简介

OTP_MTP 调试上位机 HC-IDE

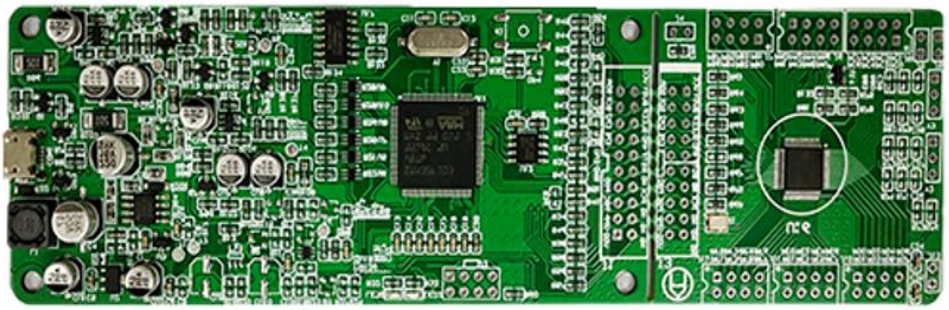
- C 代码编辑功能
- C 代码编译功能（XP 系统不支持 C 编译）
- 仿真功能
- 烧录功能
- 固件更新功能
- 支持 OTP 芯片：HC18P020L, HC18P018A0 , HC16P015B0, HC18P110A0 , SQ013L, SQ015L, HC18P122B1, HC18P235L, HC18P13XL, HC16P013A0， HC16P100B1， HC18P015A0，支持 MTP 芯片：HC18M002, HC18M003, HC18M301D, HC18M302D, HC18M303D, HC18M5823, HC18M5830, HC18M121B1,HC15P013A1,HC18M121E2,HC18M603.
- 部分语法如中断，内嵌汇编写法与原来编译器不同，详情请看 **6.1 参考代码**
- **请仔细阅读 7.2 注意事项**

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)

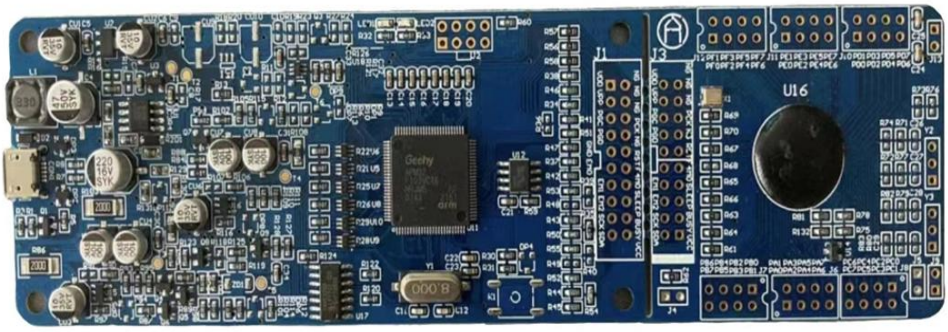
MTP 调试下位机 HC-ICDPRO V1.0

- 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)



