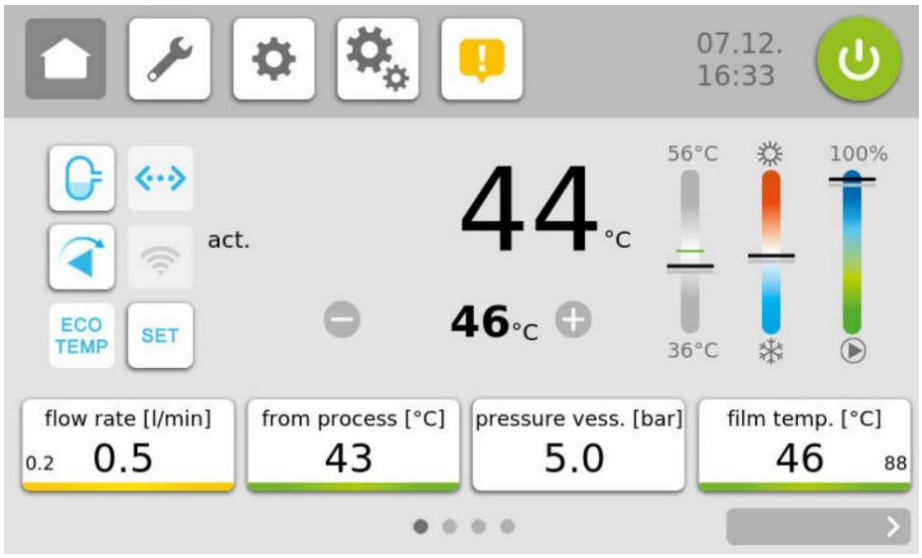
Beschreibung
Datenübertragung:

OPC-UA mit Euromap82.1

↕

Single Smart Controller – SSC



Inhaltsverzeichnis

SINGLE Temperiertechnik GmbH
Ostring 17-19
D - 73269 Hochdorf
FON +49 7153 3009 0 FAX: +49 7153 3009 50
www.single-temp.de

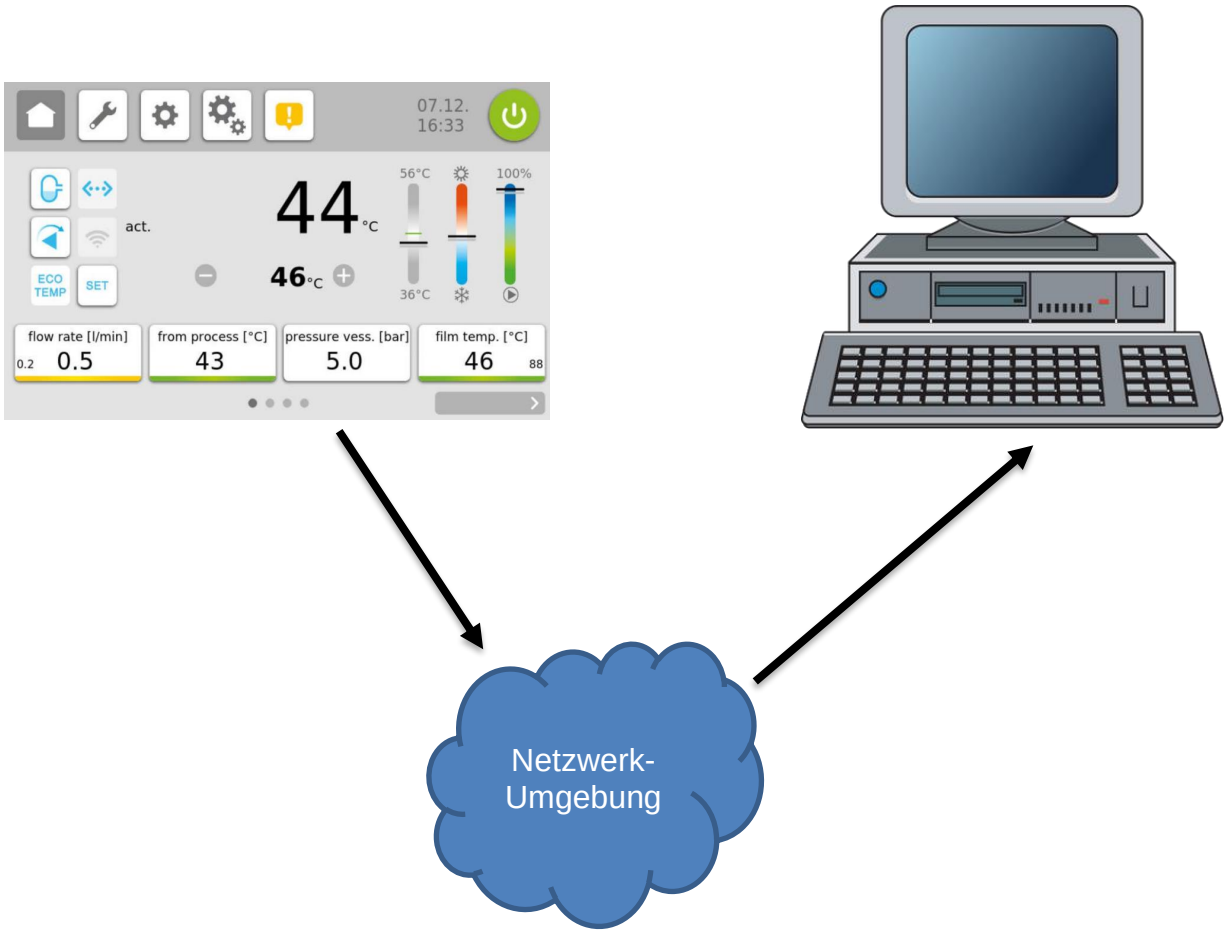
Vorwort

Diese Beschreibung wurde mit größtmöglicher Sorgfalt erstellt.
Die Angaben hierin gelten jedoch nicht als Zusicherung von Produkteigenschaften.
SINGLE Temperiertechnik GmbH übernimmt keine Haftung für Fehler.
SINGLE Temperiertechnik GmbH behält sich Änderungen, die dem technischen Fortschritt dienen, jederzeit vor.

1 OPC-UA über Ethernet Schnittstelle (TCP Format)

1.1 Einleitung

Das OPC-UA-Protokoll ist ein Protokoll das über Ethernet transportiert wird.
Da bei ist der Einsatzbereich nicht nur auf ein lokales Netzwerk beschränkt, sondern kann mit entsprechender Hardware auf andere Netzwerke wie. z.B. das Internet angewendet werden.



2 Port

Für die OPCUA Kommunikation wird im SSC der Port 4840 benutzt.

