microHMI v1.0.1 – User manual

Index

Introduction

This document is a manual for microHMI, a software initially developed as a thesis Project for the Central University of Venezuela. MicroHMI is a Human-Machine Interface environment that allows easy and dynamic data acquiring and visualization of data from elements connected to a microcontroller, such as motors, sensors, valves, etc. The data is shown on a display and this document will guide you through setting such display. The author of this guide assumes a reader possesses at least basic knowledge of informatics.

Here you will find descriptions of each one of the interface's screens, their elements and how to use the functions contained in this software.

Basics

Human-Machine Interface (HMI)

Software that allows visualization of the parameters involved on a process (typically an industrial process). This visualization often requires the use of external data acquisition devices that are independent from the interface such as sensors. A HMI may include control routines for the parameters measured. A HMI system is usually part of a SCADA system (Supervisory Control And Data Acquisition) which controls the HMI behavior and links it to external data acquisition devices.

Microcontroller board

Is an integrated programmable circuit capable of executing instructions stored in its memory. Practically, it could be said that it is a simplified computer executing a single set of instructions in an endless loop. Its main component is a microchip which stores instructions and executes them. The microcontroller – or board - also possesses data input and output points – pins, which connect the board to external elements. Finally there's the main port which allows the connection with a computer used to load the instructions onto the board. On this guide we will refer to a microcontroller as "remote".

Scan routine or scan

Is the part of the interface that acquires and manages data from the remote. It is the software's critical process and a forced interruption (e.g. unplugging the remote) can cause a fatal error.

Display

This screen allows to establish the graphic setting of a scan, including dynamic widgets such as bars and led-like widgets that allow an user-friendly visualization of the process' data.

Point

Each physical element connected to a remote and its virtual counterpart in the interface.

Configuration Database

Contains settings for a remote and its points. It must be established before a scan as it is necessary for the scan’s execution.

Before starting

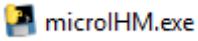
MicroHMI is built upon a single Python script using Kivy as its graphical toolkit. The software has been compiled in a standalone executable that includes the Python libraries necessary for its execution. However the user must take in account that since Kivy v0.9, OpenGL ES 2.0 is required for its functioning. OpenGL ES 2.0 is the standard graphic API for the majority of modern computers and for almost all mobile devices. OpenGL version can't be modified without deep graphics software and hardware knowledge and might even require changing of the graphic hardware in our computer.

Remember the program will fail if a remote is disconnected while executing a scan routine.

Starting microHMI

Execution and the presentation screen

MicroHMI is a standalone program and its installation it does not need to be installed. If you have all required folders you can execute it by double clicking on the *microIHM.exe* icon in *C:/MyDirectory/microHMI/dist/microHMI* (replace *MyDirectory* for the folder where you stored microHMI)



This will open the presentation screen. It contains the basic and legal information of the software and a button that allows us to start it.

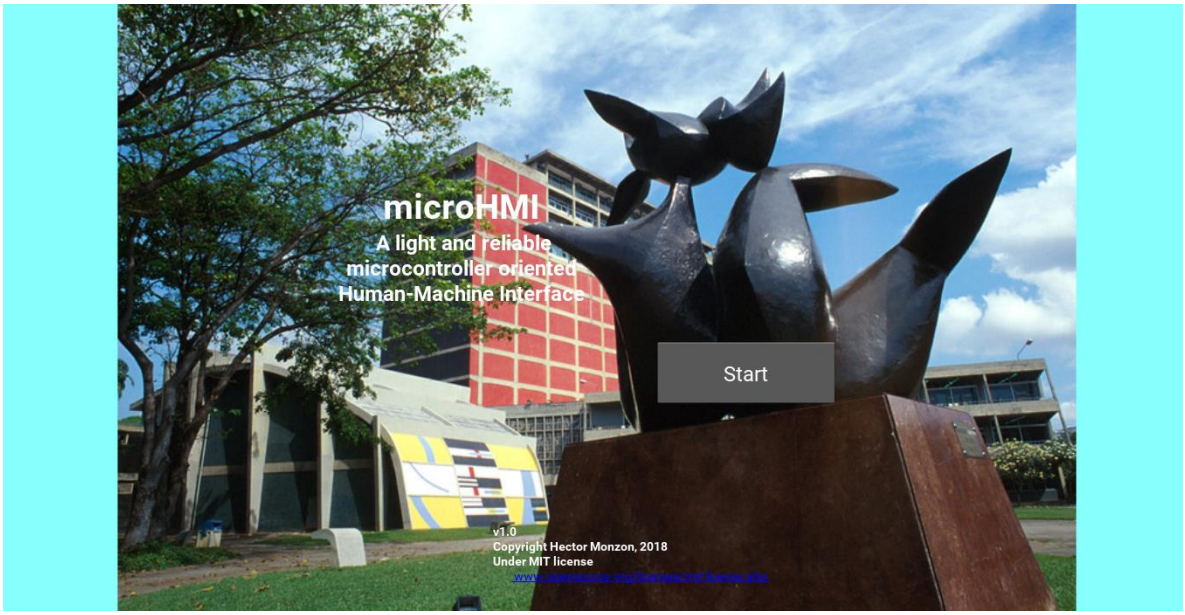


Illustration 1 – Presentation screen

Setting up a scan

The Database screen

The data that goes through microHMI is stored in a database. The user needs to configure a valid database in order to set up a scan routine. It is critical to the program that the fields on a database are completed coherently because erroneous data could lead from small malfunctions to critical failure of the software.

