Modbus Interface

Users Manual

System Interface

The implemented procedures and functions in the DLL modbus (Modbus.dll) are subsequently described in detail:

OpenCom	Open a serial link communication port.
DisableCom	Close communication port
ReadWords	Read consecutives Words in an Slave.
WriteWords	Write consecutives Words in an Slave
ReadBits	Read consecutives Bits in an Slave.
WriteBits	Write consecutives Bits in an Slave.
TimeOutValue	Set the Timeout value in all communications.

The DLL Modbus.DLL must be located in your program directory or must be accessible by your program. (automatic path to Windows or WinNt)

Caution yours prototypes follow the sensitive case.

Note :
The DLL Modbus.DLL follow the current Modbus norm.

DLL interfacing with certain development tools requires the creation of a library of importation

In C++ Builder: This library of importation (lib File) is normally created automatically if not, the Implib.exe order under Dos makes it possible to create it

In Visual C++: This library of importation is normally created automatically if not, the command lib.exe under Dos makes it possible to create it. The interfacing with the DLL requires also the writing of a module of declaration (def File to see following example).

```
LIBRARY modbus
        DESCRIPTION ‘’MODBUS DLL for Win32’’
        EXPORTS
                OpenCom@20      =OpenCom @1
                DisableCom@4    =DisableCom @2
                ReadWords@20    =ReadWords @3
                WriteWords@20   =WriteWords @4
                ReadBits@20     =ReadBits @5
                WriteBits@20    =WriteBits @6
                TimeOutValue@4  =TimeOutValue @7
```

Implementation in Pascal (Delphi or Visual Pascal)

```
//**** External in DLL Modbus.dll *********
Function   OpenCom (ComPort,BaudRate: Integer; Parity : Char; Bits,Stop : Integer ): Boolean; StdCall; External
'ModBus.Dll';

Procedure  DisableCom(ComPort: Integer); StdCall; External 'ModBus.Dll';

Function   ReadWords( Var Data; Slave,Address,Words : Integer; ComPort : Integer) :
Boolean; StdCall;  External 'ModBus.Dll';

Function   WriteWords( Var Data; Slave,Address,Words : Integer; ComPort : Integer) : Boolean; StdCall; External
'ModBus.Dll';

Function   ReadBits( Var Data; Slave,Address,Bits : Integer; ComPort : Integer) : Boolean;  StdCall;   External
'ModBus.Dll';

Function   WriteBits( Var Data; Slave,Address, Bits : Integer; ComPort : Integer) : Boolean;  StdCall;  External
'ModBus.Dll';

Procedure  TimeOutValue( Value : Integer); StdCall; External 'ModBus.Dll';
```

Implementation in Visual Basic

```
Declare Function   OpenCom  Lib " Modbus"  (ByVal ComPort As long , ByVal BaudRate As long,
                          ByVal Parity As Byte, ByVal Bits As long, ByVal Stop As long ) As  Byte

Declare Sub  DisableCom Lib " Modbus" (ByVal ComPort As long )

Declare Function   ReadWords Lib " Modbus" ( ByRef Data as any , ByVal Esclave as long , ByVal Adresse As
long , ByVal Words As long, ByVal ComPort  As long )  As Byte

Declare Function   WriteWords Lib " Modbus" ( ByRef Data as any , ByVal Esclave as long, ByVal Adresse As
long, ByVal Words As long, ByVal ComPort  As long)  As Byte

Declare Function   ReadBits Lib " Modbus" ( ByRef Data as any , ByVal Esclave as long, ByVal Adresse As long,
ByVal Bits As long, ByVal ComPort  As long)  As Byte

Declare Function   WriteBits Lib " Modbus" ( ByRef Data as any , ByVal Esclave as long, ByVal Adresse As long,
ByVal Bits As long, ByVal ComPort  As long)  As Byte

Declare Sub TimeOutValue(ByVal Value As long)
```

