Test Plan for SRTC

Test Plan for SRTC

1 Outline

This document is for the SRTC driver in Linux kernel of MVF TOWER BOARD (XTWR-VF600) with VF6XX SoC, and describes test plan for each API/feature of such unit.

2 Test Environment

Toolchain: The latest Linaro toolchain
Bootloader: u-boot 2011.12
Kernel: Freescale i.MX Linux 3.0.15 kernel
Rootfs: rootfs on NFS

3 Target Module of the Test

SRTC Driver

4 Test Plan

Use the mxc_rtc test in imx-test-12.03.00 package.

5 Conditions

When assigning CortexA5 Global Timer to kernel timer, SRTC interrupt number is 132. Therefore, change RTC_IRQS_EXPECTED value of autorun-rtc.sh to 132 as well.

Test Plan for SRTC

Datails

No.	Head	Item	Procedure	Points to be checked	Judge	Note
1	SRTC Test	Integrative test	Execute following commands on the target's console. # cd /unit_tests # ./autorun-rtc.sh	Update interrupt occurs every 1 second. Alarm interrupt occurs as it is set for 5 seconds later.	OK	**Sample LOG:** root@freescale /unit_tests$ ./autorun-rtc.sh MVFTWRVF600Board Checking for devnode: /dev/rtc0 autorun-rtc.sh: PASS devnode found: /dev/rtc0 Running test case: ./rtctest.out --no-periodic RTC Driver Test Example. Counting 5 update (1/sec) interrupts from reading /dev/rtc0: 1 2 3 4 5 Again, from using select(2) on /dev/rtc0: 1 2 3 4 5 Current RTC date/time is 1-1-1970, 00:01:54. Alarm time now set to 00:01:59. Waiting 5 seconds for alarm... okay. Alarm rang. *** Test complete *** Typing "cat /proc/interrupts" will show 1 more events on IRQ rtc. autorun-rtc.sh: PASS test case: ./rtctest.out --no-periodic rtc irqs before running unit test: 0 rtc irqs after running unit test: 11 so rtc irqs during test was: 11 checking rtc interrupts PASS autorun-rtc.sh: Exiting PASS root@freescale /unit_tests$

2		Comparison with actual time	Execute following command on the target's console. # hwclock --systohc After an interval, execute the following command and compare system clock with rtc. # date; cat /proc/driver/rtc	No significant difference between system clock and rtc.	NG	Time delay at a rate of one second in 10 seconds or so. A cause is going to be investigated at a later date. **LOG:** # hwclock --systohc; date; cat /proc/driver/rtc; sleep 60; date; cat /proc/driver/rtc Thu Jan 1 00:07:33 1970 rtc_time : 00:07:33 rtc_date : 1970-01-01 alrm_time : 00:00:00 alrm_date : 1970-01-01 alarm_IRQ : no alrm_pending : no update IRQ enabled : no periodic IRQ enabled : no periodic IRQ frequency : 1 max user IRQ frequency : 64 24hr : yes alarm_IRQ : no Thu Jan 1 00:08:33 1970 rtc_time : 00:08:27 rtc_date : 1970-01-01 alrm_time : 00:00:00 alrm_date : 1970-01-01 alarm_IRQ : no alrm_pending : no update IRQ enabled : no periodic IRQ enabled : no periodic IRQ frequency : 1 max user IRQ frequency : 64 24hr : yes alarm_IRQ : no