



# 龙芯LS2K500先锋板使用介绍

龙芯中科广州子公司 毛小川

2023年4月8日

# 目录

---

- 01 硬件介绍
- 02 使用入门
- 03 软件移植



## 1.0 硬件介绍--LoongArch龙架构

2021年，龙芯中科基于二十年的CPU研制和生态建设积累推出了龙架构（LoongArch），包括基础架构部分和向量指令、虚拟化、二进制翻译等扩展部分，近2000条指令。

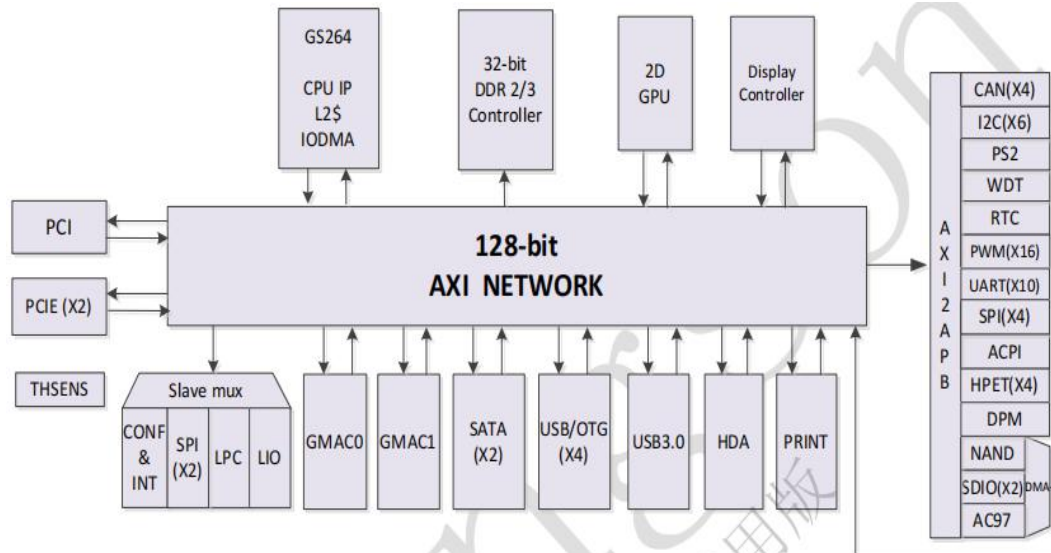
- 龙架构特点
  - 具有RISC指令架构的典型特征
  - 小尾端
  - 2个ISA子集 (LA32、LA64)
  - 4个权限级别 (PLV0 ~ PLV3)
  - 32个通用寄存器，32个浮点/向量寄存器
- CPU型号
  - 龙芯3号 (LA64)
    - 3A5000、3C5000、3D5000
  - 龙芯2号 (LA64)
    - 2K0500、2K1000LA、2K1500、2K2000
  - 龙芯1号 (LA32)
    - 1C102、1C103





## 1.1 硬件介绍--芯片介绍

2K500 采用LA264核集成丰富的外设接口，是LA64架构面向嵌入式领域的SOC，未来序列化发展，会推出2K300、2K600等芯片。其目的是为工业控制，工控互联网应用，BMC等提供多种解决方案



型号	LS2K500
主频	500-800M 硬浮点
内存	DDR3 533MHz
串口	2 个全功能 UART, 最多10个UART接口
SPI	6路 SPI
I2C	2路
CAN	4路 CAN 2.0
PWM	16路
USB	4路 USB2.0, 1路 USB3.0, 支持OTG 模式
网口	2路 10/100/1000Mbps 支持网络开机
PCIE	2路 PCIE 2.0 X1 支持RC/EP模式
显示	1路DVO和1路VGA, 最高分辨率支持1920x1080
HDA	1路7.1频道环绕立体声输出, 一路音频输入
GPIO	155位GPIO引脚, 128路支持外部中断输入
Nand	1路 最大支持4个片选, 支持系统启动
SDIO	2路 SD2.0 支持SDIO系统启动
SATA	2路 SATA2.0
功耗管理	支持ACPI v4.0a, 支持处理器核DVFS控制
封装方式	21mm * 21mm FCBGA
功耗	典型应用: 1W

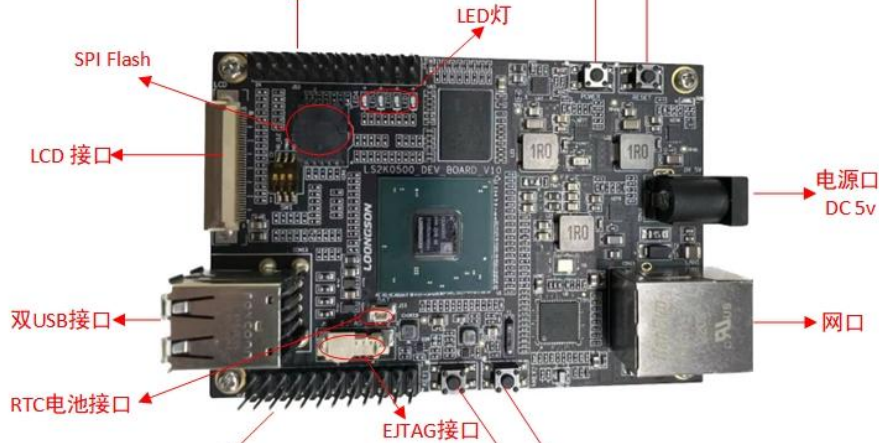


## 1.2 硬件介绍--板卡介绍

### 主要特点:

- 单板设计, 简洁小巧, 整板尺寸为: 9.3cm\*6.3cm
- 接口资源丰富, 常用接口通过插针形式引出
- 支持多种主流操作系统
- 支持多种主流GUI (Qt5/LVGL8)
- 支持快速启动
- 面向爱好者与开发者

