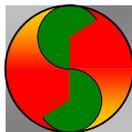


# S3964.DLL

# HANDBUCH

ab Version 1.0  
Stand: 25.05.2005

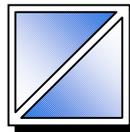
System-Software-Entwicklung  
Dipl.Inform.  
Thomas Schneider



Im Schieb 22 40668 Meerbusch

Tel. 02150 911 747  
Fax 02150 911 748

eMail: [thomas.schneider@sys-thos.de](mailto:thomas.schneider@sys-thos.de)



Wolfgang Pfortner  
Elektronische Baugruppen

Postfach 10 76704 Dettenheim  
Krautstückerweg 13 76706 Dettenheim

Tel. 07247 946060  
Fax 07247 946063

eMail: [wopfo@t-online.de](mailto:wopfo@t-online.de)

|            |   |            |
|------------|---|------------|
| <b>1</b>   | <b>ÜBERSICHT</b>                          | <b>1-1</b> |
| <hr/>      |   |            |
| <b>1.1</b> | <b>DAS PROTOKOLL S3964(R)</b>             | <b>1-1</b> |
| <b>1.2</b> | <b>HINWEISE ZUR INSTALLATION</b>          | <b>1-1</b> |
| <b>1.3</b> | <b>DATEIEN</b>                            | <b>1-2</b> |
| 1.3.1      | VERZEICHNIS DELPHI                        | 1-2        |
| 1.3.2      | VERZEICHNIS DOK                           | 1-2        |
| 1.3.3      | VERZEICHNIS DEMO                          | 1-2        |
| <b>1.4</b> | <b>HINWEISE FÜR PROGRAMMIERER</b>         | <b>1-2</b> |
| 1.4.1      | ALLGEMEINE HINWEISE                       | 1-2        |
| 1.4.2      | HINWEISE FÜR C/C++ - PROGRAMMIERER        | 1-2        |
| 1.4.3      | HINWEISE FÜR VISUAL-BASIC – PROGRAMMIERER | 1-3        |
| <br>       |   |            |
| <b>2</b>   | <b>KONSTANTEN UND TYPEN</b>               | <b>2-1</b> |
| <hr/>      |   |            |
| <b>2.1</b> | <b>RÜCKGABEWERTE DER DLL-FUNKTIONEN</b>   | <b>2-1</b> |
| <b>2.2</b> | <b>MELDUNGEN DER DLL</b>                  | <b>2-1</b> |
| 2.2.1      | MELDUNGSTYP                               | 2-1        |
| 2.2.2      | FEHLERNUMMERN                             | 2-1        |
| 2.2.3      | TYPDEFINITIONEN                           | 2-2        |
| <b>2.3</b> | <b>SONSTIGE KONSTANTEN UND TYPEN</b>      | <b>2-2</b> |
| <br>       |   |            |
| <b>3</b>   | <b>FUNKTIONEN</b>                         | <b>3-1</b> |
| <hr/>      |   |            |
| <b>3.1</b> | <b>PROGRAMMSTART</b>                      | <b>3-1</b> |
| <b>3.2</b> | <b>FUNKTIONSBESCHREIBUNGEN</b>            | <b>3-1</b> |
| 3.2.1      | S3964_INIT                                | 3-1        |
| 3.2.2      | S3964_INITEX                              | 3-1        |
| 3.2.3      | S3964_DONE                                | 3-2        |
| 3.2.4      | S3964_CFGDLG                              | 3-2        |
| 3.2.5      | S3964_XMT                                 | 3-2        |
| 3.2.6      | S3964_RCV                                 | 3-3        |
| 3.2.7      | S3964_CLR                                 | 3-3        |
| 3.2.8      | S3964_ERRSTRING                           | 3-3        |
| 3.2.9      | S3964_ABOUT                               | 3-4        |
| <b>3.3</b> | <b>KONFIGURATION</b>                      | <b>3-5</b> |
| 3.3.1      | SCHNITTSTELLE                             | 3-5        |
| 3.3.2      | PROTOKOLLPARAMETER                        | 3-6        |
| 3.3.3      | ERWEITERTE PARAMETER                      | 3-7        |
| <b>3.4</b> | <b>MELDUNGEN</b>                          | <b>3-8</b> |
| 3.4.1      | INITIALISIERUNG                           | 3-9        |
| 3.4.2      | BLOCK SENDEN                              | 3-10       |
| 3.4.3      | BLOCK EMPFANGEN                           | 3-11       |
| <br>       |   |            |
| <b>4</b>   | <b>DEMO-/TEST-PROGRAMM S3963TEST.EXE</b>  | <b>4-1</b> |
| <hr/>      |   |            |
| <b>4.1</b> | <b>PROGRAMM S3964DEMO.EXE</b>             | <b>4-3</b> |

# 1 Übersicht

Dieses Dokument beschreibt die S3964.DLL zur Kommunikation über die serielle Schnittstelle mit dem Protokoll Siemens 3964(R).

Die DLL ist lauffähig unter Windows 95/98/ME und Windows NT/2000/XP.

Der Implementierung liegt kein offizielles Dokument der Fa. Siemens zugrunde; insoweit wird eine Gewährleistung einer korrekten Funktionalität ausgeschlossen. Auch erfolgt keinerlei Zusicherung in Bezug auf die Verwendbarkeit für einen bestimmten Zweck.

Die folgende Beschreibung der darin enthaltenen Funktionen orientiert sich an der Syntax der Sprache Delphi/PASCAL; es sollte jedoch keine Probleme bereiten, die DLL auch aus anderen Sprachen anzusprechen. Das Aufrufmodell ist, wie von Microsoft empfohlen, „stdcall“.

Bis auf einige Ausnahmen sind alle Routinen der S3964.DLL Funktionen und liefern einen (negativen) Fehler-Code bzw. ein (positives) Funktionsergebnis zurück; Details s. bei der Beschreibung der jeweiligen Funktion.

Die DLL ist thread-sicher, d.h. es können mehrere Threads parallel auf die Schnittstelle zugreifen.

Versionsgeschichte:

Version 1.01: 1. Produktiv-Version

Version 1.02,

Version 1.03: Lizenztext

Version 1.04: Callbackfunktion TS3964CallBack: Aufrufmodell stdcall

## 1.1 Das Protokoll S3964(R)

Die S3964.DLL implementiert das Protokoll S3964(R) der Fa. Siemens. Dieses Protokoll definiert eine serielle Übertragung zwischen zwei Stationen auf der Basis einer RS232-Schnittstelle. Außerdem definiert es sowohl einen zeitlichen Rahmen als auch Möglichkeiten der Priorisierung der beiden Stationen und optional eine Fehlerüberwachung.

