DIgSILENT

PowerFactory

Advanced Tutorial

Relay Modelling

INTEGRATED POWER SYSTEM ANALYSIS SOFTWARE FOR

TRANSMISSION / DISTRIBUTION / INDUSTRY / GENERATION / INTEGRATION OF RENEWABLES

Publisher:
DIgSILENT GmbH
Heinrich-Hertz-Straße 9
72810 Gomaringen / Germany
Tel.: +49 (0) 7072-9168-0
Fax: +49 (0) 7072-9168-88
info@digsilent.de

Please visit our homepage at:
http://www.digsilent.de

February 2017
Version: 1
Edition: 2

Contents

1 Introduction

This tutorial demonstrates the modelling and editing of relay protective devices. The tutorial has been designed for the user who has already used, and is familiar with the basic functions and structure of *PowerFactory* .

Network models have been prepared for use. The tutorial comes with two *PowerFactory* database (.pfd) files, that will be used for this tutorial.

2 Relay Modelling

There are two types of relays provided by *DIgSILENT* : generic and manufacturer specific. A list of all available relays and models can be downloaded from our download area.

Each relay model in *PowerFactory* is made out of different blocks, where each block performs specific function of the relay. Number of blocks implemented in a relay model depends on the relay complexity. But common for all relays is that they always contain measurement units (current and/or voltage transformers), measurement blocks and logic block.

In case that relay needs information on current in the system (overcurrent, differential ,distance relays) current transformer has to be build in relay model. In case that relay needs information on voltage (directional, distance, overvoltage, undervoltage relays) we will need to provide the model additionally with voltage transformer. Measurement block is using CT and VT to transform the measured values into discrete signals and finally logic block is responsible for sending the trip signal to the switching device.

For each relay type, a corresponding technical reference document is available. The technical references contain detailed descriptions on the blocks, their signals and functionality.

3 Introduction to a relay models

Our goal here is to get better overview over relay functionalities, there blocks and signals . Therefore we will create a simple relay from scratch. In the first step an simple overcurrent relay will be created and later this relay will be expend for additional overvoltage and undervoltage features.

3.1 Creating a simple overcurrent relay 50

You will learn how to create a simple overcurrent relay with two instantaneous overcurrent blocks (i.e. ANSI standard device number 50). This means for this task we will need:

- one current transformer block (CT),
- one measurement block for reading the current values,
- two overcurrent blocks for processing the measured current,
- one logic block that will send the trip signal to the circuit breaker in case of a fault.

If you are creating a new relay from scratch you will always have to consider following steps:

- Define all slots and their connections. In our case we need 5 slots. The number of slots depends on the complexity of the relay (e.g. for modern numerical distance relays there can be over 100 slots!)
- Define the relay type and refer it to the created frame. The relay type has to contain all block types for each slot. The block type contains important type information such as current ratios, time dial setting ranges, etc.
- Use and test the relay to see how your relay reacts.

3.2 Defining Relay Frame

- Import and activate the project called RelayModelling.pfd.
- Start by inserting a new page in the graphic board. For this select the *Insert New Graphic* option from the local graphic window icon bar .
- In the popup window select the option *Block/Frame Diagram*.
- Place 5 slots on the empty Frame Diagram by selecting the from the Draw toolbox. Each slot will represent one of the blocks that our relay will contain.

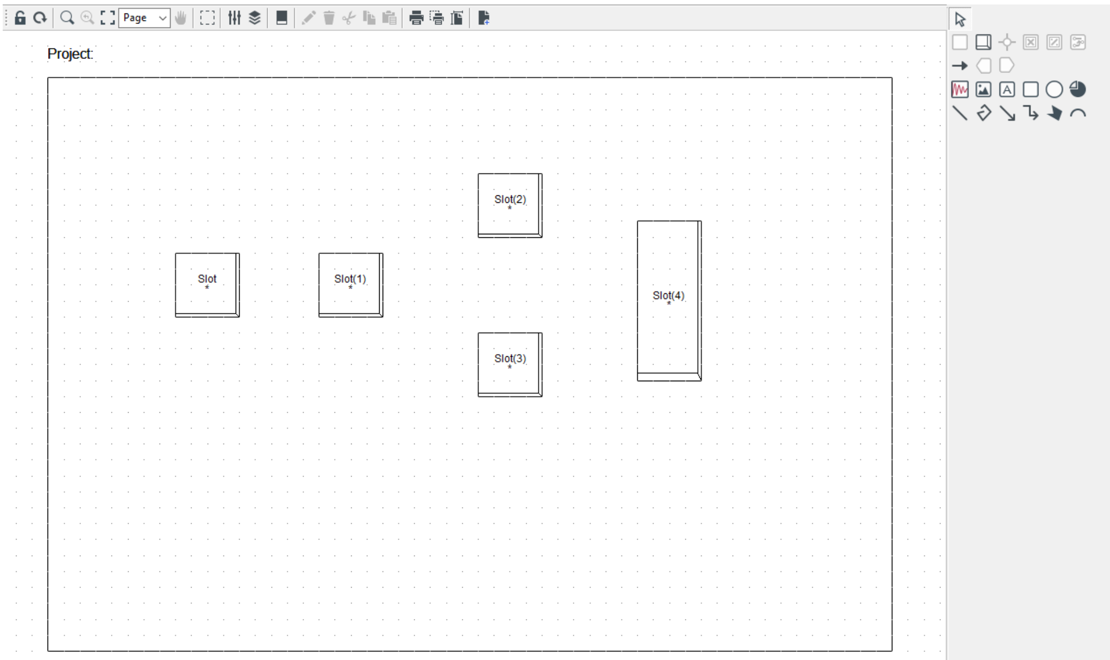


Figure 3.1: Placing Slots

- Next step is to select type and input/output signals of the slots.
  - **First Slot** name it Ct. This slot will represent the used current transformer. For *PowerFactory* to know that this will be CT write *StaCt** under Class Name option inside of the slot. The next important part is to define the input and output signals. For the Ct-block we do not need any input signals so this field will stay empty. For the output signals write I2r_A;I2i_A,I2r_B;I2i_B,I2r_C;I2i_C. This stands for real and imaginary parts of the CT's secondary phase currents. The list of all possible input and output signals are available in the technical reference for the current transformer document (TechRef_Ct). We can use this list to specify the signals we need for our relays.

Output Signal	Description	Unit
I2r_A	Phase A secondary side current real part	Sec Amps
I2r_B	Phase B secondary side current real part	Sec Amps
I2r_C	Phase C secondary side current real part	Sec Amps
I2i_A	Phase A secondary side current imaginary part	Sec Amps
I2i_B	Phase B secondary side current imaginary part	Sec Amps
I2i_C	Phase C secondary side current imaginary part	Sec Amps
I0x3r	Zero sequence secondary side current real part (calculated internally)	Sec Amps
I0x3i	Zero sequence secondary side current imaginary part (calculated internally)	Sec Amps

Figure 3.2: List of all output signals that CT can have

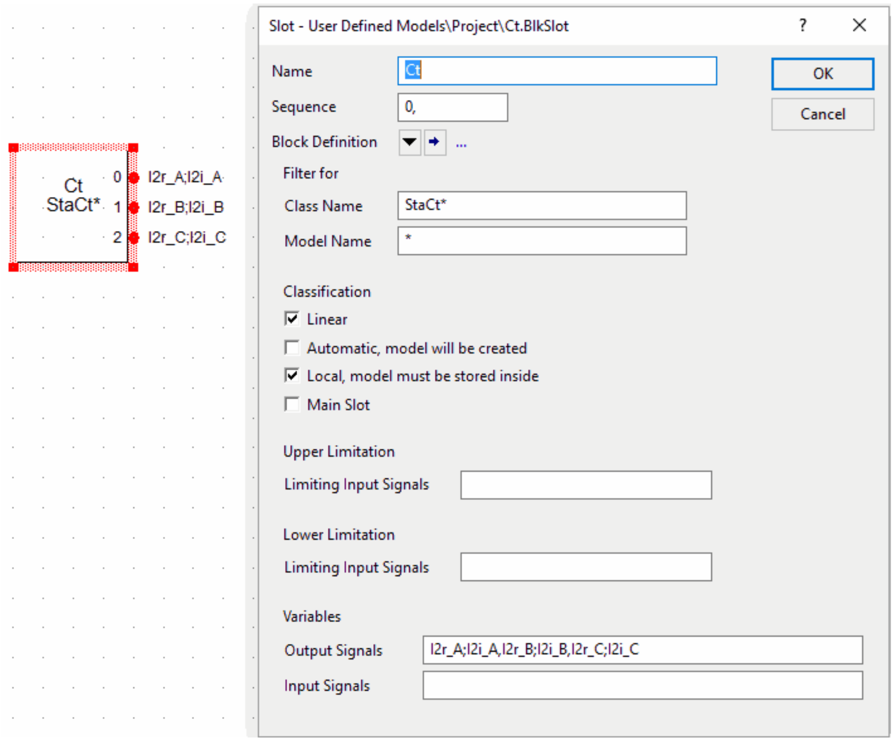


Figure 3.3: Ct.BlkSlot Block

– **Second Slot** name it Measure. Input of this block should be the output of Ct block. In technical documentation for this block you will find the name of input signals for real/imaginary part of the phase input current (wIr_A;wIi_A,wIr_B;wIi_B,wIr_C;wIi_C). This blocks provides RMS value of the phase current as output that will be send to be compared with the pick up current in 50/51 block (I_A;I_B;I_C). Class Name is *RelMeasure**.

Input Signal	Description	Unit
wIr_X	Input Current phase X	Amps
Wur_X	Input Voltage phase X	Volt

Output Signal	Description	Unit
I_x	Current phase X RMS value	Amps
Ir_X	Current phase X phasor real part (with Fourier filtering enabled only)	Amps
Ii_X	Current phase X phasor imaginary part (with Fourier filtering enabled only)	Amps
U_x	Voltage phase X RMS value	Volt
Ur_X	Voltage phase X phasor real part (with Fourier filtering enabled only)	Volt
Ui_X	Voltage phase X phasor imaginary part (with Fourier filtering enabled only)	Volt

Figure 3.4: List of input and output signals of the RelMeasure block

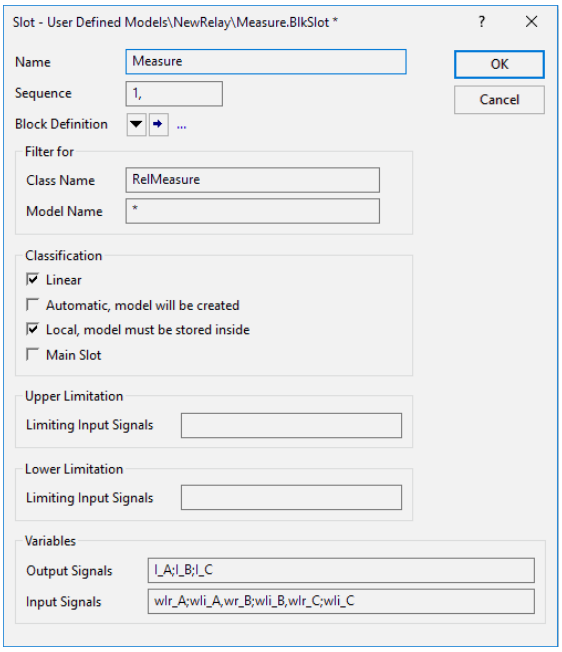


Figure 3.5: RelMeasure block

– **Third Slot** will be our $I >$, first instantaneous overcurrent block (50). It's type *Relloc** is also predefine in *PowerFactory* . Write *Relloc** under Class Name option. This block expects phase currents (Iabs_A;Iabs_B;Iabs_C) and provides us with two signals yout and y_s. For our task here is only yout tripping signal of interest.

Input Signal	Description	Unit
Iabs_A	Phase A input current	Sec Amps
Iabs_B	Phase B input current	Sec Amps
Iabs_C	Phase C input current	Sec Amps
iblock	Blocking signal	
iblock_A	Phase A Blocking signal	Sec(or 1/0 RMS/EMT simulation)
iblock_B	Phase B Blocking signal	Sec(or 1/0 RMS/EMT simulation)
iblock_C	Phase C Blocking signal	Sec(or 1/0 RMS/EMT simulation)
wfwd_A	Phase A forward current signal	Sec(or 1/0 RMS/EMT simulation)
wfwd_B	Phase B forward current signal	Sec(or 1/0 RMS/EMT simulation)
wfwd_C	Phase C forward current signal	Sec(or 1/0 RMS/EMT simulation)
wrev_A	Phase A reverse current signal	Sec(or 1/0 RMS/EMT simulation)
wrev_B	Phase B reverse current signal	Sec(or 1/0 RMS/EMT simulation)
wrev_C	Phase C reverse current signal	Sec(or 1/0 RMS/EMT simulation)
wstart_A	Phase A external starting signal (available only if iextstart = 1 in TypIoc)	Sec(or 1/0 RMS/EMT simulation)
wstart_B	Phase B external starting signal (available only if iextstart = 1 in TypIoc)	Sec(or 1/0 RMS/EMT simulation)
wstart_C	Phase C external starting signal (available only if iextstart = 1 in TypIoc)	Sec(or 1/0 RMS/EMT simulation)

Output Signal	Description	Unit
yout	Trip signal (any phase)	Sec(or 1/0 RMS/EMT simulation)
yout_A	Phase A trip signal	Sec(or 1/0 RMS/EMT simulation)
yout_B	Phase B trip signal	Sec(or 1/0 RMS/EMT simulation)
yout_C	Phase C trip signal	Sec(or 1/0 RMS/EMT simulation)
yallout	All phases trip signal	
y_s	Start signal (any phase)	Sec(or 1/0 RMS/EMT simulation)
y_A	Phase A start signal	Sec(or 1/0 RMS/EMT simulation)
y_B	Phase B start signal	Sec(or 1/0 RMS/EMT simulation)
y_C	Phase C start signal	Sec(or 1/0 RMS/EMT simulation)

Figure 3.6: Input/Output of RelIoc block

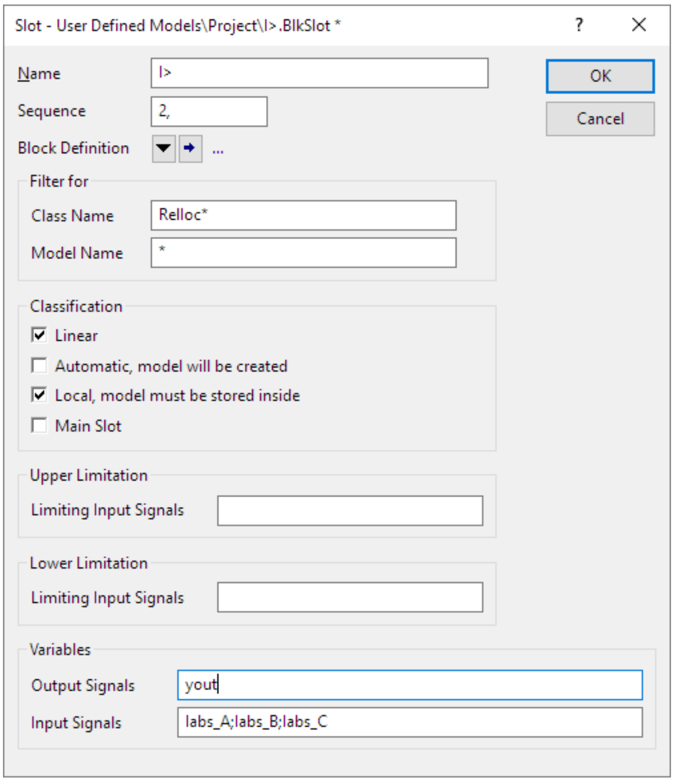


Figure 3.7: RelMeasure block

- **Fourth Slot** $I >>$. Class Name *Relloc** input and output are the same as for the $I >$.
- **Fifth Slot** Logic. The last block Logic takes the outputs from the $I >$ and $I >>$ slots and sends on the switching device signal to open if one of this two blocks sees current bigger than the pick up current.

Block Name	Block Class Name	Input Signals	Output Signals
Ct	StaCt	-	I2r_A;I2i_A,I2r_B;I2i_-B,I2r_C;I2i_C
Measure	RelMeasure	wIr_A;wIi_A,wIr_B;wIi_-B,wIr_C;wIi_C	I_A;I_B;I_C
$I >$	Relloc	Iabs_A;Iabs_B;Iabs_C	yout
$I >>$	Relloc	Iabs_A;Iabs_B;Iabs_C	yout
Logic	RelLogdip	y1,y2	yout

Table 3.1: Summery of the slots and there options

You have maybe noticed that some of the signals are separated with comma " , " and some of them are separated with semicolon " ; ". For example I2r_A;I2i_A,I2r_B;I2i_B,I2r_C;I2i_C this will create three contacts on the slot each carrying two signals (first contact I2r_A;I2i_A, second I2r_B;I2i_B and last third contact carries two last signals I2r_C;I2i_C). On this way we are avoiding to have too much lines on our slot diagram.

