



FMDIDE 用户指南

V1.0.0

Shenzhen Chipkim-Tech co., Ltd.



1. FMDIDE 简介

FMDIDE 是为 FMD 全系列单片机用户提供的开发软件，它的主要功能是调试、仿真 FMD 系列单片机的程序。目前（2015 年 10 月）版本是 V1.0.0，它将随着新器件的推出而不断更新。

FMDIDE 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 FMD 的调试器，下载、调试可执行程序；
- 在观察窗查看变量

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。

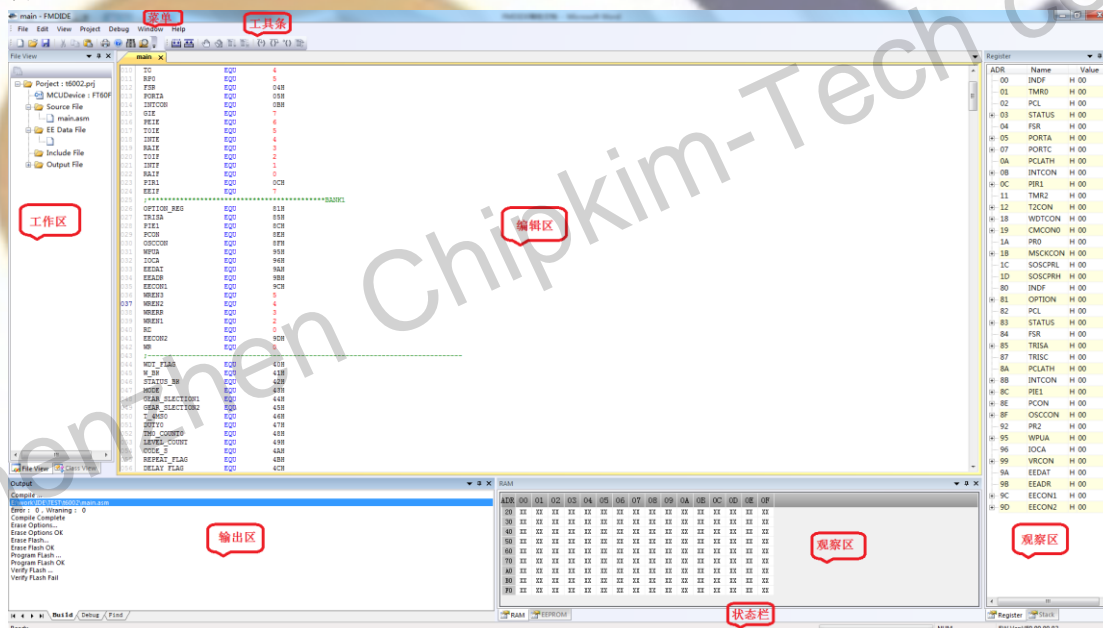


图 2-1 主界面

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。

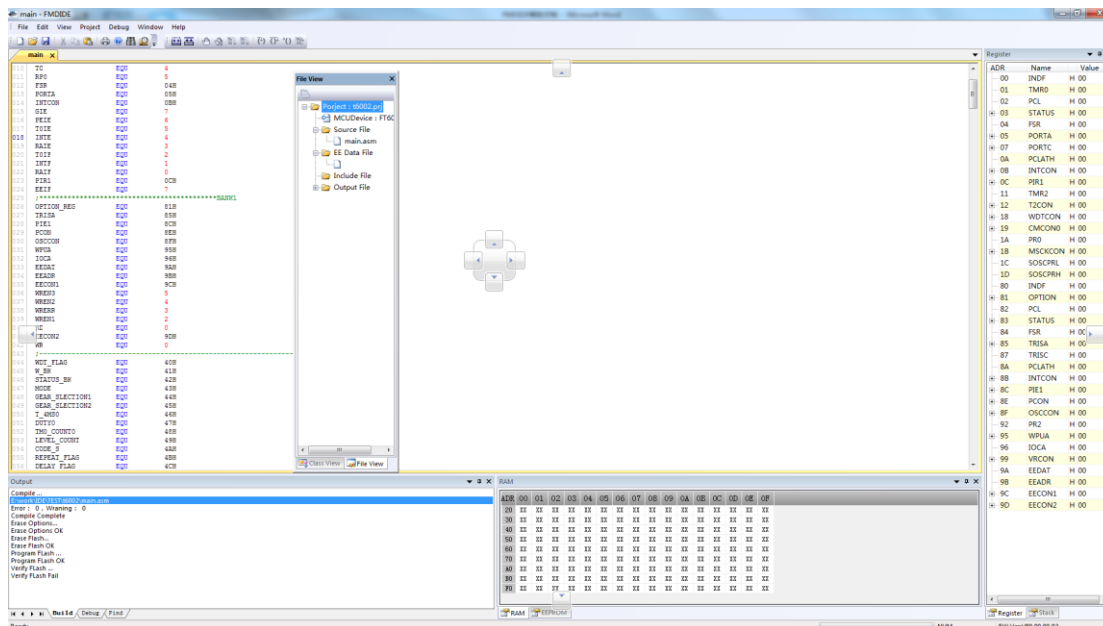


图 2-2 窗体停靠

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。

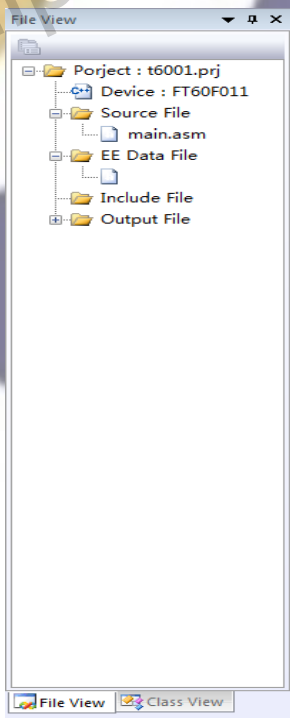


图 2-3 工作区



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。

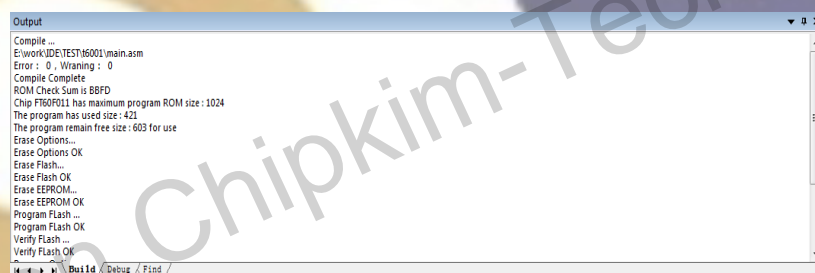


图 2-4 输出区

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。



| ADR | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F |
