GM8136

WATCHDOG

User Guide
Rev.: 1.0
Issue Date: September 2014

REVISION HISTORY

GM8136 Watchdog User Guide

Date	Rev.	From	To
Sept. 2014	1.0	-	Original

Printed in Taiwan 2014

Grain Media, Inc.
5F, No. 5, Li-Hsin Road III, Hsinchu Science Park, Hsinchu City, Taiwan 300, R.O.C.

Grain Media's home page can be found at:
http://www.grain-media.com

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

Watchdog Configuration

This chapter contains the following sections:

1.1 Watchdog Overview

Watchdog timer is a mechanism to prevent the system from hanging forever due to some unexpected conditions. Its operation is easy to understand. Once users start a watchdog timer and set a countdown number into the register, it would begin to decrease the counter. When the counter reaches zero, the system will restart. The only way to stop that is to kick the watchdog before it reaches zero. Every time when users kick the watchdog, the value of the counter would be the value set at the first time.

GM8136 provides a watchdog IP for this job. Users can simply get the function by following the descriptions of the following sections.

1.2 Build Watchdog Module and Use it

The Grain Media hardware watchdog is FTWDT010. The watchdog module is built-in in the Linux kernel image. Users can use menu configuration (make menuconfig) to choose items for FTWDT010.
In Figure 1-1, it shows the items which should be chosen in menu configuration of Linux kernel.
The steps are Device Drivers ---> Watchdog Timer Support ---> FTWDT010 watchdog

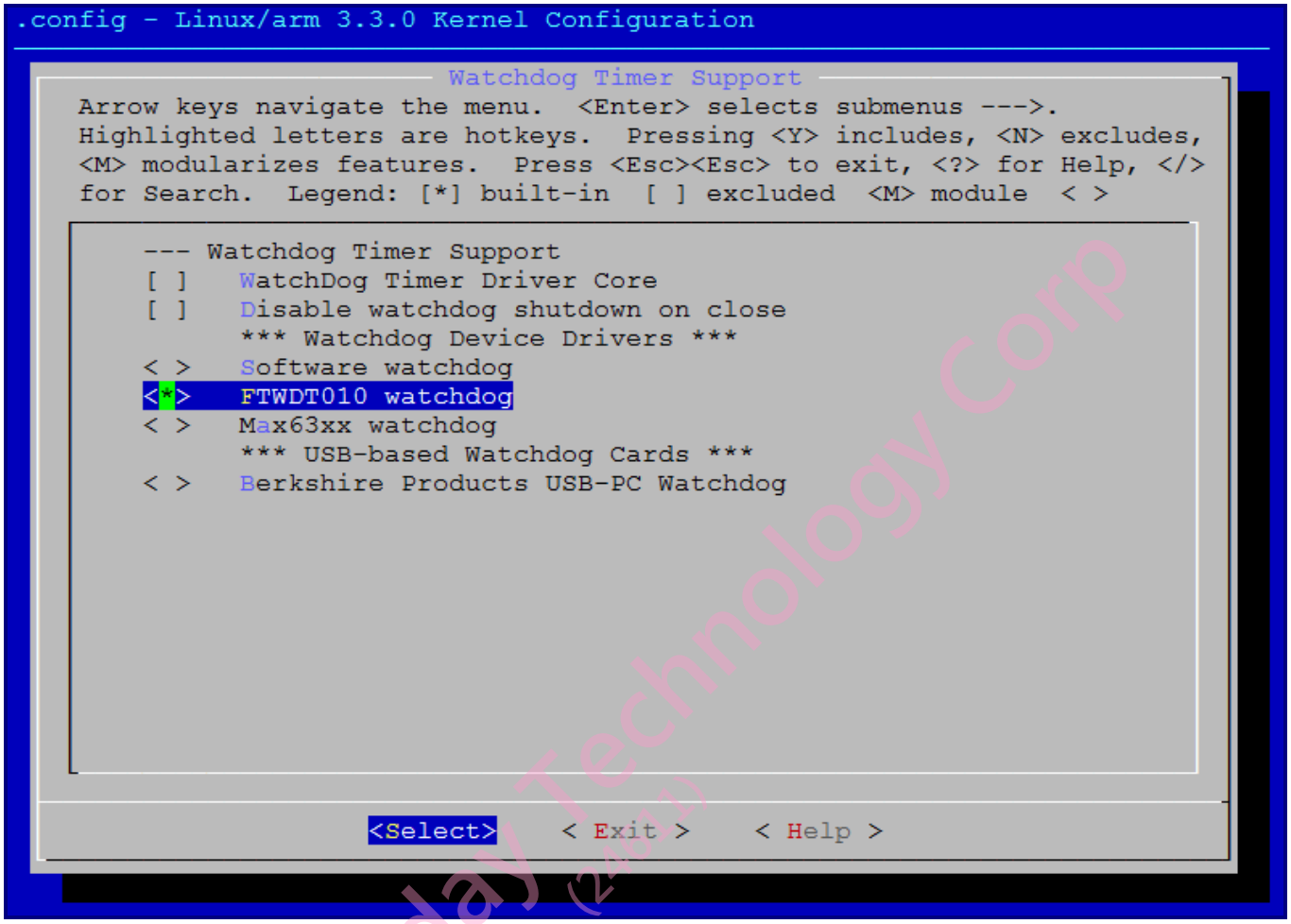

Figure 1-1. Build FTWDT010

After building the watchdog module in the kernel image, users can boot Linux and find the device node of FTWDT010 whose path is "/dev/watchdog". In Figure 1-2, users can find the device node to insert the watchdog driver into kernel.

