T.A.P.P.S. tutorial

Version 1.29 EN

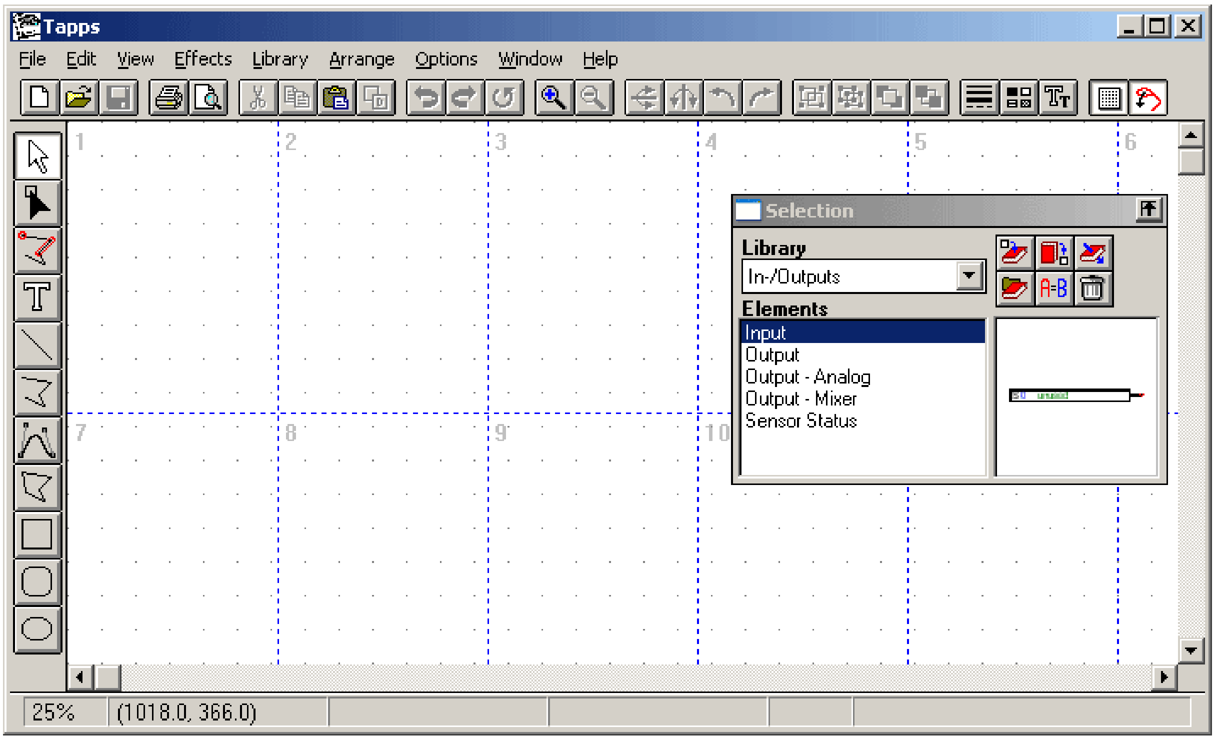
T.A.P.P.S. is a vector-based drawing program which is optimized for planning and programming controller UVR1611.
A configuration created with T.A.P.P.S. can be converted to a data format which can be loaded to the control using the Bootloader and the *Memory Manager* program.

These instructions exclusively explain the necessary tools and methods which are required to graphically create a circuit diagram and the resulting programming of the control.

The bases of the control technology or a precise description of individual function modules are not touched on in this tutorial. Information on function modules and their mode of operation can be found in the instruction manual of the UVR1611 controller unit.

Table of Contents

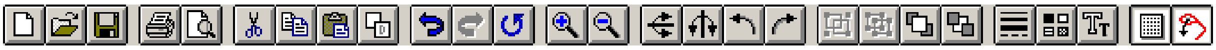
User interface



Menu bar

All functions required for setting a configuration of UVR1611 are in the <file> menu. These functions are described in detail below in these instructions.

Symbol bar



New, **Open**, **Save**:

Saving changes should also be carried out regularly while working in order to prevent a major loss of data in the event of errors (computer crash, power cut) occurring.

The project’s name must not be any longer than 8 characters and must not contain any special characters or umlauts. The path where the project is saved must also not have any special characters and umlauts.

A project consists of 2 files, while the drawing graphics is saved as [file name].eng and the associated parameterisation as [file name].par in the same directory.
When opening or saving an *.eng file with T.A.P.P.S., the associated *.par file is automatically loaded or saved.
When sending a project by e-mail or copying or dragging and dropping to the hard disk, this must be performed with both files!

Cut, Copy, Add, Duplicate:
Note: When copying/duplicating a function or message block at the drawing level, the drawing object only is copied. The parameterisation is NOT adopted!!

Undo: Undoes the changes made by the user to the drawing interface. It must be observed that it cannot be used to restore disconnected links!

Zoom: Zooms in or out of the view of the drawing by a set factor, while the object marked is displayed in the centre of the screen after zooming.

Mirror: Horizontally or vertically mirrors the object marked in the drawing.

Turn: Rotates the object marked in the drawing by 90° to the left or right.

Group, Ungroup: Groups or ungroups the objects marked in the drawing. This function is used when creating own hydraulics symbols only. Components of a logical circuit plan (inputs/outputs, functions, etc.) must never be grouped, as their configuration is lost!

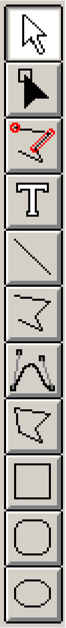
Line attributes: Changes to the characteristics, such as line weight, type or colour of the marked line. If no object is marked in the drawing, the global characteristics for future lines can be changed.

Grid on/off: An active grid helps to evenly align the objects in the drawing.

Snap mode on/off: The snap mode is necessary in order to allow linking the elements with lines. If this mode is not activated, the elements are only linked optically but not logically. For this reason, this mode must always be active when programming!

Tool bar

The tool bar provides various tools to create and edit graphics and texts.



Selection mode: In this mode objects can be marked and subsequently parameterised or dragged and dropped in the drawing.
- Multiple selection using the Shift key or the selection frame while holding the mouse button down.
- Change to an object’s attributes by double clicking.

Node mode: Mode for subsequent editing of drawing objects such as lines (links).
- Node conduct key = F8

Snap definition mode: Mode to create snap points.
This function is not required for programming, but can be useful for creating own hydraulics symbols, for instance.

Text mode: Insert or change text in the drawing.

Line mode: Drawing a line which can also be used as logical link when programming.
- Symmetry key = Ctrl
- Cancel with Esc or right mouse button

Poly line mode: Draw a poly line which is also recommended to create logical links for programming.
- Symmetry key = Ctrl
- Line conduct key = F8
- Complete line with right mouse button
- Cancellation with Esc

Holding down the symmetry key (Ctrl) or a respectively selected line conduct mode (selection with key F8) simplify drawing horizontally and vertically aligned lines.

When holding the Shift key down, highlighted elements can be moved in small steps using the arrow keys on the keyboard.
Holding the control key (Ctrl) down enables movement in large steps.

Library

The following libraries are available in the factory for planning and programming with T.A.P.P.S.:

- Hydraulic symbols (ta_hydr.lib)
- In-/Outputs (ta_inout.lib)
- Functions (ta_func.lib)
- Messages (ta_msg.lib)
- Network (ta_net.lib)
- Links (ta_line.lib)
- Online diagram (ta_visio.lib)

The <hydraulics symbols> comprises the symbols for drawing simplified hydraulics plans which serve as documentation only.
In the <online diagram> library there are elements for the graphical design of Html files for the online diagram of the Bootloader BL-NET.
All further libraries include components which are required to create the controller configuration.

When starting T.A.P.P.S. the supplied libraries are automatically loaded and displayed in the <selection> window.

Create own library

If the library should be extended by additional, own hydraulics symbols a separate library file (*.lib) must be created, as an extension of the libraries supplied from the factory would be lost when updating or reinstalling T.A.P.P.S.!

Manage libraries where new libraries can be created and saved.
Add an object marked in the drawing to the selected library.

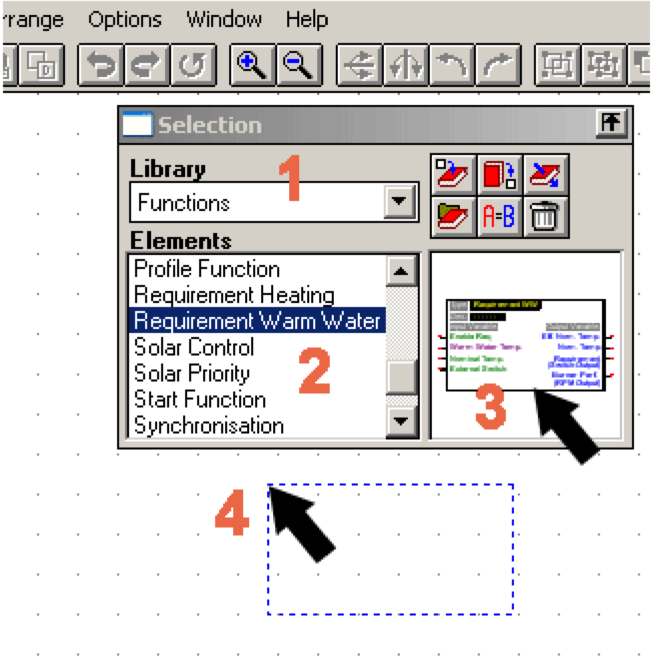
1. Call up the library management.
   1.1. Create a new library (e.g. “My library”).
   1.2. Create a target file (e.g. “mybiblio.lib”) for the new library. In order to save the file in the same folder as the factory libraries, it may be necessary to change to the program folder and from there back to the library folder!
2. Mark a symbol drawn and grouped in the drawing interface and add it to the new library.
3. Once all symbols have been inserted, these changes to the library can be saved in the management.
4. For the own library to also load when starting T.A.P.P.S., the list of loaded libraries must be newly created with the “Options” → “Save” menu item.

Programming

Programming procedure

1. Insert required elements (inputs/outputs, functions, etc.) from the libraries in the drawing and position them accordingly.
2. Graphically link these elements with poly lines, as required.
3. Parameterise inputs and outputs.
4. Parameterise functions and messages.
5. Create function overview.
6. With the “Export to translator” function create a configuration file (functional data) from the logical circuit diagram for controller.
7. The functional data can be loaded with the *Memory Manager* program from a PC to the Bootloader and in a further step to the according controller.

Add library element to the drawing



1. Select the respective library.

2. Mark the element which should be added to the drawing.

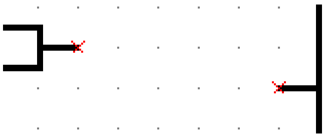
3. Drag the element by holding down the mouse button to the requested position in the drawing.

4. The element is inserted in the drawing, when letting go of the mouse button.

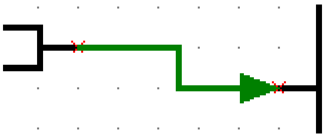
Create links of the elements

draw simple line

draw poly line (recommended for links)



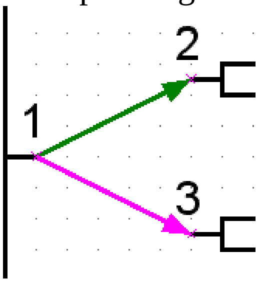
The library elements have respective connections with one snap point x each for their input and output variables.



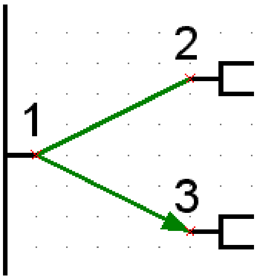
A logical link is created by drawing a (poly) line from snap point x of the first element to snap point x of the second element.

The characteristics of a (poly) line have no influence on their logical link. In order to facilitate understanding of signal routings, the lines can nevertheless be represented with an arrow, for instance.

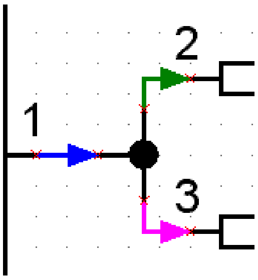
Only ever 2 points (start and end of a line) can be connected with a line. Several links to a connection can be implemented by using separate lines from the connection or by incorporating a node point from the library.



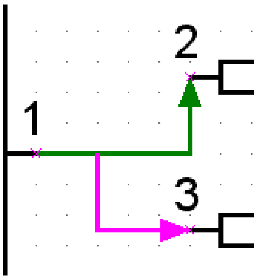
CORRECT
Connections 1-2 and 1-3



INCORRECT
Only 2-3 connected

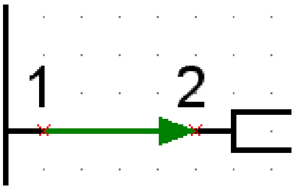


CORRECT
Connections via nodes from the library.

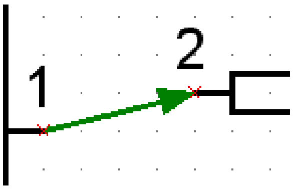


INCORRECT
Only 1-2 connected

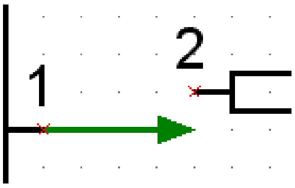
A logical connection can be simply controlled in the drawing interface by moving the respective connected objects. For a valid link, the line linked to the connection also moves.



Link can be verified by moving the objects.



Valid link for connection 2.

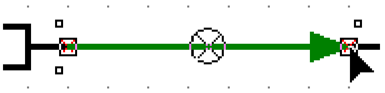


Invalid link for connection 2.

When moving or deleting a line its links are deleted. These links cannot be restored by pressing undo!
If the connected elements are moved with the line, their link remains.

