

在HighTec开发环境下的程序仿真调试介绍

——基于DAP miniWiggler（仿真器）和PLS UDE（Debugger）



1.DAP miniWiggler介绍

2.在HighTec IDE下用DAP和PLS进行仿真调试

DAP miniWiggler简介

■ 概述

- 今天给大家介绍的就是适用于英飞凌多款单片机的仿真器:**DAP miniWiggler**
- 适用于keil, Tasking, HighTec, Aurix Development Studio
- 支持XC800,XC166,XE166,XC2000,Tricore系列
- 驱动软件: **DAS**

■ DAS简介

- DAS是用来连接Infineon芯片的工具
- DAS支持支持以下设备
 - XC800
 - XC166, XE166/XC2000
 - TriCore™, AURIX™
 - XMC4000, XMC1000
- **注:**如果用到以上芯片, 必须安装DAS, 否则无法下载程序到单片机!!!

【乾勤】 DAP miniwiggler 提供技术支持
具有十年以上英飞凌单片机开发和应用经验

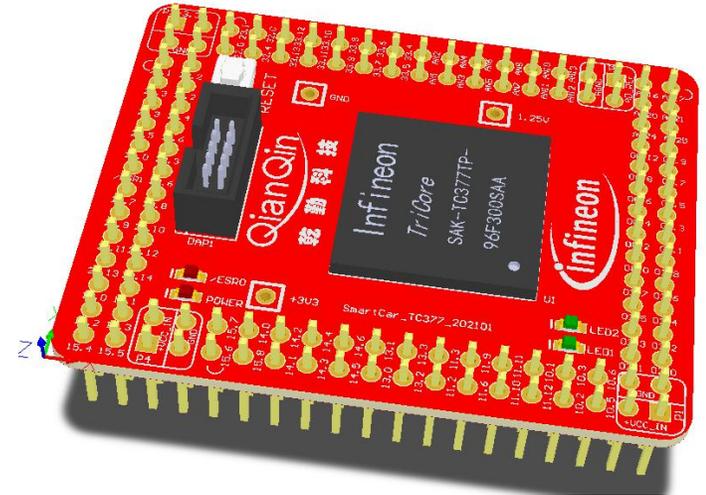
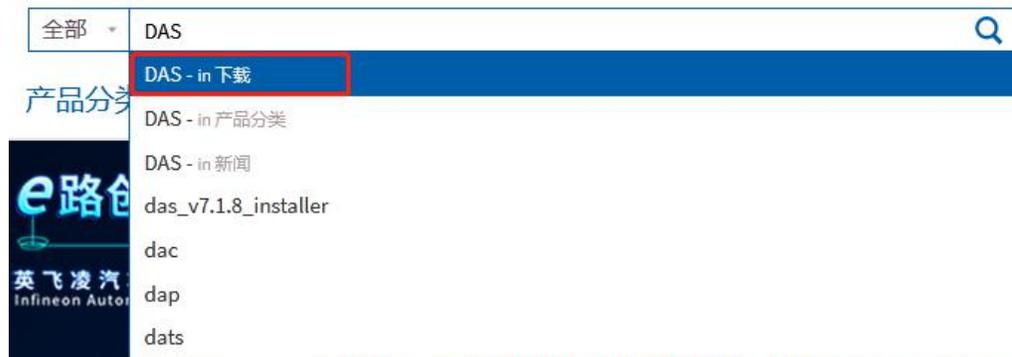


支持Tasking Compiler, Keil uvision, Hitop Hitex等多种主流软件开发程序
支持 XC800 XC166 XE166 XC2000 Tricore Aurix系列芯片

DAP miniWiggler获取

■ DAS获取途径

- 方法①：官网链接：www.infineon.com/DAS
- 方法②：
 - 搜索英飞凌官网<https://www.infineon.com/cms/cn/>
 - 在搜索栏输入DAS即可



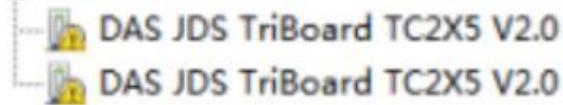
- 方法③：加入乾勤qq群->**1042381622**（2020支持群）或**930347957**（2021支持群）

DAP miniWiggler安装使用

■ DAS驱动安装方法

- 在使用DAS之前，需要安装DAS驱动
- 安装方法如下：

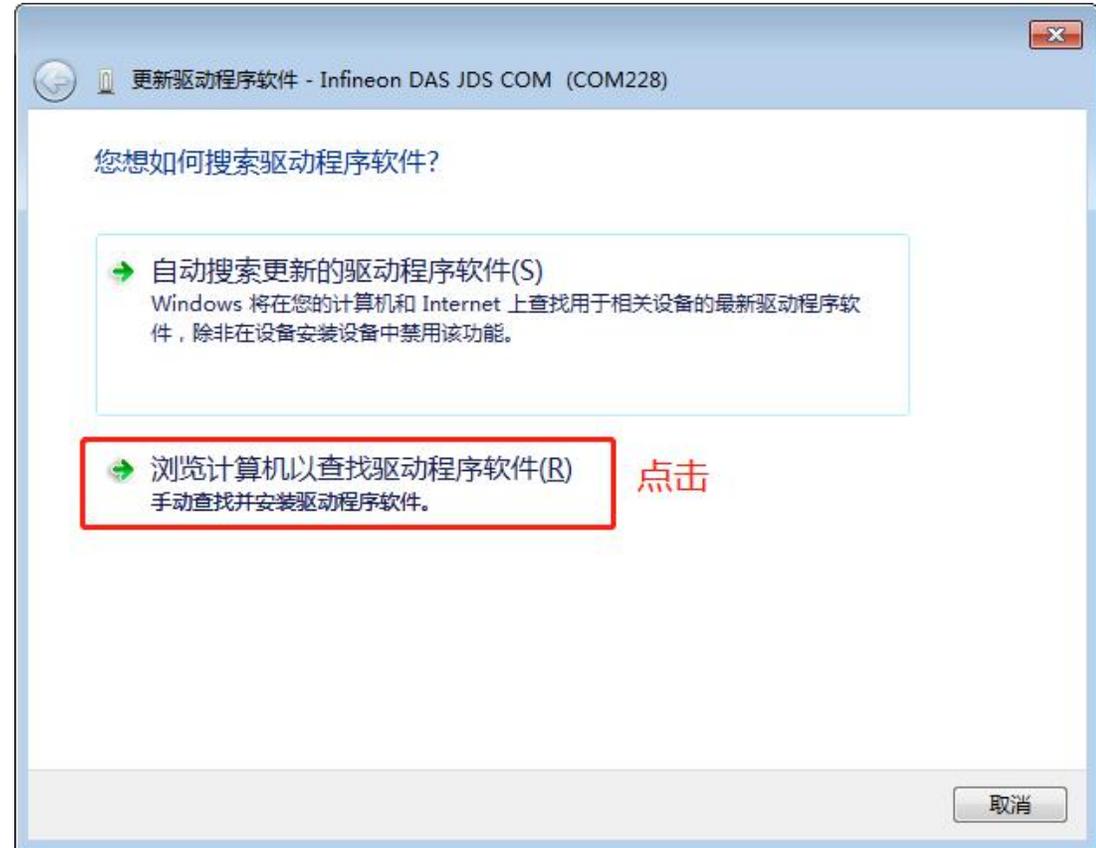
- 1.将DAP miniWiggler仿真器连接电脑，打开设备管理器
- 2.若出现以下情况，则代表未安装DAS驱动



