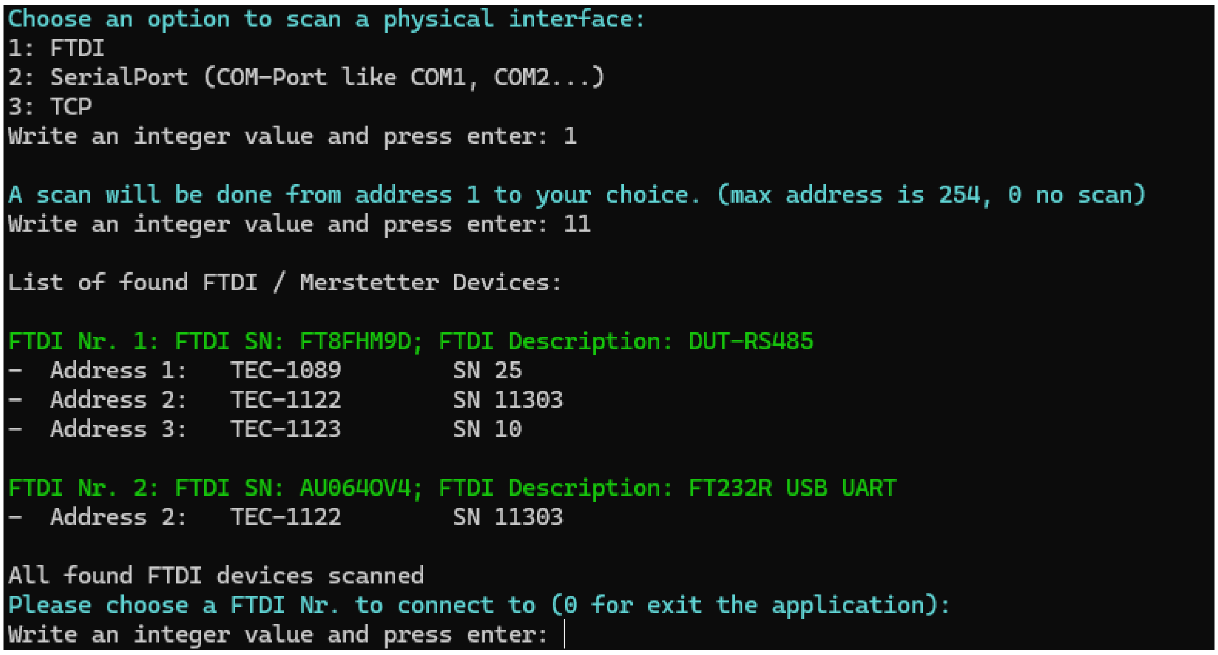
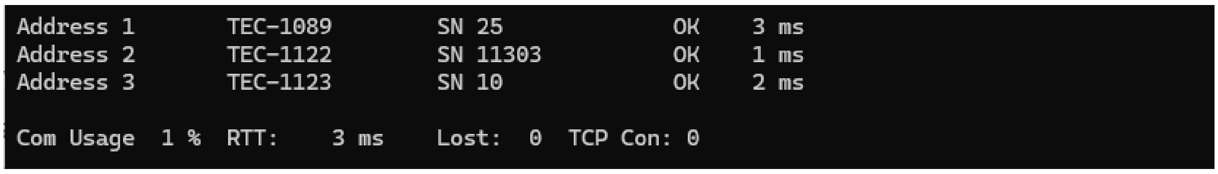
# MeCom Device Server



```
Choose an option to scan a physical interface:
1: FTDI
2: SerialPort (COM-Port like COM1, COM2...)
3: TCP
Write an integer value and press enter: 1

A scan will be done from address 1 to your choice. (max address is 254, 0 no scan)
Write an integer value and press enter: 11

List of found FTDI / Merstetter Devices:

FTDI Nr. 1: FTDI SN: FT8FHM9D; FTDI Description: DUT-RS485
-  Address 1:   TEC-1089       SN 25
-  Address 2:   TEC-1122       SN 11303
-  Address 3:   TEC-1123       SN 10

FTDI Nr. 2: FTDI SN: AU064OV4; FTDI Description: FT232R USB UART
-  Address 2:   TEC-1122       SN 11303

All found FTDI devices scanned
Please choose a FTDI Nr. to connect to (0 for exit the application):
Write an integer value and press enter: 
```



```
Address 1      TEC-1089      SN 25          OK    3 ms
Address 2      TEC-1122      SN 11303       OK    1 ms
Address 3      TEC-1123      SN 10          OK    2 ms

Com Usage  1 %  RTT:    3 ms    Lost:  0  TCP Con: 0
```

A .NET application to serve multiple master (Client) connections to one or several devices on a BUS, a USB connection or Serial Port.

Running on Windows and Linux.

Please contact our support team, if you have any specific requirements for this application.

