2

DS-5 快速使用实例

2.1 导入示例项目到 Eclipse

本文档中描述的许多任务都是使用 DS-5 提供的示例项目。要在 Eclipse 中使用示例项目，必须首先导入它们。

（1）启动 Eclipse。

- Windows 系统，选择【开始】→【所有程序】→【ARM DS-5】→【Eclipse for DS-5】菜单。
- Linux 系统，在 Unix bash shell 中输入 eclipse。

（2）ARM 建议为示例项目创建一个新的工作区，使它们保持与用户自己的项目分开。要做到这一点，可以：

- 启动 Eclipse，创建一个新的工作区目录。
- 如果 Eclipse 已经打开，从主菜单中选择【File】→【Switch Workspace】→【Other】菜单。

（3）从 Help 菜单中选择【Cheat Sheet...】菜单。

（4）展开 ARM 组。

（5）从 ARM Cheat Sheet 中选择【Automatically Import the DS-5 Example Projects into the Current Workspace】。

（6）单击【OK】按钮。

（7）按照 Cheat Sheet 中的步骤，导入所有的 DS-5 示例项目到工作区。

当示例项目被导入后，如果需要，可以选择按照其余的备忘单说明来切换工作集。

2.2 在 Eclipse 中编译 Gnometris 项目

Gnometris 是一个可以在目标板上运行和调试的 ARM Linux 应用程序。所提供的项目不包

含 Gnometris 应用程序的镜像二进制文件。要创建映像，必须编译该项目。

编译项目的步骤如下：

（1）从 ARM 网站下载可选包或从 DS-5 安装光盘获得 Linux_distribution_example.zip，包含兼容的头文件和库。

（2）从相关 ZIP 文件中导入 gnometris 和 distribution 示例项目到 Eclipse 中。

从【File】菜单中选择【Import...】。然后选择【Existing Project into Workspace】，选择 gnometris and distribution 示例项目，单击【Next】按钮。

（3）在 Project Explorer 视图中选择【gnometris 项目】。

（4）在 Project 菜单中选择【Build Project】。

Gnometris 示例项目包含编译项目的 Makefile 文件。Makefile 提供了常用的 make 规则：clean，all 和 rebuild。

编译 Gnometris 项目后，它产生以下应用程序：

- 一个不包含调试信息的 stripped 版本应用程序，这个要下载到目标板。
- 一个用于源代码级调试包含调试信息的应用程序，通常文件比较大。

2.3 命令行下编译 Gnometris 项目

在命令行编译 Gnometris 项目的步骤如下：

（1）从 ARM 网站上下载可选包或从 DS-5 安装光盘获得 Linux_distribution_example.zip，包含兼容的头文件和库。

（2）从相关 ZIP 文件中导入 gnometris 和 distribution 示例项目到 Eclipse 中。

（3）打开 DS-5 Command Prompt 命令行控制台或 Unix bash shell。

（4）进入...\ARMLinux\gnometris 目录。

（5）在提示符下输入 make。示例项目包含编译项目的 Makefile 文件。Makefile 提供了常用的 make 规则：clean，all 和 rebuild。编译 Gnometris 项目后，它产生以下应用程序：

- 一个不包含调试信息的 stripped 版本应用程序。这个要下载到目标板。
- 一个用于源代码级调试包含调试信息的应用程序，通常文件较大。

2.4 Real–Time System Model 上装载 Gnometris 程序

可以在运行的 ARM Linux 的 Real Time System Model（RTSM）模拟器上装载 Gnometris 应用程序。RTSM 模拟器能使用户在主机工作站上运行和调试应用程序，而无需使用任何的硬件标设备。

预先设定的 RTSM 连接将自动启动 Linux，启动 gdbserver，然后启动应用程序。

在 Real Time System Model（RTSM）上装载 Gnometris 的步骤如下：

（1）启动 Eclipse。

（2）单击【Explorer 视图】。

（3）展开 gnometris 项目文件夹。

（4）在 gnometris-RTSM-example.launch 上单击右键。

（5）在菜单中选择【Debug As】。

（6）在子菜单中选择【gnometris-RTSM-example】菜单。

（7）调试需要使用 DS-5 Debug 视图。弹出视图切换对话框后，单击【Yes】来切换视图。

2.5 装载 Gnometris 程序到 ARM Linux

可以装载 Gnometris 程序到运行 ARM Linux 的目标板上。DS-5 提供预先设定的目标板连接设置，该设置可连接调试器到运行在支持 ARM 架构平台上的 gdbserver。

装载程序步骤：

（1）获得目标板的 IP 地址。使用 ifconfig 命令可在 Linux 控制台读取 IP 地址。IP 地址是由 inet add 标识的。

（2）启动目标板的 Linux。

（3）启动 Eclipse。

（4）传送应用程序及相关文件到 ARM Linux 目标板上，运行应用程序，然后连接调试器。这个工作有以下几种实现方法：

- 在有些开发板上可以使用 Secure SHell（SSH），用 DS-5 提供的远程系统管理器（RSE）连接设置好的目标板，然后运行应用程序。当应用程序运行后，调试器就可以连接到正在运行的目标板。
- 对于其他的目标板，可以使用一个外部文件传输工具，如 PuTTY。

2.6 使用 SSH 连接设置和运行在 ARM Linux 上的 Gnometris

在某些目标板上，可以使用 Secure SHell（SSH），用 DS-5 提供远程资源管理器（RSE）连接目标板。

建立一个连接到 ARM Linux 目标板 Linux 的 SSH 连接，并运行 Gnometris 应用程序。步骤如下：

1. 第一步

在 DS-5 Debug 视图中添加 Remote Systems 视图。

（1）打开 DS-5 视图。使用主菜单中的视图工具栏或者选择【Window】→【Open Perspective】→【DS-5 Debug】来切换视图。

（2）单击【Window】→【Show View】→【Other...】，打开显示视图对话框。

（3）在【Remote Systems】组中选择【Remote Systems 视图】。

（4）单击【OK】按钮。

2. 第二步

在 Remote Systems 视图中，设置一个使用 SSH 连接远程目标的 Linux 连接：

（1）在 Remote Systems 视图工具栏中单击【Define a connection to remote system】。

（2）在 Remote System 类型对话框中，展开【General】组并选择【Linux】，如图 2-1 所示。

（3）单击【Next】按钮。

（4）在 Remote Linux System Connection 对话框的 Host name 编辑框中输入远程目标板的 IP 地址或计算机名，如图 2-2 所示。

