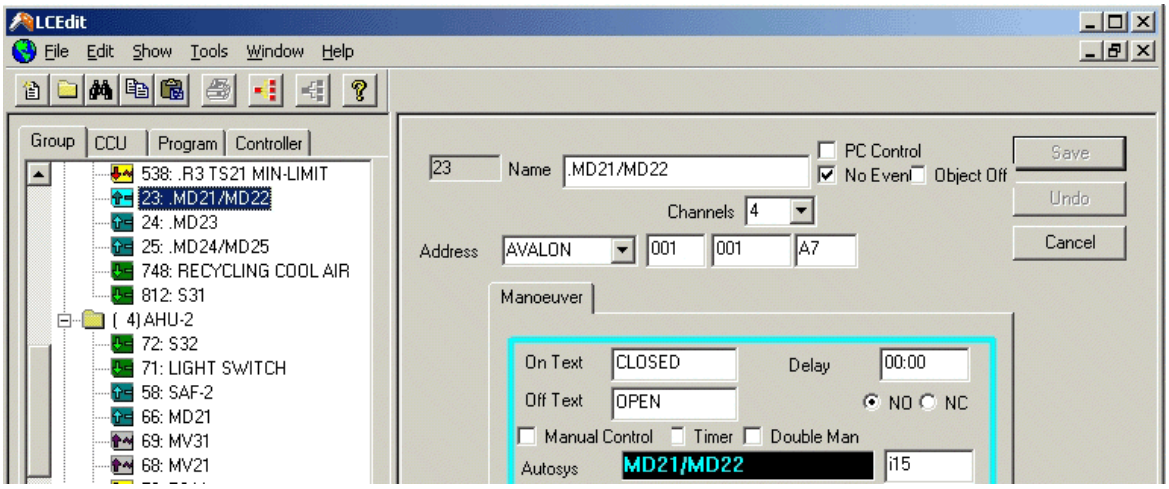
Basics



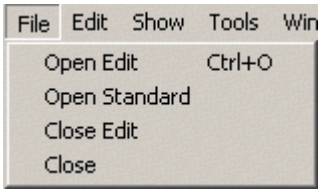
This programming-tool is used for programming of the LS920 and Avalon CCU, and also for connecting objects to Room regulator LS600 and others.

A program consists of a number of **Objects**, **Time Schedules**, **Regulators** and **Conditions**.

See also [Menu].

Menu

File



Open Edit

Opens editing.

Open Standard

Opens standard editing.

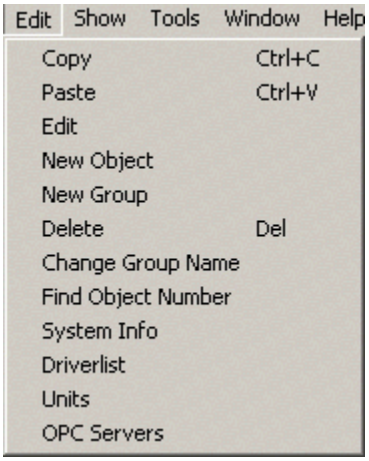
Close Edit

Closes editing.

Exit

Exits LCEdit.

Edit



Copy

Copies selected object, group or CCU.

Paste

Pastes an object, group or CCU.

Edit

Opens selected object for editing.

New Object

Creates a new empty object.

New Group

Creates a new group.

Delete

Deletes selected object or group.

Change Group Name

Changes group name.

Find Object Number

Marks searched object in the group list.

System Info

Shows information about the plant.

Driver List

Lists the programs and drivers started by LCService.

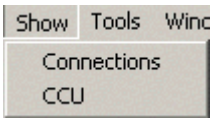
Units

Edit units.

OPC Servers

Lists the OPC Servers used.

Show



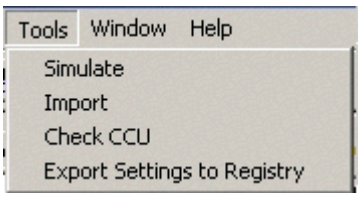
Connections

Shows the relations for selected object.

CCU

Shows the I/O-configuration in the CCU which the object belongs to.

Tools



Simulate

Starts the simulation mode.

Import

Import data from TA, EIB etc.

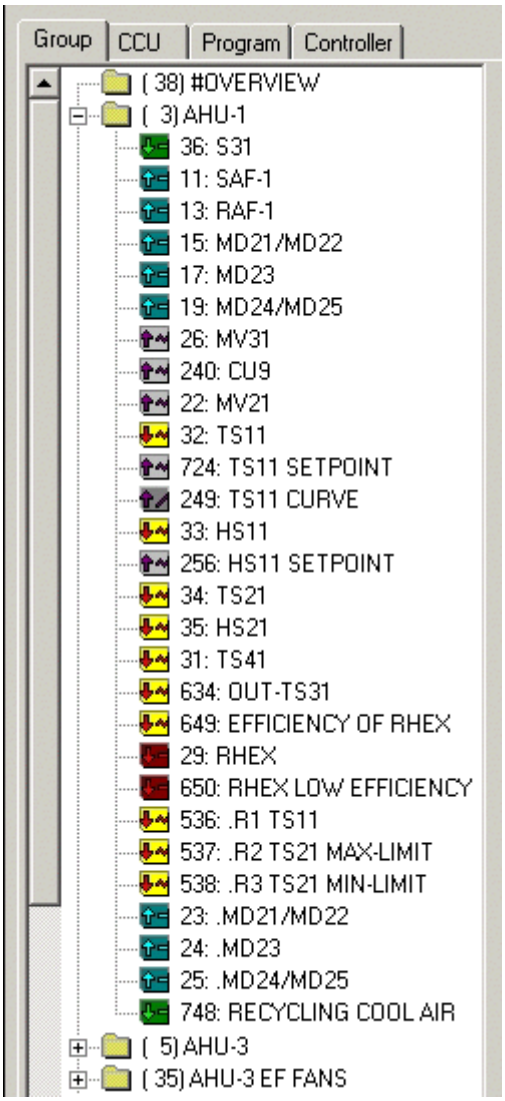
Check CCU

Dialog for checking of the CCU program

Export Settings Registry

Exports settings, loops and dial-up information to the registry.

Group presentation



The group presentation shows the groups and objects of the plant.
Double click on an object for editing.

Objects

Indication

Manages digital in-signals e.g. presence, starting of a pump etc

Alarm

Manages digital in-signals e.g. over-current.

Manoeuvre

Manages digital out-signals e.g. starting of a pump.

Analog in

Measuring of temperature, pressure, relative humidity etc.

Analog out

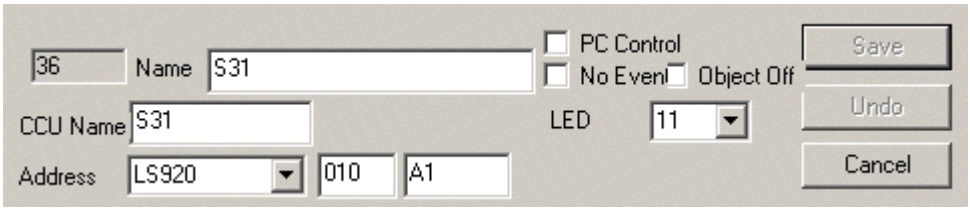
Manages control of dampers and valves. The object can also work as a set point.

Curve

Curve function.

Pulse

Manages pulses from flow- and energy meters.



In a program there are a number of different objects; **Indication**, **Alarm**, **Manoeuvre**, **Analog in**, **Analog out**, **Curve**, **Pulse**. The above part of the object dialog is common to all types of objects.

Name

The name of the object. The same as shown in the PC. Press the **CCU Name** text field to copy the object name to the CCU name. Note that the **CCU Name** text field must be empty before doing this.

CCU name

The name of the object shown in the CCU display, max. 10 characters.

Address

The address consists of an address type and an address. If the address type is **LS920**, an object for the LS920 CCU will be created

Channels

Type of relay card (4 or 7 channels). This is only used when the object is of type **Manoeuvre** or **Analog out**.

LED

Shows which LED is used for indication of the status of operation for the object.

Object Off

Inactivates the object.

No Event

Events from this object are not stored in the **Event List**.

Save

Saves the object.

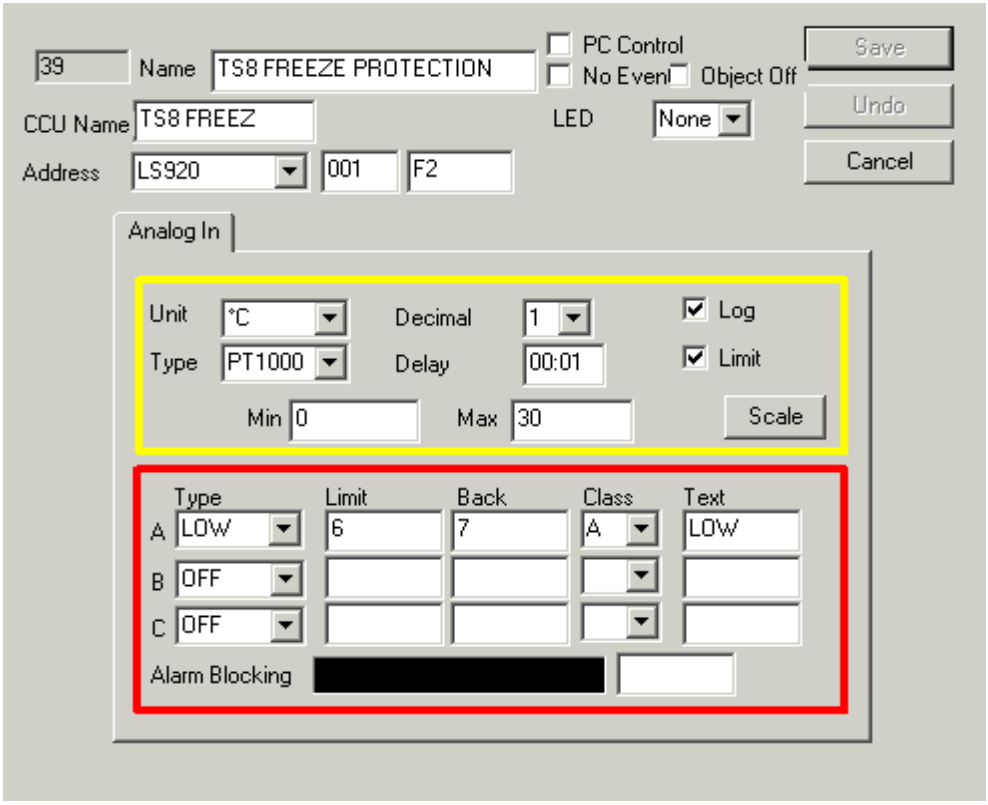
Undo

Restores changes.

