



DVK512 树莓派扩展版 用户手册

DVK512 是一款 Raspberry Pi A+/B+/2 代 B 的外扩板,带有丰富的资源及扩展接口,支持各类外围 模块的接入。



特点**:**

- 板载 USB TO UART, 方便进行串口调试。
- 板载 4 个按键和 4 个 LED。
- 板载 RTC。
- 板载 LCD1602 接口。
- 板载 UART、SPI、I2C、8I/Os 接口,兼容 Waveshare 标准,方便接入各种外围。



资源简介



[接口简介]

- 1. RPi 连接口
- 2. UART 接口
- 3. 8I/Os 接口
- 4. SPI 接口
- 5. I2C 接口
- 6. 字符 LCD 接口
- 7. USB TO UART 接口

[器件简介]

- 8. 电源 LED
- 9. 用户 LED
- 10. 用户按键
- 11. 电位器
- 12. RTC 实时时钟供电电池座
- 13. PCF8563
- 14. 32.768K 晶振
- 15. **CP2102**

[跳线说明]

- 16. CP2102 跳线
- 17. RTC 跳线
- 18. 用户 **LED 跳线**
- 19. 用户按键跳线

以上跳线使用说明:

- 短接跳线:接入到示例
 程序指定的 I/O;
- 断开跳线:可改为使用
 连接线接入自定义的
 I/O。



目:	录		
资测	原简介		2
1.	基本操作	F	4
	1.1.	系统镜像的烧写	4
	1. 2.	部署串口调试环境	4
2.	DVK512 🤉	及扩展功能测试	6
	2.1.	安装必要的 AP I	6
	2. 2.	LED 实验	6
	2. 3.	按键实验	7
	2. 4.	8 路电平转换模块实验	8
	2. 5.	摇杆实验	10
	2. 6.	蜂鸣器与 PWM 实验	10
	2. 7.	DS18B20 实验	10
	2. 8.	红外遥控实验	12
	2.9.	LCD1602 实验	13
	2. 10.	PCF8563 RTC 实验	13
	2.11.	PCF8591 AD 实验	14
	2. 12.	PCF8591 DA 实验	16
	2. 13.	重力加速度与磁力计 LSM303DLHC 模块	17
	2. 14.	角速度传感器 L3G4200D 模块	18
	2. 15.	串口实验	19
	2. 16.	GPS 实验	19
版机	又声明		21
版2	本更新记录	t <	21



1. 基本操作

1.1. 系统镜像的烧写

- 1) 到树莓派官网下载最新的镜像。
- 2) 使用 SDFormatter.exe 软件,格式化 TF 卡。

注意: TF 卡的容量不得低于 4GB。此操作必须搭配 TF 卡读卡器,用户需要另外购买。

3) 打开 Win32DiskImager.exe 软件,选择上一步准备的系统镜像,点击 write 烧写系统镜像。

🦦 Win32 Disk Imager								
Image File	- Device -							
':/install/raspberrypi/2013-05-25-wheezy-raspbian.img 📄 [I:\] 🔻								
Copy MD5 Hash: Progress								
Version: 0.8 Cancel Read Write	Exit							
Waiting for a task.								

图 1. 使用 Win32DiskImager.exe 软件烧写系统镜像

1.2. 部署串口调试环境

- 通过 mini USB 线把 DVK512 的 UART TO USB 接口连接到电脑。
 注意: DVK512 提供的 USB 转串口使用的是 mini USB 接口,和树莓派的供电接口(micro USB))不同,请注意区分。
- 2) 打开 cp2102_driver 安装驱动。
- 3) 打开串口查看软件 PuTTY.exe,进行设置。
 - Serial line: 选择对应的 COM 口,用户可通过"设备管理器"查看。
 - Speed: 115200
 - Connection type: Serial