Das Protokoll ist block-orientiert, jedoch dürfen die Datenblöcke variable Länge haben. Die Größe eines Datenblocks ist vom Protokoll nicht explizit beschränkt, jedoch ist die Datenlänge implizit durch die maximal zulässige Übertragungszeit für einen Datenblock, und damit abhängig von der Baudrate, eingeschränkt.

Die S3964.DLL arbeitet asynchron, d.h. Datenblöcke können übergeben werden und werden unabhängig vom übergebenden Programm versandt; ebenso werden Datenblöcke von der Gegenstation übernommen. Sobald ein Datenblock versandt oder empfangen wurde, wird dies dem aufrufenden Programm gemeldet (Callback-Funktion oder Windows-Message).

Zusätzlich besteht die Möglichkeit, eine beliebige Anzahl von Zwischenpuffern (Warteschlangen) für das Senden und Empfangen einzurichten.

Neben der Parametrierung der seriellen Schnittstelle und der Protokoll-Optionen bietet die S3964.DLL einige darüber hinaus gehende Möglichkeiten.

**Hinweis:** *Der Implementierung liegt kein offizielles Dokument der Fa. Siemens zugrunde; insoweit wird eine Gewährleistung einer korrekten Funktionalität ausgeschlossen. Auch erfolgt keinerlei Zusicherung in Bezug auf die Verwendbarkeit für einen bestimmten Zweck.*

## 1.2 Hinweise zur Installation

Unter allen Windows-Versionen sind keine Treiber erforderlich. Die DLL kann direkt verwendet werden. Dazu kann sie entweder in das Verzeichnis der Applikation kopiert werden (empfohlen) oder

## Übersicht

in eines der System-Verzeichnisse, die standardmäßig nach DLLs durchsucht werden (%windows%, %windows%\system, %windows%\system32 – nicht empfohlen) oder auch in eines der in der Umgebungsvariablen PATH eingetragenen Verzeichnisse (empfohlen, wenn mehrere Applikationen die DLL verwenden). Alle genannten Verzeichnisse werden beim Laden von DLLs nach der geforderten Datei durchsucht.

### 1.3 Dateien

#### 1.3.1 Verzeichnis Delphi

#### 1.3.2 Verzeichnis Dok

- S3964\_DLL.PDF Diese Datei

#### 1.3.3 Verzeichnis Demo

- S3964.DLL Protokoll-Treiber-DLL
- S3964Test.EXE Test-/DEMO-Programm
- S3964Test.INI S3964-Konfigurationsdaten für das Test-Programm (wird ggf. neu erzeugt)

Source-Code des Demoprogramms; Details s. Kap. 4, „Demo-/Test-Programm S3963Test.EXE“.

### 1.4 Hinweise für Programmierer

#### 1.4.1 Allgemeine Hinweise

- Eine DLL kann prinzipiell unabhängig von der Programmiersprache von jeder Anwendung verwendet werden, sofern die Programmiersprache/-umgebung das Einbinden von DLLs unterstützt. Notfalls kann dazu auf entsprechende Windows-API-Routinen zurückgegriffen werden (s. Hinweise für C/C++ - Programmierer).
- Integer-Typen:  

|          |                                    |
|----------|------------------------------------|
| INTEGER  | unter Win32: 32 Bit mit Vorzeichen |
| Byte     | 1 Byte ohne Vorzeichen (0..255)    |
| ShortInt | 1 Byte mit Vorzeichen (-128..+127) |
| LONGINT  | 32 Bit mit Vorzeichen              |
| WORD     | 16 Bit ohne Vorzeichen             |
| CARDINAL | 32 Bit ohne Vorzeichen             |
- Wahrheitswerte / Datentyp `BOOL`:  
4-Byte-Wert, wobei gilt: `FALSE = 0`, `TRUE = 1`

#### 1.4.2 Hinweise für C/C++ - Programmierer

- Formale Parameter, die durch das Schlüsselwort **VAR** bzw. **CONST** eingeleitet werden, sind aus C/C++-Sicht vom Typ long pointer auf eine Variable vom angegebenen Typ, z.B.

```
FUNCTION func(Handle: INTEGER; cmd: WORD; VAR Value: LONGINT): INTEGER;
```

kann übersetzt werden als

```
INT FUNCTION func(INT Handle, WORD cmd, LPLONG Value);
```

## Übersicht

- Einbinden der DLL in eigene Projekte:  
Da die DLL mit Delphi erstellt wurde existiert keine LIB-Datei um die DLL statisch an ein Projekt zu linken. Dazu gibt es zwei Lösungen:
  1. Entweder greift man auf ein entsprechendes Tool zurück, das eine LIB-Datei aus einer gegebenen DLL generiertoder
  2. Man bindet die DLL dynamisch ins Projekt ein, indem die DLL mit der WinAPI-Funktion `LoadLibrary` geladen wird und die Einsprung-Adressen der Funktionen per WinAPI-Funktion `GetProcAddress` ermittelt werden; die gefundenen Adressen werden dann einer Pointer-Variablen vom Typ der entsprechenden Funktion zugewiesen. Bei Programm-Ende muß die DLL mit `FreeLibrary` wieder freigegeben werden.

### 1.4.3 Hinweise für Visual-Basic – Programmierer

- Formale Parameter, die durch das Schlüsselwort **VAR** bzw. **CONST** eingeleitet werden, werden aus VB-Sicht **ByRef** übergeben, alle anderen **ByVal**, insbesondere gilt dies für String- bzw. PChar-Parameter.
- Die Einbindung einer Funktion aus einer DLL namens Lib.DLL geschieht über eine globale Deklaration, z.B.

```
FUNCTION ExecCfgDlg(AINIFileName: PChar): INTEGER; external 'Lib.DLL';
```

kann übersetzt werden als

```
Declare Function ExecCfgDlg "Lib.DLL" (ByVal AINIFileName AS String) AS  
LONG
```

## 2 Konstanten und Typen

Im folgenden werden alle verwendeten Konstanten und Typen beschrieben. Die PASCAL/Delphi-Deklarationen finden sich auch in der Datei S3964\_ifcTypes.PAS.