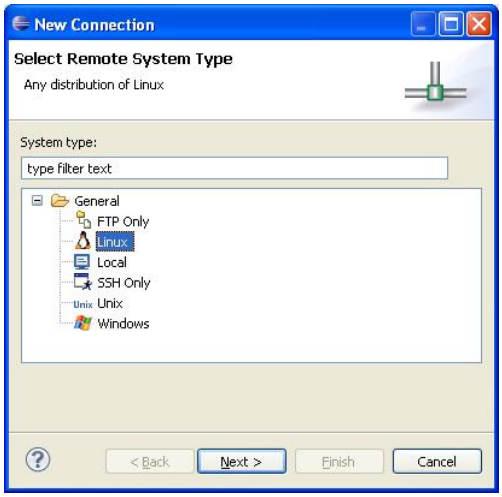

图 2-1　选择连接类型

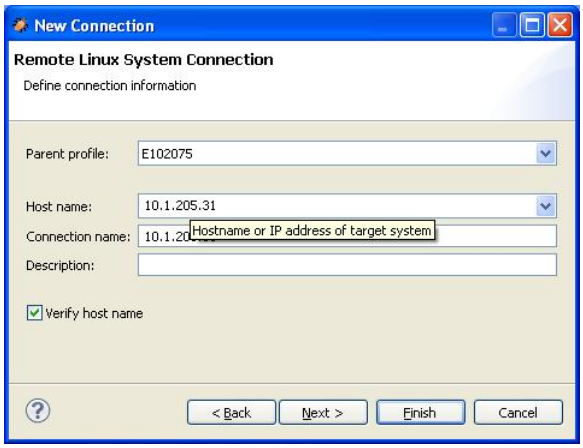

图 2-2　定义连接信息

（5）单击【Next】按钮。

（6）选择 SSH 协议文件访问，如图 2-3 所示。

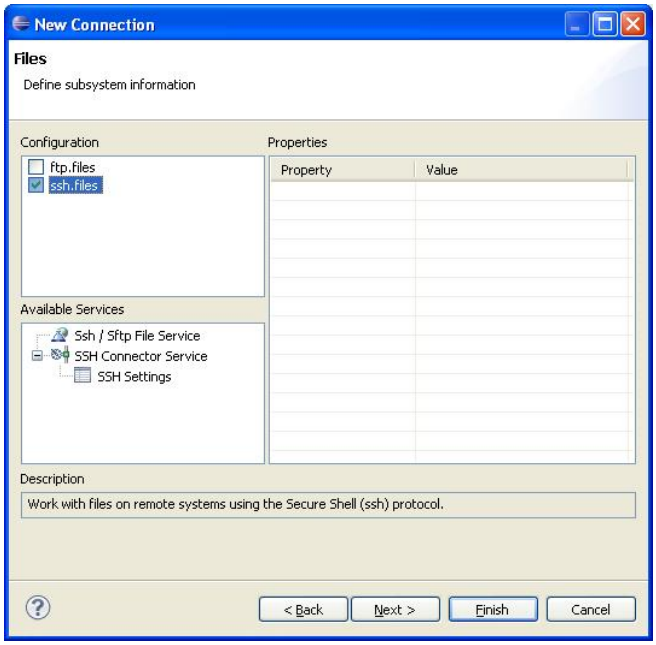

图 2-3　定义文件系统

（7）单击【Next】按钮。

（8）为 Linux 系统选择 shell 协议，如图 2-4 所示。

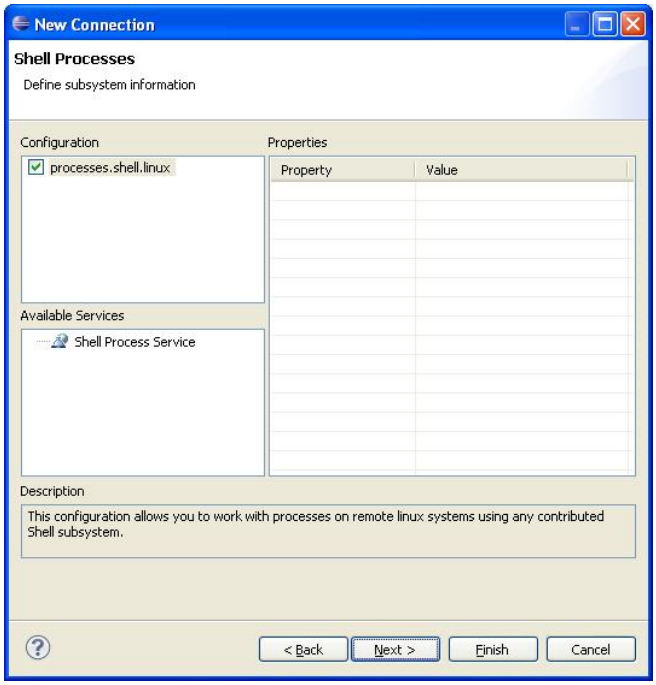

图 2-4　定义连接

（9）单击【Next】按钮。

（10）选择 SSH shells，如图 2-5 所示。

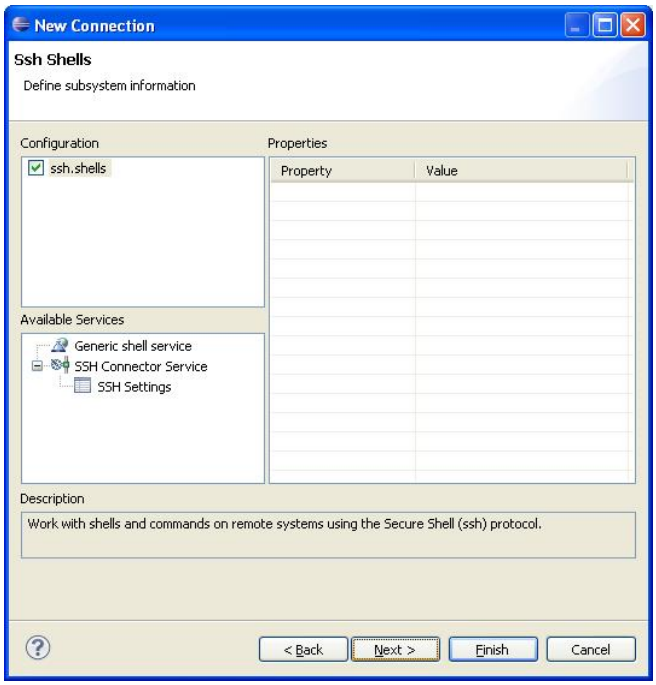

图 2-5　定义 Shell 服务

（11）单击【Next】按钮。

（12）选择 SSH Terminals，如图 2-6 所示。

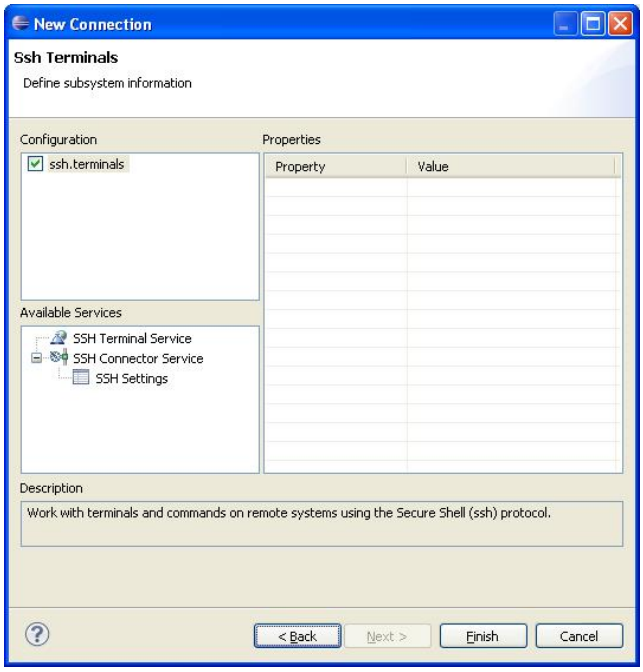

图 2-6　定义终端服务

（13）单击【Finish】按钮。

3. 第三步

在 Remote Systems 视图中:

（1）在 Linux 目标设备上右键，并从上下文菜单中选择【Connect】。

（2）如果需要，在密码输入框中输入用户名和密码。

（3）单击【OK】按钮关闭对话框。

（4）从本地文件系统将剥离版本的 gnometris 应用程序、gnometris 以及 libgames-support.so 库文件复制到目标板文件系统上。

（5）确保目标文件有执行的权限。要做到这一点，请右击每一个文件，从上下文菜单中选择【Properties】，并更改所需的复选框，如图 2-7 所示。

4. 第四步

打开一个已经连接到目标板终端的 Shell，并用 gdbserver 启动应用程序：

（1）在 Remote Systems 视图中，右击【SSH Terminals】。

（2）选择【Launch Terminal】打开终端 shell。

（3）在终端 shell 下，切换到 gnometris 程序所在目录，并执行以下命令：

```
export DISPLAY=ip:0.0
gdbserver: port gnometris
```

其中，ip 是显示 Gnometris 游戏的主机的 IP 地址；port 是 gdbserver 和程序之间的连接端口，例如 5000。

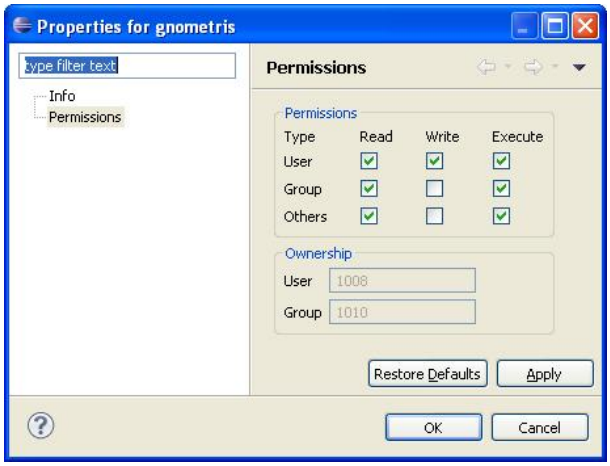

图 2-7　在 Remote Systems 视图修改文件属性

如果目标板有一个您可以使用的显示屏，那么你并不需 export DISPLAY。

2.7　连接已经在 ARM Linux 上运行的 Gnometris 程序

连接已经在 ARM Linux 上运行的 Gnometris 程序的步骤如下：

（1）从【Run】菜单中选择【Debug Configurations...】。

（2）从树形配置中选择 DS-5 Debugger 并单击【New】创建一个新的配置。也可以从工具栏中单击【Duplicate】选择一个已经存在的 DS-5 Debugger 配置 。

（3）在【Name】框中，为新的配置输入一个合适的名称。

（4）单击【Connection】选项卡查看目标板和连接选项。

（5）在 Select target 面板：

1）选择需要的平台，例如 beagleboard.org - OMAP_3530。

2）选择采用 Connect to already running gdbserver 进行调试。

（6）在 Connections 面板中，配置 gdbserver 和程序之间的连接，如图 2-8 所示。

1）输入目标板的 IP 地址。

2）输入端口号。

（7）单击【Files】选项卡，查看文件操作，如图 2-9 所示。

（8）在 Files 面板中：

1）选择【Load symbols from file】并选择包含调试信息的程序镜像，例如 H:\workspace\gnometris\gnometris。

2）单击【Add a new resource to the list】增加另一个文件目录。

3）选择【Load symbols from file】，然后选择 Gnometris 需要的共享库，例如 H:\workspace\gnometris\libgames-support.so。