Edit (restore) links

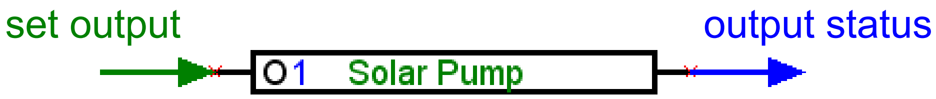
Subsequently edit lines (links)



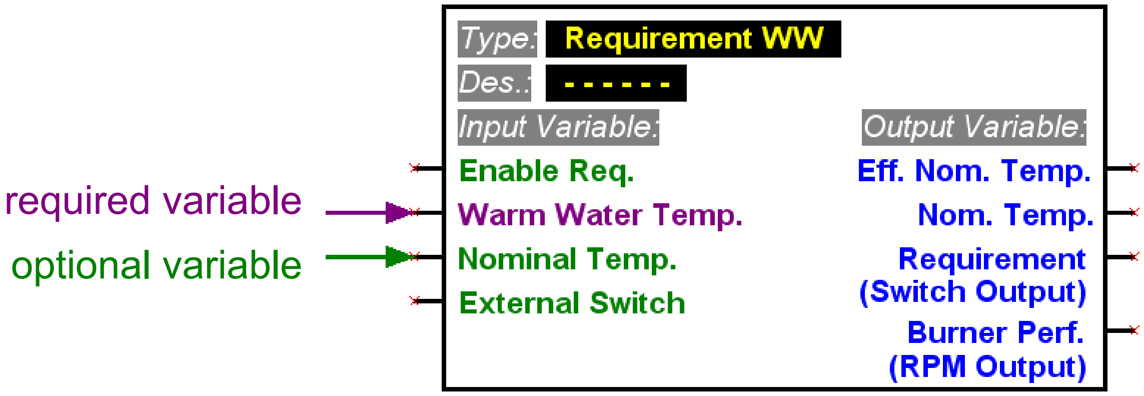
After the line has been clicked, its node point can be moved to snap point x of the requested connection with the cursor by holding down the left mouse button. If the node point appears in the colour of the snap point, the mouse button can be let go which allows creating a logical link.

Characteristics for connections and links

■ An output symbol has 2 connections, while the left connection is used to controller the output and the right connection to establish its current status.

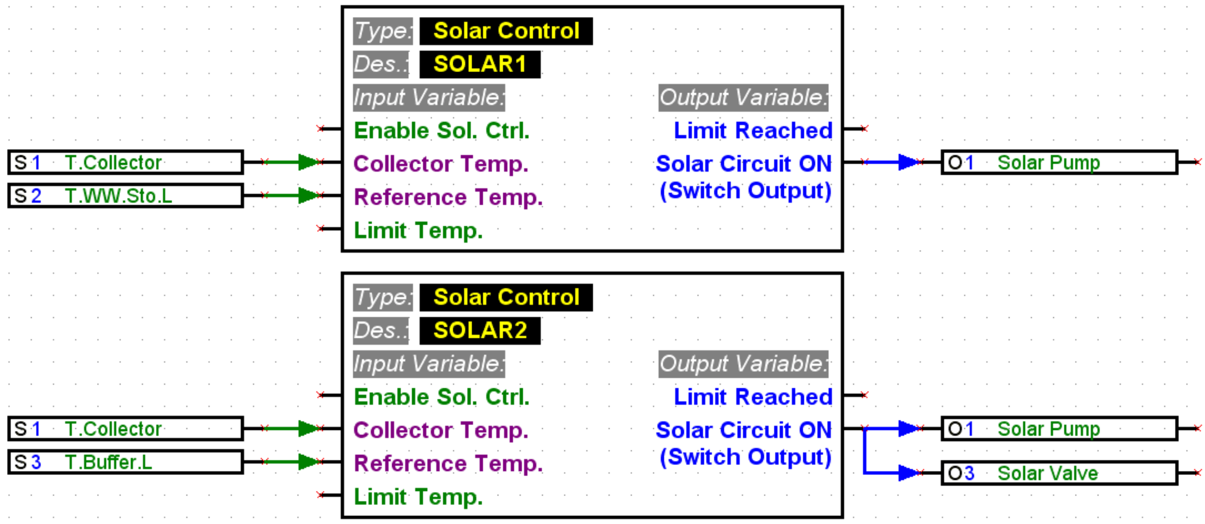


■ Contrary to the green input variables, which can be optionally used, purple input variables are mandatory for the function and must therefore be linked!

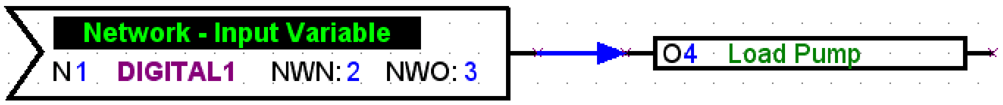


■ For output variables the type of the outputs possible for a link is stated below their name.

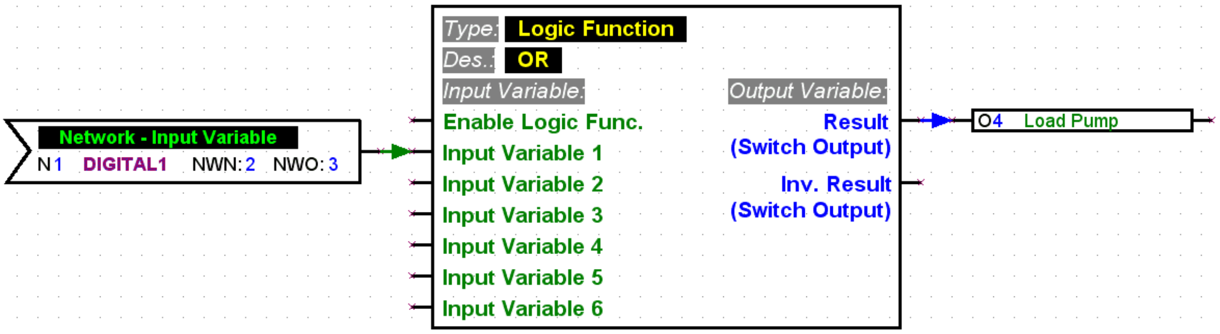
■ In order to concisely design a logical circuit diagram, it is possible to use several symbols for the same output/input, sensor status and network entry and network status in the drawing.
If there are several output symbols for the same output in the drawing and it is therefore connected with several functions, its initial condition corresponds to the result of a disjunction of all linked variables.



■ An output cannot be directly controlled from the network input. A respective logical or analogue function is required to do so.

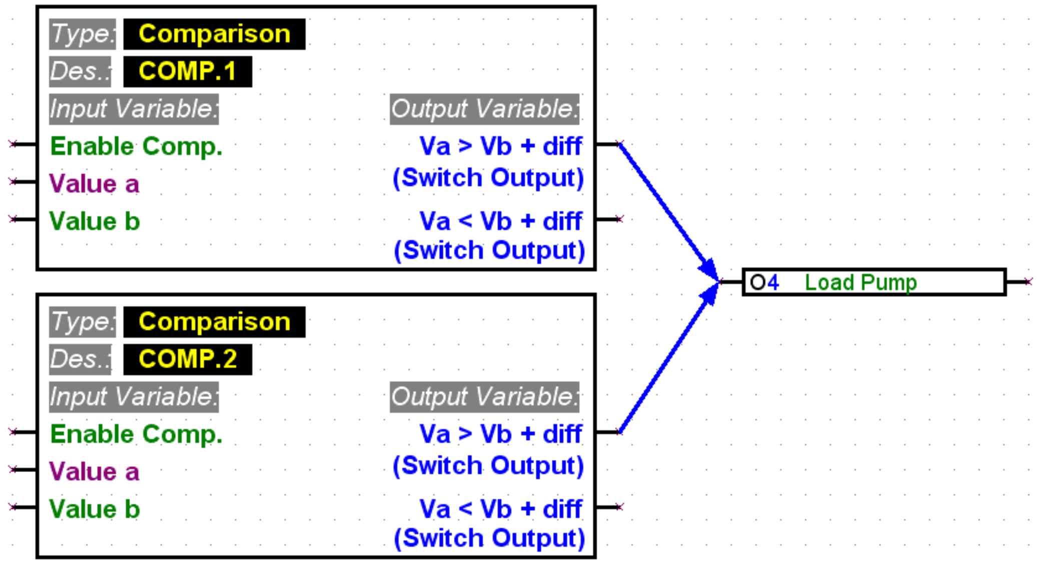


Invalid link, as the output cannot be directly controlled from the network input.

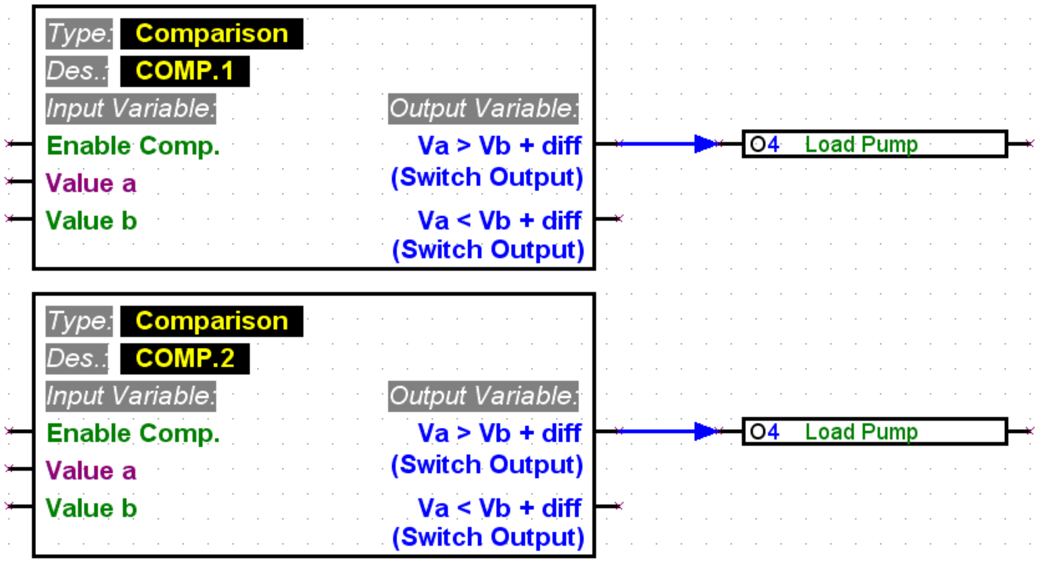


Valid link, as the output is controlled via a respective logical function (OR) from the network input.

- Output variables must not be linked to one another.

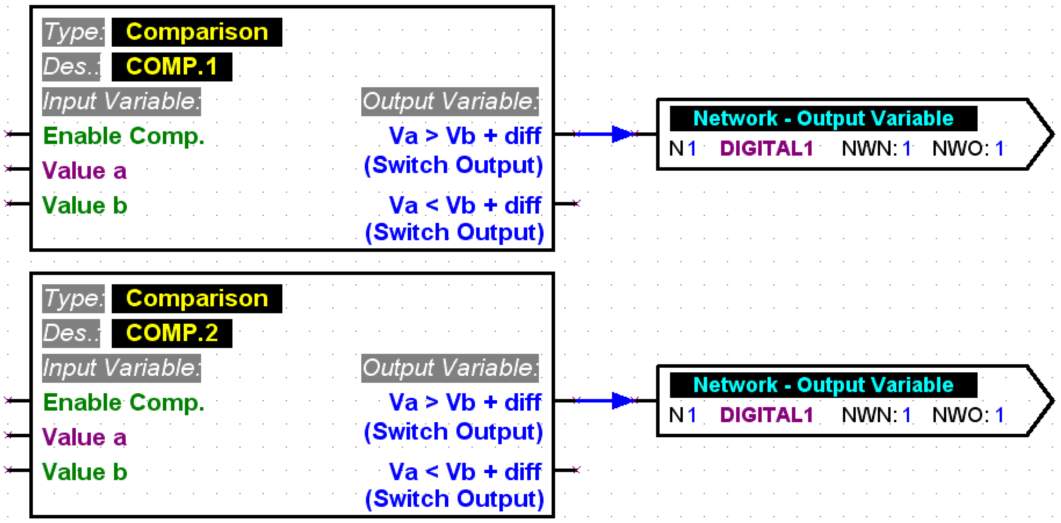


Invalid links to the output, as the output variables of both functions are also linked.

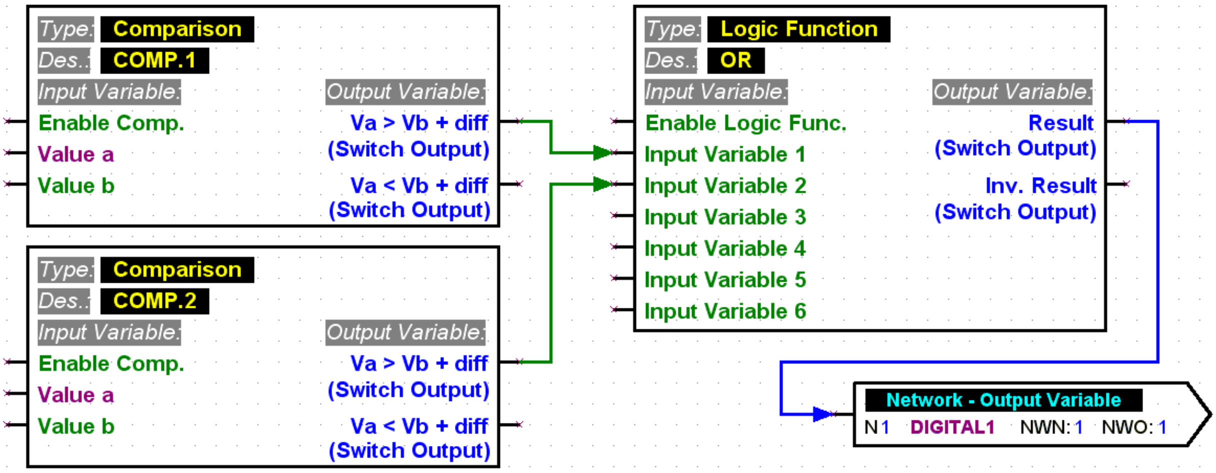


Valid link of both functions with the same output via two separate output symbols.

