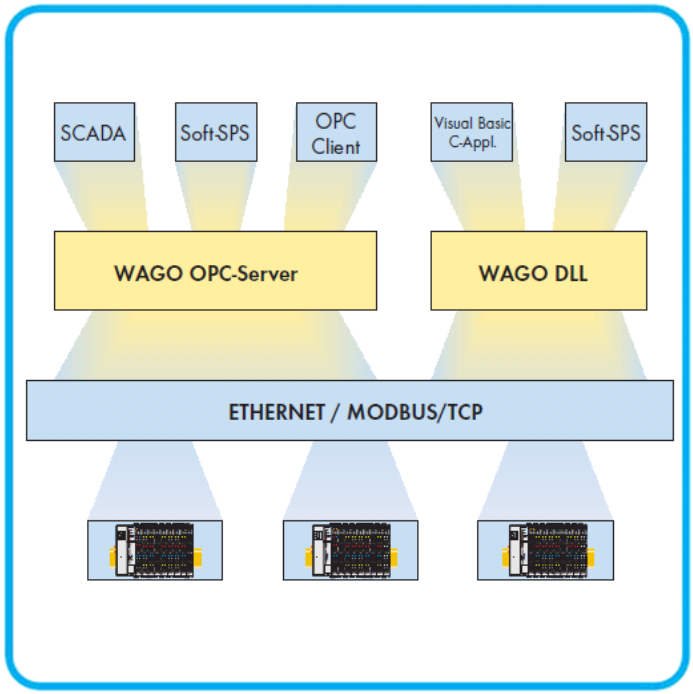
取扱説明書



WAGO Software

WAGO JAPAN DLL FOR MODBUS
WAGO_IO.dll

バージョン 1.0.0（日本語版 2011.09.26）

〒136-0071 東京都江東区亀戸 1-5-7 日鐵 ND タワー
ワゴジャパン株式会社 オートメーション
TEL：03-5627-2059 FAX：03-5627-2055
Web: http://www.wago.co.jp/

本書の作成には万全を期しておりますが、お気づきの点やご意見がございましたら下記までお知らせください。
E-Mail: io-japan@wago.com

目 次

1 重要事項

このマニュアルに記載されているユニットのスムーズな設置とスタートアップのために、この文章に含まれる情報と解説に注意しながら作業を進めていくことを強く推奨致します。

1.1 法的原則

1.1.1 著作権

本書は図表を含めてすべて著作権で保護されています。本書に明記された著作権条項に抵触する第三者による再利用は禁じられています。複製、翻訳、電子的手段または複写による保存および修正を行うには、ワゴジャパン株式会社の同意書が必要です。これに違反した場合、当社には損害賠償を請求する権利が生じます。

1.1.2 使用者の資格基準

本書で説明する製品は、PLC プログラミングの資格を有する技術者、電気機器の専門技術者、または適用規格を熟知している電気機器の専門技術者の指導を受けた者が必ず操作してください。不適切な作業による損害、または本書の内容を順守しないために発生したワゴ製品および他社製品の損害について、ワゴジャパン株式会社は一切の責任を負いかねますのでご了承ください。

1.1.3 用途

使用されるコンポーネントは各用途に応じて、専用のハードウェアおよびソフトウェアコンフィグレーションで動作するようになっています。変更する場合は、必ず本書で記述された範囲内で行ってください。ハードウェアやソフトウェアに対してそれ以外の変更を加えた場合や、コンポーネントが規格に準じて使用されなかった場合は、ワゴジャパン株式会社の責任範囲外となりますのでご注意ください。

改造版および／または新規のハードウェアまたはソフトウェアコンフィグレーションに関する要件については、ワゴジャパン株式会社まで直接お問い合わせください。

1.2 図記号

危険
人的傷害を回避するために、常にこの情報に従ってください。

警告
機器への損害を防ぐために、常にこの情報に従ってください。

注意
円滑な操作を確保するために、状態を常に注意しなければなりません。

静電気放電(ESD)
静電気放電による機器への損傷の警告。リスクを回避するための措置を確認してください。

注意
機器およびソフトウェアを効率よく最適に使用するための手順あるいはアドバイス

詳細情報
補足資料、マニュアル、データシートおよびインターネットページの参照先

1.3 フォント

イタリック	パスおよびファイル名はイタリック体で表記します。 例: *C:¥programs¥WAGO-IO-CHECK*
太字イタリック	メニュー項目は太字のイタリック体で表記します。 例: ***Save***
/	2 つの名前間のスラッシュ記号はメニュー項目の順序を表します。 例: ***File/New***
END	ボタンを押すことは下付の太字で表記します。 例: **ENTER**
< >	キーは太字の山括弧で表記します。 例: **<F5>**
`Courier New`	プログラムコードは `Courier New` フォントで表記します。 例: `END_VAR`

