

TLP224e

A TLP224 extended implementation for PalmOS
Version 0.2, 27 June 2001

Copyright © 2001 Xiring, École Nationale Supérieure des Télécommunications.

Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this manual under the conditions for verbatim copying, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this manual into another language, under the above conditions for modified versions, except that this permission notice may be stated in a translation approved by the Free Software Foundation.

Table of Contents

1	Constants	1
	enum TlpErr	1
	enum TlpKbdFlags	2
	enum TlpMode	2
	enum TlpSpeed	3
	Miscellaneous tlp macros	4
2	Functions	5
	TlpClose	5
	TlpConfigureLine	5
	TlpConfigureLinePalm	6
	TlpControl	6
	TlpDisplayText	7
	TlpGetLastError	7
	TlpIccIncomingCommand	7
	TlpIccIncomingOutgoingCommand	8
	TlpIccOutgoingCommand	9
	TlpIccPowerOff	10
	TlpIccPowerOn	11
	TlpIccStatus	12
	TlpIccWarmReset	12
	TlpKeyboardInput	13
	TlpOpen	14
	Index	17

1 Constants

enum TlpErr

DESCRIPTION

These are the error codes returned by most of the TLP224e functions.

VALUES

tlpSuccess

Successfull operation.

tlpErrWrongParameter

One or more parameters have a wrong value.

tlpErrUnknownReaderCommand

Unknow reader command (not a valid command code).

tlpErrUnsupportedSpeed

Line speed or mode not supported.

tlpErrResetUnsupportedProtocol

ICC interface or protocol not supported (warning on ATR).

tlpErrIccUnsupportedProtocol

ICC protocol not supported (ICC command rejected).

tlpErrResetMute

No ATR received (the card does not respond).

tlpErrResetParityError

ATR with parity error (after several attempts to retransmit).

tlpErrResetIncorectTck

ATR with incorect TCK (FIXME: what is a TCK? Time Clock?).

tlpErrIccMute

ICC command not answered.

tlpErrIccParityError

ICC command with parity error (maybe ICC is off).

tlpErrIccRemoved

ICC removed (the card was withdrawn while carrying out the command).

tlpErrIccAbsent

ICC not present in the terminal.

tlpErrIccNok

Warning on an ICC command (SW is not 0x9000).

tlpErrKbdAnnul

Current keyboard input terminated by ANNUL hit.

tlpErrKbdTimeOut

Current keyboard input terminated by a timeout.

tlpErrKbdValid

Current keyboard input terminated by VALID hit.

tlpErrPalmOS

Error from the Palm host (such as a serial or memory error). Call **TlpGetLastError** to get the last Palm error code.

tlpErrConversation

Conversation error between the host and the terminal.

enum TlpKbdFlags

DESCRIPTION

These are bit flags for **TlpKeyboardInput** that you can or together. The default mode (0) is the transparent mode: every key is accepted as is, without further interpretation, and the input is echoed to the terminal.

VALUES

tlpKbdCorrClear

Turn on interpretation of the CORR key: when pressed the last input key is cleared.

tlpKbdAnnulTerminate

Turn on interpretation of the ANNUL key: when pressed the input is interrupted and send back with status **tlpErrKbdAnnul**.

tlpKbdAnnulClear

Turn on interpretation of the ANNUL key: when pressed the whole input is cleared.

tlpKbdValidTerminate

Turn on interpretation of the VALID key: when pressed input is terminated and send back with status **tlpErrKbdValid** unless the exact number of characters have been entered (in which case status **tlpSuccess** is returned).

tlpKbdNoEcho

Turn off echoing.

tlpKbdEchoStars

Echo stars instead of the real characters.

enum TlpMode

DESCRIPTION

These values specify how bytes are send over the line.

VALUES

tlpAsciiMode

Each byte is send as two ascii values (for instance 0x12 is sent as '1' and '2').

tlpBinaryMode

Each byte is sent as-is.

enum TlpSpeed

DESCRIPTION

These constants are used to indicate the speed in baud to use on the line which connect the palm to the reader device. The default speed (used by the reader on reset, and by this library after `TlpOpen`) is 9600 bauds.