### 2.1 Rückgabewerte der DLL-Funktionen

#### CONST

```

s3964ifc_NoError          = 0;   kein Fehler
s3964ifc_NotImplemented = -1;   Funktion (noch) nicht implementiert
s3964ifc_Internal        = -2;   interner Fehler (sollte nicht vorkommen)
s3964ifc_CfgChanged      = -3;   Konfiguration geändert, Initialisierung erforderlich
s3964ifc_InitDone        = -4;   Schnittstelle ist bereits initialisiert
s3964ifc_InitErr         = -5;   Fehler bei der Initialisierung
s3964ifc_DoneErr         = -6;   Fehler beim Beenden / Deinitialisieren
s3964ifc_XmtQueue        = -7;   Sendewarteschlange ist voll
s3964ifc_RcvQueue        = -8;   Empfangswarteschlange ist leer

```

### 2.2 Meldungen der DLL

Über eine Callback-Funktion oder SendMessage / PostMessage im Parameter wParam, Details s.u. s3964\_Init / s3964InitEx bzw. Kap. 3.4, Meldungen:

#### 2.2.1 Meldungstyp

```

CONST // wParam          Meldungen der
s3964Res_Init           = 0;   - Initialisierungsroutine
s3964Res_Rcv            = 1;   - Empfangsroutine
s3964Res_Xmt            = 2;   - Senderoutine
s3964Res_TO             = 3;   - Timeout-Überwachung

// wParam.PostMessage: Bit 7 gesetzt:
s3964Res_Post           = $80;
s3964Res_Mask           = $7F;

```

Fehlermeldungen der S3964-Funktionen über (alternativ):

- Callback-Funktion im Parameter S3964Result.s3964Err
- SendMessage mit lParam = Pointer auf Variable vom Typ TS3964Result
- PostMessage mit lParam = Fehlernr.

#### 2.2.2 Fehlernummern

```

CONST // lParam.PostMessage bzw. S3964Result.s3964Err:

s3964_NoError          = 0;   kein Fehler

s3964Init_COM          = 1;   Öffnen der seriellen Schnittstelle fehlgeschlagen
s3964Init_NAK          = 2;   NAK konnte nicht gesendet werden

s3964Res_CBInit        = 6;   Callback-Funktion wurde installiert
s3964Res_CBDone        = 7;   Callback-Funktion wurde entfernt
s3964Res_HndInit       = 8;   Window-Handle wurde installiert

```

## Funktionen

|                   |       |  |
|-------------------|-------|--|
| s3964Res_HndDone  | = 9;  | Window-Handle wurde entfernt                       |
| s3964Xmt_XmtSTX   | = 11; | STX wurde nicht gesendet / COM nicht offen         |
| s3964Xmt_XmtBlock | = 12; | Fehler beim Versenden des Datenblocks              |
| s3964Xmt_RcvDLE   | = 13; | DLE nicht empfangen / Gegenstelle antwortet nicht  |
| s3964Xmt_RcvSTX   | = 14; | STX während Sendeversuch empfangen / Sendekonflikt |
| s3964Xmt_Count    | = 15; | Anzahl der Sendeversuche überschritten             |
| s3964Xmt_Max      | = 16; | Block zu groß zum Versenden                        |
| s3964Xmt_Size     | = 17; | Blockgröße falsch                                  |
| s3964Xmt_Buf      | = 18; | Sendebuffer voll oder nicht vorhanden              |
| s3964Xmt_TO       | = 19; | Timeout Applikation                                |
| s3964Rcv_STX      | = 21; | ungültiges Zeichen / 1. Zeichen <> STX             |
| s3964Rcv_RcvBlock | = 22; | Fehler beim Empfang des Datenblocks                |
| s3964Rcv_RcvETX   | = 23; | abschließendes DLE und/oder ETX fehlt              |
| s3964Rcv_ChkSum   | = 24; | BCC falsch   |
| s3964Rcv_Count    | = 25; | Anzahl der Empfangsversuche überschritten          |
| s3964Rcv_Size     | = 27; | Blockgröße falsch                                  |
| s3964Xmt_Buf      | = 28; | Empfangsbuffer voll oder nicht vorhanden           |
| s3964Rcv_TO       | = 29; | Timeout Gegenstelle - Kabel/SPS prüfen             |

### 2.2.3 Typdefinitionen

Typdefinition für die Callback-Funktion bzw. SendMessage; die DLL trägt in diese Struktur die Ergebnisse der letzten Aktion ein:

```
TYPE TS3964Result = RECORD
    Block:      Pointer;      Pointer auf den empfangenen / versandten Block
    BlockSize:  CARDINAL;    Größe dieses Blocks
    s3964Err:   INTEGER;     Fehlermeldung der S3964-Funktion
END;
PS3964Result = ^TS3964Result;
```

Callback-Funktion für Meldungen der S3964-Funktionen an die Applikation; zum Rückgabewert (Ergebnis) s. Kap. 3.4!

```
TYPE TS3964CallBack = FUNCTION(wParam: INTEGER; CONST S3964Result:
TS3964Result): INTEGER; stdcall;
```

## 2.3 Sonstige Konstanten und Typen

Konstanten für die Funktion s3964\_Clr:

```
CONST s3964Clr_RcvQueue = 1;
        s3964Clr_XmtQueue = 2;
        s3964Clr_Queuees = 3;      = s3964Clr_RcvQueue + s3964Clr_XmtQueue!
```

Konstanten für die Funktion s3964\_InitEx – Priorität der Station:

```
CONST s3964Low  = 0;
        s3964High = 1;
```

## 3 Funktionen

Im folgenden werden alle implementierten Funktionen beschrieben. Die PASCAL/Delphi-Deklarationen finden sich auch in der Datei S3964\_ifc.PAS.

### 3.1 Programmstart

Bei Programmstart bzw. beim Laden versucht die S3964.DLL Konfigurationsdaten aus einer Initialisierungsdatei zu lesen; die Initialisierungsdatei muß sich im Verzeichnis der Applikation befinden und die Datei-Endung „\*.ini“ und den gleichen Namen wie die Applikation haben. Wird diese Datei nicht gefunden, so werden Standardwerte eingesetzt. Details dazu s. s3964\_CfgDlg.

### 3.2 Funktionsbeschreibungen

#### 3.2.1 s3964\_Init

**FUNCTION** s3964\_Init(AWindowHandle: THandle; AInfo: INTEGER): INTEGER;

**Beschreibung** Diese Funktion initialisiert die serielle Schnittstelle mit den aktuell gesetzten Schnittstellen-Parametern; bei Programmstart sind das die Parameter aus der INI-Datei bzw. die Standard-Parameter.