Beispiel:
opc.tcp://192.168.1.187:4840

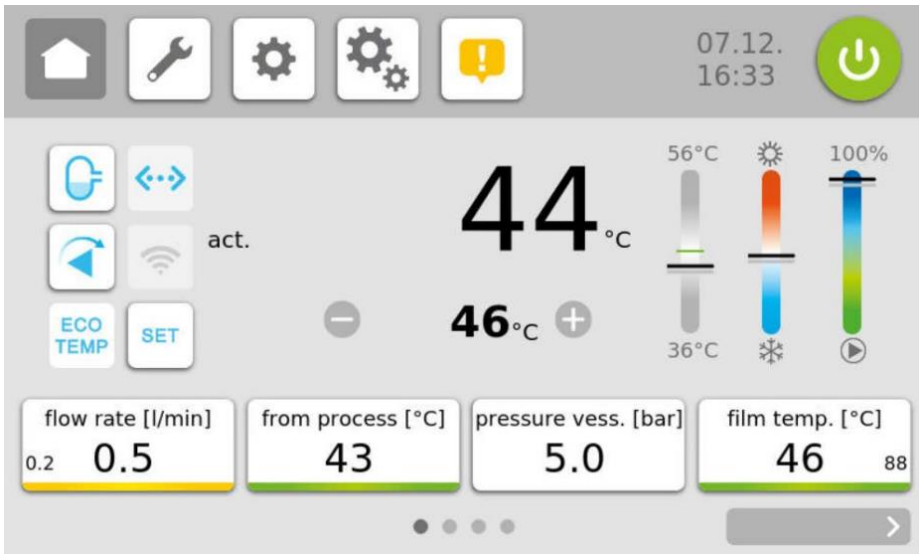
3 Anhang EUROMAP82.1

Description
Data Transmission:

OPC-UA with Euromap82.1

↕

Single Smart Controller – SSC



Content

SINGLE Temperiertechnik GmbH
Ostring 17-19
D - 73269 Hochdorf
FON +49 7153 3009 0 FAX: +49 7153 3009 50
www.single-temp.de

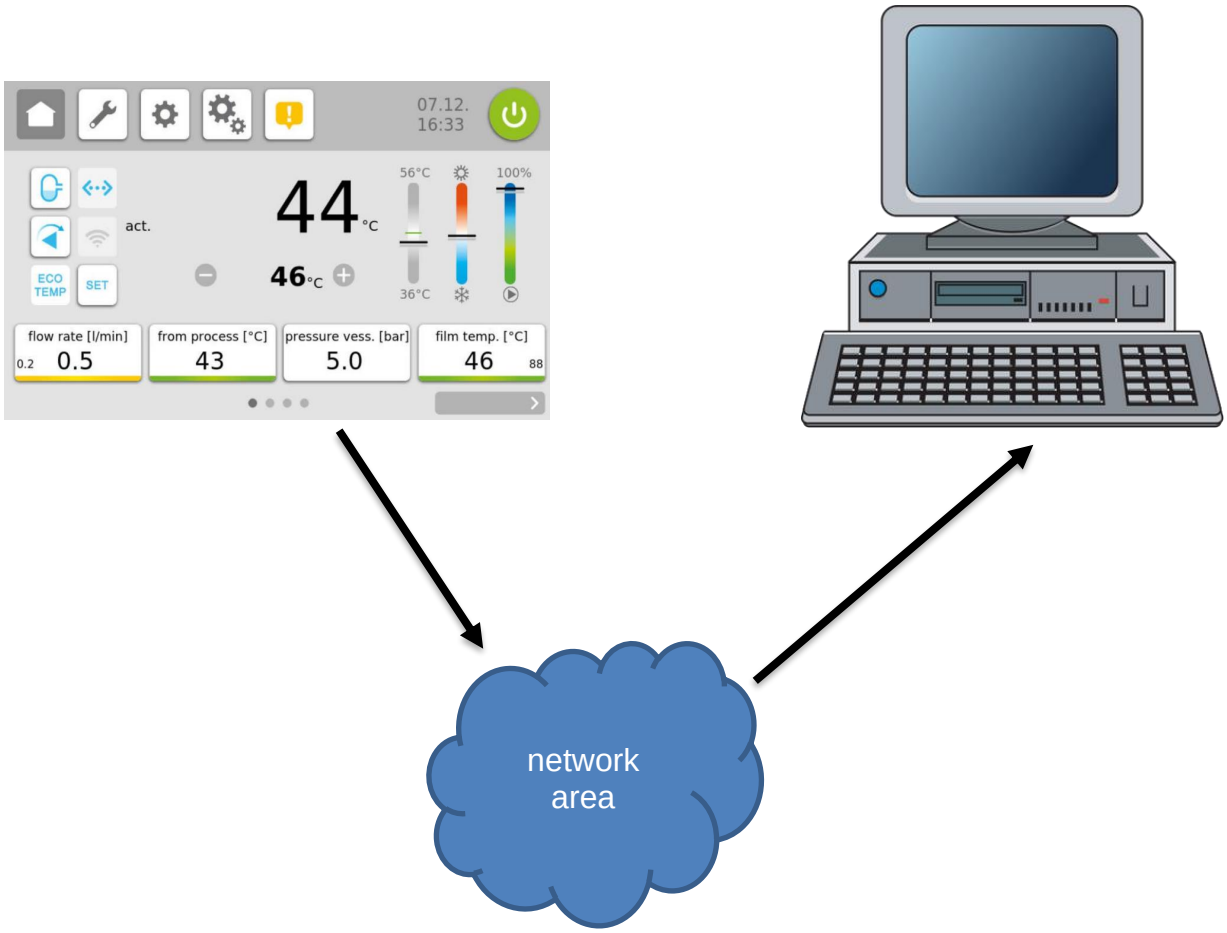
Preface

This description was prepared with the greatest possible care.
However, the information contained herein does not constitute a guarantee of product properties.
SINGLE Temperiertechnik GmbH assumes no liability for errors.
SINGLE Temperiertechnik GmbH reserves the right to make changes in the interest of technical progress at any time.

1 OPC-UA over Ethernet Interface (TCP Format)

1.1 Introduction

The OPC UA protocol is a protocol that is transported via Ethernet.
Since the range of application is not limited to a local network, it can be applied to other networks such as the Internet with appropriate hardware.



2 Port

The SSC uses port 4840 for OPCUA communication.

Sample:
opc.tcp://192.168.1.187:4840

3 Attachment EUROMAP82.1

European Plastics and Rubber Machinery

EUROMAP 82.1	OPC UA interfaces for plastics and rubber machinery – Peripheral devices – Temperature control devices

28 January 2019
27 pages

This EUROMAP recommendation was prepared by the Technical Commission of EUROMAP.

www.euromap.org

EUROMAP Technical Commission
c/o VDMA • FV KuG

Lyoner Str. 18
DE 60528 Frankfurt am Main

Phone + 49 (69) 6603-1833
Fax + 49 (69) 6603-2833
E-Mail:euromap@vdma.org

History

Date	Changes
22 February 2018	Release Candidate RC1.00 published
10 October 2018	Release Candidate RC1.00.5 published
28 January 2019	Release 1.00 published

Contents Page

1 Introduction

1.1 Scope and Application

EUROMAP 82.1 describes the interface for temperature control devices (TCD) for data exchange via OPC UA. The target of EUROMAP 82.1 is to provide a standardized interface for TCD from different manufacturers to ensure compatibility.

The following functionalities are covered:

- General information about the temperature control device
- Status information
- Process data

Safety related signals like emergency stop are not included.

1.2 References

Short name	Title	Version
OPC UA Part 1	OPC Unified Architecture – Part 1: Overview	1.04
OPC UA Part 2	OPC Unified Architecture – Part 2: Security Model	1.03
OPC UA Part 3	OPC Unified Architecture – Part 3: Address Space Model	1.04
OPC UA Part 4	OPC Unified Architecture – Part 4: Services	1.04
OPC UA Part 5	OPC Unified Architecture – Part 5: Information Model	1.04
OPC UA Part 6	OPC Unified Architecture – Part 6: Mappings	1.04
OPC UA Part 7	OPC Unified Architecture – Part 7: Profiles	1.04
OPC UA Part 8	OPC Unified Architecture – Part 8: Data Access	1.04
OPC UA Part 9	OPC Unified Architecture – Part 9: Alarms and Conditions	1.04
OPC UA Part 11	OPC Unified Architecture – Part 11: Historical Access	1.03
OPC UA Part 12	OPC Unified Architecture – Part 12: Discovery	1.03
OPC UA Part 100	OPC Unified Architecture – Part 100: OPC UA for Devices	1.01
EUROMAP 83	OPC UA interfaces for plastics and rubber machinery – General Type definitions	1.01

1.3 Abbreviations

TCD temperature control device

2 OPC UA Conformance Units and Profiles

This chapter defines the corresponding profiles and conformance units for the OPC UA Information Model for EUROMAP 82.1. *Profiles* are named groupings of conformance units. Facets are profiles that will be combined with other *Profiles* to define the complete functionality of an OPC UA *Server* or *Client.* The following tables specify the facets available for *Servers* that implement the EUROMAP 82.1 Information Model companion specification.

