FPPA IDE

整合型发展系统软件

使用手册

第 1.02 版

2019 年 8 月 20 日

6F-6, No.1, Sec. 3, Gongdao 5th Rd., Hsinchu City 30069, Taiwan, R.O.C.

TEL: 886-3-572-8688 www.padauk.com.tw

重要声明

应广科技保留权利在任何时候变更或终止产品，建议客户在使用或下单前与应广科技或代理商联系以取得最新、最正确的产品信息。

应广科技不担保本产品适用于保障生命安全或紧急安全的应用，应广科技不为此类应用产品承担任何责任。关键应用产品包括，但不仅限于可能涉及的潜在风险之死亡、人身伤害、火灾或严重财产损失。

应广科技不承担任何责任来自于因客户的产品设计所造成的任何损失。在应广科技所保障的规格范围内，客户应设计和验证他们的产品。为了尽量减少风险，客户设计产品时，应保留适当的产品工作范围安全保障。

提供本文档的中文简体版是为了便于了解，请勿忽视文中英文的部份，因为其中提供有关产品性能以及产品使用的有用信息，应广科技暨代理商对于文中可能存在的差错不承担任何责任。

目　录

修订历史:

修 订	日 期	作 者	描 述
V1.00	2013/04/10	Kent	初版
V1.01	2013/05/10	Charley	更新 UI 图片
V1.02	2019/08/20	Kent	1. 更新图片、部分内容调整 2. 原文件拆分成 IDE_UM 和 Mini-C_UM

1. 简介

FPPA™ (Field Programming Processor Array) 整合型发展系统提供一个系统开发环境，供使用者轻松的开发程序、侦错系统。本系统包括一个仿真器(ICE)、一个烧录器(Writer)、以及一套整合型发展软件(IDE)。

仿真器(ICE)是用来模拟 PADAUK FPPA™ 各系列 IC 的动作，它的功能除了执行、单步执行、断点以及暂停等基本功能外，还提供处理单元阵列(processor array)内容监看以及存储器资料监看等功能。

程序烧录器(Writer)用来将使用者开发好的程序码烧入各系列 IC。

本文仅介绍 IDE 软件的使用方法，仿真器和烧录器的使用方法请参考其各自的使用手册。

FPPA IDE 整合型发展软件是一个视窗型软件（如图 1-1），包括程序编辑(editor)、组译(assembler)、侦错(debugger)以及程序烧录(program writer)等功能模块。

图 1-1. FPPA IDE

2. 整合发展系统软件 IDE

2.1. 整合发展系统软件安装

在与硬件器材连接之前，请按照下列步骤安装 IDE 软件以及 USB 驱动程序：

步骤 1：请从网站 (http//www.padauk.com.tw) 中取得安装程序 (Setup_IDE_0.xx.exe)。执行后，视窗画面将显示如图 2-1。

图 2-1. FPPA™ IDE 安装画面

步骤 2：完成安装。

注意： 在执行 Setup 程序前，如果用户的 PC 有系统管理员与访客两种模式，请登入系统管理员模式，并暂时关闭 IDE 程序，否则无法顺利安装；如果曾经安装过 IDE 程序，则无需进入系统管理员模式，可直接安装。
以下是一些常见错误。

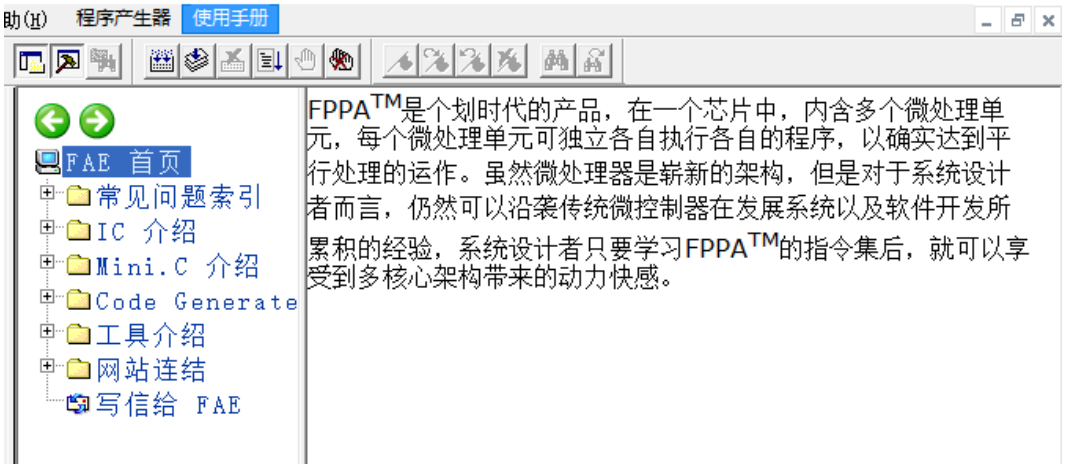
图 2-2. 请登入系统管理员模式。

图 2-3. 请先暂时关闭 IDE 程序。

3. 如何使用 FPPA IDE 整合型发展系统

FPPA IDE 的功能包括程序编辑(editor)，组译(assembler)，侦错(debugger)以及程序烧录(program writer)。在同一个视窗环境下，使用者可以轻易的完成系统程序设计、侦错以及烧录。

使用时如遇技术上的问题，用户可以从[Menu]->[求助(H)]->[使用手册]或[Menu]->[使用手册]中找到相关问题的解决办法。



3.1. 使用 IDE

成功安装软件后，在【桌面】上将会有个 "FPPA IDE"的快捷方式，如图 1-1 所示，双击即可执行程序。

IDE 发展系统不提供 soft simulation (软件模拟) 的功能。

IDE 的程序语言有两种格式可选择：

(1) 一般的 Asm 格式，程序是以应广科技的汇编语言为架构，并可混合 C 语言的语法。

(2) Mini-C 格式是以应广科技的 C 语言为架构，同时支援汇编语言的语法，它不但提供 C 的便利性，也拥有汇编语言的简洁高效。

3.2. 建立新的 Asm 项目

当第一次执行 IDE 时，用户必须建立一个新项目。请先从选单上开新项目，打开程序画面后，用户可以从选单 "**档案 -> 开新项目**" 来开始一个新的计划，也可以利用选单 "**档案 -> 开旧项目**"，来打开旧计划。图 3-1 所示为开新项目的选择画面。

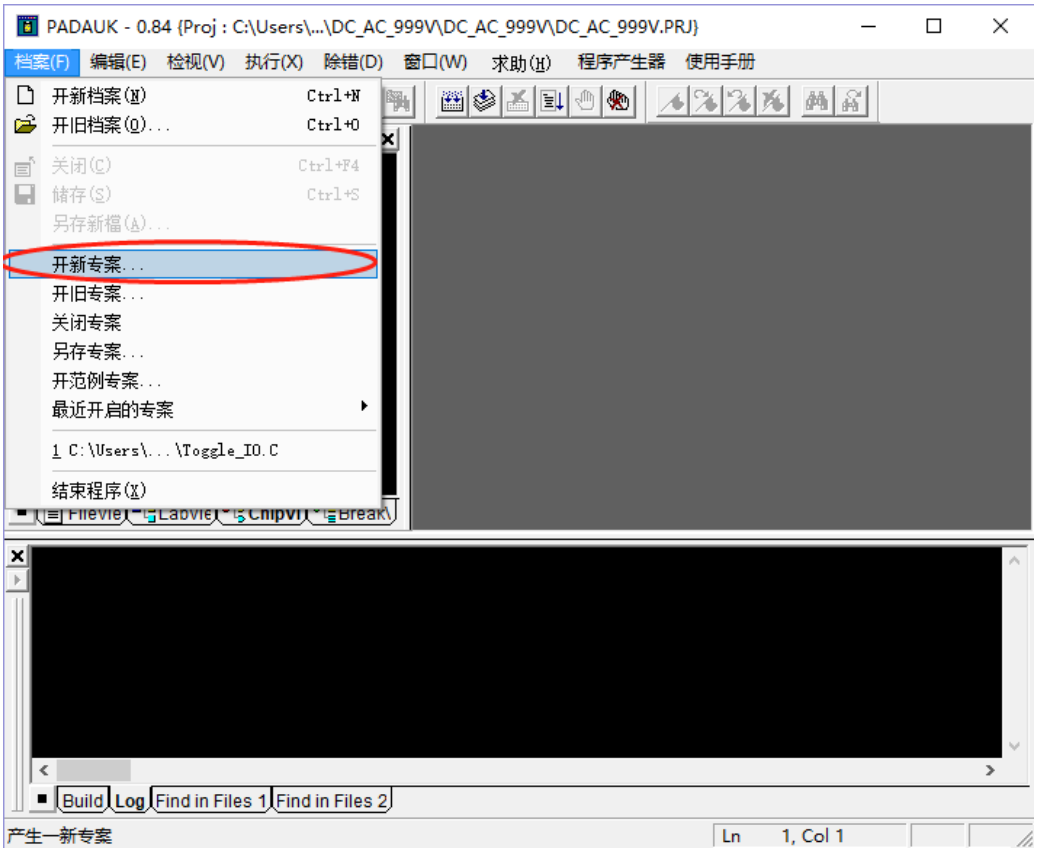


图 3-1 开新项目

当开启新项目后，将出现如图 3-2 所示对话盒：

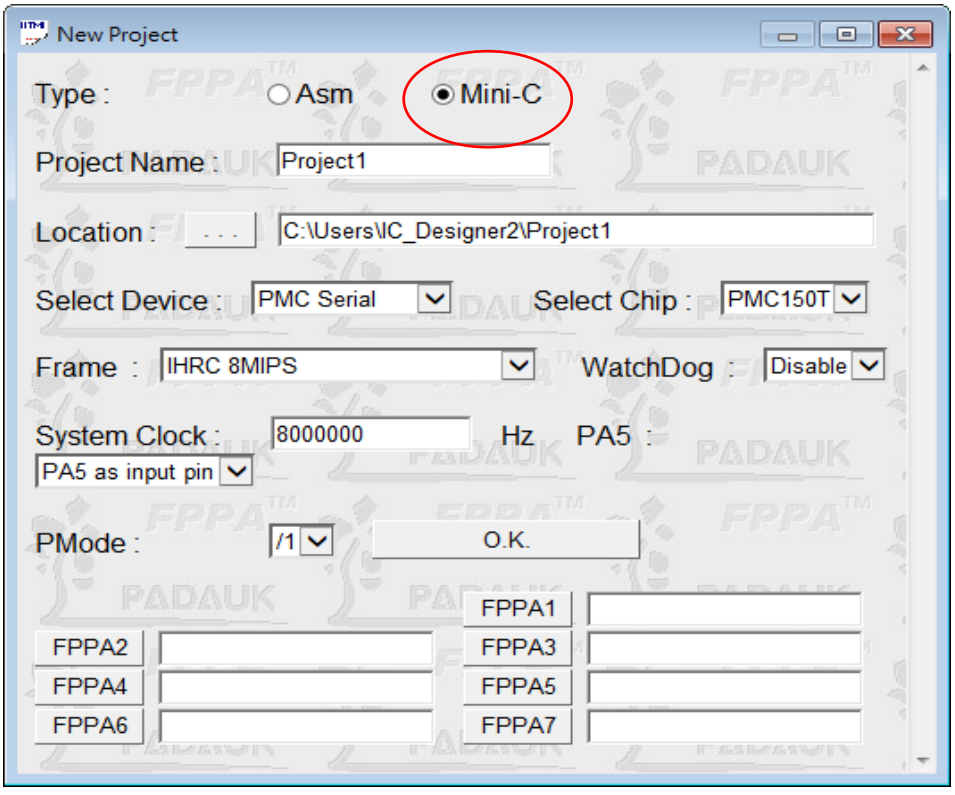


图 3-2. 开新项目 / 选择 Mini-C 新项目

这时用户有两种格式可选择：(1) 一般的 Asm 格式，程序以应广科技的汇编语言为主架构。 (2) Mini-C 格式是以应广科技的 C 语言为主架构。用户可点按 【Asm】与 【Mini-C】来做切换。图 3-3 是选择 Asm 项目的视窗画面。

