

Infineon 开发工具软件使用《中文版笔记翠》

☞英飞凌 DAVe Drive 软件安装方法...

☞TASKINGvxtoolset 在英飞凌 16 位单片机开发工具设置方法...

☞Memtool3.3.1 版本下载步骤详解...

☞英飞凌 DAP miniWiggler 的安装使用... ..



Infineon 《软件入门安装使用篇》：

英飞凌 MCU 开发套件安装方法介绍





👉 英飞凌DAVE Drive软件安装方法

无刷电机驱动装置的DAVE Drive应用代码生成器

自动代码生成器—DAVE Drive 针对电机控制

英飞凌面向电机控制发布了新一代自动代码生成器—DAVE Drive 版本 2。该工具能够通过图形用户界面（GUI）自动生成电机控制代码，无须程序员手工编写动辄数千行的代码，大大缩短了电机控制开发人员的评估时间。对硬件工程师而言，它可以让他们获得电子系统控制技术的专业知识。此外，DAVE Drive 还能最大限度发挥相关微控制器的功能，它可利用 XC886 的矢量计算机为其生成经过优化的 FOC 代码，而这一工作通常需要开发人员具备电机控制和汇编语言编程的专业知识。

DAVE Drive 还包含一整套完整参考系统板，该板包括微控制器、传感器和功率器件，可以直接和客户的应用联接，如三相电机，为培训、评估和开发项目启动提供了一个完善的平台。

DAVE Drive 是 DAVE(虚拟数字应用工程师) 系统一个以应用为中心的插件，它提供了初始化、配置和驱动代码，从而简化了初学者和控制专家的编程工作。该工具能够生成完整的算法源代码（C 语言和汇编语言），可直接被 Keil 或 Tasking 编译器等常用工具进行编译和调试。

DaveDrive 支持 BLDC 和 PMSM 电机，带霍尔传感器的换向控制和无传感 FOC 控制等不同算法，已可用于 XC800 微控制器系列产品。



DAVE™ Drive 是一个基于图形化用户界面（GUI）的软件工具，允许应用开发人员高效配置 XC800 和 XE164 Classic 软件，以控制无刷同步三相电机。利用 DAVE™ Drive，开发人员仅需轻点鼠标即可选择电机，设置电机速度、控制类型以及多种其他选项，生成相应的定制代码。DAVE™ Drive 充分利用了英飞凌微控制器的强大功能，例如，它可以利用增强型矢量计算机，生成适用于 XC878 的优化 FOC 代码，而这通常要求同时具备电机控制和汇编语言编程方面的专业知识。

DAVE™ Drive 是英飞凌 DAVE™（数字应用虚拟工程师）的一个应用为中心的插件。DAVE™ 提供了初始化、配置和驱动程序代码，以便新手和专家轻松编程。这个工具可以生成源代码形式的完整算法，并可利用诸如 Keil 编译器或 Tasking 编译器等流行工具，轻松进行编译和调试。



DAvE（数字应用虚拟工程师）Drive是英飞凌面向8位（XC886/8）和16位（XC164CM/CS）微控制器家族的应用代码生成器。它可提供初始化配置和驱动程序代码，帮助轻松实现编程。全新的DAvE Drive由于可快速、轻松地配置复杂外设，因此可缩短无刷直流（BLDC）电机控制装置的软件开发时间。电机控制装置设计者可以集中精力开发具体应用软件，如洗衣机或洗碗机程序，从而快速制出样机。与竞争性配置工具相比，DAvE Drive软件工具可以生成完整的算法，无需从数据库中获得代码。它可以根据用户选择的电机，灵活生成代码，使系统设计时间缩短大约1人年，同时使系统独具特色。此外，它还可以执行块交换和磁场定向控制。

全新的DAvE Drive软件工具包含在BLDC Drive应用套件中，是英飞凌8位和16位Easy Kit的模块化扩展。这些组合工具构成了包含微控制器硬件、软件、功率半导体和电机驱动器的工业和汽车电机控制应用完整开发环境。

Infineon(英飞凌)公司DAVE2.1软件的使用方法和一些需注意的要点，并按照流程编写了一个“Infineon XC164CM”的闪灯测试程序。通过对简单应用程序形成过程的简单描述，使编程人员不再为Infineon单片机繁多的寄存器而感到头疼，使编程工作更加快捷明了利用DAVE2.1完成Infineon单片机的软件开发流程。

从Infineon网站免费下载DAVE2.0，并正确安装。

英飞凌公司（Infineon）的DAvE（Digital Application Virtual Engineer，数字应用虚拟工程师）免费软件是该公司针对其8位（C500/C800/XC800系列）、16位（C166/XC166/XE166系列）和32位（XC2000/TriCore系列）微控制器家族而推出的应用代码生成器。它可提供初始化配置和驱动程序代码，以便轻松地实现编程。

A 使用DAvE软件，方便用户快速、轻松地配置复杂的接口设备，缩短各种控



制装置软件开发时间, 保证用户集中精力开发具体应用软件。

英飞凌8位单片机硬件的连接

■ JTAG接信号线定义:

TMS	测试模式选择
TDO	测试数据串行输出
TDI	测试数据串行输入
CPU_CLOCK	测试机时钟
TCK	测试时钟
RESET	目标系统复位信号
TRST	测试系统复位信号
OCDSE	OCDS配置 (XC800中不使用)
RCAP1/2	保留 (留作特殊应用时使用)
GND	接地信号线
VCC	电源

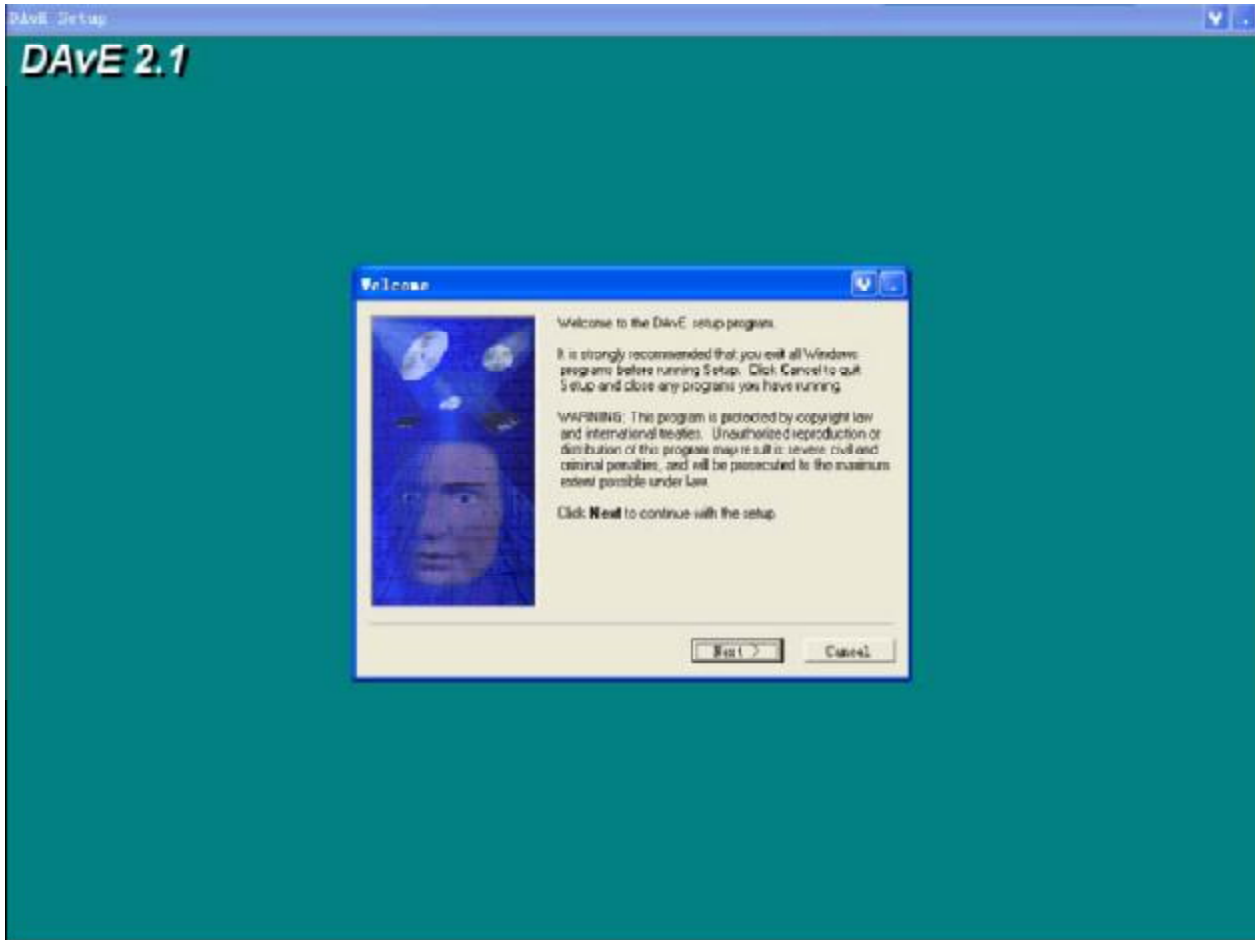
DAvE2.1的安装

A DAvE2.1 的安装步骤:

A 首先下载DAvE的安装软件和相关芯片的DIP文件。



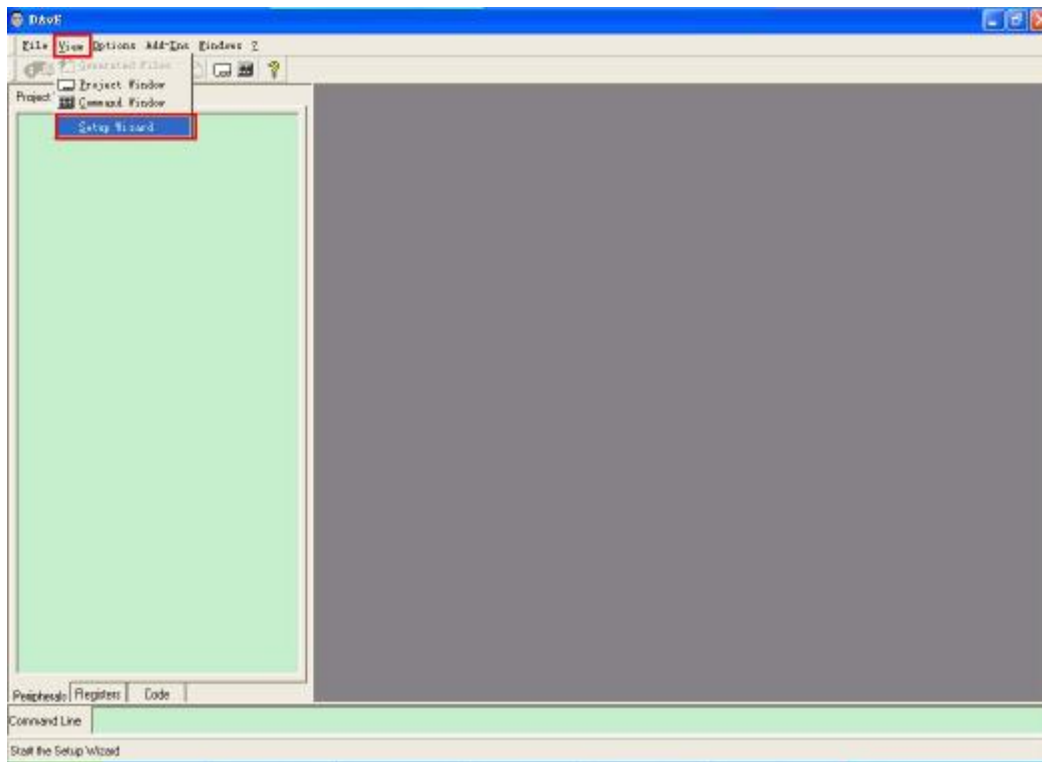
A 然后运行DAvE的安装软件，进入DAvE的安装界面



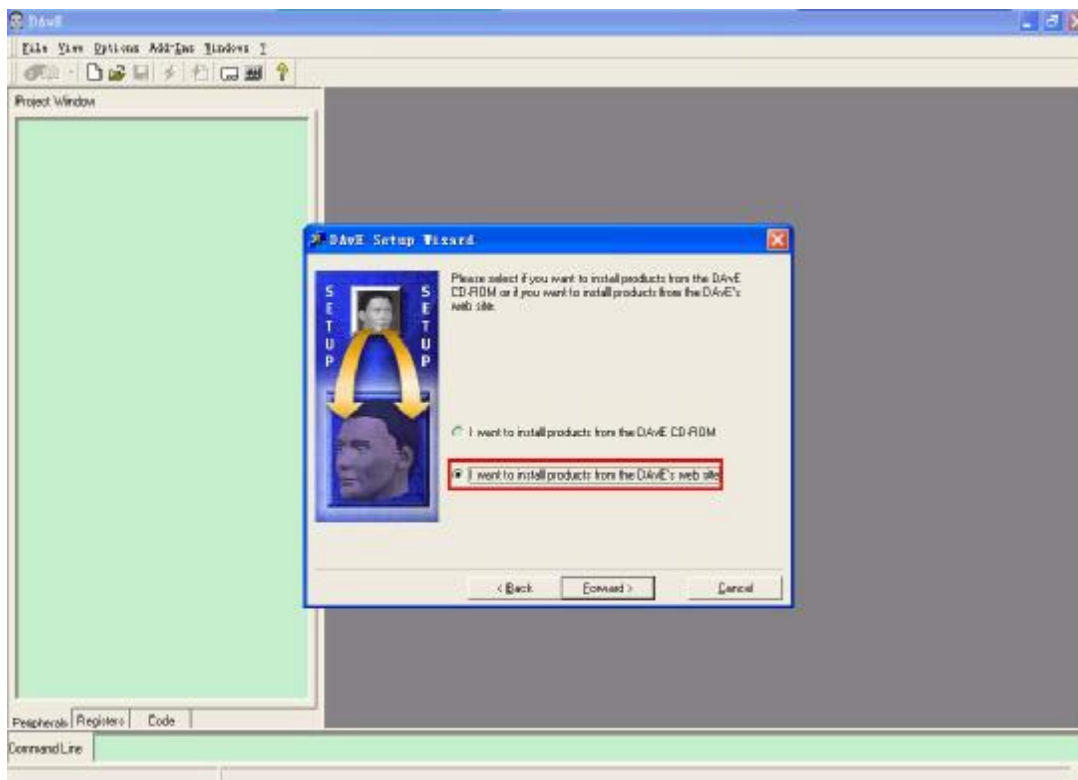
A 安装完成以后，在桌面出现DAvE的图标，双击图标进入DAvE软件的使用界面。

A 在首次进入时，需要安装相应的DIP文件。

A 选择“View → Setup Wizard”。

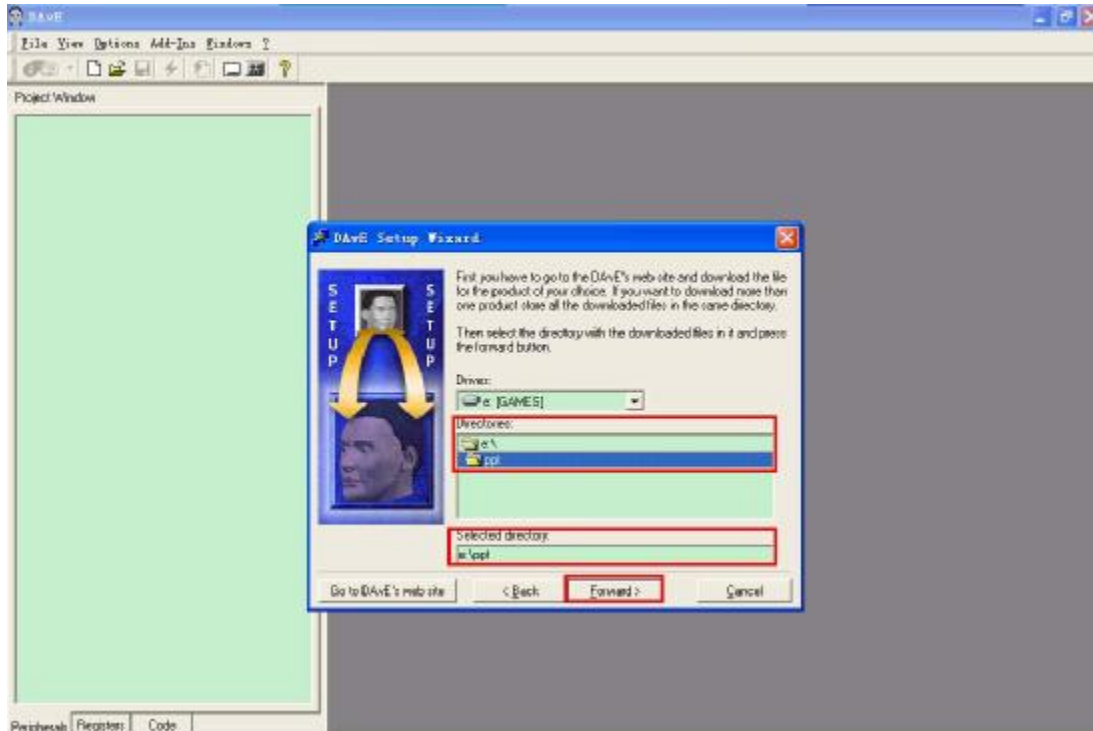


在进入设置界面后，选择“Install”然后点击“Forward >”进入到安装方式选择界面，需要选择“I want to install products from the DAVE’s web site”。

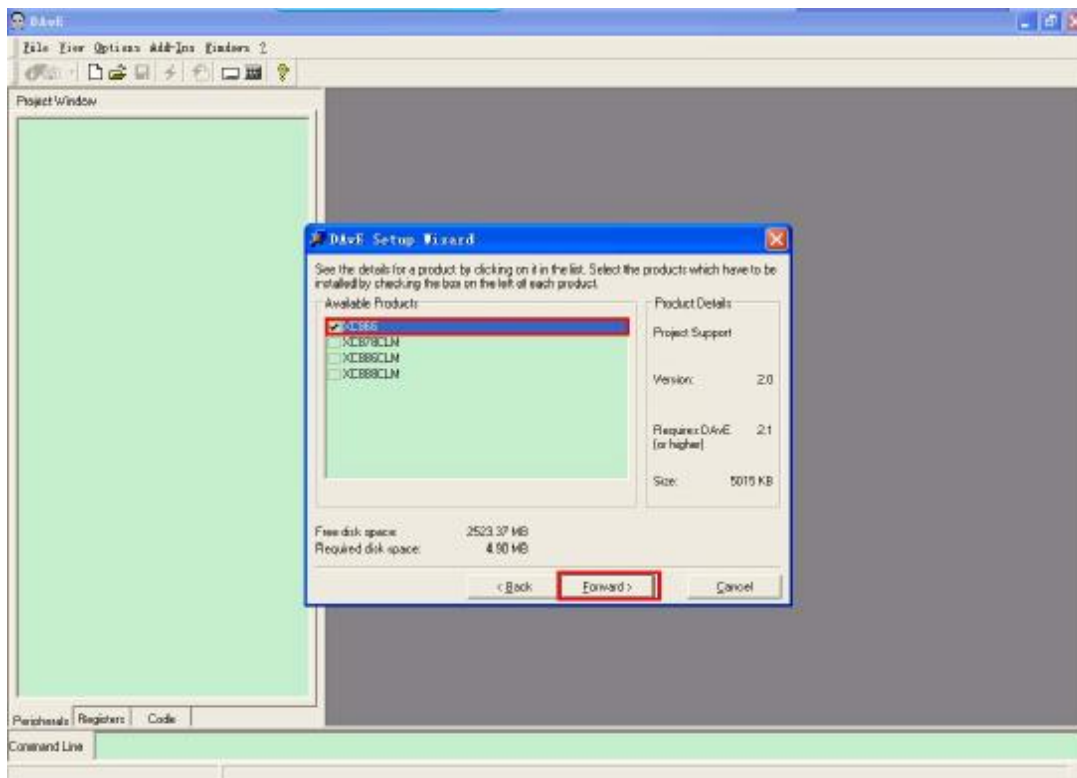




A 当进入到选择文件界面的时候, 选择DIP文件的存放文件夹。



A 在选择安装产品界面, 不同的DIP文件可以安装XC866, XC886CLM, XC888CLM和XC878CLM等产品。选中安装“XC866”或其它型号的产品, 然后点击“Forward >”进行

