Rockchip Developer Guide Linux WDT

ID: RK-KF-YF-083

Release Version: V1.1.0

Release Date: 2021-04-13

Security Level: □Top-Secret   □Secret   □Internal   ■Public

Rockchip Electronics Co., Ltd.

No.18 Building, A District, No.89, software Boulevard Fuzhou, Fujian,PRC

Website: www.rock-chips.com

Customer service Tel: +86-4007-700-590

Customer service Fax: +86-591-83951833

Customer service e-Mail: fae@rock-chips.com

Preface

Overview

When the count value of WDT is reduced to 0, a reset signal is generated to reset the system to prevent the system from being stuck due to software.

Product Version

Chipset	Kernel Version
ALL	Kernel4.4& 4.19

Intended Audience

This document (this guide) is mainly intended for:

Technical support engineers
Software development engineers

Revision History

Version	Author	Date	Change Description
V1.0.0	Simon.Xue	2019-12-23	Initial version
V1.1.0	Simon.Xue	2021-04-13	Add RK356X stop counting function

Content

WDT Driver

The driver file is in:
`drivers/watchdog/dw_wdt.c`

DTS Node Configuration

The reference document for DTS configuration is `Documentation/devicetree/bindings/watchdog/dw_wdt.txt` , this document mainly introduces the follow parameter :

- `interrupts = <GIC_SPI 120 IRQ_TYPE_LEVEL_HIGH 0>;`

The interrupt mode is to trigger an interrupt first, and then generate a reset signal after a timeout period.

- `clocks = <&cru PCLK_WDT>;`

This attribute function is to drive WDT to work, and calculate each counting cycle.

WDT Usage

The application operates the `/dev/watchdog` node to control the watchdog. Examples are as follows:

```c
int main(void)
{
    int fd = open("/dev/watchdog", O_WRONLY);    //here through open to start
watchdog
    int ret = 0;
    if (fd == -1) {
        perror("watchdog");
        exit(EXIT_FAILURE);
    }
    while (1) {
        ret = write(fd, "\0", 1);     //Feed dogs by write
        if (ret != 1) {
            ret = -1;
            break;
        }
        sleep(10);
    }
    close(fd);
    return ret;
}
```

About `close()`

1. Under normal condition, executing `close ()`, which no longer feeding the dog and watchdog restart automatically .

`echo A > /dev/watchdog`, where any characters other than character V are written

2. `write (fd," V ", 1);` Then `close ()`, write uppercase V, the kernel continues to feed the dog, the system will not automatically restart.

`echo V > /dev/watchdog`

3. Configure the macro `CONFIG_WATCHDOG_NOWAYOUT`, repeat step 2, the kernel will not continue to feed the dog, and the system will be restarted.

Kernel Configuration

```
Symbol: WATCHDOG [=y]
Type  : boolean
Prompt: Watchdog Timer Support
    Location:
(1) -> Device Drivers
    Defined at drivers/watchdog/Kconfig:6
```

FAQ

WDT Can't Stop

The legacy version of WDT does not have a matched register to configure the stop function, which can only be stopped by disable clock or soft reset. The clock or reset operation of some Rockchip' s product can only be performed in a safe environment. The new version of WDT adds a stop function in the future.

WDT Accuracy

The accuracy of WDT is only 16 levels, and the counting of adjacent level is quite different, so it cannot be counted finely.

```
0000: 0x0000ffff
0001: 0x0001ffff
0010: 0x0003ffff
0011: 0x0007ffff
0100: 0x000fffff
0101: 0x001fffff
0110: 0x003fffff
0111: 0x007fffff
1000: 0x00ffffff
1001: 0x01ffffff
1010: 0x03ffffff
1011: 0x07ffffff
1100: 0x0fffffff
1101: 0x1fffffff
1110: 0x3fffffff
1111: 0x7fffffff
```

Assuming that the wdt clock is 100MHz, the maximum timeout time is `0x7fffffff/100MHz = 21` seconds. If a larger timeout is required, the corresponding WDT clock needs to be adjusted.

PAUSE COUNTING

RK356X add pause counting function, use io command that from Rockchip or busybox's devmem to test pause counting and resume counting.

enable

```
CONFIG_DEVMEM
```

disable

```
CONFIG_STRICT_DEVMEM
```

Address 0xfdc60504 belongs to the register GRF_SOC_CON1 of SYS_GRF. write 0x1 to bit4 pause counting, write 0x0 to bit4 resume counting. Higher 16 bits are writable mask bits.

pause counting.

```
io -4 0xfdc60504 0x00100010
```

or

```
busybox devmem 0xfdc60504 32 0x00100010
```

resume counting.

```
io -4 0xfdc60504 0x00100000
```

or

```
busybox devmem 0xfdc60504 32 0x00100000
```