1:UART7\_RX 2:UART7\_TX 3:I2C1\_SCL 4:I2C1\_SDA  
5:I2C2\_SCL 6:I2C2\_SDA 7:PWM1 8:PWM2  
9:PWM5 10:PWM0 11:GPIO\_106 12:GPIO\_110  
13:GPIO\_109 14:GPIO\_108 15:GPIO\_107 16:GPIO\_112  
17:GPIO\_113 18:GPIO\_111 19:SPI5\_MISO 20:SPI5\_CS  
21:SPI5\_MOSI 22:SPI5\_CLK 23:GND 24:VCC 3.3V













1:UART2\_RX 2:UART2\_TX 3:VCC 3.3V 4:GND  
5:SPI3\_CS 6:SPI3\_MOSI 7:SPI3\_MISO 8:SPI3\_CLK  
9:UART3\_TX 10:UART3\_RX 11:UART1\_TX 12:UART1\_RX  
13:UART9\_TX 14:UART9\_RX 15:UART0\_TX 16:UART0\_RX  
17:UART5\_TX 18:UART5\_RX 19:CAN0\_TX 20:CAN0\_RX  
21:CAN1\_TX 22:CAN1\_RX 23:VCC 5V 24:GND

规格参数		
基本信息	CPU	2K500
	CPU 主频	500~800MHz
	内存容量	512MB DDR3 SDRAM存储器
	存储容量	256/512MB Nandflash存储器
接口信息	千兆以太网	1路RJ45
	USB接口	2路USB
	LCD接口	1路LCD接口 (支持24位输出)
	按钮	2个 (电源、复位按钮)
	LED灯	4个用户自定义LED灯 (1个运行灯, 3个自定义)
	按键	2个按键
	调试串口	1个3线调试串口TTL (2.54间距连接器)
	EJTAG接口	1路 (预留单排插针)
扩展接口 (插针)	I2C接口	2路
	SPI接口	2路
	串口	6路
	CAN接口	2路
	PWM接口	4路
	USB接口	2路
	GPIO接口	8路
	工作参数	电源输入
产品尺寸		93mm X 63mm
系统软件	U-Boot 2022.04/Linux 5.10 /Buildroot 2021/Openharmony 3.2	



## 2.0 使用入门--资料包介绍

-  01-原理图
-  02-固件
-  03-内核
-  04-文件系统
-  05-交叉工具链
-  06-用户手册
-  07-芯片手册
-  08-常用工具软件
-  09-虚拟机环境
-  10-视频教程

1. 资料包下载链接:

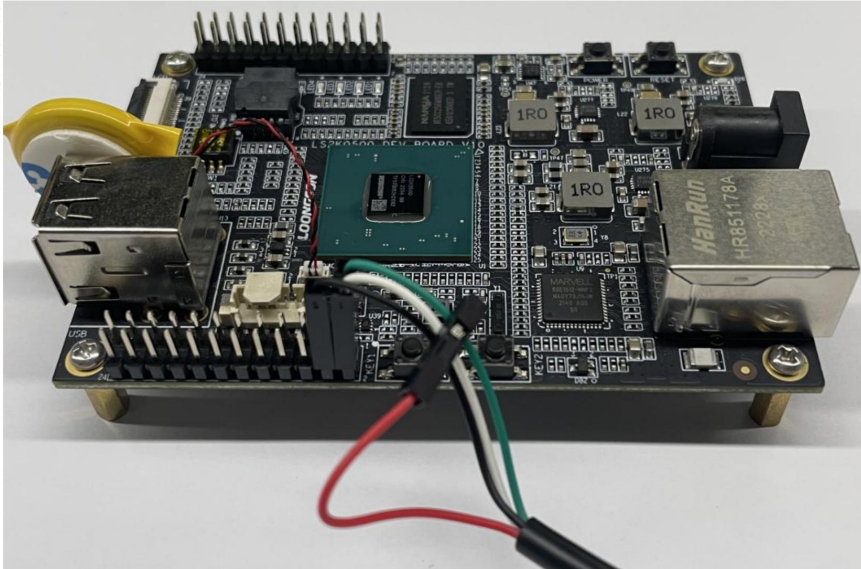
[https://pan.baidu.com/s/1\\_Ax3GkxIfIxm538oMND8-Q](https://pan.baidu.com/s/1_Ax3GkxIfIxm538oMND8-Q)

