FeSCADA

Frandos Engineering LLC

User Manual

Windows software for machine/process visualization, data recording, email alerting.

Silviu Frandos
sfrandos@gmail.com

Contents

1. Quick Start Guide

Introduction

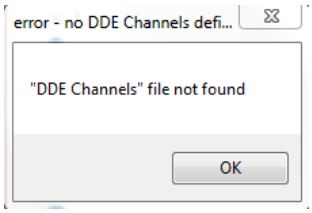
The guide will help the user to develop a new small project with *FeSCADA* software. This guide will cover the following topics:

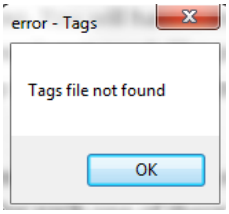
1.1. Program start
1.2. Setup of data sources
1.3. Tags creation and definition
1.4. Edit of screen zero, the first screen
1.5. Controls creation and properties
1.6. Assignment of tags to controls (push buttons, indicators, etc.)
1.7. Application testing
1.8. Application development - Adding numerical displays
1.9. Application development - Adding a new screen
1.10. Conclusions and next steps

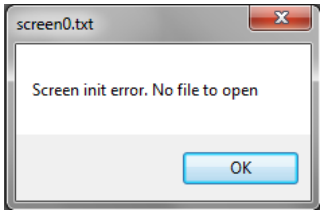
1.1) Program start

Create a new folder and copy the files *FeSCADA.exe* and *FeSCADA.lic* in it. If you do not have the file *FeSCADA.lic*, run the program *FeSCADA.exe*. You will get a message that the license was not found and another message that the license is not valid. The program will end after it will create a new file: *Hardware.lic*. This file should be sent to the software provider, who will generate a *FeSCADA.lic* file to match your hardware.

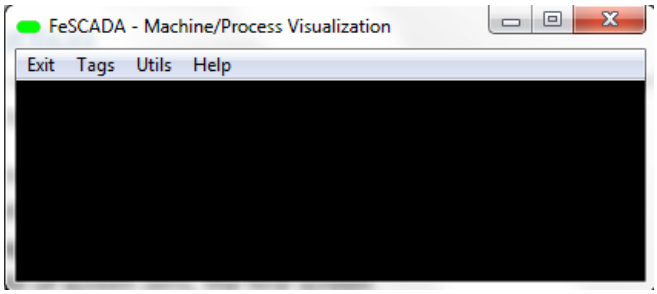
If you have the right license the program will start trying to read the configuration files for: *DDE channels*, *Tags*, and *screen0*. You will get an error for each one of these missing files.





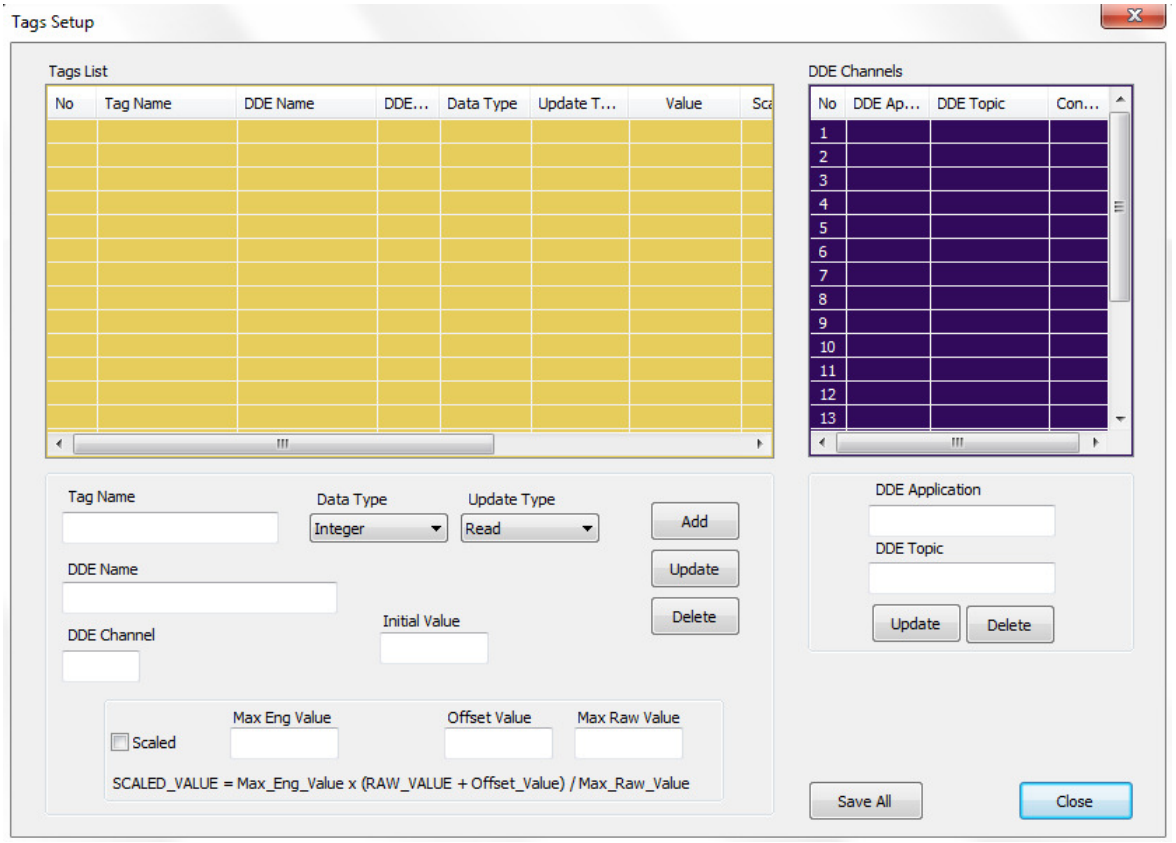


After pressing OK for all three messages the main window will show empty and with the following main menu: *Exit*, *Tags*, *Utils,* and *Help*.

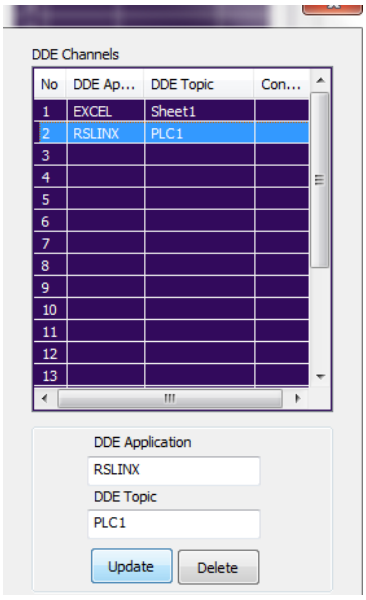


1.2) Setup of data sources

The *Tags* menu option will open the **Tags Setup** dialog window. In the right side is the **DDE Channels** list, where maximum 20 communication channels can be defined. In the left side is the **Tags List**.



First define two DDE channels: *EXCEL – Sheet1* and *RSLINX – PLC1*. For this, select the row 1 in the **DDE Channels** list, type *EXCEL* in the **DDE Application** text box, type *Sheet1* in **DDE Topic** text box and press **Update** button. After that select the row 2, type *RSLINX* in **DDE Application**, *PLC1* in **DDE Topic** and press **Update** button.



Remember that channel number 1 is for *EXCEL* and channel number 2 is for *RSLINX*. You will need this numbers when you define tags.

