V.90 modem software on ADSP 218x family

User's Guide

Version 5.3
February 6th, 1999

ANALOG DEVICES

TELINDUS

TABLE OF CONTENTS

TABLE OF FIGURES

1. INTRODUCTION

1.1. PRODUCT OVERVIEW

The Block diagram in Figure 1 gives an overview of the main building blocks (**modules**) in the system and their interconnection.

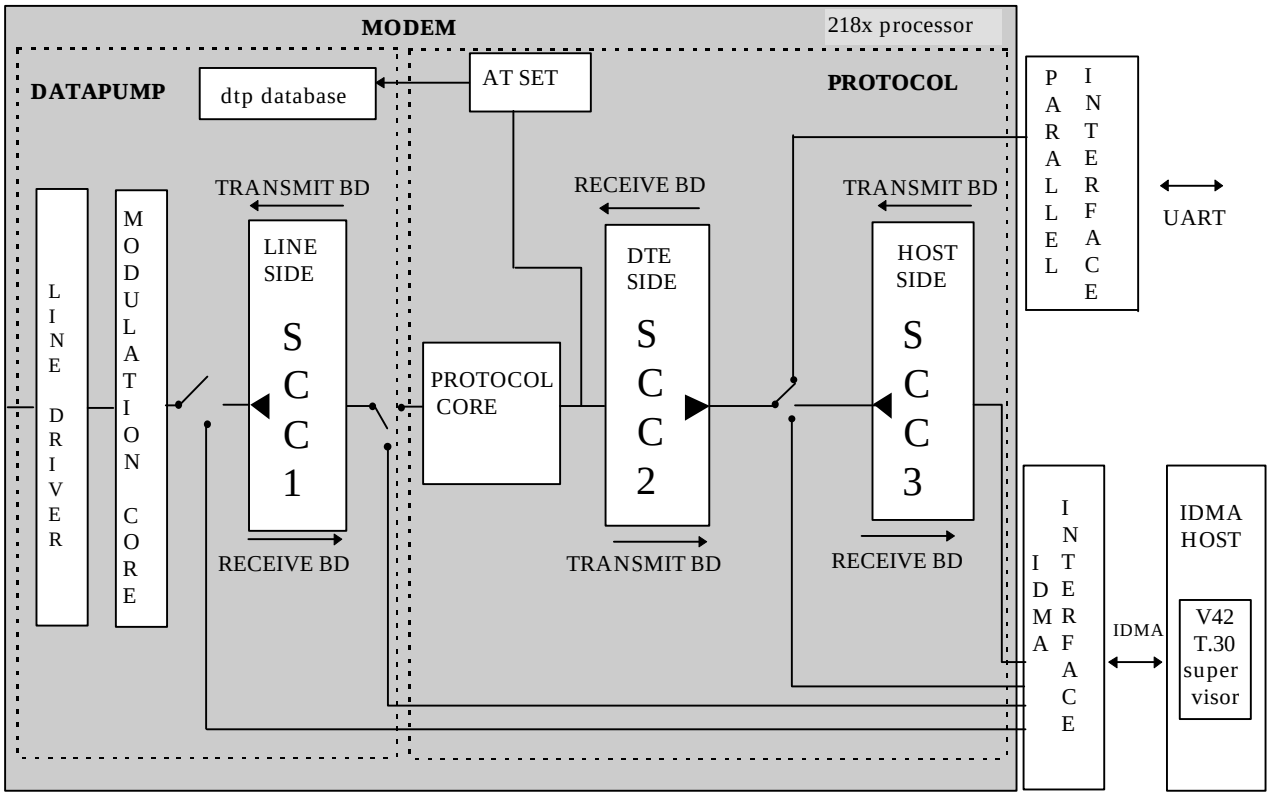


Figure 1 : System block diagram[1]

The DSP part of the software contains 2 **core modules**:

1.1.1. Modulation core

- Modulation complying with ITU-T V.90, V.34bis, V.34, V.32bis, V.32, V.22bis, V.22, V.23, V.21, Bell Recommendations Bell 212A, Bell 103.
- FAX mode modulation complying with V.17, V.27 ter, V.29, V.21 ch 2
- Country specific automatic dialler

1.1.2. Protocol core

- Error correction according to ITU-T V.42 or MNP 4 and data compression according to ITU-T V.42bis or MNP 5
- AT ® commands
- FAX class1, class2, T.30 protocol[2]

[1] SCC = Serial Communications Controller; BD = Buffer Descriptor
[2] In development

In this document a software package *with protocol and modulation part* will be called '**modem**', a package *without protocol part* will be called '**data-pump**'.

The DSP software also contains several **peripheral modules**. For each of these peripherals a number of configurations exist.

1.1.3. line drivers

- AD1843 codec (daisy chain master and slave)
- AD1819 codec
- AD73311 codec
- SGS-Thomson STLC7545 codec
- T1_E1 interface device

1.1.4. data/control interfaces for data-pump software

- idma synchronous data interface without framing
- idma data interface with asynchronous (V.14) framing through SCC1
- idma data interface with synchronous (HDLC) framing through SCC1
- idma memory mapped control interface via the data-pump database

1.1.5. data/control interfaces for modem software[3]

- idma asynchronous data interface with protocol through SCC2
- idma asynchronous data interface with protocol and HDLC framing through SCC3
- idma asynchronous control interface via AT SET commands

1.1.6. boot interface

- BDMA interface
- IDMA interface

The global software package also contains a number of modules running on the Host system, the **host modules**. These modules are written in C-language.

- V42 error correction[4] : in case of data-pump DSP systems that require a error free connection.
- T30 fax protocol : for fax class1 modem systems.
- data-pump supervisor : for control and follow up of the data-pump train and retrain behaviour.

[3] As shown in the block diagram a software configuration with an asynchronous data/control interface through a external UART component (16550) also is available . This configuration runs on a **demo-board.** A product which is at any time available for customer evaluation.

[4] The V42 protocol as a host based module is currently in development

1.2. PRODUCT FORMAT

The software package is available in several format types,

1.2.1. source files

Following items currently are available in source code :

- Source code files for the all codec drivers. They can be used as a development starting point for customers having a very specific line driver solution.
- Certain parts of the protocol core, for those applications for which the data-pump software runs on the DSP and the protocol runs on the IDMA host system.
- Certain parts of the DSP kernel that could be useful as example to build you own kernel system.
- The BDMA loader software that follows the Telindus BDMA/IDMA binary file format.
- Certain utilities are available in source form, for example the prom split program that transforms the executable files into a binary file.

1.2.2. object files

Some of the core and peripheral **modules** of the software are available in object file format. They are standalone link-able modules with a completely memory mapped interface. The object files specifically are of interest for **integrators**[5] who want to integrate these modules into their existing system. The object code is re-locatable. Following items currently are available in object format :

- modulation core modules for all modulations
- the SCC1 module for both V14 and HDLC framing
- line driver modules for all supported codec types.

1.2.3. executable/binary files

For both **data-pump** and **modem** software, we build and maintain **customised software** versions for specific user configurations[6]. These customised software versions are available for the **user** as a binary prom image file, for both IDMA and BDMA loading, or as executable (non re-locatable)code files. The latter case can be of interest for the **user** who prefers to start from a working modem or data-pump system, compiled for his platform, but wants to customise[7] this system :

- add specific tasks to the modem or data-pump task switcher tables at the executable level.
- tune the modem or data-pump by changing parameters in the set-up tables.

IMPORTANT REMARK : This document is intended for the **user** of the executable/binary format. It explains all aspects of the modem/data-pump interfacing through the IDMA port.

[5] A related document, called *Integrator's Manual*, explains the integration of the object files.
[6] A related document, called *Questionnaire,* should be used as a guide line to describe your software configuration. It's tables list all the configuration parameters together with their possible values. You should fill in the column 'customer'. It will allow us to compile customised software for you.
[7] A related document, called *Integrator's Manual*, explains the integration of the executable files.

1.3. MINIMUM PROCESSOR REQUIREMENTS

full featured modem software runs on :

- ADSP2187 (40Mips, only internal memory)
- ADSP2181 (40Mips, 3 external rams 32k*8 , 10 nsec access)

data-pump and fax class1 modem software run on :

- ADSP2181 (33Mips, only internal memory)

data-pump software without SCC1, all modulations up to V.32bis runs on :

- ADSP2186 (33Mips, only internal memory)

1.4. MEMORY AND MIPS REQUIREMENTS

The complete software is too large to fit into the DSP internal memory. Therefore, the software has been segmented into **bootpages**. Each page has a specific functionality.

Table 1: overview of bootpages, gives a brief functional description of all the pages. It lists the prom image file size and MIPS consumption for every page and also indicates the **bootpage number** for every page. All numbers are based on the RCS version TUPLE1 of June/98 and can slightly change during product development.

page nbr	Name	Functionality	MIPS[8]	prom image file size (bytes)
9	STARTUP	Loaded in memory **after reset**. It sets up the modem in a default state and provokes the loading of the dial page.		
0	Dial	provides: -ring detection (incoming calls) -tone generation (DTMF, single tone) -Country specific automatic dialler These items are explained in detail in a related document, called *'Dialler'*.	3	19752
1	V.22	covers ITU-T V.22, V.22bis, Bell B212A and modulation auto mode (EIA/TIA PN-2330)	6	21999
2	V.32	covers ITU-T V.32, V.32bis and V.33	12	36964
3	fsk	covers ITU-T V.23, V.21, V.21 ch 2, and Bell 103	3	12036
4	fax	covers ITU-T V.17, V.29, V.27ter	7	31872
6	V.8	cover ITU-T V.8	3	23368
7	INFO	cover ITU-T V.34 and V.90 phase 2	6	22292
8	V.34	cover ITU-T V.34, phase 3-4	24-30[9]	53002
10	modem protocol	covers ITU-T v42, MNP4, ITU-T v42bis, MNP5, ODP/ADP, ASYNC PPP	8[10]	73660
11	AT ®-set off line		1	41200
12	AT ® -set on line	covers TIES protocol	1	8268
13	V.90A[11]	cover ITU-T V.90, phase 3-4, APCM		
14	V.90D	cover ITU-T V.90, phase 3-4, DPCM	25..28[12]	51584
15	fax protocol[13]	covers ITU-T T.30		
16	LL page	The 'low level' page provides a number of low level functions. They are explained in detail in a related document, called *'Low level functions'*.		

Table 1: overview of bootpages

[8] For all modulation pages the MIPS have been measured for a data-pump without SCC1, to calculate the MIPS consumed by SCC1 following rule can be applied : **MIPS=(1/10000)*linebitrate**
[9] The MIPS consumption depends on the enabled features.
[10] Assuming a DTE bit rate of 115200bps, 2K dictionary and 33600bps on line.
[11] In development
[12] The MIPS consumption depends on the enabled features
[13] In development

Table 2: DM and PM of the object code modules, lists all currently available object code modules with their program and data memory size. All numbers are based on the RCS version TUPLE1 of June/98 and can slightly change during product development.

Object code module	PM size (words)	DM size (words)	
DIAL modulation core(tonedetection, generation, autodialler)	4633	1426	
V.8 modulation core	2389	1912	
INFO modulation core	5072	5636	
v.90D modulation core	11000	9290	
V.34 modulation core	12978	9660	
V.32 modulation core(V.32,V.32bis,V33)	7513	6476	
V.22 modulation core (V.22,V.22bis,automode)	5699	1619	
FAX modulation core (V29,V27ter,V17)	6705	6054	
FSK modulation core (V21,V23)	2536	1487	
linedriver	512	652	
SCC1(HDLC + V14)	4718	772	

Table 2: DM and PM of the object code modules

1.5. DOCUMENT OVERVIEW

This document is organised as follows.

- Chapter 2 'BOOT INTERFACE', describes the process of loading software in internal memory.
- Chapter 3 'LINE DRIVERS' describes the analogue interface or line interface.

These Chapters are common for data-pump and modem software.

- Chapter 4 'HOST INTERFACE (MODEM SOFTWARE)' describes the interfacing between the HOST processor and DSP for the modem software
- Chapter 5 'HOST INTERFACE (DATA-PUMP SOFTWARE)' describes the interfacing between the HOST processor and DSP for the data-pump software.

General Remark:
Throughout this document, a data memory location in the DSP is never indicated by it's absolute address, but by a **symbolic name**. The same applies for a bit position within a location.

1.6. RELATED DOCUMENTS

- 'INTEGRATORS MANUAL' : this document discusses in detailed the topic of object code and executable code integration.
- 'LOW LEVEL FUNCTIONS' : gives a detailed description of the interface of all low level functions on the LL page.
- 'DIALLER' : gives a detailed description of interface and functionality of the low level dialler functions and of the automatic dialler present on the dial page.
- 'QUESTIONNAIRE' : should be used as a guide line to describe your software configuration. It's tables list all the configuration parameters together with their possible values. It will allow us to compile customised software for you.

2. BOOT INTERFACE

2.1. THROUGH BDMA PORT

An external ROM of at least 4 Mb is required to store the software, and should be connected to the BDMA port[14]. Loading is completely controller less:

* **After reset** the DSP will automatically load the STARTUP page and start execution.
* **During runtime** the DSP loads another page when necessary.

2.2. THROUGH IDMA PORT

The HOST processor loads the software using writes through the **IDMA port**[15] :

- **After reset** the STARTUP page should be loaded. DSP execution starts automatically after completing the download.
- **During runtime** a page should be loaded when requested by the DSP. This is discussed more in detail:

1. DSP requests a page by pushing the corresponding event in the event buffer , sets the '**BOOT'** bit in **RSTATUS_DBS**, and puts itself in a wait state. The requested page is specified in location **BOOTPAGE_NR**. Table 1: overview of bootpages, gives for each possible page the corresponding number.
2. HOST processor retrieves[16] the requested page from it's memory and starts loading it in the DSP 's memory.
3. After completing the download, HOST processor sets the **BOOTFINISHEDBIT** in location **WSTATUS**.
4. DSP starts execution of the new page.

2.3. BOOT DURATION RESTRICTIONS

The duration of the boot operation is restricted. The boot operations of the software happen at moments no activity is happening on the telephone line. These idle periods are restricted and so also is the boot period. Following rules apply:

- in general the idle period (and boot operation) should never be longer than 75 ms
- when booting from INFO to V.34 idle time is restricted to 70 ms
- when booting from V.22 to V.32 idle time is restricted to 6.6 ms
- booting from Dial page can be done without any time restrictions
- retrain booting from V.34 to INFO is restricted by the retrain time out[17] of the remote modem.

[14] A BDMA hardware reference design is available
[15] A IDMA hardware reference design is available
[16] Appendix 6.1 explains how to retrieve a page in the provided file format in the case of IDMA loading.
[17] This is a vendor dependent parameter. In the case of the TELINDUS modem, this retrain time out equals 24 seconds

2.4. FLOW DIAGRAM OF BOOT SEQUENCE

In case of the modem software package the DIAL page also includes 'AT SET off line', they are booted simultaneously in one boot operation. Similarly the V.8 page includes PROTOCOL and 'AT SET on line'.

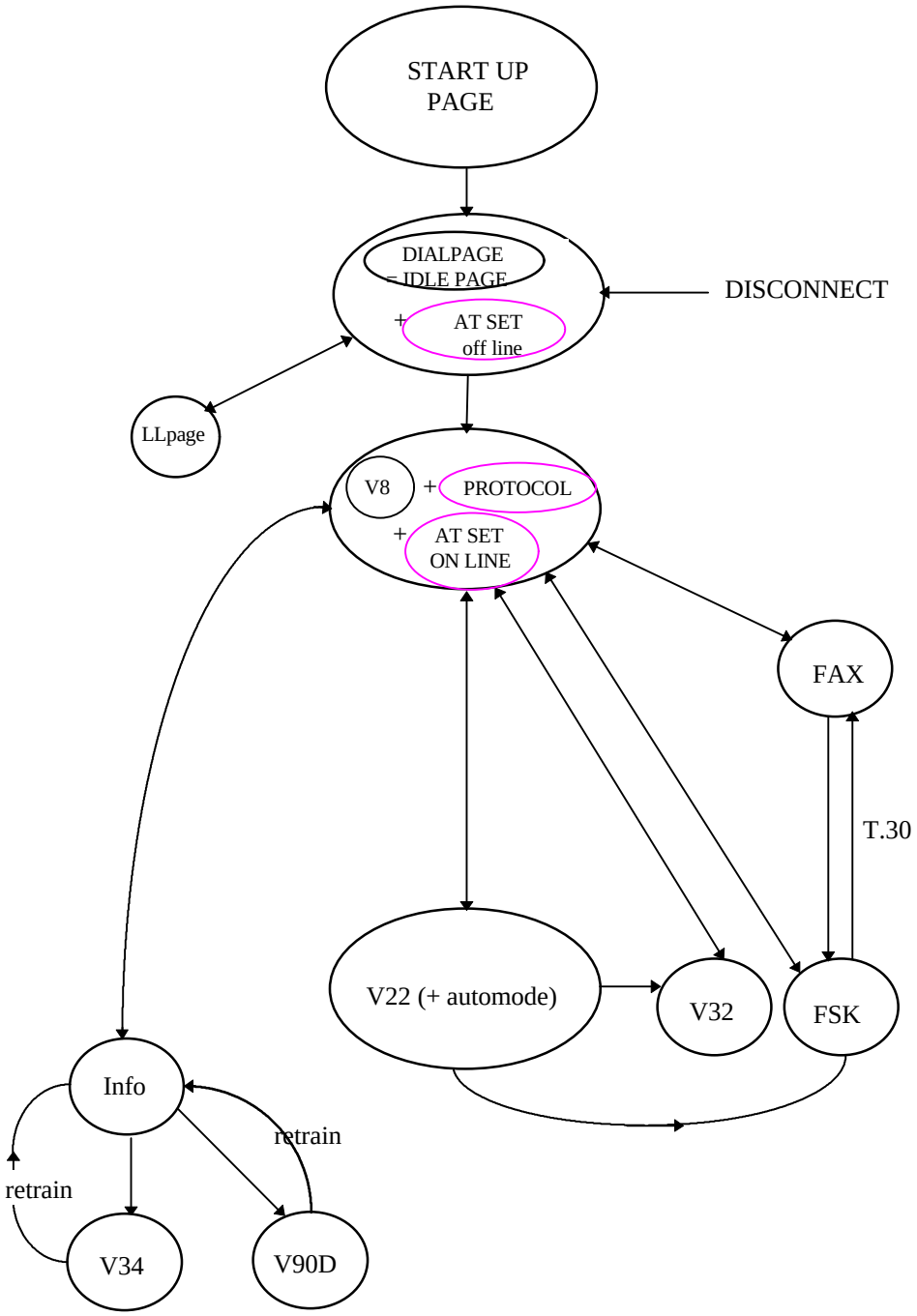


Figure 2 : boot operation flow diagram

3. LINE DRIVERS

3.1. INTRODUCTION

The DSP exchanges analogue samples with the line interface device through SPORT0.

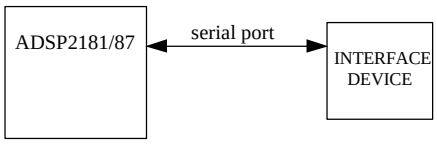


Figure 3 : line interface

Remark: In the case of PSTN applications, the ring signal enters the DSP through one of the flag-pins (PF0 .. PF7).

3.2. CODEC INTERFACES

3.2.1. AD1843 codec

DSP takes care of programming the CODEC and setting up the serial connection. This component can be used for all modulation types.
Two drivers are available :

- daisy chain master driver
- daisy chain slave driver

3.2.2. AD1819 codec

DSP takes care of programming the CODEC and setting up the serial connection. This component can be used for all modulation types.

3.2.3. AD73311 codec

DSP takes care of programming the CODEC and setting up the serial connection. This component can be used for all modulation types up to V.32bis.

3.2.4. SGS-THOMSON STLC7545 codec

DSP takes care of programming the CODEC and setting up the serial connection. This component can be used for all modulation types.

3.3. T1 (E1) INTERFACE DEVICE

In the case of a T1 (E1) interface SPORT0 is programmed for multichannel operation and u-law or A-law companding.

The 5 sport0 multichannel control registers can be programmed by the user through the **SPORT0 setup** locations of the data-pump database.

Sp0CntrlReg
Sp0MCRecL
Sp0MCRecM
Sp0MCTXL
Sp0MCTXM

These locations are explained in chapter 5.3.1

4. HOST INTERFACE (MODEM SOFTWARE)

4.1. INTRODUCTION

The HOST processor has four tasks:

- **Exchanging data packets.** The SCC (Serial communication controller) interface is used for this task.
- **Controlling the modem**, for example: go off-hook, hang up connection, dial a number, channel selection, speed selection, selection of the error correction protocol, selection of the compression protocol, etc. All functions of the modem are controlled by **AT ®** compatible commands.
- **Loading software** in the DSP 's memory[18]
- **configuration** of the system

4.1.1. modem block diagram

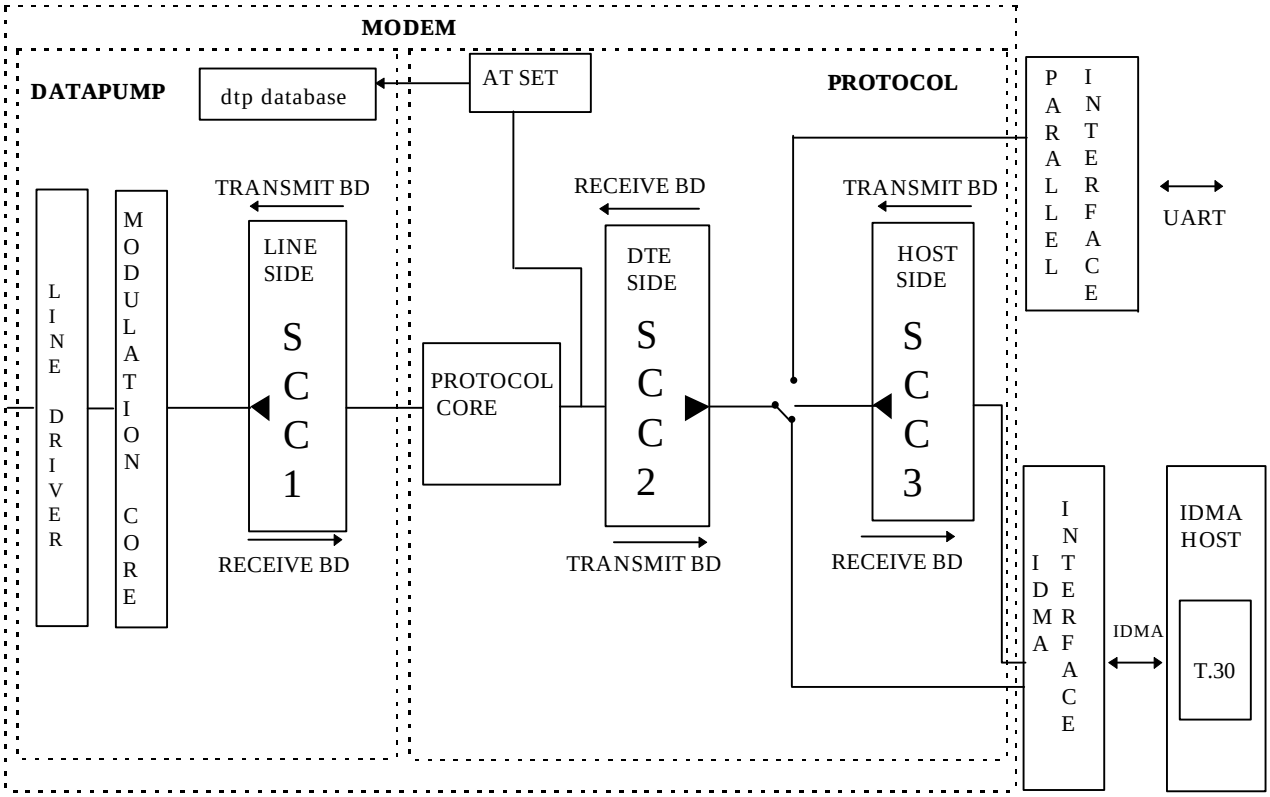


Figure 4 : modem block diagram

[18] Only if the IDMA-loading method is applied.

naming conventions:

- SCC1 = DCE side SCC = LINE side SCC
- SCC2 = DTE side SCC
- SCC3 = HOST side SCC
- modem RX side buffers =
    - data-pump RX buffers,
    - DCE SCC RX buffers,
    - DTE SCC **TX** buffers
    - HOST SCC RX buffers
- modem TX side buffers =
    - data-pump TX buffers,
    - DCE SCC TX buffers,
    - DTE SCC **RX** buffers
    - HOST SCC TX buffers

The user can interface to the modem via the **IDMA interface**[19]

4.1.2. data flow

The **IDMA host** can transfer **data** via

- a parallel asynchronous interface to V42 using the functionality of SCC2.
- a parallel asynchronous interface using the functionality of SCC3, which can handle asynchronous HDLC frames, used in asynchronous PPP connections. In this case data flow has to be configured. This is done in the location **DATACONFIG** (default=h#0000) of the data-pump write database. This location is explained in section 5.3.1

[19] As shown in the block diagram a software configuration with an asynchronous data/control interface through a external UART component (16550) also is available . This configuration runs on a **demo-board.** A product which is at any time available for customer evaluation.
This interface uses the 16550 UART component, hooked up to the parallel bus of the ADSP2181 and placed in IO space. The interface always transfers **modem control** information to the AT set command interpreter. The AT set command interpreter updates the **modem database.** The AT commands can be used to configure the modem, when it is **off line** and **online**.

4.1.3. control flow

Following sections explain the control operation for a system with IDMA HOST.

4.1.3.1. The IDMA Interface

In the case of IDMA interface an essential part of the control flow **from DSP to host** is the **event generation**. The event mechanism is used to signal to the host whenever
1. the interface signals (CTS,DCD,DSR) change [20]
2. newly received information has filled buffers at modem RX side
3. Buffers have been emptied at modem TX side and should be filled
4. a boot operation is requested for IDMA download of a new page

Whenever the modem has new information available, it generates an event, i.e.

- it pushes an **event code** on the **event buffer** . The event buffer is an eight locations wrap around buffer. The event code specifies the kind of information available.

- it increments the **event counter**. The three LSB 's of this event counter are used as a wrap around write pointer for the event buffer. The pointer points to the last written position + 1 of the event buffer. The full 16 bit constitute a 16 bit wrap around counter counting the number of events. The host system has it's own event counter, pointing to the last read location + 1. If the difference between Host and DSP event counter is higher than eight, event information has been lost.

- it generates an **event pulse** using one of the DSP flags.

The host system can use the event system in polling or interrupt mode.
In polling mode the host regularly has to compare the DSP event counter with it's internal counter. Whenever there is a difference, it should read the event(s) from the event buffer. In interrupt mode the same operation should be triggered by a interrupt.

The status of the modem is maintained in two **memory mapped databases**, the **modem database** and the **data-pump database**. Although not essential for the operation of the interface from DSP to host, it can be useful for the host to read out the strapping and status of modem and data-pump.

In the case of IDMA interface, control flow **from HOST to DSP** is the **memory mapped**.
This memory mapped interface is used to signal to the DSP whenever
1. if the interface signals (RTS,DTR) change [21]
2. Buffers have been emptied at modem RX side[22]
3. Buffers have been filled at modem TX side [23]
4. a boot operation is finished
5. to switch the system configuration from its default (in **DATACONFIG)**

On a regular basis the DSP checks the memory mapped interface and takes appropriate action.

[20] through the **UARTMIRROR** registers of the MODEM database
[21] through the register **DTESIGNALS** of the MODEM database
[22] only for the DTE, DCE and HOST SCC interfaces, the data-pump interface assumes that the host reads (RX side) and writes (TX side) in time
[23] identical remark

4.1.3.2. modem setup

The IDMA host always should transfer **modem control** information to the AT set command interpreter using the functionality of SCC2. The AT commands can be used to configure the modem, but only when it is **off line**.

The AT command set has three subtypes.

- General (Hayes standardised) AT commands. See section 4.3.1.2, 'General at_set commands'.
- 'AT%s' commands : these commands update the %s registers. All these registers are grouped together and form the **AT modem database**. For more information see section 4.3.1.4, '%S Registers'.
- 'ATs' commands : these commands update the s registers. For more information see section 4.3.1.3, 'S-Registers'.

For the software package with host IDMA interface additional control information can be exchanged through IDMA setup and status locations. These setup locations are not accessible through AT set and only apply in the case of an IDMA system. Section 4.3.2, 'IDMA control interface' explains in detail.

No data-pump registers should be modified by the IDMA host. Two exceptions exist.

1. In the case of a T1/E1 interface the **SPORT0 setup** registers have to be initialised, see chapter 5.3.1
2. In the case the signal flow does not correspond to the default, the **DATACONFIG** register has to be changed. This register is explained in detail in section 5.3.1

4.1.3.3. modem status

The IDMA host always can request **modem status** information in two ways.

With the 'at%s?' command the user can read out the status of the '%s registers' through the AT set command interpreter using the functionality of SCC2, but only when the modem is **off line**. These registers are explained in more detail in chapter 4.3.1.4

For the software package with host IDMA interface additional status information of the protocols, modem and the connection can be derived from the IDMA status locations. These status locations are not accessible through AT set and only apply in the case of an IDMA system. Section 4.3.2, 'IDMA control interface' explains in detail.

4.2. DATA INTERFACE

4.2.1. Introduction

The next sections give a general explanation of the SCC interface.

In following sections CP (communications processor) should be interpreted as
- for DTE side, SCC2 : Host
- for DCE side, SCC1 : DSP
- for HOST side, SCC3 : DSP

user should be interpreted as :
- for DTE side,SCC2 : DSP(protocol or AT set)
- for DCE side, SCC1 : DSP(protocol) or Host depending on **DATACONFIG**
- for Host side, SCC3 : HOST

For SCC1 and SCC2 the description of the buffer must be seen from the view of the V42, MNP or buffered mode. For SCC2 the bits are defined as used by these routines and it's the IDMA host who simulates the CP as if the SCC2 was transmitting the data to a serial interface. In this case, transmit BufferDescriptors are used to pass data to the host!

4.2.2. The SCC structure

The SCC channel (SCC1 and SCC2) is described by the structure below, which can be accessed by IDMA. Most of these registers are used by the internal software.

```
struct SCC
  {
   word ModeReg;
  word nbits;
  word EventReg;
  word (*TXState)();
  void (*RXState)(word data);
  word TotalBufCount;
  SCCBUFDESCRIP RXBufferDescriptor[MAXBUFDEF];
  SCCBUFDESCRIP TXBufferDescriptor[MAXBUFDEF];
  word Modereg2;
  word MaxBufLength;    /*maximum length of one buffer descriptions/
  word RXSCCBuffer;     /*receiving buffer offset 0,1,2..7*/ /*RBD*/
  word RXSCCBufferOffset; /*offset in the receive buffer ,1,2,…MaxBufLength/
  word RXSCCBufferBitOffset;
  word RXSCCCRC;
  word RXWorkReg;
  word RXWorkReg1;
  word RXWorkReg2;
  word RXBreakLength;
  word TXSCCBuffer;      /*transmitter buffer offset 0,1,2..7*/ /*TBD*/
  word TXSCCBufferOffset;
  word TXSCCBufferBitOffset;
  word TXSCCCRC;
  word TXSCCCRC1;
  word TXBitsCalculated;
  word TXWorkReg;
  word TXWorkReg1;
  word TXWorkReg2;
  word *TXPtr;
  word TXBreakLength;
  word XON1;
  word XON2;
  word XOFF1
  word XOFF2
  lword BreakAsyncReg;
  word BreakMask;
  word nbitsTx;
  };
```

The SCC tables below (for example section 4.2.3.1 for the UART SCC) give a more detailed explanation of each field of the structure and also indicate how they are accessed by the DSP and by the IDMA host.

Because the IDMA host and the DSP work asynchronously, there might be a problem when both the processors are using the same location at the same time.

For the **BufferDescriptor** fields it is bit 15 of the status location that indicates whether the host can use a BufferDescriptor or not. Bit 15 assures that the host and the DSP never operate on the same BufferDescriptor (and corresponding buffer). For the different SCCs the flow diagrams in their respective chapters explain in more detail the use of the bit 15.

For **all remaining locations** of the SCC structure, the problem is solved by the introduction of the event system. The protocol software only during initialisation writes the other fields of the list. For the EventReg and the ModeReg, this statement is true for the bits, which are relevant for the host. Whenever there has been a (re-)initialisation by the protocol software, the DSP generates an **initialisation event** to the host. The host has to do the following actions:

1. Read through IDMA the values of the fields of the SCC structure in the DSP.

2. Reinitialise its interface driver.

4.2.3. UART SCC

4.2.3.1. SCC structure

the base address of the DTESCCstructure can be found in the location DTESCCstructPtr of the modem IDMA database

The table below gives an overview of the usage of each field of the SCC structure by the protocol software, which runs on the DSP, and by the CP, running on the IDMA HOST.

Offset				IDMA host (CP)		DSP (protocols)	
				R	W	R	W
0	ModeReg			X		X	X
	D	SENDBREAK	if bit set SCC sends break and clears bit (only useful for SCC1), the break time is placed inTXBreakLength				
	B	FLOWCONTROL XON/XOFF	1 : enables XON/XOFF flow control (only useful for SCC1)				
	A	SENDXOFF	if bit set SCC sends XOF and clears bit (only useful for SCC1)				
	9	SENDXON	if bit set SCC sends XON and clears bit (only useful for SCC1)				
	8	V14	1 : use V14 syasy (must be set only for SCC 1 in async mode)				
	7, 6, 5	PARITY	0 : no parity 1: odd parity 2 : even parity 3 : space parity 4 : mark parity				
	4	ASYNC BITS	0 : 7 data bits 1 : 8 data bits				
	2	ENT	1 : enable transmitter				
	3	ENR	1: enable receiver				
	1, 0	MODE	0 : HDLC (only possible for SCC1) 1 : async				
1	Nbits		number of bits of a symbol. For the DTE interface, this number is fixed to 16 since the host is writing words to the DSP	X			X
2	EventReg						X
3	TXState		state variable of the CP transmitter	X	X		
4	RXState		state variable of the CP receiver	X	X		
5	TotalBuf Count			X	X		
6 .. 1D	RXBufferDescriptor[8*3]		Data associated with the SCC receiver is stored in buffers. Each buffer is referenced by a BufferDescriptor(BD). BD's are located in a BD array. The BD array allows to define up to eight buffers for each channel .	X	X	X	X
	offset 0	STATUS AND CONTROL					
	offset 1	DATA LENGTH					
	offset 2	DATA BUFFER POINTER					

1E .. 35	TXBufferDescriptor[8*3]		Data associated with the SCC transmitter is stored in buffers. Each buffer is referenced by a BufferDescriptor(BD). BD's are located in a BD array. The BD array allows to define up to eight buffers for each channel .	X	X	X	X
	offset 0	STATUS AND CONTROL					
	offset 1	DATA LENGTH					
	offset 2	DATA BUFFER POINTER					
36	Modereg2		indicates the number of buffers and the character operation mode of the buffers				
	F .. 4	SPARE					
	4	CharMode	1: all buffers operate with two characters per location 0: all buffers operate with one character per location default charmode = 0				
	3 .. 0	NbrofBufs	indicates the number of transmit buffers and receive buffers.The default number of the RX and TX data buffers is initialised by the modem software to : default number = 6.				
37	MaxBufLength		the maximum length of an RX or TX buffer. The default size of the RX and TX data buffers is initialised by the modem software to : default size=40 memory locations.	X			X
38	RXSCCBuffer		indicates in which BufferDescriptor the receiver CP is writing data	X	X		X
39	RXSCCBufferOffset		indicates which is the last received character in the current BufferDescriptor, even when the buffer is not closed.	X	X	X	X
3A	RXSCCBufferBitOffset						
3B	RXSCCCRC						
3C	RXWorkReg						
3D	RXWorkReg1						
3E	RXWorkReg2						
3F	RXBreakLength						
40	TXSCCBuffer		indicates from which BufferDescriptor the transmitter is sending data	X	X		X
41	TXSCCBufferOffset		indicates how many characters in the buffer already are transmitted	X	X		X
42	TXSCCBufferBitOffset						
43	TXSCCCRC						
44	TXSCCCRC1						
45	TXBitsCalculated			X	X		X
46	TXWorkReg						
47	TXWorkReg1						
48	TXWorkReg2						
49	TXPtr			X	X		
4A	TXBreakLength		If the SENDBREAK bit is set in ModeReg the transmitter will send a break for a time as indicated by TXBreakLength (1 unit is 416 uS(= 1/2400)) (not supported)				
4B	XON1		Define the xon character. Two xon characters can be specified on which the transmitter will restart transmission. If one of these characters are not used, the highest bit of the word has to be set. Flow control with the parallel interface of SCC2 has no sense because there is no serialising of the data and the flow control are regulated by not reading the data. (not supported)				

4C	XON2	Defines the second xon character (not supported)				
4D	XOFF1	Defines the first xoff character (not supported)				
4E	XOFF2	Defines the second xoff character (not supported)				
4F	BreakAsyncRegHigh					
50	BreakAsyncRegLow					
51	BreakMask					
52	nbitsTx	not in use				

Table 3 : SCC2 structure for UART operation

4.2.3.2. UART receive BufferDescriptor (RXbd)

The CP reports information about each buffer of received data by its BDs. The RX BD is shown in figure 5. The CP closes the current buffer, generates a maskable interrupt and starts to receive data in the next buffer due to any of the following events:

1. Detection of an error during message processing
2. Detection of a full received buffer
3. Reception of a programmable number of consecutive IDLE characters.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET +0	E	-	W	I	-	-	-	ID	-	-	BR	FR	PR	-	-	-
OFFSET +2	DATA LENGTH															
OFFSET +4	RX BUFFER POINTER															

Figure 5 : UART Receive BufferDescriptor

An example of the UART receive process is shown in figure 6. This shows the resulting state of the RX BDs after receipt of 10 characters, an idle period and five characters - one with a framing error. The example assumes that MaxBufLength = 8 in the SCC parameter RAM.

