

1 概述

IDE 环境需要安装如下：

- 开发软件
- 开发包

开发软件为 MDK-ARM，开发包则是 PT 基于开发软件，并针对各个 MCU 及其板级，制作的快速开发工具包，搭配参考手册，辅助用户快速熟悉 MCU 的各项特性和功能。下面就针对 IDE 环境的安装及配置进行详细说明。

2 开发软件安装

MDK-ARM 软件为基于 Cortex-M、Cortex-R4、ARM7、ARM9 处理器设备提供了一个完整的开发环境。MDK-ARM 专为微控制器应用而设计，而且功能强大，能够满足大多数苛刻的嵌入式应用。

MDK-ARM 也称 KEIL MDK-ARM、KEIL ARM、KEIL MDK 等，系为同一产品。

目前，我司提供的开发包均基于 MDK-ARM，使用 MDK-ARM，可便捷的开发我司旗下所有 MCU 产品的片内资源。

注意：

1. 安装路径不能带中文，必须是英文路径。
2. 安装目录不能与 C51 的 KEIL 或者 KEIL4 冲突，三者目录必须分开

2.1 获取 KEIL5

Keil 的安装包可到 Keil 的官网下载，地址：<https://www.keil.com/download/product/>。开发包支持 MDK4.7 及以上版本，推荐使用 MDK5.15，后续如有新版本也可使用更高版本。

图 1 MDK-ARM 下载路径

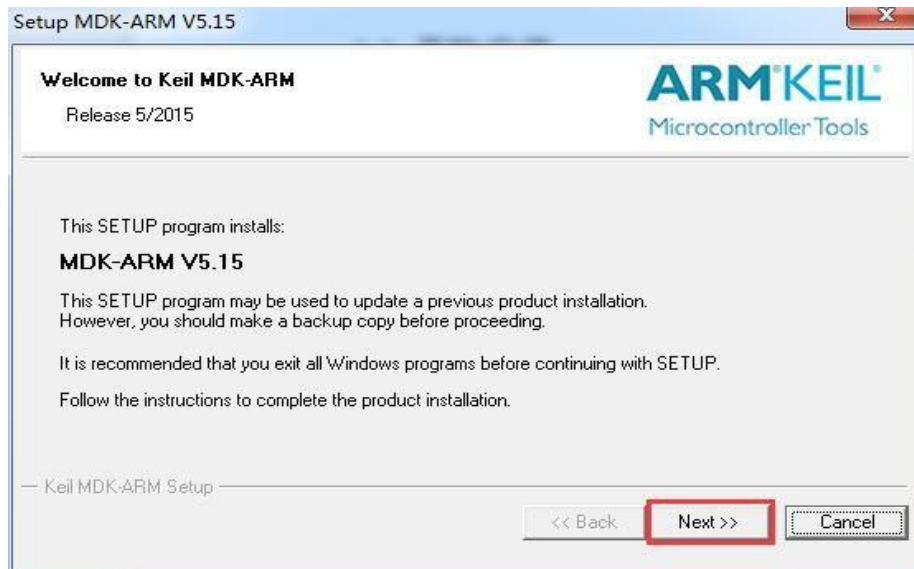


2.2 安装步骤

2.2.1 开始安装

下载完毕 MDK-ARM 安装包后，双击安装包开始安装，点击“Next”

