



帶雙OPA Flash 單片機

FC50F898

目錄

特性.....	3
概述.....	4
方框圖	4
引腳圖	4
引腳說明	5
極限參數	6
直流電氣特性.....	7
交流電氣特性.....	7
輸入 / 輸出口電氣特性	8
記憶體電氣特性.....	9
LVR/LVD 電氣特性	9
A/D 轉換器電氣特性.....	9
OPA 電路電氣特性	10
內部參考電壓電氣特性	11
灌電流發生器電氣特性	11
上電復位電氣特性.....	11
系統結構	12
Flash 程式記憶體.....	13
資料記憶體	15
特殊功能寄存器.....	16
EEPROM 資料記憶體	18
振盪器	21
工作模式和系統時鐘	21
看門狗計時器.....	26
復位和初始化.....	27
輸入 / 輸出埠.....	30
計時器模組 – TM	34
標準型 TM – STM	36
A/D 轉換器	43
OPA 電路.....	49
灌電流發生器.....	55
通用序列介面模組 – USIM	56
低電壓檢測 – LVD	75
中斷.....	76
應用電路	82
封裝信息	83

特性

CPU 特性

- 工作電壓： $f_{SYS}=8\text{MHz}$ ：2.2V~5.5V
- $V_{DD}=5\text{V}$ ，系統時鐘為 8MHz 時，運算速度為 0.5 μs
- 提供暫停和喚醒功能，以降低功耗
- 振盪器類型：
 - 內部高速 8MHz RC – HIRC
 - 內部低速 32kHz RC – LIRC
- 多種工作模式：快速、低速、空閒和休眠
- 內部集成振盪器，無需外接元件
- 所有指令都可在 1 或 2 個運算速度內完成
- 查表指令
- 63 條指令
- 4 層堆疊
- 位元元操作指令

周邊特性

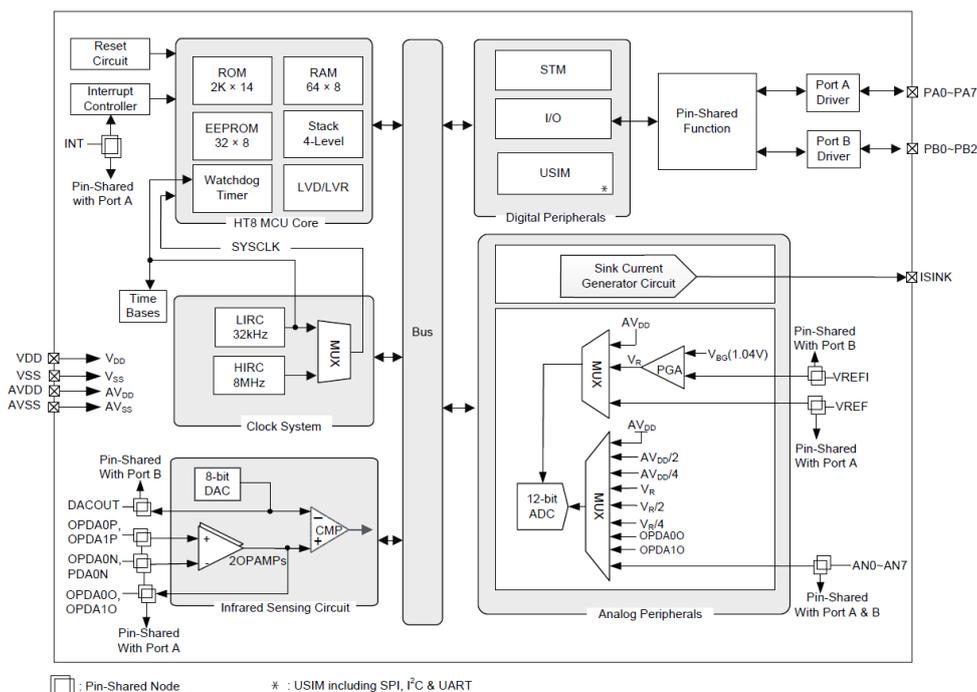
- Flash 程式記憶體：2K×14
- RAM 資料記憶體：64×8
- True EEPROM 記憶體：32×8
- 看門狗計時器功能
- 11 個雙向 I/O 口
- 1 個與 I/O 口共用的外部中斷
- 1 個 16 位元計時器模組用於時間測量、捕捉輸入、比較匹配輸出、PWM 輸出及單脈衝輸出功能
- 通用序列介面模組 – USIM，用於 SPI、I²C 或 UART 通信
- 雙時基功能可提供固定的中斷信號
- 8 個外部通道 12-bit 解析度 A/D 轉換器
- 恒定電流驅動電路（最大驅動電流 320mA）
- 兩個運算放大器
- 一個比較器
- 低電壓重定功能
- 低電壓檢測功能
- 封裝類型：8-pin SOP, 16-pin SOP

概述

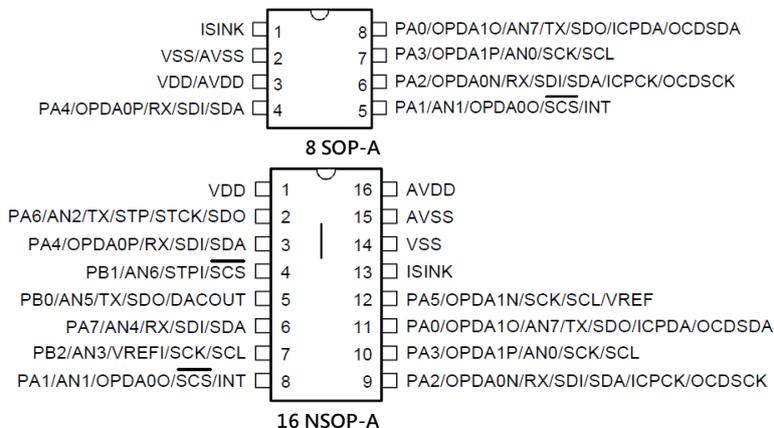
該單片機是一款 A/D 型具有 8 位元元高性能精簡指令集的 Flash 型單片機，專門為紅外線感應應用而設計。此單片機具有一系列的功能和特性，其 Flash 記憶體可多次程式設計的特性給用戶提供了極大的方便。記憶體方面，還包含了一個 RAM 資料記憶體和一個可用於存儲序號、校準資料等非易失性資料的 True EEPROM 記憶體。

在模擬特性方面，該單片機包含一個多通道12 位 A/D 轉換器，兩個運算放大器，一個比較器和其他專門為紅外線感應應用而設計的電路。極其靈活的計時器模組，可提供定時功能、脈衝產生功能及 PWM 產生功能。內建完整的 SPI, I²C 和 UART 介面功能，這種流行的介面為設計人員提供了易與外部硬體通信的方式。內部看門狗計時器、低電壓重定和低電壓檢測等保護特性，外加優秀的抗干擾和 ESD 保護性能，確保單片機在惡劣的電磁干擾環境下可靠地運行。該單片機內部集成了高 / 低速振盪器，在應用中不需要週邊元器件。使用不同的時鐘源在不同工作模式之間動態切換的能力，為用戶提供了一個優化單片機操作和減少功耗的能力。

方框圖



引腳圖



- 注：1. 若共用引腳同時有多種輸出，所需引腳共用功能通過相應的軟體控制位元決定
 2. 在較小封裝類型中可能含有未引出的引腳，需合理設置其狀態以避免輸入浮空造成額外耗電，詳見“待機電流注意事項”“輸入/輸出埠”章節。
 3. 在 8-pin SOP 封裝中，VDD/AVDD 指的是 VDD 和 AVDD 為同一引腳；VSS/AVSS 指的是 VSS 和 AVSS 為同一引腳。

引腳說明

除了電源引腳和一些相關的變壓器控制引腳外，該系列單片機的所有引腳都以它們的埠名稱進行標注，例如 PA0、PA1 等，用於描述這些引腳的數位輸入 / 輸出功能。然而，這些引腳也與其它功能共用，如模數轉換器、計時器模組引腳等。每個引腳的功能如下表所列。而引腳配置的詳細內容見規格書的其它章節。

下述引腳功能表格是參照最大封裝引腳，有部分引腳在較小封裝類型中未出現。

引腳名稱	功能	OPT	I/T	O/T	說明
PA0/OPDA10/AN7/ TX/SDO/ICPDA/ OCDSDA	PA0	PAPU PAWU PAS0	ST	CMOS	通用 I/O 口，可通過寄存器設置上拉電阻和喚醒功能
	OPDA10	PAS0	—	CMOS	OPAMP1 輸出
	AN7	PAS0	AN	—	A/D 轉換器外部輸入通道 7
	TX	PAS0	—	CMOS	UART 發送器引腳
	SDO	PAS0	—	CMOS	SPI 串列資料輸出
	ICPDA	—	ST	CMOS	ICP 位址 / 資料
	OCDSDA	—	ST	CMOS	OCDS 位址 / 資料，僅用於 EV 晶片
PA1/AN1/OPDA00/ SCS/INT	PA1	PAPU PAWU PAS0	ST	CMOS	通用 I/O 口，可通過寄存器設置上拉電阻和喚醒功能
	AN1	PAS0	AN	—	A/D 轉換器外部輸入通道 1
	OPDA00	PAS0	—	CMOS	OPAMP0 輸出
	SCS	PAS0	ST	CMOS	SPI 從機選擇
	INT	PAS0 INTEG INTCO	ST	—	外部中斷輸入
PA2/OPDA0N/RX/ SDI/SDA/ICPCK/ OCDSCK	PA2	PAPU PAWU PAS0	ST	CMOS	通用 I/O 口，可通過寄存器設置上拉電阻和喚醒功能
	OPDA0N	PAS0	ST	—	OPAMP0 反相輸入
	RX	PAS0	ST	—	UART 接收引腳
	SDI	PAS0	ST	—	SPI 串列資料登錄
	SDA	PAS0	ST	NMOS	I ² C 數據線
	ICPCK	—	ST	—	ICP 時鐘引腳
	OCDSCK	—	ST	—	OCDS 時鐘引腳，僅用於 EV 晶片
PA3/OPDA1P/AN0/ SCK/SCL	PA3	PAPU PAWU PAS0	ST	CMOS	通用 I/O 口，可通過寄存器設置上拉電阻和喚醒功能
	OPDA1P	PAS0	ST	—	OPAMP1 同相輸入
	AN0	PAS0	AN	—	A/D 轉換器外部輸入通道 0
	SCK	PAS0	ST	CMOS	SPI 串列時鐘
	SCL	PAS0	ST	NMOS	I ² C 時鐘線
PA4/OPDA0P/RX/ SDI/SDA	PA4	PAPU PAWU PAS1	ST	CMOS	通用 I/O 口，可通過寄存器設置上拉電阻和喚醒功能
	OPDA0P	PAS1	ST	—	OPAMP0 同相輸入
	RX	PAS1	ST	—	UART 接收引腳
	SDI	PAS1	ST	—	SPI 串列資料登錄
	SDA	PAS1	ST	NMOS	I ² C 數據線
ISINK	ISINK	—	—	CMOS	灌電流源
VDD	VDD	—	PWR	—	正電源
VSS	VSS	—	PWR	—	負電源，接地

引腳名稱	功能	OPT	I/T	O/T	說明
PA5/OPDA1N/SCK/ SCL/VREF	PA5	PAPU PAWU PAS1	ST	CMOS	通用 I/O 口 · 可通過寄存器設置上拉電阻和喚醒功能
	OPDA1N	PAS1	ST	—	OPAMP1 反相輸入
	SCK	PAS1	ST	CMOS	SPI 串列時鐘
	SCL	PAS1	ST	NMOS	I ² C 時鐘線
	VREF	PAS1	AN	—	A/D 轉換器外部參考電壓輸入
PA6/AN2/TX/STP STCK/SDO	PA6	PAPU PAWU PAS1	ST	CMOS	通用 I/O 口 · 可通過寄存器設置上拉電阻和喚醒功能
	AN2	PAS1	AN	—	A/D 轉換器外部輸入 2
	TX	PAS1	—	CMOS	UART 發送引腳
	STP	PAS1	—	CMOS	STM 輸出
	STCK	PAS1	ST	—	STM 時鐘輸入
	SDO	PAS1	—	CMOS	SPI 串列資料輸出
PA7/AN4/RX/SDI/ SDA	PA7	PAPU PAWU PAS1	ST	CMOS	通用 I/O 口 · 可通過寄存器設置上拉電阻和喚醒功能
	AN4	PAS1	AN	—	A/D 轉換器外部輸入 4
	RX	PAS1	ST	—	UART 接收引腳
	SDI	PAS1	ST	—	SPI 串列資料登錄
	SDA	PAS1	ST	NMOS	I ² C 數據線
PB0/AN5/TX/SDO/ DACOUT	PB0	PBS0	ST	CMOS	通用 I/O 口 · 可通過寄存器設置上拉電阻和喚醒功能
	AN5	PBS0	AN	—	A/D 轉換器外部輸入 5
	TX	PBS0	—	CMOS	UART 發送引腳
	SDO	PBS0	—	CMOS	SPI 串列資料輸出
	DACOUT	PBS0	—	CMOS	D/A 轉換器輸出
PB1/AN6/SCS/STPI	PB1	PBS0	ST	CMOS	通用 I/O 口 · 可通過寄存器設置上拉電阻
	AN6	PBS0	AN	—	A/D 轉換器外部輸入 6
	SCS	PBS0	ST	CMOS	SPI 從機選擇
	STPI	PBS0	ST	—	STM 捕捉輸入
PB2/AN3/VREFI/ CK/SCL	PB2	PBS0	ST	CMOS	通用 I/O 口 · 可通過寄存器設置上拉電阻
	AN3	PBS0	AN	—	A/D 轉換器外部輸入 3
	VREFI	PBS0	AN	—	A/D 轉換器 PGA 輸入
	SCK	PBS0	ST	CMOS	SPI 串列時鐘
	SCL	PBS0	ST	NMOS	I ² C 時鐘線
AVDD	AVDD	—	PWR	—	模擬正電源
AVSS	AVSS	—	PWR	—	模擬負電源 · 接地

注：I/T：輸入類型； O/T：輸出類型； OPT：通過寄存器選項來配置； PWR：電源；
ST：施密特觸發輸入； CMOS：CMOS 輸出； NMOS：NMOS 輸出； AN：類比信號。

極限參數

電源供應電壓.....	VSS-0.3V ~ +6.0V
輸入電壓.....	VSS-0.3V ~ VDD+0.3V
儲存溫度.....	-50°C ~ 125°C
工作溫度.....	-40°C ~ 85°C
IOL 總電流.....	80mA
IOH 總電流.....	-80mA
總功耗.....	500mW

注：這裡只強調額定功率，超過極限參數所規定的範圍將對晶片造成損害，無法預期晶片在上述標示範圍外的工作狀態，而且若長期在標示範圍外的條件下工作，可能影響晶片的可靠性。

直流電氣特性

以下表格中參數測量結果可能受多個因素影響，如振盪器類型、工作電壓、工作頻率、引腳負載狀況、溫度和程式指令等。

工作電壓特性

Ta = -40°C ~ 85°C

符號	參數	測試條件	最小	典型	最大	單位
VDD	工作電壓 – HIRC	fSYS = 8MHz	2.2	—	5.5	V
	工作電壓 – LIRC	fSYS = 32kHz	2.2	—	5.5	V

待機電流特性

Ta = -40°C ~ 85°C，除非另有說明

符號	待機模式	測試條件		最小	典型	最大	最大 @85°C	單位			
		VDD	條件								
ISTB	休眠模式	2.2V	WDT off	—	0.2	0.6	0.8	μA			
		3V		—	0.2	0.8	1				
		5V		—	0.5	1	1.4				
		2.2V		WDT on	—	2.4	4		4.8	μA	
		3V			—	3	5		6		
		5V			—	5	10		12		
	空間模式 0 – LIRC	2.2V	fSUB on	—	2.4	4	4.8	μA			
		3V		—	3	5	6				
		5V		—	5	10	12				
		空間模式 1 – HIRC		2.2V	fSUB on · fSYS = 8MHz	—	288		400	480	μA
				3V		—	360		500	600	
				5V		—	600		800	960	

注：當使用該表格電氣特性資料時，以下幾點需要注意：

1. 任何數位輸入都設置為非浮空的狀態。
2. 所有測量都在無負載且所有週邊功能關閉的條件下進行。
3. 無直流電流路徑。所有待機電流數值都是在 HALT 指令執行後測得，因此 HALT 後停止執行所有指令。

工作電流特性

Ta = -40°C ~ 85°C

符號	工作模式	測試條件		最小	典型	最大	單位
		VDD	條件				
IDD	低速模式 – LIRC	2.2V	fSYS = 32kHz	—	8	16	μA
		3V		—	10	20	
		5V		—	30	50	
	快速模式 – HIRC	2.2V	fSYS = 8MHz	—	0.6	1.0	mA
		3V		—	0.8	1.2	
		5V		—	1.6	2.4	

注：當使用該表格電氣特性資料時，以下幾點需要注意：

1. 任何數位輸入都設置為非浮空的狀態。
2. 所有測量都在無負載且所有週邊功能關閉的條件下進行。
3. 無直流電流路徑。
4. 所有工作電流數值通過一個連續的 NOP 指令迴圈程式測得。

交流電氣特性

以下表格中參數測量結果可能受多個因素影響，如振盪器類型、工作電壓、工作頻率和溫度等等。

內部高速振盪器 – HIRC – 頻率精確度

程式燒錄時，燒錄器會調整 HIRC 振盪器使其工作在使用者選擇的 HIRC 頻率和工作電壓 (3V 或 5V) 條件下。

符號	參數	測試條件		最小	典型	最大	單位
		VDD	溫度				
fHIRC	通過燒錄器調整後的 8MHz HIRC 頻率	3V/5V	25°C	-1%	8	+1%	MHz
			-40°C~85°C	-2%	8	+2%	
		2.2V~5.5V	25°C	-2.5%	8	+2.5%	
			-40°C~85°C	-3%	8	+3%	

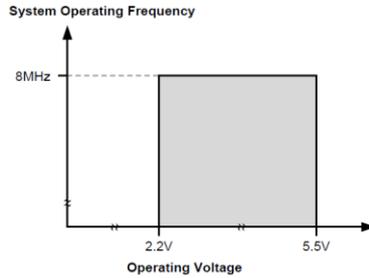
注：1. 燒錄器可在 3V/5V 這兩個可選的固定電壓下對 HIRC 頻率進行調整，在此提供 VDD=3V/5V 時的參數值。

2. 3V/5V 表格列下面提供的是全壓條件下的參數值。對於電壓範圍在 2.2V~3.6V 的應用，建議調整電壓固定在 3V；對於電壓範圍在 3.3V~5.5V 的應用，建議調整電壓固定在 5V。

內部低速振盪器電氣特性 – LIRC

符號	參數	測試條件		最小	典型	最大	單位
		VDD	溫度				
fLIRC	LIRC 頻率	5V	25°C	25.6	32	38.4	kHz
		1.8V~5.5V	25°C	12.8	32	41.6	
			-40°C ~ 85°C	8	32	60	
tSTART	LIRC 啟動時間	—	25°C	—	—	100	µs

工作頻率特性曲線



系統上電時間電氣特性

Ta=-40°C~85°C

符號	參數	測試條件		最小	典型	最大	單位
		VDD	條件				
tSST	系統啟動時間 (從 fSYS Off 條件下喚醒)	—	fSYS = fH ~ fH/64, fH = fHIRC	—	16	—	tHIRC
		—	fSYS = fSUB = fLIRC	—	2	—	tLIRC
	系統啟動時間 (從 fSYS On 條件下喚醒)	—	fSYS = fH ~ fH/64, fH = fHIRC	—	2	—	tH
		—	fSYS = fSUB = fLIRC	—	2	—	tSUB
tRSTD	系統速度切換時間 (快速模式 → 低速模式或低速模式 → 快速模式)	—	fHIRC off → on	—	16	—	tHIRC
	系統重定延遲時間 (上電重定 · LVR 硬體重定)	—	RRPOR = 5V/ms	25	50	100	ms
	系統重定延遲時間 (WDTC 軟體重定)	—	—				
系統重定延遲時間 (WDT 溢出復位)	—	—	8.3	16.7	33.3	ms	
tSRESET	軟體重定最小脈寬	—	—	45	90	375	µs

注：1. 系統啟動時間裡提到的 fSYS on/off 狀態取決於工作模式類型以及所選的系統時鐘振盪器。更多相關細節請參考系統工作模式章節。

2. tHIRC 等符號所表示的時間單位，是對應頻率值的倒數，相關頻率值在前面表格有說明。例如，tHIRC=1/fHIRC，tSYS=1/fSYS 等等。

3. 若 LIRC 被選擇作為系統時鐘源且在休眠模式下 LIRC 關閉，則上面表格中對應 tSST 數值還需加上 LIRC 頻率表裡提供的 LIRC 啟動時間 tSTART。系統速度切換時間實際上是指新使能的振盪器的啟動時間。

輸入 / 輸出電氣特性

Ta=-40°C~85°C

符號	參數	測試條件		最小	典型	最大	單位
		VDD	條件				
VIL	I/O 口低電平輸入電壓	5V	—	0	—	1.5	V
		—		0	—	0.2VDD	
VIH	I/O 口高電平輸入電壓	5V	—	3.5	—	5.0	V
		—		0.8VDD	—	VDD	
IOL	I/O 口灌電流	3V	VOL = 0.1VDD	16	32	—	mA
		5V		32	65	—	
IOH	I/O 口源電流	3V	VOH = 0.9VDD	-4	-8	—	mA
		5V		-8	-16	—	
RPH	I/O 口上拉電阻 (注)	3V	—	20	60	100	kΩ
		5V		10	30	50	
I _{LEAK}	輸入漏電流	5V	VIN = VDD 或 VIN = VSS	—	—	±1	µA
t _{CK}	STCK 輸入引腳最小脈寬	—	—	0.3	—	—	µs
t _{TPI}	STPI 輸入引腳最小脈寬	—	—	0.3	—	—	µs
t _{INT}	外部中斷引腳最小脈寬	—	—	10	—	—	µs

注：RPH 內部上拉電阻值的計算方法是：將引腳接地並設置為輸入且使能上拉電阻功能，然後在特定電源電壓下測量該引腳上電流，最後電壓除以測量的電流值從而得到此上拉電阻值。

記憶體電氣特性

Ta=-40°C~85°C · 除非另有說明

符號	參數	測試條件		最小	典型	最大	單位
		VDD	條件				
VRW	讀 / 寫工作電壓	—	—	VDDmin	—	VDDmax	V
Flash 程式 / 資料 EEPROM 記憶體							
tDEW	擦除 / 寫週期時間 - Flash 程式記憶體	—	—	—	2	3	ms
	寫週期時間 - 資料 EEPROM 記憶體	—	—	—	4	6	ms
IDDPGM	VDD 電壓下燒錄 / 擦除電流	—	—	—	—	5.0	mA
EP	電容耐久性	—	—	100K	—	—	E/W
tRETD	程式記憶體資料保存時間	—	Ta = 25°C	—	40	—	Year
RAM 資料記憶體							
VDR	RAM 資料保存電壓	—	單片機處於休眠模式	1.0	—	—	V

LVR/LVD 電氣特性

Ta=-40°C~85°C

符號	參數	測試條件		最小	典型	最大	單位	
		VDD	條件					
VLVR	低電壓重定電壓	—	LVR 使能	-5%	2.1	+5%	V	
VLVD	低電壓檢測電壓	—	LVD 使能 · 電壓選擇 2.0V	-5%	2.0	+5%	V	
			LVD 使能 · 電壓選擇 2.2V		2.2			
			LVD 使能 · 電壓選擇 2.4V		2.4			
			LVD 使能 · 電壓選擇 2.7V		2.7			
			LVD 使能 · 電壓選擇 3.0V		3.0			
			LVD 使能 · 電壓選擇 3.3V		3.3			
			LVD 使能 · 電壓選擇 4.0V		4.0			
ILVRLVDBG	工作電流	3V	LVD 使能 · LVR 使能 · VBGEN = 0	—	—	20	μA	
		5V	VBGEN = 0	—	20	25		
		3V	LVD 使能 · LVR 使能 · VBGEN = 1	—	—	25		30
		5V	VBGEN = 1	—	25	30		
tLVDS	LVDO 穩定時間	—	LVR 使能 · VBGEN = 0 · LVD off → on	—	—	18	μs	
tLVR	產生 LVR 重定的低電壓 最短保持時間	—	—	140	600	1000	μs	
tLVD	產生 LDR 中斷的低電 壓最短保持時間	—	—	40	150	320	μs	

A/D 轉換器電氣特性

Ta=-40°C~85°C

符號	參數	測試條件		最小	典型	最大	單位
		VDD	條件				
VADI	A/D 轉換器輸入電壓	—	—	0	—	VREF	V
VREF	A/D 轉換器參考電壓	—	—	2	—	AVDD	V
DNL	非線性微分誤差	3V	SAINS[3:0]=0000B, SAVRS[1:0]=01B, VREF=AVDD, tADCK=0.5μs	-3	—	+3	LSB
		5V	VREF=AVDD, tADCK=0.5μs				
		3V	SAINS[3:0]=0000B, SAVRS[1:0]=01B, VREF=AVDD, tADCK=10μs				
		5V	VREF=AVDD, tADCK=10μs				
INL	非線性積分誤差	3V	SAINS[3:0]=0000B, SAVRS[1:0]=01B, VREF=AVDD · tADCK=0.5μs	-4	—	+4	LSB
		5V	VREF=AVDD · tADCK=0.5μs				
		3V	SAINS[3:0]=0000B, SAVRS[1:0]=01B, VREF=AVDD, tADCK=10μs				
		5V	VREF=AVDD, tADCK=10μs				
IADC	A/D 轉換器使能的額外電流	3V	無負載 · tADCK=0.5μs	—	0.2	0.4	mA
		5V		—	0.3	0.6	
tADCK	A/D 轉換器時鐘週期	5V	—	0.5	—	10	μs
tON2ST	A/D 轉換器 On-to-Start 時間	—	—	4	—	—	μs
tADS	A/D 轉換器採樣時間	—	—	—	4	—	tADCK
tADC	A/D 轉換時間(包括採樣和保持時間)	—	—	—	16	—	tADCK
GERR	A/D 轉換增益誤差	3V	SAINS[3:0]=0000B, SAVRS[1:0]=01B, VREF=AVDD	-4	—	4	LSB
		5V					
OSRR	A/D 轉換 Offset 誤差	3V	SAINS[3:0]=0000B, SAVRS[1:0]=01B, VREF=AVDD	-4	—	4	LSB
		5V					
IPGA	PGA 使能額外電流	3V	無負載	—	250	500	μA
		5V					

VOR	PGA 最大輸出電壓範圍	3V	—	AVSS +0.1	—		V
		5V					
Ga	PGA 增益精確度	—	增益 = 1, VRI > 0.1V 增益 = 2, 3, 4, VRI > 0.05V	-5	—	+5	%
VIR	PGA 輸入電壓範圍	3V	增益 = 1, Ga < ±5%	AVSS +0.1	—	AVDD -1.4	V
		5V					
		3V	增益 = 2, 3, 4, Ga < ±5%	AVSS +0.05	—	VOR(Max) /Gain	
		5V					

OPA 電路電氣特性

運算放大器電氣特性

Ta = -40°C ~ 85°C

符號	參數	測試條件		最小	典型	最大	單位
		VDD	條件				
VDD	工作電壓	—	—	2.2	5.0	5.5	V
IOPA	OPA 使能的額外電流	—	無負載 · OPDAmBW=0 (m=0, 1)	—	80	128	μA
			無負載 · OPDAmBW=1 (m=0, 1)	—	200	320	
VOS	輸入失調電壓	5V	未校準 · OPDAmOF[5:0]=100000B, (m=0, 1)	-15	—	+15	mV
			校準後	-2	—	+2	
IOS	輸入失調電流	5V	VIN=1/2VCM	—	1	10	nA
VCM	共模電壓範圍	—	—	VSS+0.3	—	VDD-1.4	V
PSRR	電源電壓抑制比	5V	—	58	70	—	dB
CMRR	共模抑制比	5V	—	58	80	—	dB
AOL	開環增益	—	—	58	80	—	dB
SR	轉換速度	5V	RLOAD=1MΩ, CLOAD=60pF, OPDAmBW=0 (m=0, 1)	200	500	—	V/ms
			RLOAD=1MΩ, CLOAD=60pF, OPDAmBW=1 (m=0, 1)	720	1200	—	
GBW	增益頻寬	5V	RLOAD=1MΩ, CLOAD=100pF, OPDAmBW=0 (m=0, 1)	400	600	—	kHz
			RLOAD=1MΩ, CLOAD=100pF, OPDAmBW=1 (m=0, 1)	1300	2000	—	
VOR	最大輸出電壓範圍	3V	—	VSS+0.1	—	VDD-0.1	V
		5V					

比較器電氣特性

Ta = -40°C ~ 85°C

符號	參數	測試條件		最小	典型	最大	單位
		VDD	條件				
VDD	工作電壓	—	—	2.2	5.0	5.5	V
ICMP	比較器使能的額外電流	—	無負載 · OPDIS[1:0]=00B	—	1.7	2.7	μA
		—	無負載 · OPDIS[1:0]=01B	—	14	22	
		—	無負載 · OPDIS[1:0]=10B	—	36	57	
		—	無負載 · OPDIS[1:0]=11B	—	58	92	
VOS	輸入失調電壓	5V	未校準(OPDCOF[4:0]=10000B)	-10	—	+10	mV
			未校準	-4	—	+4	
VCM	共模電壓範圍	—	—	VSS	—	VDD-1.4	V
tRP	回應時間	3V	10mV 偏置 · CLOAD=3pF, OPDIS[1:0]=00B	—	—	35	μs
		5V					
		3V	10mV 偏置 · CLOAD=3pF, OPDIS[1:0]=01B	—	—	2.5	
		5V					
		3V	10mV 偏置 · CLOAD=3pF, OPDIS[1:0]=10B	—	—	1	
		5V					
3V	10mV 偏置 · CLOAD=3pF, OPDIS[1:0]=11B	—	—	0.7			
5V							
VHYS	遲滯寬度	3V	OPDHYS[1:0]=00B, OPDIS[1:0]=00B	0	0	5	mV
		5V					
		3V	OPDHYS[1:0]=01B, OPDIS[1:0]=01B	20	40	60	
		5V					
		3V	OPDHYS[1:0]=10B, OPDIS[1:0]=10B	50	100	150	
		5V					
3V	OPDHYS[1:0]=11B, OPDIS[1:0]=11B	80	160	240			
5V							

注：所有的測量都在比較器輸入電壓等於 (VDD-1.4)/2 · 且保持不變的條件下進行。

D/A 轉換器電氣特性

Ta=-40°C~85°C

符號	參數	測試條件		最小	典型	最大	單位
		VDD	條件				
VDD	工作電壓	—	—	2.2	5.0	5.5	V
VDACO	輸出電壓範圍	—	—	VSS	—	VREF	V
VREF	參考電壓	—	—	VDD			V
IDAC	D/A 轉換器使能的額外電流	3V	—	—	—	200	μA
		5V		—	—	280	
tST	建立時間	3V	CLOAD=50pF	—	—	7	μs
		5V		—	—	6	
DNL	非線性微分誤差	3V	VREF=VDD	-1	—	+1	LSB
		5V					
INL	非線性積分誤差	3V	VREF=VDD	-1.5	—	+1.5	LSB
		5V					
Ro	R2R 輸出電阻	3V	—	—	20	—	kΩ
		5V					
OSRR	失調誤差	3V	—	—	—	6	mV
		5V		—	—	10	
GERR	增益誤差	3V	—	—	—	12	mV

內部參考電壓電氣特性

Ta=-40°C~85°C

符號	參數	測試條件		最小	典型	最大	單位
		VDD	條件				
VBG	Bandgap 參考電壓	—	—	-5%	1.04	+5%	V

注：VBG 電壓可以用作 A/D 轉換器 PGA 輸入信號。

灌電流發生器電氣特性

Ta=-40°C~85°C

符號	參數	測試條件		最小	典型	最大	單位
		VDD	條件				
VSINK	ISINK 引腳輸入電壓	—	—	1.0	—	5.5	V
ISINK	ISINK 引腳灌電流	3V	(調整後) VSINK=3V · IDATA[5:0]=000000B	-5%	5	+5%	mA
		2.2V~5.5V	(調整後) Ta=-40°C~85°C 1.0V≤VSINK≤4.5V · IDATA[5:0]=000000B	-15%	5	+15%	
		2.2V~5.5V	(調整後) Ta=-40°C~85°C 1.0V≤VSINK≤4.5V · IDATA[5:0]=111111B	-15%	320	+15%	
%/DVSINK	輸出電流與輸出電壓調整率	3V	1.0V≤VSINK≤5.5V · ISINK=5mA	—	±1.0	±6.0	%/V
%/DVDD	輸出電流與電源電壓調整率	2.2V~5.5V	VSINK=1.0V	—	±1.0	±8.0	%/V

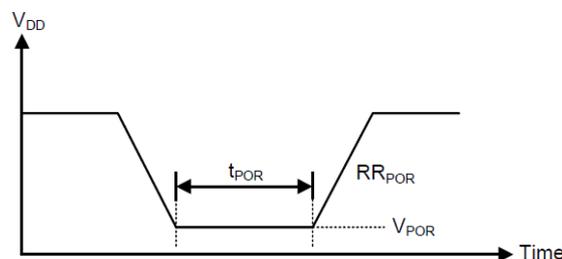
注：1. %/DVSINK=(ISINK1-ISINK2)/ISINK3×100%/(5.5V-1.0V) · ISINK1 在 VSINK=5.5V · VDD=3.0V 時測得；ISINK2 在 VSINK=1.0V · VDD=3.0V 時測得；ISINK3 在 VSINK=3.0V · VDD=3.0V 時測得。

2. %/DVDD=(ISINK1-ISINK2)/ISINK3×100%/(5.5V-2.2V) · ISINK1 在 VDD=5.5V · VSINK=1.0V 時測得；ISINK2 在 VDD=2.2V · VSINK=1.0V 時測得；ISINK3 在 VDD=3.0V · VSINK=1.0V 時測得。

上電復位電氣特性

Ta=-40°C~85°C

符號	參數	測試條件		最小	典型	最大	單位
		VDD	條件				
VPOR	上電重定電壓	—	—	—	—	100	mV
RRPOR	上電重定電壓速率	—	—	0.035	—	—	V/ms
tPOR	VDD 保持為 VPOR 的最小時間	—	—	1	—	—	ms

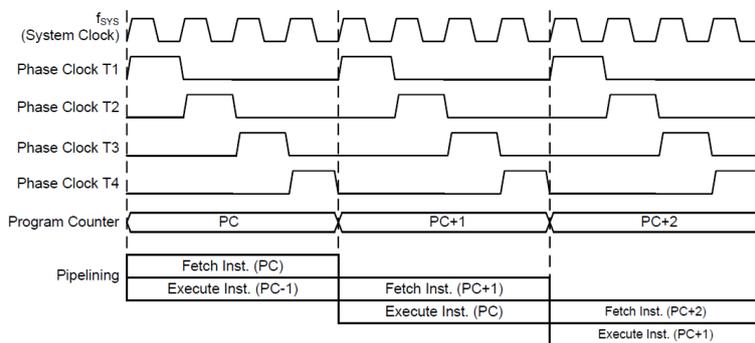


系統結構

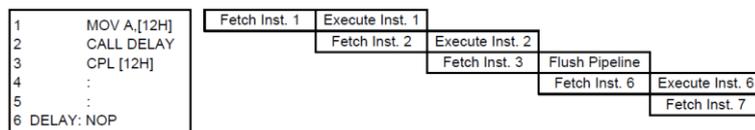
內部系統結構是 Holtek 單片機具有良好性能的主要因素。由於採用RISC 結構，該單片機具有高運算速度和高性能的特點。通過流水線的方式，指令的取得和執行同時進行，此舉使得除了跳轉和調用指令外，其它指令都能在一個運算速度內完成。8 位元元 ALU 參與指令集中所有的運算，它可完成算數運算、邏輯運算、移位元、遞增、遞減和分支等功能，而內部的資料路徑則是以通過累加器和 ALU 的方式加以簡化。有些寄存器在資料記憶體中被實現，且可以直接或間接地址。簡單的寄存器定址方式和結構特性，確保了在提供具有最大可靠性和靈活性的I/O 和 A/D 控制系統時，僅需要少數的外部器件。使得該單片機適用於低成本和大量生產的控制應用。

時序和流水線結構主系統時鐘由 HIRC 或 LIRC 振盪器提供，它被細分為 T1~T4 四個內部產生的非重疊時序。在 T1 時間，程式計數器自動加一併抓取一條新的指令。剩下的時間 T2~T4 完成解碼和執行功能，因此，一個 T1~T4 時鐘週期構成一個運算速度。雖然指令的抓取和執行發生在連續的運算速度，但單片機流水線結構會保證指令在一個運算速度內被有效執行。除非程式計數器的內容被改變，如副程式的調用或跳轉，在這種情況下指令將需要多一個運算速度的時間去執行。

如果指令牽涉到分支，例如跳轉或調用等指令，則需要兩個運算速度才能完成指令執行。需要一個額外週期的原因是程式先用一個週期取出實際要跳轉或調用的位址，再用另一個週期去實際執行分支動作，因此使用者需要特別考慮額外週期的問題，尤其是在執行時間要求較嚴格的時候。



系統時序和流水線



指令捕捉

程式計數器

在程式執行期間，程式計數器用來指向下一個要執行的指令位元址。除了 “JMP” 和 “CALL” 指令需要跳轉到一個非連續的程式記憶體位址之外，它會在每條指令執行完成以後自動加一。只有較低的 8 位元，即所謂的程式計數器低位元組寄存器 PCL，可以被用戶直接讀寫。

當執行的指令要求跳轉到不連續的位址時，如跳轉指令、副程式調用、中斷或重定等，單片機通過載入所需要的位元址到程式寄存器來控制程式，對於條件跳轉指令，一旦條件符合，在當前指令執行時取得的下一條指令將會被捨棄，而由一個空運算速度來取代。

