

AT32 IDE Project Generate 用户手册

前言

本手册用于介绍使用 AT32IDE_Project_Generate 工具把 MDK 工程转换为 AT32 IDE 工程的使用方法。

目录

1	引言.....	5
1.1	环境要求.....	5
2	软件操作.....	6
2.1	MDK5 工程转 AT32 IDE 工程.....	6
2.1.1	Windows 系统中使用.....	6
2.1.2	Linux 系统中使用.....	7
2.2	AT32 IDE 导入工程与编译.....	7
2.3	编译工程.....	9
2.4	调试工程.....	11
2.5	命令行参数.....	14
2.6	返回代码.....	14
3	文档版本历史.....	15

表目录

表 1 命令行参数列表.....	14
表 2 return codes 列表	14
表 3. 文档版本历史	15

图目录

图- 1 命令行界面.....	6
图- 2 批处理文件.....	6
图- 3 AT32 IDE 工程配置文件夹.....	7
图- 4 Linux 系统程序	7
图- 5 Shell 脚本文件	7
图- 6 导入 AT32 IDE 工程方式一.....	8
图- 7 导入 AT32 IDE 工程方式二.....	9
图- 8 工具栏编译工程	9
图- 9 编译工程	10
图- 10 构建结果	10
图- 11 AT-Link Main 标签页	11
图- 12 AT-Link Debugger 标签页.....	12
图- 13 启动调试	12
图- 14 切换透视图	13
图- 15 调试透视图	13

1 引言

1.1 环境要求

■ 软件要求

Windows 系统

需要 Windows 7 及以上操作系统支持。

Linux 系统

目前支持 Ubuntu、Fedora 等系统。

2 软件操作

2.1 MDK5 工程转 AT32 IDE 工程

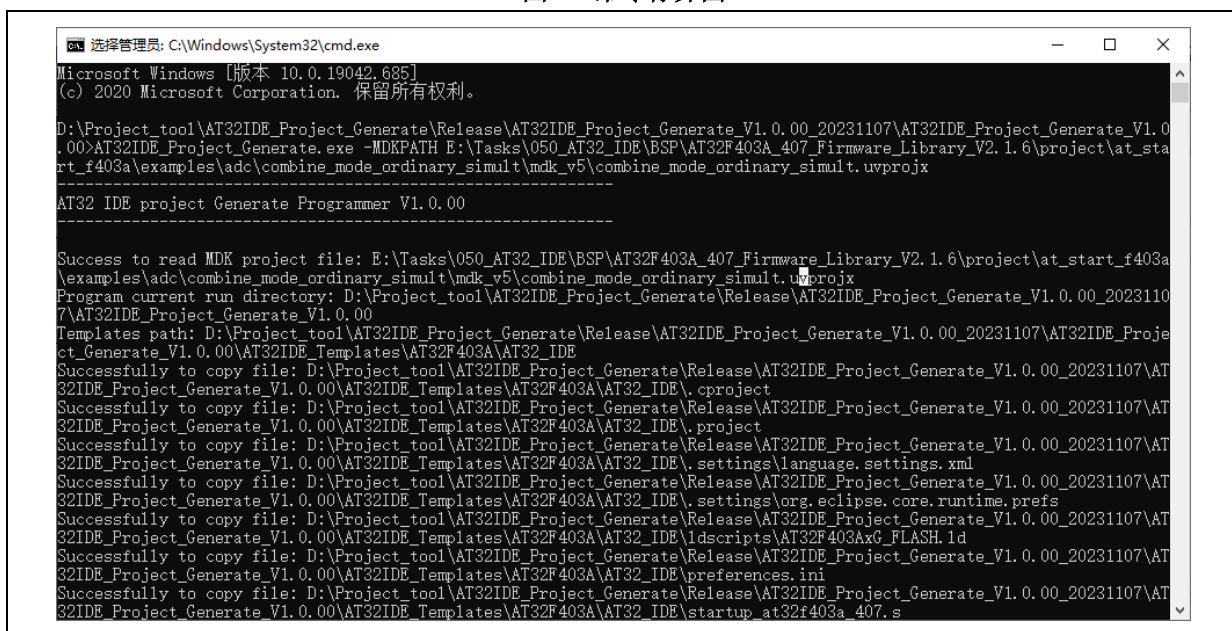
AT32_Proj_Generate 软件为命令行工具，使用时需要输入各种参数完成工程转换任务。参数的具体定义参见 2.5 章节。

2.1.1 Windows 系统中使用

使用方式 1：命令行下输入参数

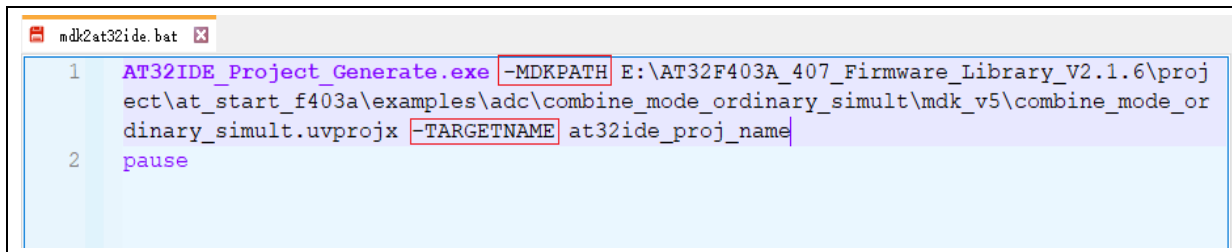
执行命令：AT32_Proj_Generate.exe -MDKPATH E:\...\lvgl_music_player.uvprojx

