
SBC21/NSD21/EC21 Series

Watchdog Timer

Application Note



Copyright 2000 – 2013 IC Nexus Corporation. All rights reserved

Release Notes

Version	Release Date	Notes
1.00	August 19 th , 2013	Initial release
1.10	October 21 st , 2013	Correct some errors
1.20	November 12 th , 2013	Add Android
1.30	December 31 st , 2013	Modify u-boot args
1.40	January 7 th , 2014	Modify file name in Ch.4.1

Table of Contents

1. INTRODUCTION	4
2. SYSTEM REQUIREMENTS	4
3. SETTING UP WATCHDOG IN UBUNTU	5
3.1 WATCHDOG INSTALLATION	5
3.2 ENVIRONMENT SETUP	6
4. TESTING WATCHDOG IN UBUNTU	7
4.1 WATCHDOG DEMO PROGRAM	7
4.2 RUNNING THE WATCHDOG DEMO	8
5. SETTING UP WATCHDOG IN ANDROID	9
5.1 WATCHDOG SETTING	9

1. Introduction

A watchdog timer is an electronic timer that is used to detect and recover from computer malfunctions. It is sometimes called a computer operating properly, COP, or simply a watchdog. During normal operation, the computer regularly restarts the watchdog timer to prevent it from elapsing, or "timing out". If because of a hardware fault or program error the computer fails to restart the watchdog, the timer will elapse and generate a timeout signal. The timeout signal is used to initiate corrective actions. These actions typically include placing the computer system in a safe state and restoring normal system operation.

2. System requirements

- SBC21/NSD21/EC21 series products
- Operating System: Ubuntu 11.10 or above / Android 4.2
- Download the **watchdog.zip** from wiki page

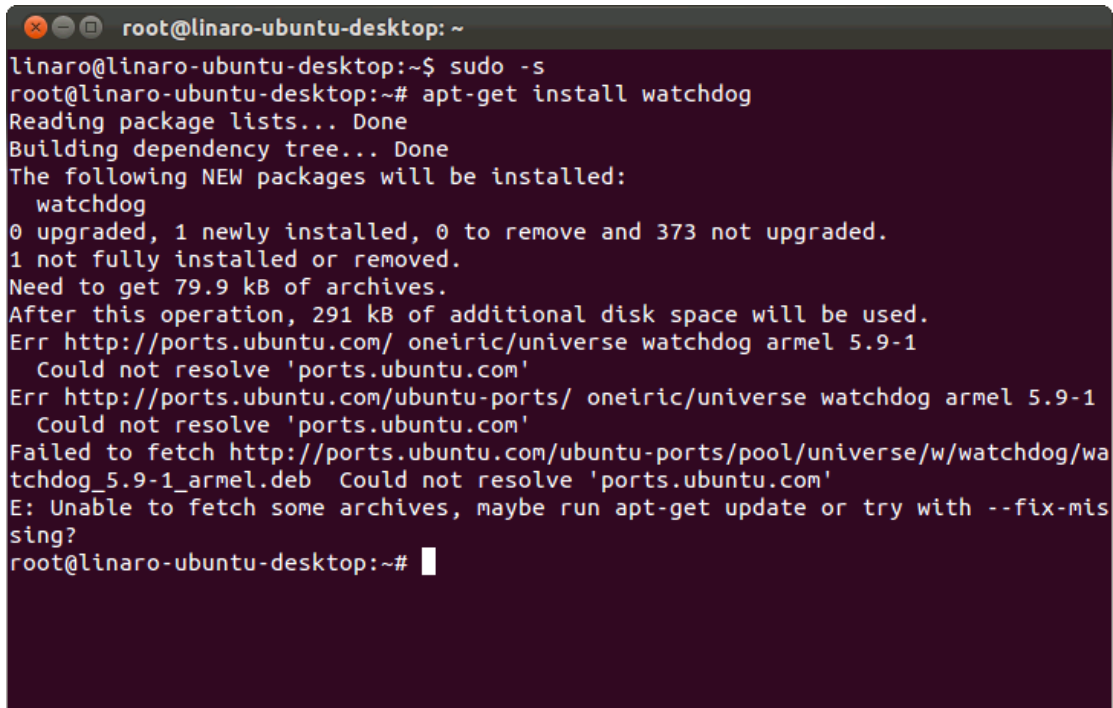
3. Setting Up Watchdog in Ubuntu

This section describes how watchdog can be installed on your Ubuntu 11.10 system.

3.1 Watchdog Installation

Follow these steps to install watchdog program on the Ubuntu system.