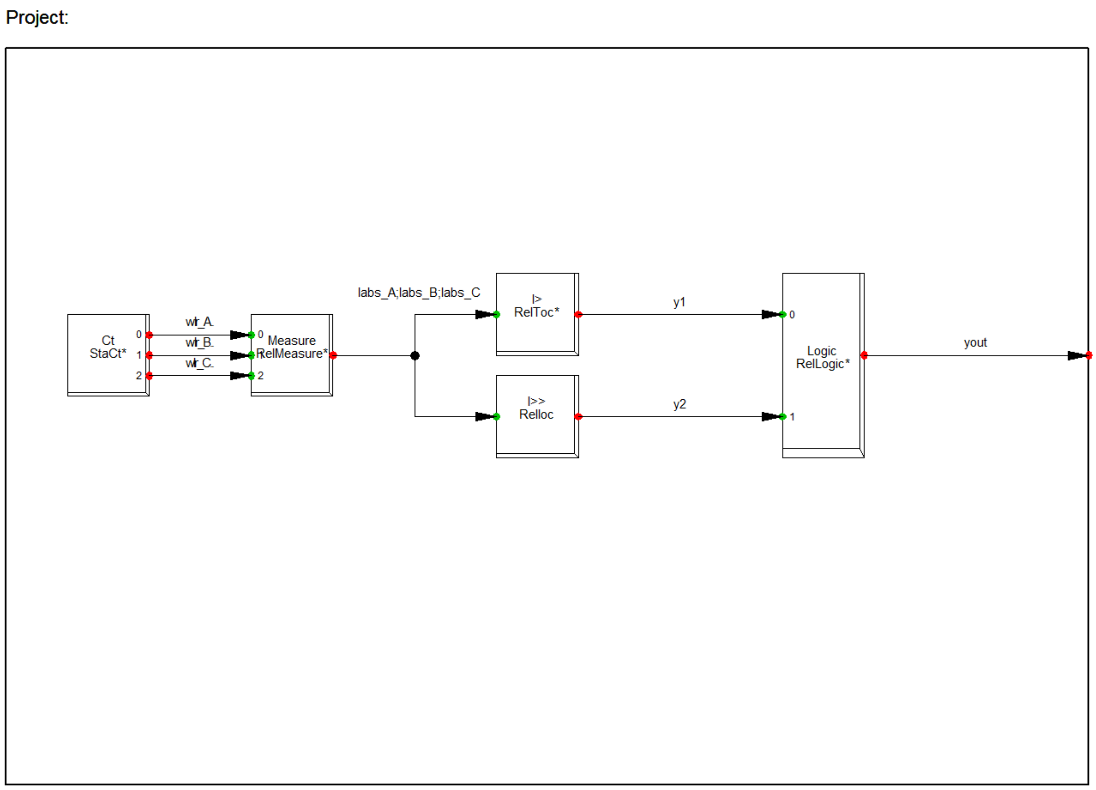


Figure 3.8: Relay Frame

- Connect plots as shown in the picture above. To connect slots use the Signal element from the Draw toolbox and select two contacts this signal is connecting.

3.3 Create Relay Type

Here we will define new Relay type that refers to the Relay frame we just created and we will be defining each relay block type closer.

- Create new object of the Type *TypRelay* inside of the *Equipment Type Library* Folder and name it “OCRelay”.

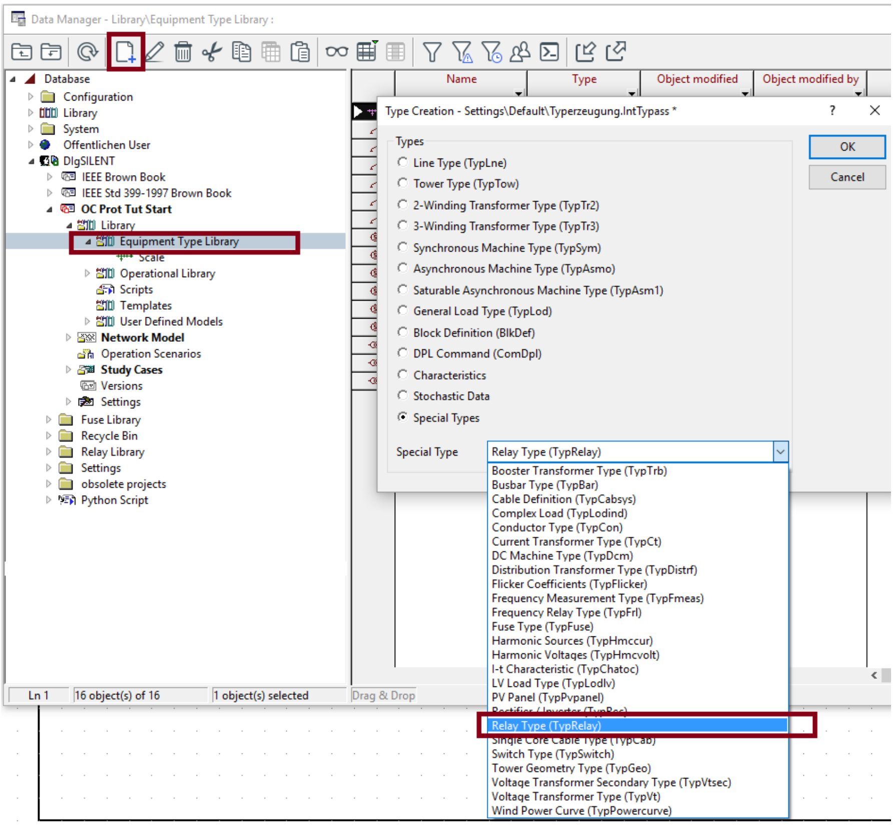


Figure 3.9: Creating new Relay type

- From the *User Defined Models* Folder (also inside of the *Library* Folder) cut the fame you created in previous sub-chapter and past it inside of the new Relay Type.
- Open the new Relay Type “OCRelay” and select the frame under *Relay Definition* option.

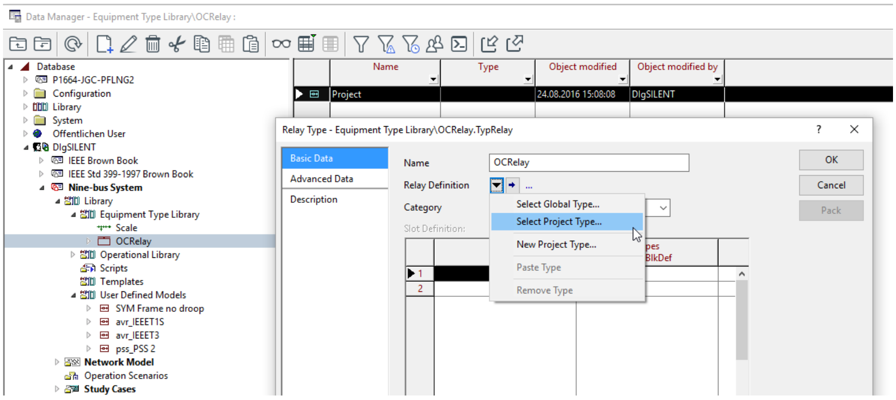


Figure 3.10: Selecting the Relay Definition

- Relay will automatically update showing all slots. Confirm with OK.
- New we will define for each slot one relay block type. To define new object inside of the “OCRelay” select OCRelay in Data Manager, press from the local icon bar and select the Special Type. The easiest way to select special type is by writing its name in selecting field.
  Define:

– one block of the type **TypMeasure** - *Type*: 3ph. Currents/Delta Voltages
  Nominal Current:1;5 A
  Nominal Voltage: 110 V
– two blocks of the type **Typloc** - *Current Range*: 0.1-100;0.01
  Time Setting Range:0.1-100;0.01
– one block of the type **TypLogdip** - Under *Basic Data* Page: *Input Signal*: y1,y2
  Tripping Signal: yout.
  Unted *Logic* Page : yout = y1.or.y2

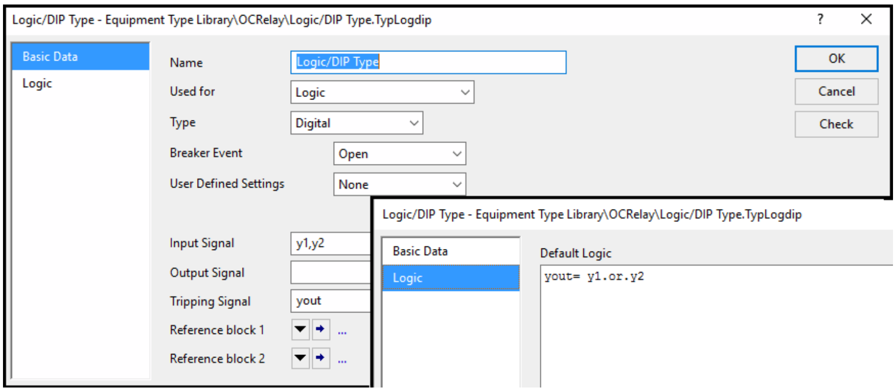


Figure 3.11: Relay Logic Block

• Open the Relay type object once again and press Slot update.

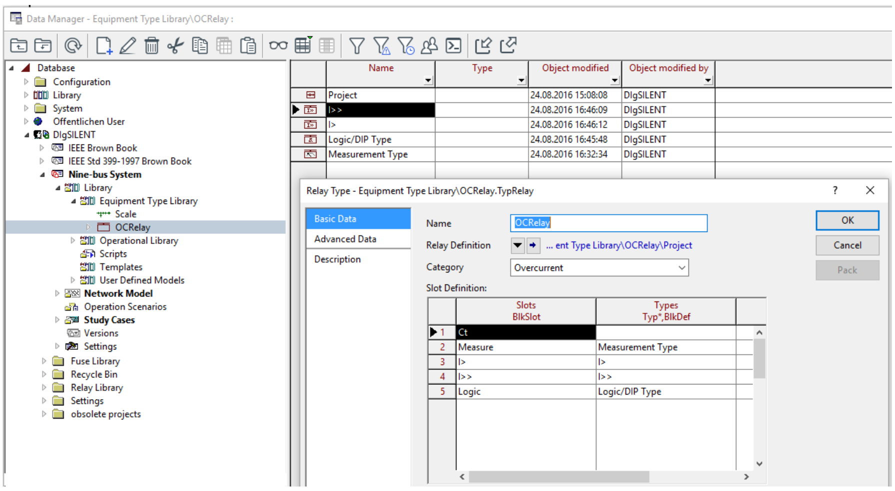


Figure 3.12: Relay Type

• If you would like to replace some of the blocks just right mouse button click on the block to be replaced and select *Reset Element/Type* option. Fill now emptied field with the type you would like to have here.

3.4 Testing

Let us now test our new created Overcurrent relay.

- Create the Relay on the *Motor Cable*. Refer to User Manual for how to create relay if needed
- Create a Current Transformer 1000/1.
- Select 0,72 pu for pickup current and 2,5 sec for time settings in $I >$ block and 8,1 pu and 0 sec in $I >>$ Block.
- Create Time-Overcurrent Plot and add the Motor characteristic on the plot as well.
- Run *LoadFlow* calculation. Does the relay trip? Why not?
- Run the 3 phase max short circuit on the end of the protected cable. Does the relay trip?

Let us have a look inside of the relay to see how this short circuit current is processed in the relay it self.

- Open the Relay in the Data Manager.

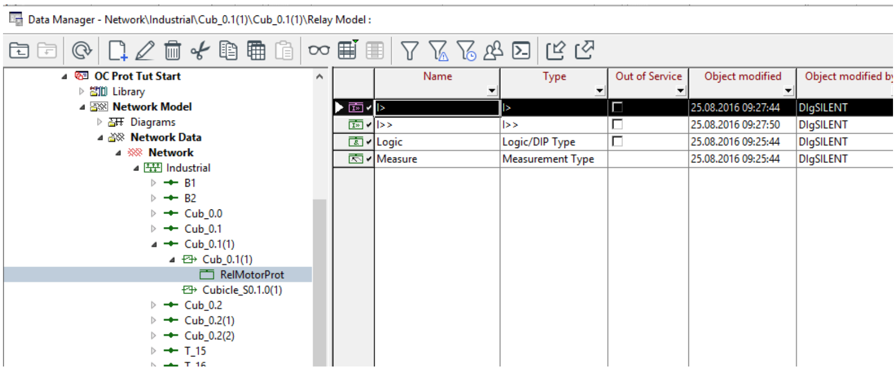


Figure 3.13: Relay in Data Manager

- Let us have a closer look at the signals of the Measure Block.
  - For this select the Measure block and activate the *Detail Mode* by selecting the from the local icon bar.
  - Switch to the *Flexible Data* Page and open *Variable selection* window.
  - Select *Signals* from the drop down menu under option *Variable Set* in the new window. Choose to see values of all input and output signals.
  - How high are the I_A,I_B,I_C Signals? Take the CT into account and check if the Relay is seeing the correct value of the current.

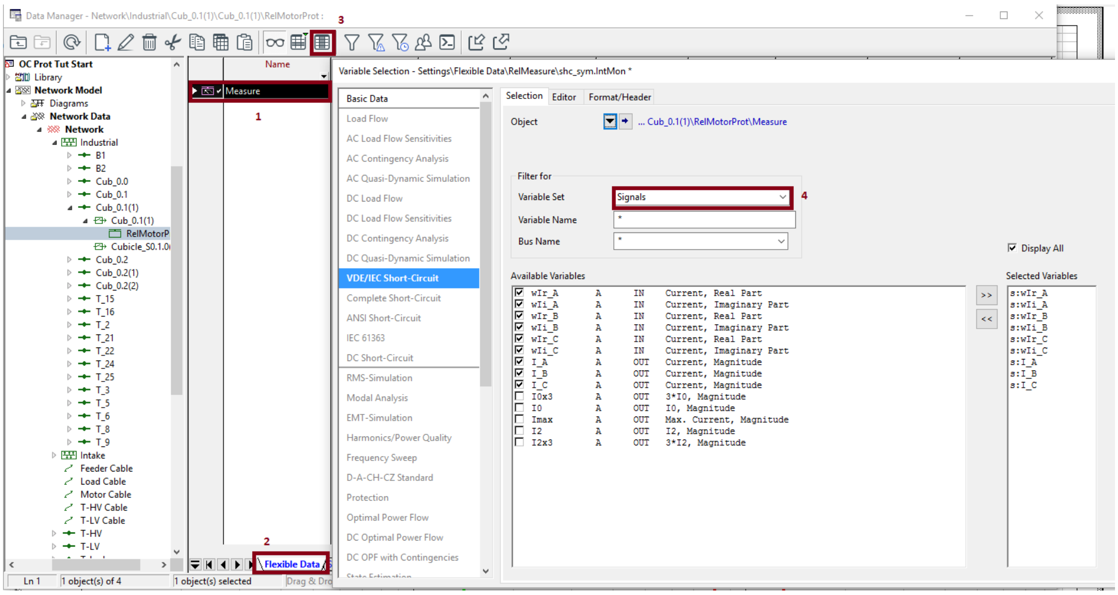


Figure 3.14: Selecting Signals for Measure block

- Let us now have a closer look at the $I >$ and $I >>$ blocks. First deactivate *Detail Mode* and repeat the steps from above for $I >$. Select the Iabs_A,Iabs_B, Iabs_C and yout from the *Signals* and pickup current Ipsetr from the *Calculation Parameter* variable set.
- Are the values of this signals as expected?

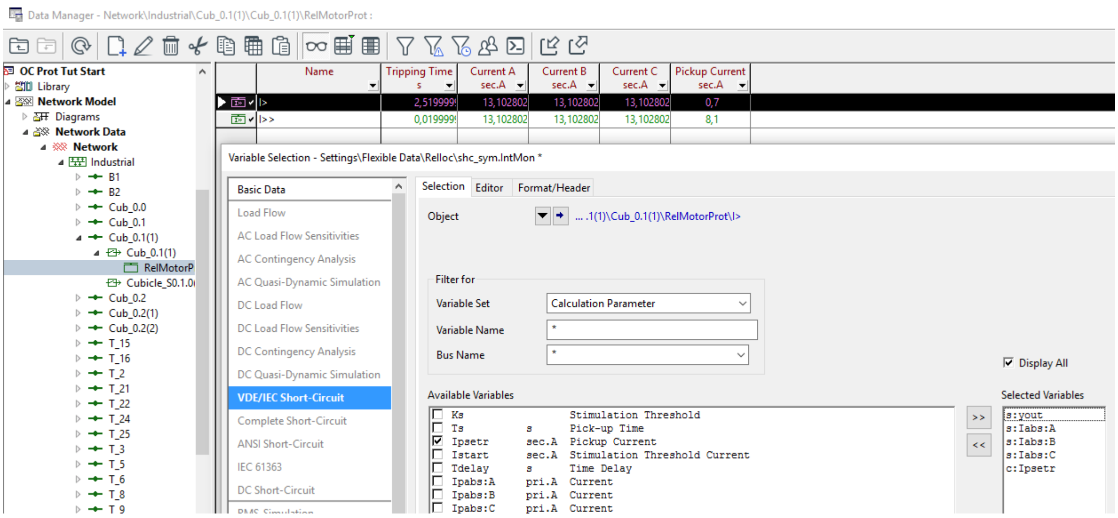


Figure 3.15: $I >$ and $I >>$ Signals

- And finally let us look at the logic. Which signal is responsible for tripping the relay.

