

## Update Information

P/N 1-960484-00  
Edition 1  
March 2000



# Intermec Fingerprint 7.31

 **intermec**  
Technologies Corporation

A **UNOVA** Company

## News in Intermec Fingerprint 7.31

*Intermec Fingerprint 7.31 is an enhancement of Intermec Fingerprint 7.2 via two intermediate versions (7.21 and 7.3), which has not been documented in any manuals.*

### General Improvements

- Supports EasyLAN 100i interface board (Ethernet), see separate documentation.
- New devices "status:" and "net1:" (for EasyLAN 100i interface)
- Supports RS 485 on "uart2:" (requires an Double Serial or Industrial Interface board), see pages 11 and 18.
- Extended Setup Mode with support for RS 485 and EasyLAN 100i, see pages 18 and 20.
- Improved data safety at power failure.
- New version of Intermec Shell (v. 4.4) with support for communication channel "net1:", see page 21.

### New Instructions

- CLIP ON/CLIP OFF      Enabling/disabling of partial fields, i.e. fields extending outside the print window, see page 2.
- GETASSOC\$              Getting a value from a string association, see page 4.
- GETASSOCNAMES\$      Traversing the tuplers of a string association, see page 5.
- MAKEASSOC              Creating an association, see page 6.
- SETASSOC                Setting a value for a tuple in a string association, see page 7.
- XORMODE ON/OFF      Enabling/disabling of xor/flip mode in connection with graphical operations, see page 8.

### Modified Instructions

- COMBUF\$                Support for communication channel 5 ("net1:" e.g. EasyLAN 100i)
- COMSET                 Support for communication channel 5 ("net1:" e.g. EasyLAN 100i)
- COMSET ON             Support for communication channel 5 ("net1:" e.g. EasyLAN 100i)
- COMSET OFF            Support for communication channel 5 ("net1:" e.g. EasyLAN 100i)
- COMSTAT                Support for communication channel 5 ("net1:" e.g. EasyLAN 100i)
- CSUM                    New checksum algorithm added, see page 3.
- DEVICES                Support for more devices ("status:" and "net1:").
- LBLCOND                New restriction. LBLCOND 2,<no. of dots> only works if the label is longer than the distance between the LSS and the printhead's dot line.
- MAP                     Support for communication channel 5 ("net1:" e.g. EasyLAN 100i)
- SETSTDIO               Support for communication channel 5 ("net1:" e.g. EasyLAN 100i)
- SETUP                  New setup parameters added, see page 9.
- SYSVAR                 New parameter (35) controls whether a new setup is saved to file or not, see page 10.  
Parameter (32), odometer resolution changed to 10 metres.

### Additions to the Fingerprint 7.2 Reference Manual (last-minute additions in Fingerprint 7.2 and 7.21)

- DEVICES                Devices "dll:", "cardx:", "par:" (for special applications only)  
Device "rs485:" for RS 485 communication.

### Other news in Fingerprint 7.31 Reference Manual

- Character sets have been revised according to Swiss 721 WGL4 BT in ubi-1b.pfr.

# CLIP ON/OFF

Field of Application	Statement for enabling/disabling the printing of partial fields.
Syntax	<hr/> <b>CLIP ON OFF</b> <hr/>
Remarks	<p>Partial fields means that the firmware will accept printable fields (text, bar codes, images, lines, and boxes) which extend outside the print window as specified by the printer's setup in regard of X-Start, Width, and Length. Even negative PRPOS values are allowed. However, all parts the printable fields outside the print window will be excluded from the printout, i.e. clipped at the borders of the print window.</p> <p>CLIP ON enables the use of partial fields; CLIP OFF disables the use and the error message 1003 "Field out of label" will result if any field extends outside the print window.</p> <p>Note the difference between the physical size of the label and the size of the print window specified by the printer's setup. It is the latter that decides where the fields will be clipped.</p>
Example	<p><i>In this example, only the last part of the text will be printed:</i></p> <pre>10      CLIP ON 20      PRPOS -100,30 30      PRTXT "INTERMEC PRINTER" 40      PRINTFEED RUN</pre>

# CSUM

<b>Field of Application</b>	<b>Statement calculating the checksum of an array of strings.</b>
<b>Syntax</b>	<hr/> <b>CSUM&lt;ncon&gt;,&lt;svar&gt;,&lt;nvar&gt;</b> <hr/> <p><b>&lt;ncon&gt;</b> is the type of checksum calculation:  1: Longitudinal Redundancy Check (LRC)  XOR in each character in each string  array[0][0] xor array[0][1] ... array[n][n]  2: Diagonal Redundancy Check (DRC)  right rotation, then XOR on each character in each string  rot(array[0][0] xor array[0][1])  3: Longitudinal Redundancy Check (LRC)  Strip string of DLE (0x10) before doing the LRC</p> <p><b>&lt;svar&gt;</b> if &lt;ncon&gt; = 1 or 2: The array of strings of which the checksum is to be calculated.  if &lt;ncon&gt; = 3: Checksum string.</p> <p><b>&lt;nvar&gt;</b> is the variable in which the result will be presented.</p>
<b>Remarks</b>	These types of checksum calculation can only be used for string arrays, not for numeric arrays. In case of CSUM3,<svar>,<nvar>, the resulting variable will be the indata for next CSUM calculation, unless the variable is reset.
<b>Example</b>	<p><i>In this example, the DRC checksum of an array of strings is calculated:</i></p> <pre> 10  ARRAY\$ ( 0 ) = " ALPHA " 20  ARRAY\$ ( 1 ) = " BETA " 30  ARRAY\$ ( 2 ) = " GAMMA " 40  ARRAY\$ ( 3 ) = " DELTA " 50  <b>CSUM 2,ARRAY\$,B%</b> 60  PRINT B% :REM DRC CHECKSUM RUN </pre> <p style="text-align: right;"><i>yields:</i></p> <p>252</p>

# GETASSOC\$

Field of Application

Function for getting a value from a string association.

