

NModbus API Manual

Version 1.2, October 2013

Written by Renee Lin

Contents

Contents	2
1. About this manual	4
2. NModbus Master API	5
2.1. CreateRtu	5
2.2. CreateAscii	6
2.3. CreateIp(TcpClient)	7
2.4. CreateIp(UdpClient)	11
2.5. Retries [Property]	12
2.6. ReadTimeout[Property]	12
2.7. ReadCoils	13
2.8. ReadInputs	14
2.9. ReadHoldingRegisters	15
2.10. ReadInputRegisters	16
2.11. WriteSingleCoil	17
2.12. WriteSingleRegister	18
3. NModbus Slave API	19
3.1. CreateRtu	20
3.2. CreateAscii	21
3.3. CreateTcp	22
3.4. CreateUdp	23
3.5. CreateDefaultDataStore	24
3.6. ModbusSlaveRequestReceived[event]	24
3.7. DataStoreWrittenTo[event]	25
3.8. Listen	28
3.9. CoilDiscretets[DO data array]	28
3.10. InputDiscretets [DI data array]	29
3.11. HoldingRegisters [AO data array]	29
3.12. InputRegisters [AI data array]	30
4. Common API	31

4.1. Dispose.....31

Appendix A : NModbus Error codes 32

Appendix B : AI, AO convert data type..... 35

1.About this manual

The manual is made for introducing API which is used in NModbus.

What is NModbus?

NModbus can achieve protocol of Modbus. It is developed and maintained on a voluntary basis and provided free of charge.

ICP DAS verified and improved the DLL based on the official released NModbus_net-2.0_1.11.0.0-source.zip. Programmers can use the DLL released by ICP DAS to develop a Modbus application for regular Windows based PCs or WinCE based devices.

The DLL features

- a.Modbus/RTU Master/Slave
- b.Modbus/ASCII Master/Slave
- c.Modbus/TCP Master/Slave
- d.Modbus/UDP Master/Slave

The relative DLL and demos can download as below.

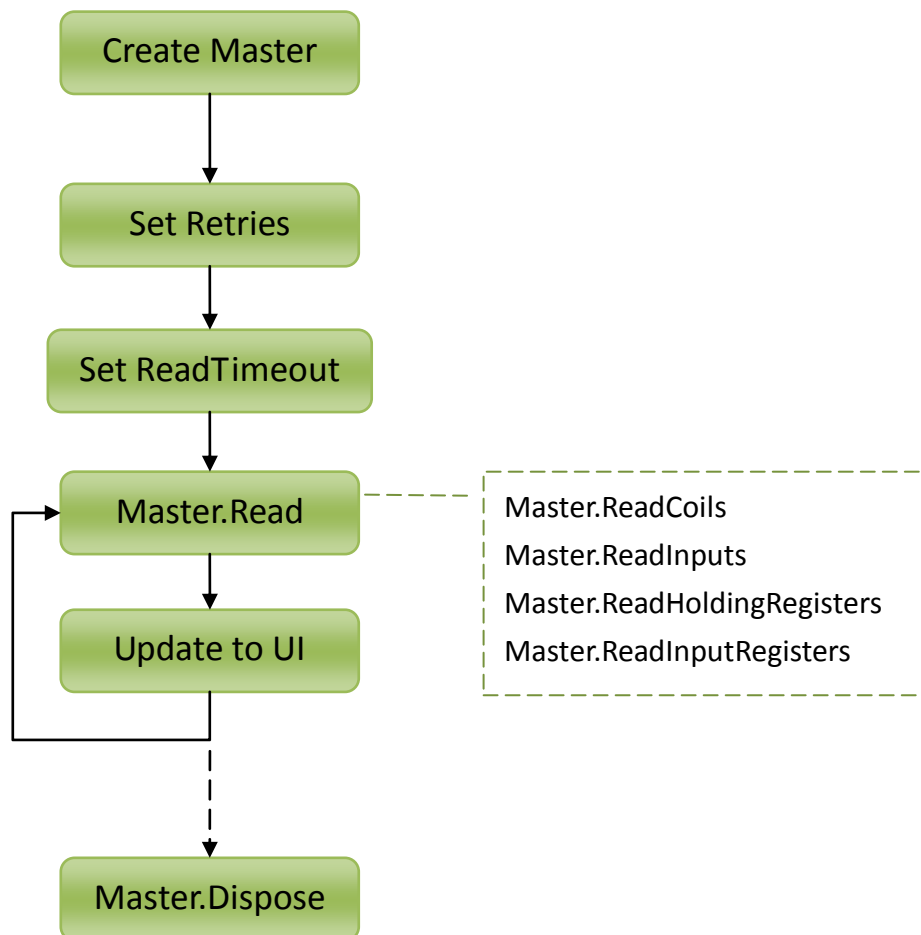
- a. WinForm
 - [DLL and document](#) : nModbusPC.dll, log4net.dll
 - [Demo](#) (For C#, VB.Net)
- b. WinCE
 - [DLL and document](#) : nModbusCE.dll, CABC.dll, FC19.dll
 - [Demo](#) (For C#, VB.Net)

More about Modbus→<http://www.icpdas.com/products/PAC/i-8000/modbus.htm>

Which is suitable for NModbus?