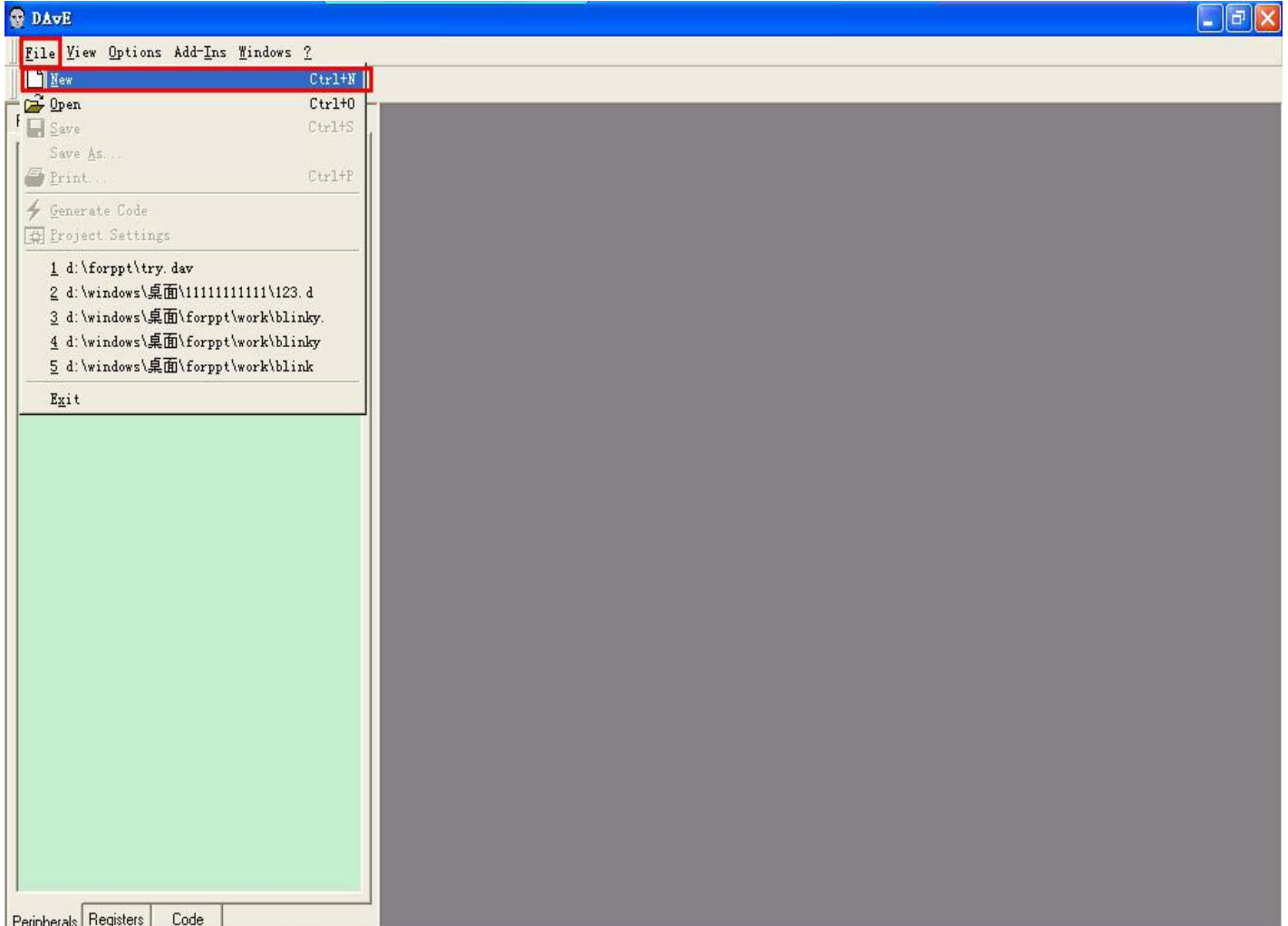


安装。



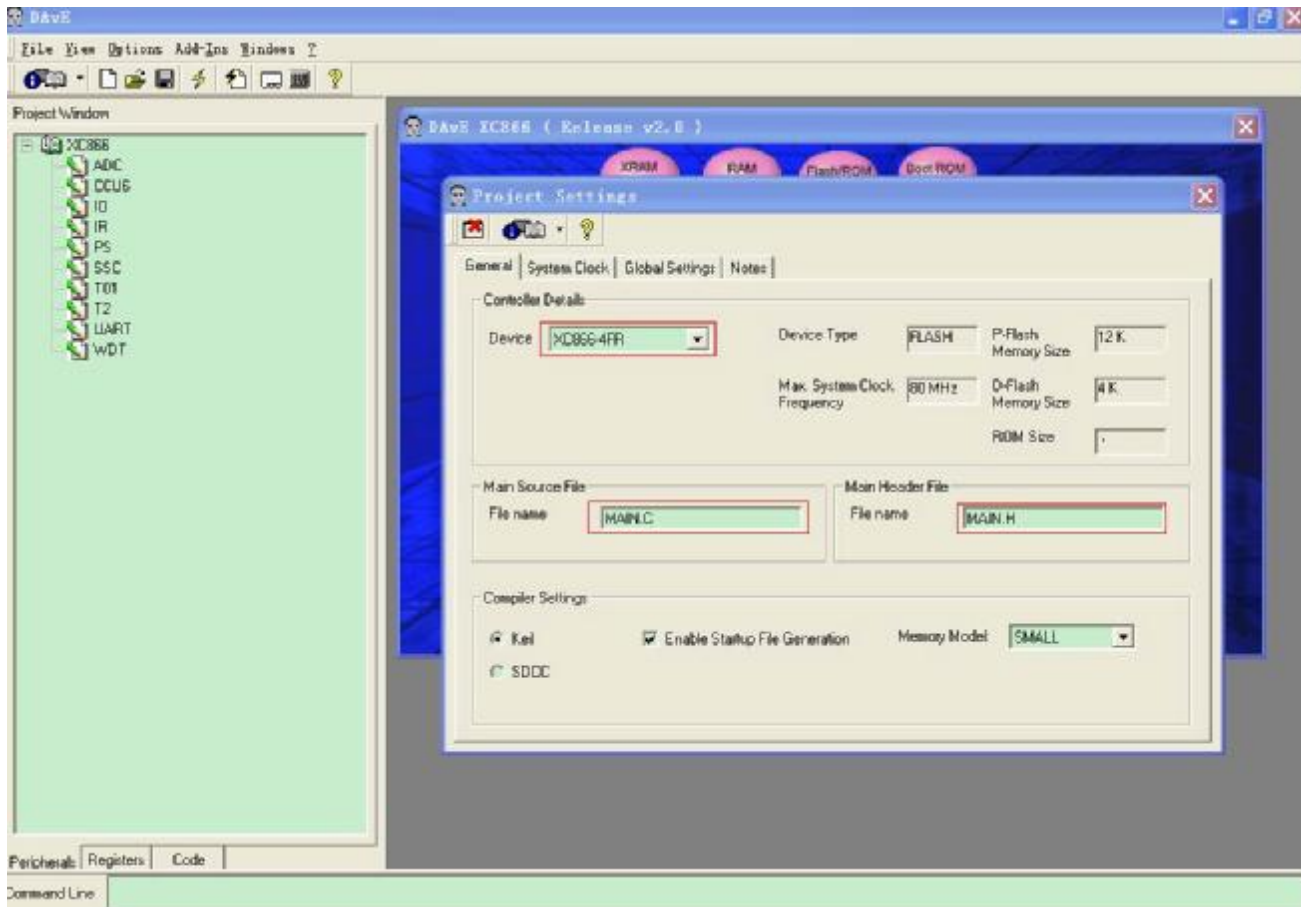
A 在完成DAvE安装后，就可以进入到DAvE的使用界面了。

A 首先建立一个新工程。选择“File → New”进入建立工程选项。

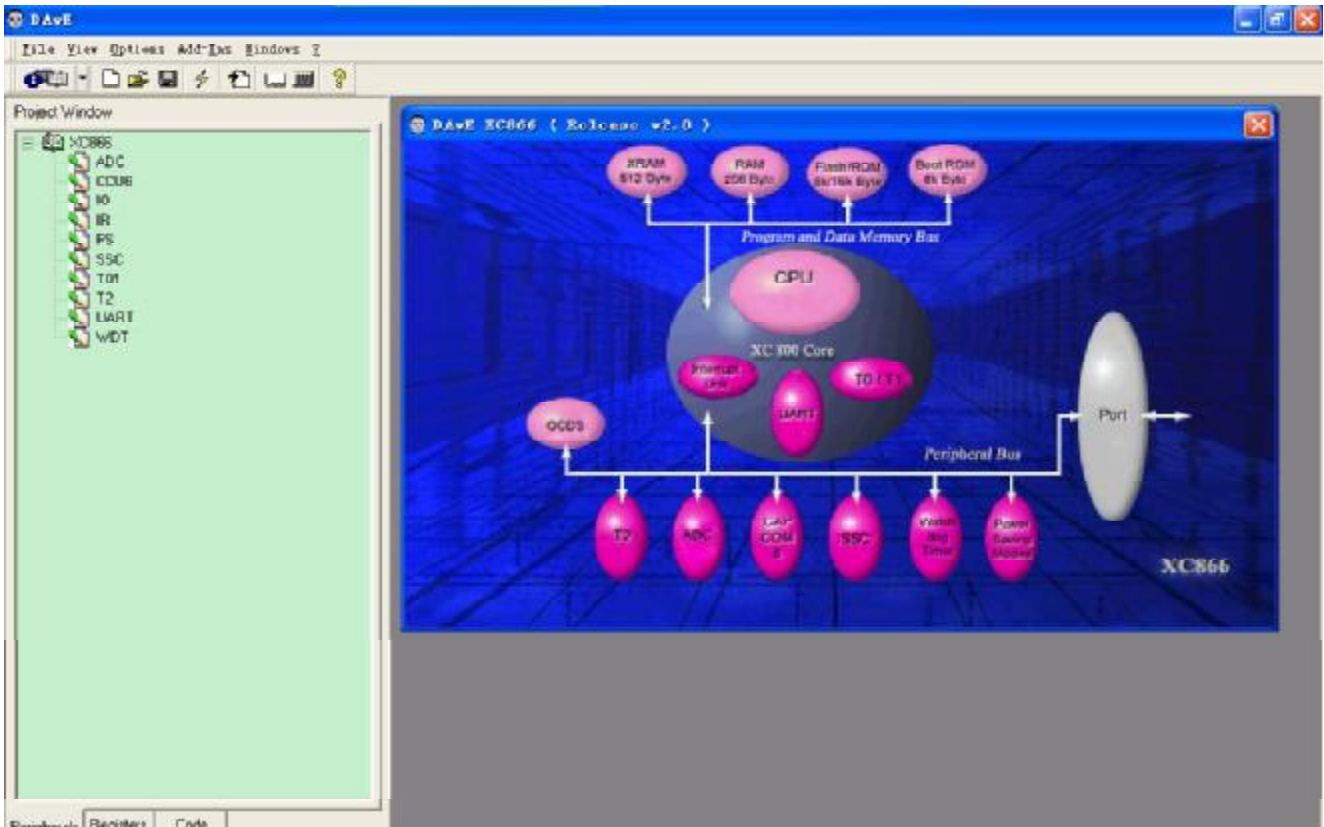


A 选择“Create”后进入到DAvE软件主界面。在这里进行基本的设置。首先需要设置的是“ProjectSettings”。在该选项卡中，选择单片机的型号，

如：选用的型号为XC866-4FR。设置项目自动生成的C程序文件的名称，以及头文件的名称等。



设置完成后，关闭选项卡，进入到各个分项目的设置

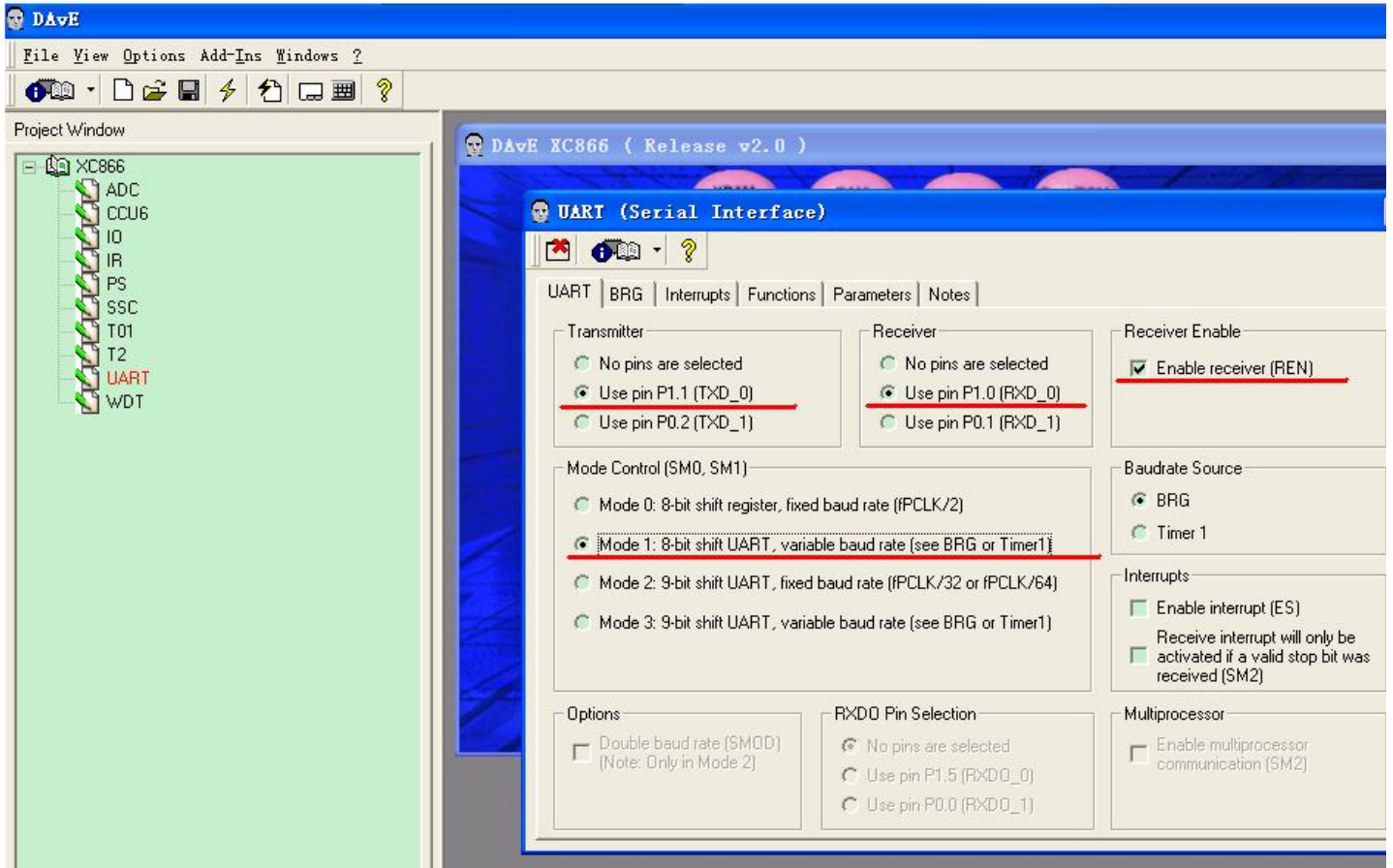




在这里，特别需要修改几个地方：选择“**UART**”选项，

选中：**Use pin P1.1 Use pin P1.0 Enable Receiver**

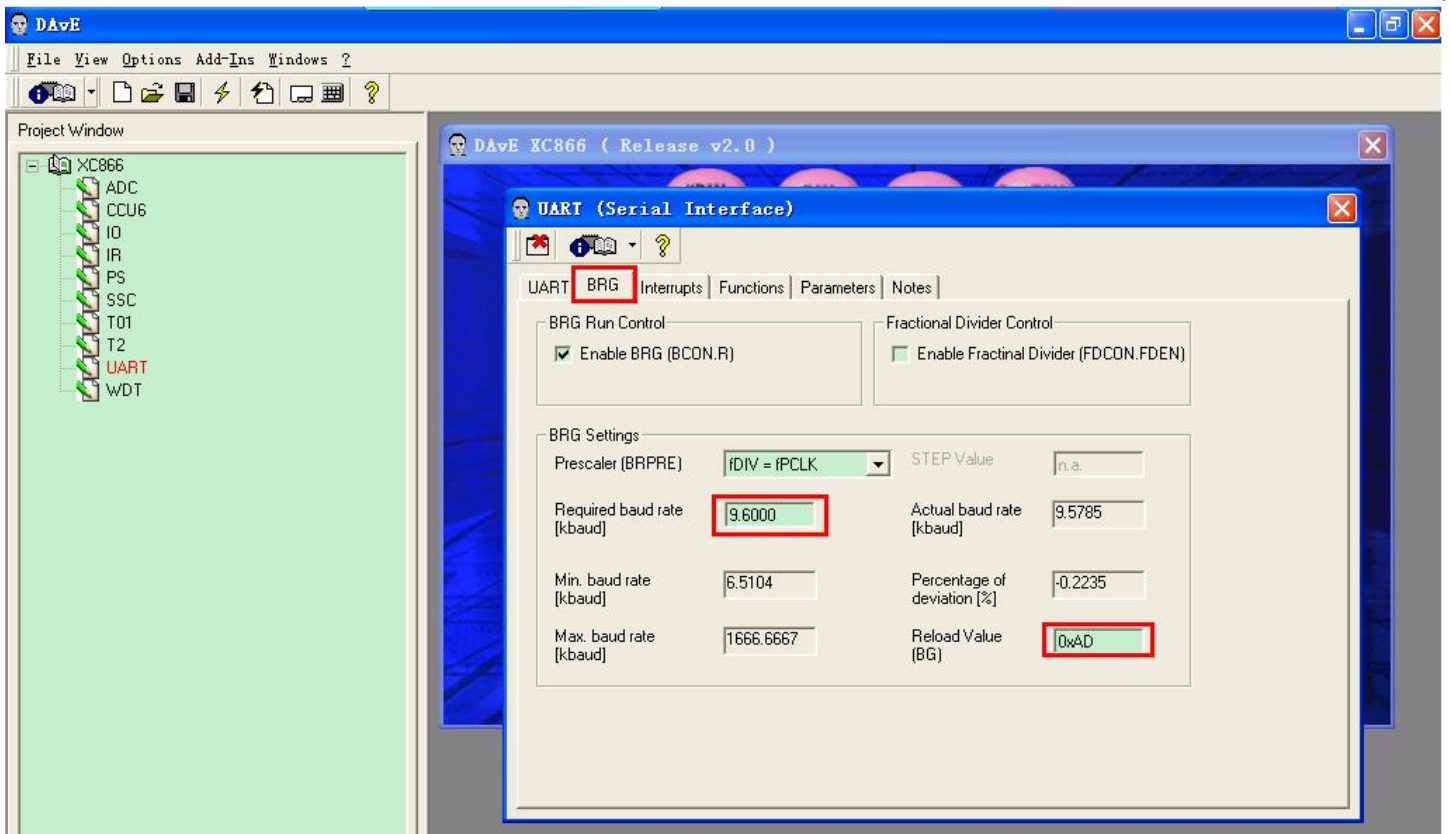
该选项设置评估板上单片机与电脑串口连接实际使用的引脚



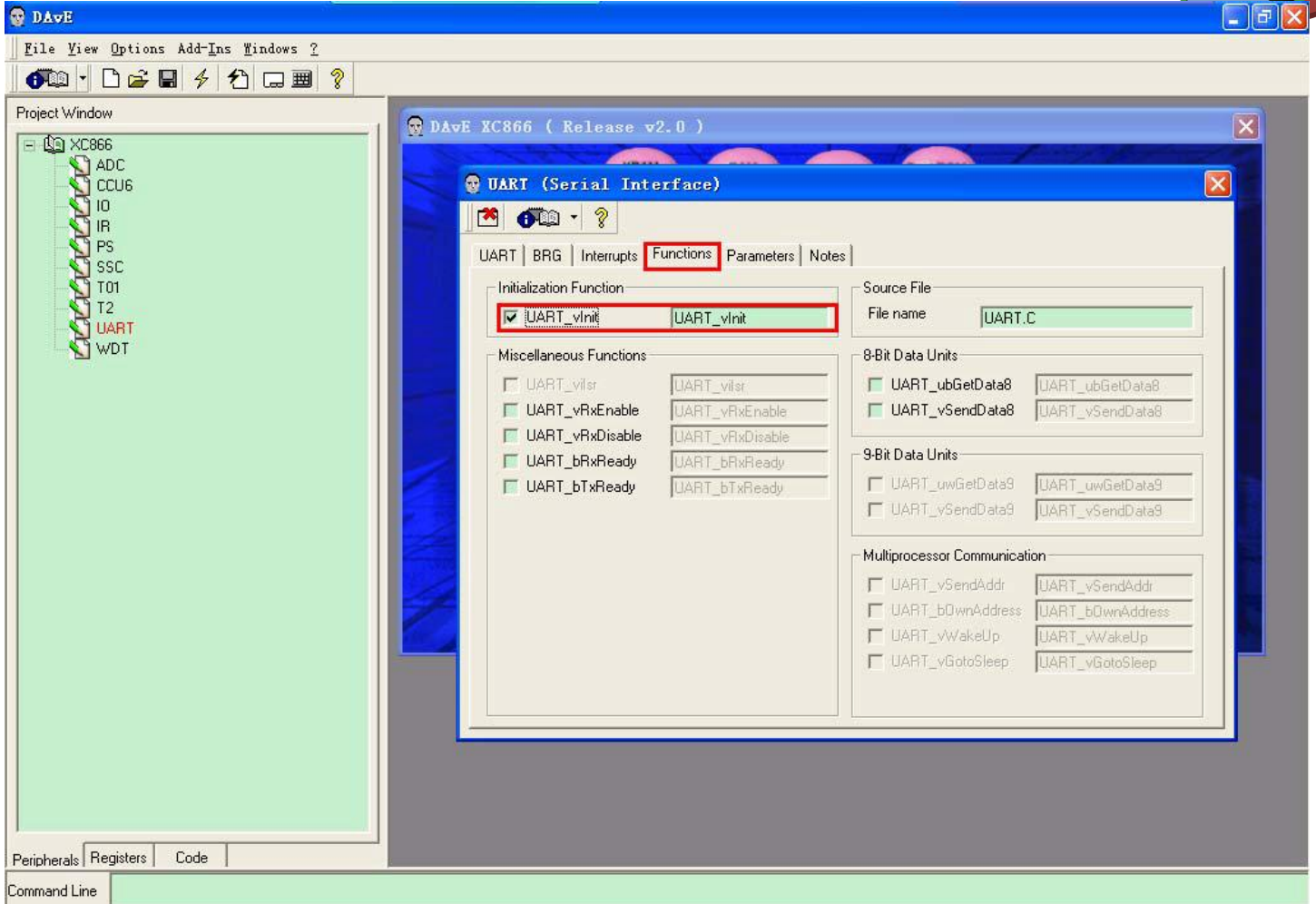
A 在“**BRG**”选项卡中需要修改**UART**的波特率重载值（**Reload Value**）。