#### Parameter

AWindowHandle Fenster-Handle, an das Meldungen der Routinen versandt werden; wird AWindowHandle = 0 übergeben, so werden keine Meldungen versandt

AInfo Dieser Parameter legt weitere Parameter abhängig von AWindowHandle fest:  
 AWindowHandle <> 0: AInfo = Windows-Msg-Nummer  
 AWindowHandle = 0: AInfo = Adresse einer Callback-Funktion vom Typ TS3964CallBack

**Funktionsergebnis** Schnittstellen-Fehlercode s3964ifc\_XXX bzw. Nr. der seriellen Schnittstelle

Weitere Details zu den Meldungen s. Kap. 3.4 – Meldungen

#### 3.2.2 s3964\_InitEx

**FUNCTION** s3964\_InitEx(AWindowHandle: THandle; AInfo: INTEGER; ACOM\_IFC: INTEGER; APriority: INTEGER): INTEGER;

**Beschreibung** Diese Funktion initialisiert die serielle Schnittstelle mit den aktuell gesetzten Schnittstellen-Parametern mit Ausnahme der Schnittstellennr. und der Priorität; bei Programmstart sind das die Parameter aus der INI-Datei bzw. die Standard-Parameter.

#### Parameter

AWindowHandle Fenster-Handle, an das Meldungen der Routinen versandt werden; wird AWindowHandle = 0 übergeben, so werden keine Meldungen versandt

AInfo Dieser Parameter legt weitere Parameter abhängig von AWindowHandle fest:  
 AWindowHandle <> 0: AInfo = Windows-Msg-Nummer  
 AWindowHandle = 0: AInfo = Adresse einer Callback-Funktion vom Typ TS3964CallBack

## Funktionen

`ACOM_IFC` Nummer der seriellen Schnittstelle

`APriority` Priorität der S3964-Station

**Funktionsergebnis** Schnittstellen-Fehlercode `s3964ifc_XXX` bzw. Nr. der seriellen Schnittstelle

**Hinweis:** *ACOM\_IFC und APriority überschreiben zwar die aktuellen Konfigurationsdaten, diese werden aber nicht in die INI-Datei geschrieben!*

Weitere Details zu den Meldungen s. Kap. 3.4 – Meldungen

### 3.2.3 s3964\_Done

**FUNCTION** `s3964_Done`: INTEGER;

**Beschreibung** Beendet die Kommunikation und gibt die Schnittstelle wieder frei.

**Parameter** keine

**Funktionsergebnis** Schnittstellen-Fehlercode `s3964ifc_XXX`

### 3.2.4 s3964\_CfgDlg

**FUNCTION** `s3964_CfgDlg`(`AINIFileName`: PChar): INTEGER;

**Beschreibung** Ruft einen Dialog zur Einstellung der Schnittstellen- und Protokollparameter auf; beim Beenden des Dialogs mit der Schaltfläche „OK“ werden die Daten in der Datei `AINIFileName` gespeichert. Die Schnittstelle wird jedoch nicht neu initialisiert; dies muß im Anschluß explizit mit der Funktion `s3964_Init` geschehen. Weitere Details s. Kap. 3.3 – Konfiguration.

#### Parameter

`AINIFileName` Name der Datei für die Initialisierungsdaten

**Funktionsergebnis** Schnittstellen-Fehlercode `s3964ifc_XXX`

### 3.2.5 s3964\_Xmt

**FUNCTION** `s3964_Xmt`(`Block`: Pointer; `BlockSize`: CARDINAL): INTEGER;

**Beschreibung** Diese Funktion schreibt den nächsten Datenblock in den Sendebuffer; sobald die Übertragung des vorherige Blocks beendet ist, wird der nächste Block aus dem Sendebuffer übertragen.

#### Parameter

`Block` Pointer auf die zu übertragenden Daten

`BlockSize` Größe des Block.

**Funktionsergebnis** Schnittstellen-Fehlercode `s3964ifc_XXX`

## Funktionen

### 3.2.6 s3964\_Rcv

**FUNCTION** s3964\_Rcv(Block: Pointer; BlockSize: CARDINAL): INTEGER;

**Beschreibung** Diese Funktion kopiert einen Datenblock aus dem Empfangsbuffer und löscht ihn im Empfangsbuffer.

#### Parameter

**Block** Pointer auf einen Buffer zur Aufnahme des Datenblocks; wird hier der Wert NIL (0) übergeben, so wird lediglich die Größe des nächsten Datenblocks als Funktionsergebnis zurückgegeben, der Block selbst verbleibt im Empfangsbuffer. Damit kann in einem 1. Aufruf die Größe des nächsten Datenblocks festgestellt werden; anschließend kann die Applikation den benötigten Speicher bereitstellen und in einem 2. Aufruf den Datenblock tatsächlich holen.

**BlockSize** Größe des Blocks.

**Funktionsergebnis** Schnittstellen-Fehlercode s3964ifc\_XXX bzw. die tatsächliche Größe des Datenblocks.

### 3.2.7 s3964\_Clr

**FUNCTION** s3964\_Clr(Action: INTEGER): INTEGER;

**Beschreibung** Diese Funktion löscht Sende- bzw. Empfangsbuffer.

#### Parameter

**Action** Kombination der Konstanten:  
s3964Clr\_RcvQueue: löscht den Empfangsbuffer  
s3964Clr\_XmtQueue: löscht den Sendebuffer  
s3964Clr\_Queue = s3964Clr\_RcvQueue + s3964Clr\_XmtQueue:  
löscht Sende- und Empfangsbuffer

**Funktionsergebnis** Schnittstellen-Fehlercode s3964ifc\_XXX

### 3.2.8 s3964\_ErrString

**FUNCTION** s3964\_ErrString(Buffer: PChar; BufferSize: CARDINAL; AS3964Err: INTEGER): INTEGER;

**Beschreibung** Liefert eine Klartextmeldung für die Fehlernummer AS3964Err

#### Parameter

**Buffer** Pointer auf einen Buffer zur Aufnahme des Strings; wird hier der Wert NIL (0) übergeben, so wird lediglich die Länge der Fehlermeldung als Funktionsergebnis zurückgegeben. Damit kann in einem 1. Aufruf die Länge des Strings festgestellt werden; anschließend kann die Applikation den benötigten Speicher bereitstellen und in einem 2. Aufruf den String tatsächlich holen.

