

CLV-5061/5061A 用户手册

(USB总线 ARINC 429通信模块)



成都科洛威尔科技有限公司

地址：成都市高新西区双柏路 68 号 23 栋

TEL: 1878-0222-336 191-3621-6517

EMAIL: cloverttech@163.com 公司官网: www.clvtech.net



目 录

适用范围.....	1
内容简介.....	2
版本历史.....	3
1. CLV-5061 模块概述.....	4
1.1. 模块特点.....	4
1.2. 软件特性.....	4
1.3. 环境特性.....	4
1.4. 平台特性.....	4
1.5. 订购信息.....	4
2. 配套资料说明.....	5
3. CLV-5061 盒式 USB-ARINC429 通信模块.....	5
3.1. 机械尺寸.....	5
3.2. 连接器信号定义.....	5
3.3. CLV-5061 模块安装.....	6
3.3.1. 安装准备.....	6
3.3.2. 板卡安装.....	7
3.3.3. Windows 驱动程序的安装.....	7
4. CLV-5061A 内嵌式 USB-ARINC429 通信模块.....	10
4.1. 机械尺寸.....	10
4.2. CLV-5061A 连接器信号定义.....	10
4.3. 测试工装.....	11
5. ARINC429 总线简介.....	13
5.1. 电气特性.....	13
5.2. 网络拓扑.....	14
5.3. 数据格式.....	15
5.3.1. 发送与接收位序.....	15
5.3.2. 各个字段的说明.....	15
6. 实现原理.....	16



7. 示例程序使用说明	17
7.1. 综述	17
7.2. 操作流程和使用说明	18
7.3. 自发自收测试	20
8. Wxworks 系统下安装及使用说明	22
8.1. 环境配置	22
8.2. 组件配置	26
8.2.1. 新建 VIP 工程	26
8.2.2. 模块功能配置	28
8.2.3. 总线功能测试	29
8.2.3.1. 新建 DKM 工程	29
8.2.3.2. 工程配置	30
8.2.3.3. 编译运行	32
8.2.3.4. 其他说明	33
9. Linux 系统下安装及使用说明	34
9.1. Libusb 安装	34
9.1.1. 配置 Libusb 在 X86 系统下使用	34
9.1.2. 配置 Libusb 在 ARM 系统下使用	36
9.2. Libusb 基本测试	38
9.2.1. X86 系统下测试	38
9.2.2. ARM 系统下基本测试	40
9.3. ARINC429 功能测试	41
10. 产品选型说明	43

适用范围

本手册适用于以下产品：

CLV5061 V2.0;

CLV5061A V1.0。

内容简介

本手册分十个章节：

第一章 模块概述；

第二章 配套资料说明；

第三章 CLV-5061 盒式 USB-ARINC429 通信模块

第四章 CLV-5061A 内嵌式 USB-ARINC429 通信模块

第五章 ARINC429 总线简介

第六章 实现原理

第七章 示例程序使用说明

第八章 VxWorks 系统下安装和使用说明

第九章 Linux 系统下安装和使用说明

第十章 产品选型说明

版本历史

日期	版次	说明
2017 年 10 月	V1.0	初始版本
2021 年 8 月	V2.0	结构接口升级
2023 年 9 月	V2.1	例程更新
2024 年 5 月	V2.2	增加数据记录查看相关说明 增加 3.3 节 数据格式 增加 6.3 节 自发自收测试
2024 年 9 月 25 日	V3.0	将 5061, 5061A 产品手册合并

1. CLV-5061 模块概述

CLV-5061 系列 USB 接口 ARINC 429 总线接口通信模块，最多可以提供 8 个标准发送和 8 个标准接收通道。该系列产品有内嵌式模块和带外壳独立盒式模块两种产品形态。CLV -5061 系列 USB-ARINC429 模块配有丰富易用的 API 函数接口、开发例程，可适配 Windows、Vxworks 等操作系统，是 ARINC429 通信接口快速实现、现有设备通信接口拓展升级、便携测试等的极佳选择。

1.1. 模块特点

- 波特率 12.5Ksps、48Ksps、50Ksps、100Ksps
- 奇偶校验
- 接收数据可编程过滤
- 循环定时发送、单次数据发送
- 字间隔、循环数据间隔可设置
- 时间标签
- API 函数库及丰富完备的开发例程

1.2. 软件特性

- 支持 windows XP/win7/win10 操作系统，可定制支持 VxWorks, Linux 等
- 支持 Microsoft VC++, labview 等多种开发工具

1.3. 环境特性

- 操作温度：-20~+70°C
- 存储温度：-40~85°C
- 相对湿度：5%~95%（无凝结）
- 可提供宽温版本

1.4. 平台特性

- USB 2.0
- 供电：5V @ 0.4A

1.5. 订购信息

- CLV-5061-mT/nR 盒式 USB 总线的 ARINC429 接口通信模块

- CLV-5061A-mT/nR 内嵌式 USB 总线的 ARINC429 接口通信模块
m 为发送的通道数, $0 \leq m \leq 8$; n 为接收的通道数, $0 \leq n \leq 8$ 。

2. 配套资料说明

产品配有二次开发支持包, 驱动程序、API 函数库、示例工程以及相关文档。

二次开发包以电子文档或者光盘的形式提供。

序号	文件目录	说明
1	/api	API 函数库动态库文件
2	/demo	示例程序工程 (开发环境 labwindows CVI 2017)
3	/demo_setup	示例程序安装文件
4	/doc	文档手册
5	/driver	驱动程序
6	/Linux	linux 系统下的驱动、例程
7	/Vx6.9	Vxworks 系统下的驱动、例程
8	其他例程	控制台程序例程、Labview 例程

3. CLV-5061 盒式 USB-ARINC429 通信模块

CLV-5061 是一款独立盒式 USB-ARINC429 通信产品, 最多可以提供 8 个标准发送和 8 个标准接收通道。



图 1 CLV-5061

3.1. 机械尺寸

机械尺寸: 120mm x 74mm x 28mm (外壳, 不含连接器长度)

3.2. 连接器信号定义

CLV5061 V2.0 对外提供一个 SCSI 50 芯 CN 型母头连接器, 信号定义如下:

引脚号	信号定义	引脚号	信号定义
1	TXA_7	26	TXA_0
2	TXB_7	27	
3	TXB_6	28	TXB_0
4	TXA_6	29	
5	TXB_5	30	TXA_1
6	TXA_5	31	
7	TXB_4	32	TXB_1
8	TXA_4	33	
9	TXB_3	34	TXA_2
10	RXA_7	35	
11	RXB_7	36	TXB_2
12	RXA_6	37	
13	RXB_6	38	TXA_3
14	RXA_5	39	
15	RXB_5	40	
16	RXA_4	41	GND
17	RXB_4	42	
18	RXA_3	43	GND
19	RXB_3	44	GND
20	RXA_2	45	GND
21	RXB_2	46	
22	RXA_1	47	
23	RXB_1	48	GND
24	RXA_0	49	GND
25	RXB_0	50	GND

信号说明:

TXA_n:ARINC429 发送信号正, n 表示通道号。

TXB_n:ARINC429 发送信号负, n 表示通道号。

RXA_n:ARINC429 接收信号正, n 表示通道号。

RXB_n:ARINC429 接收信号负, n 表示通道号。

3.3. CLV-5061 模块安装

3.3.1. 安装准备

注意静电防护。

在使用前请先检查使用环境是否良好接地。

在第一次使用时请检查模块在运输中是否有破损，如发现有破损请及时与供应商联系。

3.3.2. 板卡安装

将 USB 线缆一端接 CLV-5061 的 USB 口，另一端接计算机的 USB 口。如下图所示：



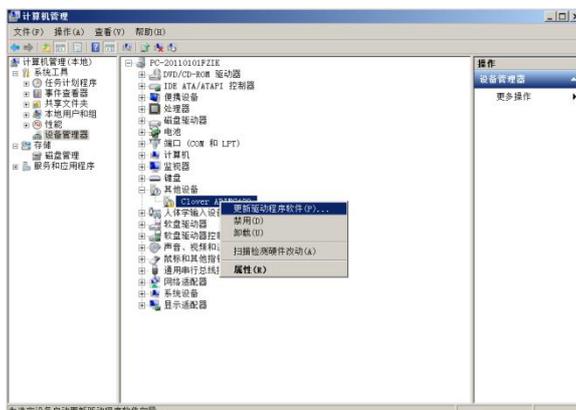
图 2 CLV-5061 与计算机连接

3.3.3. Windows 驱动程序的安装

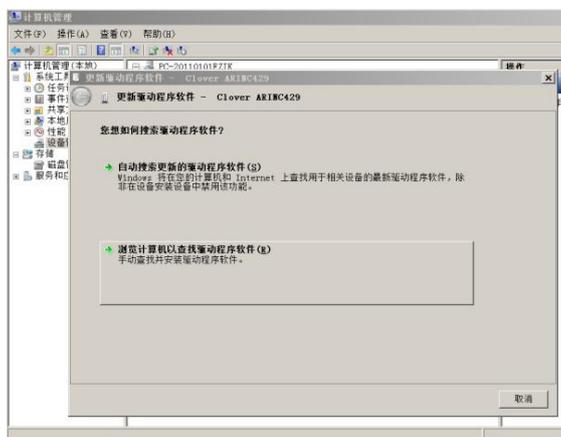
驱动程序在随产品配套的光盘上/driver 目录下。模块接入计算机后。如果是第一次使用本模块，操作系统会提示发现新硬件，要求安装驱动。在设备管理器里面，可以查看到已经发现“Clover ARINC429”设备。



在设备上点击右键，更新驱动：



再点选“浏览计算机以查找驱动程序”



在弹出的对话框里点击“浏览”，然后定位到驱动所在目录，然后点击下一步。



完成搜索后，将出现如下界面：



直接点击下一步，可能会出现如下提示：



点击是，完成安装。

4. CLV-5061A 内嵌式 USB-ARINC429 通信模块

CLV-5061A 体积小巧，可置于设备内部，基于 USB 实现 8 个标准发送和 8 个标准接收 ARINC429 通道。



图 3 CLV-5061A 内嵌式 USB-ARINC429 模块

4.1. 机械尺寸

尺寸：40*70*10.5（mm）

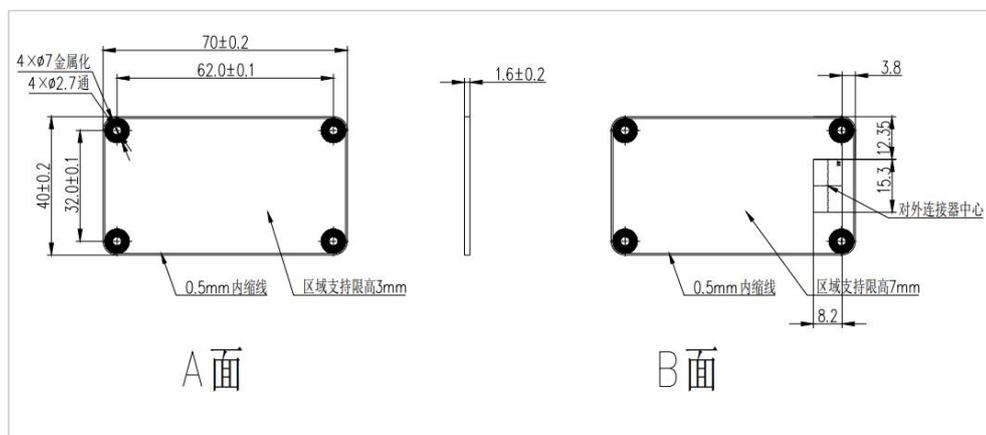


图 4 CLV-5061M 机械尺寸

4.2. CLV-5061A 连接器信号定义

CLV-5061A 对外接口连接器型号：TOLC-110-02-L-Q-A（厂商：SAMETC）。

信号定义如下：

管脚号	网络定义	说明	管脚号	网络定义	说明
A1	+5V	DC_+5V 电源正端	B1	+5V	DC_+5V 电源正端
A2	GND	DC_+5V 电源地	B2	GND	DC_+5V 电源地
A3	RX1_B	第 1 路 429 接收 B 端	B3	RX1_A	第 1 路 429 接收 A 端
A4	RX2_B	第 2 路 429 接收 B 端	B4	RX2_A	第 2 路 429 接收 A 端
A5	RX3_B	第 3 路 429 接收 B 端	B5	RX3_A	第 3 路 429 接收 A 端
A6	RX4_B	第 4 路 429 接收 B 端	B6	RX4_A	第 4 路 429 接收 A 端
A7	RX5_B	第 5 路 429 接收 B 端	B7	RX5_A	第 5 路 429 接收 A 端
A8	RX6_B	第 6 路 429 接收 B 端	B8	RX6_A	第 6 路 429 接收 A 端
A9	RX7_B	第 7 路 429 接收 B 端	B9	RX7_A	第 7 路 429 接收 A 端
A10	RX8_B	第 8 路 429 接收 B 端	B10	RX8_A	第 8 路 429 接收 A 端
C1	TX1_B	第 1 路 429 发送 B 端	D1	TX1_A	第 1 路 429 发送 A 端
C2	TX2_B	第 2 路 429 发送 B 端	D2	TX2_A	第 2 路 429 发送 A 端
C3	TX3_B	第 3 路 429 发送 B 端	D3	TX3_A	第 3 路 429 发送 A 端
C4	TX4_B	第 4 路 429 发送 B 端	D4	TX4_A	第 4 路 429 发送 A 端
C5	TX5_B	第 5 路 429 发送 B 端	D5	TX5_A	第 5 路 429 发送 A 端
C6	TX6_B	第 6 路 429 发送 B 端	D6	TX6_A	第 6 路 429 发送 A 端
C7	TX7_B	第 7 路 429 发送 B 端	D7	TX7_A	第 7 路 429 发送 A 端
C8	TX8_B	第 8 路 429 发送 B 端	D8	TX8_A	第 8 路 429 发送 A 端
C9	USB_Data+	USB 差分正端	D9	USB_VCC	USB 电源
C10	GND	USB 地	D10	USB_Data-	USB 差分负端

RxA_n: ARINC 429 接收通道正端,n 为通道号。

RxB_n: ARINC 429 接收通道负端,n 为通道号。

TxA_n: ARINC 429 发送通道正端,n 为通道号。

TxB_n: ARINC 429 发送通道负端,n 为通道号。

4.3. 测试工装

CLV-5061A 备有测试工装（底板）可供选配，便于二次开发调试。



图 5 CLV-5061A 测试工装

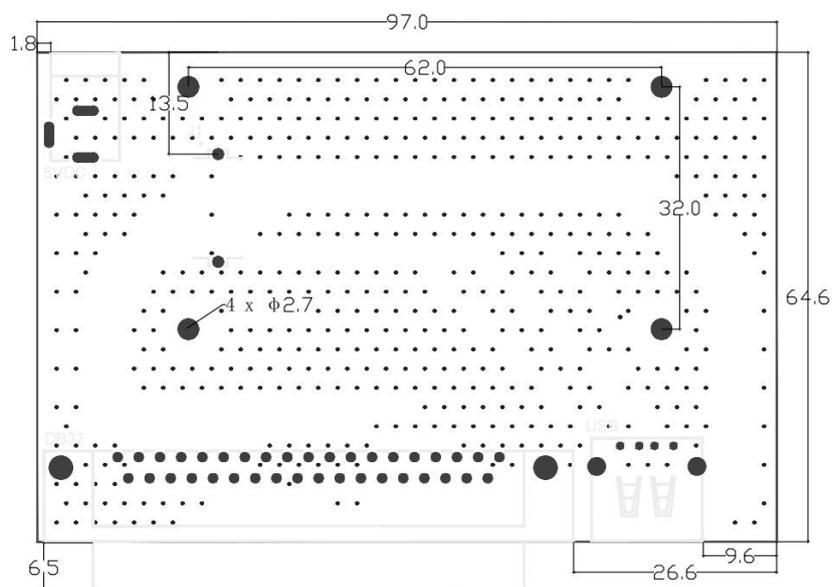


图 6 CLV-5061A 工装尺寸 (mm)

测试工装使用外接 DC 5V 电源适配器供电、通过标准 USB Type A 公头线缆接入计算机 USB 口。ARINC429 信号通过一个 DB37 连接器引出，其定义如下：