Exit

Closes the dialog.

Analog In



An analog in-object is used to present temperature, pressure, relative humidity etc.
The object can have up to three different alarm limits.

See also [Object].

Unit

The unit of the object, for instance °C, meter, second, Pa, Bar and MW.

Dec.

Number of decimals.

Type

Type, for instance PT1000, NI1000, LG1000, 0-10V and 4-20mA etc.

Delay

Time for filtering (the time for the value to reach 63% of the change of the input).

Min

The smallest possible value of the object.

Max

The biggest possible value of the object.

Log

Indicates if the value of the measurement shall be logged on the hard disk or not.

Limit

Indicates if the value of the measurement has alarm limits.

LOW

Low alarm. **Limit**: Value of alarm limit. **Back**: Value for return.

HIGH

High alarm. **Limit**: Value of alarm limit. **Back**: Value for return.

=

Same as. **Limit**: Value of alarm limit. **Back**: Blank.

#

Different. **Limit**: Value of alarm limit. **Back**: Blank.

<>

Alarm if the value is outside the range. **Limit**: Smallest value. **Back**: Biggest value.

><

Alarm if the value is inside the range. **Limit**: Smallest value. **Back**: Biggest value

Class

Alarm class.

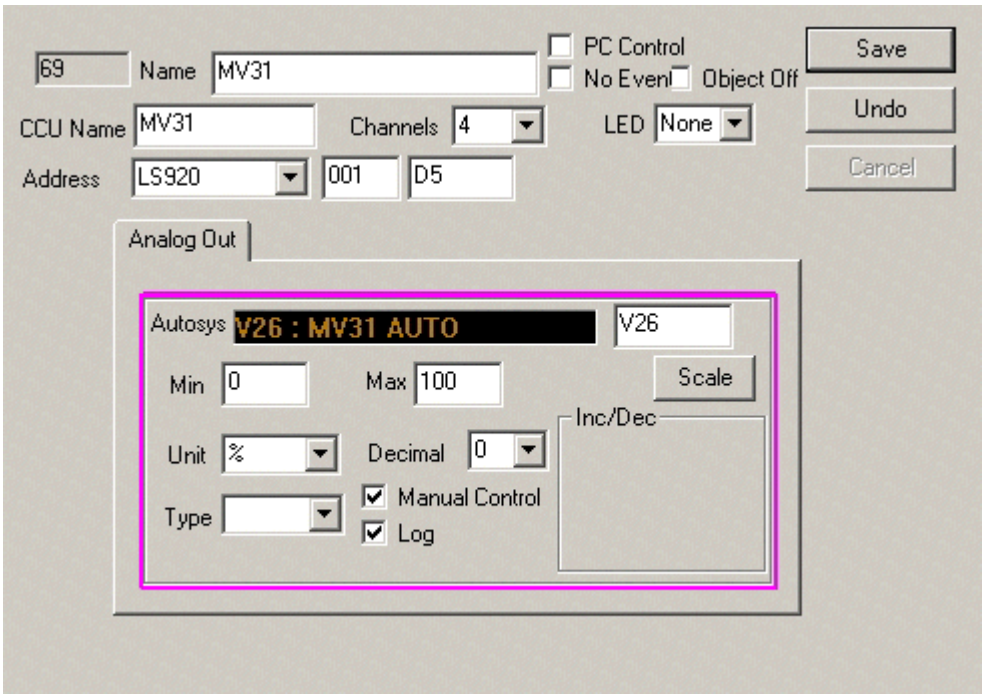
Text

Alarm text.

Alarm blocking

System number (**SYSNO**). If the value is "On" the limit function is inactivated.

Analog Out / Setting



Control of dampers, valves etc. The object can also function as settings (set points, operating times, etc).
See also [Object].

Autosys

System number (**SYSNO**). Connection to function which controls the object in **AUTO** mode.

Min

The value of the object at 0 V.

Max

The value of the object at 10 V.

Unit

The unit of the object, for instance %, °C, meter, second, Pa, Bar, MW etc.

Decimals

Number of decimals.

Manual control

Click this to enable manual controlling of the object.

Log

Indicates if the value of the measurement shall be logged on the hard disk or not.

Type

0-10V, 2-10V, Increase/Decrease etc.

When the analog out object is of the type Increase/Decrease, it is used for controlling of motors of such type. The object uses two pcs of relays, one relay for increase and one for decrease. For this type there are another three values.

Adr (Decrease)

Address for decreasing of relay.

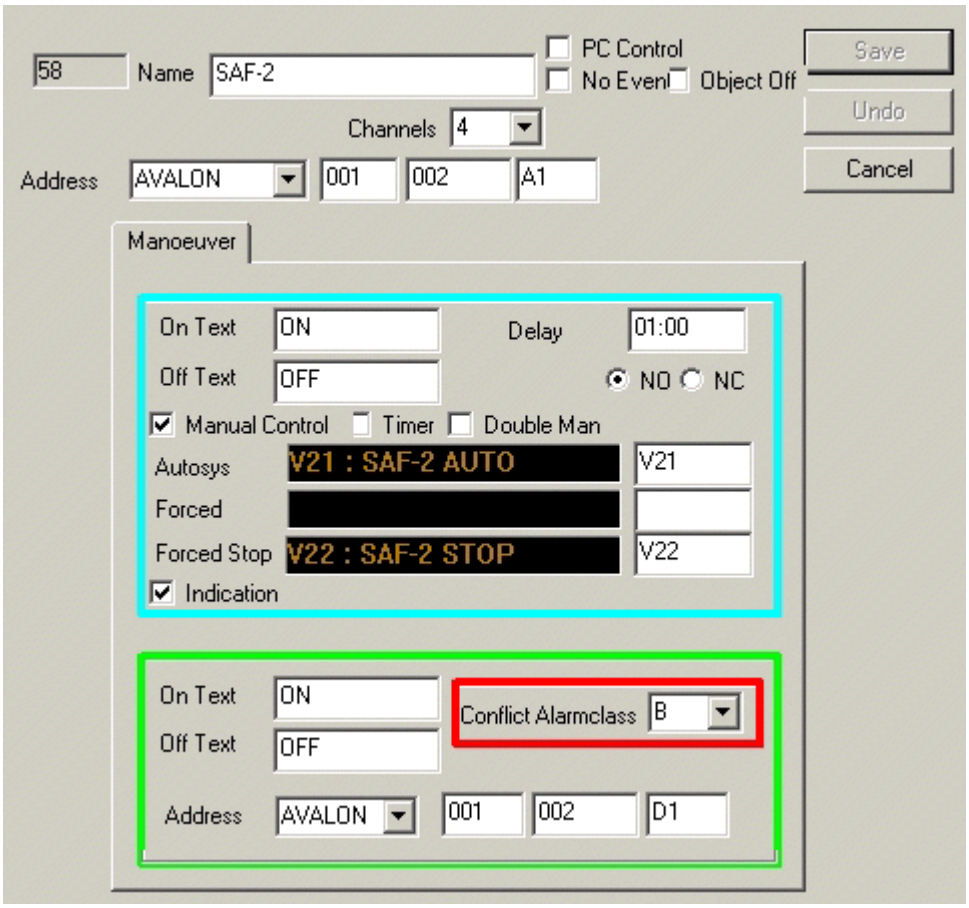
Motor time

Indicates the operating time in seconds from the extreme positions.

LED

Light emitting diode (decrease).

Manoeuvre



A Manoeuvre is used for starting/stopping of for instance fans and pumps, turning on/off of lighting etc.
A manoeuvre-object can be connected to a relay-output on the LS920 CCU.
If there is an operation response, '**With Indication**' can be used. The indication can be a digital input on a module or a calculated value from a Condition.

When the status of a manoeuvre and an indication doesn't correspond, a conflict alarm is alerted after a delay which is indicated by '**Delay**'.

See also [Object].

On text

The status name of the object when 'ON'.

Off text

The status name of the object when 'OFF'.

Manual control

Click this to enable manual controlling of the object.

Timer

When this is marked, the object work as a timer. If the manoeuvre-object is put in manual mode 'ON', the object will go 'OFF' after the time indicated by the delay.

Double manoeuvre

When this is marked, the ATLANTIS program is asking one extra time if You want to put the object in manual mode.

NO/NC

Normal Open / Normal Close.
When the object is in NO, the indication goes 'ON' when there is a signal on the digital input.

Delay

Delay of conflict alarm.

Auto connection

System number (**SYSNO**). Indicates how the object is controlled in AUTO mode (Priority 4). (Manual mode has Priority 3).

Forced start

System number (**SYSNO**). When this input is 'ON' the object goes 'ON' (Priority 2).

Forced stop

System number (**SYSNO**). When this input is 'ON' the object goes 'OFF' (Priority 1).

With Indication

Shows if the Manoeuvre-object has an indication.

On text

The status name of the object when 'ON'.

Off text

The status name of the object when 'OFF'.

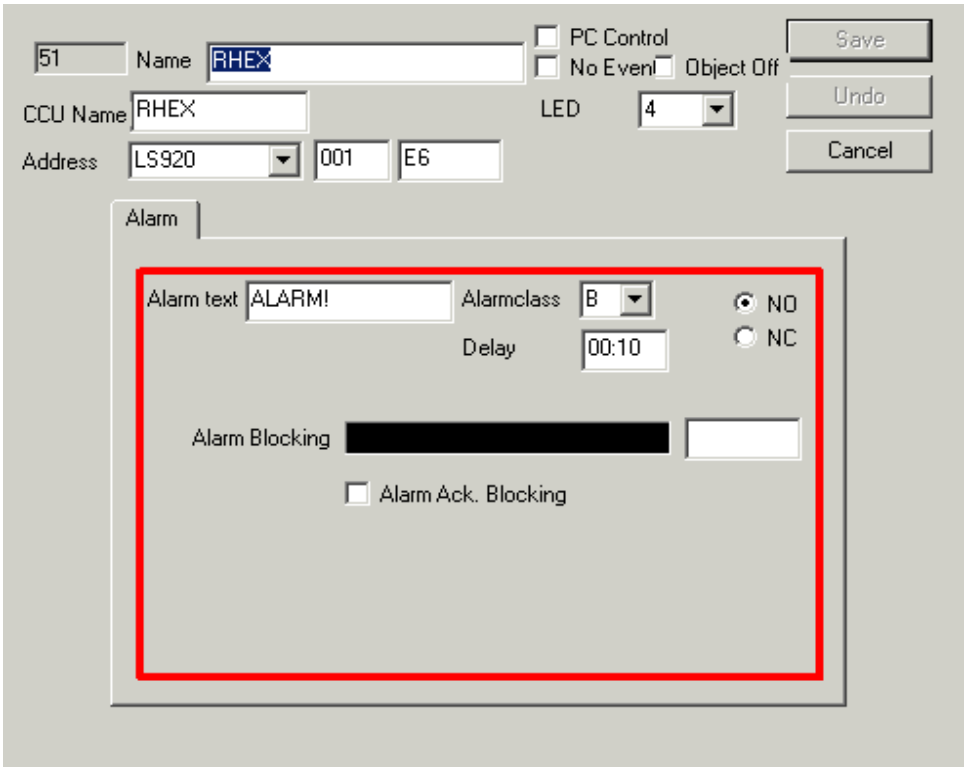
Conflict alarm class

Alarm class of the conflict alarm.

