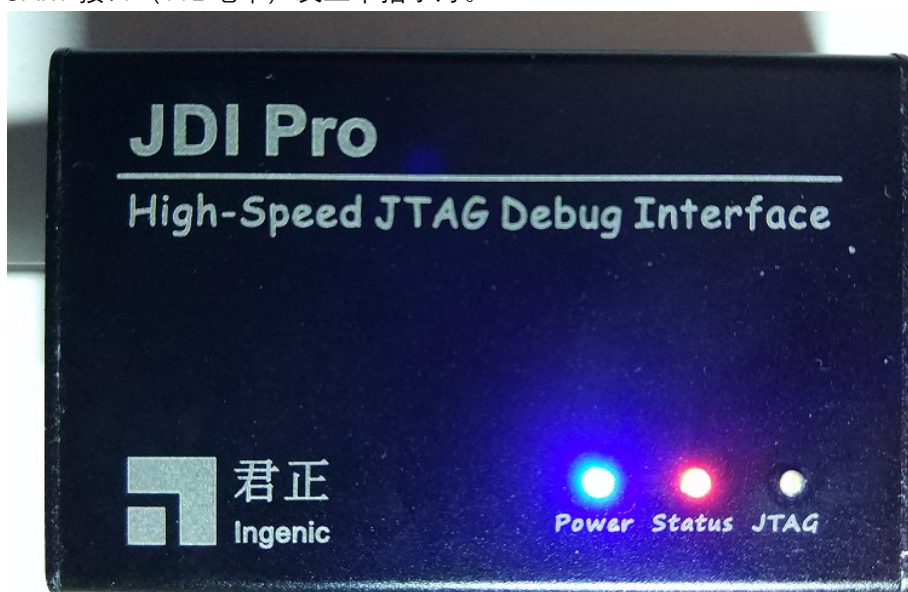
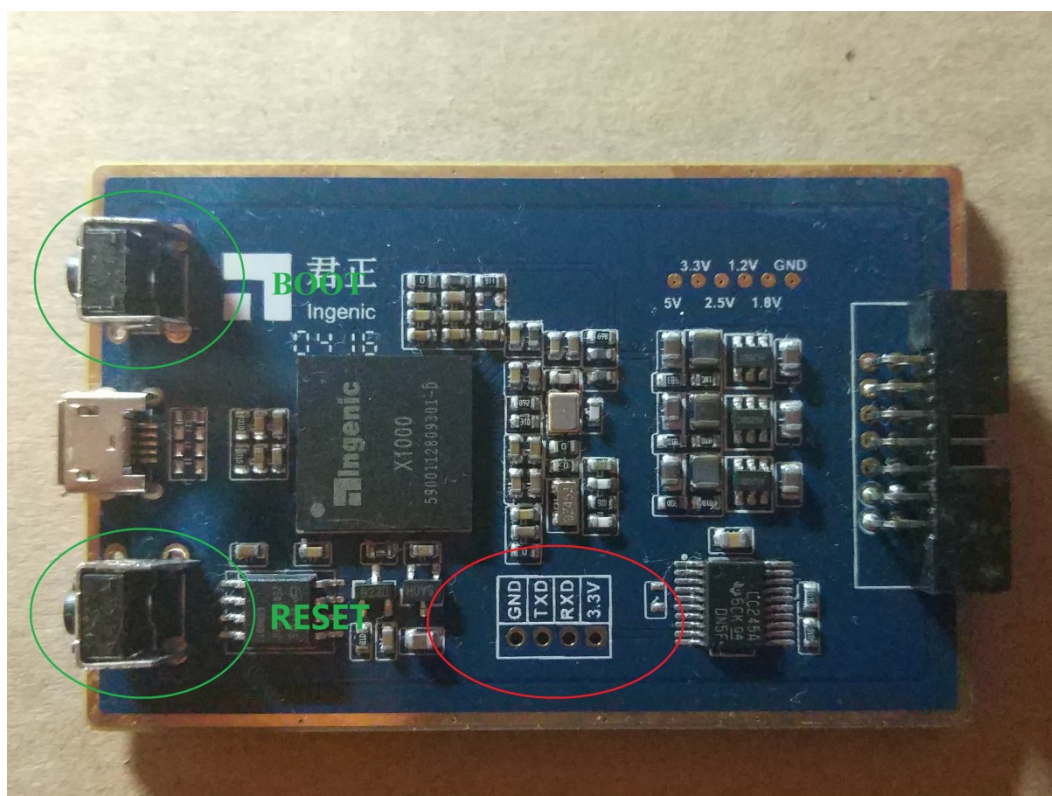


一、 JDI 硬件介绍

1.JDI 硬件包括供电及用于与 PC 通信的 USB 接口、与目标板通信的 EJTAG 接口、用于调试 JDI 的 UART 接口（TTL 电平）及三个指示灯。

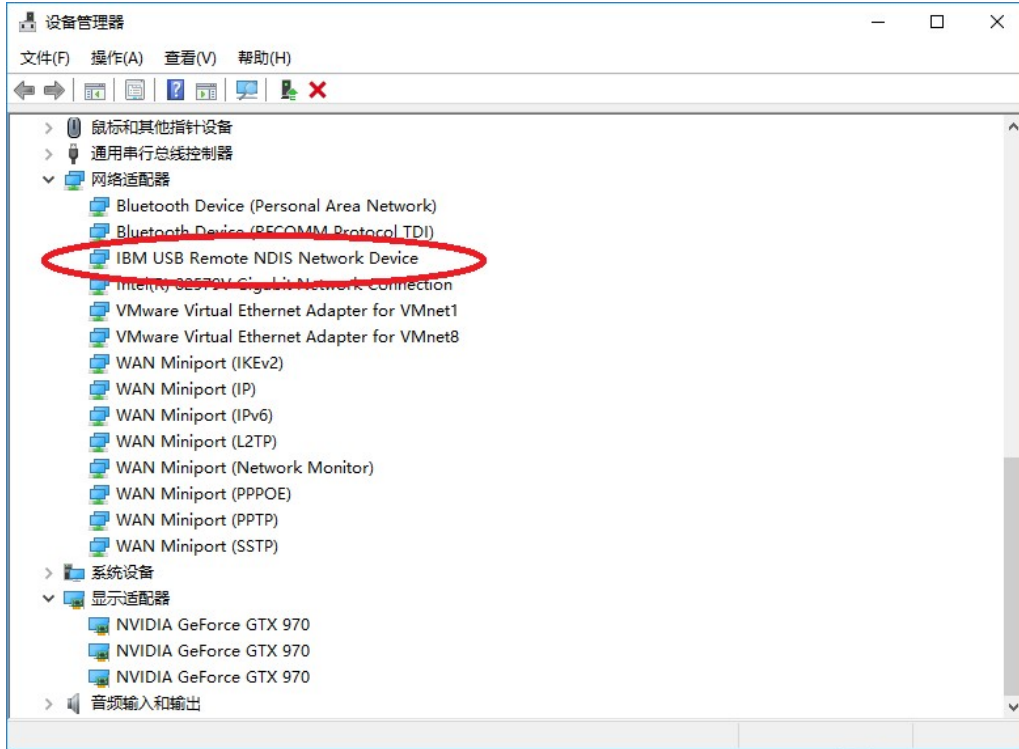


如上图所示，USB 口和 EJTAG 口分别位于 JDI 外壳左右两侧，Power 指示灯为电源指示灯，上电后处于常亮状态。Status 指示灯为状态及网络连接指示灯，当 JDI 准备就绪后会以每秒 4 次的频率快速闪烁，此时可使用配置助手对 JDI 进行配置，当配置成功并与计算机建立连接后，该指示灯时会以 1.5 秒为周期缓慢闪烁。JTAG 指示灯在 EJTAG 口有数据通信时会闪烁。下图为 JDI 内部电路板，红框中为调试 JDI 时使用的 UART 接口（TTL 电平），绿框中分别为 RESET 按钮及 BOOT 按钮，以上两按钮仅在手动更新 JDI 固件时需要使用。



二、连接和配置 JDI

1.JDI 使用 RNDIS 通过 USB 电缆同计算机连接，目前暂时使用的是 IBM 的 VID 和 PID，若使用 Windows 系统，连接成功后会出现下图所示设备：

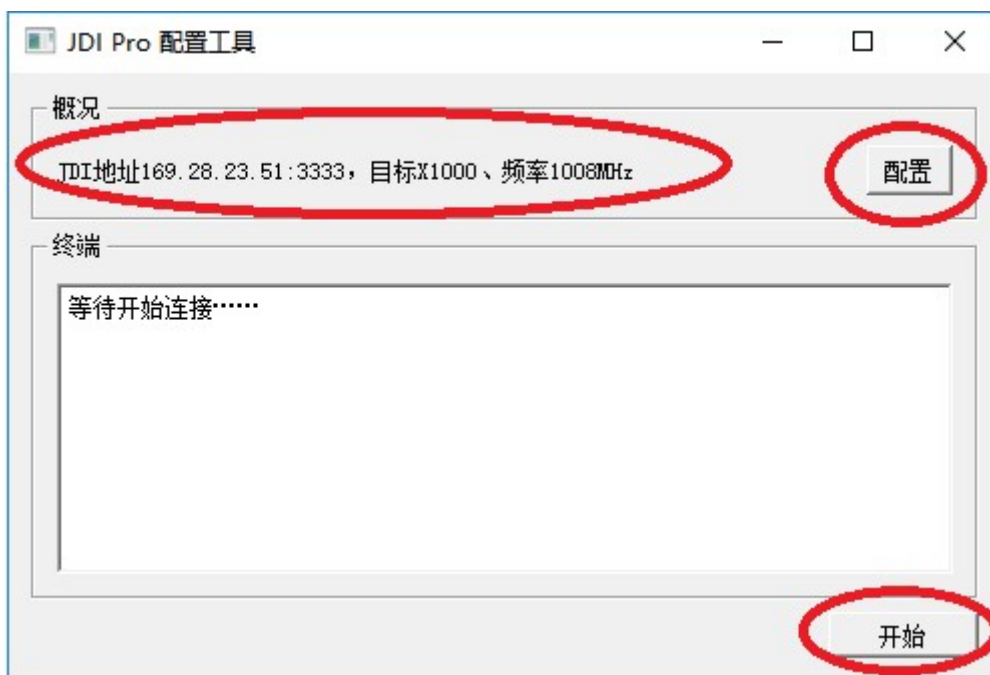


若使用 Linux 操作系统，连接成功后会出现如下设备：

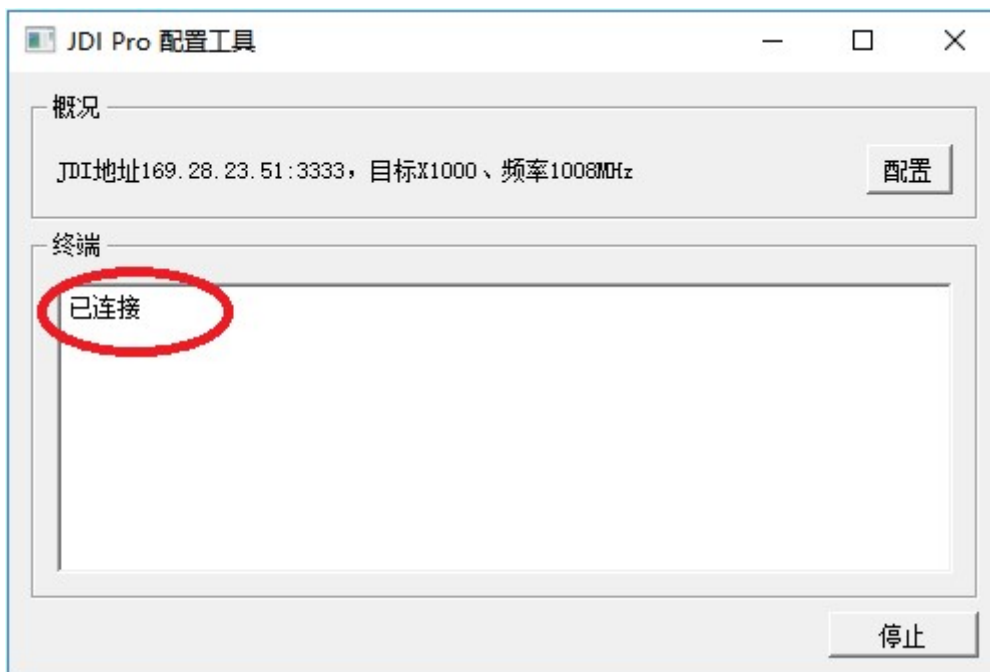
```
jdi@jdi-virtual-machine: ~  
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H) 128  
jdi@jdi-virtual-machine:~$ ifconfig  
ens33  Link encap:以太网 硬件地址 00:0c:29:01:18:e8  
inet 地址:192.168.137.140 广播:192.168.137.255 掩码:255.255.255.0  
inet6 地址: fe80::48f1:7ee9:3a9a:1db/64 Scope:Link  
UP BROADCAST RUNNING MULTICAST MTU:1500 跃点数:1  
接收数据包:34811 错误:0 丢弃:0 过载:0 帧数:0  
发送数据包:43589 错误:0 丢弃:0 过载:0 载波:0  
碰撞:0 发送队列长度:1000  
接收字节:2273213 (2.2 MB) 发送字节:2924615 (2.9 MB)  
  
enxde8deaf6004c Link encap:以太网 硬件地址 de:8d:ea:f6:00:4c  
inet 地址:169.28.23.6 广播:169.28.23.255 掩码:255.255.255.0  
inet6 地址: fe80::3e33:9fa1:45ed:aed/64 Scope:Link  
UP BROADCAST RUNNING MULTICAST MTU:1500 跃点数:1  
接收数据包:8 错误:0 丢弃:0 过载:0 帧数:0  
发送数据包:45 错误:0 丢弃:0 过载:0 载波:0  
碰撞:0 发送队列长度:1000  
接收字节:1010 (1.0 KB) 发送字节:8033 (8.0 KB)  
  
lo  Link encap:本地环回  
inet 地址:127.0.0.1 掩码:255.0.0.0  
inet6 地址: ::1/128 Scope:Host  
UP LOOPBACK RUNNING MTU:65536 跃点数:1  
接收数据包:3795 错误:0 丢弃:0 过载:0 帧数:0  
发送数据包:3795 错误:0 丢弃:0 过载:0 载波:0  
碰撞:0 发送队列长度:1000  
接收字节:202338 (202.3 KB) 发送字节:202338 (202.3 KB)  
jdi@jdi-virtual-machine:~$
```