■ Only one symbol may be used in the drawing for one network output.
If the network output value of several functions should be mapped, it must be implemented accordingly with a logical or analogue function.

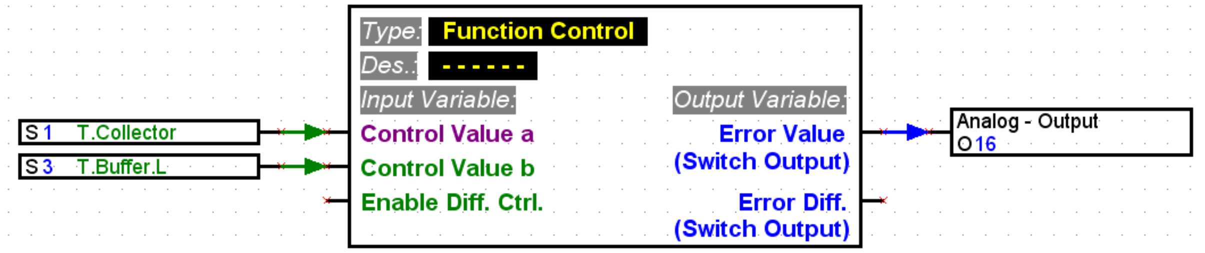

Type: Comparison
Des.: COMP.1
Input Variable:
Enable Comp.
Value a
Value b
Output Variable:
Va > Vb + diff (Switch Output)
Va < Vb + diff (Switch Output)
Network - Output Variable
N1 DIGITAL1 NWN: 1 NWO: 1
Type: Comparison
Des.: COMP.2
Input Variable:
Enable Comp.
Value a
Value b
Output Variable:
Va > Vb + diff (Switch Output)
Va < Vb + diff (Switch Output)
Network - Output Variable
N1 DIGITAL1 NWN: 1 NWO: 1

Invalid links, as the DIGITAL 1 network output occurs several times in the drawing.

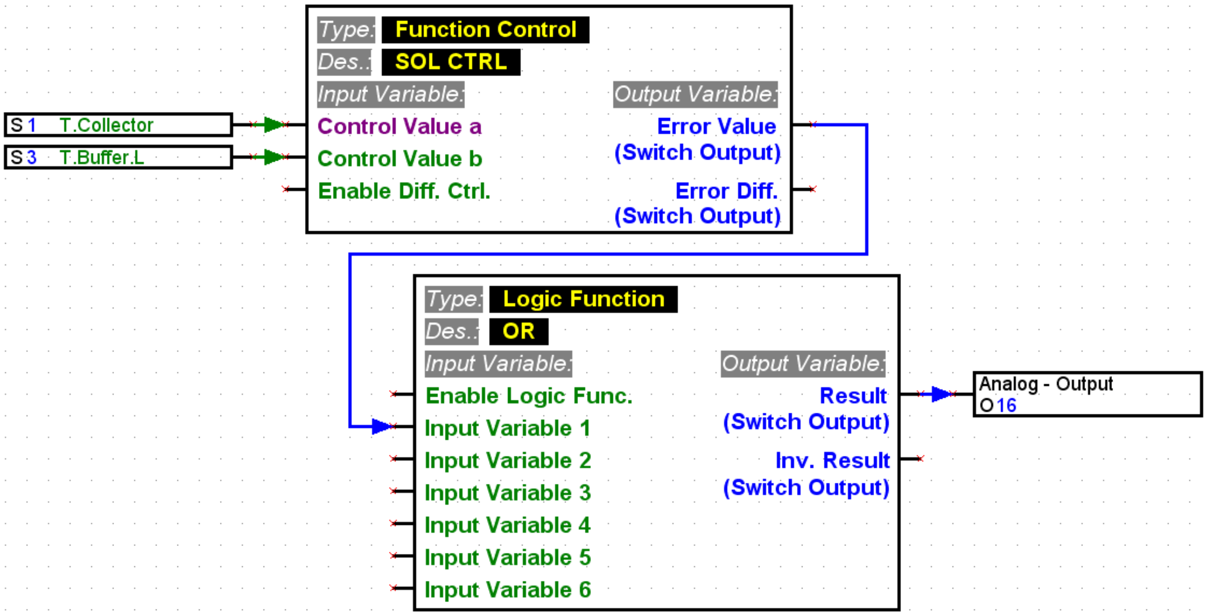


Valid link, as the two functions are linked via a respective logical function (OR) with the network output.

■ To ensure compatibility with older versions, the function modules *function control* and *boiler cascade* cannot directly activate the analogue output O16.
If such a link is necessary, it must be implemented indirectly via a logic function.

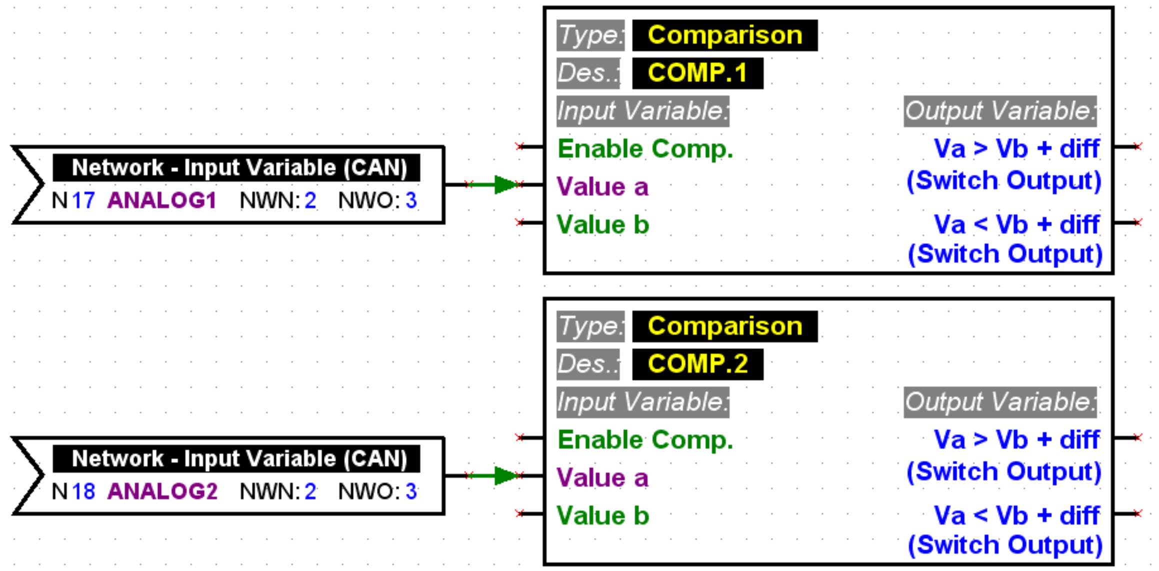


Invalid link, as the analogue output O16 cannot be activated directly from the function control.

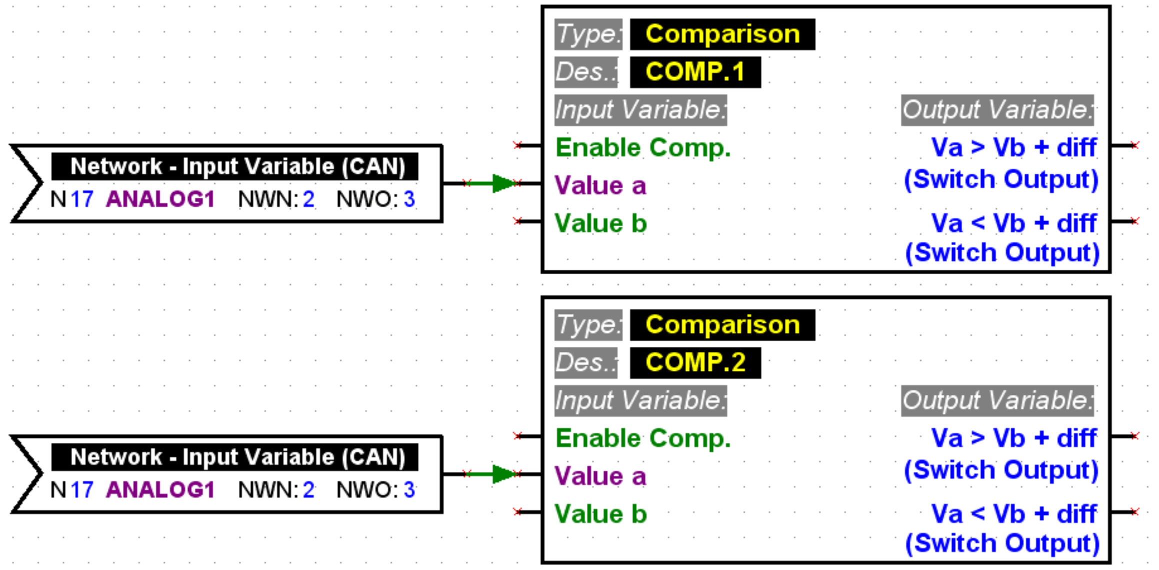


Valid link, as the analogue output O16 can be activated from the function control indirectly using a logic function (OR)

■ For every parameter which is read in from the network for one unit, only one network input may be defined.
So that it is possible to design a clearly laid out logical circuit diagram, it is possible to use several symbols for the same network input in the drawing.



Invalid links, as the two network inputs ANALOG 1 and ANALOG 2 read in the same parameters (CAN, NWN 2, NWO 3) from the network.



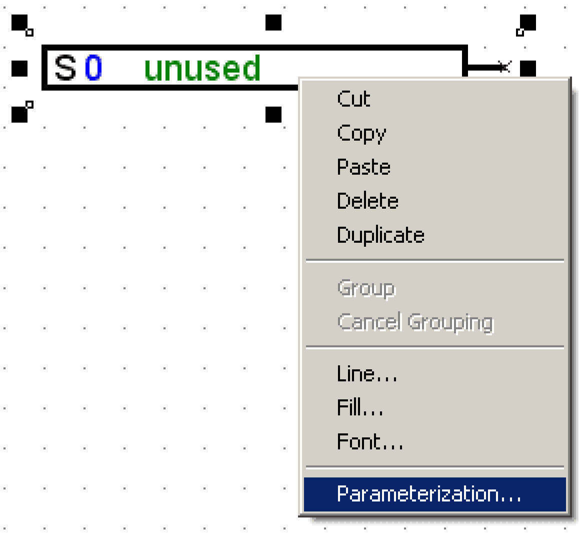
Valid links, as the parameter (CAN, NWN 2, NWO 3) is only read in from the network by the network input ANALOG 1.

Parameterise elements

The texts, in particular the names, of the library elements of a logical circuit diagram (inputs/outputs, functions, etc.) must not be directly changed at the drawing level! The names must be assigned to the elements via their parameterisation.

Additional information and comments can be inserted in the drawing as own text.

In order to configure (parameterise) an element inserted and linked from the library, the following procedure applies:



1. Mark the respective element in the drawing by clicking once on the element’s line or text with the left mouse button.

2. By pressing the right mouse button, a menu appears where the parameterisation of the element can be called up.

3. The para1611 parameterisation window is called up which can be used to configure the element.

Error-List

For a faulty or invalid link, the error log is displayed with a respective error message.
An element cannot be parameterised, until all its links are valid and complete.

Parameterise inputs and outputs

The parameterisation procedure is as follows: first the respective input or output is selected and subsequently its settings, starting with the type, are set.

The settings of all inputs or outputs are saved and imported to the drawing. The drawing object is also assigned the selected input or output.
The parameterisation window is closed.

The settings of all inputs or outputs are saved and imported to the drawing. The parameterisation window is closed.

The selected input or output is assigned to the drawing object. The parameterisation window remains open.

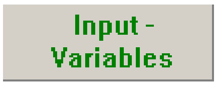
The parameterisation window is closed without saving any changes.

Parameterise functions and messages

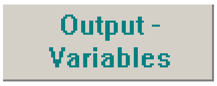
The settings of the function or message are saved and imported to the drawing. The parameterisation window is closed.

The parameterisation window is closed without saving any changes.

Switches to the parameter view, where the function or message can be configured.



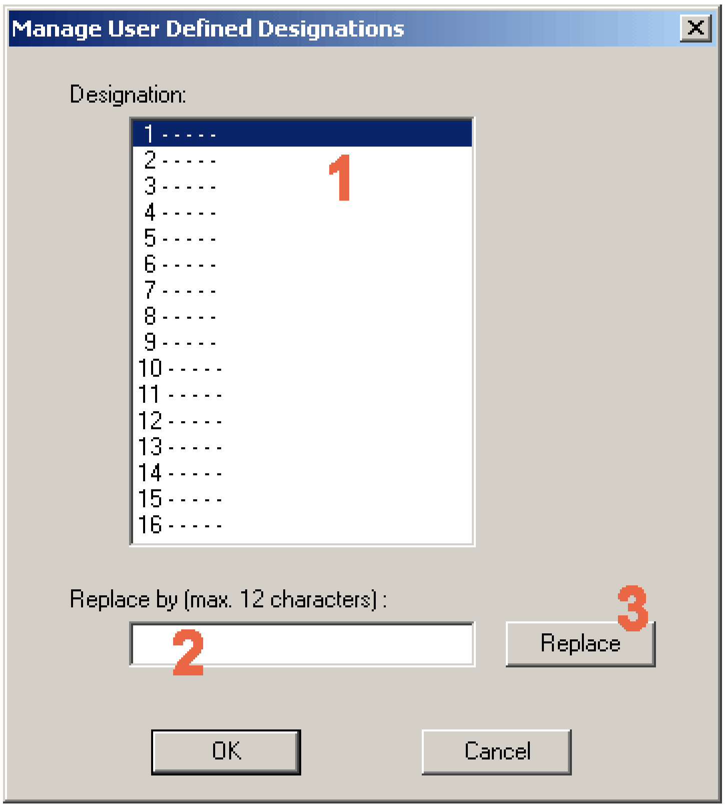
Switches to the input variables view.
Here the links created in the drawing are displayed with the input variables. Status (ON/OFF) or mode (normal/inverse) of the input variables, for instance, can also be set here.