meerstetter engineering

Member of Berndorf Group

Developed, assembled, and tested in Switzerland

# Index

**Meerstetter Engineering GmbH**
Schulhausgasse 12
CH-3113 Rubigen
Switzerland

Phone: +41 31 529 21 00
Email: contact@meerstetter.ch

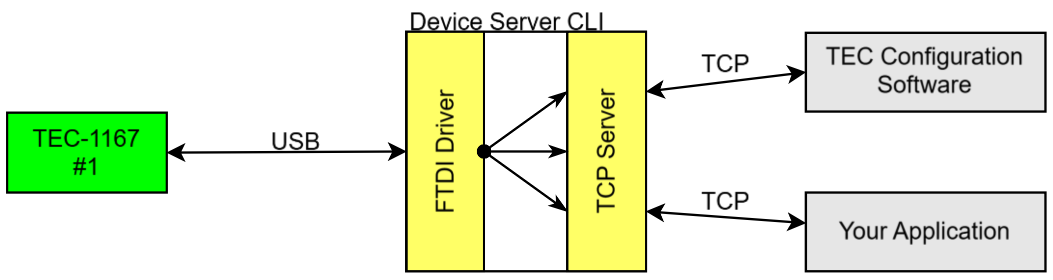
Document 5325A

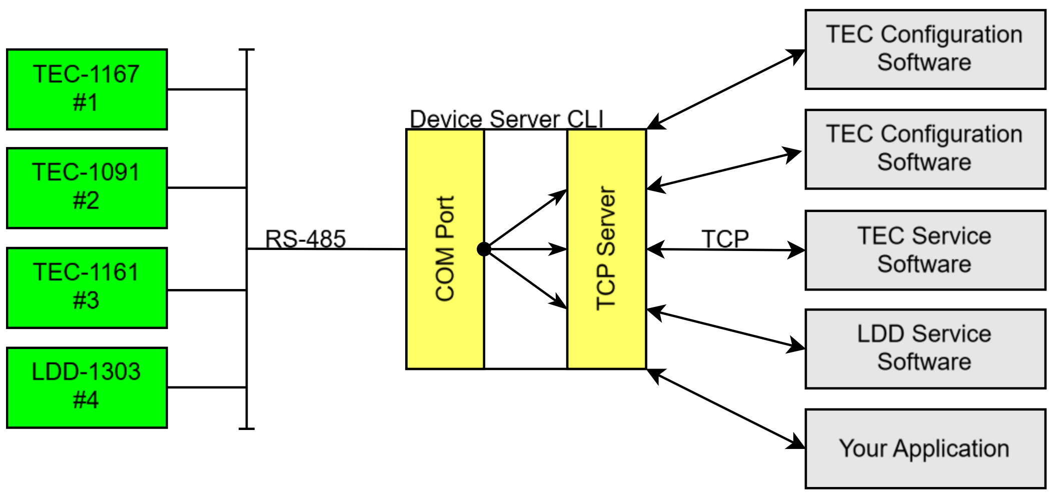
Release date: 9 May 2025

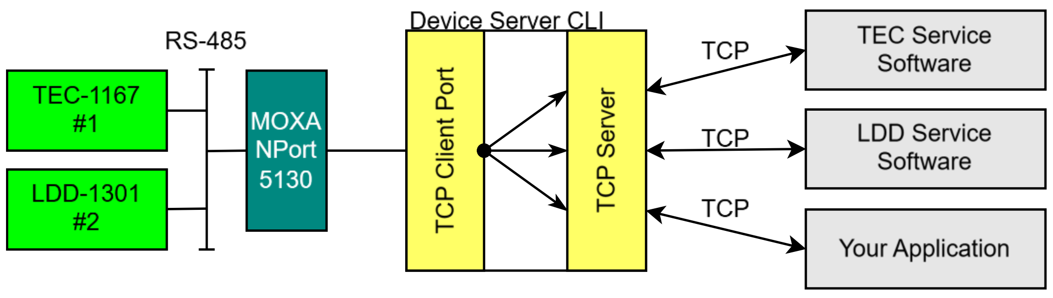
# 1 Use case

The “MeCom Device Server” is a .NET application, which works as flexible interface between Meerstetter devices and various master applications (Service Software, Configuration Software or your own application).

- It handles the traffic on the MeCom packet level.
- Accepts multiple TCP connections from master applications and forwards the requests to the downstream connections.
- Scans RS-485 busses and observes the found devices.
- Can send speed change commands or works on a fixed baud rate.
- Can directly connect to FTDI chips, COM ports or Serial Servers like the MOXA NPort.
- Interactive command line application.
- Rich set of startup arguments for unattended usage.





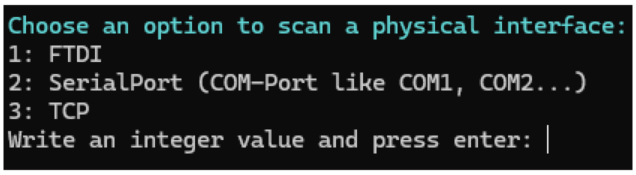


#X → Device Address.

# 2 Interactive CLI usage

If the application is started without any arguments, then the application can be used with simple commands.

First, you can choose the downstream interface, which makes the connection to the Meerstetter device.



```
Choose an option to scan a physical interface:
1: FTDI
2: SerialPort (COM-Port like COM1, COM2...)
3: TCP
Write an integer value and press enter: |
```

Options:

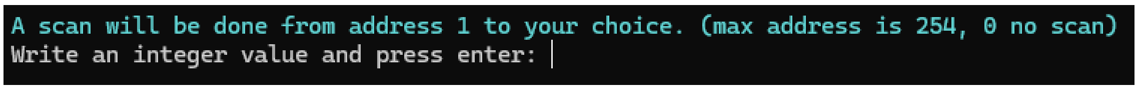
- FTDI: Most Meerstetter devices have a USB port with an FTDI Chip to handle the USB connection. There are also FTDI to RS-485 bus cables available.
- Serial Port: For native or virtual COM ports.
- TCP: Opens a TCP/IP downstream connection. For instance, if you use a MOXA Serial Server.

## 2.1 FTDI selection

This option is not possible with Linux.

If you select FTDI as downstream interface, then it will ask you the address room to scan.

If you have only one device connected, then make sure you type a high enough address, that your device will get discovered.



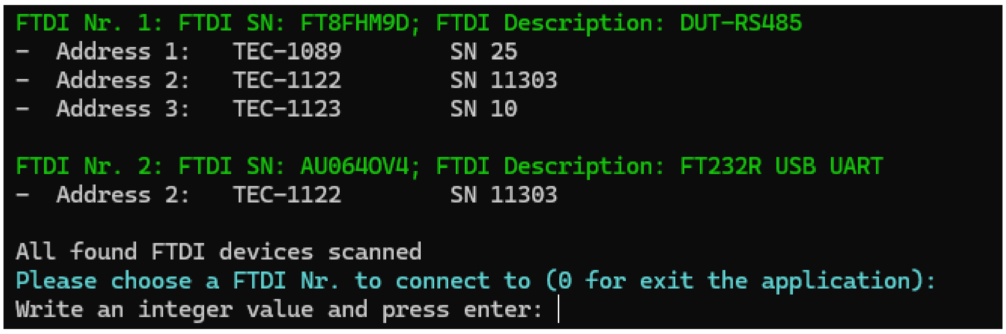
```
A scan will be done from address 1 to your choice. (max address is 254, 0 no scan)
Write an integer value and press enter: |
```

Typed 5.

In the demonstration setup, there are 2 USB connections.

- One FTDI to RS-485 cable, where 3 TEC-Controllers are connected.
  All 3 controllers have different device addresses.
- One TEC-Controller is directly connected to the computer.

The application opens each FTDI Chip and scans each device address of the chosen address room.