2. 当前版本 JDI 提供了对 X1000 和 X1000E 处理器的调试支持，后续会陆续提供对跟多处理器的支持。因此连接好 JDI 后需使用配套的配置助手对 JDI 进行配置。步骤如下：

- (1) 将 JDI 通过 USB 数据线连接至计算机，等待片刻至 status 指示灯开始快速闪烁。
- (2) 选择需要调试的处理器型号，确认无误后点击“开始”按钮进行配置。

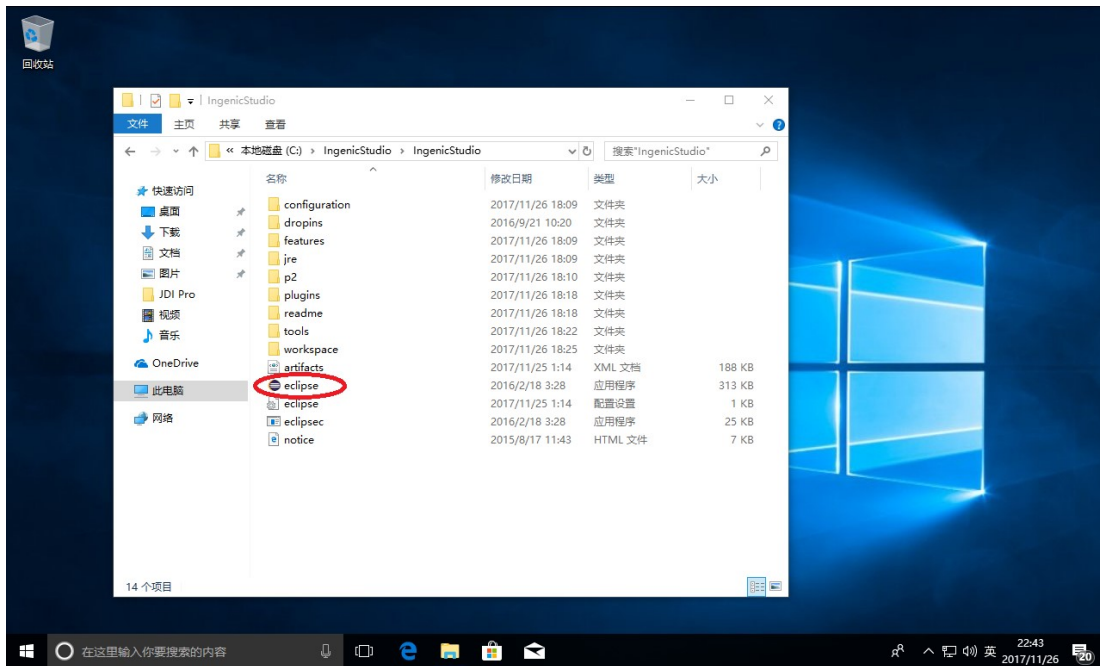


(3) 配置成功后终端会提示“已连接”，此时 status 指示灯会以约 1.5 秒一次慢速闪烁。

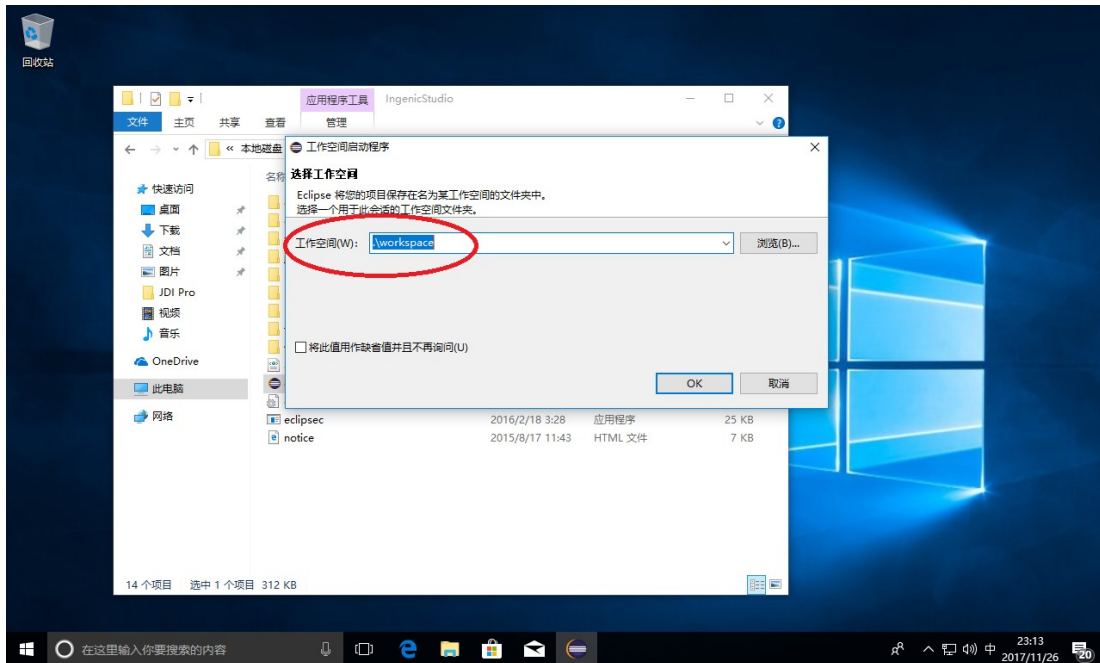


三、 IDE 的使用及在 IDE 中进行调试（裸机工程）

1. 进入 IDE 目录， 双击 eclipse.exe 启动 IDE 环境



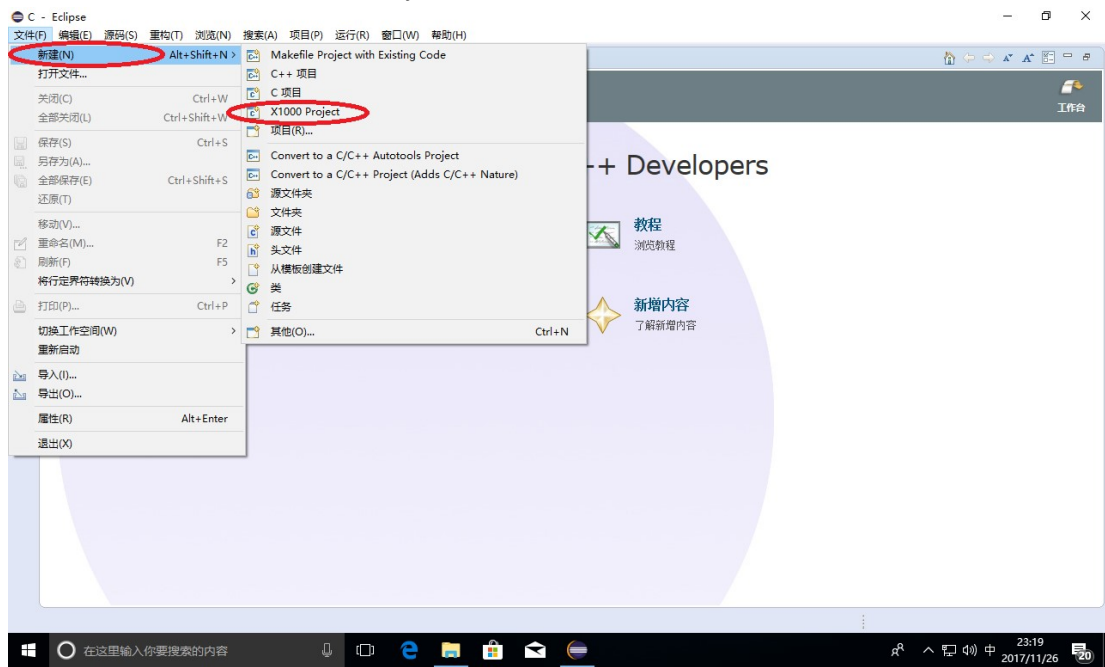
2. 首次启用 IDE 时需要设置 workspace 路径， 默认路径位于 IDE 目录下。



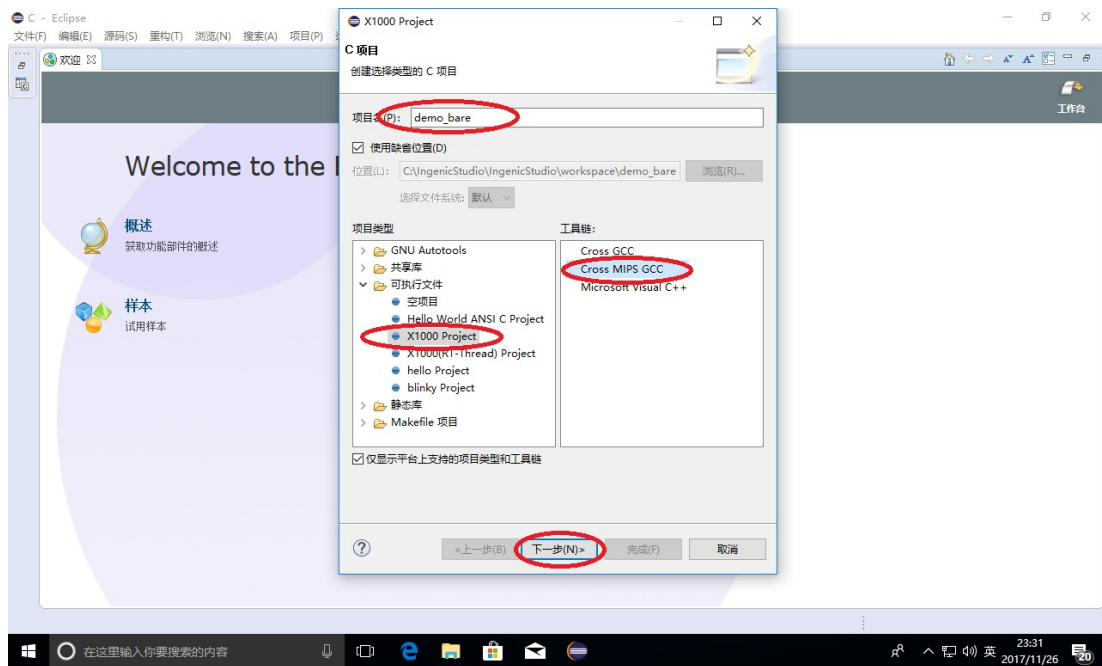
3. 点击“文件”按钮建立工程



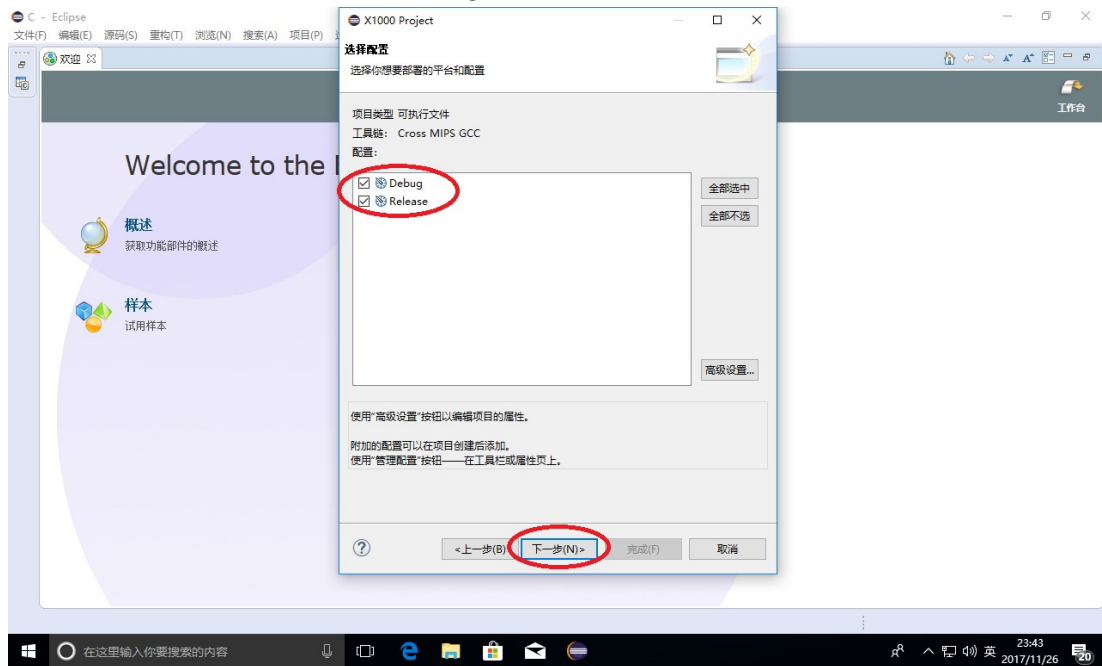
4. 选择“新建”菜单下的“X1000 Project”



