win7 下使用 Eclipse 在线调试 linux 应用程序

英创公司

2017年3月

英创公司提供的 Eclipse 是经过英创公司修改和配置的开发工具,可以在 Windows 操作系统下开发 Linux 应用程序,不需要用户编写复杂的 Makefile 文件,大大加快了应用程序的开发进度。除此之外,此 Eclipse 也提供了在线调试的功能,只要进行相应设置,就能在 Eclipse 中在线调试运行在英创主板上的 linux 应用程序。

下面在 win7 环境中, 演示通过 Eclipse 在线调试 linux 应用程序, 目标板是 em9287, 运行的是 linux-4.1.14 操作系统, 测试程序是 step2_serialtest, 目标板与 PC 在同一网络。

一、进入调试界面的设置

1、编译 Debug 版本应用程序, 放入 nfs 服务器

Itcomm-ser	ver				
🗅 🞏 rstpapp					
👕 sqlite3test		New		1	
⊳ 🞏 step1_lcdte			,		
4 🚰 step2_seria		Go Into			
Binaries		Open in New Window			
⊳ 🗊 Include:		Conv	Ctrl+C		
Release		Pasta	Ctrl+V		
⊳ <u>in</u> em9280		Delete	Delete		
Serial.c	<u>~</u>		Delete		
▷ In Serial.h	_0	Remove from Context CtrI+Alt+S	hift+Down		
⊳ <u>k</u> c step2_s		Source	•		
b t ^{oo} stan2 tang		Move			
Steps_tcpst		Rename	F2		
▷ Step4_tcpci	è	Import			
⊳ sest_3G AT	2	Export 2编译而日			
⊳ s≌ test 4a	-				
test_ad	L	Build Project		-	
⊳ 📂 test_adam		Clean Project			
👕 test_alsa	81	Refresh	F5		
👕 test_camer		Close Project		1 选择编	译版木
⊳ 📂 test_can		Close Unrelated Projects		± 201+-4m	11 1 11X/T
👕 test_can_fili		Build Configurations	•	Set Active	✓ 1 Debug
test_can_rc		Make Targets	•	Manage	2 Release
b 📂 test_eta105		Index	+	5 11 L 41	Properties 🚿 Searc
test_eta107				Build All	r matched 102 of 242 ite
b > test_eta108		Show in Remote Systems view		Clean All	^ ·
test_eta202		Profiling Tools	•	Build Selected	
test_eta/15		Convert To		Infos (2 items)	
test_freetyp		Profile As	+		
A W Binarias		Debug As	+		
⇒ _{son} binaries		Run As	+		
		Compare With	+		
4 🕞 Debug		Restore from Local History			
≥ 🛃 helld	*	Run C/C++ Code Analysis			
<		Team	+		
		Properties	Alt+Enter		

图 1 编译 Debug 版本的应用程序

如图 1 所示,选中 step2_serialtest 项目,右键选择编译版本为 Debug,然后编译这个 项目,最后将 Debug/ step2_serialtest 考备到 nfs 服务器的根目录下,本次实验使用的是 D:/01.public。

2、使用 gdbserver 运行 Debug 版程序

在目标板上挂载 nfs 服务,并使用 gdbserver 监听运行 Debug 版程序,如图 2。 mount -t nfs -o nolock 192.168.201.81:/d/01.public /mnt/nfs gdbserver :47476 /mnt/nfs/step2_serialtest 47476 是 gdbserver 监听的端口, Eclipse 将连接到此端口

[root@EM9280 /mnt]#mount -t nfs -o nolock 192.168.201.81:/d/01.public /mnt/nfs
[root@EM9280 /mnt]#ls /mnt/nfs/step2_serialtest
/mt/nfs/step2_serialtest
[root@EM9280 /mnt]#gdbserver :47476 /mnt/nfs/step2_serialtest
Process /mnt/nfs/step2_serialtest wreated; pid = 203
Listening on port 47476 🌂 日标板收听器口
日小伙鱼竹桐口