提取码: gv5f

2. 虚拟机和ubuntu的安装, ubuntu版本建议采用18.04



## 2.1 使用入门--开发板上手



- 硬件需要软件来驱动，没有软件的硬件是废柴，软件是硬件的灵魂。
- LS2K500先锋版采用的是uboot+linux+busybox的模式运行。即uboot作为板卡的固件，使用linux内核，文件系统为busybox(由buildroot构建而来)。
- 从上电后的启动顺序来说，先启动固件，固件引导内核启动，内核再引导文件系统启动。最终文件系统启动完毕后，出现终端，用户便可以进行操作。
- 如何与板卡交互：
  - Shell终端是和文件系统交互的一个手段。LS2K500先锋板有一个调试串口，调试串口从固件开始就生效，会把启动过程中的信息打印出来。等文件系统启动后，调试串口就会出现终端，就可以对系统进行操作。
  - 调试串口使用方式详见用户手册第四章第一节。(windows下的调试串口驱动见下图)
  - 使用ssh也可以打开终端来操作板卡，但是这个方式下是无法对固件进行操作的，因为ssh服务是要文件系统启动后才有的。





## 2.1 使用入门--开发板上手

```
vm@ubuntu:~/work/qemu-la/qemu-2k500/gz$ ls
disk nand.dat runqemu2k500 u-boot-with-spl.bin
vm@ubuntu:~/work/qemu-la/qemu-2k500/gz$ ./runqemu2k500
```

```
vm@ubuntu: ~/work/ubo... x  vm@ubuntu: ~/work/qem... x  vm@ubuntu: ~/work/qem... x  vm@ubuntu: ~/work/qem... x
[root@LS-GD ~]# uname -a
Linux LS-GD 5.10.0.lsgd-g81224fa0e223 #1 PREEMPT Mon Oct 31 09:05:30 CST 2022 loongarch64 GNU/Linux
[root@LS-GD ~]# lscpu
Architecture: loongarch64
Byte Order: Little Endian
CPU(s): 1
On-line CPU(s) list: 0
Thread(s) per core: 1
Core(s) per socket: 1
Socket(s): 1
BogoMIPS: 19.34
Flags: cpucfg lam ual fpu crypto
[root@LS-GD ~]# [ 57.064565] audit: type=1334 audit(1612279841.184:6): prog-id=8 op=UNLOAD
[ 57.065018] audit: type=1334 audit(1612279841.184:7): prog-id=7 op=UNLOAD
cat /proc/cpuinfo
system type : generic-loongson-machine
processor : 0
package : 0
core : 0
CPU Family : Loongson-64bit
Model Name : loongson-2k500
CPU Revision : 0x00
FPU Revision : 0x00
CPU MHz : 9.67
BogoMIPS : 19.34
TLB Entries : 2112
Address Sizes : 48 bits physical, 48 bits virtual
ISA : loongarch32 loongarch64
Features : cpucfg lam ual fpu crypto
Hardware Watchpoint : yes, iwatch count: 0, dwatch count: 0
[root@LS-GD ~]#
```

- 利用qemu来模拟2k500先锋板
  1. 解压qemu-2k500到任意目录，进到解压后的目录，将当前目录下的tmp/qemu链接到/tmp/qemu (`ln -sf `pwd`/tmp/qemu /tmp/qemu`)
  2. 安装vncview(`apt install tigervnc-viewer`)
  3. 运行runqemu2k500 脚本
  4. 运行vncview 来看图形效果 (vnc server为localhost)







## 2.2 使用入门--更新u-boot, kernel, rootfs

```
LOONGSON (GD)
bdinfo is in spi-flash
>>SPL: board_init_r()
Creating new bloblist size 400 at 900000000f030400
Trying to boot from BootSpace
spl_board_load_image(loongarch/mach-loongson)
spl_board_load_image(loongarch/mach-loongson) spl_load_fit
Finished bloblist size 400 at 900000000f030400
Jumping to U-Boot ...
```

```
*** U-Boot Boot Menu ***

[1] System boot select
[2] Update kernel
[3] Update rootfs
[4] Update u-boot
[5] Update ALL
[6] System install or recover
[7] Video resolution select
[8] U-Boot console

Press UP/DOWN to move, ENTER to select, ESC/CTRL+C to quit
```

1. 板卡启动之后串口就会有信息输出, 如果在启动的时候, 一直接住 ‘m’, 就会进入uboot的菜单, 如下图所示, 分别有对应的烧录选项。如果在启动的时候, 一直接住 ‘c’, 就会进入uboot的操作终端。

2. 更新支持U盘和网络更新两种方式, 要更新的内核 (文件名为: ulmage)、文件系统 (文件名为: rootfs-ubifs-ze.img)、固件 (文件名为: u-boot-with-spl.bin 或 u-boot.bin)。