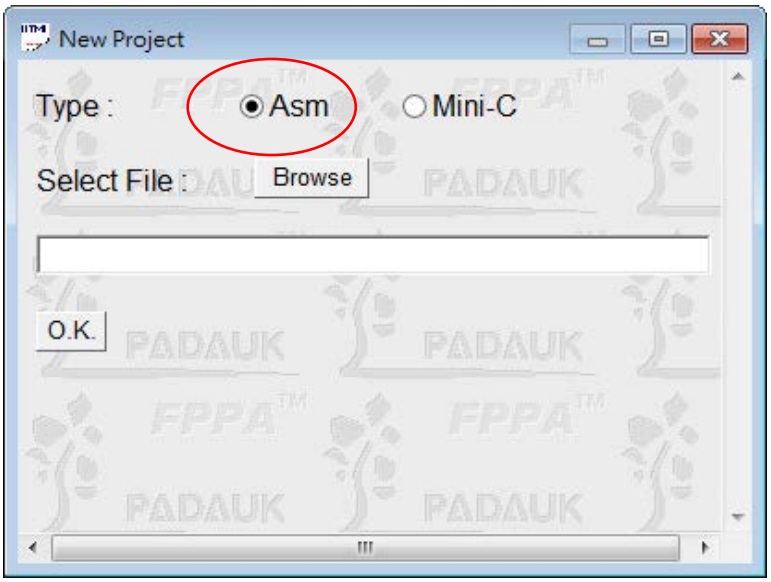


图 3-3. 选择 Asm 新项目

3.3. 在新项目内加入旧的 Asm 档

如果已经有旧的 ASM 档案，可选择 【Asm】 格式，并按 【Browse】 来选取旧档案，如图 3-4 所示。

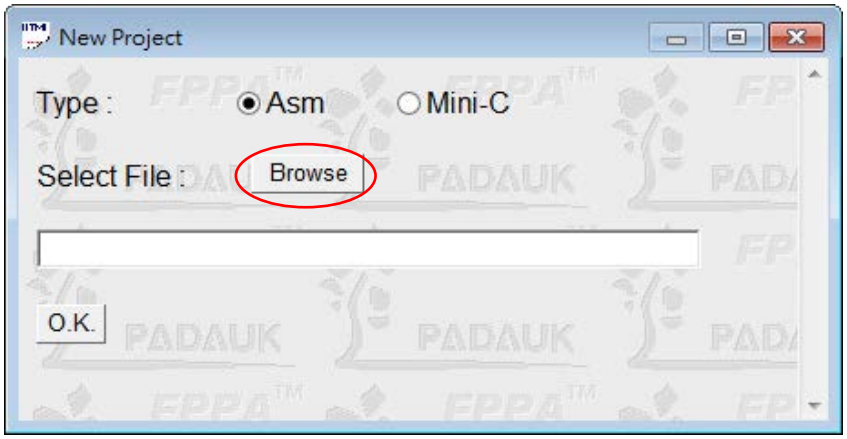


图 3-4. 在新项目内加入旧的 Asm 档案

接着出现档案选择视窗，如图 3-5 所示，请选取用户的 Asm 主档案。

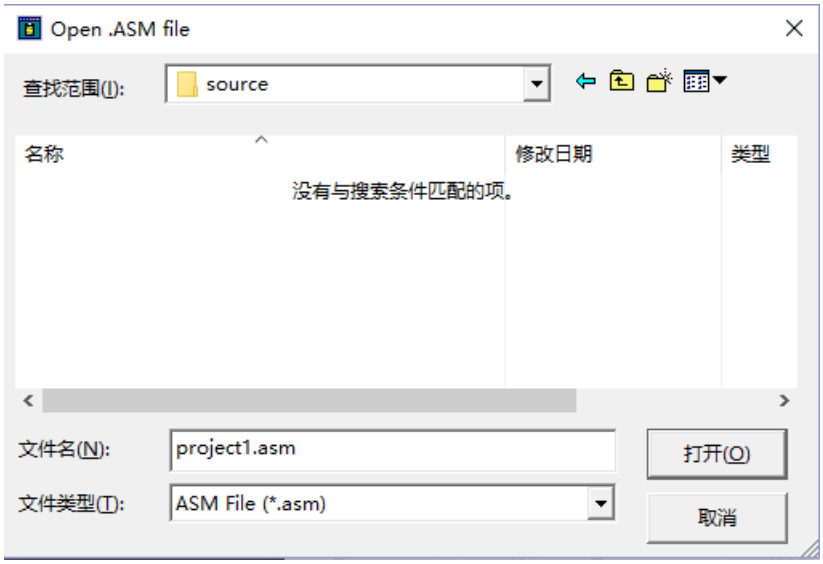


图 3-5. Asm 源文件选择

选好后，按下 【OK】，如图 3-6。

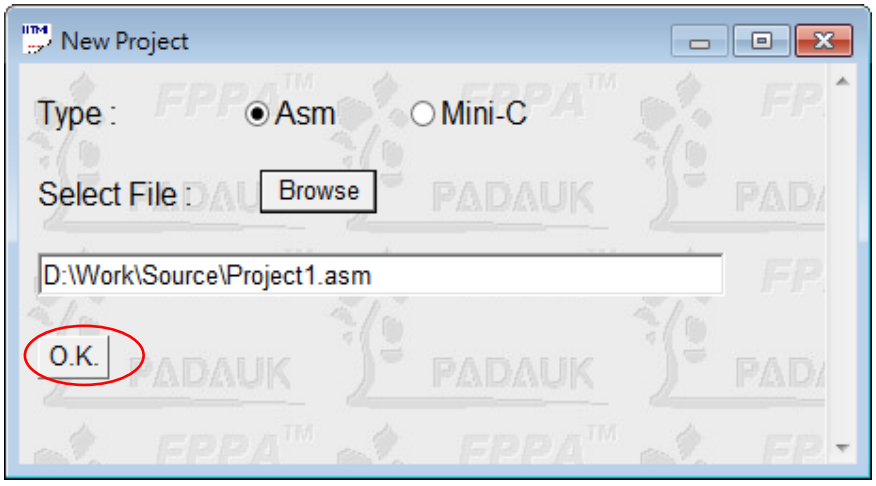


图 3-6. 档案选择后的视窗画面

如果用户选的档案并不存在，系统会询问是否要提供一个简易的 Demo Code，方便撰写时的参考，视窗如图 3-7。

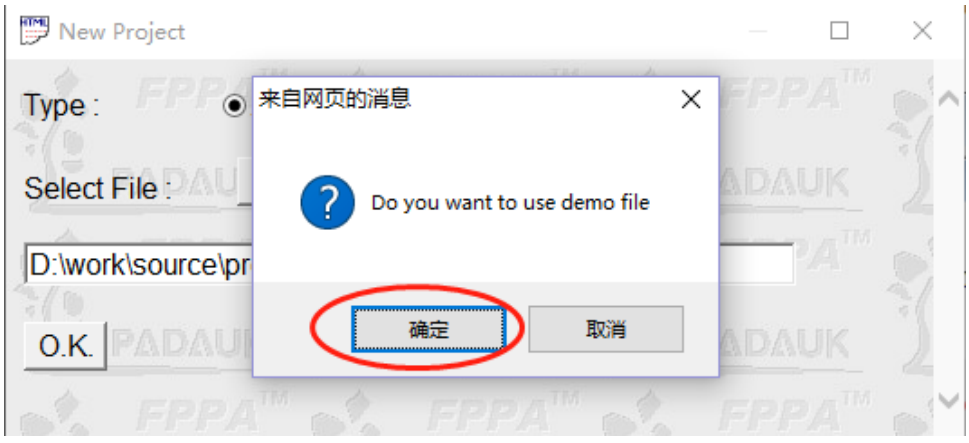


图 3-7. 提供简易 demo code 询问视窗

最后产生了一个新项目，视窗画面如图 3-8 所示：

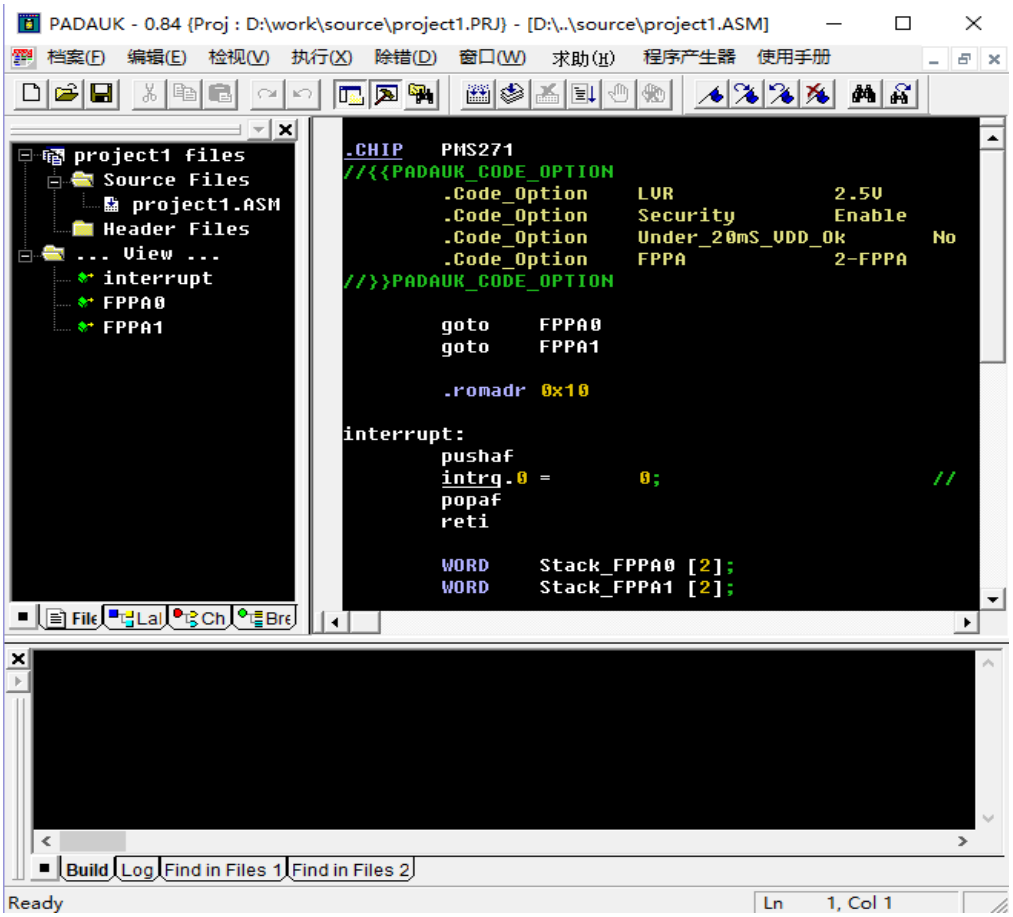


图 3-8. 新项目成立

3.4. 程序执行与侦错

程序撰写完毕后就可以开始侦错程序，请点击【建置(B)】开始程序建置。如图 3-9 所示：

图 3-9. 程序建置

程序选项 Code_Option 用来选择或修改系统设置，可从【执行】→【Code Options】打开，如图 3-10 所示：

