

SIEMENS

SIMADYN D

Configuring Instructions
T400

Technology Module T400



Edition		Edition
1	Technology Module T400	09.99

The reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, including rights created by patent grant or registration of a utility model or design, are reserved.

We have checked the contents of this Manual to ensure that they coincide with the described hardware and software. However, deviations cannot be completely ruled-out, so we cannot guarantee complete conformance. However, the information in this document is regularly checked and the necessary corrections included in subsequent editions. We are thankful for any recommendations or suggestions.

Contents

Definitions	2
Danger and warning notes	2
1. Configuring	3
2. Installation	3
3. SIMATIC Manager	5
4. HW Config	6
5. CFC	10
5.1. Calling-up blocks.....	11
5.2. Defining the sampling time.....	13
5.3. Compilation.....	15
5.4. Loading.....	15
5.5. Test mode.....	19
5.6. Sheet bar/margin.....	24
6. Configuring instructions	27
6.1. General information.....	27
6.2. Analog inputs.....	28
6.3. Analog outputs.....	29
6.4. Digital inputs.....	30
6.5. Digital inputs/outputs.....	31
6.6. Absolute value encoders.....	33
6.7. Incremental encoders.....	36
Note.....	37
6.8. Communications.....	39
6.8.1. General information.....	39
6.8.2. Serial coupling to the configuring PC.....	42
6.8.3. Coupling to additional modules (e.g. CUx, ADB, CBP, CBx).....	43
6.8.3.1. In the electronics box of a drive.....	44
6.8.3.2. In the SRT400 technology subrack.....	49
6.8.4. USS.....	53
6.8.4.1. USS master.....	53
6.8.4.2. USS slave.....	55
6.8.5. Peer to peer.....	58
6.8.6. Simolink (with SLB).....	59
6.8.7. Profibus (with CB1 or CBP).....	59
6.9. Parameterization.....	59
6.9.1. Application.....	59
6.9.2. Configuring a parameter.....	60
6.9.3. Function blocks required.....	61
6.9.4. BiCo technology: Changing connections using parameters.....	63
6.9.5. Special features and restrictions for parameters and BiCo.....	63
6.10. Permanently saving values (non-volatile).....	64
7. Basic application software	64

Definitions

NOTE

The information in these instructions does not purport to cover all details or variations in equipment, nor to provide for every possible contingency to be met in connection with installation, operation or maintenance. Should further information be desired or should particular problems arise which are not covered sufficiently for the purchaser's purposes, please contact your local Siemens office. Further, the contents of these instructions shall neither become a part of nor modify any prior or existing agreement, commitment or relationship. The sales contract contains the entire obligation of Siemens. The warranty contained in the contract between the parties is the sole warranty of Siemens. Any statements contained herein do not create new warranties nor modify the existing warranty.

Qualified personnel

For the purpose of these instructions and product labels, qualified personnel are personnel who are familiar with the installation, mounting, start-up and operation of the equipment and the hazards involved. He or she must have the following qualifications:

1. Trained and authorized to energize, de-energize, clear, ground and tag circuits and equipment in accordance with established safety procedures.
2. Trained in the proper care and use of protective equipment in accordance with established safety procedures.
3. Trained in rendering first aid

Danger and warning notes



DANGER

For the purpose of these instructions and product labels, „Danger“ indicates **death, severe personal injury and/or substantial property damage will** result if proper precautions are not taken.



WARNING

For the purpose of these instructions and product labels, „Warning“ indicates **death, severe personal injury or property damage can** result if proper precautions are not taken.



CAUTION

For the purpose of these instructions and product labels, „Caution“ indicates that **minor personal injury or material damage can** result if proper precautions are not taken.

NOTE

For the purpose of these instructions, „Note“ indicates information about the product or the respective part of these instructions which is essential to highlight.



CAUTION

This board contains components which can be destroyed by electrostatic discharge. Prior to touching any electronics board, your body must be electrically discharged. This can be simply done by touching a conductive, grounded object immediately beforehand (e.g. bare metal cabinet components, socket protective conductor contact).



WARNING

Electrical equipment has components which are at dangerous voltage levels. If these instructions are not strictly adhered to, this can result in severe bodily injury and material damage. It is especially important that the warning information in the relevant Operating Instructions is strictly observed.

1. Configuring

The T400 can be configured using the Windows 9x/NT-based graphic D7-ES configuring tool. This allows even complex customer-specific supplementary functions to be easily implemented in drives. Complete standard software packages are available for applications which are frequently used.

In the following, a T400 is configured using a small example, in order to explain the configuring process. It is assumed that a T400 has been inserted in the righthand slot of an electronics box of a MASTERDRIVE or DC-MASTER. A communications module is not inserted, i.e. the center slot of the electronics box is empty.

Not all of the functions of the configuring interface will be discussed. The diagrams were generated with version 5.0 of the configuring software.

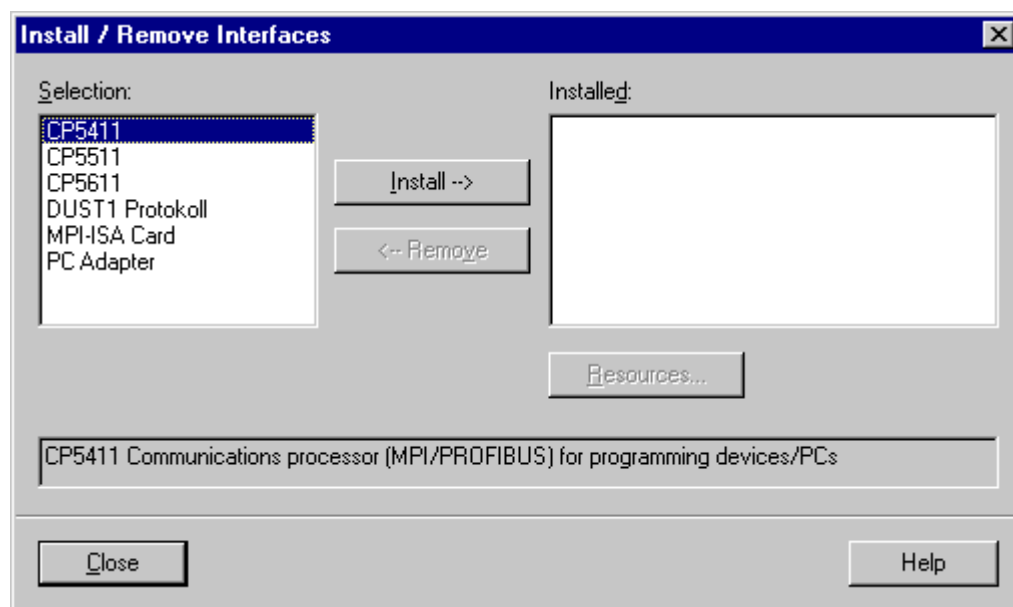
2. Installation

The following software packages must be installed when configuring a T400: STEP7, CFC and D7-SYS. These three packages form the D7-ES. The software must be installed in the following sequence:

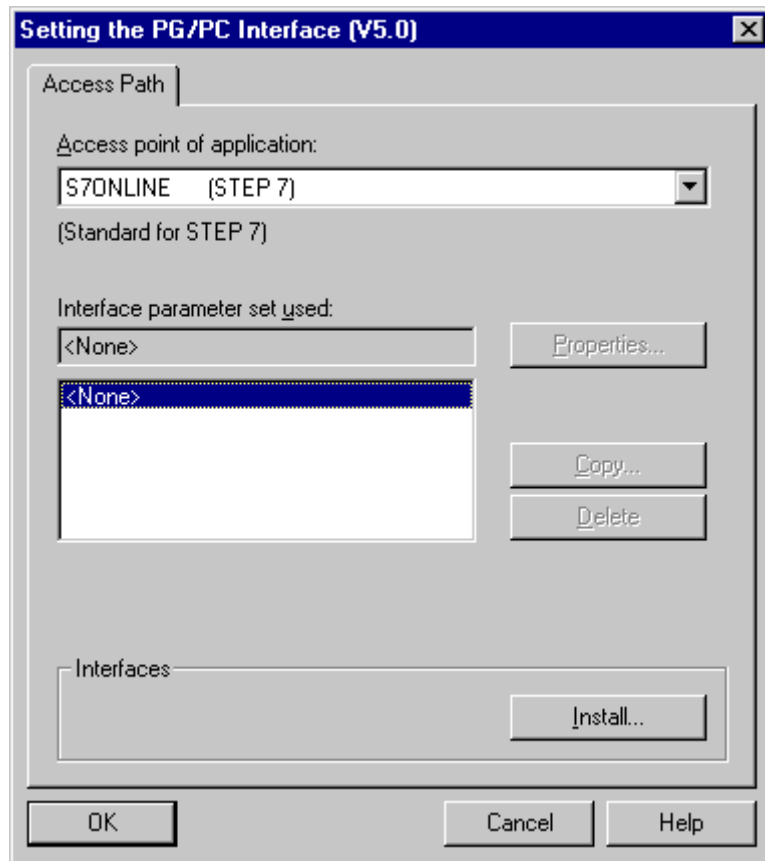
- STEP 7
- CFC
- D7-SYS

Authorization is required for STEP 7 and CFC. The software is de-installed in the inverse sequence.

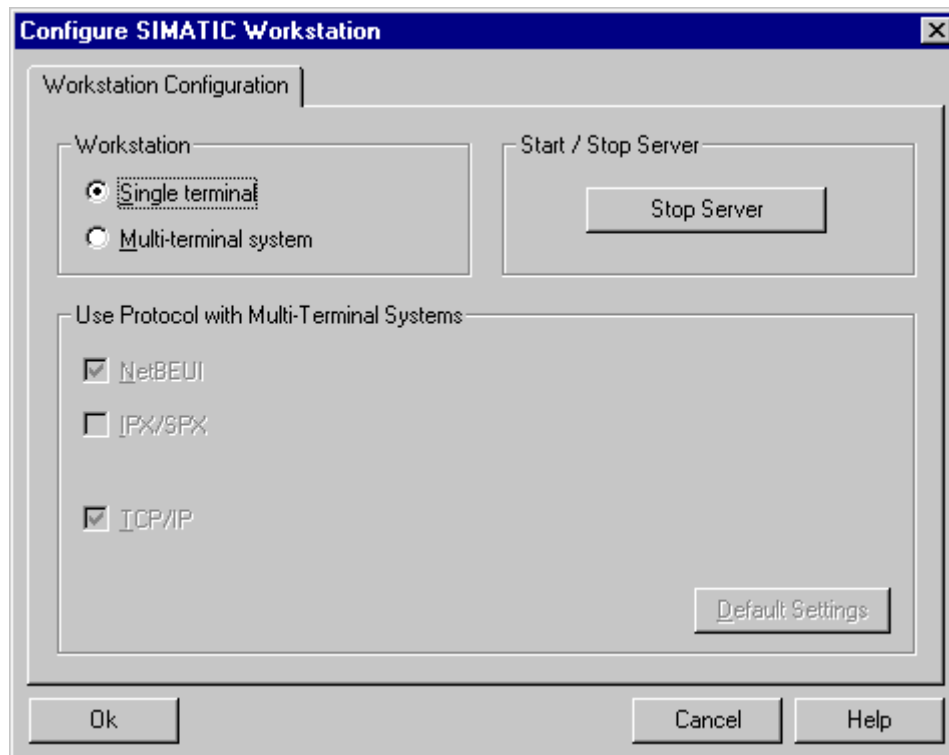
When installing STEP7, you'll be prompted for the online interface. Initially, you do not have to install anything for the T400:



Simply select **Close** and then, in the following window, **OK**:




After installing the software, you can configure the **SIMATIC Workstation** in the control panels. If you are not working as member of a networked team on projects, then you should configure this as a single terminal.

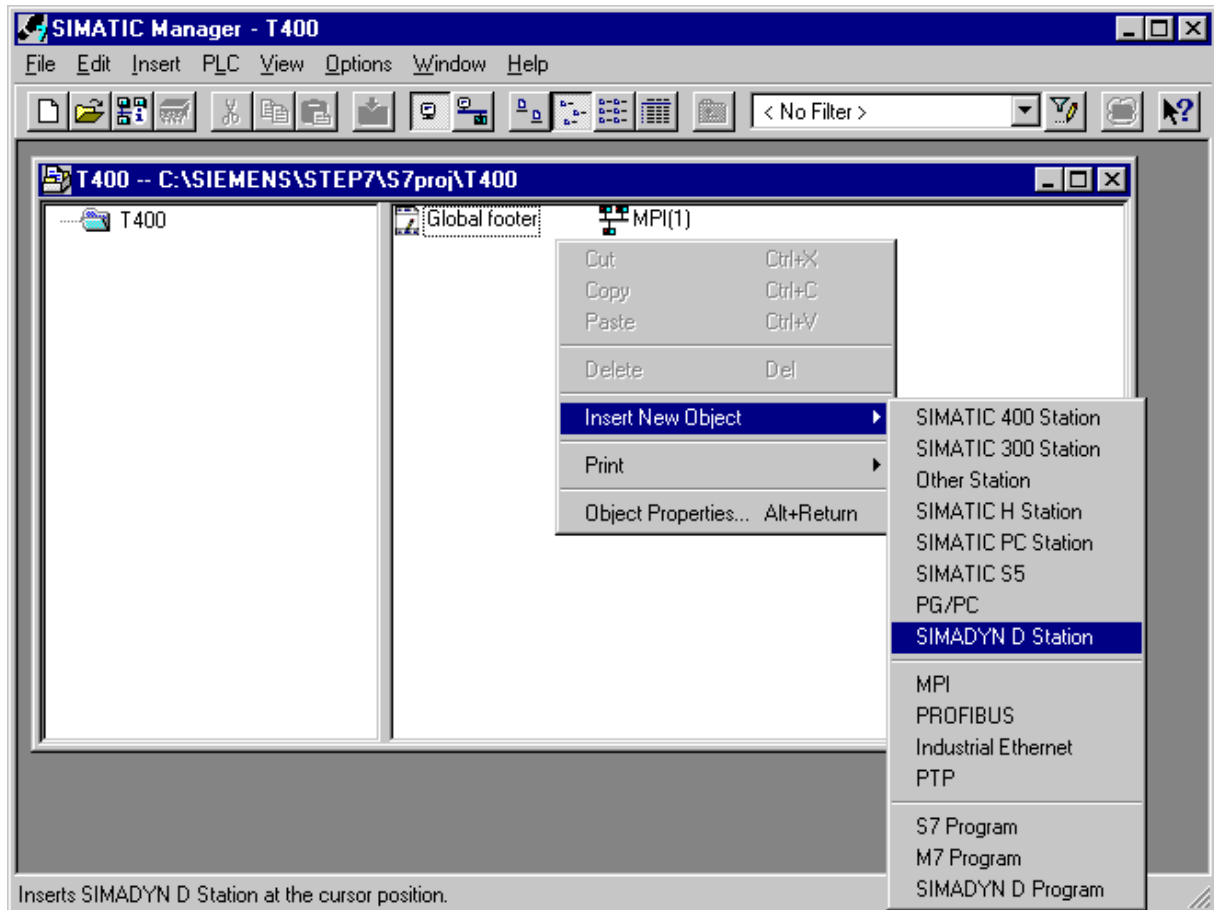


3. SIMATIC Manager




The SIMATIC Manager is started by clicking on the symbol . When first called, it can take some time to open-up. If the STEP7 Assistant is to be opened in STEP7, exit this, as it is not suitable for SIMADYN D.

Create a new project via **File New** (e.g. T400). Using the righthand mouse key, click in the righthand white field in order to obtain the context menu, and then select **Insert New Object** and **SIMADYN D Station**.

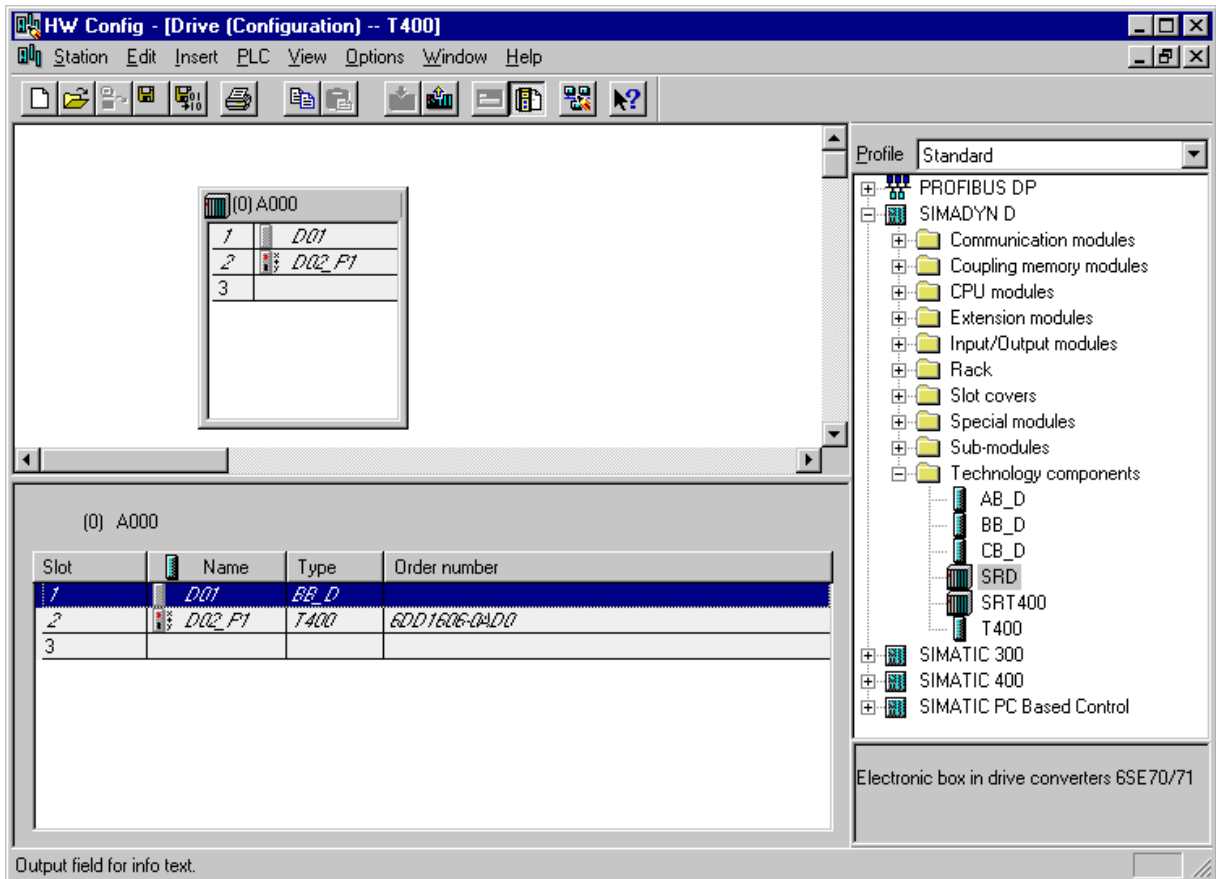


A **SIMADYN D Station** is inserted. The name of the new station is marked and can be overwritten, e.g. Drive. Click twice on the new station in order to change the view.

You'll now see the **Hardware**  symbol in the righthand window. HW Config is opened by clicking on it twice. HW Config is used to create the hardware configuration.

4. HW Config

Open the **SIMADYN D** and **Technology Components** entry in the HW Config **Hardware Catalog** split window (in the lower diagram, on the right). Then drag the **SRD** object, representing the electronics box, into the upper lefthand window. All of the function blocks which are available will now be imported. This can take a few minutes. After the function blocks have been imported, the electronics box, with the basic drive module CUx is schematically represented at slot 1 and the T400 at slot 2. This is displayed according to slot numbers, and not according to the physical arrangement (with slot 3 at the center) !



The settings required for this module can be made by double-clicking on an inserted module in HW Config.

For CUx, which is represented in HW Config as type **BB_D**, only the name, under which this module from T400 can be addressed, should be changed, e.g. to CU

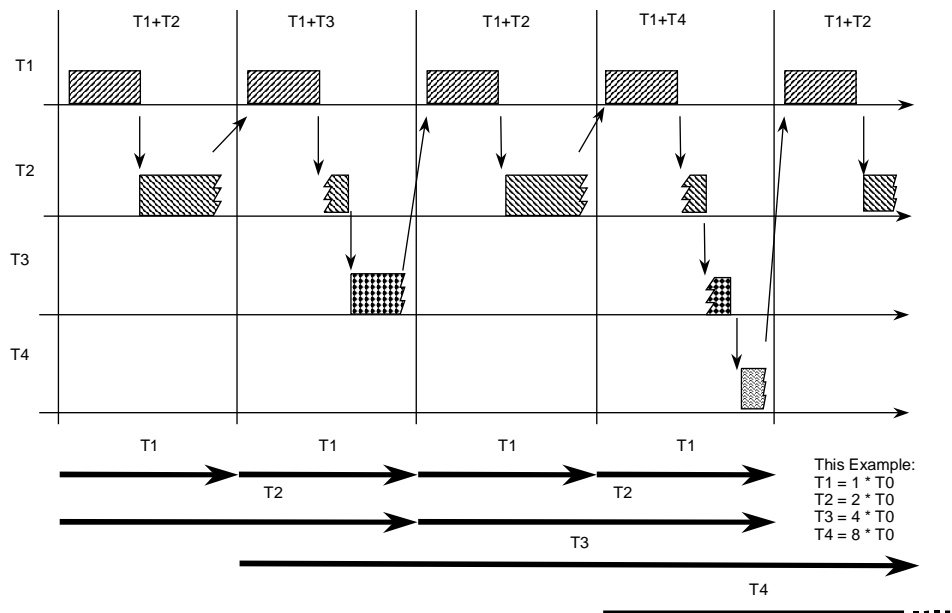
For the T400, there are several tabs to set the options:

Tab	Parameters which can be set
General	The module name can be entered, e.g. T400
Addresses	The hardware input/output names are entered here under which they can then be later addressed in the CFC.
Basic clock cycle	The basic clock cycle is entered here.
Cyclic tasks	5 sampling times are selected, which are formed from the basic sampling time to the power of 2.
Interrupt tasks	Interrupt tasks can be defined if interrupt-triggered tasks are to be used.
STOP	Not relevant for T400.

Background information:

- Sampling time

Five cyclic sampling times can be defined for SIMADYN D. They are always the basic sampling time T_0 to the power of 2. T_1 is always the shortest, T_5 the slowest sampling time. The blocks are subsequently configured in these sampling times. The individual tasks being called are shown in the following diagram:



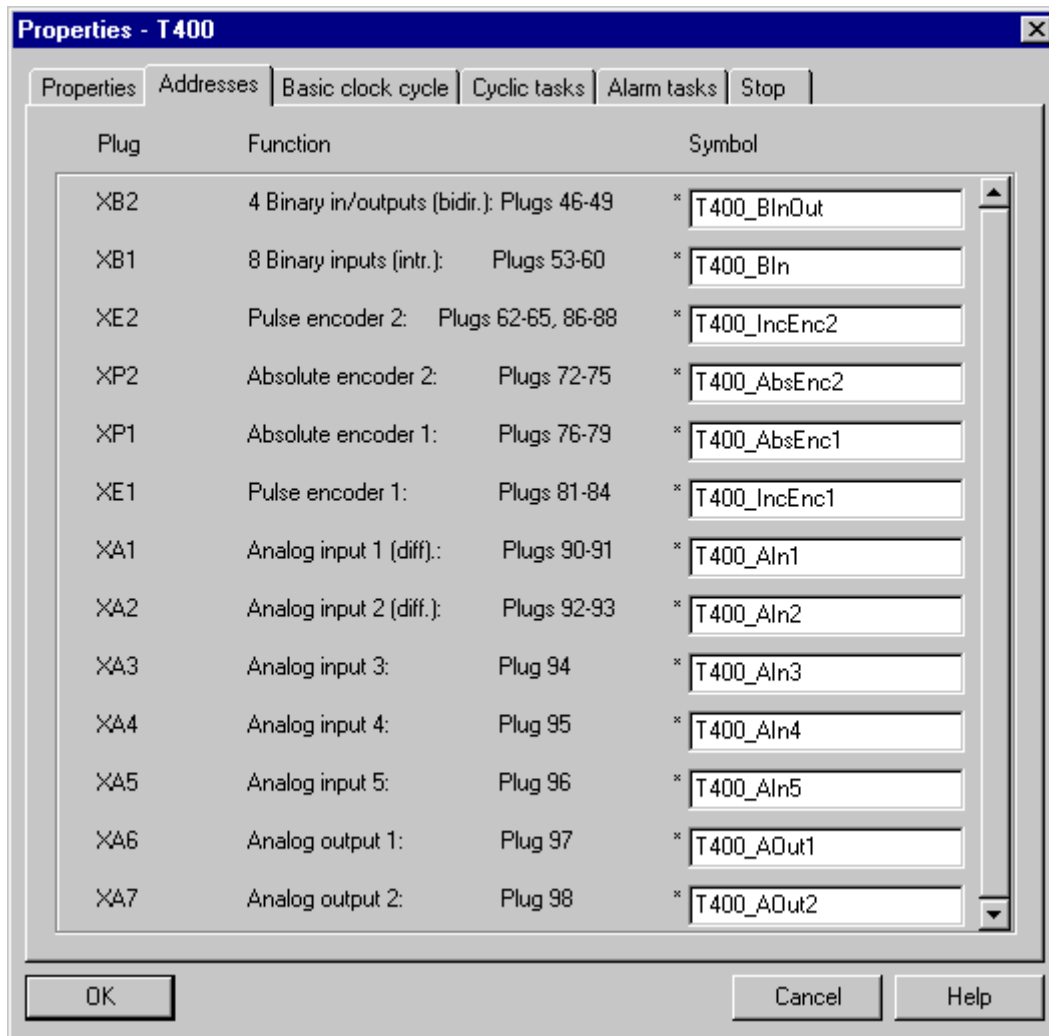
Thus, it is possible to compute individual sub tasks in different sampling times, in order that the computational performance is not unnecessarily used. For example, such a distribution could look like this.