WinForm	XPAC(WES 2009)
	Win8,Win7,Vista,Xp(.NET framework 2 or later required)
WinCE	ViewPAC(CE5)
	WinPAC(CE5)
	XPAC(CE6)

2. NModbus Master API



2.1. CreateRtu

RTU master create connection to serial port.

Syntax

```
C#  
ModbusSerialMaster CreateRtu(  
    SerialPort serialPort  
)
```

Parameters

serialPort

The serialPort is created by new SerialPort(), and serialport is opened by serialPort.Open().

If serialport doesn't specified value, it will use default property values to open port. For example, a default port name of COM1, the Parity property defaults to the None, the DataBits property defaults to 8, and the StopBits property defaults to 1.

Return Value

Return ModbusSerialMaster.

Examples

[C#]

```
SerialPort serialPort = new SerialPort(); //Create a new SerialPort object.
serialPort.PortName = "COM1";
serialPort.BaudRate = 115200;
serialPort.DataBits = 8;
serialPort.Parity = Parity.None;
serialPort.StopBits = StopBits.One;
serialPort.Open();
ModbusSerialMaster master = ModbusSerialMaster.CreateRtu(serialPort);
```

2.2. CreateAscii

Ascii master create connection to serial port.

Syntax

C#

```
ModbusSerialMaster CreateAscii(
    SerialPort serialPort
)
```

Parameters

serialPort

The serialPort is created by new SerialPort(), and serialport is opened by serialPort.Open().

Return Value

Return ModbusSerialMaster.

Examples

[C#]

```
SerialPort serialPort = new SerialPort(); //Create a new SerialPort object.
serialPort.PortName = "COM1";
serialPort.BaudRate = 115200;
serialPort.DataBits = 8;
serialPort.Parity = Parity.None;
serialPort.StopBits = StopBits.One;
serialPort.Open();
ModbusSerialMaster master = ModbusSerialMaster.CreateAscii(serialPort);
```

2.3. CreateIp(TcpClient)

IP master create connection to TCP.

Syntax

C#

```
ModbusIpMaster CreateIp(
    TcpClient tcpClient
)
```

Parameters

tcpClient

The tcpClient is connected by tcpClient.BeginConnect(), and tcpClient is created by new TcpClient().

Return Value

Return ModbusIpMaster.

Examples

[C#]

```
string ipAddress = "10.0.0.69";
int tcpPort = 502;
TcpClient tcpClient = new TcpClient(); //Create a new TcpClient object.
tcpClient.BeginConnect(ipAddress, tcpPort, null, null);
ModbusIpMaster master = ModbusIpMaster.CreateIp(tcpClient);
```

If want to reconnect when offline, refer to following codes.

[C#]

```
[DllImport("WININET", CharSet = CharSet.Auto)]
static extern bool InternetGetConnectedState(ref InternetConnectionState lpdwFlags, int
dwReserved);
enum InternetConnectionState : int
{
    INTERNET_CONNECTION_MODEM = 0x1,
    INTERNET_CONNECTION_LAN = 0x2,
    INTERNET_CONNECTION_PROXY = 0x4,
    INTERNET_RAS_INSTALLED = 0x10,
    INTERNET_CONNECTION_OFFLINE = 0x20,
    INTERNET_CONNECTION_CONFIGURED = 0x40
}
TcpClient tcpClient;
ModbusIpMaster master;
string ipAddress = "10.0.0.69";
int tcpPort = 502;
DateTime dtDisconnect = new DateTime();
DateTime dtNow = new DateTime();
bool NetworkIsOk = false;

private void btStart_Click(object sender, EventArgs e)
{
    // when button clicked, connect to Modbus TCP Server
    NetworkIsOk = Connect();
    timer1.Enabled = true;
```



```

}
private bool Connect()
{
    if (master != null)
        master.Dispose();
    if (tcpClient != null)
        tcpClient.Close();
    if (CheckInternet())
    {
        try
        {
            tcpClient = new TcpClient();
            IAsyncResult asyncResult = tcpClient.BeginConnect(ipAddress, tcpPort, null, null);
            asyncResult.AsyncWaitHandle.WaitOne(3000, true); //wait for 3 sec
            if (!asyncResult.IsCompleted)
            {
                tcpClient.Close();
                Console.WriteLine(DateTime.Now.ToString() + ":Cannot connect to server.");
                return false;
            }
            // create Modbus TCP Master by the tcpclient
            master = ModbusIpMaster.CreateIp(tcpClient);
            master.Transport.Retries = 0; //don't have to do retries
            master.Transport.ReadTimeout = 1500;
            Console.WriteLine(DateTime.Now.ToString() + ":Connect to server.");
            return true;
        }
        catch (Exception ex)
        {
            Console.WriteLine(DateTime.Now.ToString() + ":Connect process " + ex.StackTrace +
"==>" + ex.Message);
            return false;
        }
    }
    return false;
}
private bool CheckInternet()

