Interface description

OPC UA for CPX

TRUMPF

Interface description

OPC UA for CPX

Edition **2024-03**

Order Information

Please specify when ordering this document:

Interface description

OPC UA for CPX

Edition 2024-03
Document number 09-14-11-A6-BAen

Address for orders

TRUMPF Laser SE
Vertrieb
Aichhalder Straße 39
78713 Schramberg
Fon: +49 7422 515 - 0
Fax: +49 7422 515 - 436
Internet: http://www.trumpf-laser.com
E-Mail: info@de.trumpf-laser.com

© TRUMPF Laser SE

Good to know

What does this interface description contain?

This interface description presents the OPC UA (Open Platform Communications Unified Architecture) software interface.

You can find the following information in this document:

- Description of how OPC UA works:
  - Properties
  - Architecture
  - Description of data transmission, including an information model of the TRUMPF laser devices and signal descriptions for controlling the TRUMPF laser devices via OPC UA.
- Description of options for easily changing to OPC UA.

How the following operations are carried out using OPC UA is shown, for example:

- Reading data from a TRUMPF laser device.
- Sending control data to a TRUMPF laser device.
- Starting a project in Windows or LINUX.
- Creating a client application with C++.

In the context of creating client applications, different programming options and applications of OPC UA are also described.

For whom is this document intended?

The target group of this document are system integrators and application developers who have already gathered experience with software interfaces (e.g. OPC DA).

Knowledge required

To use this document, knowledge is required of programming languages (C++, .NET, JAVA), of information technology (IT) services and protocols and of Windows or LINUX.

For the implementation of an application example, a Software Development Kit (SDK) from Unified Automation was used.

Further documents

Unified Automation (www.unified-automation.com):

- Documentation of C++-based OPC UA client & server SDK
- Documentation of UaGateway

Table of contents

1	**What is OPC UA?**	**5**
1.1	Properties	5
1.2	OPC UA architecture	6
1.3	Data transfer	8
1.4	Specifications	9
1.5	Sources	10
2	**OPC UA at TRUMPF**	**11**
2.1	Application	12
2.2	Access	12
3	**Applications**	**13**
3.1	UaExpert	13
	Installation and configuration of UaExpert	14
	Using UaExpert	17
4	**Introduction to the development of OPC UA applications**	**18**
4.1	OPC UA basic services	18
4.2	WSDL	19
4.3	SDK	20
5	**Programming applications**	**22**
5.1	Setting up the OPC UA project under Windows	23
5.2	Setting up the OPC UA project under Linux	24
5.3	Client example in C++	25
5.4	Software used	42
6	**Information model of TRUMPF laser devices**	**43**
6.1	Device	43
6.2	Activation	44
	Multi-PLC	44
	PartSN	46
	TVAR	46

	XVAR	47
	OVar	47
6.3	Beam Source	56
	Pump unit	56
	Resonator	57
	SKO	57
6.4	Production	59
	Multi-PLC	60
	Pulse data	63
	System status	64
6.5	Maintenance	65
	General Maintenance	66
	Cleaning the laser circuit	67
	Carbonfilter Nitrogen	68
	Microfilter Laserhead	69
	CPX Battery	70
	Air Cleaner	71
	Active carbon filter	72
	Pump module	73
	Converter	74
	Micro filter set	76
	Main shutter error check	77
	Humidity separator (recirculating air dryer system humidity desiccant)	78
	Conversion module desiccant replacement	79
6.6	Diagnostics	80
	Message history	81
	Cooling	90
	Pulse data/Counter	91
	CalibrationLine	92
	Energy Data	95
	SmartBeo	96
6.7	Programming	97
	Laser program	97
	Parameter set	101
	Pulse shape	104
	PFO program	108
	I-PFO job	117
	PFO file (TruTops PFO)	121
6.8	Set-up	125
	Set-up.writeAssetInformation	125

System settings 126
Switch functions 127
Pulse synchronization 128
6.9 Technology 129
LLKS Data 129
Light path 130
6.10 HELP 135
Help.readAssetInformation 135
Diagnostics 135
Help.DeviceData.getOperatingPoints 136
6.11 Condition Monitoring 136
6.12 Nameplate 138
6.13 QDS 139
QDS.registerQDS 139
QDS.LaserControl_0.delete 140
QDS.LaserControl_0.read 140
QDS.LaserControl_0.reset 141
QDS.LaserControl_0.setBufSize 141
QDS.LaserControl_0.write 141
6.14 Power supply 142

7 Control of the laser devices via OPC UA Multi PLC interface 143
7.1 Input words and output words 145
Input words 145
Output words 154
7.2 Signal descriptions 169
Signals of the input word 0 169
Signals of the input word 1 179
Signals of the input word 2 181
Parameters for laser programs 182
Other input parameters 193
Signals of the alternative input words 14 and 15 195
Signals of output word 0 199
Signals of output word 1 208
Signals of the output word 2 209
Signals of the output word 3 213
Signals of output word 4 216
Signals of the alternative output word 4 218
Signals of output word 5 219
Signals of alternative output word 5 220

Signals of output word 6 221
Signals of output word 7 222
Other output parameters 223
Signals of the alternative output words 14 and 15 233
7.3 Using interface 236
Triggering 236
Timing diagrams 241
Using Input words for laser parameters 256
Messaging and RESET behavior 258
Save energy with Energy Save Concept or optimized standby modes 1-4 262
7.4 Configuring interfaces 263
Setting interface parameters 263
Configuring the sensor relay 266
7.5 Testing options 267
Multi-PLC interface (MPI), overview 267
Diagnostics Dynamic table (XVar) 269

8 Switching from OPC DA to OPC UA 270
8.1 Wrapper and proxies 270
8.2 UaGateway 271
Connecting an OPC UA server to the OPC DA client of FactorySoft 272

1. What is OPC UA?

OPC UA is a software interface which enables easy and universal access to data on various devices.

OPC stands for **O**pen **P**latform **C**ommunications **U**nified **A**rchitecture.

Further information at: www.opcfoundation.org.

1.1 Properties

Below is an overview of its most important properties:

Platform independence OPC UA can be used on systems with Windows or LINUX operating system.

UA Stack The UA Stack written in ANSI C enables OPC UA to be ported to OPC XML DA. Previous problems such as the configuration of COM/DCOM have thus been solved.

The UA Stack enables a secure connection between the server and the client and does not limit the number of programming languages available to an application developer.

Further functions of the UA Stack are the now possible use of multi-threading and single threading, configurable timeouts for every service and the grouping of large data packages (chunking).

Safety OPC UA enables encoding and data integrity according to the current state of the art (for example PKI (**P**ublic **K**ey **I**nfrastructure)).

To ensure a secure data connection, OPC UA provides the options of authentication and authorization:

- Authentication is controlled using X509v3 certificates.
- A user name with corresponding password is required for authorization. Anonymous login is possible.

SOA **S**ervice **O**riented **A**rchitecture.

The specification consists of multiple layers that build logically on each other. The OPC information model has been revised completely for this purpose and now consists of what is called a full mesh network of nodes.

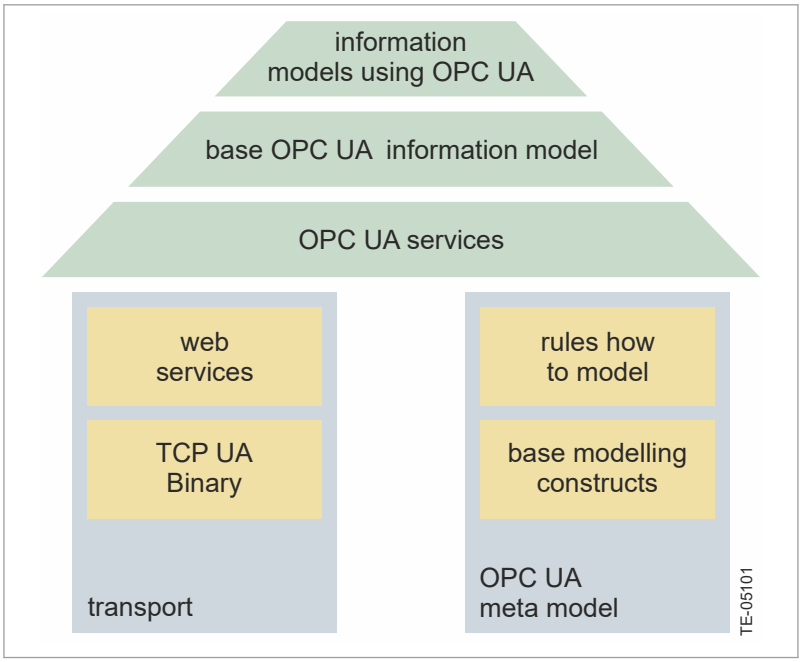
Clients can now check which specifications (DA, AE etc.) are supported by the server used.

Data transfer The data connection is monitored continuously for errors. Interruptions are identified by the server **and** by the client and are dealt with (heartbeat function).

Interruptions no longer result in the loss of data, as the data is stored temporarily in a buffer memory and requested again when the connection is re-established (buffering).

The support of redundancy implemented in OPC UA also contributes towards trouble-free data transmission.

1.2 OPC UA architecture



OPC UA basic architecture Fig. TE-05101

The OPC-UA architecture consists of logical layers built on two bases:

- Transport layer (transport)
- Data modeling (OPC UA meta model)

Transport layer

The transport layer is responsible for sending and receiving data via the network. OPC UA provides two different protocols for this purpose (as of: 06/2012):

- UA TCP
- SOAP/http

Note

The OPC Foundation is focused on being able to implement more protocols in the future.

Data modeling

The data modeling layer defines the rules and basic blocks for the structure of the OPC UA information model.

It also defines the access point to the address space and the basic data types for the type hierarchy.

OPC UA basic services

The OPC UA basic services layer is based on the transport layer and the data modelling layer. It forms the core of the architecture because, as the provider and user of the information model, it represents the interface between the server and the client.

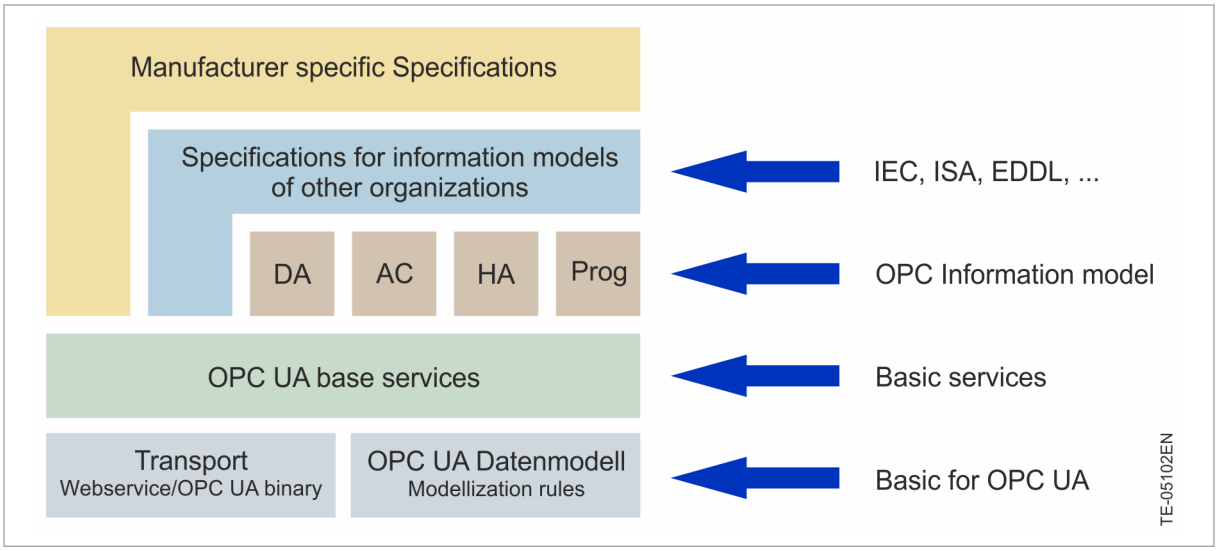
Information model

The "basic layer for an OPC UA information model" is based on the "OPC UA basic services" layer.

The OPC UA information model can be imagined as a network consisting of nodes, which is able to transmit all kinds of maintain information and diagnosis information.

A node points to an object (an object may contain, for example, measured data, trigger alarms etc.).

The following figure shows the OPC architecture again, split up into layers.



OPC UA architecture Fig. TE-05102

Note

OPC UA does not define **what** information is to be exchanged, it defines **how** this information is transferred. What information is exchanged is defined by the corresponding developer of the OPC UA server.

1.3 Data transfer

OPC UA provides two different protocols for data transfer (as of: 05/2012): **UA TCP** and **SOAP/http**.

Note

In anticipation, we would like to mention that the call of the two protocols is different for application developers:

- UA TCP: opc.tcp://host:port
- SOAP/http: https://host/server name.

Note

The Wireshark tool (open source) is available to be able to analyze the OPC UA network protocol. ascolab has added an OPC UA function to it and made it available for download.

Further Information:

- www.ascolab.com/de
- www.wireshark.org

UA TCP

UA TCP is the most important OPC UA protocol. It uses binary messages via the TCP/IP transport layer for communication and is highly efficient, uses only a few resources and is easy to use.

The algorithm from "WS Secure Conversation" is used in the standard case and converted into a binary equivalent. The universal TCP port 4840 is used by default. However, specially configured applications can also use other ports.

SOAP/http SOAP/http is a web service based protocol, which is very easy to use in terms of the firewall, as standard-http/https ports usually do not require further configuration

"WS Secure Conversation" has also been applied for SOAP/http. WSDL (**W**eb **S**ervices **D**escription **L**anguage) can also be used.

Unlike the SDKs (**S**oftware **D**evelopment **K**its), WSDL is available free of charge. However, due to its very complex implementation, TRUMPF recommends not to use WDSL, but to use SDKs instead.

1.4 Specifications

The OPC UA specifications are provided by the OPC Foundation and are normally only relevant for programmers who want to port the OPC UA Stack or program their own stack.

Parts 3, 4 and 5 could also be of interest to experienced application developers.

- Part 1 – Concepts
- Part 2 – Security Model
- Part 3 – Address Space Model
- Part 4 – Services
- Part 5 – Information Model
- Part 6 – Service Mappings
- Part 7 – Profiles
- Part 8 – Data Access
- Part 9 – Alarm and Conditions
- Part 10 – Programs
- Part 11 – Historical Access

1.5 Sources

Internet

- http://www.opcfoundation.org
- http://www.unified-automation.com
- http://www.ascolab.com/

Literature

- OPC Unified Architecture
  Wolfgang Mahnke/Stefan-Helmut Leitner/Matthias Damm,
  Publisher: Springer, ISBN 978-3-540-68898-3
- OPC: Basics, implementation and application
  Frank Iwanitz/Jürgen Lange,
  Publisher: Hüthig, 2nd edition, ISBN 3-7785-2866-1

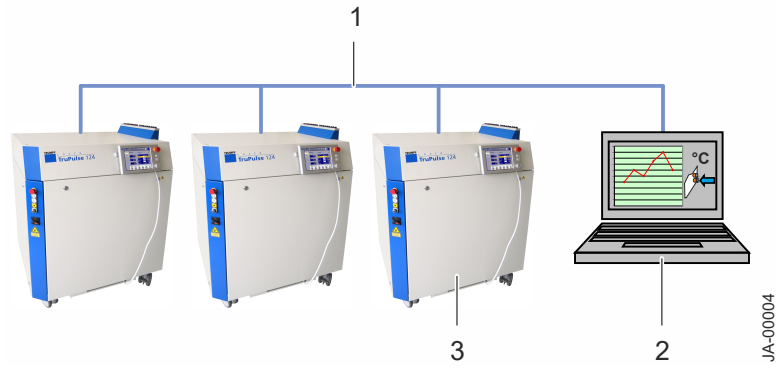
2. OPC UA at TRUMPF

To be able to access the process data of a laser device via PC, TRUMPF uses the OPC UA communication standard. That replaces the older OPC DA standard at TRUMPF.

Note

If a laser device is controlled via OPC, real time is not ensured. The response times can be short or long. They depend on the data traffic in the network and on the capacity utilization of the PC or laptop on which the OPC client is running.

An OPC client for querying and visualizing measured values or for operating the laser device must be connected to the OPC server via Ethernet. To connect the OPC client to the laser device, the Ethernet interface X1 LAN (CPX V1 and V2) or X6B LAN (CPX V3, V4 and V20) are used.



1 Ethernet

2 PC or Laptop with OPC-Client

3 Laser device (1, 2, ... n) with TruControl and integrated OPC-server

Fig. SR-000416

2.1 Application

For OPC UA, the server no longer needs to be installed on a PC, it is integrated directly in the laser device.

The OPC UA server can be addressed via an OPC UA client or a Telnet client.

2.2 Access

To ensure secure access to the OPC UA server, the server supports authentication and authorization.

User level

Depending on the user level, you can define which variables can be read and written on the server. The user level is defined via the user name.

The user level with a leading zero is defined as the user name.

Example: user level = 40: user name: "040".

The passwords for setting the user levels are available from TRUMPF.

For further information:
see software manual "TruControl 1000", Doc.-No. 10-06-80-A1-CR.

3. Applications

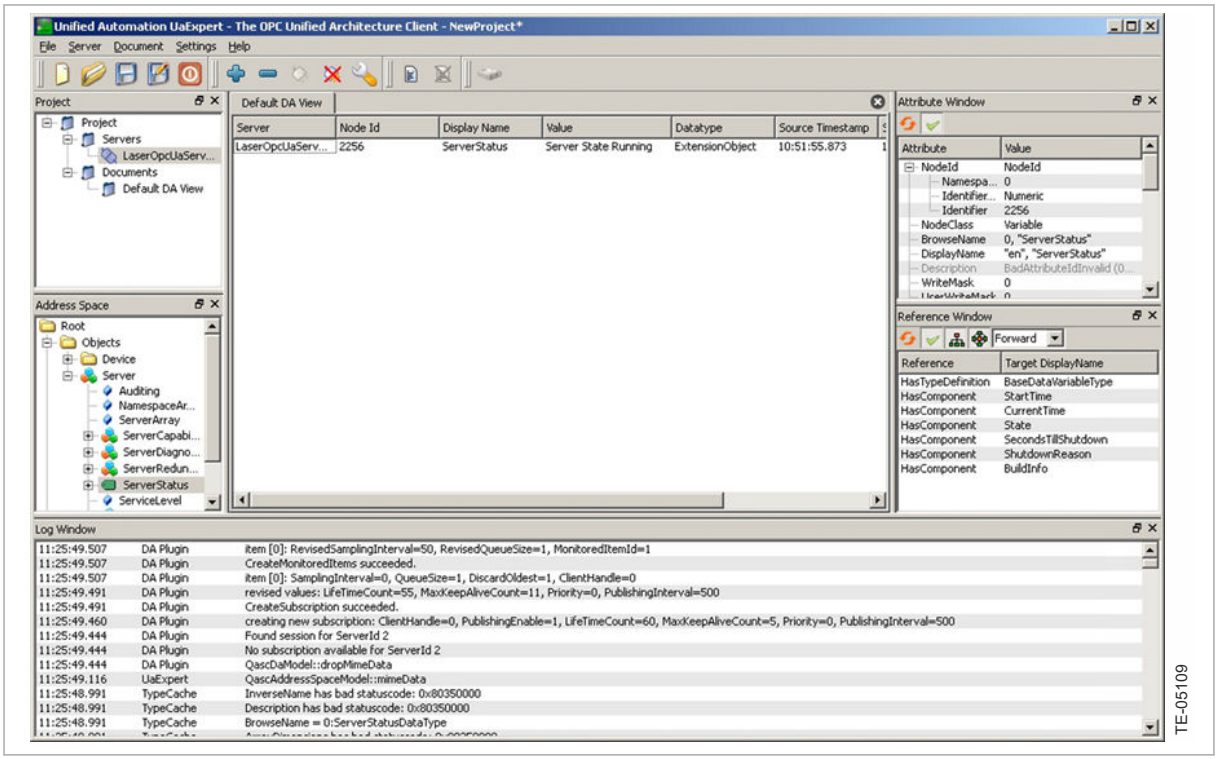
To gain first experience with OPC UA, there are clients that are already available on the market:

3.1 UaExpert

The UaExpert from Unified Automation is recommended as the test client. It is available as freeware for Windows and Linux.

It can be used to easily test the OPC UA functions as well as the connection to the server.

A quick start option is also provided for the client as download to make it easy to get started. The C++ demo server and the ANSI C server SDK are also described in the quick start.



UaExpert: user interface Fig. TE-05109

Installation and configuration of UaExpert

To be able to control a TRUMPF laser device using OPC UA, an OPC UA client that is connected via Ethernet to the OPC UA server is necessary. There are several fee-based and free OPC UA client applications on the market. Installation, configuration and operation are described in the following sections, using UaExpert from Unified Automation GmbH as an example.

Further Information:

- Online help of UaExpert
- www.unified-automation.com

Installation

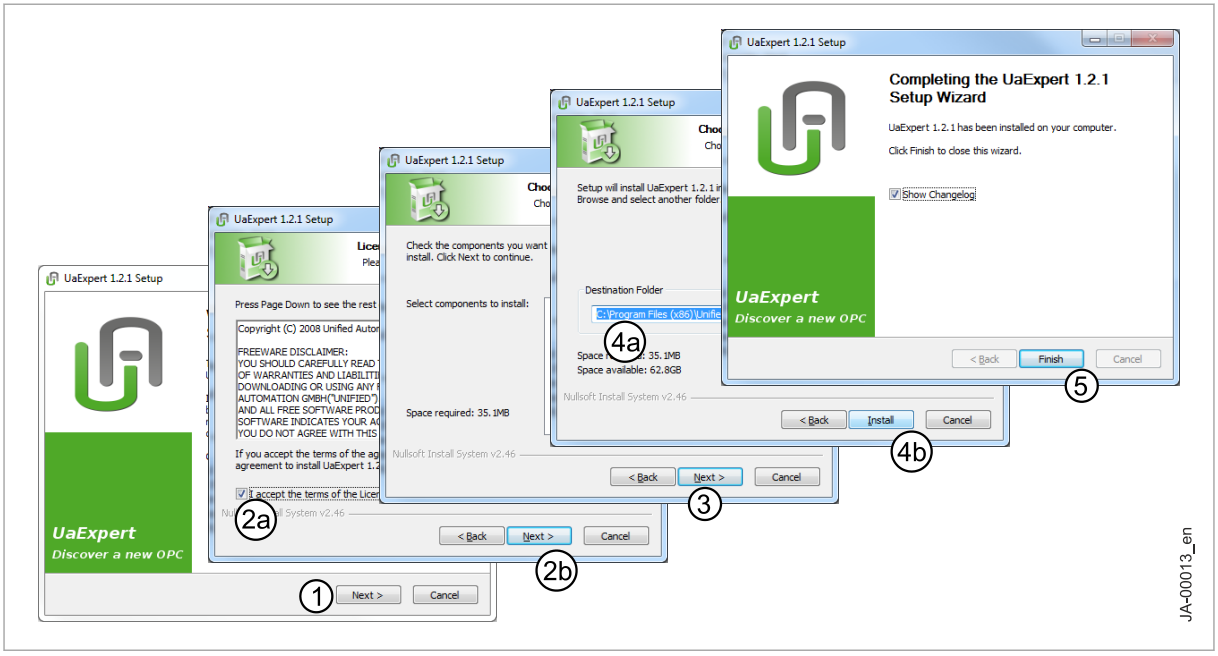


Fig. JA-00013

1. Double-click on the UaExpert installation file.
2. Continue the installation by clicking on the *Continue* button (1).
3. Read the license terms, confirm them (2a) and continue the installation with *Continue* (2b).
4. Select the components to be installed and continue the installation with *Continue* (3).
5. Enter the target directory (4a) and continue the installation by clicking on the *Installation* button (4b).

6. Complete the installation by clicking on the *Complete* button (5).

Configuring the certificate

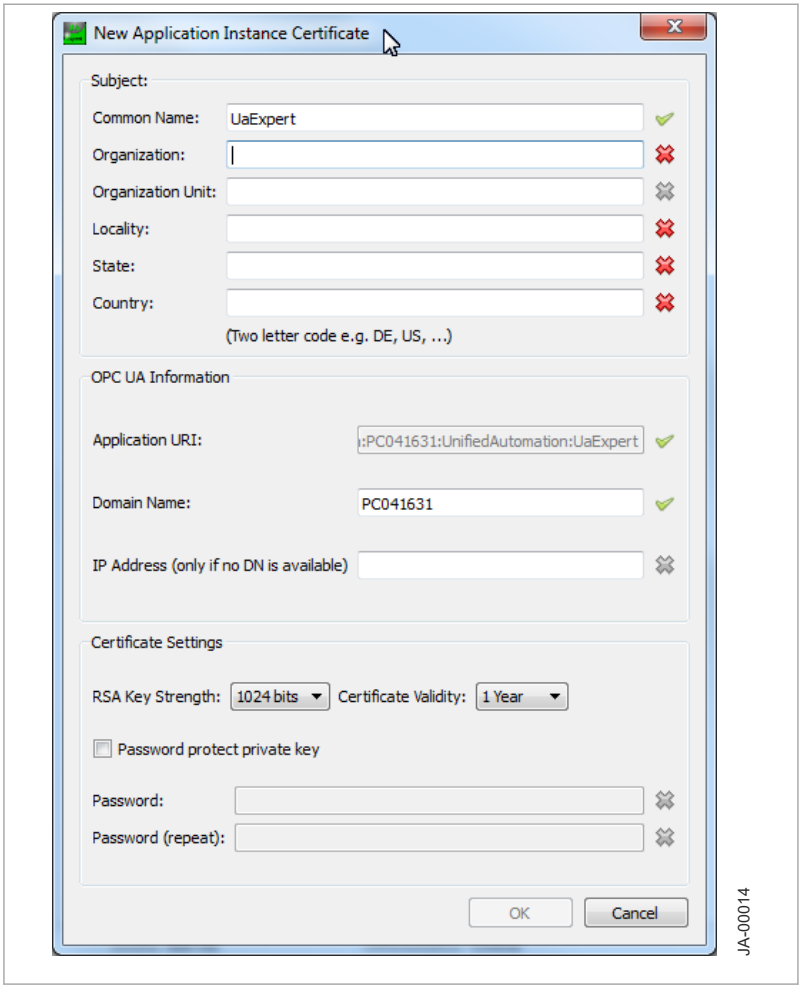


Fig. JA-00014

7. Depending on the configuration, enter the required data on the input mask and confirm it with *OK*.

Adding the server

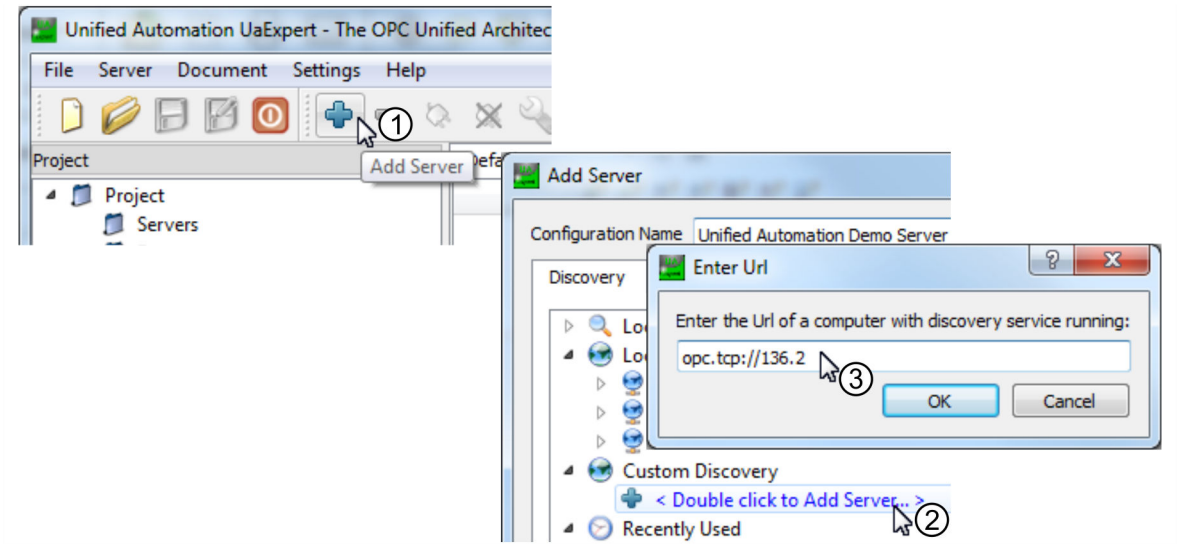


Fig. JA-00015

8. Click on the *Add Server* button.
9. Double-click the marked line: >*Double click to Add Server*.
10. Enter the IP address of the laser device and press *OK* to confirm.
11. Enter user level with preceding "0" into the field *Username* and confirm with *OK*. E.g. „010“ for user level 10 or „060“ for user level 60.
    The user levels in the OPC-UA-server correspond to the user level in the TruControl user interface.
    The password is the respective user level password.

Connecting the server

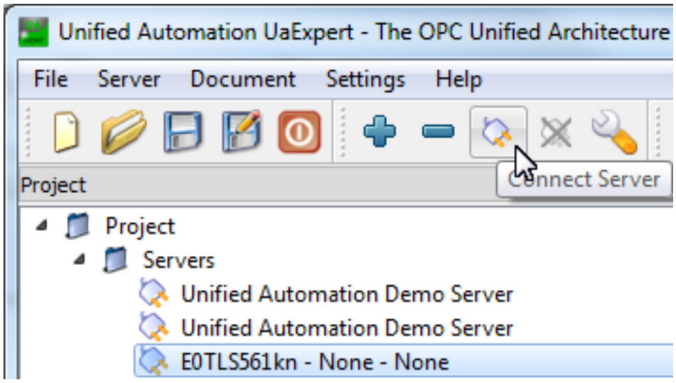
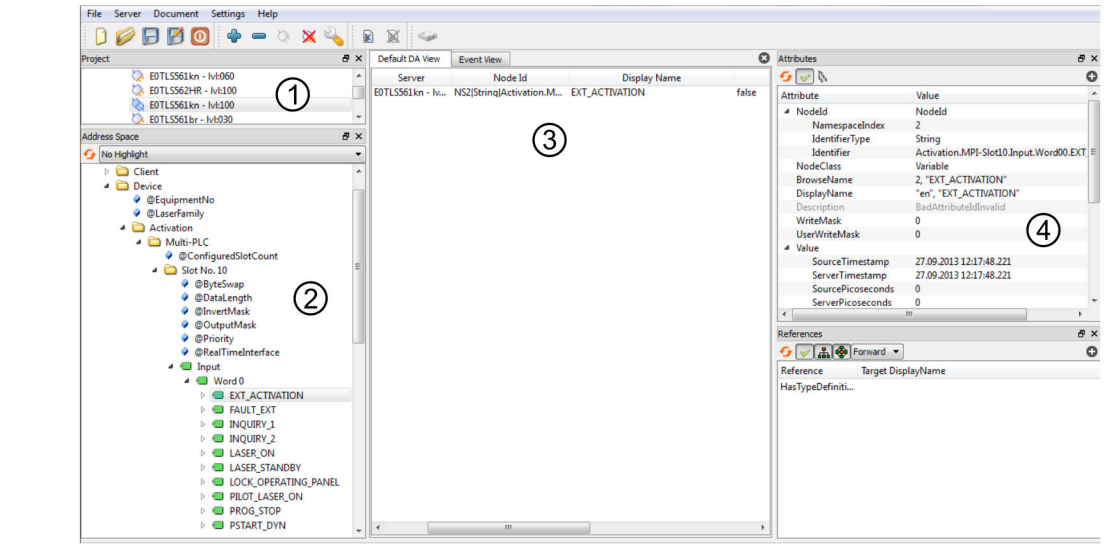


Fig. JA-00016

12. Press the *Connect Server* button.

UaExpert has been installed successfully and configured.

Using UaExpert



1 Project
2 Address Space
3 DA View
4 Attribute Window

Fig. JA-00017

Project Various servers and documents are managed in this window. Project configurations are loaded/saved via the Windows- or Linux-based file system.

Address Space The address space of the server highlighted in the >*Project* window is displayed in this window. The view is similar to a file system. With TRUMPF laser devices, the various variables and functions for querying the operation data are structured like on the user interface in TruControl. All variables and functions are available via the main **Device** node (see "Information model of TRUMPF laser devices", pg. 43).

The signals of the input words and output words for controlling the laser device are to be found in the **Activation** node (see "Activation", pg. 44). Description of the signals of the input and output words: (see "Control of the laser devices via OPC UA Multi PLC interface", pg. 143).

DA View Individual variables can be transferred from the >*Address Space* window using the drag'n'drop method. The values of multiple variables can thus be monitored simultaneously. A compilation of several variables can be saved/loaded as a >*Document* and is managed in the >*Project* window.

Attribute Window The properties of the variables or functions highlighted in the >*Address Space* window are displayed in this window.

4. Introduction to the development of OPC UA applications

4.1 OPC UA basic services

The following 10 basic services (service sets) are available for the development of an OPC UA application:

- Discovery Service Set
- Secure Channel Service Set
- Session Service Set
- Subscription Service Set
- Monitored Item Service Set
- View Service Set
- Attribute Service Set
- Method Service Set
- Node Management Service Set
- Query Service Set

The 10 basic services are described in detail below:

Discovery Service Set This service provides what is called a Discovery server that contains a list of existing OPC UA servers with corresponding end points. Clients can use this service to query the security configurations of the various end points before they connect to an end point or check if the desired OPC UA server exists at all.

Secure Channel Service Set This service is able to set up a "low-level" transport channel and security channel that ensure the integrity and completeness of the information.

Note

The functionality of this service is implemented in the OPC UA Stack and not in the API!

Session Service Set This service is required to establish a secure and trustworthy channel. A session is then even maintained in the event of temporary connection breakdowns and communication failures.

Subscription Service Set

This service can only be called during a session and is linked to the current session.

A client can subscribe to three different types of information:

- If a variable changes.
- If an event occurs.
- Monitoring of collected measured values.

This service is responsible for monitoring the different information sources.

Monitored Item Service Set

This service provides what are called monitored items.

Monitored items are used to subscribe to the types of information described in the **Subscription Service Set** service.

Monitored items are responsible for managing the information sources. To be able to distinguish the respective sources, a monitored item ID is assigned to each source.

View Service Set

Using the View service, you can browse through the server's address space as desired – regardless of the starting and end points.

Attribute Service Set

This service is required in the event of a read or write call. What is to be read or written can be selected here. For example, reading/changing a variable or reading/changing the data type of a variable.

Method Service Set

Using this service, the client can call methods integrated in the server.

Node Management Service Set

This service can be used to create, change or delete nodes.

Query Service Set

This service allows explicit information to be filtered out of the large and dynamic address space of the server.

4.2 WSDL

WSDL (**W**eb **S**ervice **D**escription **L**anguage) is a description language for network services.

The OPC Foundation provides a WSDL package for OPC UA. However, its implementation is very difficult.

Therefore, to create an OPC UA application, TRUMPF recommends using an OPC UA toolkit or SDKs.

4.3 SDK

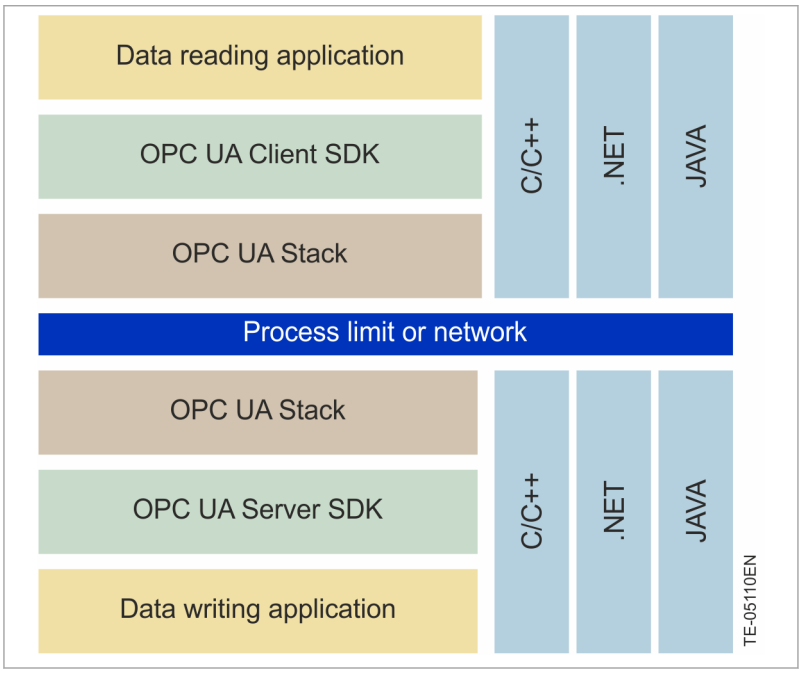
The OPC Foundation supports the developers of OPC UA applications with SDKs (**S**oftware **D**evelopment **K**its). The SDKs are available both as binary version and also as source version.

The first toolkits were provided by the OPC Foundation, Softing and Unified Automation. They conveniently provide all OPC UA functions and defined OPC UA data types.

These toolkits are based on the OPC UA Stack. Therefore, there are three software layers:

- OPC UA Stack
- OPC UA Client/Server SDK
- OPC UA application on top

These layers are not linked to any programming language. The following figure illustrates the layer structure:

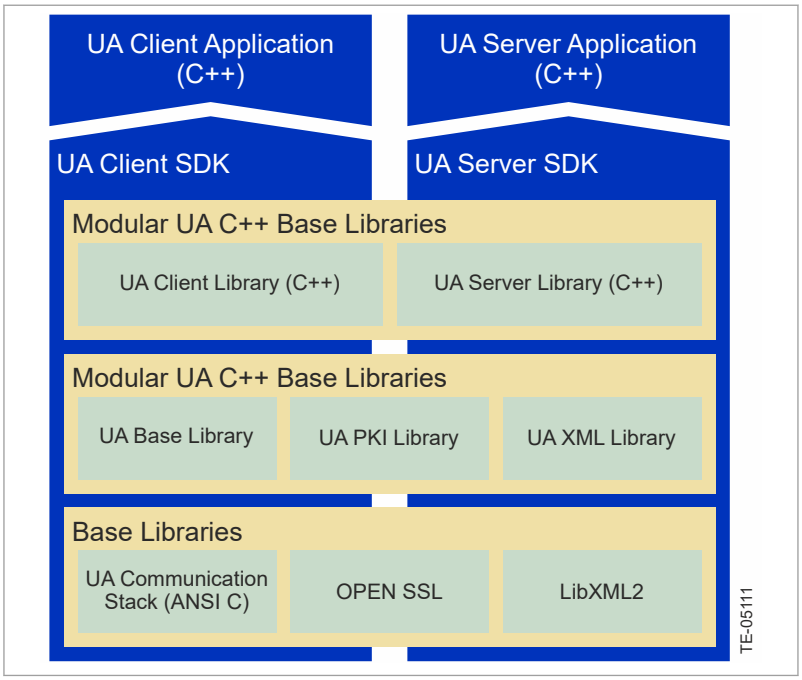


Software layers Fig. TE-05110

The UA Stack represents a secure connection between the applications and ensures that the OPC UA services can be called.

Architecture

An SDK consists of multiple libraries that build logically on each other. The following figures shows the architecture of a C++ SDK as an example.



SDK architecture Fig. TE-05111

The open source projects OpenSSL and LibXML2 are used in this example.

The SDK architecture consists of two SDKs here:

- UA Server SDK and
- UA Client SDK.

Both SDKs are based on the same libraries.

5. Programming applications

Who is this section of interest to?

The following sections shows a programming example for an OPC UA client in C++. If you are using a standard program that already has an OPC UA client, you can skip this section.

The programming example is a simple standard client that supports the browse, read, write and subscribe services.

The use of certificates and the user/password topic are also presented. An SDK from Unified Automation was used for this client. The client can run on Windows and also on Linux.

Requirement

Knowledge of the programming language: C++.

5.1 Setting up the OPC UA project under Windows

A console application is to be created.

If the SDK is installed, some operating system-specific configurations are required, before the implementation can be started.

Conditions

- All include files are in the SDK's installation directory.
- All libraries are in the SDK's installation directory.

1. Select **Win32 console application**.
2. Select **No pre-compiled headers** and **Empty project**.
3. Link to include directory: **'..\include\uastack'**.
4. Link to include directory: **'..\include\uabase'**.
5. Link to include directory: **'..\include\uaclient'**.
6. If the client is to create its own security certificates:

   Link to include: **'..\include\uapki'**.
7. Attach processor definition: **_CRT_SECURE_NO_WARNINGS**.
8. Attach processor definition: **_CRT_SECURE_NO_DEPRACATE**.
9. Enter library directory: **'..\lib'**.
10. Enter library directory: **'..\t'hird-party\openssl\out32.dbg'**.
11. Add additional dependencies
    - uastackd.lib
    - uabased.lib
    - uaclientd.lib
    - libeay32.lib
    - ws2_32.lib
    - rpcrt4.lib
12. If **oleaut32.lib** and **ole32.lib** are required (**oleaut32.lib** and **ole32.lib** are in Visual Studio 2008, project standard):

    install **oleaut32.lib** and **ole32.lib**.
13. If the include **..\'include\uapki'** was added in step 6:

    add **uapkid.lib** .

5.2 Setting up the OPC UA project under Linux

A console application is to be created.

If the SDK is installed, some operating system-specific configurations are required, before the implementation can be started.

1. Link to include directory: **'..\include\uabase'**.
2. Link to include directory: **'..\include\uastack'**.
3. Link to include directory: **'..\include\uaclient'**.
4. If the client is to create its own security certificates:

   Link to include: **'..\include\uapki'**.
5. Enter library directory: **'-L../../lib'**.

   Note

   Incorrect sequences of the additional dependencies can result in problems.
6. Add additional dependencies
   - luaclientd
   - luabased
   - luastackd
   - lssl
   - lcrypto
   - lrt
   - lpthread
   - luapkid (optional)

   The implementation can now be started.

5.3 Client example in C++

The functionality of this client can be split up into the following task blocks:

- Connecting to the server
- Browsing through the server
- Reading values
- Write values
- Subscribing to values
- Change user
- Closing the connection to the server

Note

This section only shows some code fragments as examples.

UA Platform Layer

To be able to use an OPC UA function, initialize the UA platform layer:

```
UaPlatformLayer::init();
```

Clean up the UA platform layer again at the end.

```
UaPlatformLayer::cleanup();
```

Establishing the connection

Establishing the connection is the most complex process in this example. To ensure a secure connection to an OPC UA server, various certificates are loaded via a security setup in this example:

Security certificates

Define directories in which the certificates are stored.

```
#define CLIENT_CERTIFICATE_FILE
 "..\\bin\\PKI\\CA\\certs\\uaclientcpp.der";
#define CLIENT_PKF
 "..\\bin\\PKI\\CA\\private\\uaclientcpp.pem";
#define CRLL
 "..\\bin\\PKI\\CA\\crl\\examplecrl.crl";
#define CTLL
 "..\\bin\\PKI\\CA\\certs\\"
#define SERVER_CERTIFICATE_FILE;
 "..\\bin\\PKI\\CA\\certs\\uaservercpp.der";
```

Now, initialize the PKI provider and load the certificates.

PKI provider and certificates

```
//Status
UaStatus       status;
//Directory variables
UaString       sClientCertificateFile;
UaString       sClientPrivateKeyFile;
OpcUa_StringA  CertificateRevocationListLocation;
OpcUa_StringA  CertificateTrustListLocation;
UaString       sServerCertificateFile;
//Directories are assigned
sClientCertificateFile = CLIENT_CERTIFICATE_FILE;
sClientPrivateKeyFile = CLIENT_PKF;
CertificateRevocationListLocation = CRLL;
CertificateTrustListLocation = CTLL;
sServerCertificateFile = SERVER_CERTIFICATE_FILE;
//Initialize PKI provider
status =
 sessionSecurityInfo.
  initializePkiProviderOpenSSL(
  CertificateRevocationListLocation,
  CertificateTrustListLocation);
//Troubleshooting
if(status.isBad()) { /*1*/
 /*Any error message*/
 return status;
} /*1*/
//Load client certificate
status =
 sessionSecurityInfo.
 loadClientCertificateOpenSSL(
  sClientCertificateFile,
  sClientPrivateKeyFile);
//You have the option of creating
//your own certificate here
if (status.isBad()) { /*2*/
 //Build new Client Certification
 UaPkiIdentity          identity;
 UaPkiCertificateInfo  info;
```

```
UaPkiPublicKey    subjectPubKey;
UaPkiPrivateKey   issuerPrvKey;
UaPkiRsaKeyPair   *pKeyPair;
UaPkiCertificate   *pNewCert;
int   bits = 1024;
identity.commonName =
 "OpcUaClientTrumpf@localhost";
identity.organization = "Trumpf-Laser";
identity.organizationUnit = "EES";
identity.locality = "Schramberg";
identity.state = "Baden-Württemberg";
identity.country = "DE";
info.URI = "opc.tcp://localhost:4841";
info.DNS = "localhost";
info.validTime = 3600*24*365*5; //5 years
//create new keypair
pKeyPair = new UaPkiRsaKeyPair(bits);
subjectPubKey = pKeyPair->publicKey();
issuerPrvKey = pKeyPair->privateKey();
//store key in a PEM encoded file without
//password
pKeyPair->toPEMFile(
 "..\\bin\\PKI\\CA\\private\\uaclientcpp.pem";
 0);
//create new certificate
pNewCert = new UaPkiCertificate(
 info, identity, subjectPubKey, identity,
 issuerPrvKey);
//store certificate in a DER encoded file
pNewCert->toDERFile(
 "..\\bin\\PKI\\CA\\certs\\uaclientcpp.der");
//Load new client certificate
status =
 sessionSecurityInfo.
 loadClientCertificateOpenSSL(
 sClientCertificateFile, sClientPrivateKeyFile);
if(status.isBad())
{ /*3*/
```

```
  /*Any error message*/
  return status;
 } /*3*/
} /*2*/
//Load server certificate
status =
 sessionSecurityInfo.
  loadServerCertificateOpenSSL(
  sServerCertificateFile);
if(status.isBad()) { /*4*/
 /*Any error message*/
 return status;
} /*4*/
```

Note

The same certificates must be available to the client and the server.

Finding the server

Check if the server exists with the desired security configurations. This information can be called via the **Discovery server** already mentioned. The **findServer** and **getEndpoints** member functions provide support. The **FindServer** function returns all known addresses of OPC UA servers via the **Discovery server**. The server address (e.g. 'opc.tcp://localhost:4840') has already been assigned **uaUrl**.

```
UaStatus                status;
ServiceSettings         serviceSettings;
SessionSecurityInfo    sessionSecurityInfo;
UaApplicationDescriptions
                        applicationDescriptions;
UaEndpointDescriptions endpointDescriptions;
OpcUa_UInt32           endpointCount;
OpcUa_UInt32           serverCount;
m_pDiscovery = new UaDiscovery();
//Find server
status = m_pDiscovery->findServers(
 serviceSettings,
 uaUrl,
 sessionSecurityInfo,
 applicationDescriptions);
```

If the call is successful, query the security configurations of the known end points using the **getEndpoint** function.

```
//Fetch end point information
status = m_pDiscovery->getEndpoints(
 serviceSettings,
 uaUrl,
 sessionSecurityInfo,
 endpointDescriptions);
//Repeat procedure as long as incorrect
//entries are made
do { /*5*/
 endpointCount = 0;
 serverCount = 0;
 //Loop to output all servers+
 //end points
 for (OpcUa_UInt32 i = 0;
  i < applicationDescriptions.length(); i++)
  { /*6*/
   //Output the first 10 servers+end points
   for (OpcUa_UInt32 j=0;
    j < endpointDescriptions.length(); j++ )
    { /*7*/
     //Limit to ten end points
     if (endpointCount >= 10) { /*8*/
      break;
     } /*8*/
     //Output end point configurations
     //(own method)
     printEndpointDescription(
      applicationDescriptions,
      endpointDescriptions,
      endpointCount,
      serverCount);
     endpointCount++;
    } /*7*/
    serverCount ++;
   } /*6*/
   //Select and connect to end point
```

```
    //(own method)
    ret = connectOnEndpoint(
     sessionSecurityInfo,
     applicationDescriptions,
     endpointDescriptions,
     endpointCount);
    //User wants to exit client
    if(ret == 1)
     return -1;
 } /*5*/ while (ret == -1);
/*Query for errors*/
```

Establishing the connection

The own **connectOnEndpoint** method is used to connect to the end point entered. To do this, the **Callback** class must be addressed. It was applied from the demo client version of the SDK and is for checking the status of the connection and indicating any change. A connection cannot be established without this callback.

```
UaStatus     status;
char         cEndpoint[20];
char         *ptrHelper;
int          endpoint;
UaString     uaUrl;
SessionConnectInfo  sessionConnectInfo;
ServiceSettings     serviceSettings;
//Create new instances of UaSession
//and Callback classes
m_pUaSession = new UaSession();
m_pCallback = new Callback();
memset(cEndpoint, 0, sizeof(cEndpoint));
//Connect automatically if there is only one
end point
//or server
if((endpointDescriptions.length() == 1) &&
 (applicationDescriptions.length() == 1)) { /*9*/
 endpoint = 0;
} /*9*/
else { /*10*/
 //Let user select the end point
 cout << "Connect to Endpoint: ";
```

```
 cin >> cEndpoint;
 //Small method to ask if user
 //wants to exit
 if(doExit(cEndpoint))
  return 1;
 ptrHelper = cEndpoint;
 //Convert to int with sscanf
 if(sscanf(ptrHelper,"%d",&endpoint) == -1)
 {/*11*/
  cout << "Error: sscanf failed! " << endl;
 } /*11*/
 //Query whether a valid end point was
 //entered
 if(endpoint < 0 || endpoint > 10 ||
  endpoint > endpointCount) { /*12*/
   /*Any error message*/
   /*Cleanup*/
   return -1;
 } /*12*/
} /*10*/
//Fetch end point Url
uaUrl =
 &endpointDescriptions[endpoint].EndpointUrl;
//Configure security information to the
//end point's security configurations
sessionSecurityInfo.serverCertificate =
 endpointDescriptions[endpoint].
 ServerCertificate;
sessionSecurityInfo.sSecurityPolicy =
 &endpointDescriptions[endpoint].
 SecurityPolicyUri;
sessionSecurityInfo.messageSecurityMode =
 endpointDescriptions[endpoint].SecurityMode;
//User identification (own method)
UserIdentification(sessionSecurityInfo);
//Connect to the end point
status = m_pUaSession->connect(
 uaUrl,
```

```
 sessionConnectInfo,
 sessionSecurityInfo,
 m_pCallback);
//Troubleshooting
if(status.isBad()) { /*13*/
 /*Any error message*/
} /*13*/
return 0;
```

User identification

Prior to establishing the connection, user identification may be required using the **UserIdentification** method. To be able to log in with user name and password, the end point URL and the server certificate must be read out of the end point configurations first (see "Establishing the connection"). In the following example, you can select whether to authorize yourself or not.

```
Service   *service;
char      username[50];
char      password[50];
char      yn;
service = new Service(null, null);
memset(username, 0, sizeof(username));
memset(password, 0, sizeof(password));
//Does the user want to identify himself/
herself?
cout << endl;
cout << "User Identification? [y/n] ";
cin >> yn;
cout << endl;
//Check entry
if(yn == 'y') { /*14*/
 cout << "Please insert Username: ";
 cin >> username;
 cout << "Please insert Password: ";
 //Own method for covering password
 service->coverPassword(password);
 //Set user name and password
 sessionSecurityInfo.setUserPasswordUserIdentity(
  username, password);
} /*14*/
else if (yn == 'n') { /*15*/
 /*Default*/
} /*15*/
else { /*16*/
 /*Troubleshooting*/
} /*16*/
delete service;
```

Browse The browse function can be used to browse through the entire server and collect information on the various nodes. The starting point and the direction are variable. In the following example, the root directory is defined as the starting point. The factory setting is used as the browse direction.

Initialize browsing with the starting point.

```
//Start with root
UaNodeId startingNode(OpcUaId_RootFolder);
//Start browsing
browse(startingNode, 0);
```

The UA Browse Member function is called in the **Browse** method and the node information is output.

```
UaStatus                  status;
UaByteString              continuationPoint;
UaReferenceDescriptions   referenceDescriptions;
ServiceSettings           serviceSettings;
BrowseContext             browseContext;
//Call browse service
status = m_pUaSession->browse(
 serviceSettings,
 startingNode,
 browseContext,
 continuationPoint,
 referenceDescriptions);
if(status.isBad()) { /*17*/
 /*Any error message*/
} /*17*/
else { /*18*/
 OpcUa_UInt32 i;
 //Output the node information found
 for (i=0; i<referenceDescriptions.length();
  i++) { /*19*/
  //At Root: Level = 0 etc.
  cout<< "Browse Level: " << level << endl;
  //Node information
  cout<< "Node: ";
   UaNodeId referenceTypeId(
   referenceDescriptions[i].ReferenceTypeId);
  cout<< "[Ref = "
```

```
    << referenceTypeId.toString().toUtf8()
    << "]"<< endl;
   UaQualifiedName browseName(
    referenceDescriptions[i].BrowseName);
   cout<<"BrowseName: "<<
    browseName.toString().toUtf8()<< endl;
   cout<< "Reference Description: ";
   if (referenceDescriptions[i].NodeClass &
    OpcUa_NodeClass_Object)
     cout << "Object" << endl;
   if (referenceDescriptions[i].NodeClass &
    OpcUa_NodeClass_Variable)
     cout << "Variable" << endl;
   /*etc.*/
   browse(
    referenceDescriptions[i].NodeId.NodeId,
    level ++);
  } /*19*/
  //If the continuation point is greater than
zero
  while
   (continuationPoint.length() > 0 ) { /*20*/
   //Call browseNext Service
   status = m_pUaSession->browseNext(
   serviceSettings,
   OpcUa_False,
   continuationPoint,
   referenceDescriptions);
   if (status.isBad()) { /*21*/
    /*Any error message*/
   } /*21*/
   else { /*22*/
    //Output node information found
    for
     (i=0; i<referenceDescriptions.length(); i++)
     { /*23*/
     /*See Browse*/
    } /*23*/
```

```
    } /*22*/
  } /*20*/
} /*18*/
```

Reading values

The concept of OPC UA follows the rule that all nodes are to be found in certain namespaces.
Namespaces are designated using numbers only.