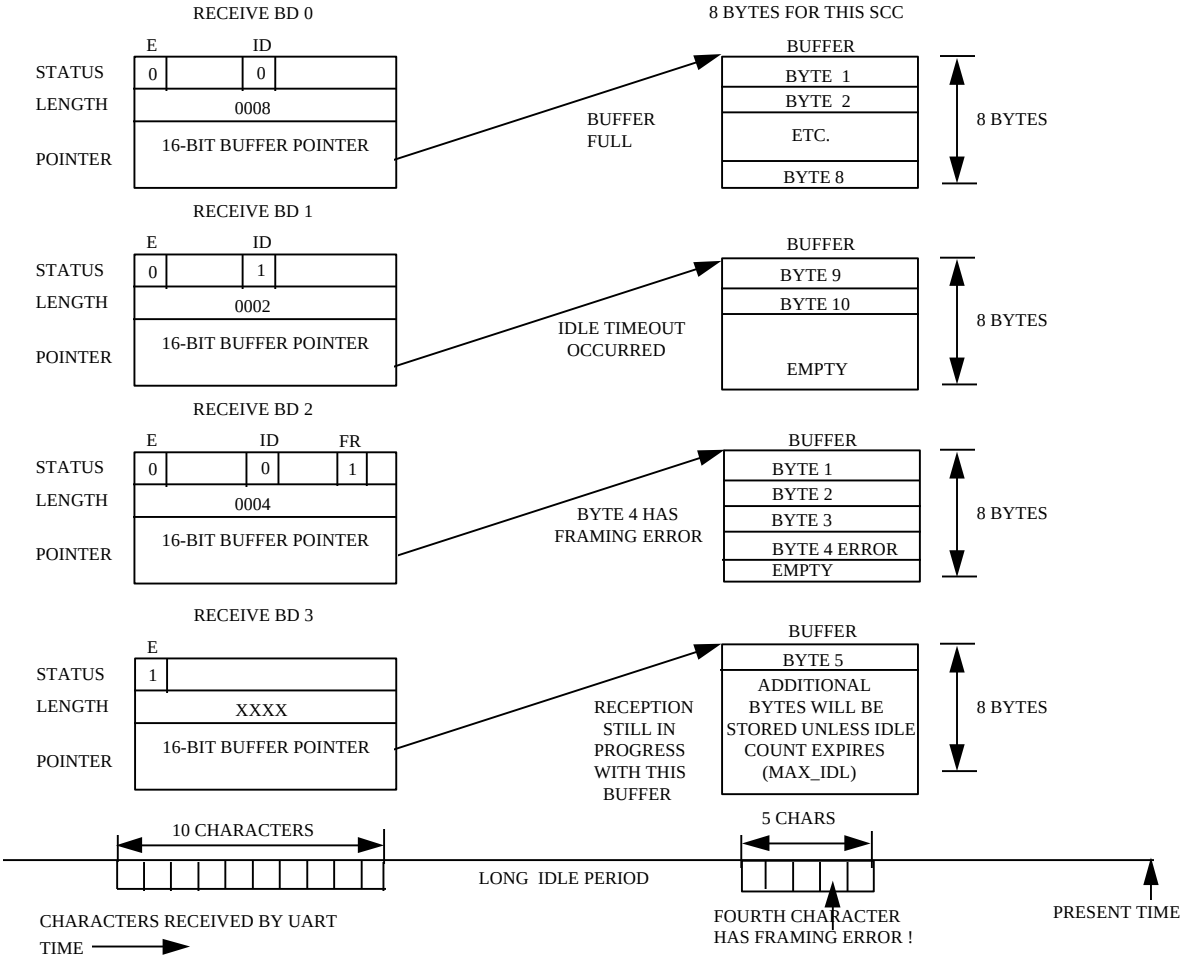


Figure 6 : UART RX BD Example

control and status bits.

E - Empty

0 = The data buffer associated with this BD has been filled with received data, or data reception has been aborted due to an error condition. The user is free to examine or write to any fields of the BD.
1 = The data buffer associated with this BD is empty. This bit signifies that the BD and its associated buffer are available to the CP. The user should not write to any fields of this BD after it sets this bit. The empty bit will remain set while the CP is currently filling the buffer with received data.

W - Wrap (Final BD in Table)

0 = This is not the last BD in the RX BD table.
1 = This is the last BD in the RX BD table. After this buffer has been used, the CP will receive incoming data into the first BD in the table, allowing the user to use fewer than eight BDs to conserve internal RAM.

NOTE

The user is required to set the wrap bit in one of the first eight BDs; otherwise, errant behavior may occur.

I = Interrupt(does not apply)

0 = No interrupt is generated after this buffer has been filled.
1 = The RX bit in the UART event register will be set when this buffer has been completely filled by the CP.

The following bits contain status information written by the CP after it has finished receiving data in the associated data buffer.

ID - Buffer closed on reception of idles(does not apply)

The buffer was closed due to the reception of the programmable number of consecutive IDLE sequences.

BR - Break Received

A break sequence was received. The break length is indicated by the data length field.

FR - Framing Error(does not apply)

A character with a framing error was received and is located in the last byte of this buffer. A framing error is detected by the UART controller when no stop bit is detected in the receive data string.

PR - Parity Error(does not apply)

A character with a parity error was received and is located in the last byte of this buffer.

Data Length

The data length is the number of octets written to this BD's data buffer.
It is written by the CP once as the BD is closed.

RX Buffer Pointer

The receive buffer pointer points to the first location of the associated data buffer.

Data is presented to the CP for transmission on an SCC channel by arranging it in buffers referenced by the channel's TX BD table.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET +0	R	X	W	I	-	-	-	-	-	-	-	-	-	-	-	-
OFFSET +2	DATA LENGTH															
OFFSET +4	TX BUFFER POINTER															

Figure 7 : UART Transmit BufferDescriptor

The first word of the TX BD contains status and control bits. The following bits are prepared by the user before transmission and set by the CP after the frame has been transmitted.

R - Ready

0 = This buffer is not currently ready for transmission. The user is free to manipulate this BD (or its associated buffer). The CP clears this bit after the buffer has been transmitted or after an error condition has been encountered.
1 = The data buffer, which has been prepared for transmission by the user, has not been transmitted or is currently transmitting. No fields of this BD may be written by the user once this bit is set.

I - Interrrupt

0 = No interrupt is generated after this buffer has been serviced.
1 = The TX bit in the UART event register will be set when this buffer has been serviced by the CP, which can cause an interrupt.

Bits 8 - 1 - Reserved for future use.

Data Length
The data length is the number of octets that the CP should transmit from this BD's data buffer. It is never modified by the CP. This value should be normally greater than zero. The data length may be equal to zero with the P bit set and only a preamble will be sent.

TX Buffer Pointer
The transmit buffer pointer points to the first location of the associated data buffer and may be even or odd. The buffer may reside in either internal or external memory.

4.2.3.4. IDMA - SCC2 receiver interaction

This flow diagram below gives an example of how an IDMA host driver could interface with the DTE SCC structure .

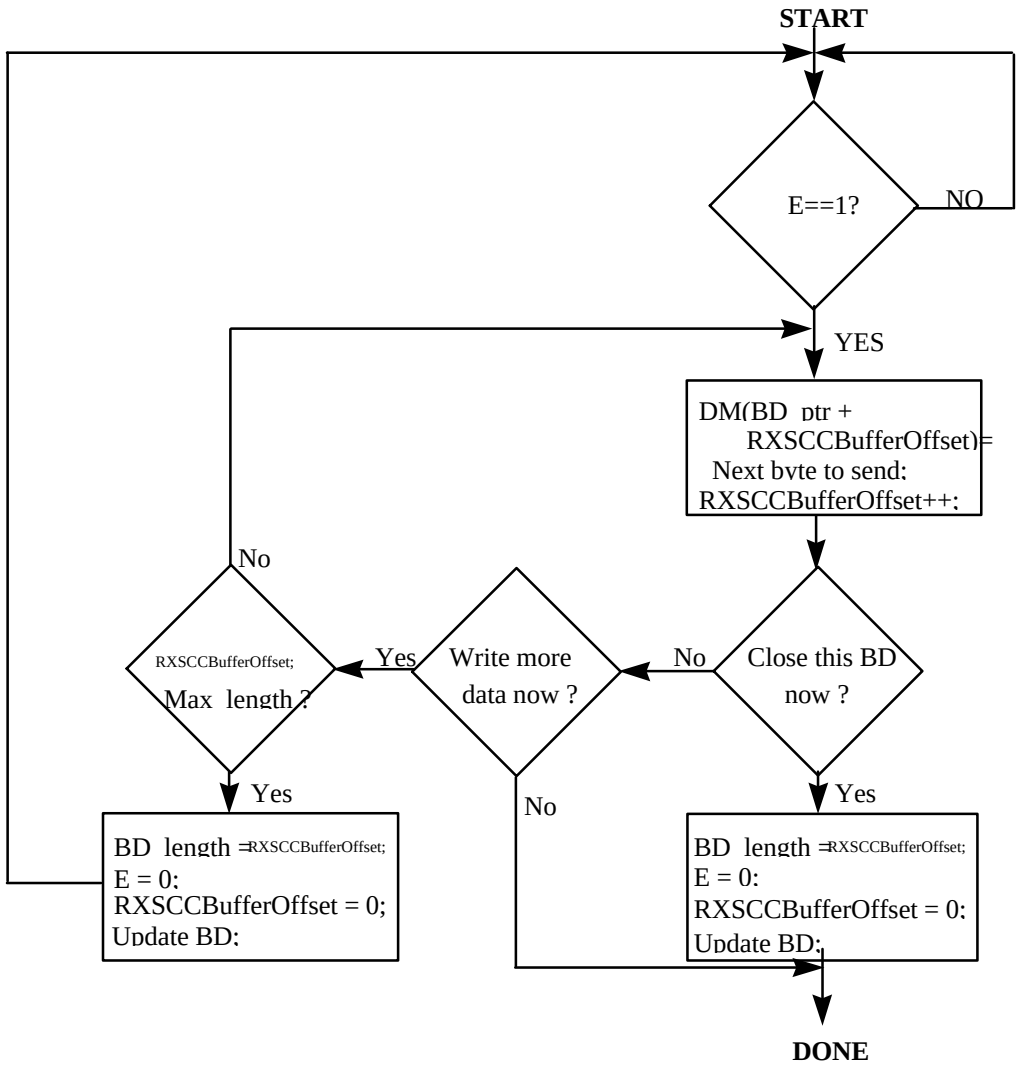


Figure 8 : flow diagram of interaction between SCC2 receiver and the IDMA host

SCC2 is directed towards the IDMA HOST, i.e. when the IDMA HOST wants to write to SCC2 (e.g. an AT-command) it will use the receive buffers in SCC2 (acting as CP)

The IDMA HOST has a variable containing currently used BufferDescriptor (BD). This variable (called BD in the flow chart) is initialised to buffer no. 0. When this variable is updated the buffer no. is incremented by 1 until it reaches 6[24], then it is set to 0 again. The IDMA HOST must always check that the E bit in the current BD is set (1) before it can start modifying the BD or its data. The DSP will set (1) this bit when it has read the data.

The sequence which the IDMA HOST should follow when writing to SCC2 is described in the flow chart in Fig.8. *(describes the case of Polling mode)*

[24] *The number of buffers is not fixed. The IDMA host should check the wrap bit in the BufferDescriptor, at least after initialisation of the SCC structure, to know the number of buffers*

Remarks:

1. *The 'start' task can be interrupt triggered by an incoming event indicating a buffer is empty*
2. *In interrupt triggered operation, checking the empty bit is allowed but not required, if the bit is not set it would indicate an error condition. Equally initiating a wait loop in not required.*
3. *test 'close this BD now' , if 'yes' it also implies you want to terminate.*
4. *test ' `write more data now' if 'no', next time the flow diagram is executed, the 'start' task should immediately jump to this test again to have error free behaviour*
5. *in case of interrupt triggered operation the test on 'Max_length', if 'yes', flow could return to the 'yes' output of the 'close BD now' test*

4.2.3.5. IDMA - SCC2 transmitter interaction

This paragraph describes a flow diagram that could be used as an example for the host driver interacting with the DTE SCC transmitter

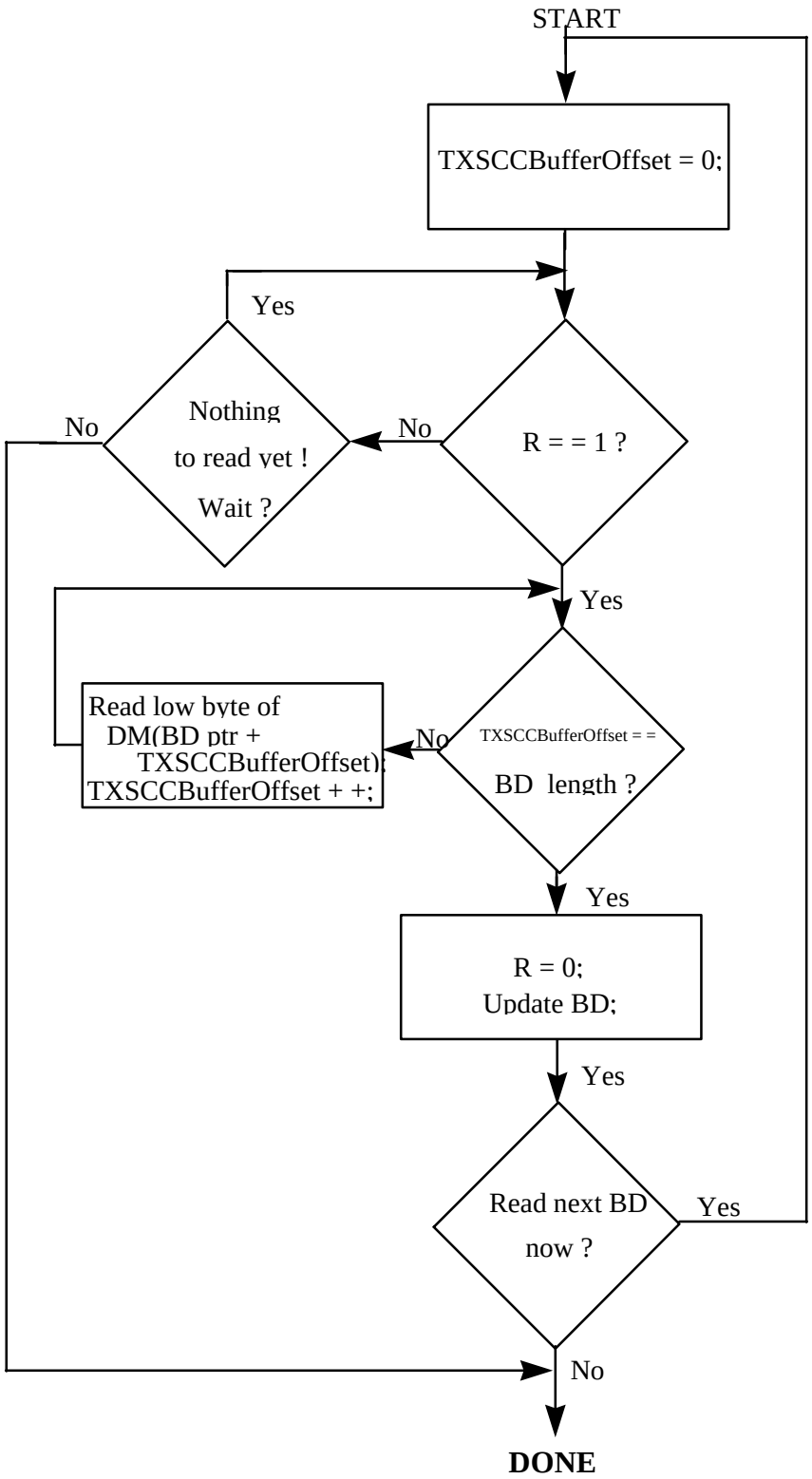


Figure 9 : flow diagram of interaction between SCC2 transmitter and the IDMA host

SCC2 is directed towards the IDMA HOST, i.e. when the IDMA HOST want to read from SCC2 (e.g. modem data without PPP) it will use the transmit buffers in SCC2 (acting as CP).
The IDMA HOST has a variable containing the current BufferDescriptor (BD) to read from. This variable (called BD in the flow chart) is initialised to buffer no. 0. When this variable is updated the buffer no. is incremented by 1 until it reaches 6[25], then it is set to 0 again. The IDMA HOST must always check that the R bit in the current BD is set (1) before it can start modifying the BD or its data.
The IDMA HOST clears (0) this bit when all data has been read. The DSP sets this bit in the BufferDescriptors that are ready to be read from the IDMA HOST.
The sequence which the IDMA HOST should follow when reading from SCC2 is described in the flow chart in Fig.3. *(describes the case of Polling mode)*

Remarks:
1. the 'START' task can be interrupt triggered by an incoming event, indicating a new buffer is full
2. in interrupt triggered operation, checking the ready bit is allowed but not required, if the bit is not set it would indicate an error condition. Equally, initiating a wait loop is not required
3. in the case of interrupt driven operation the test ' read next BufferDescriptor now' would become obsolete because part of the 'START' task.

[25] *The number of buffers is not fixed. The DP should check the wrap bit in the BufferDescriptor, at least after initialisation of the SCC structure, to know the number of buffers*

4.2.4. Asynchronous HDLC SCC

An Async HDLC module, situated in SCC3 may be activated. The functionality of the module is limited to what is required for running the async PPP protocol :

*Automatic 16-bit CRC generation and checking (CRC-CCITT)
*Automatic generation of opening and closing flags
*Reception of frames with only one “shared” flag
*Automatic generation and stripping of transparency characters according to RFC 1549 utilising transmit and receive control character maps.
*Transparent transmission of characters equal to the opening flag, closing flag or escape character.
*Automatic transmission of the ABORT sequence (escape character, closing flag) and closing of the current transmitbuffer after the STOP TRANSMIT command is issued by the user.
*Automatic transmission of the next transmitbuffer after the RESTART TRANSMIT command is issued by the user.
*Closing the current receivebuffer and waiting for the next frame after reception of the ABORT sequence.

Configuration of the Async HDLC consists of the definition of the Openflag Character, the Closingflag Character, the Escape Character, the Number of Opening Flags and the transmit and receive character maps. For all of these parameters locations are provided in the Async HDLC database.

Async HDLC database
The Baseaddress of the database can be found in the location **ACCM map address** of the modem read database.

0	TX_ ACCM : Transmit Async Control Character Map (LSW)	Initial Value FFFF	write
1	TX_ ACCM : Transmit Async Control Character Map (msw)	FFFF	write
2	RX_ ACCM : Receive Async Control Character Map (LSW)	FFFF	write
3	RX_ ACCM : Receive Async Control Character Map (msw)	FFFF	write
4	openflag_ch	7E	write
5	closingflag_ch	7E	write
6	escape_ch	7D	write
7	nmbr_of_openflags (0 : Shared Flag Option)	1	write

Table 4 : Async HDLC database

4.2.4.1. The SCC structure

the base address of the HOSTSCCstructure can be found in the location HOSTSCCstructPtr of the IDMA modem read database.

The table below gives an overview of the usage of each field of the SCC structure by the CP, which runs on the DSP, and by the IDMA HOST.

Off set				IDMA host		DSP (CP)	
				R	W	R	W
0	ModeReg				X	X	X
	F	STOP TX	if bit set SCC clears bit, sends ABORT, closes current Transmitbuffer and updates **TXSCCBuffer**		X	X	X
	E	RESTART TX	if bit set SCC clears bit and starts transmitting current Transmitbuffer		X	X	X
	D .. 4	SPARE					
	3	ENR	1: enable receiver (written by user in DATASTATE)		X	X	
	2	ENT	1 : enable transmitter (written by user in DATASTATE)		X	X	
	1, 0	SPARE					
1	ModeReg2				X	X	X
	F	TXinit	The TXBD's are initialised by the CP using the default values of MaxBufLength (= 40) and NbrofBufs (= 4). TXinit bit can be used by the user to reinitialise the TXBD's. He therefor has to disable the transmitter (ENT=0) first, then change MaxBufLength and NbrofBufs accordingly and set the TXinit and ENT bit. The TXinit bit is reset afterwards.		X	X	X
	E	RXinit	The RXBD's are initialised by the CP using the default values of MaxBufLength (= 40) and NbrofBufs (= 4). RXinit bit can be used by the user to reinitialise the RXBD's. He therefor has to disable the receiver (ENR=0) first, then change MaxBufLength and NbrofBufs accordingly and set the RXinit and ENR bit. The RXinit bit is reset afterwards.		X	X	X
	D .. 6	SPARE					
	5	Endian type	this bit should be set or reset before booting of the page or before setting the TXinit and RXinit bits. 1 : big endian for CharMode = 1 the MS-part of a DM location is accessed first, the LS-part secondly 0 : little endian (default value) for CharMode = 1 the LS-part of a DM location is accessed first, the MS-part secondly		X	X	
	4	CharMode	1: all buffers operate with two characters per location (see bit 5 Endian type for location accessing) 0: all buffers operate with one character per location (LS-part)	X			X
	3 .. 0	NbrofBufs	indicates the number of transmit buffers and receive buffers. The number of the RX and TX data buffers is initialised by the modem software to : default number = 4. This value can be over written by the user for reinitialization.	X	X	X	X
2	TXState		this location contains the state variable of the CP transmitter			X	X

3	RXState		this location contains the state variable of the CP receiver			X	X
4 .. F	RXBufferDescriptor[4*3]		Data associated with the SCC receiver is stored in buffers. Each buffer is referenced by a BufferDescriptor(BD). BD's are located in a BD array. The BD array allows to define up to four buffers for each channel.	X	X	X	X
	offset 0	STATUS AND CONTROL					
	offset 1	DATA LENGTH					
	offset 2	DATA BUFFER POINTER					
10 .. 1B	TXBufferDescriptor[4*3]		Data associated with the SCC transmitter is stored in buffers. Each buffer is referenced by a BufferDescriptor(BD). BD's are located in a BD array. The BD array allows to define up to four buffers for each channel.	X	X	X	X
	offset 0	STATUS AND CONTROL					
	offset 1	DATA LENGTH					
	offset 2	DATA BUFFER POINTER					
1C	HOSTRXBD_known		indicates whether the current RXBD has to be read			X	X
1D	HOSTTXBD_known		indicates whether the current TXBD has to be read			X	X
1E	MaxBufLength		the maximum length of an RX or TX buffer. The default size of the RX and TX data buffers is initialised by the modem software to : 40 character locations.	X	X	X	X
1F	RXSCCBuffer		indicates in which BufferDescriptor the receiver CP is writing data (from SCC2 Transmitbuffer)	X		X	X
20	RXSCCBufferOffset		indicates the number of received characters	X		X	X
21	HUNTMODE		indicates if the CP is looking for an openflag			X	X
22	Cur_RXchar		contains the current received character			X	X
23	Prev_RXchar		contains the previously received character			X	X
24	RXSCCCRC		contains the calculated CRC of all received characters			X	X
25	RXWorkReg		not used				
26	RXWorkReg1		not used				
27	RXPtr		points to current location in Rxbuffer	X		X	X
28	write_byte		indicates which byte of the current location will be written	X		X	X
29	TXSCCBuffer		indicates from which BufferDescriptor the transmitter is sending data (to the SCC2 receivebuffer)	X		X	X
2A	TXSCCBufferOffset		indicates the number of characters which aren't sent yet	X		X	X
2B	No_OpenFlags		number of openflags that still need to be sent			X	X
2C	next_TXchar_request		indicates whether a new TXchar has to be read			X	X
2D	Cur_TXchar		contains the current character to be transmitted			X	X
2E	Transp_TXchar		exored Cur_TXchar			X	X
2F	TXSCCCRC		contains the calculated 16-bit CRC of all transmitted characters			X	X
30	TXWorkReg		not used				
31	TXWorkReg1		not used				
32	TXPtr		points to current DM word (2 characters per word)	X		X	X
33	read_byte		indicates which byte of the current location will be used for Cur_TXchar	X		X	X

Data associated with this Async HDLC SCC is stored in buffers. Each buffer is referenced by a BufferDescriptor(BD). 4 ReceiveBD's and 4 TransmitBD's are provided to access the 4 Receivebuffers and 4 Transmitbuffers. The Receivebuffers are used for data received from the SCC2 Transmitbuffers; the Transmitbuffers are used to transmit data to the SCC2 Receivebuffers. Each SCC3 buffer is 20 words (1 word = 16 bits) in size but can contain a maximum of 40 characters. This is achieved by using the LS- ***and*** MS-byte of every buffer word.

The user therefore has to start writing the first character in the LS-byte of the first Transmitbufferword when he selects little endian (bit 5 of ModeReg2). The second character has to be written in the MS-byte of the same word. The third character has to be written in the LS-byte of the second Transmitbufferword, etc.
The same is true for reading a Receivebufferword. The LS-byte contains the character received first from the SCC2. The MS-byte contains the next received character.
When big endian is selected writing and reading starts with the MS byte of the memory location.

Data transfer between SCC3 and SCC2 is achieved by enabling the Async HDLC framer receiver and transmitter. Characters are transmitted at symbolrate from the SCC3 Transmitbuffer to the SCC2 Receivebuffer when the current SCC3 Transmitbuffer is Ready and the current SCC2 Receivebuffer is Empty.

When the SCC3 Receivebuffer is Empty and the SCC2 Transmitbuffer is Ready, characters from the SCC2 Transmitbuffer are received by the SCC3 Receivebuffer.

Bit 15 equal to 1 can be used by the Host to issue a STOP TRANSMIT command. This will transmit the ABORT sequence, which is equal to the escape character followed by a closing flag. The current SCC3 Transmitbuffer will be closed, **TXSCCBuffer** will be updated, bit 15 is cleared and a waitstate is started for the RESTART TRANSMIT command from the user.

Bit 14 equal to 1, the RESTART TRANSMIT command, is expected after the STOP TRANSMIT command. When this restart is issued by the user, bit 14 is cleared and the transmitter will resume transmission of the current SCC3 Transmitbuffer (indicated by **TXSCCBuffer**) as being the first buffer in the frame.

Every frame is preceded by a sequence of openflags, specified by the dbs variable NMBR_OF_OPENFLAGS, and ended by one closing flag. When NMBR_OF_OPENFLAGS is equal to 0 (shared flag option) the closing flag will open and close the frame.
In the exceptional case of startup and NMBR_OF_OPENFLAGS equal to 0 (shared flag option), one openflag is transmitted.

Bits 2 and 3 are only written by the user and read by the CP. Bits 14 and 15 are written by the user and CP. All other bits are not used.

The maximum number and size of the buffers are specified below. These are also the default values. When the User wants to change these settings he can use the TXinit and RXinit bits of the ModeReg2.

SCC side	#buffers	max number	length(words :2 bytes/location)
HOST	4	4	40

4.2.4.2. async HDLC receive BufferDescriptor (RX bd)

The Async HDLC controller uses the RX BD to report information about the received data for each buffer. The RX BD is shown in figure …

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET +0	E	-	W	-	L	-	-	-	-	-	-	-	AB	CR	-	-
OFFSET +2	DATA LENGTH															
OFFSET +4	RX BUFFER POINTER															

Table 5 : Async HDLC Receive BufferDescriptor

The first word of the RX BD contains control and status bits. Bit 13 is written by the CP when the buffer is linked to the RX BD table, and bits 11,2 and 3 are set by the CP following frame reception. Bit 15 is set by the user when the buffer is available to the ASYNC HDLC controller; it is cleared by the ASYNC HDLC controller when the buffer is full.

E - Empty

0 = The data buffer associated with this BD has been filled with received data, or data reception has been aborted due to an ABORT sequence. The user is free to examine or write to any fields of the BD.
1 = The data buffer associated with this BD is empty. This bit signifies that the BD and its associated buffer are available to the Async HDLC controller. The user should not write to any fields of this BD after it sets this bit. The empty bit will remain set while the Async HDLC controller is currently filling the buffer with received data.

W - Wrap (Final BD in Table)

0 = This is not the last BD in the RX BD table.
1 = This is the last BD in the RX BD table. After this buffer has been used, the Async HDLC controller will receive incoming data into the first BD in the table.

The following status bits are written by the Async HDLC controller after the received data has been placed into the associated data buffer.

L - Last in Frame

This bit is set by the Async HDLC controller when this buffer is the last in a frame. This implies the reception of a closing flag or an ABORT sequence. The Async HDLC controller will write the number of characters written in the Receivebuffer to the data length field.

0 = This buffer is not the last in a frame.
1 = This buffer is the last in a frame.

AB - RX Abort Sequence
An ABORT sequence was received during frame reception.

CR - RX CRC Error
This frame contains a CRC error.

Data Length
The data length is the number of characters (including two bytes for CRC) written to this BD's data buffer by the Async HDLC controller, even when this BD is the last BD in the frame (L=1).

RX Buffer Pointer
The receive buffer pointer, which always points to the first location of the associated data buffer, may reside in either internal or external memory.

An example of the Async HDLC receive process is shown in figure … This shows the resulting state of the RX BDs after receipt of a complete frame spanning two receive buffers, a second frame with an abort sequence and a few information characters of a third frame (see results of the transmit process example).

Initially the Async HDLC receiver will read the characters from the SCC2 Transmitbuffer and search for one or more openflags (PPP; 0x7E).

The information characters from the current SCC2 Transmitbuffer are compared to 0x20 and the escape character (PPP; 0x7D). When smaller than 0x20 the received character is discarded only if the bit in the Receive Map RX_ACCM (PPP; 0xFFFFFFFF) with index equal to the character value is set (information character 08 in the example). In case of an escape character, the next received character will be exored with 0x20 and written to the current Receivebufferword.

When the closingflag (PPP; 0x7E) is encountered in the SCC2 Transmitbuffer, the CRC -bit of the current RXBD is set, if the incorrect CRC characters are received.

When the current Receivebuffer is filled or when the second CRC byte is written in the Receivebuffer, the CP closes the latter and updates **RXSCCBuffer** (incremented if W-bit of current buffer is 0, otherwise = 0). The CP will enter the **RXState** WaitforHOSTRXBD, in which the Empty bit of the current Receivebuffer is checked. When the user has read this buffer, the Empty bit must be set so this buffer can be used by the CP to receive characters from the SCC2 Transmitbuffer.

When the CP has finished reading the current SCC2 Transmitbuffer, it closes the latter and **TXSCCBuffer** (of the SCC2 structure) is updated. The **RXState** WaitforDTETXBD is entered, waiting for the Ready bit of the current Transmitbuffer to be set by the protocol.

When the CP encounters an ABORT sequence in the SCC2 Transmitbuffers, the CP closes the current Receivebuffer, sets the AB- and L-bit in the current RXBD and updates **RXSCCBuffer**. The **RXState** will become WaitforHOSTRXBD. As soon as the current Receivebuffer becomes Empty, the CP will start searching the next openflag in the current SCC2 Transmitbuffer.

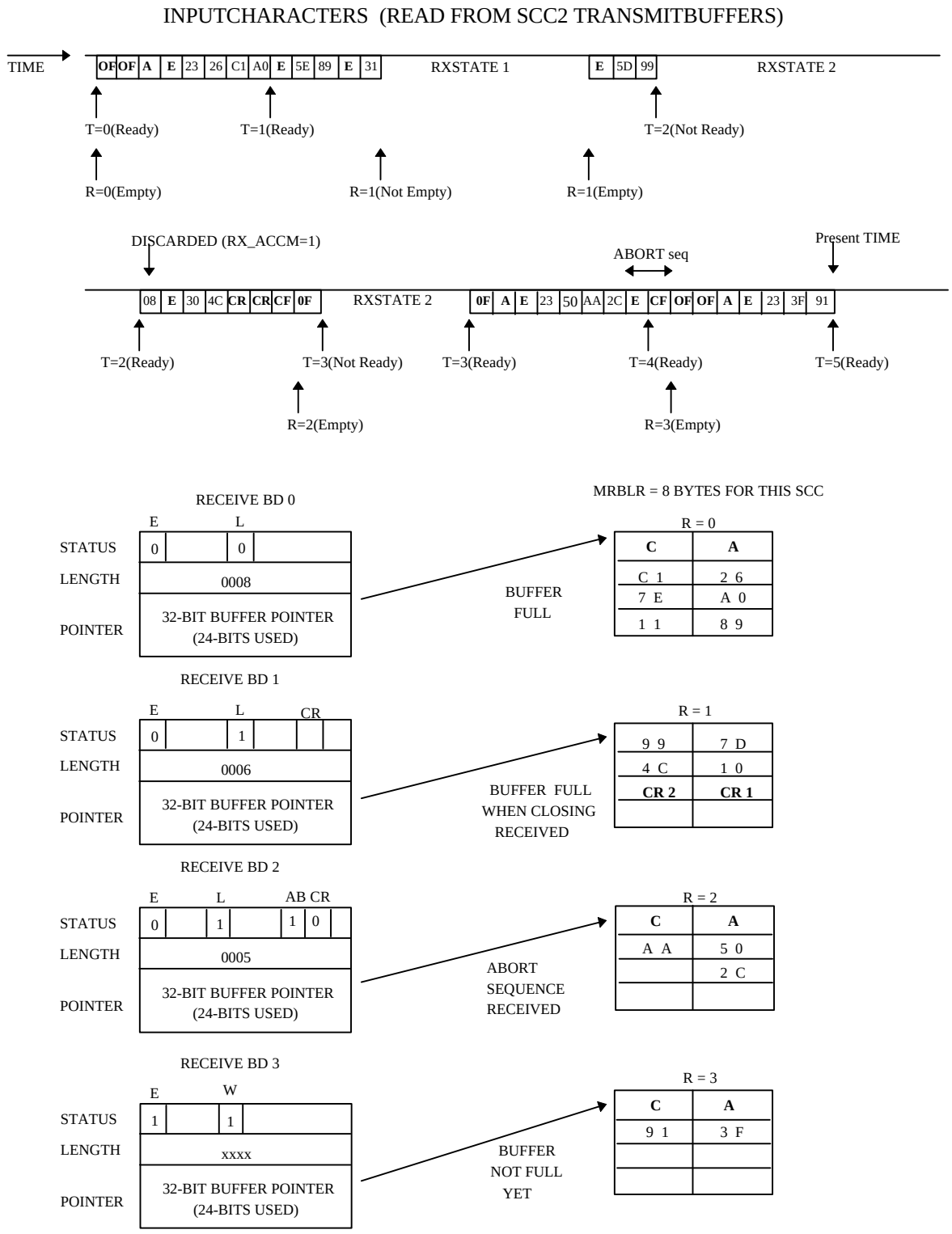


Figure 10 : Async HDLC Receive BD Example (little endian)

LEGEND

OF = OPENFLAG
CF = CLOSING FLAG
A = ADDRESS BYTE
C = CONTROL BYTE (PPP;0x03)
E = ESCAPE CHARACTER
CR = CRC BYTE

RX_ACCM = 0xFFFFFFFF
T = SCC2 TXSCCBuffer
R = SCC3 RXSCCBuffer
RXState 1 =WaitforHOSTRXBD
RXState 2 =WaitforDTETXBD

Data is presented to the Async HDLC controller for transmission to the SCC2 by arranging it in buffers referenced by the channel's TX BD table. The Async HDLC controller confirms transmission using the BDs to inform the user that the buffers have been serviced. The TX BD is shown in figure …

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET +0	R	-	W	-	L	-	-	-	-	-	-	-	-	-	-	-
OFFSET +2	DATA LENGTH															
OFFSET +4	TX BUFFER POINTER															

Figure 11 : ASYNC HDLC Transmit BufferDescriptor

The first word of the TX BD contains status and control bits. Bits 15 and 11 are prepared by the user before transmission. Bit 13 is written by the CP when the buffer is linked to the TX BD table. Bit 15 is set by the user when the buffer and BD have been prepared and is cleared by the Async HDLC controller after the frame has been transmitted.

R - Ready

0 = This buffer is not currently ready for transmission. The user is free to manipulate this BD (or its associated buffer). The Async HDLC controller clears this bit after the buffer has been fully transmitted or after the user has issued a STOP TRANSMIT command.
1 = The data buffer, which has been prepared for transmission by the user, has not yet been transmitted. No fields of this BD may be written by the user once this bit is set.

W - Wrap (Final BD in Table)

0 = This is not the last BD in the TX BD table.
1 = This is the last BD in the TX BD table. After this buffer has been used, the Async HDLC controller will transmit data from the first BD in the table.

L - Last

0 = This is not the last buffer in the frame.
1 = This is the last buffer in the current frame.

Bits 10 - 0 - Reserved for future use.

An example of the HDLC transmit process is shown in figure … This shows the resulting state of the TX BDs after transmission of a complete frame spanning two transmit buffers, a second frame with a STOP TRANSMIT command issued by the user and transmission of the next Transmitbuffer containing a new frame after the RESTART TRANSMISSION command of the user. The example assumes that MaxBufLength = 8 for both the SCC3 and SCC2 buffers and NMBR_OF_OPENFLAGS = 2.

Transmitbuffer 0 (**TXSCCBuffer**=0) is transmitted as the first buffer in a frame. Therefore two openflags are sent to the SCC2 Receivebuffer. When NMBR_OF_OPENFLAGS equals 0 no openflags will be succeeding the closingflag of the previous frame (Shared Flag Option).

Transmitting information characters starts with the address character (PPP; 0xFF) from the LS-byte and the following control character (PPP; 0x03) from the MS-byte of the first Transmitbufferword.
If one of the information characters in the Transmitbuffer is smaller than 0x20 and the bit in the Transmit Map TX_ACCM with index equal to the characters value is set, the escape character (PPP; 0x7D) is transmitted first. The information character, exored with 0x20 is transmitted next. If the information character is equal to the openflag (PPP; 0x7E), closingflag (PPP; 0x7E) or escape character the same action is performed.

The example assumes that after filling the SCC2 Receivebuffer 0, SCC2 Receivebuffer 1 is not yet Empty. The CP enters a **TXState** called WaitforDTERXBD. This state will end as soon as the Receivebuffer is made available by the protocol (setting the Empty bit of the RXBD 1). When Transmitbuffer 0 is sent the CP clears the R-bit and checks the L-bit of the TXBD 0 statusword. In this case Transmitbuffer 0 is not the last buffer in the current frame, so Transmitbuffer 1 (**TXSCCBuffer**=1) will be the current Transmitbuffer. The example assumes that this Transmitbuffer 1 is not ready for transmission. The CP enters the **TXState** called WaitforHOSTTXBD. When the user has prepared Transmitbuffer 1 and set its Ready bit, the CP will leave the current TXState and start transmitting the information characters of the Transmitbuffer 1. Since this buffer is the last of the current frame, two 8 bit CRC characters are added as information characters and a closingflag is used to close the current frame.

When a STOP TRANSMIT command is issued by the user, the CP will add the ABORT sequence to the already transmitted information characters from the Transmitbuffer 2, close the latter, update **TXSCCBuffer** (incremented if W-bit of current buffer is 0, otherwise = 0. This case: +1), clear the STOP TX bit of the **Moderegister** and return to the **TXState** WaitforHOSTTXBD. In this waitloop the CP checks continuously the RESTART TRANSMIT bit of the **Moderegister**. When set, the CP will stay in the same **TXState**, checking the Ready bit of the current Transmitbuffer (**TXSCCBuffer**=3).

After the RESTART TX command the CP will start transmitting the current Transmitbuffer as being the first of the frame, i.e. transmitting at least 1 openflag. These characters will be transmitted right after the ABORT sequence.