图 2 目标板运行 gdbserver

3、设置 Eclipse 调试功能

Step3-1: 进入 Debug Configurations

左键选中项目 step2_serialtest, 在菜单栏选择 Run >> Debug Configurations..., 如 图 3 所示。

C/C++ - Eclipse			
File Edit Source Refactor Navigate Search Project	Run) Window Help	
🔁 🕶 🔛 🐘 📥 👋 🕶 🐔 🖬 💷 🖬 🔌 🔂 🕶	6 🕪	Resume	
(a	- 88	Suspend	
Project Explorer 🔀 📄 🛱		Terminate	
💼 autodial_em335x	14	Disconnect	ι.
bt_inquiry_rssi	3.	Step Into	
▷ 🎏 bt_SSP_server	P	Step Over	
bt_ssp_server_v1	_62	Step Return	
ecno	=>]	Run to Line	
▷ ﷺ ftpcInt	৵	Use Step Filters Shift+F5	ι.
👕 gprs232	Q	Run Ctrl+F11	
▷ 📂 memleaktest	猆	Debug F11	
⊳ 👺 mstp-lib		- -	
▷ 😂 np692_em928x		Run History	
▷ 🎏 rfcomm		Run As	
Freemann Strength		Run Configurations	
Free Server		Debug History	
> 🚰 rstpapp		Debug As	
≥ Step1 (dtest		Debug Configurations 3	
✓ Step2_serialtest 1		Toggle Breakpoint Ctrl+Shift+B	
⊳ ﷺ Binaries		Toggle Line Breakpoint	
Includes		Toggle Method Breakpoint	
🔺 🗁 Debug	66,	Toggle Watchpoint	
⊳ 🗟 Serial.o - [arm/le]	2	Skin All Breakpoints	
⊳ 🐝 step2_serialtest - [arm/le]	Sila	Remove All Breakpoints	
⊳ 🚮 Step2_SerialTest.o - [arm/le]	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	Preakpoint Types	
Release		breakpoint types	
▷ lini em9280_drivers.h	Q	External Tools	
> 🔓 Serial.cpp	_		_

图 3 进入 Debug 配置

Step3-2: 新建调试应用项目

在 Debug Configurations 对话框中选择 C/C++ Remote Application,点击左上角的新 建图标,新建一个调试应用,如图 4。

Debug Configurations				
Create, manage, and run configurations		1. Alton		
I New Jaunch configuration C C/C++ Application C est_eta108 C test_wrifi C C/C++ Attach to Application C C/C++ Remote Application C C/C++ Remote Application Launch Group 	Configure launch settings from this dialog: [©] - Press the 'New' button to create a configuration of the selected type. [©] - Press the 'Duplicate' button to copy the selected configuration. [♥] - Press the 'Delete' button to remove the selected configuration. [♥] - Press the 'Belete' button to configure filtering options. - Edit or view an existing configuration by selecting it. Configure launch perspective settings from the 'Perspectives' preference page.			
Filter matched 7 of 7 items				
3	Debug	Close		

图 4 新建 C/C++远程调试项目

Debug Configurations		Program Selection		X
Create, manage, and run configurations		Choose a program to run:		
Image: Second Secon	Name: step2_serial Main ** Debu C/C++ Application: Debug/step2_serial Project: step2_serialtest Build (if required) Build configurati Enable auto bu © Use workspace	Binaries: Step2_serialtest Qualifier: Same - /step2_serialtest/De Same - /step2_serialtest/Re (?)	2 bug/step2_serialtest ease/step2_serialtest Cancel	Variables Search Project Browse 1 Browse using 'C/C++ Application' ale auto build re Workspace Settings
Filter matched 8 of 8 items Using GDB (DSF) M		Nanual Remote Debugging Launcher	- <u>Select other</u>	Apply Revert
?			Debug Close	

