



HY16F 系列

产品应用注意说明书

HY16F Series Application Note

Table of Contents

1. 前言	5
2. HY16F 系列 IP 使用注意事项	6
2.1. WDT	6
2.1.1. WDT Reset-使用注意事项	6
2.1.2. WDT Reset-范例程序	6
2.2. GPIO	7
2.2.1. GPIO 复用引脚 PT3.6/PT3.7-使用注意事项	7
2.2.2. GPIO 复用引脚 PT3.6/PT3.7-范例程序	7
2.3. 模拟电源 VDDA, VDD18, VPP	8
2.3.1. VDDA -使用注意事项	8
2.3.2. VDD18-使用注意事项	8
2.3.3. VPP-使用注意事项	10
2.4. ADC.....	11
2.4.1. ADC 通道切换-使用注意事项.....	11
2.4.2. ADC 通道切换-范例使用说明.....	11
2.4.3. ADC 启动时间稳定问题	12
2.4.4. ADC 线性度问题	12
2.4.5. ADC-IA 使用说明	13
2.5. Power 功耗.....	14
2.5.1. Active Mode --> Idle Mode/Sleep Mode-使用注意事项	14
2.5.2. Active Mode --> Idle Mode/Sleep Mode-范例程序	14
2.5.3. Idle Mode/Sleep Mode --> Active mode-使用注意事项	15
2.5.4. Idle Mode/Sleep Mode --> Active mode-范例程序	15
2.5.5. Idle Mode/Sleep Mode --> Active mode- GPIO 唤醒使用注意事项	16
2.6. 内部高频 HAO & 内部低频 LPO	17
2.6.1. 内部高频 DrvCLOCK_CalibrateHAO -使用注意事项.....	17
2.6.2. 内部高频 DrvCLOCK_CalibrateHAO -范例程序	17
2.6.3. 内部高频 HAO & 内部低频 LPO -使用注意事项.....	17
2.6.4. 内部高频 HAO =16MHz -使用注意事项	18
2.7. Flash	19
2.7.1. Flash Read/Write -使用注意事项	19

2.8.	LCD	20
2.8.1.	DrvLCD_VLCDTrim–使用注意事项	20
2.8.2.	DrvLCD_VLCDTrim–范例程序.....	20
2.8.3.	LCD 频率 LCK –使用注意事项.....	21
2.9.	实时时钟 RTC.....	22
2.9.1.	RTC –使用注意事项.....	22
2.10.	定时计数器 Timer	23
2.10.1.	TimerA –使用注意事项.....	23
2.11.	DAC.....	24
2.12.	R2ROP	25
2.12.1.	R2ROP 当 unit gain buffer –使用注意事项	25
3.	HY16F 系列开发工具应用说明	26
3.1.1.	AndeSightRDS V2.0.1 转换到 AndeSightRDS V2.1.1	26
3.1.2.	Stack 占多少 SRAM 空间问题	30
4.	HY16F 系列加密&解密功能说明.....	31
5.	HY16F 系列硬件设计注意事项	32
5.1.1.	电路设计注意事项.....	32
5.1.2.	PCB Layout 注意事项.....	33
5.1.3.	RS 防护设计注意事项.....	41
5.1.4.	ESD 防护设计注意事项	44
6.	HY16F 系列软件设计注意事项	45
6.1.	HY16F 的 VDD3V 上电稳压判断	45
7.	REVISION HISTORY	48

注意：

- 1、本说明书中的内容，随着产品的改进，有可能不经过预告而更改。请客户及时到本公司网站下载更新 <http://www.hycontek.com>。
- 2、本规格书中的图形、应用电路等，因第三方工业所有权引发的问题，本公司不承担其责任。
- 3、本产品单独应用的情况下，本公司保证它的性能、典型应用和功能符合说明书中的条件。当使用在客户的产品或设备中，以上条件我们不作保证，建议客户做充分的评估和测试。
- 4、请注意输入电压、输出电压、负载电流的使用条件，使 IC 内的功耗不超过封装的容许功耗。对于客户在超出说明书中规定额定值使用产品，即使是瞬间的使用，由此所造成的损失，本公司不承担任何责任。
- 5、本产品虽内置防静电保护电路，但请不要施加超过保护电路性能的过大静电。
- 6、本规格书中的产品，未经书面许可，不可使用在要求高可靠性的电路中。例如健康医疗器械、防灾器械、车辆器械、车载器械及航空器械等对人体产生影响的器械或装置，不得作为其部件使用。
- 7、本公司一直致力于提高产品的质量和可靠度，但所有的半导体产品都有一定的失效概率，这些失效概率可能会导致一些人身事故、火灾事故等。当设计产品时，请充分留意冗余设计并采用安全指标，这样可以避免事故的发生。
- 8、本规格书中内容，未经本公司许可，严禁用于其它目的之转载或复制。

1. 前言

本文—产品应用注意说明适用于以下纭康 HY16F 系列产品：

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

2. HY16F 系列 IP 使用注意事项

在此章节特别针对 HY16F 系列在特定 IP 使用上，说明使用注意事项。 如果使用该 IP 做应用，但没注意到 IP 的使用限制，则将可能发生不可预期的结果。使用者在使用该 IP 之前，应花时间特别研读过后，再设计使用，可减少许多开发上的错误用法。

2.1. WDT

看门狗 WDT—顾名思义的为芯片的看守者，主要用于产生唤醒事件，或芯片发生不可预期当机后，能进行基础自动重置功能。在此特别强调说明 WDT Reset 功能使用注意事项。

2.1.1. WDT Reset-使用注意事项

当WDT Reset Mode功能开启之后，就无法再切换回WDT Timer Mode，即设置WDNMI 0x40108[6]=1b之后，就无法再设置回WDNMI 0x40108[6]=0b；使用者在设置与应用此功能之前，需要先知道WDT Reset Mode有此应用限制。此外，程序设计者也需注意到WDT功能开启之后，WDT Count—WDTO 0x40108[30:16]就会开始计数，这也是无法关闭的，即设置ENWDT 0x40108[4]=1b，就无法再设置回ENWDT 0x40108[4]=0b。所以如果开启设置WDNMI 0x40108[6]=1b与ENWDT 0x40108[4]=1b之后，记得要在预期规划时间内做WDT计数器归零动作（俗称喂狗），即设置0x40108[5]=1b，以避免程序开发过程忘记清除WDT计数器，造成WDT Reset事件发生计数溢出，而产生IC 程序复位动作。

注意：当 WDT Reset 发生之后，各 IP 的寄存器状态会维持原状，即 WDT Reset 前与 WDT Reset 之后 IP 寄存器状态一样。WDT Reset 发生之后，程序只会先跳回到 Reset 向量位置，之后再往下执行 Startup code 与 CPU 初始化等相关动作，但 IP 寄存器状态不会重新初始化。这与实际对 HY16F 硬件 Reset pin 做 Reset 动作，产生的执行效果是有差异的。

2.1.2. WDT Reset-范例程序

```
DrvWDT_Open(E_IRQ , E_PRE_SCALER_D32); //设置看门狗(WDT)为 IRQ mode 及 CLK / 32  
DrvWDT_ResetEnable(); // WDT 中断工作模式选择为 Reset Mode
```

2.2. GPIO

GPIO 口都具有多个复用功能，但是一般同一时间只能设置某个功能有效，所以在不需用到复用功能时，请记得关闭复用功能。PT3.6/PT3.7 这两个复用 GPIO 在使用上需特别注意，如果同时开启模拟与数字功能的时候，将可能会发生 VDDA 电源倒灌并且造成 ADC 量测异常等问题。

2.2.1. GPIO 复用引脚 PT3.6/PT3.7-使用注意事项

PT3.7/OPO 复用引脚:

- 设计为 OPO output ; 控制寄存器 OPOE 0x41900[1]=1b, PT3PU7 =PT3OE7=PT3IE7=0b
- 设计为 PT3.7 GPIO input ; 控制寄存器 ENRFO 0x40400[1]=0b, PT3IE7=1b
- 设计为 PT3.7 GPIO output ; 控制寄存器 ENRFO 0x40400[1]=0b, PT3IE7=1b(即使不作为输入也必须强制设定), PT3OE7=1b

PT3.6/REFO 复用引脚 :

- 设计为 REFO output ; 控制寄存器 ENRFO 0x40400[1]=1b, PT3PU6=PT3OE6=PT3IE6=0b
- 设计为 REFO input ; 控制寄存器 ENRFO 0x40400[1]=0b, PT3PU6=PT3OE6=PT3IE6=0b
- 设计为 PT3.6 GPIO input ; 控制寄存器 ENRFO 0x40400[1]=0b, PT3IE6=1b
- 设计为 PT3.6 GPIO output ; 控制寄存器 ENRFO 0x40400[1]=0b, PT3IE6=1b(即使不作为输入也必须强制设定), PT3OE6=1b

