WF88 GPIO Application Note

Amp'ed RF Technology, Inc.

WF88 GPIO Application Note

1. GPIO Nodes

GPIO2, 3, 5, and 6 are available for customized application control.

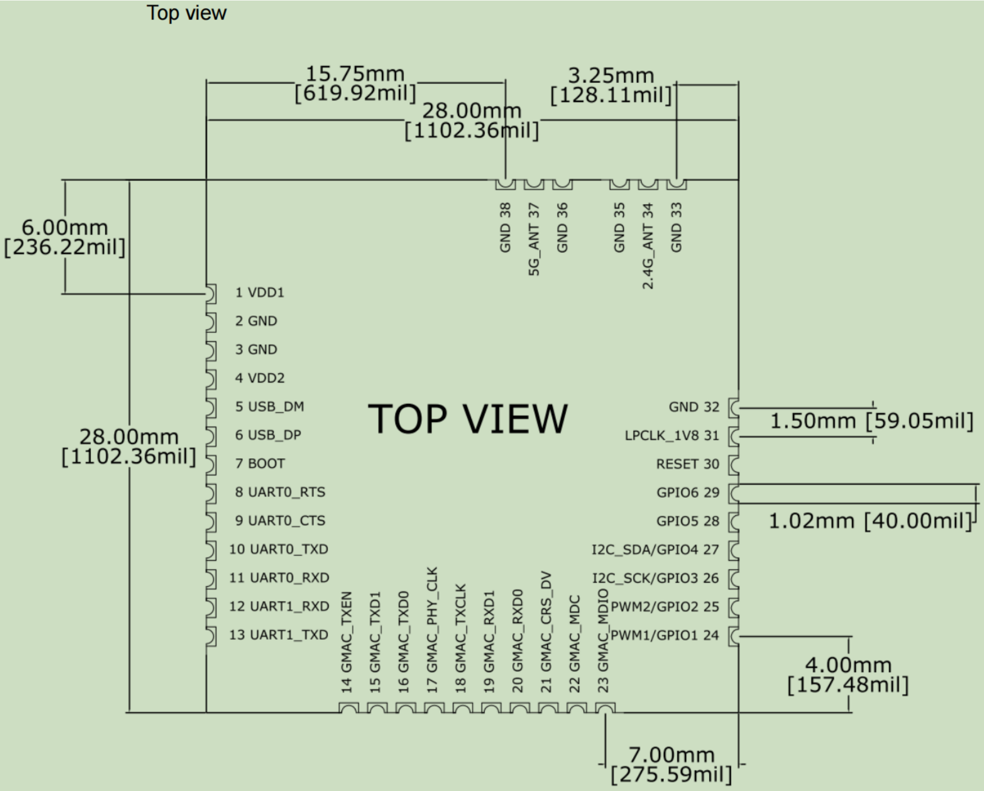
1.1. Apply for a node

Before operating a GPIO, it is necessary to apply/request for the GPIO. Note that when a GPIO has already been requested, any additional requests will fail.

1.1.1. The method for applying/releasing GPIO is as follows

```
$ cd /sys/class/gpio
$ echo [gpio_num] > export      # Apply for GPIO
$ echo [gpio_num] > unexport    # Release GPIO
```

1.1.2. GPIO pin numbers are shown in the following figure:



1.1.3. Default pin states

- After powering on WF88, in the UBoot program controls GPIO1 to send a reset signal, which is connected to the reset pin of the PHY chip to complete the reset of the PHY chip.
- GPIO2 defaults to pull-down level. After the Kernel starts, it is pulled up.
- GPIO3 and GPIO4 default to pull-up level. After the Kernel starts, they maintain high level. GPIO4 serves as

an indicator for Wi-Fi Mesh network status and TCP connection status. GPIO4 LED status display: default off (GPIO4 output high level), lit after connecting to the mesh network, flashing after successful TCP/MQTT connection.

- GPIO5 and GPIO6 default to pull-down level. After Kernel starts, they maintain low level.

1.1.4. GPIO software mapping

GPIO2: gpio_num=59

GPIO3: gpio_num=58

GPIO4: gpio_num=57

GPIO5: gpio_num=49

GPIO6: gpio_num=50

1.1.5. Apply for GPIO2 example

```
[root@Ingenic-g1_1:~]# cd /sys/class/gpio
[root@Ingenic-g1_1:gpio]# ls
export     gpio57     gpiochip0   gpiochip32  gpiochip64  unexport
[root@Ingenic-g1_1:gpio]# echo 59 > export
[root@Ingenic-g1_1:gpio]# ls
export     gpio57     gpio59     gpiochip0   gpiochip32  gpiochip64  unexport
[root@Ingenic-g1_1:gpio]#
```

1.1.6. Release GPIO2 example

```
[root@Ingenic-g1_1:gpio]# echo 59 > unexport
[root@Ingenic-g1_1:gpio]# ls
export     gpio57     gpiochip0   gpiochip32  gpiochip64  unexport
[root@Ingenic-g1_1:gpio]#
```

