

### 前言

本文档介绍了 IAR Embedded Workbench(EW) IDE 和相应的 Puya IAR pack 的安装方法。然后从零开始新建工程，实现使用串口输出“Hello World”，并对从已有的 Microcontroller Development Kit(MDK) Version 5 – Keil 工程进行移植和调试做了简要的说明，最后对在使用 IAR 过程中可能会使用到的功能进行补充。

## 目录

<b>1</b>	<b>IAR EW for Arm 9.20.2 的安装 .....</b>	<b>3</b>
1.1	软件获取、下载 .....	3
1.2	IAR 软件安装 .....	3
1.3	IAR Pack 安装 .....	7
<b>2</b>	<b>软件使用 .....</b>	<b>9</b>
2.1	新建工程 .....	9
2.2	移植工程 .....	18
2.3	常用功能 .....	20
<b>3</b>	<b>版本历史 .....</b>	<b>23</b>

# 1 IAR EW for Arm 9.20.2 的安装

## 1.1 软件获取、下载

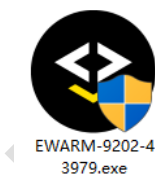
表 1 - 1 软件下载链接

软件	说明	下载链接
IAR	IAR EW for Arm 9.20.2	<a href="https://www.iar.com/products/architectures/arm/iar-embedded-workbench-for-arm/">https://www.iar.com/products/architectures/arm/iar-embedded-workbench-for-arm/</a>
IAR Pack	Puya.IAR.DFP.V01	<a href="#">Tool &amp; Software · Puya Semiconductor/Download - 码云 - 开源中国 (gitee.com)</a>

