

# 龙芯机器人控制器操作手册

龙芯 (LSIC300A) 机器人控制器通过两片 PWM 波扩展芯片 (PCA9685PW) 扩展出 32 路 PWM, 其中一路最多可扩展 16 路 PWM 信号。

## 硬件连接:

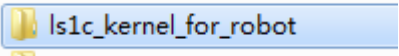
1、给龙芯机器人控制器供电, 它有两个 USB 口供电方式, 我用的是 mini usb 口供电的。(主板供电后有三个灯亮)



2、连接串口三根线 (注意地线不要插错了)

3、插上调试舵机 (DS3115mg), 注意不要插反, 棕色线朝外。我用了三路舵机调试

4、给舵机供电 (7.2V) 需要外接电源。(同样注意不要 V+和地线接反)

## 软件调试:

1、龙芯机器人控制器主板内核及 I2C 驱动下载通过 TF 卡方式下载, 打开存放编译好的内核的文件夹  将文件中的内核

 ulimage  
 vmlinuz

( ) 拷贝到 TF 卡里 (注意拷贝完过后不要立马插到机器人控制板里)

2、打开计算机串口终端, 并配置好串口 (8/N/1/115200), 然后按一下主板复位键 (如果串口不能正常打印), 此时系统会进入 U-boot 界面

```

checkboard
Board: openlongson board (CPU Speed 252 MHz/ Mem @ 126 MHz/ Bus @ 126 MHz)
DRAM: 32 MiB
NAND: 128 MiB
MMC:
SF: Detected w25x40 with page size 256 Bytes, erase size 4 KiB, total 512 KiB
*** warning - bad CRC, using default environment

In:    serial
Out:   serial
Err:   serial
SPI:   ready
Net:   Phy not found
PHY reset timed out
error: synopGMAC_reset DmaBusMode: 0x00020101
dwmac.bfe10000
Hit any key to stop autoboot:  0

Loading from nand0, offset 0x100000
** Unknown image type
Wrong Image Format for bootm command
ERROR: can't get kernel image!
uboot#

```

3、将刚才拷贝过得内核系统的 TF 卡插入龙芯机器人主板，通过 TF 卡烧写内核和文件系统通过以下的指令实现：

以下为加载内核系统：

```
nand erase.part kernel
```

```
mmc_spi 0:2
```

```
mmcinfo
```

```
fatload mmc 0:1 a0100000 uImage          注意内核加载后串口打印出来的长度
```

```
nand write 80100000 kernel xxx
```

xxx 表示使用加载后打印出来的长度，我这里设置 xxx 为 1ff951 (注意这里一定要设置 XXX 的长度，否则内核加载会出现错误)。

加载后的界面如下图所示：

```
Loading from nand0, offset 0x100000
```

```
** Unknown image type
```

```
Wrong Image Format for bootm command
```

```
ERROR: can't get kernel image!
```

```
uboot# nand erase.part kernel
```

```
NAND erase.part: device 0 offset 0x100000, size 0xd00000
```

```
Erasing at 0xde0000 -- 100% complete.
```

```
OK
```

```
uboot# mmc_spi 0:2
```

```
MMC_SPI: 0 at 0:2 hz 25000000 mode 0
```

```
uboot# mmcinfo
```

```
Device: MMC_SPI
```

```
Manufacturer ID: 0
```

```
OEM: 0
```

```
Name: Tran Speed: 0
```

```
Rd Block Len: 0
SD version 2.0
High Capacity: No
Capacity: 0 Bytes
Bus Width: 1-bit
uboot# fatload mmc 0:1 a0100000 uImage
** Bad device mmc 0 **
uboot# nand write 80100000 kernel 1ff951

NAND write: device 0 offset 0x100000, size 0x1ff951
2095441 bytes written: OK
```

下面开始加载文件系统:

```
nand erase.part root
mmc_spi 0:2
mmcinfo
fatload mmc 0:1 a0100000 rootfs-yaffs2.img
注意加载后打印出来的长度
nand write.yaffs 80100000 root xxx
```

xxx 表示使用加载后打印出来的长度。(我这里设置 xxx 为 717000。注意这里一定要设置 XXX 的长度, 否则内核加载会出现错误)。

文件系统加载后的界面如下图:

```
uboot# nand erase.part root

NAND erase.part: device 0 offset 0xe00000, size 0x3200000
Erasing at 0x3fe0000 -- 100% complete.
OK
uboot# mmc_spi 0:2
MMC_SPI: 1 at 0:2 hz 25000000 mode 0
uboot# mmcinfo
Device: MMC_SPI
Manufacturer ID: 0
OEM: 0
Name: Tran Speed: 0
Rd Block Len: 0
SD version 2.0
High Capacity: No
Capacity: 0 Bytes
Bus Width: 1-bit
```

