

HT8 MCU I²C 應用須知及範例

文件編碼：AN0590TC

簡介

I²C 是由飛利浦公司開發的兩線序列傳輸介面，是一種同步半雙工通訊，具有兩條訊號線：數據線 SDA 與時鐘線 SCL，被廣泛應用於傳感器的資料傳輸、EEPROM 資料讀取以及 MCU 之間的通訊等方面。

本文將對 HT8 MCU I²C 的通訊以及使用進行介紹，幫助客戶快速上手 HT8 MCU 的 I²C 功能，加速產品開發。

功能說明

HT8 MCU 提供了獨立的 I²C、SIM、USIM 的三種 I²C 功能介面，下表列出三種 I²C 功能主要差異。

區別	獨立 I ² C	SIM	USIM
支援通訊協議	I ² C	SPI、I ² C	SPI、I ² C、UART
I ² C 選取	無複用功能，不需選擇	通過暫存器 SIMC0 的 SIM[2:0]位元選擇 I ² C 功能	通過暫存器 SIMC0 的 UMD、SIM[2:0]位元選擇 I ² C 功能

三種 I²C 功能介面可支援的通訊協議和選取方式會略有不同，但在 I²C 應用上較為相似。

HT8 MCU 內置 I²C 介面目前僅支援從機模式，一般用於 MCU 之間的通訊。以下將以 SIM 的 I²C 為例進行介紹。若要使用 HT8 MCU 作為 I²C 主機與外設通訊，需使用軟體模擬 I²C 的方式，本文同樣提供了相關說明，請參考“範例 2- 模擬 I²C 主機存取外部 EEPROM”章節內容。

I²C 通訊說明

I²C 通訊過程主要分成主機尋址和數據傳輸兩個階段，通訊由起始訊號、從機位址與讀寫位發送、數據傳輸和停止訊號組成。I²C 通訊過程中包含應答機制，在主機尋址過程中，從機接收到讀寫位後，若位址匹配，會回傳應答訊號。在數據傳輸過程中，接收方在接收完 8-bit 數據後，若需繼續接收，則在第 9 個時鐘時發出應答訊號。

主機尋址

HT8 MCU 作為從機與主機進行通訊前，首先要通過 SIMA 暫存器設定從機位址，而後在通訊過程中，接收到相匹配的主機尋址訊號(起始訊號及從機位址)後，才能做出應答。下圖為主機發送起始訊號以及從機位址的通訊時序圖。

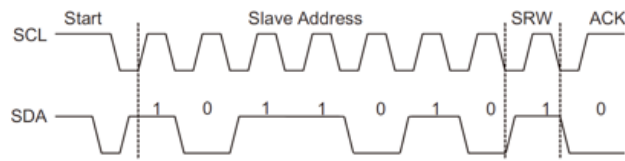


圖 1. 主機發送從機位址及讀寫位元時序圖

Start 訊號是指在 SCL 為高準位時，SDA 線上發生的從高到低的準位變化。當 HT8 MCU 偵測到 I²C 總線上的 Start 訊號時，MCU 會自動將 I²C 總線忙旗標 HBB 置位。

當主機發送的從機位址與 HT8 MCU SIMA[7:1]中設定的從機位址匹配時，HT8 MCU 從機會自動將位址匹配旗標 HAAS 置 1，讀寫狀態位則存到 SRW 位元中，隨後從機 MCU 自動向主機發送一個應答訊號 ACK，通知主機可繼續進行通訊，並根據 SRW 位判斷從機要處於接收還是發送模式。

數據傳輸

主機通過發送尋址訊號與從機建立通訊連接後即可進行數據傳輸，數據傳輸的方向由 SRW 位元決定。數據傳輸的時序圖如下：

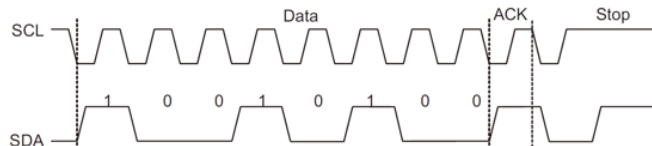


圖 2. 數據傳輸時序圖

- 從機接收數據(SRW=0)