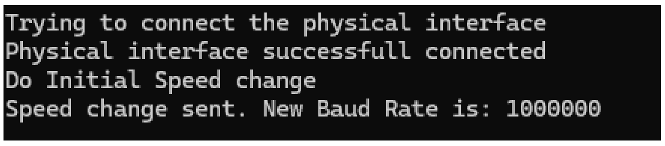


```
FTDI Nr. 1: FTDI SN: FT8FHM9D; FTDI Description: DUT-RS485
-  Address 1:   TEC-1089        SN 25
-  Address 2:   TEC-1122        SN 11303
-  Address 3:   TEC-1123        SN 10

FTDI Nr. 2: FTDI SN: AU064OV4; FTDI Description: FT232R USB UART
-  Address 2:   TEC-1122        SN 11303

All found FTDI devices scanned
Please choose a FTDI Nr. to connect to (0 for exit the application):
Write an integer value and press enter:
```

Typed 1.

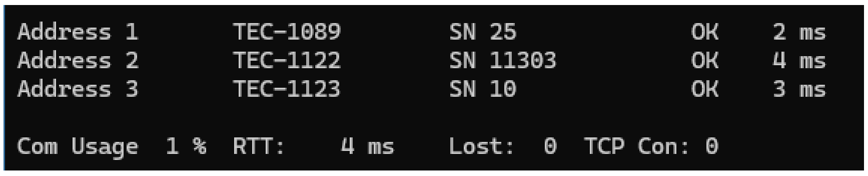
The application opens the interface again and does an initial speed change, because by default the devices do only communicate with a baud rate of 57’600 over the USB interface.

It sends a speed change command to all devices and changes the baud rate to 1’000’000.



```
Trying to connect the physical interface
Physical interface successfull connected
Do Initial Speed change
Speed change sent. New Baud Rate is: 1000000
```

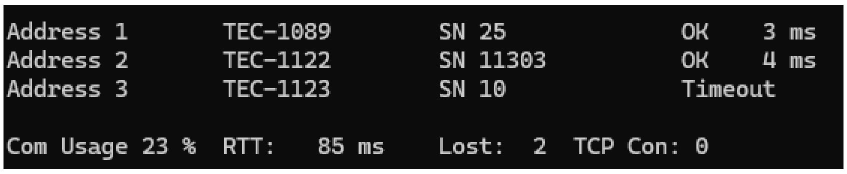
The temporary baud rate will go back to 57’600 if no communication is present. Therefore the application asks each device every 1s if it is still available.



```
Address 1       TEC-1089        SN 25               OK     2 ms
Address 2       TEC-1122        SN 11303            OK     4 ms
Address 3       TEC-1123        SN 10               OK     3 ms

Com Usage   1 %  RTT:     4 ms     Lost:  0  TCP Con: 0
```

If one device does not answer, then it shows timeout for this device.

The application will send another speed change command and tries to re-establish the connection. This does also happen if one device on the bus reboots.



```
Address 1       TEC-1089        SN 25              OK     3 ms
Address 2       TEC-1122        SN 11303           OK     4 ms
Address 3       TEC-1123        SN 10              Timeout

Com Usage 23 %  RTT:    85 ms    Lost:  2  TCP Con: 0
```

At this point the application is ready to accept TCP connections from applications like:

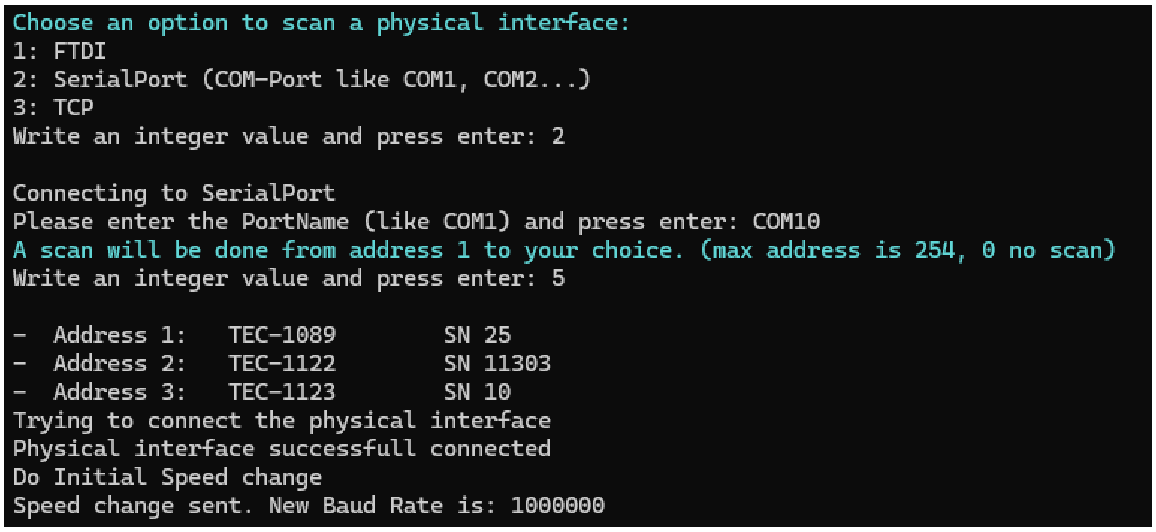
- TEC/LDD Service Software
- TEC/LDD Configuration Software
- Your custom application

By default, it accepts multiple TCP connections on port 50’000.

## 2.2 Serial Port Selection

For the serial port basically the same happens.

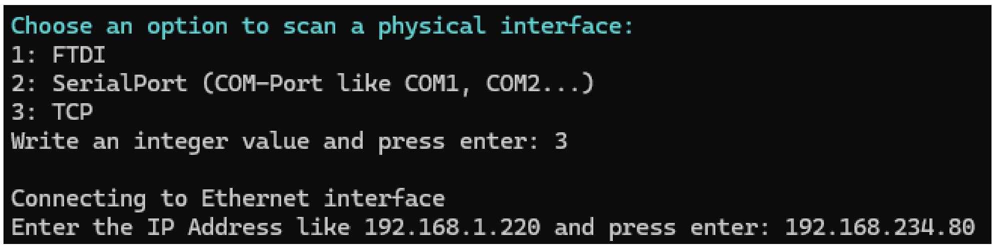
The main difference is that it does only open one COM Port. The one which is specified.



```
Choose an option to scan a physical interface:
1: FTDI
2: SerialPort (COM-Port like COM1, COM2...)
3: TCP
Write an integer value and press enter: 2

Connecting to SerialPort
Please enter the PortName (like COM1) and press enter: COM10
A scan will be done from address 1 to your choice. (max address is 254, 0 no scan)
Write an integer value and press enter: 5

-  Address 1:   TEC-1089        SN 25
-  Address 2:   TEC-1122        SN 11303
-  Address 3:   TEC-1123        SN 10
Trying to connect the physical interface
Physical interface successfull connected
Do Initial Speed change
Speed change sent. New Baud Rate is: 1000000
```

## 2.3 TCP Selection

Same behaviour as with the COM Port.