2.2.2. GPIO 复用引脚 PT3.6/PT3.7-范例程序

使用者设计 PT3.6 与 PT3.7 当做 GPIO Output 功能时候, 则 PT3.6 与 PT3.7 程序的设置初始化流程如下 :

```
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. 模拟电源 VDDA, VDD18, VPP

使用者在设计 HY16F 系列线路图的时候, 应该要特别注意到 VDDA/VDD18/VPP 这三个脚位的连接。

2.3.1. VDDA -使用注意事项

VDDA 的对地滤波电容可连接范围在 1uF~10uF, 但使用时需注意到 VDDA 开启时需要有等待稳定时间, 电容越大, VDDA 需要等待的稳定时间也越久, VDDA 的外接电容大小也与负载和电流驱动能力有关。

当 VDDA 接入一个 1uF 对地电容时, VDDA 的稳定时间至少需要大于 0.5ms, 当 VDDA 接入一个 10uF 对地电容时, VDDA 的稳定时间至少需要大于 1ms。若在 VDDA 未稳定情况下启动 ADC 功能, 将会造成读取到错误的 ADC 数据。

VDDA 外接电容的大小主要与驱动电流的能力有关, 依 HY16F 系列芯片的设计要求如下:

VDDA 驱动能力要求为 1mA 时: 外接的对地电容要为 1uF,

VDDA 驱动能力要求为 10mA 时: 外接的对地电容要为 10uF

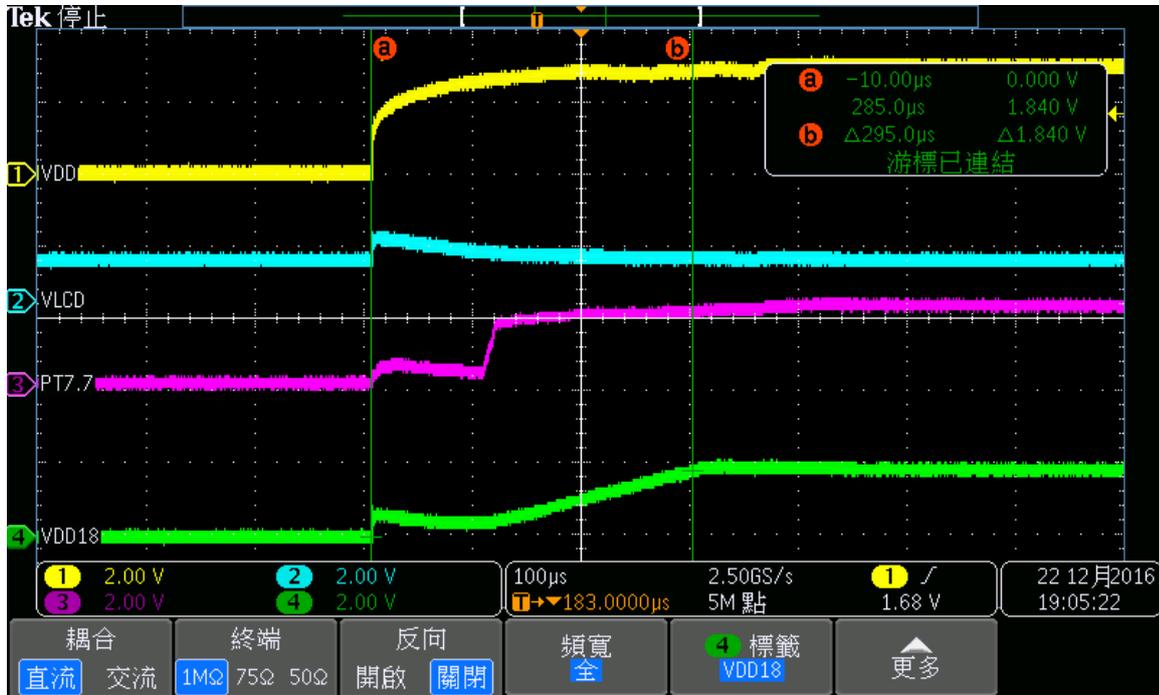
建议使用者在 VDDA 对地电容统一为 10uF, 以增强驱动能力及电压的稳定度。

注意: 因 VDDA 电容大小与驱动能力及电压的稳定度有关, 建议电容的选择为 10uF-Y5R or 10uF-X5R 规格, 电容温度与电容量变化在 15%以内的误差规格。

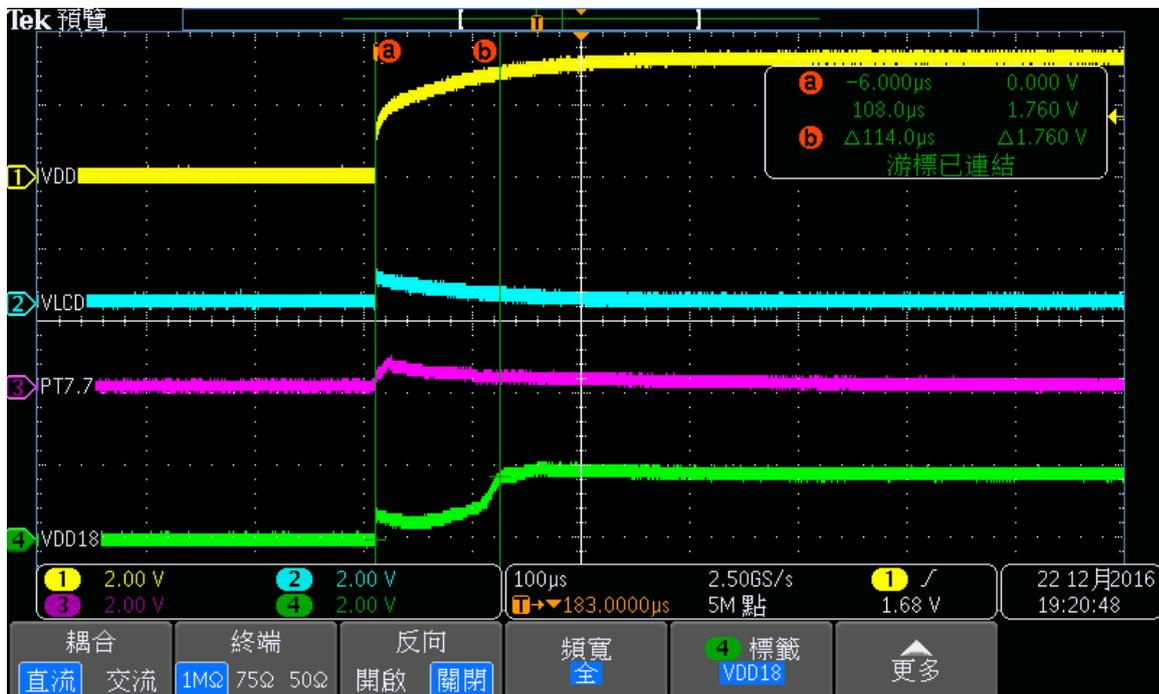
2.3.2. VDD18-使用注意事项

VDD18 的对地电容只能连接 1uF。如果 VDD18 的电容连接太大, 则可能会造成上电时候产生 LCD 显示残影问题。

如下图为例：VDD18 使用 10uF 电容图形，因为 VDD18 从 Power On 到稳定的时间较长，约为 295us，观察 SEG 脚位 PT7.7，因为控制逻辑尚未稳定，PT7.7 已经被异常拉起，导致 LCD 残影现象发生。



如下图为例：VDD18 使用 1uF 电容图形，因为 VDD18 从 Power On 到稳定的时间约为 114us，观察 SEG 脚位 PT7.7，因为控制逻辑已经稳定，PT7.7 没有被异常拉起，不会有 LCD 残影现象发生。



2.3.3. VPP-使用注意事项

当 VPP 引脚连接 0.1uF 对地电容的时候, 在使用 Andesight IDE 做在线烧录或使用 HY16F WRITER 烧录器做烧录, 都会显示烧录失败讯息。所以 VPP 引脚不能连接任何电容, 也不能做任何负载电压输出应用, VPP 引脚应该维持保留 NC 状态(保持空接状态)。

2.4. ADC

ADC 做通道切换的时候(EX:从 AIO0-AIO1 切换到 AIO2-AIO3) ,应该要注意到 ,需要做 ADC Combfilter(梳状滤波器)设置, 不然可能会造成 ADC Offset 停在不同的状态, 进而发生量测结果漂移的现象。

2.4.1. ADC 通道切换-使用注意事项

当使用者做 ADC 通道切换时, 因量测的讯号不同, 所以前两笔的 ADC 讯号是无效的。使用者可利用 ADC CombFilter(梳状滤波器)复位(CFRST=1b)控制来达到自动丢弃前两笔, 第三笔才会产生 ADC 中断来读取 ADC 数据的动作。如果实际应用中, 没有切换 ADC input channel 的需求, 量测通道都是一样的, 就可不需做复位 CFRST=1b 动作。

2.4.2. ADC 通道切换-范例使用说明

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

comb filter reset,

sample A0, 硬件自行丢弃

sample A1, 硬件自行丢弃

sample A2, 从硬件看到的第一笔中断输出值

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

comb filter reset,

sample B0, 硬件自行丢弃

sample B1, 硬件自行丢弃

sample B2, 从硬件看到的第一笔中断输出值

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

comb filter reset,

sample A3, 硬件自行丢弃

sample A4, 硬件自行丢弃

sample A5, 从硬件看到的第一笔中断输出值

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

comb filter reset,

sample B3, 硬件自行丢弃
sample B4, 硬件自行丢弃
sample B5, 从硬件看到的第一笔中断输出值

Change ADINP,ADINN to A (EX : AIO0-AIO1)
comb filter reset,
sample A6, 硬件自行丢弃
sample A7, 硬件自行丢弃
sample A8, 从硬件看到的第一笔中断输出值

Change ADINP,ADINN to B (EX : AIO2-AIO3)
comb filter reset,
sample B6, 硬件自行丢弃
sample B7, 硬件自行丢弃
sample B8, 从硬件看到的第一笔中断输出值

当 ADC OSR=32768, ADC Clock=327Khz, 则 ADC Output rate 约为 $10\text{sps}=0.1\text{sec}$ 。

因为 ADC 3rd (3 阶) comb filter 架构, 所以一开始测量时, 需要丢弃前两笔数值, 第三笔数值才是稳定状态, 也就是说每次切换通道之后的第一笔 ADC 稳定数值输出时间为 $3/10=0.3\text{sec}$ 。但如果 ADC 通道没有做切换的时候, 因为不需再做 CFRST=1b 动作, 所以第二笔, 第三笔直到第 N 笔数据的 ADC 稳定输出时间都为 0.1sec。

2.4.3. ADC 启动时间稳定问题

每一次的 VDDA 做 on/off 切换都需要等待稳定时间。当 VDDA 没有任何负载的时候, 并且在 ADC 是外部输入短路 (external input short) 情况下, 不会有 ADC 启动稳定时间的问题。而当 VDDA 有负载与 ADC 有输入阻抗的时候, 会观察到 ADC 前面几笔数据不稳定的问题。因此, 在实际应用中, 如果有做 VDDA on/off 切换, 当 VDDA on 的时候, 需要先等待 100ms 的稳定时间, 再做 ADC 数据的读取, 可以避免 ADC 前面几笔数据不稳定问题发生。

2.4.4. ADC 线性度问题

HY16F系列中, ADC的最佳可量测电压范围是会随着参考电压的设置有关, 使用者在使用时候应注意, 避免ADC线性误差问题发生。

以下为使用注意事项 :

当REFP-REFN(VREF) = 1*VDDA时,

则VINP-VINN的差动输入信号不能大于 $1/2 \cdot VDDA$ ，否则会有线性度问题。

当REFP-REFN(VREF) = $1/2 \cdot VDDA$ 时，

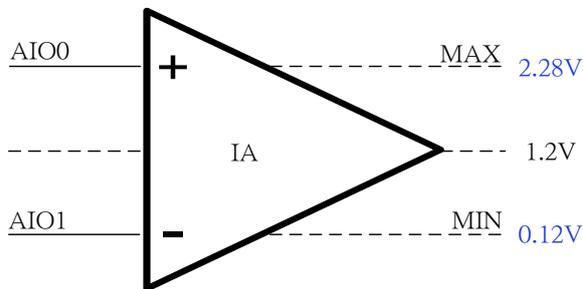
则 VINP-VINN 的差动输入信号不能大于 $0.9 \cdot VREF$ ，否则会有线性度问题。

举例：设置REFP-REFN=VDDA-VSSA，并且VDDA=2.4V，此时设置FRb=0b，满量程参考电压输入，即VREF*1；这时候ADC最大可量测的电压Vin范围是 $1/2 \cdot VDDA=1.2V$ ，并非是 $0.9 \cdot VREF=2.16V$ ；因此，在此设置条件下，如果量测电压Vin是超出1.2V以上，就会有ADC线性误差问题发生。

2.4.5. ADC-IA 使用说明

HY16F 系列中，只有HY16F3981在ADC前端具有IA(仪表放大器功能)，在使用仪表放大器需要注意到IA的量测输入信号是有输入范围 (IA Input range) 使用限制，输入范围为BIAS=1.2V, input signal=+/-1.08V。在实际使用上，可这样设计使用，ADC量测通道设置为AIO0-AIO1，在ADC的输入负端(AIO1)需要有一个BIAS参考电压 (EX:可用HY16F3981的REFO输出1.2V当做BIAS参考电压)，如果IA输入负端没有连接一个BIAS参考电压，在量测上会发生信号放大异常问题，使用者需特别注意到此IA应用限制。

HY16F3981 IA Input Range : BIAS=1.2V, input signal=+ /-1.08V



HY16F3981的IA虽然具有高输入阻抗特性，但是在连接具有高输入阻抗特性的Sensor时，此时如果程序又设置为IA Chopper On的模式(即0x41600[5:4]=11b)，当Sensor输入阻抗大于10k欧姆的时候，会有整体的ADC精度衰减的情况出现，随着输入阻抗越大，ADC精度衰减越多。建议使用者在连接高输入阻抗Sensor讯号的时候，程序设置为IA Chopper Off(即0x41600[5:4]=00b)，利用软件SW Chopper的方式来做到ADC offset噪声扣除，此方式不会受高输入阻抗而影响到ADC的精度。

2.5. Power 功耗

HY16F 在省电模式下 Idle Mode 可以达到 5uA ; Sleep Mode 可以达到 3.5uA , 当进入 Sleep Mode 之前把 VDD18 LDO 做 OFF 动作, 在 Sleep Mode 可以达到 2.5uA。使用者在进入 Idle 或 Sleep Mode 之前, 必须要确保相关模拟电源与内部高频晶振都是关闭情况下, 才可达到与规格书一样的省电功耗。此外, 当从省电模式到唤醒之后, 使用者也需要注意需要重新初始化与开启相关应用 IP , 才可避免发生进入省电模式之前 IP 工作正常, 但唤醒之后 IP 工作异常的问题发生 ; 此应用问题, 时常发生在程序开发阶段, 程序设计者应该特别注意。

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

在进入省电模式之前, 可按照顺序做以下设置 :

- 1). 把 CPU 的工作频率先切换到 LPO 低频之后, 再把 HAO 高频关掉。
- 2). 如有开启相关模拟电源输出, 则也需要做相对应的模拟电源关闭动作。
- 3). GPIO 的 Pull High 或 Pull Low 设计, 则是依照外部线路设计不同来做相对应的设置, 避免进入 Sleep/Idle 时候发生漏电或电压倒灌问题。
- 4). 同时确保在进入省电模式之前, 必须也要确保相对应的唤醒中断功能已经是已经开启, 这样才能达到与规格书一样的省电规格。

2.5.2. Active Mode --> Idle Mode/Sleep Mode-范例程序

以下范例, 程序设计有应用到 ADC 与 LCD IP, 当要进入 Sleep Mode 之前, 做相对应关闭动作 :

```
//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_CloseHOSC(); //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
```

说明：该范例程序，在进入 Sleep Mode 之前，先做关闭 VLCD 与 MCU Clock 动作，之后再 VDDA 与 ADC 关闭动作，确认相关 IP 都完全关闭之后，再进入 Sleep Mode 之前做最后的 VDD18 LDO 关闭动作，进入 Sleep Mode 之后可以量测到 2.5uA 的功耗。

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

要从 Idle 或 Sleep Mode 唤醒之后，可按照顺序做以下设置：

- 1). 首先可把 HAO 高频震荡打开，HAO 高频震荡打开之后，也需要先等待一段稳定时间。
- 2). HAO 稳定才做后续的其它模拟电源重新开启与 IP 设置。

举一个的应用实例说明: PWM 频率源选择为 HAO, 进入 Sleep Mode 之后会把 HAO 高频震荡关闭, 当芯片从 Sleep Mode 唤醒之后, 重新开启 PWM 做输出, 会发现 PWM 没有动作, 这时候是因为 HAO 没有先做重新开启动作, PWM 的频率源 HAO 没有震荡. 诸如此类的应用, 使用者在开发程序的时候, 都需要特别注意到。

2.5.4. Idle Mode/Sleep Mode --> Active mode-范例程序

以下范例程序为，当从 Sleep Mode 唤醒之后，做相对应 ADC 与 LCD IP 重新初始化动作