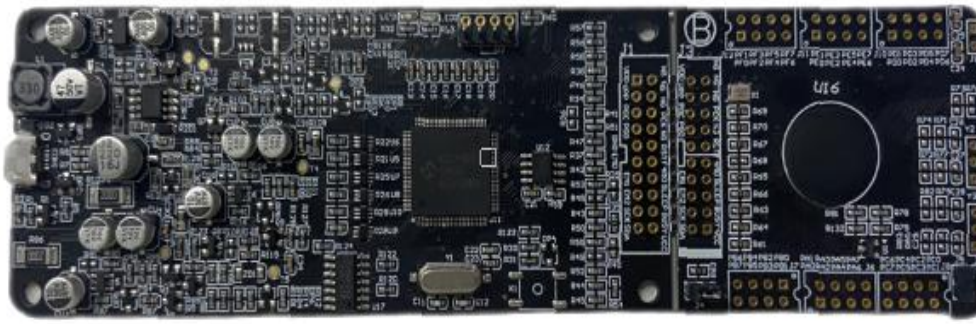
绿板支持芯片：

* 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

支持芯片：

*HC18M002

*HC18M003

*HC18M301D

*HC18M302D

*HC18M303D

*HC18M5823

*HC18M5830

*HC18M121B1

*HC18M603

HC-ICD Pro V1.0 产品实物图

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	
HC15P121B1	HC18M121B1	
HC15P013A0	HC18M121B1	
HC15P013A1	HC18M121B1	
HC18M121E2	HC18M121B1	
HC15P121E2	HC18M121B1	