當接收數據時，總線上傳輸的數據會存到暫存器 SIMD 中，讀取該暫存器即可接收到主機發送的數據；如果要繼續接收數據，則需要將 TXAK 設為 0，以此發送應答訊號通知主機繼續傳輸數據，無需接收數據時將該旗標設為 1。

需要注意的是 TXAK 位元在接收到數據之後的下一個時鐘週期傳送到總線上，因此設定 TXAK 位元的時機要選擇在接收數據完成傳輸之前。

- 從機發送數據(SRW=1)

當需要向主機發送數據，將要發送的數據寫入暫存器 SIMD，等待數據自動傳輸完成後，通過檢查 RXAK 位元的狀態可以判斷主機是否有發送應答訊號過來。

當數據傳輸完成發送方沒有收到來自接收方的應答訊號時，即認為數據傳輸完成，可由主機發送停止訊號結束通訊。

停止訊號

數據通訊結束後，可由主機在總線上發出 Stop 訊號以結束 I²C 通訊。Stop 訊號是指當 SCL 為高準位時，SDA 線上發生從低到高的準位變化。當 HT8 MCU 偵測到 I²C 總線上的 Stop 訊號後，會自動將 I²C 總線忙旗標 HBB 清零。

I²C 電氣特性

HT8 MCU I²C 外設支持最高 100kHz 標準模式和 400kHz 高速模式，不同通訊速度下對 MCU 系統時鐘的要求以及 SDA 和 SCL 總線線路的詳細特性參數說明如下。

Operating Temperature: Ta=25°C Typical

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V _{DD}	Conditions				
f _{I2C}	I ² C Standard Mode (100kHz) f _{SYS} Frequency ^(Note)	—	No clock debounce	2	—	—	MHz
		—	2 system clock debounce	4	—	—	MHz
		—	4 system clock debounce	4	—	—	MHz
	I ² C Fast Mode (400kHz) f _{SYS} Frequency ^(Note)	—	No clock debounce	4	—	—	MHz
		—	2 system clock debounce	8	—	—	MHz
		—	4 system clock debounce	8	—	—	MHz
f _{SCL}	SCL Clock Frequency	3V/5V	Standard mode	—	—	100	kHz
			Fast mode	—	—	400	
t _{SCL(H)}	SCL Clock High Time	3V/5V	Standard mode	3.5	—	—	μs
			Fast mode	0.9	—	—	
t _{SCL(L)}	SCL Clock Low Time	3V/5V	Standard mode	3.5	—	—	μs
			Fast mode	0.9	—	—	
t _{FALL}	SCL and SDA Fall Time	3V/5V	Standard mode	—	—	1.3	μs
			Fast mode	—	—	0.34	
t _{RISE}	SCL and SDA Rise Time	3V/5V	Standard mode	—	—	1.3	μs
			Fast mode	—	—	0.34	
t _{SU(SDA)}	SDA Data Setup Time	3V/5V	Standard mode	0.25	—	—	μs
			Fast mode	0.1	—	—	
t _{H(SDA)}	SDA Data Hold Time	3V/5V	—	0.1	—	—	μs
t _{VD(SDA)}	SDA Data Valid Time	3V/5V	—	—	—	0.6	μs
t _{SU(STA)}	Start Condition Setup Time	3V/5V	Standard mode	3.5	—	—	μs
			Fast mode	0.6	—	—	
t _{H(STA)}	Start Condition Hold Time	3V/5V	Standard mode	4.0	—	—	μs
			Fast mode	0.6	—	—	
t _{SU(STO)}	Stop Condition Setup Time	3V/5V	Standard mode	3.5	—	—	μs
			Fast mode	0.6	—	—	

