

# TCPCI120

---

## Hardware

Installationshandbuch

Ausgabe 1.00 vom 09.11.2007

**MSC Tuttlingen GmbH**

Rudolf-Diesel-Straße 17  
D-78532 Tuttlingen

Tel. +49 7461 925 200

Fax +49 7461 925 268

E-Mail [vertrieb@msc-tuttlingen.de](mailto:vertrieb@msc-tuttlingen.de)

[www.msc-tuttlingen.de](http://www.msc-tuttlingen.de)





## Inhaltsverzeichnis

<b>1</b>	<b>Technische Spezifikationen TCPCI120 .....</b>	<b>3</b>
1.1	Werkzeugüberwachungssystem: Übersicht.....	3
1.2	Konzept des Werkzeugüberwachungssystems .....	3
<b>2</b>	<b>Installieren der PCI-Karte .....</b>	<b>4</b>
<b>3</b>	<b>TTBUS-Vernetzung .....</b>	<b>8</b>
3.1	TTBus-Verkabelung .....	8
<b>4</b>	<b>Profibus-Vernetzung.....</b>	<b>9</b>
4.1	Profibus: Erwägungen zur Taktzeit .....	9
4.2	Sendetelegramm (Profibus-Master an TCPCI 120).....	9
4.3	Eingangstelegramm (Profibus-Master an TCPCI 120) .....	10
4.4	Die Profibus-GSD-Datei TPCI0A0B.GSD .....	11

## Abbildungsverzeichnis

Abb. 2-1: Software .....	4
Abb. 2-2: Speicherort des Treibers.....	5
Abb. 2-3: Hardware Installation .....	5
Abb. 2-4: Treiber .....	6
Abb. 2-5: Device Manager .....	6
Abb. 2-6: Treiberinformationen.....	7
Abb. 3-1: TTBus-Verkabelung .....	8

TECHNA-CHECK<sup>®</sup> ist eine eingetragene Marke von Techna-Tool Inc.,  
Hartland Wisconsin USA.

**TCPCI120**  
**Hardware-Installationshandbuch**  
**Artikel-Nr. 68 36 259**

© Copyright MSC Tuttlingen GmbH, 78532 Tuttlingen, 2007  
Technische Änderungen vorbehalten.

# 1 Technische Spezifikationen TCPCI120

## Technische Spezifikationen

PCI	PCI rev. 2.2 - 32bit 33/66 MHz
Betriebssystem	Windows XP
Minimalsystem	1Ghz Pentium 51 2MB RAM
Profibus-Dp	DPV1 Interface (9pin Dsub)
Proprietärer Bus	TTBUS (Phoenix-Stecker)

### 1.1 Werkzeugüberwachungssystem: Übersicht

Die Hauptkomponente des PCI-Systems ist eine PCI-Karte. Diese kann in eine Windows-basierte CNC-Steuerung mit freien PCI-Steckplätzen eingebaut werden. Die Karte kann wahlweise in einen eigenständigen Rechner oder in einen in den Automaten integrierten Industriecomputer eingebaut werden.

Die TCPCI120-Karte verbindet das Werkzeugüberwachungssystem mit der NC-Steuerung der Maschine und mit verschiedenen Sensoren zur Werkzeugüberwachung. Die TCPCI 120 regelt demzufolge nur den Datenaustausch. Die tatsächliche Werkzeugüberwachung wird von der Computeranwendung TTMON ausgeführt. TTMON implementiert ein Werkzeugüberwachungssystem mit maximal 20 Kanälen (Spindel), das bis zu 128 Schneidprozesse pro Kanal überwachen kann.

### 1.2 Konzept des Werkzeugüberwachungssystems

Das Werkzeugüberwachungssystem stellt eine einheitliche Schnittstelle für verschiedene Sensoren bereit. Bei dem Überwachungsvorgang wird entweder die Leistung oder die Vibration gemessen. Die NC-Steuerung stellt dem Überwachungssystem Steuerbefehle ("Start", "Learn" und "Reset"), Schnittnummern und unter Umständen auch Messwerte zur Verfügung. Das Überwachungssystem gibt bei Werkzeugproblemen Warnmeldungen aus.

Die Betriebsdetails der Werkzeugsüberwachungsfunktionen werden bei der NC-Steuerung nicht angezeigt.

