



Basic Usage of New GPIO Test Program

Version 1.2 – 16 January 2016
Kit Bishop

Document History

Version	Date	Details
Version 1.2	16 January 2016	New document describing usage of new-gpiotest program with GPIO irq capabilities

Contents

1. <i>Background</i>	1
2. <i>Pre-requisites</i>	2
3. <i>Files Supplied</i>	2
4. <i>Basic Installation</i>	2
4.1. Installing libnew-gpio.so	2
4.2. Installing the new-gpiotest Program	2
5. <i>Usage of the new-gpiotest Program</i>	2

1. Background

new-gpiotest is alternative C++ program used to access the Omega GPIO pins.

While it has some similarities to **fast-gpio** it has extended capabilities including usage of interrupts on the GPIO pins.

It consists of two main components:

- **libnew-gpio.so** – a dynamic library containing the classes used to interact with GPIO pins
- **new-gpiotest** – a simple test program for interacting with GPIO pins using **libnew-gpio.so**

Full details of the library, the program and all source code can be found at [Alternative C++ Code for GPIO Access](https://community.onion.io/topic/143/alternative-c-code-for-gpio-access) (<https://community.onion.io/topic/143/alternative-c-code-for-gpio-access>)

This document primarily outlines the basic usage of the **new-gpiotest** program.

2. Pre-requisites

To use the **new-gpiotest** program, your Omega **must** fulfil the following pre-requisites:

- Must have been upgraded to version 0.0.6-b265 or later
- Must have the `kmod-gpio-irq` package installed by running:

```
opkg update
opkg install kmod-gpio-irq
```

3. Files Supplied

There are two files required for **new-gpiotest** :

- **libnew-gpio.so** – the required dynamic library
- **new-gpiotest** – the program itself

4. Basic Installation

Installing the software is simple. It primarily consists of copying the library and test program to suitable locations on your Omega.

4.1. Installing libnew-gpio.so

Copy the **libnew-gpio.so** file to the `/lib` directory on your Omega.

Alternatively, you can copy the library to any location that may be set up in any `LD_LIBRARY_PATH` directory on your Omega. For example, I use the following for testing:

- Created directory `/root/lib`
- Copied the library to `/root/lib`
- Added the following lines to my `/etc/profile` file:

```
LD_LIBRARY_PATH=/root/lib:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH
```

4.2. Installing the new-gpiotest Program

Copy the **new-gpiotest** program file to any suitable directory on your Omega from which you wish to run it.

5. Usage of the new-gpiotest Program

The **new-gpiotest** program accepts a set of parameters to control its operation.

The program will document it's usage when the command **new-gpiotest help** is used.

In addition, the usage is shown whenever any errors are detected in the parameters.

The usage information displayed is:

```
./new-gpiotest
Usage
Commands - one of:
- ./new-gpiotest <op> <pin> <val>
- ./new-gpiotest pwm <pin> <freq> <duty>
  Starts PWM output on pin
- ./new-gpiotest pwmstop <pin>
  Stops PWM output on pin
- ./new-gpiotest irq <pin> <irqtype> <irqcmd> <debounce>
  Enables IRQ handling on pin
- ./new-gpiotest irqstop <pin>
  Terminates IRQ handling on pin

Where:
<op> is one of:
  info - to display info on pin(s)
  set - to set pin(s) value
  get - to get and return pin value
  setd - to set pin(s) direction
  getd - to get and return pin direction
  help - to display usage
<pin> is one of
  0, 1, 6, 7, 8, 12, 13, 14, 15, 16, 17, 18, 19, 23, 26, all
  A <pin> of all can only be used for an <op> of:
    info, set, setd
<val> is only required for set and setd:
  for set, <val> is 0 or 1
  for setd, <val> is in or out
<freq> is PWM frequency in Hz > 0
<duty> is PWM duty cycle % in range 0 to 100
<irqtype> is the type for IRQ and is one of:
  falling, rising, both
<irqcmd> is the shell command to be executed when the IRQ occurs
  Must be enclosed in " characters if it contains
  spaces or other special characters
  If it starts with the string [debug],
  debug output is displayed first
<debounce> is optional debounce time for IRQ in milliseconds
  Defaults to 0 if not supplied
```

Notes:

1. The return value from the command will be one of the following:
 - **255 (-1)** – indicates an error has occurred – either in the parameters or in executing the command
 - **0** – indicates normal successfully completion for an operation (<op>) other than **get** or **getd**
 - For a successful **get** operation:
 - **0** – indicates the pin is **off**
 - **1** – indicates the pin is **on**
 - For a successful **getd** operation:
 - **0** – indicates the pin is an **input** pin
 - **1** – indicates the pin is an **output** pin
2. When the **pwm** operation is used, the program forks a separate process to perform the PWM output.

This separate process continues after the program returns until such time as the **pwmstop** operation is performed on the same pin.

The ID of the separate process can be discovered by running:

new-gpiotest info <pin-number>

3. When the **irq** operation is used, the program forks a separate process to monitor and respond to pin state changes.

Each time the relevant pin undergoes the relevant change in state, the **<irqcmd>** command specified is run.

This separate process continues after the program returns until such time as the **irqstop** operation is performed on the same pin.

The ID of the separate process can be discovered by running:

new-gpiotest info <pin-number>