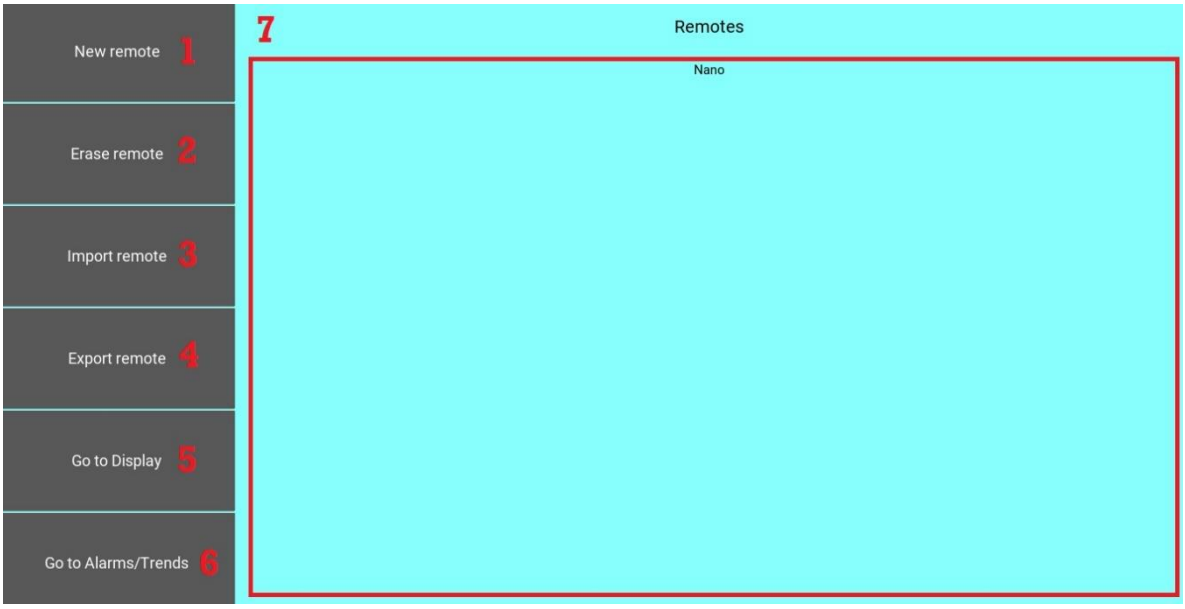


Illustration 2 – Database screen

Contents

1 – Manually include a new remote

2 – Erase a selected remote

3 – Import a remote from a .txt file

4 – Export a selected remote to a .txt file

5 – Go to the Display screen

6 – Go to the Alarms/Trends screen

7 – List of current remotes. Every remote can be selected with a click and visualized with a double click

Manually adding a remote

On pressing the "New remote" button the following remote configuration popup will be displayed:

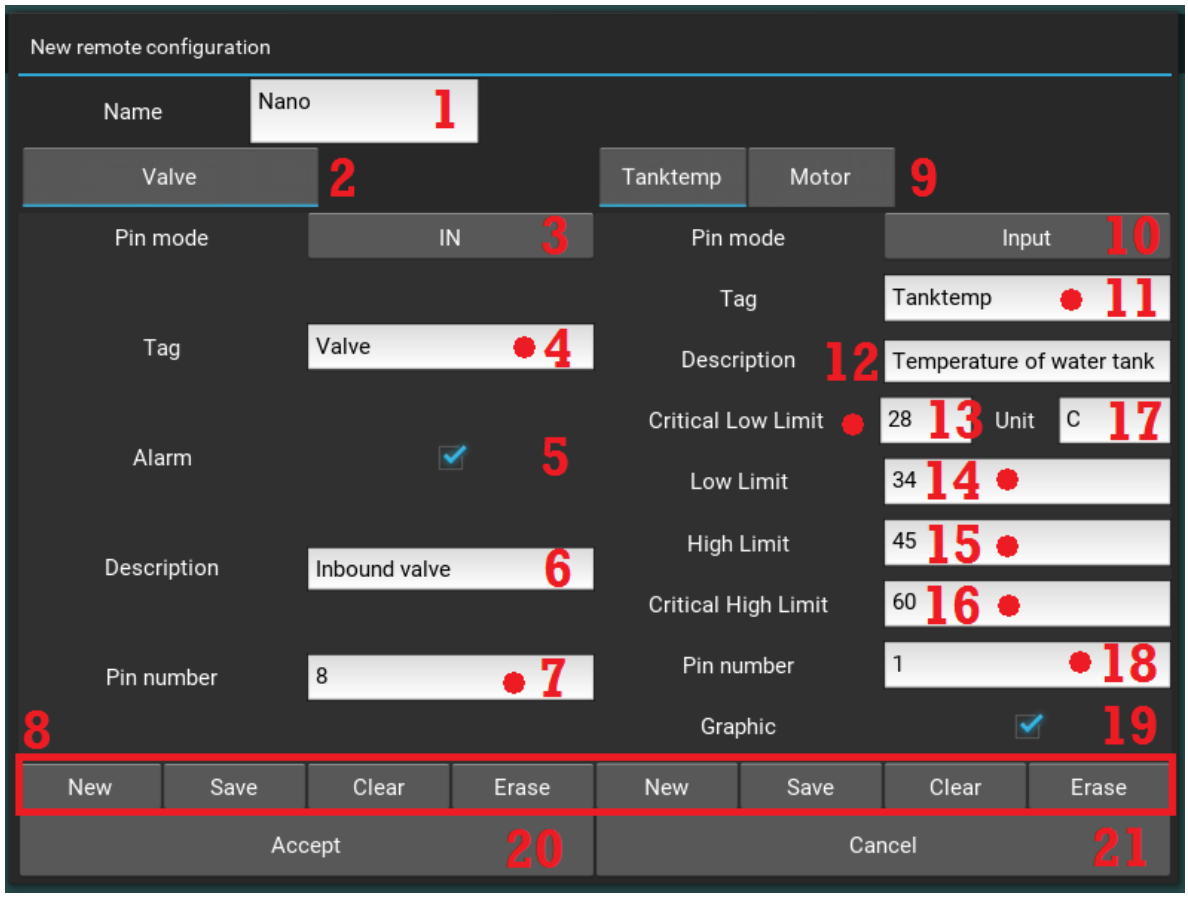


Illustration 3 – Example of a popup where the fields have been completed for a Digital and an Analogic tab. Red dots indicate critical fields.

Popup contents:

1 – Remote name

2 – Digital points tab area

3 – Toggle button for pin mode

4 – Point tag field (identifies the point)

5 – Checkbox that indicates if a point should raise alarms. Only for input points

6 – Point description field

7 – Pin number field. Indicates which board pin is to be read

8 – Tab navigation toolbar: "New" adds a new tab, "Save" writes the point to database, "Clear" cleans the fields on the tab and "Erase" suppresses current tab

9 – Analogic points tab area.

10 – Pin mode toggle. Analogic tabs have two modes: input and PWM. Clicking on this toggle button replaces current input tab for a PMW tab (and vicecersa) if current tab is empty

11 – Point tag field

12 –Point description field

13 to 16 – Point limits. Remember to complete these fields coherently, e.g. "Low limit" can't hold a higher value than "High limit"

17 – Point readings unit field

18 –Pin number field

19 – Checkbox indicating if point readings should be plotted

20 – Create remote

21 – Cancel operation

A PWM tab is similar to a Analogic Input tab but instead of limits it holds Maximum and Minimum values because this type of element doesn't raise alarms.

