ochin_CM4 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
5. 1x usb type-c cable
6. 1x GHS 4pin to usb Type A cable

Test description

The Raspberry Pi CM4 module provides only one USB port, which is used both to flash the eMMC of the cm4 and for the normal use of the board to connect devices.

The ochin board has a 4ports USB-HUB, connected to the CM4 module, in order to increase to 4 the number of the USB ports.

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 carry out the test, another hardware device must be connected to the affected USB port. This device must contain software capable of answering calls on the USB port for the test to be successful. In our case we use a Raspberry Pi 4 or a Raspberry Pi Zero in a mode called "Gadget Device 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.*

It is important to use an RPi4 or an RPi Zero because the USB must also work in device mode (it is not the case for the other Raspberry models). 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.

Preliminary Configuration

Prepare the Raspberry Compute Module 4 + ochin board:

1: flash the Raspberry Pi OS Lite Debian version 10(buster) to the sd (link: https://downloads.raspberrypi.org/raspios_oldstable_lite_armhf/images/raspios_oldstable_lite_armhf-2022-04-07/2022-04-04-raspios-buster-armhf-lite.img.xz)

I suggest to use the Raspberry Pi imager software, in order to also configure the WiFi network and enable the ssh.

2: enable the USB interface (disabled by default on CM4). Edit the file "/boot/config.txt" and add

```
dtoverlay=dwc2,dr_mode=host
```

at the end of the file

3: copy the content of the testusb.zip file in the home folder

4: install "screen" with "sudo apt-get install screen" , this is needed to keep the test running even if you disconnect from terminal

Prepare the Raspberry Pi 4 to function as a device:

1: flash the Raspberry Pi OS Lite Debian version 10(buster) to the sd (link: https://downloads.raspberrypi.org/raspios_oldstable_lite_armhf/images/raspios_oldstable_lite_armhf-2022-04-07/2022-04-04-raspios-buster-armhf-lite.img.xz)

I suggest to use the Raspberry Pi imager software, in order to also configure the WiFi network and enable the ssh.

2: edit the file “/boot/cmdline.txt” and add

modules-load=dwc2,g_zero

at the end of the first line

3: edit the file “/boot/config.txt” and add

dtoverlay=dwc2

at the end of the file

At this point the Raspberry Pi 4 is ready to work as a device. For this to happen it is necessary to connect the USB Type-C connector of the RPi4 to one of the 4 USB ports of the ochin board.

At boot the Raspberry Pi 4 will realize that it is connected to a USB port (and not to a power supply as usual) and will enter "Gadget Device Zero" mode.

To do so, you will need a USB Type-C data cable and a GHS 4Pin <-> USB Type-A (female) adapter cable.

To build the GHS USB adapter cable please look at the "öchìn CM4 - Wiring and Suggestions.pdf" doc, page 3/5.

Test execution

NOW YOU'RE READY TO START THE TEST

1: connect the RPi4 to the first USB port of the ochin board (using the adapter cables)

2: power on the ochin board

3: connect to the CM4 via ssh

4: sending the "lsusb" command you should see the list of the usb interfaces and the "Gadget Device Zero" connected

5: use screen and the test2file.sh script to start the test logging the result to a file and keep it running after the terminal disconnects (this is for long runs)

screen sudo ./test2file.sh USB3_test1_ochin_CM4.txt

not the test is running in the background...

to see the active screen process:

screen -list

There is a screen on:

2410.pts-3.pi (19/02/22 16:10:26) (Attached)

1 Socket in /run/screen/S-pi.

to stop the screen process

screen -X -S "2410.pts-3.pi" quit

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