程式計數器	
高位元組	低位元組 (PCL)
PC10~PC8	PCL7~PCL0

程式計數器

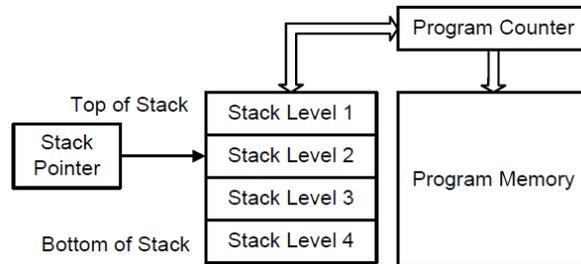
程式計數器的低位元組，即程式計數器的低位元組寄存器 PCL，可以通過程式控制，且它是可以讀取和寫入的寄存器。通過直接寫入資料到這個寄存器，一個程式短跳轉可直接執行，然而只有低位元組的操作是有效的，跳轉被限制在記憶體的當前頁中，即 256 個記憶體位址範圍內，當這樣一個程式跳轉要執行時，會插入一個空運算速度。程式計數器的低位元組可由程式直接進行讀取，PCL 的使用可能引起程式跳轉，因此需要額外的運算速度。

堆疊

堆疊是一個特殊的存儲空間，用來存儲程式計數器中的內容。該單片機有 4 層堆疊，堆疊既不是資料部分也不是程式空間部分，而且它既不是可讀取也不是可寫入的。當前層由堆疊指標 (SP) 加以指示，同樣也是不可讀寫的。在副程式調用或中斷回應服務時，程式計數器的內容被壓入到堆疊中。當副程式或中斷回應結束時，返回指令 (RET 或 RETI) 使程式計數器從堆疊中重新得到它以前的值。當一個晶片重定後，堆疊指標將指向堆疊頂部。

如果堆疊已滿，且有非遮罩的中斷發生，插斷要求標誌會被置位元，但中斷回應將被禁止。當堆疊指標減少 (執行 RET 或 RETI)，中斷將被回應。這個特性提供程式設計者簡單的方法來預防堆疊溢位。然而即使堆疊已滿，CALL 指令仍然可以被執行，而造成堆疊溢位。使用時應避免堆疊溢位的情況發生，因為這可能導致不可預期的程式分支指令執行錯誤。

若堆疊溢位，則首個存入堆疊的程式計數器資料將會丟失。



算數邏輯單位 – ALU

算數邏輯單位是單片機中很重要的部分，執行指令集中的算術和邏輯運算。

ALU 連接到單片機的資料匯流排，在接收相關的指令碼後執行需要的算術與邏輯操作，並將結果存儲在指定的寄存器，當 ALU 計算或操作時，可能導致進位元、借位或其它狀態的改變，而相關的狀態寄存器會因此更新內容以顯示這些改變，

ALU 所提供的功能如下：

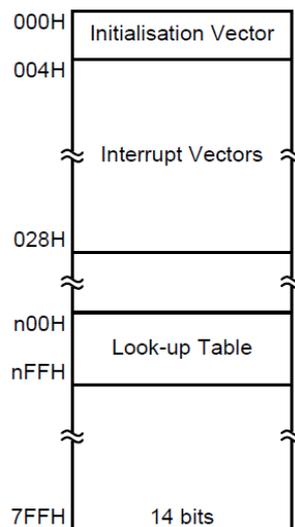
- 算數運算：ADD · ADDM · ADC · ADCM · SUB · SUBM · SBC · SBCM · DAA
- 邏輯運算：AND · OR · XOR · ANDM · ORM · XORM · CPL · CPLA
- 移位運算：RRA · RR · RRCA · RRC · RLA · RL · RLCA · RLC
- 遞增和遞減：INCA · INC · DECA · DEC
- 分支判斷：JMP · SZ · SZA · SNZ · SJZ · SDZ · SIZA · SDZA · CALL · RET · RETI

Flash 程式記憶體

程式記憶體用來存放使用者代碼即儲存程式。程式記憶體為 Flash 類型意味著可以多次重複程式設計，方便使用者使用同一晶片進行程式的修改。使用適當的單片機程式設計工具，此單片機提供用戶靈活便利的調試方法和專案開發規劃及更新。

結構

程式記憶體的容量為 2K×14 位元，程式記憶體用程式計數器來定址，其中也包含資料、表格和中斷入口。資料表格可以設定在程式記憶體的任何位元址，由表格指標來定址。



程式記憶體結構

特殊向量

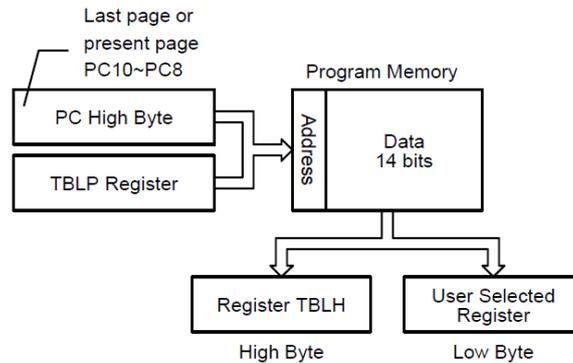
程式記憶體內部某些位址保留用做諸如復位和中斷入口等特殊用途。地址0000H 是晶片重定後的程式起始位址。在晶片重定之後，程式將跳到這個位址並開始執行。

查表

程式記憶體中的任何位址都可以定義成一個表格，以便儲存固定的資料。使用表格時，表格指標必須先行設定，其方式是將表格的位元址放在表格指標寄存器TBLP 中。此寄存器定義表格低 8 位元元總的位址。

在設定完表格指標後，表格資料可以使用 “TABRD [m]” 或 “TABRDL [m]” 指令分別從程式記憶體查表讀取。當這個指令執行時，程式記憶體中表格資料低位元元組，將被傳送到使用者所指定的資料記憶體 [m]，程式記憶體中表格資料的高位元組，則被傳送到 TBLH 特殊寄存器，而高位元組中未使用的位元將被讀取為 “0”。

下圖是查表中定址 / 資料流程程：



查表範例

以下範例說明表格指標和表格資料如何被定義和執行。這個例子使用的表格資料用 ORG 虛擬指令儲存在記憶體中。ORG 指令的值 “0700H” 指向的位址是2K 程式記憶體中最後一頁的起始位址。表格指標低位元元組寄存器的初始值設為06H，這可保證從資料表格讀取的第一筆資料位於程式記憶體位址 “0706H”，即最後一頁起始位址後的第六個位址。值得注意的是，假如 “TABRD [m]” 指令被使用，則表格指標指向當前頁中 TBLP 所指定的地址。在這個例子中，表格資料的高位元組等於零，而當 “TABRD [m]” 指令被執行時，此值將會自動的被傳送到 TBLH 寄存器。

TBLH 寄存器為唯讀寄存器，不能重新儲存，若主程序和中斷服務程式都使用表格讀取指令，應該注意它的保護。使用表格讀取指令，中斷服務程式可能會改變 TBLH 的值，若隨後在主程序中再次使用這個值，則會發生錯誤，因此建議避免同時使用表格讀取指令。然而在某些情況下，如果同時使用表格讀取指令是不可避免的，則在執行任何主程序的表格讀取指令前，中斷應該先除能，另外要注意的是所有與表格相關的指令，都需要兩個運算速度去完成操作。

表格讀取程式範例

```

tempreg1 db ?      ; temporary register #1
tempreg2 db ?      ; temporary register #2
:
:
mov a,06h          ; initialise low table pointer - note that this address
                  ; is referenced
mov tblp,a         ; to the last page or the present page
:
:
tabrdl tempreg1    ; transfers value in table referenced by table pointer
                  ; data at program memory address "0706H" transferred to
                  ; tempreg1 and TBLH
dec tblp           ; reduce value of table pointer by one
tabrdl tempreg2    ; transfers value in table referenced by table pointer
                  ; data at program memory address "0705H" transferred to
                  ; tempreg2 and TBLH
                  ; in this example the data "1AH" is transferred to
                  ; tempreg1 and data "0FH" to register tempreg2
                  ; the value "00H" will be transferred to the high byte
                  ; registerregister TBLH
:
:
org 0700h          ; sets initial address of program memory
dc 00Ah, 00Bh, 00Ch, 00Dh, 00Eh, 00Fh, 01Ah, 01Bh
:

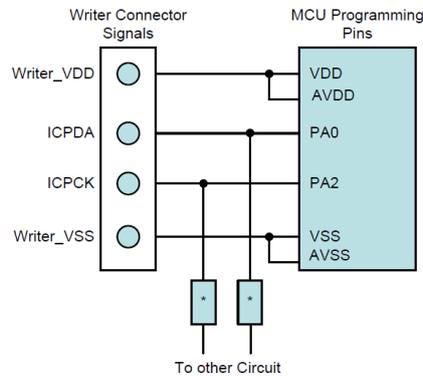
```

線上燒錄 – ICP

Flash 型程式記憶體提供使用者便利地對同一晶片進行程式的更新和修改。另外，Holtek 單片機提供 4 線介面的線上燒錄方式。用戶可將進行過燒錄或未經過燒錄的單片機晶片連同電路板一起製成，最後階段進行程式的更新和程式的燒錄，在無需去除或重新插入晶片的情況下方便地保持程式為最新版。Holtek Flash 單片機與線上燒錄器引腳的對應表如下：

Holtek 燒錄器引腳名稱	MCU 線上燒錄引腳名稱	引腳說明
ICPDA	PA0	串列資料 / 位址燒錄
ICPCK	PA2	時鐘燒錄
VDD	VDD&AVDD	電源
VSS	VSS&AVSS	地

程式記憶體可以通過 4 線的介面線上進行燒錄。其中一個引腳用於資料串列下載或上傳、另一個引腳用於串列時鐘、兩條用於提供電源。晶片線上燒寫的詳細使用說明超出此文檔的描述範圍，將由專門的參考文獻提供。燒錄過程中，用戶必須確保 ICPDA 和 ICPCK 這兩個引腳沒有連接至其它輸出腳。



注：* 可能為電阻或電容。若為電阻則其值必須大於 1kΩ，若為電容則其必須小於 1nF。

資料記憶體

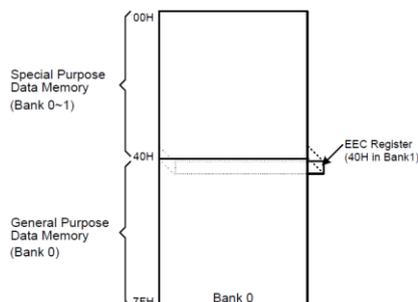
資料記憶體是內容可更改的 8 位元 RAM 內部記憶體，用來儲存臨時資料。資料記憶體分為兩個部分，第一部分是特殊功能資料記憶體。這些寄存器有固定的位址且與單片機的正确操作密切相關。大多特殊功能寄存器都可在程式控制下直接讀取和寫入，但有些被加以保護而不對用戶開放。第二部分資料記憶體是做一般用途使用，都可在程式控制下進行讀取和寫入。

結構

總的資料記憶體被分為兩個 Bank。大部分特殊功能資料寄存器均可在所有 Bank0 被訪問，除了 EEC 寄存器只位於 Bank 1 的“40H”地址。切換不同區域可通過設置存儲區指標實現。單片機的資料記憶體的起始位址是“00H”。特殊功能資料記憶體地址範圍為 00H~3FH，而通用資料記憶體地址範圍為 40H~7FH。

特殊功能資料記憶體	通用資料記憶體	
Bank 位置	容量	Bank : 地址
0 · 1	64×8	Bank 0: 40H~7FH

資料記憶體概要



資料記憶體結構

通用資料記憶體

通用資料記憶體共 64 位元組位於 Bank 0。所有的單片機程式需要一個讀 / 寫的存儲區，讓臨時資料可以被儲存和再使用，該 RAM 區域就是通用資料記憶體。這個資料存儲區可讓使用者進行讀取和寫入的操作。使用位元元操作指令可對個別的位元做置位或復位的操作，極大地方便了用戶在資料記憶體內進行位操作。

特殊功能資料記憶體

這個區域的資料記憶體是存放特殊寄存器的，這些寄存器與單片機的正确操作密切相關，大多數的寄存器可進行讀取和寫入，只有一些是被防寫而只能讀取的，相關細節的介紹請參看有關特殊功能寄存器的部分。要注意的是，任何讀取指令對記憶體中未定義的位址進行讀取將返回“00H”。

Bank 0		:	Bank 1	Bank 0		:	Bank 1
00H	IAR0			20H	PAS1		
01H	MP0			21H	PBS0		
02H	IAR1			22H	INTC0		
03H	MP1			23H	INTC1		
04H	BP			24H	INTC2		
05H	ACC			25H	PB		
06H	PCL			26H	PBC		
07H	TBLP			27H	PBPU		
08H	TBLH			28H	STMC0		
09H	MUXSEL			29H	STMC1		
0AH	STATUS			2AH	STMDL		
0BH	SCC			2BH	STMDH		
0CH	HIRCC			2CH	STMAL		
0DH	PSCR			2DH	STMAH		
0EH	TB0C			2EH	OPDSWA		
0FH	RSTFC			2FH	OPDSWB		
10H	TB1C			30H	OPDSWC		
11H	LVDC			31H	OPDSWD		
12H	WDTC			32H	OPDC0		
13H	INTEG			33H	OPDC1		
14H	PA			34H	OPDDA		
15H	PAC			35H	OPDA0CAL		
16H	PAPU			36H	OPDA1CAL		
17H	PAWU			37H	OPDCCAL		
18H	IDATA			38H	SADOL		
19H	SIMC0			39H	SADOH		
1AH	SIMC1/UUCR1			3AH	SADC0		
1BH	SIMC2/SIMA/UUCR2			3BH	SADC1		
1CH	SIMD/UTXR_RXR			3CH	SADC2		
1DH	SIMTOC/UBRG			3DH	IFS		
1EH	UUSR			3EH	EEA		
1FH	PAS0			3FH	EED		
				40H		EEC	

□ : Unused, read as 00H

特殊功能資料記憶體結構

特殊功能寄存器

大部分特殊功能寄存器的細節將在相關功能章節描述，但有幾個寄存器需在此章節單獨描述。

間接定址寄存器 – IAR0, IAR1

間接定址寄存器 IAR0 和 IAR1 的位址雖位於資料存儲區寄存器區域，但不同于普通寄存器，它們沒有實際的物理位元址。與定義實際記憶體位址的直接記憶體定址不同，間接定址是使用間接定址寄存器和記憶體指標來執行記憶體資料操作。在間接定址寄存器 IAR0 和 IAR1 上的任何動作，將對間接定址指針 MP0 和 MP1 所指定的記憶體位址產生對應的讀 / 寫操作。它們總是成對出現，IAR0 和 MP0 可以訪問 Bank0，而 IAR1 和 MP1 可以訪問任何 Bank。因為這些間接定址寄存器不是實際存在的，直接讀取將返回“00H”的結果，而直接寫入此寄存器則不做任何操作。

記憶體指針 – MP0, MP1

該單片機提供兩個記憶體指針，即 MP0 和 MP1。由於這些指標在資料記憶體中能像普通的寄存器一樣被操作，因此提供了一個定址和資料追蹤的有效方法。當對間接定址寄存器進行任何操作時，單片機指向的實際位址是由間接定址指標所指定的位址。MP0、IAR0 用於訪問 Bank 0，而 MP1 和 IAR1 可通過 BP 寄存器訪問所有的 Bank。直接定址僅可以用在 Bank 0 中，其它所有 Bank 只能使用 MP1 和 IAR1 進行間接定址。注意，對於該單片機，記憶體指針的第 7 位未被用於定址記憶體空間。當讀取 MP0 或 MP1 的第 7 位時，將返回“1”的值。

以下例子說明如何清除一個具有 4 RAM 位址的區塊，它們已事先定義成地址 adres1 到 adres4。

間接定址程式舉例

```
data .section 'data'
adres1 db ?
adres2 db ?
adres3 db ?
adres4 db ?
block db ?
code .section at 0 'code'
org 00h
start:
    mov a, 04h          ; setup size of block
    mov block, a
    mov a, offset adres1 ; Accumulator loaded with first RAM address
    mov mp0, a         ; setup memory pointer with first RAM address
loop:
    clr IAR0           ; clear the data at address defined by MP0
    inc mp0            ; increase memory pointer
    sdz block          ; check if last memory location has been cleared
    jmp loop
continue:
:
```

在上面的例子中有一點值得注意，即並沒有確定 RAM 位址。

存儲區指標 – BP

資料記憶體被分為兩個 Bank，即 Bank 0 和 Bank 1。可以通過設置存儲區指標(Bank Pointer) 值來訪問不同的資料存儲區。BP 指標的 bit 0 用於選擇資料存儲區的 Bank 0 或 Bank 1。

復位後，資料記憶體會初始化到 Bank 0，但是在休眠和空間模式下的 WDT 溢出重定，不會改變通用資料記憶體的存儲區號。應該注意的是特殊功能資料記憶體不受存儲區的影響，也就是說，不論是在哪一個存儲區，都能對特殊功能寄存器進行讀寫操作。資料記憶體的直接定址總是訪問 Bank 0，不影響存儲區指標的值。要訪問 Bank 1 之外的存儲區，則必須要使用間接定址方式。

● BP 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	—	DMBP0
R/W	—	—	—	—	—	—	—	R/W
POR	—	—	—	—	—	—	—	0

Bit 7~1 未定義，讀為 “0”
 Bit 0 DMBP0：資料存儲區選擇位元
 0：Bank 0 1：Bank 1

累加器 – ACC

對任何單片機來說，累加器是相當重要的，且與 ALU 所完成的運算有密切關係，所有 ALU 得到的運算結果都會暫時存在 ACC 累加器裡。若沒有累加器，ALU 必須在每次進行如加法、減法和移位元的運算和邏輯操作時，將結果寫入到資料記憶體，這樣會造成程式編寫和時間的負擔。另外資料傳送也常常牽涉到累加器的臨時儲存功能，例如在使用者定義的一個寄存器和另一個寄存器之間傳送資料時，由於兩寄存器之間不能直接傳送資料，因此必須通過累加器來傳送資料。

程式計數器低位元組寄存器 – PCL

為了提供額外的程式控制功能，程式計數器低位元組規劃在資料記憶體的特殊功能區域內，通過對此寄存器操作，便可直接跳轉到其他程式位址。直接給 PCL 寄存器賦值將導致程式直接跳轉到程式記憶體的某一位址，然而由於寄存器只有 8 位長度，因此只允許在本頁的程式記憶體範圍內進行跳轉，而當執行此操作時，要注意會插入一個空運算速度。

表格寄存器 – TBLP, TBLH

這兩個特殊功能寄存器對存儲在程式記憶體中的表格進行操作。TBLP 為表格指標，指向表格資料存儲的位址。它的值必須在任何表格讀取指令執行前加以設定，由於此值可以被如 “INC” 或 “DEC” 的指令所改變，這就提供了一種簡單的方法對表格資料進行讀取。表格讀取資料指令執行之後，表格資料高位元組存儲在 TBLH 中。其中要注意的是，表格資料低位元組會被傳送到使用者指定的位址。

狀態寄存器 – STATUS

這 8 位元的狀態寄存器由零標誌位元 (Z)、進位元標誌位元 (C)、輔助進位元標誌位元 (AC)、溢出標誌位元 (OV)、暫停標誌位元 (PDF) 和看門狗計時器溢出標誌位元 (TO) 組成。這些算術 / 邏輯操作和系統運行標誌位元是用來記錄單片機的運行狀態。

除了 PDF 和 TO 標誌外，狀態寄存器中的位元像其它大部分寄存器一樣可以被改變。任何資料寫入到狀態寄存器將不會改變 TO 或 PDF 標誌位元。另外，執行不同的指令後，與狀態寄存器有關的運算可能會得到不同的結果。TO 標誌位元只會受系統上電、看門狗溢出或執行 “CLR WDT” 或 “HALT” 指令影響。PDF 標誌位元只會受執行 “HALT” 或 “CLR WDT” 指令或系統上電影響。

Z、OV、AC 和 C 標誌位元通常反映最近運算的狀態。

- C：當加法運算的結果產生進位，或減法運算的結果沒有產生借位時，則 C 被置位，否則 C 被清零，同時 C 也會被帶進位元的移位元元指令所影響。
- AC：當低半位元組加法運算的結果產生進位元，或高半位元組減法運算的結果沒有產生借位時，AC 被置位，否則 AC 被清零。
- Z：當算術或邏輯運算結果是零時，Z 被置位，否則 Z 被清零。
- OV：當運算結果高兩位元的進位元狀態異或結果為 1 時，OV 被置位，否則 OV 被清零。
- PDF：系統上電或執行 “CLR WDT” 指令會清零 PDF，而執行 “HALT” 指令則會置位元元 PDF。
- TO：系統上電或執行 “CLR WDT” 或 “HALT” 指令會清零 TO，而當 WDT 溢出則會置位 TO。

另外，當進入一個中斷程式或執行副程式調用時，狀態寄存器不會自動壓入到堆疊保存。假如狀態寄存器的內容是重要的且副程式可能改變狀態寄存器的話，則需謹慎的去做正確的儲存。

● **STATUS 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	—	—	TO	PDF	OV	Z	AC	C
R/W	—	—	R	R	R/W	R/W	R/W	R/W
POR	—	—	0	0	x	x	x	x

“x”：未知

Bit 7~6 未定義，讀為 “0”

Bit 5 **TO**：看門狗溢出標誌位元

0：系統上電或執行 “CLR WDT” 或 “HALT” 指令後

1：看門狗溢出發生

Bit 4 **PDF**：暫停標誌位元

0：系統上電或執行 “CLR WDT” 指令後

1：執行 “HALT” 指令

Bit 3 **OV**：溢出標誌位元

0：無溢出

1：運算結果高兩位元的進位元狀態異或結果為 1

Bit 2 **Z**：零標誌位元

0：算術或邏輯運算結果不為 0

1：算術或邏輯運算結果為 0

Bit 1 **AC**：輔助進位元標誌位元

0：無輔助進位

1：在加法運算中低四位產生了向高四位進位，或減法運算中低四位不發生從高四位借位

Bit 0 **C**：進位元標誌位元

0：無進位

1：如果在加法運算中結果產生了進位，或在減法運算中結果不發生借位

C 標誌位元也受迴圈移位元元指令的影響。

EEPROM 資料記憶體

該單片機的一個特性是內建EEPROM 資料記憶體。由於其非易失的存儲結構，即使在電源掉電的情況下記憶體內的資料仍然保存完好。這種存儲區擴展了記憶體空間，對設計者來說增加了許多新的應用機會。EEPROM 可以用來存儲產品編號、校準值、使用者特定資料、系統組態參數或其它產品資訊等。EEPROM 的資料讀取和寫入過程也會變的更簡單。

EEPROM 資料記憶體結構

該單片機的 EEPROM 資料記憶體容量為 32×8 位。由於映射方式與程式記憶體和資料記憶體不同，因此不能像其它類型的記憶體一樣定址。使用 Bank 0 中的一個位址寄存器和一個資料寄存器以及 Bank 1 中的一個控制寄存器，可以實現對 EEPROM 的單字節讀寫操作。

EEPROM 寄存器

有三個寄存器控制內部 EEPROM 資料記憶體總的操作。位址寄存器 EEA、資料寄存器 EED 及控制寄存器 EEC。EEA 和 EED 位於 Bank 0 中，它們能像其它特殊功能寄存器一樣直接被訪問。EEC 位於 Bank 1 中，不能被直接訪問，僅能通過 MP1 和 IAR1 進行間接讀取或寫入。由於 EEC 控制寄存器位於 Bank 1 中的 “40H”，在 EEC 寄存器上的任何操作被執行前，MP1 必須先設為 “40H”，BP 被設為 “01H”。

寄存器名稱	位							
	7	6	5	4	3	2	1	0
EEA	—	—	—	EEA4	EEA3	EEA2	EEA1	EEA0
EED	D7	D6	D5	D4	D3	D2	D1	D0
EEC	—	—	—	—	WREN	WR	RDEN	RD

EEPROM 寄存器列表

● **EEA 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	EEA4	EEA3	EEA2	EEA1	EEA0
R/W	—	—	—	R/W	R/W	R/W	R/W	R/W
POR	—	—	—	0	0	0	0	0

Bit 7~5 未定義，讀為“0”

Bit 4~0 EEA4~EEA0：數據 EEPROM 地址 bit 4 ~ bit 0

● **EED 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 D7~D0：EEPROM 數據 bit 7 ~ bit 0

● **EEC 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	WREN	WR	RDEN	RD
R/W	—	—	—	—	R/W	R/W	R/W	R/W
POR	—	—	—	—	0	0	0	0

Bit 7~4 未定義，讀為“0”

Bit 3 **WREN**：數據 EEPROM 寫使能位元

0：除能

1：使能

此位元為資料 EEPROM 寫使能位元，向資料 EEPROM 寫操作之前需將此位置高。將此位元清零時，則禁止向資料 EEPROM 寫操作。

Bit 2 **WR**：EEPROM 寫控制位

0：寫週期結束

1：寫週期有效

此位元為資料 EEPROM 寫控制位元，由應用程式將此位置高將啟動寫週期。寫週期結束後，硬體自動將此位元清零。當 WREN 未先置高時，此位置高無效。

Bit 1 **RDEN**：數據 EEPROM 讀使能位元

0：除能

1：使能

此位元為資料 EEPROM 讀使能位元，向資料 EEPROM 讀操作之前需將此位置高。將此位元清零時，則禁止向資料 EEPROM 讀操作。

Bit 0 **RD**：EEPROM 讀控制位

0：讀週期結束

1：讀週期有效

此位元為資料 EEPROM 讀控制位元，由應用程式將此位置高將啟動讀週期。讀週期結束後，硬體自動將此位元清零。當 RDEN 未先置高時，此位置高無效。

注：1. 在同一條指令中 WREN、WR、RDEN 和 RD 不能同時置為“1”。WR 和 RD 不能同時置為“1”。

2. 確保 fSUB 時鐘在執行寫動作前已穩定。

3. 確保寫動作完成後才可改寫 EEC 寄存器。

從 EEPROM 中讀取數據

從EEPROM 中讀取資料，EEPROM 中讀取資料的位址要先放入EEA 寄存器中。

EEC 寄存器中的讀使能位 RDEN 置為高以使能讀功能。若 EEC 寄存器中的 RD 位被置高，一個讀週期將開始。若 RD 位已置為高而 RDEN 位還未被設置則不能開始讀操作。若讀週期結束，RD 位將自動清除為“0”，資料可以從EED 寄存器中讀取。資料在其它讀或寫操作執行前將一直保留在 EED 寄存器中。應用程式將輪詢 RD 位元以確定資料可以有效地被讀取。

寫數據到 EEPROM

寫資料至 EEPROM，EEPROM 中寫入資料的位址要先放入 EEA 寄存器中，然後將要寫入的資料放入 EED 寄存器中。EEC 寄存器中的寫使能位 WREN 先置為高以寫使能功能，然後 EEC 寄存器中的 WR 位需立即置為高以開始寫操作。這兩條指令必須在兩個運算速度內連續執行。在寫操作執行之前需將總中斷控制位 EMI 清零，在寫週期開始後再置高。若 WR 位已置為高而 WREN 位還未被置高則不能開始寫操作。由於控制 EEPROM 寫週期是一個內部時鐘，與單片機的系統時鐘非同步，所以資料寫入 EEPROM 的時間將有所延遲。可通過輪詢 EEC 寄存器中的 WR 位或判斷 EEPROM 中斷以偵測寫週期是否完成。若寫週期完成，WR 位將自動清除為“0”，通知使用者資料已寫入 EEPROM。因此，應用程式將輪詢 WR 位元以確定寫週期是否結束。

防寫

防止誤寫入的防寫有以下幾種。單片機上電後控制寄存器中的寫使能位將被清除以杜絕任何寫入操作。上電後 BP 將重置為“0”，這意味著資料存儲區 Bank 0 被選中。由於 EEPROM 控制寄存器位於 Bank 1 中，這增加了對寫操作的保護措施。在正常程式操作中確保控制寄存器中的寫使能位被清除將能防止不正確的寫操作。

EEPROM 中斷

EEPROM 寫週期結束後將產生 EEPROM 寫中斷，需先通過設置相關中斷寄存器的 DEE 位使能 EEPROM 中斷。當 EEPROM 寫週期結束，DEF 請求標誌位元將被置位元。若總中斷、EEPROM 中斷使能且堆疊未滿的情況下將跳轉到相應的 EEPROM 中斷向量中執行。當中斷被回應，EEPROM 中斷標誌位元 DEF 將自動重置且 EMI 位會被清零以除能其它中斷。更多細節將在中斷章節講述。

程式設計注意事項

必須注意的是資料不會無意寫入 EEPROM。在沒有寫動作時寫使能位被正常清零可以增強保護功能。BP 指標也可以正常清零以阻止進入 EEPROM 控制寄存器存在的 Bank 1。儘管沒有必要，寫一個簡單的讀回程式以檢查新寫入的資料是否正確還是應該考慮的。

當 WREN 位元被置為高以後，寫資料位元 WR 必須立刻置為高，以確保寫迴圈正確執行。總中斷位 EMI 在寫迴圈開始前應當被清零，寫迴圈開始後再將其使能。注意，單片機不應在 EEPROM 讀或寫操作完全完成之前進入空間或休眠模式，否則 EEPROM 讀或寫操作將失敗。

程式舉例

從 EEPROM 中讀取數據 – 輪詢法

```
MOV A, EEPROM_ADRES      ; user defined address
MOV EEA, A
MOV A, 040H              ; setup memory pointer MPl
MOV MPl, A               ; MPl points to EEC register
MOV A, 01H              ; setup Bank Pointer
MOV BP, A
SET IAR1.1              ; set RDEN bit, enable read operations
SET IAR1.0              ; start Read Cycle - set RD bit
BACK:
SZ IAR1.0               ; check for read cycle end
JMP BACK
CLR IAR1                 ; disable EEPROM read if no more read operations
                          ; are required

CLR BP
MOV A, EED               ; move read data to register
MOV READ_DATA, A
```

注：對於每一個讀操作，即使位址是連續的，都必須重新設置位址寄存器，接著再將 RD 位置高開啟一個讀週期。

寫數據到 EEPROM – 輪詢法

```
MOV A, EEPROM_ADRES      ; user defined address
MOV EEA, A
MOV A, EEPROM_DATA      ; user defined data
MOV EED, A
MOV A, 040H              ; setup memory pointer MPl
MOV MPl, A               ; MPl points to EEC register
MOV A, 01H              ; setup Bank Pointer
MOV BP, A
CLR EMI
SET IAR1.3              ; set WREN bit, enable write operations
SET IAR1.2              ; start Write Cycle - set WR bit - executed
                          ; immediately after set WREN bit

SET EMI
BACK:
SZ IAR1.2               ; check for write cycle end
JMP BACK
CLR BP
```

振盪器

不同的振盪器選擇可以讓使用者在不同的應用需求中實現更大範圍的功能。振盪器的靈活性使得在速度和功耗方面可以達到最優化。振盪器選擇和操作是通過相關控制寄存器完成的。

振盪器概述

振盪器除了作為系統時鐘源，還作為看門狗計時器和時基中斷的時鐘源。集成的兩個內部振盪器不需要任何週邊器件。它們提供的高速和低速系統振盪器具有較寬的頻率範圍。所有振盪器選擇通過配置選項選擇。較高頻率的振盪器提供更高的性能，但要求有更高的功率，反之亦然。動態切換快慢系統時鐘的能力使單片機具有靈活而優化的性能 / 功耗比，此特性對功耗敏感的應用領域尤為重要。

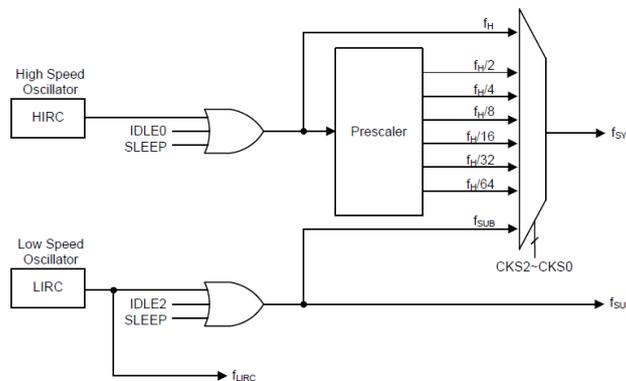
類型	名稱	頻率
內部高速 RC	HIRC	8MHz
內部低速 RC	LIRC	32kHz

振盪器類型

系統時鐘配置

該單片機有兩個系統振盪器，包括一個高速振盪器和一個低速振盪器。高速振盪器是內部 8MHz RC 振盪器 HIRC，低速振盪器是內部 32kHz RC 振盪器 LIRC。使用高速或低速振盪器作為系統時鐘的選擇是通過設置 SCC 寄存器中的 CKS2~CKS0 位元決定的，系統時鐘可動態選擇。

低速或高速系統時鐘頻率由 SCC 寄存器的 CKS2~CKS0 位決定的。



系統時鐘配置

內部高速 RC 振盪器 – HIRC

內部 RC 振盪器是一個集成的系統振盪器，不需其它外部器件。內部 RC 振盪器具有固定的頻率 8MHz。晶片在製造時進行調整且內部含有頻率補償電路，使得振盪頻率因電源電壓、溫度以及晶片製成工藝不同的影響減至最低程度。

內部 32kHz 振盪器 – LIRC

內部 32kHz 系統振盪器是一個低頻振盪器。它是一個完全集成 RC 振盪器，運行時典型頻率值為 32kHz，且無需外部元件。晶片在製造時進行調整且內部含有頻率補償電路，使得振盪器因電源電壓、溫度及晶片製成工藝不同的影響減至最低。

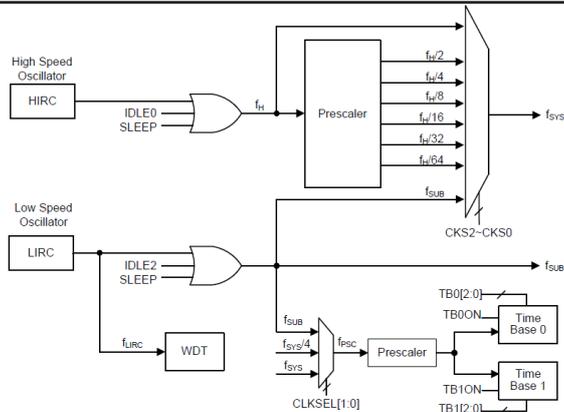
工作模式和系統時鐘

現今的應用要求單片機具有較高的性能及盡可能低的功耗，這種矛盾的要求在可攜式電池供電的應用領域尤為明顯。高性能所需要的高速時鐘將增加功耗，反之亦然。該單片機提供高、低速兩種時鐘源，它們之間可以動態切換，使用者可通過優化單片機操作來獲得最佳性能 / 功耗比。

系統時鐘

該單片機為 CPU 和週邊功能操作提供了多種不同的時鐘源。使用者使用配置選項和寄存器程式設計可獲取多種時鐘，進而使系統時鐘獲取最大的應用性能。

主系統時鐘可來自高頻時鐘源 f_H 或低頻時鐘源 f_{SUB} ，通過 SCC 寄存器中的 CKS2~CKS0 位進行選擇。高頻時鐘來自 HIRC 振盪器。低頻系統時鐘源來自 LIRC 振盪器。其它系統時鐘還有高速系統振盪器的分頻 $f_H/2 \sim f_H/64$ 。



注：當系統時鐘源 f_{SYS} 由 f_H 到 f_{SUB} 轉換時，高速振盪器將停止以節省耗電。因此，沒有為週邊電路提供 $f_H \sim f_H/64$ 的頻率。

系統工作模式

單片機有 6 種不同的工作模式，每種有它自身的特性，根據應用中不同的性能和功耗要求可選擇不同的工作模式。單片機正常工作有兩種模式：快速模式和低速模式。剩餘的 4 種工作模式：休眠模式、空閒模式 0、空閒模式 1 和空閒模式 2 用於單片機 CPU 關閉時以節省耗電。

工作模式	CPU	寄存器設置			fSYS	fH	fSUB	fLIRC
		FHIDEN	FSIDEN	CKS2~CKS0				
快速模式	On	x	x	000~110	$f_H \sim f_H/64$	On	On	On
低速模式	On	x	x	111	fSUB	On/Off ⁽¹⁾	On	On
空閒模式 0	Off	0	1	000~110	Off	Off	On	On
				111	On			
空閒模式 1	Off	1	1	xxx	On	On	On	On
空閒模式 2	Off	1	0	000~110	On	On	Off	On
				111	Off			
休眠模式	Off	0	0	xxx	Off	Off	Off	On/Off ⁽²⁾

“x”：無關

注：1. 在低速模式中， f_H 開啟或關閉由相應的振盪器使能位控制。

2. 在休眠模式中， f_{LIRC} 開啟或關閉由 WDT 功能使能或除能控制。

快速模式

這是主要的工作模式之一，單片機的所有功能均可在此模式中實現且系統時鐘由高速振盪器提供。該模式下單片機正常工作的時鐘源來自 HIRC 振盪器。

高速振盪器頻率可被分為 1~64 的不等比率，實際的比率由 SCC 寄存器中的 CKS2~CKS0 位選擇的。單片機使用高速振盪器分頻作為系統時鐘可減少工作電流。

低速模式

此模式的系統時鐘雖為較低速時鐘源，但單片機仍能正常工作。該低速時鐘源來自 fSUB，fSUB 時鐘來源於 LIRC 振盪器。

休眠模式

在 HALT 指令執行後且 FHIDEN 和 FSIDEN 位元為低時，系統進入休眠模式。在休眠模式中，CPU 停止運行，fSUB 停止為週邊功能提供時鐘。若看門狗計時器功能使能，fLIRC 繼續運行。

空閒模式 0

執行 HALT 指令後且 SCC 寄存器中 FHIDEN 位為低、FSIDEN 位元為高時，系統進入空閒模式 0。在空閒模式 0 中，CPU 停止，但低速振盪器會開啟以驅動一些週邊功能。

空閒模式 1

執行 HALT 指令後且 SCC 寄存器中 FHIDEN 和 FSIDEN 位元為高時，系統進入空閒模式 1。在空閒模式 1 中，CPU 停止，但高速和低速振盪器都會開啟以提供一個時鐘源保持一些週邊功能繼續工作。

空閒模式 2

執行 HALT 指令後且 SCC 寄存器中 FHIDEN 位為高、FSIDEN 位元為低時，系統進入空閒模式 2。在空閒模式 2 中，CPU 停止，但高速振盪器會開啟以提供一個時鐘源保持一些週邊功能繼續工作。