Address

The address of the indication.

Alarm



An Alarm object is used for collecting of digital signals from for instance lift alarms, over-current, etc.

See also [Object].

Alarm text

The status name of the object when 'ON'.

Alarm class

The status name of the object when 'OFF'.

Delay

Operation delay.

NO/NC

Normal Open / Normal Close.
When the object is in NO, the indication goes 'ON' when there is a signal on the digital input.

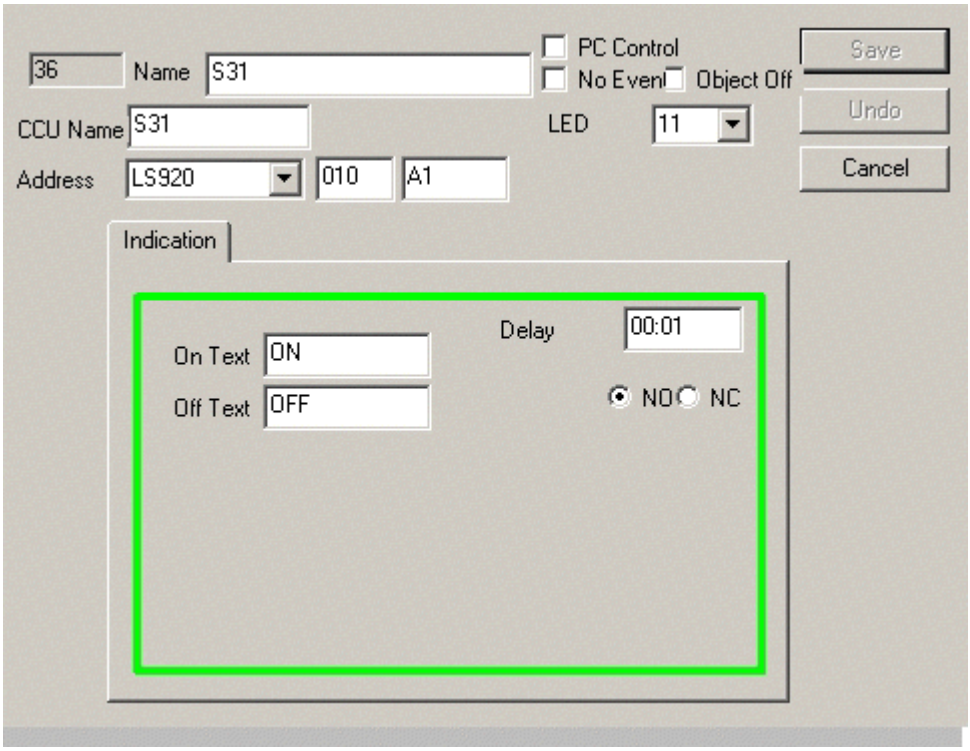
Alarm blocking

System number (**SYSNO**). Blocking of alarms. Can for instance be at startup sequences.

Acknowledgement blocking

Blocking of acknowledgement from the PC.

Indication



An Indication is used for collecting of digital signals from for presence detectors, operation indicators etc.
See also [Object].

On text

The status name of the object when 'ON'.

Off text

The status name of the object when 'OFF'.

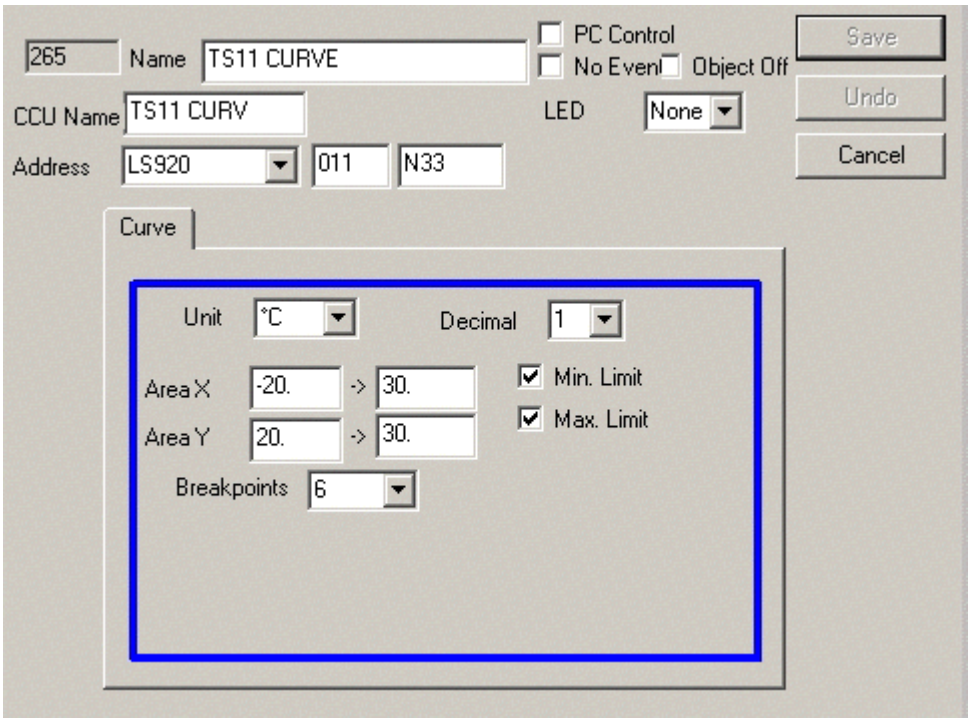
Delay

Operation delay.

NO/NC

Normal Open / Normal Close.
When the object is in NO, the indication goes 'ON' when there is a signal on the digital input.

Control Curve



A curve is used when applying a value depending on another object or condition according to a given curve. In the editing software the range and number of breakpoints are indicated. The curve is defined in the operation program Atlantis.

See also [Object].

The address of a curve must contain a system number (**SYSNO**).

Unit

The unit of the object is for instance °C, meter, second, Pa, Bar and MW.

Decimals

Number of decimals.

Range X

The scale of the X-axis.

Range Y

The scale of the Y-axis.

Breakpoints

Number of breakpoints.

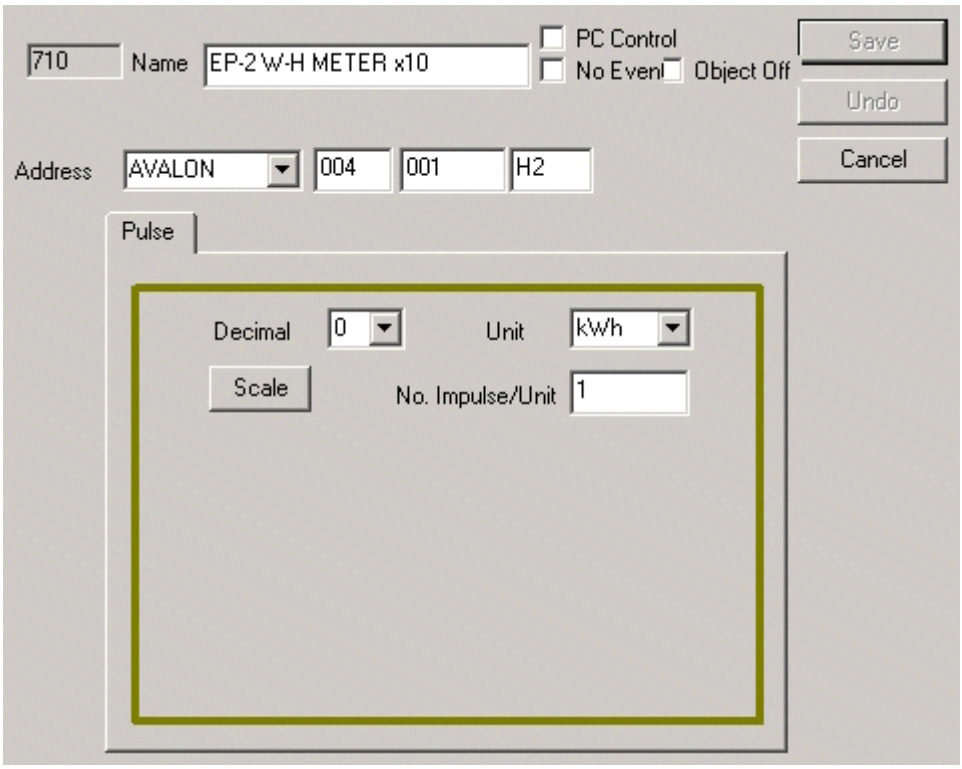
Min limited

Min. limitation. The value is not below the indicated min. value on the Y-axis.

Max limited

Max. limitation. The value is not below the indicated max. value on the Y-axis.

Pulse



The object manages pulses from e.g. flow- and energy meters.

See also [Object].

Decimals

Number of decimals.

Unit

The unit of the object, for instance MWh, KWh etc.

Pulses/Unit

Number of pulses/unit.

Address Types

Address List

LS99

xx yyy zz

xx - loop
yyy - unit
zz - address

PC

000 X

*X - **System Number (SYSNO)***

LS920

xxx Y

xxx - CCU number
*Y - addres of terminal block or **System Number (SYSNO)***

AVALON

xxx yyy Z

xxx - CCU number
*yyy - base module number, 000 if **System Number (SYSNO)***
*Z - address of terminal block or **System Number (SYSNO)***

OPC

x:Y

x - OPC server number
Y- address name

MODBUS

xx yyy zzzzzz

xx - loop
yyy - unit
zzzzzz - address

Modbus

1. **Edit the objects**

   Use the Modbus address type in the **Edit** software.
   This address type has three fields, *xx-yy-zzzzz*, where:

xx	Loop number
yy	Unit number
zzzzz	Modbus address

2.

3. **Create file for address polling**

   The file must be located in the **Larmia.edt** folder. The file must also be named **ModbusX.ini**, where the **X** is the number of the loop.
   This file states which addresses to be polled, what kind of values located at these addresses and how many values to be fetched.

