



Infoedit Manual

Version 2

A3M Ingenieurbuero Wechsung
Rebenring 33
D-38106 Braunschweig
Germany
Tel.: +49-531-233 77 40

Contents

CONTENTS	3
CONVENTIONS USED IN THIS DOCUMENTATION	5
DOCUMENTATION	6
HOW THE INFODISPLAY WORKS	6
INSTALLATION	6
STARTING THE PROGRAM	6
NEW PROJECT	6
CREATING A NEW PROJECT	6
IMPORTING FILES	7
DEFINING PROJECT SETTINGS	8
DEFINING NETWORK VARIABLES	10
DEFINING INTERNAL VARIABLES	12
SCHEDULER DEFINITION	15
ENUM DEFINITION	16
GENERATE COMPARISONS OR CALCULATIONS	17
GRAPHICAL LAYOUT	19
TOOLS	21
ARROW	21
LINE	21
RECTANGLE	22
CLOCK	22
SYSTEM CLOCK	23
TEXTBOX	23
POSITION	24
ADDITIONAL PROPERTIES	24
DIAGRAM	25
POSITION	25
VIEW	26
VARIABLE ASSOCIATION	26
BARS	28
POSITION	28
VIEW	28
BARGRAPH TYPE	28
BORDER	29
VARIABLE ASSOCIATION	29
BITMAPS	29
INSERT BITMAPS	29
CREATE BITMAPS	31
EDIT BITMAP	32
DELETE BITMAP	34
SIMULATION	35

CHECKING THE CONTENTS OF THE VARIABLES	36
OTHER MENU OPTIONS	37
FILE, EXPORT	37
CREATING THE APPLICATION FOR THE INFODISPLAY	38
LOADING THE APPLICATION INTO THE INFODISPLAY	38
ADDITIONAL NOTES	38
APPENDIX 1: BITMAPS	40
APPENDIX 2: ICONS	41
APPENDIX 3, PIXEL-FONT	42
APPENDIX 4: DEFINITIONS	44
ARRAY	44
BITMAP LIBRARY	44
DESTINATION VARIABLE	44
DIFFERENTIAL TYPE	44
ENUM DEFINITION	44
INITIALIZE	44
INTERNAL VARIABLE	44
OFFSET	44
POLLING CYCLE	44
PROJECT	44
SD STRING	45
SNVT	45
SNVT BIND INFO	45
SNVT CLASS	45
SNVT LIST	45
SNVT MODIFIER	45
SYSTEM CLOCK	45
SOURCE VARIABLE	45
VARIABLE LIST	45

Conventions used in this documentation

Text in a frame

These steps are absolutely essential!

Text in the right margin of the page:
Helpful information for quick reference

Explanations

Italics

These expressions can also be found in the chapter Index and on other pages of the documentation.

Bold face

Terms and expressions that also appear in the Infoedit.

Underlined

Terms and expressions that are to be stressed.

Documentation

How the Infodisplay works

The configured Infodisplay is a LonWorks® network node which gets its information from network variables which have to be bound. While creating the project-specific surface, the required information is configured by means of the software Infoedit. The Infodisplay receives the information from the network variables and copies it into the so-called “internal” variables. This step is necessary because network variables are often defined as structures that are composed of individual elements. The Infodisplay can display these elementary data types, but it cannot display complex structures. When displaying a page, the required values of the internal variable are read out and interpreted.

Installation

Insert the CD into the drive. The set-up program starts automatically, takes you through the installation procedure and creates an icon.

Starting the program

Double click the Infoedit Icon to start the program

New Project

Creating a new project

With **File** in the menu bar **Project** and **new**, you create a new project

Define a project name with **File** in the menu bar **Project** and **Save as...** . If the name is not defined at this point, you will be asked for it automatically after a while.

A project consists of three different file types:

- files containing the Infodisplay GUI, ending with .ife
- files containing the bitmap library, ending with .bib
- files containing the variables list, ending with .lst

To **import** files from different projects to your current project click **File** and **Import**.

Before you create the GUI you should define the SNVTs you want to bind later in the network.

Before creating the GUI, define the variables!

Importing files

When having started the program for the first time, you have to import an SNVT list into your new project: using **File** in the menu bar **Import** and **SNVT List...**

The Infoedit comes complete with the standard SNVT list snvt.nvt. This file contains all SNVTs from the SNVT Master List.

If you want to use an existing bitmap library, you have to import this list using **File** in the menu bar **Import** and **Bitmap Library...**

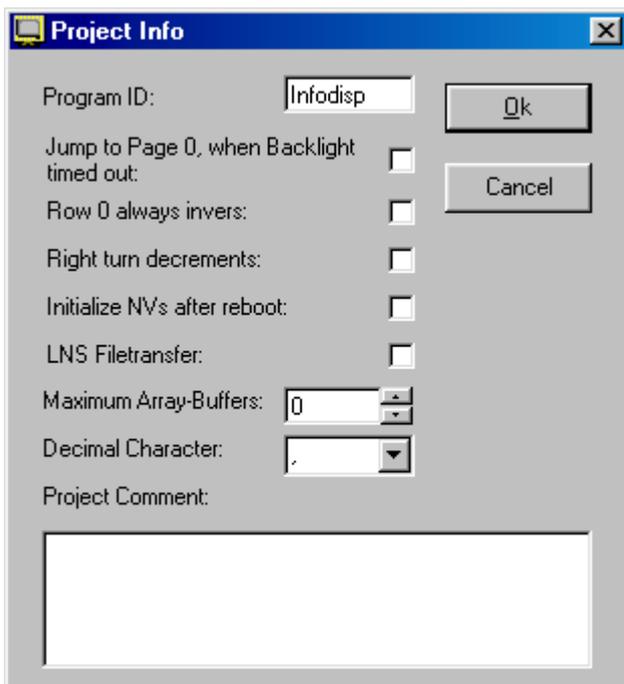
Together with the Infoedit you get the file bitmap.bib, which contains 60 bitmaps. These bitmaps are listed in the appendix.

Normally you will define new variables for a new project. If you want to use existing variables, you can import variable lists using **File** in the menu bar **Import** and **Variable List...**

Defining project settings

When starting a project, a number of basic settings have to be made. These are, on the one hand, required for later generation of the application file (*.apb), or they are settings that apply to the project as a whole but can be set by up-dating the GUI data. Settings that can only be changed when again generating an application, will be mentioned specifically.

Edit and **Project Settings...** or pressing F5 will open the following window:



- **Program ID**

The application name that is stored in the neuron chip. When reset, the Infodisplay compares the program ID with menu data names. In case of discrepancies, a message will be displayed instead of a menu.

This is set when generating the application.

- **Jump to Page 0, when Backlighting timed out**

When a time which can be set (variable: Backlighting, type: unsigned short) has run down, the display background lighting will be switched off. This setting defines whether the start page is to be displayed at the same time or whether the page currently used is to be maintained.

- **Row 0 always inverse**

This defines whether the top row in the display should always be inverse (dark).

- **Right turn decrements**

This allows you to define the sense of rotation of the knob.

The initialization value of the variable __Turn is changed (type: unsigned short).

- **Initialize NVs after reboot**

If the output variables are to be transmitted once following Infodisplay reset.

The initialization value of the variable __InitNV is changed (type: unsigned short).

- **LNS file transfer**

Generates the required network variables and integrates the option of reading out and describing internal memory areas by means of the (LonMark) file transfer. This option presupposes a network management tool with file transfer functionality.

This is set when generating the application.

- **Maximum Array Buffers**

Generates the required application structures for use of data logger functionalities. Each data logger needs a separate buffer.

The maximum number of data loggers is set when generating the application.

- **Decimal Character**

Defines the decimal separator (normally “.”).

- **Project comment**

This is where a comment can be stored for the application.