Syntax

---

**GETASSOC\$ (<sexp<sub>1</sub>>, <sexp<sub>2</sub>>)**

---

<sexp<sub>1</sub>> is the name of the association (case-sensitive).

<sexp<sub>2</sub>> is the name of a tuple in the association.

Remarks

An association is an array of tuples, where each tuple consists of a name and a value.

Example

*This example shows how a string, including three stringnames associated with three start values, will be defined and one of them (time) will be changed:*

```
10  QUERYSTRING$ =  
    "time=UNKNOWN&label=321&desc=DEF"  
20  MAKEASSOC "QARRAY", QUERYSTRING$, "HTTP"  
30  QTIME$ = GETASSOC$("QARRAY", "time")  
40  QLABELS% = VAL(GETASSOC$("QARRAY", "label")  
50  QDESC$ = GETASSOC$("QARRAY", "desc")  
60  PRINT "time=";QTIME$, "LABEL=";QLABELS%,  
    "DESCRIPTION=";QDESC$  
70  SETASSOC "QARRAY", "time", time$  
80  PRINT "time="; GETASSOC$("QARRAY", "time")  
RUN
```

*yields:*

```
time=UNKNOWN LABEL=321 DESCRIPTION=DEF  
time=153355
```

# GETASSOCNAME\$

<b>Field of Application</b>	<b>Function for traversing the tuples of a string association.</b>
<b>Syntax</b>	<hr/> <b>GETASSOCNAME\$(<i>&lt;sexp&gt;</i>,<i>&lt;nexp&gt;</i>)</b> <hr/> <i>&lt;sexp&gt;</i> is the association to be traversed (case-sensitive). <i>&lt;nexp&gt;</i> specifies the tuple in the association. <i>&lt;nvar&gt;</i> = 0 specifies first tuple. <i>&lt;nvar&gt;</i> ≠ 0 specifies next tuple.
<b>Remarks</b>	An association is an array of tuples, where each tuple consists of a name and a value. To get the first position in the string association, <i>&lt;nvar&gt;</i> should be zero. Consecutive calls to GETASSOCNAME\$ with <i>&lt;nvar&gt;</i> non zero will traverse all variables in an undefined order. When a blank string ("") is returned, the last variable has been traversed.
<b>Example</b>	<p><i>This example shows how "QARRAY" is traversed (run example from GETASSOC first):</i></p> <pre> 10 LVAL\$ = GETASSOCNAME\$( "QARRAY" , 0 ) 20 WHILE LVAL\$ &lt;&gt; " " 30 RVAL\$ = GETASSOC\$( "QARRAY" , LVAL\$ ) 40 PRINT LVAL\$ ; " = " ; RVAL\$ 50 LVAL\$ = GETASSOCNAME\$( "QARRAY" , 1 ) 60 WEND RUN </pre> <p style="text-align: right;"><i>yields:</i></p> <pre> label=321 desc=DEF time=153355 </pre>

# MAKEASSOC

Field of Application	<b>Statement for creating an association.</b>
Syntax	<hr/> <b>MAKEASSOC &lt;sexp<sub>1</sub>&gt;, &lt;sexp<sub>2</sub>&gt;, &lt;sexp<sub>3</sub>&gt;</b> <hr/> <p>&lt;sexp<sub>1</sub>&gt; specifies the name of the association to be created (case-sensitive).</p> <p>&lt;sexp<sub>2</sub>&gt; contains an argument list of parameter tuples according to the convention in &lt;sexp<sub>3</sub>&gt;.</p> <p>&lt;sexp<sub>3</sub>&gt; should always be "HTTP" (case sensitive)</p>
Remarks	HTTP implies that the argument list in <sexp <sub>2</sub> > is encoded in "x-www-url-encoding".
Example	<p><i>This example shows how a string, including three stringnames associated with three start values, will be defined and one of them (time) will be changed:</i></p> <pre> 10   QUERYSTRING\$ =       "time=UNKNOWN&amp;label=321&amp;desc=DEF" 20   <b>MAKEASSOC "QARRAY", QUERYSTRING\$, "HTTP"</b> 30   QTIME\$ = GETASSOC\$("QARRAY", "time") 40   QLABELS% = VAL(GETASSOC\$("QARRAY", "label")) 50   QDESC\$ = GETASSOC\$("QARRAY", "desc") 60   PRINT "time=";QTIME\$, "LABEL=";QLABELS%,       "DESCRIPTION=";QDESC\$ 70   SETASSOC "QARRAY", "time", time\$ 80   PRINT "time="; GETASSOC\$("QARRAY", "time") RUN </pre> <p style="text-align: right;"><i>yields:</i></p> <pre> time=UNKNOWN LABEL=321 DESCRIPTION=DEF time=153355 </pre>

