

用户手册

USER MANUAL

声明

版权所有©北京龙芯科技有限公司。

非经北京龙芯科技有限公司书面同意，任何单位和个人不得擅自摘抄、复制本手册内容的部分或全部，并不得以任何形式传播。

本手册中描述的产品中，可能包含北京龙芯科技有限公司及其可能存在的许可人享有版权的软件，除非获得相关权利人的许可，否则，任何人不能以任何形式对前述软件进行复制、分发、修改、摘录、反编译、反汇编、解密、反向工程、出租、转让、分许可以及其他侵犯软件版权的行为，但是适用法禁止此类限制的除外。

免责声明

笔者和北京龙芯公司尽最大努力编制本手册。然而，笔者与龙芯公司作出任何种类的任何担保，明示或者暗示，有关的信息内容、文件，或在本手册中包含的文件，并不为技术或编辑错误或遗漏承担任何责任。在法律允许的范围内，北京龙芯科技有限公司在任何情况下，都不对因使用本手册相关内容而产生的任何特殊的、附带的、间接的、继发性的损害进行赔偿，也不对任何利润、数据、商誉或预期节约的损失进行赔偿。

安全使用常识

1. 产品使用前，务必仔细阅读产品说明书，以免错误操作带来产品故障；
2. 为避免人体被电击或产品被损坏，在每次对主板、板卡进行拔插或重新配置时，须先关闭交流电源或将交流电源线从电源插座中拔掉；
3. 在需对板卡或整机进行搬动前，务必先将交流或者直流电源线从电源插座中拔掉；
4. 对整机产品，需增加 / 减少板卡时，务必先拔掉交流电源或者直流电源；
5. 当您需连接或拔除任何设备前，须确定所有的电源线事先已被拔掉；
6. 为避免频繁开关机对产品造成不必要的损伤，关机后，应至少等待 30 秒后再开机；
7. 对未准备安装的板卡，应将其保存在防静电保护袋或者防静电箱中；
8. 在从防静电保护袋或防静电箱中拿出板卡前，应将手先置于接地金属物体上一会儿（比如 10 秒钟），以释放身体及手中的静电；
9. 在拿板卡时，需佩戴静电保护手套，并且应该养成只触及其边缘部分的习惯；
10. 佩戴防静电腕带，确保它使皮肤良好接触。
11. 定期检查防静电腕带的电阻值，确保是 1 和 10 兆欧之间。

目录

| | |
|-------------------------|--------|
| 声明..... | II |
| 安全使用常识..... | III |
| 目录..... | IV |
| 图目录..... | VI |
| 表目录..... | VII |
| 1 产品简介..... | - 1 - |
| 1.1 产品概述..... | - 1 - |
| 1.2 COM Express 概述..... | - 1 - |
| 1.3 产品特点..... | - 1 - |
| 1.4 产品规格..... | - 2 - |
| 1.5 产品功能框架..... | - 4 - |
| 1.6 技术服务..... | - 4 - |
| 2 产品结构..... | - 5 - |
| 2.1 外观视图..... | - 5 - |
| 2.2 外形尺寸图..... | - 6 - |
| 2.3 主板丝印规则..... | - 6 - |
| 2.4 结构设计参考..... | - 8 - |
| 3 模块接口定义说明..... | - 10 - |
| 4 快速入门..... | - 15 - |
| 4.1 底板功能列表说明..... | - 15 - |
| 4.2 底板 Pin 针定义..... | - 16 - |
| 5 软件系统开发参考..... | - 20 - |
| 5.1 软件开发支持..... | - 20 - |
| 5.2 内核开发参考..... | - 20 - |
| 5.3 移植应用程序..... | - 21 - |
| 5.4 驱动移植..... | - 25 - |
| 5.5 串口对应设备..... | - 26 - |

| | | |
|-----|--------------------|--------|
| 5.6 | GPIO 对应表..... | - 27 - |
| 6 | 订货信息..... | - 27 - |
| 6.1 | 订货信息..... | - 27 - |
| 6.2 | 出货配套表清单..... | - 27 - |
| 7 | 附录 A 常用指令介绍..... | - 29 - |
| 7.1 | PMON 简介..... | - 29 - |
| 7.2 | 常用 PMON 指令..... | - 29 - |
| 7.3 | 常用 LINUX 网络命令..... | - 31 - |
| 8 | 附录 C 串口调试助手安装..... | - 35 - |

图目录

图 1-1 龙芯 2K1000-COME-系统板 Q 模块功能框图..... - 4 -

图 2-1 模块正面视图..... - 5 -

图 2-2 模块背面视图..... - 5 -

图 2-3 模块外形尺寸图..... - 6 -

图 2-4 模块接口连接器角位图..... - 7 -

图 2-5 2K1000 结构安装示意..... - 8 -

图 8-1 安装文件图示..... - 35 -

图 8-2 TERA TERM PRO 安装配置..... - 35 -

图 8-3 安装 TERA TERM PRO 完成界面..... - 36 -

图 8-4 主机串口信息查看..... - 36 -

图 8-5 启动 TERA TERM PRO 界面..... - 37 -

图 8-6 设置串口波特率..... - 37 -

图 8-7 设置串口波特率..... - 38 -

表目录

| | |
|--------------------------|--------|
| 表 1-1 产品规格表..... | - 2 - |
| 表 3-1 模块接口定义表..... | - 10 - |
| 表 6-1 订货信息表..... | - 27 - |
| 表 6-2 2K1000 出货配套清单..... | - 27 - |

1 产品简介

1.1 产品概述

本产品基于国产化龙芯（loongson） 2K1000 处理器研制的 SOC 单芯片解决方案，完全满足 PICMG COM Express 规范，兼容 COM Express Type10 的 pin 脚定义，尺寸为 Mini Module (84mm x 55mm)，标配板载 DDR3 2GB 内存，最大支持 4GB。该设计采用一个独立的核心模块，用户自行设计底板，方便用户自由扩展板上已有的资源。

模块支持 8 路 PCI Express x 1 通道，1 路 SATA 端口，2 路 CAN 端口，4 路串口，4 路 USB 2.0 端口，1 个千兆网口和 1 个 GMAC，显示部分支持 1 路 DVI, 1 路 24-bit LVDS 信号，模块整体功耗小于 10W。

本产品采用全表贴化设计，具有稳定、安全、可靠、实用性强等特点，可广泛应用于国防、政府、科研、医疗、数控、通讯、交通等领域。

1.2 COM Express 概述

COM Express 是国际工业电气协会 (PICMG) 定义的计算机模块标准，由几大嵌入式工业计算机厂商共同制定的一种计算机模块标准，是一种高度集成的单板计算机，特别适合于执行自定义的工业计算机解决方案。

目前 COM Express 对于模块尺寸规定了四种标准：

| 型号 | 尺寸 |
|----------|-----------|
| Mini | 84x55mm |
| Compact | 95x95mm |
| Basic | 95x125mm |
| Extended | 155x110mm |

1.3 产品特点

- COM-Express Mini 模块，尺寸 84*55mm，兼容 Type 10 标准针脚定义；
- 处理器：龙芯 2K1000 双核处理器，主频 800MHz-1GHz；
- 内存：板载 2GB DDR3 工业级国产内存颗粒；
- 显示：1 路 DVI 接口，1 路 24bit LVDS 接口；
- 网络：板载 1 路千兆网络接口，1 路 GMAC；
- PCI-E 接口：8 路 PCI-E x1 接口，可配置为 2 路 PCI-E x4 或 1 路 PCI-E x4 和 2 路 PCI-E x1；
- SATA 接口：1 路 SATA 接口，支持 SATA 1.5Gbps 和 SATAII3Gbps 传输；
- USB 接口：4 个 USB 接口，兼容 USB1.1 和 USB2.0 规范；
- 音频：AUDIO CODEC 音频编解码器；
- 1 路 SDIO；
- 1 路 SPI；
- RTC 功能支持；
- 4 个 GPI 接口，4 个 GPO 接口；
- 2 路 I²C；
- 12V 电压输入。

1.4 产品规格

表 1-1 产品规格表

| 龙芯 2K1000-COME-系统板 Q 模块规格表 | | |
|----------------------------|-------|-----------------------------------|
| 处理器/芯片组 | CPU | Loongson2K |
| | 核数 | 2 核 |
| | 主频 | 1GHz(工业级主频 800MHz) |
| 内存 | 类型 | 板载 DDR3 |
| | 容量 | 标配 2GB,可选 4GB |
| 存储 | FLASH | 板载 SPI FLASH，支持 PMON 装载 |
| 扩展接口 | USB | 4 路 USB 2.0Host |
| | PCIE | 默认配置为 4 路 PCI-E x1 和 1 路 PCI-E x4 |

| | | |
|-------|------------------|-------------------------------|
| | | 6 路 PCI-E x1 接口或 2 路 PCI-E x4 |
| | GMAC | 1路千兆网络接口，1路GMAC |
| | 串口 | 4 路 TTL 串口 |
| | DVI | 1 路 DVI |
| | LVDS | 1 路 24 bit 单通道 LVDS 接口 |
| | I ² C | 2 路 I ² C |
| | SPI | 1 路 SPI |
| | CAN | 2 路 CAN |
| | Audio | 支持 I2S 音频口 |
| | GPIO | 8 路 GPIO |
| | SATA | 1 路 SATA2.0 接口 |
| 电源 | 类型 | 标准 DC 12V 输入 |
| 物理参数 | 尺寸 (W×D) | 84mm×55mm (模块) |
| 环境适应性 | 常温级 | 工作温度：0℃～55℃，5～95% RH，不凝结 |
| | | 存储温度：-20℃～70℃，5～95% RH，不凝结 |
| | 宽温级 | 工作温度：-20℃～60℃，5～95% RH，不凝结 |
| | | 存储温度：-40℃～75℃，5～95% RH，不凝结 |
| | 工业级 | 工作温度：-40℃～70℃，5～95% RH，不凝结 |
| | | 存储温度：-55℃～80℃，5～95% RH，不凝结 |

