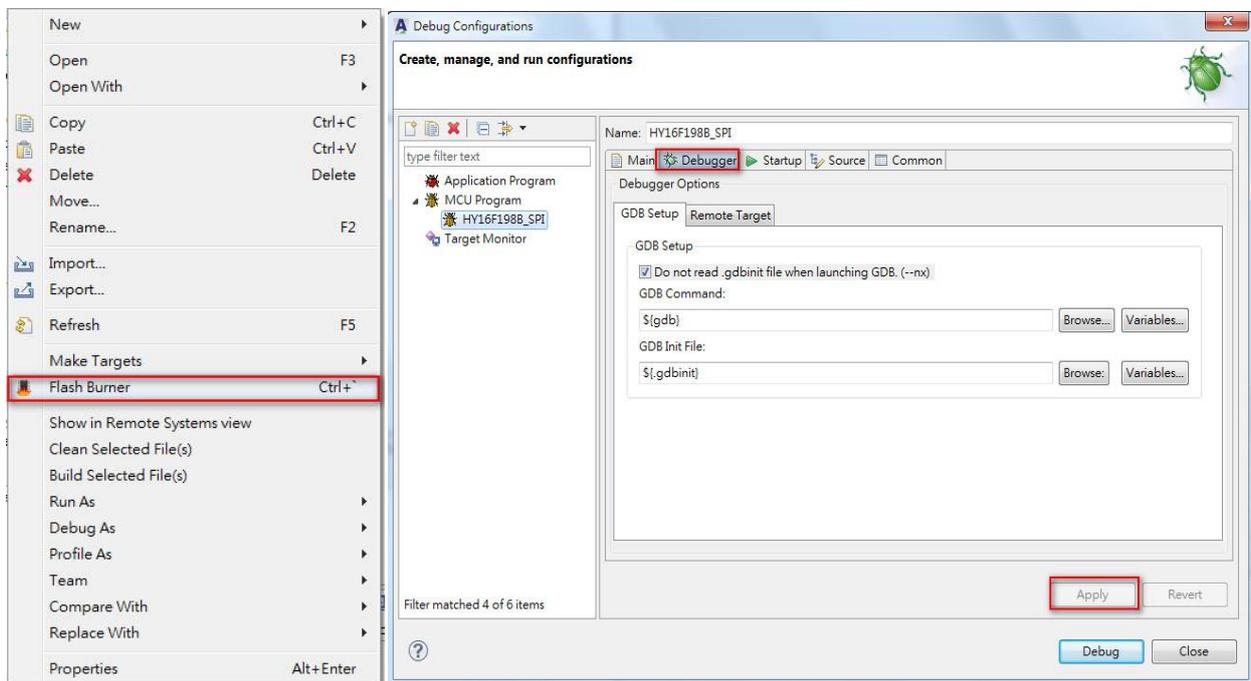
Step1 : 在 RDSV2.1.1 環境下 Import RDS V2.0.1 專案包



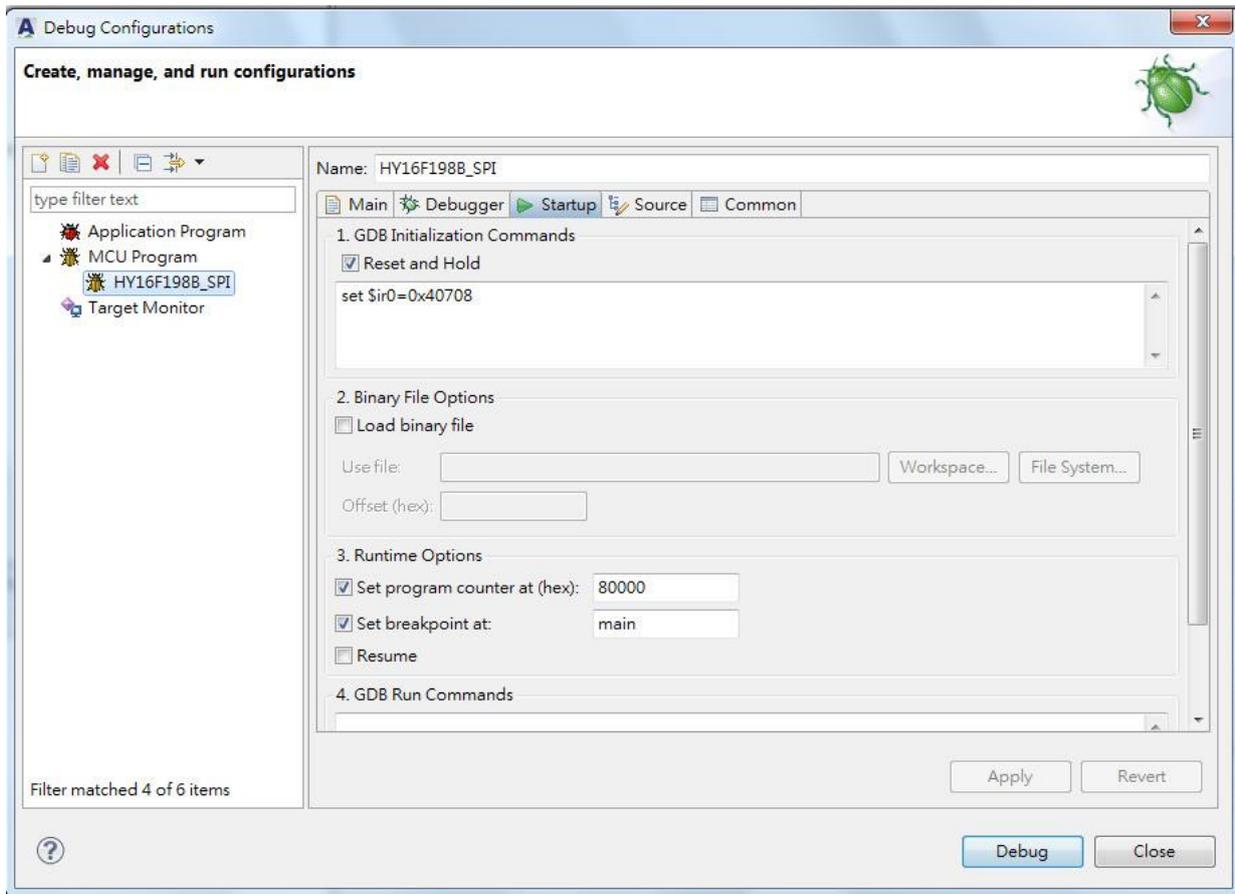
Step2 : 把 Debug 資料夾 Delete 刪除之後, 重新 Build Project, 建立與產生一個新的 Debug 資料夾.