1.2. Set input/output configuration

After applying for the GPIO, enter the “gpioN” directory, such as GPIO2(gpio59), and set the input/output configuration. See example below:

```
[root@Ingenic-g1_1:~]# cd /sys/class/gpio
[root@Ingenic-g1_1:gpio]# ls
export     gpio57     gpiochip0   gpiochip32  gpiochip64  unexport
[root@Ingenic-g1_1:gpio]# echo 59 > export
[root@Ingenic-g1_1:gpio]# ls
export     gpio57     gpio59     gpiochip0   gpiochip32  gpiochip64  unexport
[root@Ingenic-g1_1:gpio]# cd gpio59
[root@Ingenic-g1_1:gpio59]# ls
active_low  direction   edge       subsystem   uevent      value
[root@Ingenic-g1_1:gpio59]# echo out > direction     # Set GPIO2 to output mode
```

```
[root@Ingenic-g1_1:gpio59]# echo in > direction              # Set GPIO2 to input mode
[root@Ingenic-g1_1:gpio59]#
```

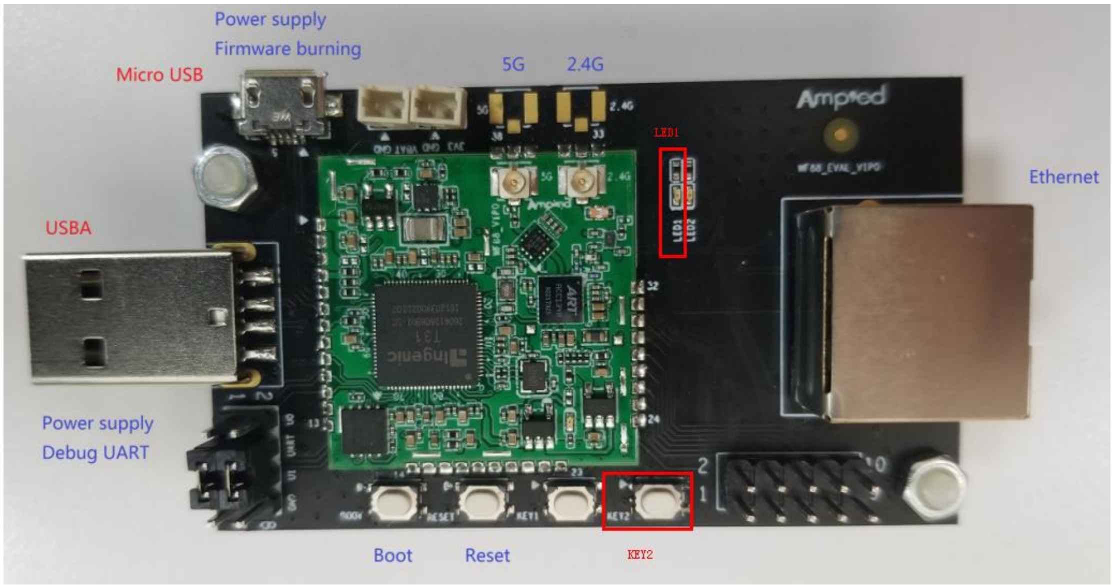
1.3. GPIO read and write

Use the value node in the “gpioN” directory to read or write. This represents the level of “gpioN”. When the GPIO is in input mode, the value read is the level of the GPIO. When the GPIO is in output mode, the value written to value is the output level of the GPIO.

For example, when configured GPIO5 (gpio49) to output test mode, proceed as follows:

```
[root@Ingenic-g1_1:~]# cd /sys/class/gpio
[root@Ingenic-g1_1:gpio]# echo 49 > export
[root@Ingenic-g1_1:gpio]# ls
export      gpio57      gpiochip0   gpiochip64
gpio49      gpio59      gpiochip32  unexport
[root@Ingenic-g1_1:gpio]# cd gpio49
[root@Ingenic-g1_1:gpio49]# ls
active_low  direction   edge        subsystem   uevent      value
[root@Ingenic-g1_1:gpio49]# echo out > direction
[root@Ingenic-g1_1:gpio49]# echo 1 > value  #Set GPIO5 output high level, LED1 goes out
[root@Ingenic-g1_1:gpio49]# echo 0 > value  #Set GPIO5 output low level, LED1 lights up
[root@Ingenic-g1_1:gpio49]#
```

