



NS11029

基于 Intel® Atom 处理器的轻量化控制系统主控板

用 户 手 册

REV1.0



北京鲲鹏凌昊智能技术有限公司

文档状态	<input type="checkbox"/> 草稿 <input checked="" type="checkbox"/> 正式发布 <input type="checkbox"/> 修订
文档标识符	
当前版本	1.0
作者	
完成日期	

历史版本

版本 / 状态	作者	参与者	发布日期	描述
1.0	Calida He	Judy Cao Hassan Han	20221206	第一次正式发布。

免责声明

北京鲲鹏凌昊智能技术有限公司（“凌昊智能”为注册商标，以下均简称“凌昊智能”）特别提醒，本用户手册中包含的信息可能会随着产品的不断升级有所更改。本文档不要求凌昊智能®就用户手册中描述的技术流程或用户手册中列出的任何产品特性提供任何保证。凌昊智能®对所述产品的使用不承担任何责任或义务，不授予该产品的任何专利、版权或掩码作品权利下的许可或所有权。本用户手册中描述的应用程序仅用于说明目的，凌昊智能®不声明或保证此类应用程序将适用于指定用途。凌昊智能®明确告知用户，本用户手册仅适用于大多数通用场景，不适用于特殊场景。如有疑问，请联系凌昊智能®。

本用户手册受版权保护，凌昊智能®保留所有权利。未经凌昊智能®书面许可，不得以任何形式或通过任何方式（电子、机械、影印、录音或其他方式）复制、传输、转录、存储在检索系统中，不得翻译成任何语言及计算机语言。本用户手册中包含的信息会随着凌昊智能®对产品的技术改进而不断更新，当前版本用户手册仅反映凌昊智能®当前版本产品的技术状态。

前言

感谢您选择由凌昊智能®自主研发的基于 Intel Atom 处理器平台的主控板 NS11029（该计算模块下文使用通用名称 NS11029，如有特殊配置则使用完整名称），本手册仅为使用此产品的工程师提供指导及参考。操作人员及可能参与本主控板操作、维护的所有人员须充分了解本手册的内容，以确保人身及设备的安全。

读者对象

- 产品使用的相关人员；
- 产品维护的相关人员；
- 产品的主控板设计相关的研发人员。

安全标志

本主控板有以下安全标志，其含义如表 1 所示：

表 1 设备安全标志

安全标志	标志含义
	表示禁止吸烟，模块工作状态时严禁吸烟。
	表示禁止烟火，模块工作状态严禁烟火，随时确保消防设施/器材完好。
	表示禁止拆卸。
	表示注意安全。
	表示当心触电。
	表示防止静电损坏板卡。

注意事项

- 操作人员必须经过系统操作及维护培训；
- 插拔主控板必须保证在断电的情况下进行；
- 插拔主控板时建议佩戴防静电手套，防止静电损坏设备；
- 以单板方式运输或贮存产品时需装入防静电袋中；
- 产品运输时包装箱内应有减震保护措施，防止运输过程中的挤压、碰撞或损坏；
- 产品贮存应注意通风、防潮、防尘，严禁具有腐蚀性的物品共同存放。

目 录

1. 概述	1
1.1 产品简介	1
1.2 原理框图	2
1.3 结构尺寸	3
1.4 产品配置	4
2. 功能规格	5
2.1 CPU	5
2.2 单片机	6
2.3 PCIe 接口	6
2.4 网口	6
2.5 指示灯	7
2.6 其他接口	9
2.7 使用要求	9
2.7.1 电源功耗	9
2.7.2 工作温度	10
2.8 散热结构	10
2.9 后出连接器接口定义	10
3. 硬件安装与拆卸	12
3.1 安装和上电	13
3.1.1 安装程序	13
3.1.2 安装步骤	13
3.2 拆卸	13
4. VxWorks 配置使用	14
4.1 新建 base 工程和编译 bsp.....	14
4.1.1 快捷方式位置	14
4.1.2 新建基础工程	14
4.2 设置显示方式和串口波特率	15
4.2.1 设置显示方式	15
4.3 设置串口参数	17
4.3.1 设置 fifo 长度	17
4.3.2 设置串口地址和中断	17

4.4	设置 shell banner	18
4.5	设置 ftp 功能	18
4.6	设置自启动脚本命令	19
4.7	添加 1860 网卡	19
4.7.1	1860 网卡驱动代码的添加	19
4.7.2	修改 priConfig.c 文件	20
4.8	添加 bt 模块内置网卡	20
4.8.1	添加文件到编译环境	20
4.9	网络包添加	21
4.10	编译 dkm 工程	22
4.10.1	新建 DKM 工程	22
4.11	编译 bootrom.bin 和 VxWorks 镜像	23
4.12	烧录工程	24
5.	VxWorks 操作	24
5.1	常用的命令	24
5.1.1	网络相关命令	24
5.1.2	加载 dkm 程序相关命令	24
5.1.3	ftp 功能	24
5.2	修改 udp 和 tcp 的 buf	25
5.3	修改 gei0 参数调节 udp 丢包问题	25
5.4	修改 i210 网卡百兆调整不自动协商	25
5.5	修改 1860 屏蔽 1860 网卡自协商设置成百兆	26
6.	操作注意事项	26
6.1	使用说明	26
6.1.1	使用前的准备和检查	26
6.1.2	开机	26
6.1.3	关机	26
6.1.4	安全保护措施及注意事项	27
6.1.5	故障应急处理	27
6.2	维护保养	27
6.2.1	日常维护保养工作	27
6.2.2	长期停放时的维护保养	27

1. 概述

1.1 产品简介

轻量化载板用于扩展主控板接口，和主控板共同完成主控逻辑的运算和系统状态维护，提供与执行子系统的通信通道并与执行子系统各功能单元进行数据交互。

逻辑子系统由主控板和网关板组成，板卡之间通过以太网进行通信。主控板里的核心组件是两片完全相同的 CPU 模块，两个 CPU 模块来实现 VSC 平台逻辑部的二取二表决机制，并将确定性边界的数据交换并进行比较，以确认两片 CPU 模块运行正常。两个 CPU 模块采用任务及同步原理，使用独立的晶振。