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

2.1HC-ICD-V4

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。

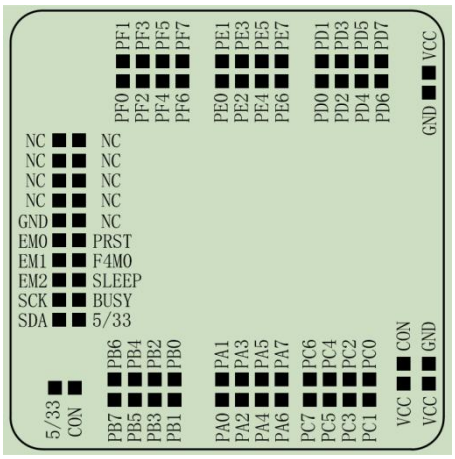


图 2-2 HC-ICD-V4 仿真子板引脚图

2.2 HC-ICDPRO V1.0

图 2-3 HC-ICDPRO V1.0 主板引脚图

仿真引脚：

VDD, VPP, PGC, PGD, GND。

烧录引脚：

VDD, VPP, PGC, PGD, GND。

3 新建项目

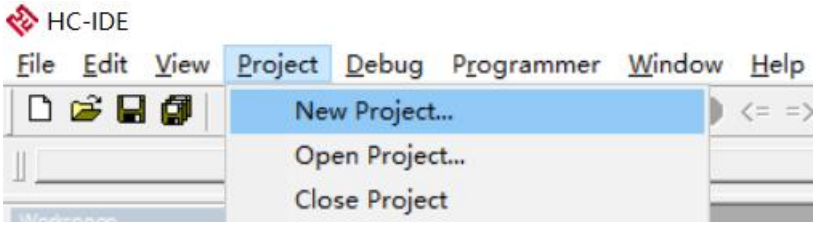


图 3-1 依次点击菜单栏“Project”，“New Project”新建一个项目

图 3-2 点击工具栏“Create Project”按钮新建一个项目

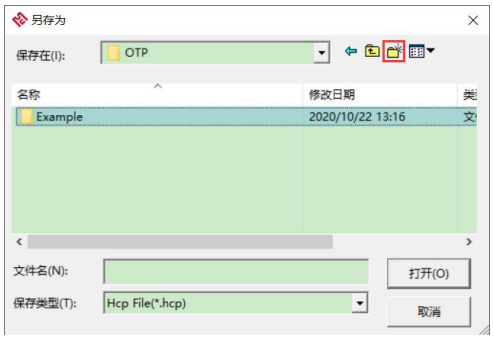


图 3-3 新建项目对话框，点击“创建新文件夹”按钮新建文件夹，注意路径不要有特殊字符

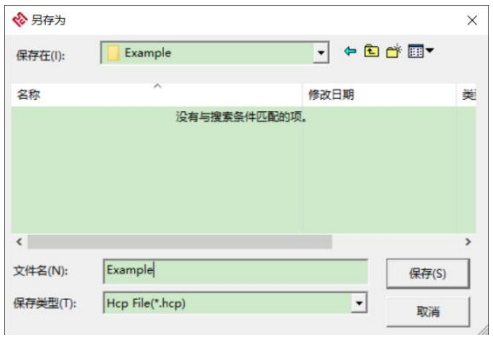


图 3-4 新建项目对话框，进入新建的“Example”文件夹，填写项目名称后点击“保存(S)”按钮

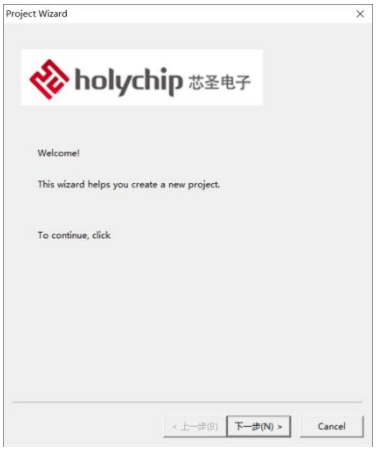


图 3-5 新建项目向导，欢迎界面，点击“下一步(N)”按钮

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

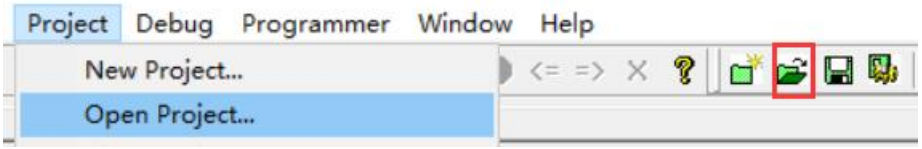


图 4-1 从菜单栏或工具栏打开一个项目

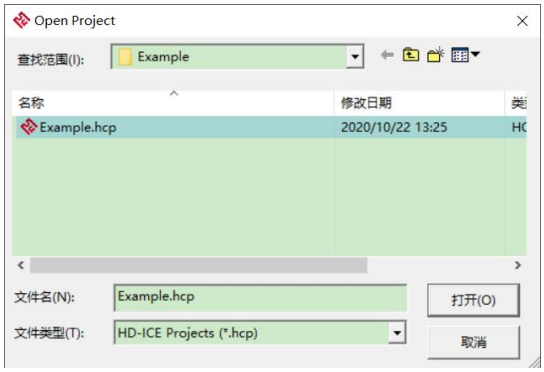


图 4-2 打开项目对话框

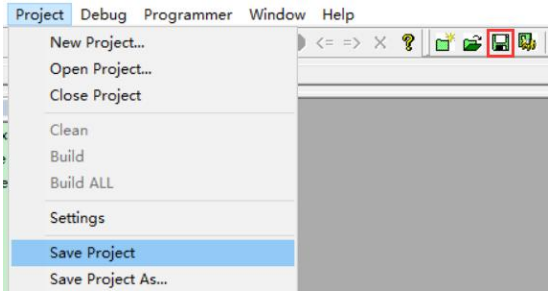


图 4-3 从菜单栏或工具栏保存者另存一个项目

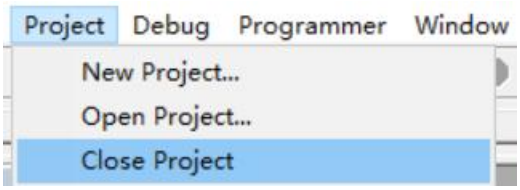


图 4-4 从菜单栏关闭一个项目

5 编辑

5.1 新建文件

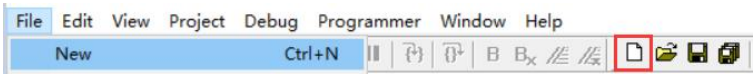


图 5.1-1 从菜单栏或工具栏新建一个文件

5.2 保存文件

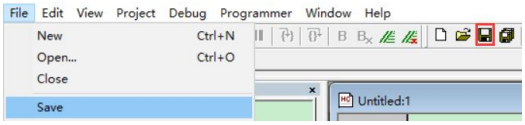


图 5.2-1 从菜单栏或工具栏保存一个文件

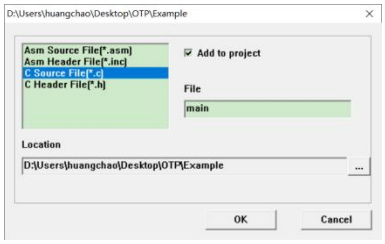


图 5.2-2 文件保存对话框

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

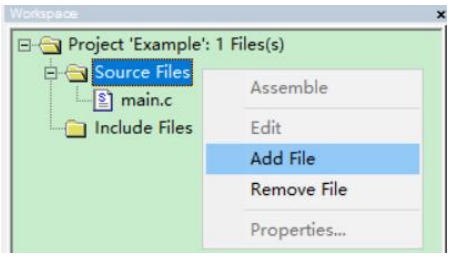


图 5.3-1 “Workspace”窗口右击“Source Files”或“Include Files”，添加或者删除源文件、头文件

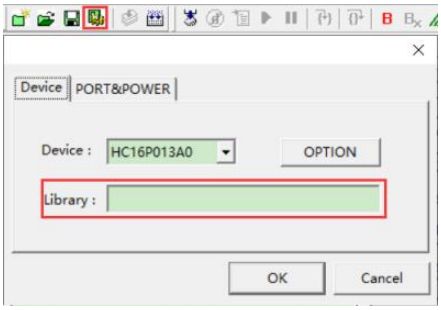


图 5.3-2 设置窗口添加/删除库文件（直接填库文件相对路径）

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

5.6 查找功能