VALUES

`tlp1200Bauds`

These constants are used to indicate the speed in baud to use on the line which connect the palm to the reader device. The default speed (used by the reader on reset, and by this library after `TlpOpen`) is 9600 bauds.

`tlp2400Bauds`

These constants are used to indicate the speed in baud to use on the line which connect the palm to the reader device. The default speed (used by the reader on reset, and by this library after `TlpOpen`) is 9600 bauds.

`tlp4800Bauds`

These constants are used to indicate the speed in baud to use on the line which connect the palm to the reader device. The default speed (used by the reader on reset, and by this library after `TlpOpen`) is 9600 bauds.

`tlp9600Bauds`

These constants are used to indicate the speed in baud to use on the line which connect the palm to the reader device. The default speed (used by the reader on reset, and by this library after `TlpOpen`) is 9600 bauds.

`tlp19200Bauds`

These constants are used to indicate the speed in baud to use on the line which connect the palm to the reader device. The default speed (used by the reader on reset, and by this library after `TlpOpen`) is 9600 bauds.

`tlp38400Bauds`

These constants are used to indicate the speed in baud to use on the line which connect the palm to the reader device. The default speed (used by the reader on reset, and by this library after `TlpOpen`) is 9600 bauds.

`tlp57600Bauds`

These constants are used to indicate the speed in baud to use on the line which connect the palm to the reader device. The default speed (used by the reader on reset, and by this library after `TlpOpen`) is 9600 bauds.

`tlp115200Bauds`

These constants are used to indicate the speed in baud to use on the line which connect the palm to the reader device. The default speed (used by the reader on reset, and by this library after `TlpOpen`) is 9600 bauds.

Miscellaneous tlp macros

`tlpLibCreator`

The creator of this library.

`tlpLibType`

The type of this library.

`tlpLibName`

The library name of the library.

2 Functions

TlpClose

PURPOSE Close the library.

PROTOTYPE

Err TlpClose (UInt16 *refNum*, UInt16 **refs*)

Trap

PARAMETERS

<i>refs</i>	OUT	number of references to this library
-------------	-----	--------------------------------------

DESCRIPTION

Close the library (disconnect and free global data). The library indicates it is still used by other applications by setting *refs* to a non zero value. If *ref* is zero, then you should unload the library by calling **SysLibRemove**.

TlpClose returns a PalmOS error code if it fails to disconnect or free the global data. This is not expected to ever happen; if it does the library is left in a unknown state.

SEE ALSO [\[TlpOpen\]](#), page 14.

EXAMPLE

```

    UInt16 refs;
    ...
    err = TlpClose (tlpRef, &refs);
    if (err)
        ... handle error ...
    if (refs == 0)
        SysLibRemove (tlpRef);

```

TlpConfigureLine

PURPOSE Select line speed and mode.

PROTOTYPE

TlpErr TlpConfigureLine (UInt16 *refNum*, TlpSpeed *speed*,
TlpMode *mode*)

Trap

PARAMETERS

<i>speed</i>	IN	the new speed to use
<i>mode</i>	IN	the new mode to use

DESCRIPTION

If the Palm can handle the rate specified with *speed*, then the reader is told to change its line speed and mode. If the command to the reader is succesfull, then the Palm side is reconfigured too.

The default mode used by this library is ASCII at 9600 bauds.

RETURN CODES

tlpSuccess

Successfull operation.

tlpErrUnknownReaderCommand

Unknow reader command (not a valid command code).

t1pErrUnsupportedSpeed

Line speed or mode not supported.

t1pErrPalmOS

Error from the Palm host (such as a serial or memory error). Call **T1pGetLastError** to get the last Palm error code.

t1pErrConversation

Conversation error between the host and the terminal.

SEE ALSO [\[T1pSpeed\]](#), page 3,
[\[T1pMode\]](#), page 2,
[\[T1pConfigureLinePalm\]](#), page 6,
[\[T1pOpen\]](#), page 14.