```

```

{
    //http://msdn.microsoft.com/en-us/library/windows/desktop/aa384702(v=vs.85).aspx
    InternetConnectionState flag = InternetConnectionState.INTERNET_CONNECTION_LAN;
    return InternetGetConnectedState(ref flag, 0);
}
private void timer1_Tick(object sender, EventArgs e)
{
    //start timer1, timer1.Interval = 1000 ms
    try
    {
        if (NetworkIsOk)
        {
            #region Master to Slave
            //read AI, AO, DI, DO
            #endregion
        }
        else
        {
            //retry connecting
            dtNow = DateTime.Now;
            if ((dtNow - dtDisconnect) > TimeSpan.FromSeconds(10))
            {
                Console.WriteLine(DateTime.Now.ToString() + ":Start connecting");
                NetworkIsOk = Connect();
                if (!NetworkIsOk)
                {
                    Console.WriteLine(DateTime.Now.ToString() + ":Connecting fail. Wait for retry");
                    dtDisconnect = DateTime.Now;
                }
            }
            else
            {
                Console.WriteLine(DateTime.Now.ToString() + ":Wait for retry connecting");
            }
        }
    }
    catch(Exception ex)

```

```

{
    if (ex.Source.Equals("System"))
    {
        //set NetworkIsOk to false and retry connecting
        NetworkIsOk = false;
        Console.WriteLine(ex.Message);
        dtDisconnect = DateTime.Now;
    }
}
}
}

```

2.4. CreateIp(UdpClient)

IP master create connection to UDP.

Syntax

```

C#
ModbusIpMaster CreateIp(
    UdpClient udpClient
)

```

Parameters

udpClient

The *udpClient* is connected by *udpClient.Connect()*, and *udpClient* is created by new *UdpClient()*.

Return Value

Return *ModbusIpMaster*.

Examples

```

[C#]
IPAddress ipAddress = "10.0.0.69";
int udpPort = 502;
UdpClient udpClient = new UdpClient(); //Create a new UdpClient object.
udpClient.Connect(ipAddress, udpPort);

```

```
ModbusIpMaster master = ModbusIpMaster.CreateIp(udpClient);
```

2.5.Retries [Property]

[Property]Number of times to retry sending message after encountering a failure such as an IOException, TimeoutException, or a corrupt message.

Syntax

C#

```
int Retries { get; set; }
```

Examples

[C#]

```
string ipAddress = "10.0.0.69"; //use TCP for example
int tcpPort = 502;
TcpClient tcpClient = new TcpClient();
tcpClient.BeginConnect(ipAddress, tcpPort, null, null);
ModbusIpMaster master = ModbusIpMaster.CreateIp(tcpClient);
master.Transport.Retries = 0;
```

Remarks

Default Value is Retries = 3.

It doesn't need to retry in NModbus and set Retries = 0.

2.6.ReadTimeout[Property]

[Property]Gets or sets the number of milliseconds before a timeout occurs when a read operation does not finish.

Syntax

C#

```
int ReadTimeout { get; set; }
```

Examples

[C#]

```
SerialPort serialPort = new SerialPort();//use RTU for example
serialPort.Open();
ModbusSerialMaster master = ModbusSerialMaster.CreateRtu(serialPort);
master.Transport.ReadTimeout = 300; //milliseconds
```

Remarks

ReadTimeout recommended value

- a. RTU: ReadTimeout = 300
- b. TCP: ReadTimeout = 1500

2.7. ReadCoils

Read coils status.

Syntax

C#

```
bool[] ReadCoils(
    byte slaveID,
    ushort startAddress,
    ushort numOfPoints
)
```

Parameters

slaveID

Address of device to read values from.

startAddress

Address to begin reading.

numOfPoints

Number of coils to read.

Return Value

Return bool[].

Examples

[C#]

```
byte slaveID = 1;
ushort startAddress = 0;
ushort numOfPoints = 10;
bool[] coilstatus = master.ReadCoils(slaveID , startAddress , numOfPoints);
```

2.8. ReadInputs

Read input status.

Syntax

C#

```
bool[] ReadInputs(
    byte slaveID,
    ushort startAddress,
    ushort numOfPoints
)
```

Parameters

slaveID

Address of device to read values from.

startAddress

Address to begin reading.

numOfPoints

Number of discrete inputs to read.

Return Value

Return bool[].

Examples

[C#]

```
byte slaveID = 1;  
ushort startAddress = 0;  
ushort numOfPoints = 10;  
bool[] status = master.ReadInputs(slaveID , startAddress , numOfPoints);
```

2.9. ReadHoldingRegisters

Read holding registers value.