# SETASSOC

Field of Application	Statement for setting a value for a tuple in a string association.
Syntax	<hr/> <b>SETASSOC &lt;sexp<sub>1</sub>&gt;, &lt;sexp<sub>2</sub>&gt;, &lt;sexp<sub>3</sub>&gt;</b> <hr/> <p>&lt;sexp<sub>1</sub>&gt;            <i>is the name of the association (case-sensitive).</i>          &lt;sexp<sub>2</sub>&gt;            <i>is the name of the tuple</i>          &lt;sexp<sub>3</sub>&gt;            <i>is the value of the tuple.</i></p>
Remarks	An association is an array of tuples, where each tuple consists of a name and a value.
Example	<p><i>This example shows how a string, including three stringnames associated with three start values, will be defined and one of them (time) will be changed:</i></p> <pre> 10   QUERYSTRING\$ =       "time=UNKNOWN&amp;label=321&amp;desc=DEF" 20   MAKEASSOC "QARRAY", QUERYSTRING\$, "HTTP" 30   QTIME\$ = GETASSOC\$("QARRAY", "time") 40   QLABELS% = VAL(GETASSOC\$("QARRAY", "label")) 50   QDESC\$ = GETASSOC\$("QARRAY", "desc") 60   PRINT "time=";QTIME\$, "LABEL=";QLABELS%,       "DESCRIPTION=";QDESC\$ 70   <b>SETASSOC "QARRAY", "time", time\$</b> 80   PRINT "time="; GETASSOC\$("QARRAY", "time") RUN </pre> <p style="text-align: right;"><i>yields:</i></p> <pre> time=UNKNOWN LABEL=321 DESCRIPTION=DEF time=153355 </pre>



# XORMODE ON/OFF

**Field of Application**      **Statement for enabling or disabling the xor/flip mode of Intermec Fingerprint in connection with graphical operations.**

---

**Syntax**                      **XORMODE ON|OFF**

---

**Remarks**                      When XORMODE is set ON, dots are reversed – as opposed to set – by all graphical operations except bar codes. This means that if, for example two black lines cross, the intersection will be white. If XORMODE is set to OFF, the intersection will be black.

Default is XORMODE OFF.

**Example**                      *The following program illustrates the difference between XORMODE ON and XORMODE OFF. The two lines to the left are drawn with XORMODE disabled and the lines to the right with XORMODE enabled.*

```
10        XORMODE OFF  
20        PRPOS 0,50  
30        PRLINE 300,30  
40        DIR 4  
50        PRPOS 100,0  
60        PRLINE 200,30  
70        XORMODE ON  
80        DIR 1  
90        PRPOS 400,50  
100       PRLINE 300,30  
110       DIR 4  
120       PRPOS 500,0  
130       PRLINE 200,30  
140       PRINTFEED  
RUN
```

# SETUP

## New Parameters

The content of setup files can be listed by the use of the program FILELIST.PRG stored in the printer's permanent memory ("rom:"), or by COPYing the file to the communication channel of the host, usually "uart1:".

When creating setup files or setup strings, there is a special syntax for each parameter that must be followed exactly. Variable numeric input data are indicated by "n" – "nnnnn", alternative data are indicated by **bold** characters separated by vertical bars (|). Compulsory space characters are indicated by underscore spaces (\_). Note that some parameters listed below may only apply to a certain printer model or an optional device.

*Important!*

*Do **not** include any underscores (\_) or vertical bars (|) when typing a setup string or file!*

"SER-COM,UART1 UART2 UART3,BAUDRATE,300 600 1200 2400 4800 9600 19200 38400 57600 115200"	
"SER-COM,UART1 UART2 UART3,CHAR_LENGTH,7 8"	
"SER-COM,UART1 UART2 UART3,PARITY,NONE EVEN ODD MARK SPACE"	
"SER-COM,UART1 UART2 UART3,STOPBITS,1 2"	
"SER-COM,UART1 UART2 UART3,FLOWCONTROL,RTS/CTS,ENABLE DISABLE"	
"SER-COM,UART1 UART2 UART3,FLOWCONTROL,ENQ/ACK,ENABLE DISABLE"	
"SER-COM,UART1 UART2 UART3,FLOWCONTROL,XON/XOFF,DATA_FROM_HOST,ENABLE DISABLE"	
"SER-COM,UART1 UART2 UART3,FLOWCONTROL,XON/XOFF,DATA_TO_HOST,ENABLE DISABLE"	
"SER-COM,UART2,PROT_ADDR,ENABLE DISABLE"	NEW!
"SER-COM,UART1 UART2 UART3,NEW_LINE,CR/LF LF CR"	
"SER-COM,UART1 UART2 UART3,REC_BUF,nnnnn"	
"SER-COM,UART1 UART2 UART3,TRANS_BUF,nnnnn"	
"SER-COM,UART2,PROT_ADDR,nn"	NEW!
"NET-COM,NET1,NEW_LINE,CR/LF LF CR"	NEW!
"FEEDADJ,STARTADJ,nnnn"	(negative value allowed)
"FEEDADJ,STOPADJ,nnnn"	(negative value allowed)
"MEDIA,MEDIA_SIZE,XSTART,nnnn"	
"MEDIA,MEDIA_SIZE,WIDTH,nnnn"	
"MEDIA,MEDIA_SIZE,LENGTH,nnnnn"	
"MEDIA,MEDIA_TYPE,LABEL_(w_GAPS) TICKET_(w_MARK) TICKET_(w_GAPS) FIX_LENGTH_STRIP VAR_LENGTH_STRIP"	
"MEDIA,PAPER_TYPE,TRANSFER DIRECT_THERMAL"	
"MEDIA,PAPER_TYPE,DIRECT_THERMAL,LABEL_CONSTANT,nnn"	
"MEDIA,PAPER_TYPE,DIRECT_THERMAL,LABEL_FACTOR,nnn"	
"MEDIA,PAPER_TYPE,TRANSFER,RIBBON_CONSTANT,nnn"	
"MEDIA,PAPER_TYPE,TRANSFER,RIBBON_FACTOR,nnn"	
"MEDIA,PAPER_TYPE,TRANSFER,LABEL_OFFSET,nnn"	
"MEDIA,CONTRAST,-10% -8% -6% -4% -2% 0% 2% 4% 6% 8% 10%"	
"PRINT_DEFS,PRINT_SPEED,nnn"	
"PRINT_DEFS,LTS_VALUE,nn"	
"NETWORK,IP_SELECTION,MANUAL DHCP BOOTP RARP"	NEW!
"NETWORK,IP_ADDRESS,nnn.nnn.nnn.nnn"	NEW!
"NETWORK,NETMASK,nnn.nnn.n.n"	NEW!
"NETWORK,DEFAULT_ROUTER,nnn.nnn.nnn.nnn"	NEW!