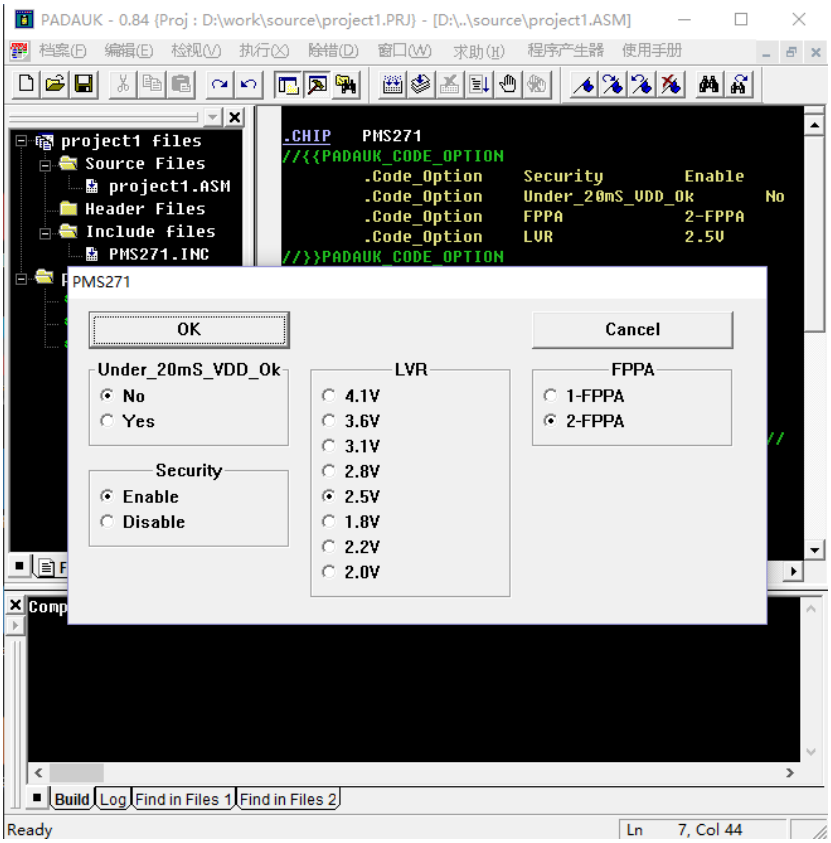


图 3-10. 程序选项 Code_Option

程序建置通过以后就可以进入侦错模式。使用快捷键 **F11** 可以单步执行程序，也可以点击窗口中的【继续执行(G)】（如图 3-11）让程序持续运行。

图 3-11. 程序继续执行

请注意：进入侦错模式之前请将 PC 与 ICE 连接进行仿真，否则系统会提示 ICE 未连接，如图 3-12 所示：

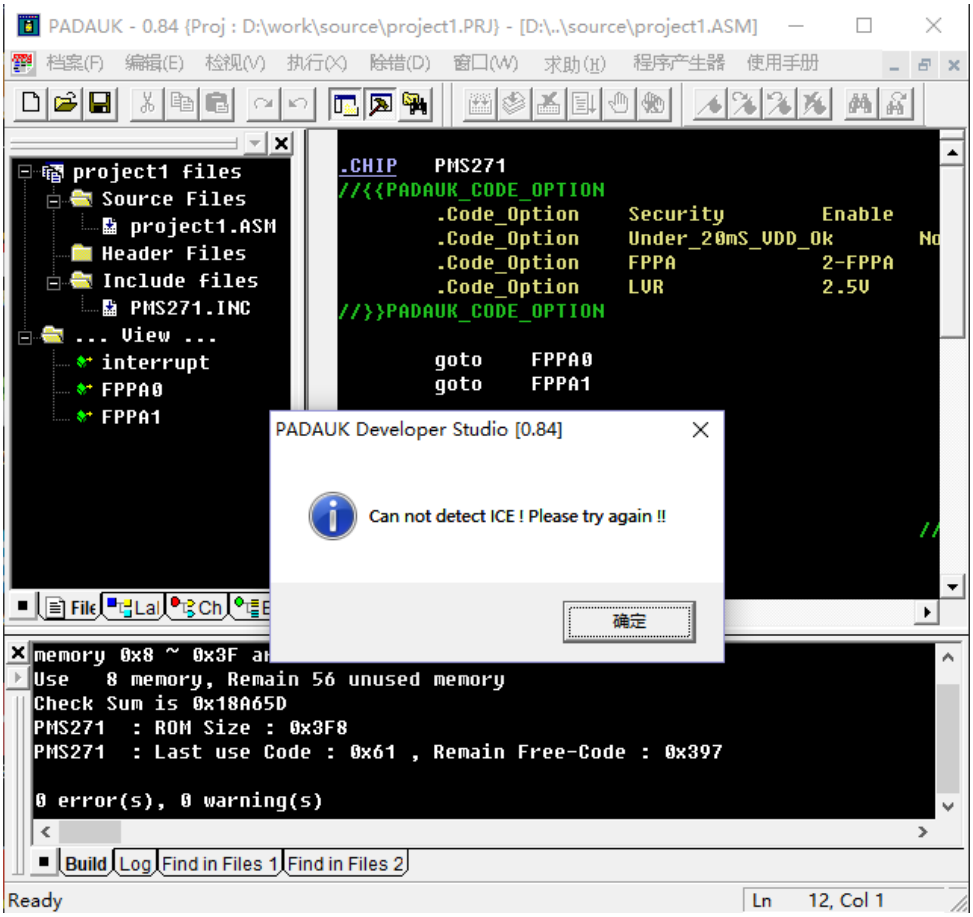


图 3-12. ICE 未连接

接上 ICE 后，可以按单步执行快捷键 **F11** 进入 Debug 模式，开始执行程序。视窗画面将如图 3-13 所示。

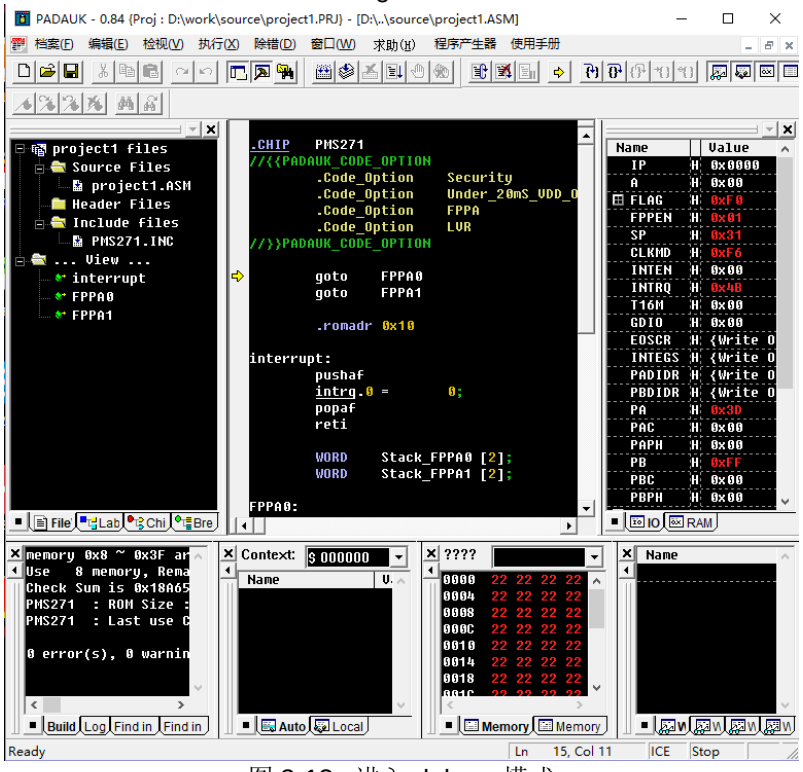


图 3-13. 进入 debug 模式

3.5. 建立新的 Mini-C 项目

若要建立 Min-C 类型的项目，步骤如下：首先，从选单上开新项目，如图 3-14 所示。

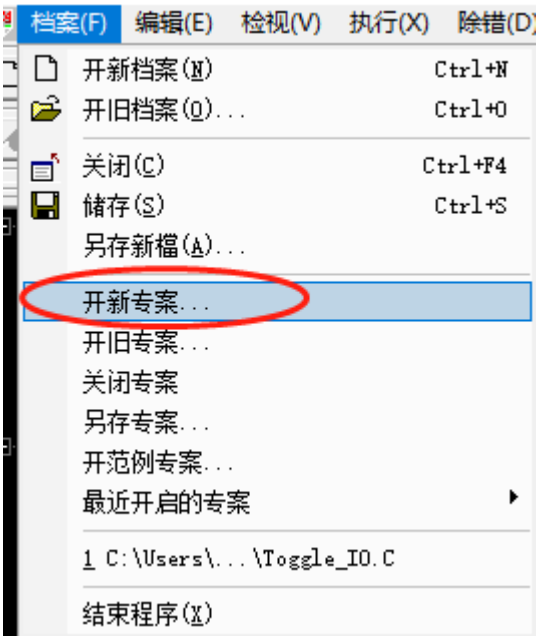


图 3-14. 开新项目选项

若是出现图 3-15 所示语言格式对话框，点选 【Mini-C】模式。

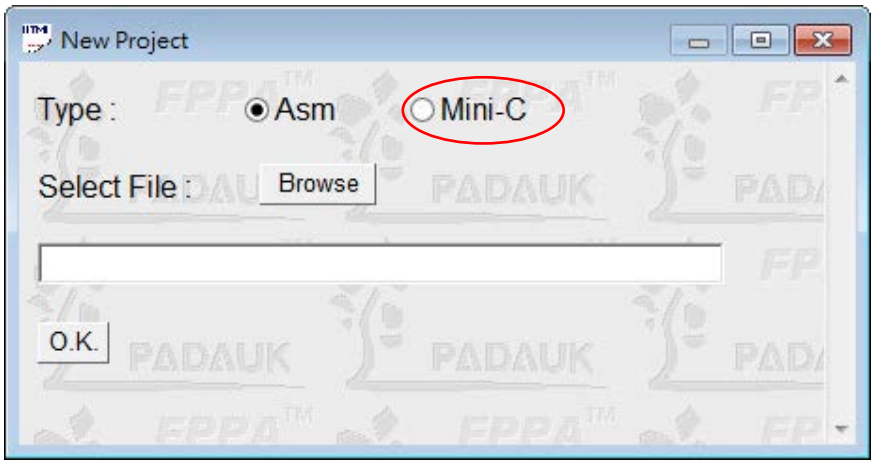


图 3-15. 语言格式选择视窗

选择 【Mini-C】 以后会出现图 3-16 对话框。用户输入项目名称 (预设为 Project1)，选择项目位置，选定 IC 及参数(可以用预设值)后按下【OK】即可。