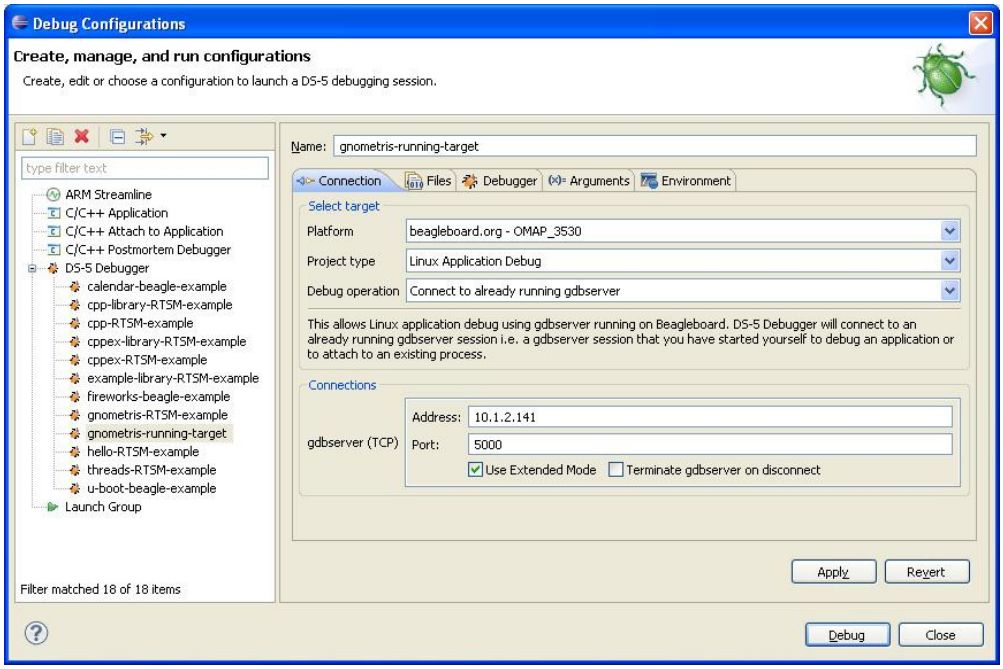

图 2-8　Beagle 开发板典型的连接配置

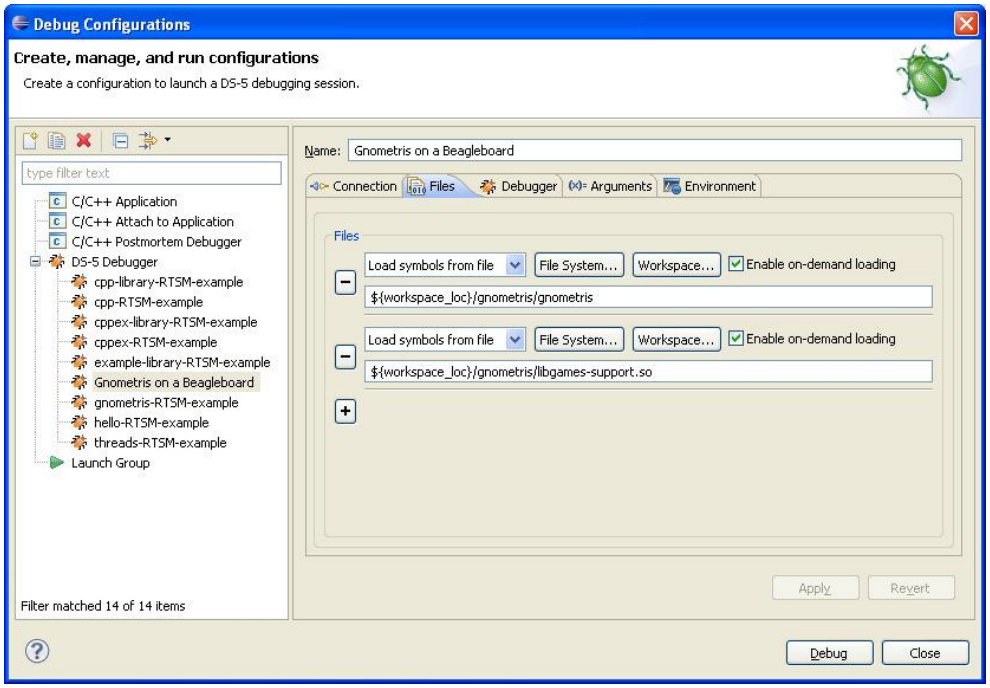

图 2-9　Beagle 开发板典型的文件选择

（9）单击【Debugger】选项卡查看配置的调试选项，如图 2-10 所示。

（10）在 Run 控制面板中：

1）选择【Debug from symbol】。

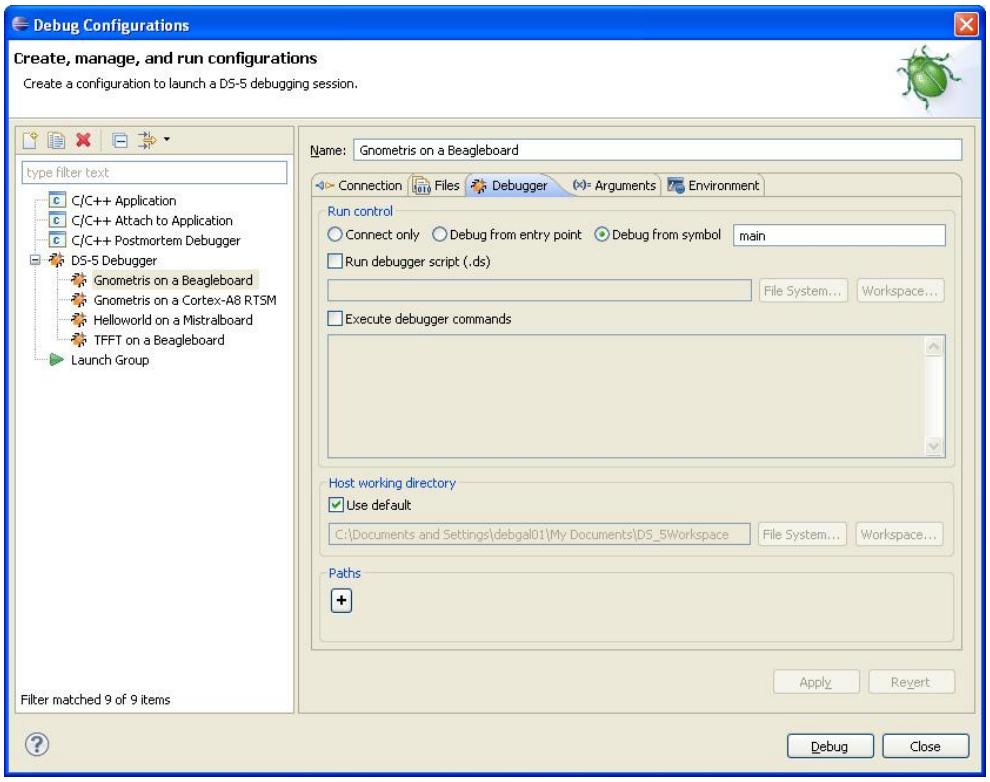

图 2-10　Beagle 开发板典型的 Debugger 设置

2）在提供的编辑框中输入 main。

（11）在 Host working directory 面板中，选择【Use default】复选框。

（12）单击【Debug】按钮开始调试，运行到 main() 函数。

（13）调试需要 DS-5 Debug 视图。如果出现确认切换视图对话框，单击【Yes】按钮切换视图。

2.8　调试 Gnometris

调试 Gnometris 程序：

（1）确保连接到了目标板，Gnometris 正在运行，且调试器停在 main() 函数等待。

（2）在 Project Explorer 视图，打开【Gnometris】目录查看源文件。

（3）双击【blockops-noclutter.cpp】打开程序的源代码文件。

（4）在 blockops-noclutter.c 文件中，找到 BlockOps::rotateBlock()这一行，双击 C/C++ 编辑器左侧的竖线添加一个断点。一个标记会放置在编辑器中的竖线处并且 Breakpoints 视图会更新显示新的信息。

（5）在 Debug Control 视图中单击【Continue】，继续运行程序。

（6）在目标板上启动一个新的 Genometris 游戏。在那个方块出现时，按向上的光标键命中断点。

（7）选择【Registers】视图，查看寄存器的值。

（8）选择【Disassembly】视图查看反汇编指令。在这一视图中，也可以双击侧竖线给单个指令设置断点。

（9）在 Debug Control 视图，单击【Step Over Source Line】移动到源文件下一行。并更新所有的视图。

（10）选择【History】视图查看在当前调试会话中所有的调试命令。可以选择一个或多个命令，然后单击【Exports the selected lines as a script】，为以后的使用创建脚本文件。

2.9 调试可装载的内核模块

可以使用 DS-5 开发和调试可装载内核模块。在开发过程中运行，可装载模块可以动态插入和删除，而不需要频繁地重新编译内核。

DS-5 提供了一个简单的字符设备驱动程序的例子-modex.c，可以编译，并在目标板上运行、调试。已构建的二进制镜像文件是提供给 Windows 用户的，镜像文件对应 DS-5 所提供的 Linux 发行版本项目。另外，查看 kernel_module 示例提供的 readme.html ，可了解 kernel_module 更多关于如何编译内核和模块的信息。