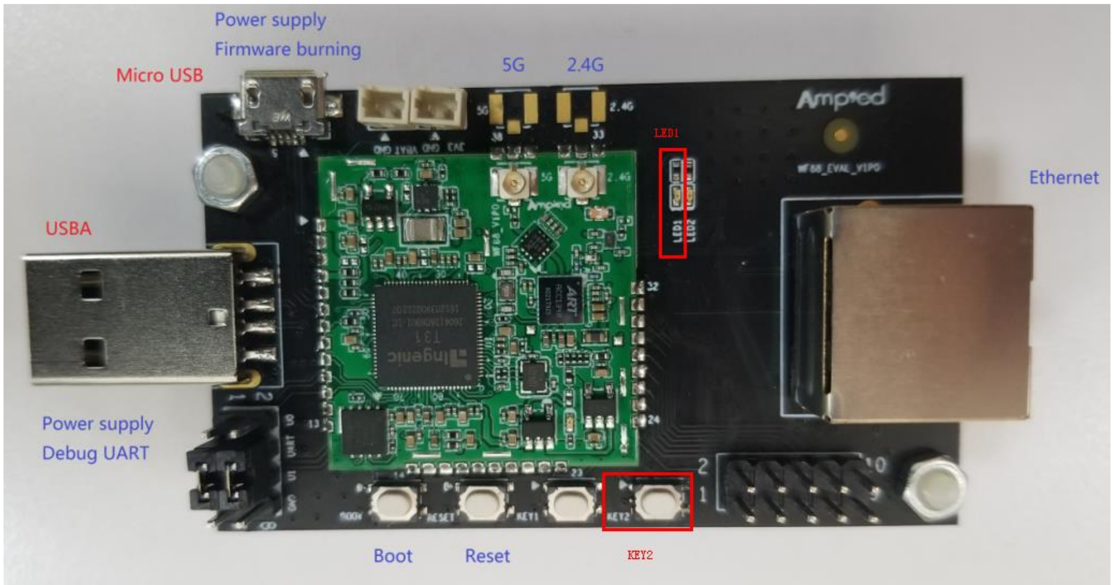
WF288 evaluation board GPIO5 example:



For example, when configured GPIO3 (gpio58) to input test mode, proceed as follows:

```
[root@Ingenic-g1_1:~]# cd /sys/class/gpio
[root@Ingenic-g1_1:gpio]# ls
export      gpio57      gpiochip0   gpiochip64
gpio49      gpio59      gpiochip32  unexport
[root@Ingenic-g1_1:gpio]# echo 58 > export
[root@Ingenic-g1_1:gpio]# ls
export      gpio57      gpio59      gpiochip32  unexport
gpio49      gpio58      gpiochip0   gpiochip64
[root@Ingenic-g1_1:gpio]# cd gpio58
[root@Ingenic-g1_1:gpio58]# ls
active_low  direction   edge       subsystem   uevent      value
[root@Ingenic-g1_1:gpio58]# echo in > direction
[root@Ingenic-g1_1:gpio58]# cat value          #Release KEY2, Read GPIO3 as high level 1
[root@Ingenic-g1_1:gpio58]# cat value          #Press and hold KEY2, Read GPIO3 as low level 0
```

WF288 evaluation board GPIO3 example:



2. Script to Control GPIO

The following script will use the vi editor to create a script file, edit the file, and finally save the file before exiting.

2.1. GPIO write script

Write a test script to control the high and low levels of GPIO5 output for example:

```
 [root@Ingenic-g1_1:~]# vi /system/etc/config/TestOutGpio.sh
#!/bin/sh

cd /sys/class/gpio/
if [ ! -d "gpio49" ]    #Determine if GPIO5 has been applied for
then
  echo 49 > export      #Apply for GPIO5
fi
cd gpio49
echo out > direction    #Set GPIO5 to output mode

while true
do
  echo 1 > value  #Set GPIO5 output high level, LED1 goes out
  sleep 1        #Set one second delay
  echo 0 > value  #Set GPIO5 output low level, LED1 lights up
  sleep 1
done

[root@Ingenic-g1_1:~]# sh /system/etc/config/TestOutGpio.sh
```

Run the script and you can see that LED1 flashes every 1 second.

2.2. GPIO read script

Write a test script to read the high and low levels of GPIO3 input for example:

```
[root@Ingenic-g1_1:~]# vi /system/etc/config/TestInGpio.sh
#!/bin/sh

cd /sys/class/gpio/
if [ ! -d "gpio58" ]    #Determine if GPIO3 has been applied for
```

```
then
  echo 58 > export      #Apply for GPIO3
fi
cd gpio58
echo in > direction    #Set GPIO3 to input mode

while true
do
  btnstat=$(cat value)    #Read the level status of GPIO3 and store the value in the btnstat variable
  if [ "$btnstat" -eq "0" ]  #Press and hold KEY2, Read GPIO3 as low level
  then
    echo "KEY2 is pressed"
  else                  #Release KEY2, Read GPIO3 as high level
    echo "KEY2 is released"
  fi
  sleep 1         #Set one second delay
done
```

```
[root@Ingenic-g1_1:~]# sh /system/etc/config/TestInGpio.sh
```

Run the script and press and release KEY2, with the following prompt messages:

```
KEY2 is released
KEY2 is pressed
```