产品名称：轻量化 Intel Atom 处理器平台主控板

产品代号：NS11029

1.2 原理框图

NS11029 主控板原理框图见图 1。

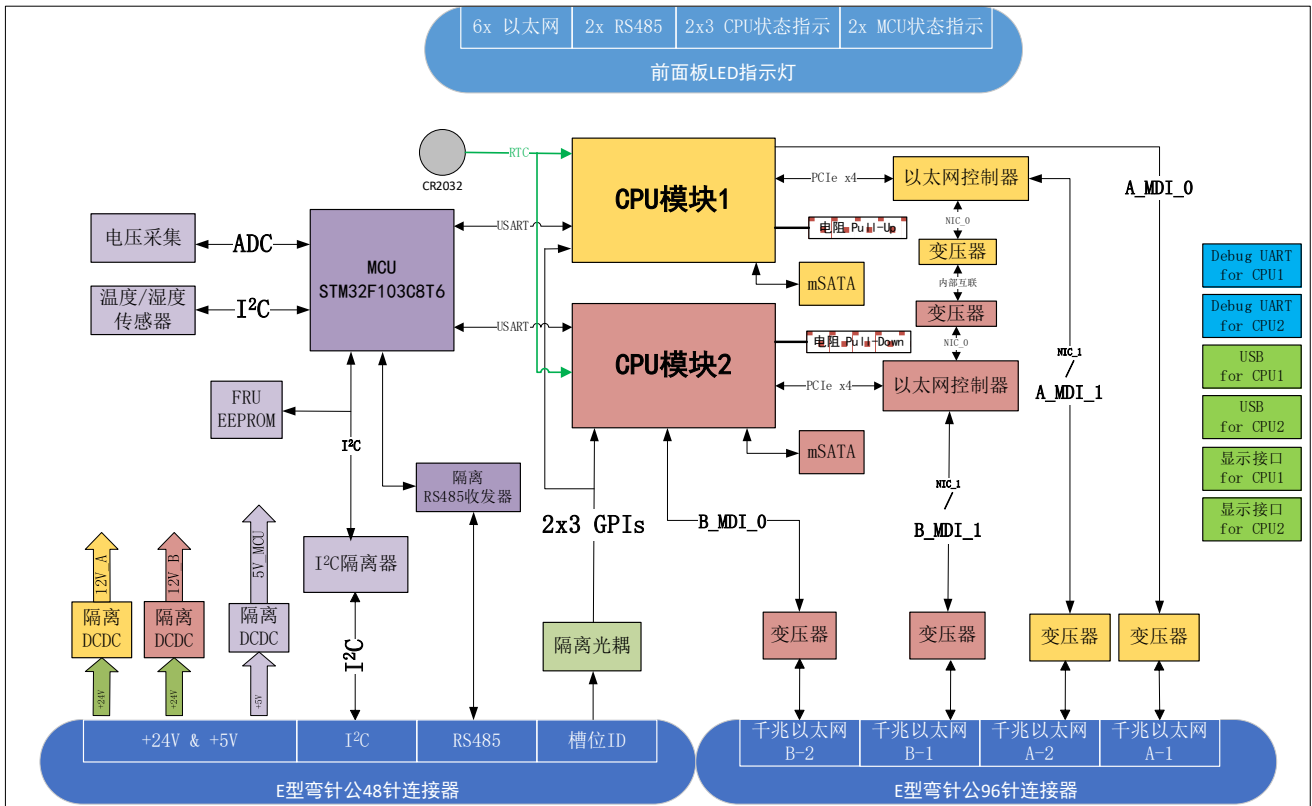


图 1 NS11029 原理框图

1.3 结构尺寸

主控板采用统一 PCB 尺寸结构，尺寸为 160mm（长）×233.35mm（高）。主控板结构要素图如图 2 所示：

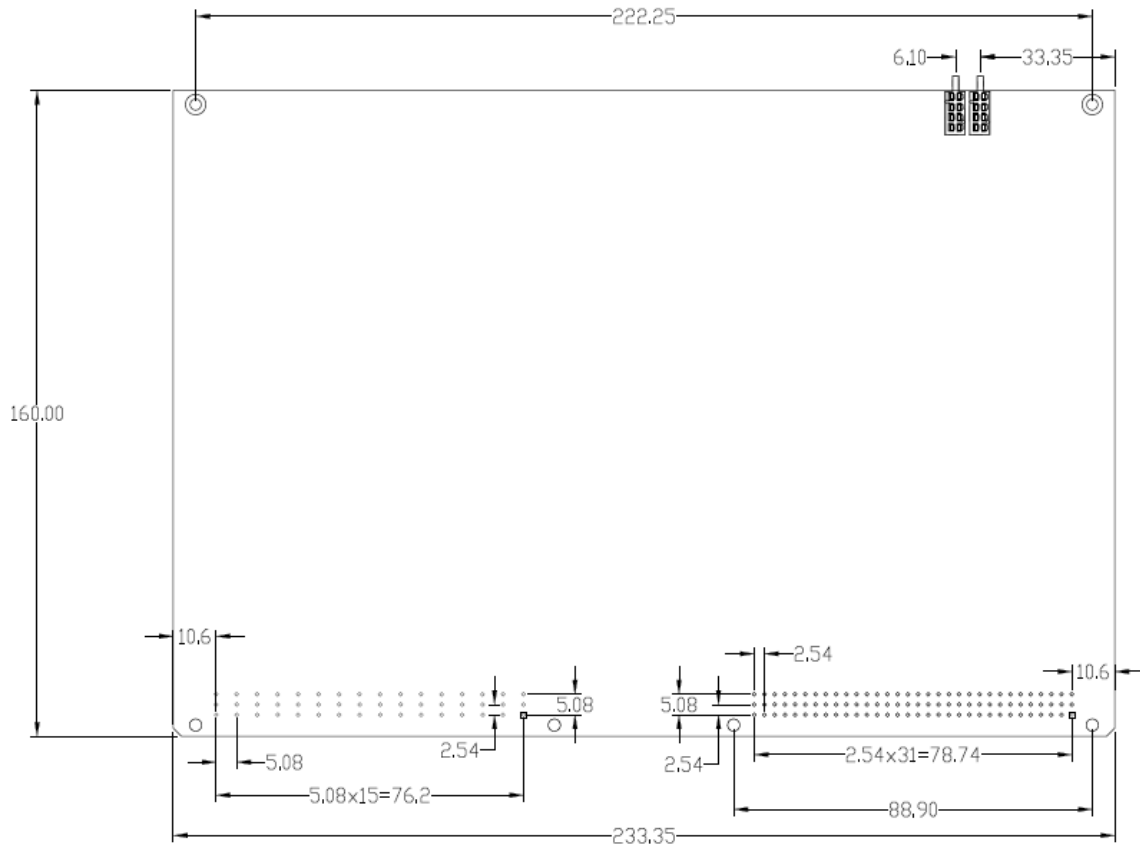


图 2 主控板结构示意图

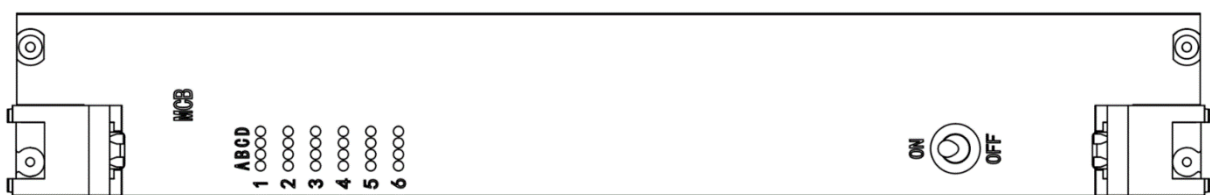


图 3 面板结构示意图

1.4 产品配置

NS11029 模块产品配置信息如表 2 所示：

表 2 NS11029 产品基本配置

产品名称	订货代码	产品简述
NS11029	10-0000-1361	6U 8HP 风冷 CPU 控制板，后出为欧式连接器，板卡搭载两个 CPU 模块。 对外接口包括：4x GbE(每个处理器两路)；1x RS485(MCU)；1x IIC(MCU)； 5x GA；2x USB TypeA；2x USB TypeC；2x DP。 工作温度：-25℃~+70℃，刷三防。

工作温度 = -25℃~+70℃，-40℃~+70℃

更多可选配置请联系凌昊智能®技术支持。

2. 功能规格

2.1 CPU

主控板 NS11029 使用了两块 CPU 模块，该模块主要技术参数如下：

- a) 84x 55mm
- b) Pin-out Type 10
- c) 工作温度：-40℃ ~+85℃
- d) 基于 Intel Atom 处理器
- e) 支持 3 路 PCIe x1 接口
- f) 1x I²C 总线
- g) 1x LPC 总线
- h) 6x USB2.0 接口
- i) 1x USB3.0 接口
- j) 1 路 10/100/1000Mb 自适应以太网
- k) 支持 2 路两线串行接口 (TX/RX)
- l) 1x HDA
- m) 1 路 SATAII 接口
- n) 8x GPIO
- o) 1x DP++
- p) 1x FAN PWM out

2.2 单片机

主控板 NS11029 采用 ST 微处理器 STM32F103C8T6 实现单片机部分功能，主要用于实现温湿度管理、电压监测、电子 ID 读取等功能。

- a) 芯片主要技术参数如下：CPU 内核：ARM Cortex-M3 CPU
- b) 最大主频：72MHz
- c) 工作电压范围：2V~3.6V
- d) 内部振荡器：有
- e) FLASH 容量：64K@x8bit RAM 总容量：20KB
- f) GPIO 端口数量：37 ADC(单元数/通道数/位数)：1@x10ch/12bit
- g) 外设/功能/协议栈：温度传感器；DMA；看门狗；POR；PWM
- h) 工作温度范围：-40℃~+85℃

IIC 和 RS485 连接至后出 E 型弯针公 48 针连接器上。通过内部 I2C 总线跟温湿度传感器互联，通过 ADC 引脚采集板卡电源电压幅值。传感器选择 TE 公司的带 I2C 的温湿度传感器 SHT20。

SHT20 是具有高可靠性和准确性的集成智能温湿度传感器。它有增强信号处理的功能，具有两个不同的用户可选择的 IIC 地址，通信速率高达 1MHz。

2.3 PCIe 接口

CPU 模块支持 PCIe2.0 接口，用于 CPU 模块扩展以太网接口进行板间通信及控制。

2.4 网口

主控板共有 6 路以太网接口，其中 4 路为 96 针 E 型欧式连接器后出接口，另外 2 路用于板内互联通信。

4 路后出网络接口通过 2 种方案实现。其中 2 路采用 2 块 CPU 模块的标准千兆网口经过网络变压器后连接至欧式 E 型 96 针连接器，并使用 ESD 进行静电防护。另外 2 路由网讯的以太网控制器 WX1860AL2 引出到 96 针 E 型欧式连接器，并使用 ESD 进行静电防护。