2.9.1 预备知识

在调试模块之前，必须确保：

- 解压缩并编译与目标板内核版本完全一样的源代码。
- 编译与目标板内核版本完全一样的可装载模块源代码。

确保您编译的镜像文件包含调试信息。调试模块的时候，调试器需要镜像运行时的信息。

2.9.2 步骤

调试可装载模块 modex.c：

（1）连接 debugger 到目标板。设备驱动示例提供了预先配置好的装载文件：

1）从【Run】菜单中选择【Debug Configurations...】。

2）在配置树中展开 DS-5 Debugger。

3）选择【module-beagle-example】。

4）Connection 选项卡包含除硬件地址外的大部分连接设置。输入调试器与调试硬件代理连接的 IP 地址或名称，如图 2-11 所示。

5）Files 选项卡包含 Linux 内核和模块装载调试信息的调试设置。在示例中，忽略了从主机上下载应用程序。选择内核映像和模块映像，如图 2-12 所示。

6）在 Debugger 选项卡的 Run control 面板中选择【Connect only】。

7）单击【Debug】，连接 debugger 到目标板上。

（2）配置并连接终端 shell 到目标板。可以使用 DS-5 提供的 Remote System Explorer（RSE）。

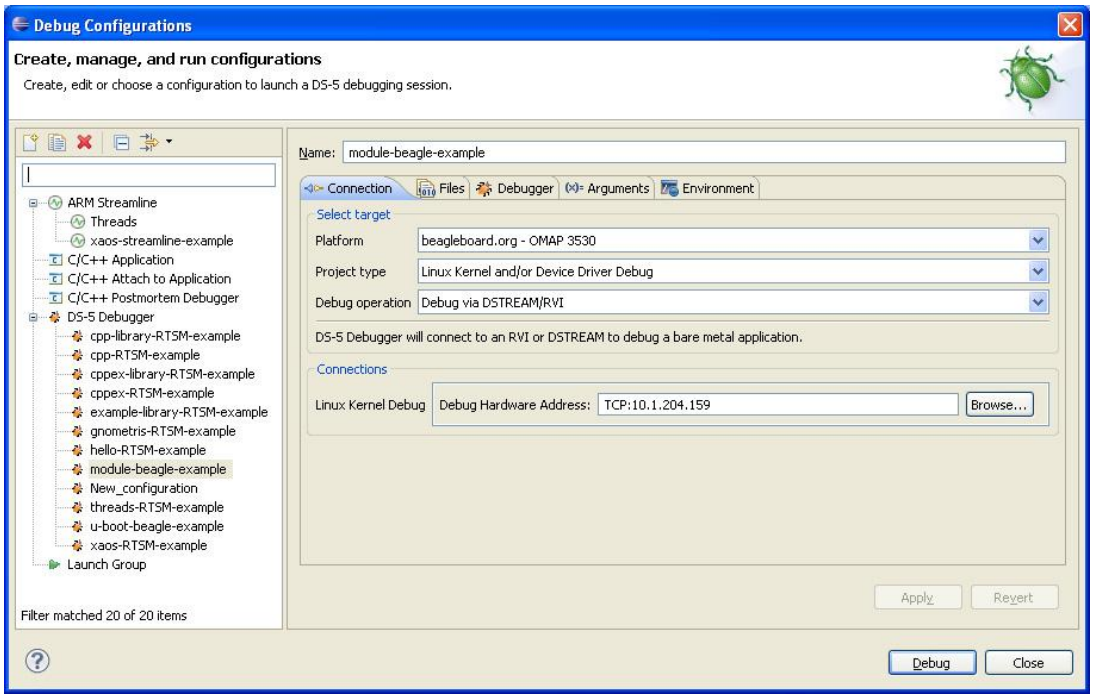

图 2-11　Linux 内核模块配置的典型连接

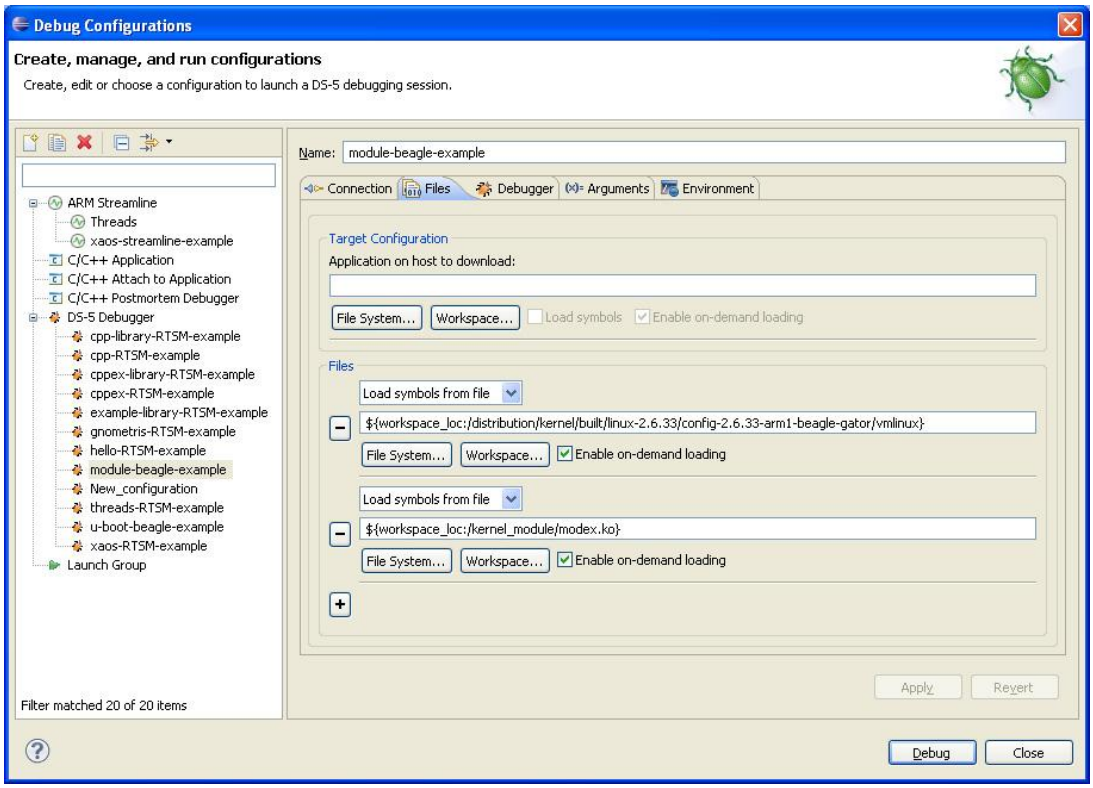

图 2-12　Linux 内核模块配置的典型文件选择

（3）使用 RSE，复制编译好的模块到目标板上。

1）在 Host 主机上切换到.../linux_system/kernel_module/stripped 目录。

2）拖放 modex.ko 模块到目标板上可写的文件夹里。

（4）在终端 shell 下，插入模块。

1）切换到模块所在的目录。

2）执行如下命令：insmod modex.ko。

3）Modules 视图进行更新，显示装载模块的信息。

（5）调试这个模块，设置断点并运行，需要的时候单步执行。

（6）修改模块的源代码：

1）移除这个模块。例如：rmmod modex。

2）重新编译这个模块。

3）如果需要，重复第（3）～（5）步。

插入和删除模块时，调试器停止目标程序，并自动解析调试信息和现有断点的内存地址。这意味着当重新编译源代码时不必停止调试器并重新连接。

可用的终端 shell 命令如下：

- lsmod 显示所有已装载模块的信息。
- insmod 插入可装载模块。
- rmmod 移除模块。

可用的 DS-5 Debugger 命令如下：

- info os-modules 显示连接后装载的操作系统模块。
- info os-log 显示操作系统日志缓冲区内容。
- info os-version 显示操作系统的版本信息。
- info processes 显示进程 ID、当前状态和相关堆栈帧的信息。
- set os-log-capture 控制操作系统日志信息捕获和打印到控制台。

连接后，装载的操作系统模块显示在 Modules 视图中。

2.10 运行在 ARM Linux 上的应用程序的线程性能分析

ARM Streamline 是一个图形化的性能分析工具。它提供时间线和分析报告，在系统、进程和线程级突出标识问题部分和应用程序热点。

2.10.1 预备知识

捕获分析数据之前，应确保：

（1）获取目标板的 IP 地址。可以在 Linux 终端上使用 ifconfig 配置程序； IP 地址由 inet addr 标识。

（2）ARM Linux 内核已经为 Streamline 添加了相应配置选项并完成编译。

（3）把线程程序复制到目标板上。

（4）gator 守护程序正运行在目标板上。

2.10.2 步骤

捕获数据的步骤如下：

（1）启动 Elipse。

（2）启动一个终端 shell 并连接到目标板。可以使用 Remote System Explorer (RSE)提供的终端 shell。

（3）在终端 shell 中，切换到复制了 threads 程序的目录下。

（4）确保在 Streamline Data 视图下，若 Streamline Data 视图未打开，可选择【Window】下的【Show View】里的【ARM Streamline Data】选项。

（5）创建新的连接。

1）在 ARM Streamline Data 视图中选择【Capture options...】工具栏图标。

2）在 Name 框中，为新建的连接输入一个合适的名称。

