

CMT2280F2/2281F2/2189B/2189C IDE操作指引

概要

本文主要介绍关于 CMT2280F2、CMT2281F2、CMT2189B、CMT2189C 四款的 SoC 的编程 IDE 操作指引。这四颗 SoC 芯片均采用相同的高性能、RISC 内核，采用 Flash ROM，支持在线调试。它们都属于 CMOSTEK NextGenRF™ 系列的产品。该产品系列包含了发射机，接收机，收发一体机和 SoC 等短距离无线通信芯片。

本文档涵盖的产品型号如下表所示。

表 1. 本文档涵盖的产品型号

产品型号	工作频率	调制方式	发射功率	灵敏度	工作电流	配置方式	封装
CMT2280F2	300 – 960MHz	OOK	--	-110dBm	3.8mA	内嵌 MCU	SOP16
CMT2281F2	300 - 960MHz	OOK	--	-109dBm	4.5mA	内嵌 MCU	SOP16
CMT2189B	240 - 960MHz	OOK	+13dBm	--	17.5mA ^[1]	内嵌 MCU	SOP14
CMT2189C	240 – 960MHz	OOK/(G)FSK	+13dBm	--	32.5mA ^[2]	内嵌 MCU	SOP14

注意：

1. CMT2189B 的发射功率和发射电流测试条件为 433.92MHz，CW 模式（即一直处于发射载波模式）；按 Duty 50%的发射模式，发射电流约为 8.5mA；
2. CMT2189C 的发射功率和发射电流测试条件为 433.92MHz，FSK 模式；

目录

1	IDE 简介	4
2	主界面	5
2.1	工作区	6
2.2	编辑区	6
2.3	输出区	6
2.4	观察区	7
2.5	菜单、工具条、状态栏	8
3	新建工程	9
4	编译	10
5	EEPROM 设置	11
6	下载	12
7	仿真	13
7.1	设置断电	13
7.2	Reset (Ctrl+F5)	13
7.3	Step Into (F11)	13
7.4	Step Over (F10)	13
7.5	Run to Cursor (Ctrl+F10)	13
7.6	Go (F5)	13
7.7	Stop (Shift+F5)	14
8	附录 1 (编译器说明)	15
8.1	变量命名规则	15
8.2	数字进制表示	15
8.3	地址标号	15
8.4	指令	16
8.5	伪指令	17
8.5.1	ORG	17
8.5.2	Include	17
8.5.3	EQU	17
8.5.4	DB	17
8.5.5	DE	17
8.5.6	DBIT	18
8.5.7	Define	18
8.5.8	MARCO	18
8.5.9	Ifdef	19

9 附录 2（单片机调试说明书） 20

9.1 概述 20

9.2 线路说明 21

9.3 注意事项 22

10 文档变更记录 23

11 联系方式..... 24

CMOSTEK Confidential

1 IDE 简介

IDE 是为 CMT2280F2、CMT2281F2、CMT2189B、CMT2189C 这四颗 CMT 旗下的 SoC 提供开发软件用，它的主要功能是调试、仿真这四颗 SoC 的程序。