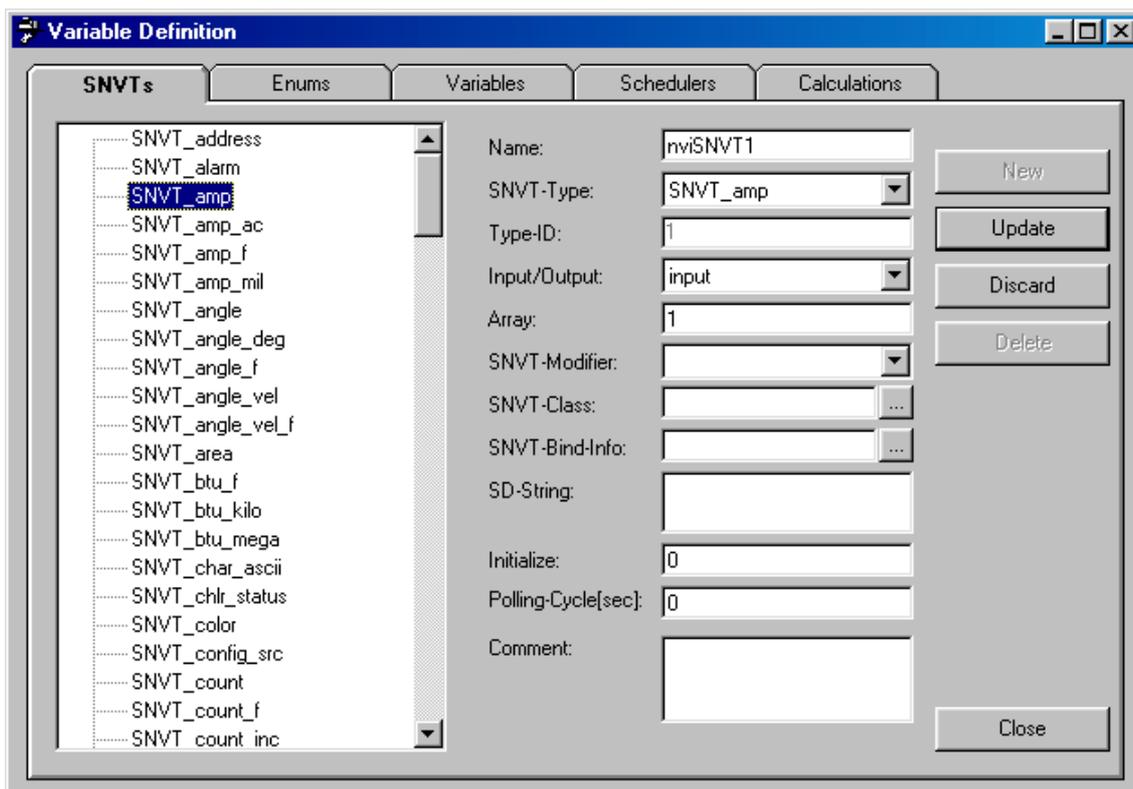
Defining network variables

You have to define network variables to be able to communicate with the LONWORKS® network, and to be able to display them on the Infodisplay. This is the purpose for which the so-called “Standard Network Variable Types“ (SNVT) are provided, which comply with the LonMark® Standard.

These network variables (the Infodisplay is able to manage a maximum of 62) can be configured by means of dialogs to be able to process and show them in the Infodisplay.

Use **Edit** and **Edit Variables...** or click  to produce the following window:

The imported SNVT list is shown on the left-hand side. On the right-hand side you have to define the SNVT.



Choose a standard variable type on the left-hand side.
Then click **New** to define a new network variable.

- **Name**

The logical, internal name of the variable that has just been defined.

This is the name by which the variable is known in the network. It starts with nvi for an input variable and with nvo for an output variable.

- **SNVT Type**

The Standard Network Variable Type (SNVT, see “LonMark[®]-SNVT Master List“)

- **Type ID**

The ID number of the SNVT (see “LonMark[®]-SNVT Master List“)

- **Input/Output**

The direction of the network variable (input or output variable)

- **Array**

Allows you to define arrays of identical network variables.

These variables are provided with indices that allow them to be accessed.

- **SNVT Modifier, SNVT Class, SNVT Bind Info**

This is where additional variable properties can be set (see “Echelon Neuron C Programmer’s Guide“)

- **SD String**

This is where the self-documentation string can be entered, i.e. short information on the network variable which can be written by yourself.

- **Initialize**

Here you can define a standard value for the variable. This value will be shown when starting the Infodisplay for the first time or after a prolonged power failure.

Initializing starts in case of a checksum error in the battery-buffered RAM.

- **Polling Cycle [sec]**

This is the time-controlled repeated read-out cycle of input variables whose value is to be read out on a time-controlled basis.

- **Propag. Cycle [sec]**

For output variables whose value will be up-dated in the defined cycle.

- **Comment**

An additional comment on the network variable.

You can **Discard** changes made or **Delete** the current SNVT.

To be able to use a standard network variable in your menus, you have to connect it to an *internal variable*. You finish the SNVT definition with **Update** and you will be asked, if you want to create an *internal variable* automatically.

It is recommended to click **Yes** here. Otherwise you can define the *internal variable* yourself on the page **Variable Definition**.

Defining internal variables

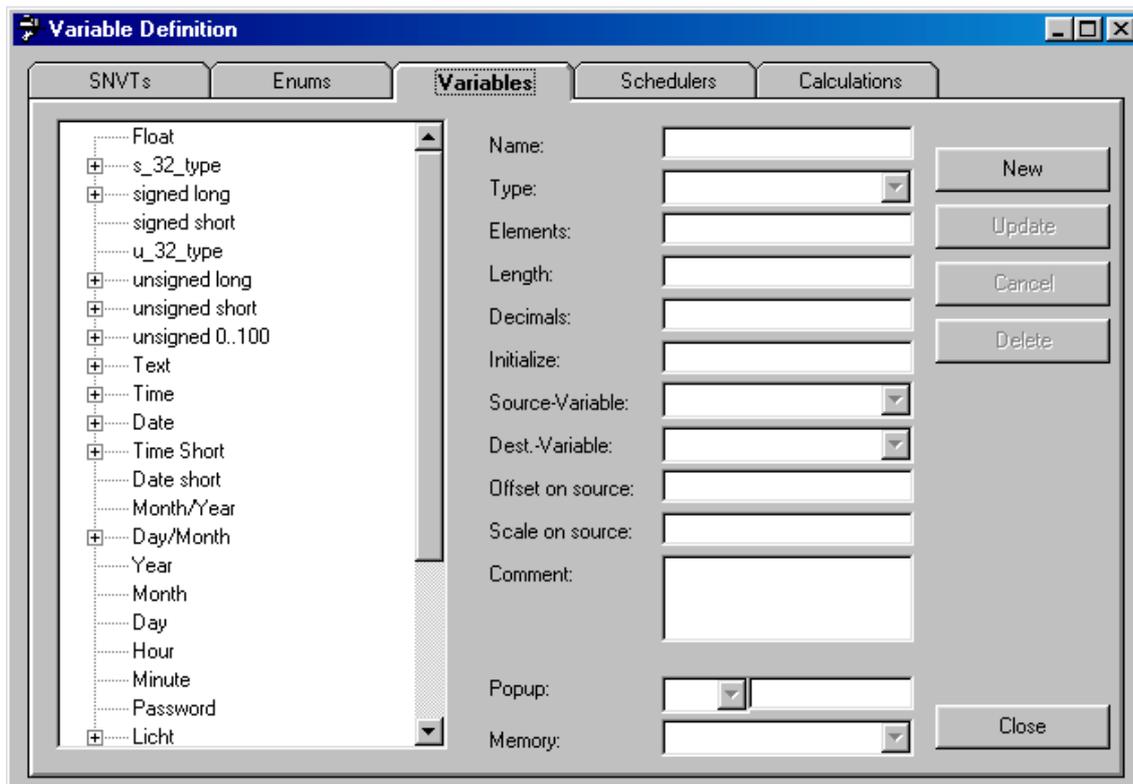
Click **Variable Definition**

On the left-hand side choose the variable type you need.

Click **New**.

Internal Variable
Definition

A special type of *internal variable* is the “password” type. To activate it you have to input a password under **Initialize**. Details are given in the chapter “textbox”. Some *internal variables* (e.g. LED, Real Time Clock, Contrast) already exist to support the hardware.



- **Name**

Here you enter the logical name of the internal variable.

This is the name used to access the variable inside the Infoedit for communication, e.g. to define the GUI.

- **Type**

This box contains the variable type you have defined before clicking **New**. These file types are the basic types that the Infodisplay is able to show on its surface or that are shown by the internal hardware (LED, Real Time Clock, etc.).

- **Elements**

Here it is possible to set the amount of elements of an array. There are three possible functions:

1. An array of SNVTs is connected to an array of *internal variables*.
2. To show a diagram, an array is created to save data in a time-controlled manner. Here you enter the total number of storage units you need (e.g. 24*7 storage units to save data for one week on an hourly basis).
3. An array is created to save data in an event-controlled manner and to show them with date and time.

If you set more than one element, part of the input box will change:

Backup Variable, Saving Cycle, Block Length and Differential Type will be shown in addition.

Dest. Variable, Offset, Scale and Popup will not be shown instead.

- **Length**

This is where you can enter the standard length in characters of strings to be shown on the Infodisplay. You can change this value when defining your menus.

- **Decimals**

Here you can set the standard number of decimal places that should be shown on the Infodisplay. The decimal point will be inserted without conversion.

For example: **Decimals** = 1 \Rightarrow 12345 changes to 1234.5 You can also change this value when defining your menus.

- **Initialize**

Here you can define a standard value for the variable. This value will be shown when starting the Infodisplay for the first time or after a prolonged power failure. Initializing starts in case of a checksum error in the battery-buffered RAM.

- **Source Variable**

This is where you enter the input variable which you take the data from. It may be a network variable or another local variable, which is to be shown or saved in a different format.

- **Dest. Variable**

This is where you enter the output variable (nvo) the internal variable should be bound with (destination variable). It may be a network variable or an internal variable for an LED allowing the LED to be switched according to the given value.

- **Offset**

In this box you enter a constant value, which is to be added to, or subtracted from, the *Source Variable*.

- **Scale**

Here you can enter constant factors the *Source Variable* will be multiplied by. If the result is less than one or if the result contains characters behind the decimal point, the variable will be calculated internally as a floating point number and will be converted into the format as defined.

- **Comment**

Here you can enter a comment for your own purposes.

- **Popup**

Here you can enter a condition. If the condition is fulfilled, the popup page will automatically be shown (page 2).

- **Memory**

The variables can be saved in the battery-buffered RAM or in the so-called dataflash. The data in the dataflash cannot get lost. But the dataflash can be overwritten only about 100,000 times. Normally, only configuration variables and values required for the data logger should be saved in the dataflash. As they are updated frequently, normal variables may destroy the dataflash component.

