

T1/E1 Application Development Toolkit

(For Windows® and Linux)

Supports both Windows® and Linux OS



Supports Microsoft® VS2005 C/C++ Compiler



Provides C++ Class Library



Obsolete and Internal Functions



Standard C++ Exception Mechanism to Handle Error Conditions



Produces Robust Code



Client Code Cleaner and Easier to Understands



Consistent Error Recovery Mechanisms



Overview

The GL T1/E1 Application Development Toolkit is an application programming interface (API) library for custom T1/E1 software development. GL now provides API for Windows® and Linux Operating Systems. The toolkit consists of:

- C and C++ Header Files and Library Files
- A 32-bit API library implemented as a DLL
- Device Driver
- Analyzer Application
- User's Manuals and Reference Manuals
- Sample Codes

The Application Development Toolkit can be used to develop new applications that use GL Communications T1/E1 cards, for example, carrying T1 or E1 traffic. The toolkit supports Linux and Windows® XP and later operating systems.

For more information on Application Development Toolkit, refer to <http://www.gl.com/toolkit.html>

Function Groups

The API library consists of the following function groups:

- [Initialization and Termination functions](#)
- [Configuration functions](#)
- [Mode](#)
- [Codec functions](#)
- [Port functions](#)
- [Buffer functions](#)
- [Register functions](#)
- [Driver functions](#)
- [Framer functions](#)
- [Bit Error Rate functions](#)
- [Timeslot functions](#)
- [Pattern file functions](#)
- [Stream I/O functions](#)
- [Miscellaneous functions](#)



GL Communications Inc.

818 West Diamond Avenue - Third Floor, Gaithersburg, MD 20878, U.S.A
(Web) <http://www.gl.com/> - (V) +1-301-670-4784 (F) +1-301-670-9187 - (E-Mail) gl-info@gl.com

System Hierarchy for T1/E1 and SDH Cards

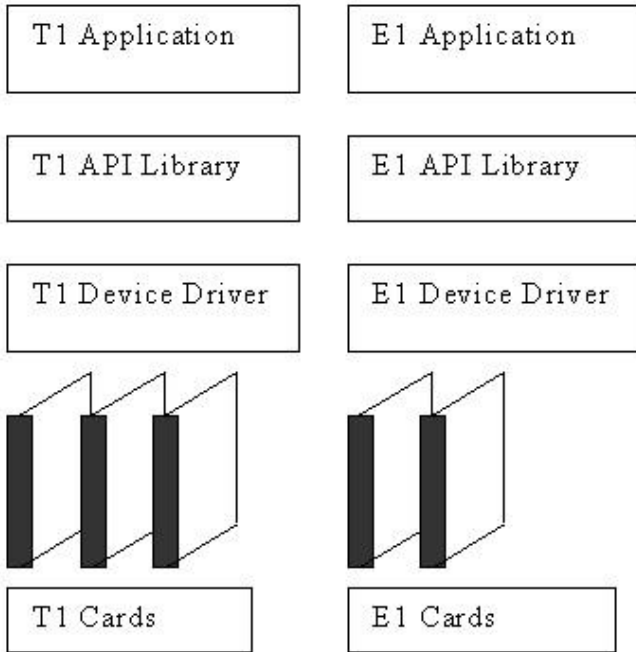


Figure: System Hierarchy for T1/E1 Cards

Initialization and Termination Functions

| Function Name | Description |
|--------------------|--|
| Initialize | Initializes cards |
| CheckCard | Checks card I/O and memory |
| CheckCardIo | Check card I/O |
| CheckCardMemory | Checks card memory |
| LoadDefaultConfig | Loads framer and driver chip registers, |
| ReadHardwareConfig | Reads I/O address, base memory address and IRQ |
| InitializeCodec | Initializes codec |
| ImplementTxIsr | Starts/stops interrupt service for Tx |
| ImplementRxIsr | Starts/stops interrupt service for Rx |

```
try
{
    GlComInterface Ifc;           // Device driver
    interface object
    Ifc.Initialize(true, GL_CARD_TYPE_DPCI_DMA); //
    Must be 1st function called!!!
    ...
}
catch (GlComExceptions except)
...