图-1 命令行界面



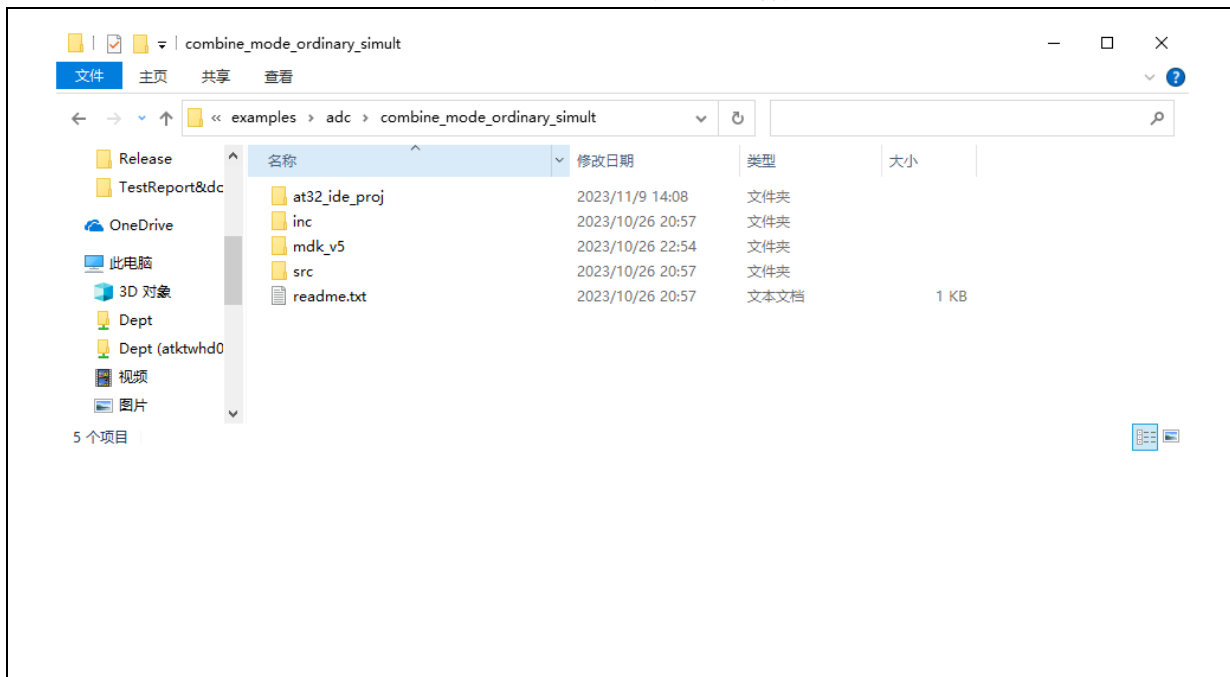
使用方式 2：使用批处理文件（常用方式，可参考 mdk2at32ide.bat），修改批处理文件中参数“-MDKPATH”与“-TARGETNAME”的内容，然后双击批处理文件自动执行。“-MDKPATH”指定将要转换的 MDK 工程文件的完整路径。“-TARGETNAME”指定转换后的 AT32 IDE 工程名称，如没有设置，将使用 MDK 的工程名称。

图-2 批处理文件



工程转换完成后，将在原来 MDK 工程文件夹的父文件夹创建一个名为“at32_ide_proj”的文件夹，存放转换得到的 AT32 IDE 工程文件。如下图：

图- 3 AT32 IDE 工程配置文件夹



2.1.2 Linux 系统中使用

1. 编辑 mdk2at32ide.sh 脚本，按照 2.5 章节中的命令行参数介绍设置工程转换的参数。并增加脚本执行权限。命令：chmod +x mdk2at32ide.sh。
2. 在终端中执行 mdk2at32ide.sh 脚本，命令：./mdk2at32ide.sh。

图- 4 Linux 系统程序

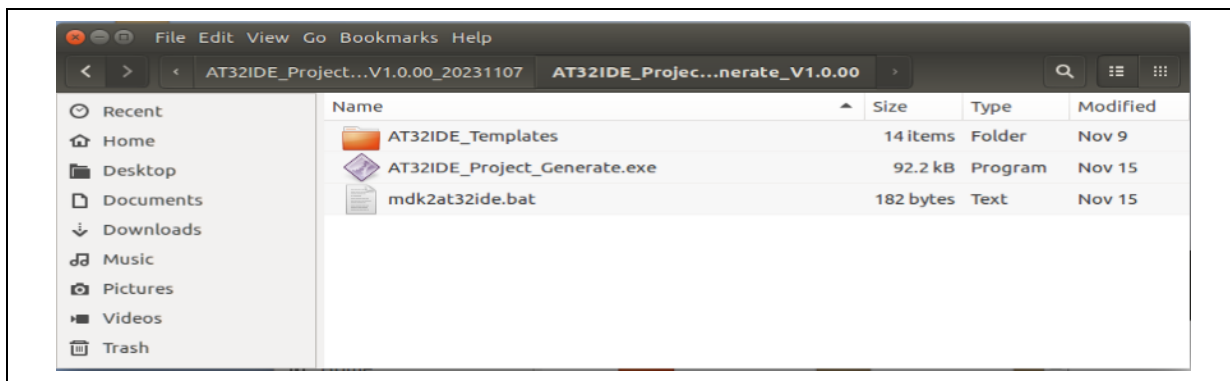
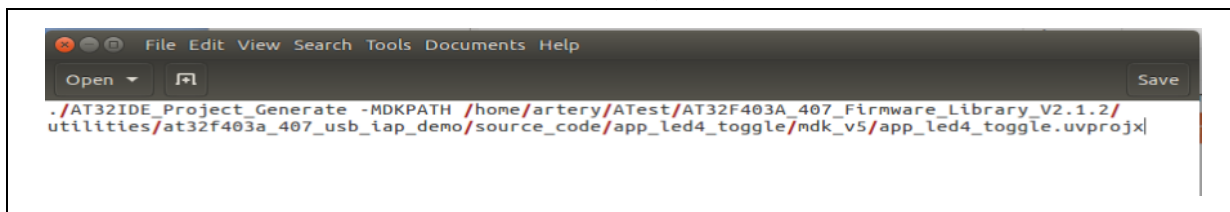