```
//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 CPUCLK=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();
```

说明：该范例程序，再从 Sleep Mode 唤醒之后，先做重新开启 MCU Clock 动作，之后才做 LCD 与 VDDA 与 ADC 等相关 IP 初始化动作。

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

HY16F 的 GPIO 时常被拿来设计做为 Sleep or Idle Mode 的唤醒按键设计，在程序的设置上需要特别注意增加各 GPIO 的 IDF 中断条件旗标判断，这可以避免发生 GPIO 唤醒 Sleep or Idle Mode 会死机的问题发生。

举例：如果设置使用 PT1.7 做为 Idle or Sleep Mode 的唤醒按键，则芯片在进入 Idle or Sleep Mode 之前，必须先判断 PT17IDF 中断条件旗标是否为 1，中断条件旗标为 1 才可进入 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. 内部高频 HAO & 内部低频 LPO

CPU 开启切换内部高频振荡做使用时，应该都需要注意等待稳定时间。不该在振荡器还没稳定的时候，程序就往下执行其他动作，进而可能产生当机问题。此外，使用者常常发生，批量生产时候，部分芯片会发生 UART 传输数据错误等问题，该问题也与 HAO 相关设置有密切关系。

2.6.1. 内部高频 DrvCLOCK_CalibrateHAO -使用注意事项

HY16F19x 系列，各频段的 HAO 有提供校正值可供使用者做使用。当把校正值写入 HATOR 寄存器之后，HAO 频率误差可以缩小在 2%以内。使用者在使用 UART 功能的时候，如果又使用 HAO 当做 UART 主频率，那一定要使用 DrvCLOCK_CalibrateHAO 此函数，可以把 HAO 频率误差落在 2%以内。可以解决批量生产时候，部分芯片发生 UART 传输数据错误等问题。

2.6.2. 内部高频 DrvCLOCK_CalibrateHAO -范例程序

范例程序如下：

```
DrvCLOCK_EnableHighOSC(E_INTERNAL,10); //Select HSRC
DrvCLOCK_SelectHOSC(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);
```

说明：该范例程序，预先设置 HAO=4MHz，这时候 HAO 的频率误差为 4MHz +/- 10%，当执行函数 DrvCLOCK_CalibrateHAO(1);之后，把 HAO=4MHz 的效正值填入在 HATOR 内，可以让 HAO 的频率为 4.147MHz +/- 2%误差。当 HAO 频率设定完成之后，UART 的频率源设置选项也需要输入正确的数值，这样可以有效解决批量生产时候，部分芯片发生 UART 传输数据错误等问题。并且需注意到，不同段的 HAO 频率有不同的 Trim 值，所以当设置 HAO=4MHz 的时候，只能选择 DrvCLOCK_CalibrateHAO(1)，同理，当设置 HAO=2MHz 的时候，只能选择 DrvCLOCK_CalibrateHAO(0)，依此类推。

2.6.3. 内部高频 HAO & 内部低频 LPO -使用注意事项

使用内部晶振需要注意事项：

- 使用 HAO 16MHz 晶振时，芯片工作电压须保持在高压供电动作。
- 设置寄存器 HAOTR 0x40304[7:0]的值，可调整 HAO 输出频率值。

举例:当设定 HAO 工作在 2MHz 时，如果实际输出只有 1.99MHz，那么即可藉由调整控制位 HAOTR[7:0]来调整频率输出，HAOTR 默认值为 0x80，往上调整即可增加 HAO 实际工作频率。

- 芯片在默认启动的震荡器为内部 HAO 2MHz，用户可进行设置寄存器 0x40300[4:3]来配置为其它的 HAO 输出频率值。
- HAO 配置为 4MHz 起震的稳定时间约为 0.5ms。
- 执行休眠(SLEEP)指令后，HAO 震荡器皆会停止震荡进入休眠状态。
- 从休眠模式(Sleep Mode)到唤醒的时间约 $1024 * \text{HAO} + 2048 * \text{LPO} \leq 64\text{ms}$ 。

注意:如果要满足上述描述唤醒时间，应该在进入 Sleep Mode 之前，CPU 频率源选择为 HAO。

- 执行待机(IDLE)指令，并不会停止 HAO 震荡器，但 CPU 会进入待机状态。
- 从 Idle Mode 到唤醒时间约 500 个指令周期，当 CPU Clock 使用预设 HAO=2MHz 时，唤醒时间约 250us。LPO 为芯片内部低速 RC 震荡器，输出频率 35KHz，具有低功耗电流特性，且在芯片上电或唤醒后即起振，同时也是无法关闭，在整个芯片工作过程 LPO 都是运行的。
- LPO 起振稳定时间约为 510us 且是看门狗(WDT)的唯一工作频率源。
- 执行休眠指令(SLEEP)后，LPO 震荡器皆会停止震荡。
- 执行待机指令(IDLE)后，LPO 不会停止震荡，但 CPU 会进入待机状态。
- 从 Idle Mode 到唤醒时间约 500 个指令周期，当 CPU Clock 使用预设 LPO=35kHz 时，唤醒时间约 14ms。

2.6.4. 内部高频 HAO =16MHz –使用注意事项

HY16F 芯片在 HAO=16MHz，并且芯片工作电压低于 2.2V~1.8V 的情况下，会有 PC counter 机率性发生跳飞的问题发生，导致程序误动作。因此在设置 HAO=16MHz 的情况下，建议设置 CPU 的主工作频率为 HAO=16MHz/2，而不要使用 HAO=16MHz/1，这可以降低 PC counter 跳飞的问题。如果应用上必须要工作在 HAO=16MHz/1，则可外部增加 Reset IC，当 HY16F 芯片工作电压 VDD3V 低于 2.2V 的时候，立刻把电压拉到 0V，避免发生芯片工作电压出现在 1.8V~2.2V 的区间。

2.7. Flash

HY16F 系列为 Flash 产品, 可以自行规划与指定程序区域以外 Flash 区块位置, 作为 Data Flash 做储存校正参数使用。在使用 Flash 做数据(Write/Read)存取之前, 也应该注意到一些使用限制。

2.7.1. Flash Read/Write –使用注意事项

- 1). 在执行 Flash 写入与读取程序指令之前, 必须先执行 SYS_DisableGIE(); 关闭全域中断使能, 这可以避免程序运行异常的行为发生。
- 2). Flash 有写入次数限制, Flash 有保证至少 2 万次的写入寿命。在进行 Flash Write 写入之前, 一定要有擦除 (Erase) 动作, 这样写入才会正常。Flash 寿命是依照 Erase 次数来计算, 非写入次数。Erase all+ N* Page Write, 仍然只算是损失一次使用寿命。同一个 Page 使用 N*page Erase, 就代表已使用了 N 次寿命。
- 3). Flash Write 与 Read 都有写入时间, 使用者可以直接使用 HY16F C 函式库的 Flash 相关控制函数做 Write 与 Read。使用 ROM_BurnWord 或 ROM_BurnPage 约花费 30ms 时间, 使用 PageErase 或 SectorErase 约花费 25ms 时间。以上函式已经包含 Erase 功能。
- 4). 执行 Flash 写入指令, 必须确保芯片工作电压 VDD3V 高于 2.7V, 如果芯片电压 VDD3V 低于 2.7V, 则可能会发生 Flash 写入错误行为。

以下说明 Flash Read/Write 函式库指令执行时间

- 1). DrvFlash_Burn_Word(unsigned int addr,unsigned int DelayTime,unsigned int data): 执行时间约 30ms, Delay time 填入时间为 0x2000 (当 HAO=2M 时候填入 0x2000, HAO=4M 时候填入 0x4000 依此类推)。
- 2). ROM_BurnPage(unsigned int addr,unsigned int DelayTime,int* data); 执行时间约 30ms, Delay time 填入时间为 0x2000 (当 HAO=2M 时候填入 0x2000, HAO=4M 时候填入 0x4000 依此类推)。
- 3). PageErase(unsigned int addr,unsigned int DelayTime); 执行时间约 25ms , Delay time 填入时间为 0x2000 (当 HAO=2M 时候填入 0x2000, HAO=4M 时候填入 0x4000 依此类推)。
- 4). SectorErase(unsigned int addr,unsigned int DelayTime); 执行时间约 25ms , Delay time 填入时间为 0x2000 (当 HAO=2M 时候填入 0x2000, HAO=4M 时候填入 0x4000 依此类推)。

2.8. LCD

HY16F19xB 系列产品有 LCD IP 功能, 使用 LCD 需要注意到 VLCD 工作电压, 也需要注意到 LCD 工作频率 LCK 设置。如果电压设置与 LCD 面板电压不符合则会产生, LCD 显示看起来亮度不足或着是 LCD 没有点亮的区块看起来像是点亮的显示问题(从可视斜角看会很明显)。如果 LCD 工作频率设置过慢, 则是会从 LCD 显示上看起来有闪频的显示异常问题。所以 LCD 工作电压 VLCD 与 LCD 工作频率 LCK 设置上都需要特别注意。

2.8.1. DrvLCD_VLCDTrim–使用注意事项

各 VLCD 电压有提供校正值可以使用, 当使用函式 DrvLCD_VLCDTrim 可以把 VLCD 校正值带入, 使得 VLCD 电压可以控制在 5%的误差范围内。如果没有使用函式 DrvLCD_VLCDTrim, 则 VLCD 电压误差则在 10% 范围内。使用者常常发生批量生产时, 发现部分芯片的 LCD 显示过亮或过暗, 这是因为 VLCD 电压批量生产 10%误差导致。如果在使用时, 有使用函式 DrvLCD_VLCDTrim 则可以把批量生产所产生的 VLCD 电压问题有效解决, 可以控制批量生产 VLCD 电压误差范围在 5%以内。

2.8.2. DrvLCD_VLCDTrim–范例程序