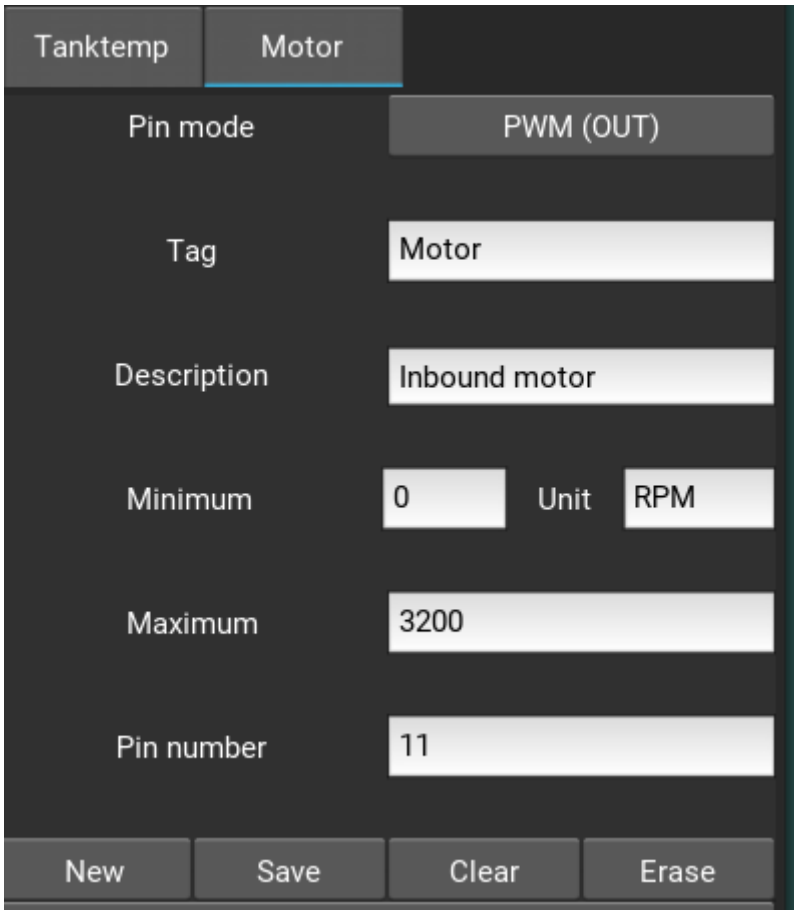


Illustration 4 – Complete PWM popup example

Each point must be saved to the database, otherwise they would be unaccesible from other database-dependent elements (widgets). In order to fix points to a database you should press "save" after filling all critical fields on each tab. After this, clicking on "Accept" will take you back to the database screen with the new remote added to the list.

Illustration 5 – The remote has been added to the list

Exporting and importing a remote

When a remote has been correctly set up its contents can be browsed by double-clicking on its name in the remotes list, which will show the popup seen on illustrations 4 and 5.

While a remote is selected, clicking on the "export remote" button on the database screen will open a popup that allows to name a file in which the information of the selected remote will be stored. The generated file is a .txt file and its structure allows it to be imported back into the software (see below). All exported files will be store on the "Reports" folder, on the microHMI's directory. In this folder two pre-made files can be found: "example.txt", which contains an example of the accepted import/export structure, and "structure.txt" which holds the empty import/export structure.

If the data for exporting is valid, a success message will appear after naming the file.

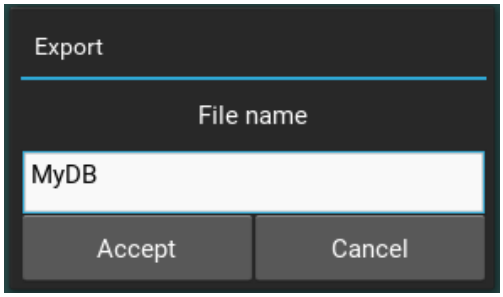


Illustration 6 – Export popup

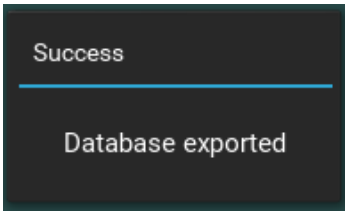


Illustration 7 – Export success message

Clicking on the "Import remote" button in the database screen will show a popup which allows to browse through the system files to find a valid import file. If the file is correctly imported, a success message will appear.

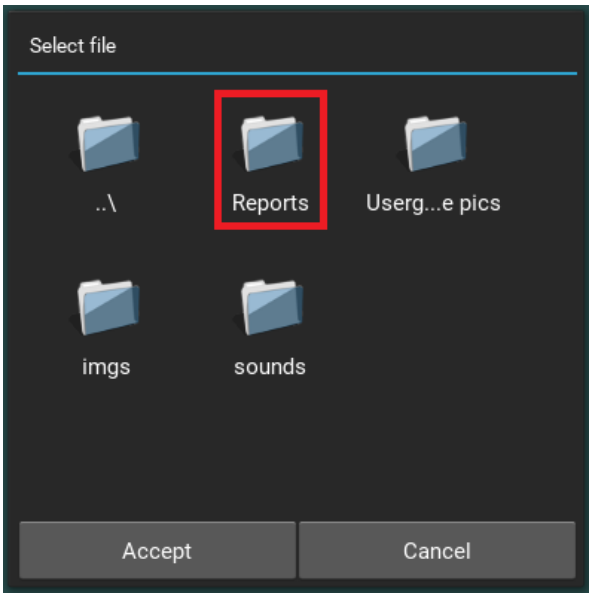


Illustration 8 – Import selection popup. All exported files will be store on the highlighted folder, as indicated above.

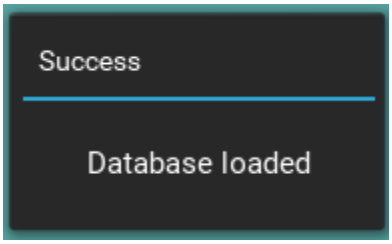


Illustration 9 – Successful import message.

Running a scan

The Display screen

Clicking on the “Go to Display” button in the database screen will lead to the display screen. In this screen a dynamic display can be set, such display allows to link widgets with assigned database points to physical electronic components connected to the microcontroller in order to visualize the components behavior.