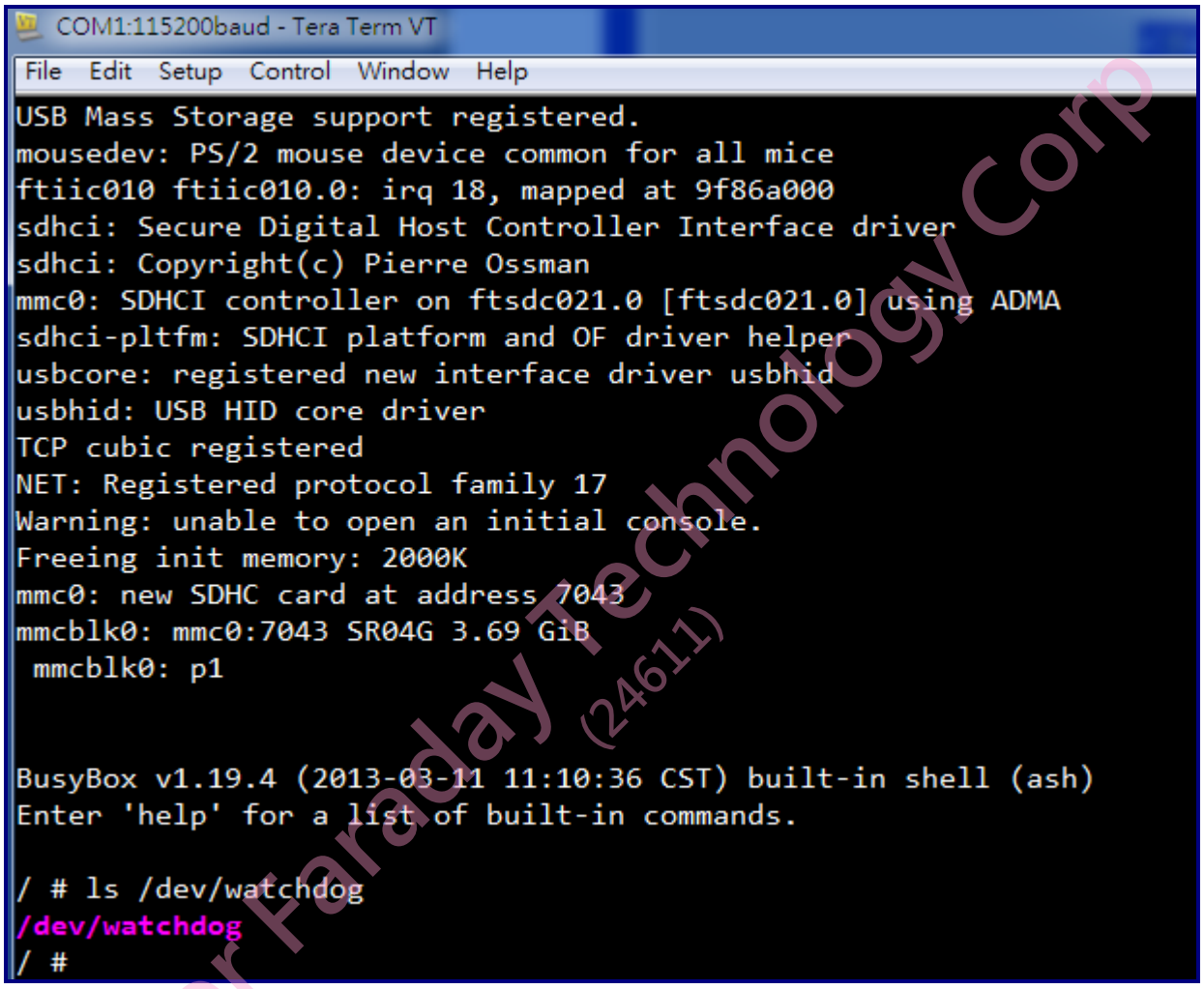

Figure 1-2. Insert Watchdog Driver into Kernel

Now, users can write the sample codes to use the watchdog module. Figure 1-3 is the sample code for triggering watchdog for reboot.

```c
/* standard C headers */
#include <stdlib.h>
#include <stdio.h>

/* POSIX's headers */
#include <fcntl.h>
#include <string.h>
#include <unistd.h>
#include <sys/time.h>
#include <sys/ioctl.h>

/* for Linux watchdog ioctl */
#include <linux/watchdog.h>

#define WDT_DEVICE_FILE         "/dev/watchdog"

int main() {
    int sec = -1, fd = -1, timeout = 0;

    fd = open(WDT_DEVICE_FILE, O_RDWR);
    if (!fd) {
        perror("open WDT device");
        return -1;
    }

    ioctl(fd, WDIOC_GETTIMEOUT, &timeout);
    printf("default timeout %d sec.\n", timeout);

    printf("We reset timeout as 20 sec.\n");
    timeout = 20;
    ioctl(fd, WDIOC_SETTIMEOUT, &timeout);
    ioctl(fd, WDIOC_GETTIMEOUT, &timeout);
    if (timeout != 20) {
        printf("WDT timeout reset error.\n");
        return -1;
    }

    printf("system reboot after 20 sec.\n");
    while (1) {
        ioctl(fd, WDIOC_KEEPALIVE, 0);
        sleep(1);
        printf("keep alive!!\n");
    }
}
```


Figure 1-3. Sample Code for Triggering Watchdog for Reboot

Currently, the watchdog restart system is set to five seconds if it is not kicked.