IDE 提供以下功能：

- 使用内置编辑器创建和编辑源程序；
- 编译源代码；
- 通过连接调试器，下载、调试可执行程序；
- 在观察窗查看变量

CMOSTEK Confidential

2 主界面

主界面如图 2-1，包含工作区、编辑区、输出区和观察区。除编辑区外其他区都是可拖动、悬停和停靠的窗口。

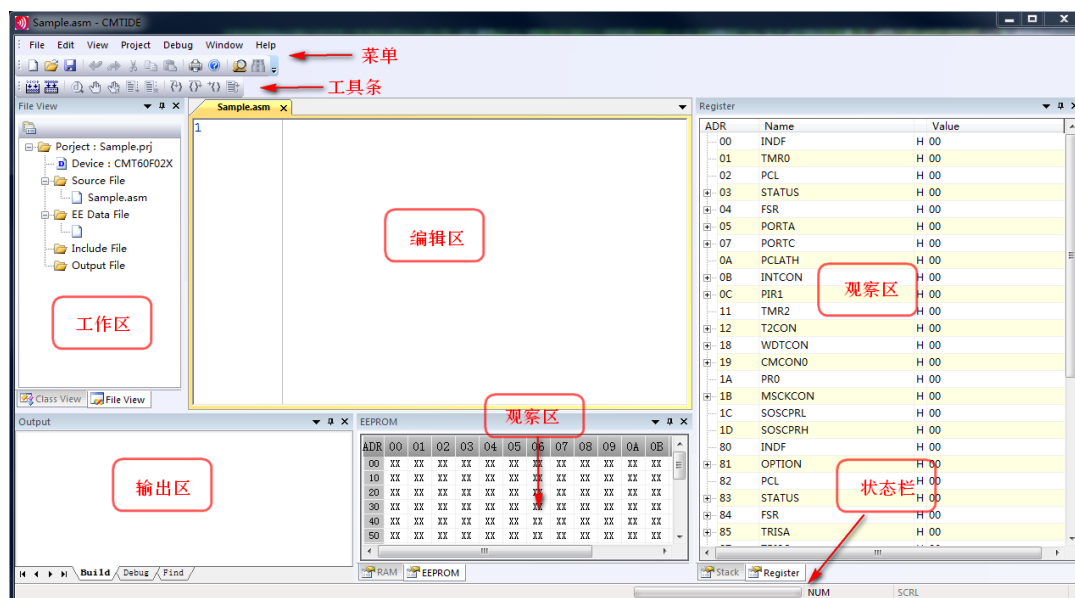


图 2-1. 主界面

例如按住鼠标左键拖动工作区时，出现停靠箭头，如图 2-2。当将工作区拖动到箭头指定的方向里时，将停靠到以编辑区为中心的箭头对应位置。当将工作区拖动到箭头指定的方向时，将停靠到以屏幕为中心的对应位置。

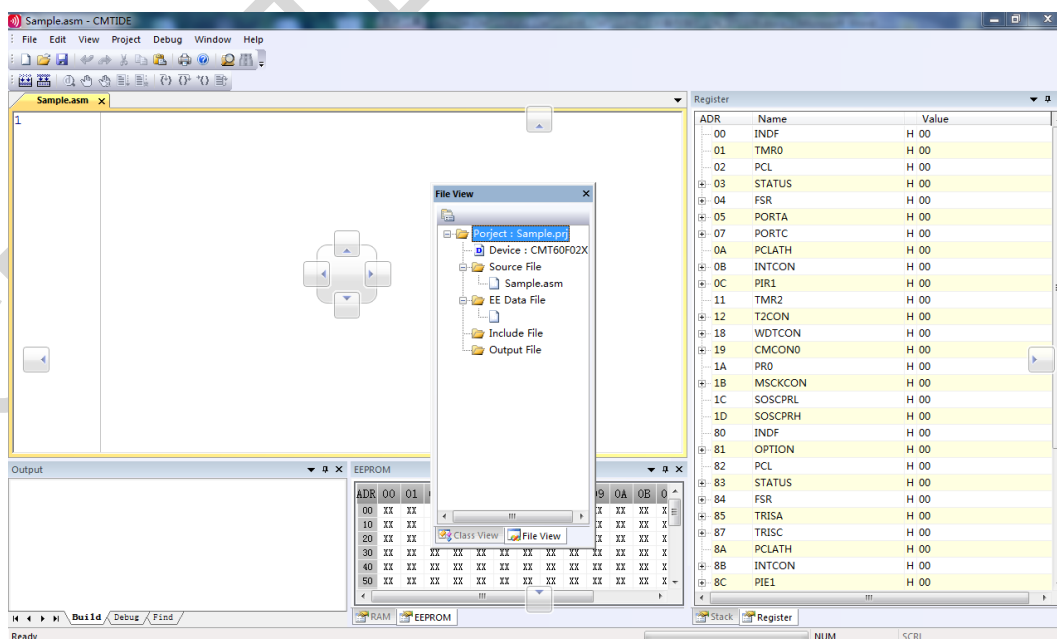


图 2-2. 窗体布局

2.1 工作区

工作区包含 **File View** 和 **ClassView** 两个窗口，如图 2-3 所示。**File View** 显示工程信息，包含器件名、源文件、数据文件、引用文件以及输出文件。双击 **Source File** 和 **Include File** 下的文件名可打开文件。**ClassView** 在编译完成后显示所有的 **Lable**，双击 **Lable** 在编辑区可以定位该 **Lable** 在源文件中的位置。

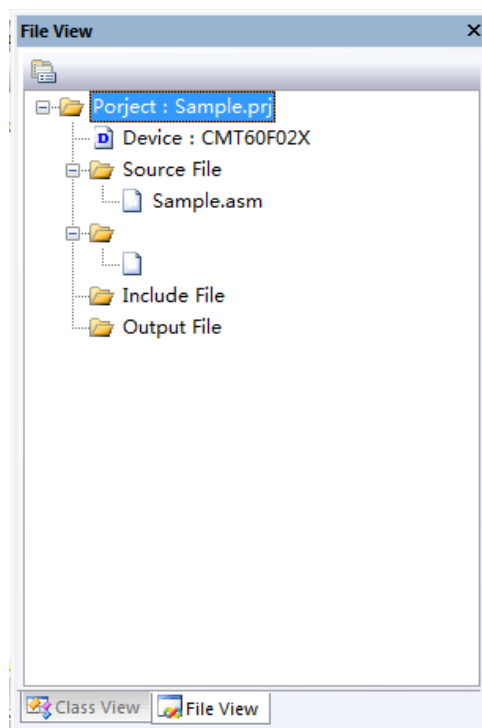


图 2-3. 工作区

2.2 编辑区

编辑区编辑源文件和引用文件，编辑器对关键字着色，指令为蓝色，立即数为红色，注释为绿色。按住 **Ctrl** 滚动鼠标中键可放大或缩小字体。在调试期间文件被编辑并保存时，自动退出调试状态，需重新编译和下载后方可再次调试。在编辑区打开的文档被外部编辑器修改后会提示重新载入，如果在调试状态，同样会退出调试，重新编译下载后方可再次调试。

