

# 如何使用 HT66F40 SIM I<sup>2</sup>C 通訊功能 (C 語言)

文件編碼：HA0286T

## 簡介

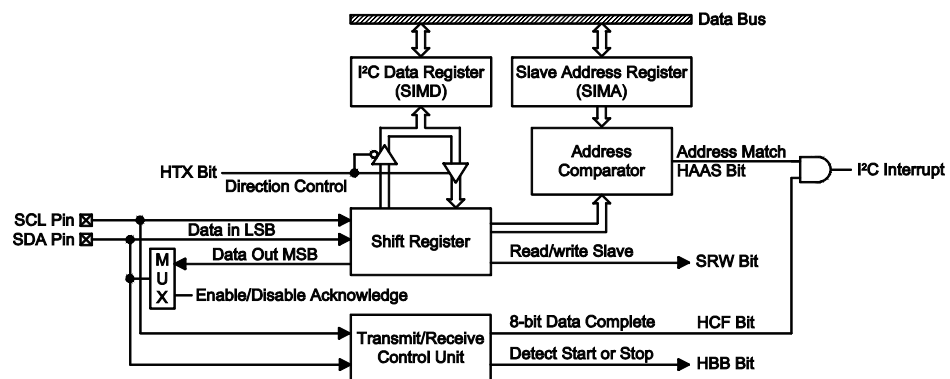
HT66Fx0 系列 Flash MCU 內建有 SIM 功能，其中包括了 SPI 和 I<sup>2</sup>C 這兩種功能，本文以 HT66F40 為母體，介紹使用 I<sup>2</sup>C 進行資料傳輸的方法和注意事項，並通過 C 語言實現 I<sup>2</sup>C 主機與從機的通訊過程。

## I<sup>2</sup>C 介面

I<sup>2</sup>C 是一種用來和傳感器、EEPROM 等外圍設備進行通訊的介面。I<sup>2</sup>C 總線最初是由飛利浦公司開發的兩線串列傳輸介面，用來同步串列傳輸資料。而現在，這種兩線式的傳輸介面，擁有通信協議簡單，可容納多個外圍設備等優點，已經成為一種非常流行的介面，應用於很多領域中。

I<sup>2</sup>C 串列介面是一種兩線式介面，一條是串列資料線 SDA，另外一條是串列時鐘線 SCL。需要注意的是，I<sup>2</sup>C 總線不存在片選線，I<sup>2</sup>C 總線上的每個設備通過唯一的設備位址被識別。兩個設備通過雙向 I<sup>2</sup>C 總線相互通信，一個設備稱為主機，另一個為從機。主機和從機都可以進行資料的發送和接收，但是，主機具有總線過速控制能力。對於在從機模式下運行的微控制器來說，I<sup>2</sup>C 總線有兩種資料傳輸方式，從機發送模式和從機接收模式。

對於 HT66F40 這塊 MCU 來說，有四個與 I<sup>2</sup>C 總線相關暫存器，SIMC0、SIMC1、SIMA、SIMD。I<sup>2</sup>C 總線上的通信需要四步驟完成，一個開始信號、一個從機位址發送、一個資料發送和一個最終的停止信號。



I<sup>2</sup>C Block Diagram

## 操作說明

HT66Fx0 系列 MCU 支援引腳重置功能，因此在使用 I<sup>2</sup>C 功能時，首先需要設定引腳重置功能 PRM0 暫存器，如下圖所示。

### PRM0 Register – HT66F40/HT66F50

Bit	7	6	5	4	3	2	1	0
Name	—	C1XPS0	—	C0XPS0	PDPRM	SIMPS1	SIMPS0	PCKPS
R/W	—	R/W	—	R/W	R/W	R/W	R/W	R/W
POR	—	0	—	0	0	0	0	0

**PDPRM:** PD3~PD0 pin-shared function Pin Remapping Control

0: No change

1: TCK2 on PD0 change to PB6, TP2\_0 on PD1 change to PB7, TCK0 on PD2 change to PD6, TCK1 on PD3 change to PD7 if SIMPS1, SIMPS0=01