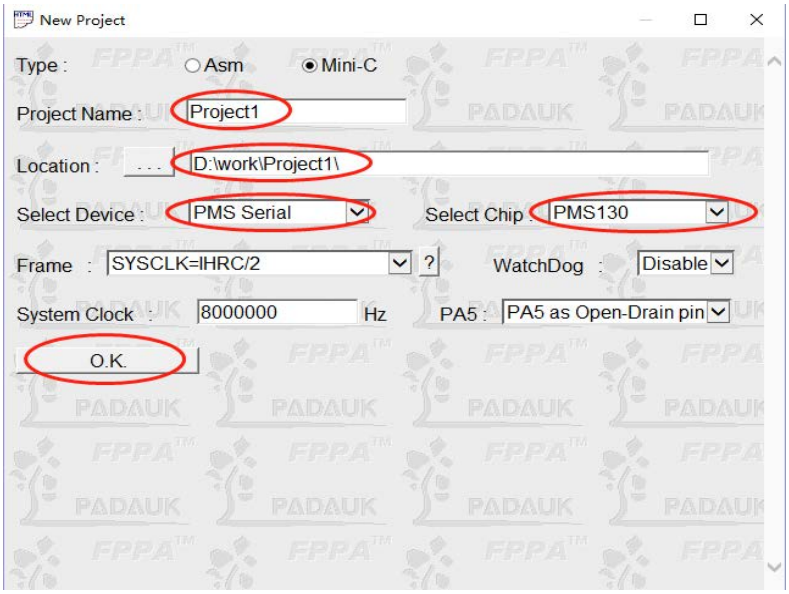


图 3-16. Mini-C 参数选择视窗

Mini-C 项目建立好之后，发展系统的视窗画面如图 3-17 所示。

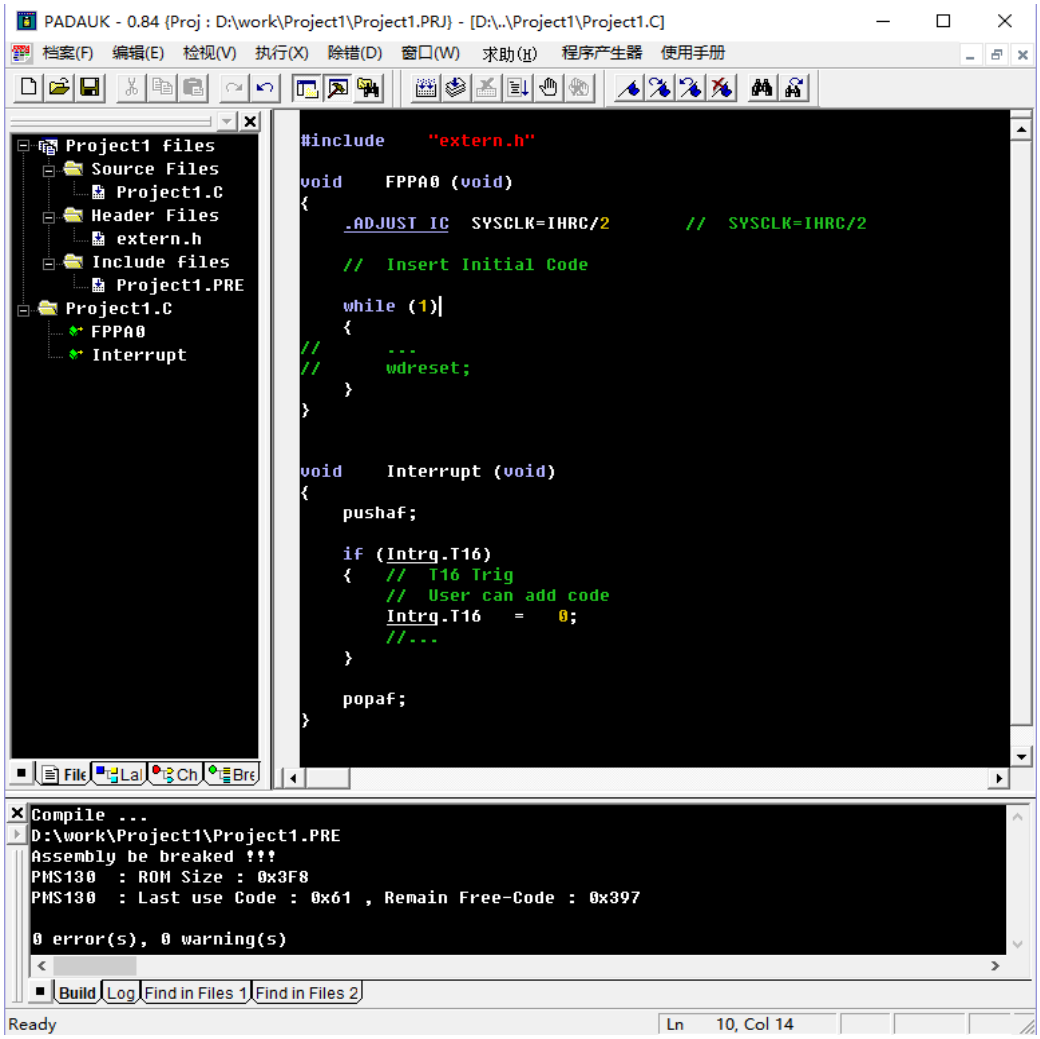


图 3-17. Mini-C 项目建立后的视窗画面

点击【建置(B)】或者按快捷键 **F11** (单步执行)打开程序选项 Code_Option，进行程序参数设置，如图 3-18 所示。

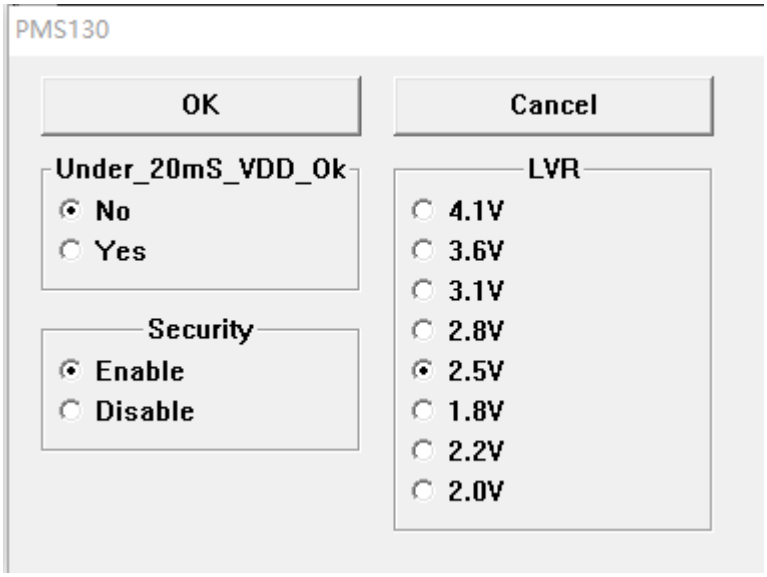


图 3-18. 程序选项视窗

LVR 的电压水平需要与系统频率（默认值为 8MHz）及 IC 工作电压相协调，这样才能使系统达到稳定的工作状态。若 LVR 选择的电压不合适，IDE 会给出 error 提示，如图 3-19：

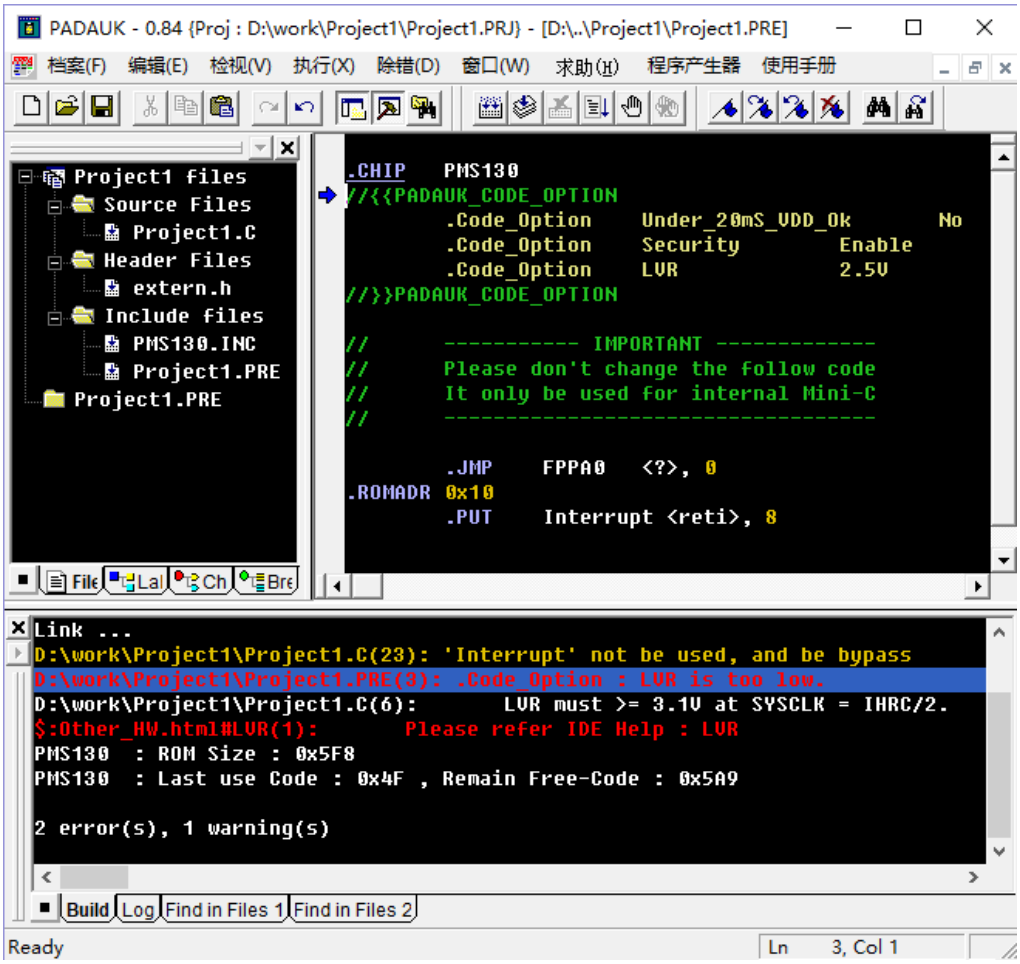


图 3-19. LVR 水平调节

若出现上述情况，使用者可以通过降低系统频率（默认为 8MHz）或者提高 LVR 水平来解决这项 error。

在 .C 文件中，将“.ADJUST_IC SYSCLK=IHRC/2”删除或是设成注解（如图 3-20），再点击【建置(B)】或【重建】按钮，就会出现 IHRC 校准、系统频率选择的页面，如图 3-21 所示：

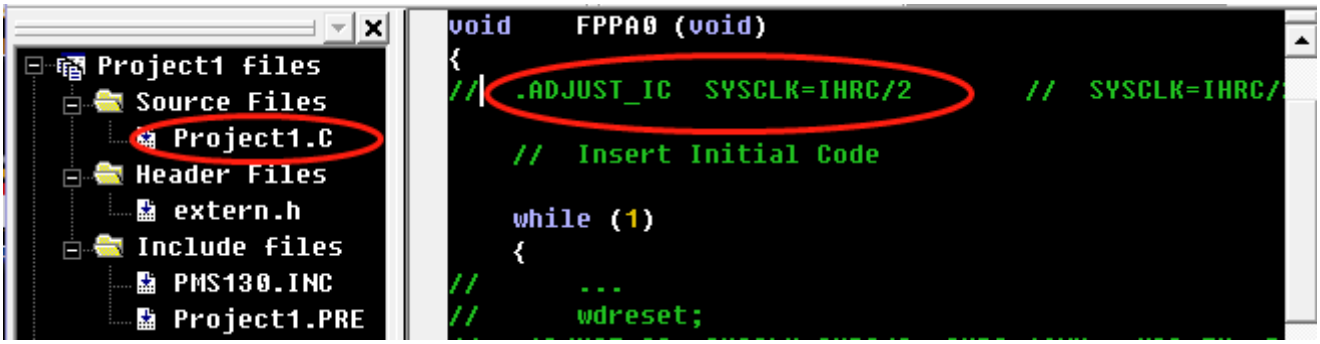


图 3-20. 系统频率校准语句

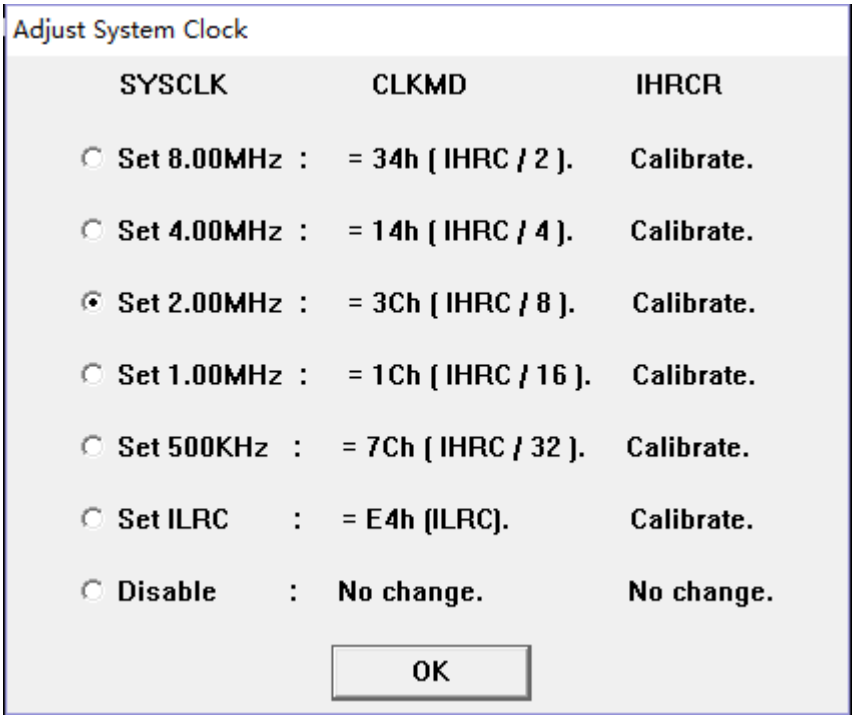


图 3-21. 系统频率选择、IHRC 校正选项

IDE 在建置部分 IC（如 PMS130）时会带有 band-gap 校准功能。在用户选定好系统频率以后，IDE 会自动弹出 band-gap 校准选项供用户选择，如图 3-22 所示：

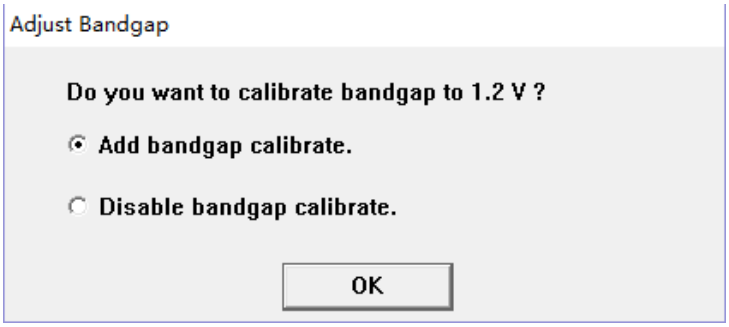


图 3-22. band-gap 校正选项

选择完毕后，程序中就可以看到 IC 的系统频率，工作电压以及 band-gap 是否校准功能，如图 3-23：

```
.ADJUST_IC  SYSCLK=IHRC/8, IHRC=16MHz, VDD=5V, Bandgap=On;
    //  WatchDog Disable, RAM 0 ~ 0xF temporary be used
//  You can add the follow code :
//      CLKMD.En_WatchDog    =    1;       // WatchDog Enable
```

