

简介

OTP_MTP 调试上位机 BL-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：BL08P122
- 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 **5.1 参考代码**
- **请仔细阅读 6.2 注意事项**

OTP 调试下位机 BL-ICD-V1

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



支持芯片：
*BL08P122

BL-ICD Pro V1.0 产品实物图

目录

1	硬件连接.....	3
1.1	BL-ICDPRO V1.0.....	3
2	新建项目	4
3	打开/保存/关闭项目	6
4	编辑.....	7
4.1	新建文件	7
4.2	保存文件	7
4.3	添加/删除源文件、头文件、库文件.....	8
4.4	指令表	9
4.4.1	汇编指令表	9
4.4.2	伪指令	10
4.6	查找功能	11
4.7	注释功能	11
4.8	字体及背景颜色设置功能.....	11
5	代码参考	12
5.1	C 参考代码.....	12
5.1.1	C 语言自定义地址变量方法	13
5.1.2	C 语言混合汇编使用方法	13
5.1.3	C 语言位定义使用方法	13
5.1.4	中断函数	13
5.1.5	可用库函数	13
6	编译	15
6.1	编译流程	15
6.2	注意事项	16
7	仿真	18
8	版本说明	20

1 硬件连接

1.1 BL-ICDPRO V1.0



图 2-3 BL-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。

2 新建项目

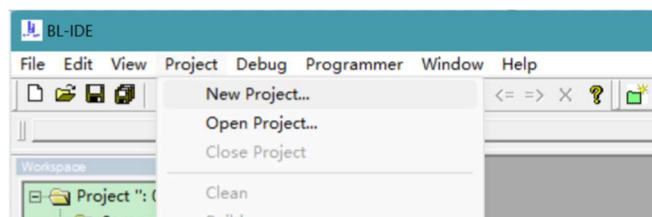


图 3-1 依次点击菜单栏“Project”，“New Project”新建一个项目

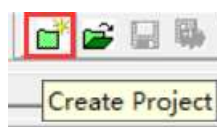


图 3-2 点击工具栏“Create Project”按钮新建一个项目

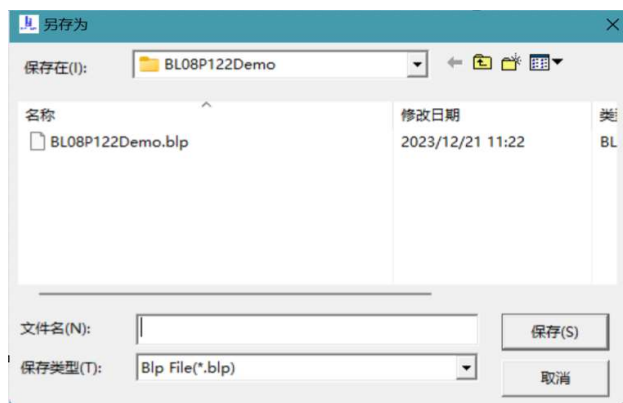


图 3-3 新建项目对话框，点击“创建新文件夹”按钮新建文件夹，注意路径不要有特殊字符

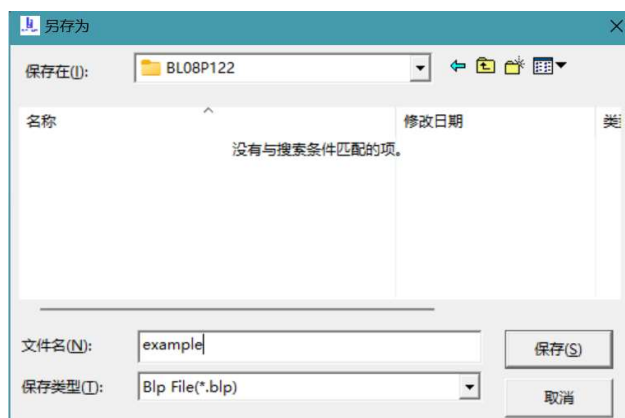


图 3-4 新建项目对话框，进入新建的“Example”文件夹，填写项目名称后点击“保存(S)”按钮

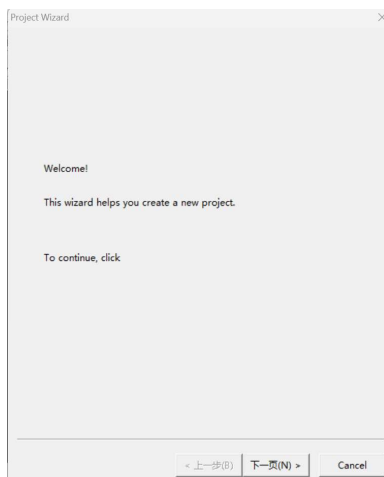