填入实际需要使用的波特率：**9.6**（并按“回车”键），这里**DAVE**会自动生

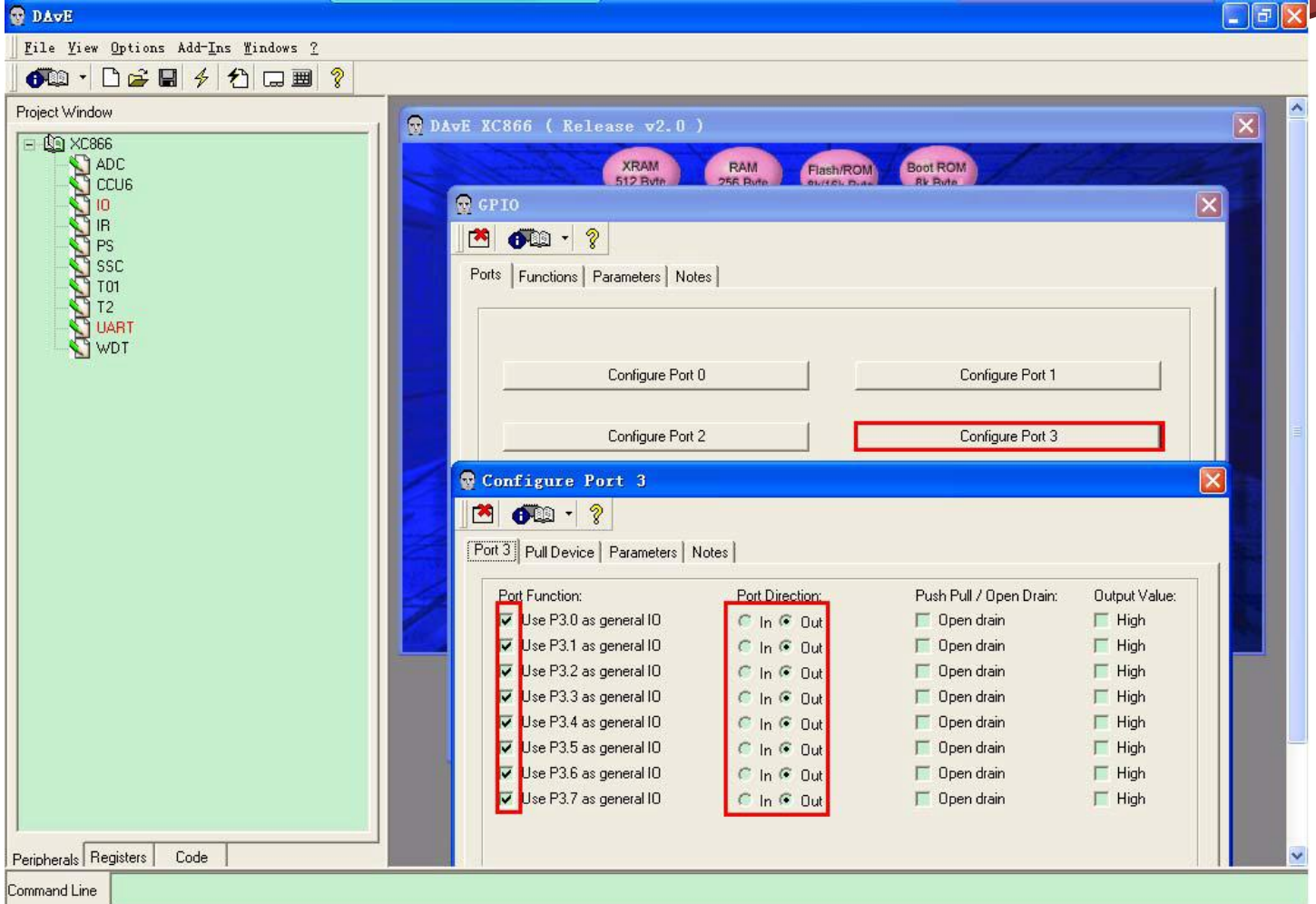
成重载值为**0xAD**，避免了手工计算，降低了工作难度。



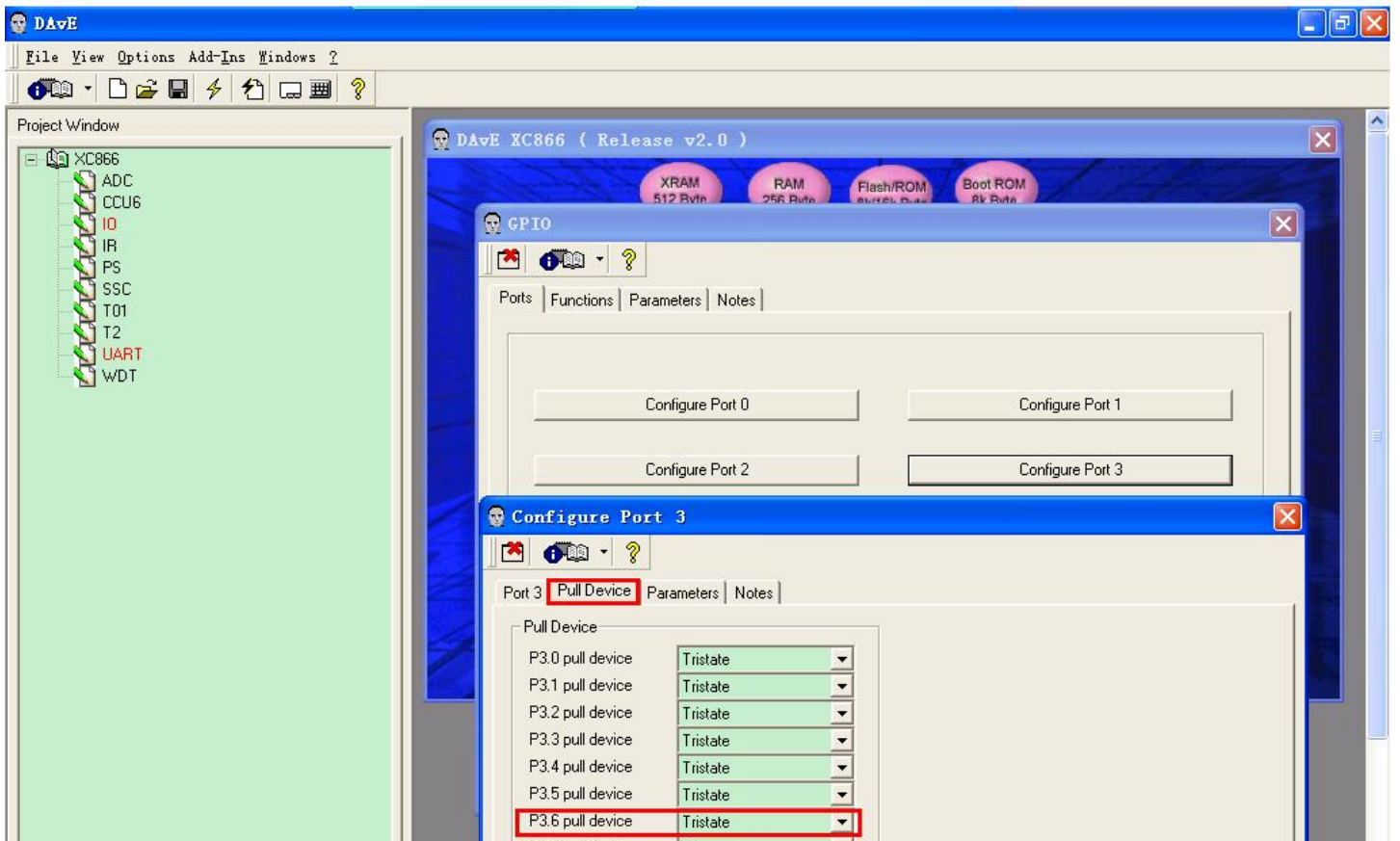
A 在“Function”选项卡中，需要在“UART_vInit”前面打勾。UART_vInit函数用于UART的初始化设置。完成修改，关闭选项卡。



A DAVE进行端口设置：选择“Port”进入端口设置，选择“Configure Port 3”，将端口3（P3.0~P3.7）全部设置为输出。



A 端口模式配制: 这里, 需要将**P3.6**口设置成为“**Tristate**”, 使用三态模式。设置完成后, 关闭该选项卡返回上层。



A 函数配制: 选择“**Functions**”选项卡, 可以选择DAvE自带函数选择。这

时, 移动光标到该函数体处, 这时会自动显示该函数的说明。**A**

选择函数:

IO_vInit

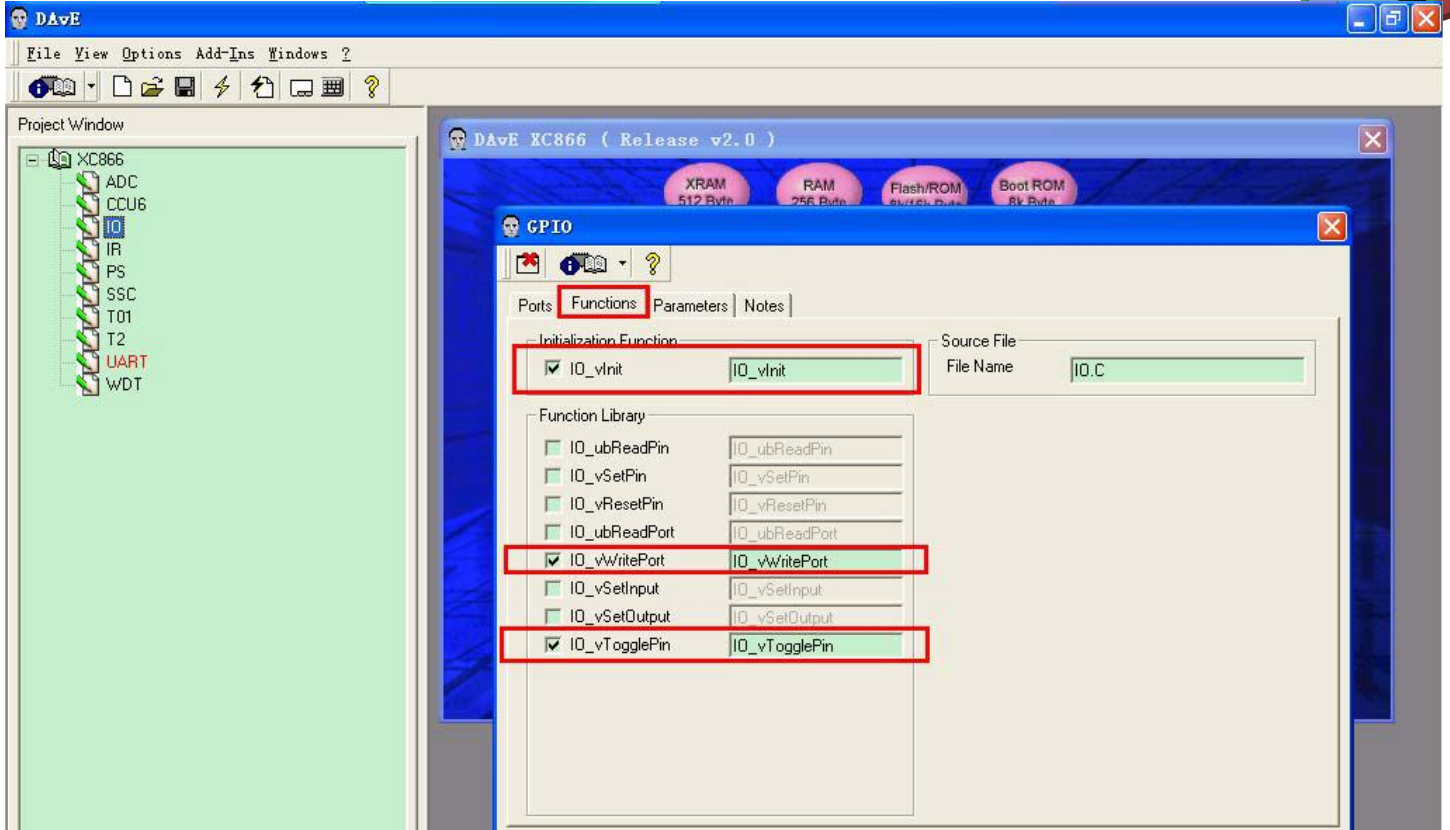
(端口初始化)

IO_vWritePort

(写端口)

IO_vTogglePin

(端口状态取反)



文件保存: 选择“File → Save”进行文件保存。这里, 保存为“blinky”。

保存后, 点击“ ”图标生成代码。

在保存目录下看到11个文件:

blinky.asm, blinky.rtf, blinky.dav, blinky.dpt, IO.C, IO.H,
UART.C, UART.H, MAIN.C, MAIN.H和START_XC.A51

其中blinky.asm、blinky.rt、blinky.dav、blinky.dpt 四个文件包含了创建工程的基本信息, 在某些工程中会用到, 不需要进行修改。

IO.H文件中, 包含各个端口, 针脚的定义。

IO.C文件中, 包含端口的初始化。如果需要对一些端口进行初始化设置, 可以在IO.C文件中的void IO_vInit(void) 函数中添加代码实现。

```
// USER CODE BEGIN (IO_Init,3)
```

```
SFR_PAGE(_pp1, noSST); // switch to page 1
```




```
P3_PUDSEL = 0xBF; // load pull up/down select register
```

```
P3_PUDEN = 0x00; // load pull up/down enable register
```

```
//设置端口3状态
```

```
SFR_PAGE(_pp0, noSST); // switch to page 0
```

```
P3_DIR = 0xFF; // load direction register
```

```
//设置端口3为输出
```

```
// USER CODE END
```

A UART.H文件包含通讯函数的原形等, 除需要添加之外建议不要修改。

A UART.C文件中具体包含了通信的一些基本设置。包括传输模式, 传输使用的波特率等。

```
// USER CODE BEGIN (UART_Init,3)
```

```
SFR_PAGE(_pp0, noSST); // switch to page 0 without saving
```

```
P1_DIR |= (ubyte)0x02; // set output direction
```

```
//设置P1口为输出端口
```

```
MODPISEL &= ~(ubyte)0x01; // configure peripheral input select
```

```
//register
```

```
BCON = 0x00; // reset baudrate timer/reload
```

```
// register
```

```
//设置波特率
```

```
SCON = 0x50; // load serial channel control register
```

```
//寄存器SCON中, 含有T0, T1中断状态标准等。
```

```
// USER CODE END
```

A START_XC.A51 文件中定义了一些标准SFR寄存器符号以及



其他的基本

参数。使用这些定义好的符号提高程序的可读性。

; Standard SFR Symbols

ACC DATA 0E0H

B DATA 0F0H

SP DATA 81H

DPL DATA 82H

DPH DATA 83H

; XC8xx specific SFR Symbols used in STARTUP code

sfr SCU_PAGE = 0xBF

sfr PLL_CON = 0xB7

sfr OSC_CON = 0xB6

sfr PASSWD = 0xBBA MAIN.H文件包含所用到的目标芯片的端口地址，特殊功能寄存器的地址。

另外MAIN.H文件中，包含了一些头文件，提供一些其他函数的调用。同

时也可以添加一些代码，用于程序调用。代码需要添加到**USER CODE**

BEGIN--USER CODE END之间，当重新修改DAvE并生成代码时，用户

代码可以得到保留。

```
#include <intrins.h>
```

```
#include "IO.H"
```

```
#include "UART.H"
```

```
// USER CODE BEGIN (MAIN_Header,11)
```



// 添加用户代码

// USER CODE END **A** 在MAIN.C文件中, 具体包括两部分:

A 头文件:

→在自动生成的头文件中, 一般都只包含一句

#include "MAIN.H"

→如果需要添加一些函数原形, 则在这句之后添加。

A 函数:

→自动生成的main.c文件中, 包含有两个函数。

- **void MAIN_vInit(void);** 该函数用于初始化。可以将一些初始化任务添加到该函数中。

- **void main(void);** 该函数为主函数, 程序将从这个函数开始执行。下面编写的代码, 也主要添加到这个函数中。

DAVE 3 是针对其 XMC4000 工业**单片机**家, 提供全面、高效的开发支持: 其 DAVE™ 3 集成式开发平台环境, 已可在英飞凌网站免费下载。它包含基于 DAVE™Apps 的自动代码生成器、免费 GNU 编译器、免费调试器以及 **Flash** 加载器等。此外, 英飞凌已经与超过 15 家合作伙伴展开合作, 不久还将有更多合作伙伴相继加入这一阵营。他们将进一步为日前发布的采用 ARM® **Cortex™** M4 处理器的 XMC4000 家族, 提供特定开发工具, 包括编译器、调试器、软件分析工具和 **Flash** 烧录工具, 以及软件**解决方案**、培训和咨询服务等。

英飞凌科技股份有限公司工业与多元市场业务部负责**单片机**产品的高级主管 Stephan Zizala 表示: “英飞凌呕心沥血打造的 DAVE 3, 旨在促进和缩短客户的软件开发流程。借助我



们免费提供的 DAVE 3，开发人员可以通过图形化用户界面，高效地使用这些针对应用而优化的创新外设。”

DAVE 3: 为 XMC4000 提供全面高效的开发支持

DAVE 3 是一个专为 XMC4000 家族打造的基于 Eclipse 的免费软件开发平台。该环境包含一个免费工具链，可支持自动生成代码，从而有助于使用被称为“DAVEApps”的预先定义并经测试的软件组件。与当前那些提供库和代码示例的典型开发工具不同的是，DAVE Apps 更加抽象，允许以图形化方式组合多个 DAVEApps，并可实现自动、无冲突地映射至芯片上的可用硬件资源。除免费 DAVE 3 之外，首批 28 个 DAVEApps 亦可供下载，以支持以太网、TCP-IP 和文件系统等应用，以及数据完整性和加密/解密**解决方案**。第一套 DAVEApps 还可支持生成脉宽调制信号（PWM）、多种不同的计数器和**定时器**用例、时钟配置、中断处理和 DMA（直接存储器存取）配置等。将陆续发布更多 DAVE 应用；这些 DAVE 应用将涵盖范围广泛的底层和中间件软件应用，包括可支持 CMSIS RTOS API（应用编程接口）的免费实时操作系统。

DAVE 3 是一个开放平台，因此，该环境允许灵活扩展和调整。除免费编译器、链接器、调试器、Flash 加载器之前，还将有商用工具厂商提供的插件。因此，用户可以继续使用其原先熟悉的 IDE，同时充分利用 DAVE 3 的自动代码生成功能。

☞ TASKINGvxtool set在英飞凌16位单片机开发工具设置方法

打开软件之前，创建好一个专用的文件夹，如 D:\myspace。打开软件，选择该文件夹为工作空间，以后创建的工程都将在这个文件夹中，其它文件夹里的工程可以按下面第一



种情况的处理方法 COPY 到本空间里。


第一种情况：调用已有程序

- 1、【FILE】 à 【IMPORT】
- 2、【general】 à 【Existing Projects into Workspace】
3. 选择工程所在路径, 如: 
- 4.、Enable 【Copy Projects into workspace】
- 5.、【finish】

第二种情况：创建新工程

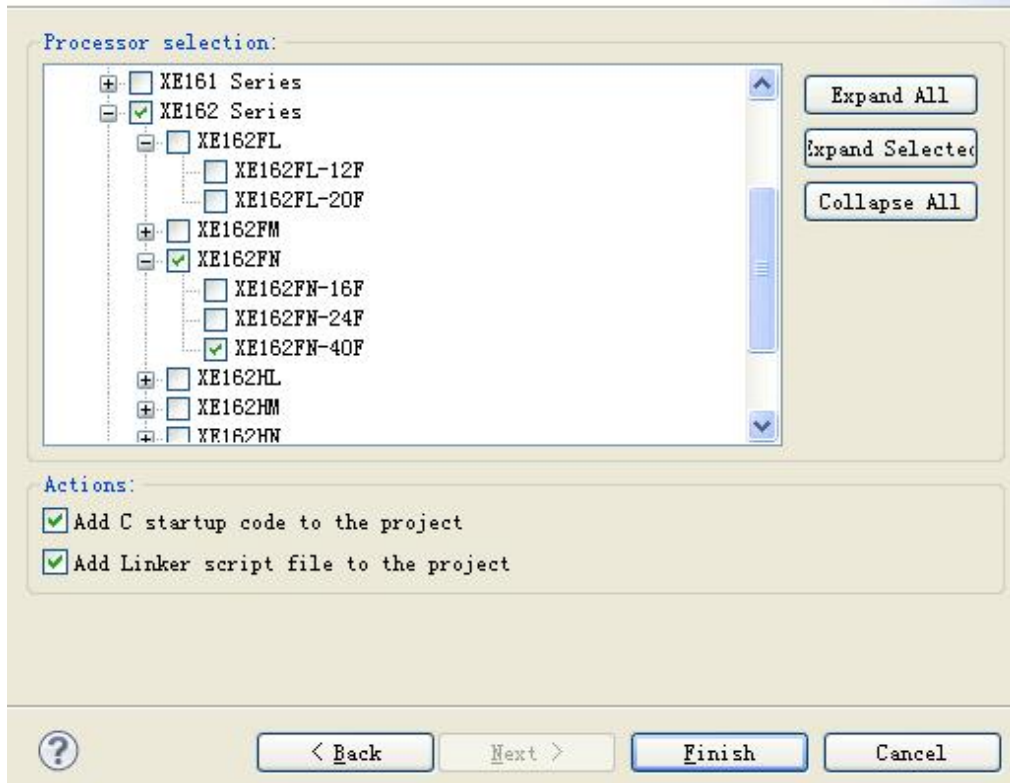
一、1. file--ànew--à TASKING VX-toolset for C166 C/C++ Project.