管脚号	功能	管脚号	功能
37	GND	19	VCC
36	TxA0	18	RxA0
35	TxB0	17	RxB0
34	TxA1	16	RxA1
33	TxB1	15	RxB1
32	TxA2	14	RxA2
31	TxB2	13	RxB2
30	TxA3	12	RxA3
29	TxB3	11	RxB3
28	TxA4	10	RxA4

27	TxB4	9	RxB4
26	TxA5	8	RxA5
25	TxB5	7	RxB5
24	TxA6	6	RxA6
23	TxB6	5	RxB6
22	TxA7	4	RxA7
21	TxB7	3	RxB7
20	-	2	-
		1	GND

5. ARINC429 总线简介

ARINC429 总线协议的全称是数字式信息传输系统 DITS。协议标准规定了航空电子设备及有关系统间的数字信息传输要求。ARINC429 广泛应用在民航客机中，如 A310/320、B-737、B757、B-767，俄制军用飞机也选用了类似的技术。我国与之对应的标准是 HB6096-SZ-01，在国产客机上，也用到了 ARINC429 协议。

5.1. 电气特性

ARINC 429 数据总线使用一对差分信号线传输数据，数据字为 32 位，字的传输至少间隔 4 位时间零（零电压）。差分信号正线，通常也被标记为“A”或“HI”；差分信号负，通常也被标记为“B”或“LO”，每根信号线上对地电平范围为 +5V~-5V，差分信号线之间的电压范围是 $10\pm 1V$ 。也即 ARINC429 采用的是双极性归零码（BPRZ），其电平识别如下：

高电平：7.25~11V（A to B）；

NULL：0.5~-0.5V（A to B）；

低电平: -7.25~11V (A to B) ;

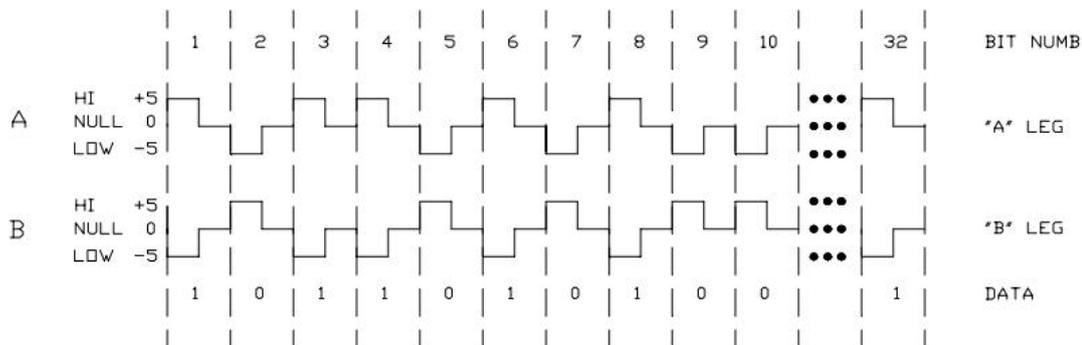


图 7 ARINC429 信号编解码

5.2. 网络拓扑

ARINC429 是一种简单可靠的点对点通信协议, 接收器输入阻抗不小于 8KΩ, 在一个 ARINC429 通信网络中可以只有一个发送端, 和最多 20 个接收端, 接收信号电压跟线缆长度和接收器挂接数量有关系。ARINC429 通信传输, 一般采用 78Ω 的屏蔽双绞线, 传输距离可达 150~300 英尺。

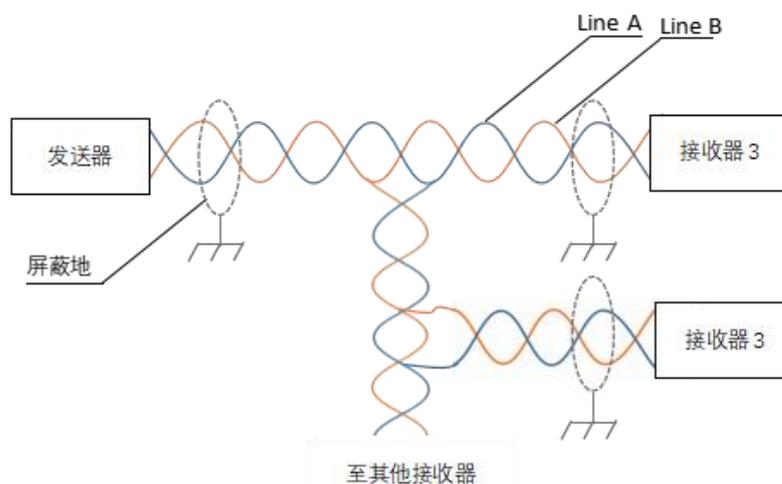


图 8 ARINC429 网络拓扑

5.3. 数据格式

5.3.1. 发送与接收位序

一个 ARINC429 字是一个 32 位的数据。写入硬件缓存后，CLV-5061 的发送位序如下：

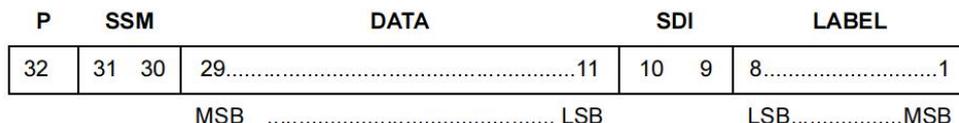


图 9 ARINC429 字发送位序

这些位在发送时，从 bit 1 开始发送，最后发出的是校验位（bit 32）。标号字段（LABEL）先发送的是最高位（MSB）。数据字段（DATA），先发送的是最低位（LSB）。

在接收时，CLV-5061 会自动按整体 32 位的 LSB 到 MSB 进行存储。也就是说，读取出来的接收数据，标号字段将被还原为和数据字段一致的高低位序。比如利用 CLV-5061 自发自收时，填入发送数据 0x80000f31,启动收发后，接收读取到的数据也是 0x80000f31。

5.3.2. 各个字段的说明

字段	英文全称及说明
Label	Information identifier
SDI	Source/Destination Identifier bits
DATA	Information data
SSM	Sign/Status Matrix bits
P	Parity bit (Odd parity)

Label: 在应用系统中根据设备类型进行定义，标示不同类型设备的数据。在实际应用中，一般用八进制 1 ~ 377 表示。

SDI: 当一个发射机连接到多个接收器，但并非所有数据都将被所有接收器使用时，将使用 SDI 字段。在这种情况下，每个接收器将被分配一个 SDI 值，并且将只查看与其 SDI 值相匹配的标签。CLV-5061 板卡默认接收所有 SDI 值的数据，且支持 SDI 过滤。

DATA: DATA 字段包含要发送的实际数据。这些数据按应用接口协议约定进行编解码。

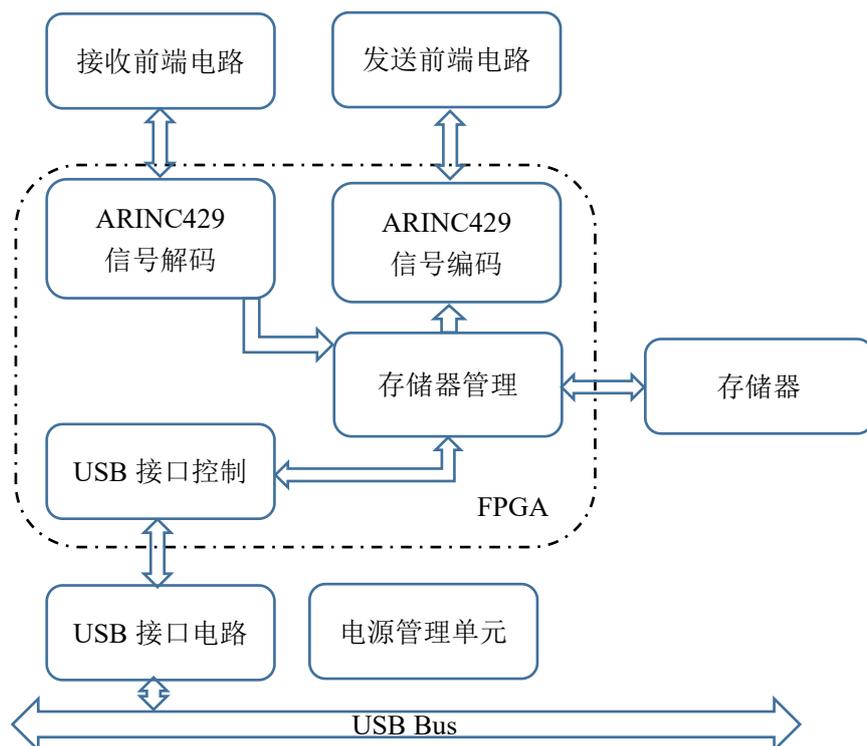
SSM: 3 位长，用于提供信息，以帮助解释数据字段中的数值。SSM 值的例子可能是加、北、东、右、到或以上。

P 是奇偶校验位。

实际上，对于用户来说，CLV-5061 板卡发送的 32 位数据，都是可以编辑的。当校验方式设置为“无”时，校验位 (bit 32) 也由应用层作为数据位写入；当校验方式设置为“奇校验”或者“偶校验”时，校验位由板卡硬件按设置生成。

6. 实现原理

CLV5061 基于 FPGA 设计，在 FPGA 中实现了 ARINC429 信号的编解码、收发数据的存储管理以及 USB 接口交互。整个模块可以基于 USB 接口实现 8 路独立 ARINC429 发送通道和 8 路独立 ARINC429 接收通道。



7. 示例程序使用说明

7.1. 综述

随 CLV-5061 通信接口模块配套有应用示例软件。该软件，界面简约，包含绝大多数模块应用的基本功能，免费向客户提供源代码。

特别说明：该示例软件作为产品附属品提供，仅作为一般功能测试和参考示例，科洛威尔不对该程序的功能完备性及场景适应性作任何担保；在购买相关硬件产品后，用户可对该程序作任何形式的修改。

示例程序的开发环境为：win7 32bit ， labwindows CVI 2017。

示例软件主要用于客户对产品的测试，功能验证以及作为二次应用开发参考。

软件运行主界面由“参数设置区”和“数据编辑及显示区”组成，如下图：



7.2. 操作流程和使用说明

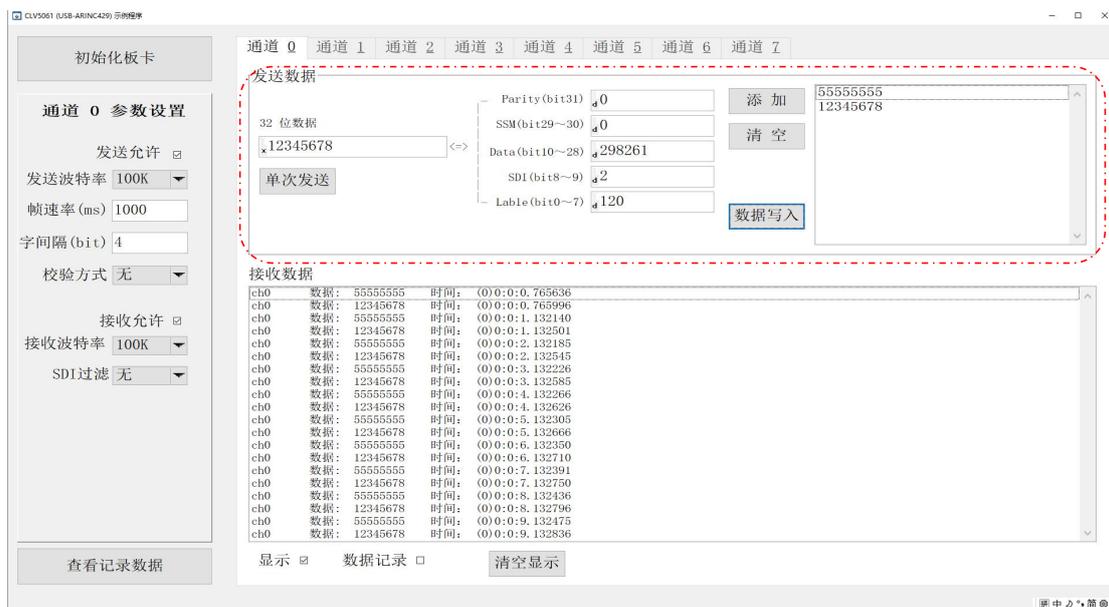
- (1) 点击“初始化板卡”按钮，初始化板卡。
- (2) 在右侧的选项页，选择要操作的通道。
- (3) 在界面左侧参数设置区，设置通道参数：



可设置参数包括：通道发送使能、发送波特率、发送帧速率、发送字间隔、

发送校验方式、通道接收使能、接收波特率、接收 SDI 过滤。

(4) 数据编辑与发送



数据编辑：

发送数据可以直接在“32 位数据”区域编辑，也可以在其右侧的字段编辑区编辑，二者是联动的。”清空“按钮可以删除一个通道上所有的周期性发送数据。

周期数据发送：

数据编辑完成后，点击“添加”按钮，将数据添加到软件缓存中，已经添加的数据在右侧的列表框中显示。当所有发送数据添加完成后，再点击“数据写入”按钮，将数据写入硬件发送缓冲区，并立即按参数设置进行周期性发送。