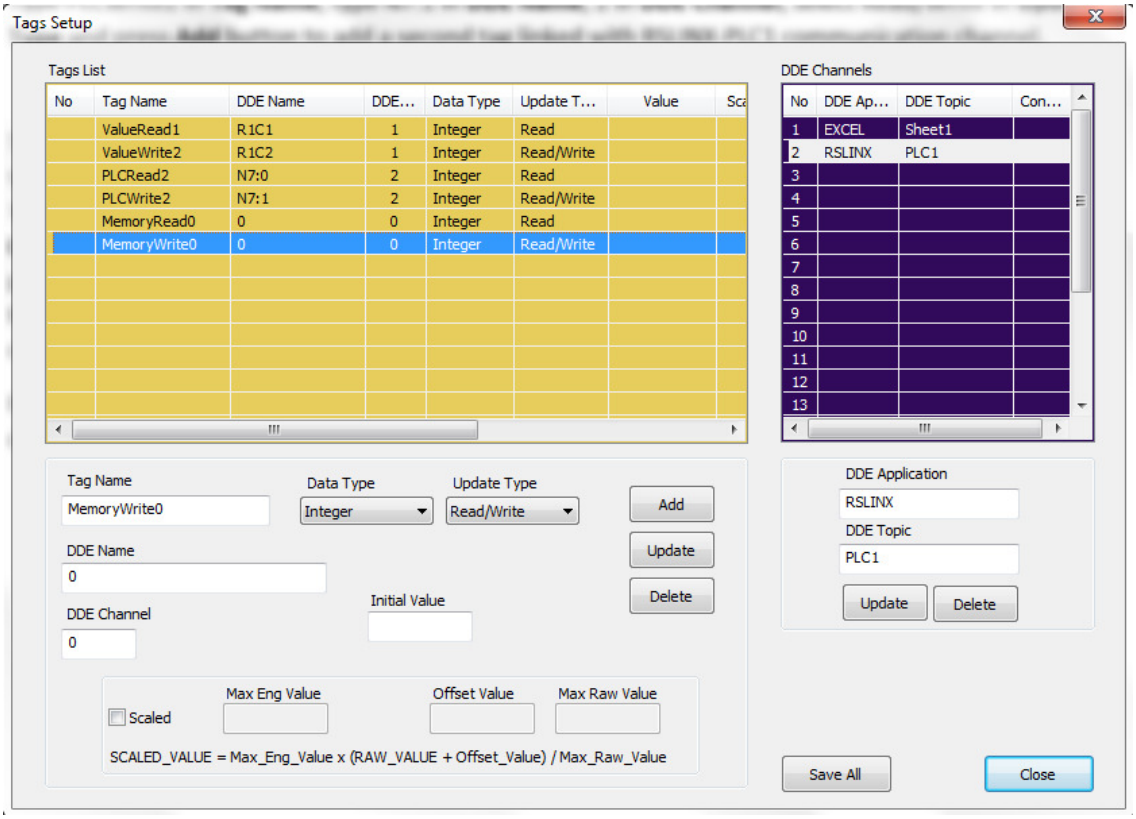
1.3) Tags creation and definition

To define a tag linked with *EXCEL-Sheet1* communication channel, type the tag name *ValueRead1* (or at your choice) in the **Tag Name** text box, type *R1C1* in the **DDE Name**, type *1* in the **DDE Channel**, select *Integer* in **Data Type** combo box, select *Read* in **Update Type** combo box and press **Add** push button. Type *ValueWrite2* in **Tag Name**, type *R1C2* in **DDE Name**, *1* in **DDE Channel**, select *Read/Write* in **Update Type** and press **Add** button, to add a second tag linked with *EXCEL-Sheet1* communication channel.

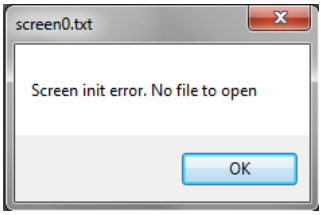
To define a tag linked with *RSLINX-PLC1* communication channel, type the tag name *PLCRead2* in the **Tag Name** text box, type *N7:0* in the **DDE Name**, type *2* in the **DDE Channel**, select *Integer* in **Data Type** combo box, select *Read* in **Update Type** combo box and press **Add** push button. Type *PLCWrite2* in **Tag Name**, type *N7:1* in **DDE Name**, *2* in **DDE Channel**, select *Read/Write* in **Update Type** and press **Add** button, to add a second tag linked with *RSLINX-PLC1* communication channel.

The program uses channel number *0* (zero) to define tags in the memory of the computer. To define a tag in the memory of the program (with no link to a communication channel), type the tag name *MemoryRead0* in the **Tag Name** text box, type *0* in the **DDE Name**, type *0* in the **DDE Channel**, select *Integer* in **Data Type** combo box, select *Read* in **Update Type** combo box and press **Add** push button. Type *MemoryWrite0* in **Tag Name**, type *0* in **DDE Name**, *0* in **DDE Channel**, select *Read/Write* in **Update Type** and press **Add** button.

If you make a mistake or you want to change the settings, select the row in the **Tags List**, correct the text, or combo box choice, and press **Update** push button. You can delete a tag by selecting the row and pressing the **Delete** button.



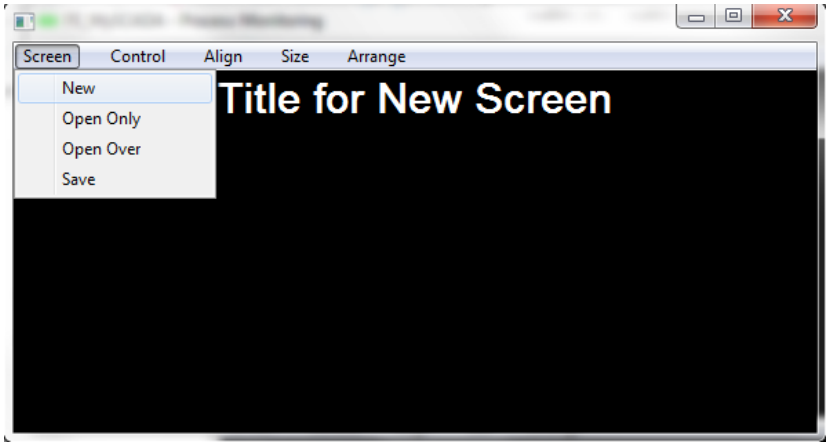
Press **Save All** and then **Close**. This will save the configurations for *DDE Channels* and *Application Tags*. The changes will be active when you restart the program. So, close *FeSCADA* program and restart it. After restart you will see only the "Screen init error" message box.

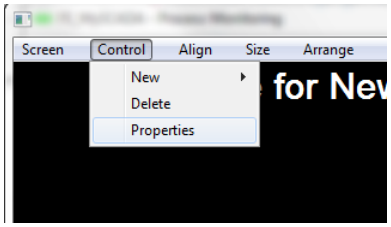


1.4) Edit of screen zero, the first screen

To edit the screen zero select from the main menu **Utils -> Screens Edit**.

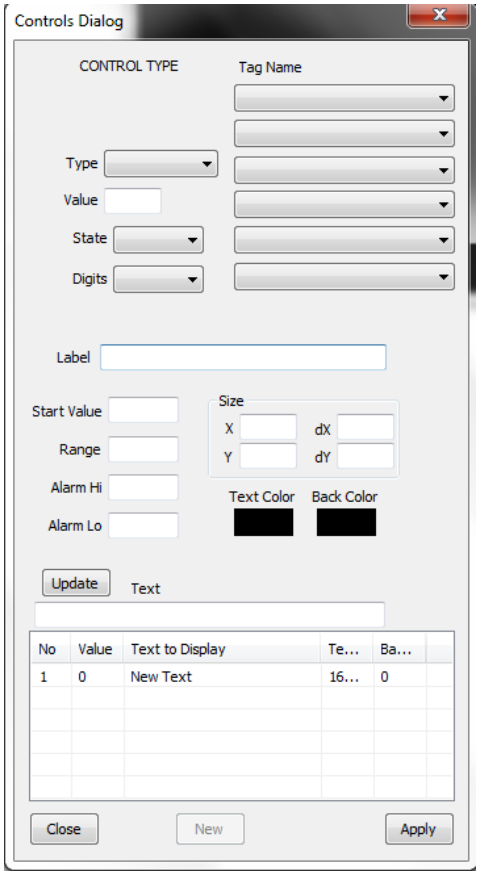
A new window will open with a different menu. This is the *Edit Screens* window. From the new menu select **Screen -> New**. After that select **Control -> Properties** to open the **Controls Dialog** properties panel.

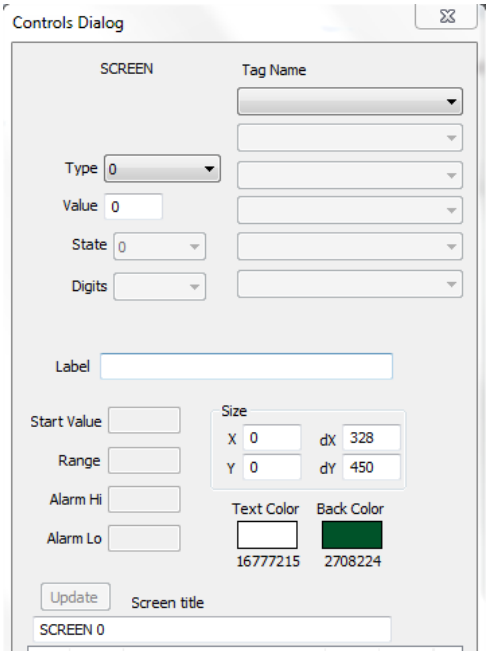




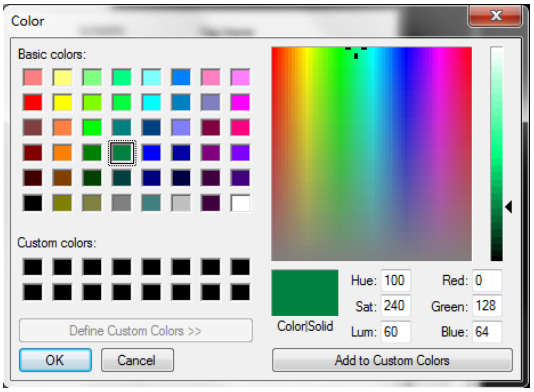
The **Controls Dialog** can stay open in the same time with the edit window. If it is closed you can open it at any time.

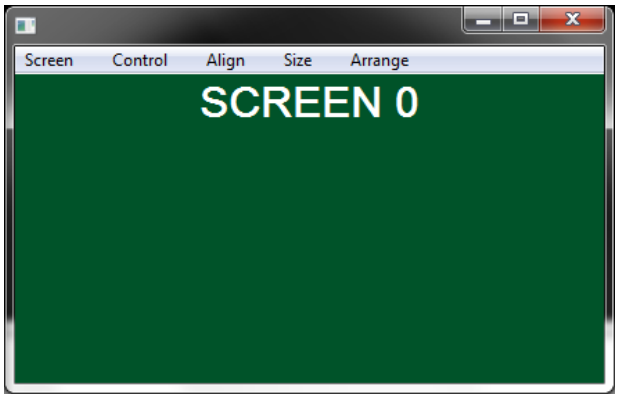
In the new screen you can change the background color and the title text and color. To do this click on the screen and the properties will be retrieved in the properties panel.