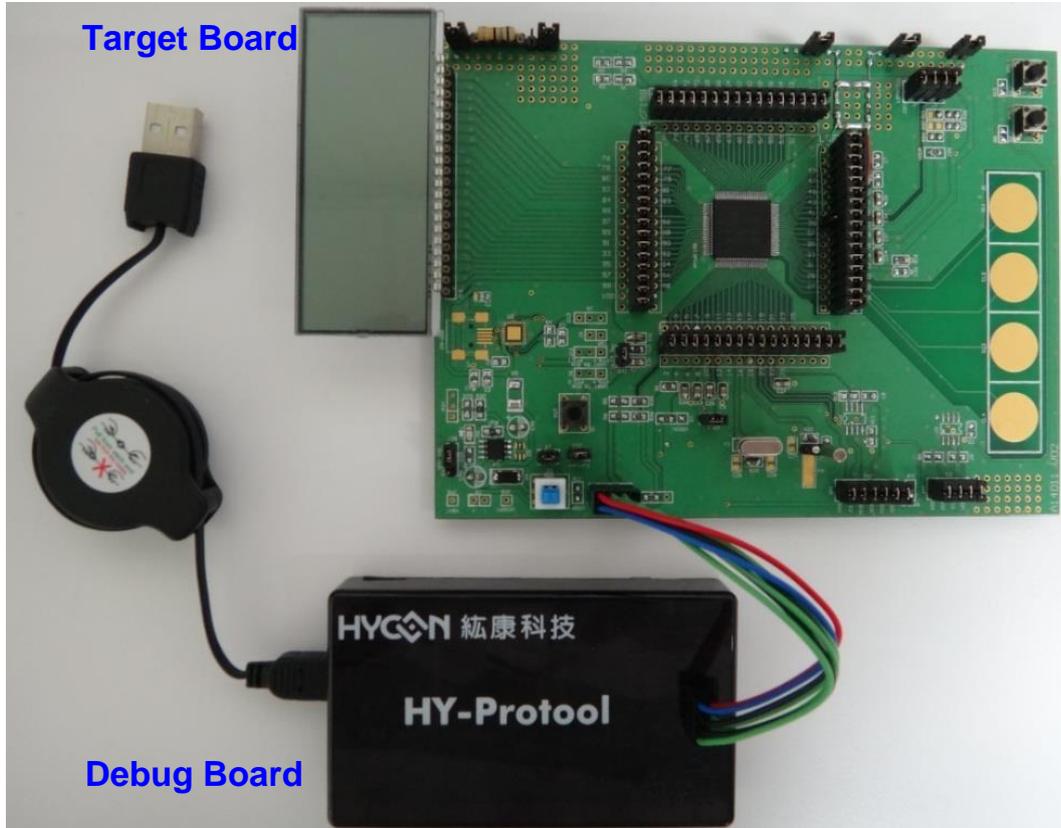
Step3 : Build Project 之後，執行 Flash Burner 進行燒 Code 動作，點選 Debug Configurations 確保 Debugger 環境設定與下圖一模一樣。



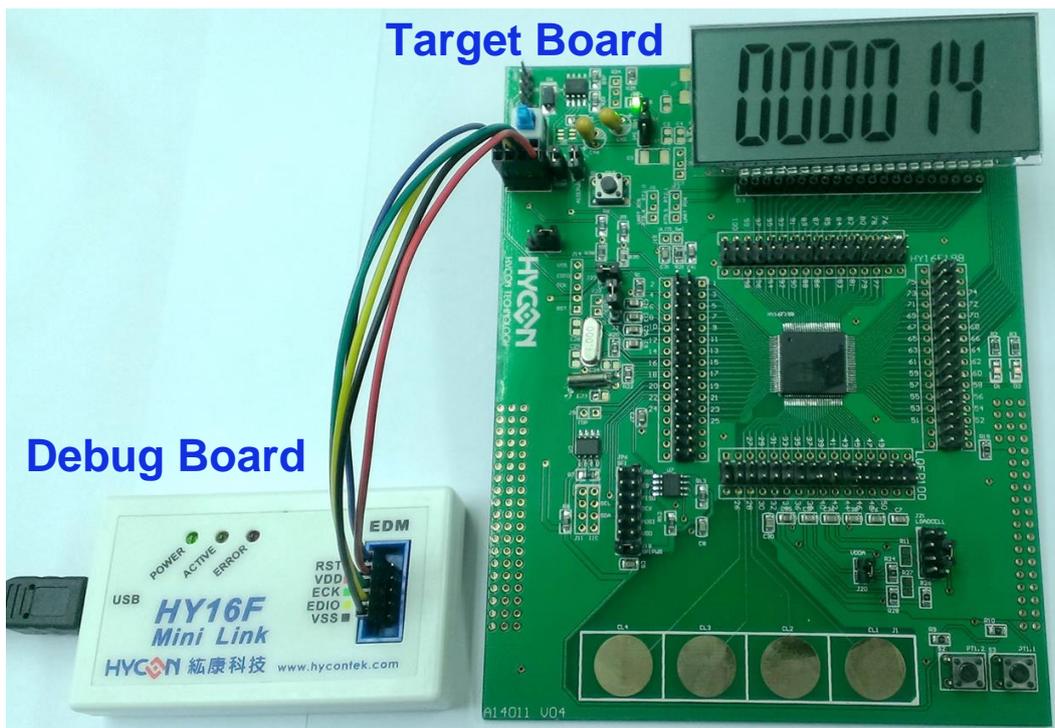
Step4 : 確認 Debug Configurations 環境下的 Startup 設定也與下圖一樣。



下示意圖為 HY16F19xB 搭配 HY-Protool 硬體開發套件(HY16F19x-DK02), 實際使用應搭配 IDE:AndeSight RDS V2.0.1 :



下示意圖為 HY16F19xB 搭配 HY16F Mini Link 硬體開發套件(HY16F19x-DK04), 實際使用應搭配 IDE:AndeSight RDS V2.1.1 :



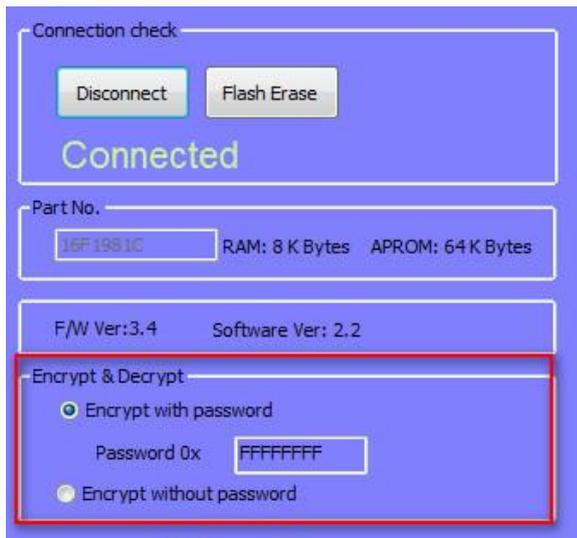
3.1.2. Stack 佔多少 SRAM 空間問題

SRAM 的組成包含四個元素 data+bss+heap+stack. 其中, stack 的起始位置是 0x1fC0, 他是動態配置的, 完全是看程式怎麼寫來動態配置長大的 如果程式寫越多層 call function, 那 stack 就會越長越大, Stack 是從 SRAM 的尾巴位置 0x1fc0 --> 0x1fbf-->0x1fbe 慢慢長回去, 而 bss 與 data 是從 SRAM 的 0x40 起始位置慢慢長上去. 直到跟 bss 與 data 區塊變數與 stack 重疊到, 那這時候可能就會發生宣告變數亂掉的問題, 導致程式錯亂.

舉例 : SRAM 使用 7.5kB 空間(data=522, bss=7239), 代表 stack 只剩下 $8128-7239-522=367$ 空間可以使用, 在此情況下繼續開發程式, 將會很容易發生 stack 與 bss/data 區塊資料重疊, 導致程式發生不可預期的錯誤發生. 因為目前使用的 Andes CPU N801 還沒有提供 stack protection 這個功能, 所以程式在 compile 也無法檔掉此問題, 使用者在開發程式時, SRAM 的記憶體規劃需特別注意.

4. HY16F 系列加密&解密功能說明

HY16F 系列產品，可透過 HY16F Writer 燒錄器，做晶片加密解密動作，可防止晶片內的程式碼被讀取出來。而 HY16F18x/HY16F19x/HY16F19xB/HY16F3981 系列產品的加密與解密行為是有些微的功能差異，以下將加強說明功能，下圖為 HY16F Writer 燒錄器加密操作畫面。



1. Encrypt with password 功能：(預設為 0xFFFFFFFF, 可自由設定密碼做保護, 即使輸入密碼正確 HY16F19xB/HY16F3981 會強迫清除部分的 Flash, 但 HY16F19x/HY16F18x 不會)

當設定 Encrypt with password 時候, 如果密碼輸入正確, 那 Code 會被清除, Code 的清除範圍依照當時設定密碼保護燒錄 APP 的範圍為基準. EX: 假設燒錄 APP 大小為 6KB 的 Flash, 選擇 Encrypt with password 輸入密碼 0x12345678. 當再一次要連線的時候, 會要求輸入密碼, 當輸入密碼正確, 此時 0x90000~0x91800 的範圍會被 Erase 掉變成 0xFF,

這時候密碼狀態會被解除為 0xFFFFFFFF, 輸入密碼正確清除 Flash, 這是 HY16F19xB/HY16F3981 的 IC 設計行為, 但是該行為在 HY16F19x/HY16F18x 不會發生.