A node ID is required to be able to read a value. A node ID consists of an identifier and a namespace index. An identifier can be numeric, a string, a GUID or a byte array. The namespace indices 0 and 1 have been permanently predefined by the OPC Foundation.

In the following example, the name of the server is read with the node ID with the identifier 2261 and the namespace 0.

```
UaStatus          status;
UaReadValueIds    nodesToRead;
UaDataValues      values;
UaDiagnosticInfos diagnosticInfos;
ServiceSettings   serviceSettings;
char              nodeIdBuffer[100];
```

First set the node ID of the value to be read.

```
//Set node ID using UA Member function
nodeId.setNodeId(2261.0);
```

Note

The node ID can also be read via console:

```
//Enter node ID:
//The node ID was entered using a special
//OPC UA XML schema (here: i=2261),
//to make it easier to
//convert it later on.
cout << endl;
cout << "Please insert NodeId to read: ";
cin >> nodeIdBuffer;
cout << endl;
//Convert the entered ID to a node ID
UaNodeId nodeId =
 UaNodeId::fromXmlString(nodeIdBuffer);
```

An array is provided with the **read** service that contains all node IDs to be read

```
//Create an array for all node IDs
//to be read
nodesToRead.create(1);
//Copy all node IDs to the array
nodeId.copyTo(&nodesToRead[0].NodeId);
//Set attribute fieId -> the value
//is to be read
nodesToRead[0].AttributeId =
 OpcUa_Attributes_Value;
//Call read service
status = m_pUaSession->read(
 serviceSettings,
 0,
 OpcUa_TimestampsToReturn_Both,
 nodesToRead,
 values,
 diagnosticInfos);
if(status.isBad()) { /*24*/
 /*Any error message*/
} /*24*/
else { /*25*/
 //Query status code of the value
 status = values[0].StatusCode;
 if(status.isGood()) { /*26*/
  //Fetch the value
  UaVariant tempValue = values[0].Value;
  //Output value
  cout << "Variable "
  << nodeId.toString().toUtf8()
  << " value = "
  << tempValue.toString().toUtf8() << endl;
 } /*26*/
 else { /*27*/
  /*Any error message*/
 } /*27*/
} /*25*/
```

Write values

The concept of the node IDs also applies to writing values. To overwrite a value, its data type must be known.

```
UaStatus           status;
UaVariant           tempValue;
UaWriteValues      nodesToWrite;
UaStatusCodeArray   results;
UaDiagnosticInfos   diagnosticInfos;
ServiceSettings    serviceSettings;
```

(See "Reading values": replace **nodesToRead** with **nodesToWrite**)

If the data type is known, it can be set in the simplest case with:

```
//Set data type UInt32
tempValue.setUInt32(value);
```

If the data type is not known it can also be determined using the **read** service (see "Reading values").

```
//Copy data type to the array
tempValue.copyTo(&nodesToWrite[0].Value.Value);
//Call write service
status = m_pUaSession->write(
 serviceSettings,
 nodesToWrite,
 results,
 diagnosticInfos);
if( status.isBad()) { /*28*/
 /*Any error message*/
} /*28*/
else { /*29*/
 if (OpcUa_IsGood(results[0])) { /*30*/
  /*Value written successfully!!*/
 } /*30*/
 else { /*31*/
  /*Any error message*/
 } /*31*/
} /*29*/
```

Subscribing to values

Values can be subscribed to. This means the client is notified every time a subscribed value changes. The Subscription and Monitored Item Service is used for this purpose. The Callback class must also be taken into account.

```
UaStatus               status;
SubscriptionSettings   subscriptionSettings;
subscriptionSettings.publishingInterval = 1000;
ServiceSettings        serviceSettings;
//Set NodeId, see read/write
UaMonitoredItemCreateRequests
 monitoredItemCreateRequests;
UaMonitoredItemCreateResults
 monitoredItemCreateResults;
//Call subscription service
status = m_pUaSession->createSubscription(
 serviceSettings,
 m_pCallback,
 0,
 subscriptionSettings,
 OpcUa_True,
 &m_pSubscription);
if(status.isBad()) { /*32*/
 /*Any error message*/
} /*32*/
else { /*33*/
 //Create a monitored item request for
 //the node IDs
 monitoredItemCreateRequests.create(1);
 nodeId.copyTo(
  &monitoredItemCreateRequests[0].
  ItemToMonitor.NodeId);
 //Configure the request
 monitoredItemCreateRequests[0].
  ItemToMonitor.AttributeId =
  OpcUa_Attributes_Value;
 monitoredItemCreateRequests[0].
  MonitoringMode =
  OpcUa_MonitoringMode_Reporting;
```

```
 monitoredItemCreateRequests[0].
  RequestedParameters.ClientHandle = 1;
 monitoredItemCreateRequests[0].
  RequestedParameters.SamplingInterval = 500;
 monitoredItemCreateRequests[0].
  RequestedParameters.QueueSize = 1;
 //Call monitored item service
 status = m_pSubscription->createMonitoredItems(
  serviceSettings,
  OpcUa_TimestampsToReturn_Both,
  monitoredItemCreateRequests,
  monitoredItemCreateResults);
 if(status.isBad()) { /*34*/
  /*Any error message*/
 } /*34*/
 else { /*35*/
  //Check result status
  status = monitoredItemCreateResults[0].
  StatusCode;
  if (status.isBad()) { /*36*/
   /*Any error message*/
  } /*36*/
  else { /*37*/
   cout << " MonitoredItemId: " <<
    monitoredItemCreateResults[0].
    MonitoredItemId << endl;
   //(No loop containing the subscription)
   //Exit subscription
   Status = m_pUaSession->deleteSubscription(
    serviceSettings, &m_pSubscription);
  } /*37*/
 } /*35*/
} /*33*/
```

Change user

It is possible to change the user level or the user name during a session. The **changeUser** member function is available for this purpose during a session. To be able to call it correctly, the user and the password are reset in the **UaUserIdentityToken**.

```
UaStatus         status;
ServiceSettings   serviceSettings;
char              username[50];
char              password[50];
memset(username, 0, sizeof(username));
memset(password, 0, sizeof(password));
UaUserIdentityTokenUserPassword  *iuToken;
iuToken = new UaUserIdentityTokenUserPassword();
/*…user name- and password query…*/
//Set new user name and password
iuToken->sUserName = username;
iuToken->sPassword = password;
UaUserIdentityToken *iToken =
 (UaUserIdentityToken *)iuToken;
//Change user
status = m_pUaSession->changeUser(
 serviceSettings,
 iToken);
if(status.isBad()) { /*38*/
 /*Any error message*/
} /*38
```

Ending the connection

The session can be exited with a simple **disconnect**. If you also want to cancel still active subscriptions of the session, you must submit **OpcUa_True** as the second argument. If you submit **OpcUa_False** instead, you can transfer your subscription to a different session.

```
UaStatus           status;
ServiceSettings  serviceSettings;
status = m_pUaSession->disconnect(
 serviceSettings, OpcUa_True);
/*Any troubleshooting*/
```

5.4 Software used

- Visual Studio 2008 project
- Header:
  - Opcua_client.h
  - Security_config.h
  - Callback.h
  - Data_types.h
  - Opcua_client_service.h
- Source:
  - Opcua_client_main.cpp
  - Opcua_client.cpp
  - Opcua_client_security.cpp
  - Data_types.cpp
  - Opcua_client_service.cpp

6. Information model of TRUMPF laser devices

The following chapters contain the description of the information model of the TRUMPF laser devices with TruControl 1000 control. This description primarily contains lists of the available variables and methods. The availability of the variables and methods depends on the configuration and the type of the laser device.

6.1 Device

Variables which can be found directly under "Device" node can be read out without special rights even with anonymous access. These variables are available to all laser types.

UaNodeId	Description
EquipmentNo	Equipment number of the laser device
LaserFamily	Laser device family (20-xx-…) Example: 20-10-03-A203: TruDisk 3002.
LaserName	Name of the laser device, as specified in system parameter WL_Name, available as of TC 2.18.0
LastMessage	Currently active message code (32-bit) including ID/NUM. If there is no message, "0" is returned as the value.
LastMessage.CompID	Component ID from the currently active message
LastMessage.CompNUM	Component NUM from the currently active message
LastMessage.MsgCode	Code for the ID/NUM of the currently active message
LastMessage.ShortText	Short text for the currently active message. The language used is defined by the client when establishing a connection.
LastMessage.Timestamp	Time stamp in UTC of the currently pending message (fault, monitoring, …)
LastMessage.Type	Type of currently active message (fault, monitoring, …)

Tab. 1

6.2 Activation

All parameters and methods for controlling the laser device are to be found under "Activation" node. The "Activation" node is divided into the following nodes:

- Multi-PLC: Parameter of the Multi-PLC-interfaces that have been configured as OPC interface.
- Part SN: Traceabiltity-Data: workpiece serial number.
- TVAR: statical setting of the laser program parameters, max. 64 parameters for max. 200 laser programs.
- XVAR: dynamical setting of the laser program parameters, max 64 parameters.
- OVAR: object control for PFO or I-PFO.

Multi-PLC

All input variables under "Multi-PLC" node, including the input variable itself, are writable. All other variables are read-only variables. The laser can be controlled with the input variables according to the MPI interface description (see "Control of the laser devices via OPC UA Multi PLC interface", pg. 143).

Prerequisites

- User level ≥ 50
- An MPI slot must be configured as an OPC UA interface (see "Configuring interfaces", pg. 263).
- Only one OPC UA client may be connected to the laser control for each MPI slot.

UaNodeId	Type	Description
Activation. Multi-PLC.ConfiguredSlotCount	UInt16	Number of configured MPI slots

Tab. 2

Under "Multi-PLC" node, one node is available for all MPI slots configured as an OPC slot. Up to 32 MPI slots can be configured.

UaNodeId	Description
Activation.MPI-Slot01	Node for slot 1
Activation.MPI-Slot02	Node for slot 2
:	:
Activation.MPI-Slot32	Node for slot 32

Tab. 3

From these nodes, only those configured in the MPI configuration in TruControl 1000 as an OPC slot are available.

The available variables are described in the following, for MPI slot 12 as an example, which was configured as an OPC slot with 16 words.

In all the following variables, the UaNodeId corresponds to the designation of the variable plus the "Activation" prefix..

Example:
"MPI-Slot12.DataLength" is equivalent to UaNodeId "Activation.MPI-Slot12.DataLength".

UaNodeId	Description
MPI-Slot12.ByteSwap	**Byte swap** (see "Setting interface parameters", pg. 263)
MPI-Slot12.DataLength	**Data length** (see "Setting interface parameters", pg. 263)
MPI-Slot12.InvertMask	**Inverting mask** (see "Setting interface parameters", pg. 263)
MPI-Slot12.OutputMask	**Output mask** (see "Setting interface parameters", pg. 263)
MPI-Slot12.Priority	**Priority** (see "Setting interface parameters", pg. 263)
MPI-Slot12.RealTimeInterface	**Real-time interface** (see "Setting interface parameters", pg. 263)
MPI-Slot12.Input	Array the the input words (see "Input words", pg. 145)
MPI-Slot12.Input.Word00	Input word 0 (see "Input words", pg. 145)
:	:
MPI-Slot12.Input.Word31	Input word 31 (see "Input words", pg. 145)
MPI-Slot12.Output	Array with the output words (see "Output words", pg. 154)
MPI-Slot12.Output.Word00	Output word 0 (see "Output words", pg. 154)
:	:
MPI-Slot12.Output.Word31	Output word 31 (see "Output words", pg. 154)
MPI-Slot12.UaSessionID	Session ID of a connected client. If MPI-Slot12.UaSessionID = Client.UaSessionID, the client with this session ID is granted exclusive write access to the variables below the node Activation.MPI-Slot12.Input. Exclusive write access is assigned to this client as long as the client is connected to the OPC-UA server. If the connection from this client to the OPC-UA server is terminated, the exclusive write access is released again. If session ID = -1, no client is connected.

Variables of the Activation.MPI-Slotx node, example x = 12, data length 16 words Tab. 4

PartSN

Activation.resolvePartSn

Method to issue the workpieces serial number.

Requirement

System parameter "FO_PFO_ScanLine" = "3.

Parameter	Type	Description
PartRefIndex	Int16	Reference index for the workpiece serial number.

Parameters (input) Tab. 5

Parameter	Type	Description
PartSN_Raw	Byte-String	Raw data of the workpiece serial number

Parameters (output) Tab. 6

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 7

TVAR

Requirement

System parameter "TVAR" = "TRUE".

UaNodeId	Description
Activation.TVAR.Prog001	Array with 64 float values for laser program 1
Activation.TVAR.Prog002	Array with 64 float values for laser program 2
:	:
Activation.TVAR.Prog200	Array with 64 float values for laser program 200

Tab. 8

XVAR

Requirement

System parameter "XVAR" = "TRUE".

UaNodeId	Description
Activation.XVAR.01	Variable XVar 1, 16 bit
Activation.XVAR.02	Variable XVar 2, 16 bit
:	:
Activation.XVAR.64	Variable XVar 64, 16 bit

Tab. 9

Further information on XVar: Parameters **PLC output (CV) x words** and **XYar.1** .... **XVar.32**(see "Other output parameters", pg. 223) and for diagnostics: (see "Diagnostics Dynamic table (XVar)", pg. 269).

OVar

Requirement

- TruControl 1000 version ≥ TC 3.52
- System parameter *OVAR* = "TRUE".
  If system parameter *OVAR* = "TRUE", for example, correction values from the image processing can be transmitted directly to a PFO or I-PFO.

The transfer of the adjustment values via OVar to a PFO is carried out in the laser program *PFO mode* = `Objektsteuerung` activated.
The following methods are available for this:

- setPFOOVar
- getPFOOVar
- getPFOOVarCounter

The transfer of the correction values via IpfOVar to an I-PFO is carried out in the laser program with *PFO mode* = `IPFO-Objektsteuerung` activated.
The following methods are available for this:

- setIPFOOVar
- getIPFOOVar
- getIPFOOVarCounter

The methods are available under the *Activation.OVar* node.

setPFOOVar

Method for transferring object values to a PFO.

Parameter	Type	Description
ListNbr	UInt16	Number in the OVar list. The OVar list number corresponds to the PFO number.
StartNbr	UInt16	First OVar data record number. Value range: 1 - 500.
TransferMode	UInt16	Data transfer mode Value must be = 0.
OVar	Ovar[]	Data structure of Ovar type. Number of data structure elements: 1 - 500, depends on the value of the StartNbr parameter.

Parameters (input) Tab. 10

Parameter	Type	Description
PfoProgNbr	UInt16	PFO program number[1)]
PfoProgramRuns	UInt16	Number of PFO program runs
Power	Double	Power [W]
LeistungOffset	Double	Power offset [W]
TransX	Double	PFO translation X [mm]
TransY	Double	PFO translation Y [mm]
TransZ	Double	PFO translation Z [mm]
RotX	Double	PFO rotation X [degrees]
RotY	Double	PFO rotation Y [degrees]
RotZ	Double	PFO rotation Z [degrees]
ScaleX	Double	PFO scaling X (100% or -100%). Only mirroring is permitted.
ScaleY	Double	PFO scaling Y (100% or -100%). Only mirroring is permitted.
ScaleZ	Double	PFO scaling Z (100% or -100%). Only mirroring is permitted.
Defocus	Double	PFO defocus [mm]
WobbleProgNbr	Double	Wobble program number
WobbleWidth	Double	Wobble width [mm]
WobbleLength	Double	Wobble length [mm]
WobbleFrequency	Double	Wobble frequency [Hz]
ApplicationDataSetNbr	UInt16	Number of the application data record

1) If Bit 15 is set, RealTimeStart is active: the PFO positions the scanner mirror to the start position and waits for the digital input "RealTimeStart". Further information: (see "Tab. 314", pg. 186).

OVar type data structure Tab. 11

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 12

If a laser program for PFO 1 with the parameters *PFO: object control* = `Single`, *PFO-Ovar* = `1` and *TransX* = `OVar` is created,
SetPfoOvar must be called up with an Ovar data structure element:

```
SetPFOOVar(
 ListNbr = 1, StartNbr = 1, TransferMode = 0,
 OvarArray[0]:
  PfoProgNbr = 0, PfoProgramRuns= 0, Leistung = 0, TransX = <Korrekturwert1>,
  TransY = 0, ….
);
```

Tab. 13

If a laser program for PFO 1 with the parameters *PFO: object control* = `Block`, *PFO: Ovar Block Start* = `1`, *PFO: Ovar Block End* = 5 and *TransX* = `OVar` is created, SetPfoOvar must be called up with five Ovar data structure elements:

```
SetPFOOVar(
 ListNbr = 1, StartNbr = 1, TransferMode = 0,
 OvarArray[0]:
  PfoProgNbr = 0, PfoProgramRuns= 0, Leistung = 0, TransX = <Korrekturwert1>,
  TransY = 0, ….
 OvarArray[1]:
  PfoProgNbr = 0, PfoProgramRuns= 0, Leistung = 0, TransX = <Korrekturwert2>,
  TransY = 0, ….
 ...
 OvarArray[4]:
  PfoProgNbr = 0, PfoProgramRuns= 0, Leistung = 0, TransX = <Korrekturwert2>,
  TransY = 0, ….
);
```

Tab. 14

If a laser program for PFO 1 with the parameters *PFO: object control* = `Variable` and *TransX* = `OVar` is created, SetPfoOvar must be called up with two Ovar data structures elements.

```
SetPFOOVar(
 ListNbr = 1, StartNbr = 1, TransferMode = 0,
 OvarArray[0]:
  PfoProgNbr = 0, PfoProgramRuns= 0, Leistung = 0, TransX = <Korrekturwert1>,
  TransY = 0, ….
 OvarArray[1]:
  PfoProgNbr = 0, PfoProgramRuns= 0, Leistung = 0, TransX = <Korrekturwert2>,
  TransY = 0, ….
);
```

Tab. 15

Note

If two PFOs are operated on a light path with a beam splitter, only the *PFO: object control* = "Single" is used. To be able to

execute several OVar data records, several lines must be created in the laser program.

Writing the OVar structure

The OVar structure can be described by an OPC client as follows:

```
UaGenericStructureArray ovarDataArray{ g_pUaSession->structureDefinition(
UaNodeId(10706,2))};
UaGenericStructureValue ovarData{ g_pUaSession->structureDefinition(
UaNodeId(10706,2)) };
for (int i = 0; i < _countOvars; i++){
 ovarData.setField("PfoProgramNumber", OpcUa_UInt16(dist(mt)));
 ovarData.setField("PfoProgramRuns", OpcUa_UInt16(dist(mt)));
 ovarData.setField("PowerAbsolute", OpcUa_Double(dist(mt)));
 ovarData.setField("PowerOffsetAbsolute", OpcUa_Double(dist(mt)));
 ovarData.setField("TransX", OpcUa_Double(dist(mt)));
 ovarData.setField("TransY", OpcUa_Double(dist(mt)));
 ovarData.setField("TransZ", OpcUa_Double(dist(mt)));
 ovarData.setField("RotX", OpcUa_Double(dist(mt)));
 ovarData.setField("RotY", OpcUa_Double(dist(mt)));
 ovarData.setField("RotZ", OpcUa_Double(dist(mt)));
 ovarData.setField("ScaleX", OpcUa_Double(dist(mt)));
 ovarData.setField("ScaleY", OpcUa_Double(dist(mt)));
 ovarData.setField("ScaleZ", OpcUa_Double(dist(mt)));
 ovarData.setField("Defocus", OpcUa_Double(dist(mt)));
 ovarData.setField("WobbleProgramNumber", OpcUa_UInt16(dist(mt)));
 ovarData.setField("WobbleWidth", OpcUa_Double(dist(mt)));
 ovarData.setField("WobbleLength", OpcUa_Double(dist(mt)));
 ovarData.setField("WobbleFrequency", OpcUa_Double(dist(mt)));
 ovarData.setField("ApplicationDataSetNumber", OpcUa_UInt16(dist(mt)));
 ovarDataArray[i] = ovarData;
}
```

Tab. 16

getPFOOVar

Method for reading out object values from a PFO.

Parameter	Type	Description
ListNbr	UInt16	Number in the OVar list. The OVar list number corresponds to the PFO number. Value range 1 - 8
StartNbr	UInt16	First Ovar data record number, value range 1 - 500.
Count	UInt16	Number of Ovar data structure elements Value range 1 - 500, depending on the StartNbr parameter.
OvarVersion		NodeID of the desired data format, default: Namespace index: 2 Identifier type: Numerical Identifier 10703

Parameters (input) Tab. 17

	Type	Description
OvarDataStructure	OVar []	OVar type data structure.

Parameters (output) Tab. 18

Parameter	Type	Description
PfoProgNbr	UInt16	PFO program number[1)]
PfoProgramRuns	UInt16	Number of PFO program runs
Power	Double	Power [W]
LeistungOffset	Double	Power offset [W]
TransX	Double	PFO translation X [mm]
TransY	Double	PFO translation Y [mm]
TransZ	Double	PFO translation Z [mm]
RotX	Double	PFO rotation X [degrees]
RotY	Double	PFO rotation Y [degrees]
RotZ	Double	PFO rotation Z [degrees]
ScaleX	Double	PFO scaling X (100% or -100%). Only mirroring is permitted.
ScaleY	Double	PFO scaling Y (100% or -100%). Only mirroring is permitted.
ScaleZ	Double	PFO scaling Z (100% or -100%). Only mirroring is permitted.
Defocus	Double	PFO defocus [mm]
WobbleProgNbr	Double	Wobble program number
WobbleWidth	Double	Wobble width [mm]
WobbleLength	Double	Wobble length [mm]
WobbleFrequency	Double	Wobble frequency [Hz]
ApplicationDataSetNbr	UInt16	Number of the application data record

1) If Bit 15 is set, RealTimeStart is active: the PFO positions the scanner mirror to the start position and waits for the digital input "RealTimeStart". Further information: (see "Tab. 314", pg. 186).

OVar type data structure Tab. 19

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 20

getPFOOvarCounter

Method for reading out the number of successful OVar write operations.

Parameter	Type	Description
ListNbr	UInt16	Number in the OVar list. The OVar list number corresponds to the PFO number. Value range 1 - 8

Parameters (input) Tab. 21

	Type	Description
Counter	UInt16	Number of successful write operations. The counter jumps to zero after 65535 write operations and starts from the beginning again.

Parameters (output) Tab. 22

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 23

setIPFOOVar

Method for transferring object values to an IPFO.

Parameter	Type	Description
ListNo	UInt16	Number in the I-PFO OVar list, corresponds to the I-PFO number. Value range: 1 - 8.
JobNo	UInt16	Number of the I-PFO job.
Mode	UInt16	Data transfer mode Value must be = 0.
IpfOVar	IpfoOvar []	Data structure of type IpfOvar. Value range 1 - 130 data structures

Parameters (input) Tab. 24

Parameter	Type	Description
CorrectionMode	UInt8	The correction mode determines whether the correction values (PosX, etc.) are transferred absolutely or relatively. Bit 0 = 0: relative = 1: absolute Bit 1 = 0: Jobline is executed = 1: Jobline is omitted Examples: CorrectionMode = 0: relative correction, Jobline is executed. CorrectionMode = 2: relative correction, Jobline is omitted. CorrectionMode = 3: absolute correction, Jobline is omitted.
CorrectionType	UInt8	The correction type defines the coordinate system which refers to the correction values (PosX, etc) currently always = 0. = 0: Workpiece coordinates = 1: Tool coordinates
PosX	Double	Position X [mm]
PosY	Double	Position Y [mm]
PosZ	Double	Position Z [mm]
OrientationX	Double	PFO orientation X [degrees]
OrientationY	Double	PFO orientation Y [degrees]
OrientationZ	Double	PFO orientation Z [degrees]

Data structure of type IpfOvar Tab. 25

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 26

Note

The I-PFO object control is **NOT**possible on a beam splitter because **NO** beam splitting is possible for I-PFO.

getIPFOOVar

Method for reading out object values from an I-PFO.

Parameter	Type	Description
ListNbr	UInt16	Number in the I-PFO Ovar list, corresponds to the I-PFO number. Value range 1 - 8
StartNbr	UInt16	First I-PFO OVar data record number Value range 1 - 130
Count	UInt16	Number of I-PFO OVar data records Value range 1 - 130, depending on the StartNbr parameter
OvarVersion		NodeID of the desired data format, default: Namespace index: 2 Identifier type: Numerical Identifier 10700

Parameters (input) Tab. 27

Parameter	Type	Description
JobNumber	UInt16	Number of the I-PFO
IpfoOvarDataS-tructure	IpfoOvar []	Data structure of type IpfOvar.

Parameters (output) Tab. 28

Parameter	Type	Description
CorrectionMode	UInt8	The correction mode determines whether the correction values (PosX, etc.) are transferred absolutely or relatively. Bit 0 = 0: relative = 1: absolute Bit 1 = 0: Jobline is executed = 1: Jobline is omitted Examples: CorrectionMode = 0: relative correction, Jobline is executed. CorrectionMode = 2: relative correction, Jobline is omitted. CorrectionMode = 3: absolute correction, Jobline is omitted.
CorrectionType	UInt8	The correction type defines the coordinate system which refers to the correction values (PosX, etc) currently always = 0. = 0: Workpiece coordinates = 1: Tool coordinates
PosX	Double	Position X [mm]
PosY	Double	Position Y [mm]
PosZ	Double	Position Z [mm]
OrientationX	Double	PFO orientation X [degrees]
OrientationY	Double	PFO orientation Y [degrees]
OrientationZ	Double	PFO orientation Z [degrees]

Data structure of type IpfOvar Tab. 29

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 30

getIPFOOvarCounter

Method for reading out the number of successful IpfoOvar write operations.

Parameter	Type	Description
ListNbr	UInt16	Number in the I-PFO OVar list, corresponds to the I-PFO number. Value range 1 - 8

Parameters (input) Tab. 31

	Type	Description
Counter	UInt16	Number of successful write operations. The counter jumps to zero after 65535 write operations and starts from the beginning again.

Parameters (output) Tab. 32

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 33

6.3 Beam Source

Prerequisite User level ≥ 60.

The "Beam Source" node contains parameters for the laser beam source and beam guidance. The node is divided into the following nodes:

- PumpUnit: Pump unit parameters.
- Resonator: Resonator parameters.
- SKO: Parameters of the scattered light monitoring optics (SKO)

UaNodeId	Type	Unit	Description
BeamGuidance.ControllerIntervention	Double	%	Control intervention, output power

Measured values, beam source node Tab. 34

Pump unit

Prerequisite User level ≥ 10.

The following parameters are available under the "Beam-Source.PumpUnit" node:

UaNodeId	Type	Unit	Description
Lightmixer.ScatteredLight	Double	V	Scattered light homogenizer, pump source
Lightmixer.Temperature	Double	°C	Temperature homogenizer, pump source
Pump Unit.Current	Double	A	Pump current

Measured values, pump unit node Tab. 35

Resonator

Prerequisite User level ≥ 10.

The following parameters are available under the "BeamSource.Resonator" node:

UaNodeId	Type	Unit	Description
Cooling.PressureInlet	Double	bar	Cooling water supply pressure, laser head
Cooling.TempInlet	Double	°C	Temperature of cooling water supply, laser head
LaserPump.TemperatureCavity	Double	°C	Temperature of the pump laser cavity
Resonator.ControllerIntervention	Double	%	Control intervention/stabilizer regulation, laser oscillator
Resonator.DewPoint	Double	°C	Resonator dew point
Resonator.Fluorescence	Double		Fluorescence of the resonator
Resonator.HVModuleVoltage	Double	kV	Voltage, high-voltage supply module
Resonator.HVSwitchTemp	Double	°C	Temperature of high-voltage supply module
Resonator.InternalPower	Double	kW	Internal power of laser oscillator
Resonator.Power	UInt32	W	Average power of laser oscillator
Resonator.PulseReadyPoint	Double	%	Maximum amplifier time percentage
Thermalization.T_ Thermalization_Ready	Boolean		Thermalization is ready

Measured values, resonator node Tab. 36

SKO

Prerequisite User level ≥ 10.

The following parameters are available under the "BeamSource.SKO" node:

UaNodeId	Type	Unit	Description
BeamModule.			
BeamguidanceMirror1	Int16	mV	Scattered light monitoring, beam guidance module mirror 1
BeamguidanceMirror2	Int16	mV	Scattered light monitoring, beam guidance module mirror 2
BeamguidanceMirror3	Int16	mV	Scattered light monitoring, beam guidance module mirror 3
BeamguidanceMirror4	Int16	mV	Scattered light monitoring, beam guidance module mirror 4
BeamGuidance.			

UaNodeId	Type	Unit	Description
BeamguidanceMirror1	UInt16	mV	Scattered light monitoring, internal beam guidance mirror 1
BeamguidanceMirror3	UInt16	mV	Scattered light monitoring, internal beam guidance mirror 3
BeamguidanceMirror5	UInt16	mV	Scattered light monitoring, internal beam guidance mirror 5
BeamguidanceMirror8	UInt16	mV	Scattered light monitoring, internal beam guidance mirror 8
BeamguidanceScraper	UInt16	mV	Scattered light monitoring, internal beam guidance scraper

Measured values, SKO node Tab. 37

6.4 Production

The variables for the active light path and laser program are to be found under "Production" node.

Requirement

User level ≥ 0.

UaNodeId	Type	Description
LightPathNo	UInt16	Number of the currently selected light path.
ProgramNo	UInt16	Number of the active laser program.
ProgramLineNo	UInt16	Number of the active line in the active laser program.

Tab. 38

Multi-PLC

Variables that can be used to query and diagnose the configuration and the state of all MPI interfaces are to be found under "Multi-PLC" node.

That applies not only to the slots that were configured directly for OPC, but also to the parallel I/O interface and the available fieldbuses, if used.

Unlike the "Activation.Multi-PLC." node, all variables are read-only variables here. The "Production.Multi-PLC." node is a pure visualization and diagnostics interface.

UaNodeId	Type	Description
Production. Multi-PLC.ConfiguredSlotCount	UInt16	Number of configured MPI slots
Production.Multi-PLC.ActiveSlot	Int32	Number of the MPI slot that is currently assigned.
Production. Multi-PLC.LaserNotAssigned	Double	Time in seconds during which the laser was not assigned to any slot.

Tab. 39

Other nodes Further nodes are available for each configured MPI slot under the "Multi-PLC" node. Up to 32 MPI slots can be configured.

UaNodeId	Description
Production.MPI-Slot01	Node for slot 1
Production.MPI-Slot02	Node for slot 2
:	:
Production.MPI-Slot32	Node for slot 32

Tab. 40

From these nodes, only those configured in the MPI configuration in TruControl 1000 are available.

The available variables are described in the following, for MPI slot 12 as an example, which was configured with 16 words.

In all the following variables, the UaNodeId corresponds to the designation of the variable plus the "Production." prefix.

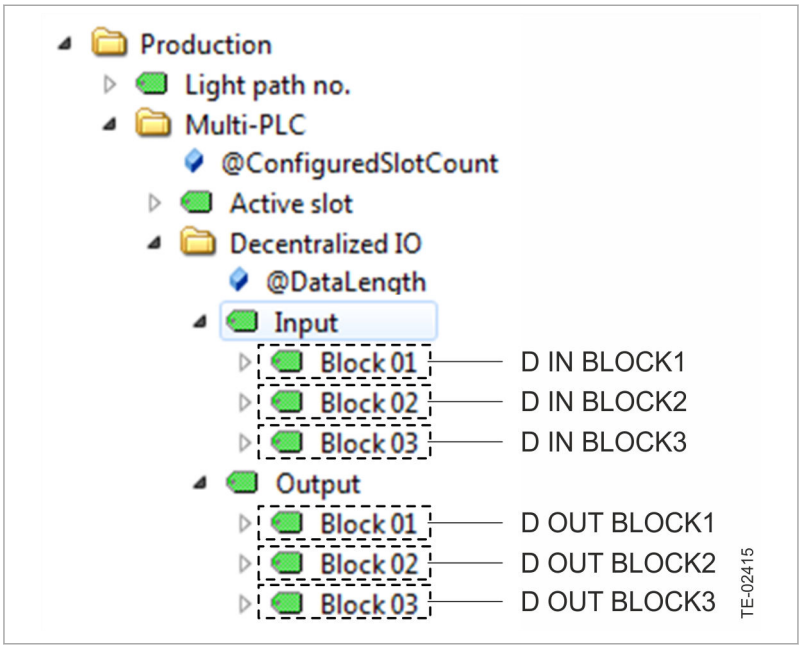
Example:
"MPI-Slot12.DataLength" is equivalent to UaNodeId "Production.MPI-Slot12.DataLength".

UaNodeId	Description
MPI-Slot12.ByteSwap	**Byte swap** (see "Setting interface parameters", pg. 263)
MPI-Slot12.DataLength	**Data length** (see "Setting interface parameters", pg. 263)
MPI-Slot12.InvertMask	**Inverting mask** (see "Setting interface parameters", pg. 263)
MPI-Slot12.OutputMask	**Output mask** (see "Setting interface parameters", pg. 263)
MPI-Slot12.Priority	**Priority** (see "Setting interface parameters", pg. 263)
MPI-Slot12.RealTimeInterface	**Real-time interface** (see "Setting interface parameters", pg. 263)
MPI-Slot12.Input	Array the the input words (see "Input words", pg. 145)
MPI-Slot12.Input.Word00	Input word 0 (see "Input words", pg. 145)
:	:
MPI-Slot12.Input.Word31	Input word 31 (see "Input words", pg. 145)
MPI-Slot12.Output	Array with the output words, (see "Output words", pg. 154)
MPI-Slot12.Output.Word00	Output word 0 (see "Output words", pg. 154)
:	:
MPI-Slot12.Output.Word31	Output word 31 (see "Output words", pg. 154)
MPI-Slot12. Counter.PowerGateBeamOFF	Duration in seconds how long the "beam off" PowerGate signal was active.
MPI-Slot12. Counter.PowerGateBeamON	Duration in seconds how long the "beam on" PowerGate signal was active.
MPI-Slot12.Counter.ProgActive	Duration in seconds how long the "program on" signal was active for this slot.
MPI-Slot12.Counter. SlotAssigned	Duration in seconds how long this slot had the assignment.
MPI-Slot12.Counter. Wait4Assignment	Duration in seconds how long this slot had to wait for the assignment.
MPI-Slot12.FieldbusInfo	Fieldbus information[1)]
MPI-Slot12.FieldbusInfo.ProfinetIO DeviceName	PROFINET IO Device Name[1)]
MPI-Slot12.FieldbusInfo.ProfinetIO DeviceName.MaxSizeByte	Maximum size of the parameter PROFINET IO Device Name[1)]
1) Only with PROFINET IO interfaces	

Variables of the Production.MPI-Slotx node, example x = 12, data length: 16 words Tab. 41

Decentralized I/O interface

If the digital I/O interface of the CPX (V3 or V4 standard version) is configured as a decentralized I/O interface, the 24 digital inputs and outputs are available to the user via OPC UA.



Digital inputs and outputs of the decentralized I/O interface in the OPC server of the CPX Fig. TE-02415

UaNodeId	Description
Production.DecentralizedIO	Main nodes of the decentralized I/O interface
Production.DecentralizedIO.DataLength	Attribute "DataLength"
Production.DecentralizedIO.Input	Array with all 24 inputs
Production.DecentralizedIO.Input.Block 0x	Array with 8 inputs of channel x, 1 ≤ x ≤ 3
Production.DecentralizedIO.Output	Array with all 24 outputs
Production.DecentralizedIO.Output.Block 0x	Array with 8 outputs of channel x, 1 ≤ x ≤ 3

Tab. 42

Additional information:
Interface description "Digital I/O interface of the CPX – Parallel I/O, real time, Sync I/O and decentralized I/O interface of the CPX V3 and V4 Standard", document no.: 18-22-91-A2-BA.

Pulse data

UaNodeId	Type	Description
PulseData.CwPower	Double	Laser power
PulseData.Energy	UInt64	Current energy of the currently active pulse
PulseData.Frequency	Double	Currently set pulse repetition frequency
PulseData.FrequencyNominal	Double	Currently measured pulse repetition frequency
PulseData.Power	Double	Currently measured laser power
PulseData.PulseDuration	UInt32	Currently set pulse duration
PulseData.PulseDurationNominal	UInt32	Currently measured pulse duration
PulseData.PulseNumber	UInt16	Currently set pulse number
PulseData.PulseNumberNominal	UInt16	Currently measured pulse number
PulseData.PulsePower	Double	Currently set pulse power
PulseData.PulsePowerNominal	Double	Currently measured pulse power
PulseData.ShortPulseEnergy	Double	Currently set short pulse energy (TruMicro)
PulseData.PulseMode	Byte	Currently set pulse mode: 0 CW mode (continuous power) 1 Pulsed operation with finite pulse number 2 Pulsed operation with infinite pulse number

Tab. 43

System status

The system status contains important status bits on the operating status of the laser device. In all of the following variables, the UaNodeId corresponds to the designation of the variable plus the "SystemState" prefix..
Example: Variable "PROG_BREAK" is equivalent to UaNodeId "SystemState.PROG_BREAK".

UaNodeId	Description
SystemState. AUXILIARYHEATING_ACTIVE	System status: heating phase is active.
SystemState.ESTOP_OPEN	System status: Emergency Stop is open.
SystemState. EXT_ACTIVATION_ACTIVE	(see "EXT_ACTIVATION_ACTIVE", pg. 199)
SystemState.FAULT_LASER	(see "FAULT_LASER", pg. 199)
SystemState. FIRMWAREUPDATE_ACTIVE	Firmware update in progress. **Note:** Do not switch off the laser device!
SystemState.FKO_OPEN	Optics functional circuit is open.
SystemState.LASER_IS_ON	(see "LASER_IS_ON", pg. 201)
SystemState.LASER_ON_PHASE	"Laser on" signal was activated. The laser is in what is called the LaserOn phase, which undergoes various switch-on functions. This state may persist for a few minutes.
SystemState.LASER_READY	(see "LASER_READY", pg. 202)
SystemState. LASER_WARNING_LAMP	(see "LASER_WARNING_LAMP_IS_ON", pg. 203)
SystemState.LIGHTPATH_BLOCKED	The actively selected light path is blocked.
SystemState.LLK_OPEN	The laser light cable circuit of the active light path is open.
SystemState.LLR_OFF	The laser power control is switched off.
SystemState. LOCK_OPERATING_PANEL	(see "LOCK_OPERATING_PANEL", pg. 171)
SystemState.MAINTENANCE	(see "MAINTENANCE", pg. 215)
SystemState.MONITORING	(see "MONITORING", pg. 203)
SystemState.PILOT_LASER_IS_ON	(see "PILOT_LASER_IS_ON", pg. 204)
SystemState.POWER_LIMIT	(see "POWER_LIMIT", pg. 204)
SystemState.PROG_ACTIVE	Laser program is active (global, not slot-related), (see "PROG_ACTIVE", pg. 204).
SystemState.PROG_BREAK	Laser program was aborted.
SystemState.PROG_COMPLETED	(see "PROG_COMPLETED", pg. 206)
SystemState.PSTART_STATICAL	(see "PSTART_STATICAL", pg. 175)
SystemState.SIK_OPEN	(see "SIK_OPEN", pg. 207)
SystemState.SLEEPMODE_ACTIVE	Optimized standby mode (sleep mode) active.
SystemState.SYNC_OPEN	Main shutter open.
SystemState.SYS_READY	System ready to communicate.
SystemState.SYS_RESETTED	System was restarted (reset).

Tab. 44

6.5 Maintenance

Depending on the laser type and hardware (control), information on the following maintenance tasks ("Maintenance") can be provided on the OPC UA server.

- General Maintenance
- Cleaning the laser circuit
- Carbonfilter Nitrogen
- Microfilter Laserhead
- CPX Battery
- Air Cleaner
- Active carbon filter
- Pump module
- Converter
- Micro Filter Set
- Main shutter error check
- Humidity separator (recirculating air dryer system humidity desiccant)
- Conversion module desiccant replacement

General Maintenance

Variables that can be used to read information about general maintenance and acknowledge the maintenance are to be found under "General Maintenance" node. The maintenance is shown for all lasers.

The maintenance is carried out regularly

- every 5000 hours
- For TruDiode every 12 months

Requirement

User level ≥ 30

UaNodeId	Type	Description
Maintenance.General.DateLastMain	String	Date of the last maintenance. Format YYYY-MM-DD.
Maintenance.General.Limits	FolderType	Subdirectory with limit values.
Maintenance.General.Limits. UpWarnLimit	Double	Upper warning limit for maintenance.
Maintenance.General.Limits@Unit	String	Unit of the warning limit.
Maintenance.General. OpHoursLastMain	String	Number of operating hours during the last acknowledgement[1)].
Maintenance.General. OpHoursLastMain@Unit	String	Unit of the operating hours values[1)].
Maintenance.General.CurrentValue	UInt32	Current number of operating hours since last acknowledgement. For laser devices of TruDiode type: current number of days since last acknowledgement.
Maintenance.General. CurrentValue@Unit	String	Unit of the operating hours values[1)].
Maintenance.General.State	Int32	Maintenance status: -1 An error occurred. However, the exact origin is unknown. 0 The maintenance type could not be identified. 1 No maintenance work required. 2 Warning! Maintenance work required! Work can continue. 3 Fault! Maintenance work required! It may not be possible to continue work. 4 Maintenance was postponed.
Maintenance.General.doAck	Method	Acknowledgement of maintenance
Maintenance.General.doAck. UserLevel	UInt16	User level for acknowledgement
1) Not available for laser devices of TruDiode type.		

Tab. 45

Cleaning the laser circuit

Variables that can be used to read information about laser cooling circuit cleaning maintenance and acknowledge the maintenance are to be found under the "Cleaning the laser circuit" node.

Prerequisites

- User level ≥ 30
- System parameter "WDB_WaterChange" = 1

UaNodeId	Type	Description
Maintenance.Water.DateLastMain	String	Date of the last maintenance.
Maintenance.Water.CurrentValue	Double	Current number of days since the last acknowledgement.
Maintenance.Water.Limits	FolderType	Subdirectory with limit values (maintenance interval).
Maintenance.Water.Limits. UpWarnLimit	Double	Upper warning limit for maintenance.
Maintenance.Water.Limits@Unit	String	Unit of the warning limit.
Maintenance.Water.State	Int32	Maintenance status: -1 An error occurred. However, the exact origin is unknown. 0 The maintenance type could not be identified. 1 No maintenance work required. 2 Warning! Maintenance work required! Work can continue. 3 Fault! Maintenance work required! It may not be possible to continue work. 4 Maintenance was postponed.
Maintenance.Water.doAck	Method	Acknowledgement of maintenance
Maintenance.Water.doAck.UserLevel	UInt16	User level for acknowledgement

Tab. 46

Carbonfilter Nitrogen

Variables that can be used to read information about activated carbon filter maintenance and acknowledge the maintenance are to be found under the "Carbon filter nitrogen" node.

Prerequisites

- User level ≥ 30
- System parameter "WDB_ActivatedCarbonFilter_Nitrogen" = 1.

UaNodeId	Type	Description
Maintenance.NgUnitFilter. DateLastMain	String	Date of the last maintenance. Format YYYY-MM-DD
Maintenance.NgUnitFilter. CurrentValue	String	Date of the next maintenance. Format YYYY-MM-DD
Maintenance.NgUnitFilter.Limits	FolderType	Subdirectory with limit values.
Setting.WDB_ActivatedCarbonFilter_ Max.Age	UInt32	Upper warning limit for maintenance.
Maintenance.NgUnitFilter.State	Int32	Maintenance status: -1 An error occurred. However, the exact origin is unknown. 0 The maintenance type could not be identified. 1 No maintenance work required. 2 Warning! Maintenance work required! Work can continue. 3 Fault! Maintenance work required! It may not be possible to continue work. 4 Maintenance was postponed.
Maintenance.NgUnitFilter.doAck	Method	Acknowledgement of maintenance
Maintenance.NgUnitFilter.doAck. UserLevel	UInt16	User level for acknowledgement

Tab. 47

Microfilter Laserhead

Variables that can be used to read information about maintenance of the fine filter in the laser head and acknowledge the maintenance are to be found under the "Microfilter Nitrogen" node.

Prerequisites

- User level ≥ 30
- Laser device type: TruMicro
- System parameter "WDB_MicroFilter_Laserhead" = 1

UaNodeId	Type	Description
Maintenance. ILHcFilter.DateLastMain	String	Date of the last maintenance. Format YYYY-MM-DD
Maintenance.ILHcFilter.CurrentValue	String	Date of the next maintenance. Format YYYY-MM-DD
Maintenance.ILHcFilter.Limits	FolderType	Subdirectory with limit values.
[A3/01].Rd_ILHC_FilterLengthOfStay	Byte	Upper warning limit for maintenance of the ILHC board. User level ≥ 10
Maintenance.ILHcFilter.State	Int32	Maintenance status: -1 An error occurred. However, the exact origin is unknown. 0 The maintenance type could not be identified. 1 No maintenance work required. 2 Warning! Maintenance work required! Work can continue. 3 Fault! Maintenance work required! It may not be possible to continue work. 4 Maintenance was postponed.
Maintenance.ILHcFilter.doAck	Method	Acknowledgement of the maintenance.
Maintenance. ILHcFilter.doAck.UserLevel	UInt16	User level for acknowledgement.

Tab. 48

CPX Battery

Variables that can be used to read information about battery status are to be found under the "CPX Battery" node.

Prerequisites

- User level ≥ 30
- CPX V1

UaNodeId	Type	Description
Maintenance.CPXBat.Limits	FolderType	Subdirectory with limit values.
Maintenance. CPXBat.Limits.LowFaultLimit	Double	Lower warning limit for maintenance.
[F8/01]. Rd_CPEXT_VoltageCpxBattery	Double	Current state of the battery. User level ≥ 10
Maintenance.CPXBat.State	Int32	Maintenance status: -1 An error occurred. However, the exact origin is unknown. 0 The maintenance type could not be identified. 1 No maintenance work required. 2 Warning! Maintenance work required! Work can continue. 3 Fault! Maintenance work required! It may not be possible to continue work. 4 Maintenance was postponed.

Tab. 49

Air Cleaner

Variables that can be used to read information about air cleaner maintenance can be found under the "Air Cleaner" node.

Requirement

User level ≥ 30

UaNodeId	Type	Description
Maintenance.AirCleaner.Limits	FolderType	Subdirectory with limit values.
Maintenance.AirCleaner.Limits. UpFaultLimit	Double	Upper error limit for the maintenance.
Maintenance.AirCleaner.Limits. UpWarnLimit	Double	Upper warning limit for maintenance.
Maintenance.AirCleaner.Limits@Unit	String	Unit of error and warning limits.
Maintenance.AirCleaner.CurrentValue	UInt32	Current number of operating hours since the last acknowledgement.
Maintenance.AirCleaner.CurrentValue @Unit	String	Unit for current number of operating hours.
Maintenance.AirCleaner.State	Int32	Maintenance status: -1 An error occurred. However, the exact origin is unknown. 0 The maintenance type could not be identified. 1 No maintenance work required. 2 Warning! Maintenance work required! Work can continue. 3 Fault! Maintenance work required! It may not be possible to continue work. 4 Maintenance was postponed.

Tab. 50

Active carbon filter

Variables that can be used to read information about activated carbon filter ACC maintenance and acknowledge it are to be found under "Active Carbon Filter" node.

Requirement

User level ≥ 30

UaNodeId	Type	Description
Maintenance.ActiveCarbonFilter. DateLastMain	String	Date of the last maintenance. Format: YYYY-MM-DD
Maintenance.ActiveCarbonFilter. Limits	FolderType	Subdirectory with limit values.
Maintenance.ActiveCarbonFilter. Limits.UpFaultLimit	Double	Upper error limit for the maintenance.
Maintenance.ActiveCarbonFilter. Limits.UpWarnLimit	Double	Upper warning limit for maintenance.
Maintenance.ActiveCarbonFilter. Limits@Unit	String	Unit of error and warning limits.
Maintenance.ActiveCarbonFilter. OpHoursLastMain	String	Number of operating hours during the last acknowledgement.
Maintenance.ActiveCarbonFilter. OpHoursLastMain@Unit	String	Unit of the operating hours values.
Maintenance.ActiveCarbonFilter. CurrentValue	UInt32	Current number of operating hours since the last acknowledgement.
Maintenance.ActiveCarbonFilter. CurrentValue@Unit	String	Unit of the operating hours values.
Maintenance.ActiveCarbonFilter. State	Int32	Maintenance status: -5 There is no replacement filter. -4 No data from the CAC/ACC component. -3 Wrong CAC/ACC filter number. -2 Error when writing to the CAC/ACC component. -1 An error occurred. However, the exact origin is unknown. 1 No maintenance work required. 2 Warning! Maintenance work required! Work can continue. 3 Fault! Maintenance work required! It may not be possible to continue work. 4 Maintenance was postponed.
Maintenance.ActiveCarbonFilter. doAck	Method	Acknowledgement of maintenance
Maintenance.ActiveCarbonFilter. doAck.UserLevel	UInt16	User level for acknowledgement
Maintenance.ActiveCarbonFilter. doAck.Input	UInt16	Input parameters for the acknowledgment: ▪ SerialNumber ACCFilter (UInt32) ▪ SerialNumber ChangeACCFilter (UInt32)

Tab. 51

Pump module

There are variables that can be used to read information about pump module maintenance and acknowledge the maintenance under "Pump Module" node.

Prerequisites

- User level ≥ 30
- PMI (Pump Module Interface) or DAP (DATA ACQUISITION PUMP MODULE) available

UaNodeId	Type	Description
Maintenance.PumpModule. DateLastMain	String	Date of the last maintenance. Format: YYYY-MM-DD
Maintenance.PumpModule. CurrentValue	UInt32	Number of hours that the laser ran in emergency mode (i.e. after a failed PMI or DAP, System.STACK_BYPASS = TRUE)
Maintenance.PumpModule.AllStates	FolderType	Subdirectory with all pump modules.
Maintenance.PumpModule. PMI_AA_BB.State or Maintenance.PumpModule. DAP_AA_BB.State	Int32	Status of the specific pump module (DAP or PMI): 0 No error (status OK) 1 Error (pump module bridged) AA Number of the pump unit (cav.) B Number of the pump module
Maintenance.PumpModule.State	Int32	Maintenance status: -1 An error occurred. However, the exact origin is unknown. 0 The maintenance type could not be identified. 1 No maintenance work required. 2 Warning! Maintenance work required! Work can continue. 3 Fault! Maintenance work required! It may not be possible to continue work. 4 Maintenance was postponed.
Maintenance.PumpModule.doAck	Method	Acknowledgement of maintenance
Maintenance.PumpModule. doAck.UserLevel	UInt16	User level for acknowledgement

Tab. 52

Converter

Variables that can be used to read information about the conversion module and acknowledge the maintenance are to be found under "Converter" node.

Available for laser devices of TruMicro type.

Prerequisites

- User level ≥ 30
- Laser device type TruMicro

UaNodeId	Type	Description
Maintenance.Converter.DateLastMain	String	Date of the last maintenance. Format: YYYY-MM-DD
Maintenance.Converter.State	Int32	Maintenance status: 1 No maintenance work required. 2 Warning! Maintenance work required! A conversion module was replaced. 3 Warning! No more memory is available. Please delete a module from the list.
Maintenance.Converter.doAck	Method	Acknowledgement of the maintenance.
Maintenance.Converter. doAck.UserLevel	UInt16	User level for acknowledgement.
Maintenance.Converter.CurrentValue	UInt32	Type of maintenance: 0 Current module is OK 1 Test module 2 Calibration module 3 New module 4 Replacement module 5 Remontage
Maintenance.Converter. ConverterModeHistory	FolderType	Subdirectory with modification log.
Maintenance.Converter. ConverterModeHistory.Position_X	FolderType	Subdirectories with all information for module X 0 latest entry : 7 oldest entry
Maintenance.Converter. ConverterMode History.Position_X.InstallDay	Byte	Day of installation (module X).
Maintenance.Converter.Converter ModeHistory.Position_X.InstallMonth	Byte	Month of installation (module X).
Maintenance.Converter.Converter ModeHistory.Position_X.InstallYear	UInt32	Year of installation (module X).
Maintenance.Converter.Converter ModeHistory.Position_X.SAPNr	String	SAP number (module X).
Maintenance.Converter.Converter ModeHistory.Position_X.SerialNo	String	Serial number (module X).
Maintenance.Converter.Converter ModeHistory.Position_X.WaveLength	UInt32	Wavelength (module X).
Maintenance.Converter.ConverterTable	FolderType	Subdirectory with all installed modules.

UaNodeId	Type	Description
Maintenance.Converter.ConverterTable.Position_X	FolderType	Subdirectories with all information for module X 0 latest entry : 7 oldest entry
Maintenance.Converter.ConverterTable.Position_X.InstallDay	Byte	Day of installation (module X).
Maintenance.Converter.ConverterTable.Position_X.InstallMonth	Byte	Month of installation (module X).
Maintenance.Converter.ConverterTable.Position_X.InstallYear	UInt32	Year of installation (module X).
Maintenance.Converter.ConverterTable.Position_X.SAPNr	String	SAP number (module X).
Maintenance.Converter.ConverterTable.Position_X.SerialNr	String	Serial number (module X).
Maintenance.Converter.ConverterTable.Position_X.WaveLength	UInt32	Wavelength (module X).
Maintenance.Converter.ConverterTable.ActiveModuleIndex	Int32	Index of the active module (0 … 7).
Maintenance.Converter.CurrentConverterMode	FolderType	Subdirectory with all the information of the active module.
[A3/01].Rd_ILHC_ConvModActivateDateDay	Byte	Day of installation of the active module
[A3/01].Rd_ILHC_ConvModActivateDateMonth	Byte	Month of installation of the active module
[A3/01].Rd_ILHC_ConvModActivateDateYear	UInt16	Year of installation of the active module
[A3/01].Rd_ILHC_ConvModIdentNumber	String	Identity number of the active module
[A3/01].Rd_ILHC_ConvModVariationIndex	String	Variation index of the active module
Maintenance.Converter.CurrentConverterMode.SAPNr	String	SAP number of the active module.
Maintenance.Converter.CurrentConverterMode.SerialNr	String	Serial number of the active module.
Maintenance.Converter.CurrentConverterMode.WaveLength	UInt32	Wavelength of the active module.