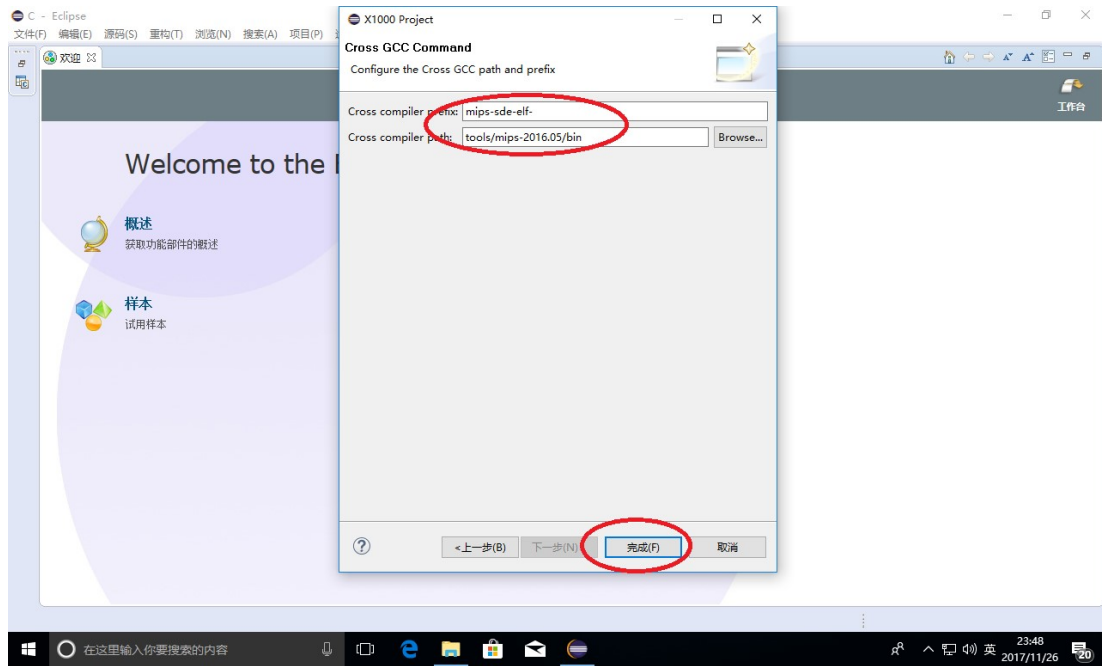
5.在弹出的对话框中填的“项目名”处填写项目名字，选择“X1000 Project”建立裸机工程，工具链需要选择 Cross MIPS GCC，随后点击“下一步”



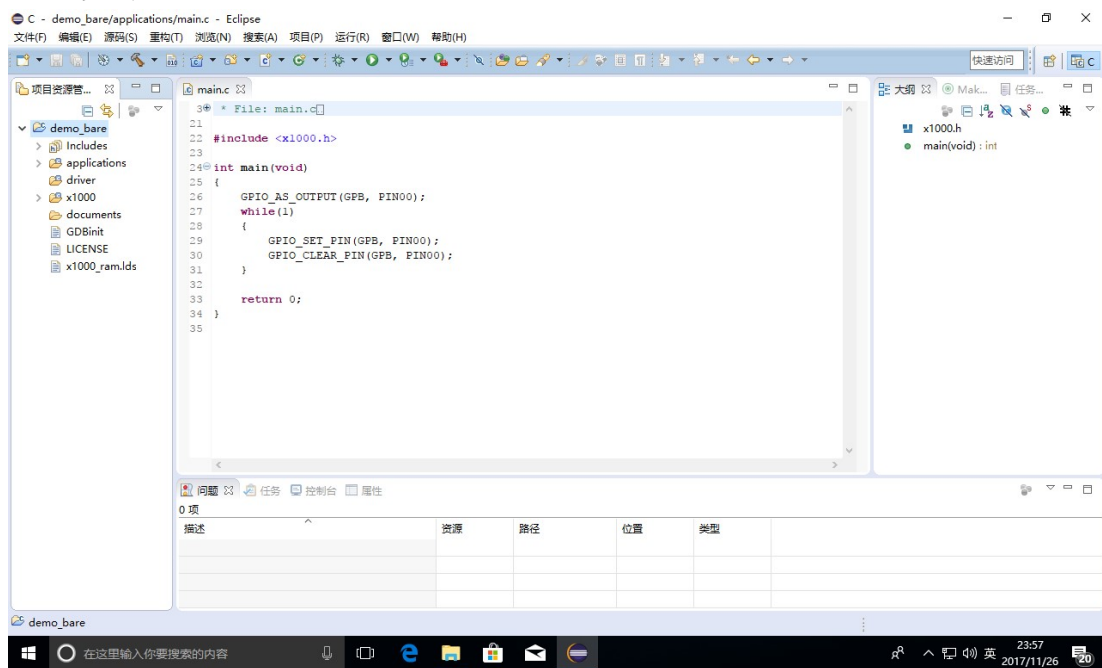
6.选择所需的编译配置，默认是 Debug 模式和 Release 模式均被选中，随后点击“下一步”



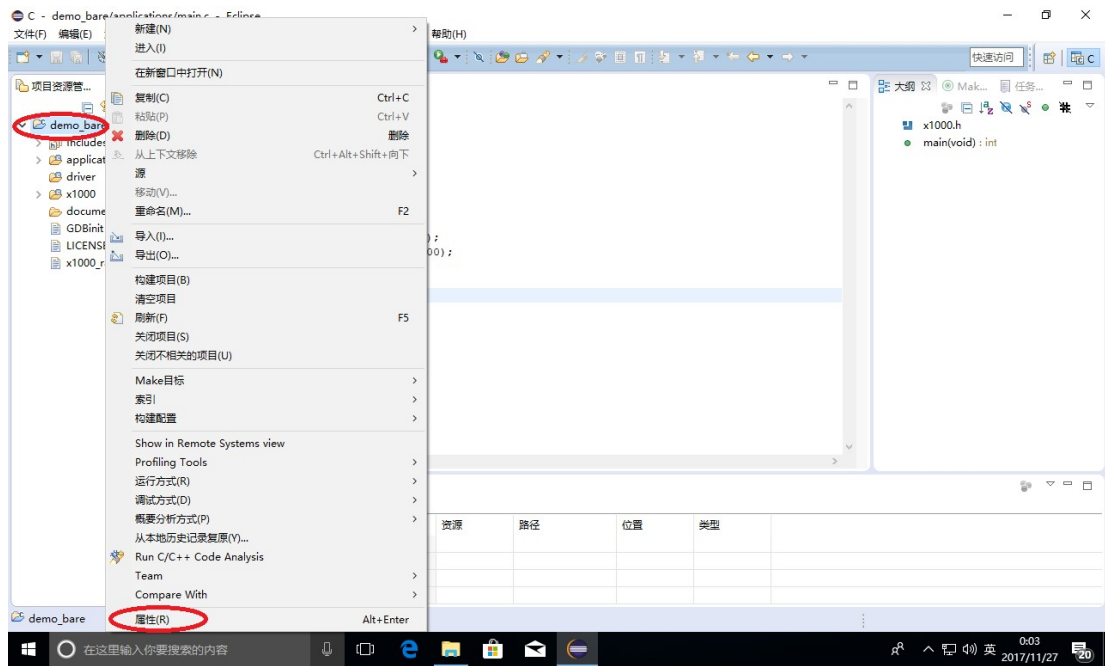
7.设置工具链前缀及 path 路径，使用默认值即可，随后点击“完成(F)”



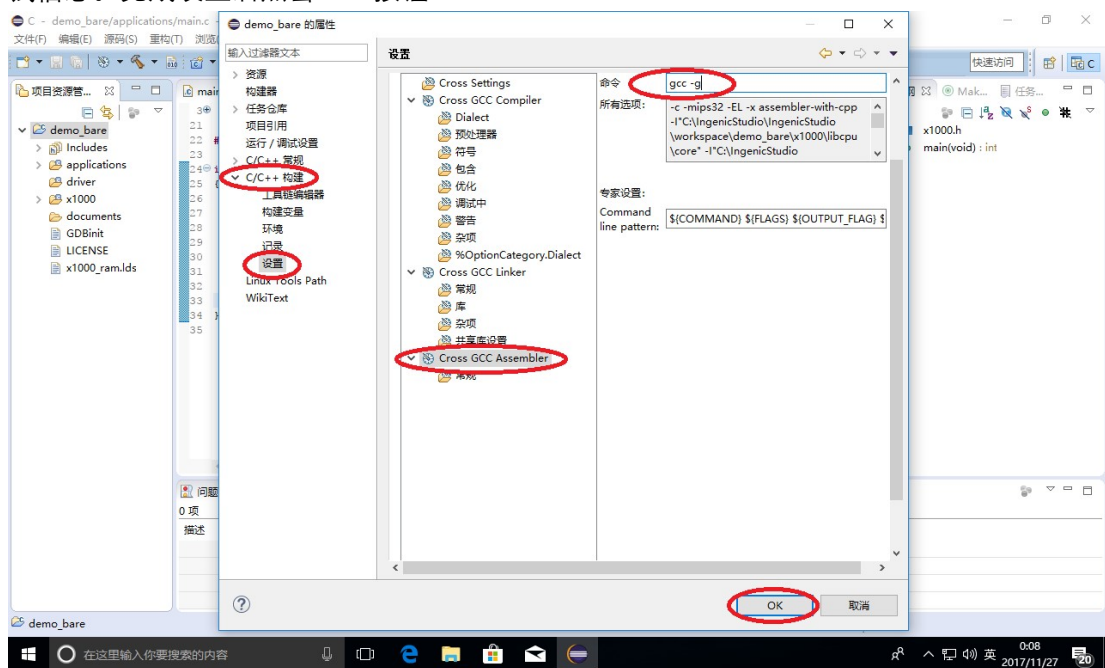
8.工程建立完毕后可见到如下界面，因为环境设置还不完善，因此需要参照步骤 9 进行一些必要的手动设置。



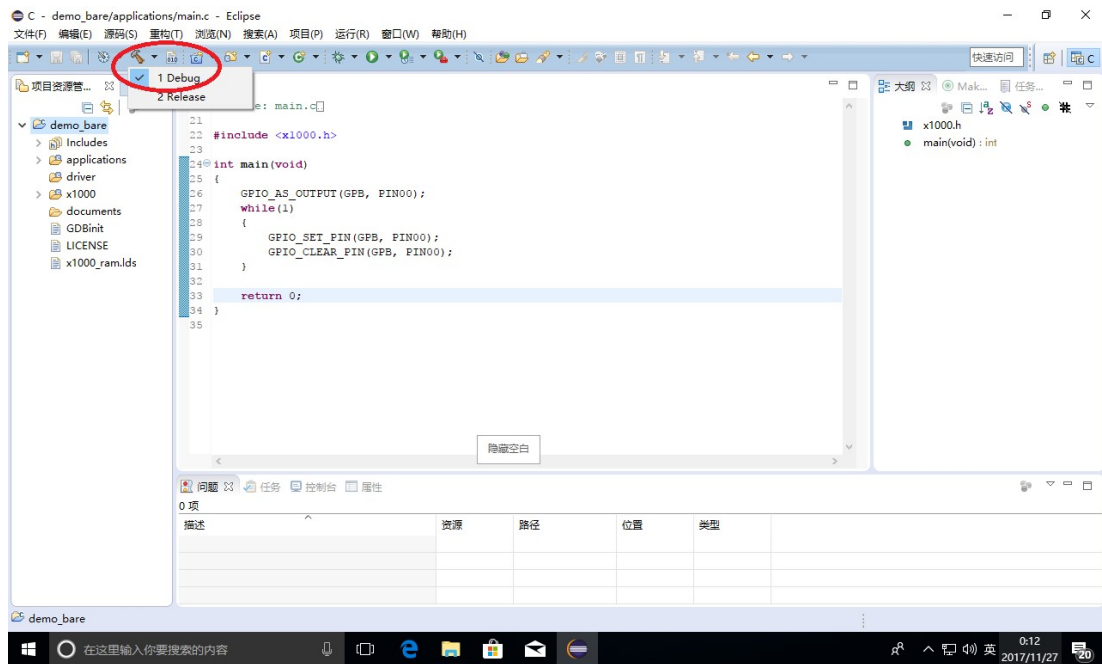
9. 在工程目录上右键单击，在弹出的菜单中选中“属性”