|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 40 | 00 | 00 | B1 | 00 | 00 | 00 | 00 | 00 | 02 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 50 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 60 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 70 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

图 2-5 RAM 和 EEPROM 观察区

| ADR | Name | Value |
|-----|---------|-------|
| 00 | INDF | H 00 |
| 01 | TMR0 | H 8C |
| 02 | PCL | H 9E |
| 03 | STATUS | H 1F |
| 04 | FSR | H 41 |
| 05 | PORTA | H 0C |
| 0A | PCLATH | H 00 |
| 0B | INTCON | H A3 |
| 0C | PIR1 | H 00 |
| 11 | TMR2 | H 00 |
| 12 | T2CON | H 00 |
| 18 | WDTCON | H 09 |
| 1A | PRO | H FF |
| 1B | MSCKCON | H 10 |
| 1C | SOSCPRL | H FF |
| 1D | SOSCPRH | H 0F |
| 80 | INDF | H 80 |
| 81 | OPTION | H 00 |
| 82 | PCL | H 9E |
| 83 | STATUS | H 1F |
| 84 | FSR | H 41 |
| 85 | TRISA | H 2F |
| 8A | PCLATH | H 00 |
| 8B | INTCON | H A3 |
| 8C | PIE1 | H 00 |
| 8E | PCON | H 00 |
| 8F | OSCCON | H D6 |
| 92 | PR2 | H FF |
| 95 | WPUA | H 02 |