對 HY16F19xB/HY16F3981 來說, 當選擇 Encrypt with password 有設定密碼的時候, 可以選擇輸入正確密碼當方式做連線與清除 Flash APP size 和密碼, 或著是暴力破解法(不需輸入密碼)選擇 FlashErase 來對全部的 64KB Flash 做清除動作

對 HY16F18x/HY16F19x 來說, 當選擇 Encrypt with password 有設定密碼的時候, 輸入密碼正確, 可以正常連線, 但是 Flash 不會被清除掉, 並且密碼狀態還是存在, 除非按下 Flash Erase 或著重新燒錄新的一組密碼, 才能解除密碼.但是如果密碼忘記,按下 Flash Erase, 還是無法清除 Flash,只有知道密碼的情況下才能做解除動作.

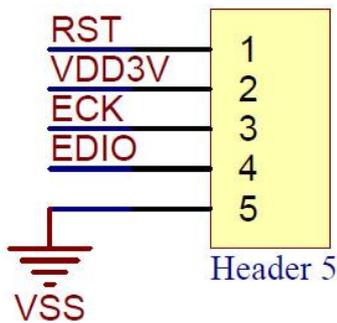
2. Encrypt without password 功能：設定之後, Flash 內容就無法被讀取出來, 只能使用 Flash Erase, 清除 64KB 的 Flash 都為 0xFF, 才能讓晶片回覆原始狀態. 這是為一解除密碼方法. 此功能, HY16F18x/HY16F19x/HY16F19xB/HY16F3981 都是相同的, 並沒有任何差異.

5. HY16F 系列硬體設計注意事項

紘康 HY16F 系列晶片是屬於 $\Sigma\Delta$ ADC+MCU 產品，具有高解析度與高精度能力，因具備最小 65nVrms 訊號的解析能力故對微小的漏電流與信號干擾特別敏感。所以在此介紹一些基本的晶片相關電路設計與 PCB 佈線觀念，讓使用者更能得心應手的開發產品。

5.1.1. 電路設計注意事項

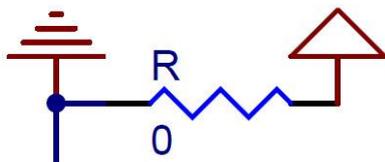
1. 電路設計規劃時，最好預留燒錄與 Debug 專用的 5pin Connector(2.54mm pitch)，腳位排列順序也要一樣，在產品開發與量產的時候，可以直接與 HY16F 燒錄器或著是 HY16F Mini Link 直接對接，方便開發與燒錄。



2. 當 ADC Output rate 小於 1k 時候，ADC 量測通道 AIO0-AIO1 與 AIO2-AIO3 等 ADC 輸入引腳外掛 100nF 電容，當 ADC Output rate 大於 1k 時候則外掛 10nF 電容，在兩個 ADC 量測通道輸入引腳之間外掛電容，可增強抗雜訊干擾

3. VDDA 連接 10uF 對地，VDD18 連接 1uF 對地，VPP 保持 NC.

4. 線路規劃時，類比地與數位地應該用不同符號標示，在類比地與數位地之間，設計一個 0 ohm 短路電阻。



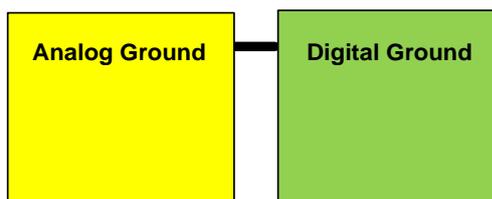
5.1.2. PCB Layout 注意事項

請務必遵守一般 PCB 線路佈局規則及知識。例如，直角走線會對 EMI 造成麻煩。使用 45 度轉角走線較佳。

5.1.2.1. 電源(Power)佈線：

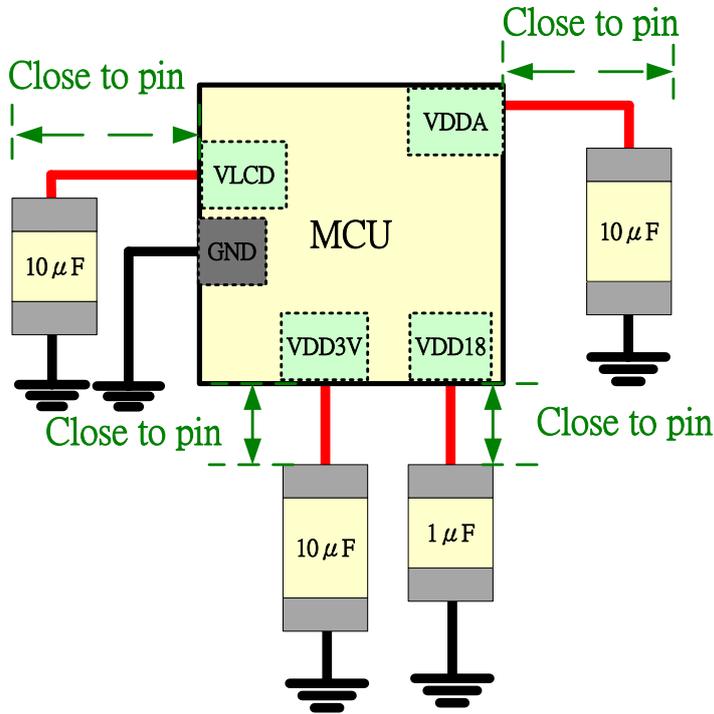
1. 類比訊號與數位訊號必須利用 VSS 鋪地隔開且獨立佈線：

此部分佈線不佳，最直接影響到的是當按鈕開關按下時，會有電流回路流進 VSS 而產生電位差(PCB 佈線電阻)。此電位差在高 ADC 解析度產品應用上容易造成測量誤差由幾個 ADC Count 至數十個 ADC Count，進而導致測量結果不正確。故建議佈線時類比與數位採獨立佈線，以避免測量誤差發生，同時為了增強抗干擾能力，兩個地線間用短路線連接，其短路位置最好靠近電池 VSS 位置做短路。如果 PCB 是雙面版，VSS 鋪地盡量多做過孔，以減少鋪地的內組。



2. 電源引線即佈線越短越好，越短即是阻抗越低越好：

引腳端的穩壓電容請 Layout 靠近晶片。如引腳：VDD, VDDA, VLCD, VDD18, REFO.

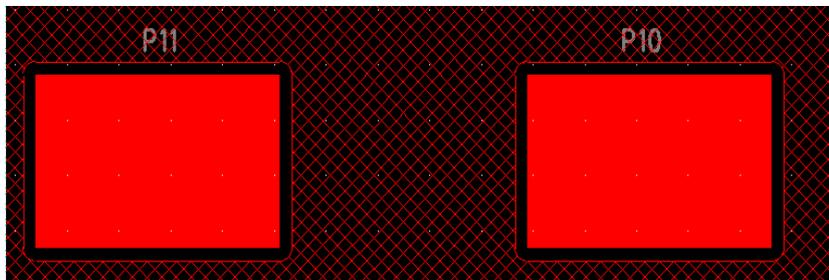


3. 鋪地銅:

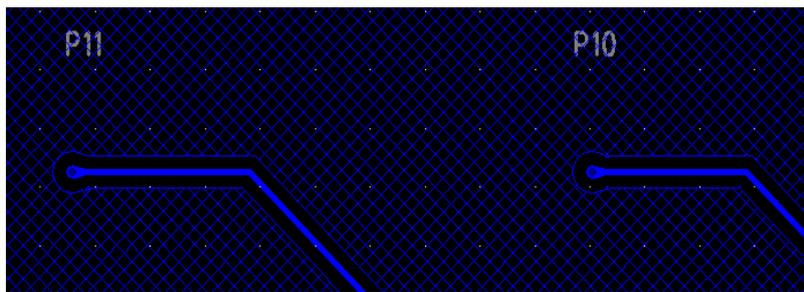
PCB 的頂層及底層都需要鋪地銅。將鋪地銅應用到線路將會增加抗擾度。我們定義鋪地銅到走線寬度為 5mil，線距為 20mil，格點距為 20mil，頂層及底層都有。由於 PCB 廠的策略，線寬必須設定為 10 mil 或更大。鋪地銅的線寬設定為 10 mil，線距為 20 mil，格點距為 40 以滿足限制。

