

# 建立 windows 下 djyos for arm 编译和调试环境

## 1. 编译环境

### 1.1. 安装 gcc 编译环境

winarm 和 gnuarm 是比较流行的 windows 下的 gcc 编译环境，这两个编译环境均无需单独安装 cygwin 就能工作，但是，实践证明，如果不安装 cygwin，makefile 比较复杂时，可能会出问题。djyosv0.3.0 版本以及更早版本，一直使用的是 winarm 作为编译器，但这次调整 djyos 目录结构并重新编写 makefile 的过程中，发现 winarm 有如下缺陷：

- 1、版本升级似乎已经停止，其官网上的最新版本是 winarm20060606，已经近三年未更新了，期间虽然出过一个 winarm20080331-testing 非正式版本，但正式版本一直没出来。导致 winarm 不能支持 cortex-m3 处理器。
- 2、winarm 不能识别在 makefile 中用\$(shell pwd)生成的类似/cygdriver/d/格式的路径名，只认识 windows 格式的路径名。

上述问题在 gnuarm 中并不存在，其版本一直在更新，编译器版本已经升级到 arm-elf-gcc4.3.2，因此从 djyos0.4.0 开始，使用 gnuarm 作为编译工具。

首先到这里：[http://www.gnuarm.org/bu-2.19\\_gcc-4.3.2-c-c++\\_nl-1.16.0\\_gi-6.8.exe](http://www.gnuarm.org/bu-2.19_gcc-4.3.2-c-c++_nl-1.16.0_gi-6.8.exe) 下载 gnuarm 编译器，然后安装，一路点 next，运行到出现下图时：

