T1/E1 Device Driver Software

Application Programming Interface

NComm, Inc., 254 North Broadway, Suite 106, Salem, NH 03079

NOTICE

T1/E1 Device Driver Software
Application Programming Interface
Printed in the USA

Salem, NH

Revision History

November, 1998	Initial Version
April, 2005	Current Version 6.42

Contents

Section 1
Introduction

This document describes the NComm device driver API for the Dallas Semiconductor family of T1/E1 framers. The device drivers provide a portable C-code interface to assist T1/E1 application development on these industry standard devices. The device driver is implemented so it does not require operating system services and thus is portable across different operating systems and platforms. Along with the details of the driver level interface (API), this document introduces the necessary background required for successful T1 and E1 implementation.

The device driver, created by NComm, conforms to the driver level API found in NComm's full T1/E1 Trunk Management Software (TMS). The API can be used to interface into overhead management functions provided by the user, or can interface directly into NComm's full TMS offering yielding a pre-integrated, fully functional and standard compliant result.

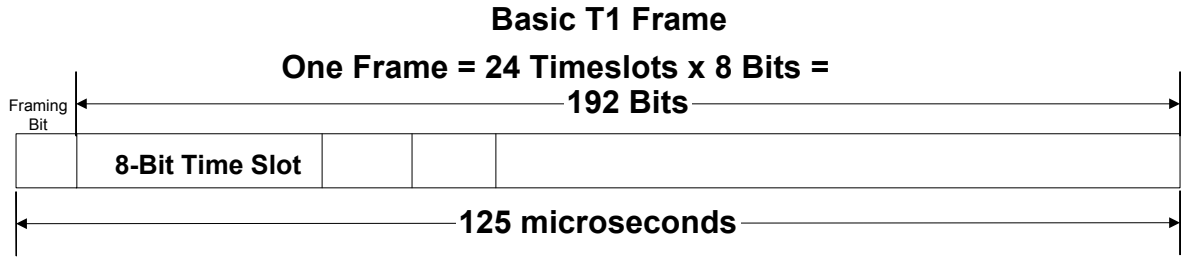
Products requiring T1/E1interfaces face the significant task of providing many low-level functions so that the applications conform to the different T1 or E1 standards. The NComm T1/E1 TMS supplies a complete solution for configuration, alarm processing and response, and Performance MONitoring including loop backs and Threshold Crossing Alerts that meet all relevant standards. If also implementing the full NComm TMS package, a higher-level API will be available to interface into the users value added functionality. Visit www.ncomm.com for more information.

For further technical information for WAN development, go to http://www.ncomm.com/new_site/main/wan_design_wan_protocols.htm and request a complementary Communications Developer Handbook (pdf).

T1

T1 provides a 1.544 MHz electrical interface. The T1 signal can carry channelized traffic or unchannelized traffic. The T1 signal consists of payload bits that are used to carry the data over the T1 trunk, and framing bits that are used to determine where the payload is located. In unchannelized applications, the payload bits carry data traffic such as frame relay or ATM. In channelized traffic, the payload is partitioned into timeslots and is used to carry voice traffic or call control such as ISDN or SS7.

The T1 signal consists of a time-multiplexed frame with one framing bit and 192 payload bits as shown in the following diagram. In unchannelized applications, the payload will consist of a stream of bits. In channelized T1 applications, the payload will be divided into 24, 8-bit time slots.



The T1 frame is repeated every 125 microseconds, which leads to the frequency of 1.544Mhz (193/0.000125). There are three main types of framing which are present on T1:

1. Super Frame format – also known as D4
2. Extended Super Frame – also known as ESF
3. SLC-96 or TR-008 Framing format

These different framing formats all use the same basic T1 frame, but the definition of the framing bit is different and will be described later.

Alarms

Alarms are used to detect and notify maintenance personnel of problems on the T1 or E1. There are three types of alarms:

1. RED alarms
2. BLUE alarms also known as Alarm Indication Signal (AIS)
3. YELLOW alarms also known as Remote Alarm Indication (RAI)

Alarms are created from defects. Defects are momentary impairments present on the trunk or line. If a defect is present for a sufficient amount of time (the integration time), then the defect becomes an alarm. Once an alarm is declared, the alarm is present until after the defect clears for a sufficient period of time. The time it takes to clear is called the de-integration time. The table below shows the defects, the alarms and the typical integration and de-integration times for T1 per ANSI T1.231.

Defect	Alarm	Integration Time	De-Integration Time
Loss of Signal Loss of Frame	RED	2.5 Seconds	10 Seconds
Remote Alarm Indication (RAI)	YELLOW	0.5 Seconds	0.5 Seconds
Alarm Indication Signal (AIS)	BLUE	2.5 Seconds	10 Seconds

Framing

The different framing formats carry the alarm information differently. To understand this, we need to look at the details of the framing formats. As indicated before, the Framing bit in a T1 frame repeated every 125 microseconds in the 193rd bit. The framing bit position consists of two types of bits, the Terminal Framing (Ft) and Signaling Framing (Fs) bits. In SF and SLC-96, the Ft bits are the same – a repeating 0, 1, 0, 1, 0, 1 pattern while the Fs bits are different.

Super Frame Framing

In Super Frame Framing, the framing patterns is as follows:

Frame	1	2	3	4	5	6	7	8	9	10	11	12
Fs		0		0		1		1		1		0
Ft	1		0		1		0		1		0	

In Super Frame Framing, frame number 6 and frame number 12 are signaling frames. In channelized T1 applications using robbed-bit signaling, these frames are used to contain the signaling information. In frame numbers 6 and 12, the least significant bit of all 24 timeslots is "robbed" to carry call state information. The bit in frame 6 is called the A bit and the bit in frame 12 is called the B bit. The combination of AB defines the state of the call for the timeslot that these two bits are located in.

Extended Super Frame Framing

Extended Super Frame (ESF) Framing is similar to Super Frame except that the super frame has been "extended" to 24 frames instead of 12 frames. In addition, the advancements in technology have eliminated the need to have a framing bit every 193rd bit. With ESF, the framing bit occurs once every 772 bits (4 frames) as shown in the FPS position below:

Frame	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
FPS				0				0				1				0				1				1
FDL	M		M		M		M		M		M		M		M		M		M		M		M	
CRC		C1				C2				C3				C4				C5				C6		

The other bit positions are used for the Facility Data Link (FDL) and a CRC-6 check sum.

The FDL is used as a point-to-point link between the customer premise and the network and is used for facility maintenance functions. The FDL does not pass through the network. That is, once a local T1, between the network and the customer premise, connects to the network, the FDL is terminated.

The FDL carries two types of traffic:

1. Bit Oriented Codes (BOC)
2. High-level Data Link Control (HDLC) Packets

BOC codes are repeated 16-bit long Binary ”111111110CCCCC0” sequences where “CCCCCC” is the BOC command word. BOC codes are used to control loopbacks, to indicate timing synchronizations source, to indicate Yellow Alarms, etc. To send a valid BOC sequence, it must be present on the T1 line for a minimum of 10 repetitions. Most recognition algorithms will recognize a BOC sequence if it receive 7 valid sequences out of 10.

The other type of traffic on the FDL is HDLC packets. Two standards cover the HDLC packets carried on the FDL.

1. ANSI T1.403 – This standard requires that, once per second, a packet is transmitted that contains performance data representing performance data that the receiver is detecting. Four seconds of information is transmitted so that recovery operations may be initiated in case an error corrupts a packet.
2. AT&T TR-54016 – This standard contains requirements for monitoring the performance of the T1. Once the performance data is collected, it can be retrieved via the FDL from the far end via a command-response protocol.

The last item carried in the framing bit position is the CRC-6 checksum. The CRC-6 pattern contains the checksum over the previous frame. It allows bit errors on the T1 to be detected.

In Extended Super Frame Framing, frame number 6, 12, 18 and 24 are signaling frames. In channelized T1 applications using robbed-bit signaling, these frames are used to contain the signaling information. In frame numbers 6, 12, 18 and 24, the least significant bit of all 24 timeslots is “robbed” to carry call state information. The bit in frame 6 is called the A bit and the bit in frame 12 is called the B bit, the bit in frame 18 is called the C bit, and the bit in the 24 frame is called the D bit. The combination of ABCD defines the state of the call for the timeslot that these four bits are located in.