2.3 输出区

输出区分为三个 **Tab**（注意只有输出区的 **Tab** 不可以分离，其他区的每个 **Tab** 都是可以分离组合的），分别是 **Build**、**Debug** 和 **Find** 窗口，如图 2-4。**Build** 显示编译、下载等输出信息，编译显示的错误包含出错的文件、行号以及该行语句，双击可以在编辑区定位到该行。

Debug 观测变量，在 **Debug** 窗口输入需要观察的变量，调试时将刷新该变量的值以及对应的地址，如果该变量为位变量，刷新的是为地址以及对应的 **bit** 值。**Find** 罗列查找到的文件以及行号，双击可以在编辑区定

位到该行。

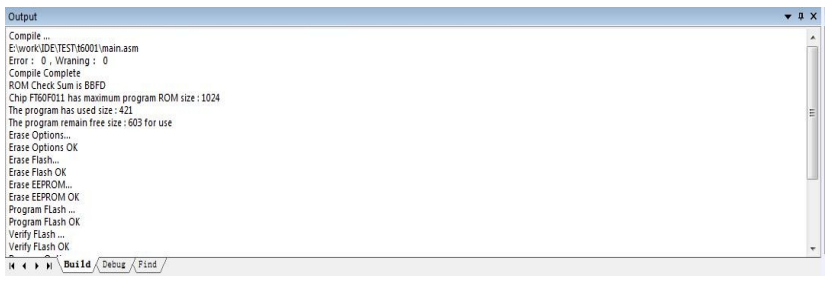


图 2-4. 输出区

2.4 观察区

观察区包含四个部分：RAM、EEPROM、STACK 和 Register，如图 2-5 和 2-6。

RAM 窗口显示 RAM 值，调试时以红色标注有变化的 RAM，开始调试后，可双击修改 RAM 值，修改后在 RAM 窗口任意位置单击，修改的值将被写入芯片中。

EEPROM 窗口显示 EEPROM 值，EEPROM 可以通过菜单“Load EEPROM Data...”导入数据，也可以通过 DE 指令在源文件中定义，同样可以在窗口中编辑，编辑方法同 RAM，双击修改 EEPROM 值，修改后在 EEPROM 窗口任意位置单击，修改的值将被写入芯片中。选择菜单“Export EEPROM Data...”可以输出 EEPROM 的数据，注意如果程序又修改 EEPROM 的值，那么在调试后输出的 EEPROM 数据有可能和编译后输出的 EEPROM 数据不同。

STACK 显示堆栈值，堆栈值不可以修改。仅用于观测芯片的压栈和弹栈情况。

Register 显示 SFR 特殊功能寄存器，包含寄存器地址、寄存器名和寄存器值以及所显示值进制（H 表示十六进制，B 表示二进制），点击“+”可以展开显示位变量，以及对应位的值，点击“-”收起位变量。在调试时刷新显示，发生变化的寄存器以红色标注。开始调试后，双击可以修改寄存器值，修改后在 Register 窗口任意位置单击，修改的值将被写入芯片中。