**SIMPS1, SIMPS0:** SIM Pin Remapping Control

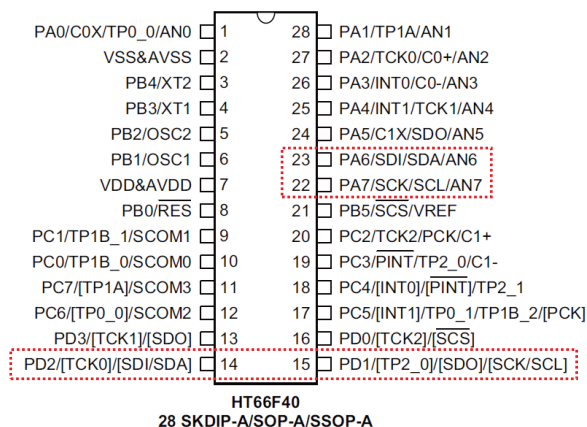
00: SDO on PA5; SDI/SDA on PA6; SCK/SCL on PA7;  $\overline{SCS}$  on PB5

01: SDO on PD3; SDI/SDA on PD2; SCK/SCL on PD1;  $\overline{SCS}$  on PD0

10: SDO on PB6; SDI/SDA on PB7; SCK/SCL on PD6; SCS on PD7

11: Undefined

用戶可根據需要通過設定 PRM0 暫存器中的 SIMPS0 和 SIMPS1 來設定 SDA 和 SCL 的位置。



要打開 I<sup>2</sup>C 功能你必須首先將配置選項的 SIM Function 選擇為 Enable，然後設定好 SIMC0 和 SIMC1 這兩個暫存器相應的位元。下面會對這兩個暫存器詳細的介紹。

HT66F40 與 I<sup>2</sup>C 功能相關的一共有 4 個暫存器。它們是 SIMC0、SIMC1、SIMA 和 SIMD。

## SIMA 暫存器

Bit	7	6	5	4	3	2	1	0
Name	IICA6	IICA5	IICA4	IICA3	IICA2	IICA1	IICA0	—
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" unknown

SIMA 暫存器是 MCU 設定從機的 7 位元位址所使用的暫存器。暫存器的 1~7 位元定義了從機位址，第 0 位元未被使用。當主機設備連接到 I<sup>2</sup>C 總線，將會發出從機位址，如果能和記憶在 SIMA 暫存器的某從機位址匹配時，該從機設備將會被選中。

## SIMD 暫存器

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" unknown

SIMD 暫存器用來記憶馬上要傳輸或者剛接收到的資料。它是和 SPI 共用的，在 HT66F40 中，SPI 與 I<sup>2</sup>C 只能二選其一。在資料寫入 I<sup>2</sup>C 之前，資料必須要放入 SIMD 暫存器才能得到傳輸。相應的，在資料從 I<sup>2</sup>C 接收到後，也只能從 SIMD 暫存器中讀出資料。總之，任何通過 I<sup>2</sup>C 進行傳送和接受的資料都必須通過 SIMD 暫存器。

## SIMC0 暫存器

Bit	7	6	5	4	3	2	1	0
Name	SIM2	SIM1	SIM0	PCKEN	PCKP1	PCKP0	SIMEN	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	—
POR	1	1	1	0	0	0	0	—

SIMC0 暫存器也是 SPI 和 I<sup>2</sup>C 共用的，它用來打開或者關閉串列總線功能，設定 SIM 模式。

**SIMEN**：SIMEN 位元決定 I<sup>2</sup>C 功能的開和關。當 SIMEN 位元被清除為零時，將關閉掉 I<sup>2</sup>C 功能，SDA 與 SCL 總線將會處於浮空狀態，I<sup>2</sup>C 工作電流將變為最小值。當這一位元變為 1 時，I<sup>2</sup>C 功能將打開。當然，必須要先在配置選項中打開 SIM 功能，這一位元才會有效。

**SIM0~SIM2**：這幾位元設定 SIM 功能的操作模式，也就是選擇 I<sup>2</sup>C 功能或者選擇 SPI 功能。當 SIM0~SIM2 為 110 時，將選中 I<sup>2</sup>C 功能，且 HT66Fx0 系列 MCU 只能工作在 I<sup>2</sup>C 從機模式。