Syntax

C#

```
ushort[] ReadHoldingRegisters(  
    byte slaveID,  
    ushort startAddress,  
    ushort numOfPoints  
)
```

Parameters

slaveID

Address of device to read values from.

startAddress

Address to begin reading.

numOfPoints

Number of holding registers to read.

Return Value

Return ushort[].

Examples

[C#]

```
byte slaveID = 1;  
ushort startAddress = 0;  
ushort numOfPoints = 10;
```

```
ushort[] holding_register = master.ReadHoldingRegisters(slaveID, startAddress,
numOfPoints);
```

2.10. ReadInputRegisters

Read input registers value.

Syntax

```
C#
-----
ushort[] ReadInputRegisters(
    byte slaveID,
    ushort startAddress,
    ushort numOfPoints
)
```

Parameters

slaveID

Address of device to read values from.

startAddress

Address to begin reading.

numOfPoints

Number of input registers to read.

Return Value

Return ushort[].

Examples

```
[C#]
byte slaveID = 1;
ushort startAddress = 0;
ushort numOfPoints = 10;
ushort[] register = master.ReadInputRegisters(slaveID, startAddress, numOfPoints);
```


2.11. WriteSingleCoil

Write a coil value.

Syntax

```
C#  
void WriteSingleCoil(  
    byte slaveID,  
    ushort coilAddress,  
    bool value  
)
```

Parameters

slaveID

Address of the device to write to.

coilAddress

Address to write value to.

value

If the address is going to be written, the value is TRUE.

If the address isn't going to be written, the value is FALSE.

Return Value

The function doesn't have return value.

Examples

```
[C#]  
byte slaveID = 1;  
ushort coilAddress = 1;  
bool value = true;  
master.WriteSingleCoil(slaveID , coilAddress ,value);
```

2.12. WriteSingleRegister

Write a holding register value.

Syntax

```
C#  
void WriteSingleRegister(  
    byte slaveID,  
    ushort registerAddress,  
    ushort value  
)
```

Parameters

slaveID

Address of the device to write to.

registerAddress

Address to write value to.

value

Value to write.