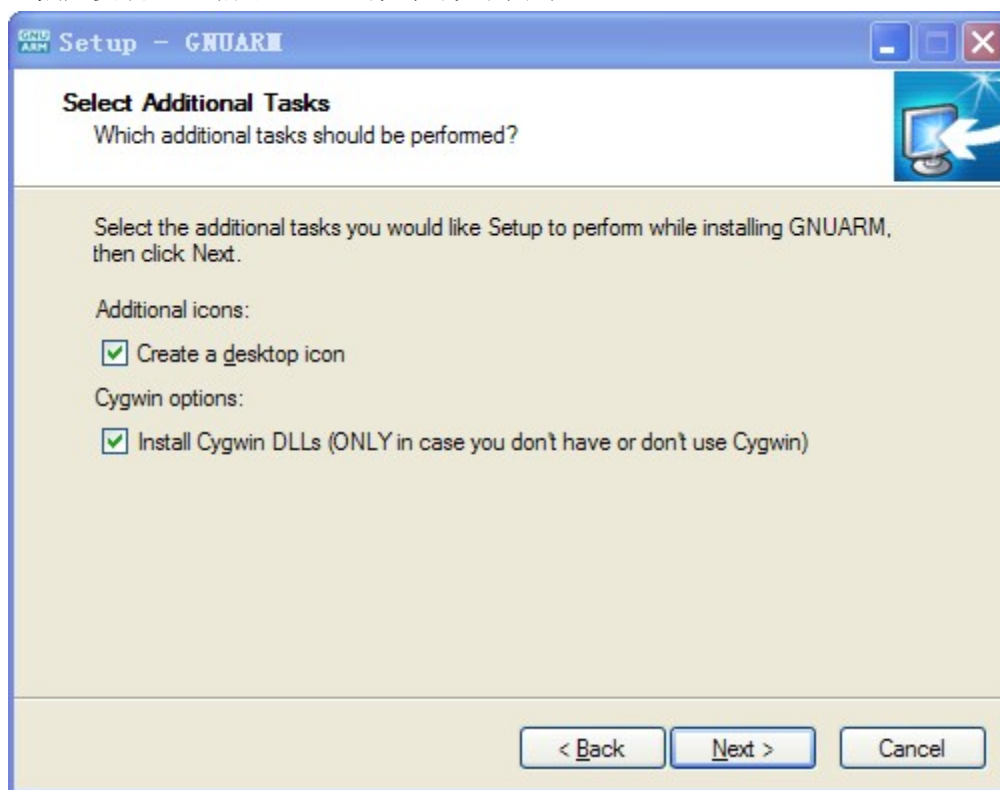


图 1

务必确保“install Cygwin DLLs”框选中，否则可能出现 cygwin 版本不匹配的错误，然后在一路 next 下去即可。

然后安装cygwin, 从地址: <http://www.cygwin.com/setup.exe> 下载setup.exe后, 点击运行, 程序会从网上直接下载安装所需的文件并按照, 一路按默认设置安装就可以了。

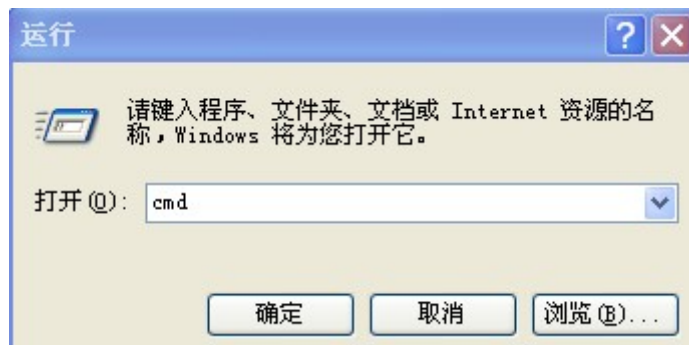
别以为到此就结束了, 无论是gnuarm还是cygwin, 都没有为你准备make.exe文件, 天哪, 没有make文件怎么编译啊? 不急, gnu的make.exe文件是通用的, 你可以到这里: <http://www.djyos.com/download/make.rar> 下载一个, 解压缩到c:\cygwin\bin目录或者C:\Program Files\GNUARM\bin目录下都可以。(假设上述安装过程都是按默认设置安装的)。

最后一步, 这一步非常简单, 但却机关重重, 我经过反复摸索才发现其中的奥秘。cygwin安装后, 并没有把c:\cygwin\bin添加到windows的path环境变量中, 需要手工添加, 而gnuarm的目录是自动添加的, 这里的机关是, 必须把C:\Program Files\GNUARM\bin放在c:\cygwin\bin的前面, 否则可能报cygwin版本不正确的错误, 最好是把他们放在path的最前头。

至此, 编译环境已经全部建立起来了。

## 1.2. 使用 gcc 编译

编译环境建立起来后, 就可以用它来编译djyos了, 在windows下, 点“开始-运行”, 在运行对话框:



中输入 cmd, 点“确定”进入命令行方式, 然后进入源程序所在目录 (makefile 文件也在这个目录中), **注意, 不是进入安装 cygwin 时带的“Cygwin Bash Shell”环境**。在这个目录中执行 make 命令即可编译。

## 2. jlink 调试环境

djyos 目前并没有实现调试代理, 不能通过串口、网口等通信口调试。可以用硬件烧录设备直接把源码包中的 run\_inram.bin 或者 run\_inflash.bin 烧录到 flash 中直接运行, 但这样只能看到运行结果。如果想源代码跟踪操作系统的启动和运行过程, 或者修改调试代码, 就必须使用 jtag 仿真器和调试器。这里讲讲 jlink+insight 或 jlink+realview2.2 两种调试环境的安装和配置。

许多开发板会带一个并口的调试工具, 如果你用台式机且有并口, 可以按本文稍后介绍的方法。用笔记本调试的话, 许多本本没有并口和串口, 而 djyos 目前只能通过串口和超级终端通信, 所以, 我们需要一个 usb 接口的仿真器, 还要一个 usb 转串口的工具。推荐几种开发工具:

- 1、 jlinkv7+usb 转串口工具, 这是一种 jlink 仿真和 usb 转串口双功能的仿真器, 串口和仿真器可以同时使用, 可节省一个 usb 口, 对一些 usb 口较少的机器还是很实用的。在淘宝上搜索“jlinkv7+usb 转串口”可以找到。
- 2、 如果你不需要串口, 可买不带串口的, 便宜一些, 但一定要选支持 GDBFULL 的, 才能用 insight 调试, 在淘宝上搜“jlinkv7”可以找到。
- 3、 上述工具适用于在 windows 下开发 (包括在 windows 下使用 GDB insight 开发), 如果你在 linux 环境下开发, 应该买 OpenJTAG, 在淘宝上搜索“openjtag”可以找到。

**注意, 开发板的 jtag 口有三种, 分别是 20 芯 2.54 间距、20 芯 2.0 间距, 10 芯 2.0 间距, 大多**

数是 20 芯 2.54 间距的，但也有其他接口的，我用的 mini2440 是 10 芯 2.0 间距的，有些仿真器只带 20 芯 2.54 间距的仿真线，造成有些开发板不能使用，最好买带 3 种 jtag 口的。

## 2.1. 安装 jlink 驱动

jlink 驱动程序版本更新比较频繁，可到[http://www.segger.com/download\\_jlink.html](http://www.segger.com/download_jlink.html) 下载最新版本驱动程序，新买回来的光盘上有一个 4.02 版本的驱动程序，一般来说够用的。

安装前，切记不要连接硬件，点击驱动程序可执行文件：Setup\_JLinkARM\_V402.exe，一路按默认设置安装即可。

## 2.2. 安装串口驱动

光盘中“windows\CDM 2.04.14”里是 USB 转串口的驱动程序，检测到“USB Serial Port”新硬件时，指定驱动程序的查找路径为这个目录即可。

## 2.3. 调试准备

step1: 按与本文同时发布，相同方法下载的《djyos 的 si 版本 makefile 文件说明.doc》一文，编译 boot\_rom.bin 和 debug.elf。boot\_rom.bin 在发布的文件包中有，如果没有修改代码的话，可以直接使用。

step2: 用 jlink 附带的 JFlashARM 把 boot\_rom.bin 烧录到 0 地址的 flash 中。

step3: 退出 JFlashARM 或者按 F9 让出 jtag 控制权，这一步很重要，否则仿真器无法控制 cpu。

step4: 复位目标机。

## 2.4. 配置 gdb insight 调试环境

gdb insight 是图形化的 gdb 调试工具，前面安装的 jlink 驱动，已经包含一个 GDBServer，但该 GDBServer 的默认配置是固定为 Big endian，如果硬件是按小端设计的（绝大多数 ARM 开发板的默认配置是小端），应该先修改 JLinkGDBServer.ini 文件，以下是用 insight 调试的步骤。

### 2.4.1. step1: 修改 GDBServer 的配置文件

进入 jlink 的安装路径：

C:\Program Files\SEGGER\JLinkARM\_V402

把 Endian = 1 这一行改为：

Endian = 0

### 2.4.2. step2: 启动 GDBServer

在安装路径下，双击执行 JLinkGDBServer.exe。

可看到如下界面，Initial JTAG speed 的默认设置是 5Khz，比蜗牛还慢，有点好笑。如果你用的是 ARM7TDMI-S 内核的 LPC2000 系列处理器（djyos 将来会移植到此处理器的），速度设为 48000Khz，其他可设为 12000Khz。