請參考以下頂層及底層的圖示。

頂層的鋪地銅:

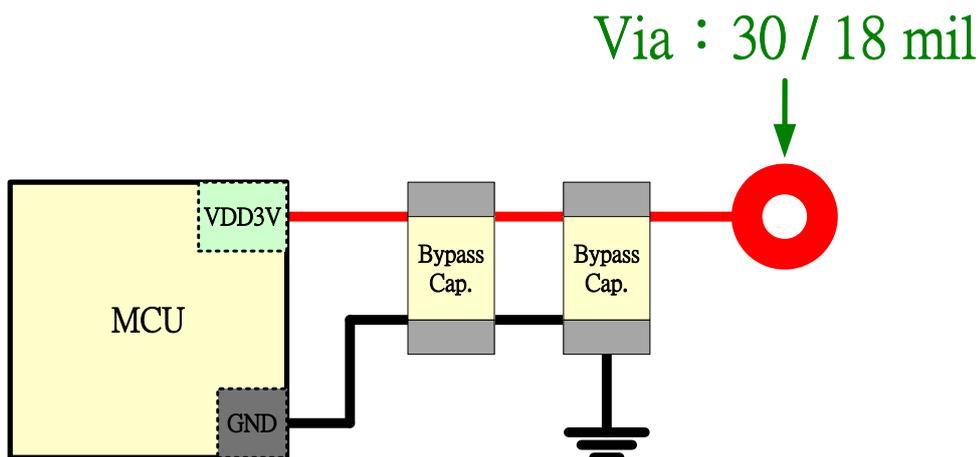


底層的鋪地銅：



4. VDD3V/GND 線路及電源供應貫穿孔:

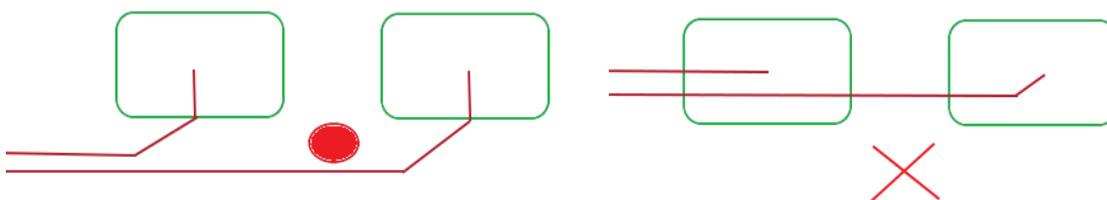
VDD3V/GND 建議使用 25~30mil . 電源供應的貫穿孔應該加到去偶電容. 建議如下圖示擺放以提供適當的去偶能力. 電源來源的雜訊會被去偶經由電容器到接地及流回電源來源. VDD3V/GND 使用的貫穿孔尺寸建議為 30/18mil 以減少貫穿孔結構的等效電感. 內部 LDO 旁路電容器線路(例如 : VDDA 與 REFO 輸出引腳), 內部 LDO 旁路電容器建議使用 10~20mil 寬的 PCB 線路以提供穩定的電源 VDD3V 貫穿孔洞及線路:



5. 雙層版 PCB 設計 :

在 2 層的 PCB 設計中, 建議使用 5~10mil 寬的線路來設計. 線路彼此間間距建議為兩倍的線寬 (建議為 10~20mil) 以避免干擾.

良好的與不良的線路走線圖如下範例 : 此範例中的 pad 代表 ADC sensor pin, 不能有走線跨到 Sensor pin 的情形發生.

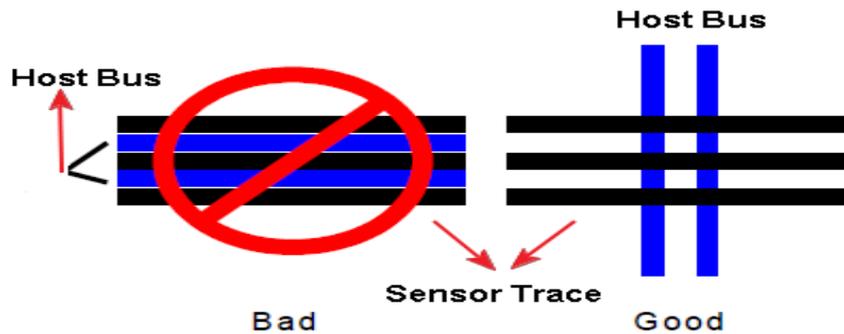


6. 介於主機及傳感控制器間的數位訊號線路 :

數位訊號是與主機間的通信線路。與其它傳感線路相比，切換的頻率比較高。時鐘及資料接腳建議以 5~10mil 寬的線路來設計數位訊號線路。時鐘及其它訊號 (電源，資料，控制，感測器線路)間間距建議至少為兩倍線寬 (應該為 10~20mil)。

禁止將這些數位訊號線路與類比線路平行走線。平行的結構會造成很大的串擾。如果不能避免數字訊號線路靠近感測器線路走線，建議交錯部分採 90 度直角走線，以減少串擾作用。

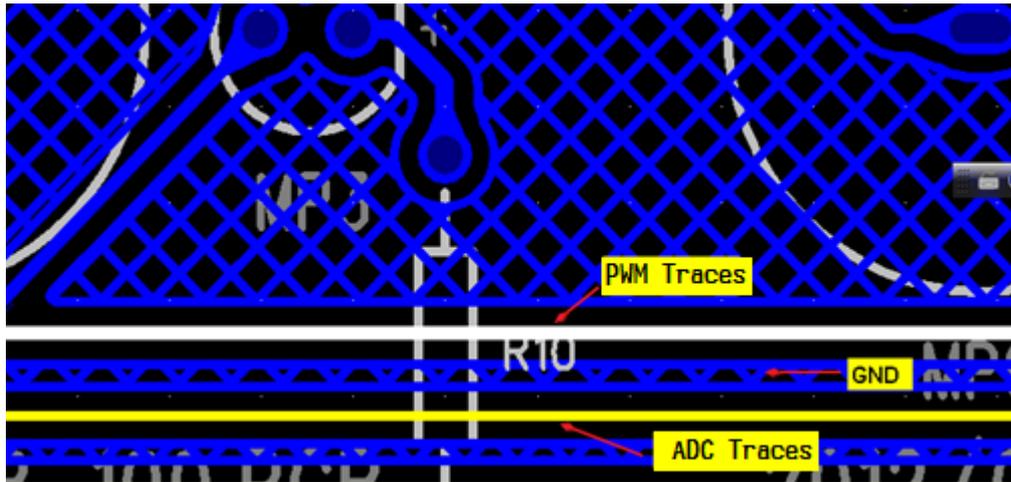
不良的及良好的主機匯流排及類比訊號線路走線：



7. 數位訊號(PWM)線路及類比訊號(ADC)線路：

請注意務必禁止將 PWM 的控制線路與 ADC 線路平行走線。請在數位控制線路及類比線路間插入鋪地銅，數位控制線路和類比線路之間若不插入鋪地銅，距離建議至少 50mil