3.5 Expanding OC Relay

Here we will expand our new created relay model by implementing new protection criteria. This simple relay will contain two instantaneous over-current blocks, one over-voltage and one under-voltage block. You will repeat same steps from above and add new blocks in the frame, add new block types in the relay model,. . . For this task one $U >$ (59) and one $U <$ (27) block will be added. We will need not just current transformer (CT) but also an voltage transformer (VT). Measurement block and also logic have to be adopted to the new changes.

3.5.1 Changing Relay Frame

- Place 3 new slots on the Frame Diagram by selecting the ☐ from the Draw toolbox. Each of them will represent a new block.
- Next step is to select type and input/output signals of the new slots.
- **First new Slot** should be named Vt. This slot will represent used voltage transformer. For *PowerFactory* to know that this slot represents VT write *StaVt** under option ClassName inside of the slot. For Vt-block as also for the Ct no input signals are needed, so this field will stay empty. For output signals following signals are to be selected (U2r_A;U2i_A,U2r_B;U2i_B,U2r_C;U2i_C).

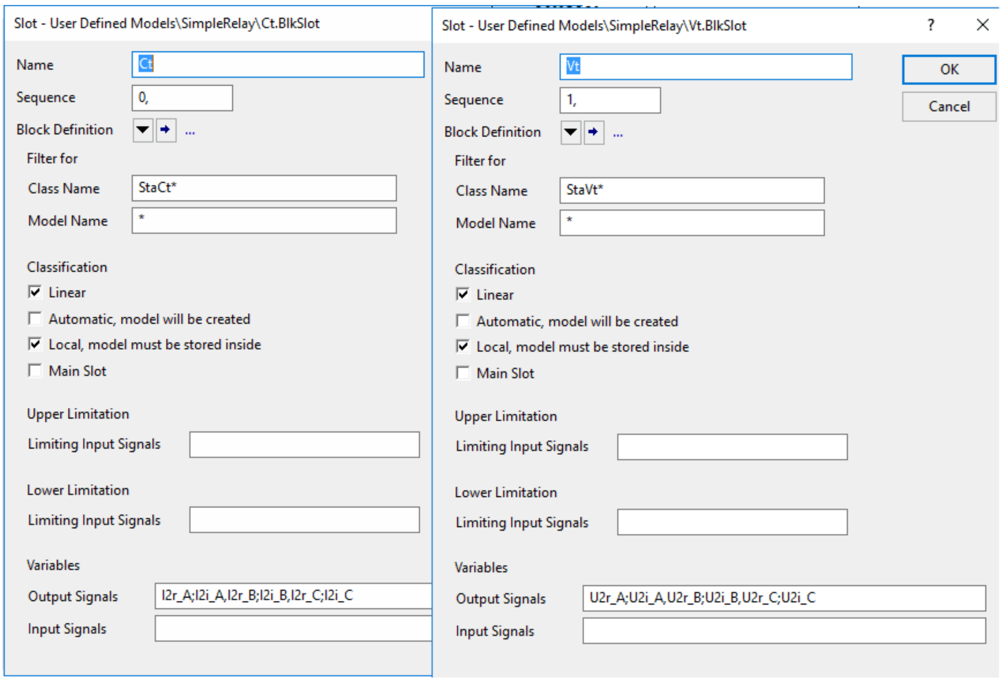


Figure 3.16: Ct and Vt Slots

- **Next two new Slots** $U >$ and $U <$. Both slots are having equal types RelUlim and are expecting the same signals, voltage for the phase a,b and c. Definition of how this signals are treated is defined later in block type.

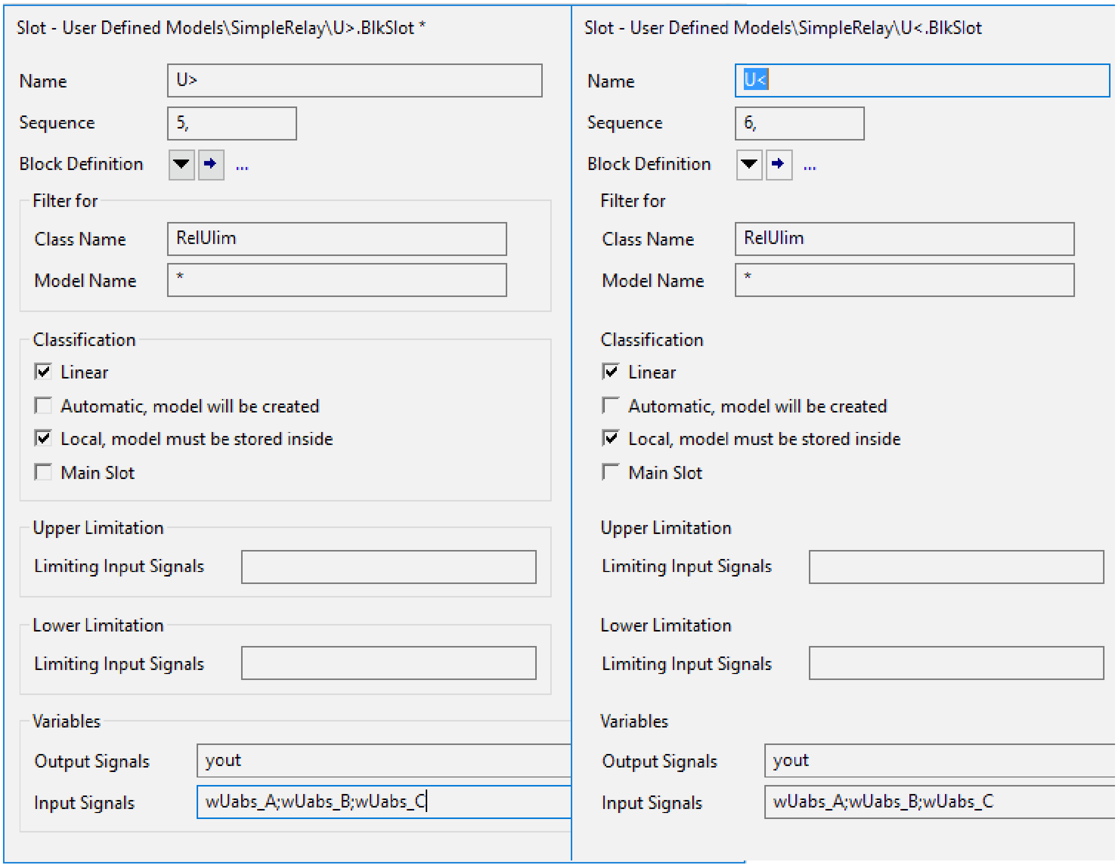


Figure 3.17: Window for over-voltage and under-voltage protection block

- Make corresponding changes in Measurement block frame and in Logic block frame.
  - Measurement should be able to receive information from VT and to provide output for $U >$ and $U <$.
  - Logic should support additional two input signals.

Block Name	Class Name	Input Signals	Output Signals
Ct	StaCt	-	I2r_A;I2i_A,I2r_B;I2i_B,I2r_C;I2i_C
Vt	StaVt	-	U2r_A;U2i_A,U2r_B;U2i_B,U2r_C;U2i_C
Measure	RelMeasure	wIr_A;wIi_A,wIr_B;wIi_B,wIr_C;wIi_C,wUr_A;wUi_A,wUr_B;wUi_B,wUr_C;wUi_C	I_A;I_B;I_C,U_A;U_B;U_C
$I >$	RelIoc	Iabs_A;Iabs_B;Iabs_C	yout
$I >>$	RelIoc	Iabs_A;Iabs_B;Iabs_C	yout
$U >$	RelUlim	wUabs_A;wUabs_B;wUabs_C	yout
$U <$	RelUlim	wUabs_A;wUabs_B;wUabs_C	yout
Logic	RelLogdip	y1,y2,y3,y4	yout

Table 3.2: Summery of the slots and there options

SimpleRelay:

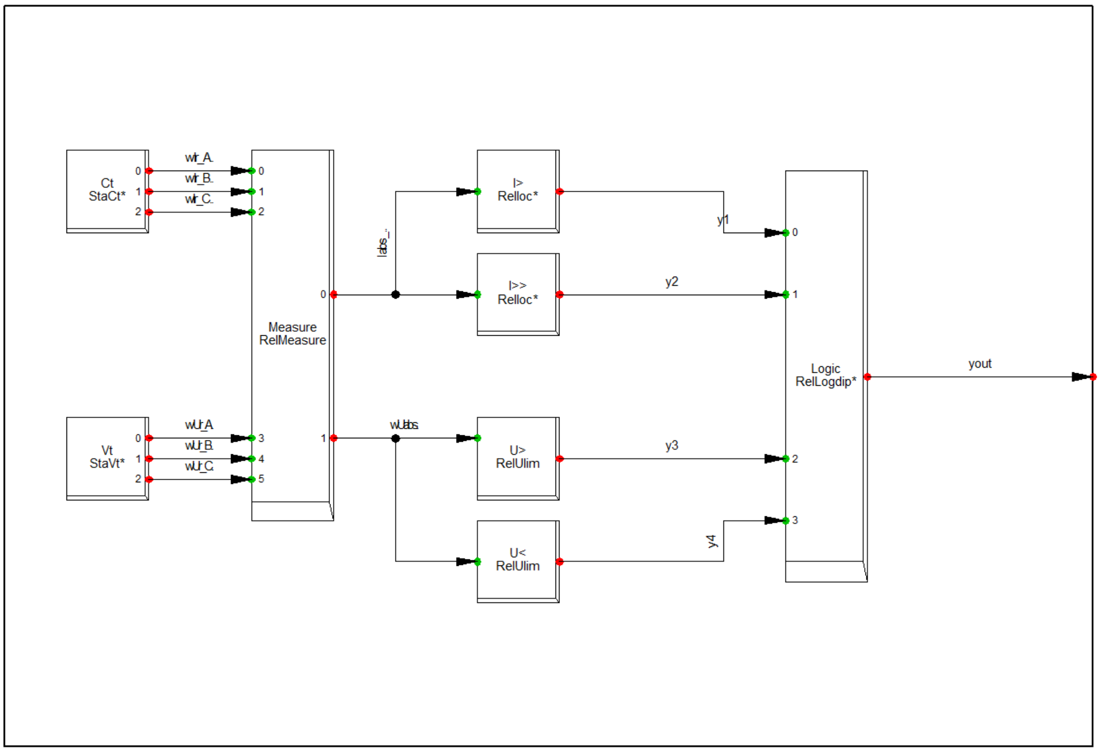


Figure 3.18: Relay Frame

- Connect plots as shown in the picture above. To connect slots use the Signal element from the Draw toolbox and select two contacts this signal is connecting.

3.5.2 Changing Relay Type

- Open the Relay type we created “OCRelay”.
- Press *Slot Update* button, new blocks will appear. Elements will be placed corresponding to there *Sequence* number written inside of the slot. If needed adjust the sequence number.
- Now for each new slot one relay block type will be defined and old blocks will be adjusted if needed. To define new object inside of the “OCRelay” just select OCRelay in DataManager, press from the local icon bar and select the Special Type. The easiest way to select special type is by writing its name in selecting field.
  Define:
  - two blocks of the type **TypUlim** - *Function*: Undervoltage/Overvoltage
    Voltage: 0.5-1.5;0.1
    Time Delay:0.1-100;0.01
    Reset Ration: 105%/95%
  - one block of the type **TypLogdip** - Under *Basic Data* Page: *Input Signal*: y1,y2,y3,y4
    Tripping Signal: yout.
    Unted *Logic* Page : yout = y1.or.y2.or.y3.or.y4

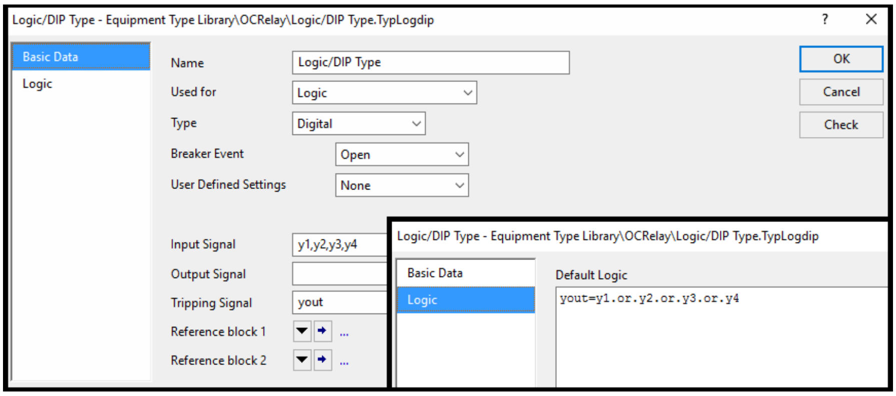


Figure 3.19: Relay Logic Block

- Open the Relay type object once again and press Slot update.

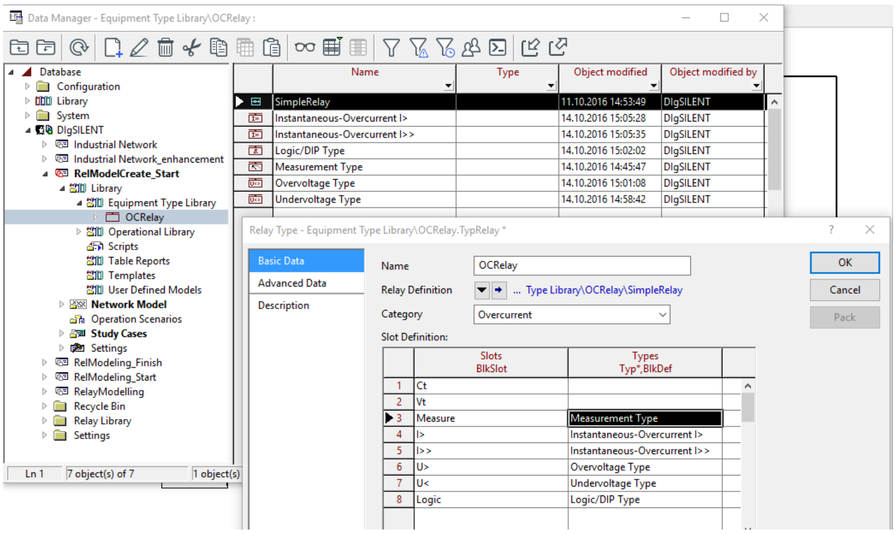


Figure 3.20: Relay Type

- If you would like to replace some of the blocks just right mouse button click on the block to be replaced and select *Reset Element/Type* option. Fill now emptied field with the type you would like to have here.

3.5.3 Testing new features

- Open the Relay build on the Motor Cable and press *Slot Update*.
- Create new VT 415/110 V.
- Set $U>$ to trip in 1 sec if voltage goes over 1.1 p.u and $U<$ to trip in 1 sec if voltage drops under 0.9 p.u.
- Change the Transformer type so that it supports tap positioning from +-20. Execute the load flow calculation and change the tap position of the transformer so that voltage goes under 0.9 and over 1.1 p.u.
- Check how relay reacts.

File Name	Description
RelayModelling_Start.pfd	File used to do the exercise
RelayModelling_FinishedOCRelay.pfd	Solution of exercise
RelayModelling_FinishedOCRelayExtended.pfd	Solution of exercise

4 Modelling of circuit breaker

Our task here is to implement a ABB PR222DS Circuit Breaker. This can be done as the task described in chapter 3.1 but we will engage a different approach. We will use a similar relay model that exists and change it.