### **SIM2, SIM1, SIM0: SIM Operating Mode Control**

000: SPI master mode; SPI clock is  $f_{SYS}/4$   
 001: SPI master mode; SPI clock is  $f_{SYS}/16$   
 010: SPI master mode; SPI clock is  $f_{SYS}/64$   
 011: SPI master mode; SPI clock is  $f_{TBC}$   
 100: SPI master mode; SPI clock is TM0 CCRP match frequency/2  
 101: SPI slave mode  
 110: I<sup>2</sup>C slave mode  
 111: Unused mode

## SIMC1 暫存器

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

SIMC1 暫存器是一個 I<sup>2</sup>C 專用的暫存器。

**RXAK**：RXAK 是從機設備接收到的應答旗標。RXAK 為 0 時，表示接收設備接收 8 個位元的資料後，在第 9 個 SCL 時鐘發出的應答信號已被發送設備收到。當從機設備處於發送模式，主機設備處於接收模式時，從機設備通過檢查 RXAK 位元來確定主機設備是否要接收下一個字節的資料，RXAK 為 0，表示主機設備要繼續接收資料，從機設備將發送下一個位元組的資料，RXAK 為 1，表示主機設備不再接收資料，從機設備將釋放 SDA 總線，主機設備將發送停止信號，釋放 I<sup>2</sup>C 總線。

**IAMWU**：此位元用來控制 I<sup>2</sup>C 位址匹配是否可以喚醒 MCU，1 為是，0 為否。

**SRW**：SRW 是一個從機的讀取/寫入模式位元。這一位元決定了主機是希望發送或者接收資料。當主機傳送的位址和從機匹配後，HAAS 位元將被設為高準位，設備將會檢查 SRW 位元來決定是應該進入發送模式或者是接收模式。如果 SRW 位元為高準位，主機會請求從 I<sup>2</sup>C 總線上讀取資料，此時從機設備為發送模式。反之，主機會寫入資料到總線上，從機為接收模式。

**TXAK**：TXAK 是從機設備發送出去的應答旗標。當從機設備處於接收模式，主機設備處於發送設備時，從機設備接收 8 個位元的資料後，在第 9 個 SCL 時鐘把 TXAK 位元作為應答信號發送給主機設備。當 TXAK 設定為 0 時，表示從機設備要接收下一個位元組的資料，TXAK 設定為 1 時，表示從機設備不再接收資料。

**HTX**：HTX 是從機的讀取/寫入模式標準位元。這一位元被設為高為發送模式，設為低為接收模式。

**HBB**：HBB 是 I<sup>2</sup>C 總線繁忙旗標。HBB 變為 1 時，表示從機設備檢測到主機設備發送的開始信號，總線處於繁忙狀態。HBB 變為 0 時，表示從機檢測到主機發送的停止信號，總線處於空閒狀態。

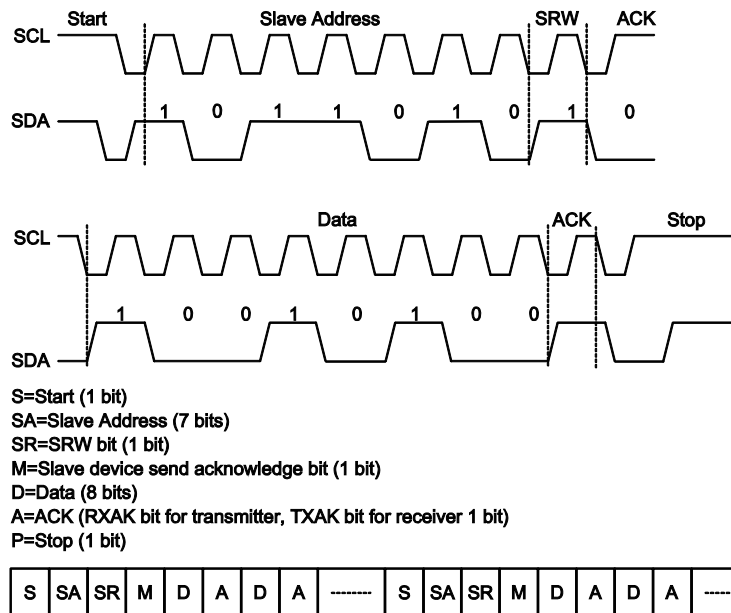
**HAAS**：HAAS 是一個位址匹配旗標。這個標準位元由從機設備的位址是否同主機發送的位址相同決定。如果位址匹配這一位會為高，不匹配則為低。

