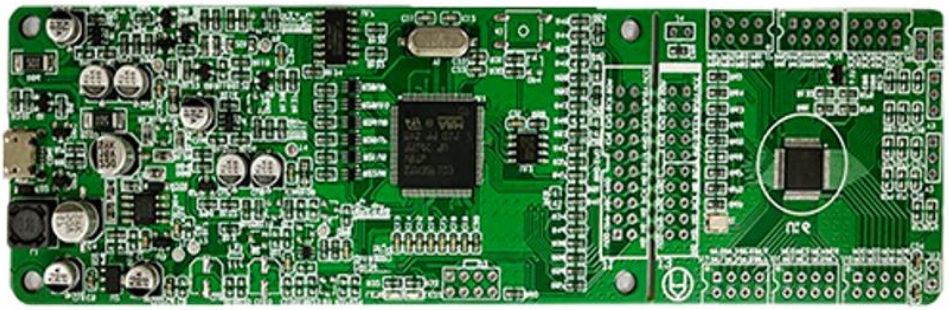
简介

OTP 调试上位机 HC-IDE

- ASM 和 C 代码编辑功能
- ASM 和 C 代码编译功能（XP 系统不支持 C 编译）
- 仿真功能
- 烧录功能
- 固件更新功能

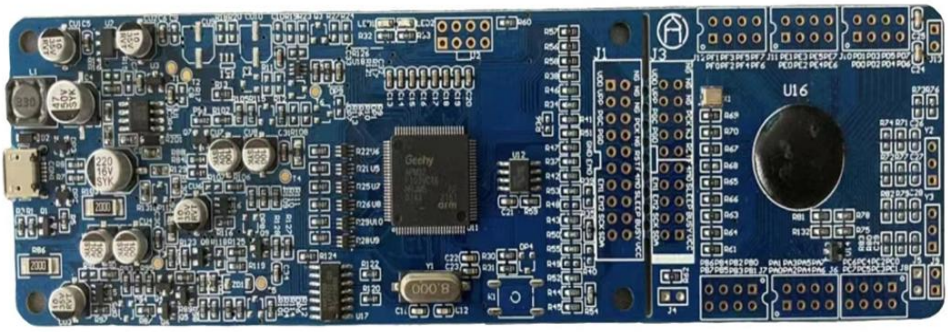
OTP 调试下位机 HC-ICD-V4

- 支持在线仿真(Support On-line Simulation)
- 支持在线烧录(Support On-line Programming)
- 支持固件升级(Support Firmware Update)
- USB 供电(USB Power Supply)



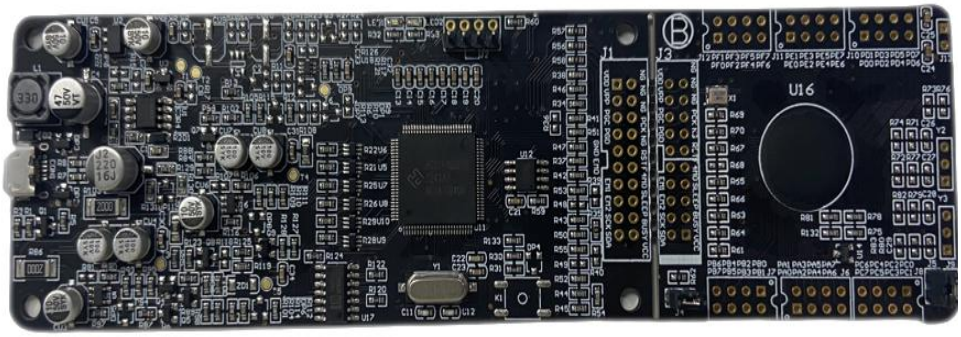
绿板支持芯片：

* HC16P013A0
* HC16P015B0
* HC18P015B0
* HC16P122B1
* HC18P122B1



蓝板支持芯片：

* HC16P122B1
* HC18P122B1
* HC16P015A0
* HC18P015A0
* HC16P015B0
* HC18P015B0
* HC18P13XL(A)
* HC18P23XL
* HC18P110X0



黑板支持芯片：

* HC16P122B1
* HC18P122B1
* HC16P015A0
* HC18P015A0
* HC16P015B0
* HC18P015B0
* HC18P13XL(A)
* HC18P23XL
* HC18P110X0

支持芯片：

* HC18P13XL
* HC18P23XL

HC-ICD-V4 产品实物图

注意事项

1、如果编程过程中出现编译缓慢,同一个工程其他电脑可编译,此电脑编译失败的情况,请关闭 360 等杀毒软

件再重新进行编译。

绿板注意事项

1、仿真器 HC16P122B1 的 2/4V 的 AD 参考电压与规格书描述相反。

蓝板注意事项

1、HC18P015A0 型号在蓝板仿真时，T0 计数器不计数，标志位无法置 1。客户开发仿真时需注意该事项，烧录芯片正常。

2、HC18P015A0 型号在蓝板仿真时，RAM 地址 0X64/0X66 数据出错，客户开发仿真时需要注意该事项，烧录芯片正常。

3、HC18P015A0 型号在蓝板仿真时，段点打在 CALL 和 GOTO 指令上时，先全速运行到段点位置再单步执行，光标会跳到下一行再进入函数，客户开发仿真时需要注意该事项，烧录芯片正常。

4、HC18P015A0 型号在蓝板仿真时，查表 PCL 指令调用错误，客户开发仿真时需要注意该事项，烧录芯片正常。

5、HC18P015A0 型号在蓝板仿真时，无上下拉功能，客户开发仿真时需要注意该事项，烧录芯片正常。

6、HC18P110B0 型号在蓝板仿真时，A0 口无功能，客户开发仿真时需要注意该事项，烧录芯片正常。

7、HC18P110B0、HC18P122B1、HC18P134L、HC18P235L 型号在蓝板仿真时，ADC 无论选择内参还是外参，无论选几 V，参考电压一直保持 2V 不变，客户开发仿真时需要注意该事项，烧录芯片正常。

8、HC18P235L、HC18P015A0 型号在蓝板仿真时，单步执行 BTFSS 和 BTFSC 不跳步，客户开发仿真时需要注意该事项，烧录芯片正常。

黑板注意事项

1、HC18P015A0 型号在黑板仿真时，单步执行 BTFSS 和 BTFSC 不跳步。客户开发仿真时需注意该事项，烧录芯片正常。

2、HC18P015A0 型号在黑板仿真时，RAM 地址 0X6E/0X66 数据出错，客户开发仿真时需要注意该事项，烧录芯片正常。

3、HC18P015A0、HC18P015B0、HC18P235L 型号在黑板仿真时，段点打在 CALL 和 GOTO 指令上时，先全速运行到段点位置再全速或单步执行，可能出现无法跳转的情况，客户开发仿真时需要注意该事项，烧录芯片正常。