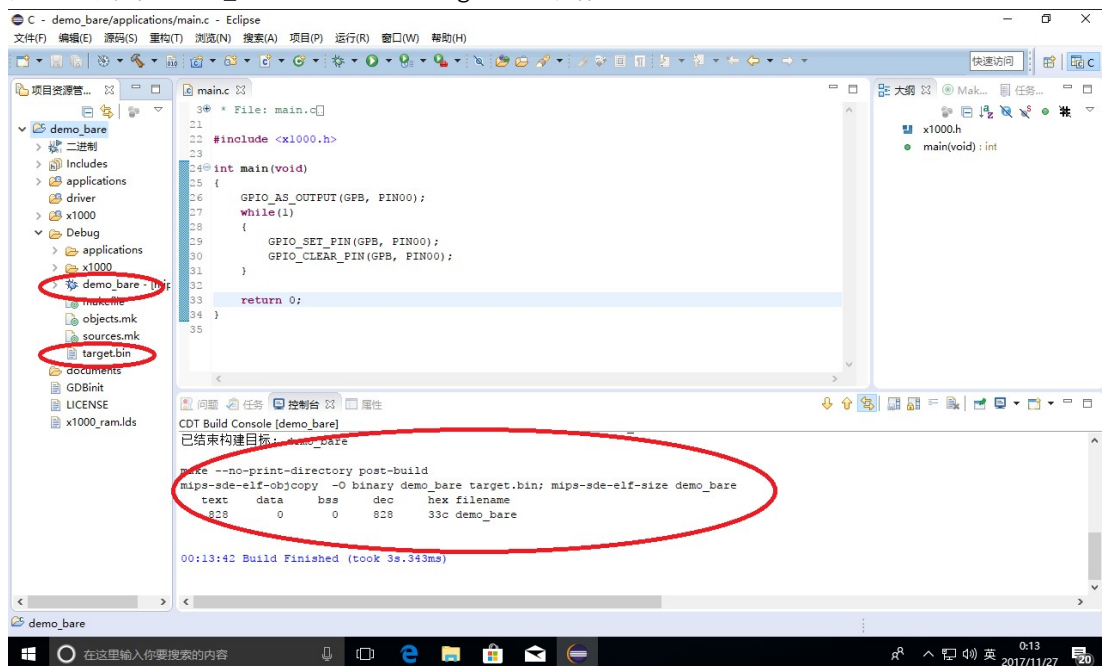
10. 在属性对话框中选中“C/C++ 构建”条目下的“设置”，之后在右侧对话框中选中“Cross GCC Assembler”，在“命令”一栏中的 gcc 后加上 -g 命令以便在生成的汇编启动函数中也包含有调试信息。完成设置后点击“OK”按钮



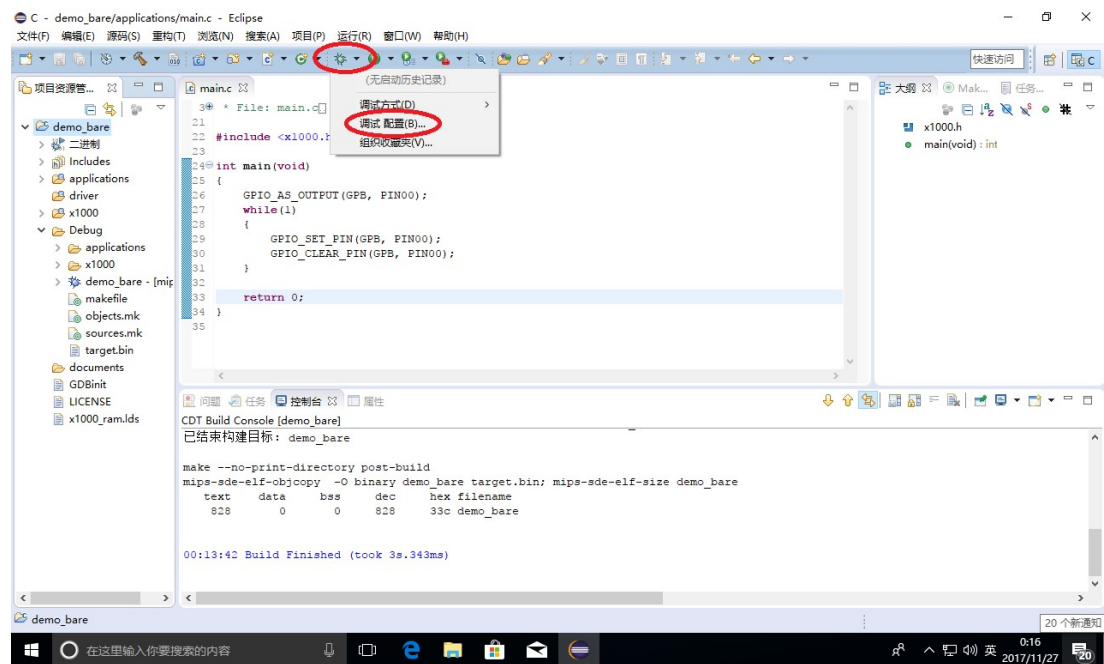
11. 在如图所示位置选择 Debug 模式进行编译



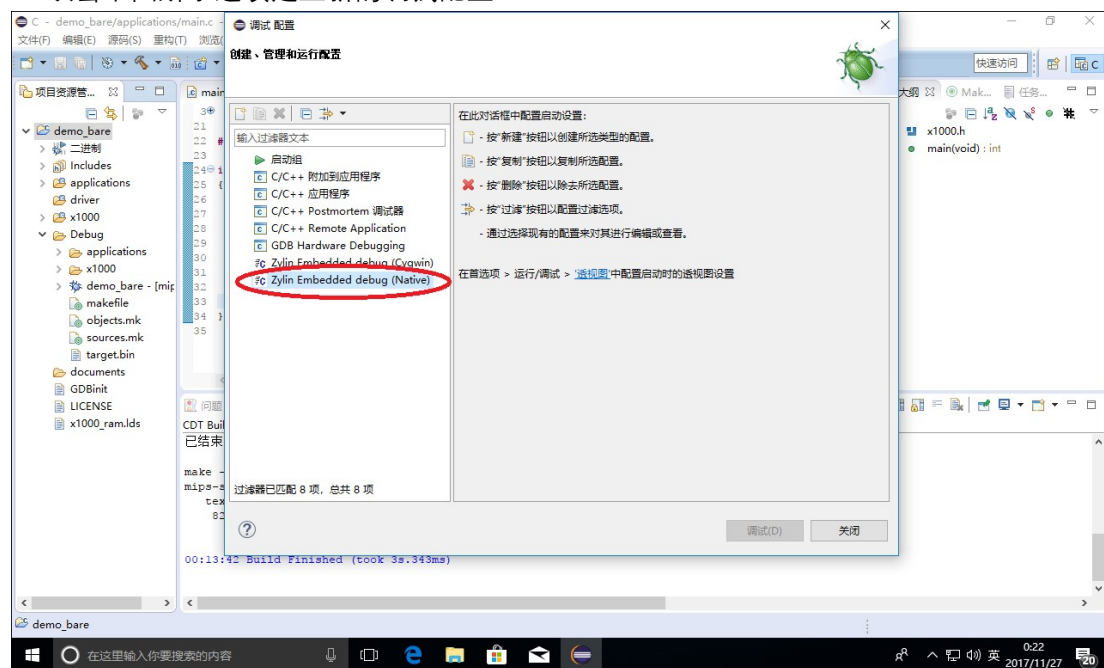
12. 正常完成编译后在控制台会有如图所示提示信息，同时在工程下的 Debug 文件夹会生成用于调试的 demo_bare.elf 文件和 target.bin 镜像。



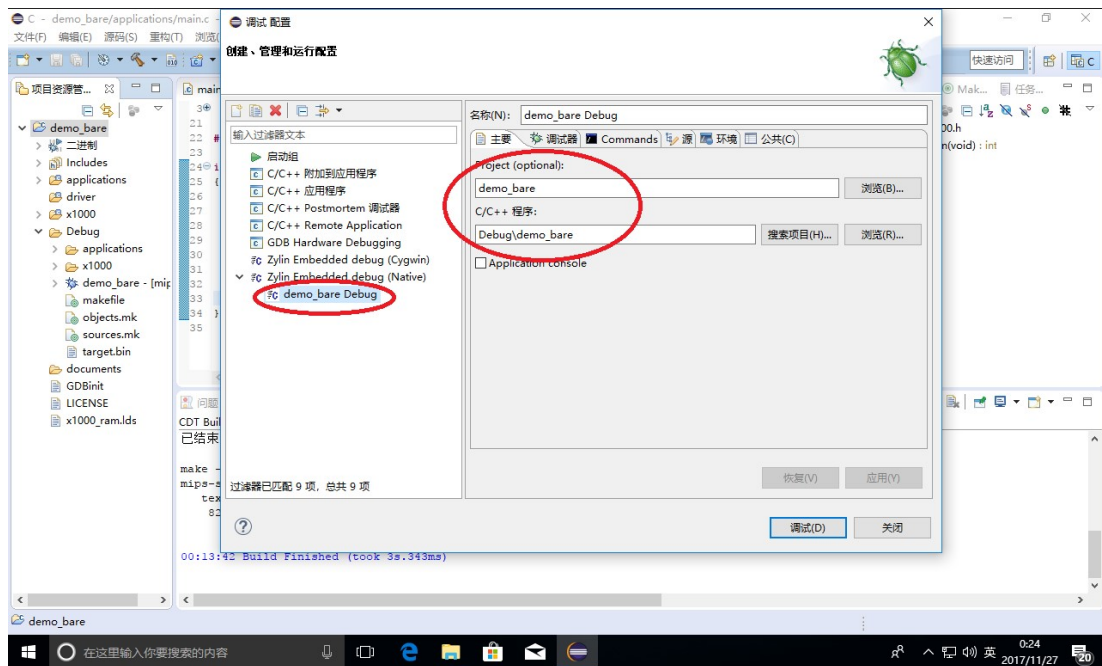
13. 点击下图所示图标，并选择“调试配置(B)”



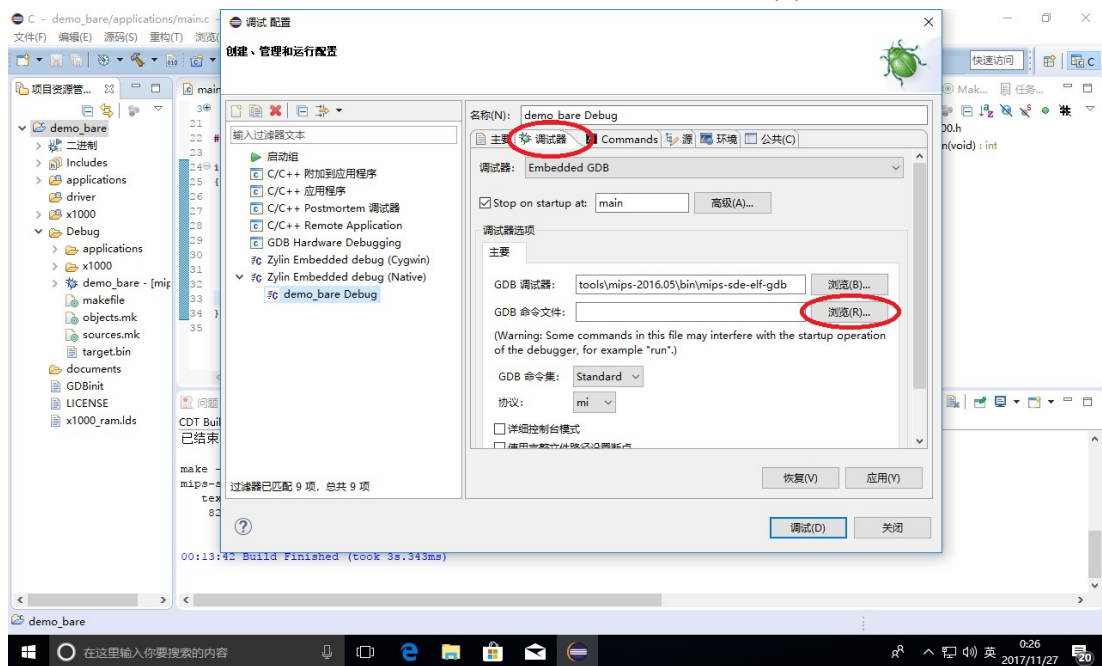
14. 双击下图所示选项建立新的调试配置



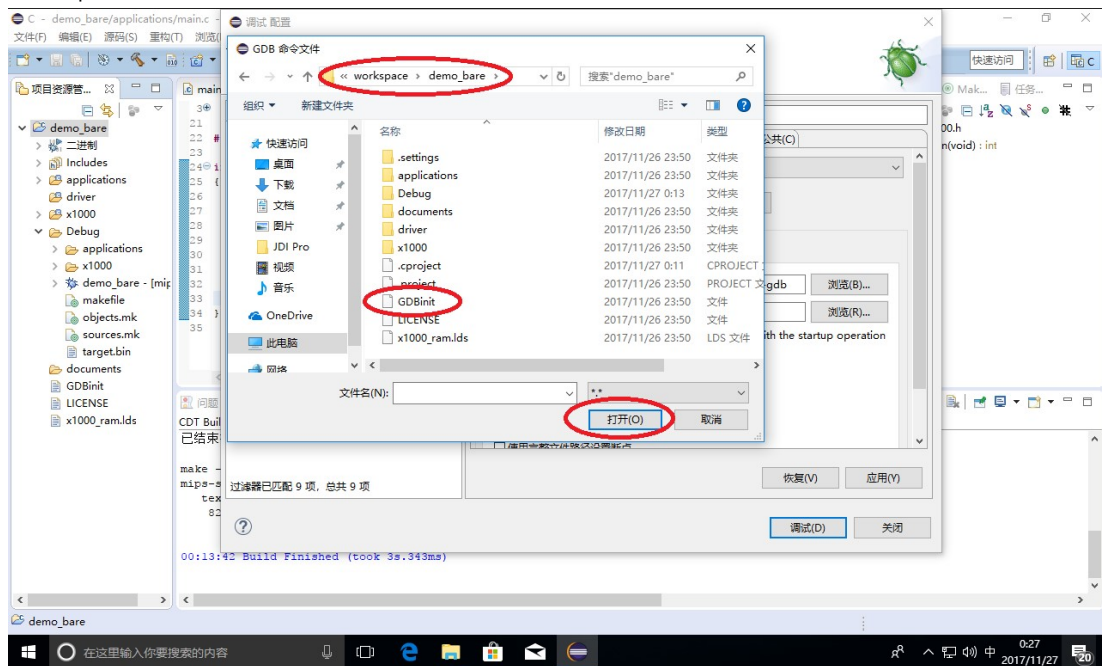
15.选中上一步骤中生成的调试配置，默认情况下会自动探测到图中所示的内容，若未能正常探测到对应信息可删掉调试配置并重复步骤 14，或者手动输入相关信息