```
//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);
```

说明：该范例程序，说明如何设定使用效正值在 VLCD 电压，使用 DrvLCD_VLCDMode 与 DrvLCD_VLCDTrim 这两个函式搭配使用。并且需注意到，因为不同的 VLCD 电压段有各自的 VLCD Trim 值，所以正确设置使用方式应该是要按照范例程序中的 Case1~Case5 其中之一来使用即可。

2.8.3. LCD 频率 LCK –使用注意事项

LCD 频率(LCK)设置问题，当 LCD 设置 1/4duty，外部电压 R-TYPE，选择 LS_CK/8/32=136HZ=LCK 的扫描频率，LCD 显示会闪屏。以 4com 来说，如果要眼睛看 LCD 不闪烁，则 LCD frame Freq 要接近 60Hz，故 LCD 扫描频率至少要 $4com \times 60 = 240Hz$ ；而 LCK 的时间是以半周期(1/2T)来算的，所以 LCK 要设置成 $240Hz \times 2 = 480Hz$ ，所以应该要改成 LS_CK/8/8=544Hz，才可避免接上 4com 有显示闪烁的问题发生。LCD 规格上会都会写 LCD frame Freq (一般为 50~60Hz)。

2.9. 实时时钟 RTC

HY16F 系列产品, RTC 为常使用的 IP 功能, 在此特别针对 RTC 的 IP 功能做注意事项介绍。

2.9.1. RTC –使用注意事项

1). 程序设计者, 如打算进入省电模式(Sleep/Idle)的时候还是持续让 RTC 做计数工作, 需要注意, RTC 可以在 Idle Mode 工作, 但是 RTC 在 Sleep Mode 是无法正常工作。以下为 RTC+Idle Mode 的参考工作电流:

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

2). RTC 的频率源选择可以是内部低频 LPO 或着是外部低频 LSXT(32768Hz), 如果选择的是外部低频 LSXT, 则需要考虑到 32768 起振稳定问题, 起振稳定问题牵涉到 PCB Layout 走线和晶振品牌规格都会有关系。HY16F 系列在连接 32768 晶振的外部引脚, 均可选择调整电容值大小(0~20pF)与示情况并连电阻(10M ohm), 来达到 32768 稳定起振。

3). RTC IP 的寄存器 0x41A04[22:16]可以提供 RTC 补偿使用, 当外部振荡器 LSXT 的震荡频率偏离 32768Hz, 可以透过补偿的方式, 让 RTC 计数频率趋近于 32768Hz, 可补偿的频率范围为: 最大的输入频率为 32772Hz, 最小的输入频率为 32763Hz。例如: 外部振荡器 LSXT 的震荡频率为 32764Hz, 这时候如果在 0x41A04[22:16] 寄存器填入 0111111b, 则补偿值为+126PPM, 可让 RTC 计数频率达到 32768Hz。

4). 在实际电路中, 可以通过示波器观察振荡波形来判断振荡器是否工作在最佳状态, 示波器在观察振荡波形时, 观察 OSCO 管脚(Oscillator output), 应选择 100MHz 频宽以上的示波器探头, 这种探头的输入阻抗高, 容抗小, 对振荡波形相对影响小。由于探头上一般存在 10~20pF 的电容, 所以观测时, 适当减小在 OSCO 管脚的电容可以获得更接近实际的振荡波形。工作良好的振荡波形应该是一个漂亮的正弦波, 峰峰值应该大于电源电压的 70%。若正弦波峰峰值小于 70%, 可适当减小 OSCI 及 OSCO 管脚上的外接负载电容。反之, 若正弦波峰峰值接近电源电压且振荡波形发生畸变, 则可适当增加负载电容。

5). 在振荡线路中, 很重要的一个特性是整个线路上的负载电容 (Load Capacitance;CL), 它是由 gate 端(OSCI) 的频率调整电容 CG, Drain 端 OSCO 的频率调整电容 CD 及杂散电容 CS 等三个参数共同组成。其中负载电容及两个频率调整电容是已知的, 可以透过下列公式求得杂散电容值:

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

通常厂家建议的 32768Hz 外接负载电容(CL)为 7pF、9pF、及 12.5pF (通常以 12.5pF 的规格居多),

而一般杂散电容 (CS) = PCB 杂散电容 + 芯片电路接面电容, 一般约在 3~5pF, 如果暂不考虑 CS, 则 $CL = (CG // CD, 12.5pF = (CG // CD, 得 CG, CD 各用 25pF。$

2.10. 定时计数器 Timer

HY16F 系列产品, Timer 为常使用的 IP 功能, 在此特别针对 Timer 的 IP 功能做注意事项介绍。

2.10.1. TimerA –使用注意事项

在使用 TimerA 时候, 程序设定 `DrvTMA_Open(0,0)` 与 `DrvTMA_Open(1,0)` 做计数器功能使用, 会发现到有计数不准与不稳定的现象。因为 HY16F 的 CPU 进出中断矢量消耗约 200 个机械指令周期, 如果 CPU=8MHz, 则总共消耗时间为 $(1/8000000) * 200 = 25\mu s$ 。当设定程序执行 `DrvTMA_Open(0,0)`, 则进入一次的中断时间为 $8000000/32(ENTAD=1b)/2(TAS=0000b) = 125kHz = 8\mu s$, $8\mu s$ 的时间远远小于 $25\mu s$, 真实情况会看到程序是一直卡在 TimerA 中断而进入不到 main 执行。