2. 输入项目名称: 如



Project name: 162fnadc

3. 选择项目默认存放位置: Use default location 。
4. 选择 TASKING C166 Application--à Empty Project
5. NEXT
6. 选择芯片类型: 如 XE162FN-40F



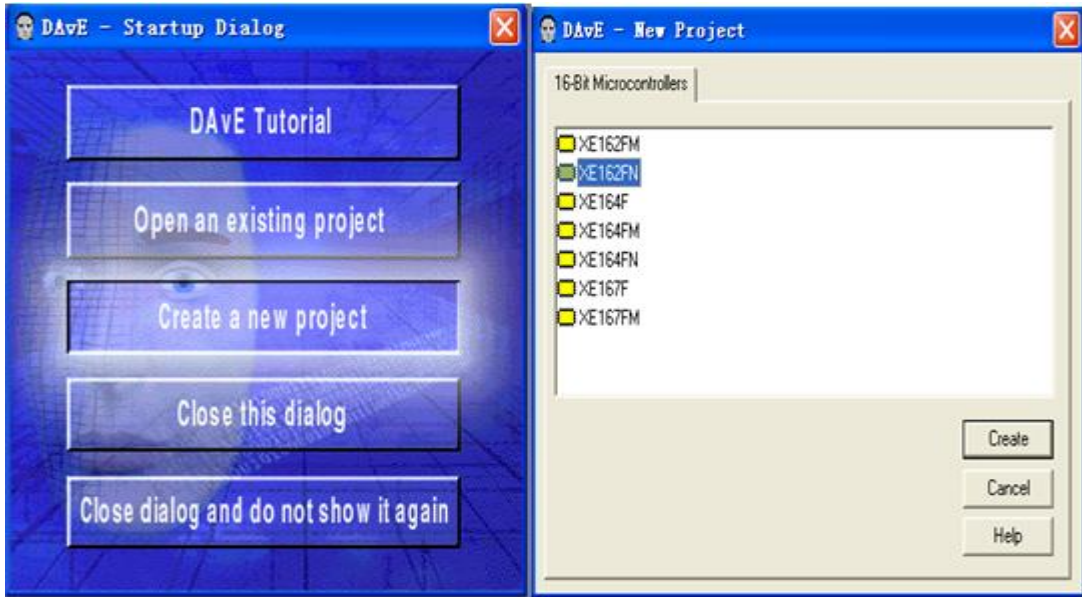
7. Finish 即完成一个新的空工程创建，工程如下图：



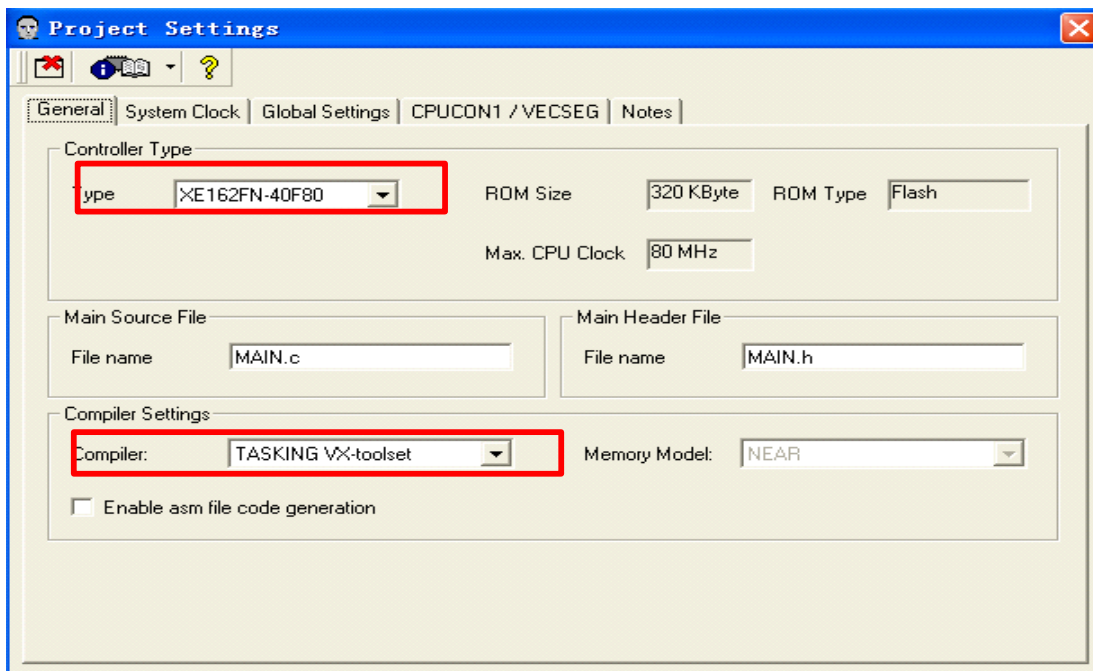
二、用 Dave 生成与芯片 XE162FN-40F 相关的 “.dav ” 工程文件。本文把 ADC 模块作为实例说明创建 “.dav ” 工程文件的过程。其它模块的使用参考手册。

1、打开 DAVE；

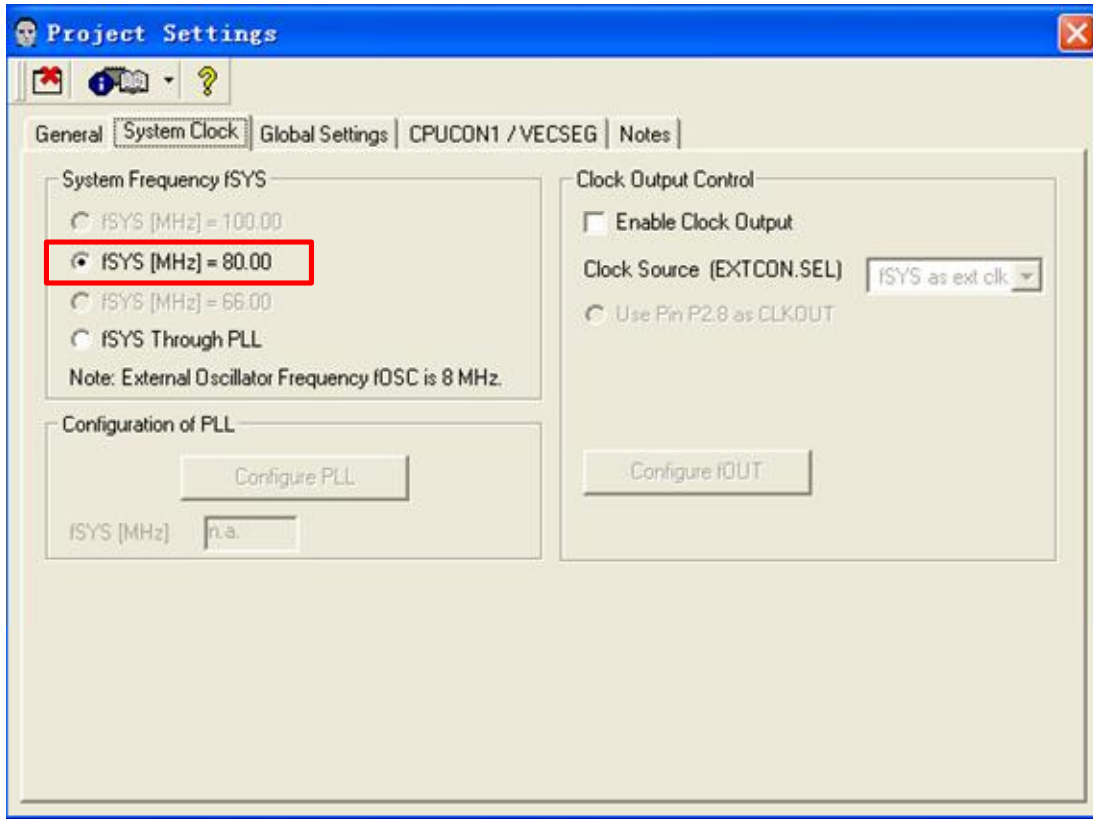
2、选择【Create a New Project】：处理器选择【XE162FN】 ---->【Create】



3. 进入 Project Settings ----> General

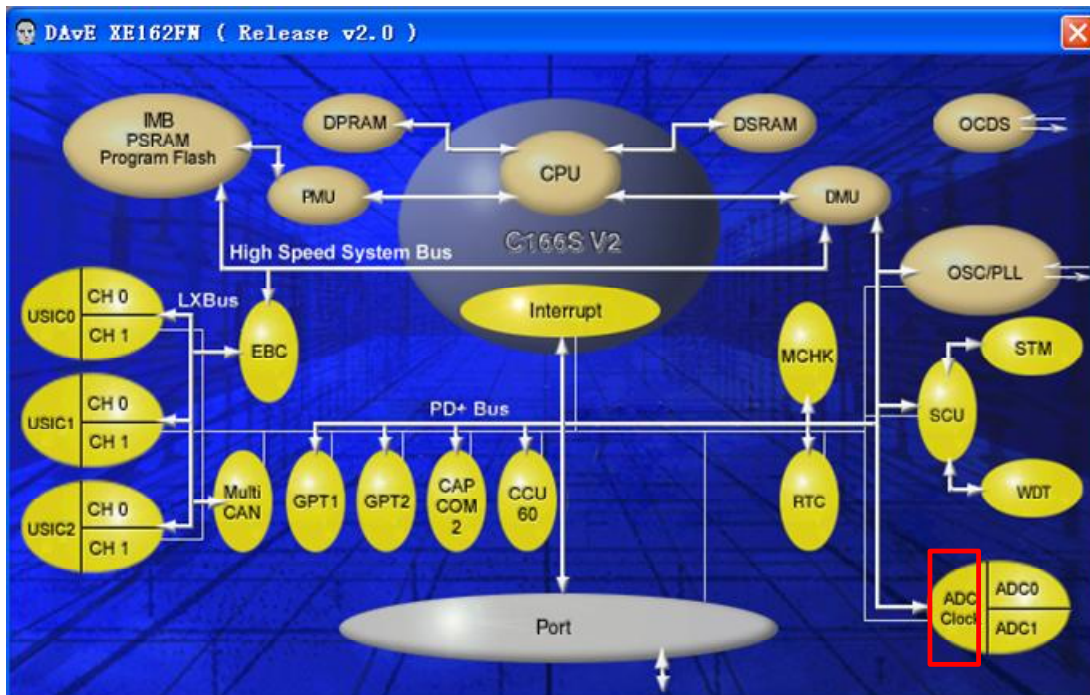


System clock

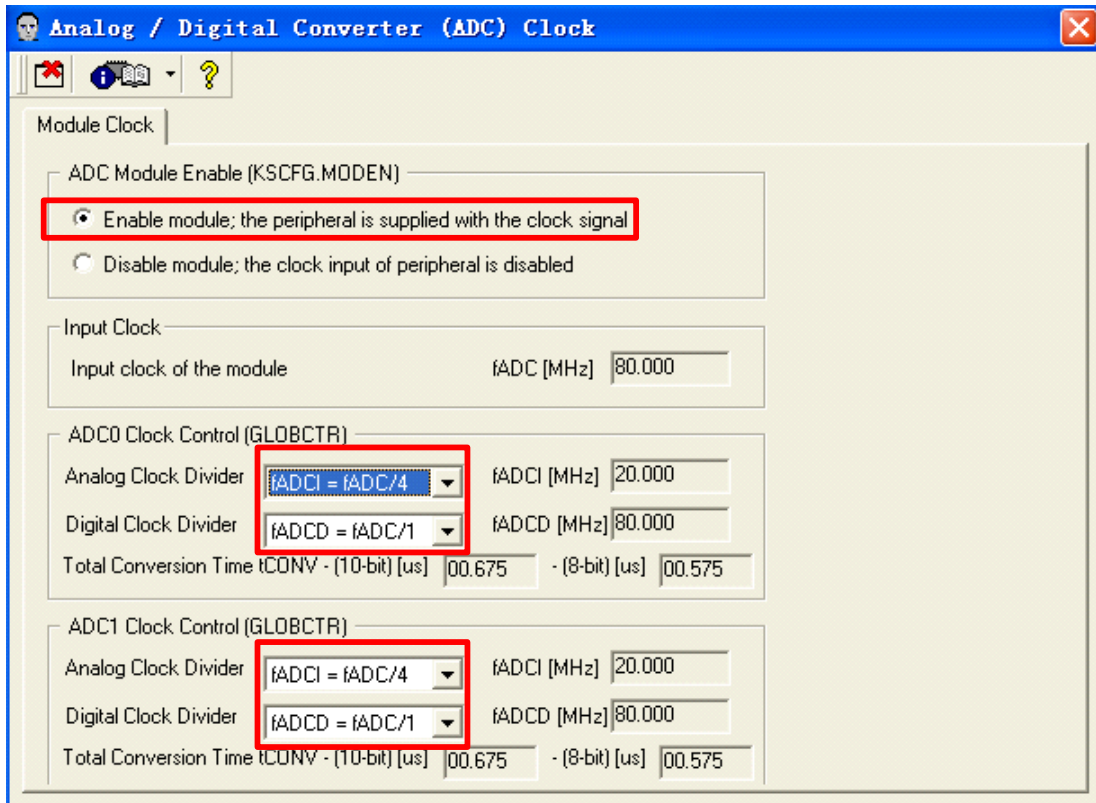


4、配置 ADC 模块 (XE162FN)