图 3-5 新建项目向导，欢迎界面，点击“下一步(N)”按钮

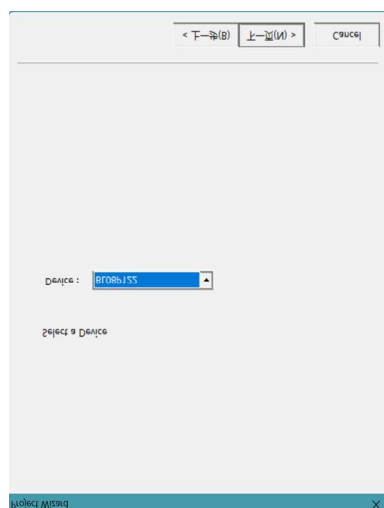


图 3-6 新建项目向导，选择芯片型号，点击“下一步(N)”按钮

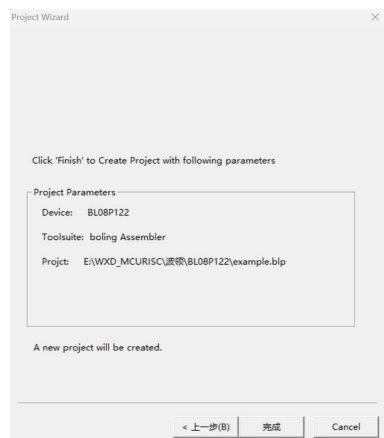


图 3-7 新建项目向导，确认界面，点击“完成”按钮，至此项目新建完成

3 打开/保存/关闭项目

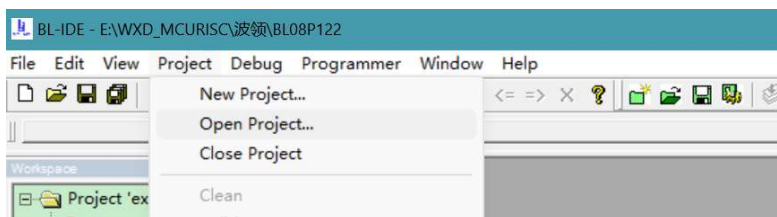


图 3-1 从菜单栏或工具栏打开一个项目

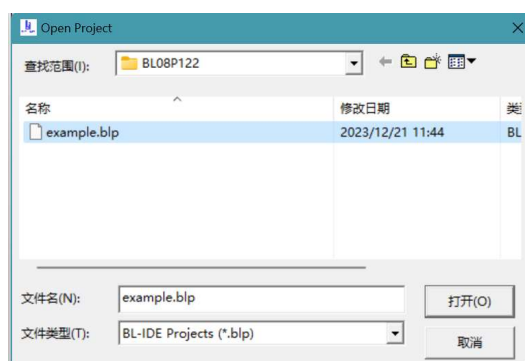


图 3-2 打开项目对话框

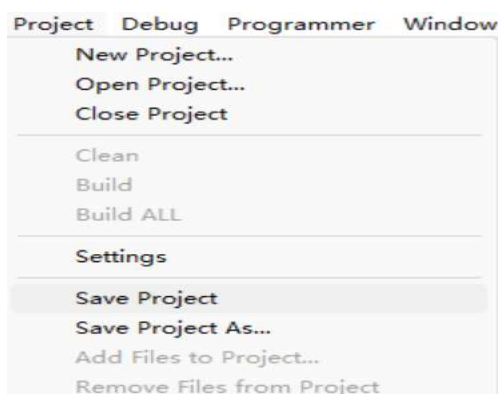


图 3-3 从菜单栏或工具栏保存者另存一个项目

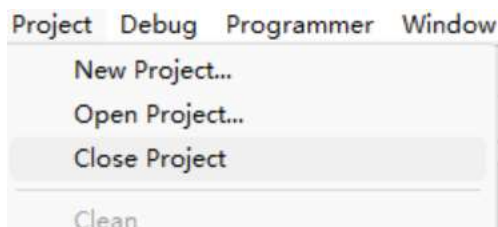


图 3-4 从菜单栏关闭一个项目

4 编辑

4.1 新建文件



图 4.1-1 从菜单栏或工具栏新建一个文件

4.2 保存文件

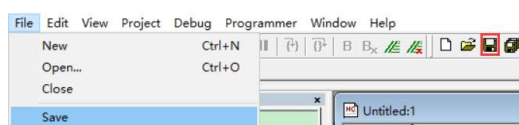


图 4.2-1 从菜单栏或工具栏保存一个文件

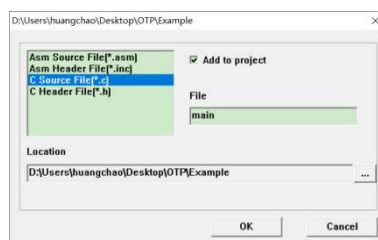


图 4.2-2 文件保存对话框

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。

4.3 添加/删除源文件、头文件、库文件

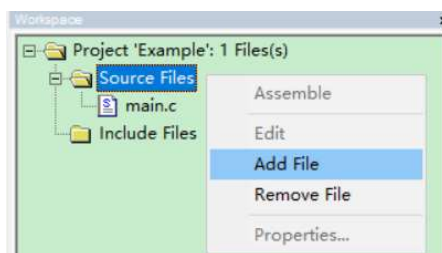


图 4.3-1 “Workspace”窗口右击“Source Files”或“Include Files”，添加或者删除源文件、头文件

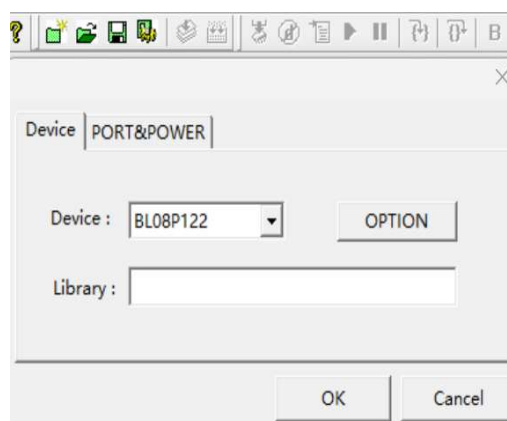


图 4.3-2 设置窗口添加/删除库文件（直接填库文件相对路径）

4.4 指令表

4.4.1 汇编指令表

Field	指令格式	描述	C	DC	Z	周期
移 动	MOVWF F	$F \leftarrow W$	-	-	-	1
	MOVF F, D	$D \leftarrow F$ (D=0 时为 W, D=1 时为 F)	-	-	√	1
	MOVLW k	$W \leftarrow k$	-	-	-	1
算 术	ADDWF F, D	$D \leftarrow W + F$ (D=0 时为 W, D=1 时为 F)	√	√	√	1