控制寄存器

寄存器 SCC 和 HIRCC 用於控制單片機系統時鐘和相應的振盪器配置。

寄存器名稱	位							
	7	6	5	4	3	2	1	0
SCC	CKS2	CKS1	CKS0	—	—	—	FHIDEN	FSIDEN
HIRCC	—	—	—	—	—	—	HIRCF	HIRCEN

系統工作模式控制寄存器清單

● SCC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	CKS2	CKS1	CKS0	—	—	—	FHIDEN	FSIDEN
R/W	R/W	R/W	R/W	—	—	—	R/W	R/W
POR	0	0	0	—	—	—	0	0

Bit 7~5 CKS2~CKS0：系統時鐘選擇位元

- 000：fH
- 001：fH/2
- 010：fH/4
- 011：fH/8
- 100：fH/16
- 101：fH/32
- 110：fH/64
- 111：fSUB

這三位元用於選擇系統時鐘源。除了 fH 或 fSUB 提供的系統時鐘源外，也可使用高頻振盪器的分頻作為系統時鐘。

Bit 4~2 未定義，讀為“0”

Bit 1 FHIDEN：CPU 關閉時高頻振盪器控制位
0：除能 1：使能

此位元元用來控制在執行 HALT 指令 CPU 關閉後高速振盪器是否停止。

Bit 0 FSIDEN：CPU 關閉時低頻振盪器控制位
0：除能 1：使能

此位元元用來控制在執行 HALT 指令 CPU 關閉後低速振盪器是否停止。

● HIRCC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	HIRCF	HIRCEN
R/W	—	—	—	—	—	—	R	R/W
POR	—	—	—	—	—	—	0	1

Bit 7~2 未定義，讀為“0”

Bit 1 HIRCF：HIRC 振盪器穩定標誌位元
0：HIRC 不穩定
1：HIRC 穩定

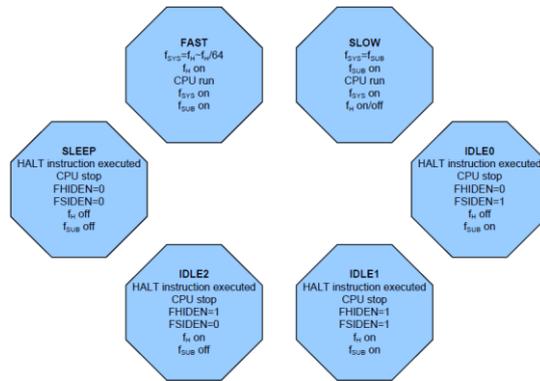
此位用於表明 HIRC 振盪器是否穩定。HIRCEN 位置高使能 HIRC 振盪器，HIRCF 位會先被清零，在 HIRC 穩定後會被置高。

Bit 0 HIRCEN：HIRC 振盪器使能控制位
0：除能
1：使能

工作模式切換

單片機可在各個工作模式間自由切換，使得使用者可根據所需選擇最佳的性能 / 功耗比。用此方式，對單片機工作的性能要求不高的情況下，可使用較低頻時鐘以減少工作電流，在可攜式應用上延長電池的使用壽命。

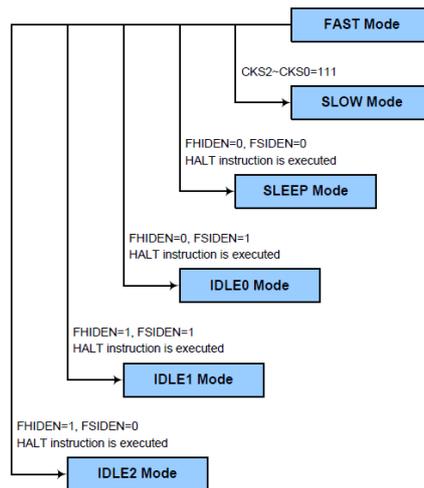
簡單來說，快速模式和低速模式間的切換僅需設置 SCC 寄存器中的 CKS2~CKS0 位即可實現，而快速模式 / 低速模式與休眠模式 / 空閒模式間的切換經由 HALT 指令實現。當 HALT 指令執行後，單片機是否進入空閒模式或休眠模式由 SCC 寄存器中的 FHIDEN 位和 FSIDEN 位決定的。



快速模式切換到低速模式

系統運行在快速模式時使用高速系統振盪器，因此較為耗電。可通過設置 SCC 寄存器中的 CKS2~CKS0 位為 “111” 使系統時鐘切換至運行在低速模式下。此時將使用低速系統振盪器以節省耗電。使用者可在對性能要求不高的操作中使用此方法以減少耗電。

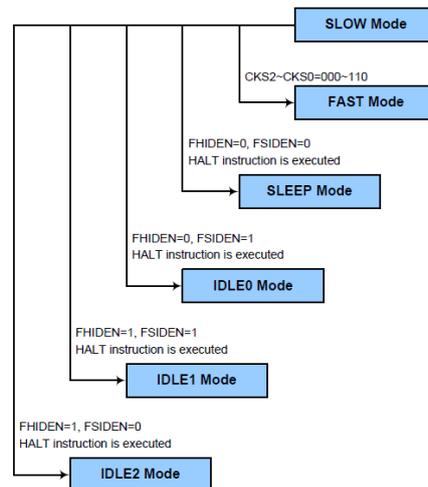
低速模式的時鐘源來自 LIRC 振盪器。因此要求這個振盪器在所有模式切換動作發生前穩定下來。



低速模式切換到快速模式

在低速模式時系統時鐘來自 fSUB。切換回快速模式時，需設置 CKS2~CKS0 位為 “000” ~ “110” 使系統時鐘從 fSUB 切換到 $f_H \sim f_H/64$ 。

然而，如果在低速模式下 f_H 因未使用而關閉，那麼從低速模式切換到快速模式時，它需要一定的時間來重新起振和穩定，可通過檢測 HIRCC 寄存器中的 HIRCF 位元進行判斷，所需的高速系統振盪器穩定時間指定在系統上電時間電氣特性中。



進入休眠模式

進入休眠模式的方法僅有一種即應用程式中執行 “HALT” 指令前需設置 SCC 寄存器中 FHIDEN 和 FSIDEN 位都為 “0” 。在上述條件下執行該指令後，將發生的情況如下：

- 系統時鐘停止運行，應用程式停止在 “HALT” 指令處。
- 資料記憶體中的內容和寄存器將保持當前值。
- 輸入 / 輸出將保持當前值。
- 狀態寄存器中暫停標誌 PDF 將被置起，看門狗溢出標誌 TO 將被清除。
- 若 WDT 功能使能，WDT 將被清零並重新開始計數；若 WDT 功能除能，WDT 將被清零並停止。

進入空間模式 0

進入空間模式 0 的方法僅有一種即應用程式中執行 “HALT” 指令前需設置 SCC 寄存器中 FHIDEN 位為 “0” 且 FSIDEN 位為 “1” 。在上述條件下執行該指令後，將發生的情況如下：

- 系統時鐘停止運行，應用程式停止在 “HALT” 指令處，時鐘 fSUB 將繼續運行。
- 資料記憶體中的內容和寄存器將保持當前值。

輸入 / 輸出將保持當前值。

- 狀態寄存器中暫停標誌 PDF 將被置起，看門狗溢出標誌 TO 將被清除。
- 若 WDT 功能使能，WDT 將被清零並重新開始計數；若 WDT 功能除能，

WDT 將被清零並停止。

進入空間模式 1

進入空間模式 1 的方法僅有一種即應用程式中執行 “HALT” 指令前需設置 SCC 寄存器中 FHIDEN 和 FSIDEN 位都為 “1” 。在上述條件下執行該指令後，將發生的情況如下：

- fH 和 fSUB 時鐘開啟，應用程式停止在 “HALT” 指令處。
- 資料記憶體中的內容和寄存器將保持當前值。
- 輸入 / 輸出將保持當前值。
- 狀態寄存器中暫停標誌 PDF 將被置起，看門狗溢出標誌 TO 將被清除。
- 若 WDT 功能使能，WDT 將被清零並重新開始計數；若 WDT 功能除能，

WDT 將被清零並停止。

進入空間模式 2

進入空間模式 2 的方法僅有一種即應用程式中執行 “HALT” 指令前需設置 SCC 寄存器中 FHIDEN 位為 “1” 且 FSIDEN 位為 “0” 。在上述條件下執行該指令後，將發生的情況如下：

- fH 時鐘開啟，fSUB 時鐘關閉，應用程式停止在 “HALT” 指令處。
- 資料記憶體中的內容和寄存器將保持當前值。
- 輸入 / 輸出將保持當前值。
- 狀態寄存器中暫停標誌 PDF 將被置起，看門狗溢出標誌 TO 將被清除。
- 若 WDT 功能使能，WDT 將被清零並重新開始計數；若 WDT 功能除能，

WDT 將被清零並停止。

待機電流注意事項

由於單片機進入休眠或空間模式的主要原因是將 MCU 的電流降低到盡可能低，可能到只有幾個微安的級別（空間模式 1 和空間模式 2 除外），所以如果要將電路的電流降到最低，電路設計者還應有其它的考慮。應該特別注意的是單片機的輸入 / 輸出引腳。所有高阻抗輸入腳都必須連接到固定的高或低電平，因為引腳浮空會造成內部振盪並導致耗電增加。這也應用於有不同封裝的單片機，因為它們可能含有未引出的引腳，這些引腳也必須設為輸出或帶上有上拉電阻的輸入。

另外還需注意單片機設為輸出的 I/O 引腳上的負載。應將它們設置在有最小拉電流的狀態或將它們和其它的 CMOS 輸入一樣接到沒有拉電流的外部電路上。還應注意的是，如果選擇 LIRC 振盪器，會導致耗電增加。

在空間模式 1 和空間模式 2 中，高速振盪器開啟。若週邊功能時鐘來自高速振盪器，額外的待機電流也可能會有幾百微安。

喚醒

單片機進入休眠模式或空閒模式後，系統時鐘將停止以降低功耗。然而單片機再次喚醒，原來的系統振盪器重新起振、穩定且恢復正常工作需要一定的時間。系統進入休眠或空閒模式之後，可以通過以下幾種方式喚醒：

- PA 口下降沿
- 系統中斷
- WDT 溢出

單片機執行 HALT 指令，系統將進入空閒或休眠模式，PDF 將被置 1；系統上電或執行清除看門狗的指令，PDF 將被清零。若由 WDT 溢出喚醒，則會發生看門狗計時器復位。看門狗計數器溢出將會置位元 TO 標誌並喚醒系統，這種重定會重置程式計數器和堆疊指標，其它標誌保持原有狀態。

PA 口中的每個引腳都可以通過 PAWU 寄存器使能下降沿喚醒功能。PA 埠喚醒後，程式將在 “HALT” 指令後繼續執行。如果系統是通過中斷喚醒，則有兩種可能發生。第一種情況是：相關中斷除能或是中斷使能且堆疊已滿，則程式會在 “HALT” 指令之後繼續執行。這種情況下，喚醒系統的中斷會等到相關中斷使能或有堆疊層可以使用之後才執行。第二種情況是：相關中斷使能且堆疊未滿，則中斷可以馬上執行。如果在進入休眠或空閒模式之前中斷標誌位元已經被設置為 “1”，則相關中斷的喚醒功能將無效。

看門狗計時器

看門狗計時器的功能在於防止如電磁的幹擾等外部不可控制事件，所造成的程式不正常動作或跳轉到未知的地址。

看門狗計時器時鐘源

WDT 計時器時鐘源來自於內部時鐘 fLIRC，由 LIRC 振盪器提供。內部振盪器 LIRC 的頻率大約為 32kHz，這個特殊的內部時鐘週期隨 VDD、溫度和製成的不同而變化。看門狗計時器的時鐘源可分為 $2^8 \sim 2^{15}$ 以提供更大的溢出週期，分頻比由 WDTC 寄存器中的 WS2~WS0 位來決定。

看門狗計時器控制寄存器

WDTC 寄存器用於控制 WDT 功能的使能 / 除能及選擇溢出週期。WRF 軟體重定標誌位元將在 RSTFC 寄存器中顯示。這個寄存器與看門狗計時器的所有操作相關。

- WDTC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	WE4	WE3	WE2	WE1	WE0	WS2	WS1	WS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	1	0	1	0	0	1	1

Bit 7~3 **WE4~WE0** : WDT 功能 軟件 控制 位

10101 : 除能

01010 : 使能

其它值：MCU 復位若因外部環境雜訊使這些位發生改變，單片機將復位。復位動作發生在 tSRESET 延遲時間後，且 RSTFC 寄存器的 WRF 位將置為 1。

Bit 2~0

WS2~WS0 : WDT 溢出週期選擇位

000 : $2^8/fLIRC$

001 : $2^9/fLIRC$

010 : $2^{10}/fLIRC$

011 : $2^{11}/fLIRC$

100 : $2^{12}/fLIRC$

101 : $2^{13}/fLIRC$

110 : $2^{14}/fLIRC$

111 : $2^{15}/fLIRC$

這三位元控制 WDT 時鐘源的分頻比，從而實現對 WDT 溢出週期的控制。

● RSTFC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	LVRF	—	WRF
R/W	—	—	—	—	—	R/W	—	R/W
POR	—	—	—	—	—	x	—	0

x" : 未知

Bit 7~3 未定義，讀為 "0"

Bit 2 **LVRF** : LVR 功能重定標誌位元

見其它章節。

Bit 1 未定義，讀為 "0"

Bit 0 **WRF** : WDT 控制寄存器軟體重定標誌位元 0 : 未發生 1 : 發生

WDT 控制寄存器軟體重定時，該位元被置為 1，且通過應用程式清除。注意，該位元只能由應用程式清零。

看門狗計時器操作

當 WDT 溢出時，會產生一個晶片重定的動作。這也就意味著正常工作期間，使用者需在應用程式中看門狗溢出前有策略地看清門狗計時器以防止其產生復位，可使用清除看門狗指令實現。無論什麼原因，程式失常跳轉到一個未知的位址或進入一個閉環，此清除指令都不能被正確執行，此種情況下，看門狗將溢出以使單片機復位。WDTC 寄存器中的 WE4~WE0 位可提供使能 / 除能控制以及看門狗計時器復位操作。如果 WE4~WE0 設置為 "10101B"，則 WDT 除能；如果 WE4~WE0 設置為 "01010B"，則 WDT 使能。如果 WE4~WE0 設置為除 "01010B" 和 "10101B" 以外的其它任意值，則單片機在 tSRESET 延遲時間後復位。上電後這些位初始化為 "01010B"。

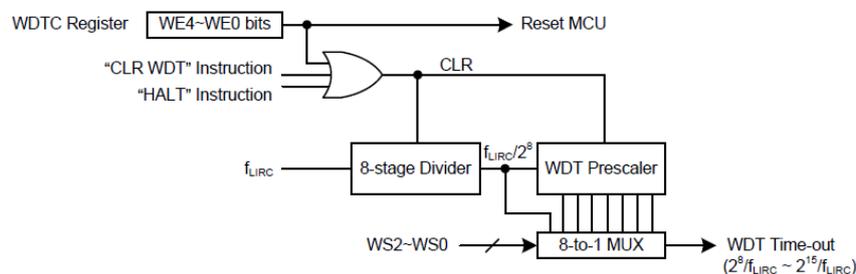
WE4~WE0 位	WDT 功能
10101B	除能
01010B	使能
其它值	MCU 復位

看門狗計時器使能 / 除能控制

程式正常運行時，WDT 溢出將導致晶片重定，並置位元狀態標誌位元 TO。若系統處於休眠或空閒模式，當 WDT 發生溢出時，狀態寄存器中的 TO 應置位，僅 PC 和堆疊指標復位。有三種方法可以用來清除 WDT 的內容。第一種是 WDTC 軟體重定，即將 WE4~WE0 位設置成除了 01010B 和 10101B 外的任意值；第二種是通過軟體清除指令，而第三種是通過 "HALT" 指令。

該單片機只使用一條清看門狗指令 "CLR WDT"。因此只要執行 "CLR WDT" 便清除 WDT。

當設置分頻比為 2¹⁵ 時，溢出週期最大。例如，時鐘源為 32kHz LIRC 振盪器，分頻比為 2¹⁵ 時最大溢出週期約 1s，分頻比為 28 時最小溢出週期約 8ms。



看門狗計時器

復位和初始化

重定功能是在任何單片機中基本的部分，使得單片機可以設定一些與外部參數無關的先置條件。最重要的復位條件是在單片機首次上電以後，經過短暫的延遲，內部硬體電路使得單片機處於預期的穩定狀態並開始執行第一條程式指令。上電重定以後，在程式執行之前，部分重要的內部寄存器將會被設定為預先設定的狀態。程式計數器就是其中之一，它會被清除為零，使得單片機從最低的程式記憶體位址開始執程式。

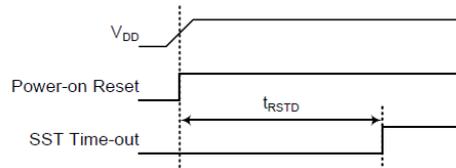
除上電復位以外，另一種重定為低電壓重定即 LVR 重定，在電源供應電壓低於 LVR 設定值時，系統會產生 LVR 重定。另一種復位為看門狗溢出單片機復位。不同方式的重定操作會對寄存器產生不同的影響。

復位功能

單片機有多種重定方式，描述如下：

上電復位

這是最基本且不可避免的復位，發生在單片機上電後。除了保證程式記憶體從開始位址執行，上電復位也使得其它寄存器被設定在預設條件。所有的輸入 / 輸出埠控制寄存器在上電復位時會保持高電平，以確保上電後所有引腳被設定為輸入狀態。

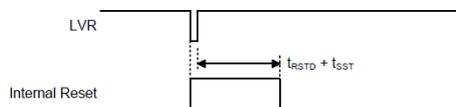


上電復位時序圖

低電壓重定 – LVR

單片機具有低電壓重定電路，用來監測它的電源電壓。當電源電壓值低於預設值，MCU 復位。

低電壓重定功能總是在快速模式或低速模式使能於特定的電壓值，VLVR，固定值為 2.1V。例如在更換電池的情況下，單片機供應的電壓可能會落在 0.9V~VLVR 之間，這時 LVR 將會自動復位單片機，並且 RSTFC 寄存器中的 LVRF 位將被自動置位為 1。LVR 包含以下的規格：有效的 LVR 信號，即在 0.9V~VLVR 的低電壓狀態的時間，必須超過 LVR/LVD 電氣特性中的 tLVR 參數的值。如果低電壓存在不超過 tLVR 參數的值，則 LVR 將會忽略它且不會執行復位功能。當單片機進入休眠或空閒模式時 LVR 功能將自動除能。



低電壓重定時序圖

● RSTFC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	LVRF	—	WRF
R/W	—	—	—	—	—	R/W	—	R/W
POR	—	—	—	—	—	x	—	0

“x”：未知

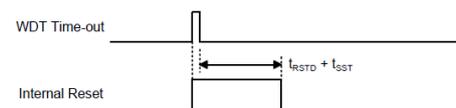
- Bit 7~3 未定義，讀為 “0”
- Bit 2 LVRF：LVR 功能重定標誌位元
- 0：未發生
- 1：發生

當特定的低電壓重定條件發生時，該位被置為 1。該位元只能由應用程式清零。

- Bit 1 未定義，讀為 “0”
- Bit 0 WRF：WDT 控制寄存器軟體重定標誌位元詳見其它章節。

正常運行時看門狗溢出復位

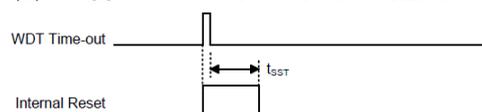
除了看門狗溢出標誌位元 TO 將被設為 “1” 之外，正常運行時看門狗溢出重定和硬體低壓重定相同。



正常運行時看門狗溢出復位時序圖

休眠或空閒時看門狗溢出復位

休眠或空閒時看門狗溢出復位和其它種類的復位有些不同。除了程式計數器與堆疊指標將被清 “0” 及 TO 位被設為 “1” 外，絕大部分的條件保持不變。圖中 tSST 的詳細說明請參考系統上電時間電氣特性。



休眠或空閒時看門狗溢出復位時序圖

重定初始狀態

不同的重定形式以不同的途徑影響重定標誌位元。這些標誌位元，即 PDF 和 TO 位元存放在狀態寄存器中，由休眠或空閒模式功能或看門狗計數器等幾種控制器操作控制。重定標誌位元如下所示：

TO	PDF	復位條件
0	0	上電復位
u	u	快速模式或低速模式時的 LVR 重定
1	u	快速模式或低速模式時的 WDT 溢出重定
1	1	空閒或休眠模式時的 WDT 溢出重定

“u”：代表不改變

在單片機上電重定之後，各功能單元初始化的情形，列於下表。

項目	重定後情況
程式計數器	復位為零
中斷	所有中斷被除能
看門狗計時器·時基	復位後清除·WDT 重新開始計數
計時器模組	所有計時器模組關閉
輸入 / 輸出	I/O 口設為輸入模式
堆疊指標	堆疊指標指向堆疊頂端

不同的復位形式對單片機內部寄存器的影響是不同的。為保證重定後程式能正常執行，瞭解寄存器在特定條件復位後的設置是非常重要的。下表即為不同方式重定後內部寄存器的狀況。注意的是，此單片機支持多種封裝類型，該表將反映較大封裝類型的情況。

寄存器	上電復位	LVR 復位(正常操作)	WDT 溢出(正常操作)	WDT 溢出(空閒 / 休眠)
IAR0	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu
MP0	1xxx xxxx	1uuu uuuu	1uuu uuuu	1uuu uuuu
IAR1	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu
MP1	1xxx xxxx	1uuu uuuu	1uuu uuuu	1uuu uuuu
BP	---- --0	---- --0	---- --0	---- --u
ACC	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu
PCL	0000 0000	0000 0000	0000 0000	0000 0000
TBLP	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBLH	-- xx xxxx	-- uu uuuu	-- uu uuuu	-- uu uuuu
MUXSEL	0 --- ----	0 --- ----	0 --- ----	u --- ----
STATUS	-- 00 xxxx	-- uu uuuu	-- 1 u uuuu	--11 uuuu
SCC	000- --00	000- --00	000- --00	uuu- uuuu
HIRCC	---- --01	---- --01	---- --01	---- --u u
PSCR	---- --00	---- --00	---- --00	---- --u u
TBOC	0 --- -000	0 --- -000	0 --- -000	u --- -uuu
RSTFC	---- -x-0	---- -1-u	---- -u-u	---- -u-u
TB1C	0 --- -000	0 --- -000	0 --- -000	u --- -uuu
LVDC	-- 00 0000	-- 00 0000	-- 00 0000	-- uu uuuu
WDTC	0101 0011	0101 0011	0101 0011	uuuu uuuu
INTEG	---- --00	---- --00	---- --00	---- --u u
PA	1111 1111	1111 1111	1111 1111	uuuu uuuu
PAC	1111 1111	1111 1111	1111 1111	uuuu uuuu
PAPU	0000 0000	0000 0000	0000 0000	uuuu uuuu
PAWU	0000 0000	0000 0000	0000 0000	uuuu uuuu
IDATA	0-00 0000	0-00 0000	0-00 0000	u-uu uuuu
SIMC0	1110 0000	1110 0000	1110 0000	uuuu uuuu
SIMC1(UMD=0)	1000 0001	1000 0001	1000 0001	uuuu uuuu
UUCR1* (UMD=1)	0000 00x0	0000 00x0	0000 00x0	uuuu uuuu
SIMC2/SIMA/ UUCR2	0000 0000	0000 0000	0000 0000	uuuu uuuu
SIMD/UTXR_RXR	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
SIMTOC (UUMD=0)	0000 0000	0000 0000	0000 0000	uuuu uuuu
UBRG* (UUMD=1)	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu

寄存器	上電復位	LVR 復位(正常操作)	WDT 溢出(正常操作)	WDT 溢出(空閒 / 休眠)
UBRG* (UUMD=1)	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
UUSR	0000 1011	0000 1011	0000 1011	uuuu uuuu
PAS0	0000 0000	0000 0000	0000 0000	uuuu uuuu
PAS1	0000 0000	0000 0000	0000 0000	uuuu uuuu
PBS0	-- 00 0000	-- 00 0000	-- 00 0000	-- uu uuuu
INTC0	-000 0000	-000 0000	-000 0000	-uuu uuuu
INTC1	0000 0000	0000 0000	0000 0000	uuuu uuuu
INTC2	- 000 -000	- 000 -000	- 000 -000	- uuu -uuu
PB	---- -1 1 1	---- -1 1 1	---- -1 1 1	---- -uuu
PBC	---- -1 1 1	---- -1 1 1	---- -1 1 1	---- -uuu
PBPU	---- -000	---- -000	---- -000	---- -uuu
STMC0	0000 0000	0000 0000	0000 0000	uuuu uuuu
STMC1	0000 0000	0000 0000	0000 0000	uuuu uuuu
STMDL	0000 0000	0000 0000	0000 0000	uuuu uuuu
STMDH	---- --00	---- --00	---- --00	---- --uu
STMAL	0000 0000	0000 0000	0000 0000	uuuu uuuu
STMAH	---- --00	---- --00	---- --00	---- --uu
OPDSWA	0000 0000	0000 0000	0000 0000	uuuu uuuu
OPDSWB	0000 0000	0000 0000	0000 0000	uuuu uuuu
OPDSWC	0000 0000	0000 0000	0000 0000	uuuu uuuu
OPDSWD	---0 0000	---0 0000	---0 0000	---u uuuu
OPDC0	0000 0000	0000 0000	0000 0000	uuuu uuuu
OPDC1	0000 0000	0000 0000	0000 0000	uuuu uuuu
OPDDA	0000 0000	0000 0000	0000 0000	uuuu uuuu
OPDA0CAL	0010 0000	0010 0000	0010 0000	uuuu uuuu
OPDA1CAL	0010 0000	0010 0000	0010 0000	uuuu uuuu
OPDCCAL	0001 0000	0001 0000	0001 0000	uuuu uuuu
SADOL	xxxx ----	xxxx ----	xxxx ----	uuuu ---- (ADRF=0)
				uuuu uuuu (ADRF=1)
SADOH	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu (ADRF=0)
				---- uuuu (ADRF=1)
SADC0	0000 0000	0000 0000	0000 0000	uuuu uuuu
SADC1	0000 -000	0000 -000	0000 -000	uuuu -uuu
SADC2	0 -- 0 0000	0 -- 0 0000	0 -- 0 0000	u-- u uuuu
IFS	-- 00 0000	-- 00 0000	-- 00 0000	-- uu uuuu
EEA	---0 0000	---0 0000	---0 0000	---u uuuu
EED	0000 0000	0000 0000	0000 0000	uuuu uuuu
EEC	---- 0000	---- 0000	---- 0000	---- uuuu

注：“u”表示不改變 “x”表示未知 “-”表示未定義

“*”：UUCR1 和 SIMC1 寄存器共用相同的記憶體位址，而 UBRG 和 SIMTOC 寄存器共用相同的記憶體位址。

UUCR1 或 UBRG 寄存器的預設值可以在重定後通過應用程式設置 UMD 位元為高時獲得。

輸入 / 輸出埠

Holtek 單片機的輸入 / 輸出口控制具有很大的靈活性。大部分引腳可在用戶程式控制下被設定為輸入或輸出。所有引腳的上拉電阻設置以及指定引腳的喚醒設置也都由軟體控制，這些特性也使得此單片機在廣泛應用上都能符合開發的需求。

該單片機提供 PA 和 PB 雙向輸入 / 輸出口。這些寄存器在資料記憶體有特定的位址。所有 I/O 口用於輸入輸出操作。作為輸入操作，輸入引腳無鎖存功能，也就是說輸入資料必須在執行 “MOV A, [m]”，T2 的上升沿準備好，m 為埠位址。對於輸出操作，所有資料都是被鎖存的，且保持不變直到輸出鎖存被重寫。

寄存器名稱	位							
	7	6	5	4	3	2	1	0
PA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
PAC	PAC7	PAC6	PAC5	PAC4	PAC3	PAC2	PAC1	PAC0
PAPU	PAPU7	PAPU6	PAPU5	PAPU4	PAPU3	PAPU2	PAPU1	PAPU0
PAWU	PAWU7	PAWU6	PAWU5	PAWU4	PAWU3	PAWU2	PAWU1	PAWU0
PB	—	—	—	—	—	PB2	PB1	PB0
PBC	—	—	—	—	—	PBC2	PBC1	PBC0
PBPU	—	—	—	—	—	PBPU2	PBPU1	PBPU0

“—”：未定義，讀為“0”

I/O 邏輯功能寄存器清單

上拉電阻

許多產品應用在埠處於輸入狀態時需要外加一個上拉電阻來實現上拉的功能。為了免去外部上拉電阻，當引腳規劃為輸入時，都可由內部連接到一個上拉電阻。這些上拉電阻可通過相關上拉電阻控制寄存器PAPU 和 PBPU 來設置，它用一個 PMOS 電晶體來實現上拉電阻功能。

注意，只有當共用功能引腳被選擇為數位輸入或 NMOS 輸出時，上拉功能才會受 PxPU 控制開啟，其它狀態時上拉功能不可用。

● **PxPU 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	PxPU7	PxPU6	PxPU5	PxPU4	PxPU3	PxPU2	PxPU1	PxPU0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

PxPUn：I/O Px 口上拉電阻功能控制位

- 0：除能
- 1：使能

PxPUn 位用於控制對應引腳的上拉電阻功能。這裡的 “x” 可以是埠A 和 B。但是，每個 I/O 埠實際有效位可能不同。

PA 口喚醒

當使用暫停指令 “HALT” 迫使單片機進入休眠或空閒模式，單片機的系統時鐘將會停止以降低功耗，此功能對於電池及低功耗應用很重要。喚醒單片機有很多種方法，其中之一就是使 PA 口的其中一個引腳從高電平轉為低電平。這個功能特別適合於通過外部開關來喚醒的應用。PA 口的每個引腳可以通過設置PAWU 寄存器來單獨選擇是否具有喚醒功能。

需要注意的是，只有當引腳被設置為通用 I/O 功能輸入類型且單片機處於休眠或空閒模式時，喚醒功能才會受喚醒控制寄存器控制。

● **PAWU 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	PAWU7	PAWU6	PAWU5	PAWU4	PAWU3	PAWU2	PAWU1	PAWU0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 PAWU7~PAWU0：PA7~PA0 喚醒功能控制位

- 0：除能
- 1：使能

輸入 / 輸出埠控制寄存器

每一個輸入/ 輸出口都具有各自的控制寄存器，即 PAC 和 PBC，用來控制輸入 / 輸出狀態。從而每個 I/O 引腳都可以通過軟體控制，動態的設置為 CMOS 輸出或輸入。所有的 I/O 埠的引腳都各自對應於 I/O 埠控制的某一位。若 I/O 引腳要實現輸入功能，則對應的控制寄存器的位元需要設置為 “1”。這時程式指令可以直接讀取輸入腳的邏輯狀態。若控制寄存器相應的位被設定為 “0”，則此引腳被設置為 CMOS 輸出。當引腳設置為輸出狀態時，程式指令讀取的是輸出埠寄存器的內容。注意，如果對輸出口做讀取動作時，程式讀取到的是內部輸出資料鎖存器中的狀態，而不是輸出引腳上實際的邏輯狀態。

● **PxC 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	PxC7	PxC6	PxC5	PxC4	PxC3	PxC2	PxC1	PxC0
R/W								
POR	1	1	1	1	1	1	1	1

PxCn : I/O Px 口輸入 / 輸出類型選擇位

0 : 輸出

1 : 輸入

PxCn 位元用於控制對應引腳的狀態類型。這裡的 “x” 可以是埠A 或 B。但是，每個 I/O 埠實際有效位可能不同。

引腳共用功能

引腳的多功能可以增加單片機應用的靈活性。有限的引腳個數將會限制設計者，而引腳的多功能將會解決很多此類問題。此外，這些引腳功能可以通過應用程式一系列寄存器進行設定。

引腳共用功能選擇寄存器

封裝中有限的引腳個數會對單片機某些功能造成影響。然而，引腳功能共用功能通過引腳功能選擇，使得小封裝單片機具有更多不同的功能。單片機包含埠 “x” 輸出功能選擇寄存器 “n” ，記為 PxSn。輸入功能選擇寄存器，記為IFS, 這些寄存器可以用來選擇多功能共用引腳的所需功能。

要注意的最重要一點是，確保所需的引腳共用功能被正確地選擇和取消。對於大部分引腳共用功能，要選擇所需的引腳共用功能，首先應通過相應的引腳共用控制寄存器正確地選擇該功能，然後在配置相應的週邊功能設置以使能週邊功能。但是，在設置相關引腳控制位元段時，一些數位輸入引腳如 INT, STCK, STPI 等，與對應的通用 I/O 口共用同一個引腳共用設置選項。要選擇這些引腳功能，除了上述的必要的引腳共用控制和週邊功能設置外，還必須將其對應的埠控制寄存器位設置為輸入。要正確地取消引腳共用功能，首先應除能週邊功能，然後再修改相應的引腳共用寄存器以選擇其他的共用功能。

寄存器名稱	位							
	7	6	5	4	3	2	1	0
PAS0	PAS07	PAS06	PAS05	PAS04	PAS03	PAS02	PAS01	PAS00
PAS1	PAS17	PAS16	PAS15	PAS14	PAS13	PAS12	PAS11	PAS10
PBS0	—	—	PBS05	PBS04	PBS03	PBS02	PBS01	PBS00
IFS	—	—	SCKSCLS1	SCKSCLS0	RXSDISDAS1	RXSDISDAS0	SCSBS1	SCSBS0

引腳共用功能選擇寄存器列表

● **PAS0 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	PAS07	PAS06	PAS05	PAS04	PAS03	PAS02	PAS01	PAS00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~6 PAS07~PAS06 : PA3 引腳共用功能選擇

00 : PA3

01 : AN0

10 : OPDA1P

11 : SCK/SCL

Bit 5~4

PAS05~PAS04 : PA2 引腳共用功能選擇

00 : PA2

01 : OPDA0N

10 : RX/SDI/SDA

11 : PA2

Bit 3~2

PAS03~PAS02 : PA1 引腳共用功能選擇

00 : PA1/INT

01 : AN1

10 : OPDA0O

11 : SCS

Bit 1~0 PAS01~PAS00 : PA0 引腳共用功能選擇
 00 : PA0
 01 : AN7
 10 : OPDA1O
 11 : SDO/TX

● **PAS1 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	PAS17	PAS16	PAS15	PAS14	PAS13	PAS12	PAS11	PAS10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~6 PAS17~PAS16 : PA7 引腳共用功能選擇
 00 : PA7
 01 : AN4
 10 : RX/SDI/SDA
 11 : PA7

Bit 5~4 PAS15~PAS14 : PA6 引腳共用功能選擇
 00 : PA6/STCK
 01 : AN2
 10 : SDO/TX
 11 : STP

Bit 3~2 PAS13~PAS12 : PA5 引腳共用功能選擇
 00 : PA5
 01 : OPDA1N
 10 : SCK/SCL
 11 : VREF

Bit 1~0 PAS11~PAS10 : PA4 引腳共用功能選擇
 00 : PA4
 01 : OPDA0P
 10 : RX/SDI/SDA 11 : PA4

● **PBS0 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	—	—	PBS05	PBS04	PBS03	PBS02	PBS01	PBS00
R/W	—	—	R/W	R/W	R/W	R/W	R/W	R/W
POR	—	—	0	0	0	0	0	0

Bit 7~6 未定義，讀為“0”

Bit 5~4 PBS05~PBS04 : PB2 引腳共用功能選擇
 00 : PB2
 01 : AN3
 10 : VREFI
 11 : SCK/SCL

Bit 3~2 PBS03~PBS02 : PB1 引腳共用功能選擇
 00 : PB1/STPI
 01 : AN6
 10 : SCS
 11 : PB1/STPI

Bit 1~0 PBS01~PBS00 : PB0 引腳共用功能選擇
 00 : PB0
 01 : AN5
 10 : SDO/TX
 11 : DACOUT

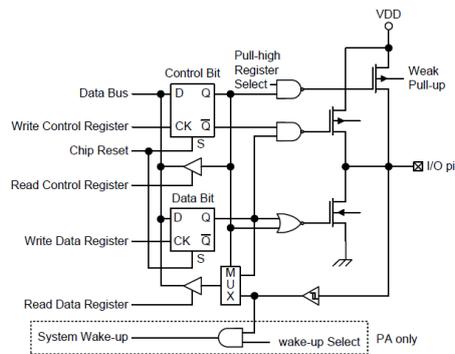
● IFS 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	SCKSCLS1	SCKSCLS0	RXSDISDAS1	RXSDISDAS0	SCSBS1	SCSBS0
R/W	—	—	R/W	R/W	R/W	R/W	R/W	R/W
POR	—	—	0	0	0	0	0	0

- Bit 7~6 未定義，讀為“0”
- Bit 5~4 SCKSCLS1~SCKSCLS0：SCK/SCL 輸入源引腳選擇
 - 00：PA5
 - 01：PA3
 - 10：PB2
 - 11：PB2
- Bit 3~2 RXSDISDAS1~RXSDISDAS0：RX/SDI/SDA 輸入源引腳選擇
 - 00：PA4
 - 01：PA2
 - 10：PA7
 - 11：PA7
- Bit 1~0 SCSBS1~SCSBS0：SCS 輸入源引腳選擇
 - 00：PB1
 - 01：PA1
 - 10：PA1
 - 11：PA1

輸入 / 輸出引腳結構

下圖為輸入 / 輸出引腳邏輯功能的內部結構圖。輸入 / 輸出引腳的準確邏輯結構圖可能與此圖不同，這裡只是為了方便對 I/O 引腳功能的理解提供的一個參考。由於存在諸多的引腳共用結構，在此不方便提供所有類型引腳功能結構圖。



邏輯功能輸入 / 輸出結構

程式設計注意事項

在程式設計中，最先要考慮的是埠的初始化。重定之後，所有的輸入 / 輸出資料及埠控制寄存器都將被設為邏輯高。所有輸入 / 輸出引腳預設為輸入狀態，而其電平則取決於其它相連接電路以及是否選擇了上拉電阻。如果埠控制寄存器某些引腳位元被設定輸出狀態，這些輸出引腳會有初始高電平輸出，除非資料寄存器埠在程式中被預先設定。設置哪些引腳是輸入及哪些引腳是輸出，可通過設置正確的值到適當的埠控制寄存器，或使用指令“SET [m].i”及“CLR [m].i”來設定埠控制寄存器中個別的位。注意，當使用這些位元元控制指令時，系統即將產生一個讀 - 修改 - 寫的操作。單片機需要先讀入整個埠上的資料，修改個別的位元，然後重新把這些資料寫入到輸出埠。

