



SigmaStar Camera Watchdog User Guide

Version 0.1



© 2020 SigmaStar Technology Corp. All rights reserved.

SigmaStar Technology makes no representations or warranties including, for example but not limited to, warranties of merchantability, fitness for a particular purpose, non-infringement of any intellectual property right or the accuracy or completeness of this document, and reserves the right to make changes without further notice to any products herein to improve reliability, function or design. No responsibility is assumed by SigmaStar Technology arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights, nor the rights of others.

SigmaStar is a trademark of SigmaStar Technology Corp. Other trademarks or names herein are only for identification purposes only and owned by their respective owners.



REVISION HISTORY

Revision No.	Description	Date
0.1	<ul style="list-style-type: none">Initial release	12/18/2019



TABLE OF CONTENTS

REVISION HISTORY i

TABLE OF CONTENTS..... ii

1. OVERVIE 1

 1.1. The General Description 1

2. WATCHDOG control 2

 2.1. Opening Watchdog 2

 2.2. Closing Watchdog 2

 2.3. Setting Timeout..... 2

 2.4. Keeping Watchdog Alive..... 3



1. OVERVIE

1.1. The General Description

The HW watchdog is designed based on standard Linux architecture. Upper application can set the desired time-out period within which to keep the watchdog alive.

The watchdog is disabled by default. User can decide whether to enable it or not. If enabled, the watchdog operation should preferably be done on the main thread, rather than the secondary thread, since the watchdog will be closed once the secondary thread is closed

2. WATCHDOG CONTROL

2.1. Opening Watchdog

To open the watchdog, open the /dev/watchdog device.

A reference code is as follows:

```
int wdt_fd = -1;
wdt_fd = open("/dev/watchdog", O_WRONLY);
if (wdt_fd == -1)
{
    // fail to open watchdog device
}
```

2.2. Closing Watchdog

A reference code for closing the watchdog is as follows:

```
int option = WDIOF_DISABLECARD;
ioctl(wdt_fd, WDIOF_SETOPTIONS, &option);
if (wdt_fd != -1)
{
    close(wdt_fd);
    wdt_fd = -1;
}
```

2.3. Setting Timeout

To set the timeout, use the standard IOCTL command WDIOF_SETTIMEOUT. The unit is second. It is suggested that the timeout period be greater than 5s.

A reference code is as follows:

```
#define WATCHDOG_IOCTL_BASE 'W'
#define WDIOF_SETTIMEOUT      _IOWR(WATCHDOG_IOCTL_BASE, 6, int)
int timeout = 20;
ioctl(wdt_fd, WDIOF_SETTIMEOUT, &timeout);
```

2.4. Keeping Watchdog Alive

To keep the watchdog alive, use the standard IOCTL command `WDIOC_KEEPALIVE`. The time to feed the watchdog is decided by the set timeout period. The feeding time should be shorter than the timeout period.

A reference code is as follows:

```
#define    WATCHDOG_IOCTL_BASE 'W'
#define    WDIOC_KEEPALIVE      _IOR(WATCHDOG_IOCTL_BASE, 5, int)
ioctl(wdt_fd, WDIOC_KEEPALIVE, 0);
```