



www.roteccontrol.com
rotec@roteccontrol.com

C/ Llibertat 29 local b
07013 - Palma de Mallorca
Balears - España
☎ + 34 971 91 39 34
Fax + 34 971 45 73 10

Programación directa

DLL
® *Windows*

***Sistemas de control
Elementos para la automatización
Reguladores de proceso***

Índice de páginas:

Programación directa con DLL para Visual Cube	2
Programación directa con DLL para Easy Basic	3



Para equipos programados en Visual Cube

PROGRAMACIÓN DIRECTA CON LENGUAJE DE ALTO NIVEL:

Entre los programas desarrollados por ROTEC para la comunicación con los equipos de control y las terminales, existe uno que puede ser utilizado por cualquiera que disponga de su propio programa de control de instalaciones. DRIVER TPC es una librería DLL estándar de Windows que funciona sobre cualquier plataforma, Windows XP, 2000, NT.

Como usar el Driver TPC:

El Driver TPC DLL dispone de funciones que pueden ser llamadas desde la aplicación desarrollada en cualquier lenguaje de alto nivel. Al tratarse de una librería de funciones que se incluye directamente sobre el lenguaje fuente, se usa como una llamada de función del propio lenguaje.

Dichas funciones permiten configurar las comunicaciones de la DLL con las terminales que tiene conectadas en el BUS o en el RS232. Una vez configuradas, la propia DLL funcionará de forma autónoma y realizará una exploración de las terminales, capturando sus valores y traspasándolas a la aplicación de alto nivel.

Las declaraciones de las funciones para el lenguaje Delphi son:

Funciones específicas de TIR

```
procedure InicializarListaTIR(TPC: Byte; Canal: Byte); stdcall; external 'DriverTPCdll.dll';
function AnadirTIR(TPC: byte; Canal: byte;Codigo: byte; Placa1: byte; Placa2: byte; Placa3: byte; Placa4: byte; Prioridad: byte): smallint; stdcall; external 'DriverTPCdll.dll';
function PonerValorTIR(TPC: byte; Canal: byte;Codigo: byte; Punto: byte; Valor: single): smallint; stdcall; external 'DriverTPCdll.dll';
function CogerValorTIR(TPC: byte; Canal: byte;Codigo: byte; Punto: byte): single; stdcall; external 'DriverTPCdll.dll';
function MostrarInfo(tpc: Byte; canal: Byte): Smallint; stdcall; external 'DriverTPCdll.dll';
procedure ProcCambiosTIR(proc1: TProcCambioValor; proc2: TProcCambioEstado); stdcall; external 'DriverTPCdll.dll';
function CogerEstadoTIR(TPC: byte; Canal: byte;Codigo: byte): smallint; stdcall; external 'DriverTPCdll.dll';
function CogerResetTIR(TPC: byte; Canal: byte;Codigo: byte): smallint; stdcall; external 'DriverTPCdll.dll';
procedure PonerEstadoTIR(TPC: byte; Canal: byte;Codigo: byte; EnteradoReset: Byte; InhibirSalidas: Byte) stdcall; external 'DriverTPCdll.dll';
procedure PonerConfPunto(TPC: byte; Canal: byte;Codigo: byte; Punto: Byte; Conf: PConfPunto) stdcall; external 'DriverTPCdll.dll';
```