	ADDLW k	$W \leftarrow W + k$	√	√	√	1
	SUBWF F, D	$D \leftarrow F - W$ (D=0 时为 W, D=1 时为 F)	√	√	√	1
	SUBLW k	$W \leftarrow k - W$ (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	$D \leftarrow F + 1$ (D=0 时为 W, D=1 时为 F)	-	-	√	1
	DECF F, D	$D \leftarrow F - 1$ (D=0 时为 W, D=1 时为 F)	-	-	√	1
逻 辑	ANDWF F, D	$D \leftarrow W$ 与 F (D=0 时为 W, D=1 时为 F)	-	-	√	1
	ANDLW k	$W \leftarrow W$ 与 k	-	-	√	1
	IORWF F, D	$D \leftarrow W$ 或 F (D=0 时为 W, D=1 时为 F)	-	-	√	1
	IORLW k	$W \leftarrow W$ 或 k	-	-	√	1
	XORWF F, D	$D \leftarrow W$ 异或 F (D=0 时为 W, D=1 时为 F)	-	-	√	1
	XORLW k	$W \leftarrow W$ 异或 k	-	-	√	1
	COMF F, D	$D \leftarrow F$ 取反 (D=0 时为 W, D=1 时为 F)	-	-	√	1
处 理	SWAPF F, D	$D[7:4, 3:0] \leftarrow F[3:0, 7:4]$ (D=0 时为 W, D=1 时为 F)	-	-	-	1
	RRF F, D	$D \leftarrow F$ 带进位右移 (D=0 时为 W, D=1 时为 F)	√	-	-	1
	RLF F, D	$D \leftarrow F$ 带进位左移 (D=0 时为 W, D=1 时为 F)	√	-	-	1
	CLRW -	$W \leftarrow 0$	-	-	√	1
	CLRF F	$F \leftarrow 0$	-	-	√	1
	CLRWD T -	清零看门狗定时器, 影响 TO, PD 位	-	-	-	1
	BCF F, d	$F[d] \leftarrow 0$ ($0 \leq d \leq 7$)	-	-	-	1
	BSF F, d	$F[d] \leftarrow 1$ ($0 \leq d \leq 7$)	-	-	-	1