用于 2 个 CPU 模块之间通信的以太网从 CPU 模块的 PCIe 接口扩展引出，PCIe 通过千兆以太网控制器 WX1860AL2 引出 1000Base-T 接口进行板内互联。

2.5 指示灯

主控板前面板引出指示灯，包含接口状态指示灯（6 个以太网 Link/Active#指示灯、1 个 RS485 数据收发指示灯）、CPU 运行状态指示灯（每个 CPU 引出 3 个状态指示灯）、检测状态指示灯（2 个从 MCU 引出）。



图 4：指示灯位置示意图

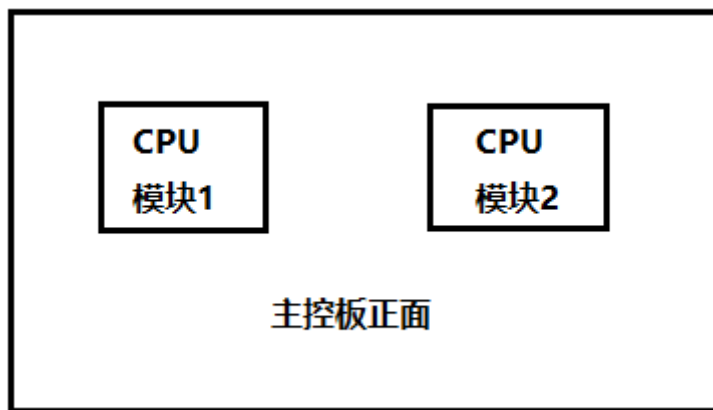


图 5：CPU 模块位置示意图

- 主控板上电后指示灯默认状态如表 3 所示。

表 3：主控板上电后指示灯状态

灯位编号	含义	状态
1-A	单片机运行状态指示灯	出厂常亮，用户可自定义
1-B	CPU 模块 1 状态指示灯	出厂常亮，用户可自定义
1-C	CPU 模块 1 状态指示灯	出厂常亮，用户可自定义
1-D	CPU 模块 1 状态指示灯	出厂常亮，用户可自定义
2-A	单片机运行状态指示灯	出厂常亮，用户可自定义
2-B	CPU 模块 2 状态指示灯	出厂常亮，用户可自定义
2-C	CPU 模块 2 状态指示灯	出厂常亮，用户可自定义
2-D	CPU 模块 2 状态指示灯	出厂常亮，用户可自定义

3-A	CPU 模块 1 上 i210 网络 act 灯	常灭
3-B	CPU 模块 1 连接的 WX1860 N1 通道网络 act 灯	常灭
3-C	CPU 模块 1 连接的 WX1860 N0 通道网络 act 灯	常灭
3-D	RS485 收发灯	常灭
4-A	CPU 模块 1 上 i210 网络 link 灯	常灭
4-B	CPU 模块 1 连接的 WX1860 N1 通道网络 link 灯	常灭
4-C	CPU 模块 1 连接的 WX1860 N0 通道网络 link 灯	常灭
4-D	RS485 收发灯	常灭
5-A	CPU 模块 2 上 i210 网络 act 灯	常灭
5-B	CPU 模块 2 连接的 WX1860 N1 通道网络 act 灯	常灭
5-C	CPU 模块 2 连接的 WX1860 N0 通道网络 act 灯	常灭
6-A	CPU 模块 2 上 i210 网络 link 灯	常灭
6-B	CPU 模块 2 连接的 WX1860 N1 通道网络 link 灯	常灭
6-C	CPU 模块 2 连接的 WX1860 N0 通道网络 link 灯	常灭

- 主控板 6 路网络有数据传输时指示灯的状态如表 3 所示。

表 4: 6 路网络有数据时指示灯状态

灯位编号	含义	状态
1-A	单片机运行状态指示灯	出厂常亮, 用户可自定义
1-B	CPU 模块 1 状态指示灯	出厂常亮, 用户可自定义
1-C	CPU 模块 1 状态指示灯	出厂常亮, 用户可自定义
1-D	CPU 模块 1 状态指示灯	出厂常亮, 用户可自定义
2-A	单片机运行状态指示灯	出厂常亮, 用户可自定义
2-B	CPU 模块 2 状态指示灯	出厂常亮, 用户可自定义
2-C	CPU 模块 2 状态指示灯	出厂常亮, 用户可自定义
2-D	CPU 模块 2 状态指示灯	出厂常亮, 用户可自定义
3-A	CPU 模块 1 上 i210 网络 act 灯	闪烁

3-B	CPU 模块 1 连接的 WX1860 N1 通道网络 act 灯	闪烁
3-C	CPU 模块 1 连接的 WX1860 N0 通道网络 act 灯	闪烁
3-D	RS485 收发灯	闪烁
4-A	CPU 模块 1 上 i210 网络 link 灯	常亮
4-B	CPU 模块 1 连接的 WX1860 N1 通道网络 link 灯	常亮
4-C	CPU 模块 1 连接的 WX1860 N0 通道网络 link 灯	常亮
4-D	RS485 收发灯	闪烁
5-A	CPU 模块 2 上 i210 网络 act 灯	闪烁
5-B	CPU 模块 2 连接的 WX1860 N1 通道网络 act 灯	闪烁
5-C	CPU 模块 2 连接的 WX1860 N0 通道网络 act 灯	闪烁
6-A	CPU 模块 2 上 i210 网络 link 灯	常亮
6-B	CPU 模块 2 连接的 WX1860 N1 通道网络 link 灯	常亮
6-C	CPU 模块 2 连接的 WX1860 N0 通道网络 link 灯	常亮

2.6 其他接口

整板接口由 E 型弯针公 48pin 和 96pin 连接器引出。48pin 连接器引出 1 路 RS485 接口、1 路 IIC 接口、6 位槽位 ID 信号接口以及 +24V 和 +5V 电源输入接口；96pin 连接器引出 4 路千兆以太网接口。

主控板通过 MCU 扩展 1 路 RS485 接口。

2.7 使用要求

2.7.1 电源功耗

主控板采用 +24V 和 +5V 供电，+24V 为整板的主供电，负责给整板的功能模块供电，+5V 给板内 MCU 功能模块供电。

整板热设计功耗预计 27.85W。

2.7.2 工作温度

工作温度为-25℃~+70℃，产品在该温度范围内能正常工作。工作温度最低可支持-40℃。

2.8 散热结构

刀片散热默认采用导冷散热形式，配置有定做的散热器。如需风冷散热器，可联系凌昊智能®定制。

2.9 后出连接器接口定义

主控板后出接口从标准欧式 E 型弯针 DIN Harting 96pin 和 48pin 连接器引出。连接器接口信号定义如表 5 所示：

表 5 E 型弯针公 48 针连接器接口定义

PIN	D	B	Z
2	+5V	+5V	+5V
4	+5V_GND	+5V_GND	+5V_GND
6			
8			+5V
10	5V_GND	GND_SHELL	IIC_SCL
12	RS_485_P	GND_SHELL	IIC_SDA
14	RS_485_N	GND_SHELL	5V_GND
16	5V_GND		
18			ID0_IN
20			ID1_IN
22	5V_GND		ID2_IN
24			ID3_IN
26			ID4_IN
28			
30	+24V	+24V	+24V
32	+24V_GND	+24V_GND	+24V_GND

注：

- 1) [D&B&Z] &[2:32]：连接器的定位编号；
- 2) +5V & +5V_GND &+24V & +24V_GND：电源输入；
- 3) RS_485_P & RS_485_N：RS485 接口信号；
- 4) IIC_SCL & IIC_SDA：IIC 接口信号；

5) ID[0:5]_IN: 槽位 ID 信号;