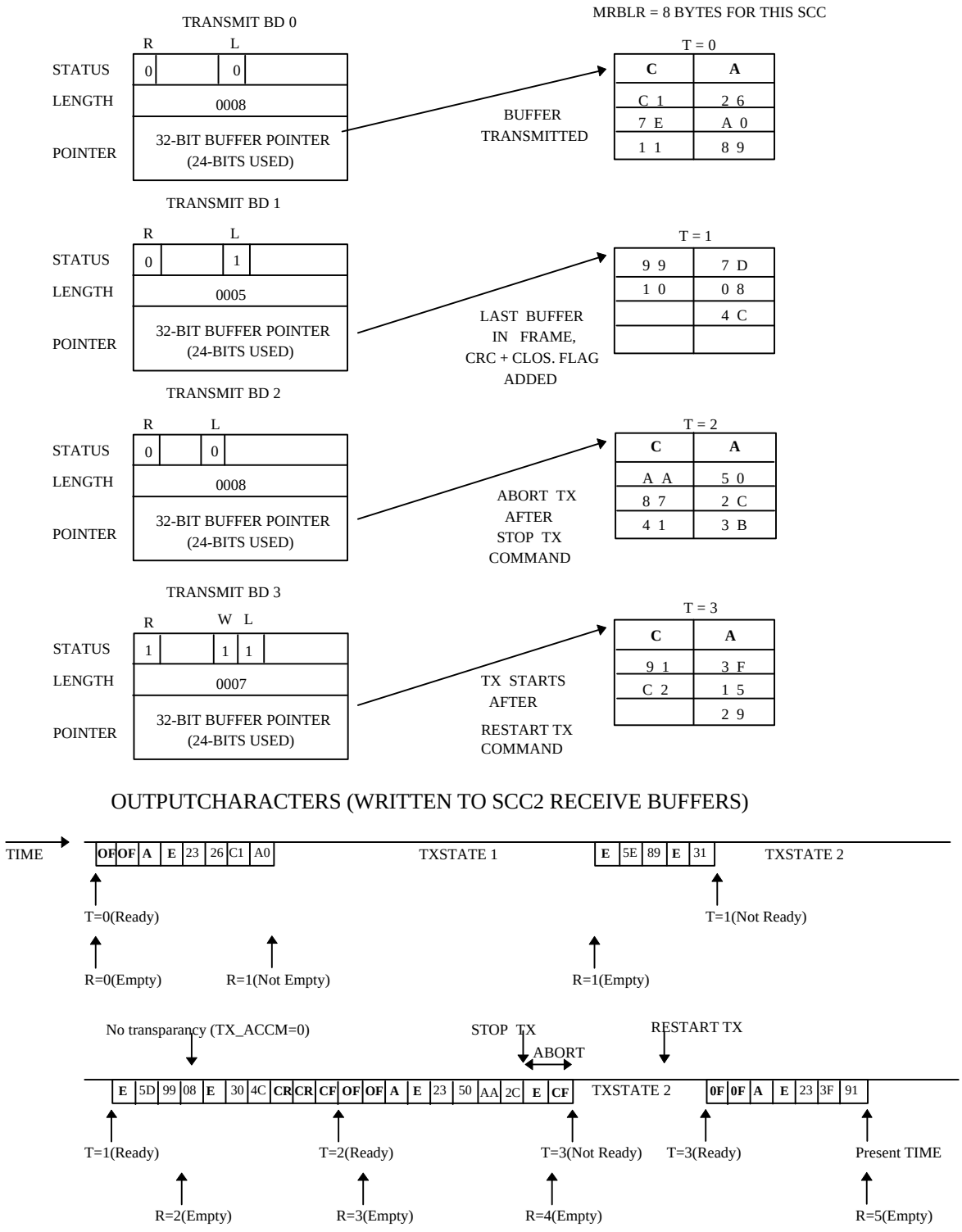


Figure 12 : Async HDLC Transmit BD Example (little endian)

LEGEND

OF = OPENFLAG
CF = CLOSING FLAG
A = ADDRESS BYTE
C = CONTROL BYTE (PPP;0x03)
E = ESCAPE CHARACTER
TX_ACCM = 0xFFFFF*E*FF
T = SCC3 TXSCCBuffer
R = SCC2 RXSCCBuffer
TXState 1 =WaitforDTERXBD
TXState 2 =WaitforHOSTTXBD

CR = CRC BYTE

E + CF = ABORT sequence

4.2.4.4. IDMA - SCC3 transmitter interaction

This paragraph describes a flow diagram that could be used as an example for the host driver interacting with the HOST SCC transmitter

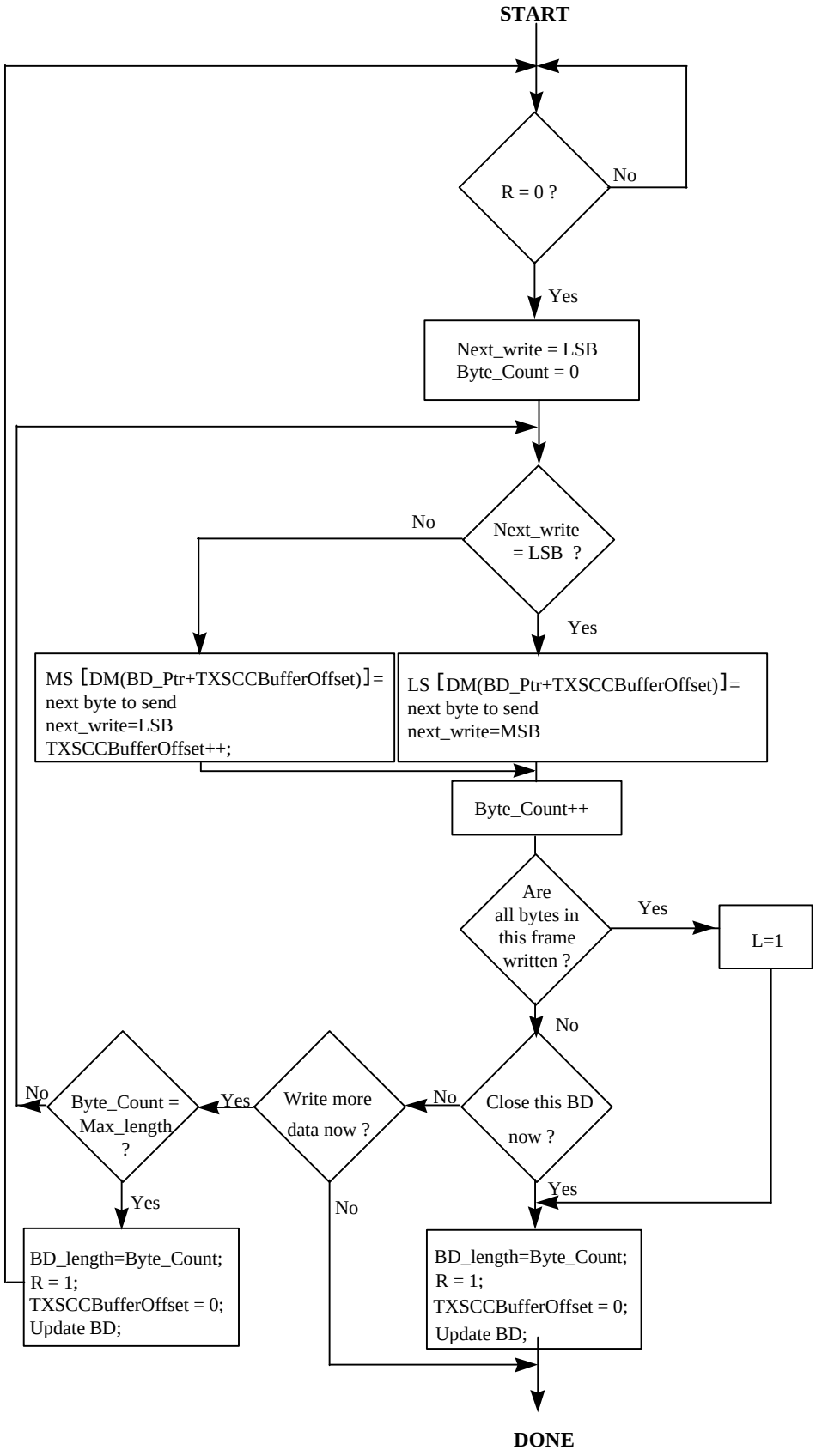


Figure 13 : IDMA - SCC3 transmitter interaction (little endian)

SCC3 is directed towards the modem firmware, i.e. when the IDMA HOST wants to write to SCC3 (e.g. modem data with PPP) it will use the write buffers in SCC3.
The IDMA HOST has a variable containing the current BufferDescriptor (BD) to write to. This variable (called BD in the flow chart) is initialised to buffer no. 0. When this variable is updated the buffer no. is incremented with 1 until it reaches 4, then it is set to 0 again. The IDMA HOST must always check that the R bit in the current BD is cleared (0) before it can start modifying the BD or its data. The IDMA HOST sets (1) this bit when all data has been written to this BD. The DSP clears this bit in the BufferDescriptors that are ready to be filled by the IDMA HOST.
When a complete PPP frame has been written to the DSP the IDMA HOST sets (1) the L bit in the BufferDescriptor containing the last data.

The sequence which the IDMA HOST should follow when writing to SCC3 is described in the flow chart in Fig.5. (describes the case of Polling mode)

Remarks:
1. The ‘start’ task can be interrupt triggered by an incoming event indicating a buffer is empty
2. in interrupt triggered operation, checking the ready bit is allowed but not required, if the bit is set it would indicate an error condition. Equally initiating a wait loop in not required.
3. test ‘close this BD now’ :on what basis will the IDMA HOST decide to close the current buffer?
4. test ‘ `write more data now’ if ‘no’, I believe next time the flow diagram is executed, the ‘start’ task should immediately jump to this test again to have error free behaviour
5. in case of interrupt triggered operation the test on ‘Max_length’, if ‘yes’, flow should best return to the ‘yes’ output of the ‘close BD now’ test

4.2.4.5. IDMA- SCC3 receiver interaction

This paragraph describes a flow diagram that could be used as an example for the host driver interacting with the HOST SCC receiver

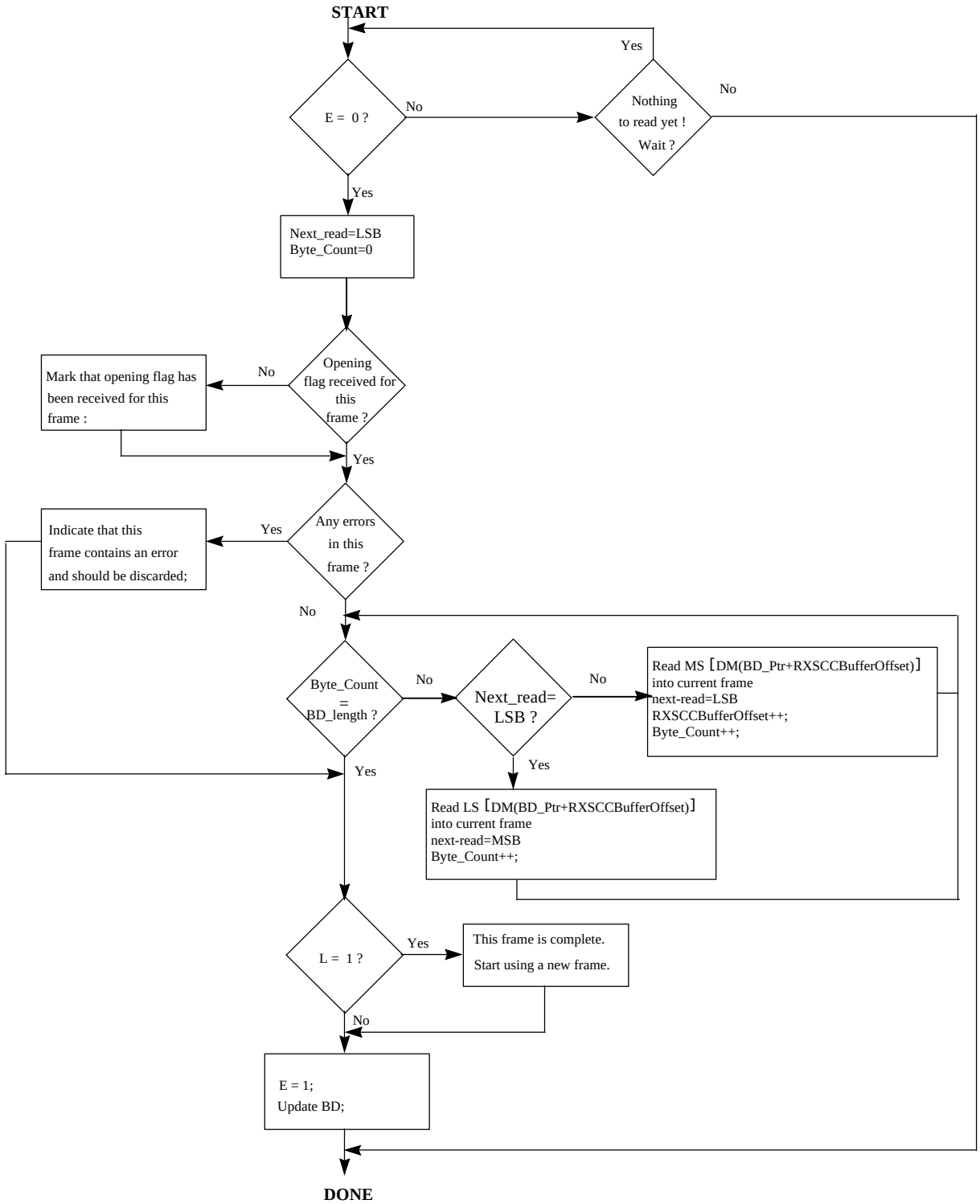


Figure 14 : IDMA - SCC3 receiver interaction (little endian)

SCC3 is directed towards the modem firmware, i.e. when the IDMA HOST wants to read from SCC3 (e.g. modem data with PPP) it will use the read buffers in SCC3.
The IDMA HOST has a variable containing the current BufferDescriptor (BD) to read from. This variable (called BD in the flow chart) is initiated to buffer no. 0. When this variable is updated the buffer no. is increased with 1 until it reaches 4[26], then it is set to 0 again. The IDMA HOST must always check that the E bit in the current BD is cleared (0) before it can start modifying the BD or its data. The IDMA HOST sets (1) this bit when all data has been read. The DSP clears this bit in the BufferDescriptors that are ready to be read from the IDMA HOST.

The IDMA HOST reads all data that belong to the same PPP frame into a buffer. Only when a complete PPP frame has been received it is send towards the application.
The sequence which the IDMA HOST should follow when reading from SCC3 is described in the flow chart in Fig.4. (describes the case of Polling mode)

Remarks:
1. the 'START' task can be interrupt triggered by an incoming event, indicating a new buffer is full
2. in interrupt triggered operation, checking the empty bit is allowed but not required, if the bit is set it would indicate an error condition. equally initiating a wait loop is not required
3. an opening flag is not essential for the correct operation and is not supported, likewise the test on the opening flag becomes obsolete

[26] *see previous remarks about the number of buffers*

4.3. CONTROL INTERFACE

4.3.1. AT set control interface

All functions of the modem software are controlled by **AT ® or ITU-T V.25ter** compatible commands. The **buffering system of SCC2** is used for the HOST processor to enter command lines and the DSP to return result codes.

4.3.1.1. Command Language Syntax

The command interpreter analyses and executes Hayes commands and sends result codes in optional English word or decimal digit form.
Command lines must begin with the ASCII characters “AT” or “at” with the exception of the A/ command. The attention code signals the modem that one or more commands are to follow.
Multiple commands may appear on a single command line and may be separated by space characters to improve readability.

Format : AT_____<carriage return>
?_____ one or more commands

A command line must end in the ASCII carriage return or linefeed character. The commands following the AT are not processed until the modem receives the carriage return character. A command line must not exceed 80 characters.
The previously issued command may be re-executed by sending A/ to the modem. The A/ command is the only command that is not preceded by the command prefix AT, nor followed by a carriage return.

4.3.1.2. General at_set commands

A/	Re-execute last command	A/ is the only command which is not to be preceded by AT.	✓
+++	Escape code sequence	Forces the modem in command mode. Additional AT-commands can then be entered.	✓
A	Go off-hook in answer mode	Causes the modem to immediately go off-hook (on-line) in answer mode regardless of the contents of register S0. This is used to manually answer an incoming call.	✓
Bn	Select modulation type or channel	This command should be used together with the Fn command. It can only be used in the on-hook state.	
	B0	Select ITU-T modulations.	✓
	B1	Select Bell modulations.	✓
	B254	The modem uses the call channel to answer an incoming call.	✓
	B255	The modem uses the answer channel to answer an incoming call (default).	✓
Dn	Dial telephone number	The dial command causes a telephone number to be dialled. 'n' represents a dial string composed of dial characters and dial modifiers. The dial characters include the decimal values 0 through 9 plus the letters A, B, C, D, and the symbols '*' and '#'. The AT-set dial modifiers are listed in Table 6 - AT-set Dial Modifiers. ATD without dial string puts the modem on line in call mode, without dialling.	✓
En	Echo command	The E command controls the echoing of characters during command state.	✓
	E0	Echo disabled	✓
	E1	Echo enabled (default setting)	✓
Fn	PSTN speed selection	Sets the PSTN speed. This command should only be used in off-hook state. See also ATBn.	
	N = 0 1 2 3 4 5 6 7 8 9 10 11 14 15 16	auto V.8 V21/Bell 103 V.22 600 videotex V.22/Bell212A 1200 V.22bis 2400 V.32 4800 V.32b 7200 V.32 9600 V.32T 9600 V.32b 12000 V.32b 14400 V.34/V.34bis auto V.90	✓
H	Hang up command	If you entered the command state by issuing the escape sequence (+++), you can force a modem to hang up by issuing the H command	✓
In	Request product code and ROM checksum		
	I0	The modem sends the product code as a ASCII string.	✓
	I1	The modem sends the product code as a ASCII string.	✓

	I2	The modem sends the product code as a ASCII string.	✓
	I3	The modem sends the product code as a ASCII string.	✓
On	Return to on-line state	If you entered the command state by issuing the escape sequence (+++), you can return to the on-line state by issuing the O command.	✓
Q	Quiet command	The Q command controls the issuing of result codes sent to acknowledge AT commands and call status events (e.g. OK, ERROR, RING).	
	Q0	Enable result codes	✓
	Q1	Disable result codes	✓
Sn=	Writing to S-Registers	Modify the contents of the S-registers. Refer to the following paragraph for more information. Format: Sn=x where n is a decimal integer from 0 to 30, and the value of x depends on the selected register.	✓
Sn?	Reading S-Registers	Display the contents of S-register n. Refer to the following paragraph for more information. Format: Sn? Where n is a decimal integer from 0 to 30.	✓
Vn	Enable short-form result codes	Table 7 - Result Codes, lists all possible result codes	
	V0	Selects the short-form result codes (one or two decimal digits). This can be useful when the modem is to be controlled by a program resident on the data terminal.	✓
	V1	Long form result codes (verbose).	✓
Wn	Define type of returned message	See Table 8 - Connect string interpretation	
	W0	modem returns negotiation progress message “CONNECT DTE speed”	✓
	W1	modem returns negotiation progress messages “CARRIER DCE speed” “PROTOCOL : protocol” “COMPRESSION : type” “CONNECT DTE speed”	✓
	W2	modem returns link speed “CONNECT DCE speed”	✓
	W3	Identical to ATW0 + indication reliable link (MNP or V.42) “CONNECT DTE speed / REL”	✓
Xn	Enable extended result code set		
	X0	Enables result codes 0 through 4, the modem ignores dial tone and BUSY signals, if allowed by the local PTT (blind dialling) .	✓
	X1	Enables all result codes except 6 and 7, the modem ignores dial tone and BUSY signals, if allowed by the local PTT (blind dialling).	✓
	X2	Enables all result codes except 7, the modem ignores busy signals, if allowed by the local PTT.	✓
	X3	Enables all result codes except 6, the modem ignores dial tone if allowed by the local PTT.	✓
	X4	Enables all result codes (default).	✓
Zn	Reset AT-set		

	Z0	Resets a number of parameters to their default value. These default values are : AT Q0 V1 W0 X4 E1 S3 = 13 S4 = 10 S5 = 8 S6 = * S7 = * S8 = * S10 = * S14 = 8 S30 = * * = country dependent This command also loads the modem database with the factory default.	✓
	Zn	Selects the user profile n (n=1 … 4). If this profile is different from the current user profile, the new profile will be loaded into the on-line database (see AT&Wn) and the command is followed by a warm reset.	
	Z255	Selects the power up profile as the active profile (cfr also AT&Yn).	
&Bn	Bits/character		
	&Bn	Defines the number of bits per character (n=8, 9, 10, 11) for asynchronous transparent operation.	
&Cn	DCD control		
	&C0	DCD always on	✓
	&C1	DCD normal	✓
	&C2	DCD ACU on	✓
&Dn	DTR control		
	&D0	DTR control 108/2 intern	✓
	&D1	DTR control 108/2 external	✓
	&D2	DTR control 108/2 external	
	&D3	DTR control 108/2 external	
&F	Initialise modem	modem initialises with Hayes default (= factory default with exceptions : DATA MODE = asynchronous, ERROR CORR PROTOCOL = auto, COMPRESSION = V42bis)	
&Kn	Flow control		
	&K0	DTE flow control disabled DCE flow control disabled	
	&K1	DTE flow control CTS DCE flow control disabled	
	&K3	DTE flow control RTS/CTS DCE flow control disabled	
	&K4	DTE flow control Xon/Xoff DCE flow control disabled	
	&K5	DTE flow control Xon/Xoff DCE flow control Xon/Xoff	
&On	Overspeed		
	&O0	Basic overspeed	
	&O1	Extended overspeed	
&Qn	Data mode		
	&Q0	Asynchronous data mode	
	&Q1	Synchronous data mode	
&Rn	CTS delay		
	&R0	CTS delay = 0 ms	
	&R1	CTS always on	
&Sn	DSR control		
	&S0	DSR always on	
	&S2	DSR ctrl in test	
	&S3	DSR on in test	
	&S8	DSR ctrl by DTR	

&Tn	modem diagnostic tests		
	&T0	Terminates any test in progress and displays error counter if error test was active	
	&T1	Local Analog Loopback (AL)	✓
	&T3	Local Digital Loopback (DL)	
	&T6	Remote Digital Loopback (RL)	
	&T7	Remote Digital Loopback (RL) with error test	
	&T8	Local Analog Loopback (AL) with error test	
		Note : on earlier releases, not all diagnostic test functionality may be implemented.	
\Ln	Compression	See also AT%C	
	\L4	Compression disabled	✓
	\L5	MNP5 compression	✓
\Nn	Protocol		
	\N0	Buffered mode	✓
	\N1	Transparent mode	✓
	\N2	Protocol MNP and fallb to buf disabled	✓
	\N3	Protocol MNP and fallb to buf enabled	✓
	\N4	Protocol V42 LAP-M and fallb to buf disabled	✓
	\N5	Protocol V42 LAP-M and fallb to buf enabled	✓
	\N6	Protocol auto and fallb to buf disabled	✓
	\N7	Protocol auto and fallb to buf enabled	✓
\Qn	DTE flow control		
	\Q0	DTE flow control disabled	
	\Q1	DTE flow control Xon/Xoff	
	\Q2	DTE flow control CTS	
	\Q3	DTE flow control RTS/CTS	
\Xn	DCE flow control		
	\X0	DCE flow control disabled	
	\X2	DCE flow control Xon/Xoff	
%Cn	Data compression type		
	%C0	no data compression	✓
	%C1	MNP 5 compression	✓
	%C2	V42bis compression with automatic switching to MNP 5 if the called modem is not equipped with V42bis.	✓
-Jn	ODP/ADP		
	-J0	ODP/ADP disabled	✓
	-J1	ODP/ADP enabled	✓

Dial ModifierDescription

T Dialling to be continued in DTMF mode.

W Wait for dial tone before processing the next character in the dial string.

, Wait for PABX dial tone before resuming dialling

or

Insert pause before resuming dialling.
(refer to S8)

$ Wait for second dial tone.

@ Wait for quiet answer: the modem will wait for a silence of 5 seconds for a time specified in register S7 before processing the next symbol in the dial string.

;	Return to command state after dialling.
?	Delimiter between number to be dialled (P#) and PSTN security access password.
!	Hookflash; Off-hook relay closed during 90 ms.
+	Insert shortkey

Table 6 - AT-set Dial Modifiers

V1: Long-form	V0: Short-form
OK	0
CONNECT	1
RING	2
NO CARRIER	3
ERROR	4
CONNECT 1.2	5
NO DIAL TONE	6
BUSY	7
NO ANSWER	8
CONNECT 600	9
CONNECT 2.4	10
CONNECT 4.8	11
CONNECT 9.6	12
CONNECT 14.4	13
CONNECT 19.2	14
CONNECT 7.2	15
CONNECT 12.0	16
CONNECT 16.8	17
CONNECT 38.4	18
CONNECT 57.6	19
CONNECT 115.2	21
CONNECT 1,2/75	22
CONNECT 75/1.2	23
CONNECT 21.6	24
CONNECT 24.0	25
CONNECT 26.4	26
CONNECT 28.8	27
CONNECT 31.2	28
CONNECT 33.6	29
CARRIER 300	40
CARRIER 600	41
CARRIER 1.2/75	44
CARRIER 75/1.2	45
CARRIER 1.2	46
CARRIER 2.4	47
CARRIER 4.8	48
CARRIER 7.2	49
CARRIER 9.6	50
CARRIER 12.0	51
CARRIER 14.4	52
CARRIER 16.8	53
CARRIER 19.2	54
CARRIER 21.6	55
CARRIER 24.0	56
CARRIER 26.4	57
CARRIER 28.8	58
CARRIER 31.2	59
CARRIER 33.6	60
COMPRESSION : CLASS5	66
COMPRESSION : V42BIS	67
COMPRESSION : NONE	69
SYNC	70
PROTOCOL : TRANSPARENT	71
PROTOCOL : BUFFERED	72
PROTOCOL : LAP-M	77
PROTOCOL : ALT	80

CARRIER	28.0	90
CARRIER	29.3	91
CARRIER	30.7	92
CARRIER	32.0	93
CARRIER	33.3	94
CARRIER	34.7	95
CARRIER	36.0	96
CARRIER	37.3	97
CARRIER	38.7	98
CARRIER	40.0	99
CARRIER	31.3	100
CARRIER	32.7	101
CARRIER	44.0	102
CARRIER	45.3	103
CARRIER	46.7	104
CARRIER	48.0	105
CARRIER	49.3	106
CARRIER	50.7	107
CARRIER	52.0	108
CARRIER	53.3	109
CARRIER	54.7	110
CARRIER	56.0	111

Table 7 - Result Codes

- The ‘CONNECT’ and ‘CARRIER’ result codes give the DTE speed (resp DCE speed) in bit/s.

- The CARRIER result codes have following meaning :
  CARRIER ‘line Rx speed’ / ‘line Tx Speed’
  In case of split speed operation the ‘CARRIER’ short form result code corresponds to the receiver speed.

- The protocol and compression result codes have the following meaning:

SYNC	for synchronous transmission
PROTOCOL : TRANSPARENT	for asynchronous transparent transmission
PROTOCOL : BUFFERED	for asynchronous buffered transmission
PROTOCOL : LAP-M	for error corrected transmission with V.42 LAPM protocol
PROTOCOL : ALT	for error corrected transmission with MNP protocol
COMPRESSION : CLASS5	for compressed transmission with MNP5 protocol
COMPRESSION : V42BIS	for compressed transmission with V.42bis protocol
COMPRESSION : NONE	no compression

Table 8 - Connect string interpretation

4.3.1.3. S-Registers

S0	Rings to answer on	Determines the number of rings before taking the line. Assigning a value from 1 to 255 places the modem in auto-answer mode. Setting S0 to 0 disables auto-answer mode.	✓
S1	Ring count	S1 is incremented each time the modem detects a ring signal on the telephone line. It is cleared if no rings occur over any eight second interval.	✓
S2	Escape code character	S2 holds the ASCII value of the escape code. The default value is 43 (ASCII "+"). S2 can be set to any value from 0 to 255. Values greater than 127 will disable escape code detection. To return to the command state when the escape code is disabled, the distant modem must hang up (local modem looses carrier).	✓
S3	Carriage return character	S3 holds the ASCII value of the carriage return character. Normally a value of 13 is used. If the data terminal equipment is non-standard, a different value can be used. This character serves as the command line terminator and the result code terminator.	✓
S4	Line feed character	S4 holds the ASCII value of the line feed character. The default value is 10.	✓
S5	Back space character	S5 holds the ASCII value of the back space control character. The default value is 8. The backspace character must not be set to a value corresponding to a printable ASCII character (i.e. between 33 and 126) or to a value greater than 127.	✓
S6	Wait for dial tone before blind dialling	This is a country dependent parameter that can not be modified. It can only be read.	✓
S7	Wait for carrier after dialling	This register gives the answer tone and silence timeout during dialling before hanging up. The default value of this register is country dependent. In some countries it may be changed by the user.	✓
S8	Pause time for comma dial modifier	The function of the "comma" separator depends on the value assigned to register S8. If S8 is set to zero (S8=0), the comma is used as "Wait for PABX dialling tone" dial modifier. If you assign a value between 1 and 255 the comma represents the pause duration of the modem before dialling the digit following the comma in the dial command line. The pause time is expressed in units of 1 second and ranges from 0 to a maximum value of 255. In some countries the maximum value is limited. If, in that case, 2 or more consecutive separators are used, only the first will be considered.	✓
S10	Carrier disconnect time (unit 100 ms)	Register S10 determines the delay time between the loss of the carrier and when the modem disconnects. This delay allows the carrier to drop momentarily without causing a disconnect. The default value of this register is country dependent and is given in 100ms units. If you assign the value to 255 the carrier disconnect function is disabled.	✓
S11	DTMF dial speed	Country dependent. Can not be modified.	✓

S19	info0_setup LSB	the definition of register S19 is identical to the lower byte of location Info0_setup in the data-pump write database. Writing to S19 changes the lower byte of location Info0_setup.	✓
S20	info0_setup MSB	the definition of register S20 is identical to the upper byte of location Info0_setup in the data-pump write database. Writing to S20 changes the upper byte of location Info0_setup.	✓
S21	V8_setup LSB	the definition of register S21 is identical to the lower byte of location V.8_setup in the data-pump write database. Writing to S21 changes the lower byte of location V.8_setup.	✓
S22	V8_setup MSB	the definition of register S22 is identical to the upper byte of location V8_setup in the data-pump write database. Writing to S22 changes the upper byte of location V8_setup.	✓
S23	V34_setup LSB	the definition of register S23 is identical to the lower byte of location V34_setup in the data-pump write database. Writing to S23 changes the lower byte of location V34_setup.	✓
S24	V34_setup MSB	the definition of register S24 is identical to the upper byte of location V34_setup in the data-pump write database. Writing to S24 changes the upper byte of location V34_setup.	✓
S29	Callback delay	Delay in callback security between initial disconnection and the start of the callback. The default value is country dependent.	
S30	No traffic disconnect timeout	A PSTN connection will be interrupted if there is no data to be transmitted (TXD) for a period of S30 ∗ 10 s. If S30 equals 0, no automatic disconnection will occur. Default values and limits are country dependent.	

4.3.1.4. %S Registers

This section gives a complete description of the modem at-set database locations, programmed by the AT set command interpreter. These locations can be changed through the at%s command.

at%s	*Name*	*Default value (descrip.)*	*Default value (decimal)*	*No. of Options*
0	PSTN MODULATION	V.34	11	13
1	PSTN MAX SPD	SPD336	27	40
2	PSTN MIN SPD	SPD24	10	40
3	PSTN SPEEDS	FULLRANGE	1	2
4	TX LEVEL PSTN	LEV06	6	16
5	RX SENS PSTN	RX48	48	2
6	ACTION ON SQ	AUTOSTEPUP	3	5
7	SQ THRESHOLD	HIGHBER	0	3
8	GUARD TONE	DISABLED	1	3
9	DISCONNECT	ENABLED	0	2
10	ANSWER TONE	REVERSALS	1	2
11	CTS DELAY port A	HIGHONACU	7	2
12	DSR CONTROL portA	CTRLTST	1	2
13	DCD CONTROL portA	CONTRL	1	3
14	FLOW CTRL portA	XONXOFF	3	4
15	PROTOCOL	V42	4	5
16	FALLB TO BUF	GOTOBUF	2	2
17	V42 ODP/ADP	ENABLED	0	2
18	COMPRESSION	V42BIS	4	3
19	START UP	STARTAUTO	2	3
20	DCE FLOW CTRL	DISABLED	1	2
21	DTR CONTROL	EXT108_2	2	2
22	AUTO ANSWER	ENABLED	0	2
23	CARRIER LOSS	DISCONN	0	4
24	ABORT TIMER	ENABLED	0	2
25	AUDIO MONITOR	MONCTRL	2	3
26	DIALLING MODE	AUTODET	0	3
27	PULSES / SEC	PULS10	0	3
28	PULSE SYSTEM	ZSYSTEM	0	3
29	NO TRAFFIC DISC	DISABLED	1	2
30	BRKDISCON	DISABLED	1	2
31	LOCECHA	ENABLED	0	2
32	BCC DUMMY	BCCDUMM	90	1
33	AACHAN	ANSW	1	2
34	FIXDTESPD	SPD1152	25	28
35	PSTN MAX SPD TX	SPD336	27	40

PSTN MOD	Determines the modulation used on PSTN.
V.90	ITU-T V.90 Recommendation
V.34	ITU-T V.34 Recommendation
V.32B	ITU-T V.32bis Recommendation
V.32	ITU-T V.32 Recommendation (Uncoded)
V.22B	ITU-T V.22bis Recommendation
V.22	ITU-T V.22 Recommendation
videotex	ITU-T V.23 Recommendation
V.21	ITU-T V.21 Recommendation
BELL212A	
BELL 103	

auto — The modem automatically adapts to the best capability of the remote modem by selecting one of the following ITU-T Recommendations :
V.32B, V.32, V.22B, V.22, V.21, videotex (also BELL212A and BELL 103).

Auto V.8 — The modem automatically adapts to the best capability of the remote modem by selecting one of the following Recommendations : ITU-T V.90, V.34, V.32B, V.32, V.22B, V.22, V.21, videotex (also BELL212A and BELL 103). This automode is conform to ITU-T Recommendation V.8.

Option	***at%s0 (PSTN MODULATION)=***	***Decimal***	***Default***
1	V.90	16	
2	V.34	11	X
3	V.32B	8	
4	V.32	7	
5	V.22B	5	
6	V.22	4	
7	VIDEOTEX	3	
8	V21	2	
9	BELL212A	1	
10	BELL103	0	
11	AUTO	12	
12	AUTOV.8	13	
13	V21FAX	15	

PSTN MAX SPD — Defines the maximum speed (in bit/s) to be used on PSTN. In case of symmetrical speed operation this location determines the speed range of transmitter and receiver. In case of asymmetrical speed operation (v34bis split speed or V90) this location determines the maximum speed of the **receiver**.

Option	***at%s1 (PSTN MAX SPEED)=***	***Decimal***	***Default***
1.	56000	49	
2.	54000+2000/3	48	
3.	53000+1000/3	47	
4.	52000	46	
5.	50000+2000/3	45	
6.	49000+1000/3	44	
7.	48000	43	
8.	46000+2000/3	42	
9.	45000+1000/3	41	
10.	44000	40	
11.	42000+2000/3	39	
12.	41000+1000/3	38	
13.	40000	37	
14.	38000+2000/3	36	
15.	37000+1000/3	35	
16.	36000	34	
17.	34000+2000/3	33	
18.	33600	27	X
19.	33000+1000/3	32	
20.	32000	31	
21.	31200	26	
22.	30000+2000/3	30	
23.	29000+1000/3	29	

24.	28800	21	
25.	28000	28	
26.	26400	20	
27.	24000	19	
28.	21600	18	
29.	19200	17	
30.	16800	16	
31.	14400	15	
32.	12000	14	
33.	9600	13	
34.	7200	12	
35.	4800	11	
36.	2400	10	
37.	1200	9	
38.	600	8	
39.	300	7	
40.	75	2	

PSTN MIN SPD

Defines the minimum speed (in bit/s) to be used on PSTN. This value must be less than or equal to the PSTN MAX SPD and PSTN MAX SPD TX. In case of symmetrical and also asymmetrical speed operation this location determines the minimum speed of transmitter and receiver.

Option	*at%s2 (PSTN MIN SPEED)=*	*Decimal*	*Default*
1.	56000	49	
2.	54000+2000/3	48	
3.	53000+1000/3	47	
4.	52000	46	
5.	50000+2000/3	45	
6.	49000+1000/3	44	
7.	48000	43	
8.	46000+2000/3	42	
9.	45000+1000/3	41	
10.	44000	40	
11.	42000+2000/3	39	
12.	41000+1000/3	38	
13.	40000	37	
14.	38000+2000/3	36	
15.	37000+1000/3	35	
16.	36000	34	
17.	34000+2000/3	33	
18.	33600	27	
19.	33000+1000/3	32	
20.	32000	31	
21.	31200	26	
22.	30000+2000/3	30	
23.	29000+1000/3	29	
24.	28800	21	
25.	28000	28	
26.	26400	20	
27.	24000	19	
28.	21600	18	
29.	19200	17	

30.	16800	16	
31.	14400	15	
32.	12000	14	
33.	9600	13	
34.	7200	12	
35.	4800	11	
36.	2400	10	X
37.	1200	9	
38.	600	8	
39.	300	7	
40.	75	2	

PSTN SPEEDS

Defines the amount of available PSTN speeds. This strap defines the number of fallback speeds.

Only max&min

Only the maximum and minimum speed as defined in PSTN MAX SPD and PSTN MIN SPD will be used as respectively PSTN main and fallback speeds.

Full range

All speeds in the range from the minimum up to the maximum value as defined in the straps PSTN MAX SPD and PSTN MIN SPD can be used. If the main speed is selected (see strap FALLBACK CTRL), the Modem will try to connect at the PSTN MAX SPD. If the line is too bad or the remote modem does not allow that speed, it will take the highest possible speed below PSTN MAX SPD it can connect.

PSTN speed / modulation	33600	31200	28800	26400	24000	21600	19200	16800	14400	12000	9600	7200	4800	2400	1200	600	300	75
V.34	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓				
V.32B									✓	✓	✓	✓	✓					
V.32											✓		✓					
V.22B														✓	✓			
V.22															✓	✓		
videotex															✓			✓
V.21																	✓	
BELL212A															✓			
BELL103																	✓	
auto									✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
auto V8	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 9 : Modulation/Speed combinations on PSTN

Option	*at%s3 (PSTN SPEEDS)=*	*Decimal*	*Default*
1	ONLYMINMAX	0	
2	FULLRANGE	1	X

TX LEVEL PSTN

The transmit level can be strapped between 0 and -15 dBm. In V.34 or V.90 mode, this strap only sets the maximum allowed transmit level. It is possible that the actually used transmit level is lower than the level set, in order to optimise performance. The default value and the value range are country dependent.

Option	*at%s4*	*Decimal*	*Default*

	(TX LEVEL PSTN)=		
1	0 dBm	0	
2	-1 dBm	1	
3	-2 dBm	2	
4	-3 dBm	3	
5	-4 dBm	4	
6	-5 dBm	5	
7	-6 dBm	6	
8	-7 dBm	7	
9	-8 dBm	8	
10	-9 dBm	9	
11	-10 dBm	10	X
12	-11 dBm	11	
13	-12 dBm	12	
14	-13 dBm	13	
15	-14 dBm	14	
16	-15 dBm	15	

RX SENS PSTN

- 38 dBm PSTN input sensitivity from 0 dBm to -38 dBm
- 48 dBm PSTN input sensitivity from 0 dBm to - 48 dBm

Option	*at%s5 (RX SENS PSTN)=*	*Decimal*	*Default*
1	-38 dBm	38	
2	-48 dBm	48	X

ACTION ON SQ

fallb+stepup	The DCE starts a retrain sequence if SQ is too low. This retrain can bring the modem to a fallback speed. If in fallback, and if SQ has been good for a sufficiently long time (typically 7 to 10 min), the modem starts a retrain which can bring it back to the main speed.
Retrain	The DCE starts a retrain sequence if SQ is too low. The modem retrains without going to the fallback speed.
Disabled	The DCE does never take the initiative to start a retrain, but will always reply to a retrain request from the remote modem. **This option is provided for field tests only.**
Fallb	The DCE starts a retrain sequence if SQ is too low. This retrain can bring the modem to a fallback speed defined in the SPEED block.
Note 1:	During data state and retraining, the modem operates within one ITU-T standard. There is never a fallback/stepup from one modulation to another.
Note 2:	As specified in ITU-T Recommendation V.22bis, retrain on SQ is only defined for 2400 bit/s, thus will only occur for that speed. Moreover, a retrain on SQ will never cause a fallback to 1200 bit/s. With V.22bis, fallback can only occur if the remote modem operates at 1200 bit/s or if fallback is forced. Hence, if options “fallb” or “fallb+ stepup” are selected, the operation will be as for “retrain”.
Note 3:	Retrain is not defined by ITU-T for V.29, V.21, videotex, and V.22 operation, nor in the Bell modes. This strap is of no concern in these modes.

Option	*at%s6 (ACTION ON SQ)=*	*Decimal*	*Default*
1	AUTOSTEPUP	3	X
2	RETRAINSQ	5	
3	DISABLED	1	
4	AUTOFALLBACK	2	

SQ THRESHOLD Defines the criterium upon which the ACTION ON SQ strap will be activated.