**BufferSize** Größe des String-Buffers Buffer.

**AS3964Err** Nummer der Fehlermeldung

## Funktionen

**Funktionsergebnis** Schnittstellen-Fehlercode `s3964ifc_XXX` bzw. die tatsächliche Größe des Strings.

### 3.2.9 `s3964_About`

**FUNCTION** `s3964_About`: INTEGER;

**Beschreibung** Stellt einen Copyright-Fenster dar.

**Parameter** keine

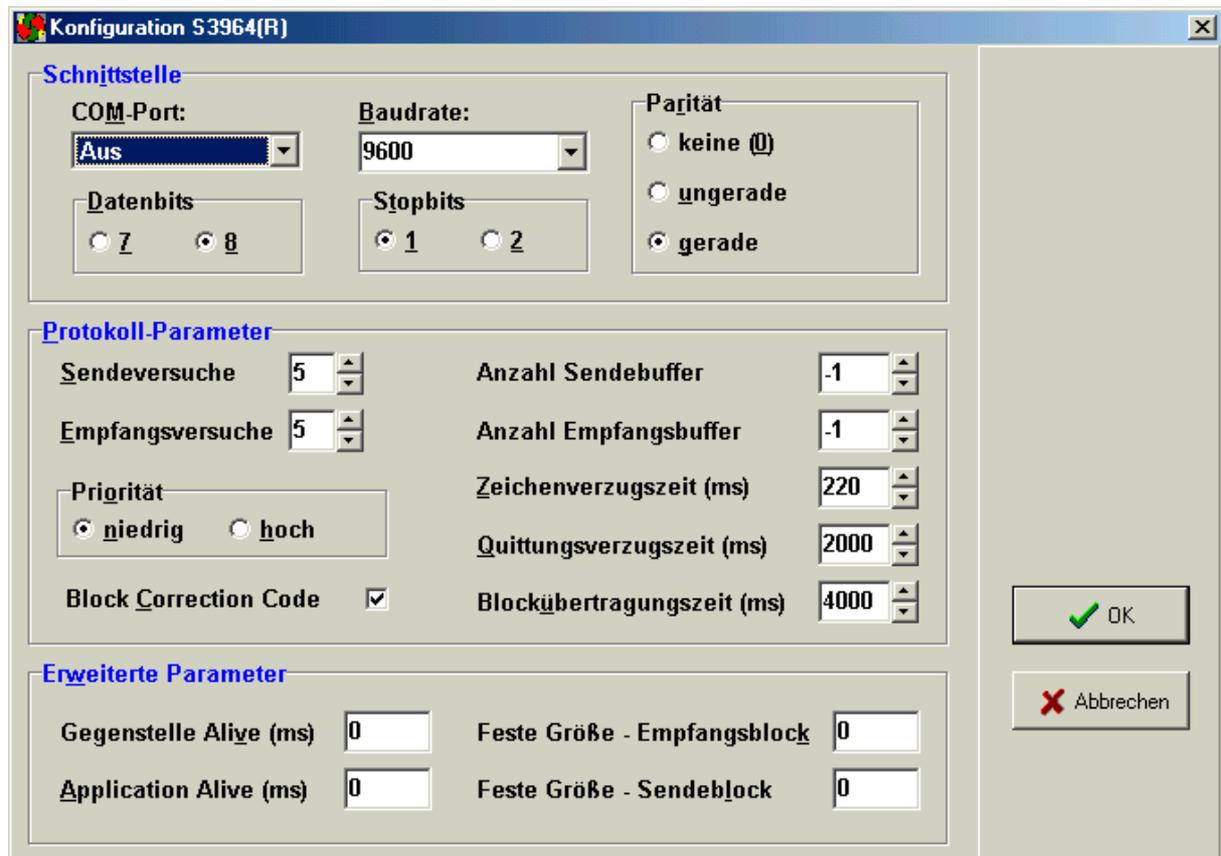
**Funktionsergebnis** Schnittstellen-Fehlercode `s3964ifc_XXX`

### 3.3 Konfiguration

Der Konfigurationsdialog teilt sich in mehrere Teile:

1. Schnittstelle: Nr. und Parameter der seriellen Schnittstelle
2. Protokoll-Parameter: Parameter des S3964-Protokolls
3. Erweiterte Parameter: weitere Möglichkeiten der Konfiguration

Die im folgenden genannten bzw. im Bild sichtbaren Standardwerte werden auch für die Initialisierung der S3964.DLL verwendet, wenn keine INI-Datei gefunden wird.



#### 3.3.1 Schnittstelle

Auswahl und Einstellung der physikalischen Eigenschaften der seriellen Schnittstelle:

|                  |  |
|------------------|--|
| <b>COM-Port</b>  | Es kann eine der verfügbaren seriellen Schnittstellen des PC ausgewählt werden. Angezeigt werden alle vorhandenen und system-konform verfügbaren Schnittstellen.<br><b>Standardwert</b> Aus (keine serielle Schnittstelle) |
| <b>Baudrate</b>  | Auswahl der Baudrate<br><b>Standardwert</b> 9600   |
| <b>Datenbits</b> | Anzahl der übertragenen Datenbits<br><b>Standardwert</b> 8   |
| <b>Stopbits</b>  | Anzahl der Stopbits<br><b>Standardwert</b> 1   |

## Funktionen

**Parität** Paritätsbits  
**Standardwert** gerade Parität

### 3.3.2 Protokollparameter

Definition der Parameter des S3964(R)-Protokolls

**Sendeversuche** Anzahl der Sende-Versuche; danach wird die Übertragung mit einer Fehlermeldung abgebrochen.  
**Standardwert** 5

**Empfangsversuche** Anzahl der Versuche, eine Empfangsaufforderung der Gegenstelle zu bedienen; danach wird ein Fehler gemeldet.  
**Standardwert** 5

**Priorität** Steuert das Verhalten bei gleichzeitigem Senden beider Stationen:  
**hoch** Die Station wiederholt den Sendeversuch, ohne die Empfangsaufforderung der Gegenstation zu quittieren.  
**niedrig** Die Station bedient zunächst die Empfangsaufforderung der Gegenstation und versucht anschließend erneut zu senden.  
Daraus wird klar, daß eine Station die Priorität „**hoch**“, die andere die Priorität „**niedrig**“ haben muß.