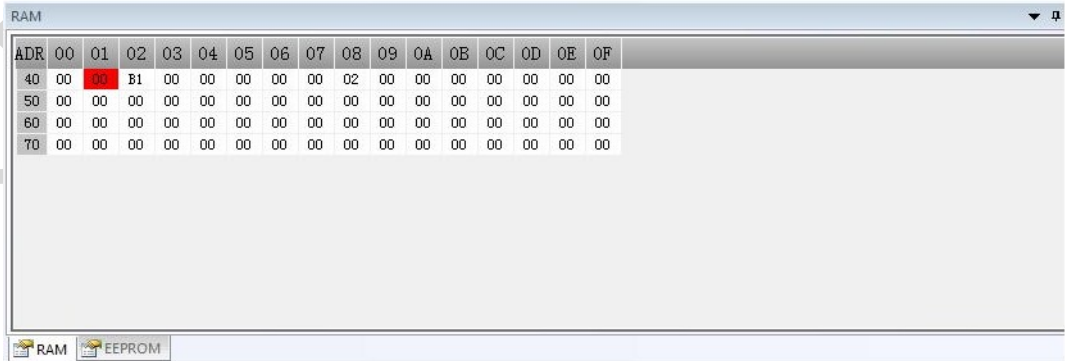


图 2-5. RAM 和 EEPROM 输出区

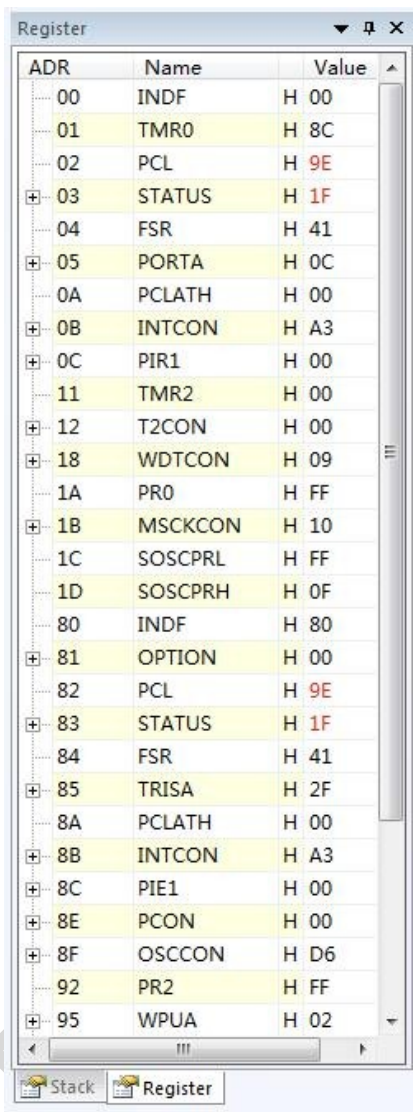


图 2-6. Stack 和 Register 观察区

2.5 菜单、工具条、状态栏

菜单包含所有操作，工具条对应常用的菜单项，状态栏显示进度条，芯片状态以及调试器版本号，如图 2-7。



图 2-7. 状态栏

3 新建工程

开发环境采用工程管理模式，每个项目都需要先建一个工程，选择 **Project** → **New Project**，弹出新建工程对话框，如图 3-1 所示。填入工程名（**Project Name**），选择工程路径（**Project Patch**），器件（**Device**）选择 **CMT60F02X**，如果需要创建工程文件夹，请勾选 **Create directory for project**。

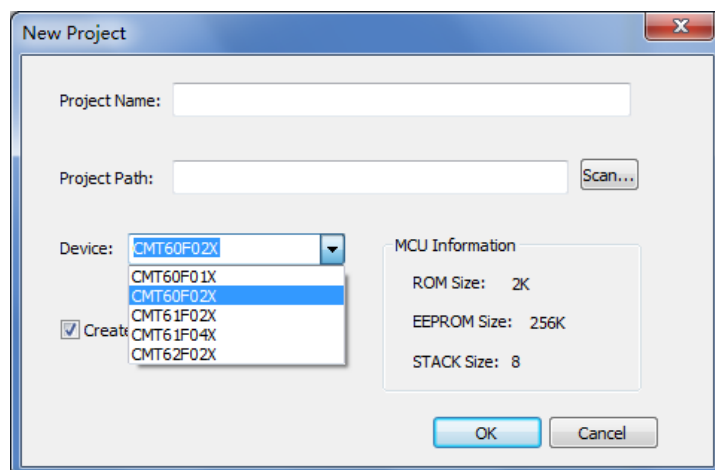


图 3-1. 新建工程


新建工程完成后，需添加源文件，有两种方式添加，一种是菜单选择 **Project** → **Add File...**，另一种是工作区选择 **Source File**，单击鼠标右键，在弹出的菜单中选择 **Add File...**。

注意：如果添加的文件不在工程目录下，将自动拷贝到工程目录下。

除了添加源文件，还可以添加 **Data** 文件。有两种方式添加，一种是菜单选择 **Project** → **Load EEPROM Data...**，另一种是工作区选择 **EE Data File**，单击鼠标右键，在弹出的菜单中选择 **Add File...**。

注意：如果添加的文件不在工程目录下，将自动拷贝到工程目录下。**Include File** 和 **Output File** 编译后自动生成，不需要添加。

4 编译

选择菜单 **Debug** → **Build** (F7) 或者点击  编译程序，编译前会弹出 **Option** 对话框如图 4-1，选择对应的选项，单击确定后开始编译，编译信息在 **Build** 窗口显示。如果有错误，双击错误信息可以定位到错误行。编译器说明参见附录 1。

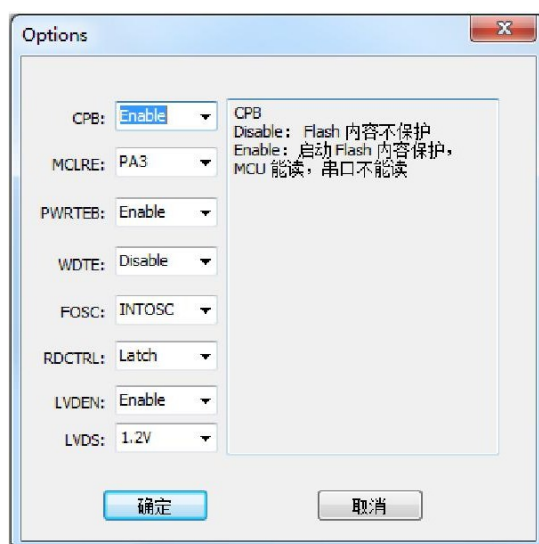


图 4-1. Options


5 EEPROM 设置

EEPROM 有三种方式导入，第一种是菜单 Project → Load EEPROM Data...，第二种是 DE 指令定义数据，第三种是双击 EEPROM 面板输入。