4、HC18P015A0 型号在黑板仿真时，查表 PCL 指令调用错误，客户开发仿真时需要注意该事项，烧录芯片正常。

5、HC18P015A0 型号在黑板仿真时，无上下拉功能，客户开发仿真时需要注意该事项，烧录芯片正常。

6、HC18P015A0 型号在黑板仿真时，外部中断会跳入复位，是由于无上拉功能导致，客户开发仿真时需要注意该事项，烧录芯片正常。

7、HC18P015A0 型号在黑板仿真时，T0 只能使用内部 CPU 时钟，无法分频，无法使用外部时钟，客户开发仿真时需要注意该事项，烧录芯片正常。

8、HC18P015A0 型号在黑板仿真时，T0 实现中断需要先清零 T0IF 再使能 T0IE，客户开发仿真时需要注意该事项，烧录芯片正常。

9、HC18P015B0 型号在黑板仿真时，在线读取 SFR 寄存器信息，读取不到 0fh 地址 INTFLAG 寄存器的 bit3 位，CMPF 的数据，无论程序对它是否置 1，读出都为 0，但不影响中断响应，客户开发仿真时需要注意该事项，烧录芯片正常。

10、HC18P110B0 型号在黑板仿真时，PA0 为施密特端口，客户开发仿真时需要注意该事项，烧录芯片正常。

11、HC18P134L 型号在黑板仿真时，ADC 无法选择外参，客户开发仿真时需要注意该事项，烧录芯片正常。

仿真板烧录注意事项

1、仿真器烧录时仅支持 IRC 校准，其他校准需取消勾选。

2、仿真器烧录文件自检失败，红灯持续闪烁，需重启上电仿真板后重新下载烧录。

芯片仿真替代说明

仿真型号	替代型号	备注
HC16P100B1	HC16P122B1	
HC18P010L	HC16P015B0	
HC18P020L	HC16P015B0	HC18P020L 选择 HC16P015B0 仿真 CODE 1K 以内大小
HC18P023L	HC18P13XL(A)	
SQ015L	HC16P015B0	
SQ013L	HC16P015A0	
HC18P110L	HC18P110X0	
HC18P111L	HC18P110X0	
HC18P12XL	HC18P13XL(A)	
HC18P13XL	HC18P13XL(A)	
SQ51xA	HC18P13XL(A)	
SQL581x	HC18P110X0	