🕵 PuTTY Configuration	×					
Category:						
Session	Basic options for your PuTTY session					
Logging	Specify the destination you want to connect to					
···· Keyboard ···· Bell	COM3 115200					
Features ⊡. Window	Connection type:					
Appearance Behaviour Translation Selection Colours Connection Data Proxy Telnet Due	Load, save or delete a stored session Saved Sessions COM3 Default Settings Load Save Delete					
Serial	Close window on exit: Always Never Only on clean exit					
About	<u>Open</u> <u>Cancel</u>					

图 2. 设置 PuTTY 软件

- 4) 设置完毕之后,点击 Open。
- 5) 系统启动后,输入用户名: pi,密码: raspberry。即可从串口终端进入树莓派。 注意:下文如果没有特别说明,均是使用 PuTTY 软件,通过串口对树莓派进行操作。此时 串口被系统调试功能所占用,不能作为普通的串口使用。具体请参见树莓派库函数安装教 程。



2. DVK512 及扩展功能测试

本节的操作如果没有特殊说明,都需要将 DVK512 通过 RPi 连接口连接到树莓派 B+,才可进行操作。

2.1. 安装必要的 API

本手册用到的程序,大部分都需要安装对应的库并设置启动对应接口内核驱动才能正常使用,您可以在 <u>http://www.waveshare.net/wiki/Pioneer600 Datasheets</u> 下载示例程序用到的 API 源码。 把它们复制到树莓派系统中并解压。如果您直接使用我们配置好的镜像,那么可以在这个目录: /home/pi/DVK512 找到所需的 API。详情请查看树莓派系列教程。

注:下面实验都是在对应程序路径下操作,例如 LED 的 bcm2835 程序。

pi@raspberrypi ~/DVK512/LED/bcm2835 \$ sudo ./led

这表示在树莓派 ~/DVK512/LED/bcm2835 路径下执行 led 这个编译生成的可执行文件。

sudo 表示已 root 用户权限执行命令,如果以 root 用户登录,则不需要 sudo。例如: root@raspberrypi:/home/pi/DVK512/LED/bcm2835# ./led

如无特殊说明,下面实验都是直接运行已编译生成的可执行文件,如需编译 bcm2835, wiringPi 和 sysfs 等 C 语言程序,则进入程序目录下面执行 make 命令编译程序, make clean 命令删除执行文件。例如:

pi@raspberrypi ~/DVK512/LED/bcm2835 \$ make clean

rm led

pi@raspberrypi ~/DVK512/LED/bcm2835 \$ make

gcc -Wall led.c -o led -lbcm2835

Python 程序不用编译,直接运行即可,例如: pi@raspberrypi ~/DVK512/LED/python \$ sudo python led.py

2.2. LED 实验

bcm2835 程序:

1) 在终端执行:

pi@raspberrypi ~/DVK512/LED/bcm2835 \$ sudo ./led

2) 可以看到 4 个 LED 交替闪烁,



3) 按键盘 Ctrl+C 结束实验。

wiringPi 程序:

1) 在终端执行:

pi@raspberrypi ~/DVK512/LED/wiringPi \$ sudo ./led

- 2) 可以看到 4 个 LED 交替闪烁。
- 3) 按键盘 Ctrl+C 结束实验。

Python 程序:

- 在终端执行:
 pi@raspberrypi ~/DVK512/LED/python \$ sudo python led.py
- 2) 可以看到 4 个 LED 交替闪烁。
- 3) 按键盘 Ctrl+C 结束实验。

shell 程序:

- 在终端执行以下命令点亮 LED0
 pi@raspberrypi ~/DVK512/LED/shell \$ sudo ./LED.sh 26 1
- 在终端执行以下命令熄灭 LED0
 pi@raspberrypi ~/DVK512/LED/shell \$ sudo ./LED.sh 26 0

2.3. 按键实验

bcm2835 程序:

- 1) 在终端执行: pi@raspberrypi ~/DVK512/KEY/bcm2835 \$ sudo ./key
- 2) 终端会显示是否有按键被按下。
- 3) 按键盘 Ctrl+C 结束实验。

wiringPi 程序:

1) 在终端输入:

pi@raspberrypi ~/DVK512/KEY/bcm2835 \$ sudo ./key

- 2) 终端会显示是否有按键被按下。
- 3) 按键盘 Ctrl+C 结束实验。

python 程序:



1) 在终端输入:

pi@raspberrypi ~/DVK512/KEY/bcm2835 \$ sudo ./key

- 2) 终端会显示是否有按键被按下。
- 3) 按键盘 Ctrl+C 结束实验。

2.4. 8路电平转换模块实验

1) 将 Logic Level Converter 接入 DVK512 的 8I/Os 接口(注意 VCCA 与 8I/Os 接口的 3V3 对应),使用杜邦线连接:

Logic Level Converter	DVK512
VCCB	5V
GND	GND
во	LEDO
B1	LED2
В3	LED3
B4	КЕҮО
В5	KEY1
B6	KEY2
В7	DEY3

表 1. Logic Level Converter 和 DVK512 管脚对应表

如下图所示:



图 3. Logic Level Converter 和 DVK512 管脚连接图

2) 在终端执行对应的程序:

执行 bcm2835 程序

pi@raspberrypi ~/DVK512/Logic-Converter/bcm2835 \$ sudo ./Logic_Converter 执行 wringPi 程序



执行 python 程序

pi@raspberrypi ~/DVK512/Logic-Converter/python \$ sudo python Logic_Converter.py

- 3) 按 DVK512 上的 KEY0-KEY3, 相应的 LED 会亮。
- 4) 按键盘 Ctrl+C 结束实验。

2.5. 摇杆实验

- 1) 将 Mix Board 接入 8I/Os 接口:
- 2) 在终端执行对应的程序:

执行 bcm2835 程序

pi@raspberrypi ~/DVK512/JOYSTICK/bcm2835 \$ sudo ./joystick

执行 wringPi 程序

pi@raspberrypi ~/DVK512/JOYSTICK/wiringPi \$ sudo ./joystick

执行 python 程序

pi@raspberrypi ~/DVK512/JOYSTICK/python \$ sudo python joystick.py

3) 按下或者摇动 Mix Board 上面的摇杆,终端会有相应的显示。

2.6. 蜂鸣器与 PWM 实验

- 1) 将 Mix Board 接入 8I/Os 接口。
- 2) 在终端执行对应的程序:
 - 执行 wringPi 程序

pi@raspberrypi ~/DVK512/Buzzer_PWM/wiringPi \$ sudo ./Buzzer

执行 python 程序

pi@raspberrypi ~/DVK512/Buzzer_PWM/python \$ sudo python buzzer.py

- 3) Mix Board 上的蜂鸣器响。
- 4) 按键盘 Ctrl+C 结束实验

2.7. DS18B20 实验

1) 将 Mix Board 接入 8I/Os 接口。



2) 将 DS18B20 插到 Mix Board 的 ONE-WIRE 接口上,如下图红框所示:

图 4. 将 DS18B20 插到 Mix Board 的 ONE-WIRE 接口上

危险:请务必检查 DS18B20 是否接反。如果接反将会产生上百摄氏度的高温,此时如果用 手指直接触碰 DS18B20 的话,可能会烫伤手指。请务必保证 DS18B20 有弧度的一面和 Mix Board 上的图示相对应,如图 4.将 DS18B20 插到 Mix Board 的 ONE-WIRE 接口上红框中所 示。

 a) 在终端执行对应的程序: 执行 sysfs 程序
 pi@raspberrypi ~/DVK512/DS18B20/fs \$ sudo ./ds18b20
 执行 python 程序
 pi@raspberrypi ~/DVK512/DS18B20/python \$ sudo python ds18b20.py