3）在 Connection 面板中，为主机和目标板之间的连接输入 IP 地址或名称和相应的端口号。

4）在 Capture 面板中，使用默认配置，如果需要的话，也可以自定义配置。

5）在 Program Images 面板中单击【Add Program...】或【Add Program from Workspace...】打开对话框，选择程序镜像文件。

6）在弹出的对话框中选定 threads 应用程序，单击【Open】或【OK】按钮关闭对话框。

7）单击【Save】保存配置。

8）开始捕获数据，在 Streamline Capture Data 视图中单击【Start capture】工具栏图标。

（6）在终端 shell 下，执行下列命令运行线程程序：./threads。

（7）运行线程程序后，在 Eclipse 中返回到 C/C++ 视图。

（8）在 Streamline Capture Data 视图中单击【Report】对图形数据进行分析，如图 2-13 所示。

图 2-13 Streamline 捕获数据文件

停止捕获数据时，自动生成一个 Streamline 数据文件，也可以双击已有的分析文件，在编辑器中查看，如图 2-14 所示。

图 2-14 Streamline 分析数据文件

2.11 调试 Android 本地 C/C++应用程序和库

本节描述了如何调试 Android Native Development Kit（NDK）提供的 hello-neon 程序。它

使用 Android SDK Platform 2.2 和 Android 目标板模拟器。

文档没描述如何安装 Android 工具。要获取更多信息，参考 Android 开发网站。

2.11.1 预备知识

在调试包含本地 C/C++代码的 Android 包前，必须做如下工作：

（1）下载和安装 Android Software Development Kit (SDK)。安装之后，可以将附带任何本地 C/C++代码的 Java 程序编译到以.apk 为文件扩展名的 Android 包中。

（2）下载和安装 Android NDK。这是一个 Android SDK 的配套工具，这个工具能够编译应用程序的关键性能部分的代码，如 C 和 C++代码。

在 Windows 下，必须下载和安装 cygwin，包括 make 包，这样就可以在 Android NDK 中运行脚本。

（3）将 DS-5 arm_directory\gdbserver\...\android 目录提供的 Android 版复制到相关的 Android NDK 工具链目录下，更新 gdbserver。本教程中的目录为...\toolchains\arm-eabi-4.4.0\prebuilt。

（4）设置 Eclipse 的 Android 插件：

1）启动 Eclipse。

2）安装 Android Development Tools (ADT) Eclipse 插件。例如，从下面的网 站下载 http://dl-ssl.google.com/android/eclipse。

3）选择【Window】→【Preferences】→【Android】并单击【Browse...】选择 Android SDK 的安装目录。

4）选择【Window】→【Android SDK and AVD Manager】，打开 Android SDK and AVD 管理对话框。

5）展开 Available packages 组，并添加需要的 SDK 平台。例如，Android SDK Platform Android 2.2。

6）创建一个新的 Android Virtual Device (AVD)。

（5）编辑 Android NDK 脚本文件，ndk-gdb 使用 DS-5 进行调试。Android NDK 包含一个脚本文件，在目标板上启动调试器前来运行 gdbserver 和应用程序。默认的，脚本文件没有设置用 DS-5 调试。要改变这种行为，你必须注释掉最后一行，如下所示：

```
#$GDBCLIENT -x `native_path $GDBSETUP
```

2.11.2 步骤

调试程序步骤如下：

（1）使用 Android NDK 提供的脚本文件编译（包含调试信息）hello-neon 源文件 。本书使用...\toolchains\arm-eabi-4.4.0\prebuilt 目录。例如：

```
./ndk-build -C samples/hello-neon NDK_TOOLCHAIN=arm-eabi-4.4.0 NDK_DEBUG=1
```