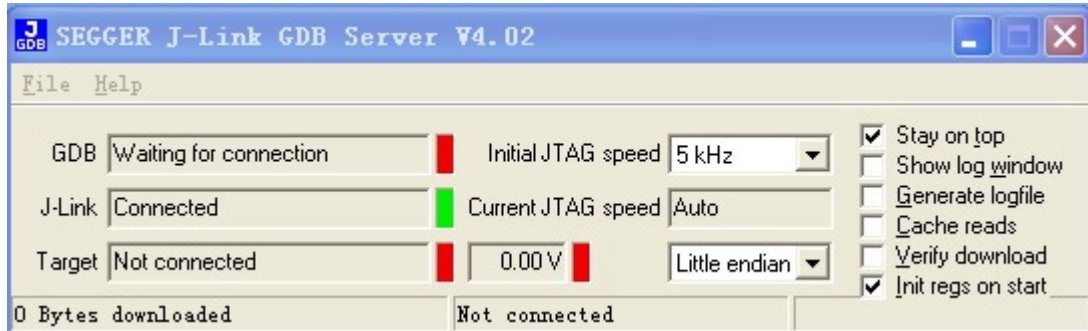


图 2

### 2.4.3. step3: 启动 insight

前面安装 gnuarm 时已经安装，点击 c:\cygwin\bin 目录下的 arm-elf-insight 运行，可看到如下界面：

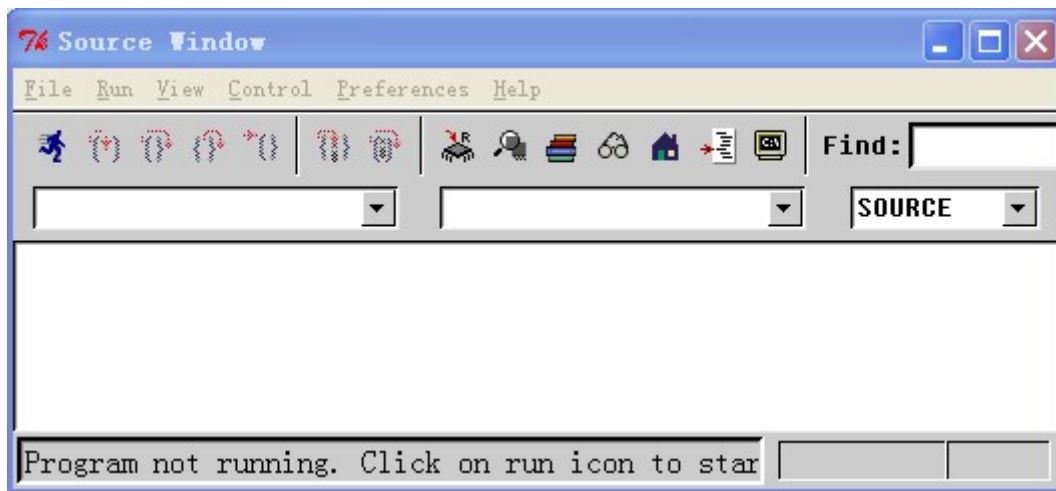


图 3

点 file-Target settings 菜单，在弹出的界面中做如下配置：

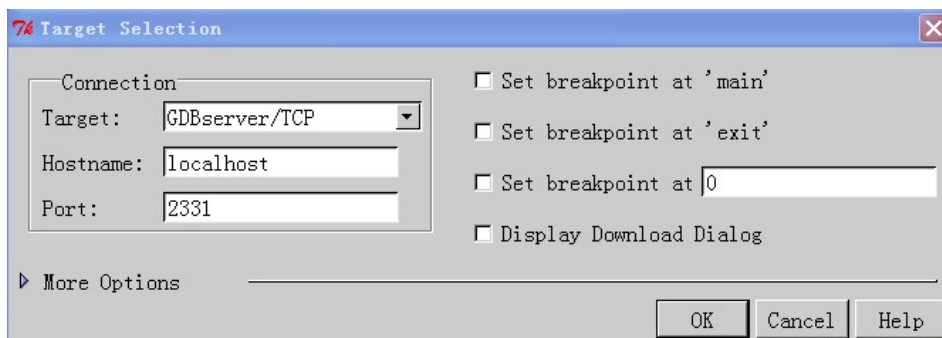
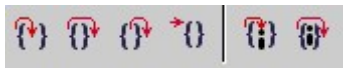