3. 通过U盘更新, U盘需格式化为FAT32, 并在U盘目录下创建update文件夹, 并将要更新的文件放到此目录。

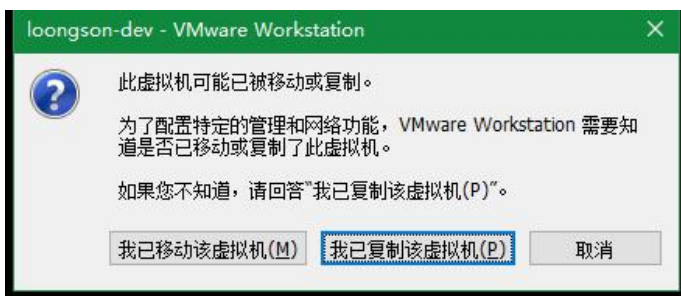
4. 通过网络更新, 将要更新的文件放到tftp服务的根目录

5. 详见用户手册第四章第三节



## 2.3 使用入门—编译环境安装

我的网盘 > 广东龙芯2K500-先锋板-v1.0 > 09-虚拟机环境 >



```
loongson@loongson-virtual-machine:~$ ls /opt/  
mips64el-linux-gcc-8.x  toolchain-loongarch64-linux-gnu-gcc8-host-x86_64-2022-07-18  
loongson@loongson-virtual-machine:~$ ls /opt/toolchain-loongarch64-linux-gnu-gcc8-host-x86_64-2022-07-18/  
bin  init-bash  lib  libexec  loongarch64-linux-gnu  share  sysroot  versions  
loongson@loongson-virtual-machine:~$
```

文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

```
loongson@loongson-virtual-machine:~$ ls /usr/local/  
bin  etc  games  include  lib  man  Qt-5.15.2-LA64  Qt-5.15.2-mips64el  sbin  share  src  
loongson@loongson-virtual-machine:~$
```

### ● 虚拟机安装

- ✓ 虚拟机安装镜像是x86的ubuntu18.04，可以自行前往网上下载。
- ✓ 同时提供了一个已经部署好的虚拟机文件，这是在Vmware15下部署的。有条件的话可以直接使用此虚拟机环境，无需自行安装虚拟机环境。第一次启动时，请选择“我已复制该虚拟机”。
- ✓ 如果想自行安装虚拟机环境，那么请参考用户手册第九章。

### ● 交叉编译环境搭建

- ✓ 如果采用提供的虚拟机，可见已经部署了交叉编译环境和Qt的交叉编译环境。只需要按照手册进行开发即可。
- ✓ 交叉编译的工具链，主要功能是在x86平台将固件，内核，文件系统，Qt程序等软件编译成目标平台（LA64）格式的二进制文件。



## 2.4 使用入门--编译u-boot

```
loongson@loongson-virtual-machine:~/Desktop/demo$ tar -zxf u-boot.tar.gz
loongson@loongson-virtual-machine:~/Desktop/demo$ cd u-boot/
loongson@loongson-virtual-machine:~/Desktop/demo/u-boot$ source set_env.sh
====>setup env for LoongArch...
loongson@loongson-virtual-machine:~/Desktop/demo/u-boot$ ls configs/ | grep loongson
loongson_2k1000_defconfig
loongson_2k1000_dp_defconfig
loongson_2k1000_jl_mb_mu_defconfig
loongson_2k500_evb_defconfig
loongson_2k500_hl_defconfig
loongson_2k500_mini_dp_defconfig
loongson_2k500_mini_dp_mini_defconfig
loongson@loongson-virtual-machine:~/Desktop/demo/u-boot$ make loongson_2k500_mini_dp_defconfig
HOSTCC scripts/basic/fixdep
HOSTCC scripts/kconfig/conf.o
YACC scripts/kconfig/zconf.tab.c
LEX scripts/kconfig/zconf.lex.c
HOSTCC scripts/kconfig/zconf.tab.o
HOSTLD scripts/kconfig/conf
#
# configuration written to .config
#
loongson@loongson-virtual-machine:~/Desktop/demo/u-boot$
```

这一条非必要



```
CC      spl/lib/abuf.o
CC      spl/lib/date.o
CC      spl/lib/rtc-lib.o
CC      spl/lib/elf.o
AR      spl/lib/built-in.o
LD      spl/u-boot-spl
OBJCOPY spl/u-boot-spl-nodtb.bin
SYM     spl/u-boot-spl.sym
CAT     spl/u-boot-spl-dtb.bin
COPY    spl/u-boot-spl.bin
CAT     u-boot-with-spl.bin
```

make 命令完成后

### ● 编译步骤

1. 先声明交叉编译工具链(源码中有脚本, 直接 source set\_env.sh即可)
2. 声明要编译的配置 (make loongson\_2k500\_mini\_dp\_defconfig)
3. 编译 (make -j4) (如果有4个核就-j4, 可以加快编译), 编译成功后会在当前目录生成u-boot-with-spl.bin 文件用于烧录更新。

### ● 详见用户手册第十章



## 2.5 使用入门--编译kernel I

```
Loongson@loongson-virtual-machine:~/Desktop/demo$ tar -zxf linux-5.10.tar.gz
Loongson@loongson-virtual-machine:~/Desktop/demo$ cd linux-5.10/
Loongson@loongson-virtual-machine:~/Desktop/demo/linux-5.10$ source set_env.sh
====>setup env for LoongArch...
Loongson@loongson-virtual-machine:~/Desktop/demo/linux-5.10$ ls ./arch/loongarch/configs/
Loongson_2k1000_defconfig  Loongson_2k500_defconfig  loongson_2k500_mini_defconfig  loongson3_defconfig
Loongson@loongson-virtual-machine:~/Desktop/demo/linux-5.10$ make loongson_2k500_defconfig
HOSTCC scripts/basic/fixdep
HOSTCC scripts/kconfig/conf.o
HOSTCC scripts/kconfig/confdata.o
HOSTCC scripts/kconfig/expr.o
LEX scripts/kconfig/lexer.lex.c
YACC scripts/kconfig/parser.tab.[ch]
HOSTCC scripts/kconfig/lexer.lex.o
HOSTCC scripts/kconfig/parser.tab.o
HOSTCC scripts/kconfig/preprocess.o
HOSTCC scripts/kconfig/symbol.o
HOSTCC scripts/kconfig/util.o
HOSTLD scripts/kconfig/conf
#
# configuration written to .config
#
Loongson@loongson-virtual-machine:~/Desktop/demo/linux-5.10$
```