1.4 記数法

記数法	例	備考
10 進数	100	通常の表記法
16 進数	0x64	C での表記法
2 進数	'100' '0110.0100'	「'」で囲む 4 ビットごとにドットで区切る

1.5 有効範囲

型番	摘要
759-xxx	API MODBUS DLL

1.6 略語

AI	アナログ入力
AO	アナログ出力
DI	デジタル入力
DO	デジタル出力
I/O	入力／出力

2 WAGO_IO, Modbus DLL

このライブラリは WAGO Ethernet バスカプラ／コントローラのみで使用することができ、Modbus Serial あるいは TCP/UDP において使用することができます。

2.1 概要

DLL は Modbus プロトコルを実行します。
この Modbus DLL はオペレーションシステム(OS) Windows XP, Windows Vista, Windows 7 に対応しており、Windows システムに含まれている TCP/IP 用の Windows Socket 2.0 インターフェースを使用します。
Modbus Serial, TCP あるいは UDP を通信プロトコルとして選択することができます。
プログラミング言語 C および Visual BASIC において使用することができます。
このライブラリは Open Modbus/TCP プロトコルにおいて、コマンド FC1, FC2, FC3, FC4, FC5, FC6, FC11, FC15 および FC16 のみに対応しています。
この DLL はマイクロソフト Visual Studio 2010 で作成され、サンプルプロジェクトには Visual Basic 6.0, Visual Basic.Net4.0 および C++用を付録しております。

2.2 インストール

WAGO_IO.dll ファイルを Windows の標準的なディレクトリにおいては system32 フォルダ内にコピーしてください。別のディレクトリを選択する場合には、そのエントリは Windows のシステム制御パスに対して環境変数が相当するように適合させなければなりません。

2.3 関数

2.3.1 WAGO_OpenCommPort

WAGO-I/O-System Elements of Dynamic Link Library			
カテゴリ:	Modbus 通信		
名称:	WAGO_OpenCommPort		
形式:	Function **X**	Function block	Program
ライブラリ名:	WAGO_IO.dll		
必要なライブラリ:			
適用:	Serial, Ethernet TCP / UDP バスカプラ／コントローラ		

入力パラメータ:	**データ型:**	**コメント:**
hConn	Long*	コネクションハンドラ Long（32 ビット）変数ポインタ
*pProp	CommProp	通信プロパティ構成
.iUsedEthernet	Int	Int(32 ビット) 0 の場合はシリアルの COM ポートを、その他は Ethernet ポートを使用
.iNodeAdr	Int	Int(32 ビット) シリアルスレーブのアドレス
.iMotorolaByteOrder	Int	Int(32 ビット) 0 の場合はインテル方式のバイト順序を、その他はモトローラ方式を使用
.iPortNo	Int	Int(32 ビット) シリアル COM ポート番号
.iParityChk	int	Int(32 ビット) 0 =パリティなし、1 =偶数、2 =奇数
.iBaudRate	int	Int(32 ビット) 通信速度
.iDataBits	int	Int(32 ビット) 7 = 7 ビット、 8 = 8 ビット
.iStopBits	int	Int(32 ビット) 1 = 1 ストップビット、2 = 2 ストップビット
.iFlowCtrl	int	Int(32 ビット) 0 = オフ、1 = xOn / xOff、2 = Material
.iErrChk	int	Int(32 ビット) 0 は CRC チェックなし、CRC チェックあり
.pcHostAddress	char*	サーバーの IP アドレスに関する文字列ポインタ（Ethernet のみ）
.iUsedTCP	int	Int(32 ビット) 0 = Modbus UDP、1 = Modbus TCP

出力パラメータ:	**データ型:**	**コメント:**
WAGO_OpenCommPort	int	Int 値(32 ビット) 0 は正常、それ以外はエラーコード

