



DVK512 Expansion Board For Raspberry Pi

User Manual

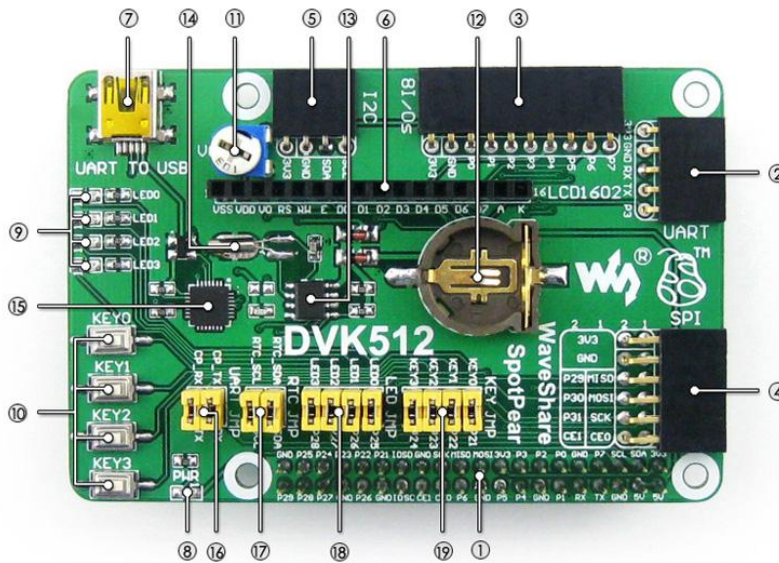
DVK512 is an expansion board designed for Raspberry Pi Model B+, integrates various components and interfaces for connecting external accessory boards. It's ideal for Raspberry Pi Model B+ evaluation and development.



Supported Pi:

- Raspberry Pi 1 Model A+
- Raspberry Pi 1 Model B+
- Raspberry Pi 2 Model B

WHAT'S ON THE DVK512



1. **Pinheaders for connecting with the RPi**
2. **UART interface:** easily connects to UART modules such as RS232, RS485, USB TO UART, etc.
3. **8I/Os interface:** easily connects to modules controlled by I/Os, such as 8 Push Buttons, Logic Level Converter, Mix Board, etc.
4. **SPI interface:** easily connects to SPI modules such as AT45DBXX Dataflash, L3G4200D Board, etc.
5. **I2C interface:** easily connects to I2C modules such as PCF8574 Expansion Board, PCF8563 RTC Board, LSM303DLHC Board, etc.
6. **Character LCD interface:** for connecting character LCDs like LCD1602
7. **USB connector:** USB TO UART, supported by onboard converter CP2102
8. **Power indicator**
9. **User LEDs**
10. **User Keys**
11. **Potentiometer:** for LCD1602 contrast adjustment
12. **RTC battery holder**
13. **PCF8563:** onboard RTC chip
14. **32.768K crystal:** RTC crystal
15. **CP2102:** onboard USB TO UART chip, for debugging
16. **CP2102 jumper**
17. **RTC jumper**
18. **User LEDs jumper**
19. **User Keys jumper**

TABLE OF CONTENTS

What's on the DVK512	2
1. Basic operations.....	4
1.1. System image file programming.....	4
1.2. Serial debugging environment deploying	4
2. DVK512 and expansion function Demos	6
2.1. Installing the libraries required	6
2.2. LED Demo	7
2.3. Key Demo.....	7
2.4. 8-channel Logic Level Convertor Demo	8
2.5. Joystick Demo	10
2.6. Buzzer and PWM.....	10
2.7. DS18B20 Demo	10
2.8. Infrared remote control	12
2.9. LCD1602 Demo.....	13
2.10. PCF8563 RTC Demo	13
2.11. PCF8591 AD Demo	14
2.12. PCF8591 DA Demo	16
2.13. Acceleration of Gravity and Magnetic Sensors LSM303DLHC	17
2.14. Angular rate Sensor L3G4200D.....	18
2.15. UART Demo.....	19
2.16. GPS Demo	19
Copy Right.....	21
Revision History	21

1. BASIC OPERATIONS

1.1. SYSTEM IMAGE FILE PROGRAMMING

- 1) Run the CD provided with the module. Then, find out the file with the expansion name .img under the directory of IMAGE and copy it to your PC (For the latest system image file, please login the website: <http://www.waveshare.net/wiki/DVK512>)

- 2) Format your TF card with the SDFormatter.exe.