NOTE: The names of the supported profiles are available in the *Server Object* under *ServerCapabilities.ServerProfileArray*

Table 1 – *EUROMAP 82.1 Basic Server Facet* Definition

Conformance Unit	Description	Optional/ Mandatory
EUROMAP 82.1 Basic	Support of *TCD_InterfaceType* and all mandatory child elements giving information on the temperature control device itself, the current configuration and status.	M
Profile		
ComplexType Server Facet (defined in OPC UA Part 7)		M
Method Server Facet (defined in OPC UA Part 7)		M
BaseDevice_Server_Facet (defined in OPC UA Part 100)		M

Table 2 – *EUROMAP 82.1 Alarms Server Facet* Definition

Conformance Unit	Description	Optional/ Mandatory
EUROMAP 82.1 Alarms	Support of *HelpOffNormalAlarmType* providing error information. If this facet is supported and a client subscribes to the events, the server shall provide all errors via alarms in addition to the error variables included in the *OperationType*	M
Profile		
A & C Alarm Server Facet (defined in OPC UA Part 7)		M

Table 3 – *EUROMAP 82.1 Maintenance Server Facet* Definition

Conformance Unit	Description	Optional/ Mandatory
EUROMAP 82.1 Maintenance	Support of *MaintenanceType* for device zones	M

3 Namespaces

3.1 Namespace and identifiers for EUROMAP 82.1 Information Model

This clause defines the numeric identifiers for all of the numeric *NodeIds* defined in this specification. The identifiers are specified in a CSV file with the following syntax:

```
<SymbolName>, <Identifier>, <NodeClass>
```

where the *SymbolName* is the *BrowseName* of a *Type Node* and the *Identifier* is the numeric value for the *NodeId*.

The *BrowsePath* for an *Instance Node* is constructed by appending the *BrowseName* of the instance *Node* to the *BrowseName* for the containing instance or type. An underscore character is used to separate each *BrowseName* in the path. Let's take for example, the *MachineInformationType ObjectType Node* which has the *ControllerName Property*. The **Name** for the *ControllerName InstanceDeclaration* within the *MachineInformationType* declaration is: *MachineInformationType_ControllerName*.

The *NamespaceUri* for all *NodeIds* defined here is http://www.euromap.org/euromap82_1/

The CSV released with this version of the specification can be found here:
http://www.euromap.org/files/Opc_Ua.EUROMAP82_1.1_00.NodeId.csv

NOTE: The latest CSV that is compatible with this version of the specification can be found here:
http://www.euromap.org/files/Opc_Ua.EUROMAP82_1.NodeId.csv

A computer processable version of the complete Information Model defined in this specification is also provided. It follows the XML Information Model schema syntax defined in OPC UA Part 6.

The Information Model Schema released with this version of the specification can be found here:
http://www.euromap.org/files/Opc_Ua.EUROMAP82_1.1_00.NodeSet2.xml

NOTE: The latest Information Model schema that is compatible with this version of the specification can be found here:

http://www.euromap.org/files/Opc_Ua.EUROMAP82_1.NodeSet2.xml

3.2 Namespace Metadata

Table 4 defines the namespace metadata for this specification. The *Object* is used to provide version information on the namespace and an indication about static *Nodes*. Static *Nodes* are identical for all *Attributes* in all *Servers*, including the *Value Attribute*. See OPC UA Part 5 for more details.

The information is provided as *Object* of type *NamespaceMetadataType*. This *Object* is a component of the *Namespaces Object* that is part of the *Server Object*. The *NamespaceMetadataType ObjectType* and its *Properties* are defined in OPC UA Part 5.

The version information is also provided as part of the ModelTableEntry in the UANodeSet XML file. The UANodeSet XML schema is defined in OPC UA Part 6.

Table 4 – NamespaceMetadata Object for this Specification

Attribute	Value		
BrowseName	Euromap82_1_NamespaceMetadata		
References	**BrowseName**	**DataType**	**Value**
HasProperty	NamespaceUri	String	http://www.euromap.org/euromap82_1/
HasProperty	NamespaceVersion	String	1.00
HasProperty	NamespacePublicationDate	DateTime	2019-01-28 08:00:00
HasProperty	IsNamespaceSubset	Boolean	False
HasProperty	StaticNodeIdTypes	IdType[]	{Numeric}
HasProperty	StaticNumericNodeIdRange	NumericRange[]	Null
HasProperty	StaticStringNodeIdPattern	String	Null

3.3 Handling of OPC UA namespaces

Namespaces are used by OPC UA to create unique identifiers across different naming authorities. The *Attributes NodeId* and *BrowseName* are identifiers. A node in the UA *Address Space* is unambiguously identified using a *NodeId*. Unlike *NodeIds*, the *BrowseName* cannot be used to unambiguously identify a node. Different nodes may have the same *BrowseName*. They are used to build a browse path between two nodes or to define a standard *Property*.

Servers may often choose to use the same namespace for the *NodeId* and the *BrowseName*. However, if they want to provide a standard *Property*, its *BrowseName* shall have the namespace of the standards body although the namespace of the *NodeId* reflects something else, for example the *EngineeringUnits Property*. All *NodeIds* of nodes not defined in this specification shall not use the standard namespaces.

Table 5 provides a list of mandatory namespaces used in a EUROMAP 82.1 OPC UA *Server*.

Table 5 – Namespaces used in a EUROMAP 82.1 Server

Namespace	Description	Use
http://opcfoundation.org/UA/	Namespace for *NodeIds* and *BrowseNames* defined in the OPC UA specification. This namespace shall have namespace index 0.	Mandatory
Local Server URI	Namespace for nodes defined in the local server. This may include types and instances used in a device represented by the server. This namespace shall have namespace index 1.	Mandatory
http://opcfoundation.org/UA/DI/	Namespace for *NodeIds* and *BrowseNames* defined in OPC UA Part 100. The namespace index is server specific.	Mandatory
http://www.euromap.org/euromap83/	Namespace for *NodeIds* and *BrowseNames* defined in EUROMAP 83. The namespace index is server specific.	Mandatory
http://www.euromap.org/euromap82_1/	Namespace for *NodeIds* and *BrowseNames* defined in this specification. The namespace index is server specific.	Mandatory
Vendor specific types and instances	A server may provide vendor specific types like types derived from *MachineType* or *MachineStatusType* or vendor specific instances of devices in a vendor specific namespace.	Mandatory

4 TCD_InterfaceType

4.1 TCD_InterfaceType Definition

This OPC UA *ObjectType* is used for the root *Object* representing a TCD with its subcomponents. It is formally defined in Table 6.