**Block Correction Code (BCC)** Fehler-Überwachung **aus/ein** (Protokoll 3964 bzw. 3964R). Das Protokoll kann sowohl ohne als auch mit Fehler-Überwachung arbeiten. Beide Stationen müssen aber natürlich auf den gleichen Modus eingestellt werden.

**Anzahl Sendebuffer** Anzahl der Sende-Buffer; so viele Datenblöcke kann die S3964.DLL zwischenspeichern.  
**Standardwert** -1 (unbegrenzt)

**Hinweis:** *Da die Datenübertragung asynchron erfolgt, muß zumindest 1 Sendebuffer zur Verfügung stehen, da sonst die Funktion s3964\_Xmt den Block dort nicht eintragen kann...*

**Anzahl Empfangsbuffer** Anzahl der Empfangs-Buffer; so viele Datenblöcke kann die S3964.DLL zwischenspeichern.  
**Standardwert** -1 (unbegrenzt)

**Hinweis:** *Steht kein Empfangsbuffer zur Verfügung, so erfolgen Meldungen an die Applikation synchron über `SendMessage` statt über `PostMessage`!*

**Zeichenverzugszeit** Max. Zeitintervall zwischen zwei empfangenen Zeichen; wird diese Zeit überschritten, wird der Empfangsversuch mit einem Fehler beendet.  
**Standardwert** 220ms

**Quittungsverzugszeit** Max. Zeit, die bis zum Eintreffen der Empfangsquittung der Gegenstelle verstreichen darf; danach wird der Sendeversuch abgebrochen.  
**Standardwert** 500ms (ohne BCC)  
2000ms (mit BCC)

**Blockübertragungszeit** Max. Zeit, innerhalb derer ein Datenblock übertragen werden muß.  
**Standardwert** 4000ms

**Hinweis:** *Die maximal zulässige Übertragungszeit beschränkt implizit zusammen mit der Baudrate die maximal zulässige Größe eines Datenblocks!*

### 3.3.3 Erweiterte Parameter

Diese Parameter sind nicht Teil des S3964-Protokolls, bilden jedoch eine sinnvolle und nützliche Erweiterung.

**Gegenstelle Alive** Es wird überwacht, daß die Gegenstelle in regelmäßigen Abständen versucht, Daten zu senden.  
**Standardwert** 0 ms (keine Überwachung)

**Application Alive** Es wird überwacht, daß die Applikation, die die S3964.DLL benutzt, in regelmäßigen Abständen versucht, Daten zu senden.  
**Standardwert** 0ms (keine Überwachung)

**Feste Größe Empfangsblock** Falls erforderlich, kann die Größe eines Empfangsblocks fest vorgegeben bzw. überwacht werden.  
**Standardwert** 0 (keine Überwachung / Vorgabe)

**Feste Größe Sendeblock** Falls erforderlich, kann die Größe eines Sendeblocks fest vorgegeben bzw. überwacht werden.  
**Standardwert** 0 (keine Überwachung / Vorgabe)

**Hinweis:** *Die durch die maximal zulässige Übertragungszeit beschränkte Größe eines Datenblocks ist die absolute Obergrenze und kann auch durch die Einstellung einer festen Blockgröße nicht außer Kraft gesetzt werden.*

### 3.4 Meldungen

Die internen Funktionen der S3964.DLL melden Ereignisse per Callback-Funktion oder SendMessage / PostMessage an die Applikation. Callback-Funktion bzw. Fenster-Handle werden bei der Initialisierung (s. s3964\_Init bzw. s3964\_InitEx) an die S3964.DLL übergeben. Abhängig von den Aufruf-Parametern wird nur einer der beiden Mechanismen (Callback-Funktion oder Send- / PostMessage) verwendet!

Je nachdem, welche Meldungsfunktion verwendet wird, haben deren Parameter folgende Bedeutung (s. a. Kap. 2, Konstanten und Typen):

Callback-Funktion:

- wParam: Konstante s3964Res\_XXX
- S3964Result: Variable vom Typ TS3964Result

SendMessage-Funktion:

- wParam: Konstante s3964Res\_XXX
- lParam: Pointer auf eine Variable vom Typ TS3964Result

PostMessage-Funktion:

- wParam: Konstante s3964Res\_XXX
- lParam: Fehler-Nummer

#### **Hinweise:**

- Der Parameter wParam definiert, welche Funktion die Meldung veranlaßt hat (Init, Rcv, Xmt oder Timeout)
  - Falls nicht anders angegeben, gilt für eine Variable vom Typ TS3964Result
    - Block = NIL
    - BlockSize = 0
    - bzw.
    - Block <> NIL
    - BlockSize = Länge von Block
- NB:** Das Feld Block ist ein Pointer, d.h. die DLL trägt in dieses Feld die DLL-interne Adresse des Datenblocks ein!
- Die Variante über eine Windows-Message ist vorzuziehen, da dann Meldungen i.d.R. asynchron über PostMessage erfolgen; eine Callback-Funktion bzw. SendMessage können die DLL-interne Verarbeitung verzögern!

### 3.4.1 Initialisierung

Die Initialisierung beim Aufruf der Funktion `s3964_Init` bzw. `s3964_InitEx` läuft in folgenden Phasen ab:

1. Initialisierung der Meldungsfunktion:

- Vor dem Überschreiben einer evtl. bereits vorhandenen Callback-Funktion, wird diese mit folgenden Parametern aufgerufen:  
`wParam = s3964Res_Init`  
`S3964Result.s3964Err = s3964Init_CBInit`
- Danach wird die neu übergebene Callback-Funktion installiert und sofort mit folgenden Parametern aufgerufen:  
`wParam = s3964Res_Init`  
`S3964Result.s3964Err = s3964Init_CBDone`
- Vor dem Überschreiben eines evtl. Fenster-Handle, wird die Funktion `SendMessage` mit folgenden Parametern aufgerufen:  
`wParam = s3964Res_Init`  
`lParam = Pointer auf S3964Result:`  
`S3964Result.s3964Err = s3964Init_CBInit`
- Danach wird das neu übergebene Fensterhandle installiert und `SendMessage` mit folgenden Parametern aufgerufen:  
`wParam = s3964Res_Init`  
`lParam = Pointer auf S3964Result:`  
`S3964Result.s3964Err = s3964Init_CBDone`

2. Initialisierung der seriellen Schnittstelle

Tritt ein Fehler auf, so wird die zuvor installierte Meldungsfunktion aufgerufen:

- Callback-Funktion:  
`wParam = s3964Res_Init`  
`S3964Result.s3964Err = s3964Init_COM`  
`S3964Result.Block = String „COMx:“, x = Nr. der seriellen Schnittstelle`

bzw.