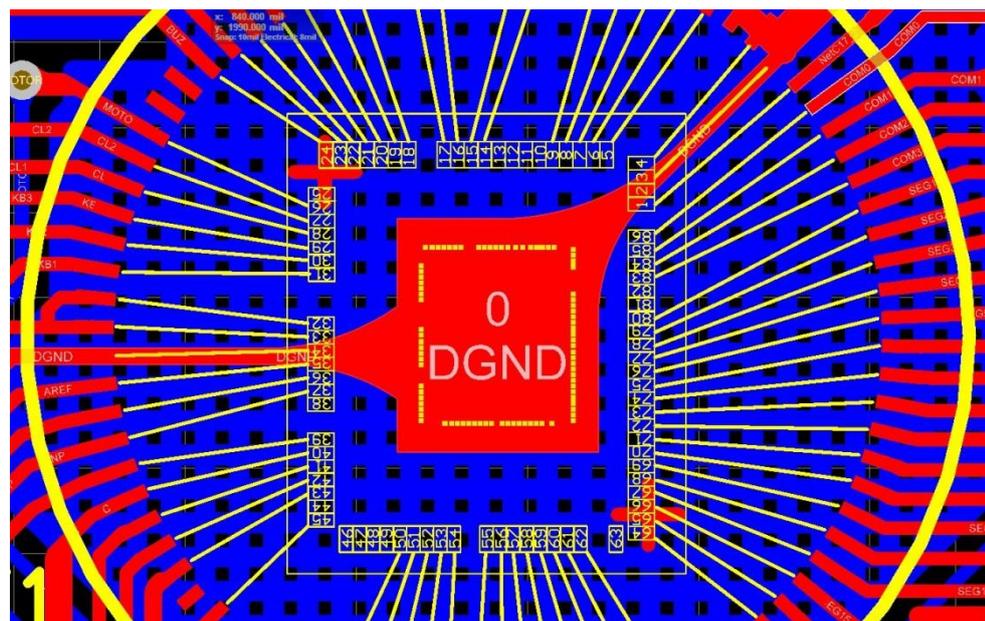
請注意：插入的地線不可只有一邊接地，否則會形成單極天線(Uni Polar Antenna)，在 EFT 干擾時會影響整體效能。PWM 線路及類比訊號線路之間的接地可參考下圖：



請注意：PWM 的控制線路如果與類比線路垂直交越，不得使用 SMT (表面貼件)元件，以避免垂直交越的距離太近，造成類比訊號受到干擾。需使用另面跳線方式，以保證 PWM 控制線路和類比線路保持在板厚 (1.6mm)的距離。

8. Dice 或著 COB 佈線：

置放晶片的金屬鋪面請連接至 DGND(Digital GND)，詳細連接方式，可參考下圖



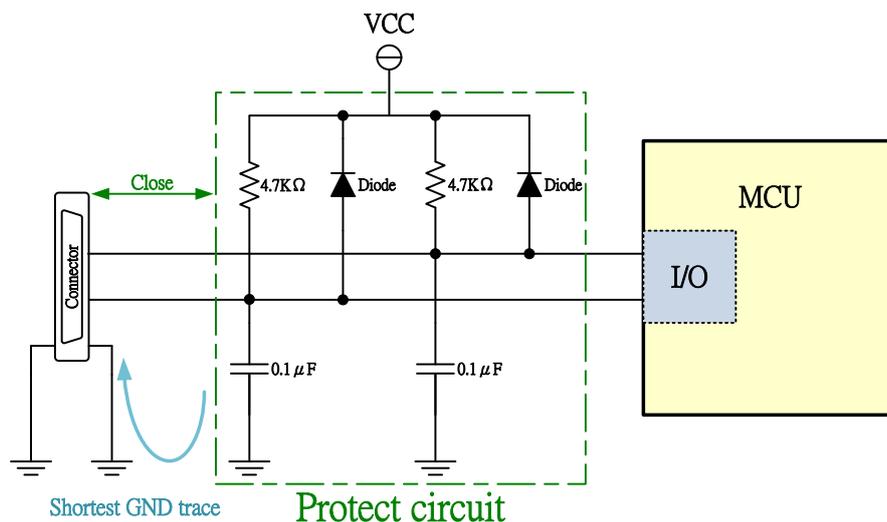
5.1.2.2. 類比(Analog)佈線：

1. 類比輸入 AIO0 與 AIO1, AIO2 與 AIO3 等等的輸入電容(0.1uF)越靠近引腳越好, 且外部訊號輸入時, 走線請先經過濾波電容後再引進到晶片類比輸入引腳.
2. AIOx 輸入引腳走線, 請平行跑線且越短越好, AIOx 引腳走線的上下板層避免有其它水平或垂直交叉走線.
3. AIOx 輸入引腳走線, 允許的情況下, 建議 AIOx 彼此腳間可鋪地線

5.1.2.3. 數位(Digital)佈線：

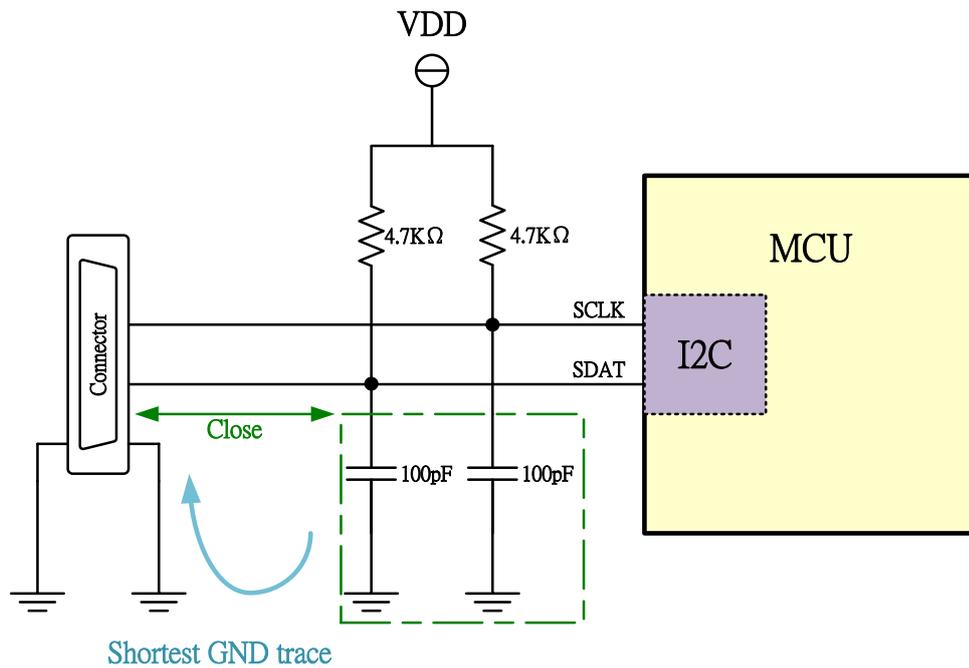
1. 外接 32768Hz 振盪器引腳(RTC), 跑線越短越好且必須利用鋪地隔離其它 I/O 產生的 AC 訊號, 以避免因訊號干擾而導致 RTC 產生的頻率受到影響.
2. 外接 32768Hz 振盪器引腳(RTC), 避免(AIOx 引腳)距離太近
3. 外接 32768Hz 振盪器引腳(RTC), 避免電源引腳 Layout 經過
4. I/O 端口:

使用任一 I/O 埠和主機或電源板溝通時, 需要加上保護電路. 保護電路需放在靠近 Connector 埠. 若是電容距離 connector 太遠則會失去保護效果. 請注意: 下地電容的地回流到電源板路徑必須愈短愈好.



