



HAL_V2 WATCHDOG

开发指南

版本号: 1.0

发布日期: 2025.7.15

版本历史

版本号	日期	制/修订人	内容描述
1.0	2025.7.15	AWA2351	初始版本



目 录

1 前言	1
1.1 文档简介	1
1.2 目标读者	1
1.3 适用范围	1
1.4 文档约定	1
1.4.1 标志说明	1
1.5 相关术语介绍	2
1.5.1 硬件术语	2
1.5.2 软件术语	2
2 模块介绍	3
2.1 模块功能介绍	3
2.2 模块配置介绍	3
2.3 模块源码结构	3
3 模块接口说明	5
3.1 接口列表	5
3.2 接口使用说明	5
3.2.1 hal_watchdog_init	5
3.2.2 hal_watchdog_disable	5
3.2.3 hal_watchdog_start	6
3.2.4 hal_watchdog_stop	6
3.2.5 hal_watchdog_feed	6
3.2.6 hal_watchdog_set_timeout	6
4 模块使用范例	7
5 FAQ	9

1 前言

1.1 文档简介

介绍 HAL_V2 中 WATCHDOG 驱动接口及使用方法，为 WATCHDOG 使用者提供参考。

1.2 目标读者

WATCHDOG 驱动的开发/维护人员。

1.3 适用范围

Table: 适用产品列表

表 1-1: 适用产品列表

产品名称	内核版本	驱动文件
T536	FreeRTOS	hal_watchdog.c
T153	FreeRTOS	hal_watchdog.c

1.4 文档约定

1.4.1 标志说明

⚠ 注意

- 提醒操作中应注意的事项。不当的操作可能会损坏器件，影响可靠性、降低性能等。

 说明

准确理解文中指令、正确实施操作而提供的补充或强调信息。

 技巧

一些容易忽视的小功能、技巧。了解这些功能或技巧能帮助解决特定问题或者节省操作时间。

1.5 相关术语介绍

1.5.1 硬件术语

表 1-2: 硬件术语

术语	解释说明
WATCHDOG	看门狗

1.5.2 软件术语

表 1-3: 软件术语

术语	解释说明
HAL	Hardware Abstraction Layer, 硬件抽象层

2 模块介绍

2.1 模块功能介绍

WATCHDOG 模块可以用于实现 cpu 核复位和 soc 复位等

2.2 模块配置介绍

menuconfig 配置项

```
DRIVERS_V2_WATCHDOG //打开驱动
HAL_TEST_WATCHDOG   //打开测试，可不开
```

2.3 模块源码结构

```
hal_v2/hal/source/WATCHDOG/
├── hal_watchdog.c
├── common_watchdog.h
├── Kconfig
├── Makefile
├── platform
│   ├── watchdog_sun8iw22.h
│   └── watchdog_sun55iw6.h
└── platform_watchdog.h
```

2. 驱动 APIs 声明头文件

```
hal_v2/include/hal/
└── hal_watchdog.h
```

3. 驱动 APIs 测试代码

```
hal_v2/hal/examples/WATCHDOG/
├── watchdog_interrupt
│   ├── watchdog_interrupt.c
│   └── Makefile
```

```
|— watchdog_reset_to_cpu_core
| |— watchdog_reset_to_cpu_core.c
| |— Makefile
|— watchdog_reset_to_whole_system
| |— watchdog_reset_to_whole_system.c
| |— Makefile
```



3 模块接口说明

3.1 接口列表

表 3-1: 模块接口列表

API	解释说明
hal_watchdog_init	WATCHDOG 模块初始化
hal_watchdog_disable	WATCHDOG 模块除能
hal_watchdog_start	WATCHDOG 模块启动
hal_watchdog_stop	WATCHDOG 模块停止
hal_watchdog_feed	WATCHDOG 模块喂狗
hal_watchdog_set_timeout	WATCHDOG 复位超时时间