NOTE: To promote interoperability of *Client*s and *Server*s, all instantiated *Devices* shall be aggregated in an *Object* called “DeviceSet” (see OPC UA for Devices)

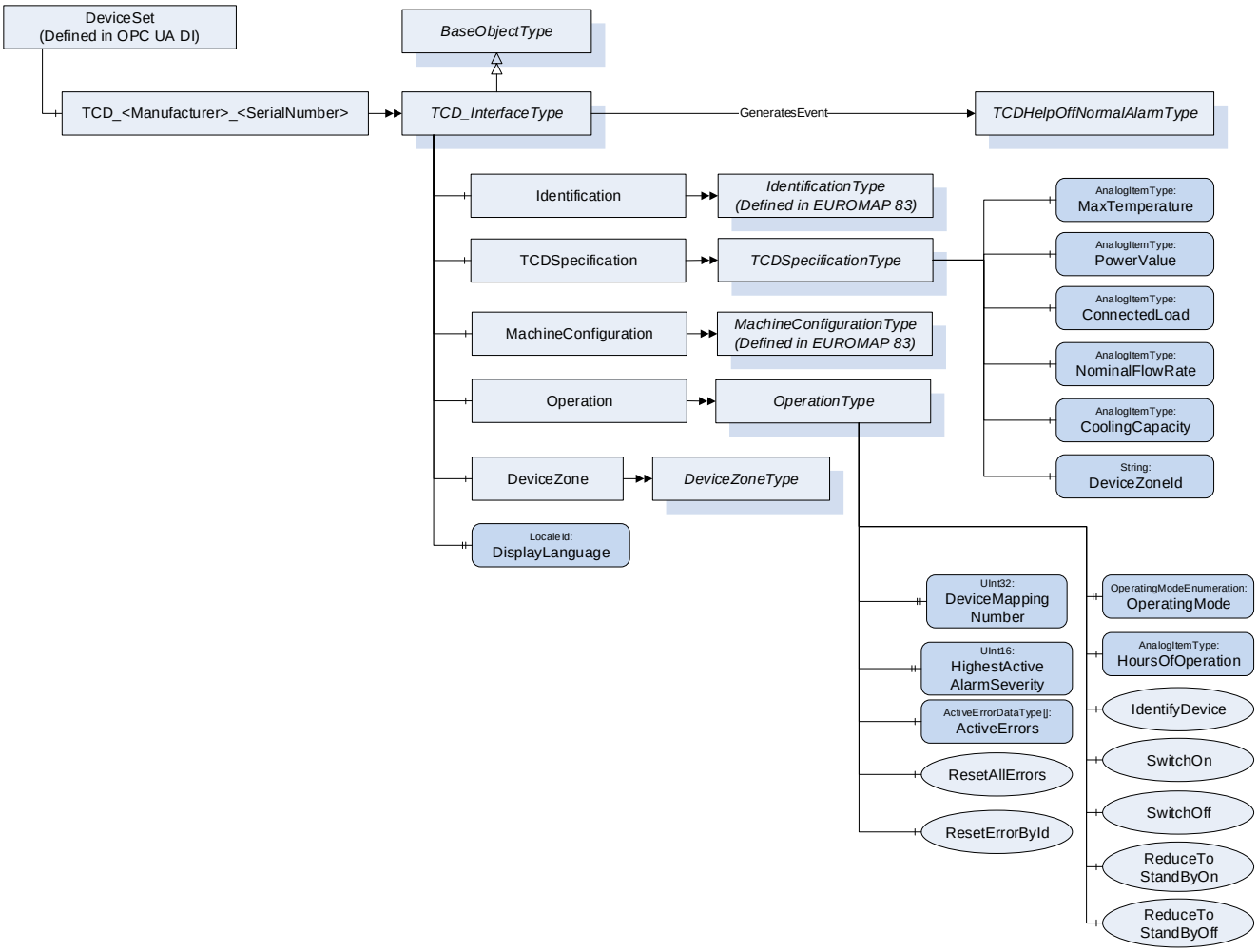


Figure 1 – TCD_InterfaceType Overview

Table 6 – TCD_InterfaceType Definition

Attribute	Value				
BrowseName	TCD_InterfaceType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling Rule**
Subtype of *BaseObjectType*					
HasComponent	Object	Identification		IdentificationType	M
HasComponent	Object	TCDSpecification		TCDSpecificationType	M
HasComponent	Object	MachineConfiguration		MachineConfigurationType	M
HasComponent	Object	Operation		OperationType	M
HasComponent	Object	DeviceZone		DeviceZoneType	M
HasProperty	Variable	DisplayLanguage	LocaleId	PropertyType	O, RW
GeneratesEvent	ObjectType	TCDHelpOffNormalAlarmType			

The *BrowseName* of the object instance shall be "TCD_<Manufacturer>_<SerialNumber>" Example: "TCD_HB-Therm_0123456".

Some TCD have several device zones (see 8). For these, the OPC UA server needs to create several instances of the TCD_InterfaceType. As the TCD has only one serial number, the BrowseNames of the objects shall include an extension e.g. “TCD_Wittmann_634_a” and “TCD_Wittmann_634_b”.

NOTE: The namespace of this *BrowseName* is the local server URI with namespace index 1 or a vendor specific namespace with server specific namespace index (see Table 5). The *BrowseNames* of the nodes below are in the namespace of the specification where used Type is defined.

Examples:

BrowseName	Namespace	Namespace index	Remarks
TCD_HB-Therm_0123456	Local Server URI or vendor specific namespace	1 or server specific	EUROMAP 82.1 only defines the *TCD_InterfaceType*. The instance is generated in the local server
↓			
Identification	http://www.euromap.org/euromap82_1/	server specific	The object *Identification* is a child of *TCD_InterfaceType* which is defined in EUROMAP 82.1
↓			
Manufacturer	http://www.euromap.org/euromap83/	server specific	The variable *Manufacturer* is a child of *IdentificationType* which is defined in EUROMAP 83.

BrowseName	Namespace	Namespace index	
TCD_HB-Therm_0123456	Local Server URI or vendor specific namespace	1 or server specific	EUROMAP 82.1 only defines the *TCD_InterfaceType*. The instance is generated in the local server
↓			
DeviceZone	http://www.euromap.org/euromap82_1/	server specific	The object *DeviceZone* is a child of *TCD_InterfaceType* which is defined in EUROMAP 82.1
↓			
ExternalChannels	http://www.euromap.org/euromap82_1/	server specific	The object *ExternalChannels* is a child *DeviceZoneType* which is defined in EUROMAP 82.1
↓			
ExternalChannel_1	Local Server URI or vendor specific namespace	1 or server specific	The objects for the extern channels are modelled as *OptionalPlaceholder*. The instances are server specific
↓			
PressureDifference	http://www.euromap.org/euromap82_1/	server specific	The object *PressureDifference* is a child *ExternalChannelType* which is defined in EUROMAP 82.1
↓			
ActualValue	http://www.euromap.org/euromap83/	server specific	The variable *ActualValue* is a child of *PressureDifference* which has the *MontoredItemType* as type definition which is defined in EUROMAP 83

BrowseName	Namespace	Namespace index	Remarks
TCD_HB-Therm_0123456	Local Server URI or vendor specific namespace	1 or server specific	EUROMAP 82.1 only defines the *TCD_InterfaceType*. The instance is generated in the local server
↓			
DeviceZone	http://www.euromap.org/euromap82_1/	server specific	The object *DeviceZone* is a child of *TCD_InterfaceType* which is defined in EUROMAP 82.1
↓			
ExternalChannels	http://www.euromap.org/euromap82_1/	server specific	The object *ExternalChannels* is a child *DeviceZoneType* which is defined in EUROMAP 82.1
↓			
NodeVersion	http://opcfoundation.org/UA/	0	The Property *NodeVersion* is defined in OPC UA

4.2 DisplayLanguage

With the *DisplayLanguage Property* the client can set the desired language on the user interface at the TCD. If the peripheral device does not support the configured language, it can keep the previous setting or use English as the default.

