

# Colibri Vybrid

Linux 开发入门



## Revision History

Date	Doc. Rev.	Colibri VFxx Version	Changes
2015-8-17	Rev. 0.1	Colibri VF50 V1.0A, V1.1A,V1.2A Colibri VF61 V1.0B, V1.1A, V1.1A, V1.2A	初始版本
2015-12-14	Rev. 0.2	Colibri VF50 V1.0A, V1.1A,V1.2A Colibri VF61 V1.0B, V1.1A, V1.1A, V1.2A	文档模板更新
2016-5-18	Rev. 0.3	Colibri VF50 V1.0A, V1.1A,V1.2A Colibri VF61 V1.0B, V1.1A, V1.1A, V1.2A	添加 Toradex 社区
2016-5-31	Rev0.4	Colibri VF50 V1.0A, V1.1A,V1.2A Colibri VF61 V1.0B, V1.1A, V1.1A, V1.2A	MQX 开发
2016-6-2	Rev 0.5	Colibri VF50 V1.0A, V1.1A,V1.2A Colibri VF61 V1.0B, V1.1A, V1.1A, V1.2A	Device tree 配置
2016-6-27	Rev 0.6	Colibri VF50 V1.0A, V1.1A,V1.2A Colibri VF61 V1.0B, V1.1A, V1.1A, V1.2A	公司地址信息变更
2017-2-9	Rev 0.7	Colibri VF50 V1.0A, V1.1A,V1.2A Colibri VF61 V1.0B, V1.1A, V1.1A, V1.2A	在 Eclipse 中使用 SDK 中的 toolchain 添加 FTP 下载地址
2018-08-08	Rev 0.8	Colibri VFXX	更新 Linux BSP v2.8b3, git checkout

## 目录

---

<b>1. 开发设备以及材料准备</b> .....	<b>5</b>
1.1 硬件准备 .....	5
1.1.1 电脑主机 .....	5
1.1.2 开发平台 .....	5
1.1.3 基本配件 .....	5
1.2 软件准备 .....	5
<b>2. 开发环境硬件配置</b> .....	<b>6</b>
2.1 硬件连接框图 .....	6
2.2 电源连接 .....	6
2.2.1 Colibri 开发板 .....	6
2.2.2 Iris 载板 .....	7
2.3 显示输出连接 .....	7
2.3.1 Colibri 开发板 .....	8
2.3.2 Iris 开发板 .....	8
2.4 调试串口连接 .....	8
2.4.1 Colibri 开发板 .....	8
2.4.2 Iris 载板 .....	11
2.4.3 串口波特率设置 .....	11
2.5 USB 键盘/鼠标连接 .....	11
2.6 网络连接 .....	12
2.7 上电启动计算机模块 .....	12
2.7.1 Colibri 开发板 .....	12
2.7.2 Iris 载板 .....	13
<b>3. 开发环境软件配置</b> .....	<b>14</b>
3.1 开发软件安装 .....	14
3.2 串口软件 Minicom 配置 .....	14
3.3 Colibri Vybrid Linux SDK .....	16
3.4 Linux BSP 下载 .....	17
<b>4. Eclipse 应用程序开发</b> .....	<b>18</b>
4.1 创建 C 工程项目 .....	18

4.2	添加工程文件.....	18
4.3	工程属性配置.....	19
4.4	Eclipse's Remote System Explorer 配置.....	22
<b>5.</b>	<b>Eclipse 调试应用程序.....</b>	<b>28</b>
5.1	Eclipse 设置.....	28
<b>6.</b>	<b>U-Boot 和 Linux 源代码下载.....</b>	<b>31</b>
6.1	U-Boot.....	31
6.2	Linux.....	31
<b>7.</b>	<b>MQX 开发.....</b>	<b>32</b>
7.1	软件下载.....	32
7.2	MQX 配置.....	32
7.3	编译 MQX.....	34
7.4	外设资源分配.....	34
7.5	运行 MQX.....	35
<b>8.</b>	<b>BSP 更新.....</b>	<b>36</b>
<b>9.</b>	<b>其他开发参考资料.....</b>	<b>37</b>
9.1	OpenEmbedded.....	37
9.2	开发者中心.....	37
9.3	Toradex 社区.....	37
<b>10.</b>	<b>Toradex 公司简介及本地支持联系方式.....</b>	<b>38</b>
10.1	Toradex 公司简介.....	38
10.2	本地支持联系方式.....	38

## 1. 开发设备以及材料准备

---

### 1.1 硬件准备

#### 1.1.1 电脑主机

- ✓ 64bit Linux PC

#### 1.1.2 开发平台

- ✓ Colibri VF50 或者 Colibri VF61 计算机模块
- ✓ Colibri Evaluation Board V3.1A/V3.2 或者 Iris V1.1 底板

#### 1.1.3 基本配件

开发所需的基本配件，按需自行配置，不包含在标准载板购买包装内。

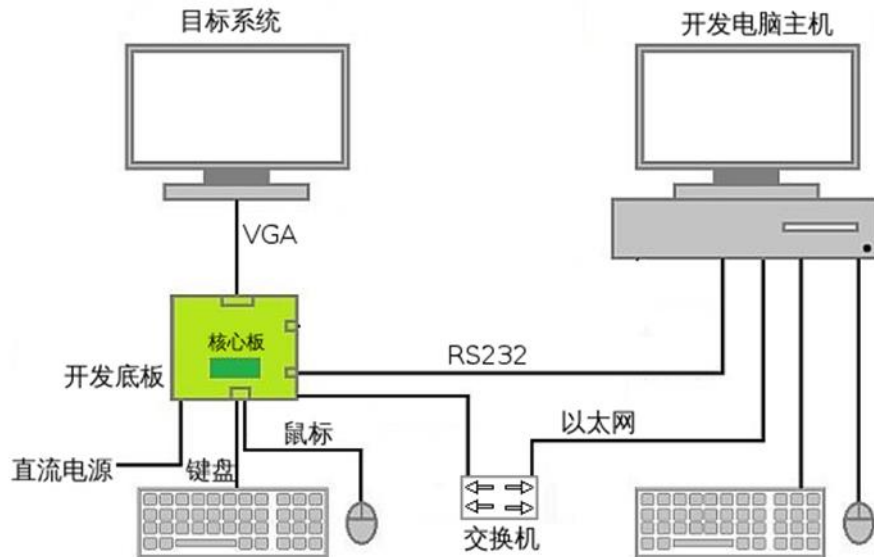
- ✓ 电源适配器/直流电源 (7 ~ 27V)
- ✓ VGA 连接线
- ✓ VGA 显示器
- ✓ 串口线
- ✓ USB 数据线

### 1.2 软件准备

- ✓ Ubuntu 14.04
- ✓ Eclipse CDT
- ✓ Minicom
- ✓ 交叉编译工具
- ✓ Colibri VF Linux BSP

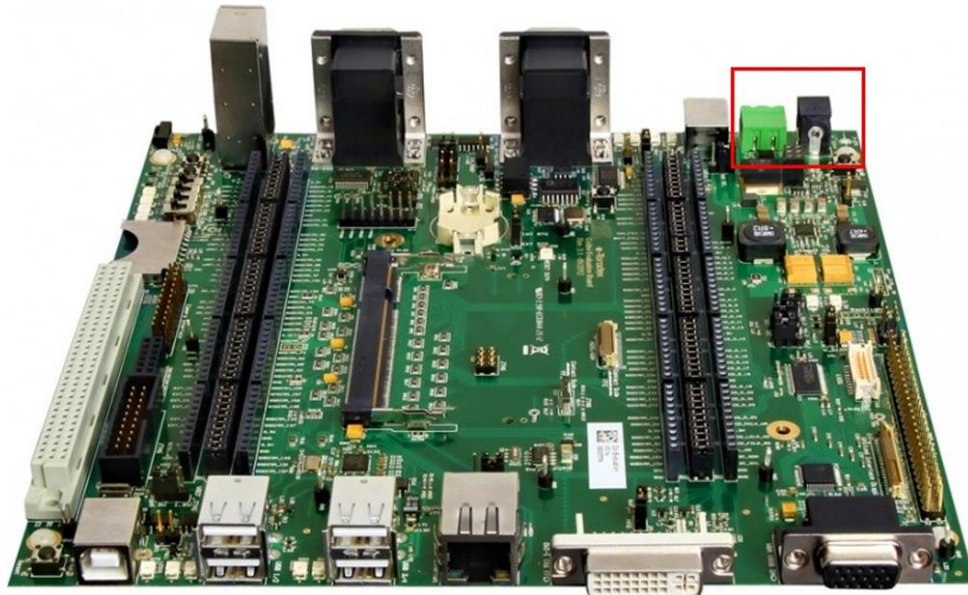
## 2. 开发环境硬件配置

### 2.1 硬件连接框图



### 2.2 电源连接

#### 2.2.1 Colibri 开发板



- ✓ Colibri Evaluation Board 底板上的 X33 和 X35 接口，任意一路连接 DC 7~27V 电源即可，请注意 VCC 和 GND 的引脚位置。

## 2.2.2 Iris 载板



- ✓ Colibri Evaluation Board 底板上的 X17 接口，任意一路连接 DC 6~27V 电源即可，请注意 VCC 和 GND 的引脚位置

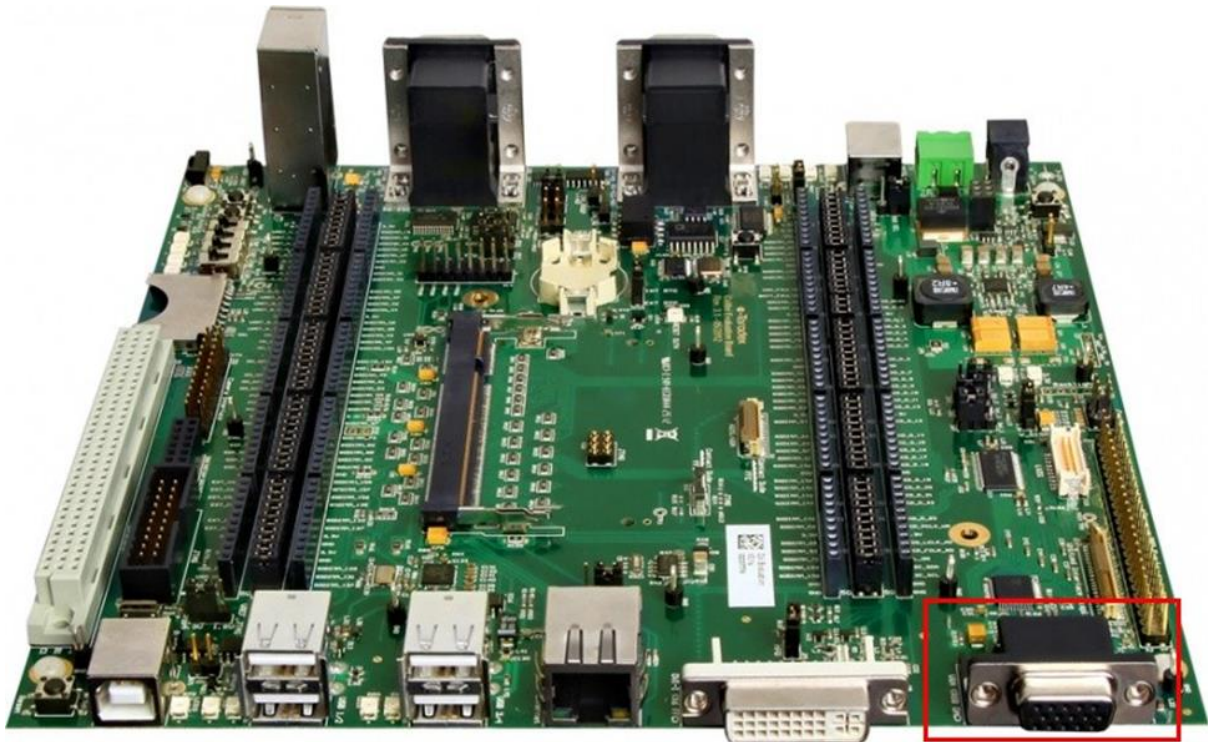
## 2.3 显示输出连接

韬睿提供的底板，大多数均支持 DVI/HDMI\*、LVDS、并口 RGB、VGA\*\*等常用显示器接口。

\* 由于 Colibri VFXX 模块均没有 DVI 接口，所以载板上的 DVI 接口没有显示输出。

\*\* Colibri Evaluation Board 上 LVDS 和 VGA 由载板上相应的 RGB 转 LVDS 以及 RGB 转 VGA 芯片实现，Colibri VFXX 本身不支持 LVDS 和 VGA 直接输出。