Tab. 53

Micro filter set

Variables that can be used to read information about micro filter maintenance and acknowledge the maintenance are to be found under the "Micro Filter Set" node.

Prerequisites

- User level ≥ 30
- System parameter "WDB_MicroFilterSet" = 1.

UaNodeId	Type	Description
Maintenance.MicroFilterSet. DateLastMain	String	Date of the last maintenance. Format: YYYY-MM-DD
Maintenance.MicroFilterSet. CurrentValue	String	Date of the next maintenance. Format: YYYY-MM-DD
Maintenance.MicroFilterSet.Limits	FolderType	Subdirectory with limit values.
Setting.WDB_MicroFilterSet_Max.Age	UInt32	Upper warning limit for maintenance.
Maintenance.MicroFilterSet.State	Int32	Maintenance status: -1 An error occurred. However, the exact origin is unknown. 0 The maintenance type could not be identified. 1 No maintenance work required. 2 Warning! Maintenance work required! Work can continue. 3 Fault! Maintenance work required! It may not be possible to continue work. 4 Maintenance was postponed.
Maintenance.MicroFilterSet.doAck	Method	Acknowledgement of the maintenance.
Maintenance.MicroFilterSet.doAck. UserLevel	UInt16	User level for acknowledgement.

Tab. 54

Main shutter error check

Variables that can be used to read information about maintenance of the central shutter and acknowledge the maintenance are to be found under the "Main Shutter Error Check" node.

The ASV board for the central shutter must be replaced regularly because the switch contacts of the relays used have a limited service life. Not applicable to ASV5 board.

Prerequisites

- User level ≥ 30
- System parameter "WDB_MainShutter_Error_Count" = 1.

UaNodeId	Type	Description
Maintenance.MainShutterAsv ErrorCheck.DateLastMain	String	Date of the last maintenance.
Maintenance. MainShutterAsvErrorCheck.Limits	FolderType	Subdirectory with limit values.
Setting.WDB_MainShutter_ Error_Count_Level1	UInt32	Upper fault limit (threshold value 1) for maintenance. Keep new ASV board ready.
Setting.WDB_MainShutter_ Error_Count_Level2	UInt32	Upper fault limit (threshold value 2) for maintenance. Replace ASV board.
[F8/01].Rd_CPEXT_Maint_ MainShutterAsvErrorCounter		Current status of the maintenance counter for the main shutter.
Maintenance. MainShutterAsvErrorCheck.State	Int32	Maintenance status: -1 An error occurred. However, the exact origin is unknown. 0 The maintenance type could not be identified. 1 No maintenance work required. 2 Warning! Maintenance work required! Work can continue. 3 Fault! Maintenance work required! It may not be possible to continue work.
Maintenance.MainShutterAsv ErrorCheck.doAck	Method	Acknowledgement of the maintenance.
Maintenance.MainShutterAsv ErrorCheck.doAck.UserLevel	UInt16	User level for acknowledgement.

Tab. 55

Humidity separator (recirculating air dryer system humidity desiccant)

Variables that can be used to find information about maintenance and the state of the humidity separator and acknowledge the maintenance are located under the "RDSHumidityDesiccant" node.

Prerequisites

- User level ≥ 30
- System parameter "WDB_RDS" = 1.

UaNodeId	Type	Description
Maintenance.RDSHumidity DesiccantX.DateLastMain	String	Date of the last maintenance for humidity separator X, X = 1...8 Format: YYYY-MM-DD
Maintenance.RDSHumidity DesiccantX.Limits	FolderType	Subdirectory with limit values for humidity separator X, X = 1...8
Maintenance.RDSHumidity DesiccantX.Limits.UpWarnLimit	UInt32	Upper monitoring limit for maintenance for humidity separator X,X = 1...8.
Maintenance.RDSHumidity DesiccantX.State	Int32	Maintenance status for humidity separator X, X = 1...8 -1 An error occurred. However, the exact origin is unknown. 0 The maintenance type could not be identified. 1 No maintenance work required. 2 Warning! Maintenance work required! Work can continue.
Maintenance.RDSHumidity DesiccantX.doAck	Method	Acknowledgement method for humidity separator X, X = 1..8
Maintenance.RDSHumidity DesiccantX.doAck.UserLevel	UInt16	User level for the acknowledgment method.

Tab. 56

Conversion module desiccant replacement

Variables that can be used to find information about maintenance of the desiccant replacement conversion module and acknowledge the maintenance are located under the "Desiccant Converter" node.

Prerequisites

- User level ≥ 30
- Laser device family 20-10-08-xx
- System parameter "WDB_DesiccantConverter" > 0.

UaNodeId	Type	Description
Maintenance.DesiccantConverter. DateLastMain	String	Date of the last maintenance. Format YYYY-MM-DD
Maintenance.DesiccantConverter. CurrentValue	String	Date of the next maintenance. Format YYYY-MM-DD
Maintenance.DesiccantConverter. Limits	FolderType	Subdirectory with limit values.
Maintenance.DesiccantConverter. Limits.UpWarnLimit	UInt32	Upper warning limit for maintenance
Maintenance.DesiccantConverter. Limits@Unit	String	Unit of the warning limit
Maintenance.DesiccantConverter. State	Int32	Maintenance status: -1 An error occurred. However, the exact origin is unknown. 0 The maintenance type could not be identified. 1 No maintenance work required. 2 Warning! Maintenance work required! Work can continue.
Maintenance.DesiccantConverter. doAck	Method	Acknowledgement of the maintenance.
Maintenance.DesiccantConverter. doAck.UserLevel	UInt16	User level for the acknowledgment method.

Tab. 57

6.6 Diagnostics

Under "Diagnostics" node variables and methods are provided for diagnostics.

UaNodeId	Type	Description
LastMessage	UInt32	Currently active message code (32-bit) including ID/NUM. If there is no message, the value is "0".
LastMessage.CompID	Byte	Component ID from the currently active message
LastMessage.CompNUM	Byte	Component NUM from the currently active message
LastMessage.MsgCode	UInt16	Code for the ID/NUM of the currently active message
LastMessage.ShortText	String	Short text for the currently active message. The language used is defined by the client when establishing a connection.
LastMessage.Type	Int32	Type of currently active message (fault, monitoring, …).
LastMessage.Timestamp	DateTime	Time of occurrence of the currently active message
doReset	Method	Acknowledgement of the currently active messages.

Tab. 58

Message history

Under "Message history" node methods are provided for retrieving the message memory, individual messages or also only texts for specified message codes.

UaNodeId	Description
MsgHistory.getAvailableLanguages	Returns a list of the available languages as ISO codes.
MsgHistory.getActiveMessageList	Returns a list of the currently active messages that have accumulated since the last time the device was switched on or since the last reset.
MsgHistory.getActiveMessageList SequenceNbrs	Returns a list with message IDs of the currently active messages that have accumulated since the last time the device was switched on or since the last reset.
MsgHistory.getMessageList	Returns a list of the requested messages from the message memory.
MsgHistory.getMessageList SequenceNbrs	Returns a list with message IDs of the requested messages from the message memory.
MsgHistory.getMessage	Returns all the information on a message including the complete message text in the requested format.
MsgHistory.getMessage2	Returns all the information on a message including the complete message text in the requested format in the desired language.
MsgHistory.getMessageShortText	Returns the short text for the requested message code.
MsgHistory.getMessageShortText2	Returns the short text for the requested message code in the desired language.
MsgHistory.getMessageText	Returns all the information on a message including the complete message text in the requested format.

Tab. 59

MsgHistory.getAvailableLanguages

Returns a list of the available languages as ISO codes.

Parameter	Type	Description
Language	String[LangCount]	Array with ISO codes of the available languages. Example: {'da','de','de-DE','en','es','fr','it','ja','pt','pt-BR','sk'}.

Parameters (output) Tab. 60

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 61

MsgHistory.getActiveMessageList

Returns a list of the currently active messages that have accumulated since the last time the device was switched on or since the last reset.

Parameter	Type	Description
Filters	UInt32	Filter for the type of the messages requested.

Parameters (input) Tab. 62

Parameter	Type	Description
Max.SequenceNo	Int64	Largest currently used internal message ID.
UsedFilter	UInt32	Filter specified when calling the method.
MsgCount	UInt16	Number of returned messages. This value corresponds to the array size of the following parameters!
SeqNo	Int64[MsgCount]	Array with the message IDs of the individual messages.
TimeStamp	DateTime[MsgCount]	Array with the time stamps of the individual messages.
MsgType	UInt16[MsgCount]	Array with the types of the individual messages.
MsgCode	UInt32[MsgCount]	Array with the message codes of the individual messages.
LPath	UInt16[MsgCount]	Array with the numbers of the light paths that were active when the corresponding message occurred.
SlotNo	UInt16[MsgCount]	Array with the numbers of the MPI slots that were selected when the corresponding message occurred.
LineNo	UInt16[MsgCount]	Array with the line numbers of the corresponding active laser program that was being executed at the moment when the message occurred.
ProgNo	UInt16[MsgCount]	Array with the numbers of the corresponding programs that were being executed at the moment when the message occurred.
State	UInt16[MsgCount]	Array with the status information of the messages. "0" if the message has not yet been acknowledged.
ShortText	LocalizedText[MsgCount]	Array with short texts for the corresponding messages.

Parameters (output) Tab. 63

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 64

MsgHistory.getActiveMessageListSequenceNbrs

Returns a list with message IDs of the currently active messages that have accumulated since the last time the device was switched on or since the last reset.

Parameter	Type	Description
Filters	UInt32	Filter for the type of the messages requested.

Parameters (input) Tab. 65

Parameter	Type	Description
Max.SequenceNo	Int64	Largest currently used internal message ID.
UsedFilter	UInt32	Filter specified when calling the method.
MsgCount	UInt16	Number of returned messages. This value corresponds to the array size of the following parameters!
SeqNo	Int64[MsgCount]	Array with the message IDs of the individual messages.

Parameters (output) Tab. 66

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 67

MsgHistory.getMessageList

Returns a list of the requested messages from the message memory.

Parameter	Type	Description
StartSeqNo	Int64	Message ID starting from which messages are to be returned. If the parameter is = "0", then the process is started at the beginning or at the end depending on the operational sign of the "MsgCount" parameter
MsgCount	Int16	Number of messages “x” which have to be returned. x>0: “x” messages which have occurred with the message ID “StartSeqNo” and afterwards are issued. x<0: “x” Messages are issued which have occurred with the message ID “StartSeqNo” and before.
Filters	UInt32	Filter for the type of the messages requested.

Parameters (input) Tab. 68

Parameter	Type	Description
Max.SequenceNo	Int64	Largest currently used internal message ID.
UsedFilter	UInt32	Filter specified when calling the method.
MsgCount	UInt16	Number of returned messages. This value corresponds to the array size of the following parameters!
SeqNo	Int64[MsgCount]	Array with the message IDs of the individual messages.
TimeStamp	DateTime[MsgCount]	Array with the time stamps of the individual messages.
MsgType	UInt16[MsgCount]	Array with the types of the individual messages.
MsgCode	UInt32[MsgCount]	Array with the message codes of the individual messages.
LPath	UInt16[MsgCount]	Array with the numbers of the light paths that were active when the corresponding message occurred.
SlotNo	UInt16[MsgCount]	Array with the numbers of the MPI slots that were selected when the corresponding message occurred.
LineNo	UInt16[MsgCount]	Array with the line numbers of the corresponding active laser program that was being executed at the moment when the message occurred.
ProgNo	UInt16[MsgCount]	Array with the numbers of the corresponding programs that were being executed at the moment when the message occurred.
State	UInt16[MsgCount]	Array with the status information of the messages. "0" if the message was not acknowledged yet.
ShortText	LocalizedText[MsgCount]	Array with short texts of the corresponding messages.

Parameters (output) Tab. 69

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 70

MsgHistory.getMessageListSequenceNbrs

Returns a list with message IDs of the requested messages from the message memory.

Parameter	Type	Description
StartSeqNo	Int64	Message ID starting from which messages are to be returned. If the parameter is = "0", then the process is started at the beginning or at the end depending on the operational sign of the "MsgCount" parameter.
MsgCount	Int16	Number of messages “x” which have to be returned. x>0: “x” messages which have occurred with the message ID “StartSeqNo” and afterwards are issued. x<0: “x” Messages are issued which have occurred with the message ID “StartSeqNo” and before.
Filters	UInt32	Filter for the type of the messages requested.

Parameters (input) Tab. 71

Parameter	Type	Description
Max.SequenceNo	Int64	Largest currently used internal message ID.
UsedFilter	UInt32	Filter specified when calling the method.
MsgCount	UInt16	Number of returned messages. This value corresponds to the array size of the following parameters!
SeqNo	Int64[MsgCount]	Array with the message IDs of the individual messages.

Parameters (output) Tab. 72

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 73

MsgHistory.getMessage

Returns all the information on a message including the complete message text in the requested format.

Parameter	Type	Description
SeqNo	Int64	Message ID of the message which is to be returned.
OutputFormat	Int16	Format in which the message text is to be returned: 0 Text 1 HTML 2 XML (Format is not documented)

Parameters (input) Tab. 74

Parameter	Type	Description
SeqNo	Int64	Message ID of the requested message.
TimeStamp	DateTime	Time stamp of the requested message.
MsgType	UInt16	Type of requested message.
MsgCode	UInt32	Message code of the requested message.
LPath	UInt16	Number of the light path that was active when the message occurred.
SlotNo	UInt16	Number of the MPI slot that was selected when the message occurred.
LineNo	Uint16	Line number of the laser program which was being executed at the moment when the message occurred.
ProgNo	UInt16	Number of the laser program that was being executed at the moment when the message occurred.
State	UInt16	Status of the message. "0" if the message has not yet been acknowledged.
ShortText	LocalizedText	Short text for the requested message.
OutputFormat	Int16	Message text format of the requested message: 0 Text 1 HTML 2 XML (Format is not documented)
Text	LocalizedText	Complete message text of the requested message in the desired format.

Parameters (output) Tab. 75

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 76

MsgHistory.getMessage2

Returns all the information on a message including the complete message text in the requested format in the desired language.

Parameter	Type	Description
SeqNo	Int64	Message ID of the message to be returned.
OutputFormat	Int16	Format in which the message text is to be returned: 0 Text 1 HTML 2 XML (Format is not documented)
Locale	String	(Local) language in which the message text is to be returned. The "Locale" parameter is specified in ISO format, for example "de-DE", "en", "pt-BR"…

Parameters (input) Tab. 77

Parameter	Type	Description
SeqNo	Int64	Message ID of the requested message.
TimeStamp	DateTime	Time stamp of the requested message.
MsgType	UInt16	Type of requested message.
MsgCode	UInt32	Message code of the requested message.
LPath	UInt16	Number of the light path that was active when the message occurred.
SlotNo	UInt16	Number of the MPI slot that was selected when the message occurred.
LineNo	UInt16	Line number of the laser program that was being executed at the moment when the message occurred.
ProgNo	UInt16	Number of the laser program that was being executed at the moment when the message occurred.
State	UInt16	Status of the message. "0" if the message has not yet been acknowledged.
ShortText	LocalizedText	Short text for the requested message.
OutputFormat	Int16	see OutputFormat, Parameter (Input).
Text	LocalizedText	Complete message text of the requested message in the desired format.

Parameters (output) Tab. 78

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 79

MsgHistory.getMessageShortText

Returns the short text of the requested message code.

Parameter	Type	Description
FullCode	UInt32	Message code (including ID/NUM) for which the short text is to be returned.

Parameters (input) Tab. 80

Parameter	Type	Description
ShortText	LocalizedText	Short text of the requested message code.

Parameters (output) Tab. 81

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 82

MsgHistory.getMessageShortText2

Returns the short text of the requested message code in the desired language.

Parameter	Type	Description
FullCode	UInt32	Message code (including ID/NUM) for which the short text is to be returned.
Locale	String	(Local) language in which the message text is to be returned. The "Locale" parameter is specified in ISO format, for example "de-DE", "en", "pt-BR"…

Parameters (input) Tab. 83

Parameter	Type	Description
ShortText	LocalizedText	Short text of the requested message code.

Parameters (output) Tab. 84

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 85

MsgHistory.getMessageText

Returns all the information on a message including the complete message text in the requested format.

Parameter	Type	Description
FullCode	UInt32	Message code (including ID/NUM) for which the message text is requested.
OutputFormat	Int16	Format in which the message text is to be returned: 0 Text 1 HTML 2 XML (Format is not documented)

Parameters (input) Tab. 86

Parameter	Type	Description
Text	LocalizedText	Complete message text of the requested message in the desired format.
OutputFormat	Int16	see OutputFormat, Parameter (Input).

Parameters (output) Tab. 87

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 88

Cooling

Prerequisite User level ≥ 10

Under "Cooling" node there are parameters for laser device cooling.

UaNodeId	Type	Unit	Description
Cooling.CabinetTemp	Double	°C	Temperature in the electrical cabinet
Cooling.DIReturnPressure	Double	bar	Current pressure of the deionized water
Cooling.ExtPressureDiff	Double	bar	External pressure difference
Cooling.Fan1Speed	Double	rpm	Fan 1 speed
Cooling.Fan2Speed	Double	rpm	Fan 2 speed
Cooling.InFlowExtPressure	Double	bar	Pressure supply of external cooling water circuit
Cooling.InFlowExtTemp[1)]	Double	°C	Temperature of the external cooling water circuit supply
Cooling.PumpFrequency	Double	Hz	Cooling water pump frequency
Cooling.PumpPressure	Double	bar	Cooling water pump pressure
Cooling.ReturnflowExtPressure	Double	bar	Pressure return flow, external cooling water circuit
Cooling.TankLevel	Double	%	Cooling water tank level
Cooling.TankTemp	Double	°C	Cooling water tank temperature
Cooling.Conductivity	Double	µS/cm	Cooling water conductivity
Cooling.Valve.OpeningAngle	Double	%	Proportional valve opening angle
1) For laser device family 20-09-10-A2000 and 20-09-10-A2001: **ExtCircuit.InflowExtTemp**.			

Tab. 89

Pulse data/Counter

Prerequisite User level ≥ 0

Under "Pulse data/Counter" node there is information about the usage times of the laser.

UaNodeId	Type	Description
PDCounter.ControlOnCount	UInt32	Counter how often the control was switched on.
PDCounter.ControlOnHours	Double	Time in hours how long the control was switched on up to now.
PDCounter.LaserOnCount	UInt32	Counter how often the laser beam source was switched on.
PDCounter.LaserOnHours	Double	Time in hours how long the laser beam source was switched on up to now.
PDCounter.ProgDuration	Double	Duration of the last executed laser program in milliseconds.
PDCounter.ProgExec	DateTime	Time of the last program start.
PDCounter.ProgNo	UInt16	Number of the last executed laser program.
PDCounter.ProgStartCount	UInt32	Counter how often a laser program was started.
PDCOunter.PulseCounter	UInt32	Number of laser pulses up to now.

Tab. 90

CalibrationLine

Methods for calling the history of the Focus Calibration Program (FCP) and Power Calibration Program (PCP) are provided under "CalibrationLine" node.

UaNodeId	Description
CalibrationLine.getFcpHistory	Returns the FCP history.
CalibrationLine.getPcpHistory	Returns the PCP history.

Tab. 91

CalibrationLine.getFcpHistory

Returns the requested FCP history.

Parame-ter	Type	Description
PfoNum	Byte	Number of the PFO for which the history is to be returned.
StartIndex	Int64	Index of the first line to be returned. If the parameter is = "0", then the process is started at the beginning or at the end depending on the operational sign of the "RowCount" parameter.
RowCount	Int16	Number of lines „x“ that are to be returned. x>0: „x“ lines that have occurred with the line "StartIndex" and afterwards are issued. x<0: „x“ lines that have occurred with the line "StartIndex" and before are issued.

Parameters (input) Tab. 92

Parameter	Type	Description
PfoNum	Byte	Number of the PFO for which the history is returned.
ItemCount	UInt16	Number of lines to be returned. This value corresponds to the array size of the following parameters.
Index	Int64[ItemCount]	Array with indices of the original table entries of the history.
PFOserialNo	String[ItemCount]	Array with the serial numbers of the PFO.
CCLserialNo	String[ItemCount]	Array with the serial numbers of the CalibrationLine interface.
ProgNo	UInt16[ItemCount]	Array with the numbers of the laser programs.
LPath	UInt16[ItemCount]	Array with the numbers of the light paths.
TimeStamp	DateTime[ItemCount]	Array with the time stamps.
Mode	Int16[ItemCount]	Array with the modes: ▪ = 0001h: XY correction ▪ = 0002h: Z correction ▪ = 0003h: XYZ correction values = 0 ▪ = 8001h: XY measurement ▪ = 8002h: Z measurement
PfoTransX	Int64[ItemCount]	Array with the values of the PFO translation in X direction [µm].
PfoTransY	Int64[ItemCount]	Array with the values of the PFO translation in Y direction [µm].
PfoTransZ	Int64[ItemCount]	Array with the values of the PFO translation in Z direction [µm].
Power	Int64[ItemCount]	Array with the laser power values [W].
PulseDuration	Int64[ItemCount]	Array with the pulse duration values [ms].
GridSize	Int64[ItemCount]	Array with the grid size values [µm].

Parameter	Type	Description
GridPoints	Int64[ItemCount]	Array with the number of grid points.
PfoCorrLimit	Int64[ItemCount]	Array with the values of the PFO correction limits [µm].
ResultCCL	Int64[ItemCount]	Array with the result codes of the CalibrationLine interface. ▪ = 0: no error ▪ ≠ 0: see TruControl
MsgCodeCCL	String[ItemCount]	Array with the message codes of the CalibrationLine interface. ▪ = 0: no error ▪ ≠ 0: see TruControl
DeviationCclX	Int64[ItemCount]	Array with the X deviations of the CalibrationLine interface [µm].
DeviationCclY	Int64[ItemCount]	Array with the Y deviations of the CalibrationLine interface [µm].
DeviationCclZ	Int64[ItemCount]	Array with the Z deviations of the CalibrationLine interface [µm].
ResultPFO	Int64[ItemCount]	Array with the result codes of the PFO. ▪ = 0: no error ▪ ≠ 0: see TruControl
MsgCodePFO	String[ItemCount]	Array with the message codes of the PFO. ▪ = 0: no error ▪ ≠ 0: see TruControl
DeviationPfoX	Int64[ItemCount]	Array with the X deviations of the PFO [µm].
DeviationPfoY	Int64[ItemCount]	Array with the Y deviations of the PFO [µm].
DeviationPfoZ	Int64[ItemCount]	Array with the Z deviations of the PFO [µm].
FocusDiametersLowerLimit	Uint32	Lower limit value for focal diameter.
FocusDiameterUpperLimit	Uint32	Upper limit value for focal diameter.
FocusCoefficientA	Double	Coefficient A focal diameter.
FocusCoefficientB	Double	Coefficient B focal diameter.
FocusCoefficientC	Double	Coefficient C focal diameter.
FocusPositionLowerLimit	Int32	Lower focus position limit value [mm].
FocusPositionUpperLimit	Int32	Upper focus position limit value [mm].

Parameters (output) Tab. 93

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 94

CalibrationLine.getPcpHistory

Returns the requested PCP history.

Parameter	Type	Description
PfoNum	Byte	Number of the PFO for which the history is to be returned.
StartIndex	Int64	Index of the first line to be returned. If the parameter is = "0", then the process is started at the beginning or at the end depending on the operational sign of the "RowCount" parameter.
RowCount	Int16	Number of lines „x“ that are to be returned. x>0: „x“ lines that have occurred with the line "StartIndex" and afterwards are issued. x<0: „x“ lines that have occurred with the line "StartIndex" and before are issued.

Parameters (input) Tab. 95

Parameter	Type	Description
PfoNum	Byte	Number of the PFO for which the history is returned.
ItemCount	UInt16	Number of lines to be returned. This value corresponds to the array size of the following parameters.
Index	Int64[ItemCount]	Array with indices of the original table entries of the history.
PPCserialNo	String[ItemCount]	Array with the serial numbers of the Power Probe Cassette (PPC).
CLIserialNo	String[ItemCount]	Array with the serial numbers of the CalibrationLine interface (CLI).
ProgNo	UInt16[ItemCount]	Array with the numbers of the laser programs.
LPath	UInt16[ItemCount]	Array with the numbers of the light paths.
TimeStamp	DateTime[ItemCount]	Array with the time stamps.
ResultState	Int16[ItemCount]	Array with the result states from the history table. ■ = 0: no error ■ ≠ 0: see TruControl
Mode	Int16[ItemCount]	Array with the modes. ■ = 0: power correction ■ = 1: power measurement
PcpPower	Int16[ItemCount]	Array with the power values of the PCP measurement [W].
PcpDuration	Int16[ItemCount]	Array with the data on the duration of the PCP measurement [ms].
PcpMonitoring Limit	Int16[ItemCount]	Array with the monitoring limits of the PCP measurement [0.1%].
PcpFaultLimit	Int16[ItemCount]	Array with the fault limits of the PCP measurement [0.1%].
ResEnergy	Int16[ItemCount]	Array with the energy values measured [J]. ■ = FFFFh: invalid
ResPower	Int16[ItemCount]	Array with the power values measured [W]. ■ = FFFFh: invalid
ResPulseDuration	Int16[ItemCount]	Array with the pulse durations measured [ms]. ■ = FFFFh: invalid
ResSinglePulse Duration	Int16[ItemCount]	Array with the individual pulse durations measured [ms]. ■ = FFFFh: invalid
ResPulseNumber	Int16[ItemCount]	Array with the pulse number measured.
CalcCorrectionFactor	Int16[ItemCount]	Array with the calculated correction factors [0.1%].
CalcDeviation	Int32[ItemCount]	Array with the calculated deviations. = 7FFFh: invalid
CalcPcpPower Reserve	Int16[ItemCount]	Array with the calculated PCP power reserves [W].

Parameters (output) Tab. 96

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

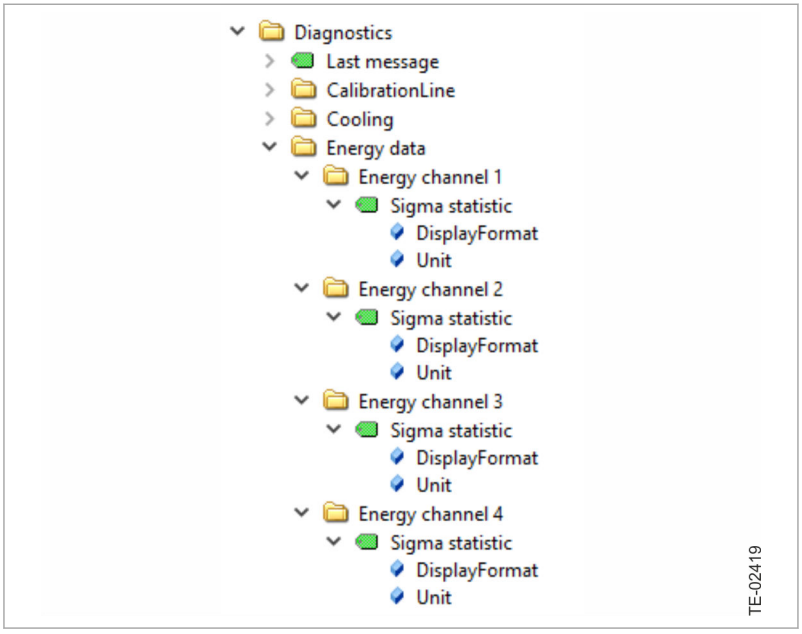
Return value Tab. 97

Energy Data

Energy values are available under the "Energy Data" node.

Requirements

- TruControl 1000 version ≥ 2.80.0.
- User level ≥ 10.
- Laser device family 20-09-10 or 20-09-13.



EnergyData Fig. TE-02419

UaNodeId	Type	Description
Diagnostics.EnergyData	FolderType	Subdirectory node with energy measurement data.
EnergyData.EnergyChannelX[1)]	FolderType	Subdirectory node for measuring channel X[1)].
EnergyChannelX[1)].SigmaStatistic	Double	Standard deviation of the parameter in the measuring channel X[1)].
EnergyChannelX[1)].SigmaStatistic. DisplayFormat	String	Display format for the standard parameter deviation in measuring channel X[1)].
EnergyChannelX[1)].SigmaStatistic.Unit	String	Display for the standard parameter deviation in measuring channel X[1)].
Energy.Data.SigmaStatistic	Double	Standard deviation pulse energy [%], laser device family 20-09-10-A2000 and 20-09-10-A2001
1) X = 1, 2, 3 or 4		

Tab. 98

Measuring channel	Standard parameter deviation	Measuring channel	Standard parameter deviation
1	IR pulse energy resonator.	3	UV pulse energy output 1.
2	IR pulse energy downstream of the external modulator.	4	UV pulse energy output 2.

Example of measuring channel assignment for laser device family 20-09-10 Tab. 99

SmartBeo

Requirements

- TruControl 1000 version ≥ 3.42.0.
- User level ≥ 0.

Various measured values are provided under the "Diagnostics.SmartBeo.<counter>.<variable name>" node for all MIO and PGM components of the BEO Smart focusing optics present in the system.

<Counter>: Number of the component MIO or PGM of the focusing optics BEO Smart, example MIO with NUM = 4: counter = 3.

<Variable name>: see table:

Variable name	Type	Description
ScatterLight	UInt16	Scattered radiation [mV]
CoolingWater	Double	Cooling water temperature return flow [°C]
PressureCrossjet	Double	Gas pressure at Crossjet [bar]
FlowProtectiveGas	Double	Flow in shielding gas [l/min]
FlowMDEGas	Double	Flow of MDE gas [l/min]
FlowFlushGas	Double	Flow of purging gas [l/min]
GlassContamination	UInt16	Protective glass contamination
LaserTimePGO	UInt32	Operating hours of objective protective glass [h]
LaserTimePGP	UInt32	Operating hours of process protective glass [h]

Tab. 100

6.7 Programming

Laser program

The format of the laser programs that can be read out and written via this interface is currently not documented and additionally protected against modifications by means of a checksum.

Requirement

User level ≥ 40.

UaNodeId	Description
LaserProg.getDirectory	Returns a list of all laser programs on the device.
LaserProg.readProgram	Returns the contents of the laser program with the specified program number.
LaserProg.writeProgram	Writes the contents of a laser program to the location with the specified program number.
LaserProg.deleteProgram	Deletes the laser program with the specified program number from the control.
LaserProg.lockProgram	Locks a laser program against modifications by other programs (TruControl 1000 and other programs)
LaserProg.releaseProgram	Unlocks a laser program for modifications by other programs.

Tab. 101

LaserProg.getDirectory

Returns a list of all laser programs on the device.

Parameter	Type	Description
ProgCount	UInt16	Number of laser programs on the device.
ProgNo	Int16[ProgCount]	Array with the numbers of the individual laser programs.
ProgName	String[ProgCount]	Array with the names of the individual laser programs.
ChangeDate	DateTime[ProgCount]	Array with the time stamps of the last modification of the individual laser programs.

Parameters (output) Tab. 102

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 103

LaserProg.readProgram

Returns the contents of the laser program with the specified program number.

Parameter	Type	Description
ProgNo	Int16	Number of the laser program to be returned.

Parameters (input) Tab. 104

Parameter	Type	Description
ProgName	String	Name of the requested laser program.
ChangeDate	DateTime	Time stamp of the last modification of the requested laser program.
Program	String	Program code of the requested laser program in a defined XML format.

Parameters (output) Tab. 105

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 106

LaserProg.writeProgram

Writes the contents of a laser program to the location with the specified program number.

Parameter	Type	Description
ProgNo	Int16	Number of the laser program to be written/replaced.
Program	String	Program code of the laser program in a defined XML format that is to be transferred to the control.

Parameters (input) Tab. 107

Parameter	Type	Description
ProgName	String	Name of the laser program[1)].
ChangeDate	DateTime	Time stamp of the last modification of the laser program[1)].
1) As stored in the XML data.		

Parameters (output) Tab. 108

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 109

LaserProg.lockProgram

Disables the laser program with the specified program number against modifications by other programs (including TruControl 1000).

Parameter	Type	Description
ProgNo	Int16	Number of the laser program to be disabled.
AddInfo1	String	Additional information on disabling. This information may also be displayed in TruControl 1000 when a disabled laser program is to be accessed and is also returned when a different client wants to disable the same laser program.
AddInfo2	String	Additional information on disabling. See also AddInfo1.

Parameters (input) Tab. 110

Parameter	Type	Description
LockAssigned	Boolean	Indicates whether the lock could be set. If TRUE, then the following output parameters contain information on their own client. If FALSE, then the following output parameters contain information on a different client which has set the lock first.
ClientInfo	String	Information on the client who disabled the specified laser program.
AddInfo1	String	Additional information on disabling specified by the corresponding client when disabling.
AddInfo2	String	Additional information on disabling. See also AddInfo1.
AlreadyLocked	Boolean	Indicates whether the laser program has already been locked before. In combination with the **LockAssigned** output parameter, it is possible to determine whether the lock had already been set by its own client or a different client.

Parameters (output) Tab. 111

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 112

LaserProg.releaseProgram

Unlocks a laser program for modifications by other programs. One restriction is that a client can unlock only those locks which it has set itself.

Parameter	Type	Description
ProgNo	Int16	Number of the laser program to be enabled.

Parameters (input) Tab. 113

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 114

LaserProg.deleteProgram

Deletes the laser program with the specified program number from the control.

Parameter	Type	Description
ProgNo	Int16	Number of the laser program to be deleted from the control.

Parameters (input) Tab. 115

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 116

Parameter set

The format of the parameter sets that can be read out and written via this interface is currently not documented and additionally protected against modifications by means of a checksum.

Requirement

User level ≥ 40.

UaNodeId	Description
ParameterSet.getDirectory	Returns a list of all parameter sets on the device.
ParameterSet.readParameterSet	Returns the contents of the parameter set with the specified number.
ParameterSet.writeParameterSet	Writes the contents of a parameter set to the location with the specified number.
ParameterSet.deleteParameterSet	Deletes the parameter set with the specified number from the control.

Tab. 117

ParameterSet.getDirectory

Returns a list of all parameter sets on the device.

Parameter	Type	Description
ParamSetCount	UInt16	Number of parameter sets on the device.
ParamSetNo	Int16[ParamSetCount]	Array with the numbers of the individual parameter sets.
ParamSetName	String[ParamSetCount]	Array with the names of the individual parameter sets.
ChangeDate	DateTime[ParamSetCount]	Array with the time stamps of the last modification of the individual parameter sets.

Parameters (output) Tab. 118

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 119

ParameterSet.readParameterSet

Returns the contents of the parameter set with the specified number.

Parameter	Type	Description
ParamSetNo	Int16	Number of the requested parameter set.

Parameters (input) Tab. 120

Parameter	Type	Description
ParamSetName	String	Name of the requested parameter set.
ChangeDate	DateTime	Time stamp of the last modification of the requested parameter set.
ParameterSet	String	Program code of the requested parameter set in the defined XML format.

Parameters (output) Tab. 121

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 122

ParameterSet.writeParameterSet

Writes the contents of a parameter set to the location with the specified number.

Parameter	Type	Description
ParamSetNo	Int16	Number of the parameter set to be written/replaced.
ParameterSet	String	Program code of the parameter set in a defined XML format that is to be transferred to the control.

Parameters (input) Tab. 123

Parameter	Type	Description
ParamSetName	String	Name of parameter set[1)].
ChangeDate	DateTime	Time stamp of the last modification of the parameter set[1)].
1) As stored in the XML data.		

Parameters (output) Tab. 124

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 125

ParameterSet.deleteParameterSet

Deletes the parameter set with the specified number from the control.

Parameter	Type	Description
ParamSetNo	Int16	Number of the parameter set to be deleted from the control.

Parameters (input) Tab. 126

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 127

Pulse shape

The format of the pulse shapes that can be read out and written via this interface is currently not documented and additionally protected against modifications by means of a checksum.

Requirement

User level ≥ 40.

UaNodeId	Description
PulseShape.getDirectory	Returns a list of all pulse shapes on the device.
PulseShape.readPulseShape	Returns the contents of the pulse shape with the specified pulse shape number.
PulseShape.writePulseShape	Writes the contents of a pulse shape to the location with the specified pulse shape number.
PulseShape.deletePulseShape	Deletes the pulse shape with the specified pulse shape number from the control.

Tab. 128

PulseShape.getDirectory

Returns a list of all pulse shapes on the device.

Parameter	Type	Description
PulseShapeCount	UInt16	Number of pulse shapes on the device.
PulseShapeNo	Int16[PulseShapeCount]	Array with the pulse shape numbers of the individual pulse shapes.
PulseShapeType	Byte[PulseShapeCount]	Array with the types of the individual pulse shapes.
PulseShapeName	String[PulseShapeCount]	Array with the names of the individual pulse shapes.
ChangeDate	DateTime[PulseShapeCount]	Array with the time stamps of the last modification of the individual pulse shapes.

Parameters (output) Tab. 129

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 130

PulseShape.readPulseShape

Returns the contents of the pulse shape with the specified pulse shape number.

Parameter	Type	Description
PulseShapeNo	Int16	Pulse shape number of the requested pulse shape.

Parameters (input) Tab. 131

Parameter	Type	Description
PulseShapeName	String	Name of the requested pulse shape.
ChangeDate	DateTime	Time stamp of the last modification of the requested pulse shape.
PulseShape	String	Program code of the requested pulse shape in a defined XML format.

Parameters (output) Tab. 132

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 133

PulseShape.writePulseShape

Writes the contents of a pulse shape to the location with the specified pulse shape number.

Parameter	Type	Description
PulseShapeNo	Int16	Pulse shape number of the pulse shape to be written/replaced.
PulseShape	String	Program code of the pulse shape in a defined XML format that is to be transferred to the control.

Parameters (input) Tab. 134

Parameter	Type	Description
PulseShapeType	Byte	Type of the pulse shape[1)].
PulseShapeName	String	Name of the pulse shape[1)].
ChangeDate	DateTime	Time stamp of the laser modification of the pulse shape[1)].
1) As stored in the XML data.		

Parameters (output) Tab. 135

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 136

PulseShape.lockPulseShape

Disables the pulse shape with the specified pulse shape number against modifications by other programs (including TruControl 1000).

Parameter	Type	Description
ProgNo	Int16	Number of the pulse shape to be disabled.
AddInfo1	String	Additional information on the disabling. This information may also be displayed in TruControl 1000 when a disabled pulse shape is to be accessed and is also returned when a different client wants to disable the same pulse shape.
AddInfo2	String	Additional information on disabling. See also AddInfo1.

Parameters (input) Tab. 137

Parameter	Type	Description
LockAssigned	Boolean	Indicates whether the lock could be set. If TRUE, then the following output parameters contain information on their own client. If FALSE, then the following output parameters contain information on a different client which has set the lock first.
ClientInfo	String	Provides information on the client that disabled the specified program.
AddInfo1	String	Additional information on disabling specified by the corresponding client when disabling.
AddInfo2	String	Additional information on disabling. See also AddInfo1.
AlreadyLocked	Boolean	Indicates whether the pulse shape has already been locked before. In combination with the **LockAssigned** output parameter, it is possible to determine whether the lock had already been set by its own client or a different client.

Parameters (output) Tab. 138

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 139

PulseShape.releasePulseShape

Unlocks a pulse shape for modifications by other programs. One restriction is that a client can unlock only those locks which it has set itself.

Parameter	Type	Description
PulseShapeNo	Int16	Number of the pulse shape to be enabled.

Parameters (input) Tab. 140

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 141

PulseShape.deletePulseShape

Deletes the pulse shape with the specified pulse shape number from the control.

Parameter	Type	Description
PulseShapeNo	Int16	Pulse shape number of the pulse shape to be deleted from the control.

Parameters (input) Tab. 142

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 143

PFO program

The format of the PFO programs that can be read out and written via this interface is documented and can be made available on request.

Requirement

User level ≥ 40.

UaNodeId	Description
PfoProgram.getDirectory	Returns a list of all PFO programs on the device.
PfoProgram.getDirectoryEx	Returns an extended list of all PFO programs on the device with additional information on the individual programs.
PfoProgram.readProgram	Returns the contents of the PFO program with the specified program number.
PfoProgram.writeProgram	Writes the contents of a PFO program to the location with the specified program number.
PfoProgram.writeProgVarData	Writes data to the specified variable. The workpiece is labeled with this data or a machine-readable code is applied to the workpiece.
PfoProgram.lockProgram	Disables the PFO program with the specified program and PFO number against modifications by other programs (including TruControl 1000).
PfoProgram.releaseProgram	Unlocks a PFO program for modifications by other programs.[1)].
PfoProgram.deleteProgram	Deletes the PFO program with the specified program number from the control.
PfoProgram.lockPFO	Disables all PFO programs for the specified PFO against modifications by other programs (including TruControl 1000).
PfoProgram.releasePFO	Unlocks all PFO programs of the specified PFO for modifications by other programs.[1)].
1) One restriction is that a client can unlock only those locks which it has set itself.	

Tab. 144

PfoProgram.getDirectory

Returns a list of all PFO programs on the device.

Parameter	Type	Description
PfoNumber	Byte	Number of the PFO for which the directory is to be returned.

Parameters (input) Tab. 145

Parameter	Type	Description
ProgCount	UInt16	Number of PFO programs on the device.
ProgNo	Int16[ProgCount]	Array with the numbers of the individual PFO programs.
ProgType	String[ProgCount]	Array with the types of the individual PFO programs.
ProgName	String[ProgCount]	Array with the names of the individual PFO programs.
RefCount	UInt16	Array with the reference counters of the individual PFO programs.
ChangeDate	DateTime[ProgCount]	Array with the time stamps of the last modification of the individual PFO programs.

Parameters (output) Tab. 146

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 147

PfoProgram.getDirectoryEx

Returns an extended list of all PFO programs on the device.

Parameter	Type	Description
PfoNumber	Byte	Number of the PFO for which the directory is to be returned.

Parameters (input) Tab. 148

Parameter	Type	Description
ProgCount	UInt16	Number of PFO programs on the device.
ProgNo	Int16[ProgCount]	Array with the numbers of the individual PFO programs.
ProgType	String[ProgCount]	Array with the types of the individual PFO programs.
ProgName	String[ProgCount]	Array with the names of the individual PFO programs.
RefCount	UInt16[ProgCount]	Array with the reference counters of the individual PFO programs.
ChangeDate	DateTime[ProgCount]	Array with the time stamps of the last modification of the individual PFO programs.
NoOfPoints	UInt32[ProgCount]	Array with the number of points of the individual PFO programs.
TotalTime	UInt32[ProgCount]	Array with the complete runtimes of the individual PFO programs.
HotTime	UInt32[ProgCount]	Array with the critical runtimes of the individual PFO programs.
NoOfGeos	UInt32[ProgCount]	Array with the number of all geometries of the individual PFO programs.
NoOfHotGeos	UInt32[ProgCount]	Array with the number of critical geometries of the individual PFO programs.
StartX	Double[ProgCount]	Array with the X coordinate of the starting points of the individual PFO programs.
StartY	Double[ProgCount]	Array with the Y coordinate of the starting points of the individual PFO programs.
StartZ	Double[ProgCount]	Array with the Z coordinate of the starting points of the individual PFO programs.
EndX	Double[ProgCount]	Array with the X coordinate of the end points of the individual PFO programs.
EndY	Double[ProgCount]	Array with the Y coordinate of the end points of the individual PFO programs.
EndZ	Double[ProgCount]	Array with the Z coordinate of the end points of the individual PFO programs.

Parameters (output) Tab. 149

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 150

PfoProgram.readProgram

Returns the contents of the PFO program with the specified program number.

Parameter	Type	Description
PfoNumber	Byte	Number of the PFO for which the PFO program is to be returned.
ProgNo	Int16	Number of the requested PFO program

Parameters (input) Tab. 151

Parameter	Type	Description
ProgType	String	Type of the requested PFO program.
ProgName	String	Name of the requested PFO program.
ChangeDate	DateTime	Time stamp of the last modification of the requested PFO program.
NoOfPoints	UInt32	Number of points of the requested PFO program.
TotalTime	UInt32	Complete runtime of the requested PFO program.
HotTime	UInt32	Critical runtime of the requested PFO program.
NoOfGeos	UInt32	Number of all geometries of the requested PFO program.
NoOfHotGeos	UInt32	Number of critical geometries of the requested PFO program.
StartX	Double	X coordinate of the starting point of the requested PFO program.
StartY	Double	Y coordinate of the starting point of the requested PFO program.
StartZ	Double	Z coordinate of the starting point of the requested PFO program.
EndX	Double	X coordinate of the end point of the requested PFO program.
EndY	Double	Y coordinate of the end point of the requested PFO program.
EndZ	Double	Z coordinate of the end point of the requested PFO program.
Program	String	Program code of the requested PFO program in the defined XML format.

Parameters (output) Tab. 152

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 153

PfoProgram.writeProgram

Writes the contents of a PFO program to the location with the specified program number.

Parameter	Type	Description
PfoNumber	Byte	Number of the PFO into which the PFO program is to be transferred.
ProgNo	Int16	Number of the PFO program to be written/replaced.
Program	String	Program code of the PFO program in a defined XML format that is to be transferred to the control.

Parameters (input) Tab. 154

Parameter	Type	Description
PfoNumber	Byte	Number of the PFO into which the PFO program was transferred.
ProgNo	Int16	Number of the PFO program that was written/replaced.
ProgName	String	Name of the PFO program[1)].
ChangeDate	DateTime	Time stamp of the last change[1)].
NoOfPoints	UInt32	Number of points[1)].
TotalTime	UInt32	Complete runtime[1)].
HotTime	UInt32	Critical runtime[1)].
NoOfGeos	UInt32	Number of all geometries[1)].
NoOfHotGeos	UInt32	Number of critical geometries[1)].
StartX	Double	X coordinate of the starting point[1)].
StartY	Double	Y coordinate of the starting point[1)].
StartZ	Double	Z coordinate of the starting point[1)].
EndX	Double	X coordinate of the end point[1)].
EndY	Double	Y coordinate of the end point[1)].
EndZ	Double	Z coordinate of the end point[1)].
1) As stored in the XML data.		

Parameters (output) Tab. 155

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 156

PfoProgram.writeProgVarData

Writes text in the specified variable.

Parameter	Type	Description
PfoNumber	Byte	Number of the selected PFO.
ProgNo	Int16	Number of the selected PFO program.
VariableNo	Byte	Number of the selected variables.
VariableText	String	This parameter contains the following machine-readable code: ▪ QR code. ▪ Data Matrix Code. ▪ Text with which the workpiece is labeled.

Parameters (input) Tab. 157

Type	Description
UaStatus	0 OK. 1 Number of the variables was not found. 2 Calculation of the variable PFO subprogram has failed.

Return value Tab. 158

PfoProgram.lockProgram

Disables the PFO program with the specified program and PFO number against modifications by other programs (including TruControl 1000).

Parameter	Type	Description
PfoNumber	Byte	Number of the PFO for which a PFO program is to be disabled.
ProgNo	Int16	Number of the PFO program to be disabled.
AddInfo1	String	Additional information on disabling. This information may also be displayed in TruControl 1000 when a disabled PFO program is to be accessed and is also returned when a different client wants to disable the same PFO program.
AddInfo2	String	Additional information on disabling. Description: See also AddInfo1.

Parameters (input) Tab. 159

Parameter	Type	Description
LockAssigned	Boolean	Indicates whether the lock could be set. If TRUE, then the following output parameters contain information on their own client. If FALSE, then the following output parameters contain information on a different client which has set the lock first.
ClientInfo	String	Information on the client who disabled the specified PFO program.
AddInfo1	String	Additional information on disabling specified by the corresponding client when disabling.
AddInfo2	String	Additional information on disabling. See also AddInfo1.
AlreadyLocked	Boolean	Indicates whether the PFO program has already been locked before. In combination with the **LockAssigned** output parameter, it is possible to determine whether the lock had already been set by its own client or a different client.

Parameters (output) Tab. 160

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 161

PfoProgram.releaseProgram

Unlocks a PFO program for modifications by other programs. One restriction is that a client can unlock only those locks which it has set itself.

Parameter	Type	Description
PfoNumber	Byte	Number of the PFO for which a PFO program is to be enabled.
ProgNo	Int16	Number of the PFO program to be enabled.

Parameters (input) Tab. 162

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 163

PfoProgram.deleteProgram

Deletes the PFO program with the specified program number from the control.

Parameter	Type	Description
PfoNumber	Byte	Number of the PFO from which the PFO program is to be deleted.
ProgNo	Int16	Job number of the PFO program to be deleted from the control.

Parameters (input) Tab. 164

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 165

PfoProgram.lockPFO

Disables all PFO programs for the specified PFO against modifications by other programs (including TruControl 1000).

Parameter	Type	Description
PfoNumber	Byte	Number of the PFO to be disabled.
AddInfo1	String	Additional information on disabling. This information may also be displayed in TruControl 1000 when a disabled PFO is to be accessed and is also returned when a different client wants to disable the same program.
AddInfo2	String	Additional information on disabling. See also AddInfo1.

Parameters (input) Tab. 166

Parameter	Type	Description
LockAssigned	Boolean	Indicates whether the lock could be set. If TRUE, then the following output parameters contain information on their own client. If FALSE, then the following output parameters contain information on a different client which has set the lock first.
ClientInfo	String	Provides information on the client who disabled the specified PFO.
AddInfo1	String	Additional information on disabling specified by the corresponding client when disabling.
AddInfo2	String	Additional information on disabling. See also AddInfo1.
AlreadyLocked	Boolean	Indicates whether the PFO has already been locked before. In combination with the LockAssigned output parameter, it is possible to determine whether the lock had already been set by its own client or a different client.

Parameters (output) Tab. 167

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 168

PfoProgram.releasePFO

Enables all PFO programs of the specified PFO for modifications by other programs. One restriction is that a client can unlock only those locks which it has set itself.

Parameter	Type	Description
PfoNumber	Byte	Number of the PFO whose PFO programs are to be enabled.

Parameters (input) Tab. 169

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 170

I-PFO job

The format of the I-PFO job that can be read out and written via this interface is currently not documented and additionally protected against modifications by means of a checksum.

Requirement

User level ≥ 40.

UaNodeId	Description
I-PfoJob.getDirectory	Returns a list of all I-PFO job on the device.
I-PfoJob.readJob	Returns the contents of the I-PFO job with the specified job number.
I-PfoJob.writeJob	Writes the contents of an I-PFO job to the location with the specified job number.
I-PfoJob.deleteJob	Deletes the I-PFO job with the specified job number from the control.
I-PfoJob.lockJob	Locks a I-PFO job against modifications by other programs (TruControl 1000 and other programs)
I-PfoJob.releaseJob	Unlocks a I-PFO job for modifications by other programs.

Tab. 171

I-PfoJob.getDirectory

Returns a list of all I-PFO jobs on the device for a specified PFO.

Parameter	Type	Description
PfoNumber	Byte	Number of the PFO for which the list of I-PFO jobs is to be returned.

Parameters (input) Tab. 172

Parameter	Type	Description
JobCount	UInt16	Number of I-PFO jobs on the device for the specified PFO.
JobNo	Int16[JobCount]	Array with the numbers of the individual I-PFO jobs.
JobName	String[JobCount]	Array with the names of the individual I-PFO jobs.
JobSource	Int16[JobCount]	Array with information on the origin of the individual I-PFO jobs. 0: I-PFO job was created by TruControl 1000. 1: I-PFO job was created by TruTopsIPFO.
ChangeDate	DateTime[JobCount]	Array with the time stamps of the last modification of the individual I-PFO jobs.

Parameters (output) Tab. 173

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 174

I-PfoJob.readJob

Returns the contents of the I-PFO job of the desired PFO with the specified job number.

Parameter	Type	Description
PfoNumber	Byte	Number of the PFO for which the I-PFO job is to be returned.
JobNo	Int16	Number of the requested I-PFO job.

Parameters (input) Tab. 175

Parameter	Type	Description
JobName	String	Name of the requested I-PFO job.
JobSource	Int16	Information on the origin of the I-PFO job. 0: I-PFO job was created by TruControl 1000. 1: I-PFO job was created by TruTopsIPFO.
ChangeDate	DateTime	Time stamp of the last modification of the requested I-PFO job.
Job	String	Program code of the requested I-PFO job in a defined XML format.

Parameters (output) Tab. 176

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 177

I-PfoJob.writeJob

Writes the contents of an I PFO job for a PFO to the location with the specified job number.

Parameter	Type	Description
PfoNumber	Byte	Number of the PFO for which the I PFO job is to be written.
JobNo	Int16	Number of the I PFO job to be written/replaced.
JobSource	Int16	Information on the origin of the I-PFO job. 0: I-PFO job was created by TruControl 1000. 1: I-PFO job was created by TruTopsIPFO.
Job	String	Program code of the laser program in the defined XML format that is to be transferred to the control.

Parameters (input) Tab. 178

Parameter	Type	Description
JobName	String	Name of the I PFO job as stored in the XML data.
ChangeDate	DateTime	Time stamp of the last modification of the I PFO job as stored in the XML data.

Parameters (output) Tab. 179

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 180

I-PfoJob.lockJob

Disables the I PFO job for the desired PFO with the specified job number against modifications by other programs (including TruControl 1000).