分 支	INCFSZ F, D	$D \leftarrow F + 1$ (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	$D \leftarrow F - 1$ (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 $F[d]=0$ ($0 \leq d \leq 7$) 则跳过下一句	-	-	-	1(2)
	BTFSS F, d	如果 $F[d]=1$ ($0 \leq d \leq 7$) 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	$W \leftarrow k$, 带参数返回	-	-	-	2
	NOP -	空操作	-	-	-	1
	SLEEP -	进入待机模式, 影响 TO, PD 位	-	-	-	1

4.4.2 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式	
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

4.6 查找功能



图 4.6-1 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

4.7 注释功能



图 4.7-1 工具栏，注释/取消注释选中代码

4.8 字体及背景颜色设置功能

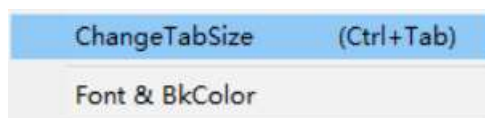


图 4.8-1 进入 Edit 菜单栏

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

5代码参考

5.1 C 参考代码

```
#include <bl08p122.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;
    }
}
```

5.1.1 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

5.1.2 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
```

5.1.3 C 语言位定义使用方法

定义 bit 变量, 如:

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

5.1.4 中断函数

中断函数定义:

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

5.1.5 可用库函数

1.求整数绝对值

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

2.#include <htc.h>