这指令 非必要

```
SORTTAB vmlinux
SYSMAP System.map
OBJCOPY arch/loongarch/boot/vmlinux.bin
GZIP arch/loongarch/boot/vmlinux.bin.gz
UIIMAGE arch/loongarch/boot/uImage.gz
Image Name: Linux-5.10.0.lsgd-gd3eccd5316bd
Created: Tue Apr 4 18:02:14 2023
Image Type: LoongArch Linux Kernel Image (gzip compressed)
Data Size: 7748449 Bytes = 7566.84 KiB = 7.39 MiB
Load Address: 00200000
Entry Point: 00c894d0
Image arch/loongarch/boot/uImage is ready
Loongson@loongson-virtual-machine:~/Desktop/demo/linux-5.10$ ls ./arch/loongarch/boot/
compressed dts Makefile tools uImage uImage.gz vmlinux.bin vmlinux.bin.gz
Loongson@loongson-virtual-machine:~/Desktop/demo/linux-5.10$
```

### ● 编译步骤

1. 先声明交叉编译工具链(有脚本, 直接 source set\_env.sh即可),
2. 声明要编译的配置(make loongson\_2k500\_defconfig),
3. 编译(make ulmage -j4) (如果有4个核就-j4, 可以加快编译), 编译成功后源码下的./arch/loongarch/boot/uImage.gz文件就是内核镜像, 将其复制为uImage用于烧录更新

### ● 详见用户手册第十一章



## 2.6 使用入门--编译rootfs

```
vm@ubuntu:~/work/fs/buildroot-2021.02.4$ make list-configs
1. 全量系统的编译配置
板卡配置名          对应适用板卡
loongson2k1000_jinlong_defconfig  1s2k1000mips版本的金龙板卡或者其他1s2k1000mips版本的板卡通用配置
loongson2k1000la_jinlong_defconfig 1s2k1000LA版本的金龙板卡或者其他1s2k1000LA版本的板卡通用配置
loongson2k500_defconfig           1s2k500板卡通用配置（串口2作为调试串口）
loongson2k500_mini_dp_defconfig   1s2k500先锋板卡通用配置（串口2作为调试串口）

2. 小型系统的编译配置
板卡配置名          对应适用板卡
loongson2k1000_jinlong_mini_defconfig 1s2k1000mips版本的金龙板卡或者其他1s2k1000mips版本的板卡小型系统通用配置
loongson2k1000la_jinlong_mini_defconfig 1s2k1000LA版本的金龙板卡或者其他1s2k1000LA版本的板卡小型系统通用配置
loongson2k500_mini_defconfig         1s2k500板卡小型系统通用配置（串口2作为调试串口）
loongson2k500_mini_dp_mini_defconfig 1s2k500先锋板卡小型系统通用配置（串口2作为调试串口）

3. ramdisk的编译配置
板卡配置名          对应适用板卡
loongson2k1000la_jinlong_ramdisk_defconfig 1s2k1000LA版本的金龙板卡或者其他1s2k1000LA版本的板卡ramdisk编译通用配置
loongson2k500_ramdisk_defconfig         1s2k500板卡ramdisk编译通用配置（串口2作为调试串口）

4. 快速启动系统的编译配置
板卡配置名          对应适用板卡
loongson2k1000la_jinlong_fastboot_defconfig 1s2k1000LA版本的金龙板卡或者其他1s2k1000LA版本的板卡快速启动系统通用配置
loongson2k500_fastboot_defconfig         1s2k500板卡快速启动系统通用配置（串口2作为调试串口）
loongson2k500_mini_dp_fastboot_defconfig 1s2k500先锋板卡快速启动系统通用配置（串口2作为调试串口）

5. 最小化纯busybox的编译配置((整个系统大小:8.9M))
板卡配置名          对应适用板卡
loongson2k1000la_jinlong_pure_busybox_defconfig 1s2k1000LA版本的金龙板卡或者其他1s2k1000LA版本的板卡编译通用配置
loongson2k500_pure_busybox_defconfig         1s2k500板卡编译通用配置（串口2作为调试串口）
vm@ubuntu:~/work/fs/buildroot-2021.02.4$
```

```
vm@ubuntu:~/work/fs/buildroot-2021.02.4/output$ cd images/
vm@ubuntu:~/work/fs/buildroot-2021.02.4/output/images$ ls
rootfs.tar rootfs.tar.gz rootfs.ubi rootfs.ubifs
vm@ubuntu:~/work/fs/buildroot-2021.02.4/output/images$
```