註：使用 Debounce 功能可使傳輸更穩定，降低受干擾影響通訊失敗的機率。

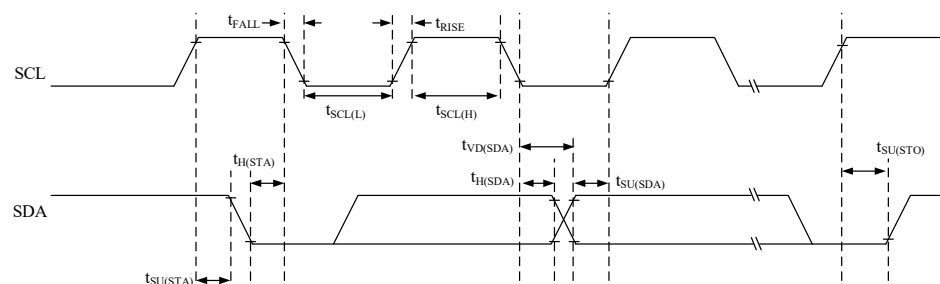


圖 3. I²C 時序圖

I²C 介面初始化

使用 HT8 MCU I²C 進行通訊前，需依以下步驟對 I²C 介面做初始化。

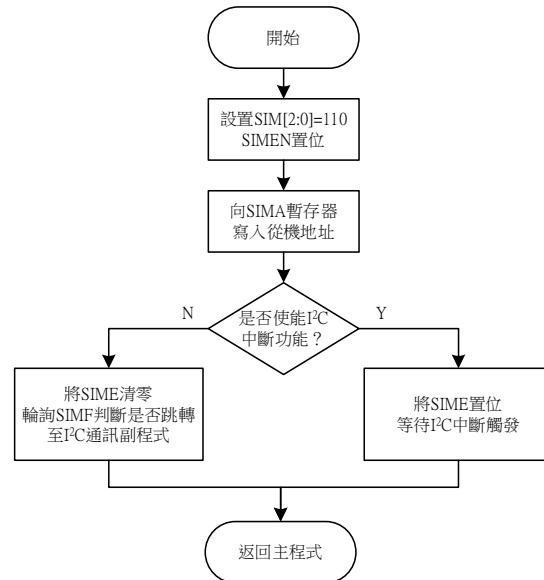


圖 4. I²C 初始化流程圖

上圖為 I²C 初始化流程，在 I²C 總線開始傳送資料前，需要先初始化 I²C 總線，初始化步驟如下：

- 步驟 1：將 SIM[2:0]設為"110"決定 I²C 從機模式。
- 步驟 2：將 SIMEN 設定為 1，以開啟 I²C 介面。
- 步驟 3：寫入從機位址於 SIMA 暫存器。
- 步驟 4：將 I²C 總線中斷控制暫存器之 SIME 設定為 1，以允許 I²C 總線中斷。

如果不使用中斷，就需要在程式中不斷的檢查 SIMF 位元，用來檢測 I²C 總線是否發生位址匹配或資料傳輸完成，這種方法比較浪費 MCU 資源，所以這邊不推薦使用。