Sampl. time	Functions
T1	1 ms Closed-loop speed control
T2	4 ms Higher-level closed-loop position control
T3	8 ms Communications
T4	32 ms Reading-in the digital signals of an input field
T5	256 ms System blocks

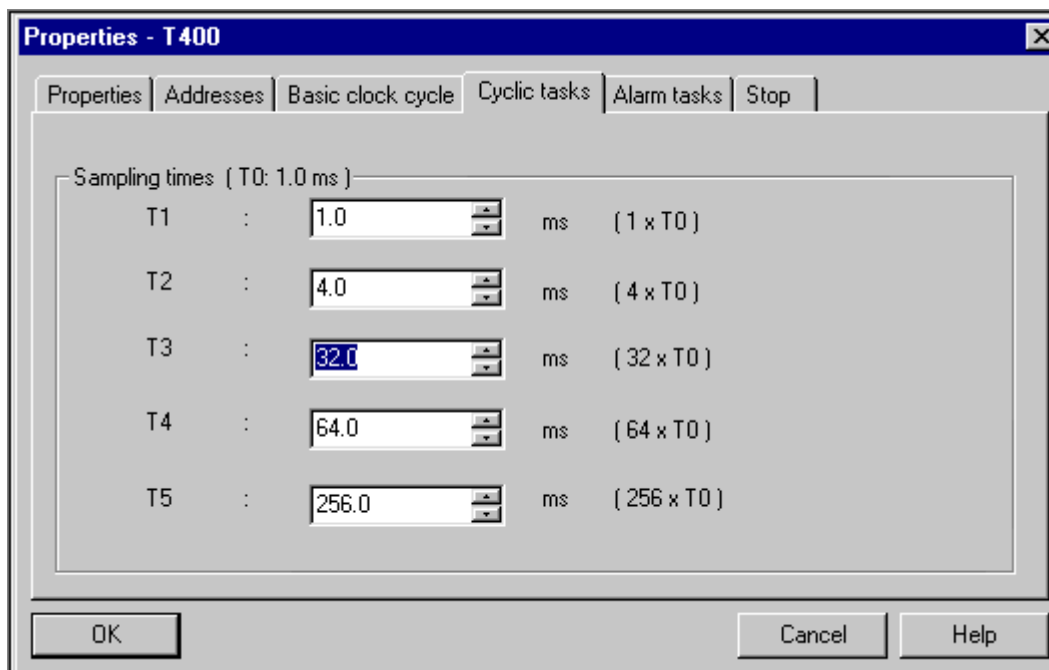
- Addresses


Within SIMADYN D, all hardware inputs and outputs are addressed via symbolic addresses. The inputs and outputs are designated in HW Config, and are later available in CFC under these names. Inputs/outputs, which are not designated in HW Config, will not be available in CFC.

For example, a completed address field could look like this: (refer to the next page)





And the tab of the cyclic sampling times as follows (if the basic clock cycle, T0=1 ms):




The hardware configuration has therefore been completed, and is saved using the  icon or **Station | Save and Compile**. HW Config can now be closed.

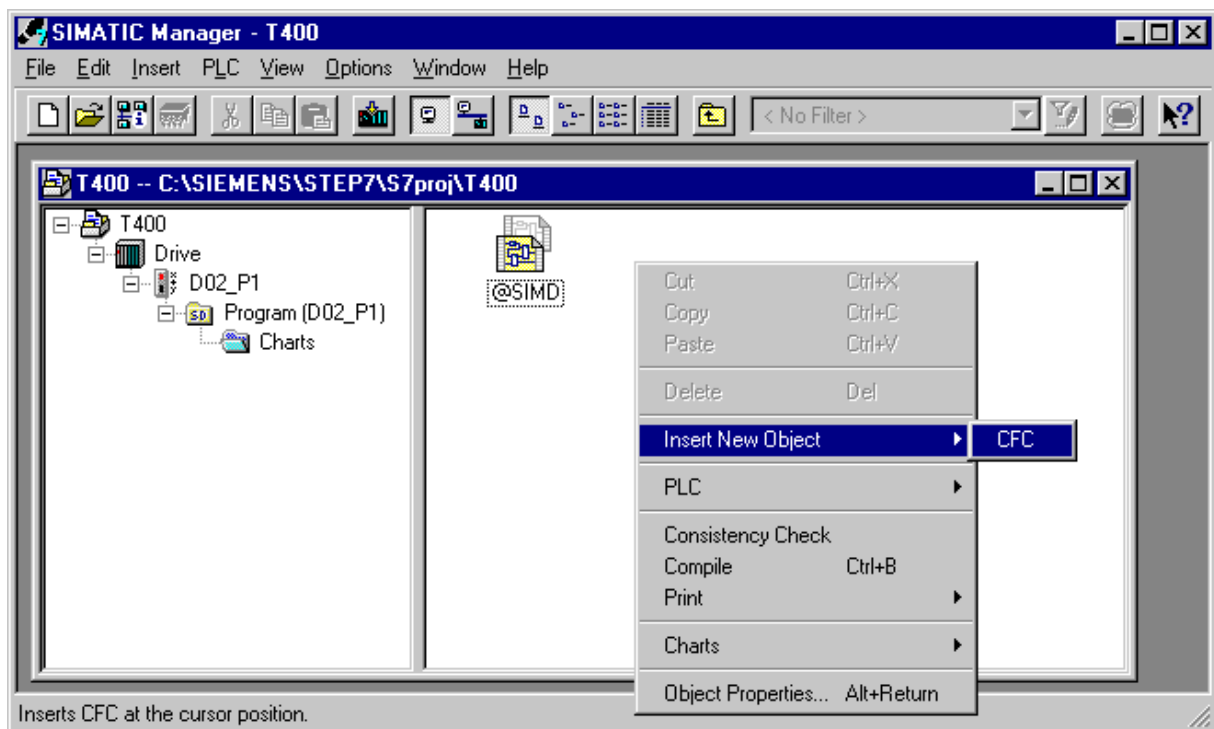
The configured CPU can now be seen in the SIMATIC Manager.

After double-clicking on the symbol of the CPU , the display indicates that a program is available for this CPU.

By double-clicking again on the SIMADYN D program symbol , this is opened and the chart container appears.

If this chart container is opened by double-clicking on the symbol , the individual CFC charts can be seen. One chart is available as standard. It involves the system chart @SIMD, which is responsible for several internal functions, e.g. controlling the diagnostic LEDs. It should not be deleted. However, the system can also run without a system chart; only the functions which are implemented in it are missing.

Then click with the righthand mouse key (context menu) in the white righthand field, and select the insert **New Object** and **CFC**




A new CFC chart is inserted, the name of the new chart is marked and can be overwritten, e.g. Example

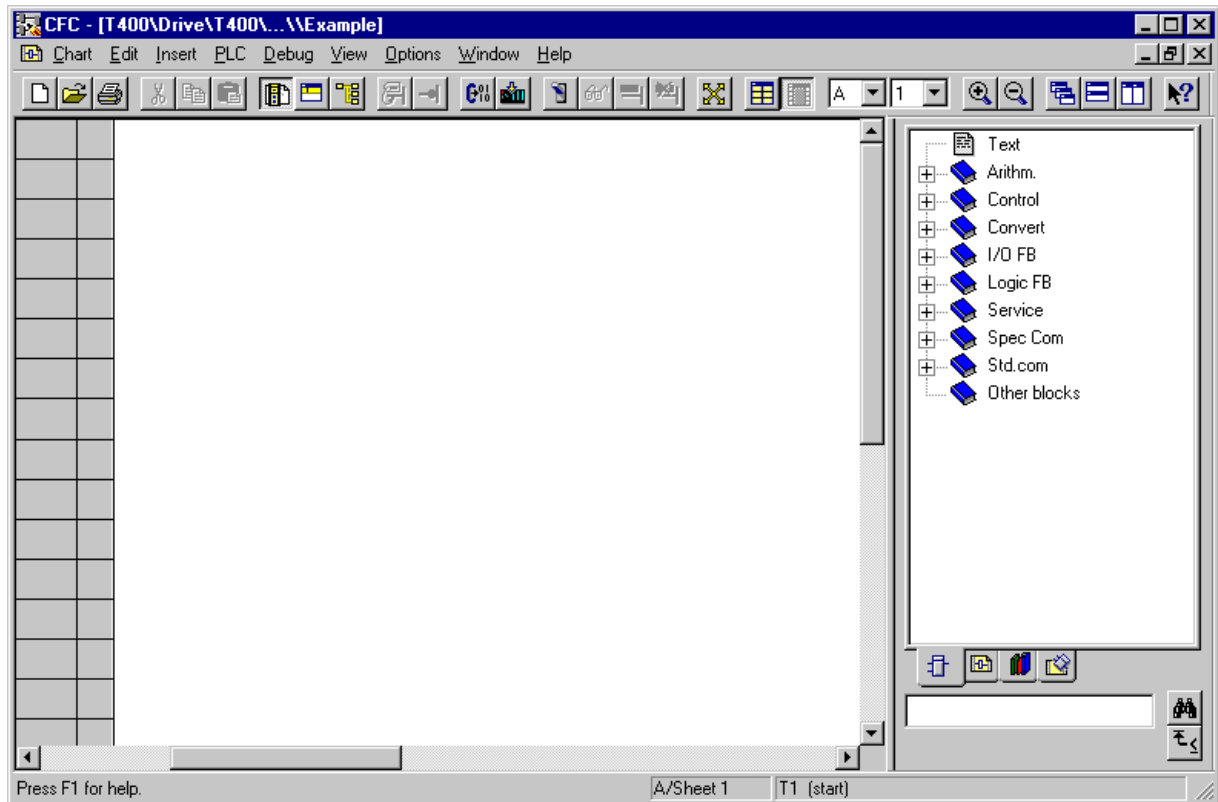
The CFC Editor is opened by double-clicking on the  symbol of the chart.

5. CFC

In CFC, the first sheet of the chart is at the left and to the right, the block catalog, in which the available function blocks are displayed.


If the catalog should not be displayed to the right, it can be enabled by clicking on the  icon using <Ctrl><K>, or in the menu **View | Catalog**.


This means that the catalog can be displayed or suppressed at any time.



We recommend that the CFC and the lefthand split display are zoomed-out to a full screen size.


A CFC chart can consist of up to 26 Chart Partitions. Always 6 pages represent a Chart Partition. New chart partitions are created using the **Insert | Chart Partition** menu items.

You can toggle between chart partitions and sheets using the following icons .


You can toggle between the overview display and sheet display using **View | Overview/Sheet View**, the  icons or by double-clicking on the background of the sheet. When changing into the sheet view, the sheet is displayed which you double-clicked on.

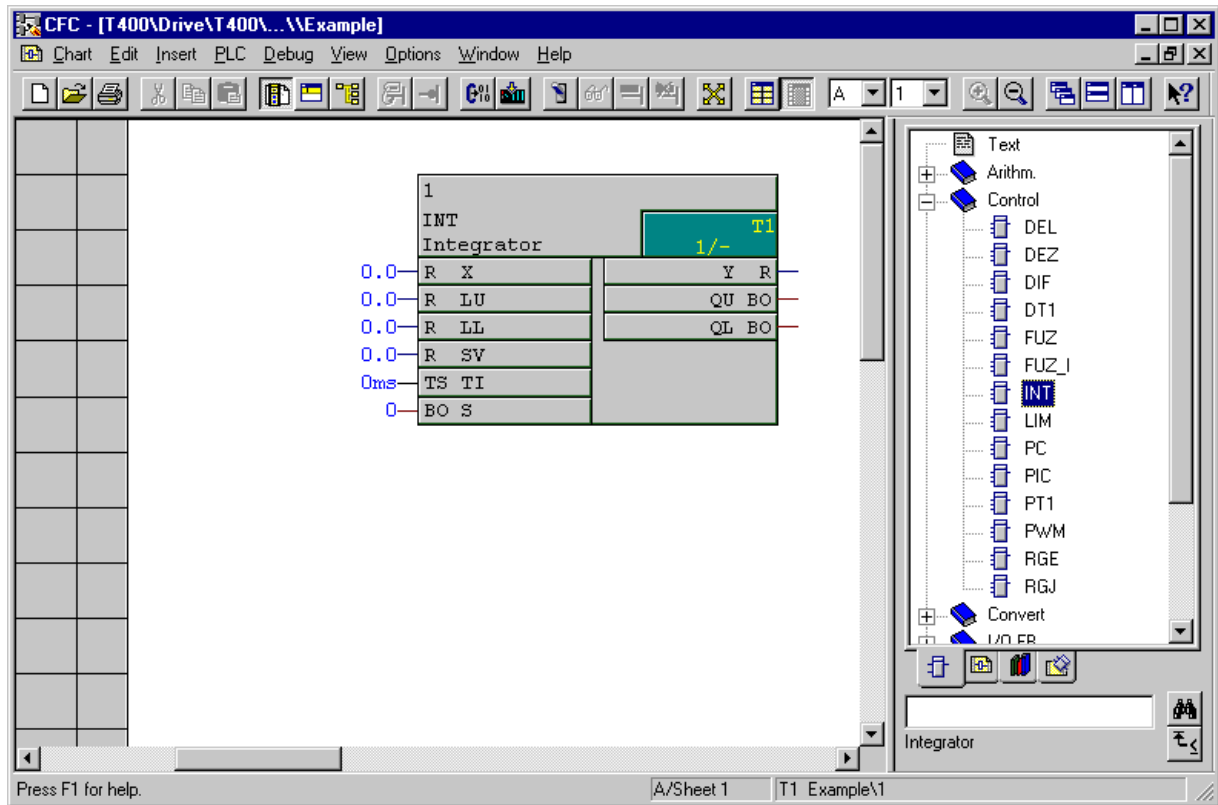
You can change the display size at any time with **View | Zoom**, the  icons or the plus and minus keys in the numerical block.

You can open an additional window of the actual chart with **Window | New Window**.

You can re-arrange all of the opened windows with **Window | Arrange** or using the  icons.

5.1. Calling-up blocks

After opening a block family, e.g. Closed-loop control, a block, e.g. INT can be dragged onto the sheet to insert how a function block. The block silhouette may not overlap the gray areas. If the function block name is known (e.g. INT), then INT can also be entered in the input field under the Catalog. By clicking on the Search symbol , select the function block and drag it onto the sheet.




In order to configure a sawtooth characteristic, an integrator is required with a fixed input value, so that it continually integrates up to this value. If the upper limit **LU** is reached, the block sets its output **QU** to 1. If this output is tracked at input **S**, then the value at input **SV** is loaded into the integrator. The integration time is entered at input **TS** in milliseconds.

An online help with detailed block description can be called-up for all blocks. To realize this, mark a function block (in the sheet or in the block catalog) and press key **F1**.

At each function block I/O, the type is located outside, and inside the block, the name of the I/O. The most important types are:

R	Real	Floating-point number
I	Integer	Integer number
BO	Bool	Digital value, 0 or 1
TS	Time	Time information as floating-point number in ms

In order to enter a value at the input, proceed as follows:

- Double-click on the input **X** , to open the input box,
- Enter, e.g. 100 and confirm.

100 is now available at input **X**.

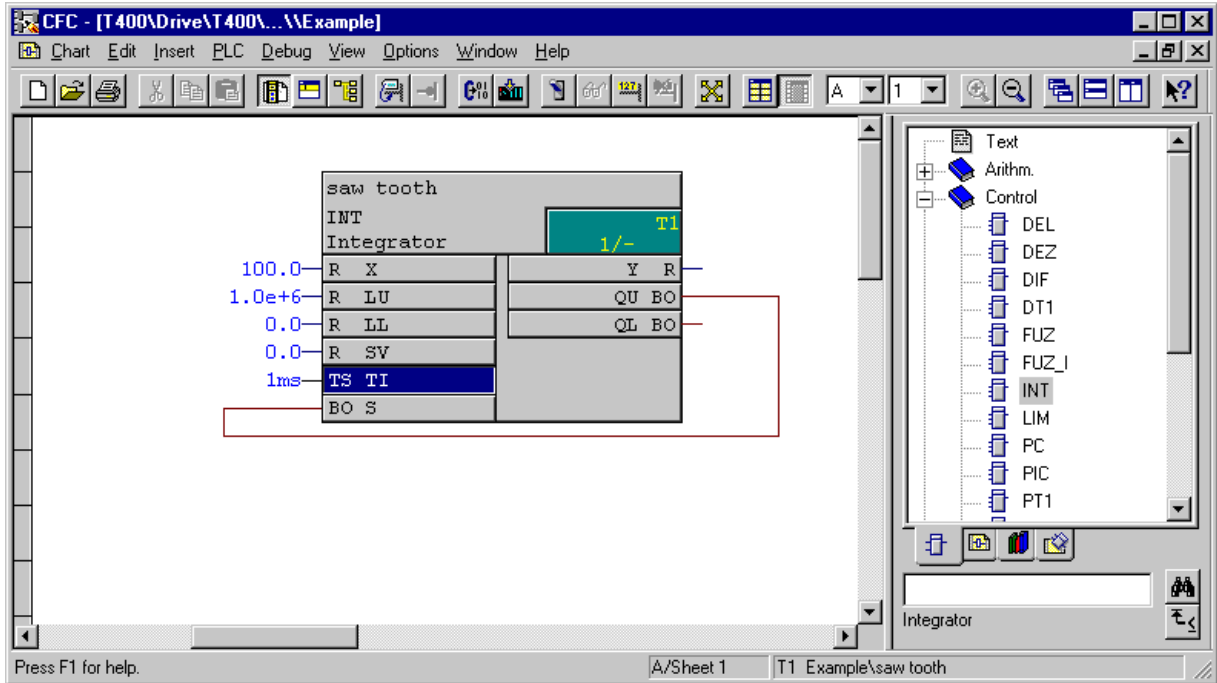
For the example, at input **LU** $1e6$ ($=10^6$) or 1000000, and at **TI** 1ms (also enter ms!).

In order to connect output **QU** to input **S**, click on output **QU** and then on input **S**.

The sawtooth is then ready. By double-clicking on the upper part of the function block, a dialog box opens in which the name of the block can be changed, e.g. from 1 into Sawtooth.

In the property dialog box of a block, the **I/O** of the block can be parameterized using the I/O tab, i.e. constant values can be written into individual I/O. In the example, at I/O **LU** and **TI**. The constants are entered in the value column of the table. The properties of the other table fields are described in the SIMADYN D documentation.

Connections can only be generated by clicking in the CFC chart.



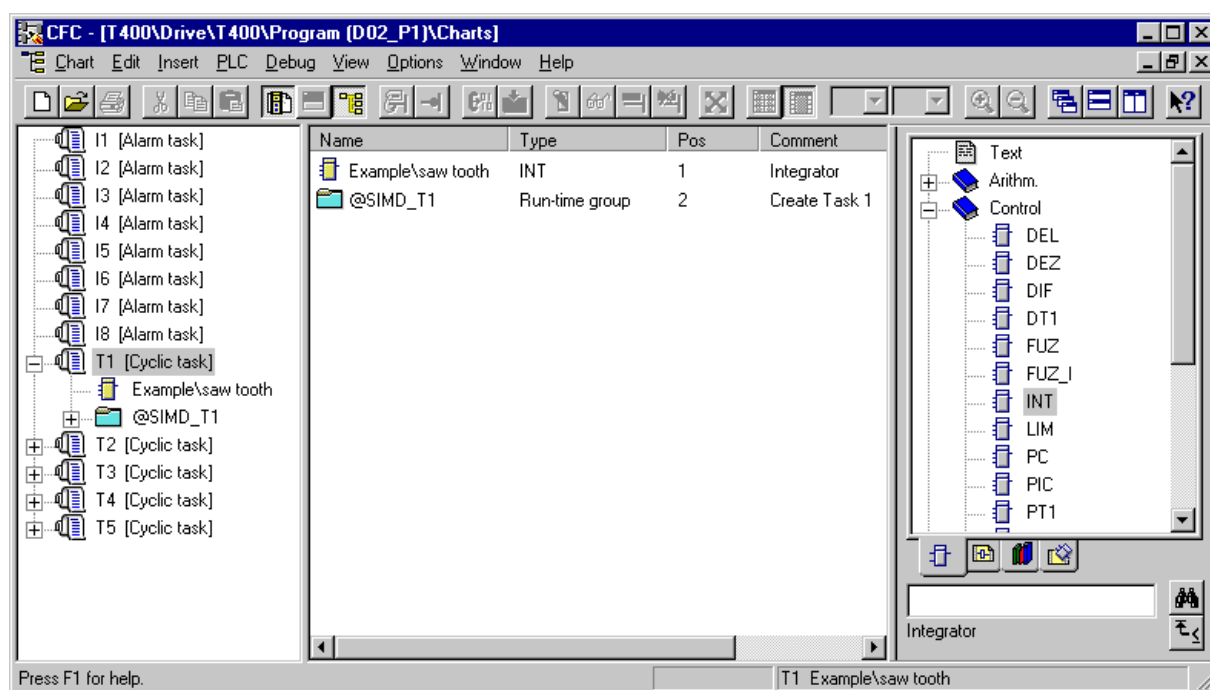
5.2. Defining the sampling time

Information on the sampling time and the run sequence within a sampling time is provided in the green field in a function block. The integrator was entered at the first position in sampling time T1. Instead of being directly set in a sampling time, function blocks can also be set in a run-time group. In this case, the second number defines in which group the function block is computed. If the block is not located in a run-time group, then, as in the display above, a line is located at the second position.

In order to shift the integrator into another sampling time, the **Run sequence** is called under **Edit**. All of the possible tasks are now listed in this dialog box. If, in the example, sampling time T1 is opened by clicking on the plus character, then the integrator is located at the first position.

Alternatively, you can also select **Go to run sequence** in the context menu of a block, and then this block is already marked in the run display.

The run display is a normal chart partition in CFC where all of the commands of the window arrangement (new window, arrange, etc.) can be used.

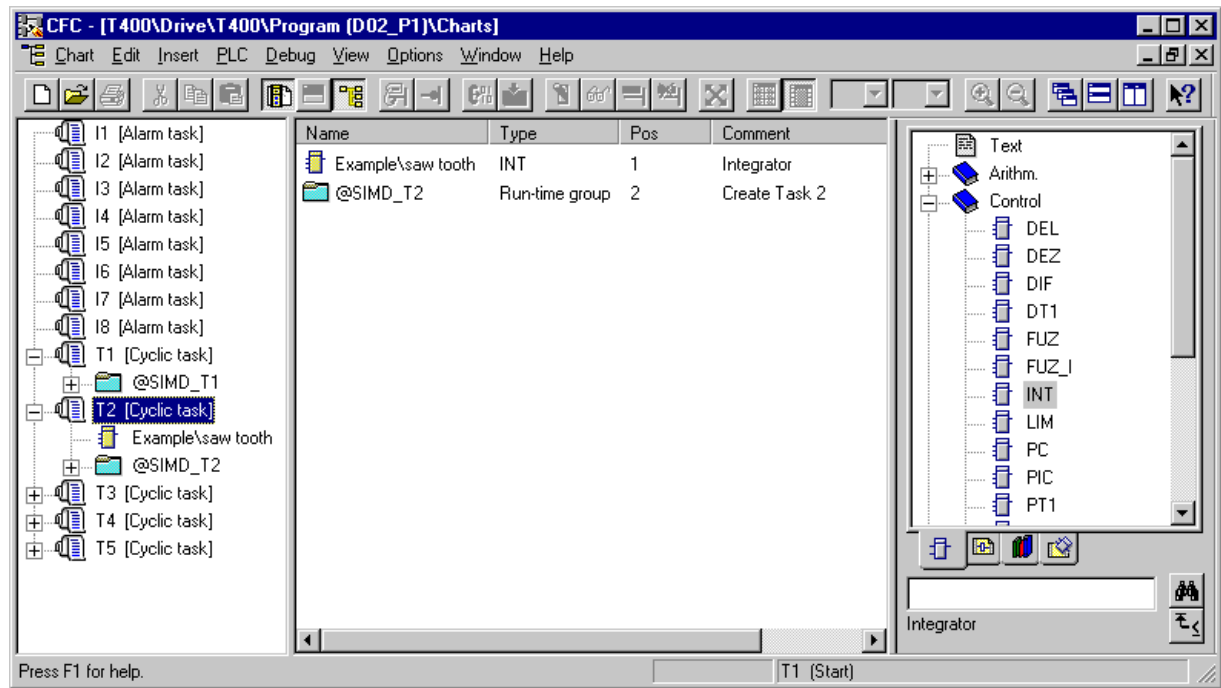


If you click on **T1 (Cyclic task)** in the lefthand split window, you'll now see a function block and a run-time group in the righthand split window (if the catalog is also included, then in the center split window). The block which is displayed is the configured integrator named sawtooth in the CFC chart example. Click on it, and drag **T2 (Cyclic task)** into the lefthand split window in order to shift it into this sampling time.

