



SIM7600 Series_Open Linux Sleep&Wake-Up_Application n Note

LTE Module

SIMCom Wireless Solutions Limited

Building B, SIM Technology Building, No.633, Jinzhong Road
Changning District, Shanghai P.R. China

Tel: 86-21-31575100

support@simcom.com

www.simcom.com

Document Title:	SIM7600 Series_Open Linux Sleep&Wake-Up_Application Note
Version:	2.00
Date:	2020.8.6
Status:	Released

GENERAL NOTES

SIMCOM OFFERS THIS INFORMATION AS A SERVICE TO ITS CUSTOMERS, TO SUPPORT APPLICATION AND ENGINEERING EFFORTS THAT USE THE PRODUCTS DESIGNED BY SIMCOM. THE INFORMATION PROVIDED IS BASED UPON REQUIREMENTS SPECIFICALLY PROVIDED TO SIMCOM BY THE CUSTOMERS. SIMCOM HAS NOT UNDERTAKEN ANY INDEPENDENT SEARCH FOR ADDITIONAL RELEVANT INFORMATION, INCLUDING ANY INFORMATION THAT MAY BE IN THE CUSTOMER'S POSSESSION. FURTHERMORE, SYSTEM VALIDATION OF THIS PRODUCT DESIGNED BY SIMCOM WITHIN A LARGER ELECTRONIC SYSTEM REMAINS THE RESPONSIBILITY OF THE CUSTOMER OR THE CUSTOMER'S SYSTEM INTEGRATOR. ALL SPECIFICATIONS SUPPLIED HEREIN ARE SUBJECT TO CHANGE.

COPYRIGHT

THIS DOCUMENT CONTAINS PROPRIETARY TECHNICAL INFORMATION WHICH IS THE PROPERTY OF SIMCOM WIRELESS SOLUTIONS LIMITED COPYING, TO OTHERS AND USING THIS DOCUMENT, ARE FORBIDDEN WITHOUT EXPRESS AUTHORITY BY SIMCOM. OFFENDERS ARE LIABLE TO THE PAYMENT OF INDEMNIFICATIONS. ALL RIGHTS RESERVED BY SIMCOM IN THE PROPRIETARY TECHNICAL INFORMATION , INCLUDING BUT NOT LIMITED TO REGISTRATION GRANTING OF A PATENT , A UTILITY MODEL OR DESIGN. ALL SPECIFICATION SUPPLIED HEREIN ARE SUBJECT TO CHANGE WITHOUT NOTICE AT ANY TIME.

SIMCom Wireless Solutions Limited

Building B, SIM Technology Building, No.633 Jinzhong Road, Changning District, Shanghai P.R.China

Tel: +86 21 31575100

Email: simcom@simcom.com

For more information, please visit:

<https://www.simcom.com/download/list-863-en.html>

For technical support, or to report documentation errors, please visit:

<https://www.simcom.com/ask/> or email to: support@simcom.com

Copyright © 2020 SIMCom Wireless Solutions Limited All Rights Reserved.

About Document

Version History

Version	Date	Owner	What is new
2020.8.6	2.00	Zhichong Huang	Update document format

Contents

About Document.....	3
Version History.....	3
Contents.....	4
1. Document Overview.....	5
2. Hardware interface.....	6
2.1 Description of related PIN.....	6
3. Module sleep.....	8
3.1 Process of sleep.....	8
3.1.1 The use of Wakelock.....	9
3.2 Sleep plan.....	9
3.2.1 Module sleep method.....	9
3.2.2 Sleep Debug.....	10
4. Wake up of Module.....	11
4.1 System wakes up.....	11
4.2 HOST wakes up.....	11
5. Module wakes up HOST.....	12
5.1 RI (PIN69) wakes up HOST method.....	12
5.2 USB remote wake-up host principle.....	12

