TeleSoft

Protocols
for Developers

TsLink3®
General Overview

TABLE OF CONTENTS

1. Introduction

TsLink3 software provides a set of powerful, flexible building blocks for adding terminal endpoint (TE)- or network (NT)-side signaling protocols on an accelerated schedule and at a lower cost. The TsLink3 software source code files are organized as software layer entities according to the International Standards Organization Open System Interconnection (OSI) 7-layer model. TsLink3 software modules handle processing required for Layers 1, 2 and 3 of the OSI model.

The fundamental standards are specified in Recommendations from the IETF, ANSI, Frame Relay Forum and Telecommunication Standardization Sector of the International Telecommunication Union (ITU-T) international standards body, formerly known as the CCITT. Specifications for different national telephone networks ("switch variants") are largely derived from the ITU-T recommendations and are issued by national Postal, Telephone, and Telegraph (PTT) authorities and by manufacturers of ISDN central office switch equipment.

TsLink3 consists of software modules that can be used with TE- and NT- equipment according to the relevant standards.

TsLink3 provides a powerful, flexible, and efficient "message-based" API to configure and control the operations of the various software layers. The optional Terminal Adapter software module interprets the industry-standard modem ‘AT’ command interface.

Most of the TsLink3 software is operating system and hardware independent. All OS dependencies are localized in a few OS-interface functions and the software protocol state machines are completely hardware independent. The flexible TsLink3 software may be compiled within a standalone embedded environment, in a Dual-Port RAM-based coprocessor configuration, or as host-based software.

As part of the TsLink3 package, TeleSoft International has developed a family of modular software device drivers that includes several popular HDLC, Ethernet, UART, transceiver and communications controller semiconductor devices.

To further accelerate product development and launch, TsLink3 customers can also license hardware reference designs for several target configurations so that they begin with a more advanced starting point for product development.

1.1 Major TsLink3 Options

There are several TsLink3 software options that can be selected at compile time and/or run-time to adapt TsLink3 to a wide variety of telecom applications.

- Embedded, Coprocessor or Host-based
- Signaling channel HDLC, user data channel HDLC and UART hardware devices
- Real-time operating system interface
- ISDN or R2 switch variants to be compiled as run-time options (EuroISDN, U.S. National ISDN, etc.)
- Data protocols (e.g., Frame Relay (FR), PPP, ML-PPP, X.25, V.120, none)
- Voice supplementary services
- TE (User) -side or NT-side

Compile time options are selected by a combination of conditional compilation flag definitions and the list of files specified in the appropriate compile and link "make" file. Compile time options are selected before compilation by editing a set of "tuning" include header files, with options related to a particular software layer entity conveniently grouped in a tuning file specific to the software layer entity.

Run-time options are selected by the setting of global "customer user profile (CUP)" parameters located in RAM. How these CUP parameters are initialized is dependent on the implementation. In a coprocessor environment, the "other" processor running TsLink3 customer software (e.g., PC processor) may initialize the CUP parameters before restarting the TsLink3 software at boot time. In a stand-alone Terminal Adapter application, a small boot routine that runs before TsLink3 could initialize the CUP parameters from EEPROM.

1.2 Portability

TsLink3 is designed to be easily ported to a wide variety of software and hardware environments. A high level of portability is provided in the following key areas:

1.2.1 Hardware Independence

Most of the TsLink3 code is hardware independent, requiring little or no change to work with different chips. Hardware dependencies have been localized in relatively small Low-Level Driver (LLD) modules with a little glue logic added in the Management Entity. The LLD's hide device-specific details from the higher layers and provide hardware control including a uniform data transfer mechanism across channels for Layers 2 and above.

Protocol Layers 2 and 3 are completely hardware independent.

TsLink3 is written in ANSI C and has been ported by TeleSoft International to the:

- Intel 80X8X, Motorola 68XXX, and Motorola PPC processor families
- Microsoft, Borland, Intermetrics, High C, Diab and Gnu compilers

TsLink3 clients have ported TsLink3 code to numerous other processors and compilers.