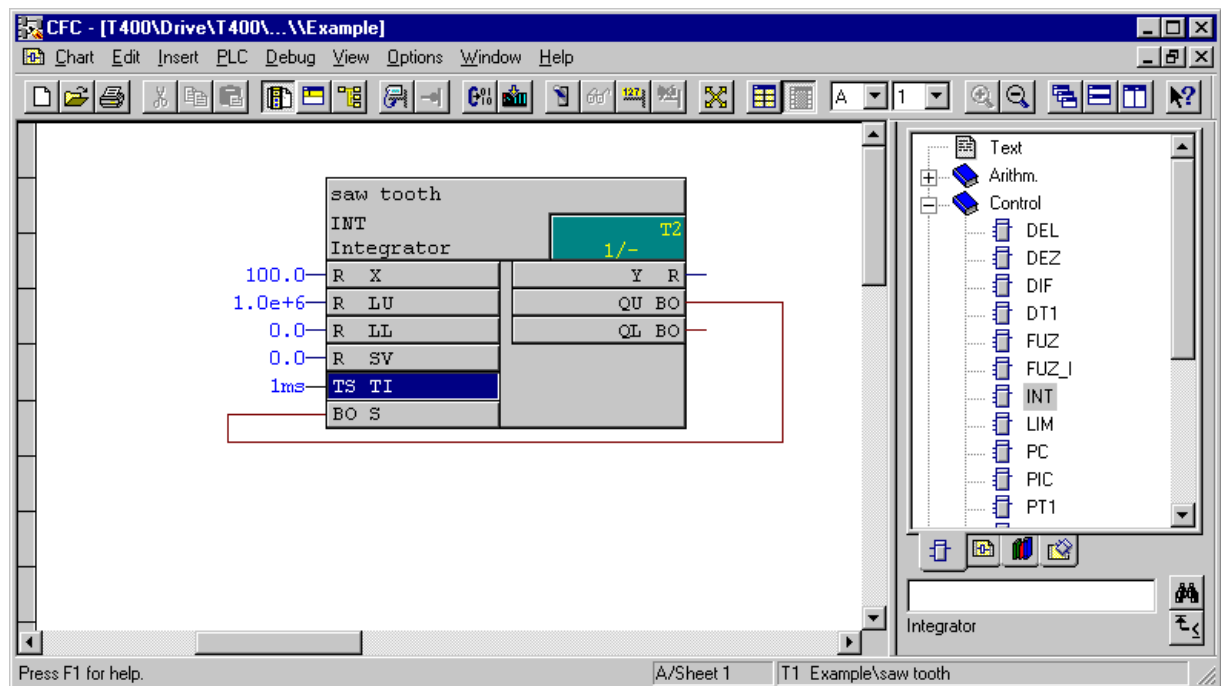
From a time perspective, several blocks can be combined in a run-time group. This means that several blocks can be shifted at once.

New run-time groups (using **Insert | Run-Time group**) can be defined in the run sequence display.

You'll find more information about the run-time groups and their use in the SIMADYN D documentation.



The integrator is now calculated as the first block in T2 and what can be seen at the block, when the run sequence display is re-closed.



In order that you don't have to shift all of the function blocks after insertion, in the run sequence, you can mark a block, and define this as **Predecessor for Installation** in the context menu. The next function block which is then inserted, is inserted directly after the integrator, i.e. in T2 at location 2. The next one would be at location 3, etc. Before you configure a function sequence from several blocks, the insertion time point should first be defined, otherwise the function blocks must first be arranged in the run sequence.


5.3. Compilation

Before you compile a chart for the first time after installation, the temporary directory, specified in the menu item **Options | Customize | Compile** should be checked, and if required set-up, or the entry modified.

If address book generation is activated in this menu, the function block names may not start with a number, i.e. you must first re-name all of the newly inserted blocks, as numbers will be automatically assigned for the newly inserted function blocks. If you do not require an address book, this restriction does not apply. The address book is only required for a coupling to a visualization system.

The address book generation should be disabled when configuring the T400.

When compiling, information about the calculated CPU utilization, hardware accesses and communications settings are saved in a CPU map listing, which can be useful when troubleshooting.

To compile, click on the following symbol:  or select **Chart | Compile Charts as Program**.

OK is sufficient in the window which then opens.

The generated program is now compiled, and is linked with the SIMADYN D operating system to create a CPU program. If faults/errors occur, these are displayed.

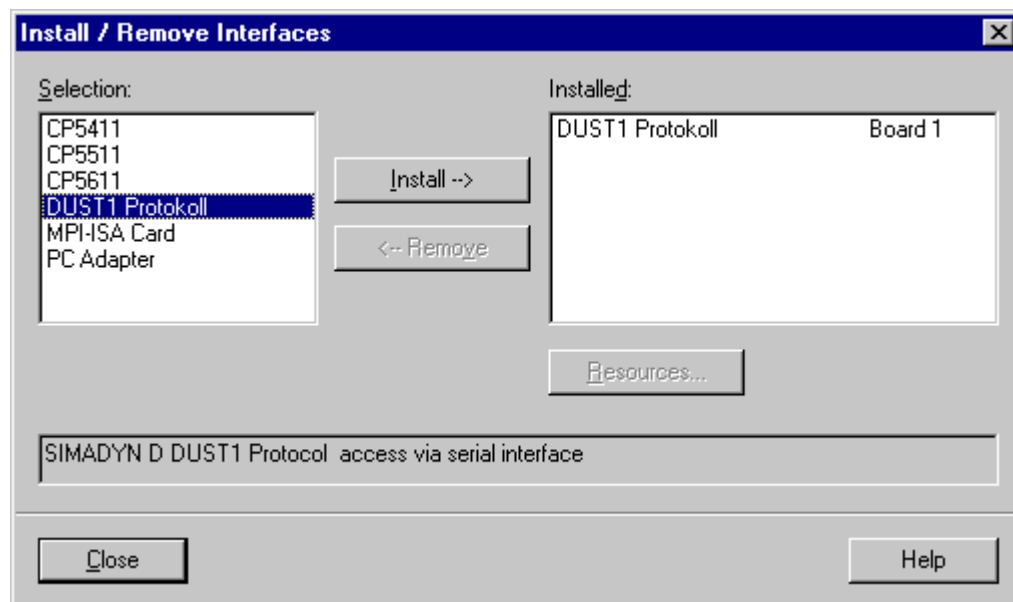
5.4. Loading

Loading is also known as download.

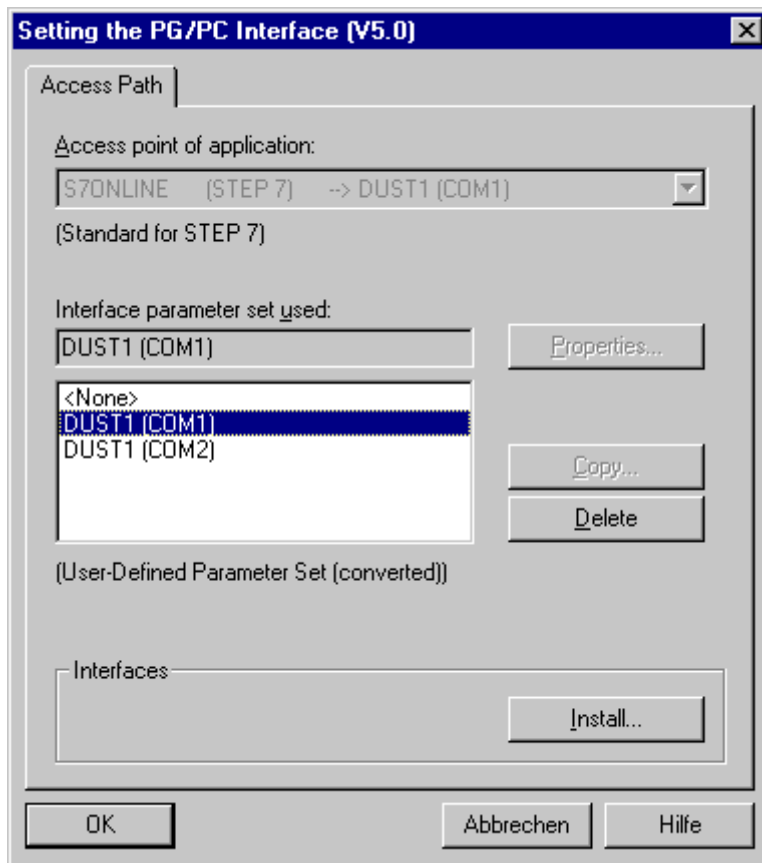
Selecting the online interface

For the T400, data can only be downloaded via a serial interface (RS232), as there is no plug-in program memory on this module. The interface to be used for online download, is set using the **PG/PC interface** program, in the control panel. Alternatively, the setting can also be called-up in the SIMATIC Manager under **Options**.

Install... must first



COM1 or COM2 can then be selected after this window closes:




The coupling between T400 and the configuring PC is realized via a serial connection. The following cable should be used.

PC			T400	
9-pin Sub D socket			Terminal	
Pin				
2	↔		68	
3	↔		67	
5	↔		69	

The cable can also be ordered under the SC400 designation, Order No. 6DD1684-0GF0.

In this case, switch S1/8 (switch 8 from S1) on the T400 must be in the OFF position (default).

Downloading:

Open the load window in the CFC, by clicking on the symbol  or using the menu item **PLC download**.

In the example, in the load window, **System and User Program**, **Online** and **Initial load** must be selected.

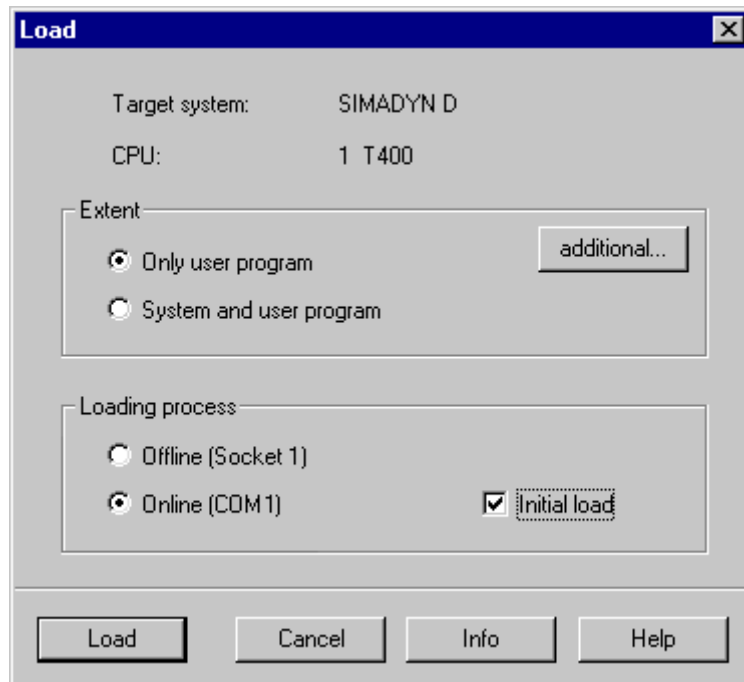
Notes

The system (the SIMADYN D operating system) should always be loaded, if it is not certain that the T400 was last loaded with the same SIMADYN D version. Otherwise, the data transfer time can be significantly reduced if only the user program is transferred.

This means that the system should be transferred when the configuring software version is changed and when new T400 modules are used.

Initialization (initial load) must always be selected, if the hardware configuration has changed, i.e. changes were made in HW Config, the configured software was just de-archived or a new project should be loaded in the T400.

T400 can only be loaded online, as the program memory is soldered-in.



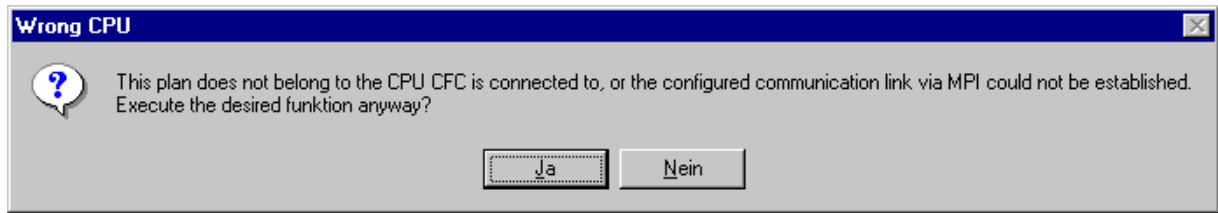
If the T400 is inserted in the electronics box, the electronics of the basic drive has power and the connecting cable between T400 and the configuring PC is inserted, loading can be start by clicking on the load button.

This operation takes several minutes as the system and the user program are loaded. The time is shown above the button Info, if the T400 is connected to the configuring device.

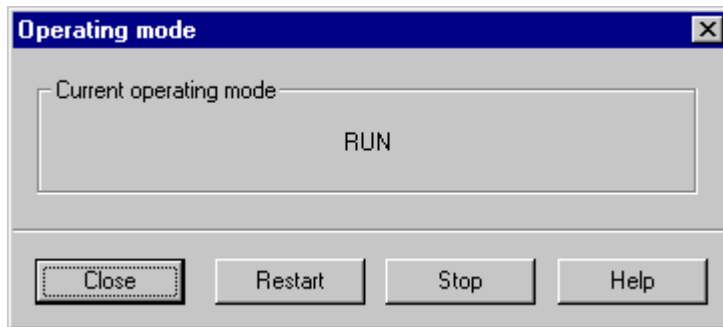
The previous program still continues to run on the T400. The loaded program which has just been loaded is only activated after a restart.

The T400 must then be reset. For safety reasons, it is checked whether the hardware configuration (more precisely, its ID) of the T400 and the configured application software, loaded in the CFC, are identical.

However, if this was just recently loaded with **Initial load**, then this is not the case. Thus, the following window is displayed which you must acknowledge with **Yes**:



The **Operating mode** window is now displayed:




A reset is initiated on the T400 using the **Restart** button and the new program is then executed. If the T400 is already registered in the basic drive, the basic drive may signal a process data error. This must then be acknowledged.

If changes on technology parameters have been made, the basic drive must be resetted too, to activate the changes.

If the restart is executed, the new program is running, and the window can now be closed.

5.5. Test mode

The test mode is often known as the online mode or CFC online.

The test mode is started using the **Test, Test mode** menu item or via the  symbol. In the test mode, the actual values from the CPU module are displayed in the software.

Changes are executed and saved, both in the CFC chart as well as in the CPU. Thus, when the test mode is called, data consistency is checked between the CFC chart and the CPU program.

If these differ, then an appropriate message is output, or access is denied.

Online changes, where the data is not consistent, can result in serious erroneous functions!

Data consistency can be checked using **PLC, Compare** and can be re-established by **Loading** again (it may first be necessary to re-compile first).

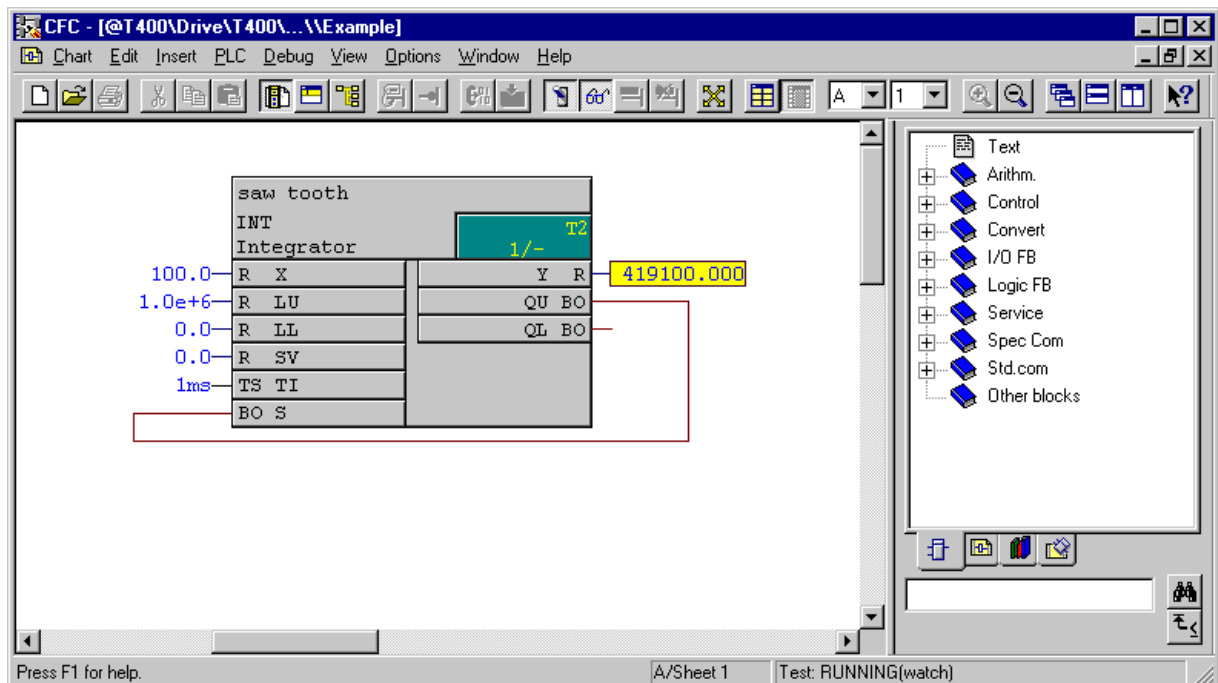
When changes are made with the CFC in the test mode, data consistency is kept; if changes are made in the standard (offline) mode of the CFC, then these changes are lost.

If changes are made in the CPU online with another program, or using another computer, or changed by accessing the technology parameters, the data are also not consistent. When calling-up the test mode, there is then the update menu item. Online changes can then be updated in the CFC. After the update the CFC has to be set to offline and the display must be updated (press F5) to show the changes in the CFC.

After a download, the CFC program may not be changed to allow a update.

Viewing and changing values

After starting the test mode, the CFC first looks the same. After an input or output or the complete block has been marked (by simply clicking on it with the mouse), **Add to Watch List** menu item can be selected in the context menu (righthand mouse key). The actual value is now displayed:



If the CFC is online, I/O can be registered and de-registered extremely quickly by clicking on them with the CTRL key pressed.

I/O can also be registered and de-registered using the following icons:  

Changing values

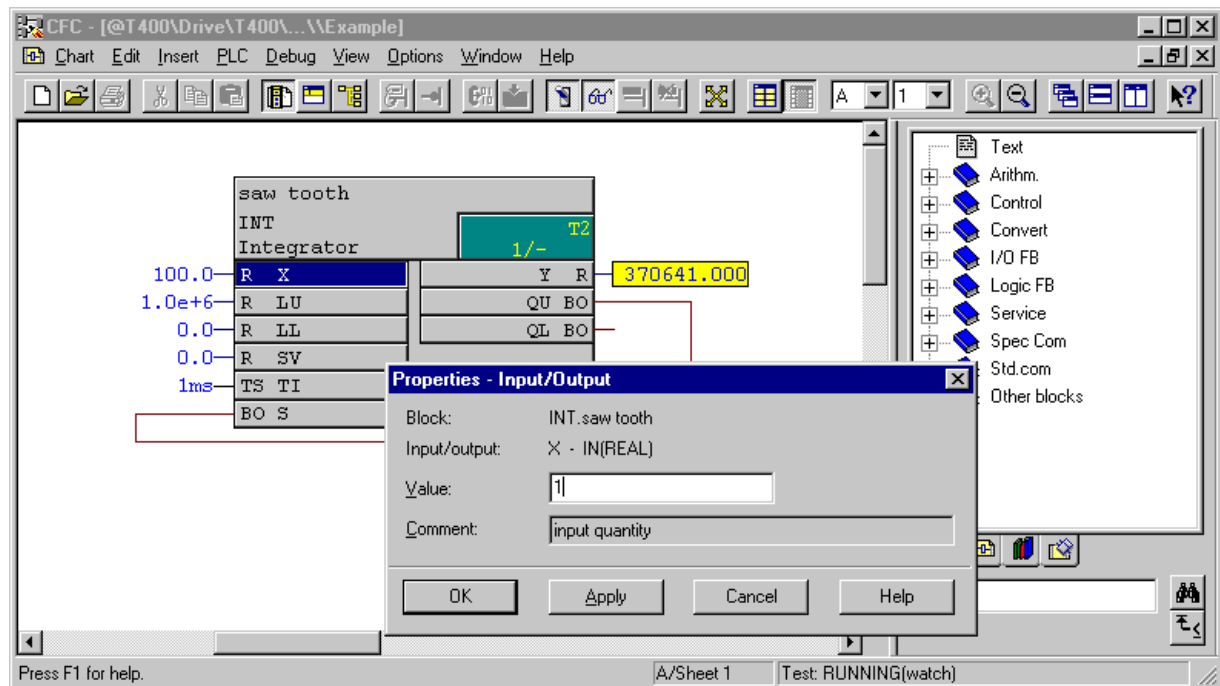
Changes made in the test mode become immediately effective in the CFC configured application software and are saved in the T400.

Values are changed just the same way as in the offline mode, by double-clicking on the value to be changed.

The new value can then be entered in the input field which opens.

For example, the input value of the integrator can be changed by double-clicking on input **X**, e.g. on 1. The integrator immediately becomes “slower”.

Block outputs can only be monitored, but not overwritten.




Inserting blocks

Most of the function blocks can be inserted in the test mode. (Function blocks with hardware access cannot be inserted online.) In this case, the block is dragged onto the sheet from the function block catalog, just the same as when configuring offline.

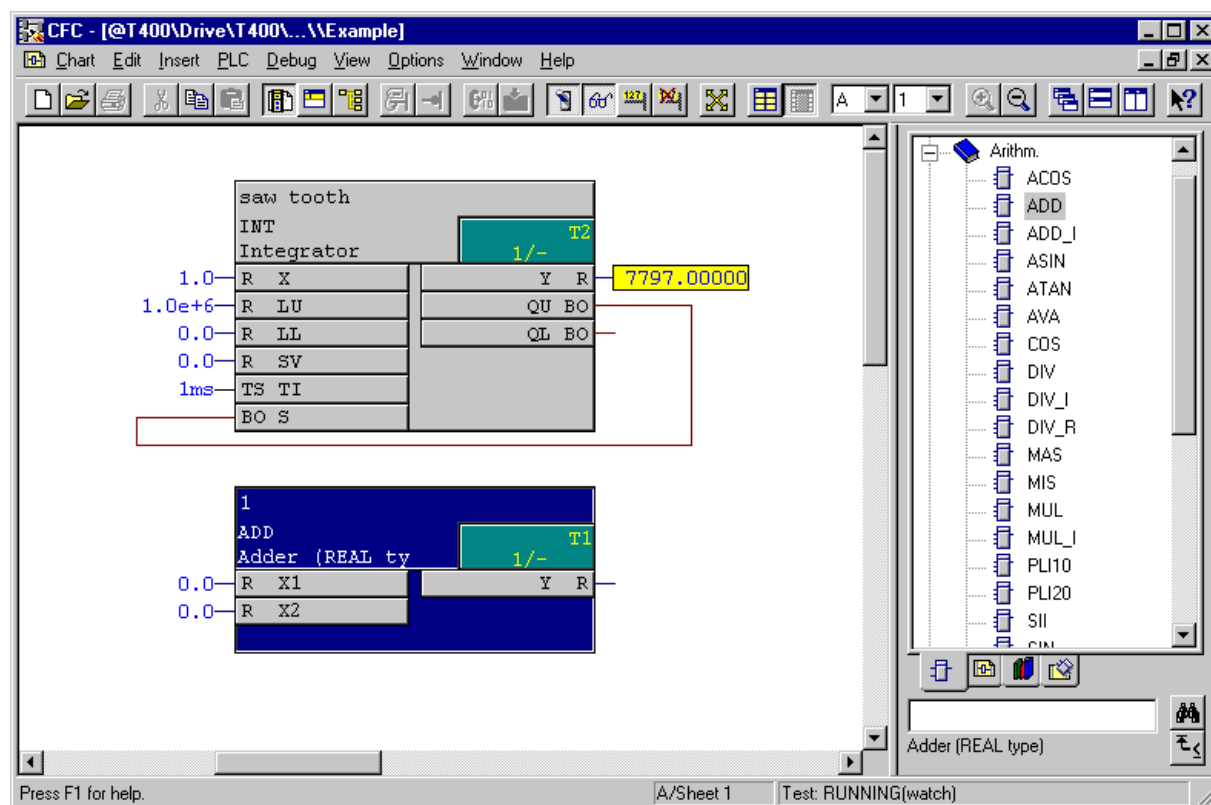
Online, it is especially important to first select the **Predecessor for installation**, as the run sequence cannot be changed online.

