SuperCom MODBUS Protokoll Bibliothek

für Windows und Linux

- ADONTEC's Kommunikationslösungen
- Auszug SuperCom MODBUS Protokoll Funktionen

x64 1010010 32 Bit und 64 Bit Versionen verfügbar!

Version 4.0

Preisliste
Produktauswahl
Rückruf

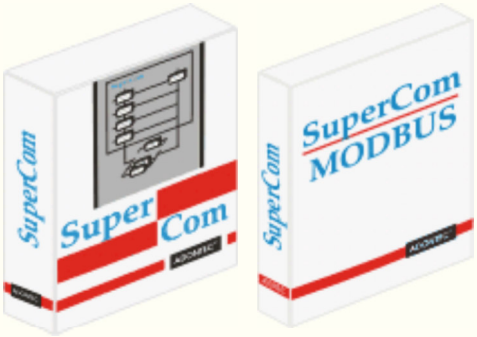
SuperCom MODBUS Protokoll Bibliothek

MODBUS ist ein Industrieprotokoll für die Kontrolle von SPS Steuerungen Automatisierungs-Maschinen verbunden mittels serielle Leitungen (RS-232, RS-422, RS-485) oder Ethernet TCP/IP Verbindungen. Die **SuperCom MODBUS Protokoll Bibliothek** unterstützt das Senden und Empfangen von Daten mit einem oder mehreren Geräten, die das MODBUS-Protokoll berücksichtigen.

Die **SuperCom MODBUS Protokoll Bibliothek** ist eine sehr komplette MODBUS Bibliothek mit Funktionen um einfach und schnell eine stabile MODBUS Anwendung zu entwickeln. Die SuperCom MODBUS Kommunikation Bibliothek kapselt das komplexe MODBUS Protokoll in ein einfach zu nutzendes Set von Funktionen (API), die auf gleicher Weise über serielle und TCP/IP Verbindungen genutzt werden. Dadurch wird eine qualitative und stabile Anwendung in weniger Zeit und mit Kostenersparnis produziert.

Führen Sie einfach Master- oder Slave-Gerätefunktionen in Ihre Anwendung aus, die beispielsweise mit C, C++, C#, Delphi, Visual Basic (inkl. NET, NET Core), Java, LabView usw. geschrieben wurde. Viele Beispiele enthalten.

Die SuperCom MODBUS Protokoll Bibliothek unterstützt die Datenkommunikation zwischen Geräten an seriellen Schnittstellen, am Bus oder Netzwerk gemäß der MODBUS Spezifikation. Das Protokol Modul unterstützt sowohl den ASCII als auch den RTU (*Modbus Remote Terminal Unit) Übertragungsmodus. Im ASCII Modus werden die Daten als ASCII Codes übertragen und im RTU Modus als binäre Byte-Folge (binär Modus).



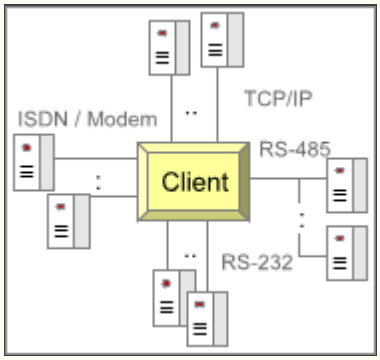
Bis zu 255 gleichzeitige Verbindungen zu Modbus Geräten sind möglich.

Es gibt nur eine API zu lernen! Unabhängig vom Verbindungstyp (seriell, TCP/IP, ISDN) oder Übertragungsmodus (Modbus ASCII, Modbus RTU, Modbus TCP) werden immer dieselben Funktionen genutzt.

Die Protokoll Funktionen erlauben den einfachen Zugriff auf Register und Variablen der SPS. Eine transparente Datenkommunikation für Benutzerspezifische Funktion-Codes und Datenpakete ist auch enthalten. Damit können maschinespezifische Erweiterungen der Automation einfach behandelt werden.

Einfache Handhabung

In den meisten Fällen wird nur eine kleine Menge an Funktionen genutzt um Daten mit einer Modbus fähigen SPS oder Controller auszutauschen. Ein bestehendes Projekt wird schnell erweitert. Weitere Funktionen sind enthalten um die unterschiedlichsten Anforderungen zu bewältigen.