# SYSVAR

**New SYSVAR parameter**     **35: Setup Saving**     (*Set or Read*)

This parameter allows you to decide whether a change in the printer's setup is to be saved as a file (i.e. be effective after a reboot or power down) or not be saved (volatile). You can also read for which alternative the printer is set. Note that the SYSVAR (35) setting at the moment when the new setup is entered decides whether it will be saved or not.

The alternatives are:

- 0 =     Setup saved to file (non-volatile)     Default
- 1 =     Setup not saved to file (volatile)

## RS 485 Communication

As an option, EasyCoder XP- and F-series printers can have a Double Serial Interface Board or an Industrial Interface Board installed that can be fitted with circuits and straps to provide RS 485 on "uart2:".

RS 485 is a 2-line screen cable point-to-point or multidrop loop connection, where the two lines switch between transmitting and receiving data according to instructions from the software. By default, the port is set to receive data. Before transmission of data, the port is switched to transmit. After the last character has been transmitted, the port is switched back to receive.

When an interface board configured for RS 485 is fitted in the printer, the serial communication setup for "uart2:" in the Setup Mode will only contain the following parameters:

- Baudrate (300 – 57600)
- Protocol address (enable/disable)
- New line (CR/LF, LF, CR)
- Receive buffer (30 – 9999)
- Transmit buffer (30 – 9999)
- Connected hardware (RS485) *[read only]*
- Protocol address (0 – 31)

### Terminology

#### Protocol header

First part of transported information that includes:

- header start (192),
- destination protocol address (0-31),
- source protocol address (0-31),
- length (1-255),
- protocol-type & checksum.

*Also see description later in this chapter.*

#### Data area

A number of data-bytes transported with or without a protocol header.

### Choice of Protocol Mode / Raw Mode

There are three main alternatives:

- A. Protocol Mode = protocol header + data area (incl. substitution)
- B. Protocol Mode with COMSET control
- C. Raw Data Mode = No protocol control of data transmission, only data-area is sent, i.e. same as ordinary RS232 or RS422

Selection of type of mode is made by a combination of device specification in the OPEN statement, selection of protocol address (enable/disable) in the Setup Mode, and specification of protocol address in the Setup Mode (also see diagram on page 18).

## RS 485 Communication, cont'd.

### A: Protocol Mode

```
OPEN "rs485:" FOR INPUT
OPEN "rs485:<prot.addr.>" FOR OUTPUT
```

### B: Protocol Mode + COMSET control

```
SETUP "SER-COM,UART2,PROT ADDR ENABLE"
OPEN "uart2:" FOR INPUT or OPEN "rs485:" FOR INPUT
OPEN "rs485:<prot.addr.>" FOR OUTPUT
```

### C. Raw Data Mode

```
SETUP "SER-COM,UART2,PROT ADDR DISABLE"
OPEN "uart2:" FOR INPUT
OPEN "uart2:" FOR OUTPUT
```

### COMSET Control with Protocol Mode (case B above)

COMSET 2, <sexp<sub>1</sub>>, <sexp<sub>2</sub>>, <sexp<sub>3</sub>>, <sexp<sub>4</sub>>, <nexp<sub>2</sub>> is same as ordinary COMSET, but the use of <sexp<sub>1</sub>>, <sexp<sub>2</sub>>, <sexp<sub>3</sub>> and <sexp<sub>4</sub>> and <nexp<sub>2</sub>> is done on the received data-area.

COMSET 2 is directly connected to the driver for "uart2:", but use of either "uart2:" and "rs485:" with PROT ENABLE is possible.

Data received on rules setup by COMSET 2, will be stored in COMBUF\$(2), even if the device "uart2:" not is opened. If <nexp<sub>2</sub>> = 0 the first character will be lost, so let <nexp<sub>2</sub>> be at least 1.

At background reception with COMSET 2.., the rules setup by using <sexp<sub>1</sub>> to <sexp<sub>4</sub>> is executed on the data-area with <nexp<sub>2</sub>> size.

If <nexp<sub>2</sub>> = 1 and all <sexp<sub>n</sub>> are set to "", then the first character of data is stored in COMBUF\$(2) and the rest of data is available by reading from "uart2:" or "rs485:".

### *Sending, example:*

*Sending sequence with protocol from unit-address 1 to 5, with data "ABCDE" when key F1 is depressed, is shown below:*

```
NEW
IMMEDIATE OFF

REM ##### RS485 SENDING #####
SETUP "SER-COM,UART2,PROT ADDR,ENABLE"
SETUP "SER-COM,UART2,PROTOCOL ADDR,1"
ON KEY 10 GOSUB XSEND : KEY 10 ON
MAIN:
GOTO MAIN

REM ----- START SENDING -----
XSEND:
  OPEN "rs485:5" FOR OUTPUT AS 1
  BUFSEND$="ABCDE"
  PRINT# 1, BUFSEND$;
  CLOSE 1
  RETURN

IMMEDIATE ON
```

