GOWARRIOR

GoDroid

Application Notes

Python Application Support

v1.0

Copyright Statement

Copyright in this document is owned by GoWarrior Community. Any person is hereby authorised to view, copy, print and distribute this document subject to the following conditions:

- The document may be used for informational purposes only
- The document may be used for non-commercial purposes only
- Any copy of this document or portion thereof must include this copyright notice

This document is provided "as is" without any warranty of any kind, either express or implied, statutory or otherwise; without limiting the foregoing, the warranties of satisfactory quality, fitness for a particular purpose or non-infringement are expressly excluded and under no circumstances will GoWarrior Community be liable for direct or indirect loss or damage of any kind, including loss of profit, revenue, goodwill or anticipated savings.

This document is intended only to assist the reader in the use of the product. GoWarrior Community makes no representations or warranties with respect to the accuracy or completeness of the contents of this document, which is used at your own risk and should not be relied upon. The information could include technical inaccuracies or typographical errors. No license, whether express, implied, arising by estoppel or otherwise, to any intellectual property right is granted by this document.

The product described in this document is subject to continuous development and improvements. GoWarrior Community also reserves the right to make changes to the specifications and product description at any time without notice.

Third-party brands and names mentioned in this publication are for identification purpose only and may be the property of their respective owners.

Android™ is a registered trademark of Google Inc. Linux® is a registered trademark of Linus Torvalds. Microsoft® and Windows® are registered trademarks of Microsoft Corporation. Supply of this product does not convey a license nor imply a right under any patent, or any other industrial or intellectual property right of Google Inc., Linus Torvalds, and Microsoft Corporation to use this implementation in any finished end-user or ready-to-use final product. It is hereby notified that a license for such use is required from Google Inc., Linus Torvalds and Microsoft Corporation.

For the latest version of this document refer to:

www.gowarriorosh.com

Copyright © 2015 GoWarrior Community All Rights Reserved.

Table of Contents

Preface 1

Overview 1

Audience 1

Applicable Products 1

Reference Documents 1

Conventions 2

How to Contact Us 3

1 Introduction 4

2 Connecting LED 5

2.1 Solution 5

2.2 Discussion 6

3 Controlling the Brightness of LED 7

3.1 Solution 7

3.2 Discussion 8

4 QPython 10

4.1 Introduction 10

4.2 Installation 10

5 Installing Pure- Python Package Index 12

5.1 Installing Pure-Python Package Index 12

5.2 Compiling Python Module 13

5.2.1 Prerequisites 13

5.2.2 Creating Your Own Recipes 14

Revision History 18

Document Change History 18

Software Changes 18

List of Tables

Table 1. Typographical Conventions 2
Table 2. Symbol Conventions 2
Table 3. Document Change History 18
Table 4. Software Change History 18

List of Figures

Figure 1. Connecting an LED to the TIGER Board 5
Figure 2. Pulse-width Modulation 8

Preface

Overview

This manual mainly describes the GoWarrior TIGER Python application support.

- **Chapter 1: Introduction**

This chapter focuses on the typical GPIO usages written in Python.

- **Chapter 2: Connecting LED**

This chapter gives compact description on how to connect LED and light it up.

- **Chapter 3: Controlling the Brightness of LED**

This chapter provides general overview and procedure about how to control the brightness of LED.

- **Chapter 4: QPython**

This chapter introduces the QPython and the guide to install it.

- **Chapter 5: Installing Pure-Python Package Index**

This chapter summarizes how to install PyPI (Pure-Python Package Index) through pip console, which is included in QPython by default.

Audience

This manual is primarily written to provide complete guidance for those who wants to exploit GoWarrior TIGER Board, such as makers, tinkers, innovators, students, etc.

Applicable Products

This manual is applicable for the GoWarrior TIGER Board.