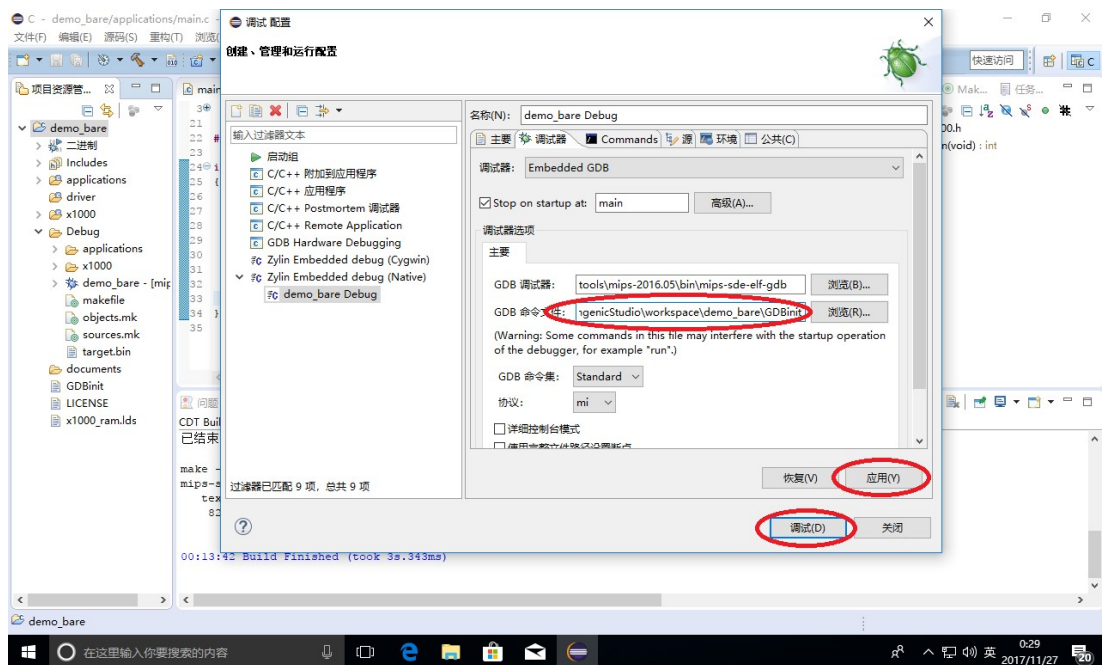
16.选中“调试器”选项卡，点击“GDB 命令文件”一栏右侧的“浏览(R)”



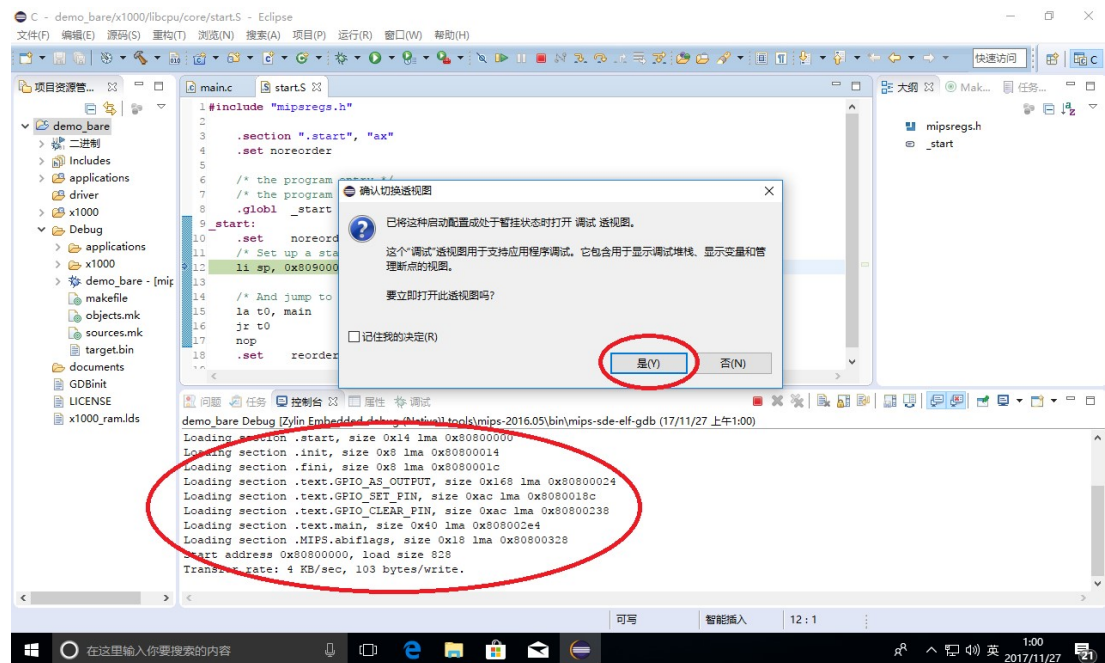
17.在弹出来的对话框中找到 GDBinit 文件，该文件位于工程目录下，本例中的路径为 workspace/demo_bare/GDBinit.选中该文件后点击“打开”



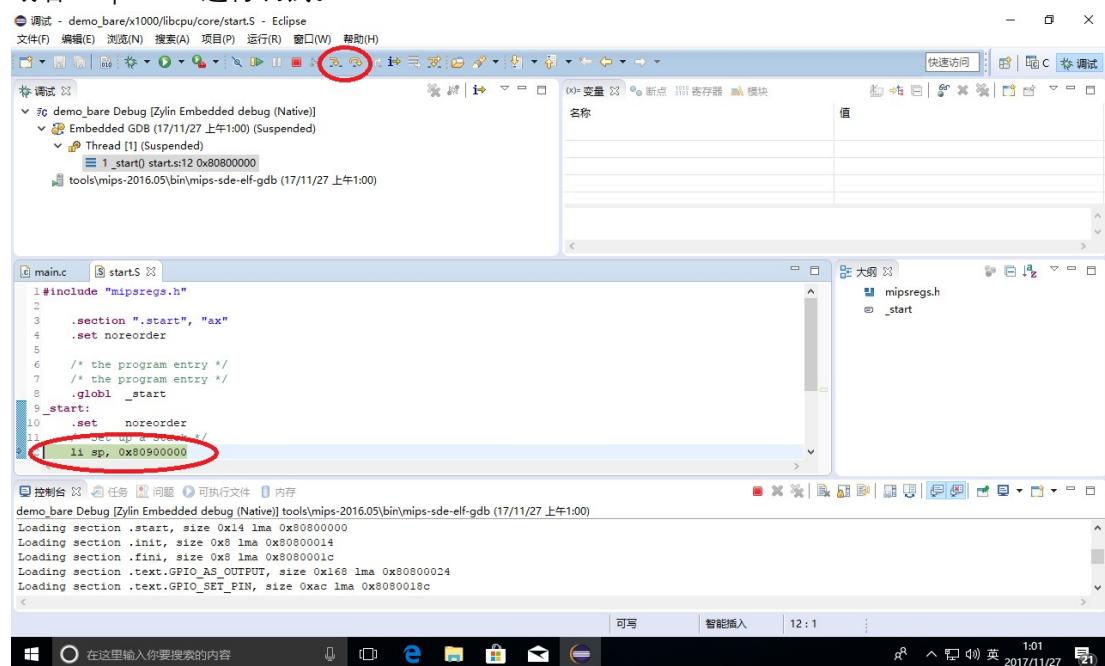
18.此时应看到在“GDB 命令文件”一栏中已经显示为上一步骤所指向的文件，随后点击右下角的“应用(Y)”，然后点击“调试(D)”按钮启动调试。注意在启动调试前应确保 JDI 的 EJTAG 接口已经通过电缆同目标板妥善连接。



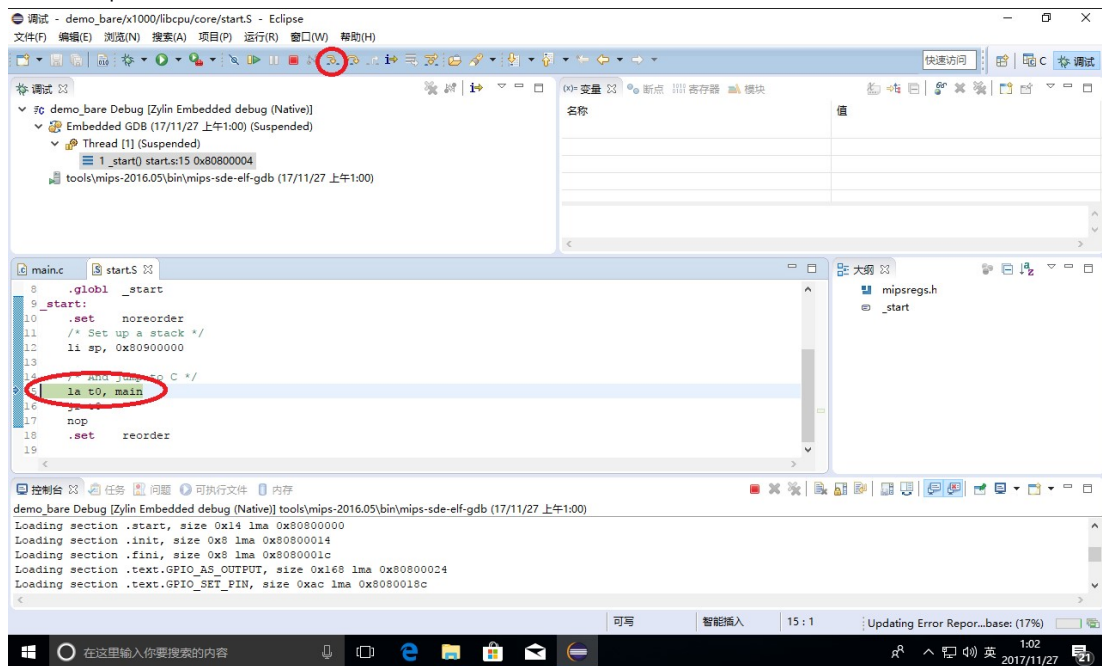
19.当 JDI 相目标板载入待调试文件结束后在下方会显示载入的目标文件的相关信息及载入速度。点击“确认切换透视图”对话框下方的“是(Y)”按钮可切换至调试界面



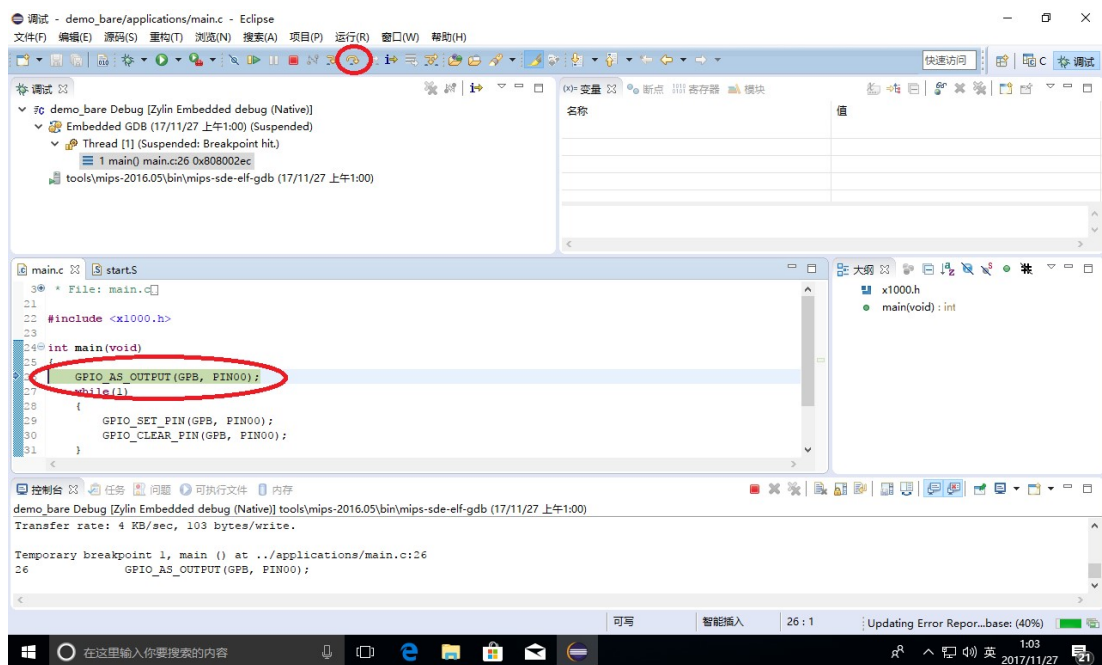
20.切换至调试界面后中部区域的光条显示的是当前即将执行的命令，可通过上方的 Step In 或者 Step Over 进行调试。