(This command has been made obsolete. The SQ threshold can be set through the S23 command)

High BER — The modem can tolerate a high Bit Error Rate (BER) before appropriate action is taken as defined by ACTION ON SQ.

Medium BER

Low BER — A minimum number of errors is sufficient to activate retrain and/or fallback as defined in ACTION ON SQ.

Option	*at%s7 (SQ THRESHOLD)=*	*Decimal*	*Default*
1	HIGHBER	0	X
2	MEDIUMBER	1	
3	LOWBER	2	

GUARD TONE — During the V.22bis handshaking procedure the answering modem sends unscrambled binary ones. The frequency spectrum of these unscrambled binary ones unfortunately contains components equal to the disconnect signal of certain telephone exchanges (Limited to Scandinavian countries).

550 Hz — Sending a guard tone of 550 Hz along with these unscrambled binary ones prevents the telephone exchange from disconnecting the link. (Used in Scandinavian countries only)

1800 Hz — Sending a guard tone of 1800 Hz along with these unscrambled binary ones prevents the telephone exchange from disconnecting the link. 1800 Hz is commonly used.

Disabled — No guard tone is sent

Option	*at%s8 (GUARD TONE)=*	*Decimal*	*Default*
1	550 Hz	2	
2	1800 Hz	3	
3	DISABLED	1	X

DISCONNECT — V.90, V.34 and V.32bis modems use a special procedure to disconnect the line (called “cleardown”). Basically, it is a retrain initiated by one of the modems (on drop of DTR) during which the modems decide to go off-line. During this procedure DSR and DCD remain high. Some DTEs however, have a fixed disconnection time-out (time between dropping and raising of DTR). This means that the DTE port raises DTR during the cleardown and consequently finds DSR and DCD in an invalid start-up state, resulting in a lock-up.

Normal — The modem disconnects by means of a retrain. This retrain contains the cleardown code, telling the remote modem to disconnect. This setting is preferred for most applications as it minimises extraneous characters when the modem abruptly disconnects from the telephone line

fast — The modem goes off line without retrain. The remote modem will disconnect on carrier loss.
This option is to be selected only if absolutely necessary since there is no guarantee of the remote being disconnected.

Option	*at%s9 (DISCONNECT)=*	*Decimal*	*Default*
1	ENABLED	0	X
2	DISABLED	1	

ANSWER TONE

This option is of importance only if you selected "auto" as PSTN speed and determines whether the answering modem transmits its answering tone with or without phase reversals.

Reversals — The 2100 Hz answering tone contains phase reversals. THIS SETTING IS PREFERRED since these reversals are used to disable the network echo cancellers on long distance links. Some V.22bis modems, however, suffer from the discontinuous answering tone and can not connect.

No reversals — The modem transmits a continuous 2100 Hz answering tone without phase reversals. This option should be selected only if the calling modem is a V.22bis modem that suffers from the phase reversals. Be ware that long distance calls might cause problems when operating according to V.32 (b) (No disabling of network echo cancellers).

Option	*at%s10 (ANSWER TONE)=*	*Decimal*	*Default*
1	REVERSALS	1	X
2	NOREVERSALS	0	

CTS DELAY

0 ms / 10 ms / 30 ms / 60 ms / 200 ms — The CTS delay value gives the delay before the modem activates the CTS signal after receiving RTS from the DTE. These options only work in transparent mode, while the modem is on line. If ATset or V.25bis are active (DTR ON and PSTN mode), CTS is also active until a training sequence is started.

High on ACU — Idem as the option 0 ms, except that CTS is active if ATset or V25bis are selected, even if DTR is not ON.

Always on — CTS is always active.

Option	*at%s11 (CTS DELAY port A)=*	*Decimal*	*Default*
1	HIGH ON ACU	7	X
2	ALWAYS ON	0	

DSR CONTROL

Data Set Ready. Sent from the modem to the terminal on pin 6, indicating that the modem is ready to operate.

Ctrl in test — DSR remains on only during AL and RL and goes off during ET and DL. DSR follows the ITU-T recommendation for normal LL and PSTN operation.

On in test — DSR remains on during all the tests. DSR follows the ITU-T recommendation for normal LL and PSTN operation.

Always on — Data Set Ready is always on. Most dumb terminals or some communication softwares need DSR permanently on. This setting can be used to prevent the DTE port from blocking when going to dial back-up.

Ctrl by DTR — DSR follows DTR, except if a PSTN is disconnected for another cause than a DTR drop: DSR will then drop for 0.5 seconds. This option is desirable to solve the lockup problem mentioned with strap "DISCONNECT".

Option	*at%s12 (DSR CONTROL portA)=*	*Decimal*	*Default*
1	CTRL IN TST	1	X
2	ALWAYS ON	0	

DCD CONTROL

normal — DCD is on in data mode. DCD is off when the autodialler is being used, during dialling, initial training, and when the modem is disconnected from the telephone line.

ACU on — 109 is on when the modem is disconnected, so that terminals (108/2 must be on) which need this signal can use the ACU (Automatic Calling Unit, i.e. the autodialler). The originate modem drops 109 after detecting the answer tone, while the answering modem drops 109 after sending the 2100 Hz. During on-line state DCD is in accordance with ITU-T specifications.

Always on — DCD is always on regardless of the modem status. This setting can be used to prevent the DTE port from blocking when going to dial back-up.

Option	*at%s13 (DCD CONTROL portA)=*	*Decimal*	*Default*
1	NORMAL	1	X
2	ACU ON	2	
3	ALWAYS ON	0	

FLOW CTRL — This strap defines the type of flow control to be used between DTE PORT and modem (this strap does not act upon the NMS Port).

CTS control — This RS-232 modem interface control signal indicates that the modem is ready to send. The modem drops CTS when its input buffer is nearly full. Similarly it raises CTS once the input buffer is empty.

RTS/CTS ctrl — Flow control is done using circuits 105 and 106. The DTE can stop the data stream coming from the DCE (circuit 104) by dropping RTS. The DCE can stop the DTE from transmitting by dropping CTS. Raising these circuits enables the data stream.

Disabled — If you disable the flow control make sure that the interface speed equals the line speed.

Xon/Xoff ctrl — A flow control protocol for serial asynchronous transmission. X-off turns the transmitter off (ASCII character Control-S). X-on turns the transmitter on (ASCII character Control-Q). The modem sends an X-off character when its input buffer is nearly full. Once the modem input buffer is empty, an X-on character is sent to enable the datastream.

Option	*at%s14 (FLOW CTRL portA)=*	*Decimal*	*Default*
1	CTS CONTROL	2	
2	RTS/CTS CTRL	0	
3	DISABLED	1	
4	XON/XOFF CTRL	3	X

PROTOCOL — This strap selects the error correction protocol. In V.21 and V.23 mode of operation, the modem always works in buffered mode, even if an error correcting protocol is selected.
Selection between ‘MNP’, ‘V42 LAPM’ and ‘auto’ is only for the calling modem. The answering Modem always follows the calling modem.

MNP — Once the communications link is established, the Modem starts handshaking for MNP. If the remote modem does not support MNP, strap “FALLB to BUF” determines whether the Modem switches to buffered operation or disconnects from line.

V42 LAPM — Once the communications link is established the Modem starts handshaking for V.42 (LAPM). If the remote modem doesn’t support V.42, strap “FALLB to BUF” determines whether the Modem switches to buffered operation or disconnects from line.

Auto — Once the communications link is established, the modem starts handshaking for V.42 LAPM. If the remote modem doesn’t support V.42, an attempt is made for an MNP connection. If the MNP handshake also fails, strap “FALLB to BUF” determines whether the Modem switches to buffered operation or disconnects.

Buffered — Use buffered mode if you want a fixed DTE speed during data state and when you are not sure the remote modem can handle one of the error correcting protocols. In synchronous operation, this selection results in the use of error correction, since it is not possible to work buffered in synchronous mode.

Transparent — Use transparent mode if the throughput delay of your modem is critical.

Fix DTE Speed	Protocol	Description
Fixed speed	Non-transparent	DTE speed always fixed.
Fixed speed	Transparent	Transparent during data state, Fixed-speed during V.25bis/AT.
Disabled	Transparent	DTE speed follows line speed with transparent mode of operation
Disabled	Non-transparent	DTE speed follows line speed with protocol mode of operation

Table - Error Correction Configurations

Option	***at%s15 (PROTOCOL)=***	***Decimal***	***Default***
1	MNPMODE	2	
2	V42	4	X
3	AUTOV42MNP	0	
4	BUFFERED	1	
5	TRANSPARENT	7	

FALLB TO BUF — This strap determines the reaction of the Modem if the negotiation on the error corrected link should fail.

Disabled — If the error correction handshake fails, the Modem disconnects from line.

Enabled — If the error correction handshake fails, the Modem switches to buffered mode of operation.

Option	***at%s16 (FALLB TO BUF)=***	***Decimal***	***Default***
1	GOTOBUF	2	X
2	GOOFFLINE	3	

V42 ODP/ADP — An originating V42 modem can start the link negotiation with a special sequence of data, the ODP (Originator Detection Pattern), which is a series of XON characters with alternating parity. If the remote modem also supports V42, it will reply with the ADP (Answerer Detection Pattern). It is recommended to send this sequence, since it will be interpreted as harmless flow control characters if the remote modem doesn't support V.42. If the ODP isn't sent, the link negotiation starts with the transmission of synchronous frames. Hence, a modem not supporting V.42 will receive a stream of illegible characters. If the remote modem is known to be supporting V.42, not sending the ODP will speed up the handshake. The ODP/ADP also negotiates the use of compression.

Enabled — The ODP is transmitted

disabled — The ODP is not transmitted

	at%s17 (V42 ODP/ADP)=	***Decimal***	***Default***
1	DISABLED	1	
2	ENABLED	0	X

COMPRESSION — Error correction must be enabled and the DTE speed must be selected fixed.

MNP 5 — MNP class 5 data compression is used. The compression algorithm uses Run Length Encoding and Adaptive Encoding (Modified Huffman).

V.42 bis — ITU-T V.42bis data compression is used. If the MNP is selected for the protocol strap, V.42bis compression is replaced by MNP 5.

Disabled

Option	*at%s18 (COMPRESSION)=*	*Decimal*	*Default*
1	DISABLED	1	
2	CLASS5	3	
3	V42BIS	4	X

START UP

auto — The modem starts negotiating the error corrected link if it is the originator of the call and replies to the negotiation if it is the acceptor of the call.
Auto selection is preferable. Beware that if fixed start up mode is selected, one modem has to be strapped **fixed orig** and the other **fixed answ**.

Fixed orig — The Modem always starts the error correction negotiation

fixed answ — The Modem only replies to an error correction handshaking initiated by the remote modem.

Option	*at%s19 (START UP)=*	*Decimal*	*Default*
1	STARTAUTO	2	X
2	FIXEDORG	0	
3	FIXEDANSW	1	

DCE FLOW CTRL — This strap controls the data flow between the local and remote modem. To ensure a proper operation the flow control in the “port” block must be set to Xon/Xoff , CTS or RTS/CTS.

Disabled — No flow control between the local and remote modems

Xon/Xoff — An Xoff is sent over the line to prevent the remote modem from sending data. Xon is sent to enable the data stream coming from the remote. Only active in buffered mode of operation.

Option	*at%s20 (DCE FLOW CTRL)=*	*Decimal*	*Default*
1	DISABLED	1	X
2	XON/XOFF	3	

DTR CONTROL

108/2 ext — 108/2 is a ITU-T signal that controls the switching of the modem to or from the line. The ON-condition, indicating the DTE is ready to operate, authorises the modem to switch to the line by supplementary means (e.g. ATA command). The OFF condition causes the modem to switch from the line. This is also called the “addressed mode”.

108/2 int — Some terminals do not supply DTR. There fore DTR can be forced active internally; this means that the modem will not test the state of DTR of the interface connector. The modem can only disconnect via the carrier disconnect function!

Option	*at%s21 (DTR CONTROL)=*	*Decimal*	*Default*
1	108_2 INT	1	
2	108_2 EXT	2	X

AUTO ANSWER Defines how the modem will react to incoming calls.

Enabled The Modem always answers an incoming call and automatically changes to answer mode for the duration of the call, regardless of the setting of the CHANNEL SEL option.

Disabled Incoming calls must be answered manually, and then transferred to the modem (e.g. by using the ATA command).

Option	*at%s22 (AUTO ANSWER)=*	*Decimal*	*Default*
1	ENABLED	0	X
2	DISABLED	1	

CARRIER LOSS

disconnect The Modem goes on hook when the carrier is lost (DCD circuit 109 OFF). Loss of synchronisation will also cause the DCE to disconnect.

No disconnect The Modem stays in off-hook position (no disconnect) even after loss of carrier.

250 ms The Modem goes on hook when the carrier is lost for 250 ms (*)

10 s The Modem goes on hook when the carrier is lost for 10 s (*)

(*) Country dependent option. Can only be selected for certain countries.

Option	*at%s23 (CARRIER LOSS)=*	*Decimal*	*Default*
1	Disconnect	0	X
2	No Disconnect	1	
3	Disconnect 250MS	2	
4	Disconnect 10S	3	

ABORT TIMER

enabled The abort timer causes the modem to hang up (auto-disconnect) after 30 seconds of unsuccessful handshaking.

Disabled The modem stays on line, even after unsuccessful handshaking.

Option	*at%s24 (ABORT TIMER)=*	*Decimal*	*Default*
1	ENABLED	0	X
2	DISABLED	1	

AUDIO MONITOR Allows the user to monitor the line.

Enabled Audio monitor always on (test purposes only)

disabled No monitoring

controlled Monitor enabled during dialling i.e. until detection of 2100 Hz answering tone.

Option	*at%s25 (AUDIO MONITOR)=*	*Decimal*	*Default*
1	MONCTRL	2	X
2	ENABLED	0	
3	DISABLED	1	

DIALLING MODE

Pulse The modem uses pulse dialling.

Option	*at%s26 (DIALLING*	*Decimal*	*Default*

	MODE)=		
1	TONE	2	X

AA CHANNEL This strap determines the channel that will be used to answer the incoming call.

Call The modem will use the call channel to answer the incoming call.
Answer The modem will use the answer channel to answer the incoming call (default setting).

Option	*at%s33 (AA CHANNEL)=*	*Decimal*	*Default*
1	CALL	0	
2	ANSW	1	X

FIX DTE SPD This strap defines the speed at which the DTE will exchange data (READONLY)

Option	*at%s34 (FIX DTE SPEED=*	*DECIMAL*	*Default*
1.	DISABLED	01	
2.	75	02	
3.	75X1200	03	
4.	1200X75	04	
5.	110	05	
6.	150	06	
7.	300	07	
8.	600	08	
9.	1200	09	
10.	2400	10	
11.	4800	11	
12.	7200	12	
13.	9600	13	
14.	12000	14	
15.	14400	15	
16.	16800	16	
17.	19200	17	
18.	21600	18	
19.	24000	19	
20.	26400	20	
21.	28800	21	
22.	38400	22	
23.	57600	23	
24.	76800	24	
25.	115200	25	X
26.	31200	26	
27.	33600	27	
28.	SPDAUTO	81	

PSTN MAX SPD TX Defines the maximum speed (in bit/s) to be used on PSTN. In case of symmetrical speed operation this location is not in use. In case of asymmetrical speed operation (v34bis split speed or V90) this location determines the maximum speed of the **transmitter**.

Option	*at%s35 (PSTN MAX SPEED)=*	*Decimal*	*Default*

1.	56000	49	
2.	54000+2000/3	48	
3.	53000+1000/3	47	
4.	52000	46	
5.	50000+2000/3	45	
6.	49000+1000/3	44	
7.	48000	43	
8.	46000+2000/3	42	
9.	45000+1000/3	41	
10.	44000	40	
11.	42000+2000/3	39	
12.	41000+1000/3	38	
13.	40000	37	
14.	38000+2000/3	36	
15.	37000+1000/3	35	
16.	36000	34	
17.	34000+2000/3	33	
18.	33600	27	X
19.	33000+1000/3	32	
20.	32000	31	
21.	31200	26	
22.	30000+2000/3	30	
23.	29000+1000/3	29	
24.	28800	21	
25.	28000	28	
26.	26400	20	
27.	24000	19	
28.	21600	18	
29.	19200	17	
30.	16800	16	
31.	14400	15	
32.	12000	14	
33.	9600	13	
34.	7200	12	
35.	4800	11	
36.	2400	10	
37.	1200	9	
38.	600	8	
39.	300	7	
40.	75	2	

4.3.2. IDMA control interface

Following locations containing information on the status of the protocols, modem and the connection. They are placed starting from DM H#3E80

00	PRTCLProgress_		• progress information of protocol handshaking • range : h#00..h#3C • An **eventcode** is associated with this location	read
	h#01		initialisation state of AT_ set	
	h#02		initialisation state of v42	
	h#24		v42 DATASTATE	
	h#32		buffered mode DATASTATE	

01	CONstring1_		code corresponding to the DATASTATE connect string	read
	F..8	DTE speed	values identical to location 34 'FIXDTESPD' of the modem AT database	read
	7..0	DCE speed	values identical to location 1 'PSTN MAX SPD' of the modem AT database	read

02	CONstring2_		code corresponding to the DATASTATE connect string	read
	F..8	compression	values identical to location 18 'COMPRESSION' of the modem AT database.	Read
	7..0	Protocol	values identical to location 15 'PROTOCOL' of the modem AT database	read

03	LINEstatus	• contains status information about line/data-pump/protocol follow up routine • An **eventcode** is associated with this location	read
	1	the line follow up routine has signalled to the modem supervisor to go to datamode because the data-pump training was successful	
	2	the line follow up routine has signalled to the modem supervisor to go off line because the data-pump training ended with incompatible speeds (only in V.32)	
	3	the line follow up routine has put the data-pump in retrain because of a hang up (no progress for more than 8 seconds) in previous training	
	4	the line follow up routine has signalled to the modem supervisor to go off line because of one of following reasons: • more than 3 consecutive retrains without a successful connection (Only if AT modem database field 'carrier loss' is set to 'disconnect') • training session with 'cleardown' setting of speed is terminated • modem protocol requested a disconnect	
	5	the line follow up routine has signalled to the modem supervisor to go off line because the remote modem does not respond to a retrain. (Only if AT modem database field 'carrier loss' is set to 'disconnect')	
	6	the line follow up routine has put the data-pump in	

		retrain because more than 10 consecutive training error recovery recycles have occurred in the previous training	
	7	the line follow up routine has signalled to the modem supervisor to go off line because the initial training abort timer has expired • abort timer =30 sec • not for retrains • only in the 'abort timer' field of the modem database is set to 'enabled'	
	8	the line follow up routine has signalled to the modem supervisor to go off line because of a DCD drop in DATASTATE. (Only if AT modem database field 'carrier loss' is set to 'disconnect')	

04..06	spare		

Following locations contain pointers to modem interface structures.

07	DTESCCstructptr	base address of the interface SCC structure for DTE SCC	read
08	HOSTSCCstructptr	base address of the interface SCC structure for HOST SCC	read
09	SPARE		
0A	ATdbaseAddress	base address of the modem database programmed through AT set	read
0B	ACCMmapaddress	base address of HOST SCC control character map TX[32],RX[32]	read
OC	SPARE		

Following locations contain information about modem statistics.

0D	CompressionEfficiency	The number of bytes transferred into the compression encoder divided by the number of bytes transferred out of the encoder, multiplied by 100 for either the current or last call. If a data compression protocol is not in use,this value shall be `100'.	Read
0E	SentOctets (msw)	The number of octets presented to the modem by the DTE.,	read
0F	SentOctets (lsw)	The number of octets presented to the modem by the DTE.	Read
10	RecOctets (msw)	The number of octets presented to the DTE by the modem.	Read
11	RecOctets (lsw)	The number of octets presented to the DTE by the modem.	Read
12	SentDataFrames(msw)	The number of data frames sent on the line interface. If there is no frame-oriented protocol in use on the line interface, this counter shall not increment. (V42,MNP2-4)	read
13	SentDataFrames (lsw)	The number of data frames sent on the line interface. If there is no frame-oriented protocol in use on the line interface, this counter shall not increment. (V42,MNP2-4)	read
14	ReceivedDataFrames (msw)	The number of data frames received on the line interface.If there is no frame-oriented protocol in use on the line interface, this counter shall not increment. (V42,MNP2-4)	read
15	ReceivedDataFrames (lsw)	The number of data frames received on the line interface.If there is no frame-oriented protocol in use on the line interface, this counter shall not increment.	read

		(V42,MNP2-4)	
16	ErrorFrames(msw)	The number of block errors received on the link. If there is no frame-oriented protocol in use on the line interface this counter shall not increment. (V42,MNP2-4)	read
17	ErrorFrames(lsw)	The number of block errors received on the link. If there is no frame-oriented protocol in use on the line interface this counter shall not increment. (V42,MNP2-4)	read
18	ResentFrames(msw)	The number of times this modem retransmits frames on the line interface. If there is no frame-oriented protocol in use on the line interface, this counter shall not increment (V42,MNP2-4)	read
19	ResentFrames(lsw)	The number of times this modem retransmits frames on the line interface. If there is no frame-oriented protocol in use on the line interface, this counter shall not increment (V42,MNP2-4)	read
1A	SentBlocksPPP(msw)	The number of data frames sent on the host interface..	read
1B	SentBlocksPPP(lsw)	The number of data frames sent on the host interface..	read
1C	RecBlocksPPP (msw)	The number of data frames received on the host interface	read
1D	RecBlocksPPP (lsw)	The number of data frames received on the Host interface	read
1E	BlockErrorsPPP(msw)	The number of block errors received on the link	read
1F	BlockErrorsPPP(lsw)	The number of block errors received on the link	read

20	UARTMIRROR		simulation of the UART hardware signals, only used in case of modem software package	read
	3	DCD	0: off, 1: on	
	2	DSR	0 : off, 1 : on	
	1	CTS	0 : off,1 : on	
	0	spare		

21	SPARE		

22	DTESIGNALS		simulation of the UART hardware signals, only used in case of modem software package	write
	F	retrain	IF 0 to 1 transition, host initiates a retrain	
	E	ratechange	IF 0 to1 transistion,host initiates a ratechange	
	…			
	1	RTS	IF 1, RTS is active	
	0	DTR	if 1, DTR is active. If 1 to 0 transistion, host initiates a disconnect	

23	EventMask		Masking of specific events, default=h#0000	write
	F	EventMaskF	If 1, the event with code ‘F’ is masked out	
	E	EventMaskE	If 1, the event with code ‘E’ is masked out	
	D	EventMaskD	If 1, the event with code ‘D’ is masked out	
	C	EventMaskC	If 1, the event with code ‘C’ is masked out	
	B	EventMaskB	If 1, the event with code ‘B’ is masked out	
	A	EventMaskC	If 1, the event with code ‘C’ is masked out	

	9	EventMask9	If 1, the event with code ‘9’ is masked out	
	8	EventMask8	If 1, the event with code ‘8’ is masked out	
	7	EventMask7	If 1, the event with code ‘7’ is masked out	
	6	EventMask6	If 1, the event with code ‘6’ is masked out	
	5	EventMask5	If 1, the event with code ‘5’ is masked out	
	4	EventMask4	If 1, the event with code ‘4’ is masked out	
	3	EventMask3	If 1, the event with code ‘3’ is masked out	
	2	EventMask2	If 1, the event with code ‘2’ is masked out	
	1	EventMask1	If 1, the event with code ‘1’ is masked out	
	0	EventMask0	if 1, all data buffer events associated with the activated buffer interface are masked out	

4.4. IDMA MODEM SCRIPTS

4.4.1. Power up sequence

1 Reset the DSP.
2 Write the boot page (99) into the DSP memory through the IDMA port. Execution of the DSP program will start because the last instruction loaded is to PM(0x0000) [27].
3 The DSP will now request for the dial page (0).
4 The dial page is loaded according to section 2.2.

After this sequence the modem is initialised and it is possible to start sending AT-commands (if SCC2 is used).

4.4.2. modem connect sequence

The modem is configured by using AT commands and initialising setup registers through idma,

The following sequence assumes that the dial page is loaded and that the modem is configured to use SCC2 as the IDMA interface (**dataconfig**).

1 On the dial page, when **PRTCLPROGESS_**=01 the SCC structure has been initialised by the off line AT_SET, the SCC structure parameters can be read by the IDMA host . AT_SET now is available.
2 The IDMA HOST has to activate the **DTR** signal in location '**DTE_SIGNALS**'
3 By default the modem is in **AUTOANSWER**, an incoming ring will activate the modem training. The command **ATA** will start the modem in answer mode, ring detection is skipped. **ATDxxxx** will start the modem in calling mode with xxxx being the dial number. In case of **ATD** without a number, the dialling procedure is skipped. Default the modem is put in **AUTOMODE** for the protocols and for the modulation[28].
4 During the call progress several buffer event will be generated, indicating call progress information is available in SCC2 .
5 If PPP will be used to transfer data, the IDMA interface should be reconfigured according to section 4.1.1. The SCC3 functionality will only be available when the protocol reaches **DATASTATE**.
6 The V.8 page (6) is loaded into the DSP according to section 2.2.
7 The Info page (7) is loaded into the DSP according to section 2.2.
8 The V.34 page (8) is loaded into the DSP according to section 2.2.
9 When **PRTCLPROGRESS_** >=2 the SCC structure is reinitialised again by the DSP, this time by the protocol software . The host should read the new values of the structure.
10 Before the protocol reaches **DATASTATE** a first buffer event will be generated, indicating the connect string is available in SCC2.
11 When **PRTCLPROGRESS_** reaches **DATASTATE** (Prtclprogress_=h#24 for v42 or Prtclprogress_=h#32 for buffered mode), IDMA HOST can now read data from and write data to SCC2 or SCC3 depending on the configuration, see section 7.4.4.

4.4.3. modem disconnect sequence

1 The IDMA HOST changes the value of DTR from 1 to 0, this will initiate a disconnect.
2 The DSP will perform a disconnect handshake.
3 When the disconnect handshake is terminated, the DSP will request for the dial page (0).
4 The dial page is loaded.
5 After this sequence the modem is initiated and it is possible to start sending AT-commands (if SCC2 is used).

4.4.4. Statistics

The statistics are retrieved from the modem database, for locations see 0, through the DSPs IDMA port.
The statistics are accessible at all time, in both on and off line.

[27] *This instruction is put in the last page block of the bootpage*
[28] *The default modulation in the current version is V34*

4.5. MODEM EVENT CODES

Note : new events can be added to this list during product development, also code assignment may change.

the dtesccrx empty and dtescctx full events only occur if **dataconfig**=b#xxxx xxxx xxxx xx00
the hostsccrx full and hostscctx empty events only occur if **dataconfig** =b#xxxx xxxx xxxx xx01

EVENT	CODE	
new value of Rstatus_ch	H#0001	
new value of Rstatus	H#0002	
new value of TRNProgress	H#0003	
new value of PRTCLprogress	H#0005	
new value of LINEstatus	H#0006	
DTESCCRX EMPTY buffer0	H#0030	For the 8 RX buffers of the SCC2
DTESCCRX EMPTY buffer1	H#0031	an event will be generated when
DTESCCRX EMPTY buffer2	H#0032	they are read by the protocol.
DTESCCRX EMPTY buffer3	H#0033	This is the signal for the host to
DTESCCRX EMPTY buffer4	H#0034	write to these RX buffers.
DTESCCRX EMPTY buffer5	H#0035	
DTESCCRX EMPTY buffer6	H#0036	
DTESCCRX EMPTY buffer7	H#0037	
DTESCCTX FULL buffer 0	H#0038	For the 8 TX buffers of the SCC2
DTESCCTX FULL buffer1	H#0039	an event will be generated when
DTESCCTX FULL buffer2	H#003A	they are written by the protocol.
DTESCCTX FULL buffer3	H#003B	This is the signal for the host to
DTESCCTX FULL buffer4	H#003C	read the TX buffers.
DTESCCTX FULL buffer5	H#003D	
DTESCCTX FULL buffer6	H#003E	
DTESCCTX FULL buffer7	H#003F	
HOSTSCCRX FULL buffer0	H#0060	
HOSTSCCRX FULL buffer1	H#0061	
HOSTSCCRX FULL buffer2	H#0062	
HOSTSCCRX FULL buffer3	H#0063	
HOSTSCCRX FULL buffer4	H#0064	
HOSTSCCRX FULL buffer5	H#0065	
HOSTSCCRX FULL buffer6	H#0066	
HOSTSCCRX FULL buffer7	H#0067	
HOSTSCCTX EMPTY buffer0	H#0068	
HOSTSCCTX EMPTY buffer1	H#0069	
HOSTSCCTX EMPTY buffer2	H#006A	
HOSTSCCTX EMPTY buffer3	H#006B	
HOSTSCCTX EMPTY buffer4	H#006C	
HOSTSCCTX EMPTY buffer5	H#006D	
HOSTSCCTX EMPTY buffer6	H#006E	
HOSTSCCTX EMPTY buffer7	H#006F	

5. HOST INTERFACE (DATA-PUMP SOFTWARE)

5.1. INTRODUCTION

The HOST processor has four tasks:

- **Exchanging data packets**
- **Controlling the data-pump**
- **Loading software** in the DSP's memory[29]
- **configuration** of the system

5.1.1. data-pump block diagram

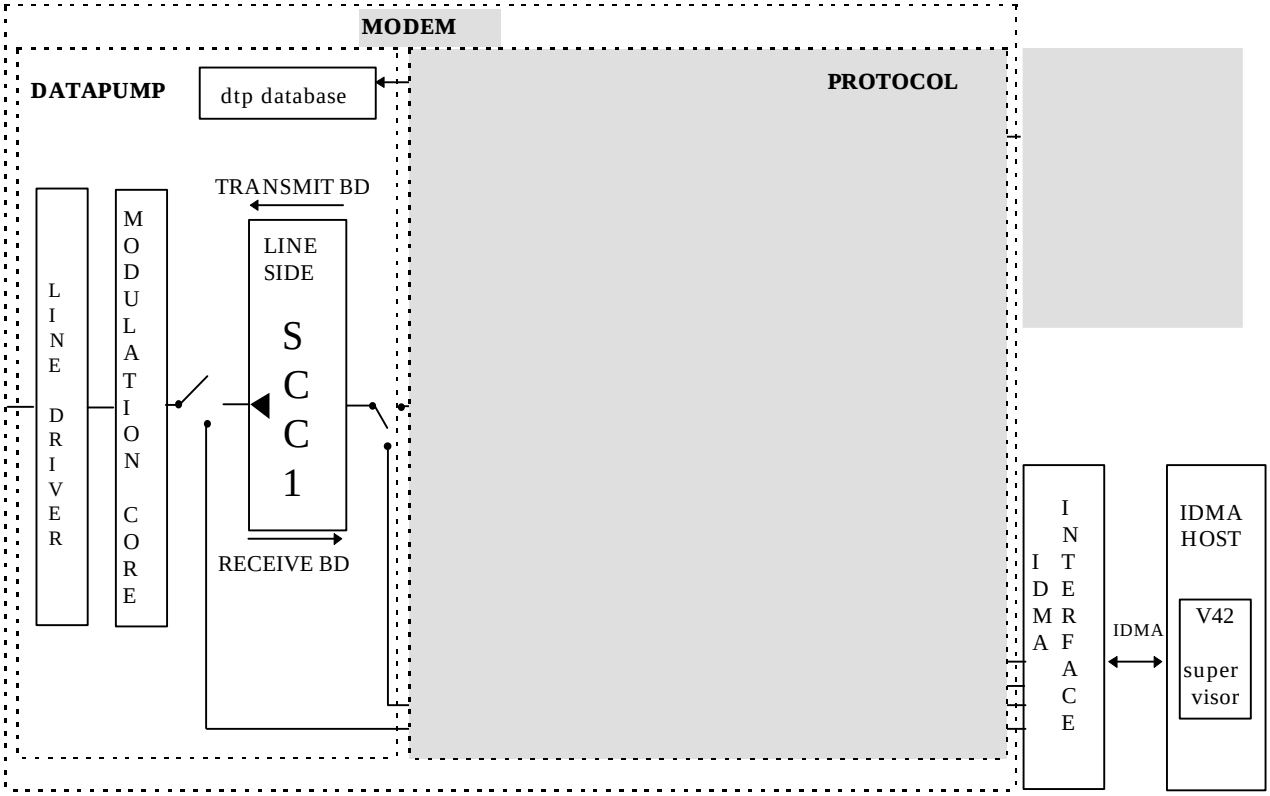


Figure 15 : data-pump block diagram

5.1.2. data flow

The user can interface to the data-pump via the **IDMA interface**

- The IDMA host can transfer data via a direct gateway to the TXD- and RXD-,buffers of the data-pump, transferring synchronous data at an average rate of 2400 Hz .

[29] Only if the IDMA-loading method is applied.

- A parallel interface to a usart (SCC1) which can handle HDLC frames and UART functionality using V14.

5.1.3. control flow

Following sections explain the control operation for a system with IDMA HOST.

5.1.3.1. The IDMA interface

No AT set is available on the data-pump, instead programming the data-pump is done in the memory mapped write database. Control flow from DSP to host uses the event mechanism. Control flow from host to DSP is memory mapped.

In the case of IDMA interface an essential part of the control flow **from DSP to host** is the **event generation**. The event mechanism is used to signal to the host whenever status of the data-pump changes, p.e.
1. the interface signals (CTS,DCD,DSR) change
2. newly received information has been filled in buffers at RX side
3. Buffers have been emptied at TX side and should be filled
4. a boot operation is requested for IDMA download of a new page
5. …

Whenever the DSP -modem has new information available, it generates an event, i.e.

- it pushes an **event code** on the **event buffer** . The event buffer is an eight locations wrap around buffer. The event code specifies the kind of information available.

- it increments the **event counter**. The three lsb's of this event counter are used as a wrap around write pointer for the event buffer. The pointer points to the last written position + 1 of the event buffer. The full 16 bit constitute a 16 bit wrap around counter counting the number of events. The host system has it's own event counter, pointing to the last read location + 1. If the difference between Host and DSP event counter is higher than eight, event information has been lost.

- it generates an **event pulse** using one of the DSP flags.

The host system can use the event system in polling or interrupt mode.
In polling mode the host regularly has to compare the DSP event counter with it's internal counter. Whenever there is a difference, it should read the event(s) from the event buffer. In interrupt mode the same operation should be triggered by a interrupt.

The status of the data-pump is maintained in the **memory mapped data-pump read database**. In some cases, following on the detection of the event, the host has to do some additional reads in the read database, p.e.
1. when, at the end of the training, the DATASTATE speed is available
2. ..

In the case of IDMA interface, control flow **from HOST to DSP** is **memory mapped**.
This memory mapped interface is used to signal to the DSP whenever
1. a boot operation is finished
2. to switch the system configuration
3. ..

On a regular basis the DSP checks the memory mapped interface and takes appropriate action.
The actual structure for the event mechanism is described in Annex 6.3

5.1.3.2. data-pump setup

After the dial page is loaded in memory, the data-pump is **setup** by programming the appropriate data memory locations. The section ‘data-pump write database’, 5.3.1, explains in detail how to program a location.
Each modification to a data memory location is activated by setting bit 14 in WSTATUS location.

5.1.3.3. data-pump Status

All status registers of interest to the IDMA host have been grouped in the datapump read database. Section 5.3.2 explains in detail.

5.2. DATA INTERFACE

5.2.1. parallel synchronous interface

The IDMA host who wants to send/receive a bit stream to/from the remote data-pump should use the parallel synchronous interface. This interface is directly connected to the modulation core of the data-pump. The data-pump does not place any framing information (for instance byte alignment) onto this bit stream.

Synchronous digital data is exchanged in packets (Datagrams) at a fixed frequency (see location **DatagramRate** of the data-pump read database, for instance 2400Hz for V32 or V34). This means that the number of bits such a package contains equals to 'linebitrate/Datagramrate' .

For all modulations except V90D, every location of the two TX-Buffers and RX-Buffers contains a Datagram. When a TX buffer is empty or a RX buffer is full an event is generated and on one of the output flags a event pulse is generated.

In the case of V90D one transmitter **Datagram** occupies three subsequent locations in the Txbuffers. The datagrams are place into the range TXBuffer0,TXBuffer1 range as indicated in Figure 16 : databuffers of the parallel synchronous interface. For the receiver datagrams the general case applies.

The datastructure below is used for the data interface between host and DSP data-pump together with the event mechanism. The data-pump read database location datastructptr, location h#91, points to the first location of this structure.

offset	bufferfield	definition (general case)	definition (V90D case)
00	TXBuffer0[0]	TXBuffer0[0] is a 16 bit location containing a main channel datagram. The bit definition corresponds to the TXD0 definition (general case) Location [0] is the oldest location of the Buffer	V90D datagram[0], the bit definition corresponds to the TXD0 definition (V90D case)
01	TXBuffer0[1]		V90D datagram[0], the bit definition corresponds to the TXD1 definition (V90D case)
02	TXBuffer0[2]		V90D datagram[0], the bit definition corresponds to the TXD2 efinition (V90D case)
03	TXBuffer0[3]		V90D datagram[1]
04	TXBuffer0[4]		V90D datagram[1]
05	TXBuffer0[5]		V90D datagram[1]
06	TXBuffer0[6]		V90D datagram[2]
07	TXBuffer0[7]		V90D datagram[2]
08	TXBuffer1[0]		V90D datagram[2]
09	TXBuffer1[1]		V90D datagram[3]
0A	TXBuffer1[2]		V90D datagram[3]
0B	TXBuffer1[3]		V90D datagram[3]
0C	TXBuffer1[4]		V90D datagram[4]
0D	TXBuffer1[5]		V90D datagram[4]
0E	TXBuffer1[6]		V90D datagram[4]
0F	TXBuffer1[7]		NOT USED
10	RXBuffer0[0]	RXBuffer0[0] is a 16 bit location containing the main channel receive data. Location [0] is the oldest location of the Buffer	
11	RXBuffer0[1]		

12	RXBuffer0[2]		
13	RXBuffer0[3]		
14	RXBuffer0[4]		
15	RXBuffer0[5]		
16	RXBuffer0[6]		
17	RXBuffer0[7]		
18	RXBuffer1[0]		
19	RXBuffer1[1]		
1A	RXBuffer1[2]		
1B	RXBuffer1[3]		
1C	RXBuffer1[4]		
1D	RXBuffer1[5]		
1E	RXBuffer1[6]		
1F	RXBuffer1[7]		

Figure 16 : databuffers of the parallel synchronous interface

5.2.2. Synchronous HDLC SCC

The framing structure of HDLC is shown below.

OPENING FLAG = 7E	INFORMATION (8N BITS)	CRC (2 x 8 BITS)	CLOSING FLAG = 7E

HDLC uses a zero insertion/deletion process (commonly known as bit-stuffing) to ensure that the bit pattern of the delimiter flag does not occur in the fields between flags.

Data is transmitted in the data field, which can vary in length depending upon the protocol using the frame.
Error control is implemented by appending a cyclic redundancy check (CRC) to the frame, which is 16-bits long.
For applications where the error control is performed by an upper OSI layer, the CRC characters can be discarded.
When the MODE1-MODE0 bits of the SCC mode register are equal to zero, the SCC functions as an HDLC controller.