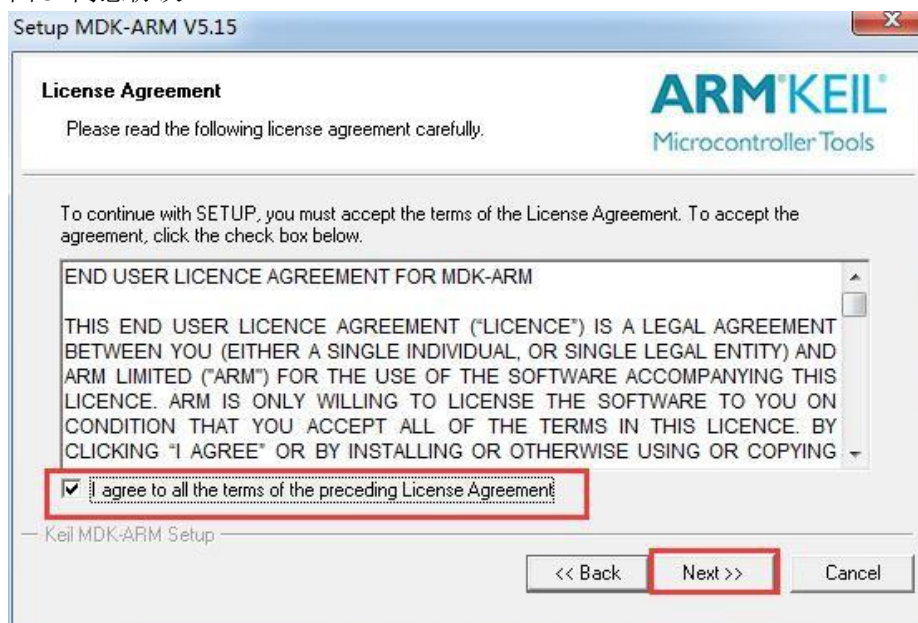
图 2 安装 MDK-ARM



2.2.2 阅读并同意协议

阅读并了解 MDK-ARM 协议，同意协议要求，勾选“I agree……”后，点击“Next”以执行下一步安装。

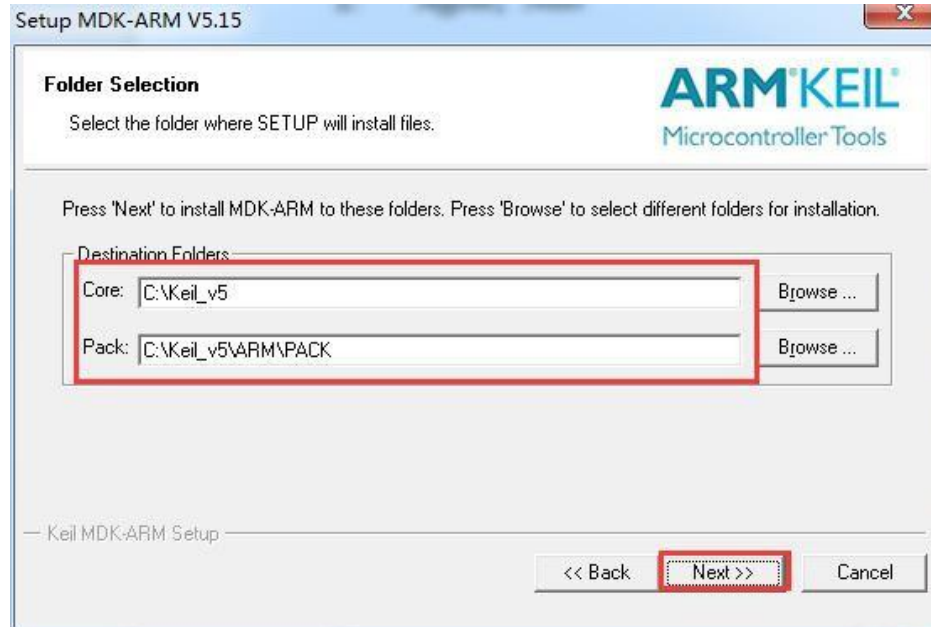
图 3 同意协议



2.2.3 安装路径

选择安装路径，路径不能带中文，并点击“Next”。

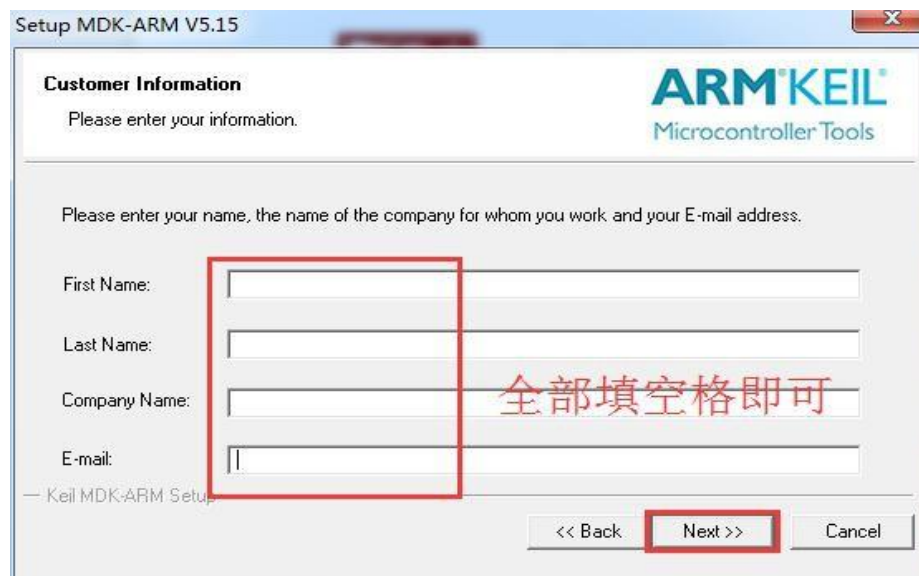
图 4 安装路径选择



2.2.4 用户信息

填写用户信息，如不想填写，则全部填充格（键盘的 space 键）即可，并点击“Next”。

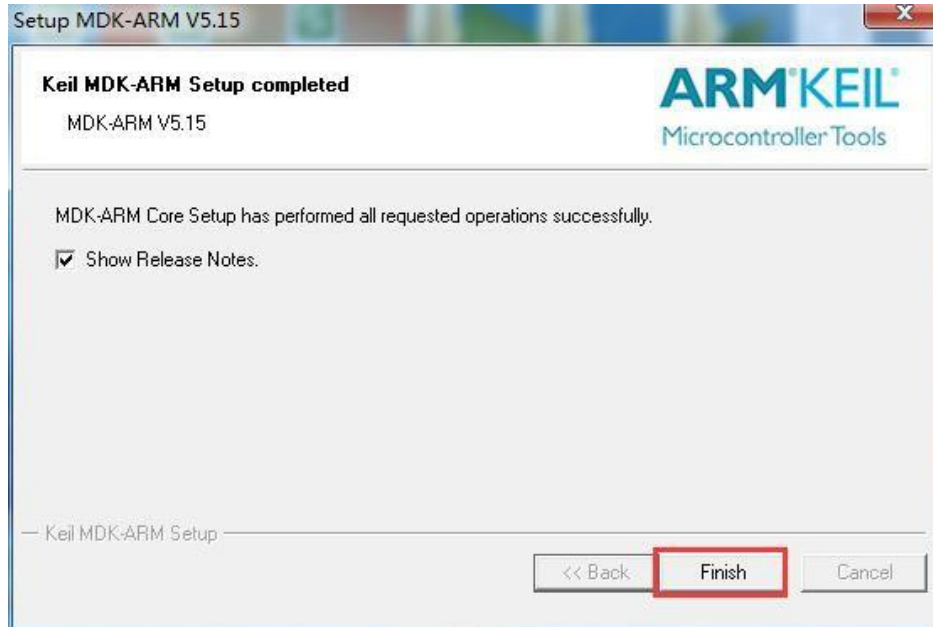
图 5 用户信息填写



2.2.5 完成安装

点击 Finish，即可完成安装。

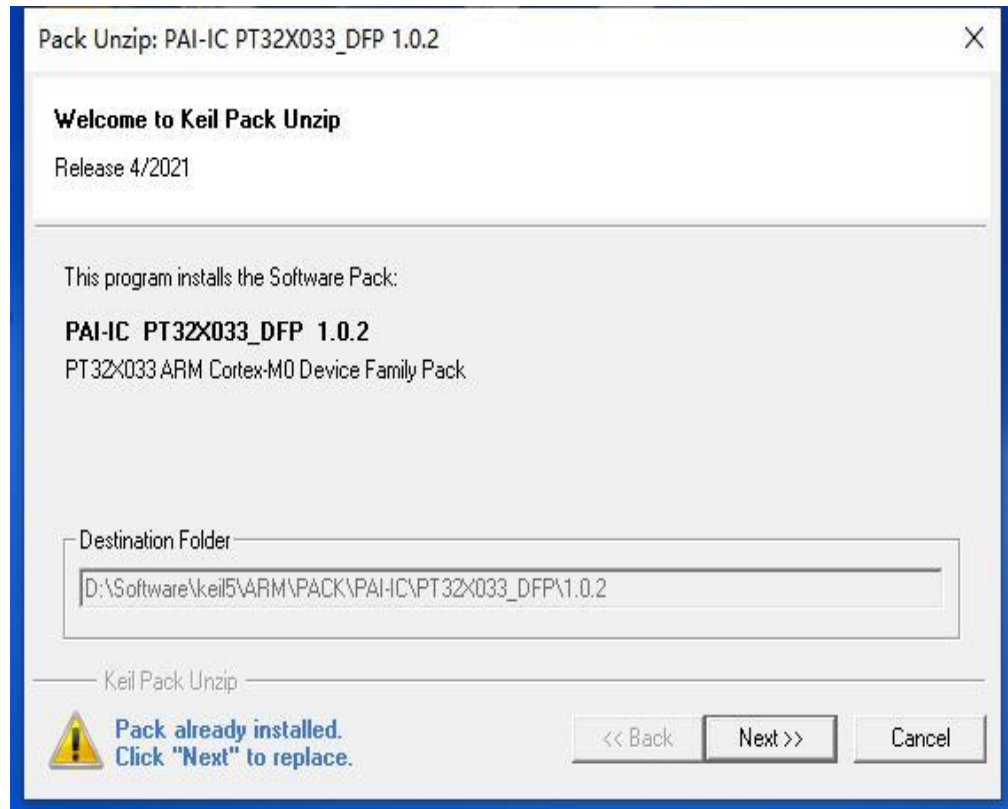
图 6 完成安装



3 安装开发支持包

PT 对旗下的所有的 MCU 在用户端的开发均提供了完整的支持，仅需安装对应的开发支持包，即可在参考手册的帮助下，配合开发包，快速的熟悉 MCU 各项特性和功能。在顺利执行完“开发软件安装”的步骤之后，我们双击从 PT 官网下载的“.pack”格式的支持包，MDK-ARM 会自动识别，并弹出如下图的“Pack Unzip”窗口(此处以“PAI-IC.PT32x_DFP.1.0.2.pack”支持包为例)：

图 7 安装开发支持包



开发支持包会自动设置安装路径，不需要人为改动。

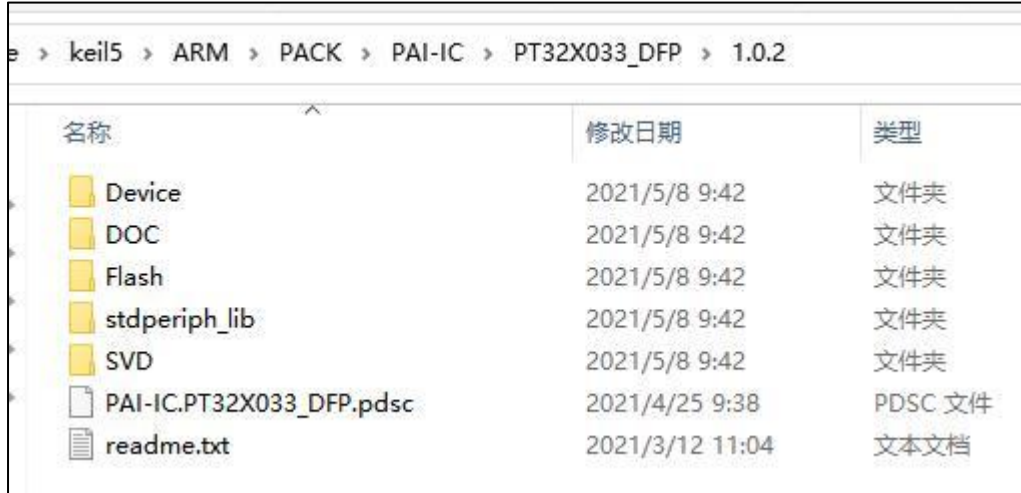