### 2.3.1 Colibri 开发板



- ✓ Colibri Evaluation Board 载板上 X24 为 VGA 输出，将其连接值 VGA 显示器，默认分辨率为 640x480。

### 2.3.2 Iris 开发板

- ✓ Iris 载板没有支持桌面显示器 (VGA/DVI/HDMI) 的输出接口，载板可以支持 LVDS 和并口 RGB 显示屏，您需要对应的屏线，连接载板和显示屏。详细的线序定义请参考 [Iris 技术手册](#)。

## 2.4 调试串口连接

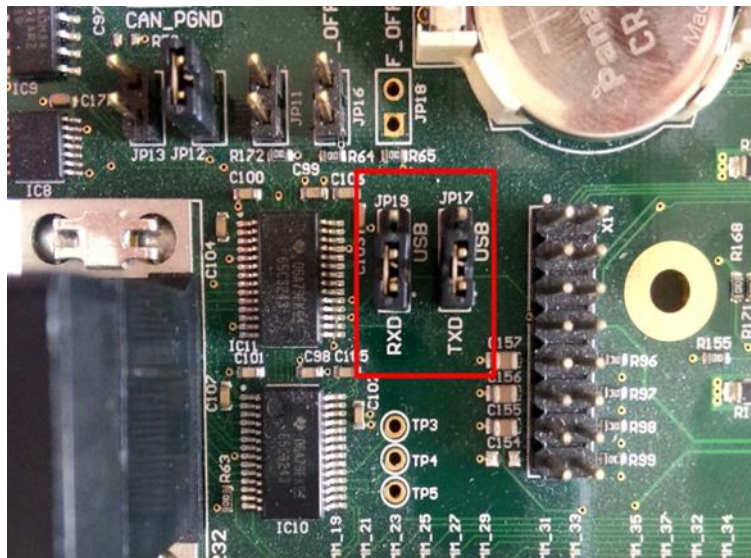
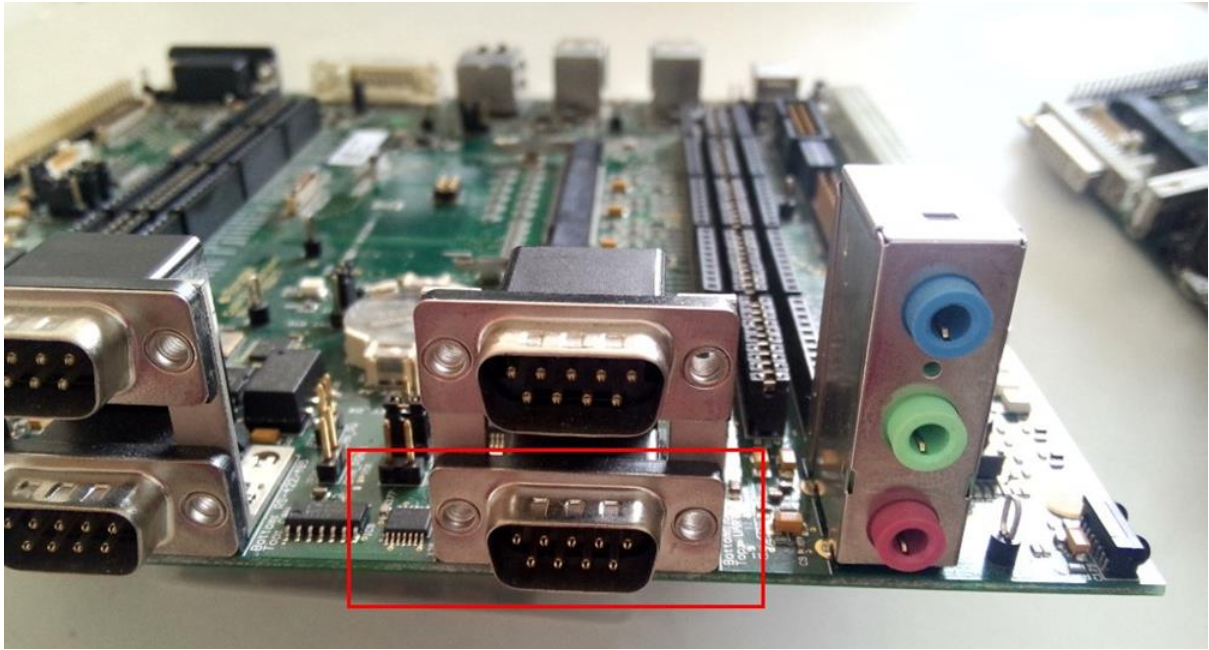
Colibri VFXX 默认情况下将 UART\_A 作为调试串口，该串口不仅可以用于应用开发时候，调试应用程序，还可以获取以及配置 Linux 和 U-Boot 系统。如果串口数量能够满足应用的前提下，一般建议保留调试串口的功能。

为了方便用户使用调试串口，Colibri Evaluation Board 提供两种方式连接串口。第一种，直接使用 RS232 串口连接。这个方法适用于台式机用户，一般台式机电脑均提供 RS232 接口。第二种，USB 串口连接。Colibri Evaluation Board 底板上的 FTDI FT232RL 芯片，将 UART 串口转为 USB 设备。这种适用于没有 RS232 接口的笔记本用户，当然，也适用于台式机。下面将介绍详细的连接方法。

### 2.4.1 Colibri 开发板

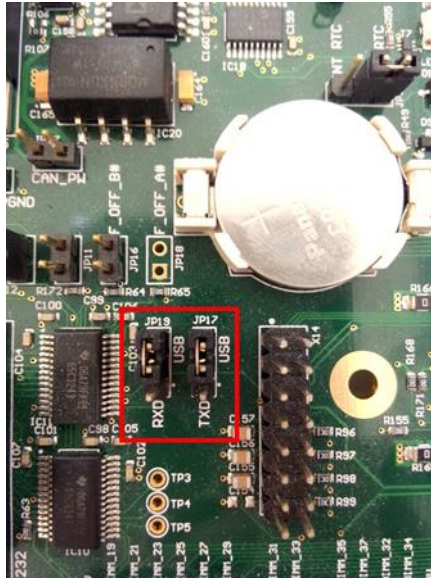
RS232 串口接口



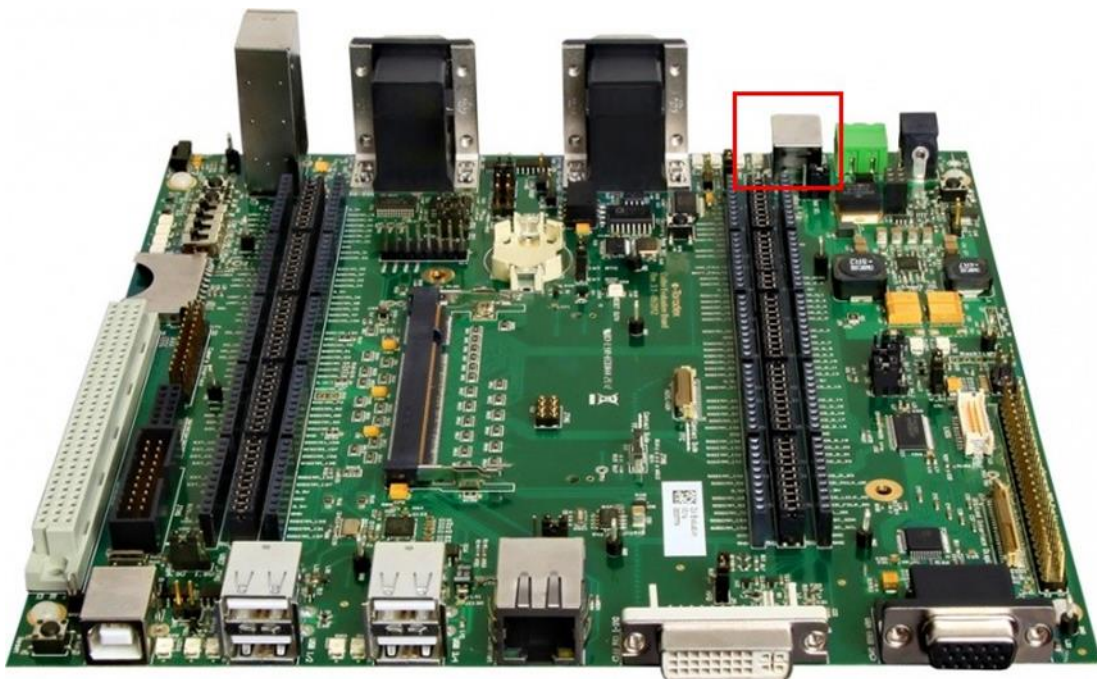


- ✓ 在 Colibri Evaluation Board 底板上 X25 的下面的串口为默认的调试串口，同时将 JP17 和 JP19 上的短路帽分别移至 RXD 和 TXD 一侧。

USB 转串口接口



- ✓ 在 Colibri Evaluation Board 底板上 X27 为 USB 调试串口，同时将 JP17 和 JP19 上的短路帽分别移至 USB 一侧。



- ✓ 使用 USB 数据线，将载板上的 X27 连接至电脑主机。

Ubuntu 中已经集成 FTDI FT232RL 驱动，使用 lsusb 命令即可发现该设备

```
ban@LinuxDev:~$ lsusb
Bus 003 Device 005: ID 0403:6001 Future Technology Devices International, Ltd FT232 USB-Serial (UART) IC
```

Windows 系统中需要先安装相应的驱动，才能访问串口。FTDI FT232RL 可以从 FTDI 网站下载 <http://www.ftdichip.com/FTDrivers.htm>