ID0 和 ID1 为预留 ID, 实际不使用; ID2~ID4 为有效的 ID 识别位。

6) GND_SHELL: 机壳地。

表 6 E 型弯针公 96 针连接器接口定义

PIN	A	B	C
1	ETH_SHELL_BACK	CPU1_ETH1_TRX3_N	CPU1_ETH1_TRX3_P
2	ETH_SHELL_BACK	CPU1_ETH1_TRX2_N	CPU1_ETH1_TRX2_P
3	ETH_SHELL_BACK	CPU1_ETH1_TRX1_N	CPU1_ETH1_TRX1_P
4	ETH_SHELL_BACK	CPU1_ETH1_TRX0_N	CPU1_ETH1_TRX0_P
5	ETH_SHELL_BACK		
6	ETH_SHELL_BACK		
7	ETH_SHELL_BACK	CPU1_ETH2_TRX3_N	CPU1_ETH2_TRX3_P
8	ETH_SHELL_BACK	CPU1_ETH2_TRX2_N	CPU1_ETH2_TRX2_P
9	ETH_SHELL_BACK	CPU1_ETH2_TRX1_N	CPU1_ETH2_TRX1_P
10	ETH_SHELL_BACK	CPU1_ETH2_TRX0_N	CPU1_ETH2_TRX0_P
11	ETH_SHELL_BACK		
12	ETH_SHELL_BACK		
13	ETH_SHELL_BACK	CPU2_ETH1_TRX3_N	CPU2_ETH1_TRX3_P
14	ETH_SHELL_BACK	CPU2_ETH1_TRX2_N	CPU2_ETH1_TRX2_P
15	ETH_SHELL_BACK	CPU2_ETH1_TRX1_N	CPU2_ETH1_TRX1_P
16	ETH_SHELL_BACK	CPU2_ETH1_TRX0_N	CPU2_ETH1_TRX0_P
17	ETH_SHELL_BACK		
18	ETH_SHELL_BACK		
19	ETH_SHELL_BACK	CPU2_ETH2_TRX3_N	CPU2_ETH2_TRX3_P
20	ETH_SHELL_BACK	CPU2_ETH2_TRX2_N	CPU2_ETH2_TRX2_P
21	ETH_SHELL_BACK	CPU2_ETH2_TRX1_N	CPU2_ETH2_TRX1_P
22	ETH_SHELL_BACK	CPU2_ETH2_TRX0_N	CPU2_ETH2_TRX0_P

注:

1) [A&B&C] &[1:22]: 连接器的定位编号;

2) CPU[1:2]_ETH[1:2]_TRX[0:3]_P/N: 千兆以太网信号。

3. 硬件安装与拆卸

本章节内容包含 NS11029 使用前的检查步骤，配置 NS11029 相关选项及 NS11029 的安装等相关内容。建议按以下顺序配置 NS11029。

使用前检查

主控板交付后应立即对包装进行彻底检查，以避免运输过程中造成的损坏。

打开包装后，应仔细检查电路板是否有物理损坏，如组件松动。若主控板出现明显损坏的状况，请立即联系凌昊智能®。

安装前准备

- 作业前要佩戴防静电手套、防静电手环，条件允许还应穿戴防静电服，防静电鞋具，此要求贯穿整个作业过程；
- 工作台应铺盖防静电台垫和必要的海绵垫板；
- 市电电压不低于 210V；
- 产品状态需要进行明确的标识，放置区域明确区分；
- 作业过程中连接板卡和接插头时不可使用蛮力；
- 如无特殊要求，在以下大气环境中进行作业：
 - 室温 (+20°C ~ +25°C) ；
 - 相对空气湿度 10% ~ 90%（无凝露）。

将主控板安装到机箱之前，请检查以下几点：

- 背板电源轨配置；
- 电源单元电流能力；

- 主控板插槽与背板插槽是否兼容。

3.1 安装和上电

3.1.1 安装程序

安装程序一般包括：

- a) 拆卸螺丝，拆卸散热器等；
- b) 安装散热器、螺丝等；
- c) 拆卸，安装所用工具包括螺丝刀，电烙铁等。

3.1.2 安装步骤

- a) 安装前确保系统电源已关闭；
- b) 将主控板滑入指定插槽，确保板卡整齐地装入滑道，且对齐槽位开始正确啮合；
- c) 将主控板推入卡笼，直到所有 E 型欧式连接器都牢固定位；
- d) 将面板螺钉拧紧；
- e) 启动系统。

3.2 拆卸

- a) 如需卸下主控板，请在关闭系统电源之前关闭应用程序和操作系统软件；
- b) 打开弹出器手柄并取出主控板。

第 章

4

4. VxWorks 配置使用

4.1 新建 base 工程和编译 bsp

4.1.1 快捷方式位置

VxWorks 的快捷方式在以下路径：（6.6 和 6.8 版本的的路径基本一致）

C:\WindRiver660\WindRiver660\workbench-3.0\wrwb\platform\eclipse

4.1.2 新建基础工程

1. 新建 bsp 工程

将工程 bsp 放至以下路径：

C:\WindRiver683\vxworks-6.8\target\config

2. 在 IDE 中新建项目 project

选择镜像为上个步骤的 bsp bt10，选择 800sbs 并为工程命名。

名称	修改日期	类型	大小
all	2022/9/15 17:44	文件夹	
comps	2022/9/15 17:44	文件夹	
idp945	2022/9/15 17:44	文件夹	
idp3100_efi	2022/9/15 17:44	文件夹	
itl_atom_800s	2022/10/17 11:15	文件夹	
itl_crownbeach	2022/9/15 17:44	文件夹	
itl_greenCity	2022/9/15 17:44	文件夹	
itl_hanlanCreek	2022/9/15 17:44	文件夹	
itl_haswell	2022/9/15 17:44	文件夹	
itl_tolapai	2022/9/15 17:44	文件夹	
miscowabic	2022/9/15 17:44	文件夹	
pcPentium	2022/9/15 17:44	文件夹	
pcPentium2	2022/9/15 17:44	文件夹	
pcPentium3	2022/9/15 17:44	文件夹	
pcPentium4	2022/9/15 17:44	文件夹	
simpc	2022/9/15 17:44	文件夹	
itl_atom_771s.zip	2022/9/18 11:41	WinRAR ZIP 压缩文件	2,820 KB
Makefile	2008/5/17 2:43	文件	5 KB

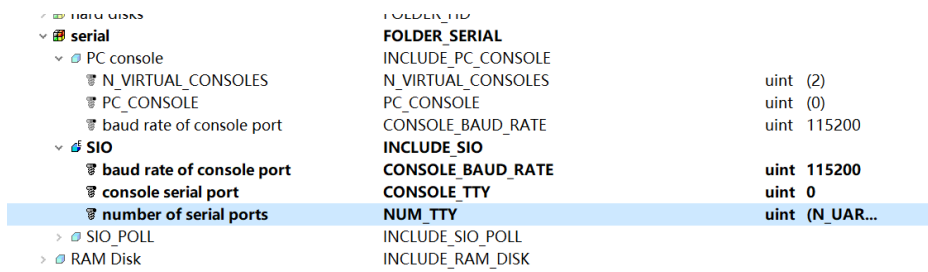
3. 选择板载支持包

编译器选择 **gnu**，进入工程后 build project。生成的文件为 VxWorks 镜像文件，文件存放于路径右键 properties 中的路径的 default 中。

4.2 设置显示方式和串口波特率

VxWorks 有两种必须的文件：bootrom.bin 文件和镜像 VxWorks 文件。两者的屏幕显示或者串口信息需要分别修改，Bootrom.bin 在 config.h 中设置；Vxworks 在图形化页面中配置。

如若以上配置显示 include pc_console，在 boot 过后会显示图中信息。



4.2.1 设置显示方式

1. 串口和屏幕显示二选一，编译时设置 console 和 baud。在 config.h 中设置 console 和 baud，相当给 bootrom.bin 设置。

- #undef include_pc_console 表示用串口
- #define include_pc_console 表示显示屏显示

```

Getting Started  config.h  vxSerialPort_v1.2 si  hwconf.c  ceshi222  confignum.h  pc.h  »1
#define ATA1_CONFIG      (ATA_GEO_CURRENT | ATA_PIO_AUTO | \
                          ATA_BITS_16      | ATA_PIO_MULTI)

#define ATA1_SEM_TIMEOUT (5)          /* ATA 1 sync. semaphore timeout */
#define ATA1_WDG_TIMEOUT (5)          /* ATA 1 watchdog timer timeout */
#define ATA1_SOCKET_TWIN (0)          /* Socket number (TWIN PCMCIA Card) */
#define ATA1_POWER_DOWN  (0)          /* ATA 1 power down mode */

/* console definitions */

#undef NUM_TTY
#define NUM_TTY          (N_UART_CHANNELS) /* number of tty channels */

#undef INCLUDE_PC_CONSOLE /* PC keyboard and VGA console */
#define INCLUDE_CONSOLE_USB_KEYBOARD
    
```