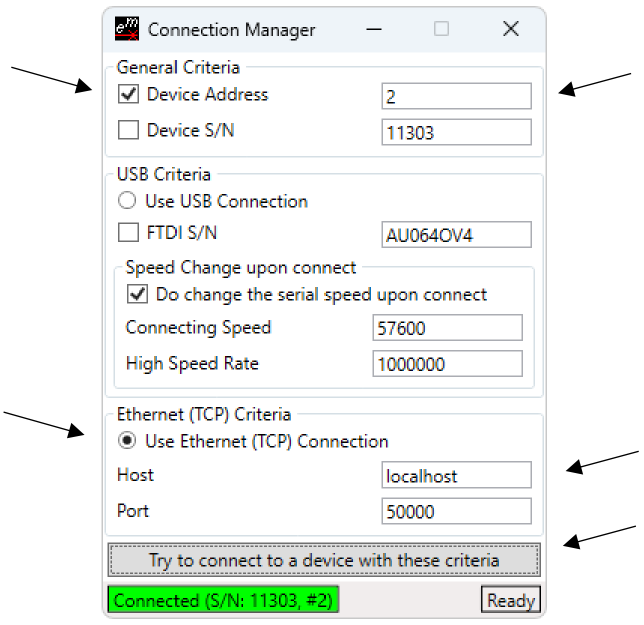


```
Choose an option to scan a physical interface:
1: FTDI
2: SerialPort (COM-Port like COM1, COM2...)
3: TCP
Write an integer value and press enter: 3

Connecting to Ethernet interface
Enter the IP Address like 192.168.1.220 and press enter: 192.168.234.80
```

## 2.4 Connections to the MeCom Device Server

### 2.4.1 Connecting a Configuration Software to the Device Server

- Open the Connection Manager
- Select **Device Address** and type the device address of the device you want to connect to. The MeCom Device Server does not accept any commands to address 0.
- Select Use Ethernet (TCP) Connection and type the hostname, where the MeCom Device Server runs.
  - If the Device Server runs on the same computer, then you can type “localhost” or “127.0.0.1”.
  - If the Device Server runs on a different machine, then you can type its IP-Address or hostname.
- Click Try to connect to a device with these criteria.



Connection Manager

General Criteria
- [✓] Device Address: 2
- [ ] Device S/N: 11303

USB Criteria
- ○ Use USB Connection
- [ ] FTDI S/N: AU064OV4

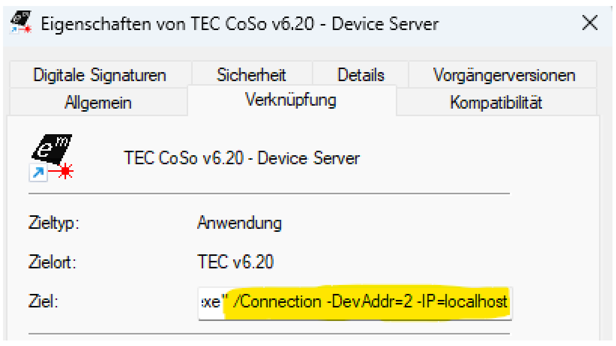
Speed Change upon connect
- [✓] Do change the serial speed upon connect
- Connecting Speed: 57600
- High Speed Rate: 1000000

Ethernet (TCP) Criteria
- ● Use Ethernet (TCP) Connection
- Host: localhost
- Port: 50000

Try to connect to a device with these criteria

Connected (S/N: 11303, #2) Ready

You can create a shortcut to the Configuration Software and pass arguments to automatically connect to a Device Server. Check the corresponding device Communication Manual for details.



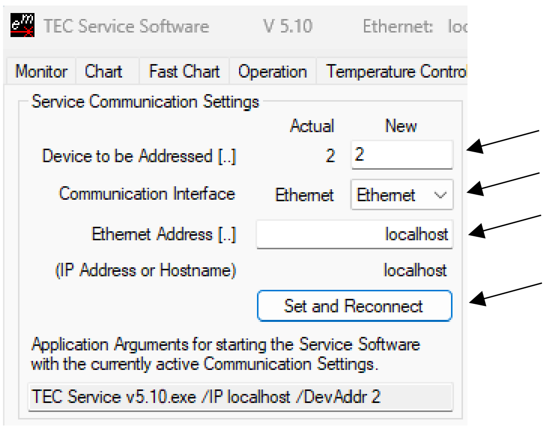
Eigenschaften von TEC CoSo v6.20 - Device Server

TEC CoSo v6.20 - Device Server

Zieltyp: Anwendung
Zielort: TEC v6.20
Ziel: xe" /Connection -DevAddr=2 -IP=localhost

### 2.4.2 Connecting a Service Software to the Device Server

- Go to the Maintenance tab.
- Type the Device to be Addressed (address 0 is not supported)
- Select Ethernet
- Type the host you want to connect to.
  Check the Configuration Software Example above for details.
- Click Set and Reconnect.



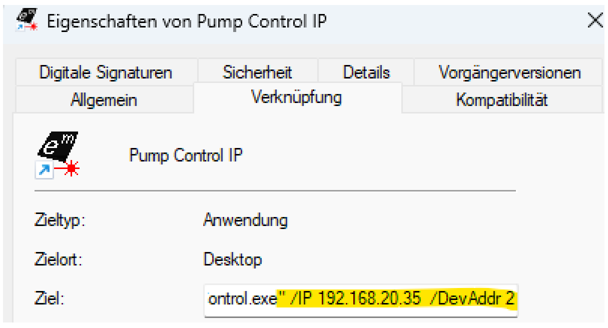
Service Communication Settings

| | Actual | New |
|---|---|---|
| Device to be Addressed [..] | 2 | 2 |
| Communication Interface | Ethernet | Ethernet |
| Ethernet Address [..] | | localhost |
| (IP Address or Hostname) | | localhost |

Application Arguments for starting the Service Software with the currently active Communication Settings.