The following features can only be accessed by means of arrays:

- **Backup Variable**

Here an existing variable of the same type and length can be selected. The contents of the original variable will be copied into the selected variable once the former is completely filled.

- **Saving Cycle**

Sets the saving cycle of the variable.

By selecting "when update occurs" the variable will be saved with a time stamp. In addition, the following variables will automatically be generated:

Type	Name	Function
Unsigned short	__xxx_in	index of the current entry (don't change!)
Unsigned short	__xxx_out	index of the current entry (for display)
Time	__xxx_Time	table of time stamps for the events

"xxx" is the name of the internal variable

- **Block Length**

Indicates the length of a bloc for display in a diagram. The length is, for instance, used to display the whole year of monthly diagrams of one data array.

- **Differential Type**

Indicates whether the variable itself is saved or whether it is to be reduced by the previous value before saving.

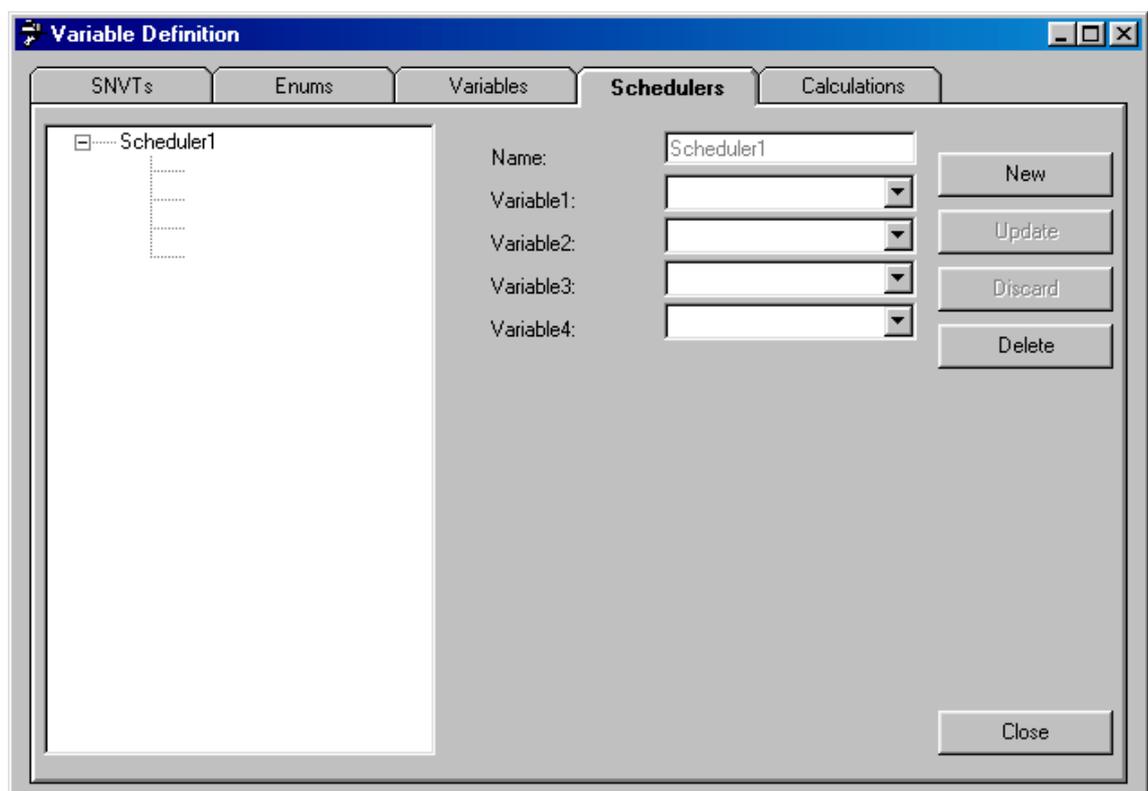
Scheduler definition

Click **Scheduler Definition**

Click **New**

A maximum of 10 schedulers can be generated. Each program contains 24 switching points, and each of these switching points can be repeated cyclically (on an hourly, daily, monthly, annual basis, etc.). For each switching point, either an optional value or no value can be entered in the internal variables.

A configuration page will automatically be produced at the end of the menu. Switch point setting can only proceed immediately in the Infodisplay.



- **Name**

A name will automatically be set for the scheduler.

- **Variable1 .. Variable4**

Here you enter the internal variables which this scheduler is to influence.

Enum Definition

To set an enumeration type (e.g. Monday, Tuesday, Wednesday,... instead of 1, 2, 3,...) you can use the page *Enum Definition*. Enumeration always starts with the contents for the value 0.

Click *Enum Definition*.
Click **New**.

Enum
Definition

- **Name**

Here you can name the new enumeration type.

- **ID**

Here you will see an internal control number for extending the source code.

- **Max. Length**

Here you can key in the maximum number of characters to be stored. When designing the GUI you can define value shorter than the word, which means that the word will be broken off at this point.

- **Entries**

Here you will see the number of entries you have already made.

- **Comment**

A comment for your own purposes.

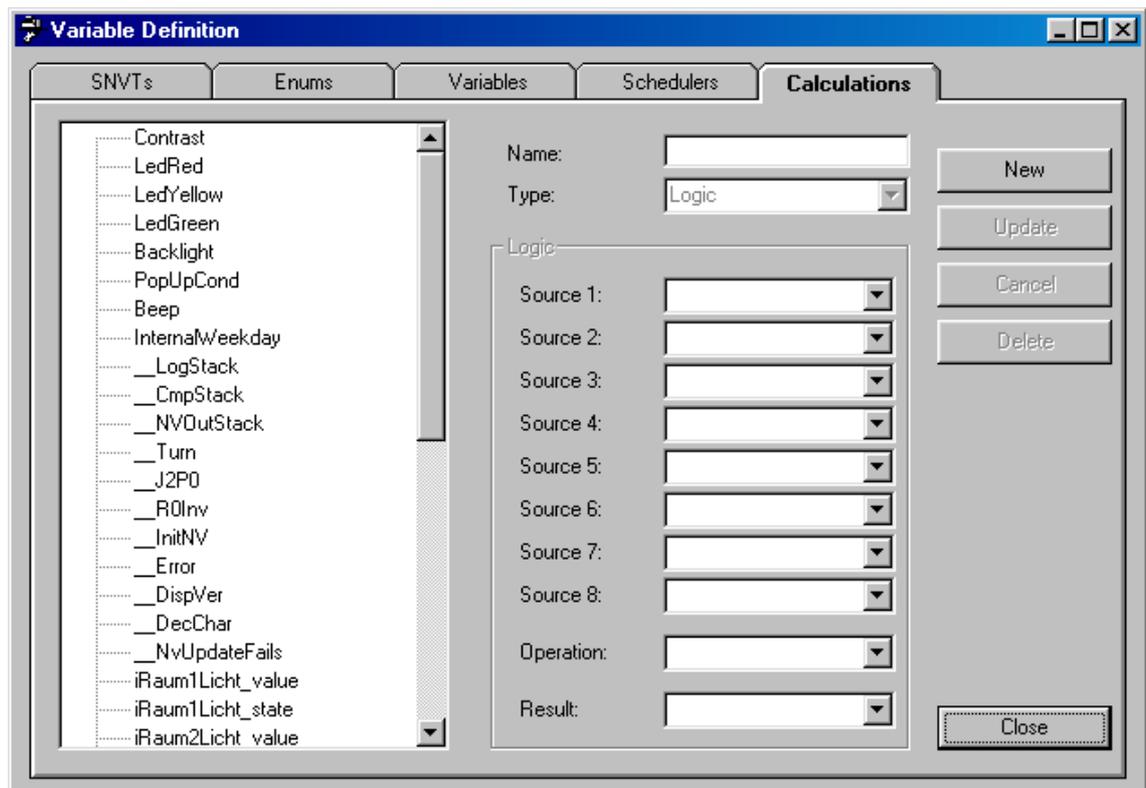
Generate comparisons or calculations

Use the page **Calculations** for comparisons or calculations. The type “unsigned short” is intended for logic and compare operations, the type “float” for calculations. All calculations will be repeated if one of the source variables changes (“Source 1...”). This is why in generating a calculation, results must not be used in a way that makes the calculation dependant on its own result.

Click **Calculations**

Select a variable as a result

Click **New**



- **Name**

This is where you can define a name for the new calculation.

- **Type**

Enter the type of calculation here:

- Logic (logic operation)
- Compare (comparison)
- Calc (calculation)

- **Source [1..8]**

Enter the source values.

- **Compare Var.**

For comparisons, the variable which the source value is to be compared is entered here.

- **Operation**

Use operation to select the operation for the intended type of calculation:

- Logic: AND, OR, XOR (exclusive-OR)
- Compare: < (smaller), > (greater), = (equal), <> (not equal)
- Calc: + (plus), - (minus), * (multiplied), / (divided)

- **Result**

The result of the calculation is copied into this variable.

Graphical layout

First you should have an idea of what the application should look like:

1. What should be shown on the Infodisplay?

What information is supplied from the LON network?
What additional information should be displayed (e.g. Time, Bus Time Table, Addresses,...)

2. Where should the information appear?

What kind of information should appear on the *start page*?
What kind of links should exist and to what page?
Is there enough space to show everything you want?
Are the pages clearly structured?

3. What has to be controlled by the user of the Infodisplay?

Should it be possible for the user to control the blinds or the lights of the room from the Infodisplay?