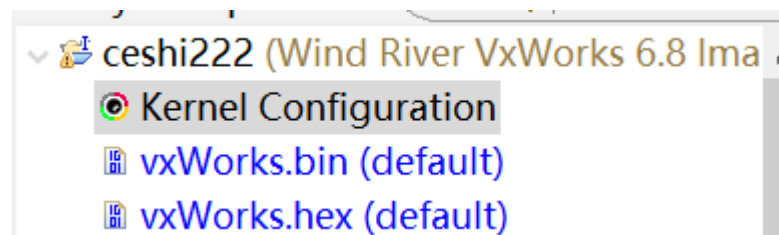
2. 设置串口波特率

```
Getting Started | config.h | vxSerialPort_v1.2_si | hwconf.c | ceshi222 | configInum.h | pc.h | »

/* CPU auto detection with sysCpuProbe() that support Intel CPUs for now */
#define INCLUDE_CPU_PROBE          /* define to perform sysCpuProbe() */
# undef CPU
# define CPU PENTIUM4             /* for CPU conditionals in BSP */
#define X86CPU_DEFAULT X86CPU_PENTIUM4 /* for sysProcessor set in BSP */

#undef CONSOLE_BAUD_RATE
#define CONSOLE_BAUD_RATE 115200 /* console baud rate */
```

3. 添加镜像的 console 和 baud



4. 查找 console 和 baud include，并设置好波特率。

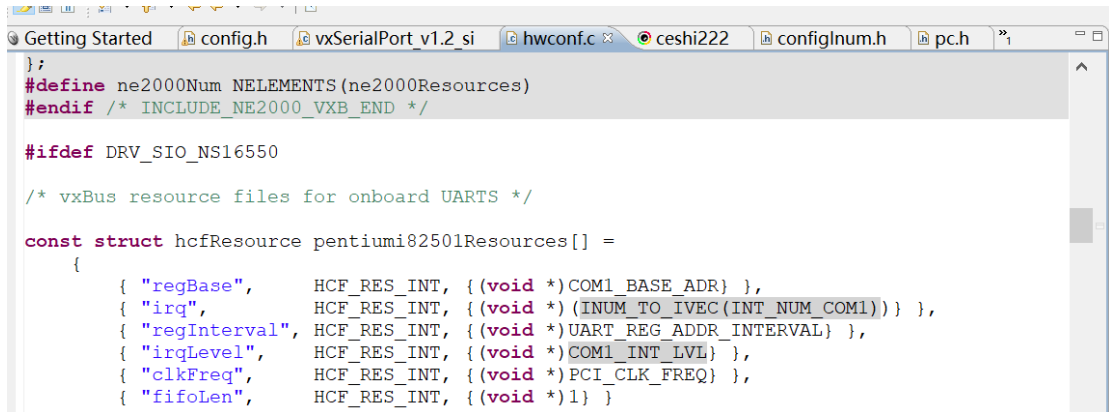
Include pc_console 则镜像打印会显示在屏幕上，目前不需要屏幕显示，只需用串口。

serial	FOLDER_SERIAL	
PC console	INCLUDE_PC_CONSOLE	
N_VIRTUAL_CONSOLES	N_VIRTUAL_CONSOLES	uint (2)
PC_CONSOLE	PC_CONSOLE	uint (0)
baud rate of console port	CONSOLE_BAUD_RATE	uint 115200
SIO	INCLUDE_SIO	
baud rate of console port	CONSOLE_BAUD_RATE	uint 115200
console serial port	CONSOLE_TTY	uint 0
number of serial ports	NUM_TTY	uint (N_UAR...
> SIO_POLL	INCLUDE_SIO_POLL	
> RAM Disk	INCLUDE_RAM_DISK	

4.3 设置串口参数

4.3.1 设置 fifo 长度

若现象是模块的串口只能收到 AHPX，发现问题在于 fifolen 的长度的问题，模块中没有 fifo。在软件设置中将 fifo 设置成 1 则可解决该问题。



```

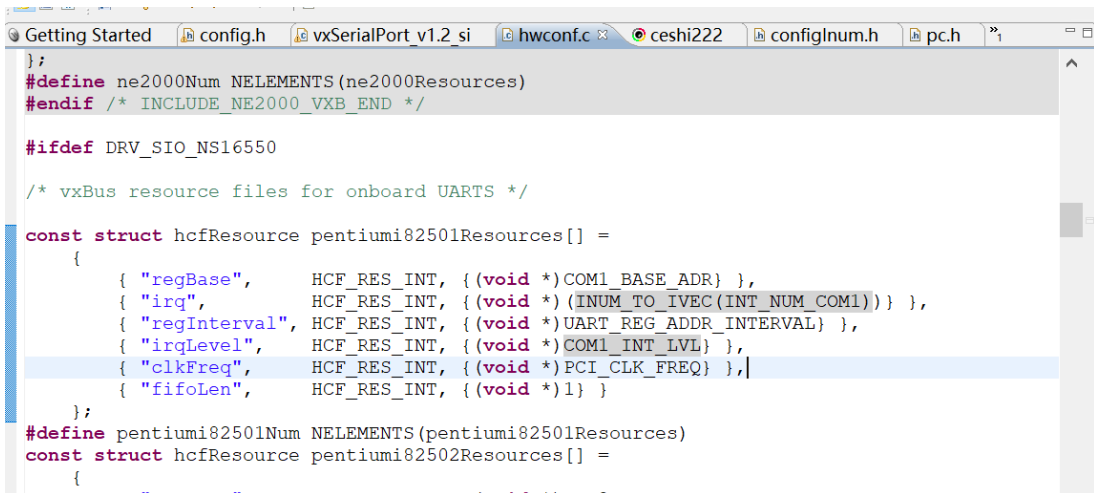
};
#define ne2000Num NELEMENTS(ne2000Resources)
#endif /* INCLUDE_NE2000_VXB_END */

#ifdef DRV_SIO_NS16550

/* vxBus resource files for onboard UARTS */

const struct hcfResource pentiumi82501Resources[] =
{
    { "regBase",      HCF_RES_INT, {(void *)COM1_BASE_ADR} },
    { "irq",          HCF_RES_INT, {(void *) (INUM_TO_IVEC(INT_NUM_COM1)) } },
    { "regInterval", HCF_RES_INT, {(void *)UART_REG_ADDR_INTERVAL} },
    { "irqLevel",     HCF_RES_INT, {(void *)COM1_INT_LVL} },
    { "clkFreq",      HCF_RES_INT, {(void *)PCI_CLK_FREQ} },
    { "fifolen",      HCF_RES_INT, {(void *)1} }
};
    
```

4.3.2 设置串口地址和中断



```

};
#define ne2000Num NELEMENTS(ne2000Resources)
#endif /* INCLUDE_NE2000_VXB_END */

#ifdef DRV_SIO_NS16550

/* vxBus resource files for onboard UARTS */

const struct hcfResource pentiumi82501Resources[] =
{
    { "regBase",      HCF_RES_INT, {(void *)COM1_BASE_ADR} },
    { "irq",          HCF_RES_INT, {(void *) (INUM_TO_IVEC(INT_NUM_COM1)) } },
    { "regInterval", HCF_RES_INT, {(void *)UART_REG_ADDR_INTERVAL} },
    { "irqLevel",     HCF_RES_INT, {(void *)COM1_INT_LVL} },
    { "clkFreq",      HCF_RES_INT, {(void *)PCI_CLK_FREQ} },
    { "fifolen",      HCF_RES_INT, {(void *)1} }
};

#define pentiumi82501Num NELEMENTS(pentiumi82501Resources)
const struct hcfResource pentiumi82502Resources[] =
{
    { "regBase",      HCF_RES_INT, {(void *)COM2_BASE_ADR} },
    { "irq",          HCF_RES_INT, {(void *) (INUM_TO_IVEC(INT_NUM_COM2)) } },
    { "regInterval", HCF_RES_INT, {(void *)UART_REG_ADDR_INTERVAL} },
    { "irqLevel",     HCF_RES_INT, {(void *)COM2_INT_LVL} },
    { "clkFreq",      HCF_RES_INT, {(void *)PCI_CLK_FREQ} },
    { "fifolen",      HCF_RES_INT, {(void *)1} }
};
    
```