这个路径依赖于 MDK-ARM 的安装路径，比如上图中，笔者将 MDK-ARM 安装在 D 盘的“Software\keil5”文件夹下。

接着点击“Next”，安装完毕后，直接点击 Finish 即可。

4 开发支持包的结构

安装完开发支持包之后，按照安装路径打开支持包目录，了解 PT 为用户提供的支持内容，如下图(此处以“PAI-IC.PT32x_DFP.1.0.2.pack”支持包为例)：

图 8 开发支持包的文件结构



名称	修改日期	类型
Device	2021/5/8 9:42	文件夹
DOC	2021/5/8 9:42	文件夹
Flash	2021/5/8 9:42	文件夹
stdperiph_lib	2021/5/8 9:42	文件夹
SVD	2021/5/8 9:42	文件夹
PAI-IC.PT32X033_DFP.pdsc	2021/4/25 9:38	PDSC 文件
readme.txt	2021/3/12 11:04	文本文档

如图 8，下面我们将详细列出用户相关文件夹/文件的功能：

- Device 文件夹 -包含 MCU 的头文件
- DOC 文件夹 -包含参考手册、数据手册、板级支持套件等相关信息
- Flash 文件夹 -包含 MCU 的下载算法，keil 自动调用
- stdperiph_lib 文件夹 -包含库函数、基于库函数的例程
- SVD 文件夹 -包含仿真时的 MCU 系统资源观测文件，keil 自动调用
- readme.txt 文件 -**用户首先应该阅读的信息！**

5 开发环境连接板级

PT 为所有的 Demo 板集成了“CMSIS-DAP”仿真器，仿真器通过“SWD”与 MCU 连接，用户也可以使用“J-LINK”或“U-LINK”等支持“SWD”协议的仿真器与 MCU 连接。