1.5 产品功能框架

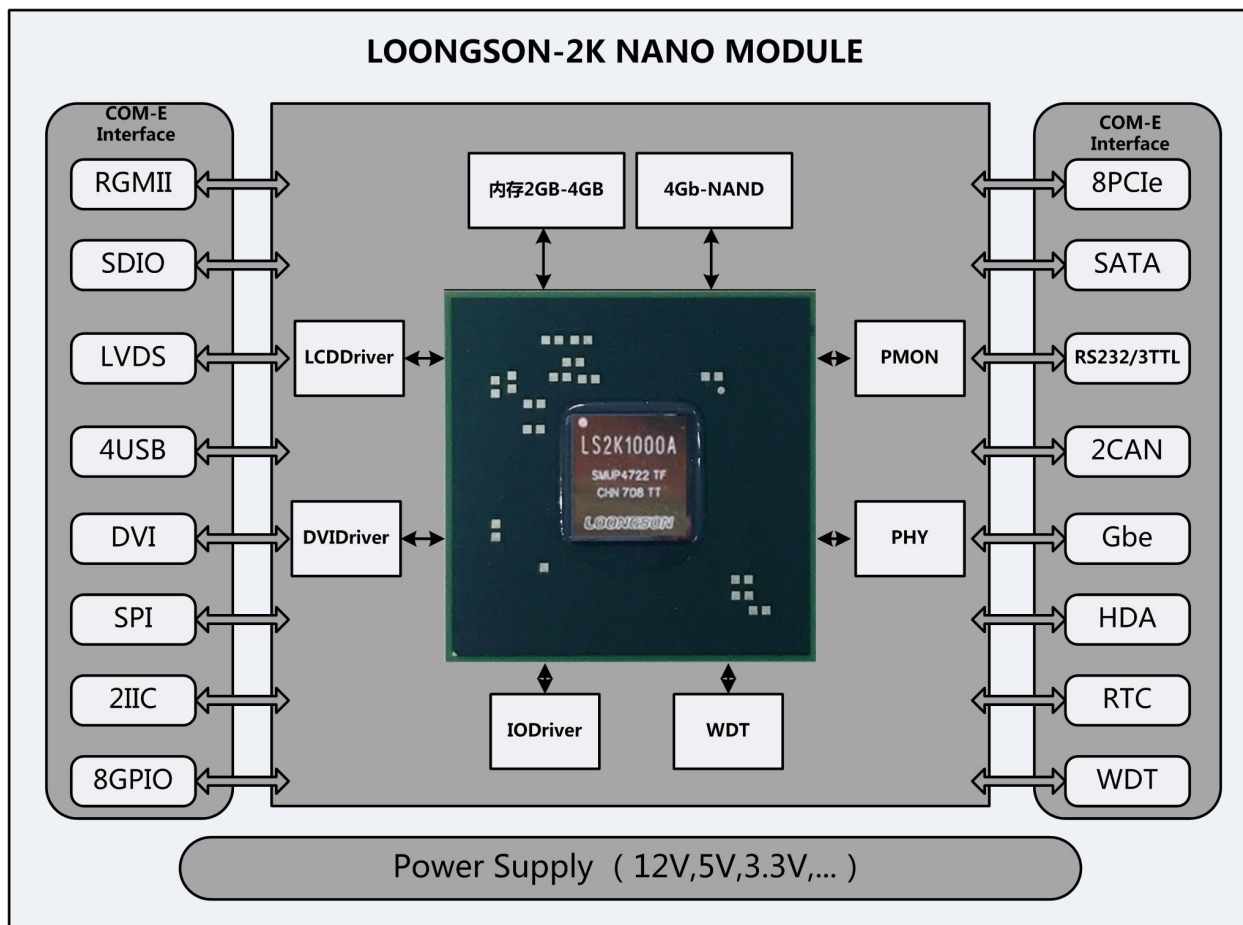


图 1-1 龙芯 2K1000-COME-系统板 Q 模块功能框图

1.6 技术服务

关于产品的任何问题（资料索要、产品规格咨询、产品问题排除等），请直接联系销售团队或者技术团队，以便于及时解决问题。

电话：010-62546668-1609

邮箱：wangjiue@loongson.cn

关于产品出现的情况，我们会先请您确认一下产品手册上的操作及援助，如果还有问题，我们会建议请你提供如下信息：

- 产品出现情况的报告信息，以便进行问题点的定位；
- 问题情况的描述，及你进行的操作分析情况；

- 产品出现情况时的操作步骤是如何？可使我们找到相应差异点；
- 产品出现情况时的软件环境是什么。

2 产品结构

2.1 外观视图

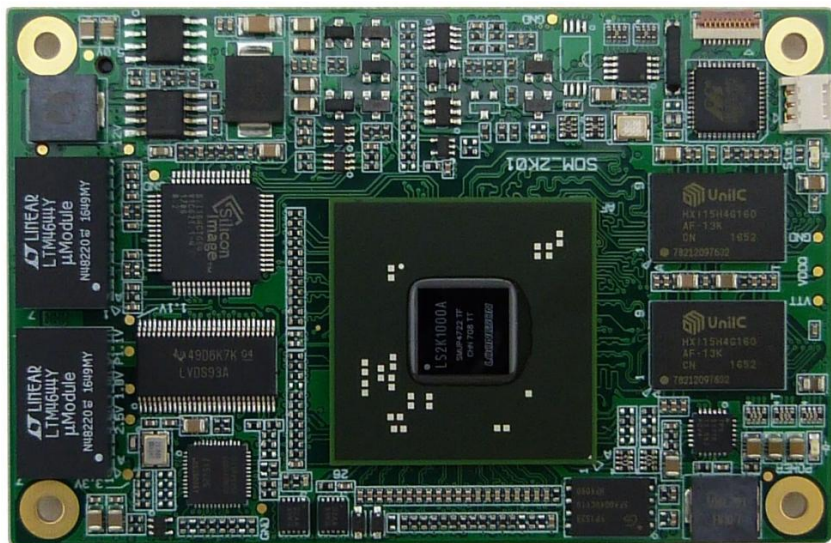


图 2-1 模块正面视图



图 2-2 模块背面视图

2.2 外形尺寸图

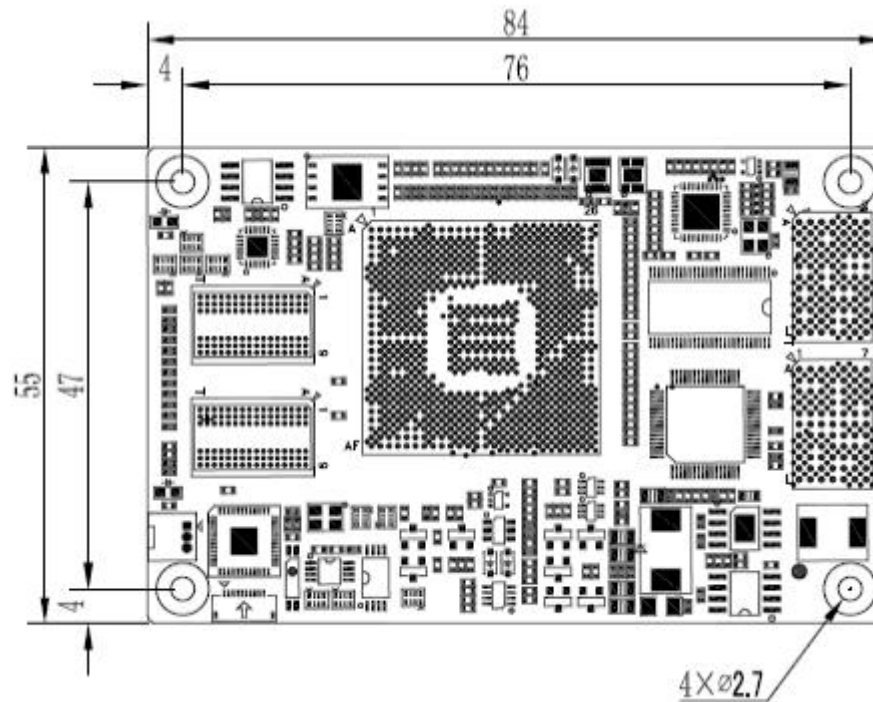


图 2-3 模块外形尺寸图

2.3 主板丝印规则

如何识别跳线、接口第一脚？

- a. 观察插头、插座旁边的文字标记，通常用“1”或加粗的线条或三角符号表示。

b. 看看背面的焊盘，通常方型焊盘为第一脚。

c. 220Pin 板到板连接器 A1/C1 脚如下图位于连接器斜开口处；B1/D1 脚位于 A1/C1 脚对应面。

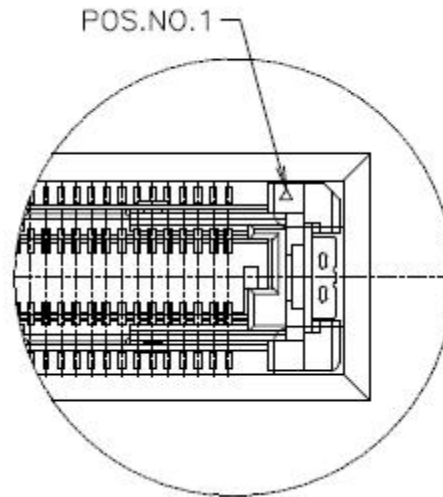


图 2-4 模块接口连接器角位图

2.4 结构设计参考

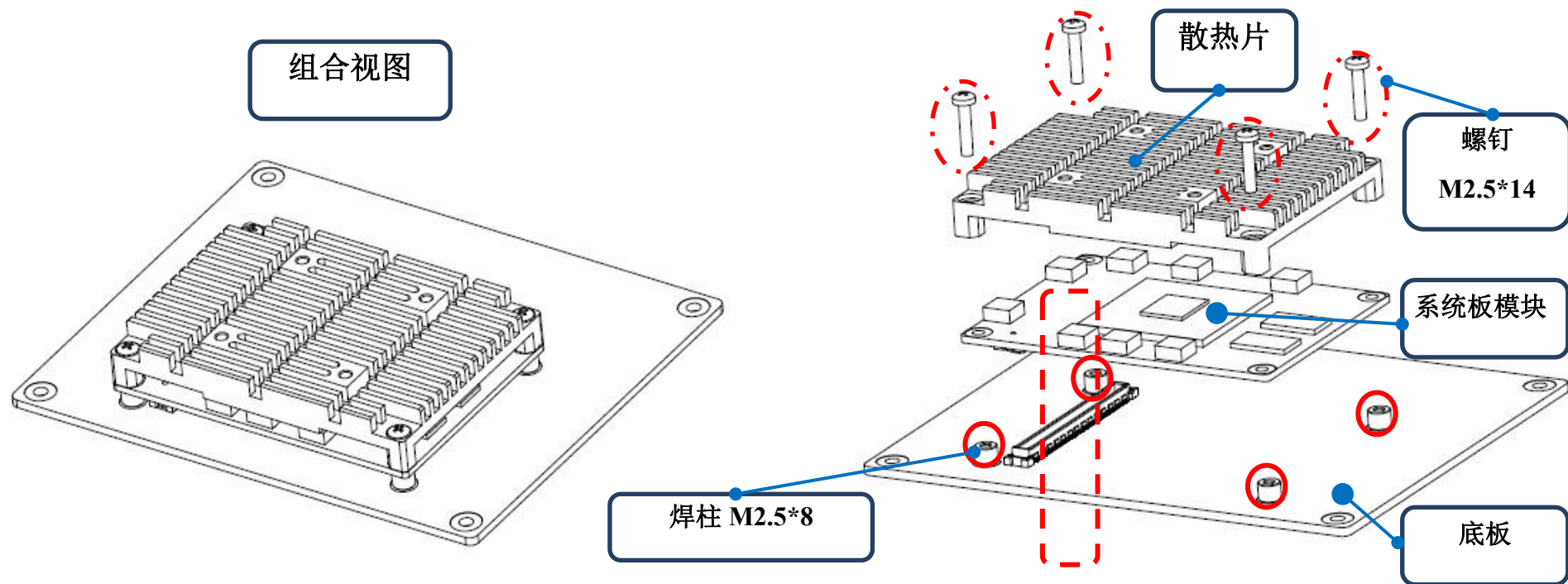


图 2-5 2K1000 结构安装示意

3 模块接口定义说明

龙芯2K1000-COME-系统板Q模块接口采用TYCO-220P高可靠板对板连接器；COM-E接口PIN脚定义如下表：；

表 3-1 模块接口定义表

| 序号 | ROWA | ROWB |
|----|---------------|--------------------|
| 1 | GND (FIXED) | GND (FIXED) |
| 2 | GBE0_P3- | GBE0_LED_ACT# |
| 3 | GBE0_P3+ | GMAC1_TXD0 |
| 4 | GBE0_LED_100# | GMAC1_TXD1 |
| 5 | GBE0_LED_1G# | GMAC1_TXD2 |
| 6 | GBE0_P2- | GMAC1_TXD3 |
| 7 | GBE0_P2+ | GMAC1_RXD0 |
| 8 | NC | GMAC1_RXD1 |
| 9 | GBE0_P1- | GMAC1_RXD2 |
| 10 | GBE0_P1+ | GMAC1_RXD3 |
| 11 | GND (FIXED) | GND (FIXED) |
| 12 | GBE0_P0- | PWRBTN# |
| 13 | GBE0_P0+ | LS2K_IIC1_SCL |
| 14 | NC | LS2K_IIC1_SDA |
| 15 | ACPI_S3N | LS2K_Alert_GPIO01 |
| 16 | SATA0_TX+ | SDIO_DATA0 |
| 17 | SATA0_TX- | SDIO_DATA1 |
| 18 | ACPI_S4N | LS2K_ACPI_SUSSTATN |
| 19 | SATA0_RX+ | SDIO_DATA2 |
| 20 | SATA0_RX- | SDIO_DATA3 |
| 21 | GND (FIXED) | GND (FIXED) |
| 22 | CAN0_TX | LS2K_RXD3 |
| 23 | CAN0_RX | LS2K_TXD3 |

| | | |
|----|------------------|-----------------|
| 24 | ACPI_S5N | LS2K_PWROK |
| 25 | CAN1_TX | GMAC1_MDCK |
| 26 | CAN1_RX | GMAC1_MDIO |
| 27 | BATLOW # | WDT |
| 28 | ATA_ACT# | LS2K_HDA_SDI2 |
| 29 | HDA_SYNC | LS2K_HDA_SDI1 |
| 30 | HDA_RST# | LS2K_HDA_SDI0 |
| 31 | GND (FIXED) | GND (FIXED) |
| 32 | HDA_BITCLK | SPKR |
| 33 | HDA_SDO | LS2K_IIC0_SCL |
| 34 | BIOS_DIS0# | LS2K_IIC0_SDA |
| 35 | THRMTRIP#_GPIO02 | NC |
| 36 | GMAC1_RXCTL | GMAC1_TXCLK |
| 37 | GMAC1_TXCTL | GMAC1_RXCLK |
| 38 | LS2K_TXD0_DEBUG | LS2K_RXD0_DEBUG |
| 39 | PCIE1_TXN3 | PCIE1_RXN3 |
| 40 | PCIE1_TXP3 | PCIE1_RXP3 |