SLC-96 Framing

AT&T invented SLC-96 framing for their SLC-96 product. SLC-96 is also generally known as TR008. We will use the term TR008 to describe our product features. The detailed description of the product can be found in the Telcordia Document GR-8-CORE. The purpose of the SLC-96 product was to provide standard telephone service (POTS e.g., Plain Old Telephone Service) in areas of high subscriber density, but back-haul the traffic over T1 facilities. To support the equipment, which is likely in an underground location, the T1s needed methods to provide:

1. Indicating equipment failures to maintenance personnel

2. Indicating failures of the POTS lines
3. Testing the POTS lines
4. Providing redundancy on the T1s

The manner that SLC-96 framing supports these features is using the framing bit position in the SLC-96 frame. The SLC-96 Super Frame, which is 72 frames long follows:

Frame	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Fs		0		0		0		1		1		1		0		0		0		1		1		1
Ft	1		0		1		0		1		0		1		0		1		0		1		0	

Frame	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
Fs		C1		C2		C3		C4		C5		C6		C7		C8		C9		C10		C11		0
Ft	1		0		1		0		1		0		1		0		1		0		1		0	

Frame	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72
Fs		1		0		M1		M2		M3		A1		A2		S1		S2		S3		S4		1
Ft	1		0		1		0		1		0		1		0		1		0		1		0	

The C1-C11 bits are the concentration bits. These bits are used to map POTS lines to timeslots on the T1 especially in oversubscribed conditions. That is, when more POTS lines are provided than can be carried on the T1s.

The M1-M3 bits are used for maintenance activities.

The A1-A2 bits are used for conveying alarm information from the remote device.

The S1-S4 bits are used for controlling protection switching.

In SLC-96 framing, frame number 6 and frame number 12 are signaling frames and every set of 12 frames there after. In channelized T1 applications using robbed-bit signaling, these frames are used to contain the signaling information. In frame numbers 6 and 12, the least significant bit of all 24 timeslots is “robbed” to carry call state information. The bit in frame 6 is called the A bit and the bit in frame 12 is called the B bit. The combination of AB defines the state of the call for timeslot that these two bits are located.

In-band Loopback Activation and De-Activation

Loopbacks are used for testing T1 lines. To support testing, an in-band loopback is used to place the T1 in remote, also known as line, loopback. A remote loopback causes the bits received on the T1 to be looped, un-modified, back to its source.

Sending the loopback pattern activates an in-band loopback. The pattern must be sent for at least 5 seconds. The pattern overwrites the entire payload in the T1, thus corrupting any calls or data

traffic. The framing bit may or may not still be present. The loopback is invoked when the pattern is removed.

The loopback is torn down when an in-band loop down pattern is transmitted for a period of 5 seconds. Of course, the times mentioned in this section are the nominal times per ANSI T1.403 but may be different in different installations.

Signaling

Signaling is how calls are passed on the T1 facility via the signaling bits indicated before. There are signaling bits in both the receive direction and the transmit direction. These bits described the state of a call on the timeslot.

In ESF framing, the ABCD bits are used, while in SF and SLC-96 framing the AB bits are used. So, how many different call states can you potentially have with two bits, AB? The obvious answer is 4. However, this is not correct. As it turns out, there are potentially 9 different call states. The way this is done is using the concept of tri-level signaling. If we look just at the A bit, it can be a 0 or a 1 but it can also be a toggle. A toggle is when in one super frame the bit is a 0 while in the next it is a 1 and it toggles every other super frame. Thus, the A bit can have 3 different states, which is termed tri-level signaling. In SF and SLC-96, a call can have 9 different states in each direction. ESF framing does not use tri-level signaling and has 16 possible states.

The method to determine how to interpret the signaling bits depends upon the call model being used. ANSI T1.403, AT&T PUB 43601, GR-303, and GR-8 all define different call models to interpret the signaling bits. The call models defined by these standards include the following:

- Loop start
- Loop start with RLCF
- Ground Start
- Ground Start with RLCF
- Loop-Reverse Battery Signaling
- Network provided reverse battery signaling
- E & M Signaling
- Customer-installation-provided loop-start supervision (FXS/FXO)
- Private line auto ring
- Ring down
- Superimposed Ringing Multiparty
- Direct Inward Dialing Dial Pulse Terminating
- Frequency Selective Ringing Multiparty
- Single Party
- Superimposed Ringing Multiparty
- Universal Voice Grade
- Coin CF/DTF
- Multiparty Signaling

CAS is the original signaling system used by E1 and provides 4 signaling bits for every channel. In CAS, channel 16 is reserved for signaling and the A/B/C/D bits for each channel are divided

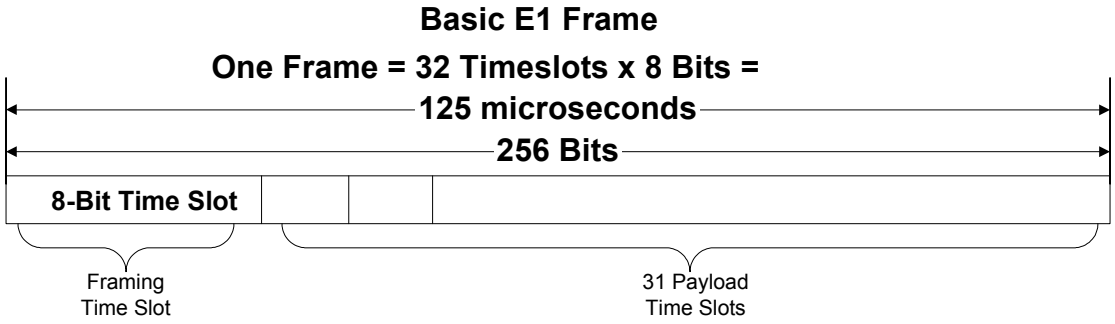
among 16 frames. Frame 0 contains the alignment signal, alarm, and spare bits. Frame 1 contains the A/B/C/D bits for channel 1 in the upper half of the channel and the A/B/C/D bits for channel 16 in the lower half. The remaining 14 frames follow the frame 1 format accordingly.

In recent years, the term Robbed-Bit Signaling (RBS) has been replaced by CAS, which is now often used to refer to bits that are associated with a specific channel whether it is in the T1 or E1 format.

E1

E1 provides a 2.048 MHz electrical interface. The E1 signal can carry channelized traffic or unchannelized traffic. The E1 signal consists of payload bits that are used to carry the data over the E1 trunk, and framing bits that are used to determine where the payload is located. In unchannelized applications, the payload bits are used to carry data traffic such as frame relay or ATM. In channelized traffic, the payload is partitioned into timeslots and is used to carry voice traffic or call control such as ISDN or SS7.

The E1 signal consists of a time-multiplexed frame with 8 framing bits and up to 248 payload bits as shown in the following diagram. In unchannelized applications, the payload will consist of a stream of bits. In channelized E1 applications, the payload will be divided into up to 31, 8-bit time slots and one framing time slot.



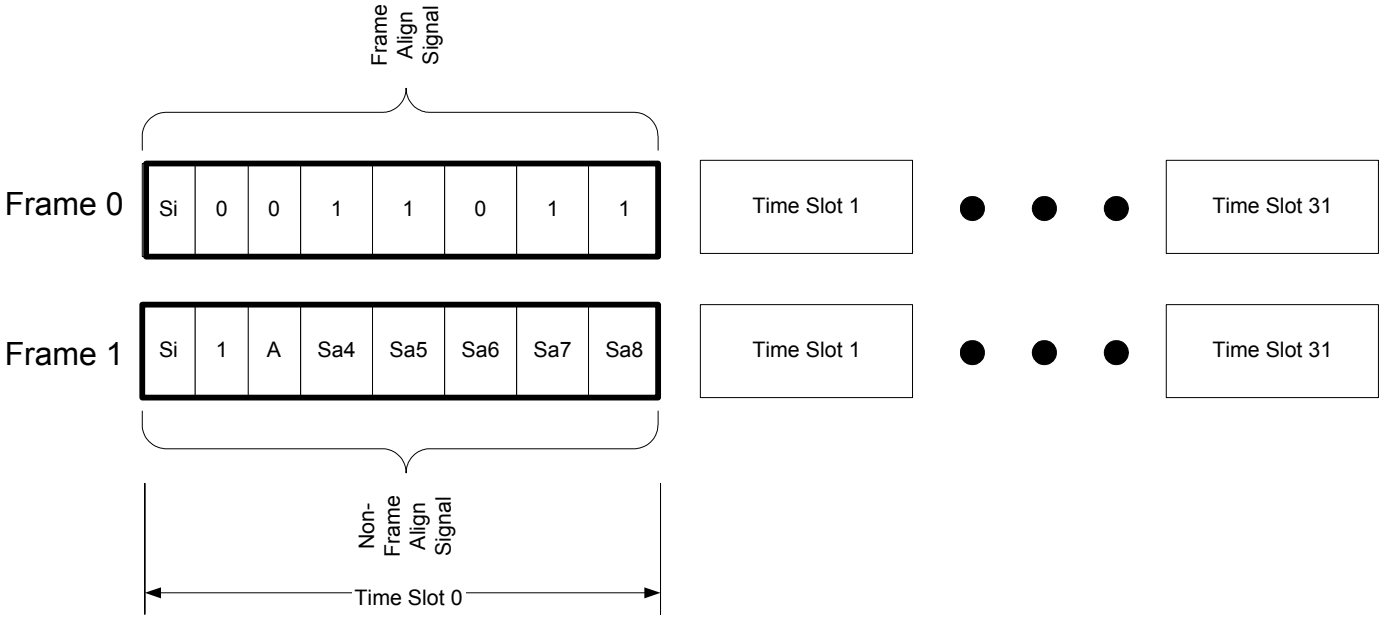
The E1 frame is repeated every 125 microseconds, which leads to the frequency of 2.048 MHz (256/0.000125). Four types of framing are present on E1:

1. Basic E1 Framing
2. E1 Framing with Signaling Multi-Frame Alignment.
3. E1 Framing with CRC4 Multi-Frame Alignment.
4. E1 Framing with CRC4 Multi-Frame Alignment and Signaling Multi-Frame Alignment.