- `PostMessage:`  
`wParam = s3964Res_Init`  
`lParam = s3964Init_COM`

3. Initialisieren des Protokolls

Tritt ein Fehler auf, so wird die zuvor installierte Meldungsfunktion aufgerufen:

- Callback-Funktion:  
`wParam = s3964Res_Init`  
`S3964Result.s3964Err = s3964Init_NAK`  
`S3964Result.Block = Adresse des Strings „COMx:“, x = Nr. der seriellen Schnittstelle`

bzw.

- `PostMessage:`  
`wParam = s3964Res_Init`  
`lParam = s3964Init_NAK`

4. Erst jetzt kehrt die Funktion `s3964_Init` bzw. `s3964_InitEx` zurück

### 3.4.2 Block senden

Wurde von der DLL ein Datenblock versandt, so wird die installierte Meldungsfunktion mit folgenden Parametern aufgerufen:

- **Callback-Funktion:**  
wParam = s3964Res\_Xmt  
S3964Result.s3964Err = s3964Xmt\_XXX  
S3964Result.Block = **Pointer auf den versandten Block**  
**Rückgabewert der Callback-Funktion:**  
Result = 0: Der Block wird aus dem Sendebuffer entfernt  
Result = 1: Der Block wird erneut ans Ende der Sendebuffers geschrieben  
Result = 2: Der Block wird vorne in den Sendebuffer geschrieben → er wird direkt als nächstes versandt

bzw.

- **PostMessage (wenn Sendebuffer zur Verfügung stehen)**  
wParam = s3964Res\_Xmt  
lParam = s3964Xmt\_XXX

bzw.

- **SendMessage (wenn keine Sendebuffer zur Verfügung stehen)**  
wParam = s3964Res\_Xmt  
lParam = **Pointer auf S3964Result:**  
**Rückgabewert der Funktion SendMessage:**  
S3964Result.s3964Err = s3964Xmt\_XXX  
S3964Result.Block = **Pointer auf den versandten Block**  
Result = 0: Der Block wurde von der Fensterfunktion verarbeitet  
**Andere Werte für Result sind nicht zulässig**

### 3.4.3 Block empfangen

Wurde ein Datenblock empfangen, so wird die installierte Meldungsfunktion mit folgenden Parametern aufgerufen:

- **Callback-Funktion:**  
wParam = s3964Res\_Rcv  
S3964Result.s3964Err = s3964Rcv\_XXX  
S3964Result.Block = **Pointer auf den empfangenen Block**  
**Rückgabewert der Callback-Funktion:**  
Result = 0: Der Block wird nicht in den Empfangsbuffer eingetragen  
Result = 1: Der Block wird in den Empfangsbuffer eingetragen
- bzw.
- **PostMessage (wenn Empfangsbuffer zur Verfügung stehen)**  
wParam = s3964Res\_Xmt  
lParam = s3964Rcv\_XXX  
**Der Block wird in den Empfangsbuffer eingetragen**
- bzw.
- **SendMessage (wenn keine Empfangsbuffer zur Verfügung stehen)**  
wParam = s3964Res\_Rcv  
lParam = **Pointer auf S3964Result:**  
**Rückgabewert der Funktion SendMessage:**  
S3964Result.s3964Err = s3964Rcv\_XXX  
S3964Result.Block = **Pointer auf den empfangenen Block**  
Result = 0: Der Block wird nicht in den Empfangsbuffer eingetragen  
Result = 1: Der Block wird in den Empfangsbuffer eingetragen

Wurde ein empfangener Datenblock in den Empfangsbuffer eingetragen, so muß er mit der Funktion s3964\_Rcv gelesen werden!

## 4 Demo-/Test-Programm S3963Test.EXE

Das Demo-Programm S3963Test.EXE dient dazu, einige Funktionen der S3964.DLL zu demonstrieren. Das Programm ist lauffähig unter Windows 9x/ME/NT/2k. Der Quelltext steht als Delphi-Programm ebenfalls zur Verfügung. Folgende Dateien enthalten den Quelltext für das Demo-Programm:

S3963Test.DPR Delphi-Projektdatei  
frm\_S3963Test.PAS Hauptfenster des Programms; die DLL-Aufrufe befinden sich jeweils in den Routinen, die bei einem Click auf das entsprechende Fensterelement aufgerufen werden; diese tragen jeweils die Namen

```
procedure TfrmS3964Test.XXXXXClick(Sender: TObject);
```

wobei das XXXXX das jeweilige Fenster-Element kennzeichnet. Die Zuordnung sollte aus den Namen ersichtlich sein (z.B. heißt die Click-Routine für den Init-Button

```
procedure TfrmI332.btnInitClick(Sender: TObject);
```

frm\_S3963Test.DFM Hauptfenster des Programms, Delphi-Ressourcen-Datei (wird von der Delphi-Entwicklungs-Umgebung erzeugt und verwaltet).  
S3964\_IFCTypes.PAS Konstanten und Typdefinitionen  
S3964\_IFC.PAS Interface zur S3964.DLL

Die übrigen Dateien werden von der Delphi-Entwicklungsumgebung erzeugt bzw. verwaltet.

**Hinweis:** *Die Auswertung der Funktionsergebnisse erfolgt durch die Funktion `s3964FktRes`. Diese erzeugt ggf. eine Fehlermeldung und liefert ansonsten das Funktionsergebnis des Aufrufs zurück.*

