??GPIO??

GPIO General-purpose input/output

GPI GPO GPIO

Jetson Orin Nano□□□　40PIN GPIO□□□□□□

Sysfs GPIO	Name	Pin	Pin	Name	Sysfs GPI
	3.3 VDC *Power*	1	2	**5.0 VDC** *Power*	
	I2C1_SDA *I2C Bus 7*	3	4	**5.0 VDC** *Power*	
	I2C1_SCL *I2C Bus 7*	5	6	**GND**	
144 gpio492	**GPIO9** *AUDIO_MCLK*	7	8	**UART1_TX** */dev/ttyTHS0*	
	GND	9	10	**UART1_RX** */dev/ttyTHS0*	
112 gpio460	**UART1_RTS**	11	12	**I2S0_SCLK**	*50* gpio398
122 gpio470	**SPI1_SCK**	13	14	**GND**	
85 gpio433	**GPIO12** *Alt: PWM*	15	16	**SPI1_CS1**	*126* gpio474
	3.3 VDC *Power*	17	18	**SPI1_CS0**	*125* gpio473
135 gpio483	**SPI0_MOSI**	19	20	**GND**	
134 gpio482	**SPI0_MISO**	21	22	**SPI1_MISO**	*123* gpio471
133 gpio481	**SPI0_SCK**	23	24	**SPI0_CS0**	*136* gpio484
	GND	25	26	**SPI0_CS1**	*137* gpio485

1 ?JetPack 5?????GPIO

1.1 ???????GPIO

??GPIO?????????

以 31号引脚为例， 454为 GPIO口编号， PQ.06为引脚名称

- 提升至 root权限，启用 GPIO□

```
sudo bash


echo 454 > /sys/class/gpio/export
```

- 设置 GPIO为输入模式

```
echo in > /sys/class/gpio/PQ.06/direction
```

- 获取 GPIO当前状态，返回值 1为高电平， 0为低电平

```
cat /sys/class/gpio/PQ.06/value
```

- 设置 GPIO为输出模式

```
echo out > /sys/class/gpio/PQ.06/direction
```

- 设置 GPIO输出， 1为高电平， 0为低电平

```
echo 1 > /sys/class/gpio/PQ.06/value
```

您也可以使用以下脚本进行测试

```
#!/bin/bash
trap 'echo PQ.06 > /sys/class/gpio/unexport; echo "GPIO PQ.06 is released"' EXIT

echo "setting GPIO PQ.06"
echo PQ.06 > /sys/class/gpio/export 2>/dev/null

# set Pin output mode
echo out > /sys/class/gpio/PQ.06/direction
```

```
# blink
while true
do
    echo 0 > /sys/class/gpio/PQ.06/value
    sleep 0.5
    cat /sys/class/gpio/PQ.06/value
    sleep 0.5
    echo 1 > /sys/class/gpio/PQ.06/value
    sleep 0.5
    cat /sys/class/gpio/PQ.06/value
    sleep 0.5
done
```

1.2 ??python??GPIO

1.2.1 ??JETSON.GPIO?

- 如果您不确定您的环境是否正常，或者您尝试使用 **conda**环境运行脚本请执行以下指令安装 Jetson 官方的 GPIO库

```
pip install JETSON.GPIO
```

Jetson GPIO库提供了四种给 I / O引脚编号的方法。

- **BOARD**: 物理引脚编号（ 40针接口顺序）。
- **BCM**: Broadcom SoC 的 GPIO 编号（ 常用）。
- **CVM**: CVM/CVB 连接器的信号名称。
- **TEGRA_SOC**: Tegra SoC 信号名称。

```
import time
import RPi.GPIO as GPIO

# define pin number
output_pin = 31

# set pin as BOARD mode
GPIO.setmode(GPIO.BOARD)

# set pin mode
GPIO.setup(output_pin, GPIO.OUT)

print("Press CTRL+C to exit")
```

```
curr_value = GPIO.HIGH
try:
    while True:
        time.sleep(1)
        print("pin {} now is {}".format(output_pin, curr_value))
        GPIO.output(output_pin, curr_value)
        # blink
        curr_value ^= GPIO.HIGH
finally:
    GPIO.cleanup()
```