## 1.2 IAR 软件安装

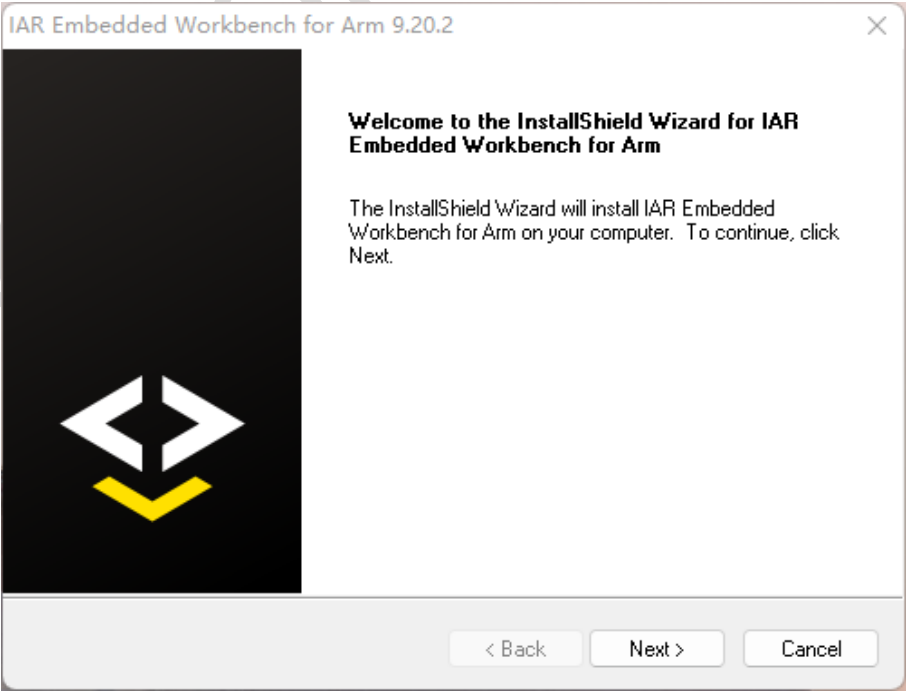
待 IAR 安装包下载完毕后，点击安装程序。

图 1-1 安装程序



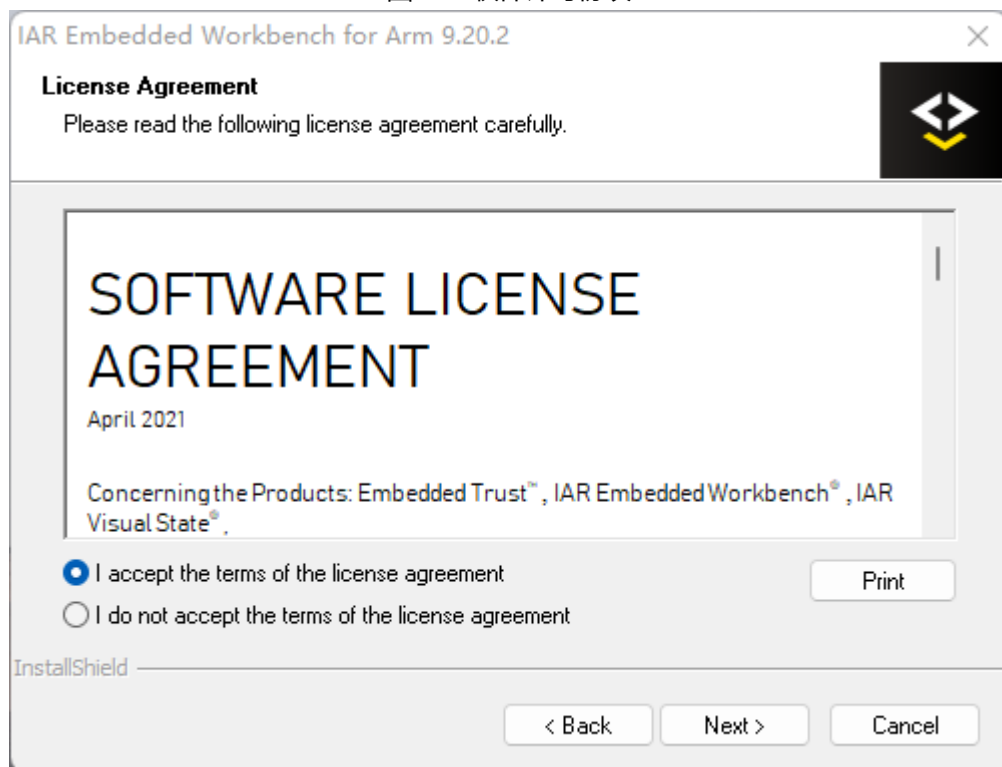
弹出如下图 1-2 的安装界面，点击 “Next” 按钮以继续安装。

图 1-2 欢迎界面



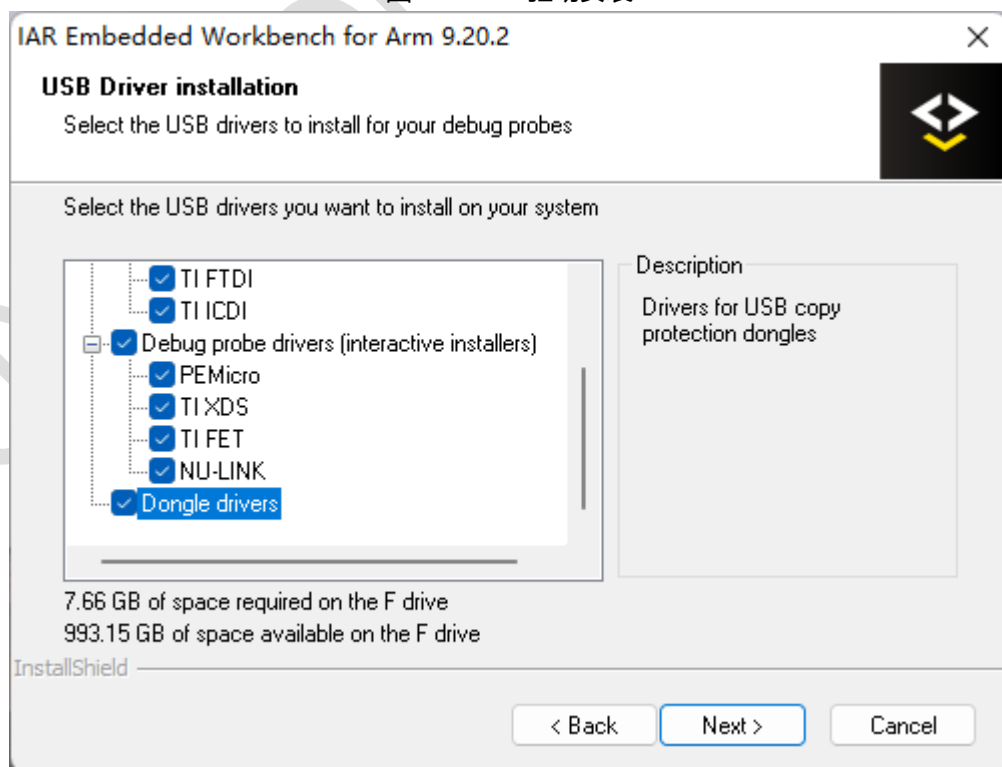
需勾选同意协议方可进行下一步的安装。

图 1-3 软件许可协议



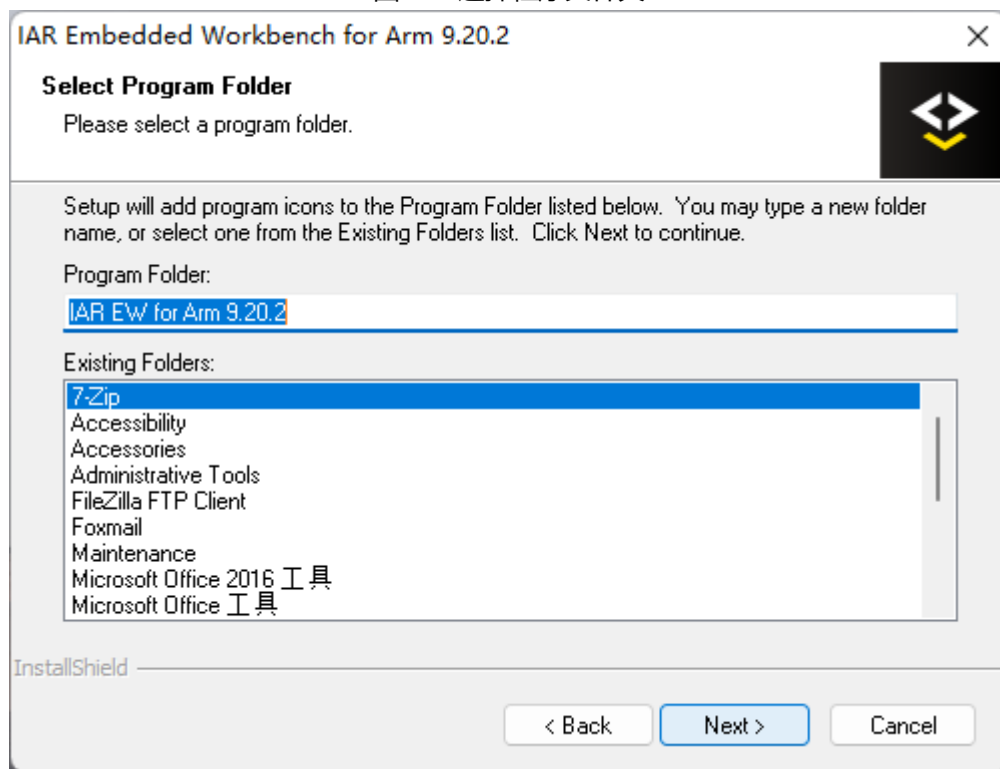
按需选择相应的 USB 驱动进行安装。

图 1-5 USB 驱动安装



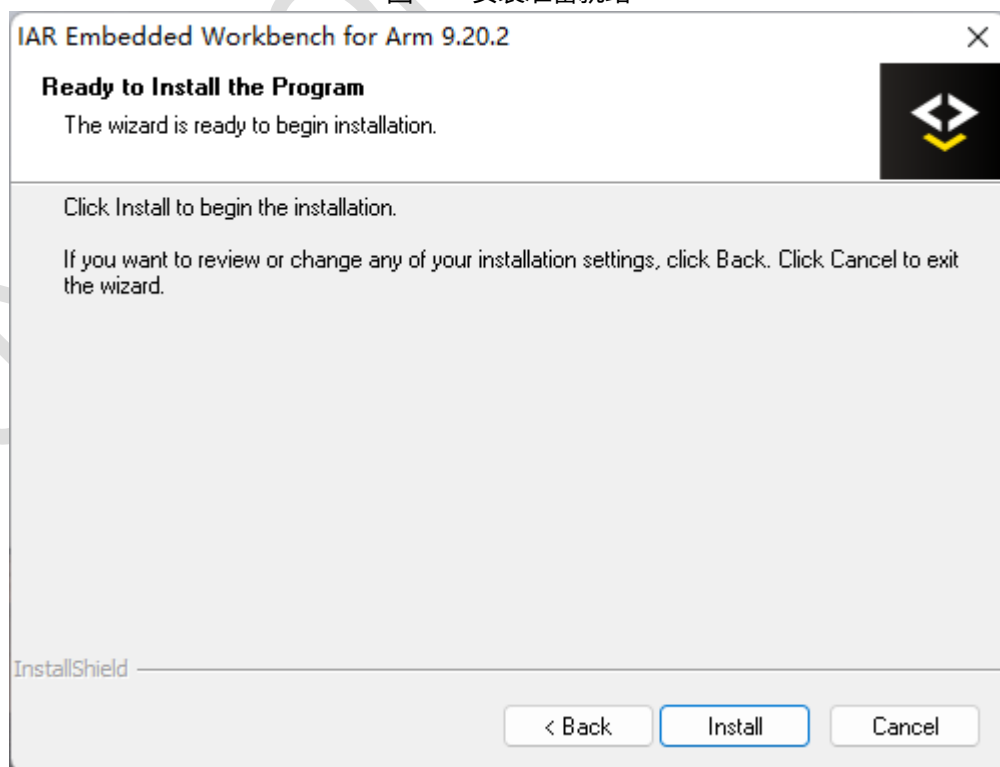
在开始菜单中创建文件夹，并点击“Next”进行下一步的安装。

图 1-6 选择程序文件夹



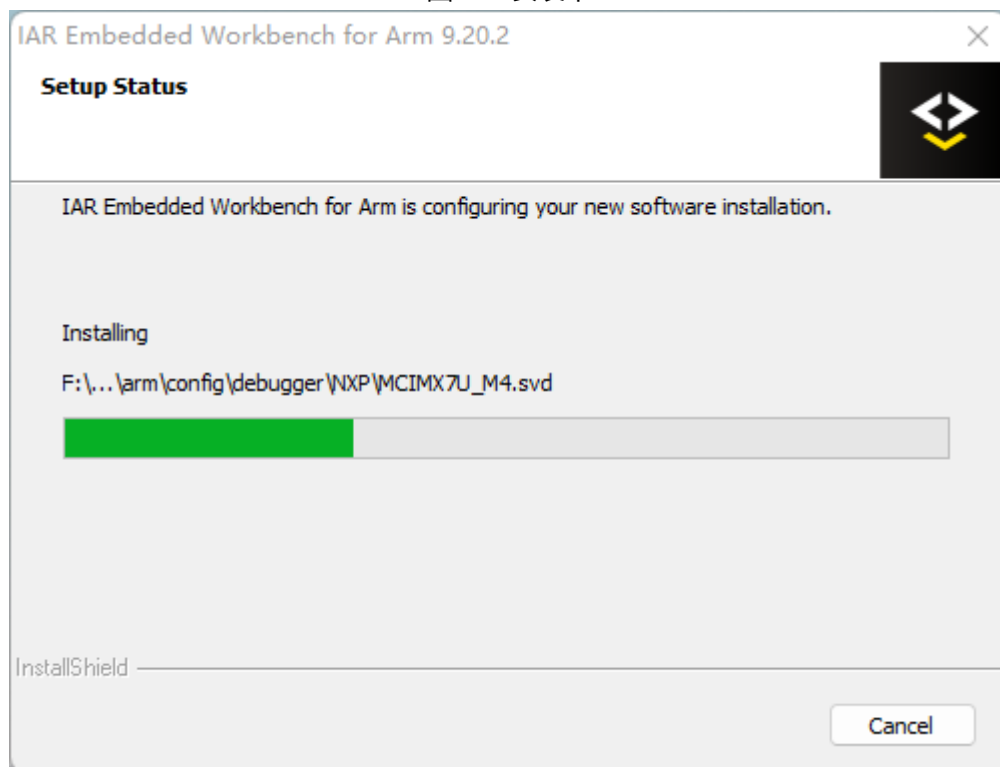
安装程序以准备好安装程序，点击“Next”进行下一步的安装。

图 1-7 安装准备就绪



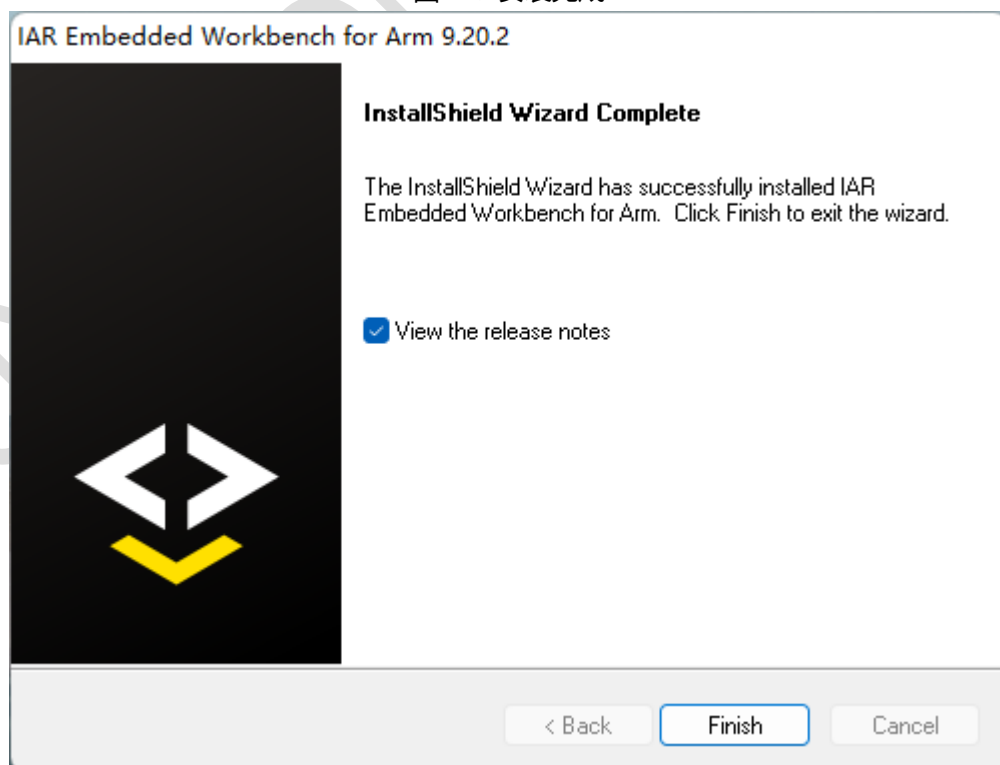
等待安装程序安装。

图 1-8 安装中



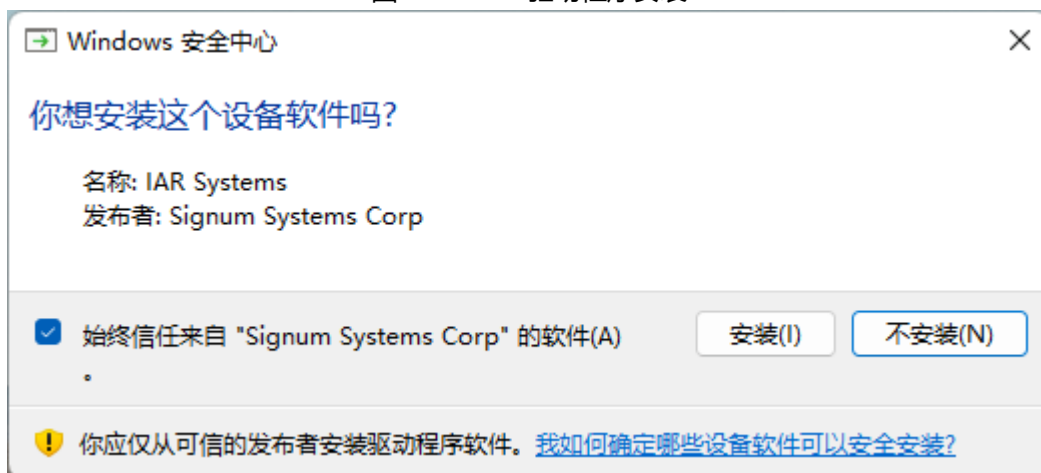