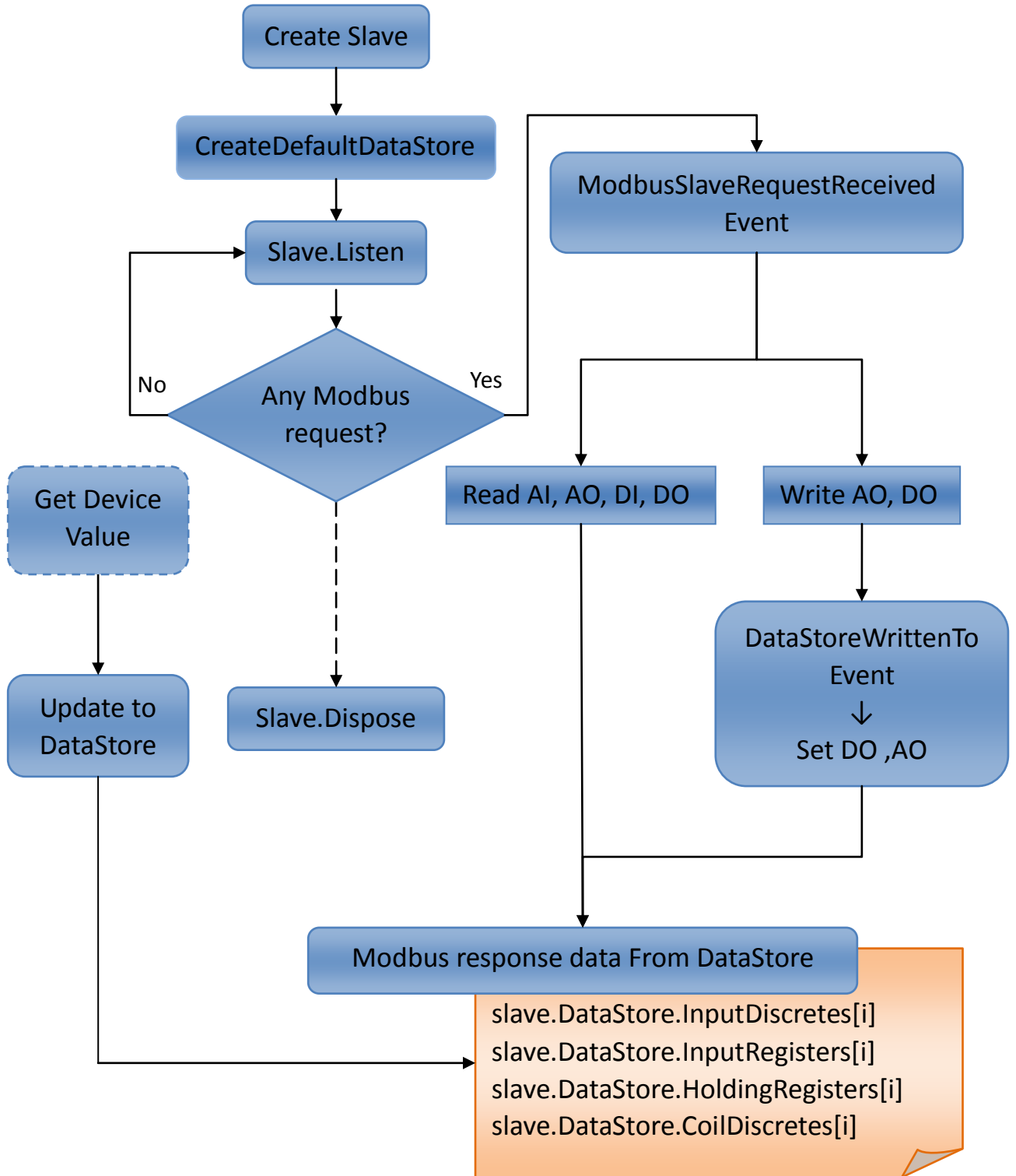
Return Value

The function doesn't have return value.

Examples

```
[C#]  
byte slaveID = 1;  
ushort registerAddress = 1;  
ushort value = 1000;  
master.WriteSingleRegister(slaveID, registerAddress, value);
```

3.NModbus Slave API



3.1.CreateRtu

Create a RTU slave connection.

Syntax

```
C#  
ModbusSerialSlave CreateRtu(  
    byte slaveID,  
    SerialPort serialPort  
)
```

Parameters

slaveID

Address of device to create.

serialPort

The serialPort is created by new SerialPort(), and serialport is opened by serialPort.Open().

Return Value

Return ModbusSerialSlave.

Examples

```
[C#]  
byte slaveID = 1;  
SerialPort serialPort = new SerialPort();  
serialPort.PortName = "COM1";  
serialPort.BaudRate = 115200;  
serialPort.DataBits = 8;  
serialPort.Parity = Parity.None;  
serialPort.StopBits = StopBits.One;  
serialPort.Open();  
ModbusSlave slave = ModbusSerialSlave.CreateRtu(slaveID, serialPort);
```

3.2. CreateAscii

Create an Ascii slave connection.

Syntax

```
C#  
ModbusSerialSlave CreateAscii(  
    byte slaveID,  
    SerialPort serialPort  
)
```

Parameters

slaveID

Address of device to create.

serialPort

The serialPort is created by new SerialPort(), and serialport is opened by serialPort.Open().

Return Value

Return ModbusSerialSlave.

Examples

```
[C#]  
byte slaveID = 1;  
SerialPort serialPort = new SerialPort();  
serialPort.PortName = "COM1";  
serialPort.BaudRate = 115200;  
serialPort.DataBits = 8;  
serialPort.Parity = Parity.None;  
serialPort.StopBits = StopBits.One;  
serialPort.Open();  
ModbusSlave slave = ModbusSerialSlave.CreateAscii(slaveID, serialPort);
```

3.3. CreateTcp

Create a TCP slave connection. Max value of TCP slave from master is 50.

Syntax

```
C#  
ModbusTcpSlave CreateTcp(  
    byte slaveID,  
    TcpListener tcpListener  
)
```

Parameters

slaveID

Address of device to create.

tcpListener

The tcpListener is created by new TcpListener (), and tcpListener start to listen by tcpListener.Start().

Return Value

Return ModbusTcpSlave.

Examples

```
[C#]  
int port = 502;  
IPHostEntry ipEntry = Dns.GetHostEntry(Dns.GetHostName());  
IPAddress[] addr = ipEntry.AddressList;  
TcpListener tcpListener = new TcpListener(addr[0], port);  
tcpListener.Start();  
  
ModbusSlave slave = ModbusTcpSlave.CreateTcp(slaveID, slaveTcpListener);
```

3.4. CreateUdp

Create a UDP slave connection.

Syntax

```
C#  
ModbusUdpSlave CreateUdp(  
    byte slaveID,  
    UdpClient client  
)
```

Parameters

slaveID

Address of device to create.

client

Client is created and initialized by new UdpClient(), and bind with specified port number.

Return Value

Return ModbusUdpSlave.

Examples

```
[C#]  
int port = 502;  
UdpClient client = new UdpClient(port);  
  
ModbusSlave slave = Modbus.Device.ModbusUdpSlave.CreateUdp(slaveID, client);  
slave.ModbusSlaveRequestReceived += new  
EventHandler<ModbusSlaveRequestEventArgs>(Modbus_Request_Event);  
slave.DataStore = Modbus.Data.DataStoreFactory.CreateDefaultDataStore();  
slave.DataStore.DataStoreWrittenTo += new  
EventHandler<DataStoreEventArgs>(Modbus_DataStoreWriteTo);  
slave.Listen();
```

3.5. CreateDefaultDataStore

Create memory space in Datastore. AI and AO's Datastore set to 0. DI and DO's Datastore set to false. For each memory default size is 65535 and range is 1 to 65535.