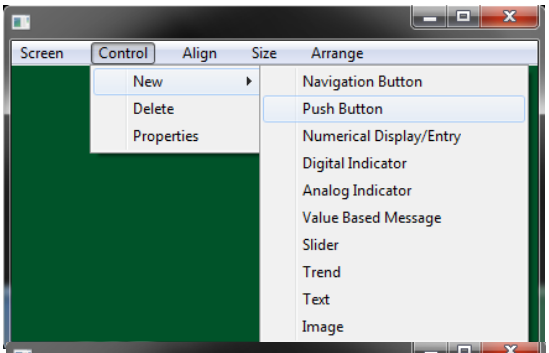
Go to **Screen title** edit box and type *SCREEN 0*, click on **Back Color** and **Text Color** and select your prefered colors. Click **Apply** for the changes to take efect.

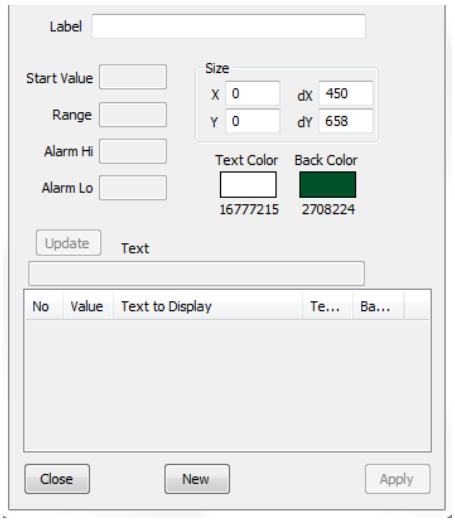


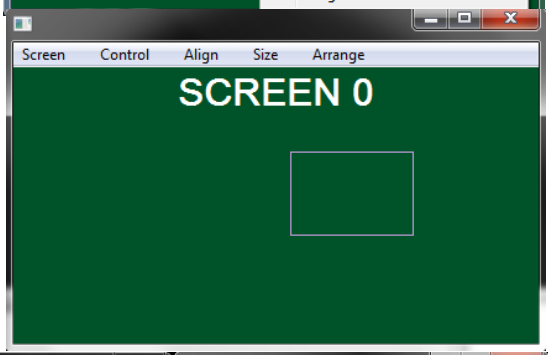


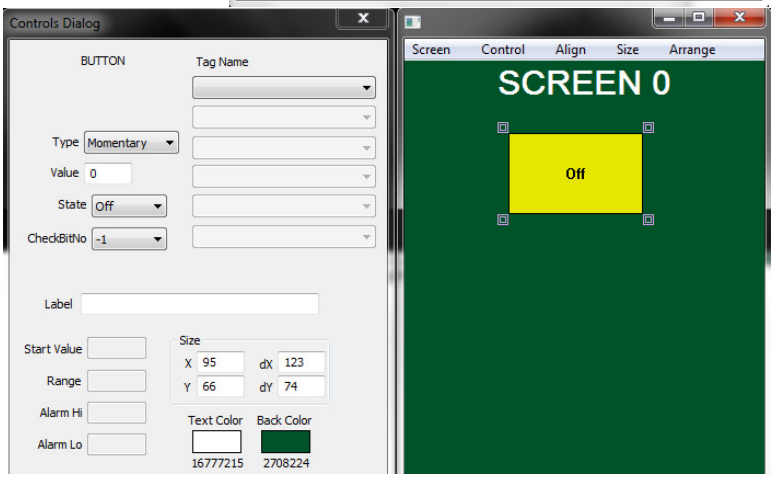
1.5) Controls creation and properties

To add a push button control go to **Control->New->Push Button**. The **New** button on the **Controls Dialog** window will become enabled. With the mouse, left click and select an area on the screen. Press **New** button in the **Controls Dialog**.







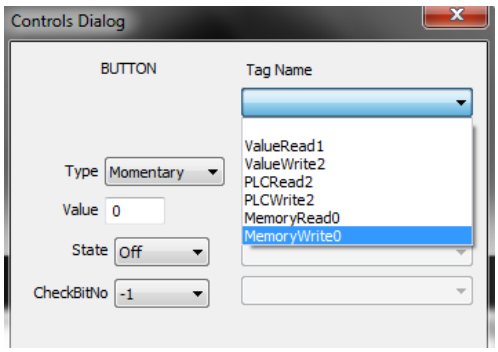


The new push button will be created on the edit screen. Click on it to retrieve its properties. In the right figure the new button is of *Momentary* type, shows the *Off* state, no tag is selected to be associated with it. Also the position (*X, Y*) and the size (*dX, dY*) of the button are shown in the **Controls Dialog**.

You can select the button with the mouse and move it inside the edit window (drag and drop). If you click and drag on any of the 4 corner

rectangles you can change de size of the button. You can change the position and the size of a control from **Controls Dialog** panel. Edit the **Size X, Y, dX, dY** values and then click **Apply**.

1.6) Assignment of tags to controls

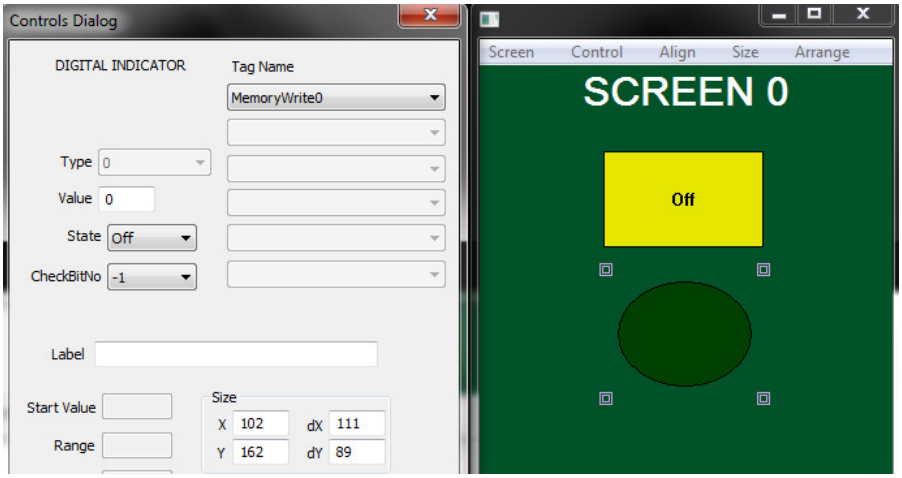


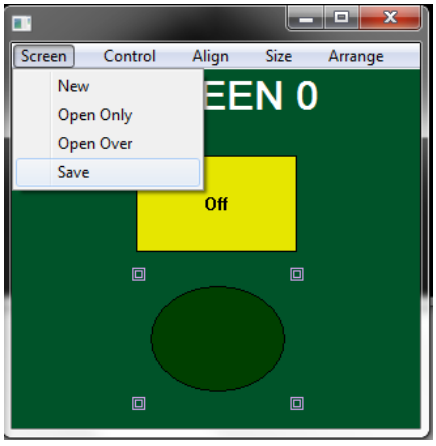
To assign a tag to this button, click on the **Tag Name** combo box in **Controls Dialog** and select *MemoryWrite0*. Then click **Apply**.

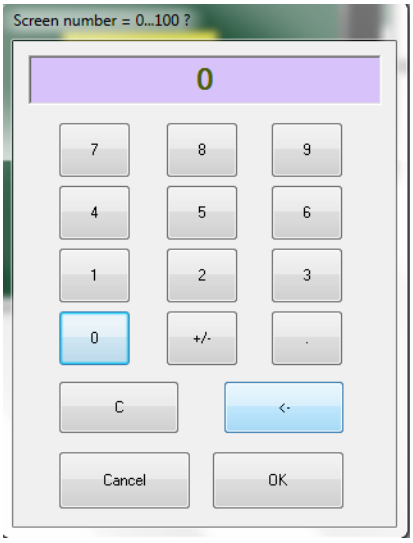
In the **Tag Name** combo box you will find all the application tags, in the order that they were defined and saved. You can change the assignment just by selecting another tag and clicking **Apply**.

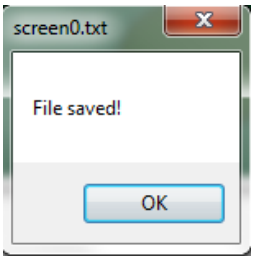
Add a digital indicator on the edit screen. Select from the menu: **Control->New->Digital Indicator**. Select an area on the screen and click **New** in the **Controls Dialog**. Select the new digital indicator to retrieve its properties. From the **Tag Name** combo box select *MemoryWrite0* and press **Apply** button.

Save the screen as screen *0* (zero), which is the first screen that the program will open when started. For this go to **Screen->Save** and type *0* (zero). Click OK. You will receive a confirmation message that *screen0.txt* was saved. Close the edit window and restart *FeSCADA*.