4. What order should be used to take the user through the menu?

Once you have got an idea of the menu structure, you should proceed as follows:

1. Start with the *start page*

At first, name the page. For your later work it is necessary to choose a name that gives you an idea of the contents of the page (e.g. "start" for the first page)

Double click page name to name the page or to change its name (before you name a page the name of the page is "no name").

Now you can create your page (see **Tools**), insert textboxes, diagrams or bitmaps.
Provide the textboxes with links or variables.

2. Continue with the next page in your application:

Click  . You will be asked if you want to insert a new page.

Answer "yes" and name the second page.

Now create this page, too (see **Tools**). Repeat these steps until all pages of your application have been created.

With  and  you can move through your created pages to check and change them.

Pages:

With  or **Page** and **New...** you can insert a new page. This new page is inserted behind the one currently displayed.

insert a page

With **Page** and **Copy...** you can copy the current page. It will be inserted behind the page currently displayed. This may be interesting if you want to use a number of objects of your current page on other pages, too.

copy a page

With  or **Page** and **Delete** you can delete the current page.

delete a page

With **Edit** and **Copy Image** it is possible to copy the current page for use in other programs to the clipboard (e.g. for documentation or presentation purposes).

Objects:

Objects are *bitmaps, diagrams, textboxes, clocks, rectangles* and *lines*. Marked objects can be...

objects...

cut to the clipboard with  or **Edit** and **Cut** or

...cut

copied to the clipboard with  or **Edit** and **Copy**.

...copy

You can paste marked objects from the clipboard with  or **Edit** and **Paste**.

...paste

You can delete marked objects with  or **Edit** and **Delete**.

...delete

Tools

Arrow

The *arrow* is used for marking and moving graphical objects.

You can also mark a *group* if you drag the cursor across all objects you want to be enclosed in the *group*.

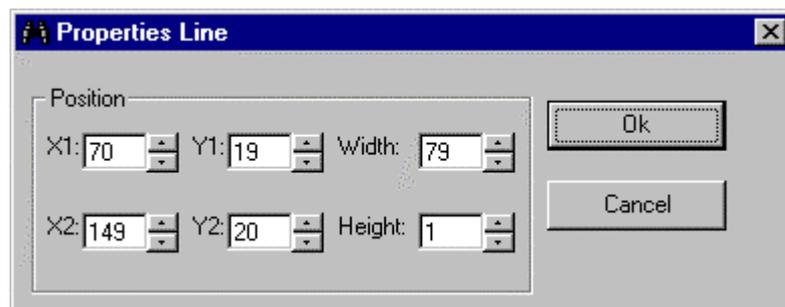
This marked group can be

- moved on the current page
- copied to the clipboard for use on another page 
- cut to the clipboard 
- deleted 

Line

You can draw horizontal, vertical and diagonal lines. The thickness of the line is constant and cannot be changed. If you need a different thickness you have to draw parallel lines.

Double click a marked line or select **Edit** and **Properties...** to open the following window:



Here you can define the position of the line as an alternative to moving it with the cursor.

For a horizontal line, the value 'height' is zero, and the values Y1 and Y2 are identical. Changing the 'width' value will change the length of the line.



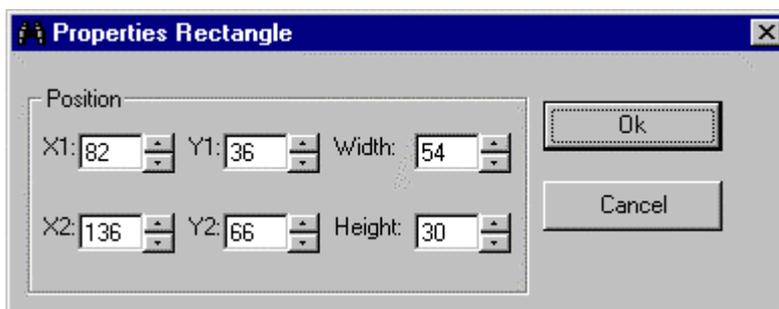
Rectangle

You can draw rectangles of any size, but they cannot be rotated or stretched, and the thickness of the lines cannot be changed. The rectangles are transparent.

To mark the rectangle, click one of its four corners.

To move it, place the cursor on one of the four sides of the marked rectangle and keep the left-hand mouse button pressed while dragging the rectangle.

Double click the marked rectangle or select **Edit** and **Properties...** to open the following window::



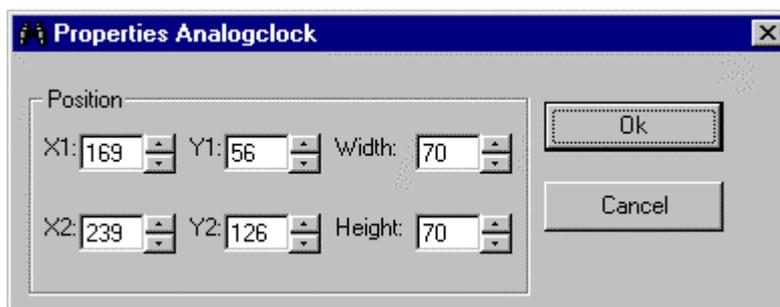
Here you can define the position of the rectangle as an alternative to moving it with the cursor.



Clock

Here you can insert the object *analogue clock*. The size of the clock is constant and cannot be changed.

Double click the marked clock or select **Edit** and **Properties...** to open the following window:



Here you can define the position of the clock as an alternative to moving it with the cursor. The values **Height** and **Width** are always 70 and cannot be changed.

System clock

If you do not make any further variable definitions, the clock is a system clock. This means that the clock has to be set when starting the Infodisplay after which it will run independently from the net.

If you want to use the LON time, you have to define a variable for it (see chapter "variable definition").

Some of the Infodisplay variables are internal variable types, e.g. the variable "internal time" for defining the time. If you use this time, you use the internal real time clock of the Infodisplay. To allow this clock to be set, it has to be editable in your menus. The variable "internal time" can also be connected with a network variable of type "time_stamp". This will synchronize the real time clock of the Infodisplay with the time of the LON net.

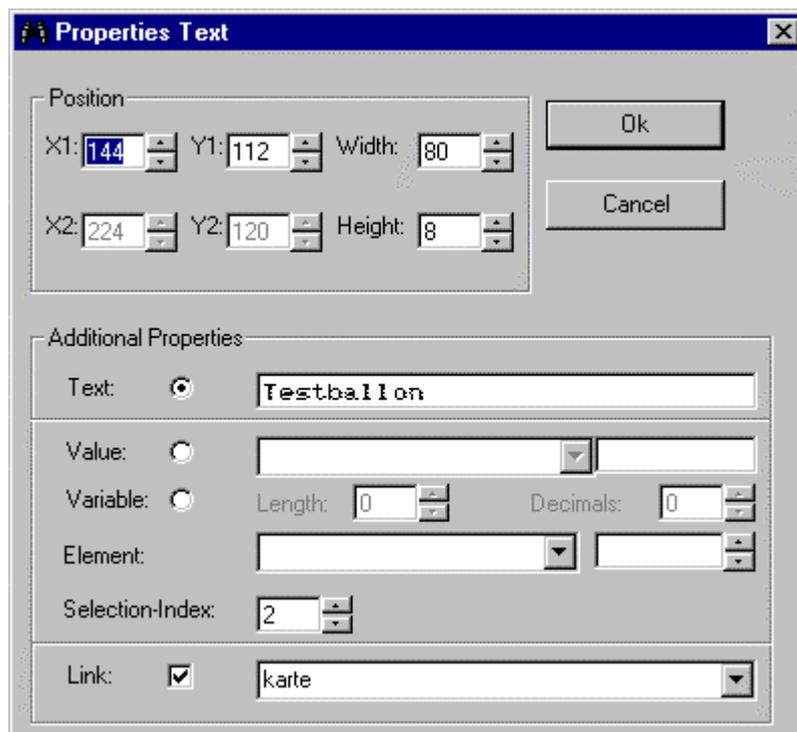
With new devices, a DCF77 receiver can be connected to the serial interface, which automatically updates the contents of the variable "internal time". These receivers can be ordered from A3M.

Textbox

Here you can insert a *Textbox*. The font and the size of the characters is constant and cannot be changed. The direction of the text is always horizontal.



Double click the marked *textbox* or select **Edit** and **Properties...** to open the following window:



The screenshot shows a dialog box titled "Properties Text" with a close button (X) in the top right corner. The dialog is divided into two main sections: "Position" and "Additional Properties".

Position: This section contains six spinners for defining the text's location and size. The values are: X1: 144, Y1: 112, Width: 80, X2: 224, Y2: 120, and Height: 8. To the right of these spinners are "Ok" and "Cancel" buttons.

Additional Properties: This section contains several options and input fields:

- Text:** A radio button is selected, and the text "Testballon" is entered in the adjacent text box.
- Value:** A radio button is unselected, with an empty text box and a dropdown arrow to its right.
- Variable:** A radio button is unselected. It includes "Length:" with a spinner set to 0 and "Decimals:" with a spinner set to 0.
- Element:** Two empty text boxes with dropdown arrows to their right.
- Selection-Index:** A spinner set to 2.
- Link:** A checked checkbox is followed by a text box containing "karte" and a dropdown arrow.

Position

Here you can set the position of the textbox by changing **X1** and **Y1**, as an alternative to moving the box with the cursor. All other **Position** values can not be changed.