5 Identification

The *IdentificationType* for the identification of the device is defined in EUROMAP 83. All mandatory nodes shall be filled with valid values from the server.

The *DeviceClass Property* in the *Identification Object* shall have the value "Temperature Control Device".

6 TCDSpecificationType

This OPC UA *ObjectType* is used representing the basic specification of a TCD temperature control device with its subcomponents. It is formally defined in Table 7.

Table 7 – TCDSpecificationType Definition

Attribute	**Value**				
BrowseName	TCDSpecificationType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling Rule**
Subtype of *BaseObjectType*					
HasComponent	Variable	MaxTemperature	Int32	AnalogItemType	M, R
HasComponent	Variable	PowerValue	Double	AnalogItemType	M, R
HasComponent	Variable	ConnectedLoad	Double	AnalogItemType	M, R
HasComponent	Variable	NominalFlowRate	Double	AnalogItemType	M, R
HasComponent	Variable	CoolingCapacity	UInt32	AnalogItemType	O, R
HasProperty	Variable	DeviceZoneId	String	Property	O, R

6.1 MaxTemperature

Description: Defines the maximum working temperature of the TCD
Unit: °C or °F
Example: 160

6.2 PowerValue

Description: Power value, defines the heating capacity of the TCD with the rated voltage
Unit: kW
Example: 8

6.3 ConnectedLoad

Description: Connected load, defines the connections of the TCD (pump performance, heating capacity and performance of the remaining components)
Unit: kW
Example: 10.2

6.4 NominalFlowRate

Description: Nominal flow rate, defines the maximum achievable flow rate of the TCD
Unit: l/min, gal/min or ft³/min
Example: 45

6.5 CoolingCapacity

Description: Power value for cooling, defines the power value for cooling at temperature difference 60 K between cooling water and heat transfer medium
Unit: kW
Example: 30

6.6 DeviceZoneId

Description: As written in 4.1, for a TCD with several device zones, there shall be several instances of *TCD_InterfaceType* with one object *DeviceZone* each. In this case, the *DeviceZoneId* shall be used to identify the different device zones.
Example: “A”

7 OperationType

This *ObjectType* contains components which are necessary to operate the TCD. It is formally defined in Table 8.

Table 8 – OperationType Definition

Attribute	**Value**				
BrowseName	OperationType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling Rule**
Subtype of *BaseObjectType*					
HasProperty	Variable	DeviceMappingNumber	UInt32	PropertyType	M, RW
HasProperty	Variable	HighestActiveAlarmSeverity	UInt16	PropertyType	M, R
HasComponent	Variable	ActiveErrors	ActiveErrorDataType[]	BaseDataVariableType	M, R
HasComponent	Method	ResetAllErrors			O
HasComponent	Method	ResetErrorById			O
HasProperty	Variable	OperatingMode	OperatingModeEnumeration	PropertyType	M, R
HasComponent	Variable	HoursOfOperation	Double	AnalogItemType	O, R
HasComponent	Method	IdentifyDevice			O
HasComponent	Method	SwitchOn			M
HasComponent	Method	SwitchOff			M
HasComponent	Method	ReduceToStandByOn			O
HasComponent	Method	ReduceToStandByOff			O

7.1 DeviceMappingNumber

Description: Unique identifier/address/number for devices of the same *DeviceType* within a local network. Several peripheral devices of the same *DeviceType* can be connected to a machine. In most applications, the machine must map the connected peripheral devices to internal logical devices and zones in a fixed configuration (e.g. hot runner systems according to the wiring or temperature control devices according to the tubing).
The mapping shall be stable after reconnecting the devices and is therefore not possible via IP addresses, which can be assigned dynamically via DHCP. *DeviceMappingNumber* sets the mapping order of peripheral devices of the same type on the local network and is therefore of type *UInt32*.
Example: 1

7.2 HighestActiveAlarmSeverity

Description: Indication of the severity of the highest active alarm (0 = no active alarm – 1000 = possible error). Together with *ActiveErrors*, it provides a minimal error handling for devices without alarm support. However, the variable shall be filled even if alarms are supported.
Example: 400

7.3 ActiveErrors

Description: List of the active errors of the device. It provides a minimal error handling for devices without alarm support. However, the variable shall be filled even if alarms are supported. The *ActiveErrorDataType* is defined in EUROMAP 83. If there is no active error, the array is empty.

7.4 ResetAllErrors

Description: Method to reset all errors of the device.
Signature:
```
ResetAllErrors();
```

7.5 ResetErrorById

Description: Method to reset one error of the device.
Signature:
```
ResetErrorById(
   [in]    String     Id);
```

Table 9 –ResetErrorById Method Arguments

Argument	Description
Id	Id of the error, listed in *ActiveErrors*, that shall be reset.

Table 10 – ResetErrorById Method AddressSpace Definition

Attribute	Value				
BrowseName	ResetErrorById				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling Rule**
HasProperty	Variable	InputArguments	Argument[]	PropertyType	Mandatory

7.6 OperatingMode

Description: Actual operating mode of the TCD.

Table 11 – OperatingModeEnumeration Values

Value	Description
OTHER_0	Operating mode of the TCD is unknown
READY_TO_OPERATE_1	TCD is ready to operate (heating, pump and cooling are switched off)
NORMAL_OPERATION_2	TCD is running in normal operating mode
LEAK_STOPPER_3	TCD is running in leak stopper operating mode
MOULD_EVACUATION_4	TCD is carrying out a mould evacuation process
PRESSURE_RELIEF_5	TCD is carrying out a pressure relief process
COOLING_6	TCD is cooling down to *StandbyTemperature* and switch off
SAFETY_COOLING_7	TCD is cooling down to *SwitchingOffTemperature* and switch off
ECO_8	TCD is running in Eco operating mode (energy is saved via the reduced pump speed)
BOOST_9	TCD is running in Boost operating mode (pump runs at maximum possible speed)

7.7 HoursOfOperation

Description: Actual hours of operation
Unit: h
Example: 4586

7.8 IdentifyDevice

Description: The TCD on which this method is called shows itself by e.g. activation of a LED.
Signature: `IdentifyDevice ();`

7.9 SwitchOn

Description: Main switch method of the TCD for switching on.
OperatingMode shows the actual state of the device.
Signature: `SwitchOn ();`

7.10 SwitchOff

Description: Main switch method of the TCD for switching off.
OperatingMode shows the actual state of the device.
Signature: `SwitchOff ();`

7.11 ReduceToStandByOn

Description: Activate the cooling down function on the TCD followed by switching off.
OperatingMode shows the actual state of the device during the cooling down process.
Signature: `ReduceToStandByOn ();`

7.12 ReduceToStandByOff

Description: Deactivate the cooling down function on the TCD. If it is already in progress, it will be interrupted and the device changes back to the last selected operating mode.
Signature: `ReduceToStandByOff ();`

8 DeviceZoneType

The *DeviceZoneType* represents the functional main component of a TCD and is therefore mandatory.