单次发送：

数据编辑完成后，点击“单次发送”按钮，数据将立即发出，并只发送一次。单次发送一般用于突发数据发送。

(5) 接收数据

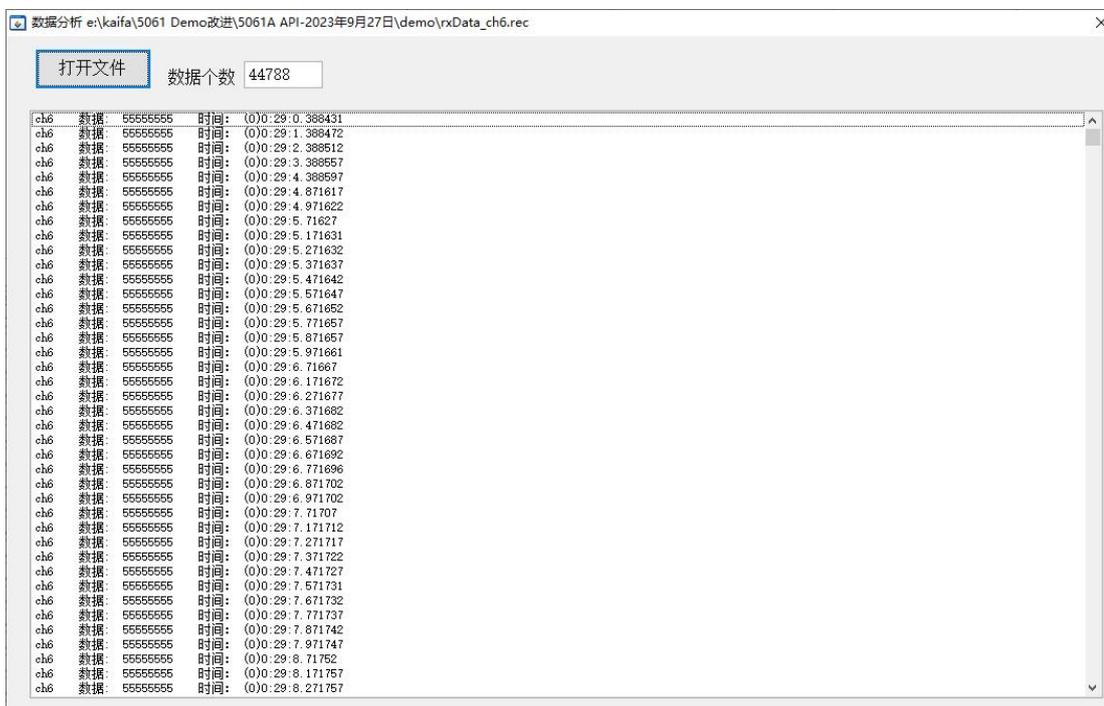
接收数据在“接收数据”显示列表框中显示。显示框下方有两个勾选框：

显示：选择数据是否实时显示；

数据记录：选择是否对接收数据进行文件记录。

(6) 数据记录查看

点击界面右下角的“查看记录数据”按钮，将打开数据查看界面：



点击“打开文件”按钮，选择数据记录文件，将可以查看记录内容。

例程中记录数据分通道存为文件，存放于工程目录下，命名形式为“rxData_chn.rec”，n 为通道号。

***注意：**当记录功能开启时，对应的记录文件处于写入操作状态，此时若点击“打开文件”，该操作将被阻止，并提示“文件正在被使用,请停止通道 x 数据记录”。打开记录文件之前，请先将对应通道的记录功能关闭。

7.3. 自发自收测试

收到 CLV-5061 板卡后，如果选用的是发送、接收通道都有的配置型号，可以通过自发自收进行板卡测试，熟悉板卡的使用方法。操作步骤如下：

- (1) 将配套 ARINC429 信号线缆发送和接收短接。
- (2) 如 4.1 章节所述正确安装板卡。

(3) 打开例程软件，如 6.2 章节所述，在已短接的发送通道上添加发送数据。

为便于测试观察，可以添加周期性发送数据。

(4) 在已经短接的接收通道对应的选项页，观察接收数据。

(5) 举例：0 通道自发自收测试

①短接 TXA0 – RXA0; TXB0 – RXB0。

②将 CLV-5061 和配套的 ARINC429 信号线缆对接，其 USB 口通过配套的 USB 线缆接入计算机，并确保驱动已经正确安装。

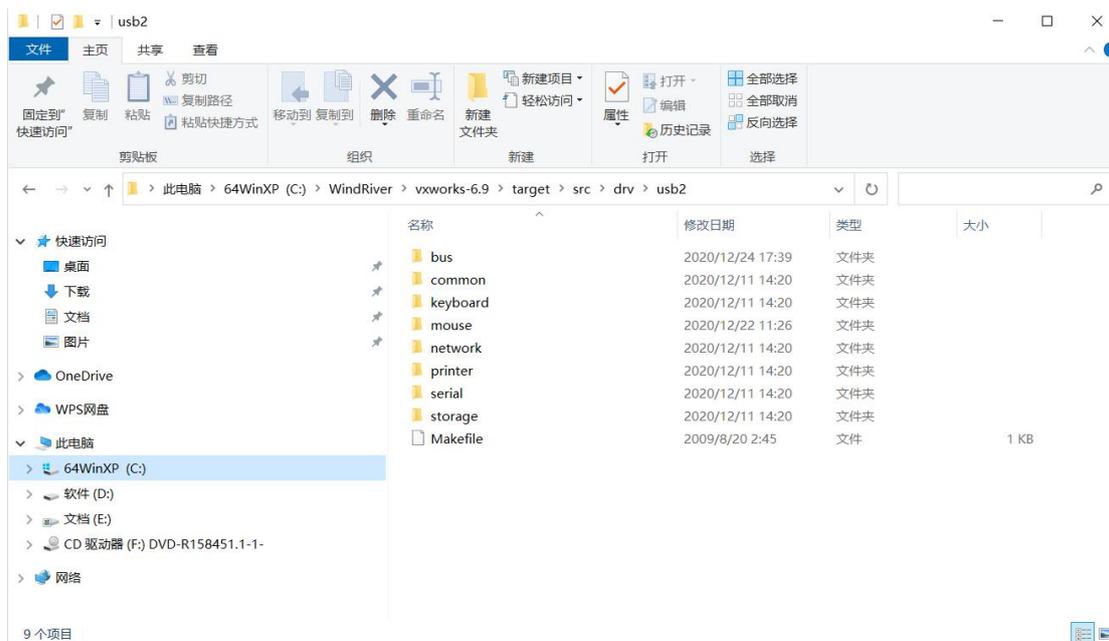
③打开示例软件，初始化板卡，并对通道 0 进行配置，添加发送数据，可按下图标示 1-4 步骤进行操作。通道 0 接收到的数据将在软件界面上显示，如下图：



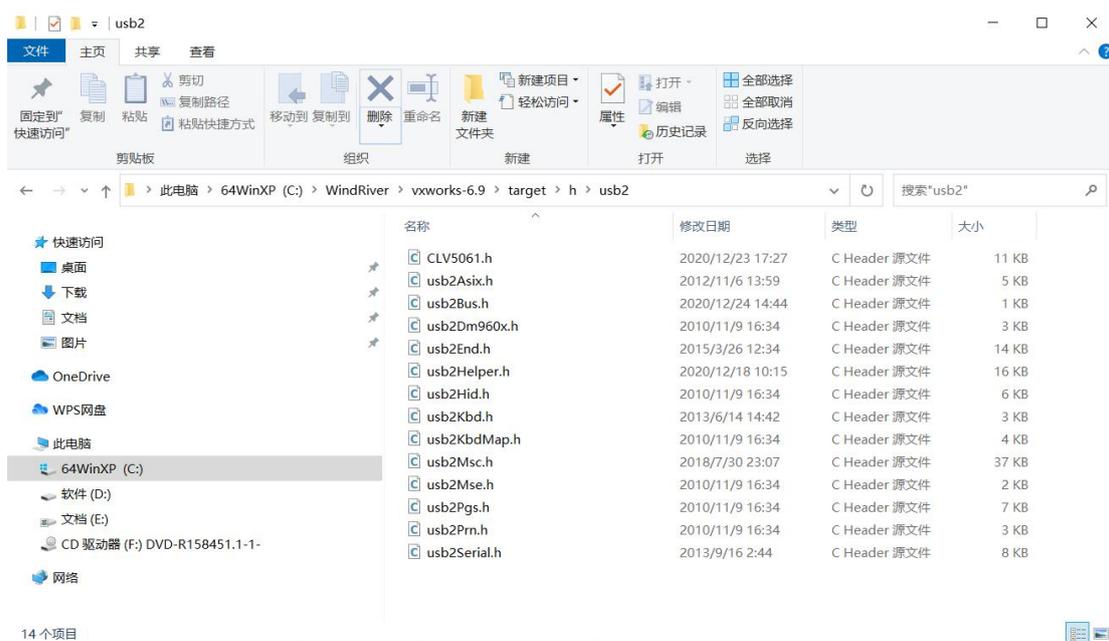
8. Wxworks 系统下安装及使用说明

8.1. 环境配置

- 复制 bus 文件夹至 vxworks-6.9\target\src\drv\usb2 目录下；如安装路径在默认路径，拷贝到 C:\WindRiver\vxworks-6.9\target\src\drv\usb2 目录下；

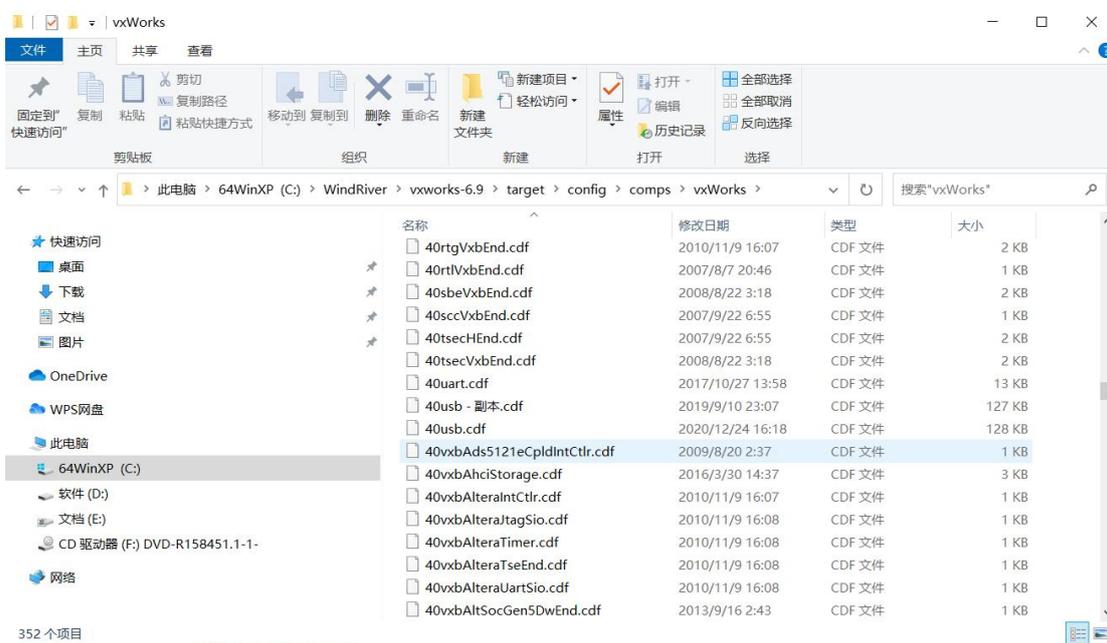


- 复制 Inc 目录下文件至 vxworks-6.9\target\h\usb2 路径下；

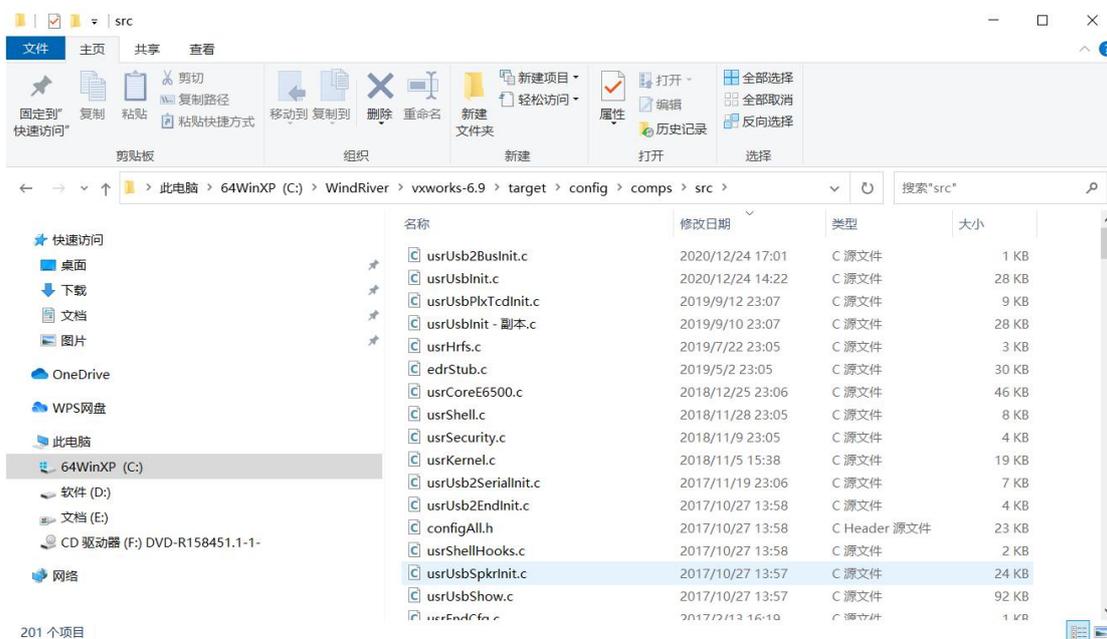


- 复制 CDF 文件夹下的文件至 vxworks-6.9\target\config\comps\vxWorks 目录

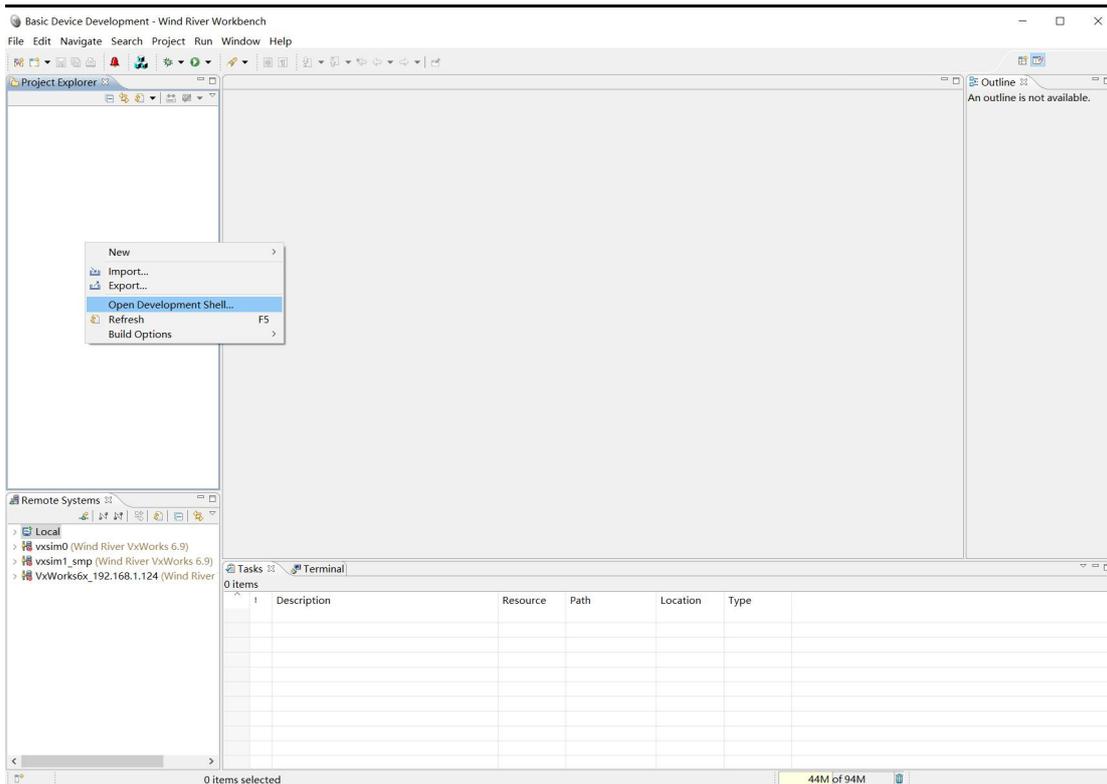
下(复制前最好先备份 vxworks-6.9\target\config\comps\vxWorks 目录下的 40usb.cdf 文件);



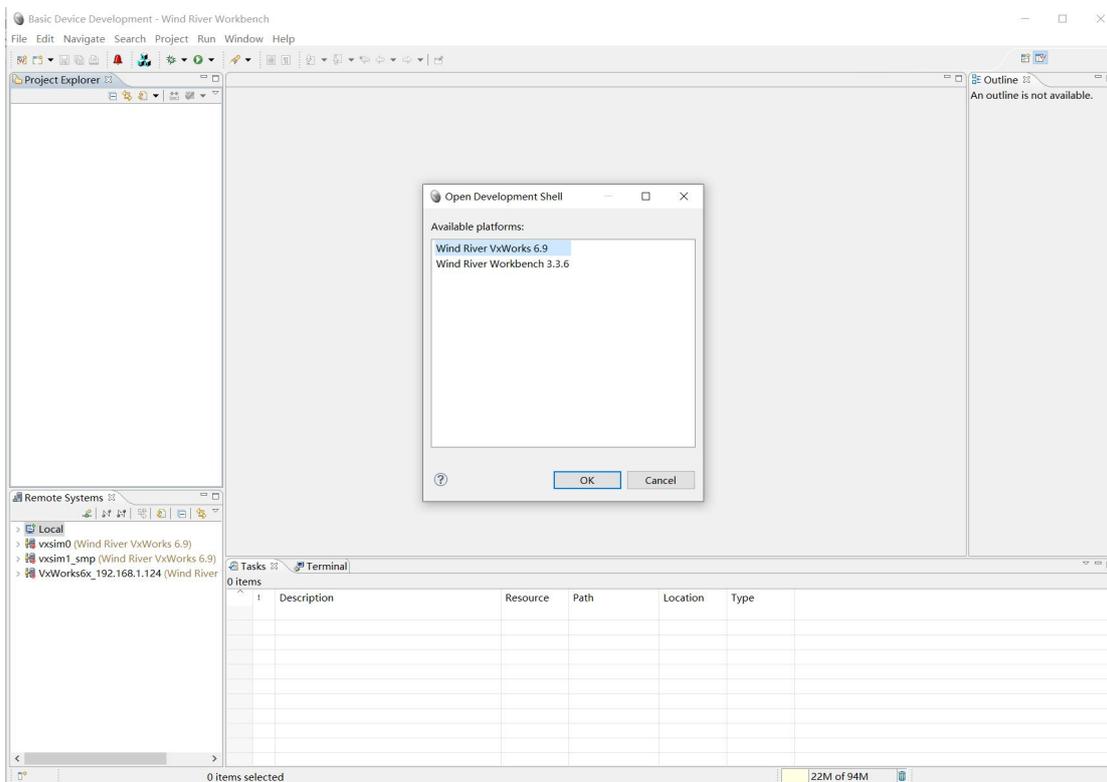
- 复制 conf 文件夹下至 vxworks-6.9\target\config\comps\src 目录下(复制前最好先备份 vxworks-6.9\target\config\comps\src 目录下 usrUsbInit.c 文件);



