RCSL No: 43-GL11703

Edition: November, 1981

Author: Harald Villemoes

Title:

RC3502 LAM108/116 Driver Reference Manual

RCSL 42-i1592

A/S REGNECENTRALEN
af 1979

Keywords:

RC3502, Real Time PASCAL, Asynchronous Multiplexor, Driver.

Abstract:

This document describes the RC3502 LAM-driver. Messages and answers are described as well as the necessary process environment.

(18 printed pages).

Printed by A/S Regnecentralen af 1979, Copenhagen

RCSL 42-i1592

INDHOLDSFORTEGNELSE

1. General Description.

1.

The RC3502 LAM-driver supports full duplex V.24 asynchronous communication on 8 or 16 lines with transmission speed selectable between 110 bps and 1200 bps on a per line basis. The driver supports program controlled

- echoing,
- input continued, i.e. collection of characters received before an input buffer arrives,
- input conversion,
- timeout monitoring,
- modem control,
- conversion table sharing between lines.

Lines are numbered 0 to 7 or 0 to 15 for both input and output lines.

2. Messages and Answers. 2.

All messages are signalled to the LAM-driver general semaphore and all answers are returned to the answer semaphore except when the driver is removed or enteres an exception.

The general format for messages is

	message	answer
u1	function	function
u2	not used	result
u3	lineno	lineno
u4	not used	unchanged

2.1 Fixed Types. 2.1

All message data types starts with first, last, next: integer in accordance with the standard driver convensions.

```
line_status_type =
  packed record
    ?:           0..1;
    line_speed:  0..3;
      (* 0:  110 bps,
         1:  300 bps,
         2:  600 bps,
         3: 1200 bps  *)
    data_size:   0..3;
      (* 0: 5 data bits / char,
         1: 6 data bits / char,
         2: 7 data bits / char,
         3: 8 data bits / char  *)
    stop_bits:   boolean;
      (* true:  2 stop bits,
         false: 1 stop bit  *)
    pty_mode:    0..3;
      (* 0: odd parity,
         1: ignore parity, parity bits received
            are stripped off and ignored; chars
            xmitted has even parity bit supplied *)
         2: even parity,
         3: no parity, characters received and
            xmitted without parity bit  *)
    ?:           0..1;
    rts, (* request to send, outgoing modem signal *)
    dtr, (* data terminal ready, outg. modem signal *)
    dcd, (* data carrier detect, inc. modem signal *)
    rfs, (* ready for sending, inc. modem signal *)
    dsr, (* data set ready, incomming modem signal *)
    ?,?: boolean;

  end;  (* line_status_type *)
```

2.2 Control Messages. 2.2

Control messages except set conversion are always executed and returned immediately. Control messages are common for the input and output lines.

2.2.1 Sense Line. 2.2.1

Format:

	message	answer
u1	0	0
u2	-	result
u3	lineno	lineno
u4	-	unchanged
buf:	-	line status

```
sense_buffer_type =
  record
    f, l, n: integer;  (* not used *)
    line_status: line_status_type;
  end;
```

function:

The current state of the three incomming modem signals: DSR, RFS, and DCD is updated and the answer is returned.

2.2.2 Line Control. 2.2.2

Format:

	message	amswer
u1	4	4
u2	-	result
u3	lineno	lineno
u4	-	unchanged
buf	line_control	line_status

```
buffer_type =
  record
    f, l, n: integer;  (* not used *)
    new_line_state,
    actual_changes: line_status;
  end;
```

Function:

The record, actual_changes, is inspected for true or non-zero values and for each such found, the corresponding line_state is set.
Finally the function, sense line, is executed, the record, new_line_state, updated, and the answer returned.