Funciones específicas de RC620-VC

```
function AnadirTRG(TPC: byte; Canal: byte;Codigo: byte): smallint; stdcall; external 'DriverTPCdll.dll';
function MostrarInfoTRG(tpc: byte; canal: byte;Codigo: Byte): Smallint; stdcall; external 'DriverTPCdll.dll';
function VerUsoCanal(tpc: byte; canal: byte): Longint; stdcall; external 'DriverTPCdll.dll';
function CogerDatosHechosTRG(tpc: byte; canal: byte; codigo: byte): Smallint; stdcall; external 'DriverTPCdll.dll';
function QuitarTRG(TPC: byte; Canal: byte;Codigo: byte): smallint; stdcall; external 'DriverTPCdll.dll';
procedure ReiniciarTRG(TPC: byte; Canal: byte;Codigo: byte); stdcall; external 'DriverTPCdll.dll';
function PonerValorTRG(TPC: byte; Canal: byte;Codigo: byte; Tipo: byte; Variable: Word; var Valor: TValorUniv): Smallint; stdcall; external 'DriverTPCdll.dll';
function CogerValorTRG(TPC: byte; Canal: byte;Codigo: byte; Tipo: byte; Variable: Word; var Valor: TValorUniv): Smallint; stdcall; external 'DriverTPCdll.dll';
procedure ProcCambiosTRG(proc1: TProcCambioTRG; proc2: TProcCambioInterno); stdcall; external 'DriverTPCdll.dll';
procedure ModoTestTRG(TPC: byte; Canal: byte;Codigo: byte; test: boolean); stdcall; external 'DriverTPCdll.dll';
function CogerEstadoTRG(tpc: byte; canal: byte; codigo: byte): boolean; stdcall; external 'DriverTPCdll.dll';
```

Funciones generales

```
function EstadoTPC(TPC: Byte): byte; stdcall; external 'DriverTPCdll.dll';
function PedirCommLibre(TPC: Byte; Canal: Byte; BitsTX: Word; BitsRX: Word; Reintentos: Byte; Buffer: PByteArray): smallint; stdcall; external 'DriverTPCdll.dll';
function DevolverCommLibre(var Estado: Smallint; var BitsRX: Word; Buffer: PByteArray): smallint; stdcall; external 'DriverTPCdll.dll';
procedure LibExit; stdcall; external 'DriverTPCdll.dll';
```

```
TProcCambioValor = procedure(TPC: byte; Canal: byte;Codigo: byte; Punto: Byte; Valor: single); stdcall;
TProcCambioEstado = procedure(TPC: byte; Canal: byte;Codigo: byte; Estado: Smallint; Reset: byte); stdcall;
```

Toda la documentación sobre los comandos está disponible dentro del DRIVER TPC.PDF que puede encontrar en nuestra dirección www.roteccontrol.com o bien en el paquete de desarrollo de Visual Cube.



Para equipos programados en Easy Basic

PROGRAMACIÓN DIRECTA CON LENGUAJE DE ALTO NIVEL:

Entre los programas desarrollados por ROTEK para la comunicación con los equipos de control y las terminales, existe uno que puede ser utilizado por cualquiera que disponga de su propio programa de control de instalaciones. DRIVER TPC es una librería DLL standard de Windows que funciona sobre cualquier plataforma, Windows XP, 2000, NT.

Como usar el Driver TPC:

El Driver TPC DLL dispone de funciones que pueden ser llamadas desde la aplicación desarrollada en cualquier lenguaje de alto nivel. Al tratarse de una librería de funciones que se incluye directamente sobre el lenguaje fuente, se usa como una llamada de función del propio lenguaje.

Dichas funciones permiten configurar las comunicaciones de la DLL con las terminales que tiene conectadas en el BUS o en el RS232. Una vez configuradas, la propia DLL funcionará de forma autónoma y realizará una exploración de las terminales, capturando sus valores y traspasándolas a la aplicación de alto nivel.

Las declaraciones de las funciones para el lenguaje Delphi son:

Funciones específicas de TIR

```

procedure ResetTPC(tpc: byte);stdcall;
Function AnadirTIR(Puerto: byte;Codigo: byte;Placa1: byte;Placa2: byte;Placa3: byte;Placa4: byte): smallint; stdcall;
Function CogerEstadoTir(Puerto: byte;Codigo: byte): smallint; stdcall;
procedure PonerEstadoTir(Puerto: byte;Codigo: byte;Presente: boolean;EnteradoReset: Boolean;InhibirSalidas: Boolean); stdcall;
procedure PonerValorTIR(Puerto: byte;Codigo: byte;punto: integer;valor: word); stdcall;
procedure PonerValoresTIRS(Puerto: byte;Codigo: byte;valor1: word;valor2: word;valor3: word;valor4: word;valor5: word;valor6: word;valor7: word;valor8: word);
stdcall;
procedure CogerValoresTIRS(Puerto: byte;Codigo: byte;var valor1: word;var valor2: word;var valor3: word;var valor4: word;var valor5: word;var valor6: word;var
valor7: word;var valor8: word); stdcall;