Reference Documents

N/A

Conventions

Typographical Conventions

Item	Format
codes, keyboard input commands, file names, equations, and math	`Courier New, Size 10.5`
Variables, code variables, and code comments	*`Courier New, Size, Italic`*
Menu item, buttons, tool names	Ebrima, Size 10.5, Bold e.g. **Select USB Debugging**
Screens, windows, dialog boxes, and tabs	Ebrima, Size 10.5, Bold Enclosed in double quotation marks e.g. Open the **"Debug Configuration"** dialog box

Table 1. Typographical Conventions

Symbol Conventions

Item	Description
Caution	Indicates a potential hazard or unsafe practice that, if not avoided, could result in data loss, device performance degradation, or other unpredictable results.
Note	Indicates additional and supplemental information for the main contents.
Tip	Indicates a suggestion that may help you solve a problem or save your time.

Table 2. Symbol Conventions

How to Contact Us

Submit all your comments and error reports to the author at:

info@gowarriorosh.com

Tel: +886-2-8752-2000

Fax: +886-2-8751-1001

For questions regarding GoWarrior, contact our support team at the email listed below:

support@gowarriorosh.com

1 Introduction

Python is a programming language that lets you work quickly and integrate systems more effectively. The GoDroid Python is based on open source Python, so it is compatible with GoDroid Python. In this document we will not introduce the basis of Python, but we will focus on the typical GPIO usages written in Python.

2 Connecting LED

This chapter describes how to connect LED and lighten it up.

2.1 Solution

Connect the LED to one of the GPIO pins using a 470Ω or 1kΩ series resistor to limit the current. To make this connection, you will need:

- Breadboard and male to male jumper wires
- 470Ω or 1kΩ resistor
- LED

Figure 1 shows how you can wire this using solderless breadboard and male-to-female jumper leads.

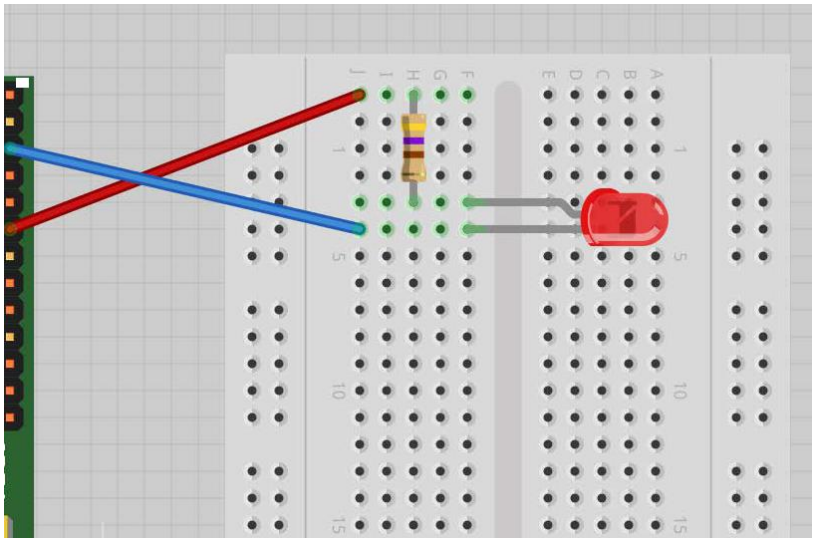


Figure 1. Connecting an LED to the TIGER Board

Having connected the LED, we need to be able to turn it on and off using commands from Python. Please follow the below mention technique to install the RPi. GPIO Python library. Start a Python console from the terminal with super user access and enter following commands:

```
$ sudo python

>>> import RPi.GPIO as GPIO

>>> GPIO.setmode(GPIO.BCM)

>>> GPIO.setup(18, GPIO.OUT)

>>> GPIO.output(18, True)

>>> GPIO.output(18, False)
```

This will make your LED on and off.

2.2 Discussion

LEDs are a very useful, cheap, and efficient way of producing light, but you need to be careful to use them. If the LEDs are connected directly to a voltage source (such as a GPIO output) which is greater than about 1.7 volts, they will draw a very large current. This can often be enough to either destroy the LED or the sour for providing the current —which is not good if your GoWarrior TIGER Board is providing the current.