当菜单导入了 Data 文件，同时 DE 定义了数据，那么编译后 DE 定义将覆盖文件导入的数据。第三种方式只能在开始仿真后双击 EEPROM 才能输入，双击修改后立即写入 EEPROM，并回读显示。

选择菜单 Project → Export EEPROM Data...，可以导出当前 EEPROM 面板里的数据，注意：EEPROM 的数据是跟程序运行相关的。


6 下载

选择菜单 **Debug** → **Bulid All** (Ctrl+F7) 或者点击, 进行擦除、下载和复位。下载成功后, 仿真工具条变为有效, 开始仿真后如果编辑文件且保存后, 仿真工具条变为无效, 需重新编译下载后方有效。在选择下载前程序未编译的话, 会先编译再进行擦除、下载和复位。

CMOSTEK Confidential

7 仿真

7.1 设置断点

第一种设置方式，鼠标单击选择需要设置的语句行，然后选择菜单 **Debug** → **BreakPoint (F9)** 或者点击 ，第一次设置为添加断点，再次设置为删除断点。第二种设置方式，在编辑区的行号上单击鼠标左键设置断点，再次点击删除断点。断点以黄色圆圈标注。目前 IDE 支持的器件只允许设置一个断点，因此每次设置一个新断点时将自动删除上一个断点。

7.2 Reset (Ctrl+F5)

复位器件，PC 恢复至 0。复位完成，刷新显示 RAM、EEPROM、STACK、Register，并在状态栏显示芯片状态。

7.3 Step Into (F11)

单步，当执行 Call 指令时，进入子函数。执行完成，刷新显示 RAM、EEPROM、STACK、Register，并在状态栏显示芯片状态，并在状态栏显示芯片状态。

7.4 Step Over (F10)

单步，当执行 Call 指令时，不进入子函数。执行完成，刷新显示 RAM、EEPROM、STACK、Register，并在状态栏显示芯片状态。

7.5 Run to Cursor (Ctrl+F10)

运行到光标处，执行前需要先将光标停留到需要停止的语句行，程序运行到光标处停止。如果光标处无法到达，程序将全速运行。执行完成，刷新显示 RAM、EEPROM、STACK、Register，并在状态栏显示芯片状态。

7.6 Go (F5)

运行到断点处，执行前需先设置断点，程序运行到断点处停止。如果断点处无法到达，程序将全速运行。执行完成，刷新显示 RAM、EEPROM、STACK、Register，并在状态栏显示芯片状态。

7.7 Stop (Shift+F5)

停止运行，停止全速运行的程序。执行完成，刷新显示 RAM、EEPROM、STACK、Register，并在状态栏显示芯片状态。

CMOSTEK Confidential

8 附录 1（编译器说明）

8.1 变量命名规则

必须以字母开头，由字母、数字和下划线组成，不区分大小写，且变量名不能为关键字，关键字包含指令和伪指令。

8.2 数字进制表示

支持二进制、十六进制和十进制，表示方法如下：

- 二进制：

- 1) 以“B”结束的 0/1 字符，例如：00010110B
- 2) 以“B”开始，以”结尾的 0/1 字符，例如：B”00010110”

- 十六进制：

- 1) 以“H”结束的十六进制数，例如：1FH
- 2) 以“0x”开始的十六进制数，例如：0x1F
- 3) 以“H”开始，以”结束的十六进制数，例如：H”1F”

- 十进制：

- 1) 不含前后缀的数，就是十进制，例如：16
- 2) 以“.”开始的十进制数，例如：.10

8.3 地址标号

标号名遵循变量命名规则，同一个名字不能重复定义。标号后面可以带冒号“:”，也可以不带冒号“:”，所有非指令且未定义的单字符串，将被识别为标号。未带冒号的标号不可以和指令在同一行，带冒号的标号可以和指令在同一行。