- 2k500先锋板的rootfs 是由buildroot 编译而来，buildroot包含了非常丰富的第三软件包，方便编译集成

- 编译步骤

1. 声明要编译的配置 (make loongson\_2k500\_mini\_dp\_defconfig) (make list-configs 可查看各配置的说明)。
2. 编译 (make -j4) (如果有4个核就-j4, 可以加快编译)，编译成功后源码下的./out/images/rootfs.ubi文件就是文件系统镜像，将其复制为rootfs-ubifs-ze.img用于烧录更新

- 详见用户手册第十二章



# 3.0 软件移植--应用软件移植

## 开源软件源码编译移植 (C/C++代码)

- 1、获取源码：通过github或第三方开源社区获取源码；
- 2、准备编译环境：安装编译器gcc等；
- 3、使用开源软件源码中的CMake或AutoConfig脚本生成 Makefile；

由于旧版本的config.guess 不支持loongarch, 在执行./configure时会出如下的错误  
configure: error: cannot guess build type; you must specify one

解决方法:

方法一. 找到 config.guess 文件编辑它, 在ia64:Linux:\*:\* ) 下面添加Loongarch 架构的支持。

```
loongarch*:Linux:*:*)
    echo ${UNAME_MACHINE}-unknown-linux-gnu
    exit ;;
```

方法二. 从网上获取最新的config.guess 和 config.sub 文件然后进行替换

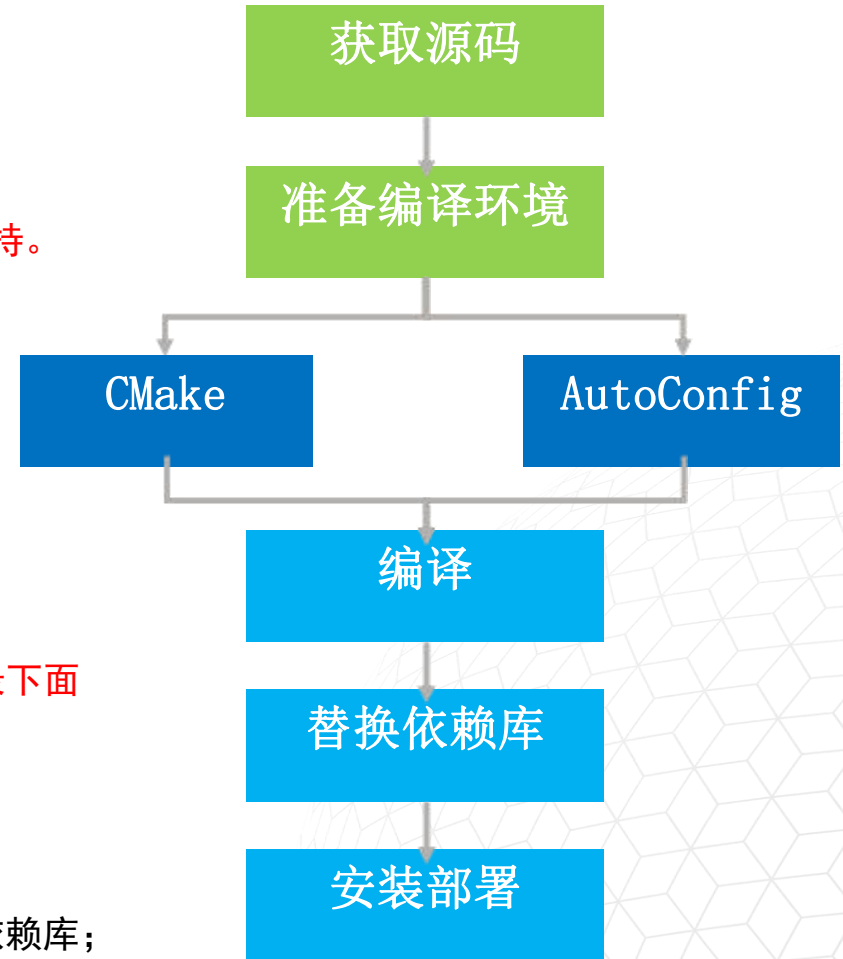
```
$ sudo wget -O /usr/share/misc/config.sub
"git.savannah.gnu.org/gitweb/?p=config.git;a=blob_plain;f=config.sub;hb=HEAD"
$ sudo wget -O /usr/share/misc/config.guess
"git.savannah.gnu.org/gitweb/?p=config.git;a=blob_plain;f=config.guess;hb=HEAD"
```

在源码目录执行下面的命令会将/usr/share/misc/config.sub 和 config.guess 替换掉工程目录下面的config.sub 和 config.guess 文件

```
$ libtoolize -f -i -c
```

通过上面的方法修改之后再运行./configure 就可以了

- 4、执行Makefile编译可执行程序；
- 5、替换依赖库：通过开源软件readme文件、编译时的动态库缺失或链接错误，重新编译或替换依赖库；
- 6、将可执行程序安装部署到生产或测试系统。
- 7、详见用户手册第5.3.1