- For this import Siemens "Sentron" Circuit Breaker model in *PowerFactory* . You will find this model in our Relay Library under Siemens/Sentron-ETUxxB.
- Copy Sentron ETU76B Relay model into our Project under *Equipment Type Library*.
- Rename it to be ABB_PR222DS.
- Delete the Default relay from the contents of the relay type.
- Open the Frame graphic to see the internal connections of the relay slots. For this *right mouse button click on the relay type* → *Show Graphic*

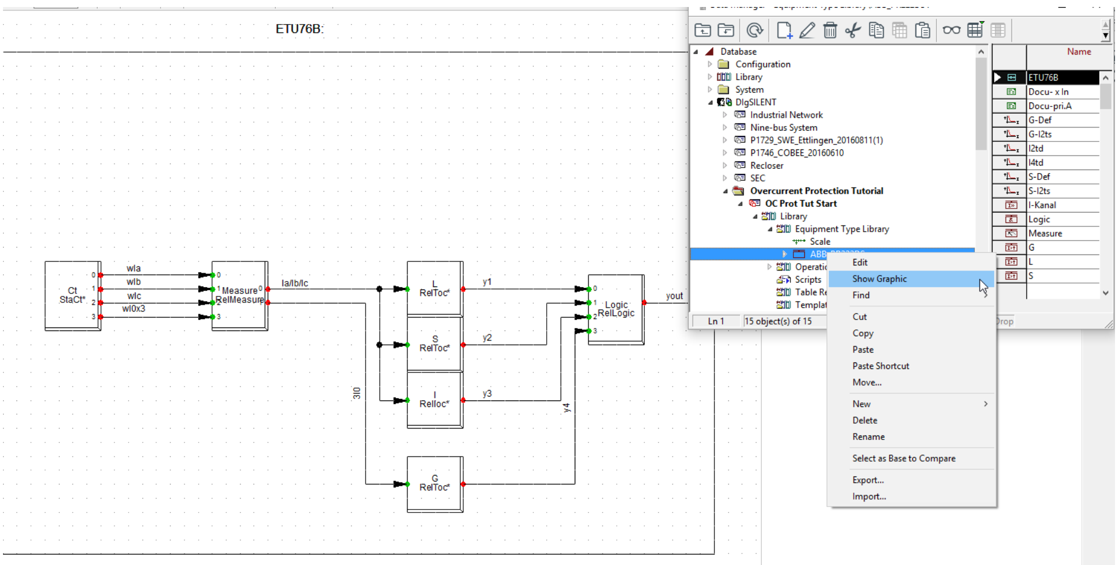


Figure 4.1: Frame of the ETU76B Circuit Breaker

- Get familiar with the relay frame. What is a difference between Measure block here and the one from the Overcurrent relay we created in previous chapter? (Additional 3I0 Signal that is separated and fed to block G RelToc)
- Following screen shots are made from ABB document DOC. N.° 1SDH000549R0001 - L7947 and represent the setting ranges for all available blocks. Our task is now to adjust the ETU76B to be our PR222DS
- First let us adjust In(A) nominal current values for this circuit breaker and write this values inside of the Measure.TypMeasure (*In* 100;150;250;300;400;600;800)

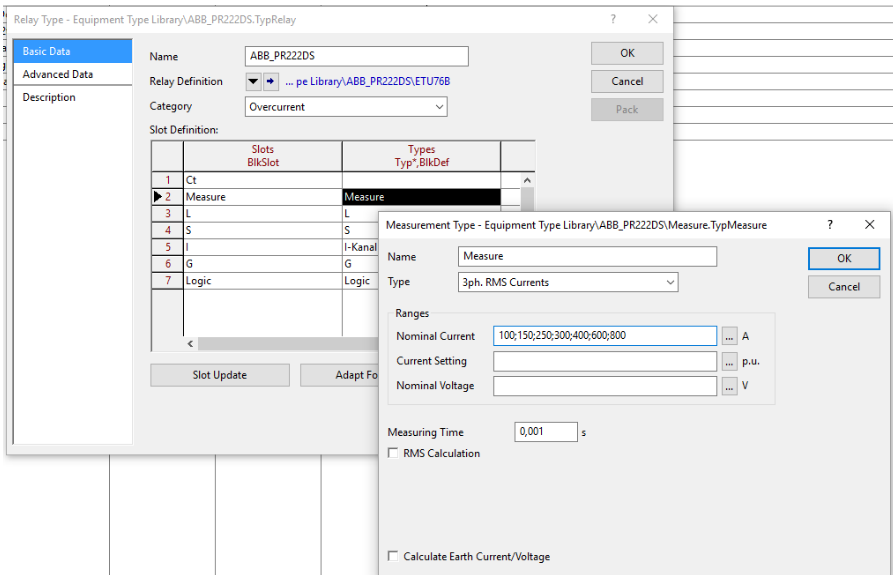
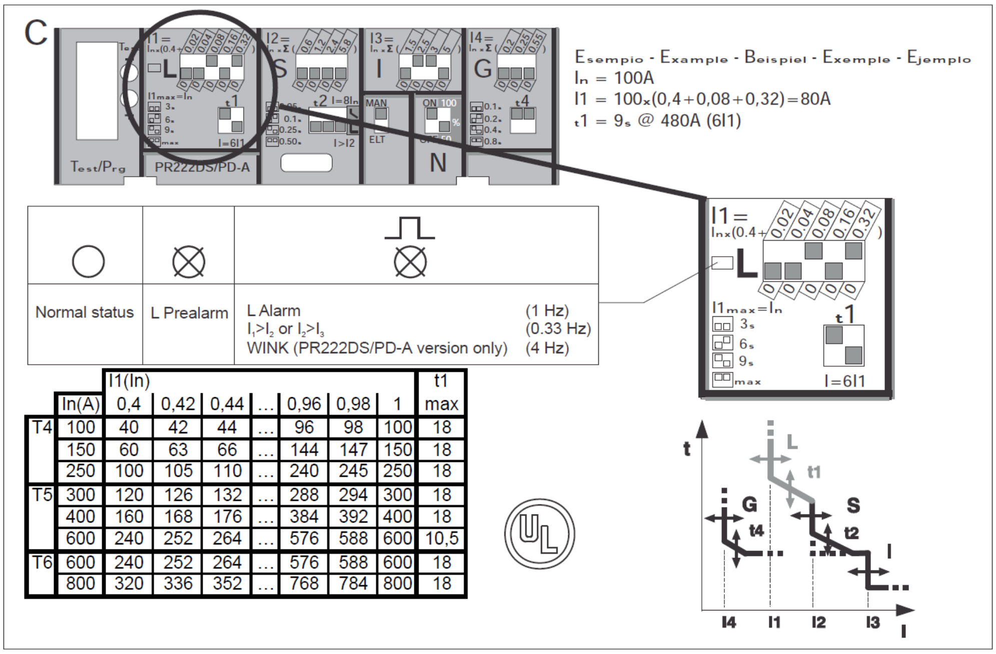


Figure 4.2: Measure Block

- From the picture below we can see that the current range for the Block L is from 0.4-1 with step of 0.02. Regarding time characteristic: this relay has 4 possibilities:
  - 3 sec 6In
  - 6 sec 6In
  - 9 sec 6In
  - tmax sec 6In (for all types except for T5 600A is tmax=18 sec)



	In(A)	I1(In) 0,4	0,42	0,44	...	0,96	0,98	1	t1 max
T4	100	40	42	44	...	96	98	100	18
	150	60	63	66	...	144	147	150	18
	250	100	105	110	...	240	245	250	18
T5	300	120	126	132	...	288	294	300	18
	400	160	168	176	...	384	392	400	18
	600	240	252	264	...	576	588	600	10,5
T6	600	240	252	264	...	576	588	600	18
	800	320	336	352	...	768	784	800	18

Figure 4.3: Block L

- Let us make changes in L block of our relay.

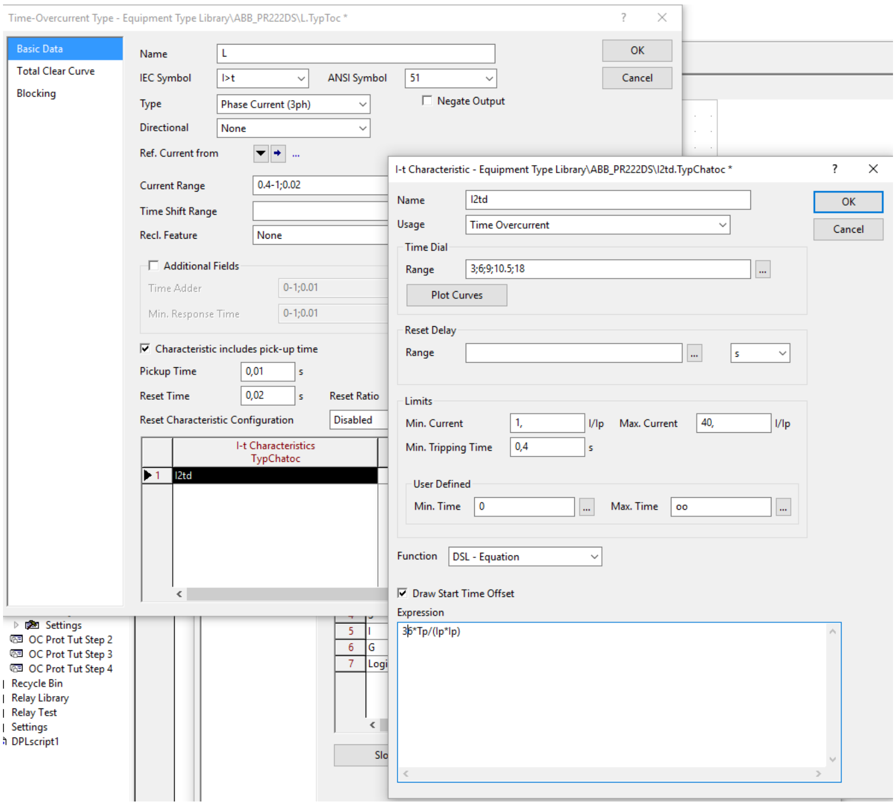
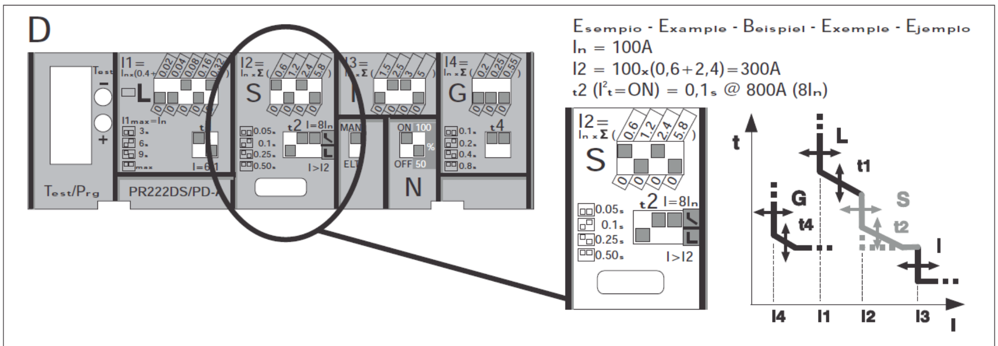


Figure 4.4: Block L

- Block L in Siemens circuit breaker ETU76B has two I-t Characteristic i^2t and i^4t. For ABB PR222DS we will just need one i^2t. So delete the one we don't need and adjust the I2td.



		I2(In)														
	In (A)	0,6	1,2	1,8	2,4	3	3,6	4,2	5,8	6,4	7	7,6	8,2	8,8	9,4	10
T4	100	60	120	180	240	300	360	420	580	640	700	760	820	880	940	1000
	150	90	180	270	360	450	540	630	870	960	1050	1140	1230	1320	1410	1500
	250	150	300	450	600	750	900	1050	1450	1600	1750	1900	2050	2200	2350	2500
T5	300	180	360	540	720	900	1080	1260	1740	1920	2100	2280	2460	2640	2820	3000
	400	240	480	720	960	1200	1440	1680	2320	2560	2800	3040	3280	3520	3760	4000
	600	360	720	1080	1440	1800	2160	2520	3480	3840	4200	4560	4920	5280	5640	5700*
T6	600	360	720	1080	1440	1800	2160	2520	3480	3840	4200	4560	4920	5280	5640	6000
	800	480	960	1440	1920	2400	2880	3360	4640	5120	5600	6080	6560	7040	7520	8000

* $I2_{max}$ T5 600= 9.5 I_n

Figure 4.5: S Block

- On the similar way adopt the S Block.

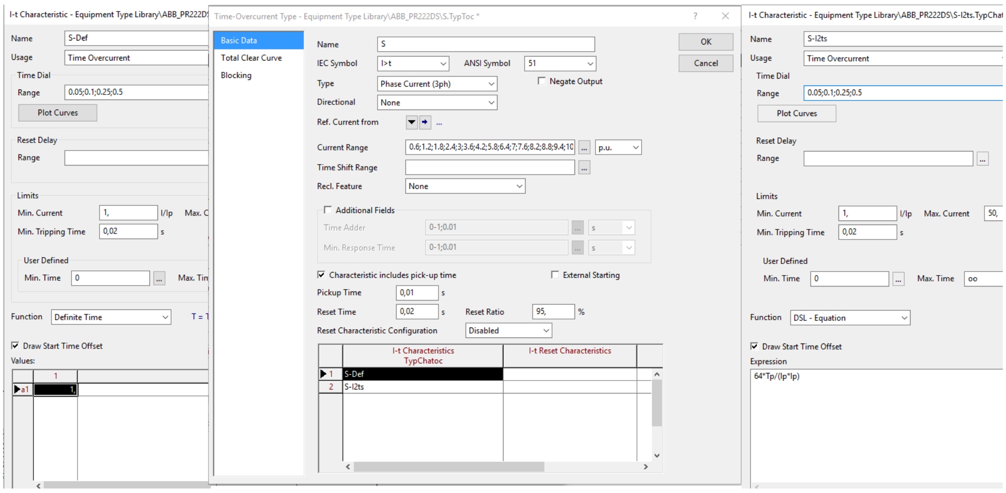
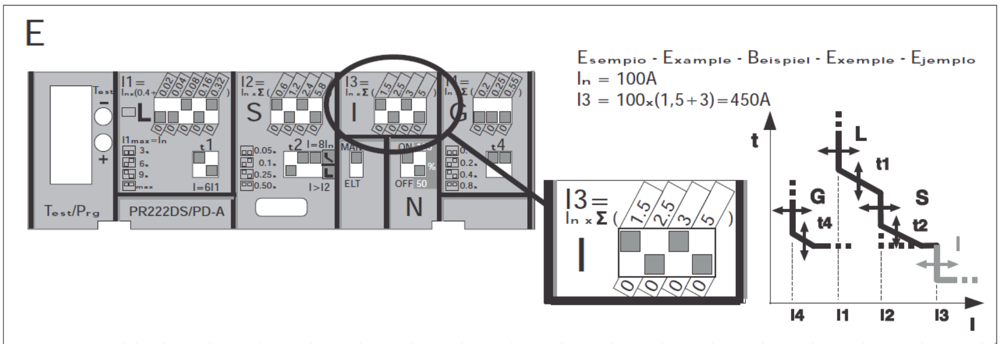


Figure 4.6: S Block

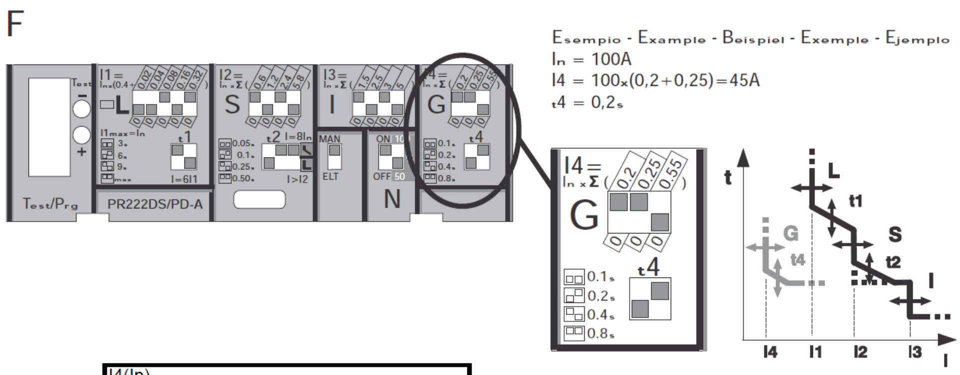
- I block is instantaneous so it is much easier to adjust.