3.2 接口使用说明

3.2.1 hal_watchdog_init

- 作用：WATCHDOG 模块初始化
- 参数：

— wdt_base 代表 WATCHDOG 基地址

WATCHDOG 配置参数结构体

- 返回值：无

3.2.2 hal_watchdog_disable

- 作用：WATCHDOG 模块除能
- 参数：

- wdt_base 代表 WATCHDOG 基地址

返回值：无

3.2.3 hal_watchdog_start

- 作用：启动 WATCHDOG 模块
- 参数：
 - wdt_base 代表 WATCHDOG 基地址
- 返回值：无

3.2.4 hal_watchdog_stop

- 作用：停止 WATCHDOG 模块
- 参数：
 - wdt_base 代表 WATCHDOG 基地址
- 返回值：无

3.2.5 hal_watchdog_feed

- 作用：WATCHDOG 模块喂狗
- 参数：
 - wdt_base 代表 WATCHDOG 基地址
- 返回值：无

3.2.6 hal_watchdog_set_timeout

WATCHDOG 超时时间

- 参数：
 - wdt_base 代表 WATCHDOG 基地址
 - timeout 代表超时时间
- 返回值：无

4 模块使用范例

可参考驱动 APIs 测试代码 (hal_v2/hal/examples/watchdog/watchdog_reset_to_whole_system/watchdog_reset_to_whole_system.c)，具体实现如下：

```
#include <stdio.h>
#include <stdlib.h>

#include <hal_interrupt.h>
#include <hal_watchdog.h>

#if defined(CONFIG_KERNEL_FREERTOS)
#include <hal_log.h>
#include <hal_cmd.h>
#include <hal_thread.h>
#include <hal_timer.h>
#elif defined(CONFIG_KERNEL_BAREMETAL)
#include "shell.h"
#endif

void hal_v2_watchdog_reset_to_whole_system(void)
{
    int to;

    printf("====WATCHDOG TEST====\n");

    wdt_init_t wdt_initstruct;
    wdt_initstruct.timeout = 5;
    wdt_initstruct.mode = RESET_TO_WHOLE_SYSTEM;
    hal_watchdog_init(WDT_CPUX, &wdt_initstruct);

    printf("\nTEST1: Feeding watchdog, timeout=5s. Should not reset...\n");
    to = 5;
    while (to-- > 0) {
        hal_watchdog_feed(WDT_CPUX);
        udelay(1 * 1000 * 1000);
    }
    udelay(1 * 1000 * 1000);

    printf("\nTEST2: Disabling watchdog, timeout=3s. Should not reset...\n");
    wdt_initstruct.timeout = 3;
    hal_watchdog_init(WDT_CPUX, &wdt_initstruct);
    hal_watchdog_disable(WDT_CPUX);
    udelay(5 * 1000 * 1000);

    printf("\nTEST3: Watchdog bark, timeout=1s. Waiting for whole system reset...\n");
    wdt_initstruct.timeout = 1;
    hal_watchdog_init(WDT_CPUX, &wdt_initstruct);
    udelay(2 * 1000 * 1000);

    return;
}
```

```
}  
  
#if defined(CONFIG_KERNEL_FREERTOS)  
FINSH_FUNCTION_EXPORT_CMD(hal_v2_watchdog_reset_to_whole_system, watchdog_reset_to_whole_system_test,  
    watchdog hal_v2 APIs tests)  
#elif defined(CONFIG_KERNEL_BAREMETAL)  
SHELL_EXPORT_CMD(  
    SHELL_CMD_PERMISSION(0)|SHELL_CMD_TYPE(SHELL_TYPE_CMD_FUNC)|SHELL_CMD_DISABLE_RETURN,  
    watchdog_reset_to_whole_system_test, hal_v2_watchdog_reset_to_whole_system, test for watchdog reset system);  
#endif
```



5 FAQ

无



声明

版权所有 © 2025 珠海全志科技股份有限公司。保留一切权利。

本文档及内容受著作权法保护，其著作权由珠海全志科技股份有限公司（“全志”）拥有并保留一切权利。

本文档是全志的原创作品和版权财产，未经全志书面许可，任何单位和个人不得擅自摘抄、复制、修改、发表或传播本文档内容的部分或全部，且不得以任何形式传播。

商标声明



（不全列）均为珠海全志科技股份有限公司的商标或者注册商标。在本文档描述的产品中出现的其它商标，产品名称，和服务名称，均由其各自所有人拥有。

免责声明

您购买的产品、服务或特性应受您与珠海全志科技股份有限公司（“全志”）之间签署的商业合同和条款的约束。本文档中描述的全部或部分产品、服务或特性可能不在您所购买或使用的范围内。使用前请认真阅读合同条款和相关说明，并严格遵循本文档的使用说明。您将自行承担任何不当使用行为（包括但不限于如超压，超频，超温使用）造成的不利后果，全志概不负责。

本文档作为使用指导仅供参考。由于产品版本升级或其他原因，本文档内容有可能修改，如有变更不另行通知。全志尽全力在本文档中提供准确的信息，但并不确保内容完全没有错误，因

使用本文档而发生损害（包括但不限于间接的、偶然的、特殊的损失）或发生侵犯第三方权利事件，全志概不负责。本文档中的所有陈述、信息和建议并不构成任何明示或暗示的保证或承诺。

本文档未以明示或暗示或其他方式授予全志的任何专利或知识产权。在您实施方案或使用产品的过程中，可能需要获得第三方的权利许可。请您自行向第三方权利人获取相关的许可。全志不承担也不代为支付任何关于获取第三方许可的许可费或版税（专利税）。全志不对您所使用的第三方许可技术做出任何保证、赔偿或承担其他义务。