

2025

# YMLIB API Referenzen SER

Version: 1.0



## YMLib API Referenz für SER Module (Windows)

---

### Inhaltsverzeichnis

1	YMLIB API Referenzen SER .....	2
2	Klasse YMRS232 .....	2
3	Erklärung der Funktionen und Methoden.....	4
4	Urheberrechte und Marken .....	5

## 1 YMLIB API Referenzen SER

Dank der standardisierten RS232-Schnittstelle benötigen die Yamutec<sup>®</sup> RS232-Module keine spezielle Treiberinstallation. Die Module können direkt mit gängigen Betriebssystemen und seriellen Schnittstellenprogrammen verwendet werden.

Die YMLib-API ermöglicht die individuelle Ansteuerung der Yamutec<sup>®</sup> RS232-Module über VB6, VB.NET, C# oder LabView und lässt sich flexibel an Ihren Anwendungsfall anpassen.

Detaillierte Programmierbeispiele erleichtern die Integration und zeigen die vollständige Umsetzung der seriellen Kommunikation mit dem Modul.

Um mit der YMLIB.dll die SER-Module anzusprechen, muss das Modul in den Progmodus versetzt werden. Dazu wird am Modul der Schalter 1 des DIP-Schalters auf 0 gesetzt.



## 2 Klasse YMRS232

YMRS232 ist die Hauptklasse in der YMLib.dll-Datei, um die Yamutec<sup>®</sup> SER-Module anzusprechen.

### 1. Methoden der Klasse YMRS232

#### **TIMEOUT\_INTERVAL\_AND\_CHANNEL\_X()**

Schaltet die ausgewählten Ausgänge eines SER-REL-Moduls ein oder aus, wenn sich das Gerät im Timeout-Status befindet.

#### **TIMEOUT\_INTERVAL\_AND\_CHANNEL\_HYBRIDX**

Schaltet die ausgewählten Ausgänge eines SER-HYBRID-Moduls ein oder aus, wenn sich das Gerät im Timeout-Status befindet.

#### **TIMEOUT\_DISABLE()**

Deaktiviert den Timeout-Status.

#### **SEND\_STATUS\_X()**

Sendet ein Schaltzustand an ein SER-REL Modul

#### **SEND\_STATUS\_HYBRIDX()**

Sendet ein Schaltzustand an ein SER-HYBRID Modul

#### **HW\_RESET()**

Setzt das Modul zurück.

#### **CALL\_ABOUT\_BOX()**

Ruft den YMLib.dll Version auf

## 2. Eigenschaften der YMRS232

### **YM\_COM\_Ports**

Gibt ein Array von Strings zurück, das die Namen aller verfügbaren COM-Ports enthält. Jedes Element dieses Arrays ist der Name eines erkannten COM-Ports auf dem Computer (z. B. "COM1", "COM2", "COM3" usw.). Falls keine seriellen Ports verfügbar sind, gibt die Methode ein leeres Array zurück.

### **YM\_SerialNR**

Gibt die Seriennummer des angeschlossenen SER-Moduls als ein String zurück.

### **YM\_TimeOutTriggered**

Gibt ein Byte mit dem Wert 1 zurück, wenn sich das SER-Modul im Timeout-Modus befindet, oder eine 0, wenn das Modul nicht im Timeout-Modus ist.

### **YM\_TimeOutActive**

Gibt ein Byte mit dem Wert 1 zurück, wenn Timeout aktiviert ist oder eine 0, wenn Timeout deaktiviert ist.

### **YM\_TimeoutStatus**

Gibt ein Byte mit dem Wert 1 zurück, wenn Timeout auslöst.

### **YM\_Con\_DeviceName**

Gibt den Modellnamen in String zurück

### **YM\_OUTPUT1 ... YM\_OUTPUT8**

Die Ausgangswerte werden kontinuierlich in eine Variable geschrieben.

### **YM\_InPut1 ... YM\_InPut8**

Die Eingangswerte werden kontinuierlich in eine Variable geschrieben.

### **YM\_SecIndex**

Übermittelt den Indexwert des Timeout-Zeitintervalls.