I²C 中斷流程

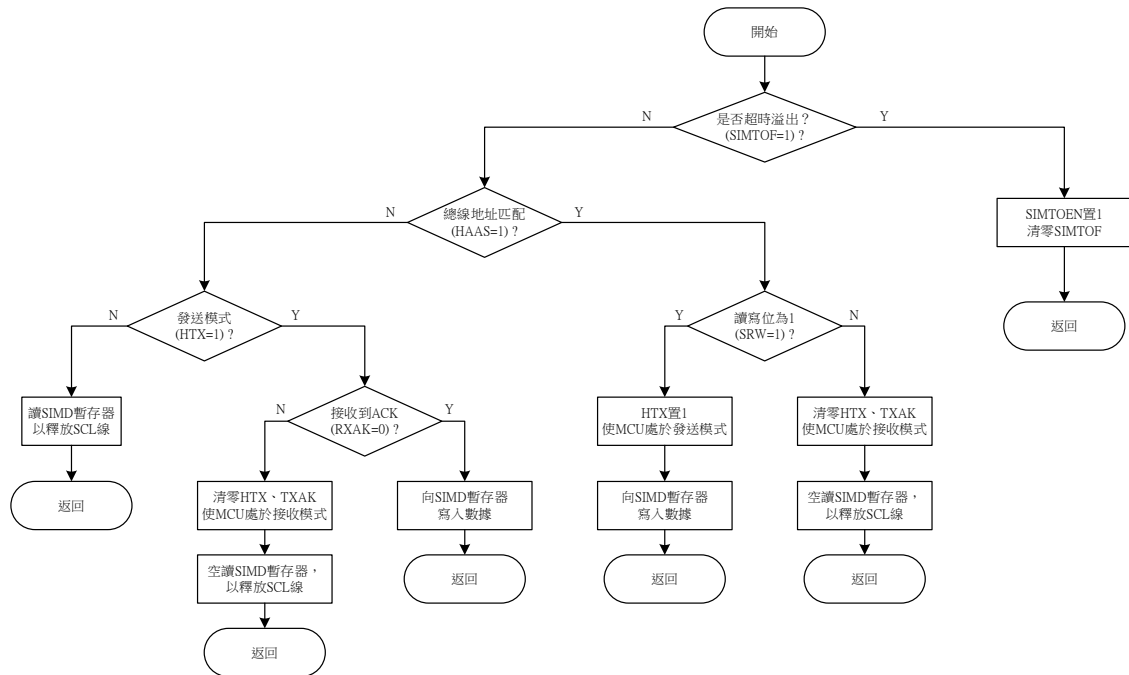


圖 5. I²C 總線中斷服務程式流程圖

I²C 的中斷會在超時發生(部分 MCU 具有超時功能)、位址匹配或數據接收完成時觸發。當 MCU 進入中斷服務程式後，應首先檢查 SIMTOF 位元和 HAAS 位元，SIMTOF 位元可以確定 I²C 總線中斷是否來自 I²C 超時，HAAS 位元則可以判斷該中斷是來自從機位址匹配，還是來自 8 位資料傳送完畢。

如果是位址匹配，程式應檢查 SRW，以確定主機是希望發送資料還是接收資料。根據此位設定好從機此時的狀態，還必須寫入資料到 SIMD 或從 SIMD 讀取資料以釋放 SCL 總線。如果是 8 位元資料傳送完畢，先判斷從機此時處於接收或者是發送模式(HTX)，如果從機設備是接收模式，就要立刻從 SIMD 中讀取資料，如果從機是發送模式，會繼續檢查 RXAK 位元以決定是否要繼續發送下一筆資料。如果沒檢測到應答訊號(RXAK=1)，從機應先切換為接收模式然後從 SIMD 虛擬讀取資料以釋放 SCL 線。

範例 1 - I²C 從機資料傳輸

本範例以 BH66F5350 作為 I²C 從機，使用 I²C 中斷的方式等待主機的請求。使用者可根據自己的需要按照程式說明自行修改。

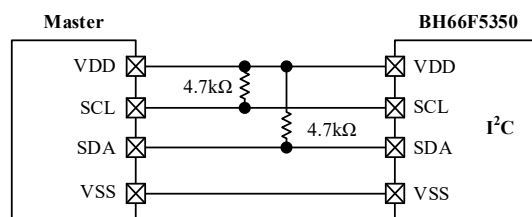


圖 6. I²C 通訊電路圖

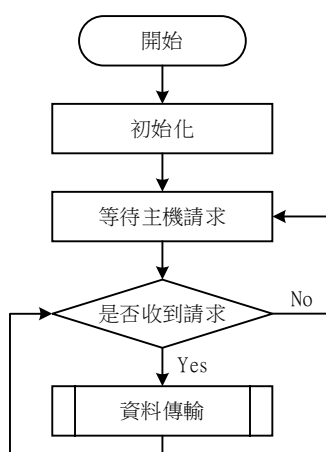


圖 7. 主流程圖

- 步驟 1：進行初始化設定，主要執行系統時鐘設置，從機相關暫存器設置，中斷設置，初始資料定義。
- 步驟 2：等待主機請求，當主機有所請求觸發 I²C 中斷進行資料傳輸。

註：1. I²C 中斷流程請參考 I²C 總線中斷服務程式流程圖(圖 5)。

2. I²C 初始化流程請參考圖 4。

3. 此篇應用筆記附檔同時提供了範例程式源文件以供用戶參考。

範例 2 - 模擬 I²C 主機存取外部 EEPROM

本範例以 HT66L2550A GPIO 模擬 I²C Master SCL/SDA 功能，實現對外部 EEPROM HT24LC02A 頁寫數據和任意位址讀數據的功能。HT24LC02A 主要規格如表 1 所示。

I²C Master 功能使用 HT66L2550A GPIO 模擬，應注意的是 I²C 設備一般採用開漏模式與總線相連，這就要求 I²C 總線需要外加提升電阻。總線上通過將 SCL/SDA 在不同時間拉低拉高，表示資料 0 或 1。若 SCL/SDA 輸出 0 時，則 GPIO 輸出接地，SCL/SDA 被拉低；若 SCL/SDA 輸出 1 時，I/O 埠不能直接輸出高準位，而是讓總線上的提升電阻去把 SCL/SDA 拉高。