Die Arbeitsweise des PC-Werkzeugüberwachungsprogramms TTMON wird in einem anderen Benutzerhandbuch beschrieben.

## 2 Installieren der PCI-Karte

Die TCPCI120-Karte ist Plug-&-Play-kompatibel und wird nach Installation und Neustart von Windows XP erkannt.

Die TCPCI120 verwendet einen Windows WDM-Treiber. Der Treiber wird von der PLX Corporation (Entwickler des Chips der PCI-Schnittstelle) bereitgestellt. Die von PLX bereitgestellte Datei PciSdk.inf stellt Windows die zur Installation des Treibers nötigen Informationen zur Verfügung.

Nach Rechnerstart und Erkennen der Karte sollte Folgendes zu sehen sein:

1. Windows erkennt die neue Hardware und öffnet das Dialogfenster "Neue Hardware gefunden". Bestätigen Sie das Dialogfenster.
2. Anschließend wird der Assistent "Neue Hardware hinzufügen" angezeigt. Der Assistent sucht nach einem geeigneten Treiber.
3. Die Karte wird als PCI-Brücke oder Anderes PCI-Brückengerät erkannt. Klicken Sie auf Weiter.



Abb. 2-1: Software

4. Wählen Sie Software von einer Liste oder bestimmten Quellen installieren (für fortgeschrittene Benutzer) aus. Klicken Sie auf Weiter.