Additional Properties

text properties...

You can set several properties of the textbox:

- **Text**

If you only want to write a text that will be shown on the display and which will not be changed, select **Text**. Now you can write the text in the box. A list of characters available for this purpose is included in the Appendix Pixel Font. You can set a **Link** to the textbox (see **Link**)

- **Value**

...value

Here you can connect your textbox with a variable which is to be shown in the textbox. The box on the right shows the variable type. You can set a **Link** to the textbox (see **Link**).

With **Length** and **Decimal** you can define the length and the decimal places of the text in this box. Your selection will be highlighted in the box **Text** by a grey colour. In the box **Selection Index** you can set the sequence in which textboxes are selected when turning the knob. For example, if you choose "2" this box will be the second box that appears on the page when the user is on that page and turns the knob.

If the type of the internal variable is "Password", stars "*" will automatically be shown. When the correct password is entered, the **Link** as defined (see **Link**) will be executed.

- **Variable**

...variable

If the user should be allowed to change the text in the box, select **Variable**. The user can select the required box by means of the knob. When pressing the knob with the textbox marked, the text can be changed by turning the knob.

With **Length** and **Decimal** you can set the length and the decimal places of the text in this box. Your selection will be highlighted in the box **Text** by a grey colour. In the box **Selection Index** you can set the sequence in which textboxes are selected when turning the knob. For example, if you choose "2" this box will be the second box that appears on the page when the user is on that page and turns the knob.

If the type of the internal variable is "Password", the password may be changed at this point.

- **Element**

...element

If an array is entered in the box *variable*, you can decide here what value of the array should be shown. For this function to be available, a variable must have been defined in **Variable definition**.

- **Selection Index**

Here you have to set the sequence of the textbox. For example, if you choose “2” this box is the second box on the page that will be chosen by the user when turning the knob.

- **Link**

Here you can set a link to the textbox. If you want to set a link, you have to enter the name of the page to which the user will get when choosing this textbox and then pressing the knob.

...link

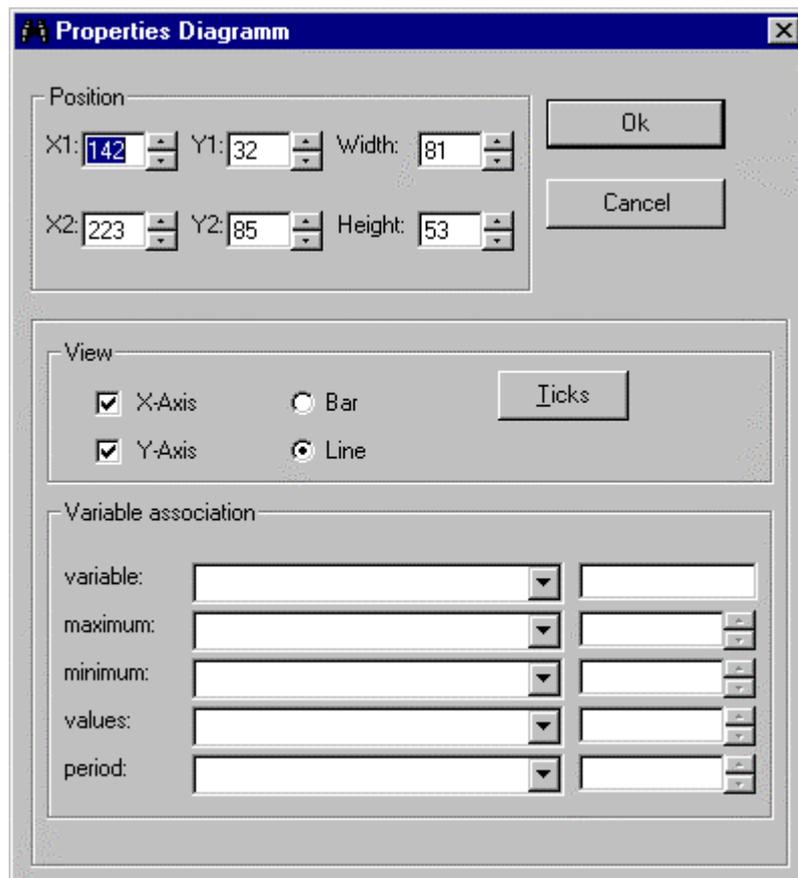
Diagram

Here you can insert diagrams. If you mark the diagram, you can change the size of it by dragging the diagram.

The *diagram coordinates* can be provided with a text by inserting textboxes after diagram configuration has been completed.



Double click the marked diagram or select **Edit** and **Properties...** to open the following window for marked diagrams:



Properties Diagram

Position

X1: 142 Y1: 32 Width: 81

X2: 223 Y2: 85 Height: 53

Ok

Cancel

View

X-Axis Bar

Y-Axis Line

Variable association

variable: [] []

maximum: [] []

minimum: [] []

values: [] []

period: [] []

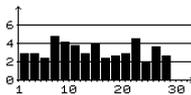
Position

Here you can define and change the position and the size of the diagram as an alternative to moving or dragging it with the cursor. The size (**Width** and **Height**) can

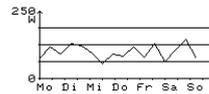
only be changed here and not by using the mouse.

View

You can decide whether the figures are to be represented in the form of *bar chart* or a *line diagram* with or without x/y axis.



bar chart

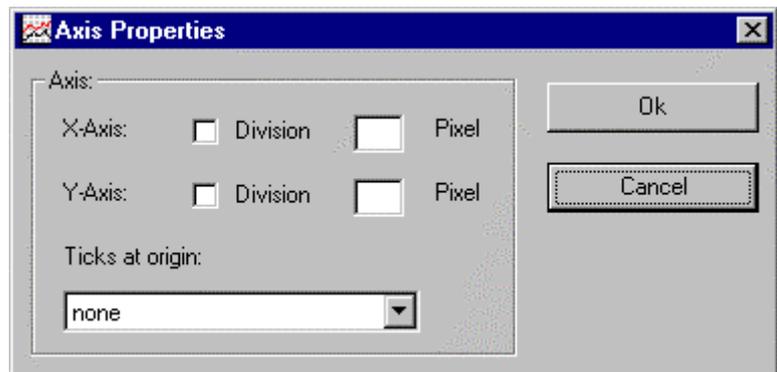


line diagram

If you click **Ticks**, you will get the following window:

Here you can set secondary graduation lines for the axis.

With **Ticks at origin** you can decide whether there should be any secondary graduation line in the origin and on which axis.



Variable Association

Here it is possible to configure the diagram:

- **variable**
You have to choose the *array* from the defined variables which should be shown in the diagram. In the box on the right, the *type* of the variable is shown.
- **maximum**
In the box on the right, the maximum value that should be shown in the diagram can be set. It is normally constant and is set as a value in the right-hand box. It is also possible to connect this value to a variable. In the box on the left you assign the variable.
- **minimum**
In the box on the right, the minimal value that should be shown in the diagram can be set. It is normally constant and is set as a value in the right-hand box. It is also possible to connect this value to a variable. In the box on the left you assign the variable.

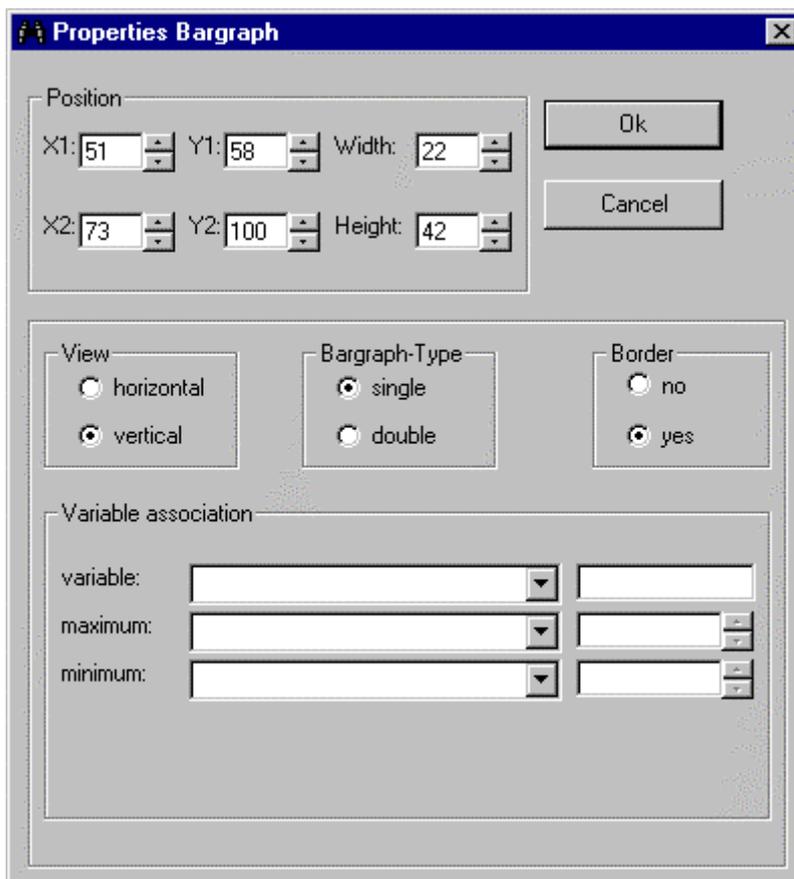
- **values**
In the box on the left you choose the variables that should be shown in the diagram as measured data. In the box on the right you set the number of data that should be shown in the diagram
- **period**
The section of the *array* that should be shown can be chosen here.