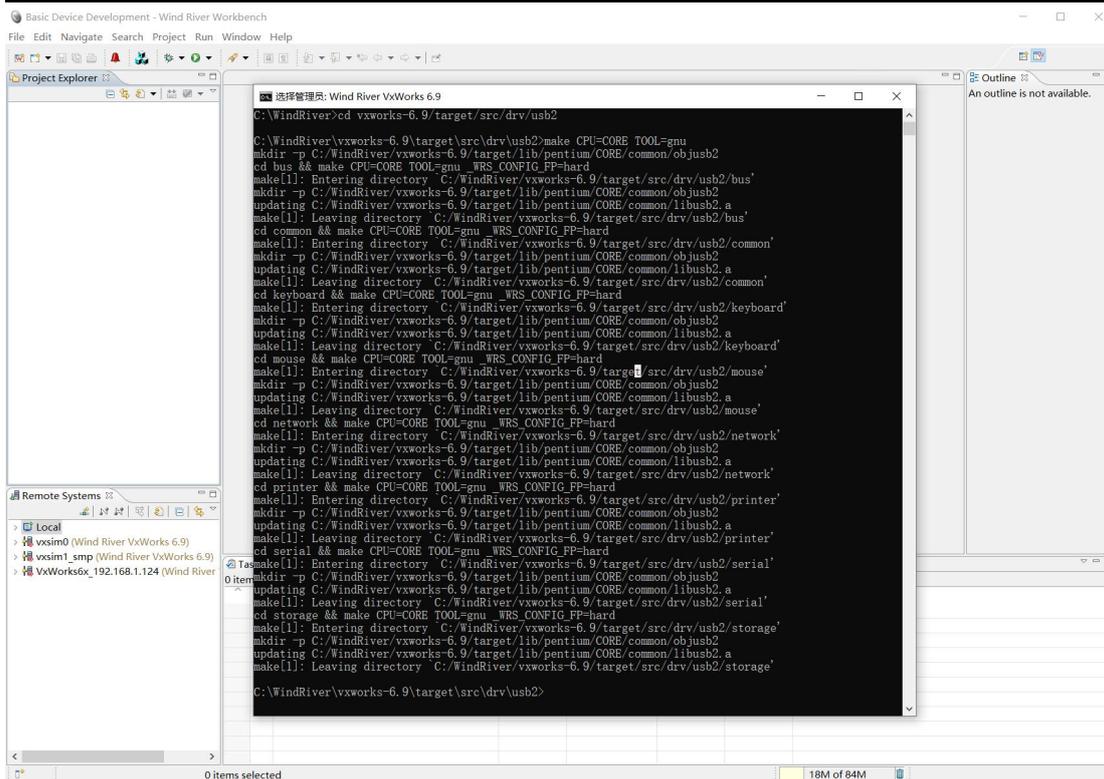
- 打开 Wind River Bench 软件，在工程栏中点击鼠标右键，选择 Open Development Shell;



- 选择 Wind River VxWorks 6.9;



- 在命令行下切换到 vxworks-6.9\target\src\drv\usb2 路径下执行编译;



- 编译命令: make CPU=XXX TOOL=XXX

注意:

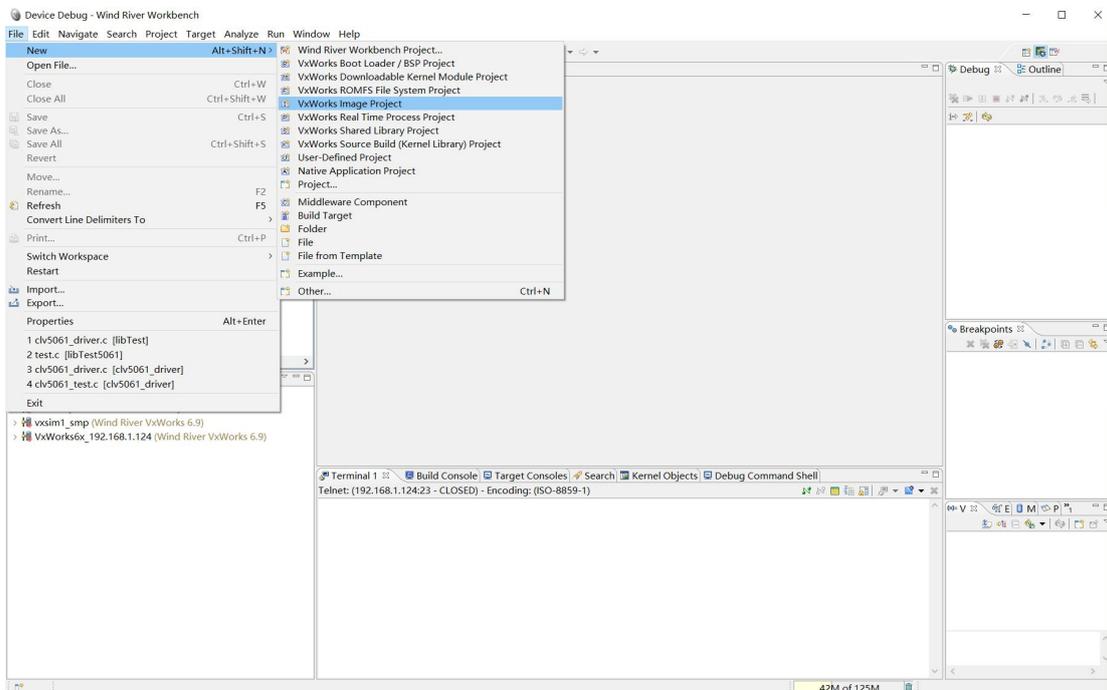
CPU : 取决于实际工作的平台

TOOL: 取决于采用的编译方式

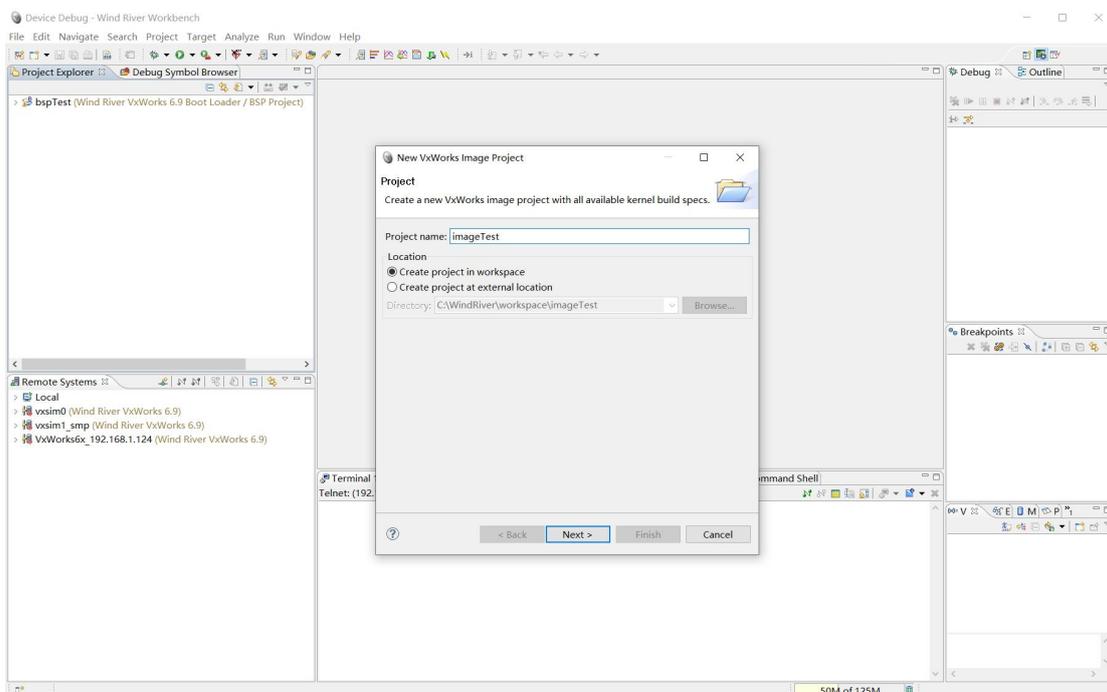
8.2. 组件配置

8.2.1. 新建 VIP 工程

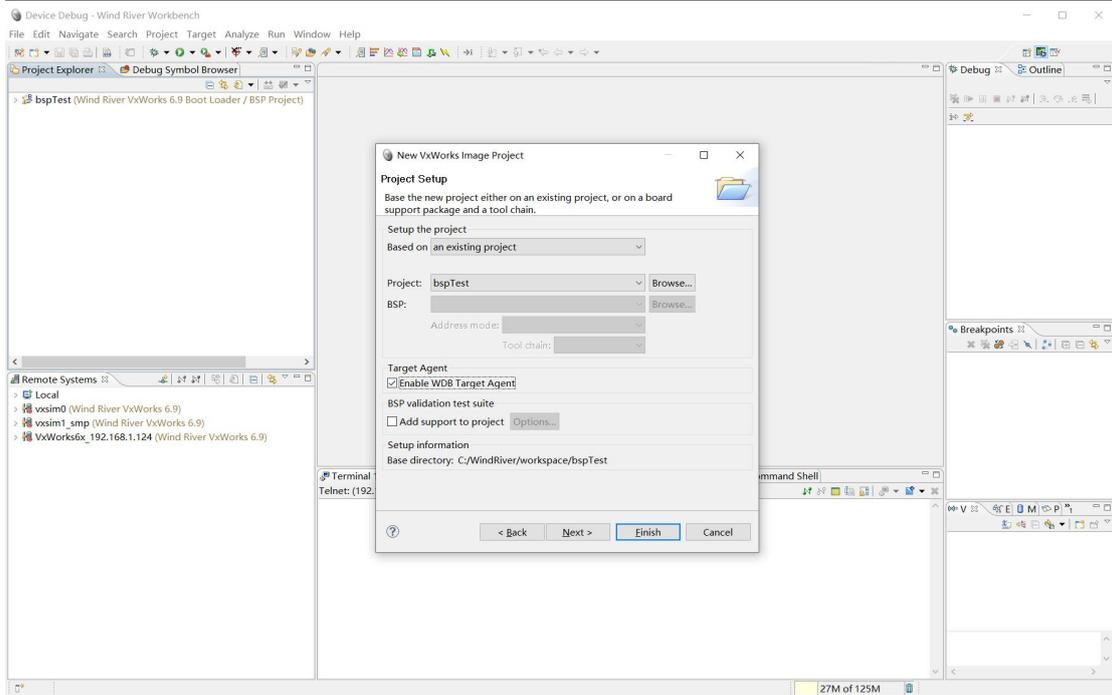
- 选择工程类型；



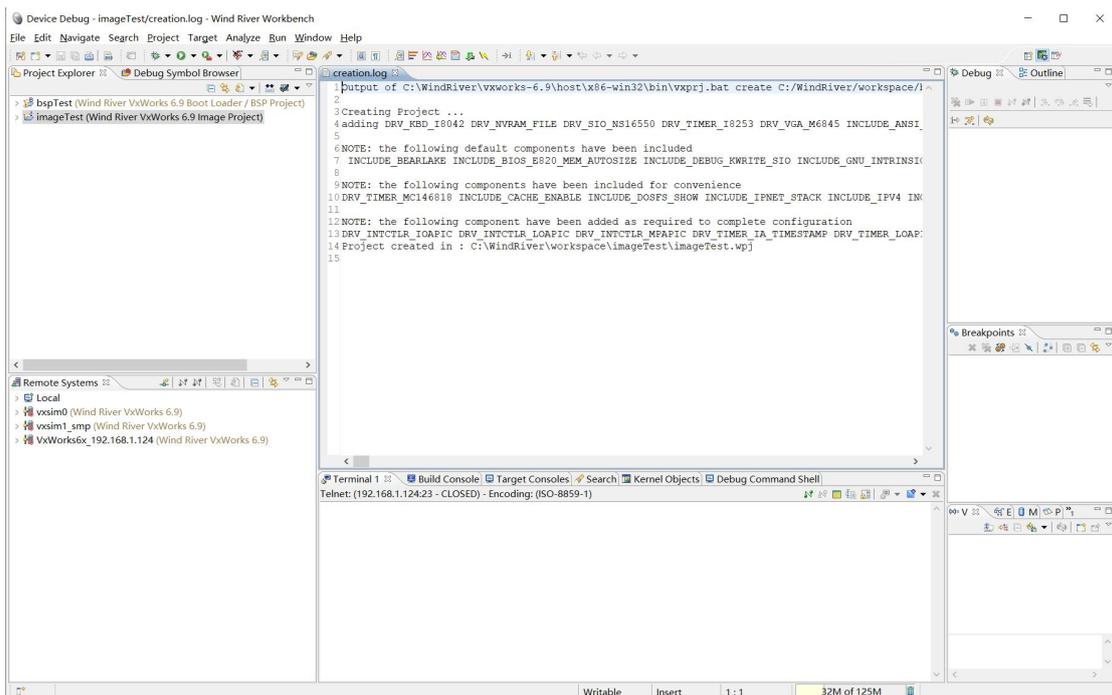
- 输入工程名；



- 选择依据 BSP 构建并选择 WDB 调试组件；



- 工程创建完成并正确读取配置项;



8.2.2. 模块功能配置

- 启用模块配置并编译;