同理

`DrvTMA_Open(1,0)`, 则一次进入中断时间为 $16\mu s$

`DrvTMA_Open(2,0)`, 则一次进入中断时间为 $32\mu s$

$32\mu s > 25\mu s$, 可以看到稳定输出, 因此, 使用 TimerA 做计数时候, 最快建议使用为 `DrvTMA_Open(2,0)`。

2.11. DAC

HY16F 的 DAC 模块本身没什么驱动负载能力, 大约 15uA。如果是 DAC+OP, 那最大负载能力有 1mA(VDDA=3V 情况下), 或 0.5mA(VDDA=2.4V 情况下)。使用者需注意 DAC 推动负载的大小, 避免 DAC 输出在大负载的时候, 电压被异常拉低的情况发生。

2.12. R2ROP

HY16F 系列产品, R2ROP 为常使用的 IP 功能, 在此特别针对 R2ROP 的 IP 功能做注意事项介绍。

2.12.1. R2ROP 当 unit gain buffer –使用注意事项

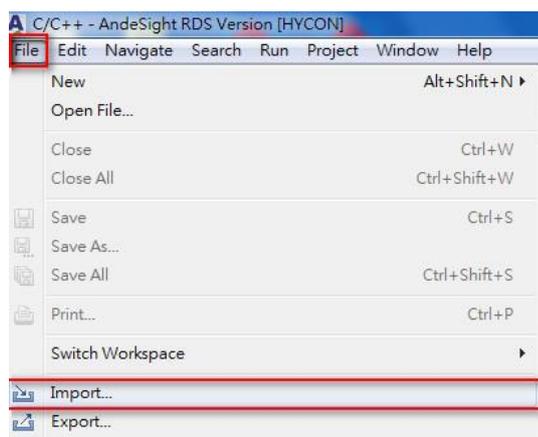
R2ROP 当 unit gain buffer 应用的时候, input 从 0~5mV 的时候会存在一个 offset 电压存在, 并且在此范围内电压变化从输出端是观察不到, 直到约 6mV 电压的时候, 电压输出才会开始有反应。如果有比较好的线性度, 则建议输入电压是在 $VSSA+0.1V \sim VDDA-0.1V$ 范围内。因为此限制, 在此提供一个参考的解决方案, 把整个量测系统输入电压垫高, 可以利用 HY16F 的 REFO 这只接脚输出 1.2V, 来当作 Bias 电压。使得参考电压点从 0V 变成 1.2V。

3. HY16F 系列开发工具应用说明

3.1.1. AndeSightRDS V2.0.1 转换到 AndeSightRDS V2.1.1

在 2017 年以前使用的标准开发工具为 IDE:AndeSight RDS V2.0.1 搭配 HY-Protocol, 在 2017 年之后, 更新升级开发工具为 IDE:AndeSight RDS V2.1.1 搭配 HY16F Mini Link. 使用者应该要使用正确的硬件开发工具与 IDE 软件版本, 才可正常做 HY16F 程序开发与 Debug 等动作。此外, 也需注意, 旧有的 AndeSight RDS V2.0.1 开发项目包, 如果要移植到 AndeSight RDS V2.1.1 环境做沿续设计使用, 需要先做以下 Step1~Step4 等设定动作, 才可以确保旧有的 RDS V2.0.1 项目包在 RDS V2.1.1 环境转换是没问题的, 可以正常做 Debug 等单步执行除错动作。

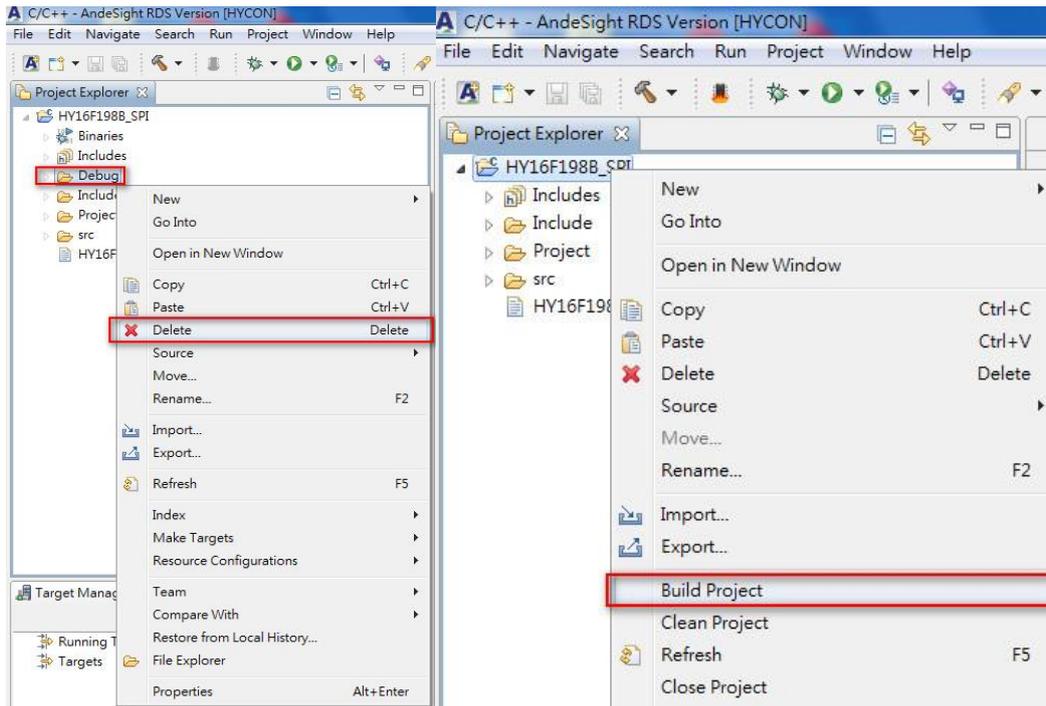
Step1 : 在 RDSV2.1.1 环境下 Import RDS V2.0.1 项目包。



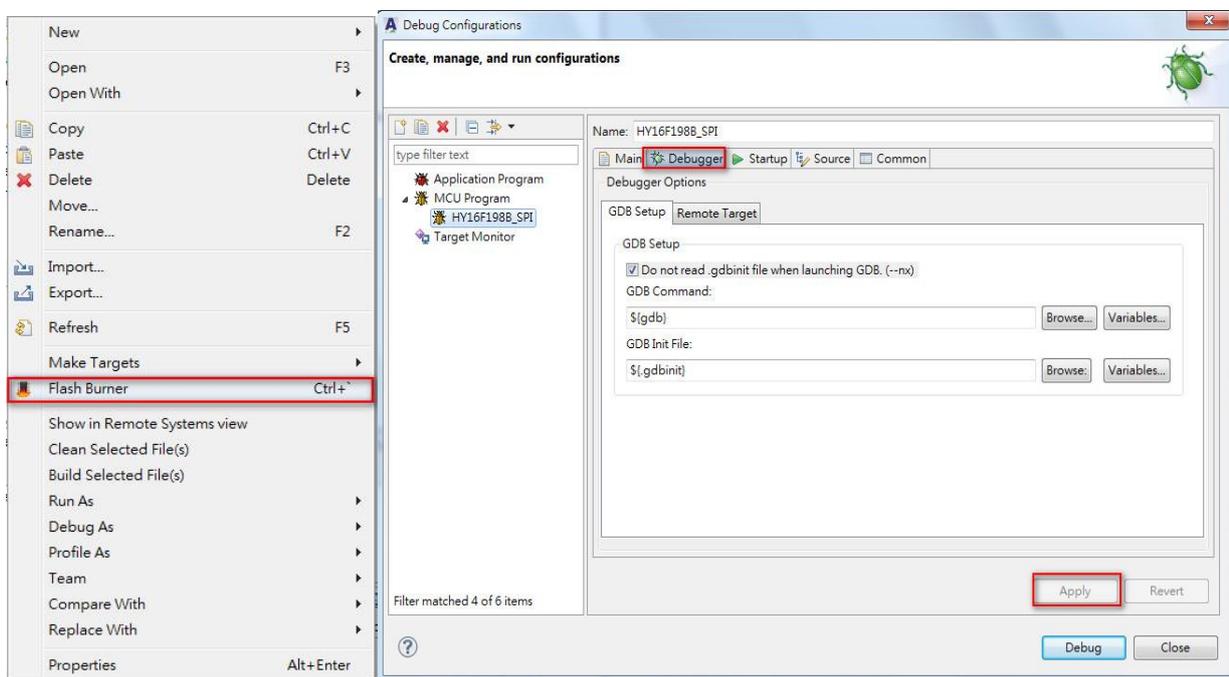
Step2 : 把 Debug 数据夹 Delete 删除之后, 重新 Build Project, 建立与产生一个新的 Debug 数据夹。

HY16F 系列

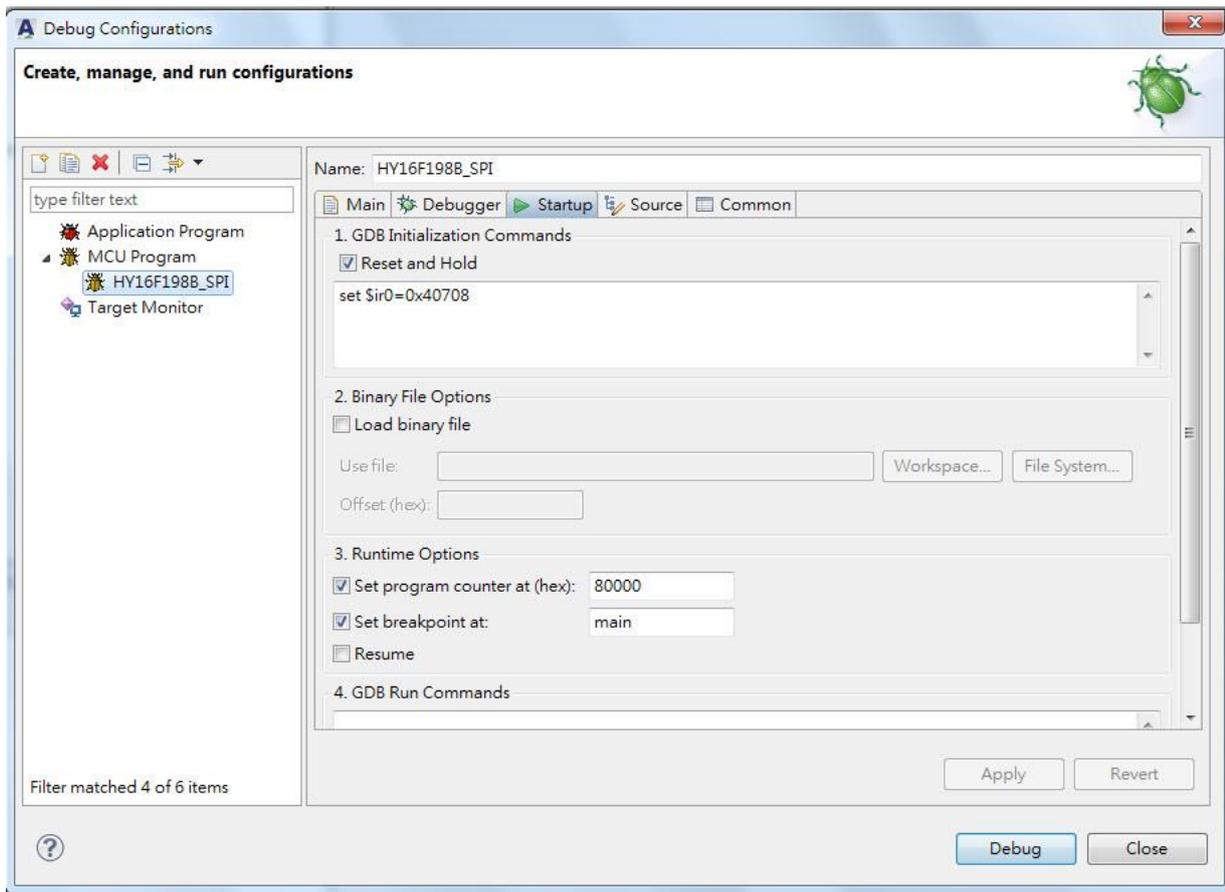
产品应用注意说明



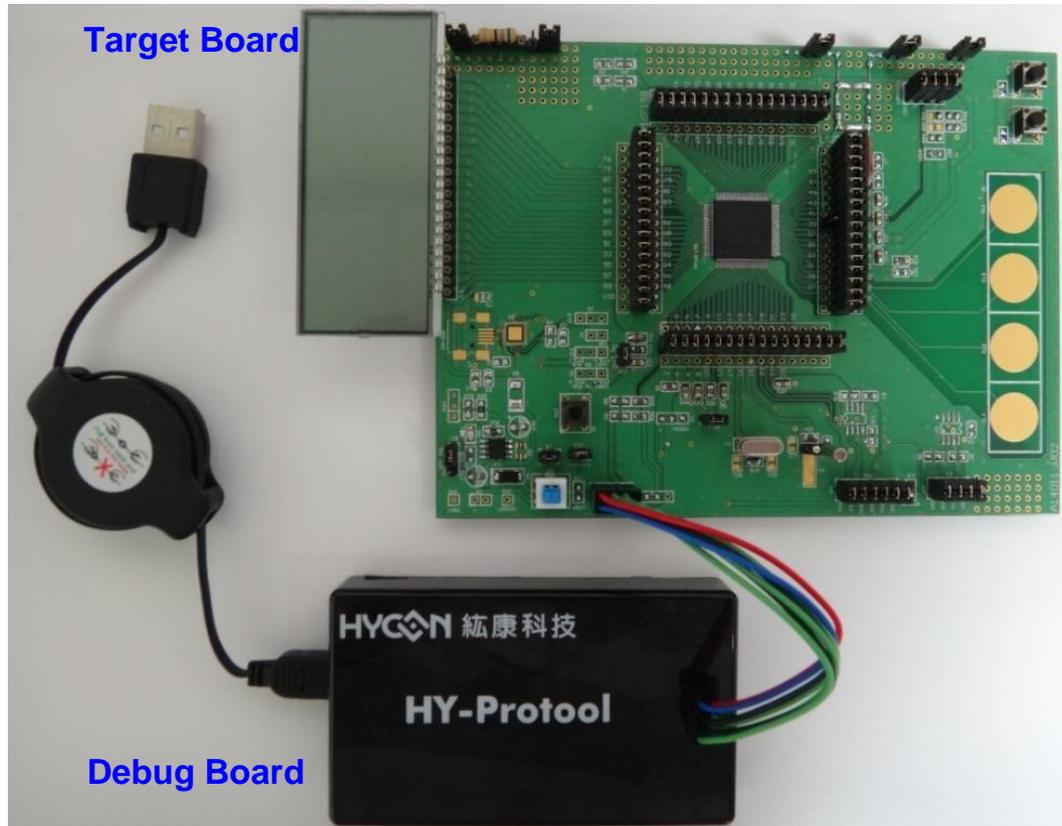
Step3 : Build Project 之后, 执行 Flash Burner 进行烧 Code 动作, 点选 Debug Configurations 确保 Debugger 环境设定与下图一模一样。



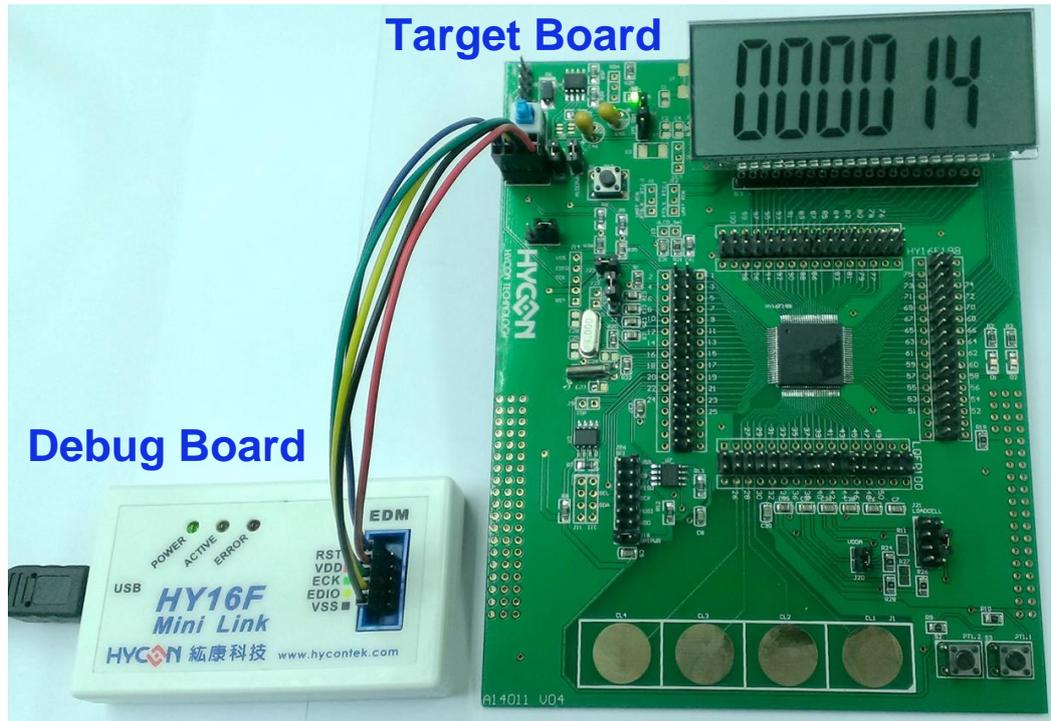
Step4 : 确认 Debug Configurations 环境下的 Startup 设定也与下图一样。



下示意图为 HY16F19xB 搭配 HY-Protool 硬件开发套件(HY16F19x-DK02), 实际使用应搭配 IDE:AndeSight RDS V2.0.1 :



下示意图为 HY16F19xB 搭配 HY16F Mini Link 硬件开发套件(HY16F19x-DK04), 实际使用应搭配 IDE:AndeSight RDS V2.1.1 :



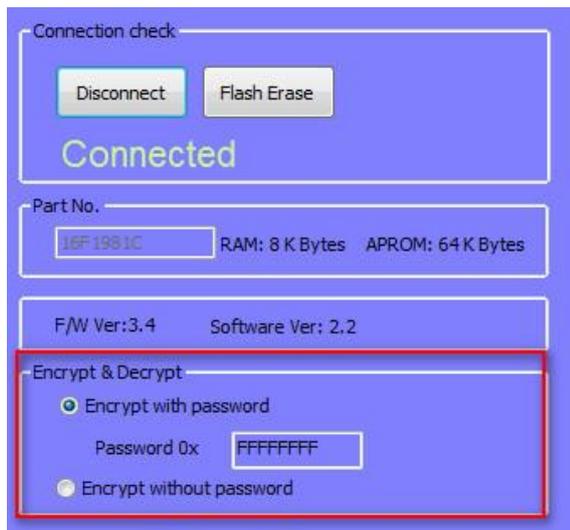
3.1.2. Stack 占多少 SRAM 空间问题

SRAM 的组成包含四个元素 data+bss+heap+stack。其中, stack 的起始位置是 0x1fc0, 它是动态配置的, 完全是看程序怎么写来动态配置长大的, 如果程序写越多层 call function, 那 stack 就会越长越大, Stack 占用地址是从 SRAM 的地址位置 0x1fc0 --> 0x1fbf-->0x1fbe 往下减小。而 bss 与 data 地址是从 SRAM 的 0x40 起始地址位置逐步往上增加上去; 直到 bss 和 data 区块变量与 stack 重叠到, 那这时候可能就会发生声明变量乱掉的问题, 导致程序错乱。