These different framing formats all use the same basic E1 frame, however the differences in the framing sequence are described below.

Framing

E1 framing consists of a dual frame pattern. The first frame pattern, shown below as frame 0, contains the Frame Align Signal (FAS). The FAS is what is used to find the remaining parts of the E1 frame. The second frame pattern contains the Non-Frame Align Signal (NFAS).

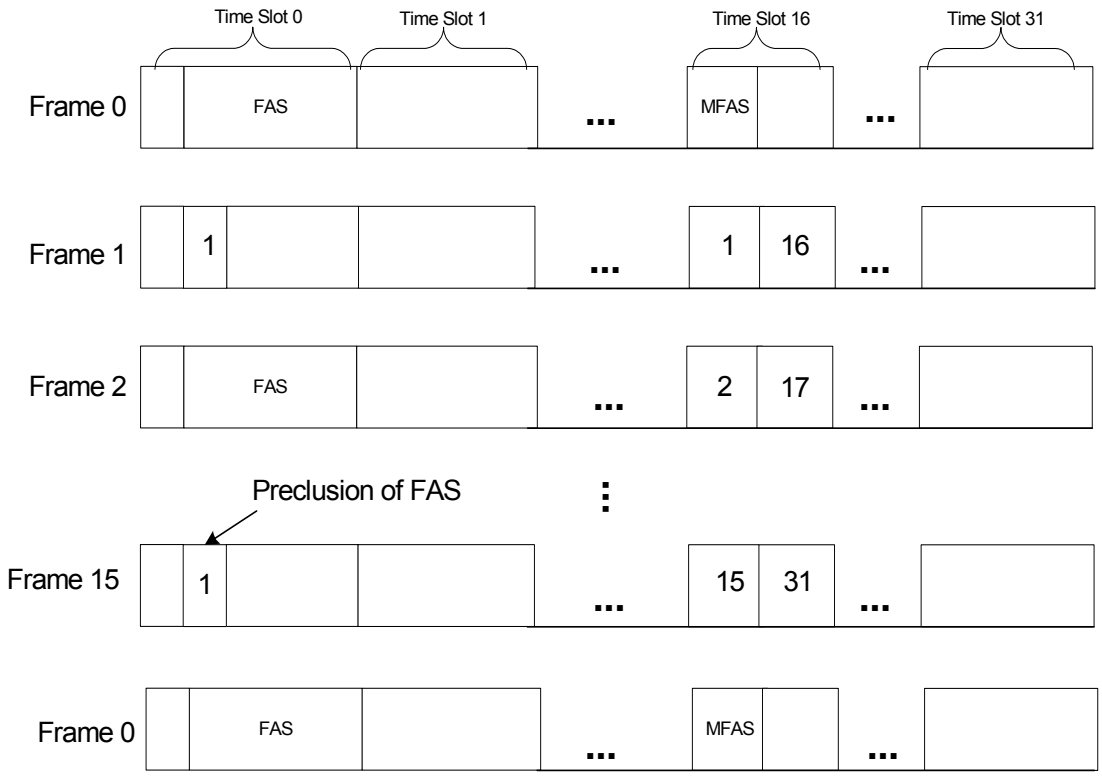


The first time slot in each frame is used to provide frame detection. In E1, an entire time slot is dedicated to framing information as well as other information. In the FAS frame, time slot 0 contains the Si (international) bit and the bit pattern 0011011. An E1 framer will look for this bit pattern to establish basic frame alignment.

In addition to the FAS pattern, the NFAS will be checked to detect the Non-Frame alignment signal. The "1" in the NFAS is used to validate the NFAS signal. The NFAS also contains the Si bit, the A-bit and five Sa bits. The A bit is used to indicate the Remote Alarm Indication (RAI) to the far end. When the A-bit is a 1, the RAI is asserted and when the A-bit is a 0 the RAI is not asserted. The Si bit is the international bit. Its use is reserved for crossing international boundaries. In most cases, the Si bit will be set to a one. There are five Sa bits, Sa4 through Sa8. The Sa bits are the national bits and are nominally set to all 1s when not used. When used, the Sa bits are for synchronization status messages, loop back requests, and other uses.

E1 Framing with Signaling Multi-Frame Framing

The E1 Framing with Signaling Multi-Frame Framing extends the E1 frame from a two-frame sequence to a 16-frame sequence as shown in the following diagram.



The E1 frame is extended to 16 frames by the use of time slot 16 to carry the CAS signaling information. The first frame, named "Frame 0" above, is used to indicate the start of the 16-frame sequence. In this frame, time slot 16 contains the Multi-Frame Alignment Signal (MFAS) that consists of a 4-bit pattern of all zeros in the first four bits of the time slot. The second 4 bits contain the three Extra Bits (X-bits) and the Multi-Frame Remote Alarm Indicator bit (Y-bit). When E1 is used to carry Channel Associated Signaling, it must use one of the E1 frame formats that contains the MFAS. Because of the use of all zeros as the Multi-Frame Alignment Signal, no other entry in time slot 16 can contain all zeros.

In addition to carrying the MFAS, time slot 16 also carries the Channel Associated Signaling bits for call processing. There are 4 bits for each voice channel and are grouped two sets of 4 bits in each timeslot. The 8 bits continued in Frame 1 correspond to the signaling bits for timeslot 1 and timeslot 17. The 8 bits contained in Frame 2 correspond to the signaling bits for timeslot 2 and timeslot 18. This pattern continues for all 30-voice channels.

E1 Framing with CRC Multi-Frame Framing

In the E1 Framing with CRC Multi-Frame the Si bits are re-defined and are used to carry a CRC-4 checksum pattern. The 16-frame Multi-Frame is divided into two Sub Multi-Frames (I and II) and the alignment is done via the CRC frame alignment sequence in the first bit. In Sub Multi-Frame I, this sequence is the pattern B"0001" and in Sub Multi-Frame II, the sequence is B"011", with both patterns located in the Si bit position of the frame. In addition to the CRC alignment bits, the first bit position contains a CRC-4 checksum and E-bits. This is shown in the chart below:

	Sub Multi-Frame	Frame Number	Bits 1 to 8 of the Frame							
			1	2	3	4	5	6	7	8
Multi-Frame	I	0	C_1	0	0	1	1	0	1	1
		1	0	1	A	S_{a4}	S_{a5}	S_{a61}	S_{a7}	S_{a8}
		2	C_2	0	0	1	1	0	1	1
		3	0	1	A	S_{a4}	S_{a5}	S_{a62}	S_{a7}	S_{a8}
		4	C_3	0	0	1	1	0	1	1
		5	1	1	A	S_{a4}	S_{a5}	S_{a63}	S_{a7}	S_{a8}
		6	C_4	0	0	1	1	0	1	1
		7	0	1	A	S_{a4}	S_{a5}	S_{a64}	S_{a7}	S_{a8}
	II	8	C_1	0	0	1	1	0	1	1
		9	1	1	A	S_{a4}	S_{a5}	S_{a61}	S_{a7}	S_{a8}
		10	C_2	0	0	1	1	0	1	1
		11	1	1	A	S_{a4}	S_{a5}	S_{a62}	S_{a7}	S_{a8}
		12	C_3	0	0	1	1	0	1	1
		13	E*	1	A	S_{a4}	S_{a5}	S_{a63}	S_{a7}	S_{a8}
		14	C_4	0	0	1	1	0	1	1
		15	E*	1	A	S_{a4}	S_{a5}	S_{a64}	S_{a7}	S_{a8}

E1 Framing with CRC-4 Multi-Frame

The CRC-4 checksum bit contains the CRC-4 checksum of the previous CRC-4 Sub Multi-Frame. The E-bits are used to inform the far end of received CRC-4 errors. When a bad CRC-4 Sub Multi-Frame is received, an E-bit in the reverse direction is set to indicate the error. If both CRC-4 Sub Multi-Frames are in error, both E-bits are set. Using the CRC-4 errors and E-bits, both ends can determine which direction has difficulties in passing traffic error free.