Parameter	Type	Description
PfoNumber	Byte	Number of the PFO for which the I PFO job is to be disabled.
JobNo	Int16	Number of the I PFO job to be disabled.
AddInfo1	String	Additional information on disabling. This information may also be displayed in TruControl 1000 when a disabled I-PFO job is to be accessed and is also returned when a different client wants to disable the same I-PFO job.
AddInfo2	String	Additional information on disabling. Description: See also AddInfo1.

Parameters (input) Tab. 181

Parameter	Type	Description
LockAssigned	Boolean	Indicates whether the lock could be set. If TRUE, then the following output parameters contain information on their own client. If FALSE, then the following output parameters contain information on a different client which has set the lock first.
ClientInfo	String	Provides information on the client which has disabled the specified I-PFO job.
AddInfo1	String	Provides additional information on the disabling operation specified by the corresponding client when disabling.
AddInfo2	String	Provides further additional information on the disabling operation specified by the corresponding client when disabling.
Already-Locked	Boolean	Indicates whether the I PFO job has already been disabled before. In combination with the **LockAssigned** output parameter, it is possible to determine whether the lock had already been set by its own client or a different client.

Parameters (output) Tab. 182

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 183

I-PfoJob.releaseJob

Unlocks a I-PFO job for modifications by other programs. One restriction is that a client can unlock only those locks which it has set itself.

Parameter	Type	Description
PfoNumber	Byte	Number of the PFO for which the I PFO job is to be enabled.
JobNo	Int16	Number of the I PFO job to be enabled.

Parameters (input) Tab. 184

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 185

I-PfoJob.deleteJob

Deletes the I PFO job from the desired PFO with the specified job number from the control.

Parameter	Type	Description
PfoNumber	Byte	Number of the PFO for which the I PFO job is to be deleted.
JobNo	Int16	Job number of the I PFO job to be deleted from the control.

Parameters (input) Tab. 186

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 187

PFO file (TruTops PFO)

The methods for reading information on the PFO program files are to be found under "PfoFile" node.

The format of the PFO program files (.pfo) that can be read out and written via this interface comes from TruTops PFO.

Requirement

User level ≥ 40.

UaNodeId	Description
PfoFile.getPFOFileDirectory	Returns a list of all PFO files on the device.
PfoFile.readReadPFOFile	Returns the contents of the PFO file with the specified program number from the requested PFO.
PfoFile.writePFOFile	Writes the contents of a PFO file to the location with the specified program number in the specified PFO.
PfoFile.deletePFOFile	Deletes the PFO file with the specified program number of the specified PFO from the control.
PfoFile.lockPFOFile	Disables a PFO file against modifications by other programs (TruControl 1000 and other programs).
PfoFile.releasePFOFile	Enables a PFO file for modifications by other programs.

Tab. 188

PfoFile.getPFOFileDirectory

Returns a list of all PFO files on the device.

Parameter	Type	Description
PfoNumber	Byte	Number of the PFO of which the list of PFO files is to be returned.

Parameters (input) Tab. 189

Parameter	Type	Description
PFOFileCount	UInt16	Number of PFO files on the device.
ProgNo	Int16[PFOFileCount]	Array with the program number of the PFO files.
ProgName	String[PFOFileCount]	Array with the program names of the PFO files.

Parameters (output) Tab. 190

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 191

PfoFile.readPFOFile

Returns the contents of the PFO file with the specified program number from the requested PFO.

Parameter	Type	Description
PfoNumber	Byte	Number of the PFO of which the PFO file is to be returned.
ProgNo	Int16	Number of the PFO program of which the PFO file is to be returned.

Parameters (input) Tab. 192

Parameter	Type	Description
FileContent	ByteString	Contents of the requested PFO file (binary).

Parameters (output) Tab. 193

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 194

PfoFile.writePFOFile

Writes the contents of a PFO file to the location with the specified program number in the specified PFO.

Parameter	Type	Description
PfoNumber	Byte	Number of the PFO to which the PFO file is to be transferred.
ProgNo	Int16	Number of the program that contains the PFO file.
PFOFileContent	ByteString	Contents of the PFO file from TruTops (binary) to be transferred to the control.

Parameters (input) Tab. 195

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 196

PfoFile.lockPFOFile

Disables a PFO file against modifications by other programs (TruControl 1000 and other programs).

Parameter	Type	Description
PfoNumber	Byte	Number of the PFO for which the PFO file is to be disabled.
ProgNo	Int16	Number of the PFO file to be disabled.
AddInfo1	String	Additional information on the disabling. This information may also be displayed in TruControl 1000 if a disabled PFO file is to be accessed. It is also returned if a different client wants to disable this PFO file.
AddInfo2	String	Additional information on disabling. Description: See also AddInfo1.

Parameters (input) Tab. 197

Parameter	Type	Description
LockAssigned	Boolean	Indicates whether the lock could be set. If TRUE, then the following output parameters contain information on their own client. If FALSE, then the following output parameters contain information on a different client which has set the lock first.
ClientInfo	String	Provides information on the client who disabled the specified PFO file.
AddInfo1	String	Additional information on disabling specified by the corresponding client when disabling.
AddInfo2	String	Additional information on disabling. See also AddInfo1.
AlreadyLocked	Boolean	Indicates whether the PFO file has already been locked before. In combination with the **LockAssigned** output parameter, it is possible to determine whether the lock had already been set by its own client or a different client.

Parameters (output) Tab. 198

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 199

PfoFile.releasePFOFile

Enables a PFO file for modifications by other programs. One restriction is that a client can unlock only those locks which it has set itself.

Parameter	Type	Description
PfoNumber	Byte	Number of the PFO for which a PFO file is to be enabled.
ProgNo	Int16	Number of the PFO file to be enabled.

Parameters (input) Tab. 200

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 201

PfoFile.deletePFOFile

Deletes the PFO file with the specified program number of the specified PFO from the control.

Parameter	Type	Description
PfoNumber	Byte	Number of the PFO from which the PFO file is to be deleted.
ProgNo	Int16	Number of the PFO file to be deleted from the control.

Parameters (input) Tab. 202

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 203

6.8 Set-up

Set-up.writeAssetInformation

Writes the user's asset information to the laser.

Asset information is information on the laser "asset". This can be the following information:

- Location
- Cost center
- Responsibility
- etc.

Prerequisites

- User level ≥ 60
- Available as of TC 2.34

Parameter	Type	Description
Data	String	Asset information

Parameters (input) Tab. 204

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 205

System settings

Under "System Settings" node the system parameters of the CPX are read out that are configured in TruControl 1000 under *>Setup >System settings*.

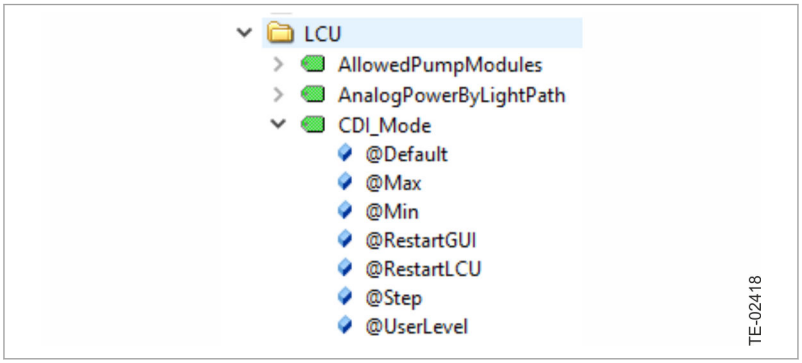
The system parameters are read only and cannot be changed via the OPC UA interface.

All system parameters are grouped into nodes, e.g. LCU, (see "Fig. TE-02418", pg. 126). Under "SystemSettings.AllParameters" node the system parameters are provided in an ungrouped format.

The UaNodeIds are defined with the following format: "Setting.Name_des_Systemparameters." Each of these nodes can contain additional features.

UaNodeId	Description
Setting.CDI_Mode	Current value of the system parameter.
Setting.CDI_Mode.Default	Default value of the system parameter.
Setting.CDI_Mode.Max	Maximum value of the system parameter.
Setting.CDI_Mode.Min	Minimum value of the system parameter.
Setting.CDI_Mode.RestartGUI	GUI is restarted if this system parameter is changed.
Setting.CDI_Mode.RestartLCU	Central processor is restarted if this system parameter is changed.
Setting.CDI_Mode.Step	Increment for this system parameter.
Setting.CDI_Mode.UserLevel	User level from which this system parameter can be changed.

System parameter, example CDI_Mode Tab. 206



System parameter, LCU node, example Fig. TE-02418

Switch functions

UaNodeId	Description
SwitchFunctions.switchPilotLaserON	Switches the pilot laser on.
SwitchFunctions.switchPilotLaserOFF	Switches the pilot laser off.

Tab. 207

SwitchFunctions.switchPilotLaserON

Switches the pilot laser on.

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value

Tab. 208

SwitchFunctions.switchPilotLaserOFF

Switches the pilot laser off.

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value

Tab. 209

Pulse synchronization

UaNodeId	Description
Pulse synchronization.setBothDelays	Writes the specified values in the parameters ProcessDelay and SystemDelay.
Pulse synchronization.setProcessDelay	Writes the specified value in the parameter ProcessDelay.
Pulse synchronization.setSystemDelay	Writes the specified value in the parameter SystemDelay.

Tab. 210

Pulse synchronization.setBothDelays

Parameter	Type	Description	Value range
System Delay	Double	System delay	0 - 500 ns
Process Delay	Double	Process delay	

Parameters (input) Tab. 211

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 212

Pulse synchronization.setProcessDelay

Parameter	Type	Description	Value range
Process Delay	Double	Process delay	0 - 500 ns

Parameters (input) Tab. 213

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 214

Pulse synchronization.setSystemDelay

Parameter	Type	Description	Value range
System Delay	Double	System delay	0 - 500 ns

Parameters (input) Tab. 215

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 216

6.9 Technology

LLKS Data

UaNodeId[1)]	Description
[<ComId>/<CompNum>].LLKsX.Available	0 = no data available
[<ComId>/<CompNum>].LLKsX.ConnectType	Connection type
[<ComId>/<CompNum>].LLKsX.CoreDiameter	Minor Diameter
[<ComId>/<CompNum>].LLKsX.NumericalAperture	Numerical aperture
[<ComId>/<CompNum>].LLKsX.FiberLength	Length of the optical fiber
[<ComId>/<CompNum>].LLKsX.MaxCWPow	Maximum continuous power
[<ComId>/<CompNum>].LLKsX.MaxPeakPow	Maximum peak power
[<ComId>/<CompNum>].LLKsX.WavelengthLow	Lower wavelength limit
[<ComId>/<CompNum>].LLKsX.WavelengthUP	Upper wavelength limit
[<ComId>/<CompNum>].LLKsX.Manufacturer	Manufacturer
[<ComId>/<CompNum>].LLKsX.DateOfManufacture	Production date
[<ComId>/<CompNum>].LLKsX.SerialNo	Serial number
[<ComId>/<CompNum>].LLKsX.ArtNo	Material number
[<ComId>/<CompNum>].LLKsX.OptionalArtNo	Optional material number
[<ComId>/<CompNum>].LLKsX.OptionalField	Optional field
1) ComId: ID of the component CompNum: NUM of the component X: Number of the LLK	

Tab. 217

Light path

UaNodeId	Description
xxx.@Name[1)]	Name of the light path.
xxx.@UserName[1)]	User name of the light path.
xxx.Active[1)]	Light path active: True: light path active. False: light path not active.
xxx.PilotLaserState[1)]	Status of the pilot laser: 0: pilot laser off. 1: pilot laser on. 2: pilot laser not present.
xxx.SIKState[1)]	Status of the SIK: 0: SIK closed. 1: SIK open. 4: SIK fault.
1) xxx: light path number: 000, 001, 002, 003, 004, 005, 006	

Tab. 218

UaNodeId	Description
Lightpath.getAllActive LightpathComponents	Provides all active components of all light paths. Contains additional information on the components. Filtering by a certain component possible.
Lightpath.getLightpath NumberForPFO	Provides the light path number that belongs to the PFO number.
Lightpath.getLightpath NumberForSFO	Provides the light path number that belongs to the SFO number.
Lightpath.getPFONum ForLightPath	Provides a byte mask of the PFO numbers in the selected light path.
Lightpath.getSFONum ForLightPath	Provides a byte mask of the SFO numbers in the selected light path.
Lightpath.getVisionLine OPCEndPoint	Provides the end point URL of the VisionLine OPC server in the selected light path. The client source IP is taken into account and the URL is adapted accordingly in order that the client can connect directly.
Lightpath.switchLightpath PilotlaserON	Switches the pilot laser of the light path on.
Lightpath.switchLightpath PilotlaserOFF	Switches the pilot laser of the light path off.

Tab. 219

Lightpath.getAllActiveLightpathComponents

Provides all active components of all light paths. Contains additional information on the components. Filtering by a certain component possible.

Requirement

Available as of TC 2.16.

Parameter	Type	Description
FilterComponentType	String	Name of the component type by which filtering is to be carried out.

Parameters (input) Tab. 220

Parameter	Type	Description
LpNum	Int16	List of light path numbers.
LpName	String	List of light path names.
LpComponentData	LightpathComponent DataStructure	Provides additional information on the component, such as name, ID, NUM, IP port, SIK and the current position.

Parameters (output) Tab. 221

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 222

Lightpath.getLightpathNumberForPFO

Provides the light path number that corresponds to the PFO number.

Requirement

Available as of TC 2.14.

Parameter	Type	Description
PFONum	Byte	PFO number

Parameters (input) Tab. 223

Parameter	Type	Description
LightPathNum	Int16	Light path number

Parameters (output) Tab. 224

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 225

Lightpath.getLightpathNumberForSFO

Provides the SFO number that corresponds to the PFO number.

Requirement

Available as of TC 3.38.

Parameter	Type	Description
SFONum	Byte	SFO number

Parameters (input) Tab. 226

Parameter	Type	Description
LightPathNum	Int16	Light path number

Parameters (output) Tab. 227

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 228

Lightpath.getPFONumForLightPath

Provides a byte mask of the PFO numbers in the selected light path.

Requirements

- User level ≥ 0
- Available as of TC 2.14

Parameter	Type	Description
LightPathNum	Int16	Light path number

Parameters (input) Tab. 229

Parameter	Type	Description
PFONum	Byte	Mask of the PFO numbers in the light path.

Parameters (output) Tab. 230

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 231

Lightpath.getSFONumForLightPath

Provides a byte mask of the SFO numbers in the selected light path.

Requirements

- User level ≥ 0
- Available as of TC 3.38

Parameter	Type	Description
LightPathNum	Int16	Light path number

Parameters (input) Tab. 232

Parameter	Type	Description
SFONum	Byte	Mask of the SFO numbers in the light path.

Parameters (output) Tab. 233

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 234

Lightpath.getVisionLineOPCEndPoint

Provides the end point URL of the VisionLine OPC server in the selected light path. The client source IP is taken into account and the URL is adapted accordingly in order that the client can connect directly.

Requirement

Available as of TC 2.14.

Parameter	Type	Description
LightPathNum	Int16	Light path number

Parameters (input) Tab. 235

Parameter	Type	Description
OPCEndpoint	String	End point URL (IP:Port) of the VisionLine OPC server.

Parameters (output) Tab. 236

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 237

Lightpath.switchLightpathPilotLaserON

Switches the pilot laser in the selected light path on.

Requirement

Available as of TC 2.14.

Parameter	Type	Description
LightPathNum	Int16	Light path number

Parameters (input) Tab. 238

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 239

Lightpath.switchLightpathPilotLaserOFF

Switches the pilot laser in the selected light path off.

Requirement

Available as of TC 2.1.

Parameter	Type	Description
LightPathNum	Int16	Light path number

Parameters (input) Tab. 240

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 241

6.10 HELP

Help.readAssetInformation

Provides the asset information of the laser (see "Set-up.writeAssetInformation", pg. 125).

Requirement

Available as of TC 2.34.

Parameter	Type	Description
Data	String	Asset information

Parameters (output) Tab. 242

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 243

Diagnostics

Diagnostics.getDiagnosticData

Provides the diagnostics data as a byte string and the file name to be used.

Parameter	Type	Description
StartDate	DateTime	Start date for the backup of the diagnostics data. Selected data will be written to the created backup package only after this date (time).

Parameters (input) Tab. 244

Parameter	Type	Description
FileName	String	File name which **must** be used for the data package. It is compulsory to use this file name, as otherwise further processing of the data cannot be guaranteed.
FileContent	ByteString	Binary data stream which can be written to the file system with the above-mentioned file name.

Parameters (output) Tab. 245

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 246

Help.DeviceData.getOperatingPoints

Supplies the parameters of the available operating points.

Requirement

Available from TC 3.34 on for TruMicro laser devices.

Parameter	Type	Description
OperationPointCount	UInt16	Number of operating points
OperatingPoints	String	Parameters of the operating points. Single operating point are separated with '. Parameter sind mit "," getrennt. Parameter per operating point: ▪ Base frequency [kHz] ▪ Burst count ▪ Short pulse energy [µJ] ▪ Burst energy [µJ] ▪ Pulse-on-Demand mode ▪ Short pulse duration min [ps] ▪ Short pulse duration max [ps] ▪ Trigger2Light ▪ Supported laser click[1)] ▪ Single shot capable[1)] ▪ Required options[1)] ▪ MaxBurstPatternLength (available from TC 3.50)
1) Available as of TC 3.38.		

Parameters (output) Tab. 247

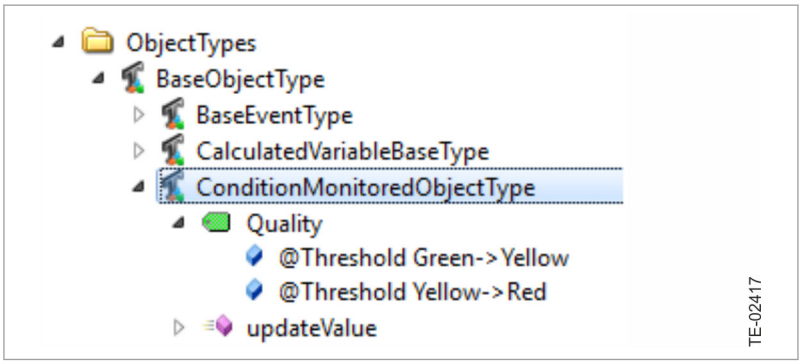
Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 248

6.11 Condition Monitoring

All objects in this section are part of the namespace *http://trumpf.com/UA/ConditionMonitoring*.

All "monitored properties" defined here are derived from the object type "ConditionMonitoredObjectType".



Condition Monitoring Fig. TE-02417

The "ConditionMonitoredObjectType" object type is derived directly from "BaseObjectType" and contains the following variables, properties and methods:

UaNodeId	Type	Description
[Object].Quality	UInt16	Variable with a value 0 – 100 which represents the state of a monitored property.
[Object].Quality. ThresholdGreenYellow	UInt16	Property with a value 0 – 100 which defines the threshold value of a "yellow" state
[Object].Quality. ThresholdYellowRed	UInt16	Property with a value 0 – 100 which defines the threshold value of a "red" state.
[Object].updateValue		Optional method (without further parameters) to initiate or immediately start the update of the "monitored properties".

Tab. 249

If necessary, the object type can generate an event of the type **MonitoredObjectExchangedEventType**.

This, however, applies only to objects for which exchange can be carried out during the runtime.

The following monitored properties are supported by the UA server depending on the laser type:

UaNodeId	Description
CoolingCircuit	Monitored property for cooling circuit maintenance (water change).
TankConductivity	Monitored property for the conductivity of the water in the tank.
TankNiveau	Monitored property for the level of the water in the tank.
MainsVoltageHigh	Monitored property for the monitoring of the voltage supply upward.
MainsVoltageLow	Monitored property for the monitoring of the voltage supply downward.
[ID/NUM].ASV-SIKCounterX	Monitored property for the number of switch cycles.

Tab. 250

6.12 Nameplate

"Nameplate" node contains values that are relevant for an electronic nameplate. The values are also to be found under other nodes, but only scattered. Here, they are grouped under one node.

Requirement

User level ≥ 0.

UaNodeId	Type	Description
@EquipmentNo	String	Equipment number of the laser.
@LaserFamily	String	Laser device family
@LaserName	String	Laser device name (name defined by the user)
PDCounter.ControlOnHours	Double	Time in hours how long the control was switched on up to now.
Nameplate. CurrentMaintenanceWork	String	Currently pending maintenance tasks. Time stamp in UTC.
Nameplate.LCUComponents	String	List of all currently registered LCU components.
Nameplate.OperatingHoursHot	Double	Time in hours how long the laser beam source was switched on up to now.
Nameplate.MessageHistory	String	Message history of the last 48 hours, however maximum of 10 entries. Time stamp in UTC.
Nameplate.ModificationHistory	String	Modification/update history of the last 14 days, however maximum of 5 entries. Time stamp in UTC.
SystemSettings.OptionCoding	UInt32	Activated options
Nameplate. PastMaintenanceWork	String	Past maintenances. Time stamp in UTC
UI-UserLevelMax	UInt8	Highest user level of all connected UIs.

Tab. 251

6.13 QDS

All objects under the node QDS (Quality Data Storage) belong to the name space*http://trumpf.com/UA/QDS*.
All objects below this node e. g. QDS.LaserControl_0, are not created before the component is registered with the method **registerQDS**.

Condition: user level ≥ 0.

UaNodeId	Type	Description
QDS.registerQDS	Method	Registers a QDS component.
QDS.LaserControl_0	Directory	"LaserControl" directory for QDS data.
QDS.LaserControl_0.BufSize	UInt16	Buffer size for the QDS data.
QDS.LaserControl_0.Counter	UInt16	Number of buffered QDS data items.
QDS.LaserControl_0.Monitored	UInt16	Number of clients that have subscribed to the counter variable.
QDS.LaserControl_0.Status	String	QDS status in data format JSON[1)]
QDS.LaserControl_0.VersionMajor	String	Major version.
QDS.LaserControl_0.VersionMinor	String	Minor version.
QDS.LaserControl_0.delete	Method	Deletes data in the buffer memory.
QDS.LaserControl_0.read	Method	Reads data in the buffer memory.
QDS.LaserControl_0.reset	Method	reset
QDS.LaserControl_0.setBufSize	Method	Sets the buffer size.
QDS.LaserControl_0.write	Method	Writes data to the buffer memory.
1) JavaScript Object Notification		

Tab. 252

QDS.registerQDS

Registers a QDS component.

Condition: user level ≥ 0, available from TC 2.34 on.

Parameter	Type	Description
Name	String	Container type (component name)
Major version	UInt16	Major version
Minor version	UInt16	Minor version
Index	UInt16	Index of the QDS component

Parameters (input) Tab. 253

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 254

QDS.LaserControl_0.delete

Deletes data in the buffer memory.

Requirement

Available as of TC 2.34.

Parameter	Type	Description
Counters	Int64[]	Array of the indices that are deleted in the QDS data buffer.

Parameters (input) Tab. 255

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 256

QDS.LaserControl_0.read

Reads data in the buffer memory.

Requirement

Available as of TC 2.34.

Parameter	Type	Description
Count	UInt16	Number of the QDS data to be read.

Parameters (input) Tab. 257

Parameter	Type	Description
Data	String	Array with the QDS data
Counters	String	Array of the counters (Idx)

Parameters (output) Tab. 258

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 259

QDS.LaserControl_0.reset

reset

Requirement

Available as of TC 2.34.

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 260

QDS.LaserControl_0.setBufSize

Setting the buffer size.

Requirement

Available as of TC 2.34.

Parameter	Type	Description
Data	String	Asset information

Parameters (output) Tab. 261

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 262

QDS.LaserControl_0.write

Writes data to the buffer memory.

Requirement

Available as of TC 2.34.

Parameter	Type	Description
Data	String	Array with the QDS data
Counters	String	Array of the counters (Idx)

Parameters (input) Tab. 263

Type	Description
UaStatus	OpcUa_Good if the function was successfully called up.

Return value Tab. 264

6.14 Power supply

Requirements

- User level ≥ 10.
- Laser device family 20-09-10-A2000 and 20-09-10-A2001.

The following parameters are available under the "Power supply" node:

UaNodeId	Type	Unit	Description
L1.Voltage	Double	V	Voltage phase 1
L2.Voltage	Double	V	Voltage phase 2
L3.Voltage	Double	V	Voltage phase 3

Measured values, power supply node Tab. 265

7. Control of the laser devices via OPC UA Multi PLC interface

If an MPI slot is configured as OPC UA, the following functions of a laser device can be controlled:

- request the laser device.
- Switching on and off.
- define the light path and laser parameters during the runtime of a laser program.
- Starting and ending selected laser program.
- synchronize the sequence of a laser program to external events.
- Synchronizing external sequences to the sequence of a laser program.
- Evaluation of important status signals of the laser device.
- Switch off the laser in the event of an external fault.
- acknowledge monitoring messages of the laser device.
- re-initialize of the control unit of the laser device after a fault.
- Switch the pilot laser, if available, on and off.

Input words

Variable "Activation.MPI-Slotxx.Input"

"Access Level" or "User Level": readable, writeable

All input words of the selected MPI slot xx are read or written; example: Activation.MPISlot10.Input.

Other variables:

- Individual input values, example: input word 0: Activation.MPI-Slot10.Input.Word00
- Individual signals; example: LIGHT_PATH_No: Activation.MPI-Slot10.Input.Word01.LIGHT_PATH_No

  Additional example: EXT_ACTIVATION: Activation.MPI-Slot10.Input.Word00.EXT_ACTIVATION

Description of the available input words and signals:
(see "Input words", pg. 145)
(see "Signal descriptions", pg. 169)

Output words

Variable Activation.MPI-Slotxx.Output

Access Level or User Level: readable

All output words of the selected MPI slot xx are read; example: Activation.MPISlot10.Output.

Other variables:

- Individual input words, example: output word 0: Activation.MPI-Slot10.Output.Word00
- Individual signals, example: LIGHT_PATH_No: Activation.MPI-Slot10.Output.Word01.LIGHT_PATH_No

  Additional example: EXT_ACTIVATION_ACTIVE: Activation.MPI-Slot10.Output.Word00.EXT_ACTIVATION_ACTIVE

Description of the available output words and signals:
(see "Output words", pg. 154)
(see "Signal descriptions", pg. 169)

7.1 Input words and output words

Input words

The laser device is controlled via input words, which are sent to the control of the laser device by one of the following units:

- Fieldbus master e.g. control (PLC) of a processing station:
  - to a fieldbus slave of the CPX (central processor of the laser control).
  - to a fieldbus slave of an interface of the TRUMPF peripheral bus.
- Master with a parallel input/output interface e.g. control (PLC) of a processing station to the parallel I/O interface of a CPX or of a TRUMPF peripheral bus.
- PC or laptop with OPC client via Ethernet to the CPX.

The control of the laser device has the following functions:

- It accepts the data of the received input words.
- It interprets data and converts it into the following parameters:
  - control commands
  - Parameters for laser programs

Parameters for laser programs

The signals for the input words 3 ... 63 can be used in the laser programs (see "Input words 3 ... 63: parameters for laser programs", pg. 150).

Signals of the input words which have been freely parameterized may not be used in laser programs.

Freely parameterizable input words

The signals of the input words 3 ... 63 can be assigned freely in the >*Setup* >*Interface parameters* dialog (see "Input words 3 ... 63: Further input parameters", pg. 152).

Input words which transmit signals for laser programs may not be freely parameterized.

Input mask

Which input signals of the MPI slot are forwarded to the laser control can be set in the >*Input mask* dialog. If an input signal is deactivated, this signal is not forwarded to the laser control.

The following input signals can be masked:	
▪ EXT_ACTIVATION ▪ LOCK_OPERATING_PANEL ▪ LASER_ON ▪ LASER_STANDBY (Only for laser device with Laser-Standby-Modus) ▪ PSTART_STATICAL ▪ PSTART_DYN ▪ PROG_STOP ▪ RESET	▪ FAULT_EXT ▪ INQUIRY_1 ▪ INQUIRY_2 ▪ Reserved (bit 12 always 1) ▪ Reserved (bit 11 always 1) ▪ UNSYNC_PSTART ▪ PILOT_LASER_ON ▪ REQUEST_LASER

Tab. 266

Factory setting 0xFFFF: The input signals are activated and are forwarded to the laser control.

Configuration of the input mask (see "Setting interface parameters", pg. 263).

Inverting mask

Which input signals of the MPI slot are forwarded in inverted form to the laser control can be set in the >*Inverting mask* dialog. If an input signal is activated, this signal is forwarded to the laser control in inverted form.

The following input signals are invertible:	
▪ EXT_ACTIVATION ▪ Reserved (bit 6 always 0) ▪ LASER_ON ▪ LASER_STANDBY (Only for laser device with Laser-Standby-Modus) ▪ PSTART_STATICAL ▪ PSTART_DYN ▪ PROG_STOP ▪ RESET	▪ FAULT_EXT ▪ INQUIRY_1 ▪ INQUIRY_2 ▪ Reserved (bit 12 always 0) ▪ Reserved (bit 11 always 0) ▪ UNSYNC_PSTART ▪ PILOT_LASER_ON ▪ REQUEST_LASER

Tab. 267

Factory setting 0x0000: The inverting mask is deactivated. The input signals are forwarded in non-inverted form.

Configuration of the inverting mask (see "Setting interface parameters", pg. 263).

Overview of the input words

Word no.	Functions
0	**Basic laser functions** External activation, laser on, program start, program stop, reset, request laser, unsynchronized start
1	**Number of light path and program** Selection of light path and number of the laser program to be started.
2	**Number of the application data** Application data for the monitoring of focusing optics (CFO or BEO smart). **Number of the laser device** The control of the laser device checks whether the correct laser is being activated.
3 : 13	**Input words 3 ... 13: IW3 ... IW13** Set value for a laser parameter or other input parameters.
14 15	**Input words 14 and 15: IW14 and IW15** ▪ If system parameter PLC_ExtendBaseIoFunction = 1: extended laser functions. ▪ If system parameter PLC_ExtendBaseIoFunction = 0: – Set value for a laser parameter or – Other input parameters.
16 : 63	**Input word 16 .. 63: IW16 .. IW63** if TruControl 1000 version ≥ 1.84.0[1)]: ▪ Set value for a laser parameter or ▪ Other input parameters.
1) Not available for the following CPX variants: ▪ CPX V1 standard, 18-21-85-A1 ▪ CPX V1 small, 18-22-21-A1 ▪ CPX V2 standard, 18-21-85-A3 ▪ CPX V2 small, 18-21-85-A4	

Input words 0 to 63 Tab. 268

Input word 0: laser base functions

A signal is active if the bit value is = "1".

Word.Bit	Signal name	Functions
0.15	EXT_ACTIVATION	**External activation** The laser device is controlled externally by means of a parallel input/output interface (TRUMPF Peripheral Bus), fieldbus or OPC.
0.14	LOCK_OPERATING_PANEL	**Locking the operating panel** The operating panel can only be used as display.
0.13	LASER_ON	**Switching on the laser** The laser power supply is turned on.
0.12	LASER_STANDBY	**Activating the laser Standby mode** Only required for laser devices with laser Standby mode.
0.11	PSTART_STATICAL	**Static program start** Static start of a laser program defined in input word 1.
0.10	PSTART_DYN	**Dynamic program start** Dynamic start of a laser program defined in input word 1.
0.9	PROG_STOP	**Cancel/stop program** A laser program started with “PSTART_DYN” is cancelled or stopped.
0.8	RESET	**Reset** Fault and monitoring messages are acknowledged. In case of fault, the control unit of the laser device is reinitialized.
0.7	FAULT_EXT	**External fault** The laser is switched off and active laser programs are cancelled.
0.6	INQUIRY_1	**Query inputs 1 and 2** The procedure of a laser program is synchronized to the state of this signal if the wait functions "W1H", "W2H", "W1L" or "W2L" are used in the laser program.
0.5	INQUIRY_2	
0.4	LOCK_PROG_PARAM	▪ Only for TruMicro: Lock operating point parameters. ▪ **Reserved** for all other laser devices.
0.3	SIMULATION	**Simulation mode** The simulation mode is activated.
0.2	UNSYNC_PSTART	**Unsynchronized activation** A program start is accepted at the earliest possible time without synchronization to the LASER_READY signal.
0.1	PILOT_LASER_ON	**Switching on the pilot laser** The pilot laser is always switched on even if the laser device has not been assigned.
0.0	REQUEST_LASER	**Requesting the laser** The laser device is requested by an external control unit.
Description (see "Signals of the input word 0", pg. 169).		

Tab. 269

Input word 1: light path and laser program number

A signal is active if the bit value is = "1".

Word.Bit	Signal name	Functions
1.15 : 1.8	LIGHT_PATH_No: bit 7 : LIGHT_PATH_No: bit 0	**Number of the selected light path (bits 7 ... 0)** An external control defines the light path to be used by the laser device for laser power output. Number of light path numbers: max. 255 dec = FF hex Format: hexadecimal. Example: light path no. 19 dec = 0001 0011 = 13 hex
1.7 : 1.0	PROGRAM_No: bit 7 : PROGRAM_No: bit 0	**Number of the selected laser program (bits 7 ... 0)** An external control defines the number of the laser program to be run. Number of laser programs: max. 200 dec = C8 hex, without program number 0. Format: hexadecimal. Example: program no. 25 dec = 0001 1001 = 19 hex
Description (see "Signals of the input word 1", pg. 179).		

Tab. 270

Input word 2: application data set and laser device number

A signal is active if the bit value is = "1".

Word.Bit	Signal name	Functions
2.15 : 2.8	AD_No: bit 7 : AD_No: bit 0	**Number of the application data set (bits 7 ... 0)** An external control unit defines the number of the application data set whose data is to be used for monitoring a focusing optics (CFO or BEO Smart). Number of selectable application data sets: max. 255 dec = FF hex Format: hexadecimal. Example: Application data set no.: 69 dec = 0100 0101 bin = 45 hex
2.7 : 2.0	LASER_No: bit 7 : LASER_No: bit 0	**Number of the selected laser device (bits 7 ... 0)** An external control unit defines the number of the laser device from which to request laser power. Number of laser device numbers: max. 255 dec = FF hex Format: hexadecimal. Example: Laser device no. 53 dec = 0011 0101 bin = 35 hex
Description (see "Signals of the input word 2", pg. 181).		

Tab. 271

Input words 3 ... 63: parameters for laser programs

A signal is active if the bit value is = "1".

Word no.	Parameter
3 : 13	Parameters for laser programs
14, 15	**Parameters for laser programs** if system parameter **PLC_ExtendBaseIoFunction** = 0.
16 ... 63	**Parameter for laser programs**, if TruControl 1000 version ≥ 1.84.0[1)].
1) Not available for the following CPX variants: ▪ CPX V1 standard, 18-21-85-A1 ▪ CPX V1 small, 18-22-21-A1 ▪ CPX V2 standard, 18-21-85-A3 ▪ CPX V2 small, 18-21-85-A4	

Tab. 272

The parameters for laser programs are listed in the following table.

Parameters for laser programs			
Laser parameters			
– Frequency	– Number of pulses	– Start ramp	– Pulse duration
– PK pulse number	– Pulse gap power	– End ramp	
– Laser power			
PFP Control and PierceLine			
– Pulse power (IW3)	– Average laser power (IW4)	– Frequency (IW5)	– PLS parameters (IW6 + IW7)
PCP			
– Monitoring	– Fault		
TruFiber			
– Pulse modulation • frequency • pulse duration	– BrightLine Mode[1)] • power core • power ring	– VariMode[1)]	
TruMicro			
– Base frequency	– Energy level	– Short pulse duration	– Short pulse count
– Work frequency	– Beam splitter	– Burst pattern	
PFO program			
– PFO program number	– PFO coordinate X, Y, Z	– PFO scaling X, Y	– Image processing mode
– PFO transformation X, Y, Z	– Scan width X, Y (only for PFO 20-2 PO)	– PFO coordinate Defokus	– PFO program runs
– PFO fieldbus control	– PFO rotation	– Simple Coordination on-the-fly mode • X, Y, Z feed	• Start delay
Process characteristic curve (PCC)			
– PCC →	– PCC #		
SeamLine Remote (SL)			
– SL: Feed rate	– SL: Configuration	– SL: Seam number	– SL: Control
BrightLine Weld / FusionLine,if a component FCM is available in the light path			
– Processing	– Power share core (for BrightLine Weld only)		
Cutting parameters for CW laser and laser with **CutAssist module**			
– Cutting speed	– Max. cutting speed	– Cutting parameter set number	
TPC program			
– Temperature when TPC mode = Temperature direct			
Motorized beam splitter			
– Beam splitter position			
PMF			
– Focus position / lens position	– RM: Position		
Traceability data			
– Seam ID • Start address • Number of words	– Workpiece serial number[2)] • Start address • Number of words	– Workpiece ID[2)] • Start address • Number of words	
1) With laser devices of the TruFiber P Series (FD89)			
2) In the properties of a laser program (see "Parameter transfer to the laser program settings", pg. 191).			
Description (see "Parameters for laser programs", pg. 182).			

Tab. 273

Input words 3 ... 63: Further input parameters

Word no.	Parameter
3 : 13	▪ Default value If the parameter has been defined as the default value, it can be used for laser programs. ▪ Other input parameters: – PLC input (CV) x words[1], x = 6, 8 or 16 – PAUSE (ESC)[2] – Line Laser (PFO) – FCU/PCU lid state – PLC input (VLCK) 13 words[3] – Sensor Relay Input x Words, x = 9, 16, 32 or 48
14 15	If system parameter **PLC_ExtendBaseIoFunction** = 0: ▪ Default value ▪ Other input parameters
16 : 63	If TruControl 1000 version ≥ 1.84.0[4]: ▪ Default value ▪ Other input parameters
1) Parameter for the image processing system VisionLine. 2) Requirement: See technical information "Energy Save Concept", document no.: 21-A0-71-CP. 3) Parameter for the image processing system VisionLine OCT Check with focusing optics BEO and OCT sensor, available from TruControl 1000 version 3.46 on. 4) Not available for the following CPX variants: ▪ CPX V1 standard, 18-21-85-A1 ▪ CPX V1 small, 18-22-21-A1 ▪ CPX V2 standard, 18-21-85-A3 ▪ CPX V2 small, 18-21-85-A4	
Description (see "Other input parameters", pg. 193).	

Tab. 274

Alternative input words 14 and 15: Extended laser functions

Note

The advanced laser functions of the input words 14 and 15 are available if the system parameter **PLC_ExtendBaseloFunction** = 1

A signal is active if the bit value is = "1".

Word.Bit	Signal name	Functions
14.15 : 14.5	–	**Reserved**
14.4	MACHINE_COOLING_ON	**Switching on external cooling** The external control requests the laser control to switch on the external cooling, for example, of a TRUMPF laser machine.
14.3	ENERGY_SAVE_ON[1)]	**Switch on energy-saving mode[1)]** The external control unit requests the laser control to switch the laser device to the energy-saving mode.
14.2	POWER_GATE	**POWER_GATE** The laser beam is suppressed.
14.1	ENERGY_SAVE_OFF[1)]	**Switching off the energy-saving mode[1)]** The external control unit requests the laser control to switch off the energy-saving mode.
14.0	LASERCONTROL_OFF	**Switching off the laser control unit** The external control unit sends a signal to the laser control to switch off.
15.15 : 15.0	–	**Reserved**
1) Prerequisites: ▪ CPX V3 or V4 is available. ▪ The input **24 V_STDBY** at X1 of the PBT board is supplied with voltage. ▪ LCU_EnergySaveMode > 0 For further information: Technical information "Energy Save Concept", doc. no. 21-A0-71-CP. Not available for laser devices with CPX V20.		
Description (see "Signals of the alternative input words 14 and 15", pg. 195).		

Tab. 275

Output words

Using output words, the laser device provides:

- Status information.
- Measured values.
- Calculated values supported by measurements for:
  - Average power
  - Pulse energy.

Freely parameterizable output words

The signals of the output words 2 ... 63 can be assigned freely in the *>Setup >Interface parameters* dialog (see "Output words 2 ... 63: Other output parameters", pg. 166).

Output mask

In the *>Output mask* dialog you can set which internal laser signals are forwarded by the laser control to the outputs of an MPI slot, even if the laser device is not assigned to the MPI slot, i.e. **LASER_ASSIGNED** is not active.

The following output signals can be masked:	
▪ EXT_ACTIVATION_ACTIVE	▪ **SET3**
▪ LASER_IS_ON	▪ SIK_OPEN
▪ LASER_READY	▪ **MONITORING**
▪ PROG_ACTIVE	▪ **FAULT_LASER**
▪ PROG_COMPLETED	▪ LASER_ASSIGNED
▪ **PROG_CANCELLED**	▪ **POWER_LIMIT**
▪ **SET1**	▪ **PILOT_LASER_IS_ON**
▪ **SET2**	▪ **LASER_WARNING_LAMP_IS_ON**

Tab. 276

Factory setting 0xB707: The output mask is active for all signals in **bold**. These signals are forwarded by the laser control to the outputs of the MPI slot, even if the laser device is not assigned to the MPI slot.

Configuration of the output mask (see "Setting interface parameters", pg. 263).

Overview of the output words

Word no.	Functions
0	**Laser device status** External actuation active, laser is on, laser is ready, program is active, program exited, laser malfunction, laser assigned.
1	**Light path and laser program number** Number of the light path and laser program set in the laser device.
2	**Status of focusing optics and position of the shutter switches**
3	**Number and status of the laser device**
4	**Message code ID and NUM, if output word 3, bit 0 = "0"** These message codes signal the component (ID) and, in the case of multiple identical components, the component in question (NUM). **or** **Measured value identifier if output word 3, bit 0 = "1"** The measured value identifier indicates which measured value is transmitted via output word 5 (e.g. distance from the focusing plane to the surface of the workpiece).
5	**Message code CODE, if output word 3, bit 0 = "0"** These message codes specify component monitoring or a component malfunction. **or** **Measured value if output word 3, bit 0 = "1"** Measured value, whose identifier is transmitted in output word 4 (e.g., distance from the focusing plane to the surface of the workpiece).
6	**Laser power** **CW laser:** Measured laser power **Pulsed laser:** Calculated value of the average power
7	**Pulsed laser: pulse energy** The pulse energy is a calculated value. **CW laser:** Not used
8 : 13	**Reserved for additional output parameters**
14 15	**Other status signals of the laser device** if system parameter PLC_ExtendBaseIoFunction = 1: Optimized standby mode is on, laser control is on, machine cooling is on or **Reserved for additional output parameters** if system parameter PLC_ExtendBaseIoFunction = 0.
16 : 63	**Reserve for further output parameters** Available, if TruControl 1000 Version ≥ 1.84.0[1)].

1) Not available for the following CPX variants:

- CPX V1 standard, 18-21-85-A1
- CPX V1 small, 18-22-21-A1
- CPX V2 standard, 18-21-85-A3
- CPX V2 small, 18-21-85-A4

Output words 0 to 63 Tab. 277

Output word 0: Standard output signals

A signal is active if the bit value is = "1".

Word.Bit	Signal name	Functions
0.15	EXT_ACTIVATION_ACTIVE	**External activation is active** The laser device can be controlled externally by means of a parallel input/output interface (TRUMPF Peripheral Bus), fieldbus or OPC.
0.14	LASER_IS_ON	**Laser is switched on** The laser is now in a state that enables it to generate laser light.
0.13	LASER_READY	**Laser is ready** The laser programs can be started.
0.12	PROG_ACTIVE	**Laser program is active** A laser program has been started and is being executed.
0.11	PROG_COMPLETED	**Laser program is completed** A laser program has been run and is completed.
0.10	PROG_CANCELLED	**Laser program has been cancelled** A laser program has been cancelled during execution.
0.9	SET_1	**Set outputs 1 ... 3** External sequences can be synchronized with the sequence of a laser program if set functions (S1H, S1L, S2H, S2L, S3H, S3L) are used in the laser programs.
0.8	SET_2	
0.7	SET_3	
0.6	SIK_OPEN	**The safety circuit (SIK) is open** The safety circuit (SIK) of a shutter or of a shutter switch in the selected light path is interrupted.
0.5	MONITORING	**A monitoring message of the laser device is active** A limit value for a monitoring message was exceeded or fallen short of.
0.4	FAULT_LASER	**A fault message of the laser device is active** A limit value for a fault message was exceeded or fallen short of.
0.3	LASER_ASSIGNED	**The laser device is assigned to an external control unit** Acknowledgement of the input signal "REQUEST_LASER".
0.2	POWER_LIMIT	**The power limit of the laser has been reached** (only with pulsed lasers) The state can be corrected by adapting the laser parameters.
0.1	PILOT_LASER_IS_ON	**Pilot laser is switched on** At least one pilot laser is switched on.
0.0	LASER_WARNING_LAMP_IS_ON	**The laser warning light is on** The capacitors of the laser device contain enough energy to allow laser light to be generated.
Description (see "Signals of output word 0", pg. 199).		

Tab. 278

Output word 1: light path and laser program number

A signal is active if the bit value is = "1".

Word.Bit	Signal name	Functions
1.15 : 1.8	LIGHT_PATH_No: bit 7 : LIGHT_PATH_No: bit 0	**Acknowledgement of the number of the currently set light path (bits 7 ...0)** The control of the laser device outputs the number of the currently set light path. This enables an external control unit to check if the requested light path corresponds to the light path that has been set. Format: hexadecimal. Example: light path no. 19 dec = 0001 0011 = 13 hex
1.7 : 1.0	PROGRAM_No: bit 7 : PROGRAM_No: bit 0	**Acknowledgement of the number of the laser program selected (bits 7 ... 0)** The control of the laser device outputs the number of the currently set laser program. This enables an external control unit to check if the requested laser program corresponds to the laser program that has been set. Format: hexadecimal. Example: program no. 25 dec = 0001 1001 bin = 19 hex
Description (see "Signals of output word 1", pg. 208).		

Tab. 279

Output word 2: Status of the focusing optics, slot messages and positions of the shutter switches

A signal is active if the bit value is = "1".

Word.Bit	Signal name	Functions
2.15	PROTECTIVE_GLASS_ LIFETIME_REACHED[1)]	**Limit value MAX2 of the protective glass contamination has been reached** The 2nd monitoring message of the protective glass contamination according to the application data set has been reached (orange range).
2.14	PROTECTIVE_GLASS_ CONTAMINATED[1)2)]	**Limit value MAX1 of the protective glass contamination has been reached** The 1st monitoring message of the protective glass contamination according to the application data set has been reached (yellow range).
2.13	APPLICATION_VALUES_ IN_RANGE[1)]	**Application values crossjet pressure, exhaustion and shielding gas are in the admissible range** All values monitored according to the application data record are within the admissible limits (green zone).
2.12	FOCUSING_OPTICS_READY/ LIGHTPATH_COMP_READY[1)]	**The focusing optics or light path component is ready for operation.**
2.11	FOCUSING_OPTICS_ AVAILABLE/LIGHTPATH_ COMP_AVAILABLE[1)]	**Focusing optics or light path component is enabled.** A focusing optics or light path component must be enabled via the dialog *>Technology >Light path*. Further information: Software manual "TruControl 1000", Dok.-Nr. 10-06-80-A1-CR. **Note:** The signals of the bits 15 ... 12 of output word 2 are only valid if the output signal **FOCUSING_OPTICS_AVAILABLE** is active.
2.10	–	**Reserve**
2.9	MONITORING_ SLOT	**A local monitoring has occurred for this slot** The external control unit connected to the slot must reset the monitoring via the input signal **RESET**.
2.8	FAULT_SLOT	**A local fault has occurred for this slot** The external control unit connected to the slot must reset the fault via the input signal **RESET**.
2.7 : 2.0	SHUTTER_SWITCH 8 (VW8) : SHUTTER_SWITCH 1 (VW1)	**Position of the shutter switches** The control unit of the laser device outputs the available shutter switches. The position of the shutter switches can be evaluated by an external control unit. 1 = shutter switch closed 0 = shutter switch open
1) These signals indicate the current status of focusing optics or a light path component in the selected light path. 2) **PROTECTIVE_GLASS_CONTAMINATED** signal is not available for focusing optics of series **BEO Smart**.		
Description (see "Signals of the output word 2", pg. 209).		

Tab. 280

Output word 3: Number and extended status of the laser device

A signal is active if the bit value is = "1".

Word.Bit	Signal name	Functions
3.15 : 3.8	LASER_No: bit 7 : LASER_No: bit 0	**Number of the laser device (bits 7 ... 0)** The control of the laser device outputs the number which is assigned to the laser device. This enables an external control unit to check if the requested laser device has responded. Format: hexadecimal. Example: laser device no. 25 dec = 0001 1001 bin = 19 hex
3.7	–	**Reserve**
3.6	–	**Reserve**
3.5	EXT_COOLING_ CIRCUIT_IS_ON	Functionality A): External cooling circuit is switched on. Functionality B): The laser device requires cooling water.
3.4	ALIVE_TICK	**Laser control is active** With this signal an external control can determine that the laser control is switched on.The signal changes status between High and Low with a frequency of 1 Hz.
3.3	CALIBRATION/SELF_TEST	**The laser is being calibrated** or **Laser selftest is active**
3.2	–	**Reserve**
3.1	MAINTENANCE	**Maintenance of the laser device is required**
3.0	MEASURING_VALUE_OUT_ ACTIVE	**Output word 4 outputs measured value ID** Output of the corresponding measured value is made via output word 5.
Description (see "Signals of the output word 3", pg. 213).		

Tab. 281

Output word 4: Message code byte 1 (ID), byte 2 (NUM)

The control unit of the laser device shows via bit 0 of output word 3 which data is transferred via output word 4:

- Bit 0 of output word 3 = "0" (inactive): the message code byte 1 (ID) and byte 2 (NUM) is transferred via output word 4.
- Bit 0 of output word 3 = "1" (active): the alternative output word 4 is used to transfer a measured value ID.

A signal is active if the bit value is = "1".

Word.Bit	Signal name	Functions
4.15 : 4.8	MESSAGE_ID: bit 7 : MESSAGE_ID: bit 0	**Message code ID (bits 7 ... 0)** The control unit of the laser device uses the message code (ID) to output the code of the component that has resulted in a message. The message code can be evaluated by an external control unit. Format: Hexadecimal. Example: 0Exx hex = ASV 3 (VW) has resulted in a message.
4.7 : 4.0	MESSAGE_NUM: bit 7 : MESSAGE_NUM: bit 0	**Message code NUM (bits 7 ... 0)** If there are several components of the same type, the control unit of the laser device uses the message code (ID) to output the NUM that indicates the component that has resulted in the message. The message code can be evaluated by an external control unit. Format: Hexadecimal. Example: xx08 hex = The 4th component of the same type has resulted in a message.
Description (see "Signals of output word 4", pg. 216).		

Tab. 282

Alternative output word 4: measured value ID

The control unit of the laser device shows via bit 0 of output word 3 which data is transferred via output word 4:

- Bit 0 of output word 3 = "0" (inactive): the message code byte 1 (ID) and byte 2 (NUM) is transferred via output word 4.
- Bit 0 of output word 3 = "1" (active): the alternative output word 4 is used to transfer a measured value ID.

A signal is active if the bit value is = "1".

Word.Bit	Signal name	Functions
4.15 : 4.0	MEASURING_VALUE_ID: bit 15 : MEASURING_VALUE_ID: bit 0	**ID of the measured value in the output word 5 (bits 15 ... 0)** The control unit of the laser device outputs the identification of the measured value being transferred in output word 5. This enables an external control unit to determine the measured value transmitted in output word 5. Format: Hexadecimal.
Description (see "Signals of the alternative output word 4", pg. 218).		

Tab. 283

Output word 5: Message code byte 3 and byte 4

The control unit of the laser device shows via bit 0 of output word 3 which data is transferred via output word 5:

- Bit 0 of output word 3 = "0" (inactive): the message code byte 3 and byte 4 is transferred via output word 5.
- Bit 0 of output word 3 = "1" (active): the alternative output word 5 is used to transfer a measured value.

  The control unit of the laser device transfers the measured value ID that indicates the type of measured value being transferred via the alternative output word 4.

A signal is active if the bit value is = "1".

Word.Bit	Signal name	Functions
5.15 : 5.0	MESSAGE_CODE: bit 15 : MESSAGE_CODE: bit 0	**Message code of the reporting component (bits 15 ...0)** The control unit of the laser device outputs the message code of the reporting component. The message code can be evaluated by an external control unit. Format: Hexadecimal. **Note:** For messages, the message codes are also displayed on the operating panel.
Description (see "Signals of output word 5", pg. 219).		

Tab. 284

Alternative output word 5: Measured value

The control unit of the laser device shows via bit 0 of output word 3 which data is transferred via output word 5:

- Bit 0 of output word 3 = "0" (inactive): the message code byte 3 and byte 4 is transferred via output word 5.
- Bit 0 of output word 3 = "1" (active): the alternative output word 5 is used to transfer a measured value.

  The control unit of the laser device transfers the measured value ID that indicates the type of measured value being transferred via the alternative output word 4.

A signal is active if the bit value is = "1".

Word.Bit	Signal name	Functions
5.15 : 5.0	MEASURING_VALUE: bit 15 : MEASURING_VALUE: bit 0	**Measured value of the measured value ID transmitted in the output word 4 (bits 15 ... 0)** The control unit of the laser device outputs the identification of the measured value being transferred in output word 4. An external control unit can evaluate the transferred measured value and use it e.g. for correction purposes. Format: Hexadecimal. Example: Distance between focal point plane and workpiece surface.
Description (see "Signals of alternative output word 5", pg. 220).		

Tab. 285

Output word 6: Laser power

A signal is active if the bit value is = "1".

Word.Bit	Signal name	Functions
6.15 : 6.0	LASER_POWER: bit 15 : LASER_POWER: bit 0	**CW laser: laser power (bits 15 ... 0)** The control unit of the laser device outputs the measured laser power. Format: Hexadecimal. Significance of the bits is defined: ▪ Bit 15 = high. ▪ Bit 0 = low. Resolution: 1 W/digit Value range: 0 W ... 65535 W. **Pulsed lasers: average power (bits 15 ... 0)** The average laser power value, calculated on the basis of measurements, is output by the control unit of the laser device. An external control unit can evaluate this laser parameter and use it for optimization of the laser utilization. Format: Hexadecimal. Significance of the bits is defined: ▪ Bit 15 = high. ▪ Bit 0 = low. Resolution: 0.1 W/digit Value range: 0.0 W ... 6553.5 W. The system parameter **PLC_PowerOutTick** can be used to set the interval at which the value is updated. PLC_PowerOutTick = 0: In case of normal utilization of the system, the value is updated approx. every 0.2 s. PLC_PowerOutTick ≥ 20 ms: In case of normal utilization of the system, the value is updated at the interval given.
Description (see "Signals of output word 6", pg. 221).		