The HDLC controller key features are as follows:

- Flexible Data Buffers with Multiple Buffers per Frame Allowed
- Flag/Abort/Idle Generation/Detection
- Zero Insertion/Deletion
- 16-Bit CRC-CCITT Generation/Checking
- Detection of Non-Octet Aligned Frames

HDLC Channel Frame Transmission Processing

When the host enables the HDLC transmitter, it will start transmitting opening flags. The HDLC controller will poll the first buffer descriptor (BD) in the transmit channel's BD table. When there is a frame to transmit, the HDLC controller will fetch the data from memory (alternating LS- and MS- part of DM-word, starting with LS or MS is specified by the Endian bit of the Modereg2) and start transmitting the frame (which is preceded by an opening flag). When the end of the current BD has been reached and the last buffer in the frame bit is set, the two CRC characters (if the NO_CRC bit of the ModeReg = 0) and the closing flag are appended.
Following the transmission of the closing flag, the HDLC controller writes the frame status bits into the BD and clears the ready bit. When the end of the current BD has been reached, and the last bit is not set (working in multibuffer mode), only the ready bit is cleared. The HDLC controller will then proceed to the next BD in the table. When this BD is not ready for transmission, an ABORT sequence (multiple FF) will be transmitted. Flags are transmitted next until the current BD is ready.

HDLC Channel Frame Reception Processing

When the host enables the receiver, it waits for an opening flag character. When this flag is detected, the HDLC controller fetches the current BD and, if empty, will start to transfer the incoming frame (all fields between the open and closing flags, including the 2 CRC characters) to the BD's associated data buffer (alternating LS- and MS- part of DM-word, starting with LS or MS is specified by the Endian bit of the Modereg2). When the data buffer has been filled, the HDLC controller clears the empty bit in the BD. If the incoming frame exceeds the length of the data buffer, the HDLC controller will fetch the next BD in the table and, if it is empty, will continue to transfer the rest of the frame to this BD's associated data buffer.
When the frame ends, the CRC field is checked against the recalculated value and is written to the data buffer. The data length written to the last BD in the HDLC frame is the length of the entire frame. The HDLC controller then sets the 'last buffer in frame' bit, writes the frame status bits into the BD, and clears the empty bit.
The HDLC controller then waits for a new frame. Back-to-back frames may be received with only a single shared flag between frames.

After reception of an ABORT sequence, the current BD will be closed by clearing bit 15 and setting the AB bit. The L and F bits are set accordingly. DATALENGTH will contain the number of received characters preceding the ABORT sequence. The HDLC controller then waits for a new frame which will be written in the next BD.

the base address of the dceSCCstructure can be found in the location DCESCCstructPtr of data-pump read database

The table below gives an overview of the usage of each field of the SCC structure by the CP, which runs on the DSP, and by the IDMA HOST.

Offset				IDMA host		DSP (CP)	
				R	W	R	W
0	ModeReg				X	X	
	F, .. 4		not used				
	3	ENR	1: enable receiver (written by user in DATASTATE)		X	X	
	2	ENT	1 : enable transmitter (written by user in DATASTATE)		X	X	
	1, 0	MODE	00 : HDLC 01 : async (V14)		X	X	
1	Nbits		number of bits of per baud-package (bits/baud) for the SCC receiver.			X	X
2	EventReg					X	X
3	TXState		During software booting this location is initialised with the address of the SCC transmitter routine. The SCC initialisation routine initialises this location with the initial value of the state variable of the CP transmitter During runtime the location contains the state variable of the CP transmitter			X	X
4	RXState		During software booting this location is initialised with the address of the SCC receiver routine. The SCC initialisation routine initialises this location with the initial value of the state variable of the CP receiver. During runtime the location contains the state variable of the CP receiver			X	X
5	TotalBuf Count		total number of received characters in current frame			X	X
6 .. 1D	RXBufferDescriptor[8*3]		Data associated with the SCC receiver is stored in buffers. Each buffer is referenced by a BufferDescriptor(BD). BD's are located in a BD array. The BD array allows to define up to eight buffers for each channel .	X	X	X	X
	offset 0	STATUS AND CONTROL					
	offset 1	DATA LENGTH					
	offset 2	DATA BUFFER POINTER					
1E .. 35	TXBufferDescriptor[8*3]		Data associated with the SCC transmitter is stored in buffers. Each buffer is referenced by a BufferDescriptor(BD). BD's are located in a BD array.. The BD array allows to define up to eight buffers for each channel .	X	X	X	X
	offset 0	STATUS AND CONTROL					
	offset 1	DATA LENGTH					
	offset 2	DATA BUFFER POINTER					

36	ModeReg2		indicates the number of buffers and the character operation mode of the buffers				
	F	TXinit	The TXBD's are initialised by the CP using the default values of MaxBufLength (= 260) and NbrofBufs (= 8). TXinit bit can be used by the user to reinitialise the TXBD's. He therefor has to disable the transmitter (ENT=0) first, then change MaxBufLength and NbrofBufs accordingly and set the TXinit and ENT bit. The TXinit bit is reset afterwards.		X	X	X
	E	RXinit	The RXBD's are initialised by the CP using the default values of MaxBufLength (= 260) and NbrofBufs (= 8). RXinit bit can be used by the user to reinitialise the RXBD's. He therefor has to disable the receiver (ENR=0) first, then change MaxBufLength and NbrofBufs accordingly and set the RXinit and ENR bit. The RXinit bit is reset afterwards.		X	X	X
	D .. 9	SPARE					
	8	NO_CRC	1: disable 16-Bit CRC Generation/Checking (default = 0)		X	X	
	7 6	SPARE					
	5	Endian type	this bit should be set or reset before booting of the page or before setting the TXinit and RXinit bits. 1 : big endian for CharMode = 1 the MS-part of a DM location is accessed first, the LS-part secondly 0 : little endian (default value) for CharMode = 1 the LS-part of a DM location is accessed first, the MS-part secondly		X	X	
	4	CharMode	1: all buffers operate with two characters per location (see bit 5 Endian type for location accessing) 0: all buffers operate with one character per location (LS-part)	X			X
	3 .. 0	NbrofBufs	indicates the number of transmit buffers and receive buffers. The number of the RX and TX data buffers is initialised by the modem software to : default number = 8. This value can be over written by the user for reinitialization.	X	X	X	X
37	MaxBufLength		the maximum length of an RX or TX buffer. The default size of the RX and TX data buffers is initialised by the modem software to : default size=130 memory locations. This value should be even when written by the user for reinitialization.	X	X	X	X
38	RXSCCBuffer		indicates in which BufferDescriptor the receiver CP is writing data	X		X	X
39	RXSCCBufferOffset		indicates the number of received characters	X		X	X
3A	RXSCCBufferBitOffset		indicates the number of tested bits of the last received baud			X	X
3B	RXSCCCRC		contains the calculated CRC of all received characters			X	X
3C	RXEvent_Log		contains information., concerning filled receive buffers, which is used to update the eventbuffer.			X	X
	F .. 6	SPARE					
	5 .. 3	NoRXSCCBuffer	indicates the number of RXSCCBuffers, for which the eventbuffer has not yet been updated. Reset by the CP after updating the eventbuffer.			X	X
	2	RXSCCBuffer	indicates the first RXSCCBuffer, in a sequence of length			X	X

	.. 0		NORXSCCBuffer, for which the eventbuffer has not yet been updated. Reset by the CP after updating the eventbuffer.				
3D	RXWorkReg1					X	X
3E	RXWorkReg2					X	X
3F	RXBreakLength					X	X
40	TXSCCBuffer		indicates from which BufferDescriptor the transmitter is sending data	X		X	X
41	TXSCCBufferOffset		indicates the current transmit memory word	X		X	X
42	TXSCCBufferBitOffset						
43	TXSCCCRC		contains the calculated 16-bit CRC of all transmitted characters				X
44	TXSCCCRC1		contains the upper 8 bits of the 16 bit CRC				X
45	TXBitsCalculated		contains the number of calculated transmitbits			X	X
46	TXEvent_Log		contains information., concerning transmitted buffers, which is used to update the eventbuffer.			X	X
	F .. 6	SPARE					
	5 .. 3	NOTXSCCBuffer	indicates the number of TXSCCBuffers, for which the eventbuffer has not yet been updated. Reset by the CP after updating the eventbuffer.			X	X
	2 .. 0	TXSCCBuffer	indicates the first TXSCCBuffer, in a sequence of length NOTXSCCBuffer, for which the eventbuffer has not yet been updated. Reset by the CP after updating the eventbuffer.			X	X
47	TXWorkReg1					X	X
48	TXWorkReg2					X	X
49	TXPtr		During runtime the location points to current DM word (2 characters per word)			X	X
4A - 51			not used				
52	nbitsTx		determines the number of bits per baud package at the transmitter side.			X	X

Table 10 : DCE SCC structure for synchronous HDLC

5.2.2.2. HDLC receive BufferDescriptor (RX bd)

The HDLC controller uses the RX BD to report information about the received data for each buffer. The RX BD is shown in figure …

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET +0	E	-	W		L	F	-	-	-	-		NO	AB	CR		
OFFSET +2	DATA LENGTH															
OFFSET +4	RX BUFFER POINTER															

Figure 17 : HDLC Receive BufferDescriptor

An example of the HDLC receive process is shown in figure … This shows the resulting state of the RX BDs after receipt of a complete frame spanning two receive buffers and a second frame with an unexpected abort sequence. The example assumes that MaxBufLength = 8.

The HDLC controller saves received characters alternating in both bytes of a DM- word, starting with the least significant part (bit 5 Modereg2 Endian type = little). After writing two characters in the LS- and MS-Part of the current DM-word, the next character is written in the LS-part of the next DM-word. Therefore when the host reads the BD's associated buffer he also has to start with the LS-part of the DM- word pointed to by the BD's associated datapointer. The second received character will be in the MS-part of the same DM-word. The third character in the LS-part of the second DM-word and so on...

The first word of the RX BD contains control and status bits. Bits 15 - 10 are written by the user before the buffer is linked to the RX BD table, and bits 5 - 0 are set by the CP following frame reception. Bit 15 is set by the user when the buffer is available to the HDLC controller; it is cleared by the HDLC controller when the buffer is full.

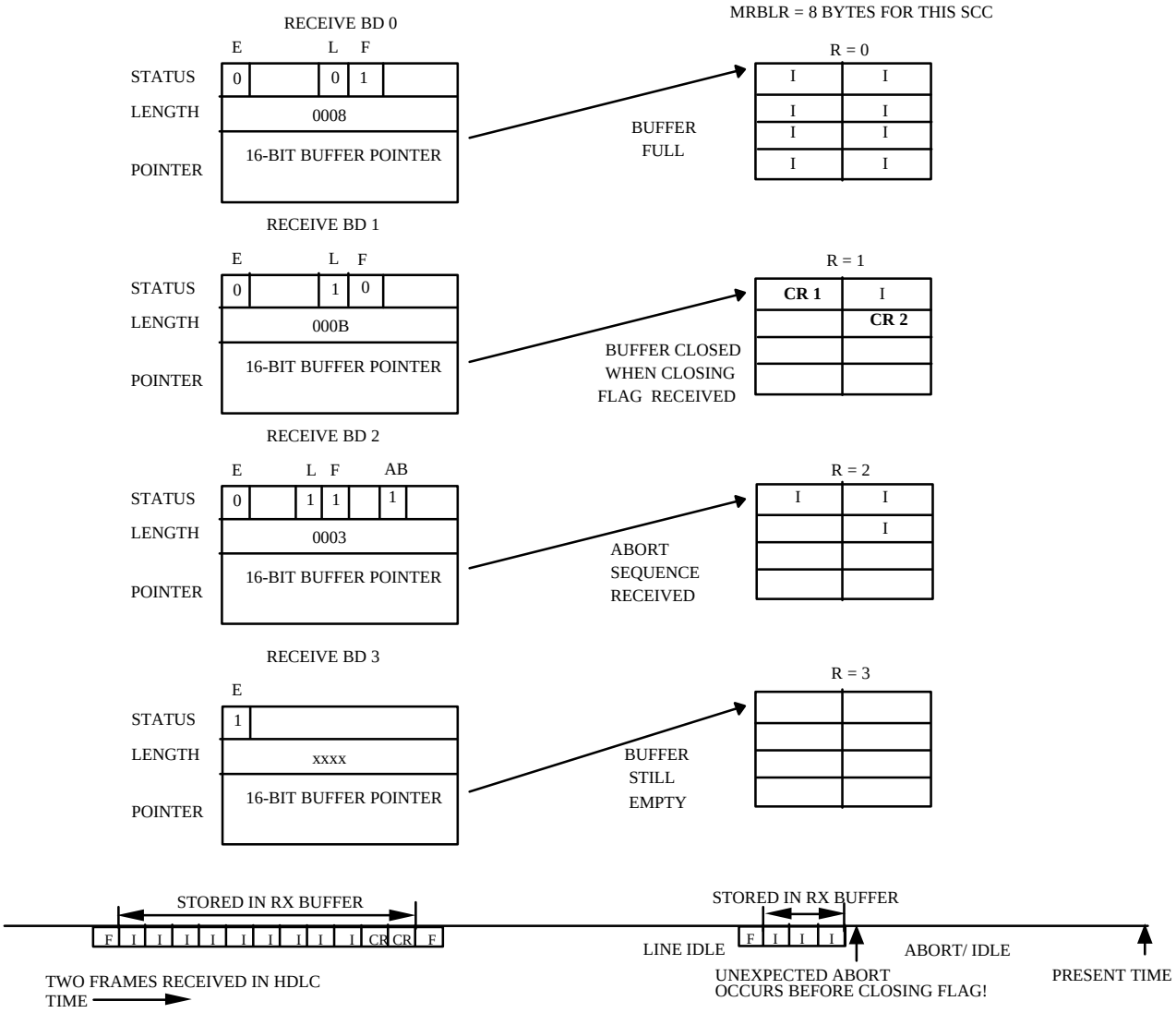


Figure 18 : Sync HDLC Receive BD Example (ENDIAN type = little)

LEGEND
F = FLAG
R = SCC1 RXSCCBuffer
I = INFORMATION BYTE
CR = CRC BYTE

E - Empty

0 = The data buffer associated with this BD has been filled with received data, or data reception has been aborted due to an error condition. The user is free to examine or write to any fields of the BD.
1 = The data buffer associated with this BD is empty. This bit signifies that the BD and its associated buffer are available to the HDLC controller. The user should not write to any fields of this BD after it sets this bit. The empty bit will remain set while the HDLC controller is currently filling the buffer with received data.

W - Wrap (Final BD in Table)

0 = This is not the last BD in the RX BD table.
1 = This is the last BD in the RX BD table. After this buffer has been used, the HDLC controller will receive incoming data into the first BD in the table.

The following status bits are written by the HDLC controller after the received data has been placed into the associated data buffer.

L - Last in Frame

This bit is set by the HDLC controller when this buffer is the last in a frame. This implies the reception of a closing flag or reception or an error, in which case one or more of the CR, AB and NO bits are set. The HDLC controller will write the number of frame octets to the data length field.

0 = This buffer is not the last in a frame.
1 = This buffer is the last in a frame.

F - First in Frame

This bit is set by the HDLC controller when this buffer is the first in a frame.
0 = The buffer is not the first in a frame.
1 = The buffer is the first in a frame.

Bits 9-6 - Reserved for future use.

NO - RX Nonoctet Aligned Frame
A frame that contained a number of bits not exactly divisible by eight was received.

AB - RX Abort Sequence
A minimum of seven consecutive ones was received during frame reception.

CR - RX CRC Error
This frame contains a CRC error.

Data Length
The data length is the number of octects written to this BD's data buffer by the HDLC controller.
When this BD is the last BD in the frame (L=1), the data length contains the total number of frame octets (including two bytes for CRC).

RX Buffer Pointer
The receive buffer pointer always points to the first location of the associated data buffer, which can reside in either internal or overlay memory. The 2 highest bits of the pointer indicate the overlay number (0, 1 or 2).

NOTE
The RX buffer pointer must be even.

Data is presented to the HDLC controller for transmission on an SCC channel by arranging it in buffers referenced by the channel's TX BD table. The HDLC controller confirms transmission (or indicates error conditions) using the BDs to inform the user that the buffers have been serviced. The TX BD is shown in figure …

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET +0	R	-	W		L	-	-	-	-	-	-	-	-	-	-	-
OFFSET +2	DATA LENGTH															
OFFSET +4	TX BUFFER POINTER															

Figure 19 : HDLC Transmit BufferDescriptor

The HDLC controller transmits both bytes of every word in the TX buffer, starting with the least significant part (bit 5 of Modereg2 Endian type = 0). After transmitting the character in the MS-Part, the CP changes to the LS-part of the next DM-word.
Therefor when the host writes to the BD's associated buffer he has to start with the LS-part of the DM- word pointed to by the BD's associated datapointer. The second transmit character has to be written in the MS-part of the same DM-word. The third character in the LS-part of the second DM-word and so on...

The first word of the TX BD contains status and control bits. Bits 15 - 10 are prepared by the user before transmission. Bit 15 is set by the user when the buffer and BD have been prepared and is cleared by the HDLC controller after the frame has been transmitted.

R - Ready

0 = This buffer is not currently ready for transmission. The user is free to manipulate this BD (or its associated buffer). The HDLC controller clears this bit after the buffer has been fully transmitted or after an error condition has been encountered.
1 = The data buffer, which has been prepared for transmission by the user, has not yet transmitted. No fields of this BD may be written by the user once this bit is set.

W - Wrap (Final BD in Table)

0 = This is not the last BD in the TX BD table.
1 = This is the last BD in the TX BD table. After this buffer has been used, the HDLC controller will transmit data from the first BD in the table.

L - Last

0 = This is not the last buffer in the frame.
1 = This is the last buffer in the current frame.

Bits 10 - 0 - Reserved for future use.

Data Length
The data length is the number of octects (not the number of DM-words) that the CP should transmit from this BD's data buffer. It is never modified by the CP. This value should be normally greater than zero.

TX Buffer Pointer
The transmit buffer pointer always points to the first location of the associated data buffer, which can reside in either internal or overlay memory.

5.2.2.4. IDMA - SCC1 transmitter interaction

This paragraph describes a flow diagram that could be used as an example for the host driver interacting with the SCC1 transmitter (Endian type = little)

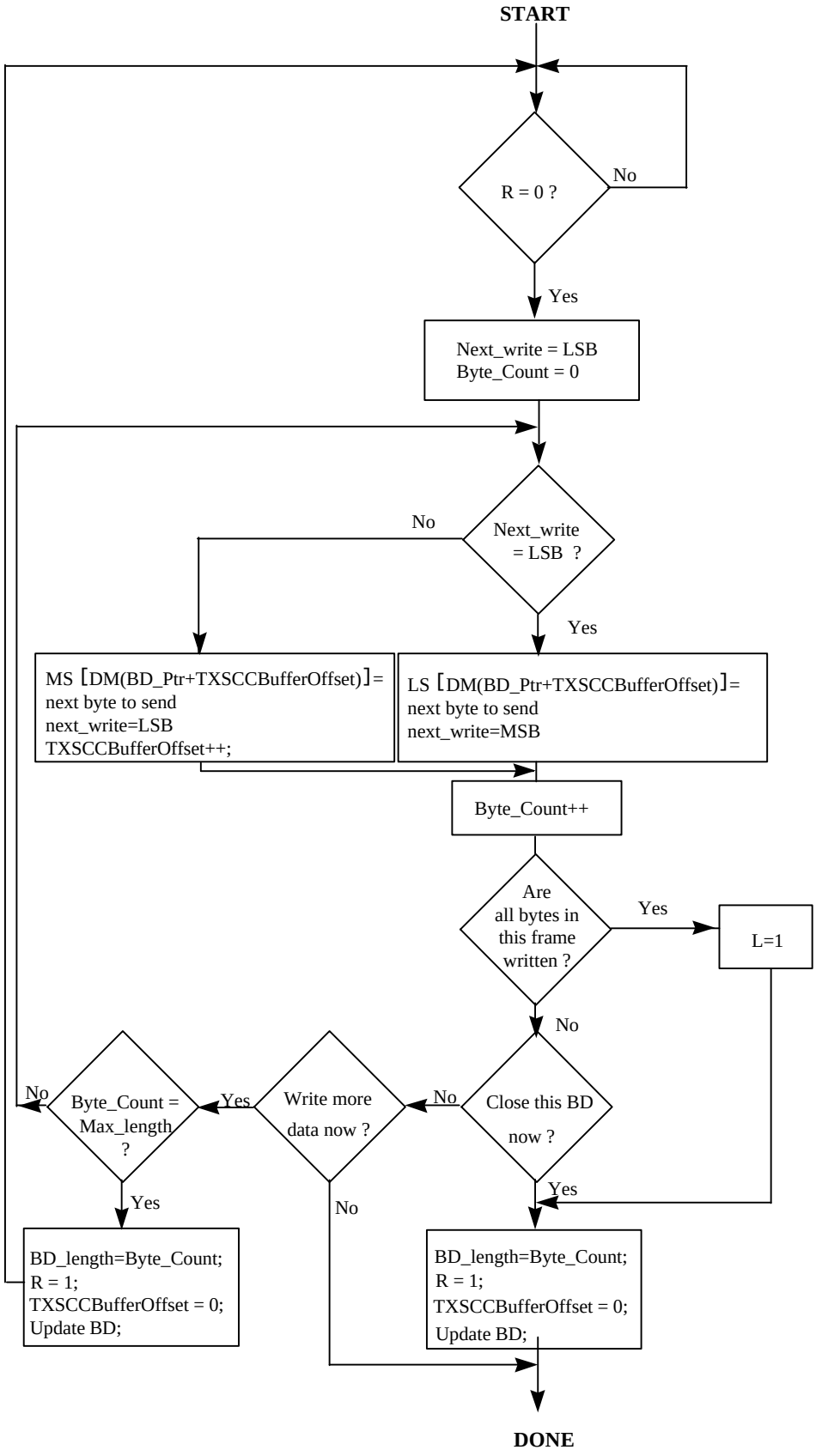


Figure 20 : IDMA - SCC1 transmitter interaction for synchronous HDLC mode (little Endian)

SCC1 is directed towards the MODEM firmware, i.e. when the IDMA HOST wants to write to SCC1 (e.g. MODEM data) it will use the write buffers in SCC1.
The IDMA HOST has a variable containing the current BufferDescriptor (BD) to write to. This variable (called BD in the flow chart) is initiated to buffer no. 0. When this variable is updated the buffer no. is increased with 1 until it reaches the wrapbuffer (default = buffer 7) plus 1, then it is set to 0 again. The IDMA HOST must always check that the R bit in the current BD is cleared (0) before it can start modifying the BD or its data. The IDMA HOST sets (1) this bit when all data has been written to this BD. The DSP clears this bit in the BufferDescriptors that are ready to be filled by the IDMA HOST.
When a complete frame has been written to the DSP the IDMA HOST sets (1) the L bit in the BufferDescriptor containing the last data.

The sequence which the IDMA HOST should follow when writing to SCC1 is described in the flow chart in Fig.5. (describes the case of Polling mode)

Remarks:
1. The ‘start’ task can be interrupt triggered by an incoming event indicating a buffer is empty
2. in interrupt triggered operation, checking the ready bit is allowed but not required, if the bit is set it would indicate an error condition. Equally initiating a wait loop in not required.
3. test ‘close this BD now’ :on what basis will the IDMA HOST decide to close the current buffer?
4. test ‘ `write more data now’ if ‘no’, I believe next time the flow diagram is executed, the ‘start’ task should immediately jump to this test again to have error free behaviour
5. in case of interrupt triggered operation the test on ‘Max_length’, if ‘yes’, flow should best return to the ‘yes’ output of the ‘close BD now’ test

5.2.2.5. IDMA- SCC1 receiver interaction

This paragraph describes a flow diagram that could be used as an example for the host driver interacting with the SCC1 receiver (Endian type = little)

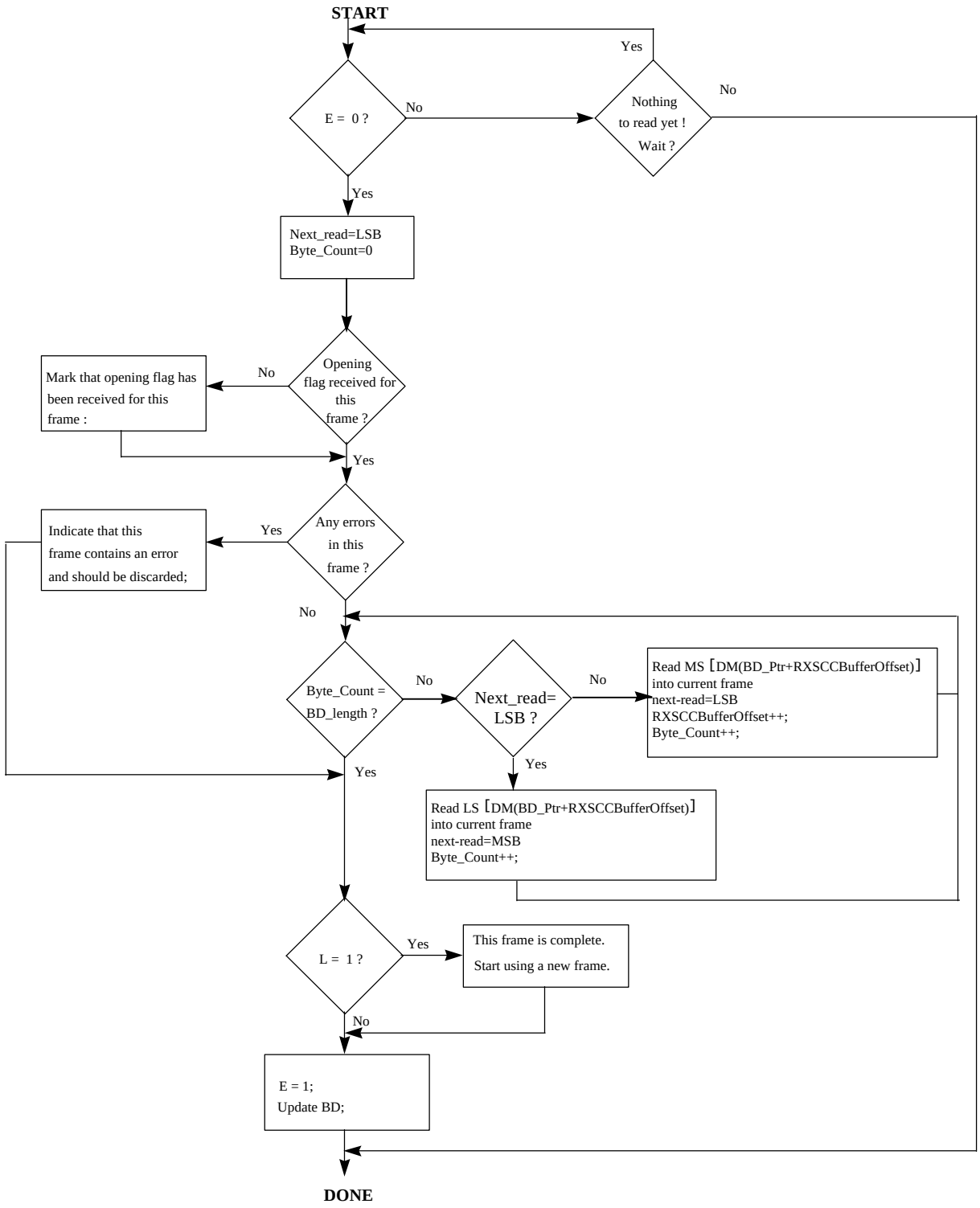


Figure 21 : IDMA- SCC1 receiver interaction for synchronous HDLC mode (little Endian)

SCC1 is directed towards the MODEM firmware, i.e. when the IDMA HOST wants to read from SCC1 (e.g. MODEM data) it will use the read buffers in SCC1.

The IDMA HOST has a variable containing the current BufferDescriptor (BD) to read from. This variable (called BD in the flow chart) is initiated to buffer no. 0. When this variable is updated the buffer no. is increased with 1 until it reaches 8[30], then it is set to 0 again. The IDMA HOST must always check that the E bit in the current BD is cleared (0) before it can start modifying the BD or its data. The IDMA HOST sets (1) this bit when all data has been read. The DSP clears this bit in the BufferDescriptors that are ready to be read from the IDMA HOST.

The IDMA HOST reads all data that belong to the same frame into a buffer. Only when a complete frame has been received it is send towards the application.
The sequence which the IDMA HOST should follow when reading from SCC1 is described in the flow chart in Fig.4. (describes the case of Polling mode)

Remarks:
1. the 'START' task can be interrupt triggered by an incoming event, indicating a new buffer is full
2. in interrupt triggered operation, checking the empty bit is allowed but not required, if the bit is set it would indicate an error condition. equally initiating a wait loop is not required
3. an opening flag is not essential for the correct operation and is not supported, likewise the test on the opening flag becomes obsolete

[30] *see previous remarks about the number of buffers*

5.2.3. Uart SCC

The character format of the Universal Asynchronous Receiver Transmitter (UART) protocol is shown below.

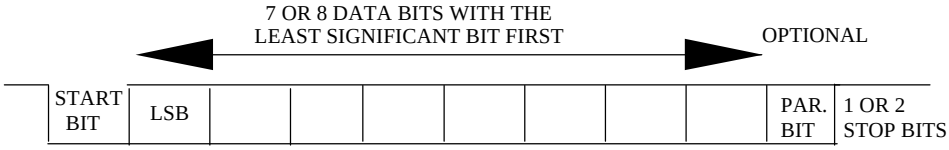


Figure 22 : character format of the UART

When data is not transmitted in the UART protocol, a continuous stream of ones is transmitted. This is called the idle condition. Since the start bit is always a zero, the receiver can detect when real data is once again present on the line. The UART also specifies an all-zeros character, called a break, which is used to abort a character transfer sequence.

By appropriately setting the SCC mode register, the SCC may be configured in V14 protocol. The latter provides the deletion of the stopbit, every 9 characters, at the transmitter to handle a difference between DTE- and DCE speed and thus to avoid overrun of the TX buffers.

The UART controller key features are as follows:

- Flexible Data Buffers
- V14 protocol
- Programmable Data Length (7 or 8 bits)
- Programmable 1 or 2 (not yet available) Stop Bits
- Even/Odd/Mark/Space/No Parity Generation
- Even/Odd/Mark/Space/No Parity Check
- Frame Error, Break and IDLE Detection
- Transmits Preamble and Break Sequences
- Freeze Transmission Option
- Flow Control Character Transmission Supported
- Alternating Transmission Of Data And IDLE characters Option

UART Channel Transmission Processing

When the host enables the UART transmitter, the controller will poll the current buffer descriptor (BD) in the transmit channel's BD table. When this BD is ready to transmit, the controller will fetch the data from memory (alternating LS- and MS- part of DM-word), add START, STOP and PARITY (according to the UART mode register) bits and start transmitting this UART frame (least significant bit of data first) to the ModemCore via the Transmit Flash Buffer. As soon as the stop bit of the current character is transmitted, the startbit of the following character is sent. When the end of the current BD has been reached and the last character is sent, the current BD is closed, the status bits are written and the next BD is fetched.

When the current BD is not ready for transmission, IDLE characters are sent. An IDLE character is defined as 9 to 12 consecutive ones (according to the UART mode register). For a given application, the number of bits in the idle character is calculated as follows:

1 + data length (either 7 or 8) + (1 if parity bit used) + number of stop bits (either 1 or 2)

When the V14 bit of the UART mode register is set, the controller will transmit every eight character without a stop bit (UART mode register 2STARTBITS = 0) or with only 1 stop bit (UART mode register 2STARTBITS = 1).

UART Channel Reception Processing

The UART receiver receives the incoming data from the ModemCore via the Receive Flash Buffer.

The length and the format of the serial word in bits are defined by the control bits in the UART mode register.
The order of reception is as follows:

Start Bit
Seven or Eight Data Bits with the Least Significant Bit First
Parity Bit (Optional)
Stop Bits

When the host enables the receiver, it thus waits for a START bit. Once the startbit is detected, the following bits are sampled, reversed in sequence until the STOP bit (second stop bit in case of 2 stop bits) of the complete character is detected. The UART controller fetches the current BD and, if empty, will transfer the incoming character to the BD's associated data buffer (alternating LS- and MS- part of DM-word).
If there is an error in this character, then the appropriate error bits will be set.

Only the data portion of the UART frame is actually stored in the data buffer. The start and stop bits are always generated and stripped by the UART controller. The parity bit may also be generated in the case of transmission, and checked during reception. Although parity is not stored in the data buffer, its value may be inferred by the reporting mechanism in the data buffer.

When the V14 bit of the UART mode register is set, the controller will receive every eight character without a stop bit or 1 stop bit when 2 stop bits are selected.
Reception of two characters without a stop bit, with less then 7 intermediate characters, results in a framing error.
Reception of two characters with only one stop bit (2 stop bits selected), with less then 7 intermediate characters, results also in a framing error.

UART Command Set

The channel **STOP TRANSMIT** command (UART mode register SENDBREAK=1) disables the transmission of characters on the transmit channel. If this command is received by the UART controller during data transmission (not yet implemented), transmission of that data is aborted. The UART transmitter will transmit a programmable number of break sequences and then reverts to idles or sends data if the RESTART TRANSMIT command was given before completion. The Transmit BD is not advanced. Upon transmission of the entire set of break characters, the transmitter sends at least one high bit before transmitting any data to guarantee recognition of a valid start bit.
A break is an all-zeros character with length equal to:

2 M + 3

where M is the number of bits between a start and a stop bit (according to the UART mode register).
The number of break sequences should be written to the TXBreakLength SCC field before this command is given to the UART controller. TXBreakLength is programmed with a value from 0 to 16383.
The STOP TRANSMIT command must be issued before the SCC mode register is used to disable the transmitter if the transmitter will be re-enabled at a later time.

Upon reception of a break-sequence by the UART receiver the current BD is closed if data is already stored in its buffer. The next BD is fetched which will be closed when the break is completely received. Its BDs Bufferlength will contain the number of break sequences received and the break bit of the BDs status is set. Data following the break will be stored in the following BD.

The channel **RESTART TRANSMIT** command (UART mode register SENDBREAK=0) re-enables the transmission of characters of the transmit channel. This command is expected by the UART after issuing a STOP TRANSMIT command and after issuing a STOP TRANSMIT and then disabling the channel using the SCC mode register.
The UART controller will resume transmission from the current transmitter BD.
If the transmitter is being re-enabled, the RESTART TRANSMIT command must be used and should be followed by the enabling of the transmitter in the SCC mode register.

The UART has the capability to recognise special control characters. Four control characters may be defined by the user in the SCC structure. Each of these characters may be either stored and cause no further action or cause dataflow to be controlled. In the latter case the character is rejected.

The **SENDXOFF** command (UART mode register bit A = 1) transmits the userdefined XOFF1 character (of the SCC structure). Transmission of the next character in the current BD is resumed directly after transmitting the XOFF1 character.

Upon reception by the UART receiver of the XOFF character, the SCC structure XOFF1 and XOFF2 are compared with the received character only when their bit 15 is set. When there is a match, the FREEZE bit of the UART mode register is set which causes IDLE characters to be transmitted. The XOFF character is not stored in the current buffer. When bit 15 is not set, the received character is considered to be a data character with no further meaning.

The **SENDXON** command (UART mode register bit 9 = 1) transmits the userdefined XON1 character (of the SCC structure). Transmission of the next character in the current BD is resumed directly after transmitting the XON1 character.
Upon reception by the UART receiver of the XON character, the SCC structure XON1 and XON2 are compared with the received character only when their bit 15 is set. When there is a match, the FREEZE bit of the UART mode register is reset which resumes data characters to be transmitted. The XON character is not stored in the current buffer.
When bit 15 is not set, the received character is considered to be a data character with no further meaning.

Flow-control characters may be transmitted at any time and are only considered as flowcontrol when UART mode register bit B (FLOWCONTROL XON/XOFF) is equal to 1.

5.2.3.1. The SCC Structure

The base address of the dceSCCstructure can be found in the location DCESCCstructPtr of data-pump read database. Table 11 : SCC1 in V14 MODE gives an overview of the usage of each field of the SCC structure by the CP, which runs on the DSP, and by the IDMA HOST.

