4.API usage of GPIO library

The Jetson GPIO library provides all the public APIs provided by the RPi. GPIO library.

1.Import Library

To import the Jetson.GPIO module：

import Jetson.GPIO as GPIO

2.Pin number

The Jetson GPIO library provides four methods for numbering IO pins. The first two correspond to the modes provided by the RPi. GPIO library, namely BOARD and BCM, respectively referencing the pin numbers of the 40 pin GPIO connector and Broadcom SoC GPIO numbers. The other two modes CVM and TEGRA_ SOC uses strings instead of numbers, which correspond to signal names on the CVM CVB connector and Tegra SoC, respectively.

To specify which mode you are using (mandatory), use the following function calls:

GPIO.setmode(GPIO.BOARD)# or

GPIO.setmode(GPIO.BCM)# or

GPIO.setmode(GPIO.CVM)# or

GPIO.setmode(GPIO.TEGRA_SOC)

To check the set mode, you can call:

mode = GPIO.getmode()

This mode must be GPIO.BOARD, GPIO.BCM, GPIO.CVM, GPIO.TEGRA_ SOC or none.

3.Warning

The GPIO you are trying to use may already be in use outside of the current application. In this case, if the GPIO configuration used is any value other than the default direction (input), the Jetson GPIO library will issue a warning to you.

If you attempt to clean up before setting the mode and channel, it will also warn you. To disable warnings, use:

GPIO.setwarnings(False)

4.Set channel

Before being used as input or output, the GPIO channel must be set first.

Set the channel as input:

GPIO. setup (channel, GPIO. IN)

Set the channel to output:

```
GPIO. setup (channel, GPIO. OUT)
```

You can also specify an initial value for the output channel:

```
GPIO. setup (channel, GPIO. OUT, initial=GPIO. HIGH)
```

When setting one channel as output, multiple channels can also be set at once:

```
Channels=[18, 12, 13]

GPIO. setup (channels, GPIO. OUT)
```

5.Input

Read the value of a channel:

```
GPIO.input(channel)
```

This will return GPIO.LOW or GPIO.HIGH.

6.Output

Set the value of the pins configured as outputs

```
GPIO.output(channel, state)
```

The status can be GPIO.LOW or GPIO.HIGH.

Output to channel list or tuple:

```
channels = [18, 12, 13] # or use tuples

GPIO.output(channels, GPIO.HIGH) # or GPIO.LOW
```

Set first channel to HIGH and rest to LOW:

```
GPIO.output(channel, (GPIO.LOW, GPIO.HIGH, GPIO.HIGH))
```

7.Clear

At the end of the program, it is recommended to clean the channels in order to set all pins to the default state.

Clean all channels:

```
GPIO.cleanup()
```

Clear a single channel or channel list or tuple:

```
GPIO.cleanup(chan1) # cleanup only chan1

GPIO.cleanup([chan1, chan2]) # cleanup only chan1 and chan2

GPIO.cleanup((chan1, chan2)) # does the same operation as previous statement
```

More Detail, please check this link: https://github.com/NVIDIA/jetson-gpio