图- 5 Shell 脚本文件

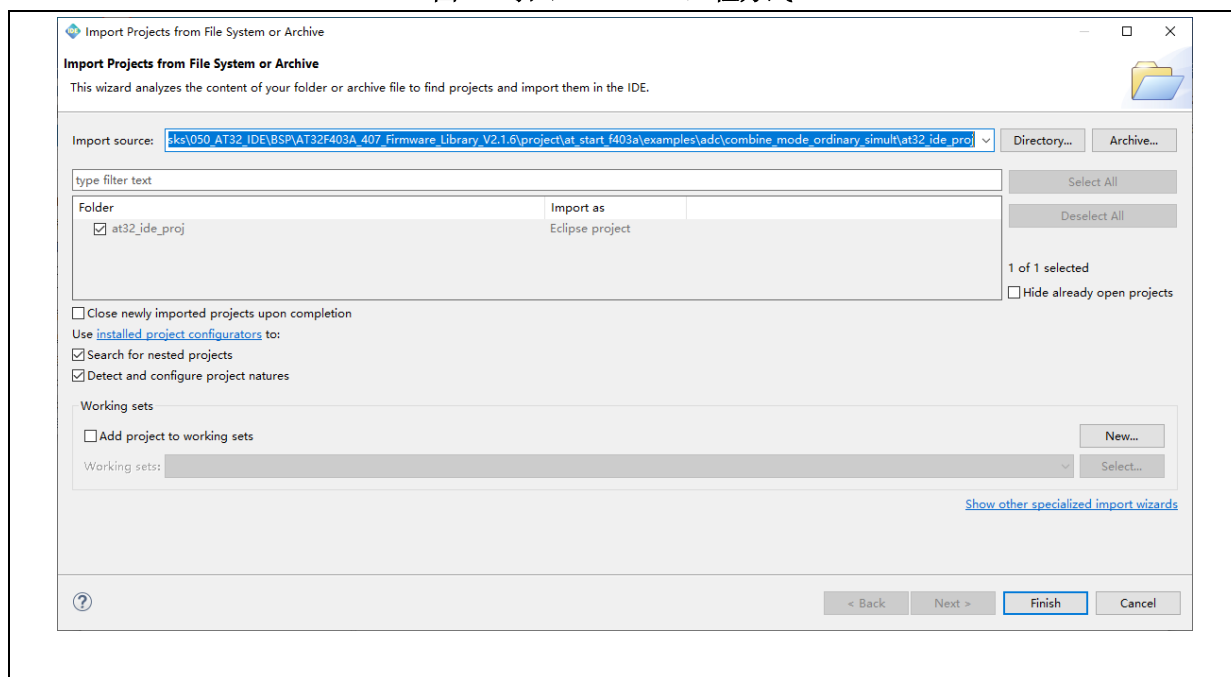


2.2 AT32 IDE 导入工程与编译

由 MDK5 工程转换得到的 AT32 IDE 工程，需要导入到 AT32 IDE 中进行编译和调试，这里有两种方式导入工程。

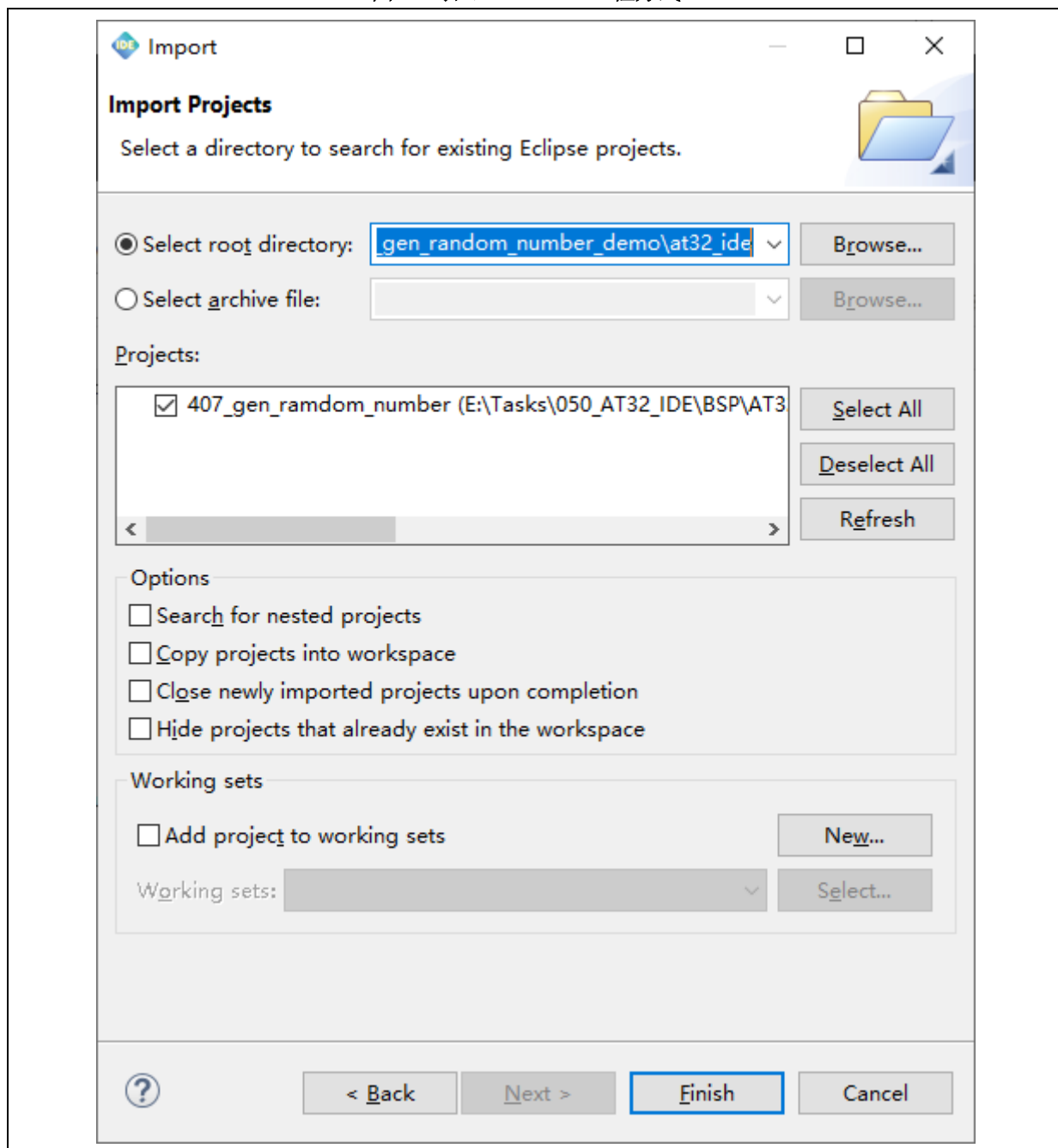
方式一：菜单【File】中点击【Open Projects From File System...】，在弹出的对话框中，【Import source】选择前面生成的工程文件夹名称“at32_ide_proj”，然后点击【Finish】按钮完成工程的导入。

图-6 导入 AT32 IDE 工程方式一



导入方式二：菜单【File】中点击【Import】，然后选择【General->Existing Projects into Workspace】，在弹出的对话框中，默认勾选的是“Select root directory”，并通过【Browse...】按钮选择工程目录，然后点击【Finish】按钮完成工程的导入。

图-7 导入 AT32 IDE 工程方式二



2.3 编译工程

右键单击工程，在弹出菜单中选择“Build Project”，或者选择工程，点击菜单栏的“Build Project”对工程进行编译，生成 elf 文件。如下图所示：