目录

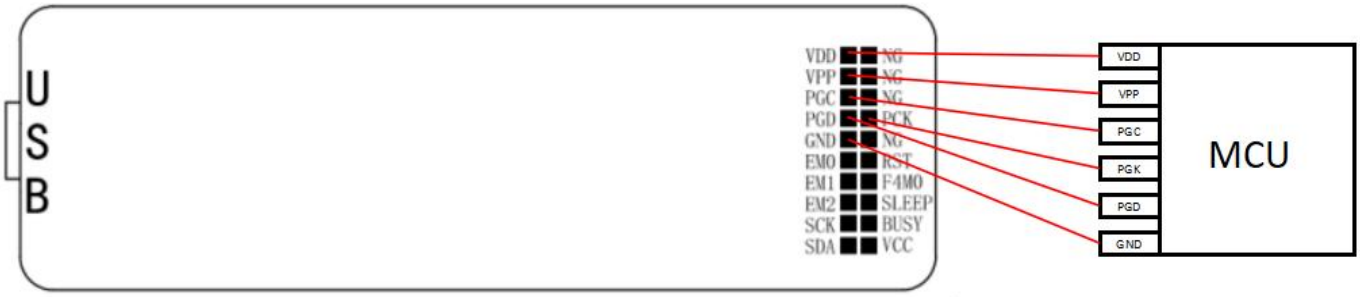
1 软件安装

请参考《TL0001_驱动安装手册》和《TL0101_OTP 调试_HC-ICD-V4_安装手册》。

2 硬件连接

图 2-1 HC-ICD-V4 主板引脚图

仿真引脚：
GND, EM0, EM1, EM2, SCK, SDA, RST, F4M0, SLEEP, BUSY, VCC。



烧录引脚：
VDD, VPP, PGC, PGD, GND, PCK。

HC-ICD-V4 出厂默认将主板与子板连接在一块。烧录时直接将 HC-ICD-V4 的烧录引脚与芯片的烧录引脚相连，无需将两块板子掰开。

客户如果手动将两块板子掰开后，仿真时请用排线或杜邦线将主板与子板相连。

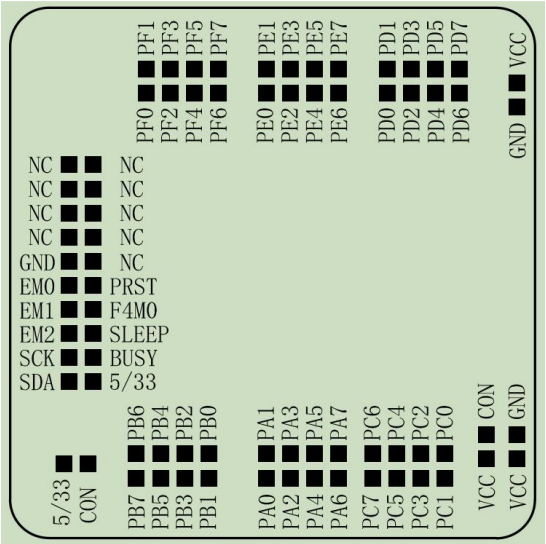


图 2-2 HC-ICD-V4 仿真子板引脚图

3 新建项目

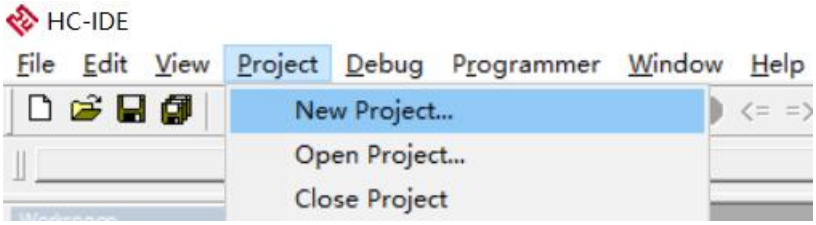


图 3-1 依次点击菜单栏“Project”，“New Project”新建一个项目

图 3-2 点击工具栏“Create Project”按钮新建一个项目

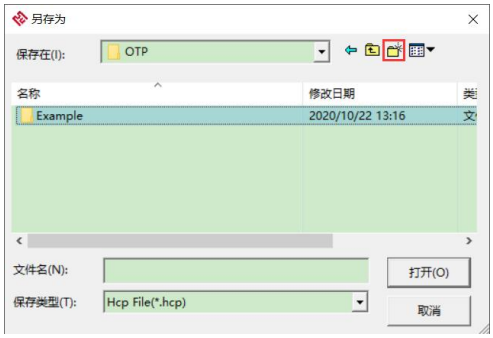


图 3-3 新建项目对话框，点击“创建新文件夹”按钮新建文件夹，注意路径不要有特殊字符

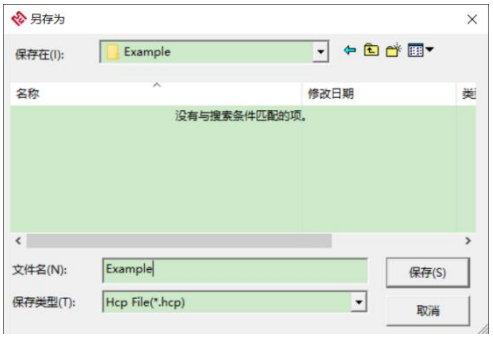


图 3-4 新建项目对话框，进入新建的“Example”文件夹，填写项目名称后点击“保存(S)”按钮

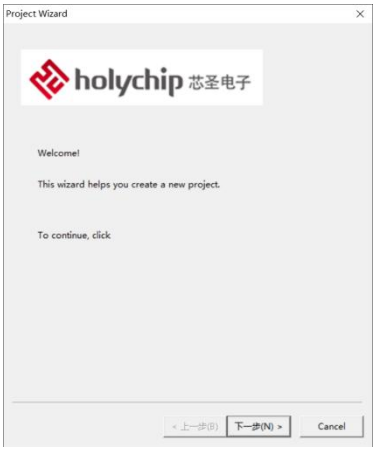


图 3-5 新建项目向导，欢迎界面，点击“下一步(N)”按钮

图 3-6 新建项目向导，选择芯片型号，点击“下一步(N)”按钮

图 3-7 新建项目向导，确认界面，点击“完成”按钮，至此项目新建完成

4 打开/保存/关闭项目

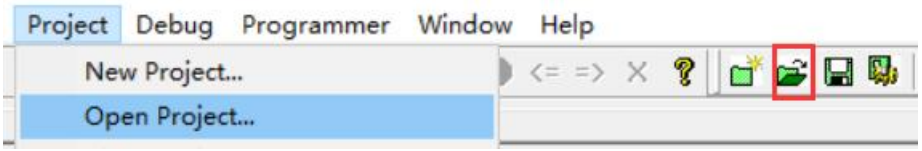


图 4-1 从菜单栏或工具栏打开一个项目

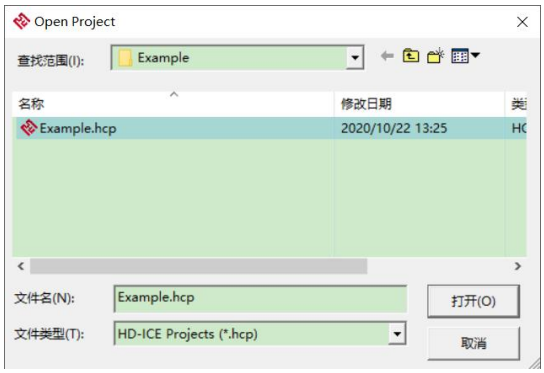


图 4-2 打开项目对话框

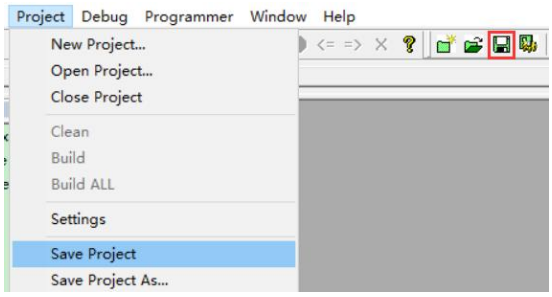


图 4-3 从菜单栏或工具栏保存者另存一个项目

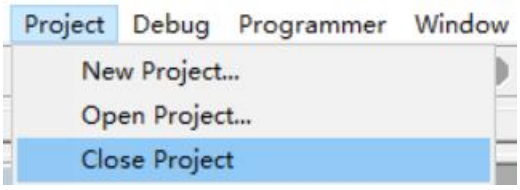


图 4-4 从菜单栏关闭一个项目

5 编辑

5.1 新建文件

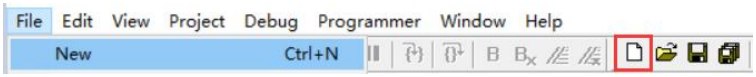


图 5.1-1 从菜单栏或工具栏新建一个文件

5.2 保存文件

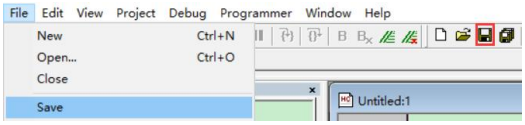


图 5.2-1 从菜单栏或工具栏保存一个文件

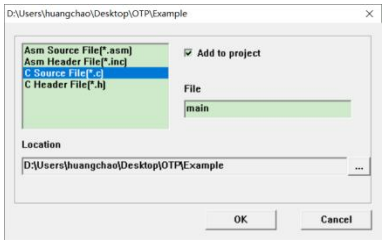


图 5.2-2 文件保存对话框

选择“Asm Source File(*.asm), Asm Header File(*.inc), C Source File(*.c), C Header File(*.h)确定文件类型；选择“Add to project”单选框确定是否将文件添加到项目中；编辑框填写文件名称；点击“OK”按钮完成文件新建。