PA 口的每個引腳都帶喚醒功能。單片機處於休眠或空間模式時，有很多方法可以喚醒單片機，其中之一就是通過 PA 任一引腳電平從高到低轉換的方式，可以設置 PA 口一個或多個引腳具有喚醒功能。

計時器模組 – TM

控制和測量時間在任何單片機中都是一個很重要的部分。該單片機提供幾個計時器模組 (簡稱 TM)，來實現和時間有關的功能。計時器模組是包括多種操作的定時單元，用以提供的操作有：定時 / 計數器，捕捉輸入，比較匹配輸出，單脈衝輸出以及 PWM 輸出等功能。每個計時器模組有兩個獨立中斷。每個 TM 外加的輸入輸出引腳，擴大了計時器的靈活性，便於用戶使用。

簡介

該單片機包含 1 個標準型 TM，簡稱為 STM。標準型 TM 的一般特性將在本節中描述，詳細的操作將在標準型 TM 部分中進行描述。STM 的主要特性在相應的表中進行了總結。

功能	STM
定時 / 計數器	√
捕捉輸入	√
比較匹配輸出	√
PWM 輸出	√
單脈衝輸出	√
PWM 對齊方式	邊沿對齊
PWM 調節週期 & 占空比	占空比或週期

TM 功能概要

TM 操作

不同類型的 TM 提供從簡單的定時操作到 PWM 信號產生等多種功能。理解 TM 操作的關鍵是比較 TM 內獨立運行的計數器的值與內部比較器的預置值。當計數器的值與比較器的預置值相同時，則比較匹配，TM 中斷信號產生，清零計數器並改變 TM 輸出引腳的狀態。使用者選擇內部時鐘或外部時鐘來驅動內部 TM 計數器。

TM 時鐘源

驅動 TM 計數器的時鐘源很多。通過設置 STM 控制寄存器的 STCK2~STCK0 位，選擇所需的時鐘源。該時鐘源來自系統時鐘 fSYS 或內部高速時鐘 fH 或 fSUB 時鐘源或外部 STCK 引腳時鐘的分頻比。STCK 引腳時鐘源用於允許外部信號作為 TM 時鐘源或用於事件計數。

TM 中斷

標準型 TM 有兩個內部中斷，分別是內部比較器 A 或比較器 P。當比較匹配發生時產生 TM 中斷。當 TM 中斷產生時，計數器清零並改變 TM 的輸出信號。

TM 外部引腳

標準型 TM 有兩個 TM 輸入引腳，分別是 STCK 和 STPI。STCK 引腳可作為 STM 時鐘源輸入腳，通過設置 STMC0 寄存器中的 STCK2~STCK0 位進行選擇。外部時鐘源可通過該引腳來驅動內部 TM。STCK 引腳可選擇上升沿有效或下降沿有效。STCK 引腳也可用作單脈衝輸出模式的外部觸發引腳。

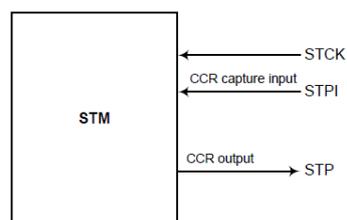
另一種 STM 輸入引腳 STPI 作為捕捉輸入腳，其有效邊沿有上升沿、下降沿和雙沿，通過設置 STMC1 寄存器中的 STIO1~STIO0 位來選擇有效邊沿類型。

該 TM 有一個輸出引腳 STP。當 TM 工作在比較匹配輸出模式且比較匹配發生時，此引腳會由 TM 控制切換到高電平或低電平或翻轉。外部 STP 輸出引腳也被 TM 用來產生 PWM 輸出波形。

因為 TM 的輸入和輸出引腳與其它功能共用，TM 輸入和輸出功能首先需要使用在引腳共用功能部分中描述的相關引腳共用功能選擇寄存器來設置。引腳共用功能選擇的詳細內容在引腳共用功能章節有描述。

STM	
輸入	輸出
STCK · STPI	STP

TM 外部引腳

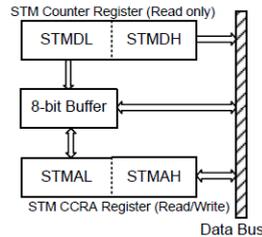


STM 功能引腳框圖

程式設計注意事項

TM 計數寄存器和捕捉/ 比較寄存器CCRA、CCRP，含有低位元組和高位元組結構。高位元組可直接訪問，低位元組則僅能通過一個內部 8-bit 的暫存器進行訪問，讀寫這些成對的寄存器需通過特殊的方式。值得注意的是 8-bit 暫存器的存取資料及相關低位元組的讀寫操作僅在其相應的高位元組讀取操作執行時發生。

CCRA 寄存器訪問方式如下圖所示，讀寫這些成對的寄存器需通過特殊的方式。建議使用 “MOV” 指令，通過以下步驟訪問 CCRA 低位元組寄存器，即 STMAL。若不採用以下步驟訪問 CCRA 將導致不可預期的結果。



讀寫流程如下步驟所示：

寫數據至 CCRA

步驟 1. 寫資料至低位元組寄存器 STMAL

注意：此時資料僅寫入 8-bit 暫存器。

步驟 2. 寫資料至高位元組寄存器 STMAH

注意：此時資料直接寫入高位元組寄存器，同時鎖存在 8-bit 暫存器中的資料寫入低位元組寄存器。

由計數器寄存器和 CCRA 中讀取資料

步驟 1. 由高位元組寄存器 STMDH 和 STMAH 讀取資料

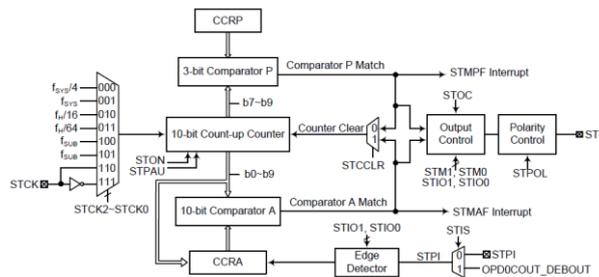
注意：此時高位元組寄存器中的資料直接讀取，同時由低位元組寄存器讀取的資料鎖存至 8-bit 暫存器中。

步驟 2. 由低位元組寄存器 STMDL 和 STMAL 讀取資料

注意：此時讀取 8-bit 暫存器中的資料。

標準型 TM – STM

標準型 TM 包括 5 種工作模式，即比較匹配輸出、定時/ 事件計數器、捕捉輸入、單脈衝輸出和 PWM 輸出模式。標準型 TM 由兩個外部輸入腳控制並驅動一個外部輸出腳。



標準型 TM 方框圖

注：1. STM STPI 輸入源可以從外部引腳 STPI 或內部 OPD0COUT_DEBOUT 信號中選擇，該信號是使用MUXSEL 寄存器中的 STIS 位選擇的。

2. 如果使用 STM 外部引腳，因這些引腳與其他功能共用引腳，在使用 STM 功能之前，應該正確地設置引腳共用功能寄存器。

標準型 TM 操作

標準型 TM 是 10 位寬度。核心是一個由使用者選擇的內部或外部時鐘源驅動的 10 位向上計數器，它還包括兩個內部比較器即比較器 A 和比較器 P。這兩個比較器將計數器的值與 CCRP 和 CCRA 寄存器中的值進行比較。CCRP 是 3 位寬度，與計數器的高 3 位比較；而 CCRA 是 10 位的，與計數器的所有位比較。通過應用程式改變 10 位元數目器值的唯一方法是使 STON 位發生上升沿跳變清除計數器。此外，計數器溢出或比較匹配也會自動清除計數器。上述條件發生時，通常情況會產生 STM 中斷信號。標準型 TM 可工作在不同的模式，可由包括來自兩個輸入腳的不同時鐘源驅動，也可以控制一個輸出腳。所有工作模式的設定都是通過設置相關寄存器來實現的。

標準型 TM 寄存器介紹

標準型 TM 的所有操作由一系列寄存器控制。一對唯讀寄存器用來存放 10 位元數目器的值，一對讀 / 寫寄存器存放 10 位 CCRA 的值，剩下兩個控制寄存器設置不同的操作和控制模式以及三個 CCRP 位元元。MUXSEL 寄存器用於選擇 STM 捕捉輸入信號。

寄存器名稱	位							
	7	6	5	4	3	2	1	0
STMC0	STPAU	STCK2	STCK1	STCK0	STON	STRP2	STRP1	STRP0
STMC1	STM1	STM0	STIO1	STIO0	STOC	STPOL	STDPX	STCCLR
STMDL	D7	D6	D5	D4	D3	D2	D1	D0
STMDH	—	—	—	—	—	—	D9	D8
STMAL	D7	D6	D5	D4	D3	D2	D1	D0
STMAH	—	—	—	—	—	—	D9	D8
MUXSEL	STIS	—	—	—	—	—	—	—

標準型 TM 寄存器列表

● **STMC0 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	STPAU	STCK2	STCK1	STCK0	STON	STRP2	STRP1	STRP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **STPAU** : STM 計數器暫停控制位

- 0 : 運行
- 1 : 暫停

通過設置此位為高可使計數器暫停，清零此位恢復正常計數器操作。當處於暫停條件時，STM 保持上電狀態並繼續耗電。當此位由低到高轉換時，計數器將保留其剩餘值，直到此位再次改變為低電平，並從此值開始繼續計數。

Bit 6~4 **STCK2~STCK0** : 選擇STM計數時鐘位

- 000 : fSYS/4
- 001 : fSYS
- 010 : fH/16
- 011 : fH/64
- 100 : fSUB
- 101 : fSUB
- 110 : STCK 上升沿時鐘
- 111 : STCK 下降沿時鐘

此三位元用於選擇 STM 的時鐘源。外部引腳時鐘源能被選擇在上升沿或下降沿有效。fSYS 是系統時鐘，fH 和 fSUB 是其它的內部時鐘源，詳細請參考振盪器章節。

Bit 3 **STON** : STM 計數器 On/Off 控制位

- 0 : Off
- 1 : On

此位元控制 STM 的總開關功能。設置此位為高則使能計數器使其運行，清零此位則除能 STM。清零此位將停止計數器並關閉 STM 減少耗電。當此位經由低到高轉換時，內部計數器將復位清零；當此位經由高到低轉換時，內部計數器將保持其剩餘值，直到此位再次改變為高電平。若 STM 處於比較匹配輸出模式時或 PWM 輸出模式或單脈衝輸出模式時，當 STON 位元元經由低到高轉換時，STM 輸出腳將復位至 STOC 位指定的初始值。

Bit 2~0 **STRP2~STRP0** : STM CCRP 3-bit 寄存器，與 STM 計數器 bit 9~bit 7 比較比較器 P 匹配週期 =

- 000 : 1024 個 STM 時鐘
- 001 : 128 個 STM 時鐘
- 010 : 256 個 STM 時鐘
- 011 : 384 個 STM 時鐘
- 100 : 512 個 STM 時鐘
- 101 : 640 個 STM 時鐘
- 110 : 768 個 STM 時鐘
- 111 : 896 個 STM 時鐘

此三位設定內部CCRP 3-bit 寄存器的值，然後與內部計數器的高三位進行比較。如果 STCCLR 位設定為 0 時，比較結果為 0 並清除內部計數器。STCCLR 位設定為低，內部計數器在比較器 P 比較匹配發生時被重置；由於 CCRP 只與計數器高三位比較，比較結果是 128 時鐘週期的倍數。CCRP 被清零時，實際上會使得計數器在最大值溢出。

● **STMC1 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	STM1	STM0	STIO1	STIO0	STOC	STPOL	STDPX	STCCLR
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~6 **STM1~STM0** : 選擇 STM 工作模式位元

- 00 : 比較匹配輸出模式
- 01 : 捕捉輸入模式
- 10 : PWM 輸出模式或單脈衝輸出模式
- 11 : 定時 / 計數器模式

這兩位元元設置 STM 需要的工作模式。為了確保操作可靠，STM 應在 STM1 和STM0 位有任何改變前先關掉。在定時/ 計數器模式，STM 輸出腳狀態未定義。

Bit 5~4 **STIO1~STIO0** : 選擇 STM STP 或 STP1 引腳功能位元比較匹配輸出模式

- 00 : 無變化
- 01 : 輸出低
- 10 : 輸出高
- 11 : 輸出翻轉

PWM 輸出模式 / 單脈衝輸出模式

- 00 : PWM 輸出無效狀態
- 01 : PWM 輸出有效狀態
- 10 : PWM 輸出
- 11 : 單脈衝輸出捕捉輸入模式
- 00 : 在 STP1 輸入信號的上升沿輸入捕捉
- 01 : 在 STP1 輸入信號的下降沿輸入捕捉
- 10 : 在 STP1 輸入信號的雙沿輸入捕捉
- 11 : 輸入捕捉除能定時 / 計數器模式未使用

此兩位用於決定在一定條件達到時 STM STP 或 STP1 如何改變狀態。這兩位值的選擇取決於 STM 運行在哪種模式下。

在比較匹配輸出模式下，STIO1 和 STIO0 位決定當比較器 A 比較匹配輸出發生時 STM 輸出腳如何改變狀態。當比較器 A 比較匹配輸出發生時 STM 輸出腳能設為切換高、切換低或翻轉當前狀態。若此兩位同時為 0 時，這個輸出將不會改變。STM 輸出腳的初始值通過 STMC1 寄存器的 STOC 位設置取得。注意，由 STIO1 和 STIO0 位得到的輸出電平必須與通過 STOC 位設置的初始值不同，否則當比較匹配發生時，STM 輸出腳將不會發生變化。在 STM 輸出腳改變狀態後，通過 STON 位元由低到高轉換復位至初始值。

在 PWM 輸出模式，STIO1 和 STIO0 用於決定比較匹配條件發生時怎樣改變 STM 輸出腳的狀態。PWM 輸出功能通過這兩位的變化進行更新。僅在 STM 關閉時改變 STIO1 和 STIO0 的值是很有必要的。若在 STM 運行時改變 STIO1 和STIO0 值，PWM 輸出的值是無法預料的。

Bit 3 **STOC** : STM STP 輸出控制位元比較匹配輸出模式

- 0 : 初始低 1 : 初始高

PWM 輸出模式 / 單脈衝輸出模式

- 0 : 低有效 1 : 高有效

這是 STM 輸出腳輸出控制位。它取決於 STM 此時正運行於比較匹配輸出模式還是 PWM 輸出模式，或單脈衝輸出模式。若 STM 處於定時 / 計數器模式，則其不受影響。在比較匹配輸出模式時，其決定比較匹配發生前 STM 輸出腳的邏輯電平值。在 PWM 輸出模式時，其決定 PWM 信號是高有效還是低有效。在單脈衝輸出模式，其決定 STON 位元由低變高時 STM 的邏輯電平。

Bit 2 **STPOL** : STM STP 輸出極性控制位

- 0 : 同相 1 : 反相

此位控制 STP 輸出腳的極性。此位為高時 STM 輸出腳反相，為低時 STM 輸出腳同相。若 STM 處於定時 / 計數器模式時其不受影響。

Bit 1 **STDPX** : STM PWM 週期 / 占空比控制位

- 0 : CCRP – 週期；CCRA – 占空比
- 1 : CCRP – 占空比；CCRA – 週期

此位決定 CCRA 與 CCRP 寄存器哪個被用於 PWM 波形的週期和占空比控制。

Bit 0 **STCCLR** : 選擇 STM 計數器清零條件位

- 0 : 比較器 P 匹配
- 1 : 比較器 A 匹配

此位用於選擇清除計數器的方法。標準型 TM 包括兩個比較器 – 比較器 A 和比較器 P。這兩個比較器每個都可以用作清除內部計數器。STCCLR 位設為高，計數器在比較器 A 比較匹配發生時被清除；此位設為低，計數器在比較器 P 比較匹配發生或計數器溢出時被清除。計數器溢出清除的方法僅在CCRP 被清除為 0 時才能生效。STCCLR 位元元在 PWM 輸出模式、單脈衝輸出模式和輸入捕捉模式時未使用。

● **STMDL 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R	R	R	R	R	R	R	R
POR	0	0	0	0	0	0	0	0

Bit 7~0 D7~D0 : STM 計數器低位元組寄存器 bit 7 ~ bit 0 STM 10-bit 計數器 bit 7 ~ bit 0

● **STMDH 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	D9	D8
R/W	—	—	—	—	—	—	R	R
POR	—	—	—	—	—	—	0	0

Bit 7~2 未定義，讀為 “0”

Bit 1~0 D9~D8 : STM 計數器高位元組寄存器 bit 1 ~ bit 0 STM 10-bit 計數器 bit 9 ~ bit 8

● **STMAL 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 D7~D0 : STM CCRA 低位元組寄存器 bit 7 ~ bit 0 STM 10-bit CCRA bit 7 ~ bit 0

● **STMAH 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	D9	D8
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	0	0

Bit 7~2 未定義，讀為 “0”

Bit 1~0 D9~D8 : STM CCRA 高位元組寄存器 bit 1 ~ bit 0 STM 10-bit CCRA bit 9 ~ bit 8

● **MUXSEL 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	STIS	—	—	—	—	—	—	—
R/W	R/W	—	—	—	—	—	—	—
POR	0	—	—	—	—	—	—	—

Bit 7 STIS : STPI 輸入信號選擇

0 : 來自 STPI 引腳 1 : 來自 OPD0COUT_DEBOUT

Bit 6~0 未定義，讀為 “0”

標準型 TM 工作模式

標準型 STM 有五種工作模式，即比較匹配輸出模式，PWM 輸出模式，單脈衝輸出模式，捕捉輸入模式或定時 / 計數器模式。通過設置 STMC1 寄存器的STM1 和 STM0 位元元選擇任意模式。

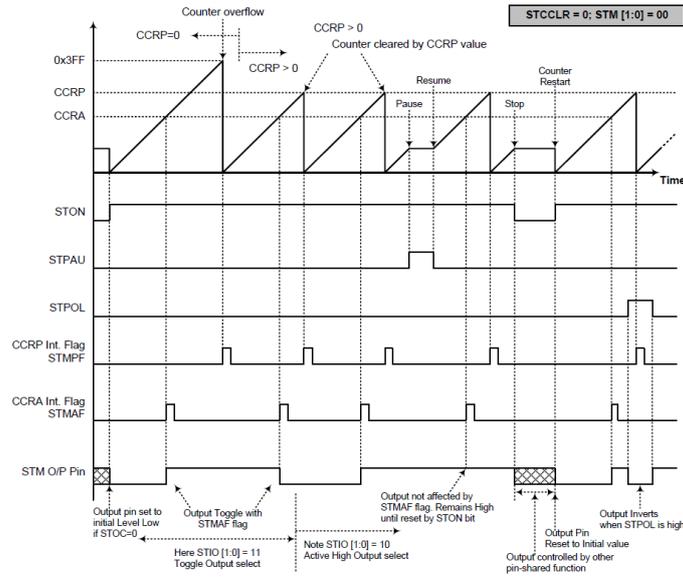
比較匹配輸出模式

為使 TM 工作在此模式，STMC1 寄存器中的 STM1 和 STM0 位元元需要設置為 “00”。當工作在該模式，一旦計數器使能並開始計數，有三種方法來清零，分別是：計數器溢出，比較器 A 比較匹配發生和比較器 P 比較匹配發生。當STCCLR 位為低，有兩種方法清除計數器。一種是比較器 P 比較匹配發生，另一種是 CCRP 所有位設置為零並使得計數器溢出。此時，比較器 A 和比較器 P 的請求標誌位元 STMAF 和 STMPF 將分別置位。

如果 STMC1 寄存器的 STCCLR 位設置為高，當比較器 A 比較匹配發生時計數器被清零。此時，即使 CCRP 寄存器的值小於 CCRA 寄存器的值，僅產生STMAF 插斷要求標誌。所以當 STCCLR 為高時，不會產生 STMPF 插斷要求標誌。在比較匹配輸出模式下，CCRA 不能設為 “0”。

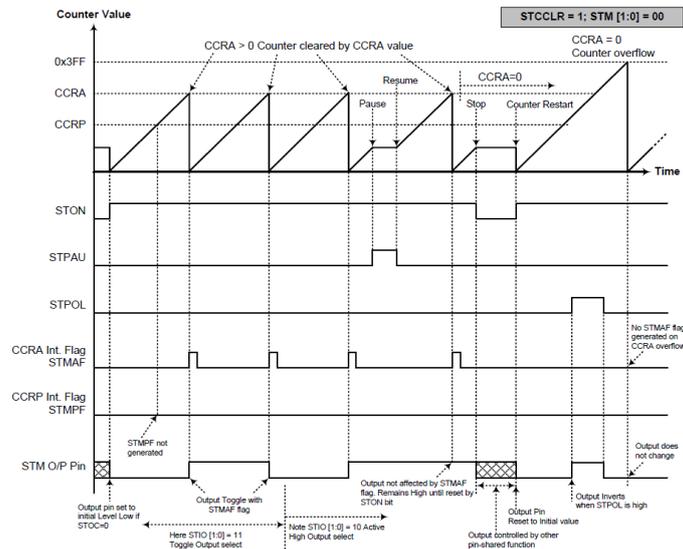
如果 CCRA 位都清除為零，當計數器的值達到最大值 3FF 時將溢出，但此時不會產生 STMAF 插斷要求標誌。

正如該模式名所言，當比較匹配發生後，STM 輸出腳狀態改變。當比較器 A 比較匹配發生後 STMAF 標誌產生時，STM 輸出腳狀態改變。比較器 P 比較匹配發生時產生的 STMPF 標誌不影響 STM 輸出腳。STM 輸出腳狀態改變方式由 STMC1 寄存器中 STIO1 和 STIO0 位決定。當比較器 A 比較匹配發生時，STIO1 和 STIO0 位決定 STM 輸出腳輸出高、低或翻轉當前狀態。在 STON 位由低到高電平的變化後，STM 輸出腳初始狀態為 STOC 位元所指的電平。注意，若 STIO1 和 STIO0 位同時為 0 時，引腳輸出不變。



比較匹配輸出模式 – STCCLR=0

- 注：1. STCCLR=0，比較器 P 匹配將清除計數器
- 2. STM 輸出腳僅由 STMAF 標誌位元控制在 STON 上升沿 STM 輸出腳復位至初始值



比較匹配輸出模式 – STCCLR=1

- 注：1. STCCLR=1，比較器 A 匹配將清除計數器 STM 輸出腳僅由 STMAF 標誌位元控制在 STON 上升沿 STM 輸出腳復位至初始值當 STCCLR=1 時，不會產生 STMPF 標誌

定時 / 計數器模式

為使 STM 工作在此模式，STMC1 寄存器中的 STM1 和 STM0 位元需要設置為 “11”。定時 / 計數器模式與比較輸出模式操作方式相同，並產生同樣的插斷要求標誌。不同的是，在定時 / 計數器模式下 STM 輸出腳未使用。因此，比較匹配輸出模式中的描述和時序圖可以適用於此功能。該模式中未使用的 STM 輸出腳用作普通 I/O 腳或其它功能。

PWM 輸出模式

為使 STM 工作在此模式，STMC1 寄存器中的 STM1 和 STM0 位元需要設置為 “10”，且 STIO1 和 STIO0 位元也需要設置為 “10”。STM 的 PWM 輸出功能在馬達控制，加熱控制，照明控制等方面十分有用。給 STM 輸出腳提供一個頻率固定但占空比可調的信號，將產生一個有效值等於 DC 均方根的 AC 方波。由於 PWM 波形的週期和占空比可調，其波形的選擇就極其靈活。在 PWM 輸出模式中，STCCLR 位元不影響 PWM 週期。CCRA 和 CCRP

寄存器決定 PWM 波形，一個用來清除內部計數器並控制 PWM 波形的頻率，另一個用來控制占空比。哪個寄存器控制頻率或占空比取決於 STMC1 寄存器的 STDPX 位。所以 PWM 波形頻率和占空比由 CCRA 和 CCRP 寄存器共同決定。當比較器 A 或比較器 P 比較匹配發生時，將產生 CCRA 或 CCRP 中斷標誌。

STMC1 寄存器中的 STOC 位決定 PWM 波形的極性，STIO1 和 STIO0 位使能 PWM 輸出或將 STM 輸出腳置為邏輯高或邏輯低。STPOL 位對 PWM 輸出波形的極性取反。

10-bit STM · PWM 輸出模式，邊沿對齊模式，STDPX=0

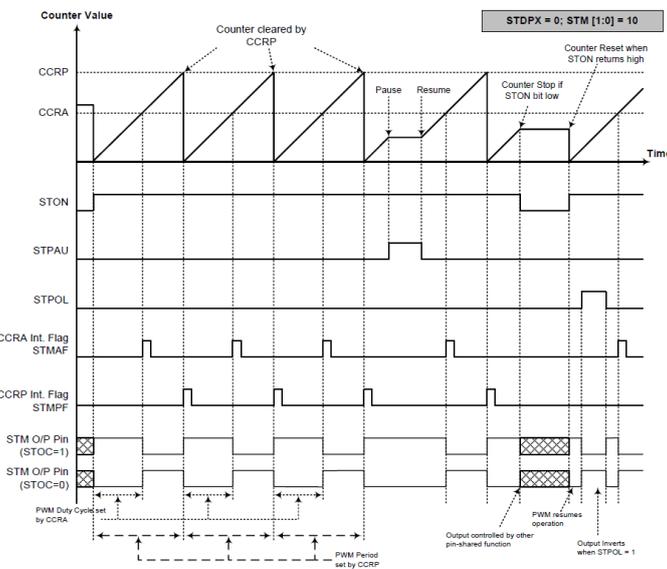
CCRP	001b	010b	011b	100b	101b	110b	111b	000b
週期	128	256	384	512	640	768	896	1024
占空比	CCRA							

若 $f_{SYS}=8MHz$ · STM 時鐘源選擇 $f_{SYS}/4$ · $CCRP=100b$ · $CCRA=128$ · STM PWM 輸出頻率 = $(f_{SYS}/4) / (512) = f_{SYS}/2048 = 4kHz$ · 占空比 = $128/(512) = 25%$ 若由 CCRA 寄存器定義的占空比值等於或大於週期值，PWM 輸出占空比為 100%

10-bit STM · PWM 輸出模式，邊沿對齊模式，STDPX=1

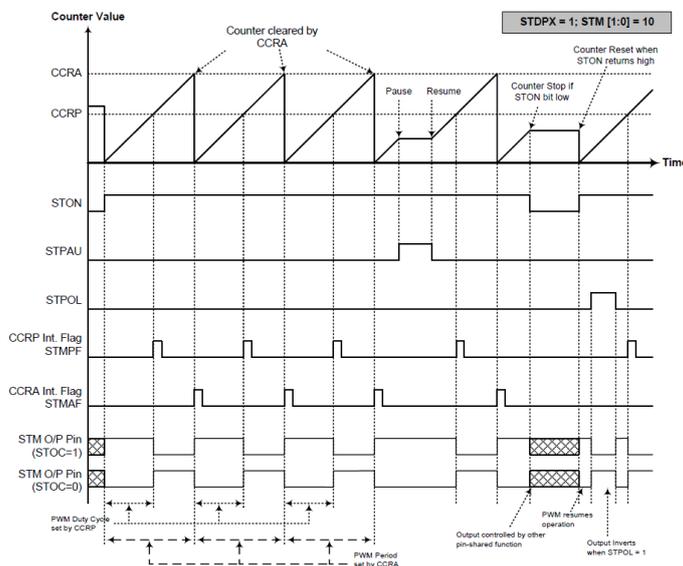
CCRP	001b	010b	011b	100b	101b	110b	111b	000b
週期	CCRA							
占空比	128	256	384	512	640	768	896	1024

PWM 的輸出週期由 CCRA 寄存器的值與 STM 的時鐘共同決定，PWM 的占空比由 CCRP 寄存器的值決定。



PWM 輸出模式 -STDPX=0

注：1. STDPX=0 · CCRP 清除計數器計數器清零並設置 PWM 週期當 STIO [1:0]=00 或 01 · PWM 輸出功能不變 STCCLR 位不影響 PWM 操作



PWM 輸出模式 - STDPX =1

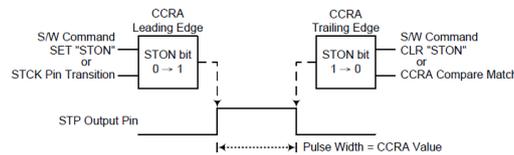
注：1. STDPX=1 · CCRA 清除計數器計數器清零並設置 PWM 週期當 STIO [1:0]=00 或 01 · PWM 輸出功能不變 STCCLR 位不影響 PWM 操作

單脈衝輸出模式

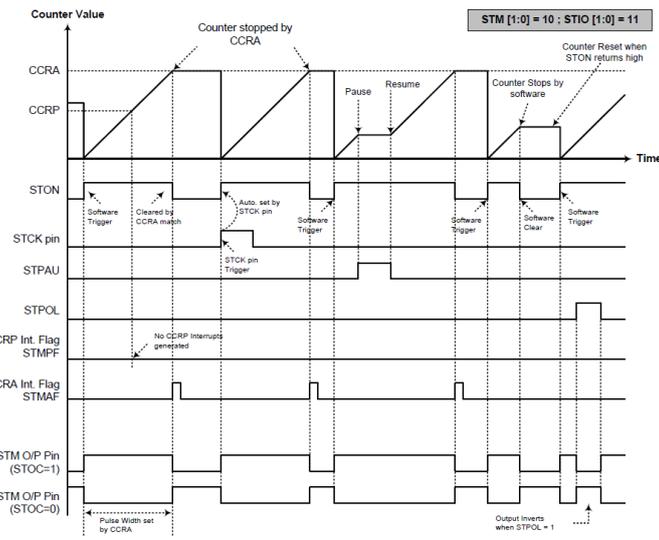
為使 STM 工作在此模式，STMC1 寄存器中的 STM1 和 STM0 位元元需要設置為 “10”，同時 STIO1 和 STIO0 位元元需要設置為 “11”。正如模式名所言，單脈衝輸出模式，在 STM 輸出腳將產生一個脈衝輸出。

通過應用程式控制 STON 位由低到高的轉變來觸發脈衝前沿輸出。而處於單脈衝輸出模式時，STON 位元元可在 STCK 腳發生有效邊沿跳轉時自動由低轉變為高，進而開始單脈衝輸出。當 STON 位轉變為高電平時，計數器將開始運行，並產生脈衝前沿。當脈衝有效時 STON 位保持高電平。通過應用程式使 STON 位元元清零或比較器 A 比較匹配發生時，產生脈衝後沿。

然而，比較器 A 比較匹配發生時，會自動清除 STON 位並產生單脈衝輸出邊沿轉換。CCRA 的值通過這種方式控制脈衝寬度。比較器 A 比較匹配發生時，也會產生 STM 中斷。STON 位在計數器重啟時會發生由低到高的轉變，此時計數器才復位至零。在單脈衝模式中，CCRP 寄存器，STCCLR 和 STDPX 位未使用。



單脈衝產生示意圖



單脈衝輸出模式

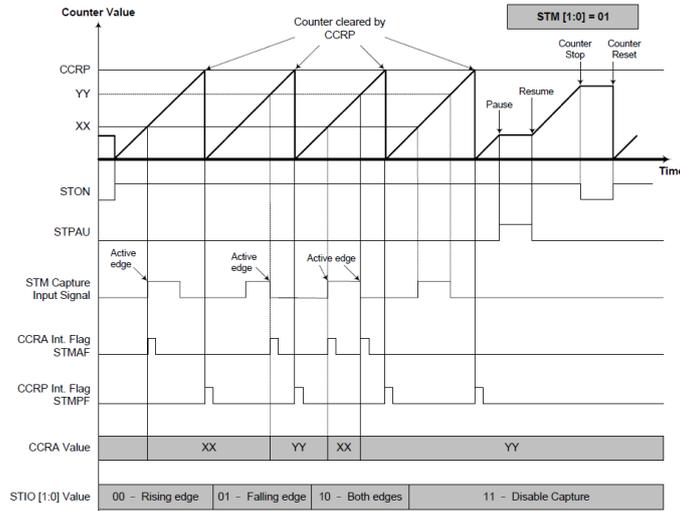
注：1. 通過 CCRA 匹配停止計數器 CCRP 未使用通過 STCK 腳或設置 STON 位為高來觸發脈衝 STCK 腳有效沿會自動置位 STON 單脈衝輸出模式中，STIO [1:0] 需置位 “11”，且不能更改。

捕捉輸入模式

為使 STM 工作在此模式，STMC1 寄存器中的 STM1 和 STM0 位元元需要設置為 “01”。此模式使能外部或內部信號捕捉並保存內部計數器當前值，因此被用於諸如脈衝寬度測量的應用中。外部或內部信號是使用 MUXSEL 寄存器中的 STIS 位選擇的。輸入信號可以是上升沿，下降沿或雙沿有效。使用 STMC1 寄存器中的 STIO1 和 STIO0 位來選擇有效邊沿轉換類型。計數器在 STON 位由低到高轉變時啟動並通過應用程式初始化。

當輸入信號出現有效邊沿轉換時，計數器當前值被鎖存到 CCRA 寄存器，並產生 STM 中斷。無論輸入信號發生哪種邊沿轉換，計數器繼續工作直到 STON 位發生下降沿跳變。當 CCRP 比較匹配發生時計數器復位至零；CCRP 的值通過這種方式控制計數器的最大值。當比較器 P CCRP 比較匹配發生時，也會產生 STM 中斷。記錄 CCRP 溢出中斷信號的值可以測量長脈寬。通過設置 STIO1 和 STIO0 位元選擇輸入信號為上升沿，下降沿或雙沿有效。如果 STIO1 和 STIO0 都設置為高，無論輸入信號發生哪種邊沿轉換都不會產生捕捉操作，但計數器繼續運行。STCCLR 和 STDPX 位元元在此模式中未使用。

當 STPI 引腳與其它功能共用，STM 工作在輸入捕捉模式時需多加注意。這是因為如果引腳被設為輸出，那麼該引腳上的任何電平轉變都可能執行輸入捕捉操作。STCCLR 和 STDPX 位元元在此模式中未使用。



捕捉輸入模式

注：1. STM[1:0]=01 並通過 STIO [1:0] 位設置有效邊沿STM 捕捉輸入腳的有效邊沿將計數器的值轉移到 CCRA 中STCCLR 位未使用無輸出功能 -STOC 和 STPOL 位未使用計數器值由 CCRP 決定，在 CCRP 為“0”時，計數器計數值可達最大

A/D 轉換器

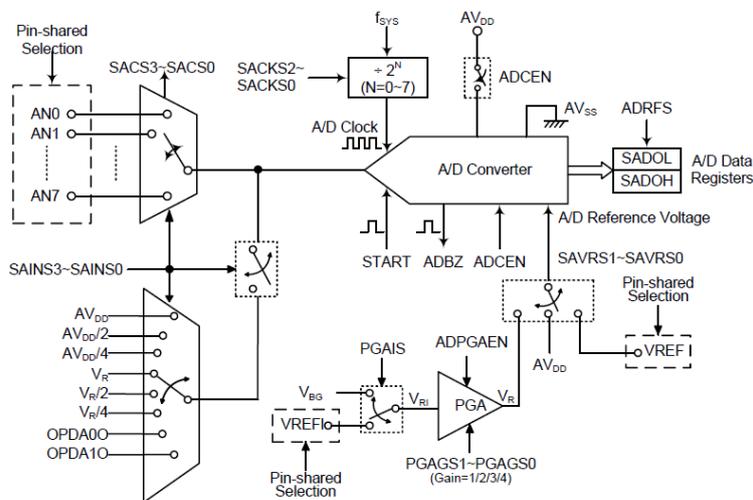
對於大多數電子系統而言，處理現實世界的類比信號是共同的需求。為了完全由單片機來處理這些信號，首先需要通過 A/D 轉換器將類比信號轉換成數位信號。將 A/D 轉換器電路集成入單片機，可有效的減少外部器件，隨之而來，具有降低成本和減少器件空間需求的優勢。

A/D 轉換器簡介

此單片機包含一個多通道的 A/D 轉換器，它們可以直接接入外部類比信號（來自感測器或其它控制信號）並直接將這些信號轉換成 12 位元的數位量。它們也可以轉換內部信號，比如將紅外線感應電路 OPAMP0 輸出，OPDA00 和紅外線感應電路 OPAMP1 輸出，OPDA10 轉換成 12 位元的數位量。選擇轉換外部或內部類比信號由 SAINS3~SAINS0 位和 SACS3~SACS0 位共同控制。當外部類比信號被轉換時，應該首先正確地配置相應的引腳共用控制位，然後使用 SAINS3~SAINS0 和 SACS3~SACS0 位來選擇外部通道輸入。注意，當轉換內部類比信號時，已選的外部輸入通道會自動斷開以避免故障。關於 A/D 輸入信號的詳細描述請參考“A/D 轉換控制寄存器”和“A/D 輸入信號”兩節內容。

外部輸入通道	內部信號	通道選擇位
AN0~AN7	AVDD, AVDD/2, AVDD/4, VR, VR/2, VR/4, OPDA00, OPDA10	SAINS3~SAINS0, SACS3~SACS0

下圖顯示了 A/D 轉換器內部結構和相關的寄存器。



A/D 轉換器結構

A/D 轉換寄存器介紹

A/D 轉換器的所有工作由五個寄存器控制。一對唯讀寄存器來存放 12 位 A/D 轉換資料的值。剩下三個控制寄存器設置 A/D 轉換器的操作和控制功能。

1 : ADC 輸出資料格式 → SADOH=D[11:8] ; SADOL=D[7:0]

此位元控制存放在兩個 A/D 資料寄存器中的 12 位 A/D 轉換結果的格式。細節方面請參考 A/D 轉換資料寄存器章節。

- Bit 3~0 **SACS3~SACS0** : A/D 轉換器外部類比頻道輸入選擇位
- 0000 : AN0
 - 0001 : AN1
 - 0010 : AN2
 - 0011 : AN3
 - 0100 : AN4
 - 0101 : AN5
 - 0110 : AN6
 - 0111 : AN7
 - 1000~1111 : 無定義，輸入浮空

● **SADC1 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	SAINS3	SAINS2	SAINS1	SAINS0	—	SACKS2	SACKS1	SACKS0
R/W	R/W	R/W	R/W	R/W	—	R/W	R/W	R/W
POR	0	0	0	0	—	0	0	0