When TsLink3 starts to run, it expects that a few basic hardware initialization steps have already been performed by client software. The steps include chip select register programming, stack pointer initialization, and compiler-dependent data segment register initialization. Compiler vendors often supply a startup routine prototype performing such steps.

The TsLink3 client usually makes a few modifications to the compiler startup routine prototype to create a small boot routine. This client-supplied boot routine sets up basic hardware sanity before jumping to the TsLink3 entry point function.

1.2.2 Operating System Independence

TsLink3 is compiled and linked as a collection of real-time tasks and interrupt service routines. Therefore, TsLink3 assumes the use of a real-time operating system but requires only the most basic type of task creation and message passing functions common to most real-time executives.

The bulk of TsLink3 code is totally operating system independent. All interactions between the real-time executive and the rest of the system are handled through a small number of OS interface routines, so portation to most real-time executives is quick and straightforward. The files that must be edited are concentrated in an OS interface subdirectory; the editing process consists primarily of changing the system calls that initialize tasks, and send and receive inter-task messages, from the syntax of an existing OS to the new OS.

For the Intel 80X8X, Motorola 68K and Motorola PPC microprocessor families, TeleSoft International has developed the optional TsRITE™ real-time executive to run with TsLink3. TsRITE is a real-time multi-tasking kernel optimized for telecom applications.

TeleSoft International has also ported TsLink3 to the EDX™, Linux, MQX, Nucleus™ RTX, pSOS+™, VRTX™, and VxWorks™ operating systems. TsLink3 clients have ported TsLink3 to many other commercial and proprietary operating systems.

1.3 Voice and Data Services

TsLink3 supports a range of voice and data services for several national and manufacturer switch variants. A different mix of services is available for each switch variant. The following fundamental "bearer capability" services are offered on all switches:

- speech
- unrestricted digital information, circuit-mode, 64 kbps
- unrestricted digital information, circuit-mode, 56 kbps [hardware-support dependent]
- unrestricted digital information, circuit-mode, 64 kbps with B-Channel X.25 implicit

Software hooks are in place for the "3.1 kHz audio" bearer service for several switch variants.

1.3.1 Data Services and Protocols

TsLink3 provides both "packet-switched" and "circuit-switched" data service options. The TsLink3 packet-switched option is X.25. For TsLink3 circuit-switched applications, the customer can choose one of the TsLink3 data protocols or a customer-supplied data protocol.

TsLink3 provides the following data protocols: Frame Relay, PPP, ML-PPP, X.25, V.120.

1.4 Code Structure

1.4.1 Standards-based

Detailed TsLink3 software structure closely follows the standards documents on which TsLink3 is based, making the standards a handy reference to most of the TsLink3 code.

1.4.2 Easy to determine the side-effects of a change

The Layer 2 and Layer 3 state processing machines are implemented with each protocol state as a separate source file using C language "case" statements for each message type or other event that can be received in that state. The key advantage of the state machine design is that it is easy to locate all of the code related to a particular stimulus (received message, timeout event, etc.) in a particular protocol state without understanding the entire software structure in detail.

By contrast, "table driven" implementations require the programmer to learn not only the underlying standards-based protocols, but also the arbitrary mechanisms created by the programmers of the state table processing motor. The need to learn the underlying protocols and the arbitrary mechanisms makes it more time-consuming and difficult with a table-driven implementation to trace all of the code executed when a given message is received in a given state, which makes it much harder to determine the side-effects of a change.

1.4.3 Performance

The efficient branching of TsLink3, based on state and received message type, typically has higher performance than the table-driven approach because the table-driven approach carries the increased overhead of numerous table searches.

2. Major TsLink3 Configurations

There are two major TsLink3 configurations that can be created at compile time, Message-Based Layer 4 Interface (see Figure 1) and Terminal Adapter (TA) (see Figure 2).

The format of the messages ("primitives") that pass between the Terminal Adapter User Interface entity (TAUI) and the Coordinating Entity (CE) are specified in the "TsLink3 Interface Reference Guide" section entitled "Coordinating Entity" and are a proper subset of the messages passed in the "Message-Based" TsLink3 configuration.

Most of the software layer entities (and the source code files that comprise each entity) are common to the two configurations. A particular configuration is selected at compile time by a combination of the conditional compilation flags that are defined and the set of files specified in the appropriate compile and link "make" file.