Bars

Here you can insert *bars*. If you mark the *bar*, you can change its size by dragging it. Text for the different bars can be added by inserting textboxes after bar configuration has been completed.



Double click the marked *bar* or select **Edit** and **Properties...** to open the following window:



Properties Bargraph

Position

X1: 51 Y1: 58 Width: 22

X2: 73 Y2: 100 Height: 42

View

horizontal
 vertical

Bargraph-Type

single
 double

Border

no
 yes

Variable association

variable: [] []

maximum: [] []

minimum: [] []

Ok

Cancel

Position

The position and the size of the bar can be set and changed here as an alternative to moving it with the mouse. The size of the *bar* (**Width** and **Height**) can only be changed here, not by means of the cursor.

View

Here you have to decide if you want to have **vertical**



or **horizontal**



Bargraph Type

single



double



Border

yes:



no:



Variable association

Here you can configure the bars:

values in the bar
chart

variable

You have to choose the *array* from the defined variables which should be shown in the bar. In the box on the right the variable type is shown.

- **maximum**

In the box on the right, the maximum value that should be shown in the diagram can be set. It is normally constant and is set as a value in the right-hand box. It is also possible to connect this value to a variable. In the box on the left you assign the variable.

- **minimum**

In the box on the right, the minimal value that should be shown in the diagram can be set. It is normally constant and is set as a value in the right-hand box. It is also possible to connect this value to a variable. In the box on the left you assign the variable.

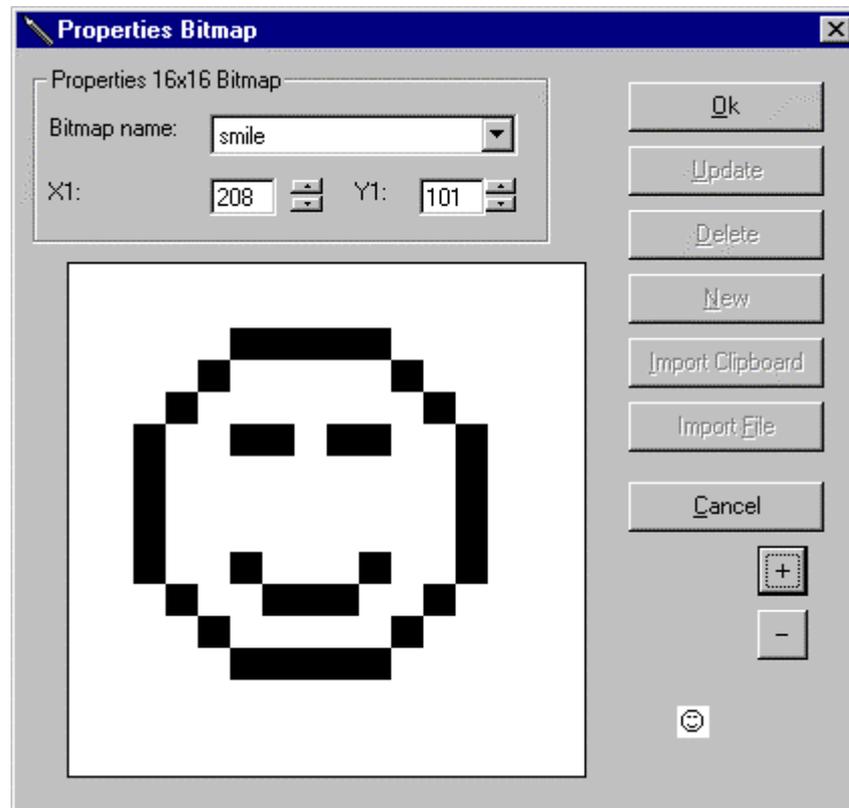
Bitmaps

Insert Bitmaps

Here you can insert bitmaps (16x16) by means of the cursor. Start by inserting an arbitrary bitmap from the library.



Double click this marked bitmap or select **Edit** and **Properties...** to open the following window:



In the field **Bitmap name**, you can select the bitmap that you want to insert here.

The position can be changed with **X1** and **Y1** if you do not want to move it by means of the mouse.

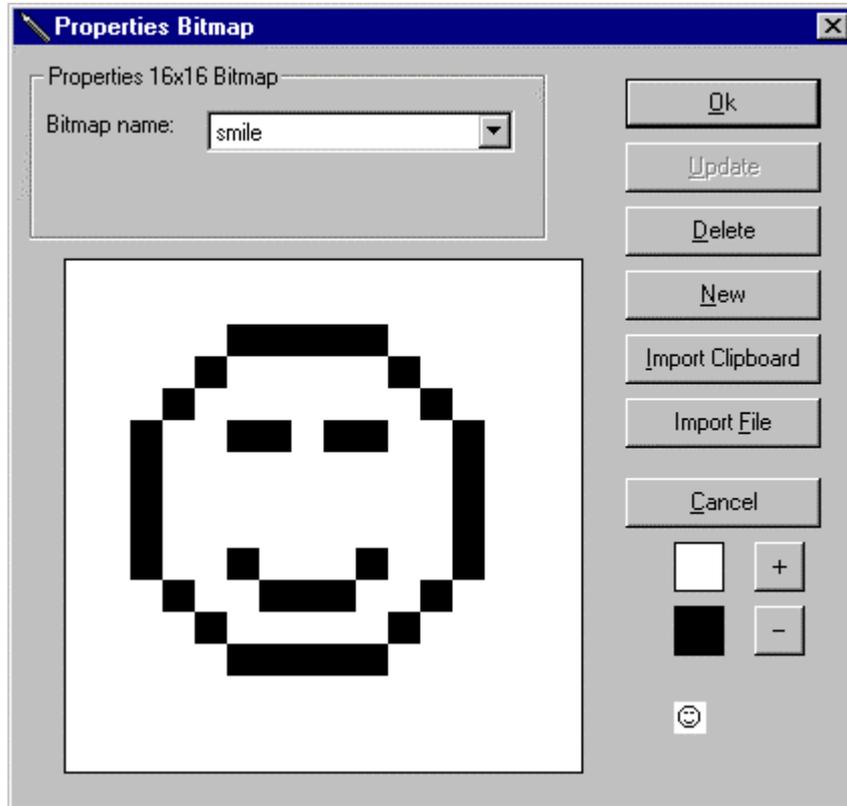
With  and  you can zoom in and out.

With **C**ancel your selection is discarded and you will get back to the menu.

With **O**k you paste the selected bitmap.

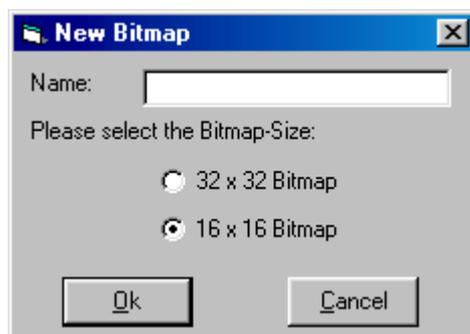
Create bitmaps

You can create a new bitmap with **Edit** and **Edit Bitmaps...** The following window will open:



At first a bitmap from the library will appear. Select **New** to start creating a new bitmap.

Zunächst erscheint eine *Bitmap* aus der vorhandenen *Bitmap Library*.
New wählen, um eine neue *Bitmap* zu erstellen.

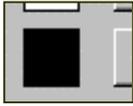


Name and size of a new *bitmap* have to be defined. The graphic display will for the time being remain unchanged.



Choose the field with the cursor. Now you can paint over the bitmap pixel by pixel by means of the cursor

To create a new bitmap, select  with the cursor. Now you can paint a new paint bitmap
bitmap pixel by pixel.

By changing between  and  you can change your
bitmap at any time.

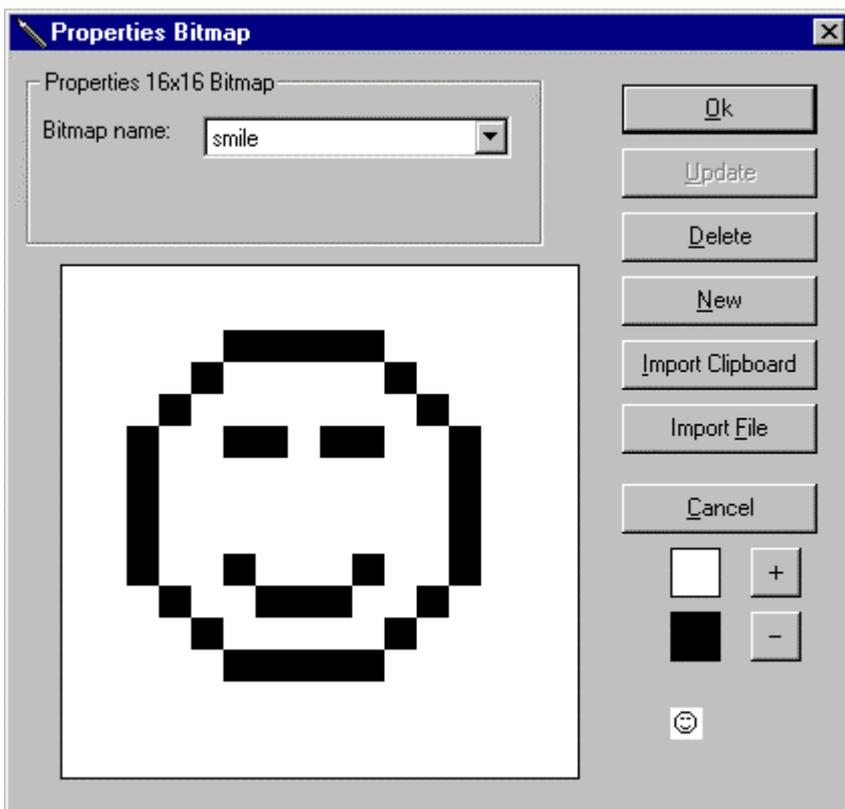
With  and  you can change the view of the bitmap to get an
impression of the result. In the right-hand corner of the window **Properties Bitmap**,
the smallest view of the bitmap is shown all the time.