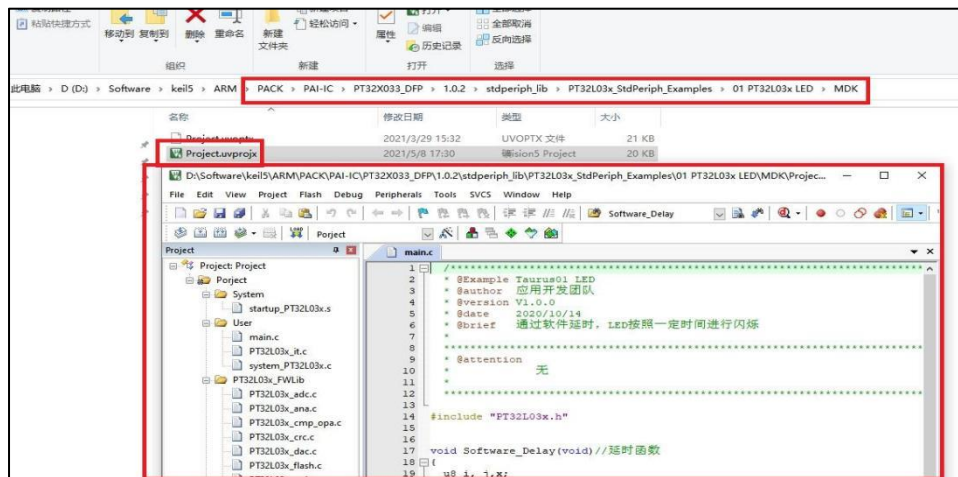
注意：

1. 下述支持包以“PAI-IC.PT32x_DFP.1.0.2.pack”为例。
2. 下述板级以“PT32L033_Demo”板为例。

5.1 打开例程

打开 stdperiph_lib 文件夹，并打开例程目录下任一例程，如下图所示：

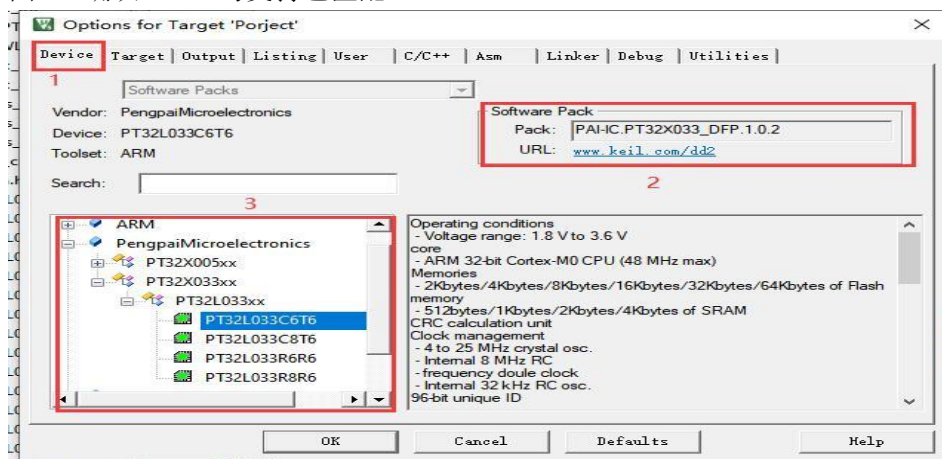
图 9 打开例程



5.2 确认 MCU 与支持包匹配

在打开的例程中点击“Options for target”，并选则 Device 框，确认 mcu 与当前的例程匹配，如下图所示：

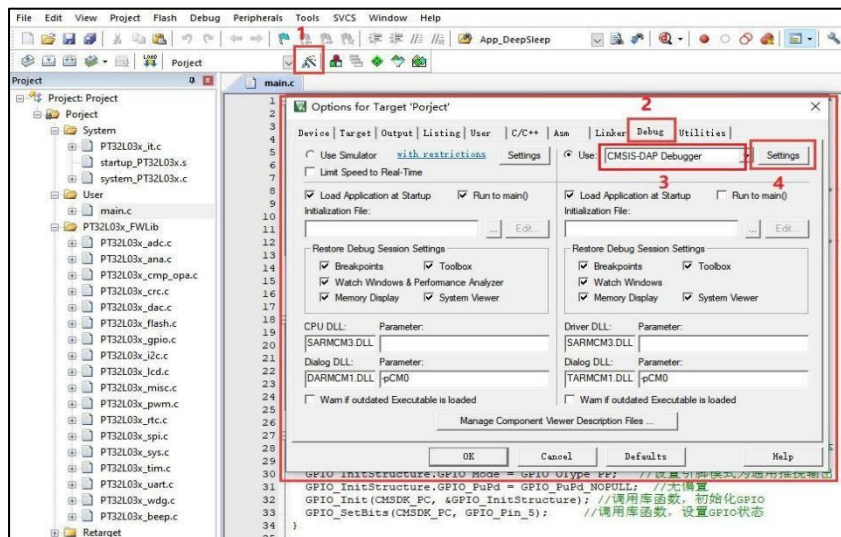
图 10 确认 MCU 与支持包匹配



5.3 确认仿真器型号匹配

Demo 板集成了“CMSIS-DAP”仿真器，仿真器选型错误将无法识别 MCU，如下图所示：

图 10 确认仿真器型号匹配



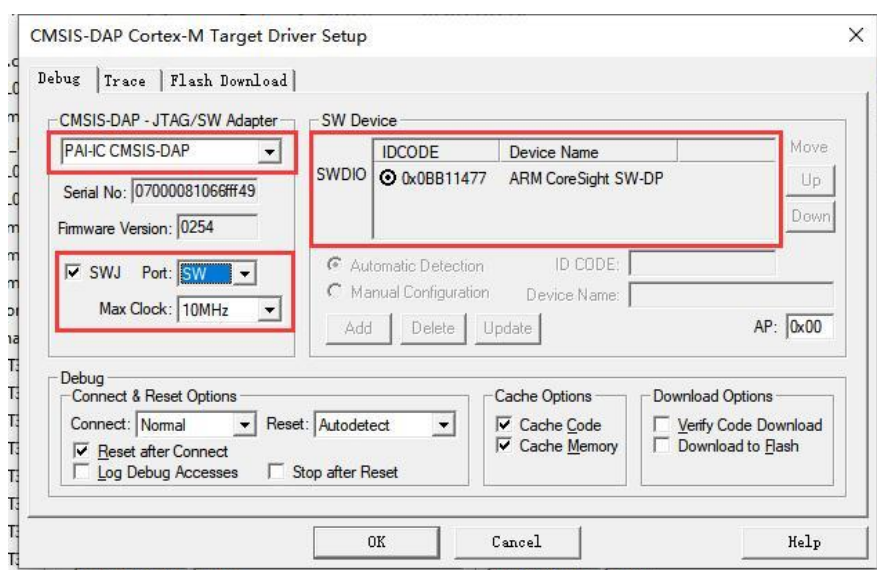
5.4 仿真器配置

点击“Setting”以在开发软件中配置仿真器的详细信息，令开发软件连接到 MCU，例程的仿真器配置应当为如下：

- 仿真器选择 PAI-IC-CMSIS-DAP
- 勾选了 SWJ 选项
- 在 Port 下拉列表中选择了 SW
- 最大时钟频率设为 10Mhz

确保仿真器配置正确，此时，“SW Device”窗口下应该就能够看到芯片信息，表示工具已经正确连接，并可以下载程序到开发板，如下图所示：