5.3 添加/删除源文件、头文件、库文件

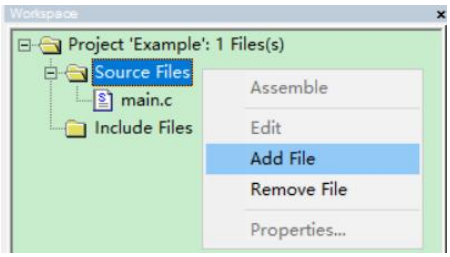


图 5.3-1 “Workspace”窗口右击“Source Files”或“Include Files”，添加或者删除源文件、头文件

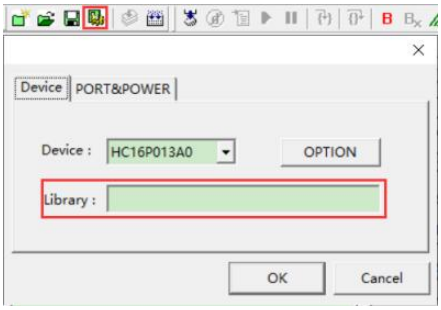


图 5.3-2 设置窗口添加/删除库文件（直接填库文件相对路径）

5.4 指令表

5.4.1 汇编指令表

Field	指令格式		描述	C	DC	Z	周期
移动	MOVWF	F	F←W	-	-	-	1
	MOVF	F, D	D←F(D =0 时为 W, D =1 时为 F)	-	-	√	1
	MOVLW	k	W←k	-	-	-	1
算术	ADDWF	F, D	D←W+F(D =0 时为 W, D =1 时为 F)	√	√	√	1
	ADDLW	k	W←W+k	√	√	√	1
	SUBWF	F, D	D←F-W(D =0 时为 W, D =1 时为 F)	√	√	√	1
	SUBLW	k	W←k -W(D =0 时为 W, D =1 时为 F)	√	√	√	1
	DAW	-	W 寄存器值进行 BCD 调整	√	√	-	1
	INCF	F, D	D←F+1(D =0 时为 W, D =1 时为 F)	-	-	√	1
	DECF	F, D	D←F-1(D =0 时为 W, D =1 时为 F)	-	-	√	1
逻辑	ANDWF	F, D	D←W 与 F(D =0 时为 W, D =1 时为 F)	-	-	√	1
	ANDLW	k	W←W 与 k	-	-	√	1
	IORWF	F, D	D←W 或 F(D =0 时为 W, D =1 时为 F)	-	-	√	1
	IORLW	k	W←W 或 k	-	-	√	1
	XORWF	F, D	D←W 异或 F(D =0 时为 W, D =1 时为 F)	-	-	√	1
	XORLW	k	W←W 异或 k	-	-	√	1
	COMF	F, D	D←F 取反(D =0 时为 W, D =1 时为 F)	-	-	√	1
处理	SWAPF	F, D	D[7:4,3:0]←F[3:0,7:4] (D =0 时为 W, D =1 时为 F)	-	-	-	1
	RRF	F, D	D←F 带进位右移(D =0 时为 W, D =1 时为 F)	√	-	-	1
	RLF	F, D	D←F 带进位左移(D =0 时为 W, D =1 时为 F)	√	-	-	1
	CLRW	-	W←0	-	-	√	1
	CLRF	F	F←0	-	-	√	1
	CLRWDT	-	清零看门狗定时器，影响 TO，PD 位	-	-	-	1
	BCF	F, d	F[d]←0(0≤d≤7)	-	-	-	1
	BSF	F, d	F[d]←1(0≤d≤7)	-	-	-	1
分支	INCFSZ	F, D	D←F+1(D =0 时为 W, D =1 时为 F)，如果 D=0 则跳过下一句	-	-	-	1(2)
	DECFSZ	F, D	D←F-1(D =0 时为 W, D =1 时为 F)，如果 D=0 则跳过下一句	-	-	-	1(2)
	BTFSC	F, d	如果 F[d]=0(0≤d≤7)则跳过下一句	-	-	-	1(2)
	BTFSS	F, d	如果 F[d]=1(0≤d≤7)则跳过下一句	-	-	-	1(2)
	GOTO	k	无条件跳转	-	-	-	2
	CALL	k	调用子程序	-	-	-	2
其他	RETURN	-	从子程序返回	-	-	-	2
	RETFIE	-	从中断返回，并置位 GIE	-	-	-	2
	RETLW	k	W←k，带参数返回	-	-	-	2
	NOP	-	空操作	-	-	-	1
	SLEEP	-	进入待机模式，影响 TO，PD 位	-	-	-	1

5.4.2 伪指令

控制伪指令：控制如何汇编代码	
#define	定义文本替换标号
#include	包含额外的源文件
end	结束程序块
equ	定义一个汇编器常数
org	设置程序起始处
processor	设置处理器类型
条件伪指令：允许汇编符合条件的代码段	
if	开始条件汇编代码块
else	开始 if 条件的备用汇编块
endif	结束条件汇编块
ifdef	如果已经定义了符号则执行
ifndef	如果未定义符号则执行
while	当条件为 TRUE 时执行循环
endw	结束 while 循环
数据伪指令：控制对存储器进行分配，并提供了用符号引用数据项的方法	
cblock	定义常数块
endc	结束自动常数块
da	在程序存储器中存储字符串
db	声明一个字节的数据
dt	定义表
dw	声明一个字的数据
fill	指定程序存储器填充值
res	保留存储器
列表伪指令：控制汇编器列表文件的格	
list	列表选项
宏伪指令：在宏定义内部控制执行和数据分配	
macro	声明宏定义
endm	结束宏定义
local	声明局部宏变量
目标文件伪指令：只有在创建目标文件时使用目标文件伪指令	
banksel	生成存储区选择代码
code	开始目标文件代码段
extern	声明一个外部定义的标号
global	导出标号
idata	开始目标文件已初始化的数据段
pagesel	生成页面选择代码
udata	开始目标文件中未初始化的数据段