- 3.然后按照图示步骤操作

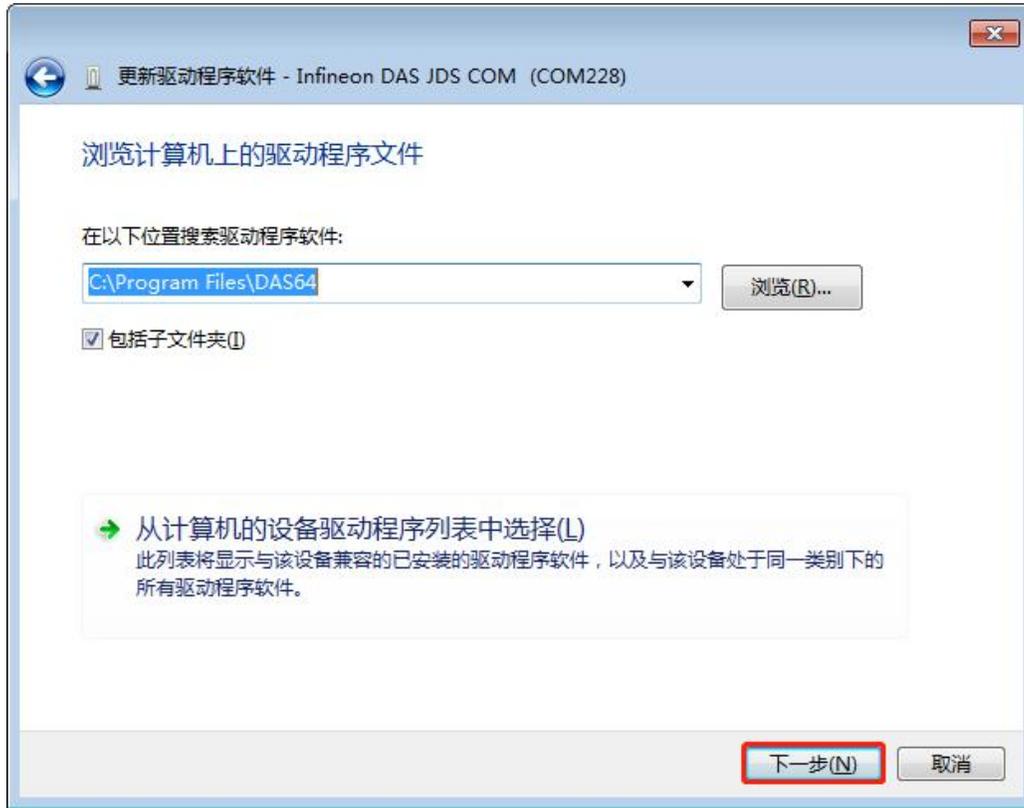


- 4.点击浏览计算机以查找驱动程序软件。



DAP miniWiggler安装使用

5. 点击浏览，找到DAS安装路径，一般在C:\Program Files\DAS64，定位到DAS64文件夹即可。



英飞凌汽车电子生态圈

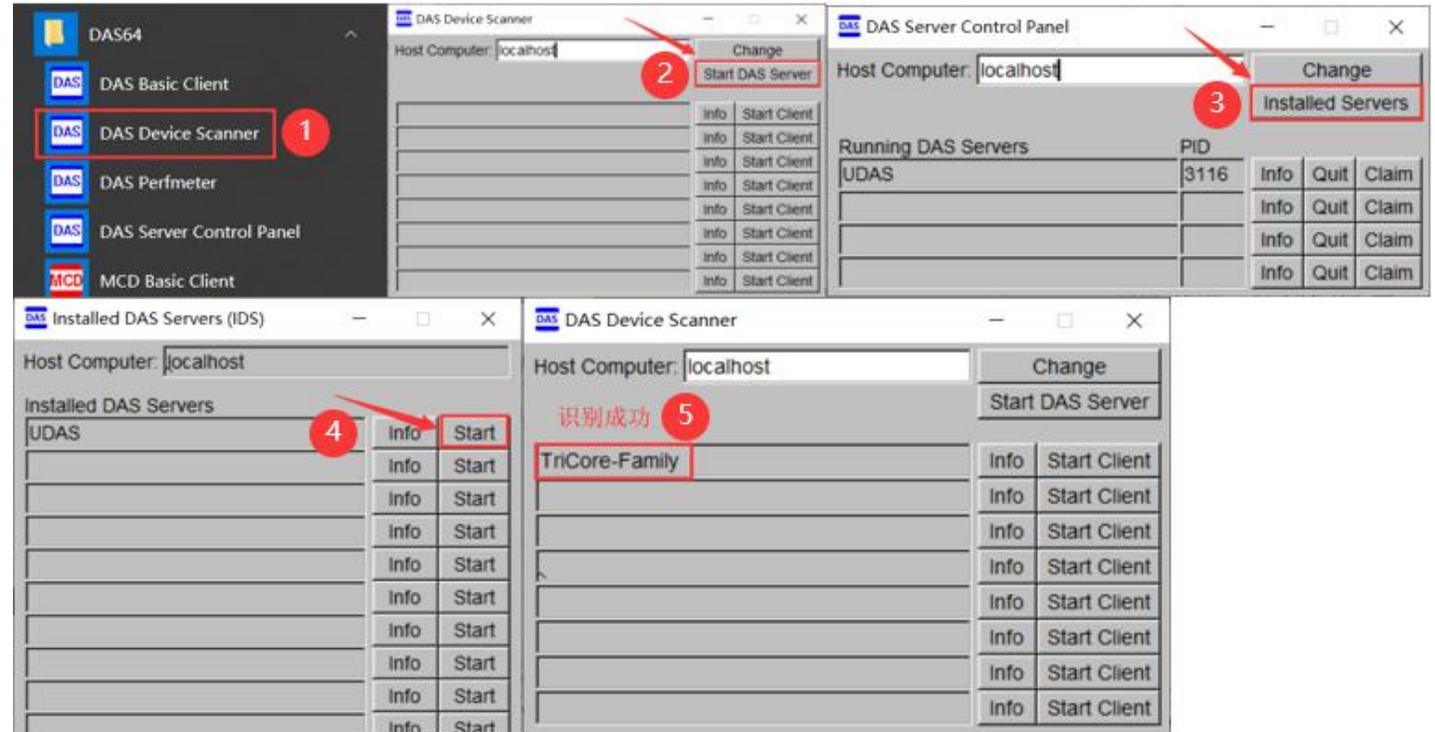
DAP miniWiggler安装使用

■ DAS使用方法

- 以TC264核心板为例，在核心板供电的情况下，将仿真器一端接核心板的下载口，另一端接电脑，如图所示：



- 打开DAS，按照下图步骤操作，若出现**Tricore-Family**，即为识别成功。



英飞凌汽车电子生态圈

1. DAP miniWiggler介绍

2. 在HighTec IDE下用DAP和PLS进行仿真调试

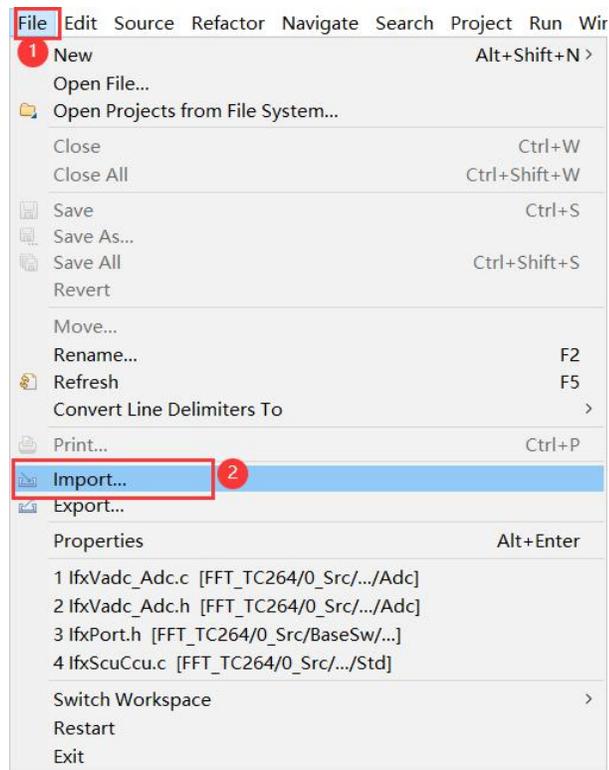
■ HighTec简介

- HighTec 的 TriCore™ 开发平台是基于 Eclipse™ 的集成开发环境，集成了GNU 编译、PLS UDE 调试工具,并与Hightec界面兼容性好，使用Hightec开发环境能实现编译下载仿真调试全过程工作
- HighTec编译器有免费license（1年）和商用License，[在此pdf里](#)，用到的是免费版本的HighTec
- HighTec4.9.3安装包链接：链接：<https://pan.baidu.com/s/1McNofW3uN6osOOgJV8cNKQ>（提取码：uk00）
- HighTec使用说明及License申请方法，可参照乾勤科技写的HighTec使用说明.pdf，链接：<https://pan.baidu.com/s/1U5smvnewcEY1vc9Z5CzZ3g>（提取码：flsi）

■ 实验介绍

- 目的：用DAP miniWiggler在HighTec下，进行下载、调试
- 准备：HighTec4.9.3， DAP miniWiggler， HighTec程序
- 例程：本文使用QianQin_HighTec_TC275_ADC例程为例
此例程可从qq群文件获取