   The file contains the following:

TYPEn	1 = Digital Out, 2 = Digital In, 3 = Analog Out, 4 = Analog In
STARTADRn	Start address
ANTn	Number of values
IDn	Unit number

   Example:

   1. Digital Out
   TYPE4=1
   STARTADR4=512
   ANT4=32
   ID4=1

   2. Digital In
   TYPE3=2
   STARTADR3=0
   ANT3=32
   ID3=1

   3. Analog Out
   TYPE2=3
   STARTADR2=512
   ANT2=32
   ID2=1

   4. Analog In
   TYPE1=4
   STARTADR1=0
   ANT1=32
   ID1=1

4. **Configure the driver**

   Add a new driver:
   LCDModbus.EXE COMz x

   Where **z** is the number of the COM port used, and **x** is the number of the loop.

See also [Adress List]

OPC Address

In the OPC address field you first enter the number of the designated OPC server (1-10) followed by a **:** (colon). Then enter the name of the point which is to be addressed. The number of the OPC server is read from the **OPC Server List**.

To read/write indexed values from an array you must use brackets (**[ ]**). The values start at index 1.
To read/write indexed bits from a value you must enter **:BITx** after the name, where **x** is the number of the bit to be read/written.

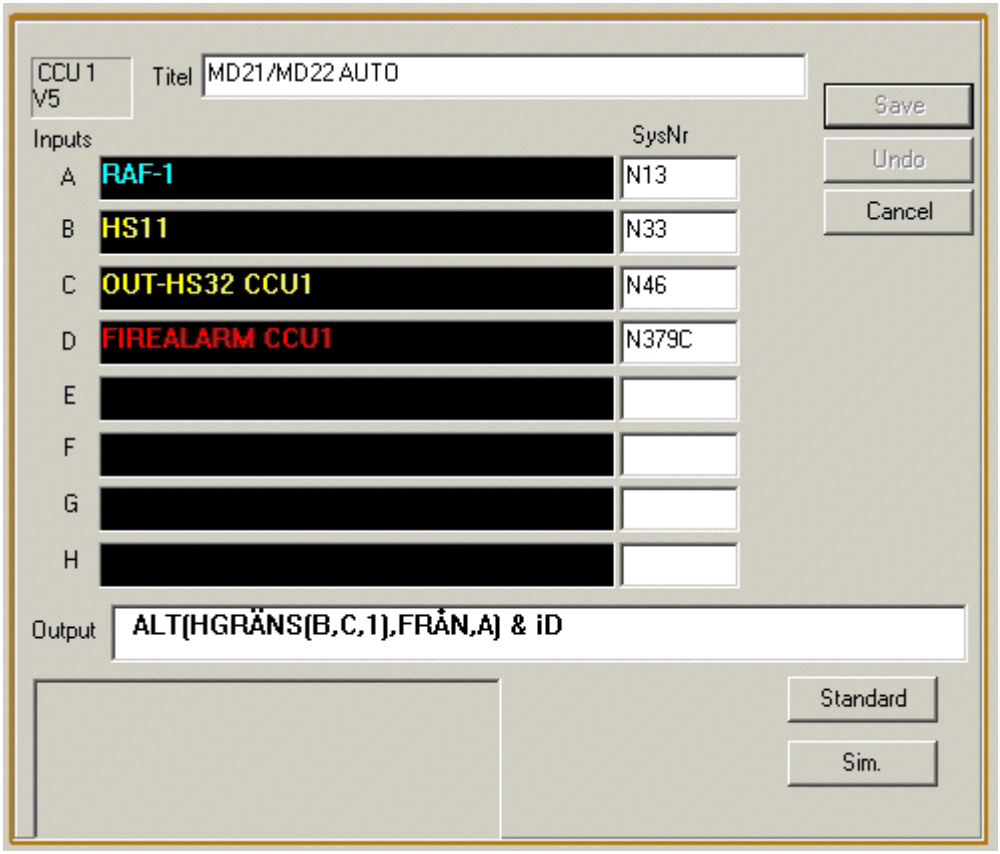
Example:

1:DUC1.MV1	OPC Server 1, OPC Value DUC1.MV1
2:DUC1.SETV[4]	OPC Server 2, Index 4 in the OPC Value DUC1.SETV array
2:DUC1.FRC:BIT5	OPC Server 2, Bit 5 in OPC Value DUC1.FRC

NB! The address name is unique for every OPC server.

See also [Address list]

Conditions



Conditions are used for formulating of the control function. In the language for conditions are used **Operators**, **Condition functions**, **Macro**, **System Numbers (SYSNO)** and **System Functions**.

Title

The name of the condition.

Inputs A-H

System Number (SYSNO).
Objects, **Time Schedules**, **Regulators** and Conditions used in the expression.

Output

Expression for condition.

Expressions for Conditions

See also [System Number (SYSNO)] and [System Functions].

Operators

`iX`

Not X
Priority: **1**

`X=Y`

X equal to Y
Priority: **4**

`X#Y`

X not equal to Y
Priority: **4**

`X>Y`

X bigger than Y
Priority: **4**

`X<Y`

X smaller than Y
Priority: **4**

`X^Y`

Only X or only Y
Priority: **5**

`X&Y`

X and Y
Priority: **6**

`XeY`

X or Y
Priority: **7**

`X*Y`

X multiplied by Y
Priority: **2**

`X/Y`

X divided by Y
Priority: **2**

`-X`

Minus X
Priority: **3**

`X+Y`

X plus Y
Priority: **3**

`X-Y`

X minus Y
Priority: **3**

Condition Functions

`MIN(X,Y)`

Smallest value of X and Y

`MAX(X,Y)`

Biggest value of X and Y

`ABS(X)`

Absolute value of X

`ROT(X)`

The square root of X

`LOG(X)`

The logarithm of X

`SIN(X)`

Sine of X (degrees)

`COS(X)`

Cosine of X (degrees)

`TAN(X)`

Tangent of X (degrees)

`ASIN(X)`

Arcsine of X (degrees)

`ACOS(X)`

Arccosine of X (degrees)

`ATAN(X)`

Arctangent of X (degrees)

`V0`

Initial value of the **Conditon** (the same condition) for feedback.

`V(X)`

The value of the **Conditon** X. X is an integral number 1->60.

`T(X)`

The status of the **Time Schedule** X. X is an integral number 1->40.

`""`

The text between these signs is not used in the calculation. (Comment).

`MINNE(S,R)`

Memory function

S=1,R=0 : result = 1, sets the memory = 1. (Set)
S=0,R=1 : result = 0, clears the memory = 0. (Reset)
S=0,R=0 : result = memory
S=1,R=1 : result = 1.

TID(M,X)

Operation delay: 1 sec – 10 years +- 1%.

Delays operation X, M minutes.
Ex:
TID(10/60,A) delays operation input A, 10 seconds. (1 hour = 60 min, 24 hours = 1440 min).

FTID(M,X)

Disconnecting delay.

As TID(M,X) but delay of disconnection.

FILTER(S,X)

Filters the value X, with the time constant S(seconds).

Ex:
FILTER(4,A) filters input A in such a way that if A is changed from 0 to 10, the value is changed (acc. to a curve) to half the change (5) after 4 seconds, 7.5 after 8 seconds and so on.

ALT(A,X,Y)

Alternative choice.

A=1 : result = X
A=0 : result = Y

PULSR(U,N,R,S,A)

Pulse counter.

U goes from 1 to 0 : result increased by 1 (Adjustment upwards).
N goes from 1 to 0 : result decreased by 1 (Adjustment downwards).
R = 1 : result = 0 (Reset).
S = value (and R = 0) :result = value (Set).

If A (number) is given, revolution counting takes place (A pcs per rev.). A = a positive whole number. The result is limited between 0 and A-1. When the result otherwise should have been A it will be 0. When the result otherwise should have been -1 it will instead be be A-1.

Ex:
PULSR(A,0,B,C*4). A adjusts upwards, adjustment downwards is not used, B sets the counter to zero, C sets the counter to 4, revolution counting is not used.

KURVA(X,type,x1,y1,x2,y2...)

Curve function with up to 6 breakpoints.

X = The entry value of the curve (the actual x-value of the curve).
This can be an input or another expression in the condition.

Type 1 = The curve continues before the first and after the last breakpoint.

Type 2 = The curve gets the same value as y1 before the first breakpoint and continues after the last breakpoint.

Type 3 = The curve continues before the first breakpoint and gets the same value as the last y after the last breakpoint.

Type 4 = The curve gets the same value as y1 before the first breakpoint and gets the same value as the last y after the last breakpoint.

TIDOPT(N,T,S,I,B,O,V,F)

Function for optimization of operation start time.

This function returns the amount of minutes (positive integer) the heating must start ahead of the designated time, if the temperature is to reach it's set point.
This value can, for example, be used as a start delay of a **Time Schedule**. If so, the value has to be negative.

N = Number (1-5).
This number must be unique for every usage of this function in a CCU..

T = The inertia of the building (1-10).
This number is used for setting the historical time period used for calculating the mean values of the indoor and outside temperatures.
A small number = short historical time period.
A big number = long historical time period.

S = Start function.
This parameter should be 'ON' when the heating is started.
This parameter should be 'OFF' when the temperature firstly has reached it's set point.
If the temperature doesn't reach it's set point, this parameter should be 'OFF' when the heating period has expired.

I = Indoor temperature.

B =Indoor set point.

O = Outside temperature.

V = Wind speed in m/s.

F = The relative humidity %

LGRÄNS(G,X,H)

Low limit (H = hysteresis, choose normally H = 1).

X smaller than G : result = 1 (limit 'on').
X bigger than G+H : result = 0 (limit 'off ').
X between G and G+H : result = unchanged (hysteresis).

HGRÄNS(G,X,H)

High limit (H = hysteresis, choose normally H = 1).

X bigger than G : result = 1 (limit 'on').
X smaller than G-H : result = 0 (limit 'off ').
X between G and G-H : result = unchanged (hysteresis).

BEGR(L,H,X)

Limits the value of X between L (Low) and H (High).

X bigger than H : result = H.
X smaller than L : result = L.
X between L and H : result = X.

Ex:
BEGR(0,100,A), result of input A is limited between 0 and 100.

VIPPA(X)

Latch function.

Result = 0, and X goes from 1 to 0 : result = 1.
Result = 1, and X goes from 1 to 0 : result = 0.

Ex:
VIPPA(A), if input A is an external spring push button, the result is 0 every second time and 1 every second time the button is pushed

INTER(M,X)

Intermittent operation (M>1).

During the time period M minutes the result is 1(ON) in time corresponding to the value of X in percent . (If X<2% the result is 0, if X>98% the result is 1).

Ex:
A set point 0-100% (input A) controls an el. radiator (on/off) with intermittent operation. Timeperiod = 10 minutes. The condition of the el. radiator is INTER(10,A).
If for example A = 70%, the result will be 'on' for 7 minutes and 'off' for 3 minutes. This will be repeated all the time.

FRYS(G,D,T,R)

Anti-freeze protection function.