## RS 485 Communi- cation, cont'd.

*Receiving, example:*

*Receiving is controlled by COMSET. Only data to addr. 5 is received.*

```

NEW
IMMEDIATE OFF

REM##### RS485 RECEIVING WITH COMSET #####
SETUP "SER-COM,UART2,PROT ADDR,ENABLE"
SETUP "SER-COM,UART2,PROTOCOL ADDR,5"
COMSET 2,"","","","",1
ON COMSET 2 GOSUB RCVNETWORK
COMSET 2 ON
MAIN:
GOTO MAIN

REM ----- RS485 INTERRUPT ROUTINE -----
RCVNETWORK:
    TICKEND%=0
    BUFRCV$=COMBUF$(2)
    OPEN "rs485:" FOR INPUT AS #3
    BUFLLEN%=LOC(3)
    WHILE ((BUFLLEN%>0) OR (TICKEND%>TICKS))
        IF BUFLLEN%>0 THEN
            BUFRCV$=BUFRCV$+INPUT$(BUFLLEN%,3)
            TICKEND%=TICKS+TICKTIMEOUT%
            BUFLLEN%=LOC(3)
        ENDIF
    WEND
    CLOSE 3
    COMSET 2 ON

REM ----- OUTPUT INFO TO CONSOLE -----
PRINT "LEN="; LEN(BUFRCV$); "MSG="; BUFRCV$
RETURN

IMMEDIATE ON

```

## RS 485 Communication, cont'd.

### RS 485 Protocol Specification

All packets of data must be preceded by a header record, in which all data are binary:

START	DST	SRC	LEN	PROTO	CRC	<Data record/Request >
Header record: 5 bytes						

START indicates the start of the header record using the character ASCII 192 decimal, see note!

DST is the destination protocol address address 0–31 (1 byte).

SRC is the source protocol address 0–31 (1 byte).

LEN is the size in bytes of the data record, max. 249 characters (1 byte).

PROTO specifies type of protocols 0 or 1 (1 byte), see below.

CRC is the checksum of the header record (1 byte), i.e. the inverted sum of DST+SRC+LEN+PROTO bytes.

- PROTO = 0

This protocol is used for ordinary transfer of data between two units. The syntax is:

START	DST	SRC	LEN	PROTO=0	CRC	<Data record>
-------	-----	-----	-----	---------	-----	---------------

- PROTO = 1

This protocol is used for communication check from a host computer (cannot be sent from a printer!). Instead of a data record, a REQUEST byte (0 or 1) appends the header record. Any printer in the loop that has an open rs485 file can be checked if it is on-line (REQUEST = 0), or inquired for the number of seconds that have passed since its last startup or reboot (REQUEST = 1). If the printer is on-line, it will answer by returning the corresponding REQUEST byte, in the latter case followed by the time expressed as a 10-digit value with leading zeros.

*Example 1. The host computer sends:*

START	DST	SRC	LEN	PROTO=1	CRC	REQUEST=0
-------	-----	-----	-----	---------	-----	-----------

*The printer replies:*

START	DST	SRC	LEN	PROTO=1	CRC	REQUEST=0
-------	-----	-----	-----	---------	-----	-----------

*Example 2. The host computer sends:*

START	DST	SRC	LEN	PROTO=1	CRC	REQUEST=1
-------	-----	-----	-----	---------	-----	-----------

*The printer replies:*

START	DST	SRC	LEN	PROTO=1	CRC	REQUEST=1+time (nnnnnnnnnn)
-------	-----	-----	-----	---------	-----	-----------------------------

## RS 485 Communication, cont'd.

### Substitution Rules

Type of data	Original	Transport byte Substitution	Destination
Start packet mark <START>	(192)	=> (192) =>	(192)
Header & data info (START)	(192)	=> (219), (220) =>	(192)
Header & data info (ESCAPE)	(219)	=> (219), (221) =>	(219)
Header info (other than above)	(x)	=> (x) =>	(x)
Data info (LF)	(LF)	=> according to New Line setup	
Data info (other than above)	(x)	=> (x)	

- **Notes about safe data transport**

RS 485 protocols 0 and 1 delivers data from one unit to another with CRC control of the header record, i.e.:

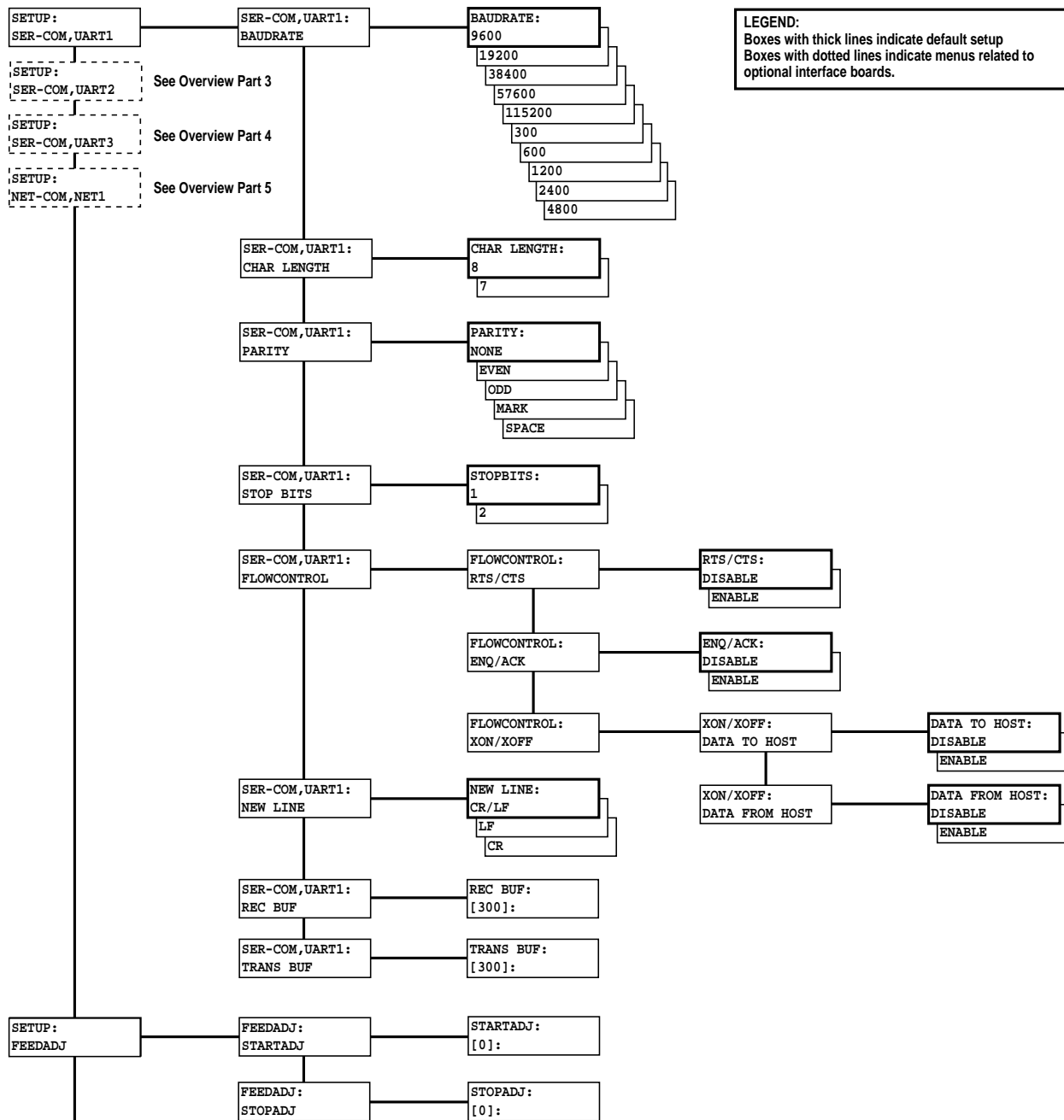
- destination
- source
- length
- protocol type