Syntax

C#

```
DataStore CreateDefaultDataStore()
```

The following syntax can determine the size of memory.

```
DataStore CreateDefaultDataStore(  
    ushort coilsCount,  
    ushort inputsCount,  
    ushort holdingRegistersCount,  
    ushort inputRegistersCount  
)
```

Examples

[C#]

```
slave.DataStore = Modbus.Data.DataStoreFactory.CreateDefaultDataStore();
```

Return Value

Return DataStore.

Remarks

“slave” is defined by ModbusSlave and create slave connection. For example: To create TCP slave connection, syntax is = ModbusTcpSlave.CreateTcp(slaveID, slaveTcpListener).

3.6. ModbusSlaveRequestReceived[event]

Occurs when a modbus slave receives a request. You can disassemble request packet and set particular action here.

Syntax

C#

Examples

```
[C#]
slave.ModbusSlaveRequestReceived += new
EventHandler<ModbusSlaveRequestEventArgs>(Modbus_Request_Event);
//trigger Modbus_Request_Event
private void Modbus_Request_Event(object sender,
Modbus.Device.ModbusSlaveRequestEventArgs e)
{
    //disassemble packet from master
    byte fc = e.Message.FunctionCode;
    byte[] data = e.Message.MessageFrame;
    byte[] byteStartAddress = new byte[] { data[3], data[2] };
    byte[] byteNum = new byte[] { data[5], data[4] };
    Int16 StartAddress = BitConverter.ToInt16(byteStartAddress, 0);
    Int16 NumOfPoint = BitConverter.ToInt16(byteNum, 0);

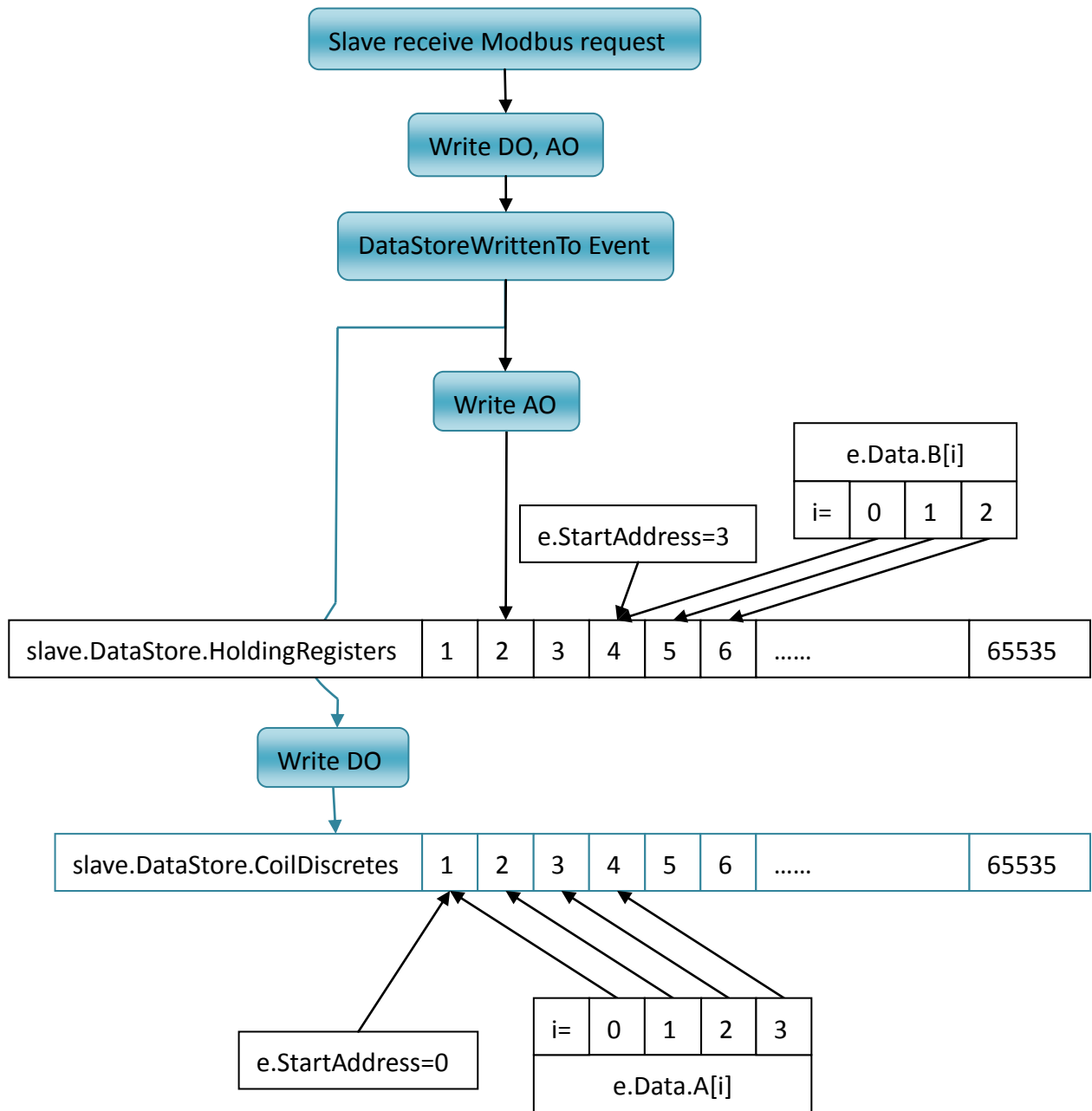
    Console.WriteLine(fc.ToString() + "," + StartAddress.ToString() + "," +
NumOfPoint.ToString());
}
```