图-8 工具栏编译工程

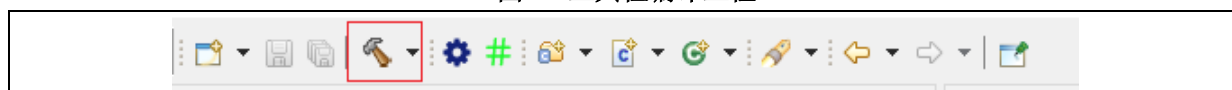
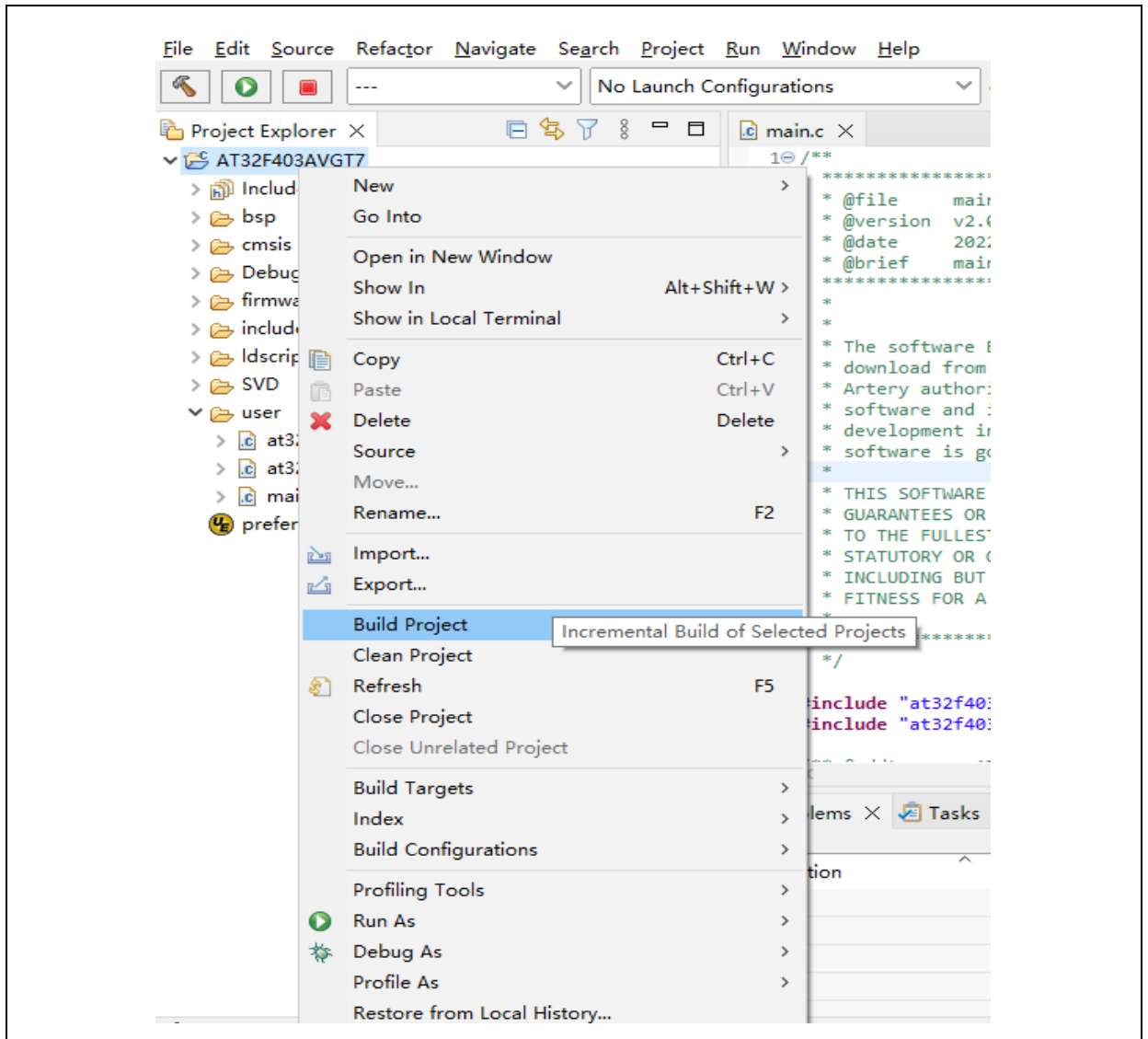
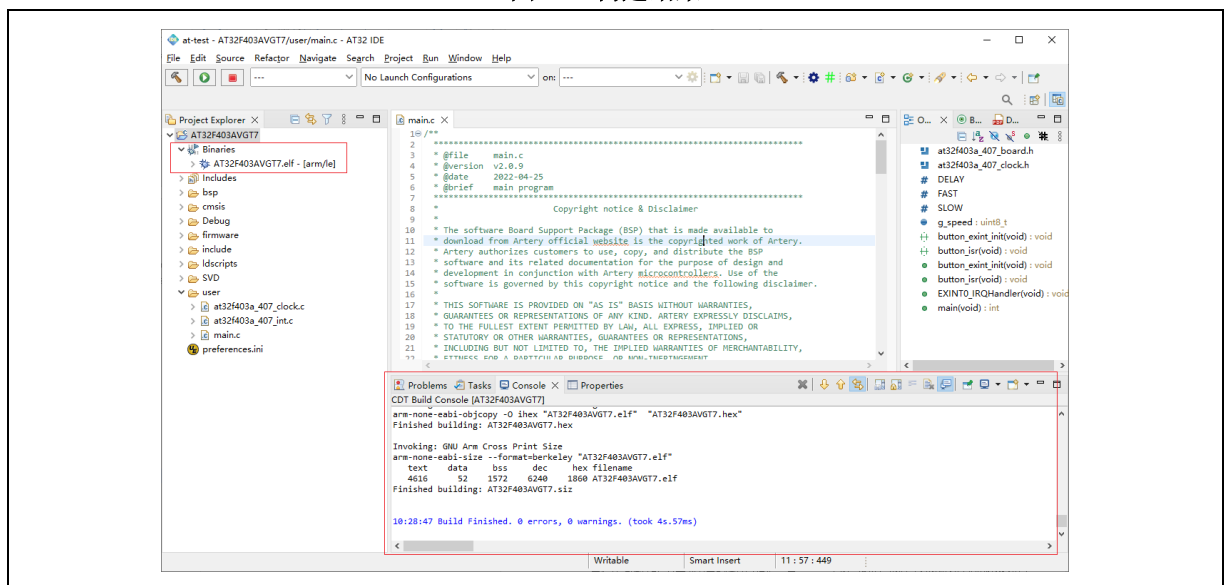