Regbase 表示基地址，需要根据 bios 中的数据进行设置。

Irqllevel 可以改 irq 中断的号，图中表示中断地址 4。

进行以上两个操作即可配置好串口，注意一定要与 bios 中的数据一样。

按 F2 进入 BIOS 查看 serio 中的配置，注意需要关闭 sio 的使能。


```

Getting Started | config.h | vxSerialPort v1.2 si | hwconf.c | ceshi222 | configInum.h | pch
* I8250_CHAN_PARAS devParas structure in sysSerial.c for additional
* changes necessary for supporting 3 or more serial ports. Sometimes
* the BIOS also needs adjustment.
* SPR# 5704
*/

#define COM1_BASE_ADR      0x3f8
#define COM2_BASE_ADR      0x2f8
#define COM3_BASE_ADR      0x3e8
#define COM4_BASE_ADR      0x2e8
#define COM1_INT_LVL        0x04
/*#define COM1_INT_LVL      0x07*/
#define COM2_INT_LVL        0x03
#define COM3_INT_LVL        0x05
#define COM4_INT_LVL        0x09
#define UART_REG_ADDR_INTERVAL 1 /* address diff of adjacent regs. */
#define N_UART_CHANNELS    2
    
```

4.4 设置 shell banner

shell banner 加入，否则不会打印 VxWorks。

双击 kernel configuration，在开发工具上右键 find，

shell banner

include

4.5 设置 ftp 功能

> FTP Client backend	INCLUDE FTP
> FTP Server	INCLUDE IPF...
Authentication attempts before disconnect	FTPS_AUTH_... ch... "3"
Authentication callback routine	FTPS_AUTH_C... fu... ftps_auth_func
Control socket backlog queue	FTPS_SOCKET... ch... "0"
Data receive timeout	FTPS_RECV_TI... ch... "30"
Data send timeout	FTPS_SEND_TI... ch... "30"
Enable proxy FTP support	FTPS_ENABLE... ch... "0"
FTP initial directory	FTPS_INITIAL... ch... "/ahci00:1"
FTP root directory	FTPS_ROOT_... ch... ""
Install FTP server callback routine	FTPS_INSTAL... BO... TRUE
Local port base number	FTPS_LOCAL_... ch... "49151"

代码中添加以下：

```
int ftps_auth_func(void *session, char * password)
```

```
{
```

```
    return 0;
```

```
}
```

```

time test.c | net.c | usrApplnit.c | ceshi bt10
19 initialize user application code.
20 */
21
22 #include <vxWorks.h>
23 #if defined(PRJ_BUILD)
24 #include "prjParams.h"
25 #endif /* defined PRJ_BUILD */
26
27 int ftps_auth_func(void *session, char * password)
28 {
29     return 0;
30 }
31 /*****
32 *
    
```

4.6 设置自启动脚本命令

需要define下面的两个宏:

```
#define INCLUDE_SHELL
#define INCLUDE_STARTUP_SCRIPT
```

RTP Startup Facility: Command shell startup script	INCLUDE RTP_APPL_INIT_CMD_SHELL ...
kernel shell startup script	INCLUDE_STARTUP_SCRIPT
Pointer to MIB-2 sysDescr	MIB2SYS_DESCR
Command shell script file.	RTP_APPL_CMD_SCRIPT_FILE
target-resident kernel shell	INCLUDE_SHELL

主程序代码中添加图中代码。

```

1
2
3 void usrAppInit (void)
4 {
5 #ifdef USER_APPL_INIT
6     USER_APPL_INIT;    /* for backwards compatibility */
7 #endif
8
9     sleep(2);
10    ioDefPathSet("/ahci00:1/");
11    usrStartupScript("/ahci00:1/startup.sh");
12
13    /* add application specific code here */
14 }
15
16

```

4.7 添加 1860 网卡

4.7.1 1860 网卡驱动代码的添加

驱动源代码的添加，直接拖入到工程中即可。

- usrAppInit.c
- usrRtpApplInit.c
- vxbNgbeEnd.c
- vxbNgbeEnd.h
- bt10.wpi

4.7.2 修改 priConfig.c 文件

1. 在 `hardWareInterFaceBusInit` 函数前添加：`extern void ngbRegister();`声明。
2. 在 `hardWareInterFaceBusInit` 结尾添加 `ngbRegister();`函数。

(在第 652 行的部分和第 670 行)

```

647
648 /*****
649 *
650 * hardWareInterFaceBusInit - bus-subsystem initialization
651 */
652 extern void ngbRegister();
653 void hardWareInterFaceBusInit (void)
654 {
655     vxLibInit();           /* vxBus Library Initialization */
656     plbRegister();        /* Processor Local Bus */
657     pciRegister();        /* Peripheral Component Interconnect Bus */
658     feiRegister();        /* fei vxBus END driver */
659     geiRegister();        /* Intel PRO/1000 VxBus Enhanced Network Driver
660     miiBusRegister();     /* MII bus controller for ethernet transceivers
661     genPhyRegister();     /* Generic 10/100/1000 ethernet copper PHY drive
662     mvPhyRegister();      /* Marvell 88E1x11 10/100/1000 ethernet copper P
663     rtgPhyRegister();     /* RealTek RTL8169 C+ 10/100 ethernet PHY */
664     rtgRegister();        /* RealTek 8139C+/8168/8169/8110/8101E VxBus END
665     ns16550SioRegister(); /* ns16550 SIO support */
666     vxI8253TimerDrvRegister(); /* i8253 timer Driver */
667     vxIntelAhciStorageRegister(); /* Supports Ahci controllers */
668     ahciShowInit();       /* Supports 82801 Intel Serial/Parallel ATA cont
669     pentiumPciRegister(); /* Pentium PCI host controller */
670     ngbRegister();
    
```

3. 编译时禁止选择 `rebuild project` 选项，否则命令会被覆盖。

4.8 添加 bt 模块内置网卡

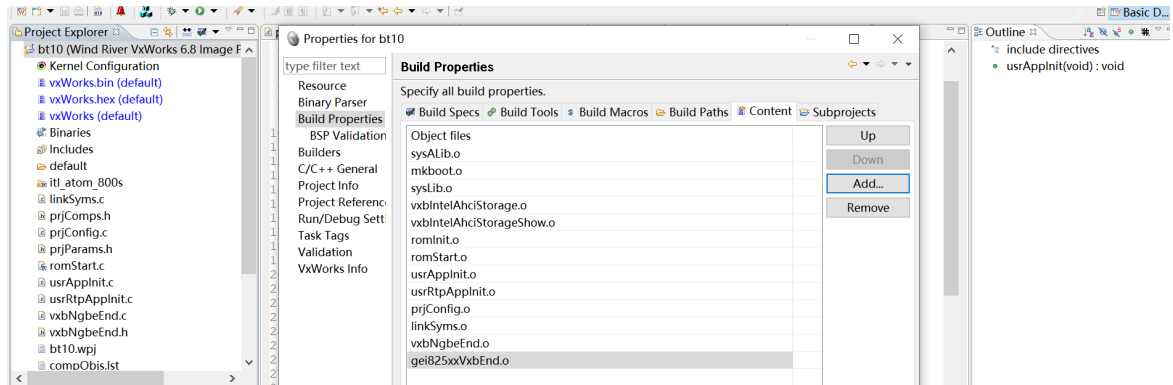
4.8.1 添加文件到编译环境

1. Makefile 中添加要编译的.o 文件。

```

31M_TEXT_ADRS = 00008000 # ROM entry address - A: or C:
32M_SIZE      = 00090000 # number of bytes of ROM space
33=lse
34ROM_TEXT_ADRS = fff20000 # ROM entry address - EPROM
35ROM_SIZE      = 0007fe00 # number of bytes of ROM space
36endif
37
38M_LOW_ADRS   = 00408000 # VxWorks image entry point
39M_HIGH_ADRS  = 04008000 # Boot image entry point
40
41TRA_DEFINE   = -DFAST_REBOOT
42
43CH_EXTRA     = mkboot.o vxIntelAhciStorage.o vxIntelAhciStorageShow.o gei825xxVxbEnd.o
44
45LEASE       += bootrom.bin mkboot.o
    
```

2. 打开 properties 选项检查编译的文件。

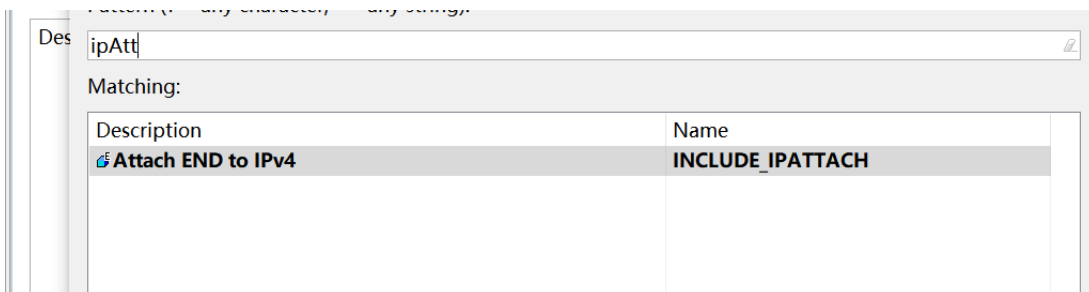


4.9 网络包添加

1. 添加 ping 命令。

ping6 wrapper	INCLUDE_IPWRAP_PING6
PING client	INCLUDE_PING
PING6 client	INCLUDE_PING6