(加载行实在太多就没截屏)

```
uboot# fatload mmc 0:1 a0100000 rootfs-yaffs2.img
** Bad device mmc 0 **
uboot# nand write.yaffs 80100000 root 717000
```

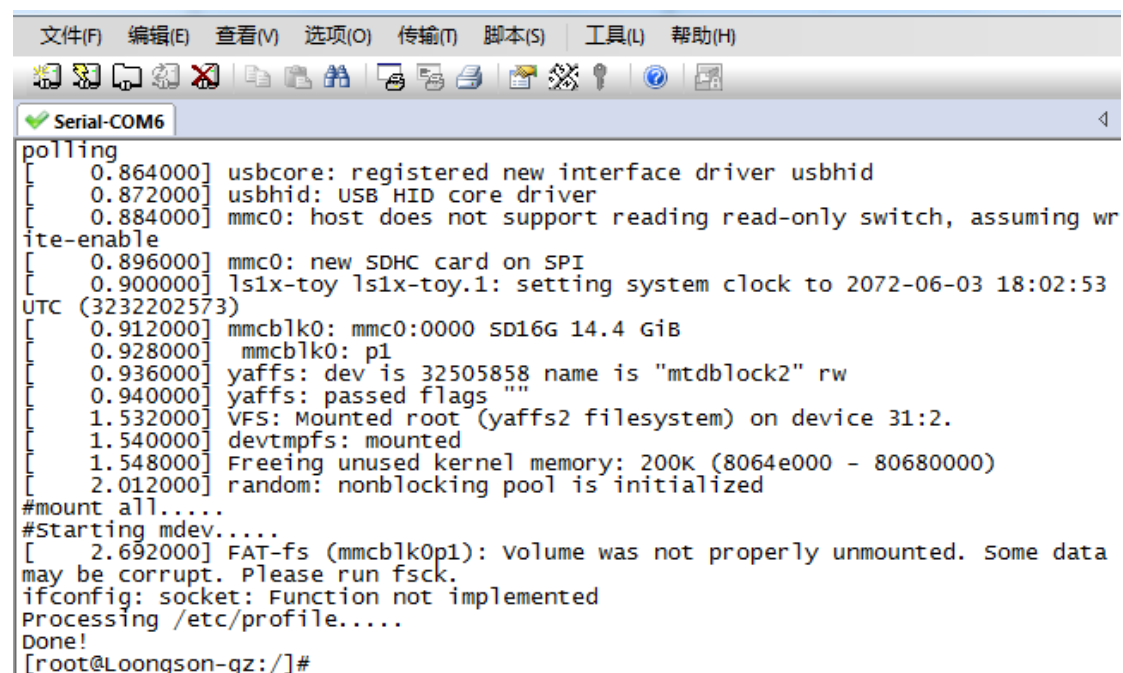
```
NAND write: device 0 offset 0xe00000, size 0x717000
7434240 bytes written: OK
uboot# <INTERRUPT>
uboot#
```

下面开始加载内核启动参数:

```
setenv bootargs console=ttyS2,115200 root=/dev/mtdblock2 noinitrd
init=/linuxrc rootfstype=yaffs2 rw
setenv bootcmd nboot kernel\;bootm 81000000
```

后拔掉龙芯机器人控制器上面的 TF 卡后，在插上 TF 卡，然后使系统复位。

复位后内核系统和文件系统已经烧入，界面如下:



```
文件(F) 编辑(E) 查看(V) 选项(O) 传输(T) 脚本(S) 工具(L) 帮助(H)
Serial-COM6
polling
[ 0.864000] usbcore: registered new interface driver usbhid
[ 0.872000] usbhid: USB HID core driver
[ 0.884000] mmc0: host does not support reading read-only switch, assuming wr
ite-enable
[ 0.896000] mmc0: new SDHC card on SPI
[ 0.900000] ls1x-toy ls1x-toy.1: setting system clock to 2072-06-03 18:02:53
UTC (3232202573)
[ 0.912000] mmcblk0: mmc0:0000 SD16G 14.4 GiB
[ 0.928000] mmcblk0: p1
[ 0.936000] yaffs: dev is 32505858 name is "mtdblock2" rw
[ 0.940000] yaffs: passed flags ""
[ 1.532000] VFS: Mounted root (yaffs2 filesystem) on device 31:2.
[ 1.540000] devtmpfs: mounted
[ 1.548000] Freeing unused kernel memory: 200K (8064e000 - 80680000)
[ 2.012000] random: nonblocking pool is initialized
#mount all.....
#Starting mdev.....
[ 2.692000] FAT-fs (mmcblk0p1): volume was not properly unmounted. Some data
may be corrupt. Please run fsck.
ifconfig: socket: Function not implemented
Processing /etc/profile.....
Done!
[root@Loongson-gz:/]#
```