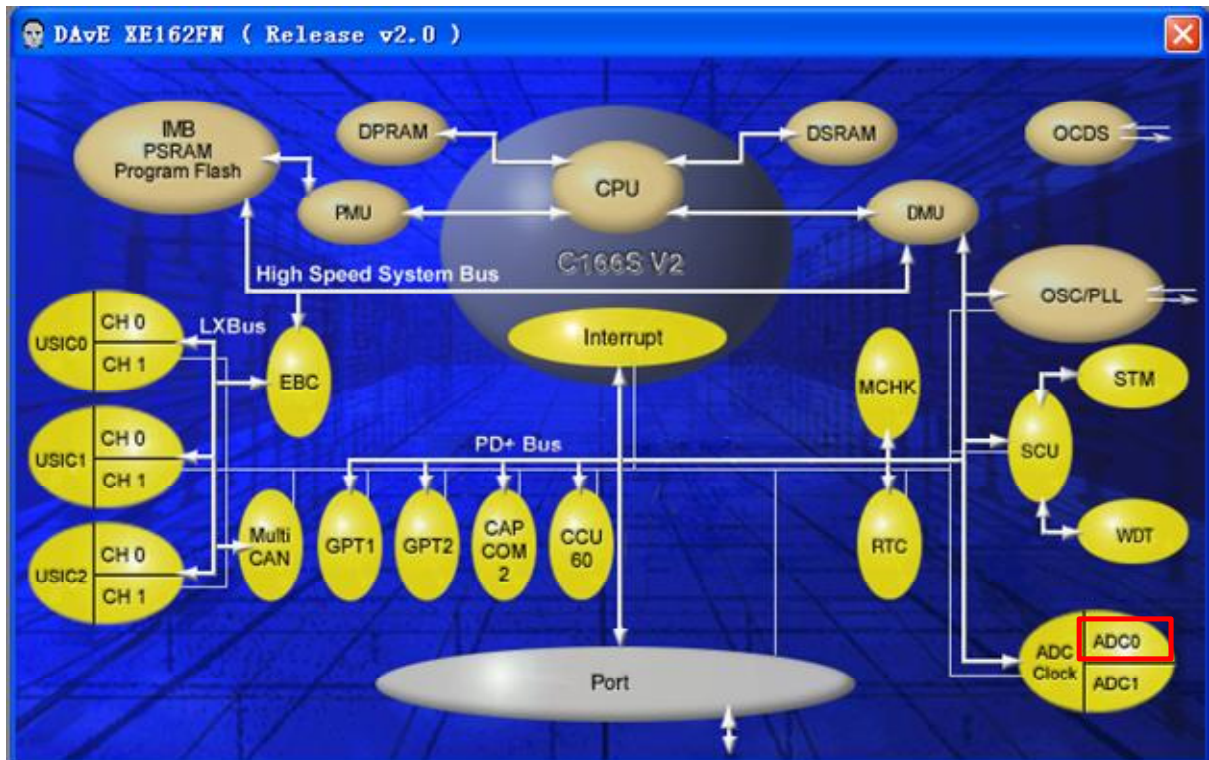
点击 ADC CLOCK 配置 ADC CLOCK



Module Clock

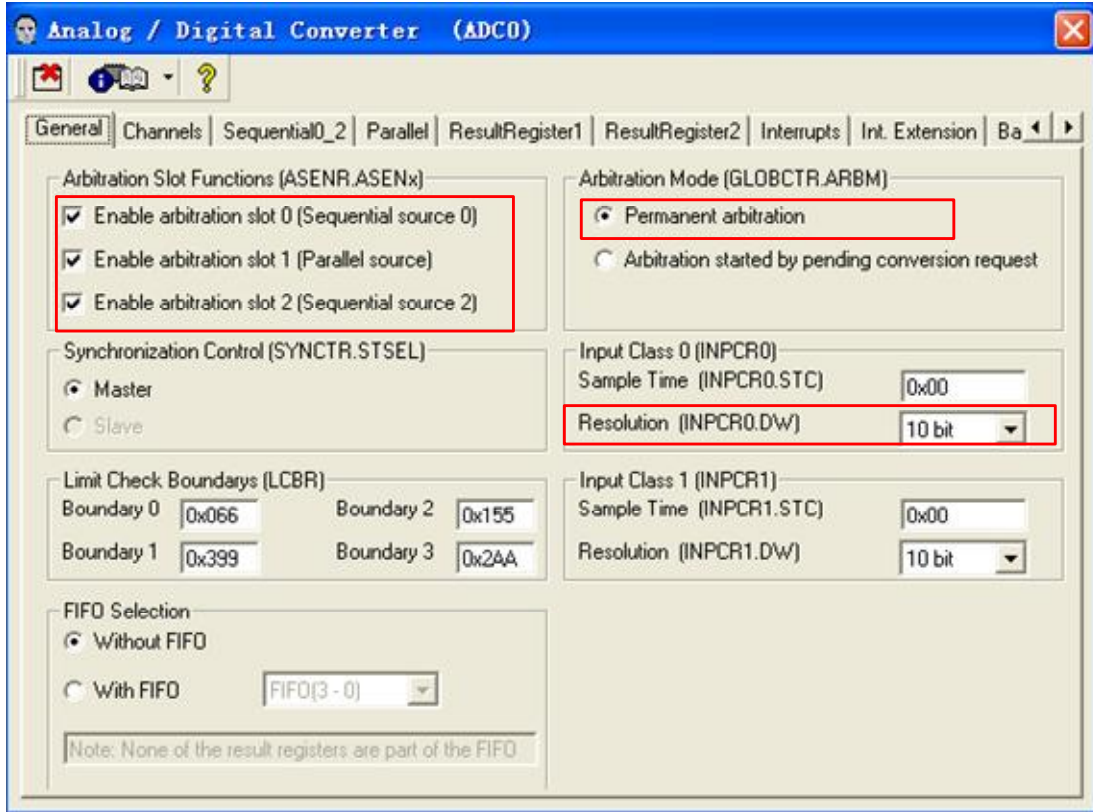


点击 ADC0 配置 ADC0



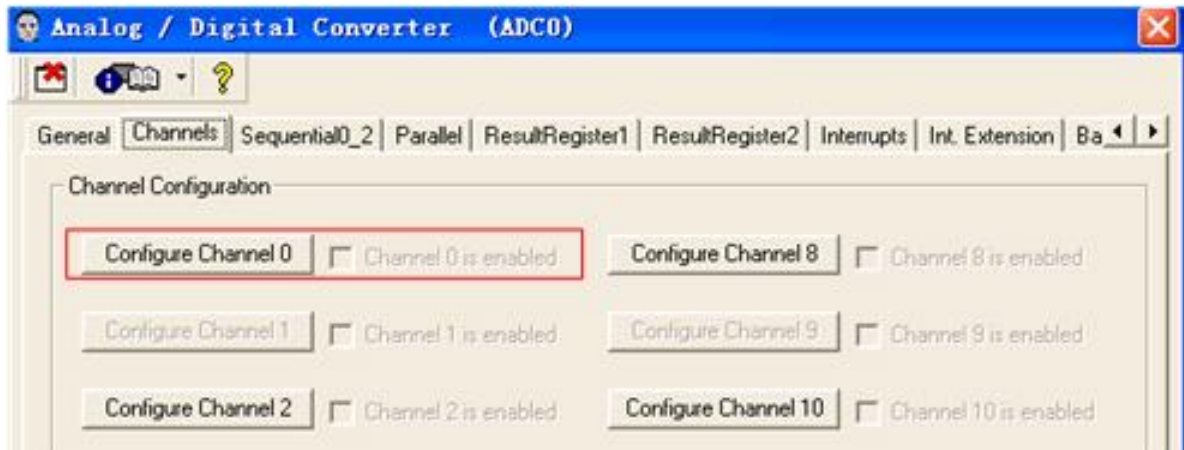


General

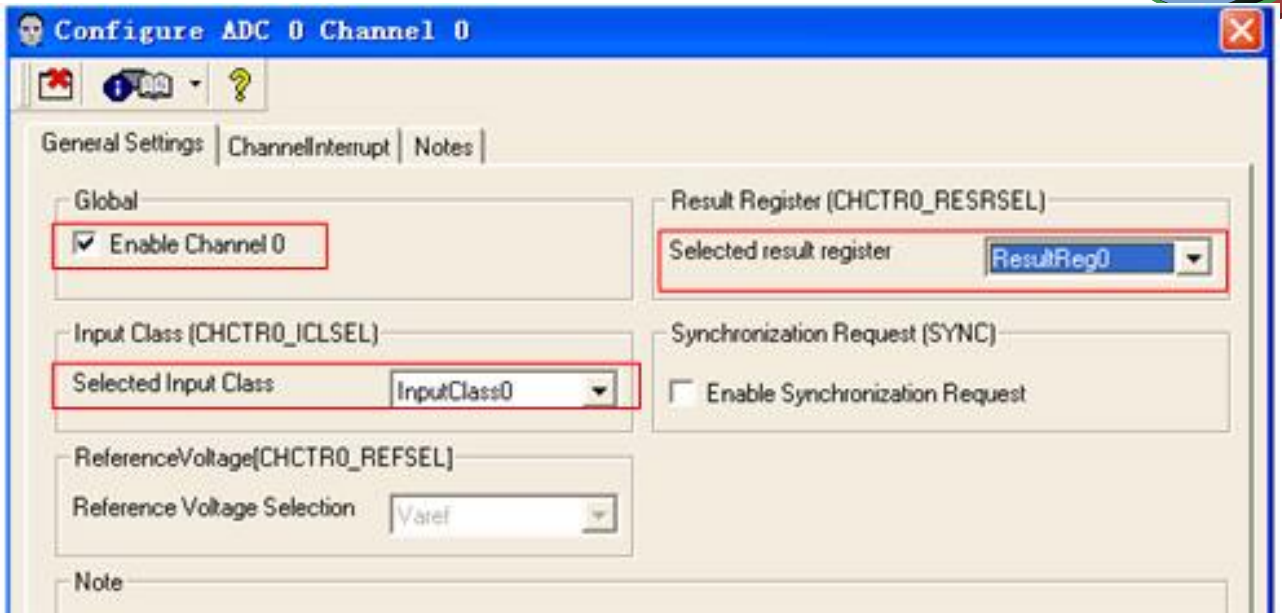


转换通道设置

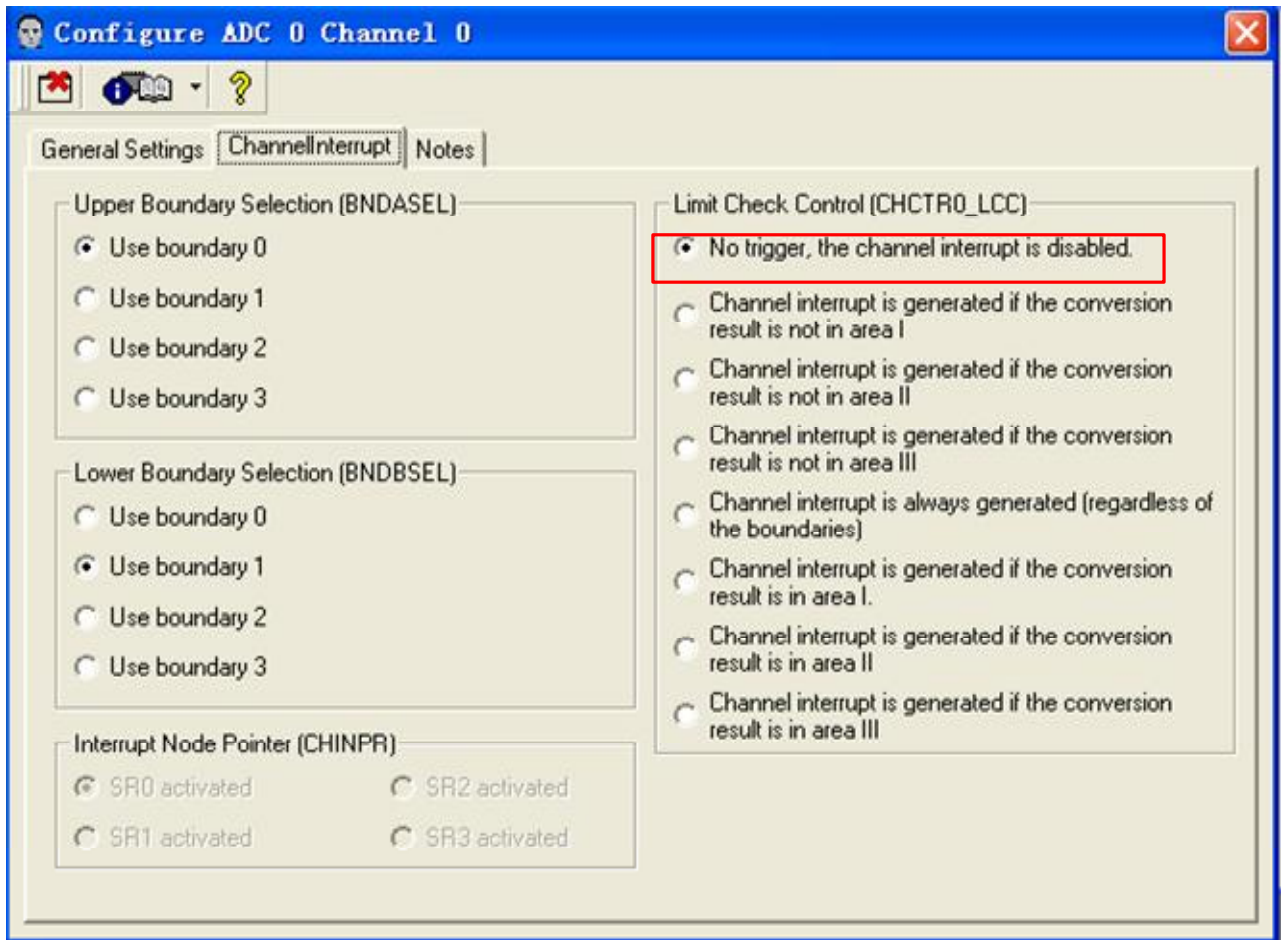
点击 Configure Channel 0



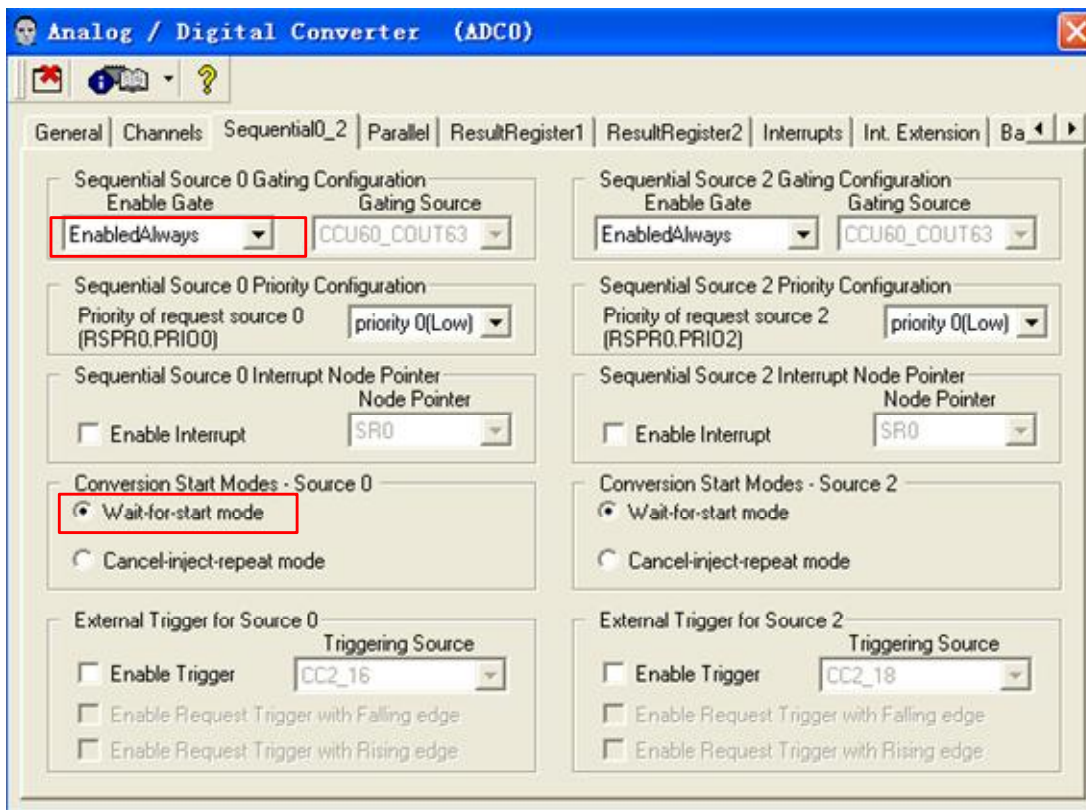
General Settings



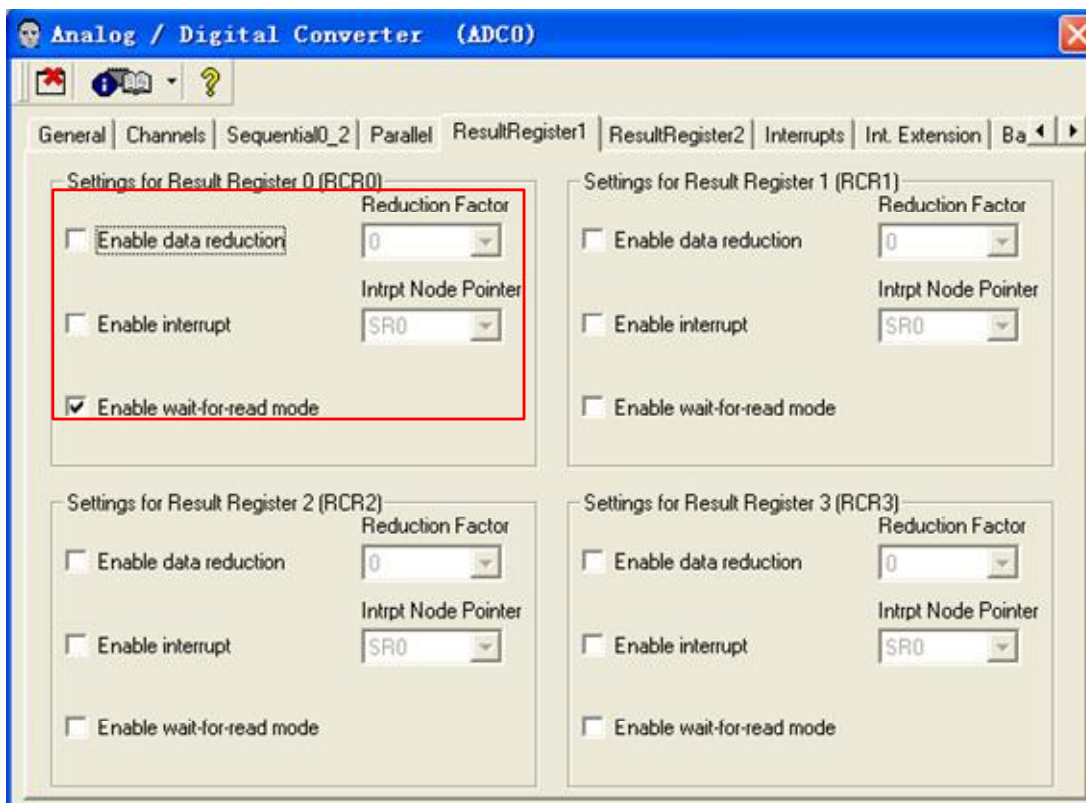
Channel Interrupt



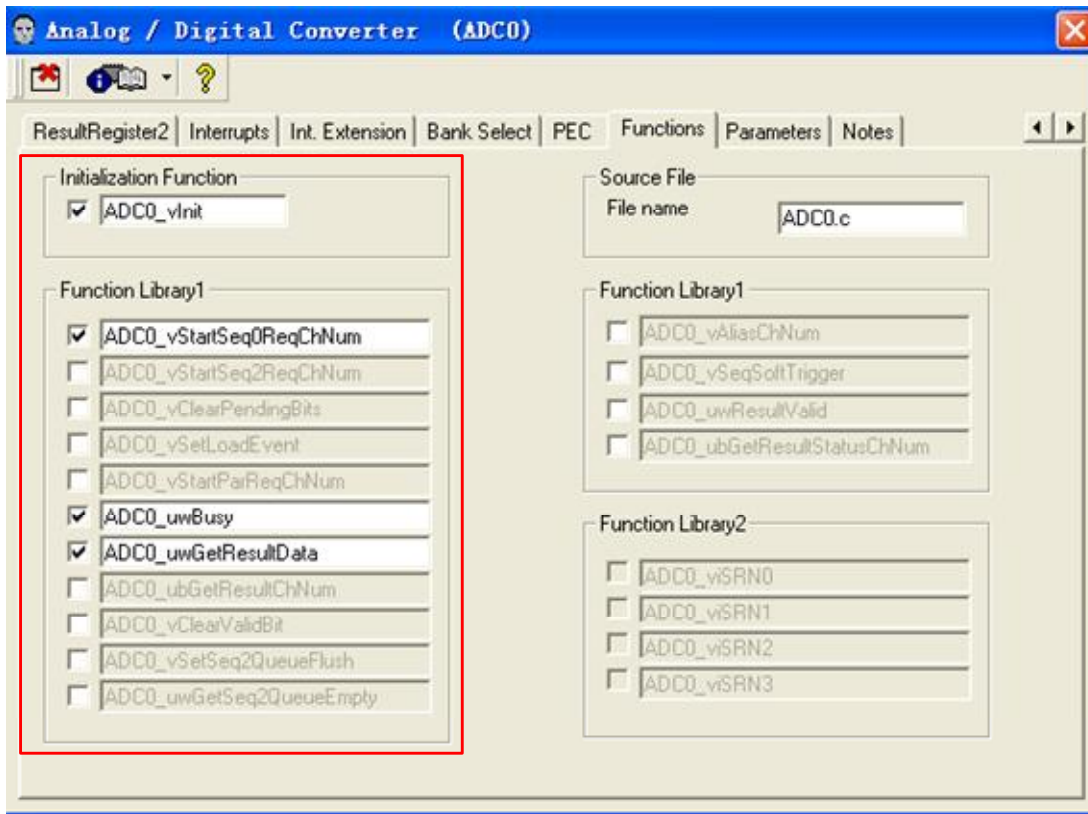
请求源 0 配置



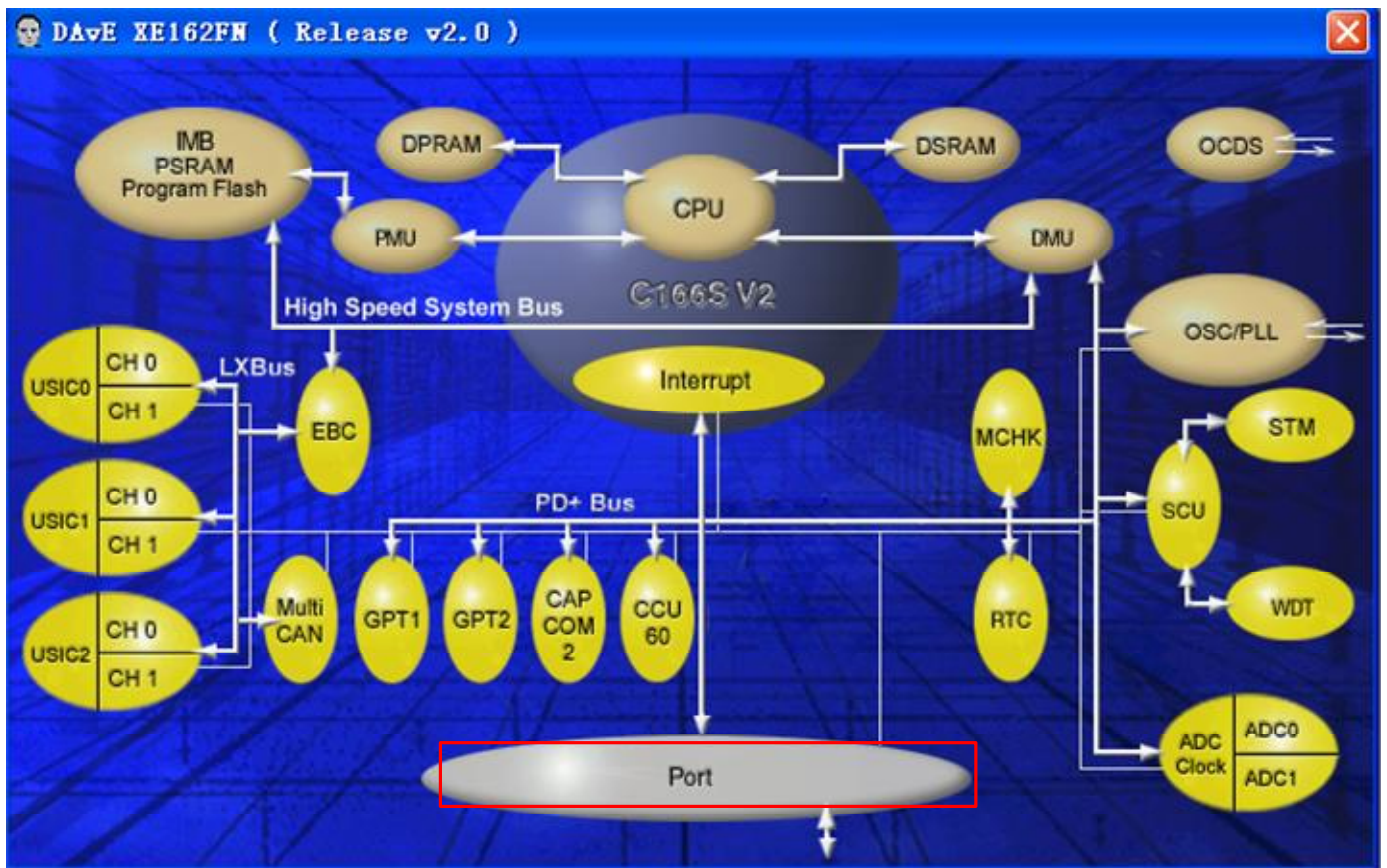
结果寄存器配置



选择需要 DAVE 生成的函数

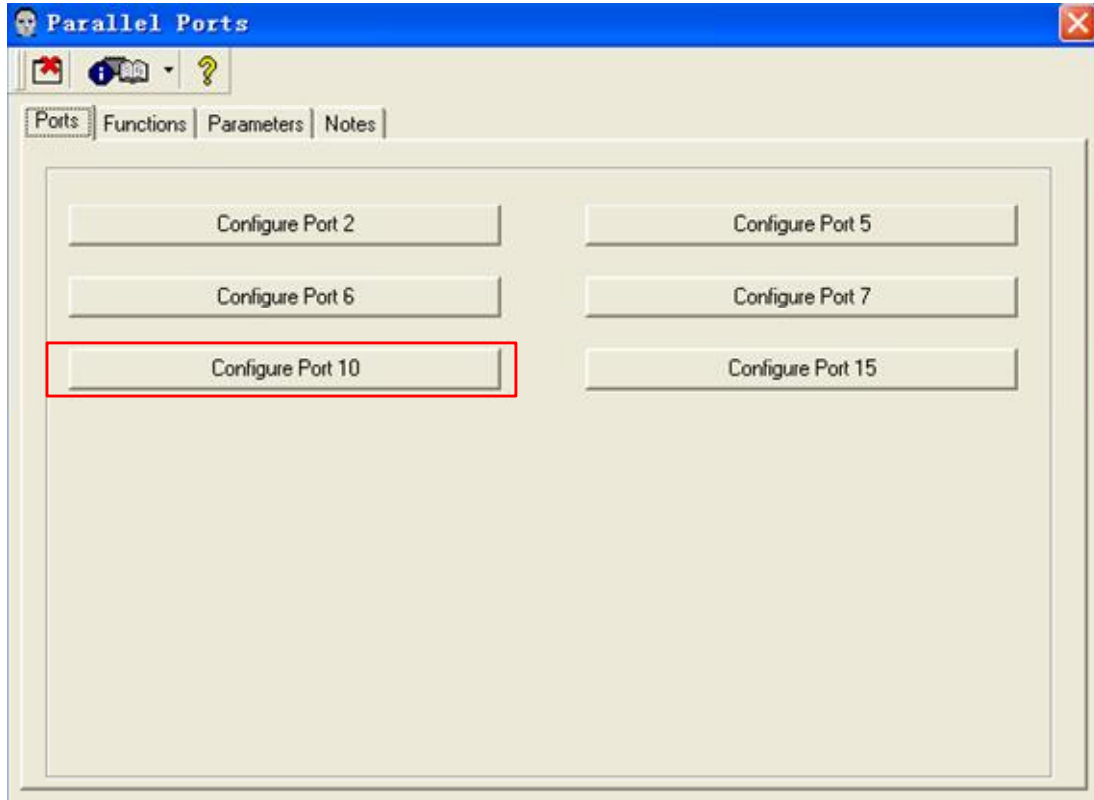


5. 关闭 ADC 配置模块，点击 port 模块，设置 I/O 口

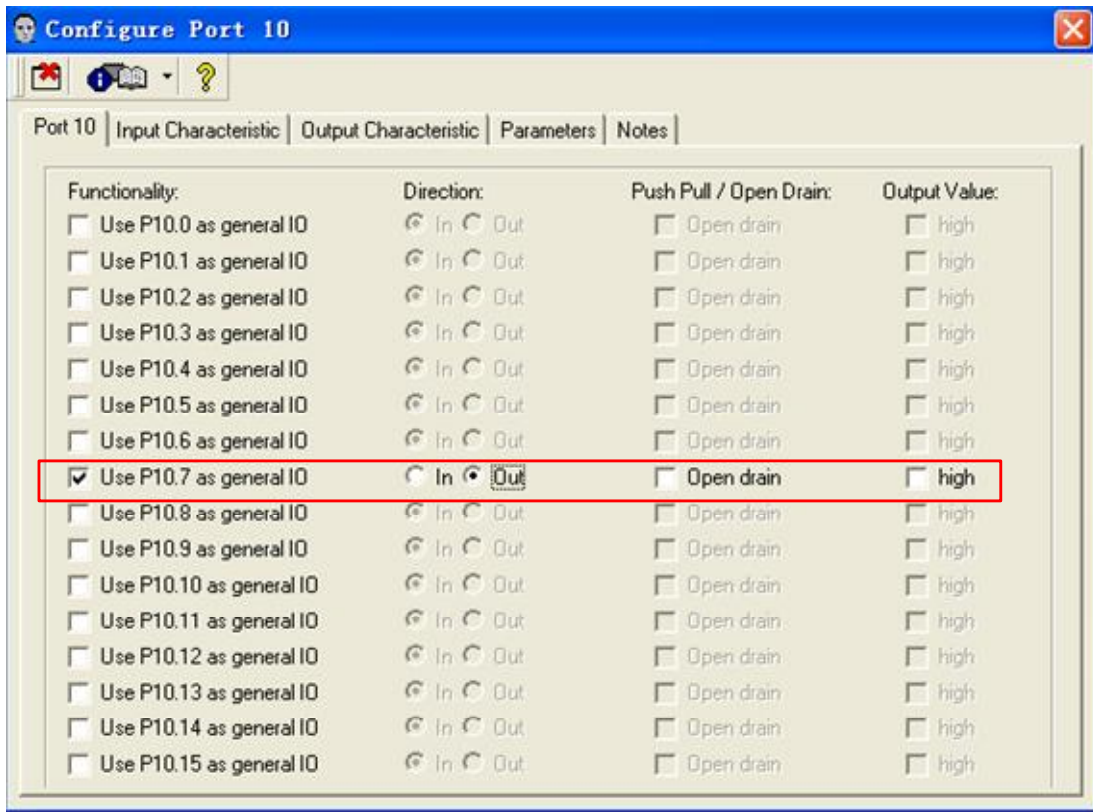




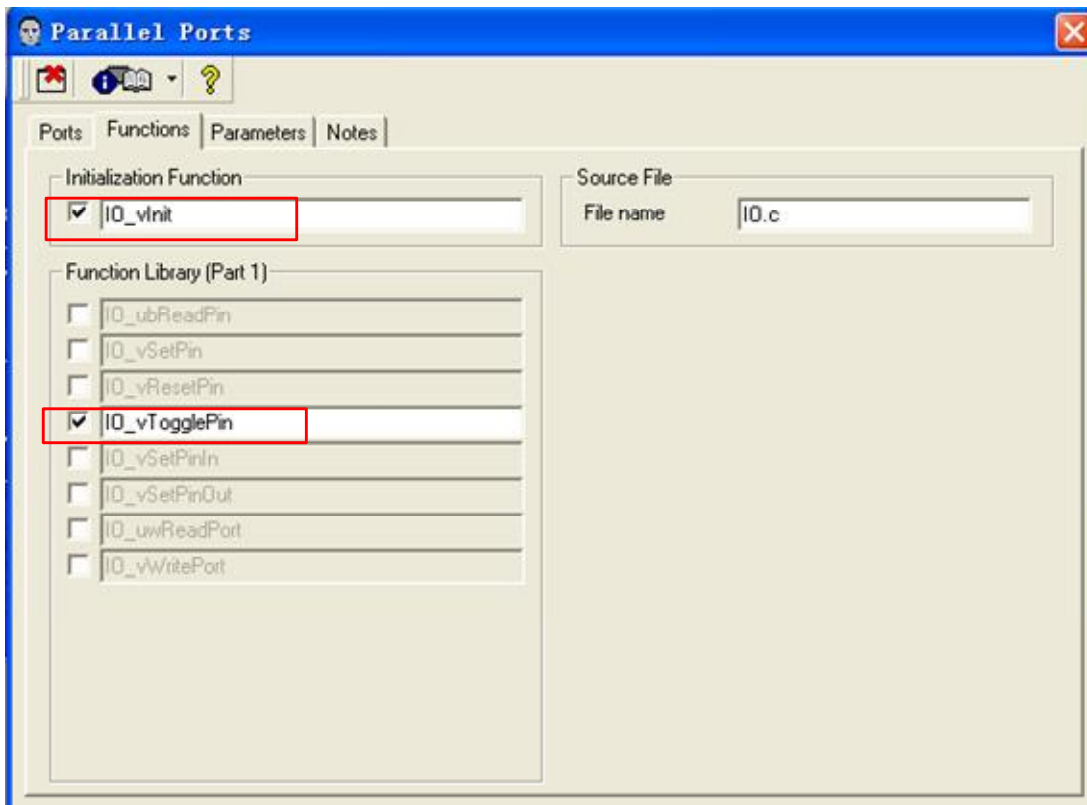
点击 Configure Port 10 因为: 最小系统板的 P10 口跟 LED 灯相连, 这里将其中之一配置为指示灯使用。