Notices: The capability of TF card in used here should be more than 4GB. In this operation, a TF card reader is also required, which has to be purchased separately.

- 3) Start the Win32DiskImager.exe, and select the system image file copied into your PC, then, click the button **Write** to program the system image file.

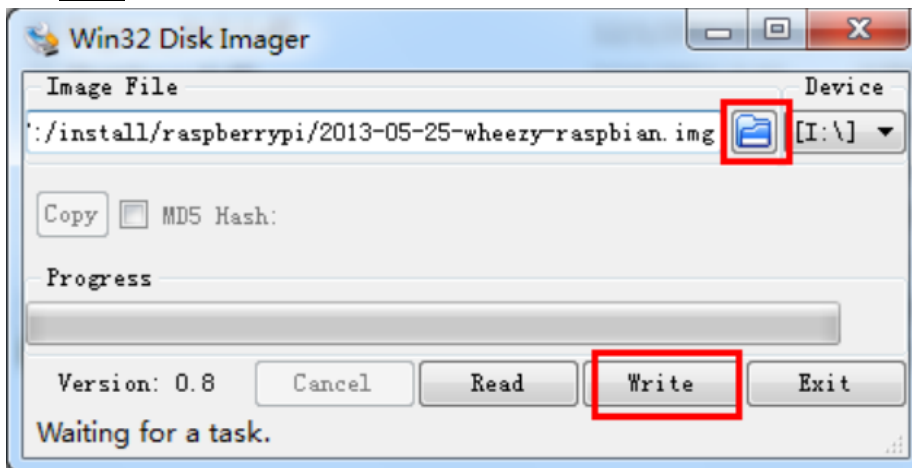


Figure 1: Programming the system image file with Win32DiskImager.exe

1.2. SERIAL DEBUGGING ENVIRONMENT DEPLOYING

- 1) Connect your PC to the UART TO USB interface on DVK512 via a mini USB cable.

Notices: The USB part of the UART TO USB interface applied by DVK512 is a mini USB interface, while the one on Raspberry Pi board is a micro USB interface. Please take a note.

- 2) Install the cp2102_driver.
- 3) Start the software PuTTY.exe, and configure the following parameters:
 - Serial line: it is used to select corresponding serial port. The serial port in used can be check by Device Manager.
 - Speed: 115200
 - Connection type: Serial

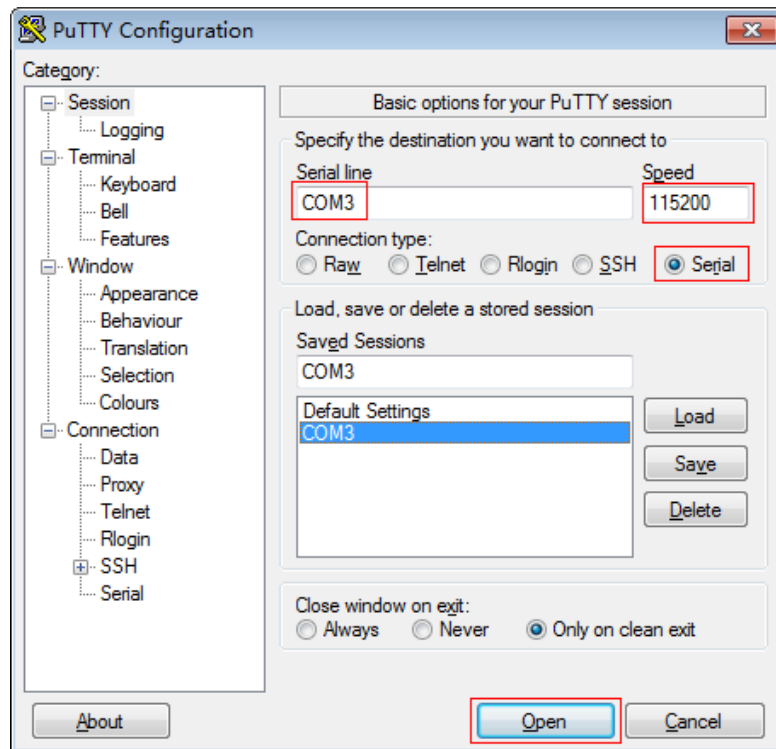


Figure 2: PuTTY settings

- 4) After the system is started up, you should input following information: User name: pi and Password: raspberry. Then, you can enter the serial terminal to communicate with the Raspberry Pi.