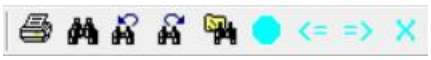


图 5.6-1 工具栏，打印，查找

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

```
#include <htc.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 interrupt    Timer0_Isr(void)
{
    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 变量类型 变量 _@ 地址 t(地址)变量; 如：

```
int P0 @ 0x100;
P0 = 0x10;
```

即可对地址 0x100 写入 0x10

6.1.2 C 语言混合汇编使用方法

```
asm("sleep");
```

在 C 语言中插入#asm ..... #enndasm; 如：

```
#asm
    nop
    nop
    //填入汇编程序
#endasm
```

6.1.3 C 语言位定义使用方法

定义 bit 变量，如：

```
bit P0;
P0 = 1;
```

6.1.4 中断函数

中断函数定义：

```
void interrupt   Timer0_Isr(void)
{
    ...
    ...
    ...
}
```

6.1.5 可用库函数

1.求整数绝对值

```
#include <stdlib.h>
int abs (int j);
```

2.#include <htc.h>

```
CLRWDT();
NOP();
SLEEP();
```

3.#include <string.h>

```
void * memchr (const void * block, int val, size_t length)

int memcmp (const void * s1, const void * s2, size_t n)

void * memcpy (void * d, const void * s, size_t n)

void * memset (void * s, int c, size_t n)
```

7 编译

7.1 编译流程

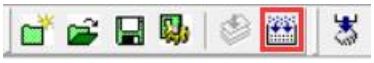
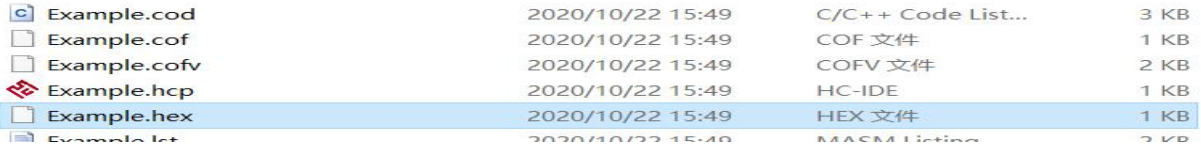


图 6-1 点击工具栏“Build”按钮

```
Memory Summary:
    Program space       used   D8h (  216) of  800h words  ( 10.5%)
    Data space          used  13Dh (  317) of  150h bytes  ( 94.3%)
    EEPROM space        None available
    Data stack space    used   0h (    0) of   Eh bytes  (  0.0%)
    Configuration bits  used   0h (    0) of   1h word   (  0.0%)
    ID Location space   used   0h (    0) of   4h bytes  (  0.0%)