图 4

再点 file-open 菜单，打开工程目录下的 debug.elf 文件。

接着点菜单项 run-Connect to Target，再点击 run-download 如果这几个按钮



变亮了，恭喜你，你可以调试了，试一下单步执行吧。

## 2.5. 配置 realview2.2 的 RVDS 调试环境

realview 是 ARM 公司推出的集成调试环境，是有版权的，这里只讲如何用它来调试 djyos，不负责版权问题。在这里我们并不适用 realview 的工程编辑和编译环境，只用它的调试环境。用 RVDS

调试有一个缺陷，就是 RVDS 不能对 gcc 编译产生的 elf 文件中的汇编源文件进行源代码调试。不过这也不是大问题，djyos 中汇编极少。

首先安装 realview，按软件本身的说明即可。

安装好后，点击此图标执行RVDS调试器，出现图 5所示界面。

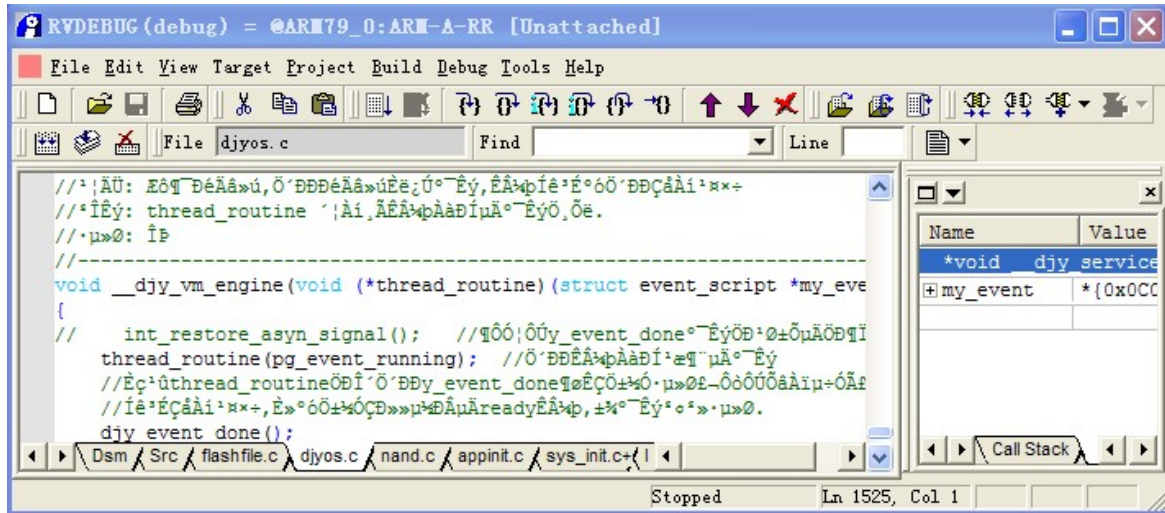



图 5

点击此图标连接仿真器，出现图 6界面，

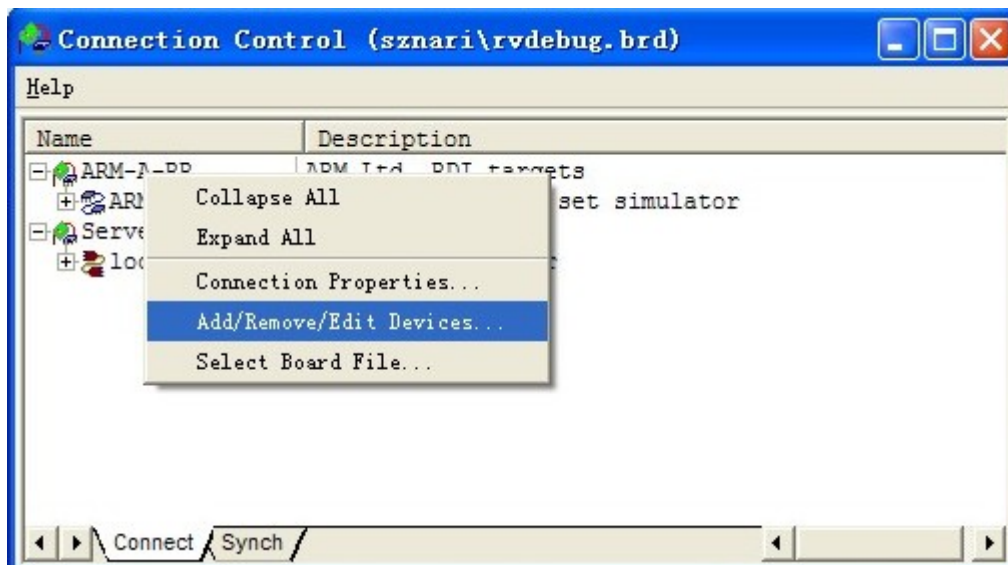


图 6

在 ARM-A-RR 中点击右键，在弹出菜单中选“Add/Remove/Edit Device...”，出现如下界面：

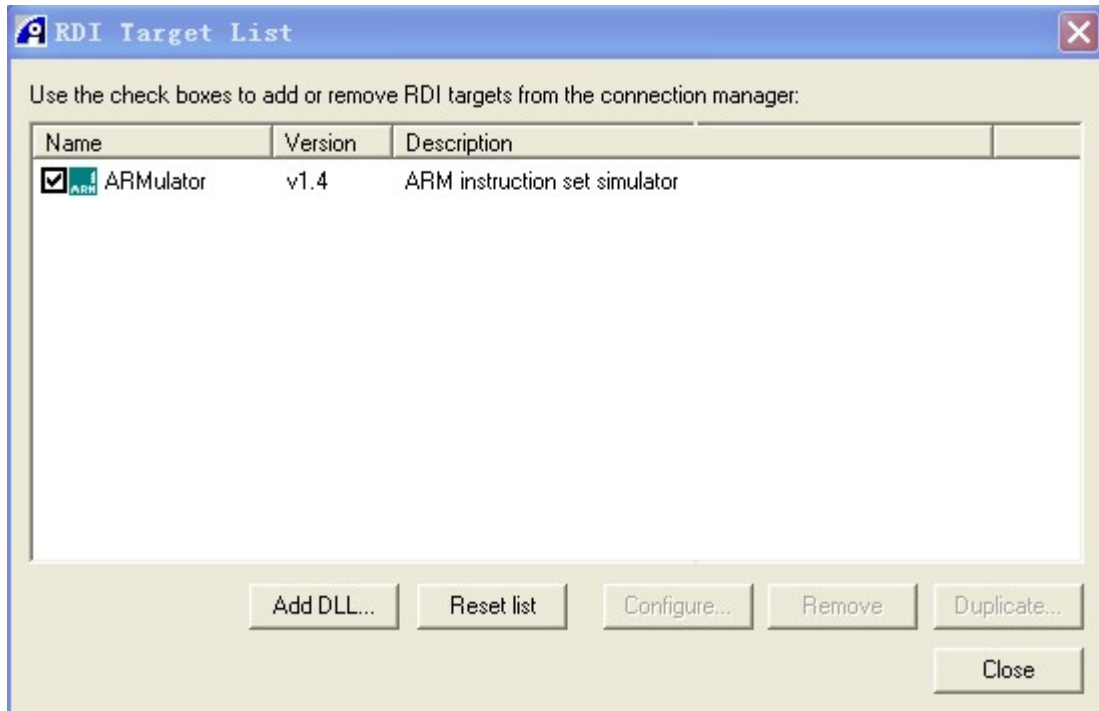
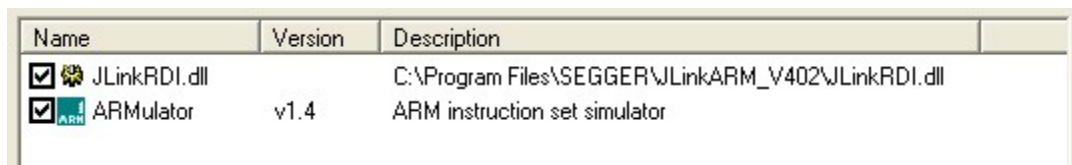