If the temperature T approaches the limit G, the result is gradually control by force up to 100%. The forced control starts from 0% at T = G + D, (D = Difference). The forced control is 100% at T = G. (R = Regulator).

Ex:
Air handling unit. Input A = Set alarmlimit, B = Temp. heating coil, C = Regulator step for the hot water valve. The condition for the hot water valve: FRYS(A,4,B,C). The forced control starts 4 degrees before the alarm limit.

VX(A,M)

Change of operation (M>1).

Change of operation with an interval of M minutes. A = Number of changes. A = 2 for twin pump. The result has the value 0 or 1.

Ex:
Four chillers: A = 4. The result is whole numbers from 0 to 3.
This function can be used together with TVILLP(V,T,M,F) and STEGVX(A,N,V,R).

STEG(A,S,R)

Stepping function.

A control signal (R) 0-100%. Ex.: A regulator is divided in different digital steps. A = Number of steps. S = The step number, numbered from 1 and upwards.

Ex:
Four chillers. Input A = Regulator.

Chiller no. 1: STEG(4,1,A).
Chiller no. 2: STEG(4,2,A).
Chiller no. 3: STEG(4,3,A).
Chiller no. 4: STEG(4,4,A).

STEGVX(A,S,V,R)

Stepfunction with change of operation. Functions as STEG(A,S,R), but with change of operation. V = Conditions for the change. Use the function VX(A,M) for change.

TVILLP(V,S,M,F)

Twin pump function.

V = Change of operation. S = Start condition for operation. M = Condition for motion. F = Fault second pump. As change of operation the function VX(2,M) can be used. S starts pump if V is 'on'. If F is 'ON' S starts pump irrespectively of V. M always starts pump.

Ex:
Twin pump P1. Input A = Change condition, B = Start condition, C = Exercise condition, D = E.g. over-current P2.
Conditions of P1: TVILLP(A,B,C,D).

Twin pump P2. Input A = Change condition, B = Start condition, C = Exercise condition, D = E.g. over-current P1.
Conditions of P2: TVILLP(iA,B,C,D). If Exercise- and/or Fault function not applicable, put 0.

VGRAD(T1,T2,T3)

Calculates efficiency of recovery unit.

Result = 100*(T2-T1)/(T3-T1).
T = Temperatures.T1 = Supply air, T2 = Supply air after heat recovery unit (HRU), T3 = Return air before HRU.

STEGD(A,R)

Step dividing function.

Functions as STEG(A,S,R), but S (The step number) is omitted. Instead the result will be = number of connected steps.

Ex:
Input A = Regulator, STEGD(15,A) has a value from 0.0 to 16.0. This function can be used together with BSTEG1-4(S).

BSTEG1-4(S)

Dividing of steps into binary step connection.

For control of for instance 4 pcs el. heaters with the capacity of 1KW, 2kW, 4kW and 8kW.
Control gives 15 steps of capacity from 1 to 15kW and complete disconnection (0kW).

Use the function STEGD(15,R) where R = Regulator, in a condition e.g. V1, in order to divide the 0-100% of the Regulator into 0-15 steps.

For control of each step, input A = V1.
1kWstep = V2eV3, has to be divided into two conditions V2=BSTEG1A(A),V3=BSTEG1B(A).
2kWstep = BSTEG2(A), 4kWstep = BSTEG3(A)..

When 3 capacity steps are needed, STEGD(7,R) is used.
When 2 capacity steps are needed, STEGD(3,R) is used.

PULS(TM,FM)

'ON' = TM minutes, 'OFF' = FM minutes.
Time in seconds can be written as sec/60, e.g. 10/60 = 10 sec.
(PULS cannot be combined with other expressions in a condition).

MPULS(M)

Pulse on/off with the period M minutes. (M>1).

Macro definitions (supplementary):

The functions LGRÄNS(G,X,H) up to and including MPULS(M) above, are so called macros. That means that they consists of different basic functions. The definitions can be interesting to know, for instance when making similar functions by Yourself.

LGRÄNS(G,X,H)

MINNE(X<G,X>G+H)&STAB

HGRÄNS(G,X,H)

MINNE(X>G,X<G-H)&STAB

VIPPA(X)

PULSR(X,0,0,0,2)

INTER(M,X)

SEK+PULSR(SEK>30,0,0,0,M)*60<(X-2)*0.624*X

FRYS(G,D,T,R)

MAX(R,KURVA(T,4,G+D,0,G,100))

VX(A,M)

PULSR(PULSR(SEK>30,0,0,0,M),0,0,0,A)

STEG(A,S,R)

(A+1)*0.01*R>S

STEGVX(A,S,V,R)

(A+1)*0.01*R>ALT(S+V>A,S+V-A,S+V)

TVILLP(V,T,M,F)

(VeF)&TeM

VGRAD(T1,T2,T3)

100*(T2-T1)/(T3-T1)

MPULS(M)

PULSR(SEK>30,0,0,0,M)>=M/2

PULS(TM,FM)

TID(FM,iTID(TM,V0))

STEGD(A,R)

(A+1)*0.01*R

BSTEG1A(S)

S>1&S<2eS>3&S<4eS>5&S<6eS>7&S<8

BSTEG1B(S)

S>9&S<10eS>11&S<12eS>13&S<14eS>15

BSTEG2(S)

S>2&S<4eS>6&S<8eS>10&S<12eS>14

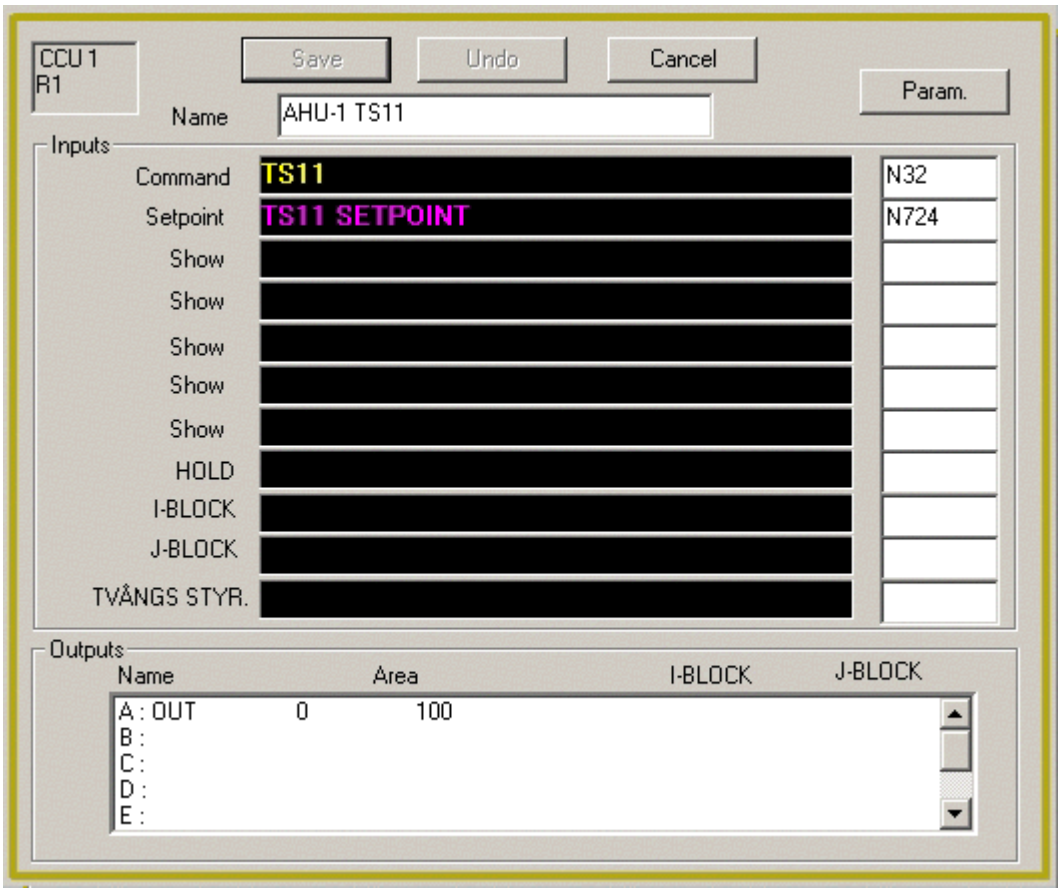
BSTEG3(S)

S>4&S<8eS>12

BSTEG4(S)

S>7

The Regulator



Regulators

Title

The name of the regulator.

Measurement

System Number (**SYSNO**). The actual measurement of the regulator.

Set point

System Number (**SYSNO**). The set point of the regulator.

Show 1-5

System Number (**SYSNO**). Values shown in the regulator menu in the CCU.

HOLD

System Number (**SYSNO**).
When this value is 'ON', the regulator stops regulating and holds it's value.

I-Block

System Number (**SYSNO**).
When this value shows 'ON' the output is forced to the value of I-Block (Outputs).

J-Block

System Number (**SYSNO**).
When this value shows 'ON' the output is forced to the value of J-Block (Outputs).

Parameters

Regulator parameters.

Inputs

Measurement

System Number (**SYSNO**). The actual measurement of the regulator.

Set point

System Number (**SYSNO**). The set point of the regulator.

Show 1-5

System Number (**SYSNO**). Values shown in the regulator menu in the CCU.

HOLD

System Number (**SYSNO**).
When this value is 'ON', the regulator stops regulating and holds it's value.

I-Block

System Number (**SYSNO**).
When this value shows 'ON' the output is forced to the value of I-Block (Outputs).

J-Block

System Number (**SYSNO**).
When this value shows 'ON' the output is forced to the value of J-Block (Outputs).

Forced control

The function adds the following functions:

1. Indicates at which value the regulator shall start after a restart.
2. You can at anytime force the regulator to the value from which it shall regulate.

If this value is a figure between 0-100, the value indicates the starting value for the regulator after a restart of the CCU. (In the older versions the regulator starts at 100%).

If You write a System Number (**SYSNO**) in this field instead of a number, e.g. V1, N345, the regulator will regulate as normal if the value of the System Number is –1.
If the value of the System Number is positive, the regulator is set to this value.
When You return to –1, the regulator will continue to regulate from this value.

Ex 1.
If You enter a constant, e.g. 70, in the **Forced control** field, the regulator starts from this value after a restart. The regulator then regulates as normal.

Ex 2.
If You enter a **Condition**, e.g. V5, in the **Forced control** field, where:

Condition 5:
Input A: Manoeuvre N432
Output: ALT(A,70,-1)

The regulator will by force be set to the value 70 if the manoeuvre is 'ON' and continue to regulate when the manoeuvre is 'OFF'.

Ex 3.
If You enter an object number, e.g. N45, in the **Forced control** field, where object number 45 is a set point with it's auto connection value set to –1.

When You set the object in auto mode the regulator will regulate as normal.