	I3(In)															
	In(A)	1,5	2,5	3	4	4,5	5	5,5	6,5	7	7,5	8	9	9,5	10,5	12
T4	100	150	250	300	400	450	500	550	650	700	750	800	900	950	1050	1200
	150	225	375	450	600	675	750	825	975	1050	1125	1200	1350	1425	1575	1800
	250	375	625	750	1000	1125	1250	1375	1625	1750	1875	2000	2250	2375	2625	3000
T5	300	450	750	900	1200	1350	1500	1650	1950	2100	2250	2400	2700	2850	3150	3600
	400	600	1000	1200	1600	1800	2000	2200	2600	2800	3000	3200	3600	3800	4200	4800
	600	900	1500	1800	2400	2700	3000	3300	3900	4200	4500	4800	5400	5700	5700*	5700*
T6	600	900	1500	1800	2400	2700	3000	3300	3900	4200	4500	4800	5400	5700	6300	7200
	800	1200	2000	2400	3200	3600	4000	4400	5200	5600	6000	6400	7200	7600	8400	8400**

* $I3_{max}$ T5 600 = 9.5 I_n
** $I3_{max}$ T6 800 = 10.5 I_n

Figure 4.7: Block I



	In(A)	I4(In) 0,2	0,25	0,45	0,55	0,75	0,8	1
T4	100	20	25	45	55	75	80	100
	150	30	37,5	67,5	82,5	112,5	120	150
	250	50	62,5	112,5	137,5	187,5	200	250
T5	300	60	75	135	165	225	240	300
	400	80	100	180	220	300	320	400
	600	120	150	270	330	450	480	600
T6	600	120	150	270	330	450	480	600
	800	160	200	360	440	600	640	800

Figure 4.8: Block G

Circuit-breakers for power distribution

Electronic trip units

PR222DS/P, PR222DS/PD and PR223DS(5) - Protection functions and parameterisations

Protection functions		Trip threshold	Trip curves(1)	Excludability	Relation t = f(I)
L	Against overload with long inverse time delay trip and trip characteristic according to an inverse time curve (I^2t= k) according to IEC 60947-2 Standard	**Manual setting** I_1 = 0.40....1 x In step = 0.02 x In	**Manual setting** at 6 x I_1 t_1 = 3 - 6 - 9 - 18s(2)	–	t = k/I^2
		Electronic setting I_1= 0.40...1 x In step 0.01 x In Trip between 1.1...1.3 x I_1	**Electronic setting** at 6 x I_1 t_1 = 3...18s step 0.5s(2) Tolerance: ± 10%		
S	Against short-circuit with inverse short time delay trip and trip characteristic with inverse time (I^2t= k) or definite time	**Manual setting** I_2 = 0.6-1.2-1.8-2.4-3-3.6-4.2-5.8-6.4-7-7.6-8.2-8.8-9.4-10 x In(3)	**Manual setting** at 8 x In t_2 = 0.05 - 0.1- 0.25 - 0.5s	■	t = k/I^2
		Electronic setting I_2 = 0.60...10 x In step 0.1 x In Tolerance: ± 10%	**Electronic setting** at 8 x In t_2 = 0.05...0.5s step 0.01s Tolerance: ± 10%(4)		
		Manual setting I_2 = 0.6-1.2-1.8-2.4-3-3.6-4.2-5.8-6.4-7-7.6-8.2-8.8-9.4-10 x In(3)	**Manual setting** t_2 = 0.05 - 0.1 - 0.25 - 0.5s	■	t = k
		Electronic setting I_2 = 0.60...10 x In step 0.1 x In Tolerance: ± 10%	**Electronic setting** t_2 =0.05...0.5s step 0.01s Tolerance: ± 10%(4)		
I	Against short-circuit with instantaneous trip	**Manual setting** I_3 = 1.5-2.5-3-4-4.5-5-5.5-6.5-7-7.5-8-9-9.5-10.5-12 x In(3)	instantaneous	■	t = k
		Electronic setting I_3 = 1.5...12 x In(3) step 0.1 x In Tolerance: ± 10%			
G	Against earth fault with inverse short time delay trip and trip characteristic according to an inverse time curve (I^2t= k)	**Manual setting** I_4 = 0.2-0.25-0.45-0.55-0.75-0.8-1 x In	**Manual setting** up to 3.15 x I_4 t_4 = 0.1s; up to 2.25 x I_4 t_4 = 0.2s; up to 1.6 x I_4 t_4 = 0.4s; up to 1.10 x I_4 t_4 = 0.80s	■	t = k/I^2 (6)
		Electronic setting I_4 = 0.2...1 x In step 0.1 x In Tolerance: ± 10%	**Electronic setting** t_4 = 0.1...0.8s step 0.01s Tolerance: ± 15%		

(1) These tolerances hold in the following conditions:
- self-powered trip unit at full power and/or auxiliary supply
- two or three-phase power supply

In conditions other than those considered, the following tollerances hold:

	Trip threshold	Trip time
S	± 20%	± 20%
I	± 20%	≤ 50ms
G	± 20%	± 20%

(2) For T4 In = 320 A and T5 In = 630 A ⇒ t_1 = 10.5s
(3) For T4 In = 320 A and T5 In = 630 A. T6 In = 1000 A ⇒ I_2max = 9.5 x In and I_3max = 9.5 x In
For T6 In = 800 A ⇒ I_3max = 10.5 x In
(4) Tolerance: ± 10 ms
(5) The setting of the PR223DS trip unit is electronic only (local/remote)
The L protection can be set at I_1 = 0.18...1 x In. For I_1 < 0.4 x In the neutral setting must be at 100% of that of the phases
(6) t = k/I^2 up to the current value indicated, t = k (equating to the chosen setting) beyond the current value indicated

Figure 4.9: Overview of Block settings form Technical catalogue 1SDC210015D0203 - Edition 2008

- Block G is a little bit complicated, that is why we will just define one I-t characteristic i^2t 3.15xIn for t=0.1 sec

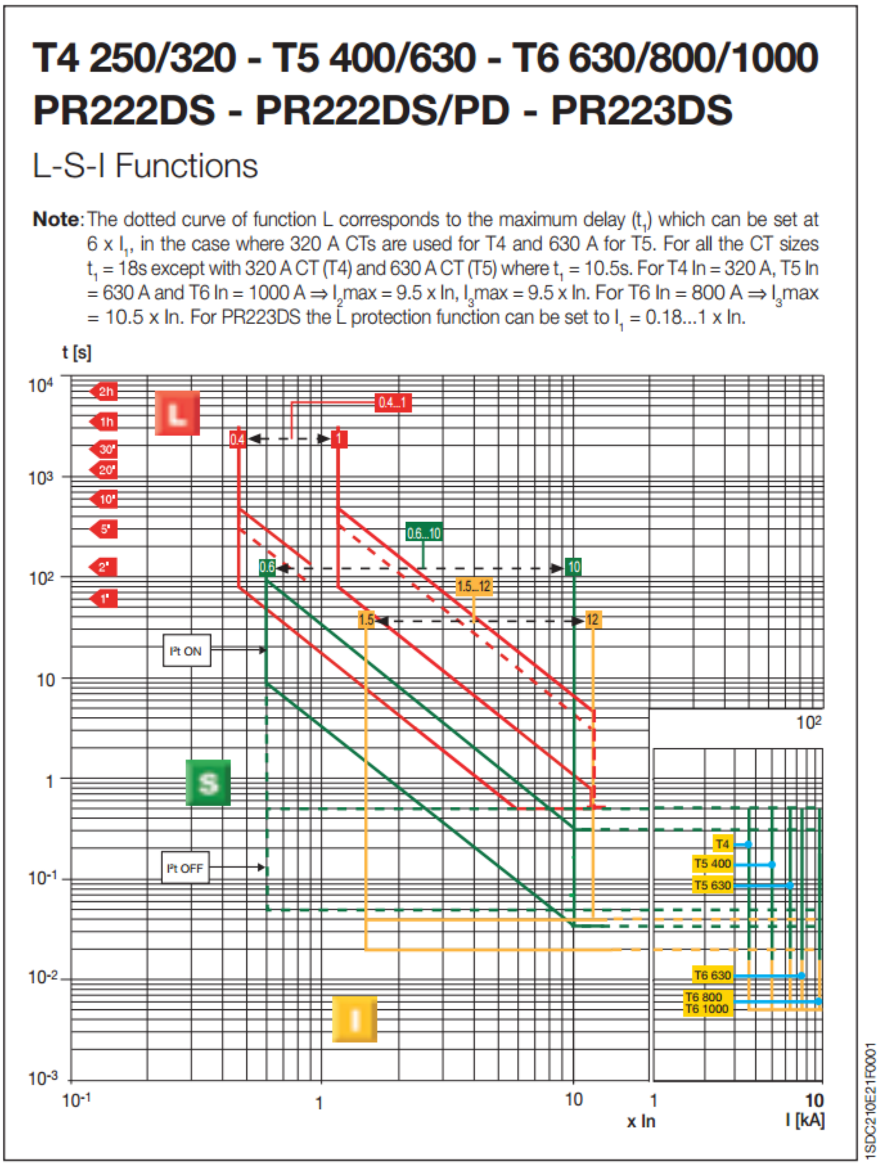


Figure 4.10: L,S and I characteristic form Technical catalogue 1SDC210015D0203 - Edition 2008

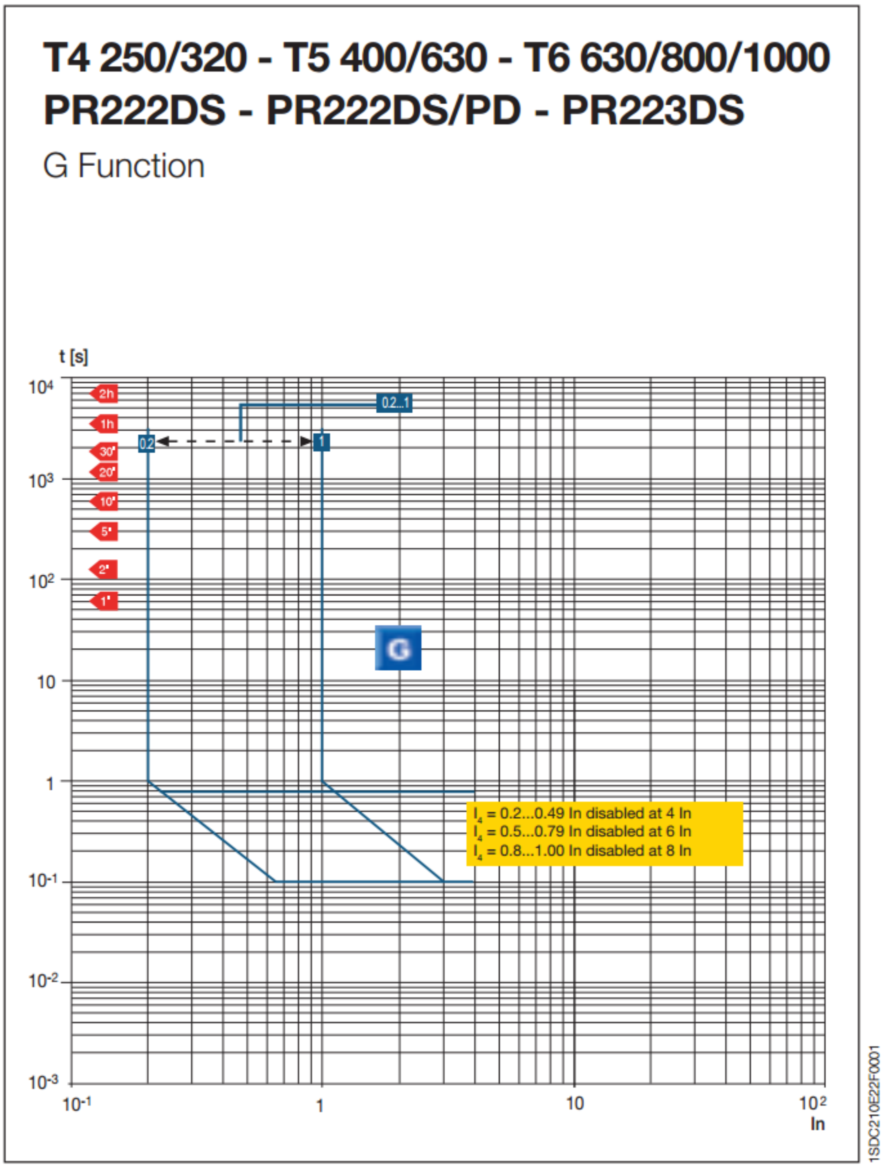


Figure 4.11: G characteristics form Technical catalogue 1SDC210015D0203 - Edition 2008

Now when the circuit breaker type is finished, insert one on the “Load Cable”. For circuit breakers in *PowerFactory* we need a CT with 1/1 A. Create the Overcurrent plot and check if the characteristic is as expected. If needed make corresponding changes.

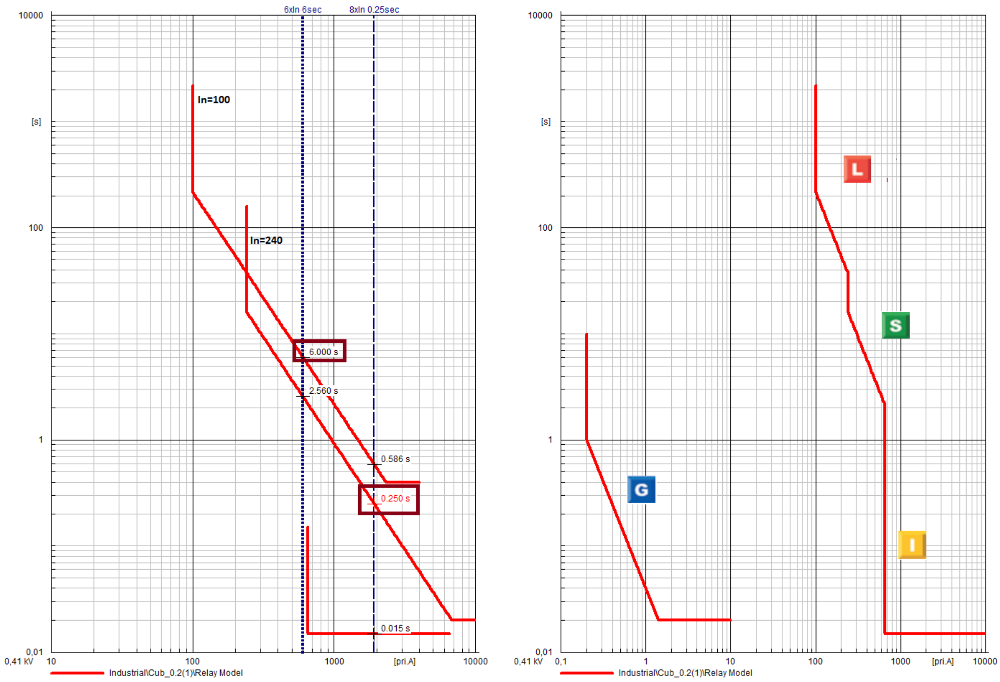


Figure 4.12: Overcurrent plots in *PowerFactory*

File Name	**Description**
RelayModelling_Start.pfd	File used to do the exercise

5 Communication between relays

Here in this example we will have a look at the two distance relays build on the Feeder Cable. They are set so that the first zone covers 90 % of the line length. This means that if the short circuit occurs on for example 95% of the line length one of the relays will trip in zone1 and the other one in zone 2. On this way cable is exposed for the longer time against short circuit current. Our task here is to define communication so that one relay send the information if it sees fault in the zone1 to speed up the second relay if this sees the fault in the zone2.

- Switch the external grid in service just to have feeding from both sides of the line in case of short circuit on the cable.
- Execute the short circuit on the 50% of the line. How do relays react? Repeat the calculation on the 5 and 95%.
- Define a path starting on the Intake/Incomer busbar and ending on the Industrial/B1 busbar. Path will contain only this one cable.
  – *Select all elements that should be in path → right mouse button click→ Path/New*. For more information on path please refer to User Manual.
- Create Time-Distance plot for the path.
  – *Select one element of the path → right mouse button click→ Path/Time-Distance Diagram*. For more information on path please refer to User Manual.
- Execute the short circuit sweep on the path.
  – *Double click on the diagram → Select Method to be "Short Circuit Sweep"→ Press "Shc-Calculation . . . and than Execute"*. For more information on path please refer to UserManual.
- Communication between two or more relays in *PowerFactory* will always be done through another third relay, which task is only to connect.
- Here we will also start by inserting a new page in the graphic board. For this select the *Insert New Graphic* option from the local graphic window icon bar .
- In the new open window select the option *Block/Frame Diagram*.
- Place 2 slots on the empty Frame Diagram by selecting the from the Draw toolbox. Each of them will represent one of two relays we are connecting.
- Next step is to select type and input/output signals of the slots.

Name	Class Name	Input Signals	Output Signals
R1	ElmRelay	yIn	yExt
R2	ElmRelay	yIn	yExt

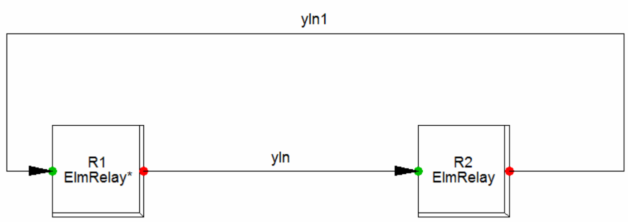


Figure 5.1: New communication Relay

- Next step is to define the Relay Type that will refer to this frame.
  - In *Equipment Type Library* define new object of the type *TypRelay*

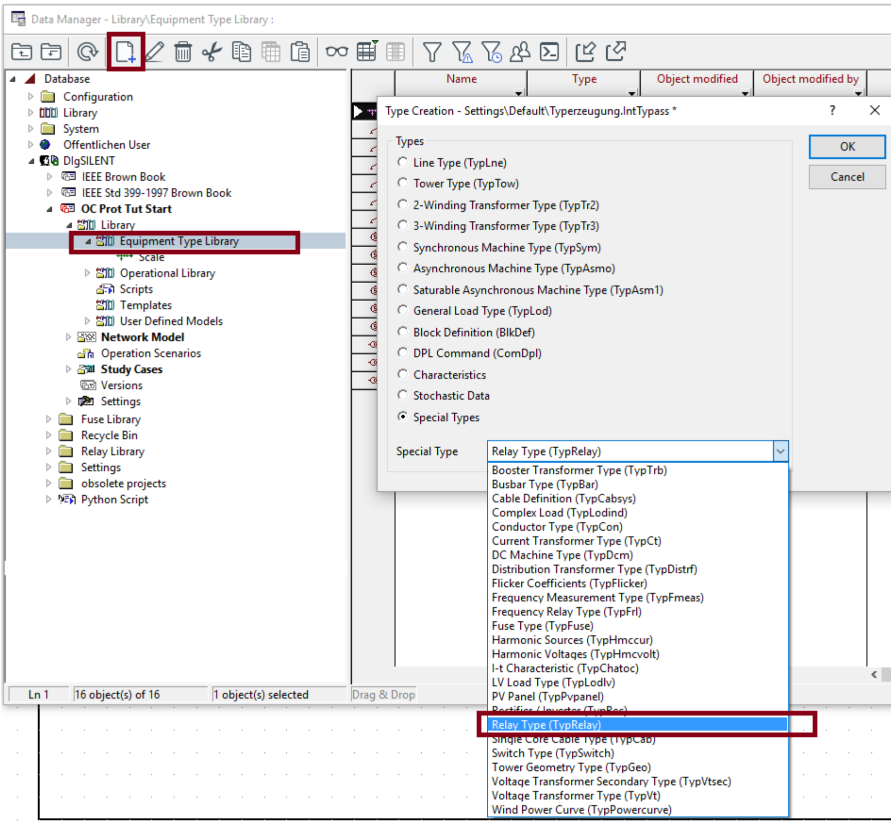


Figure 5.2: New Type object

  - Copy the frame from the *User Defined Models* folder into *TypRelay* object and link our relay type object with it under option *Relay Definition*

- New relay is ready so we just have to make small changes of the old used distance relay type. We have to prepare it so that it will be able to give and receive signals of interest. What we want is one distance relay to send information if it sees the fault in the zone 1 and to receive the same signal from the other relay. Received signal will be than sent to logic block and will pe processed.
  - Open the frame graphic of the Distance relay. *Right mouse button click on the RelType element* → *Show Graphic*
  - Deactivate the *Freeze Mode* if needed, by selecting from local icon bar.
  - Define the new input signal that will be led to logic block by just simple connecting left edge of the Frame with the logic block via signal element. Before this is possible you will have to define the input contact on the logic block.
    - Open the logic slot and define new input signal. Here is very important that this name matches name of the input signal from the connecting Relay (in out case yIn).
  - Define also the output signal of the relay that should be used as input of the second relay. In our case signal if the fault is in the first zone.
    - For this just connect output of the Zone 1 block with the right edge of the relay frame. Press rebuild button form the local icon bar .