- 启动HighTec软件，并按照以下流程导入QianQin_HighTec_TC275_ADC例程。

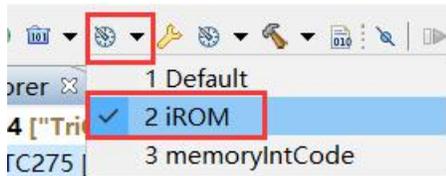


HighTec仿真调试介绍

- 选择此工程，并将此工程设为活跃



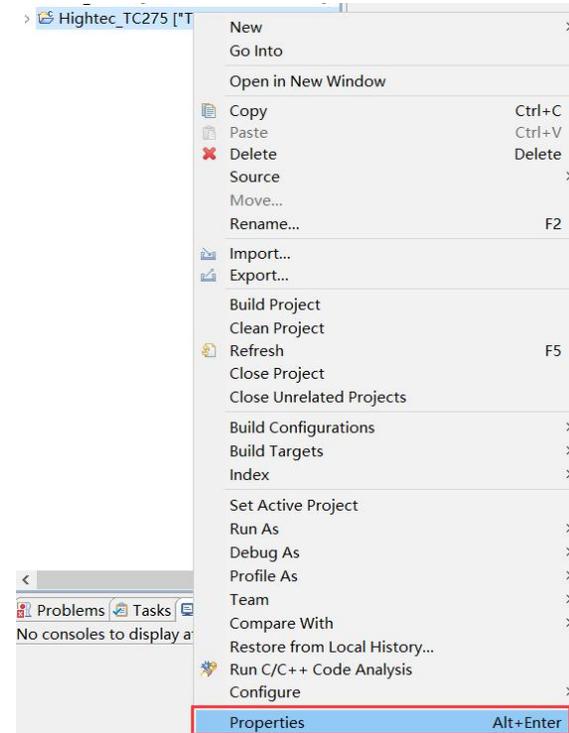
- 点击红色边框所框图标，选择iROM



- 设置链接路径，一种点击红色边框所框图标进入

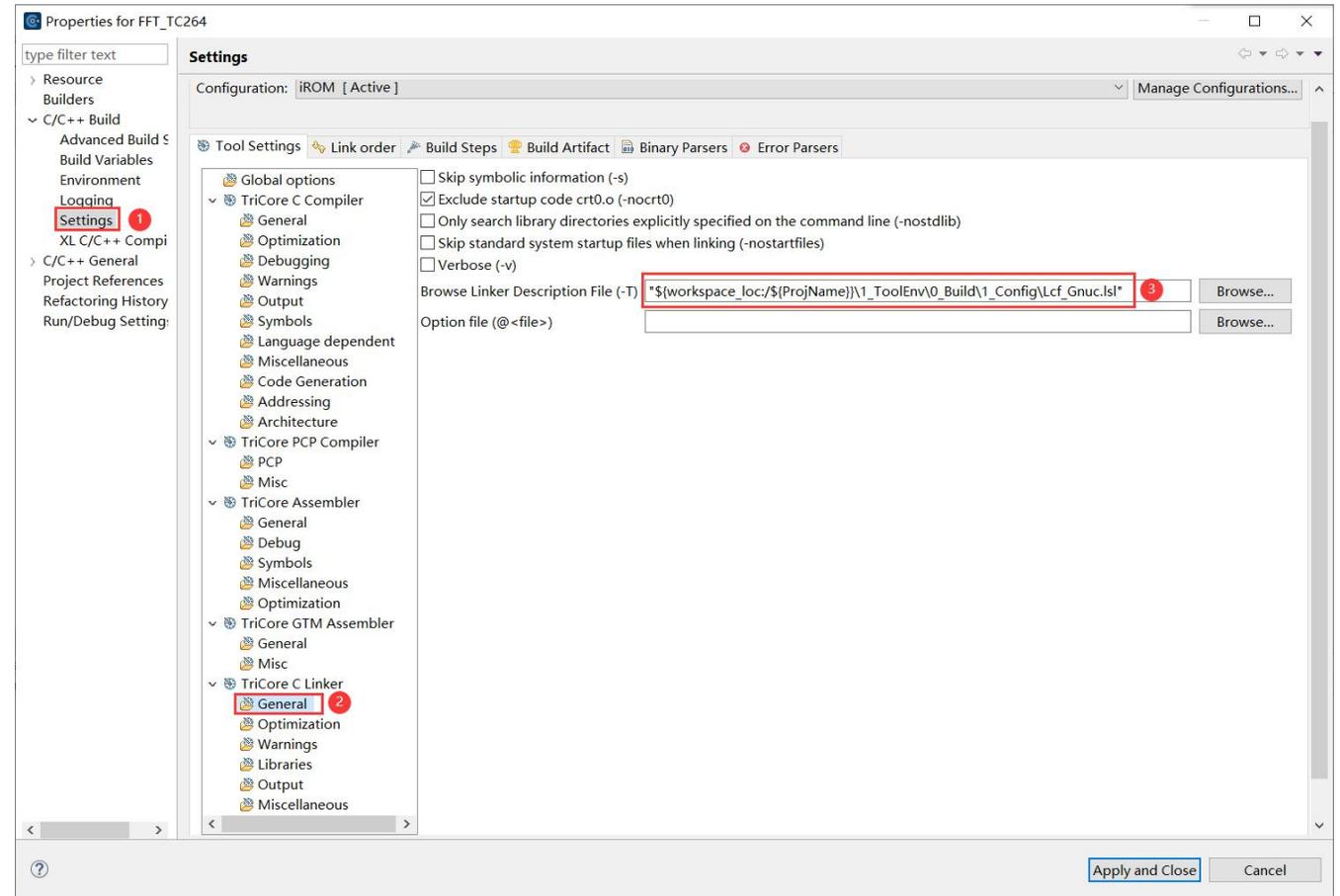


- 设置链接路径，亦可右键单击工程选择properties进入



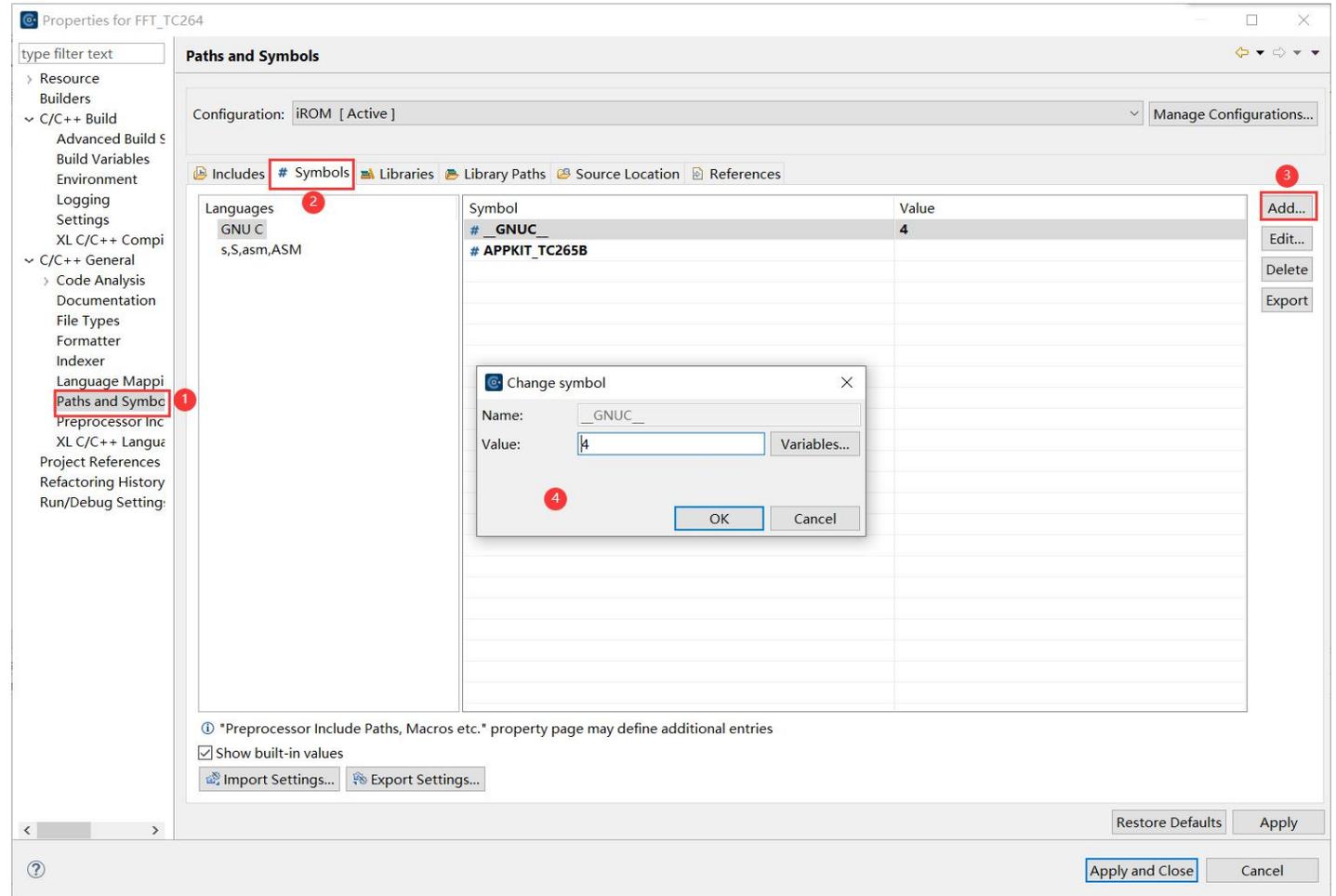
HighTec仿真调试介绍

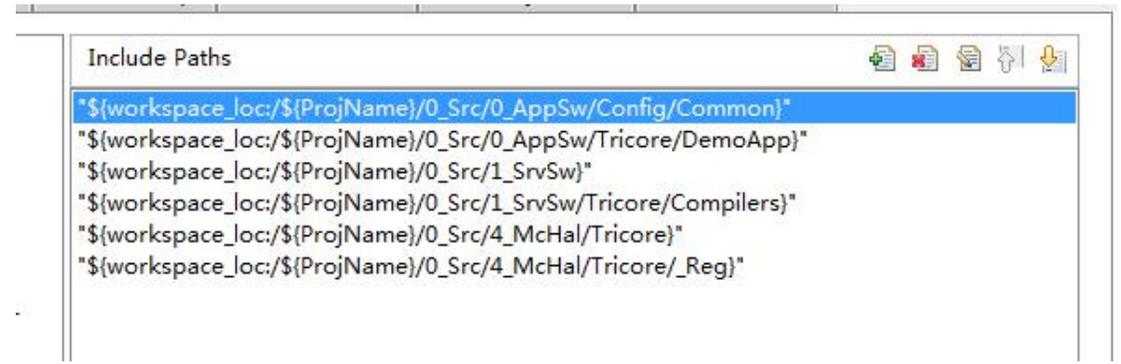
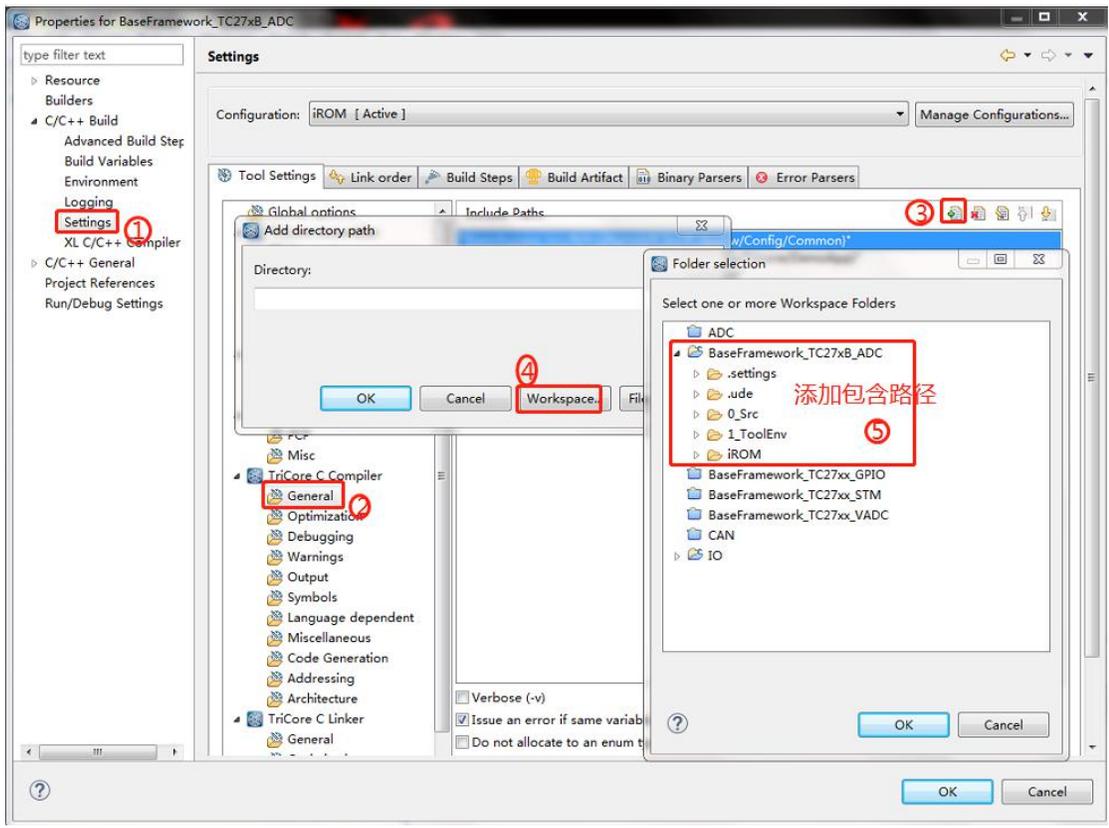
- 如右图所示，依次选择Settings->General,输入工程相对路径“`${workspace_loc}/${ProjName}}\1_ToolEnv\0_Build\1_Config\Lcf_Gnuc.lsl`”，点击Apply and Close。



HighTec仿真调试介绍

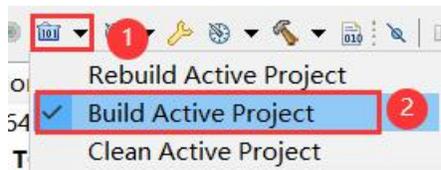