You should always use a series resistor with an LED because the series resistor is placed between the LED and the voltage source, which controls the amount of current flowing through the LED to the level that is safe for both the LED and the GPIO pin driving it.

3 Controlling the Brightness of LED

This chapter describes how to control the brightness of LED.

3.1 Solution

The RPi.GPIO library has a pulse-width modulation (PWM) feature that allows you to control the power to an LED and its brightness. To try it out, connect the LED as described in chapter 2 and run this test program:

```
import RPi.GPIO as GPIO

led_pin = 18

GPIO.setmode(GPIO.BCM)

GPIO.setup(led_pin, GPIO.OUT)

pwm_led = GPIO.PWM(led_pin, 500)

pwm_led.start(100)

while True:

duty_s = raw_input("Enter Brightness (0 to 100):")

duty = int(duty_s)

pwm_led.ChangeDutyCycle(duty)
```

Run the Python program, and you will be able to change the brightness by entering a number between 0 and 100:

```
python led_brightness.py

Enter Brightness (0 to 100):0

Enter Brightness (0 to 100):20
```

```
Enter Brightness (0 to 100):10

Enter Brightness (0 to 100):5

Enter Brightness (0 to 100):1

Enter Brightness (0 to 100):90
```

You can exit the program by pressing Ctrl+C.

3.2 Discussion

PWM is a clever technique where you vary the length of pulses while keeping the overall number of pulses per second (the frequency in Hz) constant. Figure 2 illustrates the basic principle of PWM.

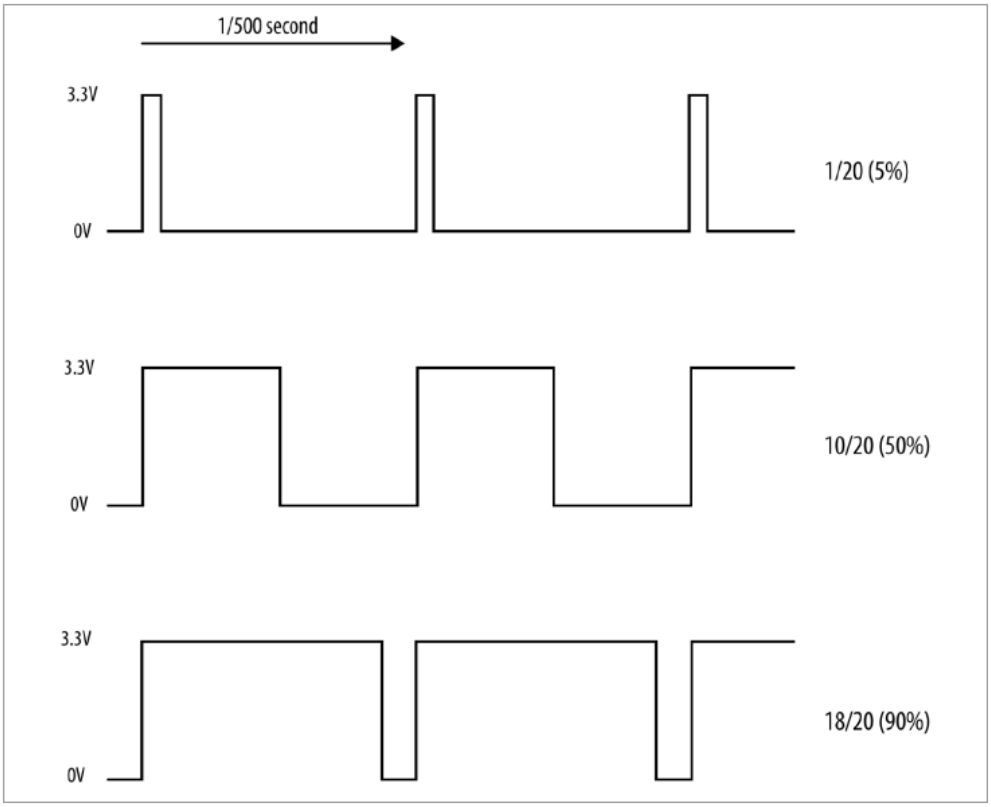


Figure 2. Pulse-width Modulation

At high frequencies, the measured PWM frequency varies somewhat from the frequency supplied as an argument. This may be something that changes in later versions of the PWM feature of RPi.GPIO.