```
CLRWDI ();  
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)
```

6编译

6.1 编译流程

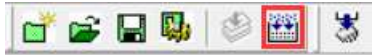


图 6-1 点击工具栏 “Build” 按钮

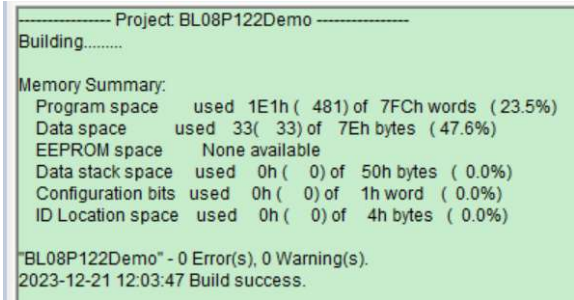




图 6-2 “Output” 窗口生成编译链接信息

 BL08P122Demo.hex	2023/12/21 12:03	HEX 文件	12 KB
 BL08P122Demo.hxl	2023/12/21 12:03	HXL 文件	2 KB
 BL08P122Demo.lst	2023/12/21 12:03	LST 文件	160 KB

6.2 注意事项

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，若有则去除
- 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、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。
- 4、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 08P122 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。



- 5、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。

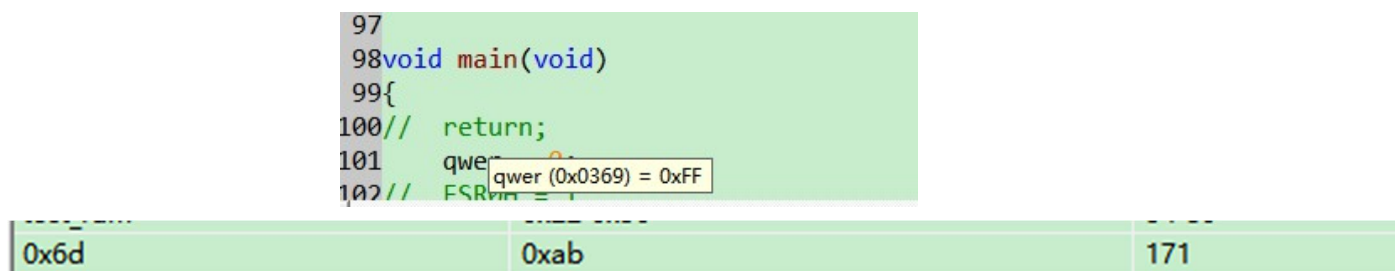
- 6、16 位芯片使用时请在程序开头加入 FSR0H=1;若后续使用 indf 时修改 FSR0H 值，请在使用完后恢复 FSR0H 值

```

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

```


7、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, $0x369/8 = 109.1$ 在 watch 窗口填入 0x6d 即可查看，余数即代表对应 Bit 位



8、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

9、中断和主程序同时使用同一个变量会有风险，主程序在操作变量未完成时进入中断，中断中改变该变量可能会造成非预期数据。

10、汇编使用函数时请使用单字母加数组组合：