## 2.4.2 Iris 载板



- ✓ 在 Iris 上 X13 为默认的调试串口。X13 为 IDC-10 接口，具体引脚定义请参考 [Iris 技术手册](#)。X13 为 RS232 串口，用户可以制作对应的转接线连接至 DB9 接口，转接线线序如下：

DB-9	IDC-10
Pin 1	Pin 1
Pin 2	Pin 3
Pin 3	Pin 5
Pin 4	Pin 7
Pin 5	Pin 9
Pin 6	Pin 2
Pin 7	Pin 4
Pin 8	Pin 6
Pin 9	Pin 8

## 2.4.3 串口波特率设置

默认串口波特率为 115200,8n1。

## 2.5 USB 键盘/鼠标连接

韜睿提供的 VFXX Linux BSP 中已经包含图形桌面系统 LXDM，用户只要连接 USB 鼠标和键盘，就可以方便地使用开发板上的 Linux 系统。除了 Iris 底板，其他底板均有多个 USB 接口，能够连接三个以上的 USB 设备，在 Iris 底板上则需要使用 USB Hub 进行 USB 接口扩展。VFXX 上的 Linux 无需安装驱动，就能够支持 USB Hub。

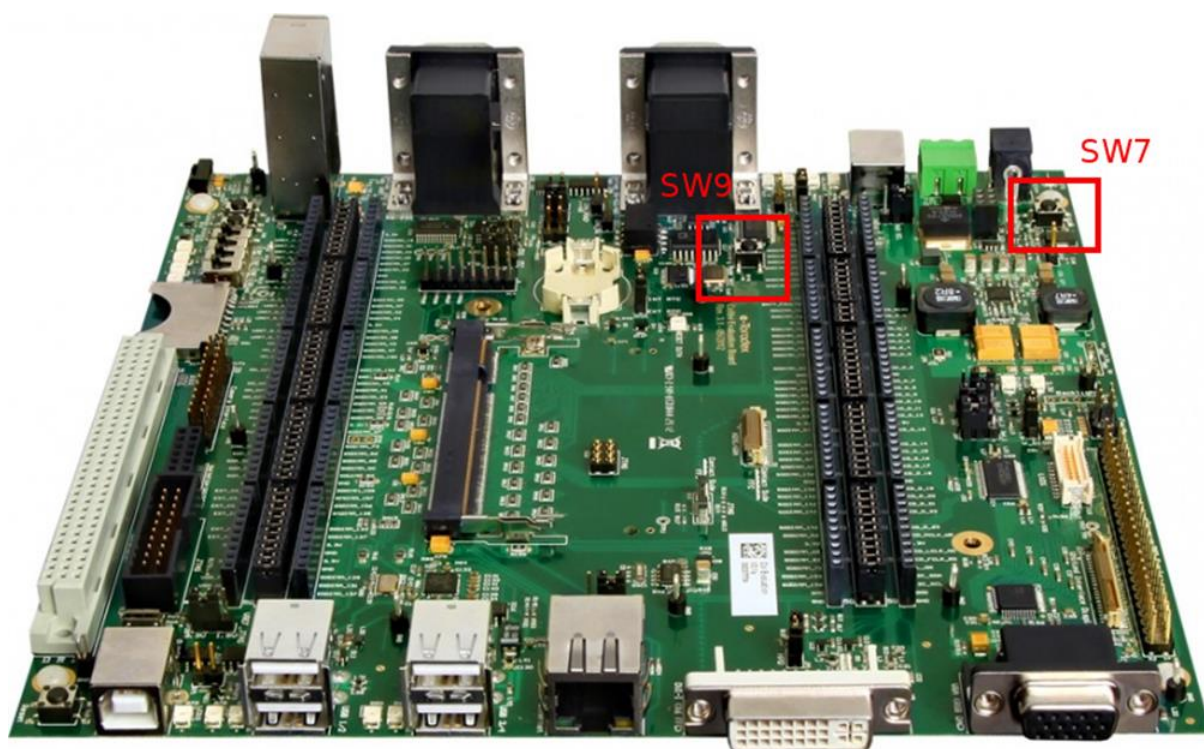
## 2.6 网络连接

在连接以太网的情况，用户可以十分方便地以 SSH 连接远程登录到 VFXX 上的 Linux 系统。同时用户也能用 GDB 实现在线应用程序调试。VFXX 上的 Linux 默认采用 DHCP 方式，自动获取 IP 地址。用户可以通过调试串口或者图形桌面系统查看，VFXX 获取到的 IP 地址。

## 2.7 上电启动计算机模块

除了 Iris 底板，其他底板均有独立的上电和复位开关，因此，需要触发开关，核心模块才能启动。Iris 底板一旦连接电源，其上面的计算机模块就会立即启动。

### 2.7.1 Colibri 开发板



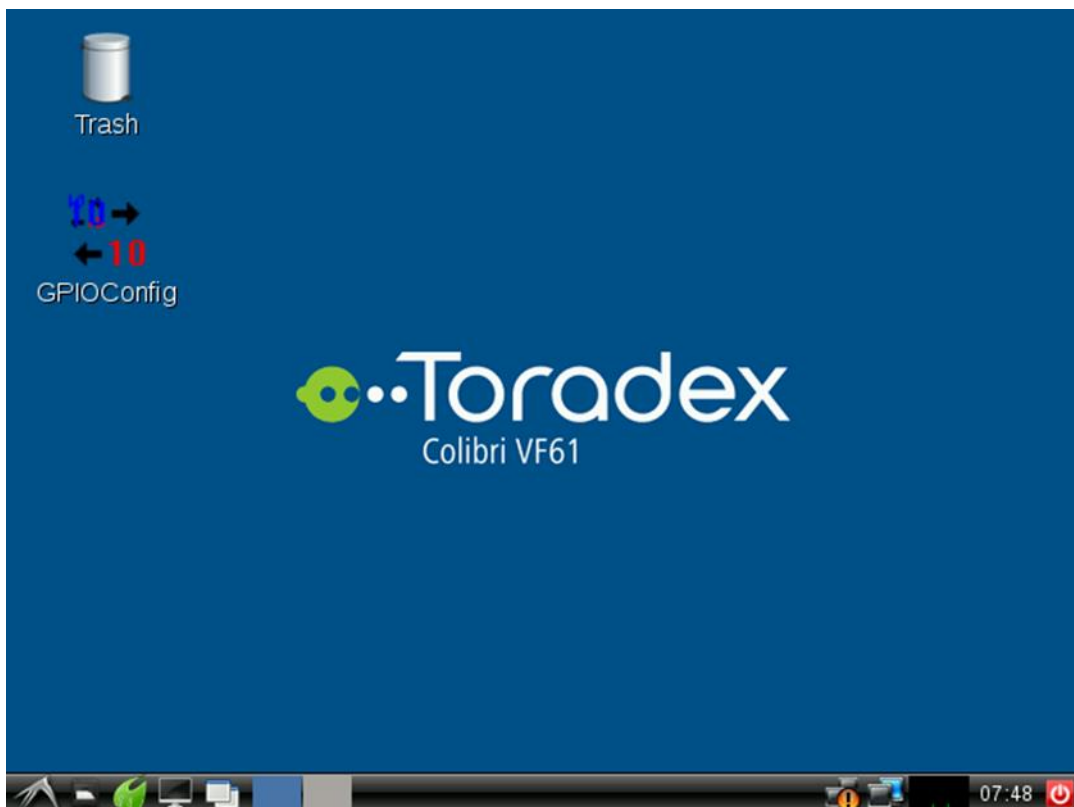
- ✓ SW7: 上电开关
- ✓ SW9: 复位

## 2.7.2 Iris 载板



- ✓ SW1: 复位

Colibri VFXX 模块正确安装到地板上，并连接相应的电源和显示连线，上电启动后，就能够在显示器上看到 Linux 桌面系统。



## 3. 开发环境软件配置

### 3.1 开发软件安装

✓ Ubuntu 14.04 软件安装:

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install build-essential meld git gawk eclipse-cdt eclipse-cdt-launch-remote eclipse-rse minicom
sudo apt-get install gtk+-2.0-dev
```

如果使用的是 64 位系统, 请安装以下软件包

```
sudo apt-get install lib32z1 lib32ncurses5 lib32bz2-1.0
```

### 3.2 串口软件 Minicom 配置

✓ Colibri VFXX 模块串口默认使用 115200, 8N1, no hardware flow control 格式协议。按照下面设置打开 minicom。

```
sudo minicom -s
```

```
+-----[configuration]-----+
```

```
| Filenames and paths |
| File transfer protocols |
| Serial port setup |
| Modem and dialing |
| Screen and keyboard |
| Save setup as dfl |
| Save setup as.. |
| Exit |
| Exit from Minicom |
+-----+
```

```
+-----+
| A - Serial Device : /dev/ttyUSB0 |
| B - Lockfile Location : /var/lock |
| C - Callin Program : |
```

```

| D - Callout Program      :                |
| E - Bps/Par/Bits       : 115200 8N1      |
| F - Hardware Flow Control : No          |
| G - Software Flow Control : No          |
|                          |               |
| Change which setting?   |               |
+-----+
+-----[Modem and dialing parameter setup]-----+
|                          |               |
| A - Init string .....   |               |
| B - Reset string .....  |               |
| C - Dialing prefix #1.... ATDT                |
| D - Dialing suffix #1.... ^M                  |
| E - Dialing prefix #2.... ATDP                |
| F - Dialing suffix #2.... ^M                  |
| G - Dialing prefix #3.... ATX1DT              |
| H - Dialing suffix #3.... ;X4D^M             |
| I - Connect string ..... CONNECT             |
| J - No connect strings .. NO CARRIER   BUSY |
|          NO DIALTONE   VOICE              |
| K - Hang-up string .....                  |
| L - Dial cancel string .. ^M              |
|                          |               |
| M - Dial time ..... 45   Q - Auto bps detect ..... No   |
| N - Delay before redial . 2   R - Modem has DCD line .. No   |
| O - Number of tries ..... 10   S - Status line shows ... DTE speed |
| P - DTR drop time (0=no). 1   T - Multi-line untag .... No   |
|                          |               |
| Change which setting?   (Return or Esc to exit)           |
+-----+

```

上电启动后，可以在 minicom 中看到 Colibri VFXX 模块输出的启动信。登录系统的默认用户名为 root，密码为空。

```
U-Boot 2015.04 (Jun 04 2015 - 15:26:53)

CPU: Freescale Vybrid VF610 at 500 MHz
Reset cause: EXTERNAL RESET
DRAM: 256 MiB
NAND: 512 MiB
MMC: FSL_SDHC: 0
In: serial
Out: serial
Err: serial
Model: Toradex Colibri VF61 256MB IT V1.1B, Serial# 04834907
Net: FEC
.....