- Bit 7~4 **SAINS3~SAINS0** : A/D 輸入信號選擇位元
- 0000 : 外部輸入 – 外部類比頻道輸入
 - 0001 : 內部輸入 – 內部 A/D 轉換器電源電壓 AVDD
 - 0010 : 內部輸入 – 內部 A/D 轉換器電源電壓 AVDD/2
 - 0011 : 內部輸入 – 內部 A/D 轉換器電源電壓 AVDD/4
 - 0100 : 外部輸入 – 外部類比頻道輸入
 - 0101 : 內部輸入 – 內部 A/D 轉換器 PGA 輸出電壓 VR
 - 0110 : 內部輸入 – 內部 A/D 轉換器 PGA 輸出電壓 VR/2
 - 0111 : 內部輸入 – 內部 A/D 轉換器 PGA 輸出電壓 VR/4
 - 1000 : 內部輸入 – 紅外線感應電路 OPAMP0 輸出 · OPDA00
 - 1001 : 內部輸入 – 紅外線感應電路 OPAMP1 輸出 · OPDA10
 - 1010~1011 : 保留 · 連接到地
 - 1100~1111 : 外部輸入 – 外部類比頻道輸入

當選擇內部類比信號時，無論 SACS3~SACS0 為何值，外部通道輸入信號都會自動關閉。

- Bit 3 未定義，讀為 “0”
- Bit 2~0 **SACKS2~SACKS0** : A/D 時鐘源選擇位元
- 000 : fSYS
 - 001 : fSYS/2
 - 010 : fSYS/4
 - 011 : fSYS/8
 - 100 : fSYS/16
 - 101 : fSYS/32
 - 110 : fSYS/64
 - 111 : fSYS/128

這三位元用於選擇 A/D 轉換器的時鐘源。

● **SADC2 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	ADPGAEN	—	—	PGAIS	SAVRS1	SAVRS0	PGAGS1	PGAGS0
R/W	R/W	—	—	R/W	R/W	R/W	R/W	R/W
POR	0	—	—	0	0	0	0	0

- Bit 7 **ADPGAEN** : PGA 使能 / 除能控制位
- 0 : 除能
 - 1 : 使能

當 PGA 輸出VR 作為 A/D 轉換器輸入或 A/D 轉換器參考電壓時，PGA 需要置高使能。否則 PGA 需把此位清零除能，以減小功耗。

- Bit 6~5 未定義，讀為 “0”

Bit 4 **PGAIS** : PGA 輸入 (VRI) 選擇位

0 : 外部 VREF1 引腳

1 : 內部獨立參考電壓 · VBG

當 VREF1 引腳的外部電壓和內部獨立參考電壓 VBG 同時被選擇作為 PGA 輸入時，硬體將只選擇內部電壓 VBG 作為 PGA 輸入。

Bit 3~2 **SAVRS1~SAVRS0** : A/D 轉換器參考電壓選擇位元

00 : 內部 A/D 轉換器電源 · AVDD

01 : VREF 引腳

1x : 內部 PGA 輸出電壓 · VR 這些位元用於選擇 A/D 轉換器的參考電壓。當內部 A/D 轉換器電源或內部 PGA 輸出電壓和 VREF 引腳的外部輸入電壓同時被選擇作為參考電壓時，硬體只能選擇內部參考電壓作為 A/D 轉換器參考電壓。

Bit 1~0 **PGAGS1~PGAGS0** : PGA 增益選擇位

00 : 增益 = 1

01 : 增益 = 2

10 : 增益 = 3

11 : 增益 = 4

A/D 操作

SADC0 寄存器中的 START 位，用於打開 A/D 轉換器。當單片機設置此位元從邏輯低到邏輯高，然後再到邏輯低，就會開始一個模數轉換週期。

SADC0 寄存器中的 ADBZ 位用於表明模數轉換過程是否正在進行。A/D 轉換成功啟動後，ADBZ 位會被單片機自動置為 1。在轉換週期結束後，ADBZ 位會自動置為 0。此外，也會置位中斷控制寄存器內相應的 A/D 插斷要求標誌位元，如果中斷使能，就會產生對應的內部中斷信號。A/D 內部中斷信號將引導程式跳轉到相應的 A/D 內部中斷位址。如果 A/D 內部中斷被禁止，可以讓單片機輪詢 SADC0 寄存器中的 ADBZ 位，檢查此位是否被清除，作為另一種偵測 A/D 轉換週期結束的方法。

A/D 轉換器的時鐘源為系統時鐘 fSYS 或其分頻，而分頻係數由 SADC1 寄存器中的 SACKS2~SACKS0 位決定。雖然 A/D 時鐘源是由系統時鐘 fSYS 和 SACKS2~SACKS0 位決定，但可選擇的 A/D 時鐘源則有一些限制。由於允許的 A/D 時鐘週期 tADCK 的範圍為 0.5μs~10μs，所以選擇系統時鐘速度時就必須小心。如果系統時鐘速度為 8MHz 時，SACKS2~SACKS0 位不能設為 “000”、 “001” 或 “111”。必須保證設置的 A/D 轉換時鐘週期不小於時鐘週期的最小值或大於時鐘週期的最大值，否則將會產生不準確的 A/D 轉換值。使用者可以參考下面的表格，被標上星號 * 的數值是不允許的，因為這些值可能超出所指定的 A/D 時鐘週期範圍。

fSYS	A/D 時鐘週期 (tADCK)							
	SACKS[2:0]=000 (fSYS)	SACKS[2:0]=001 (fSYS/2)	SACKS[2:0]=010 (fSYS/4)	SACKS[2:0]=011 (fSYS/8)	SACKS[2:0]=100 (fSYS/16)	SACKS[2:0]=101 (fSYS/32)	SACKS[2:0]=110 (fSYS/64)	SACKS[2:0]=111 (fSYS/128)
1MHz	1μs	2μs	4μs	8μs	16μs *	32μs *	64μs *	128μs *
2MHz	500ns	1μs	2μs	4μs	8μs	16μs *	32μs *	64μs *
4MHz	250ns *	500ns	1μs	2μs	4μs	8μs	16μs *	32μs *
8MHz	125ns *	250ns *	500ns	1μs	2μs	4μs	8μs	16μs *

A/D 時鐘週期範例

SADC0 寄存器中的 ADCEN 位元用於控制 A/D 轉換電路電源的開啟和關閉。該位必須置高以開啟 A/D 轉換器電源。當設置 ADCEN 位為高開啟 A/D 轉換器內部電路時，在 A/D 轉換成功開啟前需一段延時。即使選擇無引腳作為 A/D 輸入，如果 ADCEN 設為 “1”，那麼仍然會產生功耗。因此在功耗敏感的應用中，當未使用 A/D 轉換器功能時，建議設置 ADCEN 為低以減少功耗。

A/D 轉換器參考電壓

A/D 轉換器參考電壓來自正電源電壓 AVDD、外部參考源引腳 VREF 或內部 PGA 輸出電壓 VR。通過 SAVRS1 和 SAVRS0 位來選擇。當 SAVRS 位設置為 “00” 時，A/D 轉換器參考電壓將來自 AVDD。若 SAVRS 位設置為 “01” 時，A/D 轉換器參考電壓將來自 VREF 引腳。否則，A/D 轉換器參考電壓將來自 PGA 輸出 VR。由於 VREF 引腳與其它功能共用引腳，當 VREF 引腳被選擇為參考電壓引腳時，需合理設置相關的引腳共用功能選擇寄存器位，以選擇 VREF 引腳功能。此外，若選擇內部參考電壓 AVDD 或 VR 作為 A/D 轉換器參考電壓，那麼硬體將忽略 VREF 引腳電壓將只選擇內部參考電壓作為 A/D 轉換器參考電壓輸入。類比輸入值不得超過所選參考電壓 VREF 的值。

A/D 轉換器也有一個 VREF1 引腳，它可以作為 PGA 輸入以產生 A/D 轉換器參考電壓。若要選擇 VREF1 引腳輸入作為 PGA 輸入來源，需將在 SADC2 寄存器中的 PGAIS 位清零，並且應該正確配置相關引腳共用功能選擇寄存器以選擇 VREF1 引腳功能。然而，PGA 輸入可以從內部獨立參考電壓 VBG 中提供。如果應用程式同時選擇 VREF1 引腳上的外部電壓和內部電壓 VBG 作為 PGA 輸入，那麼硬體將只選擇內部參考電壓 VBG 作為 PGA 輸入。

SAVRS[1:0]	參考電壓源	說明
00	AVDD	內部 A/D 轉換器電源電壓
01	VREF 引腳	外部 A/D 轉換器參考引腳 VREF
10 或 11	VR	內部 A/D 轉換器 PGA 輸出電壓

A/D 轉換器參考電壓選擇

A/D 轉換器輸入信號

所有的 A/D 模擬輸入引腳都與 I/O 引腳及其它功能共用。使用 PxS0 和 PxS1 寄存器中的相應引腳共用功能選擇位，可以將它們設置為 A/D 轉換器模擬輸入腳或具有其它功能。如果對應的引腳作為 A/D 轉換輸入，那麼它原來的引腳功能將除能。通過這種方式，引腳的功能可由程式來控制，靈活地切換引腳功能。如果將引腳設為 A/D 輸入，則通過寄存器程式設計設置的所有上拉電阻會自動斷開。請注意，埠控制寄存器不需要為使能 A/D 輸入而先設定為輸入模式，當 A/D 輸入功能選擇位使能 A/D 輸入時，埠控制寄存器的狀態將被重置。

若 SAINS3~SAINS0 位被設置為 “0000” ， “0100” 或 “1100~1111” ，外部類比頻道輸入被選擇來轉換且 SACS3~SACS0 位被用來決定選擇哪個外部通道輸入被轉換。若 SAINS3~SAINS0 位被設置為 “0001~0011” ，選擇特定比率 AVDD、AVDD/2 或 AVDD/4 電壓進行轉換。若 SAINS3~SAINS0 位被設置為 “0101~0111” 時，選擇 PGA 輸出電壓 VR、VR/2 或 VR/4 進行轉換。若 SAINS3~SAINS0 位被設置為 “1000” 時，選擇紅外線感應電路 OPAMP0 輸出，OPDA00 進行轉換。若 SAINS3~SAINS0 位被設置為 “1001” 時，選擇紅外線感應電路 OPAMP1 輸出，OPDA10 進行轉換。

注意：當程式同時選擇外部信號 (AN0~AN7) 和內部信號 (AVDD、AVDD/2、AVDD/4、VR、VR/2、VR/4、OPDA00 或 OPDA10) 作為 A/D 轉換器輸入信號時，硬體將只選擇內部信號作為 A/D 轉換器輸入，外部信號將自動關閉。

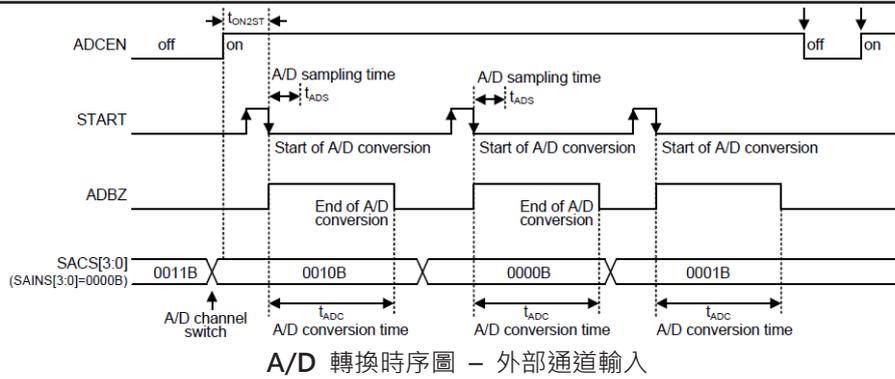
SAINS[3:0]	SACS[3:0]	輸入信號	描述
0000, 0100, 1100~1111	0000~0111	AN0~AN7	外部引腳模擬輸入
	1000~1111	—	輸入浮空，未選擇外部通道
0001	xxxx	AVDD	內部 A/D 轉換器電源電壓 AVDD
0010	xxxx	AVDD/2	內部 A/D 轉換器電源電壓 AVDD/2
0011	xxxx	AVDD/4	內部 A/D 轉換器電源電壓 AVDD/4
0101	xxxx	VR	內部 A/D 轉換器 PGA 輸出電壓 VR
0110	xxxx	VR/2	內部 A/D 轉換器 PGA 輸出電壓 VR/2
0111	xxxx	VR/4	內部 A/D 轉換器 PGA 輸出電壓 VR/4
1000	xxxx	OPDA00	紅外線感應電路 OPAMP0 輸出
1001	xxxx	OPDA10	紅外線感應電路 OPAMP1 輸出
1010~1011	xxxx	—	保留，連接到地

A/D 轉換器輸入信號選擇

A/D 轉換率及時序圖

一個完整的 A/D 轉換包含兩部分，資料採樣和資料轉換。資料採樣時間定義為 tADS，需要 4 個 A/D 時鐘週期，而資料轉換需要 12 個 A/D 時鐘週期。所以一個完整的 A/D 轉換時間，tADC，一共需要 16 個 A/D 時鐘週期。

最大 A/D 轉換率 = A/D 時鐘週期 /16 下列時序圖表示模數轉換過程中不同階段的圖形與時序。由應用程式控制開始 A/D 轉換過程後，單片機的內部硬體就會開始進行轉換，在這個過程中，程式可以繼續其它功能。A/D 轉換時間為 16tADCK，tADCK 為 A/D 時鐘週期。



A/D 轉換步驟概述

下面概述實現 A/D 轉換過程的各個步驟。

- 步驟 1：通過 SADC1 寄存器中的 SACKS2~SACKS0 位，選擇所需的 A/D 轉換時鐘。
- 步驟 2：將 SADC0 寄存器中的 ADCEN 位置高使能 A/D 轉換器。
- 步驟 3：通過 SADC1 寄存器中的 SAINS3~SAINS0 位，選擇連接至內部 A/D 轉換器的信號。若選擇外部通道輸入，接著執行步驟 4。若選擇內部類比信號，接著執行步驟 5。
- 步驟 4：若已通過 SAINS 位元選擇 A/D 輸入信號來自外部通道輸入，接著應設置相關的引腳共用功能控制位將該引腳規劃為 A/D 輸入引腳。通過設置 SACS 位選擇哪個類比頻道。接著執行步驟 6。
- 步驟 5：若已通過 SAINS 位元選擇 A/D 輸入信號來自內部類比信號，無論 SACS 為何值，外部通道輸入都會自動斷開。接著執行步驟 6。
- 步驟 6：通過在 SADC2 寄存器中配置 SAVRS1~SAVRS0 位元來選擇參考電壓源。如果選擇了 PGA 輸出電壓作為 A/D 轉換器的參考電壓，則需使能 PGA，然後在 SADC2 寄存器中配置 PGAIS 位元來選擇 PGA 輸入源。
- 步驟 7：在 SADC0 寄存器中設置 ADRFS 位選擇 A/D 轉換器輸出資料格式。
- 步驟 8：如果要使用中斷，則中斷控制寄存器需要正確地設置，以確保 A/D 中斷功能是啟動的。總中斷控制位元 EMI 需要置位元元為 “1”，以及 A/D 轉換器中斷控制位元 ADE 也需要置位元元為 “1”。
- 步驟 9：現在可以通過設置 START 位從 “0” 到 “1” 再回到 “0”，開始模數轉換的過程。
- 步驟 10：如果 A/D 轉換正在進行中，ADBZ 位元會被置為邏輯高。A/D 轉換完成後，ADBZ 位元會被置為邏輯低，並可從 SADOH 和 SADOL 寄存器中讀取輸出資料。

注：若使用輪詢 SADC0 寄存器中 ADBZ 位元的狀態的方法來檢查轉換過程是否結束時，則中斷使能的步驟可以省略。

程式設計注意事項

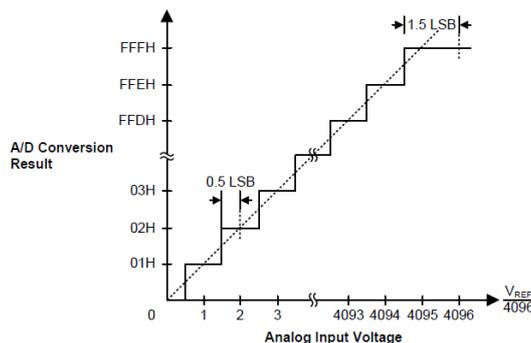
在程式設計時，如果 A/D 轉換器未使用，通過設置 SADC0 寄存器中的 ADCEN 為低，關閉 A/D 內部電路以減少電源功耗。此時，不考慮輸入腳的類比電壓，內部 A/D 轉換器電路不產生功耗。如果 A/D 轉換器輸入腳用作普通 I/O 腳，必須特別注意，輸入電壓為無效邏輯電平也可能增加功耗。

A/D 轉換功能

該單片機含有一組 12 位的 A/D 轉換器，它們轉換的最大值可達 0FFFH。由於類比輸入最大值等於 VREF 的電壓值，因此每一位可表示 VREF/4096 的模擬輸入值。

1 LSB = VREF ÷ 4096 通過下麵的等式可估算 A/D 轉換器輸入電壓值：A/D 輸入電壓 = A/D 數位輸出值 × (VREF ÷ 4096)

下圖顯示 A/D 轉換器類比輸入值和數位輸出值之間理想的轉換功能。除了數位化數值 0，其後的數位化數值會在精確點之前的 0.5 LSB 處改變，而數位化數值的最大值將在 VREF 之前的 1.5 LSB 處改變。注意，VREF 電壓是由 SAVRS 欄位確定的實際的 A/D 轉換器參考電壓。



A/D 轉換應用範例

下面兩個範常式用來說明怎樣使用 A/D 轉換。第一個範例是輪詢 SADC0 寄存器中的 ADBZ 位來判斷 A/D 轉換是否完成；第二個範例則使用中斷的方式判斷。

範例 1：使用查詢 ADBZ 的方式來檢測轉換結束

```

clr ADE                ; disable ADC interrupt
mov a,03H
mov SADC1,a            ; select fsys/8 as A/D clock
mov a,40h              ; setup PAS0 to configure pin AN0
mov PAS0,a
mov a,20h
mov SADC0,a
mov a,00h
mov SADC2,a           ; enable and connect AN0 channel to A/D converter
:
:
start_conversion:
clr START             ; high pulse on start bit to initiate conversion
set START             ; reset A/D
clr START             ; start A/D
polling_EOC:
sz ADBZ               ; poll the SADC0 register ADBZ bit to detect end
                    ; of A/D conversion
jmp polling_EOC       ; continue polling
mov a,SADOL            ; read low byte conversion result value
mov SADOL_buffer,a    ; save result to user defined register
mov a,SADOH            ; read high byte conversion result value
mov SADOH_buffer,a    ; save result to user defined register
:
:
jmp start_conversion  ; start next A/D conversion

```

範例 2：使用中斷的方式來檢測轉換結束

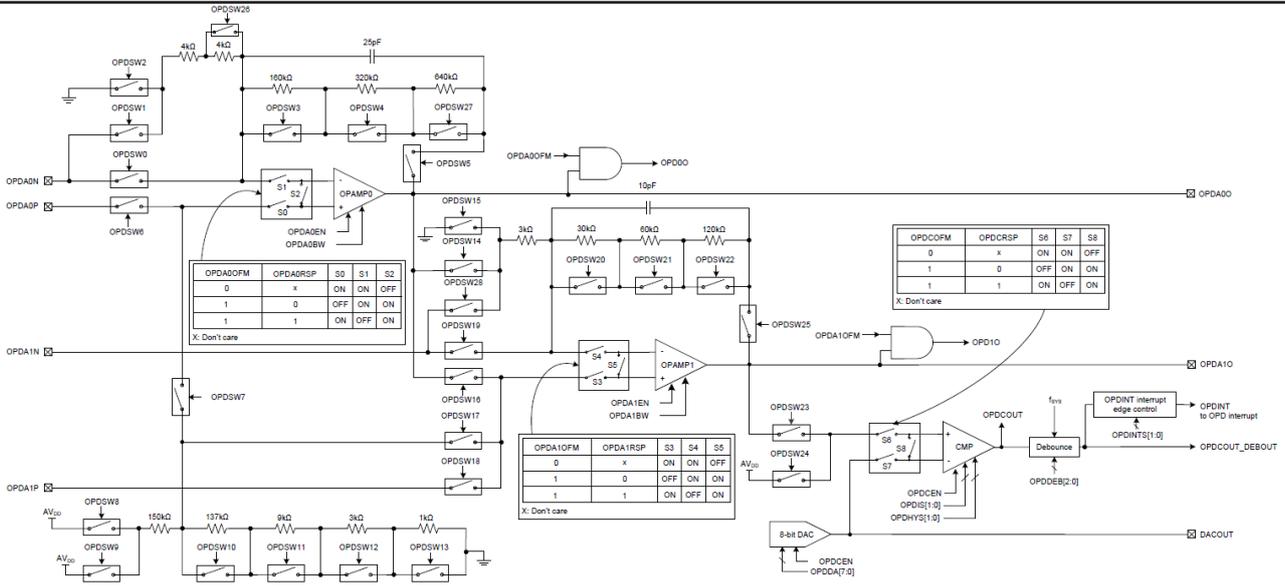
```

clr ADE                ; disable ADC interrupt
mov a,03H
mov SADC1,a            ; select fsys/8 as A/D clock
mov a,40h              ; setup PAS0 to configure pin AN0
mov PAS0,a
mov a,20h
mov SADC0,a
mov a,00h
mov SADC2,a           ; enable and connect AN0 channel to A/D converter
:
:
start_conversion:
clr START             ; high pulse on START bit to initiate conversion
set START             ; reset A/D
clr START             ; start A/D
clr ADF               ; clear ADC interrupt request flag
set ADE               ; enable ADC interrupt
set EMI               ; enable global interrupt
:
:
; ADC interrupt service routine
ADC_ISR:
mov acc_stack,a       ; save ACC to user defined memory
mov a,STATUS
mov status_stack,a    ; save STATUS to user defined memory
:
:
mov a,SADOL            ; read low byte conversion result value
mov SADOL_buffer,a    ; save result to user defined register
mov a,SADOH            ; read high byte conversion result value
mov SADOH_buffer,a    ; save result to user defined register
:
:
EXIT_INT_ISR:
mov a,status_stack
mov STATUS,a          ; restore STATUS from user defined memory
mov a,acc_stack       ; restore ACC from user defined memory
reti

```

OPA 電路

該單片機包括一個紅外線感應電路，由兩個運算放大器、一個比較器和一個 8 位 D/A 轉換器組成。兩級運算放大器電路可以放大一個小信號，增益範圍從 1 到 19600。使用者可以根據不同的應用程式選擇不同的組合，無論是反相放大或是同相放大。然後將放大的類比信號通過與 D/A 轉換器輸出參考電壓進行比較。



紅外線感應電路方框圖

紅外線感應電路操作

該電路由 OPDA0N 和 OPDA0P 輸入源電壓信號。第一級運算放大器通過開關 OPDSW0~OPDSW7 和 OPDSW26~OPDSW27 選擇不同的放大模式。同樣，第二級運算放大器也可以通過開關 OPDSW14~OPDSW22 和 OPDSW25 和 OPDSW28 進行模式選擇。兩個運算放大器組成一個 PGA 功能，選擇 PGA 的電壓從正極或是負極輸入，相應的可以產生正的或負的增益。OPAMP0 的增益可以通過 OPDSW3~OPDSW4 和 OPDSW26~OPDSW27 選擇 1x~280x 的放大倍數；而 OPAMP1 的增益可以通過 OPDSW20~OPDSW22 選擇 1x~70x 的放大倍數。對於反相放大模式，共模電壓可由開關 OPDSW8~OPDSW13 進行設置，電壓範圍為 VSS~AVDD/2。

D/A 轉換器僅用於比較器產生參考電壓。比較器可以將放大的輸入電壓和此參考電壓進行比較，比較結果經去抖處理後產生穩定的判斷信號 OPDCOUT。另外，使用 OPDC0 寄存器中的 OPDINTS0~OPDINTS1 位選擇正確的中斷邊沿類型。若有感應到紅外線，此信號將觸發中斷以告知 MCU 做出處理。

穩定信號也是 OPDINT 去抖後的信號，它可以通過 STIS 位選擇內部連接到 STPI 引腳並用於 STM 的捕捉輸入功能。

輸入信號經 OPAMP0/OPAMP1 放大後可以直接輸出到 OPDA00/OPDA10 引腳上，並且通過設置相關的寄存器來讀取放大的輸入電壓，並在內部連接到 A/D 轉換器。

紅外線感應電路寄存器

紅外線感應電路的所有操作由一系列寄存器控制。OPDSWA~OPDSWD 寄存器用於類比開關控制。OPDC0 寄存器用於運算放大器、比較器和 D/A 轉換器的使能 / 除能控制以及比較器去抖時間選擇。OPDC0 寄存器也用於控制 OPDINT 中斷邊沿。OPDC1 寄存器用於比較器遲滯電壓及偏置電流的控制以及運算放大器低電流或高頻寬的選擇。OPDDA 寄存器用於控制 D/A 轉換器的輸出電壓。OPDACAL 和 OPDCCAL 寄存器用於控制運算放大器和比較器輸入失調電壓校準功能。

寄存器名稱	位							
	7	6	5	4	3	2	1	0
OPDSWA	OPDSW7	OPDSW6	OPDSW5	OPDSW4	OPDSW3	OPDSW2	OPDSW1	OPDSW0
OPDSWB	OPDSW15	OPDSW14	OPDSW13	OPDSW12	OPDSW11	OPDSW10	OPDSW9	OPDSW8
OPDSWC	OPDSW23	OPDSW22	OPDSW21	OPDSW20	OPDSW19	OPDSW18	OPDSW17	OPDSW16
OPDSWD	—	—	—	OPDSW28	OPDSW27	OPDSW26	OPDSW25	OPDSW24
OPDC0	OPDA1EN	OPDA0EN	OPDCEN	OPDINTS1	OPDINTS0	OPDDEB2	OPDDEB1	OPDDEB0
OPDC1	OPD10	OPD00	OPDHYS1	OPDHYS0	OPDIS1	OPDIS0	OPDA1BW	OPDA0BW
OPDDA	D7	D6	D5	D4	D3	D2	D1	D0
OPDA0CAL	OPDA0OFM	OPDA0RSP	OPDA0OF5	OPDA0OF4	OPDA0OF3	OPDA0OF2	OPDA0OF1	OPDA0OF0
OPDA1CAL	OPDA1OFM	OPDA1RSP	OPDA1OF5	OPDA1OF4	OPDA1OF3	OPDA1OF2	OPDA1OF1	OPDA1OF0
OPDCCAL	OPDCOUT	OPDCOFM	OPDCRSP	OPDCOF4	OPDCOF3	OPDCOF2	OPDCOF1	OPDCOF0

紅外線感應電路寄存器清單

● **OPDSWA 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	OPDSW7	OPDSW6	OPDSW5	OPDSW4	OPDSW3	OPDSW2	OPDSW1	OPDSW0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 OPDSW7 : OPDSW7 開關 on/off 控制位元

0 : Off

1 : On

Bit 6 OPDSW6 : OPDSW6 開關 on/off 控制位元

0 : Off

1 : On

Bit 5 OPDSW5 : OPDSW5 開關 on/off 控制位元

0 : Off

1 : On

Bit 4 OPDSW4 : OPDSW4 開關 on/off 控制位元

0 : Off

1 : On

Bit 3 OPDSW3 : OPDSW3 開關 on/off 控制位元

0 : Off

1 : On

Bit 2 OPDSW2 : OPDSW2 開關 on/off 控制位元

0 : Off

1 : On

Bit 1 OPDSW1 : OPDSW1 開關 on/off 控制位元

0 : Off

1 : On

Bit 0 OPDSW0 : OPDSW0 開關 on/off 控制位元

0 : Off

1 : On

● **OPDSWB 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	OPDSW15	OPDSW14	OPDSW13	OPDSW12	OPDSW11	OPDSW10	OPDSW9	OPDSW8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 OPDSW15 : OPDSW15 開關 on/off 控制位元

0 : Off

1 : On

Bit 6 OPDSW14 : OPDSW14 開關 on/off 控制位元

0 : Off

1 : On

Bit 5 OPDSW13 : OPDSW13 開關 on/off 控制位元

0 : Off

1 : On

Bit 4 OPDSW12 : OPDSW12 開關 on/off 控制位元

0 : Off

1 : On

Bit 3 OPDSW11 : OPDSW11 開關 on/off 控制位元

0 : Off

1 : On

Bit 2 OPDSW10 : OPDSW10 開關 on/off 控制位元

0 : Off

1 : On

Bit 1 OPDSW9 : OPDSW9 開關 on/off 控制位元

0 : Off

1 : On

Bit 0 OPDSW8 : OPDSW8 開關 on/off 控制位元

0 : Off

1 : On

● **OPDSWC 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	OPDSW23	OPDSW22	OPDSW21	OPDSW20	OPDSW19	OPDSW18	OPDSW17	OPDSW16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 OPDSW23 : OPDSW23 開關 on/off 控制位元

0 : Off

1 : On

Bit 6 OPDSW22 : OPDSW22 開關 on/off 控制位元

0 : Off

1 : On

Bit 5 OPDSW21 : OPDSW21 開關 on/off 控制位元

0 : Off

1 : On

Bit 4 OPDSW20 : OPDSW20 開關 on/off 控制位元

0 : Off

1 : On

Bit 3 OPDSW19 : OPDSW19 開關 on/off 控制位元

0 : Off

1 : On

0 : Off

1 : On

Bit 2 OPDSW18 : OPDSW18 開關 on/off 控制位元

Bit 1 OPDSW17 : OPDSW17 開關 on/off 控制位元

0 : Off

1 : On

Bit 0 OPDSW16 : OPDSW16 開關 on/off 控制位元

0 : Off

1 : On

● **OPDSWD 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	OPDSW28	OPDSW27	OPDSW26	OPDSW25	OPDSW24
R/W	—	—	—	R/W	R/W	R/W	R/W	R/W
POR	—	—	—	0	0	0	0	0

Bit 7~5 未定義，讀為 "0"

Bit 4 OPDSW28 : OPDSW28 開關 on/off 控制位元

0 : Off

1 : On

Bit 3 OPDSW27 : OPDSW27 開關 on/off 控制位元

0 : Off

1 : On

Bit 2 OPDSW26 : OPDSW26 開關 on/off 控制位元

0 : Off

1 : On

Bit 1 OPDSW25 : OPDSW25 開關 on/off 控制位元

0 : Off

1 : On
 Bit 0 OPDSW24 : OPDSW24 開關 on/off 控制位元
 0 : Off
 1 : On

● **OPDC0 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	OPDA1EN	OPDA0EN	OPDCEN	OPDINTS1	OPDINTS0	OPDDEB2	OPDDEB1	OPDDEB0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **OPDA1EN** : OPD OPAMP1 使能 / 除能控制位
 0 : 除能
 1 : 使能

當此位被清除為 0 時，OPDA0O 將除能並輸出三態。

Bit 6 **OPDA0EN** : OPD OPAMP0 使能 / 除能控制位
 0 : 除能
 1 : 使能

當此位被清除為 0 時，OPDA1O 將除能並輸出三態。

Bit 5 **OPDCEN** : OPD 比較器和 D/A 轉換器使能 / 除能控制位
 0 : 除能
 1 : 使能

當此位被清除為 0 時，比較器輸出將被拉低，D/A 轉換器輸出三態。

Bit 4~3 **OPDINTS1~OPDINTS0** : OPDINT 中斷邊沿控制位
 00 : 除能
 01 : 上升沿
 10 : 下降沿
 11 : 雙沿

Bit 2~0 **OPDDEB2~OPDDEB0** : OPD 比較器去抖時間控制位
 000 : 旁路，無去抖
 001 : (1~2)×tDEB
 010 : (3~4)×tDEB
 011 : (7~8)×tDEB
 100 : (15~16)×tDEB
 101 : (31~32)×tDEB
 110 : (63~64)×tDEB
 111 : (127~128)×tDEB

注 : tDEB=1/fsys

● **OPDC1 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	OPD1O	OPD0O	OPDHYS1	OPDHYS0	OPDIS1	OPDIS0	OPDA1BW	OPDA0BW
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **OPD1O** : OPAMP1 輸出失調校準

Bit 6 **OPD0O** : OPAMP0 輸出失調校準

Bit 5~4 **OPDHYS1~OPDHYS0** : OPD 比較器滯滯電壓視窗控制位元具體參數情況見比較器電氣特性表

Bit 3~2 **OPDIS1~OPDIS0** : OPD 比較器偏置電流控制位元具體參數情況見比較器電氣特性表

Bit 1 **OPDA1BW** : OPD OPAMP1 低電流 / 高頻寬選擇位

0 : 低電流
 1 : 高頻寬

Bit 0 **OPDA0BW** : OPD OPAMP0 低電流 / 高頻寬選擇位

0 : 低電流
 1 : 高頻寬

● **OPDDA 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 D7~D0 : OPD D/A 轉換器輸出電壓控制位元 $DAC\ VOUT = (AVDD/256) \times D[7:0]$

● **OPDA0CAL 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	OPDA0OFM	OPDA0RSP	OPDA0OF5	OPDA0OF4	OPDA0OF3	OPDA0OF2	OPDA0OF1	OPDA0OF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	1	0	0	0	0	0

Bit 7 **OPDA0OFM** : OPD OPAMP0 正常操作或輸入失調電壓校準模式選擇位元

0 : 正常操作
1 : 失調校準模式

Bit 6 **OPDA0RSP** : OPD OPAMP0 輸入失調電壓校準參考選擇位元

0 : 選擇反相輸入端作為參考輸入
1 : 選擇同相輸入端作為參考輸入

Bit 5~0 **OPDA0OF5~OPDA0OF0** : OPD OPAMP0 輸入失調電壓校準控制位元

● **OPDA1CAL 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	OPDA1OFM	OPDA1RSP	OPDA1OF5	OPDA1OF4	OPDA1OF3	OPDA1OF2	OPDA1OF1	OPDA1OF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	1	0	0	0	0	0

Bit 7 **OPDA1OFM** : OPD OPAMP1 正常操作或輸入失調電壓校準模式選擇位元

0 : 正常操作
1 : 輸入失調電壓校準模式

Bit 6 **OPDA1RSP** : OPD OPAMP1 輸入失調電壓校準參考選擇位元

0 : 選擇反相輸入端作為參考輸入
1 : 選擇同相輸入端作為參考輸入

Bit 5~0 **OPDA1OF5~OPDA1OF0** : OPD OPAMP1 輸入失調電壓校準控制位元

● **OPDCCAL 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	OPDCOUT	OPDCOFM	OPDCRSP	OPDCOF4	OPDCOF3	OPDCOF2	OPDCOF1	OPDCOF0
R/W	R	R/W						
POR	0	0	0	1	0	0	0	0

Bit 7 **OPDCOUT** : OPD 比較器輸出

0 : 同相端輸入電壓 < DAC 輸出電壓
1 : 同相端輸入電壓 > DAC 輸出電壓當比較器除能時，比較器輸出為 0。

Bit 6 **OPDCOFM** : OPD 比較器正常操作或輸入失調電壓校準模式選擇位元

0 : 正常操作
1 : 失調校準模式

Bit 5 **OPDCRSP** : OPD 比較器輸入失調電壓校準參考選擇位元

0 : 選擇反相輸入端作為參考輸入
1 : 選擇同相輸入端作為參考輸入

Bit 4~0 **OPDCOF4~OPDCOF0** : OPD 比較器輸入失調電壓校準控制位元

輸入失調校準

為使運算放大器 n 或比較器工作在輸入失調校準模式，應先將 OPDAnOFM 或 OPDCOFM 位設置為 1，然後再設置 OPDAnRSP 或 OPDCRSP 位元選擇參考輸入電壓。除了設置相應的控制位外，運算放大器 n 和比較器的輸入失調校準步驟基本相似。

需注意的是因運算放大器 n 輸入引腳與其它功能共用 I/O 口，因此也要正確設置引腳共用功能選擇寄存器，選擇運算放大器引腳輸入功能。

運算放大器輸入失調校準

步驟 1：設置 OPDAnOFM=1 和 OPDAnRSP=1，運算放大器 n 進入失調電壓校準模式。為確保校準後 VOOS 降到最小值，校準模式下的輸入參考電壓大小必須與運算放大器模式下輸入的直流工作電壓相同。

步驟 2：設置 OPDAnOF[5:0]=000000，讀取 OPDnO 值。

步驟 3：將 OPDAnOF[5:0] 值加一，然後再讀取 OPDnO 值。

如果 OPDnO 值沒有改變，重複步驟 3 直到 OPDnO 值發生改變；

如果 OPDnO 值發生改變，記錄此時 OPDAnOF[5:0] 值為 VOOS1，接著執行步驟 4。

步驟 4：設置 OPDAnOF[5:0]=11111，讀取 OPDnO 值。

步驟 5：將 OPDAnOF[5:0] 值減一然後再讀取 OPDnO 值。

如果 OPDnO 值沒有改變，重複步驟 5 直到 OPDnO 值發生改變；

如果 OPDnO 值發生改變，記錄此時 OPDAnOF[5:0] 值為 VOOS2，接著執行步驟 6。

步驟 6：重新存儲 OPDAnOF[5:0]=VOOS=(VOOS1+VOOS2)/2，校準完成。若 (VOOS1+VOOS2)/2 非整數，則只取整數部分。

比較器輸入失調校準

步驟 1：設置 OPDCOFM=1 和 OPDCRSP=1，比較器進入失調電壓校準模式。為確保校準後 VCOS 降到最小值，校準模式下的輸入參考電壓大小必須與比較器模式下輸入的直流工作電壓相同。

步驟 2：設置 OPDCOF[4:0]=00000，讀取 OPDCOUT 位的值。

步驟 3：將 OPDCOF[4:0] 值加一，然後再讀取 OPDCOUT 位的值。

如果 OPDCOUT 位元的狀態沒有改變，重複步驟 3 直到 OPDCOUT 位發生改變；如果 OPDCOUT 位元的狀態發生改變，記錄此時 OPDCOF[4:0] 值為 VCOS1，接著執行步驟 4。

步驟 4：設置 OPDCOF[4:0]=11111，讀取 OPDCOUT 位的值。

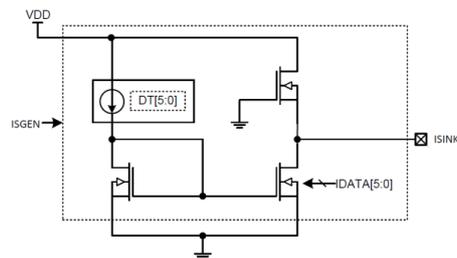
步驟 5：將 OPDCOF[4:0] 值減一然後再讀取 OPDCOUT 位的值。