When You enter a manual position between 0-100, the regulator will have this value. When You return to auto mode, the regulator will continue from this value.

Ex 4.
When You start up a speed controlled fan, You want to gradually approach a certain speed and after that connect the regulator. The following condition gradually approach up to 70% after the condition has got an approx. 2 sec. start pulse.
After that the regulator will regulate.
(This condition has not been tested, but should work).

Condition:
Input A: Start pulse manoeuvre.
Output: ALT(A,0,ALT(V0>=0) e (V0<=70),V0+1,-1).

Outputs

When the regulator shall be divided into steps, outputs are used (8 pcs A-H). Double click the output which You want to change, and the following dialogue shows up:

Name

The name of the output.

Min

Value of the regulator when the output shall be 0%.

Max

Value of the regulator when the output shall be 100%.

I-Block

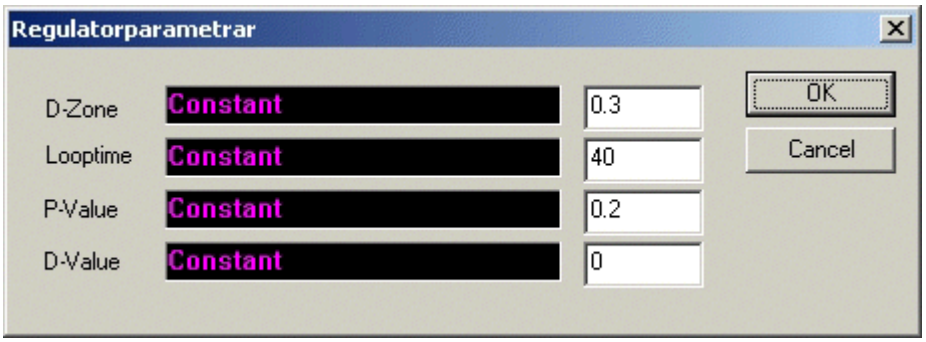
Value of the output when I-Block (Inputs) are 'ON'.

J-Block

Value of the output when J-Block (Inputs) are 'ON'.

NB! If both I- and J-Block are 'ON', the value of the J-Block is returned.

Regulator Parameters



Dead band

The error tolerance of the regulator between measurement and set point.

Loop time

Indicates how often the regulator makes a new calculation (seconds).

P Action

Indicates the size of the change the regulator shall do in relation to the regulation error.

D Action

Indicates the size of the change the regulator shall do when the measurement is changing.

System Functions

Functions can be used everywhere where **System Numbers (SYSNO)** can be used.
Example:
A digital control in a CCU can be edited with auto connection "LARM1". Then the relay will close when there is any class 1 alarm in the CCU.

Most of the system-functions are to be finding both in the PC and the CCU. The system-functions which are to be find only in the PC or in the CCU, are here marked with (PC) or (CCU).

ALARM FUNCTIONS

LARM

Remaining alarm.

LARM0–LARM6

Remaining alarms class 0-6.

KV

Unacknowledged alarm.

KV0–KV6

Unacknowledged alarm class 0-6.

LOK

Remaining and unacknowledged alarm.

LOK0–LOK6

Remaining and unacknowledged alarm class 0-6.

LEK

Remaining or unacknowledged alarm.

LEK0–LEK6

Remaining or unacknowledged alarm class 0-6.

NLARM

About 10 sec. pulse at every new alarm.

NLARM0–NLARM6

About 10 sec. pulse at every new alarm class 0-6.

GENERAL / TECHNICAL FUNCTIONS

TILL

Always on.

FRÅN

Always off.

PI

3.14

STAB
(CCU)

On when analog in objects has stabilized after a power failure.

KDxxx
(PC)

Communication error with CCU xxx

SDxxx
(PC)

Status in CCU xxx
0 = CCU has received program.
1 = Transmission with restart to CCU is requested.
3 = Transmission without restart to CCU is requested.

When loading a CCU of type Avalon this value counts down to zero.

When loading a CCU of type LS920 this value indicate which block (out of 54 blocks) is being transmitted to the CCU. A positive value indicates that a block is being transmitted to the CCU whilst a negative value indicates that the block has been acknowledged by the CCU.

HDISK
(PC & Avalon)

Available space on the hard drive in megabytes.

PC

Always on in the PC. Always off in the CCU.

DUC

Always on in the CCU. Always off in the PC.

LOGGT
(LS920)

Remaining available log time of the curve for storing in the CCU.

SENDD
(PC)

Transmission in progress (SEND) to CCU, (value = CCU number).

GENERAL / TECHNICAL FUNCTIONS

MAN

On when something is in manual position.

MANM

On when a digital control is in manual position.

MANU

On when an analog out object is in manual position.

MANS

On when a setting is in manual position.

MANT

On when a Time Schedule is in manual position.

KOD
(PC)

On when the system is not locked (someone has logged on).

KOD1-KOD5
(PC)

On when log in to level 1-5 or higher in the system has been done.

IR
(Avalon 64CE)

On when the Avalon CCU detects someone in it's vicinity.

PCCONN
(Avalon)

On when a PC is connected to the CCU.

CCU ERRORS

GFELR
(CCU)

Sensor failure (resistive sensors).

DDKOM
(CCU)

CCU to CCU communication error in the receiving object.

BATT
(LS920)

Battery failure.

MKOM
(AVALON)

Communication error with the base module. The function returns the number of objects not connected.

DUCFEL
(CCU)

Sum error in the CCU (BATT, GFELR, DDKOM, MKOM),

1 Battery failure
2 CCU to CCU communication error
3 Battery failure & CCU to CCU communication error
4 Sensor failure
5 Battery failure & sensor failure
6 CCU to CCU communication error & sensor failure
7 Battery failure & CCU to CCU communication error & sensor failure
8 Base module communication error
9 Base module communication error & Battery failure
10 Base module communication error & CCU to CCU communication error
11 Base module communication error & Battery failure & CCU to CCU communication error
12 Base module communication error & sensor failure
13 Base module communication error & Battery failure & sensor failure
14 Base module communication error & CCU to CCU communication error & sensor failure
15 Base module communication error & Battery failure & CCU to CCU communication error & sensor failure

CALENDAR FUNCTIONS

ÅR

Year (4 digits).

MÅNAD

Month 1-12.

DAG

Day 1-31.

DATUM

Year Month Day (6 digits).

VDAG

Weekday 1-7 (Monday - Sunday).

VARD

'ON' on weekdays.

MÅN

'ON' on Mondays.

TIS

'ON' on Tuesdays.

ONS

'ON' on Wednesdays.

TORS

'ON' on Thursdays.

FRE

'ON' on Fridays.

LÖR

'ON' on Saturdays.

HAFT

'ON' on Eves.

SÖN

'ON' on Sundays.

HELG

'ON' on holydays.

SPEC1

'ON' on special day 1.

SPEC2

'ON' on special day 2.

TIME FUNCTIONS

TIMME

Hour 0-23.

MINUT

Minute 0-59.

SEK

Second 0-59.

KLOCKA

Hour Minute Second (6 digits).

ASTRONOMICAL FUNCTIONS

SOL

On when the sun is up.

SOLH

Altitude of the sun in degrees (-90 to 90).

SOLR

Direction of the sun in degrees (0 - 360).

See also [Edit] [Conditions] [Regulators]

System Number (SYSNO)

A system number (abbr. SYSNO) is a connection number to another object, Time Schedule, condition, regulator, system function or a constant numerical value. A SYSNO can be a digital on/off status or a numerical value. Only objects can be connected between different CCU and PC, and the objects have individual numbers in the whole system. For other SYSNO's, for instance conditions, are valid: PC-conditions can only be used in the PC, conditions for CCU 1 can only be used in CCU 1 etc.

NB!
Any SYSNO controls of other subcentrals than CCU is controlled by the PC.

The SYSNO consists of six characters or less. The SYSNO is written in the following way:

1. If the status is going to be inverted, the first sign shall be "**i**".
2. Types of SYSNO.

N	Object (N can be omitted if inverted)
T	Time Schedule
V	Condition
R	Regulator
A number	The numerical value of the number. Numbers with decimals are written with a point.
The name of a system function	Corresponding system function.

3.
4. A list of names of system functions and description is available in ***Menu*** / ***System Functions***.
5. The number of a Time Schedule, condition or object.
   The object number is the number, which you can see to the upper left in the window of the object.
6. Additional letter (option), a letter for an alternative function.

- For ALARM	
None:	Status = 1 when remaining alarm.
A:	Status = 1 when unacknowledged alarm.
B:	Status = 1 when unacknowledged remaining alarm.
C:	Status = 1 when unacknowledged or remaining alarm.

7.

- For DIGITAL CONTROL (MANOEUVRE)	
None:	Status = 1 when digital control on.
I:	Status = 1 when indication on.
L:	Status = 1 when remaining alarm.
A:	Status = 1 when unacknowledged alarm.
B:	Status = 1 when unacknowledged remaining alarm.
C:	Status = 1 when unacknowledged or remaining alarm.
T: (Avalon)	Timer countdown in minutes
M: (Avalon)	Status = 1 when in Manual mode

8.

- For ANALOG IN	
None:	The analog value.
A:	Status = 1 when limit A is on or at remaining alarm.
B:	Status = 1 when limit B is on or at remaining alarm.
C:	Status = 1 when limit C is on or at remaining alarm.
D:	Status = 1 at limit A unacknowledged alarm.
E:	Status = 1 at limit B unacknowledged alarm.
F:	Status = 1at limit C unacknowledged alarm.
G:	Status = 1 at limit A unacknowledged remaining alarm.
H:	Status = 1 at limit B unacknowledged remaining alarm.
I:	Status = 1 at limit C unacknowledged remaining alarm.
J:	Status = 1 at limit A unacknowledged or remaining alarm.
K:	Status = 1 at limit B unacknowledged or remaining alarm.
L:	Status = 1 at limit C unacknowledged or remaining alarm.
M:	Preset limit value A.

N:	Preset limit value B.
O:	Preset limit value C.
X:	Status = 1 at sensor failure.

9.

- For ANALOG OUT	
None:	The analog value.
A: (Avalon)	Status = 1 when the increase relay is on (incr. / decr.)
B: (Avalon)	Status = 1 when the decrease relay is on (incr. / decr.)
M: (Avalon)	Status = 1 when in Manual mode.

10.

- For PULSE	
None:	The pulse value.

11.

- For REGULATOR	
None:	The value of the regulator 0-100%, (before divided into steps).
A-H:	Corresponding output steps.

12.

- For TIME SCHEDULE	
None:	The value of the Time Schedule (1 or 0).
A: (Avalon)	The status of the Time Schedule without start or stop delay.
B: (Avalon)	Returns the value of the Date Schedule. If no date is active -1 is returned.