| 41 | GND (FIXED) | GND (FIXED) |
| 42 | LS2K_USB_DM2 | LS2K_USB_DM3 |
| 43 | LS2K_USB_DP2 | LS2K_USB_DP3 |
| 44 | USB_2_3_OC# | USB_0_1_OC# |
| 45 | LS2K_USB_DM0 | LS2K_USB_DM1 |
| 46 | LS2K_USB_DP0 | LS2K_USB_DP1 |
| 47 | VCC_RTC | EXCD1_PERST# |
| 48 | EXCD0_PERST# | EXCD1_CPPE# |
| 49 | EXCD0_CPPE# | SYS_RESET# |
| 50 | NC | CB_RESET# |

| | | |
|----|-------------|----------------|
| 51 | GND (FIXED) | GND (FIXED) |
| 52 | PCIE1_TXP1 | PCIE1_RXP1 |
| 53 | PCIE1_TXN1 | PCIE1_RXN1 |
| 54 | GPI0 | GPO1 |
| 55 | PCIE1_TX0+ | PCIE1_RXP0 |
| 56 | PCIE1_TX0- | PCIE1_RXN0 |
| 57 | GND | GPO2 |
| 58 | PCIE0_TX3+ | PCIE0_RXP3 |
| 59 | PCIE0_TX3- | PCIE0_RXN3 |
| 60 | GND (FIXED) | GND (FIXED) |
| 61 | PCIE0_TX2+ | PCIE0_RXP2 |
| 62 | PCIE0_TX2- | PCIE0_RXN2 |
| 63 | GPI1 | GPO3 |
| 64 | PCIE0_TX1+ | PCIE0_RXP1 |
| 65 | PCIE0_TX1- | PCIE0_RXN1 |
| 66 | GND | WAKE0# |
| 67 | GPI2 | WAKE1# |
| 68 | PCIE0_TX0+ | PCIE0_RXP0 |
| 69 | PCIE0_TX0- | PCIE0_RXN0 |
| 70 | GND (FIXED) | GND (FIXED) |
| 71 | LVDS_A0+ | DVI_TX2+ |
| 72 | LVDS_A0- | DVI_TX2- |
| 73 | LVDS_A1+ | DVI_TX1+ |
| 74 | LVDS_A1- | DVI_TX1- |
| 75 | LVDS_A2+ | DVI_TX0+ |
| 76 | LVDS_A2- | DVI_TX0- |
| 77 | LVDS_DIG_ON | PCIE1_CK_REF0+ |

| | | |
|-----|-----------------|----------------|
| 78 | LVDS_A3+ | PCIE1_CK_REF0- |
| 79 | LVDS_A3- | LVDS_BKLT_EN |
| 80 | GND (FIXED) | GND (FIXED) |
| 81 | LVDS_A_CK+ | DVI_CLK+ |
| 82 | LVDS_A_CK- | DVI_CLK- |
| 83 | LVDS_I2C0_SCL | LS2K_PWM1 |
| 84 | LVDS_I2C0_SDA | VCC_5V_SBY |
| 85 | GPI3 | VCC_5V_SBY |
| 86 | SDIO_CLK | VCC_5V_SBY |
| 87 | DDC_AUX_SEL | VCC_5V_SBY |
| 88 | PCIE0_CLK_REF0+ | BIOS_DISABLE1# |
| 89 | PCIE0_CLK_REF0- | DVI_HPD |
| 90 | GND (FIXED) | GND (FIXED) |
| 91 | SPI_POWER | PCIE1_RX2+ |
| 92 | SPI_SDI | PCIE1_RX2- |
| 93 | GPO0 | PCIE1_TX2+ |
| 94 | SPI_CLK | PCIE1_TX2- |
| 95 | SPI_MOSI | NC |
| 96 | NC | SDIO_CMD |
| 97 | TYPE10# | LS2K_SPI_CSN2 |
| 98 | SER0_TX | DVI_CTRLCLK |
| 99 | SER0_RX | DVI_CTRLDATA |
| 100 | GND (FIXED) | GND (FIXED) |
| 101 | SER1_TX | LS2K_FAN_PWM0 |
| 102 | SER1_RX | NC |
| 103 | LS2K_ACPI_LID | NC |
| 104 | VCC_12V | VCC_12V |

| | | |
|-----|-------------|-------------|
| 105 | VCC_12V | VCC_12V |
| 106 | VCC_12V | VCC_12V |
| 107 | VCC_12V | VCC_12V |
| 108 | VCC_12V | VCC_12V |
| 109 | VCC_12V | VCC_12V |
| 110 | GND (FIXED) | GND (FIXED) |