安装完成

图 1-9 安装完成



在安装完成后会安装 USB 驱动程序，此时会弹出新的窗口，点击“安装”按钮进行安装。

图 1-10 USB 驱动程序安装



### 1.3 IAR Pack 安装

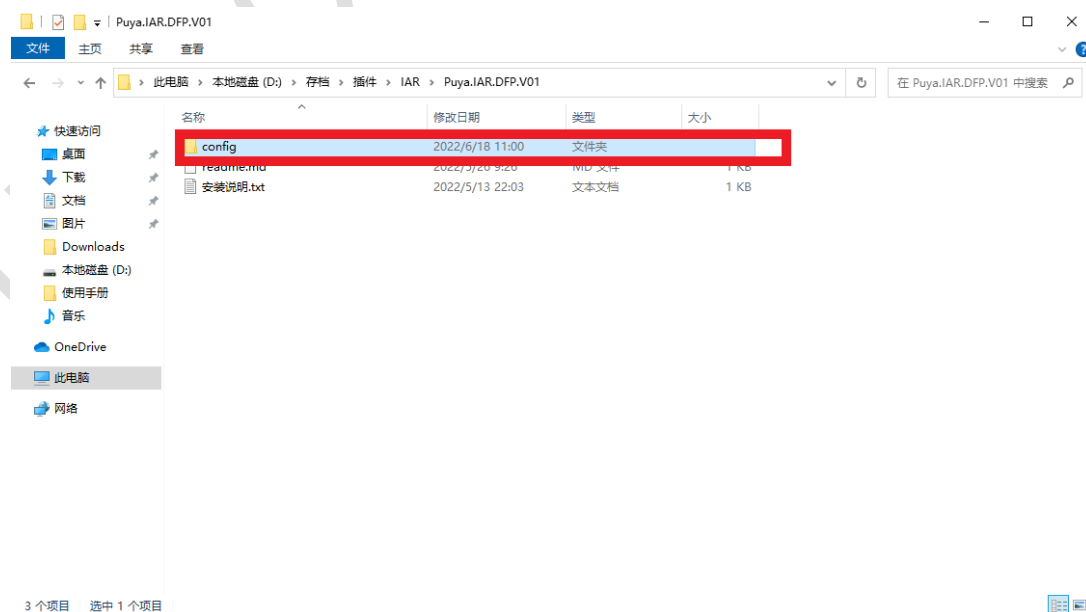
将 IAR Pack 压缩包解压。

图 1-11 IAR Pack 压缩包



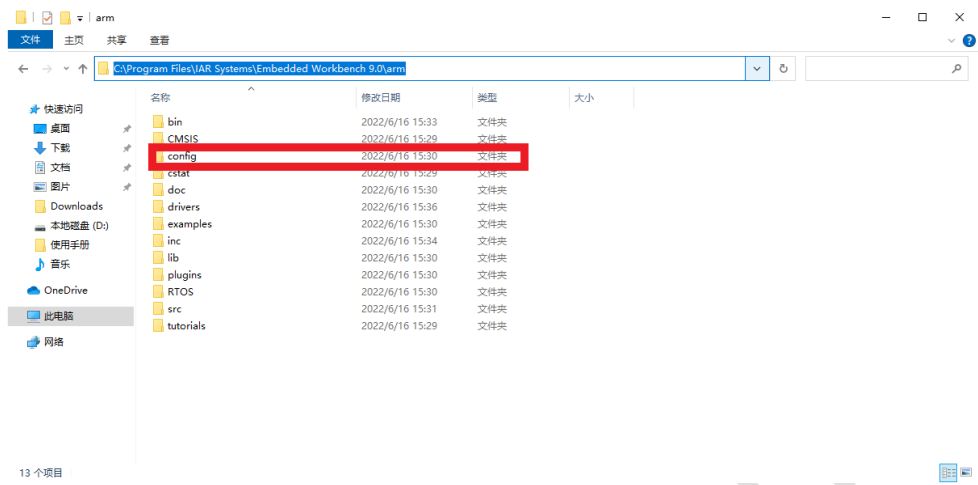
解压后得到以下文件，将下图 1-12 中框出的“config”文件夹拷贝至 IAR 安装目录下的 arm 文件夹内。如：C:\Program Files\IAR Systems\Embedded Workbench 9.0\arm。

图 1-12 解压后的文件



替换下图 1-13 中红框选中的“config”文件夹。

图 1-13 IAR Pack 安装路径



选择“替换目标中的文件”。

图 1-14 替换或跳过文件选项



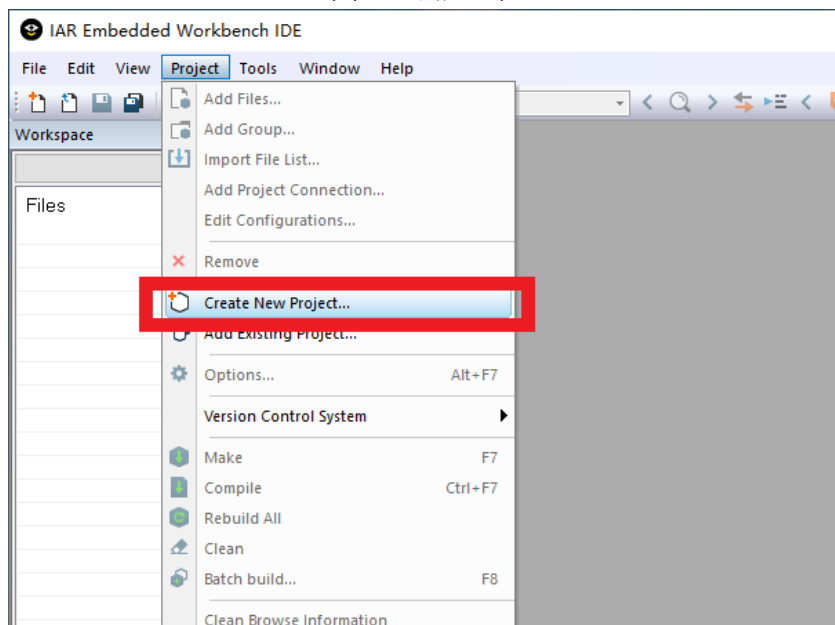


## 2 软件使用

### 2.1 新建工程

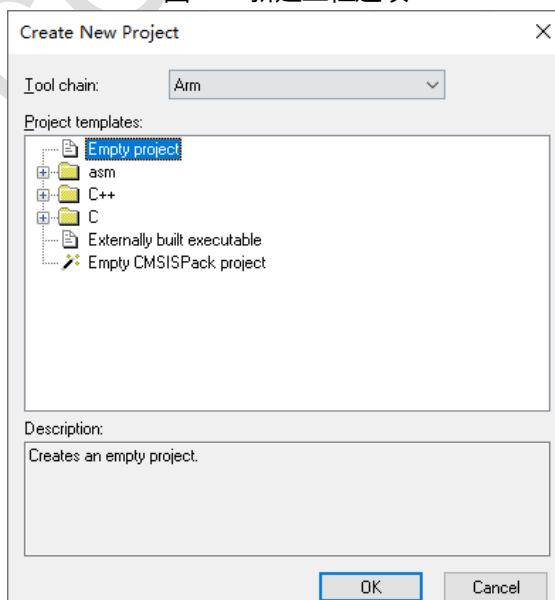
打开 IAR EW for Arm 软件，依次点击菜单栏“Project”——>“Create New Project”。