4) 终端上会把当前环境的温度打印出来。(注:关于 DS18B20 的详细教程可以查看树莓派系列教程)

2.8. 红外遥控实验

- 1) 将 Mix Board 接入 8I/Os 接口。
- 2) 将红外接收头插到 Mix Board 的 IRM 接口上,如下图红框所示:



图 5. 将红外接收头插到 Mix Board 的 IRM 接口上

注意:请务必保证红外接收头有弧度的一面和 Mix Board 上的图示相对应,如图 5. 将红外接收头插到 Mix Board 的 IRM 接口上 红框中所示。

3) 在终端执行对应的程序:

执行 bcm2835 程序

pi@raspberrypi ~/DVK512/IRM/bcm2835 \$ sudo ./irm

执行 wiringPi 程序

pi@raspberrypi ~/DVK512/IRM/wiringPi \$ sudo ./irm

执行 python 程序

pi@raspberrypi ~/DVK512/IRM/python \$ sudo python irm.py



4) 按下配套的红外遥控器上的按键,终端会把解码的红外信号以 16 进制打印出来,如下图所示:

root@raspberrypi:/home/pi# IRM							
irm test start:							
get	the	irm	key	code-hex	oxe7		
get	the	irm	key	code-hex	oxe1		
get	the	irm	key	code-hex	oxad		

图 6. 终端显示按下的遥控器按键

5) 按键盘 Ctrl+C 结束实验。

2.9. LCD1602 实验

1) 将 LCD1602 插到 DVK512 的 LCD1602 接口,如下图所示:



图 7. 将 LCD1602 插到 DVK512 的 LCD1602 接口

2) 在终端执行:

pi@raspberrypi ~/DVK512/LCD1602 \$ sudo ./lcd1602

3) LCD1602 会显示。如果没有显示,用户可以调节 1602 的电位器(如上图红框所示)。

2.10. PCF8563 RTC 实验

- 1) 设置 DVK512 的跳线:
 - RTC_SDA 连接 SDA
 - RTC_SCL 连接 SCL

- 安装 i2c-tools 如镜像没安装 i2c-tools,则执行以下命令: root@ raspberrypi:/# apt-get install i2c-tools
- 3) 在终端执行:

root@ raspberrypi:/# i2cdetect -y 1

4) 终端会打印出 PCF8563 接入树莓派 B+的地址,下图红框中显示 51,说明树莓派已经识别到 PCF8563 了。

root	t@ra	aspł	bern	rypi	i://	nome	e/pi	i# :	i.2 co	lete	ect	-¥	1				
	0	1	2	3	4	5	6	7	8	9	a	b	С	d	e	f	
00:																	
10:																	
20:																	
30:												UU					
40:																	
50:		51															
60:																	
70:																	

图 8. 树莓派 B+识别到 PCF8563 模块

5) 在终端执行对应的程序:

执行 bcm2835 程序

pi@raspberrypi ~/DVK512/PCF8563/bcm2835 \$ sudo ./pcf8563

执行 wiringPi 程序

pi@raspberrypi ~/DVK512/PCF8563/wiringPi \$ sudo ./pcf8563

执行 python 程序

pi@raspberrypi ~/DVK512/PCF8563/python \$ sudo ./pcf8563.py 终端显示 pcf8563 的时间

2.11. PCF8591 AD 实验

1) 将 PCF8591 模块接到 DVK512 的 I2C 接口上。

- 2) 使用杜邦线连接 PCF8591 模块上的 AINO 与 ADO 管脚,如下图所示:

图 9. 连接 PCF8591 模块上的 AINO 与 ADO 管脚

3) 在终端执行对应的程序:

执行 bcm2835 程序

4)

```
pi@raspberrypi ~/DVK512/PCF8591/ADC/bcm2835 $ sudo ./pcf8591
执行 wiringPi 程序
pi@raspberrypi ~/DVK512/PCF8591/ADC/wiringPi $ sudo ./pcf8591
执行 python 程序
pi@raspberrypi ~/DVK512/PCF8591/ADC/python $ sudo python pcf8591.py
终端将打印出 ADCO-ADC3(对应 PCF8591 模块上的 AINO-AIN3)的 AD 值,如下图:
```

```
root@raspberrypi:/home/pi# PCF8591-ADC
start....
adc0: 2.36 adc1: 1.33 adc2: 0.00 adc3:
                                          0.00
adc0: 2.36 adc1: 1.32 adc2: 0.00 adc3:
                                         0.00
adc0: 2.36 adc1: 1.32 adc2: 0.00 adc3:
                                          0.00
      2.36 adc1: 1.33 adc2:
adc0:
                              0.00 adc3:
                                          0.00
      2.36 adc1:
                  1.33 adc2:
adc0:
                              0.00 adc3:
                                          0.00
                              0.00 adc3:
      2.36 adc1:
                  1.32 adc2:
                                          0.00
adc0:
adc0:
                                         0.00
      2.36 adc1: 1.32 adc2: 0.00 adc3:
                                          0.00
adc0: 2.36 adc1: 1.33 adc2: 0.00 adc3:
adc0: 2.36 adc1: 1.33 adc2: 0.00 adc3:
                                         0.00
```

图 10. 终端打印 ADC0-ADC3 的 AD 值



2.12. PCF8591 DA 实验

- 1) 将 PCF8591 模块接到 DVK512 的 I2C 接口上。
- 2) 连接 PCF8591 模块 AOUT 管脚与 DVK512 的 LED 管脚,如下图所示



图 11. 连接 PCF8591 模块 AOUT 管脚与 DVK512 的 LED 管脚

 3) 在终端执行对应的程序: 执行 bcm2835 程序
 pi@raspberrypi ~/DVK512/PCF8591/DAC/bcm2835 \$ sudo ./pcf8591
 执行 wiringPi 程序
 pi@raspberrypi ~/DVK512/PCF8591/DAC/wiringPi \$ sudo ./pcf8591



执行 python 程序

pi@raspberrypi ~/DVK512/PCF8591/DAC/python \$ sudo python pcf8591.py

4) 可以看到终端开始打印数模转换值,此时 PCF8591 模块 AOUT 管脚有相应的电平变化,从 而使 LED 亮度发生变化。

2.13. 重力加速度与磁力计 LSM303DLHC 模块

1) 将 LSM303DLHC 模块接到 I2C 接口,如下图所示:



图 12. 将 LSM303DLHC 接到 I2C 接口



注意:LSM303DLHC 模块有两排管脚,此处连接到 DVK512 的是 SDA 和 SCL 所在的那一排管 脚。

2) 在终端执行对应的程序:

3)

```
执行 bcm2835 程序
pi@raspberrypi ~/DVK512/LSM303DLHC/bcm2835 $ sudo ./LSM303DLHC
执行 wiringPi 程序
pi@raspberrypi ~/DVK512/LSM303DLHC/wiringPi $ sudo ./LSM303DLHC
执行 python 程序
pi@raspberrypi ~/DVK512/LSM303DLHC/python $ sudo python LSM303DLHC.py
可以看到终端显示重力加速度与磁力计的值,如下图所示:
```



图 13. 终端打印重力加速度与磁力计的值

2.14. 角速度传感器 L3G4200D 模块

- 1) 将 L3G4200D 模块接到 SPI 接口。
- 2) 在终端执行对应的程序:

执行 bcm2835 程序

pi@raspberrypi ~/DVK512/L3G4200D/bcm2835 \$ sudo ./L3G4200D

执行 wiringPi 程序

pi@raspberrypi ~/DVK512/L3G4200D/wiringPi \$ sudo ./L3G4200D

执行 python 程序

pi@raspberrypi ~/DVK512/L3G4200D/python \$ sudo python L3G4200D.py

3) 终端打印出三轴角速度值:



图 14. 终端打印三轴角速度值



2.15. 串口实验

- (注: 树莓派的串口默认为终端调试,如要启用串口功能,需关闭串口串口终端调试功能)
- 1) 通过 mini USB 线把 DVK512 的 UART TO USB 接口连接到电脑。
- 2) 在终端执行对应的程序:

执行 wiringPi 程序 pi@raspberrypi ~/DVK512/UART/wiringPi \$ sudo ./UART 执行 python 程序 pi@raspberrypi ~/DVK512/UART/python \$ sudo ./uart.py

在 windows 中打开串口助手,选择正确的串口号,波特率设置为 115200。
 发送字符会回显到串口助手中。

2.16. GPS 实验

 系统镜像默认是把串口作为终端调试使用的,想要控制外设需要先禁用串口的调试功能。 执行:

```
sudo raspi-config
选择选择 Advanced Options -> Serial ->no 关闭串口终端调试
关闭串口终端调试功能后则不能再通过串口登陆访问树莓派,需开启后才能通过串口控制
树莓派。
```

2) 插入 GPS 模块到 UART 接口,同时需要拔掉 UART JMP 处跳线。如下图红框所示:



图 15. 拔掉 UART JMP 处跳线

a) 在终端执行以下命令设置波特率(系统重启后,串口波特率将会恢复默认的115200)
 root@raspberrypi:/# stty -F /dev/ttyAMA0 38400
 root@raspberrypi:/# gpsd /dev/ttyAMA0 -F /var/run/gpsd.sock

- 在终端执行以下命令,打开 Linux 系统自带的 GPS 软件: root@raspberrypi:/# cgps -s
- 5) 终端打印 GPS 定位信息:

8	picialpocitypi.								
1qq	Iddddddddddddddd	199999999999999999999999999999999999999	qqqqklq	qqqq	qqqqqqq	laaaaaa	laadada	iqqqqqq	lqk
х		14-07-28T14:43:27.000	Z xxP	RN:	Elev:	Azim:	SNR:	Used:	x
x	Latitude: 22	2.537478 N 1	XX	2	47	106	17	Y	x
x	Longitude: 114	1.075020 E 🤈	XX	5	42	029	14	Y	х
x	Altitude: 4.2	2 m 3	XX	6	12	121	36	Y	х
x	Speed: 0.5	5 kph	XX	8	04	049	00	Y	х
x	Heading: 0.0) deg (true)	XX	10	05	066	00	Y	х
x	Climb: 0.0) m/min	XX	12	02	206	00	Y	х
x	Status: 3D	FIX (27 secs)	XX	15	62	225	33	Y	х
x	Longitude Err:	+/- 2 m	XX	18	04	270	20	Y	х
x	Latitude Err:	+/- 2 m	XX						х
x	Altitude Err:	+/- 6 m	XX						x
х	Course Err:	n/a	XX						х
x	Speed Err:	+/- 19 kph	XX						x
х	Time offset:	0.219	XX						х
х	Grid Square:	OL72am	XX						x
mqc	aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa		aaaa ima	qqqq	aaaaaaaa	Idadadad	laaaaac	Igggggg	iq i

图 16. 终端打印 GPS 定位信息



版权声明

本手册所有权由深圳市微雪电子有限公司独家持有。未经本公司的书面许可,不得以任何方式或 形式进行修改、分发或复制本文档的任何部分,否则一切后果由违者自负。

版本更新记录

版本	日期	说明
V1.0	2014.08.18	初始发布
V1.1	2014.11.13	新增图片与手册内容
V2.1	2014.11.27	重大更新
V2.2	2014.12.03	文字修改
V2.3	2014.12.18	新增 3.5inch LCD
V2.4	2015.01.21	修正 2.13 节局部
V2.5	2015.03.06	兼容树莓派 2 代
V2.6	2015.09.06	删除过时的内容