Tab. 286

Output word 7: Pulse energy

Pulse energy is only available with pulsed laser devices.

In CW lasers and with TruMicro series lasers this output word is not used in the standard assignment.

A signal is active if the bit value is = "1".

Word.Bit	Signal name	Functions
7.15 : 7.0	ENERGY: bit 15 : ENERGY: bit 0	**Pulse energy (bits 15 ... 0)** The pulse energy value of a laser device, calculated on the basis of measurements, is output by the control unit of the laser device. An external control unit can evaluate this laser parameter and use it for optimization of the laser utilization. Format: Hexadecimal. Significance of the bits is defined: ▪ Bit 15 = high. ▪ Bit 0 = low. Resolution: 0.01 J/digit Value range: 0.00 J ... 655.35 J In case of normal utilization of the system, the measured values are updated approx. every 0.5 s. The pulse energy of the last measured pulse is output and this value then remains present until the next change.
Description (see "Signals of output word 7", pg. 222).		

Tab. 287

Output words 2 ... 63: Other output parameters

Word no.	Parameter
2 : 7	▪ **Default value**: In the preceding sections, you will find an overview of the default values preset by TRUMPF. ▪ **Other output parameters**:
8 : 13	▪ Reserve. ▪ Other output parameters.
14 15	▪ Reserve. ▪ Further output parameters if system parameter **PLC_ExtendBaseIoFunction** = 0.
16 : 63	▪ Reserve. ▪ Further output parameters if TruControl 1000 version ≥ 1.84.0[1)].
1) Not available for the following CPX variants: ▪ CPX V1 Standard, 18-21-85-A1. ▪ CPX V1 Small, 18-22-21-A1. ▪ CPX V2 Standard, 18-21-85-A3. ▪ CPX V2 Small, 18-21-85-A4.	

Tab. 288

The other output parameters are listed in the following table.

Parameter		
PLC output (CV) x words[1], x = 6, 8, 16, 32, 40, 48 or 56 output words		Device level (ESC)[2]
Slot level (ESC)[2]	Slot RTTO (ESC)[2]	Slot state (ESC)[2]
Power regulation average value (LLR)	Protective glass contamination (CFO)	Power average filtered (LMP)
Exhaustion pressure (CFO)	Shielding gas flow (CFO)	Pulse energy count (LMP)
PPC measured power	Conductivity (CLK)	Pulse energy
Input word 0 ... Input word 63	Active PFO program number	Temperature base plate (SBO)
IP address ETH2	Active workplace	State for fieldbus (SBO)
IP address LAN	Energy laser program	Pulse energy 1kJ
IP address SERVICE	Energy laser program 10J	Pulse energy 1mJ
Laser power nominal	Request laser slot 1-16	Pulse power average
Laser type	Request laser slot 17-32	Melt travel monitoring state (PFO)
Measured power (PCP)	EM pulse divider (DLC)	TPC state
SeamLine Remote result (7 words)	Amplifier pulse repetition rate (DLC)	Process light sensing status (DLC)
SeamLine Remote status	XVar.1 .... XVar.32	Actual temperature (TPC)
Focus or lens position (PMF)	Laser power EM digits (DLC)	Nominal temperature (TPC)
Rotary module orientation (PMF)	Protective glass contamination (PGM)	FCU/PCU lid state
Crossjet pressure (CFO)	Pulse counter	Short pulse duration (PDC)
BEO application number (MIO/PGM)	Shielding gas flow (MIO)	MDE gas flow (MIO)
Crossjet Pressure (MIO)	Purging gas flow (MIO)	Power portion core (FCM)
Burst Pattern (DLC/WPCI)	Base frequency (DLC/WPCI)	Operating frequency (DLC/WPCI)
Energy level (WPCI)	PFO fieldbus control state	Purging gas pressure (CRDS)
VariMode state (HPCC/SCPR)	PLC output (VLCK) 04 words[3]	Protective glass contamination (PFO)
Conductivity (CCB/ECC/CCR)	VisionLine Check Status[3]	PCP measurement
Sensor Relay Output x Words, x = 9, 16, 32 or 48		
1) Parameter for the image processing system VisionLine.		
2) Requirement: See technical information "Energy Save Concept", document no.: 21-A0-71-CP.		
3) Parameter for image processing system VisionLine OCT Check, available from TruControl 1000 version 3.46 on.		
Description (see "Other output parameters", pg. 223).		

Other output parameters Tab. 289

Alternative output words 14 and 15: Further standard output signals

Note

The other standard output signals of the output words 14 and 15 are available if the system parameter **PLC_ExtendBaseIo-Function** = 1.

Word.Bit	Signal name	Functions
15.15 ... 15.0 14.15 ... 14.8	–	**Reserve**
14.7	COOLING_ LOCKED	If this signal is active the external cooling water supply must be stopped.
14.6	EMISSION_ON	Signal is active if power > 0 W, i.e. when power is output. **Requirements** If a central shutter (VZ) is available, it must be open. The signal can only be evaluated if the pulse duration and the pulse gap are twice as large as PLC_PowerOutTick. Example: PLC_PowerOutTick = 80 ms → pulse duration and pulse gap ≥ 160 ms.
14.5	ALIGNMENT_ RECOMMENDED	On laser devices with LLK-U (option) the beam alignment must be corrected if this signal is active.
14.4	MONITORING_ PROCESS	A monitoring message has appeared at the allocated MPI slot during the machining process. Check the quality of the machined workpiece. This message can be acknowledged with RESET.
14.3	FAULT_ PROCESS	A malfunction occurred at the allocated MPI slot during the machining process. Check the quality of the machined workpiece. This message must be acknowledged with RESET.
14.2	MACHINE_ COOLING_ READY	**External cooling is active.** External cooling, for example of a laser machine, is active. Cooling water flow is present.
14.1	ENERGY_SAVE_ IS_ON[1)]	**Energy-saving mode of the laser control has been switched on.**1)
14.0	LASER_ CONTROL_IS_ON	**Laser control is on.**

1) Prerequisites:

- CPX V3 or V4 is available.
- The input **24 V_STDBY** at X1 of the PBT board is supplied with voltage.
- LCU_EnergySaveMode > 0

For further information: Technical information "Energy Save Concept", doc. no. 21-A0-71-CP.

Not available for laser devices with CPX V20.

Description (see "Signals of the alternative output words 14 and 15", pg. 233).

Tab. 290

7.2 Signal descriptions

Signals of the input word 0

EXT_ACTIVATION

Function	Input signal determines if the laser is controlled: ▪ from the operating panel (OP). ▪ by an external control unit (e.g. PLC).
Active signal	The laser can be operated by an external control unit. The following functions can be executed via the operating panel: ▪ it is NOT possible to start laser programs. ▪ laser parameters can be transferred to the laser device. ▪ messages can be acknowledged.
Feedback	With active input signal **EXT_ACTIVATION**, the output signal **EXT_ACTIVATION_ACTIVE** is activated.
Information	A laser device is only assigned to an external control unit if it is requested via the input signal **REQUEST_LASER**. When, additionally, the **LOCK_OPERATING_PANEL** input signal is active, laser parameters can no longer be transferred to the laser device via the operating panel.

Tab. 291

FAULT_EXT

Function	The external control reports an error in the environment of the installation.
Active signal	The laser device carries out the following actions: ▪ The central shutter is closed. ▪ The laser has been switched off. ▪ Laser pulses are interrupted immediately. ▪ The active laser program is cancelled. ▪ The power supply to the following components is switched off: – Pump sources – Cooling unit – Optical components ▪ Laser standby mode is ended (only for laser devices with laser standby mode).
Feedback	A monitoring message is displayed. The output signal **MONITORING** becomes active.
Information	The laser device can only be controlled again via the interface when the **RESET** input signal has been activated. If TruControl version ≥ 3.10.0, the input signal **FAULT_EXT** must be deactivated beforehand. **Note:** A fault message is **not** displayed, and the output signal **FAULT_LASER** is **not** activated.

Tab. 292

INQUIRY_1 and INQUIRY_2

Function	Both input signals **INQUIRY_1** and **INQUIRY_2** allow synchronization of the sequence of a laser program to external sequences.
Information	If a line of the laser program contains a maintenance instruction (W1H, W1L, W2H or W2L), this line is not processed until the respective input signal **INQUIRY_1** (with W1H, W1L) or **INQUIRY_2** (with W2H, W2L) adopts the required state (high or low).

Tab. 293

LASER_ON

Functions	The laser power supply is turned on.
Active signal	The laser power supply is switched on. Laser light is not yet generated.
Feedback	About 5 – 10 s after activation of the **LASER_ON** input signal, the output signals **LASER_IS_ON** and **LASER_READY** are activated simultaneously. **Laser device with laser standby mode:** About 8 – 15 s after activation of the **LASER_ON** input signal, the **LASER_IS_ON** output signal is activated.
Information	The laser power supply remains switched on as long as the input signal **LASER_ON** is active. If the output signal **LASER_READY** is active, a laser program can be started. **Notes:**: ▪ The laser must not be switched on and off cyclically (multiple times per minute). Switching off and on in succession is allowed once, e.g. if ALIGNMENT_RECOMMENDED is active (see "ALIGNMENT_RECOMMENDED", pg. 233). ▪ The laser device is also switched on by the signal **LASER_ON** if: – the external activation is not active (output signal **EXT_ACTIVATION_ACTIVE** is inactive). – the laser device is not assigned to the external control unit (output signal **LASER_ASSIGNED** is inactive). ▪ **TruDisk, TruDiode** When the cooling water temperature of the series TruDisk and TruDiode laser devices is below a certain threshold value, the laser cannot be switched on within the time mentioned above. The cooling water can be heated up to reduce the waiting time. Heating of the cooling water is activated automatically by the **LASER_ON** signal or via the *Laser* button in the *>Manual mode* main operation. Heating of the cooling water is aborted or not activated if the switching function *Force manual operation* is active or if a user is registered at the laser control with user level ≥ 70. The main operation *>Manual mode* and the function *Force manual operation* are described in the Software manual "TruControl 1000", doc. no. 10-06-80-A1-CR. ▪ **Laser device with laser standby mode:** If the **LASER_IS_ON** output signal is active, the laser device can be switched to the laser standby mode by activating the **LASER_STANDBY** input signal. About 2 – 5 s after activation of the **LASER_STANDBY** input signal, the **LASER_READY** output signal is activated.

Tab. 294

LASER_STANDBY

Functions	**Activating the laser standby mode.** The laser power is stabilized in the laser standby mode. This signal is only required for laser devices with laser standby mode.
Active signal	The laser power defined for the laser standby mode is generated without regulation and conducted from the central shutter into the absorber.
Feedback	About 2 – 5 s after activation of the **LASER_STANDBY** input signal, the **LASER_READY** output signal is activated.
Using the signal	On the basis of the following case description you can find out whether the data about the laser standby mode are relevant for your situation: ▪ When a laser device with CPX control without laser standby mode is prepared for its use, the information on the laser standby mode in this interface description can be ignored. ▪ When a laser device with CPX control with laser standby mode is prepared for its use, please observe the information on the laser standby mode in this interface description. ▪ You replace an older laser device with LCU control and laser standby mode with a laser device with CPX control without laser standby mode. Their task is to adapt existing control programs to an external control (e.g. PLC). – In this case, you are often able to continue using the existing control programs without change, in particular if you wait for the LASER_READY signal to continue a process. The LASER_STANDBY signal is inactive. – The standby line must be removed in laser programs which you want to copy from the old to the new laser device. The opening and closing of the central shutter must be programmed explicitly. Further information: Technical information "TruControl 1000 for new users", Doc.-No. 21-A0-57-CP.
Information	▪ If the **LASER_READY** output signal is active, a laser program can be started. The laser standby mode remains active as long as the signal **LASER_STANDBY** is active. ▪ The function **Stdby** has been automatically entered in the first line of the second column of a laser program. ▪ The service life of the pump sources (lamps or laser diodes) can be increased and the power required by the laser can be reduced if laser standby mode is switched off during long waiting times between laser programs. ▪ The laser device is also switched to the laser standby mode by the signal **LASER_STANDBY** if: – the external activation is not active (**EXT_ACTIVATION_ACTIVE** inactive). – the laser device has not been assigned to the external control unit (output signal **LASER_ASSIGNED** inactive).

Tab. 295

LOCK_OPERATING_PANEL

Functions	The laser device cannot be controlled any longer via the operating panel (OP).
Active signal	The following functions can be executed via the operating panel: ▪ laser parameters can no longer be transferred to the laser device. ▪ messages can further be acknowledged.
Feedback	None
Information	An active input signal **LOCK_OPERATING_PANEL** is only considered if the input signal **EXT_ACTIVATION** is active and if the laser device has been assigned.

Tab. 296

LOCK_PROG_PARAM

Functions	Only for TruMicro: Disables the changing of operating point parameters as long as the signal is active.
Active signal	▪ If the signal is active when the program starts, the parameters of the default operating point are applied. ▪ If the signal is active during a line change, the parameters of the last active operating point are applied.
Inactive signal	▪ If the signal is not active when the program starts, the INIT line is executed again and the changed operating point parameters are applied. ▪ If the signal is not active during a line change, the parameters are applied from the new operating point when the operating point is switched.

Tab. 297

PROG_STOP

Functions	Running laser programs can be ended or canceled.
Active signal	**Canceling laser program:** A running laser program is canceled when the following conditions are fulfilled: ▪ The input signal **PROG_STOP** is activated for > 1 ms **and** ▪ in the laser program **Pulse type** ≠ "Continuous wave" (for CW lasers) and **Pulse number** ≠ ∞ ("Continuous") are set. **Ending laser program:** A running laser program is ended when the following conditions are fulfilled: ▪ The input signal **PROG_STOP** is activated for > 1 ms **and** ▪ in the laser program **Pulse type** ≠ "Continuous wave" (for CW lasers) **Pulse number** = ∞ ("Continuous") are set.
Feedback	**Laser program canceled:** ▪ Output signal **PROG_ACTIVE** becomes inactive. ▪ A monitoring message is displayed. ▪ The output signals **MONITORING** and **PROG_CANCELLED** become active. The output signal **PROG_COMPLETED** is not activated because the laser program has not been completely executed. **Laser program completed:** The **PROG_ACTIVE** becomes inactive and the **PROG_COMPLETED** output signal becomes active for approx. 6 s. If, before the 6 s have expired, a change in assignment takes place, and the **PLC_ClearProgCompletedOnSlotChange** system parameter is 1, the **PROG_COMPLETED** signal will be set to inactive.
Information	A canceled laser program cannot be continued. When the **PROG_STOP** input signal is active, and a laser program is supposed to be started: ▪ a monitoring message is displayed. ▪ the output signal **MONITORING** becomes active. ▪ the laser program is not started.

Tab. 298

PILOT_LASER_ON

Function	Switching the pilot laser of the laser device on and off.	
Active signal	The pilot laser is switched on.	
Inactive signal	Pilot laser is switched off.	
Feedback	The **PILOT_LASER_IS_ON** output signal becomes active when a pilot laser of the laser device is switched on.	
Information	The input signal **PILOT_LASER_ON** and the software function "Pilot laser on" in the dialog are linked by OR.	
	Different hardware types can exist in a laser device as a pilot laser. The existing hardware type is defined as follows in the dialog *>Setup >System settings >PilotLaserType*: 1 = Central pilot laser 2 = Pilot laser in shutter switch (VW..) 3 = Pilot laser in central shutter (VZ)	
	The following pilot laser modes can be selected in dialog *>Setup >System settings >PilotLaserMode* for a pilot laser of the hardware type 2 (pilot laser in shutter switch):	
	0 =	Pilot laser is constantly off.
	1 =	The pilot laser is on when the shutter switch is in the neutral position.
	2 =	The pilot laser is on when the safety circuit is open.
	3 =	When the shutter switch is closed, the pilot laser is switched on: ▪ via the dialog *>Producing >Light path overview >Pilot laser overview* ▪ **or** or when the **PILOT_LASER_ON** input signal is active.
	4 =	Pilot laser is switched on when the safety circuit is open: ▪ via the dialog *>Producing >Light path overview >Pilot laser overview* ▪ **or** or when the **PILOT_LASER_ON** input signal is active.
	5 =	The pilot laser is switched on: ▪ via the dialog *>Producing >Light path overview >Pilot laser overview* ▪ **or** or when the **PILOT_LASER_ON** input signal is active.
General condition	With the pilot laser modes 3 ... 5 the following boundary condition applies: ▪ An external control unit sends a valid program number to the control unit of the laser device via Multi PLC interface (e.g. fieldbus): – with fixed preset light path in the laser program (e.g. light path = 2). – **or** with external preset light path (light path = IW) and defines the number of the light path in the input word 1. With the pilot laser mode 4 the following boundary condition applies: ▪ The safety circuit (SIK) is open. **Exceptions:** If an external control: ▪ sends a laser program no. "0" then the last laser program no. not equal to "0" is used. ▪ sends a light path no. "0" then the last light path no. not equal to "0" is used.	
	The pilot laser is switched off irrespective of the PilotLaserMode set if the 1-channel safety circuit (ASV3, X24 connector) is used for internal functions (e.g. protection of laser light cable in the user's compartment) **and** this is interrupted.	

Tab. 299

PSTART_DYN

Functions	Laser programs can be started by a short impulse of > 1 ms.
Active signal	A laser program is started by the ascending edge of the **PSTART_DYN** input signal when the **LASER_READY** output signal is active. The program number is read at the start of the laser program. To do so the program number must be continuously available to the inputs signals **PROGRAM_No: bit 0 ... bit 7**. **Notes:**: ▪ If an attempt is made to start a laser program while the **LASER_READY** output signal is inactive, a monitoring message is displayed and the **MONITORING** output signal is activated. ▪ The **PSTART_DYN** input signal is triggered by the signal edges and starts a laser program only once. Before starting a program again, the input signal must be switched to inactive for at least 50 ms.
Feedback	The **PROG_ACTIVE** output signal is activated if a laser program is started and remains active until the laser program is completed or canceled. The **PROG_COMPLETED** output signal is activated for approx. 6 s after the laser program has been completed. The **PROG_COMPLETED** output signal becomes prematurely inactive if, before the 6 s have expired, ▪ a laser program is started again. ▪ a change in assignment takes place, and the **PLC_ClearProgCompletedOnSlotChange** system parameter is 1. The output signals **MONITORING** and **PROG_CANCELLED** become active if a laser program is canceled.
Information	The delay in time from the activation of the **PSTART_DYN** input signal to the delivery of the laser power at the processing point is called laser delay time (LEZ). **LEZ for pulsed lasers:** 35 ... 55 ms, if optical components are moved, depending on the optical components used. 15 ... 20 ms, if the optical components are already set correctly. < 1 ms depending on the times set per software for delay and debouncing time if the functions of the real-time interface are used. **LEZ for laser devices with laser Standby mode:** 35 .. 55 ms (opening time of the shutter) If the input signal **PSTART_DYN** or **PSTART_STATCAL** is activated during the execution of a laser program: ▪ a fault message will be displayed. ▪ the output signals **MONITORING** and **FAULT_LASER** will be activated. ▪ the running laser program will be cancelled. ▪ the output signal **PROG_CANCELLED** will be activated. **Note:** If **Pulse type** = "Continuous wave" (for CW lasers) or **Pulse number** = ∞ ("Continuous") are set within the laser program, the laser program can only be ended by activating the input signals **PROG_STOP**.

Tab. 300

PSTART_STATICAL

Function	Laser programs can be started, ended or cancelled.
Active signal	A laser program is started when: ▪ the input signal **PSTART_STATICAL** for > 1 ms is activated. ▪ **and** the output signal **LASER_READY** is active. The program number is read at the start of the laser program. To do so the program number must be continuously available to the inputs signals **PROGRAM_No: bit 0 ... bit 7**. **Note:** If an attempt is made to start a laser program while the **LASER_READY** output signal inactive, a monitoring message is displayed and the **MONITORING** output signal is activated.
Setting signal to inactive	**Ending a laser program:** A laser program started with **PSTART_STATICAL** is ended when: ▪ the **PSTART_STATICAL** input signal is set to inactive prior to the end of the laser program ▪ **and** in the laser program **Pulse type** ≠ "Continuous wave" (for CW lasers) **Pulse number** = ∞ ("Continuous") is set. **Canceling a laser program:** A laser program started with **PSTART_STATICAL** is cancelled when: ▪ the **PSTART_STATICAL** input signal is set to inactive prior to the end of the laser program ▪ **and**in the laser program **Pulse type** ≠ "Continuous wave" (for CW lasers) and **Pulse number** ≠ ∞ ("Continuous") is set.
Feedback	The **PROG_ACTIVE** output signal is activated if a laser program is started and remains active until the laser program is completed or canceled. The **PROG_COMPLETED** output signal is activated after the laser program has ended. The output signals **MONITORING** and **PROG_CANCELLED** become active if a laser program is canceled.
Information	The delay in time from the activation of the **PSTART_STATICAL** input signal to the delivery of the laser power at the processing point is called laser delay time (LEZ). **LEZ for pulsed lasers:** 35 ... 55 ms if optical components are moved depending on the used optical components. 15 .. 20 ms if optical components are already set correctly. < 1 ms, depending on the times set per software for delay and debouncing time if the functions of the real-time interface are used. **LEZ for laser devices with laser Standby mode:** 35 .. 55 ms (opening time of the shutter) When the **PSTART_DYN** input signal is activated while a laser program that was started with the **PSTART_STATICAL** input signal is running: ▪ no static program start is executed. ▪ a monitoring message is displayed. ▪ the output signal **MONITORING** becomes active.

Tab. 301

REQUEST_LASER

Functions	request the laser device.
Active signal	The laser device is requested by an external control unit.

Feedback	If the laser device has been assigned to the requesting external control unit, the output signal **LASER_ASSIGNED** becomes active.
Information	**The laser device is connected to only 1 external control unit:** If only 1 external control unit is connected to the laser device, the input signal **REQUEST_LASER** must be activated by an external control unit as long as the external control unit controls the laser device. The **REQUEST_LASER** signal can also be set to permanently active in this case. **The laser device is connected with two or several external control units via additional interfaces (parallel I/O, OPC):** If two or more external control units are connected with the laser device, the input signal **REQUEST_LASER** must be activated by an external control unit as long as the external control unit controls the laser device. This guarantees that the laser device can be assigned to other external control units. **Assignment procedure** The assignment of the laser is carried out according to priority and in case of the same priority according to the round robin algorithm. Further information: (see "LASER_ASSIGNED", pg. 200). The following parameters can be used to output a bit-coded overview showing which MPI slots have requested the laser using the **REQUEST_LASER** signal: ▪ Request Laser Slot 1-16 for MPI slots 1 to 16. ▪ Request Laser Slot 17-32 for MPI slots 17 to 32. Both parameters must be assigned to an output word in the >*Setup* >*Interface parameters* dialog. The bit for its own MPI slot number is not set. Further information: (see "Other output parameters", pg. 223). **Sleep mode** If the system parameter **LCU_Sleep_1_Timeout** > 0, the input signal **LASER_ON** is active and the input signal **REQUEST_LASER** remains inactive for this time, the Sleep mode 1 is activated. For further information: see technical information "Energy Save Concept", doc. no. 21-A0-71-CP. An external control unit can start a laser program of type "No pulse triggering", even if the laser device is not assigned to the external control unit.

Tab. 302

RESET

Functions	Acknowledges laser device messages and initializes the laser device control again when a malfunction has occurred.
Active signal	The Reset function is carried out when the **RESET** input signal is active for > 1 ms.
Requirement	IF TruControl version ≥ 3.10.0 then **EXT_ACTIVATION** must be set and **FAULT_EXT** must not be set.
Feedback	With “Monitoring”: ▪ Output signal **MONITORING** becomes inactive. ▪ The displayed monitoring message is deleted. ▪ The laser program is continued. With "Monitoring" and "Program cancelled": ▪ Output signal **MONITORING** becomes inactive. ▪ The displayed monitoring message is deleted. ▪ Output signal **LASER_READY** becomes active. ▪ The laser programs can be started. With “Monitoring” and “Laser fault”: The laser control unit is reinitialized. **Note:** If no fault has occurred during initialization, the laser can be switched on again via the interface and the operation can be continued.
Information	The signal **RESET** and the message behavior are influenced by different system parameters. Further information: (see "Messaging and RESET behavior", pg. 258). If the frequency of monitoring and fault messages increases, maintenance and repair measures may be necessary.

Tab. 303

UNSYNC_PSTART

Functions	Switching on the mode for unsynchronized triggering.
Active signal	The mode for an unsynchronized triggering has been switched on.
Feedback	None
Information	In the mode for unsynchronized triggering, the laser device activates the output signal **LASER_READY**, even if the laser device has not been assigned. An active output signal **LASER_READY** means in case of active input signal **UNSYNC_PSTART**: ▪ The start of a laser program is possible. ▪ The laser device can also be assigned to another external control unit. ▪ The safety circuit SIK is closed. ▪ The laser is in the laser standby mode (only laser devices with laser standby mode). ▪ No fault of the laser device has been detected. ▪ A laser program has not been cancelled and acknowledged via the output signal **PROG_CANCELLED**. Further information: Activation of the laser device (see "Triggering", pg. 236). Timing diagram(see "Unsynchronized Triggering of a laser device", pg. 250).

Tab. 304

SIMULATION

⚠ WARNING

Laser light may be emitted in the simulation mode if the safety circuit is closed.

Laser light can irreparably damage your eyes.

Laser light can burn your skin.

- Before working in the simulation mode, make sure that the safety circuit is open and cannot be closed by unauthorized persons!
- Make sure that all persons in the dangerous area will wear laser safety goggles suitable for the wave length of the laser light (see notice signs).
- If there are persons within the danger area, surround the processing point with a protective cover.
- If you work within the danger area, do not reach between focusing optics and workpiece.
- If you notice troubles with your eyes or burnt skin, please contact a physician even if you are not able to determine laser light as the direct cause.

Functions	Activates the simulation mode
Active signal	The external control unit requests the laser control unit of the laser device to activate the simulation mode.
Requirement	A central shutter (VZ) must be available in the laser device. For laser devices without a central shutter this signal remains ineffective.
Feedback	The LASER_READY output signal also becomes active with open SIK.
Information	The simulation mode closes the central shutter. The laser beam is guided into the absorber of the central shutter. Laser light is **not** emitted in this case. In simulation mode, a laser program can be also started with open SIK. The sequence of a laser program can be observed on the workpiece in combination with a pilot laser. For further information: see software manual "TruControl 1000", doc. no. 10-06-80-A1-CR. The simulation mode can also be activated in the *>Setup* dialog.

Tab. 305

Signals of the input word 1

LIGHT_PATH_No: Bit 7 ... Bit 0

Functions	The 8 input signals **LIGHT_PATH_No: bit 7 ... bit 0** are used to select the light path through which the laser device is to output laser power.
Coding of the light path no.	The significance of the 8 input signals is defined from **LIGHT_PATH_No: bit 7** (high) to **LIGHT_PATH_No: bit 0** (low). Format: Hexadecimal. Number of selectable light path numbers: 255 dec = FF hex. Example: light path no. 25 dec = 0001 1001 bin = 19 hex
Feedback	Via the output signals **LIGHT_PATH_No: bit 7 ... bit 0** the control of the laser device acknowledges output 1, the number of the currently set light path.
Information	**Without ILN (Interface LaserNetwork)** The number of the light path must be predefined by the external control unit. The numbers of the light paths assigned to the laser device are defined in the dialog *>Technology >Light path*. **With ILN:** The light path number must be identical to the number which is transmitted from the laser device to the processing station via the LLK coding cable assigned to the light path. Further information: Interface description "ILN3", Dok.-Nr. 30-72-69-A1-BA. **With TLN-fc (TRUMPF LaserNetwork fieldbus coded)**: The light paths are fieldbus coded. External control units do not need to predefine a light path. The light paths are assigned via the dialog *>Setup >Multi PLC interface (MPI)*. Further information: Technical information "TRUMPF LaserNetwork (TLN and TLN-fc)", Doc.-No. 21-01-38-CP.

Tab. 306

PROGRAM_No: bit 7 ... bit 0

Functions	Using the 8 input signals **PROGRAM_No: Bit 7 ... Bit 0** the laser program to be started is selected.
Coding of the program no.	The significance of the 8 input signals is defined from **PROGRAM_No: bit 7** (high) to **PROGRAM_No: bit 0** (low). Format: Hexadecimal. Laser programs with special function ■ 240 dec ... 250 dec (F0 hex ... FA hex), examples: – 240 dec. (F0 hex): BeamCalibration, available if the laser device is equipped with option LLK-U (see "ALIGNMENT_RECOMMENDED", pg. 233). – 244 dec. (F4 hex): lamp factor optimization (LFO) – 245 dec. (F5 hex): Laser self test ■ 0 dec. (0 hex): Program number 0 is used to set the laser device to a defined status if all signals are set to 0. For a laser program to be created with program number 0, the system parameter **min. program no.** in the *>Setup >Program setting* menu must be set to 0. Inadmissible values: ■ 201 dec ... 239 dec (C9 hex ... EF hex) ■ 251 dec ... 255 dec (FB hex ... FF hex) Number of selectable laser programs: 200 dec = C8 hex, without program number 0.
Feedback	If, during program start, an inadmissible program number is applied by the input signals **PROGRAM_No: bit 7 ... bit 0**: ■ a monitoring message is displayed. ■ the output signal **MONITORING** becomes active. ■ no laser program is started.
Information	■ The input signals **PROGRAM_No: bit 7 ... Bit 0** are read when the input signal **PSTART_STATICAL** or **PSTART_DYN** has been activated. ■ The next program number may already be created when the **PROG_ACTIVE** output signal is active after the start of a laser program. ■ If the **PROG_COMPLETED** signal becomes active after the completion of a laser program and the new program number has already been applied, the control unit keeps the **LASER_READY** signal inactive until the optical components are set to the light path of the new laser program. ■ If in a laser device of the TruMicro 5000 series the basic frequency in the new laser program was changed, the signal **LASER_READY** will remain active until the new basic frequency has been set, which may take up to 30 sec. ■ If the program number of a laser program of the type "No pulse triggering" has been created, the signal **LASER_READY** is active and the laser program can be started without assignment. ■ An active laser program is canceled if, during the program number change, the program type changes as follows: – From "Standard" to "No pulse triggering" – From "No pulse triggering" to "Standard" ■ If a laser program of the type "No pulse triggering" is started by the external control unit with assignment, the **Program active** symbol is not displayed on the user interface of TruControl 1000. ■ If an external control unit without assignment wants to start a laser program of the type "Standard", the laser program is not started and no message is output.

Tab. 307

Signals of the input word 2

AD_No: bit 7 ... bit 0

Functions	The 8 input signals **AD_No: bit 7 ... bit°0** are used to select the number of the application data set with whose data a connected focusing optics is monitored.
Coding of the application data set no.	The significance of the 8 input signals is defined from **AD_No: bit 7** (high) to **AD_No: bit 0** (low). Format: Hexadecimal. Number of selectable data sets: 1 ... 255 dec = 01 ... FF hex. Example: application data set no. 55 dec = 0011 0111 bin = 37 hex. Monitoring of the connected focusing optics is switched off by selecting the number "0".
Feedback	Status of the focusing optics via bits 15 ... 8 of the output word 2.
Information	The dialog >*Setup* >*Application data* allows to create new application data records and to modify application data.

Tab. 308

LASER_No: bit 7 ... 0

Function	The 8 input signals **LASER_No: bit 7 ... bit 0** are used by an external control unit to select the number of the laser device which is expected to provide laser power.
Coding of the laser device no.	The significance of the 8 input signals is defined from **LASER_No: bit 7** (high) to **LASER_No: bit 0** (low). Format: Hexadecimal. Number of the selectable laser devices: 255 dec = FF hex. Example: laser device no. 12 dec = 0000 1011 bin = 0C hex.
Feedback	The control of the laser device supplies the number assigned to the laser device via the output signals **LASER_No: bit 7 ... bit 0** of output word 3.
Information	In the case of laser devices with LaserNetwork interface (ILN), the numbers of the laser devices are transmitted from the laser device to external control units by means of connected LLK coding cables. Without ILN, the external control units must define the numbers of the laser devices. If the variable "MPI_CheckLaserNumber" is set to TRUE in the dialog >*Setup* >*System settings* >*Field bus*: ▪ the control of the laser device compares the laser number at the inputs **LASER_No: bit 7 ... bit 0** with the number assigned to the laser device. ▪ the control unit of the laser device indicates a fault if the laser numbers compared are not identical. The comparison of the laser number is switched off, if the variable "MPI_CheckLaserNumber" is set to FALSE. The laser number is assigned to a laser device via the dialog >*Setup* >*System settings* >*Fieldbus* of the variable "MPI_LaserNumber".

Tab. 309

Parameters for laser programs

For the parameter transfer during the laser program runtime the following can be transmitted:

- Input words IW3 ... IW13.
- Input words IW14 and IW15 if system parameter **PLC_ExtendBaseIoFunction** = 0.
- Input words IW16 ... IW63, if TruControl 1000 version ≥ 1.84.0[1)].

A placeholder can be entered in the laser program for the transmission of parameters instead of a fixed value. It defines the input word starting from which the value has to be loaded during the runtime of a laser program.

These values can be changed during the program run. For example the laser power can be changed while the laser program is running.

The period between the time when a power value is enabled at the fieldbus interface and the moment it becomes effective at the workpiece lasts for approx. 5 ms. For OPC, the time of effectiveness depends on the system utilization.

The value of the data bits 15 ... 0 is defined from 15 (high) to 0 (low).

In the following tables, possible parameters and their resolution are listed, which can be transmitted to the control of the laser device via input words 3 ... 63.

For further information: see software manual "TruControl 1000", doc. no. 10-06-80-A1-CR.

1) Not available for the following CPX variants:

- CPX V1 standard, 18-21-85-A1
- CPX V1 small, 18-22-21-A1
- CPX V2 standard, 18-21-85-A3
- CPX V2 small, 18-21-85-A4

Parameter	Definition	Note
Laser power	1 W/Digit	For laser devices of the TruMicro series, the laser power is given in steps of 0.1 % of the rated power (see "Tab. 321", pg. 188).
Frequency	0.1 Hz/digit	The actual frequency at the set value 0 Hz and the behavior in case of a frequency change depend on the following criteria: ▪ For TruControl 1000 version ≥ 1.74.0 and the DLC versions starting from DLC_0822, DLCD1225, DLCD1610, DLCM1220 and DLCM1701 – A frequency change will become effective immediately after the end of the current laser pulse. The duration of the next pulse gap is already calculated using the new frequency. – If the frequency = 0 Hz is set, the actual frequency is 0 Hz. No laser pulse is emitted. For this reason, the laser can be triggered with the help of the frequency. ▪ For TruControl 1000 version < 1.74.0, or older DLC versions: – A frequency change will become effective after the pulse pause. – If the frequency = 0 Hz is set, the actual frequency is 0.1 Hz. A laser pulse is emitted every 10 seconds.
Pulse duration	1 µs/digit 10 µs/digit 100 µs/digit 1000 µs/digit	The resolution depends on the system parameter PLC_PulseDurationResolution. Presetting: 10 µs/digit
Number of pulses	Number	Number of laser pulses.
Start ramp	1 %/digit	For pulsed lasers only possible if pulse number > 1
End ramp	1 %/digit	
PK pulse number	1/digit	Number of pulses for monitoring of the pulse counter. PK pulse number is available with CW lasers if pulse counter check = "With monitoring" or = "With fault".
Pulse gap power	1 W/digit	Power in the pulse gap. Pulse gap power is available with CW lasers if pulse gap parameter = With power specification and pulse type = rectangle or pulse shape.

Parameters for all laser devices Tab. 310

Parameter	Definition	Note
Beam splitter position	1 increment/ digit	Position of the variable beam splitter. The beam splitter position does not equate to the splitting ratio. The position of the beam splitter must be determined by the user via the application. The position of the variable beam splitter is only changed if a line is changed in the laser program.

Parameters for motor-driven beam splitters (TM) Tab. 311

Parameter	Definition	Remark
Pulse power	1 W/Digit	Set value for the pulse power
Average power	1 W/Digit	Set value for the average power
Frequency	1 Hz/Digit	Set value for the frequency
PLS parameters	Parameters for PierceLine - PLS start - PLS number - Material thickness	
Further information: Technical Information "PFP-Control and PierceLine", Doc.-No. 21-A0-61-CP.		

Parameters for PFP Control and PierceLine Tab. 312

Parameter	Definition	Note
Monitoring	0.1 %/Digit	Limit value for monitoring message.
Errors	0.1 %/Digit	Limit value for fault message.

PCP parameters Tab. 313

Parameter	Definition	Remark
PFO Program number	Number and parameter	Bit 0 : Bit 9 If IPFO bit = 0: PFO program number, value range: 1 ... 999 dec. If IPFO bit = 1: IPFO job number, value range: 1 ... 32 dec. Bit 10 IPFO: If IPFO = 0, bits 0 - 9 = PFO program number. If IPFO = 1, bits 0 - 9 = IPFO job number. Bit 11 ExtEn: If ExtEn = 1 the job lines marked with "P" ("Active") and "P V.Aktiv" ("VarActive") will be executed with laser power. If ExtEn = 0 the job lines marked with "P" ("Active") and "P V.Aktiv" ("VarActive") will be executed without laser power. Bit 12 SeamLine: seam tracking Bit 13 RETRY: program mode selection If RETRY = 0, normal, complete execution of the PFO program. If RETRY = 1, execution starts at the time that the PFO program with the same number is aborted after being called up just before. Bit 14 ERROR_SV_LIMITER: If ERROR_SV_LIMITER = 1, a fault message is generated when the scanning volume is exceeded. If ERROR_SV_LIMITER = 0, a warning message is generated when the scanning volume is exceeded (factory setting). Bit 15 RTS: RealTimeStart (only for PFO programs) If RTS = 1, the PFO positions the scanner mirror to the start position and waits for the digital input "RealTimeStart". If RTS = 0, the PFO program is executed normally. The start is performed via the laser program.
PFO coordinate X	10 µm/Digit[1)]	With PFO 3D-2, ID=0x4E: 1 µm/Digit[1)].
PFO coordinate Y	10 µm/Digit[1)]	With PFO 3D-2, ID=0x4E: 1 µm/Digit[1)].
PFO coordinate Z	10 µm/Digit[1)]	With PFO 3D, ab PFO_3003. With PFO 3D-2, ID=0x4E: 1 µm/Digit[1)].
PFO coordinate for defocus	10 µm/Digit[1)]	With PFO 3D, ab PFO_3003. With PFO 3D-2, ID=0x4E: 1 µm/Digit[1)].
PFO Trans X	10 µm/Digit[1)]	With PFO 3D-2, ID=0x4E: 1 µm/Digit[1)].
PFO Trans Y	10 µm/Digit[1)]	With PFO 3D-2, ID=0x4E: 1 µm/Digit[1)].
PFO Trans Z	10 µm/Digit[1)]	With PFO 3D, ab PFO_3003. With PFO 3D-2, ID=0x4E: 1 µm/Digit[1)].

1) The resolution can be set in the *>Laser program editor >Laser program settings* menu with parameter *PFO ext. setting*.

Note
If parameter *PFO ext. setting* is set to 1 µm and the PFO works with a resolution of 10 µm, the set value in the laser control will be automatically converted.

Parameter	Definition	Remark
PFO Rotation[1]	The resolution depends on the system parameter **FO_PFO_TransRotationResolution**:	
	0.1°/digit	= 0: Angle in degrees. Value range: -180° ... +180°
	1 step = 0.0055°/digit[1]	= 1: Angle in steps[1] or 0.0055 °/Digit value range: 0 ... 65535 steps corresponds 0° ... 360°
	0.1°/digit or 1 step[1]	= 2: Selection in the laser program The rotation can be specified in degrees (0.1°/digit) or in increments[1] in the laser program for each line.
PFO rotation X[1]	For PFO 3D-2, ID=0x4E. Value ranges and resolution, see PFO Rotation.	
PFO rotation Y[1]		
PFO program runs	Quantity	Value range: 1 ... 65535 dec.
PFO scaling X	100 or -100	Only reflection (value 100 or -100) is allowed. PFO scaling X and Y parameters not available for a PFO 20-2 PO with command version ≥ 145.
PFO scaling Y		
PFO: scan width X	1 µm	For PFO 20-2 PO with command version ≥ 145 and TruControl version ≥ 2.78.0: width of the hardening track if laser program parameter "PFO: scan width" ≠ "No specification".
PFO: scan width Y		
Parameter for on the fly mode	If PFO parameter: on the fly = Simple Coordination: Simple linear motion by specifying the axis speeds.	
PFO: feed X	1 mm/s	If the PFO needs to be moved then the feed rates must be specified as positive. If the workpiece needs to be moved then the feed rates must be specified as negative. Value range: -1000 mm/s - 1000 mm/s
PFO: feed Y		
PFO: feed Z		
PFO: Start delay	0.1 ms	If it is not started directly at the processing point then the start can be delayed with the **Start delay** parameter. Value range: 0 ms - 6000 ms
Wobble program parameter[2]	If Wobble program - parameter specification = Yes is set in the laser program, the following parameters can be set externally via the Multi-PLC-interface:	
Wobble program		Number of the wobble program. Value range: 1 to 100.
Wobble width	1 µm/digit	Width of the wobble geometry. value range: 0.1 mm to 10 mm.
Length	1 µm/digit	Length of the wobble geometry. value range: 0.1 mm to 10 mm.
Frequency	1 Hz/Digit	Frequency of the wobble movement. Value range: ▪ Minimum value: 10 % of the set frequency in the wobble program, min. 1 Hz. ▪ Maximum value: set frequency in the wobble program, max. 1000 Hz.

1) If a PFO does not support the specification in increments and the system parameter **FO_PFO_TransRotationResolution** is > 0, the message 0000 E4AC is displayed during start-up of the system.
If an existing laser program has another resolution than programmed in the system setting, the selection is displayed as soon as the existing laser program is edited. This selection can no longer be subsequently changed in the laser program.
If the specification in increments is selected, the corresponding angle is calculated and displayed in the laser program.

2) Available from TC 3.36 on, if the PFO supports wobbling.
Which parameters are available depends on the wobble shape selected in the wobble program.
Further information: Software manual "TruControl 1000 for PFO 14, PFO 20, PFO 33, PFO 33 (KF023), PFO 3D, PFO 3D-2 PO, SFO", Dok.-Nr. 12-12-14-A2-CR

Parameter	Definition	Remark
PFO: fieldbus control		If in the laser program **PFO: fieldbus control** = is set to yes, new PFO program data sets are already provided during the active PFO program cycle via the selected input word **PFO: fieldbus control - Input signals** for the next PFO program cycle. The parameters **Store Values 1 and 2** signal that new PFO program data sets are available. If new data sets are made available, the values of the output signals **PFODataRdy 1 and 2** from the selected output word **PFO fieldbus control state** must be mirrored in the input signals **StoreValues 1 and 2** so that the new data sets are used by the laser control unit. The combination **PFO:fieldbus control** = Yes and **PFO: real-time start** = Yes is not permitted.
Bit 0	Store Values 1	
Bit 1	Store Values 2	

PFO parameters Tab. 314

Parameter	Definition	Remark
Focus position/lense position	10 µm/digit	Set value for the focus/lens position for programmable motorized focusing (PMF), with TruControl 1000 version < 2.34.0: CCM (**C**ontrol **C**ollimation **M**otor). **Notes** If FO_CollimationMotor system parameter: = 0, set value for the focus position = 1: set value for the lens position The actual value should be synchronized with the set value by the external control (e.g. PLC). The set value can only be specified by the assigned slot. The recipient of the set value is the PMF used in the laser program. While the PMF is moving to a new position, the emission of laser light is suppressed if the set values are permanently defined in the laser program. Further information: Operator's manual of the PMF, doc.-no. 22-50-01-A369-CR.
RM: Position	Set value and setting	Position and mode of the direction of rotation of the rotary modules (Rotary Module, RM) of the programmable motorized focusing (PMF), with TruControl 1000 version < 2.34.0: CCM (**C**ontrol **C**ollimation **M**otor). Bit 15: mode of the direction of rotation = 0 automatic smallest angle of rotation = 1 bit 14 is evaluated bit 14: direction of rotation, if bit 15 = 1 = 0 counterclockwise (positive) = 1 clockwise, in top view bits 8 ... 0: set value for the orientation, 0 ... 360 degrees absolute If the value is > 360 degrees, a fault message is output . **Notes** If an unassigned slot activates a laser program with rotary module, the information light path and RM position of the first line are used to move the rotary module. The actual value should be synchronized with the set value by the external control (e.g. PLC). Interface parameters **Byte swap** must be taken into account. Further information: Operator's manual of the PMF, doc.-no. 22-50-01-A369-CR. Operator's manual of the rotary module, doc-no. 22-50-01-A419-CR.

PMF parameters Tab. 315

Parameter	Definition	Remark
PCC →	Set value	Set value for the laser power, which is converted to laser power using the specified process characteristic curve.[1)]. Value range: 0 ... 32767
PCC #	Number	Number of the process characteristic curve which is used to convert the set value into laser power[1)].
1) Requirement: System parameter **ProcessCharacteristic** = 1 and the parameter *Pulse type* is set to *Process characteristic curve* in the laser program.		

Parameters for process characteristic curve (PCC) Tab. 316

Parameter	Definition	Note
Temperature	0.1 °C/digit	Process temperature during TPC mode = Temperature direct.

Parameters for TPC programs Tab. 317

Parameter	Remark
Parameters for the SeamLine Remote (SL) option for seam position control when welding with PFO, if system parameter **FO_PFO_ScanLine** = 1 and laser program parameter *PFO mode* = *SeamLine Basic*.	
SL: Feed rate	Feed rate in mm/s
SL: Configuration	SL configuration: SL_CONFIG_NO: Number of the SeamLine Remote configuration. Value range: 0 ... 65535.
SL: Seam number	SL seam number: SL_JOIN_NO
SL: Control	SL control: Bit 0: SL_WORKPIECE_START Welding process with seam position control is started.
SeamLine Remote additionally uses parameters **workpiece serial number** (see "Parameter transfer to the laser program settings", pg. 191)). Further information: operator's manual "SeamLine Remote (SE18)", doc no. 83-01-04-A0-CR.	

Parameters for SeamLine Remote Tab. 318

Parameter	Remark				
If an FCM (Focus Change Motor-driven) component is present in the light path, the laser beam is divided with the FCM and fed into a 2-in-1 LLK or 3-in-1-LLK. The division of the laser power between the core and the ring is set as follows:					
Processing	**Value**	**Power density distribution**	**Default in TruControl**	**Use**	**LLK**
	0	100 % core	Core	Arbitrary.	2-in-1. 3-in-1.
	1	100 % ring 1	Ring	Arbitrary.	2-in-1. 3-in-1.
	2	100 % ring 2	Ring 2	FusionLine.	3-in-1.
	3	**Power share core** parameter	BrightLine Weld	BrightLine Weld.	2-in-1.
Power share core	If the parameter **Processing** = *3* the power share of the core is set with this parameter. Value range: 10% ... 90%, increment: 1%, from TruControl 1000 Version 3.28: Value range 0% ... 100%.			BrightLine Weld.	2-in-1.
For further information: see software manual "TruControl 1000", doc. no. 10-06-80-A1-CR.					

Parameters for BrightLine Weld and FusionLine Tab. 319

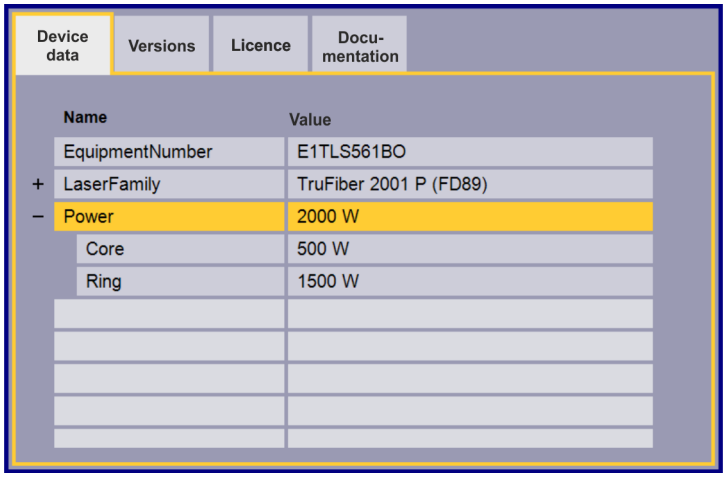
Parameter	Definition	Note
Cutting parameter set number	Number	Value range: 1 ... 200 dec.
Cutting speed	0.01 m/min/digit	
Max. cutting speed	0.01 m/min/digit	

Additional parameters for CW lasers and laser devices with CutAssist module Tab. 320

Parameter	Definition	Remark
Laser power	0.1 %/digit	Laser power in steps of 0.1 % of the rated power
Base frequency[1)]	1 Hz/digit 10 Hz/digit 100 Hz/digit 1000 Hz/digit	The resolution for the base frequency can be set using the system parameter PLC_BaseFrequencyResolution [1)]. Presettings: ▪ TruMicro 8000: 10 Hz/digit ▪ TruMicro 7000: 10 Hz/digit ▪ TruMicro 6000: 1000 Hz/digit ▪ TruMicro 5000: 1000 Hz/digit ▪ TruMicro 2000: 1000 Hz/digit
Beam splitter[1)]	1/digit	Valid values: ≥ 1[1)].
Work frequency [1)]	kHz	Set value of the work frequency between 0 Hz and the base frequency[1)]. The base frequency is used if work frequency = max. The base frequency depends on the set operating point.
Burst pattern		When Burst Mode option is activated, the burst pattern can be specified with this parameter. The highest-value bit must always be 1, i. e. Burst pattern ≥ 0x8000 hex...
Short pulse duration	1 fs/digit	Short pulse duration when the "Pulse duration flexible" option is activated.
Short pulse count	1 pulse/digit	Number of ultra-short pulses for laser devices in the family 20-20-xx-xx (FU11).
Energy level	µJ	With laser devices of the 20-20-xx-xx (FU11) family, the energy level determines the energy of a single short pulse with laser power = 100%**).
1) Depending on the laser device, the following parameters are available: ▪ Base frequency and splitter Work frequency = base frequency/beam splitter ▪ Work frequency		
Further information: See the operator's manual of the laser device.		

Additional parameters for laser devices of the TruMicro series Tab. 321

Parameter	Definition	Remark
Pulse modulation		
Pulse duration	1 µs/Digit	System parameter PulseModulationDuration_Step
Frequency	10 Hz/digit	System parameter PulseModulationRate_Step
VariMode[1)]	0 or 1	Switching on VariMode If the VariMode option[1)] is available, this option is switched on with Bit 0 = 1. If VariMode is switched on, the fiber is squeezed to change the beam properties. The laser is switched off during the switch-over of VariMode. If a program line is active, VariMode cannot be switched over. The switch-over takes place during a line change.
Laser power[1)]	Distribution of laser power in LLK core and LLK Ring if the BrightLine Mode option[1)] is used.	
Power core	1 W/digit	Power in LLK core
Power ring	1 W/digit	Power in the LLK ring
	The maximum possible set values for the power in the core and ring can be found under *>Help >Device data >Power* from TC 3.40 on.	



1) Available, e.g. for laser devices in the TruFiber P series (FD89), laser device family 20-14-01-xx.

Additional parameters for laser devices of the TruFiber series Tab. 322

Parameter	Remark					
Seam ID						
Start address	Start address for the seam ID.					
Number of words	Number of input words for the seam ID. The seam ID must be given in the ASCII format. If the seam ID is shorter than the specified number of words, the rest of the input words must each be filled with "0x0000". The central processor CPX transmits the seam ID to the focusing optics and VisionLine OCT Check. Example: seam ID: "**SeamID123**", Start address: **IW14**, number of words: **6**					
	Input word	**Value, hex**	**Character**	**Input word**	**Value, hex**	**Character**
	IW14	0x5365	Se	IW17	0x3132	12
	IW15	0x616D	on	IW18	0x3300	3
	IW16	0x4944	ID	IW19	0x0000	
workpiece series number	Description (see "Parameter transfer to the laser program settings", pg. 191)					
Workpiece ID						

Traceability data Tab. 323

Examples

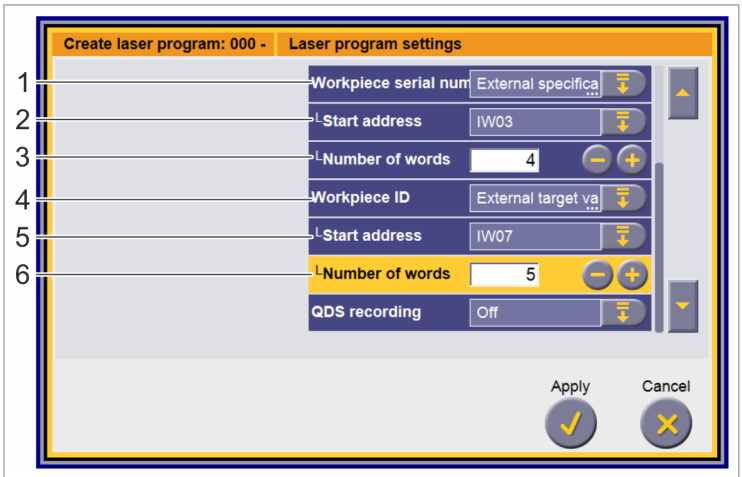
Parameter	Value	Resolution:	Data word, hexadecimal
Laser power	600 W	1 W/digit	0258
Pulse duration	5,000 µs	10 µs/digit	01F4
Frequency	45.0 Hz	0.1 Hz/digit	01C2
Start ramp	30 %	1 %/digit	001E
PFO program number	550	1/digit	0226
PFO rotation	270.5°	0.1°/digit	0A91
PFO coordinate X	5.00 mm	10 µm/digit	01F4
Cutting parameter set number	150	1/digit	0096
Cutting speed	3 m/min	10 mm/min	012C
Focal position/lens position (PMF)	1.24 mm -1.24 mm	10 µm/digit	007C FF84
Wobble program			
Wobble program	22	1/digit	0016
Wobble width	5.0 mm	1 µm/digit	1388
Length	1.0 mm	1 µm/digit	03E8
Frequency	200 Hz	1 Hz/Digit	00C8

Tab. 324

Parameter transfer to the laser program settings

The following parameters can be transmitted in the laser program settings:

- Workpiece serial number
- Workpiece ID



1	External specification of the workpiece number	4	External specification of the workpiece ID
2	Start address	5	Start address
3	Number of input words	6	Number of input words

Configuring specification of the workpiece serial number and ID in the laser program settings

Fig. TE-00133

The workpieces serial number and ID are used for different applications. Examples:

- For using focusing optics with TPC sensor, e.g. **PFO 20-2 PO**.