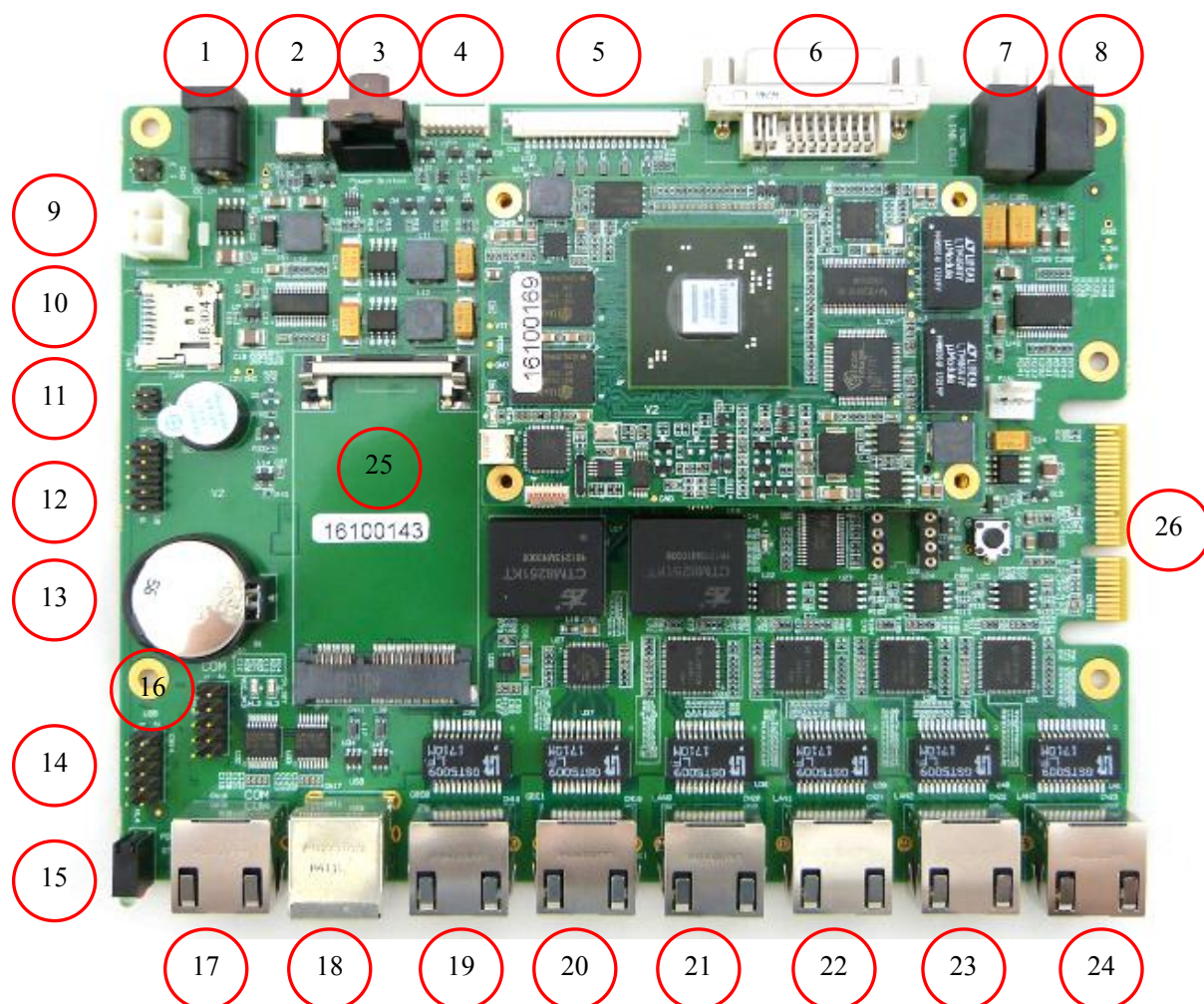
说明:

(1)板上 I2C1 挂载了两个设备, RTC 时钟芯片-地址 0xD0, E2PROM-地址 0xA0.

(2)板上 GPI/GPO 默认是上拉输入

4 快速入门

4.1 底板功能列表说明



注：上电前请务必检查散热器是否安装到位

| 序号 | 位号 | 定义 |
|----|-----|---------|
| 1 | SW1 | DC 输入 |
| 2 | SW2 | 开机模式选择 |
| 3 | SW3 | 上电开关 |
| 4 | CN1 | LVDS 背光 |
| 5 | CN2 | LVDS 信号 |
| 6 | CN4 | DVI |

| | | |
|----|------|-----------|
| 7 | CN25 | LINE OUT |
| 8 | CN24 | MIC |
| 9 | CN6 | DC 输入 |
| 10 | CN7 | TF 卡槽 |
| 11 | CN8 | CAN*2 |
| 12 | CN10 | GPIO*8 |
| 13 | B1 | RTC |
| 14 | CN14 | USB*2 |
| 15 | HL4 | POWER 状态灯 |
| 16 | CN15 | COM*3 |
| 17 | CN16 | COM0 |
| 18 | CN17 | USB*2 |
| 19 | CN18 | Eth4 |
| 20 | CN19 | Eth5 |
| 21 | CN20 | Eth0 |
| 22 | CN21 | Eth1 |
| 23 | CN22 | Eth2 |
| 24 | CN23 | Eth3 |
| 25 | CN11 | Msata 连接器 |
| 26 | CN12 | PCIE 金手指 |

4.2 底板 Pin 针定义

| 功能位号 | 序号 | 定义 | 备注 |
|------|----|----|----|
|------|----|----|----|

| | | | |
|------------------|---|----------------|--|
| CN1 Backlight | 1 | VLED_VCC_12v | |
| | 2 | VLED_VCC_12v | |
| | 3 | VLED_GND | |
| | 4 | VLED_GND | |
| | 5 | LCD_BCK_PWREN | |
| | 6 | LVDS_BKLT_CTRL | |
| | 7 | | |
| | 8 | | |

| 功能位号 | 序号 | 定义 | 备注 |
|-------------|----|-------------|----|
| CN2 LVDS | 1 | LVDS_VCC3V3 | |
| | 2 | LVDS_VCC3V3 | |
| | 3 | GND | |
| | 4 | GND | |
| | 5 | TxOut0- | |
| | 6 | TxOut0+ | |
| | 7 | GND | |
| | 8 | TxOut1- | |
| | 9 | TxOut1+ | |
| | 10 | GND | |
| | 11 | TxOut2- | |
| | 12 | TxOut2+ | |
| | 13 | GND | |
| | 14 | TxCLKOut- | |

| | | | |
|--|----|-----------|--|
| | 15 | TxCLKOut+ | |
| | 16 | GND | |
| | 17 | TxOut3- | |
| | 18 | TxOut3+ | |
| | 19 | LCD_MODE | |
| | 20 | LCD_SCAN | |

| 功能位号 | 序号 | 定义 | 备注 |
|--------------|----|------|----|
| CN6 DC_IN | 1 | GND0 | |
| | 2 | GND1 | |
| | 3 | 12V0 | |
| | 4 | 12V1 | |

| 功能位号 | 序号 | 定义 | 备注 |
|------------|----|-----------|---------------|
| CN8 CAN | 1 | CAN1_CAN+ | 默认含 120 欧终端电阻 |
| | 2 | CAN0_CAN+ | 默认含 120 欧终端电阻 |
| | 3 | CAN1_CAN- | |
| | 4 | CAN0_CAN- | |

| 功能位号 | 序号 | 定义 | 备注 |
|----------------|----|------|----|
| CN10 GPIO*8 | 1 | GPIO | |
| | 2 | GPO0 | |
| | 3 | GPI1 | |
| | 4 | GPO1 | |
| | 5 | GPI2 | |

| | | | |
|--|----|---------|--|
| | 6 | GPO2 | |
| | 7 | GPI3 | |
| | 8 | GPO3 | |
| | 9 | VCC_3V3 | |
| | 10 | GND | |

| 功能位号 | 序号 | 定义 | 备注 |
|-------------|----|------------|----|
| CN14 USB | 1 | USB_PWR0_1 | |
| | 2 | USB_PWR2_3 | |
| | 3 | HOST_USB0- | |
| | 4 | HOST_USB3- | |
| | 5 | HOST_USB0+ | |
| | 6 | HOST_USB3+ | |
| | 7 | GND | |
| | 8 | GND | |
| | 9 | | |
| | 10 | GND | |

| 功能位号 | 序号 | 定义 | 备注 |
|-------------|----|-------------|----|
| CN15 COM | 1 | COM3_TXD232 | |
| | 2 | COM3_RXD232 | |
| | 3 | COM2_TXD232 | |
| | 4 | COM2_RXD232 | |
| | 5 | COM1_TXD232 | |
| | 6 | COM1_RXD232 | |

| | | | |
|--|----|-----|--|
| | 7 | | |
| | 8 | | |
| | 9 | GND | |
| | 10 | GND | |

5 软件系统开发参考

5.1 软件开发支持

开发环境：CentOS 7 x86_64 系统

开发方式：交叉编译

开发软件：linux 内核源码-linux-3.10.0-el7.tar.gz

交叉编译器：gcc-4.9.3-gnu.tar.gz

注：实例软件及驱动源码（包含于开发光盘中）

5.2 内核开发参考

1. 将 linux 内核源码和交叉编译器拷贝到开发主机 linux 系统上并解压

```
tar zxvf linux-3.10.0-el7.tar.gz -C /opt
```

```
tar zxvf gcc-4.9.3-64-gnu.tar.gz -C /opt
```