Offset				IDMA host		DSP (CP)	
				R	W	R	W
0	ModeReg				X	X	X
	F		not used				
	E	FREEZE	if bit set SCC sends IDLE characters		X	X	X
	D	SENDBREAK	if bit set SCC sends break, the break time is placed in TXBreakLength		X	X	
	C	2STOPBITS	0 : 1 stopbit 1 : 2 stopbits (not yet available)		X	X	
	B	FLOWCONTROL XON/XOFF	1 : enables XON/XOFF flow control		X	X	
	A	SENDXOFF	if bit set SCC sends XOFF and clears bit		X	X	X
	9	SENDXON	if bit set SCC sends XON and clears bit		X	X	X
	8	V14	1 : use V14 syasy		X	X	
	7 6 5	PARITY	0 : no parity 1 : odd parity 2 : even parity 3 : space parity 4 : mark parity		X	X	
	4	ASYNC BITS	0 : 7 data bits 1 : 8 data bits		X	X	
	3	ENR	1: enable receiver (written by user in DATASTATE)		X	X	
	2	ENT	1 : enable transmitter (written by user in DATASTATE)		X	X	
	1, 0	MODE	00 : HDLC 01 : async (V14)		X	X	
1	Nbits		number of bits of per baud-package (bits/baud).			X	X
2	EventReg					X	X
3	TXState		During software booting this location is initialised with the address of the SCC transmitter routine. The SCC initialisation routine initialises this location with the initial value of the state variable of the CP transmitter During runtime the location contains the state variable of the CP transmitter			X	X
4	RXState		During software booting this location is initialised with the address of the SCC receiver routine. The SCC initialisation routine initialises this location with the initial value of the state variable of the CP receiver. During runtime the location contains the state variable of the CP receiver			X	X
5	TotalBuf Count						
6 .. 1D	RXBufferDescriptor[8*3]		Data associated with the SCC receiver is stored in buffers. Each buffer is referenced by a BufferDescriptor(BD). BD's are located in a BD array. The BD array allows to define up to eight buffers for each channel .	X	X	X	X
	offset 0	status and control					
	offset 1	data length					
	offset 2	data buffer pointer					

1E .. 35	TXBufferDescriptor[8*3]		Data associated with the SCC transmitter is stored in buffers. Each buffer is referenced by a BufferDescriptor(BD). BD's are located in a BD array.. The BD array allows to define up to eight buffers for each channel .	X	X	X	X
	offset 0	status and control					
	offset 1	data length					
	offset 2	data buffer pointer					
36	ModeReg2		indicates the number of buffers and the character operation mode of the buffers				
	F	TXinit	The TXBD's are initialised by the CP using the default values of MaxBufLength (= 260) and NbrofBufs (= 8). TXinit bit can be used by the user to reinitialise the TXBD's. He therefor has to disable the transmitter (ENT=0) first, then change MaxBufLength and NbrofBufs accordingly and set the TXinit and ENT bit. The TXinit bit is reset afterwards.		X	X	X
	E	RXinit	The RXBD's are initialised by the CP using the default values of MaxBufLength (= 260) and NbrofBufs (= 8). RXinit bit can be used by the user to reinitialise the RXBD's. He therefor has to disable the receiver (ENR=0) first, then change MaxBufLength and NbrofBufs accordingly and set the RXinit and ENR bit. The RXinit bit is reset afterwards.		X	X	X
	D	Nostartstopbit	if 1, in async mode no start bit and stop bit are generated. Nbits Baud packages are exchanged between the DTP and the RX- and TXSCCBuffers.		X	X	
	C 7	SPARE					
	6	TimeOut	1: closing the current RXBuffer, when a TimeOut has occured, is enabled. A TimeOut occurs when during the time specified in RXWorkReg1 (in msec) no characters are written in the current RXBuffer. Bit 2 of the current RXBD statusword will be set when this happens.		X	X	
	5	Endian type	this bit should be set or reset before booting of the page or before setting the TXinit and Rxinit bits. 1 : big endian for CharMode = 1 the MS-part of a DM location is accessed first, the LS-part secondly 0 : little endian (default value) for CharMode = 1 the LS-part of a DM location is accessed first, the MS-part secondly		X	X	
	4	CharMode	1: all buffers operate with two characters per location (see bit 5 Endian type for location accessing) 0: all buffers operate with one character per location (LS-part)	X			X
	3 .. 0	NbrofBufs	indicates the number of transmit buffers and receive buffers. The default number of the RX and TX data buffers is initialised by the modem software to : default number = 8.	X			X
37	MaxBufLength		the maximum length of an RX or TX buffer. The default size of the RX and TX data buffers is initialised by the modem software to : default size=130 memory locations.	X		X	X

38	RXSCCBuffer		indicates in which BufferDescriptor the receiver CP is writing data	X		X	X
39	RXSCCBufferOffset		indicates the number of received characters	X		X	X
3A	RXSCCBufferBitOffset						X
3B	RXSCCCRC					X	X
3C	RXEvent_Log		contains information., concerning filled receive buffers, which is used to update the eventbuffer.			X	X
	F .. 6	SPARE					
	5 .. 3	NoRXSCCBuffer	indicates the number of RXSCCBuffers, for which the eventbuffer has not yet been updated. Reset by the CP after updating the eventbuffer.			X	X
	2 .. 0	RXSCCBuffer	indicates the first RXSCCBuffer, in a sequence of length NORXSCCBuffer, for which the eventbuffer has not yet been updated. Reset by the CP after updating the eventbuffer.			X	X
3D	RXWorkReg1		contains the TimeOut value in milliseconds. Should be written by the user. Default = [0]		X	X	
3E	RXWorkReg2					X	X
3F	RXBreakLength		contains the number of received break characters	X			X
40	TXSCCBuffer		indicates from which BufferDescriptor the transmitter is sending data	X		X	X
41	TXSCCBufferOffset		indicates the current transmit memory word	X		X	X
42	TXSCCBufferBitOffset						
43	TXSCCCRC						
44	TXSCCCRC1						
45	TXBitsCalculated		contains the number of calculated transmitbits			X	X
46	TXEvent_Log		contains information., concerning transmitted buffers, which is used to update the eventbuffer.			X	X
	F .. 6	SPARE					
	5 .. 3	NOTXSCCBuffer	indicates the number of TXSCCBuffers, for which the eventbuffer has not yet been updated. Reset by the CP after updating the eventbuffer.			X	X
	2 .. 0	TXSCCBuffer	indicates the first TXSCCBuffer, in a sequence of length NOTXSCCBuffer, for which the eventbuffer has not yet been updated. Reset by the CP after updating the eventbuffer.			X	X
47	TXWorkReg1					X	X
48	TXWorkReg2					X	X
49	TXPtr		During runtime the location points to current DM word (2 characters per word)			X	X
4A	TXBreakLength		should contain the number of break characters to be transmitted		X	X	X
4B	XON1		(bits 0 - 7) : XON1 character (bit 15) 1 : reception of XON1 resumes DATA transmission, character is discarded 0 : no action, character is stored in buffer		X	X	
4C	XON2		(bits 0 - 7) : XON2 character (bit 15) 1 : reception of XON2 resumes DATA transmission, character is discarded 0 : no action, character is stored in buffer		X	X	
4D	XOFF1		(bits 0 - 7) : XOFF1 character (bit 15) : 1 reception of XOFF1 causes IDLE transmission (FREEZE), character is discarded 0 : no action, character is stored in buffer		X	X	

4E	XOFF2	(bits 0 - 7) : XOFF2 character (bit 15) : 1 reception of XOFF2 causes IDLE transmission (FREEZE), character is discarded 0 : no action, character is stored in buffer		X	X	
4F	BreakAsyncRegHigh				X	X
50	BreakAsyncRegLow				X	X
51	BreakMask				X	X
52	nbitsTx	determines the number of bits per baudpackage at the transmitter side.			X	X

Table 11 : SCC1 in V14 MODE

5.2.3.2. UART receive BufferDescriptor (RX bd)

The UART controller uses the RX BD to report information about the received data for each buffer. The RX BD is shown in figure …

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET +0	E	-	W	-	-	-	-	-	-	TO	BR	FR	PR	-	-	-
OFFSET +2	DATA LENGTH															
OFFSET +4	RX BUFFER POINTER															

Figure 23 : UART Receive BufferDescriptor

An example of the UART receive process is shown in figure … This figure shows the resulting state of the Rx BDs after receipt of 10 characters, an idle period, and five characters, one with a framing error. The example assumes that MaxBufLength = 8.

The UART controller saves received characters alternating in both bytes of a DM- word, starting with the least significant part (Endian = little). After writing two characters in the LS- and MS-Part of the current DM-word, the next character is written in the LS-part of the next DM-word. Therefore when the host reads the BD's associated buffer he also has to start with the LS-part of the DM- word pointed to by the BD's associated datapointer. The second received character will be in the MS-part of the same DM-word. The third character in the LS-part of the second DM-word and so on...

The first word of the RX BD contains control and status bits. Bits 15 and 13 are written by the user before the buffer is linked to the RX BD table, and bits 5 - 3 are set by the CP following reception. Bit 15 is set by the user when the buffer is available to the UART controller; it is cleared by the UART controller when the buffer is full.

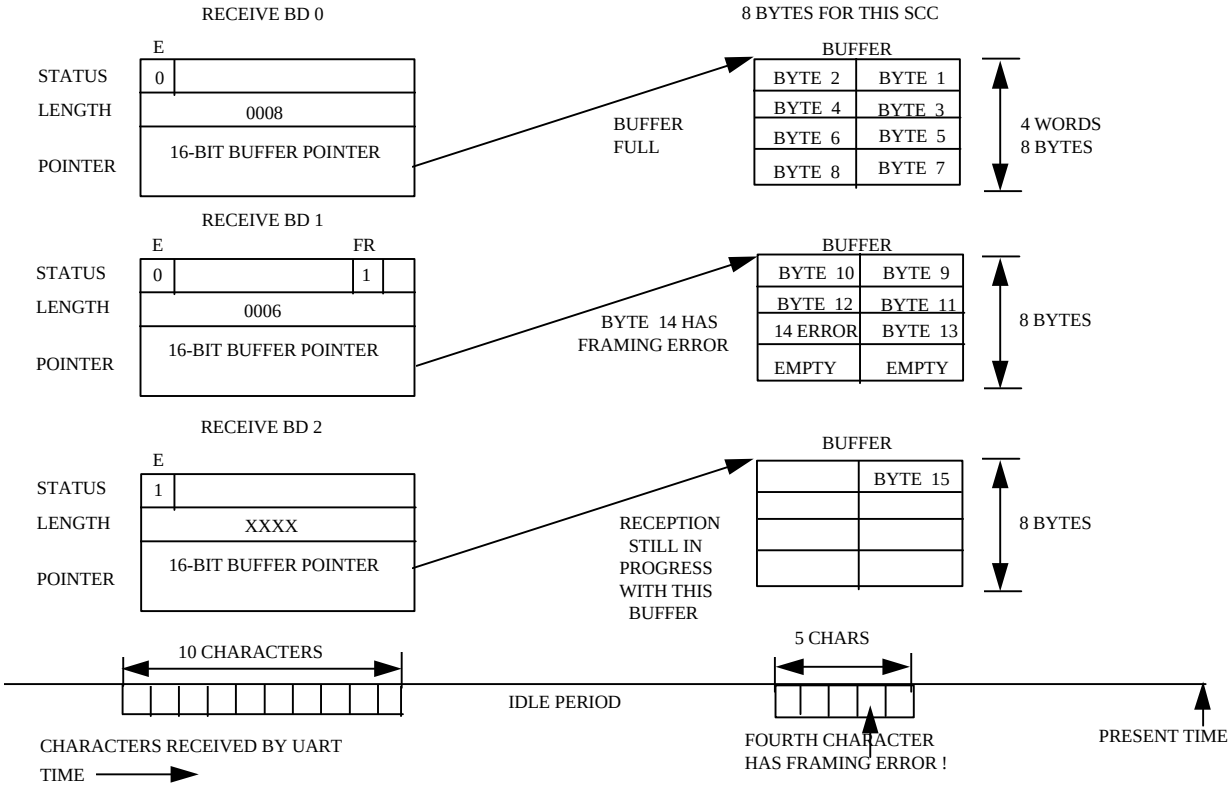


Figure 24 : UART Receive BD Example (little Endian)

E - Empty

0 = The data buffer associated with this BD has been filled with received data, or data reception has been aborted due to an error condition. The user is free to examine or write to any fields of the BD.
1 = The data buffer associated with this BD is empty. This bit signifies that the BD and its associated buffer are available to the UART controller. The user should not write to any fields of this BD after it sets this bit. The empty bit will remain set while the controller is currently filling the buffer with received data.

W - Wrap (Final BD in Table)

0 = This is not the last BD in the RX BD table.
1 = This is the last BD in the RX BD table. After this buffer has been used, the controller will receive incoming data into the first BD in the table.

NOTE

The user is required to set the wrap bit in one of the first eight BDs; otherwise, errant operation may occur.

The following status bits are written by the HDLC controller after the received data has been placed into the associated data buffer.

Bits 9-7 - Reserved for future use.

TO - Time Out

This bit indicates that the BD was closed due to no reception of characters during 200 ms. The Time Out facility has to be enabled by setting bit 6 of the ModeRegister2.

BR - Break Received

A break sequence was received while receiving data into this buffer.

FR - Framing Error

A character with a framing error was received and is located in the last byte of this buffer. A framing error is detected by the UART controller when no stop bit is detected in the receive data string and the previous character without a stop bit was received less then 7 characters before.

PR - Parity Error

A character with a parity error was received and is located in the last byte of this buffer.

Data Length

The data length is the number of octects (using both bytes in a DM word) written to this BD's data buffer by the UART controller. When the BR bit is set, data length contains the number of breakcharacters received before the next startbit is received. The current buffer contains no characters.
When the TO bit is set, data length contains the number of characters received before the Time Out occured.

RX Buffer Pointer

The receive buffer pointer, which always points to the first location of the associated data buffer, may reside in either internal or external memory.

NOTE

The RX buffer pointer must be even.

5.2.3.3. UART transmit BufferDescriptor (TX bd)

Data is presented to the UART controller for transmission on an SCC channel by arranging it in buffers referenced by the channel's TX BD table. The UART controller confirms transmission (or indicates error conditions) using the BDs to inform the user that the buffers have been serviced. The TX BD is shown in figure …

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET +0	R	-	W	-	-	-	P	AI	-	-	-	-	-	-	-	-
OFFSET +2	DATA LENGTH															
OFFSET +4	TX BUFFER POINTER															

Figure 25 : UART Transmit BufferDescriptor

The UART controller transmits both bytes of every word in the TX buffer, starting with the least significant part (Endian = little). After transmitting the character in the MS-Part, the CP changes to the LS-part of the next DM-word. Therefor when the host writes to the BD's associated buffer he has to start with the LS-part of the DM- word pointed to by the BD's associated datapointer. The second transmit character has to be written in the MS-part of the same DM-word. The third character in the LS-part of the second DM-word and so on...

The first word of the RX BD contains control and status bits. Bits 15 and 13 are written by the user before the buffer is linked to the RX BD table, and bits 5 - 3 are set by the CP following reception. Bit 15 is set by the user when the buffer is available to the UART controller; it is cleared by the UART controller when the buffer is full.

The first word of the TX BD contains status and control bits. All bits are prepared by the user before transmission. Bit 15 is set by the user when the buffer and BD have been prepared and is cleared by the controller after the buffer has been transmitted.

R - Ready

0 = This buffer is not currently ready for transmission. The user is free to manipulate this BD (or its associated buffer). The controller clears this bit after the buffer has been fully transmitted or after an error condition has been encountered.
1 = The data buffer, which has been prepared for transmission by the user, has not yet been transmitted. No fields of this BD may be written by the user once this bit is set.

W - Wrap (Final BD inTable)

0 = This is not the last BD in the TX BD table.
1 = This is the last BD in the TX BD table. After this buffer has been used, the controller will transmit data from the first BD in the table.

NOTE

The user is required to set the wrap bit in one of the first eight BDs; otherwise, errant behavior may occur.

P - Preamble

A preamble sequence gives the programmer a convenient way of ensuring that the line goes idle before transmitting new data. The preamble sequence length is 9 to 12 consecutive ones (depending on the UART mode register).
0 = No preamble sequence is sent.
1 = The UART sends one preamble sequence (IDLE character) before sending the data.

AI - Alternating IDLE transmission

0 = No IDLE character is transmitted between every two datacharacters.
1 = Between transmission of every two characters a IDLE character is sent.
Bit 7 is used by the controller as a switch, so this bit may not be changed by the user.

Data Length

The data length is the number of octects (not the number of DM-words) that the CP should transmit from this BD's data buffer. It is never modified by the CP. This value should be normally greater than zero.

TX Buffer Pointer

The transmit buffer pointer points to the first location of the associated data buffer and may be even or odd. The buffer may reside in either internal or external memory.

5.2.3.4. IDMA - SCC1 transmitter interaction

This paragraph describes a flow diagram that could be used as an example for the host driver interacting with the SCC1 transmitter

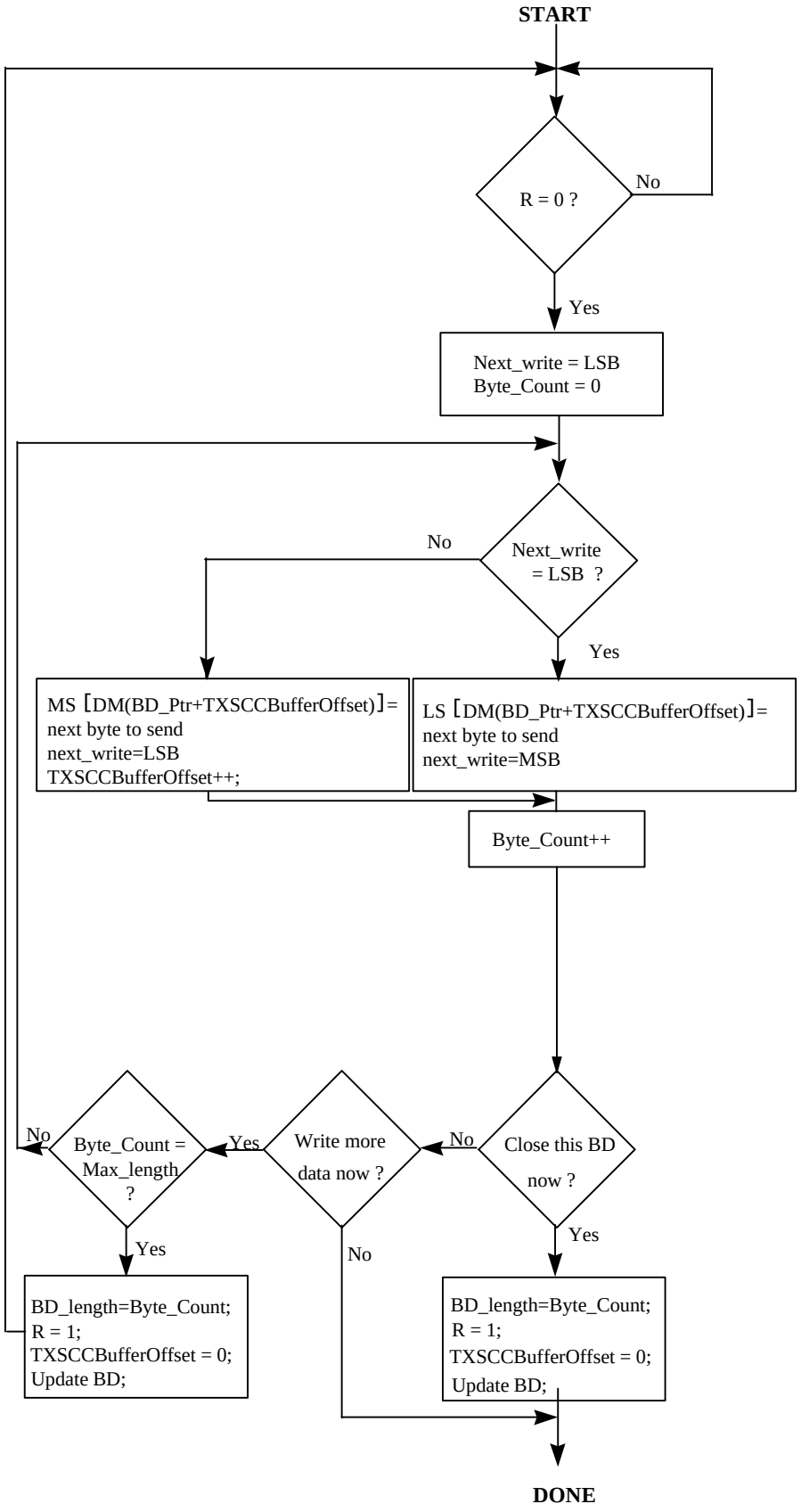


Figure 26 : IDMA - SCC1 transmitter interaction (little Endian)

SCC1 is directed towards the MODEM firmware, i.e. when the IDMA HOST wants to write to SCC1 (e.g. MODEM data) it will use the write buffers in SCC1.
The IDMA HOST has a variable containing the current BufferDescriptor (BD) to write to. This variable (called BD in the flow chart) is initiated to buffer no. 0. When this variable is updated the buffer no. is increased with 1 until it reaches 8, then it is set to 0 again. The IDMA HOST must always check that the R bit in the current BD is cleared (0) before it can start modifying the BD or its data. The IDMA HOST sets (1) this bit when all data has been written to this BD. The DSP clears this bit in the BufferDescriptors that are ready to be filled by the IDMA HOST.

The sequence which the IDMA HOST should follow when writing to SCC1 is described in the flow chart in Fig.5. (describes the case of Polling mode)

Remarks:
1. The ‘start’ task can be interrupt triggered by an incoming event indicating a buffer is empty
2. in interrupt triggered operation, checking the ready bit is allowed but not required, if the bit is set it would indicate an error condition. Equally initiating a wait loop in not required.
3. test ‘close this BD now’ :on what basis will the IDMA HOST decide to close the current buffer?
4. test ‘ `write more data now’ if ‘no’, I believe next time the flow diagram is executed, the ‘start’ task should immediately jump to this test again to have error free behaviour
5. in case of interrupt triggered operation the test on ‘Max_length’, if ‘yes’, flow should best return to the ‘yes’ output of the ‘close BD now’ test

5.2.3.5. IDMA- SCC1 receiver interaction

This paragraph describes a flow diagram that could be used as an example for the host driver interacting with the SCC1 receiver

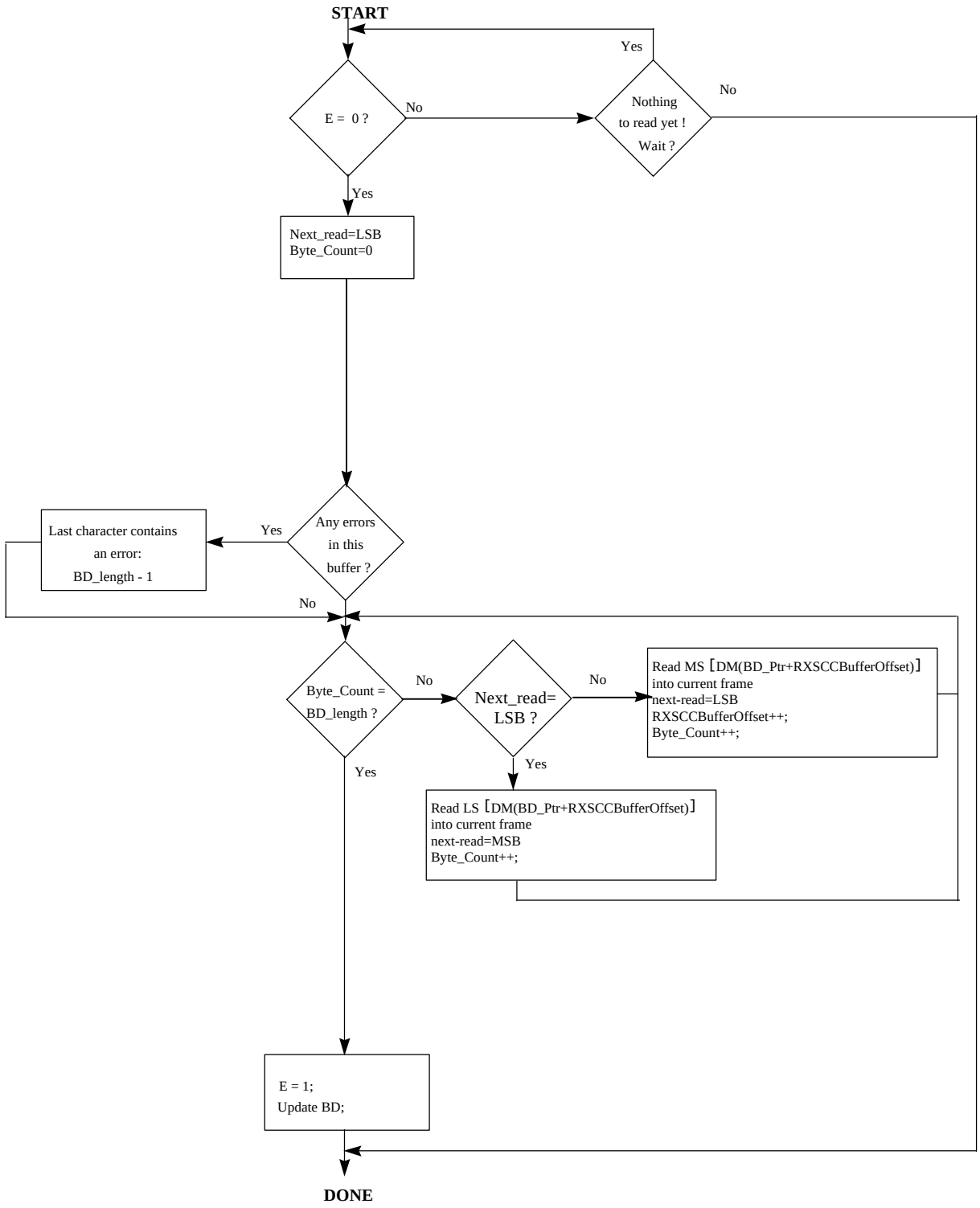


Figure 27 : IDMA - SCC1 receiver interaction (little Endian)

SCC1 is directed towards the MODEM firmware, i.e. when the IDMA HOST wants to read from SCC1 (e.g. MODEM data) it will use the read buffers in SCC1.
The IDMA HOST has a variable containing the current BufferDescriptor (BD) to read from. This variable (called BD in the flow chart) is initiated to buffer no. 0. When this variable is updated the buffer no. is increased with 1 until it

reaches 8[31], then it is set to 0 again. The IDMA HOST must always check that the E bit in the current BD is cleared (0) before it can start modifying the BD or its data. The IDMA HOST sets (1) this bit when all data has been read. The DSP clears this bit in the BufferDescriptors that are ready to be read from the IDMA HOST.

The sequence which the IDMA HOST should follow when reading from SCC1 is described in the flow chart in Fig.4. (describes the case of Polling mode)

Remarks:
1. the 'START' task can be interrupt triggered by an incoming event, indicating a new buffer is full
2. in interrupt triggered operation, checking the empty bit is allowed but not required, if the bit is set it would indicate an error condition. equally initiating a wait loop is not required

[31] *see previous remarks about the number of buffers*

5.3. CONTROL INTERFACE

This section gives an overview of all locations of the memory mapped interface.

There are 256 consecutive locations in the DSP internal memory. The address of the first location is h#3EE0.

The offsets of the locations are given together with their meaning. If different from zero, the default value is indicated between square brackets [B15…B0] in binary or hexadecimal.

The offset range [0..127] covers all locations that are written by the HOST (*write* database).
The offset range [128..255] covers all locations that are written by the DSP (*read* database).

Important remarks :

1. All parameters expressed in dBm require a calibration of the analogue front end hardware.
2. Text in *blue italics* concerns aster4 flash data-pump interface locations and bits, of no importance to the user of this users guide
3. Text in *red italics* concerns typical modulation core variables, of no importance to the user of this users guide (only for the modulator core integrator)
4. undefined and unused locations and bits are indicated with 'SPARE'
5. undefined, but used locations and bits are indicated with 'RESERVED'

5.3.1. data-pump write database

00	GEN_setup0		training defining parameters default:[0]	
	F	modemtype	hardware identification, determining the kernel drivers if 1, freesia if 0, aster4	
	E	InverseLoop	if 1, the received analog signal is looped back to the transmitter.	
	D	ALexternal	if 1, in analog loop operation mode the loop is not closed inside the DSP, it's up to the user to close the loop in hardware	
	C	EOT		
	B	DLremEnable		
	A	ALOnLine	if 1, signal will be transmitted by CODEC during analog loop (for PTT test purposes)	
	9	*Ncarrier*	*if 1, the normal carrier frequency group is selected, otherwise the german carrier frequency group is selected. (Only in Tfast)*	
	8	*spare*		
	7	ext_trn	If 0, the last TRN segment of the training will be limited to 1280 symbols, else it will be extended to 8192 symbols (Only in V.32bis or V.34)	
	6	PSTN/LL	PSTN/Leased Line selection. (Only for V.22, V.22bis and V.34, FSK) 1: the pump is working in PSTN mode. 0: the pump is working on leased line.	

	5, 4	guard	0 : no guard tone 1 : 550 Hz guard tone 2 : 1800 Hz guard tone (Only for V.22bis)	
	3, 2	compromise equaliser coefficient	0,0:disabled 0,1:normal 1,0:reduced 1,1:not used (Only in V.32bis)	
	1	DP	If 1, no phase jumps will be introduced in the V25 answer tone	
	0	*spare*		

01	GEN_setup1		operation mode parameters default:[0]	
	F	*s/p interface*	*hardware indication (Only for aster 4 flash)* *If 1, then the main data channel interface is serial*	
	E	*spare*		
	D	*FLASH*	*if 1, static ram boot mode (Only for freesia)*	
	C	*SLAVE*	*If 1, SLAVE RECEIVE clock operation will be selected for the CODEC*	
	B	*EXT*	*If 1, then External clock operation will be selected for the CODEC*	
	A	*INTCL*	*If 1, Internal clock operation will be selected for the CODEC*	
	9	AL	If 1, the modem will realise loop 3 : Analog Loop Back	
	8	S140	If 1, request of remote loop DL: Digital Loop Back (Only V.22 or V.22bis)	
	7	NORM OPERATION MODE	Setting this bit will start a operation sequence for the norm specified in Norm_L, Norm_H	
	6	DIAL OPERATION MODE	Setting this bit activates the dialler functions on the idle/dial page	
	5	IDLE OPERATION MODE	Setting this bit will start a return operation to the idle page	
	4	DISCONNECT OPERATION MODE		
	3	CH	Channel selection, 1=call or originate channel, 0=answering	
	2	WI	If 1, 2 wire operation is selected, else 4 wire	
	1	Dasen	If 0, the pump will send an answer tone when put in norm operation mode	
	0	ADET	If 1, the pump will wait for the 2100Hz answer tone before training If 0, the pump will not wait for the 2100Hz answer tone before training	

02	GEN_setup2		default:[0]	
	F, E	*spare*		
	D	DialBit	If 1, the autodialler will start the dial operation	
	C	*spare*		
	B	*spare*		
	A	*spare*		
	9	*DLL Enable*	*if 1, a down line loading session is requested (Only in V.32bis)*	
	8	Sec Ch Enable	if 1, the secondary channel is enabled (Only in V.32bis and V.34)	
	7	Ratechange Disable	if 1, the default active ratechange renegotiation handshake of V.32bis is disabled (Only in V.32bis)	
	6	TonedetEnable	If 1, the tone detection filters, described below, are enabled (Currently Only on the dial page and V.22bis)	
	5	Stepup Enable	If 1, the modem is allowed to initiate a retrain or ratechange with enabling of all (by the user in speedsel_l, speedsel_h enabled) bitrates higher then the current bitrate, if the signal quality is better then the signal quality step up threshold value	
	4	Fallback Enable	If 1, the modem is allowed to initiate a retrain or ratechange with enabling of all (by the user in speedsel_l, speedsel_h enabled) bitrates lower then the current bitrate, if the signal quality drops below the signal quality fall back threshold value	
	3	Retrain Enable	If 1, the modem is allowed to initiate a retrain or ratechange with enabling of only the current bitrate, if the signal quality drops below the signal quality fall back threshold value	
	2	*spare*		
	1	RATECH	This bit is only effective during data-state. When changed from 0 to 1 it initiates a rate change procedure	
	0	RETRAIN	This bit is only effective during data-state. When changed from 0 to 1 it initiates a retrain.	

Note: The Aster4 strapping for the parameter ‘retrain on signal quality ‘ corresponds to following settings of the retrain,fallback and stepup bit:

aster 4 setting	database setup
disabled	gen_setup2=0000
retrain	gen_setup2=0008
fallback	gen_setup2=0018
fallb+setup	gen_setup2=0038

03	DISP_setup	See appendix 6.6	

			default:[0000]	

04	V8_setup		default:[0000]	
	F	V90_APCM	if 1, analogue side V.90 modulation is enabled	
	E	V90_DPCM	if 1, digital side V.90 modulation is enabled	
	D	V90_network	if 1, digital network connection if 0, analogue network connection	
	C	H324	if 1, then H324 capability is signalled in V.8	
	B	CI	if 1, in calling mode the CI tone will be generated, in answermode the CI tone detection is enabled. (Only for V.8)	
	A, .. 0	*spare*		

05	FAX_setup			
	F, .. 3	*spare*		
	2	Fax_TEP_Enable	0 : No talker echo protection	
	1	Fax_Short_Train	0 : long Train	
	0	Fax_Transmit	0: receive, 1: transmit	

06	V34_setup		default:[0000]	
	F, .. C	*spare*		
	B	ManPower	if 1, manual power reduction is enabled. This uses following database locations : • MinReduction_dbs, • AddReduction_dbs	
	A	NoPreemph	if 1, preemphasis is not allowed	
	9	syncret	if 1, retrain on sync loss is disabled	
	8	asymm	if 1, asymmetric bitrates are allowed(Currently not supported)	
	7	*spare*		
	6		if 1,optimise RX throughputdelay in case of asymmetric bit rates (Currently not supported)	
	5		if 1, optimise TX throughputdelay in case of asymmetric bit rates (Currently not supported)	

	4		if 1, optimise throughputdelay (for RX) (Currently not supported)	
	3, 2	BERtype	BER type for Retrain 2:high BER 1:medium BER 0:low BER	
	1, 0	Secbps	selection of 200bps (00) or 400bps (01) for the secondary channel	

07	Info0_setup		default:[F0FD]	
	F	AbRX2743H	if 1, ability to receive at 2743 H	
	E	AbRX2743L	if 1, ability to receive at 2743 L	
	D	AbRX2400H	if 1, ability to receive at 2400 H	
	C	AbRX2400L	if 1, ability to receive at 2400 L	
	B	noPrecoding	if 1, Precoding is disabled	
	A 9	Trellis decoder selection	00:for high and medium BER 16 state trellis decoder, for low BER 32 state trellis decoder 01:fixed 16 state trellis decoder 10:fixed 32 state trellis decoder 11:fixed 64 state trellis decoder	
	8	reduce	if 1, ability to reduce the transmit power to a value lower than the nominal setting	
	7	AllTX3429	if 1, transmission at a symbolrate of 3429 is allowed	
	6	AbTX3200H	if 1, ability to transmit at the high carrier frequency with symbolrate 3200	
	5	AbTX3200L	if 1, ability to transmit at the low carrier frequency with symbolrate 3200.	
	4	AbTX3000H	if 1, ability to transmit at the high carrier frequency with symbolrate 3000.	
	3	AbTX3000L	if 1, ability to transmit at the low carrier frequency with symbolrate 3000.	
	2	Sup3429	if 1, 3429 symbolrate is supported	
	1	Sup2800	if 1, 2800 symbolrate is supported	
	0	Sup2743	if 1, 2743 symbolrate is supported	

08	TD	4 LSB:TX level (0 -> 0 dB, F -> -15 dB) of transmit signal default:=[06]	
09	TA	4 LSB :TX level of single tones (answer tone, dial tone,…) default=[06]	
0A	TX_LEVEL_TUNE	2 dB tuning of all TX levels according to the formula TXout = TX(0.8 + X) with 0 X < 0.2 (H#FF) default=[FF]	
0B	DCD_OFF	If the energy level of the received signal drops below the DCD_OFF threshold for a fixed period of time, the ENDT of read database location rstatus will be set to zero 0(0)..-255(H#FF) dBm default=[30]	

0C	DCD_HYST		If after a DCD drop, the energy level of the received signal rises above the DCD_On threshold (=DCD_off + DCD_hystersis) for a fixed period of time, the ENDT bit of read database location rstatus will be set to one. 0(0)..255(H#FF) dB default=[03]	
0D	TXD		TXD is a 16 bit location containing the main channel transmit data, b15 of TXD is the oldest bit i.e the first bit sent from host to DSP, b0 is the last bit sent. This implies that the data bits are left aligned on location b15 (only for ASTER4 FLASH)	

0E	WSTATUS		this location contains the write database status. Default:	
	F	SECTXPR	if 1, a secondary data bit is present	
	E	SECTXD	secondary data bit	
	D	change_wdb	if 1, then compared to the previous write communication cycle the value of one or more of the control database locations has changed (all locations except data)	
	C	bootfinished	if 1, then the booting of new data in DM or PM by the HOST, is completed	
	B, .. 5	*spare*		
	4	*change_TXdog*	*If 1, then compared to the previous write communication cycle the value of TXdog[0..3] has changed. The DSP mainprogram reads the new value, clears the changebit and increments the value modulo 16. The result is returned in the read database location RXdog[0..3] at the next RXbaud interrupt.*	
	3	*TXdog[3]*		
	2	*TXdog[2]*		
	1	*TXdog[1]*		
	0	*TXdog[0]*		

0F	*P2SD*		*P2SD is a 16 bit location containing the input data to the parallel to serial conversion module, b15 of P2SD is the oldest bit i.e the first bit sent from host to DSP, b0 is the last bit sent. This implies that the data bits are left aligned on location b15 (only for ASTER4 FLASH)* *Only if the s\p interface bit of gen_setup1 is zero, the value in P2SD will be converted. If the s\p interface bit is one, the value in RXD will be used as input for the parallel to serial converter.* *Default =[ABCD]*	

10 .. *22*	*reserved*	The locations in this range contain low level dialler interface registers. More information about low level dialler functions can be found in a related document called 'DIALLER'.	

23	MultiPlFromHost	V33 Multiplexer Configuration to be transmitted (numbering according CCITT V33, Table 5A & 5B)	
24	delaycorrection	The Tuning procedure of the modem with respect to the supplementary buffering of the kernel is : 1)setup a 'back to back' training session with a calibrated modem (e.g. your aster4) 2)note the value of the roundtripcntr: this value should be equal to zero for a calibrated modem in a back to back connection. 3)give a compensation by filling in this value in 'delay-correction' when you do the initialisation for the 'info' page. In this way your modem becomes calibrated: all the requirements concerning exact timings of the training are met and also the delay line length for the near echo canceller is correct. **Important remark** : In the case of V32 the delay-correction has to be a multiple of 3.	
25	*TXD0*	*TXD0 is a 16 bit location containing the main channel transmit data, b15 of TXD0 is the oldest bit i.e the first bit sent from host to DSP, b0 is the last bit sent. This implies that the data bits are left aligned on location b15. If the TX request bit of DI_control is set to one, the modem core asks the kernel to give a new 2400 Hz baud data packet (in the TXD0-location); this bit is cleared by the DSP after arrival of the packet.* *Incase of V90D operation :* *TXD0, b0 is the oldest bit of the Datagram, consecutive bit locations are filled up until b15, further bits are put in TXD1 and TXD2, Datagram package size can vary from 21 to 42*	
26	*TXD1*	*TXD1, b0 is the sixteenth bit of the Datagram, consecutive bit locations are filled up until b15, further bits are put in TXD2, , Datagram package size can vary from 21 to 42 (This location is only in use for V90D)*	

27	*TXD2*	*TXD2, b0 is thirtysecond bit of the Datagram, consecutive bit locations are filled up until b15, Datagram package size can vary from 21 to 42 (This location is only in use for V90D)*	

28	Norm_H	default=[0]	
	0001 0002 0004 0008 0010 0020 0040 0080 0100 0200 0400 0800 1000 2000 4000 8000	V8 V110 V18 speakerphone Low Level unload boot	

29	Norm_L		
	0001 0002 0004 0008 0010 0020 0040 0080 0100 0200 0400 0800 1000 2000 4000 8000	V21 V22 V22B V23 BEL212A BELL103 V21ch2 V29FDX V34 V27Ter V29 V17 V32ext V32bis V33 V90	

note:

- V.32 is a subset of V.32ext with only 4800 and 9600U selected
- V32ext has the same speed range as V32bis but does not support the ‘rate change procedure’
- AUTO and AutoV8 will select 9600 and 9600U if 9600 belongs to the range of speeds selected by the user, according to V.32bis

2A	speed_sel_h	This location contains the high word of the speed rate capability mask, which determines the speed range of transmitter and receiver. In case of asymmetrical operation also MAXTXSPEED, MAXRXSPEED have to be used. default=[0]	
	0001 0002 0004	24000 26400 28800	

	0008 0010 0020 0040 0080 0100 0200 0400 0800 1000 2000 4000 8000	31200 33600 9600U 	

2B	speed_sel_l	This location contains the low word the speed rate capability mask, which determines the speed range of transmitter and receiver. In case of asymmetrical operation also MAXTXSPEED, MAXRXSPEED have to be used. default=[0]	
	0001 0002 0004 0008 0010 0020 0040 0080 0100 0200 0400 0800 1000 2000 4000 8000	cleardown 75 110 150 300 600 1200 2400 4800 7200 9600 12000 14400 16800 19200 21600	

2C	Maxtimer	Period during which the MSE should be greater than the internal fallback value in order to cause a fallback. • Unit=1sec • Range: (0: -> 255 sec (h#FF)) note: maxtimer=0 will cause an immediate retrain when the modem arrives in DATASTATE default=[0]	
2D	Mintimer	Period during which the MSE should be smaller than a the internal stepup value in order to cause a stepup • Unit=6sec • Range: (0:-> 25.5 Min (h#FF)) note:mintimer=0 will cause an immediate retrain when the modem arrives in DATASTATE default=[0]	

note: aster4 settings:

maxtimer=10h (16 seconds)

mintimer=14h (120 seconds)

2E	rebootop_	default=[0]	
2F	shellinptr	default=[0]	

The following locations contain.

30 .. 4F	reserved	The locations in this range contain information for supervisory tone detection. More information on supervisory tone detection can be found in a related document called 'DIALLER'.	