举例 : SRAM 使用 7.5kB 空间(data=522, bss=7239), 代表 stack 只剩下 8128-7239-522=367 空间可以使用, 在此情况下继续开发程序, 将会很容易发生 stack 与 bss/data 区块数据重叠, 导致程序发生不可预期的错误发生。因为目前使用的 Andes CPU N801 还没有提供 stack protection 这个功能, 所以程序在 compile 也无法文件掉此问题, 使用者在开发程序时, SRAM 的内存规划需特别注意。

4. HY16F 系列加密&解密功能说明

HY16F 系列产品，可透过 HY16F Writer 烧录器，做芯片加密解密动作，可防止芯片内的程序代码被读取出来。而 HY16F18x/HY16F19x/HY16F19xB/HY16F3981 系列产品的加密与解密行为是有些微的功能差异，以下将加强说明功能，下图为 HY16F Writer 烧录器加密操作画面。



1). Encrypt with password 功能：(预设为 0xFFFFFFFF，可自由设定密码做保护，即使输入密码正确 HY16F19xB/HY16F3981 会强迫清除部分的 Flash，但 HY16F19x/HY16F18x 不会)。

当设定 Encrypt with password 时候，如果密码输入正确，那 Code 会被清除，Code 的清除范围依照当时设定密码保护烧录 APP 的范围为基准。EX：假设烧录 APP 大小为 6KB 的 Flash，选择 Encrypt with password 输入密码 0x12345678；当再一次要联机的时候，会要求输入密码，当输入密码正确，此时 0x9000~0x91800 的范围会被 Erase 掉变成 0xFF，这时候密码状态会被解除为 0xFFFFFFFF。输入密码正确清除 Flash，这是 HY16F19xB/HY16F3981 的 IC 设计行为，但是该行为在 HY16F19x/HY16F18x 不会发生。

对 HY16F19xB/HY16F3981 来说，当选择 Encrypt with password 有设定密码的时候，可以选择输入正确密码当方式做联机与清除 Flash APP size 和密码，或着是暴力破解法(不需输入密码)选择 FlashErase 来对全部的 64KB Flash 做清除动作。

对 HY16F18x/HY16F19x 来说，当选择 Encrypt with password 有设定密码的时候，输入密码正确，可以正常联机，但是 Flash 不会被清除掉，并且密码状态还是存在，除非按下 Flash Erase 或着重新烧录新的一组密码，才能解除密码。但是如果密码忘记，按下 Flash Erase，还是无法清除 Flash，只有知道密码的情况下才能做解除动作。

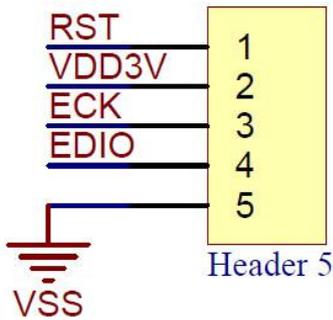
2). Encrypt without password 功能：设定之后，Flash 内容就无法被读取出来，只能使用 Flash Erase，清除 64KB 的 Flash 都为 0xFF，才能让芯片回复原始状态，这是唯一解除密码方法。此功能，HY16F18x/HY16F19x/HY16F19xB/HY16F3981 都是相同的，并没有任何差异。

5. HY16F 系列硬件设计注意事项

纭康 HY16F 系列芯片是属于 $\Sigma\Delta$ ADC+MCU 产品, 具有高分辨率与高精度能力, 因具备最小 65nVrms 讯号的解析能力, 故对微小的漏电流与信号干扰特别敏感。所以在此介绍一些基本的芯片相关电路设计与 PCB 布线观念, 让使用者更能得心应手的开发产品。

5.1.1. 电路设计注意事项

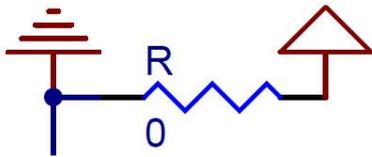
1). 电路设计规划时, 最好预留烧录与 Debug 专用的 5pin Connector(2.54mm pitch), 脚位排列顺序也要一样, 在产品开发与量产的时候, 可以直接与 HY16F 烧录器或者 HY16F Mini Link 直接对接, 方便开发与烧录。



2). 当 ADC Output rate 小于 1k 时候, ADC 量测通道 AIO0-AIO1 与 AIO2-AIO3 等 ADC 输入引脚外挂 100nF 电容, 当 ADC Output rate 大于 1k 时候则外挂 10nF 电容, 在两个 ADC 量测通道输入引脚之间外挂电容, 可增强抗噪声干扰。

3). VDDA 连接 10uF 对地电容, VDD18 连接 1uF 对地电容, VPP 保持 NC。

4). 线路规划时, 模拟地与数字地应该用不同符号标示, 在模拟地与数字地之间, 设计一个 0 ohm 短路电阻。



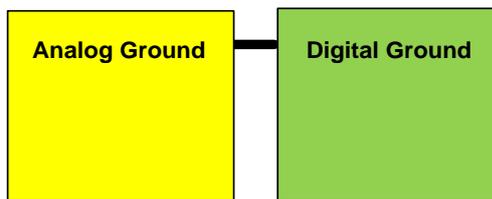
5.1.2. PCB Layout 注意事项

请务必遵守一般 PCB 线路布局规则及知识。例如，直角走线会对 EMI 造成麻烦。使用 45 度转角走线较佳。

5.1.2.1. 电源(Power)布线

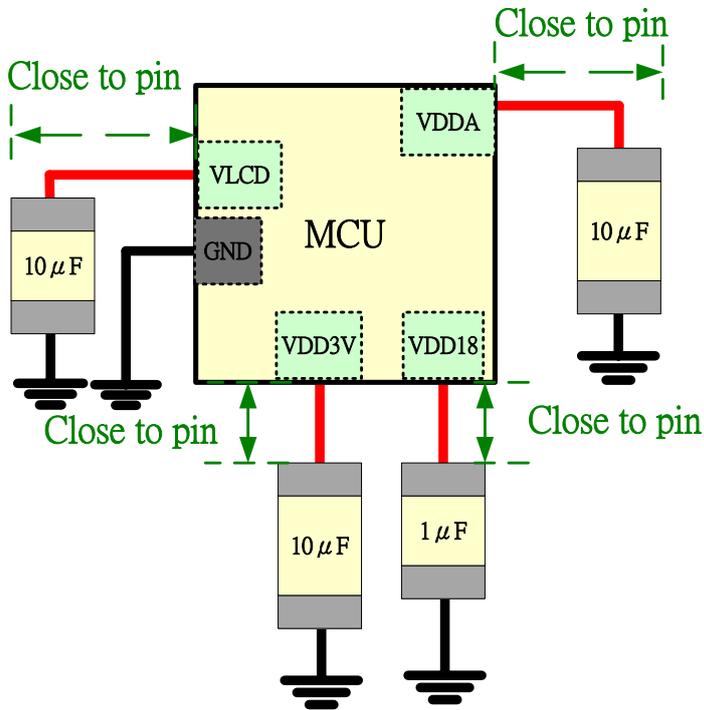
1). 模拟讯号与数字讯号必须利用 VSS 铺地隔开且独立布线

此部分布线不佳，最直接影响到的是当按钮开关按下时，会有电流回路流进 VSS 而产生电位差(PCB 布线电阻)。此电位差在高 ADC 分辨率产品应用上容易造成测量误差由几个 ADC Count 至数十个 ADC Count，进而导致测量结果不正确。故建议布线时模拟与数字采独立布线，以避免测量误差发生，同时为了增强抗干扰能力，两个地线间用短路线连接，其短路位置最好靠近电池 VSS 位置做短路。如果 PCB 是双面版，VSS 铺地尽量多做过孔，以减少铺地的内阻。