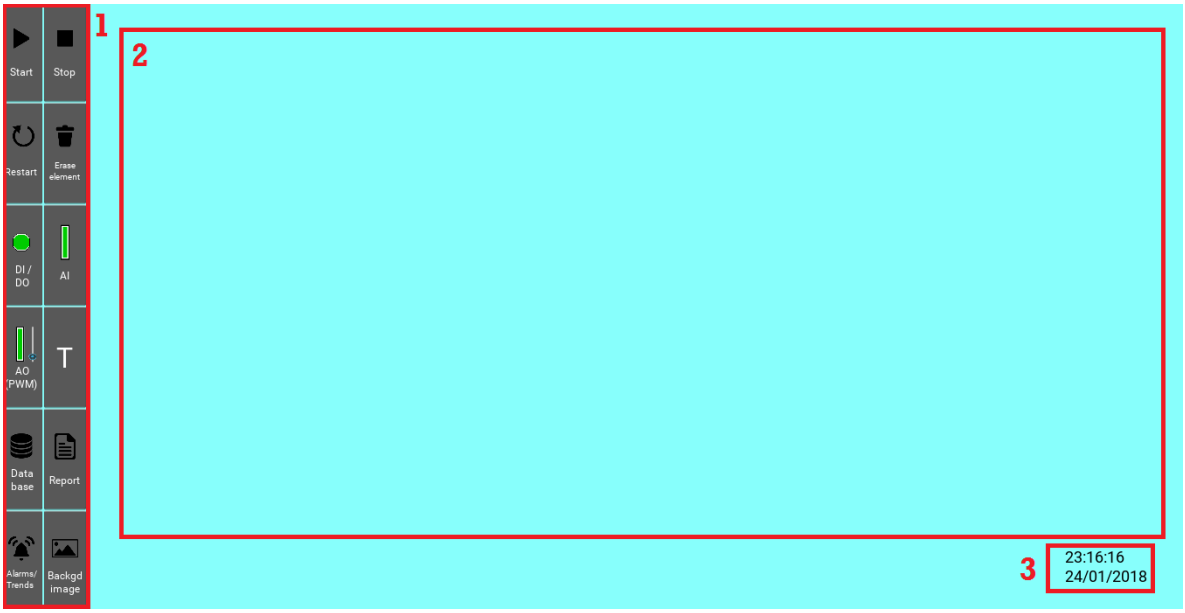


Figura 10 – Display screen

Contents:

1 – Toolbar

2 – Display environment

3 – Current date and time

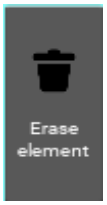
The toolbar

This is the main component that aids the user build a display. Its functions are explained below.

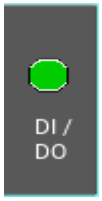
Starts a new Scan routine

Stops the current scan routine

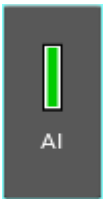
Restarts all the widget values to zero



Erase the selected widget



Adds a new digital widget

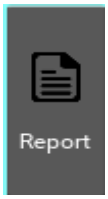


Adds a new analogic input widget

Adds a new PWM widget

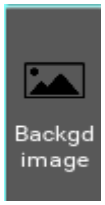
Adds a new textbox

Switches to database screen



Reports readings

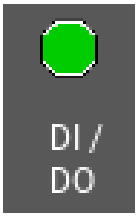
Switches to Alarms/trends screen



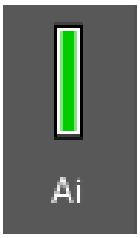
Allows selection of a new background image

Adding and linking display widgets

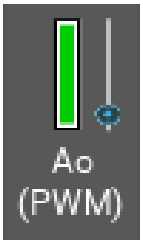
Next you will find a brief description of the three basic widgets used on the software.



Digital widget (input or output). Has two states (on or off) and indicates if a component is energized (green) or de-energized (red). When in output mode, clicking on the widget will change its state, turning the physical component on or off.



Analogic input widget. Shows the current value of an analogic component according to the scale set by the user. An alarm will be raised if the limits established for the component are broken, and the widget will change color (green is normal, yellow is for low or high limit broken and red for critical limits broken).



Analogic output widget (PWM). Can control an external component using the gage included on the widget. The bar displays the current value sent to the component according to the scale set by the user (maximum and minimum values)..

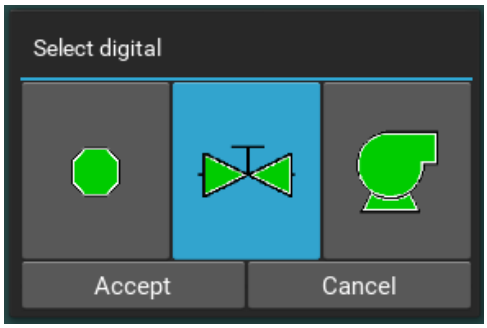


Illustration 11 – Digital widget subtype selection popup. Subtype "Valve" has been selected.