## 3.1 软件移植--移植建议

### C/C++代码编译选项、编译宏移植

功能

X86编译选项

LoongArch64编译选项

64位编译

定义编译生成的应用程序为64位，编译选项不同

`-m64`

`-mabi=lp64d`

指令集

Makefile中定义指令集类型, 由X86修改为LoongArch

`-march=skylake`

`-march=loongarch64`

编译宏

原有x86版编译宏替换为LoongArch宏

`__X86_64__`

`__loongarch__`

查看编译器自定义译宏：`echo | gcc -E -dM -`



## 3.2 软件移植--Qt软件移植

```
vm@ubuntu:~/work/fs/buildroot-2021.02.4/qt_test$ cat env_loongarch64.sh
workdir=`pwd`

DEST=$workdir/./output/host

PATH=$DEST/bin:$PATH

#QMAKESPEC=$DEST/mkspecs/linux-g++/
QMAKESPEC=$DEST/mkspecs/linux-loongarch64-g++/

QTDIR=$DEST/loongarch64-buildroot-linux-gnu/sysroot/usr

LD_LIBRARY_PATH=$DEST/loongarch64-buildroot-linux-gnu/sysroot/usr/lib

export PATH QMAKESPEC QTDIR LD_LIBRARY_PATH

vm@ubuntu:~/work/fs/buildroot-2021.02.4/qt_test$
```

```
vm@ubuntu:~/work/fs/buildroot-2021.02.4/qt_test$ ls
env_loongarch64.sh env_mips64el.sh main.cpp mainwindow.cpp mainwindow.h mainwindow.ui test.pro
vm@ubuntu:~/work/fs/buildroot-2021.02.4/qt_test$ source env_loongarch64.sh
vm@ubuntu:~/work/fs/buildroot-2021.02.4/qt_test$ qmake -v
QMake version 3.1
Using Qt version 5.15.2 in /home/vm/work/fs/buildroot-2021.02.4/output/host/loongarch64-buildroot-linux-gnu/sysro
ot/usr/lib
vm@ubuntu:~/work/fs/buildroot-2021.02.4/qt_test$ qmake test.pro
Info: creating stash file /home/vm/work/fs/buildroot-2021.02.4/qt_test/.qmake.stash
vm@ubuntu:~/work/fs/buildroot-2021.02.4/qt_test$ make -j4
```

- 移植思路:

1. 交叉编译Qt (1. 提供的虚拟机环境已包括交叉编译好的Qt 2. 用 buildroot编译Qt)
2. 发布交叉编译的Qt环境变量 (source buildroot/qt\_test/env\_loongarch64.sh )
3. qmake xx.pro
4. make -j4

或在QtCreator 配置好交叉编译的QT  
利用QtCreator 进行编译

- 详见用户手册第5.3.1





## 3.3 软件移植--相关资源

### 龙芯技术社区 ([https://bbs.elecfans.com/group\\_1650](https://bbs.elecfans.com/group_1650))



龙芯技术社区

龙芯技术交流, 可以加V 13510979604 入群

经验: 1 | 组长: jf\_39160458, wuxtao, xusiwei1236

加入小组

首页

成员列表

发帖

全部

资料

问答

2K系列

3号系列

帖子	作者/时间	回复/查看
【广东龙芯2K500先锋板试用体验】+试用5 CAN总线主机	jyaxz 2023-2-6	0 272
【广东龙芯2K500先锋板试用体验】CPU性能基准测试——CoreMark和计算质数	xusiwei1236 2023-2-3	0 3046
【广东龙芯2K500先锋板试用体验】龙芯2K500丝滑体验LVGL demo	HonestQiao 2023-1-31	0 545
【广东龙芯2K500先锋板试用体验】运行边缘AI框架——TFLM	xusiwei1236 2023-1-31	0 1305
【广东龙芯2K500先锋板试用体验】LoongArch汇编初体验	HonestQiao 2023-1-31	0 1143
【广东龙芯2K500先锋板试用体验】龙芯2K500 Linux环境基础调教和使用	HonestQiao 2023-1-31	0 642

【广东龙芯2K500先锋板试用体验】4.移植开发MQTT服务与MQTT通讯测试	ALSET 2023-1-29	0 282
【广东龙芯2K500先锋板试用体验】+试用4 Modbus-RTU主机	jyaxz 2023-1-28	1 181
【广东龙芯2K500先锋板试用体验】驱动OLED小屏播放视频	xusiwei1236 2023-1-17	0 10084
【广东龙芯2K500先锋板试用体验】3.开发http服务与通过websocket实时更新页面数据	ALSET 2023-1-16	0 1533
【广东龙芯2K500先锋板试用体验】2. C/C++开发环境搭建测试	ALSET 2023-1-16	0 887
【广东龙芯2K500先锋板试用体验】1.从开箱到上电到连接主机过程	ALSET 2023-1-16	0 1198
【广东龙芯2K500先锋板试用体验】+试用3 实时内存数据库	jyaxz 2023-1-13	1 2044
【广东龙芯2K500先锋板试用体验】试用2-处理器性能测试	jyaxz 2023-1-6	2 844
【广东龙芯2K500先锋板试用体验】编译内核	华仔stm32 2023-1-3	0 167
【广东龙芯2K500先锋板试用体验】试用1-在终端完成网络配置	jyaxz 2023-1-3	0 583
【广东龙芯2K500先锋板试用体验】CAN测试	华仔stm32 2022-12-30	0 583
【广东龙芯2K500先锋板试用体验】图形化界面配置网络	华仔stm32 2022-12-28	0 385