INCLUDE_USB

INCLUDE_USB_INIT

INCLUDE_EHCI

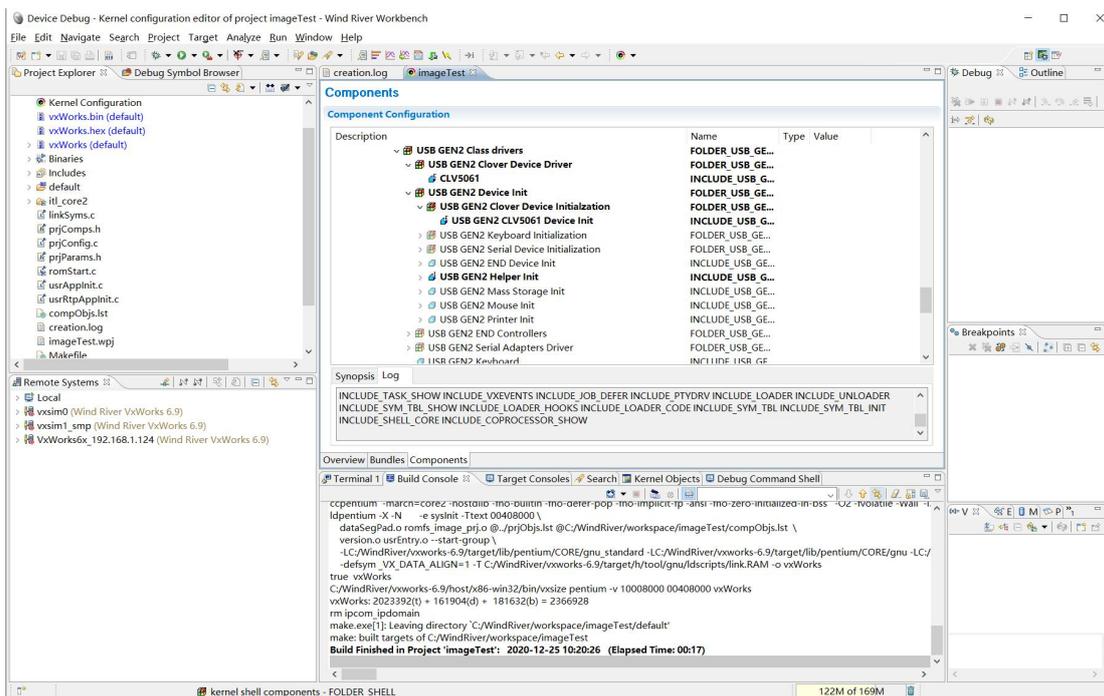
INCLUDE_EHCI_INIT

INCLUDE_HCD_BUS

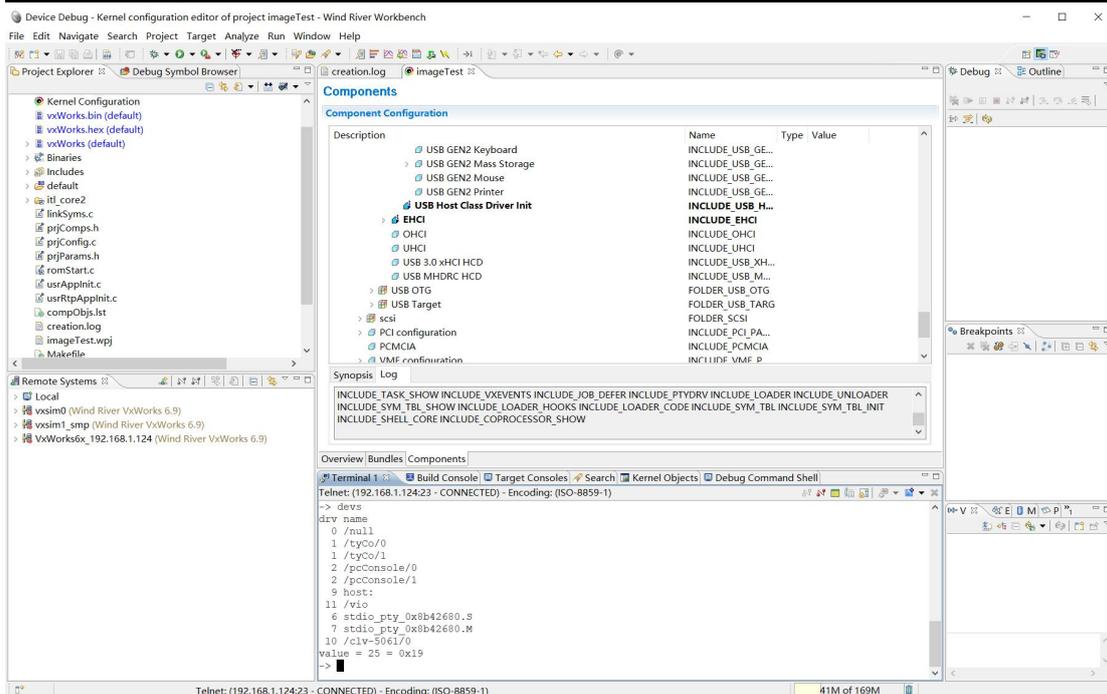
INCLUDE_USB_HOST_CLASS_INIT

INCLUDE_USB_GEN2_CLV-5061

INCLUDE_USB_GEN2_CLV-5061_INIT



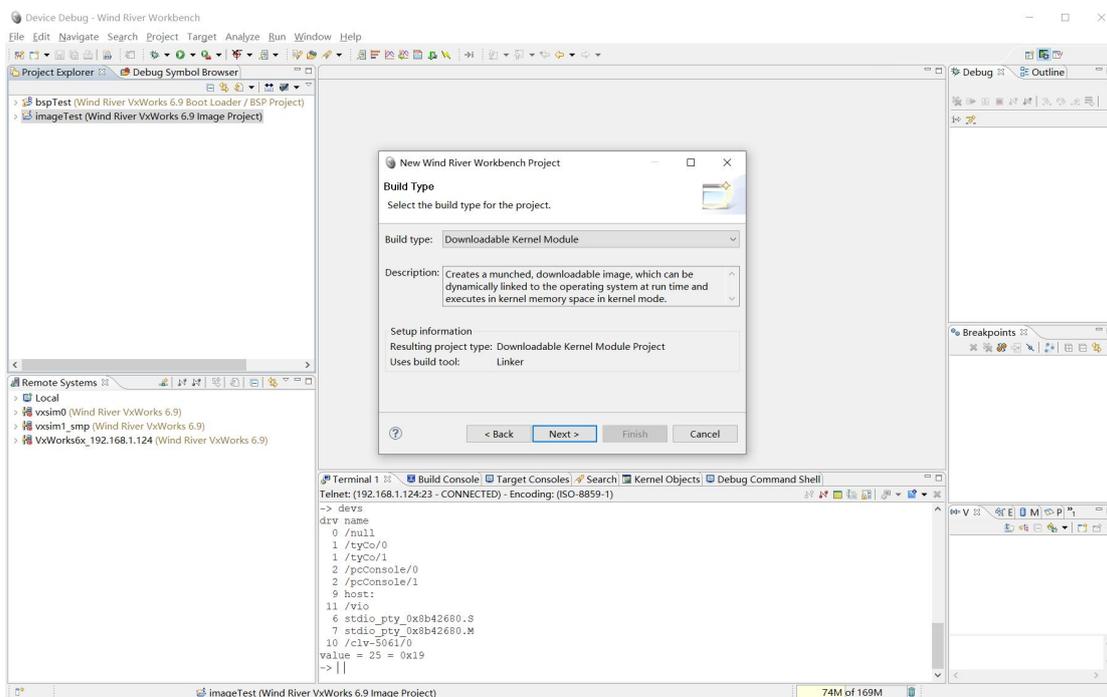
- 用生成的镜像引导设备启动后接通 CLV-5061 模块电源, 连接 USB; 输入 devsw 指令查看设备, 若操作无误会有: /clv-5061/0 设备存在;



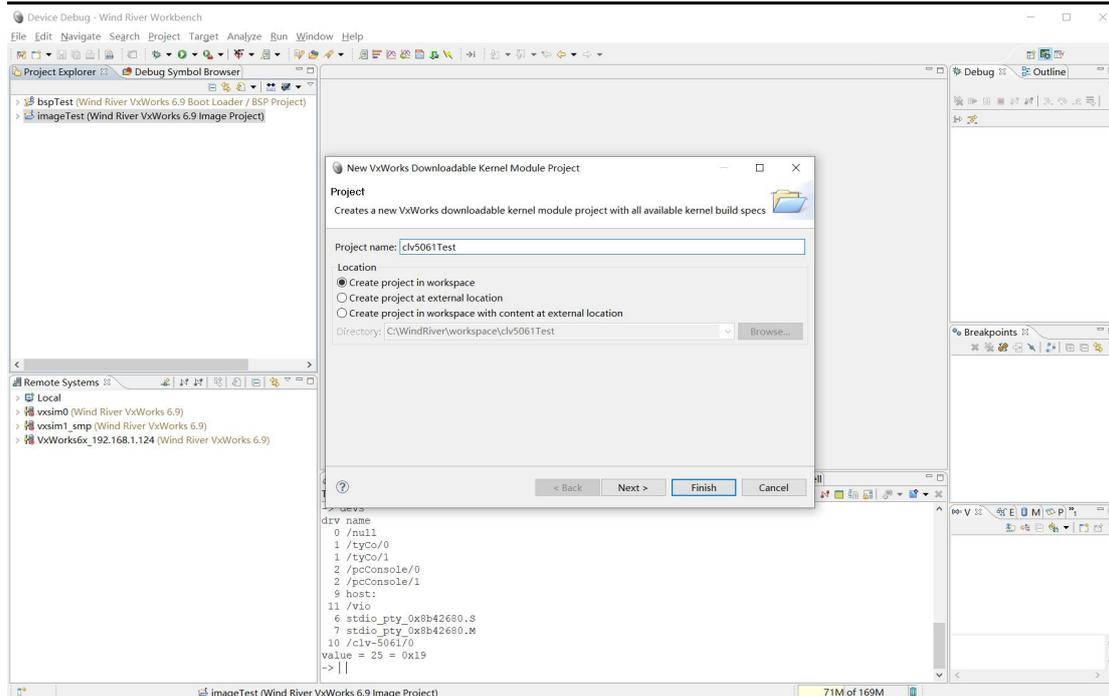
8.2.3. 总线功能测试

8.2.3.1. 新建 DKM 工程

- 选择工程类型;

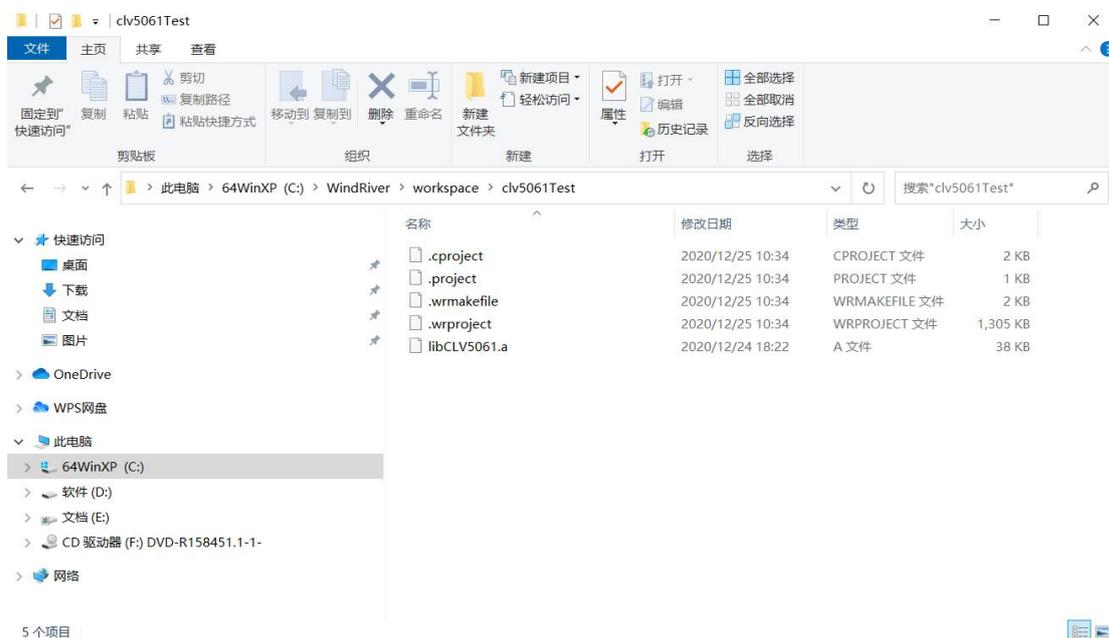


- 输入工程名;

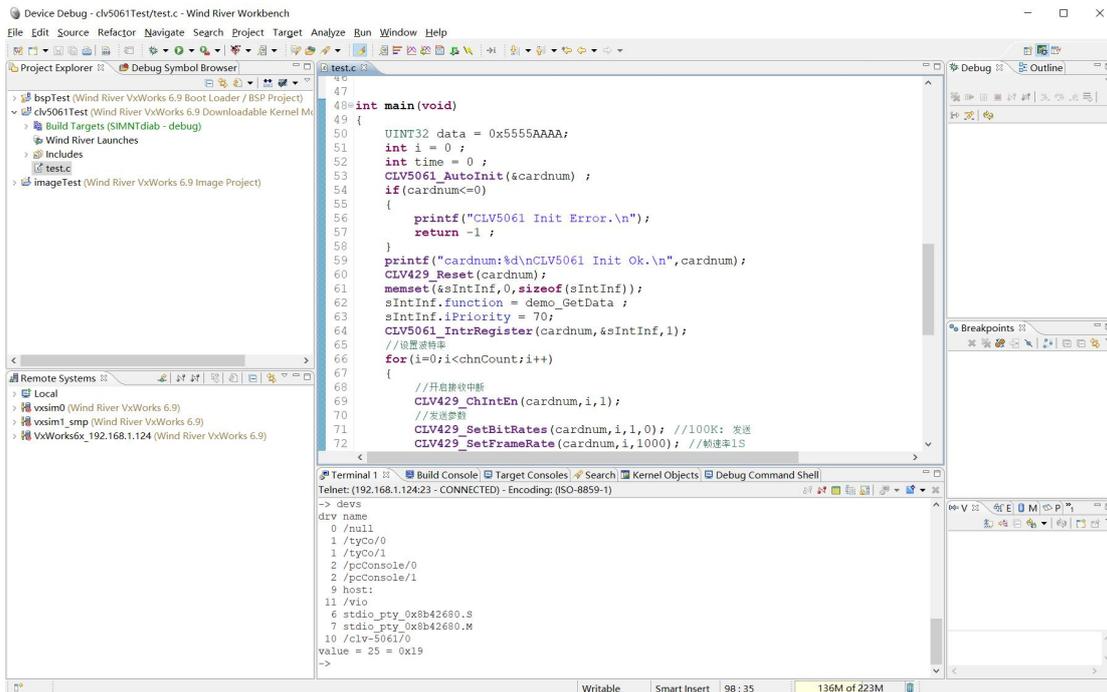


8.2.3.2. 工程配置

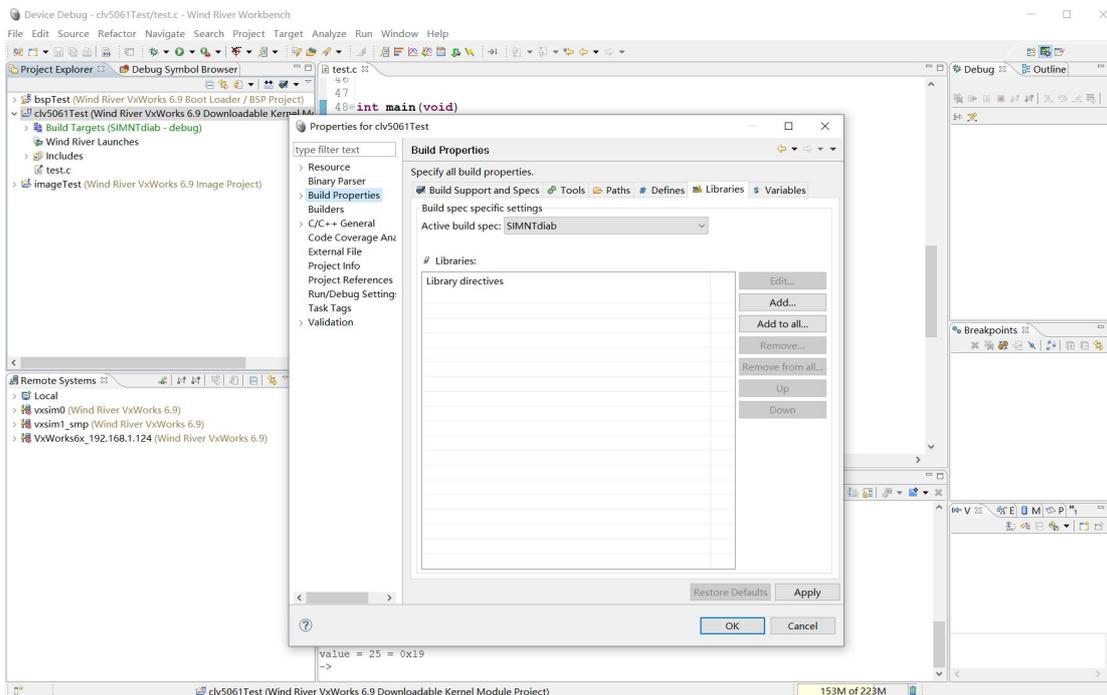
- 将 Lib 路径下的 LibCLV-5061.a 拷贝到工程目录下;



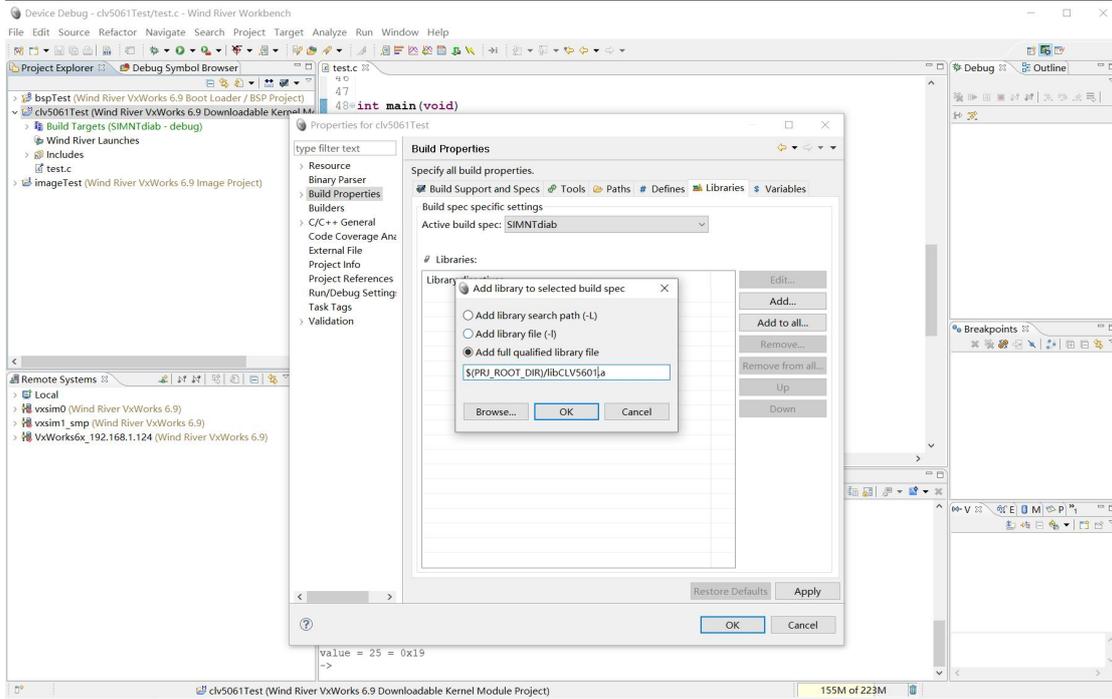
- 导入 Src 路径下的 test.c 文件到工程;