图 7

可以看到, ARM 公司预定义了一个调试目标——指令集模拟器, 由于该模拟器没有模拟定时器, djyos 必须使用定时器提供时钟脉冲, 故不能用来调试 djyos, 必须配置硬件仿真器。

点击 **Add DLL...** 图标, 选取 C:\Program Files\SEGGER\JLinkARM\_V402\JLinkRDI.dll 文件, 图 7 的调试目标列表将变为:



点击 close 关闭图 7 界面后, 图 6 界面将变成如下:

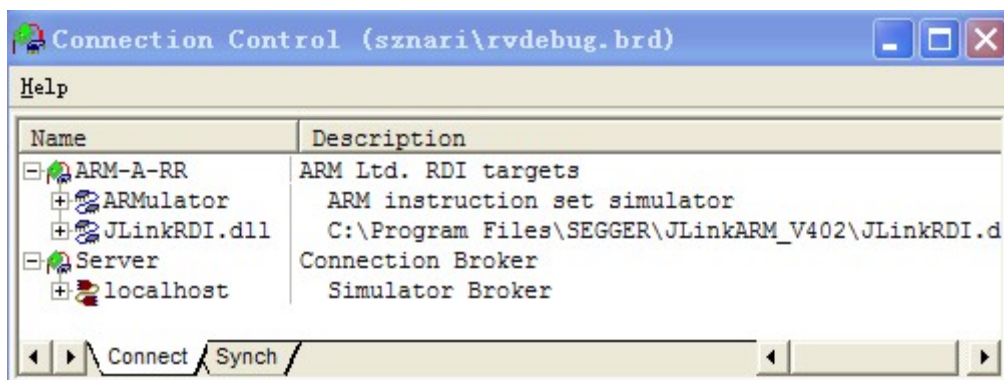
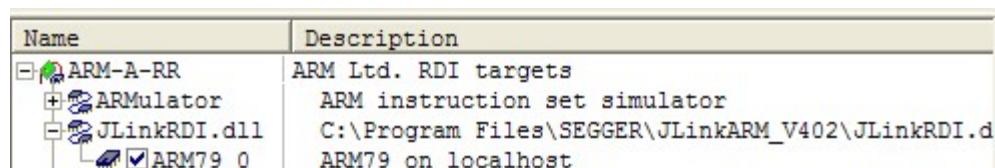


图 8

确保仿真器、目标板正常连接上电, 并且没有启动 JFlashARM, 点击 JLinkRDI.dll 前面的 + 号, 再点击展开后的方框, 当该方框变成选中状态后, 如图 9, 恭喜你, 可以加载代码调试了。如果点 + 号时出现错误, 可多点两次。



Name	Description
ARM-A-RR	ARM Ltd. RDI targets
ARMulator	ARM instruction set simulator
JLinkRDI.dll	C:\Program Files\SEGGER\JLinkARM_V402\JLinkRDI.d
<input checked="" type="checkbox"/> ARM79_0	ARM79 on localhost

图 9

至此，你的 jlink+rvds2.2 调试环境已经配置好，具体调试方法参见《运行与调试.doc》，该文档与本文档一起发布。

### 3. hjtag 调试环境

hjtag 是由 twentyone 开发的 jtag 调试软件，可以支持 realview 的 rvds 调试，但不支持 gdb 调试。在 [www.hjtag.com](http://www.hjtag.com) 网站上有详细的使用说明，包括在 rvds 下如何配置等内容，这里就不在赘述了。