2. 设置环境变量

```
export PATH=/opt/gcc-4.9.3-64-gnu/bin:$PATH
```

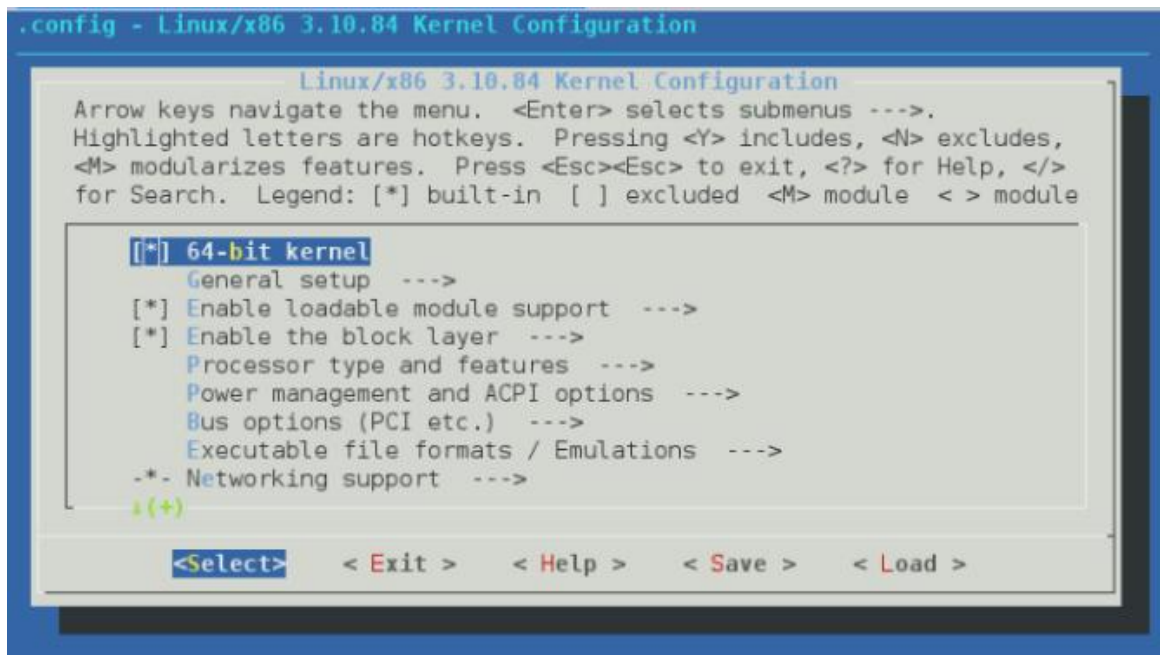
```
export MAKEFLAGS='CC=mips64el-linux-gcc\ -g\ -march=loongson3a'
```

3. 配置内核参数

```
cd /opt/linux-3.10.0-el7/
```

```
make menuconfig (要求系统已经安装 ncurses)
```

在下图所示的对话框中定义内核



4. 编译内核

make ARCH=mips CROSS_COMPILE=mips64el-linux-

```
[root@localhost linux-3.10.0-el7]# make ARCH=mips CROSS_COMPILE=mips64el-linux-
CHK      include/generated/uapi/linux/version.h
CHK      include/generated/utsrelease.h
HOSTCC   scripts/basic/fixdep
Checking missing-syscalls for N32
CC        kernel/bounds.s
GEN       include/generated/bounds.h
CC        arch/mips/kernel/asm-offsets.s
GEN       include/generated/asm-offsets.h
CALL      scripts/checksyscalls.sh
Checking missing-syscalls for O32
CALL      scripts/checksyscalls.sh
CALL      scripts/checksyscalls.sh
HOSTCC    scripts/genksyms/genksyms.o
HOSTCC    scripts/genksyms/lex.lex.o
HOSTCC    scripts/genksyms/parse.tab.o
HOSTLD    scripts/genksyms/genksyms
CC        scripts/mod/empty.o
HOSTCC    scripts/mod/mk_elfconfig
MKELF     scripts/mod/elfconfig.h
CC        scripts/mod/devicetable-offsets.s
GEN       scripts/mod/devicetable-offsets.h
HOSTCC    scripts/mod/file2alias.o
```

5. 生成的 vmlinux 文件就是更新的内核文件

5.3 移植应用程序

A. 项目程序：以 can 测试程序 can-utils 为例

1. git clone <https://github.com/linux-can/can-utils.git>

-
2. `cd can-utils`
 3. `./autogen.sh`
 4. `./configure --host=mips64el-linux --prefix=/opt/can`

```
[root@localhost can-utils]# ./configure --host=mips64el-linux --prefix=/opt/can
checking build system type... x86_64-unknown-linux-gnu
checking host system type... mips64el-unknown-linux-gnu
checking for mips64el-linux-gcc... mips64el-linux-gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... yes
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether mips64el-linux-gcc accepts -g... yes
checking for mips64el-linux-gcc option to accept ISO C89... none needed
checking how to print strings... printf
checking for a sed that does not truncate output... /usr/bin/sed
checking for grep that handles long lines and -e... /usr/bin/grep
checking for egrep... /usr/bin/grep -E
checking for fgrep... /usr/bin/grep -F
checking for ld used by mips64el-linux-gcc... /opt/gcc-4.9.3-64-gnu/mips64el-linux
/bin/ld
checking if the linker (/opt/gcc-4.9.3-64-gnu/mips64el-linux/bin/ld) is GNU ld...
yes
checking for BSD- or MS-compatible name lister (nm)... /opt/gcc-4.9.3-64-gnu/bin/m
ips64el-linux-nm
D
```

5. `make && make install`

```
checking for gettimeofday... yes
checking for localtime_r... yes
checking for memset... yes
checking for select... yes
checking for setlocale... yes
checking for socket... yes
checking for strchr... yes
checking for strerror... yes
checking for strstr... yes
checking for strtoul... yes
checking for library containing clock_nanosleep... none required
checking whether SO_RXQ_OVFL is declared... no
checking whether PF_CAN is declared... no
checking whether AF_CAN is declared... no
checking whether N_SLCAN is declared... no
checking whether to enable debugging... no
checking that generated files are newer than configure... done
configure: creating ./config.status
config.status: creating GNUmakefile
config.status: executing libtool commands
config.status: executing depfiles commands
[root@localhost can-utils]#
[root@localhost can-utils]# make &&make install
```

6. `cd /opt/can/bin` 检查生成的 candump 格式, file candump, 将程序拷贝到 2k 开发板上

```
[root@localhost bin]# ls
asc2log          canfdtest        cansend          isotpsend        log2long
bcmserver        cangen          cansniffer       isotpsniffer     slcan_attach
canbusload       cangw           isotpdump        isotpsniffer     slcand
can-calc-bit-timing canlogserver    isotpperf       isotptun         slcanpty
candump          canplayer       isotprecv        log2asc
[root@localhost bin]# file can
canbusload      cangen          cansend
can-calc-bit-timing cangw          cansniffer
candump         canlogserver
canfdtest       canplayer
[root@localhost bin]# file candump
candump: ELF 64-bit LSB executable, MIPS, MIPS64 rel2 version 1 (SYSV), dynamically
linked (uses shared libs), for GNU/Linux 2.6.32, with unknown capability 0x756e6
7000000f41 = 0x104000000070100, not stripped
```

7. 程序运行如下图

```
# ./candump
```

```
Usage: candump [options] <CAN interface>+
       (use CTRL-C to terminate candump)
```

```
Options: -t <type>    (timestamp: (a)bsolute/(d)elta/(z)ero/(A)bsolute w date)
          -c          (increment color mode level)
          -i          (binary output - may exceed 80 chars/line)
          -a          (enable additional ASCII output)
          -S          (swap byte order in printed CAN data[] - marked with ``')
          -s <level>  (silent mode - 0: off (default) 1: animation 2: silent)
          -b <can>    (bridge mode - send received frames to <can>)
          -B <can>    (bridge mode - like '-b' with disabled loopback)
          -u <usecs>  (delay bridge forwarding by <usecs> microseconds)
          -l          (log CAN-frames into file. Sets '-s 2' by default)
          -L          (use log file format on stdout)
          -n <count>  (terminate after reception of <count> CAN frames)
          -r <size>   (set socket receive buffer to <size>)
          -D          (Don't exit if a "detected" can device goes down.
          -d          (monitor dropped CAN frames)
          -e          (dump CAN error frames in human-readable format)
          -x          (print extra message infos, rx/tx brs esi)
          -T <msecs>  (terminate after <msecs> without any reception)
```

Up to 16 CAN interfaces with optional filter sets can be specified on the commandline in the form: <ifname>[,filter]*

Comma separated filters can be specified for each given CAN interface:

```
<can_id>:<can_mask> (matches when <received_can_id> & mask == can_id & mask)
<can_id>~<can_mask> (matches when <received_can_id> & mask != can_id & mask)
#<error_mask>      (set error frame filter, see include/linux/can/error.h)
[j|J]              (join the given CAN filters - logical AND semantic)
```

B. 编译单文件，以内存测试 stream 为例

1. wget <http://ftp.loongnix.org/toolchain/testtool/memory/stream.c>
2. mips64el-linux-gcc stream.c -o stream
3. 将生成的可执行应用程序在 2k 上运行，运行结果如下图：


```

-----
# ./stream
-----
STREAM version $Revision: 5.9 $
-----
This system uses 8 bytes per DOUBLE PRECISION word.
-----
Array size = 2000000, Offset = 0
Total memory required = 45.8 MB.
Each test is run 10 times, but only
the *best* time for each is used.
-----
Printing one line per active thread....
-----
Your clock granularity/precision appears to be 1 microseconds.
Each test below will take on the order of 40931 microseconds.
    (= 40931 clock ticks)
Increase the size of the arrays if this shows that
you are not getting at least 20 clock ticks per test.
-----
WARNING -- The above is only a rough guideline.
For best results, please be sure you know the
precision of your system timer.
-----
Function      Rate (MB/s)    Avg time    Min time    Max time
Copy:         878.6140      0.0367      0.0364      0.0373
Scale:        598.5336      0.0538      0.0535      0.0540
Add:          742.5743      0.0647      0.0646      0.0649
Triad:        541.7118      0.0887      0.0886      0.0888
-----
Solution Validates

```

4. 静态编译: `mips64el-linux-gcc stream.c -o stream -static`
5. 将生成的可执行程序在 2k 上运行, 运行结果如下:

```
# ./stream-static
-----
STREAM version $Revision: 5.9 $
-----
This system uses 8 bytes per DOUBLE PRECISION word.
-----
Array size = 2000000, Offset = 0
Total memory required = 45.8 MB.
Each test is run 10 times, but only
the *best* time for each is used.
-----
Printing one line per active thread....
-----
Your clock granularity/precision appears to be 1 microseconds.
Each test below will take on the order of 40500 microseconds.
    (= 40500 clock ticks)
Increase the size of the arrays if this shows that
you are not getting at least 20 clock ticks per test.
-----
WARNING -- The above is only a rough guideline.
For best results, please be sure you know the
precision of your system timer.
-----
Function      Rate (MB/s)    Avg time    Min time    Max time
Copy:         879.0243      0.0366      0.0364      0.0369
Scale:        606.5891      0.0532      0.0528      0.0535
Add:          746.1527      0.0644      0.0643      0.0647
Triad:        544.4831      0.0883      0.0882      0.0884
-----
Solution Validates
-----
```

