SigmaStar Camera Watchdog User Guide

Version 0.1

REVISION HISTORY

Revision No.	Description	Date
0.1	• Initial release	12/18/2019

TABLE OF CONTENTS

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 = WDIOS_DISABLECARD;
ioctl(wdt_fd, WDIOC_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 WDIOC_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    WDIOC_SETTIMEOUT          _IOWR(WATCHDOG_IOCTL_BASE, 6, int)
int timeout = 20;
ioctl(wdt_fd, WDIOC_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);
```