

Rockchip Developer Guide Linux WDT

文件标识：RK-KF-YF-078

发布版本：V1.2.0

日期：2022-01-28

文件密级：绝密 秘密 内部资料 公开

免责声明

本文档按“现状”提供，瑞芯微电子股份有限公司（“本公司”，下同）不对本文档的任何陈述、信息和内容的准确性、可靠性、完整性、适销性、特定目的性和非侵权性提供任何明示或暗示的声明或保证。本文档仅作为使用指导的参考。

由于产品版本升级或其他原因，本文档将可能在未经任何通知的情况下，不定期进行更新或修改。

商标声明

“Rockchip”、“瑞芯微”、“瑞芯”均为本公司的注册商标，归本公司所有。

本文档可能提及的其他所有注册商标或商标，由其各自拥有者所有。

版权所有 © 2021 瑞芯微电子股份有限公司

超越合理使用范畴，非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

瑞芯微电子股份有限公司

Rockchip Electronics Co., Ltd.

地址：福建省福州市铜盘路软件园A区18号

网址：www.rock-chips.com

客户服务电话：+86-4007-700-590

客户服务传真：+86-591-83951833

客户服务邮箱：fae@rock-chips.com

前言

概述

当WDT的计数值减为0的时候，产生一个复位信号复位系统，防止由软件导致的系统卡死。

产品版本

芯片名称	内核版本
ROCKCHIP 芯片	4.4/4.19/5.10

读者对象

本文档（本指南）主要适用于以下工程师：

技术支持工程师

软件开发工程师

修订记录

日期	版本	作者	修改说明
2019.12.23	V1.0.0	薛小明	初始发布
2021-04-13	V1.1.0	薛小明	添加RK356X暂停计数说明
2022-01-28	V1.2.0	薛小明	添加RK3588暂停计数说明

1. WDT 驱动

1.1 驱动文件

驱动文件所在位置:

```
drivers/watchdog/dw_wdt.c
```

1.2 DTS 节点配置

DTS 配置参考文档为 `Documentation/devicetree/bindings/watchdog/dw_wdt.txt`, 本文主要说明如下参数:

- `interrupts = <GIC_SPI 120 IRQ_TYPE_LEVEL_HIGH 0>;`
中断模式时候用于首先触发中断，再经过一个超时周期才产生复位信号。
- `clocks = <&cru PCLK_WDT>;`
驱动WDT工作，并且用于计算每个计数周期。

2. WDT 使用

应用操作 `/dev/watchdog` 节点来控制watchdog，示例如下:

```
int main(void)
{
    int fd = open("/dev/watchdog", O_WRONLY);           // 通过open来启动watchdog
    int ret = 0;
    if (fd == -1) {
        perror("watchdog");
        exit(EXIT_FAILURE);
    }
    while (1) {
        ret = write(fd, "\0", 1);                      // 通过write来喂狗
        if (ret != 1) {
            ret = -1;
            break;
        }
        sleep(10);
    }
    close(fd);
    return ret;
}
```

关于 `close()`

1. 正常情况下 `close()`，不再喂狗，watchdog会自动重启。

```
echo A > /dev/watchdog
```

, 这里写入的是除大写V以外的任意字符。

2. `write(fd, "V", 1);` 再 `close()`，写入大写V，内核继续喂狗，系统不会自动重启。

```
echo V > /dev/watchdog
```

3. 配置宏 `CONFIG_WATCHDOG_NOWAYOUT`，重复步奏2，内核不会继续喂狗，系统会被重启。

3. 内核配置

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

4. 常见问题

4.1 WDT无法停止

旧版本WDT没有相应的寄存器可以配置停止功能，只能通过disable clock或者软复位来停止WDT，有些芯片的clock或者复位操作只能在安全环境执行，未来新版本的WDT添加了停止功能。

4.2 WDT精度

WDT精度只有16档，相邻档位计数相差比较大，因此无法精细计数。

```
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
```

假设wdt clock为100MHz，最大超时时间 $0x7fffffff / 100MHz = 21$ 秒，如果需要更大的超时，需要调整对应的wdt clock。

4.3 RK356X暂停功能

使用Rockchip自带的io命令或者busybox的devmem命令可以实现暂停计数以及恢复计数。

打开

```
CONFIG_DEVMEM
```

关闭

```
CONFIG_STRICT_DEVMEM
```

0xfd60504来自SYS_GRF的GRF_SOC_CON1寄存器，对bit4写1暂停计数，写0恢复计数，高16位为写使能位。

暂停计数

```
io -4 0xfd60504 0x00100010
```

或者

```
busybox devmem 0xfd60504 32 0x00100010
```

恢复计数

```
io -4 0xfd60504 0x00100000
```

或者

```
busybox devmem 0xfd60504 32 0x00100000
```

4.4 RK3588暂停功能

暂停计数

```
io -4 0xfd58c000 0x00010001
```

或者

```
busybox devmem 0xfd58c000 32 0x00010001
```

恢复计数

```
io -4 0xfd58c000 0x00100000
```

或者

```
busybox devmem 0xfd58c000 32 0x00100000
```