In order to insert an adder function (function block name: ADD) in T2 after the integrator, this must first be defined as **Predecessor for installation**.

The ADD function block can now be selected in the function block family, arithmetic operations in the function block catalog.

As the function block name is known, you can also enter ADD in the input field under the catalog and then click on the Search symbol , to quickly select the function block which is then dragged onto the sheet.

Function block names cannot be changed in the test mode.



Inserting and changing interconnections

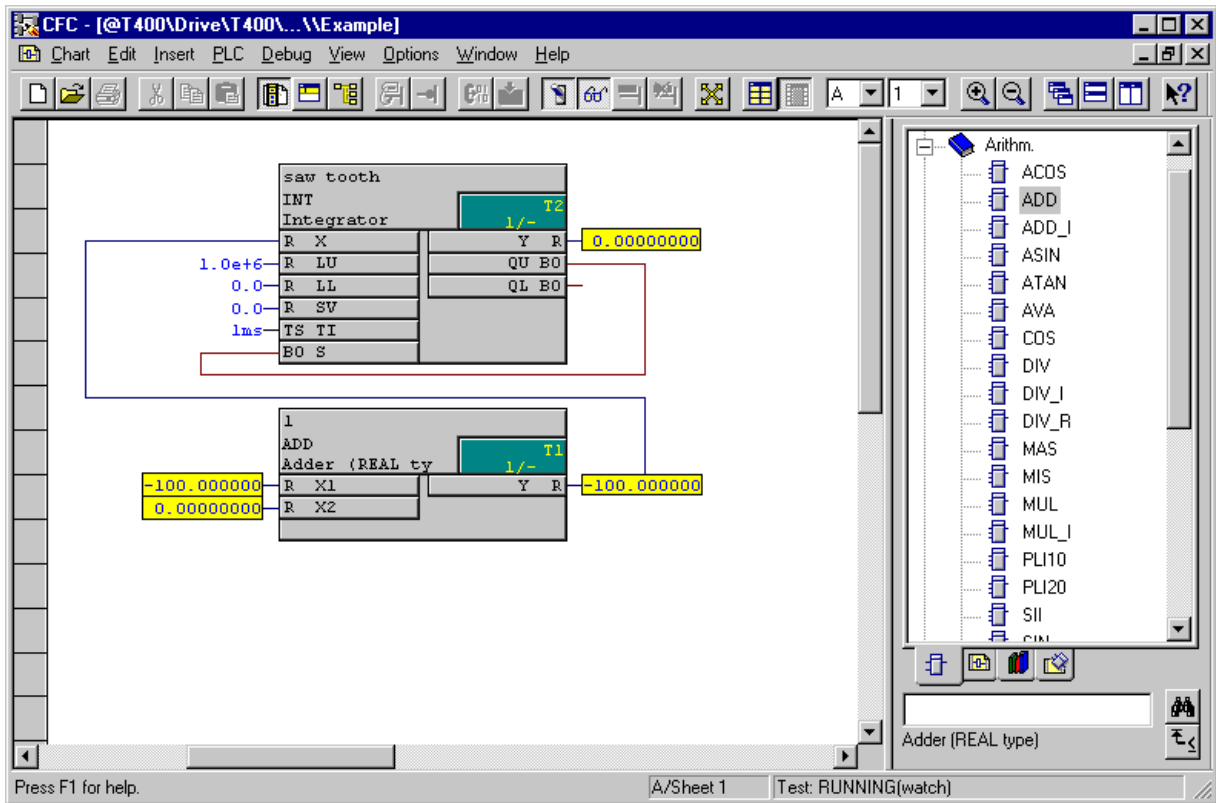
Connections can be generated, deleted and changed online just the same as offline.

Simply click on output **Y** of adder 1 and then on input **X** of the sawtooth integrator.

The new connection has a different color than the previous one, as a different color is used for each numerical format. (In this case, firstly boolean and then floating-point/real)

The integrator now remains stationary, as a zero is present at its input **X**.

If you write a value which is not equal to zero at one of the inputs of the adders (by double-clicking on **X1** of the adder, the input dialog box opens), and it starts again. If the value is negative, then the integrator output value decreases until it reaches the lower limit (**LL**).

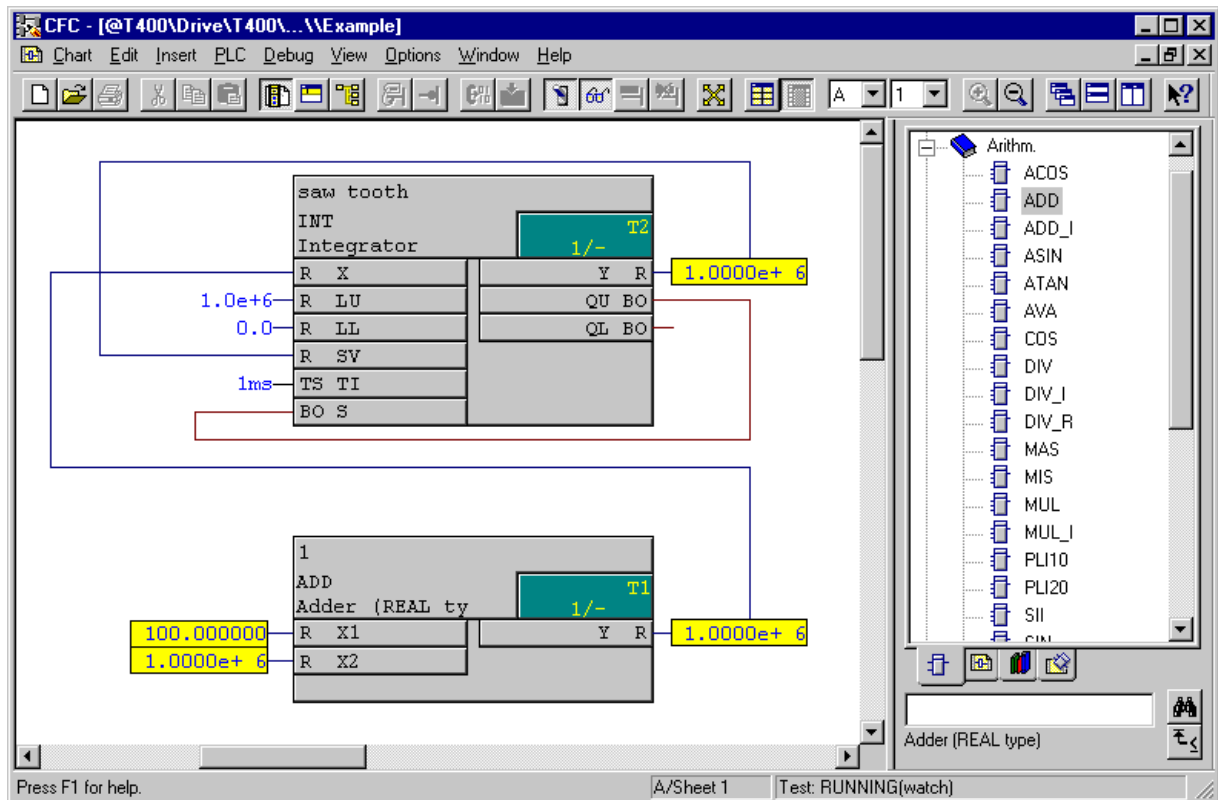


Write 100 at input **X1** of adder 1

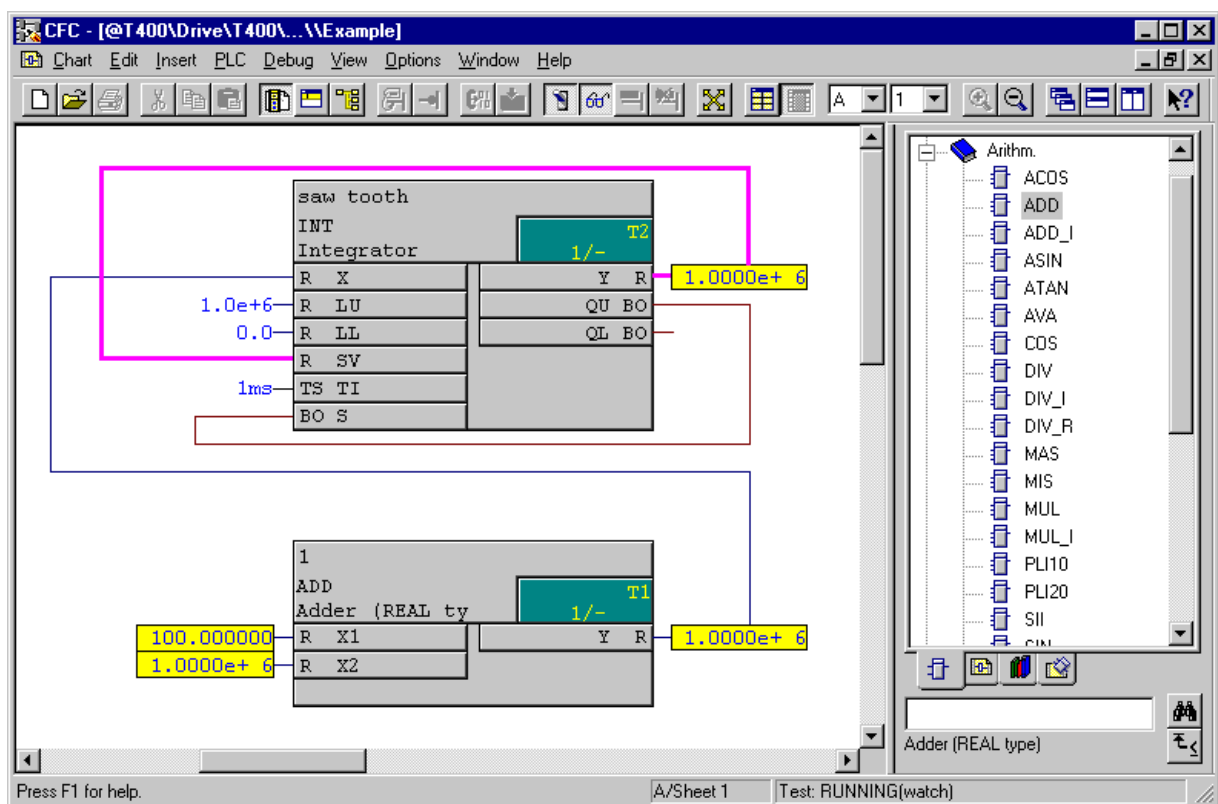
Now connect output **Y** of the sawtooth integrator to input **X2** of the adder. The integrator quickly reaches the maximum value so that the increase can no longer be actually monitored.

To change the connection, click on input **X2** of adder 1 and drag the mouse pointer to input **SV** of the sawtooth integrator.

The integrator is now fixed at a value of 10^6 .



If you click on a connecting line, this flashes (also offline):



Deleting connections or blocks

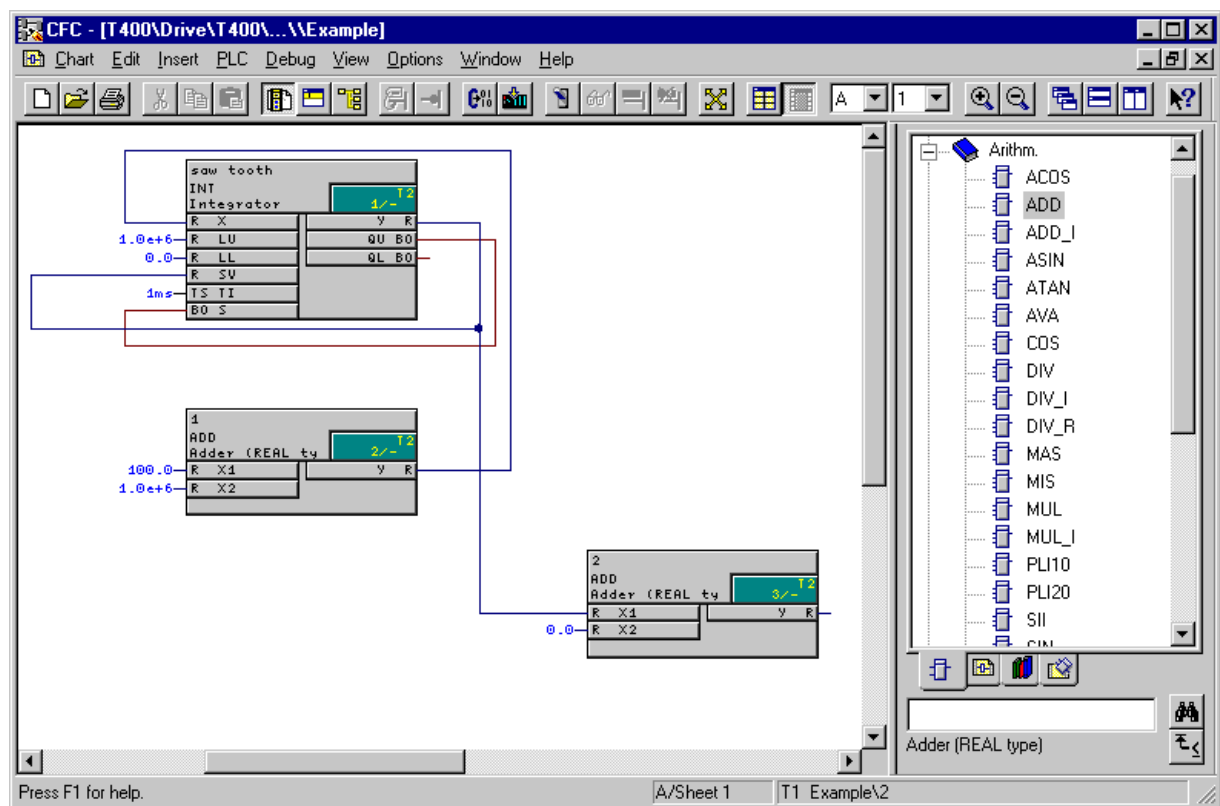
The connection is deleted if you mark an input or output, and select **Remove connection** in the context menu.

If a function block is marked, it can be deleted by selecting **Remove** in the context menu (righthand mouse key), or by pressing the Del key.

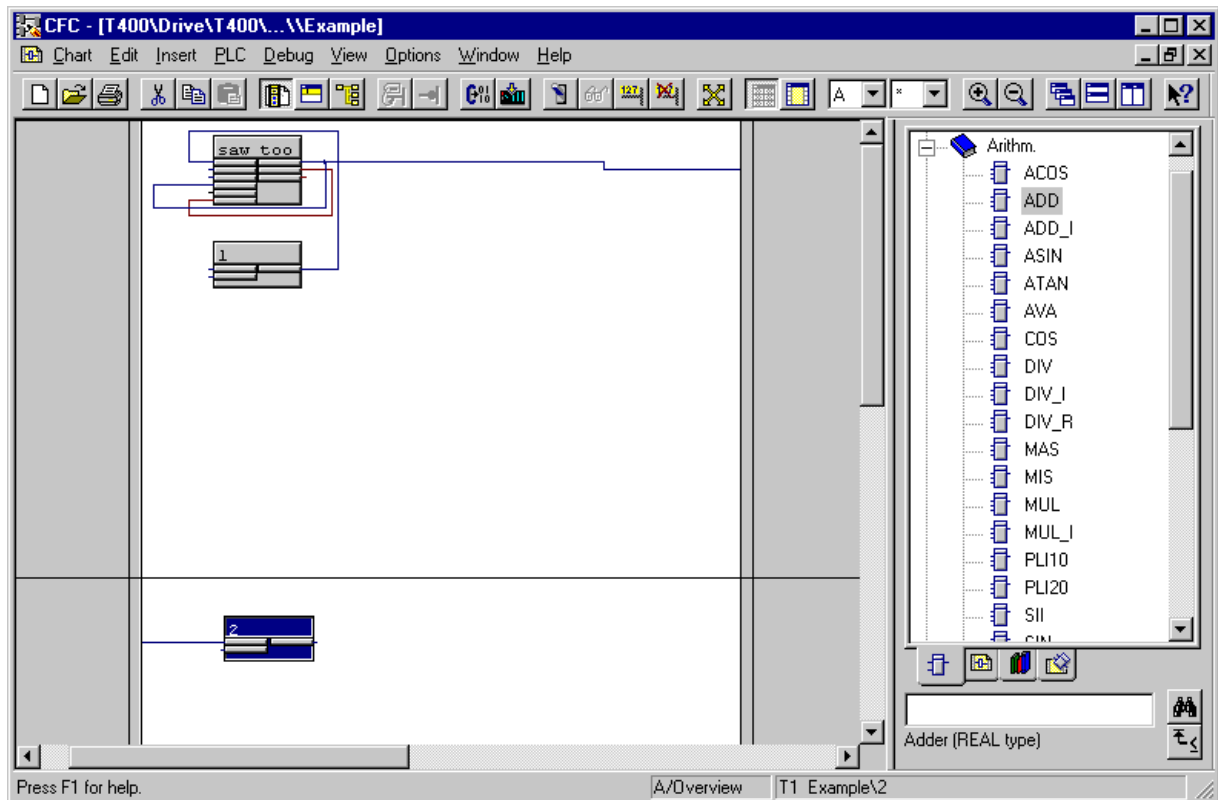
Exit the test mode by again selecting the **Test | Test mode** menu item or by clicking on the  symbol.

5.6. Sheet bar/margin

Enter another block, e.g. another adder and then establish a connection between the blocks.



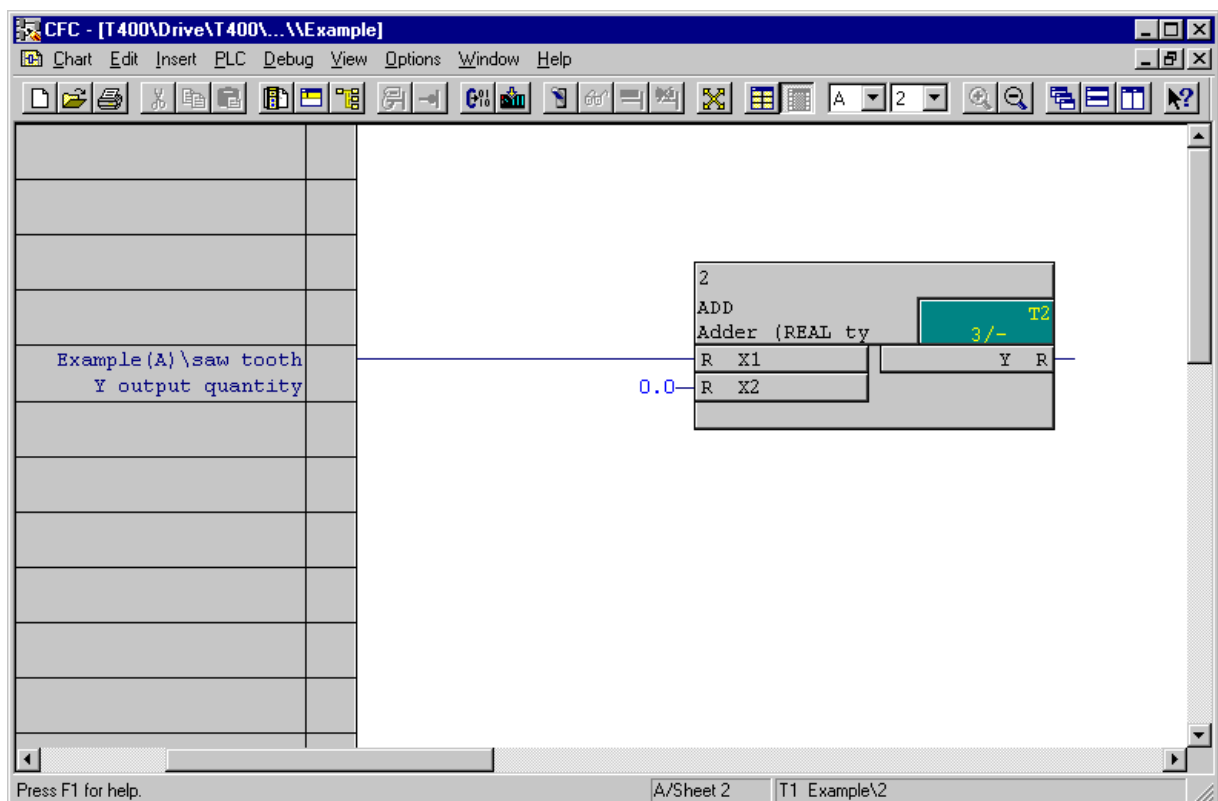
Change into the overview display (e.g. by double-clicking on the background). Drag one of the function blocks onto Page 2.



The connection can now no longer be directly displayed, as this sheet is an overview. Thus, a reference is generated in the margin.

Change back into the page view, Page 2 by double-clicking on the background on Page 2 .

The catalog is now no longer displayed (click on )



The reference in the margin indicates that the signal in the chart 'Example' (chart partition A), comes from the function block with the name `Sawtooth` and its output **Y**. The comment `Output quantity` is entered at this connection.

By double-clicking on the margin entry, you can easily trace this signal. You will see the **Jump back** entry in the context menus. This allows you to jump back.

This now completes the description of the CFC and the example.

In order to read-in or output inputs/outputs of the T400, the input and output blocks must be called. This is shown in the following Section.

In order to be able to address the basic drive (MASTERDRIVE or DC-MASTER) from the T400, communications must be established between the basic drive and T400. This is also shown in the following Section.

6. Configuring instructions

in this Section the steps, which are necessary to handle certain tasks, are discussed, e.g. communications via PROFIBUS, or reading-in data from an absolute value encoder.

6.1. General information

principle function

The function block reads-in values at the hardware terminals, and these values are available at its outputs for other FBs.

Values, which are connected at a function block input, are output by the FB at terminals via the hardware.

Hardware address

The connection to the symbolic address, entered in HW Config is established via connection **AD**.

In the context menu of the **AD** connection, all of the inputs/outputs are available which are suitable for this block type (and were named in HW Config) under the point **Insert Connection to address**

The selection field remains empty, if there is no the suitable connection or if it was named in HW Config.

Direct mode

Input **DM** is parameterized to define whether the converted value is read-in in the normal- or system mode.

For **DM** = 1, data is read-in/written in the normal mode, just when if the FB is calculated.

For **DM** = 0 (no direct mode), data is read-in/written in the system mode. The system mode is always calculated at the beginning and end of the sampling time (process image).

Sampling time

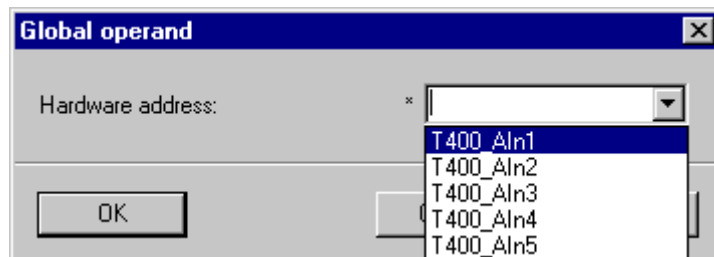
The sampling time, in which the input/output- and communication blocks are called, defines how often the hardware is read (or written). The block should therefore be called in the sampling time, in which the function blocks, which further process the value which was read-in (or generate the value to be output) are calculated.

6.2. Analog inputs

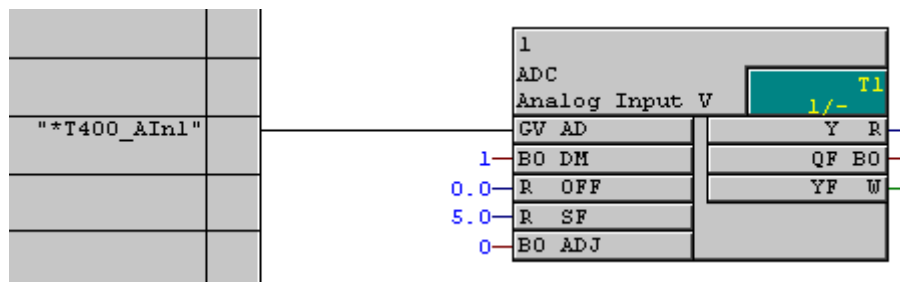
The ADC function block is used to read-in signals at the analog input (terminals 97 to 99). The block outputs the value connected to the hardware at its block output. The function block belongs to the family of **On/Off** blocks in the catalog.

In order to establish the assignment to the symbolic hardware address, assigned in HW Config (in the example ***T400_AIn1**), call-up the context menu of the I/O **AD** and select **Insert connection to address**.

In the following window, open the selection to display the possibilities for this particular connection, and select e.g. **T400_AIn1**. (the star symbol is located in front of the selection field)



In the CFC, you'll now see a cross reference with the symbolic hardware name ***T400_AIn1**.



As ***T400_AIn1** was assigned in HW Config for terminals 97/98/99, the analog value available at this terminal, is read-in by the block. The output value at Y is calculated according to the following formula:

$$Y = \frac{V}{5V} \cdot SF - OFF$$

Whereby V is the measured voltage in volt.

For the standard assignment of the inputs **SF** (scaling factor) and **OFF** (offset), for 1.0 V, 1.0 is output, i.e. the output corresponds to the voltage connected in volt.

The analog input can be self-calibrated using connection ADJ.

Output **QF** is set to 1, if there is a conversion error.