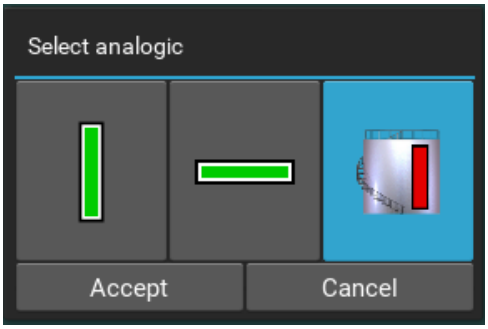


Illustration 12 – Anaalogic input widget subtype selection popup. Subtype "Tank" has been selected.

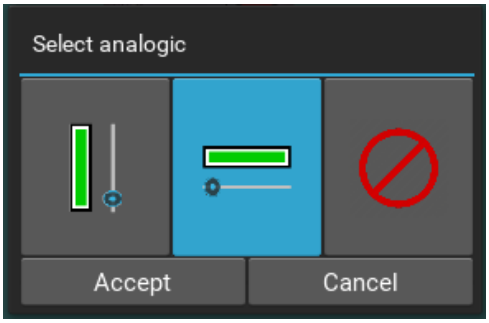


Illustration 13 – Analogic output widget subtype selection popup. "Horizontal bar" has been selected.

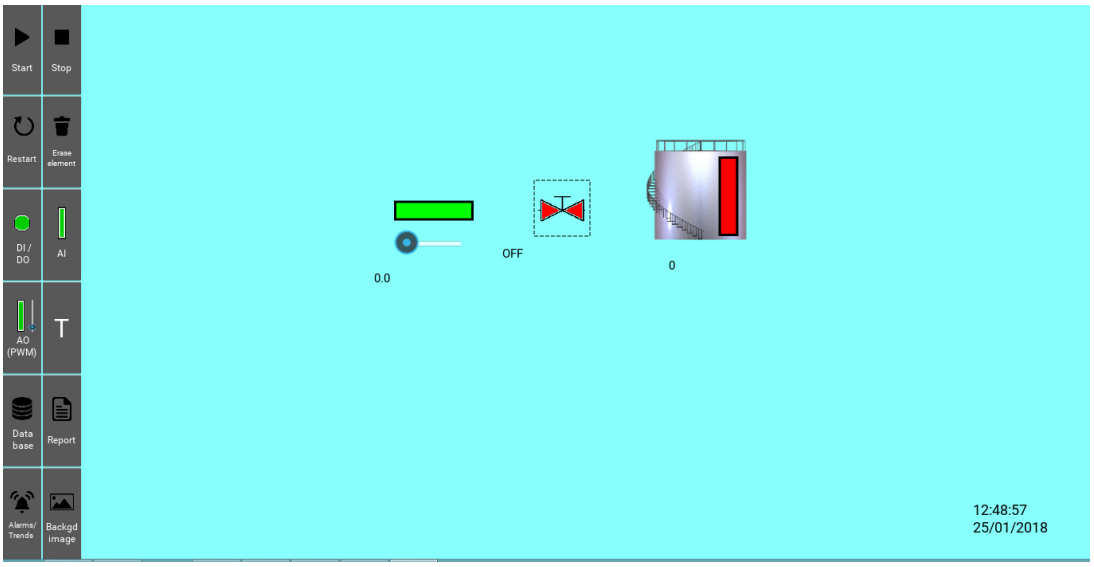


Illustration 14 – The three widgets have been added to the display. "Valve" widget is selected.

All widgets can be dragged through the display area unless a scan has been started. Double clicking on any widget will open a popup allowing to link a widget to a database element. Using the dropdown menus on this popup a remote and a point can be selected. If a widget has been correctly linked to a point it will show the point's information. Please note that a point can only be linked to a single widget.

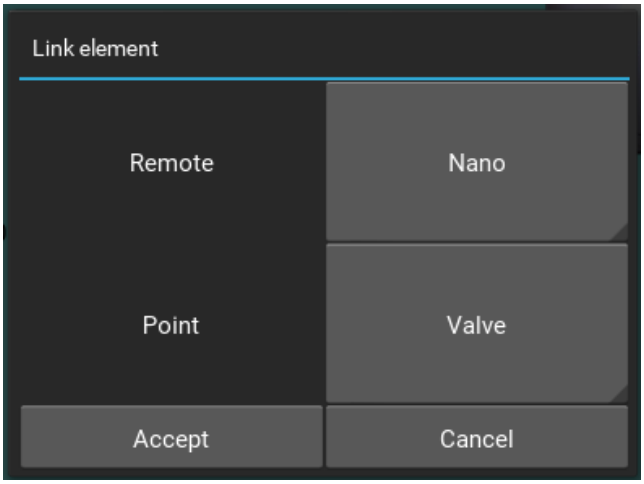


Illustration 15 – Widget linking popup for "Valve"

