



Coprocessor management troubleshooting grid



Contents



A quality version of this page, approved on 7 December 2020, was based off this revision.

Some typical issues related to the management of a coprocessor are listed below. Solutions or debugging methods are proposed for these issues.

If your issue is not listed, try looking in the articles in the Coprocessor management Linux, Coprocessor management STM32Cube or troubleshooting grids categories.



1 Coprocessor firmware loading and control

Symptom	Resolution
<p>The coprocessor traces are not available on Linux® side</p> <pre>Board \$> cat /sys/kernel/debug /remoteproc/remoteproc0/trace0 No such file or directory</pre>	<p>This may happen for two reasons:</p> <ul style="list-style-type: none"> The firmware does not include any resource table or the resource table does not define any trace. <p>Update the firmware resource table and rebuild the firmware</p> <ul style="list-style-type: none"> The ".resource_table" section is empty or not defined in the elf file. <p>Use the following command to verify it in the generated elf file:</p> <pre>PC \$> readelf -l <elf file> ... 02 .data . resource_table .bss . _user_heap_stack</pre>
<p>When starting the coprocessor from the bootloader (u-boot):</p> <pre>"unsupported fw ver: 0 Remote Processor 0 resource table Not found : 0x00000000-0x0 "</pre>	<p>The firmware does not include any resource table. This is only a warning that does not prevent the firmware from starting properly. "rproc load_rsc" step can be bypassed.</p>



2 Inter processor communication

Symptom	Resolution
<p>Frozen firmware as consequence of a deadlock in OpenAMP during IP communication with the the main processor.</p>	<p>This Issue probably comes from <code>rpmsg_virtio_rx_callback</code> or <code>rpmsg_virtio_send_offchannel_raw</code> (<code>rpmsg_virtio.c</code>) functions that are called in interrupt context. These functions use a mutex lock in <code>rpmsg_device</code> struct when accessing the index of the virtio queue index. Rework your code so that these functions are not called in interrupt context.</p>
<p>Linux kernel trace:</p> <pre>stm32-ipcc 4c001000.mailbox: Try increasing MBOX_TX_QUEUE_LEN</pre>	<p>On each IPCC interrupt, the coprocessor treats all the buffered RPMsgs (one IPCC event for several RPMsgs). On Linux side, one IPCC signal is programmed for each RPMsg sent. This can result is an overflow warning on Linux since too many IPCC events are queued. No RPMsgs are dropped but this message could be interpreted as the coprocessor reaches its capacity to treat the received messages in time. Consider reworking the code so that the coprocessor processes messages more efficiently or decreasing the rate of messages sent by Linux.</p>
<p>Linux kernel trace (example with <code>rpmsg_tty</code> driver):</p> <pre>rpmsg_tty virtio0.rpmsg-tty-channel. -1.0: timeout waiting for a tx buffer</pre>	<p>This message means that there is no more TX buffer available to transmit messages to the remote processor. This may happen for two reasons:</p> <ul style="list-style-type: none"> • The firmware implementation is incomplete and the coprocessor does not process any IPCC interrupt (eg interrupt disabled or interrupt handler not defined). Fix the firmware code: refer to <code>IPCC_internal_peripheral</code> for details on the peripheral. • The coprocessor is busy, frozen or crashed. This can be confirmed by performing a debug tool analysis.
<p>Linux kernel trace:</p>	<p>There is no more space in the TTY temporary buffer on reception. The root cause is probably that the Linux application did not</p>



Symptom	Resolution
<pre>rpmsg_tty virtio0.rpmsg-tty-channel. -1.0: No memory for tty_prepare_flip_string</pre>	<p>read the tty device on time to treat the incoming TTY stream. Consider reworking the Linux application so that it takes less time to process messages or decreasing the message exchange rate.</p>
<p>Linux kernel trace:</p> <pre>remoteproc remoteproc0: stm32_rproc_kick: failed (<mbx>, <error value>)</pre>	<p>The Linux remoteproc driver cannot use the IPCC mailbox. This may happen for two reasons:</p> <ul style="list-style-type: none"> • The Linux kernel is built without the support of the stm32 IPCC mailbox: to enable it, refer to IPCC configuration. • The mailboxes are not or incorrectly defined in the Linux kernel DeviceTree: fix it as described in mailbox DeviceTree.
<p>stm32Cube error: VIRT_UART_Transmit() returns an error until a first message is received from the Cortex-A7.</p>	<p>Before the Cortex-M4 can send any message, the RMPmsg protocol requires that the Linux application has sent a first message (to provide its address to the Cortex-M4).</p> <p>The Linux and/or the Cortex-M4 applications shall be reworked to follow this constraint.</p>

Linux® is a registered trademark of Linus Torvalds.

Transmit

Inter-Processor Communication Controller

Remote Processor Messaging

TeleTYpewriter

Universal Asynchronous Receiver/Transmitter

Cortex®