  Further information: Operator's manual "Programmable focusing optics PFO 20-2 PO", doc no. 12-12-21-A5-CR.
- When using focusing optics with SeamLine Remote, e.g. PFO with SeamLine Remote.

  Further information: Operator's manual "SeamLine Remote (SE18)", doc no. 83-01-04-A0-CR.
- When using a focusing optics with VisionLine OCT Check, e.g. PFO with OCT sensor.

  Further information: Software manual "VisionLine OCT Check", doc. no. 83-01-07-A0-CR.

The central processor CPX transfers the workpiece serial number and ID to the focusing optics and to VisionLine OCT Check. The quality data of the focusing optics is recorded on a QDS PC. The QDS PC receives the workpiece serial number and ID for the corresponding quality data about the OPC from the central processor CPX.
The **workpiece serial number** (1) can be set externally. To do so, the start address (2), the serial number and the number of the input words (3) are entered in the laser program settings.

The workpiece serial number must be specified in the ASCII format. If the workpiece serial number is shorter than the specified number of words, the rest of the input words must each be filled with *0x0000*.
Example:
Workpiece serial number: "WPSN12"
Start address: IW06
Number of words: 4

Input word	Value, hex	Character
IW06	0x5750	WP
IW07	0x534E	SN
IW08	0x3132	12
IW09	0x0000	

Workpiece serial number, example Tab. 325

The **workpiece ID** (4) can be specified externally as well. To do so, the start address (5) and the number of input words (6) is entered.
The workpiece ID must be specified in the ASCII format. If the workpiece ID is shorter than the specified number of words, the rest of the input words must each be filled with *0x0000*.
Example:
Workpiece serial number: "WPID90"
Start address: IW10
Number of words: 4

Input word	Value, hex	Character
IW10	0x5750	WP
IW11	0x4944	ID
IW12	0x3930	90
IW13	0x0000	

Workpiece ID, example Tab. 326

The laser program settings are called up in the laser program editor with the *Change properties* button.
Further information: Software manual "TruControl 1000", Dok.-Nr. 10-06-80-A1-CR.

Other input parameters

If TruControl 1000 version ≥ 1.88.0 in the dialog >*Setup* >*Interface parameters*, Tab >*Inputs* the following input words can be assigned freely:

- Input words 3 ... 13.
- Input words 14 and 15 if system parameter **PLC_ExtendBaseloFunction** = 0.
- Input words 16 ... 63[1)].

1) Not available for the following CPX variants:

- CPX V1 standard, 18-21-85-A1
- CPX V1 small, 18-22-21-A1
- CPX V2 standard, 18-21-85-A3
- CPX V2 small, 18-21-85-A4

Further information: software manual TruControl 1000, doc. no: 10-06-80-A1-CR.

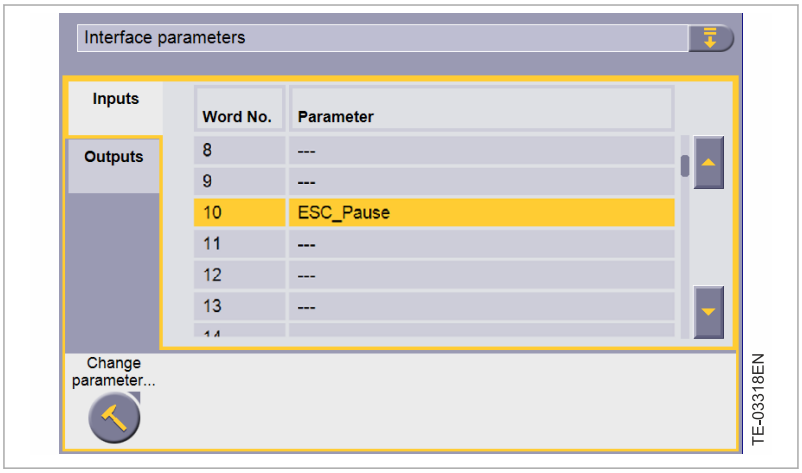


Fig. TE-03318

	If you want to assign another input parameter to an input word, select *Change parameter*.

Tab. 327

If for the input words 2 .. 63 the parameter **Default value** is selected, then the input words 2 ... 63 can be used for laser programs.

The following table contains the description of the other input parameters.

Parameter	Description
PLC input (CV) x words[1)]	Parameter for the image processing system VisionLine, x = 6, 8 or 16 input words.
Pause (ESC)	Parameter "Pause (ESC)" contains the signal ESC_PAUSE. The ESC_PAUSE signal is used to put the laser device in an energy-saving state or to exit this state (see "ESC_PAUSE", pg. 194).
Line Laser (PFO)	Switches on line laser of a PFO. The input word is bit-coded with the following assignment: Bit 0: PFO 1 ... Bit 7: PFO 8.
FCU/PCU lid state	Parameter for opening or closing the lid of an FCU or PCU. High byte: numbers of the CLI with FCU. Low byte: numbers of the CLI with PCU. The number is specified according to the NUM scheme. Example: In order to open the PCU lid of the CLI with NUM = 0x04, the sequence 0x0004 must be transferred in the input word. All other lids of the available FCU or PCU are closed if these were open.
PLC input (VLCK) 13 words[2)]	Parameter for option VisionLine Check with focusing optics BEO and OCT sensor: ▪ Word 1: VLCK-Control: – Bit 0: PreTrigger – Bit 1: InTrigger ▪ Word 2,3: X-position, 4 bytes ▪ Word 4,5: Y-position, 4 bytes ▪ Word 6,7: Z-position, 4 bytes ▪ Word 8,9: A-rotation, 4 bytes ▪ Word 10,11: B-rotation, 4 bytes ▪ Wort 12,13: C-rotation, 4 bytes
Sensor Relay Input x Words	Input data of the sensor relay, x = 9, 16, 32 or 48 (see "Configuring the sensor relay", pg. 266)
1) Parameter PLC input (CV) x words are not available to the VisionLine interface. 2) Available from TruControl 1000 version 3.46 on, further information: Software manual "VisionLine OCT Check", Doc.-no. 83-01-07-A0-CR.	
The availability of these parameters depends on the configuration of the laser device.	

Tab. 328

ESC_PAUSE

Functions	Input signal **ESC_PAUSE**.
Active signal	▪ If ESC_PAUSE > 0 and ESC_PAUSE < 65535 (= 0xFFFF): The external control sends "Start pause" with a specification of the pause time in seconds. The corresponding Energy Save Level is switched to depending on the pause time. ▪ If ESC_PAUSE = 65535 (= 0xFFFF): The external control sends "End Pause" to end the pause. The laser control changes to the status "Ready_To_Operate". ▪ If ESC_PAUSE = 0: The external control acknowledges the receipt of the output signals. The laser control status does not change.
Information	Technical information "Energy Save Concept", doc.-no. 21-A0-71-CP.

Tab. 329

Signals of the alternative input words 14 and 15

ENERGY_SAVE_ON

Functions	Switches on the energy-saving mode (Sleep mode 3) of the laser control.
Active signal	By applying the positive edge of the signal, the external control unit requests the laser control to switch the laser device to the energy-saving mode.
Requirement	This input signal is only available if the following conditions have been met: ▪ The laser control is equipped with a central processor: CPX V3 or CPX V4. ▪ System parameter **PLC_ExtendBaseIoFunction** = 1. ▪ System parameter **LCU_EnergySaveMode** > 0. ▪ The input **24 V_STDBY** at X1 of the PBT board is supplied with voltage. If the system parameter **ESC_Enable** is active, the laser devices switch between Energy Save Level and Ready_To_Operate with the signals ESC_Pause > 0 and ESC_Pause = 0 . In this case, you should not use the signal ENERGY_SAVE_ON.
Information	The laser device must not be assigned to the external control. The laser device and all laser control components except for the CPX and PBT are switched off. Optimized standby mode is ended with the signal **ENERGY_SAVE_OFF**. For further information: see technical information "Energy Save Concept", doc. no. 21-A0-71-CP. Not available for laser devices with CPX V20.
Feedback	The following output words indicate that the energy-saving mode has been switched on: ▪ LASER_CONTROL_IS_ON = FALSE ▪ ENERGY_SAVE_IS_ON = TRUE

Tab. 330

ENERGY_SAVE_OFF

Functions	Switches off the energy-saving mode.
Active signal	The external control unit requests the laser control to switch itself on and to terminate the energy-saving mode.
Requirement	This input signal is only available if the following conditions have been met: ▪ The laser control is equipped with a central processor: CPX V3 or CPX V4. ▪ System parameter **PLC_ExtendBaseIoFunction** = 1. ▪ System parameter **LCU_EnergySaveMode** > 0. ▪ The input **24 V_STDBY** at X1 of the PBT board is supplied with voltage. The main switch is set to "On". If the system parameter **ESC_Enable** is active, the laser devices switch between Energy Save Level and Ready_To_Operate with the signals ESC_Pause > 0 and ESC_Pause = 0. In this case, you should not use the signal ENERGY_SAVE_OFF.
Information	The laser device does not have to be assigned to the external control unit. After switch-on, the laser control performs a device reset. For further information: see technical information "Energy Save Concept", doc. no. 21-A0-71-CP. Not available for laser devices with CPX V20.
Feedback	The following output signals indicate that the laser control has been switched on and is not in the optimized standby mode: ▪ LASER_CONTROL_IS_ON = TRUE ▪ ENERGY_SAVE_IS_ON = FALSE

Tab. 331

LASER_CONTROL_OFF

Function	Switch off laser device. OSB 4 is activated.
Active signal	The external control requests the laser control to switch off the laser device.
Prerequisite	This input signal is only available if the system parameter **PLC_ExtendBaseIoFunction** = 1. The laser is assigned to the external control; the output signal **LASER_ASSIGNED** is active. The laser control is switched on; the output signal **LASER_CONTROL_IS_ON** is active.
Feedback	All output signals will be inactive.

Tab. 332

MACHINE_COOLING_ON

Functions	Switch on external cooling.
Active signal	The external control requests the laser control to switch on the external cooling.
Requirement	This input signal is only available if the system parameter **PLC_ExtendBaseIoFunction** = 1.
Information	This input signal is required, for example, if a TRUMPF laser machine and the integrated laser device use the cooling of the laser machine jointly. The signal is OR-linked via all MPI slots. This means that as soon as an external control (e.g. PLC) has set the signal, the external cooling will be switched on.
Feedback	The output signal **MACHINE_COOLING_READY** becomes active.

Tab. 333

POWER_GATE

Functions	**POWER_GATE** via Multi PLC interface (e.g. fieldbus or OPC)
Active signal	The laser power is suppressed if this input is active.
Requirement	This input signal is only available if the system parameter **PLC_ExtendBaseloFunction** = 1.
Information	The system parameter RTI_PULSE_INTERLOCK_MODUS must be set in the dialog >*Setup* >*System setting* >*Real-time interface* via fieldbus to the following value: 5 = POWER_GATE via fieldbus. **RTI_PULSE_INTERLOCK_MODUS** = 1 is set ex works. Further information: ▪ Digital I/O interface of the CPX – Parallel I/O, Real time, sync I/O and Decentralized I/O interface of the CPX V3 and CPX V4 standard, Doc. no. 18-22-91-A2-BA. ▪ Real-time and Sync I/O interface of the CPX V4 Basic, Doc.-No. 18-23-04-A1-BA. ▪ Parallel I/O, Real-time and Sync I/O interface of the CPX, Doc. no.: 18-21-85-A1-BA (for CPX V1 and V2). The signal **POWER_GATE** reacts as follows: ▪ If the system parameter RTI_LEVEL_SETTING = 0: – Bit not set: Laser power is off. – Bit set: Laser power is on. ▪ If the system parameter RTI_LEVEL_SETTING = 1: – Bit not set: Laser power is on. – Bit set: Laser power is off. If the system parameter **RTI_POWER_GATE_FAST** = TRUE, signal changes up to 5 kHz are possible. For this option, no PLCView data is recorded for the RTI. Requirement for signal changes up to 5 kHz: System parameter **RTI_PULSE_INTERLOCK_MODUS** = 4, POWER_GATE mode or = 5, POWER_GATE mode via fieldbus. If the laser power is to be triggered with a frequency > 1 kHz, system parameter **FG_TriggerOnSetValue** = 3, automatic pulse mode, must be used. Here, there is automatic switching between pulse mode and modulation mode. Further information: (see "Setting interface parameters", pg. 263). If the central processor CPX V20 is used, e.g. with a laser device from the TruMicro 2000 series (FU11), laser device family 20-20-xx-xx, no real-time interfaces are available.

Tab. 334

Signals of output word 0

EXT_ACTIVATION_ACTIVE

Function	Acknowledgement signal of the **EXT_ACTIVATION** input signal.
Active signal	The laser device can be controlled via the Multi PLC interface (fieldbus, OPC or parallel I/O interface).
Information	As long as the **LOCK_OPERATING_PANEL** input signal is not active, laser parameters and laser programs can continue to be modified on the operating panel (OP).

Tab. 335

FAULT_LASER

Functions	The output signal indicates a fault of the laser device.
Active signal	The laser device control activates the **FAULT_LASER** output signal when a fault limit value for electrical or optical components has been exceeded.
Information	The displayed message indicates possible causes of the fault. By activating the **RESET** input signal: ▪ the displayed message is reset. ▪ the output signals **MONITORING** and **FAULT_LASER** are set to inactive. ▪ the control unit of the laser device is initialized again. **Note:** If a fault occurs repeatedly, the cause of the fault must be eliminated first. The message behavior and the RESET signal are influenced by different system parameters. Further information: (see "Messaging and RESET behavior", pg. 258).

Tab. 336

LASER_ASSIGNED

Functions	Acknowledgement signal of the **REQUEST_LASER** input signal.
Active signal	The laser device is assigned to the requesting external control unit. The external control unit receives in the output words status data from the laser.
Assignment process	The assignment of the laser is carried out according to priority and in case of the same priority according to the round robin algorithm. A priority parameter is assigned to every MPI slot. The value range of the **Priority** parameter is from 0 to 255. The higher the value, the higher the priority of the MPI slot. Further information: (see "Configuring interfaces", pg. 263). If several MPI slots with different priorities have activated the REQUEST_LASER signal, the MPI slot with the highest priority is assigned. If several MPI slots with the same priority have activated the REQUEST_LASER signal, the assignment is performed in accordance with the round robin algorithm: in the numerical order of the MPI slots. The assignment process starts at MPI slot 1. As soon as the assignment has finished processing the MPI slots used, it starts again at MPI slot 1. Example for the round robin algorithm: The laser is assigned to the external control unit at the MPI slot 1. The MPI slots 1, 2, 3 and 5 have the same priority value. The external control units at the MPI slots 2, 5 and 3 request the laser one after another. As soon as the external control unit releases the laser at MPI slot 1, the external control unit receives the assignment at MPI slot 2. If this control unit releases the laser, the external control unit at MPI slot 3 is processed and not the external control unit at MPI slot 5. The external control unit at MPI slot 5 receives the assignment as soon as the external control unit at MPI slot 3 releases the laser. If the external control unit at MPI slot 1 sends a request to the laser in the meantime, this request is processed only after release of the external control unit at MPI slot 5.
Information	When the **LASER_READY** output signal is also active, the external control unit can start laser programs. An external control unit can start a laser program of type "No pulse triggering", even if the laser device is not assigned to the external control unit. The output mask can be used to define what status signals of the laser are received by the external control even if the **LASER_ASSIGNED** signal is not active. Further information: (see "Output words", pg. 154).

Tab. 337

LASER_IS_ON

Functions	Acknowledgement signal of the **LASER_ON** input signal.
Active signal	The laser is on.
Information	The output signal remains active as long as the laser is switched on. If there is a fault with the laser device, the laser device is automatically switched off. **Note** The output signal remains inactive as long as the output signal **CALIBRATION/SELF_TEST** is active. If the **LASER_READY** output signal is also active, laser programs can be started. **Laser device with laser standby mode:** The laser device can be switched to the laser standby mode by activating the **LASER_STANDBY** input signal. **Energy Save Concept** or **Sleep mode** If the laser control unit activates an Energy Save Level or Sleep mode, the output signals **LASER_READY** and **LASER_IS_ON** are deactivated. For further information: see technical information "Energy Save Concept", doc. no. 21-A0-71-CP.

Tab. 338

LASER_READY

Functions	Output signal shows whether laser programs can be started.
Active signal	The laser programs can be started.
Information	After the start of a laser program, the signal becomes inactive until the laser program has been ended and a new laser program can be started. The **LASER_READY** output signal becomes inactive when the safety circuit is opened because the laser device, for reasons of safety, becomes disabled to give out laser power. If an attempt is made to start a laser program while the **LASER_READY** output signal is inactive: ■ a monitoring message is displayed. ■ the output signal **MONITORING** becomes active. ■ the laser program is not started. If a PCU is available, the output signal **LASER_READY** is only activated if the PCU is also ready. An external control can start a laser program of the type "No pulse release", even if the laser device is not assigned to the external control. The **LASER_READY** signal becomes active as soon as the program number of a "No pulse release" type laser program is applied. **Laser with optical components:** If the program number of another laser program is given after the execution of a laser program, an already active **LASER_READY** output signal is set to inactive again (approx. 35 ms ... 40 ms) until the laser control has set the optical components to the light path specified in the laser program. The output signal **LASER_READY** can only be evaluated once the new program number and light path number (see "LIGHT_PATH_No: Bit 7 ... Bit 0", pg. 208) have been reported to the interface. **TruMicro 5000:** If, after execution of a laser program, the program number of another laser is given, an already active **LASER_READY** output signal is again changed to inactive until the new basic frequency has been set, which may take up to 30 sec. **Laser device with laser standby mode:** The **LASER_READY** output signal becomes active approx. 2 ... 5 s active after the laser has been switched into laser standby mode. **Energy Save Concept** or **Sleep mode** If the laser control unit activates an Energy Save Level or Sleep mode, the output signals **LASER_READY** and **LASER_IS_ON** are deactivated. For further information: see technical information "Energy Save Concept", doc. no. 21-A0-71-CP. If system parameter **CV_Beam_Tracker** = 1: After switching on the laser, **LASER_READY** is not set until the beam position check is complete.

Tab. 339

LASER_WARNING_LAMP_IS_ON

Functions	The output signal shows whether the laser warning light is switched on.
Active signal	The control unit activates the **LASER_WARNING_LAMP_IS_ON** output signal when the laser: ▪ is ready to generate laser power. ▪ is generating laser power.
Information	**Note:** The output signal becomes active when the voltage of a lamp or diode power supply exceeds 40 V and stays active until the voltage drops below 40 V.

Tab. 340

MONITORING

Functions	The output signal indicates when a monitoring message was emitted during operation.
Active signal	The **MONITORING** output signal becomes active when the control unit of the laser device has detected a condition that: ▪ can lead to switching off the laser device. ▪ has led to switching off the laser device.
Information	If, in connection with the output signal **MONITORING**, the output signals **PROG_CANCELLED**, **SIK_OPEN** and **FAULT_LASER** are also evaluated, the causes for a monitoring can be determined more exactly. **MONITORING active:** A condition was detected by the laser device control that has not yet led to the switching off of the laser device. The displayed message gives hints about the possible causes (e.g., the conductivity of the DI water has exceeded the monitoring limit value). **MONITORING + SIK_OPEN active:** A monitoring message was emitted because a safety circuit has been interrupted (see "SIK_OPEN", pg. 207). **MONITORING + FAULT_LASER active:** The laser device was switched off due to a malfunction of the laser device (see "FAULT_LASER", pg. 199). **MONITORING + PROG_CANCELLED active:** A running laser program has been cancelled, for example using the input signal **PROG_STOP** or **FAULT_EXT** (see "PROG_CANCELLED", pg. 205). The message behavior and the RESET signal are influenced by different system parameters. Further information: (see "Messaging and RESET behavior", pg. 258).

Tab. 341

PILOT_LASER_IS_ON

Function	Acknowledgement signal of the **PILOT_LASER_ON** input signal.
Active signal	The **PILOT_LASER_IS_ON** output signal becomes active when a pilot laser is switched on.
Information	**Notes:** Pilot laser in shutters and shutter switches are controlled by the ASV3 boards of the laser control.

Tab. 342

POWER_LIMIT

Functions	The output signal indicates when the power limit of the laser device has been reached (only with pulsed lasers).
Active signal	The power limit of the laser device has been reached. The following parameters influence reaching the power limit of the laser device: ▪ Pulse power ▪ Pulse duration ▪ Pulse repetition frequency
Information	When the power limit is reached: ▪ a monitoring message is displayed that must be acknowledged by means of RESET before work with the laser device can be continued. ▪ the power limit of the laser device can be met by adapting the parameters pulse power, pulse duration and pulse repetition frequency.

Tab. 343

PROG_ACTIVE

Functions	Acknowledgement of the input signals **PSTART_STATICAL** or **PSTART_DYN**.
Active signal	A laser program has been started and is being executed.
Information	The output signal **PROG_ACTIVE** remains active as long as the laser program is run.

Tab. 344

PROG_CANCELLED

Functions	The output signal indicates that the laser program was aborted.
Active signal	The output signal **PROG_CANCELLED** becomes active if a laser program was not completely executed. The following actions proceed: ▪ a monitoring message is displayed. ▪ The output signal **MONITORING** becomes active. **Note:** Laser programs can only be restarted if the message was reset by activating the input signal **RESET**.
Information	Laser programs are aborted under the following conditions: ▪ Input signal **PSTART_STATICAL** became inactive before a statically started laser program elapsed, if **Pulse type** ≠ "Continuous Wave" (for CW lasers) and **Pulse number** ≠ ∞ ("Continuous") are set. ▪ Input signal PROG_STOP was activated during the sequence of a laser program, if **Pulse type** ≠ "Continuous Wave" (for CW lasers) and **Pulse number** ≠ ∞ ("Continuous") are set. ▪ Laser control has detected a malfunction. ▪ Input signal **FAULT_EXT** was activated. ▪ Safety or EMERGENCY STOP circuit was interrupted during the running of a laser program. The output signal **PROG_CANCELLED** remains after an assignment change at the affected MPI slot. The output signal **PROG_CANCELLED** must be reset at the affected MPI slot, independent of the assignment. If there is a change in assignment, the output signal **LASER_ASSIGNED** changes from one MPI slot to another. If the output signal **PROG_CANCELLED** is set in the output mask, the output signal is output at this MPI slot even if this MPI slot is not assigned. This makes sense when, for example, a superordinate system control reacts to this signal. Further information: (see "Output words", pg. 154).

Tab. 345

PROG_COMPLETED

Functions	The output signal indicates the end of a laser program.
Active signal	The **PROG_COMPLETED** output signal is: ▪ active if a laser program has been executed completely. ▪ **not** active when a laser program was canceled.
Information	The timing of the output signal depends on whether a laser program has been started statically with the input signal **PSTART_STATICAL** or dynamically with the input signal **PSTART_DYN**. **Program start static:** ▪ Infinite laser program (Pulse type ≠ "Continuous wave" (for CW lasers) and Pulse number ≠ ∞ ("Continuous")): After the end of a laser program, the **PROG_COMPLETED** output signal becomes active and remains active until the **PSTART_STATICAL** input signal is switched to inactive. ▪ Endless laser program (Pulse type = "Continuous wave" (for CW lasers) Pulse number = ∞ ("Continuous")): If the laser program is finished by setting the **PSTART_STATICAL** input signal inactive, the **PROG_COMPLETED** output signal will be activated for approx. 6 seconds[1)]. When **PROG_COMPLETED** becomes active, current values of the parameters **PLC output (CV) x words** are available. **Program start, dynamically:** After the end of a laser program, the **PROG_COMPLETED** output signal becomes active for approx. 6 s[1)].
1) The **PROG_COMPLETED** output signal becomes prematurely inactive if, before the 6 s have expired, ▪ a laser program is started again (PSTART_STATICAL or PSTART_DYN). ▪ a change in assignment takes place, and the **PLC_ClearProgCompletedOnSlotChange** system parameter is 1.	

Tab. 346

SET_1 ... SET_3

Functions	The 3 output signals **SET_1 ... SET_3** can be used to synchronize an external control unit (e.g. PLC) with the sequence of a laser program.
Active signal	The output signals can be set to active (S1H, S2H, S3H) or inactive (S1L, S2L, S3L) by commands in the program lines of a laser program.
Information	Output signals **SET_1 ... SET_3** activated by a laser program remain active and are only set to inactive if: ▪ a laser program is started again. ▪ a cancelled laser program is reset with the **RESET** input signal. ▪ they are set to inactive by commands (S1L, S2L, S3L) within a laser program. **Note:** The output signals **SET_1 ... SET_3** can also be used to monitor the sequence of a laser program externally.

Tab. 347

SIK_OPEN

Function	The output signal indicates that a safety circuit of the laser device is interrupted.
Active signal	The **SIK_OPEN** output signal becomes active, when the safety circuit is open: ▪ of the central shutter (VZ). ▪ of the currently selected light path.
Information	When the **SIK_OPEN** output signal is active, no laser program can be started. **Determining the safety circuit status** The **SIK_OPEN** output signal is updated continuously for all external controls (slots), even if they have not bee assigned. There are two different cases: ▪ The light path is preset in the laser program. ▪ The light path is predefined externally. **The light path is permanently defined in the laser program** In this case, the SIK_OPEN output signal is determined for all light paths that occur in the laser program. **SIK_OPEN = "0",** if the safety circuits of all light paths given in the laser program are closed. **SIK_OPEN = "1",** if a safety circuit is open that is included in the laser program. **The light path is predefined externally** For this case, the SIK_OPEN output signal is only determined for the externally predefined light path. **SIK_OPEN = "0",** if the safety circuit of the externally predefined light path is closed. **SIK_OPEN = "1",** if the safety circuit of the externally predefined light path is open.

Tab. 348

Signals of output word 1

LIGHT_PATH_No: Bit 7 ... Bit 0

Functions	The 8 output signals **LIGHT_PATH_No: bit 7 ... bit 0** are used by the laser to output the light path number which is set for the emission of laser light.
Coding of the light path no.	The value of the 8 output signals is defined from **LIGHT_PATH_No: bit 7** (high) to **LIGHT_PATH_No: bit 0** (low). Format: Hexadecimal. Number of selectable light paths: max. 255 dec = FF hex
Information	Depending on the output number of the light path, an external control unit can: ▪ check whether the set light path is correct. ▪ keep a record of the light path used. **Note:** If an external control is connected to the laser device via a fieldbus-coded TRUMPF Laser-Network (TLN-fc), the light path number set in the dialog *>Setup >Multi PLC interface (MPI)* will be transmitted.

Tab. 349

PROGRAM_No: Bit 7 ... Bit 0

Function	Acknowledgement of the program number of the laser program set by means of the output signals **PROGRAM_No: bit 7 ... bit 0**.
Coding of the program no.	The significance of the 8 output signals is defined from **PROGRAM_No: bit 7** (high) to **PROGRAM_No: bit 0** (low). Format: Hexadecimal. Permitted program numbers: 0 ... 200 dec (0 ... C8 hex) Program numbers for special functions: 240 ... 250 dec (F0 ... FA hex) Program number 0 is only available if system parameter **min. program no.** = 0.
Information	Depending on the output number of the program, an external control unit can: ▪ check whether the right laser program has been selected. ▪ keep a record of the laser program used.

Tab. 350

Signals of the output word 2

APPLICATION_VALUES_IN_RANGE

Functions	Status signal indicating that the application values crossjet pressure, exhaustion and shielding gas are in the admissible range.
Active signal	The output signal **APPLICATION_VALUES_IN_RANGE** is active when the application values for crossjet pressure, exhaust system and shielding gas are within the admissible range (green zone). **Notes:** The output signal becomes inactive if the input signals **AD_No: Bit 7 ... Bit 0** of the input word 2 send the following values: ▪ Application data set no. "0" was sent: the monitoring of the focusing optics was switched off. ▪ An invalid application data set no. was sent.
Information	In the *>Programming >CFO data* or *>BEO app.data* dialog it is possible to: ▪ set the limit values for crossjet pressure, exhaustion and shielding gas. ▪ activate and deactivate monitoring of crossjet pressure, exhaustion and shielding gas. Before starting a program, you can check if the application values are within the admissible range and the focusing optics is ready for operation. For this purpose, a valid application data set no. and a light path no. must be selected 50 ms before starting a program. If a valid application data set no. has been defined, the control unit of the laser device needs max. 50 ms until the output signals **APPLICATION_VALUES_IN_RANGE** and **FOCUSING_OPTICS_READY** are valid.

Tab. 351

PROTECTIVE_GLASS_LIFETIME_REACHED

Functions	Status signal for limit value MAX2 of the protective glass contamination.
Active signal	The output signal **PROTECTIVE_GLASS_LIFETIME_REACHED** becomes active when the limit value MAX2 (orange zone) of the protective glass contamination has been reached.
Information	In the *>Programming >CFO data* or *>BEO app.data* dialog it is possible to: ▪ set the limit value MAX2 of the protective glass contamination. ▪ to switch the monitoring of the protective glass contamination on and off.

Tab. 352

PROTECTIVE_GLASS_CONTAMINATED

Functions	Status signal for limit value MAX1 of the protective glass contamination.
Active signal	The output signal **PROTECTIVE_GLASS_CONTAMINATED** becomes active when the limit value MAX1 (yellow range) of the protective glass contamination has been reached.
Information	In the *>Programming >CFO data* or *>BEO app.data* dialog it is possible to: ▪ set the limit value MAX1 of the protective glass contamination. ▪ to switch the monitoring of the protective glass contamination on and off.

Tab. 353

FAULT_SLOT

Functions	The status signal indicates a local fault of this slot.
Active signal	The output signal **FAULT_SLOT** is active if a local fault occurs for this slot.
Information	The external control unit connected to the slot must reset a local fault via the input signal **RESET**. A RESET performed by the external control connected to the slot has the following effect: ▪ The local message is acknowledged if the laser device of the processing station has **not** been assigned. ▪ The local message and a monitoring or fault message are acknowledged if the laser device has been assigned to the external control unit. Possible causes of a local fault: ▪ A fault limit of the focusing optics at the light path selected has been reached during the run of a laser program. ▪ The attempt was made to start a laser program that does not exist. **Notes:**: ▪ In case of a local fault, the output signal **FAULT_LASER** is not activated automatically. ▪ Check the messages displayed on the operating panel for further details on local monitoring. ▪ With TLN-fc, one slot corresponds to the assigned light path. The message behavior and the RESET signal are influenced by different system parameters. Further information: (see "Messaging and RESET behavior", pg. 258).

Tab. 354

FOCUSING_OPTICS_AVAILABLE/ LIGHTPATH_COMP_AVAILABLE

Functions	Status signal for focusing optics or light path component registered in the light path.
Active signal	The output signal **FOCUSING_OPTICS_AVAILABLE** is active if focusing optics or a light path component is registered in the selected light path.
Information	Focusing optics or light path components can be activated or deactivated in the dialog >*Technology* >*Light path*. Further information : Software manual "TruControl 1000", doc.-no. 10-06-80-A1-CR. **Note:** Only if the output signal **FOCUSING_OPTICS_AVAILABLE** is active, are the signals of the bits 15 ... 12 of output word 2 active. If a programmable motor-driven focus setting (PMF) is present in the light path, then the signal can only become active once the PMF is enabled[1)].

Tab. 355

FOCUSING_OPTICS_READY/LIGHTPATH_COMP_READY

Functions	Status signal: The focusing optics or light path is ready for operation.
Active signal	The output signal **FOCUSING_OPTICS_READY** is active if the focusing optics or light path component in the light path is ready for operation.
Information	Example: Focusing optics CFO is ready for operation if: ▪ **no** value for protective glass contamination, crossjet pressure, exhaustion and shielding gas reaches the fault limit values defined in the application data set (red range). ▪ **and** a protective glass cartridge is plugged in. ▪ and the focusing optics has **no** malfunction. ▪ **and** the cooling values of the focusing optics are within the admissible range. **Note:** If the focusing optics or light path component is not ready, details can be found in the messages displayed on the control panel. If a programmable motor-driven focus setting procedure (PMF) is in the light path, the signal can only be active if the PMF is ready for operation[1)].
1) Further information: Operator's manual "Programmable motor-driven focusing", doc no. 22-50-01-A369-CR.	

Tab. 356

Examples for focusing optics		**Examples of light path components**	
CFO	CONTROLLED FOCUSING OPTICS	TPC	TEMPERATURE PROCESS CONTROLLER
MFO	MANUAL FOCUSING OPTICS	PCU	POWER CALIBRATION UNIT
PFO	PROGRAMMABLE FOCUSING OPTICS	PMF	PMF: PROGRAMMABLE MOTORIZED FOCUSING

Examples of focusing optics and light path components

Tab. 357

MONITORING_SLOT

Functions	The status signal indicates local monitoring of this slot.
Active signal	The output signal **MONITORING_SLOT** is active if local monitoring occurs for this slot.
Information	The external control unit connected to the slot must reset a local monitoring via the input signal **RESET**. A RESET performed by the external control connected to the slot has the following effect: ▪ The local message is acknowledged if the laser device of the processing station has **not** been assigned. ▪ The local message and a monitoring or fault message are acknowledged if the laser device has been assigned to the external control unit. **Notes:**: ▪ Check the messages displayed on the operating panel for further details on local monitoring. ▪ With TLN-fc, one slot corresponds to the assigned light path. The message behavior and the RESET signal are influenced by different system parameters. Further information: (see "Messaging and RESET behavior", pg. 258).

Tab. 358

SHUTTER_SWITCH 8 ... 1

Functions	Via the 8 output signals **SHUTTER_SWITCH 8 ... 1** the laser device outputs the positions of the available shutter switches.
Position of the shutter switches	The significance of the 8 output signals is defined from **SHUTTER_SWITCH 8** (high) to **SHUTTER_SWITCH 1** (low). The bit value of a closed shutter switch is "1". The bit value of an open shutter switch is "0". Format: Hexadecimal. Maximum value: 255 dec = FF hex, if: ▪ 8 shutter switches are available **and** ▪ all 8 shutter switches are closed.
Information	An external control can use the output signals SHUTTER_SWITCH 8 ... 1: ▪ check the positions of the available shutter switches. ▪ keep a record of the shutter switch positions.

Tab. 359

Signals of the output word 3

ALIVE_TICK

Functions	The output signal indicates by means of the regular switching between High and Low that the laser control unit is active.
Active signal	With this signal an external control can determine that the laser control is switched on. The signal changes status between High and Low with a frequency of 1 Hz if system parameter PLC_AliveTick = 1000 ms.
Information	The **ALIVE_TICK** signal query should only be made after the **LASER_IS_ON** signal has been activated for the first time during the start-up phase of the laser control. This prevents an external control from triggering an emergency switch-off during the start-up phase of the laser control. An emergency switch-off by an external control unit due to the missing **ALIVE_TICK** signal should be triggered only for the local MPI slot, for example by opening the safety circuit for the light path in question. It may happen, for example, that the connection of an MPI slot to the superordinate control is interrupted even though work is carried out at a different workstation. If the EMERGENCY STOP circuit of the laser device is opened, the laser device stops sending laser light to all light paths. An external control must scan the ALIVE_TICK signal at double frequency. Scanning frequency = 2 Hz if the system parameter **PLC_AliveTick** = 1000 ms. The **ALIVE_TICK** signal is inactive, if the system parameter **PLC_AliveTick** = 0 ms.

Tab. 360

CALIBRATION/SELF_TEST

Functions	**Laser calibration/laser self test**
Active signal	**The laser is being calibrated**or **the laser self test is active**.
Information	Possible calibration procedures include, for example, lamp factor optimization (LFO) or pump module self-test (PMST) Lamp factor optimization (LFO) is started manually, automatically or by laser program 244. The laser self test is started by laser program 245. The LASER_READY signal is inactive as long as CALIBRATION/SELF_TEST is active. Further information on TruControl 1000 can be found in the TruControl 1000 software manual doc. no. 10-06-80-A1-CR.

Tab. 361

EXT_COOLING_CIRCUIT_IS_ON

This output signal is used in various ways.

Functionality A)

Functions	The output signal shows the state of the external cooling circuit.
Active signal	If the laser device controls the external cooling circuit, the active signal reports the following status of the external cooling circuit: ▪ Valve open ▪ Pump active
Requirement	TruControl 1000 ≥ 1.88.0 CMD version ≥ 2653 (command version BV ≥ 79)

EXT_COOLING_CIRCUIT_IS_ON, functionality A) Tab. 362

Functionality B)

Functions	The output signal indicates that the laser device requires external cooling water.
Active signal	If the superordinate system, e.g. laser machine, controls the external cooling circuit, then the active signal reports that the laser device requires cooling water.
Requirement	TruControl 1000 ≥ 3.14.0 CMD3 version ≥ CMD34908 (command version CV ≥ 6) CCB version ≥ CCB_2814 (command version CV ≥ 7) CCBM version ≥ CCBM3111 (command version CV ≥ 7)

EXT_COOLING_CIRCUIT_IS_ON, functionality B) Tab. 363

LASER_No: bit 7 ... 0

Function	Via the 8 output signals **LASER_No: bit 7 ... bit 0** the control of the laser device outputs the currently assigned number.
Coding of the laser device no.	The significance of the 8 output signals is defined from **LASER_No: bit 7** (high) to **LASER_No: bit 0** (low). Format: Hexadecimal. Number of the selectable laser devices: max. 255 dec = FF hex
Information	Depending on the output number of the laser, an external control unit can: ▪ check whether the requested laser program has responded. ▪ keep a record of the laser device used. **Note:** The laser number is assigned to the variable "MPI_LaserNumber" in the dialog *>Setup >System setting >Fieldbus*.

Tab. 364

MAINTENANCE

Functions	The output signal indicates due maintenance on the laser device.
Active signal	The output signal **MAINTENANCE** becomes active when maintenance on the laser device is required.
Information	An external control unit can evaluate the output signal **MAINTENANCE** and transfer it accordingly in order to initiate the execution of maintenance work on the laser device. In the dialog >*Maintenance/Commissioning* of the operating software of the laser device: ▪ you can see which maintenance work is due. ▪ you have to acknowledge maintenance work that has been carried out.

Tab. 365

MEASURING_VALUE_OUT_ACTIVE

Functions	The output signal indicates the output of a measured value via the output words 4 and 5.
Active signal	The output signal **MEASURING_VALUE_OUT_ACTIVE** becomes active when the output word 4 receives a measured value ID.
Information	When the output signal **MEASURING_VALUE_OUT_ACTIVE** is active, : ▪ the output word 4 receives a measured value ID. ▪ the output word 5 receives the measured value assigned to the measured value ID (e.g. the distance between focal point plane and workpiece surface).

Tab. 366

Signals of output word 4

MESSAGE_ID: bit 7 ... bit 0

Functions	The 8 output signals **MESSAGE_ID: bit 7... bit 0** are used by the laser device to output the message code (ID) of the component which triggered a message.
Coding of the message code (ID)	The significance of the 8 output signals is specified by **MESSAGE_ID: bit 7** (high) to **MESSAGE_ID: bit 0** (low). Format: Hexadecimal. Number of possible IDs: max. 255 dec = FF hex
Information	The output message code (ID) can be used by an external control unit to identify and log which component of the laser device has caused a message. A message code at the output words 4 and 5 is active for at least 200 ms. Before outputting a new message code, the value 0000 0000 is output at the output words 4 and 5 for approx. 5 ms. The message code is output individually at the MPI slots: ▪ In case of a slot fault/slot monitoring, the message code is output only at the relevant MPI slot. ▪ In case of a laser fault/laser monitoring, the message code is output at all MPI slots. Exceptions ▪ If output word 3 bit 0 signal MEASURING_VALUE_OUT_ACTIVE = 1, the message code is overwritten by a measured value, for example the distance of the focus level to the surface of the workpiece. ▪ If the output word 4 or 5 has been overwritten by the Multi PLC interface (e.g. field-bus), the message code is not displayed. The IDs of the most important components are listed in the following table. Components with multiple functions are assigned several IDs, for example component DLC: ID 06: DLC (ILV), ID 30: DLC (LMP), ...

Tab. 367

ID	Component (function)	ID	Component (function)	ID	Component (function)
00	Central processor	30	DLC (LMP)	D0	CFO
06	DLC (ILV)	34	CMD (INV)	D8	CLI
0A	ASV3 (V)	37	DLC (LLR)	E3	IIO..
0B	ASV3 (SW)	38	CMD (CLK)	E4	PFO
0D	ASV3 (VZ)	3C	LLKM	EB	CPSP
0E	ASV3 (VW)	4A	DLC (FGD)	F6	MPI
0F	ASV3 (SZ)	4E	PFO version 2		
18	DMB	90	SFO		

Component IDs, examples

Tab. 368

MESSAGE_NUM: bit 7 ... bit 0

Functions	The 8 output signals **MESSAGE_NUM: bit 7 ... bit 0** are used by the laser device to output the message code (NUM) that indicates, in case of several components of the same type, which component has triggered a message.
Coding of the message code (NUM)	The value of the 8 output signals is defined by the **MESSAGE_NUM: bit 7** (high) to **MESSAGE_NUM: bit 0** (low). Format: binary (bit coded). Number of possible NUM: 8 (hex: 01, 02, 04, 08, 10, 20, 40, 80)
Information	On the basis of the output message code (NUM), an external control unit can: ▪ determine the component that has caused a message in case there are several components of the same type. ▪ keep a record of the message code (NUM) together with the signaling component. Example: MESSAGE_ID = 0E hex, MESSAGE_NUM = 04 hex. The third component of type ASV3 (VW) resulted in a message. A message code at the output words 4 and 5 is active for at least 200 ms. Before outputting a new message code, the value 0000 0000 is output at the output words 4 and 5 for approx. 5 ms. The message code is output individually at the MPI slots: ▪ In case of a slot fault / slot monitoring the message code is output only at the relevant MPI slot. ▪ In case of a laser fault / laser monitoring the message code is output at all MPI slots. Exceptions ▪ If output word 3 bit 0 signal MEASURING_VALUE_OUT_ACTIVE = 1, the message code is overwritten by a measured value, for example the distance of the focus level to the surface of the workpiece. ▪ If the output word 4 or 5 has been overwritten by the Multi PLC interface (e.g. fieldbus), the message code is not displayed.

Tab. 369

Signals of the alternative output word 4

MEASURING_VALUE_ID: bit 15 ... bit 0

Functions	The 16 output signals **MEASURING_VALUE_ID: bit 15 ... bit 0** are used by the laser device to output the measured value ID, which indicates which measured value contains the output word 5.
Coding of the measured value ID	The value of the 16 output signals is defined by **MEASURING_VALUE_ID: bit 15** (high) to **MEASURING_VALUE_ID: bit 0** (low). Format: Hexadecimal. **Definition:** The output word 4 contains: ▪ with MEASURING_VALUE_ID: bit 15 ... bit 8 = FF hex: measured value ID (see below). or ▪ with MEASURING_VALUE_ID: bit 15 ... bit 8 ≠ FF hex: message code (ID and NUM) (see "Signals of output word 4", pg. 216). Identifications of the measured values output via output word 5: ▪ FF01 hex: – Distance between the focal point plane and workpiece surface. ▪ FF02 hex ... FFFF hex: – Reserve
Information	**Note:** The output word 4 contains a measured value ID if the output signal **MEASURING_VALUE_ACTIVE** (bit 0) of the output word 3 is active.

Tab. 370

Signals of output word 5

MESSAGE_CODE: bit 15 ... bit 0

Functions	The laser device outputs the message code of a reporting component via the 16 output signals **MESSAGE_CODE: bit 15 ... bit 0**.
Coding of the message code	The significance of the 16 output signals is defined from **MESSAGE_CODE: bit 15** (high) to **MESSAGE_CODE: bit 0** (low). Format: Hexadecimal. The value range of monitoring and fault messages is defined: ▪ 0000 hex ... C7FF = fault message ▪ C800 hex ... FFFF = monitoring message
Information	The imessage code is composed of: ▪ the MESSAGE ID of the component causing the message. ▪ MESSAGE_ NUM of the component causing the message if several components of the same type exist. ▪ MESSAGE_CODE indicating the cause of a message. Example of a message code structure XXYY ZZZZ: ▪ XX = message code MESSSAGE_ID (output word 4). ▪ YY = message code MESSAGE_NUM (output word 4). ▪ ZZZZ = message code MESSAGE_CODE for the cause of a message. A message code at the output words 4 and 5 is active for at least 200 ms. Before outputting a new message code, the value 0000 0000 is output at the output words 4 and 5 for approx. 5 ms. The message code is output individually at the MPI slots: ▪ In case of a slot fault / slot monitoring the message code is output only at the relevant MPI slot. ▪ In case of a laser fault / laser monitoring the message code is output at all MPI slots. Exceptions ▪ If output word 3 bit 0 signal MEASURING_VALUE_OUT_ACTIVE = 1, the message code is overwritten by a measured value, for example the distance of the focus level to the surface of the workpiece. ▪ If the output word 4 or 5 has been overwritten by the Multi PLC interface (e.g. fieldbus), the message code is not displayed.

Tab. 371

Signals of alternative output word 5

MEASURING_VALUE: bit 15 ... bit 0

Functions	The 16 output signals **MEASURING_VALUE_ID: bit 15 ... bit 0** are used by the laser device to output the measured value, which is assigned to the alternative output word 4.
Requirement	A teach cartridge has been plugged in into the focusing optics.
Coding of the measured value	The significance of the 16 output signals is defined from **MEASURING_VALUE: bit 15** (high) to **MEASURING_VALUE: bit 0** (low). Format: Hexadecimal. Neutral value range: 0 ... 65535 dec = 0000 ... FFFF hex The ranges of values for the measured values are defined depending on the measured variables.
Information	**Measured variable: focus position** The measured variable focus position specifies the distance between focal point plane and workpiece surface. Defined value range: -200 dec ... +200 dec with 0.1 mm/digit The value range corresponds to a distance between focal plane and workpiece surface from -20.0 mm (focal point in the workpiece) to +20.0 mm (focal point above the surface of the workpiece). Examples: Hex value = 0000 corresponding to 0.0 mm Hex value = FFFF corresponding to -0.1 mm Hex value = 0001 corresponding to +0.1 mm Measurements are made at intervals of 300 ms. **Note:** To enable transmission of a measured value to an external control unit (e.g. PLC), it must be known which light path of the laser device is assigned to the external control unit. For this purpose, the light path is used which is specified in the first line of the laser program. The external control specifies the number of the laser program used via the input word 1 of the Multi PLC interface. With TLN-fc, the light path is permanently assigned to an external control unit via the dialog *>Setup >Multi PLC interface (MPI)*.

Tab. 372

Signals of output word 6

LASER_POWER: bit 15 ... bit 0

Functions	Using the 16 output signals **LASER_POWER: bit 15 ... bit 0** the laser device: ▪ outputs the measured laser power value of CW lasers. ▪ outputs the average laser power value of pulsed lasers calculated on the basis of measurements.
Coding of the laser power	The value of the 16 output signals is defined from **LASER_POWER: bit 15** (high) to **LASER_POWER: bit 0** (low). Format: Hexadecimal. **With CW lasers:** Value range: 0000 hex ... FFFF hex equates to 0 W ... 65535 W Resolution: 1 W/digit **With pulsed lasers:** Value range: 0000 hex ... FFFF hex equates to 0.0 W ... 6553.5 W Resolution: 0.1 W/digit
Information	**For CW lasers:** The laser power measured by means of a power and energy measuring sensor (LEM) is binary-coded and output as a measured value. Example: Hex value = 0DAC corresponds to a laser power of 3500 W. **For pulsed lasers:** The value calculated from the measured pulse power, pulse duration and frequency is output as average laser power (P_{av}). Example: Hex value = 0BB8 corresponds to an average laser power of 300.0 W. The system parameter PLC_PowerOutTick can be used to set the interval at which the value is updated. PLC_PowerOutTick = 0: In case of normal utilization of the system, the value is updated approx. every 0.2 s. PLC_PowerOutTick ≥ 20 ms: In case of normal utilization of the system, the value is updated at the interval given. **For laser devices of the laser device family 20-20-xx-xx:** The resolution of the laser power is set with system parameter **PLC_LaserPowerResolution**: ▪ 0.000001 W/digit ▪ 0.001 W/digit ▪ 0.1 W/digit ▪ 1 W/digit (factory setting) ▪ 10 W/digit ▪ 100 W/digit ▪ 1000 W/digit

Tab. 373

Signals of output word 7

ENERGY: bit 15 ... bit 0

Functions	Via the 16 output signals **ENERGY: bit 15 ... bit 0** a pulsed laser outputs the calculation value based on measurements for the pulse energy of a laser pulse.
Coding of the pulse energy	The significance of the 16 output signals is defined from **ENERGY: bit 15** (high) to **ENERGY: bit 0** (low). Format: Hexadecimal. Value range: 0000 hex ... FFFF hex equates to 0.00 J ... 655.35 J Resolution: 0.01 J/digit
Information	**With CW lasers:** not used. **For pulsed lasers:** The value calculated from the measured pulse power and pulse duration is output as pulse energy. Example: Hex value = 1388 corresponds to a pulse energy of 50.00 J. In case of normal utilization of the system, the measured values are updated approx. every 0.5 s. The pulse energy of the last measured pulse is output and this value then remains present until the next change.

Tab. 374

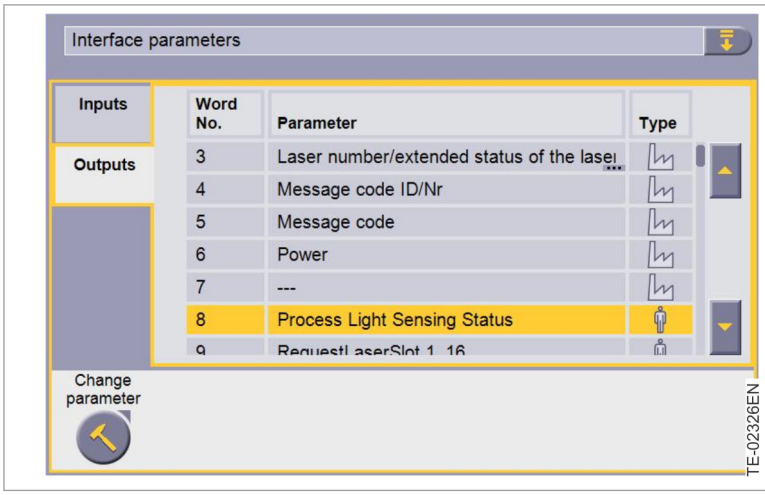
Other output parameters

In the dialog >*Setup* >*Interface parameters* the following output words can be assigned freely

- Output words 2 ... 13
- Output words 14 and 15 if system parameter **PLC_ExtendBaseIoFunction** = 0.
- Output words 16 ... 63,
  if TruControl 1000 version ≥ 1.84.0[1)].

1) Not available for the following CPX variants:

- CPX V1 standard, 18-21-85-A1
- CPX V1 small, 18-22-21-A1
- CPX V2 standard, 18-21-85-A3
- CPX V2 small, 18-21-85-A4



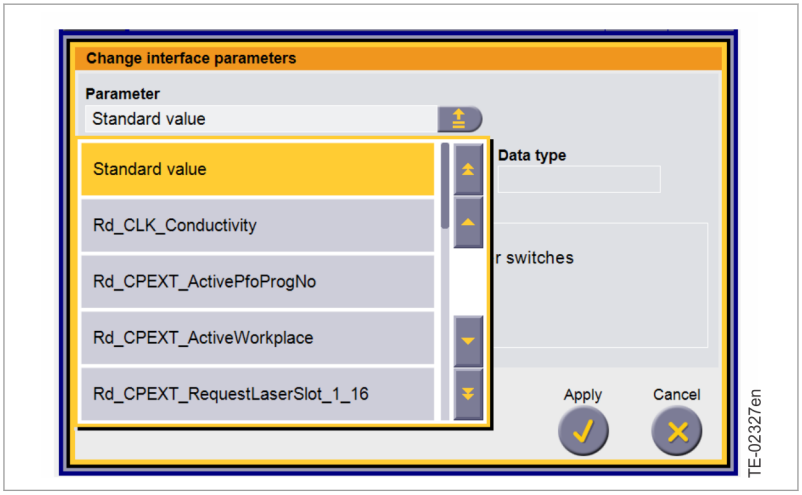
Interface parameters dialog Fig. TE-02326

Column Type	Description
	Default value: The output word is preset by TRUMPF.
	Other output parameters: The output word is set by the user.

Tab. 375

	If you want to assign another output parameter to an output word, select *Change parameter*.

Tab. 376



Changing interface parameters dialog Fig. TE-02327

The output words 2 ... 7 are preset with defaults values (standard output parameters) by TRUMPF. They have been described in the preceding sections. If you assign any other value to these output words, the default values are no longer available.

If you select the **Default value** parameter for output word 7 of a CW laser, output word 7 is not used. If you select the output words 8 .. 63 **Default value** for the output words 8 ... 63, the output words 8 ... 63 are not used.

