Using Linux CAIF

Author: Sjur Brendeland/ sjur.brandeland@stericsson.com

Start

If you have compiled CAIF for modules do:

```
$modprobe crc_ccitt
$modprobe caif
$modprobe caif_socket
$modprobe chnl_net
```

Preparing the setup with a STE modem

If you are working on integration of CAIF you should make sure that the kernel is built with module support.

There are some things that need to be tweaked to get the host TTY correctly set up to talk to the modem. Since the CAIF stack is running in the kernel and we want to use the existing TTY, we are installing our physical serial driver as a line discipline above the TTY device.

To achieve this we need to install the N_CAIF ldisc from user space. The benefit is that we can hook up to any TTY.

The use of Start-of-frame-extension (STX) must also be set as module parameter "ser_use_stx".

Normally Frame Checksum is always used on UART, but this is also provided as a module parameter "ser_use_fcs".

```
$ modprobe caif_serial ser_ttyname=/dev/ttyS0 ser_use_stx=yes
$ ifconfig caif_ttyS0 up
```

PLEASE NOTE:
There is a limitation in Android shell. It only accepts one argument to insmod/modprobe!

Trouble shooting

There are debugfs parameters provided for serial communication. /sys/kernel/debug/caif_serial/<tty-name>/

- ser_state: Prints the bit-mask status where
  - 0x02 means SENDING, this is a transient state.
  - 0x10 means FLOW_OFF_SENT, i.e. the previous frame has not been sent and is blocking further send operation. Flow OFF has been propagated to all CAIF Channels using this TTY.
- tty_status: Prints the bit-mask tty status information
  - 0x01 - tty->warned is on.
  - 0x04 - tty->packed is on.
  - 0x08 - tty->flow.tco_stopped is on.
  - 0x10 - tty->hw_stopped is on.
  - 0x20 - tty->flow.stopped is on.
- last_tx_msg: Binary blob Prints the last transmitted frame.

  This can be printed with:

  ```
  $od --format=x1 /sys/kernel/debug/caif_serial/<tty>/last_rx_msg.
  ```

  The first two tx messages sent look like this. Note: The initial byte 02 is start of frame extension (STX) used for re-syncing upon errors.
  - Enumeration:

    ```
    0000000  02 05 00 00 03 01 d2 02
             |  |     |  |  |  |
             STX(1)   |  |  |  |
                Length(2)|  |  |
                      Control Channel(1)
                         Command:Enumeration(1)
                            Link-ID(1)
                                Checksum(2)
    ```

  - Channel Setup:

    ```
    0000000  02 07 00 00 00 21 a1 00 48 df
    ```

```
|  |     |  |  |  |  |  |
STX(1)   |  |  |  |  |  |
   Length(2)|  |  |  |  |
         Control Channel(1)
            Command:Channel Setup(1)
               Channel Type(1)
                   Priority and Link-ID(1)
                     Endpoint(1)
                         Checksum(2)
```

- last_rx_msg: Prints the last transmitted frame.

  The RX messages for LinkSetup look almost identical but they have the bit 0x20 set in the command bit, and Channel Setup has added one byte before Checksum containing Channel ID.

  NOTE:

  Several CAIF Messages might be concatenated. The maximum debug buffer size is 128 bytes.

Error Scenarios

- last_tx_msg contains channel setup message and last_rx_msg is empty -> The host seems to be able to send over the UART, at least the CAIF ldisc get notified that sending is completed.
- last_tx_msg contains enumeration message and last_rx_msg is empty -> The host is not able to send the message from UART, the tty has not been able to complete the transmit operation.
- if /sys/kernel/debug/caif_serial/<tty>/tty_status is non-zero there might be problems transmitting over UART.

  E.g. host and modem wiring is not correct you will typically see tty_status = 0x10 (hw_stopped) and ser_state = 0x10 (FLOW_OFF_SENT).

  You will probably see the enumeration message in last_tx_message and empty last_rx_message.