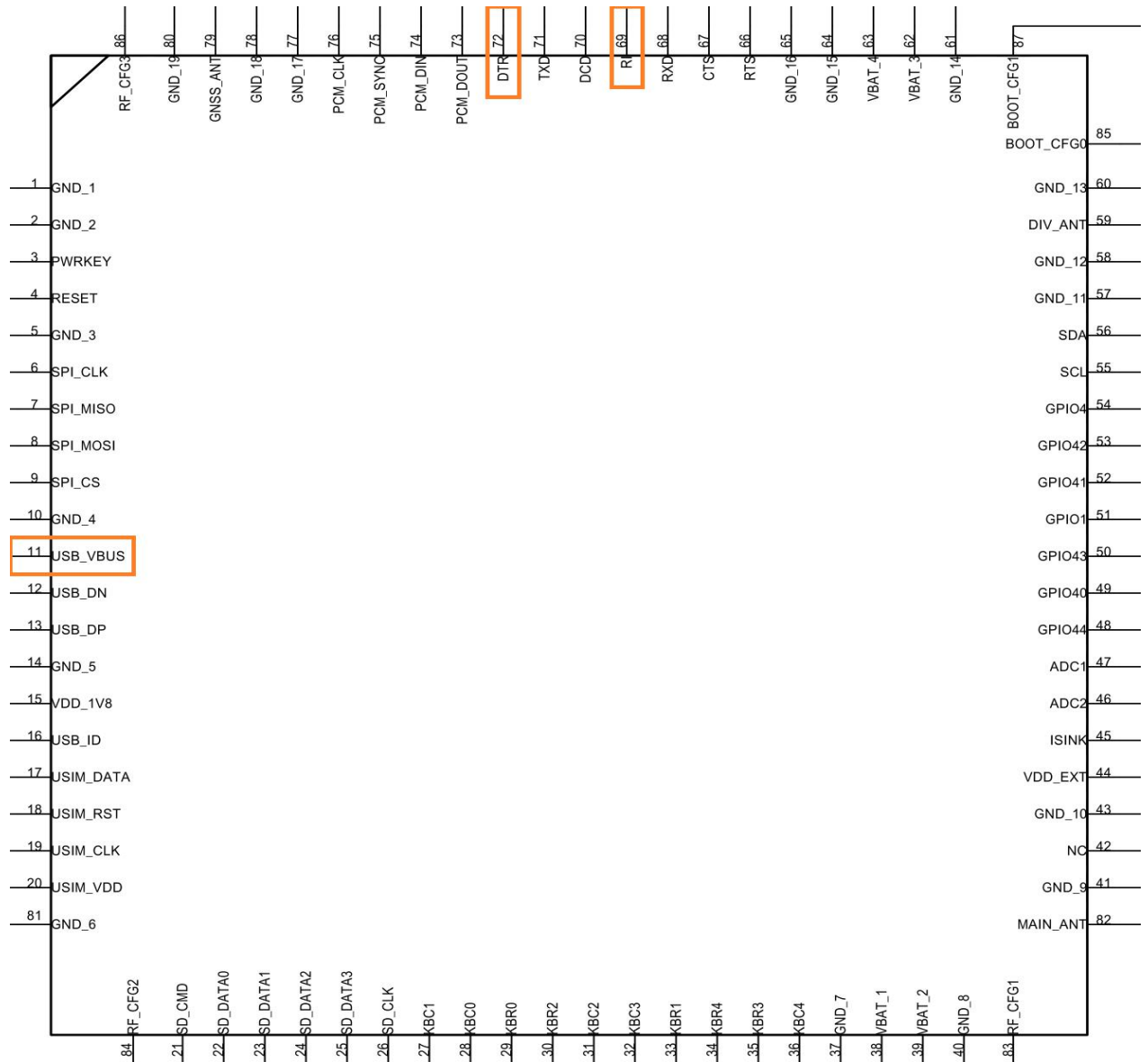
1. Document Overview

This document describes the sleep wake-up mechanism and operation flow of the SIM7600 module. It is applied to the secondary development of the module to open Linux systems. The documentation mainly describes the hibernation and wake-up of the module Linux system.

SIMCom
Confidential

2. Hardware interface

The red PIN in the following figure is related to sleep and wake-up.