- Eine oder mehrere **gleichzeitige Verbindungen** zu Modbus Geräten sind möglich (bis zu 255 Verbindungen pro Anwendung).
- Arbeitet reibungslos, auch wenn mehrere Verbindungen gleichzeitig ausgeführt werden
- Gleichzeitige Ausführung von mehreren Modbus Klienten Verbindungen.
- Sehr kurze Reaktionszeiten. Erreicht Transaktionszeiten von 2 ms pro Anfrage. (Siehe auch PDF)
- Gleichzeitige Verbindung von MODBUS RTU oder MODBUS ASCII Klient zu Modbus Server.
- Modbus Geräte Simulation inkl. Beispiel
- Auf hohen Datendurchsatz optimierte Funktionen.
- Ein gemeinsames und portables API für Windows, Linux, seriell, TCP/IP usw.
- Abhängig von der genutzten SuperCom Software kann das SuperCom MODBUS Protokoll Modul mehrere serielle und/oder TCP/IP Verbindungen gleichzeitig kontrollieren.
- Die SuperCom MODBUS Protocol Bibliothek kann gleichzeitig mehrere Verbindungen verwalten. Die gleichzeitige Abfrage von mehreren SPS ist möglich.
- Empfangen von Modbus Datenpaketen und *Simulation von Modbus Server oder Protokoll Gateway*.
- Erzeugung des eigenen **Modbus Gateway** oder **Bridge** um Daten zwischen seriellen Schnittstellen und TCP/IP Netzwerk zu leiten z.B. ein *MODBUS TCP zu MODBUS RTU Gateway*.
- Erzeugung eigener **Modbus Server** mit automatische Daten- und Anfrage-Verwaltung oder manuell, durch die Weitergabe der Anfrage an die Anwendung, oder gemischt.
- Ein **Modbus Server**, der mit der SuperCom MODBUS Protocol Bibliothek erzeugt wird, kann bis zu 254 gleichzeitige Klienten Verbindungen unterstützen (Beispiele sind enthalten).
- Mit SuperCom Modbus bridging (z.B. seriell zu TCP/IP).
- Erzeugung eines **Protokoll Gateway** unter Nutzung der, mit SuperCom angebotenen, Industrie Protokollen (SPS Protokolle) oder auch mit eigenen.
- Einfache Nutzung mit SPS von SIEMENS, Allen Bradley, Schneider, WAGO und andere Modbus fähige SPS, Modbus Geräte und Controller. Ein vorinstalliertes MODBUS Protokoll Modul auf SPS oder angeschlossenen Controller oder Gerät vorausgesetzt.
- Empfangen und Senden von Modbus-Rohdaten wird auch unterstützt
- **Datenaufzeichnung** auch auf Telegramm Ebene möglich.
- Unterstützt NET ab 2.0 und NET Core. Mehr ...

Die **SuperCom MODBUS Protokoll Bibliothek** nutzt den *SuperCom Communication Layer*, der ein solides Fundament für die Datenkommunikation bietet, um ohne Kopfzerbrechen stabile Datenkommunikationssoftware schnell zu entwickeln. Dabei macht es für den Programmierer keinen Unterschied ob das MODBUS Protokoll über TCP/IP oder über serielle Leitung (RS-232, RS-422, RS-485, Modem, TAPI) genutzt wird.

Die **SuperCom MODBUS Protokoll Bibliothek** greift direkt auf die entfernte Modbus Station zu ohne Verwendung einer anderen Softwareschicht (z. B. OPC-Server oder Treiber von Drittanbietern), die zuVerzögerungen führen kann.

SuperCom ermöglicht die Anwendung Verbindungen zu einer oder mehreren SPS aufzubauen (nahezu von überall z.B. Modem, TAPI, ISDN, Ethernet (TCP), serielle Leitungen z.B. RS-232, RS-422, RS-485) und Daten auszutauschen bzw. Operanten auszulesen oder zu beschreiben.