Switches to the output variables view.
For a function, the outputs graphically linked to the output variables in the drawing are displayed.
For a message, links are created with outputs in this view instead of the drawing.

User-defined names

If it is not possible to locate a suitable description in the various pre-defined groups, a total of 16 individual descriptions can be defined, which are then available for inputs/outputs, messages and functions. However for functions displayed on the UVR1611 only 9 of the maximum 12 possible characters can be displayed.


Manage User Defined Designations
Designation:
1
Replace by (max. 12 characters) :
2
3
Replace
OK
Cancel

1 - - - - -
2 - - - - -
3 - - - - -
4 - - - - -
5 - - - - -
6 - - - - -
7 - - - - -
8 - - - - -
9 - - - - -
10 - - - - -
11 - - - - -
12 - - - - -
13 - - - - -
14 - - - - -
15 - - - - -
16 - - - - -

1. Select the name to be changed. This name can be imported to the input field for editing by double clicking it.

2. Enter the new user-defined name. It must not be longer than 12 characters and not contain any special characters and umlauts.

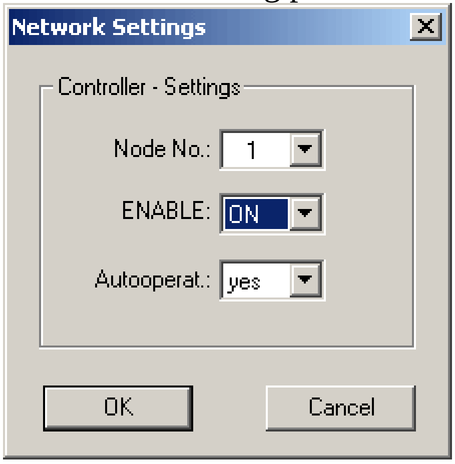
3. Replace the selected name with the ones in the input field.

Network (CAN- Bus)

In order to exchange data in a CANopen network with other devices of the UVR1611 family, the UVR1611 has network inputs and outputs.

The configuration must be created for every device in a network in a respectively separate drawing file.

Basic, global settings for the CAN network (node number of the device, enable, etc.) can be made during parameterisation of the network inputs and outputs.



Node No.: 1

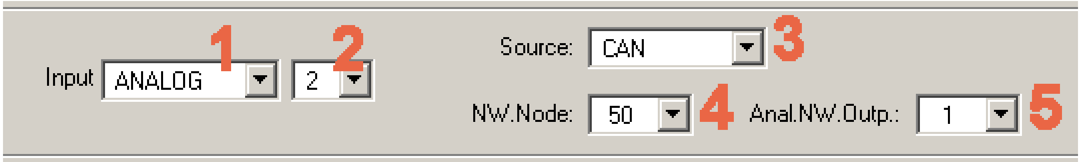
ENABLE: ON

Autooperat.: yes

Each unit in the network must have a unique node number. Therefore, never allocate multiple network units the same node number!

Parameterise network inputs

The reading in of data from the CAN network takes place via the network inputs.
In total, 16 digital and well as 16 analogue network inputs are available for reading in data from the CAN or DL bus. The corresponding source (CAN or DL) must be set for the respective network input.



Input ANALOG 2

Source: CAN

NW.Node: 50

Anal.NW.Outp.: 1

1. Select the type of the variable which should be imported from the network.

2. Select an unused network input on the device.

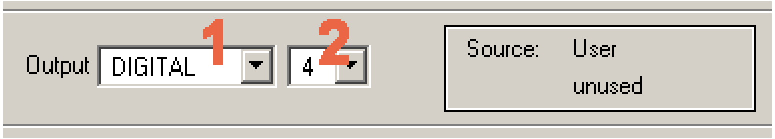
3. Select the source, CAN bus.

4. Select the node number of the sending device whose data are imported.

5. Select the network output via which the sending device issues the data.

Parameterise network outputs

For outputting data to the CAN network with a UVR1611 there are a total of 16 digital as well as 16 analogue network outputs available.



Output DIGITAL 4

Source: User unused

1. Select the type of the variable which should be exported to the network.

2. Select an unused network output on the device for the export.

The settings of all network variables are saved and imported to the drawing. The drawing object is also assigned the selected input or output variable.
The parameterisation window is closed.

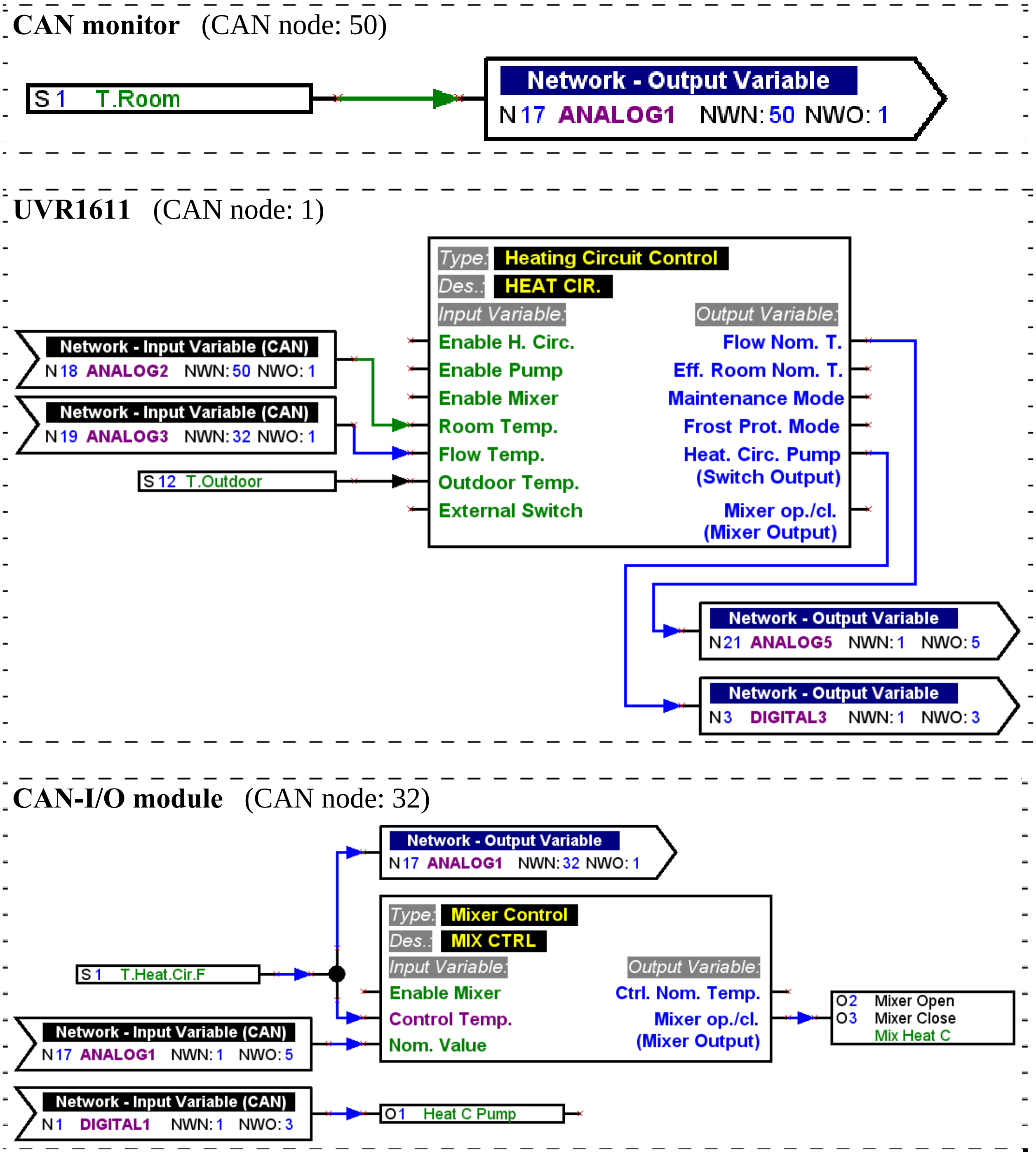
The settings of all network variables are saved and imported to the drawing. The parameterisation window is closed.

The selected input or output variable is assigned to the drawing object. The parameterisation window remains open.

The parameterisation window is closed without saving any changes.

Example:

A network with 3 devices: UVR1611, CAN monitor and CAN-I/O module.
The room temperature is recorded by the CAN monitor and the pump and mixer of the heating circuit are controlled via the CAN-I/O module.


CAN monitor (CAN node: 50)
S 1 T.Room
Network - Output Variable
N 17 ANALOG1 NWN: 50 NWO: 1
UVR1611 (CAN node: 1)
Network - Input Variable (CAN)
N 18 ANALOG2 NWN: 50 NWO: 1
Network - Input Variable (CAN)
N 19 ANALOG3 NWN: 32 NWO: 1
S 12 T.Outdoor
Type: Heating Circuit Control
Des.: HEAT CIR.
Input Variable:
Enable H. Circ.
Enable Pump
Enable Mixer
Room Temp.
Flow Temp.
Outdoor Temp.
External Switch
Output Variable:
Flow Nom. T.
Eff. Room Nom. T.
Maintenance Mode
Frost Prot. Mode
Heat. Circ. Pump (Switch Output)
Mixer op./cl. (Mixer Output)
Network - Output Variable
N 21 ANALOG5 NWN: 1 NWO: 5
Network - Output Variable
N 3 DIGITAL3 NWN: 1 NWO: 3
CAN-I/O module (CAN node: 32)
Network - Output Variable
N 17 ANALOG1 NWN: 32 NWO: 1
S 1 T.Heat.Cir.F
Type: Mixer Control
Des.: MIX CTRL
Input Variable:
Enable Mixer
Control Temp.
Nom. Value
Output Variable:
Ctrl. Nom. Temp.
Mixer op./cl. (Mixer Output)
O 2 Mixer Open
O 3 Mixer Close
Mix Heat C
Network - Input Variable (CAN)
N 17 ANALOG1 NWN: 1 NWO: 5
Network - Input Variable (CAN)
N 1 DIGITAL1 NWN: 1 NWO: 3
O 1 Heat C Pump

⚠ T.A.P.P.S. can only be used to create configurations for the UVR1611. The configurations of CAN monitor and CAN I/O module can be made directly on the device itself or using the software *F-Editor*.

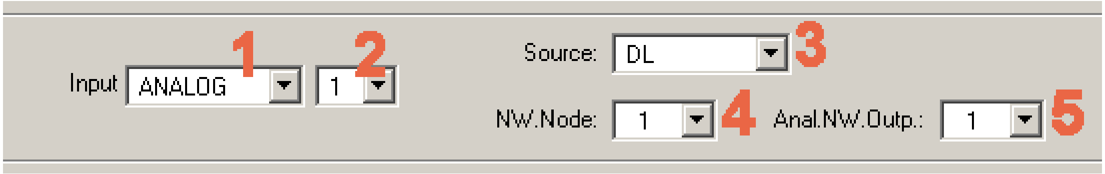
DL Bus (data line)

From serial number 13286 and for operating systems from A3.00, the UVR1611 has the option of recording measured values over the DL bus (data line) from sensors that are fitted with a DL-interface.
Such sensors are for example the electronic volume flow encoder VFS2-40DL and the electronic pressure sensor RPS0-6DL.

⚠ To be able to read in measured values from the DL bus, output 14 of the UVR1611 must be defined as type "data line"!

Parameterise the network inputs

The reading in of data from the DL bus takes place via the network inputs.
In total, 16 digital and well as 16 analogue network inputs are available for reading in data from the CAN or DL bus. The corresponding source (CAN or DL) must be set for the respective network input.



Input ANALOG 1 Source: DL NW.Node: 1 Anal.NW.Outp.: 1

1. Select the type of variable to be read in from the DL bus.
2. Select an unused network input on the unit.
3. Select the source, DL bus.
4. Set the **sensor address** as the node number.

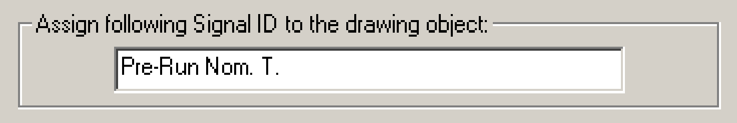
   Never connect more than one sensor with the same address to the DL bus! Details of sensor address allocation are contained in the sensor data sheet.
5. Set the **measured value index**, which is to be read in by the sensor, as the network output, Take the corresponding index for the desired measured value from the sensor data sheet.

Signal transfer / signal adoption

In order to design a logical circuit diagram concisely, there are, in addition to a node point, symbols for “virtual” links in the library <links>. This means that a logical connection can be implemented without having to draw the respective line diagonally across the entire page of the drawing for the link.

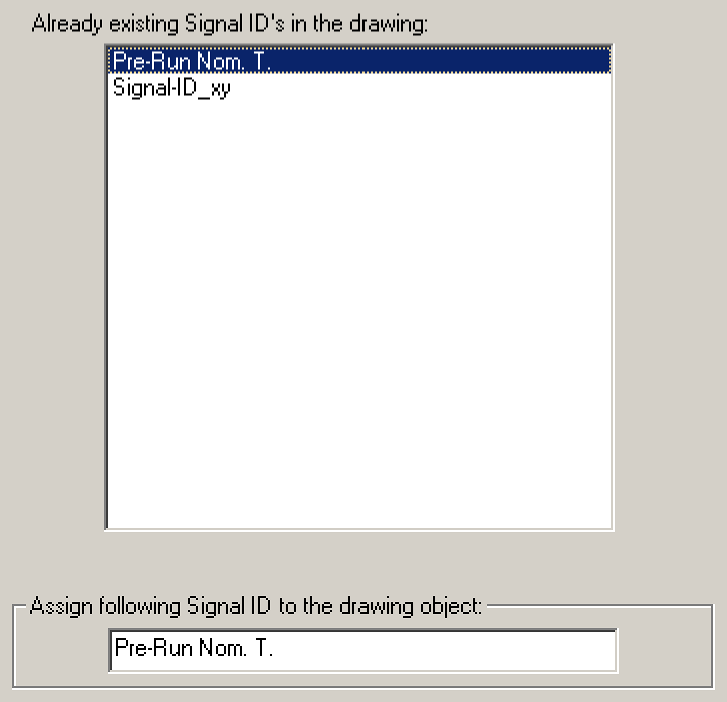
ℹ The use of signal transfers/adoptions should be kept to a minimum in a drawing, as many different signals make a circuit diagram a lot more difficult to understand!