（2）启动 Eclipse。

（3）创建一个 Android 项目。

1）选择【File】→【New】→【Project...】。

2）展开 Android 组并选择【Android Project】。

3）单击【Next】。

4）输入一个合适的项目名，如 HelloNeon。

5）选择【Create project from your existing source】并找到 hello-neon 目录。

6）如果选中【Use default location】选项，项目将在显示的默认目录中创建。否则，取消此选项，并定位到首选项目目录。

7）选择需要的编译目标，例如，Android 2.2。

8）输入一个合适的程序名，例如，Hello, Neon。

9）输入一个合适的包名，例如，com.example.neon。

10）输入一个合适的 Activity 名，例如，HelloNeon。

11）单击【Finish】。

（4）确保应用程序编译时包含了调试信息。可以这样做：

1）打开 AndroidManifest.xml 文件。

2）单击【Application】选项卡。

3）在 Debuggable 框中选择 true。

4）保存更改并关闭该文件。

（5）清除并重新编译 Android 项目。

（6）如果应用程序已经安装在目标板上，必须卸载它。例如：

path\adb uninstall com.example.neon

（7）安装应用程序。例如：

path\adb install samples/hello-neon/bin/HelloNeon.apk

（8）运行 ndk-gdb 脚本启动程序并连接 gdbserver。例如：

./ndk-gdb --project=samples/hello-neon --verbose --port=5000 --start --force --adb=adb

关于使用脚本文件和命令行选项的更多信息，请参阅 Android NDK 的文档。

（9）使用 gdbserver TCP 连接 DS-5 和应用程序。

1）从【Run】菜单中选择【Debug Configurations...】。

2）从配置树中选择【DS-5 Debugger】并单击【New】创建一个新的连接。此外，可以选择一个已经存在的 DS-5 Debugger 配置，并单击工具栏的【Duplicate】。