- 设置Symbols，如右图所示依次选择Paths and Symbols->Symbols->Add
- 添加定义：
Name: __GNUC__
Value: 4





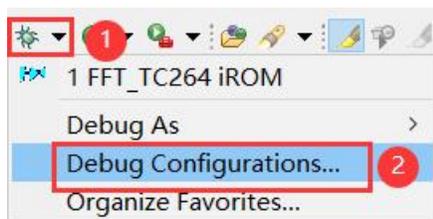
- 设置Include paths,如图所示,依次选择 Settings->General,选择添加,在Workspace中选择文件包含路径,如有其余文件包含路径,配置方法类似。

- 选择Build Active Project, 进行编译工程并生成elf文件, [使用DAS连接TC275](#)

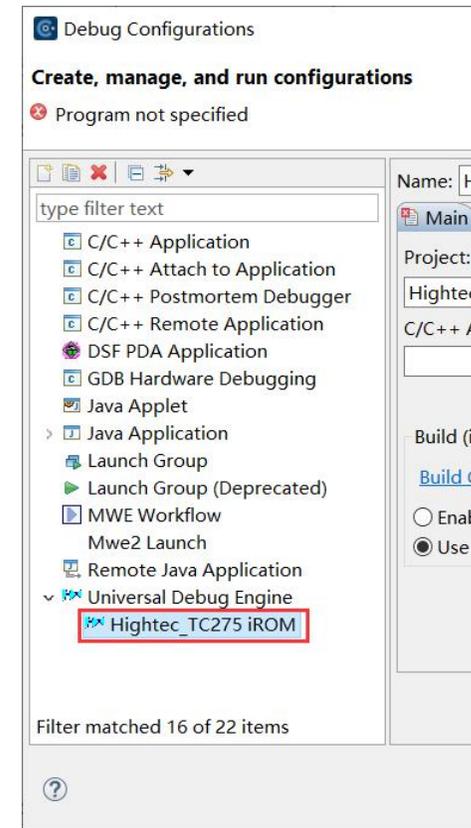
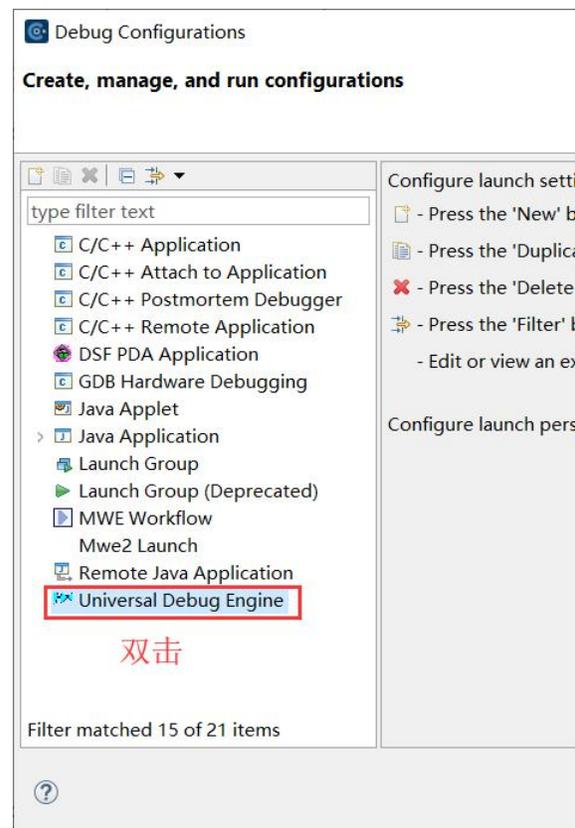


```
'Finished building: ../_Src/_AppSw/Tricore/Main/Cpu2_Main.c'  
'Finished building: ../_Src/_AppSw/Tricore/Main/Cpu0_Main.c'  
.  
.  
'Finished building: ../_Src/_AppSw/Tricore/APP_driver/VadcApp.c'  
'Building target: Hightec_TC275.elf'
```