**HCF**：HCF 是一個資料傳送旗標。當資料正被傳送時，它將被設定為 0。當傳送完成後，這一位會變為 1，同時將產生一個中斷。

## I<sup>2</sup>C 總線通信

I<sup>2</sup>C 總線上的通信需要四步驟完成，一個開始信號、一個從元件位址發送、一個資料發送和一個最終的停止信號。當開始信號被寫入 I<sup>2</sup>C 總線時，所有的總線上的元件都會接收信號並且被通知總線上會立即有位址資料到達。資料的前 7 位元是從元件位址，高位元在前，低位元在後。如果從元件位址匹配，MCU 會將 HAAS 設定為 1，同時產生 I<sup>2</sup>C 總線中斷。進入中斷服務程式後，系統要檢測 HAAS 位元，以確定 I<sup>2</sup>C 總線中斷是來自從機位址匹配，還是來自 8 位元資料傳送完畢。HAAS 為 1 時，是位址匹配引起 I<sup>2</sup>C 中斷，HAAS 為 0 時，是 8 個位元的資料傳輸完畢引起的 I<sup>2</sup>C 中斷。在資料傳遞中，需要注意的是，在 7 位元從機位址被發送後，接下來的一位，即第 8 位元，是讀取/寫入控制位，該位元的值會反映到 SRW。從元件通過檢測 SRW 以確定從機是要進入發送模式還是接收模式。

I<sup>2</sup>C 的時序圖如下圖所示：



I<sup>2</sup>C Communication Timing Diagram

### 開始信號

開始信號只能由連接 I<sup>2</sup>C 總線的主機設備產生，總線上的所有元件必須偵測開始信號，當檢測到後，表明 I<sup>2</sup>C 總線處於忙狀態，從機設備會設定 HBB 位元。開始信號是指在 SCL 為高時，SDA 發生準位從高到低的變化。

### 從機位址

總線上的所有元件都會偵測由主機發起的開始信號的發送。發送開始信號後，主機必須發送從機位址以選擇要進行資料傳輸的從機設備。所有在 I<sup>2</sup>C 總線上的從機設備都會接收到這個從機位址(7 位元)，並與各自內部的從機位址進行比較。如果位址匹配，該從機會產生一個中斷，並將接下來的一位元資料 (即第 8 位元) 保存到 SRW 位元，並發出一個應答信號，即第 9 位元的低準位信號。當從元件位址匹配時，還會設定位元狀態旗標 (HAAS)。

由於 I<sup>2</sup>C 總線有兩個中斷源，在中斷服務副程式中，通過檢測 HAAS 位元可以確定 I<sup>2</sup>C 總線中斷是來自從機位址匹配，還是來自 8 個位元資料傳送完畢。當是位址匹配引起的中斷時，則從機必定是用於設定發送模式或是接收模式，所以必須寫入資料到 SIMD 或從 SIMD 虛擬讀取資料以釋放 SCL。

### SRW 位元

SIMC1 暫存器的 SRW 位元表示主控制器是要從 I<sup>2</sup>C 總線上讀取資料還是要將資料寫到 I<sup>2</sup>C 總線上。從機設備則通過檢測該位以確定自己是作為發送設備還是接收設備。SRW 為 1 時，表示主機設備要從 I<sup>2</sup>C 總線讀資料，從機設備必須將資料寫到 I<sup>2</sup>C 總線，即從機設備做為發送器；SRW 為 0 時，表示主機設備要寫資料到 I<sup>2</sup>C 總線，從機設備要從總線讀取資料，即從機設備做為接收器。

### 應答位元

在主控制器發送呼叫位址後，當從機內部位址與其匹配時，會發送一個應答信號。應答信號會通知主控制器從元件已經接收了呼叫位址。如果沒有應答信號，主控制器必須發送停止 (STOP) 信號以結束通訊。當從機的 SIMC1 暫存器的第 6 位元 (HAAS) 是高時，表示位址匹配，則從機需檢查 SRW，以確定自己是作為發送器還是作為接收器。如果 SRW 位元為高，從機設定成發送器，這樣需要將 SIMC1 暫存器的 HTX 位元設定為 1。如果 SRW 位元為低，從機設定成接收器，這樣會將 SIMC1 暫存器的 HTX 位元清除為零。