"11" - 0 Error(s), 6 Warning(s).
2022-06-06 08:53:35 Build success.
```

图 6-2 “Output”窗口生成编译链接信息



7.2 注意事项

7.2.1 仿真板注意事项

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。

7.2.2 编译器 注意事项

1、如果出现编译报错不是内部或外部命令，也不是可运行的程序，请检查 IDE 文件夹路径是否有括号等特殊符号，若有则删除。

2、由于变量初始化功能关闭，声明变量时无法赋值，必须在函数中赋值。

3、汇编工程中需要将变量定义头文件放在最前面，函数定义的头文件放在最后，详情可参考汇编例程。

4、不会影响到芯片寄存器值和功能的变量和语句会被编译器优化，变量可加 volatile 关键字防止优化。

5、由于 HC18P110A0 变量可能不会自动分配到 bank1(A0-BF 地址)，使用 HC8P110A0 时若遇到 ram 未用完却提示空间不足，可以手动将变量定义到 bank1 地址可解决如：int temp @ 0xA0。

6、bank 最后区域为特殊寄存器的型号，由于编译器会将这部分寄存器也算进 ram 一起统计，在编译后显示的 ram 使用百分比会不准确，如 18M121B1 用满 ram 后最多显示到 120%多, 实际分配的地址无问题，多出的 20%是把特殊寄存器也算进去了，若 ram 超出会正常报 ram 不足的 error。

7、语句相似或者位置相近的语句，由于编译器内部优化，在内部部分语句会进行合并，多条 C 语言共用同一条指令,仿真跳转可能会出现显示错误，跳转位置顺序不对，跳到另一个语句上，或者要走到下一条语句才赋值，实际功能不受影响,已实际结果为准。

8、部分情况下部分语法由于编译器内部优化合并，在暂停和单步时无法停留在该语句上，实际程序执行，已实际运行结果为准，如：空 while 循环、switch 中的 case 到 break 写在同一行、连着写 2 个 switch 等情况。

9、16 位芯片使用时请在程序开头加入 FSR0H=1;若后续使用 indf 时修改 FSR0H 值，请在使用完后恢复 FSR0H 值

```
9void main(void)
0{
1//systen_rest:
2    FSR0H = 0X01;
3    inital();              //端口
4    DelayMs(50);           //延时
5    T0_sysinitial();
6    T1_sysinitial();
7    clr_ram();
```

10、bit 变量无法在 watch 窗口通过输入变量名直接查看，需要手动输入地址，如图 qwer bit 变量显示地址 0x369,需要除 8，0x368/8 = 109.1 在 watch 窗口填入 0x6d 即可查看，余数即代表对应 Bit 位

```
97
98void main(void)
99{
100//  return;
101   qwer
102// FSR
```

qwer (0x0369) = 0xFF

0x6d	0xab	171

11、ram 跨两个 bank 的型号如 hc18p235l， 18m303d，18m003，18m002，18m5830 在定义通过用 ram 变量时禁止使用@来指定通用 ram 地址。不支持特殊寄存器地址的间接寻址，特殊寄存器地址的变量不支持用指针、函数数组传参，数组 for 循环赋值。

12、由于 pic 编译器本身 bug，使用 const 数组时，[]内不推荐使用减法，需要将减法放在外面运算。

```
//const数组测试
for(i=0; i<10; i++)
{
    temp = i2 - 1;
    acs[i] = stemp[temp];
}
```

正确