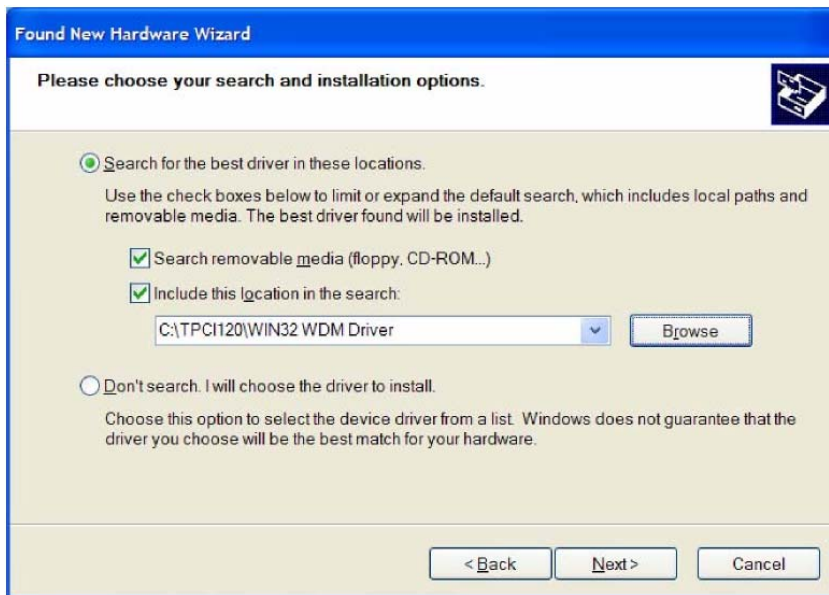


Abb. 2-2: Speicherort des Treibers

5. Navigieren Sie zum Speicherort des Treibers. Klicken Sie auf Weiter.

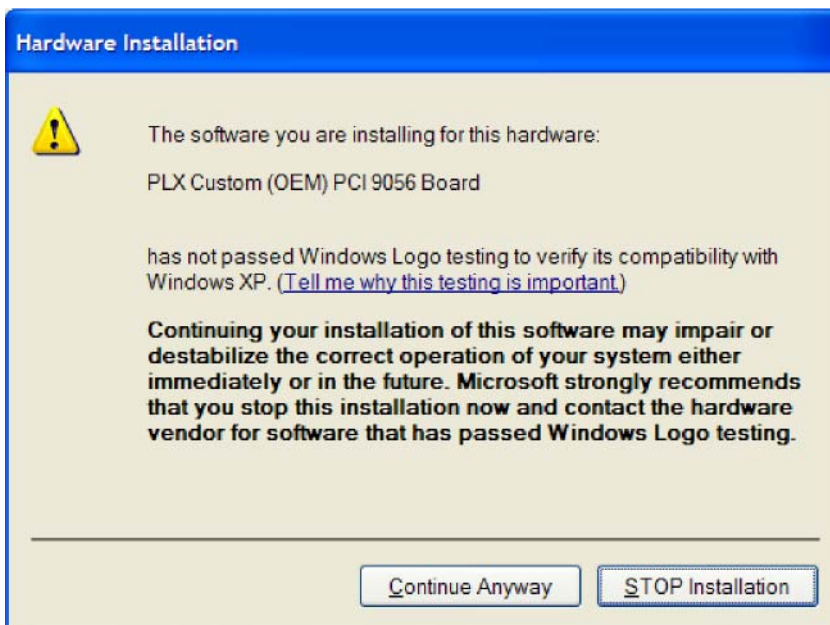


Abb. 2-3: Hardware Installation

6. Der Treiber wird Microsoft zur so genannten Treibersignierung übermittelt. Wählen Sie Dennoch fortfahren aus.



Abb. 2-4: Treiber

7. Der Treiber wird installiert.

Der Treiber befindet sich nun im Gerätemanager unter Andere Geräte - PLC Custom (OEM) PCI 9056 Board.