如果 OPDCOUT 位元的狀態沒有改變，重複步驟 5 直到 OPDCOUT 位發生改變；如果 OPDCOUT 位元的狀態發生改變，記錄此時 OPDCOF[4:0] 值為 VCOS2，接著執行步驟 6。

步驟 6：重新存儲 OPDCOF[4:0]=VCOS=(VCOS1+VCOS2)/2，校準完成。若 (VCOS1+VCOS2)/2 非整數，則只取整數部分。

灌電流發生器

無論電壓為 2.2V~5.5V 範圍中何值，灌電流發生器都可以提供恒定的電流。恒流值由 IDATA 寄存器控制，灌電流範圍為 5mA~320mA。



灌電流發生器方框圖

灌電流發生器寄存器

灌電流發生器的功能是通過寄存器 IDATA 控制。主要進行灌電流發生器的使能/除能控制和 ISINK 引腳的灌電流設置。

● IDATA 寄存器

Bit	7	6	5	4	3	2	1	0
Name	ISGEN	—	D5	D4	D3	D2	D1	D0
R/W	R/W	—	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	—	0	0	0	0	0	0

Bit 7 ISGEN：灌電流發生器使能 / 除能控制位

0：除能 1：使能

當 ISGEN=0 時，ISINK 引腳狀態為：VISINK = 浮空，ISINK=0。

Bit 6 未定義，讀為“0”

Bit 5~0 D5~D0：ISINK 引腳的灌電流控制位電流值 (mA)=5+5×(D[5:0])

通用序列介面模組 – USIM

該單片機內有一個通用序列介面模組，包括易與外部設備通信的序列介面：四線 SPI 介面、兩線 I²C 介面和兩線 UART 介面。這種介面具有相當簡單的通信協議，單片機可以通過這些介面與感測器、快閃記憶體或 EEPROM 記憶體等硬體設備通信。USIM 介面的引腳與 I/O 引腳共用，所以要使用 USIM 功能時應先通過相應的引腳功能選擇位。因為所有的介面共用引腳和寄存器，所以要通過一個 SIMC0 寄存器中的 SIM2~SIM0 位來選擇 UART、SPI 還是 I²C 通信介面，使用 UMD 中的 UART 模式選擇位元元和 SPI/I²C 工作模式控制位元元。若 USIM 功能使能並且對應的引腳被用作 USIM 輸入引腳，可通過上拉電阻控制寄存器選擇與輸入 / 輸出共用的 USIM 腳的上拉電阻。

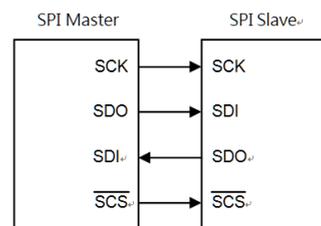
SPI 介面

SPI 介面常用於與外部設備如感測器、快閃記憶體或 EEPROM 記憶體等通信。四線 SPI 介面最初是由摩托羅拉公司研製，是一個有相當簡單的通信協定的串列資料介面，這個協定可以簡化與外部硬體的程式設計要求。

SPI 通信模式為全雙工模式，且能以主 / 從模式的工作方式進行通信，單片機既可以做為主機，也可以做為從機。雖然 SPI 介面理論上允許一個主機控制多個從機，但此處的 SPI 中只有一個片選信號引腳 SCS。若主機需要控制多個從機，可使用輸入 / 輸出引腳選擇從機。

SPI 介面操作

SPI 介面是一個全雙工串列資料傳輸器。SPI 介面的四線為：SDI、SDO、SCK 和 SCS。SDI 和 SDO 是資料的輸入和輸出線。SCK 是串列時鐘線，SCS 是從機的選擇線。SPI 的介面引腳與普通 I/O 口和 I²C/UART 的功能腳共用。通過設定 SIMC0/SIMC2 寄存器的對應位和相關引腳重置控制位，來使能 SPI 介面引腳功能。連接到 SPI 介面的單片機以從主 / 從模式進行通信，且主機完成所有的資料傳輸初始化，並控制時鐘信號。由於單片機只有一個 SCS 引腳，所以只能擁有一個從機設備。可通過軟體控制 SCS 引腳使能與除能，設置 CSEN 位為 “1” 使能 SCS 功能，設置 CSEN 位為 “0”，SCS 引腳將處於浮空狀態。

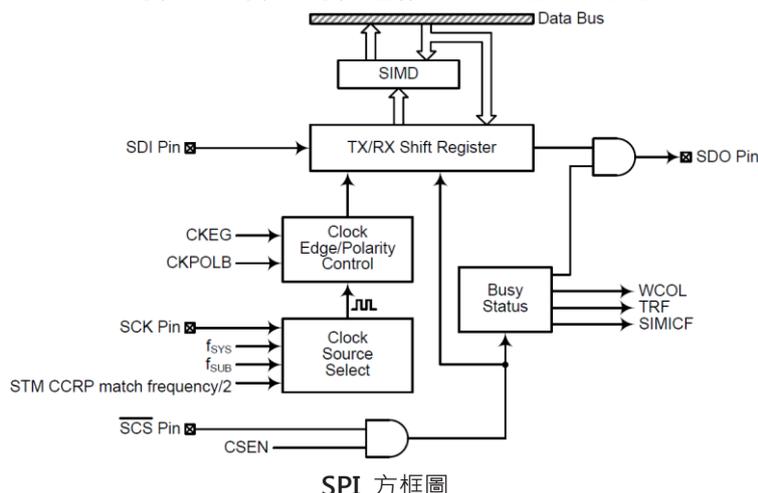


SPI 主機 / 從機連接方式

該單片機的 SPI 功能具有以下特點：

- 全雙工同步資料傳輸
- 主從模式
- 最低有效位先傳或最高有效位元先傳的資料傳輸模式
- 傳輸完成標誌位元
- 時鐘源上升沿或下降沿有效

SPI 介面狀態受很多因素的影響，如單片機處於主機或從機的工作模式和 CSEN、SIMEN 位元的狀態。



SPI 方框圖

SPI 寄存器

有三個內部寄存器用於控制 SPI 介面的所有操作，其中有一個資料寄存器SIMD、兩個控制寄存器 SIMC0 和 SIMC2。注意，SIMC2 和 SIMD 寄存器以及它們的 POR 值，僅在 SIMC0 寄存器中正確配置 UMD 和 SIM2~SIM0 位元時，選擇 SPI 模式。

寄存器名稱	位							
	7	6	5	4	3	2	1	0
SIMC0	SIM2	SIM1	SIM0	UMD	SIMDEB1	SIMDEB0	SIMEN	SIMICF
SIMC2	D7	D6	CKPOLB	CKEG	MLS	CSEN	WCOL	TRF
SIMD	D7	D6	D5	D4	D3	D2	D1	D0

SPI 寄存器列表

SPI 資料寄存器

SIMD 寄存器用於存儲發送和接收的資料。這個寄存器由SPI 和 I²C 功能所共用。在單片機尚未將資料寫入到 SPI 匯流排中時，要傳輸的資料應先存在 SIMD 中。

SPI 匯流排接收到資料之後，單片機就可以從 SIMD 資料寄存器中讀取。所有通過 SPI 傳輸或接收的資料都必須通過 SIMD 實現。

● SIMD 寄存器

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	x	x	x	x	x	x	x	x

“x” : 未知

Bit 7~0 D7~D0 : USIM SPI/I²C 資料寄存器 bit 7 ~ bit 0

SPI 控制寄存器

單片機中也有兩個控制 SPI 介面功能的寄存器，SIMC0 和 SIMC2。寄存器SIMC0 用於控制使能 / 除能功能和設置資料傳輸的時鐘頻率。寄存器 SIMC2 用於其它的控制功能如 LSB/MSB 選擇，寫衝突標誌位元等。

● SIMC0 寄存器

Bit	7	6	5	4	3	2	1	0
Name	SIM2	SIM1	SIM0	UMD	SIMDEB1	SIMDEB0	SIMEN	SIMICF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	1	1	1	0	0	0	0	0

Bit 7~5 SIM2~SIM0 : USIM SPI/I²C 工作模式控制位

000 : SPI 主機模式 ; SPI 時鐘為 fSYS/4

001 : SPI 主機模式 ; SPI 時鐘為 fSYS/16

010 : SPI 主機模式 ; SPI 時鐘為 fSYS/64

011 : SPI 主機模式 ; SPI 時鐘為 fSUB

100 : SPI 主機模式 ; SPI 時鐘為 STM CCRP 匹配頻率 / 2

101 : SPI 從機模式

110 : I²C 從機模式

111 : 未使用模式

當 UMD 位清除為零時，這幾位用於設置 USIM 功能中 SPI 或 I²C 的工作模式，用於選擇 SPI 的主從模式和 SPI 的主機時鐘頻率及 I²C 或 SPI 功能。SPI 時鐘源可來自於系統時鐘也可以選擇來自 STM 和 fSUB。若選擇的是作為 SPI 從機，則其時鐘源從外部主機而得。

Bit 4 UMD : UART 模式選擇位元元

0 : SPI 或 I²C 模式

1 : UART 模式

此位元元用於選擇 UART 模式。當此位清除為零，實際 SPI 或 I²C 模式可以使用SIM2~SIM0 位選擇。注意，在 SPI 或 I²C 模式中 UMD 位元元必須置零。

Bit 3~2 SIMDEB1~SIMDEB0 : I²C 去抖時間選擇位

這些位元只有在 USIM 設置成 I²C 介面模式時才有效。請參考 I²C 寄存器部分。

Bit 1 **SIMEN** : USIM SPI/I²C 使能控制位
0 : 除能
1 : 使能

此位元為 USIM SPI/I²C 介面的開 / 關控制位。此位元為 “0” 時，SIM 介面除能，SDI、SDO、SCK 和 SCS 或 SDA 和 SCL 腳將失去 SPI 或 I²C 功能，SIM 工作電流減小到最小值。此位元為 “1” 時，USIM SPI/I²C 介面使能。若 USIM 經由 UMD 和 SIM2~SIM0 位設置為工作在 SPI 介面，當 SIMEN 位元由低到高轉變時，SPI 控制寄存器中的設置不會發生變化，其首先應在應用程式中初始化。若 USIM 經由 UMD 和 SIM2~SIM0 位設置為工作在 I²C 介面，當 SIMEN 位元由低到高轉變時，I²C 控制寄存器中的設置，如 HTX 和 TXAK，將不會發生變化，其首先應在應用程式中初始化，此時相關 I²C 標誌，如 HCF、HAAS、HBB、SRW 和 RXAK，將被設置為其預設狀態。

Bit 0 **SIMICF** : USIM SPI 未完成標誌位元
0 : 未發生
1 : 發生

此位元僅當 USIM 配置在 SPI 從機模式時有效。如果 SPI 工作在從機模式且 SIMEN 和 CSEN 位都為 “1”，但在 SPI 資料傳輸完全結束前 SCS 線被外部主機拉高，SIMICF 和 TRF 位元都會被置高。在這種情況下，如果相應的中斷功能使能將產生一個中斷。然而，如果 SIMICF 位元是由軟體應用程式設為 1，那麼 TRF 位將不會置高。

● **SIMC2 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	CKPOLB	CKEG	MLS	CSEN	WCOL	TRF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~6 **D7~D6** : 未定義位

使用者可通過軟體程式對這兩位元進行讀寫。

Bit 5 **CKPOLB** : SPI 時鐘線的基礎選擇位元
0 : 當時鐘無效時，SCK 口為高電平
1 : 當時鐘無效時，SCK 口為低電平

此位元決定了時鐘線的基礎狀態，當時鐘無效時，若此位為高，SCK 為低電平，若此位為低，SCK 為高電平。

Bit 4 **CKEG** : SPI 的 SCK 有效時鐘邊沿類型位元
CKPOLB=0
0 : SCK 為高電平且在 SCK 上升沿抓取資料
1 : SCK 為高電平且在 SCK 下降沿抓取資料
CKPOLB=1
0 : SCK 為低電平且在 SCK 下降沿抓取資料
1 : SCK 為低電平且在 SCK 上升沿抓取資料

CKEG 和 CKPOLB 位元用於設置 SPI 匯流排上時鐘信號輸入和輸出方式。在執行資料傳輸前，這兩位必須被設置，否則將產生錯誤的時鐘邊沿信號。CKPOLB 位元決定時鐘線的基本狀態，若時鐘無效且此位元為高，則 SCK 為低電平，若時鐘無效且此位元為低，則 SCK 為高電平。CKEG 位元決定有效時鐘邊沿類型，取決於 CKPOLB 的狀態。

Bit 3 **MLS** : SPI 資料移位元順序位元
0 : LSB 優先
1 : MSB 優先

資料移位元選擇位元，用於選擇資料傳輸時高位優先傳輸還是低位元優先傳輸。此位元元設置為高時高位優先傳輸，為低時低位優先傳輸。

Bit 2 **CSEN** : SPI SCS 引腳控制位
0 : 除能
1 : 使能

CSEN 位用於 SCS 引腳的使能 / 除能控制。此位元為低時，SCS 除能並處於浮空狀態。此位元為高時，SCS 使能並作為選擇腳。

Bit 1 **WCOL** : SPI 寫衝突標誌位元
0 : 無衝突
1 : 衝突

WCOL 標誌位元用於監測資料衝突的發生。此位元為高時，資料在傳輸時被寫入 SIMD 寄存器。若資料正在被傳輸時，此操作無效。此位元可被應用程式清零。

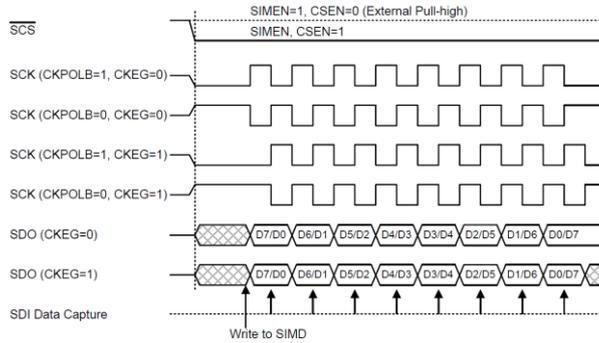
Bit 0 **TRF** : SPI 發送 / 接收結束標誌位元
0 : 資料正在發送
1 : 資料發送結束

TRF 位元為發送 / 接收結束標誌位元，當 SPI 資料傳輸結束時，此位自動置為高，但需通過應用程式設置為“0”。此位也可用於產生中斷。

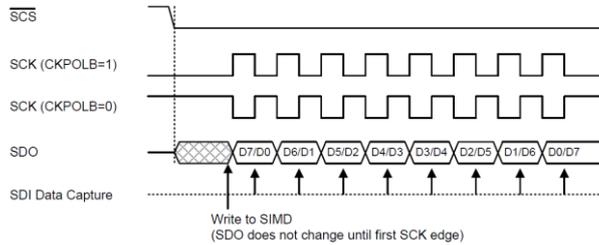
SPI 通信

將 SIMEN 設置為高，使能 SPI 功能之後，單片機處於主機模式，當資料寫入到寄存器 SIMD 的同時傳輸 / 接收開始進行。資料傳輸完成時，TRF 位將自動被置位但清除只能通過應用程式完成。單片機處於從機模式時，收到主機發來的信號之後，會傳輸 SIMD 中的資料，而且在 SDI 引腳上的資料也會被移位元到 SIMD 寄存器中。主機應在輸出時鐘信號之前先輸出一個 SCS 信號以使能從機，從機的資料傳輸功能也應在與 SCK 信號相關的適當時候準備就緒，這由 CKPOLB 和 CKEG 位決定。所附時序圖表明瞭在 CKPOLB 和 CKEG 位元各種設置情況下從機資料與 SCK 信號的關係。

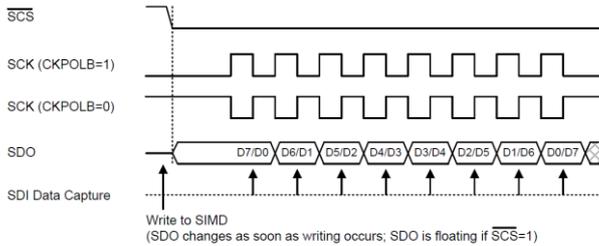
即使在單片機處於空閒模式，如果所選的 SPI 時鐘源有效，SPI 主機功能仍將繼續執行。



SPI 主機模式時序

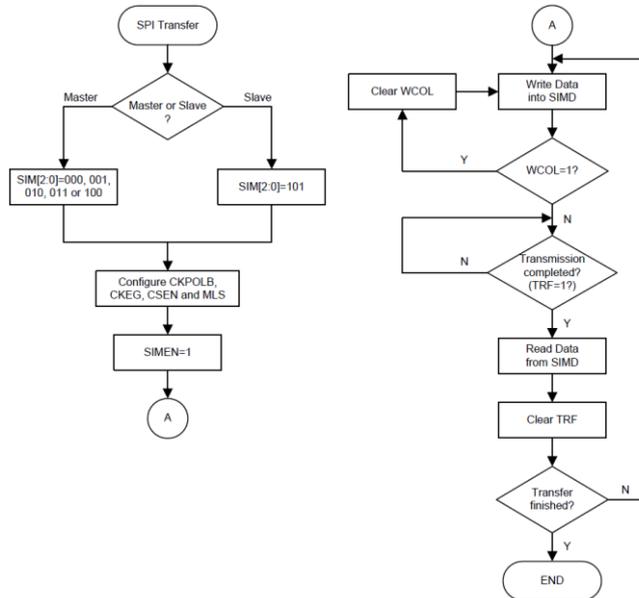


SPI 從機模式時序 - CKEG=0



Note: For SPI slave mode, if SIMEN=1 and CSEN=0, SPI is always enabled and ignores the SCS level.

SPI 從機模式時序 - CKEG=1



SPI 傳輸控制流程圖

SPI 使能 / 除能

設置 CSEN=1、SCS=0 將使能 SPI 匯流排，然後等待寫資料到 SIMD 寄存器(TXRX 暫存器)。單片機處於主機模式，資料寫入 SIMD 寄存器後，自動開始資料傳輸或接收操作。資料傳輸完成時，TRF 位將自動被置位。單片機處於從機模式，SCK 引腳上收到脈衝信號之後，會傳輸 TXRX 中的資料，或將 SDI 引腳上的資料移入。
當 SPI 匯流排除能時，通過設置相應控制位，SCK、SDI、SDO、SCS 可作為 I/O口或其它功能引腳使用。

SPI 操作步驟

四線制 SPI 介面可完成所有主 / 從模式通信工作。

在 SIMC2 寄存器中，CSEN 位元控制 SPI 介面的所有功能。設置此位元為高，SCS 信號線有效將使能 SPI 介面。設置此位元為低，SPI 介面將除能，SCS 信號線處於浮空狀態因此不能控制 SPI 介面。CSEN 位和 SIMC0 寄存器中的 SIMEN 位設置為高，使得 SDI 信號線處於浮空狀態且 SDO 信號線為高電平。主機模式中，如果 SCK 信號線為高還是低取決於 SIMC2 寄存器中的時鐘極性選擇位元 CKPOLB。從機模式中，SCK 信號線處於浮空狀態。如果 SIMEN 位設置為低，SPI 介面被除能，通過設置相應引腳共用控制位，SCS、SDI、SDO 和 SCK 可作為 I/O 口或其它功能引腳使用。主機模式中，當資料被寫入 SIMD 寄存器後，主機完成所有的資料傳輸初始化，並控制時鐘信號。從機模式中，由外部主機發出資料傳送 / 接收時鐘信號。下面介紹主從模式中資料傳輸步驟。

主機模式

- 步驟 1：設置 SIMC0 控制寄存器中的 UMD 和 SIM2~SIM0 位元，選擇 SPI 主機模式和時鐘源。
- 步驟 2：設置 CSEN 和 MLS 位元元，選擇高位或低位元資料優先傳送，這必須與從機設備一致。
- 步驟 3：設置 SIMC0 控制寄存器中的 SIMEN 位元，使能 SPI 介面功能。
- 步驟 4：對於寫操作：寫資料到 SIMD 寄存器，實際上此時資料會被存儲在 TXRX 暫存器中。再使用 SCK 和 SCS 信號線將資料輸出。跳至步驟 5。
對於讀操作：從 SDI 信號線移入的資料將被存儲在 TXRX 暫存器中，直到所有資料接收完畢，此時資料全部鎖存至 SIMD 寄存器。
- 步驟 5：檢測 WCOL 位，若此位為高，則發生資料衝突並跳回至步驟 4；若為低，則繼續執行下麵的步驟。
- 步驟 6：檢測 TRF 位元或等待 USIM SPI 串列匯流排中斷發生。
- 步驟 7：從 SIMD 寄存器中讀數據。
- 步驟 8：清除 TRF。
- 步驟 9：跳回至步驟 4。

從機模式

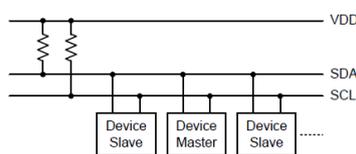
- 步驟 1：設置 SIMC0 控制寄存器中的 UMD 和 SIM2~SIM0 位，選擇 SPI 從機模式。
- 步驟 2：設置 CSEN 和 MLS 位元元，選擇高位或低位元資料優先傳送，這必須與主機設備一致。
- 步驟 3：設置 SIMC0 控制寄存器中的 SIMEN 位元，使能 SPI 介面功能。
- 步驟 4：對於寫操作：寫資料到 SIMD 寄存器，實際上此時資料會被存儲在 TXRX 暫存器中。等待主機時鐘 SCK 信號和 SCS 信號。跳至步驟 5。
對於讀操作：從 SDI 信號線移入的資料將被存儲在 TXRX 暫存器中，直到所有資料接收完畢，此時資料全部鎖存至 SIMD 寄存器。
- 步驟 5：檢測 WCOL 位，若此位為高，則發生資料衝突並跳回至步驟 4；若為低，則繼續執行下麵的步驟。
- 步驟 6：檢測 TRF 位元或等待 USIM SPI 串列匯流排中斷發生。
- 步驟 7：從 SIMD 寄存器中讀數據。
- 步驟 8：清除 TRF。
- 步驟 9：跳回至步驟 4。

錯誤偵測

SIMC2 寄存器中的 WCOL 位用於資料傳輸期間監測資料衝突的發生。此位由 SPI 序列介面設置為高，而由應用程式來清除為零。在資料傳輸期間如果寫資料到 SIMD，此位元被置高提示資料衝突發生，並阻止資料繼續被寫入。

I²C 介面

I²C 可以和感測器、EEPROM 記憶體等外部硬體介面進行通信。最初是由飛利浦公司研製，是適用於同步串列資料傳輸的雙線式低速序列介面。I²C 介面具有兩線通信，非常簡單的通信協定和在同一匯流排上和多個設備進行通信的能力的優點，使之在很多的應用場合中大受歡迎。

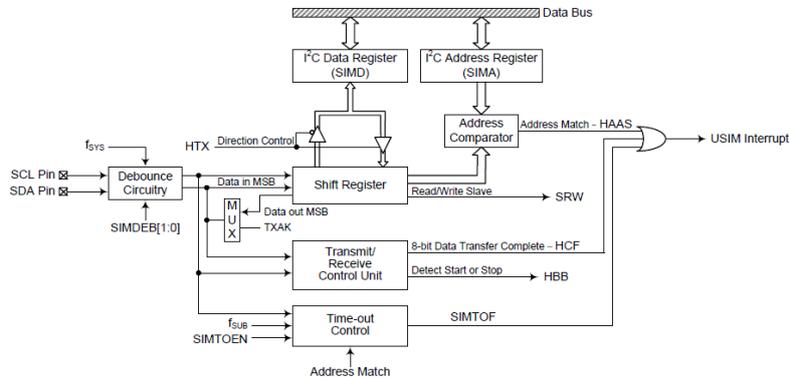


I²C 主從匯流排連接圖

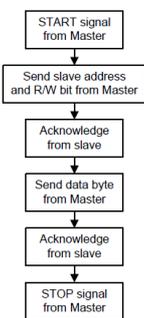
I²C 介面操作

I²C 序列介面是一個雙線的介面，有一條串列資料線 SDA 和一條串列時鐘線 SCL。由於可能有多個設備在同一條匯流排上相互連接，所以這些設備的輸出都是開漏型輸出。因此應在這些輸出口上都應加上拉電阻。應注意的是，I²C 匯流排上的每個設備都沒有選擇線，但分別與唯一的位址一一對應，用於 I²C 通信。如果有兩個設備通過雙向的 I²C 匯流排進行通信，那麼就存在一個主機和一個從機。主機和從機都可以用於發送和接收資料，但只有主機才可以控制匯流排動作。

那些處於從機模式的設備，要在 I²C 匯流排上傳輸資料只有兩種方式，一是從機發送模式，二是從機接收模式。即使 I²C 設備被啟動，與 SCL/SDA 引腳共用的 I/O 口上拉電阻控制功能仍有效，其上拉電阻功能由相應的上拉電阻控制寄存器控制。



I²C 方框圖



I²C 介面操作

SIMDEB1 和 SIMDEB0 位元決定 I²C 介面的去抖時間。這個功能可以使用內部時鐘在外部時鐘上增加一個去抖間隔，會減小時鐘線上毛刺發生的可能性，以避免單片機發生誤動作。如果選擇了這個功能，去抖時間可以選擇 2 個或 4 個系統時鐘。為了達到需要的 I²C 資料傳輸速度，系統時鐘 f_{sys} 和 I²C 去抖時間之間存在一定的關係。I²C 標準模式或者快速模式下，使用者需注意所選的系統時鐘頻率與標準匹配去抖時間的設置，其具體關係如下表所示。

I ² C 去抖時間選擇	I ² C 標準模式 (100kHz)	I ² C 快速模式 (400kHz)
2 個系統時鐘去抖時間	f _{sys} > 4MHz	f _{sys} > 10MHz
4 個系統時鐘去抖時間	f _{sys} > 8MHz	—

I²C 最小 f_{sys} 頻率要求 I²C 寄存器 I²C 匯流排有三個控制寄存器 SIMC0、SIMC1 和 SIMTOC，及一個從機位址寄存器 SIMA 和一個資料寄存器 SIMD。應注意的是，SIMC1、SIMD、SIMA 和 SIMTOC 寄存器以及它們的 POR 值只有當 I²C 模式被正確配置在 SIMC0 寄存器中的 UMD 和 SIM2~SIM0 位時才可用。

寄存器名稱	Bit							
	7	6	5	4	3	2	1	0
SIMC0	SIM2	SIM1	SIM0	UMD	SIMDEB1	SIMDEB0	SIMEN	SIMICF
SIMC1	HCF	HAAS	HBB	HTX	TXAK	SRW	IAMWU	RXAK
SIMD	D7	D6	D5	D4	D3	D2	D1	D0
SIMA	SIMA6	SIMA5	SIMA4	SIMA3	SIMA2	SIMA1	SIMA0	D0
SIMTOC	SIMTOEN	SIMTOF	SIMTOS5	SIMTOS4	SIMTOS3	SIMTOS2	SIMTOS1	SIMTOS0

I²C 寄存器列表

I²C 資料寄存器

SIMD 用於存儲發送和接收的資料。這個寄存器由 SPI 和 I²C 功能所共用。在單片機將資料寫入到 I²C 匯流排之前，要傳輸的資料應先存在 SIMD 中。I²C 匯流排接收到資料之後，單片機就可以從 SIMD 資料寄存器中讀取。所有通過 I²C 傳輸或接收的資料都必須通過 SIMD 實現。

● SIMD 寄存器

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	x	x	x	x	x	x	x	x

“x”：未知

Bit 7~0 D7~D0：USIM SPI/I²C 資料寄存器 bit 7 ~ bit 0

I²C 位址寄存器

SIMA 寄存器也在 SPI 介面功能中使用，但其名稱改為 SIMC2。SIMA 寄存器用於存放 7 位從機位址，寄存器 SIMA 中的 bit 7 ~ bit 1 是單片機的從機地址，bit 0 未定義。

如果接至 I²C 的主機發送出的位址和寄存器 SIMA 中存儲的位址相符，那麼就選中了這個從機。

● SIMA 寄存器

Bit	7	6	5	4	3	2	1	0
Name	SIMA6	SIMA5	SIMA4	SIMA3	SIMA2	SIMA1	SIMA0	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~1 SIMA6~SIMA0：I²C 從機地址位 SIMA6~SIMA0 是 I²C 從機地址 bit 6 ~ bit 0

Bit 0 D0：保留位元，此位元可通過軟體程式進行讀寫。

I²C 控制寄存器

單片機中有三個控制 I²C 介面功能的寄存器，SIMC0、SIMC1 和 SIMTOC。寄存器 SIMC0 用於控制使能 / 除能功能和設置資料傳輸的時鐘頻率。寄存器 SIMC1 包括多個用於表明 I²C 傳輸狀態的相關標誌位元。SIMTOC 寄存器用於控制 I²C 匯流排超時功能，在 I²C 超時控制章節有描述。

● SIMC0 寄存器

Bit	7	6	5	4	3	2	1	0
Name	SIM2	SIM1	SIM0	UMD	SIMDEB1	SIMDEB0	SIMEN	SIMICF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	1	1	1	0	0	0	0	0

Bit 7~5 SIM2~SIM0：USIM SPI/I²C 工作模式控制位元

000：SPI 主機模式；SPI 時鐘為 fSYS/4

001：SPI 主機模式；SPI 時鐘為 fSYS/16

010：SPI 主機模式；SPI 時鐘為 fSYS/64

011：SPI 主機模式；SPI 時鐘為 fSUB

100：SPI 主機模式；SPI 時鐘為 STM CCRP 匹配頻率 / 2

101：SPI 從機模式

110：I²C 從機模式

111：未使用模式

當 UMD 位清除為 0 時，這些位設置了 USIM 功能的 SPI 或 I²C，用於選擇 SPI 的主從模式和 SPI 的主機時鐘頻率及 I²C 或 SPI 功能。SPI 時鐘源可來自於系統時鐘也可以選擇來自 STM 和 fSUB。若選擇的是作為 SPI 從機，則其時鐘源從外部主機而得。

Bit 4 UMD：UART 模式選擇位元

0：SPI 或 I²C 模式 1：UART 模式

此位元用於選擇 UART 模式。當此位清除為零，實際的 SPI 或 I²C 模式可以通過SIM2~SIM0 位選擇。注意，對於 SPI 或 I²C 模式 UMD 必須設置為低。

Bit 3~2 **SIMDEB1~SIMDEB0** : I²C 去抖時間選擇位
00 : 無去抖時間
01 : 2 個系統時鐘去抖時間
1x : 4 個系統時鐘去抖時間

當設置UMD 位為 “0” 和 SIM2~SIM0 位為 “110” 將 SIM 設置為I²C 介面功能時，這兩個位用於選擇 I²C 去抖時間。

Bit 1 **SIMEN** : USIM SPI/I²C 使能控制位
0 : 除能
1 : 使能

此位元為 USIM SPI/I²C 介面的開 / 關控制位。此位為 “0” 時，USIM SPI/I²C 介面除能，SDI、SDO、SCK 和 SCS 或 SDA 和 SCL 腳將失去 SPI 或 I²C 功能，USIM 工作電流減小到最小值。此位為 “1” 時，USIM SPI/I²C 介面使能。若USIM 經由 UMD 和 SIM2~SIM0 位設置為工作在 SPI 介面，當 SIMEN 位元由低到高轉變時，SPI 控制寄存器中的設置不會發生變化，其首先應在應用程式中初始化。若 SIM 經由 UMD 和 SIM2~SIM0 位設置為工作在 I²C 介面，當 SIMEN 位元由低到高轉變時，I²C 控制寄存器中的設置，如HTX 和 TXAK，將不會發生變化，其首先應在應用程式中初始化，此時相關 I²C 標誌，如 HCF、HAAS、HBB、

SRW 和 RXAK，將被設置為其預設狀態。

Bit 0 **SIMICF** : USIM SPI 未完成標誌位元

此位元元僅當 SIM 配置在 SPI 從機模式時有效。請參考 SPI 寄存器部分。

● **SIMC1 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	HCF	HAAS	HBB	HTX	TXAK	SRW	IAMWU	RXAK
R/W	R	R	R	R/W	R/W	R	R/W	R
POR	1	0	0	0	0	0	0	1

Bit 7 **HCF** : I²C 匯流排資料傳輸結束標誌位元
0 : 資料正在被傳輸
1 : 8 位資料傳輸完成

資料正在傳輸時該位元為低。當 8 位資料傳輸完成時，此位為高並產生一個中斷。

Bit 6 **HAAS** : I²C 匯流排位址匹配標誌位元
0 : 地址不匹配
1 : 地址匹配

此標誌位元用於決定從機位址是否與主機發送位址相同。若地址匹配此位為高，否則此位為低。Bit 5 **HBB** : I²C 匯流排忙標誌位元

0 : I²C 匯流排閑
1 : I²C 匯流排忙

當檢測到 START 信號時 I²C 忙，此位變為高。當檢測到 STOP 信號時 I²C 匯流排停止，該位元變為低。

Bit 4 **HTX** : I²C 從機處於發送或接收模式選擇位元元
0 : 從機處於接收模式
1 : 從機處於發送模式

Bit 3 **TXAK** : I²C 匯流排發送確認標誌位元
0 : 從機發送確認標誌
1 : 從機沒有發送確認標誌

單片機接收 8 位元資料之後會將該位元在第九個時鐘傳到匯流排上。如果單片機想要接收更多的資料，則應在接收資料之前將此位元設置為 “0”。

Bit 2 **SRW** : I²C 從機讀 / 寫標誌位元
0 : 從機應處於接收模式
1 : 從機應處於發送模式

SRW 位是 I²C 從機讀寫標誌位元。決定主機是否希望發送或接收來自 I²C 匯流排的資料。當傳輸位址和從機的位址相同時，HAAS 位會被設置為高，從機將檢測SRW 位元元來決定進入發送模式還是接收模式。如果 SRW 位元為高時，主機會請求從匯流排上讀數據，此時從機處於發送模式。當 SRW 位元為低時，主機往匯流排上寫資料，從機處於接收模式以讀取該資料。

Bit 1 **IAMWU** : I²C 位址匹配喚醒控制位

0：除能

1：使能

此位應設置為 “1” 使能 I²C 位址匹配以使系統從休眠或空閒模式中喚醒。若進入休眠或空閒模式前 IAMWU 已經置高以使能 I²C 地址匹配喚醒功能，在系統喚醒後須應用程式清除此位元以確保單片機正確地運行。

Bit 0 RXAK：I²C 匯流排接收確認標誌位元

0：從機接收到確認標誌

1：從機沒有接收到確認標誌

RXAK 位元是接收確認標誌位元。如果 RXAK 位為 “0” 即 8 位資料傳輸之後，從機在第九個時鐘有接受到一個正確的確認位。如果從機處於發送狀態，從機會檢查 RXAK 位元來判斷主機是否願意繼續接收下一個位元組。因此直到 RXAK 為 “1” 時，從機傳輸方停止發送資料。這時，從機傳輸方將釋放 SDA 線，主機發出停止信號釋放 I²C 匯流排。

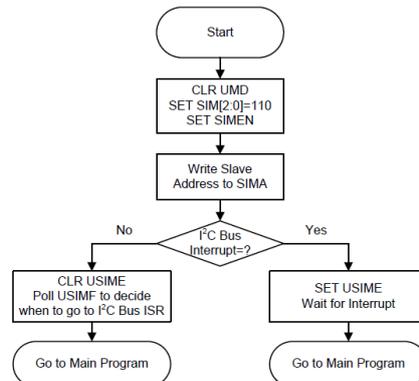
I²C 匯流排通信

I²C 匯流排上的通信需要四步完成，一個起始信號，一個從機位址發送，一個資料傳輸，還有一個停止信號。當起始信號被寫入 I²C 匯流排時，匯流排上的所有從機都會接收到這個起始信號並且被通知匯流排上會即將有資料到達。資料的前 7 位是從機位元址，高位在前，低位元元在後。如果發出的位址和從機位址匹配，SIMC1 寄存器的 HAAS 位會被置位，同時產生 USIM 中斷。進入中斷服務程式後，系統要檢測 HAAS 位元和 SIMTOF 位，以判斷 I²C 匯流排中斷是來自從機位址匹配，還是來自 8 位元資料傳遞完畢，或是來自 I²C 匯流排超時。在資料傳遞中，注意的是，在 7 位從機位址被發送後，接下來的一位，即第 8 位，是讀 / 寫控制位，該位的值會反映到 SRW 位中。從機通過檢測 SRW 位以確定主控制器是要進入發送模式還是接收模式。在 I²C 匯流排開始傳送資料前，需要先初始化 I²C 匯流排，初始化 I²C 匯流排步驟如下：

步驟 1：設置 SIMC0 寄存器中 UMD 位為 “0”，SIM2~SIM0 位為 “110” 和 SIMEN 位為 “1”，以使能 I²C 匯流排。

步驟 2：向 I²C 匯流排位址寄存器 SIMA 寫入從機位址。

步驟 3：設置中斷控制寄存器中的 USIME 中斷使能位，以使能 USIM 中斷。



I²C 匯流排初始化流程圖

I²C 匯流排起始信號

起始信號只能由連接 I²C 匯流排的主機產生，而不是由從機產生。匯流排上的所有從機都可以偵測到起始信號。如果有從機偵測到起始信號，則表明 I²C 匯流排處於忙碌狀態，並會置位元 HBB。起始信號是指在 SCL 為高電平時，SDA 線上發生從高到低的電平變化。

I²C 從機地址

進行資料傳輸的從機。所有在 I²C 匯流排上的從機接收到 7 位元位址資料後，都會將其與各自內部的位址進行比較。如果從機從主機上接收到的位址與自身內部的位址相匹配，則會產生一個 USIM I²C 匯流排中斷信號。位址位接下來的一位元為讀 / 寫狀態位元（即第 8 位），將被保存到 SIMC1 寄存器的 SRW 位，隨後發出一個低電平應答信號（即第 9 位）。當單片機從機的位址匹配時，會將狀態標誌位元 HAAS 置位元。

USIM I²C 匯流排有三個中斷源，當程式運行至中斷服務副程式時，通過檢測 HAAS 位和 SIMTOF 位，以判斷 I²C 匯流排中斷是來自從機位址匹配，還是來自 8 位元資料傳遞完畢，或是來自 I²C 匯流排超時。當是從機位址匹配發生中斷時，則從機或是用於發送模式並將資料寫進 SIMD 寄存器，或是用於接收模式並從