```
//const数组测试
for(i=0; i<10; i++)
{
    acs[i] = stemp[i2 - 1];
}
```

错误

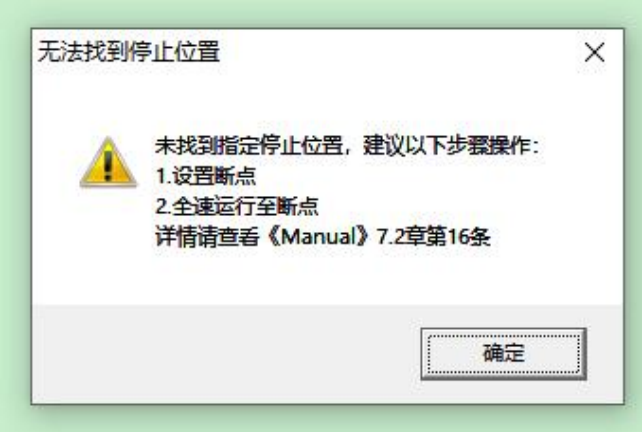
13、由于 pic 编译器本身 bug，不推荐使用 long int 和 long long 语法。

14、C 语言内嵌汇编使用函数时请使用单字母加数组组合，call 用 fcall 代替：

```
#asm
    k123:
        movf FSR0L
    goto k123
#endasm;
```

15、WIN7 系统编译时若提示缺少 api-ms-win-crt-runtime-l1-1-0.dll 等文件或者无反应,未编译生成任何文件,请打开软件目录下的 win7 运行库文件夹，先安装 Windows6.1-KB2999226 程序，再安装 vc_redist(x86 的必须装)和 KB2533623 即可解决。若安装 Windows6.1-KB2999226 时提示不适用你的计算机，请先在 https://www.catalog.update.microsoft.com/Search.aspx?q=KB976932 中下载 service pack 1 包安装，全部安装完成后需要重启电脑。

16、由于单步/暂停时程序可能处于长循环语句中导致长时间无法找到可停止位置，可在 C 语句上设置断点，全速运行到断点避开循环停止如下图：



while 循环长时间无法找到停止位置