T1pConfigureLinePalm

PURPOSE Select line speed and mode.

PROTOTYPE

Err T1pConfigureLinePalm (UInt16 *refNum*, T1pSpeed *speed*, T1pMode *mode*) Trap

PARAMETERS

<i>speed</i>	IN	the new speed to use
<i>mode</i>	IN	the new mode to use

DESCRIPTION

Change the line speed and mode on the Palm side, without sending anything to the reader. You typically use this functions when the default mode and speed set up by **T1pOpen** is not the default mode and speed of the reader device.

The default mode used by this library is ASCII at 9600 bauds.

Note that this function return a PalmOS Error code, not a T1pError code.

SEE ALSO [\[T1pSpeed\]](#), page 3,
[\[T1pMode\]](#), page 2,
[\[T1pConfigureLine\]](#), page 5,
[\[T1pOpen\]](#), page 14.

T1pControl

PURPOSE Direct connection to the reader.

PROTOTYPE

T1pErr T1pControl (UInt16 *refNum*, UInt8 *command*, const UInt8 **commandData*, UInt16 *commandDataLength*, UInt8 **responseData*, UInt16 **responseDataLength*, UInt16 *responseTimeOut*) Trap

PARAMETERS

<i>command</i>	IN	command code to send
<i>commandData</i>	IN	data to send
<i>commandDataLength</i>	IN	size of <i>commandData</i>
<i>responseData</i>	OUT	data received
<i>responseDataLength</i>	IN OUT	size of <i>responseData</i>
<i>responseTimeOut</i>	IN	response timeout in seconds

DESCRIPTION

Send a *command* and *commandData* to the reader, and fill *responseData* with a received response up to *responseDataLength* bytes. On exit *responseDataLength* is set to the actual number of bytes read. Both *commandData* and *responseData* can be NULL.

This function allow to use reader features not implemented in this library. The return value can be any `TlpErr` code. If no response is received within *responseTimeOut* seconds, the PalmOS serial functions will return a timeout error and thus this library will return `tlpErrPalmOS`. Use value 0 to wait infinitely.

SEE ALSO [\[TlpErr\], page 1](#).

TlpDisplayText

PURPOSE Display text

PROTOTYPE

`TlpErr TlpDisplayText (UInt16 refNum, const Char *text)` Trap

PARAMETERS

text IN the text to display

DESCRIPTION

Print the zero-terminated string *text* on the reader's display.

RETURN CODES

`tlpSuccess`

Successfull operation.

`tlpErrPalmOS`

Error from the Palm host (such as a serial or memory error). Call `TlpGetLastError` to get the last Palm error code.

`tlpErrConversation`

Conversation error between the host and the terminal.

TlpGetLastError

PURPOSE Return the last PalmOS error code.

PROTOTYPE

`Err TlpGetLastError (UInt16 refNum)` Trap

DESCRIPTION

When a function return `tlpErrPalmOS` to indicate that there was an error when calling any Palm system function, you can use this function to get the PalmOS code associated to this failure.

TlpIccIncomingCommand

PURPOSE Send an APDU with incomming data.

PROTOTYPE

`TlpErr TlpIccIncomingCommand (UInt16 refNum, const UInt8 *commandData, UInt8 *response, UInt16 *responseLength)` Trap

PARAMETERS

<i>commandData</i>	IN		the APDU to send
<i>response</i>		OUT	the command result
<i>responseLength</i>	IN	OUT	the size of <i>response</i>

DESCRIPTION

Send a case-3 APDU to the ICC, i.e., an APDU with command data (a.k.a. incoming data) but no response data.

commandData is a raw APDU (CLA, INS, P1, P2, Length [,data...]). whose length is determined by looking at the fifth byte.

response is filled with the card response up to *responseLength* bytes, and *responseLength* is adjusted to the number of bytes written. This response is expected to be 2 bytes wide since it only contains SW1 and SW2.

response can be NULL to discard the ICC response.