6. 查看两个可执行程序区别：可以看到一个是用动态链接库，另一个是静态链接

```
[root@localhost can]# mips64el-linux-gcc stream.c -o stream
[root@localhost can]# mips64el-linux-gcc stream.c -o stream-static -static
[root@localhost can]# file stream stream-static
stream:      ELF 64-bit LSB executable, MIPS, MIPS64 rel2 version 1 (SYSV), dyna
mically linked (uses shared libs), for GNU/Linux 2.6.32, with unknown capability 0
x756e67000000f41 = 0x10400000070100, not stripped
stream-static: ELF 64-bit LSB executable, MIPS, MIPS64 rel2 version 1 (SYSV), stat
ically linked, for GNU/Linux 2.6.32, with unknown capability 0x756e67000000f41 =
0x10400000070100, not_stripped
```

5.4 驱动移植

1. 将驱动文件 helloworld.c 和 Makefile 拷贝到主机

2. 编译驱动 `make -k -C /opt/linux-3.10.0-el7/ SUBDIRS=$PWD ARCH=mips CROSS_COMPILE=mips64el-linux- modules`

```
[root@localhost hello]# make -k -C /opt/2k/linux-3.10.0-el7/ SUBDIRS=$PWD ARCH=mips CROSS_COMPILE=mips64el-linux- modules
make: Entering directory `/opt/2k/linux-3.10.0-el7'
CC [M] /home/zc/can/hello/helloworld.o
Building modules, stage 2.
MODPOST 1 modules
CC /home/zc/can/hello/helloworld.mod.o
LD [M] /home/zc/can/hello/helloworld.ko
make: Leaving directory `/opt/2k/linux-3.10.0-el7'
```

3. `modinfo helloworld.ko` 查看驱动信息

```
[root@localhost hello]# modinfo helloworld.ko
filename:      /home/zc/can/hello/helloworld.ko
author:        allgo
license:       GPL
depends:
vermagic:      3.10.84+ SMP mod_unload modversions LOONGSON2K 64BIT
```

4. 拷贝到 2k 主板上运行

执行 `insmod helloworld.ko` , `lsmod` 查看模块挂载信息

```
# lsmod
helloworld 809 0 - Live 0xffffffffc0000000 (0)
# dmesg |tail -n 10
[ 24.546875] sd 1:0:0:0: [sda] 15630336 512-byte logical blocks: (8.00 GB/7.45 GiB)
[ 24.550781] sd 1:0:0:0: [sda] Write Protect is off
[ 24.550781] sd 1:0:0:0: [sda] Mode Sense: 43 00 00 00
[ 24.550781] sd 1:0:0:0: [sda] Write cache: disabled, read cache: enabled, doesn't support DPO or FUA
[ 24.562500] sda: sda1
[ 24.566406] sd 1:0:0:0: [sda] Attached SCSI removable disk
[ 26.085937] FAT-fs (sda1): Volume was not properly unmounted. Some data may be corrupt. Please run fsck.
[ 320.585937] calling hellokernel_init+0x0/0x2c [helloworld] @ 124
[ 320.589843] Hello kernel!
[ 320.593750] initcall hellokernel_init+0x0/0x2c [helloworld] returned 0 after 4105 usecs
#
```

5.5串口对应设备

| 模块 PIN | 底板[CN15]&[CN16] | Linux 内核下设备名称 |
|-----------------|-----------------|---------------|
| SER0_TX | COM1_TXD232 | ttyS5 |
| SER0_RX | COM1_RXD232 | |
| SER1_TX | COM2_TXD232 | ttyS3 |
| SER1_RX | COM2_RXD232 | |
| LS2K_TXD3 | COM3_TXD232 | ttyS4 |
| LS2K_RXD3 | COM3_RXD232 | |
| LS2K_TXD0_DEBUG | COM0_TXD232 | ttyS0 |
| LS2K_RXD3_DEBUG | COM0_RXD232 | |

5.6GPIO 对应表

| 模块 PIN | 底板[CN10] |
|--------|----------|
| GPI_0 | GPIO63 |
| GPI_1 | GPIO62 |
| GPI_2 | GPIO61 |
| GPI_3 | GPIO60 |
| GPO_0 | GPIO59 |
| GPO_1 | GPIO58 |
| GPO_2 | GPIO57 |
| GPO_3 | GPIO56 |

6 订货信息

6.1 订货信息

表 6-1 订货信息表

| 序号 | 订货型号 | 规格描述 | 备注 |
|----|-----------------------------------|---|----|
| | 龙芯 2K1000-COME-系 统板 Q | 常温版：主频 1GHz 内存 2GB,工作温度：0° C~+55° C | |
| | 龙芯 2K1000-COME-系 统板 Q-i(内存 4G) | 工业版：主频 800MHz 内存 4GB,工作温度：-40° C~+70° C | |
| | 龙芯 2K1000-COME-系 统板-底板 Q | 底板，6 个网口，4 个 USB，4 个串口，1 路 LVDS，1 路 DVI | |

6.2 出货配套表清单

订购 2K1000 模块，出货配套清单如下表，清单外物料需要单独订购说明，直接联系龙芯相关业务人员；

表 6-2 2K1000 出货配套清单

| 序号 | 品名 | 数量 | 单位 | 备注 |
|----|-------------------------------|----|----|----|
| 1 | 龙芯 2K1000-COME-系统板 Q 模块 | 1 | 个 | 标配 |
| 2 | 十字槽半圆头螺钉 公制 M2. 5*14mm 含弹簧和垫片 | 4 | 个 | 选配 |

| | | | | |
|---|--------------------------|---|---|----|
| 3 | 散热器（带风扇） | 1 | 个 | 选配 |
| 4 | 龙芯 2K1000-COME-系统板-底板 Q | 1 | 个 | 选配 |
| 5 | 配套光盘（用户手册、软件开发、底板设计辅助资料） | 1 | 个 | 标配 |
| 6 | 装箱清单 | 1 | 张 | 标配 |

7 附录 A 常用指令介绍

7.1 PMON 简介

PMON 是一个兼有 BIOS 和 boot loader 部分功能的开放源码软件，多用于嵌入式系统。基于龙芯的系统采用 PMON 作为类 BIOS 兼 bootloader，并在其基础上做了很多完善工作，支持 BIOS 启动配置，内核加载，程序调试，内存寄存器显示、设置以及内存反汇编等。

7.2 常用 PMON 指令

a. 查看设备

命令: devls

b. 配置 ip 地址

命令: ifaddr syn0 xxx.xxx.xxx.xxx (syn0 为用 devls 查看得到的网卡名字)

c. 简单测试网络

命令: ping other-ip (other-ip 为在同一网段其他 PC 机器 IP)

d. pmon 下通过网络加载内核到内存

命令: load tftp://tftp-server-ip/vmlinux

注: 此命令正常执行会打印加载信息

e. PMON 下通过网络烧写更新替换新 pmon

命令: load -r -f 0xbfc00000 tftp://tftp-server-ip/gzrom.bin

注: 0xbfc00000 为龙芯处理器启动的第一条指令地址 (映射到 Spiflash 中, 即从 Spiflash 启动。

gzrom.bin 为 pmon 的二进制名字。

f. 烧写内核到 nandflash (通过网络)

命令: devcp tftp://tftp-server-ip/vmlinux /dev/mtd0

注: 这里也可以是 /dev/mtd1 或者 /dev/mtd2, 分别烧写到不同分区, 需要在内核加载参数中设置不同的内核启动位置。

g. 从 nandflash 中加载内核

命令: load /dev/mtd0

注: /dev/mtd0 需要根据内核的实际位置来修改 (可能是 /dev/mtd1 /dev/mtd2 等)。

h. 内核添加参数启动

命令: g console=ttyS0,115200 rdinit=/sbin/init

注：使用方式是 g 后面跟内核参数，例子中的参数根据实际的使用来修改。

i. 设置内核启动加载位置环境变量

命令：set al /dev/mtd0

注：/dev/mtd0 需要根据内核的实际位置来修改（可能是/dev/mtd1 /dev/mtd2 等）。

PMON 启动后会自动根据 al（autoload）的值来自动加载内核。

j. 设置内核启动参数的环境变量

命令：set append “console=ttyS0,115200 rdinit=/sbin/init”

注：内核启动参数根据实际情况修改。

PMON 在启动后根据 al 的值自动加载内核，而后自动用 append 的值做为内核启动参数来启动内核系统。

k. 设置延时启动时间环境变量

命令：set bootdelay 3

注：单位为秒（3：为延时 3 秒）

PMON 启动后根据 bootdelay 的值来延时加载启动系统内核。

l. 重启 PMON

命令：reboot

m. 擦除 nandflash 第一个分区（跳过坏块）

命令：mtd_erase /dev/mtd0

mtd_erase /dev/mtd0b

mtd_erase /dev/mtd0c

mtd_erase /dev/mtd0y

n. 擦除 nandflash 第一个分区（擦除所有块包括坏块）

命令：mtd_erase /dev/mtd0r

o. 设置 NANDFlash 分区（分三个分区第一个大小 10m，属性只读，名字是 kernel，第二个分区大小 40m，属性可读可写，名字是 rootfs，第三分区大小剩下的所有空间，属性可读可写，名字是 other）

命令：set mtdparts nand-flash:10m(kernel)ro,40m(rootfs),-(other)

