



# 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](http://www.gowarriorosh.com)

**Copyright © 2015 GoWarrior Community All Rights Reserved.**

# Table of Contents

<b>Preface .....</b>	<b>1</b>
Overview .....	1
Audience .....	1
Applicable Products.....	1
Reference Documents.....	1
Conventions.....	2
How to Contact Us.....	3
 <b>1   Introduction .....</b>	 <b>4</b>
 <b>2   Connecting LED.....</b>	 <b>5</b>
2.1   Solution.....	5
2.2   Discussion .....	6
 <b>3   Controlling the Brightness of LED.....</b>	 <b>7</b>
3.1   Solution.....	7
3.2   Discussion .....	8
 <b>Revision History .....</b>	 <b>10</b>
Document Change History.....	10
Software Changes.....	10

## List of Tables

Table 1. Typographical Conventions.....	2
Table 2. Symbol Conventions .....	2
Table 3. Document Change History .....	10
Table 4. Software Change History.....	10

## List of Figures

Figure 1. Connecting an LED to a GoWarrior 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.

## 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	<code>Courier New, Size 10.5</code>
Variables, code variables, and code comments	<i>Courier New, Size, Italic</i>
Menu item, buttons, tool names	Ebrima, Size 10.5, Bold e.g. Select USB Debugging
Screens, windows, dialog boxes, and tabs	<b>Ebrima, Size 10.5, Bold</b> <b>Enclosed in double quotation marks</b> e.g. Open the “Debug Configuration” dialog box

**Table 1. Typographical Conventions**

### Symbol Conventions

Item	Description
 <i>Caution</i>	Indicates a potential hazard or unsafe practice that, if not avoided, could result in data loss, device performance degradation, or other unpredictable results.
 <i>Note</i>	Indicates additional and supplemental information for the main contents.
 <i>Tip</i>	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](mailto: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](mailto: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

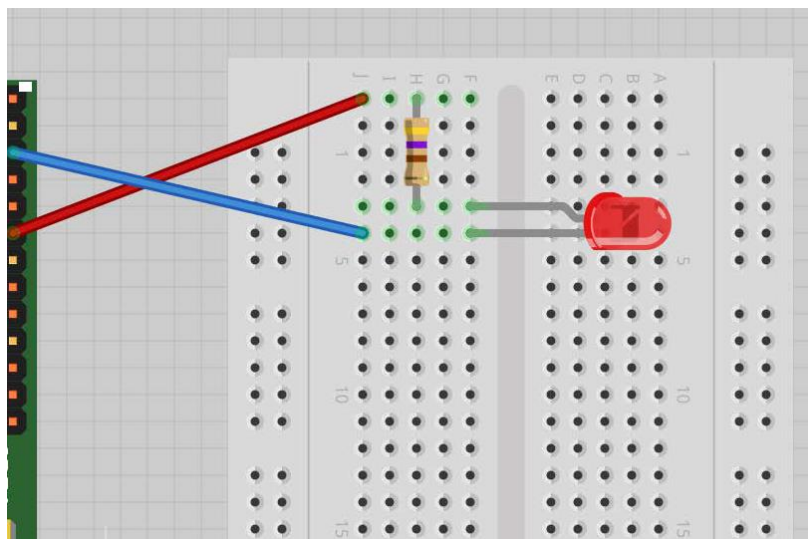
This chapter describes how to connect LED and lighten it up.

### 2.1 Solution

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

- Breadboard and male to male jumper wires
- $470\ \Omega$  or  $1k\ \Omega$  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 source 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

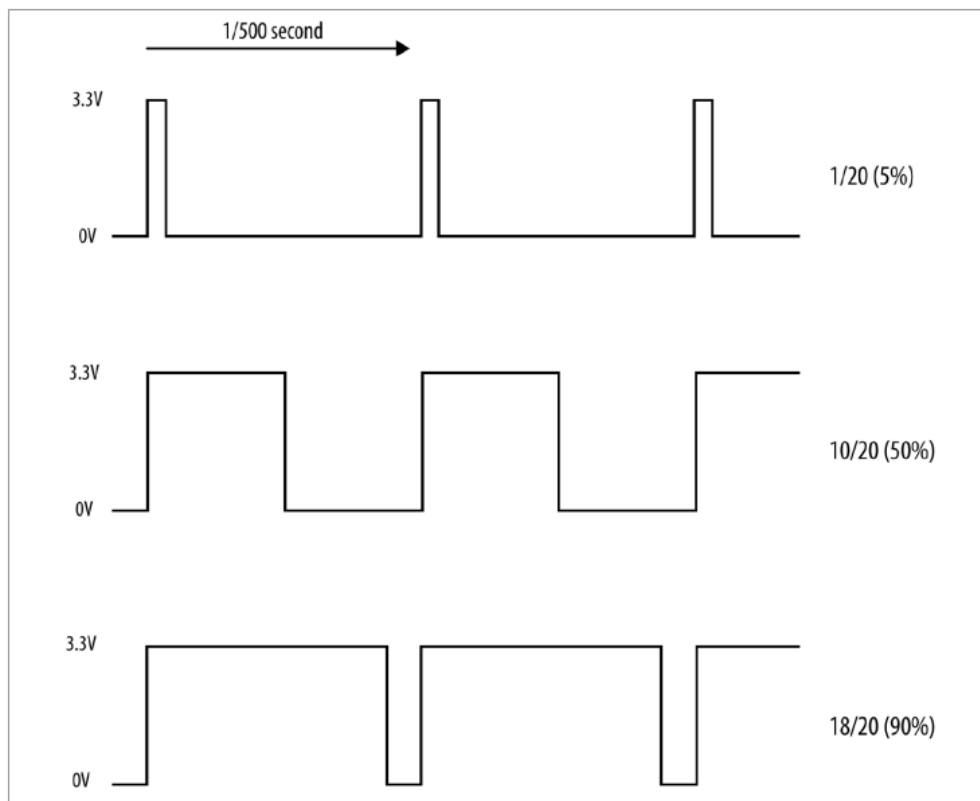
Enter Brightness (0 to 100):20
```

```
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.

# Revision History

## Document Change History

Revision	Changes	Date
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](http://www.gowarriorosh.com)

Headquarters

Tel: +886-2-8752-2000

Fax: +886-2-8751-1001