3）在【Name】字段中，为新的配置输入一个合适的名称。

4）单击【Connection】选项卡，配置 DS-5 调试器目标板连接，如图 2-15 所示。

5）单击【Files】选项卡，选择【app_process object】文件，如图 2-16 所示。

6）单击【Debugger】选项卡，在 Run Control 面板中选择【Connect only】。

7）选择【Execute debugger commands】，在相关的文本框中输入 sharedlibrary，从所有共享库中装载调试信息到调试器。

8）在 Paths 面板中，指定主机上共享库的搜索目录，调试器显示源代码时会使用到，如图 2-17 所示。

9）单击【Debug】连接目标板。

（10）调试需要 DS-5 Debug 视图。如果出现确认切换视图对话框，单击【Yes】切换视图。

（11）要调试应用程序，根据需要设置断点、运行和单步执行。

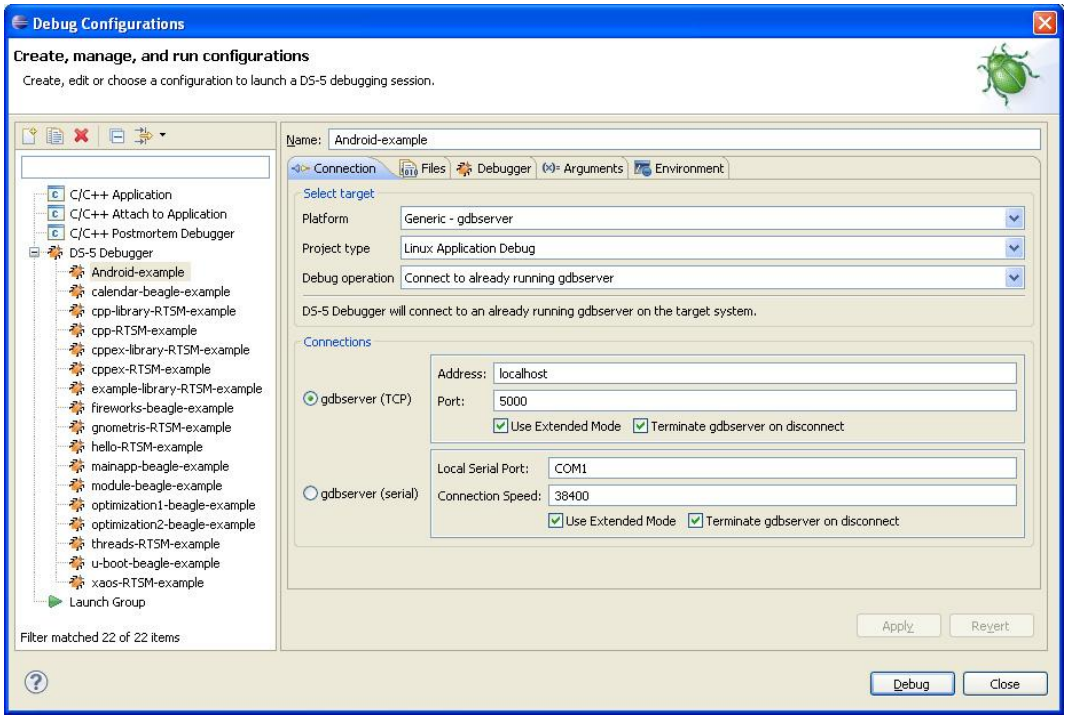

图 2-15　Android 应用程序的典型连接配置

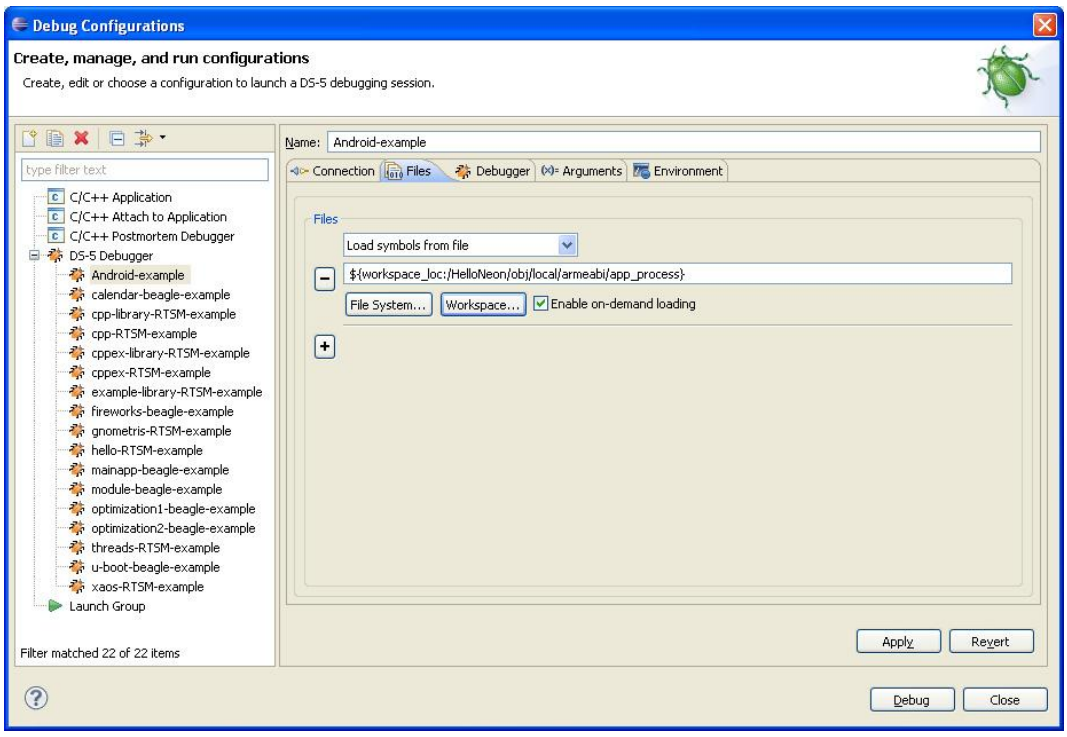

图 2-16　Android 应用程序典型的文件选择

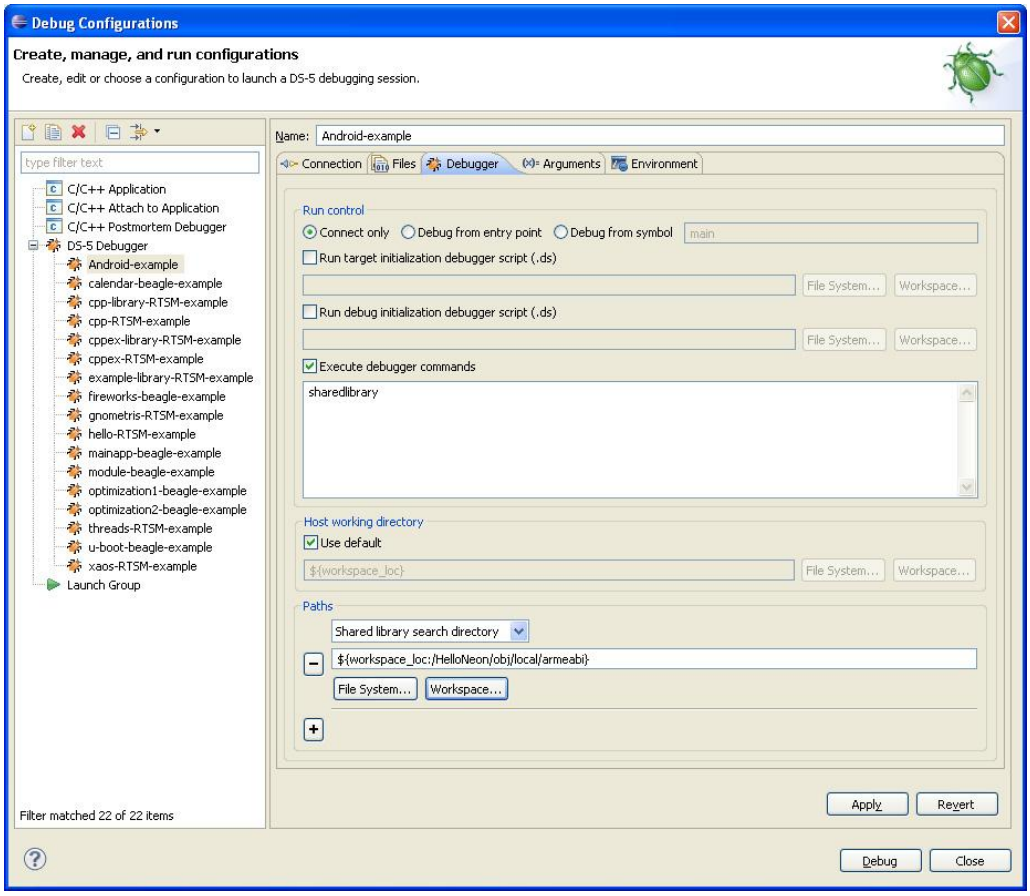

图 2-17　Android 应用程序典型的调试器连接设置

2.12　DS–5 许可管理

在 Eclipse 中，可以管理 DS-5 许可，从【Help】菜单中选择【ARM License Manager...】，安装的许可显示在 ARM License Manager 对话框中，如图 2-18 所示。

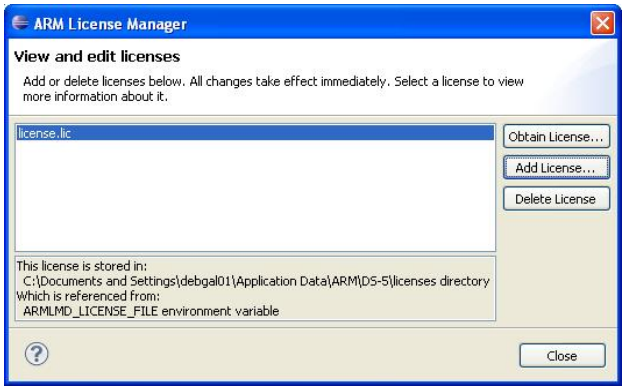

图 2-18　查看和编辑许可

- 单击【Obtain License...】按钮，按照对话框的提示申请一个新的许可，如图 2-19 所示。

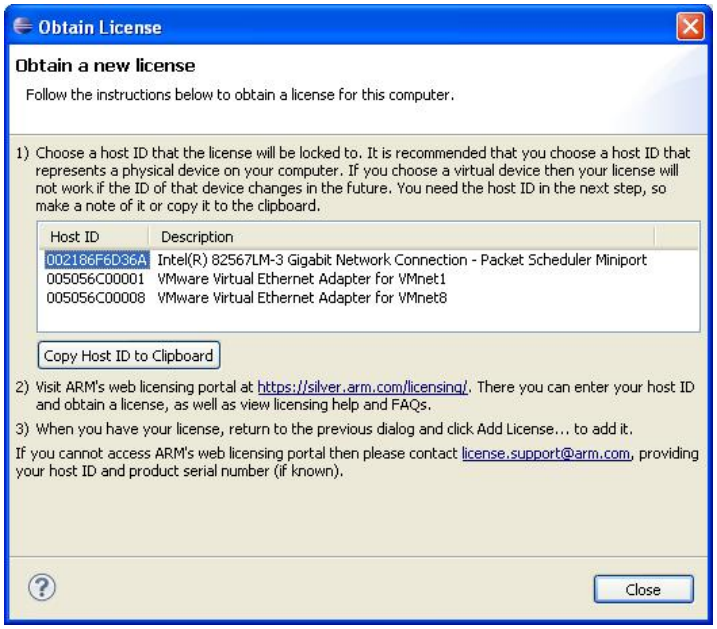

图 2-19　获取一个新的许可

- 单击【Add License...】按钮，安装一个新的许可。在 Windows 下，许可文件复制到 %APPDATA%\ARM\DS-5\licenses 文件夹；在 Linux 下，许可文件复制到 $HOME/.ds-5/licenses 文件夹。服务器许可证可以分别输入主机和端口字段，或者可在 Host 框中输入完整的字符串 port@host，如图 2-20 所示。

图 2-20　增加一个新许可

- 单击【Delete License】按钮卸载许可，并从 DS-5 许可文件夹中删除文件。