You can change the PWM frequency by modifying following line:

```
pwm_led = GPIO.PWM(led_pin, 500)
```

The value is in the unit of Hz, so in this case, the frequency is set to 500 Hz.

4 QPython

4.1 Introduction

QPython is a script engine which runs Python on android devices. It lets your android device run Python scripts and projects. It contains the Python interpreter, console, editor, and the SL4A Library for Android.

It offers the development kit which lets you easily develop Python projects and scripts on your Android device.

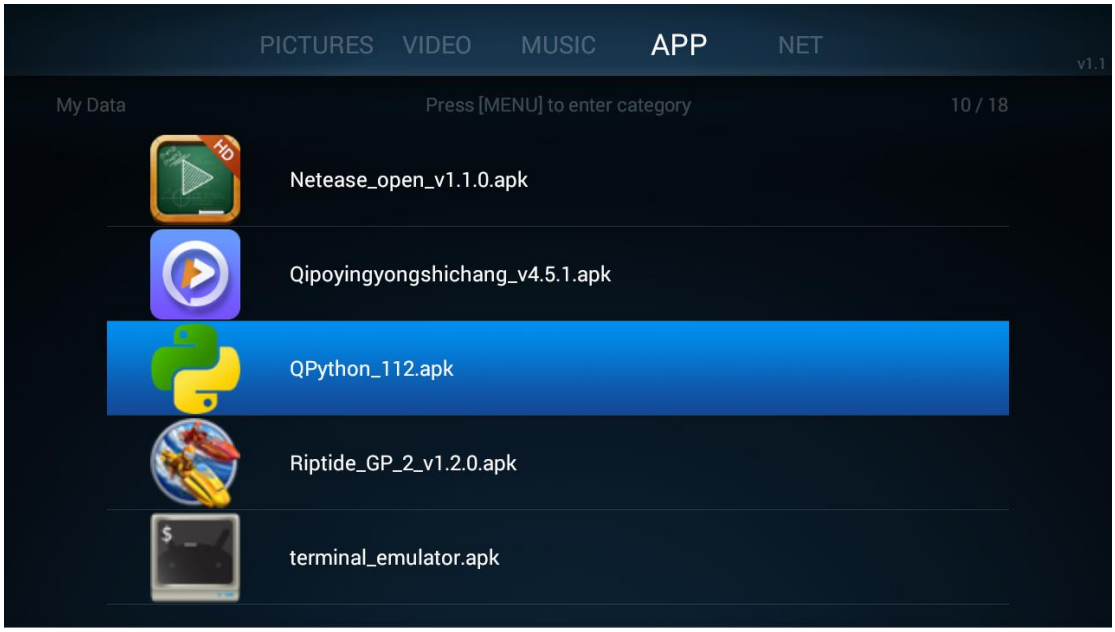
To download the APK and to get documentation please refer to http://qpython.com/

4.2 Installation

In case if the GPIO/I2C/SPIdev functions will be used in QPython, please install the QPython by following the steps mentioned below:

1. Run the **"Tango browser"** from the GoDroid desktop.

2. Find the **"QPython_112.apk"** in the "APP" page, then select it and press "OK" key to start installation.



As compare with the official version, it includes the GPIO/I2C/SPIdev libraries, and able to run Python script with the root privileges.