图-9 编译工程



编译的工程日志在控制台进行打印。如下图所示：

图-10 构建结果

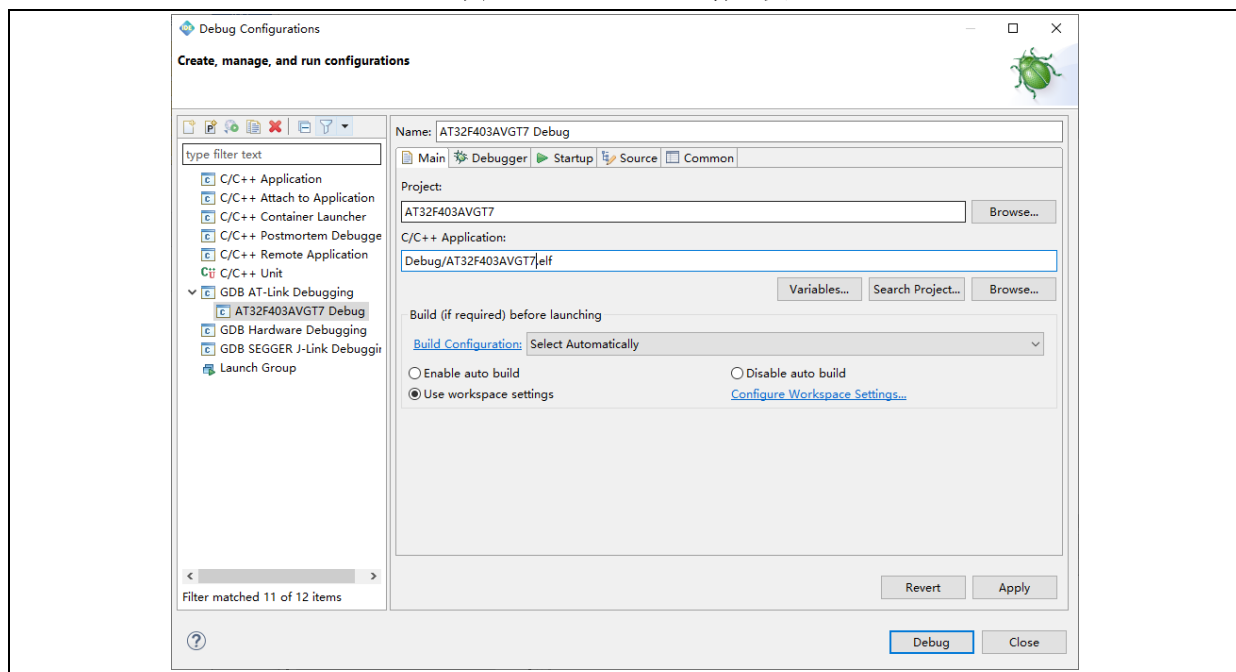


2.4 调试工程

选择菜单【Debug】>【Debug Configurations...】打开调试配置对话框。双击 GDB AT-Link Debugging，将自动创建 AT-Link 调试配置，并在右边进行显示。所有的调试配置信息都会自动生成，一般可以不需要进行修改。

Main 标签页中，“Project”为当前的工程名，创建时会自动生成。调试器一般会搜索工程中构建的 elf 文件，这里自动填写 Debug\xxxxx.elf。如果没有提前构建工程，找不到 elf 文件，则自动填写 Debug\\${project_name}.elf，当构建产生以工程名称命名的 elf 时，这里的配置也可以正常使用。

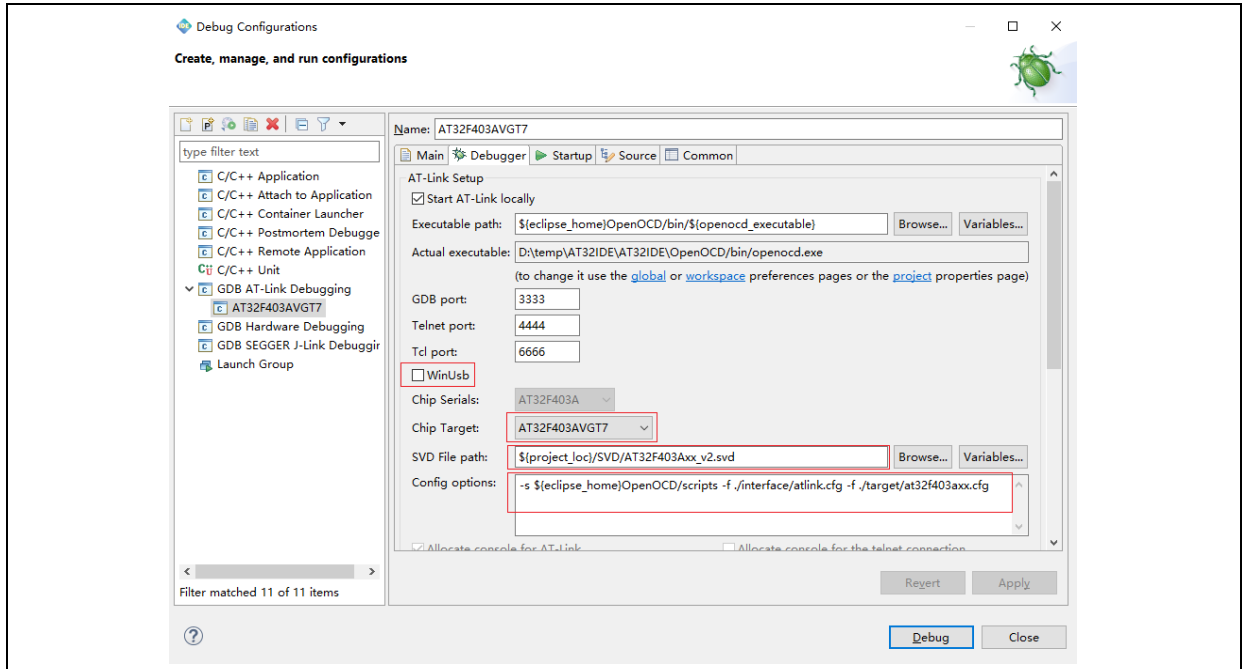
图- 11 AT-Link Main 标签页



Debugger 标签页中的信息创建时也自动进行了配置。OpenOCD 的执行程序和脚本文件配置路径是相对路径，因为这些工具都已存放在 AT32 IDE 的安装文件中。这里自动配置了执行程序路径、SVD 文件路径，并根据芯片类型配置了 Config options。当然也可以根据实际情况就行修改。如需要使用 WinUSB，则需要勾选“WinUSB”，则 Config options 的信息中将自动配置为“-f ./interface/atlink_dap_v2.cfg”。