When using the CRC-4 Multi-Frame, the Sa bits also have expanded functionality. Instead of being only one bit, each Sa bit becomes 8 bits in the CRC-4 Multi-Frame. One use of the expanded Sa bits is to carry the Synchronization Status Message. Since the SSM is only 4-bits long, the Four Sa bits are repeated in each Sub Multi-Frame. This is shown in the chart below:

S_{an1}, S_{an2}, S_{an3}, S_{an4} **n = Sub-Frame** **$_a$ = 4, 5, 6, 7, 8 Depending upon the network**	**Synchronization Quality Level (QL) description**
0000	Quality unknown (existing synchronization network)
0001	Reserved
0010	See ITU G.811
0011	Reserved
0100	SSU-A - See G.812

0101	Reserved
0110	Reserved
0111	Reserved
1000	SSU-B – See G.812
1001	Reserved
1010	Reserved
1011	Synchronous Equipment Timing Source (SETS)
1100	Reserved
1101	Reserved
1110	Reserved
1111	Do not use for synchronization

E1 Framing with CRC Multi-Frame and Signaling Multi-Frame Framing

The E1 framing with CRC Multi-Frame and Signaling Multi-Frame combine both the CRC-4 multi-frame as well as the Signaling Multi-Frame. Although both Multi-Frame signaling are 16 frames long, they are not necessarily aligned with each other. When both framing schemes are present, the features associated with each are available.

Alarms

Alarms are used to detect and notify maintenance personnel of problems on the E1. The alarms present in E1 are very similar to those of T1. These alarms are defined below:

1. Loss of Signal (LOS) alarms
2. Loss of Frame (LOF) alarms
3. Alarm Indication Signal (AIS) alarms
4. Remote Alarm Indication (RAI) alarms

Alarms are created from defects. Defects are momentary impairments present on the trunk or line. If a defect is present for a sufficient amount of time (the integration time), then the defect becomes an alarm. Once an alarm is declared, the alarm is present until after the defect clears for a sufficient period of time. The time it takes to clear is called the de-integration time. The table below shows the defects, the alarms and the default integration and de-integration times for E1. The times selected for E1 are the same as the times for T1 since there is no specific time specified for E1 integration and de-integration timers.

Defect	Alarm	Integration Time	De-Integration Time
Loss of Signal	LOS	2.5 Seconds	10 Seconds
Loss of Frame	LOF	2.5 Seconds	10 Seconds
Remote Alarm Indication (RAI)	RAI	0.5 Seconds	0.5 Seconds
Alarm Indication Signal (AIS)	AIS	2.5 Seconds	10 Seconds

A note should be made about the Loss of Frame defect in E1. With the four different types of framing in E1, the Loss of Frame defect is a composite of the defects associated with each framing component as shown in the following table:

	Basic E1 Frame	E1 with Signaling Multi-Frame	E1 with CRC-4 Multi-Frame	E1 with CRC-4 Multi-Frame and Signaling Multi-Frame
Loss of Basic Frame Alignment Inability to locate the FAS	Used to detect LOF defect	Used to detect LOF defect	Used to detect LOF defect	Used to detect LOF defect
Loss of CRC-4 Frame Alignment Inability to detect the CRC-4 framing pattern			Used to detect LOF defect	Used to detect LOF defect
Loss of Signaling Frame Alignment Inability to locate the MFAS		Used to detect LOF defect		Used to detect LOF defect

Signaling

Signaling is how calls are passed on the E1 facility. This is done via the signaling bits indicated in the two E1 framing formats that have Signaling Multi-Frame. Signaling bits are transferred in both the receive direction and the transmit direction. Unlike T1 where the signaling bits describe

the state of the call, interpreting the signaling bits in E1 depends upon the previous state of both the receive and transmit direction.

The method to determine how to interpret the signaling bits depends upon the call model being used. ITU specification Q.422 defines the call model to interpret the signaling bits in E1. As an alternative to CAS signaling, ISDN and SS7 may be used to place phone calls over an E1 facility. Typically, the timeslot used for carrying ISDN or SS7 is also time slot 16. Consequently, basic E1 or E1 with CRC-4 Multi-Frame framing schemes must be used if ISDN or SS7 is to be carried.

Framer Driver Interface

The NComm framer interface is designed to be a standardized, chip-independent interface to the hardware. NComm supplies drivers for common framer chips such as the Infineon and Dallas Semiconductor families of T1/E1 devices. However, the driver for any framer can be easily implemented via the Framer Driver Interface API.

The Framer Driver Interface API consists of the following three main functions:

_te1DrvCTRL (drvref, fcode, …)
_te1DrvCLBK (drvref, fcode, …)
_te1DrvPOLL (drvref, fcode, …)

These three functions permit asynchronous control, notification, and polling of information to/from TMS.

_te1DrvCTRL	This function is called by TMS to configure the framer hardware. For example, `_te1DrvCTRL` is called to set the framer into ESF framing format. `_te1DrvCTRL` is part of the T1/E1 framer driver. If a framer chip is designed into the hardware, and it does not have an NComm supplied driver, the customer needs to write `_te1DrvCTRL`.
_te1DrvCLBK	This function calls TMS to notify the line manager about events that have occurred on the T1/E1 line. For example, `_te1DrvCLBK` is called when a Bit Oriented Code (BOC) word is received on the Facility Data Link (FDL).
_te1DrvPOLL	This function is called by TMS to find out instantaneous information about the line. For example, find the counts of CRC errors for the last second. If a framer chip is designed into the hardware, and it does not have an NComm supplied driver, the customer needs to write `_te1DrvPOLL`.

Figure 3 shows how the device driver API is logically arranged. Note that the framer device driver owns two of the three API calls, whereas TMS owns the third. When an NComm-supplied device driver is used, the two API calls are already implemented within that driver.

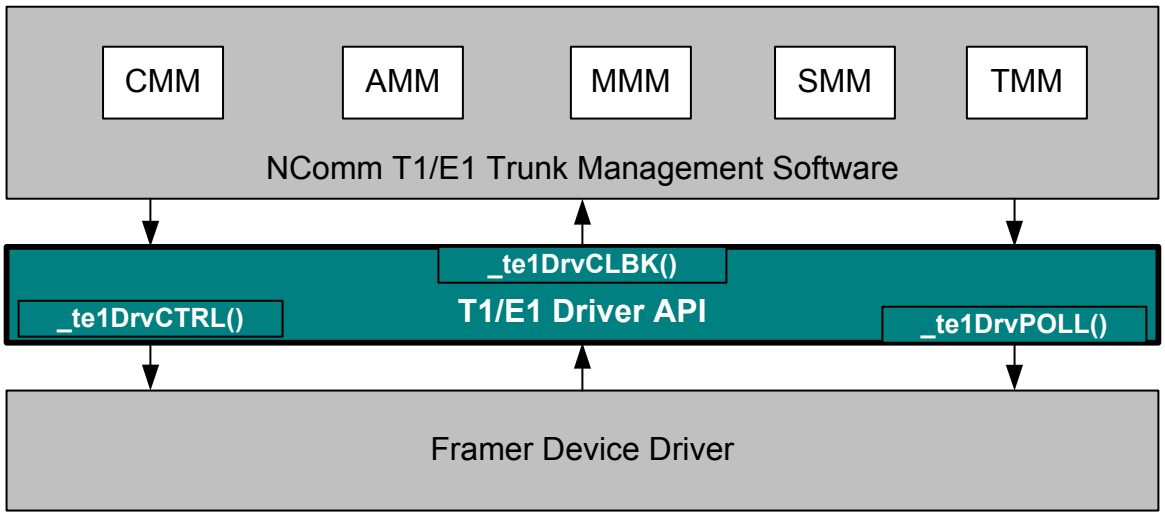


Figure 3: T1/E1 Device Driver Management API

Where:

CMM	Configuration Management Module: Base Software
AMM	Alarm Management Module: Base Software
MMM	Maintenance Management Module: Optional Software
SMM	Signal Management Module: Optional Software
TMM	TR008 Management Module: Optional Software

Standards Compliance

The NComm TMS provides customers with T1 and E1 Trunk Management according to the listed standards below. It is provided as standard ANSI-C code with two levels of APIs. The first level provides an easy interface, an API, between the customer's application and the NComm TMS. The second level provides a similar API interface between the NComm TMS and the device drivers.

Typically, NComm also provides the device drivers. NComm device drivers provide a portable C-code interface to assist T1/E1 application development on the industry standard T1/E1 devices. The device driver is implemented so it does not require operating system services and thus is portable across different operating systems and platforms.

The features described in this document are designed to be used in production systems. They provide a standardized interface that is preserved across different platforms with different devices. The NComm TMS creates the interface between trunk management requirements, the customer's application, and the hardware-framing device.