Notices: In this document, the software PuTTY is always used to control the Raspberry Pi via serial port communication, unless otherwise specified. When a serial port is used for Raspberry Pi terminal debugging, it cannot service as a common serial port any more, since it is occupied by system debugging function.

2. DVK512 AND EXPANSION FUNCTION DEMOS

Before performing any operations described in this section, you should connect DVK512 to RPi with onboard pin headers, unless otherwise specified.

2.1. INSTALLING THE LIBRARIES REQUIRED

Before using the DVK512 programs, you shall install the the bcm2835, wiringPi and python libraries to the RPi to add the additional APIs, and configure the settings to start up the core drivers of I2C, SPI and UART after the libraries installed. For more detailed information about the installation and configuration of the RPi library functions, please refer to [Libraries Installation for RPi](#). Of course, we have provided a system image file with the libraries installed as well. In case that you don't want to reinstall the libraries, you can program this ready-to-use system image file to your Raspberry Pi board. You can find the API source code in the directory /home/pi/DVK512 only if using the ready-to-use system image.

Note: Each demo should be run under the corresponding directory, e.g. to use the bcm2835 demo of the LED, execute the command:

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

This means that, you should first enter the path ~/DVK512/LED/bcm2835 than execute the command `sudo ./led` (./led is a compiled executable file).

The sudo means that the command is executed by a super user. The sudo is not required if you log in as a root user, e.g. `root@raspberrypi:/home/pi/DVK512/LED/bcm2835# ./led`

Unless otherwise specified, the following demos are pre-compiled to executable files which can be run directly. If you intend to compile the C program of bcm2835, wiringPi, sysfs, etc. then please enter the demo directory and execute the command "make" to compile the program while the command "make clean" to remove the executable files. For example:

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

```
rm led
```

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

```
gcc -Wall led.c -o led -lbcm2835
```

Python program can be run directly without compilations, e.g.

```
pi@raspberrypi ~/DVK512/LED/python $ sudo python led.py
```

2.2. LED DEMO

bcm2835 program:

- 1) Enter the Linux terminal, and run the following commands:

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

- 2) The 4 LEDs will light up one by one.
- 3) Press the keys **Ctrl+C** to end the program.

wiringPi program:

- 1) Enter the Linux terminal, and run the following commands:

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

- 2) The 4 LEDs will light up one by one.
- 3) Press the keys **Ctrl+C** to end the program.

Python program:

- 1) Enter the Linux terminal, and run the following commands:

```
pi@raspberrypi ~/DVK512/LED/python $ sudo python led.py
```

- 2) The 4 LEDs will light up one by one.
- 3) Press the keys **Ctrl+C** to end the program.

shell program:

- 1) Turn on LED0 by this command:

```
pi@raspberrypi ~/DVK512/LED/shell $ sudo ./LED.sh 26 1
```

- 2) Turn off LED0 by this command:

```
pi@raspberrypi ~/DVK512/LED/shell $ sudo ./LED.sh 26 0
```

2.3. KEY DEMO

bcm2835 program:

- 1) Enter the Linux terminal, and run the following commands:

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

- 2) The terminal will show whether there is key-press.
- 3) Press the keys **Ctrl+C** to end the program.

wiringPi program:

- 1) 在终端输入:

```
pi@raspberrypi ~/DVK512/KEY/bcm2835 $ sudo ./key
```
- 2) The terminal will show whether there is key-press.
- 3) Press the keys **Ctrl+C** to end the program.

Python program:

- 1) 在终端输入:

```
pi@raspberrypi ~/DVK512/KEY/bcm2835 $ sudo ./key
```
- 2) The terminal will show whether there is key-press.
- 3) Press the keys **Ctrl+C** to end the program.

2.4. 8-CHANNEL LOGIC LEVEL CONVERTOR DEMO

- 1) Connect the 8-channel Logic Level Converter to the onboard 8I/Os interface, as Figure 3 shows (Notices: The VCCA pin header on the Logic Level Converter should be connected to the 3V3 pin header of the 8I/Os connector on DVK512).