## 3.3 软件移植--相关资源

龙芯开源社区 (<http://www.loongnix.cn>), 龙芯基础软件最新成果的发布地。

### 操作系统

[Loongnix操作系统](#)

### API与基础软件

[龙芯浏览器](#) | [Java](#) | [.NET](#) | [音视频技术](#) | [Electron](#) | [Node.js](#) | [CEF](#) | [NW.js](#) | [龙芯Python仓库](#) | [LCF应用二进制兼容框架](#)

### 图形图像

[龙芯VBIOS生成工具](#)

### 工具链

[GNU 编译工具链](#) | [龙芯LLVM编译器](#) | [龙芯Golang编译器](#) | [龙芯Rust编译器](#)

### 调测工具

[ebpf](#) | [ftrace](#) | [gdb](#) | [kdump](#) | [kprobe](#) | [strace](#)

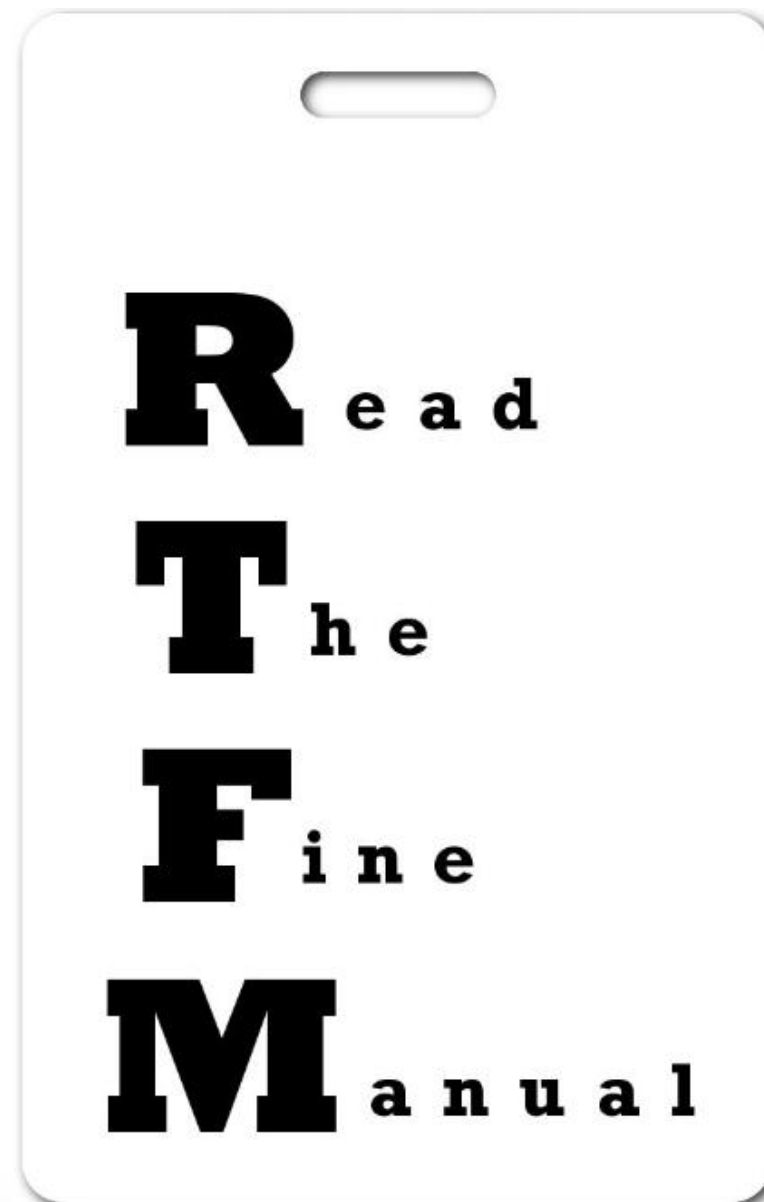
### 云计算

[KVM](#) | [容器镜像仓库](#) | [Kubernetes](#) | [Docker](#) | [Alpine](#)



## LoongArch软件开发--开发文档

1. 计算机体系结构基础 (LoongArch)-3rd  
<https://foxsen.github.io/archbase/bookdown.pdf>
2. 龙芯架构参考手册 - 卷一：基础架构  
<https://github.com/loongson/LoongArch-Documentation/releases/latest/download/LoongArch-Vol1-v1.02-CN.pdf>
3. 龙芯架构 ELF psABI  
<https://loongson.github.io/LoongArch-Documentation/LoongArch-ELF-ABI-CN.pdf>
4. 龙芯架构工具链约定  
<https://loongson.github.io/LoongArch-Documentation/LoongArch-toolchain-conventions-CN.pdf>
5. 龙芯架构 SMBIOS 规范  
<https://loongson.github.io/LoongArch-Documentation/LoongArch-Processor-SMBIOS-Spec-EN.html>
6. 龙芯架构文档  
<https://loongson.github.io/LoongArch-Documentation/README-CN.html>



自主决定命运  
创新成就未来



龙芯中科  
LOONGSON TECHNOLOGY