图 3-23. band-gap 校正选项

其中工作电压 VDD 是指 ICE 仿真时的工作电压，用户可以自行调节其大小，结合系统频率和 LVR 的设置，使系统稳定工作。

注意：上图 3-19 中，指令“.CHIP xxxxxx”用于选择芯片型号，本例选择 PMS130，用户可以在此自行更改所需 IC 型号。

在设定完程序参数后，用户可以书写自己的程序功能，经【建置(B)】或【重建】以后即可进入侦错环境，开始运行程序，视窗画面如图 3-24 所示：

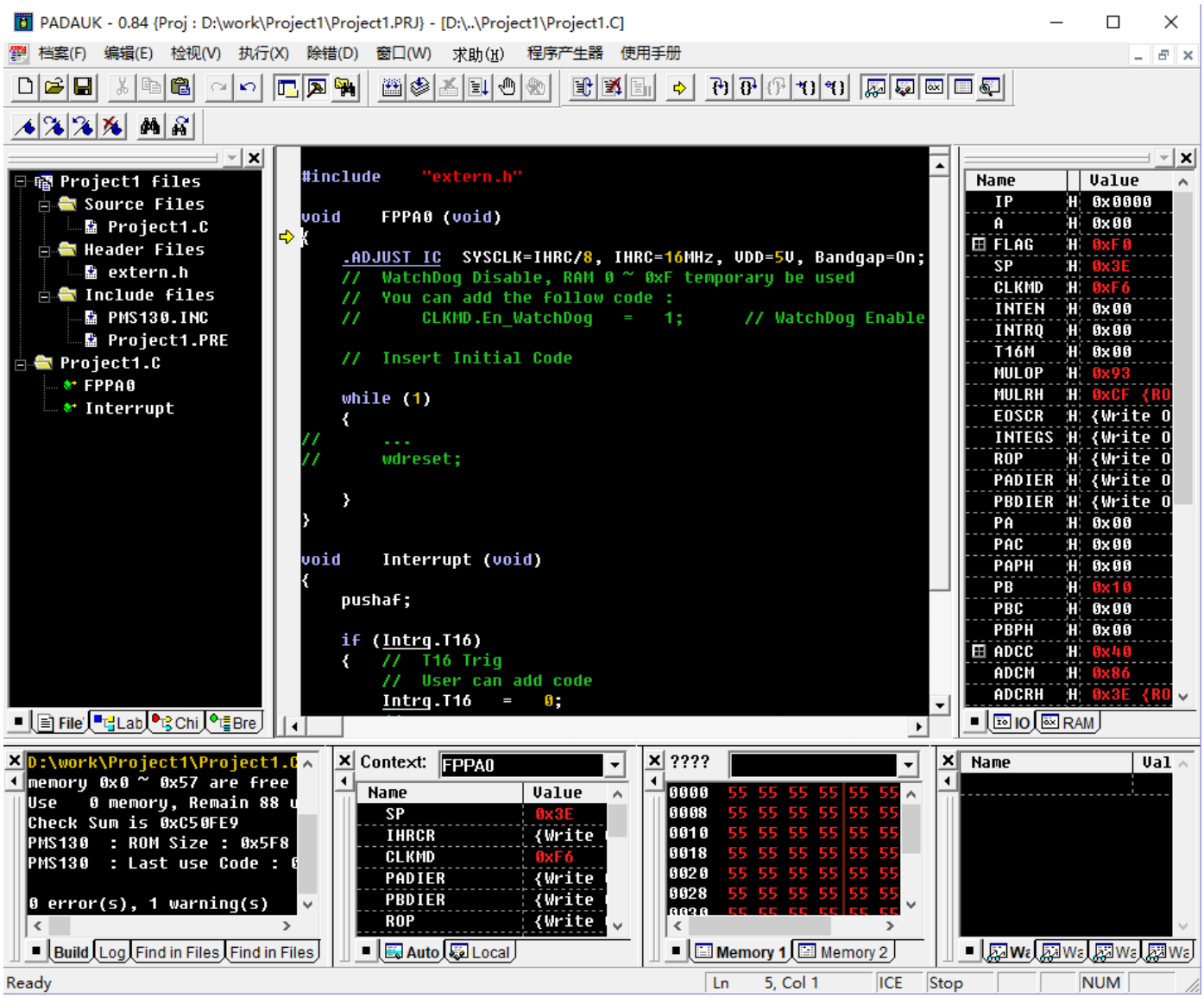


图 3-24. Mini-C 项目程序 debug 视窗画面

4. FPPA™ IDE 发展系统的环境、视窗介绍

4.1. Mini-C 的组译环境

Mini-C 程序执行的选项，如图 4-1 所示。要组译程序码，可按选单的 【建置(F7)】 项目。当然，如果使用者直接按 Debug 功能，如：单步执行(F11)， 跨步执行(F10)，继续执行(F5)，…，系统也会自动组译程序码，让操作更方便。

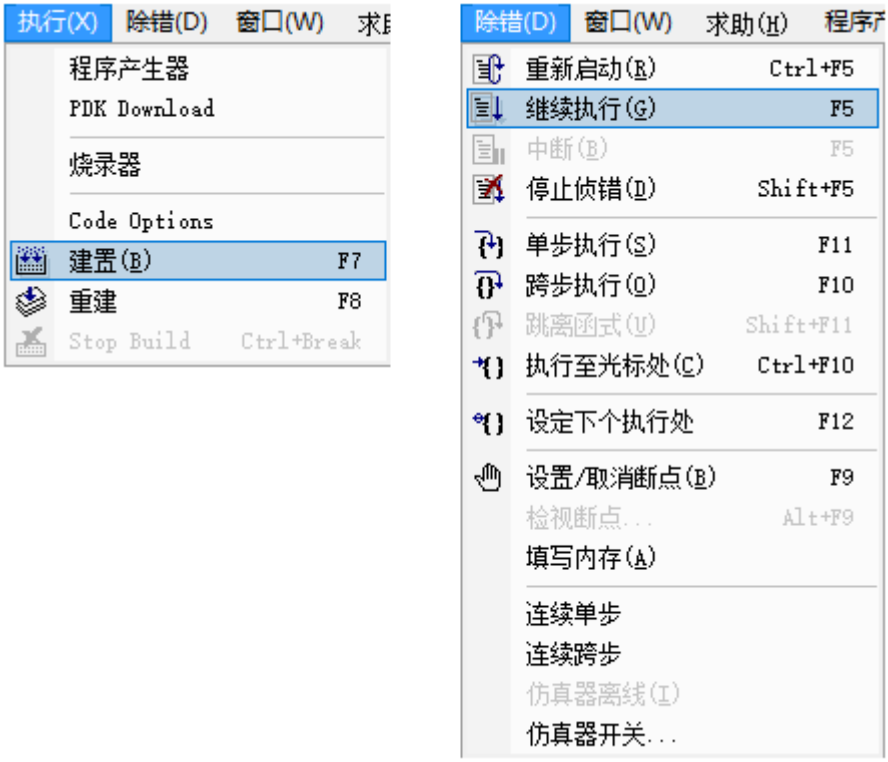


图 4-1. Mini-C 程序执行选项

- 【执行】选单中的【程序参数(Code Options)】选项：支持用户更改一些程序参数，如 Security、LVR 等，更改后这些参数将被自动贴在程序码中。
- 【执行】选单中的【重建(F8)】选项：所有的程序码重新组译。
- 【执行】选单中的【建置(F7)】选项：为了加快程序执行速度，只会针对程序更新的部份做组译，而组译过程中所建立的 OBJ 档，会存在项目目录下的 OBJ 子目录。当用户第一次组译后，OBJ 子目录就会被自动建立。
- 【执行】选单中的【烧录器】选项：可以作芯片的烧录。
- 【执行】选单中的【程序产生器】选项：可以产生用户需要的范例程序码。产生出来的程序码，将会被贴在剪贴板上，用户可用 “ Ctrl + V ”将其放到指定地方。

4.2. 发展工具的组译讯息视窗

发展工具在程序组译后，视窗下方可以看到如图 4-2 所示的 Mini-C 程序组译讯息视窗：

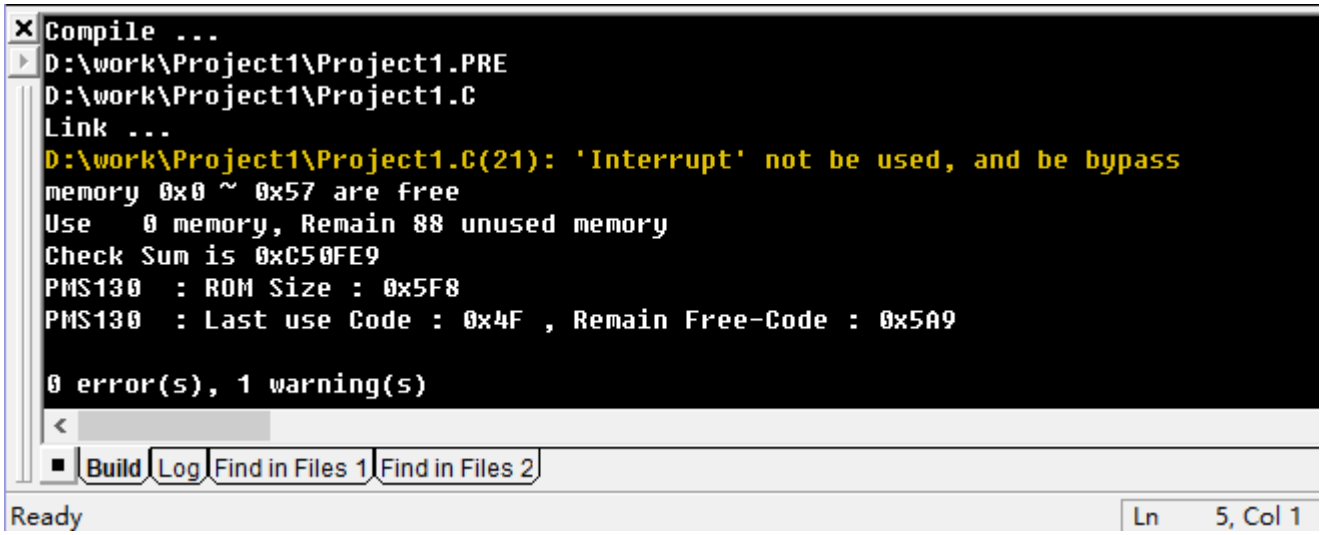


图 4-2. Mini-C 程序组译讯息视窗

- 只要视窗没有红色的 error 讯息，就表示组译成功。
- 黄色的 Warning 讯息，通常是发展工具侦测到未使用的变量或程序码，提醒使用者注意。
- 利用鼠标双击视窗内的 Error/Warning 讯息，可以指定到相应的程序位址。
- 按鼠标右键会出现一个功能清单，可以帮助用户快速到达前一个或后一个错误处，如图 4-3 所示。

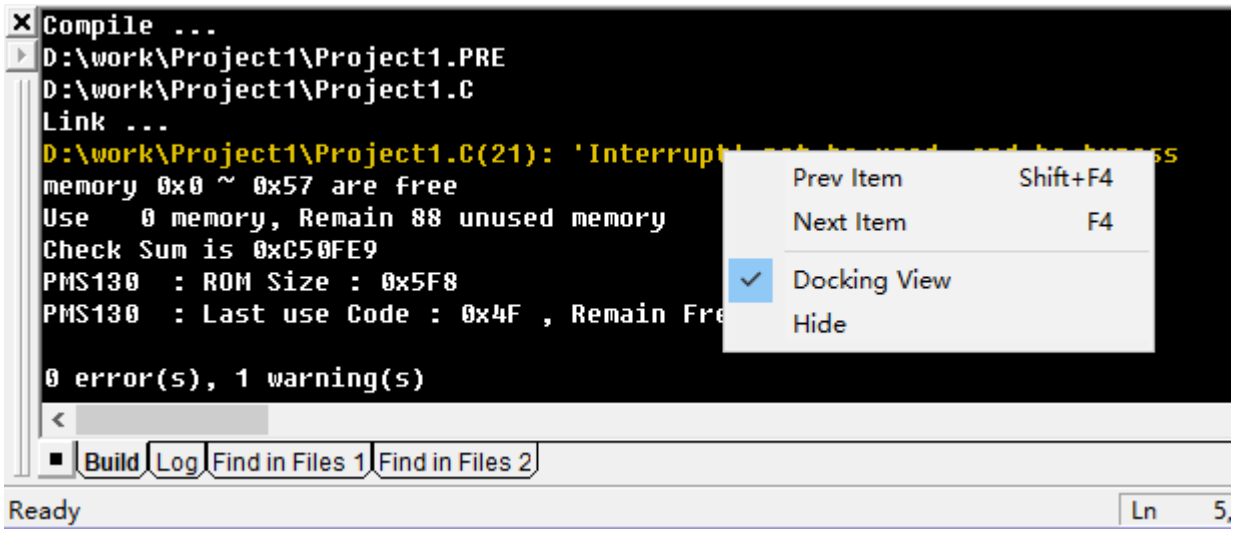


图 4-3. Mini-C 组译讯息快速除错方法

4.3. 发展工具的工作区管理以及侦错环境

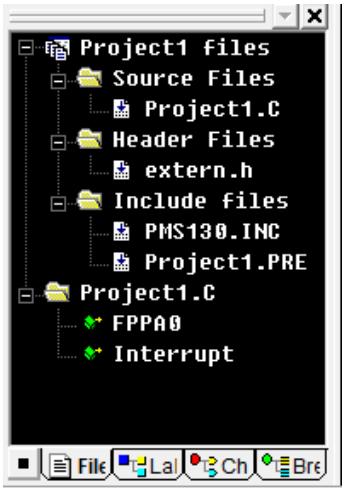


图 4-4. 工作区管理视窗

在发展工具左边工作区，可以看到如图 4-4 所示的内容。

在工作区内利用视窗切换，可以包含下列四种功能：

(A). FileView：用来显示目前项目由哪些档案所组成

(B). LabView：用来显示目前开启的档案定义那些 Label

(C). ChipView：用来显示目前每个 FPPA 的个别信息

(D). BreakView：用来控制 ICE 的各种 Break 开关

用户可以通过鼠标的双击来开启任何想看的物件，或按鼠标右键后，显现出该项的功能清单。

范例： 双击+档案 -> 开启档案

双击+Label -> 游标移到 Label 定义处

4.3.1. FileView 使用说明

(1) 变更 IC 设定前置档案

在 Source Files 或 Project File 项目上按右键，接着在开启的功能表中，选择【Change .PRE to Project】，将目前的设定前置档案做变更，如图 4-5 所示。