图 5 选择调试的目标程序

Step3-3: 设置 Main 页面

在新的对话框中的 Main 页面, Search Project,选择 Debug 版本的目标程序,点击 OK,如图 5;继续在 main 页面点击最下面的 Select other...,在弹出的对话框中勾选 Use configuration specific settings,在 Launchers 中选择 GDB(DSF) Manual Remote Debugging Launcher,最后选择 OK,如图 6

Debug Configurations	2
Create, manage, and run configurations (2) Remote executable path is not specified.	
Image: Contract of the second seco	Name: step2_serialtest Debug Main Image: Select Preferred Launcher C/C++ Applica This dialog allows you to specify which launcher to use when multiple launchers are available for a configuration and launch mode. Project: Search Project Build (if requesting a flow and and the control of BOSF) Manual Remote Debugging Launcher Browse Build config Standard Remote Create Process Launcher 3 Image: Brow application that was manually started on a remote system under control of GDB debugger integrated using the Image: New Properties Remote Absot Image: Application that was manually started on a remote system under control of GDB debugger integrated using the Browse Browse Skip download to target path. Image: Application that was manually started on a remote system under control of GDB debugger integrated using the Browse Browse
Filter matched 8 of 8 items	Using GDB (DSF) Automatic Remote Debugging Launcher - <u>Select other</u> 1
?	Debug

图 6 选择 Launcher

Step3-4: 设置 Debugger 页面

点击 Debugger 页面, 在 GDB debugger 中通过 Browse 选择 arm-none-linux-gnueabi-gdb.exe 的安装路径,再点击 connection 继续设置,如图 7。

Debug Configurations	×
Create, manage, and run configurations	- A
Image: Second	Name: step2_serialtest Debug Main Main Stop on stadup at main Debugger Options 3 Main Shared Libraries Connection 2 GDB debugger: C\Program Files (x86)\CodeSourcery\Sourcery G++ Lite\bin\arm-none-linux-gnueabi-gdb.exe Browse GDB command file: gdbinit Browse (Warning: Some commands in this file may interfere with the startup operation of the debugger, for example "run".) Non-stop mode (Note: Requires non-stop GDB) Inshe Reverse Debugging at startury (Note: Requires Reverse GDB) Force thread list update on suspend Automatically debug forked processes (Note: Requires Multi Process GDB) Tracepoint mode: Normal •
Filter matched 8 of 8 items	Using GDB (DSF) Manual Remote Debugging Launcher - Select other Apply Revert
?	Debug

图 7 选择 gdb 工具路径

在 Connection 中 Type:选择 TCP; Host name or IP address 和 Port number 中分别**输入 目标板(EM9287)的 IP 地址和 gdbserver 监听端口号**,如图 8,最后点击 Apply。

Main Shared Libraries Connection			
Туре: ТСР 🔻			
Host name or IP address: 192.168.201.93			
Port number:	47476		

图 8 设置与 gdbserver 建立连接的参数

4、进入调试界面及界面切换

点击对话框最下面的 Debug 开始调试,弹出的对话框选择 yes (可以勾选前面的 Remember my decision,后边将不会再弹出此对话框),进入调试界面,至此,可以进行调试工作了,如图 9 和图 10。



图 9 切换调试界面提示框

Debug - step2_se	rialtest/Step2_SerialTest.cpp - Eclipse						x
The tot source network namena series region where the minimum methods and the source series of the source network network and the source of th					_		
					guilt meets		bug
Debug SS E step2.serial B tep2.serial B tep2.serial E step2.serial E s	test Delong (IC/E++ Remote Application) histant datase	be Vasible 22 % Designed. 20 keys Anne * • bh c0 m_045 c0 m_04	Type CSerial Const iet iet iet cher (1500) iet pthread_t		4% (A) 4% (A)	. * * 10 6	
		4					6
() 3992,5613774.6 412 } 412 } 413 } 414 } 414 } 415 } 416 } 416 } 417 } 418	<pre>pp EF [] is find-op baudRate = 9600; trf("port:%d baudrate:%d\n", portno, baudRate); macumeremanymansmented macumeremanymansmented ite0 = macumeremanymansmented ite1 = macumeremanymansmented ite2 = macumeremansmented ite2 = macumeremansmented ite</pre>			Counter 21 Bir Done Marchard Carlos Marchard	s (Serial d D) 1 int	A N X + ₩ * *	
🖸 Console 🛛 Tasi	is 🖹 Problems 🔘 Executables 🟮 Memory 💱				📑 🛃 mg que 💽	4 H 😫 5 + * * *	

