

RapidIO subsystem Channelized Messaging character device driver (rio_cm.c)

1. Overview

This device driver is the result of collaboration within the RapidIO.org Software Task Group (STG) between Texas Instruments, Prodrive Technologies, Nokia Networks, BAE and IDT. Additional input was received from other members of RapidIO.org.

The objective was to create a character mode driver interface which exposes messaging capabilities of RapidIO endpoint devices (mports) directly to applications, in a manner that allows the numerous and varied RapidIO implementations to interoperate.

This driver (RIO_CM) provides to user-space applications shared access to RapidIO mailbox messaging resources.

RapidIO specification (Part 2) defines that endpoint devices may have up to four messaging mailboxes in case of multi-packet message (up to 4KB) and up to 64 mailboxes if single-packet messages (up to 256 B) are used. In addition to protocol definition limitations, a particular hardware implementation can have reduced number of messaging mailboxes. RapidIO aware applications must therefore share the messaging resources of a RapidIO endpoint.

Main purpose of this device driver is to provide RapidIO mailbox messaging capability to large number of user-space processes by introducing socket-like operations using a single messaging mailbox. This allows applications to use the limited RapidIO messaging hardware resources efficiently.

Most of device driver's operations are supported through 'ioctl' system calls.

When loaded this device driver creates a single file system node named `rio_cm` in `/dev` directory common for all registered RapidIO mport devices.

Following ioctl commands are available to user-space applications:

- **RIO_CM_MPORT_GET_LIST:**
Returns to caller list of local mport devices that support messaging operations (number of entries up to `RIO_MAX_MPORTS`). Each list entry is combination of mport's index in the system and RapidIO destination ID assigned to the port.
- **RIO_CM_EP_GET_LIST_SIZE:**
Returns number of messaging capable remote endpoints in a RapidIO network associated with the specified mport device.
- **RIO_CM_EP_GET_LIST:**
Returns list of RapidIO destination IDs for messaging capable remote endpoints (peers) available in a RapidIO network associated with the specified mport device.
- **RIO_CM_CHAN_CREATE:**
Creates RapidIO message exchange channel data structure with channel ID assigned automatically or as requested by a caller.
- **RIO_CM_CHAN_BIND:**
Binds the specified channel data structure to the specified mport device.
- **RIO_CM_CHAN_LISTEN:**
Enables listening for connection requests on the specified channel.
- **RIO_CM_CHAN_ACCEPT:**
Accepts a connection request from peer on the specified channel. If wait timeout for this request is specified by a caller it is a blocking call. If timeout set to 0 this is non-blocking call - ioctl handler checks for a pending connection request and if one is not available exits with `-EGAIN` error status immediately.
- **RIO_CM_CHAN_CONNECT:**
Sends a connection request to a remote peer/channel.
- **RIO_CM_CHAN_SEND:**
Sends a data message through the specified channel. The handler for this request assumes that message buffer specified by a caller includes the reserved space for a packet header required by this driver.
- **RIO_CM_CHAN_RECEIVE:**
Receives a data message through a connected channel. If the channel does not have an incoming message ready to return this ioctl handler will wait for new message until timeout specified by a caller expires. If timeout value is set to 0, ioctl handler uses a default value defined by `MAX_SCHEDULE_TIMEOUT`.
- **RIO_CM_CHAN_CLOSE:**
Closes a specified channel and frees associated buffers. If the specified channel is in the `CONNECTED` state, sends close notification to the remote peer.

The ioctl command codes and corresponding data structures intended for use by user-space applications are defined in 'include/uapi/linux/rio_cm_cdev.h'.

2. Hardware Compatibility

This device driver uses standard interfaces defined by kernel RapidIO subsystem and therefore it can be used with any mport device driver registered by RapidIO subsystem with limitations set by available mport HW implementation of messaging mailboxes.

3. Module parameters

- 'dbg_level'
 - This parameter allows to control amount of debug information generated by this device driver. This parameter is formed by set of bit masks that correspond to the specific functional block. For mask definitions see 'drivers/rapidio/devices/rio_cm.c' This parameter can be changed dynamically. Use CONFIG_RAPIDIO_DEBUG=y to enable debug output at the top level.
- 'cmbox'
 - Number of RapidIO mailbox to use (default value is 1). This parameter allows to set messaging mailbox number that will be used within entire RapidIO network. It can be used when default mailbox is used by other device drivers or is not supported by some nodes in the RapidIO network.
- 'chstart'
 - Start channel number for dynamic assignment. Default value - 256. Allows to exclude channel numbers below this parameter from dynamic allocation to avoid conflicts with software components that use reserved predefined channel numbers.

4. Known problems

None.

5. User-space Applications and API Library

Messaging API library and applications that use this device driver are available from RapidIO.org.

6. TODO List

- Add support for system notification messages (reserved channel 0).