Parameterise signal transfer


Assign following Signal ID to the drawing object:
Pre-Run Nom. T.

1. Enter an identification for the signal which is not yet used in the drawing. This identification must not be any longer than 20 characters!
2. Assign the selected signal identification to the symbol in the drawing.

Parameterise signal adoption


Already existing Signal ID's in the drawing:
Pre-Run Nom. T.
Signal-ID_xy

Assign following Signal ID to the drawing object:
Pre-Run Nom. T.

1. Select the identification of the signal to be adopted by double click it in the list of signal identifications which are already available in the drawing.

2. Assign the selected signal identification to the symbol in the drawing.

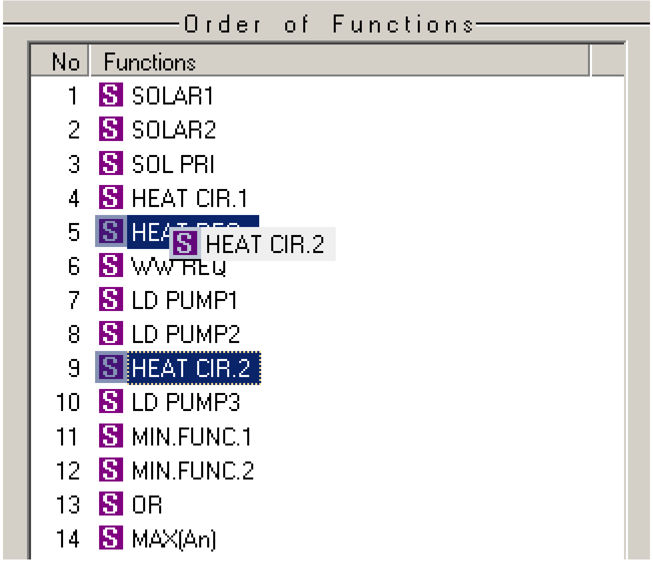
Assign The selected signal identification is assigned to the drawing object. The parameterisation window is closed.

Arrange functions

File → Arrange functions…

The order of the functions in the UVR1611 generally corresponds to the sequence which is inserted into the drawing from the library.

In order to change the sequence of the function, the following procedure:



Order of Functions

No	Functions
1	S SOLAR1
2	S SOLAR2
3	S SOL PRI
4	S HEAT CIR.1
5	S HEAT REQ
6	S WW REQ
7	S LD PUMP1
8	S LD PUMP2
9	S HEAT CIR.2
10	S LD PUMP3
11	S MIN.FUNC.1
12	S MIN.FUNC.2
13	S OR
14	S MAX(An)

1. Select the functions whose position should be changed in the sequence (click).

2. Move the function to the respective position by holding the mouse button.

3. When letting go of the pressed mouse button, the function is inserted at the new position.

Arrange messages

File → Arrange messages…

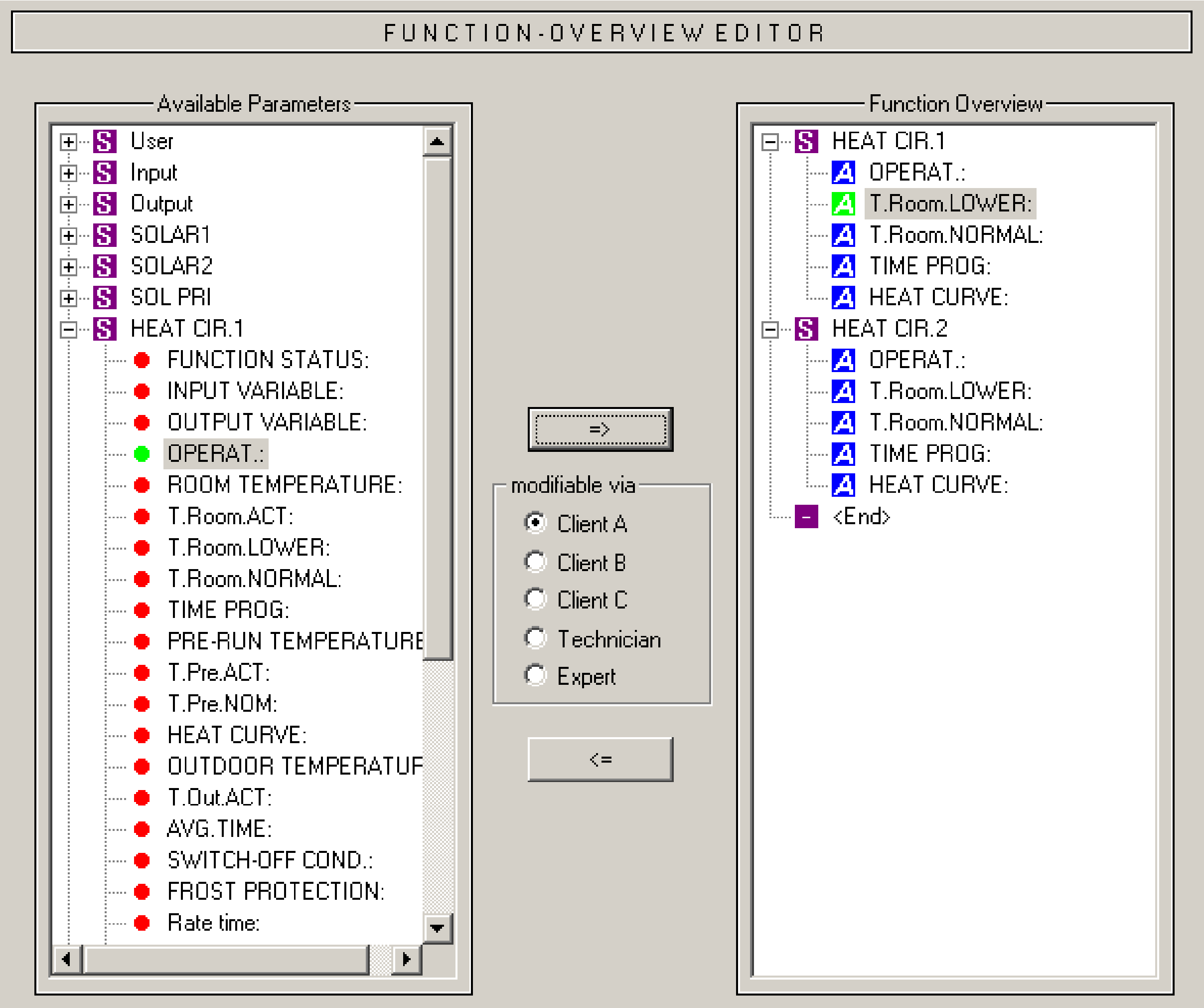
The order of the messages in the UVR1611 generally corresponds to the sequence which is inserted into the drawing from the library.

The procedure to change the sequence corresponds to that of the functions.

Function overview editor

File → Function overview editor

The function overview is a menu page in controller which only displays the information which is relevant to the customer.
The parameters displayed there can also be assigned an authorisation for changing the setting values.



Available Parameters

- User
- Input
- Output
- SOLAR1
- SOLAR2
- SOL PRI
- HEAT CIR.1
  - FUNCTION STATUS:
  - INPUT VARIABLE:
  - OUTPUT VARIABLE:
  - OPERAT.:
  - ROOM TEMPERATURE:
  - T.Room.ACT:
  - T.Room.LOWER:
  - T.Room.NORMAL:
  - TIME PROG:
  - PRE-RUN TEMPERATURE
  - T.Pre.ACT:
  - T.Pre.NOM:
  - HEAT CURVE:
  - OUTDOOR TEMPERATUF
  - T.Out.ACT:
  - AVG.TIME:
  - SWITCH-OFF COND.:
  - FROST PROTECTION:
  - Rate time:

modifiable via

● Client A
○ Client B
○ Client C
○ Technician
○ Expert

Function Overview

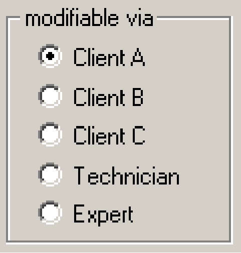
- HEAT CIR.1
  - OPERAT.:
  - T.Room.LOWER:
  - T.Room.NORMAL:
  - TIME PROG:
  - HEAT CURVE:
- HEAT CIR.2
  - OPERAT.:
  - T.Room.LOWER:
  - T.Room.NORMAL:
  - TIME PROG:
  - HEAT CURVE:
- <End>

Procedure for inserting a display parameter into the function overview:

1. Select the parameter to be displayed in the function overview from the list of available parameters.
2. Select the position in the list of parameters of the function overview, above which this display parameter is be inserted.
3. Select the authorisation for changing the parameter.
4. Insert the selected display parameter in the function overview.

=> and the *Ctrl* + *arrow right* key combination insert the parameter selected in the list <Available Parameters> at the requested position of the list <Function Overview>.

<= and the *Del* key delete the display parameter marked in the <Function Overview> list.



modifiable via

- ● Client A
- ○ Client B
- ○ Client C
- ○ Technician
- ○ Expert

Authorisations for changing the parameter:

Client	The display parameter can be viewed and changed by all users.
Technician	The display parameter can be viewed by all users but changed by a technician or expert only.
Expert	The display parameter can be viewed by technicians and experts and may be changed by experts only.

The A, B and C areas of application are only relevant in combination with a CAN monitor. If the application area A is set for the client on the CAN monitor, he only sees the technician's display parameters and those of application area A which can be edited by him.

At the controller itself, it makes no difference to the client whether a parameter is assigned the A, B or C area of application.

The editing authorisation assigned is displayed in the <Function Overview> list for every parameter and can be subsequently edited for a marked display parameter with respective key combinations.

Ctrl + A … client A	Ctrl + B … client B	Ctrl + C … client C
Ctrl + T … technician	Ctrl + E … expert	

CAN - Data logging

File → CAN - data logging…

For data logging using the UVR1611 via a CAN bus, two data records can be defined with the parameters to be captured. One data record comprises 16 analogue and 13 digital parameters, which can be freely defined.
The left column shows all the parameters available for data logging. These can be added to the selected data record in the right column. In this respect the following details must be observed:

Speed stages of outputs 1, 2, 6 and 7:

To capture the speed stage of an output, the digital parameter with the same number must be allocated to the output in data record 1. The output A6 must be allocated to parameter DIGITAL 6. If the output is allocated to another parameter, then status output still occurs (ON/OFF), but no speed stage output.

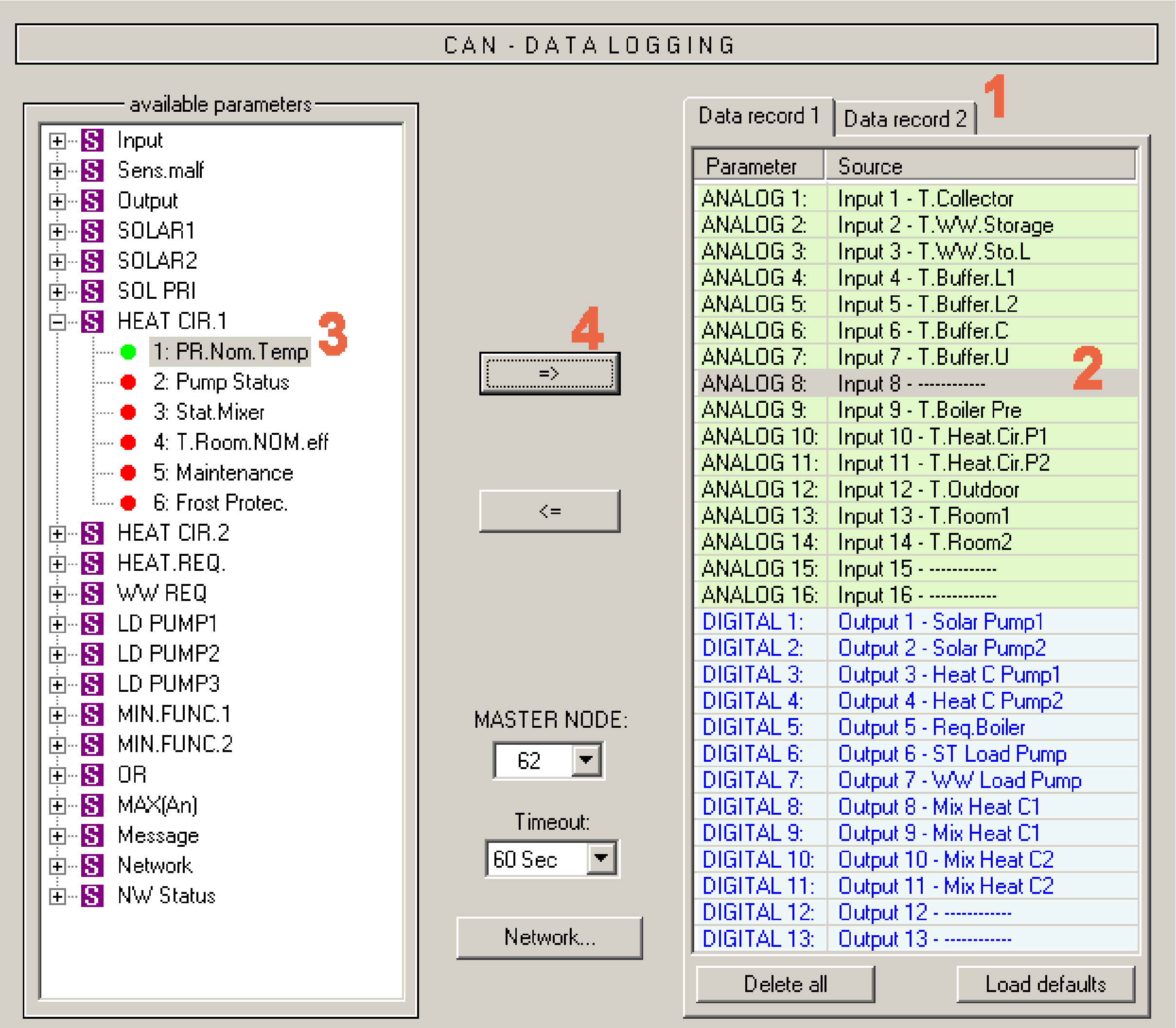
"Heat quantity counter" functions:

The output variables of the "heat quantity counter" functions are, according to the sequence in the function list (see "ordering functions"), automatically linked by the UVR1611 to the two data records (heat quantity counters 1 and 2 in data record 1, heat quantity counters 3 and 4 in data record 2). The parameters of a "heat quantity counter" function are therefore also not displayed in the list of available parameters.