TEC Service v5.10.exe /IP localhost /DevAddr 2

You can create a shortcut to the Service Software and pass arguments to automatically connect to a Device Server. Check the corresponding device Communication Manual for details.



Pump Control IP

| | |
|---|---|
| Zieltyp: | Anwendung |
| Zielort: | Desktop |
| Ziel: | ontrol.exe" /IP 192.168.20.35 /DevAddr 2 |

### 2.4.3 Connecting your own application to the Device Server

Open a TCP connection to the port 50’000 of the Device Server. Send the same data, as you would send over a serial communication interface.

Keep the connection open and wait for the forwarded answer of the Device Server.

# 3 Arguments

For unattended operation there are a lot of startup arguments, which can be passed to the Device Server. For example, in a shortcut to the Device Server application.

Please read first the capture 2.

Most arguments are optional and the shown value below is the default value.

## 3.1 Connecting to an FTDI Device

FTDI is not supported with Linux.

| **Parameters for `/ClientFTDI`** | **Description** |
|---|---|
| `-OpenByDescription=MeTEC1` | Opens the FTDI device by its description. The description of the FTDI chip can be set using the PT_Prog application from FTDI Chip. |
| `-OpenBySerialNumber=DM01ZCWT` | Opens the FTDI device by its serial number. The serial number is individual for each FTDI chip. The serial number is shown if a FTDI scan is done with the Device Server app without arguments. |
| `-OpenByIndex=0` | Opens the FTDI device by index.<br>If only one device is connected to a host, then this might be a good choice. |
| `-Speed=57600` | Optional base baud rate.<br>If not passed, default is 57600. |
| `-Timeout=400` | Timeout for a request. Default is 400ms |

Only one Open… parameter can be used at the same time.

## 3.2 Connecting to an COM Device

| **Parameters for `/ClientCOM`** | **Description** |
|---|---|
| `-Name=COMxxx` | Required. Name of the COM interface. |
| `-Speed=57600` | Optional base baud rate.<br>If not passed, default is 57600. |
| `-Timeout=400` | Timeout for a request. Default is 400ms |

## 3.3 Connecting to an TCP Server

This can be something like the MOXA NPort 5130.
Check our application note 5218 to configure the NPort.

| **Parameters for `/ClientTCP`** | **Description** |
|---|---|
| `-Host=(IP or Name)` | Required. For Example: 192.168.234.80<br>or device44.local |
| `-Port=50000` | Optional. TCP/IP Port. Default is 50000 |
| `-Timeout=400` | Timeout for a request. Default is 400ms |

## 3.4 Scan Arguments

| **Parameters for** `/Scan` | **Description** |
|---|---|
| `-Start=1` | Start address for scanning. |
| `-End=25` | End address for scanning. |
| `-NrOfTries=1` | Number of tries per address. |
| `-NoScan=0` | If NoScan=1 disables the initial scanning. |

The shown parameters are the default values if nothing is passed.

## 3.5 Speed Change Arguments

| **Parameters for** `/SpeedChange` | **Description** |
|---|---|
| `-HiSpeed=1000000` | Baud rate of the higher speed. |
| `-Initial=1` | An initial broadcast speed change is sent. |
| `-Periodic=0` | A periodic speed change is sent every x millisecond, independent from the device observation. |
| `-PeriodicOldCmd=0` | Same as Periodic, but with the old command used for LDD-112x devices. |
| `-OnObserveTimeout=1` | A speed change is sent if an observed device does not answer anymore. |
| `-Interval=1000` | Observation and speed change interval in milliseconds. |

The shown values are the default values if nothing is passed for ClientFTDI and ClientCOM.

For ClientTCP no speed change is possible, because the speed of the connected serial server cannot be changed. No /SpeedChange arguments are allowed.

If the “Initial” speed change is disabled, then “Periodic”, “PeriodicOldCmd” and “OnObserveTimeout” also needs to be disabled.

## 3.6 Local TCP Server Configuration Arguments

| **Parameters for** `/Server` | **Description** |
|---|---|
| `-Port=50000` | TCP Server Port. Port 50000 is default. |
| `-ReTries=3` | Default number of re tries, in case the timeout elapses. Default is 3. |

The shown value is default if nothing is passed. Can be used if more then one server is needed on one host.

# 4 Installation

## 4.1 Windows

The Device Server is built on top of .NET 8 by Microsoft. Therefore, to be able to run the application it is necessary to first download and install the .NET 8 Desktop Runtime. This runtime can be download from the following page on the official Microsoft website:
Download .NET 8.0 (microsoft.com).

## 4.2 Linux AMD x64

Tested on a x64 Debian 12 computer without graphical interface.

The provided binary is compiled for:

- Target framework: net8.0
- Deployment mode: Framework-dependent
- Target runtime: linux-x64

Therefore, the .NET 8.0 needs to be installed on the target computer.

Debian specific:
https://learn.microsoft.com/en-us/dotnet/core/install/linux-debian?tabs=dotnet8

More General:
https://learn.microsoft.com/en-us/dotnet/core/install/linux

**Effectively used commands to install the .NET environment:**

- `wget https://packages.microsoft.com/config/debian/12/packages-microsoft-prod.deb -O packages-microsoft-prod.deb`
- `sudo dpkg -i packages-microsoft-prod.deb`
- `rm packages-microsoft-prod.deb`
- `sudo apt-get update`
- `sudo apt-get install -y dotnet-runtime-8.0`

**Install the Device Server:**

- Copy the Device Server to the Linux machine.
- Set MeSoft.MeCom.DeviceServer as executable
- Launch it with `./MeSoft.MeCom.DeviceServer`

## 4.3 Linux ARM 64 Raspberry Pi

It is possible to run the device server on a Raspberry Pi. The following steps were tested on a Raspberry Pi 3 Model B running Raspberry Pi OS (64-bit) Release 5.5 November 2024:

Follow the steps in 4.2 Linux with the following changes:

1. The target runtime must be linux-arm64
2. The install via package manager mentioned in the Linux install section is only supported on x86-based systems. Instead, follow the manual/scripted install steps described at https://learn.microsoft.com/en-us/dotnet/core/install/linux-scripted-manual to install the .NET runtime. Raspberry Pi OS is based on Debian, where required, follow the install steps for Debian.
3. Run `sudo raspi-config` and enable the serial interfaces if you intend to use them.
   I. Select “3 Interface Options”
   II. Select “I6 Serial Port”
   III. “Would you like a login shell to be accessible over serial?” → No
   IV. “Would you like the serial port hardware to be enabled?” → Yes
   V. Ok
   VI. Finish
   VII. “Would you like to reboot now?” → Yes
   VIII. You should now see a `/dev/ttyS0` interface which can be used by the device server.