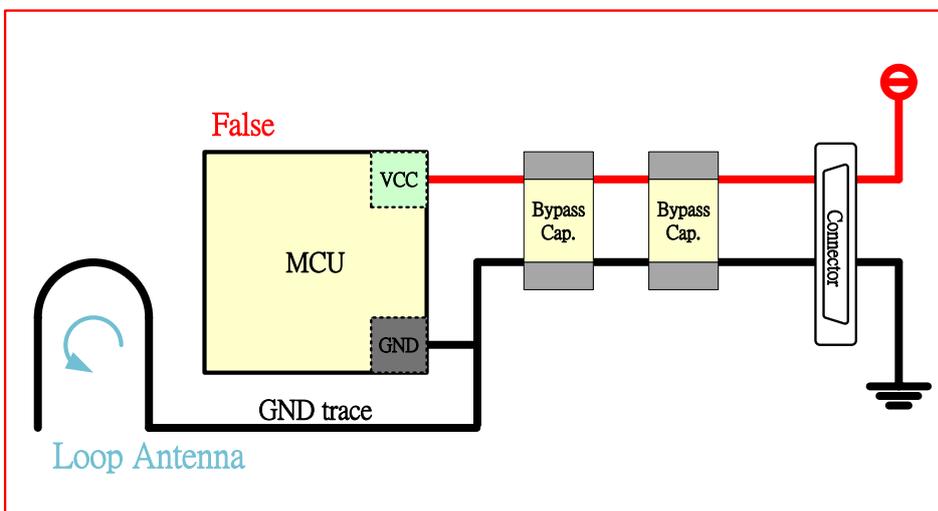
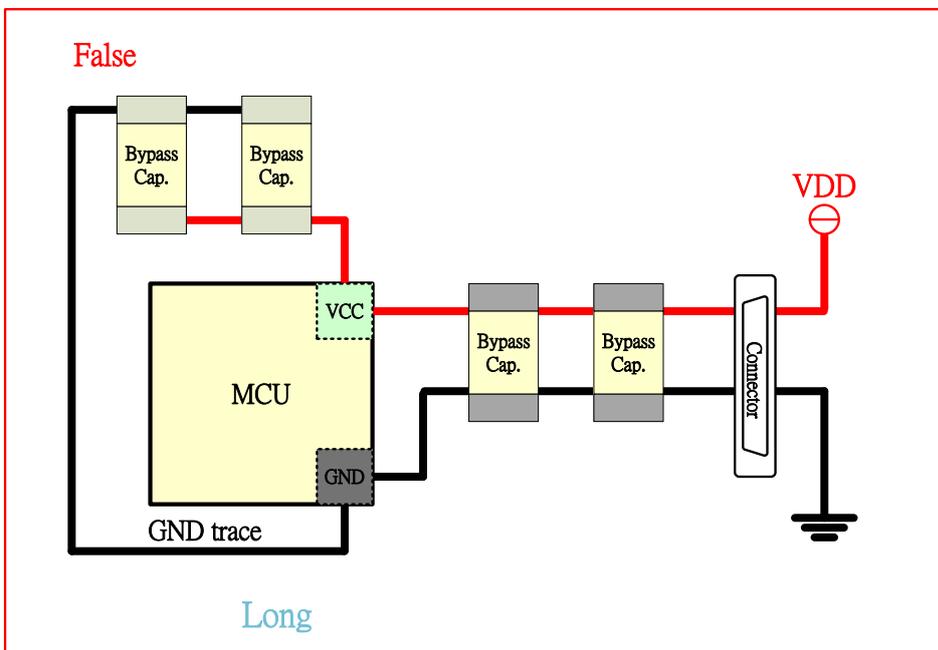
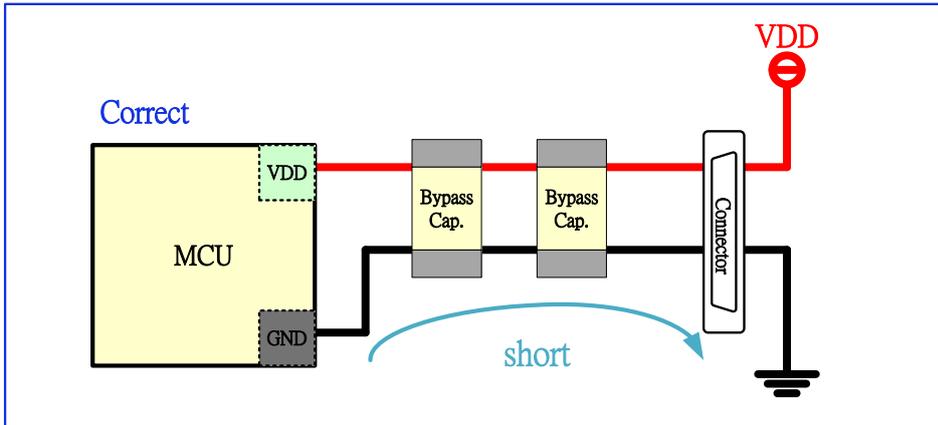
5. I2C 界面:

當使用 I2C 作為主機接口時, SCLK/SDAT 需要 4.7KΩ 上拉電阻以適應應適當的電子特性. 在不同的應用和信號負荷上可能會有微小的差異. 使用者可以使用在 PCB 上的外部上拉電阻. 外部使用小於 100 pF 的去偶電容, 可用來減少電源雜訊和資料處理雜訊(Noise). 去偶電容放置應靠近與主機端連接的接頭(Connector), 接地回路應儘量短, 直接回流到 Connector 端的地(GND).



6. IC 的地回流路徑：

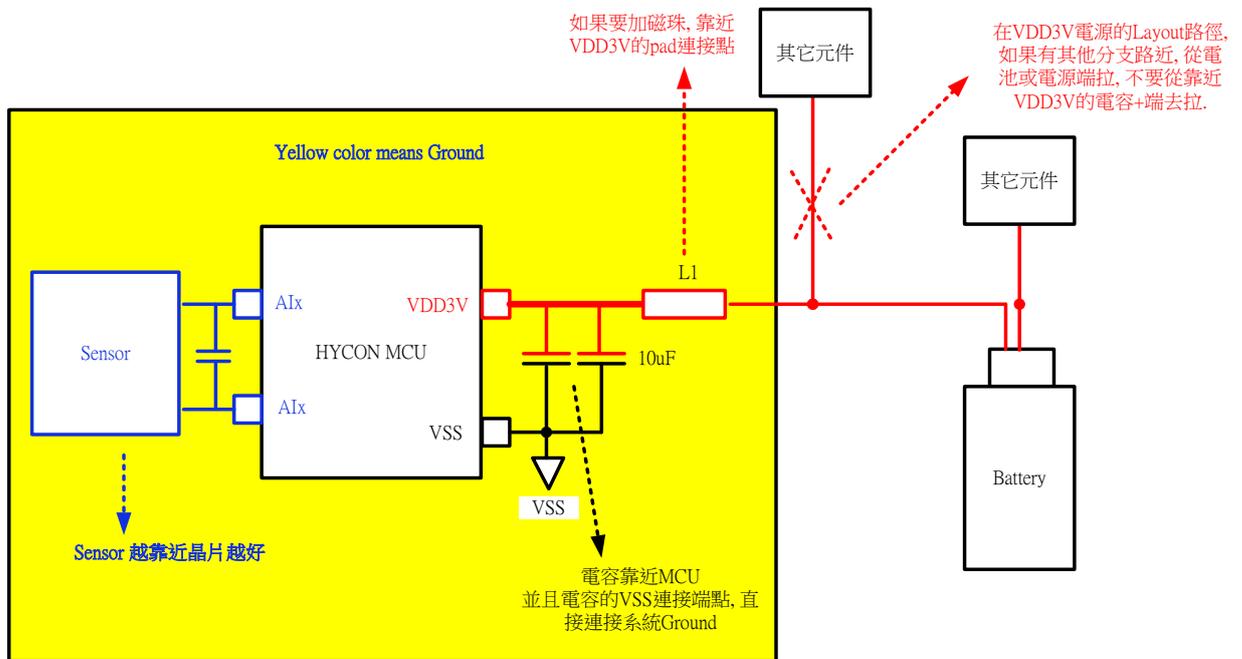
MCU 的地回流路徑，需經過 VDD/VDD18 濾波電容的接地點。在流回電源 connector 埠時的走線需儘量短，避免過多的繞線。也要避免成為 “Loop Antenna”。