2.1 Description of related PIN

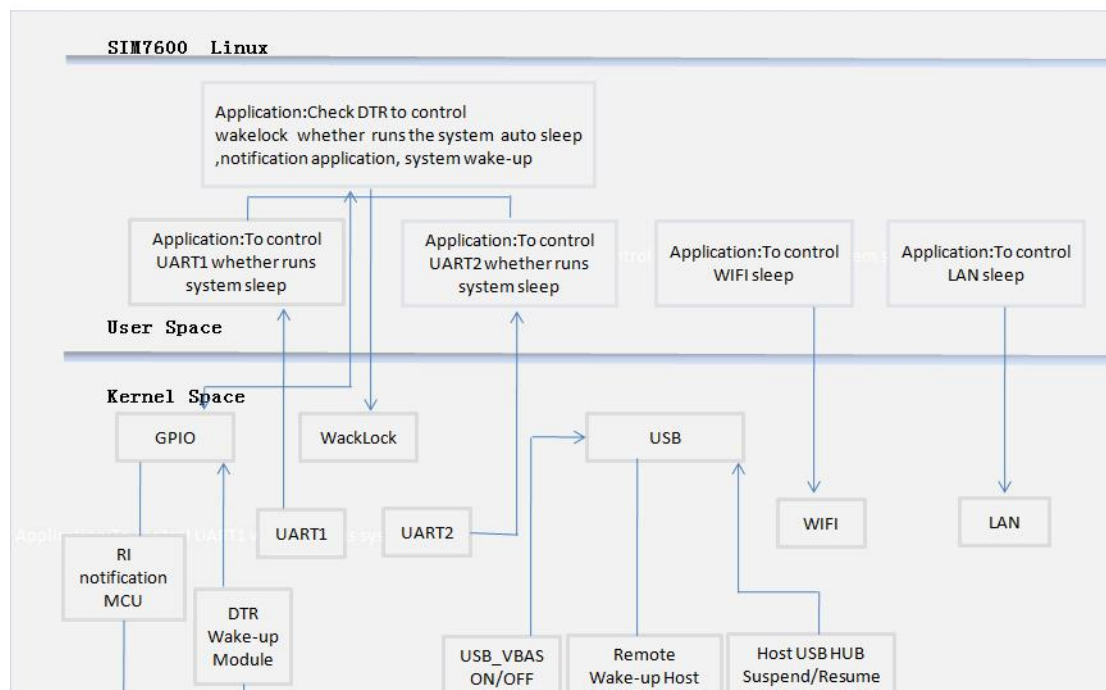
PIN name	PIN No.	IN / OUT	Instructions
RI	69	OUT	RI is mainly used for module 2 development applications to wake up HOST. The application can raise/lower the RI by operating the GPIO.
USB_VBUS	11	IN	Power-on and power-down of USB_VBUS can affect USB sleep/wake-up. USB sleep/wake is directly related to system sleep/wake-up.
DTR	72	IN	DTR is mainly used for module 2 development of application management system sleep/wake-up. The application program can detect the DTR interrupt in real time and determine the system's sleep/keep awake according to the level of GPIO.

SIMCom
Confidential

3.Module sleep

3.1 Process of sleep

The following is a system sleep flow chart:



Linux sleep process

Due to some differences in the hardware versions of the SIM7600 and the devices in the Linux system are also have some differences. Let's take the MIFI version as an example. The hibernation of Linux depends on the status of each device in the system and the application's use of the wakelock.

1. USB sleep requires HOST suspend module USB or VBus power off.
2. WIFI must be turned off before the system goes to sleep. WIFI can be turned off by AT command AT + CWMAP = 0 or API relates WIFI.
3. If there is a LAN port, call the LAN driver to uninstall the API and let the LAN port go to sleep.
4. The UART's sleep needs application management. UART does not sleep so that the system will not enter sleep.
5. All apps release Wakelock. Linux will sleep only if all devices are in suspending state and Wakelock is not in use.

3.1.1 The use of Wakelock

Wakelock needs to be used when the application wants to perform certain tasks while the system remains dormant. The method of use Wakelock is as follows:

When the application wants the system not to sleep, open the `/sys/power/wake_lock` file and write a specific string to the file (the naming rule of the string: `[process name]_[do not sleep reason]`).

After the application finishes the task it needs to release Wakelock. When it is released, open the `/sys/power/wake_unlock` file and write the string to the file. The string must be the same as the one written to Wakelock before.

The system will be allowed to sleep when all applications release Wakelock.