記入:

```
int WAGO_OpenCommPort
   (
   long* hConn,
   CommProp *pProp
   );
```

関数について:

この関数は COM ポートを初期化およびオープンし、hConn 数は COM ポートのリファレンスであり、この COM ポートを使用するのに必要な関数に対して同じにしなければなりません。

2.3.2 WAGO_Data

WAGO-I/O-System Elements of Dynamic Link Library			
カテゴリ:	Modbus 通信		
名称:	WAGO_Data		
形式:	Function **X**	Function block	Program
ライブラリ名:	WAGO_IO.dll		
必要なライブラリ:			

WAGO-I/O-System Elements of Dynamic Link Library		
適用:	Serial, Ethernet TCP / UDP バスカプラ／コントローラ	
入力パラメータ:	**データ型:**	**コメント:**
hConn	long	接続拡張子 Long (32 ビット)変数
sNodeAdr	int	Int(32 ビット) シリアルスレーブアドレス
sFunCode	int	Int(32 ビット) 機能コード
sAddrRead	int	Int(32 ビット) 読込を開始するアドレス
sSizeRead	Int	Int(32 ビット) 読込ビットあるいはワード数
pwDataRead	WORD*	WORD（16 ビット） 読込データ配列ポインタ
sAddrWrite	Int	Int(32 ビット) 書込を開始するアドレス
sSizeWrite	Int	Int(32 ビット) 書込ビットあるいはワード数
pwDataWrite	WORD*	WORD（16 ビット） 書込データ配列ポインタ
出力パラメータ:	**データ型:**	**コメント:**
WAGO_Data	Int	Int(32 ビット) 0 は正常、それ以外はエラーコード

記入:

```
int WAGO_Data
   (
   long hConn,
   int sNodeAdr,
   int sFunCode,
   int sAddrRead,
   int sSizeRead,
   WORD* pwDataRead,
   int sAddrWrite,
   int sSizeWrite,
   WORD* pwDataWrite
   );
```

関数について:

この関数でスレーブ／サーバーのデータの読込・書込を可能にします。
対応機能コード:
1: Read Coils
2: Read Discrete Inputs
3: Read Holding Registers
4: Read Input Register
5: Write Single Coil
6: Write Single Register
11: Get Com Event Counter
15: Write Multiple Coils
16: Write Multiple Registers
23: Read/Write Multiple Registers (Ethernet Modbus のみ)

2.3.3 WAGO_CloseCommPort

WAGO-I/O-System Elements of Dynamic Link Library			
カテゴリ:	Modbus 通信		
名称:	WAGO_CloseCommPort		
形式:	Function **X**	Function block	Program
ライブラリ名:	WAGO_IO.dll		
必要なライブラリ:			
適用:	Serial, Ethernet TCP / UDP バスカプラ／コントローラ		

WAGO-I/O-System Elements of Dynamic Link Library		
入力パラメータ:	**データ型:**	**コメント:**
hConn	long*	接続識別子 Long（32 ビット）変数ポインタ
出力パラメータ:	**データ型:**	**コメント:**
WAGO_CloseCommPort	Int	Int(32 ビット)　0 は正常、それ以外はエラーコード
記入:		

`int WAGO_CloseCommPort`
`  (`
`  long hConn,`
`  );`		

関数について:		
この関数は COM ポートのリファレンス（hConn）によって選択される COM ポートをリセットおよびクローズします。		

2.4 エラーコード

エラーコード	内容
0	エラーなし
-20	COM ポート　オープニングエラー
-21	不正 “パリティ” パラメータ
-22	不正 “ストップビット” パラメータ
-23	不正 “フロー制御” パラメータ
-24	COM ポートパラメータ書込エラー
-25	COM ポート　クロージングエラー
-30	TCP ソケット無効
-31	TCP ソケット基本設定エラー
-32	ソケットのアドバンス設定エラー
-33	ホスト不明
-34	IP アドレス無効
-35	TCP ソケット　クロージングエラー
-40	UDP ソケット無効
-41	UDP ソケットパラメータ設定エラー
-42	ホスト不明
-43	UDP ソケット　クロージングエラー
-101	通信オブジェクトを得ることができない
-102	接続情報を得ることができない
-105	機能コード未対応
-106	データ受信エラー
-107	データ送信エラー
-108	CRC 回答 ID 不正
-109	回答が応答エラー
-201	不正関数
-202	不正データアドレス
-203	不正データ値
-204	スレーブデバイス不良