項目	HT24LC02A
容量	2K(256×8)
通訊介面	I ² C
設備位址	1010000(7-bit)
操作模式	位元組寫、頁寫、當前位址讀、任意位址讀、順序位址讀
寫迴圈週期	最大 5ms

表 1. HT24LC02A 主要規格

硬體說明

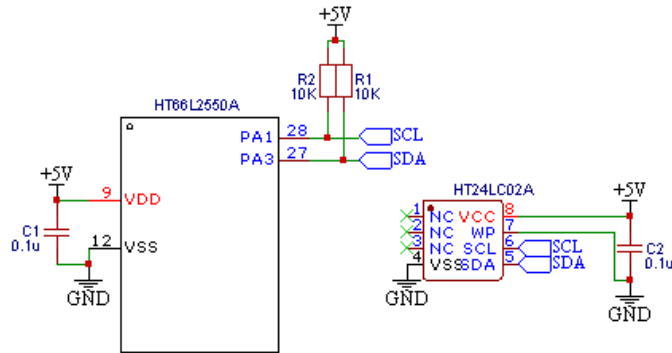


圖 8. I²C 存取外部 EEPROM 電路圖

註：本範例未使用寫保護功能，WP 腳位接 GND。

軟體說明

本範例主要實現由 GPIO 模擬 I²C Master 對 HT24LC02A 的 Page0 先寫入數據，最後再讀取 Page0 數據並校驗的功能。寫入/讀取資料依 HT24LC02A Page size 設定 Send_buffer 和 Receive_buffer size 為 8-byte，客戶可根據實際 EEPROM 修改。

下面從主程式流程圖、寫入數據流程圖、讀取數據流程圖介紹 I²C 存取外部 EEPROM 的過程。

主流程圖

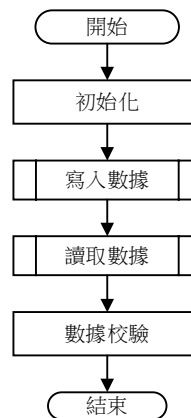


圖 9. 主流程圖

主程式說明

- 步驟 1：進行初始化設定，主要執行系統時鐘設定、GPIO(模擬 SDA、SCL)初始化設定等。
- 步驟 2：採用頁寫模式對 Page0 寫入 Send_buffer 數據。
- 步驟 3：採用任意位址讀模式讀取 Page0 數據存放在 Receive_buffer。
- 步驟 4：比較 Send_buffer 和 Receive_buffer 數據是否相同。