The cause is displayed at **YF**.

More detailed information is provided in the block documentation or in the online help.

An additional **ADC** block must be configured (with another address at **AD**) to read-in one other analog input.

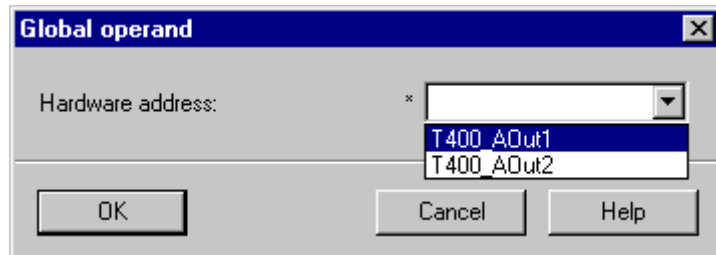
Function block **ADC** cannot be used with T400.

6.3. Analog outputs

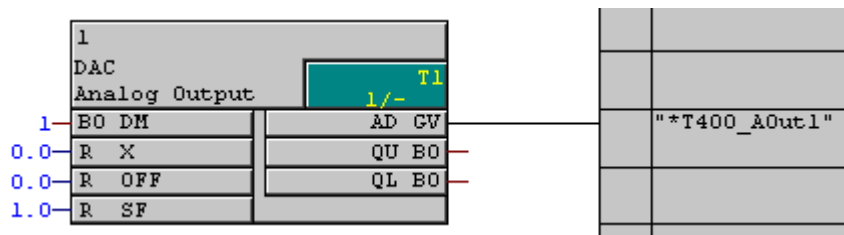
Function block DAC is used to output signals at the analog output (terminals 97 to 99). The block outputs the value at its block output as voltage via the hardware.

In order to establish the assignment to the symbolic hardware address assigned in HW Config (in the example **T400_AOut1**), call-up the context menu of the I/O **AD** and select **Insert connection to address**.

In the following window, open the selection to display the possibilities for this particular connection, and e.g. select **T400_AOut1**. (the star symbol is located in front of the selection field)



In the CFC, you'll now see a cross reference with the symbolic hardware name ***T400_AOut1**.



As ***T400_AOut1** was assigned terminals 97/99 in HW Config, the analog value, available at the block is output at these terminals. The output voltage is calculated using the following formula:

$$V_{AD} = \frac{X + OFF}{SF} \cdot 5V$$

Whereby V_{AD} is the output voltage in volt.

For the standard assignment of inputs **SF** (scaling factor) and **OFF** (offset), a value of 1.0 V is output for a value of 1.0, i.e. the output corresponds to the value in volt.

Output **QF** is set to 1 if there is a conversion error.

The cause is displayed at **YF**.

More detailed information on this is provided in the block documentation or in the online help.

In order to use the second analog output, an additional DAC block must be configured (with another address at **AD**)

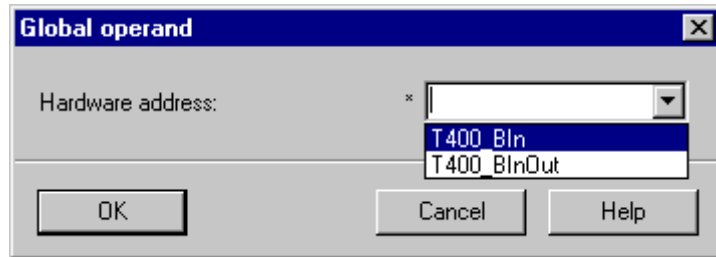
6.4. Digital inputs

The digital inputs (terminals 53 to 60) are read-in using block BII8. In this case, 24V corresponds to a signal of **1** and 0V, a signal of **0**.

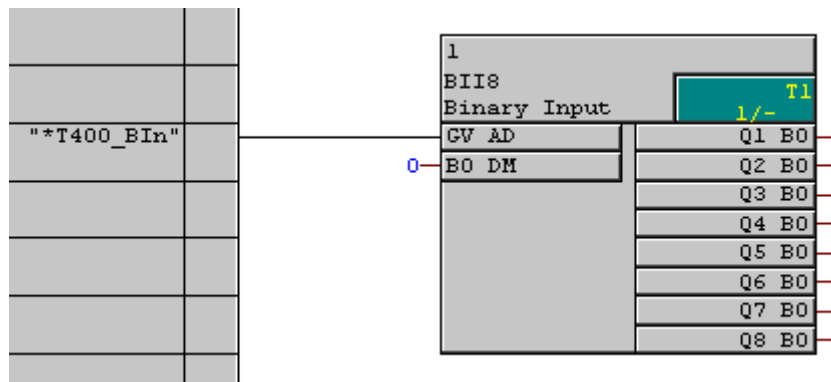
The function block belongs to the **On/Off** family in the block catalog.

In order to establish the assignment to the symbolic hardware address, assigned in HW Config (in the example ***T400_BIn**), call-up the context menu of the I/O **AD** and select **Insert connection to address**.

In the following window, open the selection to display the possibilities for this particular connection, and e.g. select **T400_BIn**. (the star symbol is located in front of the selection field)



In the CFC, you'll now see a cross reference with the symbolic hardware name ***T400_BIn**.



The voltage available at terminals 53 to 60 is output at function block outputs **Q1** to **Q8**.

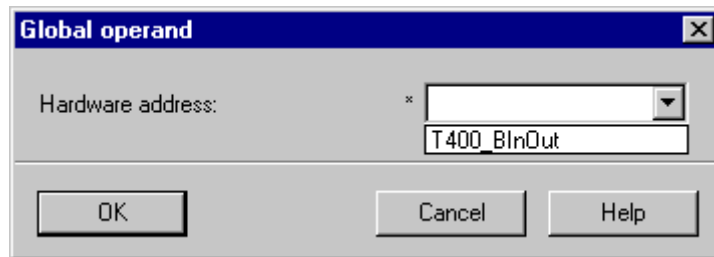
6.5. Digital inputs/outputs

The digital inputs and outputs (terminals 46 to 49 and 51,52,84 and 65) are, for the T400, read-in and output using the **BIQT** block. 24V corresponds to a signal of **1** and 0V, to a signal of **0**.

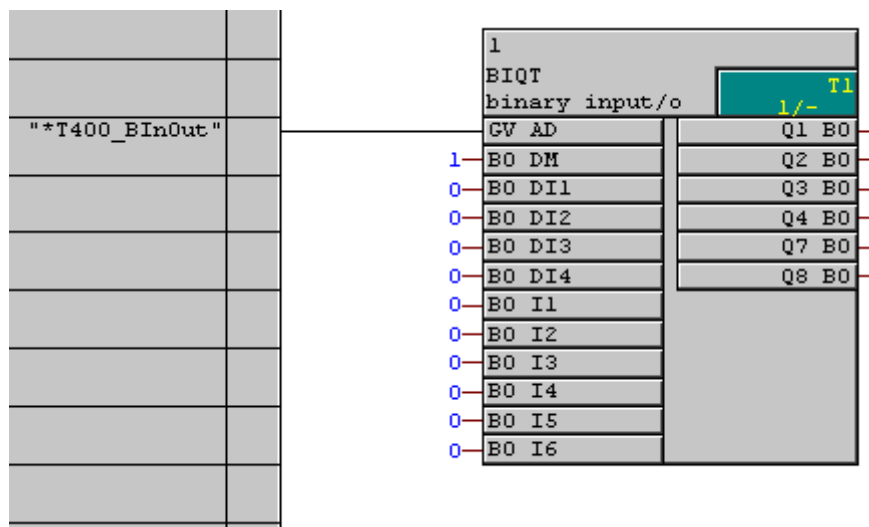
The function block belongs to the **On/Off** block family in the catalog.

In order to establish the assignment to the symbolic hardware address, assigned in HW Config (in the example ***T400_BInOut**), call-up the context menu of the I/O **AD** and select **Insert connection to address**.

In the following window, open the selection to display the possibilities for this particular connection, and e.g. select **T400_BInOut**. (the star symbol is located in front of the selection field)



In the CFC, you'll now see a cross reference with the symbolic hardware name *** T400_BInOut**.



Connections **DI1** to **DI4** are used to define whether the bi-directional input/output terminals should be used as input or output terminals. This can be defined individually for each of the 4 terminals. For **DIx=0**, the terminal is defined as input and for **DIx=1**, as output.

The values, which are available at function block inputs **I_x**, are output from the block via the hardware. The signals available at the input terminals of the T400 are output at function block outputs **Q_x**.

Information:

Connections **DM** and **DIx** are initialization inputs, which means that although they can be changed online, the change is only effective after the T400 restarts. **BIQT** cannot be entered online.

Please observe that the function of **I1** to **I4** and **Q1** to **Q4** depends on the setting of function block inputs **DI1** to **DI4**.

Value at DIx	Input Ix	Output Qx
0	active	--- *)
1	---	active

*) The inverted value of the signal at Ix is output.

I5 and **I6** are always active, also **Q7** and **Q8**.

Terminal 84 (= **I5**) and terminal 65 (= **I6**) are also used as coarse pulse input.

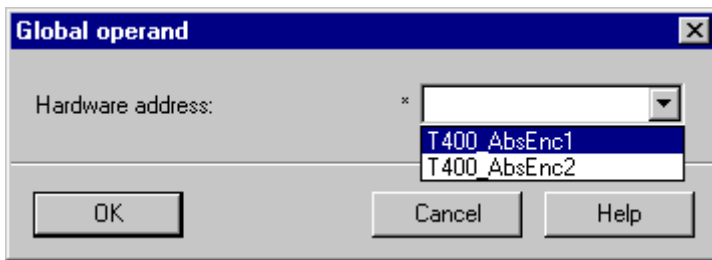
Connect.	Function
DI1	Mode of terminal 46, 0=output, 1=input
DI2	Mode of terminal 47, 0=output, 1=input
DI3	Mode of terminal 48, 0=output, 1=input
DI4	Mode of terminal 49, 0=output, 1=input
I1	Value which should be output at terminal 46, (only if DI=0)
I2	Value which should be output at terminal 47, (only if DI 2=0)
I3	Value which should be output at terminal 48, (only if DI 3=0)
I4	Value which should be output at terminal 49, (only if DI 4=0)
I5	Value which should be output at terminal 51
I6	Value which should be output at terminal 52
Q1	Value which is available at terminal 46, (only if DI 1=1, otherwise, inverse to I1)
Q2	Value which is available at terminal 47, (only if DI 2=1, otherwise, inverse to I2)
Q3	Value which is available at terminal 48, (only if DI 3=1, otherwise, inverse to I3)
Q4	Value which is available at terminal 49, (only if DI 4=1, otherwise, inverse to I4)
Q7	Value which is available at terminal 84
Q8	Value which is available at terminal 65

6.6. Absolute value encoders

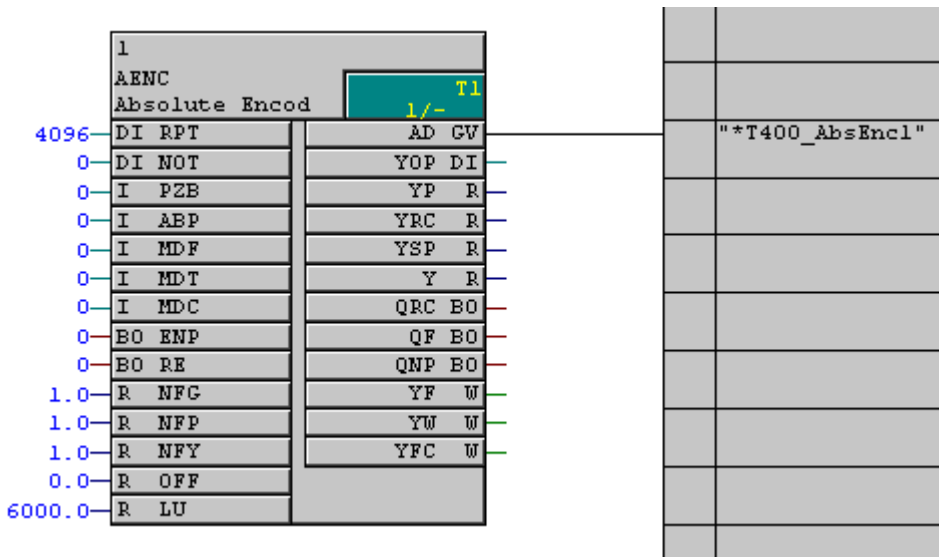
The AENC function block is used to evaluate absolute value encoder inputs with SSI- or EnDat interface. The block outputs the absolute position and a calculated speed at its block output. The function block belongs to the **On/Off** block family in the catalog.

In order to establish the assignment to the symbolic hardware address, assigned in HW Config (in the example ***T400_AbsEnc1**), call-up the context menu of the I/O **AD** and select **Insert connection to address**.

In the following window, open the selection to display the possibilities for this particular connection, and e.g. select ***T400_AbsEnc1**. (the star symbol is located in front of the selection field)



In the CFC, you'll now see a cross reference with the symbolic hardware name ***T400_AbsEnc1**.



Mode of operation

The data transfer format is defined when initializing the block (**RPT**, **NOT**, **PZB**, **ABP**, **MDF**, **MDT**, **MDC**). After this, for SSI encoders, the block starts to cyclically read-out the position values corresponding to its configured sampling time.

If an EnDat encoder is connected, after initialization, the encoder-specific error- and warning masks are first read-out. The encoder signals an internal error by setting an alarm bit. The **AENC** block then reads the alarm status word of the encoder and outputs this at output **YF**. The warning status word of the encoder indicates that the encoder has to be serviced. This information is not critical from a time perspective. It is cyclically interrogated and output at **YW** (it is interrogated every 65536 sampling cycles; for a 10 ms cycle, therefore approximately every 11 min).

For EnDat encoders, the serial position data transfer, under extreme cases, takes longer than 1 ms. Thus, reading a position is sub-divided into two phases, which are processed in two consecutive sampling cycles. After the actual position has been processed, the **AENC** block starts the next position data transfer. The new value is then available one sampling cycle later. The read-in position value and start new measurement are realized in the system mode, which guarantees a high equidistance (time) of the measurements.

Position value and revolution counter (multi-turn encoder) data is transferred, for several encoder types, with a resolution exceeding 23 bit. As, in these cases, when a real format is output (float), information can be lost, the position data is distributed over three connections: Angular position (**YP**), No. of revolutions (**YRC**) and the sum of the two (**YSP**). All of the three outputs are weighted with the normalization factor **NFP**.

To define a zero position, an offset **OFF** can be switched to the output. The offset has the same normalization as the position outputs, and is subtracted from the encoder position.

A position value overflow from the encoder is taken into account by correcting the revolution counter. In order to ensure error-free correction, the **AENC** block must be processed at least twice per revolution. This must be taken into account when defining the sampling time **TA** (recommendation: $TA \leq 10$ ms).

Speed calculation

The **AENC** block calculates the speed (code encoder) or the velocity (length measuring system) from the change in the position values. Two system-related speed limits must be observed. On one hand, code encoders have a speed limit due to their mechanical design (connection **LU**), above which, no valid positions can be determined. The speed actual value is undefined if this limit is exceeded.

The second speed limit is obtained due to the configured sampling time. For single-turn encoders, speed and position can only be uniquely measured, if at least two measurements are made per revolution. If this is not guaranteed, the data from the revolution counter **YRC** also lose their validity. A plausibility check, which uses the magnitude of the speed change, identifies whether this speed limit is exceeded.

In both of these cases, where the speed is exceeded, in addition to a group error message, an appropriate display is made in the error code word. The speed calculation is continued, whereby the block cannot identify from which instant in time, valid speed actual values are again available. Thus, the error messages are not reset.

When determining the speed, it is assumed that the encoder determines the position at the start of the position data transfer. If this is not guaranteed, the speed output is unsteady, even if the encoder is rotating at a constant rate.

I/O:

RPT	No. of steps which can be differentiated between per revolution for code encoders
NOT	No. of revolutions which can be differentiated between for a multi-turn encoder
PZB	No. of non-relevant bits at the start of the position value transfer (only for SSI)
ABP	Position of the alarm bits within the data transfer protocol (only for SSI)
MDF	Clock cycle frequency for data transfer
MDT	Encoder type
MDC	Data coding of the connected absolute value encoder
ENP	Enabling parity monitoring for an SSI encoder
RE	Resetting the block and deleting error signals (QF, YF) and alarms (YW)
NFG	Taking into account the ratio of a gearbox located between the encoder and the drive system
NFP	Normalization factor for position outputs YP, YRC, YSP and the normalization basis for the offset input OFF
NFY	Speed actual value normalization
OFF	Position offset
LU	Maximum operating speed of the encoder
YOP	Initial encoder position in steps
YP	Actual angular position of the absolute encoder
YRC	Revolution counter
YSP	Sum of YP and YRC
Y	Supplies the speed calculated from the position changes of the encoder or the velocity for code-length measuring systems
QF	Group error message. QF indicates whether the actual values for position, revolutions and speed are valid. For QF=1, YP, YRC, YSP, Y are <u>not</u> valid!
QRC	Overflow revolution counter
QNP	Receive new position
YF	Error status word, EnDat encoder
YW	Alarm word, EnDat encoder
YFC	Error output of the AENC function block

You'll find detailed information in the online help and in the SIMADYN D documentation.

Note:

After the absolute position has been determined, the speed is calculated and the position overflows are summed. However, position overflows and speeds are only clearly identified if there is maximum of one overflow between two position sensing operations. For applications with single-turn encoders, it must therefore be ensured, that also at the highest operating speed, there are at least two measurements per revolution.

The calculated speed information does not have the same high accuracy and dynamic performance as, for example, is the case for pulse encoder inputs due to the measuring principle. Thus, generally this technique is not suitable for closed-loop speed controls with a high dynamic performance.

Only one AENC block may be configured per hardware input.

The 2nd serial interface (peer-to-peer or USS) cannot be used at the same time as the absolute value encoder input 2, as the same terminals are used.

Many AENC connections are initialization inputs, where changes only become effective after a restart.

6.7. Incremental encoders

The incremental speed inputs are evaluated using the *NAV* and *NAVS* blocks.

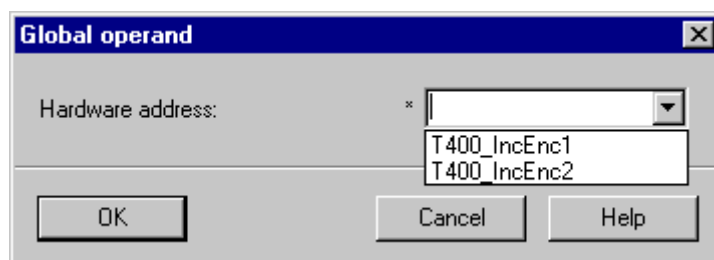
The function blocks belong to the **On/Off** block family of the catalog. Only one function block may be configured for each speed sensing.

The *NAV* block is used, if one (or two independent) speed- or position sensing functions are required. The *NAVS* function block is used for synchronous operation applications.

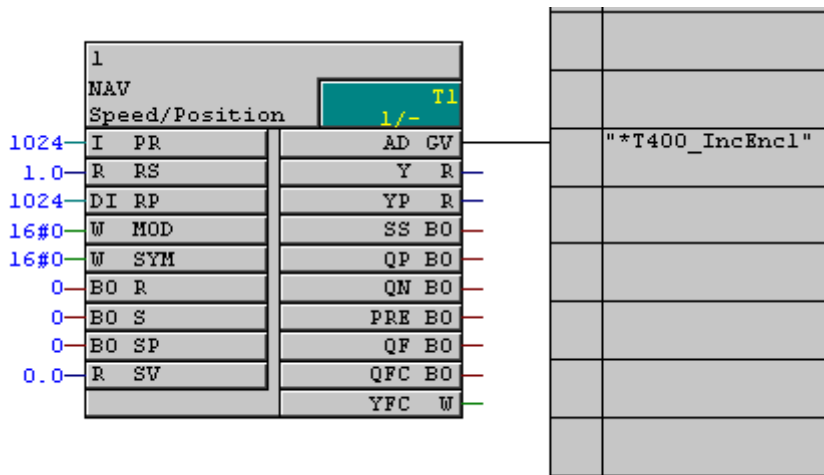
NAV

In order to establish the assignment to the symbolic hardware address, assigned in HW Config (in the example **T400_IncEnc1*), call-up the context menu of the I/O **AD** and select **Insert connection to address**.

In the following window, open the selection to display the possibilities for this particular connection, and e.g. select **T400_IncEnc1**. (the star symbol is located in front of the selection field)



In the CFC, you'll now see a cross reference with the symbolic hardware name **T400_IncEnc1*.



In addition to the actual speed, the block also provides a position actual value:

- Speed:
Inputs **PR** and **PS** are evaluated to output the actual speed sensing at output **Y**.
- Position sensing:
Position information can be retrieved by sensing the individual pulses. These position information are output at **YP**. Output normalization can be specified using **RP**. For **RP=1**, 1.0 is output after one revolution. For instance, a gearbox can be taken into account using other values.
The position actual value can be set or reset using **SP**, **SV**, **R** and **S**. This function is used, for example, for a reference point approach.

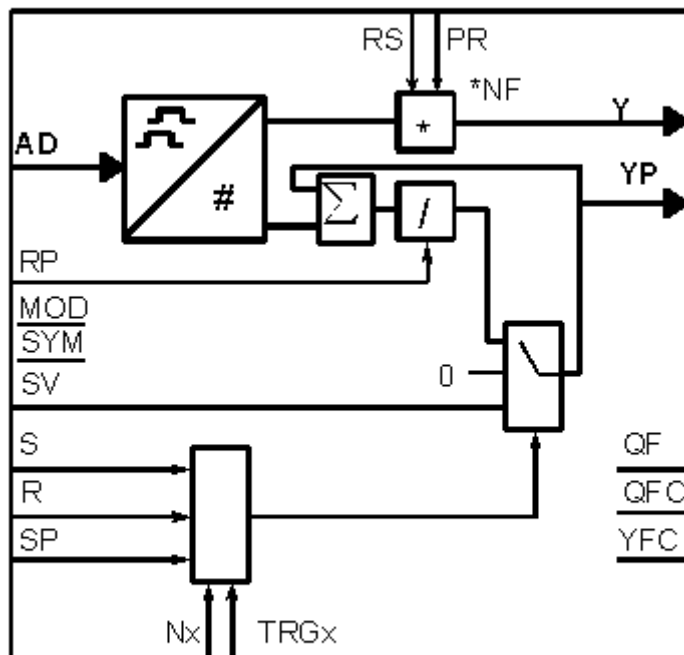
Note

RP and **SYM** are initialization connections, which when changed, only become effective after a restart.

Connect.	Function
PR	No. of pulses per revolution of the encoder.
RS	Rated speed, where 1.0 should be output at Y.
RP	No. of pulses, so that 1.0 is output at YP.
MOD	Operating mode selection: Encoder type, filter, setting type, standstill limit, zero pulse; The data is entered in the hexadecimal format. Additional information is provided in the online help and documentation.
SYM	Operating mode selection: Setting mode and coarse pulse The data is entered in the hexadecimal format. Additional information is provided in the online help and documentation.
R	The position actual value is to 0.0.
S	Set the position actual value to SV (or YP-SV, refer to MOD)
SP	Set the position actual value when synchronizing (e.g. zero pulse, refer to SYM)
SV	Position setting value (is used for SP and S)

Connect.	Function
Y	Actual speed
YP	Actual position value
SS	Position is set when in synchronism
QP	Position actual value overflow in the positive direction
QN	Position actual value overflow in the negative direction
PRE	Pulses were identified
QF	Monitoring error, encoder alarm
QFC	Group fault (cause is displayed in YFC)
YFC	Error code

Block diagram

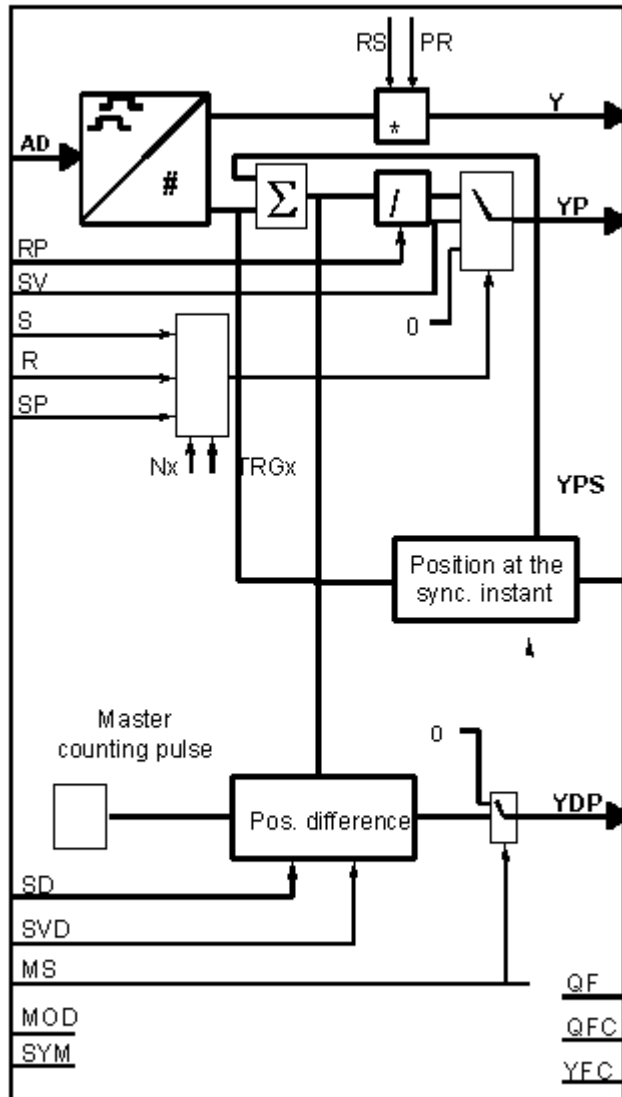


Nx Zero pulse signal
TRG Trigger signal

NAVS

The NAVS function block is the same as the NAV function block, expanded by angular synchronous functions:

Block diagram



Nx - Zero pulse signal
TRGx - Trigger signal

In application software for synchronous operation controls, the pulse sensing of each drive involved is assigned a function block NAVS. In order to differentiate between master- and slave drives, a 1 ("master" or master drive) or 0 ("slave" or slave drive) is configured at input **MS**.