- 在工程名上点击鼠标右键，选择 Properties->Build Properties;

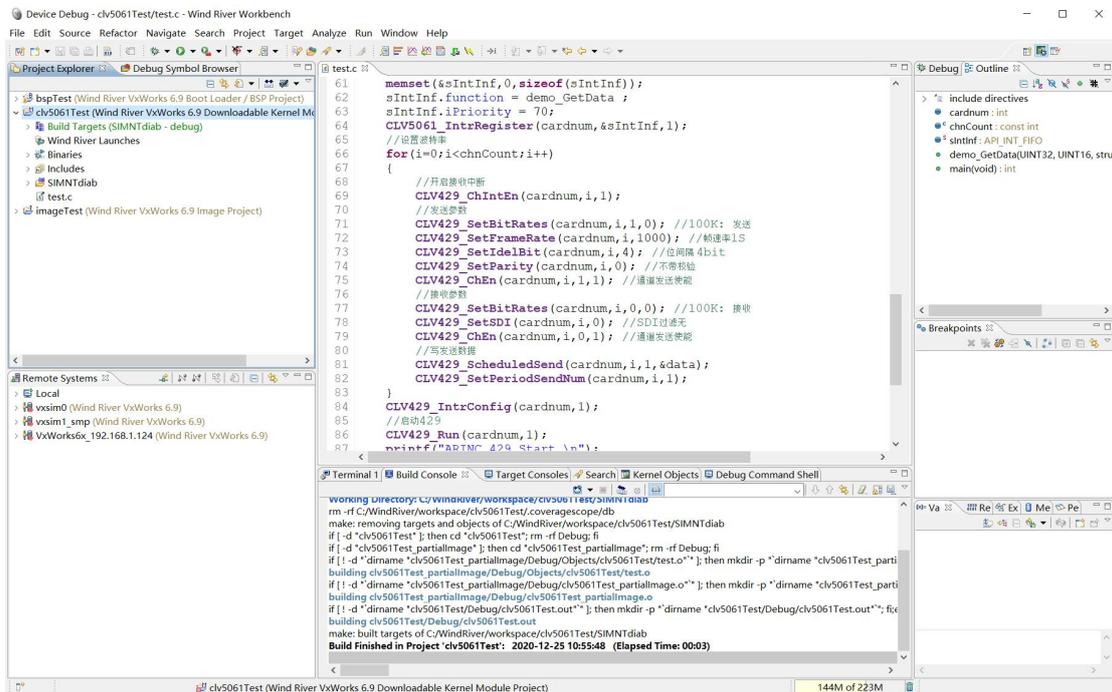


- 添加库路径\$(PRJ_ROOT_DIR)/libCLV5601.a; 点击 Apply 后再确认;

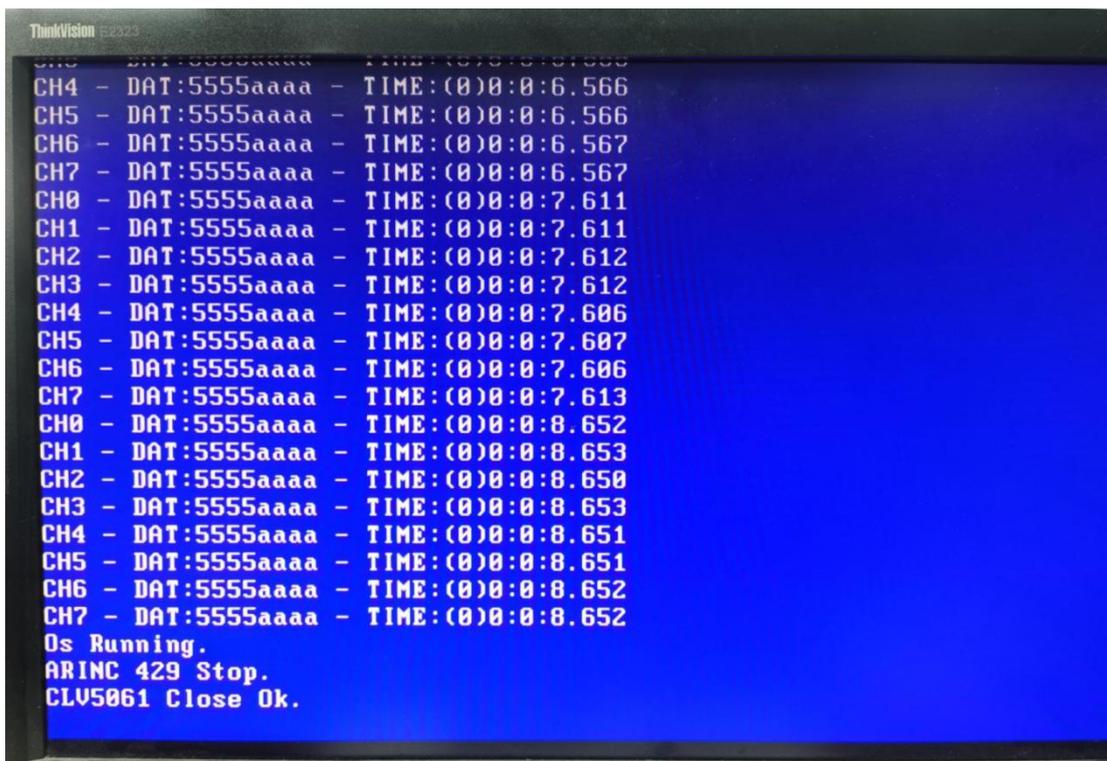
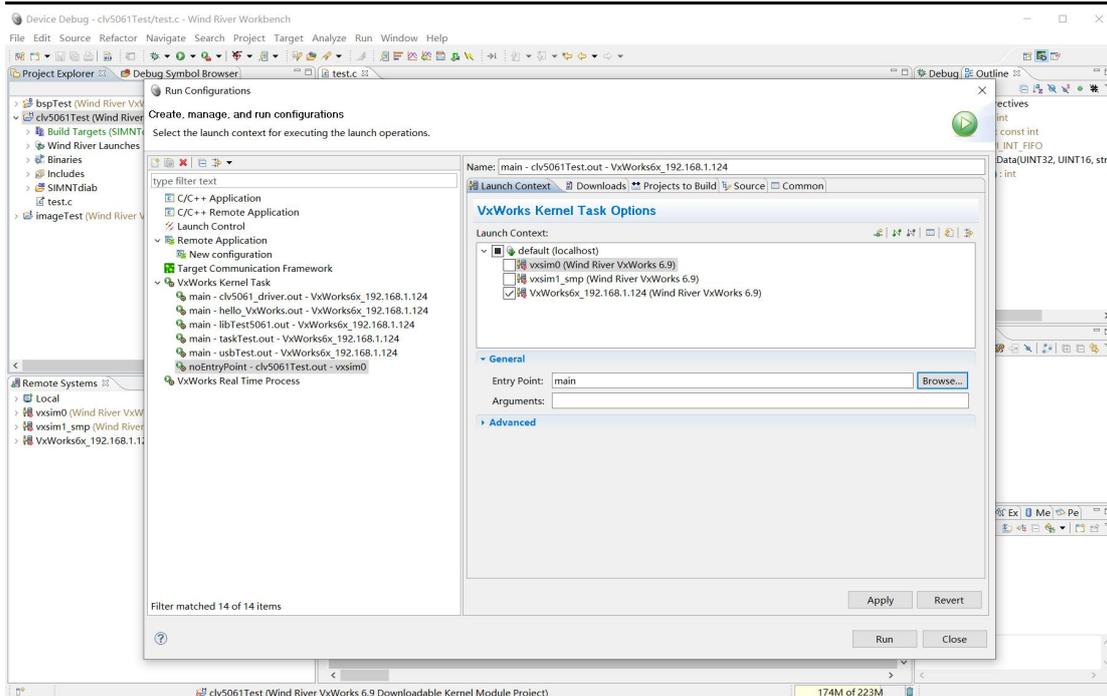


8.2.3.3. 编译运行

● 编译



● 下载运行



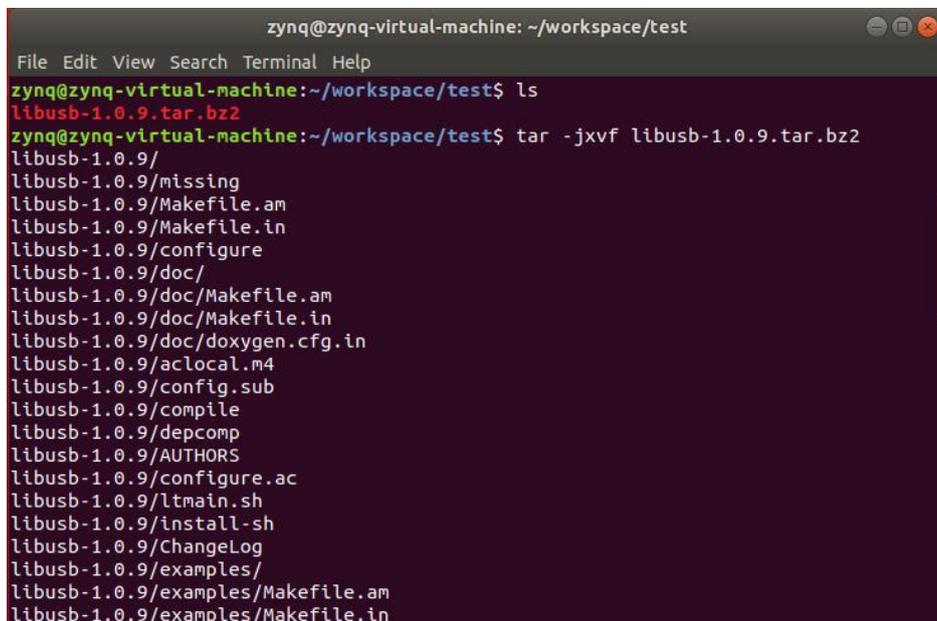
8.2.3.4. 其他说明

libCLV-5061 库默认采用的编译方式为 SIMNTdiab；若因客户采用的编译工具链不一致导致库不能正常编译或者运行，可联系本公司处理；

9. Linux 系统下安装及使用说明

9.1. Libusb 安装

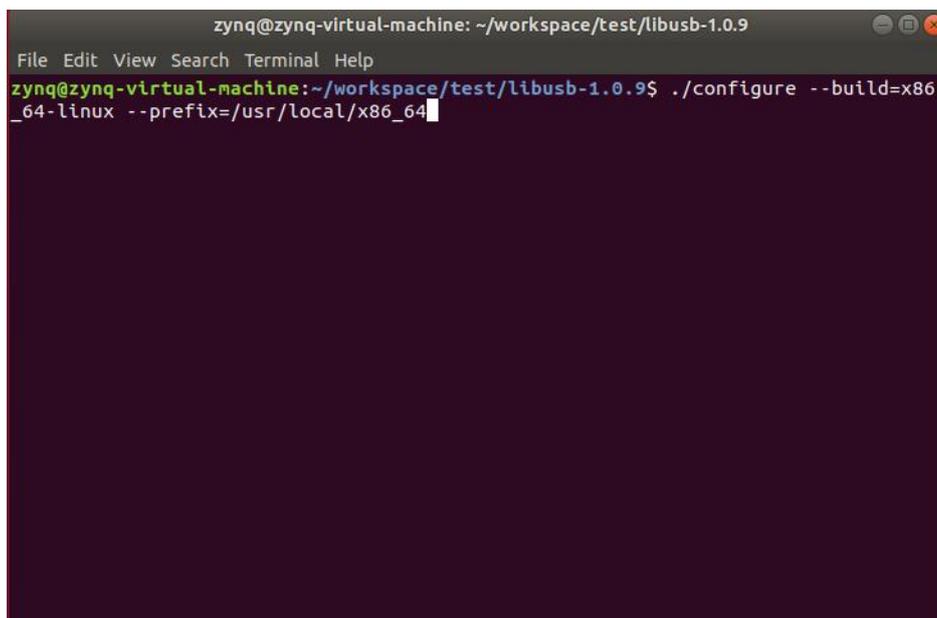
解压提供的 libusb-1.0.9.tar.bz2;



```
zynq@zynq-virtual-machine: ~/workspace/test
File Edit View Search Terminal Help
zynq@zynq-virtual-machine:~/workspace/test$ ls
libusb-1.0.9.tar.bz2
zynq@zynq-virtual-machine:~/workspace/test$ tar -jxvf libusb-1.0.9.tar.bz2
libusb-1.0.9/
libusb-1.0.9/missing
libusb-1.0.9/Makefile.am
libusb-1.0.9/Makefile.in
libusb-1.0.9/configure
libusb-1.0.9/doc/
libusb-1.0.9/doc/Makefile.am
libusb-1.0.9/doc/Makefile.in
libusb-1.0.9/doc/doxygen.cfg.in
libusb-1.0.9/aclocal.m4
libusb-1.0.9/config.sub
libusb-1.0.9/compile
libusb-1.0.9/depcomp
libusb-1.0.9/AUTHORS
libusb-1.0.9/configure.ac
libusb-1.0.9/ltmain.sh
libusb-1.0.9/install-sh
libusb-1.0.9/ChangeLog
libusb-1.0.9/examples/
libusb-1.0.9/examples/Makefile.am
libusb-1.0.9/examples/Makefile.in
```

9.1.1. 配置 Libusb 在 X86 系统下使用

执行./configure --build=x86_64-linux --prefix=/usr/local/x86_64



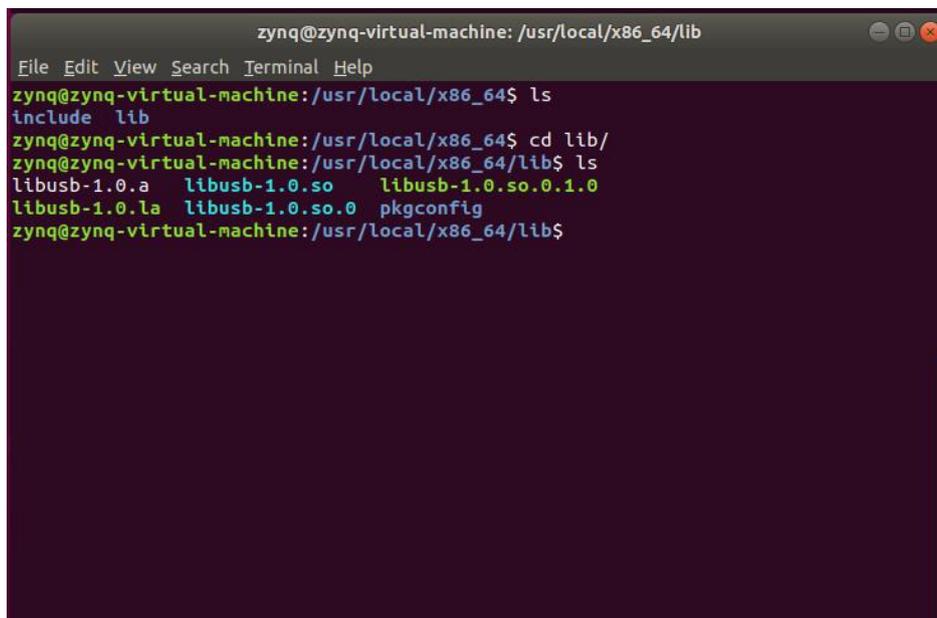
```
zynq@zynq-virtual-machine: ~/workspace/test/libusb-1.0.9
File Edit View Search Terminal Help
zynq@zynq-virtual-machine:~/workspace/test/libusb-1.0.9$ ./configure --build=x86_64-linux --prefix=/usr/local/x86_64
```

```
zynq@zynq-virtual-machine: ~/workspace/test/libusb-1.0.9
File Edit View Search Terminal Help
checking poll.h presence... yes
checking for poll.h... yes
checking sys/timerfd.h usability... yes
checking sys/timerfd.h presence... yes
checking for sys/timerfd.h... yes
checking whether TFD_NONBLOCK is declared... yes
checking whether to use timerfd for timing... yes
checking for struct timespec... yes
checking for sigaction... yes
checking sys/time.h usability... yes
checking sys/time.h presence... yes
checking for sys/time.h... yes
checking for gettimeofday... yes
configure: creating ./config.status
config.status: creating libusb-1.0.pc
config.status: creating Makefile
config.status: creating libusb/Makefile
config.status: creating examples/Makefile
config.status: creating doc/Makefile
config.status: creating doc/doxygen.cfg
config.status: creating config.h
config.status: executing depfiles commands
config.status: executing libtool commands
zynq@zynq-virtual-machine:~/workspace/test/libusb-1.0.9$
```