# 5 Various

## 5.1 Error Messages of the Device Server

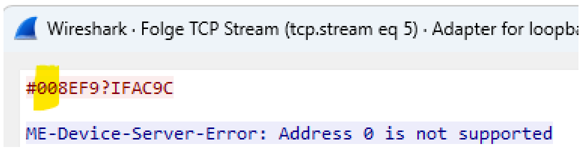
If a serious error occurs, then the Device Server sends a last message to the client and closes the TCP connection. It uses this prefix: "`ME-Device-Server-Error: `".

## 5.2 Device Address 0

By default, the Service Software or Configuration Software applications try to connect to the device address 0.

The Device Server does not support the device address 0 at all.

It closes the TCP connection immediately:

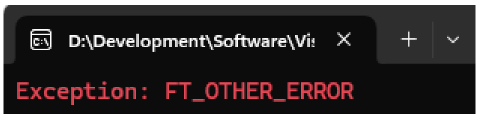

Wireshark · Folge TCP Stream (tcp.stream eq 5) · Adapter for loopb

```
#008EF9?IFAC9C
ME-Device-Server-Error: Address 0 is not supported
```

The correct device address needs to be specified. If you do not know it, use the Service or Configuration Software with a direct USB connection and check the device address.

## 5.3 Downstream disconnect

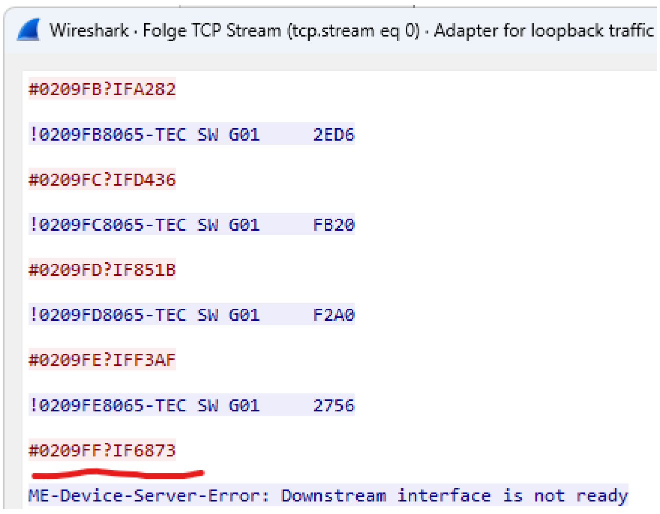
If there is a serious problem with the downstream interface, it shows an error message like this



```
Exception: FT_OTHER_ERROR
```

If then a client application tries to query data, it immediately closes the TCP connection:

The connection was disconnected at the red line.


Wireshark · Folge TCP Stream (tcp.stream eq 0) · Adapter for loopback traffic

```
#0209FB?IFA282
!0209FB8065-TEC SW G01      2ED6
#0209FC?IFD436
!0209FC8065-TEC SW G01      FB20
#0209FD?IF851B
!0209FD8065-TEC SW G01      F2A0
#0209FE?IFF3AF
!0209FE8065-TEC SW G01      2756
#0209FF?IF6873
ME-Device-Server-Error: Downstream interface is not ready
```

## 5.4 Multiple Device Server instances on the same computer

By default, only one Device Server can be started on a computer, because the standard TCP port 50'000 can only be opened once at a time.

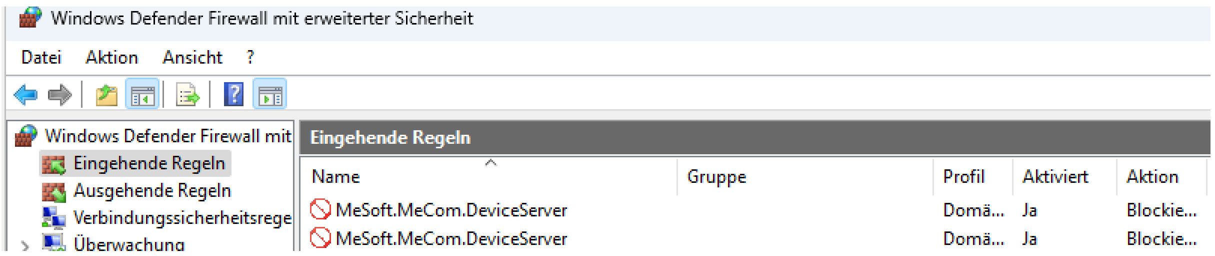
Using the `/Server -Port=50001` argument, it is possible to start several instances of a Device Server. This might be a use case if more than 1 USB connection to the computer is present.

## 5.5 Firewall

The Device Server does accept TCP connections from the whole network, but especially on Windows, the firewall needs to be configured correctly.

By default, Windows asks you to allow connections from outside at the first start.

If you cancelled this request, you must delete the created firewall rules and then start the application again. Windows will ask you again to allow the connections and then create automatically the correct allow rules.


Windows Defender Firewall mit erweiterter Sicherheit
Datei
Aktion
Ansicht
?
Windows Defender Firewall mit
Eingehende Regeln
Ausgehende Regeln
Verbindungssicherheitsrege
Überwachung
Eingehende Regeln

| Name | Gruppe | Profil | Aktiviert | Aktion |
|---|---|---|---|---|
| MeSoft.MeCom.DeviceServer | | Domä... | Ja | Blockie... |
| MeSoft.MeCom.DeviceServer | | Domä... | Ja | Blockie... |

## 5.6 Timeout and ReTries

By default, the timeout is set to 400ms for the downstream interface. This is more than enough time for (nearly) all requests.

ReTries is set to 3 by default. If the Device Server does not receive an answer from the device, it tries up to 3 times to get an answer. If no answer is received, the upstream TCP channel is closed.

If you use a Moxa Serial Device Server with an unstable (ex. WIFI) network, this time might be too short if a lot of TCP/IP packages get lost, because Windows does automatically re-transmit not acknowledged TCP/IP packages again. By default, Windows sends a re-transmit after 300ms, therefore 400ms is a good default value.