Note that the CE exchanges the same set of messages with the entity "above" it -- L4 or TAUI -- for both major configurations. In the "message-based" configuration, the messages are exchanged between the CE and the TeleSoft International client. In the TA configuration, the messages are exchanged between the CE and the TeleSoft-supplied (or client-provided) TAUI software.

The messages passed between the CE and the entity above it are used to setup and teardown voice and data calls, pass call progress information, and, optionally, transfer user data.

In the message-based configuration, Layers 4 through 7 are supplied by the TsLink3 client. The top TsLink3 layer is the CE which resides between Layers 3 and 4. Layer 4 (L4) communicates with the CE through one of a variety of possible physical interfaces (e.g., task message queues if L4 and CE run on the same processor, shared memory mailboxes if L4 and CE run on different processors).

It is easy to adapt TsLink3 to any physical interface because all messages from L4 to the CE are funneled through one "process message from L4" C language function and all messages from the CE to L4 are funneled through one "send message to L4" C function. Layer 4 may run on the same microprocessor as the CE or each entity may run on a different processor. Figure 1 illustrates the OSI software layers for the message-based TsLink3 configuration.

The format of the messages (also referred to as "primitives") that pass between L4 and CE in both directions are specified in the "TsLink3 Interface Reference Guide" section entitled "Coordinating Entity."