2). 电源引线即布线越短越好，越短即是阻抗越低越好

引脚端的稳压电容请 Layout 靠近芯片。如引脚：VDD, VDDA, VLCD, VDD18, REFO。

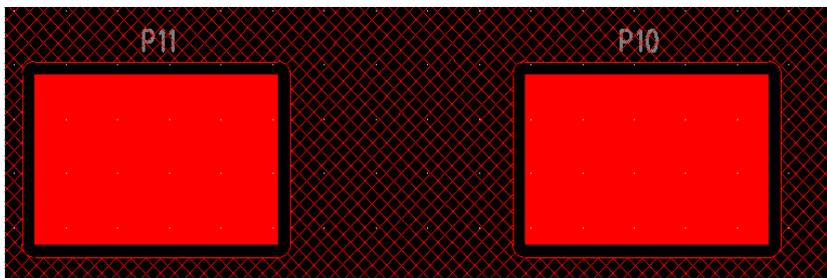


3). 铺地铜

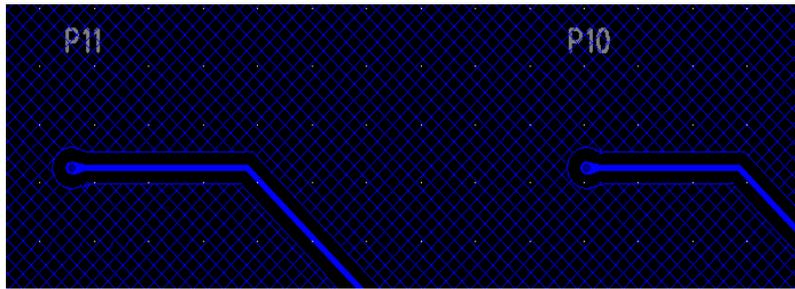
PCB 的顶层及底层都需要铺地铜，将铺地铜应用到线路将会增加抗扰度。我们定义铺地铜到走线宽度为 5mil，线距为 20mil，格点距为 20mil，顶层及底层都有。由于 PCB 厂的策略，线宽必须设定为 10 mil 或更大。铺地铜的线宽设定为 10 mil，线距为 20 mil，格点距为 40 以满足限制。

请参考以下顶层及底层的图示。

顶层的铺地铜:



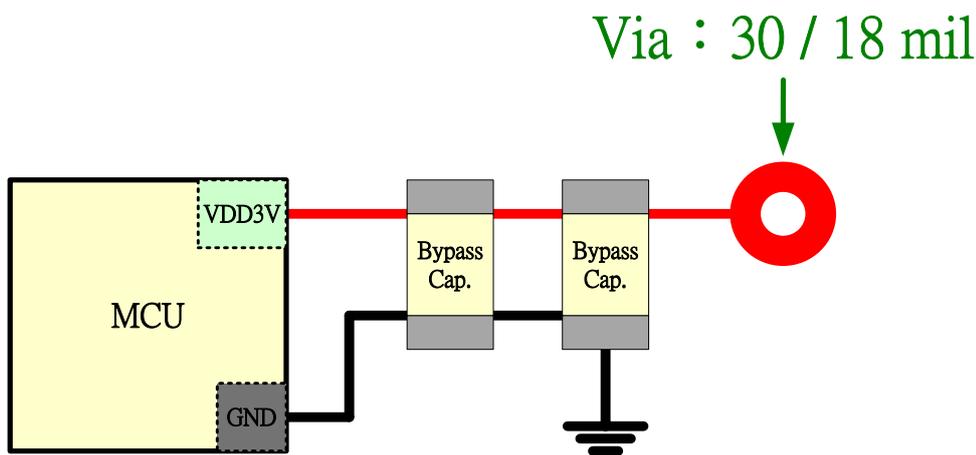
底层的铺地铜



4). VDD3V/GND 线路及电源供应贯穿孔

VDD3V/GND 建议使用 25~30mil。电源供应的贯穿孔应该加上去偶电容，建议如下图示摆放以提供适当的去偶能力。电源来源的噪声会被去偶经由电容器到接地及流回电源来源。VDD3V/GND 使用的贯穿孔尺寸建议为 30/18mil 以减少贯穿孔结构的等效电感。内部 LDO 旁路电容器线路(例如：VDDA 与 REFO 输出引脚)，内部 LDO 旁路电容器建议使用 10~20mil 宽的 PCB 线路以提供稳定的电源。

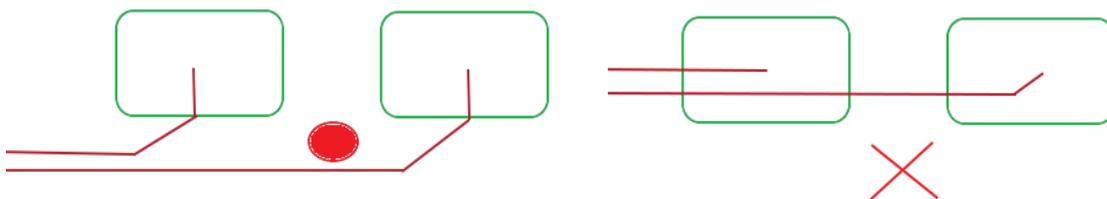
VDD3V 贯穿孔洞及线路:



5). 双层版 PCB 设计

在 2 层的 PCB 设计中，建议使用 5~10mil 宽的线路来设计。线路彼此间的间距建议为两倍的线宽 (建议为 10~20mil) 以避免干扰。

良好的与不良的线路走线图如下范例：此范例中的 pad 代表 ADC sensor pin，不能有走线跨到 Sensor pin 的情形发生。



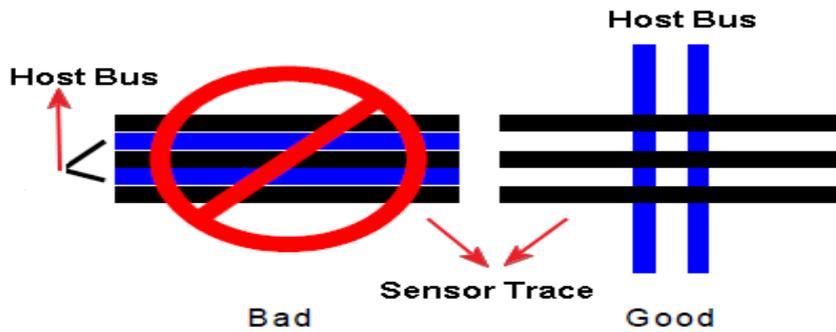
6). 介于主机及传感控制器间的数字讯号线路

数字讯号是与主机间的通信线路，与其它传感线路相比，切换的频率比较高。时钟及数据接脚建议以

5~10mil 宽的线路来设计数字讯号线路。时钟及其它信号 (电源, 数据, 控制, 传感器线路)间的间距建议至少为两倍线宽 (应该为 10~20mil)。

禁止将这些数字讯号线路与模拟线路平行走线, 平行的结构会造成很大的串扰。如果不能避免数字讯号线路靠近传感器线路走线, 建议交错部分采 90 度直角走线, 以减少串扰作用。

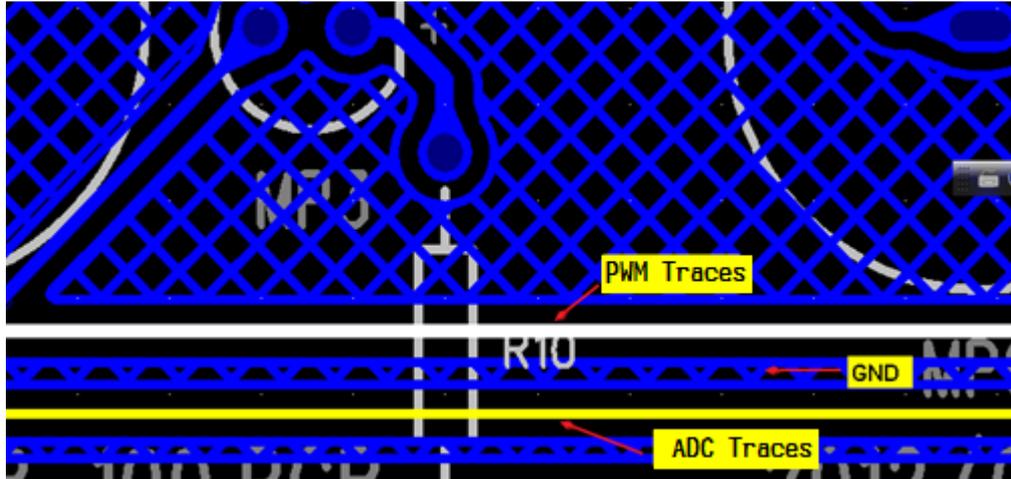
不良的及良好的主机总线及模拟讯号线路走线:



7). 数字讯号(PWM)线路及模拟讯号(ADC)线路

请注意务必禁止将 PWM 的控制线路与 ADC 线路平行走线。请在数字控制线路及模拟线路间插入铺地铜, 数字控制线路和模拟线路之间若不插入铺地铜, 距离建议至少 50mil。

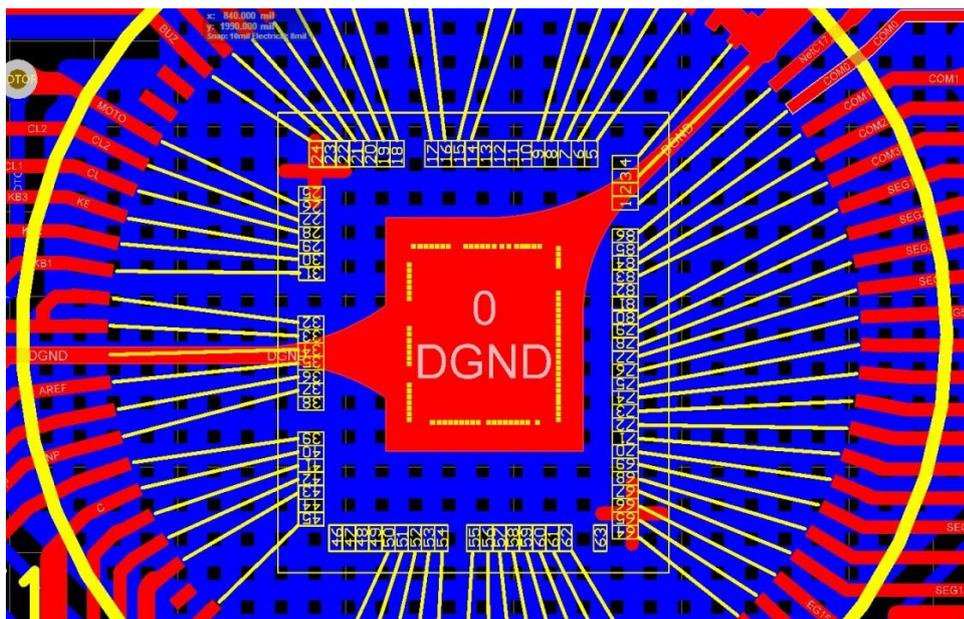
请注意: 插入的地线不可只有一边接地, 否则会形成单极天线(Uni Polar Antenna), 在 EFT 干扰时会影响整体效能。PWM 线路及模拟讯号线路之间的接地可参考下图:



请注意: PWM 的控制线路如果与模拟线路垂直交越, 不得使用 SMT (表面贴片)组件, 以避免垂直交越的距离太近, 造成模拟讯号受到干扰。需使用另面跳线方式, 以保证 PWM 控制线路和模拟线路保持在板厚(1.6mm)的距离。

8). Dice 或着 COB 布线

置放芯片的金属铺面请连接至 DGND(Digital GND), 详细连接方式, 可参考下图。



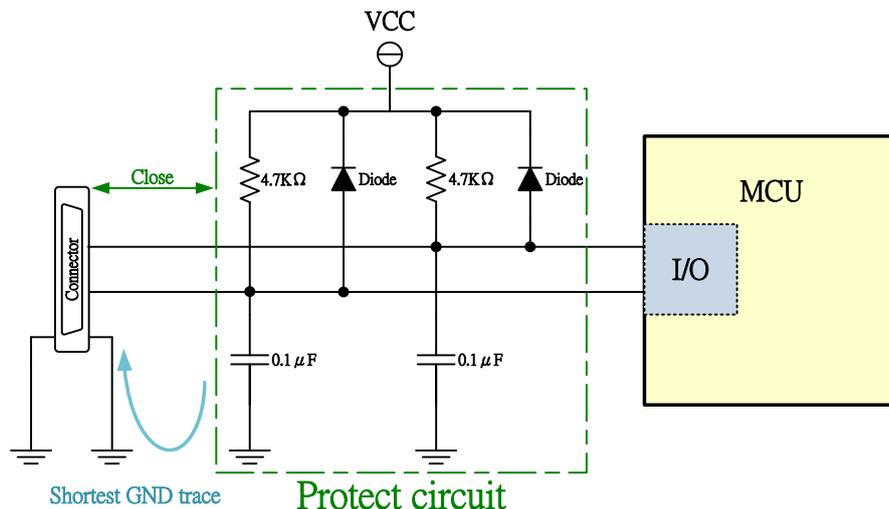
5.1.2.2. 模拟(Analog)布线

- 1). 模拟输入 AIO0 与 AIO1, AIO2 与 AIO3 等等的输入电容(0.1uF)越靠近引脚越好, 且外部信号输入时, 走线请先经过滤波电容后再引进到芯片模拟输入引脚。
- 2). AIOx 输入引脚走线, 请平行走线且越短越好, AIOx 引脚走线的上下板层避免有其它水平或垂直交叉走线。
- 3). AIOx 输入引脚走线, 允许的情况下, 建议 AIOx 彼此脚间可铺地线。

5.1.2.3. 数位(Digital)布线

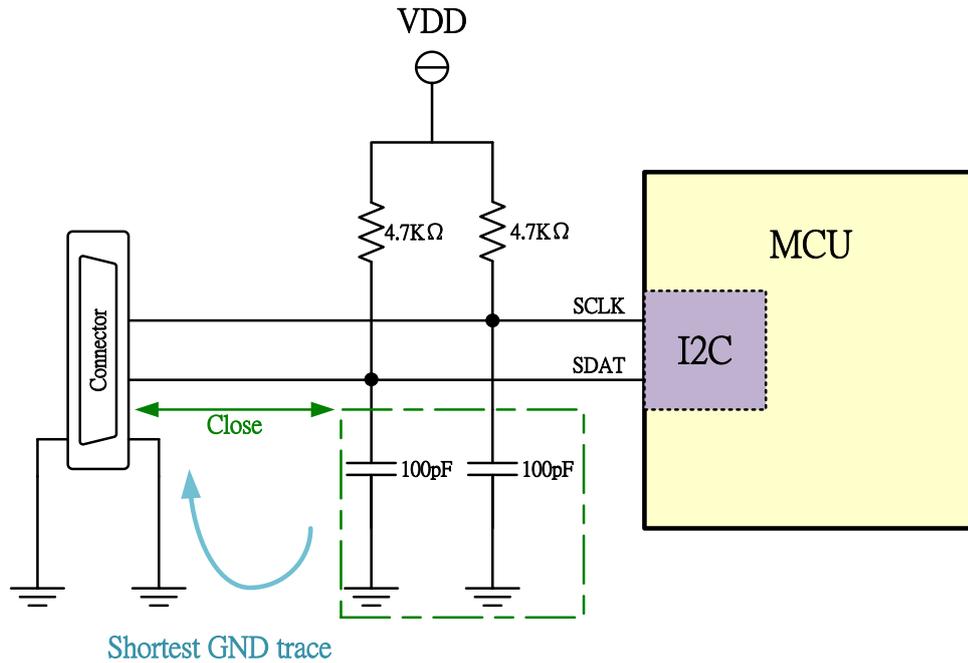
- 1). 外接 32768Hz 振荡器引脚(RTC), 走线越短越好且必须利用铺地隔离其它 I/O 产生的 AC 讯号, 以避免因信号干扰而导致 RTC 产生的频率受到影响。
- 2). 外接 32768Hz 振荡器引脚(RTC), 避免(AIOx 引脚)距离太近。
- 3). 外接 32768Hz 振荡器引脚(RTC), 避免电源引脚 Layout 经过。
- 4). I/O 端口:

使用任一 I/O 端口和主机或电源板沟通时, 需要加上保护电路。保护电路需放在靠近 Connector 埠。若是电容距离 connector 太远则会失去保护效果。请注意: 下地电容的地回流到电源板路径必须愈短愈好。



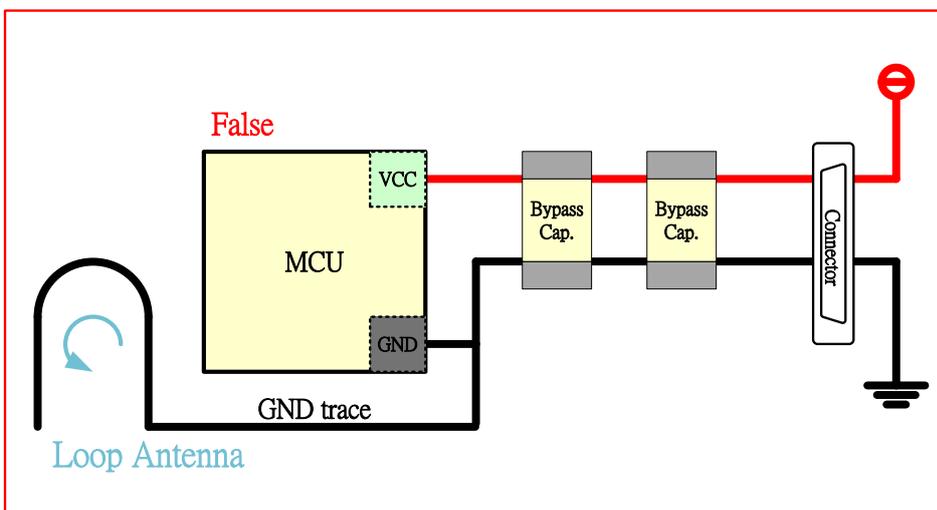
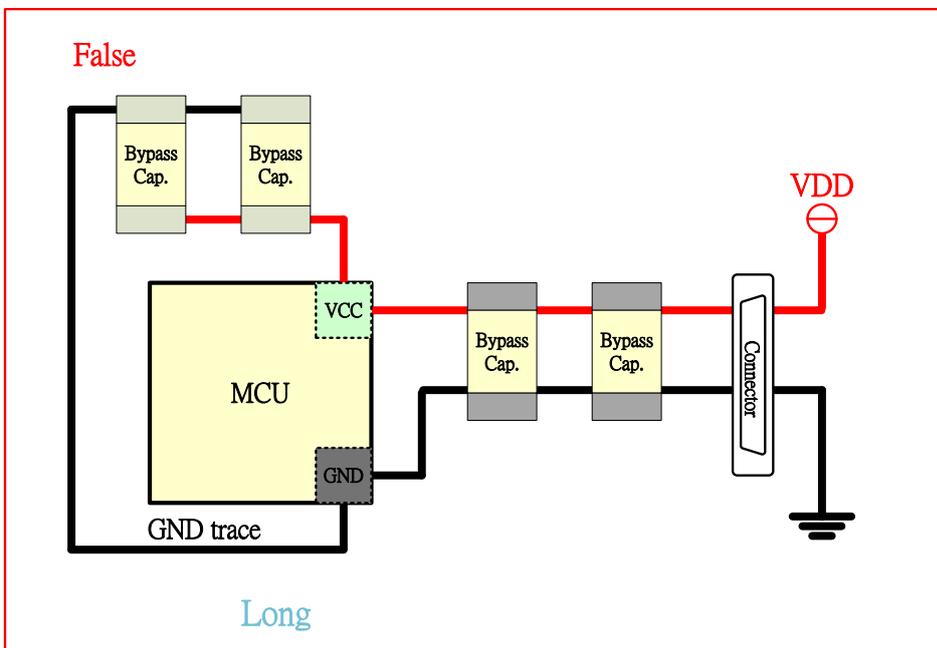
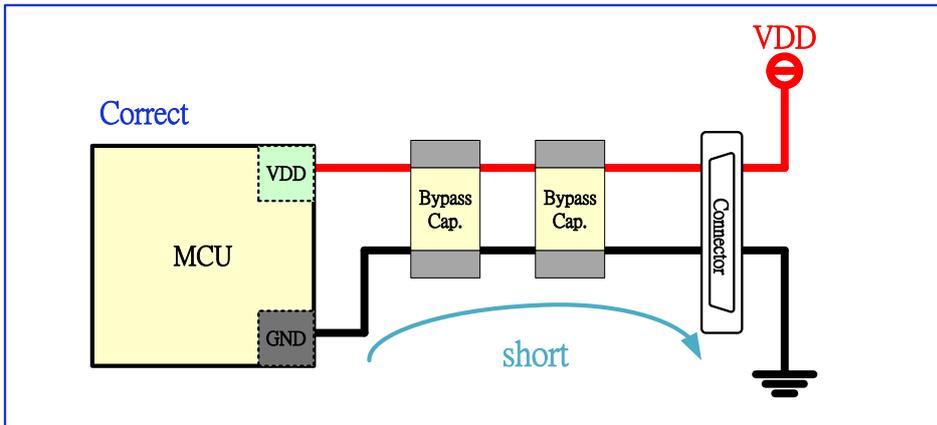