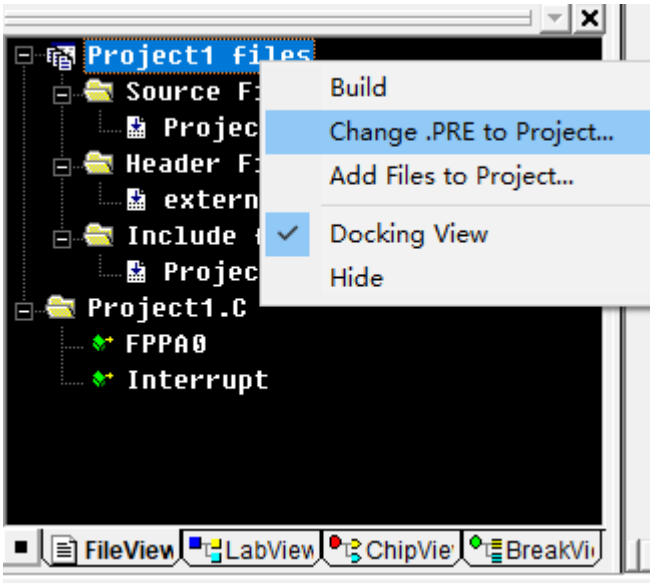


图 4-5. 变更 IC 设定前置档案

(2) 连结档案

在 Source Files 或 Project File 项目上按右键，接着在开启的功能表中，选择【Add Files to Project】，将想要连结的档案，连结到项目来，如图图 4-6 所示。

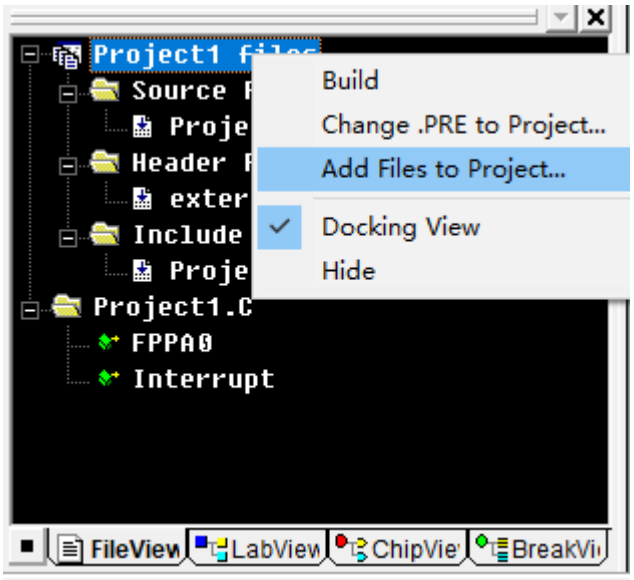


图 4-6. 档案连结

(3) 移除档案连结

在 Source Files 底下的“.C”文件上按右键，接着在开启的功能表中，选择【Delete】，将不要连结的档案，从项目中移除，如图图 4-7 所示。

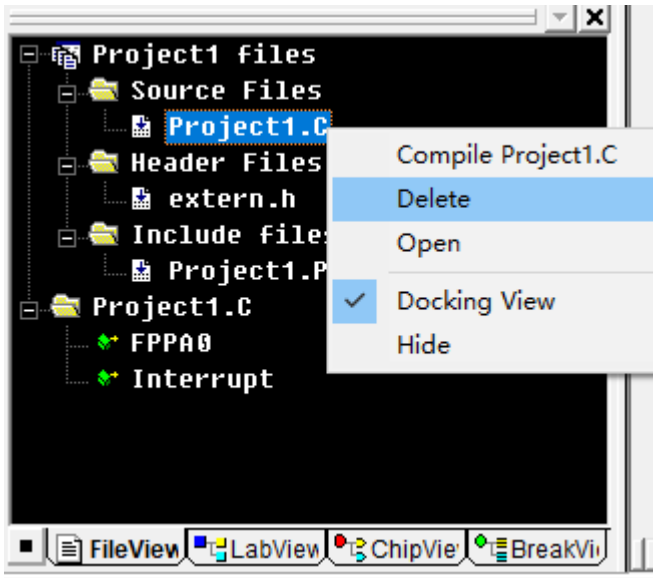


图 4-7. 移除档案连结

(4) Head File 的参考档案

在 Head Files 项目上按右键，接着在开启的功能表中，选择 【Add Files to Folder】，将想要参考的档案，连结到项目来，不过，此功能纯属快速参考文件档，不会影响组译结果。

(5) 在 Asm 模式的项目，只允许一个主档案在 Source Files 项目中；而在 Mini-C 模式的项目中，则可单独组译各个连结档案（“Compiler xxx.C”）。

4.3.2. LabView 使用说明

(1) 此视窗在组译尚未成功时，并无信息，如图 4-8 所示：

图 4-8. LabView 组译尚未成功的讯息

(2) 此视窗在组译成功后，显示信息如图 4-9 所示：

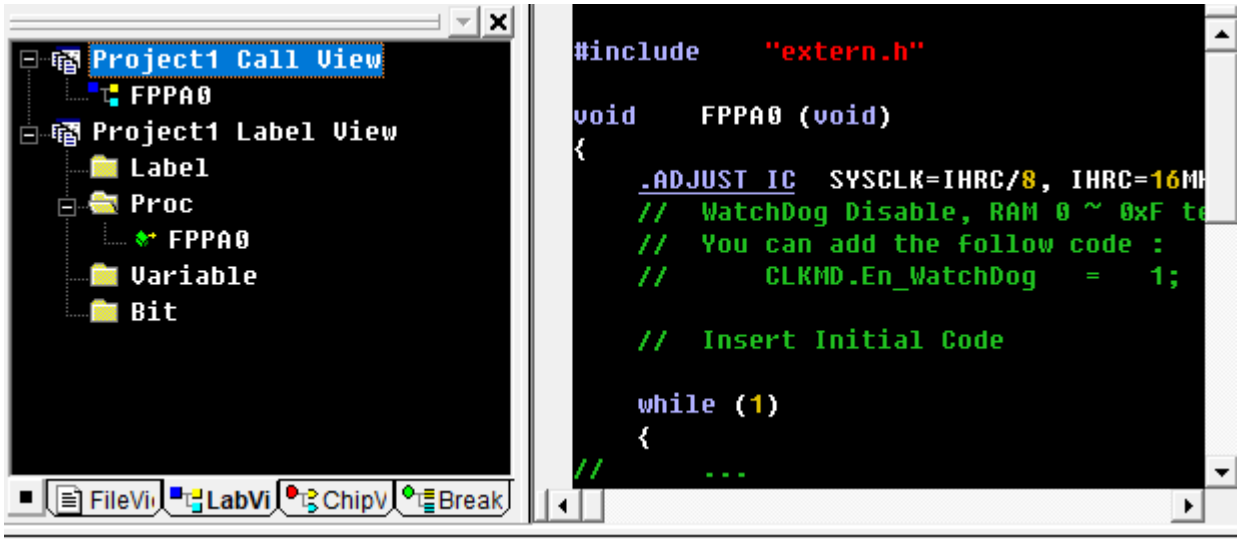


图 4-9. LabView 组译成功的讯息

LabelView 项目中，存放已知符号的列表，分四大类: Label / Proc / Variable / Bit，描述如下：

- Label： 在 ASM 项目中，语法为 “xxx : “ 皆是。
- Proc： 在 Mini-C 项目中，语法为 “void xxx (void) {} “ 皆是。
- Variable： 语法为 “BYTE /WORD/EWORD/DWORD xxx“ 皆是。
- Bit： 语法为 “BIT xxx [: yyy.n]“ 皆是。

在图 4-9 例子中，连续双击某一项目即可快速切换至目标位置，方便使用者搜寻。

4.3.3. 发展工具的侦错环境

使用【除错】选单的重新启动(Ctrl+F5)，继续执行(F5)，单步执行(F11)，跨步执行(F10)，执行至游标处(Ctrl+F10)，设定下一个执行处(F12)，连续单步以及连续跨步等操作，系统将进入侦错环境，功能选择如图 4-10 所示。

图 4-10. 侦错功能选择

进入侦错环境后，看到的视窗如图 4-11 所示。

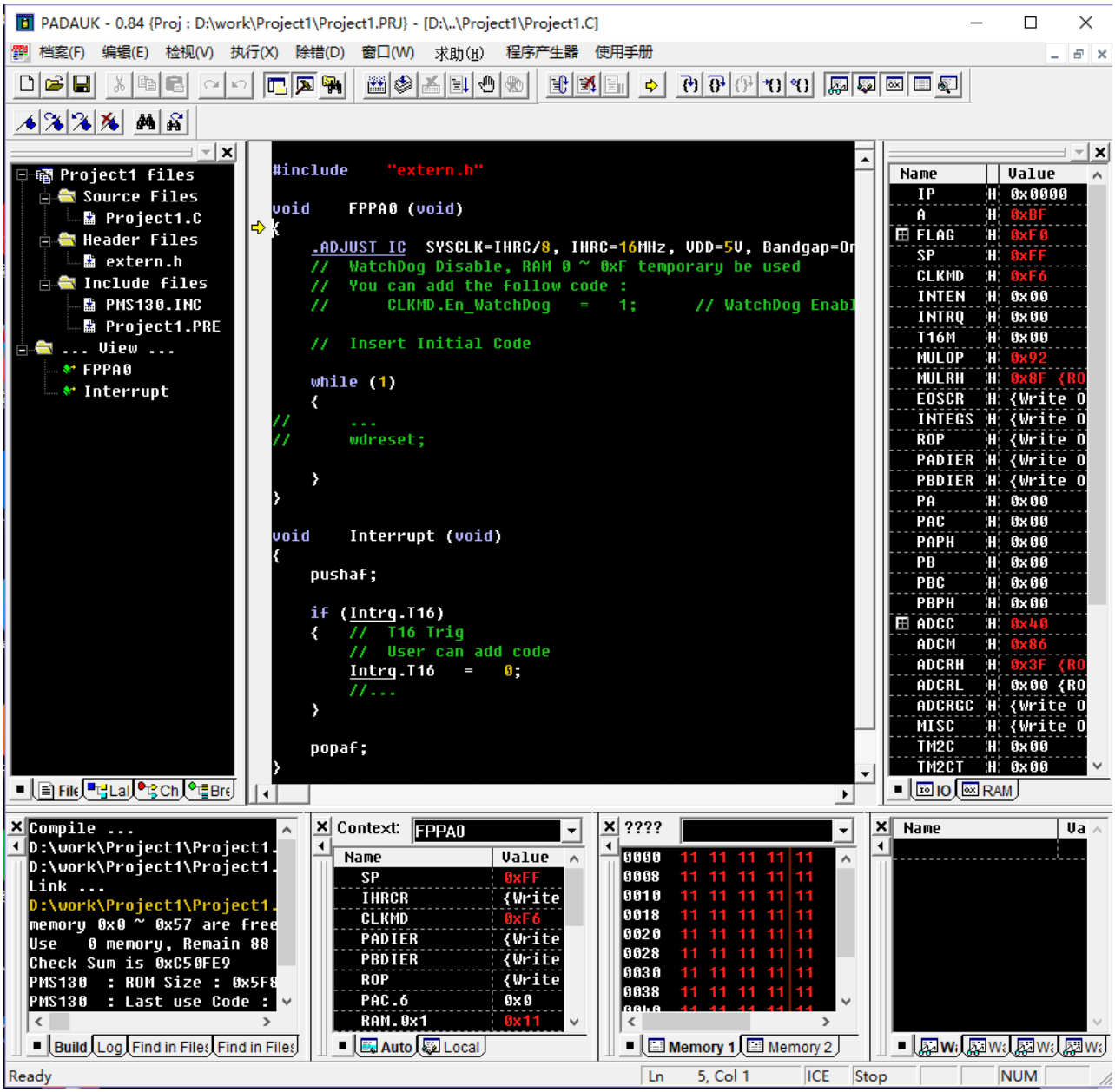


图 4-11. 侦错的视窗画面

(1) **右上方寄存器视窗：** 记录着 IC 各个相关寄存器的信息。

(2) **右下方 Watch 视窗：** 显示使用者输入的变量。

(3) **下方中右为 Memory 视窗：** 以 Hex 格式显示出 RAM，LCD 的资料内容。

(4) **下方中左 Variable 视窗：** 记录着三种信息：

1) Context：目前程序的堆栈信息

2) Auto：显示下次执行可能被使用的 Variable

3) Local：一个程序的区域变量

(5) 左上方 Workspace 的 Chip View 视窗：

Chip View 视窗记录着 FPPA 的 A、FLAG、SP 等。如图 4-12，用户可以在这个视窗中快速查看和修改 FPPA 的 A、FLAG、SP 的数值。