The following table contains the description of the other output parameters.

Parameter	Description	Type
Default value	The default values (standard output parameters) are described in the preceding sections.	
PLC output (CV) x words[1)]	Parameters for the VisionLine image processing system[1)], x = 6, 8, 16, 32, 40, 48 or 56 output words. The parameters **PLC output (CV) x words** are transferred in a block if **PROG_COMPLETED** is active (see "PROG_COMPLETED", pg. 206) Diagnostics: (see "Diagnostics Dynamic table (XVar)", pg. 269).	
Device level (ESC)	Contains signal ESC_DEVICE_LEVEL, bit 0 ... 7 Energy Save Level of the laser device(see "Device level (ESC)", pg. 231).	
Slot level (ESC)	Contain signal ESC_SLOT_LEVEL, bit 0 ... 7: Energy Save Level of the MPI slot (see "Slot level (ESC)", pg. 231)	
Slot RTTO (ESC)	Contains signal ESC_SLOT_RTTO, Bit 0 ... 15: Time until the MPI slot has reached the "Ready To Operate" state(see "Slot RTTO (ESC)", pg. 231).	
Slot state (ESC)	Contains the following status information on Energy Save Concept (see "Slot state (ESC)", pg. 232):	
Bit 0	ESC_AVAILABLE.	
Bit 1	ESC_ACCEPTED.	
Bit 2	ESC_NOT_ACCEPTED.	
1) Parameter PLC output x words (CV) are not available to the VisionLine interface.		

Parameter	Description	Type
Input word	**Input word 0 ... 15**, if TruControl 1000 Version ≥ 1.76.0. **input word 0 ... 63.**[1], if TruControl 1000 Version ≥ 1.84.0. The input word is output unchanged to the output word. In this way, the external control receives a confirmation that the content of the input word has been received by the laser control. The output word can also be used to synchronize with the external control.	
IP address ETH2	IP address of the Ethernet interface "ETH2" 2 output words. output format: — Example IP address 192.013.010.011: Word 1 low-order word — 0x0A0B: 010.011 word 2 high-order word — 0xC00D: 192.013	
IP address LAN	IP address of the "LAN" Ethernet interface: For description see IP Address ETH2.	
IP address SERVICE	IP address of the "SERVICE" Ethernet interface: For description see IP Address ETH2.	
Laser power nominal	Rated power of the laser device unit: Watt.	
Laser type 1. byte 2. byte	Coding of the laser device type 2. Group of the laser device type. 3. Group of the laser device type. Examples for the 2. group: 06: TruPulse. — 10: TruDisk. 09: TruMicro. — 11: TruDiode. Example for laser type: TruDisk 6003 (6C), laser device family 20-10-03-A346: Output value: 10-03 ≙ 0x0A03. The first group of the laser device type is not emitted. For laser devices, it is always = 20. The fourth group is not output either.	
Measured power (PCP)	Laser power of the last measurement with Power Calibration Program (PCP). Further information: operator's manual CalibrationLine Focus Calibration Program & Power Calibration program, doc.-no. 07-12-12-A8-C). ▪ Parameter is output individually for each MPI slot. ▪ Parameter is set to 0xFFFF (invalid) by the positive edge of PROG_ACTIVE. ▪ Parameter is output by the positive edge of PROG_COMPLETED. Resolution: 1 W. value range: 0 ... 65534 W. invalid value: 0xFFFF.	
SeamLine Remote status[2] 0x0001 0x0002 0x0004	Status of the seam position control[2] 0x0001: SL_WORKPIECE_IS_ACTIVE. 0x0002: SL_RESULT_OK. 0x0004: SL_RESULT_NOT_OK.	
SeamLine Remote result (7 words)[2]	Test result of seam position control for QDS. One bit is set per seam if the seam is OK. If the bit = 0, then the seam is not OK. Overall, 112 seams can be evaluated (7 words * 16 bits/word)[2].	

1) Not available for the following CPX variants:
- CPX V1 Standard, 18-21-85-A1.
- CPX V1 Small, 18-22-21-A1.
- CPX V2 Standard, 18-21-85-A3.
- CPX V2 Small, 18-21-85-A4.

2) Further information: Operator's manual "SeamLine Remote (SE18)", doc no. 83-01-04-A0-CR.

Parameter	Description	Type
Melt travel monitoring state	Current status of the melt travel monitoring in the light path.	
(PFO) 0x0000	In the light path there is no melt travel monitoring or a reset has been carried out.	
0x0001	Melt travel monitoring active.	
0x0002	Melt travel monitoring ended successfully.	
0x0004	Melt travel monitoring ended, limit violated.	
0x0008	Melt travel monitoring aborted.	
0x0020	Melt travel not reached.	
0x0100	Start position blow limit value.	
0x0200	Start position above limit value.	
0x0400	Radiation duration below limit value.	
0x0800	Radiation duration above limit value.	
0x1000	Melt travel below limit value.	
0x2000	Melt travel above limit value.	
TPC state	Current state of the temperature control (TPC) in the light path:	
0x0000	There is no temperature control in the light path or reset has been carried out.	
0x0001	Temperature control active.	
0x0002	Temperature control ended successfully.	
0x0004	Temperature control finished, limit value violated.	
0x0008	Temperature control aborted.	
0x0010	Heat-up temperature reached.	
0x0100	Temperature below limit value.	
0x0200	Temperature above limit value.	
0x0400	Power below limit value.	
0x0800	Power above limit value.	
0x1000	Average temperature below limit value.	
0x2000	Average temperature below limit value.	
0x4000	Average power below limit value.	
0x8000	Average power above limit value.	
Focus or lens position (PMF)	Current focus/lens position of the PMF in the light path, for TruControl 1000 version < 2.34.0 **CCM_MotColliPosLightPath**. If system parameter FO_CollimationMotor = 0, the focus position is output. If system parameter FO_CollimationMotor = 1, the lens position is output. Resolution: 10 µm. Invalid value: 0x7FFF: There is no PMF in the light path. Further information: Operator's manual of the PMF, doc.-nr.: 22-50-01-A369-CR.	
Rotary module orientation (PMF)	Current position of the rotary module of the PMF in the light path In case of TruControl 1000 Version < 2.34.0: **CCM_RotaryModulePosLightPath**.	
Bit 15	Rotary module ready.	
bits 0 ... 8	Orientation value in degrees, value range 0° ... 360°.	
	Take Multi-PLC-interface parameter **Byte-Swap** into account. Invalid value: 0x7FFF: In the light path there is no PMF. Further information: operator's manual of the PMF, dok.-no.: 22-50-01-A369-CR.	
Crossjet pressure (CFO)	**Pressure Crossjet**: Resolution: 0.01 bar. Value range: -327.67 ... 327.66 bar. Invalid values: 0x7FFF and 0x8000.	
Exhaustion pressure (CFO)	**Pressure exhauster.** Resolution: 1 mbar. Value range: -32767 ... 32766 mbar. Invalid values: 0x7FFF and 0x8000	

Parameter	Description	Type
Protective glass contamination (CFO)	**Protective glass contamination (CFO).** Invalid values: 0x7FFF and 0x8000	
Shielding gas flow (CFO)	**Shielding gas flow.** Resolution: 0.1 l/min. Value range: -3276.7 ... 3276.6 l/min. Invalid values: 0x7FFF and 0x8000.	
Conductivity (CLK)	**Conductivity** of the cooling. Resolution: 0.5 µS. Value range: 0 ... 32767 µS. Invalid value: 0xFFFF.	
Conductivity (CCB/CCBM/EEC/CCR)	**Conductivity** of the cooling water. Invalid values: 0x7FFF and 0x8000. If a CCBM is installed, then the parameter Conductivity (CCB) does not contain any evaluable value. If a CCBM and an EEC are installed, then only Conductivity (ECC) contains evaluable values.	
Active PFO program number	**Number of the active PFO program**. Invalid value: 0x0000.	
Active workplace	**Number of the active external control (MPI slot).**	
Energy laser program	**Measured energy value**. The entire emitted laser energy of a laser program is displayed. ▪ Available if the following conditions are met: – TruControl 1000 version ≥ 1.88.0. – Board DLC2: ≥ DLCD1226, command version ≥ 42 or – from CPX V3 board DLCA: ≥ DLCD1617, command version ≥ 129. or – Board DLCM: ≥ DLCM1733 and TruControl version ≥ 3.20.0. ▪ The measured value is set to zero by the positive edge of the PROG_ACTIVE signal. ▪ After program end, the value remains unchanged until the next program start. ▪ The system parameter LCU_EnergyLaserprogramCycleTime is used to set the time after which the value is updated. Resolution 1 J/Digit. Invalid value: 0x0000.	
Energy laser program 10J	Description: see Energy laser program. Resolution 10 J/Digit. Invalid value: 0x0000.	
Request laser slot 1-16 Bit 0 : Bit 15 | Slot 1 : Slot 16	Bit-coded information on MPI slots which have requested the laser via the signal **REQUEST_LASER** part 1: MPI slot 1 ... 16. The bit for the own slot is not set. Example: Slots 1, 2 and 3 have requested the laser. Output at the Multi-PLC interface: Slot 1: 0000 0000 0000 0110. | Slot 3: 0000 0000 0000 0011. Slot 2: 0000 0000 0000 0101. | Slot 4: 0000 0000 0000 0111.	
Request laser slot 17-32 Bit 0 : Bit 15 | Slot 17 : Slot 32	Bit-coded information on MPI slots which have requested the laser via the signal **REQUEST_LASER** part 2: MPI slot 17 ... 32. The bit for the own slot is not set. Example: Slots 17, 18 and 19 have requested the laser. Output at the Multi-PLC interface: Slot 17: 0000 0000 0000 0110. | Slot 19: 0000 0000 0000 0011. Slot 18: 0000 0000 0000 0101. | Slot 20: 0000 0000 0000 0111.	

Parameter	Description	Type
EM pulse divider (DLC)	Current pulse splitter, available with ultra-short pulse laser devices, laser device family 20-09-xx-xx. Invalid value: 0xFFFF.	
Amplifier pulse repetition rate (DLC)	Active short pulse frequency in the resonator, available with ultra-short pulse laser devices, laser device family 20-09-xx-xx. Invalid value: 0xFFFF.	
Laser power EM digits (DLC)	Currently output laser power, available with ultra-short pulse laser devices, laser device family 20-09-xx-xx Resolution, depending on the laser device family: 20-09-06-xx: 1 W/Digit. | 20-09-07-XX: 1 W/Digit. | 20-09-09-xx: 0.01 W/Digit. Invalid value: 0xFFFF.	
Process light sensing status (DLC)	**Status information for PierceLine**: see Technical information "PFP-Control und PierceLine", Doc.-no. 21-A0-61-CP.	
Pulse counter	**Pulse counter.** Resolution: 1. Value range: 0 ... 65534. Invalid value: 0xFFFF.	
Power regulation average value (LLR)	**Average control intervention.** Resolution: 1 %. Value range: -32767 ... 32766 %. Invalid value: 0x7FFF and 0x8000.	
Power average filtered (LMP)	**Average laser power.** Resolution: 1 W. Value range: 0 ... 65534 W. Invalid range: 0xFFFF. This parameter is available if the laser is used in the operating mode PFP control, see technical information"PFP-Control and PierceLine", doc no. 21-A0-61-CP.	
Pulse energy count (LMP)	**Number of pulses** that were averaged (for calculation of the average pulse energy). Invalid value: 0xFFFF.	
PPC measured power	**Laser power** measured with the power measurement cassette (PPC). Resolution: 1 W. Value range: 0 ... 65534 W. Invalid value: 0xFFFF.	
Pulse energy	**Pulse energy** calculated from measured pulse power and pulse duration. Resolution: 1 J. Invalid value: 0xFFFF.	
Pulse energy 1kJ	**Pulse energy** calculated from the measured pulse power and pulse duration. Resolution: 1 kJ. Invalid values: 0xFFFF.	
Pulse energy 1mJ	**Pulse energy** calculated from measured pulse power and pulse duration. Resolution: 1 mJ. Invalid value: 0xFFFF.	
Pulse power average	**Average value of laser power**, calculated from the measured pulse power, pulse duration and frequency. Resolution: 1 W. Value range: 0 ... 65534 W. Invalid value: 0xFFFF.	

Parameter	Description	Type
Temperature of base plate (SBO)	Temperature of a line optics base plate, monitored by the SBO board. Invalid values: 0x7FFF and 0x8000.	
State for fieldbus (SBO)	SBO board status.	
XVar.1 ... XVar.32	**Output of the variables XVar.1 ... XVar.32:** With these variables, laser program parameters such as pulse power, pulse duration, frequency and PFO translation can be predefined. The variables are set via OPC items or the image processing interface. Further information: - (see "XVAR", pg. 47) - Interface description VisionLine Interface – TruControl 1000 Ethernet interface for image processing, doc. no. 12-12-40-A4-BA. The updating of the parameters is done cyclically every 500 ms. The totality of all XVars is not compulsorily transmitted in one block. The parameters **PLC output (CV) x words** are available as an alternative; these are transmitted in one block. Diagnostics: (see "Diagnostics Dynamic table (XVar)", pg. 269)	
BEO application number (MIO/PGM)	Application data set number of a BEO smart.	
Protective glass contamination (PGM)	Protective glass soiling invalid values: 0x7FFF and 0x8000.	
Actual temperature (TPC)[1)]	Current process temperature[1)]. Resolution: 0.1°C/Digit. Invalid value: 0xFFFF.	
Nominal temperature (TPC)[1)]	Process temperature to be regulated to[1)]. resolution: 0.1°C/Digit. Invalid value: 0xFFFF.	
Power portion core (FCM)[1)]	Power distribution core/ring/BrightLine Weld (2-in-1- or 3-in-1-LLK)[1)].	
Byte 1	0: 100 % core | 1: 100 % ring 1 | 2: 100 % ring 2 3: BrightLine Weld: variable distribution core/ring 1 via byte 2.	
Byte 2	Power share of core in %. Invalid value: 0xFFFF.	
Short pulse duration (PDC)	Short pulse duration resolution 1 fs/digit	
Crossjet pressure (MIO)	Pressure Crossjet resolution: 0.1 bar. value range: -3276.7 ... 3276.6 bar. Invalid values: 0x7FFF and 0x8000.	
Purging gas flow (MIO)	Purging gas flow rate resolution: 0.1 l/min. value range: -3276.7 ... 3276.6 l/min. Invalid values: 0x7FFF and 0x8000.	
MDE gas flow (MIO)	MDE gas flow rate resolution: 0.1 l/min. value range: -3276.7 ... 3276.6 l/min. invalid values: 0x7FFF and 0x8000.	
1) Output per slot and light path: Slot with fixed light path allocation: Values from the components in the allocated light path. Slot without light path allocation: Values from the components in the light path currently selected by the slot e.g. by a laser program or input word. If a light path does not contain a component, value 0xFFFF (invalid) is sent. Note byte swap parameter. Description applies for byte swap = False.		

Parameter	Description	Type
Shielding gas flow (MIO)	Shielding gas flow rate resolution: 0.1 l/min. value range: 0 ... 6553.5 l/min. invalid values: 0x7FFF and 0x8000.	
FCU/PCU lid state	Status of the lid of an FCU or PCU. High byte: numbers of the CLI with FCU. Low byte: numbers of the CLI with PCU. The number is specified according to the NUM scheme. Example: The output word has the value 0x0120: FCU lid of the CLI with NUM = 0x01 is open. PCU lid of the CLI with NUM = 0x20 is open. All other lids of the available FCU or PCU are closed.	
PFO fieldbus control state	If **PFO: fieldbus control** = Yes is set in the laser program, new PFO program data records for the next PFO program cycle are already made available in the active PFO program cycle.	
Bit 0 and Bit 1	With the parameters **PfoDataRdy1** and **PfoDataRdy2** a signal is sent that the new data sets can be provided. **PFODataRdy 1** and **2** must be reflected in the input parameters **Store-Values 1** and **2**.	
Bit 2	Parameter PfoProgActive, indicates that a PFO-Program is active.	
Bit 3 - Bit 7	Reserved.	
Bit 8 - Bit 15	The PstartCntr parameter indicates how many cycles the PFO program has completed.	
Purging gas pressure (CRDS)	Current pressure of the gas supply for the gas purge unit. Unit: mbar. Invalid value: 0xFFFF.	
VariMode state	VariMode state, for laser devices of the TruFiber P series (FD89), laser device family 20-14-xx-xx 0 = VariMode is off 1 = VariMode is on If VariMode is switched on, the fiber is squeezed to change the beam properties.	
Burst pattern (DLC/WPCI)[1)2)]	When Burst Mode option is activated, the burst pattern is output with this parameter.	
Base frequency (DLC/WPCI)[1)]	Base frequency	
Operating frequency (DLC/WPCI)[1)]	Operating frequency Invalid value: 0xFFFF.	
Energy level (WPCI)[1)2)]	Energy level	
VisionLine Check Status	Parameter for option VisionLine Check with focusing optics PFO and OCT sensor[3)]: Bit 0: MeasuringReady Bit 1: SensorReady Bit 2: SeamOK Bit 3: SeamNotOK Bit 4: PartOK Bit 5: PartNotOK	
PLC output (VLCK) 04 words	Parameter for VisionLine Check with focusing optics BEO and OCT sensor[3)]:	
Word 1	VLCK status, 2 bytes: see parameter **VisionLine Check Status**.	
Word 2	X-Position, 2 bytes	
Word 3	Y-Position, 2 bytes	
Word 4	Z-Position, 2 bytes	

1) Further information: See the operator's manual of the laser device.

2) An operating point change is signaled by briefly setting *energy level (WPCI)* and *burst pattern (DLC/WPCI)* to zero.

3) Available from TruControl 1000 version 3.46, further information: Software manual VisionLine OCT Check, doc.-no.: 83-01-07-A0-CR.

Parameter	Description	Type
Protective glass contamination (PFO)	Protective glass contamination (PFO) Invalid values: 0x7FFF and 0x8000.	
PCP measurement	Status of the PCP measurement	
Bit 0 Bit 1 Bit 2	= 1: Measurement is active = 1: Measurement successful = 1: Measurement not successful	= 0: Measurement not active = 0: Value has no meaning = 0: Value has no meaning
Sensor Relay Output x Words	Output data of the sensor relay, x = 9, 16, 32 or 48 (see "Configuring the sensor relay", pg. 266)	
The availability of these parameters depends on the configuration of the laser device.		

Tab. 377

Device level (ESC)

Functions	The output word **Device level (ESC)** contains the output signal **ESC_DEVICE_LEVEL**.
Active signal	This byte returns the Energy Save Level which is currently active in the laser device.
Information	Technical information "Energy Save Concept", doc.-no. 21-A0-71-CP.

Tab. 378

Slot RTTO (ESC)

Functions	The output word **Slot RTTO (ESC)** contains the output signal **ESC_SLOT_RTTO**.
Active signal	This word contains the Regular-Time-To-Operate (RTTO) in seconds for the slot. That is the time the laser device requires to change to the Ready_To_Operate status. The RTTO depends on the signal **ESC_SLOT_LEVEL** The external control unit must "wake up" the laser device at the ESC_SLOT_RTTO time before the start of production with ESC_PAUSE = 0xFFFF (END_PAUSE) (see "ESC_PAUSE", pg. 194).
Information	Technical information "Energy Save Concept", doc.-no. 21-A0-71-CP.

Tab. 379

Slot level (ESC)

Functions	The output word **Slot level (ESC)** contains the output signal **ESC_SLOT_LEVEL**.
Active signal	This byte contains the Energy Save Level which is set for the slot.
Information	Technical information "Energy Save Concept", doc.-no. 21-A0-71-CP.

Tab. 380

Slot state (ESC)

Bit	Output signal
0	ESC_AVAILABLE
1	ESC_ACCEPTED
2	ESC_NOT_ACCEPTED
3 ... 15	Reserve

Signals of the output word slot state (ESC) Tab. 381

ESC_AVAILABLE

Functions	**ESC_AVAILABLE** output signal.
Active signal	This signal is active if Energy Save Concept for this slot has been activated in the Multi PLC interface settings.
Information	Technical information "Energy Save Concept", doc.-no. 21-A0-71-CP.

Tab. 382

ESC_ACCEPTED

Functions	Output signal **ESC_ACCEPTED**. Response to the input signal ESC_PAUSE.
Active signal	This signal becomes active if the pause time (ESC_PAUSE > 0) has been accepted.
Information	The signal becomes inactive if the pause time has been canceled (ESC_PAUSE = 0). For further information: see technical information "Energy Save Concept", doc. no. 21-A0-71-CP.

Tab. 383

ESC_NOT_ACCEPTED

Functions	Output signal **ESC_NOT_ACCEPTED**. Response to the input signal ESC_PAUSE.
Active signal	This signal becomes active if the sent pause time (ESC_PAUSE > 0) has not been accepted. The signal is sent in these cases: ▪ An external control unit has sent ESC_PAUSE > 0 (Start pause) while a laser program is active for the slot. ▪ The pause time is shorter than the pause time available in the device. ▪ The pause time is shorter than the pause time transmitted before. ▪ An external control (e.g. PLC) sends Start pause, although it has activated the optimized standby mode via the ENERGY_SAVE_ON (input word 14) signal .
Information	The signal becomes inactive if the pause time has been canceled (ESC_PAUSE = 0). For further information: see technical information "Energy Save Concept", doc. no. 21-A0-71-CP.

Tab. 384

Signals of the alternative output words 14 and 15

ALIGNMENT_RECOMMENDED

Functions	Signal **ALIGNMENT_RECOMMENDED** indicates that a beam position correction is required.
Active signal	A beam position correction is required. The active process can be ended. To avoid a fault, switch the device off and then on again with LASER_ON, LASER_ON = 1 → 0 → 1 (see "LASER_ON", pg. 170).
Requirement	This output signal is only available if the following conditions have been met: ▪ Option LLK-U is used. ▪ TruControl Version ≥ 3.02 ▪ System parameter CV_HCFBC_Enable = 1.
Information	A beam position correction can also be started via the Multi PLC interface with laser program 240 ("BeamCalibration"), independent of system parameter CV_HCFBC_enable, if TruControl version ≥ 3.06 (see "PROGRAM_No: bit 7 ... bit 0", pg. 180).
1) HCFBC: Hollow Core Fiber Beam Calibration	

Tab. 385

COOLING_LOCKED

Functions	Signal **COOLING_LOCKED** indicates that the external cooling water supply must be stopped.
Active signal	The external cooling water supply should be stopped.
Requirement	This output signal is only available if the following conditions have been met: ▪ TruControl 1000 ≥ 3.14.0 ▪ CMD3 version ≥ CMD34908 (command version CV ≥ 6) ▪ CCB version ≥ CCB_2814 (command version CV ≥ 7) ▪ CCBM version ≥ CCBM3111 (command version CV ≥ 7)

Tab. 386

EMISSION_ON

Functions	Signal **EMISSION_ON** indicates that the laser is emitting power.
Active signal	The laser outputs power > 0 W.
Prerequisite	If a central shutter (VZ) is available, it must be open. This output signal can only be evaluated if the pulse duration and the pulse gap are twice as large as PLC_PowerOutTick. For example if PLC_PowerOutTick = 80 ms, the pulse duration and pulse gap must be at least 160 ms.

Tab. 387

ENERGY_SAVE_IS_ON

Functions	The output signal **ENERGY_SAVE_IS_ON** indicates that the laser device is in the energy-saving mode (Sleep mode 3). Response to the input signal **ENERGY_SAVE_ON**.
Active signal	The laser device and all laser control components except for CPX and PBT are switched off.
Requirement	This output signal is only available if the following conditions have been met: ▪ The laser control is equipped with a central processor, CPX V3 or V4. ▪ System parameter **PLC_ExtendBaseIoFunction** = 1. ▪ System parameter **LCU_EnergySaveMode** > 0. ▪ The input **24 V_STDBY** at X1 of the PBT board is supplied with voltage.
Information	The control can be switched on again with the signal **ENERGY_SAVE_OFF**, if the main switch is on "On". For further information: see technical information "Energy Save Concept", doc. no. 21-A0-71-CP. Not available for laser devices with CPX V20.

Tab. 388

FAULT_PROCESS

Function	Signal **FAULT_PROCESS** indicates when a fault message at the associated MPI slot was triggered during the machining process.
Active signal	Message which affects the process on the user side; examples: ▪ Excessive temperatures when using focusing optics with temperature control (TPC), e.g. PFO 20-2 PO. ▪ When using PFO with SeamLine Remote.
Information	Check the quality of the machined workpiece. The signal must be acknowledged with RESET.

Tab. 389

LASER_CONTROL_IS_ON

Functions	Signal **LASER_CONTROL_IS_ON** indicates that the laser control is switched on.
Active signal	The laser control is active.
Prerequisite	This output signal is only available if the system parameter **PLC_ExtendBaseIoFunction** = 1.

Tab. 390

MACHINE_COOLING_READY

Functions	The output signal **MACHINE_COOLING_READY** indicates that the machine cooling is on. Response to the input signal MACHINE_COOLING_ON.
Active signal	The machine cooling is on. Cooling water flow is available.
Requirement	This output signal is only available if the system parameter **PLC_ExtendBaseIoFunction** = 1.
Information	This output signal is required, for example, if a TRUMPF laser machine and the integrated laser device use the cooling unit of the TRUMPF laser machine jointly. The machine axes can be released.

Tab. 391

MONITORING_PROCESS

Function	Signal **MONITORING_PROCESS** indicates when a monitoring message at the associated MPI slot was triggered during the machining process.
Active signal	Message which affects the process on the user side; examples: ▪ Excessive temperatures when using focusing optics with temperature control (TPC), e.g. PFO 20-2 PO. ▪ When using PFO with SeamLine Remote.
Information	Check the quality of the machined workpiece. The signal can be acknowledged with RESET.

Tab. 392

7.3 Using interface

⚠ WARNING

Laser light can be emitted without warning following a malfunction in the energy supply (power failure) and restoration of the energy flow.

Serious injuries or extensive property damage can result.

- Condition external controls so they do not automatically or randomly send signals to the laser control to activate the laser following failure and restoration of the energy flow.

Triggering

The laser control system can be activated in two different ways:

- Synchronized triggering
- Unsynchronized triggering

Synchronized Triggering

The synchronized activation is the common activation of a laser device. It is characterized by a specified order of request and acknowledgement signals.

Example

1. An external control sets REQUEST_LASER.
2. The laser control responds with LASER_ASSIGNED.
3. The laser control indicates with LASER_READY that it is ready.
4. The external control starts the laser program.
5. The external control resets REQUEST_LASER.
6. Another external control sets REQUEST_LASER.
7. Steps 2 to 6 are repeated.

If a program start is activated without waiting for a previous acknowledgment signal, the laser control indicates a monitoring message or fault message.

If the assignment is changed with a static program start:

- the laser program is **not** started.
- a monitoring message F6xx DE28 is displayed.

The laser program is only started with the positive edge of the program start.

Note

The control unit does not report a fault if a laser program is started from a processing station to which the laser device is not assigned. The started laser program is not carried out.

Chronological sequence of the synchronized activation of two external control units (see "Alternating activation of a laser device", pg. 247).

Unsynchronized Triggering

The unsynchronized activation is intended for applications which require rapid switching between different external control units.

For continuous path control, a synchronization with the help of signals is limited. Acknowledgment signals can be ignored with an exact adjustment to external sequences.

Properties

With an unsynchronized triggering:

- the acknowledgment signals LASER_ASSIGNED and LASER_READY are no longer necessary.
- the signal for start of a laser program can be applied while the laser is not assigned yet.
- the central shutter remains open after execution of a laser program so that the laser is ready for the next request without any delay.

UNSYNC_PSTART in detail

In the UNSYNC mode, a start signal can already be applied before the laser has displayed the LASER_READY or LASER_ASSIGNED message.

If the UNSYNC_PSTART signal is fed constantly or simultaneously with the program start, no fault message is given if the laser is not ready. The laser only starts once it is ready for a program start.

This function can be used for rapid switching (about 80 ms) between various light paths.

Example:

Laser program 1 is active. The UNSYNC PSTART mode was already set at laser program start (**important**).

While the laser program is still active, the laser device is requested by another external control unit also in the UNSYNC_PSTART mode, and laser program 4 is started.

At the end of laser program 1, the central shutter will **not** be closed, but only the position of the shutter switches is adjusted to the new light path. Laser program 4 is started as soon as the switches are in the right position. The switchover time is approx. 80 ms.

Note

The switching to unsynchronized triggering already becomes active when **one** of the connected external control units has activated the UNSYNC_PSTART signal.

Mixed operation

When working with a mixed operation of unsynchronized and synchronized triggering, UNSYNC_PSTART must be deactivated by all connected external control units before the end of the last program in order to enable a synchronized triggering.

If a connected external control unit has activated UNSYNC_PSTART, they work all in the UNSYNC mode.

Switching from SYNC → UNSYNC

When switching from the synchronized to the unsynchronized triggering:

- the central shutter remains open after the program end.
- a laser program is started with the positive edge of LASER_READY if a program start is active.

Switching from UNSYNC → SYNC

When switching from unsynchronized to the synchronized triggering:

- the central shutter remains open after the program end.
- the external control unit is not allowed to carry out a program start with synchronized triggering if the laser device is not assigned.
- in case of an active program start and a not assigned laser device:
  - the laser program is not started.
  - a monitoring message is displayed.

Quitting the UNSYNC mode

The UNSYNC mode is closed if **none** of the connected external control units has activated UNSYNC_PSTART. With the falling edge of UNSYNC_PSTART, the central shutter is closed.

Starting time of UNSYNC_PSTART

If the program start takes place in the UNSYNC_PSTART mode while, for example, a laser program of another assigned external control unit is running, the starting time of the laser program depends on the end of the previous laser program.

It is not possible to define the moment for the use of the laser device by means of a program start. There is **no** time limit.

If the laser program is started in the UNSYNC_PSTART mode, the laser program does not start in the laser device unless:

- the laser device has been requested, assigned and is ready.
- the safety circuit is closed.
- a pump capacity optimization is finished (if active).
- a capacity check is terminated (if active).
- the laser program of the previous external control is finished.
- the laser device is released by the previous external control unit.

If another laser program is started in the UNSYNC mode before the end of the program, this start request is saved and carried out after the end of the running laser program.

Example:

In a laser program, the rampdown of the laser power is initiated with the **WPSL** function (Wait for Program Start "Low"). If a new start request is activated during the rampdown, the information is retained and carried out after the end of the rampdown.

Note

If a laser program is switched to the function WPSL by the Wait function via query inputs (e.g. W1H), the program duration must be reduced accordingly.

LASER_READY with UNSYNC_PSTART

When UNSYNC_PSTART is active, the PROGRAM_ACTIVE and LASER_ASSIGNED signals are **not** incorporated in the LASER _READY signal.
LASER_READY is only controlled by the following information:

- State of the safety circuits of the light paths specified by MPI and/or the laser program.
- A pump capacity optimization or a capacity check must not be active.
- Signals:
  - FAULT_LASER
  - PROG_CANCELLED
- Laser devices with laser standby mode: internal laser in the standby mode

PROG_COMPLETED with UNSYNC_PSTART

A laser program started in the UNSYNC mode does not activate the PROG_COMPLETED signal or only activates it for a short period of time at the end of the laser program if switching to another external control unit takes place immediately.

In this case, the external control must be synchronized, if necessary, to the deactivation of PROG_ACTIVE instead of PROG_COMPLETED.

Safety circuit is opened in the UNSYNC mode

If, in a light path that has just been selected, the safety circuit is opened, the laser control ends the UNSYNC mode and the central shutter is closed if present.

Note for practical use

Counting the PROG_ACTIVE signals

The number of program starts can be checked by counting the output signals PROG_ACTIVE. In the UNSYNC mode, it cannot be guaranteed by the laser device that each program request is carried out.

The chronological sequence of the unsynchronized triggering by means of two external control units is described in a separate section (see "Unsynchronized Triggering of a laser device", pg. 250).

Timing diagrams

Below, you will see the timing diagrams for:

- Starting a finite laser program dynamically.
- Starting a finite laser program statically.
- Starting a continuous laser program dynamically.
- Starting a continuous laser program statically.
- Alternating activation of a laser device.
- Unsynchronized triggering of a laser device.
- Parallel control of a laser device for the simultaneous execution of laser programs with and without pulse triggering (TruControl 1000 Version 1.78.0 and higher).

The shown timing diagrams refer to pulsed laser devices. They apply similarly to CW laser devices.

Start finite laser program dynamically

An external control unit requests the laser device, switches it on and starts the finite laser program number 5 dynamically via the PSTART_DYN input signal.

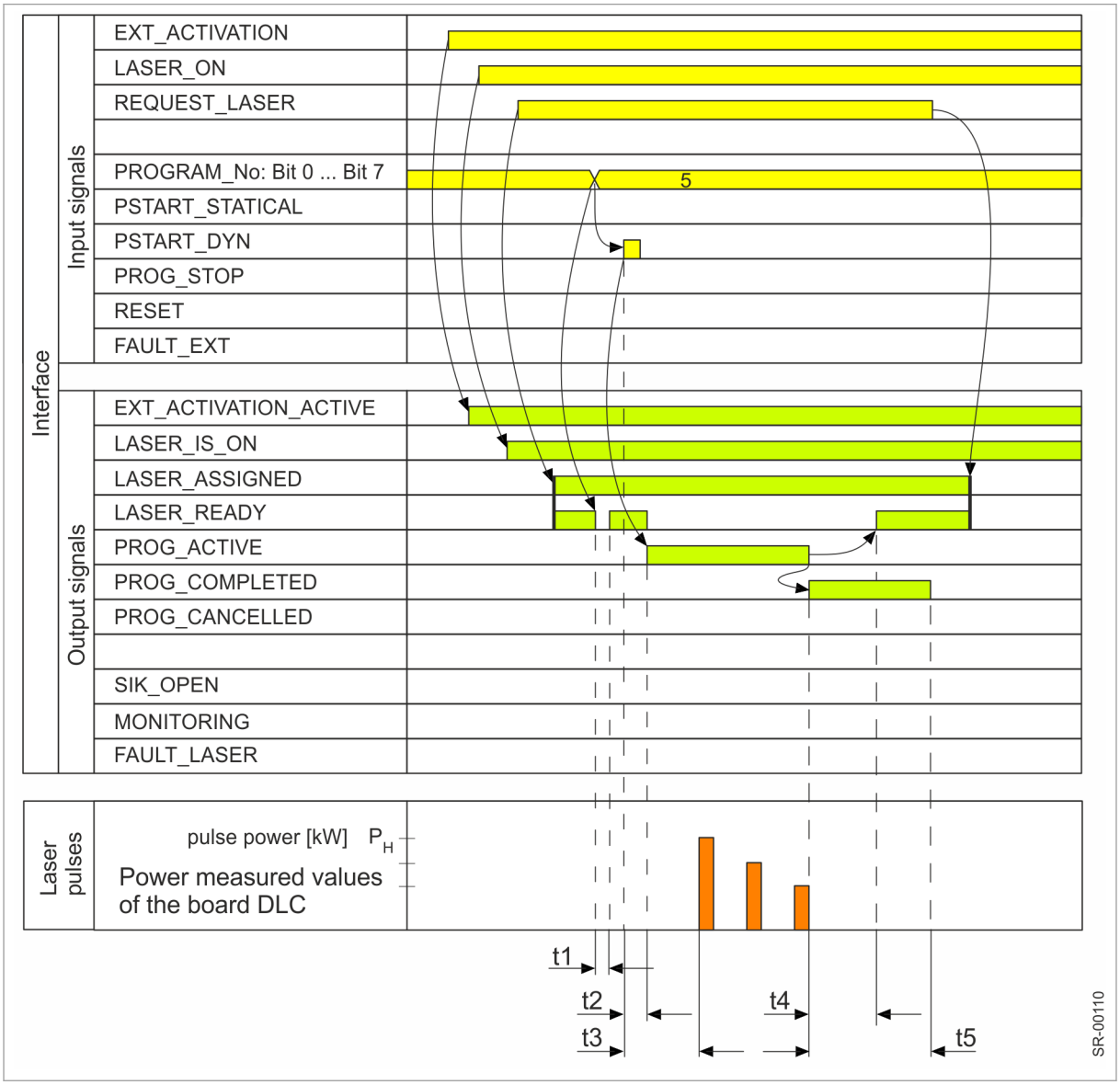


Fig. SR-00110

In laser program 5, 3 laser pulses are programmed with different laser power.

t1 The laser device control sets the **LASER_READY** output signal to inactive if it has to reset the light path when presetting the laser program. Setting of the light path may take up to 45 ms if several optical components are moved.

t2 The time from the start of the laser program to the acknowledgement by the **PROG_ACTIVE** output signal is approx. 150 µs.

t3 The time from the start of the laser program to the start of the first laser pulse lies within the range of:

- 15 ms ... 20 ms if no optical component is moved (applies only to pulsed lasers).
- 35 ms ... 55 ms if an optical component is moved.

Note

With pulsed lasers, the jittering between the request for a laser pulse and the executed laser pulse can be reduced using a real-time interface < ±40 µs.

t4 The time the laser requires after the end of a laser program until it is ready for pulsing again.

The time t4 is determined by:

- the loading time of the capacitor in the laser device (only for pulsed lasers).
- the time needed for moving the optical components.

The loading time of the capacitor in the laser device depends on the pulse energy of the last laser pulse. Depending on the laser device, loading times for minimum to maximum pulse energy are in the range of 1 ms ... 1500 ms.

The time for moving an optical component after the end of a laser program is max. 40 ms. If several optical components are moved, the time is approx. 45 ms ... 55 ms.

The time t4 is at least as long as the longest individual time plus a program running time of approx. 150 µs in the laser control.

t5 After the end of a laser program, the **PROG_COMPLETED** output signal becomes active for approx. 6 s.

The **PROG_COMPLETED** output signal becomes prematurely inactive if one of the following operations is carried out before the 6 seconds have passed:

- A laser program is restarted (with PSTART_STATICAL or PSTART_DYN).
- If system parameter **PLC_ClearProgCompletedOnSlotChange** = 1 and a change of assignment takes place.

Note

The times t1 ... t5 given describe exclusively the behavior of the laser device control. Transmission runtimes due to fieldbus or OPC are not taken into account.

Start finite laser program statically

An external control unit requests the laser device, switches the laser on and starts the finite laser program number 5 statically via the input signal PSTART_STATICAL.

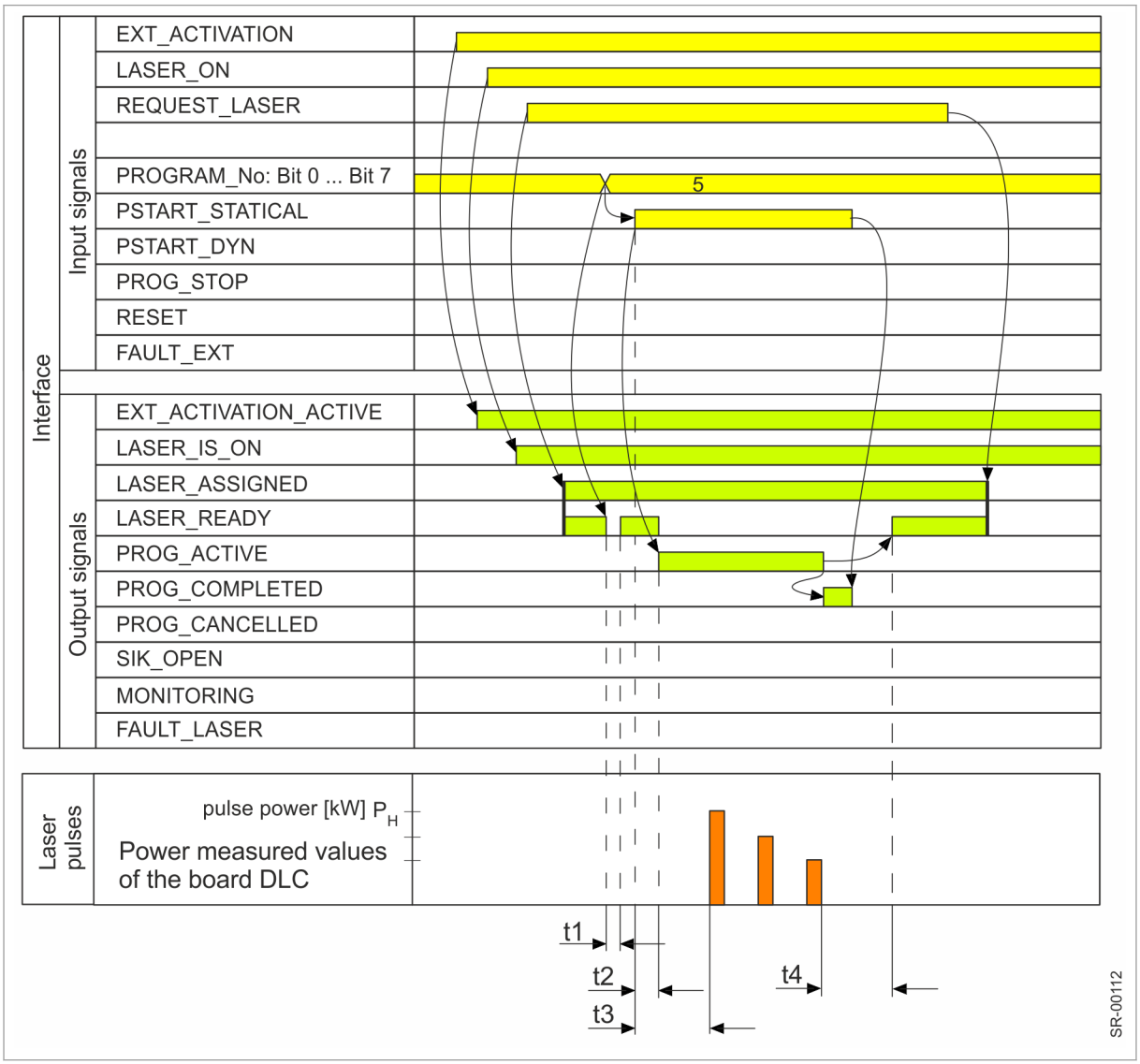


Fig. SR-00112

Note

The timely sequence of a laser program is identical if the laser program has been started dynamically or statically.

Start infinite laser program dynamically

An external control requests the laser device, switches on the laser and starts the infinite laser program No 7 dynamically via the input signal PSTART_DYN. A dynamically started laser program with pulse number = ∞ ("Continuous") or with pulse type = "Continuous wave" (with CW lasers) can stop an external control by means of the input signal PROG_STOP.

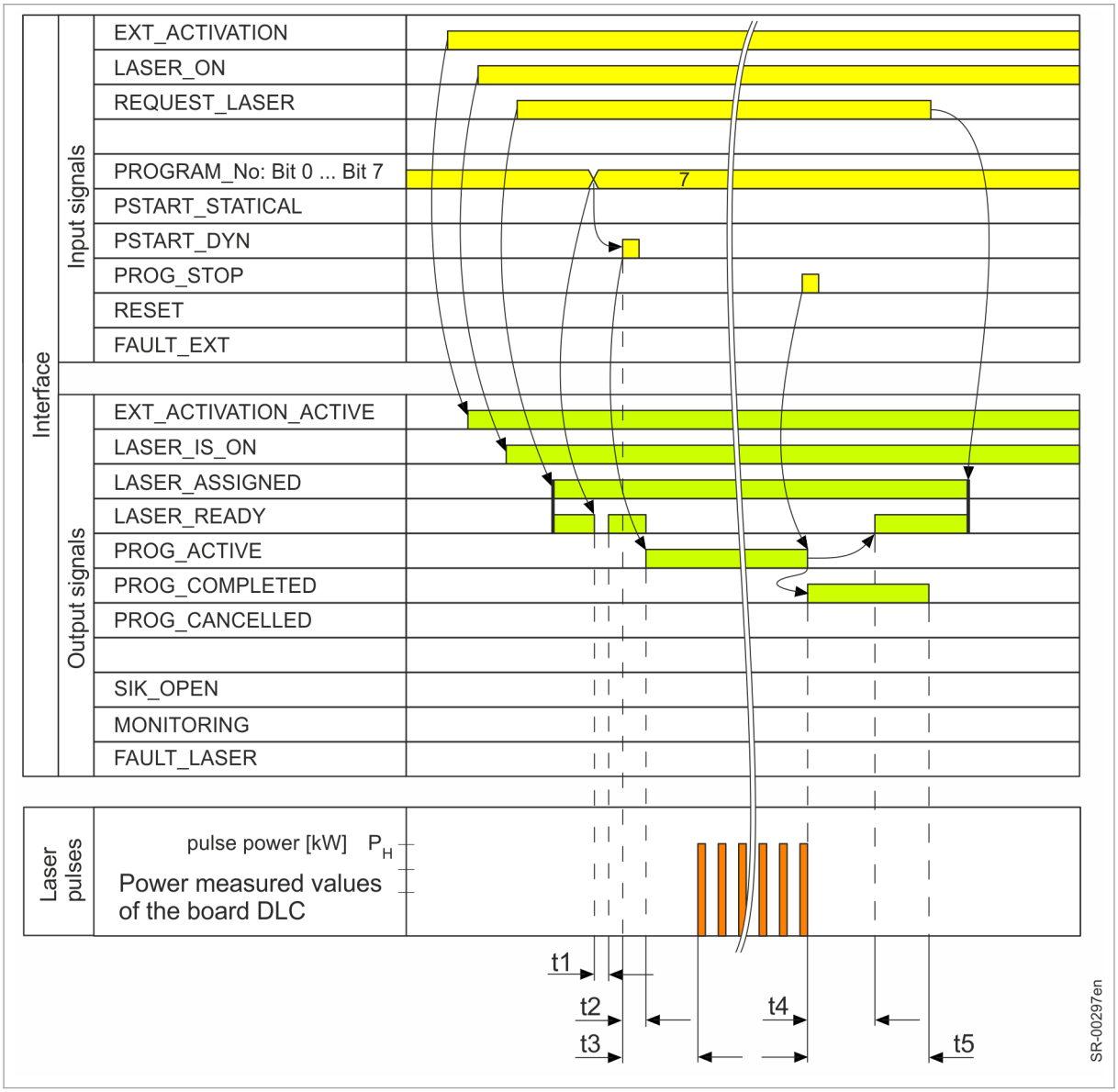


Fig. SR-00297

Start infinite laser program statically

An external control requests the laser device, switches the laser on and starts the infinite laser program No 7 statically via the input signal PSTART_STATICAL. A statically started laser program with pulse number = ∞ ("Continuous") or with pulse type = "Continuous wave" (with CW lasers) can stop an external control by means of the input signal PSTART_STATICAL.

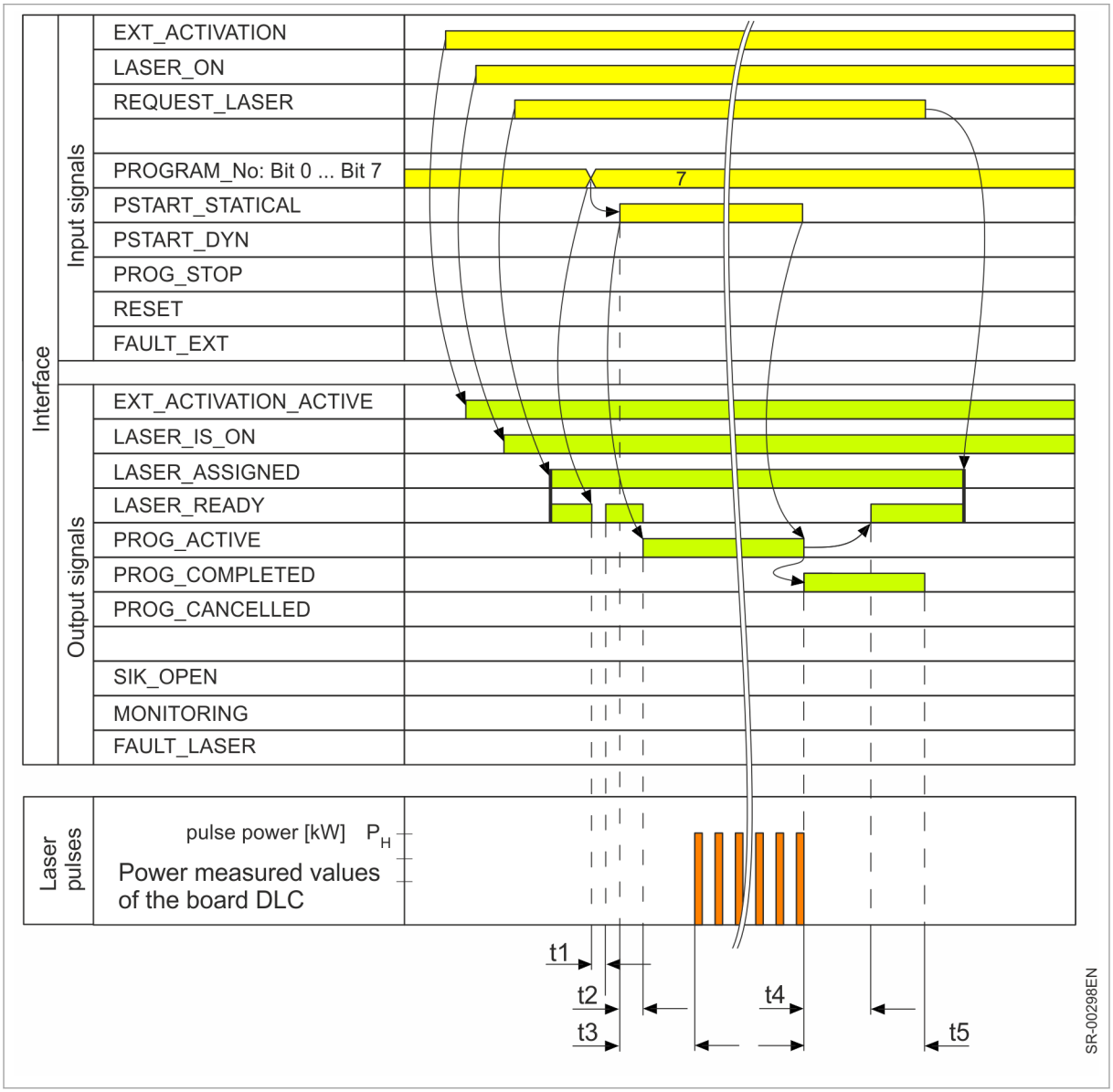
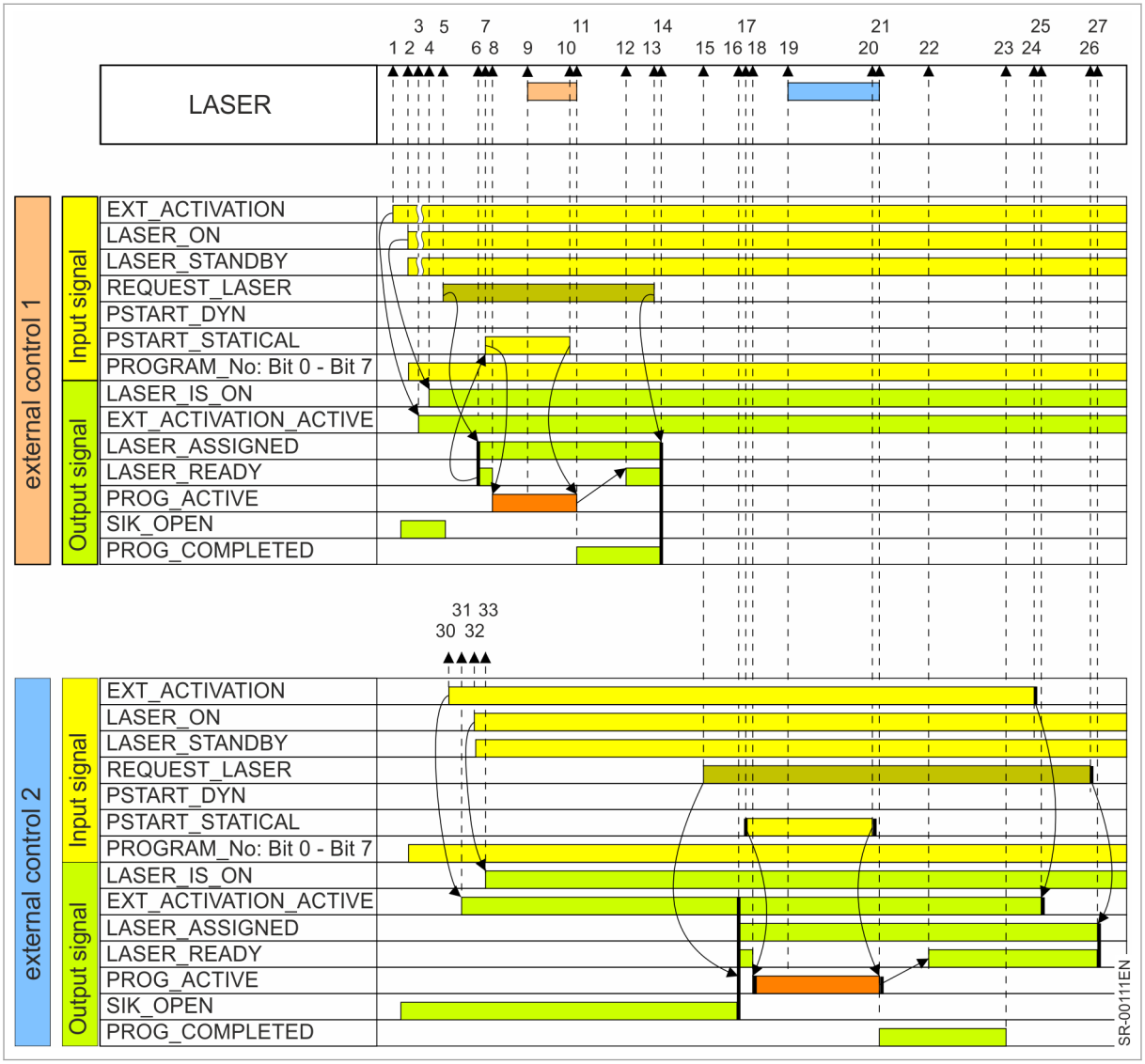


Fig. SR-00298

Alternating activation of a laser device

Two external controls request the laser device and wait for assignment and start an infinite laser program when being

assigned. In the timing diagram shown below, the two controls are connected with the laser device via a Multi PLC interface (fieldbus, OPC or parallel I/O interface). The entered times are explained in the following table.