- 选择Debug Configurations, 进行Debug 配置

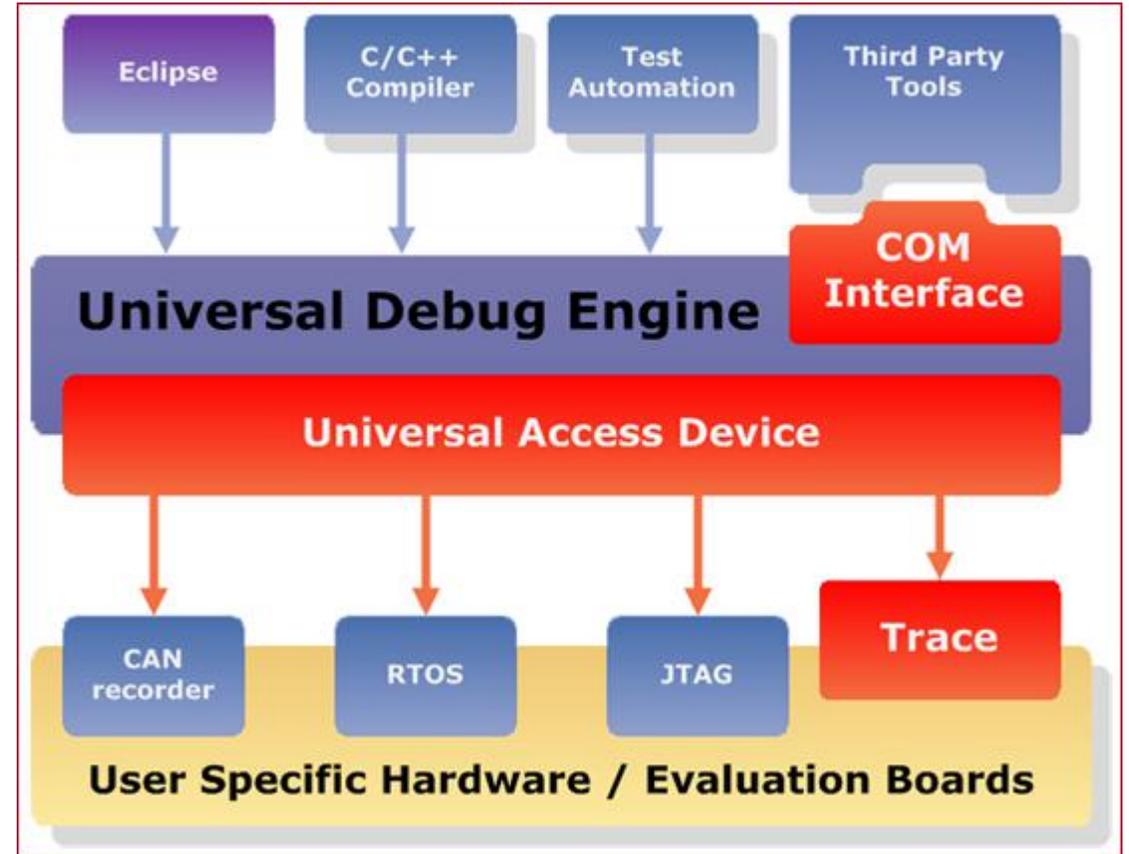


- 双击Universal Debug Engine, 自动生成工程的iROM



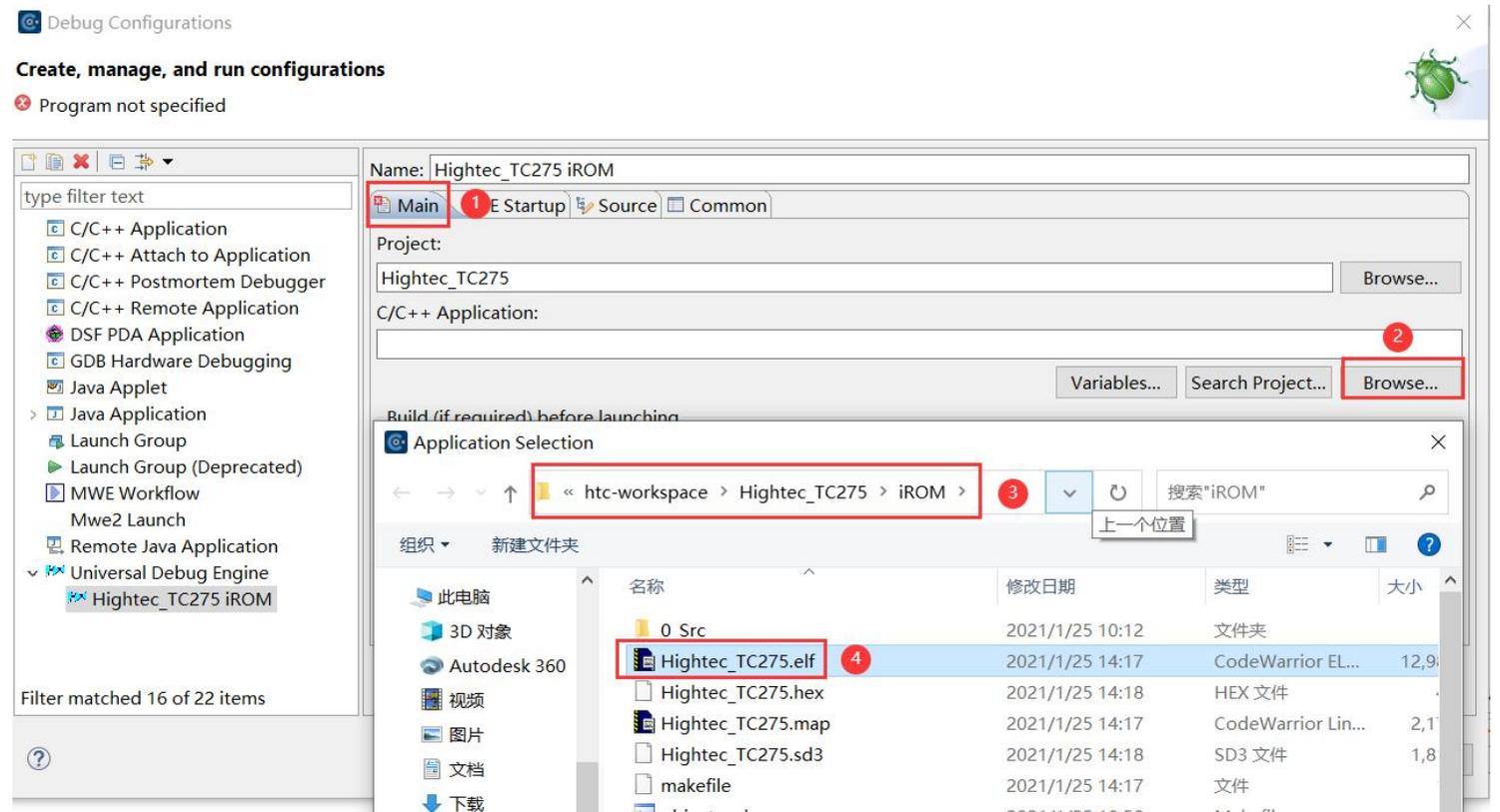
UDE (Universal Debug Engine) 介绍

- UDE (通用调试引擎)** 是用于应用程序开发的最先进的开发工作台。作为软件包的一部分，UDE是功能强大的HLL调试器和仿真器，用于测试以C / C ++语言和/或程序集创建的微控制器应用程序。与通用访问设备，UDE的通信附加组件以及可能使用的目标监视器一起，它可以在客户目标系统上的断点之间实时执行。通过集成工具使用集成开发环境可以节省时间。只有一个图形用户界面可用于开发，调试和仿真。多个高速接口允许非常灵活地访问目标系统。通用调试引擎是一个开放平台，为其他工具提供了不同的接口。