```

#asm
    k123:
        movf FSR0L
        goto k123
#endasm;

```

11、(推荐使用 win10 操作系统) ， 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。

7 仿真

仿真前请将 BL-ICD-V1 的 USB 与电脑相连，仿真接口与仿真芯片相连。



图 7-1 工具栏点击设置按钮

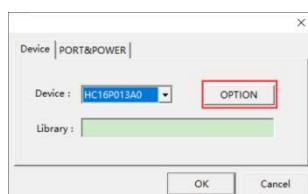


图 7-2 设置对话框，点击 OPTION 按钮



图 7-3 OPTION 设置对话框，请参考芯片数据手册

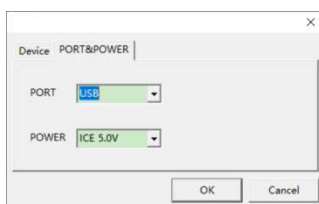


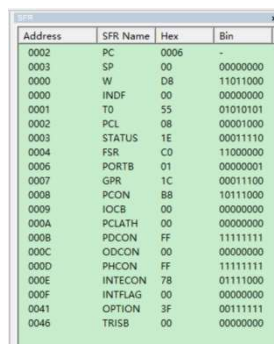
图 7-4 端口和电源设置对话框，选择正确的设备端口，设置供电方式



图 7-5 工具栏仿真相关按钮

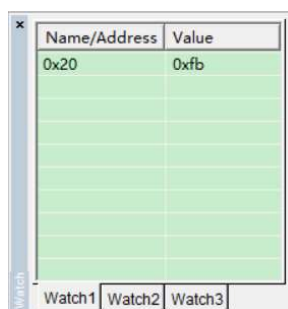
从左到右依次为以下按钮:

- 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



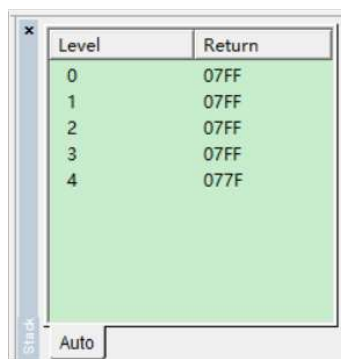
Address	SFR Name	Hex	Bin
0002	PC	0006	-
0003	SP	00	00000000
0000	W	D8	11011000
0000	INDF	00	00000000
0001	TO	55	01010101
0002	PCL	08	00001000
0003	STATUS	1E	00011110
0004	FSR	C0	11000000
0005	PORTB	01	00000001
0007	GPR	1C	00011100
0008	PCON	B8	10111000
0009	IOCB	00	00000000
000A	PCLATH	00	00000000
000B	PDCON	FF	11111111
000C	ODCON	00	00000000
000D	PHCON	FF	11111111
000E	INTECON	78	01111000
000F	INTFLAG	00	00000000
0041	OPTION	3F	00111111
0046	TRISB	00	00000000

图 7-6 “SFR” 窗口，查看 SFR 寄存器



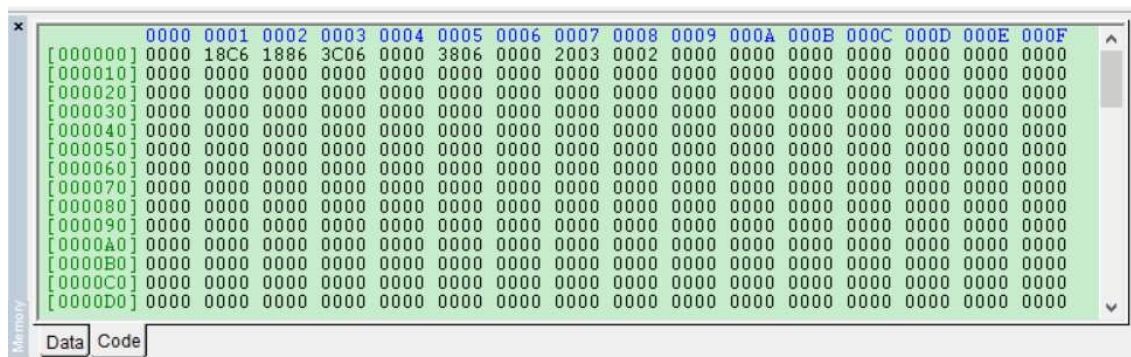
Name/Address	Value
0x20	0xfb

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



Level	Return
0	07FF
1	07FF
2	07FF
3	07FF
4	077F

图 7-8 “Stack” 窗口，查看堆栈



Address	0000	0001	0002	0003	0004	0005	0006	0007	0008	0009	000A	000B	000C	000D	000E	000F
[000000]	0000	18C6	1886	3C06	0000	3806	0000	2003	0002	0000	0000	0000	0000	0000	0000	0000
[000010]	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
[000020]	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
[000030]	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
[000040]	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
[000050]	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
[000060]	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
[000070]	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
[000080]	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
[000090]	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
[0000A0]	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
[0000B0]	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
[0000C0]	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
[0000D0]	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

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

8 版本说明

版本	日期	描述
Ver1.00	2023/12/21	初版

BOLING公司保留对以下所有产品在可靠性、功能和设计方面的改进作进一步说明的权利。BOLING不承担由本手册所涉及的产品或电路的运用和使用所引起的任何责任，BOLING的产品不是专门设计来应用于外科植入、生命维持和任何BOLING产品产生的故障会对个体造成伤害甚至死亡的领域。如果将BOLING的产品用于上述领域，即使这些是由BOLING在产品设计和制造上的疏忽引起的，用户应赔偿所有费用、损失、合理的人身伤害或死亡所直接或间接所产生的律师费用，并且用户保证BOLING及其雇员、子公司、分支机构和销售商与上述事宜无关。

波领科技