寫入數據流程圖

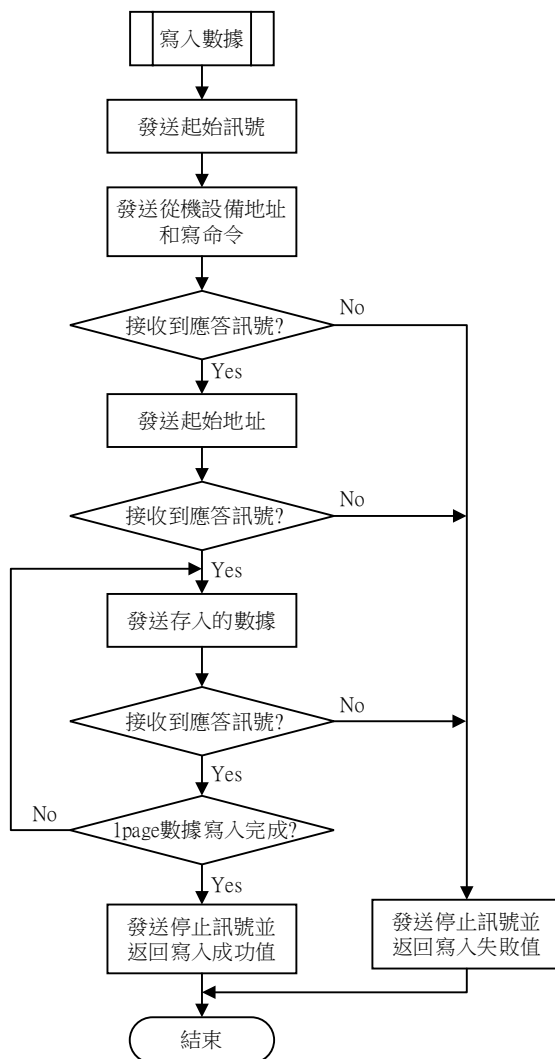


圖 10. 寫入數據流程圖

寫入數據說明

- 步驟 1：主機呼叫 {SW_I2C_Send_Start()} 發送起始訊號，然後呼叫 {SW_I2C_Send_Addr(0x50, TX_Mode)} 發送從機設備位址和寫命令，等待從機的應答。
- 步驟 2：若從機有應答 (ACK=0)，則進行下一個步驟，若無應答 (ACK=1)，則轉跳到步驟 8。
- 步驟 3：主機呼叫 {SW_I2C_Send_Data(WriteAddr)} 發送 Page0 首位址，等待從機應答。
- 步驟 4：若從機有應答 (ACK=0)，則進行下一個步驟，若無應答 (ACK=1)，則轉跳到步驟 8。
- 步驟 5：主機呼叫 {SW_I2C_Send_Data(Buffer_data[i])} 發送需要存入外部 EEPROM 的數據。
- 步驟 6：等待從機應答，若有應答 (ACK=0)，則進行下一個步驟，若無應答 (ACK=1)，則進行步驟 8。
- 步驟 7：判斷是否寫完 1page 數據，若寫完則進行步驟 9，否則進行步驟 5。
- 步驟 8：主機呼叫 {SW_I2C_Send_Stop()} 發送停止訊號，返回寫入失敗提示 0xff，寫入結束。
- 步驟 9：主機呼叫 {SW_I2C_Send_Stop()} 發送停止訊號，返回寫入成功提示 0x01，寫入結束。

讀取數據流程圖

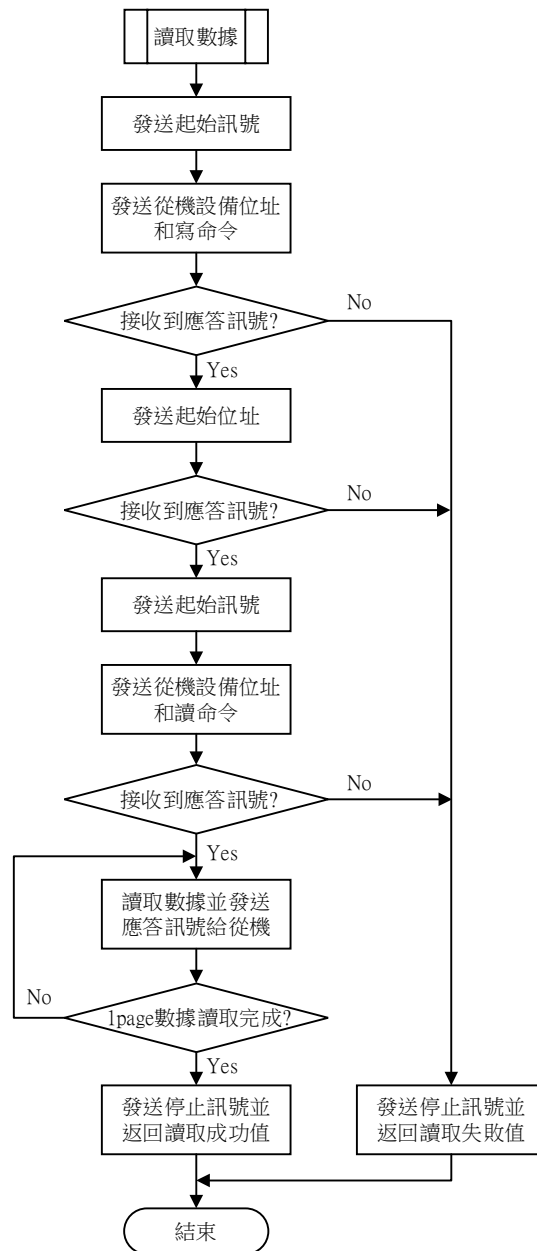


圖 11. 讀取數據流程圖

讀取數據說明

- 步驟 1：主機呼叫 {SW_I2C_Send_Start()} 發送起始訊號，然後呼叫 {SW_I2C_Send_Addr(0x50, TX_Mode)} 發送從機設備位址和寫命令，等待從機的應答。
- 步驟 2：若從機有應答 (ACK=0)，則進行下一個步驟，若無應答 (ACK=1)，則轉跳到步驟 8。
- 步驟 3：主機呼叫 {SW_I2C_Send_Data(ReadAddr)} 發送 Page0 首位址，等待從機應答。
- 步驟 4：若從機有應答 (ACK=0)，則進行下一個步驟，若無應答 (ACK=1)，則轉跳到步驟 8。