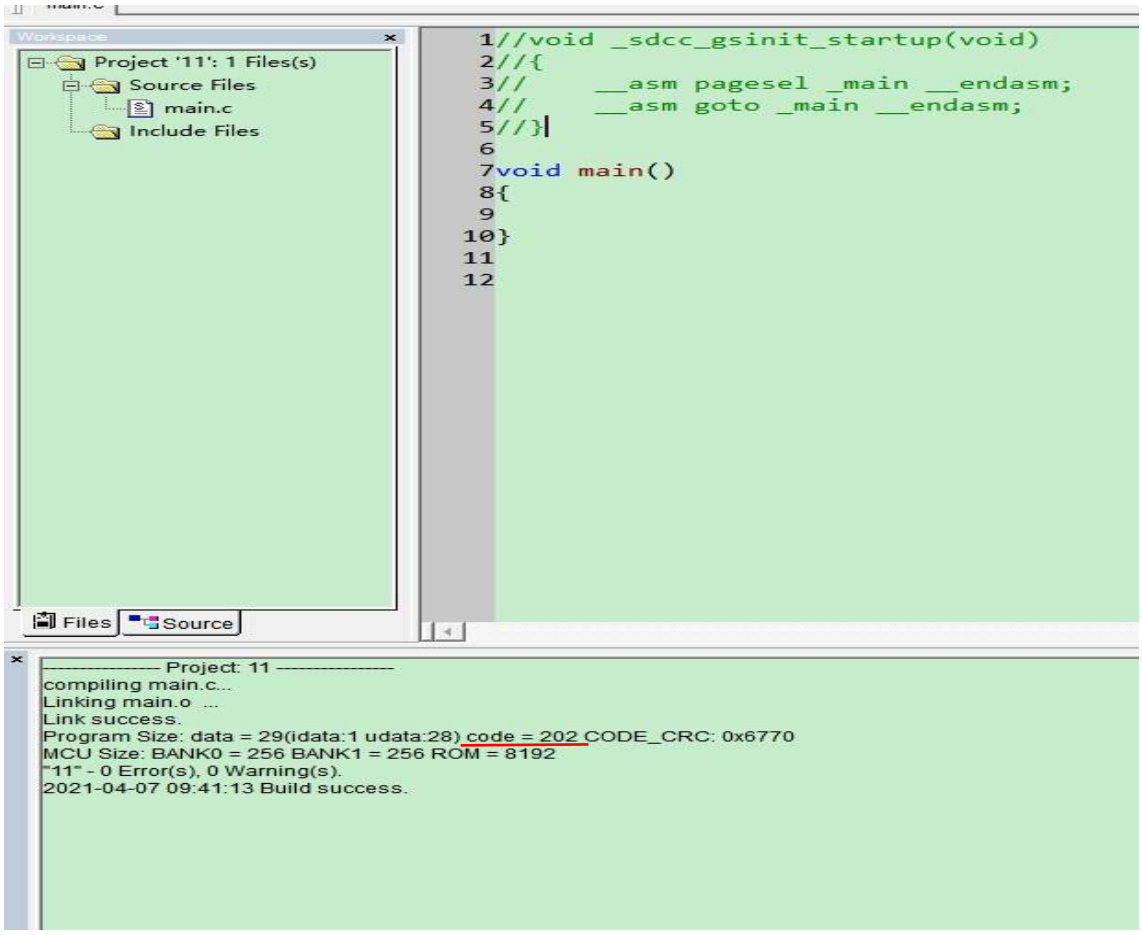
5.5 CODE 优化

5.5.1 优化方式

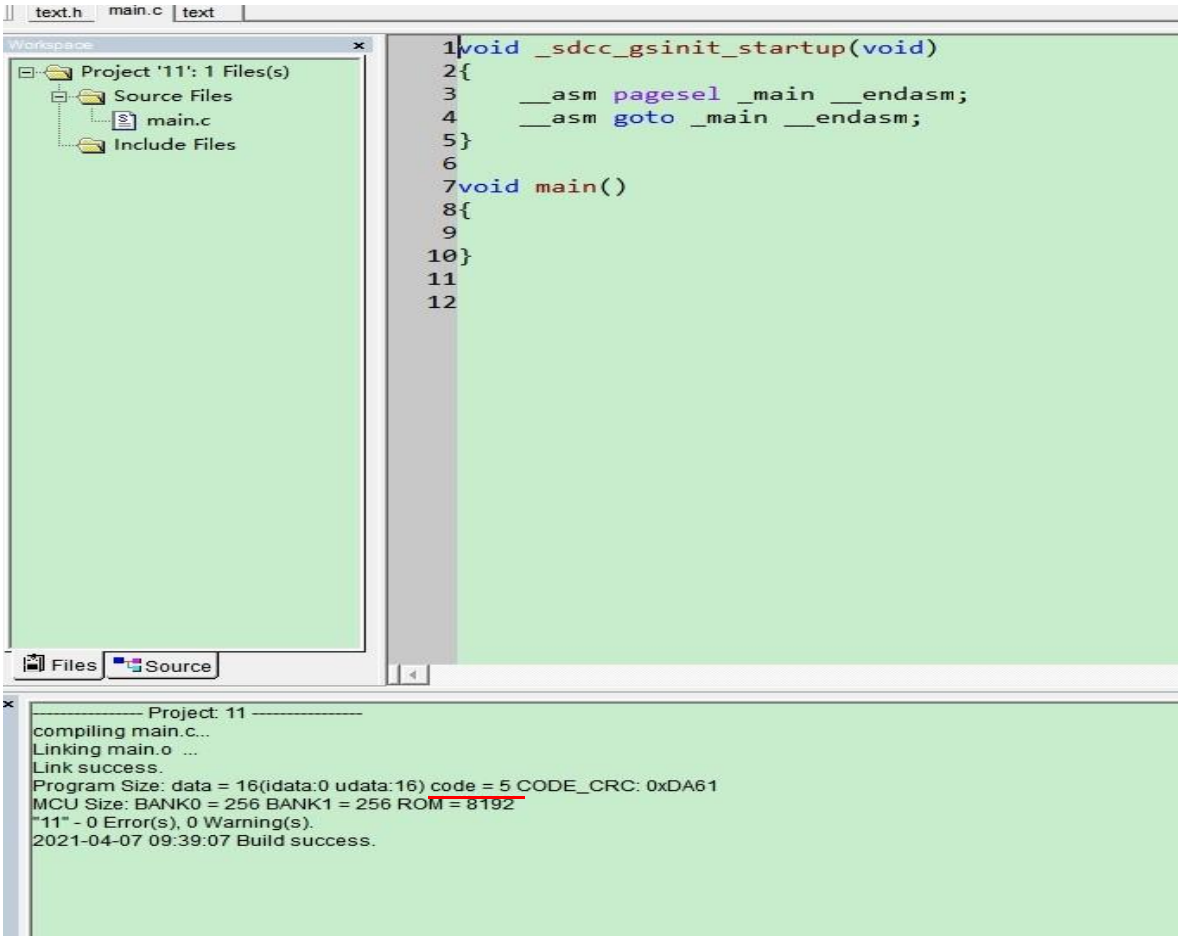
在文件开头加入指定函数 ：

```
void _sdcc_gsinit_startup(void)

{

    __asm pagesel _main __endasm;

    __asm goto _main __endasm;

}
```

即可减少 CODE 占用大小
加入前：



加入后：


text.h
main.c
text
Workspace
Project '11': 1 Files(s)
Source Files
main.c
Include Files
Files
Source
1void _sdcc_gsinit_startup(void)
2{
3 __asm pagesel _main __endasm;
4 __asm goto _main __endasm;
5}
6
7void main()
8{
9
10}
11
12
---------- Project: 11 ----------
compiling main.c...
Linking main.o ...
Link success.
Program Size: data = 16(idata:0 udata:16) code = 5 CODE_CRC: 0xDA61
MCU Size: BANK0 = 256 BANK1 = 256 ROM = 8192
"11" - 0 Error(s), 0 Warning(s).
2021-04-07 09:39:07 Build success.