图 11 识别到 MCU 型号



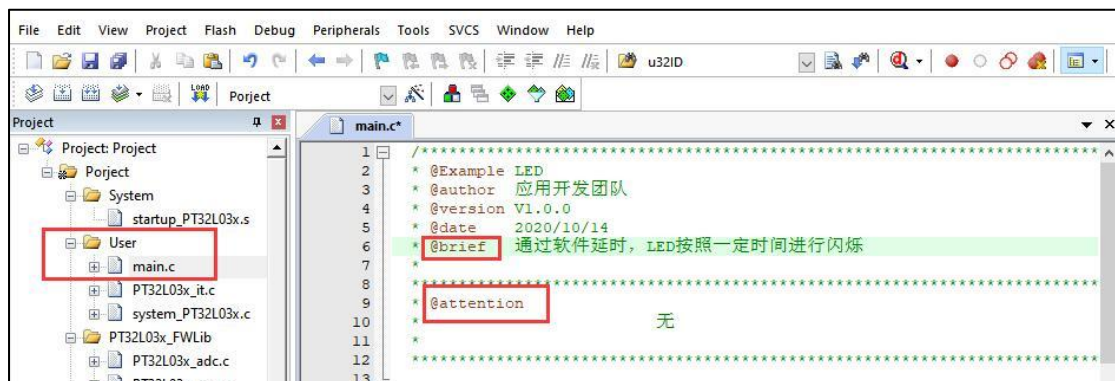
6 例程

双击打开任一例程的工程文件，如“01_PT32L03x_LED”例程“MDK”文件夹下的“Project.uvprojx”工程文件。

打开工程文件后，在 User 工程目录下，就包含了 main.c 文件。

在 main.c 的头部，包含了应用团队编辑的“brief”信息和“attention”信息，这些信息提供了如何将例程与 Demo 板配合使用，如下图所示：

图 12 例程



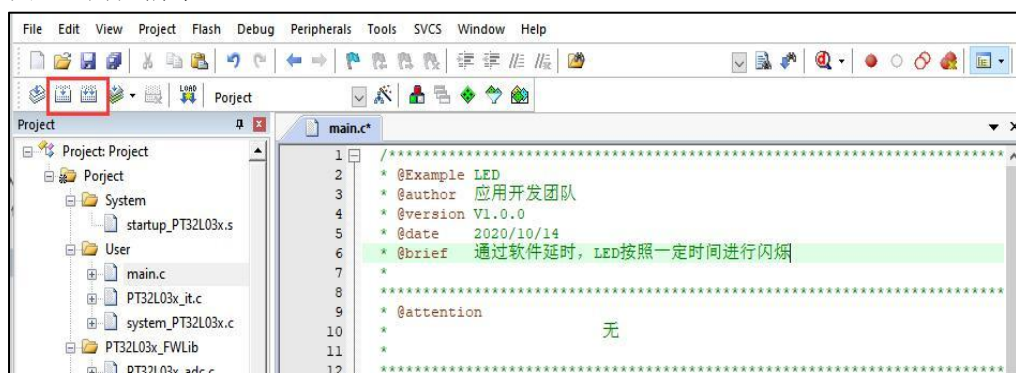
注意：

1. 下述支持包以“PAI-IC.PT32x_DFP.1.0.2.pack”为例。
2. 下述板级以“PT32L033_Demo”板为例。

6.1 例程编译

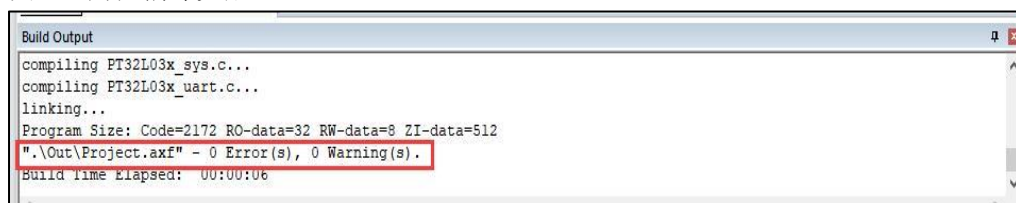
编译程序，直接按快捷键 F7,或点击工具栏上编译按钮，如下图所示：

图 13 例程编译



编译过程中，Keil 会弹出编译信息窗，用户可查看是否编译成功，如下图所示：

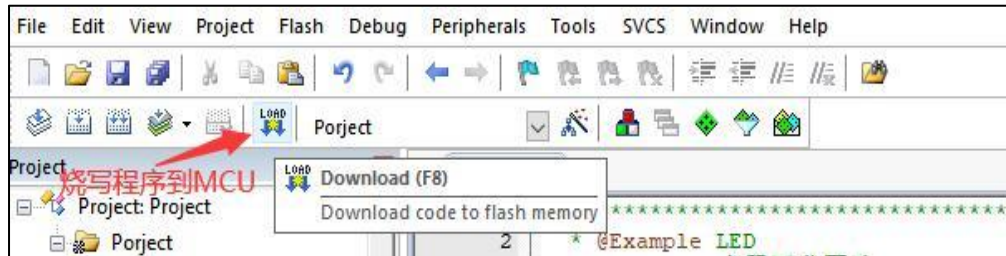
图 14 例程编译完成



6.2 例程烧录

按快捷键 F8,或点击工具栏上的“Download”项, 如下图所示:

图 15 例程烧录



6.3 例程调试

6.3.1 开始调试

点击工具栏上的调试按钮, 开始调试 (如果代码有重新编译过, 会自动启动烧写步骤), 如下图所示:

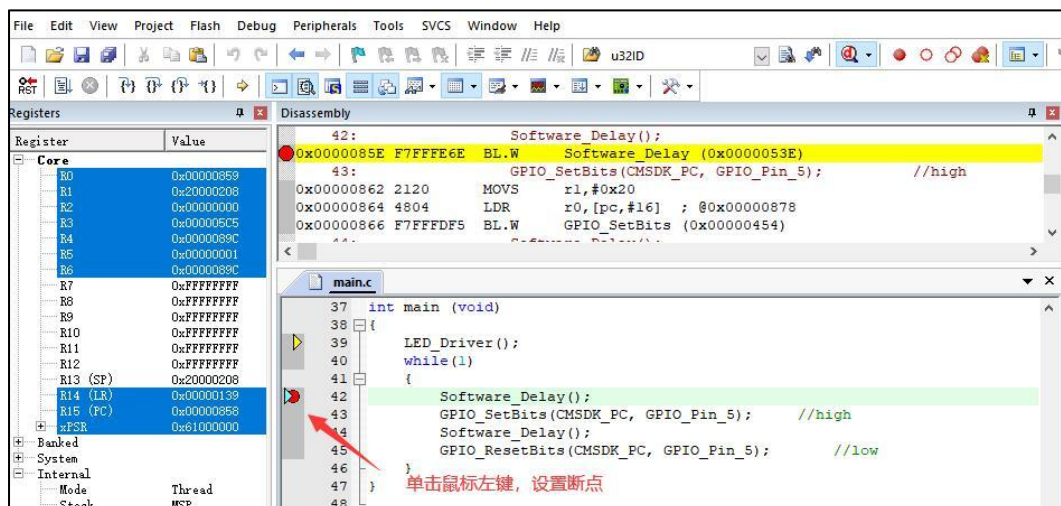
图 16 调试模式



6.3.2 设置断点

程序运行断点设置如下图、在有效代码处(灰色部分)点击鼠标左键, 程序运行到此处会暂停运行:

图 17 设置断点



6.3.3 跳出断点

程序运行到断点即停止运行，此时，可以按照下述操作跳出断点继续执行程序：

- 全速执行(F5)
- 单步执行(F10,F11)

7 IDE 高级应用

高级应用旨在帮助用户在参考手册的基础上，利用 IDE 快速的了解 MCU 资源信息。为此，PT 提供了一些基于 IDE 的可视化、系统化的便捷工具，使用这些工具，能够对 MCU 进行快速配置并且加深对 MCU 全局的把控。

注意:

3. 下述支持包以“PAI-IC.PT32x_DFP.1.0.2.pack”为例。
4. 下述板级以“PT32L033_Demo”板为例。

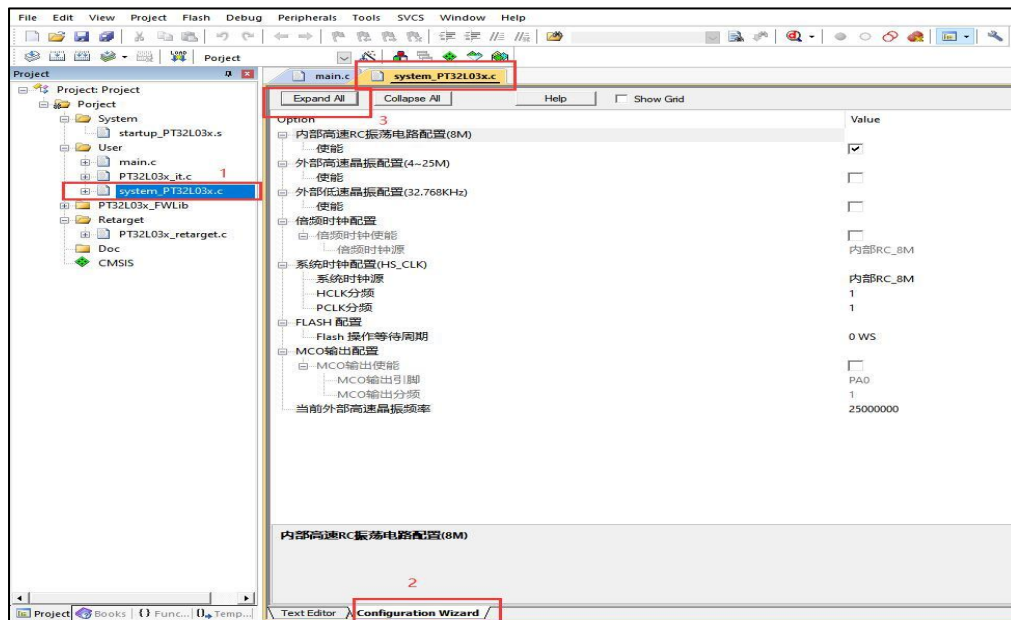
7.1 配置向导

PT 为旗下所有的 MCU 提供了可视化的时钟配置方式(配置向导)，配置向导被封装在“system_pt32L03x.c”文件当中，接下来，将以任一例程为例、简述如何通过可视化配置、配置 MCU 的系统时钟。

注意:

1. 例程使用默认系统时钟配置，如需修改，请做好备份!
2. MCU 存在外部晶振，且外部晶振更换时，应该同步修改“当前外部高速晶振频率”项目，单位 (hz)

图 18 配置向导



- 1.打开“system_pt32L03x.c”文件
- 2.点击进入“Configuration Wizard”的界面
- 3.点击“Expand All”，展开所有子选项