Implementation in Builder C++

```
Extern ‘’C’’_declspec(dllimport)bool WINAPI OpenCom (DWORD , DWORD , BYTE , DWORD , DWORD ) ;
Extern ‘’C’’_declspec(dllimport)void WINAPI  DisableCom(DWORD );
Extern ‘’C’’_declspec(dllimport)bool WINAPI  ReadWords( LpVoid , DWORD , DWORD , DWORD  , DWORD ) ;
Extern ‘’C’’_declspec(dllimport)bool WINAPI  WriteWords( LpVoid , DWORD , DWORD , DWORD  , DWORD ) ;
Extern ‘’C’’_declspec(dllimport)bool WINAPI  ReadBits( LpVoid , DWORD , DWORD , DWORD , DWORD  ) ;
Extern ‘’C’’_declspec(dllimport)bool WINAPI  WriteBits( LpVoid , DWORD , DWORD , DWORD , DWORD  ) ;
Extern ‘’C’’_declspec(dllimport)void WINAPI  TimeOutValue( DWORD );
```

Implementation in Visual C++

```
Extern ‘’C’’
{
bool _stdcall  OpenCom (DWORD , DWORD , BYTE , DWORD , DWORD ) ;
void  _stdcall DisableCom(DWORD );
bool _stdcall  ReadWords( LpVoid , DWORD , DWORD , DWORD  , DWORD ) ;
bool _stdcall  writeWords( LpVoid , DWORD , DWORD , DWORD  , DWORD ) ;
bool _stdcall  ReadBits( LpVoid , DWORD , DWORD , DWORD , DWORD  ) ;
bool _stdcall  WriteBits( LpVoid , DWORD , DWORD , DWORD , DWORD  ) ;
void _stdcall  TimeOutValue( DWORD );
}
```

OpenCom Function

The OpenCom function open a serial communication port. This function receive all the necessaries parameters et return an Boolean result. The result is true if the communication port is correctly open.

Function OpenCom (**ComPort**,**BaudRate**: Integer; **Parity** : Char; **Bits,Stop** : Integer): **Boolean**;

Parameters :

ComPort Communication port (32 Bits integer)
1 : Com1
2 : Com2
3 : Com3
4 : Com4
…..
16 : Com16

BaudRate Speed. (32 Bits integer)
110, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 115200, 256000

Parity Parity. (Byte)
Char ‘E’ : Even
‘O’ : Odd
‘N’ : None

Bits Bits. (32 Bits integer)
7 or 8

Stop Bits stop. (32 Bits integer)
1 or 2

Return True if the port is correctly open . (Boolean)
False if port is not open.
Causes : Port missing
Port already open.
Bad speed for this port.

Examples

In Pascal **If** OpenCom (1,9600,’N’,8,1) **Then** Writeln(‘OK’) ;

In C **If** (OpenCom (1,9600,’’N’’,8,1)) Printf(’’ OK’’)

In Basic **If** OpenCom (1,9600,’’N’’,8,1) **Then** Print(‘’ OK’’) ;
End If;

DisableCom

Procédure

The DisableCom procedure close the communication openned with then OpenCom function.

Procedure DisableCom (**ComPort**: Integer);

Parameters :

ComPort Communication port (32 Bits integer)
1 : Com1
2 : Com2
3 : Com3
4 : Com4
…..
16 : Com16

Examples

In Pascal DisableCom (1) ;

In C DisableCom (1) ;

In Basic DisableCom (1) ;

ReadWords Function

The ReadWords Function read a flow of consecutive Words in a Slave.

Function ReadWords(Var **Data**; **Slave,Address,Words** : Integer; **ComPort** : Integer) : **Boolean**;

Parameters :

Data Reception Array Address for the words readen in the Slave. (Pointer)

Slave Slave address (0..255). (32 Bits integer)

Words Number of words to read in the Slave. (32 Bits integer)

ComPort Comport number (32 Bits integer)
1 : Com1
2 : Com2
3 : Com3
4 : Com4

Return True if the read operation is correct.
False if the read operation fail
Causes : Time Out.
Inexistant Slave.
Bad parameters.

Examples

In Pascal

```
Var
  Tablo : Array [1..50] Of Words ;
Begin
   If ReadWords(Tablo,1,$200,50,1) then Writeln('Ok') ;
End ;
```

In C

```
Short tablo[50] ;
If (ReadWords(&Tablo,1,0x200,50,1))  Printf("Ok") ;
```

WriteWords Function

The WriteWords function write a flow of consecutive Words in a Slave.