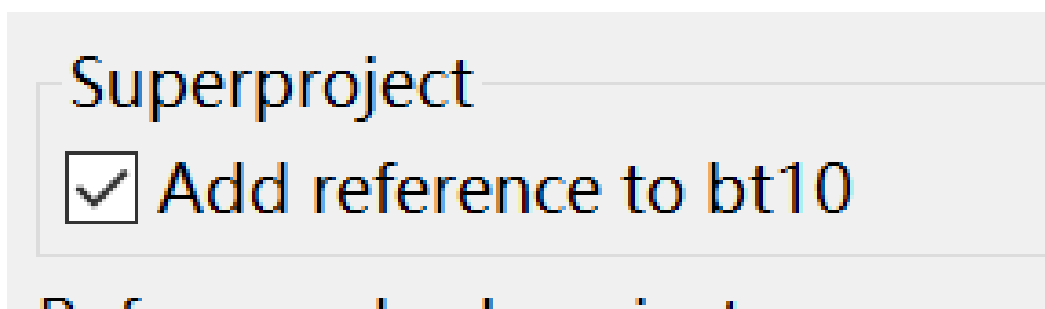
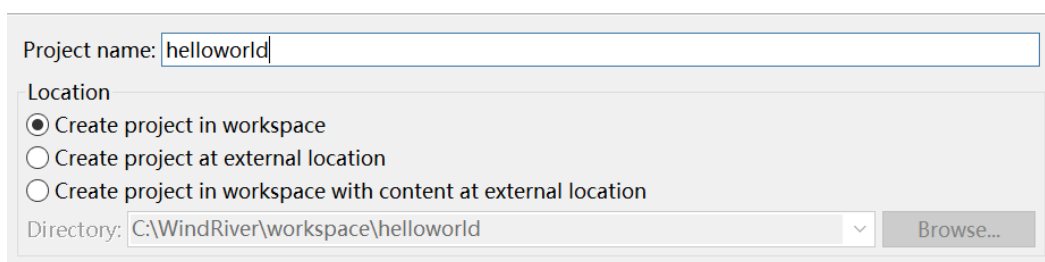
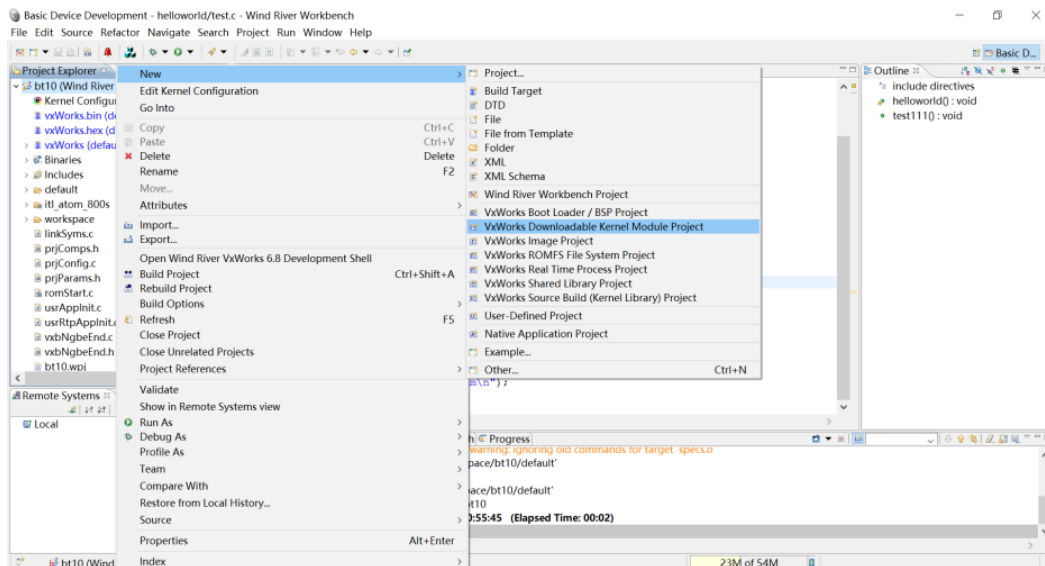
2. 添加新建网络操作命令