The Angstrom Distribution colibri-vf ttyLP0

Angstrom v2014.12 - Kernel

Colibri_VF_LinuxImageV2.4_20150604

colibri-vf login: root
```

### 3.3 Colibri Vybrid Linux SDK

从 FTP 服务器以下目录中下载 SDK

CustomerFTP/Linux/OpenEmbedded/V2.6/SDK/colibri\_vybrid

ftp 地址: <ftp://ftp.toradex.cn>

将交 SDK 安装到用户指定的目录中 /home/ban/Toradex/oe-core-tegra/LinuxSDK/v2.6/ colibri\_vybrid

```
sudo chmod a+x ./angstrom-glibc-x86_64-armv7at2hf-vfp-neon-v2015.12-toolchain.sh
./angstrom-glibc-x86_64-armv7at2hf-vfp-neon-v2015.12-toolchain.sh
```



### 3.4 Linux BSP 下载

你可以从下面链接下载韜睿最新发布的 BSP 文件，以及相应的发行说明。建议用户关注韜睿主页，我们会及时发布 BSP 更新信息，新版本 BSP 一般会解决之前发布版本中存在的一些问题，或者提供某些功能的性能改善。

<http://developer.toradex.com/files/toradex-dev/uploads/media/Colibri/Linux/Images/>

例如，下载 Colibri\_VF\_LinuxImageV2.6Beta2\_20160630.tar.bz2，需要以 root 权限解压。

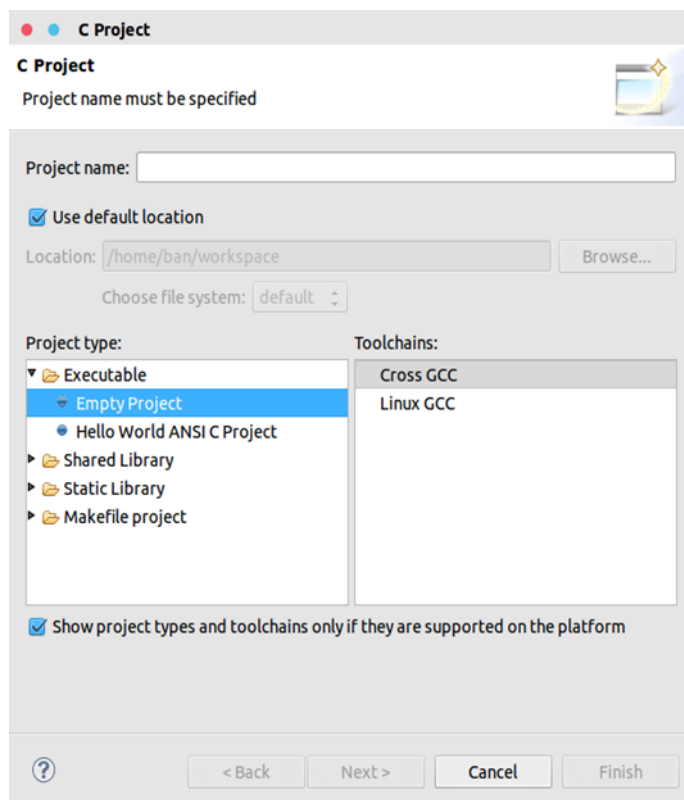
```
|-----|  
| sudo tar vxf Colibri_VF_LinuxImageV2.6Beta2_20160630.tar.bz2 |  
|-----|
```

## 4. Eclipse 应用程序开发

在 Linux 系统中进行应用程序开发，可以有多种 IDE 供用户选择，其中 Eclipse 是较为常用的一种。Eclipse 能够自动生成 Makefiles 文件，这对于初学者来说，是一个极大的便利，可以免去学习 Makefiles 文件编写规则，方便用户快速上手 Linux 应用程序开发。下面将介绍如何在 Eclipse 中进行嵌入式应用程序开发。

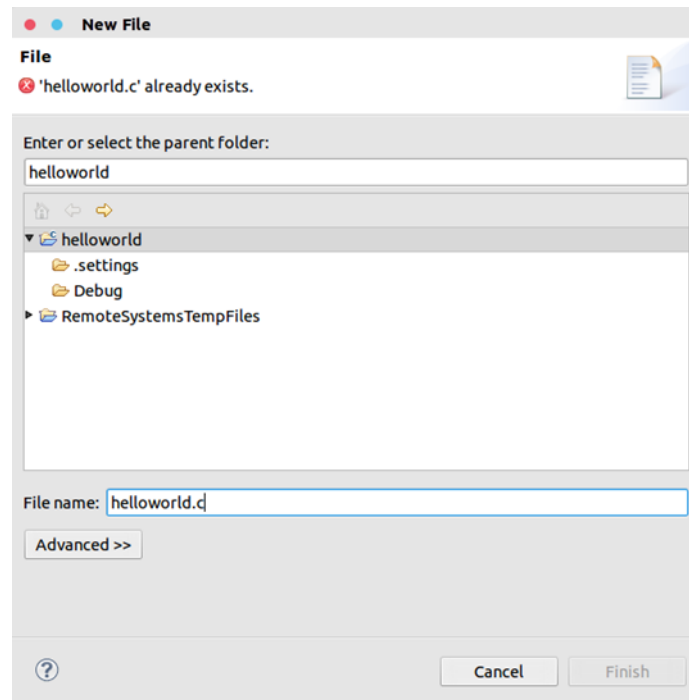
### 4.1 创建 C 工程项目

- ✓ 菜单选项 --> File --> New --> C Project
- ✓ Project type: Empty Project
- ✓ Toolchains: Cross GCC

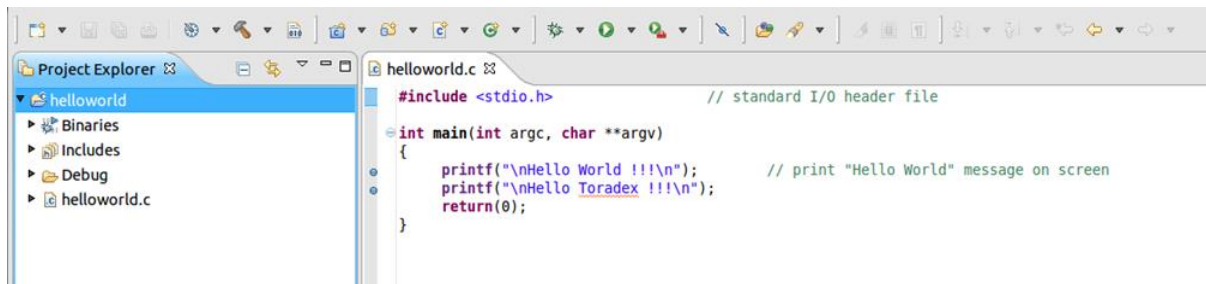


### 4.2 添加工程文件

- ✓ 在 Project Explorer 中的工程文件夹上右击，New --> File
- ✓ file name : hello\_word.c



在 helloworld.c 文件中输入 C 程序



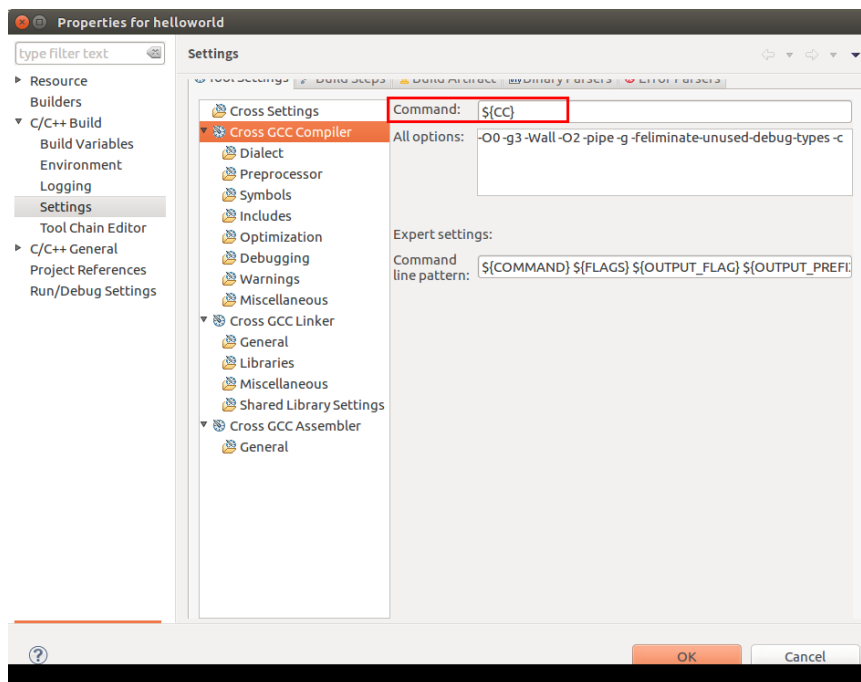
### 4.3 工程属性配置

默认的 Eclipse 中的编译器是针对 x86 平台，所以编译出来的程序无法运行在 ARM 嵌入式平台。因此，需要重新配置项目的属性。这里我们将使用 SDK 中的交叉编译工具。

- ✓ 在 Project Explorer 中的工程文件夹上右击，选择 Properties.
- ✓ 依次选择，C/C++ Build --> Settings

Cross GCC Compiler

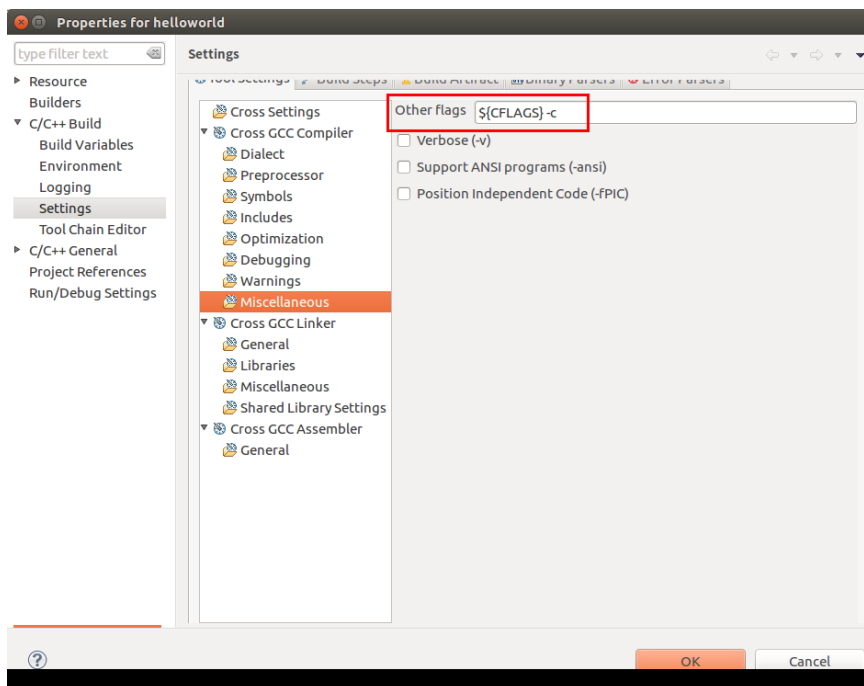
Command : \${CC}



该路径以实际电脑上安装为准，需要做相应的调整，请参考“交叉编译工具安装”章节。

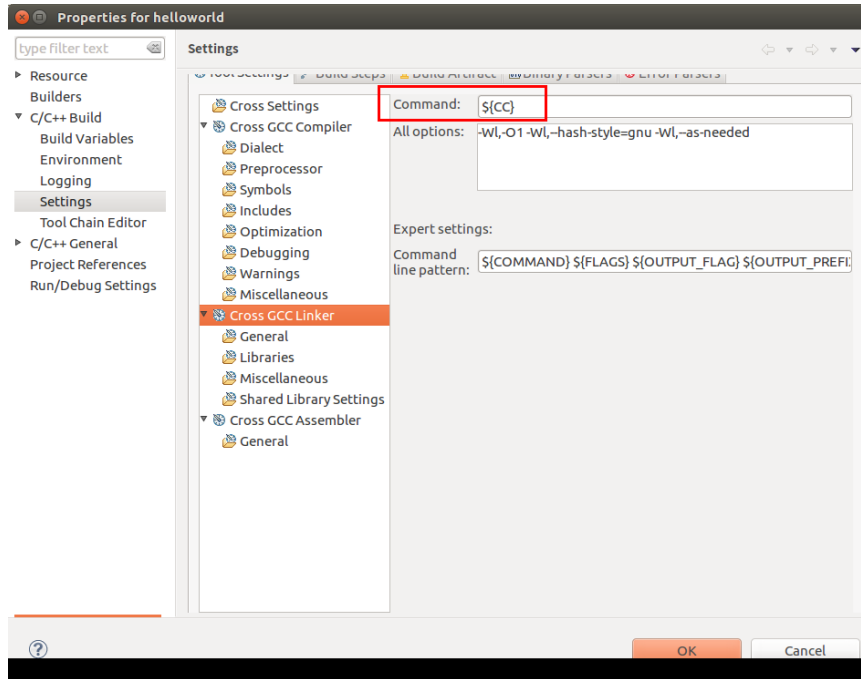
Miscellaneous

Other flags: `${CFLAGS} -c`



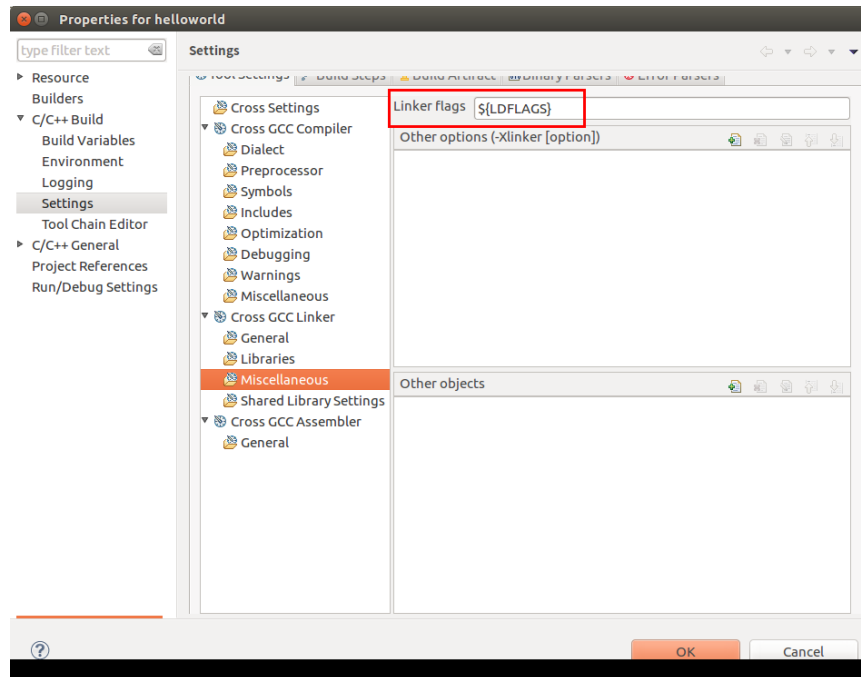
Cross GCC Linker

Command: `${CC}`



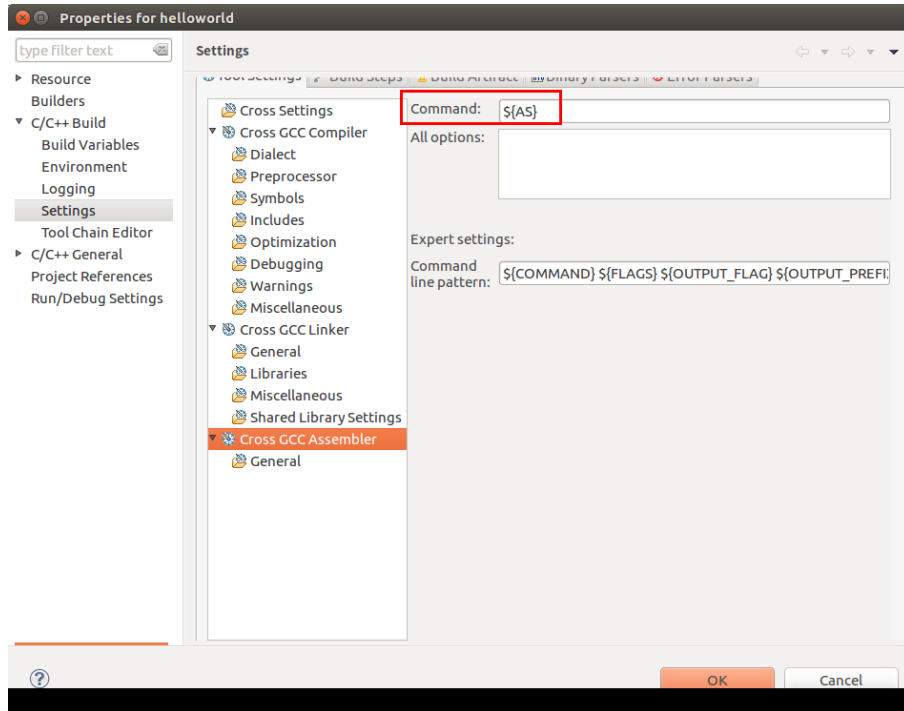
Miscellaneous

Linker flags: `${LDLFLAGS}`



Cross GCC Assembler

Command : `${AS}`

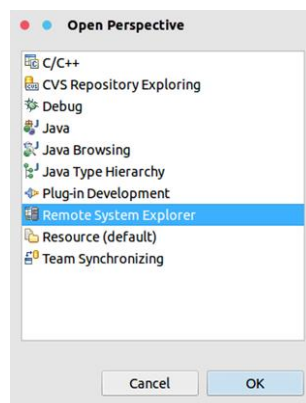


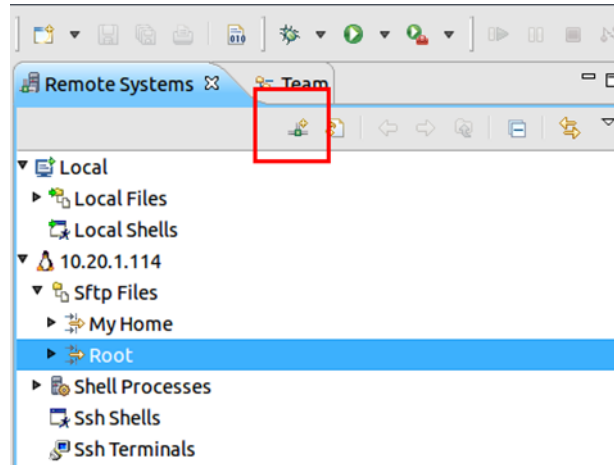
配置完毕后，就可以点击 Project --> Build All 编译项目。

#### 4.4 Eclipse's Remote System Explorer 配置

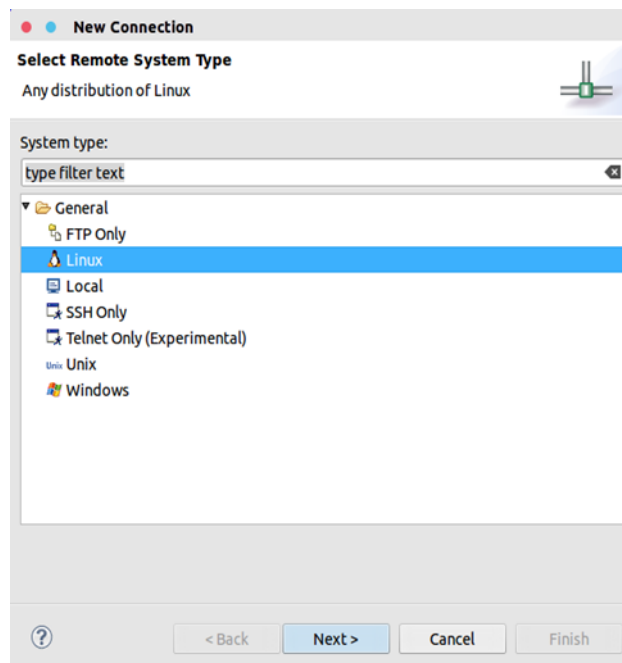
使用 Remote System Explorer 后，可以直接在 Eclipse 中访问目标板上的文件系统，方便将文件从电脑上复制到目标板，部署应用程序至开发板。