RETURN CODES

tlpSuccess	Successful operation.
tlpErrUnknownReaderCommand	Unknown reader command (not a valid command code).
tlpErrIccUnsupportedProtocol	ICC protocol not supported (ICC command rejected).
tlpErrResetUnsupportedProtocol	ICC interface or protocol not supported (warning on ATR).
tlpErrIccParityError	ICC command with parity error (maybe ICC is off).
tlpErrIccRemoved	ICC removed (the card was withdrawn while carrying out the command).
tlpErrIccAbsent	ICC not present in the terminal.
tlpErrIccNok	Warning on an ICC command (SW is not 0x9000).
tlpErrPalmOS	Error from the Palm host (such as a serial or memory error). Call TlpGetLastError to get the last Palm error code.
tlpErrConversation	Conversation error between the host and the terminal.

SEE ALSO [\[TlpIccOutgoingCommand\]](#), page 9,
[\[TlpIccIncomingOutgoingCommand\]](#), page 8.

TlpIccIncomingOutgoingCommand

PURPOSE Send an APDU with incoming data.

PROTOTYPE

TlpErr	TlpIccIncomingOutgoingCommand (UInt16 <i>refNum</i> ,	Trap
	const UInt8 * <i>commandData</i> , UInt16 <i>commandLength</i> ,	
	UInt8 * <i>response</i> , UInt16 * <i>responseLength</i>)	

PARAMETERS

<i>commandData</i>	IN	the APDU to send
<i>commandLength</i>	IN	the length of <i>commandData</i>
<i>response</i>	OUT	the command result
<i>responseLength</i>	IN OUT	the size of <i>response</i>

DESCRIPTION

Send a case-4 APDU to the ICC, i.e., an APDU with command data (a.k.a. ingoing data) expecting response data (a.k.a. outgoing data). This is the general APDU case.

commandData is a raw APDU (CLA, INS, P1, P2 [, Length [, data... [, ExpectedLength]]]) of length *commandLength*.

response is filled with the card response up to *responseLength* bytes, and *responseLength* is adjusted to the number of bytes written. This response is expected to be ExpectedLength+2 bytes wide because SW1 and SW2 are also copied.

RETURN CODES

tlpSuccess	Successfull operation.
tlpErrUnknownReaderCommand	Unknow reader command (not a valid command code).
tlpErrIccUnsupportedProtocol	ICC protocol not supported (ICC command rejected).
tlpErrResetUnsupportedProtocol	ICC interface or protocol not supported (warning on ATR).
tlpErrIccParityError	ICC command with parity error (maybe ICC is off).
tlpErrIccRemoved	ICC removed (the card was withdrawn while carrying out the command).
tlpErrIccAbsent	ICC not present in the terminal.
tlpErrIccNok	Warning on an ICC command (SW is not 0x9000).
tlpErrPalmOS	Error from the Palm host (such as a serial or memory error). Call TlpGetLastError to get the last Palm error code.
tlpErrConversation	Conversation error between the host and the terminal.

SEE ALSO [\[TlpIccIncomingCommand\]](#), page 7,
[\[TlpIccOutgoingCommand\]](#), page 9.

TlpIccOutgoingCommand

PURPOSE Send an APDU with incomming data.

PROTOTYPE

TlpErr	TlpIccOutgoingCommand (UInt16 <i>refNum</i> ,	Trap
	const UInt8 * <i>commandData</i> , UInt8 * <i>response</i> ,	
	UInt16 * <i>responseLength</i>)	

PARAMETERS

<i>commandData</i>	IN		the APDU to send
<i>response</i>		OUT	the command result
<i>responseLength</i>	IN	OUT	the size of <i>response</i>

DESCRIPTION

Send a case-2 APDU to the ICC, i.e., an APDU without command data but expecting response data (a.k.a. outgoing data).

commandData is a raw APDU (CLA, INS, P1, P2, ExpectedLength) whose length is always 5.

response is filled with the card response up to *responseLength* bytes, and *responseLength* is adjusted to the number of bytes written. This response is expected to be ExpectedLength+2 bytes wide because SW1 and SW2 are also copied.

