Test Plan for LPTMR

Test Plan for LPTMR

1 Outline

This document is for the LPTMR (Low-Power Timer) 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

LPTMR Driver

4 Test Plan

Create testing driver and use it for the test since LPTMR driver does not have the operational interface for application.

5 Testing Method

1. Preparation

Have the following setting in kernel configuration ON.
Character devices --->
[*] Low Power Timer Module support

Copy test_program/mvf_testmodule.c to drivers/char/.
そThen add the following to the end of drivers/char/Makefile.
obj-y += mvf_testmodule.o

2. Test of each timer

Test runs automatically as booting the kernel built by #1 above.

Test Plan for LPTMR

Datails

No.	Head	Item	Procedure	Points to be checked	Judge	Note
1		Timer allocation	Call lpt_alloc_timer function via testing driver.	Timer handle is obtained.	OK	
2		Timer start	Continue from the test above. Call lpt_enable_timer via testing driver.	Negative value is returned and an error occurs (since setting by lpt_param_set function is not done.)	OK	
3			Continue from the test above. Call lpt_param_set function via testing driver and set value, then call lpt_enable_timer function.	Successful timer start is returned.	OK	
4	Interrupt	Periodic event	Continue from the test 1 above. Call lpt_param_set function via testing driver and set maximum counter value of event handler and timer, then call lpt_enable_timer function.	Event handler is called for each specified count.	OK	
5		Change in timer value	Continue from the test 1 above. Call lpt_param_set function via testing driver and change the maximum value of timer to half of the #4 above, then call lpt_enable_timer function.	Event handler is called for each specified count. Event occurs twice more often than the operation of #4 above.	OK	
6		Change in frequency division	Continue from the test 1 above. Call lpt_param_set function via testing driver. Set mode: timer counter mode and bypass:LPT_PARAM_PB_GF_BYPASS and call lpt_enable_timer function. Change the value of prs_value and recall lpt_enable_timer function.	Event handler is called for each specified count. Frequency of event occurrence changes as the value of prs_value is changed.	OK	
7	Output Test	Timer read	Continue from the test above. Call lpt_read_counter function via testing driver.	Have the return value of 0 and have the timer value for pointer.	OK	
8		Timer stop	Continue from the test above. Call lpt_disable_timer function via testing driver.	Successful timer stop is returned. No event handler call (set at #4 above) occurs after stopping.	OK	
9		Timer release	Continue from the test above. Call lpt_free_timer function via testing driver.	Successful timer release is returned.	OK	