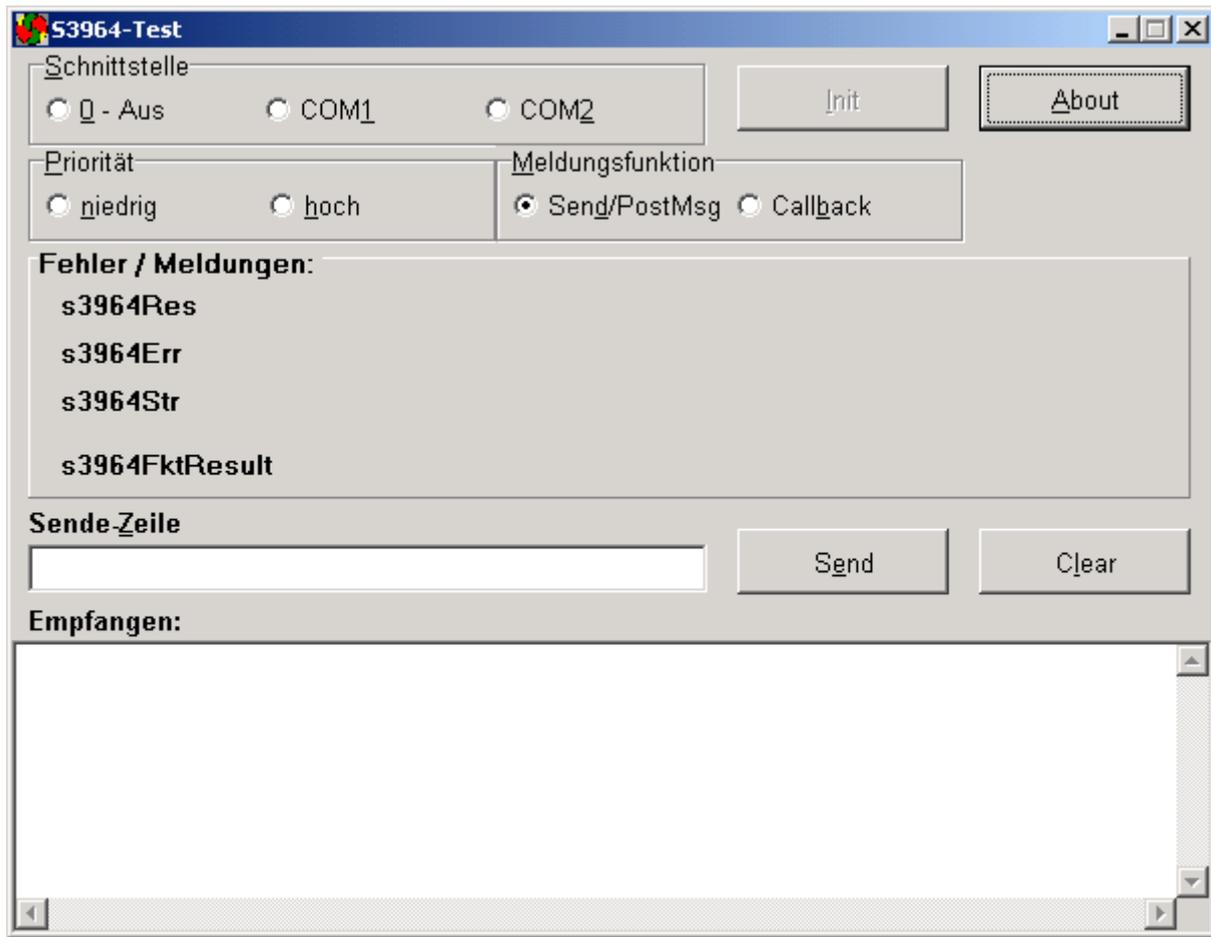
*Delphi stellt weiterhin einen komfortablen Mechanismus zur Verarbeitung von Windows-Messages zur Verfügung; für unseren Zweck wird die Methode*

```
PROCEDURE wmS3964Msg(VAR Msg: TMessage); message wm_S3964Msg;
```

*deklariert, die die Windows-Message mit der Nr. `wm_S3964Msg` verarbeitet. Diese ist als Konstante*

```
CONST wm_S3964Msg = WM_USER;
```

*vereinbart. Im Parameter `Msg` werden dabei die aktuellen Werte für `wParam` und `lParam` übergeben; das Ergebnis eines `SendMessage`-Aufrufs wird in `Msg.Result` zurückgegeben.*



**Schnittstelle** Im Feld Schnittstelle kann eine serielle Schnittstelle ausgewählt werden.

**Priorität** In diesem Feld kann die Priorität der S3964-Station festgelegt werden.

*Hinweis:* Diese beiden Einstellungen überschreiben ggf. die Konfigurationsdaten!

**Meldungsfunktion** In diesem Feld kann die Art der Rückmeldung festgelegt werden.

**Init** `procedure TfrmI332.btnInitClick`  
Der Button ruft die Funktion `s3964_InitEx` mit den ausgewählten Parametern für die Schnittstelle und die Priorität auf. Er ist nur aktiv, wenn auch tatsächlich gültige Werte gewählt wurden.  
Wurde im Feld **Meldungstyp** "Send/PostMsg ausgewählt", so wird als WindowHandle das Fensterhandle zusammen mit der Msg-Nr. `wm_S3964Msg = WM_USER` übergeben.  
Wurde als **Meldungstyp** "Callback ausgewählt", so wird `WindowHandle = 0` und die Adresse der Funktion `S3964Callback` übergeben.

**Cfg** `procedure TfrmI332.btnCfgClick`  
Ruft den Konfigurationsdialog auf

**About** `procedure TfrmI332.btnAboutClick`  
Ruft die Copyright-Meldung auf.

**Sende-Zeile** Die eingegebenen Zeichen werden als (ASCII-)Datenblock versandt

**Send** `procedure TfrmI332.btnSendClick`  
Dieser Button sendet den in der **Sende-Zeile** eingegebenen Text

**Clr**            **procedure** TfrmI332.btnClrClick  
Dieser Button löscht sowohl Sende- und Empfangsbuffer als auch das Empfangsfeld.

**Fehler / Meldungen**   Hier werden die Meldungen der Callback-Funktion bzw. von SendMessage und PostMessage sowie die Ergebnisse der Funktions-Aufrufe dargestellt.

**s3964Res:**    Meldungstyp (Konstante s3964Res\_XXX)  
**s3964Err:**    Fehlernr.  
**s3964Str:**    empfangener bzw. versandter Block  
**s3964FktRes:** Ergebnis des letzten Aufrufs einer S3964.DLL-Funktion (s3964\_XXX)

**Empfangen**    In diesem Feld werden die empfangenen Daten dargestellt.

## 4.1            **Programm S3964Demo.EXE**

Das Programm entspricht S3964Test.EXE mit folgenden Einschränkungen:

- Es werden die Standardeinstellungen verwendet (insbesondere für die serielle Schnittstelle: 8E1, 9600 Baud).
- Konfiguration außer COM-Schnittstelle und Priorität ist nicht möglich; der Button "**Cfg**" ist nicht verfügbar.
- Die S3964.DLL wird nicht benötigt; die notwendigen Funktionen der DLL sind im Programm selbst enthalten und mit den DLL-Funktionen identisch.