Click **Update** when you have finished the new bitmap. The bitmap will then be saved in
the bitmap library.

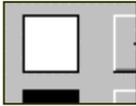
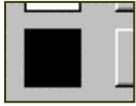
Click **Ok** to close the window **Properties Bitmap**.

Edit bitmap

You can edit a bitmap with **Edit** and **Edit Bitmap...** and you will open the following
window:



At first a bitmap from the existing library will appear. With **Bitmap name**, you can choose the bitmap you want to edit.

To change the bitmap choose  or  by means of the cursor. Now you will be able to paint your changes with the cursor.

With  and  you can change the view of the bitmap to get an impression of the result. In the right-hand corner of the window **Properties Bitmap**, the smallest view of the bitmap is shown all the time.

To finish your changes, click **Update**. The changes of your bitmap will then be saved in your library.

Click **Ok** to close the window **Properties Bitmap**.

Delete bitmap

- Delete *bitmap* on the current page:

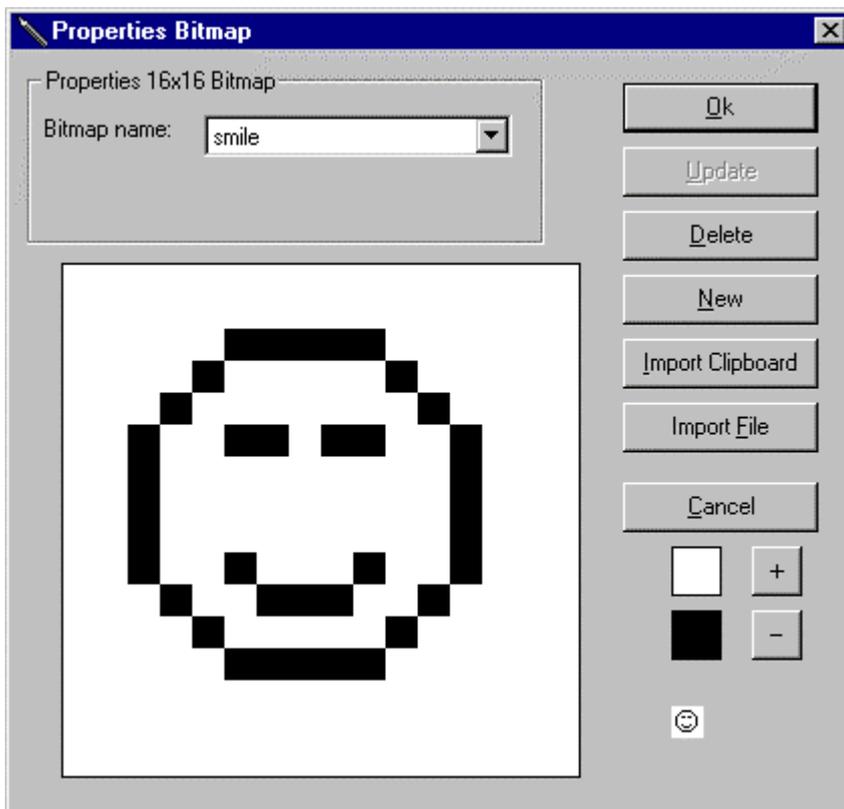
Mark the bitmap. You can now delete the bitmap by pressing the delete key or by

clicking



- Delete *bitmap* from the *bitmap library*:

With **Edit** and **Edit Bitmaps...** you will open the following window:



An arbitrary bitmap from the bitmap library will appear. In **Bitmap name** select the bitmap you want to delete. Afterwards click **Delete**. Now the bitmap is deleted from the library.

Simulation

With **Simulation** and **activated** you can simulate the application. This allows you to check the links that you have defined.

The **Ok** Button (press) and the two arrows (turn) are used to simulate the knob of the Infodisplay.



marked object

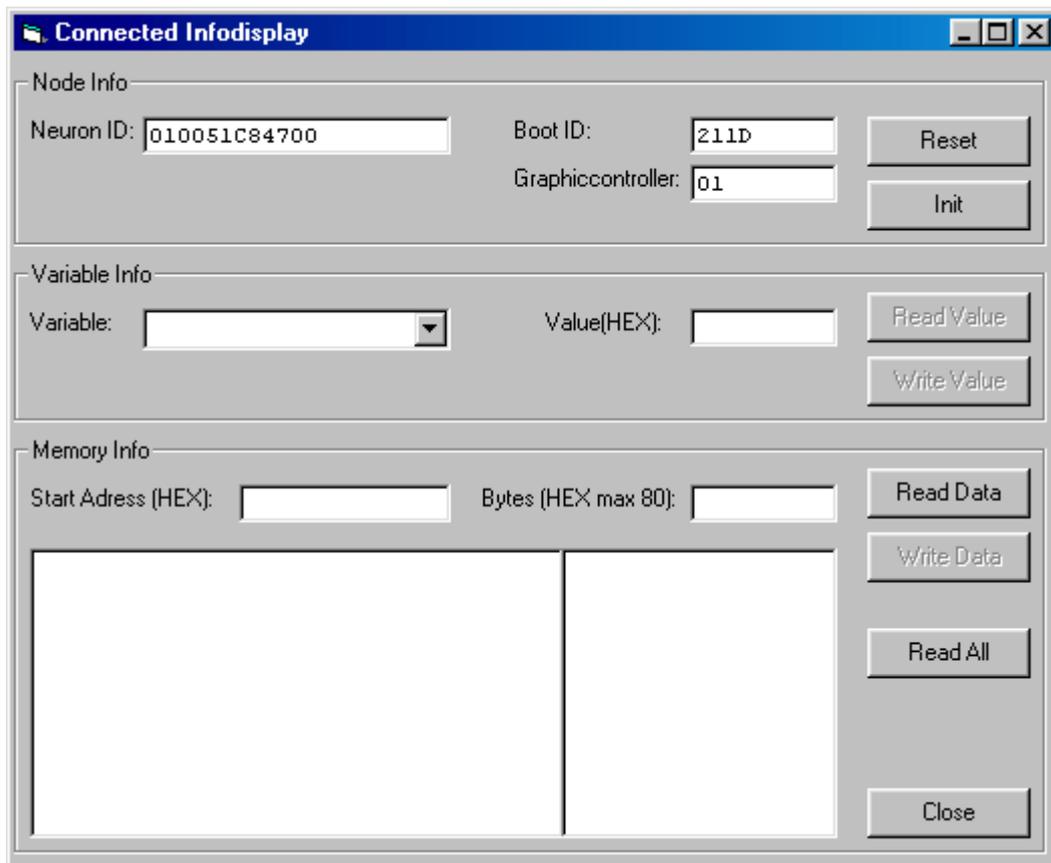
button to simulate
pressing of the
knob

arrows to simulate
turning the knob

Checking the contents of the variables

Use **Extras** and **Connected Infodisplay** for checking some of the data.

Depending on the version installed, the following window will open:



If "00" is shown for the graphics controller, the bottom end of this window will not be displayed.

Reset is used to reset the Infodisplay.

Init restores the initialization values in the Infodisplay that were set when configuring the variables.

Select **Variable** and **Read Value** to read out values from the Infodisplay; use **Write Value** to transfer data from **Value** (hexadecimal notation) into the Infodisplay.

Enter an address (**Start Address**) and a number of data bytes (**Bytes**) to read out and describe memory areas of the display. **Read All** reads all the storage areas (RAM, flash, data flash) in one go.

Other menu options

File, Export

Export in the menu bar and **File-Transfer-File** is used to generate a file, which can be exported by LonMark – File transfer in the form of “MENU Data“ into the Infodisplay.

Export in the menu bar **File** and **NeuronC** generates a NeuronC application file.

Creating the application for the Infodisplay

Normally the application file (.apb) is created by A3M. Once you have created the GUI, it is, therefore, necessary that you send the three project-specific files ("project name" .lst, "project name" .bib. and project name" .ife) by e-mail to the A3M address infoedit@a3m.com .

After a few hours you will receive the files "project name" .apb, "project name" .nxe, "project name" .xfb. and project name" .xif. If you need a different file format, please let us know by e-mail.

Loading the application into the Infodisplay

You can upload this file into the Infodisplay by means of any network management tool.

The menu data can

- be loaded into the Infodisplay via the serial interface (RS232) or
- be transmitted via file transfer through the bus line using a network management tool.

To send the menu data via the serial interface, you have to connect the Infodisplay to a serial port of your PC. Use **File**, **Options** and **Comm-Port...** for configuration of this port. Having done this, select **File** and **Send** to send the data.