RETURN CODES

tlpSuccess	Successfull operation.
tlpErrUnknownReaderCommand	Unknow reader command (not a valid command code).
tlpErrIccUnsupportedProtocol	ICC protocol not supported (ICC command rejected).
tlpErrResetUnsupportedProtocol	ICC interface or protocol not supported (warning on ATR).
tlpErrIccParityError	ICC command with parity error (maybe ICC is off).
tlpErrIccRemoved	ICC removed (the card was withdrawn while carrying out the command).
tlpErrIccAbsent	ICC not present in the terminal.
tlpErrIccNok	Warning on an ICC command (SW is not 0x9000).
tlpErrPalmOS	Error from the Palm host (such as a serial or memory error). Call TlpGetLastError to get the last Palm error code.
tlpErrConversation	Conversation error between the host and the terminal.

SEE ALSO [\[TlpIccIncomingCommand\]](#), page 7,
[\[TlpIccIncomingOutgoingCommand\]](#), page 8.

TlpIccPowerOff

PURPOSE Power the card off.

PROTOTYPE

TlpErr	TlpIccPowerOff (UInt16 <i>refNum</i>)	Trap
---------------	---	------

DESCRIPTION

Power off the card, if it is present.

RETURN CODES

tlpSuccess	Successfull operation.
-------------------	------------------------

<code>tlpErrUnknownReaderCommand</code>	Unknow reader command (not a valid command code).
<code>tlpErrIccRemoved</code>	ICC removed (the card was withdrawn while carrying out the command).
<code>tlpErrIccAbsent</code>	ICC not present in the terminal.
<code>tlpErrPalmOS</code>	Error from the Palm host (such as a serial or memory error). Call <code>TlpGetLastError</code> to get the last Palm error code.
<code>tlpErrConversation</code>	Conversation error between the host and the terminal.

SEE ALSO [\[TlpIccPowerOn\], page 11.](#)

TlpIccPowerOn

PURPOSE Wait for an ICC and power it on.

PROTOTYPE

`TlpErr TlpIccPowerOn (UInt16 refNum, UInt8 timeOut, UInt8 *controlerType, UInt8 *iccType, UInt8 *atr, UInt16 *atrLength)` Trap

PARAMETERS

<i>timeOut</i>	IN	maximum insertion card wait time
<i>controlerType</i>	OUT	controler type
<i>iccType</i>	OUT	card type
<i>atr</i>	OUT	the card ATR string
<i>atrLength</i>	IN OUT	the <i>atr</i> length

DESCRIPTION

Wait for a card for at most *timeOut* seconds (0-255), then power the card. An ICC cold reset is ensured by a prior power off.

atr receive the ATR string sent by the ICC upon reset, up to *atrLength* bytes. *atrLength* is then set to the exact number of bytes written to *atr*.

Note that the ATR string includes the trailing status bytes (usually 0x90 and 0x00) returned by the card.

RETURN CODES

<code>tlpSuccess</code>	Successfull operation.
<code>tlpErrUnknownReaderCommand</code>	Unknow reader command (not a valid command code).
<code>tlpErrResetMute</code>	No ATR received (the card does not respond).
<code>tlpErrResetParityError</code>	ATR with parity error (after several attempts to retransmit).
<code>tlpErrResetIncorectTck</code>	ATR with incorect TCK (FIXME: what is a TCK? Time Clock?).

tlpErrIccRemoved
ICC removed (the card was withdrawn while carrying out the command).

tlpErrIccAbsent
ICC not present in the terminal.

tlpErrPalmOS
Error from the Palm host (such as a serial or memory error). Call **TlpGetLastError** to get the last Palm error code.

tlpErrConversation
Conversation error between the host and the terminal.

SEE ALSO [\[TlpIccWarmReset\]](#), page 12,
[\[TlpIccPowerOff\]](#), page 10.

TlpIccStatus

PURPOSE Return the card status.

PROTOTYPE

TlpErr TlpIccStatus (UInt16 *refNum*) Trap

DESCRIPTION

Return **TlpSuccess** if a card is present.