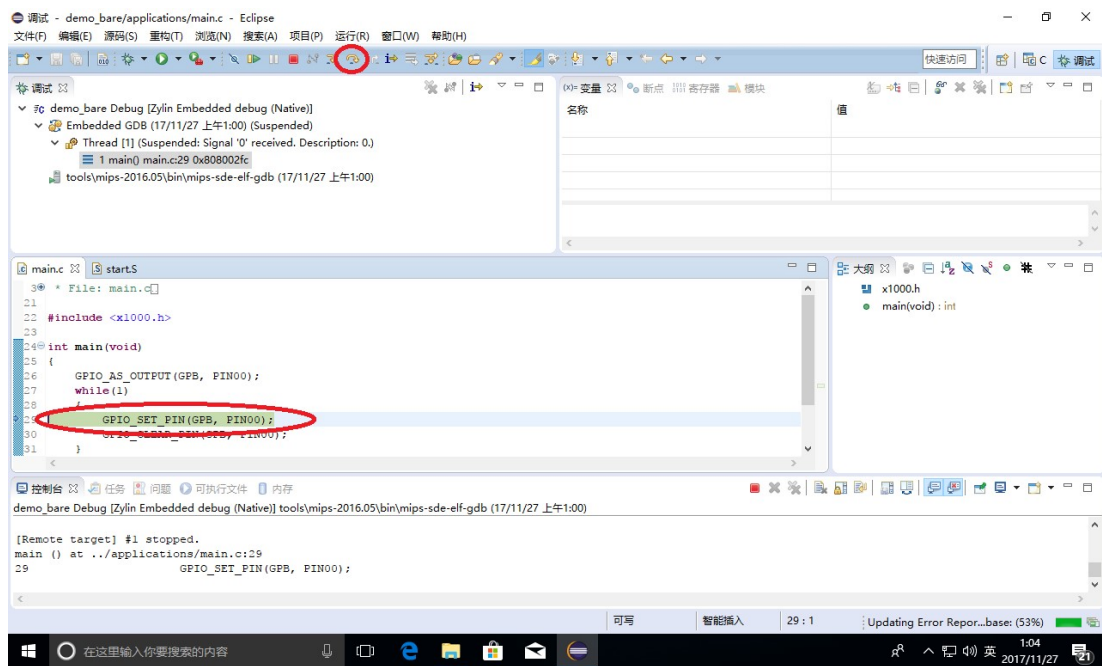
21.此处以 Step In 为例，可以看见点击一次 Step In 后光标由第 12 行跳至了第 15 行。再次点击 Step In 按钮，则会继续向后执行。



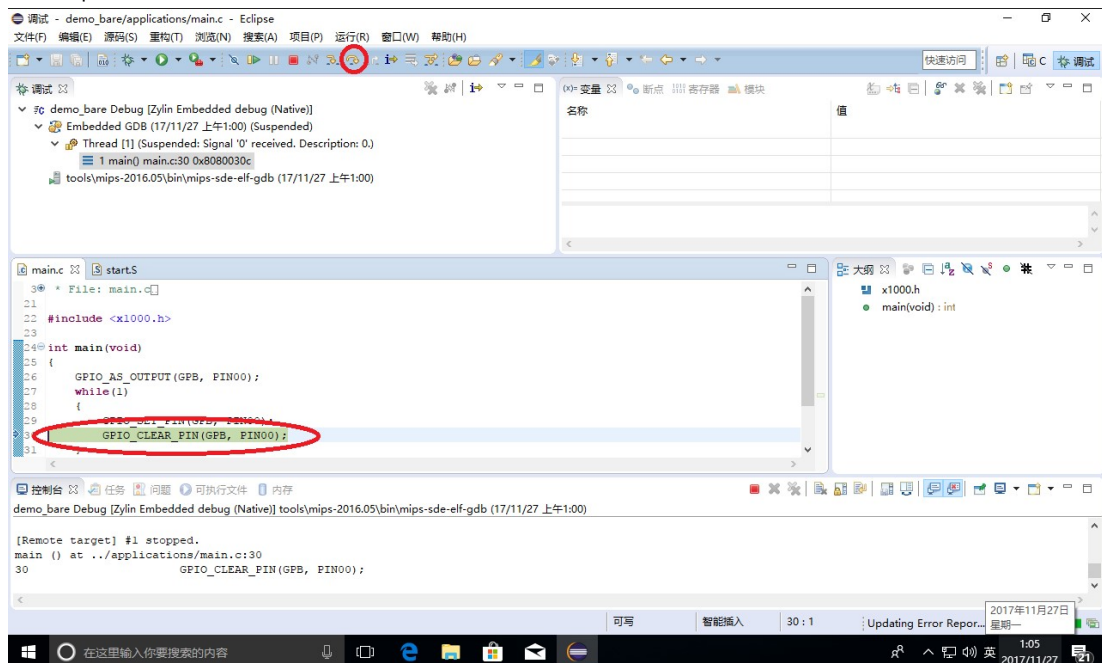
22.此时可以看见已经跳到了C文件中的 main 函数内。现在以 Step Over 为例，点击 Step Over 后会直接执行完当前待执行的整个函数，而不会像 Step In 一样进入到函数内部逐条命令执行。执行完图中所示的命令后，就已经将目标板的 PB00 设置为了 GPIO 输出模式



23.此时可见到已经执行到 While 循环内部，当前待执行的函数将会使 PB00 输出高电平，点击 Step Over 按钮执行完该条命令后，可用万用表在 PB00 引脚测到一个大约 3.3V 左右的高电平输出。



24.如图所示，当前待执行的函数会使 PB00 输出低电平，继续点击 Step Over 按钮执行完该条命令后，可用万用表在 PB00 引脚测到一个接近于 0V 的低电平输出。此后如果继续点击 Step Over 将会在 while 中不断循环，同时万用表上可以检测到对应的高低电平。若此后点击 Step In 则会进入到两个函数内部，逐条命令的执行。

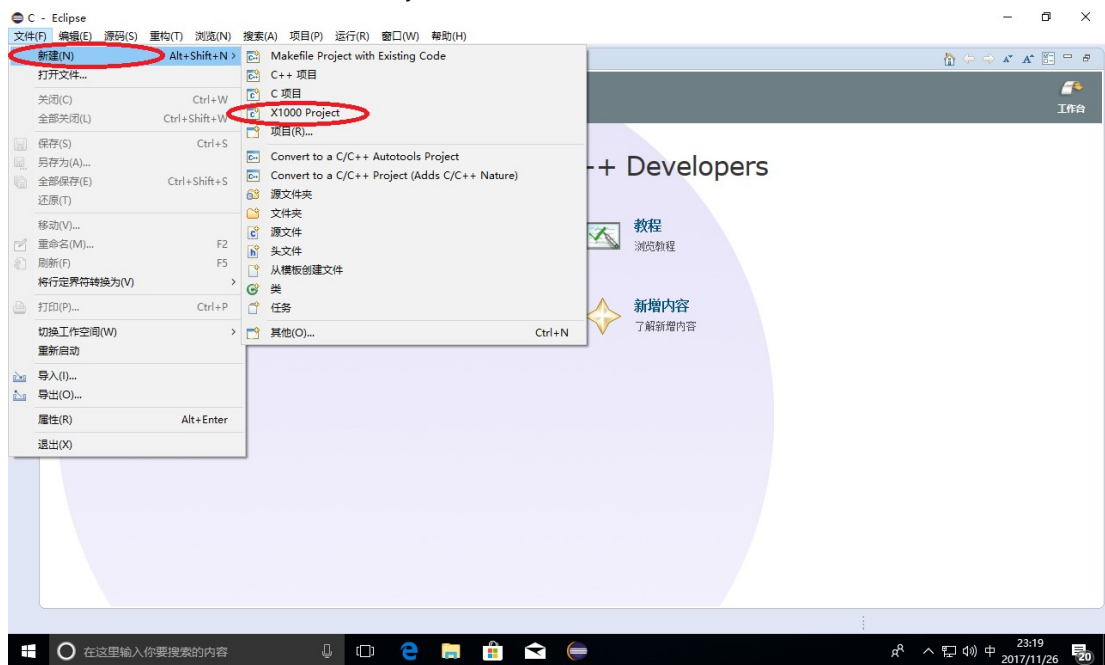


四、IDE 的使用及在 IDE 中进行调试 (RT-Thread)

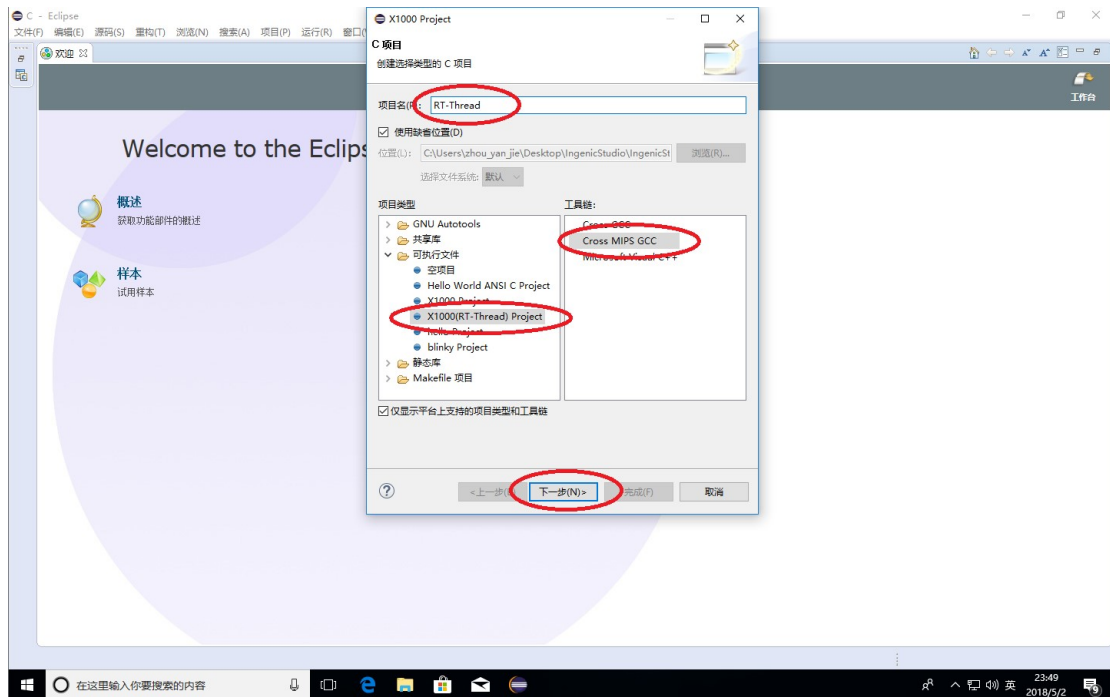
1. 点击“文件”按钮建立工程



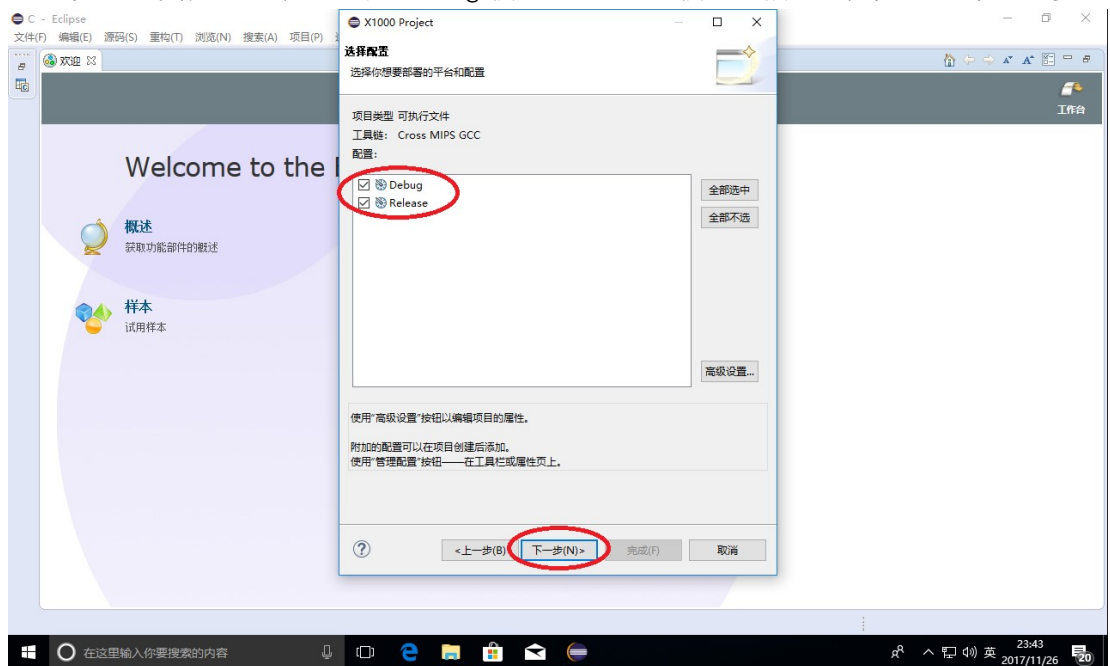
2. 选择“新建”菜单下的“X1000 Project”