Please keep in mind, that lost TCP/IP packages is not the same as when the Meerstetter device does not send an answer.

If TCP/IP packages get lost, the operating system is responsible to re-transmit the packages. If windows does not receive an ACK for a sent package, then the upstream TCP connection is immediately closed after the -Timeout value.

If the device does not answer or the MeCom package gets corrupted (for example on a RS-485 bus), then the Device Server is responsible to re-transmit the package. The Device Server will re-transmit after the -Timeout setting and will do that up to the -ReTries settings.

## 5.7 Firmware upgrade through a Device Server

Some older Meerstetter devices do not answer a few seconds during the clear memory command. Therefore, the Device Server detects this command and sets the timeout temporarily for this command to 10 seconds.

## 5.8 ClientTCP downstream disconnect

If as downstream connection TCP is used, then by default it takes much longer on Linux to detect a failed TCP connection. This is because of the different `tcp_retries2` setting.

If a failed connection is detected, then it tries to reconnect automatically.

- On Windows it takes around 20 seconds to detect a failed connection.
- On Linux it takes around 15 minutes to detect a failed connection.

On Linux this setting can easily be changed using this command:

```
sysctl -w net.ipv4.tcp_retries2=5
```

To set this value permanently, update the `net.ipv4.tcp_retries2` setting in `/etc/sysctl.conf`. To verify after rebooting, run `sysctl net.ipv4.tcp_retries2`.

## 5.9 Client applications connected to the Device Server

Guidelines for your own application, which connects to the Device Server:

- Do not change any TCP/IP timeout settings of your TCP-Client. Let your operating system decide about the correct timeout.
- Do not re-transmit any meCom packages if you do not get answer in a certain time. The Device Server will automatically re-transmit the packages if they get lost on the downstream interface. If the IP package to the Device Server gets lost, the TCP/IP stack of your operating system will automatically re-transmit the package.
- Be prepared, that the Device Server closes the TCP connection, if something fails on the downstream interface. For example, if the device did not answer after some timeout and/or number of re-tries. You can then re-establish a connection after a few 100ms and try again.

## 5.10 MOXA NPort Configuration

Meerstetter provides setup examples for the MOXA NPort 5130 in the document 5218.

Especially important is the detection of the end of a MeCom package, because not every single byte is transferred in a separate IP package.

Most Serial Servers provide an option to set a “Delimiter” to detect the end of a serial message. For the Meerstetter Communication Protocol, 0x0D (Carriage Return) should be used as detection for the end.

## 5.11 Virtual COM Port delay Optimization

If some bytes arrive at the COM port, then the operating system does not immediately forward these bytes to the applications. This increases the round-trip-time of a request significantly.

Example: TEC-1091 connected to one TEC Configuration Software over the Device Server.

- Without Optimization: The Com Usage is around 65%.
- With Optimization: The Com Usage is around 25%.

There are basically two options to reduce the round-trip-time, but not all systems provide both options:

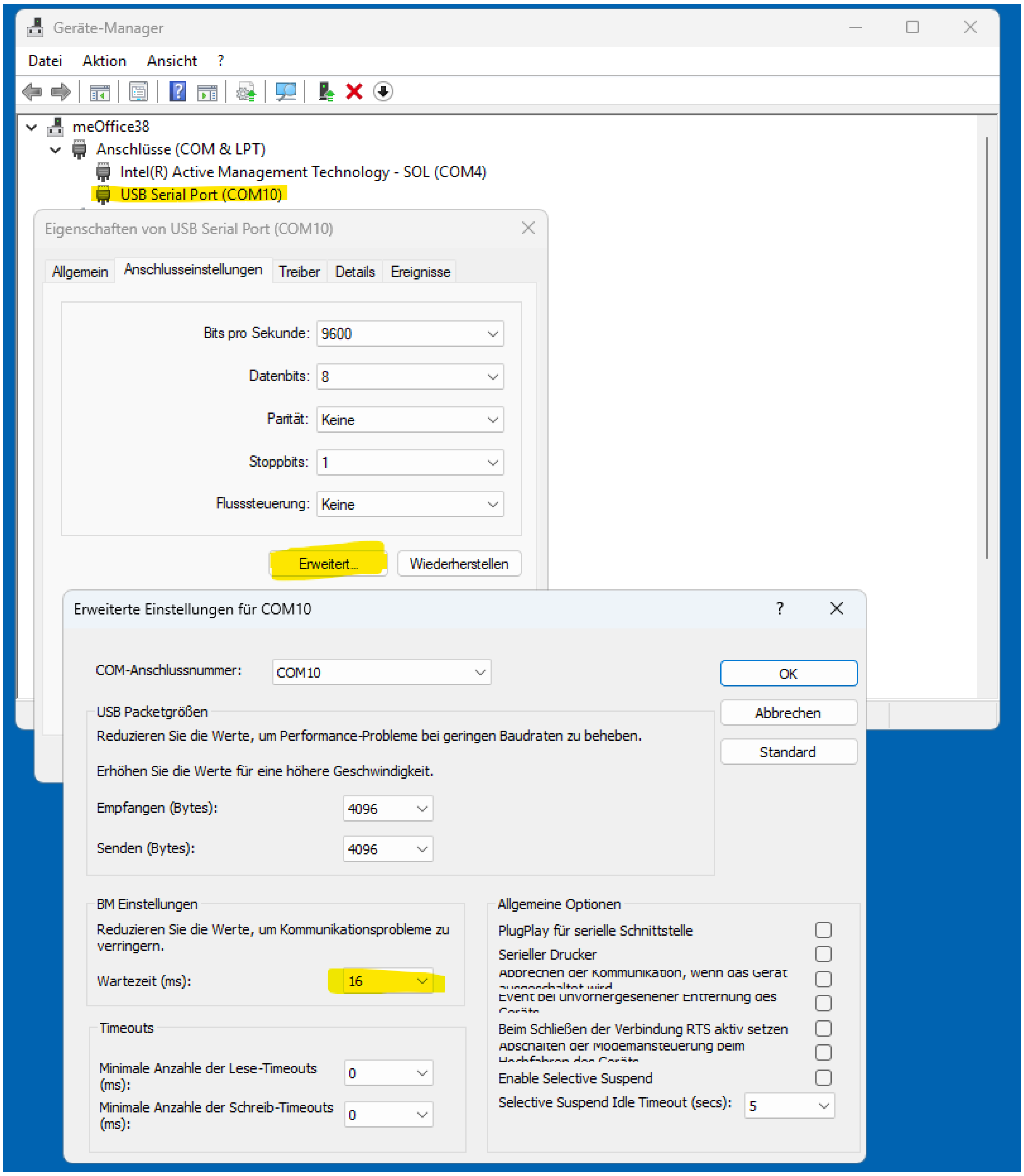
- ➔ Reduce the latency timer value
- ➔ Configure an event character (best option):
  The MeCom protocol has always a 0x0D (Carriage Return) at the end of each package. If this character is configured as event character, then the package is immediately forwarded if this character is received.