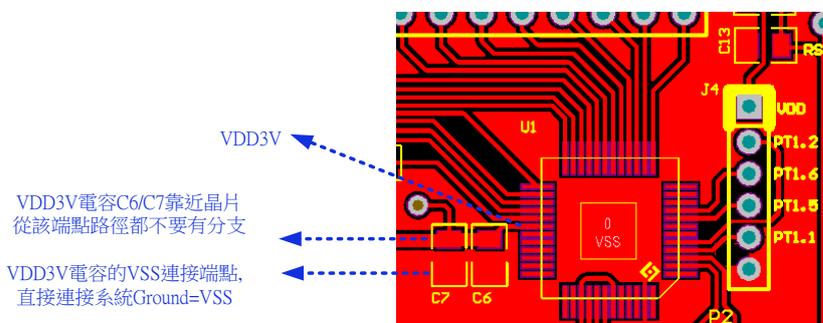
5.1.3. RS 防護設計注意事項

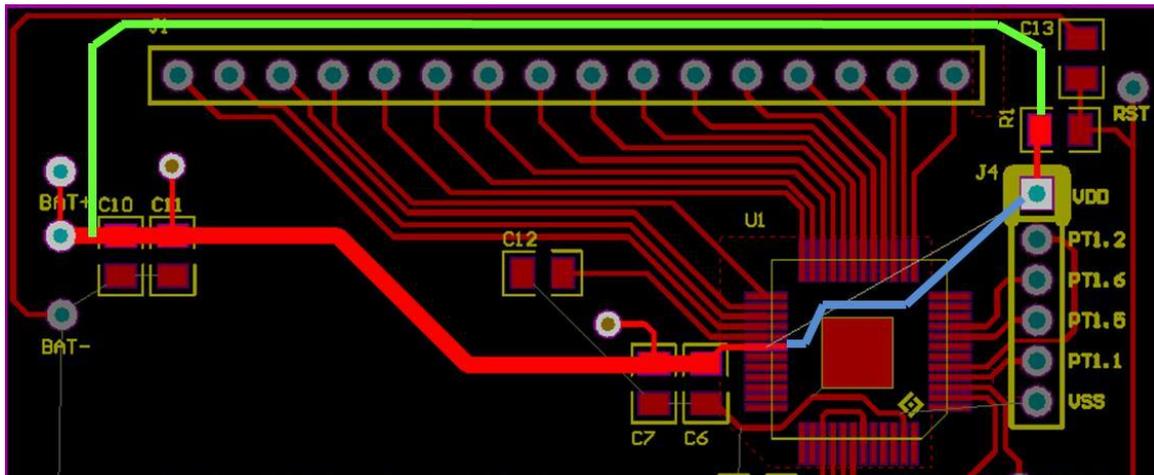
RS抗干擾, PCB Layout 設置建議教學：

- Sensor 與 Aix 間的走線別繞太遠或經過通信線
- Ground 鋪的面積要大.
- VDD power, 如果要加保護元件磁珠, 要先串磁珠後再接 VDD3V, 此 VDD3V 要先接 10uF 再接 104pF(0.1uF) 後才能接至 MCU 的 VDD3V pin.
- VDD3V 對地的電容要最短路徑, 並且VDD3V 對地電容的路徑, 電容兩端接點盡量不要再有分支路徑. 如果有其他分支路徑需求, 可參考以下示意圖.



VDD3V 的對地電容連接示意圖, 如下



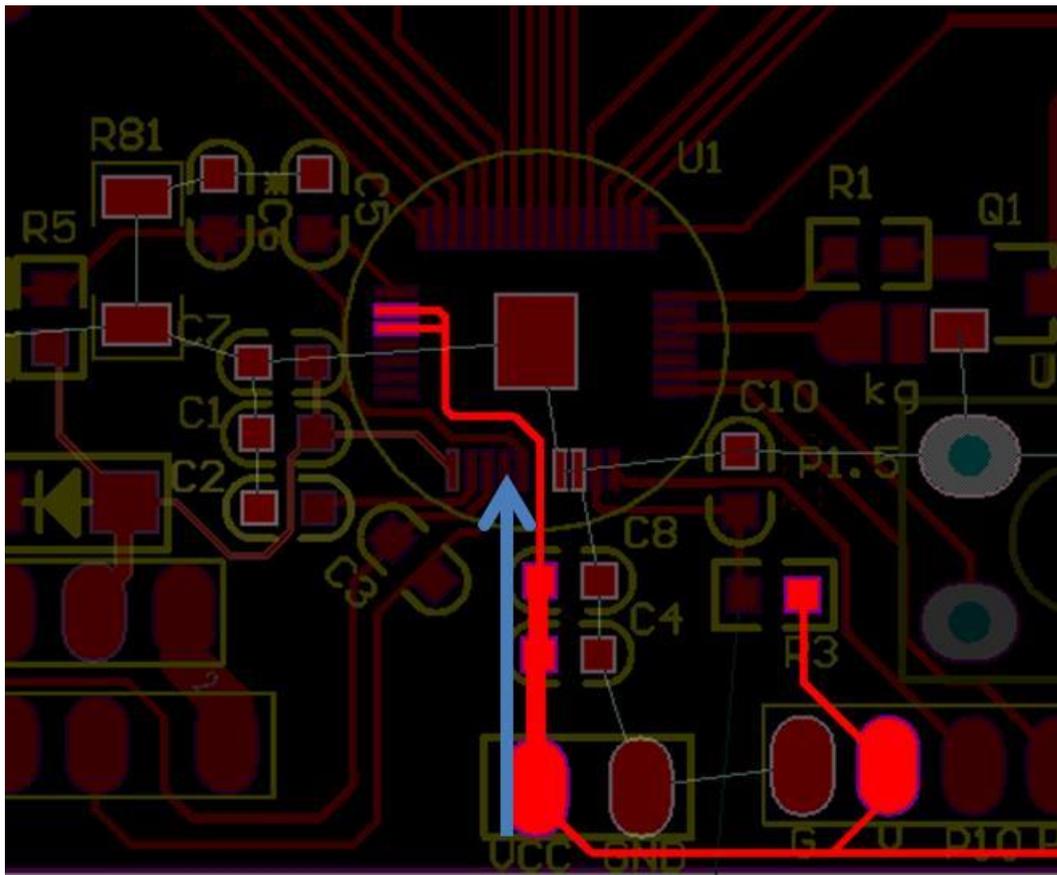


上圖範例，可注意兩個地方

注意 1：VDD 電容選用 C7、C6 是正確的，C10、C11 為錯誤示範

注意 2：晶片連線到 VDD 有藍線與綠線兩條路徑，綠線是正確，藍線為不良示範

當外部電源直接從 VDD pad 灌入，藍線路徑等同晶片的 VDD3V pin 腳是先直接連接到 VDD pad，繞很遠才經過 C10、C11 電容，最後才到 C6、C7 電容。



上圖範例，可注意 1 個地方

注意 1：Power 進到 PCB 後，經 C4、C8 後就到 IC 的 VDD3V pin，此為正確示範

- Layout 避險環形天線效應產生

圖 1

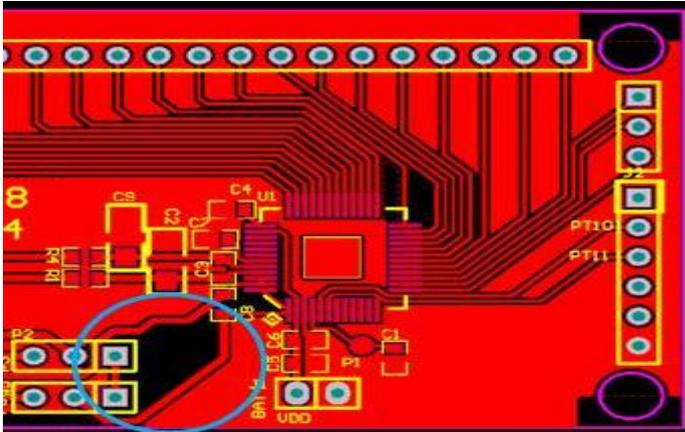
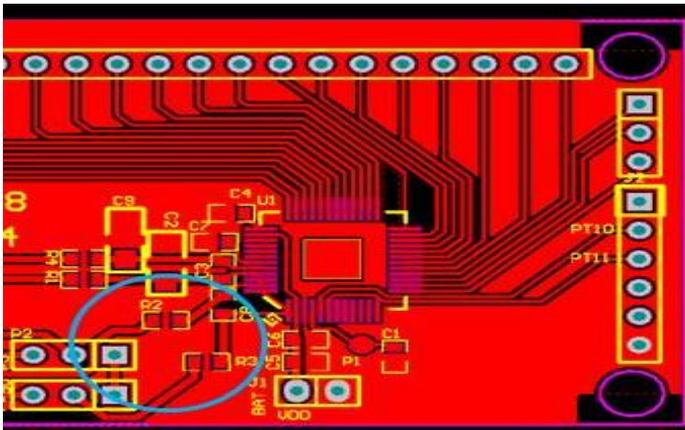


