NMODBUS

NModbus Sample Code

Simple Modbus serial RTU master write holding registers example.

```
using (SerialPort port = new SerialPort("COM1"))
{
	// configure serial port
	port.BaudRate = 9600;
	port.DataBits = 8;
	port.Parity = Parity.None;
	port.StopBits = StopBits.One;
	port.Open();

	// create modbus master
	IModbusSerialMaster master = ModbusSerialMaster.CreateRtu(port);

	byte slaveID = 1;
	ushort startAddress = 100;
	ushort[] registers = new ushort[] { 1, 2, 3 };

	// write three registers
	master.WriteMultipleRegisters(slaveID, startAddress, registers);
}
```

Simple Modbus serial ASCII master read holding registers example.

```
using (SerialPort port = new SerialPort("COM1"))
{
	// configure serial port
	port.BaudRate = 9600;
	port.DataBits = 8;
	port.Parity = Parity.None;
	port.StopBits = StopBits.One;
	port.Open();

	// create modbus master
	IModbusSerialMaster master = ModbusSerialMaster.CreateAscii(port);

	byte slaveID = 1;
	ushort startAddress = 1;
	ushort numRegisters = 5;

	// read five registers
	ushort[] registers = master.ReadHoldingRegisters(slaveID, startAddress, numRegisters);

	for (int i = 0; i < numRegisters; i++)
		Console.WriteLine("Register {0}={1}", startAddress + i, registers[i]);
}

// output:
// Register 1=0
// Register 2=0
// Register 3=0
// Register 4=0
// Register 5=0
```

Simple Modbus serial USB RTU master write multiple coils example.

```
using (FtdUsbPort port = new FtdUsbPort(0))
{
	// configure usb port
	port.BaudRate = 9600;
	port.DataBits = 8;
	port.Parity = FtdParity.None;
	port.StopBits = FtdStopBits.One;
	port.Open();

	// create modbus master
	IModbusSerialMaster master = ModbusSerialMaster.CreateRtu(port);

	byte slaveID = 1;
	ushort startAddress = 1;

	// write three coils
	master.WriteMultipleCoils(slaveID, startAddress, new bool[] { true, false, true });
}
```

Donate

NModbus is developed and maintained on a voluntary basis and provided free of charge.

Make A Donation

Upcoming Features

- ~~WPF NModbus Master~~
- Master Console Wrapper
- FTDI USB Slave

Simple Modbus serial USB ASCII master write multiple coils example.

```
using (FtdUsbPort port = new FtdUsbPort(0))
{
	// configure usb port
	port.BaudRate = 9600;
	port.DataBits = 8;
	port.Parity = FtdParity.None;
	port.StopBits = FtdStopBits.One;
	port.Open();

	// create modbus master
	IModbusSerialMaster master = ModbusSerialMaster.CreateAscii(port);

	byte slaveID = 1;
	ushort startAddress = 1;

	// write three coils
	master.WriteMultipleCoils(slaveID, startAddress, new bool[] { true, false, true });
}
```

Simple Modbus TCP master read inputs example.

```
using (TcpClient client = new TcpClient("127.0.0.1", 502))
{
	ModbusIpMaster master = ModbusIpMaster.CreateTcp(client);

	// read five input values
	ushort startAddress = 100;
	ushort numInputs = 5;
	bool[] inputs = master.ReadInputs(startAddress, numInputs);

	for (int i = 0; i < numInputs; i++)
		Console.WriteLine("Input {0}={1}", startAddress + i, inputs[i] ? 1 : 0);
}

// output:
// Input 100=0
// Input 101=0
// Input 102=0
// Input 103=0
// Input 104=0
```

Simple Modbus UDP master write coils example.

```
using (UdpClient client = new UdpClient())
{
	IPEndPoint endPoint = new IPEndPoint(new IPAddress(new byte[] { 127, 0, 0, 1 }), 502);
	client.Connect(endPoint);

	ModbusIpMaster master = ModbusIpMaster.CreateUdp(client);

	ushort startAddress = 1;

	// write three coils
	master.WriteMultipleCoils(startAddress, new bool[] { true, false, true });
}
```

Simple Modbus serial RTU slave example.

```
using (SerialPort slavePort = new SerialPort("COM2"))
{
	// configure serial port
	slavePort.BaudRate = 9600;
	slavePort.DataBits = 8;
	slavePort.Parity = Parity.None;
	slavePort.StopBits = StopBits.One;
	slavePort.Open();

	byte unitID = 1;

	// create modbus slave
	ModbusSlave slave = ModbusSerialSlave.CreateRtu(unitID, slavePort);
	slave.DataStore = DataStoreFactory.CreateDefaultDataStore();

	slave.Listen();
}
```