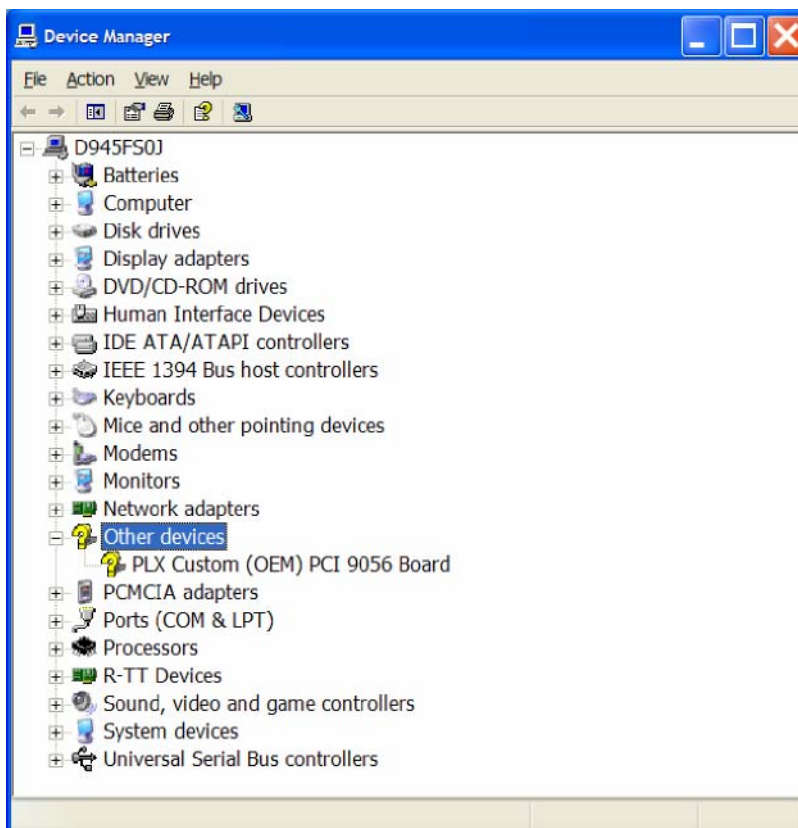


Abb. 2-5: Device Manager

## Treiberinformationen:

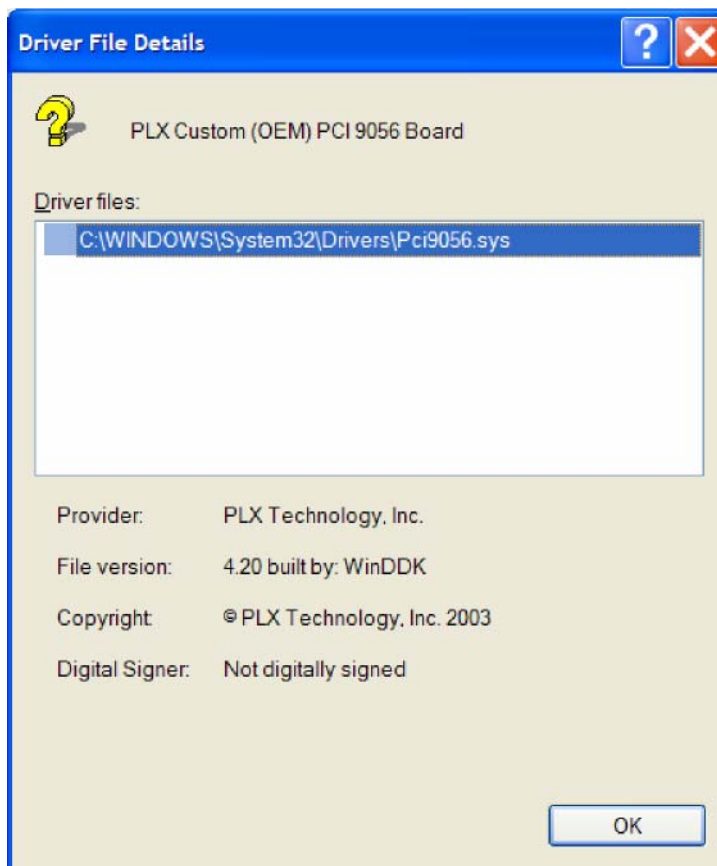


Abb. 2-6: Treiberinformationen

### 3 TTBUS-Vernetzung

Der proprietäre Kommunikationsbus TTBUS ist darauf ausgelegt, mehrere Messwertnehmer und/oder digitale Ein-/Ausgabegeräte mit dem Werkzeugüberwachungssystem zu verbinden. TTBUS basiert auf der standardmäßigen RS485-Hardware.

Bisher wurden drei TTBUS-Geräte entwickelt.

PWM350T	3-Phasen-Leistungsaufnehmer
VM100T	Vibrationssensoren-Schnittstelle
I/O 100T	Digitales Ein-/Ausgabegerät

Möglicherweise werden in Zukunft weitere Sensorentypen hinzugefügt.

Eine detaillierte Beschreibung der TTBUS-Geräte findet sich im Anhang dieses Handbuchs. Jede TTBUS-Einheit erhält innerhalb des Netzwerkes eine eindeutige Adressierung. Die Adresse wird über zwei BCD-Schalter an der Vorderseite der Geräte eingestellt. Das Überwachungssystem erkennt die TTBUS-Wandler automatisch. Über ein TTMON-Menü werden die Geräte den einzelnen Kanälen zugewiesen.

#### 3.1 TTBUS-Verkabelung

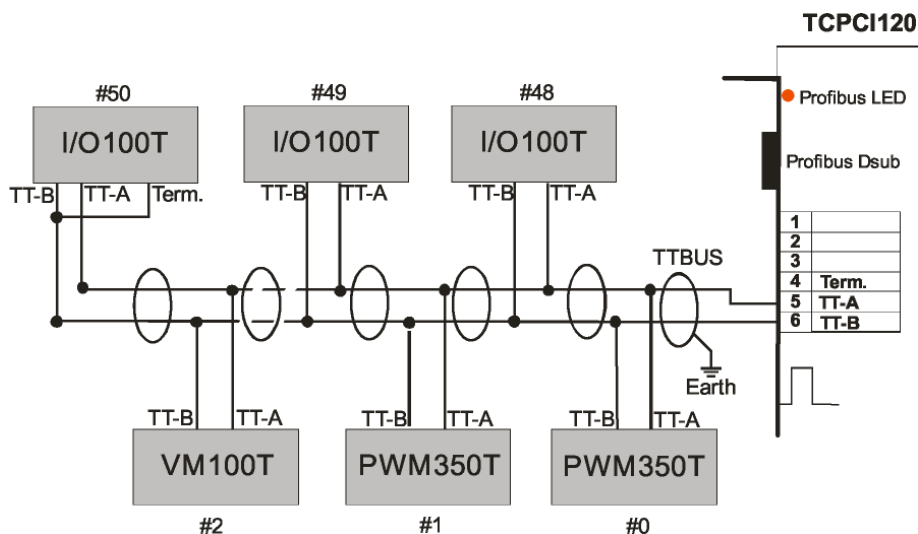


Abb. 3-1: TTBUS-Verkabelung

#### Wichtig:

Benutzen Sie im TTBUS-Netzwerk bitte nur hochwertige, niederohmige, verdrehte und entstörte Kabel, die an einem oder beiden Enden geerdet sind.

Das letzte Gerät des TTBUS-Netzwerkes muss terminieren. Alle Geräte können durch den Einsatz eines externen Kabels terminieren. Kabelstümpfe sollten so kurz wie möglich gehalten werden.