其他具体用法和例程可参考 项目官网_____

1.3??C/C++??GPIO

- 安装 libgpio-dev

```
sudo apt install libgpio-dev
```

参考例程:

```
/**
 * License - MIT.
*/
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <gpiod.h>

#define GPIO_CHIP       "/dev/gpiochip0"
#define GPIO_LED        3

int main()
{
    int ret = 0;
    struct gpiod_chip *gpiochip;
    struct gpiod_line *gpioline;

    // Open driver.
    gpiochip = gpiod_chip_open(GPIO_CHIP);
```

```
    if (NULL == gpiochip) {
        printf("Error in gpiod_chip_open.\n");
        ret = -1;
        goto out1;
    }

    // Get gpio.
    gpioline = gpiod_chip_get_line(gpiochip, GPIO_LED);

    if (NULL == gpioline) {
        printf("Error in gpiod_chip_get_line.\n");
        ret = -1;
        goto out2;
    }

    // Set gpio direction.
    ret = gpiod_line_request_output(gpioline, "gpio", 0);

    if (ret != 0) {
        printf("Error in gpiod_line_request_output.\n");
        ret = -1;
        goto out2;
    }

    // Blink.
    for (int i = 0; i < 10; i++) {
        printf("%d times.\n", i);

        gpiod_line_set_value(gpioline, 1);
        sleep(1);

        gpiod_line_set_value(gpioline, 0);
        sleep(1);
    }

    // Release.
    gpiod_line_release(gpioline);

out2:
    gpiod_chip_close(gpiochip);
```

```
out1:
    return ret;
}
```

- 编译源文件

```
g++ test_gpio.cc -o test_gpio -lgpiod
```

- 执行测试程序

```
./test_gpio
```

2 ?JetPack 6?????GPIO

2.1??libgpiod??GPIO

在 **JetPack 6.2/r36.4.3** 中， NVIDIA已经移除了传统的 sysfs GPIO 接口（ /sys/class/gpio）不建议再使用 sysfs 来控制 GPIO。推荐使用 Linux统一的 **libgpiod** 管理 GPIO

- 安装前置软件包

```
sudo apt-get install busybox automake autoconf libtool
```

- 验证安装是否成功

```
jetson@jetson-desktop:~$ gpioinfo
gpiochip0 - 164 lines:
□line   0:      "PA.00" "regulator-vdd-3v3-sd" output active-high [used]
□line   1:      "PA.01"       unused   input  active-high
□line   2:      "PA.02"       unused   input  active-high
□line   3:      "PA.03"       unused   input  active-high
□line   4:      "PA.04"       unused   input  active-high
□line   5:      "PA.05"       unused   input  active-high
□line   6:      "PA.06"       unused   input  active-high
□line   7:      "PA.07"       unused   input  active-high
□line   8:      "PB.00"       unused   input  active-high
□line   9:      "PC.00"       unused   input  active-high
...........
```

- 读取 GPIO 状态

```
gpioget $(gpiofind "PQ.06")
```

- 将 GPIO 设置为输出模式

```
sudo busybox devmem 0x2430070 w 0x004
```

各个引脚对应的寄存器地址从官方文档中可以查询到

- 保持高电平：

```
gpioset --mode=wait `gpiofind "PQ.06"`=1
```

- 保持低电平：

```
gpioset --mode=wait `gpiofind "PQ.06"`=0
```

2.2 ??python??GPIO

> 在 JetPack6 系统中仍需要修改寄存器内容来更改引脚 GPIO模式以使用高级语言控制 GPIO

- 将 PQ.06 引脚设置为输出模式

```
sudo busybox devmem 0x2430070 w 0x004
```

- 如果您尝试使用 conda 环境运行脚本请执行以下指令安装 Jetson 官方的 GPIO 库

```
pip install JETSON.GPIO
```

- 建议执行以下指令，重新安装系统自带的 JETSON.GPIO 库

```
sudo rm -rf /usr/lib/python3*/dist-packages/Jetson
sudo rm -rf /usr/local/lib/python3*/dist-packages/Jetson
git clone https://github.com/NVIDIA/jetson-gpio.git
cd jetson-gpio
sudo pip3 install .
```