Ex. of SYSNO:

- **T12** = Time Schedule 12.
- **V7** = Condition 7.
- **R3B** = Regulator 3 step B.
- **N36** = Indication on.
- **N44C** = Unacknowledged or remaining alarm.
- **25** = the value 25.

In conditions and regulators objects can be used directly (see respective function), and any SYSNO is not needed. If any additional letter is needed it can be added afterwards.

CCU to CCU communication

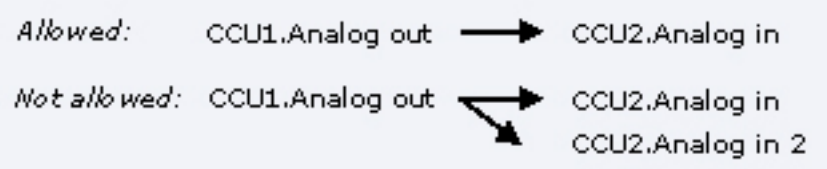
Between CCU's of type LS920

The list below shows the valid combinations of sender and receiver objects for CCU to CCU communication. Only the "original value" of the object is sent, i.e. suffixes cant be used.

Sending object in CCU x	Receiving object in CCU y
Manoeuvre	Indication
Indication	Indication
Alarm	Indication
Analog out	Analog in
Analog in	Analog in

Other regulations:

1. An object can't act as both sender and receiver, i.e. an object that receives a value form a CCU can't pass this value on to another CCU.
2. Only one object in a CCU can act as a receiver for a specific sending object in another CCU.
   Ex.:



3. Don't forget to reload the CCU containing the sending object when you create a receiver object (the CCU with the sending object must know where to send it's value).

Between CCU's of type Avalon

The list below shows the valid combinations of sender and receiver objects for CCU to CCU communication. Only the "original value" of the object is sent, i.e. suffixes cant be used.

Sending object in CCU x	Receiving object in CCU y
Manoeuvre	Indication
Indication	Indication
Analog out	Analog in
Analog out	Curve
Analog out	Pulse
Analog in	Analog in
Analog in	Curve
Analog in	Pulse
Pulse	Analog in
Pulse	Curve
Pulse	Pulse
Curve	Analog in
Curve	Curve
Curve	Pulse

(The "Other regulation" above doesn't apply to CCU's of type Avalon).

Communications check

Atlantis keeps statistics on the communication with the LS920 CCU. The following information is available:

- The number of received messages.
- The number of messages with errors (CRC error). Indicated i %.

All of the values are calculated from the startup of Atlantis or from when an 'Update PC' command has been issued.

Editing

The number of received messages:
Create a **Pulse object** with the 'KOM' address type.
Enter the following address: **920C-xx-yyy**, where **xx** is the loop id and **yyy** is the CCU id.

The number of messages with errors (%):
Create an **Analog In object** with the 'KOM' address type.
Enter the following address: **920C-xx-yyy**, where **xx** is the loop id and **yyy** is the CCU id.

Example:

`920C-01-001`

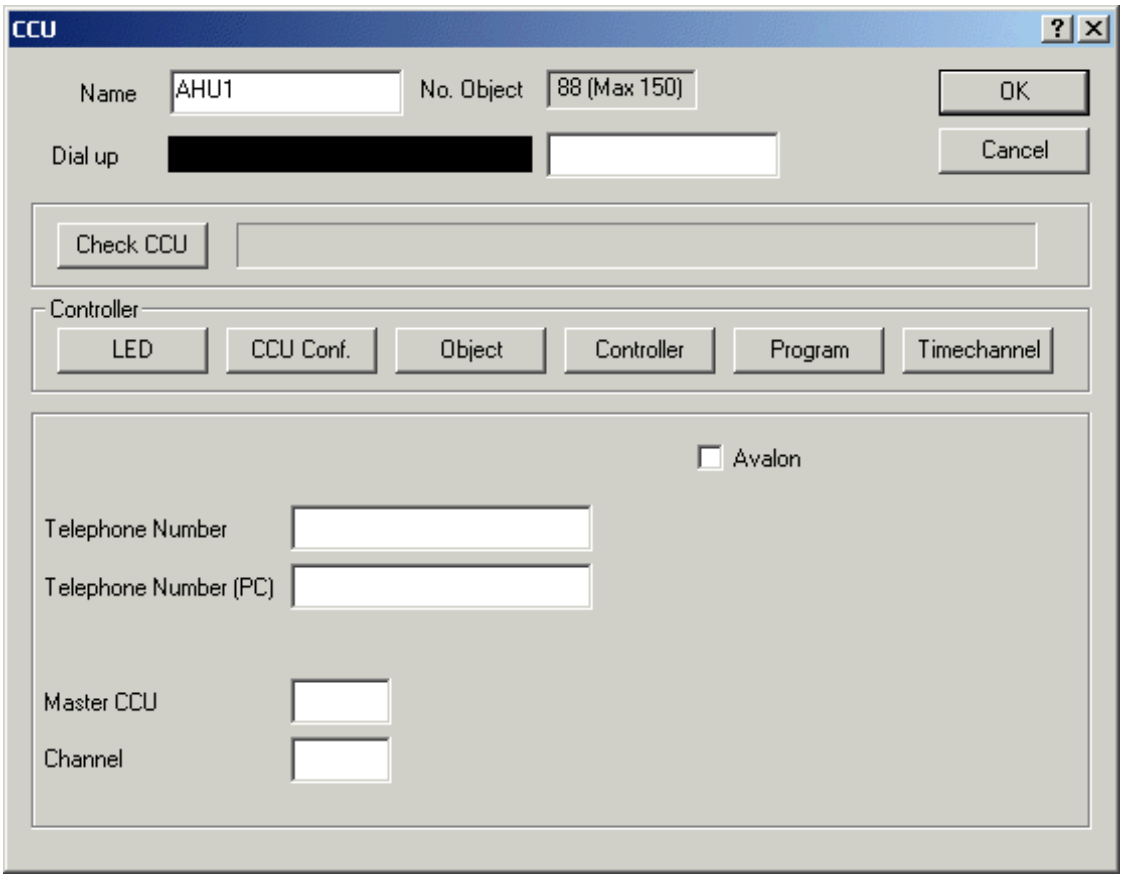
The number of received messages from CCU1 on loop 1.

`920E-02-007`

The error percentage from CCU7 on loop 2.

CCU Info

LS920



Name

The name of the CCU.

Dial-up

Condition or object stating when a CCU should perform a dial-up connection. This object's output is an approx. 4 minute pulse when the CCU is making the connection.

Check

Check the program, connections, CCU to CCU communication configurations, module addresses, LED's etc.

Lists

Create lists for printing.

CCU Phone number

The phone number used by the Atlantis Server to make a dial-up connection to a CCU.

Atlantis Server phone number

The phone number used by a CCU to make a dial-up connection to the Atlantis Server when the "Dial-up" function creates a pulse.

Master CCU

The id number of the CCU acting as a master (i.e. the CCU that has the "Dial-up" function activated and is connected to the modem).

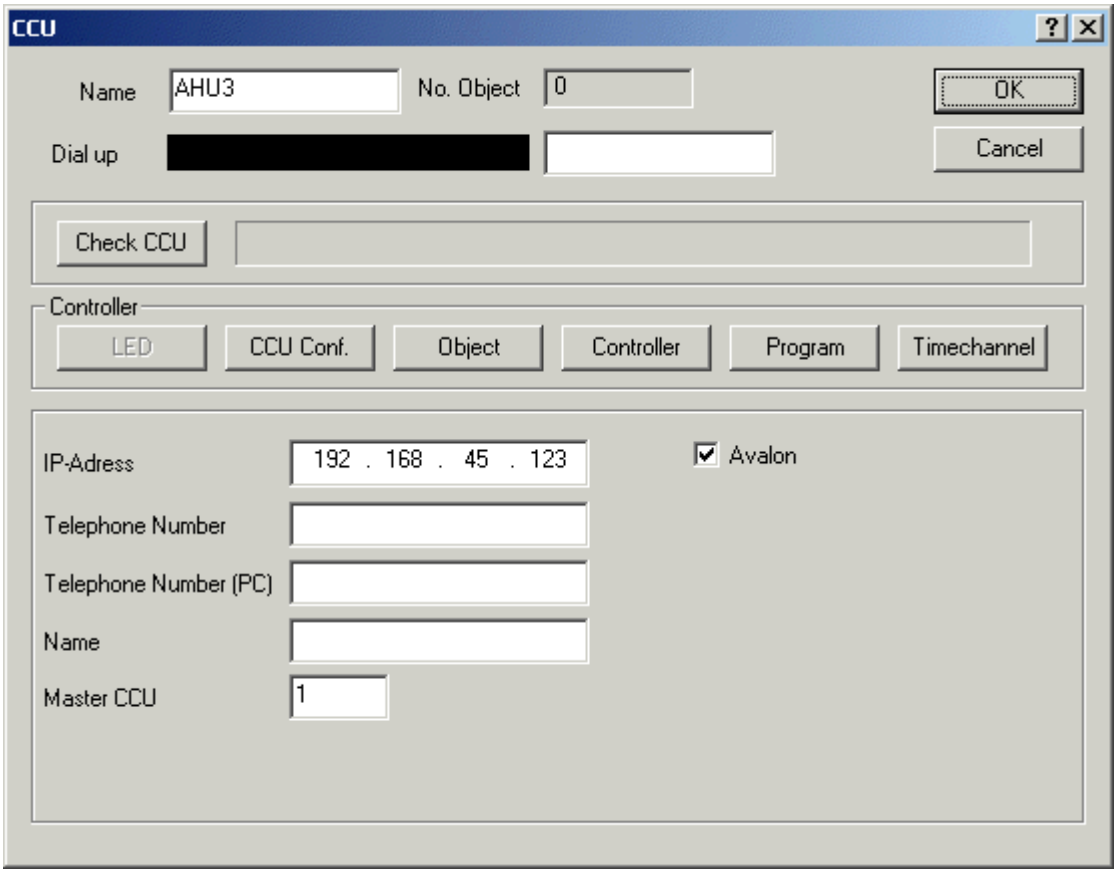
Loop

The loop id of the CCU. The loop id refers to the startup switch for the driver configured for the COM port which the CCU is connected to (see **Driver List** for details).

Avalon

Click this checkbox if the CCU is of type Avalon. The CCU will be converted if it's already programmed.

Avalon



Name

The name of the CCU.

Dial-up

Condition or object stating when a CCU should perform a dial-up connection. This object's output is an approx. 3-5 second pulse when the CCU is making the connection.

Check

Check the program, connections, CCU to CCU communication configurations, module addresses etc.

Lists

Create lists for printing.

IP Address

The IP address of the CCU. Must be unique.

If the CCU uses a dial-up connection and there are several buildings in the plant the CCU connects to, the third digit in the IP address must be different for each building.