7.2 SVD 文件

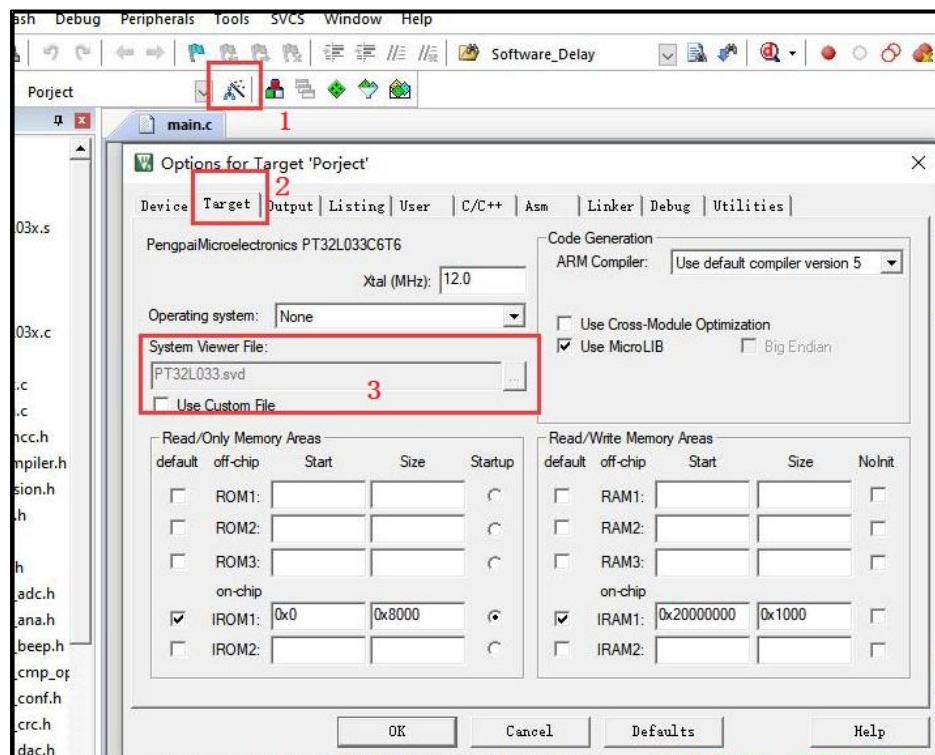
PT 为旗下所有的 MCU 提供了“System Viewer File”文件；

System Viewer File 集成了所有片内资源的地址，与用户手册的外设一一对应，通过它，可以非常方便的在仿真模式下，即时的获取到目标外设/寄存器的信息。

7.2.1 确认当前 SVD 文件与 MCU 匹配

确保按照“5.2 确认 MCU 与支持包匹配”一节、选择正确的 Device，接着操作步骤如下图所示：

图 19 确认 SVD 文件与 MCU 匹配

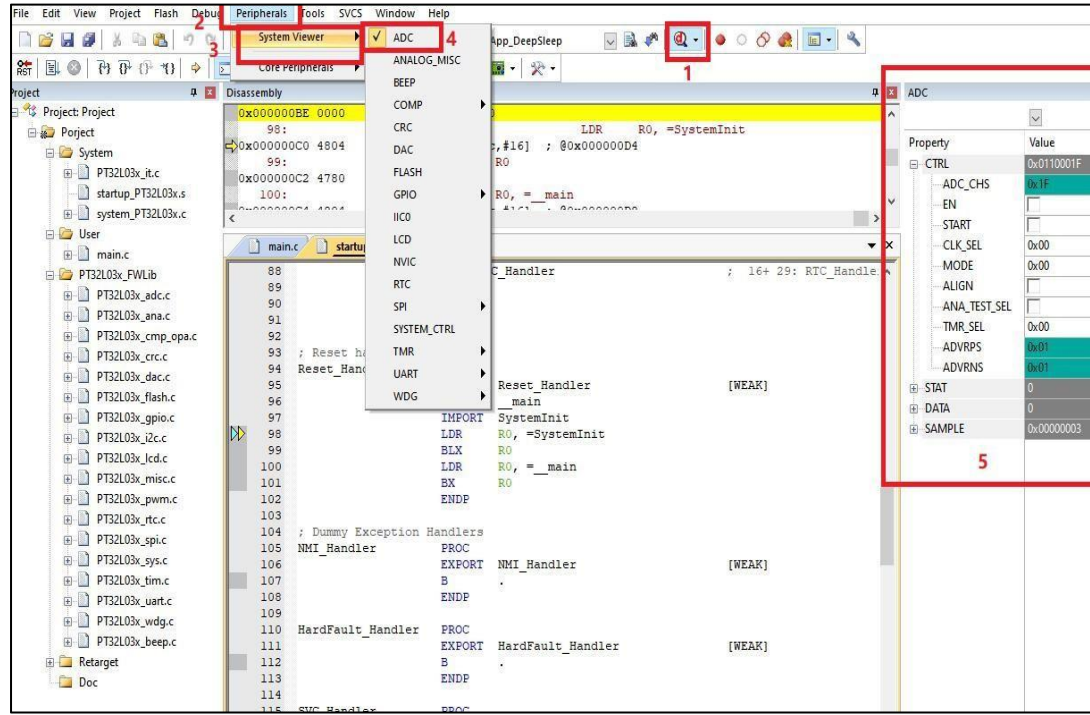


- 1. 点击 Options
- 2. 进入 Target 选项
- 3. 确认 SVD 文件与 MCU 匹配

7.2.2 应用 SVD 文件

SVD 文件的使用需要在仿真模式下，操作步骤如下图所示：

图 20 应用 SVD 文件

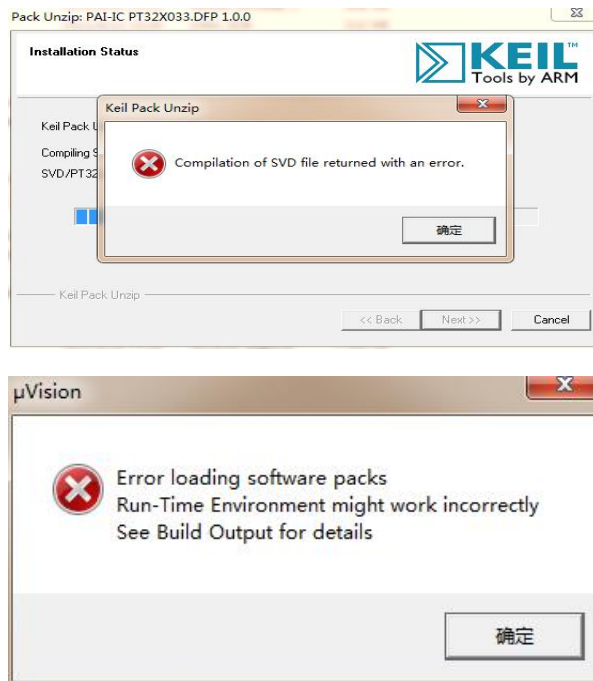


- 1.进入仿真模式
- 2.选择 Peripherals
- 3.点击 System Viewer 子选项
- 4.从 Peripherals 表中选择需要观察的外设
- 5.从 MDK-ARM 右边弹出的 System Viewer 视图中观察目标外设的值