50	*RXSAMPLE_0*	*At symbol rate the kernel writes 3,4 or 5 samples in the RXSAMPLE-buffer. The number of samples equals the ratio of sample- and symbolrate. This ratio can be found in the read database location 'Samplebuffersize'.*	
51	*RXSAMPLE_1*		
52	*RXSAMPLE_2*		
53	*RXSAMPLE_3*		
54	*RXSAMPLE_4*		
55	*RXSAMPLE_5*		
56	*BulkOutNeaRX*	*At symbolrate the kernel fetches the X sample of the Near Bulk output point of the bulk delay line and puts it in this location.*	
57	*BulkOutNearY*	*At symbolrate the kernel fetches the Y sample of the Near Bulk output point of the bulk delay line and puts it in this location.*	
58	*BulkOutpuTX*	*At symbolrate the kernel fetches the oldest X sample of the Bulk delay line and puts it in this location.*	
59	*BulkOutputY*	*At symbolrate the kernel fetches the oldest Y sample of the Bulk delay line and puts it in this location.*	
5A..63	*spare*		
64..6E	*reserved*		
6F	Bootcoreptr	location indicating the starting address of the boot loader module.	read

70	Sp0CntrlReg	This location has an identical bit setting as the SPORT0 Control Register of the ADSP2181 Used to set the multichannel operation for Serial Port 0. This location must be initialised after system reset before the start-up page is loaded.	
71	Sp0MCRecL	Location with identical bit setting as the SPORT0 Multichannel Receive Word Enables, bits 0-15. This location must be initialised after system reset before the start-up page is loaded.	
72	Sp0MCRecM	Location with identical bit setting as the SPORT0	

		Multichannel Receive Word Enables, bits 16-31 This location must be initialised after system reset before the start-up page is loaded.	
73	Sp0MCTXL	Location with identical bit setting as the SPORT0 Multichannel Transmit Word Enables, bits 0-15 This location must be initialised after system reset before the start-up page is loaded.	
74	Sp0MCTXM	Location with identical bit setting as the SPORT0 Multichannel Transmit Word Enables, bits 16-31 note: These 4 last locations are used to program whether the words in the different channels in the multichannel operation of SPORT0 are enabled or not. This location must be initialised after system reset before the start-up page is loaded.	

75	MinReduction_dbs	Location specifying the minimum transmission level reduction that remote modem should install compared to is original nominal level, the reduction request is only send to the remote modem if the bit Manpower of V34_setup is set .	
76	AddReduction_dbs	Location specifying the additional transmission level reduction that remote modem may install, only active if the bit Manpower of V34_setup is set.	

77	DATACONFIG			write
	F, E,	data-pump [32]	0 : use SCC1 data 1 : use IDMA TXD write buffers and RXD read buffers 2 : not used 3 : not used	
	D .. 2	Spare		
	1, 0	SCC2 DTE side[33]	0 : SCC2 parallel asynchronous interface through IDMA 1 : SCC2 is connected to SCC 3 2 : not used 3 : not used	

78	V34SLOT	For a TDM line interface V34SLOT indicates the SPORT0 slot used for V34 modemoperation. V34SLOT should be initialised before ModemOperation is initiated in the DIAL page. 0..31 default=[0]	

[32] data-pump use SPORT1 : this solution is hard-coded in a specific software package
[33] SCC2 uses the serial interface or UART: this solution is hard-coded in a specific software package

79	speed_sel_V90_H	In case of V.90 operation this location determines the speed range of V90D transmitter or V90A receiver. In case of asymmetrical operation also MAXTXSPEED_V90, MAXRXSPEED_V90 have to be used. Default=[0]	
	0001 0002 0004 0008 0010 0020 0040 0080 0100 0200 0400 0800 1000 2000 4000 8000	49000+1000/3 50000+2000/3 52000 53000+1000/3 54000+2000/3 56000	

7A	speed_sel_V90_L	In case of V.90 operation this location determines the speed range of V90D transmitter or V90A receiver. In case of asymmetrical operation also MAXTXSPEED_V90, MAXRXSPEED_V90 have to be used. Default=[0]	
	0001 0002 0004 0008 0010 0020 0040 0080 0100 0200 0400 0800 1000 2000 4000 8000	28000 29000+1000/3 30000+2000/3 32000 33000+1000/3 34000+2000/3 36000 37000+1000/3 38000+2000/3 40000 41000+1000/3 42000+2000/3 44000 45000+1000/3 46000+2000/3 48000	

7B	Info0D_setup			
	F			
	E			
	D			
	C			

	B			
	A			
	9 ..8	lookaheadmax	A number between 1 and 3 indicating the digital modem's maximum lookahead for spectral shaping.	
	7	upstream3429	Set to 1 indicates the ability to operate V.90 with an upstream symbolrate of 3429	
	6	PCMcoding	PCM coding in use by the digital modem. 0=uLaw, 1=aLaw	
	5	MeasurePoint	Set to 1 indicates the digital's modem's power shall be measured at the output of the codec. Otherwise the digital modem's power shall be measured at it's terminals.	
	4 .. 0	MaxPower	Maximum digital modem transmit power. This is represented in -0.5 dBm0 steps where 0 represents -0.5dBm0 and 31 represents -16 dBm0	

7C	MAXTXSPEED	In case of V.34 split speed operation this location determines the maximum speed of the v34 transmitter. It's value should remain within the boundaries of the speed rate capability mask (SPEED_SEL_H, SPEED_SEL_L). Maximum transmit bit rate = MAXTxSPEED*2400 Default = [14]	
7D	MAXTXSPEED_V90	In case of V.90 operation this location determines the maximum speed of the v90 transmitter. It's value should remain within the boundaries of the speed rate capability mask (SPEED_SELV90_H, SPEED_SELV90_L). Maximum transmit bit rate = 28000 + MAXTxSPEED*4000/3 Default = [21]	

7E	MAXRXSPEED	In case of V.34 split speed operation this location determines the maximum speed of the v34 receiver. It's value should remain within the boundaries of the speed rate capability mask (SPEED_SEL_H, SPEED_SEL_L). Maximum transmit bit rate = MAXTxSPEED*2400 Default = [14]	
7F	MAXRXSPEED_V90	In case of V.90 operation this location determines the maximum speed of the v90 receiver. It's value should remain within the boundaries of the speed rate capability mask (SPEED_SELV90_H, SPEED_SELV90_L). Maximum transmit bit rate = 28000 + MAXTxSPEED*4000/3 Default = [21]	

5.3.2. data-pump read database

important remark :

1. All parameters expressed in dBm require a calibration of the analogue front end hardware.
2. The host should only read these registers. The read database is mapped into the address offset range 128 until 255.

80	DatagramRate		location specifying the frequency at which the Datagrams of the synchronous data-pump interface are transferred.	
	F .. 8	DatagramrateRx	0:2400 1:600 2:1333 (=8000/6)	
	7 .. 0	DatagramrateTx	0:2400 1:600 2:1333 (=8000/6)	

81	DATASTATEspeedTx			
	F .. *6*	*spare*		
	5	TxSpeedselFormat	if 1, use V90 speed range format (see locations **speed_selV90_h, speed_selV90_l**) if 0, use V34 speed range format (see locations **speed_sel_h,speed_sel_l**)	
	4 ..0	TxSpeednumber	This field indicates the speed selected in DATASTATE for the modem transmitter. It contains a bit index number ranging from 0 to 31. This bit index points to the bit position in the 32 bit longword **speed_sel_h,speed_sel_l** or **speed_selV90_h, speed_selV90_l** depending on the **Txspeed_sel_format** bit. Every bit is this longword corresponds with a certain speed. For interpretation of the bit either the V90 or V34 speed range format has to be used. Remark:In case of symmetrical operation the user should use the speednumber field in the location datastatespeed to determine the transmitter speed.	

82	DATASTATESpeed			
	F .. *E*	*spare*		
	D	**SpeedselFormat**	if 1, use V90 speed range format (see locations **speed_selV90_h, speed_selV90_l**) if 0, use V34 speed range format (see locations **speed_sel_h,speed_sel_l**)	
	C	trellis bit	indicates if trellis bit b8 is activated in the final rateword (Only for V32bis)	
	B	sec channel	if 1,then the secondary channel is active in DATASTATE	

	A	dll	if 1, then down line loading is active in DATASTATE	
	9 ..5	normnumber	number indicating the bitposition in norm_l, norm_h of the modulation selected in DATASTATE	
	4 ..0	speednumber	This field indicates the speed selected in DATASTATE for the modem receiver. It contains a bit index number ranging from 0 to 31. This bit index points to the bit position in the 32 bit longword **speed_sel_h,speed_sel_l** or **speed_selV90_h, speed_selV90_l** depending on the **speed_sel_format** bit. Every bit is this longword corresponds with a certain speed. For interpretation of the bit either the V90 or V34 speed range format has to be used. In case of symmetrical operation this location also is valid for the transmitter.	

83 84	V33MultiPlToHost ErrorMessage	V33 Received Multiplexer Configuration (numbering according CCITT V33, Table 5A & 5B) ErrorMessage of Tfast	

85	*Symbolrate*		
	8	*8000/6*	
	7	*1600*	
	6	*1200*	
	5	*3429 (=3000 * 8/7)*	
	4	*3200*	
	3	*3000*	
	2	*2800*	
	1	*2743 (=2400 * 8/7)*	
	0	*2400*	

86	*Samplerate*		
	8	*8000*	
	7	*14400*	
	6	*12000*	
	5	*10287 (=9000 * 8/7)*	
	4	*9600*	
	3	*9000*	
	2	*8400*	
	1	*8229 (=7200 * 8/7)*	
	0	*7200*	

87	*Samplebuffersize*	*The ratio of sample- and symbolrate=number of samples exchanged per symbolclockperiod.*	

88	ToneLevelA	output tone detection circuit A in dB. *scale and units =1 dBm *number type = Unsigned integer *range = +6 dBm (3F) -> -57 dBm 0(00) *accuracy = ± 0.5 dBm from +6 to -35 dBm, ± 1dBm from -35 to -45 dbm, otherwise, measurement result will always remain smaller then -45 dBm	

89	ToneLevelB	output tone detection circuit B in dB. *scale and units =1 dBm *number type = Unsigned integer *range =+6 dBm (3F) -> -57 dBm 0(00)	

		*accuracy = ± 0.5 dBm from +6 to -35 dBm, ± 1dBm from -35 to -45 dbm, otherwise, measurement result will always remain smaller then -45 dBm	

8A	VersionA_L	contains XXS value for processor A described in document RD/C07 DSPware revision control, S located at the lowest nibble (Currently not supported on Aster4 Flash)	
8B	VersionB_L	contains XXS value for processor B described in document RD/C07 DSPware revision control, S located at the lowest nibble (Currently not Supported on Aster4 Flash)	
8C	HW_Norm_H	Modulations not supported, based on Hardware limitations. The definition of HW_norm_H is identical to Norm_H of the Write Database. If the corresponding bit is put to one, the modulation is not supported. Default=[FFFE]	
8D	HW_Norm_L	Modulations not supported, based on Hardware limitations. The definition of HW_norm_L is identical to Norm_L of the Write Database. If the corresponding bit is put to one, the modulation is not supported. Default=[8000]	

8E	ZerCrosCntr	*definition =No. of zero crossings. The signal is taken before the receive filter. Detection can be done for tones between 50 Hz and 4800 Hz.The counter is a continuous 16 bit wrap around counter, if ZerCrosPer is equal to 0. Otherwise, this register will be cleared periodically as indicated by ZerCrosPer.	

8F	Eventbufferlength	location containing the length of the eventbuffer default = [0008]	
90	Eventstructptr	location containing the first address of the event structure	
91	Datastructptr	location containing the first address of the parallel synchronous data interface	
92	Unitimer	real time clock which ticks at 1 kHz frequency, can be used as a **watchdog**	
93	UnitimerLSW		
94	Unitimerdelta		
95	LLFbasePtr	location contain the base address of the 'low level functions' database of Channel 00 .	
96	DCESCCstructptr	location containing the first address of the SCC1 data interface	
97	LLFbaseLength	location containing the length of the 'low level functions' database. The LLFbaseptr combined with the LLFbaseLength makes it possible to calculated the base address of the LLF database of all 32 Channels, since they are placed in memory sequentially.	

98	RXLevel	*definition=10 LOG (AVERAGE POWER OF THE FAREND SIGNAL referencing to a 600 ohm load / 1 Mw). The receive signal is measured after the receive filter, so only the inband power will be measured, with an exception for the dial mode where the power is measured before the receive filter.This parameter is measured during training and DATASTATE. *scale and units =1 dBm *number type = Unsigned integer *range =+6 dBm (3F) -> -57 dBm (00) *accuracy =± 0.5 dBm from +6 to -35 dBm, ± 1dBm from -35 to -45 dbm, otherwise, measurement result will always remain smaller then -45 dBm	

99	EcLevel	*definition =10 LOG (AVERAGE POWER OF THE ECHO SIGNAL referencing to a 600 ohm load / 1 Mw).The echo signal is measured after the receive filter, so only the inband power will be measured. This parameter is measured during training and DATASTATE. *scale and units = 1 dBm *number type = Unsigned integer *range =+6 dBm (3F) -> -57 dBm (00) *accuracy = ± 0.5 dBm from +6 to -35 dBm, ± 1dBm from -35 to -45 dbm, otherwise, measurement result will always remain smaller then -45 dBm	

9A 9B 9C	NearEcLevel FarEcLevel FarEchoPhaseRoll	*definition = cfr Echolevel *definition = cfr Echolevel *definition = cfr Frequency offset (Only measured in V.32bis)	

9D	SNRatio	Signal To Noise Ratio *definition = 10 LOG (AVERAGE SIGNAL POWER/AVERAGE SQUARED ERROR).This measurement is performed on the received phasepoint diagram. *scale and units = 0.5 dB *number type = Unsigned integer *range =8 dB(00) -> 40 dB (3F) *accuracy = ± 0.5 dB	

9E	FreqOffset	Frequency Offset *definition =difference between the carrier frequency of the received signal and the nominal carrier frequency *scale and units = 0.4 Hz *number type = 2's complement integer *range =-12.8 Hz (E0) -> 12.4 Hz (1F) *accuracy = ± 0.2 Hz over the entire range	

9F	TimOffset	Timing Offset *definition = difference between the frequency of the receive baud clock and the local baud clock *scale and units = 0.001 % *number type = 2's complement integer *range =-0.032 % (E0) -> 0.031 % (1F) *accuracy = ± 0.001 %	

A0	AM_MOD	AM_Modulation *definition =An average gain is calculated using a lowpass filter with a time constant of 1 sec. Min and max values of the average gain are calculated for a period of 10 seconds. The percentage of low frequency AM modulation is =((Max-Min)/(Max+Min))*100 (Only measured in V.32bis) *scale and units = 1 % *number type = Unsigned integer *range =0 % (00) -> 63 % (3F) *accuracy = ± 1% from 0 to 0.5 Hz, ± 2% from 0.5 to 2 Hz	

A1	PeakGainErr	PeakGainErr (Gain Hits, only V.32) *definition =An average gain is calculated using a lowpass filter with a time constant of 1 sec. The actual gain will be calculated using a filter with a time constant which is 10 times faster. The following will be calculated for a 10 sec period= MAX(|Actual gain - Average gain| / Average gain)*100 *scale and units = 2 % *number type = Unsigned integer *range =0 % (00) -> 124 % (3F) *accuracy = ± 0.20 * readout from 0 to 60%, ± 0.25 ,* readout from 60 to 100%, 0.25 * readout from 100 to 126%	

A2	PhaseJit	Single frequency phase jitter *definition =Average peak to peak value of the single frequency phase jitter *scale and units = 1 degrees *number type = Unsigned integer *range =0 (00) -> 63 degrees (3F) *accuracy Jitter < 3 degr result = zero, 4 < jitter < 39 degr ± 1 degr	

A3	PeakPhasErr	Peak Phase Error *definition=Maximum amplitude of the phase error measured over a period of 10 seconds. *scale and units = 0.5 degrees *number type = Unsigned integer *range =0 (00) -> 31.5 degrees (3F) *accuracy = ± 2 degrees from 0 to 17 degrees,2 degrees above 17 degrees	

A4	INR	Impulse Noise Ratio {only V.32} *definition =10 LOG (AVERAGE SIGNAL POWER / MAXIMUM ERROR) This formula will be evaluated on the received phasepoint diagram during intervals of 10 sec *scale and units = 0.5 dB *number type = Unsigned integer *range =0 (00) -> 31 dB (3F) *accuracy = ± 0.5 dB	

A5	SNRPROB	SNR at receiver slicer, as projected by the line probing phase of V.34	
A6	Signalquality	MAE (mean absolute error) calculated at the receiver slicer	

A7	RTDelay	Round Trip Delay *definition =MT and NT values as defined in V.32 *scale and units = 10 msec *number type = Unsigned integer *range =0 (00) -> 2550 msec (7F) *accuracy =5 msec	

A8 … *B7*	*reserved* *reserved*		

B8 .. *BB*	*spare* *spare*		

BC	EYESAMPLE_0	At symbol rate the modem core writes a (X,Y) sample for the eye-pattern in this buffer. High byte : Y coordinate of the eyepattern;Low byte: X coordinate of the eyepattern	
BD	EYESAMPLE_1	At symbol rate the modem core writes a (X,Y) sample for the eye-pattern in this buffer. High byte : Y coordinate of the eyepattern;Low byte: X coordinate of the eyepattern	
BE	EYESAMPLE_2	At symbol rate the modem core writes a (X,Y) sample for the eye-pattern in this buffer. High byte : Y coordinate of the eyepattern;Low byte: X coordinate of the eyepattern	
BF	*Gen_Control*		

C0	*RXD*	*RXD is a 16 bit location that contains 16 data bits b15..b0 for the main channel receive data, b15 is the oldest bit i.e the first bit sent from DSP to host, b0 is the last bit sent. This implies that the data bits are left aligned on position b15.*	

C1	*changeBITS*			
	F	*ch_rstatus_ch_dbs*	*Set to one during one Host-Kernel RX_2400 communication cycle.If one, the value of rstatus_ch_dbs has changed.*	
	E	*ch_rstatus_dbs*	*Set to one during one Host-Kernel RX_2400 communication cycle.If one, the value of rstatus_dbs has changed.*	
	D	*ch_trnprogress_dbs*	*Set to one during one Host-Kernel RX_2400 communication cycle.If one, the value of trnprogress_dbs has changed.*	
	C .. *4*	*spare*		
	3	*Rxdog[3]*	*return value of the watchdog system*	

	2	*Rxdog[2]*	*return value of the watchdog system*	
	1	*Rxdog[1]*	*return value of the watchdog system*	
	0	*Rxdog[0]*	*return value of the watchdog system*	

C2	*S2PD*	*S2PD is a 16 bit location that contains output data from the serial-to-parallel conversion module, b15 is the oldest bit i.e the first bit sent from DSP to host, b0 is the last bit sent. This implies that the data bits are left aligned on position b15. (only for ASTER4 FLASH).*	
C3	RSTATUS_CH_dbs	definition identical to Rstatus_ch This location is a variable of the interrupt driven interface between dsp and host, It's value is updated from the internal variable rstatus_ch just before the event to the host is generated and remains untouched	
C4	RSTATUS_dbs	definition identical to Rstatus This location is a variable of the interrupt driven interface between dsp and host, It's value is updated from the internal variable rstatus just before the event to the host is generated and remains untouched	
C5	TRNPROGRESS_dbs	definition identical to TrnProgess This location is a variable of the interrupt driven interface between dsp and host, It's value is updated from the internal variable trnprogress just before the event to the host is generated and remains untouched	
C6	PF_setup	This location contains the value of the programmable flags after reset.	
C7	*TXSAMPLE_0*	*At symbol rate the modem core writes 3,4,5 or 6 samples to transmit in this buffer. The number of samples equals the ratio of sample- and symbolrate. This ratio can be found in the read database location 'Samplebuffersize'.*	
C8	*TXSAMPLE_1*		
C9	*TXSAMPLE_2*		
CA	*TXSAMPLE_3*		
CB	*TXSAMPLE_4*		
CC	*TXSAMPLE_5*		

CD	*DI_control*			
	F	*TX request bit*	*if 1, the modem core asks the kernel to give a new 2400 Hz baud data packet (in the TXD0,TXD1,TXD2-location); this bit is cleared by the DSP after arrival of the packet.*	
	E	*RX valid bit1*	*if 1, the data in RXD1 is valid.*	
	D	*RX valid bit0*	*if 1, the data in RXD0 is valid.*	
	C	*SPARE*		
	B	*CODECclocking*		
	A	*slavebit*		
	9	*syncbit*		

	8 *..* *0*	*spare*		

CE	*RXD0*	*RXD0 is a 16 bit location that contains 16 data bits b15..b0 for the main channel receive data, b15 is the oldest bit i.e the first bit sent from DSP to host, b0 is the last bit sent. This implies that the data bits are left aligned on position b15.* *If RX_valid_bit0 of DI_control is 1, the data in RXD0 is valid*	
CF	*RXD1*	*RXD1 is a 16 bit location that contains 16 data bits b15..b0 for the main channel receive data, b15 is the oldest bit i.e the first bit sent from DSP to host, b0 is the last bit sent. This implies that the data bits are left aligned on position b15.* *If RX_valid_bit1 of DI_control is 1, the data in RXD1 is valid*	
D0	bootpage_nr	parameter specifying the page number to be loaded 0:dial 1:V.22 2:V.32 3:fsk 4:fax 6:V.8 7:info 8:V.34 9: startup 10: protocol 11: at-set offline 12:at-set online 13:V90 A 14:V90 D 15: fax protocol 16:LLpage 17:V110 18:V18 19:speakerphone … 50:unload 51:powerdown interrupt	
D1	*spare*		

D2	*Init8kRoutine*		
D3	*Core8kRoutine*		
D4	*ShellOutptr*		
D5	*GEN_CONTROL*		
D6	*CODECRXPllPhaseShift*		
D7	*uOffset*		
D8	*CoreRoutine*		
D9	*InitRoutine*		
DA	*RTVal*		

DB	*BaudInfo*		
DC	*Nearbulklength*	*parameter specifying the length of the Near Echo canceller delay line. This parameter gives the length in number of (X,Y) couples.*	
DD	*BulkLength*	*parameter specifying the circular length of the Far Echo canceller Bulk delay line. This parameter gives the number of (X,Y) couples that have to be stored in the bulk.*	
DE	*BulkInputX*	*at Symbolrate the V.34 modemCore offers the kernel a X sample, which should be placed in the Bulk delay line by the Kernel.*	
DF	*BulkInputY*	*at Symbolrate the V.34 modemCore offers the kernel a Y sample, which should be placed in the Bulk delay line by the Kernel.*	

E0	*Rstatus_ch*		
	F *change_h*	*set to one if b8..b14 has changed*	
	E *spare*		
	D *SPEEDTX*	*if 1, then the transmitter speed speed is available in the DATASTATETX read database location*	
	C *Ratechange*	*if 1, then the data-pump is doing a rate -change*	
	B *SPEED*	*if 1, then the receiver speed is available in the DATASTATE read database location*	
	A *CTS*	*if 1, then the data channel CTS is active*	
	9 *DSR*	*if 1, then data channel DSR is active*	
	8 *DCD*	*if 1, then data channel DCD is active*	
	7 *change_l*	*set to one if b0..b6 has changed*	
	6 *spare*		
	5 *spare*		
	4 *SQALARM*	*IF 1, then the MSE measured during DATASTATE, crosses the training threshold value of the selected DATASTATE speed*	
	3 *DIAL*	*If 1,then the DSP digit register contains a value, which still has to be transmitted. If 0, then the digit register is empty*	
	2 *SECTXRq*	*if 1,then the host is asked to offer a new secondary transmit data bit*	
	1 *SECRXPR*	*if 1, then a secondary receive data bit is present*	
	0 *SECRXD*	*secondary receive data bit*	

E1	*Rstatus*		
	F .. *E* *spare*		
	D *CI*	*if 1, CI signal is detected in V.8*	
	C *Online*	*if 1, the data-pump is in on line state*	

	B	*ringvalid*	*if 1, the auto dialler has detected a valid ring tone*	
	A	*corebit*	*if 1, the modulation core has been executed*	
	9	*autodialfinished*	*if 1, the auto dialler has terminated the dialling procedure*	
	8	*bootrequestbit*	*if 1, the modem core asks to boot the next page, indicated in the register bootpage_nr*	
	7	*Flowcntrl*	*if 1, then use of the communication pipe_line is prohibited. If 0, then the communication pipe_line is available to the user*	
	6	*Boot*	*if 1, then the DSP starts booting from a boot page to another. During this boot action the communication pipe_line can not be used*	
	5	*TST*	*if 1, then the data channel is in a test condition. (only used in V.22 or V.22bis mode)*	
	4	*DL*	*if 1, then a request for a V.54 loop 2 has been received. (only used in V.22 or V.22bis mode)*	
	3	*RING*	*if 1, then a valid ring signal is present*	
	2	*Frq*	*Remote DCE Powerfail Alarm (proprietary). This is a supervisory tonedetection ciruit consisting of a bandpass filter centered around 1300 Hz with a very low threshold level. When a pure 1300 Hz tone is recognised, the bit is set to one.*	
	1	*ENDT*	*if 1, then carrier energy above the selected on-threshold is detected. if 0, then carrier energy below the selected off-threshold is detected*	
	0	*ZerCr*	*The ZerCr bit will be set if the number of zerocrossing during a period defined by ZerCrosPer is between the limits defined by ZerCrosMin and ZerCrosMax and the received signal level is bigger than the value specified in ZerCrosEner. The default values can be used for 2100 Hz answertone detection.*	

E2	*TrnProgress*	*Training progress as defined on the training drawings.* *For V33 this register has to be looked to as 2 nibbles. The highest nibble contains the training progress of the transmitter, the lowest nibble the training progress of the receiver.*	

E3 *...* *FF*	*reserved* *reserved*		

5.4. IDMA DATA-PUMP SCRIPTS

All scripts are presented in a table format. However, the user can translate these to scripts that run on its host processor and do the register initialisation through IDMA. Database starting address through IDMA is H#7EE0.

5.4.1. IDMA control sequences

1. After the dial page is loaded in memory, the data-pump is **setup** by programming the appropriate data memory locations through IDMA with following sequence.

GEN_SETUP0	0x00C4	extended training in V.32bis and V.34, pstn mode enabled, normal equalizer
GEN_SETUP1	0x0040	
GEN_SETUP2	0x0000	
INFO0_SETUP	0Xf0FD	
TD	0x0006	
TA	0x0006	
TX_LEVEL_TUNE	0x00FF	
DCD_OFF	0x0030	
DCD_HYST	0x0000	
MINTIMER	0x0003	
MAXTIMER	0x0003	
WSTATUS	0x2000	{bit h#D has to be set, for the settings to become activated}

Table 12 : data-pump initialisation sequence

Remark: the above initializations should only be done the first time the dial page is loaded after reset or power up.

2. Selection of the recommendation and the speeds:

NORM_H NORM_L	0x0001 0x0100	V.34 and V8 selected
SPEED_SEL_H SPEED_SEL_L	0x001F 0xFF00	{select all speeds from 33600 to 2400}

Table 13 : modulation and speed selection

3. Following scripts have to be run when the PSTN connection already has been established (by means of manual dialling, a data-pump dial script or any other way). The training scripts should be executed when the dial page is active. After the script is executed, then the dial page requests the host to boot the V.8 page.

Gen_setup1	0x048c	calling mode, 2wire, internal clock, norm operation mode
GEN_setup2	0x0030	disable tone detection, enable signal quality step up and fallback
Wstatus	0x2000	bit h#D has to be set, for the settings to become activated

Table 14: calling mode training

Gen_setup1	0x0484	answer mode, 2wire, internal clock, norm operation mode
GEN_setup2	0x0030	disable tone detection, enable signal quality step up and fallback
Wstatus	0x2000	bit h#D has to be set, for the settings to become activated

Table 15: answer mode training

Gen_setup1	0x068C	analog loop, 2wire, internal clock, norm operation
Wstatus	0x2000	bit h#D has to be set, for the settings to become activated

Table 16: analog loop test mode training

4.To initiate a cleardown sequence, the host has to start a call or answer **retrain** or **ratechange** with 'cleardown' selected in speed_sel_l, while the data-pump is running in 'datastate'

Speed_sel_h	0x0000	disable all speeds from 33600 to 24000
Speed_sel_l	0x0001	only enable the cleardown bit
GEN_setup2	0x0001	set retrain bit
Wstatus	0x2000	bit h#D has to be set, for the settings to become activated

Table 17: cleardown mode training

5. When the modem is training, in DATASTATE or in the cleardown state this script will ask to start a boot operation to the dial page.

Gen_setup1	0x0424	idle operation mode
Wstatus	0x2000	bit h#D has to be set, for the settings to become activated

Table 18: idle mode sequence

5.4.2. data-pump supervisory rules

below are listed some definitions:

- DATASTATE : when trnprogress_dbs = d0
- cleardown : when trnprogress_dbs = e0 of trnprogress_dbs = ea

Supervisor and line-follow up are two modules running on the host processor, not inside the datapump.

Basically the supervisor can be in 5 different states: INIT, TRAIN, SUPDATA, RETRAIN and DISC.
The host system but also the line follow-up module can cause changes of the supervisor state. For every state of the supervisor the line follow up module executes a different flow diagram.

The line follow up starts in the INIT state. This is when the datapump training hasn't started up yet. In this state nothing is done. When the training starts (incoming call or host decides to build a connection) and more specifically at the moment the request to load the V8 page is generated by the data-pump, some initialisations have to be done and the AbortTimer (40s) and TrainingTimer (15s) are installed.

The supervisor goes to the next state TRAIN, indicating that the modem is in an initial training, and the line follow-up executes the TRAIN flow diagram.

In TRAIN the line follow-up first checks if there is a timeout of the AbortTimer. On timeout of the AbortTimer, it brings the supervisor to DISC state. If there is no timeout, the line follow up first checks if TrnProgress equals D0.

If so, the line follow up brings the supervisor in the next supervisor state SUPDATA and the DCDOffTimer is installed.

If not, the line follow up changes the supervisor state to DISC if TrnProgress equals E0 (Cleardown) or EA (V32INCOMPAT).

If the line follow up stays too long in a specific training state, it gets a timeout of the TrainingTimer. Then, if the number of (re)trains it has counted, indicated by retraincounter, exceeds 3, it also goes to DISC. Otherwise, a new (re)train is initiated and the retraincounter is incremented.

If no cleardown and no timeout of the TrainingTimer, it checks whether there is a timeout of the RetrainResponsTimer (Is there a remote modem?). This test can only have effect in the beginning of the (re)training. The ResponseTimFlg indicates that the training progress is before or after 0x3A (for V34) or 0x50 (for other modulations). On timeout during the initial phase of the retraining, the line follow up has to bring the supervisor to DISC.
Then, it has to see if there is progress in the training. If there is progress, the TrainingTimer is installed again and LastStatus, which stores the latest value of TrnProgress, is updated. If there is no progress, nothing is done. But if there is no progress for too much time, the line follow=up gets a timeout of the trainingtimer, as explained before.
It is also possible that, because of some recovery mechanisms in the training, the TrnProgress is smaller than LastStatus. In that case, the line follow up checks the RetrainAutofallbackcount. If this count exceeds 10, meaning that we had a recovery situation for more than 10 times during the last (re)train, it starts a fresh retrain. Otherwise, the TrainingTimer and RetrainResponsTimer are installed again, LastStatus is reset to 0 and ResponseTimFlg is made 1 again, since it is possible the training fell back to a state before 0x3A (for V34) or 0x50 (for other modulations).

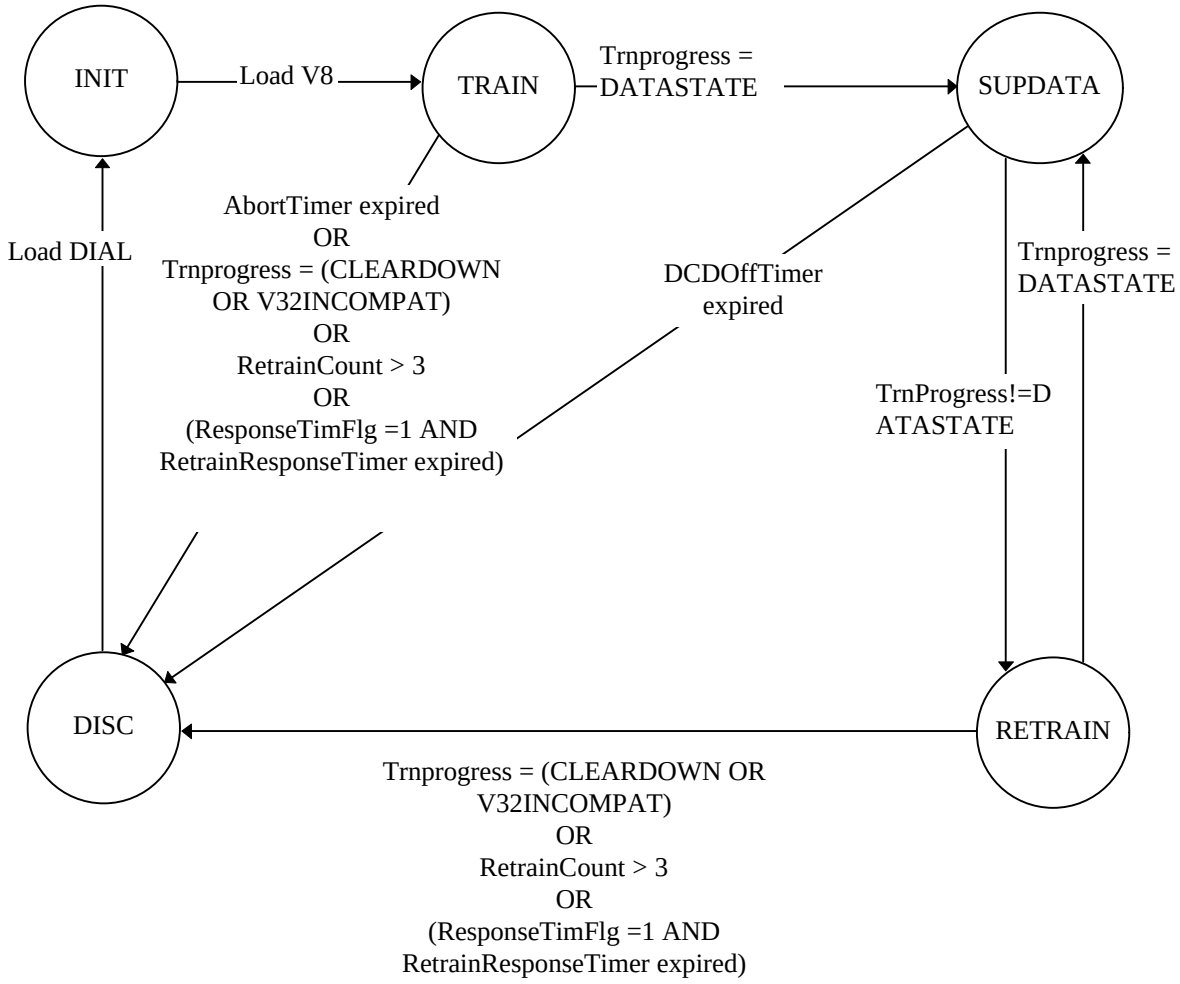
In SUPDATA, the line follow up checks if TrnProgress still equals D0.
If not, the datapump is in a retrain or ratechange and the supervisor state variable is switched to RETRAIN. Some timers are installed (TrainingTimer to check whether we don't stay too long in a training state and the RetrainResponsTimer checking if we get a response from the remote modem).
If the datapump still is in D0, the DCDbit of rstatus_ch and the ENDTbit of rstatus are checked. If there is no DCD during a certain amount of time(DCDOffTimer, which is country dependent), the state changes to DISC. If there still is DCD, the DCDOffTimer is installed again.
In SUPDATA, the retraincounter is decremented every 15 seconds until it is 0.

RETRAIN is the supervisor state when the modem is doing a retrain or a ratechange. So, this state can only be reached after SUPDATA. In this state the line follow up has to do the same things as in TRAIN, except for the checking of the AbortTimer.

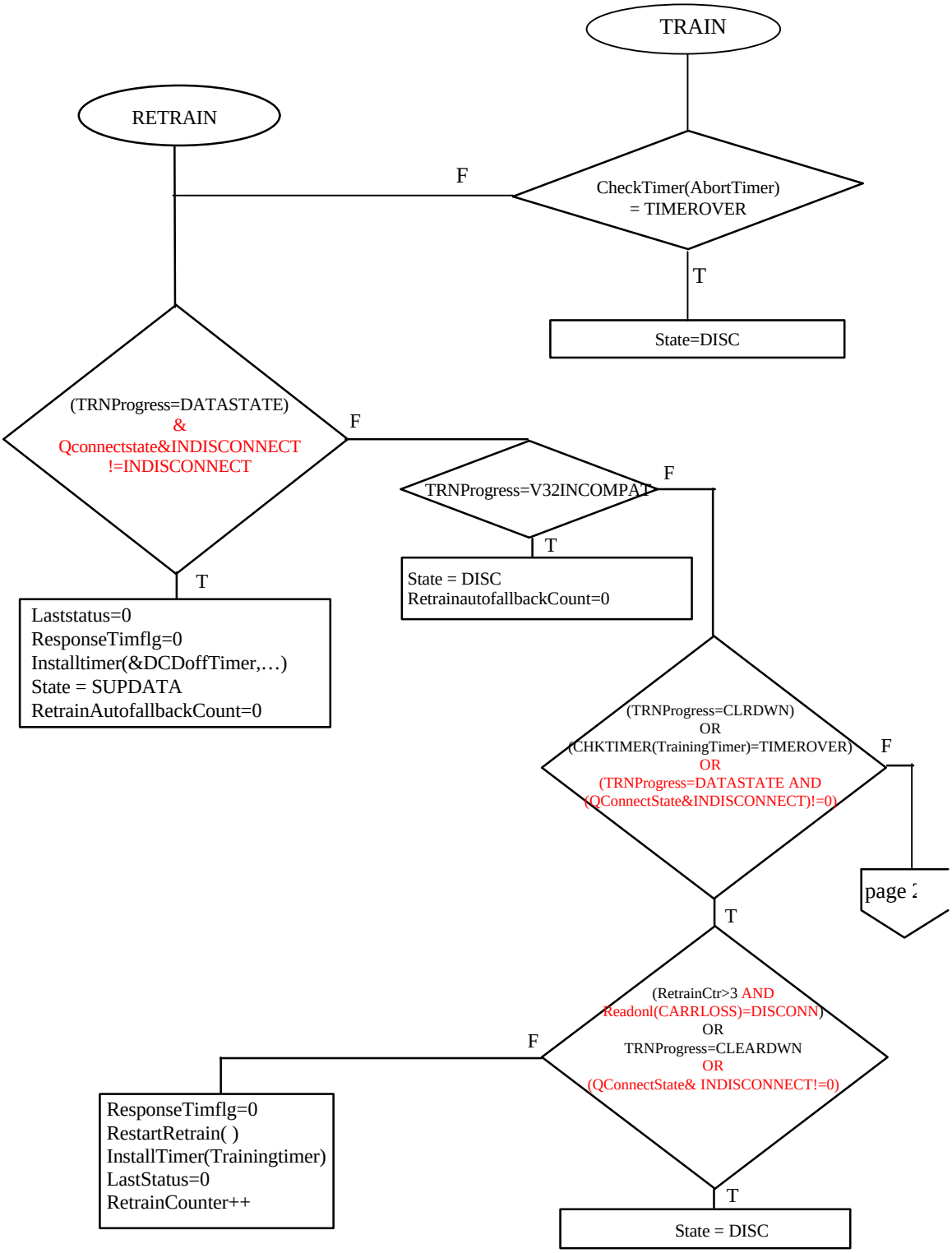
In DISC, the modem has to go onhook. This is done by starting the idle mode sequence. When a request to load the dial page is seen, the line follow up brings the supervisor back to INIT.

