

2022

# YMLIB API Referenzen ETH

Version: 1.0



## YMLib API Referenz für ETH Module (Windows)

---

### Inhaltsverzeichnis

1	YMLIB API Referenzen ETH .....	2
2	Klasse YMTCPData .....	2
2.1	Methoden der Klasse YMTCPData.....	2
2.2	Eigenschaften der Klasse YMTCPData.....	3
2.3	Event Handler der Klasse YMTCPData .....	3
3	Schaltzustand an das Modul senden .....	4

## 1 YMLIB API Referenzen ETH

Sie können das Yamutec<sup>®</sup> Ethernet Module im gleichen Netzwerk problemlos ansteuern. Besitzt das Netzwerk einen Zugang zum Internet können Sie die Yamutec<sup>®</sup> Ethernet Module mit der Yamutec I/O Remote App ganz leicht von jedem Standort aus fernwarten und -steuern und dessen Zustände in Echtzeit einsehen.

Die YMLib-API erlaubt Ihnen die individuelle Anpassung der Yamutec<sup>®</sup> Ethernet Module per VB6, VB.NET, C# oder LabView an Ihren Anwendungsfall.

Anhand der gelieferten Programmierbeispiele, können Sie die gesamte Programmierung entnehmen.

## 2 Klasse YMTCPData

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

### 2.1 Methoden der Klasse YMTCPData

`YMCom.Connect(ip, port)`

Baut eine Verbindung zum Modul auf.

`YMCom.Disconnect()`

Trennt die Verbindung zum Modul.

`YMCom.CALL_IP_SETTINGS()`

Ruft Informationen vom Modul ab.

`YMCom.CALL_ABOUT_BOX()`

Ruft den YMLib.dll Version auf.

`YMCom.CALL_SCHEDULE_SETTINGS()`

Ruft die Einstellungen für das Programmablauf (Optional)

`YMCom.CALL_USER_SETTINGS()`

Ruft die Einstellungen für den User-Login

`YMCom.TIMEOUT_INTERVAL_AND_CHANNEL_XX()`

Schaltet die ausgewählten Ausgänge aus/ein, wenn sich das Gerät im Timeout-Status befindet.

`YMcom.TIMEOUT_DISABLE()`

Deaktiviert den Timeout-Status.

`YMcom.HW_RESET()`

Setzt das Modul zurück.

`YMcom.SEND_OUTPUT_STATUS_X()`

Sendet ein Schaltzustand an das Modul

## 2.2 Eigenschaften der Klasse YMTCPData

YCom.Disconnect = Boolean

Gibt zurück, wenn die Verbindung getrennt wird.

YCom.IsOpen = Boolean

Gibt zurück, ob Port geöffnet ist oder nicht.

YCom.YM\_Con\_DeviceName = String

Gibt den Modulnamen zurück.

YCom.YM\_TimeOutTriggered = Byte

Gibt zurück, wenn sich das Gerät derzeit im Timeout-Status befindet.

YCom.YM\_OUTPUT1, YCom.YM\_OUTPUT2, YCom.YM\_OUTPUT3, YCom.YM\_OUTPUT4

YCom.YM\_OUTPUT5, YCom.YM\_OUTPUT6, YCom.YM\_OUTPUT7, YCom.YM\_OUTPUT8 = Integer

Gibt die Zustände der Ausgänge zurück.

YCom.YM\_Parse1, YCom.YM\_Parse2, YCom.YM\_Parse3, YCom.YM\_Parse4,

YCom.YM\_Parse5, YCom.YM\_Parse6, YCom.YM\_Parse7, YCom.YM\_Parse8 = String

Gibt die Zustände der Ausgänge zurück.

YCom.YM\_ProgrammSequenceAktive = Byte

Gibt zurück, wenn sich das Modul im Programm-Sequenz befindet.

YCom.YM\_SecIndex = Integer

Gibt den Timeout Intervall zurück

YCom.YM\_TimeOutActive = Byte

Gibt zurück, ob der Timeout aktiviert oder deaktiviert ist.

## 2.3 Event Handler der Klasse YMTCPData

YCom.FirstConnect

Wird nach dem Verbindungsaufbau, von der Klasse YMTCPData getriggert.

YCom.DuringConnected

Wird von der Klasse getriggert, solange die Verbindung steht.

### 3 Schaltzustand an das Modul senden

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

#### Relais Module

ETH Modul mit 4 Ausgänge

```
YCom.SEND_OUTPUT_STATUS_4(SendeByte1)
```

ETH Modul mit 8 Ausgänge

```
YCom.SEND_OUTPUT_STATUS_8(SendeByte1)
```

ETH Modul mit 16 Ausgänge

```
YCom.SEND_OUTPUT_STATUS_16(SendeByte1, SendByte2)
```

ETH Modul mit 32 Ausgänge

```
YCom.SEND_OUTPUT_STATUS_32(SendeByte1, SendByte2, SendByte3, SendByte4)
```

ETH Modul mit 64 Ausgänge

```
YCom.SEND_STATUS_64(SendeByte1, SendByte2, SendByte3, SendByte4  
SendeByte5, SendByte6, SendByte7, SendByte8)
```

#### Hybrid Module

ETH Modul mit 8 Ausgänge und 8 Eingänge

```
YCom.SEND_HYBRID_STATUS_8(SendeByte1)
```

ETH Modul mit 16 Ausgänge und 16 Eingänge

```
YCom.SEND_HYBRID_STATUS_16(SendeByte1, SendByte2)
```

ETH Modul mit 32 Ausgänge und 32 Eingänge

```
YCom.SEND_HYBRID_STATUS_32(SendeByte1, SendByte2, SendByte3, SendByte4)
```

#### Parameter:

SendeByte1, SendByte2, SendByte3, SendByte4  
SendeByte5, SendByte6, SendByte7, SendByte8

Die Ausgangszustände des Moduls werden ständig gelesen und in der Variable **Dim** `SendeByte1` **As Byte** festgehalten. Zum Schalten der Ausgänge muss daher nur geschrieben werden was ein- bzw. ausgeschaltet werden soll.

z.B. Variablenwert `SendeByte1=77` bedeutet, dass die Ausgänge 1,3,4, und die 7 eingeschaltet sind. Möchte man jetzt nur den Ausgang 3 ausschalten, so ist aus dem Variablenwert 77 die 4 zu subtrahieren und der Wert 73 wird an das Modul gesendet.