RETURN CODES

tlpSuccess
Successful operation.

tlpErrUnknownReaderCommand
Unknown reader command (not a valid command code).

tlpErrIccRemoved
ICC removed (the card was withdrawn while carrying out the command).

tlpErrIccAbsent
ICC not present in the terminal.

tlpErrPalmOS
Error from the Palm host (such as a serial or memory error). Call **TlpGetLastError** to get the last Palm error code.

tlpErrConversation
Conversation error between the host and the terminal.

TlpIccWarmReset

PURPOSE Warm reset of the ICC.

PROTOTYPE

TlpErr TlpIccWarmReset (UInt16 *refNum*,
 UInt8 **controlerType*, UInt8 **iccType*, UInt8 **atr*,
 UInt16 **atrLength*) Trap

PARAMETERS

<i>controlerType</i>	OUT	controler type
<i>iccType</i>	OUT	card type
<i>atr</i>	OUT	the card ATR string
<i>atrLength</i>	IN OUT	the <i>atr</i> length

DESCRIPTION

Warm reset the ICC, which must have been powered on.

atr receive the ATR string sent by the ICC upon reset, up to *atrLength* bytes. *atrLength* is then set to the exact number of bytes written to *atr*.

Note that the ATR string includes the trailing status bytes (usually 0x90 and 0x00) returned by the card.

RETURN CODES

tlpSuccess

Successfull operation.

tlpErrUnknownReaderCommand

Unknow reader command (not a valid command code).

tlpErrResetMute

No ATR received (the card does not respond).

tlpErrResetParityError

ATR with parity error (after several attempts to retransmit).

tlpErrResetIncorectTck

ATR with incorect TCK (FIXME: what is a TCK? Time Clock?).

tlpErrIccRemoved

ICC removed (the card was withdrawn while carrying out the command).

tlpErrIccAbsent

ICC not present in the terminal.

tlpErrPalmOS

Error from the Palm host (such as a serial or memory error). Call **TlpGetLastError** to get the last Palm error code.

tlpErrConversation

Conversation error between the host and the terminal.

SEE ALSO [\[TlpIccPowerOn\]](#), page 11.

TlpKeyboardInput

PURPOSE Keyboard detection with timout and input options.

PROTOTYPE

TlpErr TlpKeyboardInput (UInt16 *refNum*, UInt8 **response*, Trap
 UInt8 **responseLength*, TlpKbdFlags *flags*, UInt8 *charTimeOut*,
 UInt16 *totalTimeOut*)

PARAMETERS

<i>response</i>		OUT	the received keys
<i>responseLength</i>	IN	OUT	the size of <i>response</i>
<i>flags</i>	IN		options
<i>charTimeOut</i>	IN		intercharacter timeout
<i>totalTimeOut</i>	IN		global timeout for the processing

DESCRIPTION

Block until *responseLength* character are entered, or the input is ended by VALID, or ANNUL, or *charTimeOut* seconds elapse between two key strokes.

charTimeOut allows to setup an intercharacter timeout between 1s and 255s (0 means 120s, the default timeout value); this timeout is used by the reader which can return `tlpErrKbdTimeOut` if it expires.

totalTimeOut is the number of seconds that the library should wait for a response of the reader. Because the user might type a key sequence like `Ⓛ ⓈⓈⓈⓈ Ⓛ ⓈⓈⓈⓈ` ... infinitely there is no way for the library to compute the maximum wait time. Use value 0 to wait infinitely. This timeout is handled by the PalmOS system, thus this library will return `tlpErrPalmOS` in this event.

flags can be used to specify several options, including whether the input must be echoed or whether the `ⓈⓈⓈⓈ` key is active. 0 is transparent mode: every key is accepted as is (not interpreted) and the input is echoed.

Not that this functions will block (up to *charTimeOut* * **reponseLength* seconds) until the keyboard output is received.

RETURN CODES

`tlpSuccess`

Successfull operation.

`tlpErrUnknownReaderCommand`

Unknow reader command (not a valid command code).