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

FMDIDE 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”和器件“Device”，如果需要创建工程文件夹，请勾选“Create directory for project”。

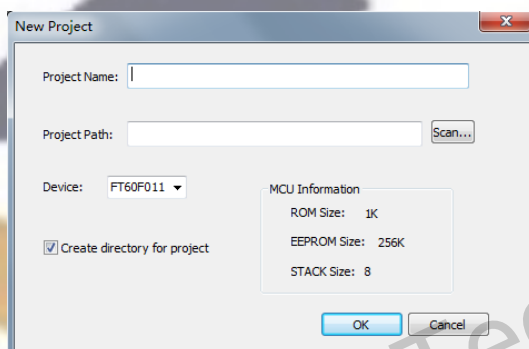


图 3-1 新建工程

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。

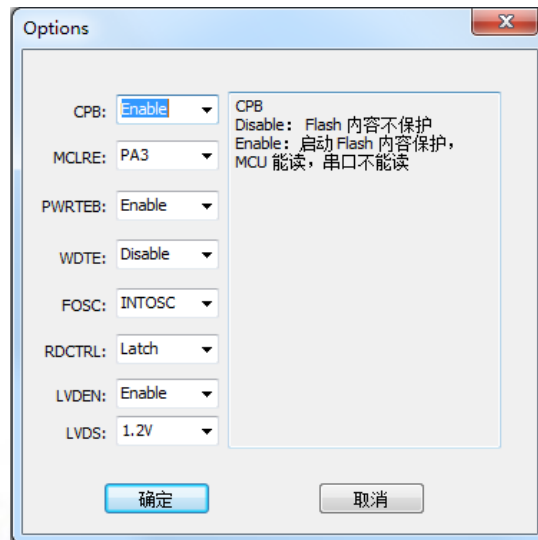


图 4-1 Options


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->Build 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。

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，再次单击删除断点。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



8. 附录 1 FMD 编译器说明

8.1 支持的器件

FT60F011

8.2 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。

8.3 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

8.4 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。



8.5 指令

| Mnemonic Operands | Function | Operation | Status |
|----------------------|--------------------------|--------------------------|----------|
| BCR R, b | Bit clear | 0-> R(b) | NONE |
| BSR R, b | Bit set | 1-> R(b) | NONE |
| BTSC R, b | Bit test, skip if 0 | Skip if R(b)=0 | NONE |
| BTSS R, b | Bit test, skip if 1 | Skip if R(b)=1 | NONE |
| | | | |
| NOP | No operation | None | NONE |
| CLRWDT | Clear WDT | 0-> WDT | /PF, /TF |
| SLEEP | ENTER SLEEP MODE | 0-> WDT, STOP OSC | /PF, /TF |
| STTMD | Store W TO TMODE | W-> TMODE | NONE |
| CTLIO R | Control IO direction reg | W-> IODIRr | NONE |
| STR R(MOVWF) | Store W to reg | W-> R | NONE |
| LDR R, d(MOVF) | Load reg to d | R-> d | Z |
| SWAPR R,d | Swap halves reg | [R(0-3)R(4-7)]-> d | NONE |
| INCR R, d | Increment reg | R+ 1-> d | Z |
| INCRSZ R, d | Increment reg, skip if 0 | R+ 1-> d | NONE |
| ADDWR R, d | Add W and reg | W+ R-> d | C, HC, Z |
| SUBWR R, d | Sub W from reg | R- W-> d R+ /W+ 1-> d | C, HC, Z |
| DECR R, d | Decrement reg | R- 1-> d | Z |
| DECRSZ R, d | Decrement reg, skip if 0 | R- 1-> d | NONE |
| ANDWR R, d | AND W and reg | R& W-> d | Z |
| IORWR R, d | Inclu.OR W and reg | W R-> d | Z |

8.6 伪指令

8.6.1 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

8.6.2 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:

```
#INCLUDE <FMD6001.inc>
#include "LED.ASM"
```

8.6.3 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: LEDLEVEL EQU 0X40

8.6.4 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 ?