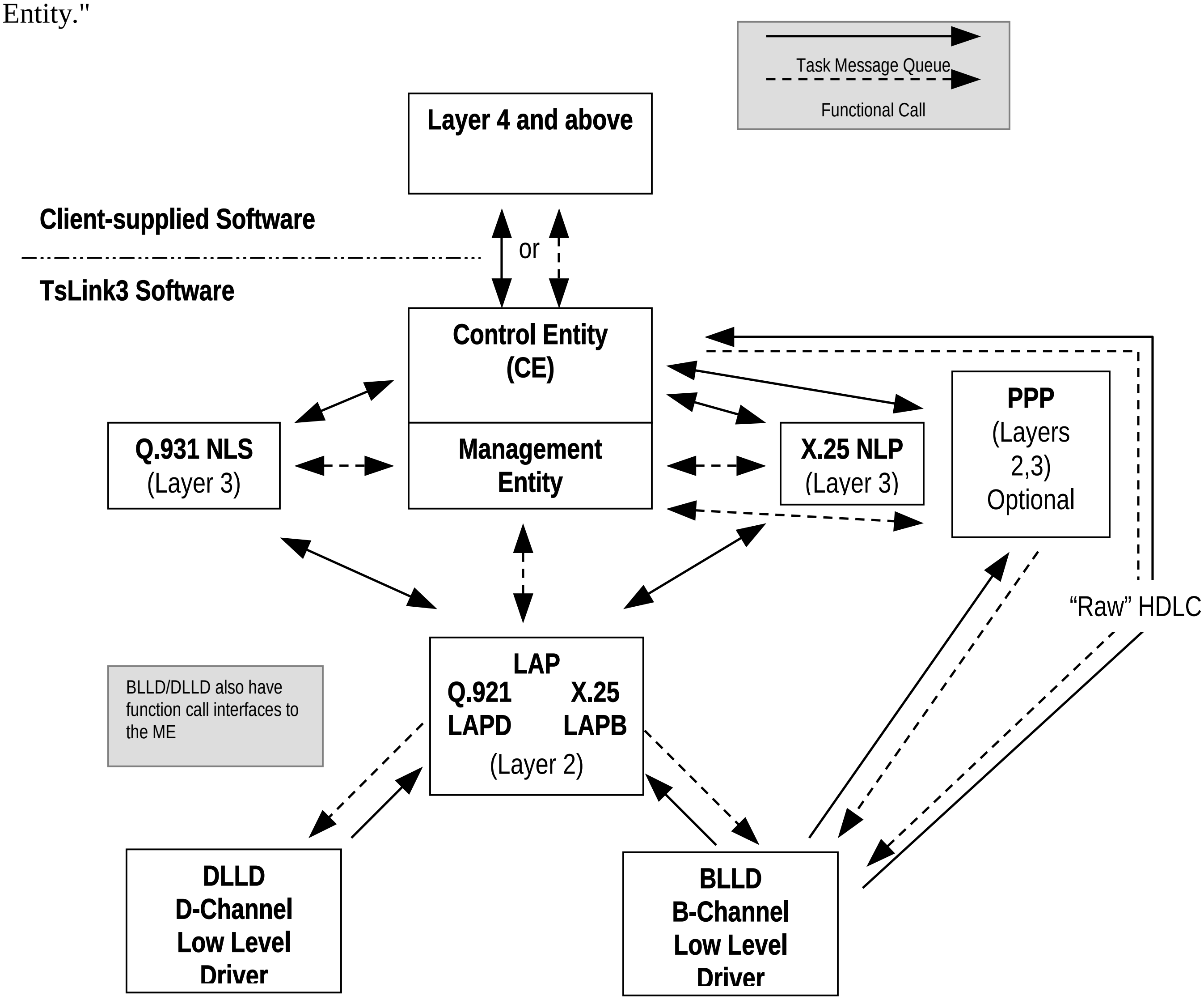

Figure 1. TsLink3 Message-Based Interface Software Block Diagram

The TsLink3 software emulates a modem supporting a subset of the modem AT command set. All required code modules are included in TsLink3 and run on a single microprocessor. Figure 2 illustrates the OSI software layers for the TsLink3 TA configuration. Note that the interface between the "ULLD" may be either a physical connection (such as an RS-232 port) or a logical "COM" port.