图 2-1 新建工程



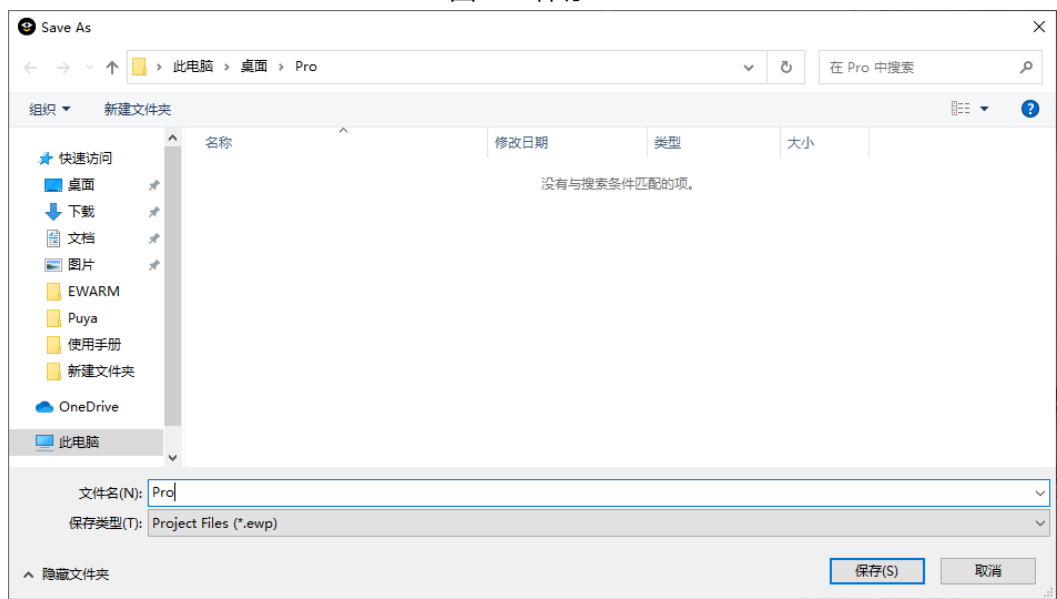
弹出新的窗口，选择“Empty project”（若选择 C/C++ 则会在新的工程中带有 main.c），然后点击“OK”。

图 2-2 新建工程选项



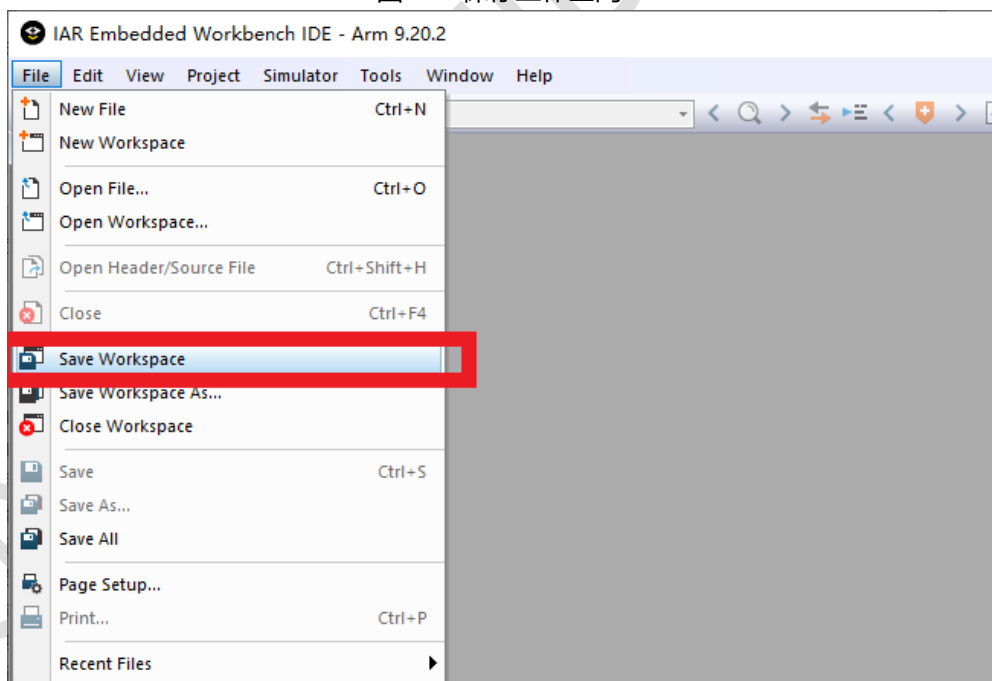
接下来弹出保存窗口，选择合适的路径进行保存。

图 2-3 保存至



然后点击 “File” -> “Save Workspace”，将工作空间进行保存，

图 2-4 保存工作空间



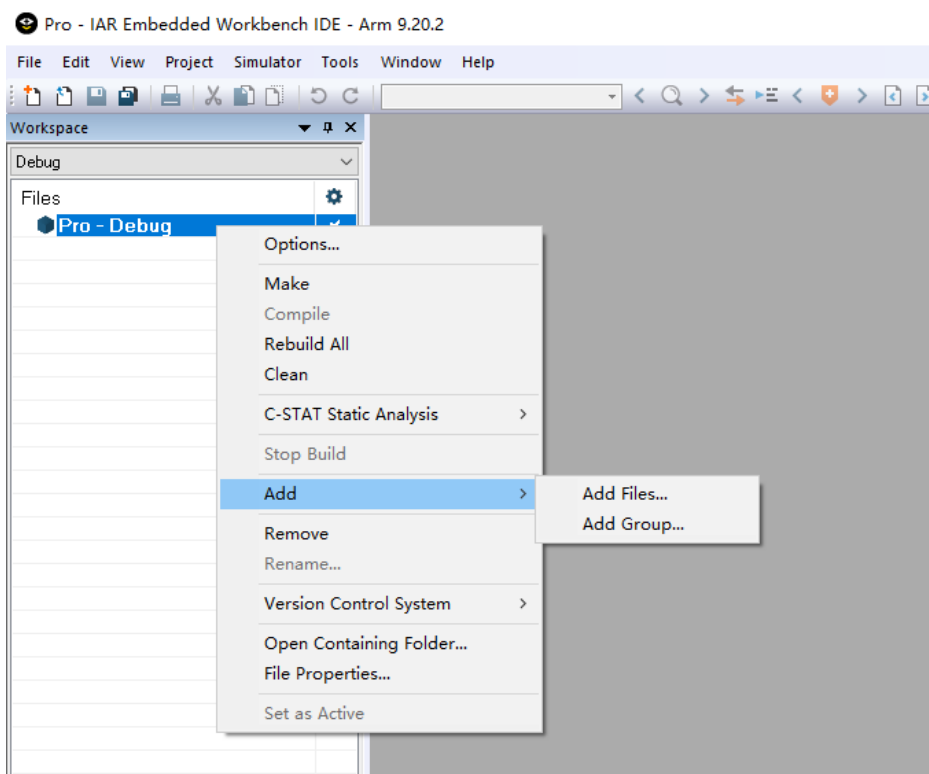
在工作空间保存后，工程文件夹中会有一个 工程名称.eww 的文件，之后可直接点击该文件打开 EWARM 工程。

