Watchdog on Linux/Ubuntu

Background

Watchdog timers are commonly found in embedded systems and other computer-controlled equipment where humans cannot easily access the equipment or would be unable to react to faults in a timely manner. In such systems, the computer cannot depend on a human to reboot it if it hangs; it must be self-reliant.

ODROID-XU3/XU4 kernel support s3c2410_wdt module to control the Power Management Unit (PMU).

s3c2410_wdt driver is build as a loadable module so that watchdog daemon can configure the driver.

s3c2410_wdt could be loader with per-configurable parameters.

- tmr_margin - Watchdog tmr_margin in seconds.
- tmr_atboot - Watchdog is started at boot time if set to 1
- nowayout - Watchdog cannot be stopped once started
- soft_noboot - Watchdog action, set to 1 to ignore reboots

Note that the watchdog driver is available in the Kernel update 3.10.82 / 4.9.51 or higher.

```
odroid@odroid:~$ uname -a
Linux odroid 3.10.82-52 #1 SMP PREEMPT Thu Aug 27 11:45:33 BRT 2015 armv7l
armv7l armv7l GNU/Linux
```

Test Watchdog module

Watchdog driver s3c2410_wdt is configurable for Odroid XU3/XU4.

```
$ sudo modprobe s3c2410_wdt
```

You should be able to see /dev/watchdog and /dev/watchdog0 device files being created.

```
$ ls -l /dev/watchdog*
crw------- 1 root root  10, 130 Aug 28 09:57 /dev/watchdog
crw------- 1 root root 253,     Aug 28 09:57 /dev/watchdog0
```

Watchdog daemon will trigger and reboot if we access the device file manually.

```
$ sudo cat /dev/watchdog
[ 7639.726211] watchdog watchdog0: watchdog did not stop!
```

To manually stop watchdog to reboot.

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

Install Watchdog daemon

To install watchdog daemon

```
$ sudo apt-get install watchdog
```

Create dir for watchdog logs files

```
$ sudo mkdir -p /var/log/watchdog
```

Remove the watchdog module from black list. **/etc/modprobe.d/blacklist-watchdog.conf**

```
#blacklist s3c2410_wdt
```

Append the default watchdog configuration. **/etc/default/watchdog**

```
# Start watchdog at boot time? 0 or 1
run_watchdog=1
# Start wd_keepalive after stopping watchdog? 0 or 1
run_wd_keepalive=1
# Load module before starting watchdog
watchdog_module=s3c2410_wdt
# Specify additional watchdog options here (see manpage).
watchdog_options="-s -v -c /etc/watchdog.conf"
```

Watchdog demon configuration files

Note: Watchdog drivers start automatically as it's buildin, but only if a watchdog daemon to configure the times.

You need to edit the **/etc/watchdog.conf** file to un-comment and so actually use the **/dev/watchdog** device access to the module. Otherwise the watchdog will not use the hardware and rely only on its internal code to soft-reboot a broken machine

```
$ cat /etc/watchdog.conf
#ping                   = 172.31.14.1
#ping                   = 172.26.1.255
#interface              = eth0
file                    = /var/log/syslog
#change                 = 1407

# Uncomment to enable test. Setting one of these values to '0' disables it.
# These values will hopefully never reboot your machine during normal use
# (if your machine is really hung, the loadavg will go much higher than 25)
#max-load-1             = 24
#max-load-5             = 18
#max-load-15            = 12

# Note that this is the number of pages!
# To get the real size, check how large the pagesize is on your machine.
#min-memory             = 1
#allocatable-memory     = 1

#repair-binary          = /usr/sbin/repair
#repair-timeout         =
#test-binary            =
#test-timeout           =

watchdog-device = /dev/watchdog

# Defaults compiled into the binary
#temperature-device     =
#max-temperature        = 120

# Defaults compiled into the binary
admin                   = root
interval                = 1
logtick                = 1
log-dir                = /var/log/watchdog

# This greatly decreases the chance that watchdog won't be scheduled before
# your machine is really loaded
#realtime                = yes
#priority                = 1

# Check if rsyslogd is still running by enabling the following line
#pidfile                = /var/run/rsyslogd.pid

# set watchdog timer
watchdog-timeout        = 15

# set heartbeat setting
heartbeat-file = /var/log/watchdog/heartbeat.log
heartbeat-stamps = 300
```

For more configuration please follow link below.
http://www.sat.dundee.ac.uk/psc/watchdog/watchdog-configure.html

Start Watchdog Service and Verify

on Ubuntu 14.04.x enable watchdog service status

In order to start service we need to append /etc/rc.local service watchdog restart Watchdog service somehow doesn't start automatically. For now if the service doesn't start, it can be started with small HACK.

```
root@odroidxu4m:~# cat /etc/rc.local
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

service watchdog restart

exit
```

on Ubuntu 16.04.x enable watchdog service status

```
sudo systemctl enable watchdog
sudo systemctl start watchdog
```

Verify watchdog service in running correctly

```
root@odroidxu4m:~# service watchdog status
● watchdog.service - watchdog daemon
    Loaded: loaded (/lib/systemd/system/watchdog.service; static; vendor
preset: enabled)
    Active: active (running) since Fri 2015-08-28 10:48:41 UTC; 2s ago
    Process: 4736 ExecStart=/bin/sh -c [ $run_watchdog != 1 ] || exec
/usr/sbin/watchdog $watchdog_options (code=exited, status=/SUCCESS)
    Main PID: 4738 (watchdog)
    CGroup: /system.slice/watchdog.service
               └─4738 /usr/sbin/watchdog -s -v -c /etc/watchdog.conf

  Aug 28 10:48:41 odroidxu4m watchdog[4738]: hardware watchdog identity:
S3C2410 Watchdog
  Aug 28 10:48:41 odroidxu4m systemd[1]: Started watchdog daemon.
  Aug 28 10:48:41 odroidxu4m watchdog[4738]: current load is
  Aug 28 10:48:41 odroidxu4m watchdog[4738]: was able to ping process 2033
(/var/run/rsyslogd.pid).
  Aug 28 10:48:42 odroidxu4m watchdog[4738]: still alive after 1 interval(s)
  Aug 28 10:48:42 odroidxu4m watchdog[4738]: current load is
  Aug 28 10:48:42 odroidxu4m watchdog[4738]: was able to ping process 2033
```

```
(/var/run/rsyslogd.pid).
  Aug 28 10:48:43 odroidxu4m watchdog[4738]: still alive after 2 interval(s)
  Aug 28 10:48:43 odroidxu4m watchdog[4738]: current load is
  Aug 28 10:48:43 odroidxu4m watchdog[4738]: was able to ping process 2033
(/var/run/rsyslogd.pid).
```

Once the watchdog demon is configured it tries to continuously reset the watchdog timer. When/if it fails to do it (because of unresponsive system), the timer will expire and the board will reboot.

Another way to test watchdog device is killing the watchdog demon after it has started.

```
root@odroid64:~#
root@odroid64:~# pkill -9 watchdog
```