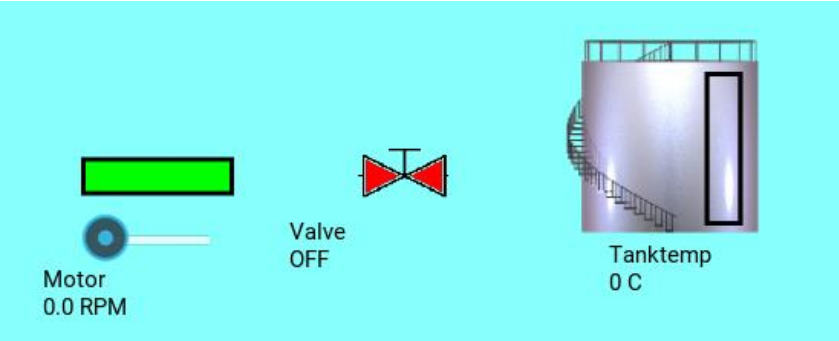


Illustration 16 – Three correctly linked widgets. Point tags and minimum values or current states are shown below each widget.

Running a scan

Once at least one element is linked to the database a scan routine can be initiated by clicking the start button on the toolbar. This will open a popup that allows to select auto detected ports to which remotes are connected. An error message will appear if there are no widgets on the display or if no widgets have been linked.

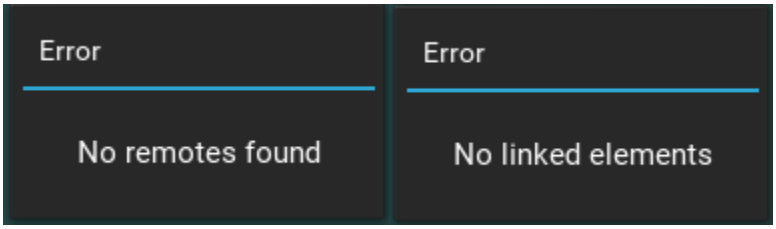


Figura 17 – Error messages.

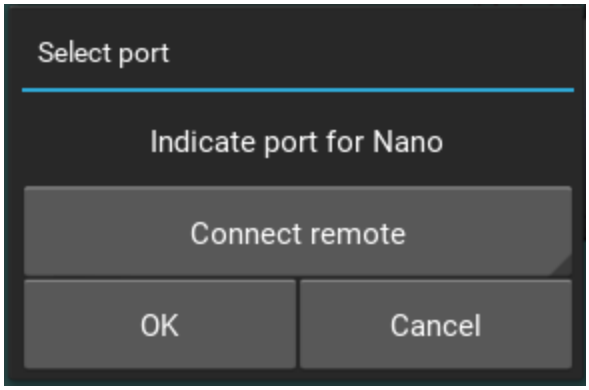


Figura 18 – Port selection popup.

The dropdown list present in this popup will not hold any elements if no remote is connected to the computer running the software. Selecting a valid port and clicking "OK" will initiate the scan routine.

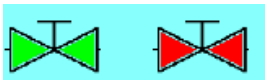


Illustration 19 – Digital widget states.

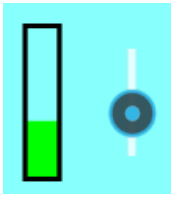


Illustration 20 –PWM widgets may be controlled using the gage included on the widget.

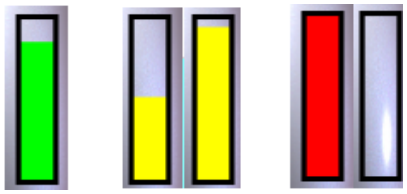


Illustration 21 – Analogic input widget states. Normal (green), over/under high/low limits (yellow) and over/under critical high/low limits (red).

If limits for an analogic input widget are broken or an alarmed digital input widget state changes, an alarm will be raised. This will cause a button to show on the display, which allows to go to the Alarm/Tendencies screen (also accessible from the toolbar).

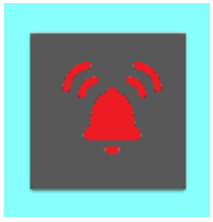


Illustration 22 – Alarm event button.

As another alarm indicator, a high-pitch beep will be emited while a widget is on alarm state.

The Alarms/Trends screen

This screen shows the plots designated by the user (only analogic input widget values can be plotted) and allows alarm recognition. Alarms are raised under two situations:

- Analogic input widget established limits are broken
- State change of a digital input widget (if it has been configured to raise alarms)

Alarms may be recognized by the user, disappearing from the workspace but still present on the alarms database.