### 資料位元組

在從機發出應答信號後，就會進行資料傳輸，一個資料長度為 8 位元，高位在前，低位元在後。接收器在接收到資料後會發出一個應答信號 ("0") 以繼續接收下一個資料。如果發送器沒有檢測到應答信號，發送器將釋放 SDA 線，同時，主機將發出 STOP 信號以釋放 I<sup>2</sup>C 總線。如果從機設定成發送器，從機必須將資料寫到 SIMD；如果從機設定成接收器，從機必須從 SIMD 讀取資料。

### 接收和發送應答位元

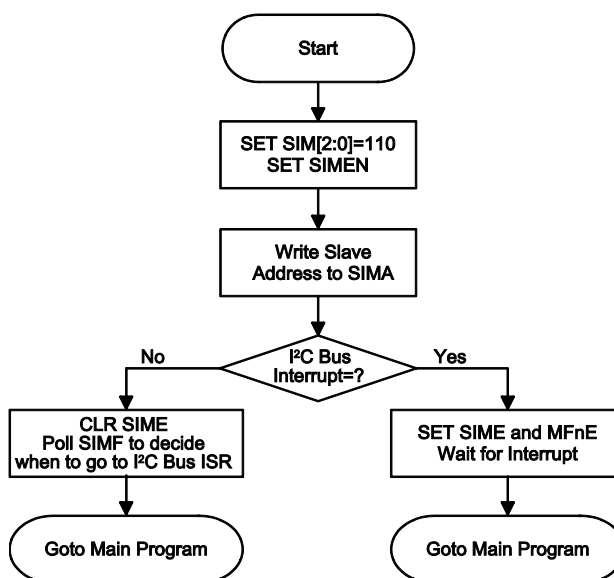
當從機作為接收器，想要繼續接收下一個資料時，必須在第 9 個時鐘發出應答信號 (TXAK)。當從機作為發送器時，檢測應答信號 (RXAK) 以決定是繼續寫資料到 I<sup>2</sup>C 總線，還是改變為接收模式並虛讀 SIMD 暫存器以釋放 SDA 線，同時主機發出停止信號。

## S/W 流程圖

在 I<sup>2</sup>C 總線開始傳送資料前，需要先初始化 I<sup>2</sup>C 總線，在初始化 I<sup>2</sup>C 總線時必須注意以下幾點：

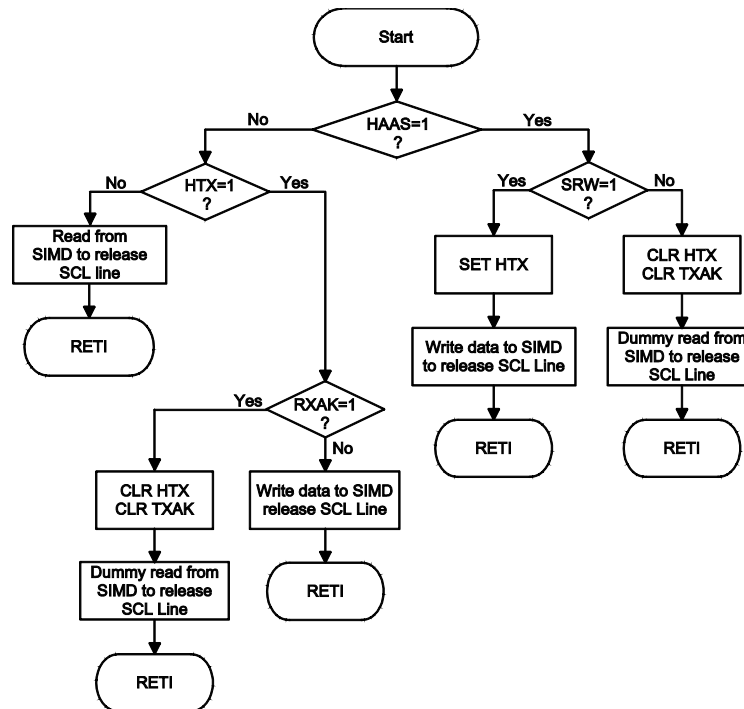
- 步驟 1  
向 SIMA 暫存器寫入從機位址。
- 步驟 2  
將 I<sup>2</sup>C 總線控制暫存器 SIMC0 的 SIMEN 位元設定為 1，以打開 I<sup>2</sup>C 總線。
- 步驟 3  
將 I<sup>2</sup>C 總線中斷控制暫存器 (INTC2) 的 MF2E 位元設定為 1，以允許 I<sup>2</sup>C 總線中斷。如果不需要中斷，就需要在程式中不斷的檢查 HCF 位元，用來檢測 I<sup>2</sup>C 是否有資料傳輸完成，這種方法不推薦。