### 5.11.1 Windows

On Windows, when using FTDI Chips, it is possible to set the Latency Timer (msec):.

The default value is 16 milliseconds. We recommend setting this value to 3 milliseconds.

➔ On Windows, when using FTDI USB connections. Do not use the virtual comport. Use the FTDI option to connect. This way, this optimization is automatically set.


Geräte-Manager
Datei Aktion Ansicht ?
meOffice38
Anschlüsse (COM & LPT)
Intel(R) Active Management Technology - SOL (COM4)
USB Serial Port (COM10)
Eigenschaften von USB Serial Port (COM10)
Allgemein Anschlusseinstellungen Treiber Details Ereignisse

Bits pro Sekunde: 9600
Datenbits: 8
Parität: Keine
Stoppbits: 1
Flusssteuerung: Keine

Erweitert...
Wiederherstellen

Erweiterte Einstellungen für COM10

COM-Anschlussnummer: COM10

USB Packetgrößen

Reduzieren Sie die Werte, um Performance-Probleme bei geringen Baudraten zu beheben.

Erhöhen Sie die Werte für eine höhere Geschwindigkeit.

Empfangen (Bytes): 4096

Senden (Bytes): 4096

BM Einstellungen

Reduzieren Sie die Werte, um Kommunikationsprobleme zu verringern.

Wartezeit (ms): 16

Timeouts

Minimale Anzahle der Lese-Timeouts (ms): 0

Minimale Anzahle der Schreib-Timeouts (ms): 0

Allgemeine Optionen

PlugPlay für serielle Schnittstelle ☐
Serieller Drucker ☐
Abbrechen der Kommunikation, wenn das Gerät ausgeschaltet wird ☐
Event bei unvorhergesehener Entfernung des Geräts ☐
Beim Schließen der Verbindung RTS aktiv setzen ☐
Abschalten der Modemansteuerung beim Hochfahren des Geräts ☐
Enable Selective Suspend ☐
Selective Suspend Idle Timeout (secs): 5

### 5.11.2 Linux

The settings can be found here:

```
/sys/bus/usb-serial/devices/
```

Go to the folder of your interface. For example, `ttyUSB0`

**Set the event character to 0x0D (Carriage Return):**

You can set the event character by echoing a value to event_char. The value is the ASCII value of the event character (eg 'CR' is 13) plus 256 (this sets the 9th bit to 1 which enables event character support) - so for 'CR' the value is 13 + 256 = 269:

```
# echo 269 > event_char
```

**Set the latency timer**

```
# echo 3 > latency_timer
```

These settings are not persistent.

## 5.12 USB on Linux

On Linux the numbering of the `/dev/ttyUSB#` character devices might change in unpredictable ways. It is recommended to use the corresponding symbolic link in `/dev/serial/by-id/` instead.

Example: `/dev/serial/by-id/usb-FTDI_FT230X_Basic_UART_DK0AC8HL-if00-port0`

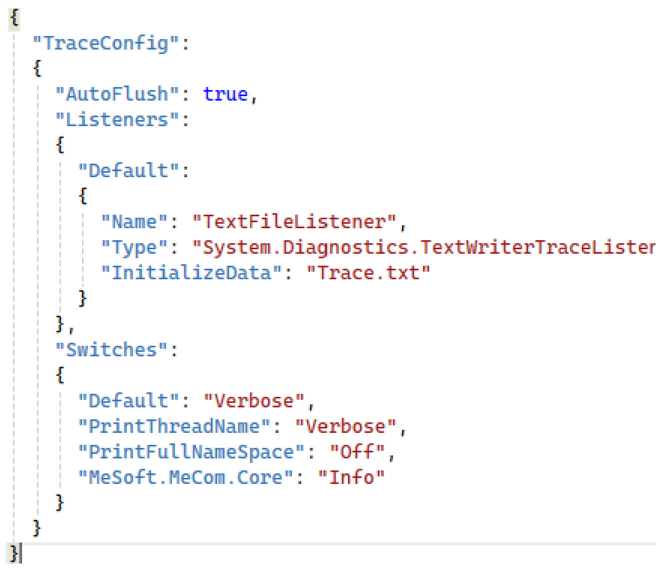
Furthermore, please note that the device server currently does not support hot plugging for USB connected devices on Linux.

## 5.13 Trace

The Device Server creates a Trace.txt file next to the .exe file.

It is possible to control the output using the appsettings.json file.

The Switches can be “Verbose”, “Info”, “Warning”, “Error” or “Off”.



```
{
  "TraceConfig":
  {
    "AutoFlush": true,
    "Listeners":
    {
      "Default":
      {
        "Name": "TextFileListener",
        "Type": "System.Diagnostics.TextWriterTraceLister
        "InitializeData": "Trace.txt"
      }
    },
    "Switches":
    {
      "Default": "Verbose",
      "PrintThreadName": "Verbose",
      "PrintFullNameSpace": "Off",
      "MeSoft.MeCom.Core": "Info"
    }
  }
}
```

## 5.14 LDD-1121, LDD-1124, LDD-1125 Devices

These devices can only accept LDD Service Software commands over the RS-485 channel 2 or the USB connection.

The standard baud rate can only be set on the RS-485 channel 1.

This leads to some limitations, when a TCP/IP Serial Server is used as an interface to the Device Server:

The serial baud rate is limited to 57600, because a speed change is not possible over a TCP/IP Serial Server like the MOXA NPort.

No limitations are present in the following setups:

- USB connection to the Device Server
- USB to RS-485 connection to the RS-485 channel 2.

# A Change History

| **Date of change** | **Doc/Ver-sion** | **Changed/ Approved** | **Software Version** | **Change / Reason** |
|---|---|---|---|---|
| 29 April 2025 | A | ML / SC | 2.20 | • Initial Version |