### **YM\_TimeOutChecked**

Gibt ein Boolean mit dem Wert **True** zurück, wenn Timeout aktiviert ist oder ein **False**, wenn Timeout deaktiviert ist.

## 3. Funktionen der YMRS232

### **Connect(ComPort\_Name, BaudRate)**

Die Funktion stellt eine serielle Verbindung mit dem ausgewählten COM-Port und der Baudrate her.

### **Is\_Port\_Open()**

Gibt ein Boolean mit dem Wert **True** zurück, wenn COM-Port geöffnet ist

### **Close\_RS232\_Port()**

Gibt ein Boolean mit dem Wert **True** zurück, wenn COM-Port geschlossen ist

### 3 Erklärung der Funktionen und Methoden

#### Schaltzustand an das Modul senden

Je nachdem, welches SER-Modul angesprochen werden soll, muss für das Senden die richtige Methode ausgewählt werden.

#### SER-REL Module

SER Modul mit 8 Ausgänge  
SEND\_STATUS\_8(SendeByte1)

SER Modul mit 16 Ausgänge  
SEND\_STATUS\_16(SendeByte1, SendeByte2)

SER Modul mit 32 Ausgänge  
SEND\_STATUS\_32(SendeByte1, SendeByte2, SendeByte3, SendeByte4)

SER Modul mit 64 Ausgänge  
SEND\_STATUS\_64(SendeByte1, SendeByte2, SendeByte3, SendeByte4, SendeByte5, SendeByte6, SendeByte7, SendeByte8)

#### Hybrid Module

SER Modul mit 8 Ausgänge und 8 Eingänge  
SEND\_STATUS\_HYBRID8(SendeByte1)

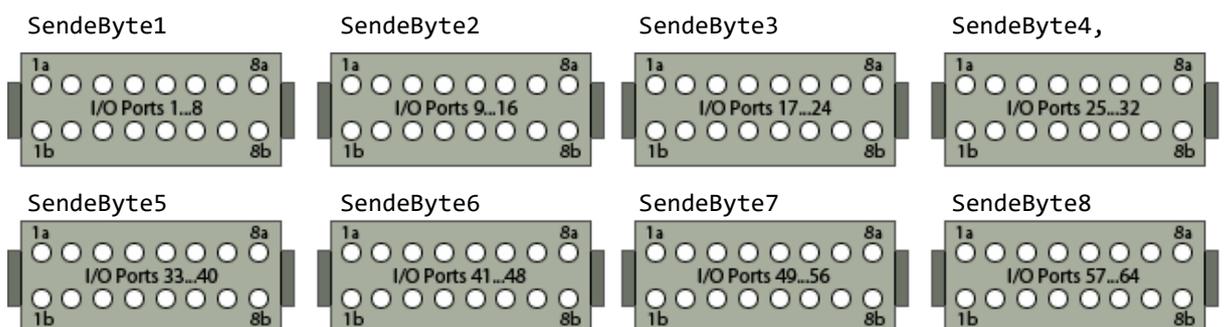
SER Modul mit 16 Ausgänge und 16 Eingänge  
SEND\_STATUS\_HYBRID16(SendeByte1, SendeByte2)

SER Modul mit 32 Ausgänge und 32 Eingänge  
SEND\_STATUS\_HYBRID32(SendeByte1, SendeByte2, SendeByte3, SendeByte4)

#### Parameter SendeByte1

Die Ausgangszustände des Moduls werden kontinuierlich ausgelesen und in der Variablen **Dim SendeByte1 As Byte** gespeichert. Zum Schalten der Ausgänge muss daher nur der Zustand geändert werden, der ein- oder ausgeschaltet werden soll.

Beispiel: Der Variablenwert SendeByte1 = 77 bedeutet, dass die Ausgänge 1, 3, 4 und 7 eingeschaltet sind. Möchte man nun nur Ausgang 3 ausschalten, muss die entsprechende Bit-Wertigkeit (4) vom aktuellen Wert 77 subtrahiert werden. Der neue Wert 73 wird dann an das Modul gesendet.



## 4 Urheberrechte und Marken

LabVIEW ist eine eingetragene Marke von National Instruments.

Windows, C#, VB.NET, VB6 und Visual Studio sind eingetragene Marken der Microsoft Corporation.