3 構文例

付属のサンプルプロジェクトはここでの例よりも複雑な構造をしていますので、ここでの例では WAGO-I/O システムと PC 間通信を確立するのに必要な基本的な事項について示してあります。

3.1 Visual Basic

Visual BASIC における制御の基本はエントリフィールド形式の制御エレメント SlaveID, Functioncode, ReadAdr, ReadSize, WriteAdr および WriteSize と同様の制御エレメント btnOpen, btnRequest および btnClose を備えたフォームです。
Modbus 接続は btnOpen ボタンをクリックすることによりノードに設定されます。
シリアルスレーブのアドレスはエントリフィールド SlaveID に入力します。
Modbus の機能コードはエントリフィールド FunctionCode に入力します。
有効にするアドレスはエントリフィールド ReadAdr あるいは WriteAdr に入力します。
必要なデータ数はエントリフィールド ReadSize あるいは WriteSize に入力します。
ボタン btnRequest をクリックした後、データは送信あるいは読み込まれます。
イベントの処理はボタンのイベント "Click" により呼び出されます。
ノードへの接続は btnClose をクリックする、あるいはフォームを閉じることにより切断されます。

3.1.1 Visual BASIC 6.0

3.1.1.1 概要

VB6 プロジェクトでは宣言なしに変数を使用することができませんので、最初の記載は "Option Explicit" にしなければなりません。

VB6 ではフォームに User Data Type を宣言できません、VB6 の "Standard module" を使用してください。この module には User data Type, グローバル変数および DLL 関数を宣言します。

VB6 と C++ではデータ型が同じサイズではありません:
C++は整数値に 32 ビットを使用しますが VB6 は 16 ビットで、VB6 には 32 ビットの Long データ型を使用しなければなりません。
C++でワードは 16 ビットを使用し、VB6 にはそれがないので VB6 の整数値データ型には 16 ビットを使用します。
VB6 にはストラクチャが存在しないので Type 宣言を使用しなければなりません。

3.1.1.2 モジュールのｵﾌﾟｼｮﾝ、ｽﾄﾗｸﾁｬ、DLL 関数およびｸﾞﾛｰﾊﾞﾙ変数の宣言

```
' Option
Option Explicit

' Data Type
   Type tagCommProp
      iUsedEthernet As Long
      iNodeAdr As Long
      iMotorolaByteOrder As Long
      iPortNo As Long                  ' Used by Serial COM Port
      iParityChk As Long               ' Used by Serial COM Port
      iBaudRate As Long          ' Used by Serial COM Port
      iDataBits As Long          ' Used by Serial COM Port
      iStopBits As Long          ' Used by Serial COM Port
      iFlowCtrl As Long          ' Used by Serial COM Port
      iErrChk As Long                  ' Used by Serial COM Port
      pcHostAddress As String          ' Used by Ethernet
      iUsedTCP As Long           ' Used by Ethernet
   End Type

' DLL functions
   Public Declare Function WAGO_OpenCommPort Lib "WAGO_IO.DLL" _
      (ByRef hConn As Long, ByRef pProp As tagCommProp) As Long

   Public Declare Function WAGO_CloseCommPort Lib "WAGO_IO.DLL" _
      (ByVal hConn As Long) As Long
```

```
    Public Declare Function WAGO_Data Lib "WAGO_IO.DLL" _
        (ByVal hConn As Long, ByVal iNodeAdr As Long, ByVal iFunCode As Long, _
        ByVal iStartReadAddr As Long, ByVal iReadSize As Long, _
        ByRef pReadData As Integer, ByVal iStartWriteAddr As Long, _
        ByVal iWriteSize As Long, ByRef pWriteData As Integer) As Long

' Global Variable
Public hConn As Long
```