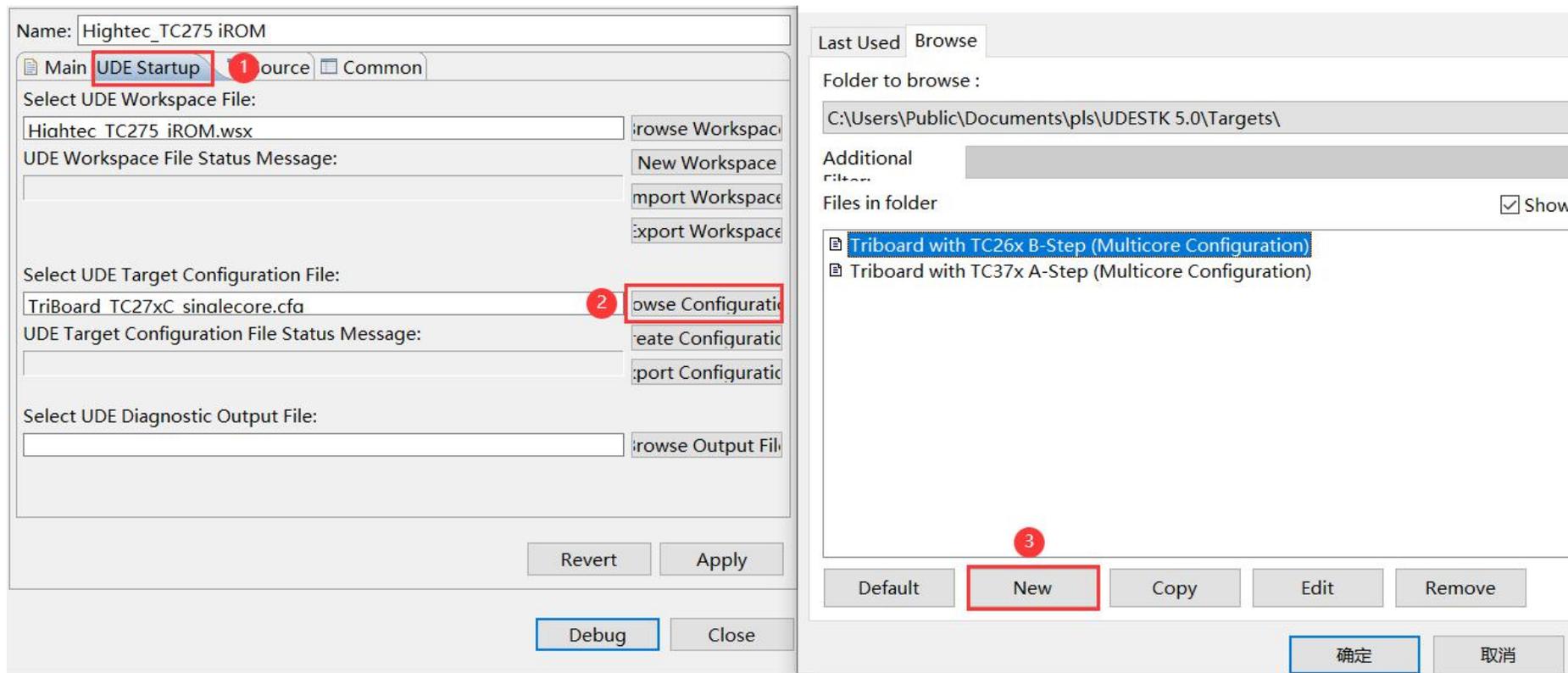


HighTec仿真调试介绍

- 选择Main项目，在C/C++ Application栏下点击Browse，选择载入工程的iROM路径下刚才编译生成的elf文件

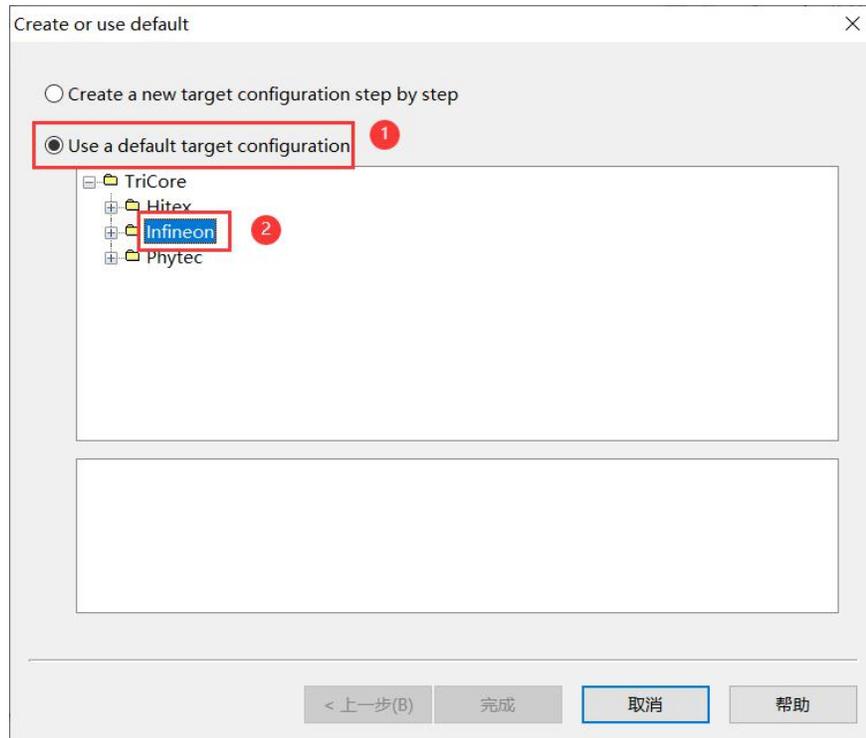


- 选择UDE Startup项目，在Select UDE Target Configuration File栏下点击Browse Configuration，在弹出的窗口中点击New