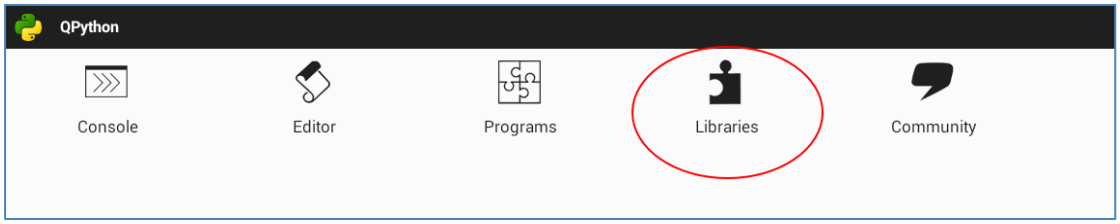
In case if the users don't require these functions, they can download and install QPython from official website, http://qpython.com/

5 Installing Pure-Python Package Index

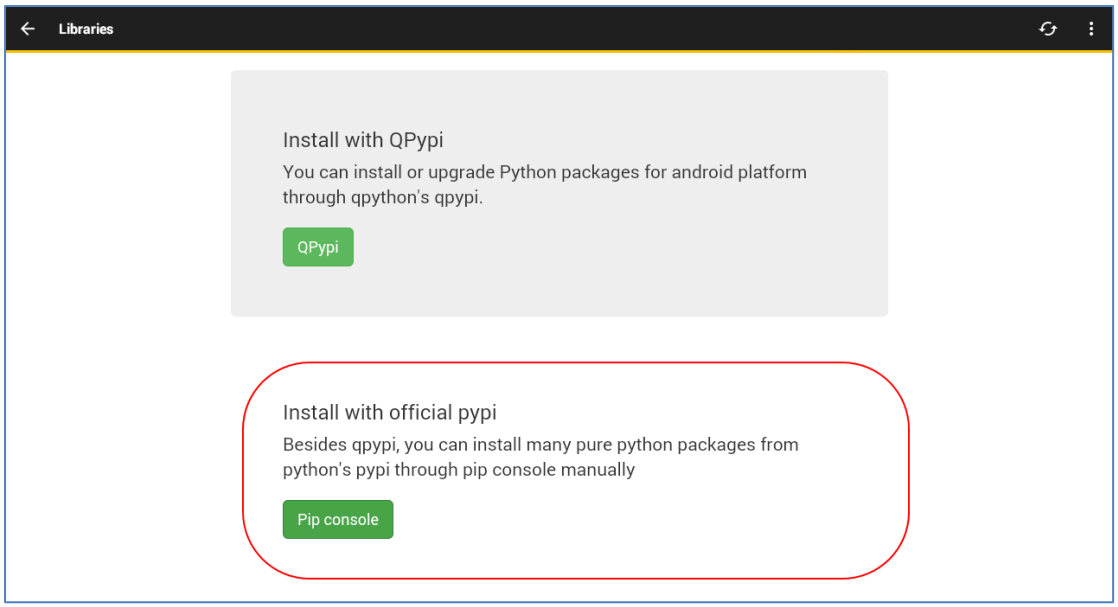
5.1 Installing Pure-Python Package Index

PyPI (Pure-Python Package Index) can be installed through pip console, which is included in QPython by default.

First of all run "QPython"; then select the "Libraries" icon.

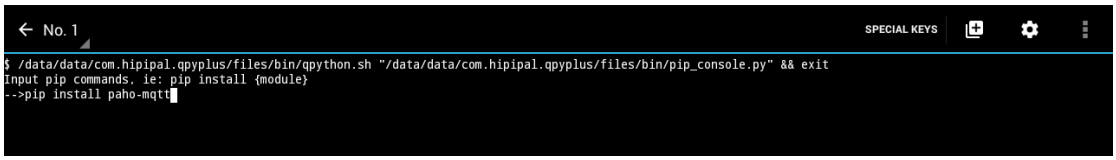


Select the "Pip console"



Input the command in the pip console. For example, the command `pip`

`install paho-mqtt` is used to install the paho-mqtt module.



Note:

Only pure Python package could be installed through the pip console

5.2 Compiling Python Module

As there isn't any C/C++ toolchain on the board, so a Python module including C/C++ code couldn't install through the Python *setuptools*. The user should build the module by Android NDK, and then install it to the system.