Procedure for linking a parameter into data logging:

(for further clarification, see the diagram on the next page)

1. Select the data record to be defined from the right column.
2. Highlight the position in which a new parameter is to be inserted in the data record.
3. Select the parameter which is to be inserted in the data record from the list of <available parameters>.
4. Insert the selected parameter in the data record. After insertion of the parameter, the data record automatically selects the following parameter.



- Input
- Sens.malf
- Output
- SOLAR1
- SOLAR2
- SOL PRI
- HEAT CIR.1
  - ● 1: PR.Nom.Temp
  - ● 2: Pump Status
  - ● 3: Stat.Mixer
  - ● 4: T.Room.NOM.eff
  - ● 5: Maintenance
  - ● 6: Frost Protec.
- HEAT CIR.2
- HEAT.REQ.
- WW REQ
- LD PUMP1
- LD PUMP2
- LD PUMP3
- MIN.FUNC.1
- MIN.FUNC.2
- OR
- MAX(An)
- Message
- Network
- NW Status

MASTER NODE: 62

Timeout: 60 Sec

Data record 1 | Data record 2

Parameter	Source
ANALOG 1:	Input 1 - T.Collector
ANALOG 2:	Input 2 - T.WW.Storage
ANALOG 3:	Input 3 - T.WW.Sto.L
ANALOG 4:	Input 4 - T.Buffer.L1
ANALOG 5:	Input 5 - T.Buffer.L2
ANALOG 6:	Input 6 - T.Buffer.C
ANALOG 7:	Input 7 - T.Buffer.U
ANALOG 8:	Input 8 - -----------
ANALOG 9:	Input 9 - T.Boiler Pre
ANALOG 10:	Input 10 - T.Heat.Cir.P1
ANALOG 11:	Input 11 - T.Heat.Cir.P2
ANALOG 12:	Input 12 - T.Outdoor
ANALOG 13:	Input 13 - T.Room1
ANALOG 14:	Input 14 - T.Room2
ANALOG 15:	Input 15 - -----------
ANALOG 16:	Input 16 - -----------
DIGITAL 1:	Output 1 - Solar Pump1
DIGITAL 2:	Output 2 - Solar Pump2
DIGITAL 3:	Output 3 - Heat C Pump1
DIGITAL 4:	Output 4 - Heat C Pump2
DIGITAL 5:	Output 5 - Req.Boiler
DIGITAL 6:	Output 6 - ST Load Pump
DIGITAL 7:	Output 7 - WW Load Pump
DIGITAL 8:	Output 8 - Mix Heat C1
DIGITAL 9:	Output 9 - Mix Heat C1
DIGITAL 10:	Output 10 - Mix Heat C2
DIGITAL 11:	Output 11 - Mix Heat C2
DIGITAL 12:	Output 12 - -----------
DIGITAL 13:	Output 13 - -----------

=> as well as the keyboard shortcut "*Ctrl + right arrow*" insert the parameter selected in the <available parameters> list in the selected position in the data record. The cursor is then automatically placed in the next position in the data record.

<= as well as the button "*Delete*" remove the parameter highlighted in the data record. The cursor is then automatically placed in the next position in the data record.

Delete all Deletes all parameters in the selected data record.

Load defaults Loads the default settings for the selected data record.

Network... Basic, global settings for the CAN network (device node number, enable, etc.). These settings can also be set when parameterizing the network inputs and network outputs.

Creating functional data (Export to translator)

File → Export to translator → Functional data…

Compiler

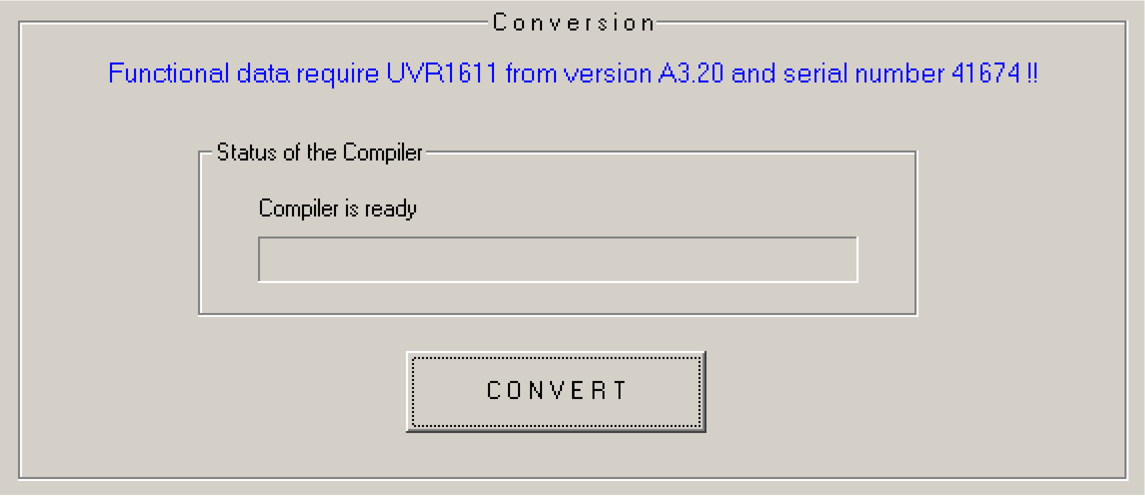
In order to import the configuration created in T.A.P.P.S. into the respective controller, the logical circuit diagram in the drawing must be converted to a respective data format (functional data).

Error-List

For faulty or invalid links in the drawing, the error log is displayed with a respective error message. Functional data can only be created when all links are valid and complete.

Minimum requirements for function data of the UVR1611

Operating system A2.28 should preferably be used for older UVR1611 units (serial number < 41674 and factory operating system < A3.00). An operating system A3.xx should only be used with these units if this is absolutely necessary, i.e. when the functionality of this new version is required. In particular, A3.xx operating systems cause a markedly slower reaction time with older units during operation.
So that TAPPS can be used to produce function data for the current operating system as well as for version A2.28, the minimum requirements (operating system and serial number) are determined for the function data generated from the drawing and displayed in the compiler window.


Conversion

Functional data require UVR1611 from version A3.20 and serial number 41674 !!

Status of the Compiler

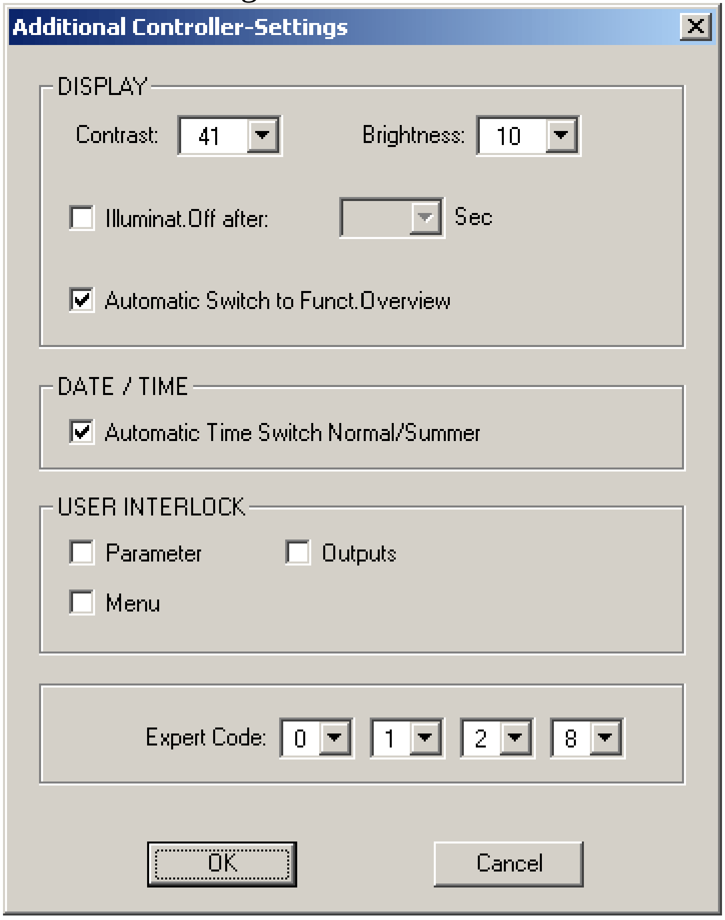
Compiler is ready

Select target path

Before functional data (*.dat) is created by the compiler, the path and the name of the target file can be changed. Contrary to the drawing, the file name may be longer than 8 characters. The selected target path is also used as the basic setting for future projects.

User settings

Additional settings can be made for the controller here.



Additional Controller-Settings

DISPLAY

Contrast: 41 Brightness: 10

☐ Illuminat.Off after: Sec

☑ Automatic Switch to Funct.Overview

DATE / TIME

☑ Automatic Time Switch Normal/Summer

USER INTERLOCK

☐ Parameter ☐ Outputs

☐ Menu

Expert Code: 0 1 2 8

User interlock:

Parameter – The user can henceforth only change enabled parameters in the function overview.

Outputs – In the "Outputs" menu, the user can no longer switch these to manual mode.

Menu – For the client and technician, only the function overview is henceforth visible. The menu is hidden.

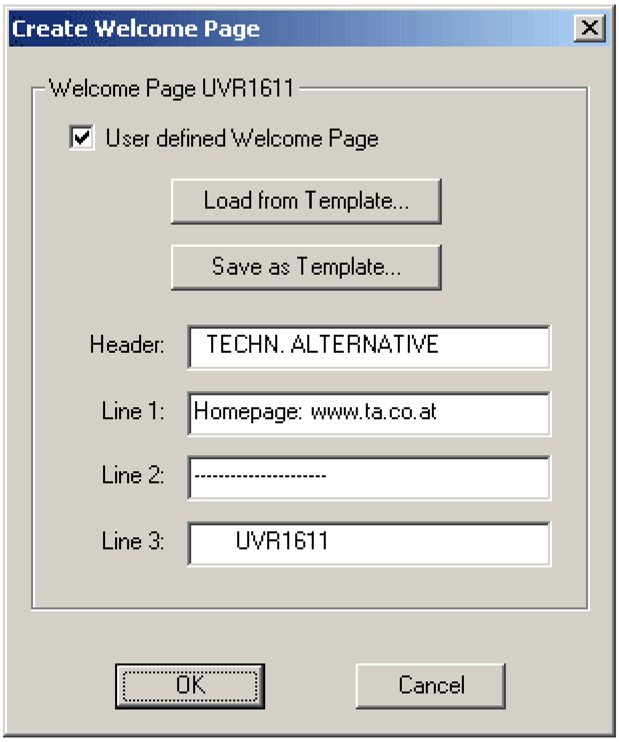
Expert Code:

The code for expert mode can also be changed. This code also applies as basic setting for future projects.

Network settings

Basic settings for the CAN network (node number, enable, etc.), which could already have been made during parameterizing of the network variables or the CAN data logging.

Welcome page



☑ User defined Welcome Page

Header: TECHN. ALTERNATIVE

Line 1: Homepage: www.ta.co.at

Line 2: ----------------------

Line 3: UVR1611

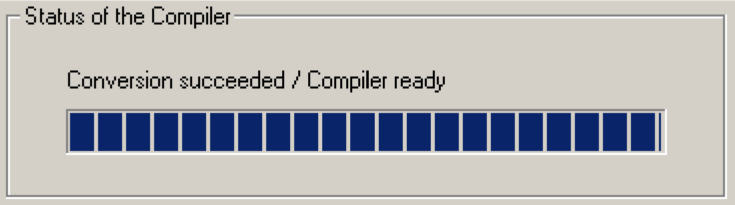
T.A.P.P.S. offers the possibility to create a user-defined welcome page (version page) for the UVR1611.
To this end, 4 rows each with a maximum of 22 characters are available.

If the same welcome page is to be used for further programmings, it can be saved as a template.

If no user-defined welcome page was set, it will generally remain unchanged on the controller.

CONVERT

The selected file is created from the logical circuit diagram in the drawing. This file includes the configuration in a data format with corresponds to the controller. This functional data (*.dat) can subsequently be imported into the UVR1611with the help of the *Memory Manager* program and a Bootloader.