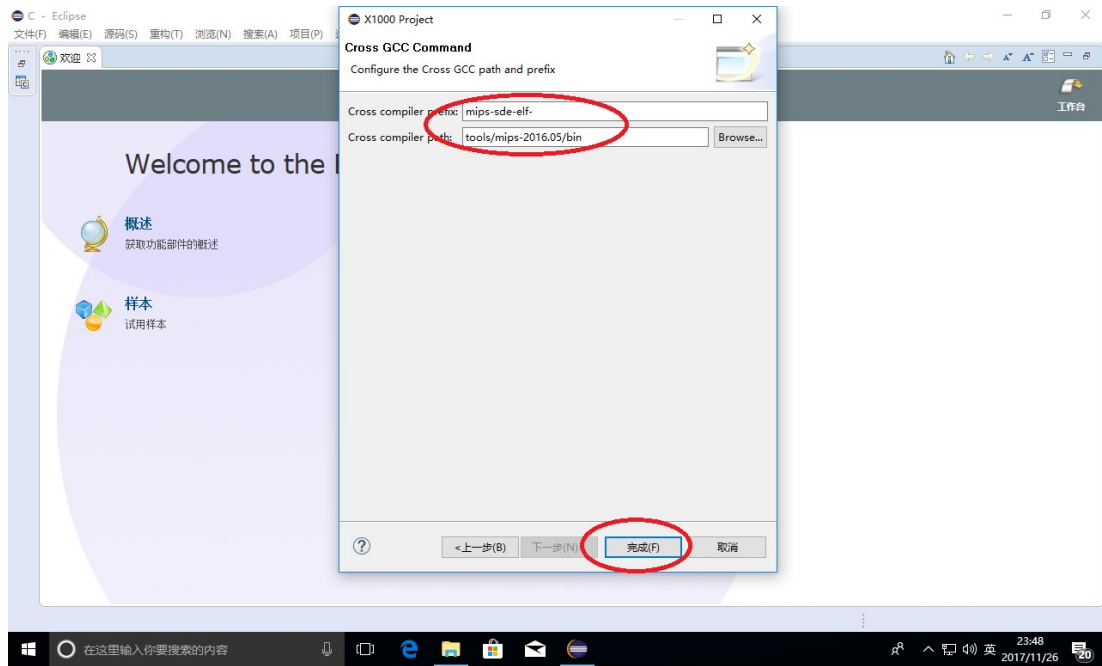
3.在弹出的对话框中填的“项目名”处填写项目名字，选择“X1000(RT-Thread) Project”建立 RT-Thread 工程，工具链需要选择 Cross MIPS GCC，随后点击“下一步”



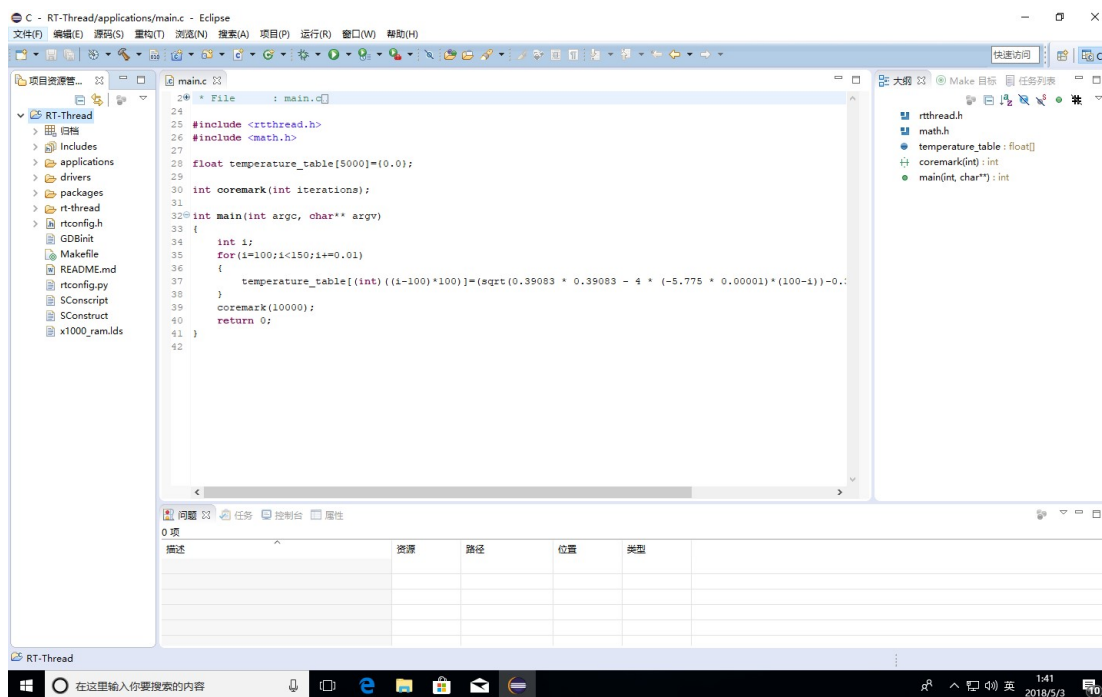
4.选择所需的编译配置，默认是 Debug 模式和 Release 模式均被选中，随后点击“下一步”



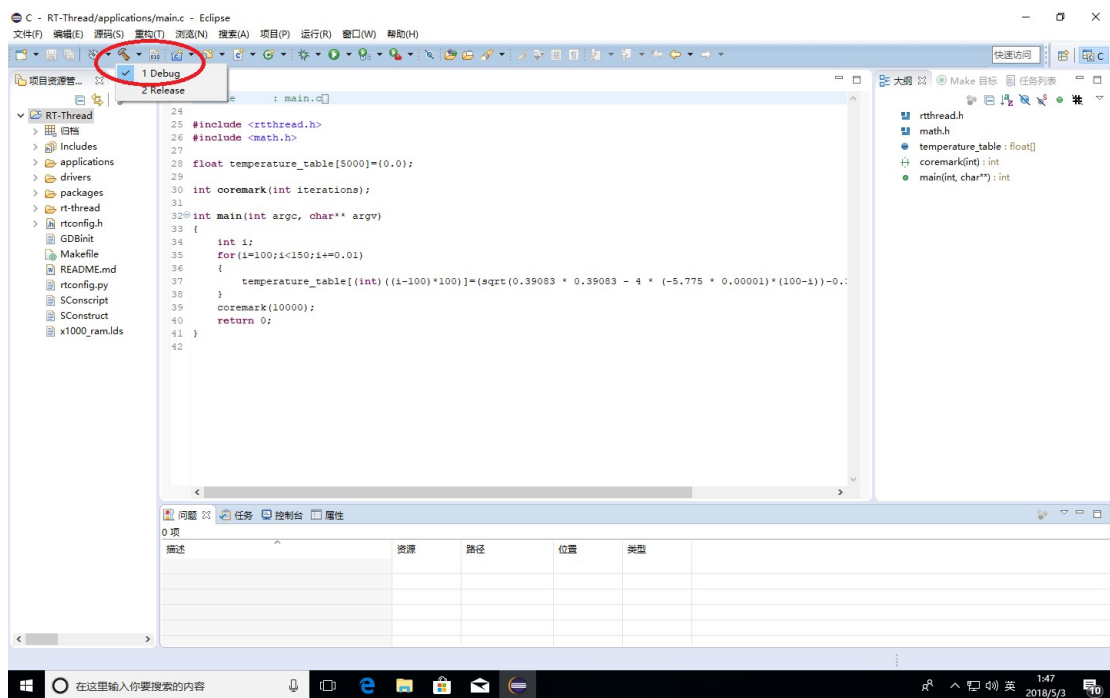
5.设置工具链前缀及 path 路径，使用默认值即可，随后点击“完成(F)”



