

??GPIO??

GPIO □ General-purpose input/output



□ GPI □ □ □ □ □ □ □ □

GPO □ □ □ □ □ □ □ □ □ □

GPIO □ □

Jetson Orin Nano 40PIN GPIO

Sysfs GPIO	Name	Pin	Pin	Name	Sysfs GPIO
	3.3 VDC <i>Power</i>	1	2	5.0 VDC <i>Power</i>	
	I2C1_SDA <i>I2C Bus 7</i>	3	4	5.0 VDC <i>Power</i>	
	I2C1_SCL <i>I2C Bus 7</i>	5	6	GND	
144 gpio492	GPIO9 <i>AUDIO_MCLK</i>	7	8	UART1_TX <i>/dev/ttyTHS0</i>	
	GND	9	10	UART1_RX <i>/dev/ttyTHS0</i>	
112 gpio460	UART1_RTS	11	12	I2S0_SCLK	50 gpio398
122 gpio470	SPI1_SCK	13	14	GND	
85 gpio433	GPIO12 <i>Alt: PWM</i>	15	16	SPI1_CS1	126 gpio474
	3.3 VDC <i>Power</i>	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??????????

```
❏ 31❏❏❏❏❏ 454❏ GPIO❏❏❏❏ PQ.06❏❏❏❏
```

- ❏❏ root❏❏❏❏ GPIO❏

```
sudo bash
```

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

- ❏❏ GPIO❏❏❏❏❏

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

- ❏❏ GPIO❏❏❏❏❏❏❏❏❏ 1❏❏❏❏❏ 0❏❏❏❏

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

- ❏❏ GPIO❏❏❏❏❏

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

- ❏❏ GPIO❏❏❏❏ 1❏❏❏❏❏ 0❏❏❏❏

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

```
❏❏❏❏❏❏❏❏❏❏❏❏❏❏❏❏
```

```
#!/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?

- `pip install JETSON.GPIO` `conda install JETSON.GPIO` Jetson

```
pip install JETSON.GPIO
```

Jetson GPIO `BOARD` I/O `BCM`

- **BOARD** `BOARD` 40 `BCM`
- **BCM** Broadcom SoC `BCM` GPIO `BCM` `BCM`
- **CVM** CVM/CVB `CVM`
- **TEGRA_SOC** Tegra SoC `TEGRA_SOC`

```

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()

```



1.3??C/C++??GPIO

- `libgpio-dev`

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



```

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

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

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

    // Open driver.
    gpiochip = gpiochip_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);
```


- `GPIO`

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

- `GPIO`

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

```
XXXXXXXXXXXX          XXXX_XXXX
```

- `XXXXXX`

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

- `XXXXXX`

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

2.2 python GPIO

JetPack6 `XXXXXXXXXXXXXXXXXXXX`

GPIO `XXXXXXXXXXXX`

GPIO

- `PQ.06` `XXXXXXXXXXXX`

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

- `XXXXXX` `conda` `XXXXXXXXXXXXXXXXXXXX` `Jetson` `XXXX` `GPIO` `XX`

```
pip install JETSON.GPIO
```

- `XXXXXXXXXXXXXXXXXXXX` `JETSON.GPIO` `XX`

```
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 `XXXXXX` I/O `XXXXXXXXXXXX`

- **BOARD** `XXXXXXXXXXXX` 40 `XXXXXXXXXXXX`
- **BCM** `XX` Broadcom SoC `XX` GPIO `XXXX` `XX` `XX`



```
/**
 * 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);
```


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

#37

edit 10 2025 06:39:30

edit 18 2025 07:00:05