Simple Modbus Serial ASCII slave example.

```
using (SerialPort slavePort = new SerialPort("COM2"))
{
	// configure serial port
	slavePort.BaudRate = 9600;
	slavePort.DataBits = 8;
	slavePort.Parity = Parity.None;
	slavePort.StopBits = StopBits.One;
	slavePort.Open();

	byte unitID = 1;

	// create modbus slave
	ModbusSlave slave = ModbusSerialSlave.CreateAscii(unitID, slavePort);
	slave.DataStore = DataStoreFactory.CreateDefaultDataStore();

	slave.Listen();
}
```

Simple Modbus serial RTU slave example.

```
using (SerialPort slavePort = new SerialPort("COM2"))
{
	// configure serial port
	slavePort.BaudRate = 9600;
	slavePort.DataBits = 8;
	slavePort.Parity = Parity.None;
	slavePort.StopBits = StopBits.One;
	slavePort.Open();

	byte unitID = 1;

	// create modbus slave
	ModbusSlave slave = ModbusSerialSlave.CreateRtu(unitID, slavePort);
	slave.DataStore = DataStoreFactory.CreateDefaultDataStore();

	slave.Listen();
}
```

Simple Modbus TCP slave example.

```
byte slaveID = 1;
int port = 502;
IPAddress address = new IPAddress(new byte[] { 127, 0, 0, 1 });

// create and start the TCP slave
TcpListener slaveTcpListener = new TcpListener(address, port);
slaveTcpListener.Start();

ModbusSlave slave = ModbusTcpSlave.CreateTcp(slaveID, slaveTcpListener);
slave.DataStore = DataStoreFactory.CreateDefaultDataStore();

slave.Listen();

// prevent the main thread from exiting
Thread.Sleep(Timeout.Infinite);
```

Simple Modbus UDP slave example.

```
using (UdpClient client = new UdpClient(502))
{
	ModbusUdpSlave slave = ModbusUdpSlave.CreateUdp(client);
	slave.DataStore = DataStoreFactory.CreateDefaultDataStore();

	slave.Listen();

	// prevent the main thread from exiting
	Thread.Sleep(Timeout.Infinite);
}
```

Modbus TCP master and slave example.

```
byte slaveID = 1;
int port = 502;
IPAddress address = new IPAddress(new byte[] { 127, 0, 0, 1 });
```

```
// create and start the TCP slave
TcpListener slaveTcpListener = new TcpListener(address, port);
slaveTcpListener.Start();
ModbusSlave slave = ModbusTcpSlave.CreateTcp(slaveID, slaveTcpListener);
Thread slaveThread = new Thread(slave.Listen);
slaveThread.Start();

// create the master
TcpClient masterTcpClient = new TcpClient(address.ToString(), port);
ModbusIpMaster master = ModbusIpMaster.CreateTcp(masterTcpClient);

ushort numInputs = 5;
ushort startAddress = 100;

// read five register values
ushort[] inputs = master.ReadInputRegisters(startAddress, numInputs);

for (int i = 0; i < numInputs; i++)
      Console.WriteLine("Register {0}={1}", startAddress + i, inputs[i]);

// clean up
masterTcpClient.Close();
slaveTcpListener.Stop();

// output
// Register 100=0
// Register 101=0
// Register 102=0
// Register 103=0
// Register 104=0
```

Modbus serial ASCII master and slave example.

```
using (SerialPort masterPort = new SerialPort("COM1"))
using (SerialPort slavePort = new SerialPort("COM2"))
{
      // configure serial ports
      masterPort.BaudRate = slavePort.BaudRate = 9600;
      masterPort.DataBits = slavePort.DataBits = 8;
      masterPort.Parity = slavePort.Parity = Parity.None;
      masterPort.StopBits = slavePort.StopBits = StopBits.One;
      masterPort.Open();
      slavePort.Open();

      // create modbus slave on seperate thread
      byte slaveID = 1;
      ModbusSlave slave = ModbusSerialSlave.CreateAscii(slaveID, slavePort);
      Thread slaveThread = new Thread(new ThreadStart(slave.Listen));
      slaveThread.Start();

      // create modbus master
      ModbusSerialMaster master = ModbusSerialMaster.CreateAscii(masterPort);

      master.Transport.Retries = 5;
      ushort startAddress = 100;
      ushort numRegisters = 5;

      // read five register values
      ushort[] registers = master.ReadHoldingRegisters(slaveID, startAddress, numRegisters);

      for (int i = 0; i < numRegisters; i++)
            Console.WriteLine("Register {0}={1}", startAddress + i, registers[i]);
}

// output
// Register 100=0
// Register 101=0
// Register 102=0
// Register 103=0
// Register 104=0
```

Write and Read 32 bit value example.

```
uint largeValue = UInt16.MaxValue + 5;

ushort lowOrderValue = BitConverter.ToUInt16(BitConverter.GetBytes(largeValue), 0);
ushort highOrderValue = BitConverter.ToUInt16(BitConverter.GetBytes(largeValue), 2);

// write large value in two 16 bit chunks
master.WriteMultipleRegisters(slaveID, startAddress, new ushort[] { lowOrderValue, highOrderValue });

// read large value in two 16 bit chunks and perform conversion
ushort[] registers = master.ReadHoldingRegisters(slaveID, startAddress, 2);
uint value = ModbusUtility.GetUInt32(registers[1], registers[0]);
```

Logging example.