A small window will be shown which allows you to watch the data transfer.

Additional notes

Should the message "incompatible menu" appear after automatic reset, this means that the application file (*project name.apb*) does not match the menu data in the Infodisplay.

Other status messages are:

N-ID means Neuron-ID
B-ID means Boot-ID
P-ID means Program-ID (in ASCII)

Configured/Online
Unconfigured

The yellow LED is the service LED when no application is activated. The service pin can be accessed through a small hole between the connecting terminals at the rear of the unit. For each reset or application restart, the Infodisplay will send a service pin message.

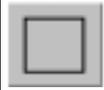
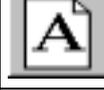
Pressing the input knob briefly means that an entry is made. Pressing the knob a little longer (max. 3 sec) will take you to the first page of the menu. Keeping the knob pressed even longer means that the application is restarted,
Turning the knob while keeping it pressed means that the contrast setting is changed; this setting will be adopted immediately and stored in the internal variable "Contrast".

Appendix 1: Bitmaps

	Infodisplay		Meter		Heat
	Clock		Blind		Water
	Alarm Clock		Room		Work
	Smile		Window		Power
	No Smile		Rain		Consumption Water
	Neutral Smile		Wind		Diagram
	News		Moon		Phone
	Question Mark		Sun		Temperature
	Contrast		Star		Plug
	Overview		Pollen		Traffic Light Green
	Back		Ozone		
	Go on		Info		
	Configuration		Calendar		
	Adjustment		Average		
	Copyright		Voltage		
	Computer		Current		
	Building		Costs		
	Pyramid		Address		
	Boiler		People		
	Tank		Bus		
	Power Plant Station		Car		
	Solar Plant		Traffic Light Red		
	Heating System		Traffic Light Top		
	Water		Traffic Light Middle		

	Light		Traffic Light Bottom		
---	-------	---	----------------------	--	--

Appendix 2: Icons

	Draw a line		Delete the current page
	Draw a rectangle		Cut marked objects
	Insert the clock		Copy marked objects
	Insert a textbox		Paste copied objects
	Insert a diagram		Delete marked objects
	Insert a bar chart		Edit the variable list
	Insert a bitmap		Edit the bitmap list
	Open a new project		Previous page
	Save the current project		Next page
	Insert new page		Simulation

Appendix 3, Pixel-Font

TYPETOOL FONT PRINTOUT

PixelPoint

06/02/00 02:40:39 PM

Page 1/2

24	25	26	27	28	29	30	31	32 space	33 exclam	34 quotedbl	35 numbersign
									!	"	#
36 dollar	37 percent	38 ampersand	39 quoteright	40 parenleft	41 parenright	42 asterisk	43 plus	44 comma	45 hyphen	46 period	47 slash
\$	%	&	'	()	*	+	,	-	.	/
48 zero	49 one	50 two	51 three	52 four	53 five	54 six	55 seven	56 eight	57 nine	58 colon	59 semicolon
0	1	2	3	4	5	6	7	8	9	:	;
60 less	61 equal	62 greater	63 question	64 at	65 A	66 B	67 C	68 D	69 E	70 F	71 G
<	=	>	?	@	A	B	C	D	E	F	G
72 H	73 I	74 J	75 K	76 L	77 M	78 N	79 O	80 P	81 Q	82 R	83 S
H	I	J	K	L	M	N	O	P	Q	R	S
84 T	85 U	86 V	87 W	88 X	89 Y	90 Z	91 bracketleft	92 backslash	93 bracketright	94 asciicircum	95 underscore
T	U	V	W	X	Y	Z	[\]	^	_
96 quoteleft	97 a	98 b	99 c	100 d	101 e	102 f	103 g	104 h	105 i	106 j	107 k
`	a	b	c	d	e	f	g	h	i	j	k
108 l	109 m	110 n	111 o	112 p	113 q	114 r	115 s	116 t	117 u	118 v	119 w
l	m	n	o	p	q	r	s	t	u	v	w
120 x	121 y	122 z	123 braceleft	124 bar	125 braceright	126 asciitilde	127	128 .0128	129 .0129	130 quotingslba	131
x	y	z	{		}	~		€	€	,	
132 quotedblbas	133 ellipsis	134 dagger	135	136 .0136	137 perthousand	138 Scaron	139	140 OE	141 .0141	142 .0142	143 .0143
„	...	†		‰	‰	Š		Œ	†	†	†
144 .0144	145 .0145	146 .0146	147 quotedblleft	148 quotedblright	149 bullet	150 endash	151 endash	152 .0152	153 trademark	154 scaron	155
‡	`	´	„	”	•	–	–	ˆ	™	Š	

TYPETOOL FONT PRINTOUT

PixelPoint

06/02/00 02:40:41 PM

Page 2/2

156 oe	157 .0157	158 .0158	159 Ydieresis	160 .0160	161	162 cent	163 sterling	164	165 yen	166	167
168 .0188	169 copyright	170 ordfeminine	171 guillemotleft	172 logicalnot	173 .0173	174 registered	175	176 degree	177 plusminus	178 twosuperior	179 threesuperio
180 .0180	181 mu	182 paragraph	183 periodcenter	184 .0184	185 onesuperior	186 ordmasculine	187 guillemotrigh	188 onequarter	189 onehalf	190 threequarter	191 questiondow
192 Agrave	193 Aacute	194 Acircumflex	195 Atilde	196 Adieresis	197 Aring	198 AE	199	200	201 Eacute	202	203 Edieresis
204 Igrave	205 Iacute	206 Icircumflex	207 Idieresis	208 Eth	209 Ntilde	210 Ograve	211 Oacute	212 Ocircumflex	213 Otilde	214 Odieresis	215 multiply
216 Oslash	217 Ugrave	218 Uacute	219 Ucircumflex	220 Udieresis	221 Yacute	222 Thorn	223 germandbls	224 agrave	225 acute	226 acircumflex	227 atilde
228 adieresis	229 aring	230 ae	231 cedilla	232 egrave	233 eacute	234 ecircumflex	235 edieresis	236 igrave	237 iacute	238 icircumflex	239 idieresis
240 eth	241 ntilde	242 ograve	243 oacute	244 ocircumflex	245 otilde	246 odieresis	247 divide	248 oslash	249 ugrave	250 uacute	251 ucircumflex
252 udieresis	253	254 thorn	255 ydieresis	256 .notdef	257 .null	258 CR					

Appendix 4: Definitions

Array

Allows you to define arrays of identical network variables. These variables are provided with an index to be able to access them.

Bitmap Library

A file which is needed for a project (*.bib). It contains all bitmaps created in, or imported into, your project. Together with the Infoedit you will get a bitmap library containing 60 bitmaps for your own use. These bitmaps are listed in the appendix.

Destination Variable

Variable into which the data will be written. This may be a network variable or a local variable.

Differential Type

If, instead of the original value, you want to save the difference between two values measured one after another, for example in a diagram, you have to select "Yes".

Enum Definition

To create an enumeration type (e.g. Monday, Tuesday, Wednesday,....; instead of 1, 2, 3,...) you can use the page *Enum Definition*.

Initialize

Here you can define a standard value for the variable. This value will be shown when starting the Infodisplay for the first time or a after prolonged power failure. Initializing starts with a checksum error in the battery-buffered RAM.

Internal Variable

To be able to use an SNVT in your menus, you have to connect it to an *internal variable*. Some *internal variables*, such as LED, Real Time Clock, Contrast, are already defined to support the Hardware..

Offset

In this box you have to enter a constant value, which is to be added to, or subtracted from, the *source variable*.

Polling Cycle

This is the time-controlled-repeated-read-out-cycle from input variables

Project

A project is a specific application for the Infodisplay. It consists of three different types of files:

- Files containing the GUI of the Infodisplay, ending with .ife
This GUI is created by means of Infoedit. It contains a minimum of one page.
- Files containing the bitmap library, ending with .bib
Infoedit will come complete with a basic number of bitmaps for your first applications. They are contained in the file **bitmap.bib**: Appendix A provides a list

- of these bitmaps
- Files containing the variable list, ending with .lst

To **import** files of different projects into your current project, click **File** and **Import**.

SD String

Self documentation string, some short information about the network variable, which can be written by yourself.

SNVT

To communicate with the LONWORKS® network, you have to define network variables that are available in your network and are to be shown on the Infodisplay. For this purpose, the LonMark® Standard has defined “standard network variable types” (SNVTs).

These network variables can be configured by means of dialogues to create them and to visualize them on the Infodisplay (62 SNVTs per Infodisplay can be created with Infoedit)

SNVT Bind Info

Additional variable properties (see “Echelon Neuron C Programmer’s Guide“)

SNVT Class

Additional variable properties (see “Echelon Neuron C Programmer’s Guide“)

SNVT List

List which contains all SNVTs from the LONMARK SNVT Master List.

SNVT Modifier

Additional variable properties (see “Echelon Neuron C Programmer’s Guide“)

System clock

The Infodisplay variables also include some *internal variables*, for example definition of the time (variable “Internal Time”). If you use this time, you use the internal Real Time Clock of the Infodisplay. To allow this clock to be set , it has to be editable in your menus.

Source Variable

This is the variable from which you take the data. It may be a network variable or another local variable, which is to be saved or displayed in a different format.

Variable List

A file which is needed for a project (*.lst). It contains all the variables created in or imported into your project.