5.4.2.1. data-pump supervisor state diagram

Figure 28: supervisor state diagram



5.4.2.2. data-pump line-follow-up flow diagram



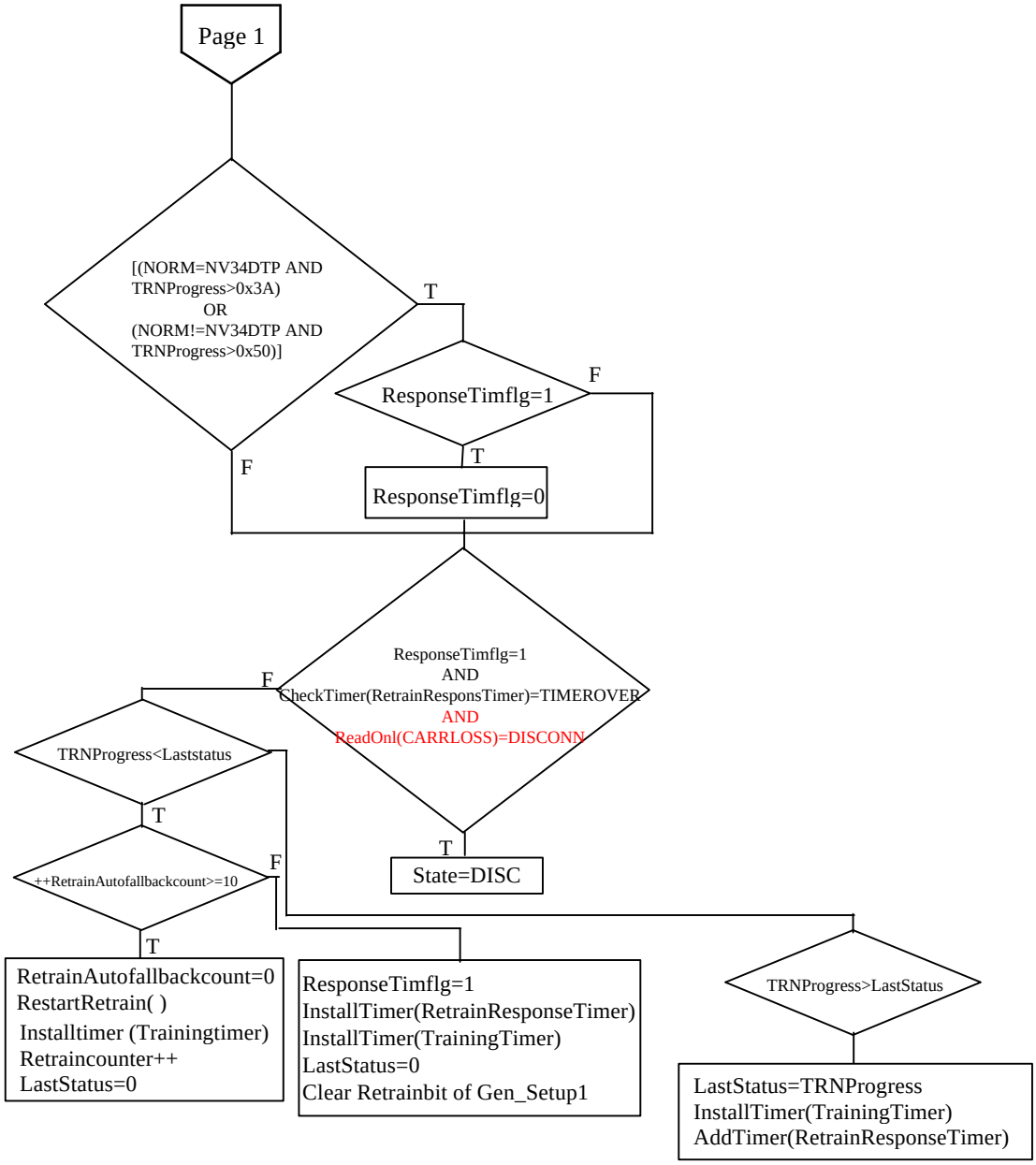


Figure 29: line follow up flow diagram for train and retrain state

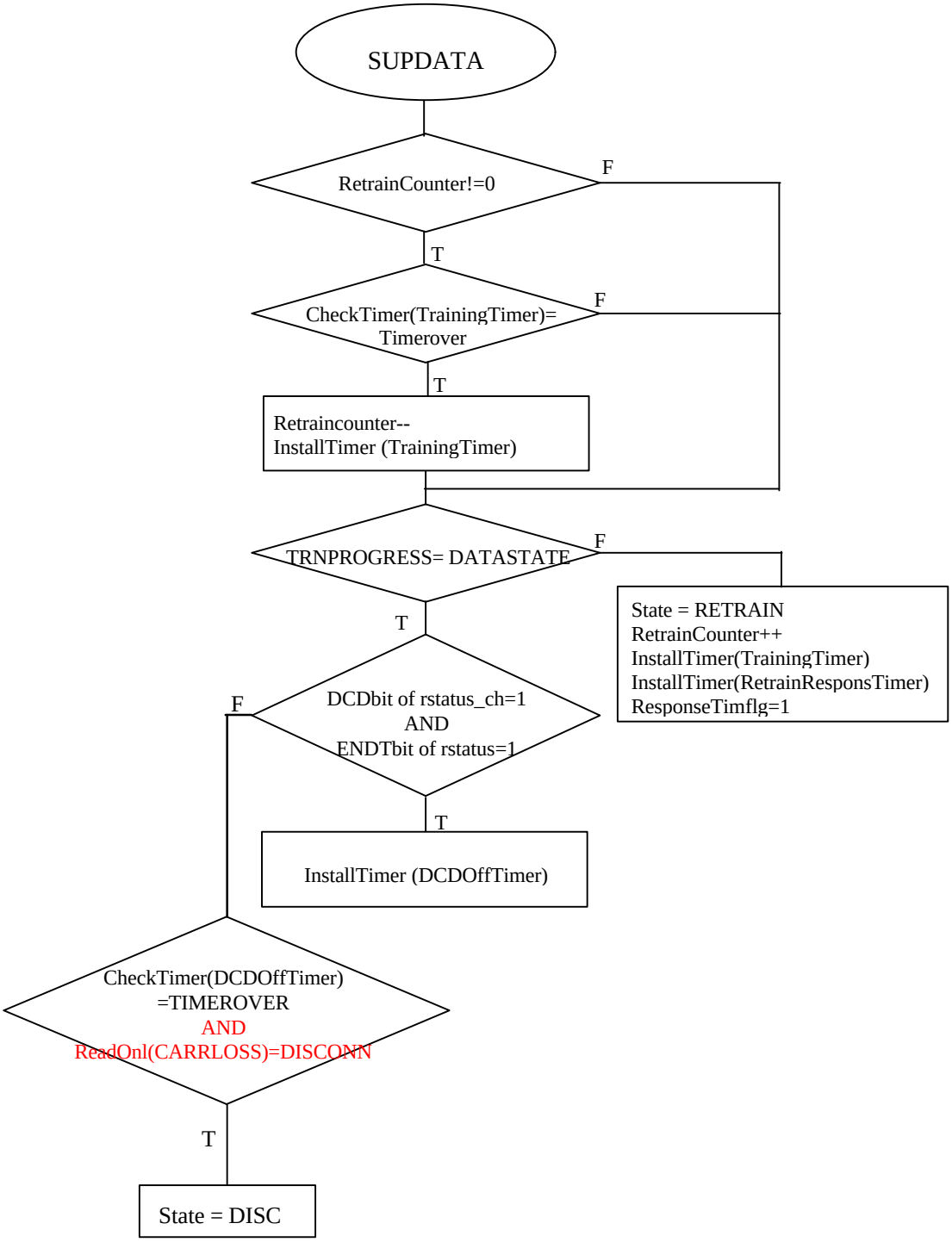


Figure 30: line follow up flow diagram for supdata

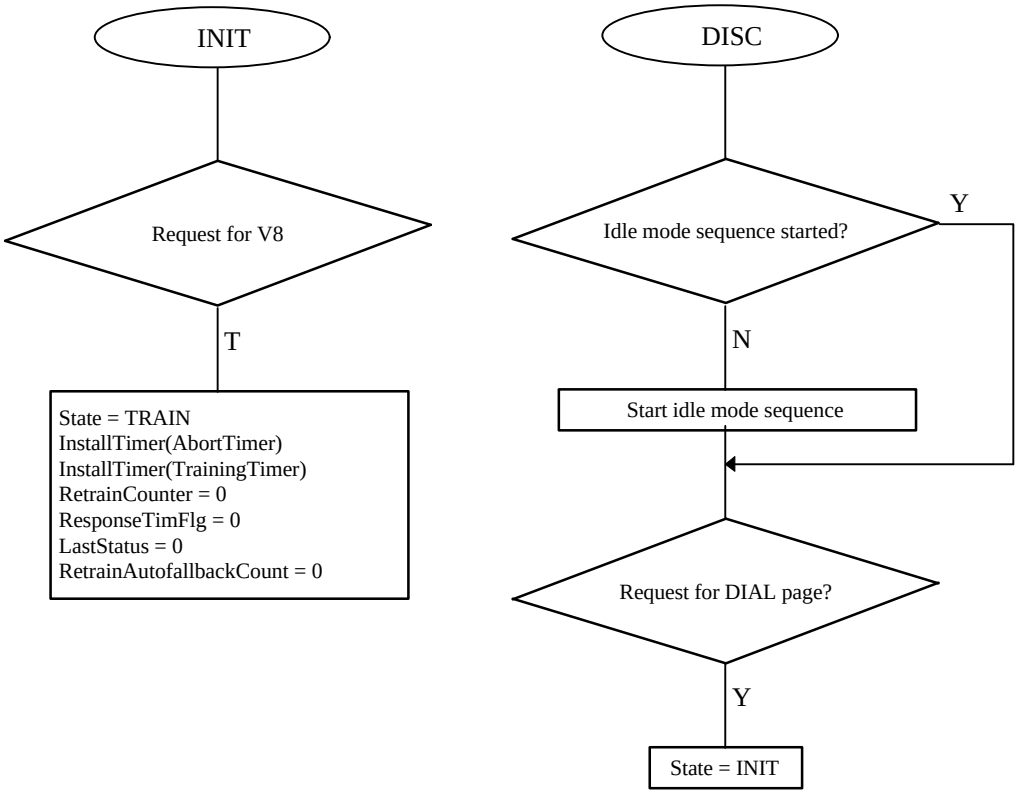


Figure 31: line follow up flow diagram for INIT and DISC states

5.5. DATA-PUMP EVENT CODES

note : new events can be added to this list during product development, also code assignment may change.

EVENT	CODE	
new value of Rstatus_ch	H#0001	
new value of Rstatus	H#0002	
new value of TRNProgress	H#0003	
RX BUFFER0 FULL	H#0020	
RX BUFFER1 FULL	H#0021	
TX BUFFER0 EMPTY	H#0028	
TX BUFFER1 EMPTY	H#0029	
DCESCCRX FULL buffer0	H#0040	For the 8 RX buffers of the SCC1
DCESCCRX FULL buffer1	H#0041	an event will be generated when
DCESCCRX FULL buffer2	H#0042	newly received data from lineside
DCESCCRX FULL buffer3	H#0043	has filled these buffers. This is the
DCESCCRX FULL buffer4	H#0044	signal for the host to read these
DCESCCRX FULL buffer5	H#0045	buffers.
DCESCCRX FULL buffer6	H#0046	
DCESCCRX FULL buffer7	H#0047	
DCESCCTX EMPTY buffer0	H#0050	For the 8 TX buffers of the SCC1
DCESCCTX EMPTY buffer1	H#0051	an event will be generated when
DCESCCTX EMPTY buffer2	H#0052	the data in these buffers is transmit-
DCESCCTX EMPTY buffer3	H#0053	ted over the line. This is the signal
DCESCCTX EMPTY buffer4	H#0054	for the host to refill these buffers.
DCESCCTX EMPTY buffer5	H#0055	
DCESCCTX EMPTY buffer6	H#0056	
DCESCCTX EMPTY buffer7	H#0057	

For the ‘new value’ events, the actual new value is available in the corresponding ‘_dbs’ location, i.e. rstatus_ch_dbs, rstatus_dbs, trnprogress_dbs of the data-pump read database. These values are updates from the internal variables rstatus_ch, rstatus and trnprogress just before the event is generated and remain untouched.

6. Appendix

6.1. FILE FORMAT FOR IDMA BOOTING

The binary file consists of two parts : a header table and a body table. Both parts follow directly after each other and are explained in following paragraphs.

6.1.1. header table

Offset (in 16 bit words)	description	
0	headerlength	number of bytes in the header, headerlength location included
1 2	no. of bytes (32 bit)	number of bytes in the binary file, header not included (i.e. the length in bytes of the body table)
3 4	checksum (32bit)	checksum of the binary file, header not included (i.e. checksum of the body table)
5	[15..12] : version number S [11..0] : version number Y,Y,Y	for Telindus internal use only
6	[15..8] : version number X,X [7..0] : time stamp year	for Telindus internal use only
7	[15..8] : time stamp month [7..0] : time stamp day	in BCD[34] format in BCD format
8	[15..8] : time stamp hour [7..0] : time stamp minute	in BCD format in BCD format
9	[15..0] : file format	bit 0 : if set to 1 then the IDMA overlay field is present in every pageblock bit 8 : 0 : standard format used for program code 1 : compact format used for program code
10	[15..0]:maximum pageblock length	length in bytes of the longest page block (without header)
11…18	configuration code of the software	for Telindus internal use only
19…26	RCS label of the software	for Telindus internal use only

6.1.2. body table

The body table consists of three parts : the intro part, the index part and the page part. The intro part gives some general information about the body table. The index part describes which pages are contained in the file and where (expressed as an offset from the beginning of the body table) they can be found in the file. The page part is composed of one or more pageblocks. Every pageblock begins with a header and is followed by the data of that block. The pageblocks follow directly after each other. A pageblock header length of zero indicates the end of that page. The meaning of the fields in each of the parts is explained in the table below.

[34] BCD : Binary Coded Decimal

part	offset (in 16 bit words)	description	
intro	0[35] 1	STARTUP page offset (32 bit)	indicates the position[36] of the STARTUP page
	2	index table length (16 bit)	number of longwords(32-bit) in the index table
index	3 4	offset page 0 (32 bit)	indicates the position of the page with index 0
	5 6	offset page1 (32 bit)	indicates the position of the page with index 1
	…	….	
	[37] …	….	
page	offset page0	pageblock header: length	number of 16 bit words in the page block
	+1	pageblock header:IDMA overlay	if bit 0 of the file format field in the header table is set to zero then this field is not present if bit 0 of the file format field in the header table is set to one then following definition is used : definition identical to control register 0x3fe7 of the 2187 • b15=1 • b14..b8 : reserved, set to zero • b7..b4 : id dmovlay • b3..b0: id pmovlay
	+2	pageblock header:IDMA control	definition identical to control register 0x3fe0 of the 2187/2187 • b15=0 • b14 indicates DM (if 1) or PM (if 0) • b13..b0 : destination address
	+3	page block word	[38]
		….	
		….	
		pageblock header length	
		pageblock header IDMA overlay	
		pageblock header IDMA control	
		….	
		….	
		H#0000	a 0 page block length indicates the end of this page
	offset page1	….	
		….	
		……..	
		H#0000….	
	…		

[35] We restart the numbering of the words from 0, but this table is directly following the header table.
[36] All positions are expressed in 16-bit words relative to position 0 of the body table. A 0(zero) means that the page is not contained in the file.
[37] If the startup page is contained in this file, then it is placed between the index table and page 0
[38] For program memory, the bytes can be inserted in two ways : standard or compact format. In the standard format every program instruction uses 4 bytes : a zero is inserted at byte 3. In the compact format, every program instruction takes 3 bytes. If needed one byte is inserted at the end of a pageblock to allign again with the 16-bit word offset used in this table.

6.2. FILE FORMAT FOR BDMA BOOTING

The format as described in the previous paragraph is also used for BDMA booting.

The resulting table (header + body table) can be placed anywhere in ROM but preferably immediately after the bootcode. The bootcode contains the same code as the startup page, but is generated with the spl21.exe tool of ADI with the loader option enabled. The bootcode must start at address 0 of the ROM. The bootcode is modified at word 31 (Msbyte=byte93, Lsbyte=byte94) : the NOP instruction is replaced by the 16 bit startaddress in ROM of the header table.
As mentioned in the previous paragraph, program memory can be encoded in a pageblock in two variants : a standard and a compact format. For BDMA booting always the compact format is used.

After reset the DSP will automatically load the bootcode from address 0 in ROM and start execution. During runtime the DSP loads another page when necessary.

6.3. EVENT INTERFACE REGISTERS

The structure below is used for the event interface between host and DSP.
The data-pump read database location eventstructureptr, location h#90, points to the first location of this structure.

	REGISTER	COMMENT
0	eventbuffer[0]	
1	eventbuffer[1]	
2	eventbuffer[2]	
3	eventbuffer[3]	
4	eventbuffer[4]	
5	eventbuffer[5]	
6	eventbuffer[6]	
7	eventbuffer[7]	
8	eventcounter	

6.4. MODEM INTERFACE LOCATIONS

The tables below list the memory mapped data/control interface locations. Functionally the interface has following parts,

- DTE sideSCC
- Host Side SCC
- AT modem database.
- IDMA modem database
- Async HDLC database
- event mechanism

All addresses are relative to a specific base address, This base address still may changed during product development.

- DTE side SCC

The base address of the dteSCCstructure can be found in the location DTESCCstructPtr of the modem IDMA database. A complete description can be found in the section 4.2.3.1 'SCC structure'.

- Host Side SCC

the base address of the hostSCCstructure can be found in the location HostSCCstructPtr of the modem IDMA database. A complete description can be found in the section 4.2.4.1 'The SCC structure'.

- AT %S registers database

The variable **ATdbaseAddress** from the **modem IDMA database** points to the first location of this database. These locations also can be read through at_set. For a complete description of these locations, see Chapter 4.3.1.4, '%S Registers'.

- IDMA modem database

Following locations containing information on the status of the protocols, modem and the connection. They are placed starting from DM H#3E80

- Async HDLC database

The Baseaddress of the Async database can be found in the location **ACCM map address** of the modem read database.

6.5. DATA-PUMP INTERFACE LOCATIONS

- data-pump write database

The data-pump write database interface consists out of 128 consecutive locations in the DSP's internal data memory. The address of the first location is h#3EE0.
Chapter 'data-pump write database', 5.3.1 gives an explanation of all locations.

- data-pump read database

The data-pump write database interface consists out of 128 consecutive locations in the DSP's internal data memory. The absolute address of the first location is h#3F60.
Chapter 'data-pump read database', 5.3.2 gives an explanation of all locations.

- parallel synchronous data interface

The data-pump read database location datastructptr, offset h#91, points to the first location of this buffer structure, used to transmit and receive data.
Chapter 'parallel synchronous interface', 5.2.1 gives an explanation of all locations.

- Line Side SCC

the base address of the dceSCCstructure can be found in the location DCESCCstructPtr of the data-pump IDMA database
A complete description can be found in the section 'Synchronous HDLC SCC', 5.2.2, and 'Uart SCC', 5.2.3.

6.6. EYE PATTERN DISPLAY

The oscilloscoop eye pattern is a very easy way to determine the correct functioning of the modem software. For each of the pages an eyepattern exists, that will display in real time a number of modem parameters. With each run of the main program (i.e. at symbolrate) a new value is generated in the eyepattern location of the read database (which can be eyesample_0, eyesample_1 or eyesample_2 depending of the page). To generate samples for the eye monitor, you must divide eyesample_0 and eyesample_1 in their high and low byte. The high byte is the X-value, the low byte the Y-value. This value is located in the read database at :

- eyesample_0 for the V.8 eyepattern (read database location 3C)
- eyesample_2 for the Info eyepattern (read database location 3E)
- In V.34, the database locations eyesample_0 and eyesample_1 are used. Both locations contain a combination of X and Y values, so the rate for the eye monitor is twice the symbol rate. When disp_setup = 0, the standard eye monitor is generated, which shows the equaliser and the phase points for the receiver, together with a status line on top. The standard status line shows the symbol rate, the carrier for the transmitter and for the receiver (H or L), the bit rate (in DATASTATE), and a debug location.

In a future version of this manual a graphical description of several of the eyepatterns will be included.

6.6.1. V.32.

DISP_setup	B7 B6 B5 B4 B3 B2 B1 B0	default:[0073]
7..2	spare	
1,0	eye	
	3	combined DSP-a DSP-b display
	2	DSP-b display
	1	echo canceller display
	0	DSP-a display

6.6.2. V.34.

DISP_setup: B7 B6 B5 B4 B3 B2 B1 B0
B2 B1 B0:
These bits select the kind of display:
0:receiver display + normal statusline
1:echo canceller display + normal statusline
2:line compensation spectrum of equaliser + normal statusline
3:probing spectrum + normal statuslinr
4:equaliser display + TX-statusline
5:equaliser display + RX-statusline

For the Normal statusline:
example:
3429 HL 28.8/26.4 D0
displayed parameters:
Symbolrate: The symbol rate (or nominal bandwidth)
possible values:
2400: bandwidth 2400 Hz
2743: bandwidth 2743 Hz
2800: bandwidth 2800 Hz (not used)
3000: bandwidth 3000 Hz
3200: bandwidth 3200 Hz
3429: bandwidth 3429 Hz

Carrier selection for TX path, Carrier selection for RX path
possible values:
H: High carrier selected
L: Low carrier selected

Bit speed selected for both RX and TX (RX/TX)
possible values
2.4 -> 28.8, with increments of 2.4
if the secondary channel is enabled, this value can be incremented by 0.2 (200 bps) or 0.4 (400 bps)

Measured value
Different parameters can be measured and displayed within the normal statusline (see below)

For the TX-statusline:
example:
TX: P:1 S:1 T:0 NL:0 F:1
displayed parameters:

P:	0: precoding off in TX path 1: precoding on in TX path
S:	0: reduced shaping in TX path 1: expanded shaping in TX path
T:	0: 16 state trellis in TX path 1: 32 state trellis in TX path 2: 64 state trellis in TX path
NL:	0: no Non-Linear encoding present in TX path 1: Non-Linear encoding present in TX path
F:	Filter: 0,1,2,..,9,A: indicating the preemphasis filter which is selected by the remote side for the local transmitter

For the RX-statusline:
example:
RX: P:1 S:1 T:0 NL:0 F:1
displayed parameters:

P:	0: precoding off in RX path 1: precoding on in RX path
S:	0: reduced shaping in RX path 1: expanded shaping in RX path
T:	0: 16 state trellis in RX path 1: 32 state trellis in RX path 2: 64 state trellis in RX path
NL:	0: no Non-Linear encoding present in RX path 1: Non-Linear encoding present in RX path
F:	Filter: 0,1,2,..,9,A: indicating the preemphasis filter which is selected by the local side for the remote transmitter

B3:
setting this bit will clear the errormessage. The bit is automaticly cleared afterwards.
B7 B6 B5 B4:
When using the normal statusline, these bits will select the parameter which is measured on scope (eye-monitor,Output in hex). These parameters are displayed as the last value on the normal statusline

0: TX_precLev: scaling constant in TX path which deals with compensation for non-linear encoding. Q15 format

1: RX_precLev: scaling constant in RX path which deals with compensation for non-linear encoding. Q15 format

2: AverRXCompensation: scaling constant in RX path which is calculated using a postagc at the beginning of DATASTATE. Q15 format

3: -

4: Average Free processing cycles in DSP-B per 4D symbol (2 baud)

5: RoundTripDelay in number of 2D-symbols (used for Bulk delay)

6: -

7: Decision length for trellis in hex, expressed in 4D symbol intervals
Possible values ar 0x5,0xA or 0x12.
If the line quality is good, the smaller value is taken in order to shorten the througput delay.

8: TX Training Progress
(=^AB_DFECoeff + 0)

9: Mean Absolute Error (DSP Read register 0x0D)
(=^AB_DFECoeff + 1)

A: Remote Info0 (RemoteInfo0)
(=^AB_DFECoeff + 2)
This 16 bit word contains the following information about the remote modem:

0: if 1, symbol rate 2743 is supported
1: if 1, symbol rate 2800 is supported
2: if 1, symbol rate 3429 is supported
3: if 1, symbol rate 3000,Low Carrier is supported in TX
4: if 1, symbol rate 3000,High Carrier is supported in TX
5: if 1, symbol rate 3200,Low Carrier is supported in TX
6: if 1, symbol rate 3200,High Carrier is supported in TX
7: if 0, transmission at symbol rate 3429 is disallowed
8: if 1, remote is able to reduce it's transmit power
9-B: value in the range 0-5, indicating the maximum allowed difference in symbol rates in the transmit and receive directions
C: if 1, symbol rate 2400,Low Carrier is supported in RX
D: if 1, symbol rate 2400,High Carrier is supported in RX
E: if 1, symbol rate 2743,Low Carrier is supported in RX
F: if 1, symbol rate 2743,High Carrier is supported in RX

B: RX Training Progress
(=^AB_DFECoeff + 3)

C: IF INFO Display is selected this location contains:FREQUENCY OFFSET

OTHERWISE : Remote MP sequence (MPRXVaria)
(=^AB_DFECoeff + 4)
This 16 bit word contains following information:

0: MP type:
0=type 0
1=type 1

1: 0

2-5: Maximum Call to Answer Signalling rate. The value is filled out by the Answer modem.
1=2400 bps
2=4800 bps
3=7200 bps
4=9600 bps
5=12000 bps
6=14400 bps
7=16800 bps
8=19200 bps
9=21600 bps
A=24000 bps
B=26400 bps
C=28800 bps

6-9: Maximum Answer to Call Signalling rate. The value is filled out by the Call modem.
1=2400 bps

2=4800 bps
3=7200 bps
4=9600 bps
5=12000 bps
6=14400 bps
7=16800 bps
8=19200 bps
9=21600 bps
A=24000 bps
B=26400 bps
C=28800 bps

A: Auxiliary channel bit
0: remote modem does not support auxiliary channel
1: remote modem supports auxiliary channel

B-C: Transmit channel Trellis Code selection:
0: 16 state code
1: 32 state code
2: 64 state code

D: Transmit channel Non-Linear encoding selection
0: Non-Linear encoding disabled
1: Non_Linear encoding enabled

E: Transmit channel shaping selection
0: Minimum shaping
1: Expanded shaping

F: MP acknowledge bit

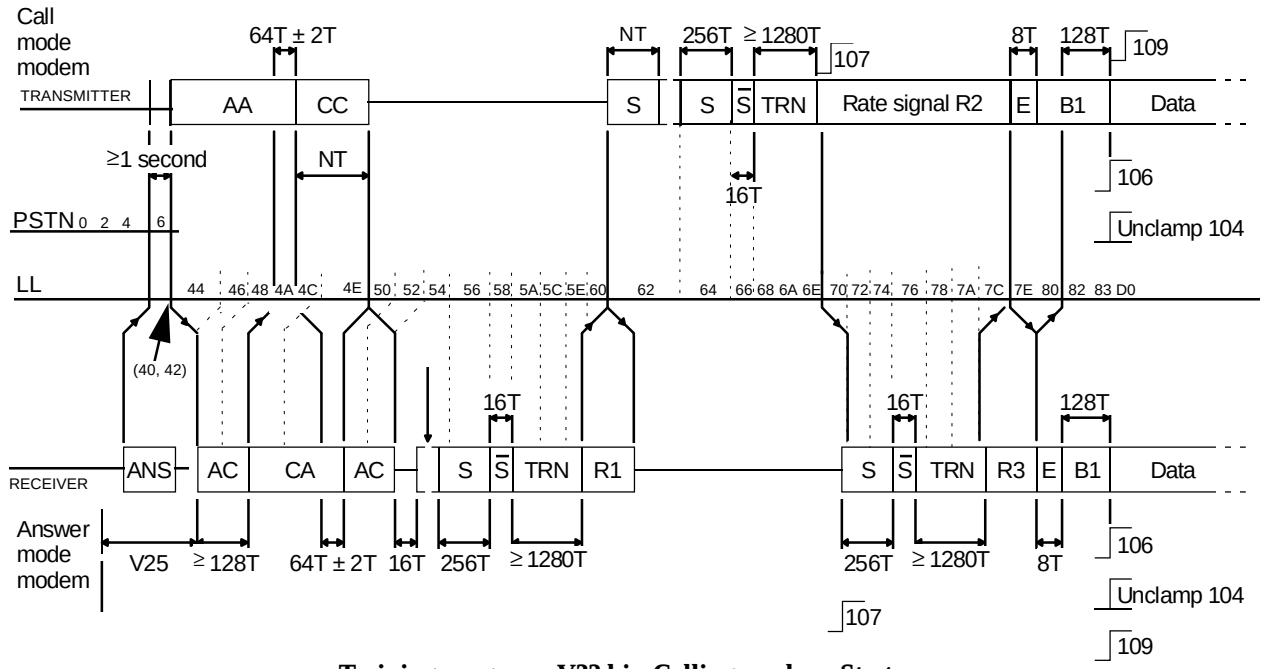
D: IF INFO Display is selected this location contains:FREE

OTHERWISE :FREE
(=^AB_DFECoeff + 5)

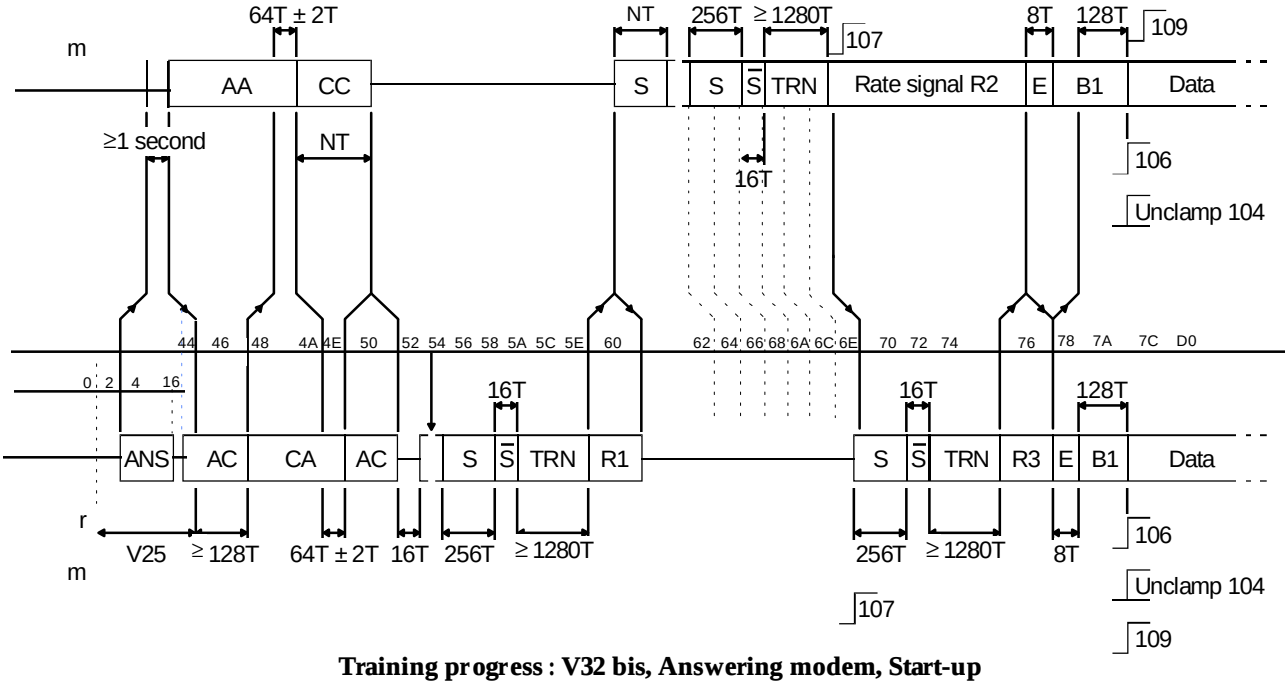
E: Transmit Buffering before read (even, should be >=2)

F: -

6.7. TRAINING STATE DIAGRAMS



Training progress : V32 bis, Calling modem, Start-up



Training progress : V32 bis, Answering modem, Start-up

6.8. SNR TABLE

Following mathcad calculations give the relation between SNR (at the receiver slicer) and bitrate for all the symbolrates in the case of V.34.

$$Sc := \begin{bmatrix} 0 \\ 1 \\ \frac{8}{7} \\ \frac{7}{6} \\ \frac{5}{4} \\ \frac{4}{3} \\ \frac{10}{7} \end{bmatrix}$$

HIGH BER SNR VALUES

$i := 1..6$ $\quad S_i := Sc_{i,0} \cdot 2400$

$j := 1..14$ $\quad R_j := j \cdot 2400$ $\quad R_{14} = 3.36 \cdot 10^4$

$$SNR_{i,j} := 4 + 3 \cdot \left\langle \frac{R_j}{S_i} \right\rangle$$

$SNR_{i,0} := S_i$ $\quad SNR_{0,j} := R_j$

$k := 2..6$ $\quad SNR_{k,1} := 0$

$k := 10..14$ $\quad SNR_{1,k} := 0$

$k := 12..14$ $\quad SNR_{2,k} := 0$

$k := 12..14$ $\quad SNR_{3,k} := 0$

$k := 13..14$ $\quad SNR_{4,k} := 0$

$k := 14$ $\quad SNR_{5,k} := 0$

$$SNR = \begin{bmatrix} 0 & 2400 & 4800 & 7200 & 9600 & 12000 & 14400 & 16800 & 19200 & 21600 & 24000 & 26400 & 28800 & 31200 & 33600 \\ 2400 & 7 & 10 & 13 & 16 & 19 & 22 & 25 & 28 & 31 & 0 & 0 & 0 & 0 & 0 \\ 2742.9 & 0 & 9.3 & 11.9 & 14.5 & 17.1 & 19.8 & 22.4 & 25 & 27.6 & 30.3 & 32.9 & 0 & 0 & 0 \\ 2800 & 0 & 9.1 & 11.7 & 14.3 & 16.9 & 19.4 & 22 & 24.6 & 27.1 & 29.7 & 32.3 & 0 & 0 & 0 \\ 3000 & 0 & 8.8 & 11.2 & 13.6 & 16 & 18.4 & 20.8 & 23.2 & 25.6 & 28 & 30.4 & 32.8 & 0 & 0 \\ 3200 & 0 & 8.5 & 10.8 & 13 & 15.3 & 17.5 & 19.8 & 22 & 24.3 & 26.5 & 28.8 & 31 & 33.3 & 0 \\ 3428.6 & 0 & 8.2 & 10.3 & 12.4 & 14.5 & 16.6 & 18.7 & 20.8 & 22.9 & 25 & 27.1 & 29.2 & 31.3 & 33.4 \end{bmatrix}$$

6.9. OVERVIEW MODEM STANDARDS

The table below gives an overview of the modulation standards present in the current software package, on all pages.

BITRATE	SNR DEC	SYMBOL RATE	MODULATION	LINE TYPE	DXTYPE (+#WIRES)	CHANN. DIV. TECHN.	CARR. FREQ.	NOM. BAND WIDTH	EXCESS BAND WIDTH
V21 0-300 bps	15	0-300	FSK	LL PSTN	2WFDX	FDM	1080 980 1180 TX-CALL-LOW 1750 1650 1850 TX-ANSWER-HIGH	+/- 100Hz +/- 100Hz	+=BINARY 0 = SPACE -=BINARY 1 =MARK
V.22 1200		600	4DPSK	LL PSTN	2WFDX	FDM	1200 TX-CALL-LOW 2400 TX-ANS-HIGH	+/- 300Hz +/- 300Hz	75%
						fixed compromise EQ (adaptive EQ on data not CCITT specified)			
600 V.22bis 2400	11..12 20	600 600	2DPSK 16QAM	LL PSTN	2WFDX	FDM	1200 TX-CALL-LOW 2400 TX-ANSWER-HIGH	+/- 300Hz +/- 300Hz	 75%
						fixed compromise EQ and adaptive EQ (on training and on data)			
1200	14	600	4QAM						
FORW V23 0-1200	15..16	0-1200	FSK	LL PSTN	2ASYMDX	FDM	1700 1300 2100 1500 1300 1700 420 390 450	+/- 400Hz +/- 200Hz +/- 30Hz	+ = BINARY 0 = SPACE - = BINARY 1 = MARK
BELL212A highspeed 1200 (~V.22/1200) lowspeed 0-300 (=Bell 103)		600 0-300	4DPSK FSK	LL PSTN LL PSTN	2WFDX 2WFDX	FDM FDM	1200 2400 1170 TX-CALL-LOW 2125 TX-ANSWER-HIGH	+/- 300Hz +/- 100Hz +/- 100Hz	75% + = BINARY 0 = SPACE - = BINARY 1 = MARK
V26 : forw 2400 (altA/B) backw V23 V26 bis : forw 2400 =V26 alt B 1200 backw V23 V26 ter : 2400 =V26 alt A 1200	14	1200 0..75 1200 1200	4DPSK A(0) B (45) 2DPSK (90, -90) 2DPSK (0, 180)	LL (M1020) PSTN LL/ PSTN	4WFDX 2WHDX 2WFDX	4 wire 2 wire EC	1800 Fixed compromise EQ (adaptive EQ (on data) not CCITT specified) 1800 1800 Fixed compromise EQ or adaptive EQ (on data)	+/- 600 +/- 600 +/- 600	No CCITT spec (V27bis: > or = 50%) No CCITT spec (V27bis: > or = 50%) 100%
V27 : forw 4800 backw V23 V27 bis : forw 4800 =V27+ad. Equal. 2400 =V26altA+ ad. Equal. backw V23 V27 ter =V27 bis (+ext.training for echo protection)	19 20	1600 0-75-? 0-75-?	8DPSK	LL (M1020) LL (M1040) PSTN	2WHDX 4WHDX 4WFDX 4WFDX 2WHDX	2 wire 4 wire Manual adjusted EQ Adaptive EQ (on training and on data) Adaptive EQ (on training)	1800	+/- 800	50% 50% >=50%

V29 9600 7200 4800	22 18 4	2400	16ADPSK 8ADPSK 4DPSK	LL (M1025)	2WHDX 4WHDX 4WFDX	4 wire Ad.EQ	1700	+/-1200	No CCITT spec
V.32 4800 9600 9600T V.32E 4800 7200T 9600 9600T 12000T 14400T	14 20 18 21 24	2400 2400	4QAM 16QAM 32TCQAM (trellis coded) 4QAM 16TCQAM 16QAM 32TCQAM 64TCQAM 128TCQAM	LL PSTN -	2WFDX -	EC Ad. EQ (on tr.)	1800 ?	+/- 1200	No CCITT spec (Telindus=15%)
V.32bis=V.32E - 9600 + Rate Negotiation (fast Step Up - Fallback Without retrain)									
V33 12000T 14400T		2400	64TCQAM 128TCQAM	LL	4WFDX	4 wire Ad. EQ (on tr.)	1800	+/- 1200	No CCITT spec (Telindus=15%)
V17 (~V33) 7200T 9600T 12000T 14400T		2400	16TCQAM 32TCQAM 64TCQAM 128TCQAM	PSTN	2WHDX	2 wire	1800	+/- 1200	No CCITT spec (Telindus=15%)
V.34 2400 4800 …. 28800		2400 2743 2800 3000 3200 3429	4D-TC-QAM	LL PSTN	2WFDX	EC	1800….. (2 carriers/ SYMBOL RATE)	+/- $\frac{\text{FSYMBOL}}{2}$	No CCITT spec

LEGEND

LL =	leased lines (private)
PSTN =	public switched telephone network
DX =	duplex
HDX =	half duplex (transmission in only one way (at a time))
FDX =	full duplex (simultaneous transmission in two ways, with same data rate in both ways)
ASYM DX =	asymmetrical duplex (simultaneous transmission in two ways, with different data rates in each way)
2W =	2 wire
4W =	4 wire
FSK =	frequency shift keying
DPSK =	differential phase shift keying
ADPSK =	amplitude modulated differential phase modulation
QAM =	quadrature amplitude modulation
TCQAM =	trellis coded QAM (2 dimensional)
4DTCQAM =	4 dimensional trellis coded QAM
FDM =	frequency division multiplexing = band splitting
EC =	echo cancellation
Excess bandwidth =	bandwith extra to the nominal bandwidth