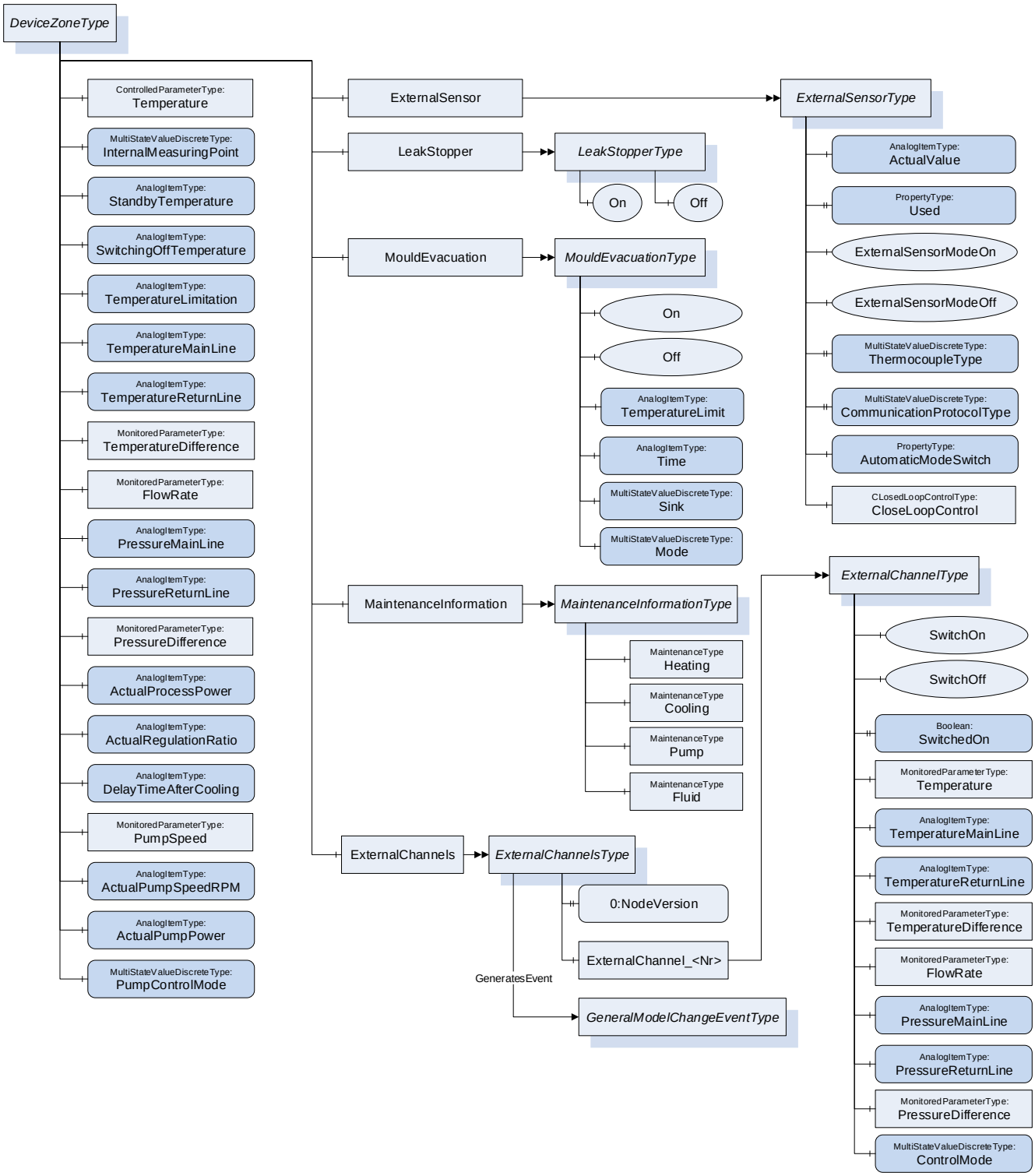


Figure 2 –DeviceZoneType Overview

Table 12 – DeviceZoneType Definition

Attribute	Value				
BrowseName	DeviceZoneType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling Rule**
Subtype of *BaseObjectType*					
HasComponent	Object	Temperature		ControlledParameterType	M
HasComponent	Variable	InternalMeasuringPoint	UInt16	MultiStateValueDiscreteType	O, RW
HasComponent	Variable	StandbyTemperature	Double	AnalogItemType	O, RW
HasComponent	Variable	SwitchingOffTemperature	Double	AnalogItemType	O, RW
HasComponent	Variable	TemperatureLimitation	Double	AnalogItemType	O, RW
HasComponent	Variable	TemperatureMainLine	Double	AnalogItemType	O, R
HasComponent	Variable	TemperatureReturnLine	Double	AnalogItemType	O, R
HasComponent	Object	TemperatureDifference		MonitoredParameterType	O
HasComponent	Object	FlowRate		MonitoredParameterType	O
HasComponent	Variable	PressureMainLine	Double	AnalogItemType	O, R
HasComponent	Variable	PressureReturnLine	Double	AnalogItemType	O, R
HasComponent	Object	PressureDifference		MonitoredParameterType	O
HasComponent	Variable	ActualProcessPower	Double	AnalogItemType	O, R
HasComponent	Variable	ActualRegulationRatio	Double	AnalogItemType	O, R
HasComponent	Variable	DelayTimeAfterCooling	Double	AnalogItemType	O, RW
HasComponent	Object	PumpSpeed		MonitoredParameterType	O
HasComponent	Variable	ActualPumpSpeedRPM	Double	AnalogItemType	O, R
HasComponent	Variable	ActualPumpPower	Double	AnalogItemType	O, R
HasComponent	Variable	PumpControlMode	UInt16	MultiStateValueDiscreteType	O, RW
HasComponent	Object	ExternalSensor		ExternalSensorType	O
HasComponent	Object	LeakStopper		LeakStopperType	O
HasComponent	Object	MouldEvacuation		MouldEvacuationType	O
HasComponent	Object	MaintenanceInformation		MaintenanceInformationType	O
HasComponent	Object	ExternalChannels		ExternalChannelsType	O

NOTE: The *MonitoredParameterType* is defined in EUROMAP 83. If the Parameter is not only monitored but controlled, the subtype *ControlledParameterType* can be used.

8.1 Temperature

Description: Setting and/or monitoring of the temperature in the main or return line (see *InternalMeasuringPoint*) or active external Sensor (*ExternalSensorModeOn*)
Unit: °C or °F
Example (for *ActualValue)*: 120

8.2 InternalMeasuringPoint

Description: This determines whether the temperature of the main or the return is to be controlled.

The *TypeDefinition* is *MultiStateValueDiscreteType*, so the *Properties EnumValues* and *ValueAsText* must be filled with the supported values out of Table 13.