3.1.1.3 フォームの変数およびイベント処理の宣言

```
' Option
Option Explicit

' btnOpen
    Private Sub btnOpen_Click()

         Dim dtProp As tagCommProp
        Dim r As Long
        Dim IPaddr As String
        IPaddr = "192.168.0.2"

       dtProp.iUsedEthernet = 1
       dtProp.iNodeAdr = 1
       dtProp.iMotorolaByteOrder = 0
       dtProp.iPortNo = 1
       dtProp.iParityChk = 0
       dtProp.iBaudRate = 9600
       dtProp.iDataBits = 8
       dtProp.iStopBits = 1
       dtProp.iFlowCtrl = 0
       dtProp.iErrChk = 1
       dtProp.pcHostAddress = IPaddr
       dtProp.iUsedTCP = 1

       r = WAGO_OpenCommPort(hConn, dtProp)

         If r < 0 Then
             MsgBox("Error : " & r)
             WAGO_CloseCommPort(hConn)
         End If

End Sub

' btnRequest
     Private Sub btnRequest_Click()

       Dim iBufferRead(0 To 255) As Integer
       Dim iBufferWrite(0 To 255) As Integer
       Dim r As Long

       r = WAGO_Data(hConn, CLng(SlaveID.Text), CLng(FunctionCode.Text),
       CLng(ReadAdr.Text), CLng(ReadSize.Text), iBufferRead(0), CLng(WriteAdr.Text),
       CLng(WriteSize.Text), iBufferWrite(0))

         If r < 0 Then
             MsgBox("Error : " & r)
         End If

    End Sub

' btnClose
    Private Sub btnClose_Click()
         WAGO_CloseCommPort(hConn)
        End
    End Sub

' Closing the form
```

```
Private Sub Form_QueryUnload (Cancel As Integer, UnloadMode As Integer)
      WAGO_CloseCommPort(hConn)
    End
End Sub
```

3.1.2 Visual BASIC.Net 4.0

3.1.2.1 概要

初期設定により既に有効になっているため、VB.Net では ”Option Explicit” を追加する必要はありません。
VB.Net と C++ではデータ型は同じサイズです。

3.1.2.2 フォームのストラクチャおよび DLL 関数の宣言

```
' Data Type
   Structure tagCommProp
      Dim iUsedEthernet As Integer
      Dim iNodeAdr As Integer
      Dim iMotorolaByteOrder As Integer
      Dim iPortNo As Integer            ' Used by Serial COM Port
      Dim iParityChk As Integer ' Used by Serial COM Port
      Dim iBaudRate As Integer  ' Used by Serial COM Port
      Dim iDataBits As Integer          ' Used by Serial COM Port
      Dim iStopBits As Integer          ' Used by Serial COM Port
      Dim iFlowCtrl As Integer          ' Used by Serial COM Port
      Dim iErrChk As Integer            ' Used by Serial COM Port
      Dim pcHostAddress As String       ' Used by Ethernet
      Dim iUsedTCP As Integer   ' Used by Ethernet
   End Structure

' DLL functions
   Public Declare Function WAGO_OpenCommPort Lib "WAGO_IO.DLL" _
      (ByRef hConn As Integer, ByRef pProp As tagCommProp) As Integer

   Public Declare Function WAGO_CloseCommPort Lib "WAGO_IO.DLL" _
      (ByVal hConn As Integer) As Integer

   Public Declare Function WAGO_Data Lib "WAGO_IO.DLL" _
      (ByVal hConn As Integer, ByVal iNodeAdr As Integer, _
       ByVal iFunCode As Integer, ByVal iStartReadAddr As Integer, _
       ByVal iReadSize As Integer, ByRef pReadData As UShort, _
       ByVal iStartWriteAddr As Integer, ByVal iWriteSize As Integer, _
       ByRef pWriteData As UShort) As Integer
```

