Jetson Nano hardware control basic tutorial

The Jetson TX1, TX2, AGX Xavier and Nano development boards include a 40-pin GPIO, which similar to the 40-pin GPIO of the Raspberry Pi. The Python libraries provided in the Jetson GPIO Library package can be used to control the digital inputs and outputs of these GPIOs. This library has the same API as the RPi.GPIO library of Raspberry Pi.
So users can move applications running on the Raspberry Pi to the Jetson development board.

Packaging component

In addition to this document, the Jetson GPIO library package contains the following:
The **lib/python/** subdirectory contains Python modules that implement all library function. The gpio.py module is the main component that will import the application and provide the required API.
The **samples/** subdirectory contains sample applications to help you to become familiar with the API library and start with the application. These **simple_input.py** and **simple_output.py** applications show that how to perform read and write operations on GPIO pins. **Button_led.py, button_event.py** and **button_interrupt.py** show that how to use the button control the use of busy-waiting, blocking-waiting and interrupting callback to make the LED flash.

This document describes what is included in the Jetson GPIO library, how to configure the system, and how to run the sample application and API library already provided.

For detailed description of the Jetson.GPIO library:
https://pypi.org/project/Jetson.GPIO/
Or
https://github.com/NVIDIA/jetson-gpio

40 pins comparison table

BCM	Function	Physical pin		Function	BCM
	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

Environmental configuration

1.Download jetson-gpio:

We can input the following command:

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

```
nano@nano-desktop:~$ git clone https://github.com/NVIDIA/jetson-gpio
Cloning into 'jetson-gpio'...
remote: Enumerating objects: 47, done.
remote: Counting objects: 100% (47/47), done.
remote: Compressing objects: 100% (26/26), done.
remote: Total 47 (delta 20), reused 40 (delta 13), pack-reused 0
Unpacking objects: 100% (47/47), done.
nano@nano-desktop:~$
```

2.Moving the downloaded file to the specified directory: **/opt/nvidia**

Note: If your directory already has this library, then you need to back up the original directory, using the following command:

```
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.

3.Install pip3 tool:

sudo apt-get install python3-pip

4.Enter jetson-gpio library folder and install library

cd /opt/nvidia/jetson-gpio

sudo python3 setup.py install

5. Before using, you also need to create a gpio group, add your current account to this group, and give use permission.

using the following command:
sudo groupadd -f -r gpio
sudo usermod -a -G gpio user_name
sudo cp /opt/nvidia/jetson-gpio/etc/99-gpio.rules /etc/udev/rules.d/
sudo udevadm control --reload-rules && sudo udevadm trigger

(!Note: user_name is your user name,for example:nano)

Routine test:
Once the environment is configured, you can test the routines. There are a few simple routines available in the jetson-gpiofolder.

You can input the following command to enter program directory:
cd ~/opt/nvidia/jetson-gpio/samples/

1.simple_input.py
This is a simple input program that uses the BCM pin coding mode to read the value of PIN12 and print it to the terminal.

You can input the following command to run program:
sudo python3 simple_input.py

Experimental phenomena：
After running the program, we can see the print information of the terminal. By default, the value of Pin18 is low. Find a DuPont line and connect the 12th pin to 3.3V. You can see that the read value becomes HIGH. If connected to GND, LOW will be displayed.
As shown below:

```
nano@nano-desktop:/opt/nvidia/jetson-gpio/samples$ sudo python3 simple_input.py
Starting demo now! Press CTRL+C to exit
Value read from pin 18 : LOW

Value read from pin 18 : HIGH


Value read from pin 18 : LOW
Value read from pin 18 : HIGH
```

【Note】

- Here Pin18 is the BCM code, and the above PIN12 refers to the physical code.

- The working level of the pins of Jetson nano is 3.3V, so do not to connect 5V level when using.

2.simple_out.py

The program will output high and low levels (alternating every 2 seconds) to the physical pin PIN12.

You can input the following command to run program:

sudo python3 simple_out.py

Experimental phenomena：

We need to connect LED to PIN12, after running the program, we can see that LED flash.

```
nano@nano-desktop:/opt/nvidia/jetson-gpio/samples$ sudo python3 simple_out.py
/usr/local/lib/python3.6/dist-packages/Jetson.GPIO-0.1.2-py3.6.egg/Jetson/GPIO/gpio.py:202: Runt
imeWarning: This channel is already in use, continuing anyway. Use GPIO.setwarnings(False) to di
sable warnings
Starting demo now! Press CTRL+C to exit
Outputting 1 to pin 18
Outputting 0 to pin 18
Outputting 1 to pin 18
Outputting 0 to pin 18
Outputting 1 to pin 18
Outputting 0 to pin 18
```

3.button_led.py

The program uses polling to control the LEDs by pressing a button.

Hardware Wiring:

The button is required to connect PIN18 and GND, a pull-up resistor is required to connect PIN18 and 3.3V, and connect an LED to the PIN12.

You can input the following command to run program:

sudo python3 button_led.py

Experimental phenomena：

After connecting the hardware, running the program. Due to the pull-up resistor, PIN18 defaults to high level and the LED goes out. When the button is pressed, PIN18 become low, it is judged that the button is pressed, the LED is go on.