3.2 Sleep plan

3.2.1 Module sleep method

1. Close WIFI
2. If there has a 9730 LAN port (that is, HSIC turn to LAN port), you need to perform the following operations.
 - 1) `AT+CHSICDEVSLEEP=1`
 - 2) `AT+CLANCTRL=5`.
3. For the SGMII port, it needs execute command as follow.
`AT+CLANCTRL=5`
Note: The versions after 2018-7-12 does not need to call the LAN port driver for uninstall API.
4. The application releases all Wakelocks.
5. HOST pulls DTR high and allows the module to sleep.
6. The application stops sending data to the UART and sets the UART to sleep.
Set to allow the serial port to sleep:
 - a) `echo auto > /sys/class/tty/ttyHS0/device/power/control`
 - b) `echo auto > /sys/class/tty/ttyHS1/device/power/control`
 - c) `echo auto > /sys/class/tty/ttyHS2/device/power/control`
7. HOST suspends USB or disconnects VBUS

3.2.2 Sleep Debug

1. Check whether WIFI is off, AT+CWMAP? 0 means is off.
 2. Check DTR level status
 3. Check the usage of wake_lock, adb shell cat /sys/power/wakelock
 4. Check whether UART is allowed to sleep, auto allows to sleep
 - a) adb shell cat /sys/class/tty/ttyHS0/device/power/control
 - b) adb shell cat /sys/class/tty/ttyHS1/device/power/control
 - c) adb shell cat /sys/class/tty/ttyHS2/device/power/control
 5. Check device sleep status,
 - i. adb shell cat /sys/kernel/debug/wakeup_sources > wake.log
 - ii. adb shell cat /sys/kernel/debug/wakeup_sources > wake1.log
- Comparing the wake.log and wake1.log files, the difference is the device that cannot sleep.

SIMCom
Confidential

4. Wake up of Module

The wake-up of the module is mainly divided into two situations: the system wakes up and the HOST wakes up.

4.1 System wakes up

The system wakes up. Generally affect the host has the following 3 kinds of situations:

1. Calls and SMS
2. Web server sends data to module
3. RTC timer

After all self-wakeups complete the task, the system automatically goes to sleep.

4.2 HOST wakes up

Host wakes up module mainly through USB, GPIO:

1. There are two conditions for the host to wake the system via USB:
 - a) Module USB is in suspending state and host wakes up USB bus and module USB.
 - b) Module USB powers off Vbus. Host powers on Vbus and enumerates USB devices to wake up the system.
2. DTR(PIN72) can support wake-up interrupt
For example: Configure wake-up on the falling edge of DTR the following parameters are required:
 - a) `echo falling > /sys/class/gpio/gpio74/edge`
 - b) `echo 1 > /sys/class/gpio/gpio74/ can_wakeup`

Any one of the above conditions can wake up the Linux subsystem.

5. Module wakes up HOST

There are generally two methods for the module to wake up the host: RI (PIN69) and USB. The GPIO wakes up the host through the pulse, and the USB wakes up the host through REMOTE_WAKEUP.

5.1 RI (PIN69) wakes up HOST method

The wakeable GPIO of HOST in hardware connects the RI (PIN69) of module. HOST interrupts detection RI (PIN69). The module application changes the status of the RI (PIN69) to notify HOST when the module has a message to notify HOST. It is generally recommended to pull a pulse, such as the following waveform:



5.2 USB remote wake-up host principle

When the host goes to sleep, the USB bus of the host is in the suspend state. If the URC module reports to the host at this time, the module requests to resume communication in the suspended state and initiates a resume signal to the host USB controller. The host USB controller enters the resume state and holds for 20ms. This 20ms ensures that all devices on the bus will receive the resume signal and resume the USB bus after 20ms. The USB bus enters the idle state and must start sending SOF packets within 3ms otherwise the module USB will hang again.

Description of Module Remote Wakes up Host Features:

This feature allows the USB controller to accept the resume signal from the module. When the USB controller is remotely woken up, the host is also woken up.

The module also needs the feature that supports Remote wakeup. This feature is set in the 'bmAttributes' field of the device descriptor. Host side enables module's Remote Wakeup feature through SET_FEATURE and turn off the module's Remote Wakeup feature via CLEAR_FEATURE.