The NComm TMS handles the requirements of:

- ANSI T1.231, Telecommunications - Digital Hierarchy - Layer 1 In-Service Digital Transmission Performance Monitoring
- ANSI T1.403, Telecommunications - Network and Customer Installation Interfaces - DS1 Electrical Interface
- ANSI T1.408/T1.403.01, Telecommunications - Network and Customer Installation Interfaces - ISDN Primary Rate Layer 1 Electrical Interface Specification
- ATT TR-54016, Technical Reference Requirements For Interfacing Digital Terminal Equipment To Services Employing The Extended Superframe Format
- ITU-T G.703, Series G: Transmission System And Media, Digital Systems And Networks; Digital Transmission Systems - Terminal Equipment - General; Physical/Electrical Characteristics Of Hierarchical Digital Interfaces
- ITU-T G.704, Series G: Transmission Systems And Media, Digital Systems And Networks; Digital Transmission Systems - Terminal Equipment- General; Synchronous Frame Structures And Used At 1544, 6312, 2048 and 44 736 Kbit/S Hierarchical Levels
- ITU-T G.826, Series G: Transmission And Media, Digital Systems And Networks; Digital Transmission Systems - Digital Networks - Quality And Availability Targets;

Error Performance Parameters And Objectives For International, Constant Bit Rate Digital Paths At Or Above The …

- ITU-T Q.422, Clauses For Exchange Line Signaling Equipment

The NComm TMS package also contains the robbed-bit signaling models that handle the robbed-bit signaling requirements of:
ANSI T1.403, Telecommunications - Network and Customer Installation Interfaces - DS1 Electrical Interface

AT&T TR-008, Digital Interface Between The SLC-96 Digital Loop Carrier System And A Local Digital Switch

BELLCORE/Telcordia GR-303, Integrated Digital Loop Carrier System Generic Requirements Objectives And Interface, Tables 12-3 and 12-4.

ATT PUB 43801, Digital Channel Bank Requirements and Objectives, November 1982.

NComm's T1/E1 robbed-bit signaling provides users with clean detection of call state changes to their application(s). The NComm package handles signaling freezing and de-bouncing. Timers are provided for guard times, wink times, flash times, and digit pulse times as well as dial pulse collection and generation.

T1/E1 Terminology

The terminology found in this document is based on the definitions found in the various standards documents. The most commonly used terms are noted below.

Alarm Indication Signal (AIS) - A signal transmitted in lieu of the normal signal to maintain transmission continuity and to indicate to the receiving equipment that there is a transmission interruption located either at the equipment originating the AIS signal or upstream of that equipment.

Alternate Mark Inversion (AMI) - A line code that employs a ternary signal to convey binary digits, in which successive binary ones are represented by signal elements that are normally of alternating positive and negative polarity and of equal amplitude, and in which binary zeros are represented by signal elements that have zero amplitude. North American implementations use signal elements representing binary ones that are non-zero for only half the unit interval (50% duty cycle).

Bipolar Violation - A non-zero signal element in an AMI signal that has the same polarity as the previous non-zero signal element.

B8ZS (Bipolar with 8-zero substitution) - An AMI line code with the substitution of a unique code to replace occurrences of eight consecutive zero signal elements. 000VB0VB replaces each block of eight successive zeros, where B represents an inserted non-zero signal element conforming to the AMI rule, and V represents an inserted non-zero signal element that is a bipolar violation.

Blue Alarm - An AIS signal.

Bit Oriented Code (BOC) - A message sent over the FDL of an ESF formatted T1 that controls maintenance operations on the T1.

Bursty Errored Seconds (BES) - See Severely Errored Sections (SES).

CAS (Channel Associated Signal) - A method of signaling that assigns signaling bits that correspond to their timeslot.

Channelized, Channel, Channel Timeslot - A DS1 frame is said to be channelized if the payload timeslots are assigned in a fixed pattern to signal elements from more than one source, each operating at a slower digital rate. For which case the 192 payload bits represent 24, 8-bit channel time slots, making up 24 individual 64kbits/s (DS0) bit streams; each DS0 is referred to as a channel. The eight contiguous digit timeslots associated with a DS0 channel are referred to as a channel time slot.

Cyclic Redundancy Check (CRC) - A method of detecting the existence of errors in the transmission of a digital signal using polynomial division.

D4 Frame - Fourth generation digital channel bank.

DS1 (Digital Signal 1, T1) - A digital signal transmitted at the nominal rate of 1.544 Mbits/s.

E1 - A digital signal transmitted at the nominal rate of 2.048 Mbits/sec

Elastic Store/Slip Buffer - Used to adjust for differences in timing between the T1/E1 interface and the system timing.

Errored Second (ES) - A one second interval with an error. See TR 54016 (T1) or see G.823 (E1).

Excessive Zeros (EXZ) - The occurrence of more than 15 contiguous zeros in an AMI signal. For a B8ZS coded signal, when more than 7 contiguous zeros occur.

Extended Super Frame (ESF) - A DS1/T1 framing format of 24 frames. In this format, 2 Kbps are used for framing pattern sequence, 4 Kbps are used for the Facility Data Link, and the remaining 2 Kbps are used for CRC. A one second interval with an error. See TR 54016.

Facility Data Link (FDL) - An embedded overhead channel within the ESF format.

Frame - T1: A set of 192 timeslots for the information payload, preceded by a one-digit timeslot containing the framing (F) bit, for a total of 193 timeslots. The payload is often DS0-channelized into 24 channel timeslots. E1: A set of 256 bits organized into 32 timeslots numbered 1 to 32. Timeslot 1 contains the framing pattern, CRC-4, Si bits, Sa Bits, and A bit. When CAS signaling is used, timeslot 17 is used to carry the signaling bits for each channel.

Framer Loopback - An internal (within the framer) loopback that tests the path up to where framing is introduced. Used for diagnostics.

High-Level Data Link Control (HDLC) - A very common bit-oriented data link protocol (OSI layer 2), standardized by ISO.

HDB3 - A zero substitution code used in E1 signaling.

In-Band - Using or involving the information digit timeslots of a DS1 frame; i.e., bit assignments of a frame exclusive of the framing bit.

Line Build-Out (LBO) - An electrical network used to increase the electrical length of a cable section.

Line Coding Violation (LCV) - The occurrence of either a Bipolar Violation or Excessive Zeros.

Line Loopback - A loopback in which the signal returned toward the source of the loopback command consists of the full 1.544 Mbits/s signal with (1) bit sequence integrity maintained, (2) no change in framing, and (3) no removal of bipolar violations.

Local Loopback - An internal (within the framer) diagnostic loopback in which the signal returned towards the source is framed.

Loopback - A state of a transmission facility in which the received signal is returned towards the sender.

Loop Down Code - Code sent to disable loopback.

Loop Up Code - Code sent to set up loopback.

Loss Of Signal (LOS) - When no pulses are detected of either positive or negative polarity.

Multi-Frame - A method used in E1 to provide CAS signaling.

Out Of Frame (OOF) - A framing error occurred.

Path Coding Violation (PCV) - See Bipolar Violation.

Payload - The 192 information bits of a DS1 frame. Can be used with or without CRC-4 multi-frame.

Payload Loopback - A loopback in which the signal returned toward the source of the loopback command consists of the payload of the received signal (with bit sequence integrity retained) and newly generated ESF framing (not necessarily maintaining the integrity of the channel timeslots, frames, or superframes of the received signal.). The newly generated ESF data link contains a valid performance report message with a value of one in every LB-labeled bit position for the duration of the loopback indicating the signal is the result of a payload loopback.

Severely Errored Seconds (SES) - This is a T1/E1 performance measure. See TR-54016 or G.826 for detailed information.

Signal Bits - Special bits on the T1/E1 used for placing calls.

SLC-96 (Subscriber loop carrier) - another DS1 framing format.

Super Frame (SF) - a DS1/T1 framing format of 12 frames.

Yellow Alarm - A Remote Alarm Indication signal. Indication from the far end equipment that it is having difficulties receiving the near end signal.

Zero Destuffing -Used in HDLC packets; remove the zero following 5 consecutive ones.

This page is left blank intentionally

Section 2
The Framer Driver Interface Functions

This section describes the three main generic framer driver interface functions:

- _te1DrvCTRL
- _te1DrvCLBK
- _te1DrvPOLL

The information includes:

- A description of the function
- The function and arguments
- The function codes and their parameter values
- Descriptions of the function codes
- The return values
- Example

The section also contains some of the typedefs and constants that are used by the functions of this section.

The _te1DrvCTRL Function

Description

The `_te1DrvCTRL`() function permits the TMS to asynchronously control and configure T1/E1 span capabilities in the low-level framer device driver. All `_te1DrvCTRL` calls will change the behavior of the trunk.

The implementation of the `_te1DrvCTRL` function must be done as part of the framer device driver. The framer device driver must implement the predefined function codes given in the `apiTE1Dv.h` header file. Calls to the `_te1DrvCTRL`() function must *not* block.

The Function and Arguments

The `_te1DrvCTRL` function takes the following form:

```
#include “apiTE1Dv.h”
extern int _te1DrvCTRL (void *drvref, TE1DCTRL_FC fcode…);
```

Where:

drvref	A void * pointer to the drivers internal storage containing line related information. This pointer is returned to TMS from the driver during the TE1DCTRL_RESET call.
fcode	The function code that identifies the action to be performed by the framer circuit. This is typically one of several predefined values as specified in the `apiTE3Dv.h` header file. Each function code may have zero or more parameters. These parameters communicate further information to the function code. Refer to the following table for fcode definitions and other parameter values.
…	Depending upon the **fcode**, more arguments may be required. See the specific **fcode** for further details.