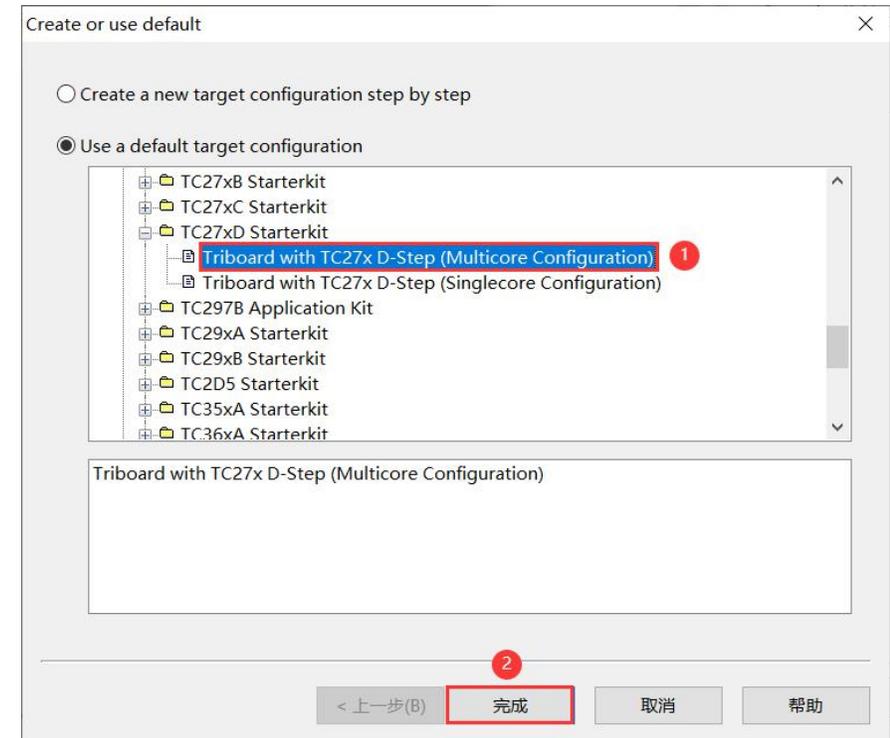


HighTec仿真调试介绍

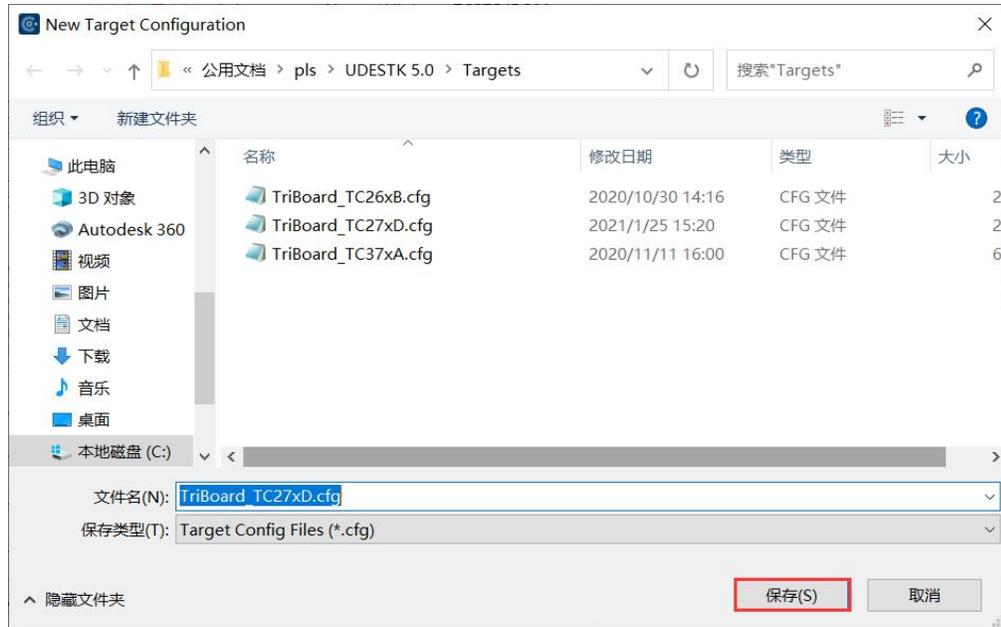
- 在弹出的窗口中选择Use a default target configuration项，展开Infineon



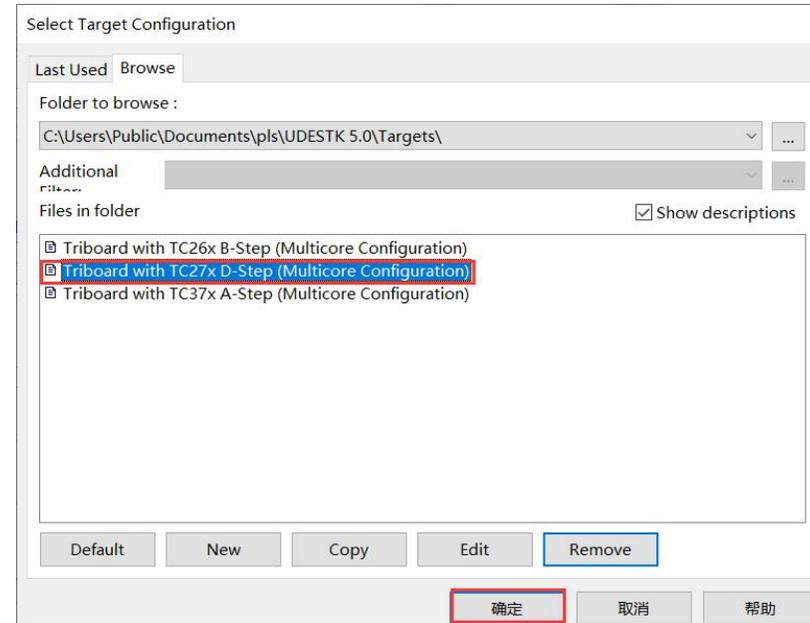
- 如下图所示选择控制板型号，并点击完成



- 在弹出的窗口中选择点击保存

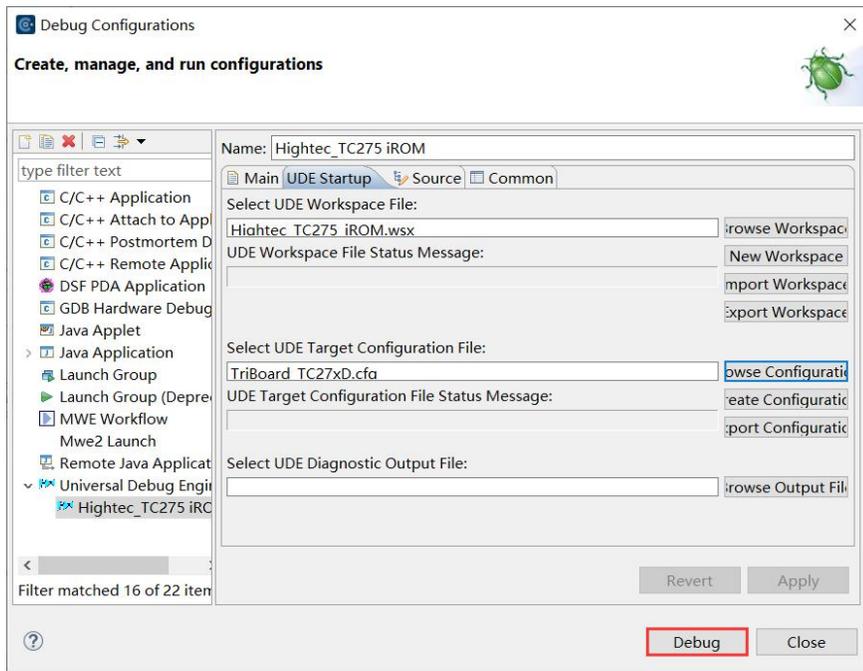


- 在弹出的窗口点击确定

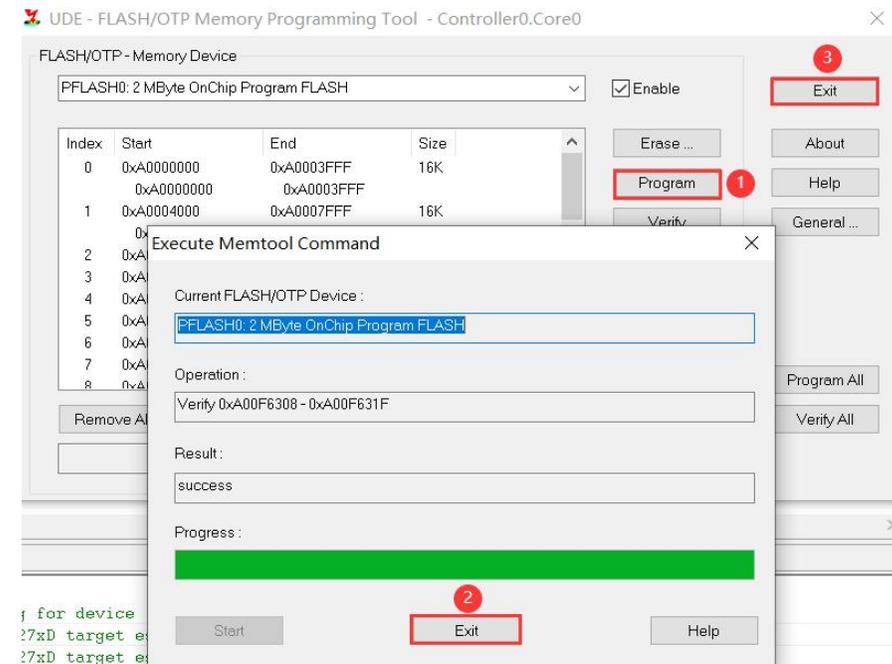


HighTec仿真调试介绍

- 在Debug Configuration窗口中点击Debug



- 进入工程在线调试模式，点击Program下载，下载成功后点击Exit退出



PLS UDE窗口仿真调试介绍

- 在工具栏点击红色边框所框图标，运行程序



- 在工具栏点击红色边框所框图标，跳过子函数，单步执行程序



- 在工具栏点击红色边框所框图标，进入子函数，单步执行程序



- 在工具栏点击红色边框所框图标，跳出当前子函数，单步执行程序，此图标当程序运行至子函数内时，会高亮显示



PLS UDE窗口仿真调试介绍

- 在主函数97行程序左侧红色框选处双击，生成断点，再次双击消去断点，如下图所示，其余处生成断点方法类

```

65 * 参数说明: 无
66 * 函数返回: 无
67 * 修改时间: 2020年6月27日
68 * 备注:
69 *****
70
71 int core0_main(void)
72 {
73     /*
74     * !!WATCHDOG0 AND SAFETY WATCHDOG ARE DISABLED HERE!!
75     * Enable the watchdog in the demo if it is required and also service the watchdog periodically
76     */
77     IfxScuWdt_disableCpuWatchdog(IfxScuWdt_getCpuWatchdogPassword());
78     IfxScuWdt_disableSafetyWatchdog(IfxScuWdt_getSafetyWatchdogPassword());
79
80     /* Initialise the application state */
81     g_AppCpu0.info.pllFreq = IfxScuCcu_getPllFrequency();
82     g_AppCpu0.info.cpuFreq = IfxScuCcu_getCpuFrequency(IfxCpu_getCoreIndex());
83     g_AppCpu0.info.sysFreq = IfxScuCcu_getSpbFrequency();
84     g_AppCpu0.info.stmFreq = IfxStm_getFrequency(&MODULE_STM0);
85
86     /* Enable the global interrupts of this CPU */
87     IfxCpu_enableInterrupts();
88
89     /* Demo init */
90     module_port_init(); //初始化4个LED用作采集到的电压值反馈
91
92     vadc_moudleinit(); //初始化ADC采集
93     /* background endless loop */
94     while (TRUE)
95     {
96
97         autoscan_run(); //循环采集
98     }
99
100     return 0;
101 }
102
103
104
105 /** \} */
    
```