`tlpErrKbdAnnul`

Current keyboard input terminated by `ⓈⓈⓈⓈ` hit.

`tlpErrKbdTimeOut`

Current keyboard input terminated by a timeout.

`tlpErrKbdValid`

Current keyboard input terminated by `ⓈⓈⓈⓈ` hit.

`tlpErrPalmOS`

Error from the Palm host (such as a serial or memory error). Call `TlpGetLastError` to get the last Palm error code.

`tlpErrConversation`

Conversation error between the host and the terminal.

SEE ALSO [\[TlpKbdFlags\]](#), page 2.

TlpOpen

PURPOSE Initialize the library.

PROTOTYPE

Err TlpOpen (UInt16 *refNum*, UInt16 *port*)

Trap

PARAMETERS

port IN the port to open

DESCRIPTION

This functions allocate global data for the library, and establish a connection to the port *port*. Use `serPortCradlePort` to designate the serial port. See the documentation for `SrmOpen` in the *Palm OS Reference Manual* for other values.

`TlpOpen` returns a PalmOS error code if it fails to establish the connection or allocate data. It is safe to call `TlpOpen` again if the previous call failed but not if it was successful (because in the latter case the library intialized its internal state).

This function will return `sysErrRomIncompatible` if a needed feature (such as the new serial manager) is missing.

The default mode used by this library is ASCII at 9600 bauds, if this do not correspond to the reader configuration, reconfigure the line with `TlpConfigureLinePalm`.

SEE ALSO [\[TlpClose\]](#), page 5,
[\[TlpConfigureLinePalm\]](#), page 6.

EXAMPLE

```
UInt16 tlpRef;
Err err;

err = SysLibFind (tlpLibName, &tlpRef);
if (err)
    err = SysLibLoad (tlpLibType, tlpLibCreator, &tlpRef);
if (err)
    ... handle error ...
err = TlpOpen (tlpRef, serPortCradlePort);
if (err)
    ... handle error ...
```


Index

C

constants 1

E

enum TlpErr 1

enum TlpKbdFlags 2

enum TlpMode 2

enum TlpSpeed 3

F

functions 5

T

tlp115200Bauds 3

tlp1200Bauds 3

tlp19200Bauds 3

tlp2400Bauds 3

tlp38400Bauds 3

tlp4800Bauds 3

tlp57600Bauds 3

tlp9600Bauds 3

tlpAsciiMode 2

tlpBinaryMode 2

TlpClose 5

TlpConfigureLine 5

TlpConfigureLinePalm 6

TlpControl 6

TlpDisplayText 7

TlpErr 1

tlpErrConversation 2

tlpErrIccAbsent 1

tlpErrIccMute 1

tlpErrIccNok 1

tlpErrIccParityError 1

tlpErrIccRemoved 1

tlpErrIccUnsupportedProtocol 1

tlpErrKbdAnnul 1

tlpErrKbdTimeOut 1

tlpErrKbdValid 2

tlpErrPalmOS 2

tlpErrResetIncorectTck 1

tlpErrResetMute 1

tlpErrResetParityError 1

tlpErrResetUnsupportedProtocol 1

tlpErrUnknownReaderCommand 1

tlpErrUnsupportedSpeed 1

tlpErrWrongParameter 1

TlpGetLastErr 7

TlpIccIncomingCommand 7

TlpIccIncomingOutgoingCommand 8

TlpIccOutgoingCommand 9

TlpIccPowerOff 10

TlpIccPowerOn 11

TlpIccStatus 12

TlpIccWarmReset 12

tlpKbdAnnulClear 2

tlpKbdAnnulTerminate 2

tlpKbdCorrClear 2

tlpKbdEchoStars 2

TlpKbdFlags 2

tlpKbdNoEcho 2

tlpKbdValidTerminate 2

TlpKeyboardInput 13

tlpLibCreator 4

tlpLibName 4

tlpLibType 4

TlpMode 2

TlpOpen 14

TlpSpeed 3

tlpSuccess 1