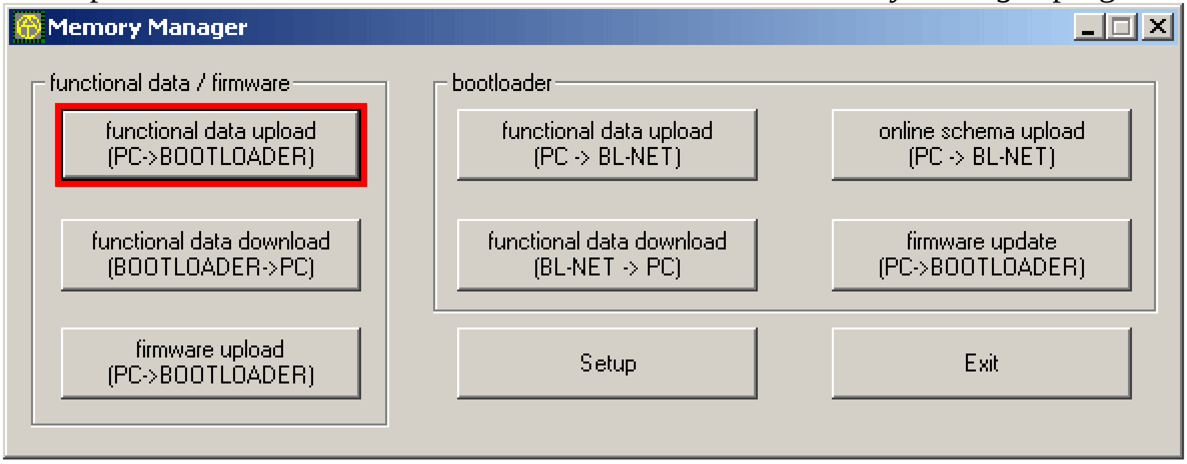
Conversion succeeded / Compiler ready

Close Once the functional data have been successfully created, the program window can be closed.

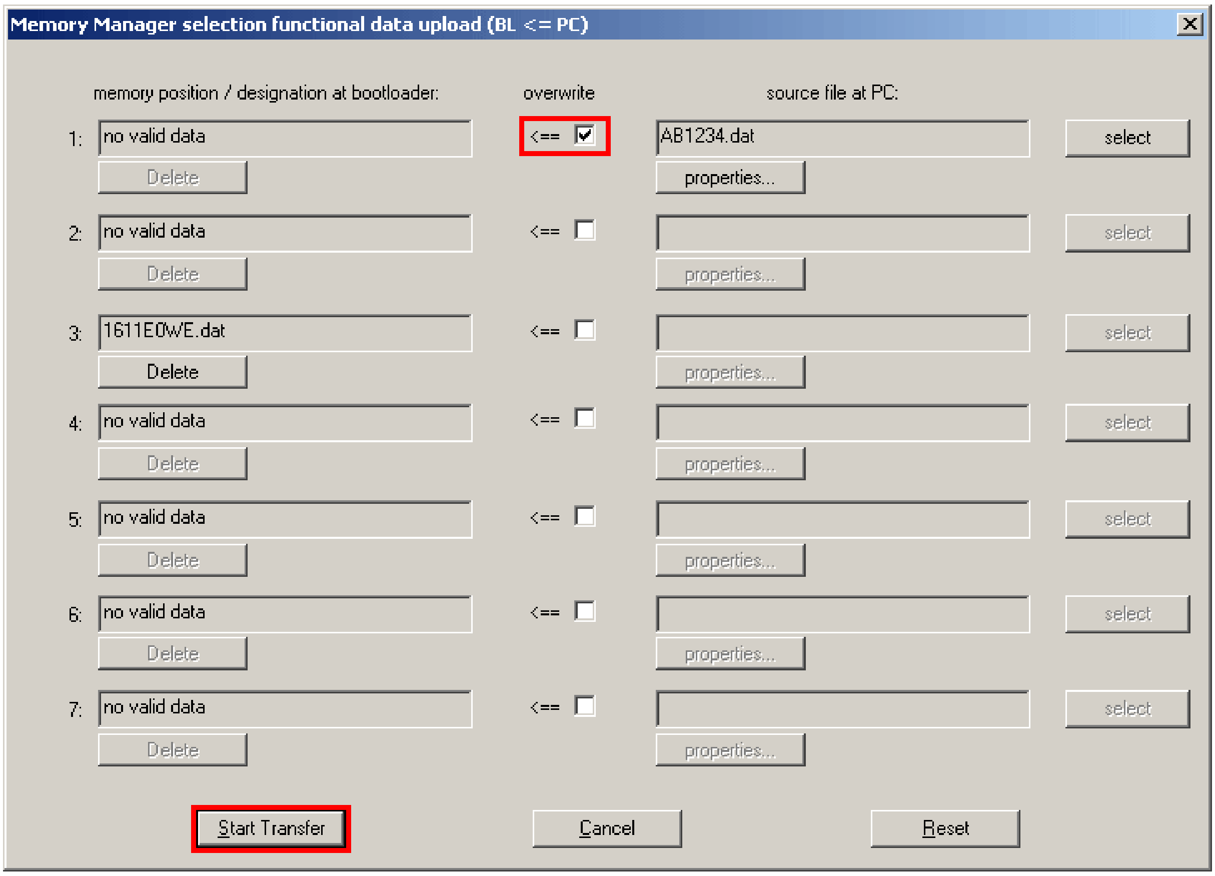
Import configuration to the controller

The functional data (*.dat) created with T.A.P.P.S. from the logical circuit diagram of the drawing can be imported to the Bootloader with the *Memory Manager* program. The Bootloader is a device which represents the interface between PC and controller. The functional data saved on the Bootloader can be imported to the respective controller in a further step.

1. Import the functional data to the Bootloader with the *Memory Manager* program.


Memory Manager
functional data / firmware
functional data upload (PC->BOOTLOADER)
functional data download (BOOTLOADER->PC)
firmware upload (PC->BOOTLOADER)
bootloader
functional data upload (PC -> BL-NET)
online schema upload (PC -> BL-NET)
functional data download (BL-NET -> PC)
firmware update (PC->BOOTLOADER)
Setup
Exit

The Bootloader has 7 memory slots in order to save several functional data at the same time. The functional data already saved in the Bootloader is displayed in the left column.


Memory Manager selection functional data upload (BL <= PC)

	memory position / designation at bootloader:	overwrite	source file at PC:	
1:	no valid data Delete	<== ☑	AB1234.dat properties...	select
2:	no valid data Delete	<== □	 properties...	select
3:	1611E0WE.dat Delete	<== □	 properties...	select
4:	no valid data Delete	<== □	 properties...	select
5:	no valid data Delete	<== □	 properties...	select
6:	no valid data Delete	<== □	 properties...	select
7:	no valid data Delete	<== □	 properties...	select

After selecting the requested memory slot and the source file, this functional data are imported to the Bootloader by pressing start transfer .

2. The functional data saved on the Bootloader can be imported to the UVR1611 either via infrared or CAN interface. This data transfer is possible in Expert operation mode. The controller must also be prepared for this, as specified below:

```
        MENU
----------------------
          :
Data Administration  ◄
```

Menu for data transfer with the Bootloader

```
 DATA ADMINISTRATION
----------------------
          :
DATA <=> BOOTLOADER:
Upload Data:
CONTR. => BOOTLD.
Download Data:
BOOTLD. => CONTR.    ◄
```

Backup functional data of the controller on the Bootloader

Import functional data from the Bootloader to the controller

```
  BOOTLD. => CONTR.
----------------------
DATA SOURCE: Bootld.
Storage Point:  1
TARGET: Controller
Functional Data
Overwrite?      yes
Factory Settings
Overwritten?    no
          :
DOWNLOAD DATA
REALLY START ?  yes  ◄
```

Memory slot of the functional data on the Bootloader

Overwrite the functional data of the controller

The functional data can simultaneously be saved as factory setting.

The controller switches to receipt mode

```
       NODE: 63

 READY FOR DOWNLOAD
```

The controller is ready for data transfer which is started by pressing the START key on the Bootloader.

```
       NODE: 63

     PROGRAMMING
```

Data transfer is carried out.

```
       NODE: 63

 SUCCESSFUL DOWNLOAD
```

The functional data was successfully loaded to the controller.

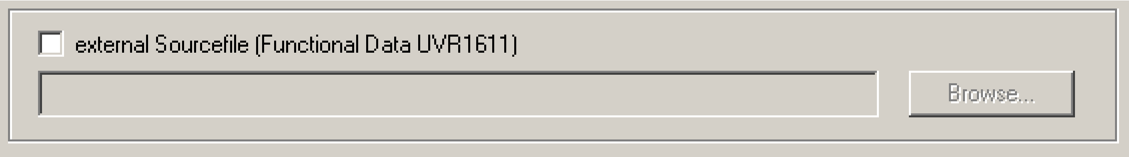
Parameter documentation

File → Export to translator → Functional data…

Parameter-Docu

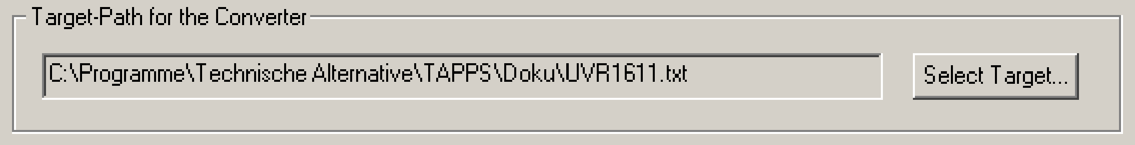
T.A.P.P.S. provides the option of creating a text file (*.txt) with a list of all setting parameters of the programming.
Such a parameter list can also be created for functional data which are exported from a UVR1611.

1. In order to create a documentation for functional data which were exported from a UVR1611, these functional data (*.dat) must be stated as an external source file.
   If no external source file is stated, a documentation of the opened drawing is created.



☐ external Sourcefile (Functional Data UVR1611)

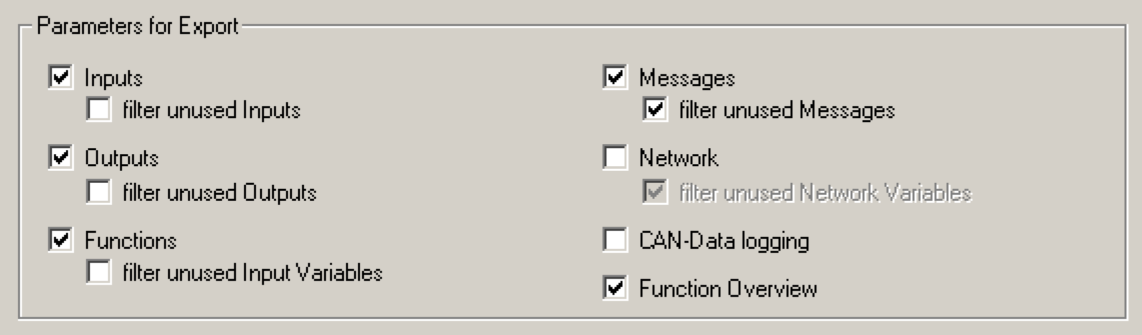
2. Select the target path of the target file (*.txt) to be created.



Target-Path for the Converter

C:\Programme\Technische Alternative\TAPPS\Doku\UVR1611.txt

3. Select the parameters to be exported for the documentation.

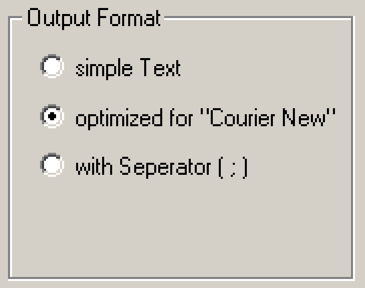


Parameters for Export

☑ Inputs
☐ filter unused Inputs
☑ Outputs
☐ filter unused Outputs
☑ Functions
☐ filter unused Input Variables

☑ Messages
☑ filter unused Messages
☐ Network
☑ filter unused Network Variables
☐ CAN-Data logging
☑ Function Overview

4. Select output format

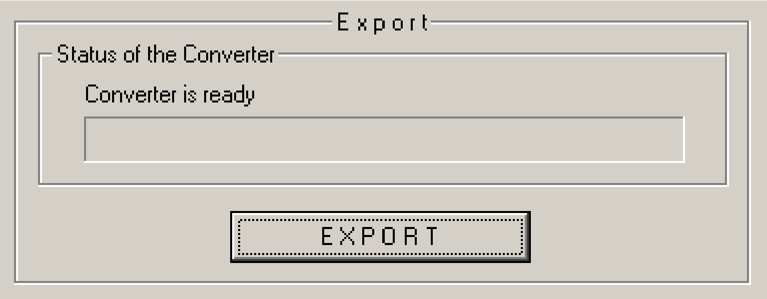


Output Format

○ simple Text
● optimized for "Courier New"
○ with Seperator (;)

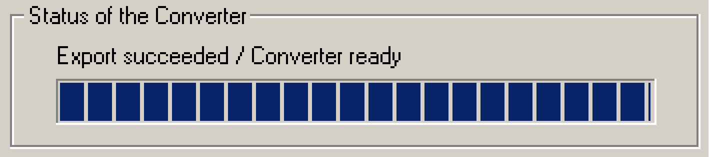
Several output formats can be selected, while the *optimised for "Courier New"* format is recommended for a documentation which is printed with a text editor. This format is optimised for fonts with letters of the same width.

5. EXPORT

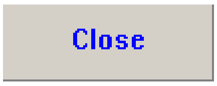


Converter is ready

From the selected source file (external functional data or the opened drawing) the text file (*.txt) is created which contains a list of the requested setting parameters from programming. This text file can subsequently be saved in a preferred format with any text editor (e.g. Microsoft® Word) and printed.



Export succeeded / Converter ready



Once the parameter list has been successfully created, the program window can be closed.

Parameter import (Import from translator)

File → Import from translator → Functional data…

Parameter-Import

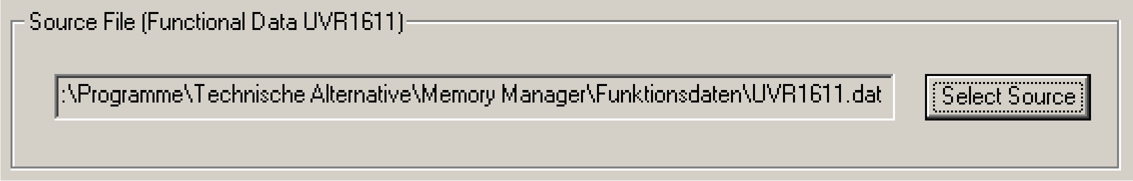
Parameter settings changed on the controller can be imported to the original drawing under certain conditions.