点击 菜单 > Windows --> Open Perspective --> Other..., 选择 Remote System Exploere





- ✓ 在 Remote System --> Define a connection to remote system 点击



- ✓ Host name: 目标板的 IP 地址
- ✓ 勾选 Verify host name

**New Connection**

**Remote Linux System Connection**

✖ Please enter a valid connection name.

Parent profile: LinuxDev

Host name: 10.20.1.114

Connection name:

Description:

Verify host name

[Configure proxy settings](#)

? < Back Next > Cancel Finish

**New Connection**

**Files**

Define subsystem information

Configuration	Properties				
<input type="checkbox"/> dstore.files <input type="checkbox"/> ftp.files <input checked="" type="checkbox"/> ssh.files	<table border="1"> <thead> <tr> <th>Property</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> </tr> </tbody> </table>	Property	Value		
Property	Value				

Available Services

- Ssh / Sftp File Service
- SSH Connector Service
  - SSH Settings

Description

Work with files on remote systems using the Secure Shell (ssh) protocol.

? < Back Next > Cancel Finish



**New Connection**

**Processes**  
Define subsystem information

Configuration	Properties				
<input type="checkbox"/> dstore.processes <input checked="" type="checkbox"/> processes.shell.linux	<table border="1"> <thead> <tr> <th>Property</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> </tr> </tbody> </table>	Property	Value		
Property	Value				

Available Services

- Shell Process Service

Description

This configuration allows you to work with processes on remote linux systems using any contributed Shell subsystem.

**New Connection**

**Shells**  
Define subsystem information

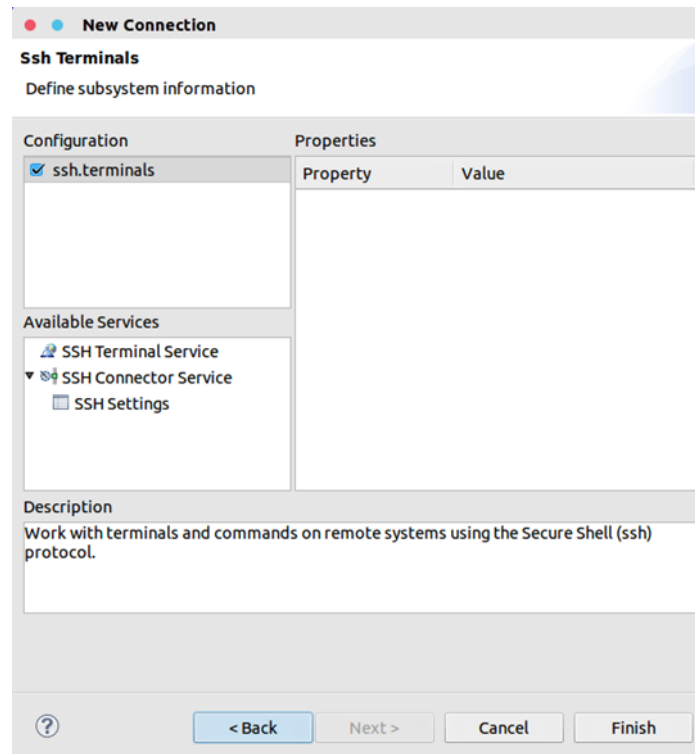
Configuration	Properties				
<input type="checkbox"/> dstore.shells <input checked="" type="checkbox"/> ssh.shells	<table border="1"> <thead> <tr> <th>Property</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> </tr> </tbody> </table>	Property	Value		
Property	Value				

Available Services

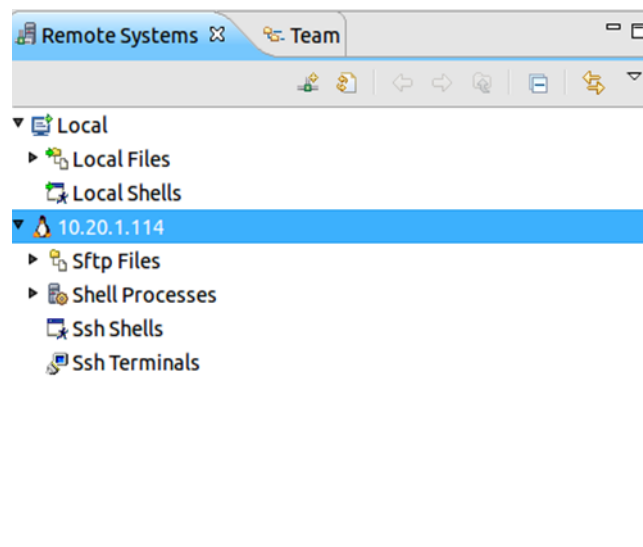
- Generic shell service
- SSH Connector Service
  - SSH Settings

Description

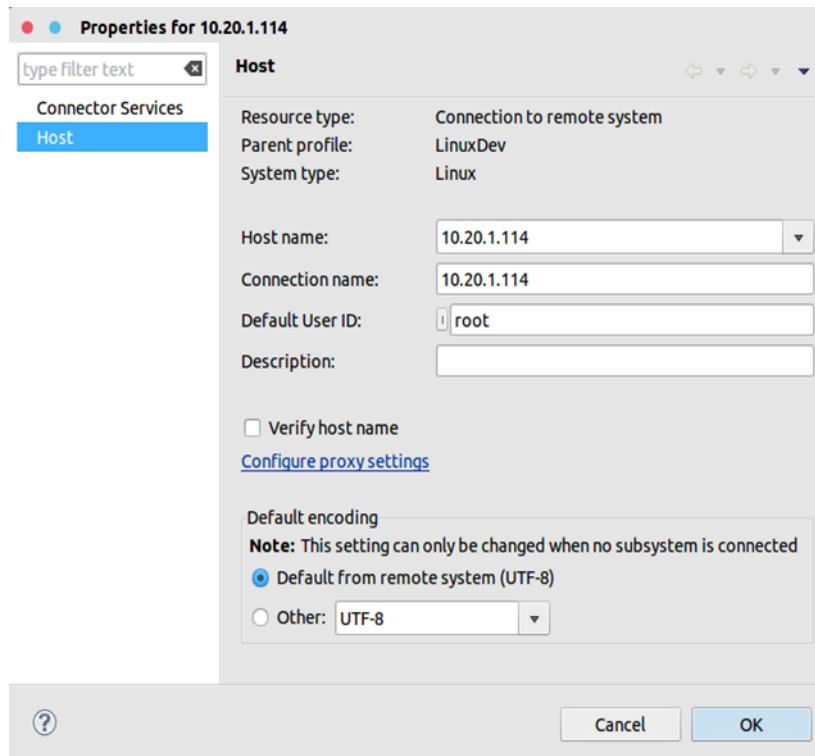
Work with shells and commands on remote systems using the Secure Shell (ssh) protocol.



- ✓ 配置完成后，在 Remote Systems 中可以看到目标板的信息



默认配置下，Eclipse 会使用电脑的用户名来登录目标板系统。这里需要更改登录默认用户名以及密码。



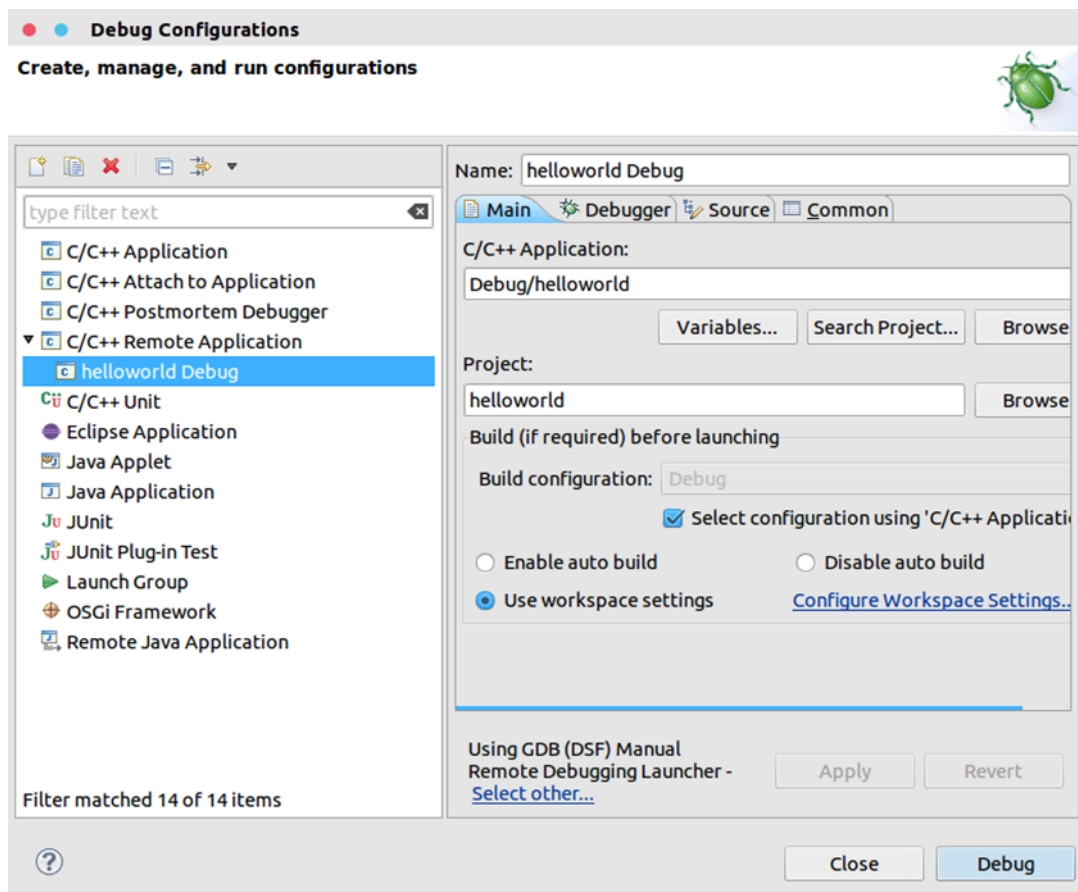
- ✓ Remote Systems 中右击目标板的 IP 地址，点击 Connect 即可连接。
- ✓ 在 Eclipse 中将编译好的程序复制到 Remote System --> Sftp files --> My home。使用 SSH 终端连接目标板，运行下面命令：