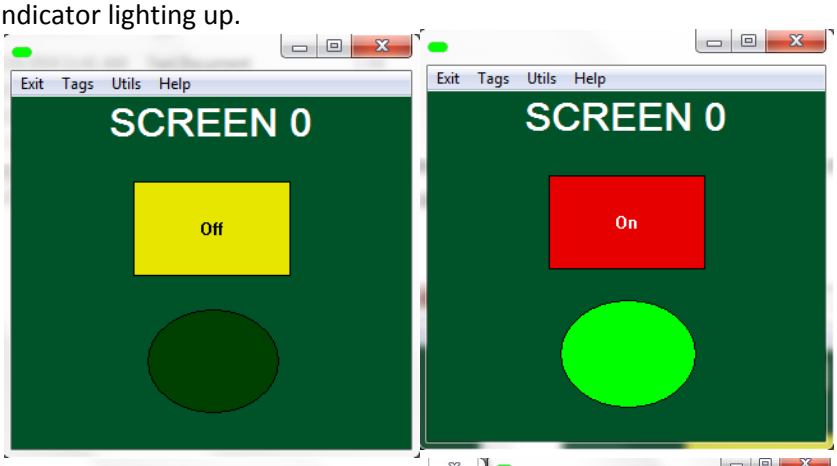


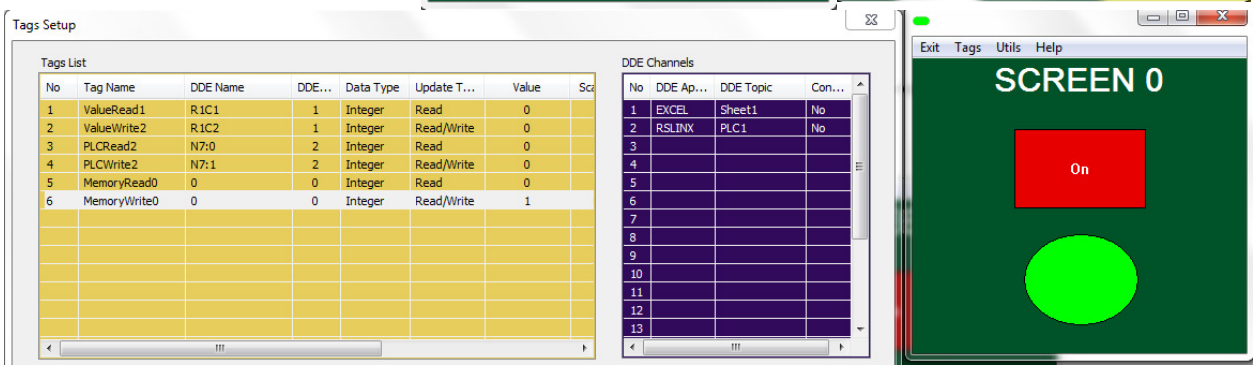


1.7) Application testing

After you restart the program the window will open in the same place where it was saved. If you press the button you will see the digital indicator lighting up.

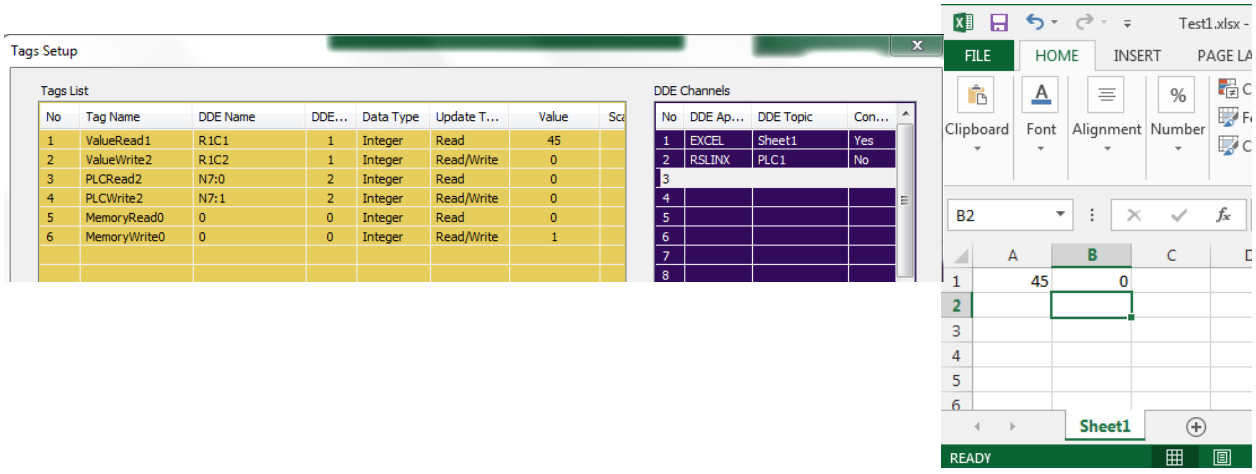
If you open the **Tags** dialog panel you can see for the tag *MemoryWrite0* a value of 0 (zero) when the button is released and a value of 1 (one) when the button is pressed.





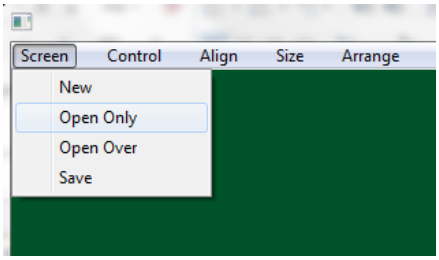
The DDE Channels list is indicating that there is no connection with either channel 1 (*EXCEL*), nor channel 2 (*RSLINX*).

If you open a new EXCEL file and put the number 45 in the first cell (*R1C1*), you will get the result shown in the figures below. In **Tags Setup** dialog, **DDE Channels**, channel 1 is connected (*Yes*), and a value of *45* is displayed in the **Tags List**, for *ValueRead1*.



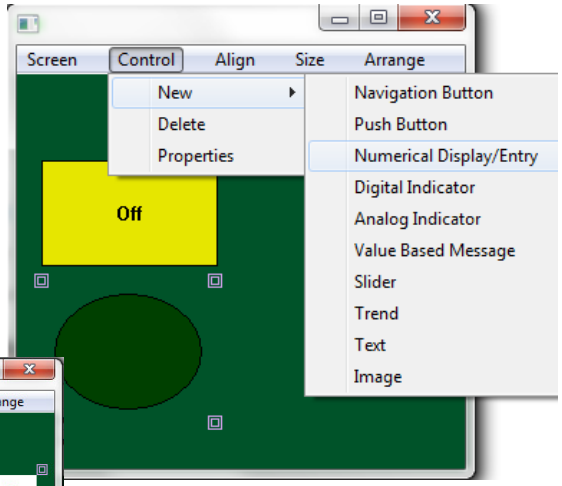
1.8) Application development - Adding numerical displays

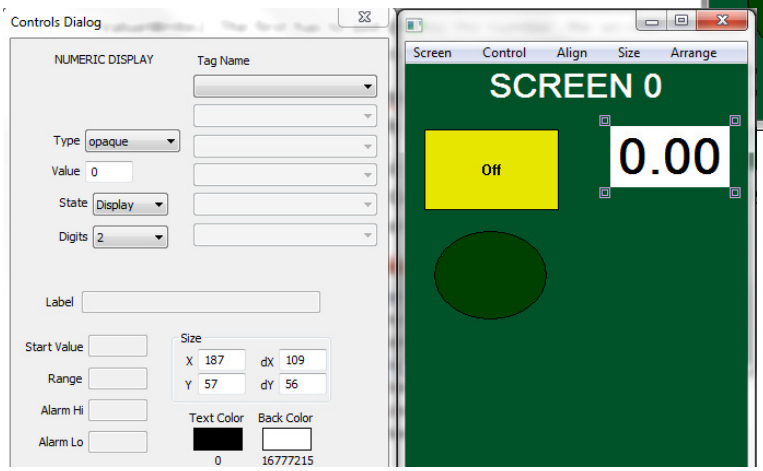
You can add numerical displays for the tags *ValueRead1* and *ValueWrite2*. The first has to just display the number, the second has to be an entry, to change the number.



Open the edit window from main menu **Utils->Screens Edit**. In the edit window select **Screen-> Open Only**. Select 0 (zero) and press OK.

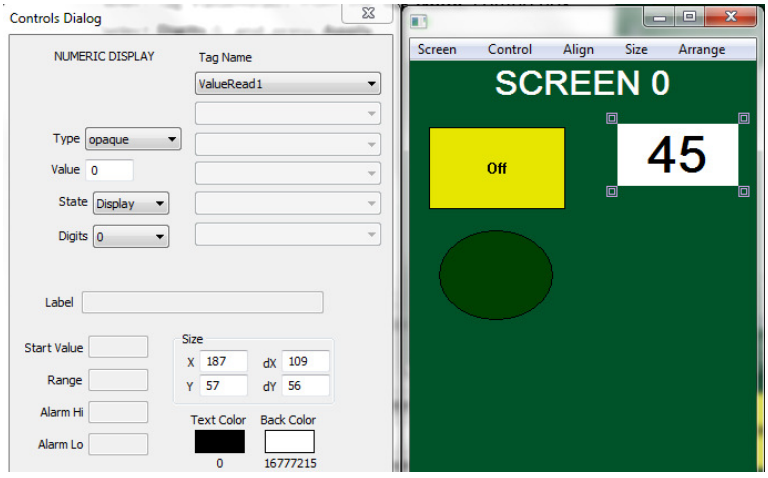
The Edit window will load the controls saved in the *screen0.txt* file. To add a numerical display select **Control->New->Numerical Display/Entry**. Select an area on the screen and press **New**. With the mouse select the new control to retrieve its properties in the **Controls Dialog** panel. This numerical display is *black* text on *white* background, the background is *opaque*, is only a *Display* and it will show the number with 2 digit precision.