具體過程如下圖所示：



I<sup>2</sup>C Bus Initialisation Flow Chart

需要注意的是，當 I<sup>2</sup>C 總線中斷被關掉時，必須去不斷的檢查 HCF 位元來確定 I<sup>2</sup>C 有無資料傳輸完成。



I<sup>2</sup>C Bus ISR Flow Chart

如上圖所示，這是一個 I<sup>2</sup>C 的中斷服務程式的流程圖，當進入中斷服務程式後，會首先檢查 HAAS 位元。通過檢測 HAAS 位元可以確定 I<sup>2</sup>C 總線中斷是來自從元件位址匹配，還是來自 8 位元資料傳送完畢。

如果是位址匹配者，會檢查 SRW，以確定主機是希望發送資料還是接受資料。根據此為設定好從機此時的狀態，還必須寫入資料到 SIMD 或從 SIMD 虛擬讀取資料以釋放 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個資料。如果沒檢測到應答信號 (RXAK=1)，從機要從 SIMD 虛擬讀取資料以釋放 SDA 線，同時主控制器將發出 STOP 信號以釋放 I<sup>2</sup>C 總線。



## 程式說明

本範例是 HT66F40 作為從機的 I<sup>2</sup>C 程式，由主程式以及 I<sup>2</sup>C 的中斷服務程式 2 部分組成，在主程式中，會對 MCU 進行初始化操作，主要是對 I<sup>2</sup>C 的相關暫存器進行初始化操作，要注意的是因為使用 PA 埠的其他功能會對 I<sup>2</sup>C 的使用造成影響，在初始化中必須關閉這些功能，這些功能包括 CP 和 AD 轉換。

主循環中通過 PA0 埠的按鍵掃描來更改 Slave 要發送的數值 tx\_data，當需要進行 Slave 發送資料時把資料發送出去。

I<sup>2</sup>C 的中斷服務程式 I2C\_int 中，Slave 發送時會將 tx\_data 的資料發送到 I<sup>2</sup>C 總線，接收時會將從 I<sup>2</sup>C 總線上接收到的資料送到 rx\_data 中，在範例程式中，把 rx\_data 中的值傳給 PC 埠，可以通過在 PC 埠接 LED 燈觀察資料的傳輸，用戶可以自行修改和進行資料的後續操作。

配置選項中，要使用 I<sup>2</sup>C 總線，必須將 SIM Function 設為 Enable，其他選項由用戶使用情況決定。

如下範例代碼為驗證 I<sup>2</sup>C 功能所使用的主機及從機程式，主機也使用 HT66F40，並利用 I/O（PA6:SDA，PA7:SCL）類比的方法實現 I<sup>2</sup>C 主機，程式中通過 PA0 按鍵掃描來設定主機是通過 I<sup>2</sup>C 發送或接受資料，當主機處於發送資料的狀態時通過 PA1 的按鍵掃描來改變發送的資料。當主機處於接收狀態時資料可以通過主機的 PC 埠輸出，具體程式的實現方法用戶可參考附件。