```
┌-----┐  
│ chmod +x hello_world │  
└-----┘
```

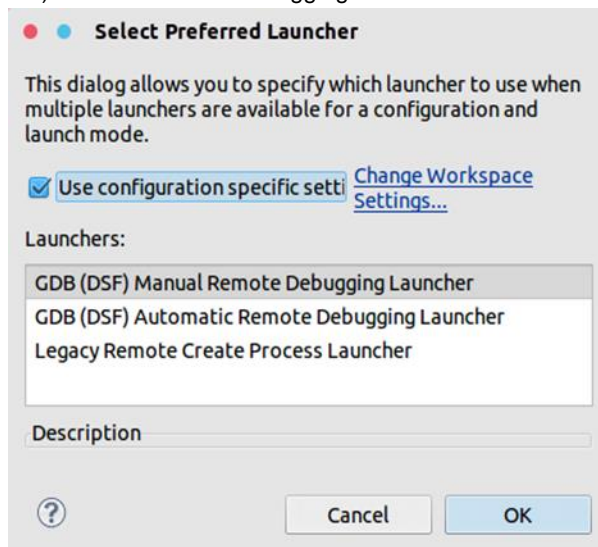
## 5. Eclipse 调试应用程序

### 5.1 Eclipse 设置

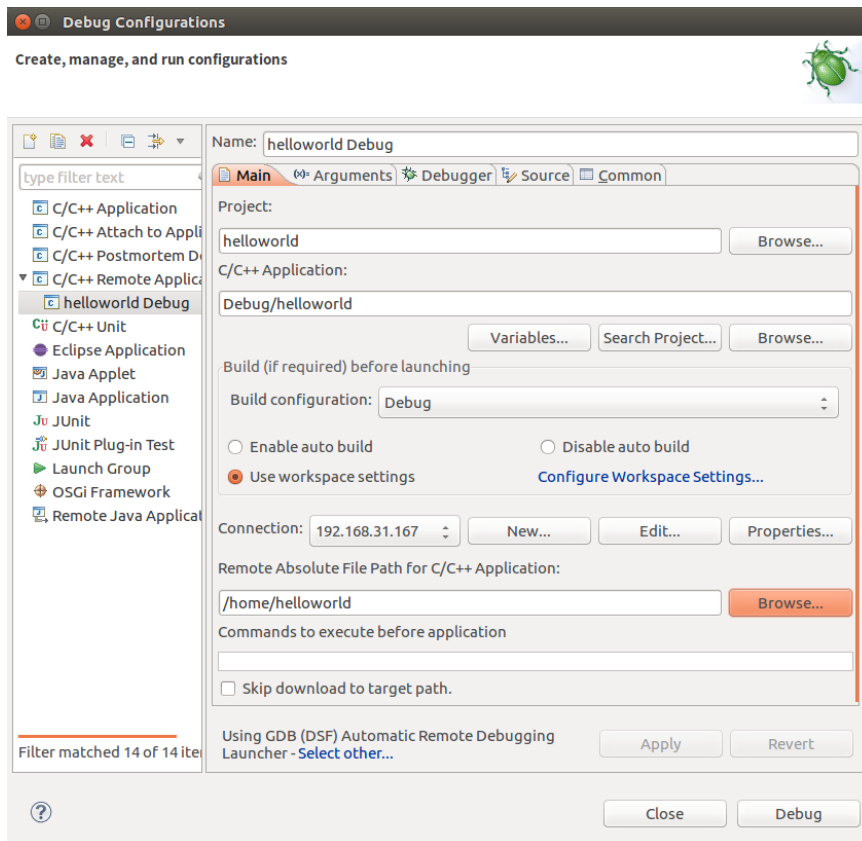
- ✓ 在 Eclipse 中, 点击 菜单 --> Run --> Debug Configurations...
- ✓ 创建一个 C/C++ Remote Application



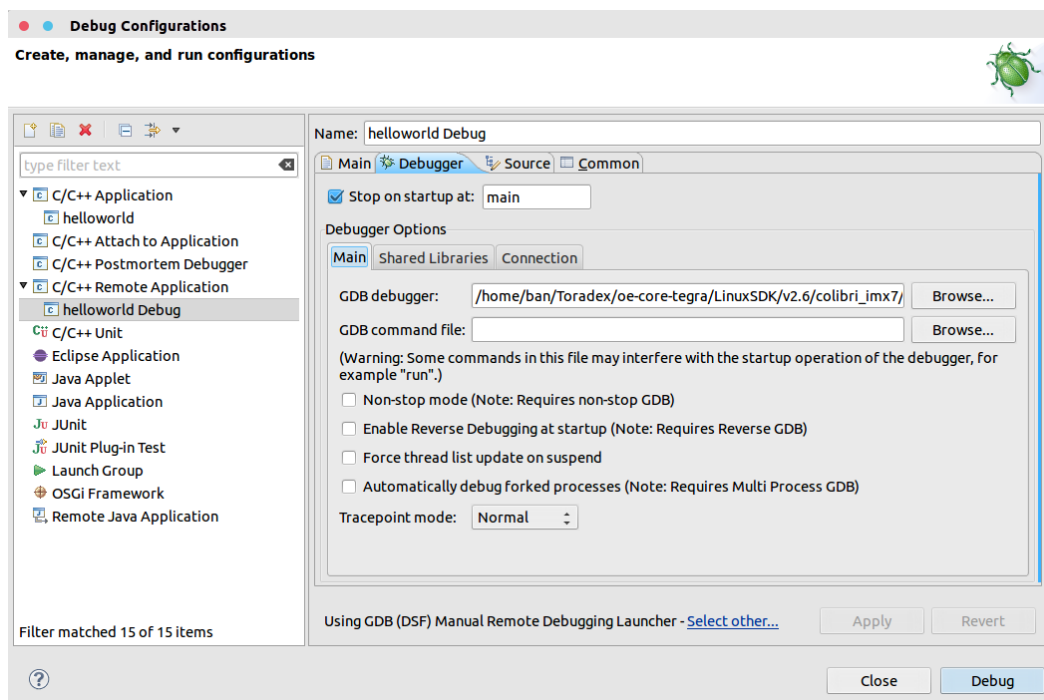
- ✓ 点击底部 Using GDB(DSF) Manual Remote Debugging Launcher – Select other...



- ✓ 点击 Browser，选择开发板上应用目录，一般选择 /home/root 目录

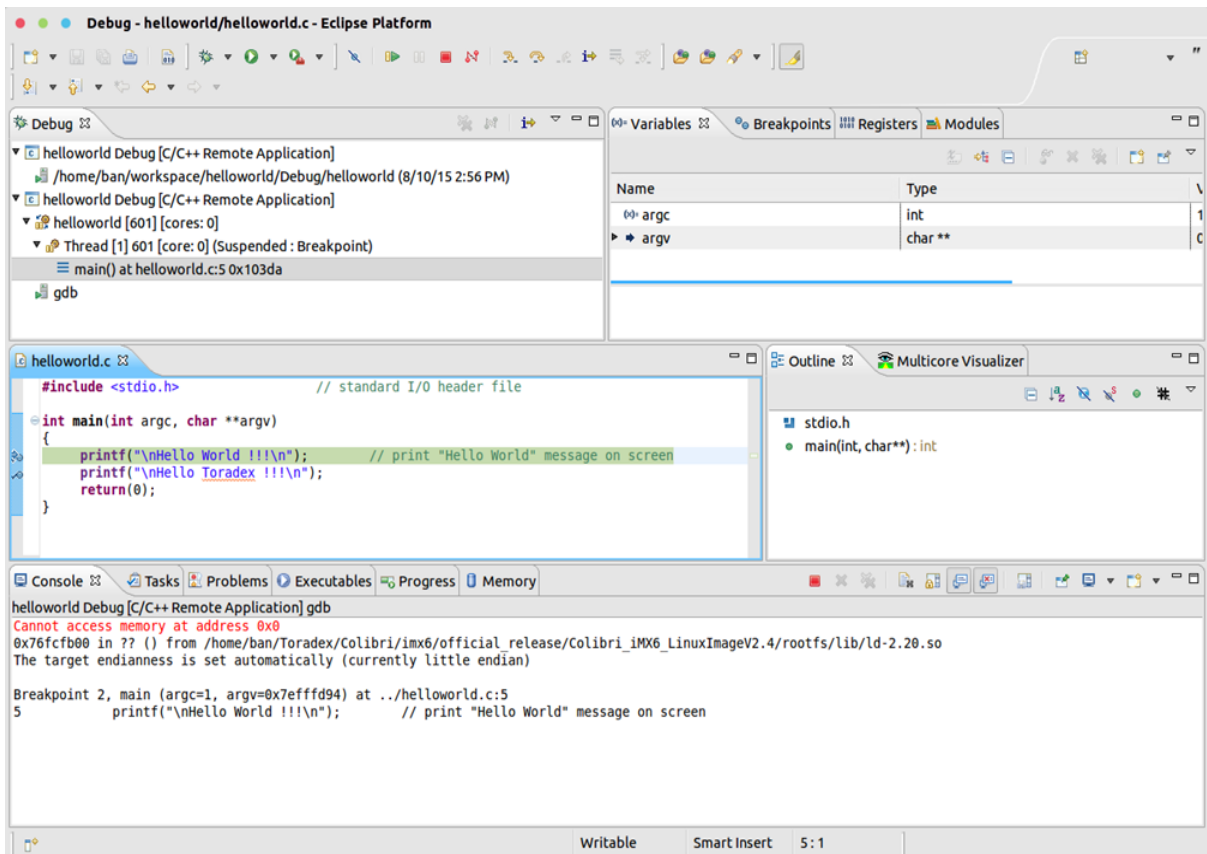
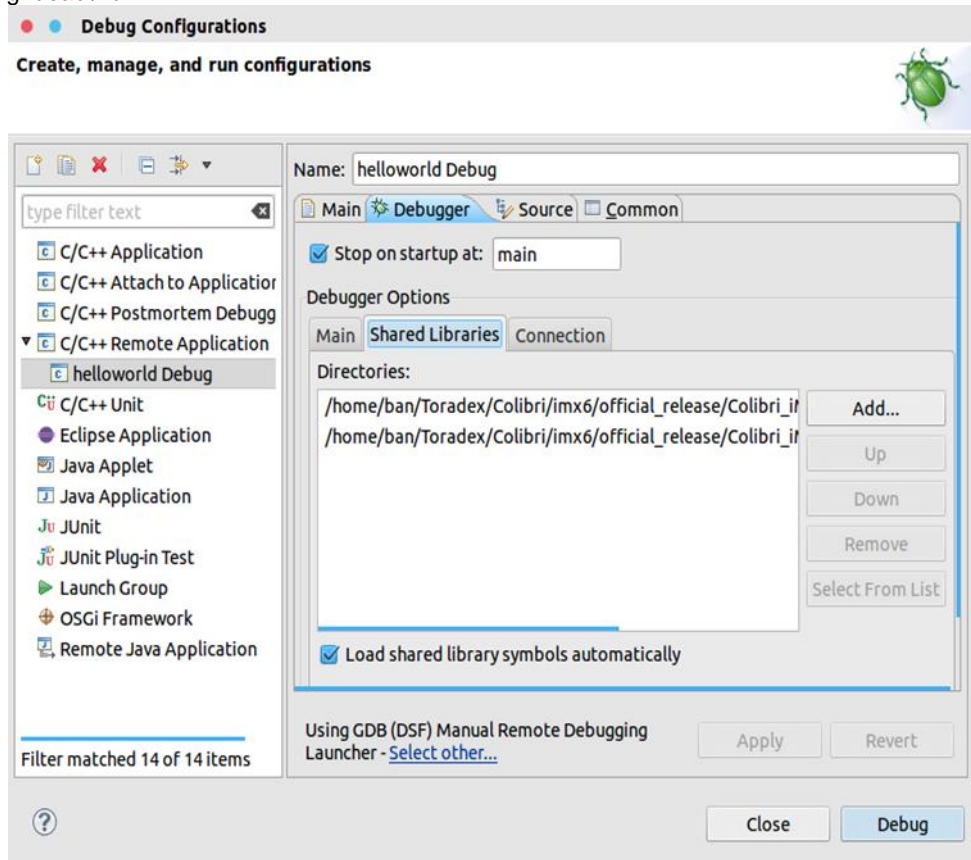


- ✓ 选择 GDB debugger  
/home/ban/Toradex/oe-core-tegra/LinuxSDK/v2.6/colibri\_vybird/sysroots/x86\_64-angstromsdk-linux/usr/bin/arm-angstrom-linux-gnueabi/arm-angstrom-linux-gnueabi-gdb



- ✓ 在 Debugger 标签中也，添加共享库文件目录，库文件位于 SDK 文件夹