* Double click on this signal line. In the new open window you will see that this signal connects output signal *yout* of the *Ph-Ph Polygonal 1 Delay* block with the output signal *yZ1* of the whole relay *F21 Polygonal*. And this is fix- not changeable.

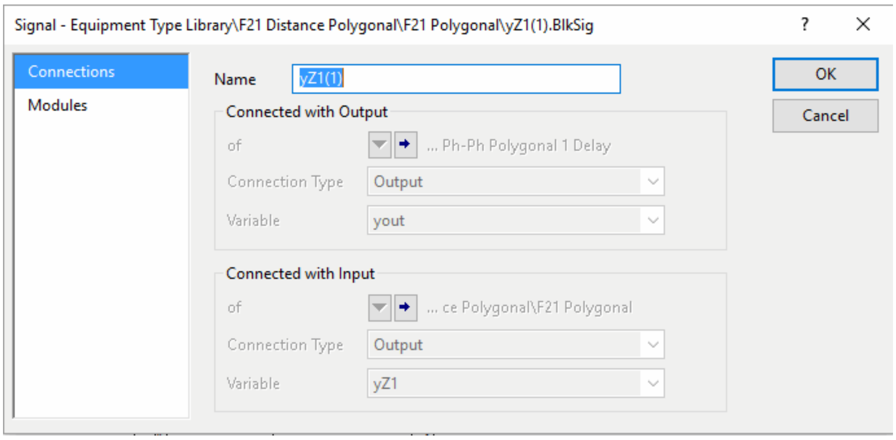


Figure 5.3: Output signal windows

* This is why we will also have to adopt the Connection Relay by changing the output of the two blocks available here. R1 and R2 from yExt to yZ1

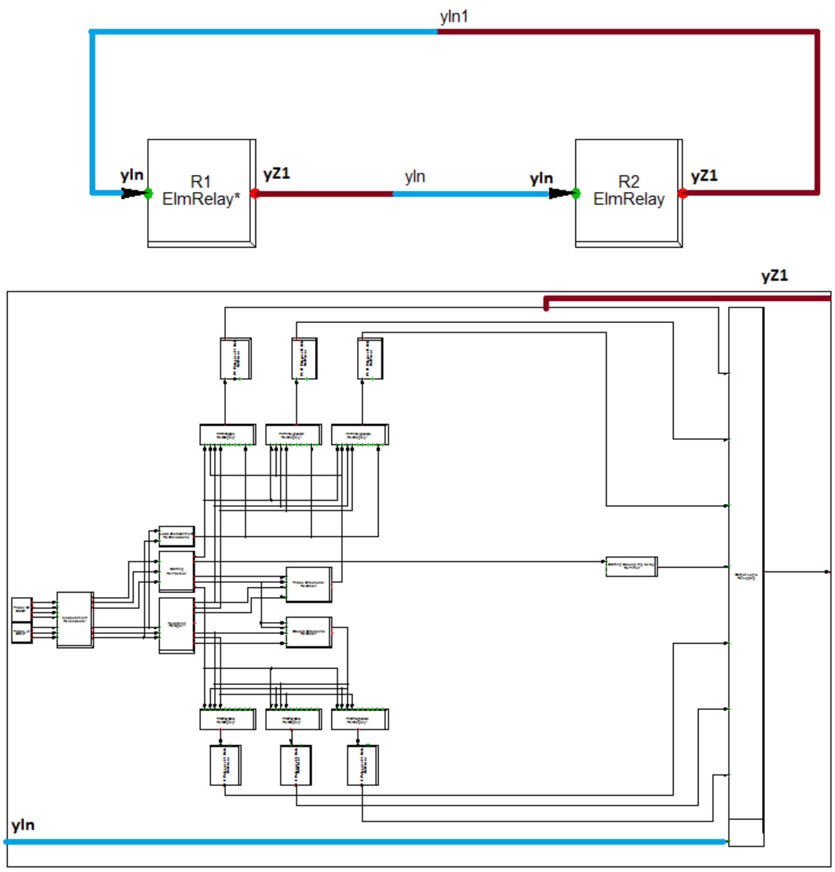


Figure 5.4: New Connecting and changed Distance relay frames

– Last thing we have to adjust is Logic block. We have to change the logic so that if opposite relay sees the fault in zone 1 our relay trips instantaneously.

* Open the logic type block and adjust it.

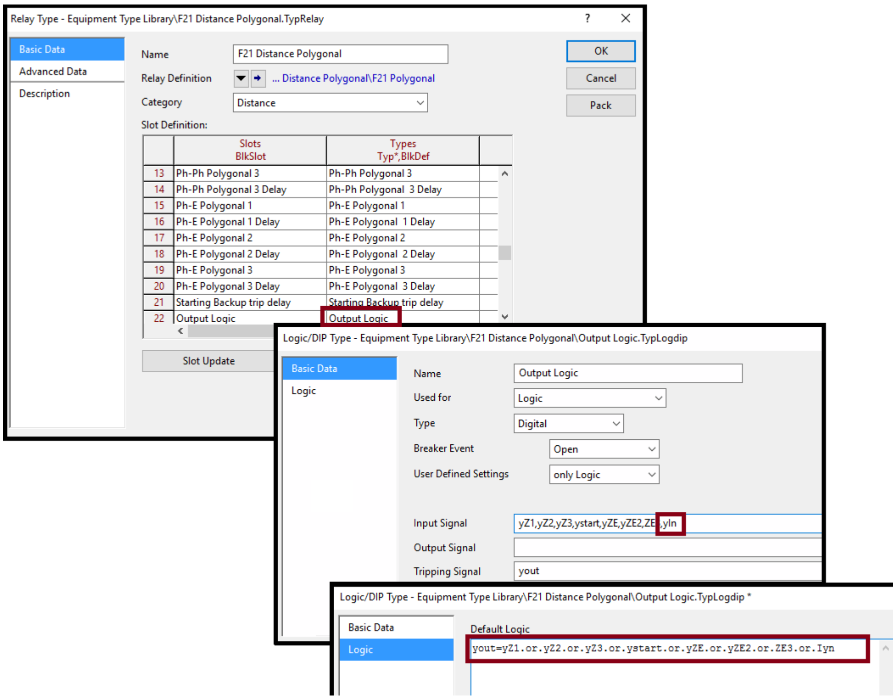


Figure 5.5: Logic adjustment

5.1 Testing the new connecting relay

- Build the new relay of the type we created above and select the two relays R1 and R2 that should be connected.
  - New relay will be filled with only one R1 relay. To remove one of them just *right mouse button click on one of the relays → Remove ElementType*
  - Select the R2 relay on the field R2. One way is by double clicking the empty field and selecting the relay R2. Other way is to open the cubical where the relay R2 is, to copy it and past it here inside the empty field.

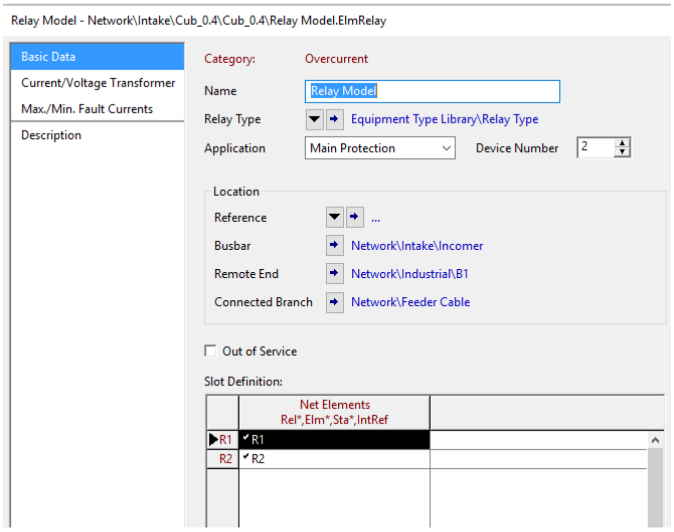


Figure 5.6: New relay

- To test this relay switch to the Time Distance diagram and run the short circuit sweep once again. Are there any changes?

File Name	Description
RelayModelling_Start.pfd	File used to do the exercise
RelayCommunication_Finish.pfd	Solution of exercise

6 Understanding distance relay modelling in *PowerFactory*

The objective of this exercise is to introduce directional protection models and blocks that one distance relay must contain in *PowerFactory* . We will concentrate here on 7SA6 Relay model from Siemens.

- Import and activate the project “RelayDistance_Start.pfd”
- Have a look at the relay model and identify all used relay blocks.
  - Open the Relay inside of the *Data Manager*

    Option 1 : *Right mouse click on the relay it self → Edit and Browse Date . . .*

    Option 2 : *Open the relay → press CTRL and E on the key board*

    Option 3 : Open *Data Manager* and navigate to Relay
  - Go with the right mouse taster on the relay and select *Show Graphic* from context menu.

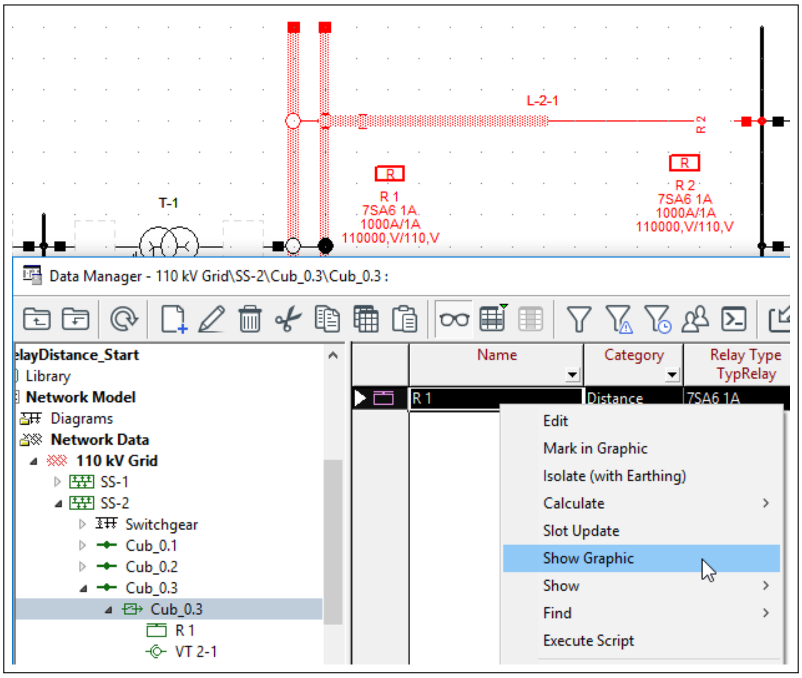


Figure 6.1: Show Graphic of the Relay model

  - Clock with the right mouse button on the new opened page and select *Show Library Object*

In distance relay 6 impedances for each fault-loop (L1-E;L2-E;L3-E;L1-L2;L2-L3;L3-L1) will always be calculated and compared with impedances set inside of the zones (Z1,Z1b,Z2,Z3,Z4,Z5). In case that starting block detects an fault this calculated impedance will be compared with the impedance setting of the zone. If the calculated impedance is smaller than the one set in the zone, this zone will see a fault and send this information to logic. In case that none of the active zones sees the fault than the back-up (directed or undirected) protection should react, if set so.
$Z_{Lx-E} = U_{Lx-E}/I_{Lx}$
$Z_{Lx1-Lx2} = U_{Lx1-Lx2} - /(I_{Lx1} - I_{Lx2})$
This is easy to be seen on the R-X Diagram, where each calculated value in *PowerFactory* is represented by one red arrow.

- Run *Load Flow calculation*. Present the calculated values for line-ground voltage its magnitude and angle in the result box for terminals and current magnitude and angle for branch elements.
- Use the formulas above to recalculate the impedance that relay sees in load flow case. We are having symmetrical load flow so all impedances seen by the relay are being equal.
- Change to R-X Diagram and check if the *PowerFactory* results and the one you have calculated are the same. (R1=205,203exp(24,565))
- Execute the 2ph min Short circuit at 50% of the first line and have a look how the impedance at the R-X diagram changes. Also red arrows are appearing on the R-X diagram. Each arrow represents one of 6 impedances.
- Does the Relay trip and how fast?

6.1 Starting Block

One of the first blocks that information from current and voltage transformers has to be passed to is the **starting block**. This block is responsible for detecting fault and sending this information to various blocks for further investigation. The Starting block is getting the phase and zero sequence currents from the measurement block/blocks and its output signal is used to activate the directional block (“Dir-Z”) and the zone blocks.

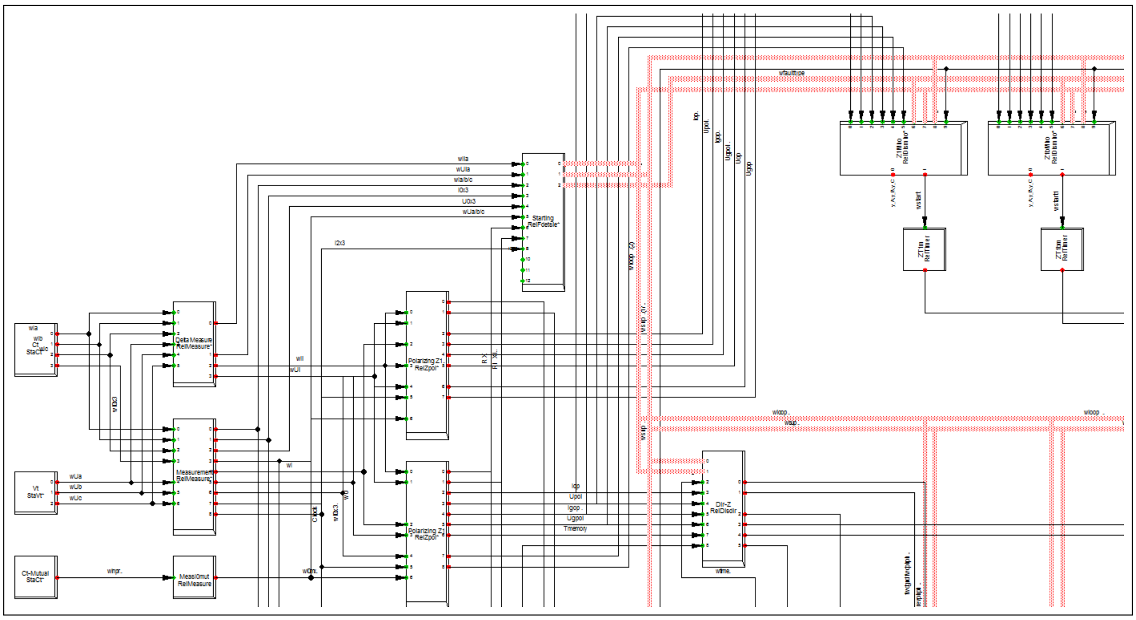


Figure 6.2: StartingBlock

In many distance relays this block is also having undirectional backup protection function.

There are three types of this block in *PowerFactory*

- General starting block (**RelFdetect**).
- Siemens starting block (**RelFdetsie**).
- ABB starting block (**RelFdetabb**).

The general starting block implements the more common fault detection logics present in the distance relays (Overcurrent (phase and ground); Underimpedance as voltage restrained overcurrent; Underimpedance with impedance calculation).

The Siemens starting block implements the fault detection logic available in the AB 7SA5x and 7SA6xx relay family. Four different types of fault detection are available: the “Overcurrent $I >>$”, the “Underimpedance U/I” , the “Impedance Z” and the “Underimpedance U/I/phi” detection logic.

The ABB starting block implements the fault detection logic available in the ABB REL100, REL 316, REL 5xx relay family and REL670 relay. Two different types of fault detection are available: “Overcurrent” and “Impedance Z” (underimpedance) detection logic.