There is no check or CRC control of the data packet. If any check fails, no packet will be resent and there is no guarantee that the transmitted packet will be received by the destination unit.

To make sure that data will be transported correctly, you should let each data packet make a reply to the source unit.

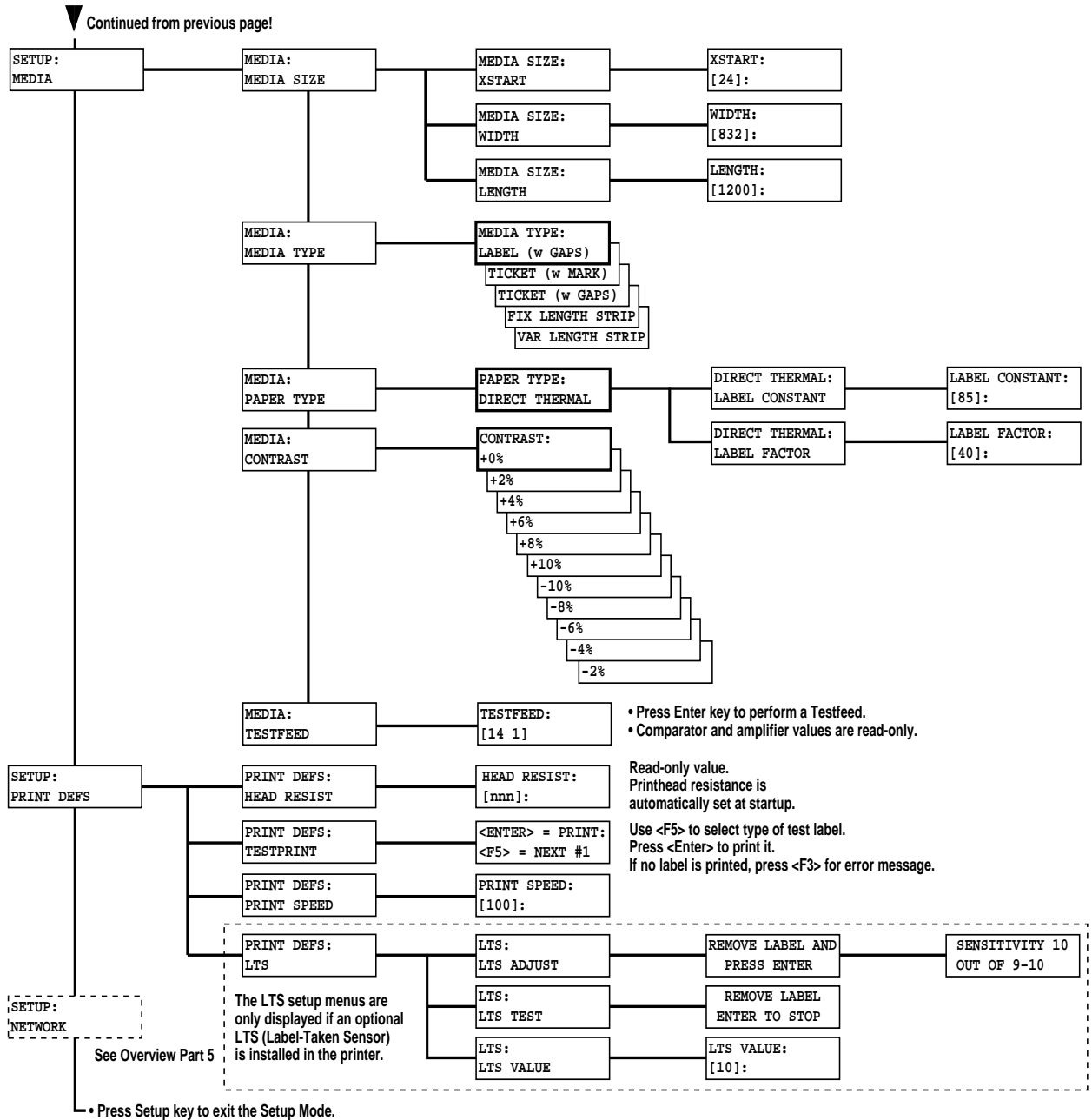


# Setup Mode Overview, Part 1



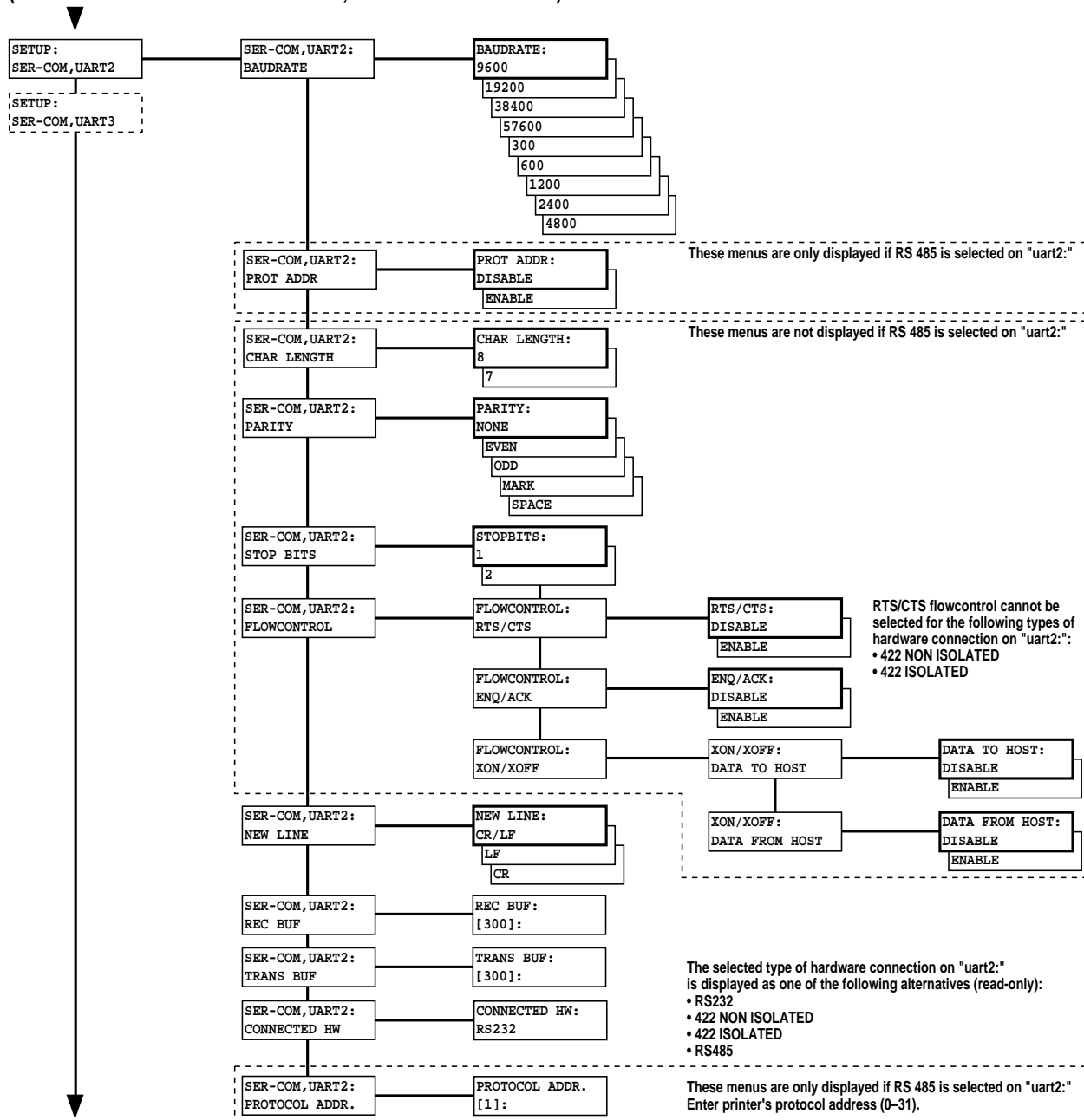
Continued on next page!