```

Built-in Debugging Features

The API library has a built in debugging facility. This facility is enabled only in the DEBUG builds of the applications. However, even in the debug builds the facility can be disabled by calling the function `GlcEnableDbgMsg(false)`.

Function Parameters

Provides functions to validate parameters. If a parameter is invalid the appropriate **GlComExceptions** exception is thrown:

- ExceptInvParm1 – the first parameter is invalid
- ExceptInvParm2 – the second parameter is invalid, etc.

```
bRet = Ifc.CheckFr( 0,
    _EfrIsReceiveLossOfSync); // Causes
    ExceptInvParm1
    // because Device number should be
    1..NumberOfInstalledDevices and cannot be 0

```

Mode Functions

| Function Name | Description |
|---------------|---|
| CheckMode | Checks whether a particular mode is set |
| SetMode | Sets a mode |

```
bool b1;
GlComInterface Ifc;           // Device driver
interface object
Ifc.Initialize(true, GL_CARD_TYPE_DPCI_DMA); //
Must be 1st function called!!!

Ifc.SetMode( SelectedDeviceNo, _SetBridgeMode );
b1 = Ifc.CheckMode ( SelectedDeviceNo,
    _IsBridgeModeSelected);
Ifc.SetMode( SelectedDeviceNo, _SetMonitorMode );
b1 = Ifc.CheckMode ( SelectedDeviceNo,
    _IsMonitorModeSelected);
Ifc.SetMode( SelectedDeviceNo,
    _SetTerminateMode );
b1 = Ifc.CheckMode ( SelectedDeviceNo,
    _IsTerminateModeSelected);

```



818 West Diamond Avenue - Third Floor, Gaithersburg, MD 20878, U.S.A
 (Web) <http://www.gl.com/> - (V) +1-301-670-4784 (F) +1-301-670-9187 - (E-Mail) gl-info@gl.com

Configuration Functions

| Function Name | Description |
|----------------------------------|--|
| GetMaxTimeslot | Returns maximum time slot number for the devices |
| GetMaxDeviceCount | Returns maximum number of devices supported by device driver |
| GetNoOfDevicesInstalled | Returns number of devices installed and configured |
| GetIoBaseAddr | Returns base I/O address |
| GetMemOffset | Returns memory offset |
| GetInterrupt1 | Returns interrupt number |
| GetMultiFrameSize | Returns multiframe size |
| GetMultiFrameDuration | Returns multiframe duration |
| GetIdleCode | Returns idle code used for transmission |
| GetNoOfFramesPerMF | Returns number of frames per multiframe |
| GetMultiFrameSize | Returns multiframe size |
| GetMultiFrameDuration | Returns duration of multiframe |
| GetUsableSoftBufferSize | Returns usable software buffer size |
| GetNoOfMultiFramesPerSoft Buffer | Returns number of multiframes fit in software buffer |
| GetProtocolBytesPerMs | Number of bytes transferred per mSec full frame |
| GetNoOfFramesPerSoftBuffer | Returns number of frames fit in software buffer |

```

try
{
    GlComInterface Ifc;          // Device driver
    interface object
    Ifc.Initialize(true, GL_CARD_TYPE_DPCI_DMA); //
    Must be 1st function called!!!
    ...
    printf("Frames in Multiframe = %d",
    Ifc.GetNoOfFramesPerMF(SelectedDeviceNo));
#ifdef T1
    BitsInMltFrm = 193 * Ifc.GetNoOfFramesPerMF();
#else // E1
    BitsInMltFrm = 256 * Ifc.GetNoOfFramesPerMF();
#endif
}
catch (GlComExceptions except)
{... }