The master simultaneously saves the counter status of its pulse sensing function and that of the slave. This therefore ensures data consistency when calculating the speed-, position- and position difference of the master-slave cascade, i.e. for speed, position and position difference, the pulse numbers are retrieved from the same measuring interval.

Additional information is provided in the online help or in the documentation

RP, **SYM** and **MOD** are initialization connections, which only become effective after a restart.

6.8. Communications

6.8.1. General information

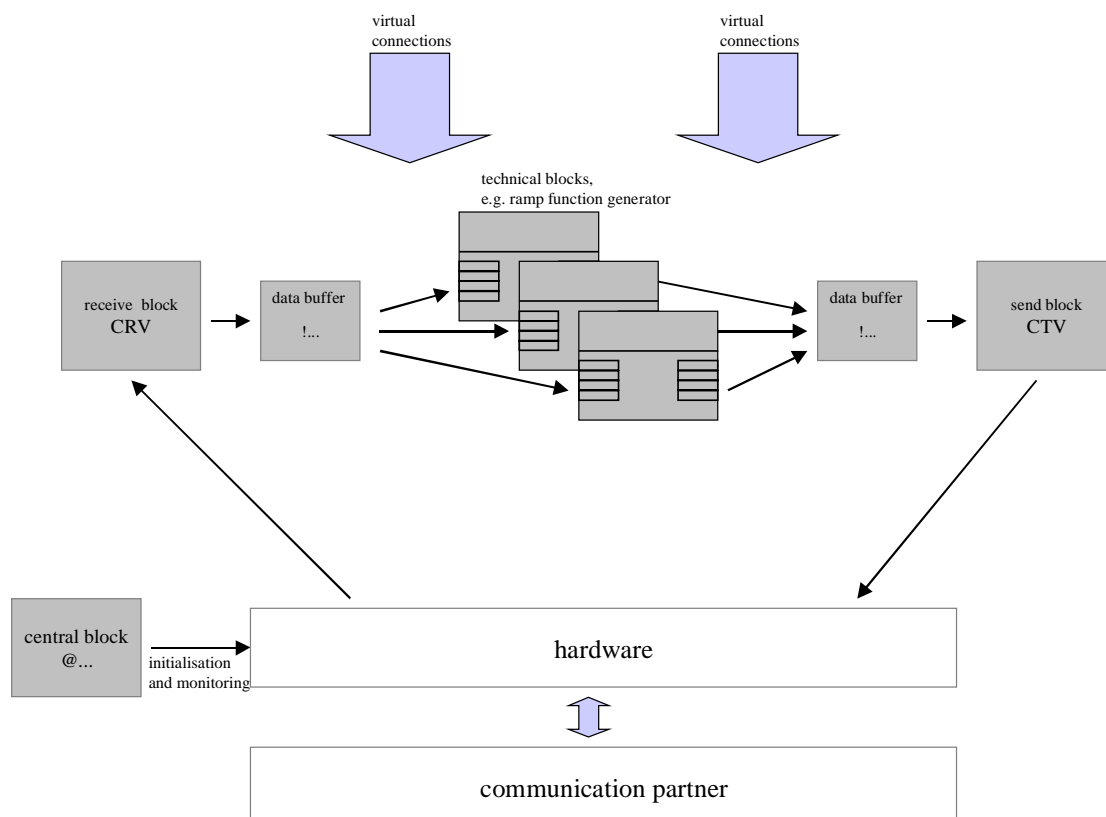
Communications for SIMADYN D are (almost) always the same:

- A central coupling block initializes and monitors communications. Generally, the central coupling blocks must be configured in a sampling time between 32 and 256ms. The central coupling block name always starts with a @ . Coupling-specific data, e.g. baud rate are specified at the central coupling block. Further, the communications connection is specified (module name and possibly connector). General errors cause the block to be disabled which is displayed at output **QTS** of the central coupling block.
- The send- and receive blocks **CTV** and **CRV** send or receive the data from the hardware and save these in a data buffer. Station-related faults/errors are output at outputs **QTS**, **YTS** and **YEV** of the send- and receive blocks.

The name of the data buffer is specified at **CRT** or **CRR** of the send- and receive blocks, and always starts with a !

- The function block accesses the data buffer via virtual connections, and therefore the data to be transferred. The buffer name (with sequence number) is written at the block inputs or outputs.

The following schematic indicates the interrelationships:



- A central coupling block is configured for each coupling type (communications connection).
- A send- and a receive block (**CTV** and **CRV**) are configured for each coupling partner (station).

Station-specific data are entered at the **AT** and **AR** connections of the send- and receive blocks CTV and CRV. These can include, for example, the station number. The data is specified in three stages: 'Channel name.Stage2.Stage3'. The stages are separated by points. Stages 2 and 3 do not have to be specified for all couplings.

- Any channel name can be specified at the first position. This must be unique for serial couplings, i.e. it may not be used twice, and it may not have another significance.
- If Stage 2 is used, then a station address is specified
- If Stage 3 is used, then additional details are specified here, e.g. the numerical format

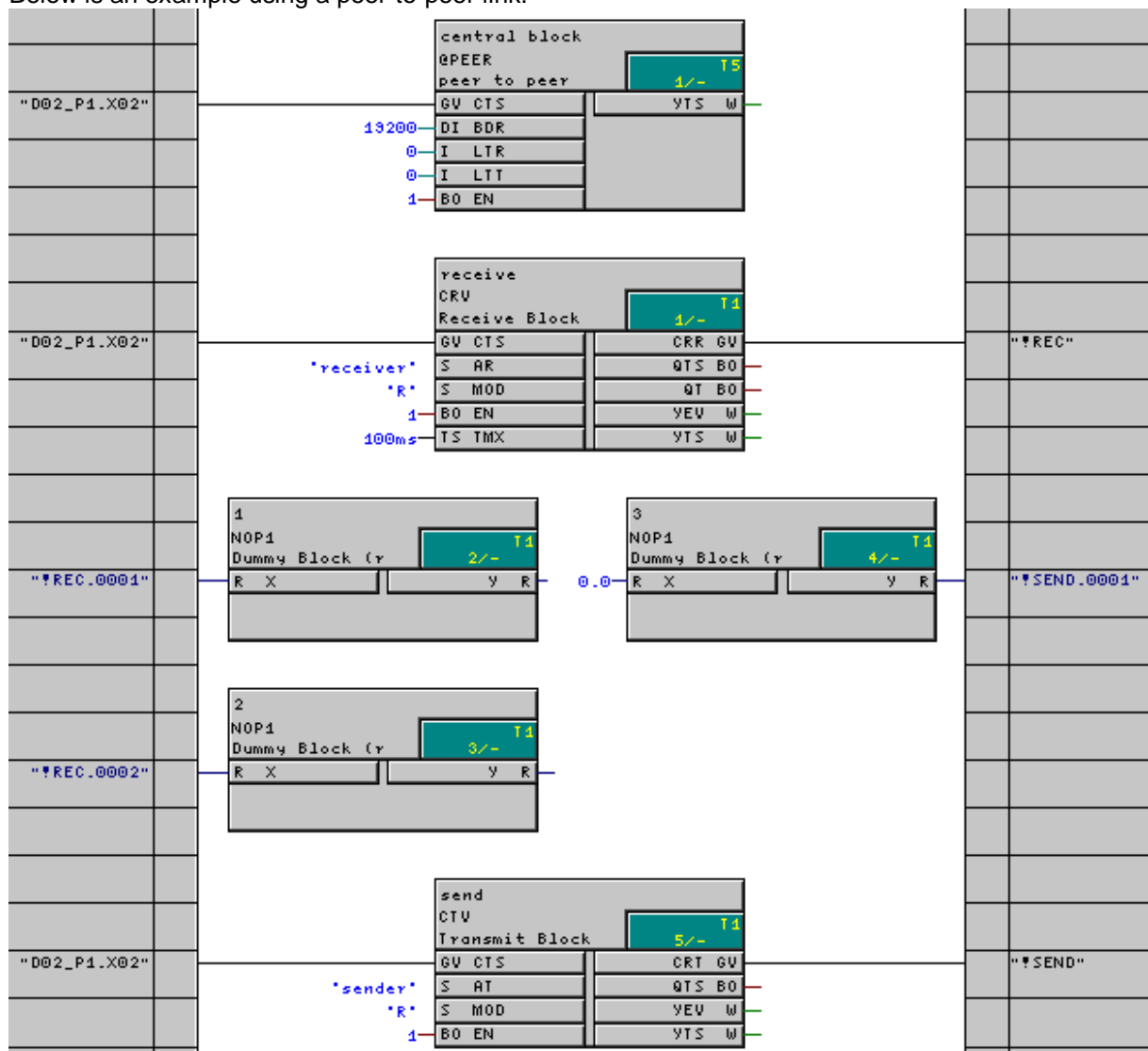
The **AT** and **AR** connections and their configuration for the particular coupling are described in the following sections.

The operating mode is defined at the **MOD** input of the send- and receive blocks CTV and CRV. Generally, 'R' can be used for refresh. The other possibilities are described in the SIMADYN D documentation (communications software).

The sampling time of the send- and receive blocks defines the rate at which net data are transferred!
On the other hand, the central coupling block should always be computed in a slow sampling time

Please observe that the central coupling blocks often have initialization I/O, i.e. changes made at these initialization I/O are only effective after a restart.

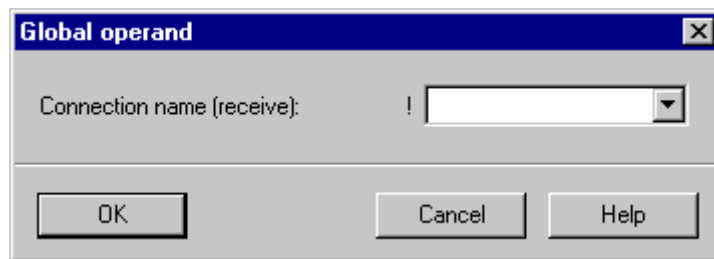
Below is an example using a peer-to-peer link:



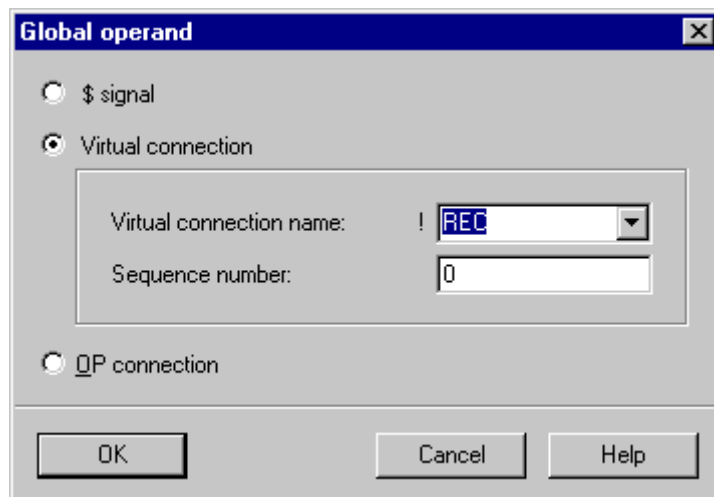
- The @PEER central coupling block initializes the coupling via the 2nd serial interface of the T400 (D02_P1.X02). General coupling parameters are defined at the central coupling block, e.g. the baud rate. (a detailed description of the @PEER is provided in Section 6.8.5)
- The CRV type block receives the data and saves it in the data buffer **!REC**.
- Blocks 1 and 2 read-out the **!REC** buffer and process the information. Naturally, any other blocks can be used instead of the NOP1 blocks.
- Block 3 saves the data in the **!SEND** buffer
- The CTV type block sends the contents of the **!SEND** buffer via the 2nd serial interface of the T400.

In the example, the blocks are computed in the sequence defined by the signal flow, i.e. the receive block is the first and the send block is the last block computed in T1. Thus, a signal only requires one sampling time to completely flow through the program.

The virtual connections are entered in two steps. Initially, the name of the virtual connection is defined: By clicking with the righthand mouse key, select **Insert connection to address** to call-up the **Global operand** menu. A new name is entered at connection **CRR** of the CRV or **CRT** of the CTV block:



In a second step, at a "standard" block I/O, the **Global operand** menu is called-up by clicking with the righthand mouse key:



The possible virtual connection names can be selected in the center section from the pulldown menu (the Fig. above only shows the center buffer section). The sequence number which is entered, specifies which data should be used from the buffer.

Please note that the number for the virtual connections only specifies the sequence in which the files are saved in the buffer, not the precise position!

This means, that you can start with a 0, 1 or any other number which you select. The sequence numbers do not have to be consecutively assigned. For example, steps of 5 can be used.

If you only wish to send a value in the 5th word, you must additionally generate 4 words, and write the first four words in the buffer, as otherwise the value would be sent in the first word.

Dummy blocks, e.g. `NOF1_W`, can be used to generate these words.

The same is true for data receive. If you wish, for example, to only read the third word, you must also read the first two words! In this case, you can also use dummy blocks, e.g. `NOF1_W`, and ignore its output.

Please note, that for several formats, 2 words are used, e.g. for real- and double integer formats. In this case, 2 words are automatically saved in the buffer.

In the example, 4 words are received and 2 words are sent (real numbers are 32-bit values, so that each real value takes-up 2 words).

When compiling, if it is not disabled, a CPU-MAP listing `pnmap.pro` is generated. (Menu **Options | Customize | Compilation**). The precise path data is provided in the compilation protocol, which you can then view with **Details** after compilation, or otherwise, in the **Options | Protocols** menu. You can view this file using the Windows Editor (Notepad.exe). A list of the configured virtual connections is included in the CPU-MAP listing.

Excerpt from a CPU-MAP listing:

Communications configuring - Send direction (virtual connections)

Sender Name	Tx	Sender block Connector	Length	No.	Generator Tx	Connector	
S_BASE	T1	IF_CU.Send_BASE.CRT	12	0001	T1	IF_CU.Sammeln.Y1	2 (INT)
				0002	T1	IF_CU.Sammeln.Y2	2 (INT)
				0003	T1	IF_CU.Sammeln.Y3	2 (INT)
				0004	T1	IF_CU.Sammeln.Y4	2 (INT)
				0005	T1	IF_CU.Sammeln.Y5	2 (INT)
				0006	T1	IF_CU.Sammeln.Y6	2 (INT)
S_PEER	T1	IF_Peer.Send_PEER.CRT	10	0001	T1	IF_Peer.Zustandswort1.QS	2 (WORD)
				0002	T1	IF_Peer.Sammeln-R.Y2	4 (REAL)
				0003	T1	IF_Peer.Sammeln-R.Y3	4 (REAL)

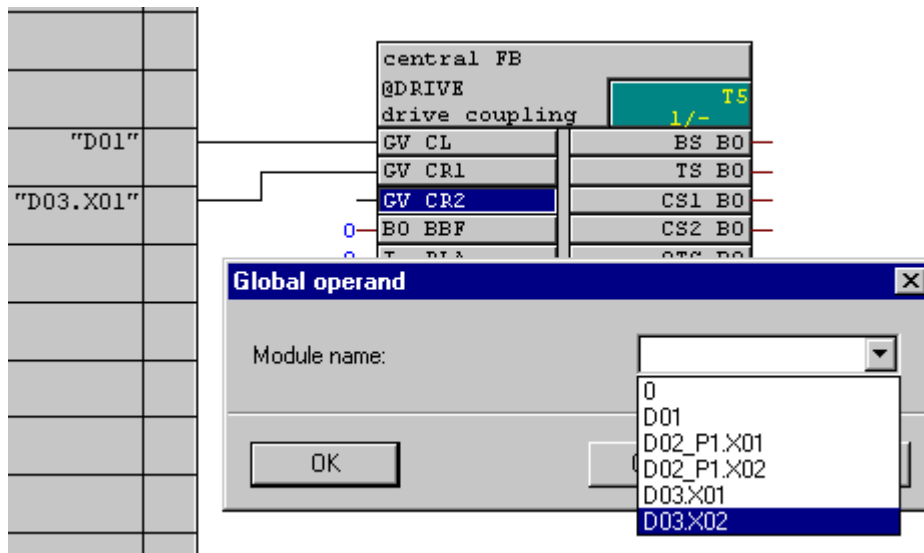
6.8.2. Serial coupling to the configuring PC

Communications between the configuring PC and T400 is realized using a serial interface (RS232) and does not have to be separately configured. The cable required and the selection of the PC/PG interface, is described in Section 5.3.

6.8.3. Coupling to additional modules (e.g. CUx, ADB, CBP, CBx)

The central coupling block for this coupling type is @DRIVE. The existing coupling partners are specified at inputs **CL** and **CR1/CR2**. Their names were assigned in HW Config.

If there are no coupling partners, a 0 is specified at the appropriate input. The names are specified in the context menu of the **CL**, **CR1** and **CR2** inputs. The **Global operand** and the necessary name should be selected there. The name is then specified in the margin. In the following diagram, **CR2** has just been selected, **CL** and **CR1** have already been defined:



The other connections of @DRIVE will be discussed in the following Sections.

Notes:

The module name D03 is the standard name of the module in slot 3 assigned in HW Config. If you have selected another name, this will be displayed instead of D03.

Depending on the number of interfaces of the modules used, several entries are available for selection, e.g. two for ADB.

D01_P1.X01 and D01_P1.X02 are the serial interfaces of the T400. It is not practical to specify these at @DRIVE.

@DRIVE must be configured in a sampling time between 100 ms (not 10 ms) and 200 ms).

6.8.3.1. In the electronics box of a drive

In this case, in HW Config, an SRD was selected as subrack, and there is always a CUx at slot 1 and the T400 is inserted at slot 2.

The name of the CUx, which was assigned in HW Config, e.g. D01 should be specified at connection **CL**, as this is inserted at slot 1.

The following possibilities exist for slot 3 (if D03 was selected as the communications module name) :

Module		CR1	CR2
None		0	0
CBx		D03	0
ADB	only below	D03.X01	0
	only on top	0	D03.X02
	both	D03.X01	D03.X02

As the basic drive module CUx is responsible for initializing a communications module in the electronics box, no data has to be specified in the T400 regarding station address, baud rate and similar. The only thing that has to be specified in the T400 is if there is a communications module in slot 3 and whether it involves a CBx or an ADB.

For an ADB, only the ADB is configured in HW Config. It is not specified in HW Config whether one or two sub-modules are used. At @Drive, it is only specified, at CR1 and CR2, whether a sub-module is inserted at the slots or not.

I/O of @Drive:

CL	The name of the CUx which was configured is specified here.
CR1	If relevant, the configured name and the ADBs slots used or the configured name of the CBx are specified here Refer to the sep. table
CR2	
BBF	Parameter range of the parameters defined in the T400, always set to 0 in the electronics box
PLA	Language selection for technology parameters, refer to Section 7
CF1	The SIMADYN D data types REAL and SDTIME are transferred as floating parameter data type (CF1=1) or as 32-bit integer "I4" (CF1=0) via the parameter channel of the first/lower communications module; refer to text.
CF2	The SIMADYN D data types REAL and SDTIME are transferred as floating parameter data type (CF2=1) or as 32-bit integer "I4" (CF2=0) via the parameter channel of the first/lower communications module; refer to text.
TF	No significance in the electronics box
PEN	Enables parameter changes: 1 = enabled, 0 = inhibited
BS	1 = Coupling to CUx OK, 0 = coupling faulted
TS	No significance in the electronics box
CS1	1 = coupling to CBx / ADB (bottom) OK, 0 = coupling faulted
CS2	1 = coupling to ADB (top) OK, 0 = coupling faulted
QTS	1 = block operational, 0 = block is disabled with error output at YT1
YT1 – YT6	0 = parameter channels are error-free, additional information, refer to online help

Up to 16 words can be sent and received to/from the basic drive and to/from each communications module.

For parameter tasks, values can be transferred to the T400 in the floating point- or in the double-integer format. Inputs CF1 and CF2 can be used to define, for each interface, as to whether the parameter values should be interpreted as floating-point values. This is practical, if a system is used as communications partner, which can process floating-point values, e.g. a central SIMADYN D system.

Only a unique channel name must be specified at the **AT** and **AR** connectors for coupling to the basic drive or the communication modules:

Address Stage2	Address Stage3

Example: 'NAME'

If a T400 is being used, the CUx cannot send data directly to a communications module. This data must first be sent from the CUx to the T400. This then receives it and the T400 transfers it on to the communications module.

Communications to CUx is realized via dual port RAM. In this case, it is irrelevant for the CUx, whether a CBx, a T400 or an ADB (when the lower slot is being used!) is the coupling partner for the CUx, as they all behave the same. Thus, the CUx is parameterized the same for all of these three cases.

In the first word, which is sent to CUx, bit 10 (counting starts with 1) must be set, so that the CUx accepts the process data as being valid. Thus, in the first word, the control word should be transferred with bit 10 set.

The values which the CUx should send to T400, are specified in the following basic drive parameters:

CU1, CU2, CU3	P694, Index 1 = 1 st word ... Index 16 = 16 th word
CUVC, CUMC, CUD1	U734, Index 1 = 1 st word ... Index 16 = 16 th word

The values sent from the T400 to the CUx, are assigned in the CUx, range 3000:

CU1, CU2, CU3	Enter 3001 to 3016 into the appropriate parameters; enter word 1 for 3001 and word 16 for 3016 e.g. P554.1=3001 for the On command of T400, P443.1=3002 for the main setpoint from T400
CUVC, CUMC, CUD1	The first word is saved in connector K3001 and the 16 th word in K3016

In the following example, a communications board is not used. Thus, **CR1** and **CR2** are assigned **0** at @DRIVE.

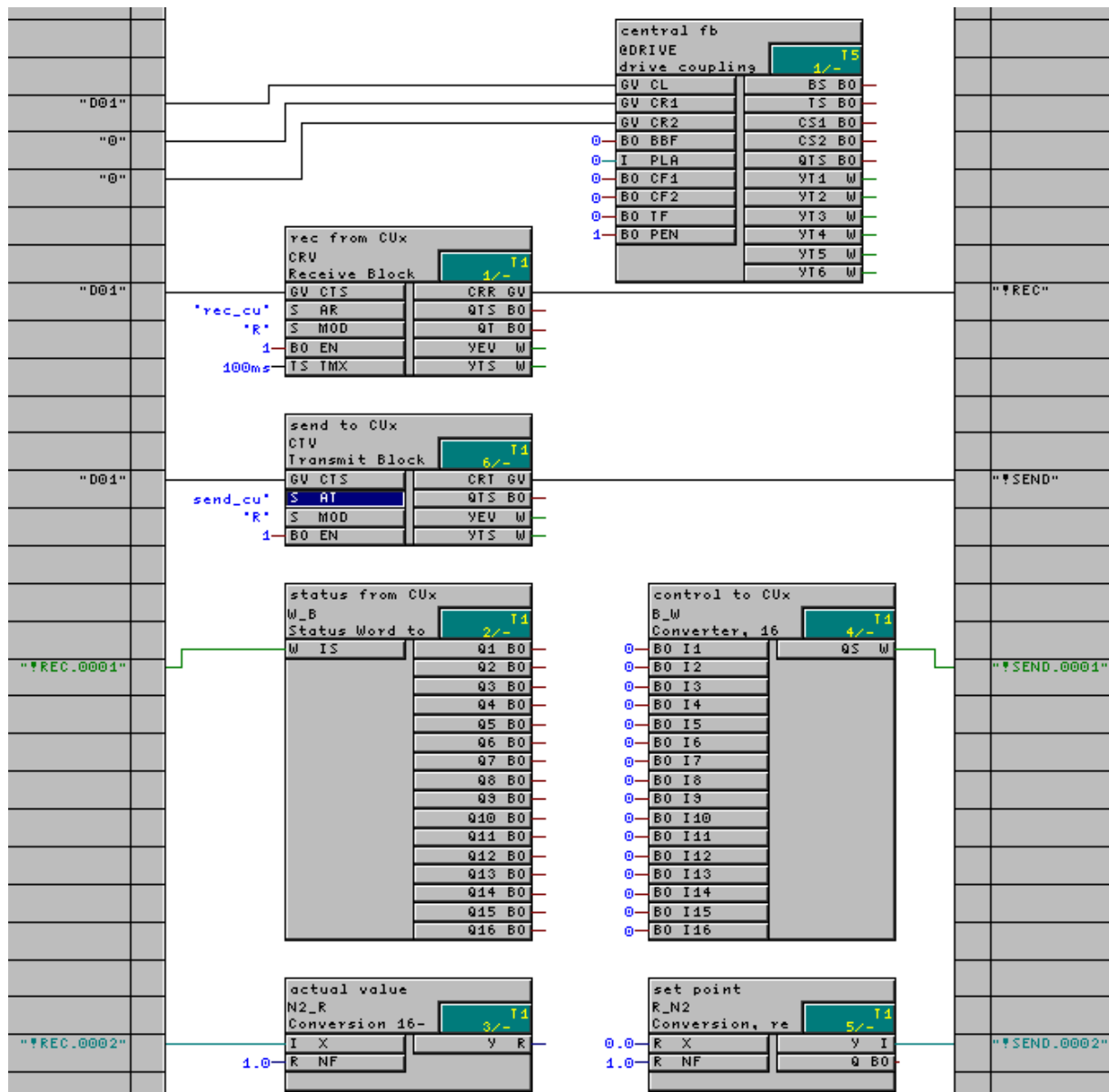
The lower four blocks are conversion blocks:

The type **w_b** block splits-up the status word into individual bits.