选择 Use 10.7 Port as general I/O, 输出选择 out。





选择需要 DAVE 生成的函数



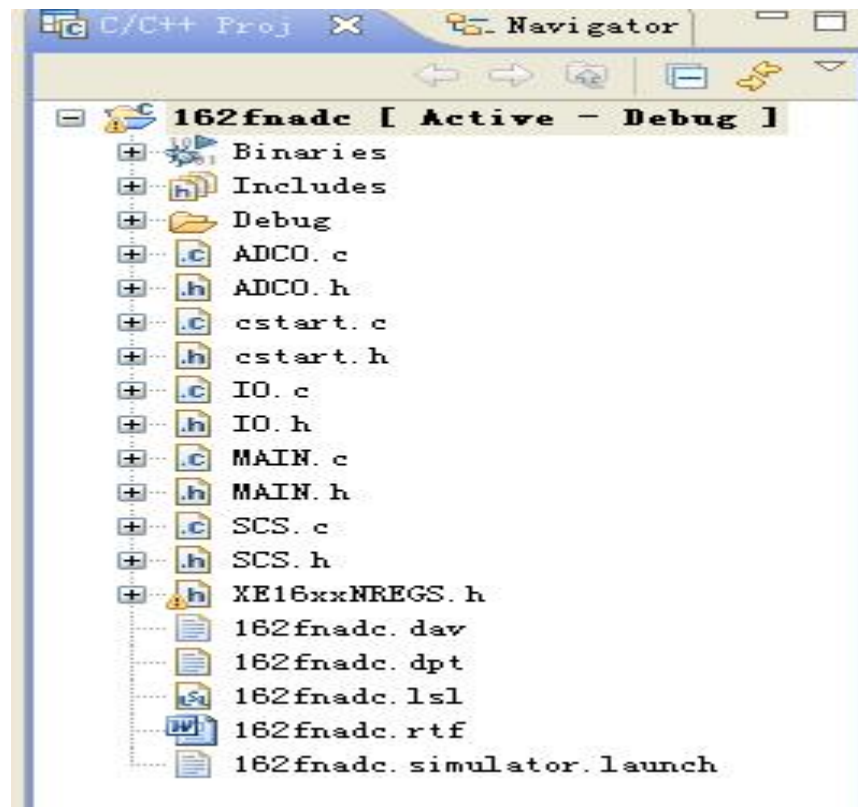
6. 利用 DAVE 生成代码



点击  保存为【如: 162fnadc.dav】, 注意将文件放在所建文件夹中。点击  自动生成代码, 生成的代码即包括前面所选择的函数。



7. 如此“ .dav ”工程文件创建完毕, 同时 Tasking VX-tool set 中的工程自动更新如下。



三、修改用户代码

注意: 在包含自定义头文件.h, 自定义变量, 引用外部变量, 函数时, 要放在正确的// USER

CODE BEGIN (MAIN_Header,3) 和 // USER CODE END 之间。如头文件放在:

//*****



**

```
// @Project Includes
```

```
//*****
```

**

```
// USER CODE BEGIN (MAIN_Header, 2)
```

```
#include "SPWM_VVVF.h"
```

```
// USER CODE END
```

之间。

B. 自定义变量放在:

```
//*****
```

**

```
// @Global Variables
```

```
//*****
```

**

```
// USER CODE BEGIN (MAIN_General, 7)
```

```
long i, j, uwADCResult, CC16value, temp;
```

```
// USER CODE END
```

C. 引用外部的变量放在:

```
//*****
```

**

```
// @Imported Global Variables
```



```
//*****
```

```
**
```

```
// USER CODE BEGIN (CC2_General,6)
```

```
extern long CC16value;
```

```
// USER CODE END
```

1. **注意这些细节后**，开始添加用户代码（XE164F 与 XE164FM 添加代码基本相同）

在 main 函数中添加下列代码（在 main 函数的末尾处）

```
// USER CODE BEGIN (Main,2)
```

```
uword i,j,uwADCResult;
```

```
// USER CODE END
```

```
MAIN_vInit();
```

```
while(1)
```

```
{
```

```
// USER CODE BEGIN (Main,4)
```

```
ADC0_vStartSeq0ReqChNum(0, 0, 0,0);
```

```
while(ADC0_uwBusy());
```

```
uwADCResult=ADC0_uwGetResultData(RESULT_REG_0);
```

```
I0_vTogglePin(I0_P10_7);
```

```
for(i=0; i<uwADCResult; i++)
```



```
for(j =0; j <10000; j ++);
```

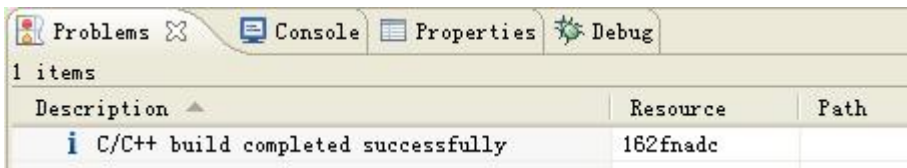
```
// USER CODE END
```

```
}
```



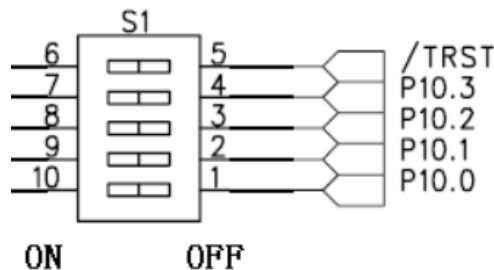
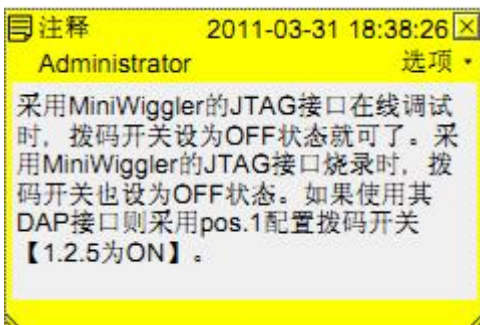
2. 点击图标 进行编译连接。如有错误进行更改，直到出现 0 errors。出现的

3615 warnings，大可不管。 0 Problems，如下图：




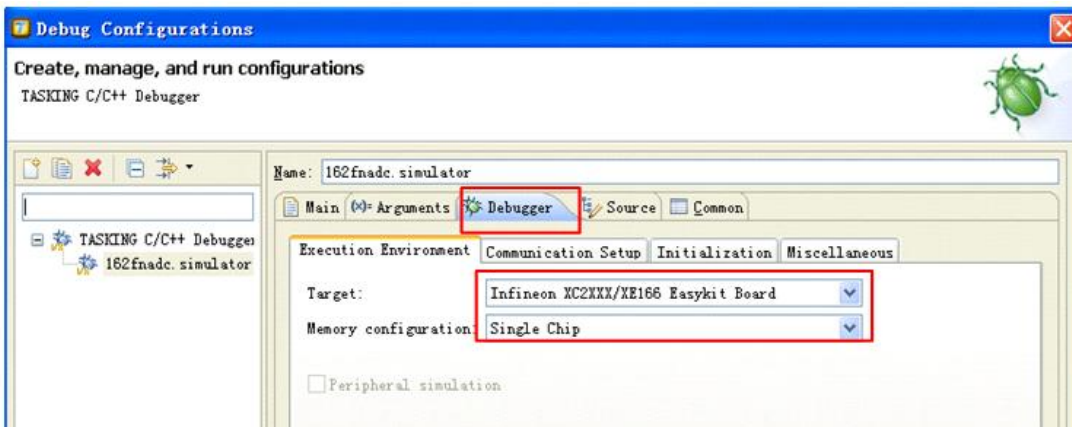
四、在线调试

1. 设置调试板的拨码开关，连接好调试板；



2. 点击  进入调试；


3. 设置调试参数；在  的下拉菜单中的 Debug Configurations—> Debugger；

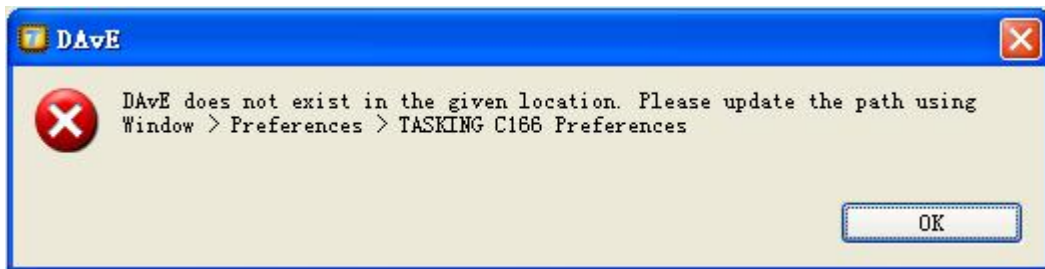




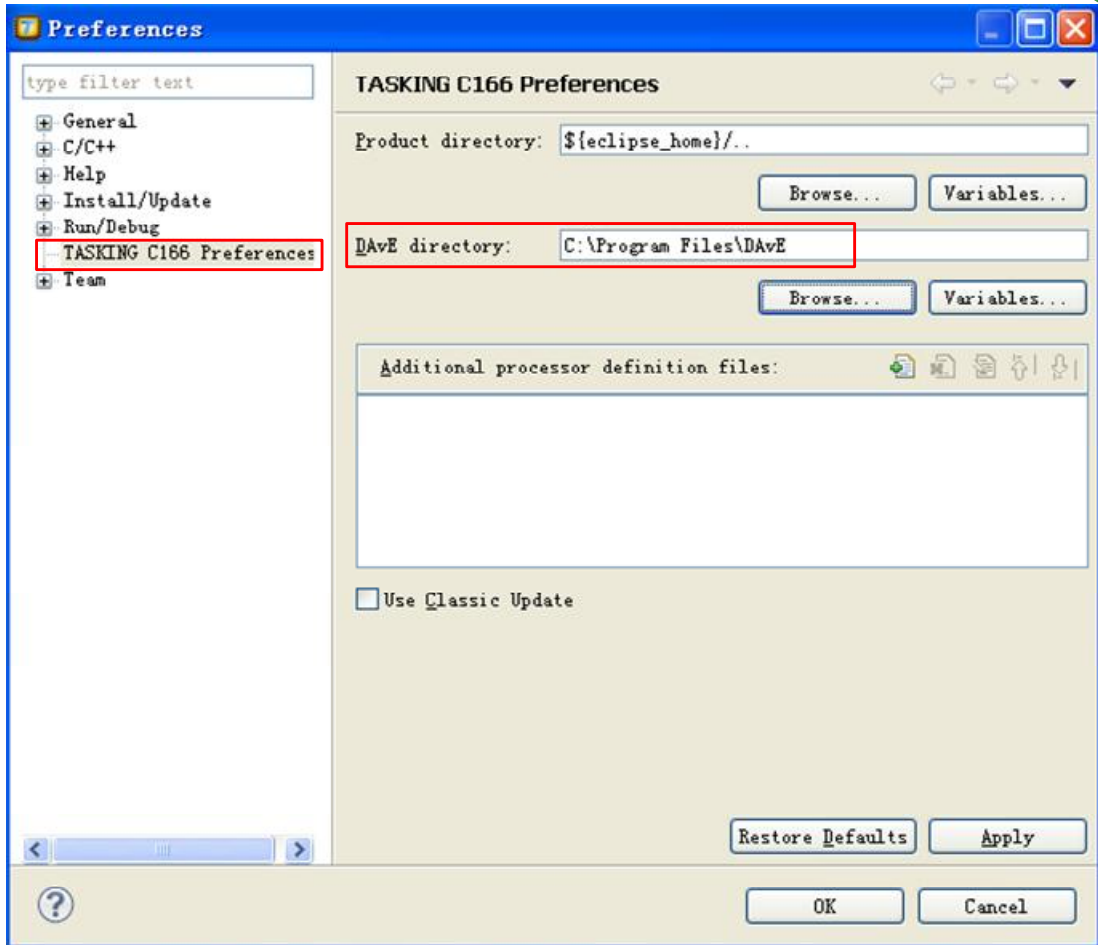
4. 点击 Debug ->或者

五、DAVE 更新配置


1. 首次点击  图标, 出现如下提示:



2. 需要配置 DAVE 程序的运行路径: Window → Preferences → TASKING C166 Preferences;



3. OK

4. 再次点击  时, 便可以重新配置你的工程了。

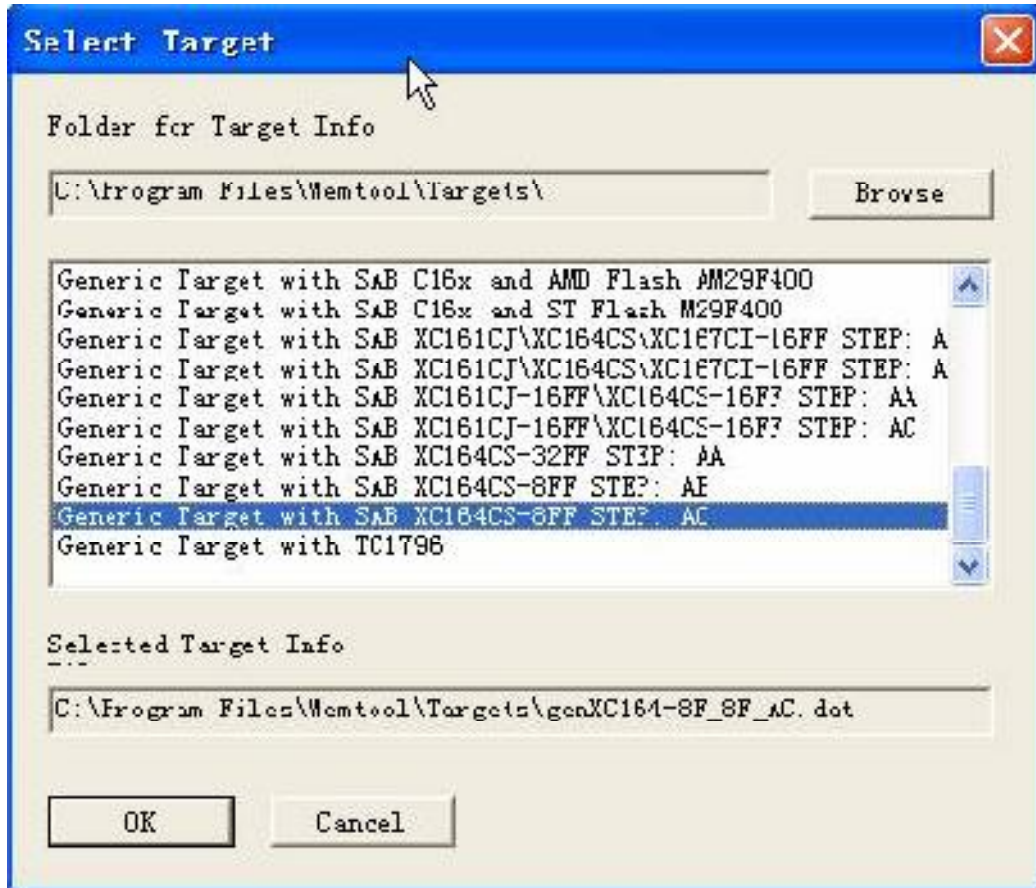
Memtool 是Infineon提供的编程工具, 可以对C500/C800/C166/XC166/TriCore等所有现有产品进行编程, 包括片内存储器编程和外部Flash编程。-memtool Infineon is the programming tools, C500/C800/C166/XC166/TriCore be right all the existing products Programming, including program- chip memory and external Flash programming.

👉 Memtool 3.3.1版本下载步骤详解

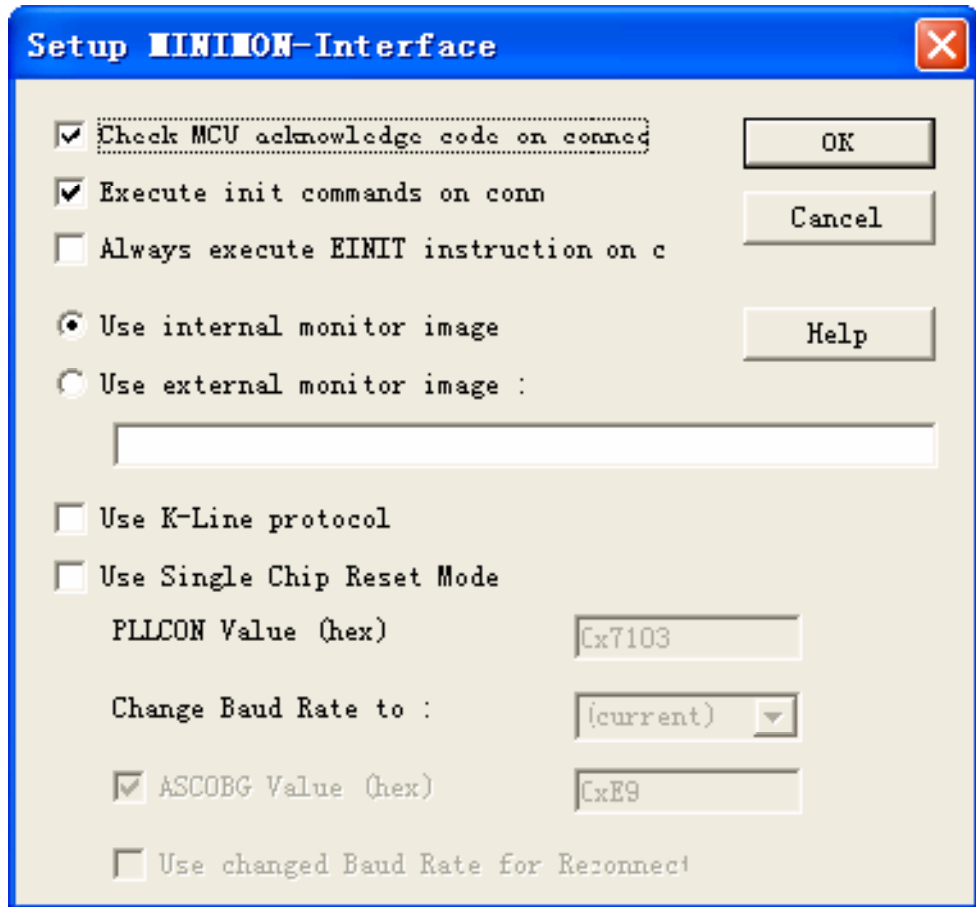
下载并安装最新版的Memtool 3.3.1版本, 这个版本可以支持XC164CS-8F, 由于我们当时没有XC164CS-16F, 所以我们评估板上用得是XC164CS-8F单片机



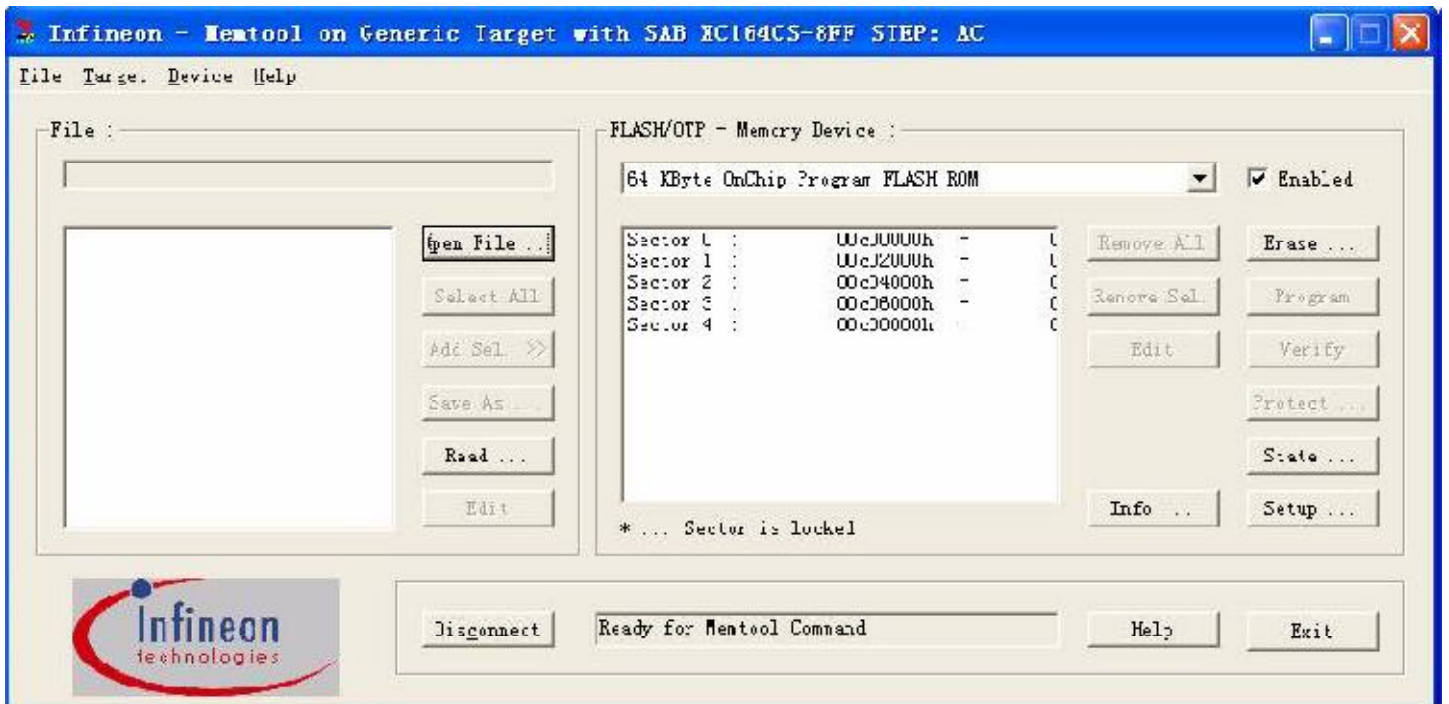
与C16x单片机一样，还需要Select Target，选择XC164CS—8FF STEP:AC，事实上我们评估板上用得芯片是STEP:AD，但是在3.3.1版本中没有。幸亏实践证明：用STEP: AC也没有问题，如下图所示：