```
13
14    while(1)
15    {
16        while(time --)
17        {
18        }
19        time = 125;
```

设置断点全速运行停止

7.2.3 M303D系列注意事项

1. 仿真模式下 HSRCRDY 标志位始终为 1 因此在仿真状态下不能作为芯片高低频切换标志，但烧录到芯片可以作为标志位正常使用。

2. 仿真状态暂时无法查看 EEPROM，非仿真状态下，导入外部 EEPROM 后在 EEPROM 窗口打开的状态下，烧录时才会将 EEPROM 数据写入芯片

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

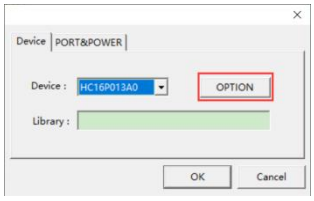


图 7-2 设置对话框，点击 OPTION 按钮

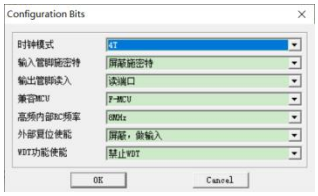


图 7-3 OPTION 设置对话框，请参考芯片数据手册

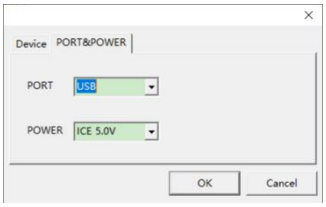


图 7-4 端口和电源设置对话框，选择正确的设备端口，设置供电方式

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

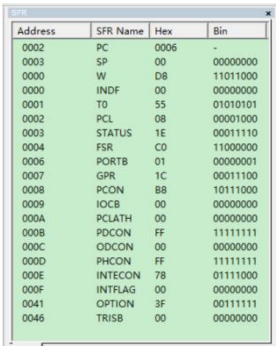


图 7-6 “SFR”窗口，查看 SFR 寄存器

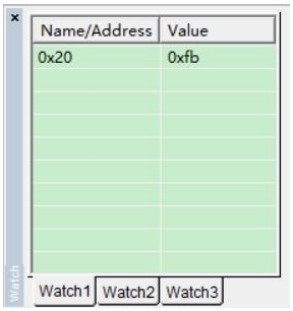


图 7-7 “Watch”窗口，查看 RAM 变量，“EQU”伪指令定义的变量请直接输入地址查看

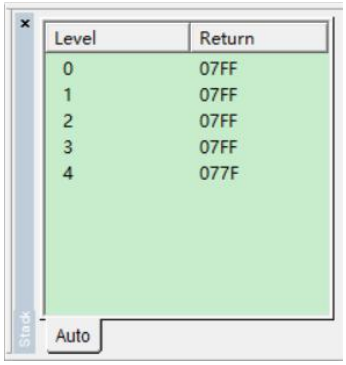


图 7-8 “Stack”窗口，查看堆栈

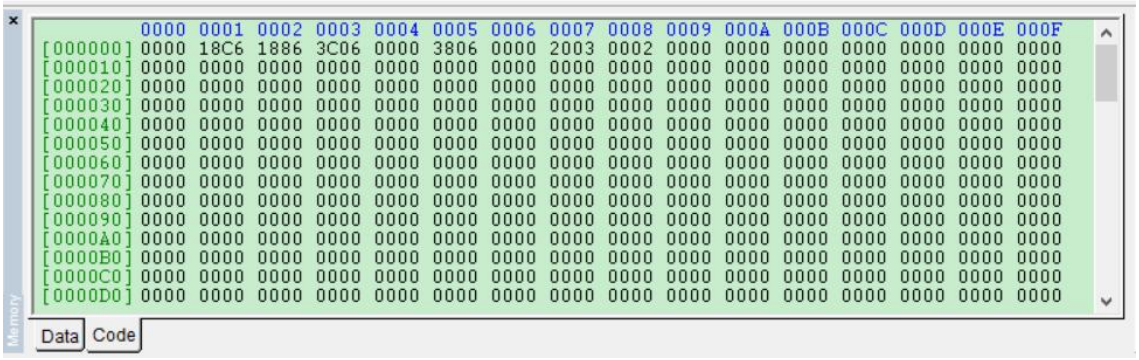


图 7-9 “Memory”窗口，查看 RAM/ROM 数据

默认只显示前 3 行数据，窗口内空白地方双击鼠标左键查看全部数据

9 烧录

9.1OTP 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。

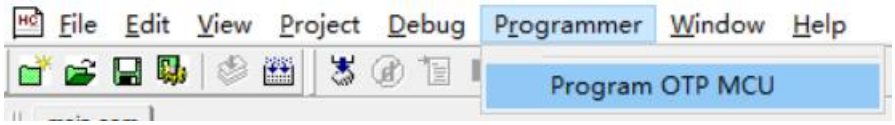


图 8-1 菜单栏依次点击“Programmer”，“Program OTP MCU”按钮，打开 HC-PM18 软件