## 4 Profibus-Vernetzung

### 4.1 Profibus: Erwägungen zur Taktzeit

Die Taktung des Profibus sollte auf maximal 10-15 ms (Millisekunden) eingestellt sein. Das entspricht einer Übermittlung von 67-100 Profibus-Telegrammen pro Sekunde. Dazu ist gewöhnlich eine Profibus-Übertragungsgeschwindigkeit (Baud-Rate) von 1 MB oder mehr nötig. Wird diese Bedingung nicht erfüllt, kann die Synchronisierung unter den Durchläufen leiden. Auch die Messgenauigkeit kann leiden (falls Messwerte über das Profibus-Netzwerk übertragen werden).

### 4.2 Sendetelegramm (Profibus-Master an TCPCI 120)

Das Sendetelegramm hat immer eine Länge von 80 Byte – 4 Byte pro Kanal. Es werden immer Daten für 20 Kanäle übermittelt, unabhängig der tatsächlich genutzten Kanäle. Die für nicht verwendete Kanäle übermittelten Daten haben den Wert 0. Sendetelegramme übermitteln dem TCPCI120-Werkzeugkontrollsystem Kontrollbefehle, Schnittnummern und eventuelle Messdaten.

Telegrammformat:

```
BitFlags#1, Measurement#1, CutNumber#1 ,BitFlags#2, Measurement#2,
CutNumber#2, ..... BitFlags#20, Measurement#20, CutNumber#20
```

Telegramm Daten - Byte nummerierung:

Byte-Nr.

0	BitFlags#1	Kanal-Nr. 1 –	8 Bit
1, 2	Measurement#1	Kanal-Nr. 1 –	16 Bit
3	CutNumber#1	Kanal-Nr. 1 –	8 Bit
4	BitFlags#2	Kanal-Nr. 2 –	8 Bit
5, 6	Measurement#2	Kanal-Nr. 2 –	16 Bit
7	CutNumber#2	Kanal-Nr. 2 –	8 Bit
.....			
76	BitFlags#20	Nr. 20 –	8 Bit
77, 78	Measurement#2 0	Nr. 20 –	16 Bit
79	CutNumber#2 0	Nr. 20 –	8 Bit

BitFlags# - b7b6b5bb3b2b1b0

```
#define PROFIBUS_MODE_MASK
```

```
(PROFIBUS_MODE1 | PROFIBUS_MODE2 | PROFIBUS_MODE3) // b6b5b4
```

```
// PROFIBUS - bit_signals -
```

```
Extern erzeugte Signale - Eingänge
```

```
#define START_SIGNAL_ACTIVATED 0x0 1 // b0 = Start Signal
#define LEARN_SIGNAL_ACTIVATED 0x02 // b1 = Learn Signal
#define RESET_ALARM_SIGNAL_ACTIVATED 0x0 4 // b2 = Reset Signal
#define PROFIBUS_SPARE 0x0 8 // b3 = not used
#define PROFIBUS_MODE1 0x1 0 // b4 = Profibus mode
#define PROFIBUS_MODE2 0x2 0 // b3 = Profibus mode
#define PROFIBUS_MODE3 0x4 0 // b2 = Profibus mode
#define PROFIBUS_UNIT_PRESENT 0x8 0 // b1 = Channel
```

Measurement# - 2 Byte MSB, LSB

16 bit measurement value 0 - 1000 decimal = 0.0 - 100.0%

CutNumber# - 1 Byte

### 4.3 Eingangstelegramm (Profibus-Master an TCPCI 120)

Das Eingangstelegramm (an den Profibus-Master) ist immer 20 Byte lang – 1 Byte pro Kanal. Kanal-Nr. 1 liegt auf dem ersten, Kanal-Nr. 20 auf dem letzten Byte. Es werden Warnmeldungen und andere Statusinformationen an den Master (die NC-Steuerung) übermittelt.

Telegrammformat:

**InputFlags#1, InputsFlags#2 ..... InputFlags#20**

**InputFlags -**

b7b6b5bb3b2b1b0

```
#define ACTIVE_READY      0x01 // b0    = Po Measured or Touched
                          (BK MICRO)
#define SPARE1            0x02 // b1    = bit not used
#define TOUCHED           0x04 // b2    = Touched
#define IDLE_ALARM        0x08 // b3    = IDLE_ALARM
#define BLUNTCOUNT_ALARM  0x10 -{}-// b4
#define MISSING_ALARM     0x20 // b5    = MISSING_ALARM
#define BLUNT_ALARM       0x40 // b6    = BLUNT_ALARM
#define BREAK_ALARM       0x80 // b7    = BREAK_ALARM
```

#### **ACTIVE\_READY**

Das Bit wird bei der Aktivierung der Werkzeugüberwachung gesetzt.