The build system is based on the *kivy/python-for-android* project, please refer: https://github.com/kivy/python-for-android

5.2.1 Prerequisites

1. Prepare the minimal environment for building Python

```
sudo apt-get install build-essential patch git-core ccache ant
python-pip python-dev
```

If you are on a 64 bit distro, you should install the following packages too:

```
sudo apt-get install ia32-libs  libc6-dev-i386
```

On Debian Squeeze amd64, those packages were found to be necessary:

```
sudo apt-get install lib32stdc++6 lib32z1
```

Ensure you have the latest Cython version:

```
pip install --upgrade cython
```

2. Download Android NDK, SDK. You can download then at:

NDK: http://dl.google.com/android/ndk/android-ndk-r9d-linux-x86.tar.bz2

SDK: http://dl.google.com/android/android-sdk_r21.0.1-linux.tgz

3. Export some environment variables:

```
export ANDROIDSDK="/path/to/android-sdk-linux_86"

export ANDROIDNDK="/path/to/android/android-ndk-r9d"

export ANDROIDNDKVER=r9d

export ANDROIDAPI=19
```

(Also, you must configure the paths mentioned in ANDROIDSDK and ANDROIDNDK)

4. Clone Python-for-android:

```
git clone git://github.com/kivy/python-for-android

git checkout old_toolchain
```

5. Build a distribution with OpenSSL module, PIL and Kivy:

```
cd python-for-android

./distribute.sh -m "openssl pil kivy"
```

5.2.2 Creating Your Own Recipes

A recipe is a script that contains the "definition" of a module to compile.

If a module is not in the Python-for-android project, this chapter shows that how to create the recipe.

For example, if you want to create a recipe for ***pygpio***:

1. Create a directory name pygpio in the Python-for-android/recipes

```
cd python-for-android/recipes

mkdir pygpio
```

2. Create a *recipe.sh* file from a template

```
cp recipe.sh.tmpl pygpio/recipe.sh

sed -i 's#XXX#pygpio#' pygpio/recipe.sh
```

3. Then, edit the *pygpio/recipe.sh* to adjust other information (version, url) and complete the build function.

```
#!/bin/bash

VERSION_pygpio=${VERSION_pygpio:-1.0}

DEPS_pygpio=(python)

URL_pygpio=http://10.41.150.237/pygpio-$VERSION_pygpio.tgz

MD5_pygpio=

BUILD_pygpio=$BUILD_PATH/pygpio/$(get_directory $URL_pygpio)

RECIPE_pygpio=$RECIPES_PATH/pygpio

function prebuild_pygpio() {

    true

}

function shouldbuild_pygpio() {

    if [ -d "$SITEPACKAGES_PATH/pygpio" ]; then

        DO_BUILD=0

    fi

}

function build_pygpio() {

    cd $BUILD_pygpio/

    push_arm

    try $HOSTPYTHON setup.py build

    try $HOSTPYTHON setup.py install -O2

    try $HOSTPYTHON setup.py clean

    pop_arm

}

# function called after all the compile have been done

function postbuild_pygpio() {

    true

}
```

4. Create the *pygpio* module

```
./distribute.sh -m "pygpio"
```

5. Push the *pygpio* module files to system:

Change the directory to "dist/default/python-install/lib/python2.7". The compiled files can be found in this directory or in the "site-packages" sub-directory.

For example: the "pygpio" module is created in a sub-directory "site-packages/RPi"

```
rodger.lin@shsa02:~/work/rpi2/kivy/python-for-android/dist/default/python-install/lib/python2.7$
rodger.lin@shsa02:~/work/rpi2/kivy/python-for-android/dist/default/python-install/lib/python2.7$ pwd
/shsa022/usrhome/rodger.lin/work/rpi2/kivy/python-for-android/dist/default/python-install/lib/python2.7
rodger.lin@shsa02:~/work/rpi2/kivy/python-for-android/dist/default/python-install/lib/python2.7$
rodger.lin@shsa02:~/work/rpi2/kivy/python-for-android/dist/default/python-install/lib/python2.7$ ls site-packages/RPi
GPIO.so  __init__.py  __init__.pyc  __init__.pyo
rodger.lin@shsa02:~/work/rpi2/kivy/python-for-android/dist/default/python-install/lib/python2.7$
```