3.1.2.3 フォームの変数およびイベント処理の宣言

```
' Global Variable
Public hConn As Integer

' btnOpen
Private Sub btnOpen_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles btnOpen.Click

        Dim dtProp As tagCommProp
       Dim r As Integer
       Dim IPaddr As String = "192.168.0.2"

      dtProp.iUsedEthernet = 1
      dtProp.iNodeAdr = 1
      dtProp.iMotorolaByteOrder = 0
      dtProp.iPortNo = 1
      dtProp.iParityChk = 0
      dtProp.iBaudRate = 9600
      dtProp.iDataBits = 8
      dtProp.iStopBits = 1
      dtProp.iFlowCtrl = 0
      dtProp.iErrChk = 1
```

```
        dtProp.pcHostAddress = IPaddr
        dtProp.iUsedTCP = 1

        r = WAGO_OpenCommPort(hConn, dtProp)

          If r < 0 Then
              MsgBox("Error : " & r)
              WAGO_CloseCommPort(hConn)
          End If

End Sub

' btnRequest
    Private Sub btnRequest_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles btnRequest.Click

        Dim iBufferRead(0 To 255) As UShort
        Dim iBufferWrite(0 To 255) As UShort
        Dim i,r As Integer

        For i = 0 To 24
            iBufferRead(i) = 0
            iBufferWrite(i) = 1
        Next

        r = WAGO_Data(hConn, CInt(SlaveID.Text), CInt(FunctionCode.Text),
        CInt(ReadAdr.Text), CInt(ReadSize.Text), iBufferRead(0), CInt(WriteAdr.Text),
        CInt(WriteSize.Text), iBufferWrite(0))

          If r < 0 Then
              MsgBox("Error : " & r)
          End If

    End Sub

' btnClose
    Private Sub btnClose_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles btnClose.Click
          WAGO_CloseCommPort(hConn)
        End
    End Sub

' Closing the form
    Private Sub Form_FormClosing(ByVal sender As Object, _
        ByVal e As System.Windows.Forms.FormClosingEventArgs) _
        Handles Me.FormClosing
          WAGO_CloseCommPort(hConn)
        End
    End Sub
```

3.2 Visual C++

3.2.1 概要

C における以下のプログラム例では、DLL は#pragma コメント(ライブラリ “WAGO_IO.lib”)コマンドで呼び出されます。
その際に、接続は WAGO_OpenCommPort にて設定されます。
接続設定成功後、出入力の書込および読込についての関数 WAGO_Data が呼び出されます。
状態は各機能を呼出後、モニタ上に表示されます。
プログラムの最後に、接続は関数 WAGO_CloseCommPort で切断されます。

3.2.2 ヘッダー

```
#ifndef _WAGO_IO_H_
#define _WAGO_IO_H_
    #ifdef __cplusplus
              extern "C"
              {
      #endif
      //------------------------------------------------------------------------
      // Type Definitions
      //------------------------------------------------------------------------
      typedef struct tagCommProp I
      {
              int             iUsedEthernet;
              int             iNodeAdr;
              int             iMotorolaByteOrder;
              // Used COM Port
              int             iPortNo;
          int         iParityChk;
              int             iBaudRate;
           int                iDataBits;
              int             iStopBits;
              int             iFlowCtrl;
              int             iErrChk;
              // Used Ethernet
              const char*     pcHostAddress;
              int             iUsedTCP;
              // the service port will be assigned 502 port if normal case.
      } CommProp;
      //------------------------------------------------------------------------
      // Prototypes
      //------------------------------------------------------------------------

      #if !defined(WAGO_DLL_FUNC_CALL)
              #define WAGO_DLL_FUNC_CALL __declspec(dllimport) int __stdcall
      #endif //WAGO_DLL_FUNC_CALL
      WAGO_DLL_FUNC_CALL WAGO_OpenCommPort( long* hConn,
                                            CommProp *pProp );
      WAGO_DLL_FUNC_CALL WAGO_CloseCommPort( long hConn );
      WAGO_DLL_FUNC_CALL WAGO_Data( long hConn, int sNodeAdr,
                                    int sFunCode, int sAddrRead,
                                    int sSizeRead, short* pwDataRead,
                                    int sAddrWrite, int sSizeWrite,
                                    short* pwDataWrite);
      #ifdef __cplusplus
              }
      #endif
#endif
```