Remarks

“slave” is defined by ModbusSlave and create slave connection. For example: To create TCP slave connection, syntax is = ModbusTcpSlave.CreateTcp(slaveID, slaveTcpListener).

3.7.DataStoreWrittenTo[event]

Occurs when a slave receive write AO or DO command to Datastore from master via a Modbus command. Address starts from 1 to 65535.



Syntax

C#

```
EventHandler<DataStoreEventArgs> DataStoreWrittenTo
```

Examples

[C#]

```
slave.DataStore = Modbus.Data.DataStoreFactory.CreateDefaultDataStore();
```

```

slave.DataStore.DataStoreWrittenTo += new
EventHandler<DataStoreEventArgs>(Modbus_DataStoreWriteTo);

//when receive write AO or DO command from master, it will trigger following function
private void Modbus_DataStoreWriteTo(object sender, Modbus.Data.DataStoreEventArgs e)
{
    switch (e.ModbusDataType)
    {
        case ModbusDataType.HoldingRegister:
            for (int i = 0; i < e.Data.B.Count; i++)
            {
                //Set AO
                //e.Data.B[i] already write to
slave.DataStore.HoldingRegisters[e.StartAddress + i + 1]
                //e.StartAddress starts from 0
                //You can set AO value to hardware here
            }
            break;
        case ModbusDataType.Coil:
            for (int i = 0; i < e.Data.A.Count; i++)
            {
                //set DO
                //e.Data.A[i] already write to
slave.DataStore.CoilDiscretes[e.StartAddress + i + 1]
                //e.StartAddress starts from 0
                //You can set DO value to hardware here
            }
            break;
    }
}
}

```

Remarks

“slave” is defined by ModbusSlave and create slave connection. For example: To create TCP slave connection, syntax is = ModbusTcpSlave.CreateTcp(slaveID, slaveTcpListener).

3.8.Listen

Slave starts listening for requests.

Syntax

C#

```
void Listen()
```

Examples

```
[C#]
int port = 502; //use Tcp for example
IPHostEntry ipEntry = Dns.GetHostEntry(Dns.GetHostName());
IPAddress[] addr = ipEntry.AddressList;
TcpListener tcpListener = new TcpListener(addr[0], port);
tcpListener.Start();

ModbusSlave slave =ModbusTcpSlave.CreateTcp(slaveID, tcpListener);
slave.ModbusSlaveRequestReceived += new
EventHandler<ModbusSlaveRequestEventArgs>(Modbus_Request_Event);
slave.DataStore = Modbus.Data.DataStoreFactory.CreateDefaultDataStore();
lave.DataStore.DataStoreWrittenTo += new
EventHandler<DataStoreEventArgs>(Modbus_DataStoreWriteTo);
slave.Listen();
```

Return Value

The function doesn't have return value.

3.9.CoilDiscretes[DO data array]

Data array of DO values. Address starts from 1 to 65535.

Syntax

C#

```
ModbusDataCollection<bool> CoilDiscretes { get; private set; }
```

Examples

```
[C#]  
slave.DataStore.CoilDiscretes[1] = true;  
slave.DataStore.CoilDiscretes[65535] = false;
```

3.10. InputDiscretes [DI data array]

Data array of DI values. You can store DI values in the array. Address starts from 1 to 65535.

Syntax

```
C#  
ModbusDataCollection<bool> InputDiscretes { get; private set; }
```

Examples

```
[C#]  
slave.DataStore.InputDiscretes[1] = true;  
slave.DataStore.InputDiscretes[65535] = false;
```

3.11. HoldingRegisters [AO data array]

Data array of AO values. Address starts from 1 to 65535.

Syntax

```
C#  
ModbusDataCollection<ushort> HoldingRegisters { get; private set; }
```

Examples

```
[C#]  
slave.DataStore.HoldingRegisters[1] = 222;  
slave.DataStore.HoldingRegisters[65535] = 333;
```

3.12. InputRegisters [AI data array]

Data array of AI values. You can store AI values in the array. Address starts from 1 to 65535.