## Setup Mode Overview, Part 2



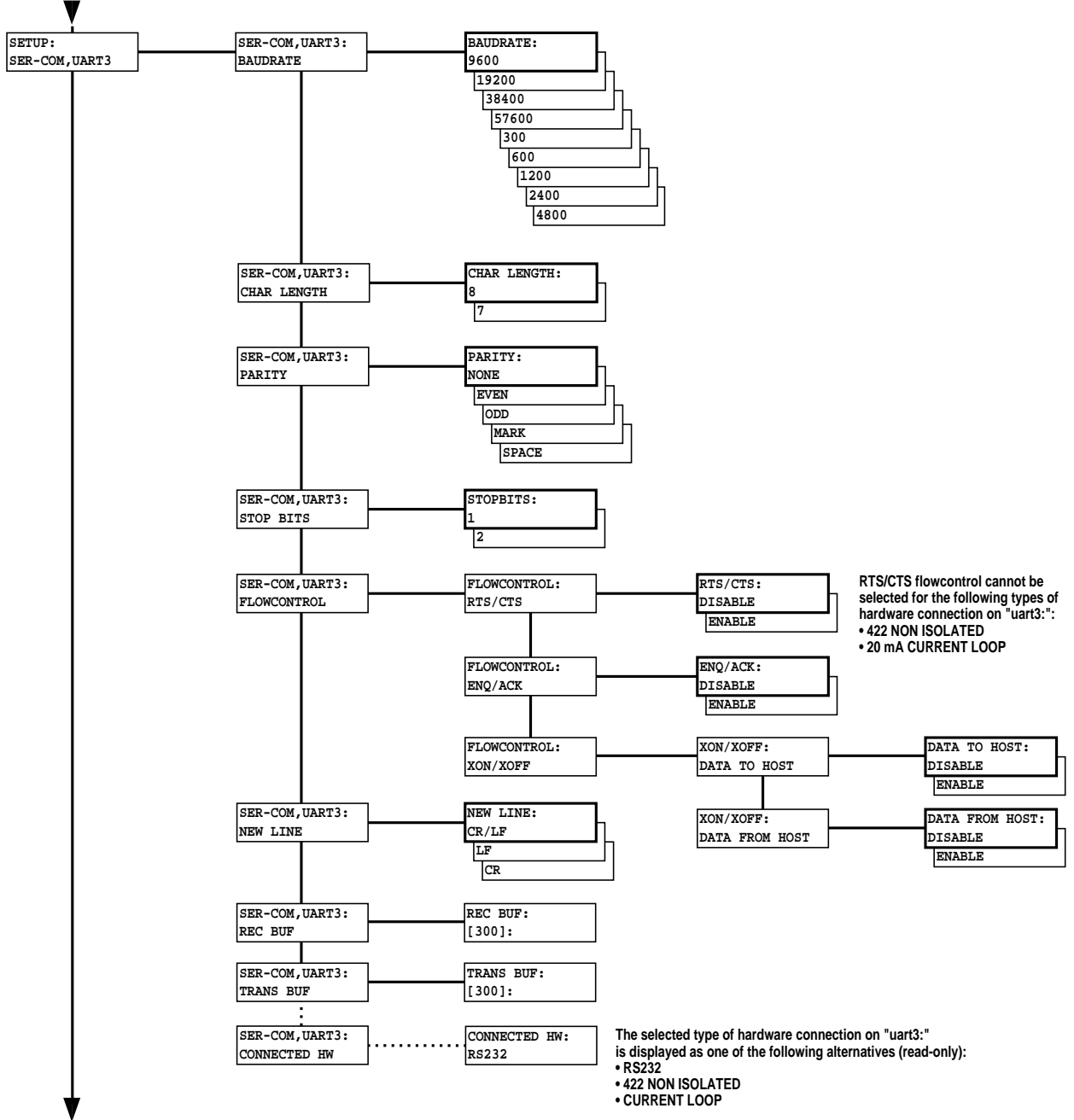
## Setup Mode Overview, Part 3

If an optional Double Serial or Industrial Interface Board is fitted, additional menus will be displayed (in case of Double Serial Interface Board, also see Overview Part 4):



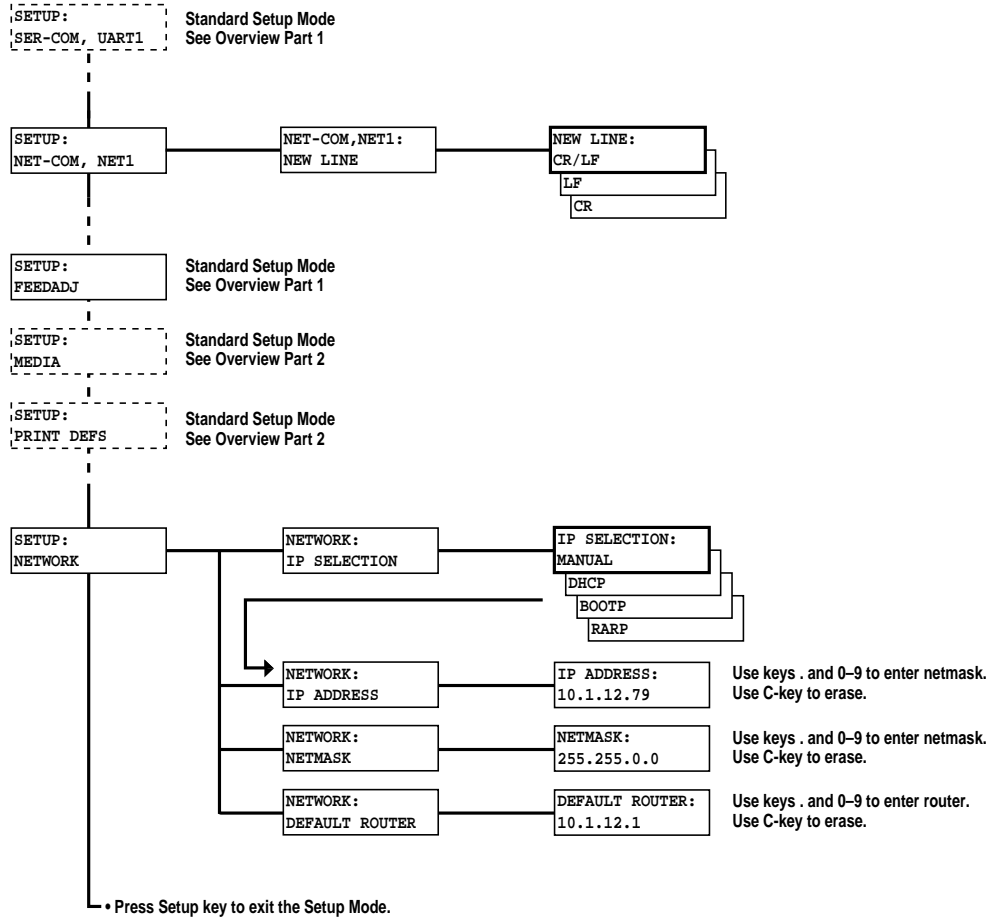
## Setup Mode Overview, Part 4

If an optional Double Serial Interface Board is fitted, additional menus will be displayed:



## Setup Mode Overview, Part 5

If an optional an EasyLAN 100i interface board is fitted, additional menus will be displayed.



# Intermec Shell 4.4