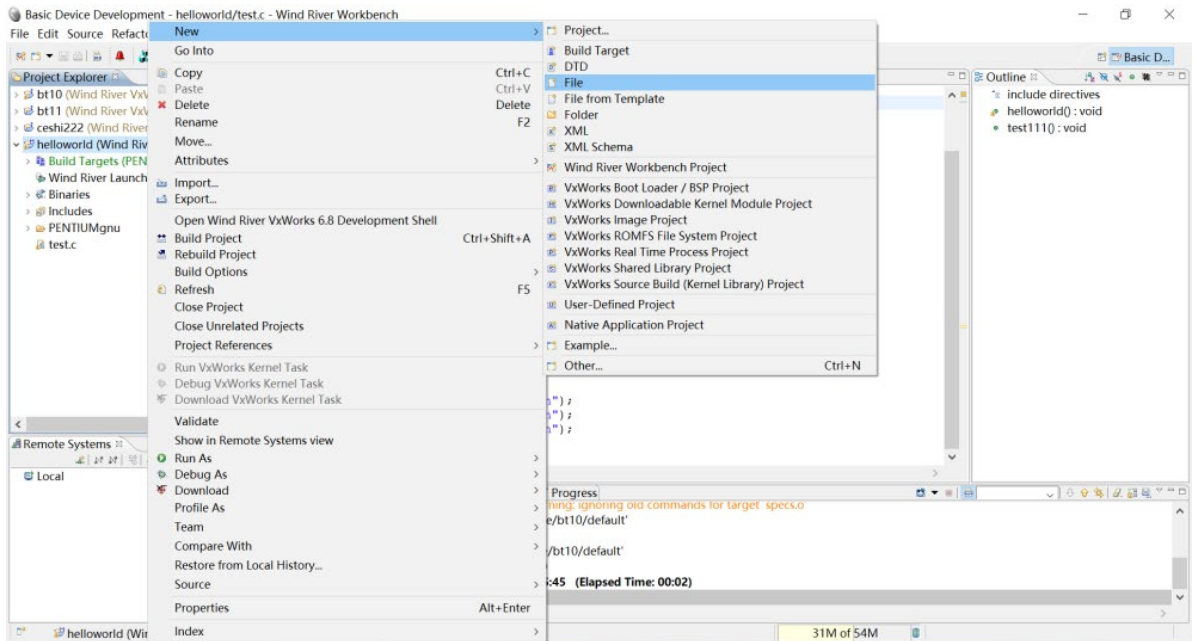


4.10 编译 dkm 工程

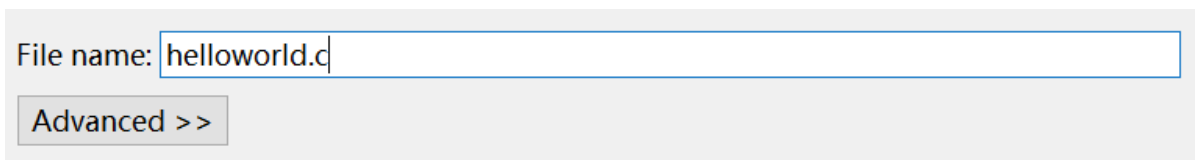
4.10.1 新建 DKM 工程

1. 完成以下图中操作后单击完成即可生成工程。

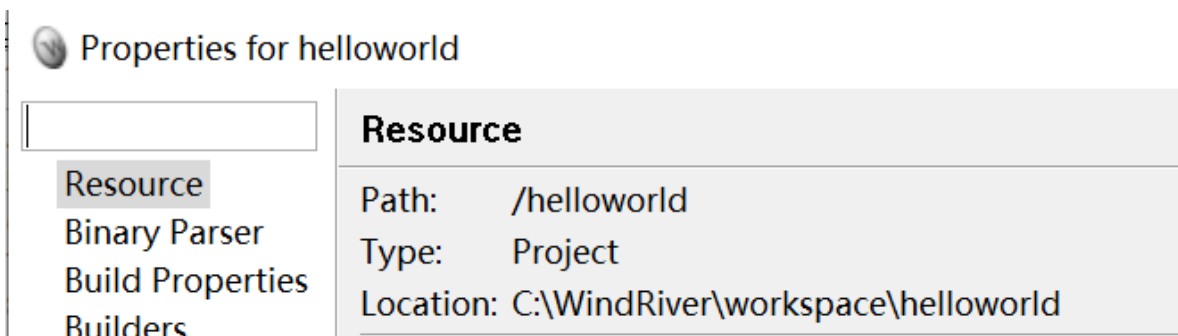




- 新建文件，在工程中修改文件内容编译即可。



- 文件编译之后生成的.out 文件在工程右键 properties 中查看。



4.11 编译 bootrom.bin 和 VxWorks 镜像

- 进入工程，直接 build project。

生成的文件是 VxWorks，VxWorks 镜像文件在路径右键 properties 中的路径的 default 里。

- 右键 open shell 后直接 make 即可。

- 使用 cd /d 命令找到板载支持包的路径 C:\WindRiver683\vxworks-6.8\target\config 下的路径

make clean

make bootrom.bin

生成的路径就在该目录中。

4.12 烧录工程

将生成的 bootrom.bin、vxworks 和 helloworld.out 放到系统盘中即可。

5. VxWorks 操作

5.1 常用的命令

5.1.1 网络相关命令

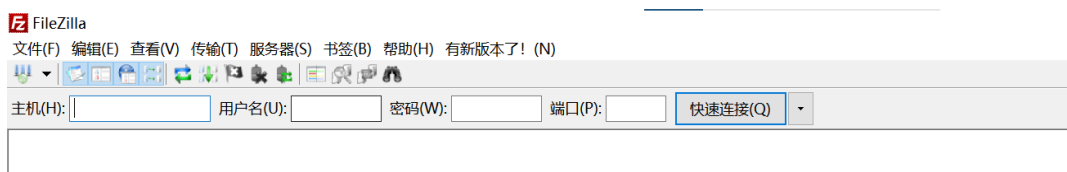
命令	含义
ipAttach()	表示添加网卡。如： ipAttach(0, "ngb") 表示添加 ngb0 网卡 ipAttach(1, "ngb") 表示添加 ngb1 网卡
ifconfig "ngb0 192.168.3.1 up"	表示设置网卡 ip 地址 并激活。
ping "192.168.1.1",100	表示 ping 100 次。
muxShow	表示显示网络设备。
ipDetach(0,"ngb")	表示删除网卡。

5.1.2 加载 dkm 程序相关命令

ld < hellword.out 执行其中的函数即可。

5.1.3 ftp 功能

使用 ftp 软件，并设置密码。



5.2 修改 udp 和 tcp 的 buf

Description	Name	Ty...	Value
Network Sockets	FOLDER_IPNE...		
Netlink socket	INCLUDE_IPNE...		
Routing socket support (default)	INCLUDE_IPN...		
Socket backend (default)	INCLUDE_IPN...		
Socket support (default)	INCLUDE_IPN...		
Address Notify	IPNET SOCK ...	ch...	"1"
AnonPortMax	IPNET SOCK ...	ch...	"65535"
AnonPortMin	IPNET SOCK ...	ch...	"49152"
Default socket receive buffer size	IPNET SOCK ...	ch...	"100000"
Default socket send buffer size	IPNET SOCK ...	ch...	"100000"
Maximal clone count	IPNET_ROUTE...	ch...	"1000"
Maximum number of sockets	IPNET SOCK ...	uint	1024
Socket API (default)	INCLUDE_SOC...		

5.3 修改 gei0 参数调节 udp 丢包问题

```

logLib.h sysLib.h vxTypesOld.h gei825xxVxbEnd.h test_time.c gpio_test.c
1674#define GEI_JUMBO_MTU 9000
1675#define GEI_CLSIZE 1536
1676#define GEI_NAME "gei"
1677#define GEI_TIMEOUT 10000
1678#define GEI_INTRS (GEI_RXINTRS|GEI_TXINTRS|GEI_LINKINTRS)
1679#define GEI_RXINTRS (GEI_ICR_RXO|GEI_ICR_RXT0)
1680#define GEI_TXINTRS GEI_ICR_TXDW
1681#define GEI_LINKINTRS GEI_ICR_LSC
1682
1683/* The PRO/1000 hardware requires at least 128 descriptors per DMA ring */
1684
1685/****#define GEI_RX_DESC_CNT 128****/
1686/****#define GEI_TX_DESC_CNT 128****/
1687
1688#define GEI_RX_DESC_CNT 256
1689#define GEI_TX_DESC_CNT 256
1690
1691
1692/****#define GEI_DEFAULT_TUPLE_CNT_FACTOR 3*****/
1693#define GEI_DEFAULT_TUPLE_CNT_FACTOR 16
1694
1695#define GEI_TX_MARK_THRESH 8
1696#define GEI_TX_CLEAN_THRESH 32
1697
1698#define GEI_MAXFRAG 16
1699#define GEI_MAX_RX 32
1700
1701
    
```

5.4 修改 i210 网卡百兆调整不自动协商

导入自动协商的例程，目前使用的是 i210 百兆不自动协商，用户的交换机是自协商的，1860 网卡可以正常使用，但是自适应情况下的时候 i210 网卡无法使用，所以需要设置 i210 的网卡设置要求百兆不进行自协商。

5.5 修改 1860 屏蔽 1860 网卡自协商设置成百兆

为了解决交换机导致的 1860 网卡和单片机数据的不通信问题，在 1860 代码里修改程序，设置 1860 不自动协商。

```
1102         ngbPnywrite(pDev, 0x12, 0xa42, value);
1103         ngbPhyInit(pDev);
1104 /***** 为了配合交换机 将1860的网卡改成了 100兆 不要自协商
1105         ngbSetupLinkInternal(pDev, NGB_LINK_SPEED_1GB_FULL |
1106                                 NGB_LINK_SPEED_100_FULL |
1107                                 NGB_LINK_SPEED_10_FULL);
1108 *****/
1109         ngbSetupLinkInternal(pDev, NGB_LINK_SPEED_100_FULL); /*
1110         1
```

6. 操作注意事项

6.1 使用说明

6.1.1 使用前的准备和检查

- NS11029 安放应当平稳，保留必要的通风、工作空间；
- 检查外接线缆完好，连接牢靠；
- NS11029 表面应干净整洁，无灰尘、油污、水渍；
- 拿取和使用主控板时，注意 ESD 防护，设备应进行接地，操作人员应着防静电服装、佩戴静电释放环；

如对板卡使用有任何疑问，请联系凌昊智能®咨询。

6.1.2 开机

将 NS11029 安装到开发底板上，接好电源、显示、键鼠和风扇，检查电源线连接是否可靠，即可上电开机。

6.1.3 关机

NS11029 支持正常关机和断电关机。

- 正常关机：通过点击左下角图标关机按钮或者终端界面输入 “poweroff” 即可进行软

件关机，待设备关机、显示器熄灭且前出 PWR 灯闪烁后，将电源开关断电。

- 断电关机：在应用程序数据做好正常的保存退出后，直接将电源开关断电。

注意：尽量避免直接拔出电源线缆或在应用程序数据收发过程中直接断电等其它可能存在风险的关机操作。

6.1.4 安全保护措施及注意事项

- 绝对避免带电插拔 NS11029；
- 严格执行加、断电规范，确保加、断电顺序的正确性；
- 测试仪器须按使用要求接地；
- 产品按要求做防静电保护；
- NS11029 未安装任何杀毒软件，使用 U 盘前务必先在其它计算机上对 U 盘杀毒；
- 在试验过程中遇到意外情况时，立即直接断电关闭 NS11029。

6.1.5 故障应急处理

- 当 NS11029 出现故障，操作人员应首先给 NS11029 断电；
- 记录故障现象，上报设备管理员；
- 遇到不可解决问题时，应尽快联系凌昊智能®，进行问题分析及排除；
- 在故障排查过程中，不得对 NS11029 进行拆卸处理。

6.2 维护保养

6.2.1 日常维护保养工作

- 关机断电后，应保证开关处于断开状态，避免电源插上后，模块直接上电；
- 用完毕拆除电缆；
- 应定期对 NS11029 表面灰尘进行处理，可用无尘布或酒精擦洗。

6.2.2 长期停放时的维护保养

- 确保 NS11029 断电关机；
- 清理表面灰尘；
- 将清理后的 NS11029 及线缆等配件，均放入包装箱或包装盒。