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

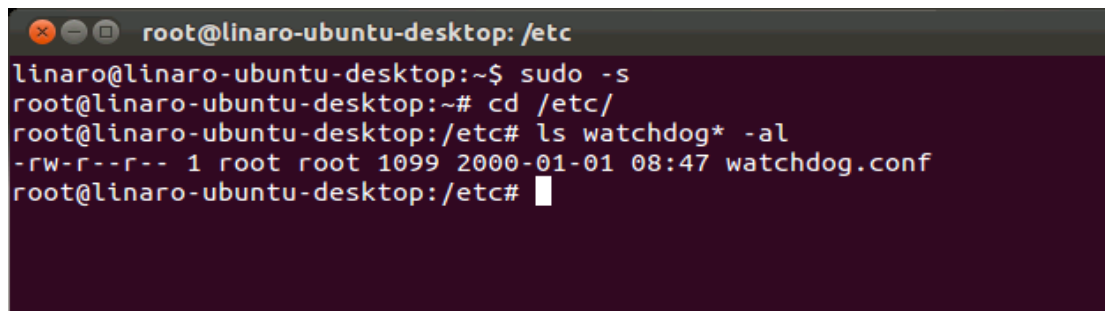


```
root@linaro-ubuntu-desktop: ~  
linaro@linaro-ubuntu-desktop:~$ sudo -s  
root@linaro-ubuntu-desktop:~# apt-get install watchdog  
Reading package lists... Done  
Building dependency tree... Done  
The following NEW packages will be installed:  
  watchdog  
0 upgraded, 1 newly installed, 0 to remove and 373 not upgraded.  
1 not fully installed or removed.  
Need to get 79.9 kB of archives.  
After this operation, 291 kB of additional disk space will be used.  
Err http://ports.ubuntu.com/ oneiric/universe watchdog armel 5.9-1  
  Could not resolve 'ports.ubuntu.com'  
Err http://ports.ubuntu.com/ubuntu-ports/ oneiric/universe watchdog armel 5.9-1  
  Could not resolve 'ports.ubuntu.com'  
Failed to fetch http://ports.ubuntu.com/ubuntu-ports/pool/universe/w/watchdog/wa  
tchdog_5.9-1_armel.deb  Could not resolve 'ports.ubuntu.com'  
E: Unable to fetch some archives, maybe run apt-get update or try with --fix-mis  
sing?  
root@linaro-ubuntu-desktop:~#
```

3.2 Environment Setup

Copy the watchdog configure file (watchdog.conf) to the /etc directory.

```
$ sudo -s  
# cp </your file path/watchdog.conf> /etc
```

A terminal window titled 'root@linaro-ubuntu-desktop: /etc' showing the execution of the setup commands. The user runs 'sudo -s' to become root, then 'cd /etc/' to change to the /etc directory, and finally 'ls watchdog* -al' to list the files. The output shows a file named 'watchdog.conf' with permissions '-rw-r--r--', owner 'root', group 'root', size '1099', and creation date '2000-01-01 08:47'.

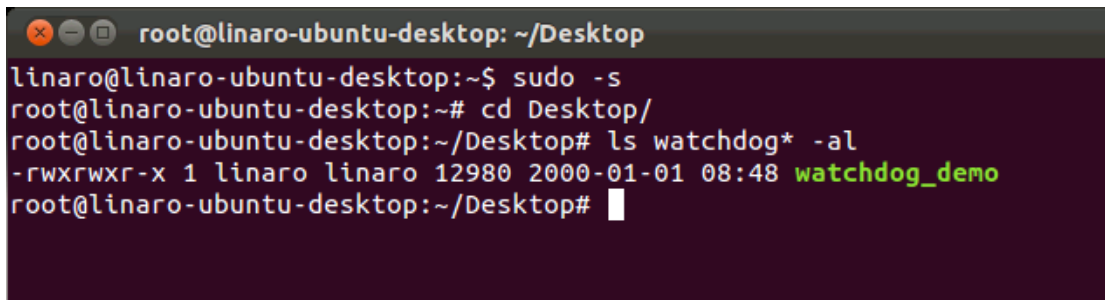
```
root@linaro-ubuntu-desktop: /etc  
linaro@linaro-ubuntu-desktop:~$ sudo -s  
root@linaro-ubuntu-desktop:~# cd /etc/  
root@linaro-ubuntu-desktop:/etc# ls watchdog* -al  
-rw-r--r-- 1 root root 1099 2000-01-01 08:47 watchdog.conf  
root@linaro-ubuntu-desktop:/etc#
```

4. Testing Watchdog in Ubuntu

4.1 Watchdog Demo Program

Copy the watchdog demo program (watchdog_demo) to ~/Desktop directory.

```
$ sudo -s  
# cp </your file path/watchdog_demo> ~/Desktop
```



```
root@linaro-ubuntu-desktop: ~/Desktop  
linaro@linaro-ubuntu-desktop:~$ sudo -s  
root@linaro-ubuntu-desktop:~# cd Desktop/  
root@linaro-ubuntu-desktop:~/Desktop# ls watchdog* -al  
-rwxrwxr-x 1 linaro linaro 12980 2000-01-01 08:48 watchdog_demo  
root@linaro-ubuntu-desktop:~/Desktop#
```