8.6.5 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。

```
ORG 4110H
DE 0x10,0x11,0x12,0x13,0x14,0x15,0x16,0x17
DE 0x18,0x19,0x1A,0x1B,0x1C,0x1D,0x1E,0x1F
```



8.6.6 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:

```
V1_1 DBIT ?
```

8.6.7 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: #define Defname 1+5

```
ORG 0000H
```

```
.....
```

```
LDWI Defname
```

8.6.8.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。

例:

```
Delaysms macro a1,a2,a3
```

```
LDWI a1
```

```
STR DELAYCNT1
```

```
LDWI a2
```

```
STR DELAYCNT2
```

```
LDWI a3
```

```
STR DELAYCNT3
```

```
CALL DELAYLOOP
```

```
Endm
```

```
ORG 0000H
```

```
.....
```

```
Delaysms 0xF0,0x49,0x30
```



.....

8.8.9.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:

```
ifdef    defname
LJUMP  RESTART_WDT
DECRSZDELAYCNT1,F
LJUMP  POWERDOWN_2SLOOP
else
DECRSZDELAYCNT2,F
LJUMP  POWERDOWN_2SLOOP
DECRSZDELAYCNT3,F
LJUMP  POWERDOWN_2SLOOP
endif
```



9. 附录 2 FMDlink 单片机调试器说明书

9.1 概述

本文档旨在描述并定义 FMD MCU 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。

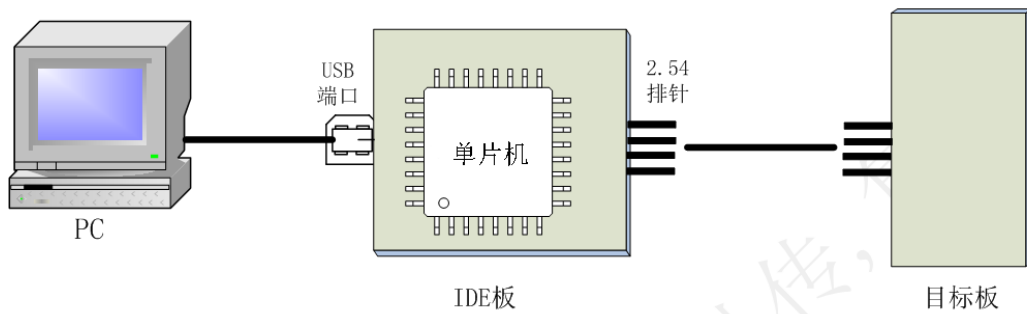


图 1.1 MCU 开发环境示意图

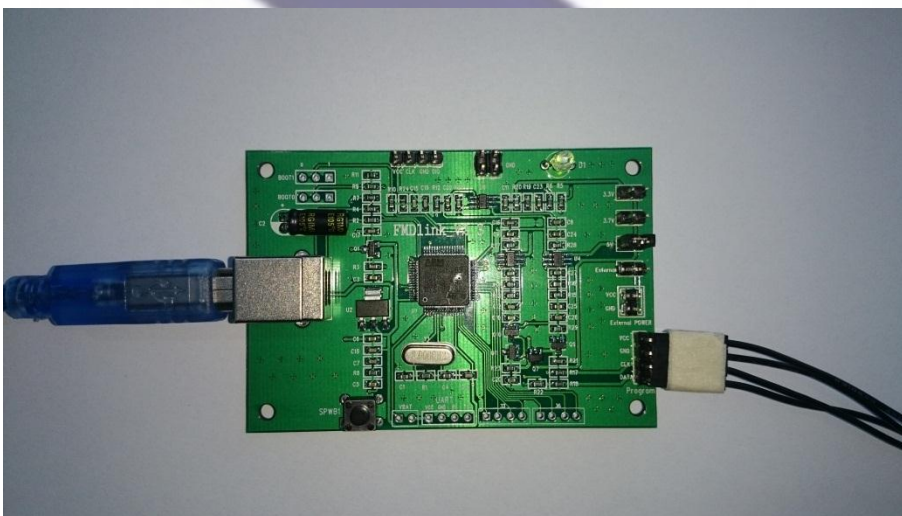
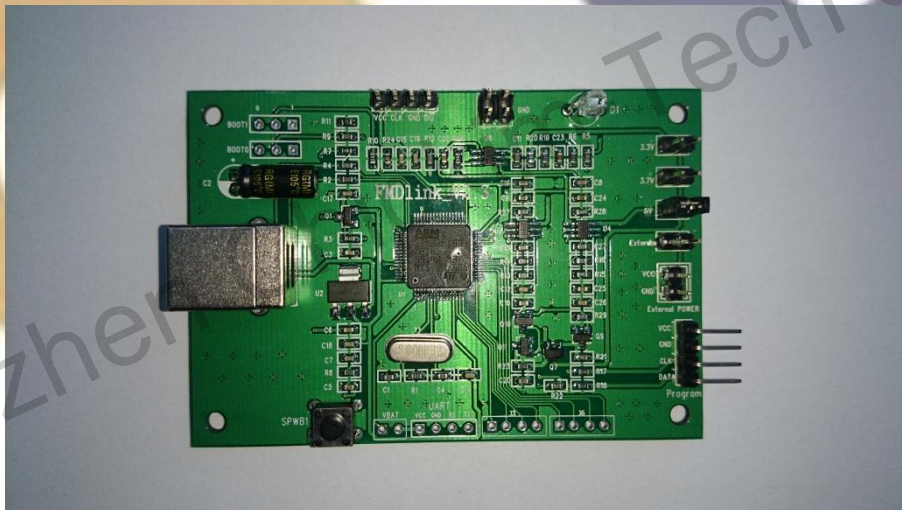