The Function Code and Other Parameter Values

If needed, other parameters as specified by the fcode definitions.

Fcode	Details
TE1DCTRL_RESET	Description: The RESET call initializes the device to a known state. The device will be left transmitting AIS, thus allowing the software to be initialized before starting the device. This call is called before the Operating System is running. param1 = (int) line number param2 = (void *) hardware device base address param3 = (int) hardware device sub channel param4 = (void *) pointer to TE1_CONFIG param5 = (void *) TMS reference param6 = (void *) pointer to load DRV reference
TE1DCTRL_UNRESET	Description: Unbind and shut down a Line. param1 = (void *) pointer to TE1_CONFIG
TE1DCTRL_CALLBACK	Description: Used to support a programmable driver callback. For example: if you want to use the driver to assist with diagnostics before running the primary application, you can set the callback to a diagnostics-flavored callback which prints out diagnostic information, then, when the diagnostics are done, the callback can be reset to the normal primary-app function. When the NComm T1/E1 TMS is initialized it automatically sets the proper callback. param1 = (void *) callback function-pointer
TE1DCTRL_UTILINIT	Description: Enable or disable the NON-alarm processing. param1 = (int) ENABLE or DISABLE
TE1DCTRL_SETSIDE	Description: Set the line operating mode as either user side or network side. param1 = (int) USR_SIDE or NET_SIDE
TE1DCTRL_OSTIC	Description: Send a tick. Should be sent at the rate previously set up with the TE1DCLBK_HOOKOSTIC. param1 = (unsigned long) user-defined

Fcode	Details
TE1DCTRL_DEVICE	Description: A catchall, generic fcode that lets the user define up to 4 unsigned longs as parameters. Thus the application can, via a call to the span-level TE1LCTRL_DEVICE fcode, gain access to the device driver without making a direct driver call, thus maintaining architectural integrity in the software. This code bypasses the NComm T1/E1 Trunk Management Software. param1 = (unsigned long) user-defined param2 = (unsigned long) user-defined param3 = (unsigned long) user-defined param4 = (unsigned long) user-defined
TE1DCTRL_TXCLOCK	Description: Used to select the Transmit clock source for the device. This is typically used if the clock is to be reprogrammed from its default setting. param1 = (int) one of CLOCK_TYPE
TE1DCTRL_ONESEC	Description: Enable or disable the one second interrupt. param1 = (int) ENABLE or DISABLE
TE1DCTRL_ALARMINIT	Description: Enables or disables alarm handling. param1 = (int) ENABLE or DISABLE
TE1DCTRL_ALARMTX	Description: Enables or disables a alarm transmission. param1 = (int) one of ALARM_TYPE param2 = (int) ENABLE or DISABLE
TE1DCTRL_RESYNC (E1)	Description: For the framer receiver to re-acquire the frame. No additional parameters

Maintenance Manager Module Fcodes

The following Fcodes are only available if the Maintenance Manager Module is present.

Fcode	Detail
TE1DCTRL_LOOPINIT*	Description: Enables or disables Loopback handling. param1 = (int) ENABLE or DISABLE

Fcode	Detail
TE1DCTRL_LOOP*	Description: Enables or disables a loopback function. param1 = (int) one of LOOP_TYPE param2 = (int) SET or CLEAR param3 = (int) code param4 = (int) bits
TE1DCTRL_EXCESSZERO* (T1)	Description: Enables or disables the counting of Excess Zeros. param1 = (int) ENABLE or DISABLE
TE1DCTRL_T1403INIT* (T1)	Description: Initialize T1.403 handling. param1 = (int) ENABLE or DISABLE
TE1DCTRL_T1403TXPKT* (T1)	Description: Sends a packet over the facility data link. param1 = (int) byte count of packet param2 = (void *) pointer to packet
TE1DCTRL_T1403TXBOC* (T1)	Description: Sends a Bit Oriented Code over the facility data link. param1 = (int) ENABLE or DISABLE param2 = (int) BOC to be enabled/disabled
TE1DCTRL_T1231INIT* (T1)	Description: Initialize T1.231 handling. param1 = (int) ENABLE or DISABLE
TE1DCTRL_TR54016INIT* (T1)	Description: Initialize TR-54016 handling. param1 = (int) ENABLE or DISABLE
TE1DCTRL_G826INIT* (E1)	Description: Initialize the framer's E1 G.826 handling. param1 = (int) ENABLE or DISABLE
TE1DCTRL_E1SASIINIT* (E1)	Description: Initialize the framer's E1 SASI handling. param1 = (int) ENABLE or DISABLE
TE1DCTRL_E1SASISABYTE* (E1)	Description: Sets up transmission of the Sa byte for an E1 trunk with CRC framing. param1 = (int) one of E1SABYTE_TYPE param2 = (int) 8-bit Sa byte value
TE1DCTRL_E1SASISABIT* (E1)	Description: Sets the value of the Sa Bits. param1 = (int) Sa4-Sa8 bits

Fcode	Detail
TE1DCTRL_E1SASISIBIT* (E1)	Description: Sets the value of the Si bits. Only valid in Non-CRC formatted E1 frames. param1 = (int) SET or CLEAR
TE1DCTRL_E1SASIXBIT* (E1)	Description: Sets the Extra bits of a multi-frame E1 trunk. param1 = (int) 3-bit X-Bit value

Signaling Manager Module Fcodes

The following Fcodes are only available if the Signaling Manager Module is present.

Fcode	Detail
TE1DCTRL_SIGNLINIT**	Description: Enable or disable signaling. param1 = (int) ENABLE or DISABLE
TE1DCTRL_SIGNLCLEAR**	Description: Disable robbed-bit signaling on specified timeslot. Note: timeslot numbers start from 1 not from 0 so the timeslot range is 1-24 for T1 and 1-32 for E1. param1 = (int) timeslot
TE1DCTRL_SIGNLIDLE**	Description: Used to configure a robbed-bit timeslot for idle transmission instead of robbed bit or clear-channel. Also defines the Idle bit pattern to be placed into a time slot when the time slot is set to idle. The Idle pattern will default to 0x17. Note: timeslot numbers start from 1 not from 0 so the timeslot range is 1-24 for T1 and 1-32 for E1. param1 = (int) timeslot param2 = (int) idle pattern
TE1DCTRL_SIGNLRBIT**	Description: Enable robbed-bit signaling on specified timeslot. In E1 applications, it enables CAS signaling. Note: timeslot numbers start from 1 not from 0 so the timeslot range is 1-24 for T1 and 1-32 for E1. param1 = (int) timeslot

Fcode	Detail
TE1DCTRL_SIGNLTXBITS**	Description: Send the specified bits on the specified timeslot. The format for the bits is 0000DCBA for ESF formatted T1 or multi-frame formatted E1. 0000baBA for SF formatted T1. If b=1 means the B bit is tri-level. If a=1 means the A bit is tri-level. Note: timeslot numbers start from 1 not from 0 so the timeslot range is 1-24 for T1 and 1-32 for E1. param1 = (int) timeslot param2 = (int) DCBA bits (MSB =D)

TR008 Manager Module Fcodes

The following Fcodes are only available if the TR008 Manager Module is present.

Fcode	Detail
TE1DCTRL_TR008INIT***	Description: Initializes TR008 handling. param1 = (int) ENABLE or DISABLE
TE1DCTRL_TR008CMAS***	Description: Enable/disable a CMAS channel. param1 = (int) bitmap of CMAS channels

Note: In the DRVR API, Timeslot values range from 0 to 23 for T1 and 0 to 31 for E1.
* Valid only if *Maintenance Manager* module is present.
** Valid only for *Signaling Manager* module.
** Valid only for *TR008 Module Manager* module.

(T1) – Valid only for a T1 Trunk
(E1) – Valid only for an E1 Trunk

Return Value

If the function call:

is supported/successful; returns a +1.
is not supported: the function returns ERR_NOSUPPORT (0).
errors: returns one of the predefined negative numbers. See the file apiTE1Dv.h for a list of error conditions.

Note that failure to support all predefined function codes may result in non-deterministic span behavior.

Example

This is an example of a control call that would be made to the driver. This call is to start transmitting AIS on span 0 – the first span.

```
#include apiTE1Ln.h
int retval;

retval =  _te1DrvCTRL(lineptr, TE1DCTRL_ALARMTX, TMS_TXAIS, SET);
```

The _te1DrvCLBK Function

Description

The `_te1DrvCLBK()` function permits the low-level framer driver to synchronously notify, via call back, the TMS of events and occurrences on the T1/E1 span. This call is typically placed within the framer driver interrupt handler so that notification can be made at the appropriate time. The framer driver must make this call with one of the predefined function codes as specified in the `apiTE1Dv.h` header file.

The `_te1DrvCLBK` call will *not* block in the NComm Trunk Management Software.