To assign a tag to this control, select tag *ValueRead1* from the **Tag Name** combo box. Select **Digits** *0*, and press **Apply**.

To **Copy** and **Paste** a control, select the numerical display with the mouse. The four small rectangles will aprear at the corners of the control. Press *Ctrl+C*. Put the mouse cursor under the first numerical display and press *Ctrl+V*. A new control will be created at the position of the mouse cursor. Select the new numerical display. Change the **Tag Name** to *ValueWrite2*, and **State** from *Display* to *Entry*. Press **Apply**. Save as screen 0 (zero) and close the Edit window. Restart the application.

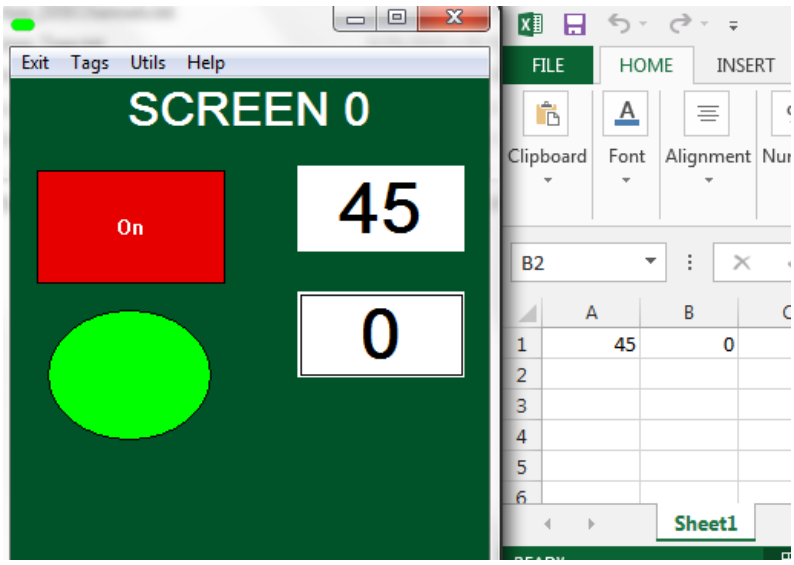


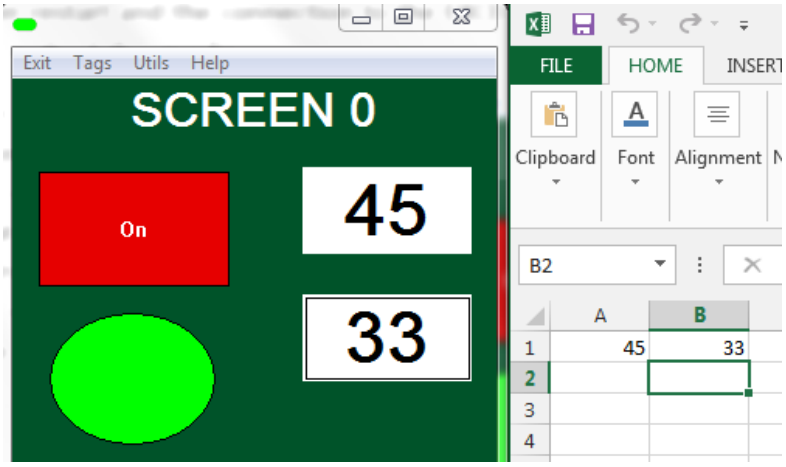
After restart, and reconnection to *EXCEL –Sheet1* communication channel, the screen 0 will look like in the figure from the right.

The numerical display that is an *Entry* has a thin line rectangle of same color as the text color. That means it is a numerical entry. If you double click on it the numerical input screen will pop up. Write *33* and press OK.

You can see the change in EXCEL spread sheet.

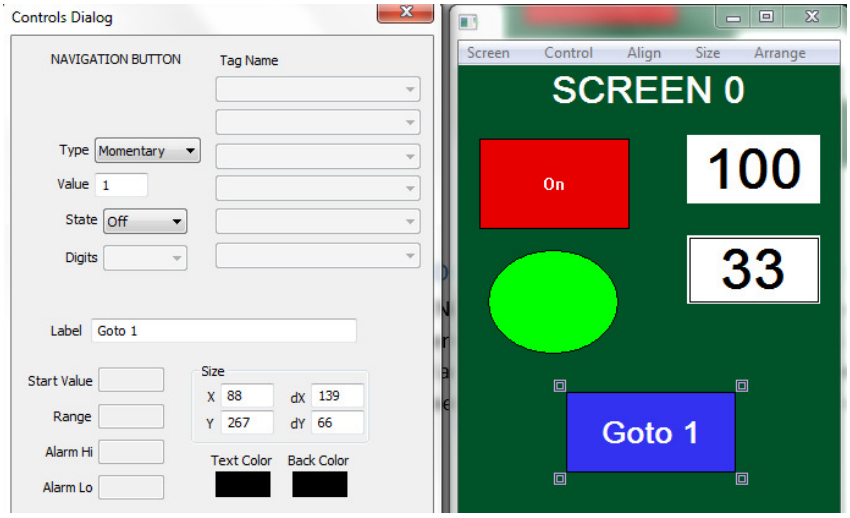
Also, in EXCEL spread sheet you can change the value in *R1C1* from *45* to any other value. The new value will be displayed in the top numerical display from SCREEN 0.





1.9) Application development - Adding a new screen

To add a new screen, SCREEN 1, and a navigation button to open the new window, open the Edit window from the main menu **Utils->Screens Edit**. In the Edit window select **Screen-> Open Only**, select 0 (zero) and press OK. Add a navigation button to screen 0. Select **Control->New->Navigation Button**. Select an area on the screen and press **New**. Select the new navigation button. Type *"Goto 1"* in **Label** text box and type *1* in **Value**. That means the button will open screen 1 when it is



pressed. Click **Apply** for the changes to take effect. Save as screen 0. Do not close the Edit window yet.

In the Edit window select the screen (click on the screen) and change the screen's title and background color. Type *SCREEN 1* as the new title and select a brown color. Click **Apply**. Select the navigation button and edit it with the **Label** *Goto 0* and **Value** *0*. Click **Apply**.

Move the screen to the right of the application window. Save this screen as screen 1 (one). Close the Edit window and restart the application.

Press the navigation button **Goto 1**. The second screen will open in the position where it was saved.

You can redefine the buttons, the digital indicators and numerical displays from the screens 0 and 1 with other tag assignments, positions and sizes. There is no program limitation for the number of tags and controls. The number of screens is limited to 101 (0 + 100 screens).

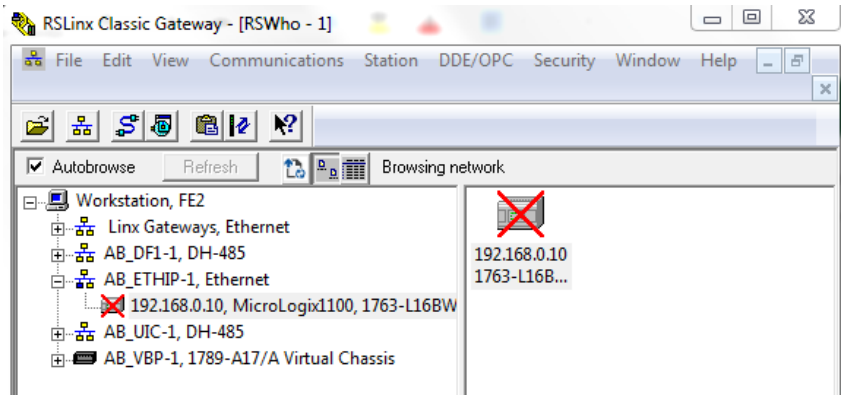
1.10) Conclusions and next steps

This exercise has introduced the first steps required to work with *FeSCADA* software for reading and writing (monitoring and controlling) numerical values from different DDE servers. The example used EXCEL because it is a readily available and recognized software for office work.

In the industrial field, the majority of the vendors for PLCs and other controllers already have a DDE server in a communication software that is used with other SCADA programs.

For example, a lot of VFDs and PLCs are supporting the Modbus TCP or Modbus RTU protocols and there are many software packages on the market that have Modbus TCP/RTU protocols included with a DDE server.