8.4 指令

CMT	PIC	指令周期	Function	Operation	Status
BCR R, b	BCF F, b	1	Bit clear	$0 \rightarrow R(b)$	NONE
BSR R, b	BSF F, b	1	Bit set	$1 \rightarrow R(b)$	NONE
BTSC R, b	BTFSC F, b	1 (2)	Bit test, skip if 0	Skip if $R(b)=0$	NONE
BTSS R, b	BTFSS F, b	1 (2)	Bit test, skip if 1	Skip if $R(b)=1$	NONE
NOP	NOP	1	No operation	None	NONE
CLRWDT	CLRWDT	1	Clear WDT	$0 \rightarrow WDT$	/PF, /TF
SLEEP	SLEEP	1	Enter Sleep Mode	$0 \rightarrow WDT$, STOP OSC	/PF, /TF
STTMD	OPTION	1	Store W TO TMODE	$W \rightarrow TMODE$	NONE
CTLIO R	TRIS R	1	Control IO direction reg	$W \rightarrow IODIRr$	NONE
STR R	MOVWF F	1	Store W to reg	$W \rightarrow R$	NONE
LDR R, d	MOVF F, d	1	Load reg to d	$R \rightarrow d$	Z
SWAPR R, d	SWAPF F, d	1	Swap halves reg	$[R(0-3)R(4-7)] \rightarrow d$	NONE
INCR R, d	INCF F, d	1	Increment reg	$R+1 \rightarrow d$	Z
INCRSZ R, d	INCFSZ F, d	1 (2)	Increment reg, skip if 0	$R+1 \rightarrow d$	NONE
ADDWR R, d	ADDWF F, d	1	Add W and reg	$W+R \rightarrow d$	C, HC, Z
SUBWR R, d	SUBWF F, d	1	Sub W from reg	$R-W \rightarrow d$ or $R+W+1 \rightarrow d$	C, HC, Z
DECR R, d	DECf F, d	1	Decrement reg	$R-1 \rightarrow d$	Z
DECRSZ R, d	DECFSZ F, d	1 (2)	Decrement reg, skip if 0	$R-1 \rightarrow d$	NONE
ANDWR R, d	ANDWF F, d	1	AND W and reg	$R \& W \rightarrow d$	Z
IORWR R, d	IORWF F, d	1	Inclu.OR W and reg	$W R \rightarrow d$	Z
XORWR R, d	XORWF F, d	1	Exclu.OR W and reg	$W \wedge R \rightarrow d$	Z
COMR R, d	COMF F, d	1	Complement reg	$/R \rightarrow d$	Z
RRR R, d	RRF F, d	1	Rotate right reg	$R(n) \rightarrow R(n-1)$, $C \rightarrow R(7)$, $R(0) \rightarrow C$	C
RLR R, d	RLF F, d	1	Rotate left reg	$R(n) \rightarrow R(n+1)$, $C \rightarrow R(0)$, $R(7) \rightarrow C$	C
CLRW	CLRW	1	Clear working reg	$0 \rightarrow W$	Z
CLRR R	CLRF F	1	Clear reg	$0 \rightarrow R$	Z
RETI	RETFIE	2	Return from interrupt	Stack $\rightarrow PC$, $1 \rightarrow GIE$	NONE
RET	RETURN	2	Return from subroutine	Stack $\rightarrow PC$	NONE
LCALL N	CALL k	2	Long CALL subroutine	$N \rightarrow PC$, $PC+1 \rightarrow Stack$	NONE
LJUMP N	GOTO	2	Long JUMP address	$N \rightarrow PC$	NONE
LDWI I	MOVLW k	1	Load immediate to W	$I \rightarrow W$	NONE
ANDWI I	ANDLW k	1	AND W and imm	$W \& I \rightarrow W$	Z
IORWI I	IORLW k	1	Inclu.OR W and imm	$W I \rightarrow W$	Z
XORWI I	XORLW k	1	Exclu.OR W and imm	$W \wedge I \rightarrow W$	Z
RETW I	RETLW k	2	Return, place imm to W	Stack $\rightarrow PC$, $I \rightarrow W$	NONE
ADDWI I	ADDLW k	1	Add imm to W	$W+I \rightarrow W$	C, HC, Z
SUBWI I	SUBLW k	1	Subtract W from imm	$I-W \rightarrow W$	C, HC, Z

8.5 伪指令

8.5.1 ORG

格式: ORG ADDR

说明: 定义 PC 地址, ADDR 不能小于当前 PC, 也不能大于最大 PC。

例如:

```
ORG     0000H
Goto    START
ORG     0004H             ;中断入口
JUMP    INTtimer0
```

8.5.2 Include

格式: #include<文件名>、#include “文件名”

说明: <文件名>为系统目录下的文件,“文件名”为工程目录下的文件,文件类型可以是“.H”或“.HIC”的头文件,也可以是“.ASM”的源文件。头文件必须在 PC 地址为 0 之前引入,源文件可以在文件任意位置引入。

例如:

```
#INCLUDE <CMT2189B.inc>
#include "LED.ASM"
```

8.5.3 EQU

格式: 变量名 EQU RAM 地址

说明: 变量名遵循变量命名规则,同一个变量名不允许对应多个地址,多个变量允许对应同一个地址,注意当变量没有被使用时,不检查 RAM 地址是否有效。

例如:

```
LEDLEVEL     EQU   0x40
```

8.5.4 DB

格式: 变量名 DB ?

说明: 变量名遵循变量命名规则,同一个名字不能重复定义。不指定对应 RAM 地址,由编译器自动分配地址。注意当变量没有使用时不分配地址

例如:

```
V1   DB   ?
```

8.5.5 DE

格式: DE Data0, Data1,Datan

说明: Data EEPROM 数据表,数据表的地址必须在 0x4100 以后,DE 表后的数据从 ORG 定义的地址开始顺序排放,数据个数无限制,但是必须在同一行。

例如:

```
ORG     4110H
```

```
DE    0x10, 0x11, 0x12, 0x13, 0x14, 0x15, 0x16, 0x17
DE    0x18, 0x19, 0x1A, 0x1B, 0x1C, 0x1D, 0x1E, 0x1F
```

8.5.6 DBIT

格式: 变量名 DBIT ?