SIMD 寄存器中讀取空值以釋放 SCL 線。

I²C 匯流排讀 / 寫信號

SIMC1 寄存器的 SRW 位元用來表示主機是要從 I²C 匯流排上讀取資料還是要將資料寫到 I²C 匯流排上。從機則通過檢測該位以確定自己是作為發送方還是接收方。當 SRW 置 “1” ，表示主機要從 I²C 匯流排上讀取資料，從機則作為發送方，將資料寫到 I²C 匯流排；當 SRW 清 “0” ，表示主機要寫資料到 I²C 匯流排上，從機則做為接收方，從 I²C 匯流排上讀取資料。

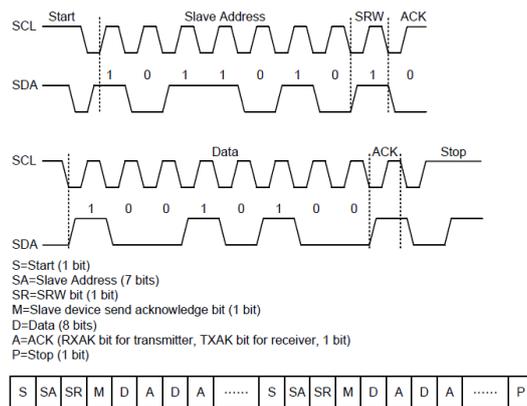
I²C 匯流排從機位址確認信號

主機發送呼叫位址後，當 I²C 匯流排上的任何從機內部位址與其匹配時，會發送一個應答信號。此應答信號會通知主機有從機已經接收到了呼叫位址。如果主機沒有收到應答信號，則主機必須發送停止(STOP) 信號以結束通信。當 HAAS 為高時，表示從機接收到的位址與自己內部位址匹配，則從機需檢查SRW 位，以確定自己是作為發送方還是作為接收方。如果 SRW 位為高，從機須設置成發送方，這樣會置位 SIMC1 寄存器的 HTX 位。如果 SRW 位為低，從機須設置成接收方，這樣會清零 SIMC1 寄存器的 HTX 位。

I²C 匯流排資料和確認信號

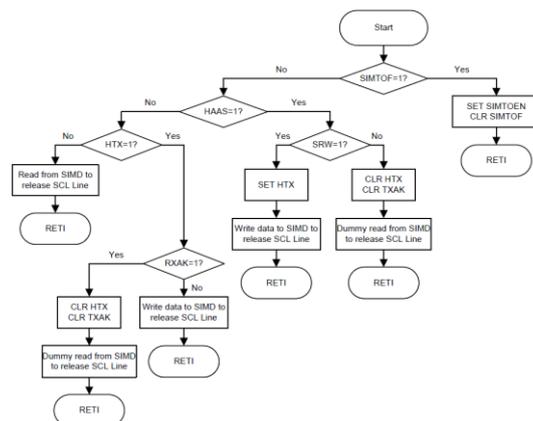
在從機確認接收到從機位址後，會進行 8 位寬度的資料傳輸。這個資料傳輸順序是的高位在前，低位元在後。接收方在接收到 8 位元資料後必須發出一個應答信號 (“0”) 以繼續接收下一個資料。如果發送方沒接收到應答信號，發送方將釋放 SDA 線，同時，主機將發出 STOP 信號以釋放 I²C 匯流排。所傳送的資料存儲在 SIMD 寄存器中。如果設置成發送方，從機必須先將欲傳輸的資料寫到 SIMD 寄存器中；如果設置成接收方，從機必須從 SIMD 寄存器讀取資料。

當從機接收器想要繼續接收下一個資料時，必須在第 9 個時鐘發出應答信號(TXAK)。被設為發送方的從機將檢測寄存器 SIMC1 中的 RXAK 位以判斷是否傳輸下一個位元組的資料，如果單片機不傳輸下一個位元組，那麼它將釋放 SDA 線並等待接收主機的停止信號。



I²C 通信時序圖

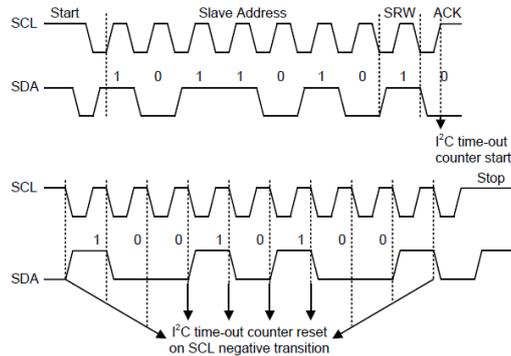
注：當從機位址匹配時，單片機必須對單設置為發送模式還是接收模式。若設置為發送模式，需寫資料至 SIMD 寄存器；若設置為接收模式，需立即從 SIMD 寄存器中虛讀數據以釋放 SCL 線。



I²C 匯流排 ISR 流程圖

I²C 超時控制

超時功能可減少 I²C 接收錯誤的時鐘源而引起的鎖死問題。如果連接到 I²C 匯流排的時鐘源經過一段時間還未接收到，則在一定的超時週期後，I²C 電路和寄存器將重定。超時計數器在 I²C 匯流排 “START” 和 “位址匹配” 條件下開始計數，且在 SCL 下降沿清零。在下一個 SCL 下降沿到來之前，如果超時時間大於 SIMTOC 寄存器指定的超時週期，則超時發生。I²C “STOP” 條件發生時超時功能終止。



I²C 超时时序圖

當 I²C 超時計數器溢出時，計數器將停止計數，SIMTOEN 位被清零，且 SIMTOF 位被置高以表明超時計數器中斷發生。超時計數器中斷使用的也是 USIM 中斷向量。當 I²C 超時發生時，I²C 內部電路會被重定，寄存器也將發生如下重定情況。

寄存器	I ² C 超時發生後
SIMD, SIMA, SIMC0	保持不變
SIMC1	復位至 POR

超時發生後的 I²C 寄存器 SIMTOF 標誌位元由應用程式清零。共有 64 個超時週期，可通過 SIMTOC 寄存器的 SIMTOS 位進行選擇。超時週期可通過公式計算： $((1\sim64)\times(32/f_{SUB}))$ 。由此可得超時週期範圍為 1ms~64ms。

● SIMTOC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	SIMTOEN	SIMTOF	SIMTOS5	SIMTOS4	SIMTOS3	SIMTOS2	SIMTOS1	SIMTOS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **SIMTOEN** : USIM I²C 超時功能控制位

- 0 : 除能
- 1 : 使能

Bit 6 **SIMTOF** : USIM I²C 超時標誌位元

- 0 : 未發生
- 1 : 發生

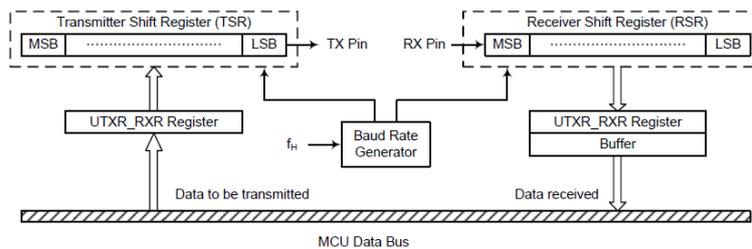
當超時發生時，此位設置為 1，並且只能通過應用程式將其清除為零。Bit 5~0 **SIMTOS5~SIMTOS0** : USIM I²C 超時時間選擇位 I²C 超時時鐘源是 $f_{SUB}/32$ 。I²C 超時時間計算方法： $(SIMTOS[5:0]+1)\times(32/f_{SUB})$ 。

UART 介面

單片機具有一個全雙工的非同步串列通信介面 – UART，可以很方便的與其它具有串列介面的晶片通信。UART 具有許多功能特性，發送或接收串列資料時，將資料組成一個 8 位或 9 位元的資料塊，連同資料特徵位元一併傳輸。具有檢測資料覆蓋或幀錯誤等功能。UART 功能與 SPI 和 I²C 共用一個內部中斷向量，當接收到資料或資料發送結束，觸發 UART 中斷。內置的 UART 功能包含以下特性：

- 全雙工通用非同步接收器 / 發送器
- 8 位或 9 位傳輸格式
- 奇數同位檢查、偶校驗或無校驗
- 1 位或 2 位停止位

- 8 位元預分頻的串列傳輸速率發生器
- 奇偶、幀、雜訊和溢出檢測
- 支持位址匹配中斷 (最後一位 =1)
- 獨立的發送和接收使能
- 2-byte FIFO 接收緩衝器
- RX 引腳喚醒功能
- 發送和接收中斷
- 中斷可由下列條件觸發：
- 發送器為空
- 發送器空閒
- 接收完成
- 接收器溢出
- 位元址模式檢測



UART 資料傳輸方框圖

UART 外部引腳

內部 UART 有兩個外部引腳 TX 和 RX，可與外部序列介面進行通信。TX 和 RX 分別為 UART 發送腳和接收腳，與 I/O 口或其它功能共用引腳。在使能 UART 功能之前，需先通過 IFS 寄存器選擇 TX 和 RX 引腳。當 UMD 位、UREN 位、UTXEN 和 URXEN 位置高時，將自動設置這些引腳作為 TX 輸出和 RX 輸入，並且除能 TX 和 RX 引腳上的上拉電阻功能。當 UMD、UREN、UTXEN 或 URXEN 位清零除能 TX 或 RX 引腳功能後，TX 或 RX 引腳將處於浮空狀態。這時 TX 或 RX 引腳是否連接內部上拉電阻可由相應的 I/O 上拉電阻控制位決定。

UART 資料傳輸方案

上圖顯示了 UART 介面的整體資料傳輸結構。需要發送的資料首先通過應用程式寫入 UTXR_RXR 寄存器，接著此資料被傳輸到發送移位暫存器 TSR 中，然後在串列傳輸速率發生器的控制下將 TSR 寄存器中資料一位元位地移到 TX 引腳上，低位在前。UTXR_RXR 寄存器被映射到單片機的資料記憶體中，而發送移位暫存器沒有實際位址，所以發送移位暫存器不可直接操作。

資料在串列傳輸速率發生器的控制下，低位元元在前高位在後，從外部引腳 RX 進入接收移位暫存器 RSR。當資料接收完成，資料從接收移位暫存器移入可被使用者程式操作的 UTXR-RXR 寄存器中。UTXR-RXR 寄存器被映射到單片機資料記憶體中，而接收移位暫存器沒有實際位址，所以接收移位暫存器不可直接操作。需要注意的是，發送和接收都是共用同一個位址的資料寄存器，即 UTXR_RXR 寄存器。

UART 狀態和控制寄存器

與 UART 功能相關的有六個寄存器，用以選擇 UART 模式的 SIMC0 寄存器中的 UMD 位。控制 UART 模組整體功能的 UUSR、UUCR1 和 UUCR2 寄存器，控制串列傳輸速率的 UBRG 寄存器，管理發送和接收資料的資料寄存器 UTXR_RXR。注意，只有通過 SIMC0 寄存器中設置 UMD 位為“1”且選擇 UART 模式時，UART 相關寄存器和它們的 POR 值才有效。

寄存器名稱	位							
	7	6	5	4	3	2	1	0
SIMC0	SIM2	SIM1	SIM0	UMD	SIMDEB1	SIMDEB0	SIMEN	SIMICF
UUSR	UPERR	UNF	UFERR	UOERR	URIDLE	URXIF	UTIDLE	UTXIF
UUCR1	UREN	UBNO	UPREN	UPRT	USTOPS	UTXBRK	URX8	UTX8
UUCR2	UTXEN	URXEN	UBRGH	UADDEN	UWAKE	URIE	UTIIE	UTEIE
UTXR_RXR	UTXR7	UTXR6	UTXR5	UTXR4	UTXR3	UTXR2	UTXR1	UTXR0
UBRG	UBRG7	UBRG6	UBRG5	UBRG4	UBRG3	UBRG2	UBRG1	UBRG0

UART 寄存器列表

● SIMCO 寄存器

Bit	7	6	5	4	3	2	1	0
Name	SIM2	SIM1	SIM0	UMD	SIMDEB1	SIMDEB0	SIMEN	SIMICF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	1	1	1	0	0	0	0	0

Bit 7~5 **SIM2~SIM0** : USIM SPI/I²C 工作模式控制位

當 UMD 位清除為零時，這幾位用於設置 USIM 功能中 SPI 或 I²C 的工作模式。請參考 SPI 或 I²C 寄存器部分。

Bit 4 **UMD** : UART 模式選擇位元

0 : SPI 或 I²C 模式

1 : UART 模式

此位元用於選擇 UART 模式。當此位清除為零，實際 SPI 或 I²C 模式可以使用 SIM2~SIM0 位選擇。注意，在 SPI 或 I²C 模式中 UMD 位元必須置零。

Bit 3~2 **SIMDEB1~SIMDEB0** : I²C 去抖時間選擇位請參考 I²C 寄存器部分。

Bit 1 **SIMEN** : USIM SPI/I²C 使能控制位

這些位元只有在 USIM 配置在 SPI 或 I²C 模式下工作且 UMD 位元設為低時才有效。請參考 SPI 或 I²C 寄存器部分。

Bit 0 **SIMICF** : USIM SPI 未完成標誌位元請參考 SPI 寄存器部分

● UUSR 寄存器

寄存器 UUSR 是 UART 的狀態寄存器，可以通過程式讀取。所有 UUSR 位是唯讀的。詳細解釋如下：

Bit	7	6	5	4	3	2	1	0
Name	PERR	NF	FERR	OERR	RIDLE	RXIF	TIDLE	TXIF
R/W	R	R	R	R	R	R	R	R
POR	0	0	0	0	1	0	1	1

Bit 7 **PERR** : 同位出錯標誌位元

0 : 同位正確

1 : 同位出錯

UPERR 是同位出錯標誌位元。若 UPERR=0，同位正確；若 UPERR=1，接收到的資料同位出錯。只有使能了同位此位才有效。可使用軟體清除該標誌位元，即先讀取 UUSR 寄存器再讀 UTXR_RXR 寄存器來清除此位。

Bit 6 **UNF** : 雜訊干擾標誌位元

0 : 沒有受到雜訊干擾

1 : 受到雜訊干擾

UNF 是雜訊干擾標誌位元。若 UNF=0，沒有受到雜訊干擾；若 UNF=1，UART 接收資料時受到雜訊干擾。它與 URXIF 在同週期內置位，但不會與溢出標誌位元同時置位元。可使用軟體清除該標誌位元，即先讀取 UUSR 寄存器再讀 UTXR_RXR 寄存器將清除此標誌位元。

Bit 5 **UFERR** : 幀錯誤標誌位元

0 : 無幀錯誤發生

1 : 有幀錯誤發生

UFERR 是幀錯誤標誌位元。若 UFERR=0，沒有幀錯誤發生；若 UFERR=1，當前的資料發生了幀錯誤。可使用軟體清除該標誌位元，即先讀取 UUSR 寄存器再讀 UTXR_RXR 寄存器來清除此位。

Bit 4 **UOERR** : 溢出錯誤標誌位元

0 : 無溢出錯誤發生

1 : 有溢出錯誤發生

UOERR 是溢出錯誤標誌位元，表示接收緩衝器是否溢出。若 UOERR=0，沒有溢出錯誤；若 UOERR=1，發生了溢出錯誤，它將影響下一組資料的接收。可通過軟體清除該標誌位元，即先讀取 UUSR 寄存器再讀 UTXR_RXR 寄存器將清除此標誌位元。

Bit 3 **URIDLE** : 接收狀態標誌位元

0 : 正在接收資料

1 : 接收器空閒

URIDLE 是接收狀態標誌位元。若 URIDLE=0，正在接收資料；若 URIDLE=1，接收器空閒。在接收到停止位元和下一個資料的起始位元之間，URIDLE 被置位，表明 UART 空閒，RX 腳處於邏輯高狀態。

Bit 2 **URXIF** : 接收寄存器狀態標誌位元

0 : UTXR_RXR 寄存器為空

1 : UTXR_RXR 寄存器含有有效資料

URXIF 是接收寄存器狀態標誌位元。當 URXIF=0，UTXR_RXR 寄存器為空；當 URXIF=1，UTXR_RXR 寄存器接收到新資料。當數據從移位暫存器載入到UTXR_RXR 寄存器中，如果 UCR2 寄存器中的 RIE=1，則會觸發中斷。當接收資料時檢測到一個或多個錯誤時，相應的標誌位元UNF、UFERR 或 UPERR 會在同一週期內置位。讀取 UUSR 寄存器再讀 UTXR_RXR 寄存器，如果 UTXR_RXR 寄存器中沒有新的資料，那麼將清除 URXIF 標誌。

Bit 1 **UTIDLE** : 資料發送完成標誌位元

- 0 : 資料傳輸中
- 1 : 無數據傳輸

UTIDLE 是資料發送完成標誌位元。若 UTIDLE=0，資料傳輸中。當 UTXIF=1 且資料發送完畢或者暫停字被發送時，UTIDLE 置位。UTIDLE=1，TX 引腳空間且處於邏輯高狀態。讀取 UUSR 寄存器再寫 UTXR_RXR 寄存器將清除 UTIDLE 位。資料字元或暫停字就緒時，不會產生該標誌位元。

Bit 0 **UTXIF** : 發送資料寄存器 UTXR_RXR 狀態位元

- 0 : 資料還沒有從緩衝器載入到移位暫存器中
- 1 : 資料已從緩衝器載入到移位暫存器中 (UTXR_RXR 資料寄存器為空)

UTXIF 是發送資料寄存器為空標誌位元。若 UTXIF=0，資料還沒有從緩衝器載入到移位暫存器中；若 UTXIF=1，資料已從緩衝器中載入到移位暫存器中。讀取UUSR 寄存器再寫 UTXR_RXR 寄存器將清除 UTXIF。當 UTXEN 被置位，由於發送緩衝器未滿，UTXIF 也會被置位。

● UUCR1 寄存器

UUCR1 和 UUCR2 是 UART 的兩個控制寄存器，用來定義各種 UART 功能，例如 UART 整體的開 / 關控制、同位控制和傳輸資料的長度等等。詳細解釋如下：

Bit	7	6	5	4	3	2	1	0
Name	UREN	UBNO	UPREN	UPRT	USTOPS	UTXBRK	URX8	UTX8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	W
POR	0	0	0	0	0	0	x	0

“x” : 未知

Bit 7 **UREN** : UART 功能使能位

- 0 : UART 除能，TX 和 RX 腳為浮空狀態
- 1 : UART 使能，TX 和 RX 腳作為 UART 功能引腳

此位為 UART 的使能位。UREN=0，UART 除能，RX 和 TX 設為浮空狀態；UREN=1，若 UMD 被設置，UART 使能，TX 和 RX 將分別由 UTXEN 和URXEN 控制。當 UART 被除能將清除緩衝器，所有緩衝器中的資料將被忽略，另外串列傳輸速率計數器、錯誤和狀態標誌位元被重定，UTXEN、URXEN、UTXBRK、URXIF、UOERR、UFERR、UPERR 和 UNF 清零而 UTIDLE、UTXIF 和 URIDLE 置位，UUCR1、UUCR2 和 UBRG 寄存器中的其它位保持不變。若 UART 工作時UARTEN 清零，所有發送和接收將停止，模組也將重定成上述狀態。當 UART再次使能時，它將在上次配置下重新工作。

Bit 6 **UBNO** : 發送資料位元數選擇位元

- 0 : 8-bit 傳輸資料
- 1 : 9-bit 傳輸資料

UBNO 是發送資料位元數選擇位元。UBNO=1，傳輸資料為 9 位元；UBNO=0，傳輸資料為 8 位元。若選擇了 9 位資料傳輸格式，URX8 和 UTX8 將分別存儲接收和發送資料的第 9 位。

Bit 5 **UPREN** : 同位使能位

- 0 : 同位除能
- 1 : 同位使能

此位為同位使能位。UPREN=1，使能同位；UPREN=0，除能同位。

Bit 4 **UPRT** : 同位選擇位

- 0 : 偶校驗
- 1 : 奇數同位檢查

同位選擇位。UPRT=1，奇數同位檢查；UPRT=0，偶校驗。

Bit 3 **USTOPS** : 停止位的長度選擇位

- 0 : 有一位停止位
- 1 : 有兩位停止位

此位用來設置停止位的長度。USTOP=1，有兩位停止位；USTOP=0，只有一位停止位。

Bit 2 **UTXBRK** : 暫停字發送控制位

- 0 : 沒有暫停字要發送
- 1 : 發送暫停字

UTXBRK 是暫停字發送控制位。UTXBRK=0，沒有暫停字要發送，TX 引腳正常操作；UTXBRK=1，將會發送暫停字，發送器將發送邏輯 “0”。若UTXBRK 為高，緩衝器中資料發送完畢後，發送器將至少保持 13 位寬的低電平直至 UTXBRK 復位。

Bit 1 **URX8** : 接收 9-bit 資料傳輸格式中的第 9 位 (唯讀)

此位元只有在傳輸資料為 9 位元的格式中有效，用來存儲接收資料的第 9 位。UBNO是用來控制傳輸位數是 8 位還是 9 位。

Bit 0 **UTX8** : 發送 9-bit 資料傳輸格式中的第 9 位 (只寫)

此位元只有在傳輸資料為 9 位元的格式中有效，用來存儲發送資料的第 9 位。UBNO是用來控制傳輸位數是 8 位還是 9 位。

● **UUCR2 寄存器**

UUCR2 是 UART 的第二個控制寄存器，它的主要功能是控制發送器、接收器以及各種 USIM UART 中斷源的使能或除能。它也可用來控制串列傳輸速率，使能接收喚醒和位址偵測。詳細解釋如下：

Bit	7	6	5	4	3	2	1	0
Name	UTXEN	URXEN	UBRGH	UADDEN	UWAKE	URIE	UTIE	UTEIE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **UTXEN** : UART 發送使能位

0 : UART 發送除能

1 : UART 發送使能

此位為發送使能位。UTXEN=0，發送將被除能，發送器立刻停止工作。另外緩衝器將被復位，此時 TX 引腳將設為浮空狀態。若 UTXEN=1 且 UMD 和 UREN=1，則發送將被使能，TX 引腳將由 UART 來控制。在資料傳輸時清除 UTXEN 將中止資料發送且重定發送器，此時 TX 引腳將設為浮空狀態。

Bit 6 **URXEN** : UART 接收使能位

0 : UART 接收除能

1 : UART 接收使能

此位為接收使能位。URXEN=0，接收將被除能，接收器立刻停止工作。另外緩衝器將被復位，此時 RX 引腳將設為浮空狀態。若 URXEN=1 且 UMD 和 UREN=1，則接收將被使能，RX 引腳將由 UART 來控制。在資料傳輸時清除 URXEN 將中止資料接收且重定接收器，此時 RX 引腳將設為浮空狀態。

Bit 5 **UBRGH** : 串列傳輸速率發生器高低速選擇位元

0 : 低速串列傳輸速率

1 : 高速串列傳輸速率

此位元為串列傳輸速率發生器高低速選擇位元，它和 UBRG 寄存器一起控制 UART 的串列傳輸速率。UBRGH=1，為高速模式；UBRGH=0，為低速模式。

Bit 4 **UADDEN** : 地址檢測使能位

0 : 地址檢測除能

1 : 地址檢測使能

此位為位址檢測使能控制位。UADDEN=1，位址檢測使能，此時資料的第 8 位(UBON=0) 或第 9 位 (UBON=1) 為高，那麼接到的是位址而非資料。若相應的中斷使能且接收到的值最高位為 1，那麼插斷要求標誌將會被置位元，若最高位為0，那麼將不會產生中斷且收到的資料也會被忽略。

Bit 3 **UWAKE** : RX 腳下降沿喚醒 UART 功能使能位

0 : RX 腳下降沿喚醒 UART 功能除能

1 : RX 腳下降沿喚醒 UART 功能使能

此位用於控制 RX 引腳下降沿時是否喚醒 UART 功能。此位元僅當 UART 時鐘源fH 關閉時有效。若 UART 時鐘源fH 還開啟，則無RX 引腳喚醒 UART 功能無效。若此位置高且 UART 時鐘 fH 關閉，當 RX 引腳發生下降沿時會產生 UART 喚醒請求。若相應的中斷使能，將產生 RX 引腳喚醒 UART 的中斷，以告知單片機使其通過應用程式開啟 UART 時鐘源 fH，從而喚醒 UART 功能。否則，若此位為低，即使 RX 引腳發生下降沿也無法恢復 UART 功能。

Bit 2 **URIE** : 接收中斷使能位

0 : 接收中斷除能

1 : 接收中斷使能

此位為接收中斷使能或除能位。若 URIE=1，當 UOERR 或 URXIF 置位時，UART 的插斷要求標誌置位元；若 URIE=0，USIM 插斷要求標誌 USIMF 不受 UOERR 和 URXIF 影響。

Bit 1 **UTIE** : 發送器空間中斷使能位

0 : 發送器空間中斷除能

1 : 發送器空間中斷使能

此位為發送器空間中斷的使能或除能位。若 UTIE=1，當 UTIDLE 置位時，USIM 的插斷要求標誌 USIMF 置位元；若 UTIE=0，USIM 插斷要求標誌 USIMF 不受 UTIDLE 的影響。

Bit 0 **UTEIE** : 發送寄存器為空中斷使能位

0 : 發送寄存器為空中斷除能

1 : 發送寄存器為空中斷使能

此位為發送寄存器為空中斷的使能或除能位。若 UTEIE=1，當 UTXIF 置位時，USIM 的插斷要求標誌 USIMF 置位元；若 UTEIE=0，USIM 插斷要求標誌 USIMF不受 UTXIF 的影響。

● **UTXR_RXR 寄存器**

UTXR_RXR 是一個資料寄存器，用來存儲 TX 引腳將要發送或 RX 引腳正在接收的資料。

Bit	7	6	5	4	3	2	1	0
Name	UTXR7	UTXR6	UTXR5	UTXR4	UTXR3	UTXR2	UTXR1	UTXR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	x	x	x	x	x	x	x	x

x" : 未知

Bit 7~0 UTXR7~UTXR0 : UART 發送 / 接收資料位

● **UBRG 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	UBRG7	UBRG6	UBRG5	UBRG4	UBRG3	UBRG2	UBRG1	UBRG0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	x	x	x	x	x	x	x	x

x" : 未知

Bit 7~0 UBRG7~UBRG0 : 串列傳輸速率值

軟體設置 UBRGH 位元 (設置串列傳輸速率發生器的速度) 和 UBRG 寄存器 (設置串列傳輸速率的值)，一起控制 UART 的串列傳輸速率。

注：若 UBRGH=0，串列傳輸速率 = fH/[64×(N+1)]；若 UBRGH=1，串列傳輸速率 = fH/[16×(N+1)]。

串列傳輸速率發生器

UART 自身具有一個串列傳輸速率發生器，通過它可以設定資料傳輸速率。串列傳輸速率是由一個獨立的內部 8 位元數目器產生，它由 UBRG 寄存器和 UUCR2 寄存器的 UBRGH 位來控制。UBRGH 是決定串列傳輸速率發生器處於高速模式還是低速模式，從而決定計算公式的選用。UBRG 寄存器的值 N 可根據下表中的公式計算，N 的範圍是 0 到 255。

UUCR2 的 UBRGH 位	0	1
串列傳輸速率 (BR)	fH/[64×(N+1)]	fH/[16×(N+1)]

為得到相應的串列傳輸速率，首先需要設置 UBRGH 來選擇相應的計算公式從而算出 UBRG 的值。注意，由於 UBRG 的值不連續，所以實際串列傳輸速率和理論值之間有一個偏差。

下面舉例怎樣計算

UBRG 寄存器中的值 N 和誤差。串列傳輸速率和誤差的計算若選用 4MHz 時鐘頻率且 UBRGH=0，若期望的串列傳輸速率為 4800，計算它的 UBRG 寄存器的值 N，實際串列傳輸速率和誤差。根據上表，串列傳輸速率 BR = fH / [64 (N+1)] 轉換後的公式 N = [fH / (BR×64)] - 1

帶入參數 N = [4000000 / (4800×64)] - 1 = 12.0208

取最接近的值，十進位 12 寫入 UBRG 寄存器，實際串列傳輸速率如下

BR = 4000000 / [64 × (12+1)] = 4808 因此，誤差 = (4808 - 4800) / 4800 = 0.16%

UART 模塊的設置與控制

UART 採用標準的不歸零碼傳輸資料，這種方法通常被稱為 NRZ 法。它由 1 位起始位，8 位或 9 位元資料位元和 1 位或者兩位停止位組成。同位是由硬體自動完成的，可設置成奇數同位檢查、偶校驗和無校驗三種格式。常用的資料傳輸格式由 8 位元資料位元，1 位元停止位，無校驗組成，用 8、N、1 表示，它是系統上電的預設格式。資料位元數、停止位元數和同位由 UUCR1 寄存器的 UBNO、UPRT、UPREN 和 USTOPS 設定。用於資料發送和接收的串列傳輸速率由一個內部的 8 位元串列傳輸速率發送器產生，資料傳輸時低位元元在前高位在後。儘管 UART 發送器和接收器在功能上相互獨立，但它們使用相同的資料傳輸格式和串列傳輸速率，在任何情況下，停止位元是必須的。

UART 的使能和除能

UART 是由 UUCR1 寄存器的 UREN 位來使能和除能的。當 UART 模式通過在 SIMCO 寄存器中設置 UMD 位為 "1" 選擇時，若 UREN、UTXEN 和 URXEN 都為高，則 TX 和 RX 分別為 UART 的發送埠和接收埠。若沒有資料發送，TX 引腳預設狀態為高電平。

UREN 清零將除能 TX 和 RX，這兩個引腳通過配置相應的共用引腳控制位可用作普通 I/O 口或其它引腳共用功能。當

UART 功能被除能時將清空緩衝器，所有緩衝器中的資料將被忽略，另外一些使能控制、錯誤標誌和狀態標誌將被重定，如 UTXEN、URXEN、UTXBRK、URXIF、UOERR、UFERR、UPERR 和 UNF 清零，而 UTIDLE、UTXIF 和 URIDLE 置位，UUCR1、UUCR2 和 UBRG 寄存器中的其它位保持不變。若 UART 工作時 UARTEN 清零，所有發送和接收將停止，模組也將重定成上述狀態。當 UART 再次使能時，它將在上次配置下重新工作。

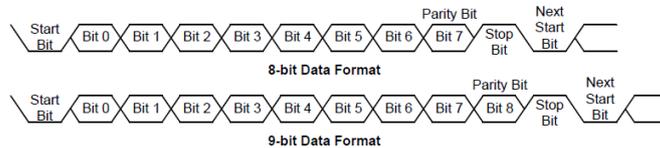
資料位元、停止位元位元數以及同位的選擇

資料傳輸格式由資料長度、是否校驗、校驗類型、位址位以及停止位長度組成。它們都是由 UUCR1 寄存器的各個位控制的。UBNO 決定資料傳輸是 8 位還是 9 位；UPRT 決定校驗類型；UPREN 決定是否選擇同位；而 USTOPS 決定選用 1 位還是 2 位停止位。下表列出了各種資料傳輸格式。若位址檢測功能使能，位址位元，即資料位元組的最高位元，用來確定此幀是位址還是資料。停止位元的長度和資料位元的長度無關，且只有發送器需設置停止位長度。接收器只接收一個停止位。

起始位	數據位元	地址位	校驗位	停止位
8 位元數據位元				
1	8	0	0	1
1	7	0	1	1
1	7	1	0	1
9 位元數據位元				
1	9	0	0	1
1	8	0	1	1
1	8	1	0	1

發送和接收資料格式

下圖是傳輸和接收 8 位和 9 位元資料的波形。



UART 發送器

UUCR1 寄存器的 UBNO 位是控制資料傳輸的長度。UBNO=1 其長度為 9 位，第 9 位元 MSB 存儲在 UUCR1 寄存器的 UTX8 中。發送器的核心是發送移位暫存器 TSR，它的資料由發送寄存器 UTXR_RXR 提供，應用程式只須將發送資料寫入 UTXR_RXR 寄存器。上組資料的停止位元發出前，TSR 寄存器禁止寫入。如果還有新的資料要發送，一旦停止位元發出，待發資料將會從 UTXR_RXR 寄存器載入到 TSR 寄存器。TSR 不像其它寄存器一樣映射到資料記憶體，所以應用程式不能對其進行讀寫操作。UTXEN=1，發送使能，但若 UTXR_RXR 寄存器沒有資料或者串列傳輸速率沒有設置，發送器將不會工作。先寫 UTXR_RXR 寄存器再置高 UTXEN 也會觸發發送。當發送器使能，若 TSR 寄存器為空，資料寫入 UTXR_RXR 寄存器將會直接載入到 TSR 寄存器中。發送器工作時，UTXEN 清零，發送器將立刻停止工作並且復位，此時 TX 輸出引腳用作普通 I/O 口或其它引腳共用功能。

發送資料

當 UART 發送資料時，資料從移位暫存器中移到 TX 引腳上，其低位元元在前高位在後。在發送模式中，UTXR_RXR 寄存器在內部匯流排和發送移位暫存器間形成一個緩衝。如果選擇 9 位資料傳輸格式，最高位 MSB 取自 UUCR1 寄存器的 TX8。

發送器初始化可由如下步驟完成：

- 正確地設置 UBNO、UPRT、UPREN 和 USTOPS 位元以確定資料長度、校驗類型和停止位長度。
- 設置 UBRG 寄存器，選擇期望的串列傳輸速率。
- 置高 UTXEN，使能 UART 發送器且使 TX 作為 UART 的發送端。
- 讀取 UUSR 寄存器，然後將待發資料寫入 UTXR_RXR 寄存器。注意，此步驟會清除 UTXIF 標誌位元。
- 如果要發送多個資料只需重複上一步驟。

當 UTXIF=0 時，資料將禁止寫入 UTXR_RXR 寄存器。可以通過以下步驟來清除 UTXIF：

1. 讀取 UUSR 寄存器
2. 寫 UTXR_RXR 寄存器

唯讀標誌位元 UTXIF 由 UART 硬體置位元。若 UTXIF=1，UTXR_RXR 寄存器為空，其它資料可以寫入而不會覆蓋之前的資料。若 UTEIE=1，UTXIF 標誌位元會產生中斷。

在資料傳輸時，寫 UTXR_RXR 指令會將待發資料暫存在 UTXR_RXR 寄存器中，當前資料發送完畢後，待發資料被載入到發送移位暫存器中。當發送器空閒時，寫 UTXR_RXR 指令會將資料直接載入到 TSR 寄存器中，資料傳輸立刻開始且 UTXIF 置位。當發送完停止位或暫停幀後，表示一幀資料已發送完畢，此時 UTIDLE 位將被置位。

可以通過以下步驟來清除 UTIDLE：

1. 讀取 UUSR 寄存器
2. 寫 UTXR_RXR 寄存器

注意，清除 UTXIF 和 UTIDLE 軟體執行次序相同。

發送暫停字

若 UTXBRK=1，下一幀將會發送暫停字。它是由一個起始位、13×N (N=1, 2,.....) 位元邏輯 0 組成。置位 UTXBRK 將會發送暫停字，而清除 UTXBRK 將產生停止位，傳輸暫停字不會產生中斷。需要注意的是，暫停字至少 13 位寬。若 UTXBRK 持續為高，那麼發送器會一直發送暫停字；當應用程式將 UTXBRK 清零後，發送器結束最後一幀暫停字的發送後接著發送一位或兩位停止位。最後一幀暫停字的結尾自動為高電平，以確保下一幀資料起始位元的檢測。

UART 接收器

UART 接收器支援 8 位或者 9 位元資料接收。若 UBNO=1，資料長度為 9 位，而最高位 MSB 存放在 UUCR1 寄存器的 URX8 中。接收器的核心是串列移位暫存器 RSR。RX 引腳上的資料送入資料恢復器中，它在 16 倍串列傳輸速率的頻率下工作，而串列移位元器工作在正常串列傳輸速率下。當在 RX 引腳上檢測到停止位，若 UTXR_RXR 寄存器為空，資料從 RSR 寄存器中載入到 UTXR_RXR 寄存器。

RX 引腳上的每一位元資料會被採樣三次以判斷其邏輯狀態。RSR 不像其它寄存器一樣映射在資料記憶體，所以應用程式不能對其進行讀寫操作。

接收資料

當 UART 接收資料時，資料低位元在前高位在後，連續地從 RX 引腳進入移位暫存器。在唯讀模式下，UTXR_RXR 寄存器在內部匯流排和接收移位暫存器間形成一個緩衝。UTXR_RXR 寄存器是一個兩層的 FIFO 資料緩衝器，它能保存兩幀資料的同時接收第三幀資料。注意，應用程式必須保證在接收完第三幀前讀取

UTXR_RXR 寄存器，否則忽略第三幀資料並且發生溢出錯誤。接收器的初始化可由如下步驟完成：

- 正確地設置 UBNO、UPRT 和 UPREN 位元以確定資料長度和校驗類型。
- 設置 UBRG 寄存器，選擇期望的串列傳輸速率。
- 置高 URXEN，使 RX 作為 UART 的接收端。此時接收器被使能並檢測起始位。

接收資料將會發生如下事件：

- 當 UTXR_RXR 寄存器中包含有效資料時，UUSR 寄存器中的 URXIF 位將會置位元，溢出錯誤發生之前至多還有一幀資料可讀。
- 若 URIE=1，資料從 RSR 寄存器載入到 UTXR_RXR 寄存器中將產生中斷。
- 若接收器檢測到幀錯誤、雜訊干擾錯誤、奇偶出錯或溢出錯誤，那麼相應的錯誤標誌位置位元。

可以通過如下步驟來清除 URXIF：

1. 讀取 UUSR 寄存器
2. 讀取 UTXR_RXR 寄存器

接收暫停字

UART 接收任何暫停字都會當作幀錯誤處理。接收器只根據 UBNO 位元的設置外加一個停止位元來確定一幀資料的長度。若暫停字位數大於 UBNO 位指定的長度外加一個停止位，接收器認為接收已完畢，URXIF 和 UFERR 置位，TXR_RXR 寄存器清 0，若相應的中斷允許且 URIDLE 為高將會產生中斷。暫停字只會被認為包含資訊 0 且會置位元 UFERR 標誌位元。如果檢測到較長的暫停信號，接收器會將此信號視為包含一個起始位元、資料位元和無效的停止位元的資料幀並且置位元 UFERR 標誌位元。在下個開始位到來之前，接收器必須等待一個有效的停止位。接收器不會假定線上的暫停信號是下一個開始位元。暫停字將會載入到緩衝器中，在接收到停止位元前不會再接收資料，需要注意的是，沒有檢測到停止位也會置位元唯讀標誌位元 URIDLE。UART 接收到暫停字會產生以下事件：