Table 13 – Values for InternalMeasuringPoint

EnumValue	ValueAsText	Description
0	MAIN_LINE	Control of the forward flow temperature
1	RETURN_LINE	Control of the return flow temperature

8.3 StandbyTemperature

Description: The standby value temperature is approached with the Method *ReduceToStandByOn*. The TCD switches off.
Unit: °C or °F
Example: 35.0

8.4 SwitchingOffTemperature

Description: Defines the temperature to which the TCD must be cooled down before it switches off.
Unit: °C or °F
Example: 70.0

8.5 TemperatureLimitation

Description: This setpoint is for temperature limitation of the mould circuit e.g. to protect the connected tubes or the downstream water distribution system.
Unit: °C or °F
Example: 120

8.6 TemperatureMainLine

Description: Actual temperature in the main line.
Unit: °C or °F
Example: 100

8.7 TemperatureReturnLine

Description: Actual temperature in the return line.
Unit: °C or °F
Example: 105

8.8 TemperatureDifference

Description: Setting and/or monitoring of the temperature difference between return and main line. Positive if temperature in return line is higher than in main line.
Unit: °C or °F
Example (for *ActualValue*): 5

8.9 FlowRate

Description: Setting and/or monitoring of the flow rate.
Unit: l/min, gal/min or ft^3/min
Example (for *ActualValue*): 10.0

8.10 PressureMainLine

Description: Actual pressure in the main line (Pressure return line + pump pressure).
Unit: bar or lbf/in^2 (=psi)
Example: 6.5

8.11 PressureReturnLine

Description: Actual pressure in the return line.
Unit: bar or lbf/in² (=psi)
Example: 6

8.12 PressureDifference

Description: Setting and/or monitoring of the pressure difference between main and return line
Unit: bar or lbf/in² (=psi)
Example: 2.8

8.13 ActualProcessPower

Description: Actual calculated process performance (from the view of the TCD: heating = positive value, cooling = negative value)
Unit: kW
Example: - 2.3

8.14 ActualRegulationRatio

Description: Actual Regulation Ratio (heating = positive value, cooling = negative value)
Unit: %
Example: -0.15

8.15 DelayTimeAfterCooling

Description: Delay Time after cooling before switching off the TCD
Unit: min
Example: 2

8.16 PumpSpeed

Description: Setting and/or monitoring the speed of the pump in percent of maximum speed
Unit: %
Example (for *ActualValue*): 100

8.17 ActualPumpSpeedRPM

Description: Actual speed of the pump in revolutions per minute
Unit: min^{-1}
Example: 3000

8.18 ActualPumpPower

Description: Actual power of the pump in kW
Unit: kW
Example: 1.5

8.19 PumpControlMode

Description: Defines to which setpoint or function the pump is controlled. The *TypeDefinition* is *MultiStateValueDiscreteType*, so the *Properties EnumValues* and *ValueAsText* must be filled with the supported values out of Table 14.

Table 14 – Values for PumpControlMode

EnumValue	ValueAsText	Description
0	NORMAL	Normal Operation: fixed pump rotational speed
1	AUTO	Automatic adjustment of the pump rotational speed
2	SPEED	Speed controlled: the pump is controlled according to the specified nominal value *PumpSpeed*
3	FLOW	Flow rate controlled: the pump is controlled according to the specified nominal value *FlowRate*
4	TEMP_DIFF	Temperature difference controlled: the pump is controlled according to the specified nominal value *TemperatureDifference*
5	PRESS_DIFF	Pressure difference controlled: the pump is controlled according to the specified nominal value *PressureDifference*
6	BOOST	Boost mode: the pump is operated at the maximum possible rotational pump speed

8.20 ExternalSensorType

ExternalSensor is an optional component from *DeviceZoneType* and includes variables for the operation with an external temperature sensor. The temperature sensor is connected at the TCD directly or the value can come from the connected machine.

Table 15 – ExternalSensorType Definition

Attribute	Value				
BrowseName	ExternalSensorType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling Rule**
Subtype of *BaseObjectType*					
HasComponent	Variable	ActualValue	Double	AnalogItemType	M, RW
HasProperty	Variable	Used	Boolean	PropertyType	M, R
HasComponent	Method	ExternalSensorModeOn			M
HasComponent	Method	ExternalSensorModeOff			M
HasComponent	Variable	ThermocoupleType	UInt16	MultiStateValueDiscreteType	M, RW
HasComponent	Variable	CommunicationProtocolType	UInt16	MultiStateValueDiscreteType	M, RW
HasComponent	Variable	AutomaticModeSwitch	Boolean	PropertyType	O, RW
HasComponent	Object	ClosedLoopControl		ClosedLoopControlType	O

8.20.1 ActualValue

Description: Actual value of external temperature sensor
Unit: °C or °F
Example: 41.0

The value is only writeable if the *CommunicationProtocolType* is OPC_UA (value 3).

8.20.2 Used

Description: Return whether an external temperature sensor is used for control
Example: true

8.20.3 ExternalSensorModeOn

Description: Activate the mode where the external temperature sensor is used for temperature control

Signature: `ExternalSensorModeOn ();`

8.20.4 ExternalSensorModeOff

Description: Deactivate the mode where the external temperature sensor is used for temperature control

Signature: `ExternalSensorModeOff ();`

8.20.5 ThermocoupleType and CommunicationProtocolType

This two *Variables* are used to specify the type of connected external temperature sensor and the used communication protocol between the sensor and the control system of the TCD.

The *TypeDefinition* for both *Variables* is *MultiStateValueDiscreteType*, so the *Properties EnumValues* and *ValueAsText* must be filled with the supported values out of Table 16 and Table 17.

Table 16 – Values for ThermocoupleType

EnumValue	ValueAsText	Description
0	OTHER	Other sensor type
1	E	Type E sensor: NiCr-CuNi
2	J	Type J, L sensor: Fe-CuNi
3	K	Type K sensor: NiCr-Ni
4	N	Type N sensor: NiCrSi-NiSi
5	T	Type T sensor: Cu-CuNi
6	PT100	Pt 100-Sensor

Table 17 – Values for CommunicationProtocolType

EnumValue	ValueAsText	Description
0	OTHER	Other connection type
1	LOCAL	Communication integrated in the local control system (local input)
2	PROFIBUS	Values via Profibus
3	OPC_UA	Values via OPC UA
4	I2C	Values via I2C
5	CAN	Values via CAN

Which sensor types and protocols and combinations are supported is device dependent. Especially when the *CommunicationProtocolType* has the value 1 (LOCAL), the *ThermocoupleType* could be set to a fixed value by the TCD.

8.20.6 AutomaticModeSwitch

Setting whether switching to external sensor is performed automatically (TRUE) or manually (FALSE). TRUE is when the external sensor is plugged regulated to this and again switched automatically when unplugging the internal measurement site.

8.20.7 ClosedLoopControl

With this *Object* of C*losedLoopControlType* (defined in EUROMAP 83) the client can do settings for the closed loop control for the sensor.

8.21 LeakStopperType

LeakStopperType is an optional component of *DeviceZoneType* and is used for switching the leak stopper mode.

Table 18 – LeakStopperType Definition

Attribute	Value				
BrowseName	LeakStopperType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling Rule**
Subtype of *BaseObjectType*					
HasComponent	Method	On			M
HasComponent	Method	Off			M

8.21.1 On

Description: Activate the leak stopper mode (emergency operation in case of leaks in the system)
Signature: `On ();`

8.21.2 Off

Description: Deactivate the leak stopper mode
Signature: `Off ();`

8.22 MouldEvacuationType

MouldEvacuationType is an optional component of *DeviceZoneType* and includes parameters and nodes for mould evacuation.

Table 19 – MouldEvacuationType Definition

Attribute	Value				
BrowseName	MouldEvacuationType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling Rule**
Subtype of *BaseObjectType*					
HasComponent	Method	On			M
HasComponent	Method	Off			M
HasComponent	Variable	TemperatureLimit	Double	AnalogItemType	O, RW
HasComponent	Variable	Time	Double	AnalogItemType	O, RW
HasComponent	Variable	Sink	UInt16	MultiStateValueDiscreteType	O, RW
HasComponent	Variable	Mode	UInt16	MultiStateValueDiscreteType	O, RW

8.22.1 On

Description: Activate evacuation mode
Signature: `On ();`

8.22.2 Off

Description: Deactivate evacuation mode
Signature: `Off ();`

8.22.3 TemperatureLimit

Description: Temperature Limitation of the mould evacuation. TCD is cooled to this temperature first if necessary.
Unit: °C or °F
Example: 70

8.22.4 Time

Description: Duration of the mould evacuation.
Unit: s
Example: 45

8.22.5 Sink

Description: Defines where the medium is to be emptied
The *TypeDefinition* is *MultiStateValueDiscreteType*, so the *Properties EnumValues* and *ValueAsText* must be filled with the supported values out of Table 20.