- 步驟 5：主機呼叫 {SW_I2C_Send_Start()} 發送起始訊號，然後呼叫 {SW_I2C_Send_Addr(0x50,RX_Mode)} 發送從機設備位址和讀命令，等待從機的應答。
- 步驟 6：若從機有應答 (ACK=0)，則進行下一個步驟，若無應答 (ACK=1)，則轉跳到步驟 8。
- 步驟 7：主機連續讀取 1page 數據，存放於數組 Receive_buffer 中，爾後主機呼叫 {SW_I2C_Send_Stop()} 發送停止訊號，返回讀取成功提示 0x01，讀取結束。
- 步驟 8：主機呼叫 {SW_I2C_Send_Stop()} 發送停止訊號，返回讀取失敗提示 0xff，讀取結束。

函數說明

本文提供之範例程式主要有以下 3 個函數，具體實現過程可查看附檔的 HT8_SW_I2C_Master.c 文檔以及 I2C_TO_EEPROM.c 文檔。

函數名	u8 I2C_EEPROM_PageWrite(u8 *Buffer_data, u8 WriteAddr, u8 NumByteToWrite)
功能	對指定頁寫入數據
輸入	Buffer_data：發送數據的 Buffer 指針
	WriteAddr：頁位址
	NumByteToWrite：發送數據的位元組數
輸出	0x01：寫入成功提示；0xff：寫入失敗提示

表 2. I2C_EEPROM_PageWrite

函數名	u8 I2C_EEPROM_BufferWrite(u8 *Buffer_data, u8 WriteAddr, u16 NumByteToWrite)
功能	從指定位址開始寫入數據
輸入	Buffer_data：發送數據的 Buffer 指針
	WriteAddr：寫入的首位址
	NumByteToWrite：發送數據的位元組數
輸出	0x01：寫入成功提示；0xff：寫入失敗提示

表 3. I2C_EEPROM_BufferWrite

函數名	u8 I2C_EEPROM_BufferRead(u8* Receive_buffer, u8 ReadAddr, u16 NumByteToRead)
功能	從指定位址開始讀取數據
輸入	Receive_buffer：讀取發送數據的 Buffer 指針
	ReadAddr：讀取的首位址
	NumByteToRead：發送數據的位元組數
輸出	0x01：讀取成功提示；0xff：讀取失敗提示

表 4. I2C_EEPROM_BufferRead

應用注意事項

在使用 I²C 總線進行通訊時，請注意以下事項。

- 從機位址

在設定 HT8 MCU 的從機位址時需要注意：從機位址暫存器 SIMA 的有效位元為高七位元，其暫存器映射表如下。

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

- 提升電阻選擇

大部分 I²C 介面都具有開漏特性，這就要求 I²C 總線需要外加提升電阻，其結構如下圖。

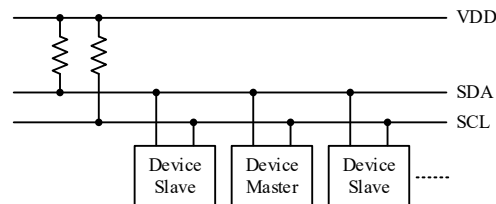


圖 12. I²C 主從總線連接圖

該提升電阻的阻值也有一定的要求，一般其最小值計算公式如下：

$$R_P(\min) = \frac{(V_{DD} - V_{OL}(\max))}{I_{OL}}$$

其中， V_{DD} 為 I/O 埠電源電壓， $V_{OL}(\max)$ 為總線輸出低準位時的最大電壓值（一般為 0.4V）， I_{OL} 為 I/O 埠灌電流，該公式的參數因 MCU 的不同而略有差別，使用前請參考相應的 Datasheet。