图 2-5 工程图标



在以上步骤完成后，软件中将会有一个工程文件，右键单击工程文件后，选择“Add”可以添加文件。（Add File 为添加文件，Add Group 为建立文件夹）

图 2-6 添加组



点击 Add Group，并在新弹出的窗口中输入文件夹的名称，建立下图中的文件夹。其中 Output 文件夹不用建立，编译时会自动生成。

图 2-7 输入组名称

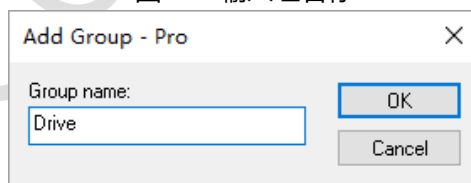
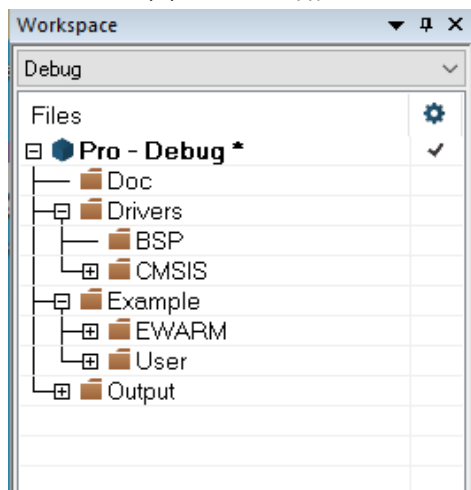
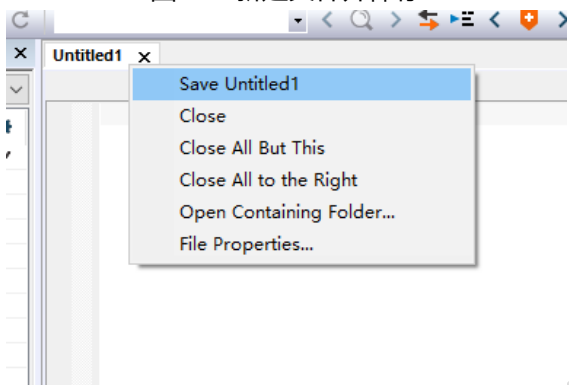


图 2-8 目录结构



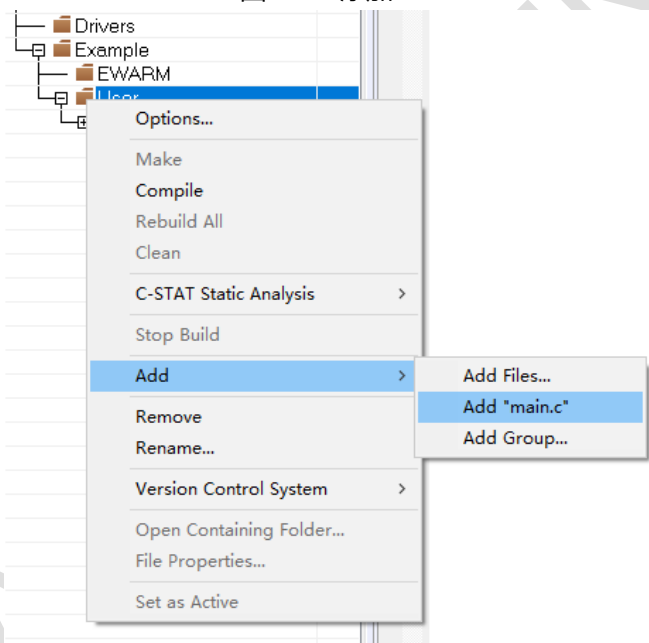
单击菜单栏 File->New File，右键单击新建文件的标题，将其命名为 main.c 并保存。

图 2-9 新建文件并保存



在保存完成后，右键单击 User 文件夹，Add->Add "main.c"，将刚保存的文件添加进来，若已有 main.c 可点击 Add->Add Files 直接添加。

图 2-10 添加 "main.c"



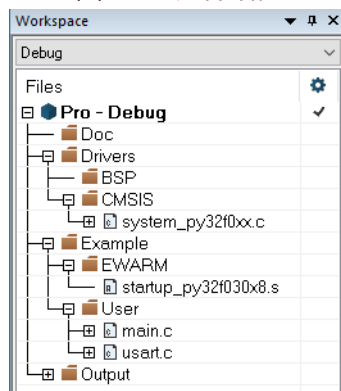
在新建工程时，需要相应的文件，这些文件应当复制一份到工程文件夹中。

图 2-11 所需文件

main.c	2022/6/29 14:31	C Source	1 KB
Pro.dep	2022/6/29 14:34	DEP 文件	5 KB
Pro.ewd	2022/6/29 14:01	EWD 文件	108 KB
Pro.ewp	2022/6/29 14:33	EWP 文件	74 KB
Pro.ewt	2022/6/29 14:33	EWT 文件	208 KB
	2022/6/24 18:15	IAR IDE Worksp...	1 KB
py32f0xx.h	2022/2/9 14:48	C/C++ Header	7 KB
py32f030x8.h	2022/3/16 11:10	C/C++ Header	366 KB
startup_py32f030x8.s	2022/3/7 14:42	Assembler Source	11 KB
system_py32f0xx.c	2022/2/9 14:50	C Source	5 KB
system_py32f0xx.h	2022/2/9 14:49	C/C++ Header	4 KB
usart.c	2022/6/29 14:19	C Source	3 KB
usart.h	2022/6/29 11:27	C/C++ Header	2 KB

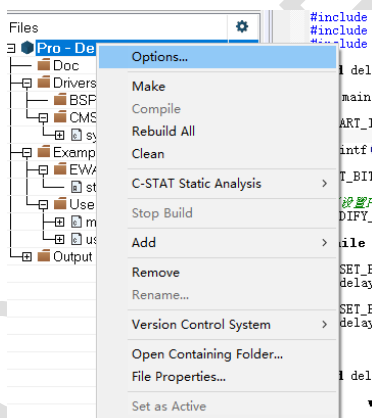
其中 “startup\_py32f030x8.s” 文件应添加到 Example->EWARM 文件夹下 (注意: EWARM 与 MDK 的启动文件有所区别, 在下节移植工程中有进行说明), “syste\_py32f0xx.c” 文件应添加至 Drivers->CMSIS 文件夹下, 文件结构建好后如下图。

图 2-12 文件结构



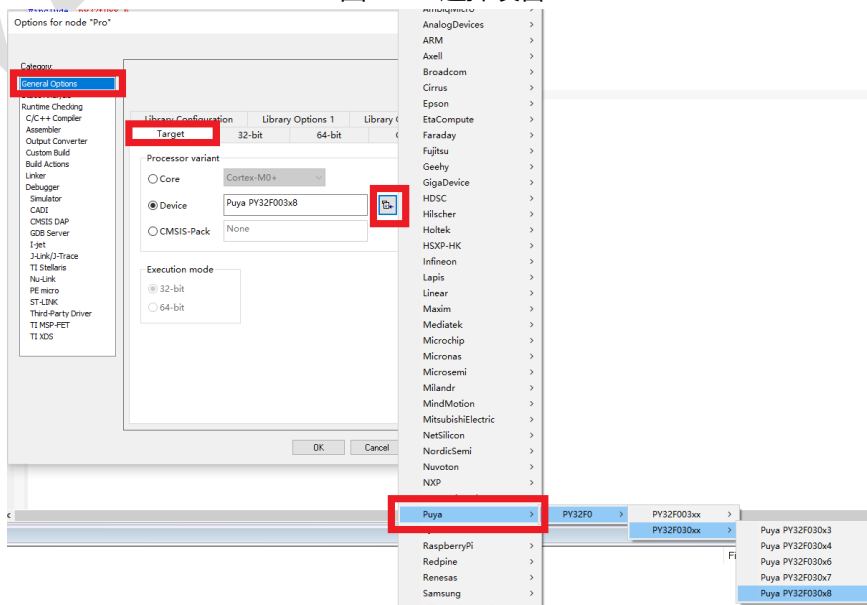
右键单击工程选择 “Options” 可以对其进行设置。

图 2-13 工程设置



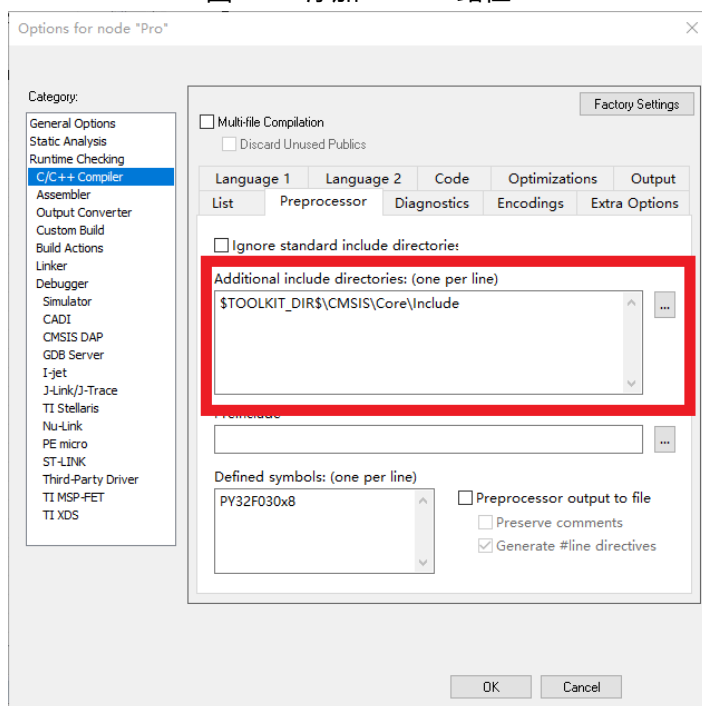
设置中 “General Options->Target->Device” 可选择对应的设备。若列表中无 Puya 选项, 请参考前文的 Pack 包安装过程进行安装或重新安装。