The type **B_w** block combines the control word.

N2_R and **R_N2** convert the floating-point numbers, used in the T400, into the normalized format of the drives and vice versa.

The control-related part of the user program must be located between them from a time perspective. In order to transfer additional data, virtual connections must be specified at the other blocks.



In order to expand the example above by a communications module, the following supplements must be made:

The communications module must be specified and identified in HW Config, e.g. D03.

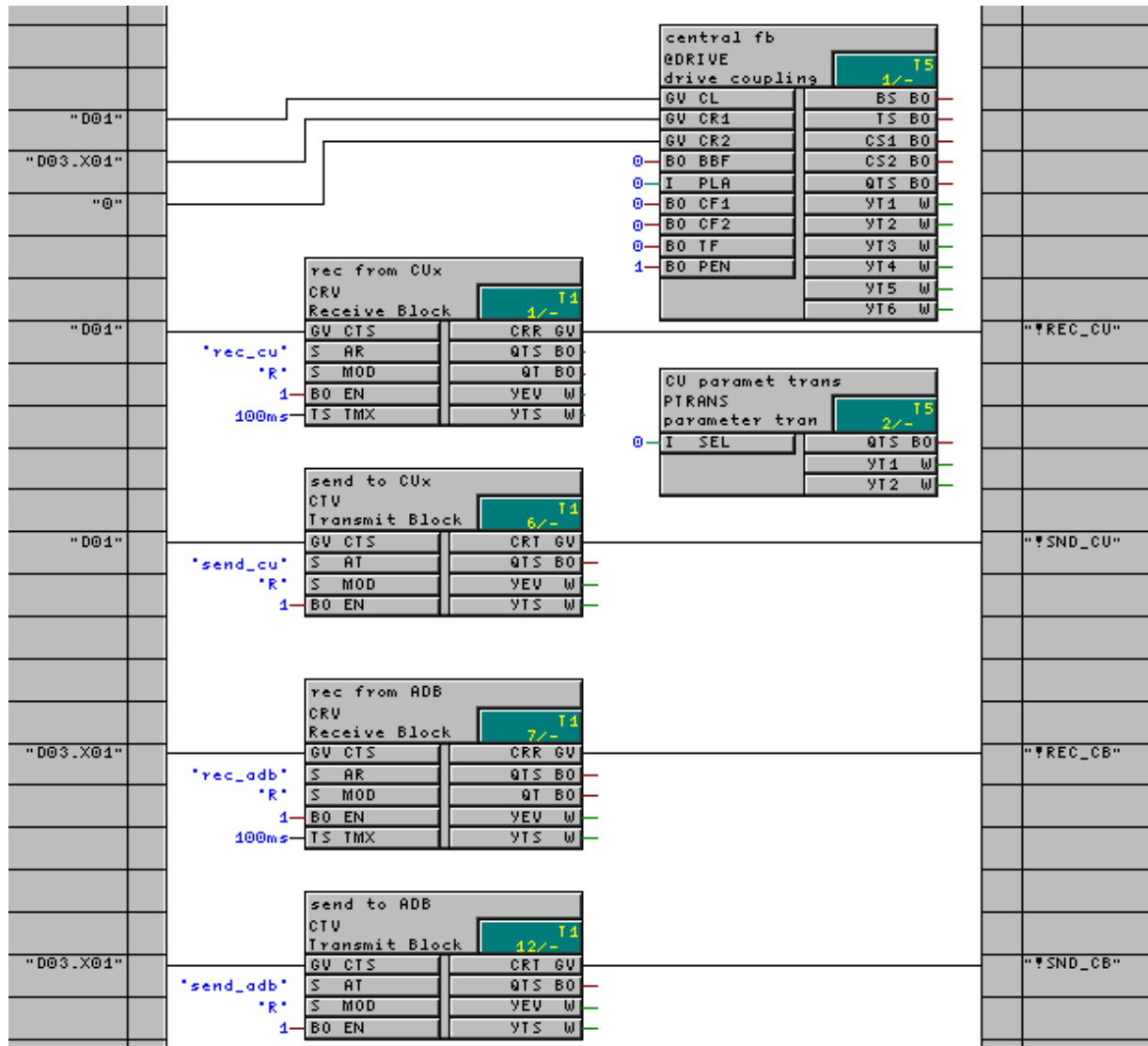
The communications module is also specified at connection **CR1** or **CR2** of @Drive.

One send and one receive block, type **CTV** and **CRV** are configured for every coupling partner. For ADB, each module which is inserted should be considered a coupling partner.

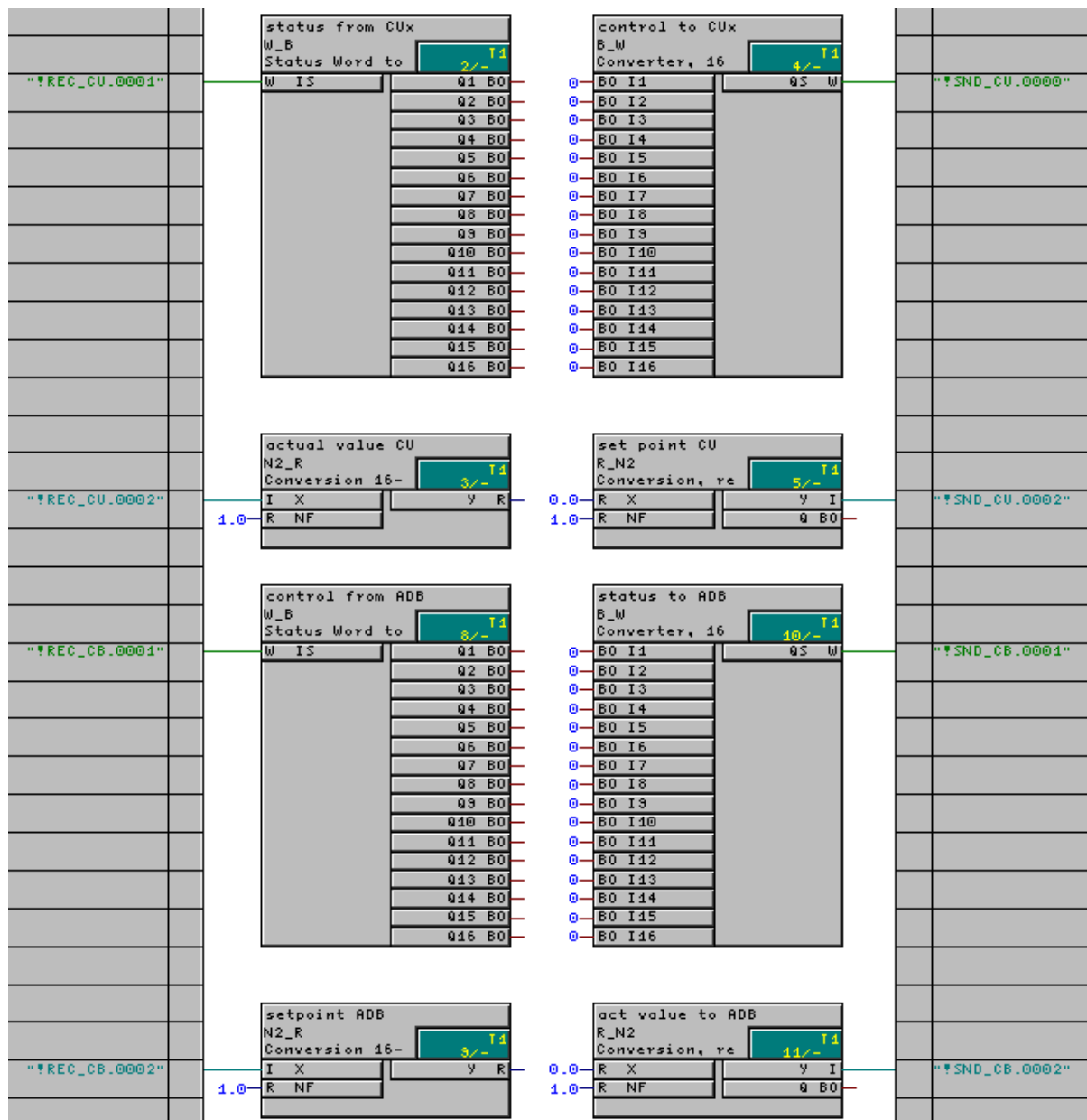
The **PTRANS** block must be used to transfer parameter tasks to the CUX, which were received via the communications.

Here is an example with CUx, T400 and a CBP at the lower slot of an ADB:

Initially, the central-, send- and receive blocks:



Here are the blocks which generate/process the data.



If the blocks on this sheet are connected with one another 1:1, the T400 does not have a technological function, and transfers the receive data to the other module unchanged.

6.8.3.2. In the SRT400 technology subrack

In this case, in HW Config, an SRT400 was selected as subrack, and a T400 is always inserted at slot one.

In the following text, this is designated as the first T400, even if there is no second T400.

In an SRT400 with two T400, @DRIVE must be configured for both.

The following data have to be entered at @Drive of the first/only T400:

- A 0 must be entered at **CL**
- There are two possibilities for **CRx** (slot 2), (if, e.g. D02 was selected as name for the communications module):

Module		CR1	CR2
None		0	0
T400		D02_P2	0
CBx		D02	0
ADB	only bottom	D02.X01	0
	only top	0	D02.X02
	both	D02.X01	D02.X02

If a second T400 is used, then the following entries must be made at its @Drive:

- The name of the first T400 is specified at **CR**, standard: D01_P1
- **CL1** and **CL2** are always 0

Only the ADB is configured in HW Config if an ADB is used. It is not specified in HW Config whether one or two sub-modules are inserted. At @Drive, CR1 and CR2 are only used to specify as to whether a sub-module is inserted at the two slots.

I/O of @Drive:

CL	First T400: 0, second T400: configured name of the first T400.
CR1	If required, the configured name and slots of the ADBs used, the configured name of the CBx or the second T400 should be specified here. Refer to the separate table
CR2	
BBF	Parameter range of the parameters defined in the T400, if 2 T400 are used, use BBF=1 for the first, otherwise always BBF=0
PLA	Language selection for technology parameters, refer to Section 7
CF1	The SIMADYN D data types REAL and SDTIME are transferred as floating parameter data type (CF1=1) or as 32-bit integer "I4" (CF1=0) via the parameter channel of the first/lower communications module; refer to text.
CF2	The SIMADYN D data types REAL and SDTIME are transferred as floating parameter data type (CF2=1) or as 32-bit integer "I4" (CF2=0) via the parameter channel of the first/lower communications module; refer to text.
TF	The SIMADYN D data types REAL and SDTIME are transferred as parameter data type floating (TF=1) or as 32-bit integer "I4" (TF=0) via the parameter channel of the TECHBOARD interface (second T400); refer to the text.
PEN	Enables parameter changes: 1 = enabled, 0 = inhibited
BS	1 = coupling to the first T400 OK, 0 = coupling faulted
TS	1 = coupling to the second T400 OK, 0 = coupling faulted
CS1	1 = coupling to CBx / ADB (bottom) OK, 0 = coupling faulted
CS2	1 = coupling to ADB (top) OK, 0 = coupling faulted
QTS	1 = block operational, 0 = block is disabled with error output at YT1
YT1 – YT6	0 = error-free status of the parameter channels, additional information, refer to online help

Up to 16 words can be sent and received to/from each communications module or between the two T400.

For parameter tasks, values can be transferred to the T400 in the floating-point- or in the double-integer format. Inputs **CF1**, **CF2** and **TF** can be used to define, for each interface, as to whether the parameter values should be interpreted as floating-point values. This is practical, if a system is used as communications partner, which can process floating-point values, e.g. a central SIMADYN D rack.

A unique channel name must be specified at the connectors **AT** and **AR** for the coupling to the communication modules or to the other T400:

Address Stage2	Address Stage3

Example: 'NAME'

The T400 initializes a communications module in the SRT400, i.e. information must be provided in the T400 about the station address, baud rate and similar information. The `CBCONF` block is used for this purpose. If an ADB is used with two modules, a dedicated `CBCONF` must be used for each module.

The `CBCONF` block saves the configured configuration data in the administration range of the communications module. It automatically executes this when the T400 starts.

In the RUN condition, it provides diagnostics data from the communications module at its outputs.

New configuration data can be transferred to the communications module in the RUN condition with a positive (rising) edge at input **SET**.

I/O of the `CBCONF`:

CTR	Configured communications module name CTR=<module name> for CBx, .CTR=<module name>.<connector> for ADB
SET	With a positive edge at this output, the configuration data are re-sent to the COMBOARD.
MMA	Station number (e.g. USS: 0..30, PROFIBUS DP: 3..125)
P01-P28	Max. 28 additional COMBOARD-specific configuration parameters.
QTS	Block status:· QTS=1: Block is operational.· QTS=0: The block is disabled with error output at YTS
YTS	Error number 0 = error-free, for additional information refer to online help
D01...D28	Max. 28 words of diagnostics data

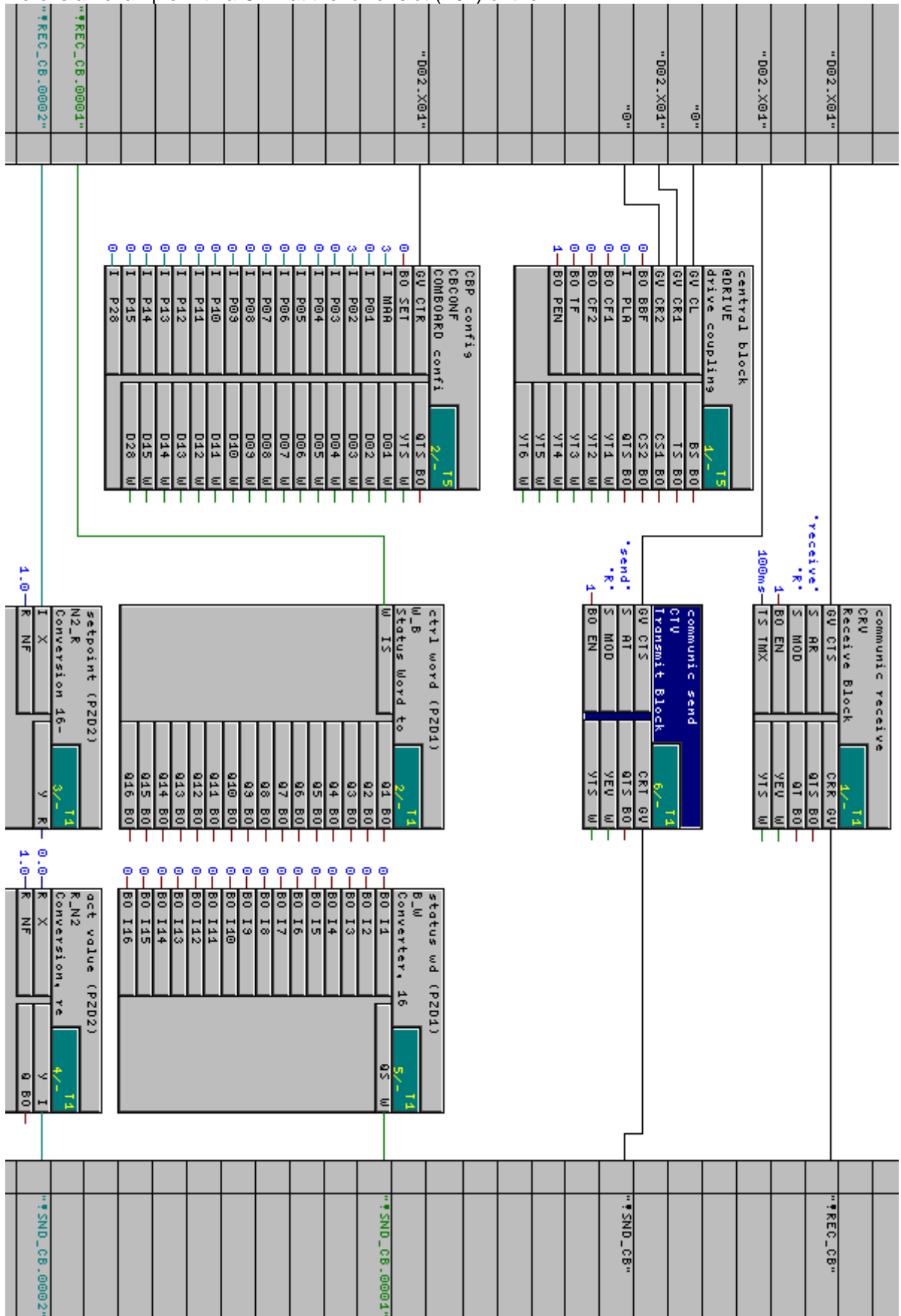
The parameters to initialize the communication modules should be taken from the following tables. The "CB-Param" and "SCB-Param" columns establish the assignment to the User Manuals of the communication modules.

Significance of the configuration I/O for several communication modules:

Input	CB-Param	CB1/CBP: DP	CB2: CAN	SCB-Param	SCB2: USS slave	SCB2: Peer
MAA	P918	Bus address	Bus address	P683.2	Bus address	(not used)
P01	P696	(not used)	PKW task	P682	SCB1/SCB2 protocol selection: 0:CAN, 1:USS 4wire, 2:USS 2 wire, 3:Peer	
P02	P697	PPO type	PZD receive	P685.2	PKW: 0: no, 127: yes, 3: oneword, 4: oneDword	(not used)
P03	P698	(END)	PZD send	P686.2	Process data: No. of words	(not used)
P04	P699		PZD send length	P684.2	Baud rate	
P05	P700		PZD send rate	P687.2	Telegram failure time	
P06	P701		PZD receive broadcast		(END)	(END)
P07	P702		PZD receive multicast			
P08	P703		PZD receive, transverse			
P09	P704		PKW task broadcast			
P10	P705		Baud rate			
P11	P706.1		CAN layer			
P12	P706.2		Bus timing			
P13	P706.3		(END)			
P14	P706.4					
P15 – P28						

Output	CB-Param	CB1: DP	CB2: CAN	SCB-Param	SCB2: USS slave	SCB2: Peer
D01	r731.1		Configuration error			
D02	r731.2		...			
D03	r731.3					
D04	r731.4				Counter, error-free telegrams	
D05	r731.5				Counter, error-free telegrams	
D06	r731.6				Error, telegram frame error	
D07	r731.7				Counter, overrun	
D08	r731.8				Counter, parity	
D09	r731.9				Counter, start of telegram error	
D10	r731.10				Counter, end of telegram error	
D11	r731.11				Counter, BCC error	Counter, lengths error
D12	r731.12				Counter, lengths error	
D13	r731.13				Counter, timeout	
D14	r731.14					
D15	r731.15					
D16	r731.16					
D17-D27						
D28	r731.29				Init error: 0: OK 1: Baud rate 2: Process data length 3: Parameter data length 4: Station number 5: Module type	

Here is an example with a CBP at the lower slot (X01) of the ADB:



At CBCONF, the connections P16 to P27 and D16 to D27 have been switched so that they are not visible.

6.8.4. USS

6.8.4.1. USS master

The USS-master coupling allows the following

- Process data to be transferred
- Parameters to be handled and visualized.

The following are available to transfer process data, approx.:

- 16 send channels (actual values) and
- 16 receive channels (setpoints)

Possible coupling partners include:

- Drives
- VD1 numerical display
- OP2 operator control panel
- Additional T400

The USS bus is a master-slave bus, where all of the nodes are connected to one another via a bus cable. Data is transferred using a half-duplex technique along a 2-wire cable according to the RS485 Standard.

A terminating resistor is required at both physical ends of the bus.

Terminating resistors are provided in the T400, and can be switched-in using the following switches, if the T400 is located at one end of the bus:

	Terminals	Switch S1
Serial interface 1	70 / 71	1, 2
Serial interface 2 (2-wire)	74 / 75	3, 4

Additional information on VD1 and OP2 and how they are configured is provided in Catalog DA99 or in the SIMADYN D documentation.

I/O of the @USS_M:

CTS	T400 module name and connector X01 or X02, on which the USS master should run. Note: When using X01 for USS with RS485, switch S1/8 must be set to ON on the T400 module. It is then no longer possible to execute Service/Diagnostics/Load via X01.					
BDR	Baud rate, value range: 9600, 19200, 38400 and (only on X02) 93750, 187500					
MTL	Mirror telegram to test the data transfer link (MTL=1), request from slaves or standard operation with transfer of the configured net (useful) data (MTL=0).					
EN	Enable initialization (default value: 1; i.e. the block is operational)					
LL1, LL2	Slave life list, bit-coded. If a slave has correctly responded to the last master telegram, then the associated bit is set, otherwise it is not set. The following table shows the assignment of the bits to the slave station numbers:					
	Output	Bit 16	Bit 15	...	Bit 2	Bit 1
	LL1	15	14	...	1	0
	LL2	-	30	...	17	16
CBC	(Counter Bus Cycle) This counter is incremented at each USS bus circulation					
CRE	(Counter Receive Error) This counter is incremented with each error in a receive telegram (character, telegram frame, length, length sum, etc.)					
QTS	Block status: 1: Block is operational. 0: Block is disabled with error output at YTS					
YTS	Status display: 0: OK status, additional information refer to online help					

All of the inputs of the @USS_M are initialization inputs, where changes only become effective after a restart.

A USS connection cannot be simultaneously used with absolute value encoder input 2, as the same terminals are used.

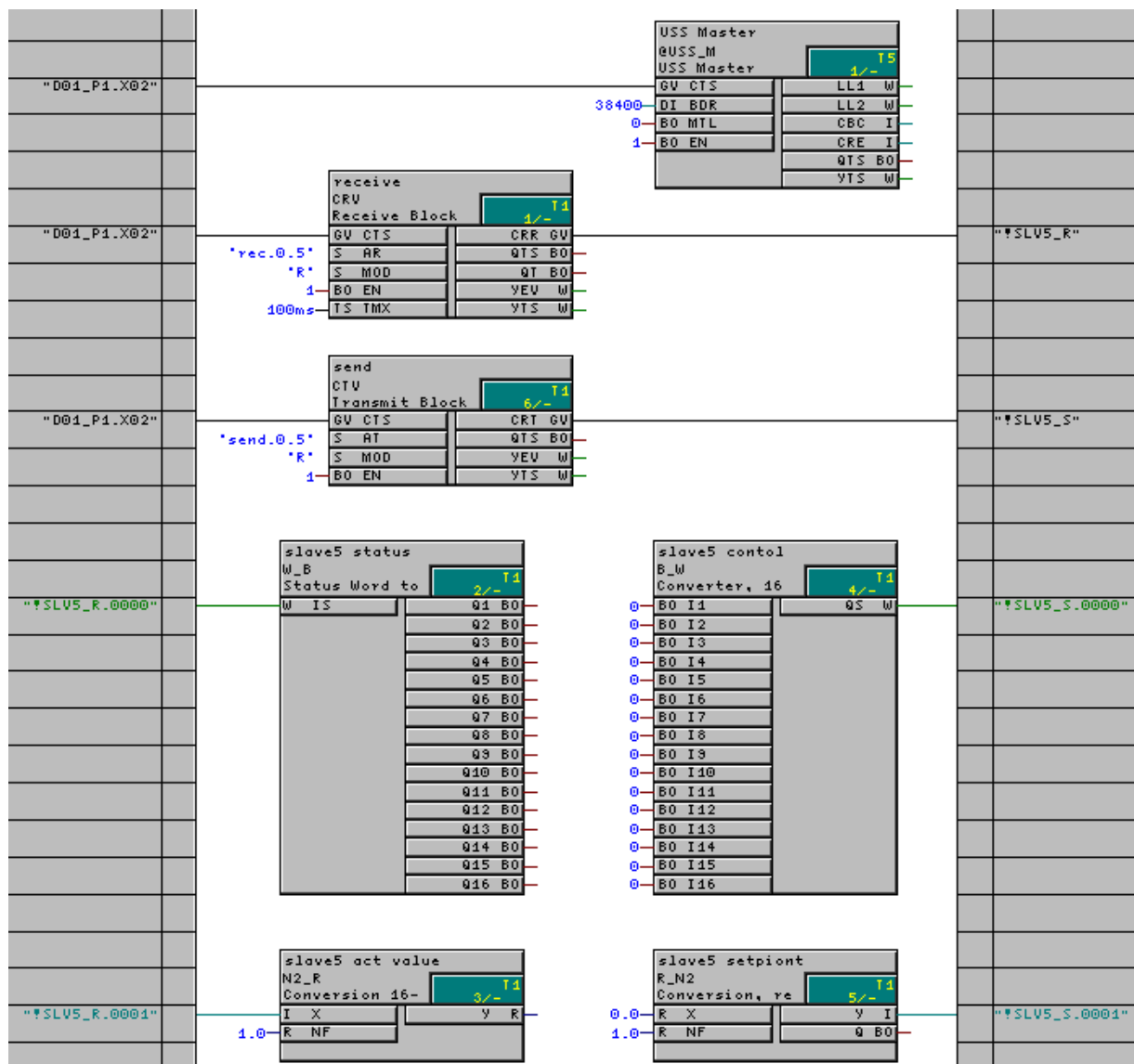
In addition to a unique channel name, additional information must be specified at the **AT** and **AR** connectors:

Address Stage2	Address Stage3
Telegram type: 0: Drive telegram 1: OP2 telegram	Station address: 0 –30: Slave station address 99: Broadcast

Example: 'NAME.1.3'

It is not possible to use broadcast telegrams to the drives (telegram type: drive)!