- Execute 2ph minimal short circuit at the busbar “SS1-1” once again and open the Relay R1 in *Data Manager*.
- Select Starting block and activate *Detail Mode*

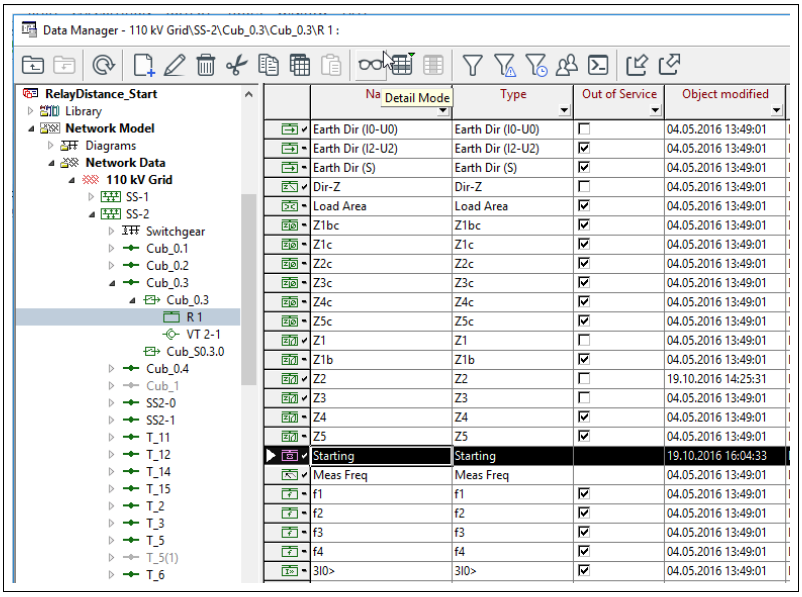


Figure 6.3: Selecting Starting block

- Change to *Flexible Data* page and print all available signals for this block in output window. Are input Signals of the starting block as expected?
- Investigate output signals. Does the relay trip and why?
  Two signals *yloop*-fault loop, *yfaulttype*- fault type are containing integer values starting with 0. This values have following meaning:
  FaultLoop {
  NONE=0, L1E, L2E, L3E, L1L2, L2L3, L3L1, L1E_L2E, L2E_L3E, L3E_L1E, L1L2_L2L3, L2L3_L3L1, L3L1_L1L2, L1E_L1L2, L2E_L2L3, L3E_L3L1, L2E_L1L2, L3E_L2L3, L1E_L3L1, L1E_L2E_L1L2, L2E_L3E_L2L3, L3E_L1E_L3L1, L1E_L2E_L3E, L1L2_L2L3_L3L1, L1E_L2E_L3E_L1L2_L2L3_L3L1, LOOPNOTAVAILABLE };

  FaultType {
  NOFAULT=0, ABCF, AE, BE, CE, AB, BC, CA, ABE, BCE, CAE};
- Open the input window for the starting block. What type of starting is selected? How high is the pick-up current?

- Open the R-X Diagram for the relay R2. Does this relay trip? Why?

Over-current method can be used in networks with small line impedances and strong infeed. It can be applied at any distance relay if in the case of minimal short-circuit, the current seen by the relay is sufficient large.

- Activate “U/I Starting” *Variation* and run the 2ph minimum short circuit on the 20 % of the line “L-1-3” starting at the terminal i.
- Switch to R-X Diagram representing R1 relay. Does the relay trip?
- Even if one of the impedances (in this case Zl_B) is inside of the R-X diagram (Zone 3) that should lead to tripping of the relay (in 0.8 sec), this relay does not trip.
- Let us have a look inside of the relay:
  - Print the input signal values of the Logic block of this relay in output window. Are there any tripping signals sent to Logic block?
  - Have a look at the zone 3 block the one we would expect to trip. Show all input values of this block. Concentrate on the relevant signals coming from the Starting block. Does the Starting block recognize this fault?
  - Have a look at the input and output signals of the starting block. What was the problem? (Ikss=1,288sec.A ¡ Ipickup=1,5sec.A)

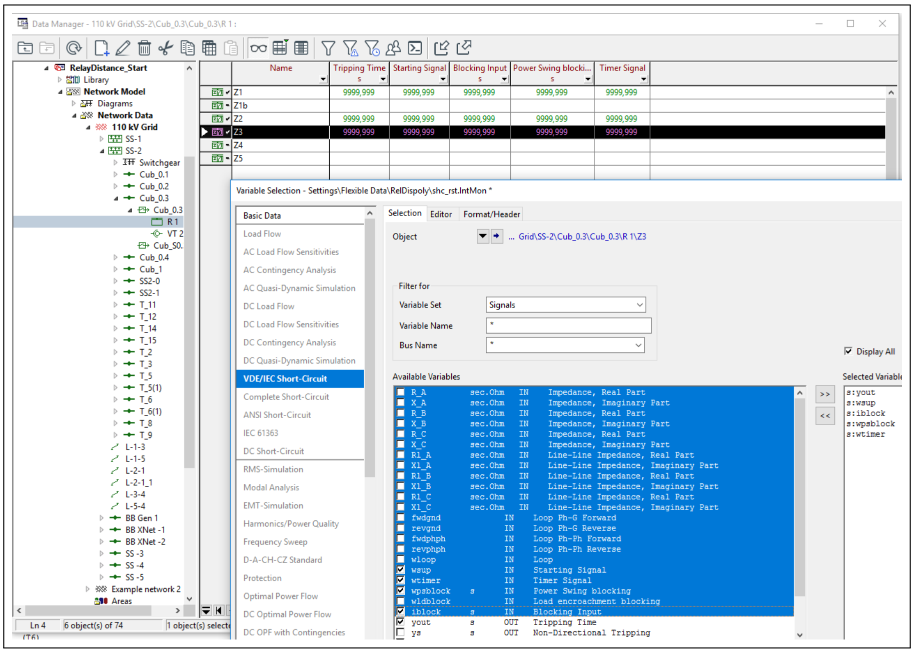


Figure 6.4: Input signals from zone 3 block

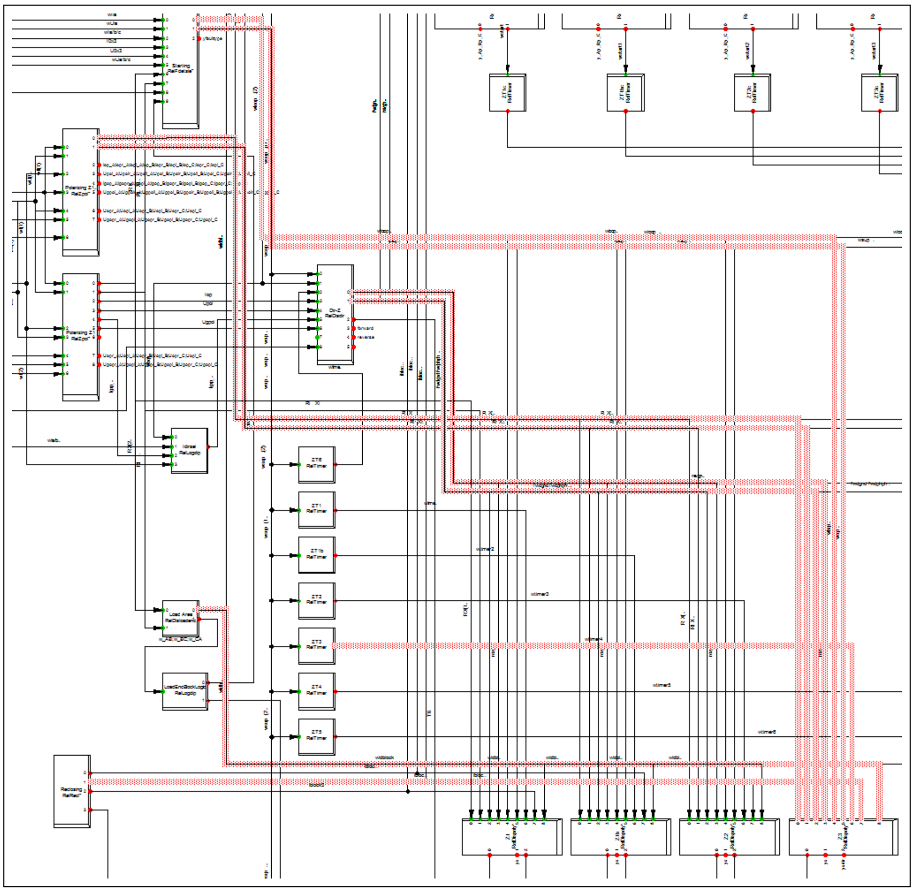


Figure 6.5: Input signals from zone 3 block

In case that we have a weak source, high earth current limiting impedance in the earthing of transformer or current splitting through parallel paths, short circuit current can drop under the overcurrent limit. In this case a different starting method then the overcurrent should be considered.

- One of the ways to solve the problem would be to decrease pick-up overcurrent $Iph >>$ but here we are facing problem that nominal current of the cable is 1.2kA and 1.288kA is the 2ph min short circuit. That is why we are introducing U also as relevant parameter to be considered.
- Select Underimpedance U/I as type of starting and select
  $Iph >= 0.5I_N; Iph >>= 2.5I_N$
  $U(I >) = 70\%U_N; U(I >>) = 100\%U_N;$
  $Prog.U/I = LE\_Uphe/LL\_Uphph$
- Run the short circuit once again. Does the relay trip? Have a look inside of the Starting block signals. Why does the relay trip?
- Execute the 2ph minimum short circuit on the same line but on the 50% of the line length. Change to R-X diagram. How fast did the relay trip? Check the time settings for the starting block.
- Find out which block was responsible for this tripping (T6 signal; Directional Block with its directional back-up protection feature)

6.2 Polarizing Block

Polarizing block is providing the distance protection zones the “polarized” current and voltage signals which will be used to evaluate the trip rules. It is getting as input the voltage and current signals and is providing as output:

- Operating currents
- Polarizing voltages
- Operating voltages

If set **polarizing** block calculates also the impedance values. It supports different representations of the earth fault compensation factor. This factor is usually used by the distance protection to allow the user inserting the same trip impedance values for the phase-phase faults and the phase-ground faults. This block is providing the signals to distance zones and to the directional block.

- Execute 1ph maximum short circuit on the 50% of the Line L-2-1 and switch to R-X diagram for the R1 relay. Are the red arrows correctly represented.
- Open two polarization blocks and adjust earth factors k0 and Angle according to the line parameters. Easiest way is to adjust this by clicking on the *Assume k0* button. Also set the same line angel.
- Execute once again short circuit and have a look at R-X diagram. Any graphical changes on the R-X Diagram?
- Define a path starting on the SS2 busbar and ending on the SS1 busbar. Path will contain only one Line. Important not to forget to mark coupling between SS2-1 and SS2-0.
  - *Select all elements that should be in path* $\rightarrow$ *right mouse button click*$\rightarrow$ *Path/New*.
- Create Time-Distance plot for the path.
  - *Select one element of the path* $\rightarrow$ *right mouse button click*$\rightarrow$ *Path/Time-Distance Diagram*.
- Execute the 1ph maximum short circuit sweep on the path.
  - *Double click on the diagram* $\rightarrow$ *Select Method to be “Short Circuit Sweep”*$\rightarrow$ *Press “Shc-Calculation . . . and than Execute”*.
- Does the relay reacts as expected on the whole path?
- Execute the 2ph minimum short circuit sweep on the path just to check if phase to phase faults are recognised correctly.
- Execute the 1ph maximum short circuit on the 50 % of the line “L 2-1” and change to R-X diagram. Is this fault recognised as 1ph short circuit?
- Open the input window for the starting block and navigate to *Earth Fault Detection* page. How high are the $Ie >$ and $Ue.grd >$ settings for Earthed Networks?
- Open starting block in *Data Manager* and check the values for input signals I0 and U0.
- Open starting block once again and change the *Earth Fault detection* option inside of the *Earth Fault Detection* page from $Ue > andIe >$ to $Ue > orIe >$.
- Repeat short circuit sweep. Any changes?

6.3 Directional Block

Detection of current direction is quite important when it comes to two sides in-feed. The distance directional block (RelDisdir(TypDisdir class)) is mocking up the directional characteristic present inside the more common distance relays. It is also able to model exactly the directional elements of the Siemens and of the ABB relays. It gets as input signals the current and voltage values and calculate the impedance values. The directional evaluation is performed on the R-X diagram.

- Execute the 1ph minimum short circuit on 80% of the line “L-1-3” starting from terminal j and change to R-X diagram of the relay “R2”.
- Open the two Polarizing blocks and adjust the k0 factor and Angle and repeat short circuit calculation. Does the relay trip and how fast?
- Activate the Zone 3 and select its setting so that singe phase short circuit from above is recognised and turned off in 0.8 sec.
- Change the type of used directional block according to the figure 6.6.

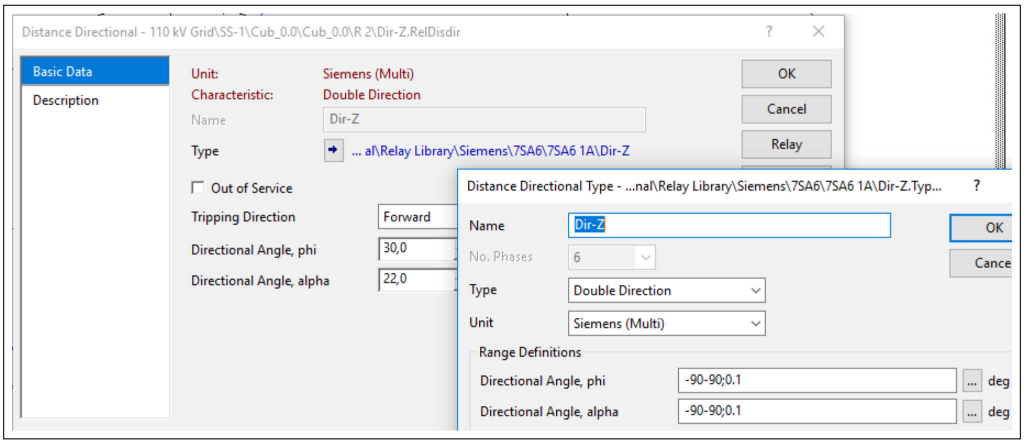


Figure 6.6: Changes of the directional block

- Move to R-X diagram and open the input window for the directional block. Make changes of both *Directional Angle phi* and *Directional Angle alpha* and pay attention how R-X diagram changes.
- Set *Directional Angle phi* to -40 deg and run the same short circuit. Does the relay trip? Why?
- Set this angels as they were on beginning phi=30 deg and alpha=22 deg.
- Execute single phase ground maximum fault on the 65% of the same line.
- How fast does the relay trip? Set the *Tripping Direction* parameter inside of directional block to be *Reverse* instead of *Forward*. Repeat short circuit. How fast does the relay trip? Why?
- Open the input window for current transformer and change the orientation to be $-- >$*Busbar*. Repeat the short circuit calculation. Any changes?

6.4 Polygonal and Logic Block

The **Polygonal** block implements the typical distance protection polygonal characteristics. As input signals each polygonal block is getting calculated values of impedance from polarising block, direction signal from directional block, time delay signal from timer .... As output this block provides logic block with tripping information. List of all input/output signals is available in technical reference documentation, one copy of input/output signals for Siemens relays is given below.

Input Signal	Description	Unit
R_A	Phase A - Ground loop resistance	Ohm
X_A	Phase A - Ground loop inductance	Ohm
R_B	Phase B - Ground loop resistance	Ohm
X_B	Phase B - Ground loop inductance	Ohm
R_C	Phase C - Ground loop resistance	Ohm
X_C	Phase C - Ground loop inductance	Ohm
Rl_A	Phase A - Phase A loop resistance	Ohm
Xl_A	Phase A - Phase A loop inductance	Ohm
Rl_B	Phase B - Phase C loop resistance	Ohm
Xl_B	Phase B - Phase C loop inductance	Ohm
Rl_C	Phase C - Phase A loop resistance	Ohm
Xl_C	Phase C - Phase A loop inductance	Ohm
fwdgnd	Ground forward signal (available when the fault is detected as a forward fault)	
revgnd	Ground reverse signal (available when the fault is detected as a reverse fault)	
fwdphph	Phase forward signal (available when the fault is detected as a forward fault)	
revphph	Phase reverse signal (available when the fault is detected as a reverse fault)	
wloop	ID of the loop from which the fault must be removed	
wsup	Supervising Signal	Y/N(or sec in the RMS simulation)
wtimer	Timer Signal (used to add an additional delay to the trip time)	Y/N(or sec in the RMS simulation)
iblock	Blocking signal	Y/N
Output Signal	**Description**	**Unit**
yout	Tripping signal/time	Y/N or seconds
y_A	Phase A tripping signal/time	Y/N or seconds
y_B	Phase B tripping signal/time	Y/N or seconds
y_C	Phase C tripping signal/time	Y/N or seconds

Figure 6.7: Input, Output signals of Polygonal Block

The Logic/DIP object (RelLogdip class) is used to implement user defined trip logics, dip switch inputs and simplified calculation of analogue signals. It allows defining the block behaviour and usage. Input, output signals and logic inside of this block is completely user definable.

Our task here will be to change relay model so that first zone Z1 only trips for the faults phase to phase. There are many ways to do this but we will select one in order to recapitulate learned from the previous chapter and to learn better how to work with signals and logic block. Here we will learn how to make any kind of changes that may be of help if creating new or changing existing relay models.

- Open the relay type topology in new window.