图 2-14 选择设备



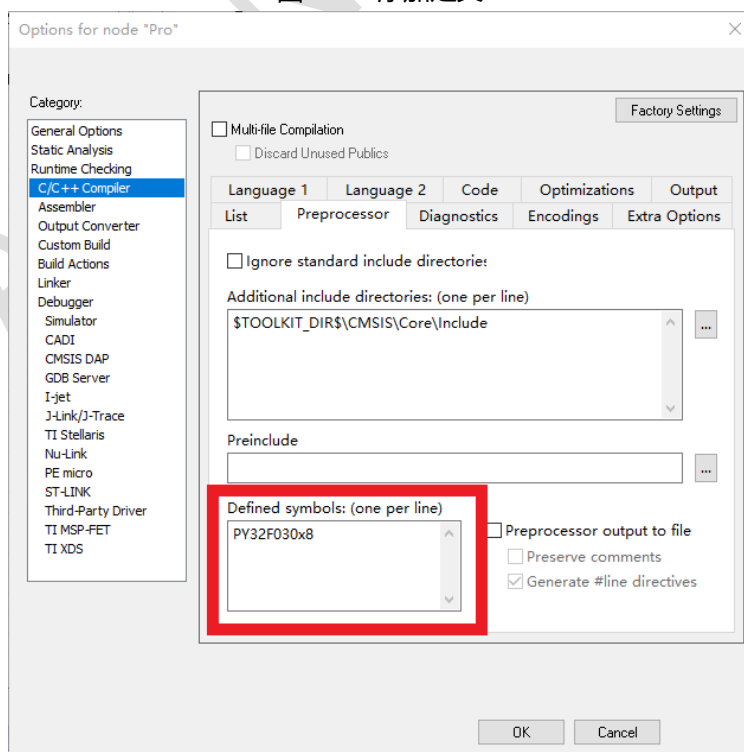
设置中 “C/C++ Compiler>Preprocessor->Additional include directories:(one per line)” 可以添加 include 路径, “\$TOOLKIT\_DIR\$\CMSIS\Core\Include” 为安装目录下的 core include。

图 2-15 添加 include 路径



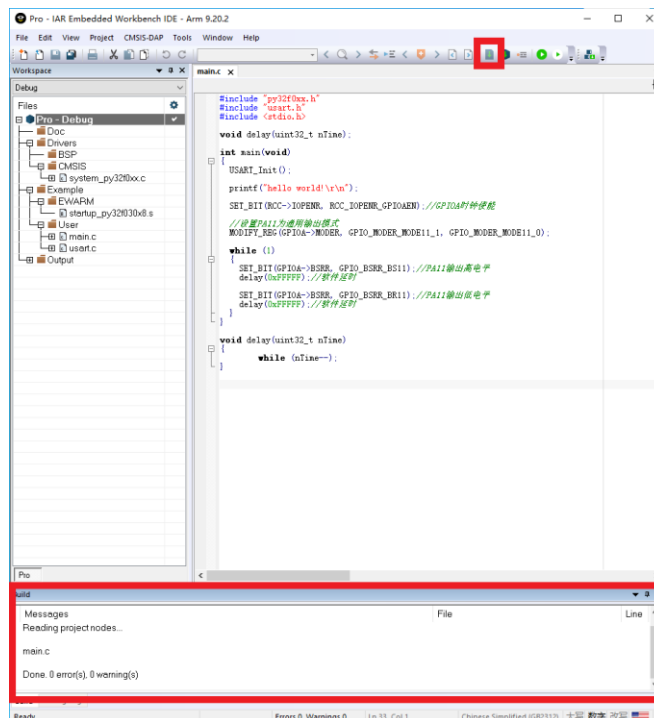
设置中 “C/C++ Compiler>Preprocessor->Defined symbols:(one per line)” 可以添加定义。

图 2-16 添加定义



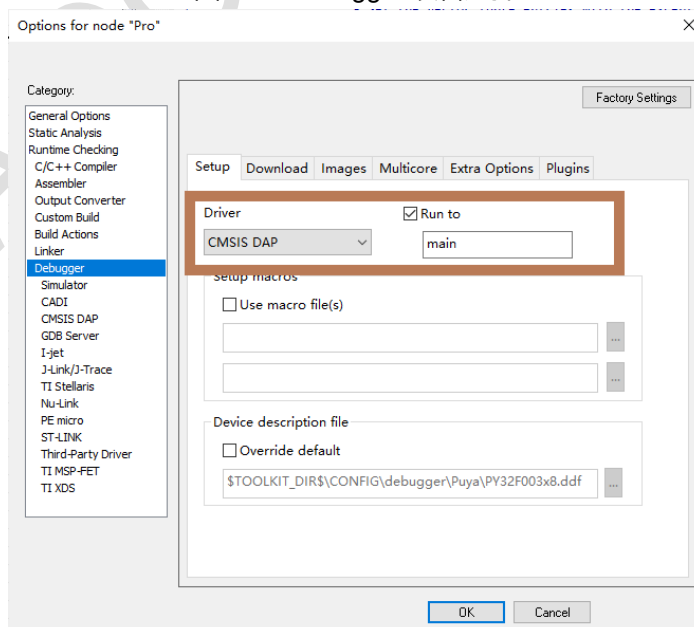
将所有文件、路径、定义设置完成后可以进行编译，点击红框中的按钮进行编译。消息窗口提示 Done. 0 error(s), 0 warning(s)

图 2-17 编译



接下来要将程序下载到 MCU 中进行测试，将 PY32F030xx8 通过 SWD 与 PY-LINK 相连，并将 PA9 引脚与 UART 的 RX 引脚相连，将 PY-LINK 仿真器与电脑相连。在工程设置中 “Debugger>Setup->Driver” 中选择 “CMSIS DAP” 并勾选右侧的 “Run to main”。

图 2-18 Debugger 设备选择



在使用 PY-Link 连接时，需在设置中“CMSIS DAP ->Interface-> Interface” 设置为 1MHz 才能下载，否则可能弹出错误窗口。

图 2-19 接口频率选择

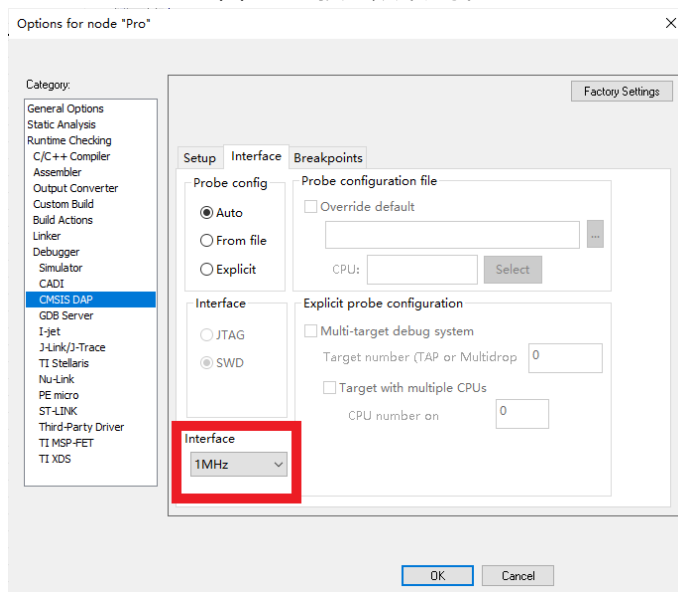
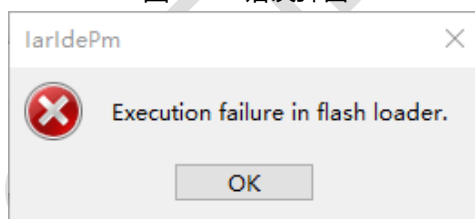
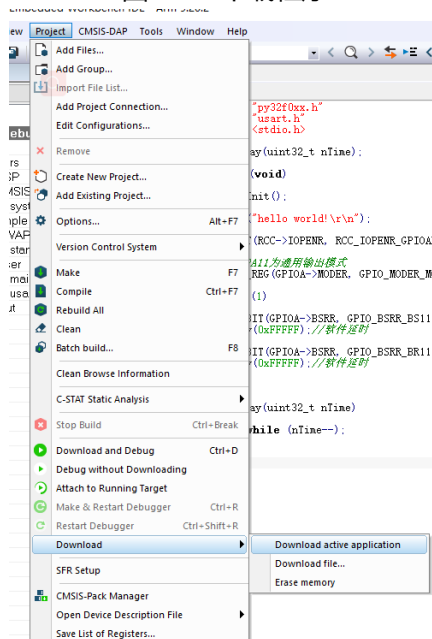


图 2-20 错误弹窗



设置好后点击“Project->Download-> Download active application”进行下载。(若经常使用此功能，请参考 2.3.3 章节，将该功能置于工具栏。)

