ochin_CM4v2 Hardware test number 7

USBs stress test

Devices used for tests

1. ochin CM4 carrier board
2. Raspberry Pi CM4 module with eMMC
3. Power Supply 0-30Vdc
4. Raspberry Pi Zero configured as Gadget Zero Device

Test description

This test is intended to verify the operation of the USB interfaces for long sessions of reads and writes in different modes. The goal is to verify the robustness of the system, testing both the integrated components involved, the connections and couplings between them.

The program used to test the USBs is the "testusb.c" file, already present and compiled in linux OS and designed for the purpose of testing USB device drivers.

To perform the tests, a device capable of responding to the SW must be connected to the desired USB port. For this purpose, I used a Raspberry Pi Zero configured in Device Gadget Zero mode.

For the sake of the test, the /dev/zero source will be used since it's a device file which generates a continuous stream of zeros. It could be also used as destination path, in which case it will accept ad discards all data written to it.

Test execution

To configure a Raspberry Pi in Device Gadget Zero mode, simply add the following command at the end of the "cmdline.txt" file:

```
… modules-load=dwc2,g_zero
```

From http://www.linux-usb.org/gadget/

***Gadget Zero** … is essential for controller driver testing. It provides two configurations: one for bulk traffic source/sink, and another for bulk traffic loopback. On the* device side *it helps verify that the driver stack pass USB-IF and other tests (needed for at least USB branding). On the* host side *it helps test the USB stack, such as the Linux-USB HCDs and usbcore. If built as a module, start it by* modprobe g_zero*; no other initialization is needed.*

The raspberry pi zero is connected to the USB ports on the ochin board, which are tested one at a time. For the repetitive execution of the "testusb.c" software, the "test.sh" script was used, which runs the test in the following modes indefinitely.

```
usbtest 2-2.4:3.0: TEST 0:  NOP
usbtest 2-2.4:3.0: TEST 1:  write 512 bytes 1000 times
usbtest 2-2.4:3.0: TEST 2:  read 512 bytes 1000 times
usbtest 2-2.4:3.0: TEST 3:  write/512 0..512 bytes 1000 times
usbtest 2-2.4:3.0: TEST 4:  read/512 0..512 bytes 1000 times
usbtest 2-2.4:3.0: TEST 5:  write 1000 sglists 32 entries of 512 bytes
usbtest 2-2.4:3.0: TEST 6:  read 1000 sglists 32 entries of 512 bytes
usbtest 2-2.4:3.0: TEST 7:  write/512 1000 sglists 32 entries 0..512 bytes
usbtest 2-2.4:3.0: TEST 8:  read/512 1000 sglists 32 entries 0..512 bytes
usbtest 2-2.4:3.0: TEST 9:  ch9 (subset) control tests, 1000 times
usbtest 2-2.4:3.0: TEST 10:  queue 32 control calls, 1000 times
usbtest 2-2.4:3.0: TEST 14:  1000 ep0out, 0..255 vary 1
```

For this test, we have chosen to stop the execution after 4 hours for each USB port.

For more information about the USB Testing on Linux, please follow this link .

Test result

The test worked as expected and took 16 hours in total. The log file related to the tests done on each port could be found in the "ochin_CM4_USBstresstest_MMYYYY.zip" file.

Test Passed