Derzeit unterstützte Compiler

Wie mit SuperCom üblich werden sehr viele bekannte Compiler unterstützt. Weitere kommen ständig dazu. Die SuperCom MODBUS Bibliothek kann aktuell mit folgende Compiler Sprachen und Compiler eingesetzt werden: C/C++, C#, Visual C++, C++ Builder, Java, Pascal, Delphi, LabVIEW, Visual Basic, Visual Basic NET (VB .NET), VBA, PowerBuilder, PureBasic. Sollten Sie einen vermissen, bitte anfragen. (Siehe auch)

Beispiele - MODBUS Protokoll API:

1. Ein Ausgangs-Bit ("coil") auslesen

C/C++

Init Sequence: ComInit | RS_OpenLink | Klasse CSuperCom / CTcpClient

```
#define SLAVE_ID  1
TCOMMID Com = COM_2; // Com Index z.B. serielle COM2


    // Seriell, TCP, ... - natives API
ComInit (Com);
ComSetState (Com, 9600, ...);

  :

-- Zugriff --

RS_MBSetConfig(Com,,MODBUS_MODE_RTU,,);

if (RS_MBReadCoil (Com,
                   SLAVE_ID,
                   wCoil,
                   &Buffer))
{
    printf("Coil[%d] = %s ", wCoil, Buffer?"TRUE":"FALSE");
}
else
{
    int ErrorCode = RS_MBGetLastError(Com);

    if (ErrorCode == MB_ERR_EXCEPTION)
        printf("Exception %02X reported from server ", RS_MBGetException(Com));
    else
        printf("Error %d", ErrorCode);
}

ComReset(Com);
```

C/C++ C# Delphi Visual Basic

Obige Beispiele sind fast komplette Programme. Weitere Init-Sequenzen sind hier gelistet.

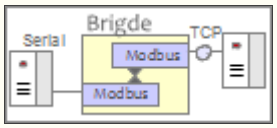
2. Lesen/Schreiben auf Register

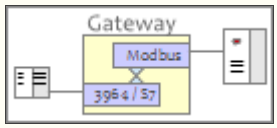
C/C++

```
#define SLAVE_ID  1
TCOMMID Com = COM_2; // Com Index z.B. serielle COM2

WORD Buffer [10];
WORD wStart=0x0000;
WORD wCount=1;
WORD wValue=0x0020;
     :
-- Init Sequenz (siehe oben) --
     :
RS_MBSetConfig(Com,,MODBUS_MODE_RTU,,);

if (RS_MBWriteRegister(Com, SLAVE_ID, wStart, wValue))
    printf ("Read Success.\n");
else
    printf ("Error: %d\n", RS_MBGetLastError(Com));

if (RS_MBReadHoldingRegisters(Com,
                              SLAVE_ID,
                              wStart,
                              &wCount,
                              Buffer))
{
    printf("Read %d Register:", wCount);
    for (int i=0; i<wCount; i++)  printf ("%4X",Buffer[i]);
}
else
    printf ("Error: %d\n", RS_MBGetLastError(Com));
```

C/C++ C# Delphi Visual Basic

Datenüberwachung, Weiterleitung - MODBUS Server oder Gateway Funktionen





Die Datenüberwachung und Weiterleitung von MODBUS Datenpaketen wird unterstützt. Empfangen von Datenpaketen und senden von Antworten (Slave Funktionalität) wird unterstützt. Die Entwicklung eines Protokoll-Konverters oder Gateway von MODBUS in ein anderes Protokoll wird unterstützt. Die *Simulation* eines MODBUS Slave kann u.a. auch während der Anwendungsentwicklung von Nutzem sein (Slave Beispiele verfügbar).

Unterstützte Protokolle

Die *SuperCom MODBUS Protokoll Bibliothek* ermöglicht eine sichere und stabile Daten-Kommunikation mit unterschiedlichen Modbus Geräten unterschiedlicher Hersteller. Es unterstützt nahezu alle bekannte Modbus Optionen und Variationen.

Die *SuperCom MODBUS Protokoll Bibliothek* implementiert das standard MODBUS Protokoll für serielle und TCP/IP Verbindungen* basierend auf der offiziellen Spezifikation der MODBUS Organisation. Das bedeutet **MODBUS ASCII** und **MODBUS RTU** Protokolle über serielle Schnittstellen (RS-232, RS-422, RS-485, Modem, TAPI) und

MODBUS TCP/IP (**MODBUS/TCP**) und **Open MODBUS TCP** über das TCP/IP Netzwerk.

*Eine entsprechende SuperCom Lizenz vorausgesetzt (Seriell und/oder TCP/IP).

> Die *SuperCom MODBUS Protokoll Bibliothek* ist ein sehr umfangreicher und kompletter MODBUS Protokoll-Stack!

Auch enthalten, nicht standardisierte Varianten des MODBUS Protokolls z.B. *RTU over IP* (auch bekannt als MODBUS RTU/IP, MODBUS RTU over TCP) und ermöglichen die SuperCom Anwendung mit OMTS Geräte (OMTS = Out of the MODBUS TCP/IP specification) zu kommunizieren. Da keine offizielle Spezifikation, können andere Namen und Variationen existieren.