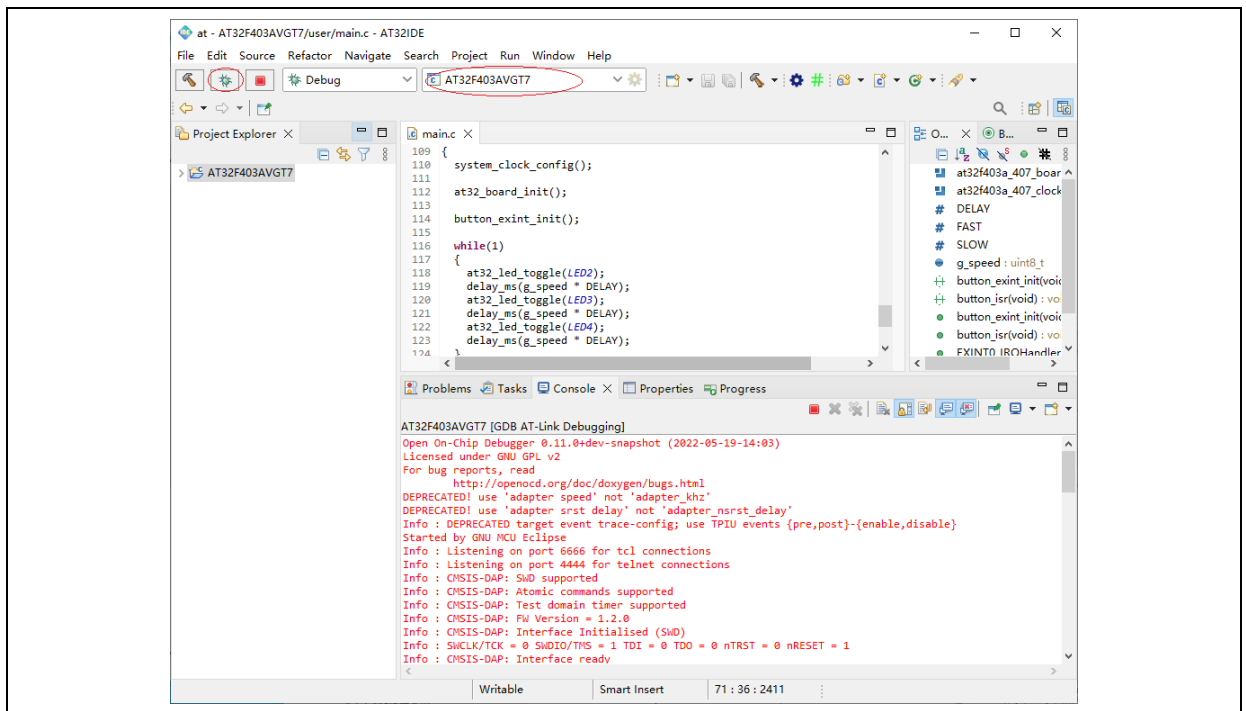
如下图所示：

图- 12 AT-Link Debugger 标签页



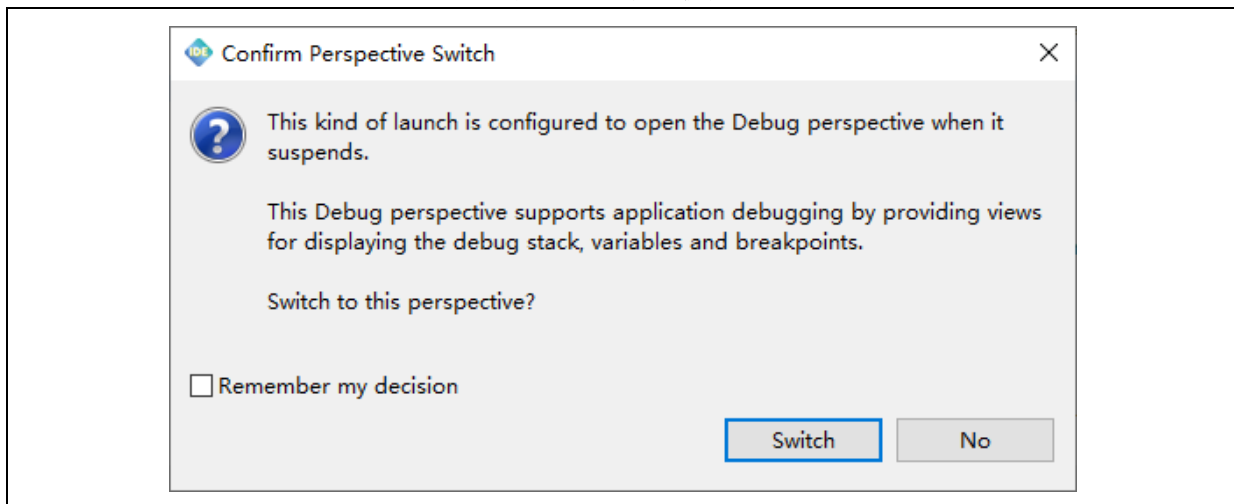
调试器配置完成后，点击工具栏“Launch in ‘Debug’ mode”按钮，启动工程调试。如下图所示：