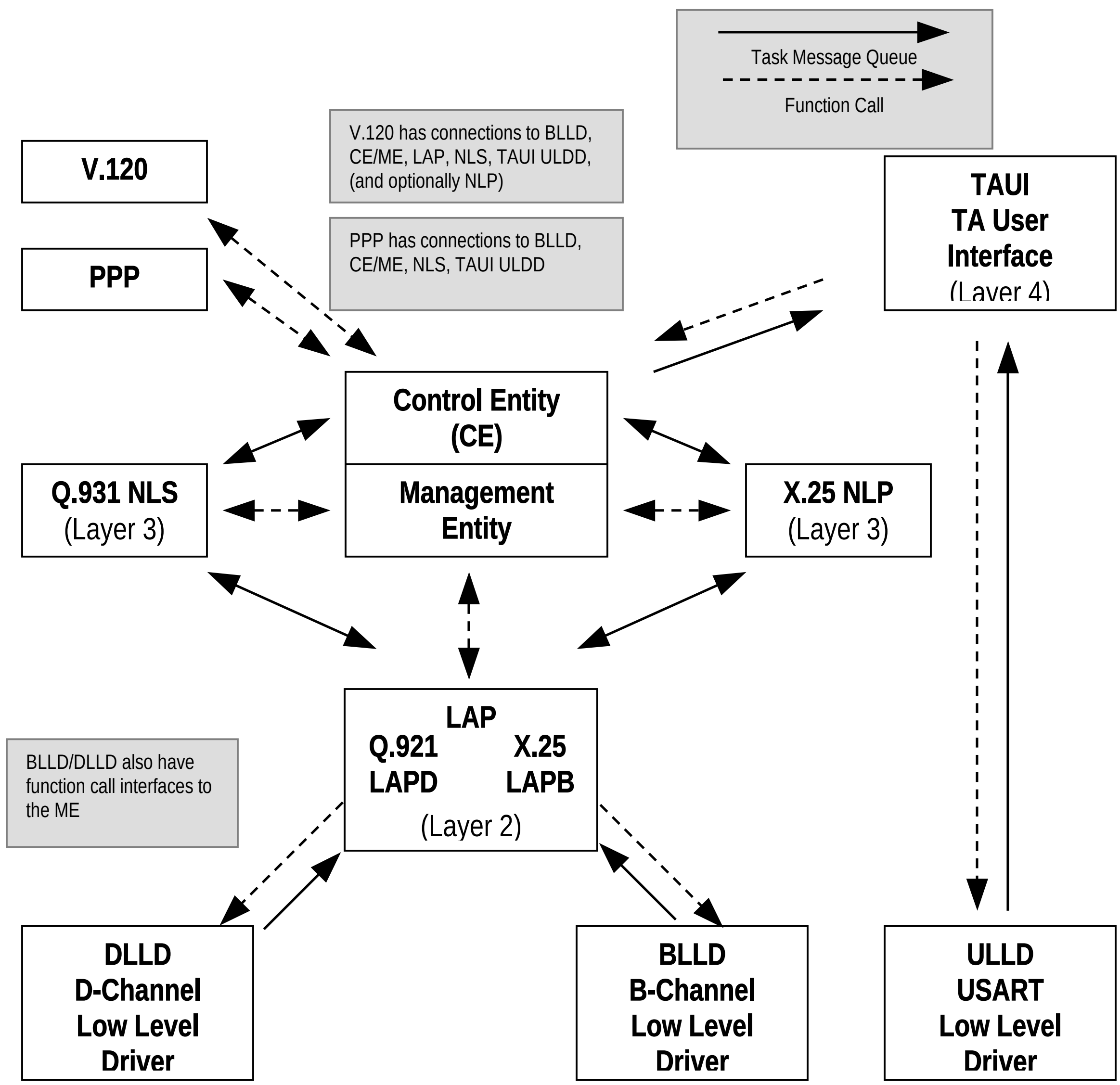

Figure 2. TsLink3 Terminal Adapter Software Block Diagram

2.3 TsLink3 Software Entities Common to Both Message-Based and TA Configurations

The individual software entities are briefly described in the following subsections.

2.3.1 Coordinating Entity – *applies to all protocols*

The Coordinating Entity (CE) coordinates (routes) commands/information between Layer 4 (or the Terminal Adapter User Interface software for TA applications) and the appropriate software entity: Layer 3 NLS/PRI_NLS/CAS, Frame Relay, X.25 NLP, Layer 2 LAPx, PPP, ML-PPP, or directly to the appropriate Low-Level Driver.

2.3.2 Management Entity – *applies to all protocols*

The Management Entity (ME) manages resources needed by all the layers. The resources include buffers, timers, and address parameters (e.g., Layer 2 Q.921 TEI's and Layer 3 X.25 LCI's). The ME performs TsLink3 power-on initialization, including a call to an initialization function for each software layer, and hardware device-specific initialization and control.

2.3.3 Layer 3 Network Layer Signaling Entity (NLS/PRI_NLS/CAS) – *ISDN and CAS specific*

The Layer 3 signaling entity is in charge of establishing, maintaining, and terminating network connections (i.e. connecting a path across the network) between two terminating end points. Layer 3 signaling procedures are switch dependent, each ISDN switch variant (e.g., U.S. National ISDN, Lucent 5ESS, Northern Telecom DMS-100, EuroISDN, QSIG) has slightly different signaling requirements which are called out clearly in the TsLink3 code. The ISDN signaling and routing procedures are described in ITU-T Recommendation Q.931.

NOTE: “CAS” includes E1 Channel Associated Signaling (CAS), R2, and T1 Robbed-bit signaling.

2.3.4 Point-to-Point Protocol (PPP) (includes Multi-Link PPP) – *PPP and ML-PPP specific*

TsLink3 PPP replaces OSI model layers 2 and 3 as a single layer entity. TsLink3 PPP is based on the IETF standards.

2.3.5 Q.921 LAPD, X.25 LAPB Layer 2,V.120 Layer 2 Entity (TsLink®) – *L2 specific*

Layer 2 uses the Link Access Protocol (LAP) class of HDLC protocols to provide an error-free transport mechanism for Layer 3. Layer 2 carries sequenced envelopes for the Layer 3 entities across the interface in any of the physical channels (D-Channel or B-Channels). The Layer 3 simply places the message targeted for its peer Layer 3 inside this envelope and hands it down to the Layer 2. It is then the responsibility of Layer 2 to carry this information to the other side. If the envelope is lost in the system, Layer 2 is responsible for re-delivering the envelope transparently to all higher layers.