Example for an alternating activation of a laser device Fig. SR-00111

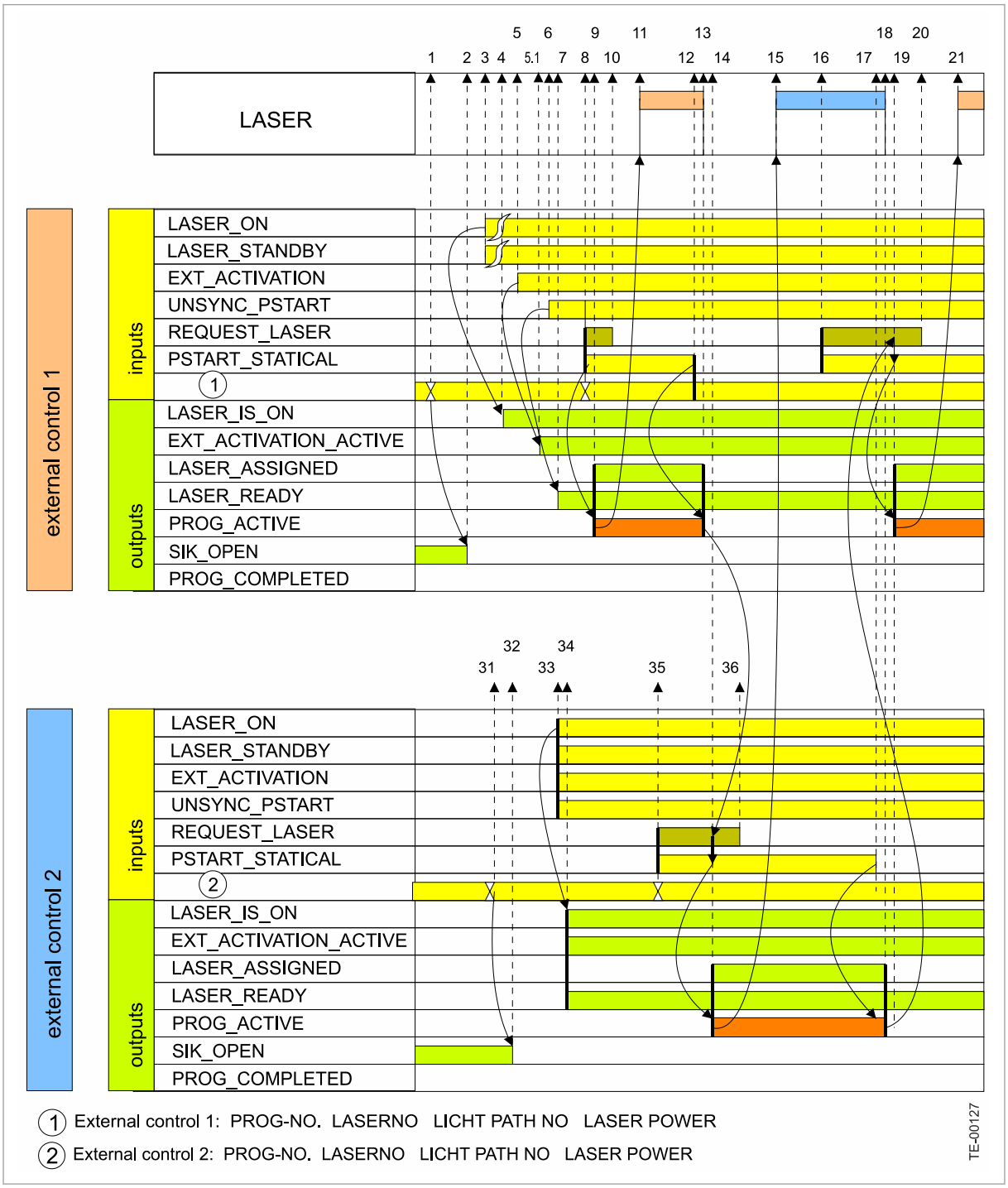
Time of event	Definition
1	The external control 1 activates the input signal EXT_ACTIVATION for an external control of the laser device.
2	The external control 1 activates the input signal LASER_ON and with laser devices with laser standby mode also the input signal LASER_STANDBY, so that it is ensured that the laser is already switched on when being requested.
3	The laser device acknowledges the external activation by the active output signal EXT_ACTIVATION_ACTIVE.
4	The laser device activates the output signal LASER_IS_ON.

Time of event	Definition
5	The laser device is requested by the external control 1 using the active REQUEST_ LASER input signal.
6	The laser device is assigned by the external control 1 and activates the LASER_ ASSIGNED output signal. If the laser is ready (no fault, safety circuit closed), the laser device activates the output signal LASER_READY. A laser program can be started.
7	The external control 1 starts a laser program.
8	The laser device activates the output signal PROG_ACTIVE. If the central shutter is closed, it is opened via a command in the laser program.
9	Laser light is emitted via the preset light path.
10	The external control 1 sets the input signal PSTART_STATICAL inactive.
11	The laser control unit activates the output signal PROG_COMPLETED for approx. 6 s.
12	The laser device activates the LASER_READY output signal. The central shutter can be closed via a command in the laser program.
13	The control 1 sets the input signal REQUEST_LASER inactive.
14	The laser device sets the output signal LASER_ASSIGNED inactive. The output signal PROG_ COMPLETED is likewise deactived if system parameter PLC_ClearProgCompletedOnSlotChange = 1.
30	The external control 2 activates the input signal EXT_ACTIVATION for an external control of the laser device.
31	The laser device acknowledges the external activation by the active output signal EXT_ACTIVATION_ACTIVE.
32	The external control 2 activates the input signal LASER_ON and with laser devices with laser standby mode also the input signal LASER_STANDBY, so that it is ensured that the laser is already switched on when being requested.
33	The laser device activates the output signal LASER_IS_ON.
15	The laser device is requested by the external control 2 using the active REQUEST_ LASER input signal.
16	The laser device is assigned by the external control 2 and activates the LASER_ ASSIGNED output signal. If the laser is ready (no fault, safety circuit closed), the laser device activates the output signal LASER_READY. A laser program can be started.
17	The external control 2 starts a laser program.
18	The laser device activates the output signal PROG_ACTIVE. If the central shutter is closed, it is opened via a command in the laser program.
19	Laser light is emitted via the preset light path.
20	The external control 2 sets the input signal PSTART_STATICAL inactive.
21	The laser control activates the output signal PROG_COMPLETED.
22	The laser device activates the LASER_READY output signal. The central shutter can be closed via a command in the laser program.
23	The output signal PROG_COMPLETED is deactivated approx. 6 seconds after the end of the laser program.
24	The input signal EXT_ACTIVATION must be set inactive, if the laser device should to be controlled via the operating panel. The laser device can no longer be addressed via the Multi PLC interface (fieldbus, OPC or parallel-I/O interface).
25	The laser device set the output signal EXT_ACTIVATION_ACTIVE inactive. The laser device can be controlled via the operating panel.
26	The external control 2 sets the input signal REQUEST_LASER inactive.
27	The laser device is no longer assigned to the external control 2.

Explanation of the times with alternating activation Tab. 393

Unsynchronized Triggering of a laser device

In unsynchronized triggering, a control unit requests the laser device and starts the laser program without waiting for the assignment (see "UNSYNC_PSTART", pg. 177).



Example for unsynchronized triggering of a laser device Fig. TE-00127

Time of event	Information
1	The light path number is created by the external control 1. The previously selected light path 0 is invalid.
2	The safety circuit of the currently selected light path is closed.
3	The external control unit 1 activates the signal LASER_ON and, for lasers with standby mode, also the signal LASER_STANDBY in order to make sure that the laser device is already on when it is requested.
4	The laser device responds with the LASER_IS_ON signal.
5	The external control unit 1 activates preliminarily the EXT_ACTIVATION signal.
5.1	The laser device acknowledges by sending the EXT_ACTIVATION_ACTIVE signal. **Note** No preparatory activation of the EXT_ACTIVATION signal is necessary. It is recommended to do without a change in favor of the dynamic behavior.
6	The external control unit 1 activates the UNSYNC_PSTART (program start without laser being ready).
7	In the UNSYNC_PSTART mode, the laser is ready even if the interface has not been assigned. LASER_READY becomes active. **Note** In this mode, "LASER_READY" means: The program start is possible. The laser device may still be connected to another processing station. The safety circuit is closed. A laser device with laser standby mode works in the standby mode. No fault is present. There is not any program cancellation signalized by the PROG_CANCEL-LED signal.
8	The external control unit 1 simultaneously requests the laser device, selects the number of the laser program and starts a laser program. The light path number is not changed.
9	The laser device acknowledges: The laser device is assigned to the external control unit 1. The signals LASER_ASSIGNED, EXT_ACTIVATION_ACTIVE and PROG_ACTIVE are active. The laser program is performed. The LASER_READY signal remains active since in this mode an active laser program does not deactivate LASER_READY. The shutter is opened. If the central shutter is open, laser light will be emitted after about 60 ms.
10	The REQUEST_LASER signal is deactivated by the external control unit 1. The laser program is carried out as long as the PSTART_STATICAL signal is active. Another control unit can request the laser device and prevent the shutter from closing upon completion of the laser program. This allows a quick switch-over.
11	The shutter is open and laser light is emitted on the selected light path.
12	The PSTART_STATICAL signal is deactivated by the external control unit 1, the laser program is finished.
13	The REQUEST_LASER signal is deactivated by the laser device, since the laser device is no longer required by the external control unit 1. Laser light is no longer emitted.
:	
31	The external control 2 changes the light path number.
32	The safety circuit of the new light path is closed. The SIK_OPEN signal is deactivated by the laser device. **Note** In order for the laser control to be able to correctly determine the SIK_OPEN signal, the light path numbers, laser program and laser device must be constantly active. This optimizes the time sequence.

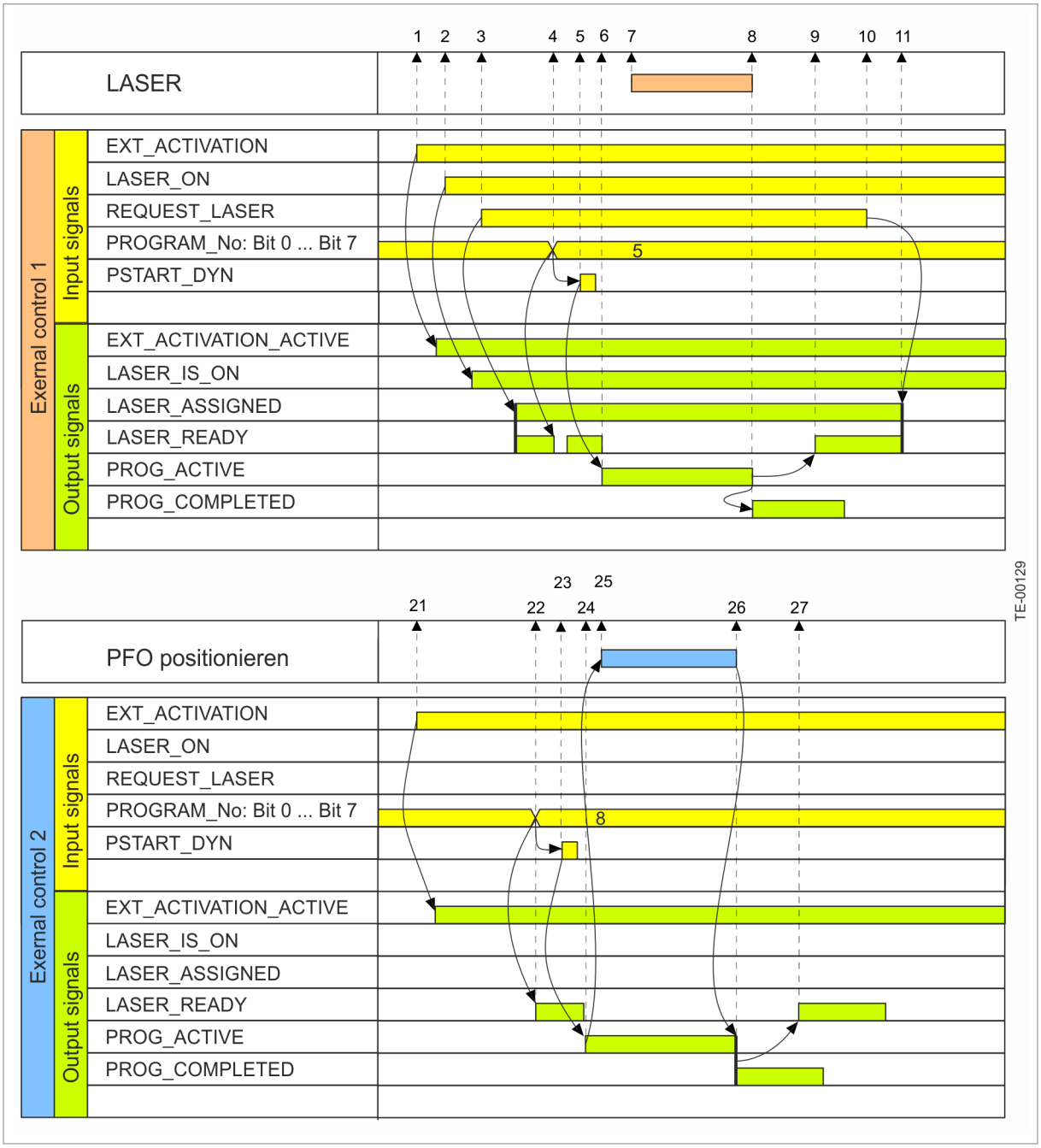
Time of event	Information
33	The external control unit 2 activates simultaneously the signals LASER_ON, EXT_ACTIVATION and UNSYNC_PSTART and, for lasers with a laser standby mode, also the LASER_STANDBY signal. The UNSYNC_PSTART mode is on.
34	The laser device is on and is ready in the UNSYNC_PSTART mode.
35	While a laser program is running from the external control unit 1, the external control unit 2 requests the laser device and activates the PSTART_STATICAL signal to start a laser program.
:	
14	The request and the program start of an external control unit 2 are processed. The acknowledgements are sent to the external control unit 2. The external control unit 2 receives the assignment and the laser program is started.
15	The laser light is emitted on the selected light path.
16	While a laser program is running from the external control unit 2, the external control unit 1 requests the laser device again and starts a laser program. The external control unit 1 activates the signals REQUEST_LASER and PSTART_STATICAL.
17	The PSTART_STATICAL signal is deactivated by the external control unit 2, the laser program is finished.
18	The REQUEST_LASER signal is deactivated by the laser device, since the laser device is no longer required by the external control unit 2. Laser light is no longer emitted.
19	The request and the program start of an external control unit 1 are processed. The acknowledgements are sent to the external control unit 1. The laser program is started. The laser device is assigned to the external control unit 1. The signals LASER_ASSIGNED, EXT_ACTIVATION_ACTIVE and PROG_ACTIVE are active. The laser program is performed. The LASER_READY signal remains active since in this mode an active laser program does not deactivate LASER_READY. The shutter remains open. The light path is switched over. The number of the laser program and the laser data are switched over. If the light path is set, laser light will again be emitted after about 60 ms.
20	The REQUEST_LASER signal is deactivated by the external control unit 1. The laser program is carried out as long as the PSTART_STATICAL signal is active. Another control unit can request the laser device and prevent the shutter from closing upon completion of the laser program. This allows a quick switch-over.
21	The laser light is emitted on the selected light path.

Tab. 394

Indicated times The times given do not include the times of the external bus systems as their basic times as well as cycle times are not known.

Parallel activation of a laser device

An external control unit has requested the laser device and started a laser program of the type "Standard" (with pulse triggering). During the laser processing, a different external control unit starts a laser program of the type "No pulse triggering", e.g. for positioning the PFO. In the timing diagram shown below, the two control units are connected with the laser device via a Multi PLC interface. The entered times are explained in the following table.



Example for parallel activation of a laser device Fig. TE-00129

Time of event	Definition
1	The external control 1 activates the input signal EXT_ACTIVATION for an external control of the laser device. The laser control unit activates the output signal EXT_ACTIVATION_ACTIVE.
2	The external control unit 1 activates the input signal LASER_ON. The laser control unit activates the output signal LASER_IS_ON.
3	The laser device is requested by the external control 1 using the active REQUEST_ LASER input signal. The laser control unit activates the output signals LASER_ASSIGNED and LASER_READY. The laser device is assigned to the external control unit 1 and the laser is ready.
4	The external control unit 1 sends the number of a laser program of the type "Standard", example: 5. The laser control unit shortly deactivates the output signal LASER_READY and activates the signal again.
5	The external control unit 1 starts the laser program (input signal PSTART_DYN).
6	The laser device activates the output signal PROG_ACTIVE and deactivates the output signal LASER_READY. If the central shutter is closed, it is opened via a command in the laser program.
7	Laser light is emitted via the preset light path.
8	The laser program is finished. The laser control unit deactivates the output signal PROG_ACTIVE and activates the output signal PROG_COMPLETED for approx. 6 s. No more laser light is emitted.
9	The laser control unit activates the signal LASER_READY.
10	The external control unit 1 deactivates the input signal REQUEST_LASER.
11	The laser control unit deactivates the signals LASER_ASSIGNED and LASER_READY. The laser device is no longer assigned to the external control unit 1 and the laser is no longer ready.
21	The external control 2 activates the input signal EXT_ACTIVATION for an external control of the laser device. The laser control unit activates the output signal EXT_ACTIVATION_ACTIVE.
22	The external control unit 2 sends the number of a laser program of the type "No pulse triggering", e.g. for positioning of a PFO, example: 8. The laser control unit activates the output signal LASER_READY. The laser program of the type "No pulse triggering" can be executed although the laser is assigned to the external control unit 1 and the laser light is emitted.
23	The external control unit 2 starts the laser program of the type "No pulse triggering".
24	The laser control unit deactivates the output signal LASER_READY and activates the output signal PROG_ACTIVE.
25	The PFO is positioned.
26	The positioning of the PFO is complete. The laser control unit deactivates the output signal PROG_ACTIVE and activates the output signal PROG_COMPLETED for approx. 6 s.
27	The laser control unit activates the signal LASER_READY.

Explanation of the times with parallel activation Tab. 395

Notes

- If an external control unit without assignment attempts to start a laser program of the type "Standard", this laser program is not executed. No message is displayed.
- An active laser program is cancelled if, during the program change, the laser program type changes as follows:

  - From "Standard" to "No pulse triggering".

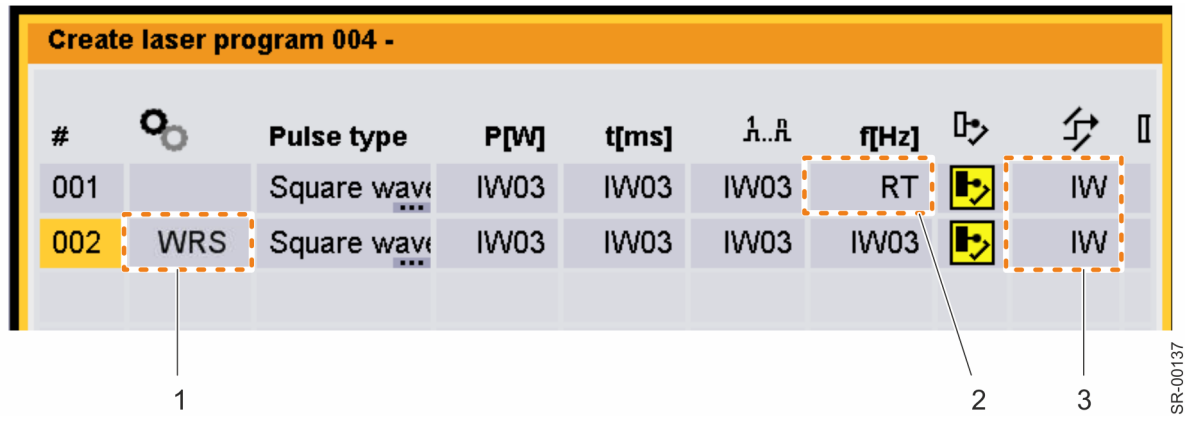
  - From "No pulse triggering" to "Standard".
- If the assigned control unit executes a laser program of the type "No pulse triggering", the symbol Program active is not displayed in the TruControl interface.
- From TruControl version 1.82.0, a laser program of the type "No pulse triggering" can be started in parallel with a laser program of the type "Standard" also during image processing with the two laser program functions WIP (Wait for Image Processing) and WIPJ (Wait for Image Processing Job).

  Further information:
  Operator's manual "VisionLine", Doc-no. 22-50-12-A0-CR
  Interface description "VisionLine Interface – TruControl 1000 Ethernet interface for image processing", doc-no. 12-12-40-A4-BA.

Using Input words for laser parameters

The input words to be interpreted as laser parameters by the fieldbus or OPC are defined in laser programs. For this purpose, the input word IW.. is entered in the laser program instead of the value of a laser parameter. This input word will then be used to transfer the laser parameter value.

In the following example, the values for pulse power, pulse duration, number of pulses and frequency are defined by means of the input words IW3 ... IW6. The light path no. is defined via IW (bit 15 ... bit 8 of the input word 1).



1 Function: real-time start 2 External frequency 3 Light path no.

Example: Laser program with laser parameters from the field bus or OPC Fig. SR-00137

The example shown above consists of 2 program lines:

- In program line 001:
  - the laser parameters are transferred using the input words IW03, IW04 and IW05.
  - is transmitted via IW (bit 15 ... bit 8 of the input word 1) of the light path no.
  - laser pulses are triggered externally via the **EXTERNAL_FREQUENCY** input signal of a real-time interface.
- In program line 002:
  - the laser parameters are transferred using the input words IW03, IW04, IW05 and IW06.
  - is transmitted via IW (bit 15 ... bit 8 of the input word 1) of the light path no.
  - laser pulses are triggered externally using the **REAL-TIME_START** signal of a real-time interface.

Laser parameters have to be transferred before starting a laser program if they are used as input words IW.. in a laser program.

If, for example, laser power parameters are transferred during the runtime of a laser program, it takes about 5 ms + transmis-

sion cycle of the external control unit (e.g. PLC) for the laser to be set to the new power values.

Further Information:

- Software manual "TruControl 1000", doc no. 10-06-80-A1-CR".
- Interface descriptions:
  - "Digital I/O interface of the CPX – parallel I/O, real-time, Sync I/O and decentralized I/O interface of the CPX V3 and CPX V4 standard",doc. no. 18-22-91-A2-BA
  - "Real-time and Sync I/O interface of the CPX V4 Basic",Doc.-No. 18-23-04-A1-BA
  - "Parallel I/O, Real-time and Sync I/O interface of the CPX",doc. no.: 18-21-85-A1-BA

Note

- If the central processor CPX V20 is used, e.g. with a laser device from the TruMicro 2000 series (FU11), laser device family 20-20-xx-xx, no real-time interfaces are available.

Messaging and RESET behavior

Notifications

There are four categories of messages which are output at the Multi PLC interface with the output word 4 and 5.

- Slot monitoring
  Signal **MONITORING_SLOT**, output word 2 bit 9
- Laser monitoring
  Signal **MONITORING**, output word 0 bit 5
- Slot fault
  Signal **FAULT_SLOT**, output word 2 bit 8
- Laser fault
  Signal **FAULT_LASER**, output word 0 bit 4

RESET

An external control must respond to messages with **RESET**, input word 0 Bit 8.

System parameter

These four message categories and RESET are influenced by the following system parameters:

- **PLC_Word_2_OutShutter**
  Assignment of output word 2
  - 0: Slot fault/-monitoring and status of the focusing optics
  - 1: Shutter position
- **LCU_GlobalLaserFaultReset**
  - 0: RESET for FAULT_LASER only possible from the assigned MPI slot.
  - 1: RESET for FAULT_LASER also possible from an unassigned MPI slot.
- **LCU_GlobalSlotReset**
  - 0: RESET for slot fault/monitoring only for the assigned MPI slot.
  - 1: RESET for slot fault/monitoring for all MPI slots
- **PLC_SeparatedSlotFault**
  - 0: with a slot fault, laser monitoring is also output.
  - 1: with a slot fault, no laser monitoring is output.

Status indicators

The following symbols show monitoring and fault messages in the status bar of TruControl 1000.

	Slot fault
	Laser monitoring
	Laser fault

Tab. 396

Slot monitoring

If slot monitoring is used, the message code (output words 4 and 5) and the output signal MONITORING_SLOT are output only at the relevant MPI slot.

Slot monitoring is output, for example when resetting REQUEST_LASER as long as the signal PROGRAM_START is still set.

If additional slot-/laser monitoring follow, then these are output at the Multi PLC interface.

Slot monitoring must be reset using RESET at the relevant MPI slot, irrespective of the assignment.

If the system parameter LCU_GlobalSlotReset = 1, all slot monitoring of the assigned MPI slot can be reset by means of RESET.

Laser monitoring

If laser monitoring is used, the message code (output word 4 and 5) and the output signal MONITORING is output at all MPI slots.

Laser monitoring is output, for example, if the level of the cooling water in the tank is too low.

If additional slot-/laser monitoring follow, then these are output at the Multi PLC interface.

If the system parameter LCU_GlobalLaserFaultReset = 0, a laser monitoring must be reset at the assigned MPI slot by means of RESET.

If the system parameter LCU_GlobalLaserFaultReset = 1, a laser monitoring must be reset at each assigned MPI slot by means of RESET.

Slot fault

If a slot fault occurs, the message code (output word 4 and 5) and the output signals FAULT_SLOT, MONITORING_SLOT and MONITORING are output only at the relevant MPI slot.

A slot fault is output, for example, if an attempt is made to start a non-existent laser program.

If additional slot-/laser monitoring or slot faults occur, then these are not output at the Multi PLC interface.

If the system parameter LCU_GlobalSlotReset is 0, a slot fault must be reset at the relevant MPI slot by means of RESET, irrespective of the assignment!

If the system parameter LCU_GlobalSlotReset = 1, all slot faults of the assigned MPI slot can be reset by means of RESET.

Laser fault

If a laser fault occurs, the message code (output words 4 and 5) and the output signals FAULT_LASER and MONITORING are output at all MPI slots.

A laser fault is output, for example, if an attempt is made to start a laser program which contains a deleted light path.

If additional slot-/laser monitoring or
slot-/laser faults occur, then these are not output at the Multi PLC interface.

If the system parameter LCU_GlobalLaserFaultReset = 0, a laser fault must be reset at the assigned MPI slot by means of RESET.

If the system parameter LCU_GlobalLaserFaultReset is 1, a laser fault can be reset at every MPI slot by means of RESET irrespective of the assignment!

Recommended system settings

We recommend configuring the system as follows:

Setting	System parameter
Output word 2 Output of the signals slot fault and monitoring	**PLC_Word_2_OutShutter** = 0
Reset of a slot fault/monitoring only for an MPI slot	**LCU_GlobalSlotReset** = 0
Reset of a laser fault only for the assigned MPI slot	**LCU_GlobalLaserFaultReset** = 0
Separation between laser fault and slot fault	**PLC_SeparatedSlotFault** = 1
Programmable output words 4 and 5	Default setting, i.e., the message code is output.
MPI settings output mask (see "Output mask", pg. 154)	**PROG_CANCELLED** = 0 **PILOT_LASER_IS_ON** = 0

Tab. 397

Save energy with Energy Save Concept or optimized standby modes 1-4

The Energy Save Concept allows the laser devices to be switched to different energy-saving states in different ways:

- PROFIenergy for ProfiNet.
- Pause concept for Multi PLC interfaces.

Which ways are available depends on the type of laser device and the configuration.

If the Energy Save Concept is not available, there are, depending on the laser device type and the configuration, up to four sleep modes.

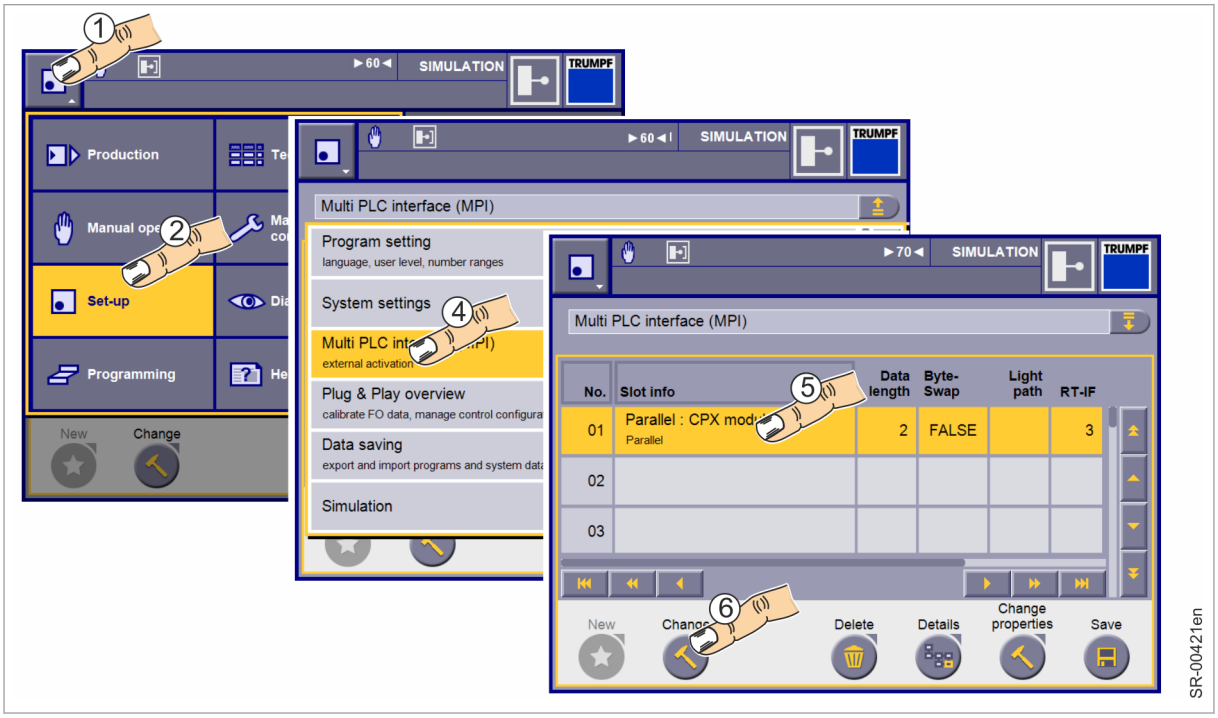
Further information: Technical information "Energy Save Concept", Doc.-no. 21-A0-71-CP.

7.4 Configuring interfaces

An OPC interface is a software interface only. The parameters of an OPC interface can be set using operating software.

Setting interface parameters

With user level 60 and higher, the parameters of an OPC interface can be set using the dialog: >*Setup* >*Multi PLC interface (MPI)* >*OPC*.



Selecting dialog for setting the interface parameters Fig. SR-00421

The following table list the settings possible for the OPC interface.

Parameter	Functions	Default	Values
File length	**Setting the data length of the interface.** The data length of an OPC interface can be defined.	4	TruControl 1000 version < 1.84.0: 3 ... 16 TruControl 1000 version ≥ 1.84.0: 3 ... 64[1)]
Byte swap	**Setting Byte swap.** The data are interpreted and transmitted by the laser control unit word by word. The data bytes can be swapped within the data words. **FALSE:** Default setting The two data bytes of the data word are not swapped. **TRUE:** The two data bytes of the data word are swapped. This setting is required, for example for the external control units with Intel processors (Siemens PLC). **Note:** If you notice that the data does not arrive at the expected bit positions, you should change the "byte swap" value.	FALSE	TRUE, FALSE
Light path	**Assigning a light path.** A light path is permanently assigned to an OPC interface if a fieldbus-coded control (TLN-fc) of the laser device is used. The coding of the light path at the ILN interface is not required for TLN-fc.	No allocation	No assignment, 0, 1, ..
Real-time interface[2)]	**Selecting a real-time interface.** A free real-time interface of the CPX can be assigned to an OPC interface. Pin assignment and signals of the real-time interfaces are described in the following documents: ▪ Interface description "Digital I/O interface of the CPX – Parallel I/O, Real-time, Sync I/O and Decentralized I/O interface of the CPX V3 and CPX V4 standard", doc. no.: 18-22-91-A2-BA ▪ Interface description "Real-time and Sync I/O interface" for CPX V4 Basic, doc. no. 18-23-04-A1-BA. ▪ Interface description "Parallel I/O, Real-time and Sync I/O interface of the CPX", doc. no.: 18-21-85-A1-BA	None	None, 1, 2, 3
MFO	**Assigning the manual focusing optics.** An OPC interface can be permanently assigned to an MFO.	None	None, 1 ... 8
Input mask	**Input mask** Description: (see "Input mask", pg. 146)	0xFFFF	0x1800 - 0xFFFF
1) Not available for the following CPX variants: ▪ CPX V1 standard, 18-21-85-A1 ▪ CPX V1 small, 18-22-21-A1 ▪ CPX V2 standard, 18-21-85-A3 ▪ CPX V2 small, 18-21-85-A4 2) If the central processor CPX V20 is used, e.g. with a laser device from the TruMicro 2000 series (FU11), laser device family 20-20-xx-xx, no real-time interfaces are available.			
Output mask	**Output mask** Description: (see "Output mask", pg. 154)	0xB707	0x0000 - 0xFFFF

Parameter	Functions	Default	Values
Inverting mask	**Inverting mask** Description: (see "Inverting mask", pg. 146)	0x0000	0x0000 - 0xE7BF
Priority	**Priority of the MPI slot** This parameter defines the priority of the MPI slot for the assignment, (see "LASER_ASSIGNED", pg. 200). The higher the value of the parameter, the higher is the priority of the MPI slot.	3	0 ... 255
POWER_GATE	**Configuring POWER_GATE** The source and the signal level for the **POWER_GATE** signal can be set: ▪ Fieldbus: High active Source: Fieldbus or OPC input word 14, bit 2, signal level High active ▪ Fieldbus: Low active Source: Fieldbus or OPC input word 14, bit 2, signal level Low active ▪ RTI: Low active if RTI_LEVEL_SETTING_x = 2[1] Source: PULSE_INTERLOCK signal of the real-time interface[2] x[1] ▪ RTI: High active, if RTI_LEVEL_SETTING_x = 1[1] Source: PULSE_INTERLOCK signal of the real-time interface[2] x[1]	Not active	Not active, Feldbus: High active, Feldbus: Low active, RTI: Low active (RTI_LEVEL_SETTING_x = 2[1]) RTI: High active (RTI_LEVEL_SETTING_x = 1[1])
	Further information: (see "POWER_GATE", pg. 198).		
Energy Save Concept	Setting the Energy Save Concept These parameters define how the Energy Save Concept is used.		
	Energy control	MPI input / output	None MPI input / output
	Access control	Direct	Direct Shared
	For further information, see the technical information "Energy Save Concept", doc. no. 21-A0-71-CP.		
Sensor slot[3]	If "TRUE", this slot is being used for communication between the connected third-party device and an external control, e.g. robot control connected to another MPI slot.	FALSE	FALSE TRUE
Assigned sensor slot[3]	Specifies the slot number of the third-party device with which the external control can communicate directly via the Multi PLC interface, e.g. robot control.	None	None, 1 - 32

1) x: Number of the real-time interface.
RTI: Real-time interface

2) If the central processor CPX V20 is used, e.g. with a laser device from the TruMicro 2000 series (FU11), laser device family 20-20-xx-xx, no real-time interfaces are available.

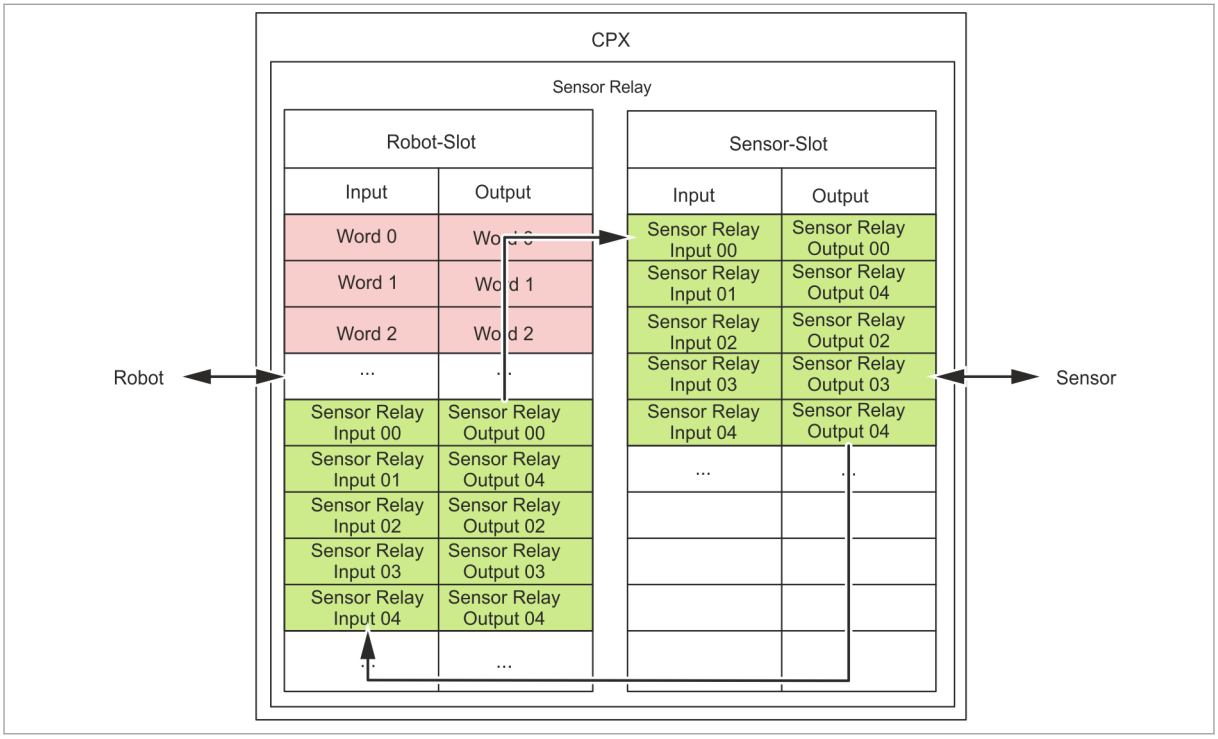
3) Further information: (see "Configuring the sensor relay", pg. 266).
Diagnostics: (see "Diagnostics Sensor relay", pg. 268).

Tab. 398

Configuring the sensor relay

If an external control, e.g. a robot control with a third-party device, wants to communicate via the multi-PLC interfaces, a sensor relay can be set up.

A sensor slot is assigned to the MPI slot on the external control, see "Assigned sensor slot" parameter. The MPI slot on the third-party device is set up as a sensor slot, see "Sensor slot" parameter. The parameters "Sensor Relay Input x Words" and "Sensor Relay Output x Words" must be configured in the input and output words of the interface parameters, x = 9, 16, 32 or 48. Diagnostics: (see "Diagnostics Sensor relay", pg. 268).



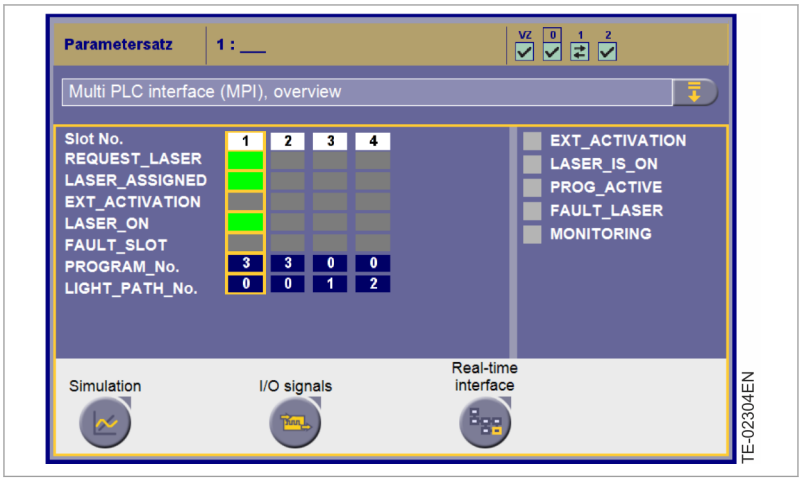
Sensor relay

Fig. 142468

7.5 Testing options

Multi-PLC interface (MPI), overview

The dialog >*Produce* >*Multi-PLC interface (MPI), overview* can be displayed from any user level. The status of important signals, the pulse energy, the average laser power and the utilization of the laser device are shown to the user per assigned processing station (slot).



Multi PLC Interface (MPI), overview dialog Fig. TE-02304

When selecting a slot, buttons are displayed in the context bar which enable activation of further functions. Depending on the configuration the following buttons are available:

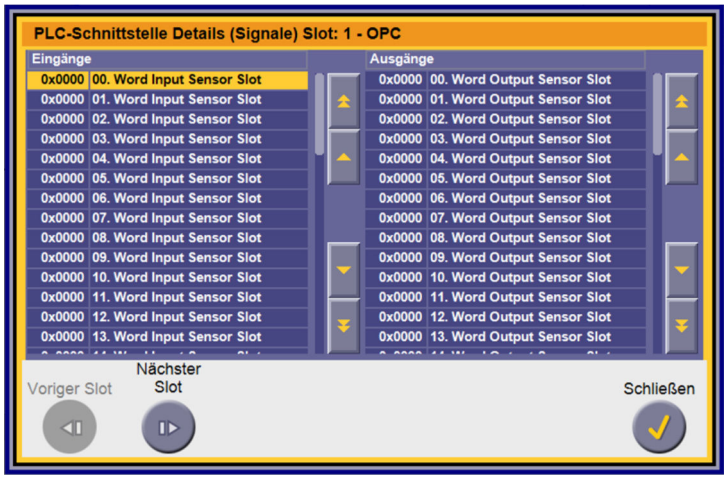
- Simulation
- I/O signals indicate the signals of the input and output words of the selected slot, e.g. the signals of a sensor relay (see "Diagnostics Sensor relay", pg. 268).
- Real-time interface.
- Manual focusing optics (option).

For further information: see software manual „TruControl 1000“, doc.-no. 10-06-80-A1-CR, chapter 5, „Diagnostics“.

Diagnostics Sensor relay

If a multi-PLC interface is defined as a sensor slot, the current values of the input and output words of the relay of the relay can be displayed in the dialog >*PLC interface details (signals)*.....

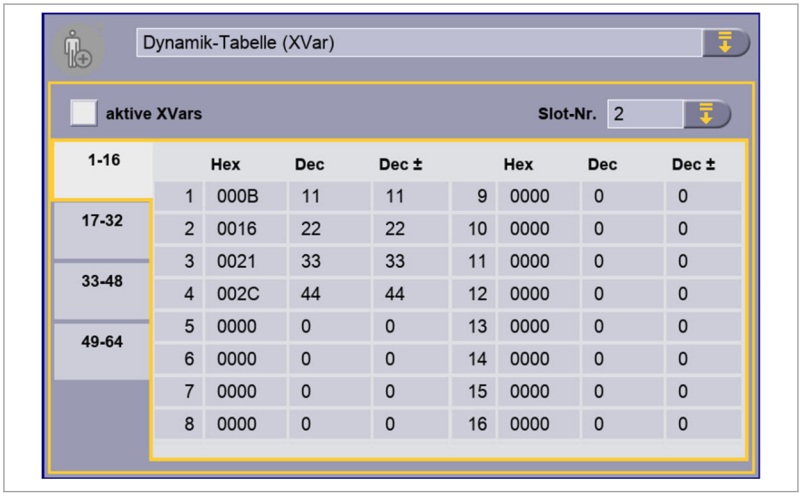
The dialog **PLC interface details (signals) ...** is displayed in the dialog >*Production* >*Multi-PLC interface (MPI), overview* by pressing *I/O signals*:



Sensor relay diagnostics, PLC interface details (signals) slot 1 - OPC Fig. 142469

Diagnostics Dynamic table (XVar)

The diagnostics of the dynamic table (XVar) can be displayed in the dialog >*Diagnostics* >*Dynamic table (XVar)* with a user level ≥ 60 .
The XVar tables of the individual slots are active if the VL parameters "PLC Output (CV) X Words" are used in the output words of the interface parameters.



Dynamic table diagnostics (XVar) Fig. 142470

- If XVar is transferred via OPC:
  - If no slot is assigned, the XVar will only be entered in the "active" XVar table.
  - If a slot is assigned, the XVars are entered in the "active" XVar table and the XVar table of the slot.
- If XVar is transferred from VisionLine:
  - Regardless of the assignment, the XVar is written into the XVar table of the slot which started the VL.
  - If the requesting slot is assigned, the XVar is also written into the "active" XVar table.
- If the assignment changes:
  - If a slot is assigned, the XVar of the slot's XVar table is copied to the "active" XVar table.
- If a laser program is started, the XVar is reset in the interface parameter "PLC output (CV) x words". The XVar table of the slot is not deleted. As a result, the display of the XVar table of the slot shows different values than the interface parameter "PLC output (CV) x words".

8. Switching from OPC DA to OPC UA

For all application developers who have gathered experience with OPC DA, TRUMPF enables switching over to OPC UA without trouble:

- On the one hand, it is possible to only equip an OPC UA server with the familiar OPC DA features and to build on this layer-by-layer.
- On the other hand, it is possible to integrate OPC UA applications without trouble into existing OPC DA applications. What are called wrappers and proxies are used for this. Wrappers and proxies form a bridge between the OPC UA and OPC DA applications.

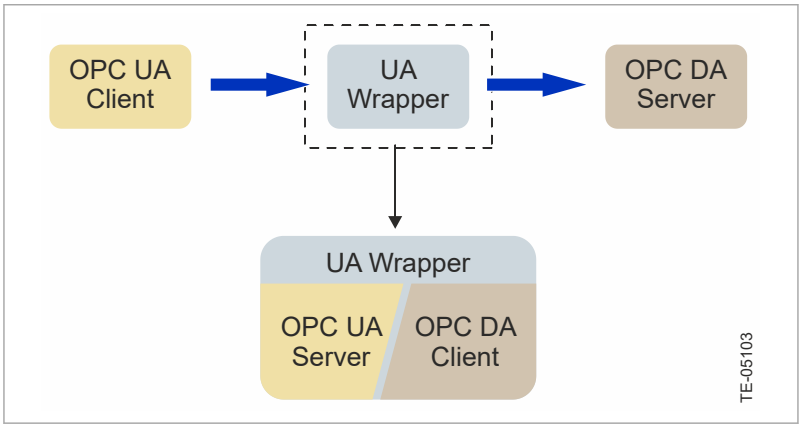
8.1 Wrapper and proxies

Wrapper

An OPC UA wrapper enables access to an OPC DA server using an OPC UA client.

An OPC UA server and an OPC DA client are integrated in this wrapper.

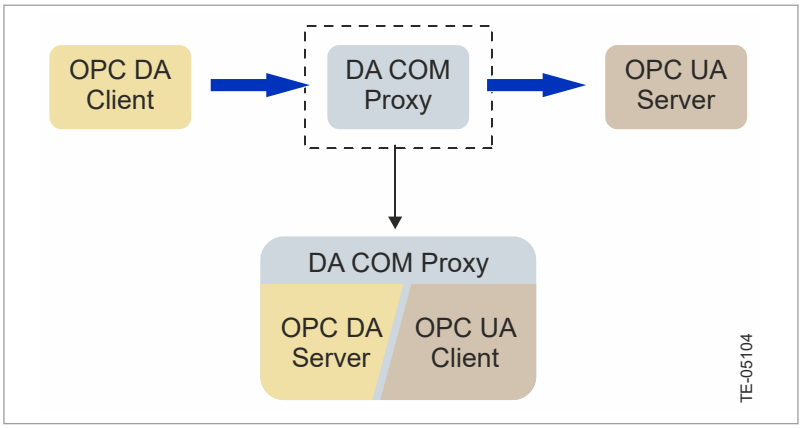
The following figure illustrates the connection between an OPC UA client and an OPC DA server:



OPC UA wrapper Fig. TE-05103

The OPC DA client in the wrapper creates a session, which is mapped as an OPC server object on the OPC DA server.

Proxies OPC DA COM proxies do the opposite: they connect an OPC DA client and an OPC UA server. A DA server and an UA client form the bridge.



OPC DA COM proxy Fig. TE-05104

8.2 UaGateway

With the UaGateway, the **Unified Automation** company provides a simple combination of both methods:

Installation To install the UaGateway, start the installation file and follow the instructions of the installation wizard. You can find a brief introduction with a description of the configuration in the file: '[installation path]\doc\quickstart.pdf'.

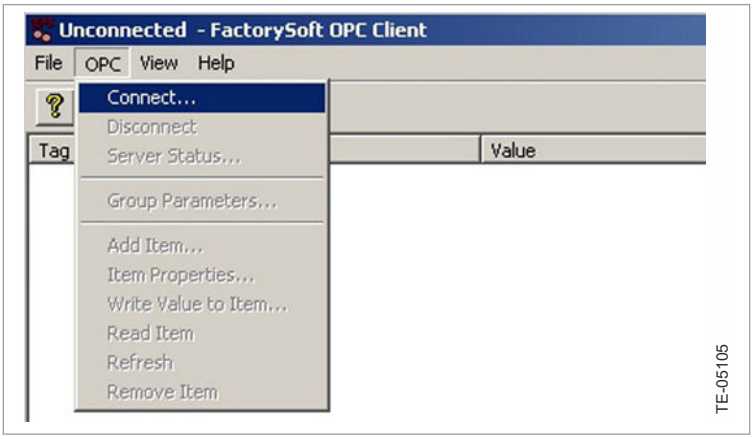
Use Once all configurations have been made, you can connect via UaGateway.

An example follows of a connection between an OPC UA server and the OPC DA client of **FactorySoft**.

Connecting an OPC UA server to the OPC DA client of FactorySoft

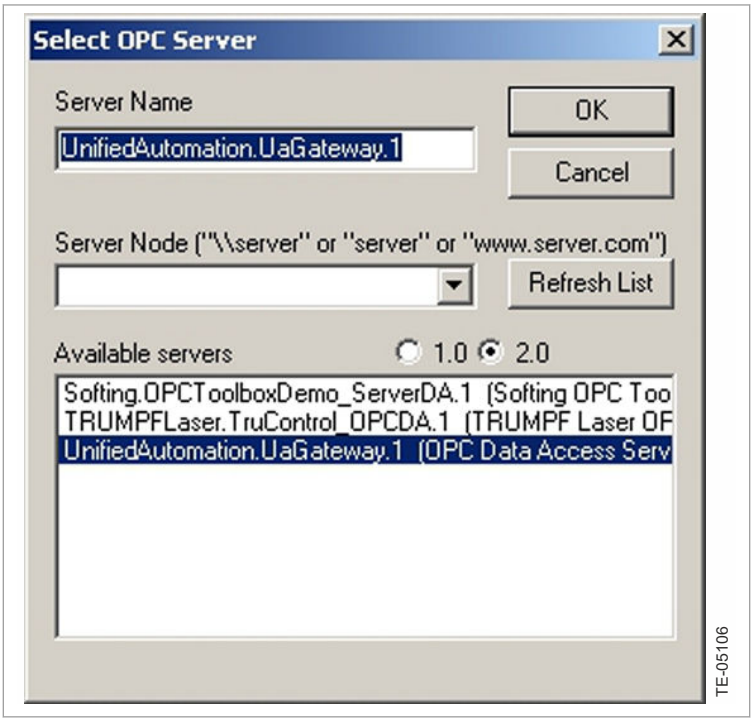
Conditions

- Configuration of the UaGateway on the desired OPC UA server (see quickstart.pdf)
- UaGateway must be started.



FactorySoft OPC client menu: OPC Connect Fig. TE-05105

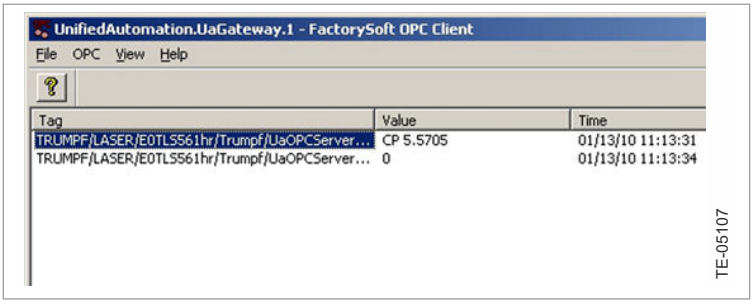
1. In the >*OPC* menu, select **Connect...**.



Select server Fig. TE-05106

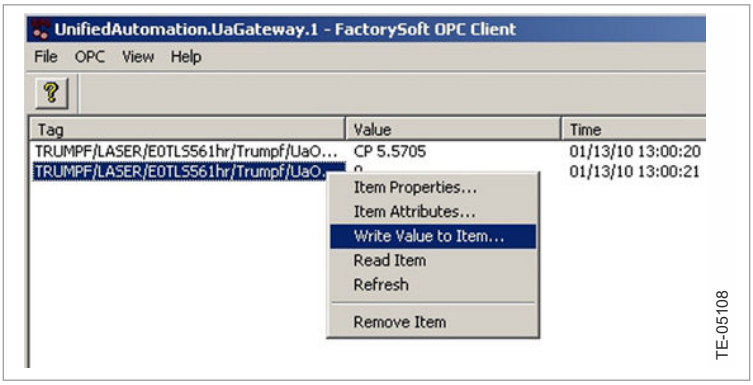
2. Select **UaGateway** as the server.
3. Select *Ok*.

If the connection is established successfully, items of the OPC UA server can be selected. This process does not differ from that for a connection to an OPC DA server.



Add item Fig. TE-05107

You can write values and use write and command variables in the *>Write Value to item* context menu.



Write values Fig. TE-05108