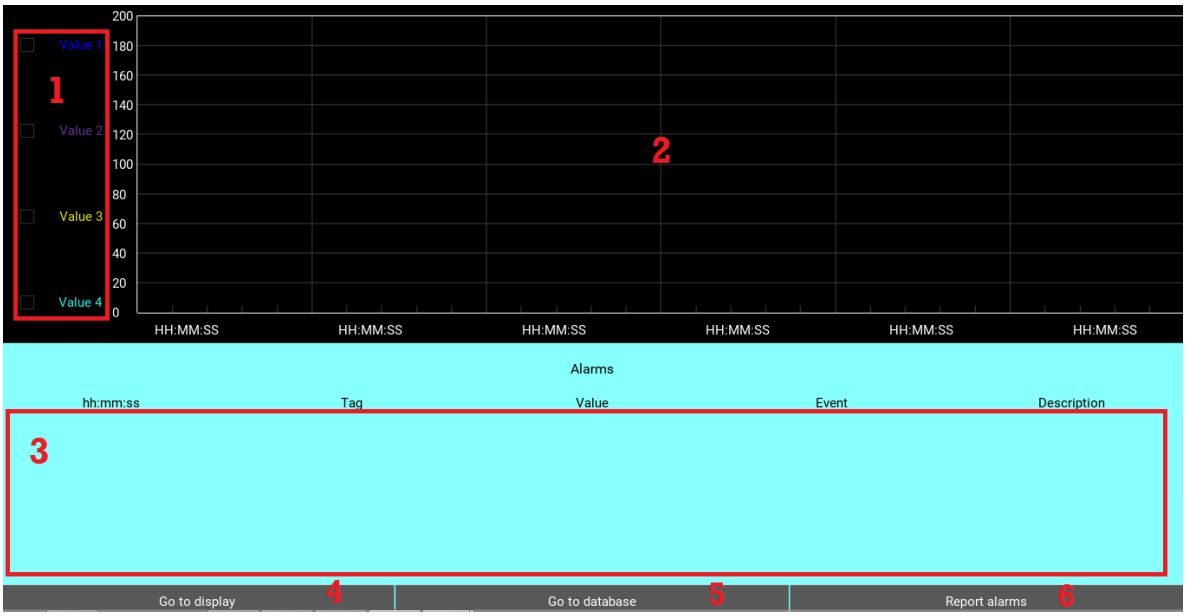


Illustration 23 – Alarm/Trends screen

Screen contents:

1 – Plot tag area. Tags indicate the name of the plotted widgets and hold the same color as the plot line. Checkboxes indicate plot visibility.

2 – Real-time plot area

3 – Selectable alarm list. It controls alarm recognition. Alarms on this list will blink until recognized (with click): if the alarm is still happening the blinking will stop but the alarm will remain on the list. If the alarm has stopped, it will disappear from the list.

4 – Leads to Display screen.

5 – Leads to Database screen.

6 – Alarm report button (see reports section).

Please note that only 4 plots can be shown per program execution.

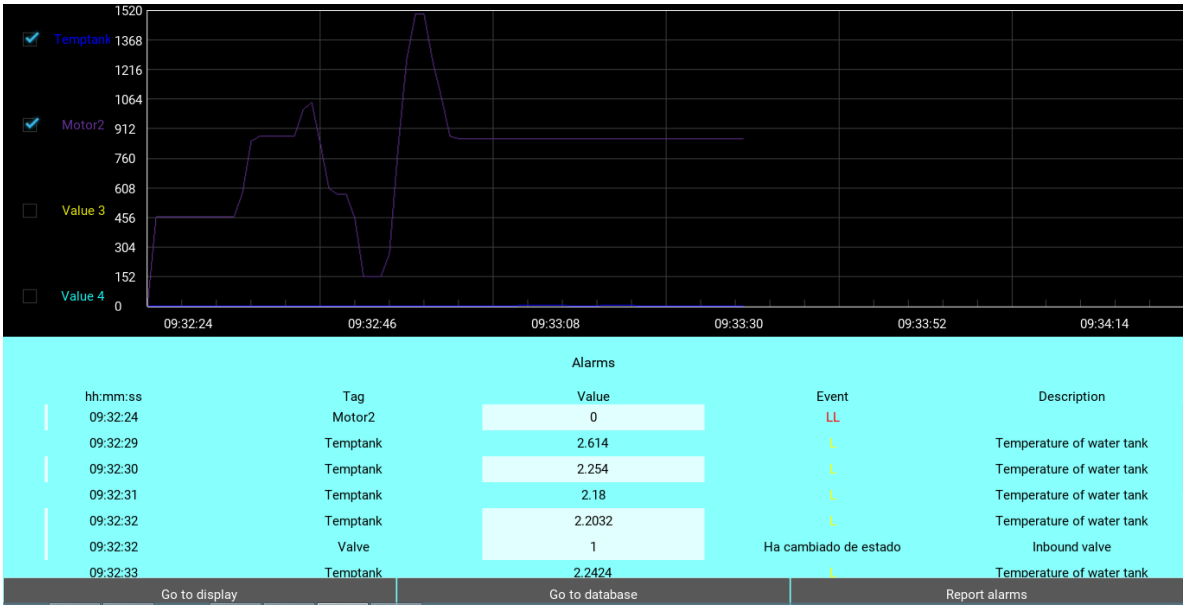


Illustration 24 – Alarms/trends screen while running a scan.