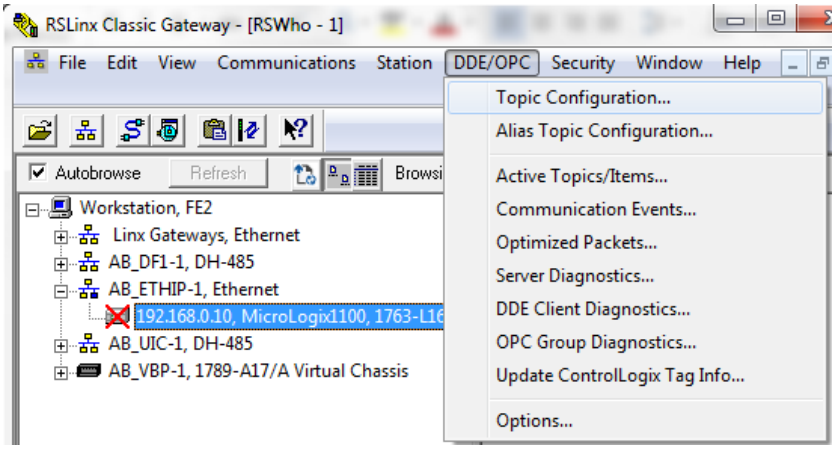
In US, a very significant vendor for industrial automation components is Rockwell Automation (Allen Bradley). They have developed the RSLinx software to communicate with all Allen Bradley PLCs. The RSLinx is a DDE server.



Here is an example on how to setup a DDE communication channel *RSLINX->PLC1* with a MicroLogix1100.

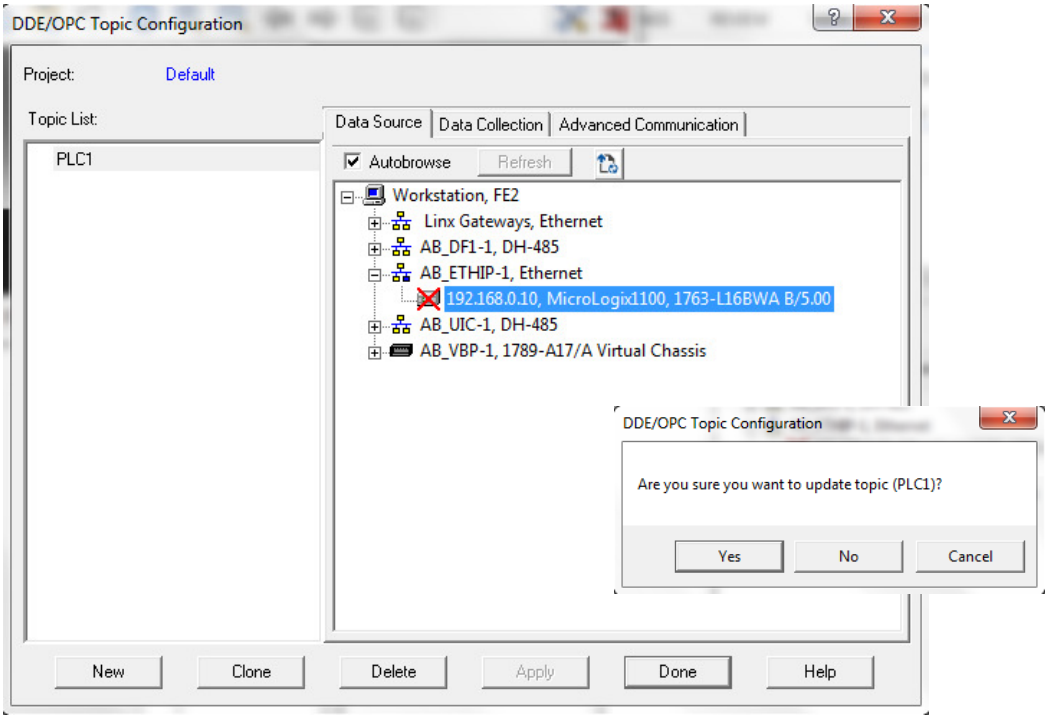
Open RSLinx and establish a connection over Ethernet/IP with a MicroLogix1100. In our case it will be the PLC with the IP address: 192.168.0.10.

From the menu select **DDE/OPC->Topic Configuration**.



In the new window, select your MicloLogix1100 PLC and press **New**. Edit the *NEW_TOPIC* to rename it *PLC1*.

Press Enter and click **Yes** to the question to update the topic *PLC1*. Press **Done** to finish.



2. Tags

Introduction

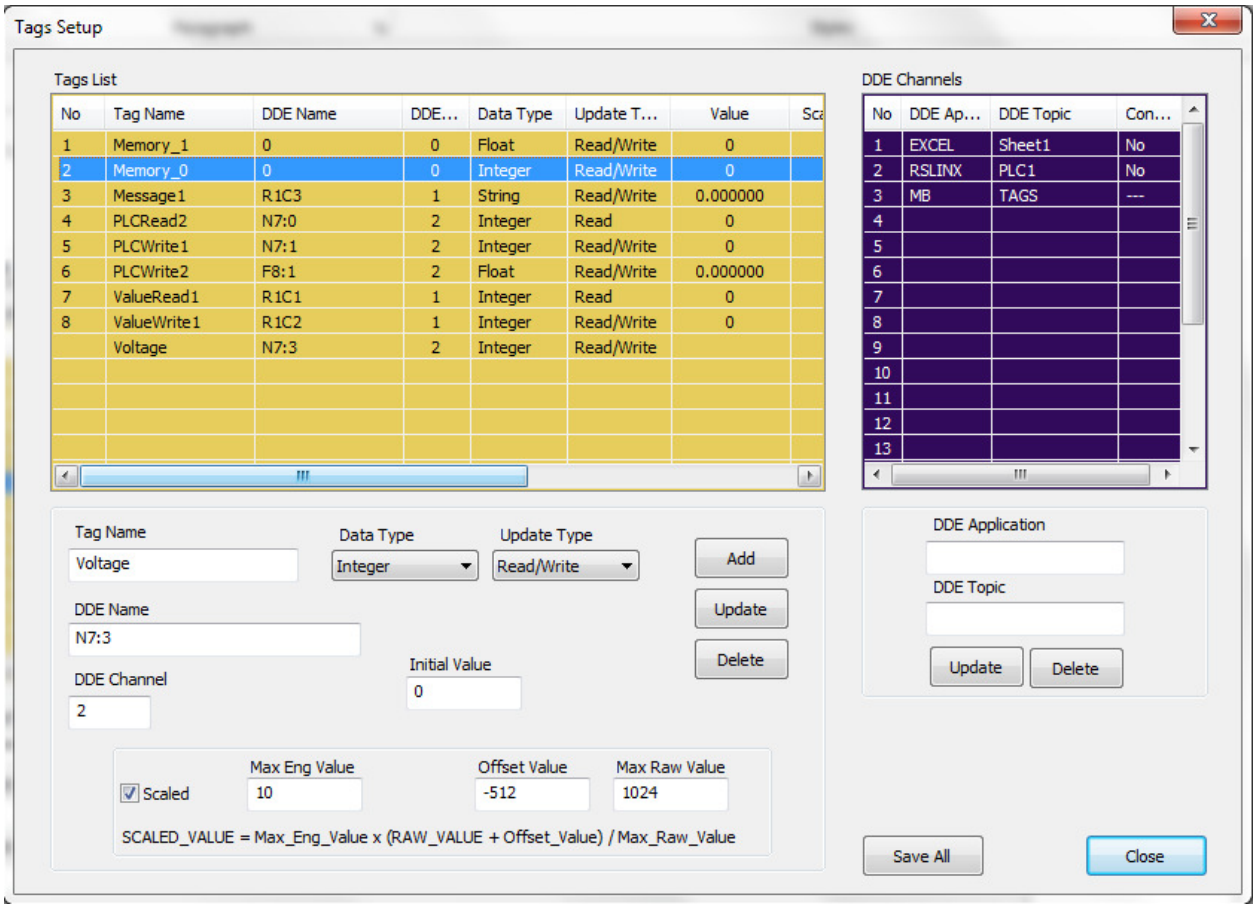
Tags are variables of different data types: integer, real and string. Tags make possible a description of the real world in numbers or text messages. A switch is on (1) or off (0), a temperature is high (110) or low (-17), etc.

This chapter will cover the following topics:

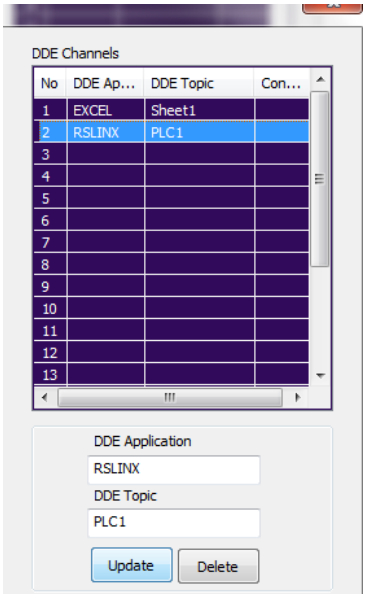
2.1. Setup of data sources
2.2. Tags creation and definition
2.3. Scaling tags
2.4. String tags
2.5. Sorting tags
2.6. Conclusions

2.1) Setup of data sources

The *Tags* menu option will open the **Tags Setup** dialog window (see the picture below). In the right side is the **DDE Channels** list where maximum 20 channels can be defined. In the left side is the **Tags List**.



The **DDE Channels** are the communication gates with other software programs called DDE Servers. These programs usually perform the physical communication with PLCs and other controllers and facilitate the access to the real equipment by allowing data to be shared and/or changed by DDE Clients. In this regard *FeSCADA* is a DDE Client.



To define a DDE channel, for example *EXCEL – Sheet1*, select a row from 1 to 20 in the **DDE Channels** list. In the **DDE Application** text box type the application name, in our example *EXCEL*. In **DDE Topic** text box type the topic name, *Sheet1*. Press **Update** button.