图 2-21 下载程序





下载成功后，在电脑上打开串口工具，由于该例程在文件中规定了校验方式，波特率等信息，故需在工具中选择与文件相同的选项。

图 2-22 串口接收选项



按下开发板的 Reset 按钮，串口工具显示如下图。

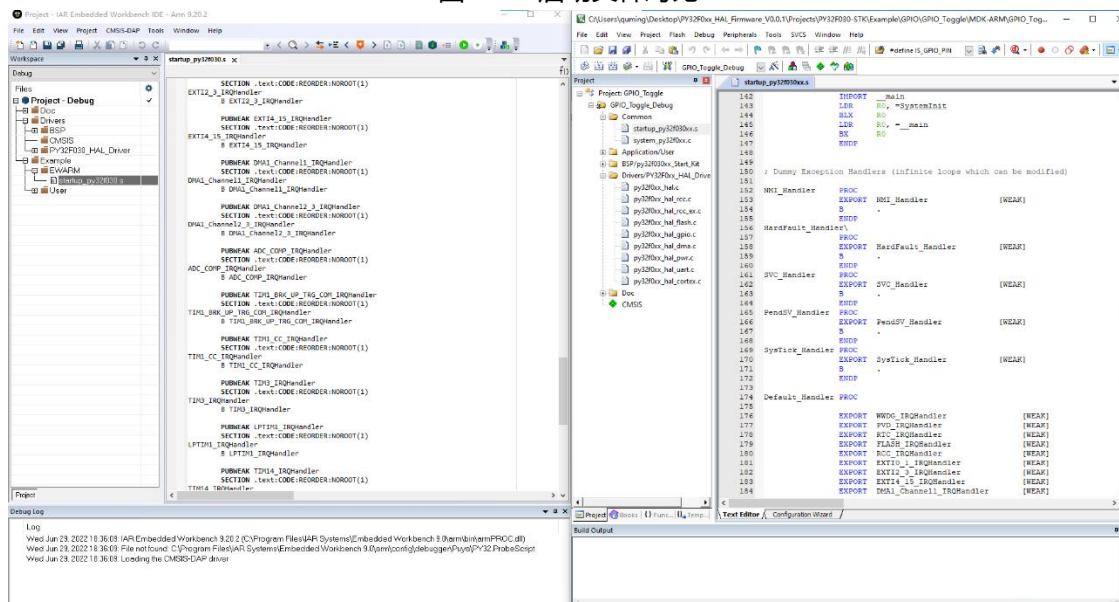
图 2-23 串口输出

hello world!

## 2.2 移植工程

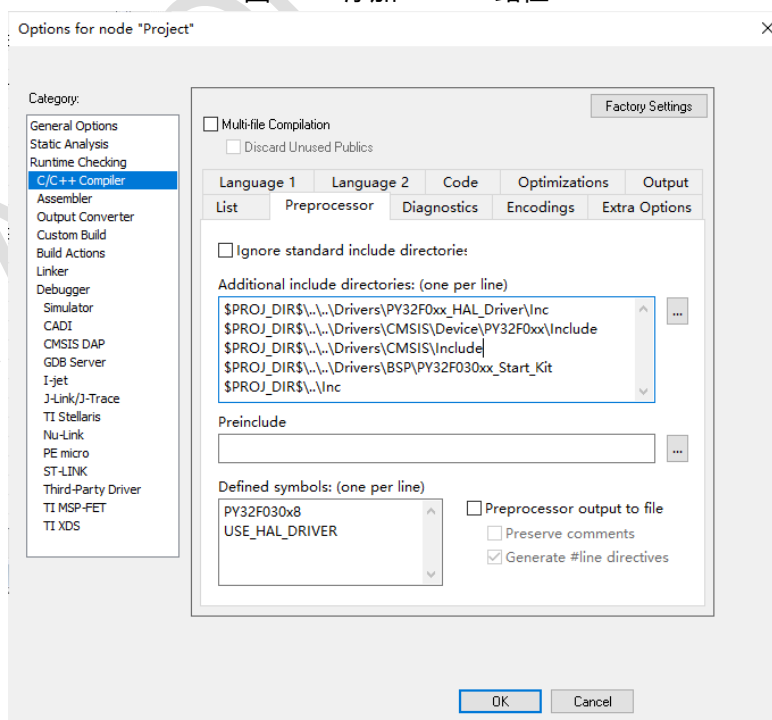
本节简要介绍了从 MDK 移植工程，其中大部分操作同上一节的新建工程一样，需将文件添加至工程中，但其中启动文件的写法有所不同，特别是中断的写法，移植时应当修改。

图 2-24 启动文件对比



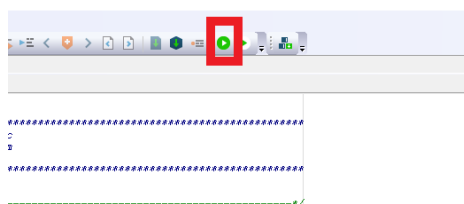
在工程设置中“C/C++ Compiler>Preprocessor->Additional include directories:(one per line)”可以添加 include 路径，如：“\$PROJ\_DIR\$\\Drivers\CMSIS\Include”。