The master CCU (the CCU with the modem) uses two IP addresses; The address entered and an IP address which is the entered address +1 (in the last field).

CCU Phone number

The phone number used by the Atlantis Server to make a dial-up connection to a CCU.

Atlantis Server phone number

The phone number used by a CCU to make a dial-up connection to the Atlantis Server when the "Dial-up" function creates a pulse.

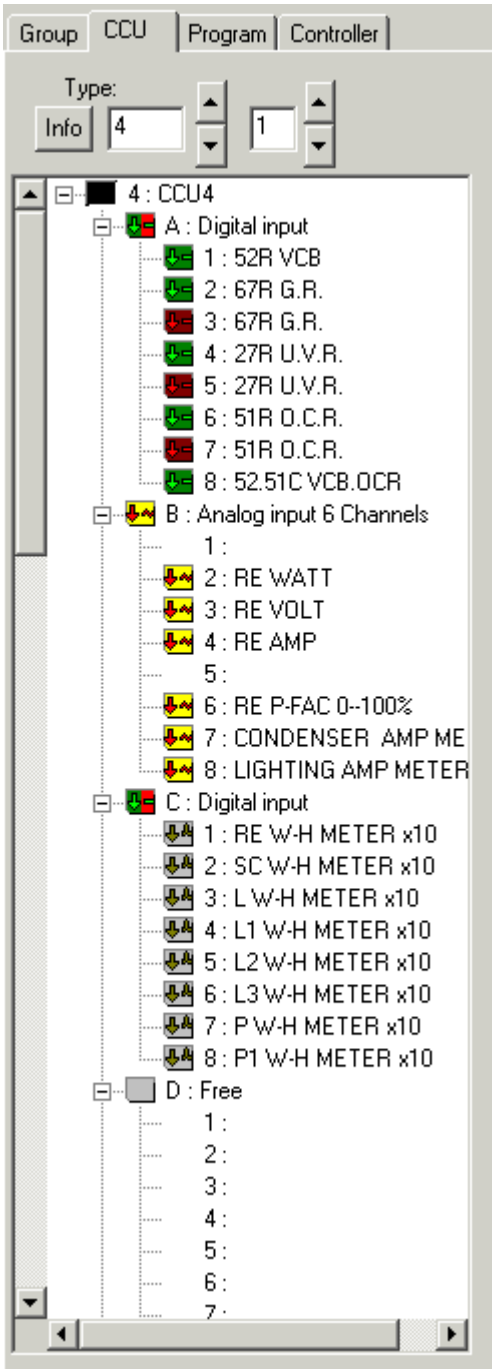
Phone book name

The post in the phone book to be used (**NB**! Only when using a leased line).

Master CCU

The id number of the CCU acting as a master (i.e. the CCU that has the "Dial-up" function activated and is connected to the modem).

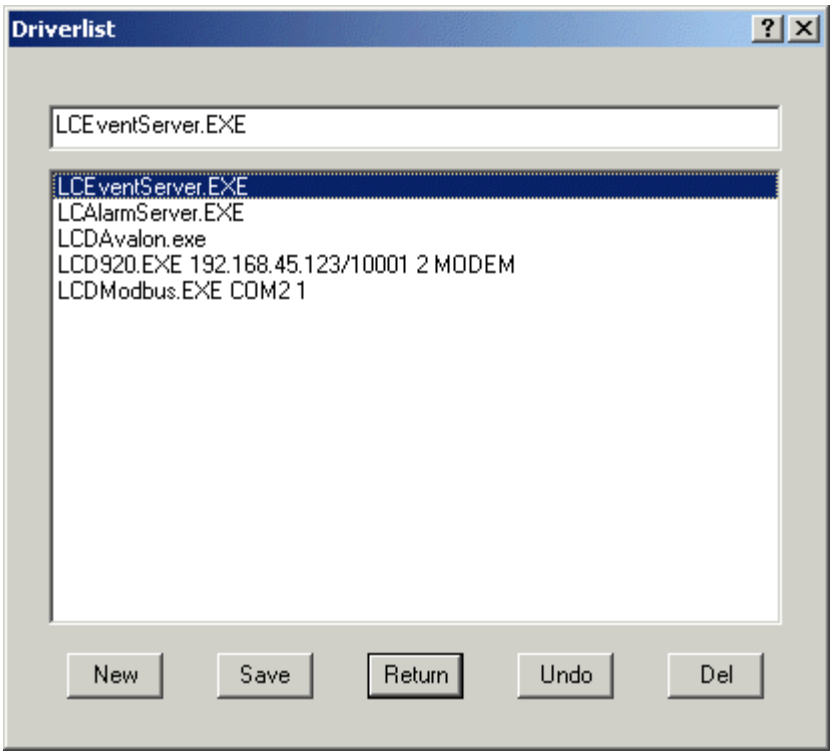
Module display for the LS920 and Avalon CCU



Displays the modules of the CCU.
The id of the base module is also shown if the CCU is of type Avalon.

Press ***Info*** to show information about loops, phone numbers, IP addresses etc.

Driver List



This list shows the drivers and/or other processes that will be started by Atlantis (The service LCService).
Create a new entry:
Press ***New*** and enter the name of the driver/process and then press ***Save***.

Delete an entry:
Click the row to be erased, press ***Delete*** and then ***Save***.

Exit the Driver List by pressing ***Return***.

Example of programs and drivers which can be used:

LCEventServer.EXE

Manages trend logging.

LCAlarmServer.EXE

Manages alarm transmissions to SMS, Email, Fax etc.

LCD920.EXE COM1 1

Driver for the LS920 CCU.
(Communicates with loop 1, through a RS485 modem connected to COM1)

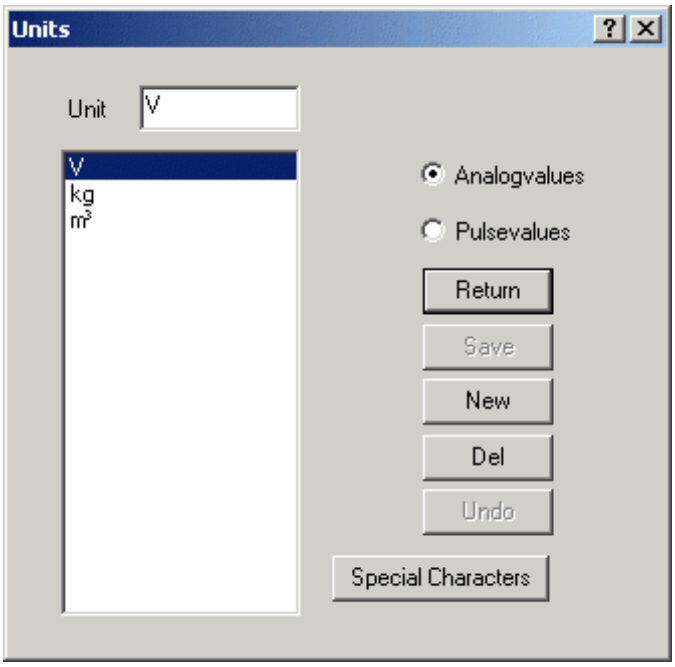
LCD920.EXE 192.168.12.231/10001 2 MODEM

Driver for the LS920 CCU.
(Communicates with loop 1 through an IP Port)

LCDAvalon.EXE

Driver for the Avalon CCU.

Unit List



This dialog is used for creating new units. It contains both a list for analog values and a list for pulse values.

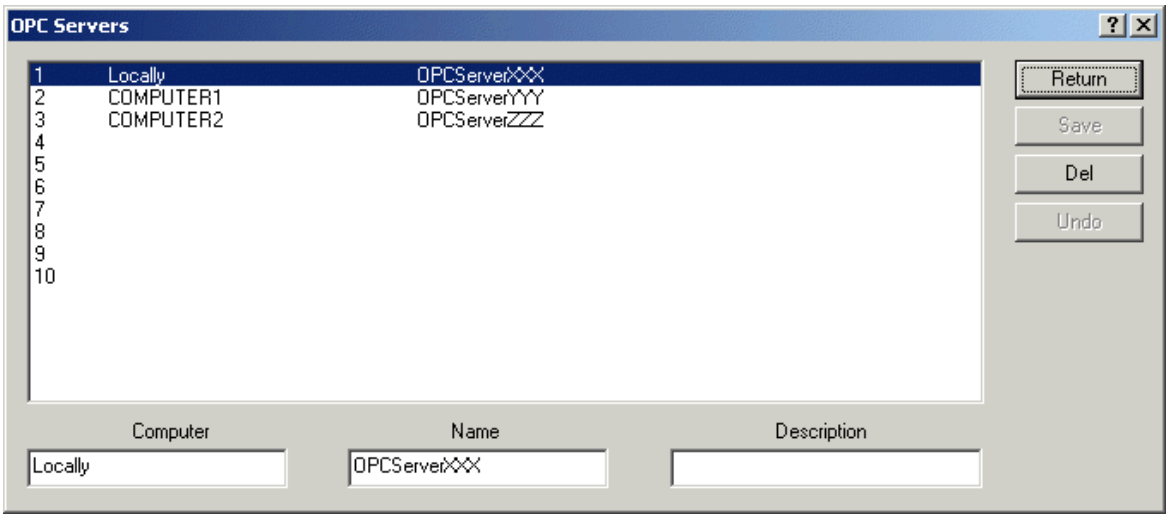
Create a new unit:
Press ***New***, enter the unit and press ***Save***.

Delete a unit:
Click on the unit to be deleted, press ***Delete*** and then press ***Save***.

Edit a unit:
Click on the unit to be edited, edit the unit and the press ***Save***.

Insert a special character:
Press ***Special Characters***. This starts the **CharMap.exe** software.

OPC Server List



This list shows which OPC servers are used.
To add a new server, click on an empty row and enter the name of the OPC server and press ***Save***. If the OPC server is installed on the same computer as Atlantis the **Computer** field must be left empty, otherwise enter the name of the computer where the OPC server is installed.

See also [Address specifications for OPC]

Export energy values

All values from the **Pulse objects** are exported to a text file. This file can be imported into, for example, Microsoft Excel.
The file is called **PulsExport.txt** and is located in the **Larmia.dok** folder. The file is updated two times/hour and contains the current value.

There is also a file called **PulsExportYYYMM.txt** (**YYYY** denotes the year and **MM** denotes the month) where the monthly values are stored.

The files have the following format:
Object number;Date;Group name;Object name;Value;Unit
.
.
.

If only some of the values are to be used, or if they must be sorted in a special order, a list of these objects can be entered in a file called **PulsExport.ini** which is located in the **Larmia.edt** folder.

The file must have the following format:
342
543
668
876
.
.
.

Version Requirements:
LCserver.exe version 4.0.003.