说明: 定义位变量, 变量名遵循变量命名规则。不指定对应 RAM 地址, 由编译器自动分配地址。注意当变量没有使用时不分配地址。该指令定义的变量只能用于位操作指令。位操作指令包含: BCR、BSR、BTSC、BTSS, BCF、BSF、BTFSC、BTFSS

例如:

```
V1_1    DBIT ?
```

8.5.7 Define

格式: #define 标识符 字符串

说明: 无参数的宏定义, 宏定义是用宏名来表示一个字符串, 以该字符串取代宏名, 这只是一种简单的代换, 字符串中可以含任何字符, 可以是常数, 也可以是表达式, 预处理程序对它不作任何检查。如有错误, 只能在编译已被宏展开后的源程序时发现。

例如:

```
#define  Defname    1+5
ORG      0000H
.....
LDWI     Defname
```

8.5.8 MARCO

格式: 宏名 MARCO par1.....parn
;宏内容
ENDM

说明: 带参数的宏, 宏名和参数名遵循变量命名规则, 以 ENDM 结束宏模块。在宏定义内不允许有标号。

例如:

```
Delayms macro a1, a2, a3
LDWI a1
STR DELAYCNT1
LDWI a2
STR DELAYCNT2
LDWI a3
STR DELAYCNT3
CALL DELAYLOOP
Endm

ORG 0000H
.....
Delayms 0xF0, 0x49, 0x30
```

.....

8.5.9 Ifdef

```
格式:   ifdef      条件宏
        ;程序段 1
        else
        ;程序段 2
        endif
```

说明: 如果条件宏不等于 0, 执行程序段 1, 否则执行程序段 2。注意该指令不能嵌套使用。

例如:

```
        ifdef      defname
        LJUMP      RESTART_WDT
        DECRSZ     DELAYCNT1, F
        LJUMP      POWERDOWN_2SLOOP
        else
        DECRSZ     DELAYCNT2, F
        LJUMP      POWERDOWN_2SLOOP
        DECRSZ     DELAYCNT3, F
        LJUMP      POWERDOWN_2SLOOP
        endif
```

9 附录 2（单片机调试说明书）

9.1 概述

本文档旨在描述并定义产品集成开发环境中 IDE 开发板与外界的连接界面，包括相关硬件和软件接口。

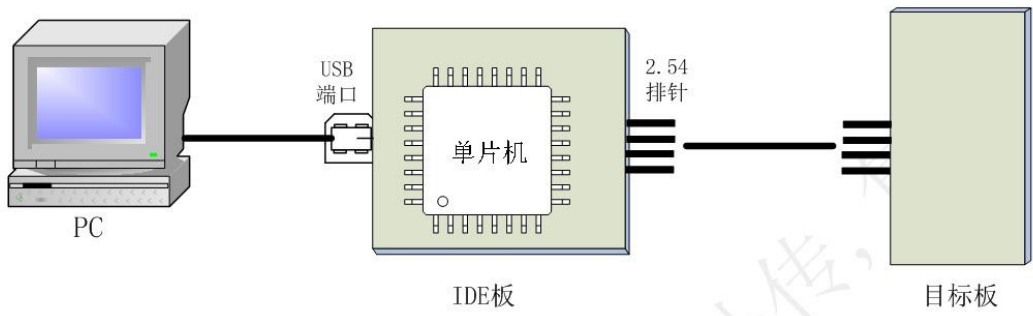
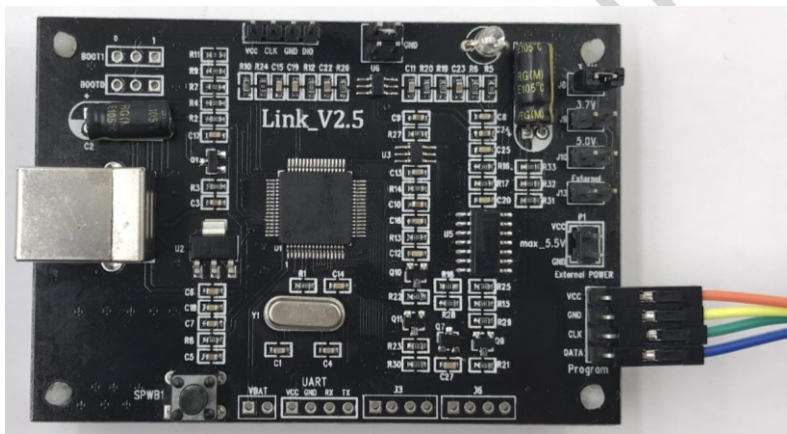


图 9-1. 开发环境搭建示意图



9.2 线路说明

以下是图 9-3 的接口说明：

- USB1: USB 口与 PC 连接
- J11: Program 目标芯片通信控制端口，从上往下分别是：

表 9-1. 调试接口和各芯片的连接关系表

J11 接口	CMT2280F2	CMT2281F2	CMT2189B	CMT2189C
VCC	DVDD Pin13	DVDD Pin13	DVDD/AVDD Pin1/Pin6	VDD Pin1/Pin11
GND	GND Pin2	GND Pin3	GND Pin7	GND Pin2/Pin10/Pin13
CLK	PA0/ICSPCLK Pin9	PA0/ICSPCLK Pin9	PA0/ICSPCLK Pin12	PA0/ICSPCLK Pin8
DATA	PA1/ICSPDAT Pin10	PA1/ICSPDAT Pin10	PA1/ICSPDAT Pin13	PA1/ICSPDAT Pin9