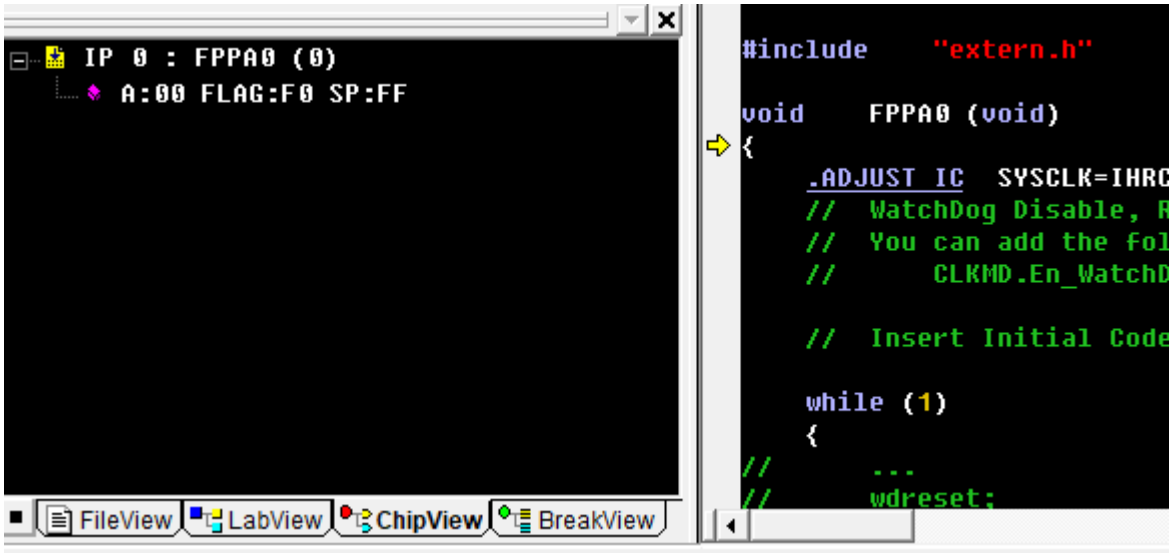


图 4-12. ChipView 的视窗讯息

对于多核心系列的 IC，Chip View 视窗可以看到每个 FPPA 的信息，并且有 Mult. Traps 与 Single Trap 两种模式可选，IDE 系统默认是在 Single Trap 模式。如图 4-13 所示。

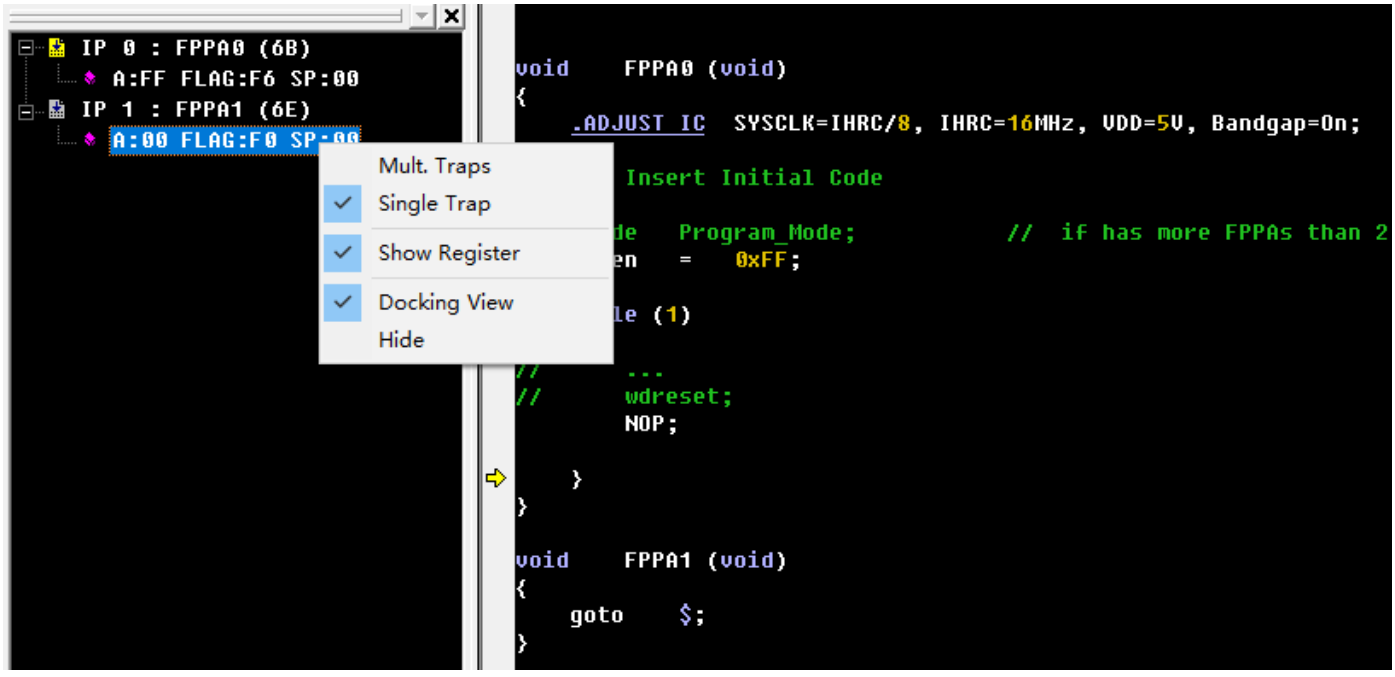


图 4-13.多核 IC 的 ChipView 视窗讯息

在 Single Trap 模式下，一次只能监看一个 FPPA，如果想切换所监看的 FPPA，只要用鼠标双击想监看的 FPPA 就可以了，如图 4-14 所示的绿框文字。

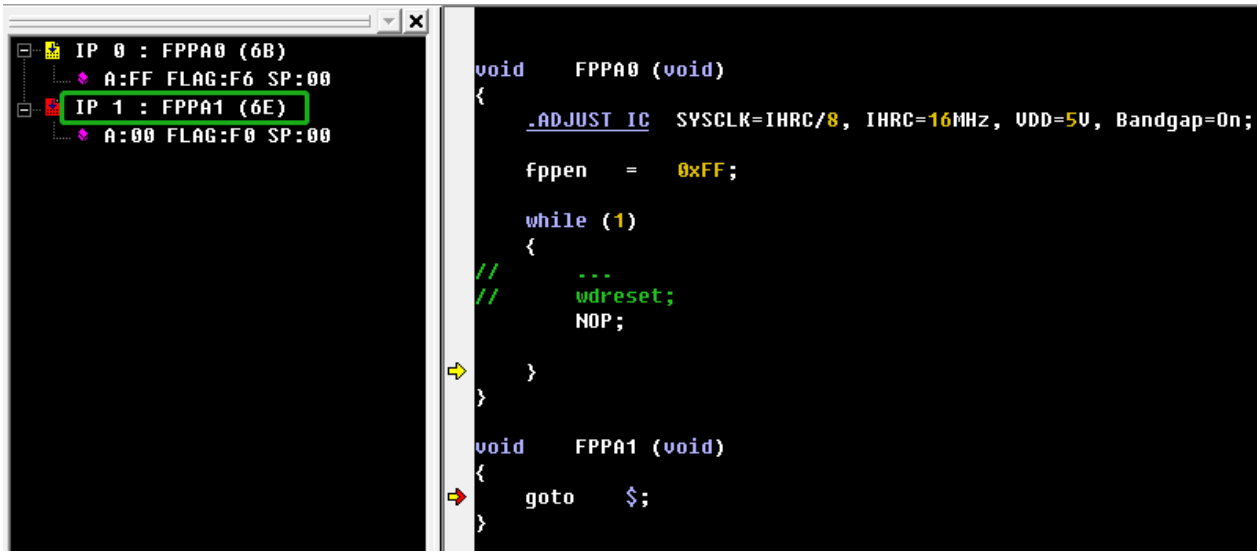


图 4-14. Single Trap 切换 FPPA 的视窗讯息

图 4-14 显示的意义为：IP 0 是目前程序所停的 FPPA (➪显示处)， IP 1 则是程序下次想停的 FPPA (➪显示处)，当再执行一次单步执行(F11)，所监视的 FPPA 就已经改为 IP 1 了如图 4-15 所示：

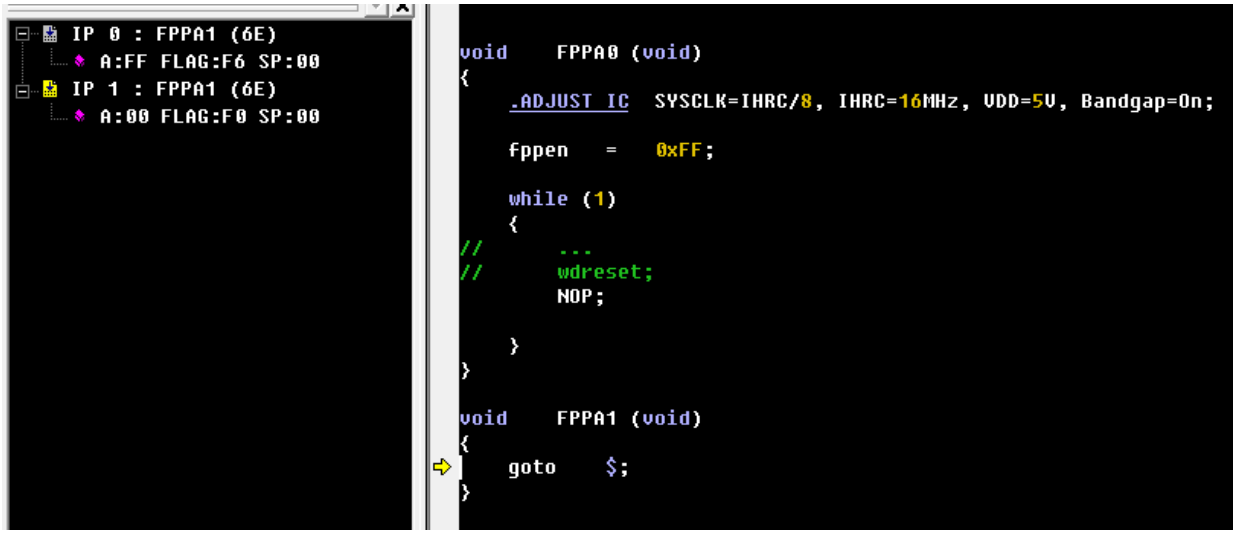


图 4-15. 切换 FPPA 后的视窗讯息

若想同时监看多个 FPPA，可以按鼠标右键，选择 Mult Traps 方式。Mult Traps 视窗如图 4-16 所示：

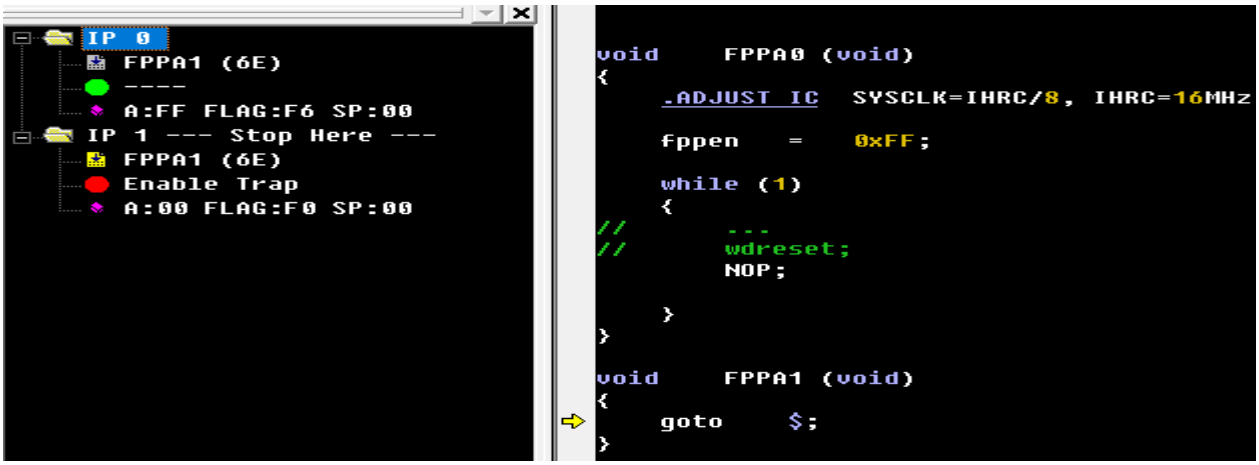


图 4-16. Mult. Traps 视窗

在 Mult. Traps 模式下，使用者可以选择想要监看的 FPPA，只要 Enable Trap (可用鼠标切换状态) 就可以同时监看各个 FPPA 的内容，当然也可以修改各个 FPPA 相关的内容，包括 A、FLAG、SP，如图 4-17 所示。