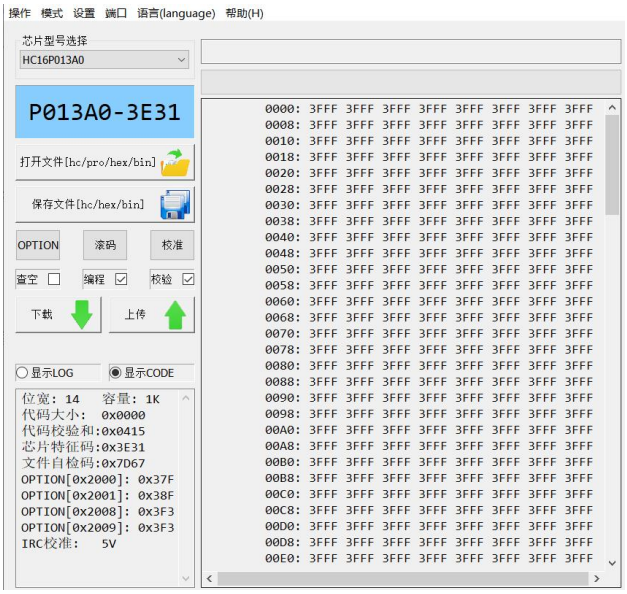


图 8-2 HC-PM18 软件界面

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。

9.2MTP 烧录

烧录前请将 HC-ICD PRO V1.0 的 USB 与电脑相连，烧录接口与 MTP 芯片相连。

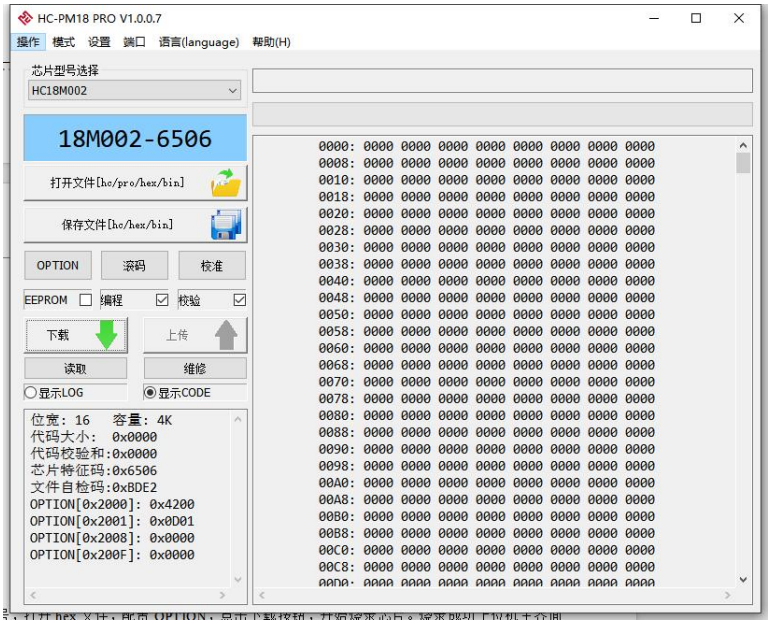


图 8-1 HC-PM18 PRO 软件界面

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 PRO V1.0 绿灯亮，烧录失败上位机主界面状态栏显示 FAIL、HC-ICD PRO V1.0 红灯亮。更多烧录配置请参考《MTP 烧录-HC-PM18 PRO-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/HC-ICD PRO V1.0的USB与电脑相连，参考图8-1打开HC-PM18/HC-PM18 PRO软件。

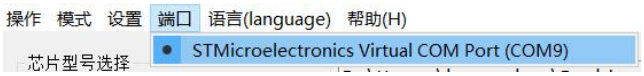


图 9.2-1 菜单栏“端口”，确定设备端口

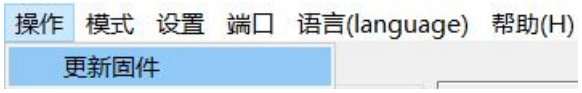


图 9.2-2 菜单栏“操作”，点击“更新固件”

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	2022/6/6	初版
Ver2.00	2022/7/18	增加注意事项
Ver3.00	2022/7/18	增加注意事项 14 位芯片关于 const,switch 语法注意事项,16 位型号关于 for 语法注意事项
Ver3.00	2022/8/17	注意事项增加语法事项
Ver4.00	2022/11/21	根据版本特性修改注意实现
Ver5.00	2022/3/3	修改注意事项，增加仿真板图
Ver6.00	2022/4/20	修改注意事项，增加 235l.,303d,中断注意项
Ver7.00	2023/11/7	增加 ICDPRO,修改注意事项，增加 C 参考代码-可用库函数。
Ver8.00	2023/12/15	ICD PRO 增加型号：HC18M121B1
Ver9.00	2023/12/26	修改注意事项，增加芯片仿真替代说明
Ver10.00	2024/2/29	ICD PRO 添加型号：HC15P013A1 HC18M121E2 HC18M603