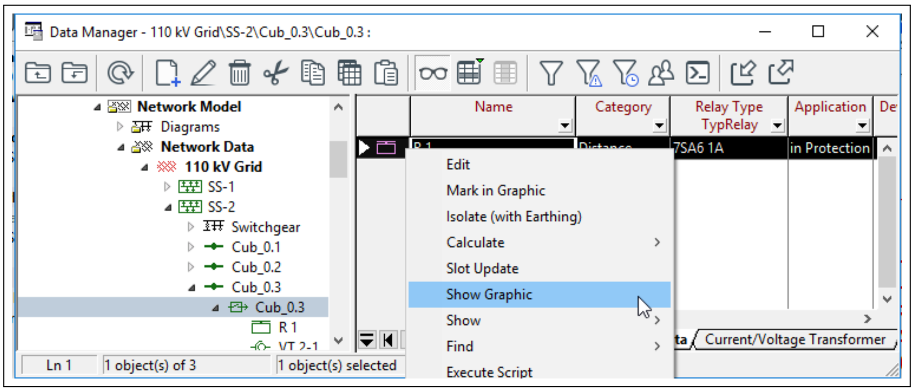


Figure 6.8: Open the relay type in new window

- *Right mouse taster on the new open window → Show Library Object*.
- If needed change dimensions of the used page by selecting *Drawing Format ...* from local icon bar.
- Zoom to the Z1 block and have a look at used input signals. Compare used input signals with those from the figure 6.7. Is the blocking signal input occupied?

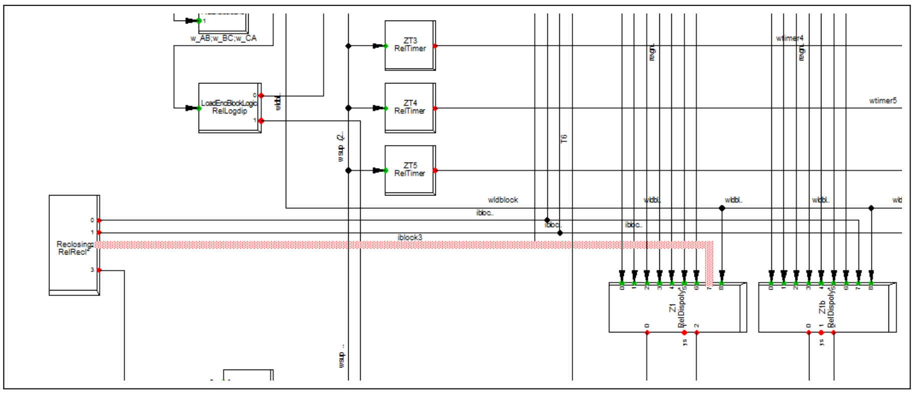


Figure 6.9: iblock signal of the Z1 block

- Place new Slot close to the Z1 and name it “NewBlockingLogic”. This block slot of the class *RelLogdip* will contain one input and one output signal.

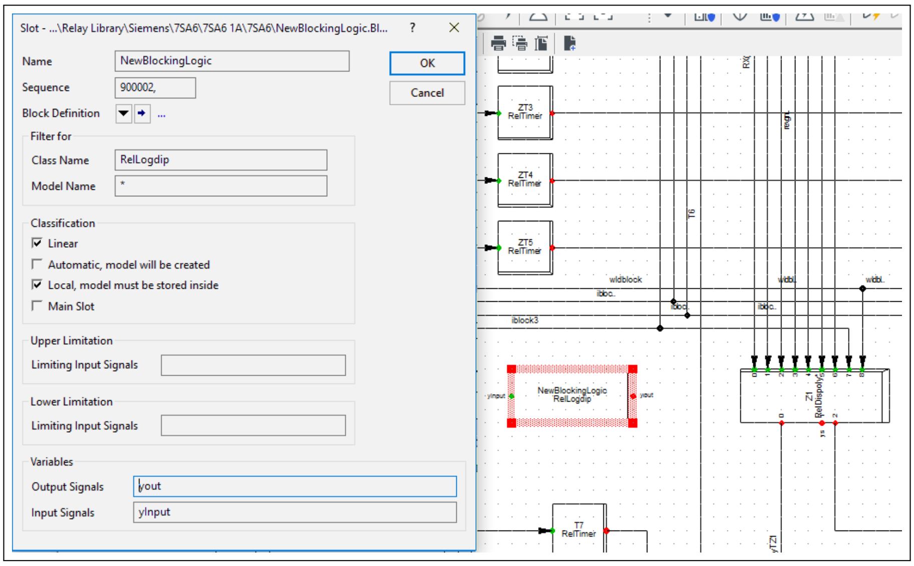


Figure 6.10: New Logic Slot

- Now we have to find one available signal inside of the relay that can help us to recognise type of the fault. We will use here output of the starting block*yfaulttype*. This signal is explained in sub-chapter 6.1. We could also use *I0x3* signal from the Measurement block. . .
- Release iblock input of the *Z1* by deleting connection signal to *Recloser*. This is input 7 of the *Z1*. Signal is marked in figure 6.9.
- Connect output of this new logic slot with input 7 of *Z1* and input of logic should be connected with the output 2 of the *starting* block.

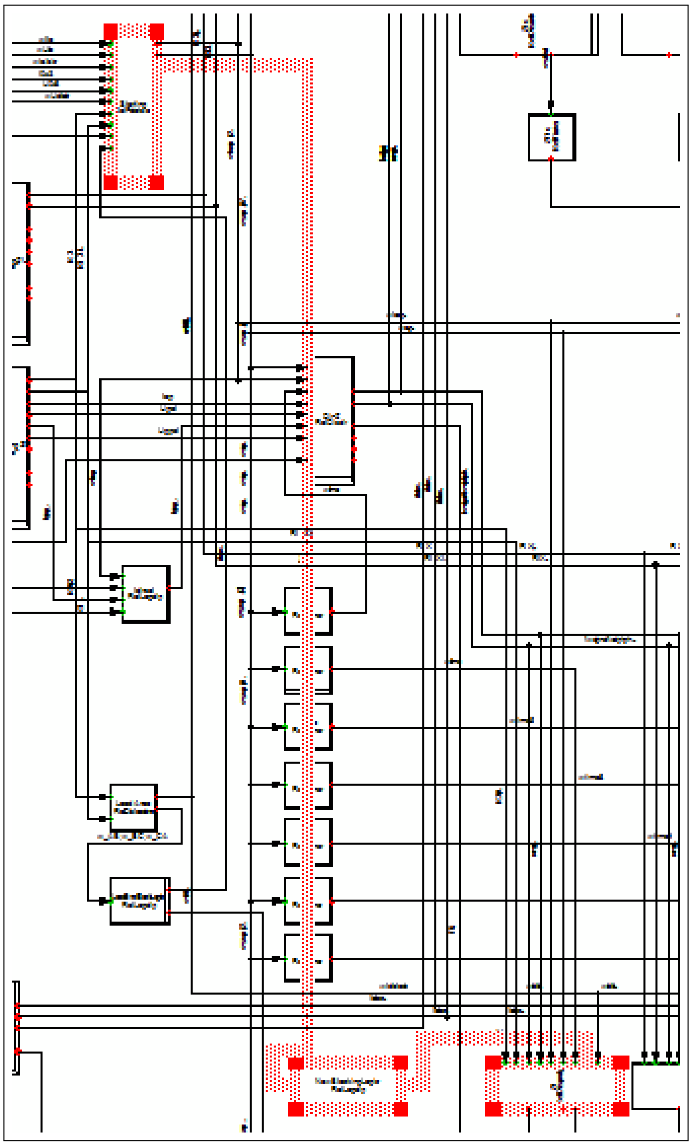


Figure 6.11: New connection of the slots

- Navigate to type of the relay inside of the *DataManager*, open it and press the *SlotUpdate* button. Confirm this change by pressing *OK*.

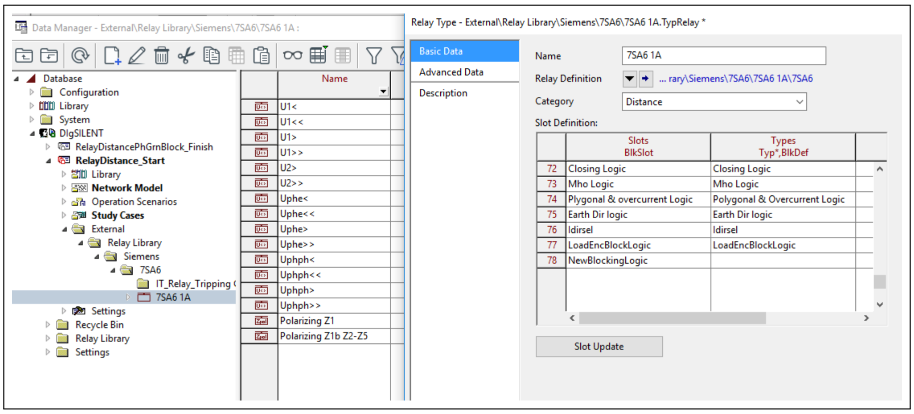


Figure 6.12: New connection of the slots

- Copy one of the existing *TypLogdip* blocks inside of the 7SA6 1A (for example Mho Logic) and rename it “NewBlockingLogicType”.
- Select Analogue as *Type* and None as *Breaker Event*. Set yInput and yout signals as shown in figure 6.13.

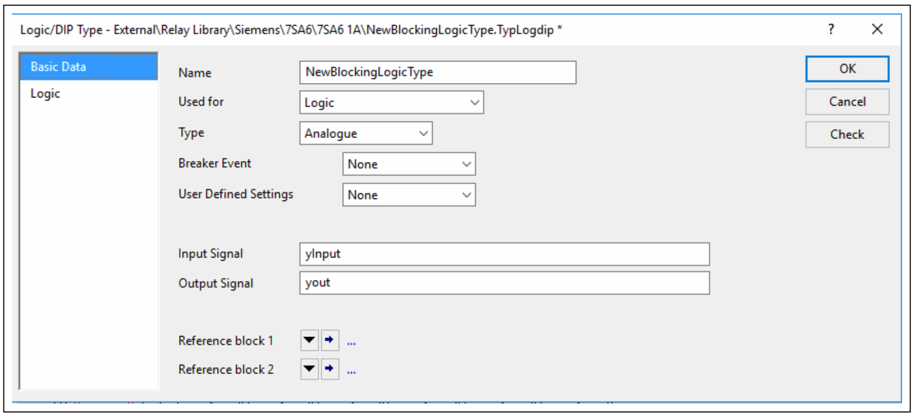


Figure 6.13: New Logic Type

- Inside of the *Logic* page we have to define how this signals are treated and to define blocking logic.

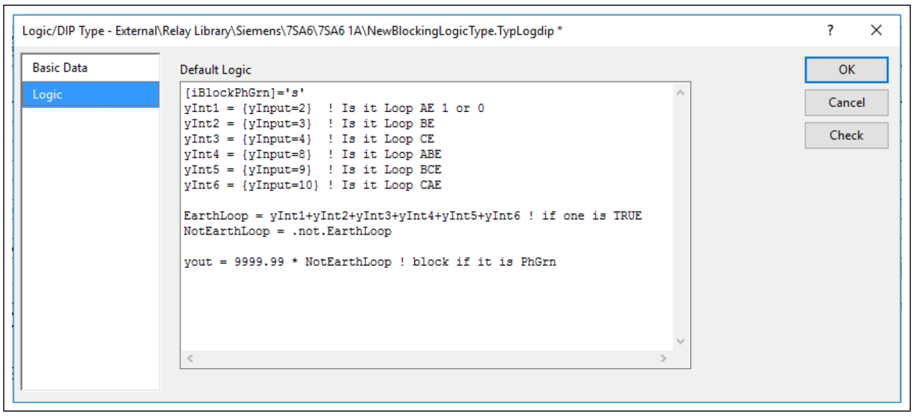


Figure 6.14: New Logic Type

- Select the new created type for corresponding slot inside of the TypRelay object by double click on the empty field and navigating to logic block type.

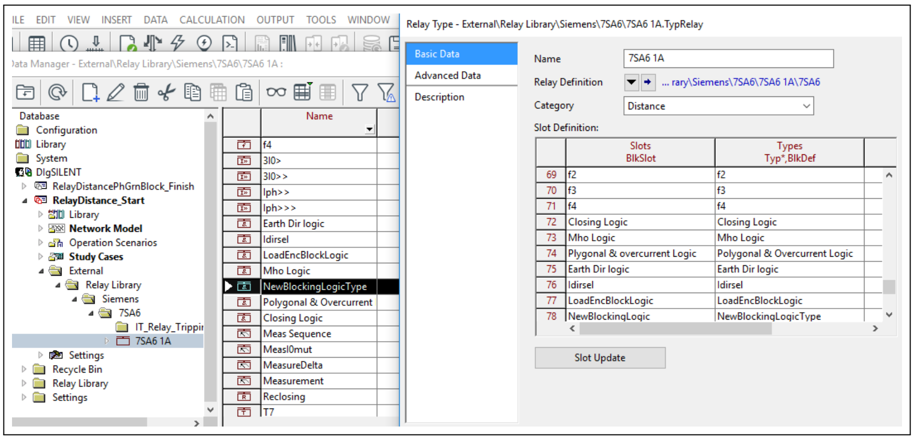


Figure 6.15: Select type for corresponding slot

- Open default relay and press *Slot Update*. Repeat this also for relays R1 and R2.
- Execute 3ph short circuit on the line “L-2-1” and change to R-X diagram of the relay R1. Does the relay trip and how fast?
- Execute now 2phph and single phase to ground fault. How does the relay react?
- Open new logic block in DataManager and present the values of the signals.

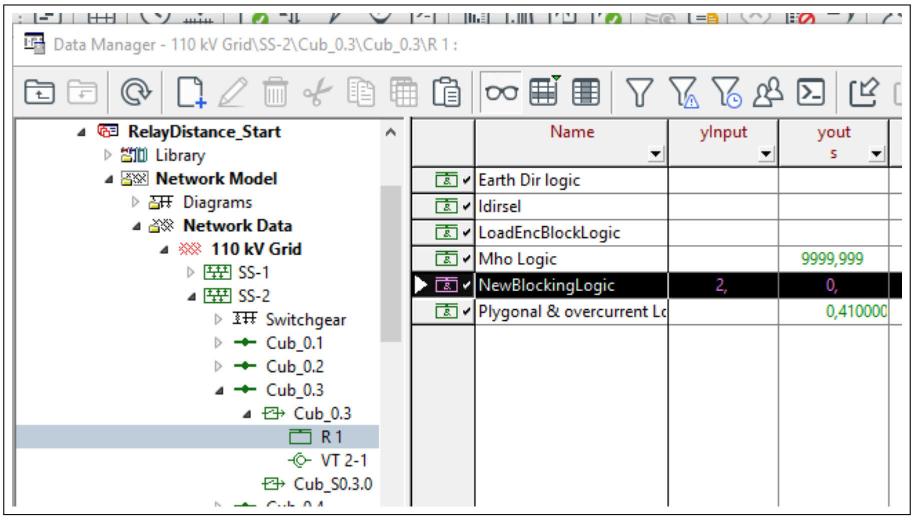


Figure 6.16: Input/Output signals and its values

- If having interest in values of internal variables that we have been using in new logic block, such as yInt1,yInt2,EarthLoop,. . . . Just define them as output signals in TypLogdip object. This is very useful by debugging.

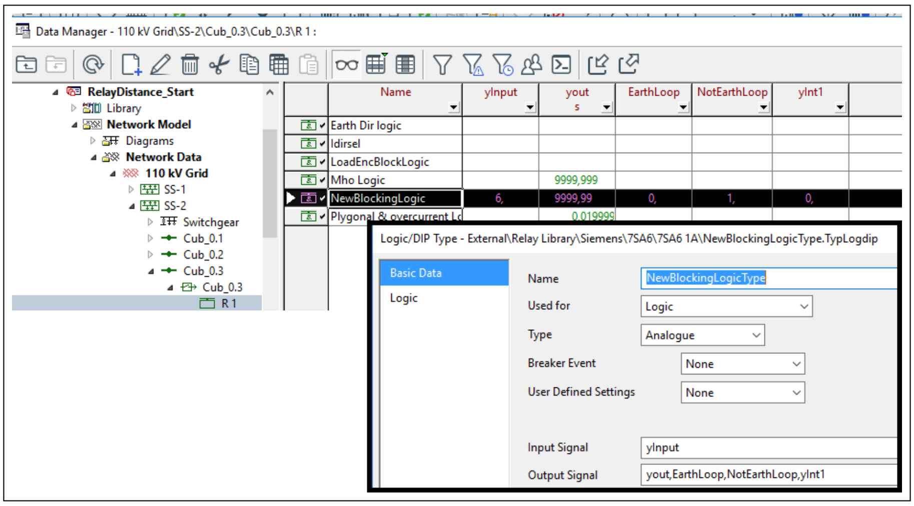


Figure 6.17: Input/Output signals and its values

PowerFactory files

File Name	Description
RelayDistance_Start.pfd	File used to do the exercise
RelayDistance_PolaFinish.pfd	Solution of exercise
RelayDistancePhGrnBlock_Finish.pfd	Solution of exercise