```

Funciones específicas de R6xx EB

```

function CogerEstadoTRG2(tpc: byte;codigo: byte): boolean;stdcall;
procedure AnadirLecturaTRG2(tpc: byte; canal: byte; codigo: word; bloque: byte; inicio: word; Tamano: word; TipoComunicacion :byte; cadenaModem: string); stdcall;
procedure LeerMemoriaTRG2(tpc: byte; canal: byte; codigo: word; bloque: word; inicio: word; Tamano: word; buffer: PByteArray); stdcall;
procedure EscribirMemoriaTRG2(tpc: byte; canal: byte; codigo: word; bloque: word; inicio: word; Tamano: word; buffer: PByteArray); stdcall;
procedure MostrarInfoTRG2(tpc: byte; canal: byte; Codigo: word); stdcall;
procedure QuitarTrg2(tpc: byte; canal: byte; Codigo: word); stdcall;
Function PedirCommLibre(TPC: Byte; Canal: Byte; BitsTX: Word; BitsRX: Word; Reintentos: Byte; Buffer: PByteArray): smallint; stdcall;
Function DevolverCommLibreRs(TPC: Byte; TamTX: byte; TamRx: byte; Timeout: cardinal; Buffer: PByteArray): smallint; stdcall;
Function DevolverCommLibre(var Estado: Smallint; var BitsRX: Word; Buffer: PByteArray): smallint; stdcall;
procedure PonerMemoriaACeroSifallo(A_Cero: Boolean);stdcall;
procedure DelayEntreTranmisiones(NuevoDelay:integer);stdcall;
Function LeerLongRegulador(tpc: byte;RS: Boolean; codigo: word; bloque: byte; inicio: word; var datoLong: Longint ):Boolean;stdcall;
Function LeerFloatRegulador (tpc: byte;RS: Boolean; codigo: word; bloque: byte; inicio: word; var Dato: single ):Boolean;stdcall;
Function LeerByteRegulador (tpc: byte;RS: Boolean; codigo: word; bloque: byte; inicio: word; var Dato: Byte):Boolean;stdcall;
Function LeerWordRegulador (tpc: byte;RS: Boolean; codigo: word; bloque: byte; inicio: word; var Dato: Word):Boolean;stdcall;
Function EscribirFloatRegulador(tpc: byte;RS: Boolean; codigo: word; bloque: byte; inicio: word; Dato: single ):Boolean;stdcall;
Function EscribirByteRegulador (tpc: byte;RS: Boolean; codigo: word; bloque: byte; inicio: word; Dato: Byte):Boolean;stdcall;
Function EscribirWordRegulador (tpc: byte;RS: Boolean; codigo: word; bloque: byte; inicio: word; Dato: Word):Boolean;stdcall;
Function EscribirLongRegulador (tpc: byte;RS: Boolean; codigo: word; bloque: byte; inicio: word; Dato: Longint):Boolean;stdcall;
Function LeerBloqueMemoriaRegulador (tpc: byte;RS: Boolean; codigo: word; bloque: byte; inicio: word; tamano: word; var buffer : buffDatos):Boolean;stdcall;
Function EscribirBloqueMemoriaRegulador(tpc: byte;RS: Boolean; codigo: word; bloque: byte; inicio: word; tamano: word; var buffer : buffDatos):Boolean;stdcall;
Function LeerBloqueEpromRegulador(tpc: byte;RS: Boolean; codigo: word; bloque: byte; inicio: word; tamano: word; var buffer : buffDatos):Boolean;stdcall;
Function EscribirBloqueEpromRegulador (tpc: byte;RS: Boolean; codigo: word; bloque: byte; inicio: word; tamano: word; var buffer : buffDatos):Boolean;stdcall;

```

Toda la documentación sobre los comandos está disponible dentro del Funciones de la DLL Easy Basic.PDF que puede encontrar en nuestra dirección www.rotecontrol.com o bien en el paquete de desarrollo de EASY BASIC.