Function WriteWords(Var **Data**; **Slave,Address,Words** : Integer; **ComPort** : Integer) : **Boolean**;

Parameters :

Data Array Address to transmit into the Slave. (Pointer)

Slave Slave address (0..255). (32 Bits integer)

Words Number of words to write into the Slave. (32 Bits integer)

ComPort Comport number (32 Bits integer)
1 : Com1
2 : Com2
3 : Com3
4 : Com4

Return True if the write operation is correct.
False if the write operation fail
Causes : Time Out.
Inexistant Slave.
Bad parameters.

Examples

In Pascal

```
Var
  Tablo : Array [1..50] Of Words ;
Begin
   If WriteWords(Tablo,1,$200,50,1) then Writeln('Ok') ;
End ;
```

In C

```
Short tablo[50] ;
If (WriteWords(&Tablo,1,0x200,50,1))  Printf("Ok") ;
```

ReadBits Function

The ReadBits Function read a flow of consecutive Bits in a Slave.

Function ReadBits(Var **Data**; **Slave,Address,Bits** : Integer; **ComPort** : Integer) : **Boolean**;

Parameters :

Data Reception Array Address for the bits readen in the Slave. (Pointer)

Slave Slave address (0..255). (32 Bits integer)

Bits Number of bits to read in the Slave. (32 Bits integer)

ComPort Comport number (32 Bits integer)
1 : Com1
2 : Com2
3 : Com3
4 : Com4

Return True if the read operation is correct.
False if the read operation fail
Causes : Time Out.
Inexistant Slave.
Bad parameters.

Examples

In Pascal

```
Var
  Tablo : Array [1..50] Of Boolean ;
Begin
   If ReadBits(Tablo,1,0,50,1) then WriteIn('Ok') ;
End ;
```

In C

```
Bool tablo[50] ;
If (ReadWords(&Tablo,1,0,50,1))  Printf("Ok") ;
```

WriteBits Function

The WriteBits function Write a flow of consecutive Bits in a Slave.

Function WriteBits(Var **Data**; **Slave,Address,Bits** : Integer; **ComPort** : Integer) : **Boolean**;

Parameters :

Data Array Address for the transmission into the Slave. (Pointer)

Slave Slave address (0..255). (32 Bits integer)

Bits Number of bits to write into the Slave. (32 Bits integer)

ComPort Comport number (32 Bits integer)
1 : Com1
2 : Com2
3 : Com3
4 : Com4

Return True if the write operation is correct.
False if the write operation fail
Causes : Time Out.
Inexistant Slave.
Bad parameters.

Examples

In Pascal

```
Var
  Tablo : Array [1..50] Of Boolean ;
Begin
   Tablo[1] :=True ;
   Tablo[2] :=False ;
   If WriteBits(Tablo,1,0,50,1) then Writeln('Ok') ;
End ;
```

In C

```
Bool tablo[50] ;
If (WriteWords(&Tablo,1,0,50,1))  Printf("Ok") ;
```

TimeOutValue Procedure

The TimeOutValue procedure set the current time to wait if there is not reception from the slave.
If the Slave no respond, the read or write procedures is waiting for the timeOut value. If the timeOut is reached, the procedure automatic stop the current transaction et return a status (false) to the main program.

Function TimeOutValue(**Value** : Integer);

Parameters :

Value TimeOut value in ms. (32 Bits integer)
Minimum 50ms. The fixed default timeOut is 1seconde (1000 ms).

Examples

In Pascal

```
TimeOutValue(100) ;
```

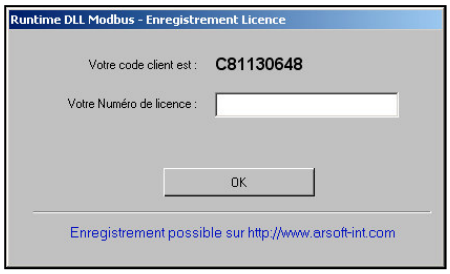
In C

```
TimeOutValue(100) ;
```

Including in the CD ROM

Utility to Test the Serial Link with an Modbus equipment.

Get your machine number for the licence installation.



For each exploitation The Modbus DLL is running under licence.
Send us your machine number for unlock the DLL.

Transfert the licence to another PC.

Support ARSoft International :

hotline@arsoft-int.com