The Function and Arguments

The `_te1DrvCLBK` function takes the following form:

```
#include "apiTE1Dv.h"
extern int _te1DrvCLBK (void *drvref, TE1DCLBK_FC fcode…);
```

Where:

drvref	A void * pointer to the drivers internal storage containing line related information. This pointer is returned to TMS from the driver during the TE1DCTRL_RESET call and is passed back with each callback call.
fcode	The function code that identifies the action to be performed by the framer circuit. This is typically one of several predefined values as specified in the `apiTE3Dv.h` header file. Each function code may have zero or more parameters. These parameters communicate further information to the function code. Refer to the following table for fcode definitions and other parameter values.
…	Depending upon the **fcode**, more arguments may be required. See the specific **fcode** for further details.

The Function Code and Other Parameter Values

If needed, other parameters as specified by the fcode definitions. In the Driver API, Timeslot values range from 0 to 23 for T1 and 0 to 31 for E1.

Fcode	Detail
TE1DCLBK_HOOKINTERRUPT	Description: Informs the TMS that Hardware Interrupts have been enabled. param1 = (void *) driver specific – typically, the interrupt service routine which lives in the driver param2 = (void *) driver specific – typically, the span number
TE1DCLBK_UNHOOKINTERRUPT	Description: Informs the TMS that it has unhook/disable device interrupt. No additional parameters
TE1DCLBK_HOOKOSTIC	Description: Informs the TMS that the OS tick is enabled at the desired millisecond rate. The TE1DCTRL_OSTIC implements this rate. param1 = (int) milliseconds per tick param2 = (void *) load with returned handle param3 = (void *) function pointer param4 = (unsigned long) user defined structure param5 = (unsigned long) user defined structure
TE1DCLBK_UNHOOKOSTIC	Description: Informs the TMS that the OS tick has been disabled. param1 = (void *) pointer returned in the TE1DCLBK_HOOKOSTIC call.
TE1DCLBK_ONESEC	Description: This call back is done one time per second based upon the timing on the trunk. No additional parameters
TE1DCLBK_DEVICE	Description: Informs the TMS that a TE1DCTRL_DEVICE call has been completed. param1 = (unsigned long) user-defined param2 = (unsigned long) user-defined param3 = (unsigned long) user-defined param4 = (unsigned long) user-defined

Fcode	Detail
TE1DCLBK_FATALERROR	Description: Informs the TMS that the driver has returned a fatal error. param1 = (unsigned long) error code
TE1DCLBK_CRC	Description: Asks the TMS to calculate a 16-bit CRC. Param1 points to the CRC value to be returned. Param2 contains the length (in bytes) of the block of data. Param3 points to the block of data to be CRC'd. param1 = (void *) calculated CRC value param2 = (int) byte count param3 = (unsigned char *) pointer
TE1DCLBK_AUTOCRC	Description: Enables or disables the TMS from checking/generating the 16-bit CRC on all packets. If AUTOCRC is disabled, the TMS does not expect the CRC. param1 = (int) ENABLE or DISABLE
TE1DCLBK_AUTORAI	Description: Informs the TMS that Auto RAI is enabled or disabled. param1 = (int) ENABLE or DISABLE
TE1DCLBK_ALARMRX	Description: Informs the TMS that the trunk is receiving a new alarm condition. param1 = (int) one of ALARM_TYPE param2 = (int) SET or CLEAR

Maintenance Manager Module Fcodes

The following Fcodes are only available if the Maintenance Manager Module is present.

Fcode	Detail
TE1DCLBK_LOOP* (T1)	Description: Informs the TMS that the trunk has a new loopback status. param1 = (int) one of LOOP_TYPE param2 = (int) SET or CLEAR param3 = (int) code param4 = (int) bits
TE1DCLBK_SIGNLDEBOUNCE* (E1)	Description: Tell the TMS the status of the Signal Bit "de-bounce". param1 = (int) ENABLE or DISABLE

Fcode	Detail
TE1DCLBK_T1403RXBOC* (T1)	Description: Passes the received Bit Oriented Code (BOC) to the TMS. Returns (-1) when transmission ceases. param1 = (int) BOC code
TE1DCLBK_T1403RXPKT* (T1)	Description: Passes the HDLC packet received on the trunk to the TMS. If the TMS is validating the checksum, the checksum should be included. If the TMS is NOT validating the checksum, the packet should be passed without the checksum. param1 = (int) byte count of message param2 = (void *) pointer to message
TE1DCLBK_T1403PKTXMTCOMPLETE* (T1)	Description: Informs the TMS that the HDLC packet has completed transmission. No parameters
TE1DCLBK_E1SASISABIT* (E1)	Description: Informs the TMS of a new Sa bit value. param1 = (int) Sa4-Sa8 bits
TE1DCLBK_E1SASISABYTE* (E1)	Description: Informs the TMS of a new Sa byte values. New values every 100 ms. param1 = (int *) pointer to int array[5]
TE1DCLBK_E1SASISABIT* (E1)	Description: Informs the TMS of a new Sa bit value. param1 = (int) Sa4-Sa8 bits
TE1DCLBK_E1SASISIBIT* (E1)	Description: Informs the TMS of a new Si bit value. param1 = (int) SET or CLEAR
TE1DCLBK_E1SASIXBIT* (E1)	Description: Informs the TMS of a new Extra bit value. param1 = (int) 3-bit X-Bit value
TE1DCLBK_E1SASISAMIRROR* (E1)	Description: Informs TMS that mirroring of the Sa values is on/off. param1 = (int) ENABLE or DISABLE

Fcode	Detail
TE1DCLBK_E1SASISADEBOUNCE* (E1)	Description: Asks the TMS to "de-bounce" the Sa bits. param1 = (int) ENABLE or DISABLE
TE1DCLBK_E1SASISIDEBOUNCE* (E1)	Description: Asks the TMS to "de-bounce" the Sa bits param1 = (int) ENABLE or DISABLE
TE1DCLBK_E1SASIXDEBOUNCE* (E1)	Description: Asks the TMS to "de-bounce" the Sa bits param1 = (int) ENABLE or DISABLE

* Valid only for *Maintenance Manager* module.
** Valid only for *Signaling Manager* module.
(T1) – Valid only for a T1 Trunk
(E1) – Valid only for an E1 Trunk

Signaling Manager Module Fcodes

The following Fcodes are only available if the Signaling Manager Module is present.

Fcode	Detail
TE1DCLBK_SIGNLDEBOUNCE**	Description: Tells the TMS whether or the driver will handle "de-bouncing" of signaling bits. ENABLE means that TMS needs to handle this. param1 = (int) ENABLE or DISABLE

** Valid only for *Signaling Manager* module.

TR008 Manager Module Fcodes

The following Fcodes are only available if the TR008 Manager Module is present.

Fcode	Parameters
TE1DCLBK_CMASRX***	Description: Informs the TMS software that it has received new CMAS bits param1 = (unsigned long) 24-bits param2 = (int) 0 or 1 - mirrored bits
TE1DCLBK_CMASTX***	Description: Retrieves transmit CMAS bits. param1 = (unsigned long *) pointer to bits param2 = (int mirror) of the 24-bits

* Valid only for *Maintenance Manager* module.
** Valid only for *Signaling Manager* module.
*** Valid only for *TR008 Manager* module.
(T1) – Valid only for a T1 Trunk
(E1) – Valid only for an E1 Trunk

Return Value

If the function call:

is supported/successful: returns a `+1`.
is not supported: the function returns `ERR_NOSUPPORT (0).`
errors: returns one of the predefined negative numbers in `apiTE1Dv.h` for the appropriate error condition.

Note that failure to support all predefined function codes may result in non-deterministic span behavior.

Example

This is an example of a call back that the TMS package would make to the application. If TMS has just started receiving AIS on span 0 – the first span, then this call would be made.

```
int retval;

retval = _te1DrvCLBK(lineptr, TE1DCLBK_ALARMRX, TMS_RXAIS, SET);
```

The _te1DrvPOLL Function

Description

The `_te1DrvPOLL()` function permits the TMS to asynchronously poll the framer driver for various instantaneous status information. The framer driver must implement this function and accommodate the predefined function codes given in the `apiTE1Dv.h` header file.

This call must *not* block.

The Function and Arguments

The `_te1DrvPOLL` function takes the following form:

```
#include "apiTE1Dv.h"
extern int _te3DrvPOLL (void *drvref, TE3DPOLL_FC fcode…);
```

Where:

drvref	A void * pointer to the drivers internal storage containing line related information. This pointer is returned to TMS from the driver during the TE1DCTRL_RESET call.
fcode	The function code that identifies the action to be performed by the framer circuit. This is typically one of several predefined values as specified in the `apiTE1Dv.h` header file. Each function code may have zero or more parameters. These parameters communicate further information to the function code. Refer to the following table for fcode definitions and other parameter values.
…	Depending upon the **fcode**, more arguments may be required. See the specific **fcode** for further details.

The Function Code and Other Parameter Values

If needed, other parameters as specified by the fcode definitions.