> Kurze *SuperCom MODBUS Server Beispiele* werden hier vorgestellt.

Wie einsetzen ?

Die **SuperCom MODBUS Protokoll** Bibliothek kann über jeden Verbindungstyp genutzt werden, die von SuperCom unterstützt wird (aktuell Seriell (RS-232, RS-422, RS-485, Modem, TAPI), TCP/IP, ISDN). Eine Liste von SuperCom Paketen wo MODBUS Protokoll enthalten ist, finden Sie in der folgenden Tabele.

Auch hier gilt: Ein gemeinsames API für Seriell, TCP/IP oder ISDN (ein SuperCom ActiveX API und / oder DLL API).

MODBUS Server Simulator

Beschleunigt den Entwicklungsprozess. Durch die enthaltene Modbus Server Unterstützung steht auch Modbus Simulator Software bereit. Speziell im Paket SuperCom Suite. Eine benutzerfreundliche und schnell arbeitende Testumgebung zur Bereitstellung hochwertiger Software. Es unterstützt auch die Erstellung von auf Software basierende (virtuelle) Modbus-Geräte für Modbus Kienten.

Was bestellen ?

Viele günstigen Kombinations-Pakete mit MODBUS lieferbar (z.B. SuperCom Serial inkl. MODBUS, SuperCom für TCP/IP inkl. MODBUS, SuperCom Protocol Engine, SuperCom Suite).
Die Artikelnummern 638 bzw. 638400 sind nur dann notwendig, wenn das MODBUS Protokoll Modul später zu einem SuperCom zugekauft wird. Am meisten bevorzugt (und im günstigsten Preis) ist der Kauf eines Pakets aus obiger Preisliste, dass bereits die MODBUS-Funktionalität enthält.

Lizenzbedingung

Ausführbare Anwendungen (.EXE) können frei weitergegeben werden. Mehr ....

Unterstützte Compiler

C++, C#, Delphi, Visual C++, Visual Basic, Visual Basic NET, C++ Builder, Borland C/C++, Microsoft C/C++, MinGW, Borland Pascal, Java, LabVIEW, PowerBuilder, PureBasic, VBA und andere Windows Entwicklungsumgebungen (MS .NET ?).

Beispiele

für C/C++, C#, Visual C++, C++ Builder, Java, Pascal, Delphi, Visual Basic, Visual Basic NET (VB .NET), LabVIEW, PureBasic, ...

Die SuperCom Suite enthält weitere Beispielprogramme und insbesondere Modbus Server Beispiele, die mehrere Klienten unterstützen.

PDF Dokumente:

- PDF-Dokument mit weiteren Informationen und Bildern zu einigen der enthaltenen Beispielprogramme und

Geschwindigkeit Messungen
- SuperCom-MODBUS-Software-Bibliothek

Eine flexible und portable Bibliothek für die Nutzung in Windows oder Linux Anwendungen*.

* Bitte selektieren Sie die entsprechenden Windows oder Linux Artikelnummern.

Erfahrungen anderer Entwickler mit SuperCom Modbus

... die Anwendung erreicht jetzt kurze Poll-Zeiten!
M.., Deutschland.

Die Einbindung hat funktioniert und die modbus Daten kommen flüssig an!
O.., Deutschland.

Unsere Modbus Anwendung läuft sehr stabil jetzt.
P.., Österreich.

... und jetzt liest und speichert die Anwendung schnell die Daten ab.
Pi.., Schweiz.

MODBUS Bibliothek | MODBUS RTU | MODBUS ASCII | MODBUS TCP | MODBUS over IP | MODBUS Serielle Bibliothek | MODBUS RS-485 | MODBUS Server | MODBUS Beispiel C, C++, C#, Pascal, Delphi, Java, LabView, Visual Basic, VBA | MODBUS C Bibliothek | MODBUS C++ Bibliothek | MODBUS C# Bibliothek | MODBUS Delphi Bibliothek | MODBUS Pascal Bibliothek | MODBUS VB Bibliothek | MODBUS RTU Beispiel | MODBUS C Code Beispiel | MODBUS Server Simulator

---

ADONTEC®
It Simply Works!

Updated on: 2022-09-07 08:22:17
Page generation time: 0.47 sec