图 1.2 FMDlink

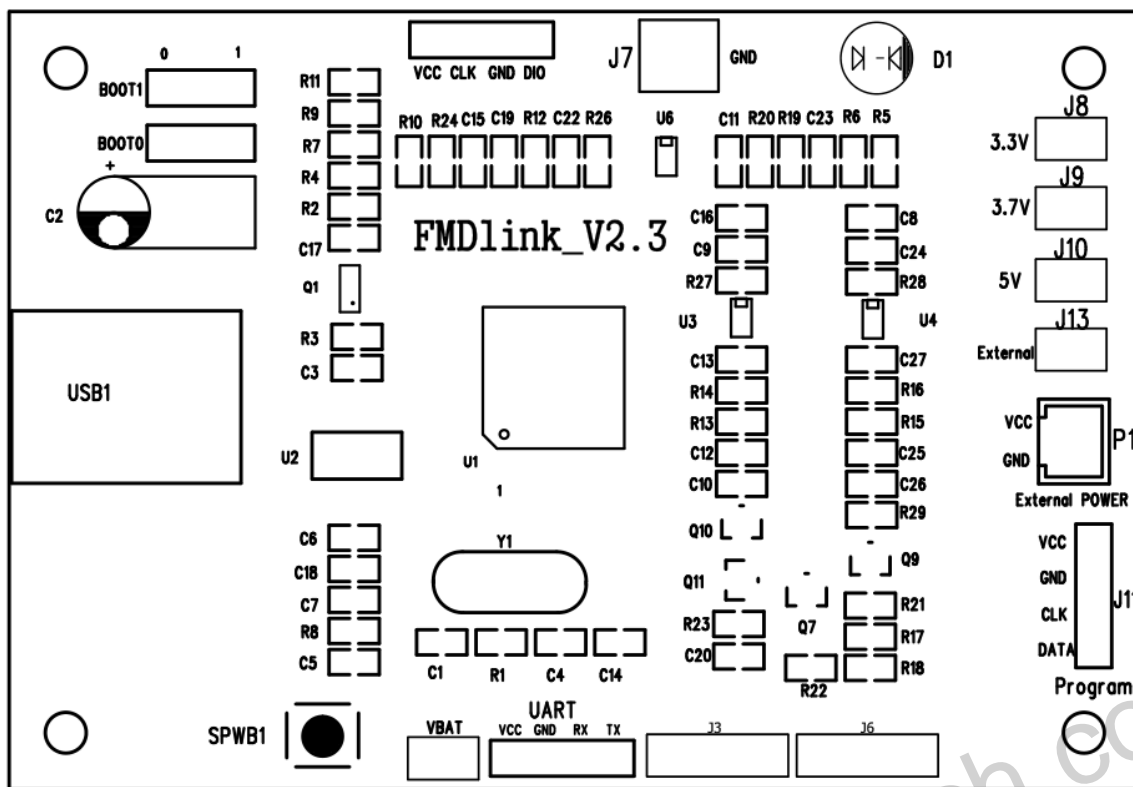


图 1.3 FMDlink Board 示意图

IDE 开发环境:上位机软件安装需要在 XP 及以上版本。

9.2.线路说明

以下是图 1.3 的接口说明

USB1:

USB 口与 PC 连接。

J11:

Program 目标 MCU 通信控制端口，从上往下分别是：

VCC ——> 目标 IC VCC

GND ——> 目标 IC GND

CLK ——> 目标 IC PA0

DATA ——> 目标 IC PA1

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

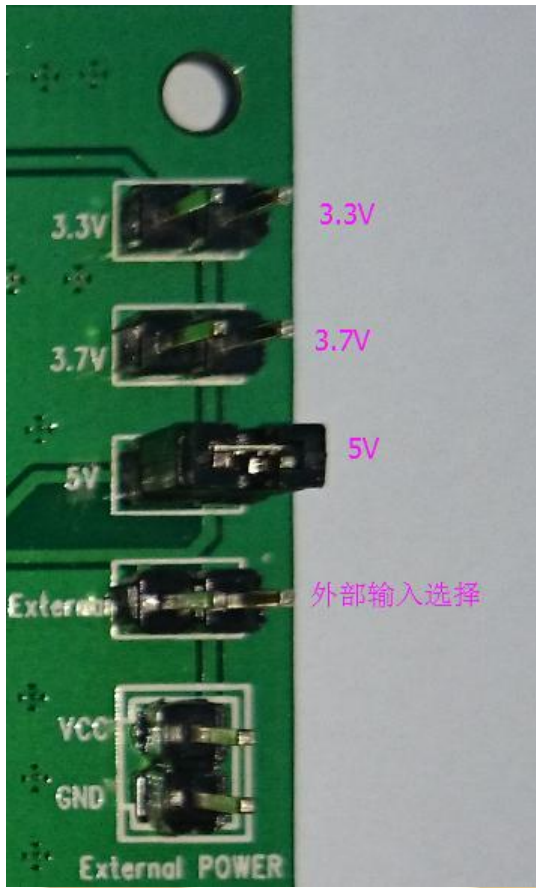
J10:

目标 MCU5V 供电选择

J13:



外部输入电源供电选择



(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

- 1、红色 LED (常亮): 上电指示, 这时 FMDlink 没有与 PC 通信成功
- 2、绿色 LED (常亮): FMDlink 与 PC 已连接
- 3、黄色 LED (常亮): FMDlink 处于 Boot 模式, 等待在线更新固件
- 4、黄色 LED (闪烁): FMDlink 正在工作

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 FMDlink 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、FMDlink 板的全部供电都有 PC 经 USB 口 5V 供电, 用户使用 FMDlink 板对目标 IC 进行调试的时候可以由 FMDlink 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 FMDlink 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 IC 之后都需要按一下 FMDlink 板上的复位按键, 对目标 IC 的重新连接, 等 FMDlink 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跟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。



深圳市智博金科技有限公司

Shenzhen Chipkim-Tech co.,Ltd.

网址：<http://www.chipkim.com>

TEL：+86 137 5104 0879（微信同步）

+86 133 9286 6434（微信同步）

Q Q：2851090001



扫一扫 关注公众号



扫一扫 关注阿里店铺