图 10 Eclipse 调试界面



图 11 C/C++编辑界面和调试界面切换

如图 11 所示,点击右上角的 C/C++与 Debug 可以在开发编辑界面和调试界面之间切换。

5、后续使用说明

上述设置只需要一次,以后可以直接点击下图所示的小爬虫图标直接进入调试界面,小爬虫右边的倒三角可以选择要调试的项目,如图 12;或者通过 Run >> Debug History 选择 要调试的项目,如图 13。



图 12 运行调试程序工具栏图标

	Run) Window Help		
	8	Resume		🖸 • 🞯 • 🕸 • 🔕 • 💁 (
		Suspend		
9		Terminate		p 🛛
	14	Disconnect		
		Resume Without Signal		
	P.	Step Into		ittr(p_info->port_fd, ICSANO
	P	Step Over		
	_P	Step Return		not set the serial port par
	=>]	Run to Line		
	Ţ	Use Step Filters	Shift+F5	
	Q	Run	Ctrl+F11	
	粅	Debug	F11	
		Run History	•	
		Run As	+	前面建立的调试项目
		Run Configurations		
		Debug History	•	1 step2 serialtest Debug
		Debug As		2 sten2 serialtest
		Debug Configurations		
		bebag configurations		
	Θ	Toggle Breakpoint	Ctrl+Shift+B	
	Θ	Toggle Line Breakpoint		
	Θ	Toggle Method Breakpoint		
	66	Toggle Watchpoint		<pre>reThreadFunc(void* lparam)</pre>
	X	Skip All Breakpoints		
	*	Remove All Breakpoints		(CSerial*)lparam;
		Breakpoint Types	•	10];
	Q	External Tools	+	
		- 303 fd	set fdRead.	_

图 13 菜单栏运行调试程序

二、使用 Eclipse 进行调试

1、设置/取消断点

使用 Eclipse 进行调试,与其他图形界面的集成开发环境一样,直接在代码编辑界面的 左侧双击即可设置/取消断点,如图 14。

```
C Step2_SerialTest.cpp 🔀 💦 Serial.cpp
42
         else
 43
         {
 44
             baudRate = 9600;
 45
         }
 46
 47
         printf( "port:%d baudrate:%d\n", portno, baudRate);
 48
 49
 50
         //打开串口相应地启动了串口数据接收线程
         i1 = m_Serial.OpenPort( portno, baudRate, '8', '1', 'N');
051
 52
         if( i1<0 )
 53
         {
 54
             printf( "serial open fail\n");
 55
             return -1;
 56
         }
 57
 58
         Buf[0] = 0xaa;
 59
         Buf[1] = 0xaa;
         Buf[2] = 0x0D;
 60
 61
         Ruf[3] = 0x34.
```

图 14 设置断点

2、单步调试及运行到光标所在行

可以在断点所在行右键选择 Run to Line (ctrl+r),直接执行到此行,如图 15。



图 15 运行到光标所在行

也可以使用调试工具栏的按键 (F6),一步一步执行;使用 (F8)可以让程序一直运行,直到遇到断点,如图 16。

i 🗙 i 🔜 🔍 i 👁 i 🔈 🕒 💷 🖬 🌼 🔹 🕗 🕶 💁 🕶

图 16 调试工具栏

3、多线程运行

在 Serial.cpp 中 read 函数处设置断点,点击 ,让程序一直运行,此时主线程一 直发送数据,而接收线程一直监听是否有数据到来,等待触发断点,如图 17。