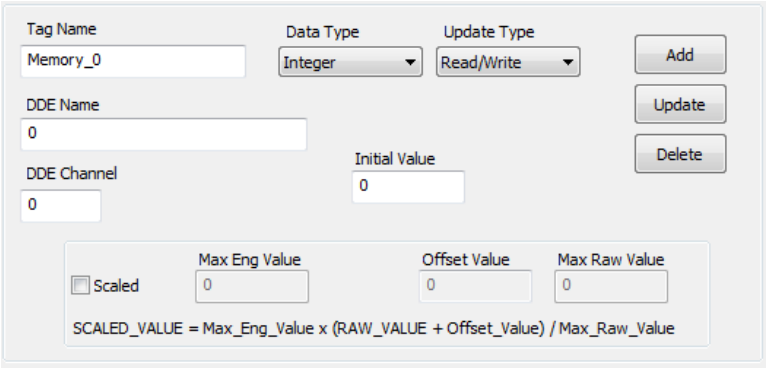
The channel number defined here is linked with the tags definition.

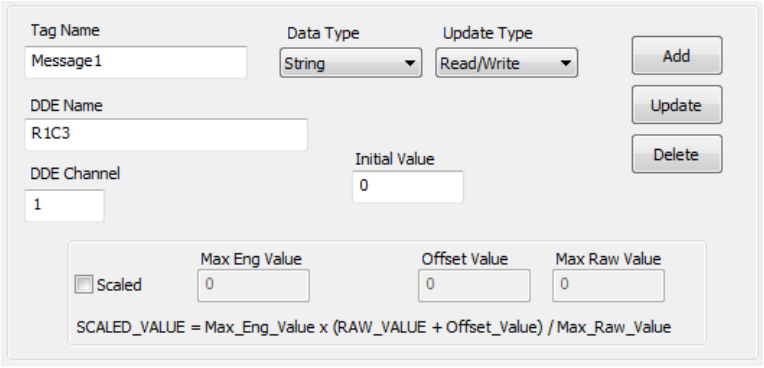
2.2) Tags creation and definition

A tag is a memory allocation for any of the 3 types of data: integer, float, string. The integer and float types are 32 bits long. The string tag is 255 bytes long.

In *FeSCADA* software, every tag has a name used internally. In addition to that, there is a distinction between a *memory* tag and an *external* tag. An *external* tag will have to communicate with a DDE server either to read or to write a value/text. That is why an external tag has a **DDE Channel** number and a DDE item name associated with it. A *memory* tag does not have to communicate with any DDE server. For memory tags the channel number is set to 0 (zero) and the **DDE Name** doesn't matter.

To define a memory tag, type a name in the **Tag Name** text box, type *0* in the **DDE Name**, type *0* in the **DDE Channel**, select *Integer, Float or String* in **Data Type** combo box, select *Read, Write or Read/Write* in **Update Type** combo box and press **Add** push button.





If you make a mistake or you want to change the settings, select the row in the **Tags List**, correct the text, or combo box choices, and press **Update** push button. You can delete a tag by selecting the row and pressing the **Delete** button.

To define a tag linked with a DDE Channel (e.g. *EXCEL-Sheet1*), type the tag name in the **Tag Name** text box,

type the DDE Item name (e.g. *R1C1)* in the **DDE Name**, type the channel number (e.g. *1*) in the **DDE Channel**, select the **Data Type**, **Update Type** and press **Add** push button.

When the program is started an initial value is loaded into every tag. The default value is 0 (zero). To setup a different value modify the **Initial Value** edit box and press **Update**.

2.3) Scaling tags

The tags of type integer and float can be scaled so that they represent engineering values. When a tag is scaled, the value in *FeSCADA* is a floating (real) number, even if it was defined as an *integer* type.

To scale a tag press on the **Scaled** check box. The scaling edit boxes will be enabled. **Max Eng Value** is the maximum value in *FeSCADA* program when the raw value will become **Max Raw Value** and the **Offset Value** is 0 (zero). The **Offset Value** will be applied to the raw value before making any other multiplications.

$$ScaledValue = MaxEngValue \cdot \frac{RawValue + Offset}{MaxRawValue}$$

Example: A tag that is defined as integer, read from a PLC as a byte, will have values between 0 and 255. One can scale this as a final engineering value of 0 to 100 %.

$$ScaledValue = 100 \cdot \frac{RawValue + 0}{255}$$

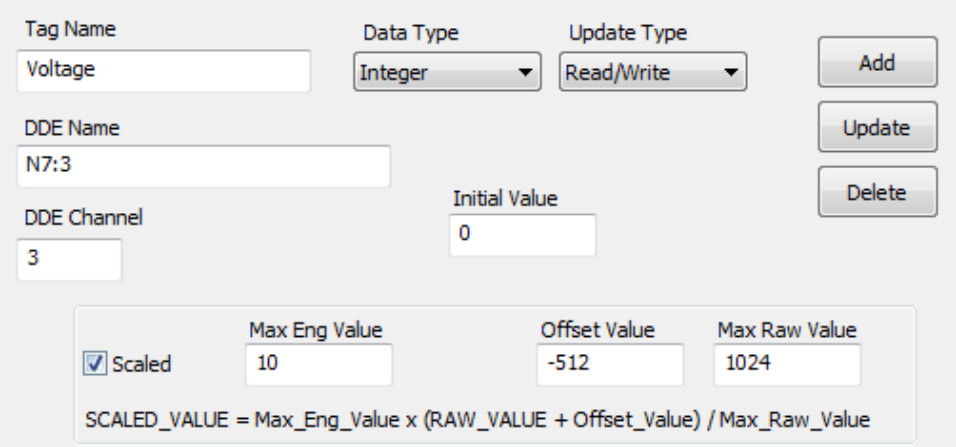
This is called *scale-up*, and this number will be shown for this tag in *FeSCADA* program.

In the case the operator is changing the tag with a numerical display or with a slider, and the tag is of update type *Write* or *Read/Write*, the tag will be sent to the DDE Channel after it will be scaled-down. *Scale-down* is doing the arithmetic operations in reverse.

$$RawValue = ScaledValue \cdot \frac{MaxRawValue}{MaxEngValue} - Offset$$

In our example, if the operator is setting a 50% value to the 1 byte integer tag, the value sent to the DDE Channel is 127.

The example shown in the figure to the right is considering a 10 bit integer from an analog input card of a PLC and scale this value from (0 … 1024) to (-5.0 … 5.0).



2.4) String tags

A string tag is maximum 255 bytes in length. It can be of *Read*, *Write* or *Read/Write* type. It can be displayed with text controls in any screen. The scaling is not relevant for strings.

If the user is defining the special tag names: *SysDate* and *SysTime* of type string, the program will write the date and time in them, 2 times a second, with the format: "MM/DD/YYYY" for the date and "HH:MM:SS" for the time. The user can then put the tags in any screen as a text, with different colors and sizes.

2.5) Sorting tags

You can sort tags in the **Tags List** by clicking on the columns headers. For example: one click on **Tag Name** column header will arrange all the tags alphabetically increasing after tag name. A second click will arange the tags alphabetically decreasing after tag name. The same is true for other columns headers: **DDE Name**, **DDE Channel**, **Data Type**, **Update Type**.

You have to save the tags list if you sort the tags in a desired way and you want to keep the order when the program will restart. By default the tags are sorted in the order of their creation/definition.

You have to restart the program for the **Value** column to become active.

The figures on the right show the tags sorted after **Tag Name**, alphabetically increasing, and after the **DDE Channel** value, increasing.

Tags List

No	Tag Name	DDE Name	DDE...	Data Type	Update T...	Value	Sca
2	Memory_0	0	0	Integer	Read/Write	0.000000	
1	Memory_1	0	0	Float	Read/Write	0.000000	
3	Message1	R1C3	1	String	Read/Write		
10	Message2	0	0	String	Read/Write		
4	PLCRead2	N7:0	2	Integer	Read	0	
5	PLCWrite1	N7:1	2	Integer	Read/Write	0	
6	PLCWrite2	F8:1	2	Float	Read/Write	0.000000	
7	ValueRead1	R1C1	1	Integer	Read	0	
8	ValueWrite1	R1C2	1	Integer	Read/Write	0	
9	Voltage	N7:3	2	Integer	Read/Write	0.000000	

Tags List

No	Tag Name	DDE Name	DDE...	Data Type	Update T...	Value	Sca
2	Memory_0	0	0	Integer	Read/Write	0.000000	
1	Memory_1	0	0	Float	Read/Write	0.000000	
10	Message2	0	0	String	Read/Write		
3	Message1	R1C3	1	String	Read/Write		
7	ValueRead1	R1C1	1	Integer	Read	0	
8	ValueWrite1	R1C2	1	Integer	Read/Write	0	
4	PLCRead2	N7:0	2	Integer	Read	0	
5	PLCWrite1	N7:1	2	Integer	Read/Write	0	
6	PLCWrite2	F8:1	2	Float	Read/Write	0.000000	
9	Voltage	N7:3	2	Integer	Read/Write	0.000000	

2.6) Conclusions

The tags are at the core of any other action that *FeSCADA* software is able to do. Tags are the first to be defined when starting a new project. Tags can be *integer*, *float,* or *string*. The numeric tags can be scaled. The *memory* tags are used locally in the program. The *external* tags link data from *FeSCADA* with other DDE Servers' data.