执行 make install;

```
zynq@zynq-virtual-machine: ~/workspace/test/libusb-1.0.9
File Edit View Search Terminal Help
zynq@zynq-virtual-machine:~/workspace/test/libusb-1.0.9$ sudo make install
[sudo] password for zynq:
Making install in libusb
make[1]: Entering directory '/home/zynq/workspace/test/libusb-1.0.9/libusb'
make[2]: Entering directory '/home/zynq/workspace/test/libusb-1.0.9/libusb'
test -z "/usr/local/x86_64/lib" || /bin/mkdir -p "/usr/local/x86_64/lib"
/bin/bash ../libtool --mode=install /usr/bin/install -c libusb-1.0.la '/usr/local/x86_64/lib'
libtool: install: /usr/bin/install -c .libs/libusb-1.0.so.0.1.0 /usr/local/x86_64/lib/libusb-1.0.so.0.1.0
libtool: install: (cd /usr/local/x86_64/lib && { ln -s -f libusb-1.0.so.0.1.0 libusb-1.0.so.0 || { rm -f libusb-1.0.so.0 && ln -s libusb-1.0.so.0.1.0 libusb-1.0.so.0; }; })
libtool: install: (cd /usr/local/x86_64/lib && { ln -s -f libusb-1.0.so.0.1.0 libusb-1.0.so || { rm -f libusb-1.0.so && ln -s libusb-1.0.so.0.1.0 libusb-1.0.so; }; })
libtool: install: /usr/bin/install -c .libs/libusb-1.0.lai /usr/local/x86_64/lib/libusb-1.0.la
libtool: install: /usr/bin/install -c .libs/libusb-1.0.a /usr/local/x86_64/lib/libusb-1.0.a
libtool: install: chmod 644 /usr/local/x86_64/lib/libusb-1.0.a
libtool: install: ranlib /usr/local/x86_64/lib/libusb-1.0.a
libtool: finish: PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin:/sbin" ldconfig -n /usr/local/x86_64/lib
```

在 `usr/local/x86_64` 目录下存放着相关文件;



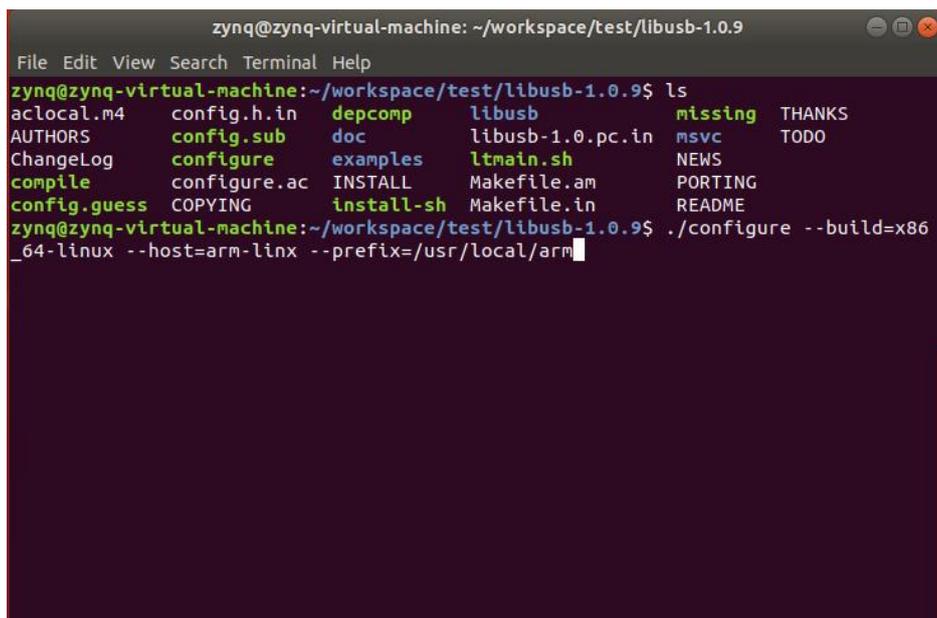
```
zynq@zynq-virtual-machine: /usr/local/x86_64/lib
File Edit View Search Terminal Help
zynq@zynq-virtual-machine: /usr/local/x86_64$ ls
include lib
zynq@zynq-virtual-machine: /usr/local/x86_64$ cd lib/
zynq@zynq-virtual-machine: /usr/local/x86_64/lib$ ls
libusb-1.0.a  libusb-1.0.so  libusb-1.0.so.0.1.0
libusb-1.0.la  libusb-1.0.so.0  pkgconfig
zynq@zynq-virtual-machine: /usr/local/x86_64/lib$
```

9.1.2. 配置 Libusb 在 ARM 系统下使用

执行一下命令:

```
./configure --build=x86_64-linux --host=arm-linux --prefix=/usr/local/arm
```

CC=XXX(交差编译链)



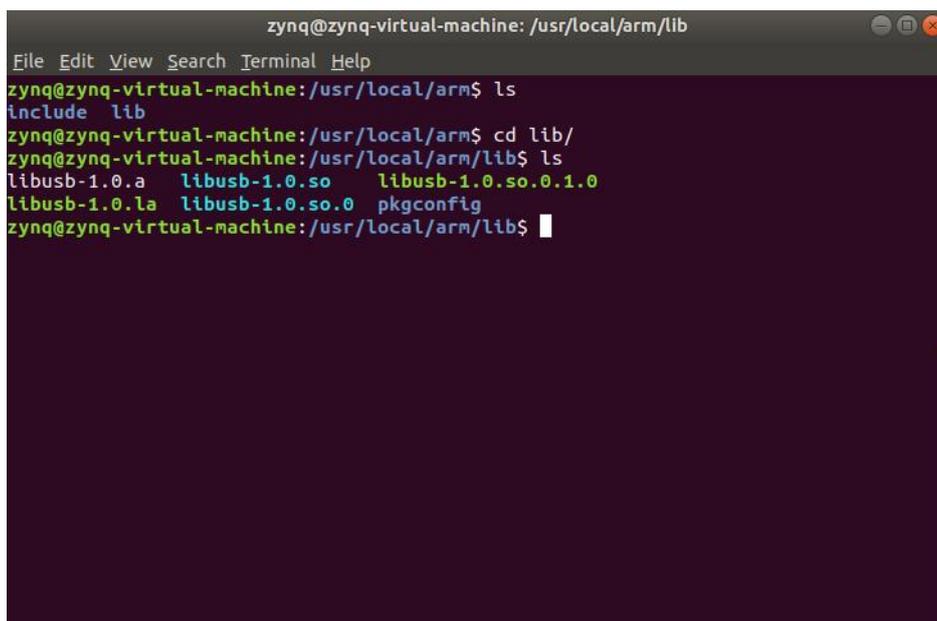
```
zynq@zynq-virtual-machine: ~/workspace/test/libusb-1.0.9
File Edit View Search Terminal Help
zynq@zynq-virtual-machine:~/workspace/test/libusb-1.0.9$ ls
aclocal.m4  config.h.in  depcomp  libusb  missing  THANKS
AUTHORS    config.sub  doc      libusb-1.0.pc.in  msvc     TODO
ChangeLog  configure  examples  ltmain.sh  NEWS
compile    configure.ac  INSTALL  Makefile.am  PORTING
config.guess  COPYING    install-sh  Makefile.in  README
zynq@zynq-virtual-machine:~/workspace/test/libusb-1.0.9$ ./configure --build=x86
_64-linux --host=arm-linx --prefix=/usr/local/arm
```

```
zynq@zynq-virtual-machine: ~/workspace/test/libusb-1.0.9
File Edit View Search Terminal Help
checking poll.h presence... yes
checking for poll.h... yes
checking sys/timerfd.h usability... yes
checking sys/timerfd.h presence... yes
checking for sys/timerfd.h... yes
checking whether TFD_NONBLOCK is declared... yes
checking whether to use timerfd for timing... yes
checking for struct timespec... yes
checking for sigaction... yes
checking sys/time.h usability... yes
checking sys/time.h presence... yes
checking for sys/time.h... yes
checking for gettimeofday... yes
configure: creating ./config.status
config.status: creating libusb-1.0.pc
config.status: creating Makefile
config.status: creating libusb/Makefile
config.status: creating examples/Makefile
config.status: creating doc/Makefile
config.status: creating doc/doxygen.cfg
config.status: creating config.h
config.status: executing depfiles commands
config.status: executing libtool commands
zynq@zynq-virtual-machine:~/workspace/test/libusb-1.0.9$
```

执行 make install;

```
zynq@zynq-virtual-machine: ~/workspace/test/libusb-1.0.9
File Edit View Search Terminal Help
zynq@zynq-virtual-machine:~/workspace/test/libusb-1.0.9$ sudo make install
[sudo] password for zynq:
Making install in libusb
make[1]: Entering directory '/home/zynq/workspace/test/libusb-1.0.9/libusb'
make[2]: Entering directory '/home/zynq/workspace/test/libusb-1.0.9/libusb'
test -z "/usr/local/x86_64/lib" || /bin/mkdir -p "/usr/local/x86_64/lib"
/bin/bash ../libtool --mode=install /usr/bin/install -c libusb-1.0.la '/usr/local/x86_64/lib'
libtool: install: /usr/bin/install -c .libs/libusb-1.0.so.0.1.0 /usr/local/x86_64/lib/libusb-1.0.so.0.1.0
libtool: install: (cd /usr/local/x86_64/lib && { ln -s -f libusb-1.0.so.0.1.0 libusb-1.0.so.0 || { rm -f libusb-1.0.so.0 && ln -s libusb-1.0.so.0.1.0 libusb-1.0.so.0; }; })
libtool: install: (cd /usr/local/x86_64/lib && { ln -s -f libusb-1.0.so.0.1.0 libusb-1.0.so || { rm -f libusb-1.0.so && ln -s libusb-1.0.so.0.1.0 libusb-1.0.so; }; })
libtool: install: /usr/bin/install -c .libs/libusb-1.0.lai /usr/local/x86_64/lib/libusb-1.0.la
libtool: install: /usr/bin/install -c .libs/libusb-1.0.a /usr/local/x86_64/lib/libusb-1.0.a
libtool: install: chmod 644 /usr/local/x86_64/lib/libusb-1.0.a
libtool: install: ranlib /usr/local/x86_64/lib/libusb-1.0.a
libtool: finish: PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin:/sbin" ldconfig -n /usr/local/x86_64/lib
```

在 `usr/local/arm` 目录下存放着相关文件;

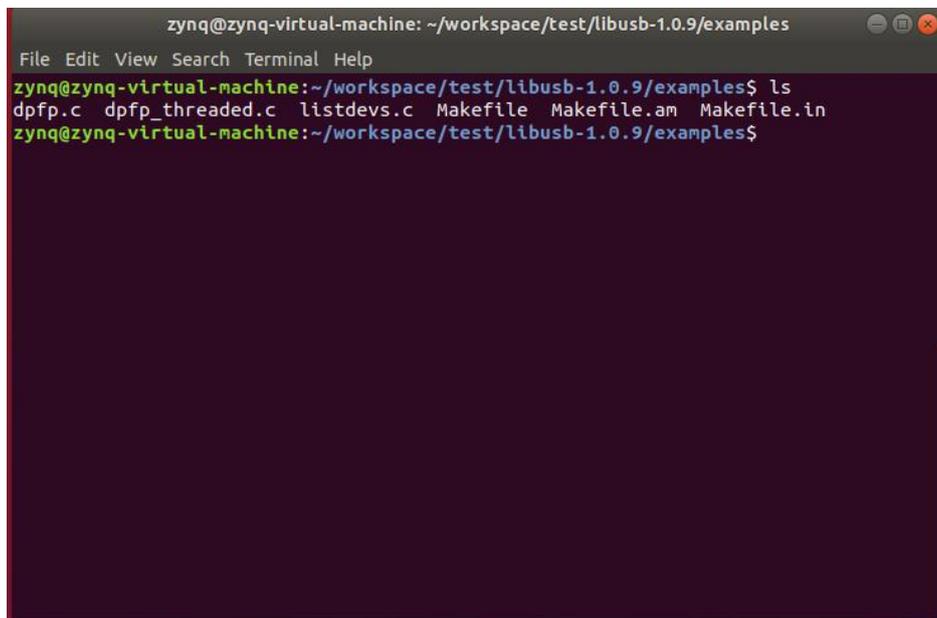


```
zynq@zynq-virtual-machine: /usr/local/arm/lib
File Edit View Search Terminal Help
zynq@zynq-virtual-machine: /usr/local/arm$ ls
include lib
zynq@zynq-virtual-machine: /usr/local/arm$ cd lib/
zynq@zynq-virtual-machine: /usr/local/arm/lib$ ls
libusb-1.0.a  libusb-1.0.so  libusb-1.0.so.0.1.0
libusb-1.0.la  libusb-1.0.so.0  pkgconfig
zynq@zynq-virtual-machine: /usr/local/arm/lib$
```

9.2. Libusb 基本测试

9.2.1. X86 系统下测试

进入到`**/libusb-1.0.9/examples`路径下;



```
zynq@zynq-virtual-machine: ~/workspace/test/libusb-1.0.9/examples
File Edit View Search Terminal Help
zynq@zynq-virtual-machine: ~/workspace/test/libusb-1.0.9/examples$ ls
dpfp.c  dpfp_threaded.c  listdevs.c  Makefile  Makefile.am  Makefile.in
zynq@zynq-virtual-machine: ~/workspace/test/libusb-1.0.9/examples$
```

执行 make 编译程序;

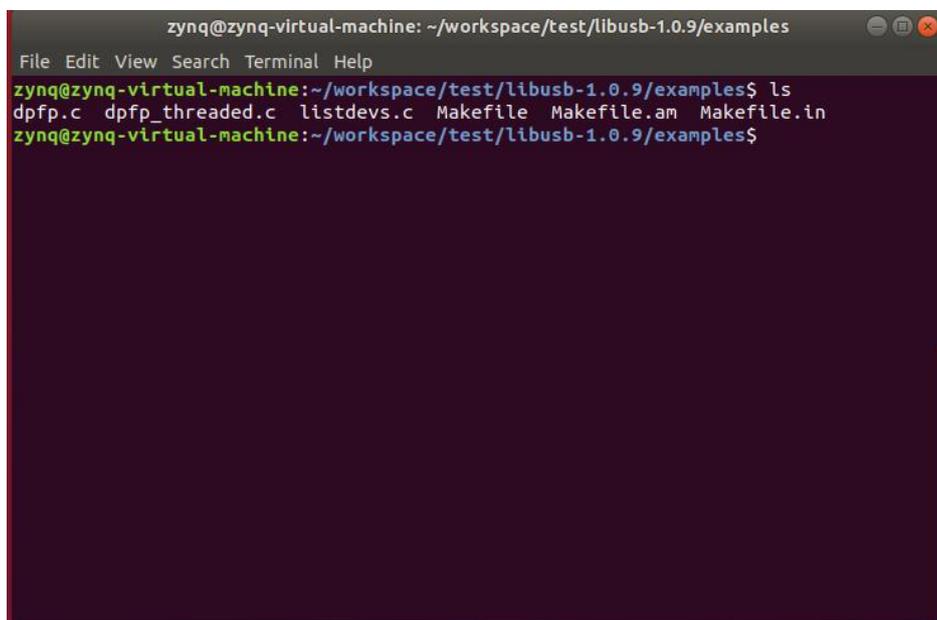
```
zynq@zynq-virtual-machine: ~/workspace/test/libusb-1.0.9/examples
File Edit View Search Terminal Help
zynq@zynq-virtual-machine:~/workspace/test/libusb-1.0.9/examples$ make
CC      listdevs.o
CCLD    listdevs
CC      dpfp.o
CCLD    dpfp
CC      dpfp_threaded-dpfp_threaded.o
CCLD    dpfp_threaded
zynq@zynq-virtual-machine:~/workspace/test/libusb-1.0.9/examples$ ls
dpfp  dpfp_threaded  listdevs  Makefile
dpfp.c  dpfp_threaded.c  listdevs.c  Makefile.am
dpfp.o  dpfp_threaded-dpfp_threaded.o  listdevs.o  Makefile.in
zynq@zynq-virtual-machine:~/workspace/test/libusb-1.0.9/examples$
```

