Interface Modbus

Manuel Utilisateur

Interface avec la DLL Modbus.DLL

Les procédures et fonctions implémentées dans la DLL Modbus.dll sont décrites sommairement :

OpenCom	Ouvre un port de communication série.
DisableCom	Ferme un port de communication
ReadWords	Lecture de mots consécutifs dans un esclave.
WriteWords	Ecriture d'une série de mots consécutifs dans un esclave.
ReadBits	Lecture de bits consécutifs dans un esclave.
WriteBits	Ecriture de bits consécutifs dans un esclave.
TimeOutValue	Fixe la valeur du Timeout dans une communication.

La DLL Modbus.DLL doit être dans le même répertoire que vos programmes ou dans un répertoire accessible à tous vos programmes (ex : Windows ou WinNt)

Attention dans vos prototypes respectez les majuscules et minuscules.

Note :
La DLL Modbus.DLL respecte la norme Modbus en vigeur.
L' interfaçage d'une Dll avec certains outils de développement nécessite la création d'une librairie d'importation.

En C++ Builder : Cette librairie d'importation (Fichier .lib) est normalement créée automatiquement sinon, la commande Implib.exe sous Dos permet de la créer.

En Visual C++ : Cette librairie d'importation est normalement créée automatiquement sinon, la commande lib.exe sous Dos permet de la créer.
L'interfaçage avec la Dll nécessite aussi l'écriture d'un module de déclaration (Fichier .def voir exemple suivant).

```
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
```

Implémentation en Pascal (Delphi ou Visual Pascal)

```
//**** External dans 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; Esclave,Adresse,Words : Integer; ComPort : Integer) :
Boolean; StdCall;  External 'ModBus.Dll';

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

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

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

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

Implémentation en 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)
```