Beispiel 1: Das Startsignal wurde aktiviert und die Leerlaufleistung berechnet.

Beispiel 2: Das Vorhandensein des Zielwerkzeugs wurde geprüft – "BK Mikro" wird aufgerufen.

In manchen Fällen kann Durchlaufzeit gespart werden, indem das Werkzeug erst aufgerufen wird, wenn dieses Bit aktiv ist. Eine alternative (traditionelle) Vorgehensweise ist es, eine vorgegebene Verzögerung festzulegen. Diese muss lang genug sein, um die Leerlaufleistung zu berechnen.

#### **SPARE1**

Wird nicht verwendet.

#### **TOUCHED**

Wird mit der Funktion "Touch-Limit" verwendet und gesetzt, sobald das Werkzeug das Arbeitsteil berührt – das Signal erreicht den Grenzwert.

#### **IDLE\_ALARM**

Zeigt das Vorhandensein von IDLE\_ALARM.

#### **BLUNTCOUNT\_ALARM**

Zeigt das Vorhandensein von BLUNTCOUNT\_ALARM.

#### **MISSING\_ALARM**

Zeigt das Vorhandensein von MISSING\_ALARM.

#### **BLUNT\_ALARM**

Zeigt das Vorhandensein von BLUNT\_ALARM.

#### **BREAK\_ALARM**

Zeigt das Vorhandensein von BREAK\_ALARM.

#### 4.4 Die Profibus-GSD-Datei TPCI0A0B.GSD

```

; =====
; Techna Tool Inc.
;
; File : TPCI0A0B.GSD
; Revision : 1.0
; Last Modification : 05/09/2005
; =====
;
;#Profibus_DP
; General device information
GSD_Revision          = 1
Vendor_Name           = "Techna Tool Inc."
Model_Name            = "TPCI120"
Revision              = "V1.0"
Ident_Number          = 0x0A0B
Protocol_Ident        = 0          ; 0 = PROFIBUS-DP only
Station_Type          = 0          ; 0 = DP-Slave
FMS_supp              = 0          ; FMS is not supported
Hardware_Release      = "A1"
Software_Release      = "V1.0"

; Supported baudrates
9.6_supp              = 1
19.2_supp             = 1
45.45_supp           = 1
93.75_supp           = 1
187.5_supp           = 1
500_supp              = 1
1.5M_supp            = 1
3M_supp              = 1
6M_supp              = 1
12M_supp             = 1

; MaxTsdr default values for supported baudrates
MaxTsdr_9.6          = 60
MaxTsdr_19.2         = 60
MaxTsdr_45.45        = 60
MaxTsdr_93.75        = 60
MaxTsdr_187.5        = 60
MaxTsdr_500          = 100
MaxTsdr_1.5M         = 150
MaxTsdr_3M           = 250
MaxTsdr_6M           = 450
MaxTsdr_12M          = 800

; General supported features
Redundancy            = 0          ; Redundancy not supported
Repeater_Ctrl_Sig     = 2          ; RTS Signal with TTL level
24V Pins              = 0          ;
Implementation_Type   = "ASIC_solution, VPC3+"

; DP Slave related information
Freeze_Mode_supp      = 0          ; Freeze-Mode not supported
Sync_Mode_supp        = 0          ; Sync.-Mode not supported
Auto_Baud_supp        = 1          ; Automatic baud control supported
Max_Diag_Data_Len     = 6
Set_Slave_Add_supp    = 0          ; Set Slave address not supported
User_Prm_Data_Len     = 05         ;
Min_Slave_Intervall   = 5          ; 500us
Slave_Family          = 1@TT@TPCI

; Modules information
Modular_Station       = 0
Max_Module            = 1
Max_Input_Len         = 20
Max_Output_Len        = 80
Max_Data_Len          = 100

Module = "80 Byte out/ 20 Byte In" 0x2f,0x2f,0x2f,0x2f,0x2f,0x1f,0x13

EndModule

```