5.5.2 注意事项

由于优化的是启动文件中全局变量初始化的一个功能，在优化后全局变量将不能初始化，需要在 main()函数中赋值。

错误：

```
1void _sdcc_gsinit_startup(void)
2{
3    __asm pagesel _main __endasm;
4    __asm goto _main __endasm;
5}
6
7int i = 2;
8
9void main()
10{
11    if(i == 2)
12    {
13
14    }
15}
16
17
```

，由于全局变量不能初始化，此时 i 不等于 2，相当于没有赋

正确：

```
1void _sdcc_gsinit_startup(void)
2{
3    __asm pagesel _main __endasm;
4    __asm goto _main __endasm;
5}
6
7int i;
8
9void main()
10{
11    i = 2;
12    if(i == 2)
13    {
14
15    }
16}
17
18
```

声明后，在 main 函数中赋值，此时 i 等于 2 值

5.6 查找功能

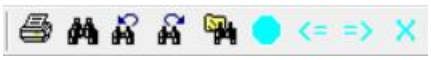


图 5.6-1 工具栏，打印，查找

Print，打印/输出 PDF 文档
Find，在当前文档查找
Find Previous，向前查找
Find Next，向后查找
Find in files，在多个文档中查找
Toggle Bookmark，标记当前行
Goto prev bookmark，查找前一个标记
Goto next bookmark，查找后一个标记

5.7 注释功能

图 5.7-1 工具栏，注释/取消注释选中代码

5.8 字体及背景颜色设置功能

图 5.8-1 进入 Edit 菜单栏

ChangeTabSize 修改 Tab 键大小为 4/8
Font&BkColor 修改字体及背景颜色
按住键盘左“Ctrl”键，移动鼠标光标至代码编辑窗口，滚动鼠标滑轮可放大/缩小字体大小

6 代码参考

6.1 C 参考代码

```
//更多例程请去官网 http://www.holychip.cn/pro.php?id=80 下载
#include <SQ013L.h>
#define SET_BIT(VAR, BIT) VAR  =  (VAR | (0x0000000000000001 << BIT))
#define CLEAR_BIT(VAR, BIT)  VAR  =  (VAR & (~(0x0000000000000001 << BIT)))
#define GET_BIT(VAR, BIT)      ((VAR >> BIT) & 0x0000000000000001)
void main(void)
{
    TRISB = 0x00;     //输入输出设置   1=输入，0=输出
    PORTB = 0x00;     //PORT 口输出高低电平设置   1=高电平，0=低电平
    PHCON = 0xFF;     //上拉设置  1=DISABLED PULL-UP   ;0=PULL-UP
    PDCON = 0xFF;     //下拉设置  1=DISABLED PULL-UP   ;0=PULL-UP
    ODCON = 0x00;     //开漏设置  0=DISABLED PULL-UP   ;1=PULL-UP
    INTECON = 0x81;
    OPTION = 0x00;    //Ftimer0 1/2
    T0 = 126;
    __asm__("clrwdt");//OPTION 设置 WDT 使能
    PCON = 0x80;      //更改   上下电复位

    while (1)
    {
        CLEAR_BIT(PORTB, 1);     //PORTB1 清零

        __asm
        clrwdt
        sleep
        __endasm;

        SET_BIT(PORTB, 1);    //PORTB1 置 1
    }
}
void Intr(void) __interrupt 0
{
    if(T0IF)
    {
        T0IF = 0;
        T0 = 126;                 //重置 T0 值
        PORTB0 = !PORTB0;
    }
}
```

6.1.1 C 语言自定义地址变量方法

定义__sfr__at(地址)变量；如：

```
__sfr __at (0x80) P0;
P0 = 0x10;
```

即可对地址 0x80 写入 0x10
详情请参考 DEMO 请去官网 http://www.holychip.cn/pro.php?id=80 下载

6.1.2 C 语言混合汇编使用方法

在 C 语言中插入__asm ..... __endasm；如：

```
__asm
nop
nop
//填入汇编程序
__endasm;
```

即可使用汇编语言
详情请参考 DEMO 请去官网 http://www.holychip.cn/pro.php?id=80 下载

6.1.3 C 语言位定义使用方法

定义

```
typedef struct __bits8_t
{
    unsigned bit0:1;
    unsigned bit1:1;
    unsigned bit2:1;
    unsigned bit3:1;
    unsigned bit4:1;
    unsigned bit5:1;
    unsigned bit6:1;
    unsigned bit7:1;
}bits8;
Bits8 a;
a. bit1 = 1
```

即可定义使用位
详情请参考 DEMO 请去官网 http://www.holychip.cn/pro.php?id=80 下载

6.2 ASM 参考代码

```
;更多例程请去官网 http://www.holychip.cn/pro.php?id=80 下载
    list p = SQ013L
W               EQU     H'0000'
F               EQU     H'0001'
T0              EQU     H'0001'
STATUS          EQU     H'0003'
PORTB           EQU     H'0006'
PCON            EQU     H'0008'
PCLATH          EQU     H'000A'
INTECON         EQU     H'000E'
INTFLAG         EQU     H'000F'
OPTION          EQU     H'0041'
TRISB           EQU     H'0046'

UDL_main_0  udata
PSAVE   res  1
SSAVE   res  1
WSAVE   res  1
UDATA0 res  1

    ORG     0000H
    NOP
    NOP
    GOTO MAIN
    ORG     0008H
    GOTO INTERRUPT

MAIN:
    CLRF    TRISB
    CLRF    PORTB
    MOVLW 0x81
    MOVWF INTECON
    CLRF    OPTION
    MOVLW 0x7E
    MOVWF T0
    MOVLW 0x80
    clrwdt
    MOVWF PCON
WHILE_1:
    clrwdt
    sleep
    GOTO    WHILE_1
    RETURN
INTERRUPT:
```

```
;SAVE_W_STATUS_PCLATH
        MOVWF WSAVE
        SWAPF   STATUS,W
        CLRF    STATUS
        MOVWF SSAVE
        BANKSEL     PCLATH
        MOVF    PCLATH,W
        CLRF    PCLATH
        MOVWF PSAVE
;END_OF_SAVE_W_STATUS_PCLATH

        BTFSS   INTFLAG,0
        GOTO    RESTORE_W_STATUS_PCLATH

        BCF     INTFLAG,0
        MOVLW 0x7E
        MOVWF T0

        CLRF    UDATA0
        BTFSC   PORTB,0
        INCF    UDATA0,F
        MOVF    UDATA0,W
        MOVLW 0x00
        BTFSC   STATUS,2
        MOVLW 0x01
        MOVWF UDATA0
        RRF     UDATA0,W
        BTFSS   STATUS,0
        BCF     PORTB,0
        BTFSC   STATUS,0
        BSF     PORTB,0

RESTORE_W_STATUS_PCLATH:
        MOVF    PSAVE,W
        MOVWF PCLATH
        CLRF    STATUS
        SWAPF   SSAVE,W
        MOVWF STATUS
        SWAPF   WSAVE,F
        SWAPF   WSAVE,W
        RETFIE

        end
```