Implémentation en 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 );
```

Implémentation en 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 Fonction

La fonction OpenCom permet l'ouverture d'un canal de communication série. Cette fonction reçoit tous les paramètres nécessaires et renvoie un résultat booléen indiquant si l'ouverture du canal spécifié c'est correctement effectué.

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

Paramètres :

ComPort Numéro du canal série désiré (Entier 32 Bits) 16 coms Maxi.
1 : Com1
2 : Com2
3 : Com3
4 : Com4
…..
16 : Com16

BaudRate Vitesse désirée sur le port série spécifié. (Entier 32 Bits)
110, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 115200, 256000

Parity Contrôle de Parité sur la communication série. (Byte)
Caractère 'E' : Paire
'O' : Impaire
'N' : Sans parité

Bits Nombre de bits émis par caractères. (Entier 32 Bits)
7 ou 8

Stop Nombre de bits de stop. (Entier 32 Bits)
1 ou 2

Retour True si l'ouverture du canal série c'est effectué correctement. (Booléen)
False si impossibilité d'ouvrir le canal série spécifié.
Causes : Port désigné absent
Port déjà ouvert dans l'application.
Port ne supportant pas la vitesse demandée.

Exemples

En Pascal **If** OpenCom (1,9600,'N',8,1) **Then** Writeln('OK') ;

En C **If** (OpenCom (1,9600,''N'',8,1)) Printf('' Ok'')

En Basic **If** OpenCom (1,9600,''N'',8,1) **Then** Print('' OK'') ;
End If;

DisableCom

Procédure

La procédure DisableCom permet la fermeture du canal de communication série ouvert avec la fonction OpenCom.

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

Paramètres :

ComPort Numéro du canal série désiré (Entier 32 Bits)
1 : Com1
2 : Com2
3 : Com3
4 : Com4
….
16 : Com16

Exemples

En Pascal DisableCom (1) ;

En C DisableCom (1) ;

En Basic DisableCom (1) ;

ReadWords Fonction

La fonction ReadWords permet la lecture d'un nombre de mots consécutifs dans un esclave donné.

Function ReadWords(Var **Data**; **Esclave,Adresse,Words** : Integer; **ComPort** : Integer) : **Boolean**;

Paramètres :

Data Adresse du tableau de réception des mots lus dans l'esclave. (Pointer)

Esclave Adresse de l'esclave (0..255). (Entier 32 Bits)

Words Nombre de mots à lire dans l'esclave spécifié. (Entier 32 Bits)

ComPort Numéro du canal série désiré (Entier 32 Bits)
1 : Com1
2 : Com2
3 : Com3
4 : Com4

Retour True si l'opération de lecture c'est effectué correctement.
False si l'opération de lecture a échoué
Causes : Time Out.
Esclave inexistant.
Paramètres erronés

Exemples

En Pascal

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

En C

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

WriteWords Fonction

La fonction WriteWords permet l'écriture d'un nombre de mots consécutifs dans un esclave donné.

Function WriteWords(Var **Data**; **Esclave,Adresse,Words** : Integer; **ComPort** : Integer) : **Boolean**;

Paramètres :

Data Adresse du tableau de mots (16 Bits) à émettre vers l'esclave.

Esclave Adresse de l'esclave (0..255). (Entier 32 Bits)

Words Nombre de mots à écrire dans l'esclave spécifié. (Entier 32 Bits)

ComPort Numéro du canal série désiré (Entier 32 Bits)
1 : Com1
2 : Com2
3 : Com3
4 : Com4

Retour True si l'opération d'écriture c'est correctement effectué.
False si l'opération d'écriture a échoué
Causes : Time Out.
Esclave inexistant.
Paramètres erronés

Exemples

En Pascal

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

En C

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

ReadBits Fonction

La fonction ReadBits permet la lecture d'un nombre de Bits consécutifs dans un esclave donné.

Function ReadBits(Var **Data**; **Esclave,Adresse,Bits** : Integer; **ComPort** : Integer) : **Boolean**;

Paramètres :

Data Adresse du tableau de réception des bits lus dans l'esclave. (Pointer)

Esclave Adresse de l'esclave (0..255). (Entier 32 Bits)

Bits Nombre de bits à lire dans l'esclave spécifié. (Entier 32 Bits)

ComPort Numéro du canal série désiré (Entier 32 Bits)
1 : Com1
2 : Com2
3 : Com3
4 : Com4

Retour True si l'opération de lecture c'est effectué correctement. (Booléen)
False si l'opération de lecture a échoué
Causes : Time Out.
Esclave inexistant.
Paramètres erronés

Exemples

En Pascal

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

En C

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

WriteBits

Fonction

La fonction WriteBits permet l'écriture d'un nombre de Bits consécutifs dans un esclave donné.

Function WriteBits(Var **Data**; **Esclave,Adresse,Bits** : Integer; **ComPort** : Integer) : **Boolean**;

Paramètres :

Data Adresse du tableau de bits à émettre lus dans l'esclave. (Pointer)

Esclave Adresse de l'esclave (0..255). (Entier 32 Bits)

Bits Nombre de bits à écrire dans l'esclave spécifié. (Entier 32 Bits)

ComPort Numéro du canal série désiré (Entier 32 Bits)
1 : Com1
2 : Com2
3 : Com3
4 : Com4

Retour True si l'opération d'écriture c'est correctement effectué. (Booléen)
False si l'opération de lecture a échoué
Causes : Time Out.
Esclave inexistant.
Paramètres erronés

Exemples

En 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 ;
```

En C

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

TimeOutValue Procédure

La procédure TimeOutValue permet de fixer le temps d'attente maximum en réception d'une réponse de l'esclave. Si l'esclave ne répond pas, les procédures de lecture ou d'écriture attendent la valeur du timeOut. Si le temps d'attente est atteint, les procédures abandonnent la transaction en cours et rendent la main au programme principal avec un statut de retour à false.

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

Paramètres :

Temps Temps de TimeOut en ms. (Entier 32 Bits)
Minimum 50ms. Le Temps de timeOut est par défaut à 1seconde (1000 ms).

Exemples

En Pascal

```
TimeOutValue(100) ;
```

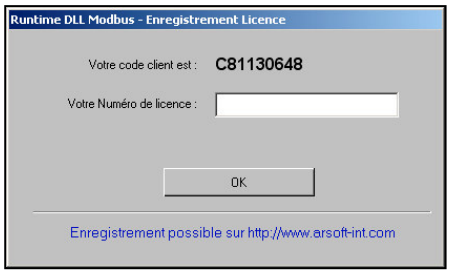
En C

```
TimeOutValue(100) ;
```

Sont inclus dans le CD ROM

Test de la liaison série avec un équipement Modbus

Détermine le numéro de votre machine pour installation de la licence.



La DLL Modbus est soumise à licence pour chaque exploitation.
Nous communiquer le numéro de votre machine afin de déverrouiller votre DLL définitivement.

Désinstallation et transfert de votre licence vers un autre PC.

Support ARSoft International :

hotline@arsoft-int.com