8 常见问题

8.1 安装开发支持包提示错误

错误提示如下两张图：

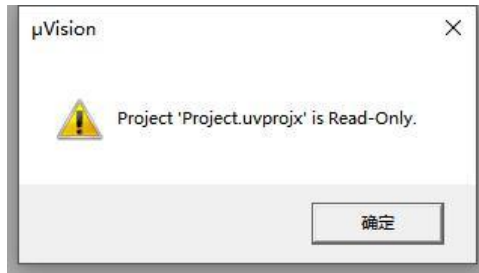


8.1.1 解决办法

出现该提示是由于当前系统上安装的 MDK 开发软件是基于 C51 或者其他内核的开发环境，请参考“2 开发环境安装”，安装基于 ARM 内核的 MDK 开发环境。

8.2 打开例程提示只读警告

错误提示如下图：



8.2.1 解决办法

出现该警告是由于文件属性为只读，按照 `readme.txt` 文档描述即可。

8.3 我想把 I/O 另做它用

当你想使用某个 IO 口用作其他用处的时候，请先查看板级的原理图，确定如下信息：

- 该 IO 口是否有连接在板级的某个电路上？
- 如果有，该电路的信号是否会对你的使用造成干扰？

先确定有无干扰，再使用这个 IO。

8.4 例程运行的结果不理想

每个例程，均经过我们应用工程师的测试，并且在例程的“`main.c`”源文件内，均有说明注意事项，请确定当前操作是否严格按照例程的注意事项，如有疑问，请联系我们。

8.5 仿真无法进入 main 函数

一般有两种可能：

- 当前 mcu 烧录的程序，与仿真时的程序不一致，可以重新编译一次，再进行烧录仿真。
- 程序中使用了 c 语言标准库依赖函数，比如 `printf`。而工程里面又没有包含“Retarget”目录下的 `PT32Xxxx_retarget.c` 和 `PT32Xxxx_serial.c` 文件或者没有勾选 keil 中的“use MicroLib 选项”

这种情况下，可以按上述的两个原因去排查。

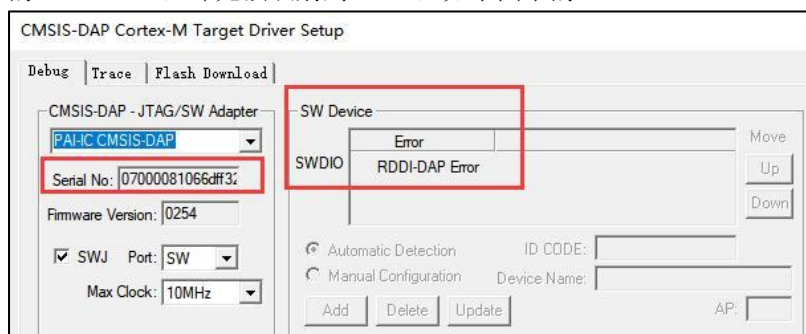
8.6 使用 DEMO 板，设备管理器无法识别到串口

DEMO 板集成的 USB 转 TTL 功能，如果连接 DEMO 板，却无法在设备管理器中找到串口，应当确认：

- 1.确认当前的系统是否为 win10 系统，集成的 USB 转 TTL 功能仅支持 WIN10 及以上版本。
- 2.如果满足 1，则排查板级，是否存在连接不稳等情况。

8.7 SWD 连接正常，却无法识别到 MCU

确认 SWD 接口连接正常后，此时 MDK-ARM 能够正常识别到仿真器，如下图中的“Serial No”，却无法识别到 MCU，如下图中的“SW Device”。

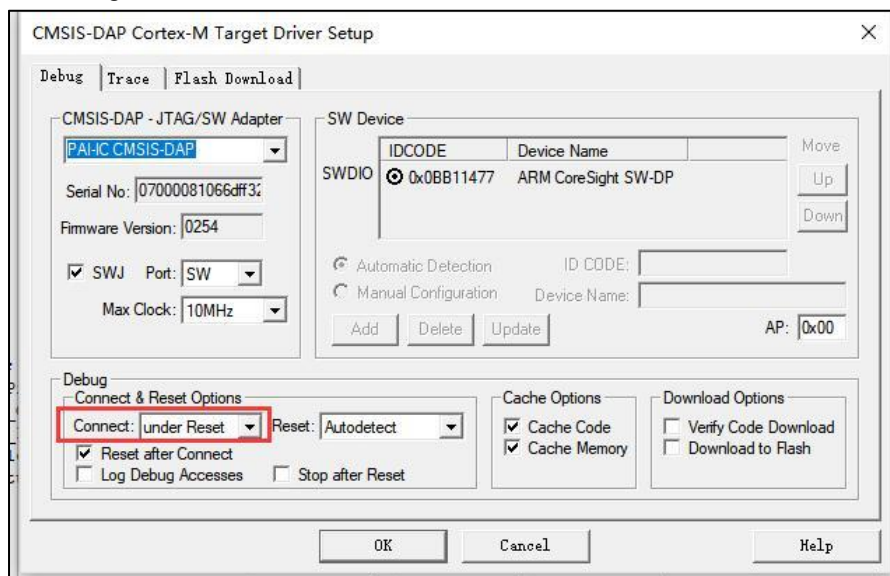


再排除仿真器到 MCU 间的线路不存在问题之后，有可能的原因如下：

- MCU 处于深度睡眠状态
- SWD 的 IO 引脚功能被复用

8.7.1 解决措施

将仿真器的复位信号“NRST”与 MCU 的复位脚“NRST”连接，在仿真器的配置界面下的“Debug”菜单中，配置为下图中的 under Reset 形式，点击“OK”以应用设置即可：



9 版本历史

表 1 文档版本历史

日期	版本	作者	变更
2021-02-24	1.0	李联辉	初始发行
2021-05-12	1.1	李联辉	适配 pack 包
2021-06-01	1.2	李联辉	新增常见问题 5.5
2022-02-08	1.3	李联辉	1.新增概述 2.兼容性设计 3.新增“6 IDE 仿真高级应用配置” 4.新增“常见问题 7.6” 5.新增“常见问题 7.7” 6.新增“例程”一节 7.新增“7.1 配置向导”一节