NModbus uses log4net. Your application can easily be configured to capture the NModbus log statements, as well as your own. Try adding the following xml to your application's configuration settings. NModbusConsoleAppender consumes only log statements of level INFO and higher while NModbusFileAppender consumes all log statements (output to NModbusLog.txt).

```
<configSections>
    <section name="log4net" type="log4net.Config.Log4NetConfigurationSectionHandler, log4net" />
</configSections>

<log4net>
    <root>
        <level value="DEBUG" />
        <appender-ref ref="NModbusFileAppender" />
        <appender-ref ref="NModbusConsoleAppender" />
    </root>

    <appender name="NModbusFileAppender" type="log4net.Appender.FileAppender">
        <file value="NModbusLog.txt" />
        <layout type="log4net.Layout.PatternLayout">
            <conversionPattern value="%-5level %logger %method - %message%newline" />
        </layout>
    </appender>

    <appender name="NModbusConsoleAppender" type="log4net.Appender.ConsoleAppender">
        <layout type="log4net.Layout.PatternLayout">
            <conversionPattern value="%-5level %logger %method - %message%newline" />
        </layout>
        <filter type="log4net.Filter.LevelRangeFilter">
            <levelMin value="INFO" />
        </filter>
    </appender>
</log4net>
```

To configure log4net based on your application's configuration settings add the following line.

```
log4net.Config.XmlConfigurator.Configure();
```

Example log4net appender output for a the modbus serial RTU master write registers example.

```
// ConsoleAppender
INFO  Modbus.IO.ModbusTransport UnicastMessage - TX: 1, 16, 0, 100, 0, 3, 6, 0, 1, 0, 2, 0, 3
INFO  Modbus.IO.ModbusTransport UnicastMessage - RX: 1, 16, 0, 100, 0, 3

// FileAppender (NModbusLog.txt)
INFO  Modbus.IO.ModbusTransport UnicastMessage - TX: 1, 16, 0, 100, 0, 3, 6, 0, 1, 0, 2, 0, 3
DEBUG Modbus.IO.ModbusRtuTransport Read - Read 4 bytes.
DEBUG Modbus.IO.ModbusRtuTransport ReadResponse - Frame start 1, 16, 0, 100.
DEBUG Modbus.IO.ModbusRtuTransport Read - Read 4 bytes.
DEBUG Modbus.IO.ModbusRtuTransport ReadResponse - Frame end 0, 3, 193, 215.
INFO  Modbus.IO.ModbusTransport UnicastMessage - RX: 1, 16, 0, 100, 0, 3
```

For more information visit the **log4net** project page.

Add your own custom messages.

So your device has custom messages? Not a problem, NModbus exposes interfaces IModbusMessage and IModbusMessageWithData just for that reason. Simply implement custom request and response messages and call the appropriate ModbusMaster.ExecuteCustomMessage overload.

Example from a test demonstrationg both ExecuteCustomMessage overloads:

```
ushort testAddress = 120;
ushort[] testValues = new ushort[] { 10, 20, 30, 40, 50 };
CustomReadHoldingRegistersRequest readRequest = new CustomReadHoldingRegistersRequest(3, SlaveAddress, testAddress, (ushort) testValue
CustomWriteMultipleRegistersRequest writeRequest = new CustomWriteMultipleRegistersRequest(16, SlaveAddress, testAddress, new RegisterC

ushort[] originalValues = Master.ExecuteCustomMessage<CustomReadHoldingRegistersResponse, ushort>(readRequest);
Master.ExecuteCustomMessage<CustomWriteMultipleRegistersResponse>(writeRequest);
ushort[] newValues = Master.ExecuteCustomMessage<CustomReadHoldingRegistersResponse, ushort>(readRequest);
Assert.AreEqual(testValues, newValues);
writeRequest = new CustomWriteMultipleRegistersRequest(16, SlaveAddress, testAddress, new RegisterCollection(originalValues));
Master.ExecuteCustomMessage<CustomWriteMultipleRegistersResponse>(writeRequest);
```

Check out these example custom message implementations:

CustomReadHoldingRegistersRequest
CustomReadHoldingRegistersResponse

CustomWriteMultipleRegistersRequest
CustomWriteMultipleRegistersResponse