Syntax

C#

```
ModbusDataCollection<ushort> InputRegisters { get; private set; }
```

Examples

[C#]

```
slave.DataStore.InputRegisters[1] = 222;  
slave.DataStore.InputRegisters[65535] = 333;
```

4. Common API

4.1. Dispose

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Syntax

```
C#  
void Dispose()
```

Parameters

None.

Return Value

The function doesn't have return value.

Examples

```
[C#]  
string ipAddress = "10.0.0.69"; //use TCP master for example  
int tcpPort = 502;  
TcpClient tcpClient = new TcpClient(); //Create a new TcpClient object.  
tcpClient.BeginConnect(ipAddress, tcpPort, null, null);  
ModbusIpMaster master = ModbusIpMaster.CreateIp(tcpClient);  
master.Dispose();
```

Appendix A : NModbus Error codes

Code	Name	Meaning
01	ILLEGAL FUNCTION	The function code received in the query is not an allowable action for the server.
02	ILLEGAL DATA ADDRESS	The data address received in the query is not an allowable address for the server.
03	ILLEGAL DATA VALUE	A value contained in the query data field is not an allowable value for the server.
04	SLAVE DEVICE FAILURE	An unrecoverable error occurred while the server attempting to perform the requested action.
05	ACKNOWLEDGE	This response is returned to prevent a timeout error from occurring in the client (or master) when the server (or slave) needs a long duration of time to process accepted request.
06	SLAVE DEVICE BUSY	The server (or slave) is engaged in processing a long-duration program command, and the client (or master) should retransmit the message later when the server (or slave) is free.
08	MEMORY PARITY ERROR	The server (or slave) attempted to read record file, but detected a parity error in the memory.
0A	GATEWAY PATH UNAVAILABLE	The gateway is misconfigured or overloaded.
0B	GATEWAY TARGET DEVICE FAILED TO RESPOND	No response was obtained from the target device. Usually means that the device is not present on the network. °

Examples

[C#]

```
//use TCP master for example
try
{
    #region Master to Slave
    //read AI, AO, DI, DO
    #endregion
}
catch (Exception exception)
{
    //Connection exception
    //No response from server.
    //The server may close the connection, or response timeout.
    if (exception.Source.Equals("System"))
    {
        NetworkIsOk = false;
        Console.WriteLine(exception.Message);
        this.Text = "Off line " + DateTime.Now.ToString();
        dtDisconnect = DateTime.Now;
    }
    //The server return error code.You can get the function code and exception code.
    if (exception.Source.Equals("nModbusPC"))
    {
        string str = exception.Message;
        int FunctionCode;
        string ExceptionCode;

        str = str.Remove(0, str.IndexOf("\r\n") + 17);
        FunctionCode = Convert.ToInt16(str.Remove(str.IndexOf("\r\n")));
        Console.WriteLine("Function Code: " + FunctionCode.ToString("X"));

        str = str.Remove(0, str.IndexOf("\r\n") + 17);
        ExceptionCode = str.Remove(str.IndexOf("-"));
        switch (ExceptionCode.Trim())
        {
```

```
    case "1":
        Console.WriteLine("Exception Code: " + ExceptionCode.Trim() + "----> Illegal
function!");
        break;
    case "2":
        Console.WriteLine("Exception Code: " + ExceptionCode.Trim() + "----> Illegal data
address!");
        break;
    case "3":
        Console.WriteLine("Exception Code: " + ExceptionCode.Trim() + "----> Illegal data
value!");
        break;
    case "4":
        Console.WriteLine("Exception Code: " + ExceptionCode.Trim() + "----> Slave device
failure!");
        break;
    }
}
```

Appendix B : AI, AO convert data type

The input and output value of register are ushort type. These examples help you convert value between ushort and float.

If you need further convert examples, please go to:

http://ftp.icpdas.com/pub/cd/8000cd/napdos/modbus/nmodbus/demo/c%23.net/convertdatatype_c%23/

Ushort(Int16) to float(Int32)

```
[C#]
//Convert ushort array to Float
ushort[] data = new ushort[2] { 59769, 17142};
float[] floatData = new float[data.Length / 2 ];
Buffer.BlockCopy(data, 0, floatData, 0, data.Length * 2);
for (int index = 0; index < floatData.Length; index++)
{
    //print out the value
    Console.WriteLine(floatData[index / 2].ToString("0.0000")); //123.4560
}
Console.ReadLine();
```

Float(Int32) to ushort(Int16)

```
[C#]
//Convert Float to short
ushort[] uintData = new ushort[2];
floatData = new float[1] { 223.4560f };
Buffer.BlockCopy(floatData, 0, uintData, 0, 4);
for (int index = 0; index < uintData.Length; index++)
{
    //uintData[0] = 29884; uintData[1] = 17247
    Console.WriteLine(string.Format("uintData[{0}] = {1}", index, uintData[index]));
}
Console.ReadLine();
```