7 编译

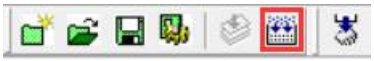


图 6-1 点击工具栏“Build”按钮

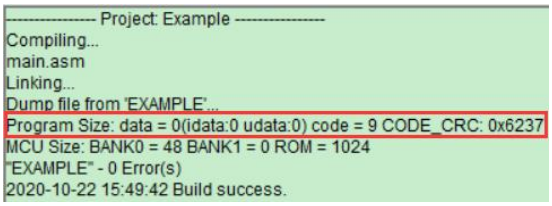


图 6-2 “Output”窗口生成编译链接信息

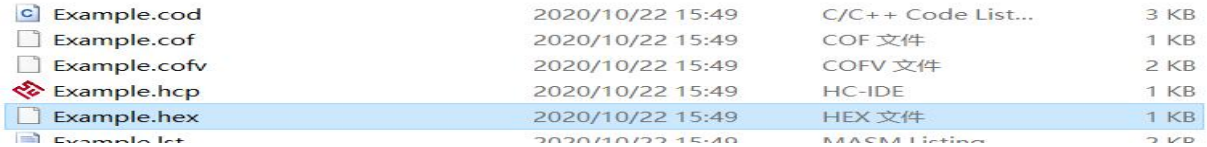


图 6-3 编译后生成目标*.hex 文件

错误/警告	原因
error:Missing definition for "STK12",required by "main.o"	没有定义 main 函数
Undefined identifier 'm'	m 未声明
syntax error:token->'}';column 1	在'}'之前的某个语句缺少分号
error 102:too few parameters	使用函数时，输入的实参不足
error 0: Duplicate symbol 'a',symbol IGNORED	变量 a 重复定义
10: error:#include expects "FILENAME" or <FILENAME>	include 语句文件名没有加分号或者尖括号
syntax error:token->'int';column 9	for 语句中定义了变量
syntax error:token->'int';column 4	变量没有声明在函数起始处
error 48:cannot assign values to aggregates	赋值时类型不兼容
error 78:incompatible types	变量引用不正确
error 33:Attempt to assign value to a constant variable(=)	常量再次赋值
error:Missing definition for symbol "_abcdef", required by "main.o"	函数 abcdef 有声明但没有定义
error 65:function 'Test' already has body	函数 Test 重复定义
19:fatal error:test.h:No such file or directory	没有这个文件或文件夹
warning 147:excess elements in array initializer after 'm'	数组越界
warning 110:conditional flow changed by optimizer:so said EVELYN the modified DOG	条件恒为假或者恒为真
warning 126:unreachable code	永远不会执行到的代码
warning 59:function 'Test' must return value	函数没有返回值
warning 85:in function Test unreferenced function argument:'a'	没有引用的形参
error: No target memory available for section "S_fsdiv___fsdiv".	float 导致的 ram 空间不足

表 6-1 常见编译错误/警告（供参考）

8 仿真

仿真前请将 HC-ICD-V4 的 USB 与电脑相连，仿真接口与仿真芯片相连，参考《2 硬件连接》。

图 7-1 工具栏点击设置按钮

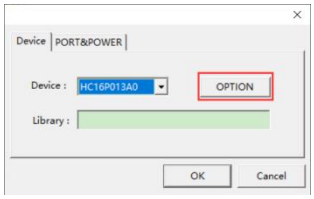


图 7-2 设置对话框，点击 OPTION 按钮

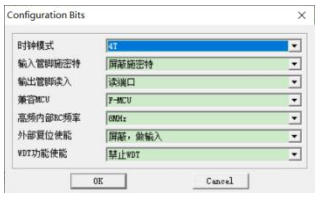


图 7-3 OPTION 设置对话框，请参考芯片数据手册

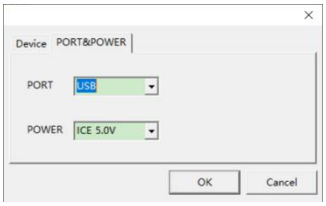


图 7-4 端口和电源设置对话框，选择正确的设备端口，设置供电方式

图 7-5 工具栏仿真相关按钮

从左到右依次为以下按钮:

1、“Download (F7)”按钮，进入仿真模式
2、“Stop Debug Session”按钮，退出仿真模式
3、“Reset (F4)”按钮，芯片复位
4、“Run (F5)”按钮，全速执行
5、“Halt (Shift+F5)”按钮，暂停执行
6、“Step Into (F11)”按钮，逐语句执行
7、“Step Over (F10)”按钮，逐过程执行
8、“Toggle Breakpoint (F9)”按钮，生成一个断点
9、“Clear All Breakpoint (F8)”按钮，清除全部断点