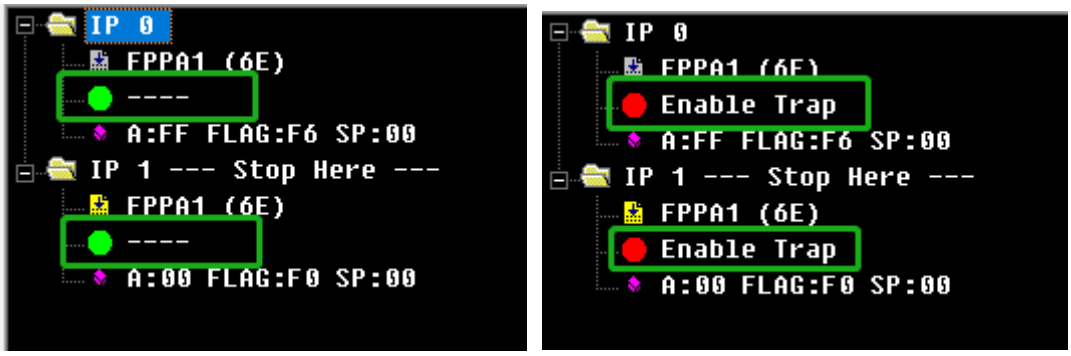


图 4-17. ChipView 的视窗讯息

(6) 左上方 Workspace 的 Break View 视窗：

1)Break View 视窗的功能，是用来控制 ICE 各种 Break 的开关，如图 4-18 所示。

图 4-18. Break View 功能视窗

2)Reset Break：侦测 Reset 是否发生。

4.3.4. 发展工具的反组译码

在纯粹汇编语言的环境上，单步，或许可以解释成执行一条汇编语言指令；但在 Source Level Debug 的环境上，“单步”意味着执行完一行指令，这一行指令可能包含数条汇编语言指令，如巨集，或 C 语言。

如果用户需要查看汇编语言的单步执行，请在进入侦错环境后，选择如图 4-19 所示的“反组译码”，在反组译视窗中的单步执行 (F11)，就是以一条汇编语言指令为单位的单步执行。

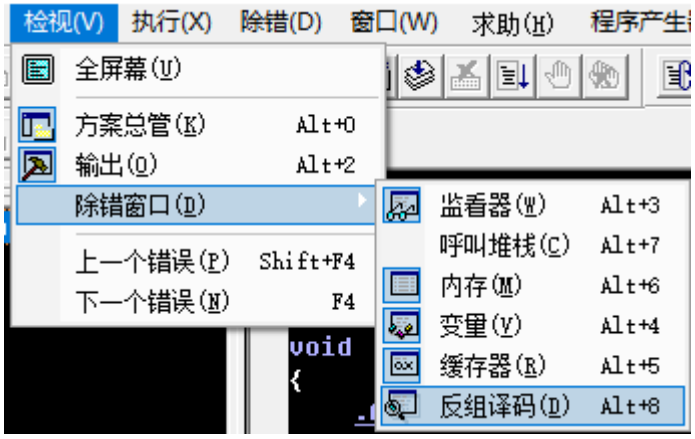


图 4-19. 反组译码选择视窗

在反组译码视窗中，按下右键后的弹跳视窗，用户可选择将 LIST 档储存下来。

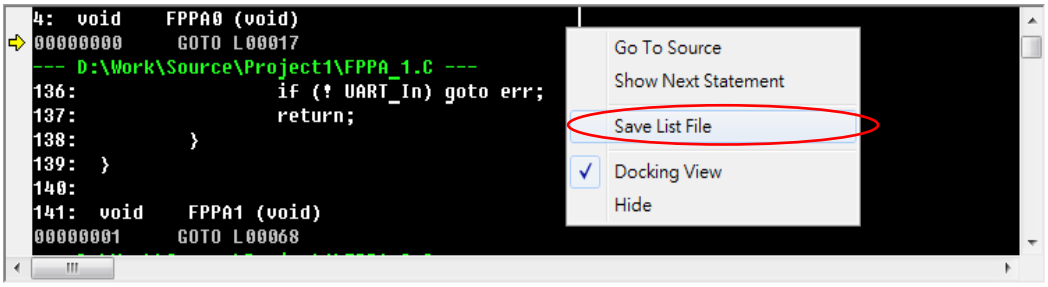


图 4-20. 反组译码视窗

4.4. 实际范例应用

假设一个错误范例如图 4-21 所示。

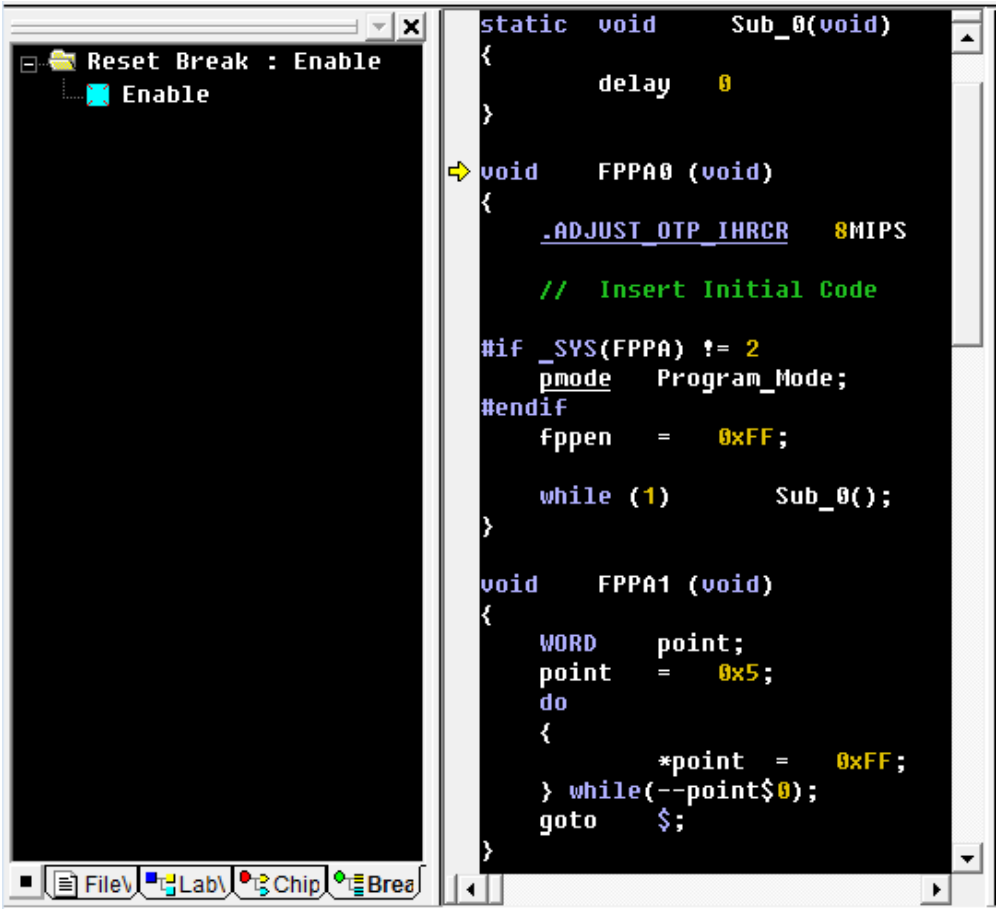


图 4-21. 错误范例视窗

FPPA0 在校正 IHRC 后，即进入一个无穷循环，并呼叫 Sub_0；FPPA1 则负责清除 RAM。在 ICE 执行 Free Run 后，因为“.ADJUST_OTP_IHRCR”（现在的用法“.ADJUST_IC SYSCLK=IHRC/x,IHRC=16MHz .....”）在校正 IHRC 时，会用到查表指令，导致 ROM Break 被触发，使程序停下来，这是正常现象，用户可以不理会它。

再次执行 Free Run，ICE 出现如下讯息，程序从 0x18 跳到不合法的 0x1F3B。如图 4-22 所示。

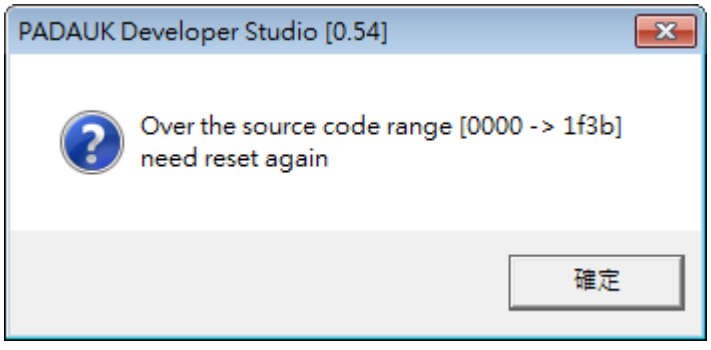


图 4-22. ICE 出现不合法资料视窗

查看 Disassembly 视窗，原来位址 0x18 是 FPPA0 在作 RET 指令，检查 SP 值，是在 0x00，再检查 Memory 视窗，SP 还在 FPPA0 的堆栈范围内(系统命名为?_Stk_0)，没有问题，但 RAM 0/1 却被破坏成 0x3B, 0x1F 了，难怪执行 RET 后，IP 变成 0x1F3B (以此例 ICE 的 IP 有效范围到 0x1FFF，所以 0xFF3B --> 0x1F3B) 。

最可能的情况，就是别的 FPPA 的 Point，不小心越界修改到了 RAM 0/1 (如 IDXM point, A，*point = xx 等语法皆是)。此时检查 FPPA1，发现果然是 FPPA1 的 Point 越界了，如图 4-23 所示。虽然在实际的范例上，可能是更复杂的情形，但大致上都是用这种方法去检查。

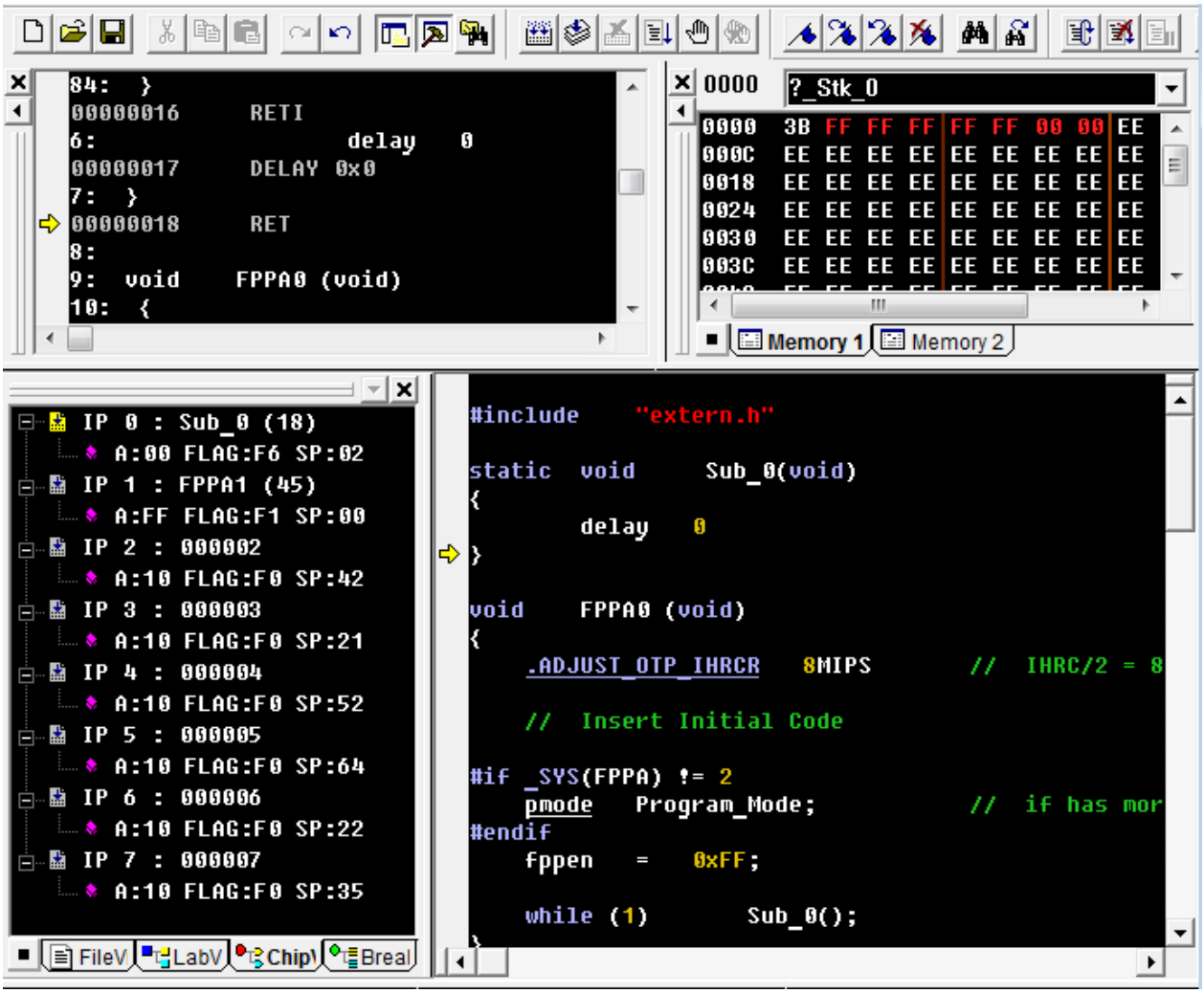


图 4-23. 检查 FPPA1 的视窗