332 if (ret > 0)	
333 {	
334 //判断是否读事件	
<pre>335 if (FD_ISSET(pSer->m_fd,&fdRead))</pre>	
336 {	
337 //data available, so get it!	
<pre>pSer->m_DatLen = read(pSer->m_fd, pSer->DatBuf, 100</pre>);
339 // 对接收的数据进行处理,这里为简单的数据回发	
340 if(pSer->m_DatLen > 0)	
341 {	
<pre>342 printf("DatLen:%d\n",pSer->m_DatLen);</pre>	
343	
344 for (i=0; i <pser->m_DatLen; i++)</pser->	
345 {	
346 //buff[pSer->DatLen] = pSer->DatBuf[i];	
347 //pSer->DatLen++;	
<pre>348 printf("%02x ", pSer->DatBuf[i]);</pre>	
349 }	
3510 /*if(huff[nSer->DatLen-1] == 0x30)	

图 17 等待断点触发

在电脑端向 step2_serialtest 使用的串口发送数据,将会触发断点,Debug 窗口页面也可以 看到目前执行到了第二个线程的 ReceiveThreadFunc(),如图 18。



图 18 Debug 窗口程序运行状态

4、Variables 窗口

通过 Variables 窗口可以观察程序运行中的所有变量值,如图 19。

🛛 🕬= Variables 🔀 💁 Breakpoints 👯 Reg	isters 🛋 Modules	🖆 🕫 🖂 🖉 💥 🎇 📑 🖻
Name	Туре	Value
➡ Iparam	void *	0x12538
⊿ 🔿 pSer	CSerial *	0x12538
(×)= m_fd	int	3
(×)= m_DatLen	int	4
(×)⊧ DatLen	int	0
🔺 🥭 DatBuf	char [1500]	0x1254c
Þ 🔚 [099]	char [100]	0x1254c
Þ 🧱 [100199]	char [100]	0x125b0
N III 1000 2001	char [100]	0v1261/
	III	

- 图 19 变量观察窗口
- 5、停止及重启

🏇 Debu	g 🖾						
4 🖸 🔹	<terminated>step</terminated>	2_serialtest Debug [C/C+	+ Rei	mote Application]			
<pre><terminated, 0="" exit="" value:="">gdb</terminated,></pre>			D	Copy Stack	Ctrl+(
				Find	Ctri+i		
			-9	Drop To Frame			
			3.	Step Into			
			3	Step Over			
			_62	Step Return			
			i⇒	Instruction Stepping Mode			
			P	Use Step Filters			
				Resume Without Signal			
				Resume			
				Suspend			
				Terminate			
			₽,	Terminate and Relaunch		- h	
C Step2	_SerialTest.cpp	🔓 Serial.cpp 🔀 💽 (gd	14	Disconnect		re	ea
280	ret	urn -1;	\odot^{\diamond}	Debug New Executable			
281	}		84	Connect			
282	notunn	ctatuc.	34	Remove All Terminated			
285	}	status,	Q	Relaunch			
285	,		C	Edit step2_serialtest Debug			
286 CSerial::CSerial()			E./	Edit Source Lookup			
287	287 { 288 m_DatLen = 0; 289 DatLen = 0;		é.	Terminate and Remove			
288				Terminate/Disconnect All			
289				reminate/Disconnect All			
290	m_ExitT	hreadFlag = 0;		Properties			
291	}		_				

图 20 停止和重启调试程序

点击 ● ♪ , 停止 gdb 调试,目标板将退出 gdbserver。再次调试,需要在目标板 上重新运行 gdbserver,然后 Eclipse 中可以选中 Debug 页面中的调试项目,点击鼠标右 键,选择 Relaunch 既可以重新启动调试,如图 20。