图 2-25 添加 include 路径



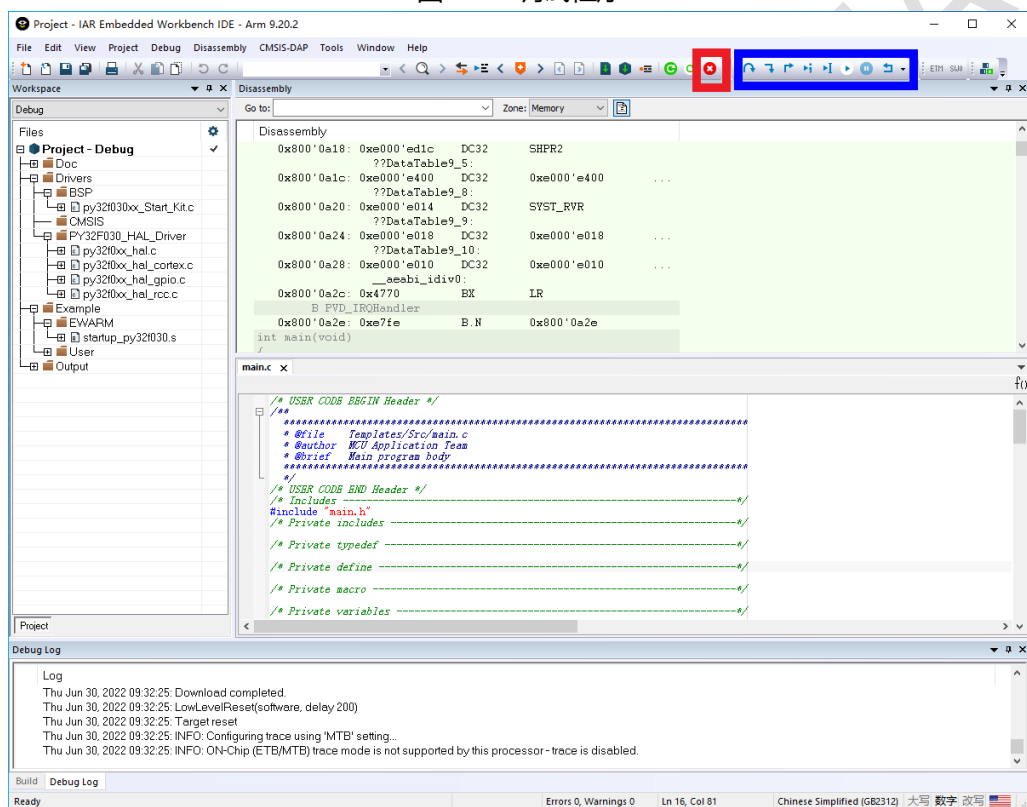
在编译成功后可进入调试模式，点击图中红框中的按钮即可进行调试。

图 2-26 调试按钮



调试时界面如下图，红框中的按钮可以结束仿真，蓝框中的按钮可以控制仿真的步骤（单步，全速，Reset等）。

图 2-29 调试程序

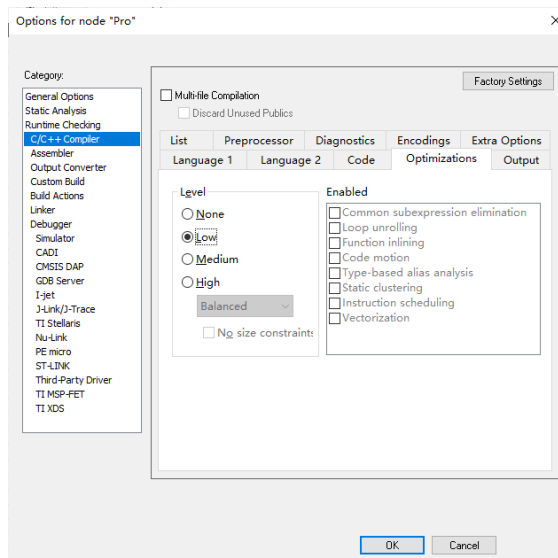


## 2.3 常用功能

### 2.3.1 编译器优化

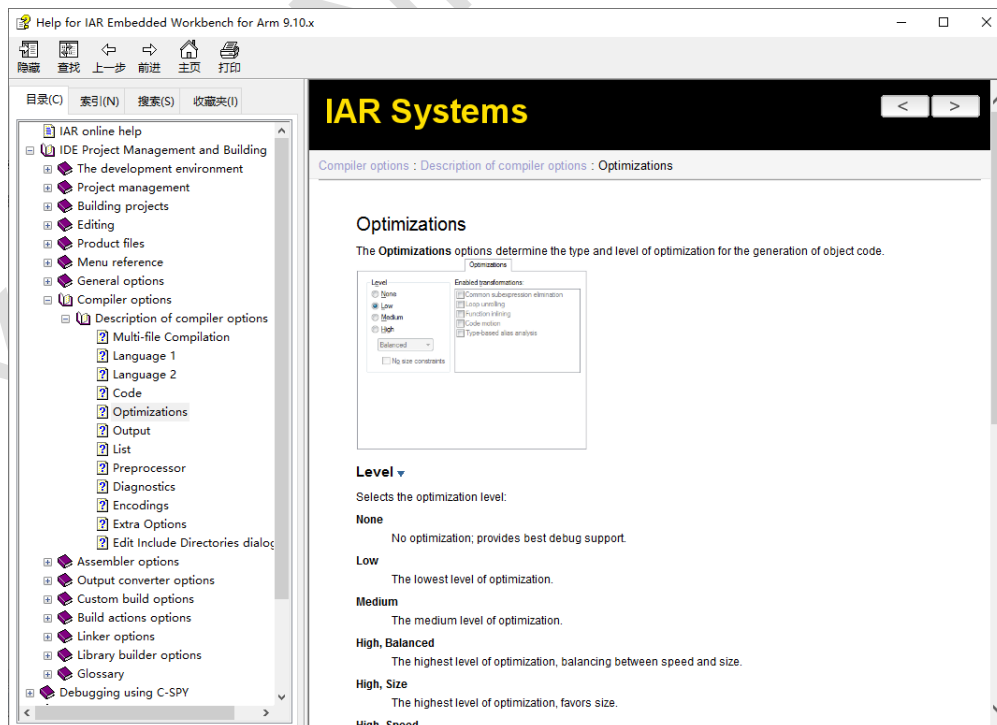
工程设置中“C/C++ Compiler>Optimizations->Level”可以调整编译器的优化等级。

图 2-30 编译器优化



具体请参考 IAR online help 中的说明，可在菜单栏 “Help->Content” 中，按下图所示路径查看。

图 2-31 在线帮助文档

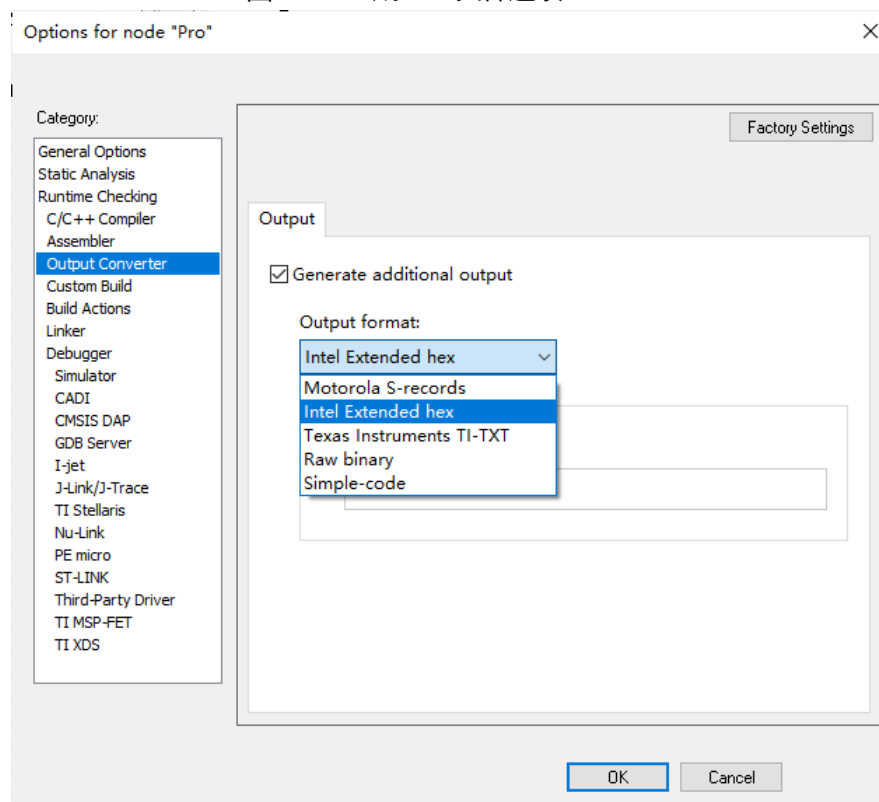


### 2.3.2 生成 hex 文件

在工程设置中 “Output Converter> Output” 可以将 “Generate additional output” 勾选，并在下面的 “Output format” 选择 “Intel Extended hex” 来在编译时生成 “.hex” 文件，

生成的该文件位于工程文件夹中的“\Debug\Exe”路径下。

图 2-32 生成 hex 文件选项



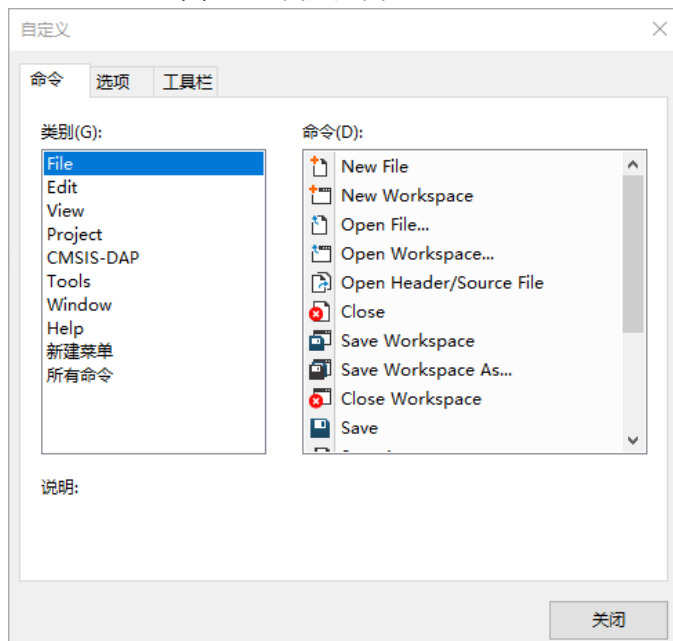
### 2.3.3 自定义工具栏

点击工具栏最右侧的小三角，如图所示，依次点击“添加或删除按钮->Customize”，弹出自定义窗口，如下图所示。

图 2-33 自定义工具栏

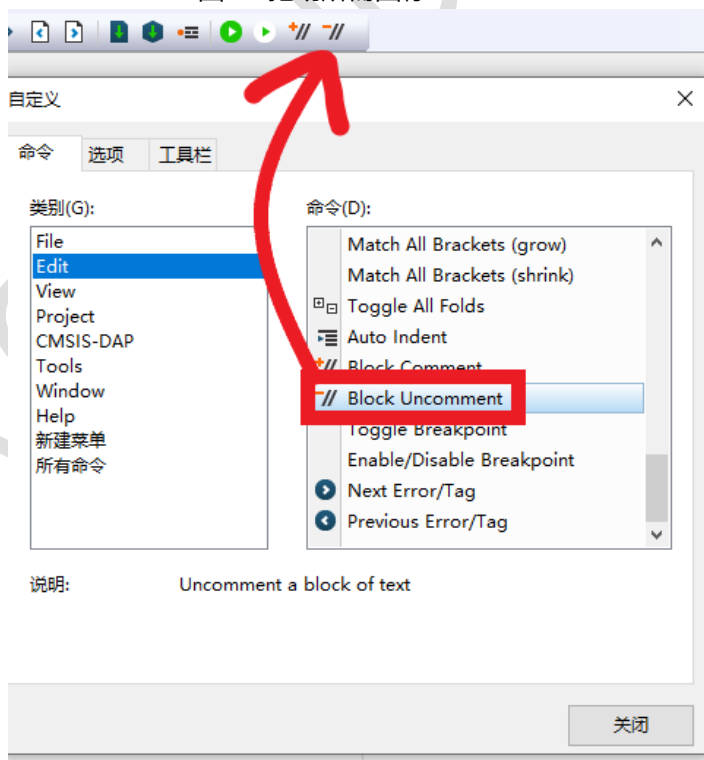


图 2-34 自定义窗口



找到需要使用的图标，鼠标按住将其拖入工具栏可使该图标固定到工具栏中，如图所示。

图 2- 拖动所需图标



### 3 版本历史

Version	Content	Date
V1.0	Initial Release	2022.08.01



Puya Semiconductor Co., Ltd.

**IMPORTANT NOTICE**

Puya Semiconductor reserves the right to make changes without further notice to any products or specifications herein. Puya Semiconductor does not assume any responsibility for use of any its products for any particular purpose, nor does Puya Semiconductor assume any liability arising out of the application or use of any its products or circuits. Puya Semiconductor does not convey any license under its patent rights or other rights nor the rights of others.