2.2.3 Set Conversion.

Format:

```
      message       answer
u1       8            8
u2       -          result
u3    lineno        lineno
u4       -          unchanged
buf   conv_spec     conv_spec

conv_spec =
  record
    f, l, n: integer;  (* not used *)
    conv_control: integer;
      (* -2:   clear and push current conversion
               buffer, if any, under this message
               and  return this message,
         -1:   set conv_tab in this buffer as conver-
               sion table for this line. If a
               conversion table is already set for
               this line it is returned.
         0-15: set the conversion table for the
               specified line as conversion table
               well. If a conversion table is already
               set for this line, it is returned.  *)
    end;
```

Conversion is executed for input and attention operations only. Incomming characters are converted and classified using the value of the incomming characters for table lookup in conv_tab.
The type, conv_integer, in interpreted in one of two ways depending on the value of conv_integer.normal-conv . If normal_conv is true the value is used for normal classification of incomming characters while special conversion is performed when normal_conv is false.

```
conv-integer =
  packed record
    normal-conv: boolean;
      (* false: the value of conv-integer is used as
                index in conv-tab for the start-
                integer of a special conversion
                record,
         true:  this element is used for conversion
                and classification of the received
```

```
                character. *)
    attention: boolean;
      (* true: this character is an attention char.
         i.e: if an input-operation is present, it is
         terminated and returned with status att.
         with the value delivered, if specified.
         If an attention-operation and no input-
         operation is present, the attention-operation
         is terminated and returned with the value
         delivered, if specified.
         If an input- or attention-operation is pre-
         sent and an output-operation is being
         executed, this output-operation is terminated
         with status: attention.
         Note however that if no input- or attention-
         is present, the attention charecter is
         stored in the internal buffer but not
         executed. *)
    termination: boolean;
      (* an input operation is terminated with this
         character, which is delivered if specified.  *)
    blind: boolean;
      (* the value is not delivered *)
    noecho: boolean;
      (* the value is not echoed *)
    erase_last: boolean;
      (* if current buffer is empty, this character
         is delivered, if specified, but not echoed.
         If current buffer is non-empty, the last
         stored character is erased and the received
         character is delivered, if specified, and
         echoed, if specified.  *)
    erase_all: boolean;
      (* all characters in the current buffer are
         erased and the received character is
         delivered, if specified, and echoed,
         if specified.  *)
    mark: boolean;
      (* status mark is set in the result of the
         current input buffer if value is stored. *)
    conv_char: byte;
      (* the value to be delivered and echoed.  *)
  end;
```

If normal_conv is false, the binary value of conv-integer is the index in conv_tab of the first integer of a special conversion record:

```
spec_conv_record =
  packed record
```

```
      spec_conv_integer: conv_integer;
        (* values as normal conversion. *)
      length: integer;
        (* length of special echo (in bytes) *)
      spec_echo: array (1..length) of byte;
    end;
```

Function:

Conversion is set or cleared as specified in conv_spec. When an old conversion table is returned by a clear-function, this is done by PUSH´ing the old one under the clear-message and then returning this. Then an old conversion table is returned by setting up a new one, this is done by returning the old one.

2.2.4 Set Timers.

Format:

	message	answer
u1	12	12
u2	-	result
u3	lineno	lineno
u4	-	unchanged
buf	timers	timers

```
timers =
  record
    f, l, n: integer;  (* not used *)
    itimer1: integer;
      (* defines the timeout before first character
         input to a buffer. *)
    itimer2: integer;
      (* defines the timeout between chars. input. *)
    otimer1: integer;
      (* defines the timeout per character output
         from a buffer or echo. *)
  end;
```

All timers are in units of 1 second and a zero value means no timeout.
Note however that the value n means a period between n-1 and n seconds, e.g: the value 1 is senseless.

Function:

The timer values are set and the answer returned.

2.2.5 Reset. 2.2.5

Format:

	message	answer
u1	16	16
u2	-	result
u3	lineno	lineno
u4	-	unchanged
buf	-	unchanged

Function:

All operations in progress on the input and output line are terminated and all messages are returned with status: reset (= not processed). Finally the reset operation is returned with result: ok.