运行 listdevs 测试是否正常;

```
zynq@zynq-virtual-machine: ~/workspace/test/libusb-1.0.9/examples
File Edit View Search Terminal Help
zynq@zynq-virtual-machine:~/workspace/test/libusb-1.0.9/examples$ make
CC      listdevs.o
CCLD    listdevs
CC      dpfp.o
CCLD    dpfp
CC      dpfp_threaded-dpfp_threaded.o
CCLD    dpfp_threaded
zynq@zynq-virtual-machine:~/workspace/test/libusb-1.0.9/examples$ ls
dpfp  dpfp_threaded  listdevs  Makefile
dpfp.c  dpfp_threaded.c  listdevs.c  Makefile.am
dpfp.o  dpfp_threaded-dpfp_threaded.o  listdevs.o  Makefile.in
zynq@zynq-virtual-machine:~/workspace/test/libusb-1.0.9/examples$ ./listdevs
zynq@zynq-virtual-machine:~/workspace/test/libusb-1.0.9/examples$ ./listdevs
0e0f:0003 (bus 2, device 2)
1d6b:0002 (bus 1, device 1)
0e0f:0002 (bus 2, device 3)
1d6b:0001 (bus 2, device 1)
zynq@zynq-virtual-machine:~/workspace/test/libusb-1.0.9/examples$
```

9.2.2. ARM 系统下基本测试

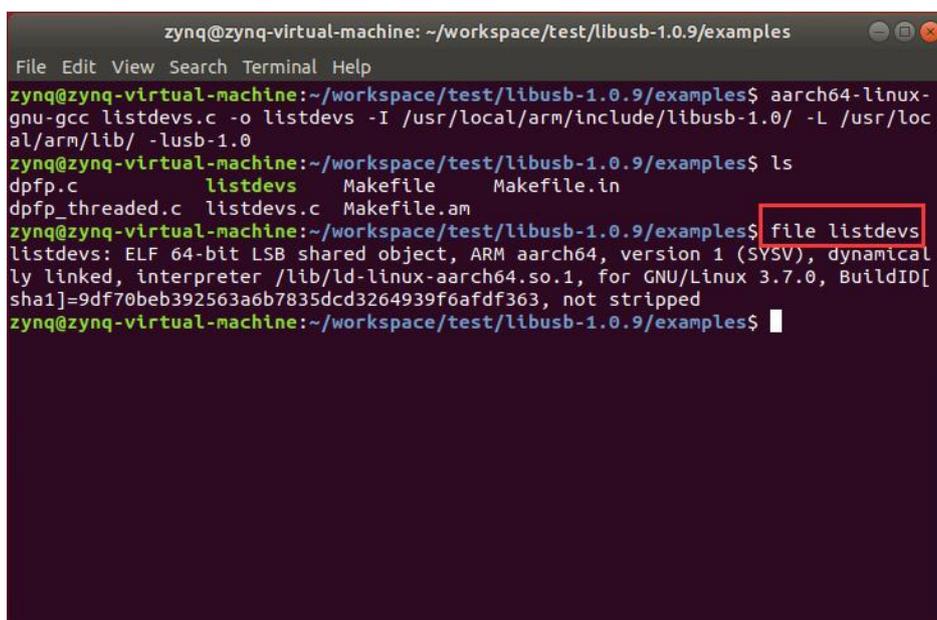
进入到**/libusb-1.0.9/examples 路径下;



```
zynq@zynq-virtual-machine: ~/workspace/test/libusb-1.0.9/examples
File Edit View Search Terminal Help
zynq@zynq-virtual-machine:~/workspace/test/libusb-1.0.9/examples$ ls
dpfp.c dpfp_threaded.c listdevs.c Makefile Makefile.am Makefile.in
zynq@zynq-virtual-machine:~/workspace/test/libusb-1.0.9/examples$
```

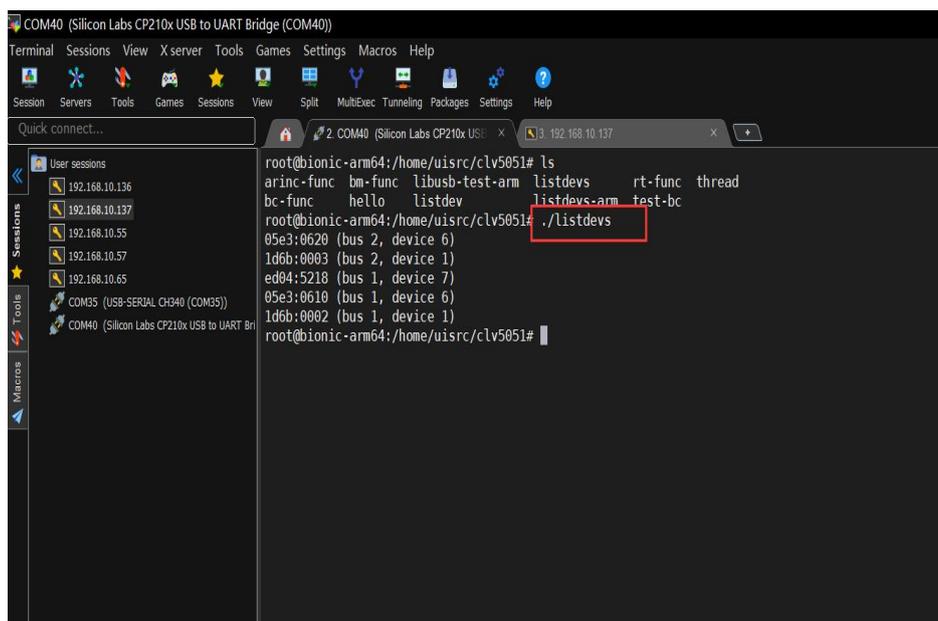
执行交叉编译，然后查看文件类型是否为 ARM 环境下使用;

```
aarch64-linux-gnu-gcc listdevs.c -o listdevs -I
/usr/local/arm/include/libusb-1.0/ -L /usr/local/arm/lib/ -lusb-1.0
```



```
zynq@zynq-virtual-machine: ~/workspace/test/libusb-1.0.9/examples
File Edit View Search Terminal Help
zynq@zynq-virtual-machine:~/workspace/test/libusb-1.0.9/examples$ aarch64-linux-
gnu-gcc listdevs.c -o listdevs -I /usr/local/arm/include/libusb-1.0/ -L /usr/loc
al/arm/lib/ -lusb-1.0
zynq@zynq-virtual-machine:~/workspace/test/libusb-1.0.9/examples$ ls
dpfp.c listdevs Makefile Makefile.in
dpfp_threaded.c listdevs.c Makefile.am
zynq@zynq-virtual-machine:~/workspace/test/libusb-1.0.9/examples$ file listdevs
listdevs: ELF 64-bit LSB shared object, ARM aarch64, version 1 (SYSV), dynamical
ly linked, interpreter /lib/ld-linux-aarch64.so.1, for GNU/Linux 3.7.0, BuildID[
sha1]=9df70beb392563a6b7835dcd3264939f6afdf363, not stripped
zynq@zynq-virtual-machine:~/workspace/test/libusb-1.0.9/examples$
```

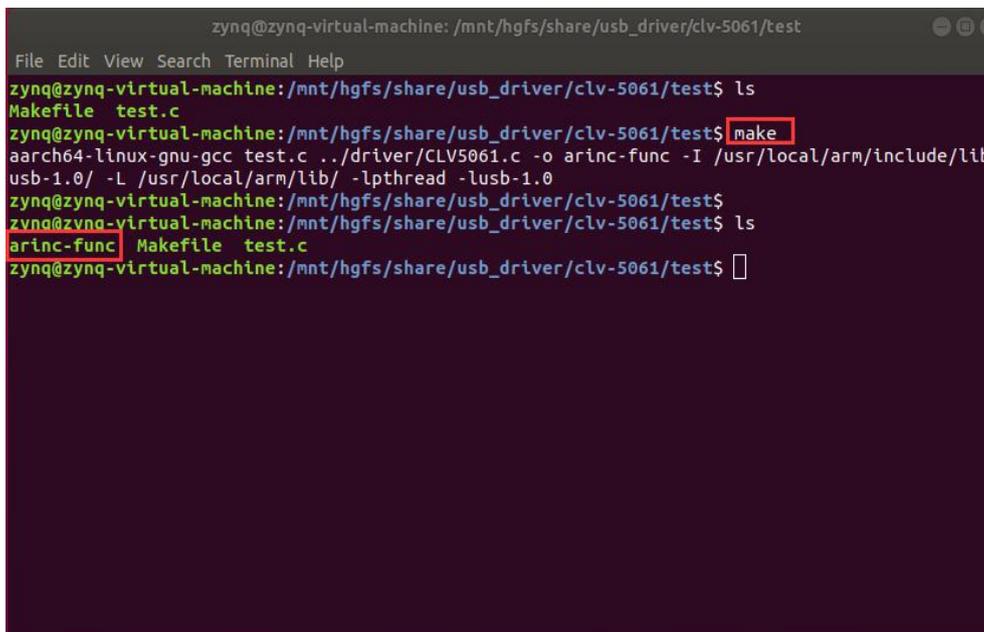
将 listdevs 可执行文件下载到目标板上运行测试；



```
COM40 (Silicon Labs CP210x USB to UART Bridge (COM40))
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
User sessions
192.168.10.136
192.168.10.137
192.168.10.55
192.168.10.57
192.168.10.65
Tools
COM35 (USB-SERIAL CH340 (COM35))
COM40 (Silicon Labs CP210x USB to UART Br
Macros
root@bionic-arm64:/home/uisc/clv5051# ls
arinc-func  bm-func  libusb-test-arm  listdevs  rt-func  thread
bc-func    hello    listdev         listdevs-arm  test-bc
root@bionic-arm64:/home/uisc/clv5051# ./Listdevs
05e3:0620 (bus 2, device 6)
1d6b:0003 (bus 2, device 1)
ed04:5218 (bus 1, device 7)
05e3:0610 (bus 1, device 6)
1d6b:0002 (bus 1, device 1)
root@bionic-arm64:/home/uisc/clv5051#
```

9.3. ARINC429 功能测试

将提供的 linux/src 软件包拷贝到 linux 系统，进入到/clv-5061/test/路径下，
执行编译；



```
zynq@zynq-virtual-machine: /mnt/hgfs/share/usb_driver/clv-5061/test
File Edit View Search Terminal Help
zynq@zynq-virtual-machine: /mnt/hgfs/share/usb_driver/clv-5061/test$ ls
Makefile test.c
zynq@zynq-virtual-machine: /mnt/hgfs/share/usb_driver/clv-5061/test$ make
aarch64-linux-gnu-gcc test.c ../driver/CLV5061.c -o arinc-func -I /usr/local/arm/include/lib
usb-1.0/ -L /usr/local/arm/lib/ -pthread -lusb-1.0
zynq@zynq-virtual-machine: /mnt/hgfs/share/usb_driver/clv-5061/test$
zynq@zynq-virtual-machine: /mnt/hgfs/share/usb_driver/clv-5061/test$ ls
arinc-func Makefile test.c
zynq@zynq-virtual-machine: /mnt/hgfs/share/usb_driver/clv-5061/test$
```

将模块的通道接口短接进行自回环测试；将当前可执行文件拷贝到 ARM 环

境下运行后效果如下：

```

root@bionic-arm64:/home/uisc/clv5061# ls
arinc-func
root@bionic-arm64:/home/uisc/clv5061# ./arinc-func
CLV5061 Init Ok.
ARINC 429 Start.
CH0 DAT:5555aaaa Lable:aa SDI:02 Data:5556a SSM:02 Parity:0 TIME:(0)0:0:0.729413
CH0 DAT:5555aaaa Lable:aa SDI:02 Data:5556a SSM:02 Parity:0 TIME:(0)0:0:1.729453
CH0 DAT:5555aaaa Lable:aa SDI:02 Data:5556a SSM:02 Parity:0 TIME:(0)0:0:2.729497
CH0 DAT:5555aaaa Lable:aa SDI:02 Data:5556a SSM:02 Parity:0 TIME:(0)0:0:3.729538
CH0 DAT:5555aaaa Lable:aa SDI:02 Data:5556a SSM:02 Parity:0 TIME:(0)0:0:4.729578
CH0 DAT:5555aaaa Lable:aa SDI:02 Data:5556a SSM:02 Parity:0 TIME:(0)0:0:5.729623
CH0 DAT:5555aaaa Lable:aa SDI:02 Data:5556a SSM:02 Parity:0 TIME:(0)0:0:6.729663
CH0 DAT:5555aaaa Lable:aa SDI:02 Data:5556a SSM:02 Parity:0 TIME:(0)0:0:7.729702
CH0 DAT:5555aaaa Lable:aa SDI:02 Data:5556a SSM:02 Parity:0 TIME:(0)0:0:8.729748
CH0 DAT:5555aaaa Lable:aa SDI:02 Data:5556a SSM:02 Parity:0 TIME:(0)0:0:9.729788
CH0 DAT:5555aaaa Lable:aa SDI:02 Data:5556a SSM:02 Parity:0 TIME:(0)0:0:10.729827
CH0 DAT:5555aaaa Lable:aa SDI:02 Data:5556a SSM:02 Parity:0 TIME:(0)0:0:11.729873
CH0 DAT:5555aaaa Lable:aa SDI:02 Data:5556a SSM:02 Parity:0 TIME:(0)0:0:12.729913
CH0 DAT:5555aaaa Lable:aa SDI:02 Data:5556a SSM:02 Parity:0 TIME:(0)0:0:13.729953
CH0 DAT:5555aaaa Lable:aa SDI:02 Data:5556a SSM:02 Parity:0 TIME:(0)0:0:14.729998
CH0 DAT:5555aaaa Lable:aa SDI:02 Data:5556a SSM:02 Parity:0 TIME:(0)0:0:15.730037
CH0 DAT:5555aaaa Lable:aa SDI:02 Data:5556a SSM:02 Parity:0 TIME:(0)0:0:16.730078
CH0 DAT:5555aaaa Lable:aa SDI:02 Data:5556a SSM:02 Parity:0 TIME:(0)0:0:17.730123
CH0 DAT:5555aaaa Lable:aa SDI:02 Data:5556a SSM:02 Parity:0 TIME:(0)0:0:18.730162
CH0 DAT:5555aaaa Lable:aa SDI:02 Data:5556a SSM:02 Parity:0 TIME:(0)0:0:19.730203
CH0 DAT:5555aaaa Lable:aa SDI:02 Data:5556a SSM:02 Parity:0 TIME:(0)0:0:20.730243
CH0 DAT:5555aaaa Lable:aa SDI:02 Data:5556a SSM:02 Parity:0 TIME:(0)0:0:21.730288
CH0 DAT:5555aaaa Lable:aa SDI:02 Data:5556a SSM:02 Parity:0 TIME:(0)0:0:22.730328
  
```

10. 产品选型说明

我们提供各种总线接口的 ARINC429 产品，可适应不同应用需求，欢迎来电咨询。产品型号编码说明如下：