⚠ Importing the settings parameters is only possible if no functions and messages are added or deleted and no links changed in the functional data (*.dat) of the controller and the associated T.A.P.P.S. drawing!

⚠ **No drawing can be created from functional data!**

1. Open the original drawing with which the functional data was created.

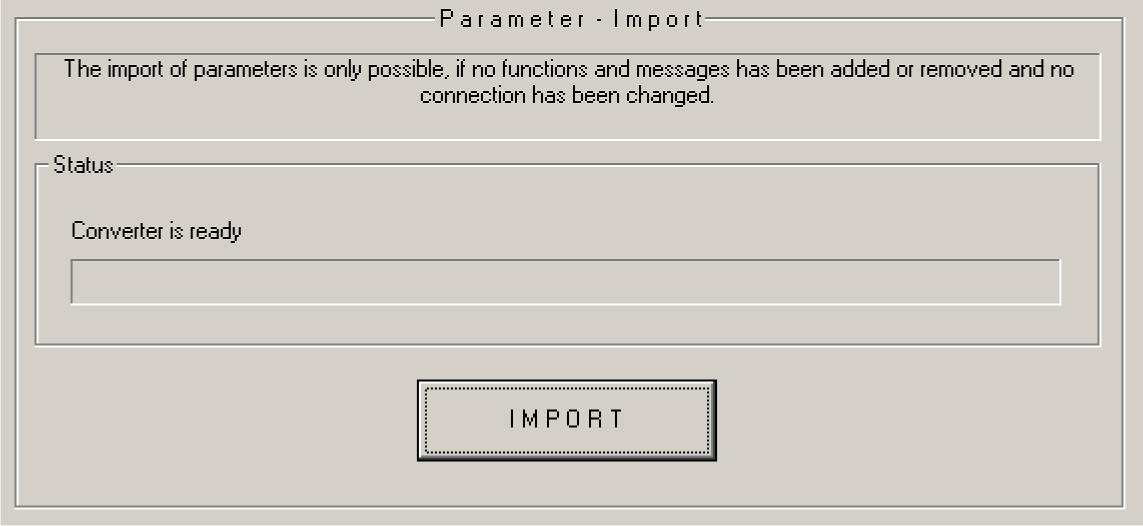
2. Select the functional data (*.dat) exported from the controller as source file for importing the settings parameters.



Source File (Functional Data UVR1611)

:\Programme\Technische Alternative\Memory Manager\Funktionsdaten\UVR1611.dat

3. IMPORT



Parameter-Import

The import of parameters is only possible, if no functions and messages has been added or removed and no connection has been changed.

Status

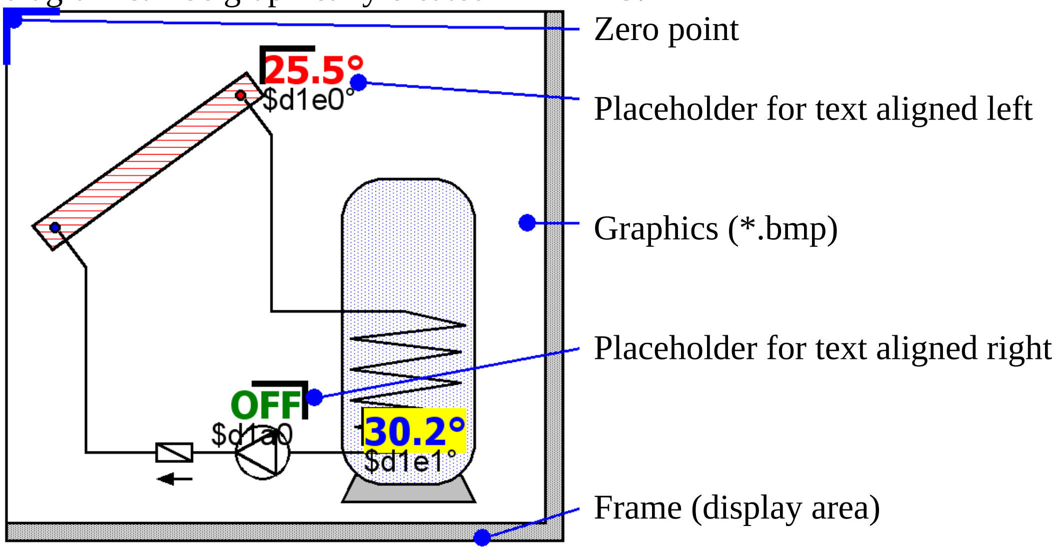
Converter is ready

After successful verification of the compatibility, the parameters are imported from the source file and the drawing updated.

BL-NET online diagram

The Bootloader BL-NET provides the option on an online visualisation where a graphical illustration of the current system conditions can be displayed in a web browser via LAN or the internet.
This online diagram consists of a graphics file (e.g. hydraulics circuit diagram) and the associated HTML file where the display parameters are defined.

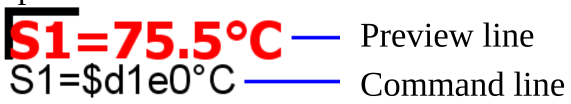
With the components of the <online diagram> library, the HTML file for the online diagram can be graphically created in TAPPS.



Procedure for the graphical design of HTML files:

1. Drag and drop the frame representing the display area from the library into the drawing.
2. Import the chart (e.g. hydraulics circuit diagram) of the online diagram into the drawing as a 24-bit bitmap (*.bmp): File → Import…
   If the chart is in a different file format (e.g. *.gif), the required bitmap can be created using a graphics program (e.g. Microsoft® Paint).
3. The chart must be aligned on the zero point (top left corner) of the frame, as the display parameters would otherwise be displayed in the wrong place for the online diagram. Similarly, the size of the chart must not be changed in TAPPS.
4. Adjust the size of the frame (display area), which is normally identical to the chart, accordingly.
5. Where required, import placeholders for the display parameters (temperatures, output conditions) from library, then position and configure them.
   To this end, placeholders for text aligned left or right are available.

A placeholder has 2 lines which can be edited by double clicking.



♦ **Preview line:** The top line is used as a preview and reflects the position and the appearance of the display parameter.
The following text attributes of this line are imported into the HTML file:

- Font
- Font size (4…20)
- Font style (bold, italic)
- Font colour
- Background colour

The entered text of this line has no influence on the online diagram and serves as a preview only which facilitates positioning the placeholder.

♦ **Command line:** The bottom line is only visible in TAPPS and serves to define the actually displayed text. The syntax of the command line is as follows:

`S1=$d1e0°C`

`S1=`… optional, static text in front of the display value
`$d1e0`… definition of the display value
`°C`… optional, static text after the display value (e.g. measurement value unit)

Umlauts, spaces and special characters (°, ß, ²) are automatically converted into the respective HTML code by TAPPS when creating the HTML file. The “&“, “<“ and “>“ characters must not be used.
The definitions of the display values are replaced in the online diagram by the Bootloader BL-NET by respective, current measurements. *Information on the possible display values and their definitions can be found in the operating instructions of the Bootloader BL-NET.*
A definition of a display value always starts with the “$“ character which must therefore never be used in the static text.
Several display values can be defined in the command line of a placeholder. An exclusively static text can also be placed in the online diagram by using a placeholder.

6. The HTML file for the online diagram can be created from the graphical illustration: File → Export to Translator → Online Diagram…
More information about the HTML file as well as the requirements for the boot loader BL-NET are described on the following page.

The HTML file is created by **TAPPS (from version 1.26)** in a format which makes possible updating of the display values in the online schematic, without having to reload the graphic each time. This format of the **HTML file requires** a boot loader **BL-NET** with an operating system **from version 1.40**!

TAPPS creates the HTML file with the following properties:

- The graphic file (.gif) is loaded by the boot loader BL-NET.
- The display values are automatically updated in an interval of 60 seconds.
- A link "update values" is displayed beneath the graphic, which makes possible manual updating of the display values via a mouse click.

If necessary, this factory setting can be retrospectively altered directly in the HTML file using a text editor as follows:

Graphic file path:

In the factory, the graphic file is loaded by BL-NET. In this case, the variable 'GraphicFile' must be defined as "GRAFIK.gif"!

```
var GraphicFile = "GRAFIK.gif";
```

In order to, for example, reduce the graphic loading time, this can be loaded from a web server rather than from the boot loader. To do this, the variable 'GraphicFile' must be allocated to the corresponding graphic file path.
For example:

```
var GraphicFile = "http://www.website.at/graphic.gif";
```

Automatic updating of the display values:

The variable 'AutoRefreshTime' can be used to define the interval (in seconds), after which the display values in the browser window are automatically updated. An interval time smaller than 30 seconds is not permitted and means that automatic display updating will not take place!

```
var AutoRefreshTime = 60;
```

If automatic updating of the display values is not required, this function can be deactivated by setting the 'AutoRefreshTime' variable to zero.

```
var AutoRefreshTime = 0;
```

Manual updating of the display values:

The variable 'RefreshLink' can be used to define the display text of a link, which is displayed beneath the graphic and which enables manual updating of the display values by a mouse click.

```
var RefreshLink ="Update values";
```

If the link for manual updating of the display values is not required, then it can be hidden by deleting the link text.

```
var RefreshLink ="";
```

FAQs

Is it possible to install several, different language versions of T.A.P.P.S. on one computer?

Yes, but it is not possible to run both versions at the same time.
The different language versions must be installed in an own folder! For example "C:\...\TappsDE" for the German version and "C:\...\TappsEN" for the English version.

T.A.P.P.S. causes a general protection fault when starting!?

***1.** The user of T.A.P.P.S. must have writing access on the computer for the directory of the program and its subfolders!*
***2.** As some printers (especially network printers) as standard printers can cause such problems, try defining another local printer or, for instance, Adobe® Acrobat® PDF Writer or pdf24 creator as standard.*
You can also try as a workaround to hold the Esc key when starting T.A.P.P.S., as soon as the screen splash appears.

T.A.P.P.S. cannot be started correctly?

***1.** The user of T.A.P.P.S. must have write rights to the computer directory containing the program and its sub-directories.*
***2.** Under certain circumstances, T.A.P.P.S. must be installed outside the Windows program folder (e.g.: "C:\Technische Alternative\Tapps\").*
***3.** Under Windows 7, it may be necessary to activate the Windows XP compatibility mode for the application "Tapps.exe". To do this click on the T.A.P.P.S. desktop symbol and call the menu "Properties".*
***4.** T.A.P.P.S. will not run under the 64 bit version of the Windows operating system.*

Opening a drawing is not possible!?

The file name must not exceed 8 characters.
The file name and the entire path where the drawing is saved must also not have any special characters or umlauts.

When opening a drawing an error is made when loading the parameterisation!?

The parameter file belonging to the parameter file [file name].par could not be loaded. Close the drawing without saving any changes and ensure that the associated parameter file [file name].par is saved in the same folder as the drawing [file name].eng.

How can the library window <selection> be again displayed, once it was mistakenly closed?

The library window can be displayed again in the menu list via “Window” → “Selection”.

Does a hydraulics diagram need to be drawn in T.A.P.P.S. to create a configuration (functional data)?

No, a drawn hydraulics diagram is only used for documentation.
The configuration of the UVR1611 is determined by a logical circuit diagram and its components (inputs/outputs, functions, etc.) only.

Why are inputs/outputs and network inputs still visible on the controller although they have been deleted from the drawing?

If inputs/outputs or network inputs are deleted from the drawing, they may not have any influence on the controller algorithm, but they are still defined. Unusable inputs/outputs and network inputs must therefore be defined in their parameterisation as unused.
An unused but still defined network input would, for instance, cause a timeout error.

Can T.A.P.P.S. be used to create functional data for a UVR1611 with a foreign language operating system?

Yes, the functional data is independent of the language of the controller’s operating system. The selected names are automatically displayed in the respective language on the controller.

Can a drawing be edited with a different language version of T.A.P.P.S.?

Yes, the texts of the circuit diagram’s library elements are automatically converted when opening the drawing to the respective language.

Are the symbols of the circuit diagram not automatically translated when opening a different language drawing!?

Automatic translating of a circuit image’s symbols is only possible for drawing objects from a library of T.A.P.P.S. from version 1.20. Older symbols cannot be translated and therefore remain unchanged.

Can functional data exported from the controller be edited with T.A.P.P.S.?

No, as a rule the configuration file (.dat) with the functional data exported from controller serves as a backup copy which cannot be edited. With T.A.P.P.S. a text file (*.txt) can be created as a documentation of this functional data which includes a list of all setting parameters.*

Can changed parameter settings be imported in the associated T.A.P.P.S. drawing?

Under certain circumstances changed controller settings can be imported into the original drawing.

Importing the changes settings parameters is only possible if no functions and messages are added or deleted and no links changed in the functional data (.dat) of the controller and the associated T.A.P.P.S. drawing!*

Can the parameter settings of a configuration created on the controller and redrawn in T.A.P.P.S. be imported into the drawing from the functional data of the controller?

No, importing settings parameters is only possible for the T.A.P.P.S. drawing with which the controller configuration was originally created.

The drawing must be accordingly compatible with the functional data (.dat) to perform an import. This cannot be guaranteed when a configuration created on the controller is redrawn on the PC.*

Can the operating mode of a programming created in T.A.P.P.S. be simulated?

A simulation is not possible with T.A.P.P.S. But a simulation board is available for the UVR1611 with which the sensors at the inputs of the controller can be simulated.

Technische Alternative elektronische Steuerungsgerätegesellschaft m. b. H.
A-3872 Amaliendorf, Langestr. 124
Tel ++43 (0)2862 53635
Fax ++43 (0)2862 53635 7
E-Mail: mail@ta.co.at
--- www.ta.co.at ---