注意：由于memtool默认是用External Monitor，而XC16x只需要选择internal monitor就可以了。所以选择菜单：Target—Setup Target Interface，出现下图所示的图片，选择Use internal monitor image。



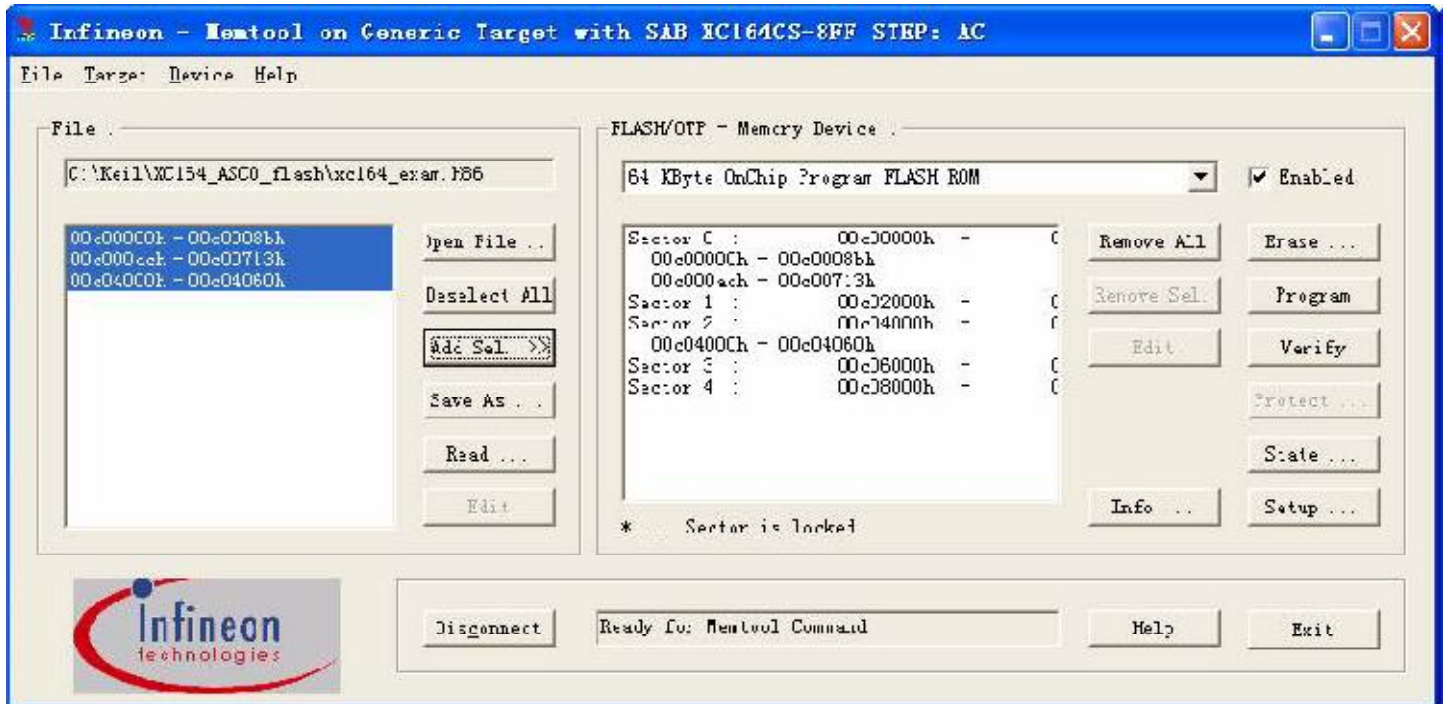
这时就可以connect了，出现下图：



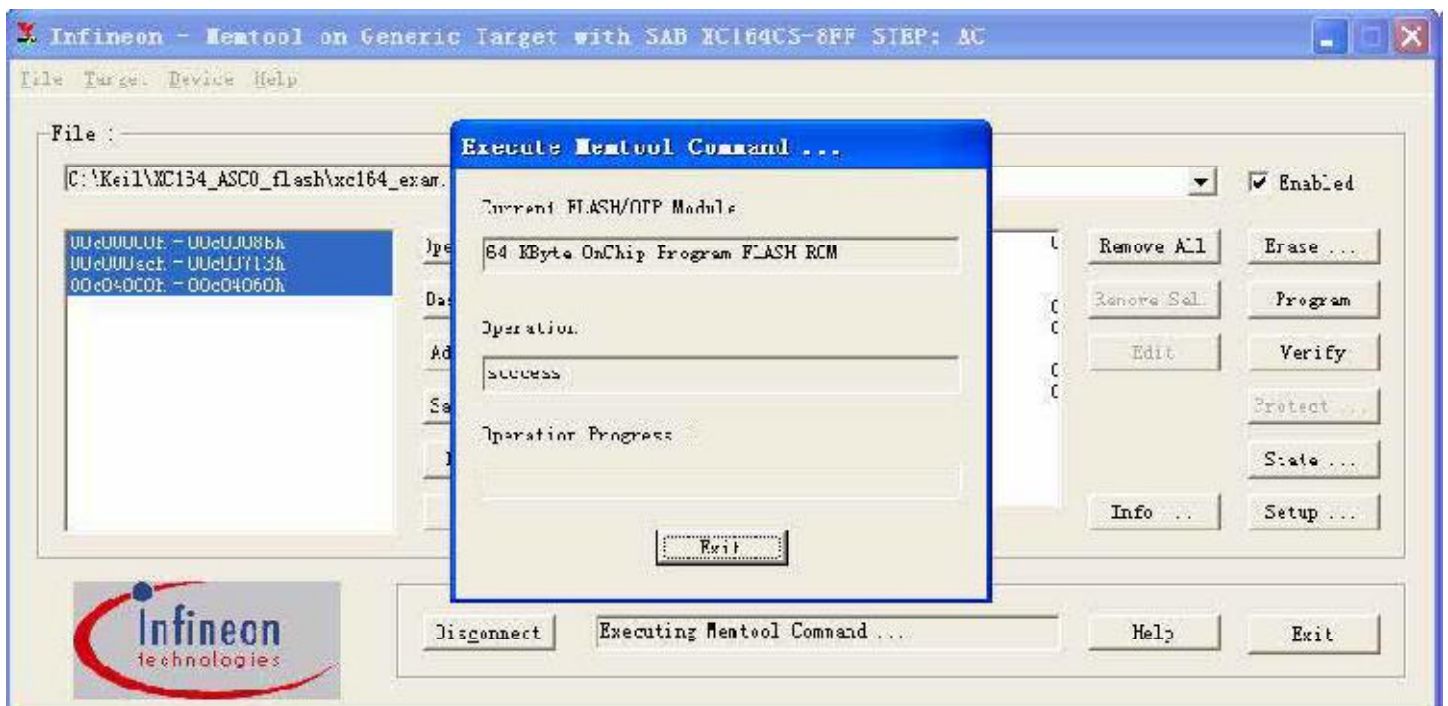
点击上图中的Open File，选择自己想要下载的程序；



随后点击Select All，再选择Add Sel. 》，这时出现下图



点击上图中的Program，出现下图所示的执行结果。图中的Operation/success表示成功。



当您把附件中的代码下载到Flash中以后，如果需要运行程序，请把S4配置成(OFF OFF OFFOFF)。把S2的第4个拨码配置成OFF，否则单片机复位后进入Bootload模式。然后复位，



就可以运行了。运行的效果如图:



在“字符串输入框”输入‘q’就从Hello World 变成Hello Tsinghua,

在Hello

World的时候, 后面的数字是递增, 在Hello Tsinghua的时候是递减的。

当在“字符串输入框”输入非‘q’字符, 就从Hello Tsinghua变成Hello

World,

如下图所示:



👉 英飞凌 DAP mini Wi ggler 的安装使用





DAP mini Wiggler 简介

DAP Mini Wiggler USB→JTAG调试器



- USB接口，可方便地与上位机连接
- 符合JTAG IEEE 1149.1规范，同时支持英飞凌XC800 / XC166 / XE166 / XC2000 / TriCore系列
- 可方便地嵌入Keil uVision、TASKING和Hitex等多种集成开发调试环境
- DAP Mini Wiggler可同时支持JTAG和DAP协议的调试和下载



经济划算的高性能调试工具

mini Wiggler 是英飞凌面向未来的经济划算的高性能调试工具。在主机侧，它具备一个USB接口。每台计算机都具备USB接口。在器件侧，则可通过英飞凌10-针DAP或16-针OCDSL1接口，进行通信。mini Wiggler 经专门设计，可配合英飞凌调试访问软件（DAS）使用。

最新版本的DAS可在 www.infineon.com/das 下载。



应用:

- 调试,
- Flash 烧录.

特性:

- 兼容英飞凌 DAP 和 SPD
- 兼容 JTAG/IEEE 1149.1
- 时钟速率最高达 30 MHz (可编程)
- 所有信号均为 5.5V, 可下调至 1.65V
- USB 2.0 (高速)
- 经认证的驱动程序, 适用于微软 Windows 2000、XP 和 VISTA 操作系统
- USB、JTAG 和 DAP/SPD 热插拔和拔下
- DAP1/SPD 引脚上提供 UART 功能, 以支持 BSL 和 BMI 烧录
- 3 个板上状态 LED 指示灯
- 支持 OCDS1 16-针和 DAP 10-针连接器

可支持的器件:

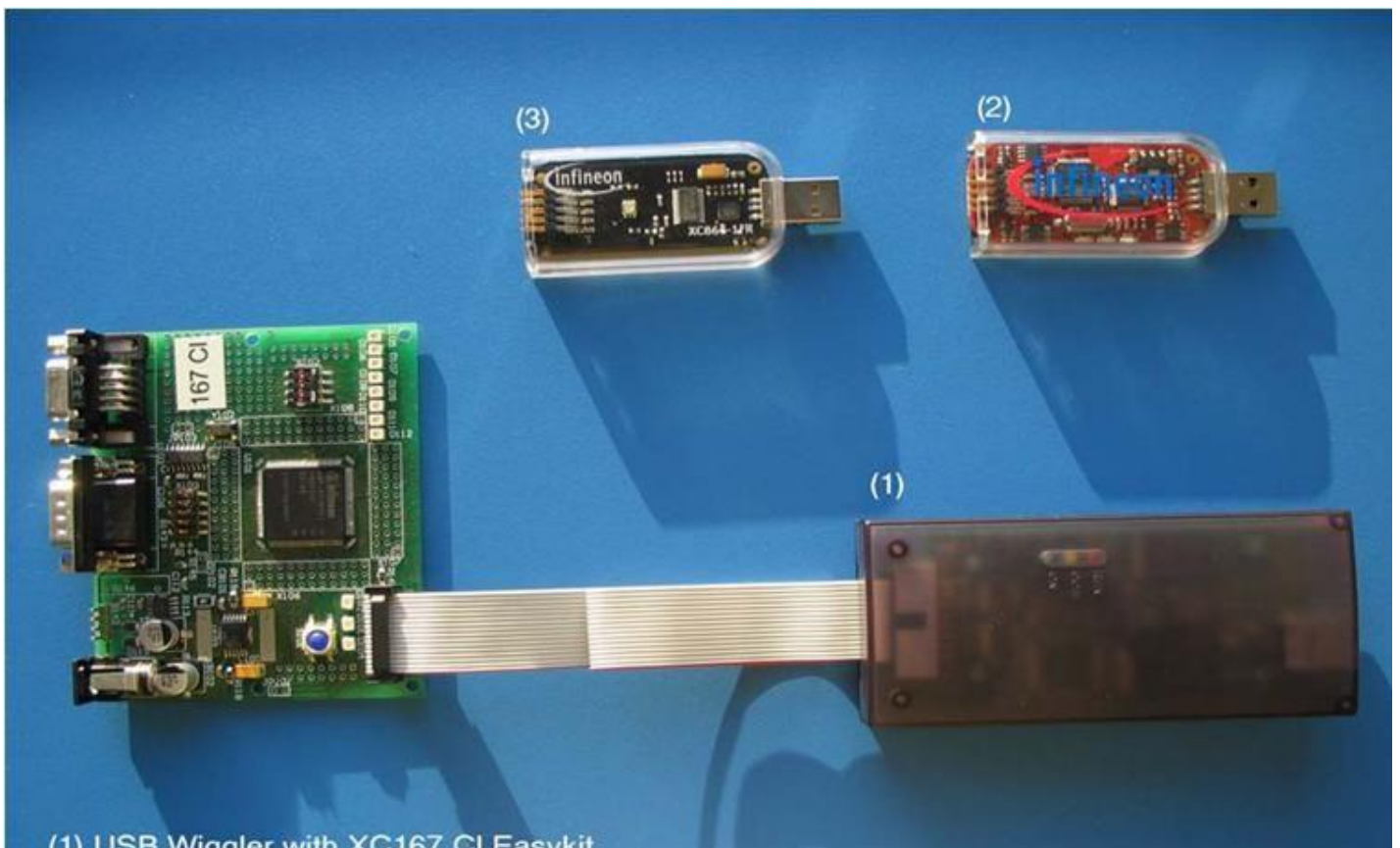
- 英飞凌 8、16 和 32 位衍生产品.

可支持的工具:

- ALTIUM/编译器
- DAVE™ Bench
- Hi TOP HI TEX
- KEIL uVision



单片机开发除了必要程序编写外同样也离不开下载器与仿真器。miniWiggler 是目前英飞凌单片机最流行的仿真器。



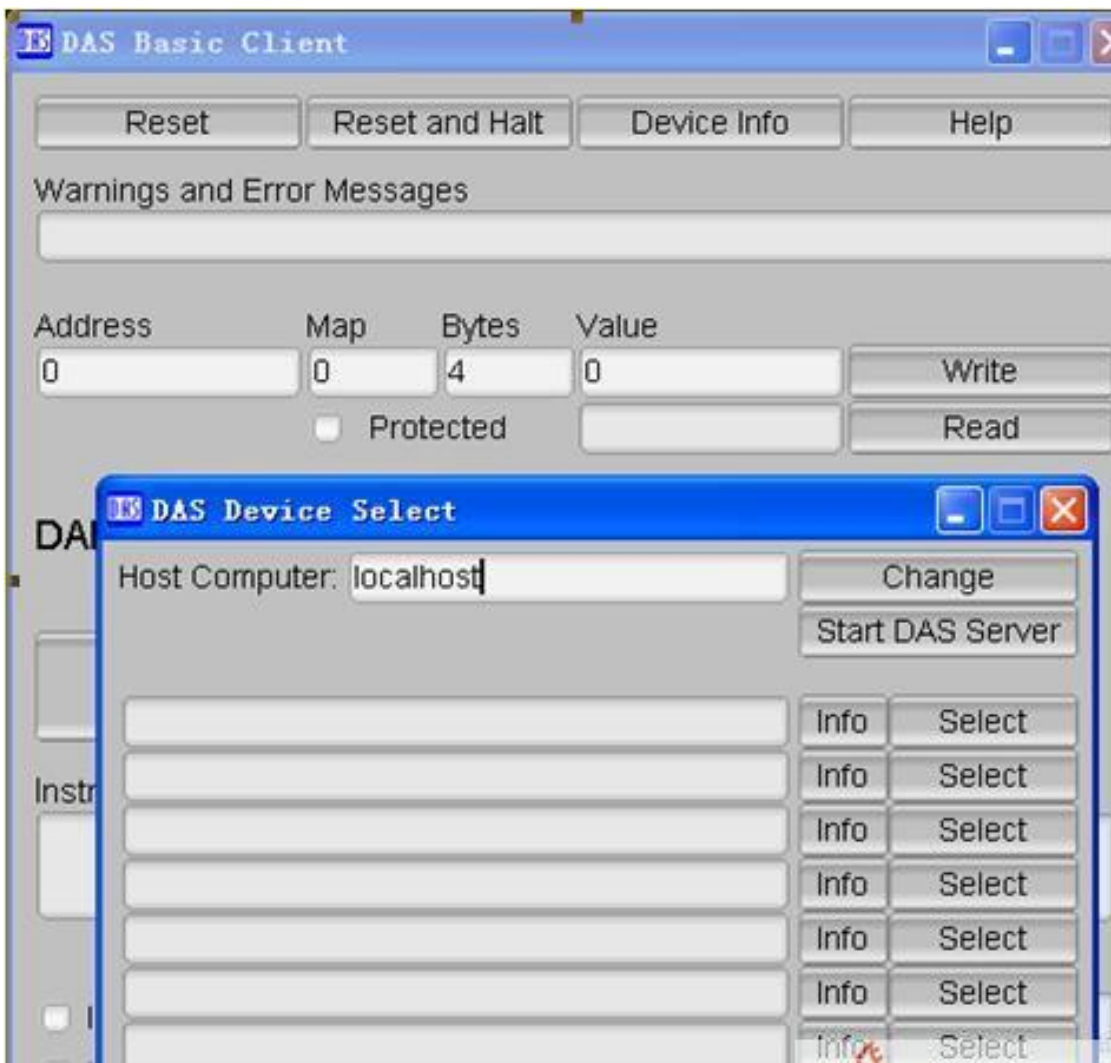


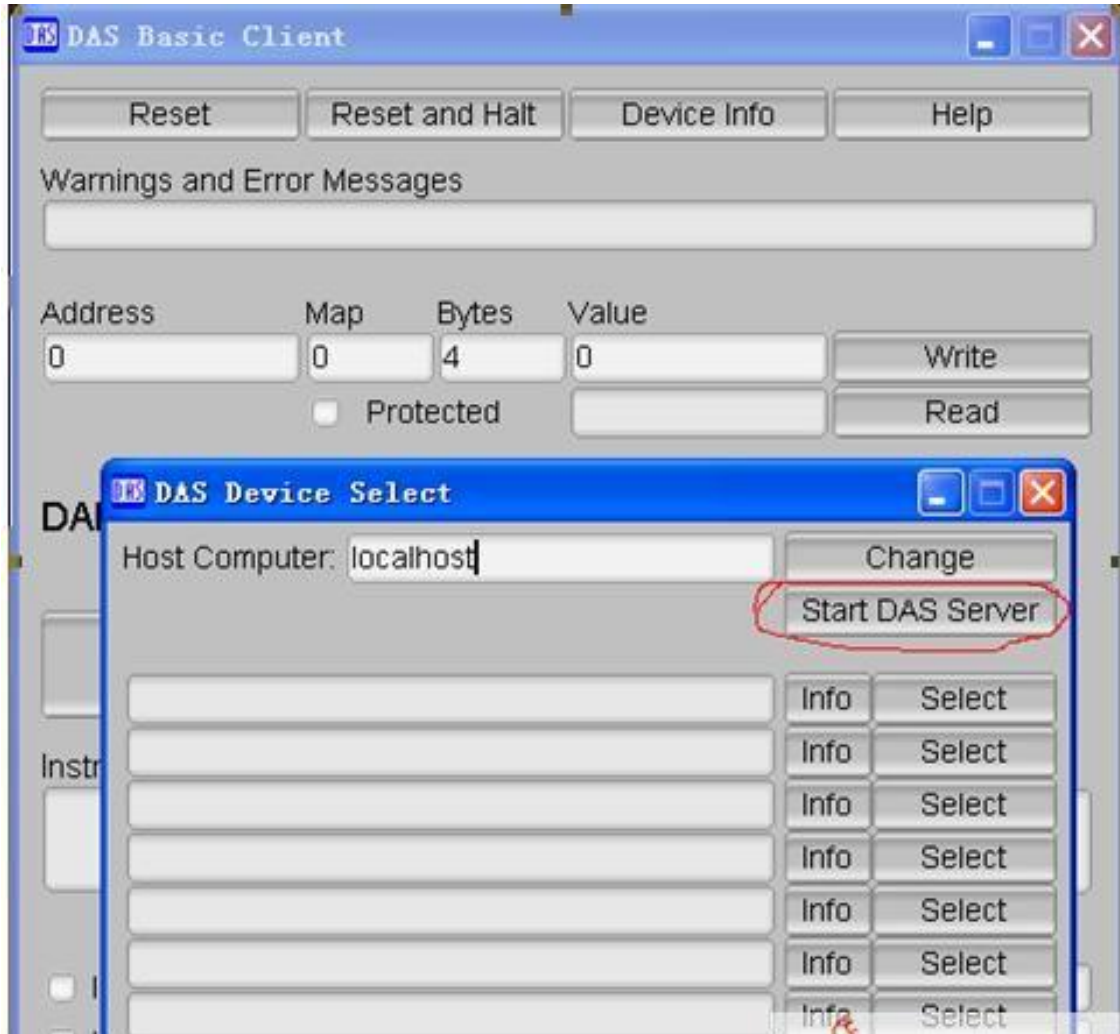
英飞凌 mini wi ggl er 安装使用步骤

- 1、安装最新版本的 DAS，从供应商或从以下链接下载 (www.infineon.com/miniwiangler)
- 2、把 mini wi ggl er 连接到电脑上的任意一个 USB 接口。电脑会自动识别这个新设备并自动安装相应的驱动程序。
- 3、把下载线连接到目标板上
- 4、启动您的调试工具选择 DAS 的服务器 udas 或以上的 USB 芯片的 JTAG。