设置后需要重启生效命令：reboot

p. 查看当前 NANDFlash 分区信息

命令：mtdparts

q. 恢复 NANDFlash 默认值分区

命令: `unset mtdparts`

7.3 常用 LINUX 网络命令

a. Ifconfig 指令

1.作用

`ifconfig` 用于查看和更改网络接口的地址和参数, 包括 IP 地址、网络掩码、广播地址, 使用权限是超级用户。

2.格式

`ifconfig -interface [options] address`

3.主要参数

-interface: 指定的网络接口名, 如 `eth0` 和 `eth1`。

-up: 激活指定的网络接口卡。

-down: 关闭指定的网络接口。

-broadcast address: 设置接口的广播地址。

-pointopoint: 启用点对点方式。

-address: 设置指定接口设备的 IP 地址。

-netmask address: 设置接口的子网掩码。

4.应用说明

`ifconfig` 是用来设置和配置网卡的命令行工具。为了手工配置网络, 这是一个必须掌握的命令。使用该命令的好处是无须重新启动机器。要赋给 `eth0` 接口 IP 地址 207.164.186.2, 并且马上激活它, 使用下面命令:

```
#ifconfig eth0 210.34.6.89 netmask 255.255.255.128 broadcast 210.34.6.127
```

该命令的作用是设置网卡 `eth0` 的 IP 地址、网络掩码和网络的本地广播地址。若运行不带任何参数的 `ifconfig` 命令, 这个命令将显示机器所有激活接口的信息。带有“-a”参数的命令则显示所有接口的信息, 包括没有激活的接口。注意, 用 `ifconfig` 命令配置的网络设备参数, 机器重新启动以后将会丢失。

如果要暂停某个网络接口的工作, 可以使用 `down` 参数:

```
#ifconfig eth0 down
```

b. Ping 指令

1.作用

ping 检测主机网络接口状态，使用权限是所有用户。

2.格式

ping [-dfnqrRv][[-c][[-i][[-I][[-l][[-p][[-s][[-t] IP 地址

3.主要参数

- d: 使用 Socket 的 SO_DEBUG 功能。
- c: 设置完成要求回应的次数。
- f: 极限检测。
- i: 指定收发信息的间隔秒数。
- I: 网络界面使用指定的网络界面送出数据包。
- l: 前置载入，设置在送出要求信息之前，先行发出的数据包。
- n: 只输出数值。
- p: 设置填满数据包的范本样式。
- q: 不显示指令执行过程，开头和结尾的相关信息除外。
- r: 忽略普通的 Routing Table，直接将数据包送到远端主机上。
- R: 记录路由过程。
- s: 设置数据包的大小。
- t: 设置存活数值 TTL 的大小。
- v: 详细显示指令的执行过程。

c. Netstat 指令

1.作用

检查整个 Linux 网络状态。

2.格式

netstat [-acCeFghilMnNoprstuvVwx][[-A][--ip]

3.主要参数

- a--all: 显示所有连线中的 Socket。
- A: 列出该网络类型连线中的 IP 相关地址和网络类型。
- c--continuous: 持续列出网络状态。
- C--cache: 显示路由器配置的快取信息。
- e--extend: 显示网络其它相关信息。

-F--fib: 显示 FIB。

-g--groups: 显示多重广播功能群组组员名单。

-h--help: 在线帮助。

-i--interfaces: 显示网络界面信息表单。

-l--listening: 显示监控中的服务器的 Socket。

-M--masquerade: 显示伪装的网络连线。

-n--numeric: 直接使用 IP 地址，而不通过域名服务器。

-N--netlink--symbolic: 显示网络硬件外围设备的符号连接名称。

-o--timers: 显示计时器。

-p--programs: 显示正在使用 Socket 的程序识别码和程序名称。

-r--route: 显示 Routing Table。

-s--statistic: 显示网络工作信息统计表。

-t--tcp: 显示 TCP 传输协议的连线状况。

-u--udp: 显示 UDP 传输协议的连线状况。

-v--verbose: 显示指令执行过程。

-V--version: 显示版本信息。

-w--raw: 显示 RAW 传输协议的连线状况。

-x--unix: 和指定“-A unix”参数相同。

--ip--inet: 和指定“-A inet”参数相同。

d. Route 指令

1.作用

route 表示手工产生、修改和查看路由表。

2.格式

```
#route [-add][[-net|-host] targetaddress [-netmask Nm][dev]If]
```

```
#route [-delete][[-net|-host] targetaddress [gwGw] [-netmask Nm] [dev]If]
```

3.主要参数

-add: 增加路由。

-delete: 删除路由。

-net: 路由到达的是一个网络，而不是一台主机。

-host: 路由到达的是一台主机。

-netmask Nm: 指定路由的子网掩码。

gw: 指定路由的网关。

[dev]If: 强迫路由链指定接口。

4.应用实例

route 命令是用来查看和设置 Linux 系统的路由信息，以实现与其它网络的通信。要实现两个不同的子网之间的通信，需要一台连接两个网络的路由器，或者同时位于两个网络的网关来实现。

在 Linux 系统中，设置路由通常是为了解决以下问题：该 Linux 系统在一个局域网中，局域网中有一个网关，能够让机器访问 Internet，那么就需要将这台机器的 IP 地址设置为 Linux 机器的默认路由。使用下面命令可以增加一个默认路由：

示例：route add 0.0.0.0 192.168.

8 附录 C 串口调试助手安装

为了通过串口连接开发板，需要在 PC 机上使用一个终端仿真程序。Windows XP 上自带超级终端软件，位于“开始-->程序-->附件-->通讯”，而 Windows 7 上需要额外安装，这里使用的软件是 Tera Term Pro，可以在网上自行下载。