4.2 Running the Watchdog Demo

Run the watchdog demo program. **watchdog_demo -i <seconds>**

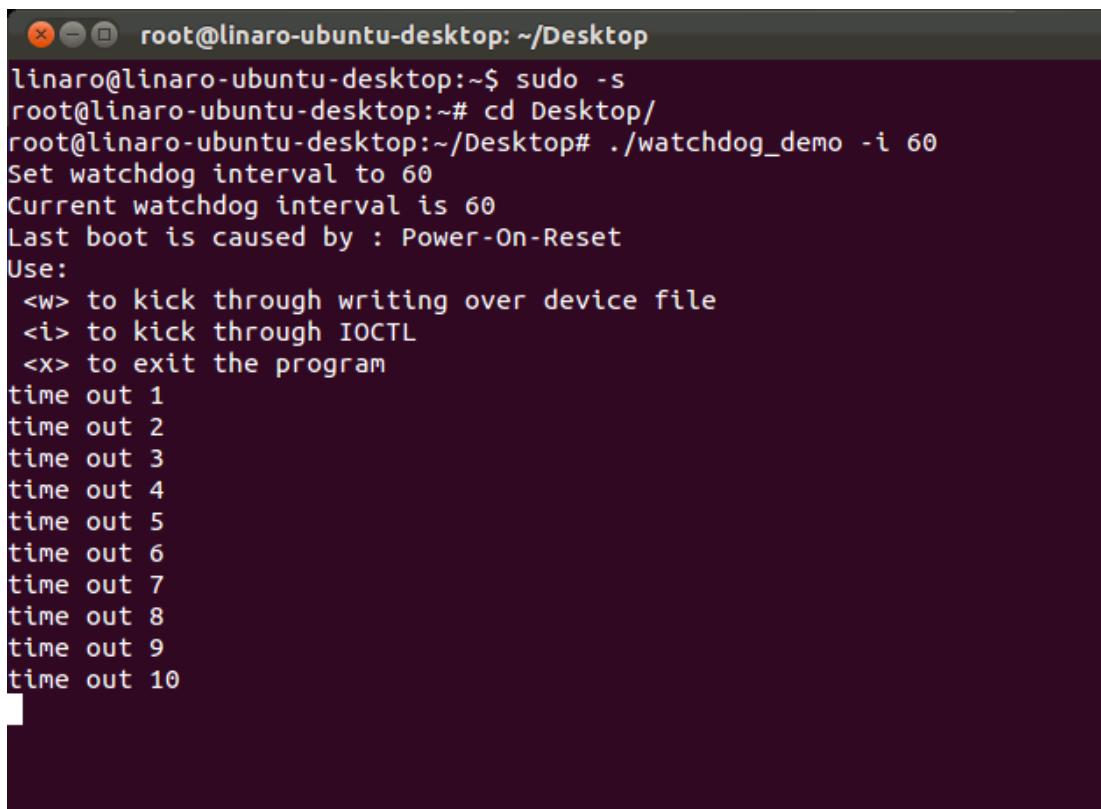
```
$ sudo -s
# cd ~/Desktop
# ./watchdog_demo -i 60
```

If watchdog has been installed and setup successfully, then the system will restart after 60 seconds. If it does not restart successfully, then uninstall the watchdog program and reinstall it again.

Press the <w> or <i> buttons to reset the timer.

<Notes>

- <w> to kick through writing over device file
- <i> to kick through IOCTL (input output control)
- <x> to exit the program



```
root@linaro-ubuntu-desktop: ~/Desktop
linaro@linaro-ubuntu-desktop:~$ sudo -s
root@linaro-ubuntu-desktop:~# cd Desktop/
root@linaro-ubuntu-desktop:~/Desktop# ./watchdog_demo -i 60
Set watchdog interval to 60
Current watchdog interval is 60
Last boot is caused by : Power-On-Reset
Use:
<w> to kick through writing over device file
<i> to kick through IOCTL
<x> to exit the program
time out 1
time out 2
time out 3
time out 4
time out 5
time out 6
time out 7
time out 8
time out 9
time out 10
```


5. Setting Up Watchdog in Android

This section describes how to set the watchdog timer on your Android 4.2 system.

5.1 Watchdog Setting

Watchdog timer setting is in U-Boot environment. Please enter into the U-Boot environment first and find out the argument **bootargs=<string value>**.

```
Hit any key to stop autoboot: 3
```

```
// quickly hit the Enter key
```

```
> printenv
```

```
...
```

```
bootargs=arm_freq=1000
```

```
...
```

bootargs – the option argument. Your *string value* may different from mine.

Please add following argument at the end of bootargs value: **wdt=x:y**

x – to be time (seconds) of restarting a watchdog timer (clear watchdog timer)

y – to be watchdog timeout time (seconds)

```
> setenv bootargs '<your string value> wdt=x,y'
```

For example, set restarting cycle time to be 10 seconds and timeout time to be 20 seconds.

```
> setenv bootargs 'arm_freq=1000 wdt=10,20'
```

```
> saveenv
```

```
Saving Environment to SPI Flash...
```

```
Erasing SPI flash...Writing to SPI flash.....SUCCESS
```

```
done
```

If you set the value that **x > y**, then the system would reboot in **y** seconds after power on.

If you set the value that **x=y=0**, then watchdog timer would be disable, like following:

```
> setenv bootargs '<your string value> wdt=0,0'
```