5). I2C 界面:

当使用 I2C 作为主机接口时, SCLK/SDAT 需要 4.7KΩ 上拉电阻以适应适当的电子特性。在不同的应用和信号负荷上可能会有微小的差异, 使用者可以使用在 PCB 上的外部上拉电阻。外部使用小于 100 pF 的去偶电容, 可用来减少电源噪声和数据处理噪声(Noise)。去偶电容放置应靠近与主机端连接的接头(Connector), 接地回路应尽量短, 直接回流到 Connector 端的地(GND)。



6). IC 的地回流路径：

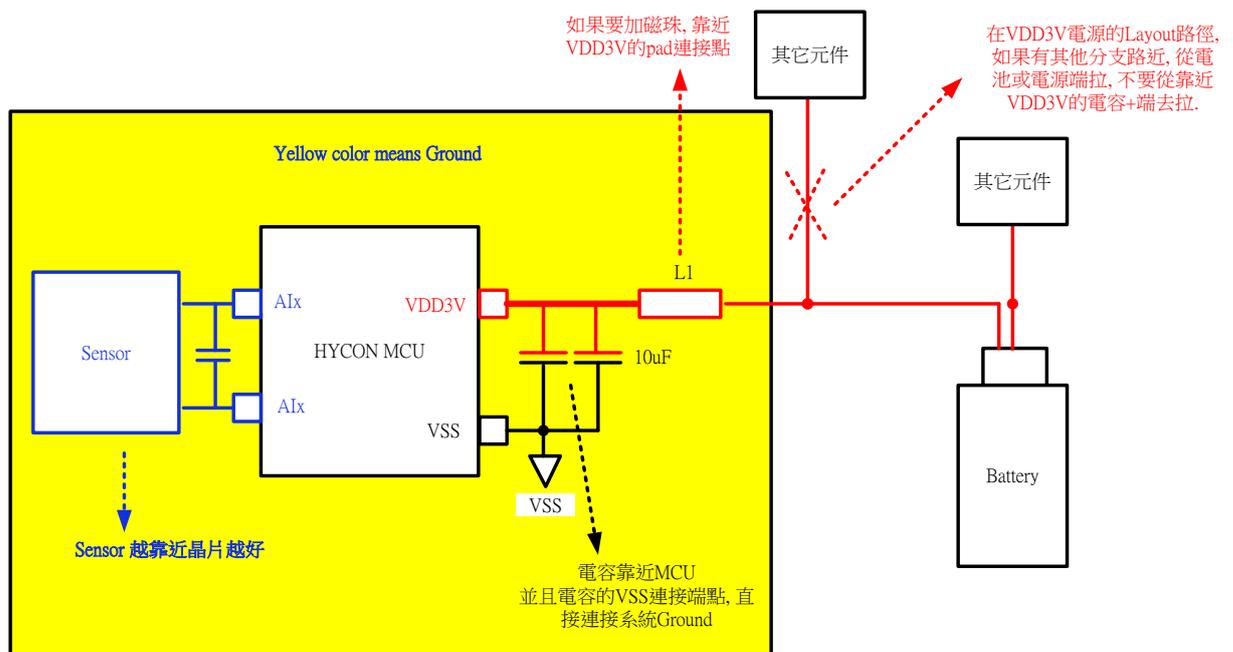
MCU 的地回流路径，需经过 VDD/VDD18 滤波电容的接地点。在流回电源 connector 埠时的走线需尽量短，避免过多的绕线，也要避免成为“Loop Antenna”。



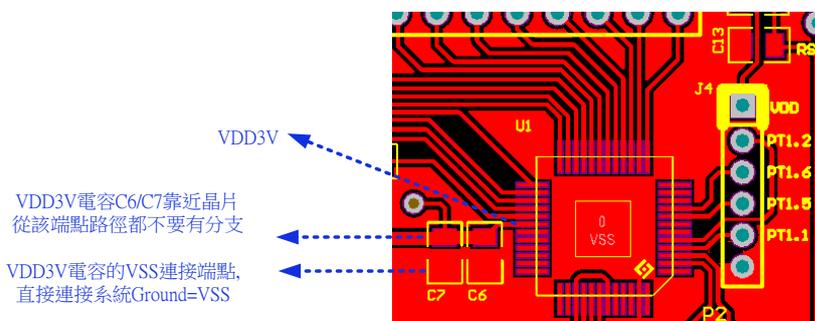
5.1.3. RS 防护设计注意事项

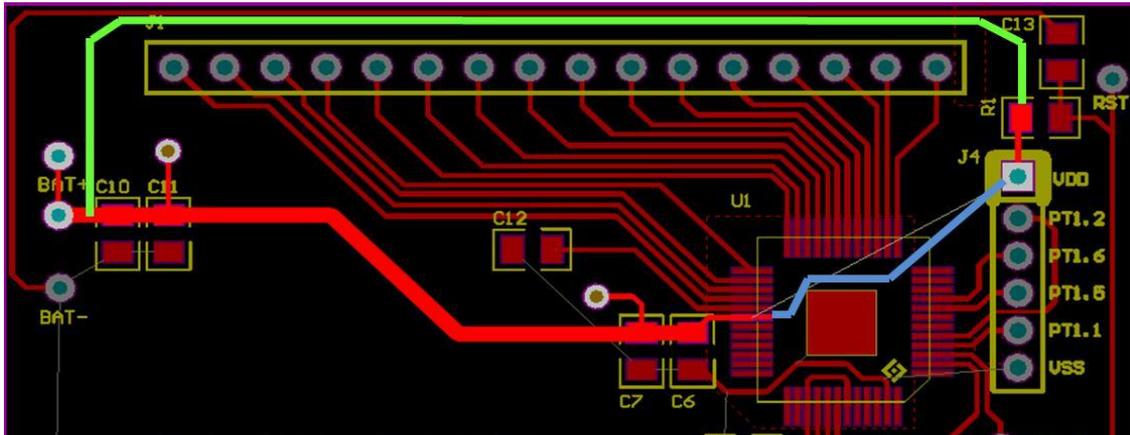
RS抗干扰, PCB Layout 设置建议 :

- Sensor 与 Aix 间的走线别绕太远或经过通信线。
- Ground 铺的面积要大。
- VDD power, 如果要加保护组件磁珠, 要先串磁珠后再接VDD3V, 此VDD3V 要先接10uF 再接104pF(0.1uF) 后才能接至 MCU 的 VDD3V pin。
- VDD3V 对地的电容要最短路径, 并且VDD3V 对地电容的路径, 电容两端接点尽量不要再有分支路径。如果有其它分支路径需求, 可参考以下示意图。



VDD3V的對地電容連接示意圖, 如下



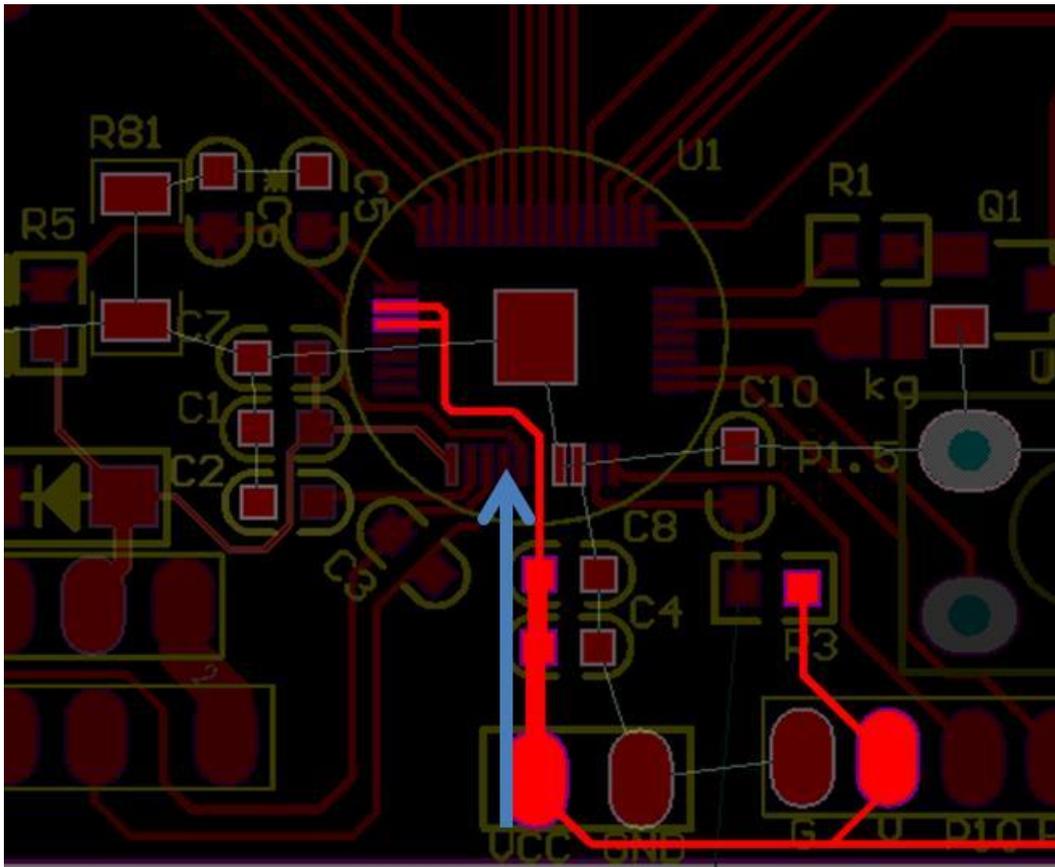


上图范例，可注意两个地方：

注意 1：VDD 电容选用 C7、C6 是正确的，C10、C11 为错误示范。

注意 2：芯片联机到 VDD 有蓝线与绿线两条路径，绿线是正确，蓝线为不良示范。

当外部电源直接从 VDD pad 灌入，蓝线路径等同芯片的 VDD3V pin 脚是先直接连接到 VDD pad，绕很远才经过 C10、C11 电容，最后才到 C6、C7 电容。



上图范例，可注意 1 个地方

注意 1：Power 进到 PCB 后，经 C4、C8 后就到 IC 的 VDD3V pin，此为正确示范。

- Layout 避免环形天线效应产生

图 1

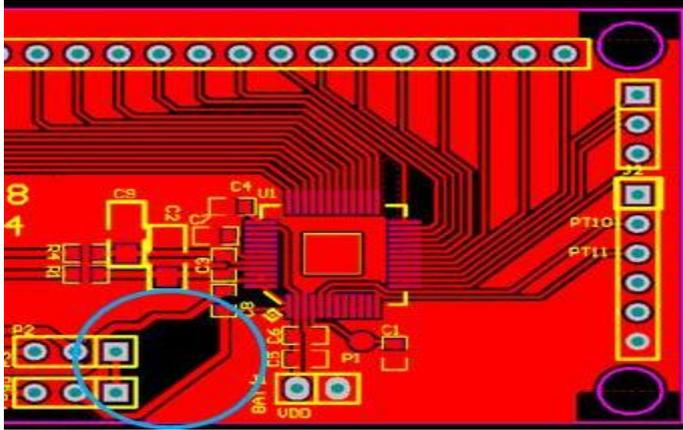
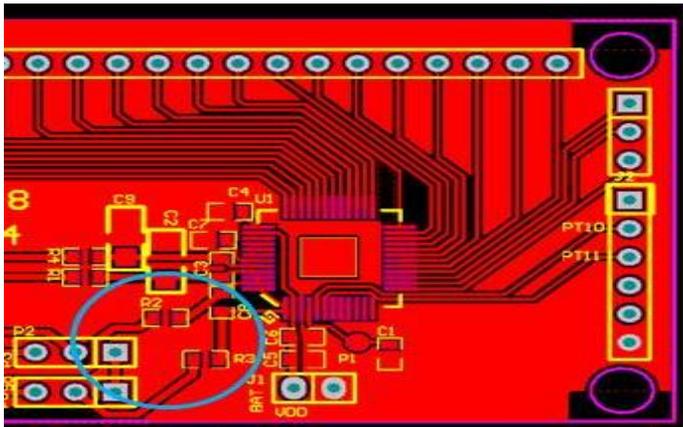


图 2

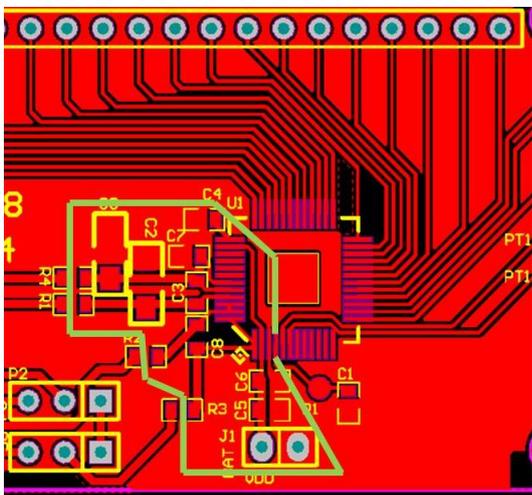


上图 1 与图 2 范例，此两图为相似的 PCB，但差一点在 IC 前是否有串连两个电阻(R2、R3)。

图 1 可以避免环形天线效应，图 1 为正确示范。

图 2 会有环形天线效应，图 2，从 R2、R3 电阻中间可以连到 GND，但是会导致还形天线效应产生。

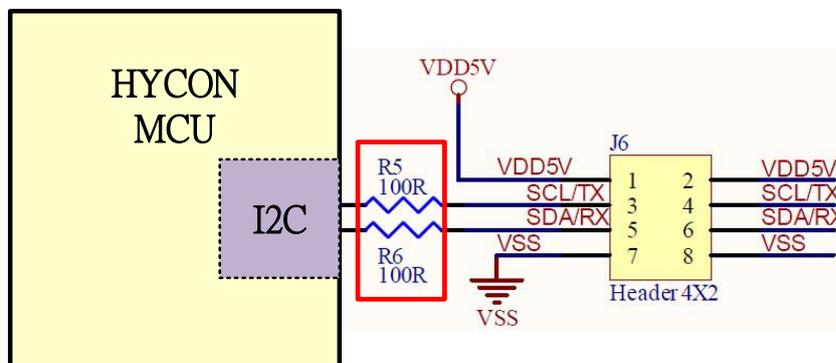
下图有串电阻的 PCB，VSS 产生了环形天线，路径如绿色线路描出。



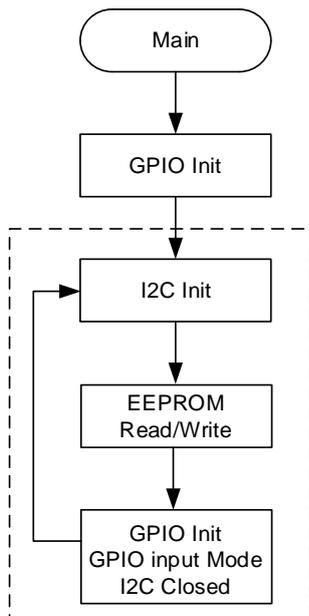
5.1.4. ESD 防护设计注意事项

ESD 的防护与机构设计与 PCB Layout 走线方式都有关联, 但是从 HY16F 的硬体程序做设定, 可以部份改善与强化 ESD 能力.

- 1). HY16F 主芯片的工作频率设置为 HAO/2 会比 HAO/1 抗 ESD 噪声来的好。
 - 2). 没有使用到的 GPIO pin, 最好都设定为 output low。
 - 3). 如果应用中有使用到 I2C Port, 可以着手从 I2C 硬件与软件设定做 ESD 防护强化。
- 硬件 : SCL 与 SDA 都串上 100 ohm 限流电阻(线流电阻加在靠近 HY16F 主芯片端)。



软件 : I2C 程序部分修改为, 在每一次的读取/写入完成 EEPROM 之后就把 I2C 的 IP Closed, 让 I/O 状态为 GPIO Input Mode, 只有在每一次读取/写入 EEPROM 之前, 才会再重做 I2C initial 动作。



6. HY16F 系列软件设计注意事项

此章节提供 HY16F 在软件&硬体设计时需要注意的事项.

6.1. HY16F 的 VDD3V 上电稳压判断

为了确保 HY16F 更稳定的工作, 在软件设计端, 建议针对 HY16F 的 VDD3V 加入电压稳压的判断(即 main 的第一行), 确保 VDD3V 稳定在系统的工作电压之后(建议 VDD3V=2.75V 以上), 程序再继续往下运行. 以下将提供范例, 利用 HY16F198B 的 CMP 比较器来做 VDD3V 电压稳压判断.

程序代码说明 : 芯片上电即利用函式 VDD3V_2_75_Check()来判断 VDD3V 是否大于 2.75V(利用变量 _timeout, 最多等待 VDD3V 上电的稳压时间约 8 秒钟)。如果 VDD3V 小于 2.75V, LCD 显示 0, 程序 while(1) 卡死 ; 如果 VDD3V 大于 2.75V, LCD 显示 1, 程序可继续往下执行。 以下为程序代码提供参考:

```
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
V02	All	2022/09/07	初版发行