Because the "adb" command could not push multiple files, so all the files should be compressed into a single file and the file package should be extracted once it pushed to the platform.

```
tar zcvf RPi.tar.gz RPi/

adb remount

adb push RPi.tar.gz/system/lib/python2.7/site-packages/

adb shell

cd /system/lib/python2.7/site-packages/

tar zxvf RPi.tar.gz
```

Revision History

Document Change History

Revision	Changes	Date
v1.0a	Two new chapters have been added to the manual. The latest chapters are, "Chapter 4: QPython" and "Chapter 5: "Installing Pure-Python Package Index"	October 28, 2015
v1.0	Initial Release	September 07, 2015

Table 3. Document Change History

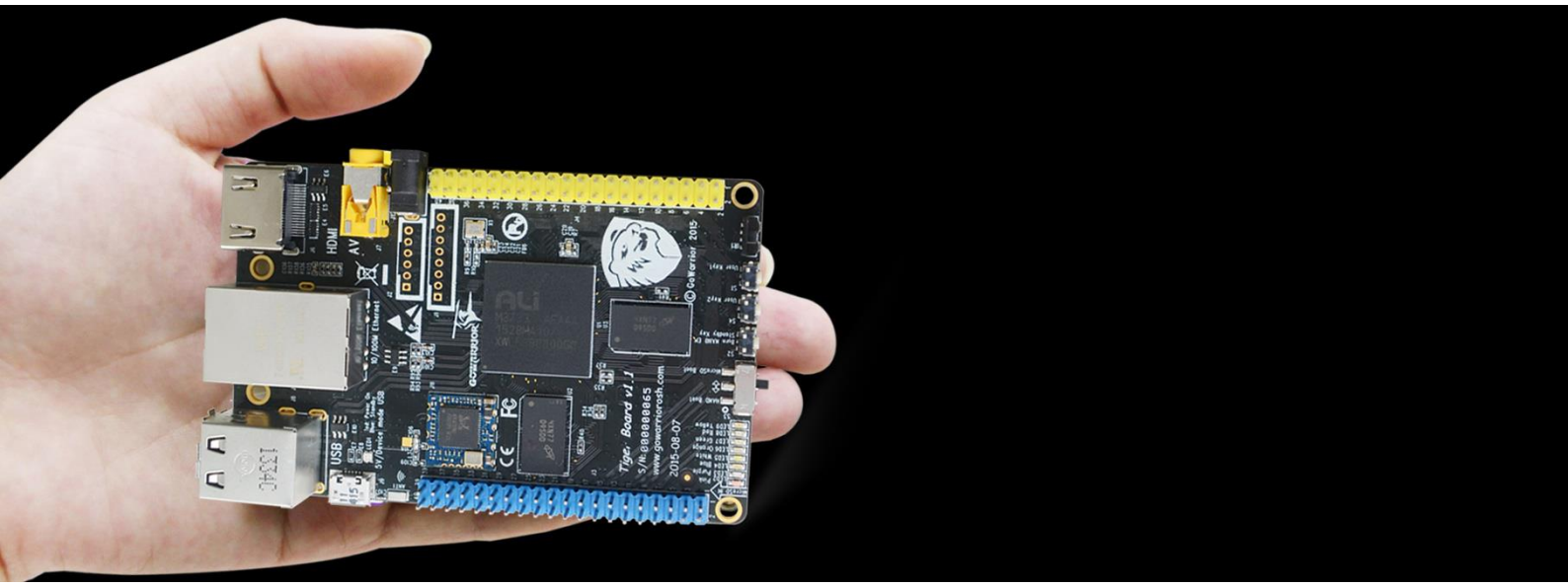
Software Changes

Revision	Changes	Date
v1.0	Initial Release	September 07, 2015

Table 4. Software Change History

www.gowarriorosh.com

Headquarters

Tel: +886-2-8752-2000

Fax: +886-2-8751-1001