- 点击运行图标，程序自动运行至断点处停下

```

* 参数说明: 无
* 函数返回: 无
* 修改时间: 2020年6月27日
* 备注:
*****

int core0_main(void)
{
    /*
    * !!WATCHDOG0 AND SAFETY WATCHDOG ARE DISABLED HERE!!
    * Enable the watchdog in the demo if it is required and
    */
    IfxScuWdt_disableCpuWatchdog(IfxScuWdt_getCpuWatchdogPassw
    IfxScuWdt_disableSafetyWatchdog(IfxScuWdt_getSafetyWatchdc

    /* Initialise the application state */
    g_AppCpu0.info.pllFreq = IfxScuCcu_getPllFrequency();
    g_AppCpu0.info.cpuFreq = IfxScuCcu_getCpuFrequency(IfxCpu
    g_AppCpu0.info.sysFreq = IfxScuCcu_getSpbFrequency());
    g_AppCpu0.info.stmFreq = IfxStm_getFrequency(&MODULE_STM0

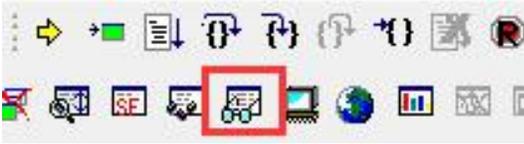
    /* Enable the global interrupts of this CPU */
    IfxCpu_enableInterrupts();

    /* Demo init */
    module_port_init(); //初始化4个LED用作采集到的电压

    vadc_moudleinit(); //初始化ADC采集
    /* background endless loop */
    while (TRUE)
    {
        autoscan_run(); //循环采集
    }
}
    
```

PLS UDE窗口仿真调试介绍

- 点击红色框选处Watch图标，打开Watch窗口



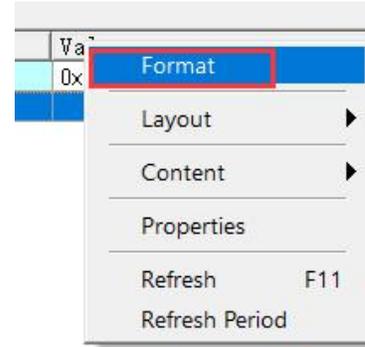
- 复制需要监控的数组

```
#include "Vadc/Adc/IfxVadc_Adc.h"
// VADC handle
IfxVadc_Adc vadc;
IfxVadc_Adc_Group adcGroup;
IfxVadc_Adc_Config adcConfig;
IfxVadc_Adc_GroupConfig adcGroupConfig;
// create channel config
IfxVadc_Adc_ChannelConfig adcChannelConfig[8];
IfxVadc_Adc_Channel adcChannel[8];
unsigned int result[8];
```

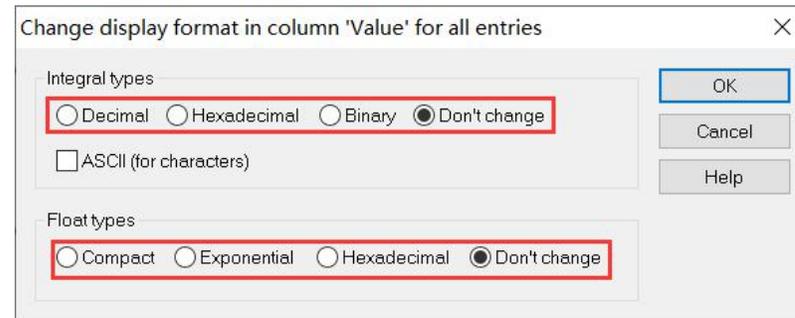
- 粘贴至Watch窗口，添加被监控对象

Name	Value
result	0x6000000C
<new variable>	

- 在Value处右键，选择Format，

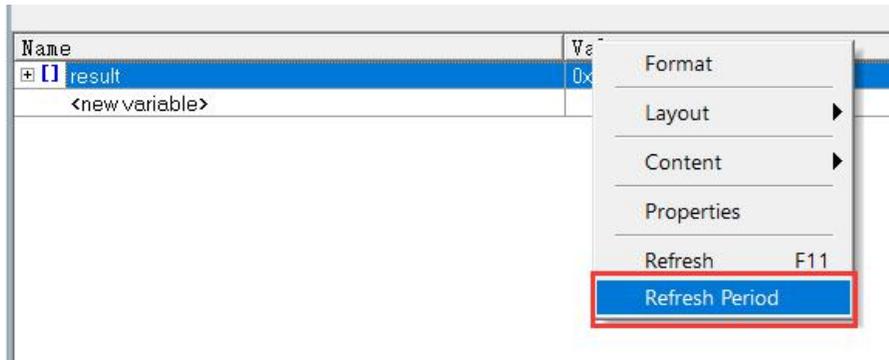


- 在弹出的窗口中，调整整型或浮点型的进制



PLS UDE窗口仿真调试介绍

- 在空白处右键，选择Refresh Period，并在弹出窗口勾选和设置刷新周期

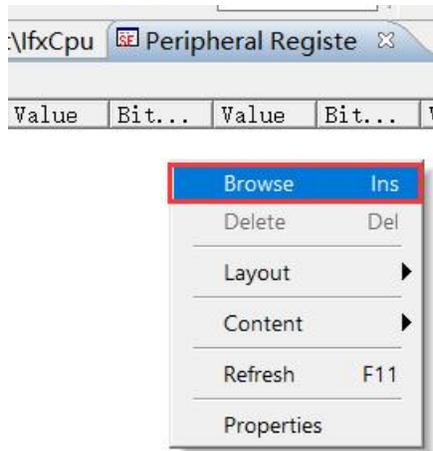


PLS UDE窗口仿真调试介绍

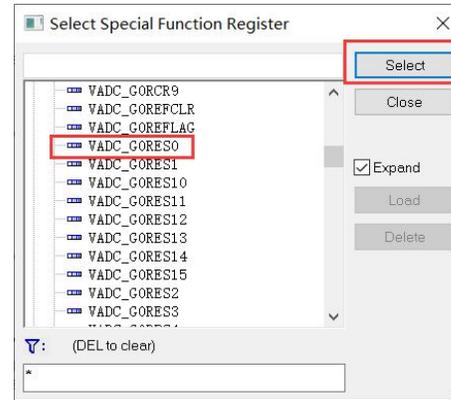
- 点击红色框选处Peripheral Register图标，打开Peripheral Register窗口



- 右键空白处，选择Browse



- 在弹出的窗口中依次选择VADC->VADC_G0RES0,点击Select



- 在Peripheral Register窗口，监控各外设寄存器的各位的状态情况

Name	Value	Bit f...	Value	Bit ...	Value	B...	Value
VADC_G0RES0	0x00000000	VF	0x0 (No n...	FCR	0x0 (Sig...	CRS	0x0 (Request source
		EMUX	0x0	CHNR	0x00	DRC	0x0
		RESULT	0x0000				



2021乾勤科技智能车交流群

2021乾勤科技智能车交流群



在B站学习智能车开发



(2020已满)



■ 北京乾勤科技发展有限公司

电话： 010—82642324

传真： 010—82641640

地址： 北京市中关村北二街四号 水清木华园 5 # 212

邮编： 100190

信箱： service@qianqin.com.cn

网址： www.qianqin.com.cn

若无某种大胆放肆的猜想，一般是不可能有知识的进展的。

——爱因斯坦