## 程式範例

```
//-----
//main.c 主函數文件
//-----
#include "macro.h"
/** VECTOR MAPPING *****/
#pragma vector I2C_Int @ 0x20 //interrupt vector declare
/** DECLARATIONS *****/
//*****
void key_scan(void);
void init_ram(void) //initial ram
{
    _mp0=0x80;
    for(_tblp=0;_tblp<C_Ram0Number;_tblp++)
    {
        _bp = 0;
        _iar0=00;
        _mp0++;
    }
    _mp1=0x80;
    for(_tblp=0;_tblp<C_Ram1Number;_tblp++)
    {
        _bp = 1;
        _iar1=00;
        _mp1++;
    }
}
//*****
```

```

void init_port(void)                //initial IO
{
    _pac = 0xff;
    _pbc = 0xff;
    _pdc = 0x00;
    _pec = 0xff;
    _pfc = 0xff;
    _papu = 0xff;
    _pbpu = 0xff;
    _pcpu = 0xff;
    _pdpu = 0xff;
    _pepu = 0xff;
    _pfpu = 0xff;

    _pcc = 0;                       //PC as output
    _pc = 0;

    _adcr1 = 0;
    _adoff = 1;                     //close AD
    _acerl = 0;

    _cp0c = 0;
    _cp1c = 0;                     //close cp

    _hlclk = 1;                     //system clock=fH

    _intc0 = 0;
    _intc1 = 0;
    _intc2 = 0;                     //close interrupt
}
//*****
void init_i2c(void)                 //initial I2C
{
    _simps0 = 0;
    _simps1 = 0;                   //simps0:simps1=00: SDA on PA6,SCL on PA7

    _emi = 1;
    _mf2e = 1;
    _sime = 1;                     //enable sim interrupt

    _sima = C_DeviceAddr;          //set i2c slave address 0x14
    _simc0 = 0xc2;                 //enable i2c,i2c slave mode
}
void I2C_Int(void)
{
    if(_haas)
    {
        // _haas=1,address match trig interrupt
        if(_srw)
        {
            //srw=1:slave in transfer mode
            _htx = 1;
            _simd = tx_data;
        }
        else
        {
            //srw=0:slave in receive mode
            _htx = 0;
            _txak = 0;
            _acc = _simd;           //dummy read from simd to release scl line
        }
    }
    else
    {
        // _haas=0,data trig interrupt
    }
}

```

```

if(_htx)
{
    //htx=1:slave in write state;
    if(_rxak)
    {
        //rxak=1:master stop receiving next byte, master releases scl bus
        _htx = 0;
        _txak = 0;
        _acc = _simd;    //dummy read from simd to release scl line
    }
    else
    {
        //rxak=0:master wants to receive next byte;
        _simd = tx_data;
    }
}
else
{
    //htx=0:slave in read state
    rx_data = _simd;
    //
    _pc = rx_data;    //receive data show by pc port
    _txak = 0;
}
}
_simf = 0;    //clr i2c interrupt flag
}
// Main program entry
void main()
{
    init_ram();
    init_port();
    init_i2c();
    while(1)
    {
        _clrwdt1();
        _clrwdt2();
        _pd++;    //for test ic is working
        key_scan();
        _pc = rx_data;    //receive data show by pc port
    }
}
void key_scan()
{
    if(key1==0)
    {
        _delay(10000);    //delay 10ms
        if(key1==0)
        {
            f_key1_down = 1;
        }
    }

    if(f_key1_down)
    {
        _delay(10000);
        if(key1)
        {
            f_key1_down = 0;
            tx_data++;
        }
    }
}
//-----
//macro.h 參數變量定義文件
//-----

```

```
// BYTE type definition
#include "HT66F40.h"
#ifndef _BYTE_DEF_
#define _BYTE_DEF_
//*****//
typedef unsigned char      uchar;      //8bit
typedef unsigned int uint16;      //16bit
typedef unsigned long      uint32;      //32bit
//*****//
//const define
#define C_RamAddress      0x80      //;;RAM start address
#define C_Ram0Number      0x80      //;;RAM=128
#define C_Ram1Number      0x40      //;;RAM=64
#define C_DeviceAddr      0x14      //;;I2C Slave address
//*****//
#define key1      1a_0      //pa0
//*****//
uchar tx_data,rx_data;

//*****//
//bit type must be define in rambank0
#pragma rambank0
uchar Flag1 @ 0xfe;
#define f_key1_down      ff_0
#pragma norambank
//*****//
#endif
```

## 結論

本文講解了 HT66F40 的 I<sup>2</sup>C 相關的操作與注意事項，本範例進行 I<sup>2</sup>C 傳輸正常，用戶可以直接使用 I<sup>2</sup>C 的中斷服務程式。