Logic Level Converter	DVK512
VCCB	5V
GND	GND
B0	LED0
B1	LED2
B3	LED3
B4	KEY0
B5	KEY1
B6	KEY2
B7	DEY3

Table 1.: Pin relationships between Logic Level Converter and DVK512



Figure 3: The Connection between Logic Level Converter and DVK512

- 2) Enter the terminal, and execute corresponding commands:

bcm2835 program:

```
pi@raspberrypi ~/DVK512/Logic-Converter/bcm2835 $ sudo ./Logic_Converter
```

wiringPi program:

```
pi@raspberrypi ~/DVK512/Logic-Converter/wiringPi $ sudo ./Logic_Converter
```

python program:

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

- 3) Press the KEY0-KEY3 on the DVK512, and the relative LED will light up.
- 4) Press the keys **Ctrl+C** to end the program.

2.5. JOYSTICK DEMO

- 1) Connect the Mix Board to the 8I/Os interface.
- 2) Enter the terminal, and execute corresponding commands:

bcm2835 program:

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

wiringPi program:

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

python program:

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

- 3) Press or move the joystick, the terminal will show the action.

2.6. BUZZER AND PWM

- 1) Connect the Mix Board to the 8I/Os interface.
- 2) Enter the terminal, and execute corresponding commands:

wiringPi program:

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

python program:

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

- 3) The buzzer on the Mix Board will buzz in 4 seconds.
- 4) Press the keys **Ctrl+C** to end the program.

2.7. DS18B20 DEMO

- 1) Connect the Mix Board to the 8I/Os interface.

- 2) Insert the DS18B20 to the ON-WIRE interface on Mix Board.



Figure 4: Inserting the DS18B20 to the ON-WIRE interface on Mix Board

Dangerous: Please make sure the DS18B20 is connected correctly. The DS18B20 in wrong connection may generate a high temperature more than 100°C which may cause injuries when you touch it. The figure shows the correct connection between the DS18B20 and the Mix Board, in which the DS18B20 should be placed according to the shape of the icon shown on the ONE-WIRE interface of Mix Board.

- 3) Enter the terminal, and execute corresponding commands:

sysfs program:

```
pi@raspberrypi ~/DVK512/DS18B20/fs $ sudo ./ds18b20
```

python program:

```
pi@raspberrypi ~/DVK512/DS18B20/python $ sudo python ds18b20.py
```

- 4) The terminal will display the current temperature information.

2.8. INFRARED REMOTE CONTROL

- 1) Connect the Mix Board to the 8I/Os interface.
- 2) Connect the infrared receiver to the IRM interface on Mix Board.



Figure 5: Connecting the infrared receiver to the IRM interface on Mix Board

Notices: The infrared receiver should be placed according to the shape of the icon shown on the IRM interface of Mix Board.

- 3) Enter the terminal, and execute corresponding commands:

bcm2835 program:

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

wiringPi program:

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

python program:

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

- 4) Press any key on the infrared remote controller provided by Waveshare. The terminal will display the decoded infrared signal in hexadecimal format.

```
root@raspberrypi:/home/pi# IRM
irm test start:
get the irm key code-hex 0xe7
get the irm key code-hex 0xe1
get the irm key code-hex 0xad
```

Figure 6: Terminal displays the key pressed on the infrared remote controller

- 5) Press the keys **Ctrl+C** to end the program.

2.9. LCD1602 DEMO

- 1) Insert the LCD1602 into the LCD1602 interface on DVK512.



Figure 7: Inserting the LCD1602 into the LCD1602 interface on DVK512

- 2) Enter the Linux terminal, and run the following commands:


```
pi@raspberrypi ~/DVK512/LCD1602 $ sudo ./lcd1602
```
- 3) LCD1602 will display relative information. If there is nothing shown, please adjust the potentiometer on the LCD1602. The red box in Figure 11 shows the position of the potentiometer.