```
/home/ban/Toradex/oe-core-tegra/LinuxSDK/v2.6/colibri_vybrid/sysroots/armv7at2hf-vfp-neon-angstrom-  
linux-gnueabi/usr/lib  
/home/ban/Toradex/oe-core-tegra/LinuxSDK/v2.6/colibri_vybrid/sysroots/armv7at2hf-vfp-neon-angstrom-  
linux-gnueabi/lib
```



## 6. U-Boot 和 Linux 源代码下载

---

韜睿将 Colibri VFxx 的 U-Boot 和 Linux 源代码均公开在 韜睿的 git 服务器上，并持续维护更新。用户可以直接从服务器上下载源码，建议用户使用最新版本的 U-Boot 和 Linux，并下载最新的分支。

### 6.1 U-Boot

- ✓ `git clone -b 2016.11-toradex git://git.toradex.cn/u-boot-toradex.git`

### 6.2 Linux

- ✓ `git clone -b toradex_vf_4.4 git://git.toradex.cn/linux-toradex.git`

U-Boot 和 Linux 具体的配置和编译方法，请参考 [Build U-Boot and Linux Kernel from Source Code](#)

<http://developer.toradex.cn/software-resources/arm-family/linux/board-support-package/build-u-boot-and-linux-kernel-from-source-code>

在下载 Linux 源码时，请根据对应的模块选择需要的分支版本

[https://developer.toradex.cn/knowledge-base/build-u-boot-and-linux-kernel-from-source-code#Image\\_Versions](https://developer.toradex.cn/knowledge-base/build-u-boot-and-linux-kernel-from-source-code#Image_Versions)

当前系统安装的 BSP 版本通过下面命令查看