Table 20 – Values for Sink

EnumValue	ValueAsText	Description
0	DRAIN	medium is passed into the cooling or system water outlet
1	TANK	medium is conducted in a separate outlet

8.22.6 Mode

Description: Defines how the medium is to be emptied.
The *TypeDefinition* is *MultiStateValueDiscreteType*, so the *Properties EnumValues* and *ValueAsText* must be filled with the supported values out of Table 21.

Table 21 – Values for Mode

EnumValue	ValueAsText	Description
0	PUMP	Evacuation by the pump
1	COMPRESSED_AIR	Evacuation with compressed air

8.23 MaintenanceInformationType

Information on the maintenance status of heating, cooling, pump and fluid.

Table 22 – MaintenanceInformationType Definition

Attribute	Value				
BrowseName	MaintenanceInformationType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling Rule**
Subtype of *BaseObjectType*					
HasComponent	Object	Heating		MaintenanceType	O
HasComponent	Object	Cooling		MaintenanceType	O
HasComponent	Object	Pump		MaintenanceType	O
HasComponent	Object	Fluid		MaintenanceType	O

The *MaintenanceType* is defined in EUROMAP 83.

8.24 ExternalChannelsType

This *ObjectType* is a container for the external channel(s). It is formally defined in Table 23.

Table 23 – ExternalChannelsType Definition

Attribute	Value				
BrowseName	ExternalChannelsType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling Rule**
Subtype of *BaseObjectType*					
HasProperty	Variable	0:NodeVersion	String	PropertyType	M, R
HasComponent	Object	ExternalChannel_<Nr>		ExternalChannelType	OP
GeneratesEvent	ObjectType	GeneralModelChangeEventType			

When instances for device zones are created, the *BrowseNames* shall be "ExternalChannel_<Nr>" (starting with 1).

8.25 ExternalChannelType

ExternalChannelType includes information for monitoring or controlling of external temperature, flow rate or pressure channels. (One zone of the TCD is split into several external channels).

Table 24 – ExternalChannelType Definition

Attribute	Value				
BrowseName	ExternalChannelType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling Rule**
Subtype of *BaseObjectType*					
HasComponent	Method	SwitchOn			O
HasComponent	Method	SwitchOff			O
HasProperty	Method	SwitchedOn	Boolean	PropertyType	O, R
HasComponent	Object	Temperature		MonitoredParameterType	O
HasComponent	Variable	TemperatureMainLine	Double	AnalogItemType	O, R
HasComponent	Variable	TemperatureReturnLine	Double	AnalogItemType	O, R
HasComponent	Object	TemperatureDifference		MonitoredParameterType	O
HasComponent	Object	FlowRate		MonitoredParameterType	O
HasComponent	Variable	PressureMainLine	Double	AnalogItemType	O, R
HasComponent	Variable	PressureReturnLine	Double	AnalogItemType	O, R
HasComponent	Object	PressureDifference		MonitoredParameterType	O
HasComponent	Variable	ControlMode	UInt16	MultiStateValueDiscreteType	O, RW

NOTE: The *MonitoredParameterType* is defined in EUROMAP 83. If the Parameter is not only monitored but controlled, the subtype *ControlledParameterType* can be used.

8.25.1 SwitchOn

Description: Switch method of the external channel for switching on.
SwitchedOn shows the actual state of the channel.

Signature: `SwitchOn ();`

8.25.2 SwitchOff

Description: Switch method of the external channel for switching off.
SwitchedOn shows the actual state of the channel.

Signature: `SwitchOff ();`

8.25.3 SwitchedOn

Description: Information if the external channel is switched on. If the methods *SwitchOn* and *SwitchOff* are provided, also this Property shall be available.
Example: TRUE

8.25.4 Temperature

Description: Setting and/or monitoring of the temperature
Unit: °C or °F
Example (for *ActualValue)*: 120

8.25.5 TemperatureMainLine

Description: Actual temperature in the main line
Unit: °C or °F
Example: 120

8.25.6 TemperatureReturnLine

Description: Actual temperature in the return line
Unit: °C or °F
Example: 115

8.25.7 TemperatureDifference

Description: Setting and/or monitoring of the temperature difference between return and main line. Positive if temperature in return line is higher than in main line.
Unit: °C, K or °F
Example (for *ActualValue*): 5

8.25.8 FlowRate

Description: Setting and/or monitoring of the flow rate.
Unit: l/min, gal/min or ft^3/min
Example (for *ActualValue*): 10,0

8.25.9 PressureMainLine

Description: Actual value of the pressure in the main line.
Unit: bar or lbf/in^2 (=psi)
Example: 6

8.25.10 PressureReturnLine

Description: Actual value of the pressure in the return line.
Unit: bar or lbf/in^2 (=psi)
Example: 5

8.25.11 PressureDifference

Description: Setting and/or monitoring of the pressure difference between main and return line
Unit: bar or lbf/in^2 (=psi)
Example (for *ActualValue*): 2.5

8.25.12 ControlMode

Description: Defines to which setpoint the external channel is controlled
The *TypeDefinition* is *MultiStateValueDiscreteType*, so the *Properties EnumValues* and *ValueAsText* must be filled with the supported values out of Table 25.

Table 25 – Values for ControlMode

EnumValue	ValueAsText	Description
0	NONE	No control, only monitoring
1	TEMPERATURE	Temperature controlled
2	FLOW	Flow rate controlled
3	TEMP_DIFF	Temperature difference controlled
4	PRESS_DIFF	Pressure difference controlled

9 Alarms

For alarms (alarms, warnings, information) of the TCD the *TCDHelpOffNormalAlarmType* as defined in Table 26 shall be used, if the alarm facet is supported. A machine which connects to a TCD via EUROMAP 82.1 shall subscribe this event.

Table 26 – TCDHelpOffNormalAlarmType Definition

Attribute	Value				
BrowseName	TCDHelpOffNormalAlarmType				
IsAbstract	False				
Subtype of	HelpOffNormalAlarmType (defined in EUROMAP 83)				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling Rule**
HasProperty	Variable	DeviceMappingNumber	UInt32	PropertyType	M, R

The value of *DeviceMappingNumber* corresponds to the value given in the instance of the *OperationType* (see 7.1) for assigning the alarm to a device.

For unique identification of the alarm event, the *SourceNode* (included in *BaseEventType*) of the device needs to be sent for every alarm message. The *SourceNode* includes the namespace number and the Identifier from the object instance of *TCD_InterfaceType* (for general events) or the *NodeId* of a child element (e.g. a variable of *MonitoredParameterType* if this is out of tolerance).

For the *Severity Property* (included in *BaseEventType*) the following classes shall be used:

Table 27 – Severity Classes

Range of Severity	Description
667-1000	Messages of high urgency (error, system alarm): Limit values have been exceeded. The transgression has a direct influence on the operational safety of the unit. Acknowledgement: compulsory
334-666	Messages of medium urgency (warning, process alarm): Limit values have been exceeded. The transgression has no influence on the operational safety of the unit. Acknowledgement: not compulsory
1-333	Messages of low urgency (Information)