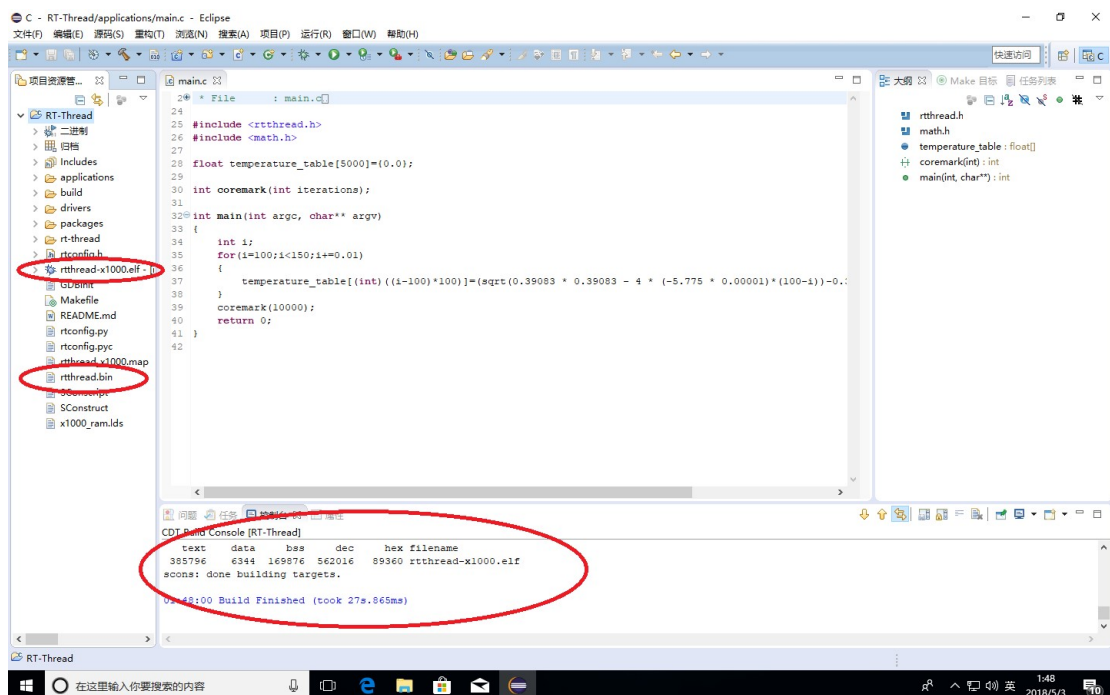
6.工程建立完毕后可见到如下界面。



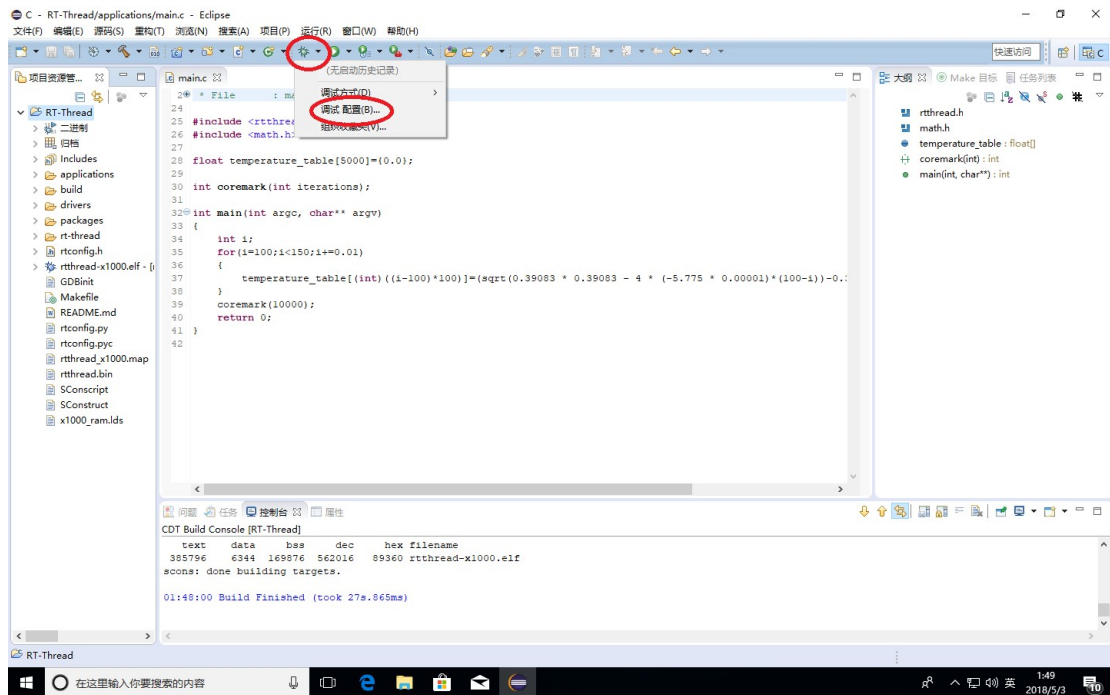
7.在如图所示位置选择 Debug 模式进行编译



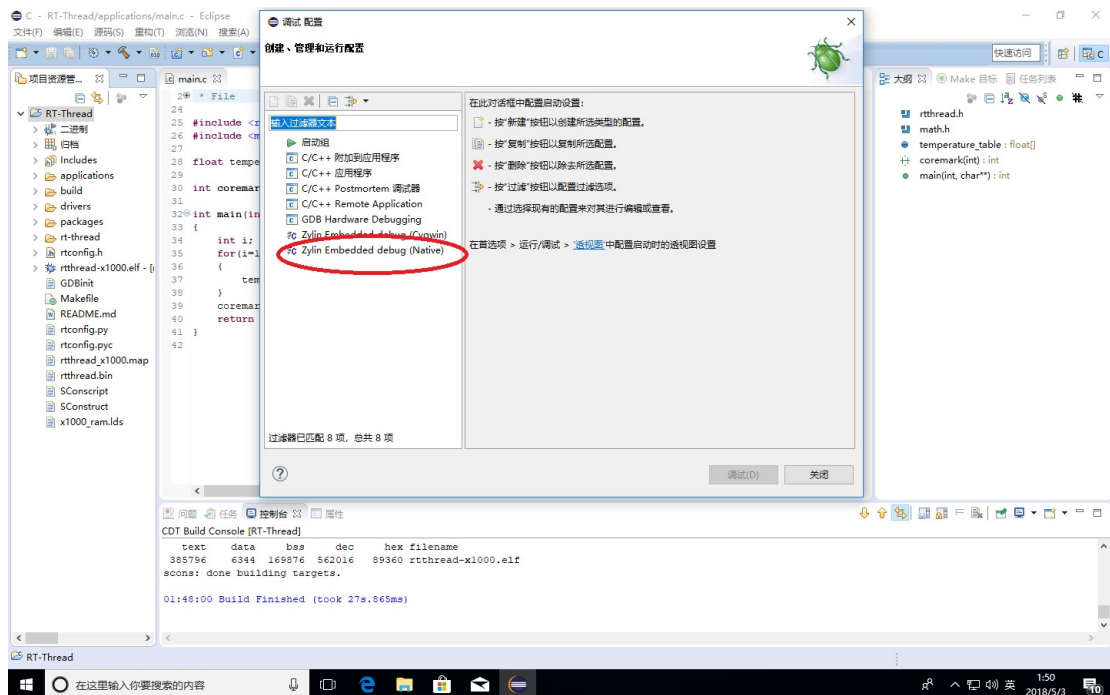
8.正常完成编译后在控制台会有如图所示提示信息，同时在工程下会生成用于调试的 rtthread-x1000.elf 文件和 rtthread.bin 镜像。



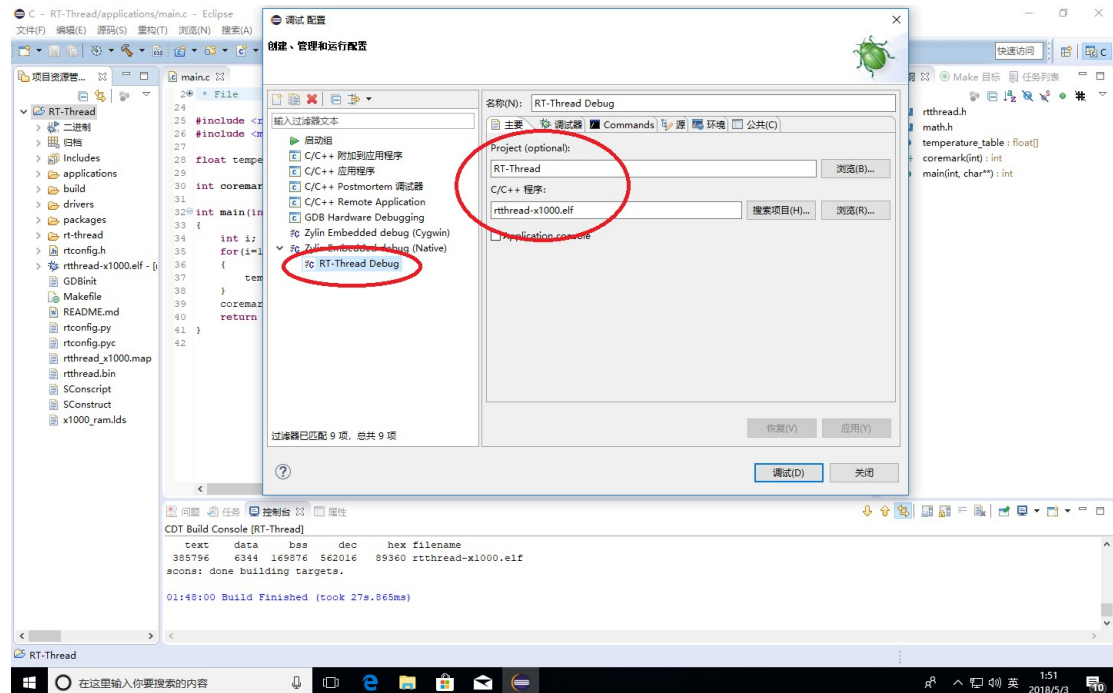
9. 点击下图所示图标，并选择“调试配置(B)”



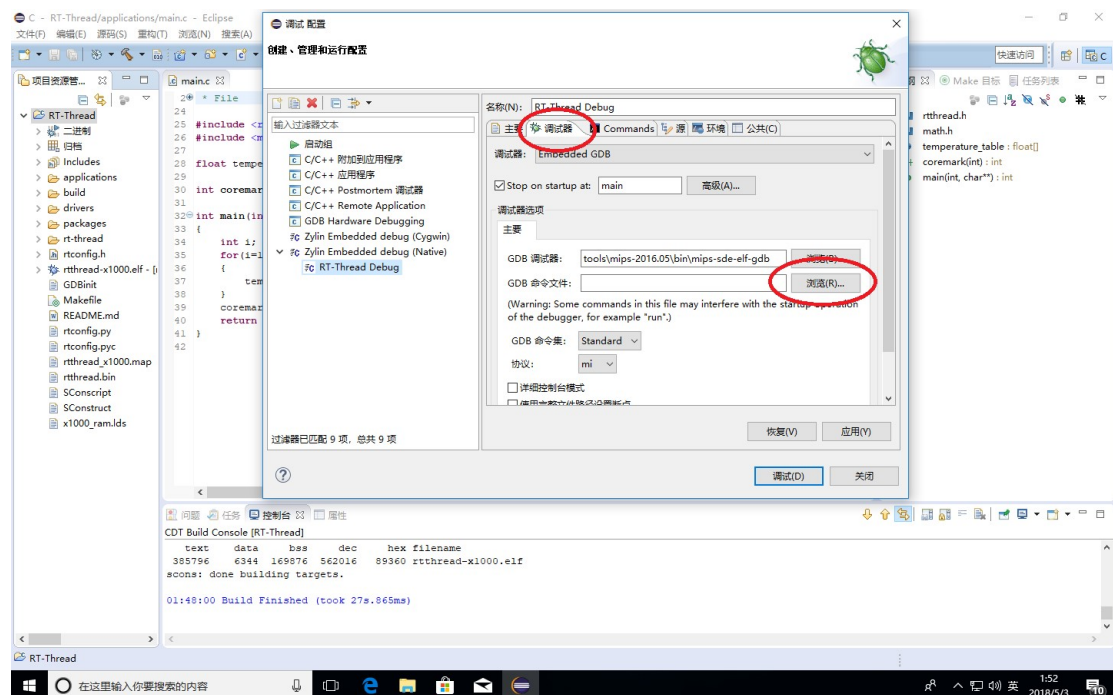
10. 双击下图所示选项建立新的调试配置



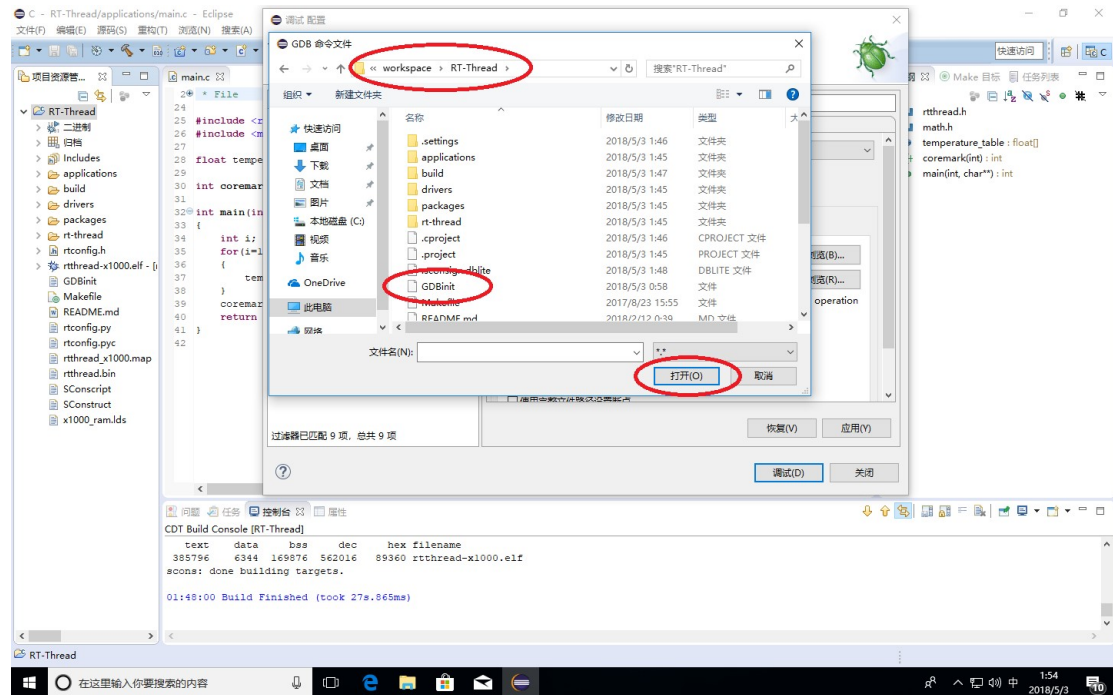
11.选中上一步骤中生成的调试配置，默认情况下会自动探测到图中所示的内容，若未能正常探测到对应信息可删掉调试配置并重复步骤 10，或者手动输入相关信息



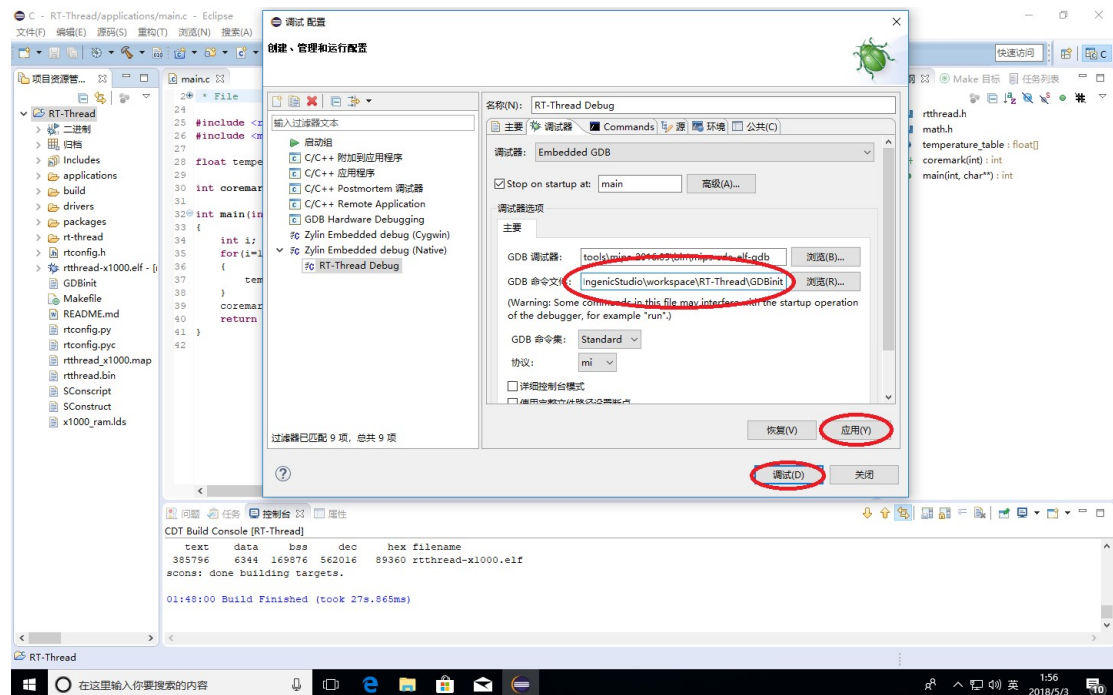
12.选中“调试器”选项卡，点击“GDB 命令文件”一栏右侧的“浏览(R)”



13.在弹出来的对话框中找到 GDBinit 文件，该文件位于工程目录下，本例中的路径为 workspace/RT-Thread/GDBinit.选中该文件后点击“打开”



14.此时应看到在“GDB 命令文件”一栏中已经显示为上一步骤所指向的文件，随后点击右下角的“应用(Y)”，然后点击“调试(D)”按钮启动调试。注意在启动调试前应确保 JDI 的 EJTAG 接口已经通过电缆同目标板妥善连接。



15.当 JDI 相目标板载入待调试文件结束后在下方会显示载入的目标文件的相关信息及载入速度。点击“确认切换透视图”对话框下方的“是(Y)”按钮可切换至调试界面