2.10. PCF8563 RTC DEMO

- 1) Set the jumpers on DVK512:
 - Connect RTC_SDA to SDA

- Connect RTC_SCL to SCL
- 2) Install i2c-tools:
If you have not installed i2c-tools yet, please enter the terminal, and input:
`root@ raspberrypi:/# apt-get install i2c-tools`
 - 3) Enter the Linux terminal, and run the following commands:
`root@ raspberrypi:/# i2cdetect -y 1`
 - 4) Then, you will see the device address of PCF8563 connected to Raspberry Pi module. Here, the device address of PCF8563 is 51, indicating that the PCF8563 is detected by Raspberry Pi.

```

root@raspberrypi:/home/pi# i2cdetect -y 1
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  UU  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  51  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
    
```

Figure 8: PCF8563 is detected by Raspberry Pi B+ module

- 5) Enter the terminal, and execute corresponding commands:
bcm2835 program:
`pi@raspberrypi ~/DVK512/PCF8563/bcm2835 $ sudo ./pcf8563`
wiringPi program:
`pi@raspberrypi ~/DVK512/PCF8563/wiringPi $ sudo ./pcf8563`
python program:
`pi@raspberrypi ~/DVK512/PCF8563/python $ sudo ./pcf8563.py`
 Time of PCF8563 will be shown on the terminal.

2.11. PCF8591 AD DEMO

- 1) Connect the PCF8591 module to the I2C interface on DVK512.

- 2) Connect the pins AIN0 and AD0 on the PCF8591 together via a cable.

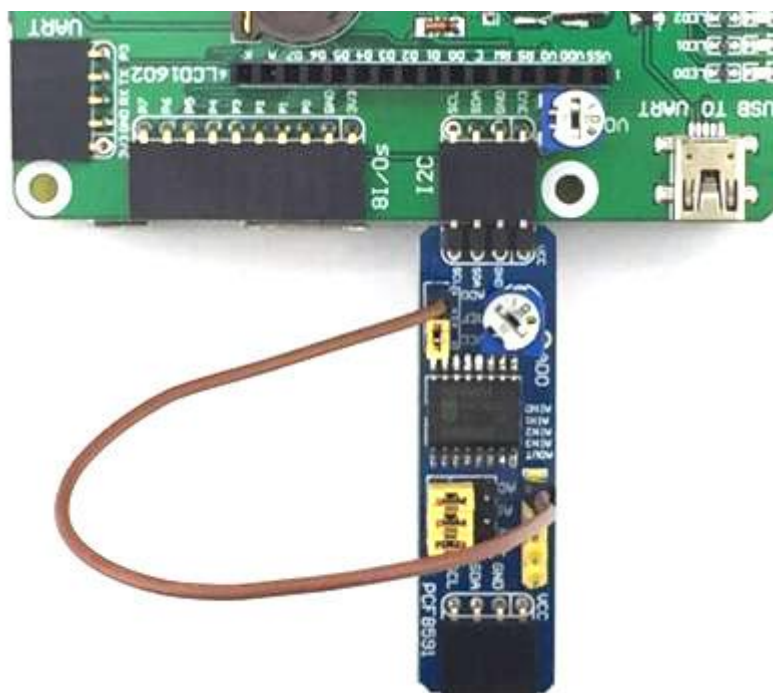


Figure 9: Connecting the pins AIN0 and AD0 together

- 3) Enter the terminal, and execute corresponding commands:

bcm2835 program:

```
pi@raspberrypi ~/DVK512/PCF8591/ADC/bcm2835 $ sudo ./pcf8591
```

wiringPi program:

```
pi@raspberrypi ~/DVK512/PCF8591/ADC/wiringPi $ sudo ./pcf8591
```

python program:

```
pi@raspberrypi ~/DVK512/PCF8591/ADC/python $ sudo python pcf8591.py
```

- 4) The terminal will display the AD values read from ADC0-ADC3 (corresponding to the pins AIN0-AIN3 on PCF8591).

```
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
adc0:  2.36  adc1:  1.33  adc2:  0.00  adc3:  0.00
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
adc0:  2.36  adc1:  1.33  adc2:  0.00  adc3:  0.00
adc0:  2.36  adc1:  1.33  adc2:  0.00  adc3:  0.00
```

Figure 10: Displaying the AD values read from ADC0-ADC3

2.12. PCF8591 DA DEMO

- 1) Connect the PCF8591 module to the I2C interface on DVK512.
- 2) Connect the Pin AOUT on PCF8591 to the LED pin header on DVK512.