3.2.3 プログラム

```
#include "stdafx.h"
#include <windows.h>
#include <tchar.h>
#include <assert.h>
#include <stdio.h>
#include "WAGO_IO.h"

#pragma comment(lib, "WAGO_IO.lib")

int main(int argc, char* argv[])
{

	CommProp *pComm = new CommProp;
	int ret = 0;
	long hConn = 0;
	int iSlaveID = 0;
	int iFunctionCode = 0;
	int iReadAdress = 0;
	int iReadQuantity = 0;
	int iWriteAdress = 0;
	int iWriteQuantity = 0;

	char sContinue[80] = "";

	short pReadBuffer[255];
	short pWriteBuffer[255];

	for(int i=0;i<255;i++)
	{
		pWriteBuffer[i] = 0x3333;
		pReadBuffer[i] = 0;
	}

	char* IPaddr = "192.192.10.74";

	pComm->iBaudRate = 9600;
	pComm->iDataBits = 8;
	pComm->iErrChk = 0;
	pComm->iFlowCtrl = 0;
	pComm->iMotorolaByteOrder = 0;
	pComm->iNodeAdr = 1;
	pComm->iParityChk = 0;
	pComm->iPortNo = 1;
	pComm->iStopBits = 1;
	pComm->iUsedEthernet = 1;
	pComm->iUsedTCP = 1;
	pComm->pcHostAddress = IPaddr;

	printf( "¥n Call WAGO_OpenCommPort¥n");

	ret = WAGO_OpenCommPort(&hConn, pComm);

	printf( "¥n WAGO_OpenCommPort returns: %d ¥n", ret);

	if( ret == 0 )
	{
		do
		{
			printf ("¥n Enter the Slave Serial Address (if not use 0):
");
			scanf_s ("%i", &iSlaveID);

			do
			{
				printf ("¥n Function Code available : 1, 2, 3, 4,
```

```
5, 6, 11, 15, 16, 23.¥n Enter your choise: ");
                                scanf_s ("%i", &iFunctionCode);
                        }while (iFunctionCode != 1 && iFunctionCode != 2 &&
iFunctionCode != 3 &&
                                iFunctionCode != 4 && iFunctionCode != 5 &&
iFunctionCode != 6 &&
                                iFunctionCode != 11 && iFunctionCode != 15 &&
iFunctionCode != 16 &&
                                iFunctionCode != 23);

                        switch(iFunctionCode)
                        {
                                case 1:
                                case 2:
                                case 3:
                                case 4:
                                        printf ("¥n Enter the Read Address: ");
                                        scanf_s ("%i", &iReadAdress);

                                        printf ("¥n Enter the Read Quantity: ");
                                        scanf_s ("%i", &iReadQuantity);

                                        iWriteAdress = 0;

                                        iWriteQuantity = 0;
                                        break;
                                case 5:
                                case 6:
                                case 15:
                                case 16:
                                        iReadAdress = 0;

                                        iReadQuantity= 0;

                                        printf ("¥n Enter the Write Address: ");
                                        scanf_s ("%i", &iWriteAdress);

                                        printf ("¥n Enter the Write Quantity: ");
                                        scanf_s ("%i", &iWriteQuantity);
                                        break;
                                case 11:
                                        iReadAdress = 0;

                                        iReadQuantity= 1;

                                        iWriteAdress = 0;

                                        iWriteQuantity = 1;
                                        break;
                                case 23:
                                        printf ("¥n Enter the Read Address: ");
                                        scanf_s ("%i", &iReadAdress);

                                        printf ("¥n Enter the Read Quantity: ");
                                        scanf_s ("%i", &iReadQuantity);

                                        printf ("¥n Enter the Write Address: ");
                                        scanf_s ("%i", &iWriteAdress);

                                        printf ("¥n Enter the Write Quantity: ");
                                        scanf_s ("%i", &iWriteQuantity);
                                        break;
                        }

                        printf( "¥n Call WAGO_Data¥n");

                        WAGO_Data(hConn, iSlaveID, iFunctionCode, iReadAdress,
iReadQuantity, pReadBuffer, iWriteAdress, iWriteQuantity, pWriteBuffer);
```

```
                        printf( "¥n WAGO_Data returns: %d ¥n", ret);

                        printf ("¥n Do you whant to continue ?");
                        scanf_s ("%s", sContinue);

                }while(sContinue[0] == 'y'|| sContinue[0] == 'Y');

        }

        printf( "¥n Call WAGO_CloseCommPort¥n");

        ret = WAGO_CloseCommPort(hConn);

        printf( "¥n WAGO_CloseCommPort returns: %d ¥n", ret);

        return 0;
}
```

ワゴジャパン 株式会社

〒136-0071 東京都江東区亀戸 1-5-7 日鐵 ND タワー

TEL (03)5627-2050㈹ FAX (03)5627-2055

E-Mail info-jp@wago.com

ホームページ http://www.wago.co.jp