HT8 MCU 中與 I²C 共用的 I/O Port 一般都具有提升功能，使用者可參考對應 I/O 的提升電阻值，根據實際應用情況選用 I/O 內部提升電阻或外接提升電阻。

- 超時功能

I²C 傳輸建議開啟超時功能，可以避免接收異常的時鐘訊號導致傳輸停擺，超時計數器在時鐘線 SCL 下降緣的時候清零，如果在 SCL 下一個下降緣到來前超出設定的超時時間，就會觸發 I²C 電路及其個別相關暫存器重置。超時發生後，I²C 相關暫存器的具體狀態請參考實際選用 MCU 的 Datasheet。

使能超時功能時，需要注意只有在起始訊號以及從機位址匹配時才會開始進行超時計數，停止訊號產生時超時功能停止。

● 去抖功能

I²C 傳輸時建議使用去抖功能，可以有效避免總線訊號毛刺而導致通訊失敗，特別適用於較長時間或較長距離通訊等應用情形。去抖時間需根據 I²C 數據傳輸速度和系統時鐘關係選擇 2 個或 4 個系統時鐘，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)
無去抖時間	f _{sys} > 2MHz	f _{sys} > 4MHz
2 個系統時鐘去抖時間	f _{sys} > 4MHz	f _{sys} > 8MHz
4 個系統時鐘去抖時間	f _{sys} > 4MHz	f _{sys} > 8MHz

● 避免傳輸鎖死

為避免在 I²C 通訊時發生鎖死現象，除從機應使用超時、去抖功能外，主機讀寫總線時序需注意如下事項。

1. 主機不可以在從機忙碌時發送時鐘訊號。
2. 主機讀寫總線前需判斷總線是否為空閒，且通訊過程應加入超時功能，避免 I²C 發生異常時影響主機執行效率與動作。
3. 當主機使用 GPIO 模擬 SCL/SDA 功能時，若 SCL/SDA 需輸出高準位，應採用 GPIO 設為 Input Mode，利用 GPIO 內部或外部 Pull-High 電阻使 SCL/SDA 為高準位，不建議採用直接設定 GPIO Output Mode 且 Output High。

結論

本文詳細說明了 HT8 MCU I²C 的功能以及工作原理，進一步羅列了應用中需要注意的事項，並提供 I²C 主機存取外部 EEPROM 範例，用戶通過閱讀本文可以對 HT8 MCU I²C 介面有全面的認識，提高用戶在使用 I²C 介面時的深入程度。

參考資料

如需進一步瞭解，敬請瀏覽 Holtek 官方網站 www.holtek.com.tw。

版本及修改資訊

日期	作者	發行	說明
2024.01.12	郭勝斌	V1.40	修改 I ² C 電氣特性表格中參數 t _{HSTAY} 值
2023.08.23	趙德銘	V1.30	新增 I ² C 特性說明及從機資料傳輸範例
2023.02.01	楊運祺	V1.20	新增 I ² C 存取外部 EEPROM 範例
2021.11.09	郭家輝	V1.10	增補應用注意事項說明
2021.06.28	郭家輝	V1.00	第一版發行

免責聲明

本網頁所載的所有資料、商標、圖片、連結及其他資料等（以下簡稱「資料」），只供參考之用，盛群半導體股份有限公司及其關聯企業（以下簡稱「本公司」）將會隨時更改資料，並由本公司決定而不作另行通知。雖然本公司已盡力確保本網頁的資料準確性，但本公司並不保證該等資料均為準確無誤。本公司不會對任何錯誤或遺漏承擔責任。

本公司不會對任何人士使用本網頁而引致任何損害（包括但不限於電腦病毒、系統故障、資料損失）承擔任何賠償。本網頁可能會連結至其他機構所提供的網頁，但這些網頁並不是由本公司所控制。本公司不對這些網頁所顯示的內容作出任何保證或承擔任何責任。

責任限制

在任何情況下，本公司並不須就任何人由於直接或間接進入或使用本網站，並就此內容上或任何產品、資訊或服務，而招致的任何損失或損害負任何責任。

管轄法律

本免責聲明受中華民國法律約束，並接受中華民國法院的管轄。

免責聲明更新

本公司保留隨時更新本免責聲明的權利，任何更改於本網站發佈時，立即生效。