Figure 11: Connecting the Pin AOUT on PCF8591 to the LED pin header on DVK512

- 3) Enter the terminal, and execute corresponding commands:

bcm2835 program:

```
pi@raspberrypi ~/DVK512/PCF8591/DAC/bcm2835 $ sudo ./pcf8591
```


wiringPi program:

```
pi@raspberrypi ~/DVK512/PCF8591/DAC/wiringPi $ sudo ./pcf8591
```

python program:

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

- 4) The digital conversion value is displayed in the terminal. And the brightness of the LED indicator on DVK512 will change with the level variation on the pin AOUT.

2.13. ACCELERATION OF GRAVITY AND MAGNETIC SENSORS LSM303DLHC

- 1) Connect the LSM303DLHC module to the I2C interface on DVK512.



Figure 12: Connecting the LSM303DLHC module to the I2C interface on DVK512

Notices: LSM303DLHC module has two rows of pins, and the one connected to DVK512 contains the pins SDA and SCL.

- 2) Enter the terminal, and execute corresponding commands:

bcm2835 program:

```
pi@raspberrypi ~/DVK512/LSM303DLHC/bcm2835 $ sudo ./LSM303DLHC
```

wiringPi program:

```
pi@raspberrypi ~/DVK512/LSM303DLHC/wiringPi $ sudo ./LSM303DLHC
```

python program:

```
pi@raspberrypi ~/DVK512/LSM303DLHC/python $ sudo python LSM303DLHC.py
```

- 3) The terminal will display the relative values read from acceleration of gravity and magnetic sensor.

```
*****          LSM303DLHC          *****
X_A=-0.320000 m/s^2
Y_A=0.280000 m/s^2
Z_A=10.200000 m/s^2
X_M=-332.394379 T
Y_M=-107.042252 T
Z_M=-247.887329 T
north=208.116699 degree
```

Figure 13: Relative values read from acceleration of gravity and magnetic sensor

2.14. ANGULAR RATE SENSOR L3G4200D

- 1) Connect the L3G4200D to the SPI interface on DVK512.
- 2) Enter the terminal, and execute corresponding commands:

bcm2835 program:

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

wiringPi program:

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

python program:

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

- 3) The terminal will display the angular velocity values in three factors of x axis, y axis and z axis, respectively.

```
***** L3G4200D *****
x=-0.140000 rad/s
y=-2.670000 rad/s
z=0.080000 rad/s
```

Figure 14: Angular velocity values

2.15. UART DEMO

Note: The Serial Port of Raspberry pi is used for terminal debugging by default, so for using UART function, the Serial terminal debugging function needs to be closed.

- 1) Connect the onboard UART TO USB port to a PC via a mini USB cable.
- 2) Enter the terminal, and execute corresponding commands:

wiringPi program:

```
pi@raspberrypi ~/DVK512/UART/wiringPi $ sudo ./UART
```

python program:

```
pi@raspberrypi ~/DVK512/UART/python $ sudo ./uart.py
```

- 3) Run Serial monitor software, choose a corresponding COM port and set the baud rate as 115200. The field sent by the software will be sent back and shown on the software.

2.16. GPS DEMO

- 1) In the default boot mode of the system image provided by Waveshare, the serial port works as terminal debugging output. In order to control peripherals via serial port communication, you should input corresponding commands on the terminal to disable the debugging function of the serial port.

```
sudo raspi-config
```

Choose Advanced Options -> Serial ->no to disable the debugging function of the serial port.

Once the debugging function of the serial port is disabled, you cannot access the Pi using Serial port but you can re-enable the debugging function by raspi-config.

- 2) Insert a GPS module into the UART interface on DVK512, and remove the jumper wire from UART JMP.



COPY RIGHT

All rights reserved by Waveshare Electronics Co., Ltd.

Not allow to modify, distribute, or copy without permission.

REVISION HISTORY

Revision	Date	Description
V1.0	Aug. 18, 2014	Initial revision
V1.1	Nov. 13, 2014	Figures and tables update
V2.1	Nov. 27, 2014	Major update
V2.2	Dec. 03, 2014	Amendment
V2.3	Dec. 18, 2014	3.5inch LCD update
V2.4	Jan. 01, 2015	Amend Chap. 2.13
V2.5	Mar. 06, 2015	Compatible with Raspberry Pi 2
V2.6	Sep. 08, 2015	Deleted obsolete content