2.3 Transput Messages. 2.3

The driver supports message data stacks of a maximal depth of 1, i.e. on stacked (chained) buffers. All transput messages are formatted according to standard using first, last, and next indices.

2.3.1 Input. 2.3.1

Format:

	message	answer
u1	inpfunc	inpfunc
u2	-	result
u3	lineno	lineno
u4	-	unchanged
buf	-	rec-data

Function:

The value of inpfunc in u1 defines the actual input function for the message and is composed of a sum of binary numbers, each defining a specific characteristic of the input function:

inpfunc 1 : Basic input function.

inpfunc +2 : Echoing of significant characters are to be performed.

inpfunc +4 : Continuing input, i.e. characters received from returnal of the previous input operation and onto initiation of this input message are accepted instead of skipped.

inpfunc +8 : Attention, only characters, which are classified with attention, are accepted.

inpfunc +16 : Flow control by means of XON / XOFF characters are to be performed. The following action is taken:

- When the driver initiates execution of a flow control input message an XON-character (dec: 17) is output.
- When the driver terminates execution of a flow control input message an XOFF-character (dec: 19) is output.

2.3.2 output

2.3.2

Format:

	message	answer
u1	2	2
u2	-	result
u3	lineno	lineno
u4	-	unchanged
buf	data	unchanged

Function:

The data in the buffer is output and the answer

returned. A timeout condition or the recognition of an attention character will terminate the output-operation prematurely.

2.4 Results. 2.4

Basic Results:

0 operation executed succesfully.
1 operation not processed but returned by a reset operation.
2 error described by result modification.
4 illegal function code or lineno.

Result Modifications:

+0 timeout. The operation is terminated.
+8 echo error. The operation continues.
+16 attention. The operation terminates.
+32 parity or stopbit error. The oper. continues.
+64 overrun or character lost, e.g: hardware overrun or internal buffer overrun. The oper. terminates.
+128 mark. The operation continues.

2.5 Treatment of Characters Received. 2.5

Normally characters received are placed in an internal buffer (size= 32 chars) until an input operation is available. They are then converted, if conversion is defined, echoed, if specified, and stored in the input buffer.
If echoing or transmission of flow control characters are to be performed and the output line is busy executing an output operation no echoing is performed but status echo-error is set in the answer. The same applies to a timeout error during output of echo characters.
A character received with parity or stop-bit error is substituted by the character SUB (dec: 26) and status parity is set in the answer.
A received break signal is treated as a character with conversion index -1 when conversion is defined. If conversion is not defined, however, a break signal is treated as a character with parity error and a SUB-character is stored.

3. LAM-Driver Process. 3.

The LAM driver process is created with the following format:

```
PROCESS lamdriver(var lamsem: semaphore;
                  lamlevel: integer);
```

where lamsem is the driver input semaphore, and
lamlevel is the interrupt level of the lam.

LAM-driver stack size: 1440 words.

Recommended LAM-driver priority: 0.

RETURN LETTER

Title: RC3502 LAM108/116 Driver Reference Manual

RCSL No.: 43-GL11703

A/S Regnecentralen af 1979/RC Computer A/S maintains a continual effort to improve the quality and usefulness of its publications. To do this effectively we need user feedback, your critical evaluation of this manual.

Please comment on this manual's completeness, accuracy, organization, usability, and readability:

_____

_____

_____

_____

Do you find errors in this manual? If so, specify by page.

_____

_____

_____

_____

How can this manual be improved?

_____

_____

_____

_____

Other comments?

_____

_____

_____

_____

_____

Name: _____ Title: _____

Company: _____

Address: _____

Date: _____

Thank you

42-i 1288

..... Fold here .....

..... Do not tear - Fold here and staple .....

Affix
postage
here

A/S REGNECENTRALEN
af 1979

Information Department
Lautrupbjerg 1
DK-2750 Ballerup
Denmark