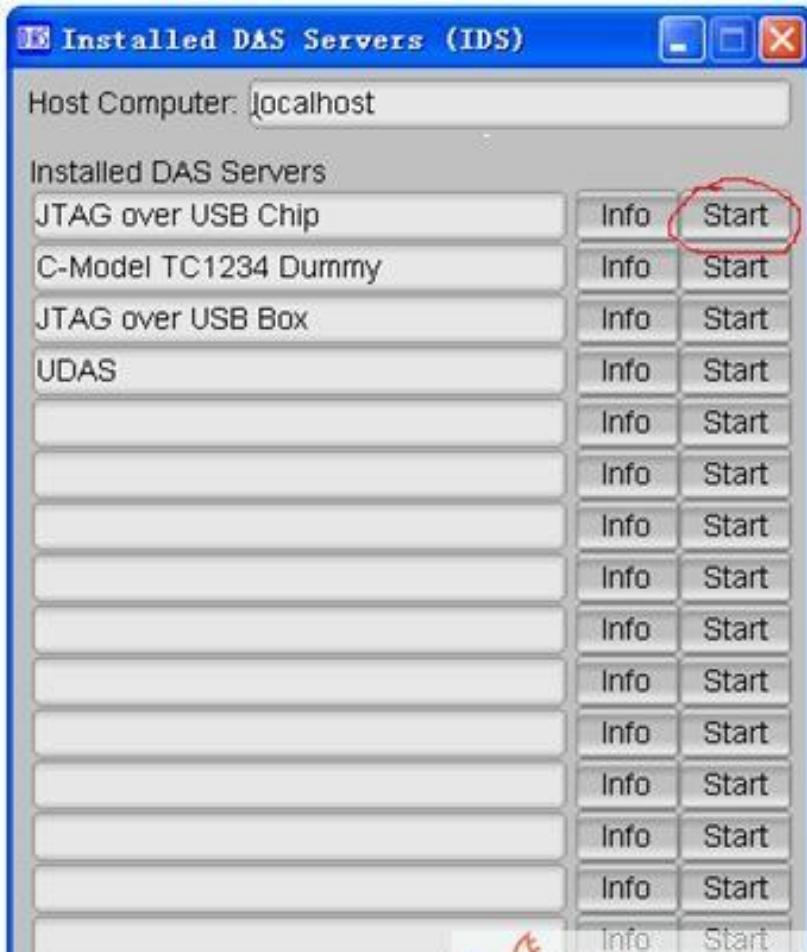
DAS 的服务器的使用:

- 1、启动调试工具选择 DAS



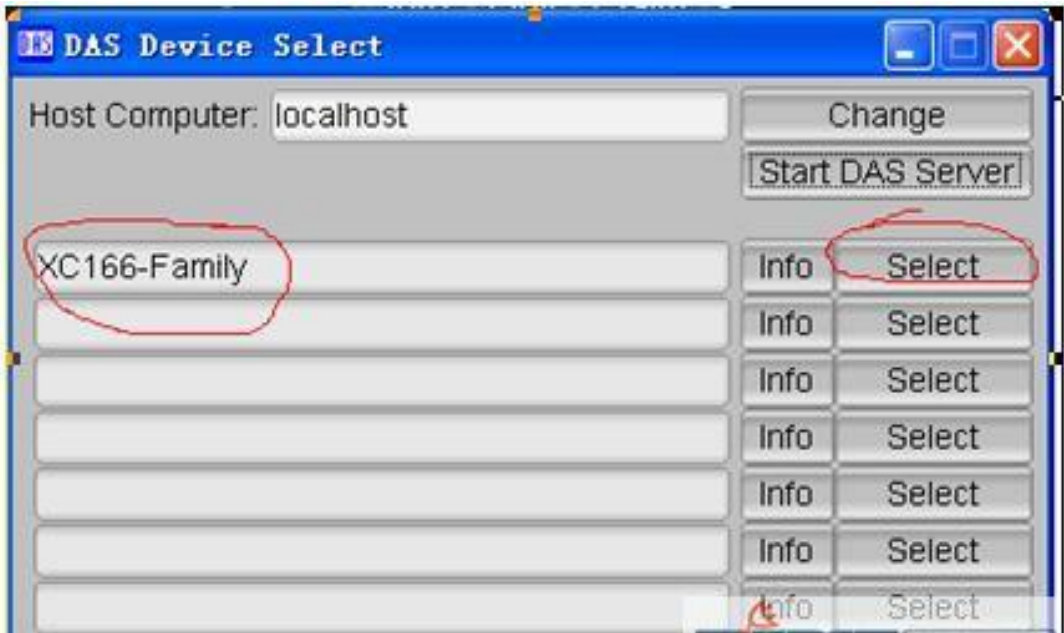


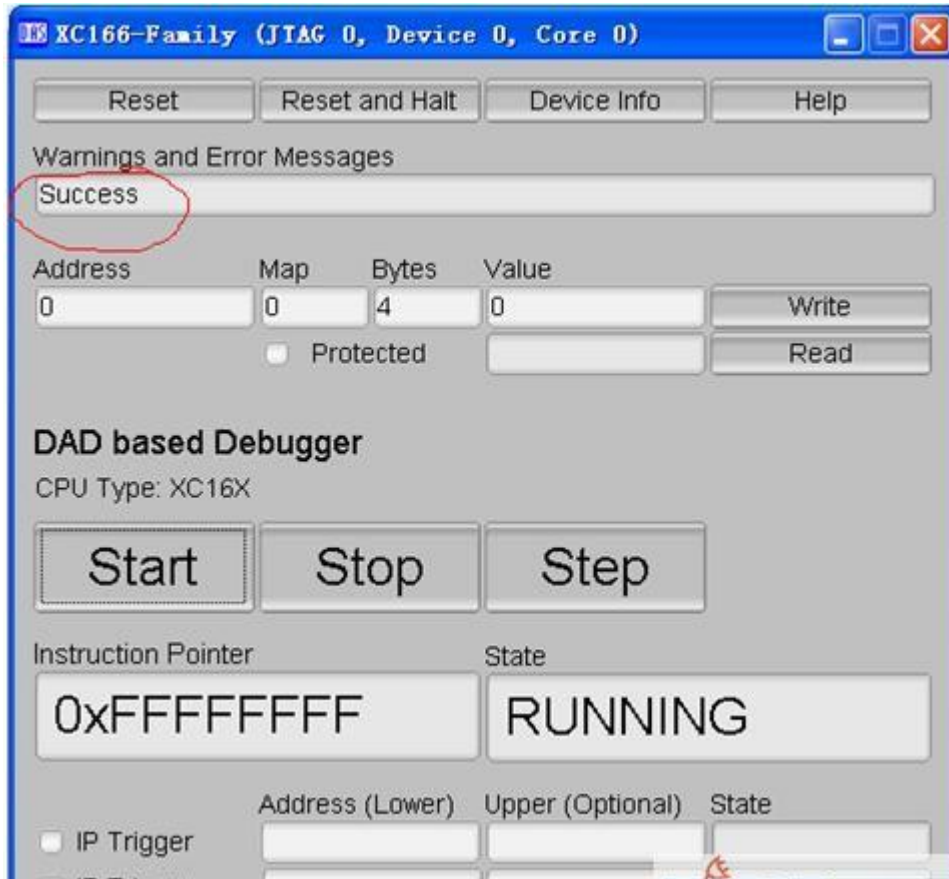
2



3、

4、最后出现 XC166-Family 表示安装成功

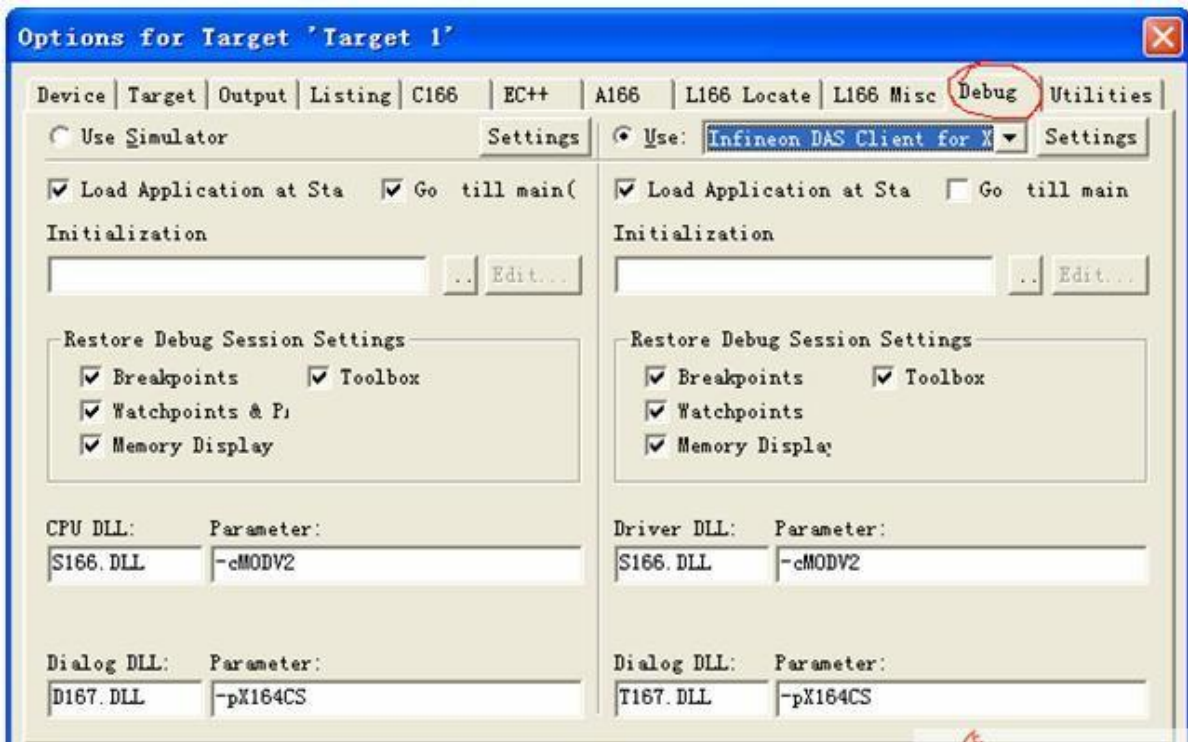




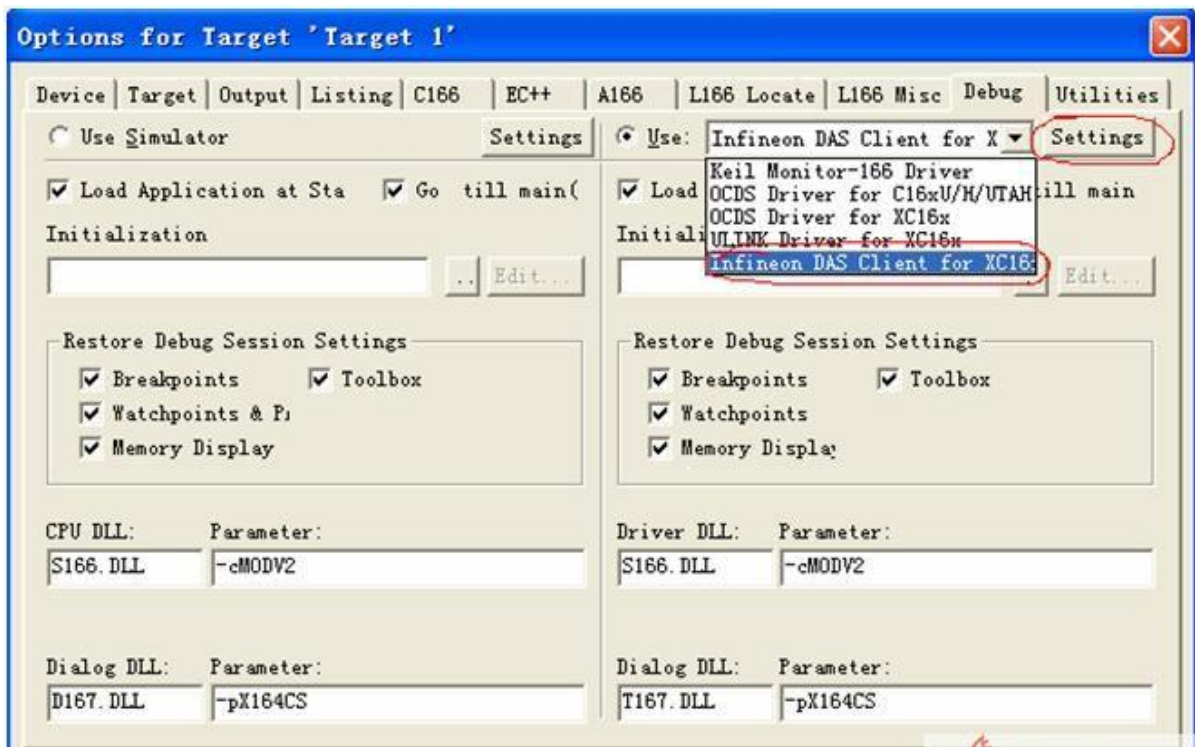
Keil C166 与 miniwiggler 的使用

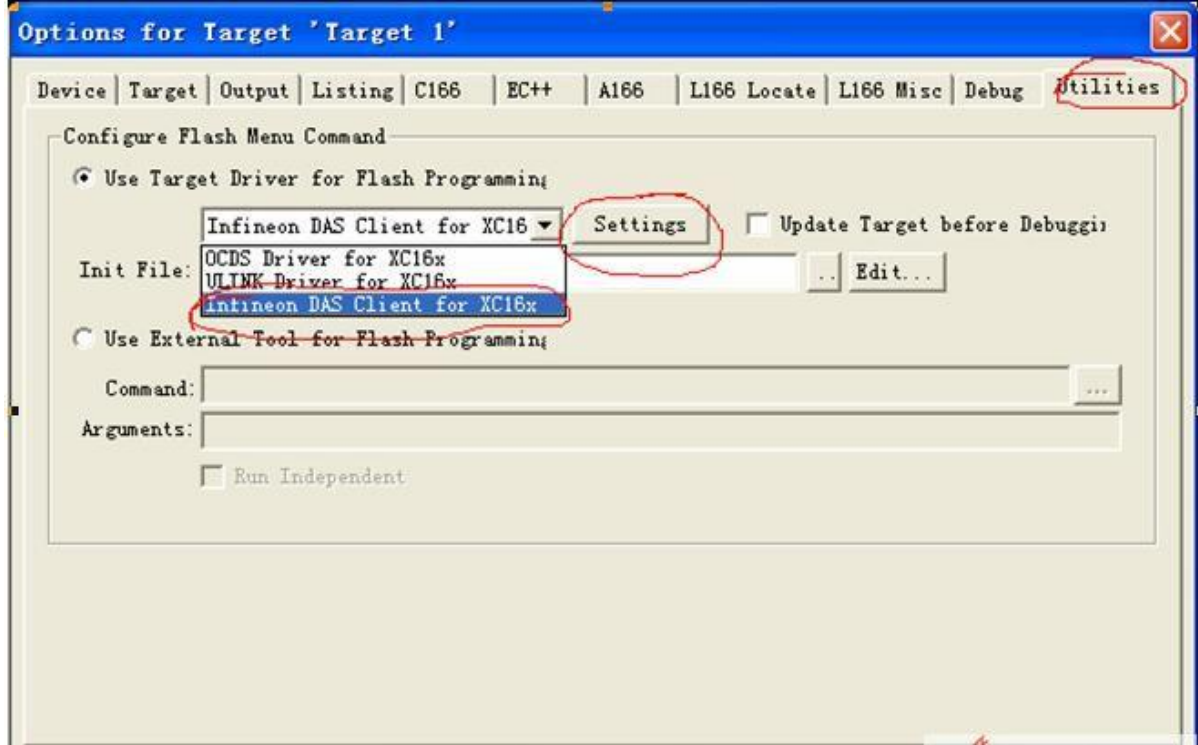
工程的详细设置

首先点击 Project 窗口中的 Target1 Project->Option for Target1 “Debug” 即出现对工程设置的对话框。

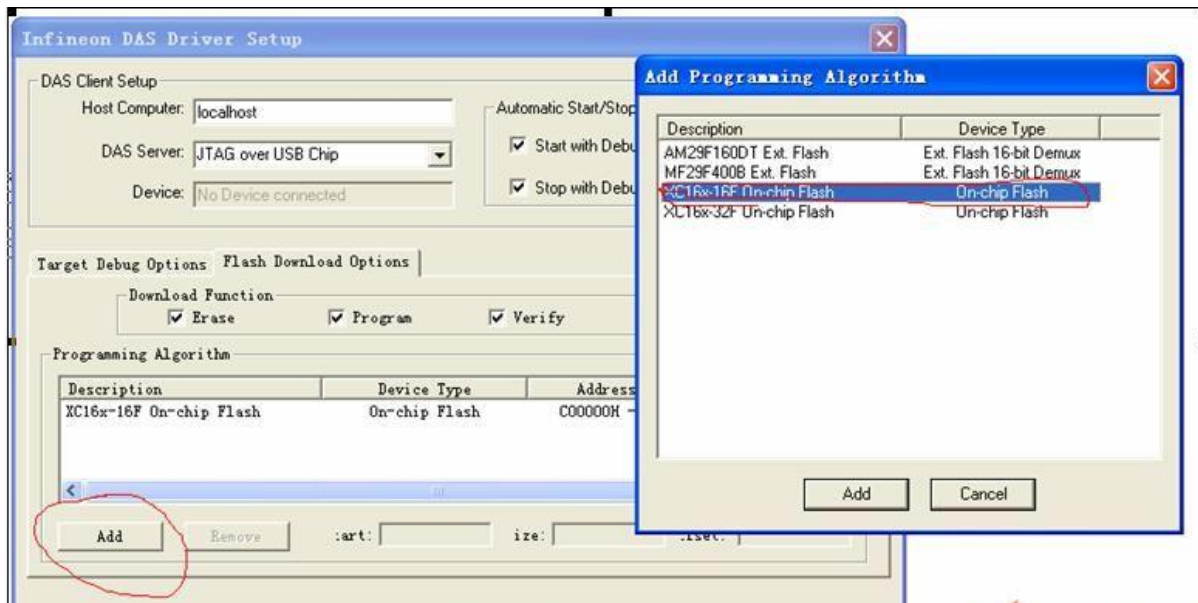


选择” Infineon DAS Client for XC166”





选择片内的 FLASH.



仿真与下载

以上即完成了工程的相关设置，接下来可以进行编译，连接。选择菜单 Project Build target 或单击图标对当前工程进行连接。编译过程中的信息将出现在输出窗口中的 Build 页，如果源程序中有语法错误，会有错误报告出现，单击该行会有相应的错误



报告出现。编译成功后提示获得*.hex 文件, 该文件可被编译器读入并写入芯片中, 同时还产生了一些其他相关文件可用于 Keil 的仿真与调试。

在对工程成功编译, 连接后, 按 F5 或点击菜单 Debug Start/Stop Debug Session 或单击图即可进入调试状态。

经济划算的高性能调试工具

miniWi ggler 是英飞凌面向未来的经济划算的高性能调试工具。在主机侧, 它具备一个 USB 接口。每台计算机都具备 USB 接口。在器件侧, 则可通过英飞凌 10-针 DAP 或 16-针 OCDSL1 接口, 进行通信。miniWi ggler 经专门设计, 可配合英飞凌调试访问软件 (DAS) 使用。

最新版本的 DAS 可在 www.infineon.com/das 下载。

应用:

调试,

- Flash 烧录.

特性:

- 兼容英飞凌 DAP 和 SPD
- 兼容 JTAG/IEEE 1149.1
- 时钟速率最高达 30 MHz (可编程)
- 所有信号均为 5.5V, 可下调至 1.65V
- USB 2.0 (高速)
- 经认证的驱动程序, 适用于微软 Windows 2000、XP 和 VISTA 操作系统



- USB、JTAG 和 DAP/SPD 热插拔和拔下
- DAP1/SPD 引脚上提供 UART 功能，以支持 BSL 和 BMI 烧录
- 3 个板上状态 LED 指示灯
- 支持 OCDS1 16-针和 DAP 10-针连接器

可支持的器件:

- 英飞凌 8、16 和 32 位衍生产品.

可支持的工具:

- ALTIUM/编译器

- [DAVE™ Bench](#)

- Hi TOP HI TEX

- KEIL uVi si on

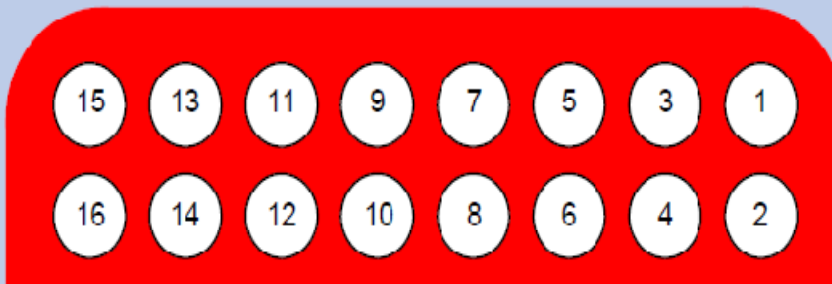
DAP Mini Wi ggl er 开发工具实物图片



Low Cost Debug Solution from Infineon



Pin Out



- | | |
|-----------------------------|--|
| 1 - TMS | 9 - $\overline{\text{TRST}}$ |
| 2 - Vdd | 10 - $\overline{\text{DAS_TRG1, BRKOUT}}$ |
| 3 - TDO | 11 - TCLK |
| 4 - GND | 12 - GND |
| 5 - empty | 13 - $\overline{\text{DAS_TRG0, BRKIN}}$ |
| 6 - GND | 14 - $\overline{\text{DAS_USR0, OCDSE}}$ |
| 7 - $\overline{\text{TDI}}$ | 15 - empty |
| 8 - $\overline{\text{RST}}$ | 16 - empty |