A drive with station address 5 is addressed in the following example:



6.8.4.2. USS slave

The USS slave coupling permits

- Process data to be transferred
- Parameters to be handled and visualized

A maximum of the following is available to transfer process data

- One send channel (actual values) and
- One receive channel (setpoints).

Up to 32 process quantities, each 16 bits can be transferred via these channels.

- It is not possible to simultaneously use two USS couplings (USS master, USS slave) at serial interfaces X01 and X02.
- A USS coupling on serial interface X02 cannot be simultaneously used with the absolute value encoder input 2, as the same terminals are used.
- When using X01 for USS with RS485, switch S1/8 must be set to the ON position on the T400 module. Service/Diagnostics/Load is then not possible via X01

Data is transferred using the half-duplex technique via a 2-wire cable according to the RS485 Standard.

However, the USS slave coupling can also be used to communicate with a PC via serial interface 1 (with S1/8=OFF).

A terminating resistor is required at both physical ends of the USS bus.

Terminating resistors are provided in the T400, and they can be switched-in using the following switches:

	Terminals	Switch S1
Serial interface 1	70 / 71	1, 2
Serial interface 2 (2-wire)	74 / 75	3, 4
Serial interface 2 (4-wire)	72 / 73 / 74 / 75	3, 4, 5, 6

The USS-slave coupling can also be used to communicate to a PC via the serial interface (with S1/8=OFF). In this case, terminating resistors are not required.

Only a unique channel name has to be specified at connectors **AT** and **AR**:

Address Stage2	Address Stage3

Example: 'NAME'

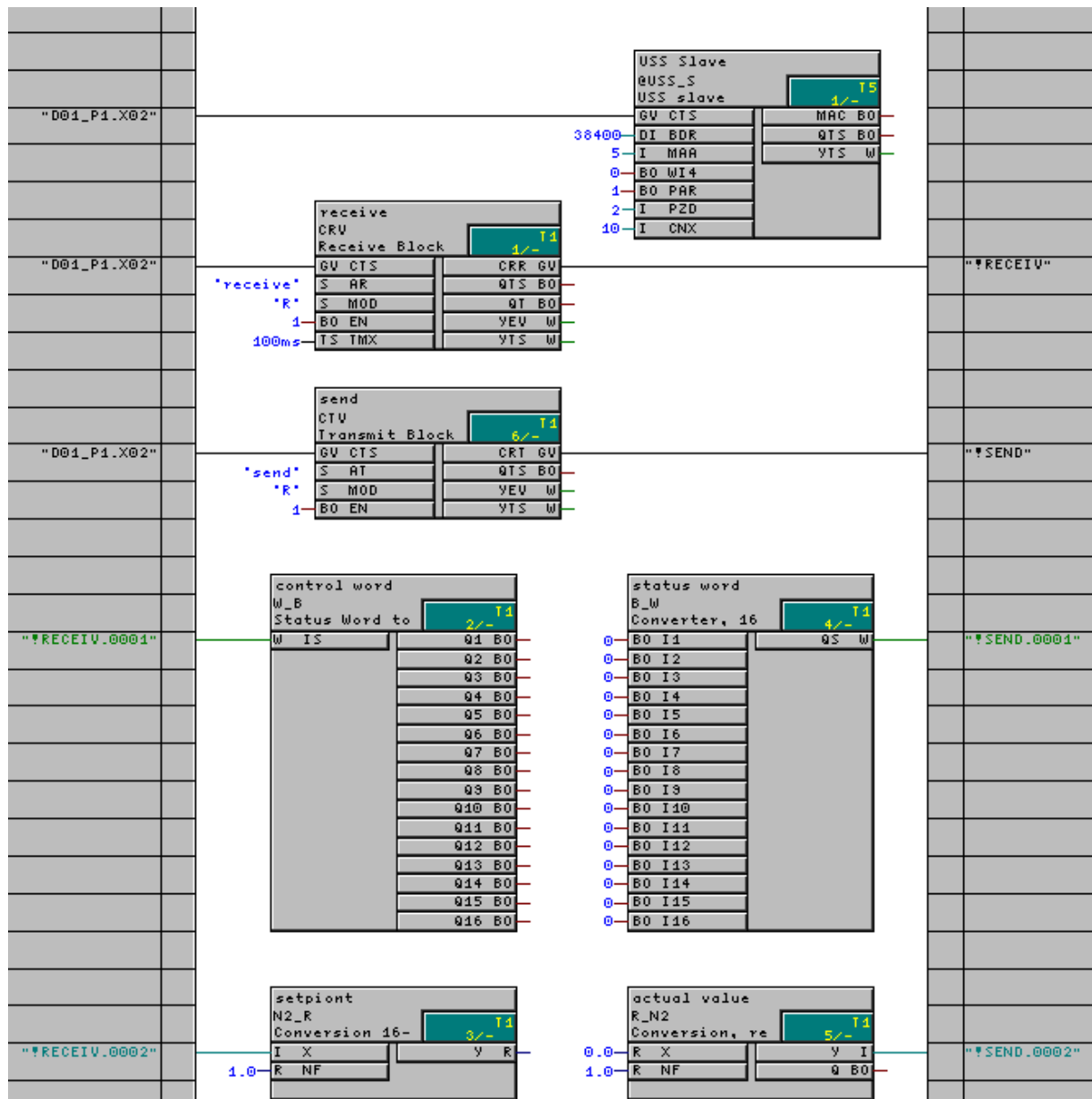
Notes:

- If more data are configured, using virtual connections, then are specified at connector PZD of the function block @USS_S, then the excess data are cut-off.
- If fewer data are configured, using virtual connections, then are specified at connector PZD of the function block @USS_S, then the telegram is filled with "zeros".
- If you configured more data, using virtual connections, then are specified at connector PZD of the function block @USS_S, then the excess data is filled with "zero".
- If you have configured less data, using virtual connections, than are specified at connector PZD of the function block @USS_S, then the excess receive data are rejected.
- The telegram is received asynchronously to the sampling time of function block CRV, i.e. the sampling time of function block CRV is the maximum deadtime between receiving and processing data.
- The send telegram is started asynchronously to the sampling time of function block CTV after a telegram has been received.

I/O of the @USS_S:

CTS	T400 module name and connector X01 or X02 at which the USS master should run. Note: When using X01 for USS with RS485, switch S1/8 must be set to ON on the T400 module. It is then no longer possible to execute Service/Diagnostics/Load via X01.	
BDR	Baud rate, value range: 9600, 19200, 38400 and (only on X02) 93750, 187500	
MAA	Own USS station address, value range: 0...30	
WI4	Setting the data transfer cable:	
	At connector X01:	At connector X02:
	0: RS485/2 wire	0: RS485/2 wire
	1: RS232	1: RS485/4 wire
PAR	Transferring parameter tasks (for processing using @DRIVE) in the USS telegram (PAR=1) or not (PAR=0). When connecting an OP1 or SIMOVIS, PAR=1 must be configured due to the telegram structure. For PAR=0, only process data are transferred.	
PZD	Number of process data words (each 16 bits), which are transferred in the USS telegram. If more process data (using virtual connections) are configured, then these are cut-off; if fewer data are configured, the telegram is filled with zeros. The entry is identical for the send- and receive directions. Value range: 0...32	
CNX	Scan rate to set the monitoring time: If a valid telegram is not received from the master within CNX x sampling time, then output MAC is set to 0. After a valid telegram has been received, MAC is set to 1. The monitoring is disabled with CNX=0 (or negative values), and then MAC remains 0.	
MAC	Master is active. (Also refer to CNX)	
QTS	Block status: 1: Block is operational. 0: Block is disabled with error output at YTS	
YTS	Status display: 0: OK status, additional information refer to online help	

Here is an example in which 2 words are exchanged:



6.8.5. Peer to peer

Up to 5 words can be exchanged via the peer to peer coupling.

I/O of the @PEER

CTS	T400 module name and connector X02.
BDR	Baud rate, value range: 9600, 19200, 38400, 93750, 187500
LTR, LTT	Telegram length, receive and send in words (16 bits). Value range: 0 to 5. If more data (process data, virtual connections) are configured, then these are cut-off; if fewer data are configured, the telegram is filled with zeros. When a "0" is entered, the telegram length aligns itself to the number and type of configured virtual connections.
EN	Enable initialization (initialization input)
YTS	Status display: 0: OK status, additional information refer to the online help

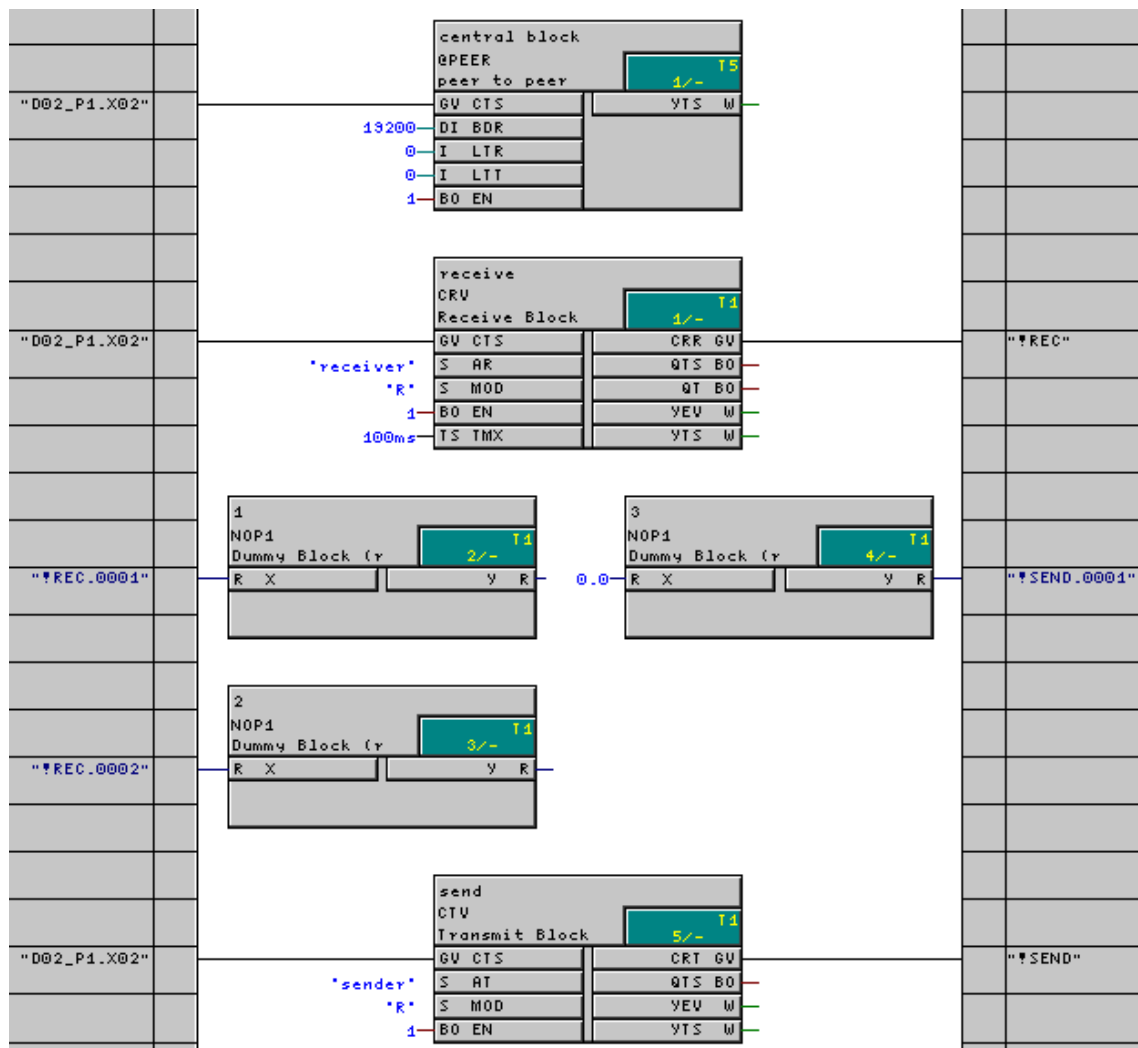
A unique channel name must be specified at the **AT** and **AR** connectors:

Address Stage2	Address Stage3

Example: 'NAME'

The peer to peer connection cannot be simultaneously used with absolute value encoder input 2, as the same terminals are used.

As the transferred values are floating-point values, i.e. 32-bit values, in the example, 4 words are received and 2 words are sent.



6.8.6. Simolink (with SLB)

It is presently not possible to use the SLB on the ADB, in conjunction with the T400. The SLB must be directly inserted on the CUx, and it only communicates with the CUx. If data are transferred which are intended for the T400, they must first be sent by the CUx to the T400. The same is essentially true in the opposite direction: T400 sends to the CUx and this, in turn, to the SLB. This results in delay times for the drive coupling.

6.8.7. Profibus (with CB1 or CBP)

This is a coupling to a additional modul, a CB1 or a ADB with a CBP on it. Details are described in capter 6.8.3.

6.9. Parameterization

6.9.1. Application

- Values at the I/Os of function blocks can be viewed and changed using **parameters** (inputs and outputs).
- **BiCo** technology (binectors/connectors):
Connections between function blocks with parameters can be changed by configuring so-called technology connectors at the block outputs (@TC_).
- The parameters can be viewed and changed using the unit **operator panels** (PMU, OP1S) or via the serial interface (USS, PROFIBUS-DP), if the T400 is used in a drive unit (MASTERDRIVES, SIMOREG 6RA70).
Parameters, e.g. from the **SIMOVIS** PC operator control program or from an automation system, can be edited via the serial interfaces.
- All parameter changes are saved in an **EEPROM**. This permits a **maximum of 100,000 write accesses!** This limitation must be observed, if frequent write accesses are made, e.g. from an automation system via PROFIBUS or USS (e.g. in the form of a "permanent change task"). Continuous changes should be made as change task (in the RAM).

Additional configurable functions of a parameter:

- Min./max. limits for input (block PLIM)
- Can only be changed in certain conditions and with specific access authorization levels (blocks PLIM, PSTAT)
- Multi-lingual parameter names can be assigned and selected (block PNAME)

6.9.2. Configuring a parameter

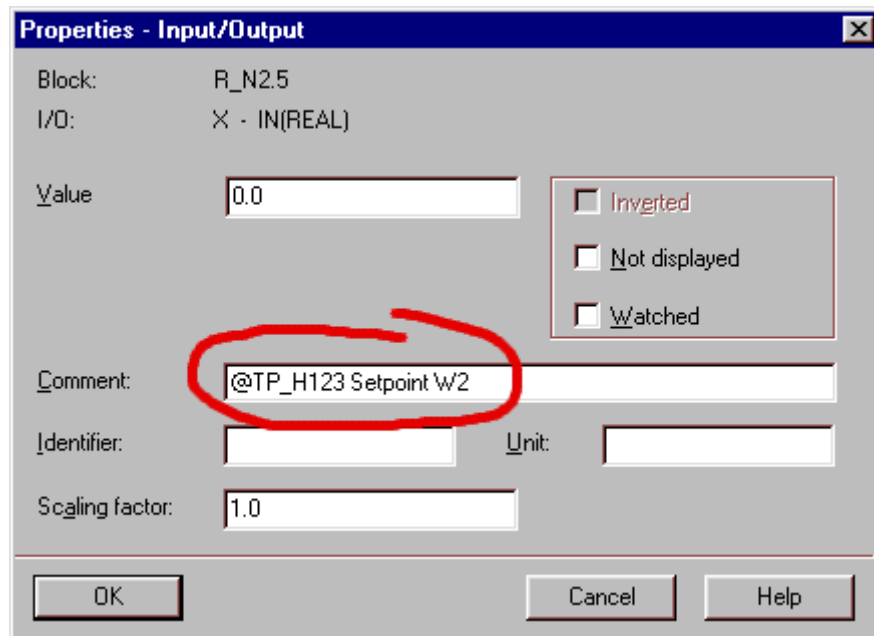
In order to access the connector of a function block using a parameter, the special character combination is configured in the comments line of the associated connector:

@TP_Kxyz with
 K: c, d, H, L as identification for the range
 xyz: 3-digit parameter number

Input (can be changed)	Output (cannot be changed)	Number range (in the parameter ident.)
H	d	1000-1999
L	c	3000-3999

Typical assignment of parameter numbers

In the H/d or L/c range, each parameter number (xyz) may only be assigned once! The code generator evaluates such a so-called “pseudo comment” which must be located at the beginning of the comments field. This can be followed by additional pseudo comments, separated by blanks and the normal comment.



Configuring a parameter with pseudo comment

Connector comment (pseudo comment)	Configurable connection	Connection	Display at the operator unit (PMU,OP1S) or SIMOVIS		Function
			T400 as tech. board (in the drive converter or SRT400, 2 nd slot)	T400 as baseboard (SRT400, 1 st slot) CPU module (in SIMADYN D subrack)	
@TP_Hxyz		without / OP connection (flag)	Hxyz	Pxyz	Change value
@TP_Lxyz			Lxyz	Uxyz	
@TP_Hxyz	Input	! (virtual connection)	Hxyz	Pxyz	Display value
@TP_Lxyz			Lxyz	Uxyz	
@TP_Hxyz	Output	Without / with	dxyz	rxyz	Display value
@TP_Lxyz			cxyz	nxyz	
@TP_dxyz	Input / output	Without / with	dxyz	rxyz	Display value
@TP_cxyz			cxyz	nxyz	
@TP_Hxyz, @TP_Lxyz	Input	@TC_nnnn	nnnn (number of the technology connector)	Display/change TC connection	
@TP_cxyz, @TP_dxyz				Not a practical function, as the parameter value cannot be changed (and therefore also the connection)!	
@TC_nnnn	Output	Without / with	Not possible		Source for connection via BiCo
	Input				Not permitted ! (compiler error message)

Configuring, display and function of parameters and technology connectors

Connection type SIMADYN D	BO Bool	I Integer and connection at @TC technology connector	DI Double Integer	W Word	R Real	TS SDTime
Parameter type	B	I2	I4	V2	I4	I4
Display on the operator panel	0/1	4-digit number	8-digit number	Hex, 4-digit	Decimal No., 3 decimal places	Decimal No., 3 decimal places

Converting the connector types into parameter types

Parameter value changes can either be temporarily saved (RAM, deleted at restart) or permanently saved (EEPROM). The PMU and OP1S operator panels always save the changes permanently. For SIMOVIS, the system can also be set so that changes are only made in the RAM.

6.9.3. Function blocks required

Note: Additional details on the function blocks and their connections can be taken from the online help (F1) or the User Manual.

@DRIVE Always required when using the T400 in the drive unit;
Sampling time: 100 ms up to 200 ms, typ. approx. 100 ms; the block is configured once

PTRANS Required, if parameters have to be transferred between communication module(s) (e.g. ADB with 2 communication modules) and the T400 as well as the CUx basic module;
sampling time: according to the required cycle time for parameter transfer; the block is configured once

Optional function blocks for supplementary functions:

PLIM for real connector types;
 for other types: PLIM_B (Bool), PLIM_I (INT, 16-bit), PLIM_D (DINT, 32-bit);
 The block refers to a parameter specified at a connector;
 a) Definition of min./max. setting limits
 b) Definition of the status, in which the parameter can be changed
 (defines the actual status with PSTAT)
 in the example: 0005: changes can be made in statuses 1 and 3
 c) Definition of a required access authorization level to change the parameter
 (defines the actual access authorization level with PSTAT)
 Sampling time: typ. 200 ms and higher (is only executed at initialization [initial load])

PSTAT a) Defines the actual status (in the example: 3) regarding the ability to change parameters
 b) Defines an access stage regarding how parameters can be changed by entering passwords for the various access stages (in the example: 17 to stage 4)

PNAME Configuring parameter names (max. 16 characters) for up to 20 parameters per block for plain text display on the OP1S operator panel or SIMOVIS. Input syntax:

Kxyz:Parameter-NameXY with
 K: c, d, H, L as identification for the range
 xyz: 3-digit parameter number

Multiple languages are configured using several PNAME. The configuring engineer assigns a common language ID to all PNAME of a language. A language is selected at @DRIVE by entering this language ID. After the next restart, the PNAME are activated with the appropriate ID.
 Sampling time: Typ. 200 ms and higher (is only executed at initialization)

The image shows four screenshots of SIMATIC Manager parameter blocks:

- Block 5 (PLIM):** Operator contr. (2/-) with parameters: minimum (1.5), maximum (50.0), allowed sta (16#5), User Level (4). Status disp is 15.
- Block 6 (PSTAT):** Change enable (3/-) with access stages: Password (0), Level1 (5), Level2 (0), Level3 (0), Level4 (17), Level5 (12), Level6 (0), Level7 (0), Level8 (0), Device Stat (16#3). User Level is 15, State is 3, Status disp is 15.
- Block 7 (PNAME - English):** english names (4/-) with parameter names: *H003:Fault LED*, *H127:source setp*, *H0128:source act*, *H0129:kp-factor*.
- Block 8 (PNAME - German):** german names (8/-) with parameter names: *H003:Störmeldung*, *H127:Sollwertque*, *H0128:Istwertque*, *H0129:kp-Faktor*.

Blocks for supplementary parameter functions (parameter names are in some cases not represented in their full length)

- Not for connector types, assigned using parameters and which can be connected with BiCo technology: GV, DW, BY
- No connections can be generated with BiCo at **non-connected inputs**. At the time that code is generated, the input must have a connection to a technology connector.
- A connection (at the input) **cannot be deleted** using BiCo.
- When **updating** a project with CFC (transferring the changes into the target system), only permanently (**EEPROM**) saved parameter changes are taken into account. It is possible to establish data consistency between the target system and configured application software on the PC.
- Making simultaneous changes with BiCo technology and with the **CFC test mode**:
Although it is possible to simultaneously make changes with both commissioning tools, this should be avoided, as this can result in inconsistent statuses and subsequent erroneous operation!
Connection changes using BiCo cannot, for example, be simultaneously updated in the CFC. Vice versa, when a connection is changed with CFC, the appropriate parameter for the BiCo connection still has the previous value.
- If the CFC test mode is called-up after changing a connection with BiCo, the "**Differing software**" warning is displayed. It is possible to continue, but error-free operation is no longer guaranteed! Thus, before starting the CFC test mode, the project should first be updated, compiled and then re-loaded onto the T400.
- In a chart, which is to be generated for a **function block type (typical)**, pseudo comments may neither be configured at the (inner) block I/O nor at the chart interface!
- When using **chart nesting**, pseudo comments may be configured at the (inner) block I/O, but not at the chart interface!
- Number of **newly parameterizable BiCo connections** (after a code generation):

Within a sampling time (task):	Unlimited
Between various sampling times:	<ul style="list-style-type: none">• min. 10 or• 20% of the connections between tasks, configured in the task, or• 100% of the technology connectors @TC_ configured in the task

6.10. Permanently saving values (non-volatile)

All of the data in the T400 are lost at reset/restart, and are re-calculated. If calculated data, such as level, diameter, position or similar are to be kept when the power supply is powered-down, the SAV_TR block should be used. Up to 10 real values can be saved in a non-volatile fashion.

7. Basic application software

The documentation of the basic software is available under **START | SIMATIC | S7 Manuals** after D7-SYS has been installed.

The configured application software itself is saved during the standard installation under the following path:

\\SIEMENS\STEP7\Examples\D7-T400-basic-prog.

Select **File | Open** and then **Browse** in order to open the basic configured application software. The above specified path should then be used there.

We recommend that a project is first copied before modifications are made to it.

Siemens AG
Automation & Drives
Postfach 3269, D-91050 Erlangen

