

RT-Thread WATCHDOG开发指南

文件标识: RK-KF-YF-103

发布版本: V1.1.0

日期: 2021-04-13

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

产品版本

芯片名称	RT-Thread 版本
RK2108/Pisces	3.1.3

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

版本号	作者	修改日期	修改说明
V1.0.0	Simon	2020-02-21	第一次版本发布
V1.0.1	Simon	2020-05-27	修正格式
V1.1.0	Simon	2021-04-13	添加RK356X暂停计数功能

目录

RT-Thread WATCHDOG开发指南

- 1. RT-Thread WATCHDOG配置
 - 1.1 RT-Thread WATCHDOG CONFIG
 - 1.2 RT-Thread 常用API
 - 1.3 RT-Thread 使用示例
- 2. TEST
 - 2.1 CONFIG配置
 - 2.2 USAGE
 - 2.3 RK356X暂停功能
- 3. WDT精度
- 4. 开发指南

1. RT-Thread WATCHDOG配置

1.1 RT-Thread WATCHDOG CONFIG

```
scons --menuconfig

RT-Thread Components --->
  Device Drivers --->
    [*] Using Watch Dog device drivers
```

1.2 RT-Thread 常用API

```
int wdt_dev_init(void);
void rt_wdt_irqhandler(void);
rt_err_t dw_wdt_control(rt_watchdog_t *wdt, int cmd, void *arg);
rt_err_t dw_wdt_init(rt_watchdog_t *wdt);
rt_err_t dw_wdt_start(uint32_t type);
rt_err_t dw_wdt_stop(void);
```

1.3 RT-Thread 使用示例

使用示例：

```
wdt_dev_init(void); /* 注册中断，注册设备 */
dw_wdt_init(rt_watchdog_t *wdt); /* 使能clock，初始化WDT */
dw_wdt_start(uint32_t type); /* 设置工作模式，并开启WDT */
dw_wdt_stop(void); /* 停止WDT */
```

2. TEST

2.1 CONFIG配置

```
RT-Thread bsp test case --->
  RT-Thread Common Test case --->
    [*] Enable BSP Common WDT TEST
```

2.2 USAGE

使用示例：

```
wdt_test probe dw_wdt /* 打开WDT设备 */
wdt_test settimout 10 /* 设置10秒超时 */
wdt_test start type /* 设置运行模式并启动且自动喂狗, type = 1:中断模式, type = 0:立即重启模式 */
wdt_test reboot /* 停止喂狗 */
```

2.3 RK356X暂停功能

使用Rockchip自带的io命令可以实现暂停计数以及恢复计数。

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
```

恢复计数

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

3. 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. 开发指南

我司 WDT 驱动遵循 RTT 系统标准 WDT 驱动框架，因此可直接参考 RTT 官方[WDT开发指南](#)。