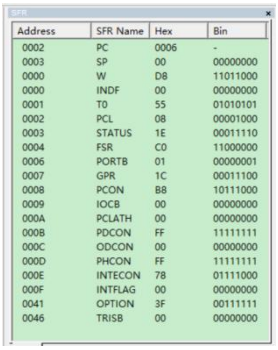


图 7-6 “SFR”窗口，查看 SFR 寄存器

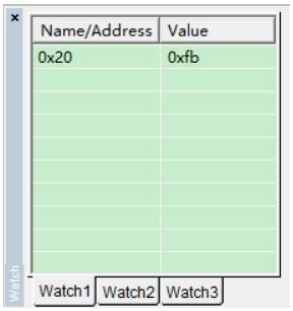


图 7-7 “Watch”窗口，查看 RAM 变量，“EQU”伪指令定义的变量请直接输入地址查看

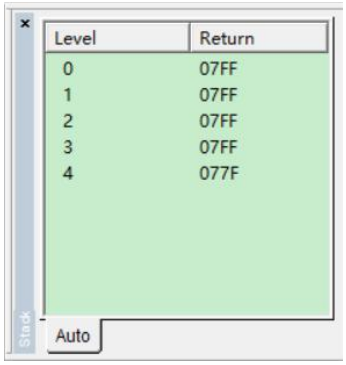


图 7-8 “Stack”窗口，查看堆栈

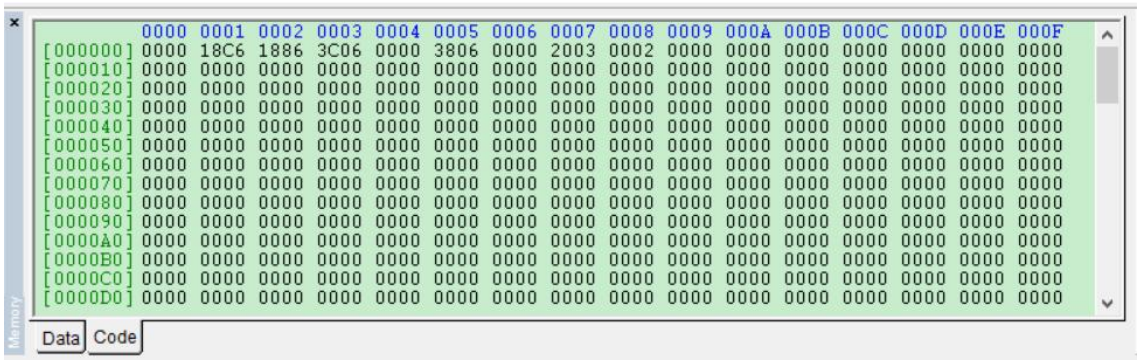


图 7-9 “Memory”窗口，查看 RAM/ROM 数据

默认只显示前 3 行数据，窗口内双击鼠标左键查看全部数据

9 烧录

烧录前请将 HC-ICD-V4 的 USB 与电脑相连，烧录接口与 OTP 芯片相连。

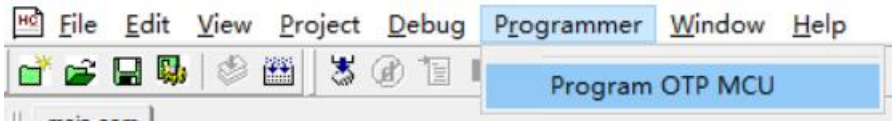


图 8-1 菜单栏依次点击“Programmer”，“Program OTP MCU”按钮，打开 HC-PM18 软件

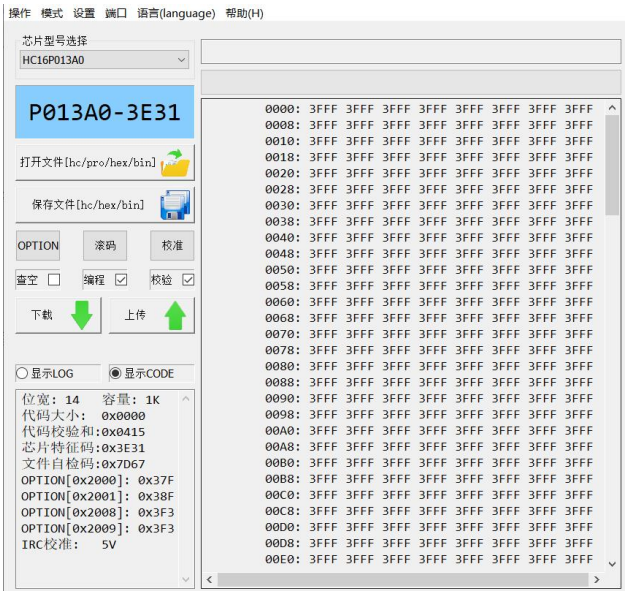


图 8-2 HC-PM18 软件界面

选择芯片型号，打开 hex 文件，配置 OPTION，点击下载按钮，开始烧录芯片。烧录成功上位机主界面状态栏显示 PASS、HC-ICD-V4 绿灯亮，烧录失败上位机主界面状态栏显示 FAIL、HC-ICD-V4 红灯亮。更多烧录配置请参考《OTP 烧录-HC-PM18-V5_使用手册》文档。

10 软件&固件更新

10.1 软件更新

上位机软件每次打开时都会自动连接芯圣官网，如果官网软件有更新，上位机软件会自动弹出软件更新提示窗口，用户可去芯圣官网（http://www.holychip.cn）下载最新软件。

10.2 固件更新

在线烧录芯片时，上位机软件会自动检查下位机固件是否是最新版本，如果固件不匹配上位机软件会提示用户更新固件。

固件更新前请将 HC-ICD-V4 的 USB 与电脑相连，参考图 8-1 打开 HC-PM18 软件。

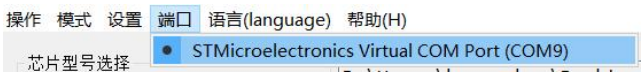


图 9.2-1 菜单栏“端口”，确定设备端口

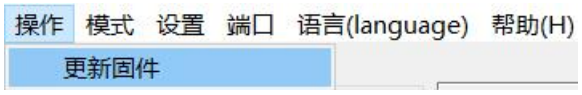


图 9.2-2 菜单栏“操作”，点击“更新固件”

图 9.2-3 固件更新，运行中...

图 9.2-4 固件更新成功，HC-ICD-V4 LED 灯先灭后亮

11 版本说明

版本	日期	描述
Ver1.00	2020/11/6	初版
Ver2.00	2022/1/4	增加指令集,code 优化
Ver3.00	2022/2/7	加入 C 语言中使用汇编,位定义,自定义地址的参考例程，编译注意事项
Ver4.00	2022/5/20	增加 HC-ICD-V4 新、旧蓝板实物图以及分别支持的芯片系列
Ver4.01	2022/6/7	增加蓝板使用注意事项
Ver5.00	2022/7/26	增加错误信息,修改芯片支持型号
Ver6.00	2023/2/28	增加蓝板、仿真板烧录、黑板使用注意事项和仿真替代芯片
Ver7.00	2023/7/3	修改注意事项，删除绿板支持芯片：SQ013L