Layer 2 protocols used to carry D-Channel signaling procedures are described in the ITU-T Q.921 Recommendation. LAPD can also be used to carry D-Channel user data. The TsLink3 Layer 2 protocol for optional data transfer on the B-Channel is described in ITU-T Recommendation X.25 (LAPB) and as layer 2 Single Link Protocol for X.75. V.120 is a variant of LAPD and provides reliable transfer of user data, usually across ISDN B-channels.

2.3.6 Q.922A Frame Relay PVC Support with Local Management Interface – *FR specific*

The Q.922A "Data Core" protocols can be used in place of Layer 2 (TsLink) for efficient transport of Frame Relay Permanent Virtual Circuits (PVCs). Frame Relay Switched Virtual Circuits (SVCs), written in accordance with the Frame Relay Forum agreement 4 (FRF.4), use the data core protocols for data transport.

2.3.7 Q.933 Frame Relay SVC Support, with Case B ISDN access – *FR SVC specific*

It is possible to access the FRF.4 subset of Q.933 via either BRI or PRI ISDN (Case B). Whether accessed via a circuit-switched connection or as a "nailed-up" HDLC link, the Q.933 SVCs allow signaling access for transient SVC establishment and disconnection. Both Network-side and User-side parts of the User-Network Interface (UNI) are provided.

2.3.8 X.25 Layer 3 Network Layer Packet Entity (NLP) – *X.25 specific*

NLP is an optional entity (not supported for PRI) that provides the X.25 packet layer protocol (PLP) capabilities for user-data transfer operations on the B-Channel. The NLP uses Layer 2 to transfer information to/from the LLDs and/or V.120.

The NLP entity supports ISO 8208 DTE-to-DTE functionality.

To provide its data transport service, the NLP performs a set of functions on user data units (or packets), such as segmenting and blocking, sequencing, flow control, expedited transfer, and reset.

There are two major TsLink3 configurations that can be created at compile time:

2.3.9 D-Channel LLD (DLLD) – *ISDN specific*

The primary function of this Low-Level Driver (LLD) is to handle the physical layer interface ('S' or 'U') transceiver activation/deactivation; secondarily, the LLD handles D-channel HDLC on devices that support it. Device-specific initialization and special chip features are controlled by the Management Entity.

The low level driver controls a transceiver chip such as the Motorola MC145574 or Infineon 2115 IPAC 'S' interface transceiver, or the MC145572 'U' interface transceiver. The IPAC combines both the transceiver function and the D-Channel HDLC function.

Devices such as the Motorola transceivers that do not have integrated hardware for D-Channel HDLC, require use of a separate HDLC chip (e.g., PPC SCC configured for HDLC). When a separate HDLC chip is used, D-Channel HDLC commands to/from Layer 2 are routed through the DLLD to/from the appropriate HDLC LLD (e.g., PPC SCC).

2.3.10 Bearer Channel LLD (BLLD) – *applies to all protocols*

The BLLD driver provides the user data transfer capability by handling the HDLC device programming for Layer 2. Device-specific initialization and special chip features are controlled by the Management Entity.

The driver controls the Infineon 82525 HSCX, Motorola PPC SCC (in HDLC mode) or similar HDLC interface hardware.

2.4 TsLink3 Software Entities Typically Used Only in the TA Configuration

2.4.1 V.120 Rate Adaption

The V.120 module provides the V.120-based protocol capabilities for data transfer operations. V.120 is an ITU-T protocol used to "rate adapt" between the relatively slow bit rates of the typical async RS-232 interface (e.g., 38400 bps) and the higher-speed 64 kbps ISDN interface. V.120 may also be used in a host-based application for special applications which need interworking with terminal equipment or special low-overhead data transfers.

2.4.2 Terminal Adapter User Interface (TAUI)

The Terminal Adapter User Interface module allows user access to the ISDN protocol layers via an RS-232 interface (ITU-T 'R' Reference point). The modem AT command set is mostly implemented in the TAUI entity with a small section implemented in the ULLD for real-time performance issues.

2.4.3 USART LLD

The USART device driver provides communication with the user when a UART or USART is used as the user-interface device. User data can be routed to/from the B-Channel using a rate adaption protocol such as V.120.