- a. 打开下载并解压好的 Tera Term Pro 的文件夹，选择 Setup.exe 文件进行安装。

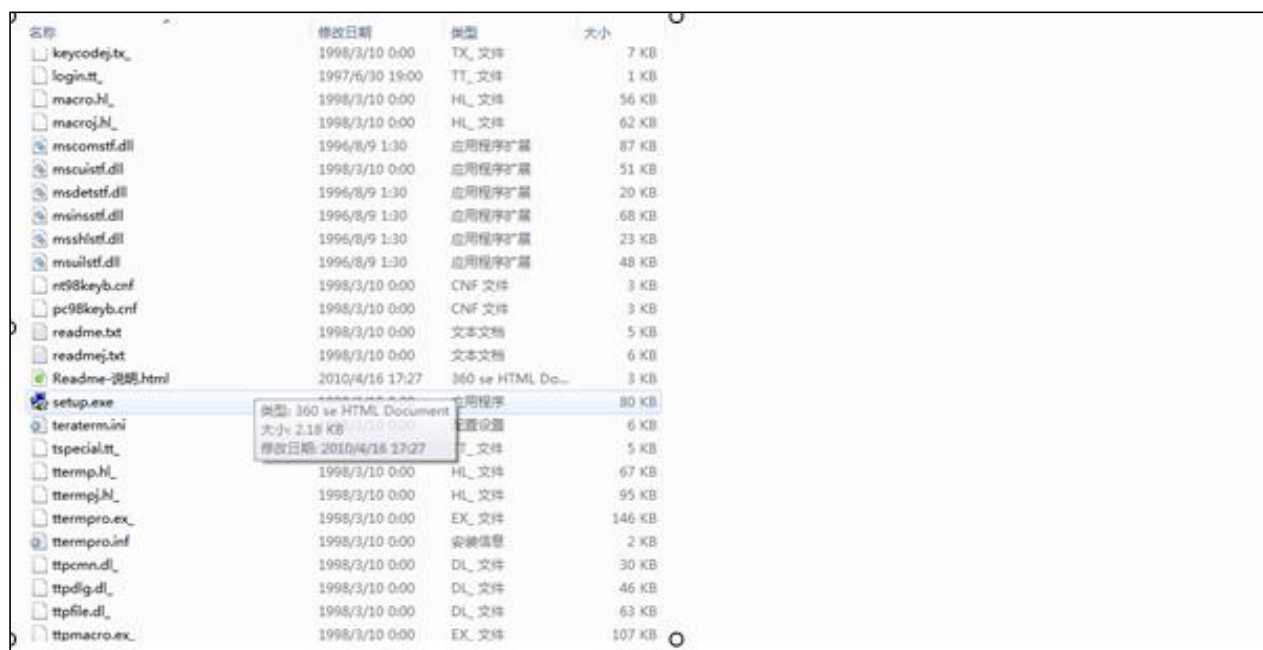


图 8-1 安装文件图示

- b. 进行安装配置

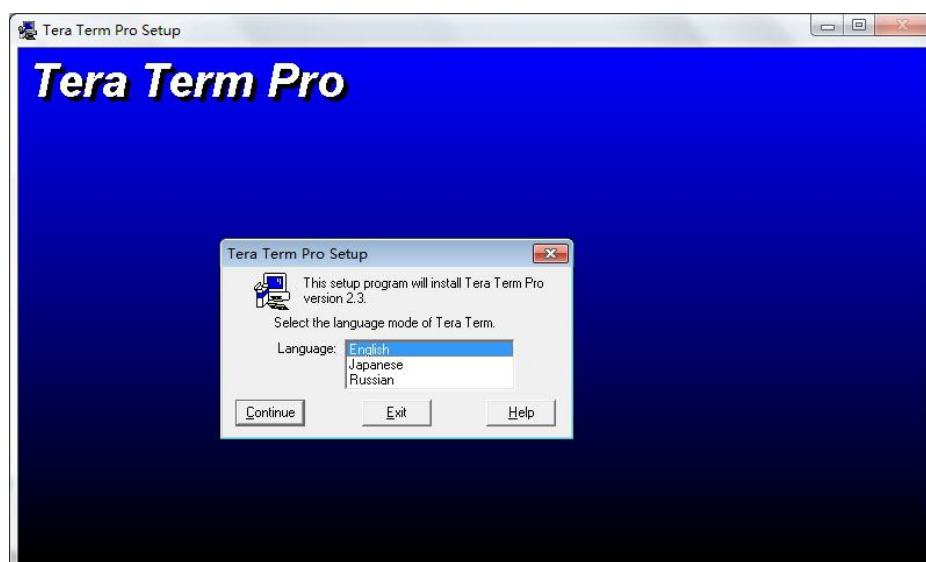


图 8-2 Tera Term Pro 安装配置

- c. 安装完毕后弹出对话框，表示安装成功

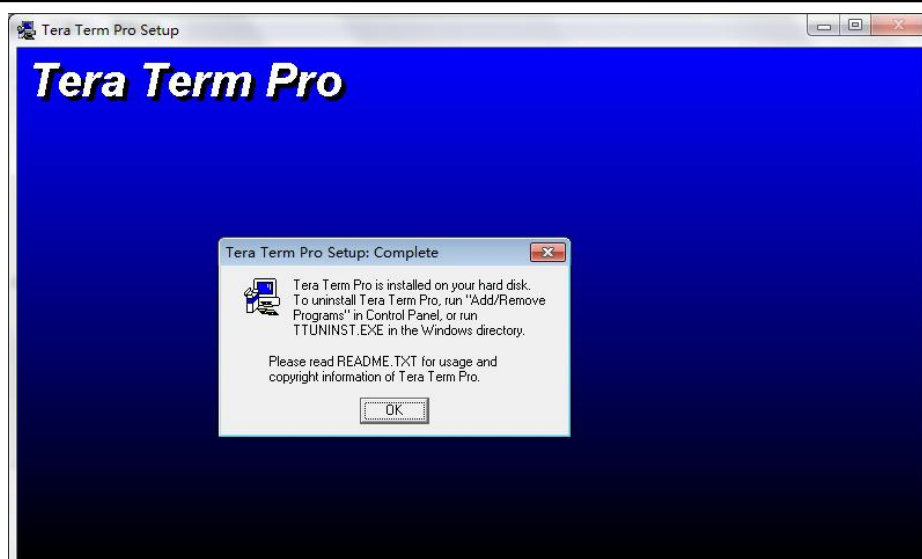


图 8-3 安装 Tera Term Pro 完成界面

- d. 查看主机的串口号（如果笔记本没有串口需使用 USB 转串口工具，只有插入 USB 转串口工具并安装硬件驱动后才能查看串口号）主机串口号在设备管理器中可以查看，如下图：

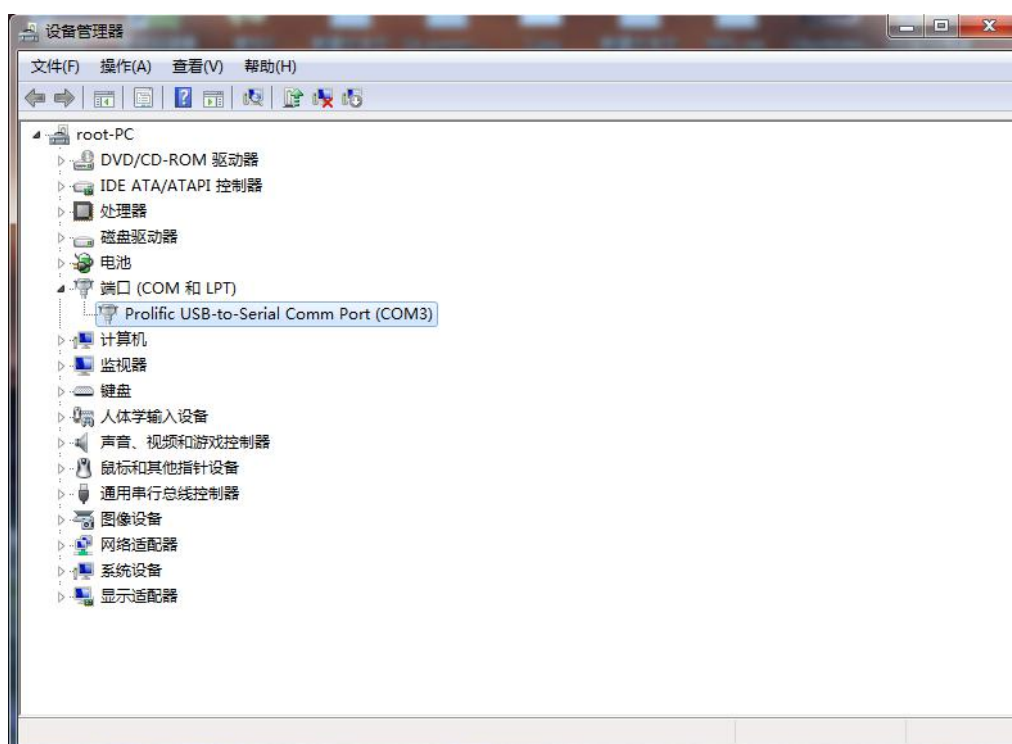


图 8-4 主机串口信息查看

用串口线连接主机和开发板，并打开 Tera Term Prom，这里选择 Serial(串口)模式，在 Port 下拉菜单中选择主机对应的串口号，

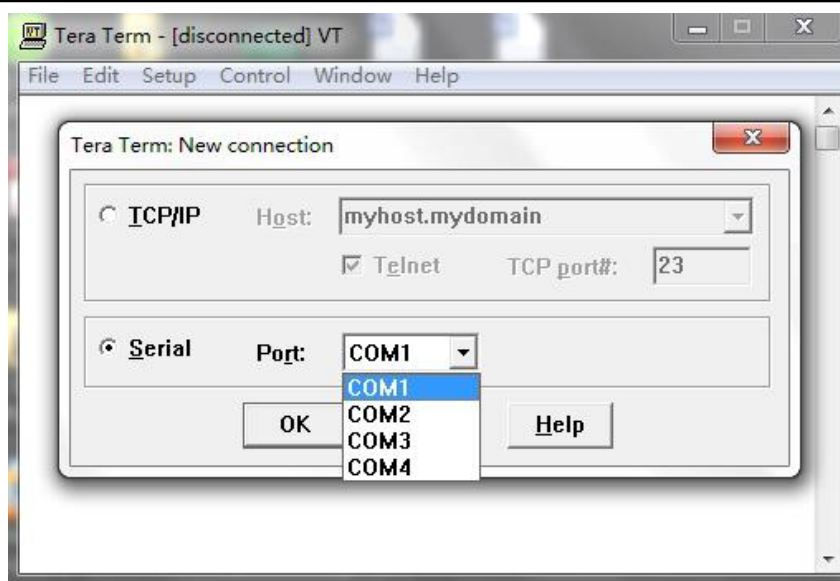


图 8-5 启动 Tera Term Pro 界面

- e. 设置通讯波特率，即模拟线路信号的速率，以波形每秒的振荡数来衡量。如果数据不压缩，波特率等于每秒钟传输的数据位数，在菜单栏中的 Setup 中的 Serial Port 选项进行设置，这里的数值为 115200；

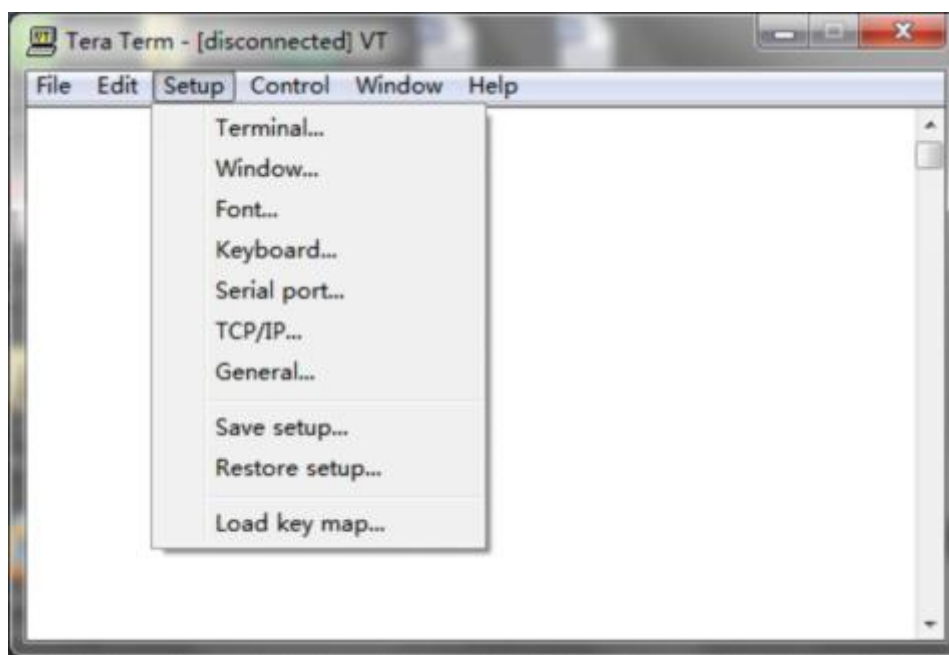


图 8-6 设置串口波特率

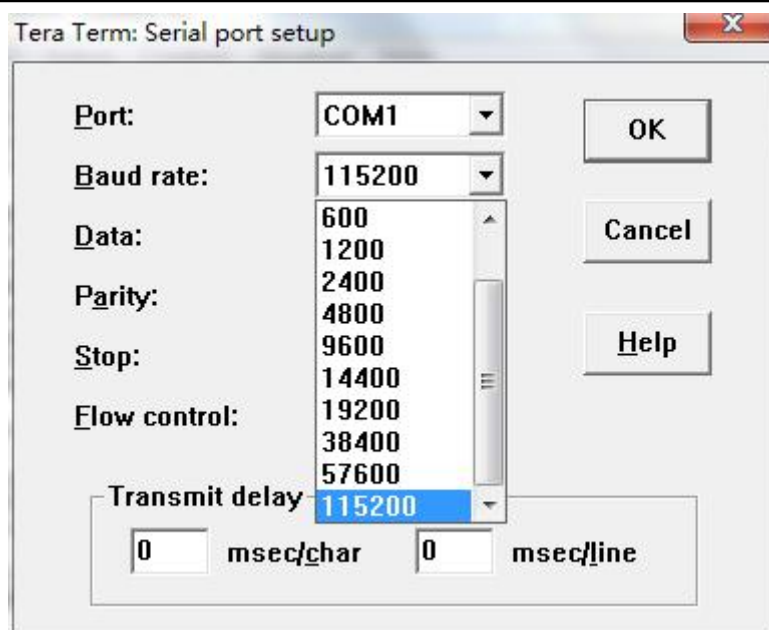


图 8-7 设置串口波特率