Jetson GPIO 库提供了四种给 I / O引脚编号的方法。

- **BOARD**：物理引脚编号（40针接口顺序）。
- **BCM**：Broadcom SoC 的 GPIO 编号（常用）。

- **CVM**: CVM/CVB 连接器的信号名称。
- **TEGRA_SOC**: Tegra SoC 信号名称。

```
import time
import RPi.GPIO as GPIO

# define pin number PQ.06
output_pin = 31

# set pin as BOARD mode
GPIO.setmode(GPIO.BOARD)

# set pin mode
GPIO.setup(output_pin, GPIO.OUT)

print("Press CTRL+C to exit")
curr_value = GPIO.HIGH
try:
    while True:
        time.sleep(1)
        print("pin {} now is {}".format(output_pin, curr_value))
        GPIO.output(output_pin, curr_value)
        # blink
        curr_value ^= GPIO.HIGH
finally:
    GPIO.cleanup()
```

其他具体用法和例程可参考 项目官网_____

2.3??C/C++??GPIO

> 在 JetPack6 系统中仍需要修改寄存器内容来更改引脚 GPIO模式以使用高级语言控制 GPIO

- 将 PQ.06 引脚设置为输出模式

```
sudo busybox devmem 0x2430070 w 0x004
```

- 安装 libgpio-dev

```
sudo apt install libgpiod-dev
```

□□□□□

```
/**
 * License - MIT.
*/
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <gpiod.h>

#define GPIO_CHIP       "/dev/gpiochip0"
#define GPIO_LED        106

int main()
{
    int ret = 0;
    struct gpiod_chip *gpiochip;
    struct gpiod_line *gpioline;

    // Open driver.
    gpiochip = gpiod_chip_open(GPIO_CHIP);

    if (NULL == gpiochip) {
        printf("Error in gpiod_chip_open.\n");
        ret = -1;
        goto out1;
    }

    // Get gpio.
    gpioline = gpiod_chip_get_line(gpiochip, GPIO_LED);

    if (NULL == gpioline) {
        printf("Error in gpiod_chip_get_line.\n");
        ret = -1;
        goto out2;
    }

    // Set gpio direction.
    ret = gpiod_line_request_output(gpioline, "gpio", 0);
```

```
    if (ret != 0) {
        printf("Error in gpiod_line_request_output.\n");
        ret = -1;
        goto out2;
    }

    // Blink PQ.06
    for (int i = 0; i < 30; i++) {
        printf("%d times.\n", i);

        gpiod_line_set_value(gpioline, 1);
        sleep(1);

        gpiod_line_set_value(gpioline, 0);
        sleep(1);
    }

    // Release.
    gpiod_line_release(gpioline);

out2:
    gpiod_chip_close(gpiochip);

out1:
    return ret;
}
```

- 编译源文件

```
g++ test_gpio.cc -o test_gpio -lgpiod
```

- 执行测试程序

```
./test_gpio
```

PADCTL????GPIO????

> 我们从官方文档中查找摘取了以下信息供您参考，您也可以自行查阅官方文档。
>
> Pinmux和 TRM

PADCTL_A0(PADCTL_G3) 0x02430000
PADCTL_A4(PADCTL_G4) 0x02434000
PADCTL_A16(PADCTL_EDP) 0x02440000
PADCTL_A24(PADCTL_G7) 0x02448000

7 GPIO09 PADCTL_G7_SOC_GPIO59_0 0x30 gpio-492 PAC.06 0x02448030
15 GPIO12 PADCTL_EDP_SOC_GPIO39_0 0x20 gpio-433 PN.01 0x02440020
29 GPIO01 PADCTL_G3_SOC_GPIO32_0 0x68 gpio-453 PQ.05 0x02430068
31 GPIO11 PADCTL_G3_SOC_GPIO33_0 0x70 gpio-454 PQ.06 0x02430070
32 GPIO07 PADCTL_G4_SOC_GPIO19_0 0x80 gpio-389 PG.06 0x02434080
33 GPIO13 PADCTL_G4_SOC_GPIO21_0 0x40 gpio-391 PH.00 0x02434040

□□□ #1
 edit □□□ 15 □□ 2025 09:31:53
 edit □□□ 15 □□ 2025 09:31:53