- 幀錯誤標誌位元 UFERR 置位元。
- UTXR_RXR 寄存器清零。
- UOERR、UNF、UPERR、URIDLE 或 URXIF 可能會置位。

空間狀態

當 UART 接收資料時，即在起初位和停止位之間，UUSR 寄存器的接收狀態標誌位元 URIDLE 清零。在停止位元和下一幀資料的起始位元之間，URIDLE 被置位，表示接收器空間。

接收中斷

UUSR 寄存器的唯讀標誌位元 URXIF 由接收器的邊沿觸發置位。若 URIE=1，資料從移位暫存器 RSR 載入到 UTXR_RXR 寄存器時產生中斷，同樣地，溢出也會產生中斷。

接收錯誤處理

UART 會產生幾種接收錯誤，下面部分將描述各錯誤以及怎樣處理。

溢出 – UOERR 標誌

UTXR_RXR 寄存器是一個兩層的 FIFO 緩衝器，它能保存兩幀資料的同時接收第三幀資料，應用程式必須保證在接收完第三幀前讀取 UTXR_RXR 寄存器，否則發生溢出錯誤。

產生溢出錯誤時將會發生以下事件：

- UUSR 寄存器中 UOERR 被置位。
- UTXR_RXR 寄存器中資料不會丟失。
- RSR 寄存器資料將會被覆蓋。
- 若 URIE=1，將會產生中斷。

先讀取 UUSR 寄存器再讀取 UTXR_RXR 寄存器可將 UOERR 清零。

雜訊幹擾 – UNF 標誌

資料恢復時多次採樣可以有效的鑒別出雜訊幹擾。當檢測到資料受到雜訊幹擾時將會發生以下事件：

- 在 URXIF 上升沿，UUSR 寄存器中唯讀標誌位元 UNF 置位元。
- 資料從 RSR 寄存器載入到 UTXR_RXR 寄存器中。
- 不產生中斷，但此位置位發生在 URXIF 置位產生中斷的同週期內。先讀取 UUSR 寄存器再讀取 UTXR_RXR 寄存器可將 UNF 清零。

幀錯誤 – UFERR 標誌

若在停止位上檢測到 0，UUSR 寄存器中唯讀標誌 UFERR 置位元。若選擇兩位停止位，此兩位都必須為高，否則將置位 UFERR。此標誌位元同接收的資料分別記錄在 UUSR 寄存器和 UTXR_RXR 寄存器中，此標誌位元可被任何重定清零。

同位錯誤 – UPERR 標誌

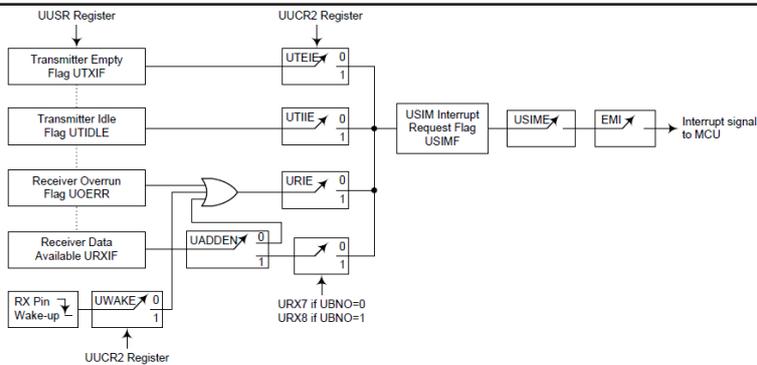
若接收到資料出現同位錯誤，UUSR 寄存器中唯讀標誌 UPERR 置位元。只有使能了同位，選擇了校驗類型，此標誌位元才有效。此標誌位元同接收的資料分別記錄在 UUSR 寄存器和 UTXR_RXR 寄存器中，此標誌位元可被任何重定清零。注意，在讀取相應的資料之前必須先訪問 UUSR 寄存器中的 UFERR 和 UPERR 錯誤標誌位元。

UART 模組中斷結構

幾個獨立的 UART 條件可以觸發一個 USIM 中斷。當條件滿足時，會產生一個低脈衝信號。發送寄存器為空、發送器空間、接收器資料有效、溢出和位址檢測和 RX 引腳喚醒都會產生中斷。若總中斷使能、USIM 中斷控制位元使能且堆疊未滿，程式將會跳轉到相應的中斷向量執行中斷服務程式，而後再返回主程序。其中四種情況，若其 UUCR2 寄存器中相應中斷允許位被置位，則 UUSR 寄存器中對應中斷標誌位元將產生 USIM 中斷。發送器相關的兩個中斷情況有各自對應的中斷允許位，而接收器相關的兩個中斷情況共用一個中斷允許位。這些允許位元可用於禁止個別 USIM UART 模式中斷源。

位元址檢測也是 USIM UART 模式的中斷源，它沒有相應的標誌位元，若 UUCR2 寄存器中 UADDEN=1，當檢測到位址將會產生 USIM 中斷。RX 引腳喚醒也可以產生 USIM UART 模式的中斷源，它沒有相應的標誌位元，當 UART 時鐘源 fH關閉且 UUCR2 中的 UWAKE 和 URIE 位被置位，RX 引腳上有下降沿時會產生 USIM 中斷。注意，在發生 RX 喚醒中斷時，將會有一個週期延遲，通常稱為系統啟動時間，以便振盪器在系統恢復正常工作之前重新開機和穩定。

注意，UUSR 寄存器標誌位元為唯讀狀態，軟體不能對其進行設置，和其它一些中斷一樣，在進入相應中斷服務程式時也不能清除這些標誌位元。這些標誌位元僅在 UART 特定動作發生時才會自動被清除，詳細解釋見 UART 寄存器章節。整體 UART 中斷的使能或除能可由中斷控制寄存器中的 USIM 中斷使能控制位控制，其插斷要求由 UART 模組決定。



UART 中斷框圖

位元址檢測模式

置位 UUCR2 寄存器中的 UADDE 將啟動位元址檢測模式。若此位元為 “1”，可產生接收資料有效中斷，其請求標誌位元為 URXIF。若 UADDE 使能，只有在接收到資料最高位元為 1 才會產生中斷，中斷允許位 USIM 和 EMI 也要使能才會產生中斷。地址的最高位為第 9 位 (UBNO=1) 或第 8 位 (UBNO=0)，若此位為高，則接收到的是位址而非資料。只有接收的資料的最後一位元為高才會產生中斷。若 UADDE 除能，每接收到一個有效資料便會置位元 URXIF，而不用考慮資料的最後一位元。位址檢測和同位在功能上相互排斥，若位元址檢測模式使能，為了確保操作正確，必須將同位使能位清零以除能同位。

UADDE	Bit 9 (UBNO=1) Bit 8 (UBNO=0)	產生 USIM 中斷
0	0	√
	1	√
1	0	×
	1	√

UADDE 位功能

UART 模組暫停和喚醒

UART 時鐘 fH 關閉後 UART 模組將停止運行。當傳送資料時 UART 時鐘 fH 關閉，發送將停止直到 UART 模組時鐘再次使能。同樣地，當接收資料時單片機進入空閒或休眠模式，資料接收也會停止。當單片機進入空閒或休眠模式，UUSR、UUCR1、UUCR2、接收 / 發送寄存器以及 UBRG 寄存器都不會受到影響。建議在單片機進入空閒或休眠模式前先確保資料發送或接收已完成。

UART 功能中包括了 RX 引腳的喚醒功能，由 UUCR2 寄存器中 UWAKE 位使能或除能。當 UART 時鐘 fH 關閉時，若 UMD 位與 UART 使能位 UREN、接收器使能位 URXEN 和接收器中斷使能位 URUE 都被置位，則 RX 引腳的下降沿可觸發產生 RX 引腳喚醒 UART 的中斷。注意，喚醒後系統需延時一段時間才能正常工作，在此期間，RX 引腳上的任何資料將被忽略。

若要產生喚醒 UART 的中斷，除了喚醒使能控制位和接收中斷使能控制位需置位外，全域中斷使能位 EMI 和 USIM 中斷使能控制位 USIME 也必須置位；若 EMI 和 USIME 位沒有被置位，那麼，可發生喚醒事件但不會產生中斷。注意，喚醒後系統需一定的延時才能正常工作，然後才會產生 USIM 中斷。

低電壓檢測 – LVD

該單片機都具有低電壓檢測功能，即 LVD。該功能使能用於監測電源電壓 VDD，若電源電壓低於一定值可提供一個警告信號。此功能在電池類產品中非常有用，在電池電壓較低時產生警告信號。低電壓檢測也可產生中斷信號。

LVD 寄存器

低電壓檢測功能由 LVDC 寄存器控制。VLVD2~VLVD0 位用於選擇 8 個固定的電壓參考點。LVDO 位元被置位元時低電壓情況發生，若 LVDO 位元為低表明 VDD 電壓工作在當前所設置低電壓水準值之上。LVDEN 位元用於控制低電壓檢測功能的開啟 / 關閉，設置此位為高使能此功能，反之，關閉內部低電壓檢測電路。低電壓檢測會有一定的功耗，在不使用時可考慮關閉此功能，此舉在功耗要求嚴格的電池供電應用中值得考慮。

● **LVDC 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	—	—	LVDO	LVDEN	VBGEN	VLVD2	VLVD1	VLVD0
R/W	—	—	R	R/W	R/W	R/W	R/W	R/W
POR	—	—	0	0	0	0	0	0

Bit 7~6 未定義，讀為 “0”

Bit 5 **LVDO** : LVD 輸出標誌位元

0 : 未檢測到低電壓

1 : 檢測到低電壓

Bit 4 **LVDEN** : 低電壓檢測使能控制位元

0 : 除能1 : 使能

Bit 3 **VBGEN** : Bandgap 緩衝控制位

0 : 除能

1 : 使能

注意，當 LVD 或 LVR 功能使能或 VBGEN 位置高時，Bandgap 電路使能。

Bit 2~0 **VLVD2~VLVD0** : LVD 電壓選擇位

000 : 2.0V

001 : 2.2V

010 : 2.4V

011 : 2.7V

100 : 3.0V

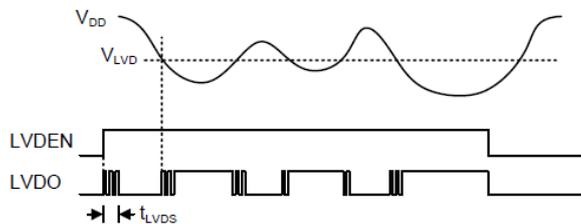
101 ; 3.3V

110 : 3.6V

111 : 4.0V

LVD 操作

低電壓檢測功能的工作原理是比較電源電壓 VDD 與 LVDC 寄存器定義的預置電壓值。預置電壓的範圍為 2.0V~4.0V。當電源電壓 VDD 低於預置電壓值時，LVDO 位元被置為高，表明低電壓產生。當單片機進入休眠模式，即使 LVDEN 位元為高，低電壓檢測器除能。低電壓檢測器使能後，讀取 LVDO 位元前，電路穩定需要一定的延時 tLVDS。注意，VDD 電壓可能上升或下降比較緩慢，在 VLVD 電壓值附近時，LVDO 位可能有多種變化。



LVD 操作

中斷

中斷是單片機一個重要功能。當外部事件或內部功能如計時器模組或 A/D 轉換器有效，並且產生中斷時，系統會暫時中止當前的程式而轉到執行相對應的中斷服務程式。此單片機提供多個外部中斷和內部中斷功能，外部中斷由 INT 引腳動作產生，而內部中斷由各種內部功能，如計時器模組、LVD、USIM 和 A/D 轉換器等產生。

中斷寄存器

中斷控制基本上是在一定單片機條件發生時設置請求標誌位元，應用程式中中斷使能位元的設置是通過位於特殊功能資料記憶體中的一系列寄存器控制的。寄存器總的分為兩類。第一類是 INTC0~INTC2 寄存器，用於設置基本的中斷；第二類是 INTEG 寄存器，用於設置外部中斷邊沿觸發類型。

寄存器中含有中斷控制位元和插斷要求標誌位元。中斷控制位用於使能或除能各種中斷，插斷要求標誌位元用於存放當前插斷要求的狀態。它們都按照特定的模式命名，前面表示中斷類型的縮寫，緊接著的字母 “E” 代表使能/除能位，“F” 代表請求標誌位元。

功能	使能位	請求標誌	注釋
總中斷	EMI	—	—
INT 腳	INTE	INTF	—
紅外線感應電路中斷	OPDE	OPDF	—
A/D 轉換器	ADE	ADF	—
STM	STMPE	STMPF	—
	STMAE	STMAF	—
LVD	LVE	LVF	—
EEPROM	DEE	DEF	—
時基	TBnE	TBnF	n=0~1
USIM	USIME	USIMF	—

中斷寄存器位元命名模式

寄存器名稱	位							
	7	6	5	4	3	2	1	0
INTEG	—	—	—	—	—	—	INTS1	INTS0
INTC0	—	ADF	OPDF	INTF	ADE	OPDE	INTE	EMI
INTC1	DEF	LVF	STMAF	STMPF	DEE	LVE	STMAE	STMPE
INTC2	—	USIMF	TB1F	TB0F	—	USIME	TB1E	TB0E

中斷寄存器列表

● **INTEG 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	INTS1	INTS0
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	0	0

- Bit 7~2 未定義，讀為“0”
- Bit 1~0 INTS1~INTS0：INT 腳中斷邊沿控制位
- 00：除能
- 01：上升沿
- 10：下降沿
- 11：雙沿

● **INTC0 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	—	ADF	OPDF	INTF	ADE	OPDE	INTE	EMI
R/W	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	—	0	0	0	0	0	0	0

- Bit 7 未定義，讀為“0”
- Bit 6 ADF：A/D 轉換器插斷要求標誌位元
- 0：無請求
- 1：插斷要求
- Bit 5 OPDF：紅外線感應電路插斷要求標誌位元
- 0：無請求
- 1：插斷要求
- Bit 4 INTF：INT 插斷要求標誌位元
- 0：無請求
- 1：插斷要求
- Bit 3 ADE：A/D 轉換器中斷控制位
- 0：除能
- 1：使能
- Bit 2 OPDE：紅外線感應電路中斷控制位元 (OPDINT)
- 0：除能
- 1：使能

Bit 1 **INTE** : INT 中斷控制位

0 : 除能

1 : 使能

Bit 0 **EMI** : 總中斷控制位

0 : 除能

1 : 使能

● **INTC1 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	DEF	LVF	STMAF	STMPF	DEE	LVE	STMAE	STMPE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **DEF** : 資料 EEPROM 插斷要求標誌位元

0 : 無請求

1 : 插斷要求

Bit 6 **LVF** : LVD 插斷要求標誌位元

0 : 無請求

1 : 插斷要求

Bit 5 **STMAF** : STM 比較器 A 匹配插斷要求標誌位元

0 : 無請求

1 : 插斷要求

Bit 4 **STMPF** : STM 比較器 P 匹配插斷要求標誌位元

0 : 無請求

1 : 插斷要求

Bit 3 **DEE** : 資料 EEPROM 中斷控制位元

0 : 除能

1 : 使能

Bit 2 **LVE** : LVD 中斷控制位

0 : 除能

1 : 使能

Bit 1 **STMPE** : STM 比較器 A 匹配中斷控制位

0 : 除能

1 : 使能

Bit 0 **STMPE** : STM 比較器 P 匹配中斷控制位

0 : 除能

1 : 使能

● **INTC2 寄存器**

Bit	7	6	5	4	3	2	1	0
Name	—	USIMF	TB1F	TBOF	—	USIME	TB1E	TBOE
R/W	—	R/W	R/W	R/W	—	R/W	R/W	R/W
POR	—	0	0	0	—	0	0	0

Bit 7 未定義，讀為 "0"

Bit 6 **USIMF** : USIM 插斷要求標誌位元

0 : 無請求

1 : 插斷要求

Bit 5 **TB1F** : 時基 1 插斷要求標誌位元

0 : 無請求

1 : 插斷要求

Bit 4 **TBOF** : 時基 0 插斷要求標誌位元

0 : 無請求

1 : 插斷要求

Bit 3 未定義，讀為 "0"

Bit 2 **USIME** : USIM 中斷控制位

0：除能

1：使能

Bit 1 **TB1E**：時基 1 中斷控制位

0：除能

1：使能

Bit 0 **TB0E**：時基 0 中斷控制位

0：除能

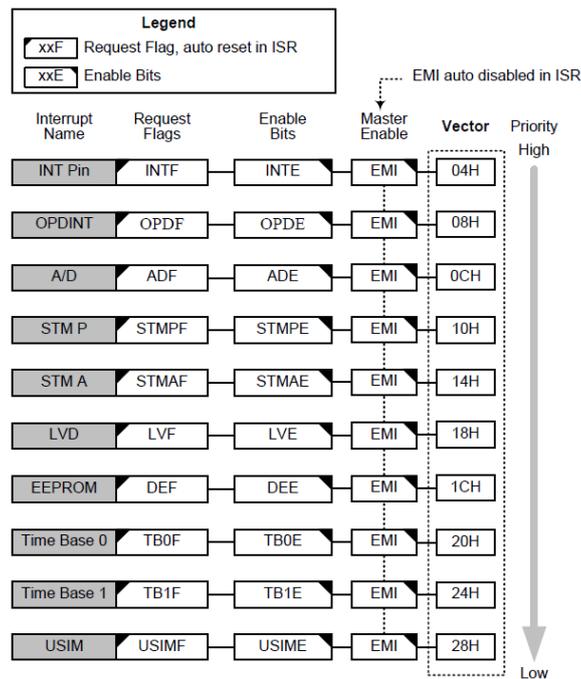
1：使能

中斷操作

若中斷事件條件產生，如一個 TM 比較器 P、比較器 A 匹配或 A/D 轉換結束等等，相關插斷要求標誌將置起。中斷標誌產生後程式是否會跳轉至相關中斷向量執行是由中斷使能位的條件決定的。若使能位元為 “1”，程式將跳至相關中斷向量中執行；若使能位元為 “0”，即使插斷要求標誌置起中斷也不會發生，程式也不會跳轉至相關中斷向量執行。若總中斷使能位為 “0”，所有中斷都將除能。

當中斷發生時，下條指令的位元址將被壓入堆疊。相應的中斷向量位址載入至 PC 中。系統將從此向量取下條指令。中斷向量處通常為 “JMP” 指令，以跳轉到相應的中斷服務程式。中斷服務程式必須以 “RETI” 指令返回至主程序，以繼續執行原來的程式。

各個中斷使能位元以及相應的請求標誌位元，以優先順序的次序顯示在下圖。每個中斷源有自己的向量。一旦中斷副程式被回應，系統將自動清除 EMI 位元，所有其它的中斷將被遮罩，這個方式可以防止任何進一步的中斷嵌套。其它插斷要求可能發生在此期間，雖然中斷不會立即回應，但是插斷要求標誌位元會被記錄。如果某個中斷服務副程式正在執行時，有另一個中斷要求立即回應，那麼 EMI 位元應在程式進入中斷副程式後置位元，以允許此中斷嵌套。如果堆疊已滿，即使此中斷使能，插斷要求也不會被回應，直到 SP 減少為止。如果要求立刻動作，則堆疊必須避免成為儲滿狀態。請求同時發生時，執行優先順序如下流程圖所示。所有被置起的插斷要求標誌都可把單片機從休眠或空閒模式中喚醒，若要防止喚醒動作發生，在單片機進入休眠或空閒模式前應將相應的標誌置起。



中斷結構

外部中斷

通過 INT 引腳上的信號變化可控制外部中斷。當觸發沿選擇位設置好觸發類型，外部插斷要求標誌 INTF 被置位元時外部插斷要求產生。若要跳轉到相應中斷向量位址，總中斷控制位 EMI 和相應中斷使能位 INTE 需先被置位。此外，必須使用 INTEG 寄存器使能外部中斷功能並選擇觸發沿類型。外部中斷引腳和普通 I/O 口共用，如果相應中斷寄存器中的外部中斷使能位被置位，需通過相應引腳共用功能選擇位選擇外部中斷引腳，此引腳將被作為外部中斷腳使用。此時該引腳必須通過設置埠控制寄存器，將該引腳設置為輸入口。當中斷使能，堆疊未滿並且外部中斷腳狀態改變，將調用外部中斷向量子程式。當回應外部中斷服務副程式時，插斷要求標誌位元 INTF 會自動重定且 EMI 位會被清零以除能其它中斷。注意，即使此引腳被

用作外部中斷輸入，其上拉電阻仍保持有效。

寄存器 INTEG 被用來選擇有效的邊沿類型，來觸發外部中斷。可以選擇上升沿還是下降沿或雙沿觸發都產生外部中斷。注意 INTEG 也可以用來除能外部中斷功能。

紅外線感應中斷

當紅外線感應電路檢測到紅外線時，紅外線感應插斷要求標誌位元 OPDF 被置位元，紅外線感應插斷要求發生。另外，必須通過 OPDC0 寄存器中的 OPDINTS0~OPDINTS1 位選擇正確的中斷邊沿類型。當總中斷使能位 EMI 和紅外線感應中斷使能位 OPDINTE 被置位元，允許程式跳轉到相應的中斷向量位址。總當中斷使能，堆疊未滿且檢測到紅外線時，將調用紅外線感應中斷向量副程式。當回應中斷服務副程式時，相應的插斷要求標誌位元 OPDF 會自動清零。EMI 位也會被清零以除能其它中斷。

A/D 轉換器中斷

A/D 轉換器中斷由 A/D 轉換動作的結束來控制。當 A/D 轉換器插斷要求標誌 ADF 被置位，即 A/D 轉換過程完成時，插斷要求發生。若要跳轉到相應中斷向量位址，總中斷控制位 EMI 和 A/D 轉換器中斷使能位 ADE 需先被置位。當中斷使能，堆疊未滿且 A/D 轉換動作結束時，將調用 A/D 轉換器中斷向量子程式。當回應中斷服務副程式時，相應的插斷要求標誌位元 ADF 會自動重定且 EMI 位會被清零以除能其它中斷。

TM 中斷

標準型 TM 有兩個中斷，分別來自比較器 P、A 匹配。所有類型的 TM 都有各自的向量，它們都有兩個插斷要求標誌位元及兩個使能位。當 TM 比較器 P、A 匹配情況發生時，相應 TM 插斷要求標誌被置位元，TM 插斷要求產生。若要程式跳轉到相應中斷向量位址，總中斷控制位 EMI、相應 TM 中斷使能位需先被置位。當中斷使能，堆疊未滿且 TM 比較器匹配情況發生時，可跳轉至相關 TM 中斷向量子程式中執行。當 TM 中斷回應，TM 插斷要求標誌自動重定且 EMI 將被自動清零以除能其它中斷。

LVD 中斷

當低電壓檢測功能檢測到一個低電壓時，LVD 插斷要求標誌 LVF 被置位元，LVD 插斷要求產生。若要程式跳轉到相應中斷向量位址，總中斷控制位元 EMI、低電壓中斷使能位元 LVE 需先被置位。當中斷使能，堆疊未滿且低電壓條件發生時，可跳轉至 LVD 中斷向量子程式中執行。當 LVD 中斷回應，插斷要求標誌 LVF 將自動重定且 EMI 將被自動清零以除能其它中斷。

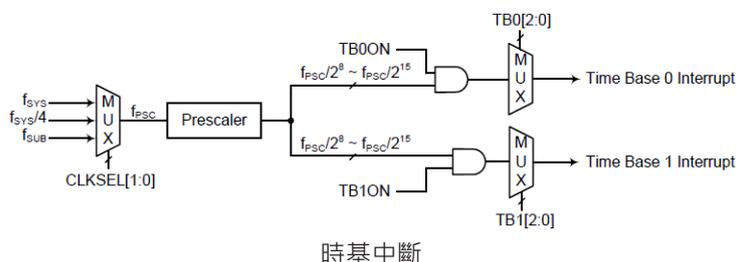
EEPROM 中斷

當寫週期結束，EEPROM 寫插斷要求標誌 DEF 被置位元，EEPROM 寫插斷要求產生。若要程式跳轉到相應中斷向量位址，總中斷控制位 EMI、EEPROM 寫中斷使能位 DEE 需先被置位。當中斷使能，堆疊未滿且 EEPROM 寫週期結束時，可跳轉至 EEPROM 中斷向量子程式中執行。當 EEPROM 寫中斷回應，插斷要求標誌 DEF 將自動重定且 EMI 將被自動清零以除能其它中斷。

時基中斷

時基中斷提供一個固定週期的中斷信號，由各自的計時器功能產生溢出信號控制。當各自的插斷要求標誌 TBOF 或 TB1F 被置位時，插斷要求發生。當總中斷使能位 EMI 和時基使能位 TBOE 或 TB1E 被置位元，允許程式跳轉到各自的中斷向量位址。當中斷使能，堆疊未滿且時基溢出時，將調用它們各自的中斷向量子程式。當回應中斷服務副程式時，相應的插斷要求標誌位元 TBOF 或 TB1F 會自動復位且 EMI 位會被清零以除能其它中斷。

時基中斷的目的是提供一個固定週期的中斷信號。其時鐘源 fPSC 來自內部時鐘源 fSYS, fSYS/4 或 fSUB，然後經過分頻器，分頻率由程式設置 TBOC 和 TB1C 寄存器相關位獲取合適的分頻值以提供更長的時基中斷週期。控制時基中斷週期的時鐘源，可來自 PSCR 寄存器中的 CLKSEL1~CLKSEL0 位。



● PSCR 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	CLKSEL1	CLKSEL0
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	0	0

Bit 7~2 未定義，讀為 “0”
 Bit 1~0 CLKSEL1~CLKSEL0：預分頻時鐘源選擇位元
 00：fSYS
 01：fSYS/4
 1x：fSUB

● TB0C 寄存器

Bit	7	6	5	4	3	2	1	0
Name	TB0ON	—	—	—	—	TB02	TB01	TB00
R/W	R/W	—	—	—	—	R/W	R/W	R/W
POR	0	—	—	—	—	0	0	0

Bit 7 TB0ON：時基 0 控制位
 0：除能
 1：使能
 Bit 6~3 未定義，讀為 “0”
 Bit 2~0 TB02~TB00：時基 0 溢出週期選擇位
 000：2⁸/fPSC
 001：2⁹/fPSC
 010：2¹⁰/fPSC
 011：2¹¹/fPSC
 100：2¹²/fPSC
 101：2¹³/fPSC
 110：2¹⁴/fPSC
 111：2¹⁵/fPSC

● TB1C 寄存器

Bit	7	6	5	4	3	2	1	0
Name	TB1ON	—	—	—	—	TB12	TB11	TB10
R/W	R/W	—	—	—	—	R/W	R/W	R/W
POR	0	—	—	—	—	0	0	0

Bit 7 TB1ON：時基 1 控制位
 0：除能
 1：使能
 Bit 6~3 未定義，讀為 “0”
 Bit 2~0 TB12~TB10：時基 1 溢出週期選擇位
 000：2⁸/fPSC
 001：2⁹/fPSC
 010：2¹⁰/fPSC
 011：2¹¹/fPSC
 100：2¹²/fPSC
 101：2¹³/fPSC
 110：2¹⁴/fPSC
 111：2¹⁵/fPSC

USIM 中斷

通用序列介面模組中斷，即 USIM 中斷。當 USIM 介面中斷標誌位元 USIMF 置位元時，產生插斷要求。由於 USIM 介面可工作在三個模式下：SPI 模式、I²C 模式和 UART 模式，USIMF 標誌位置位元可由不同情況觸發，取決於所選擇的介面

模式。

若選擇 SPI 或 I²C 模式，當一個位元組資料已由 SPI/I²C 介面接收或發送完，或 I²C 從機位址匹配，或 I²C 超時，USIM 插斷要求產生。若選擇 UART 模式，USIM 中斷由幾種 UART 傳輸條件控制。當發送器為空、發送器空閒、接收器資料有效、接收器溢出、位址檢測和 RX 引腳喚醒，USIM 插斷要求標誌 USIMF 被置位，USIM 插斷要求產生。

若要程式跳轉到相應中斷向量位址，總中斷控制位 EMI 和通用序列介面中斷使能位 USIME 需先被置位。當中斷使能，堆疊未滿且以上任一情況發生時，將調用相應的 USIM 中斷向量子程式。當回應中斷服務副程式時，通用序列介面中斷標誌位元 USIMF 會自動重定且 EMI 將被自動清零以除能其它中斷。

注意，當 USIM 中斷是由 UART 介面觸發產生的，當中斷響應後，UUSR 寄存器裡的標誌位元只有在對 UART 執行特定動作時才會被清零，詳細參考 UART 章節。

中斷喚醒功能

每個中斷都具有將處於休眠或空閒模式的單片機喚醒的能力。當插斷要求標誌由低到高轉換時喚醒動作產生，其與中斷是否使能無關。因此，儘管單片機處於休眠或空閒模式且系統振盪器停止工作，如有外部中斷腳上產生外部邊沿跳變、低電壓或比較器輸入改變都可能導致其相應的中斷標誌被置位元，由此產生中斷，因此必須注意避免偽喚醒情況的發生。若中斷喚醒功能被除能，單片機進入休眠或空閒模式前相應插斷要求標誌應被置起。中斷喚醒功能不受中斷使能位的影響。

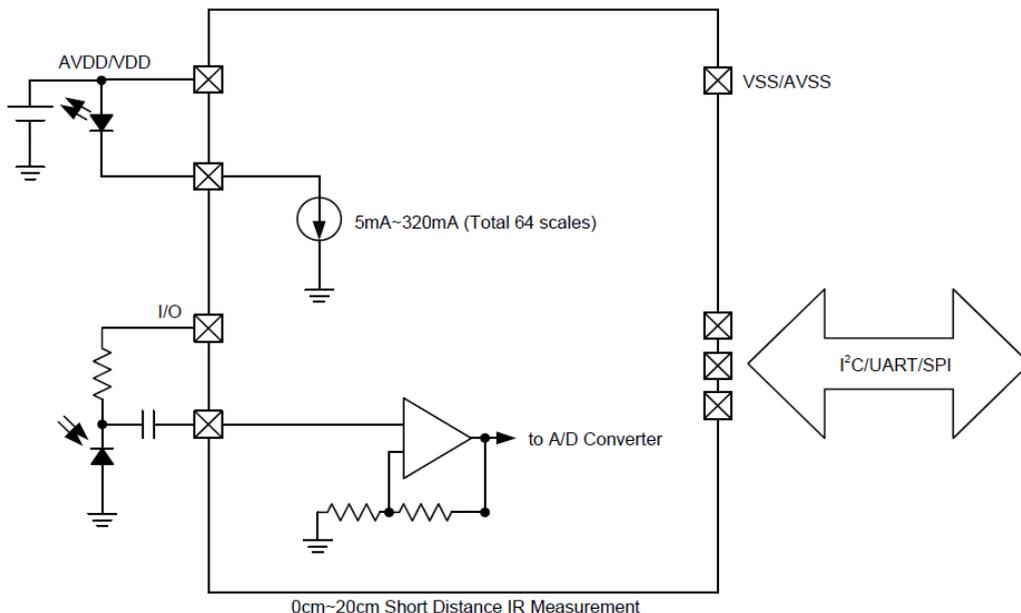
程式設計注意事項

通過禁止相關中斷使能位，可以遮罩插斷要求，然而，一旦插斷要求標誌位元被設定，它們會被保留在中斷控制寄存器內，直到相應的中斷服務副程式執行或請求標誌位元被應用程式清除。

建議在中斷服務副程式中不要使用 “CALL 副程式” 指令。中斷通常發生在不可預料的情況或是需要立刻執行的某些應用。假如只剩下層堆疊且沒有控制好中斷，當 “CALL 副程式” 在中斷服務副程式中執行時，將破壞原來的控制序列。所有中斷在休眠或空閒模式下都具有喚醒功能，當插斷要求標誌發生由低到高的轉變時都可產生喚醒功能。若要避免相應中斷產生喚醒動作，在單片機進入休眠或空閒模式前需先將相應請求標誌置為高。當進入中斷服務程式，系統僅將程式計數器的內容壓入堆疊，如果中斷服務程式會改變狀態寄存器或其它的寄存器的內容而破壞控制流程，應事將這些資料保存起來。

若從中斷副程式中返回可執行 RET 或 RETI 指令。除了能返回至主程序外，RETI 指令還能自動設置 EMI 位為高，允許進一步中斷。RET 指令只能返回至主程序，清除 EMI 位，除能進一步中斷。

應用電路



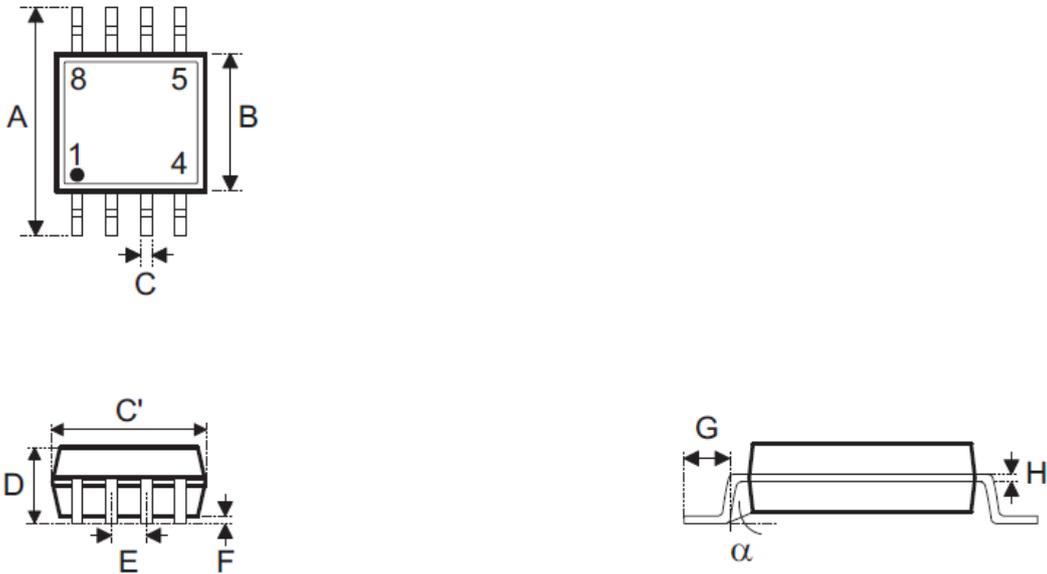
封裝信息

請注意，這裡提供的封裝資訊僅作為參考。由於這個資訊經常更新，提醒使用者諮詢 [Holtek 網站](#) 以獲取最新版本的封裝信息。

封裝資訊的相關內容如下所示，點擊可連結至 Holtek 網站相關資訊頁面。

- 封裝資訊 (包括外形尺寸、包裝帶和卷軸規格)
- 封裝材料資訊
- 紙箱信息

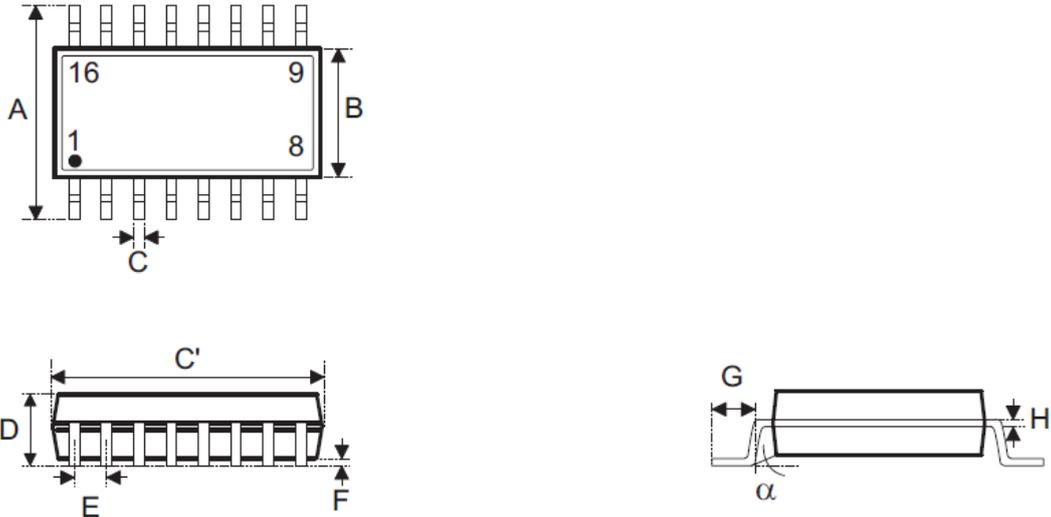
8-pin SOP (150mil) 外形尺寸



符號	尺寸 (單位 : inch)		
	最小值	典型值	最大值
A	—	0.236 BSC	—
B	—	0.154 BSC	—
C	0.012	—	0.020
C'	—	0.193 BSC	—
D	—	—	0.069
E	—	0.050 BSC	—
F	0.004	—	0.010
G	0.016	—	0.050
H	0.004	—	0.010
α	0°	—	8°

符號	尺寸 (單位 : mm)		
	最小值	典型值	最大值
A	—	6.00 BSC	—
B	—	3.90 BSC	—
C	0.31	—	0.51
C'	—	4.90 BSC	—
D	—	—	1.75
E	—	1.27 BSC	—
F	0.10	—	0.25
G	0.40	—	1.27
H	0.10	—	0.25
α	0°	—	8°

16-pin NSOP (150mil) 外形尺寸



符號	尺寸 (單位 : inch)		
	最小值	典型值	最大值
A	—	0.236 BSC	—
B	—	0.154 BSC	—
C	0.012	—	0.020
C'	—	0.390 BSC	—
D	—	—	0.069
E	—	0.050 BSC	—
F	0.004	—	0.010
G	0.016	—	0.050
H	0.004	—	0.010
α	0°	—	8°

符號	尺寸 (單位 : mm)		
	最小值	典型值	最大值
A	—	6.00 BSC	—
B	—	3.90 BSC	—
C	0.31	—	0.51
C'	—	9.90 BSC	—
D	—	—	1.75
E	—	1.27 BSC	—
F	0.10	—	0.25
G	0.40	—	1.27
H	0.10	—	0.25
α	0°	—	8°