Hardware Library Configuration

Jetson.GPIO - Linux for Tegra

1. Introduction

The Jetson Orin NX, Jetson Orin Nano, and Jetson Nano development boards contain a 40-pin GPIO header similar to the 40-pin header in the Raspberry Pi. The digital inputs and outputs of these GPIOs can be controlled using the Python library provided in the Jetson GPIO Library package. This library has the same API as the Raspberry Pi's RPi.GPIO library in order to provide an easy way to move applications running on the Raspberry Pi to the Jetson board.

In addition to this document, the Jetson GPIO Library package also contains the following:

1. The lib/python/ subdirectory contains the Python modules that implement all of the library functionality. The gpio.py module is the main component that will be imported into the application and provides the required APIs. The gpio_event.py and gpio_pin_data.py modules are used by the gpio.py module and cannot be imported directly into the application.
2. The samples/ subdirectory contains sample applications to help you get familiar with the library API and get started using applications. The simple_input.py and simple_output.py applications show how to perform read and write operations on GPIO pins, respectively, while button_led.py, button_event.py, and button_interrupt.py show how to use a button press to blink an LED using busy wait, blocking wait, and interrupt callbacks, respectively.

This document will describe what is included in the Jetson GPIO library package, how to configure your system, and run the provided sample applications and library API. Here we briefly introduce how to use this library about Jetson.GPIO, here are detailed instructions:
https://pypi.org/project/Jetson.GPIO/
or
https://github.com/NVIDIA/jetson-gpio

2. Pin diagram

GPIO and BCM comparison table

BCM编码	功能名	物理引脚		功能名	BCM编码
	3V3	1	2	5V	
2	SDA	3	4	5V	
3	SCL	5	6	GND	
4	D4	7	8	D14(TXD)	14
	GND	9	10	D15(RXD)	15
17	D17	11	12	D18	18
27	D27	13	14	GND	
22	D22	15	16	D23	23
	3V3	17	18	D24	24
10	D10	19	20	GND	
9	D9	21	22	D25	25
11	D11	23	24	D8	8
	GND	25	26	D7	7
0	DO(ID_SD)	27	28	D1(ID_SC)	1
5	D5	29	30	GND	
6	D6	31	32	D12	12
13	D13	33	34	GND	
19	D19	35	36	D16	16
26	D26	37	38	D20	20
	GND	39	40	D21	21

3. Environment configuration

Environment configuration

1. Download jetson-gpio:
   git clone https://github.com/NVIDIA/jetson-gpio

```
jetson@jetson-desktop:~$ git clone https://github.com/NVIDIA/jetson-gpio
Cloning into 'jetson-gpio'...
remote: Enumerating objects: 168, done.
remote: Counting objects: 100% (168/168), done.
remote: Compressing objects: 100% (97/97), done.
remote: Total 597 (delta 79), reused 135 (delta 48), pack-reused 429
Receiving objects: 100% (597/597), 128.43 KiB | 38.00 KiB/s, done.
Resolving deltas: 100% (267/267), done.
```

2. Move the downloaded file to the directory: /opt/nvidia
   If this library exists in your directory, we need to back up the original directory as follows:

```
nano@nano-desktop:/opt/nvidia$ sudo mv jetson-gpio jetson-gpio_bak
[sudo] password for nano:
nano@nano-desktop:/opt/nvidia$ ls
jetson-gpio_bak  l4t-usb-device-mode
nano@nano-desktop:/opt/nvidia$
```

Then put the downloaded file in the opt/nvidia/ directory. Because the author puts the folder in the path ~/ and is currently in opt/nvidia/, you can execute the following command to move the folder
sudo mv ~/jetson-gpio ./

```
jetson@jetson-desktop:/opt/nvidia$ sudo mv ~/jetson-gpio ./
[sudo] password for jetson:
jetson@jetson-desktop:/opt/nvidia$ ls
jetson-gpio  jetson-io  l4t-bootloader-config  l4t-usb-device-mode  vpi  vpi-0.4
jetson@jetson-desktop:/opt/nvidia$ cd ~
```

3. Install the pip3 tool:
   sudo apt-get install python3-pip
4. Enter the jetson-gpio library folder and install the library.
   cd /opt/nvidia/jetson-gpio

sudo python3 setup.py install

```
creating 'dist/Jetson.GPIO-2.0.12-py3.6.egg' and adding 'build/bdist.linux-aarc
removing 'build/bdist.linux-aarch64/egg' (and everything under it)
Processing Jetson.GPIO-2.0.12-py3.6.egg
Copying Jetson.GPIO-2.0.12-py3.6.egg to /usr/local/lib/python3.6/dist-packages
Adding Jetson.GPIO 2.0.12 to easy-install.pth file

Installed /usr/local/lib/python3.6/dist-packages/Jetson.GPIO-2.0.12-py3.6.egg
Processing dependencies for Jetson.GPIO==2.0.12
Finished processing dependencies for Jetson.GPIO==2.0.12
```

5. Before use, you also need to create a gpio group, add your current account to this group, and grant usage permissions
   sudo groupadd -f -r gpio
   sudo usermod -a -G gpio user_name

```
nano@nano-desktop:/opt/nvidia/jetson-gpio$ sudo groupadd -f -r gpio
nano@nano-desktop:/opt/nvidia/jetson-gpio$ sudo usermod -a -G gpio nano
```

sudo cp /opt/nvidia/jetson-gpio/lib/python/Jetson/GPIO/99-gpio.rules /etc/udev/rules.d/
In order for the new rules to take effect, you need to reboot or reload the udev rules by running the following command
sudo udevadm control --reload-rules && sudo udevadm trigger
Note: user_name is your username, for example jetson

Pay special attention to the official instructions:

2521704	On Jetson Nano, when a GPIO is released (e.g. using the `sysfs` unexport file, the `cleanup()` function in the `Jetson.GPIO` Python module, or the `gpio_free()` function in the Linux kernel), the kernel configures the pin as a special function (SFIO) rather than as a GPIO input. In some cases, this causes Jetson to drive a signal onto the pin. If another device is also connected to that pin, and is also driving a signal, this causes an electrical conflict, which may damage the hardware. This issue is particularly relevant for the pins on the 40-pin GPIO expansion header.

Use GPIO.cleanup() as little as possible