然后通过以上指令可以看到系统已经找到 PWM 对应的芯片 pwmchip0 和 pwmchip17 页面如下:

```

[root@Loongson-gz:/]#ls
bin          home         lost+found  root        tmp
dev          lib          mnt        /sbin       usr
etc          linuxrc     proc        sys         var
[root@Loongson-gz:/]#cd sys
[root@Loongson-gz:/sys]#ls
block      class      devices    fs          module
bus        dev        firmware   kernel
[root@Loongson-gz:/sys]#cd class
[root@Loongson-gz:/sys/class]#ls
bdi          i2c-adapter mem          pwm          scsi_host    ubi
block       i2c-dev    misc         rtc          spi_master   udc
bsg          input      mmc_host     scsi_device  spidev       vc
gpio         leds       mtd          scsi_disk    tty          vtconsole
[root@Loongson-gz:/sys/class]#cd pwm
[root@Loongson-gz:/sys/class/pwm]#ls
pwmchip0    pwmchip17

```

查单 pwmchip0 文件夹下面对应的文件接口，同理你也可以查看 pwmchip17 文件夹下面的文件接口页面如下：

```

[root@Loongson-gz:/sys/class/pwm]#ls
pwmchip0    pwmchip17
[root@Loongson-gz:/sys/class/pwm]#cd pwmchip0
[root@Loongson-gz:/sys/devices/platform/ls1x-i2c.0/i2c-0/0-0040/pwm/pwmchip0]#ls
device      export      npwm        subsystem    uevent      unexport
[root@Loongson-gz:/sys/devices/platform/ls1x-i2c.0/i2c-0/0-0040/pwm/pwmchip0]#

```

打开 pwmchip0 中的 pwm0 输出通道页面如下：

```

[root@Loongson-gz:/sys/class/pwm]#cd pwmchip0
[root@Loongson-gz:/sys/devices/platform/ls1x-i2c.0/i2c-0/0-0040/pwm/pwmchip0]#ls
device      export      npwm        subsystem    uevent      unexport
[root@Loongson-gz:/sys/devices/platform/ls1x-i2c.0/i2c-0/0-0040/pwm/pwmchip0]#echo 0 > export
[root@Loongson-gz:/sys/devices/platform/ls1x-i2c.0/i2c-0/0-0040/pwm/pwmchip0]#ls
device      export      npwm        pwm0         subsystem    uevent      unexport
[root@Loongson-gz:/sys/devices/platform/ls1x-i2c.0/i2c-0/0-0040/pwm/pwmchip0]#cd
pwm0
[root@Loongson-gz:/sys/devices/platform/ls1x-i2c.0/i2c-0/0-0040/pwm/pwmchip0/pwm0]#ls
duty_cycle  enable     period     polarity    uevent
[root@Loongson-gz:/sys/devices/platform/ls1x-i2c.0/i2c-0/0-0040/pwm/pwmchip0/pwm0]#echo 1 > enable
[root@Loongson-gz:/sys/devices/platform/ls1x-i2c.0/i2c-0/0-0040/pwm/pwmchip0/pwm0]#

```

因为龙芯控制器扩展 32 路 PWM 波信号这里打开了标号 0（0-15 标号）

设置 pwm0 输出信号的周期为 20ms：

```

#echo 20000000 > period
#

```

（因为在串口终端全贴字太小会看不清，望谅解）

**注意：时间单位为 ns；**

然后设置 PWM 周期中高电平所占的时间为 1.5ms，页面如下：

```

#echo 1500000 > duty_cycle
#

```

在保证龙芯控制器舵机供电正常的情况下，你会看到舵机会转动。

你可以设置高电平所占的时间比如为 1.9ms 舵机转的幅度会大点，反之一样。

## 说明：

**注意：**不同的舵机设置最大占空比和最小占空比会有所不同，这具体要看你使用的舵机，具体请参考你使用的舵机的参考手册。

注意：以上介绍只说明怎么通过龙芯机器人控制器控制一路舵机，我在调试的时候使用了三路舵机，其他两路舵机使用情况和以上一样，另外两路我就不一一叙述。

以上龙芯机器人使用说明书写的有不妥之处，希望广大开发者敬请谅解。