Fcode	Detail
TE1DPOLL_DEVICE	Description: Similar to the TE1DCTRL_DEVICE fcode. Uses user-defined parameters. The user can, via a call to TE1LPOLL_DEVICE; gain access to the driver without making direct driver calls, thus maintaining architecture integrity in the software. This code bypasses the TMS. param1 = unsigned long user-defined param2 = unsigned long user-defined param3 = unsigned long user-defined param4 = unsigned long user-defined
TE1DPOLL_ALARMRX	Description: Returns the current state of the Receive Alarm condition. param1 = (int) ALARM_TYPE param1 = (int*) load with SET or CLEAR

Maintenance Manager Module Fcodes

The following Fcodes are only available if the Maintenance Manager Module is present.

Fcode	Detail
TE1DPOLL_LOOP* (T1)	Description: Returns the current state of reception of loopback code. param1 = (int) LOOP_TYPE param1 = (int*) load with SET or CLEAR
TE1DPOLL_PERFINFO*	Description: Returns the various performance parameters by filling out the appropriate fields in the TE1_T1PERFINFO or TE1_E1PERFINFO data structures. param1 = (void *) pointer to TE1_T1PERFINFO (for T1 trunks) or TE1_E1PERFINFO (for E1 trunks)
TE1DPOLL_E1SASISABYTE* (E1)	Description: Returns the Sa byte for CRC framing. param1 = (int *) load with 5-bit SaBits

Fcode	Detail
TE1DPOLL_E1SASISABIT* (E1)	Description: Returns the 5-bit Sa bits. param1 = (int *) load with 5-bit SaBits
TE1DPOLL_E1SASISIBIT* (E1)	Description: Returns the Si bit detected on the receive side. param1 = (int *) SET or CLEAR
TE1DPOLL_E1SASIXBIT* (E1)	Description: Returns the Extra bits detected on the receive side. param1 = (int *) load with 3-bit Xbits

* Valid only if *Maintenance Manager* module is present.
** Valid only for *Signaling Manager* module.
(T1) – Valid only for a T1 Trunk
(E1) – Valid only for an E1 Trunk

Signaling Manager Module Fcodes

The following Fcodes are only available if the Signaling Manager Module is present.

Fcode	Detail
TE1DPOLL_SIGNLALLBITS**	Returns all timeslot signal bits in an array. param1 = (int*) array of bits (32)

Note: In the DRVR API, Timeslot values range from 0 to 23 for T1 and 0 to 31 for E1.
* Valid only if *Maintenance Manager* module is present.
** Valid only for *Signaling Manager* module.
(T1) – Valid only for a T1 Trunk
(E1) – Valid only for an E1 Trunk

Return Value

If the function call:

is supported/successful: returns a `+1`.
is not supported: the function returns `ERR_NOSUPPORT (0)`.
errors: returns one of the predefined negative numbers. See the file `apiTE1Dv.h` for a list of error conditions.

Note that failure to support all predefined function codes may result in non-deterministic span behavior.

Example

This is an example of a poll call that would be made to driver to find the status of something. In this case, it is to see whether TMS is currently receiving AIS on span 0 – the first span.

```
int retval, ais_status;

retval = _te1DrvPOLL(lineptr, TE1DPOLL_ALARMRX, TMS_RXAIS,
&ais_status);
if ( ais_status == SET)
     {
          .
          .
          .
     }
```

C Definitions Used

The following is a list of typedefs and constants that are used by the Driver Functions. This section is for reference only. Please check the appropriate header files.

ALARM_TYPE

The following is list of alarm types. These are located in the nciTMS.h header file.

```
#define TMS_RXLOS        0x20000001
#define TMS_RXOOF        0x20000002
#define TMS_RXAIS        0x20000004
#define TMS_RXRAI        0x20000008
#define TMS_RXRFI        0x20000010
#define TMS_RXBSF        0x20000020
#define TMS_RXBSD        0x20000040
#define TMS_RXLOP        0x20000080
#define TMS_RXTIM        0x20000100
#define TMS_RXUNEQ       0x20000200
#define TMS_RXPLM        0x20000400
#define TMS_RXRFIS       0x20000800
#define TMS_RXRFIC       0x20001000
#define TMS_RXRFIP       0x20002000
#define TMS_RXIDLE       0x20004000
#define TMS_RXOOFDL      0x20008000     /* TR008 alarm */
#define TMS_RXRED        0x20010000
#define TMS_TXRAI        0x40000001
#define TMS_TXAIS        0x40000002
#define TMS_TXRFI        0x40000004
#define TMS_TXRFIS       0x40000008
#define TMS_TXRFIC       0x40000010
#define TMS_TXRFIP       0x40000020
#define TMS_TXIDLE       0x40000040
```

BOC Codes

```
/* 'function' codes for BOC registration */
#define BOC_DEFAULT           0x00 /* do T1.403 default action */
#define BOC_PAYLOADLOOP_ON    0x01
#define BOC_PAYLOADLOOP_OFF   0x02
#define BOC_LINELOOP_ON       0x04
#define BOC_LINELOOP_OFF      0x08
#define BOC_UNIVERSAL_OFF     0x10
```

CLOCK_TYPE

```
typedef enum {
```

```
     LOCALOSC = 100,     /* Select a local oscillator */
     RXCLOCK,            /* Select the transmit clock */
     SYSCLOCK,           /* Select the system clock */
     OTHERCLK            /* Select some other clock source */
} CLOCK_TYPE;
```

E1_PERFINFO

```
typedef struct _perfinfoE1 {
     unsigned long CVcount;
     unsigned long CRCcount;
     unsigned long EBITcount;
     unsigned long FAScount;
} E1_PERFINFO;
```

LOOP_TYPE

The following is list of loop types. These are located in the nciTMS.h header file.

```
#define LOOP_CODE            0x40000000       /* don't touch the sign-bit */
#define LOOP_SWITCH          0x20000000       /*  onoff  codeSet */
#define LOOP_REMSWITCH       0x20000001       /*    X            */
#define LOOP_LOCSWITCH       0x20000002       /*    X            */
#define LOOP_PAYSWITCH       0x20000004       /*    X            */
#define LOOP_FRMSWITCH       0x20000008       /*    X            */
#define LOOP_REMTXUPCODE     0x40000010       /*    X       X    */
#define LOOP_REMTXDNCODE     0x40000020       /*    X       X    */
#define LOOP_REMRXUPCODE     0x40000040       /*            X    */
#define LOOP_REMRXDNCODE     0x40000080       /*            X    */
```

T1_PERFINFO

```
typedef struct _perfinfoT1 {
      unsigned long FEcount;        /* framing error count */
      unsigned long CVcount;        /* coding violations + EXzeros */
      unsigned long CRCcount;       /* crc error count */
      unsigned long EBcount;        /* errored block count */
      unsigned  int SLevent;        /* Rx slip event SET or CLEAR */
      unsigned  int SEFEevent;      /* severely errored framing event
                                     * SET or CLEAR */
} T1_PERFINFO;
```

TE1_CODING

```
typedef enum {
LC_JBZS=1,      /* Japanese line coding */
LC_B8ZS=2,      /* B8ZS line coding */
LC_HDB3=3,      /* HDB3 line coding */
LC_ZBTSI=4,     /* ZBTSI line coding
LC_AMI=5,       /* AMI line coding */
LC_OTHER=6      /* Any other line coding */
```

```
} TE1_CODING;  /* define the line coding */
```

TE1_CONFIG

```
typedef struct {
     TE1_TYPE  t1e1;
     TE1_FORMATframing;
     TE1_CODINGcoding;
     int  lbo; /* this one's a raw signed value */
          /* the lbo value is in feet (short haul) */
          /* or one of the following values */
          /* -1 being 0db long haul */
          /* -2 being -7.5db long haul */
          /* -3 being -15db long haul */
          /* -4 being -22.5db long haul */
} TE1_CONFIG;
```

TE1_FORMAT

```
typedef enum {
FF_OTHER=1,     /* other frame format */
FF_ESF=2,       /* extended superframe for T1 */
FF_D4=3,        /* superframe for T1 */
FF_E1=4,        /* E1 framing format */
FF_E1_CRC=5,    /* E1 framing format with CRC4 */
FF_E1_MF=6,     /* E1 multi-frame format */
FF_E1_CRC_MF=7,/* E1 multi-frame format with CRC4 */
FF_SLC96=8,     /* SLC-96 frame format for T1/E1*/
} TE1_FORMAT;  /* define the frame format */
```

Appendix A: Supported Devices

The following is a list of devices supported by the device driver:

DS21455 T1 and E1
DS21458 T1 and E1
DS2155 T1 and E1
DS21X52 T1
DS21X54 E1
DS21QX52 T1
DS21QX54 E1
DS2151 T1
DS2152 T1
DS2153 E1
DS21Q55 T1 and E1
DS26528 T1 and E1
DS26502 T1 and E1
DS26503 T1 and E1

This page has been intentionally left blank.