```
┌-----┐  
| ~# cat /etc/issue |  
└-----┘
```

如果你需要使用特定的版本，除了制定分支外，还需要 checkout 至对应的 tag

<http://git.toradex.cn/cgit/linux-toradex.git/refs/tags>

## 7. MQX 开发

Colibri VF61 的处理器除了 Cortex-A5 之外，还具有 Cortex-M4，可以在该核上运行如 MQX、eCos 等实时操作系统。由于 MCC 版本关系，Toradex Colibri VF61 Linux BSP 从 V2.6 开始不再支持 MCC，因此，建议使用 Colibri\_VF\_LinuxImageV2.5\_20151215。

### 7.1 软件下载

MQX 4.1.2 for vybrid

<https://nxp.flexnetoperations.com/control/frse/download?element=6003621>

交叉编译工具

[https://launchpad.net/gcc-arm-embedded/4.8/4.8-2014-q1-update/+download/gcc-arm-none-eabi-4\\_8-2014q1-20140314-linux.tar.bz2](https://launchpad.net/gcc-arm-embedded/4.8/4.8-2014-q1-update/+download/gcc-arm-none-eabi-4_8-2014q1-20140314-linux.tar.bz2)

Colibri\_VF\_LinuxImageV2.5\_20151215

[http://developer1.toradex.com/files/toradex-dev/uploads/media/Colibri/Linux/Images/Colibri\\_VF\\_LinuxImageV2.5\\_20151215.tar.bz2](http://developer1.toradex.com/files/toradex-dev/uploads/media/Colibri/Linux/Images/Colibri_VF_LinuxImageV2.5_20151215.tar.bz2)

```
-----  
| $ tar xzf "~/Downloads/MQX RTOS 4.1.2 for Vybrid Linux Base.gz.gz"  
| $ tar xjf ~/Downloads/gcc-arm-none-eabi-4_8-2014q1-20140805-linux.tar.bz2  
|-----
```

### 7.2 MQX 配置

```
-----  
| $ cd FSLMQXOS_4_1_2_Vybrid/  
| $ gedit build/common/make/global.mak  
|-----
```

global.mak:

TOOLCHAIN\_ROOTDIR = /home/yourname/gcc-arm-none-eabi-4\_8-2014q1/

配置 MQX 使用的外设，这里只使用 TTYC 对应 Colibri VF61 的 UARTB。修改 config/twrvf65gs10\_m4/user\_config.h



```

...
#define BSPCFG_ENABLE_TTYB          0
#define BSPCFG_ENABLE_ITTYB        0

#define BSPCFG_ENABLE_TTYC          1
#define BSPCFG_ENABLE_ITTYC        0
...
...
#define BSPCFG_ENABLE_I2C0          0
#define BSPCFG_ENABLE_I12C0        0
...
...
#define BSPCFG_ENABLE_SPI0          0
...
...
#define BSPCFG_ENABLE_RTCDEV        0
...
...
#define BSPCFG_ENABLE_ESDHC         0
#define BSPCFG_ENABLE_SAI           0
...

```

修改时钟 ./mqx/source/bsp/twrvf65gs10\_m4/twrvf65gs10\_m4.h

```

...
#define BSP_CORE_CLOCK              (166700000)
#define BSP_BUS_CLOCK               (166700000)
#define BSP_SYSTEM_CLOCK            (83300000)
#define BSP_IPG_CLOCK               (83300000)
...

```

MQX 4.1.2 上 TTYC 的引脚配置同 Toradex 定义的 UARTB 引脚不同，需要在下面文件重新配置引脚  
FSLMQXOS\_4\_1\_2\_Vybrid/mqx/source/bsp/twrvf65gs10\_m4/init\_gpio.c

```

case 2: // SCI2
    IOMUXC_SCI_FLX2_IPP_IND_SCI_RX_SELECT_INPUT = 0x00000000;
    IOMUXC_SCI_FLX2_IPP_IND_SCI_TX_SELECT_INPUT = 0x00000000;
    IOMUXC_RGPI0(79) = 0x002021A2;
    IOMUXC_RGPI0(80) = 0x002021A2;
    break;
}
return MQX_OK;
}

```

## 7.3 编译 MQX

进入 `mqx/examples/hello/build/make/hello_twrvf65gs10_m4/` 目录，使用下面的文件替换原来的 `build_gcc_arm.sh`

<http://docs.toradex.com/103324-build-gcc-arm-sh-tar.gz>

```
-----  
| # cd mqx/examples/hello/build/make/hello_twrvf65gs10_m4/  
| # ./build_gcc_arm.sh  
| ...  
| Build done  
|  
| Press any key to continue...  
|-----
```

生成二进制应用程序

```
-----  
| $ export PATH=~/gcc-arm-none-eabi-4_9-2014q4/bin:$PATH  
| $ arm-none-eabi-objcopy -O binary gcc_arm/ram_release/hello_twrvf65gs10_m4.elf  
| gcc_arm/ram_release/hello_twrvf65gs10_m4.bin  
|-----
```

查看应用入口地址，由于二进制程序没有可以指明加载地址或入口地址的头部信息，因此需要在加载的时候手动添加。

```
-----  
| $ arm-none-eabi-objdump -f gcc_arm/ram_release/hello_twrvf65gs10_m4.elf  
|  
| gcc_arm/ram_release/hello_twrvf65gs10_m4.elf: file format elf32-littlearm  
| architecture: arm, flags 0x00000112:  
| EXEC_P, HAS_SYMS, D_PAGED  
| start address 0x3f000791  
|-----
```

## 7.4 外设资源分配

VF61 处理器没有外设资源管理器，Linux 和 MQX 均可以访问所有外设资源。为了不引起访问冲突，MQX 可以采用上面提到的方法，设置所需的资源。Linux 则使用 Device tree 对外设资源进行管理。

修改 `linux-toradex/arch/arm/boot/dts/vf-colibri-eval-v3.dtsi`，在 Linux 禁用 UARTB。

```
-----  
| &uart2 {  
|     status = "disabled";  
| };  
|-----
```

重新编译 device tree 文件，并将生成的 vf610-colibri-eval-v3.dtb 复制到 Colibri VF61 的 /boot/ 目录，替换原来的文件。

Linux 会禁用不使用外设的时钟，但是 UARTB 会在 M4 端被 MQX 所使用，因此仍旧需要为 UARTB 分配时钟。在 U-Boot 使 Linux 忽略不使用的外设的时钟。

```
Colibri VFxx # setenv defargs clk_ignore_unused
Colibri VFxx # saveenv
```

重新启动后，Linux 已经无法访问 UARTB。

```
root@colibri-vf:~# ls /dev | grep ttyLP
ttyLP0
ttyLP1
```

详细的编译请参考：

Build U-Boot and Linux Kernel from Source Code

<http://developer.toradex.com/knowledge-base/build-u-boot-and-linux-kernel-from-source-code>

## 7.5 运行 MQX

将 UARTB 连接指定电脑串口中断，波特率为 115200，8n1 格式。将 MQX 应用加载到 OCRAM。除了 OCRAM (256KB)，还可以使用 DRAM (从 UBoot 中地址空间) 以及 TCML (32KB)。详细的地址请参考 Vybrid 手册。

```
# modprobe mcc
# mqxboot hello_twrvf65gs10_m4.bin 0x3f000000 0x3f000791
```

在 UARTB 的串口终端里可以看到

```
Hello World
```

详情请参考：MQX™ on the Cortex-M4 of a Colibri VF61

<http://developer.toradex.com/knowledge-base/mqx-on-the-cortex-m4-of-a-colibri-vf61>

## 8. BSP 更新

---

韬睿会不定期地发布针对 Colibri VFxx 模块的 BSP，这些 BSP 会解决历史版本中遗留的或者用户报告的问题，以及为了提高相应模块的性能而增加的补丁等。关于 BSP 的说明请参考以下网页 Linux BSP Release Details：

<http://developer.toradex.com/software-resources/arm-family/linux/linux/release-details?view=all>

- ✓ 下载 Colibri\_VF\_LinuxImageVx.yz.tar.bz2，以 root 权限解压
- ✓ 准备 1GB 以上 VFAT 格式的 SD 卡
- ✓ 执行脚本 `sudo update.sh -o /PATH_TO_SD_CARD`
- ✓ EvalBoard V3.1A/V3.2A:

将 SD 卡插入载板上 SD/MMC socket (X15)

- ✓ Iris V1.1A:

将 micro SD 插入载板 SD card socket (X10)

复位之后，重新进入 U-Boot 命令模式，执行下面命令，完成 BSP 更新

```
Colibri VFxx # run setupdate  
Colibri VFxx # run update
```

详细的说明，请参考 Flashing Embedded Linux to Vybrid Modules

<http://developer.toradex.com/knowledge-base/flashing-linux-on-vybrid-modules>

## 9. 其他开发参考资料

---

### 9.1 OpenEmbedded

韜睿的 BSP 通过 OpenEmbedded/Yocot 进行发布，用户可以下载、编译完整的 BSP，以及自行裁剪 BSP，或者添加第三方软件等，具体请参考 OpenEmbedded (core)

<http://developer.toradex.com/software-resources/arm-family/linux/board-support-package/openembedded-%28core%29>

### 9.2 开发者中心

韜睿的开发者中心提供了丰富的学习资料，内容涵盖驱动移植、应用开发、多媒体显示、外设操作、内核编译等嵌入式开发常见知识。用户可以从中学习简单的开发，到具体应用的深入研究。上面的文章均有韜睿位于瑞士总部的研发工程师编写和维护，直接保证文章的时效性和可用性。

<http://developer.toradex.com>

更多关于 Linux 的开发，请点击下面网页：

<http://developer.toradex.com/knowledge-base/getting-started-with-toradex-arm-modules#Linux>

### 9.3 Toradex 社区

在 Toradex 社区里，任何人都可以询问关于我们产品的问题，我们的研发人员将会解答您的问题。我们创建这个社区，是因为我们发现很多用户都会遇到同样或者类似的问题，公开这些解答，使得用户能够更快地找到答案。

<https://www.toradex.com/community>

## 10. Toradex 公司简介及本地支持联系方式

---

### 10.1 Toradex 公司简介

Toradex 是一个领先的 ARM®计算机模块 (CoM) / 系统模块 (SoM) 厂商, 产品被众多嵌入式应用所采用。使用先进的 ARM 处理器, 例如 Freescale® i.MX 6 & Vybrid™, NVIDIA® Tegra™, 和 Marvell® PXA, 韜睿的计算机模块系列在价格、性能、功耗和接口方面为用户提供丰富的选择。使其成为满足各种嵌入式计算市场需求的理想模块。在嵌入式市场中, Toradex 因其产品的稳定性、长生命周期、免费的高级技术支持和透明的价格体系而与众不同。Toradex 能够帮助用户降低其嵌入式产品的上市时间以及研发成本。

### 10.2 本地支持联系方式

韜睿 (上海) 计算机科技有限公司

公司网址 – [www.toradex.cn](http://www.toradex.cn)

公司地址 – 上海市徐汇区虹桥路 188 号 1806 室

联系电话 - 021-54380582

以上所有的信息仅供您的参考, 其中并不包含任何承诺。Toradex 会不定期发布软件更新, 以上信息不保证能够适用于最新的软件。关于文中任何的错误、遗漏或者链接内容, 我们对此不承担责任。