```

Codec functions

| Function Name | Description |
|--------------------------|--|
| InitializeCodec | Initializes codec |
| SetCodec | Sets gain, Rx/Tx timeslots, sets clock to recovered, internal or external, sets CASF |
| ResetCodec | Resets to BER |
| GetCodec | Queries gain, timeslot and BER settings |
| EnableCodec | Enables speaker, VF transmit, and drop insert |
| DisableCodec | Disables speaker, VF transmit, and drop insert |
| SetCodecInterfaceLatch | Sets Codec interface latch |
| ResetCodecInterfaceLatch | Resets Codec interface latch |
| CheckCodecTxRxLatch | Checks interface latch status for Tx or Rx |

```

GlComInterface Ifc;          // Device driver
interface object
Ifc.Initialize(false, GL_CARD_TYPE_DPCI_DMA); //
Must be 1st function called!!!
...
/*****
/* Get current settings for the current device
*/

TxTs = Ifc.GetCodec(DevNo, _GetTxTimeslot); //
Read this setting from driver
RxTs = Ifc.GetCodec(DevNo, _GetRxTimeslot); //
Read this setting from driver
RxGain = Ifc.GetCodec(DevNo,
_GetGainForVfRx ); // Read VF gain for Rx
TxGain = Ifc.GetCodec(DevNo,
_GetGainForVfTx ); // and TX
bSpeaker = Ifc.CheckCodec(DevNo, _IsSpeakerOn );
bInsert = Ifc.CheckCodec(DevNo, _IsVfTxOn );

Ifc.DisableCodec(DevNo, _TurnOffSpeaker ); // If
was On set Off
Ifc.EnableCodec(DevNo, _TurnOnSpeaker); // If was
Off set On
Ifc.DisableCodec(DevNo, _TurnOffVfTx ); // If was
On set Off
Ifc.EnableCodec(DevNo, _TurnOnVfTx); // If was
Off set On
Ifc.SetCodec( DevNo, _SetRxTimeslot, (BYTE)
timeslot );
Ifc.SetCodec( DevNo, _SetTxTimeslot, (BYTE)
timeslot );
Ifc.SetCodec( DevNo, _SetGainForVfRx, (BYTE)
gain );
Ifc.SetCodec( DevNo, _SetGainForVfTx, (BYTE)
gain );

```



818 West Diamond Avenue - Third Floor, Gaithersburg, MD 20878, U.S.A
 (Web) <http://www.gl.com/> - (V) +1-301-670-4784 (F) +1-301-670-9187 - (E-Mail) gl-info@gl.com

Try { } Catch { } Blocks

To handle the errors properly, the API library incorporate client application code into try { } catch { } blocks.

```
try
{
  GlComInterface Ifc; // Device driver interface object
  Ifc.Initialize(); // Must be first member function
  called!!!
}
catch (GlComExceptions except)
{
  GlCMsg( "Driver Interface Error", "Exception description:%s",
    GetExceptionDescription( except ) );
  return 11;
}
catch ( DWORD lastError )
{
  GlCMsg( "Driver Error", "GetLastError() = %ul", lastError );
  return 12;
}
catch (...)
{
  printf("Unexpected internal error\n");
  return 13;
}
```

Buyer's guide

- XX010 – Application Development Toolkit for Windows® 7 / 8 (Programmer's Manual)
- XX011 – Application Development Toolkit for Linux

Related Hardware

[PTE001](#) - tProbe™ Dual T1 E1 Laptop Analyzer (Requires Basic Software)

[FTE001](#) - QuadXpress T1E1 Main Board

[ETE001](#) - OctalXpress T1E1 Main Board plus Daughter Board

[HTE001](#) - Universal T1/E1 Card (Requires Basic Software)

[UTE001](#) - Portable USB based Dual T1/E1 Laptop Analyzer (Requires Basic Software)



818 West Diamond Avenue - Third Floor, Gaithersburg, MD 20878, U.S.A
 (Web) <http://www.gl.com/> - (V) +1-301-670-4784 (F) +1-301-670-9187 - (E-Mail) gl-info@gl.com