圖 2

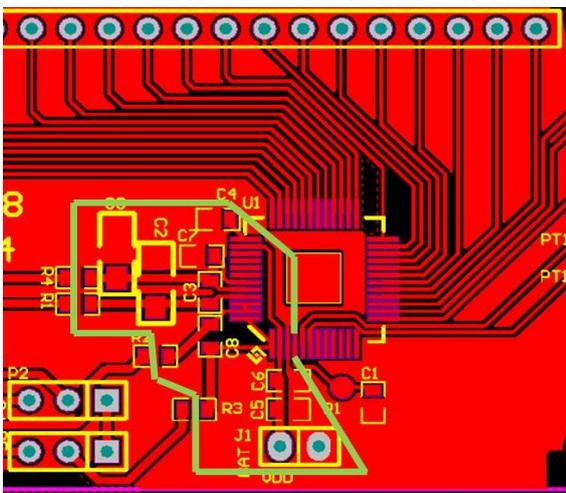


上圖 1 與圖 2 範例，此兩圖為相似的 PCB，但差一點在 IC 前是否有串連兩個電阻(R2、R3)

圖 1 可以避免環形天線效應，圖 1 為正確示範

圖 2 會有環形天線效應，圖 2，從 R2、R3 電阻中間可以連到 GND，但是會導致環形天線效應產生

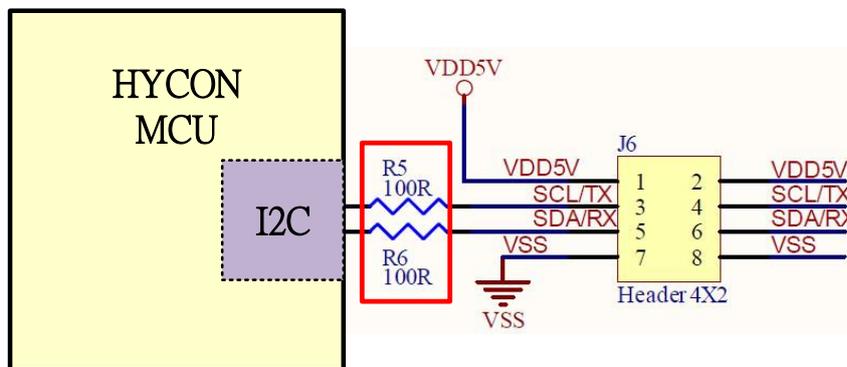
下圖有串電阻的 PCB，VSS 產生了環形天線，路徑如綠色線路描出



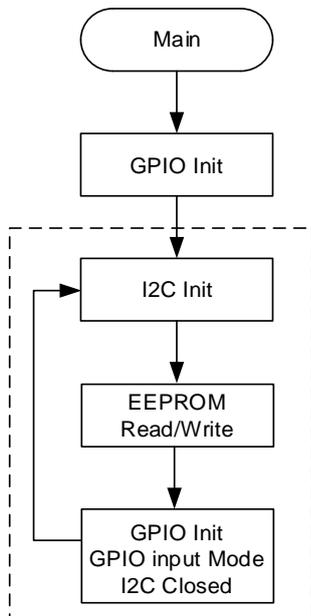
5.1.4. ESD 防護設計注意事項

ESD 的防護與機構設計與 PCB Layout 走線方式都有關聯，但是從 HY16F 的韌體做設定，可以部份改善與強化 ESD 能力。

1. HY16F 主晶片的工作頻率設置為 HAO/2 會比 HAO/1 抗 ESD 雜訊來的好。
2. 沒有使用到的 GPIO pin, 最好都設定為 output low
3. 如果應用中有使用到 I2C Port, 可以著手從 I2C 硬體與軟體設定做 ESD 防護強化
 硬體：SCL 與 SDA 都串上 100 ohm 限流電阻(線流電阻加在靠近 HY16F 主晶片端)



軟體：I2C 程式部分修改為，在每一次的讀取/寫入完成 EEPROM 之後就把 I2C 的 IP Closed, 讓 I/O 狀態為 GPIO Input Mode, 只有在每一次讀取/寫入 EEPROM 之前，才會再重做 I2C initial 動作



5.1.5. COB 打線注意事項

生產機台在實際打線時，第一隻打線的接腳應為 VSSESD，接下來才是打 VSS 這隻接腳，先打 VSSESD，可以避免生產時候所造成的 ESD 擊穿等晶片損壞問題。

- | | |
|---|--------|
| 1 | VLCD |
| 2 | NC |
| 3 | VSS |
| 4 | VSSESD |

6. HY16F 系列軟體設計注意事項

此章節提供 HY16F 在軟體&韌體設計時需要注意的事項。

6.1. HY16F 的 VDD3V 上電穩壓判斷

為了確保 HY16F 更穩定的工作，在軟體設計端，建議針對 HY16F 的 VDD3V 加入電壓穩壓的判斷(即 main 的第一行)，確保 VDD3V 穩定在系統的工作電壓之後(建議 VDD3V=2.75V 以上)，程式再繼續往下運行。以下將提供範例，利用 HY16F198B 的 CMP 比較器來做 VDD3V 電壓穩壓判斷。

程式碼說明：晶片上電即利用函式 VDD3V_2_75_Check()來判斷 VDD3V 是否大於 2.75V(利用變數 _timeout, 最多等待 VDD3V 上電的穩壓時間約 8 秒鐘)。如果 VDD3V 小於 2.75V, LCD 顯示 0, 程式 while(1) 卡死。如果 VDD3V 大於 2.75V, LCD 顯示 1, 程式可繼續往下執行。以下為程式碼提供參考：

```
int main(void)
{
    unsigned char x;
    unsigned int x_timeout=5000;
    //8S timeout
    while(x_timeout--)
    {
        x=VDD3V_2_75_Check();
        if(x==1)
        {
            x_timeout=0;
        }
    }
    if(x==0)
    {
        DisplayInit();
        ClearLCDframe();
        LCD_DATA_DISPLAY(x); //if x=0 means failure
        while(1);
    }
    if(x==1)
    {
        DisplayInit();
        ClearLCDframe();
        LCD_DATA_DISPLAY(x); //if x=1 means success
    }
    return 0;
}
```

```
unsigned int VDD3V_2_75_Check()
{
    unsigned char i=0;
```

```
unsigned char j=0;
unsigned char k=0;
outw(0x40300,0x00002101); //ENHAO + HAO=2M
outw(0x41104,0x10000000); //select precise voltage
outw(0x40400,0x03031010); //ENLDO
//outw(0x41804,0x08083333); //CMP, 0808 means VDD3V>2.4V
outw(0x41804,0x07073333); //CMP, 0808 means VDD3V>2.75V
outw(0x41800,0x01010303); //CMP
outw(0x41808,0x01011212); //CMP
//Delay a while for CMP stable
for(j=0;j<200;j++)
{
    asm("nop");
}
for(j=0;j<10;j++) //compare 10 times
{
    i=((inw(0x41800)&(1<<16))>>16); //read CMPO
    if(i==0)
    {
        k++; //VDD3V > goal voltage
    }
    if(i==1)
    {
        //VDD3V < goal voltage
    }
}
if(k>=5) //that means VDD3V is stable >=2.4V
{
    return 1;
}
else
{
    return 0;
}
}
```

7. Revision History

Version	Page	Date	Revision Summary
V01	All	2018/11/05	First edition
V02	All	2022/06/07	新增章節 1: 前言 新增章節 6 : HY16F 系列軟體設計注意事項 新增章節 6.1 : HY16F 的 VDD3V 上電穩壓判斷 Flash 章節 : 修正 Flash Read/Write 使用壽命的內容描述.
V03	All	2023/03/21	新增章節 2.13 : ECK/EDIO, 主要說明該引腳在硬體連接需 要注意的事項. 新增章節 5.1.5 : COB 打線注意事項 新增章節 2.7.2 : Flash 資料變異 –建議與改善方針