- P1: External Power 外部电源输入口
- J7: 下载板 GND
- J8: 目标 MCU 3.3V 供电选择
- J9: 目标 MCU 3.7V 供电选择
- J10: 目标 MCU 5V 供电选择（仅 CMT2281F2 适用）
- J13: 外部输入电源供电选择

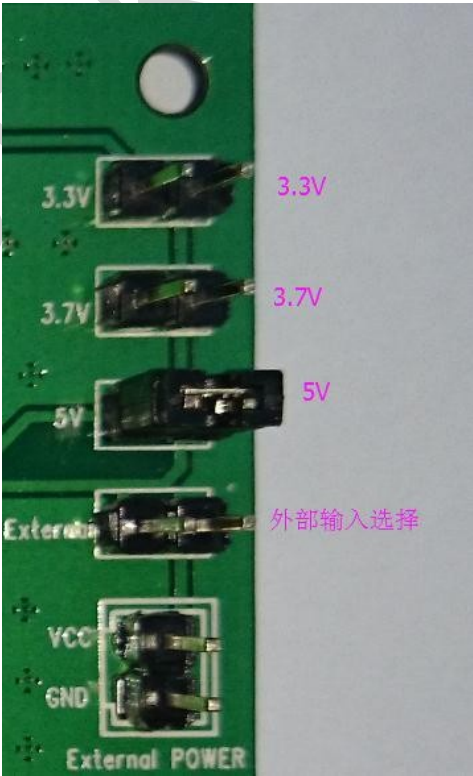


图 9-4. 调试板供电接口

注意:

- J8、J9、J10、J13 四个短路点有且仅能短接其中一个;
 - SPWB1: 下载板复位按钮
 - D1: 指示灯
- 红色 LED (常亮): 上电指示, 这时调试器没有与 PC 通信成功;
- 绿色 LED (常亮): 调试器与 PC 已经连接成功;
- 黄色 LED (常亮): 调试器处于 BOOT 模式, 等待在线更新固件;
- 黄色 LED (闪烁): 调试器在工作中

9.3 注意事项

1. PC host 与 IDE 开发板之间的连接很简单, 即标准的 USB 接口, 使用调试器不需要额外安装其他驱动, 第一次使用插上 PC 系统会自动安装驱动, 等驱动安装完成即可正常使用。
2. 调试器板的全部供电都有 PC 经 USB 口 5V 供电, 用户使用调试器板对目标芯片/板进行调试的时候可以由调试器直接供电, 供电电压可通过短路帽切换调整, 板卡可供选择的电压有 3.3V、3.7V 和 5V, 同一时间只能短路其中一个短路帽。如果用户需要外部供电, 则需要把目标板的电源断开, 然后把这个外部电源接到调试器板的 External Power 接口, 然后再经过 Program 接口的 VCC 把电源送给目标板, 这个 External Power 接口的最大输入电压不能超过 12V。
3. 每次切换电源或者更换目标芯片之后都需要按一下调试器板上的复位按键, 对目标芯片的重新连接, 等调试器板上的 LED 灯变为绿色则表示与上位机通讯成功, 这个时候才可以开始操作上位机。
4. 每次上电上位机会检测固件版本号, 如果发现版本过低系统会跳入 Boot 模式, 这个时候 LED 灯常亮黄色上位机会自动更新固件, 中途不可以把 USB 断开, 如果中途断开或者固件更新失败则需重新按复位键或者重新拔插 USB 则可以再次进入 Boot 模式, 重新更新固件。

10 文档变更记录

表 10. 文档变更记录表

版本号	章节	变更描述	日期
1.0	全部	初始版本发布	2017-11-25
1.1	2	更新部分 IDE 截图	2017-11-28

CMOSTEK Confidential

11 联系方式

无锡泽太微电子有限公司深圳分公司

中国广东省深圳市南山区前海路鸿海大厦 203 室

邮编: 518000

电话: +86 - 755 - 83235017

传真: +86 - 755 - 82761326

销售: sales@cmostek.com

技术支持: support@cmostek.com

网址: www.cmostek.com

Copyright. CMOSTEK Microelectronics Co., Ltd. All rights are reserved.

The information furnished by CMOSTEK is believed to be accurate and reliable. However, no responsibility is assumed for inaccuracies and specifications within this document are subject to change without notice. The material contained herein is the exclusive property of CMOSTEK and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of CMOSTEK. CMOSTEK products are not authorized for use as critical components in life support devices or systems without express written approval of CMOSTEK. The CMOSTEK logo is a registered trademark of CMOSTEK Microelectronics Co., Ltd. All other names are the property of their respective owners.