图- 13 启动调试



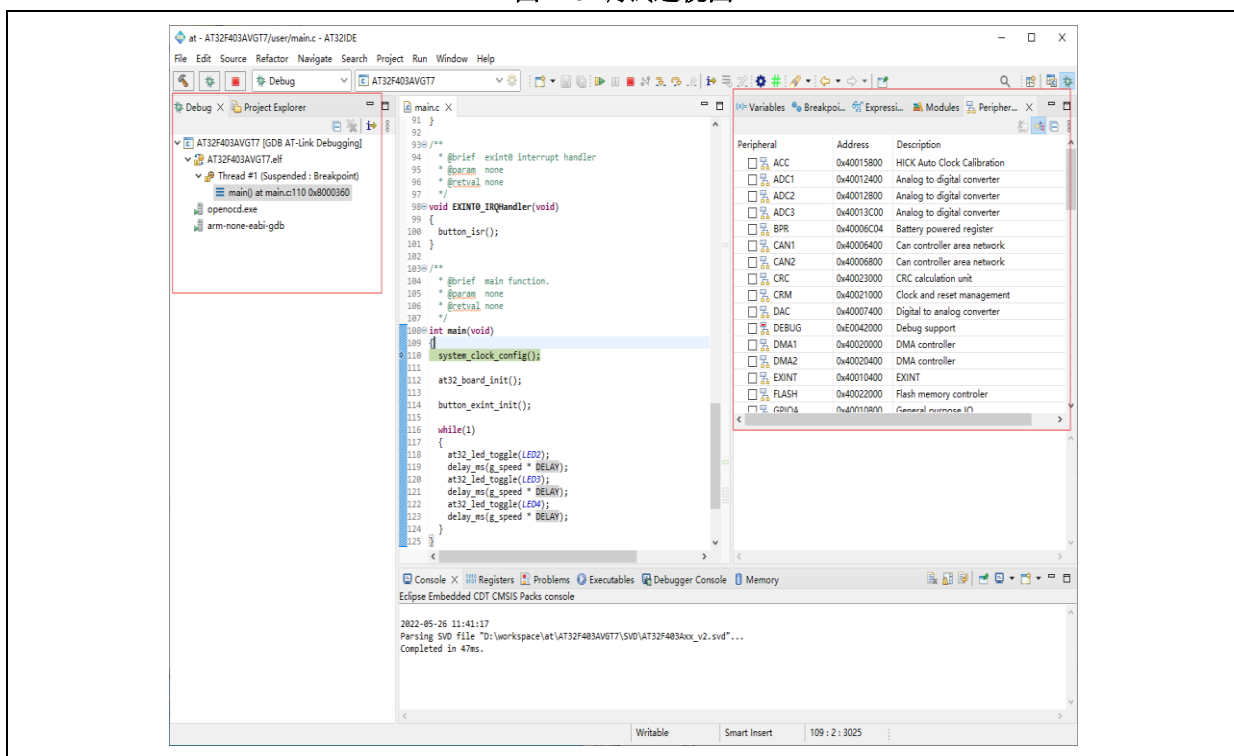
弹出切换透视图对话框，点击“切换”按钮切换到 Debugger 透视图。如下图所示：

图-14 切换透视图



工程启动调试，并切换到调试透视图。在调试透视图可以进行各种调试功能操作。如查看变量值、表达式值、外围寄存器等。如下图所示：

图-15 调试透视图



2.5 命令行参数

表 1 命令行参数列表

Command	Sub Command	Remarks
-MDKPATH		Specify the full path to the MDK project file “.uvprojx”.
-TARGETNAME		AT32 IDE project name.If this parameter is not set, the target project name is the original MDK project name.

2.6 返回代码

执行 AT32IDE Project Generate 命令，当执行成功时 return code（Errorlevel）为 0，执行失败时为-1。

表 2 return codes 列表

Return code	Command	Error
0	All	Executed successfully.
-1	All	Executed failure.

3 文档版本历史

表 3. 文档版本历史

日期	版本	变更
2023/11/09	V1.00	1. 初始版本。

重要通知 - 请仔细阅读

买方自行负责对本文所述雅特力产品和服务的选择和使用，雅特力概不承担与选择或使用本文所述雅特力产品和服务相关的任何责任。

无论之前是否有过任何形式的表示，本文档不以任何方式对任何知识产权进行任何明示或默示的授权或许可。如果本文档任何部分涉及任何第三方产品或服务，不应被视为雅特力授权使用此类第三方产品或服务，或许可其中的任何知识产权，或者被视为涉及以任何方式使用任何此类第三方产品或服务或其中任何知识产权的保证。

除非在雅特力的销售条款中另有说明，否则，雅特力对雅特力产品的使用和 / 或销售不做任何明示或默示的保证，包括但不限于有关适用性、适合特定用途（及其依据任何司法管辖区的法律的对应情况），或侵犯任何专利、版权或其他知识产权的默示保证。

雅特力产品并非设计或专门用于下列用途的产品：（A）对安全性有特别要求的应用，例如：生命支持、主动植入设备或对产品功能安全有要求的系统；（B）航空应用；（C）航天应用或航天环境；（D）武器，且/或（E）其他可能导致人身伤害、死亡及财产损害的应用。如果采购商擅自将其用于前述应用，即使采购商向雅特力发出了书面通知，风险及法律责任仍将由采购商单独承担，且采购商应独力负责在前述应用中满足所有法律和法规要求。

经销的雅特力产品如有不同于本文档中提出的声明和 / 或技术特点的规定，将立即导致雅特力针对本文所述雅特力产品或服务授予的任何保证失效，并且不应以任何形式造成或扩大雅特力的任何责任。

© 2023 雅特力科技 保留所有权利