There is no program limit for the number of tags that can be defined by the user.

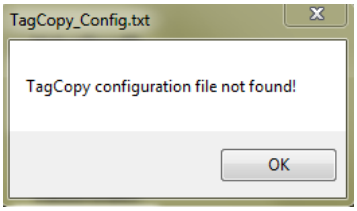
3. Tag Copy

Introduction

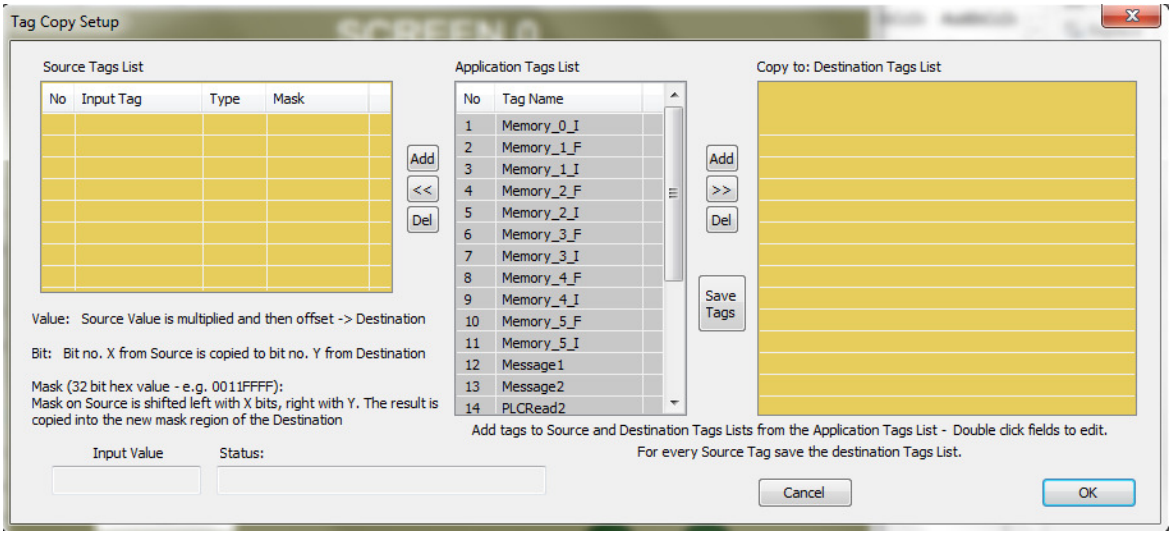
The utility allows the user to copy the value of one tag in other tags. The copy can be done in three ways: by value, by bit or by bit mask. Topics:

3.1) Description – useful situations

There are numerous situations in which the operator needs to send a *Start* signal to more than one machine, in the same time, or to more than one actuator, in the same machine. This can be also true for an analog value, for example the temperature set point for more than one temperature controller. In these cases the **Tag Value Copy** feature is very useful.

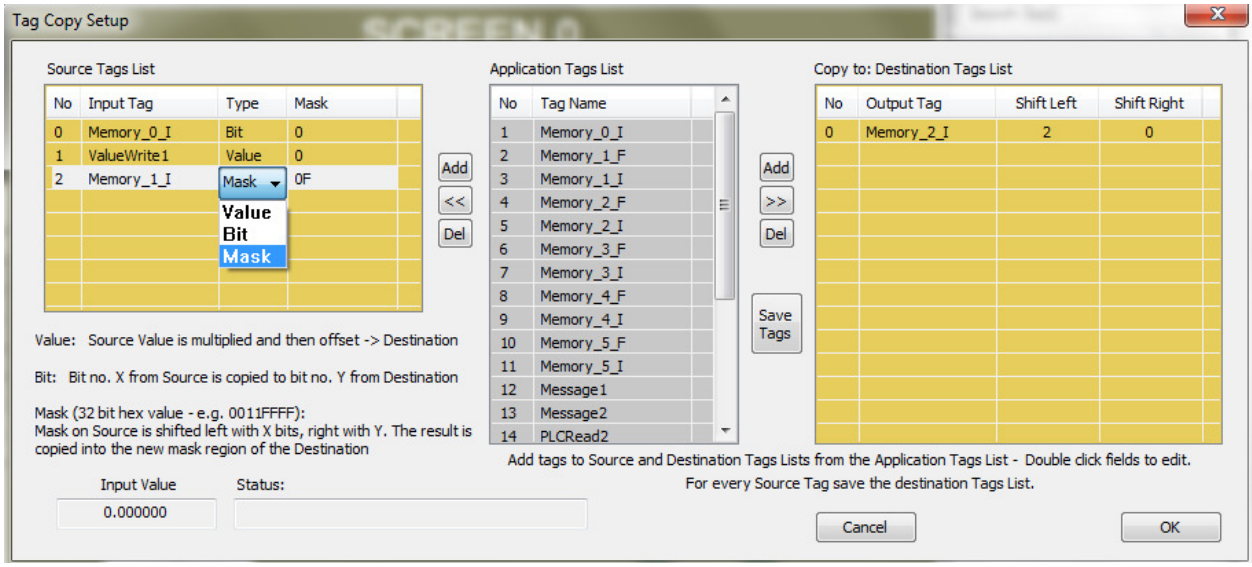


To start the setup dialog window, in the main window go to **Utils->Tag Value Copy**. If the configuration file does not exist an information message will show up. Click OK and the **Tag Copy Setup** window will open. In the left side there is the list of source tags. In the middle there is the list of application tags. The list of destination tags is in the right side.



3.2) Setup of source tags

To setup a source tag, select an application tag with the mouse and press the **Add** button from the left of **Application Tags List**. If you want to change an existing tag from the source tags list with another, select the source tag, select the new application tag and press **<<** button. To delete a source tag, select the tag and press **Del** button. You can insert multiple times the same source tag. Once you added a tag, double click the **Type** cell to select between: *Value*, *Bit* and *Mask* copy types. For *Mask* type you can double click the **Mask** cell and edit the mask value, in hexadecimal numbers.



The source tags list will be saved when you exit the setup dialog by pressing **OK** button.

3.3) Setup of destination tags

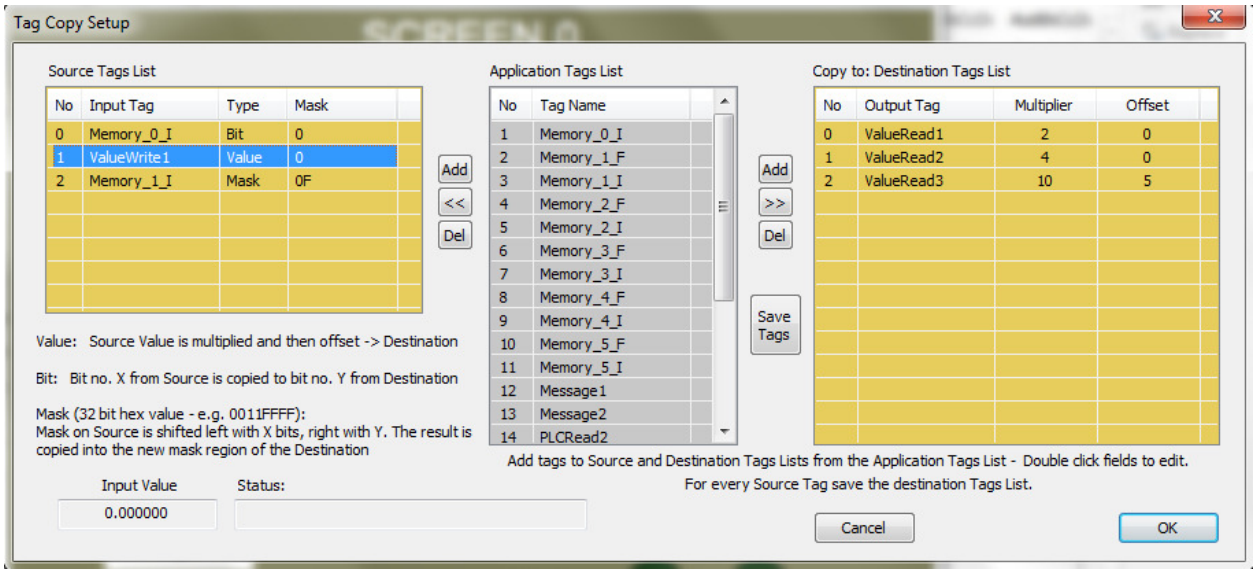
For every source tag you can add multiple destination tags in the list from the right. To do this, select application tags and press the **Add** button from the right side. To change a destination tag with a new one, select the new application tag and press **>>** button. To delete a destination tag, select it and press the **Del** button from the right side.

Observation! Do not click on another source tag while editing the destination tags list for a tag. First press **Save Tags** button. The destination tags will be saved in a file and retrieved when you click again on that source tag.

3.4) Value copy

To copy a value from the source tag to the destination tags, select *Value* in the copy **Type** column of the source tag. The destination list will change the last two column names to **Multiplier** and **Offset**. The value of the source (integer or float) will be multiplied with the value setup in the **Multiplier** column and, after that, it will be added with the value from the **Offset** column. You can edit these values by double clicking inside the cells and typing from the keyboard. Press Enter when finished.

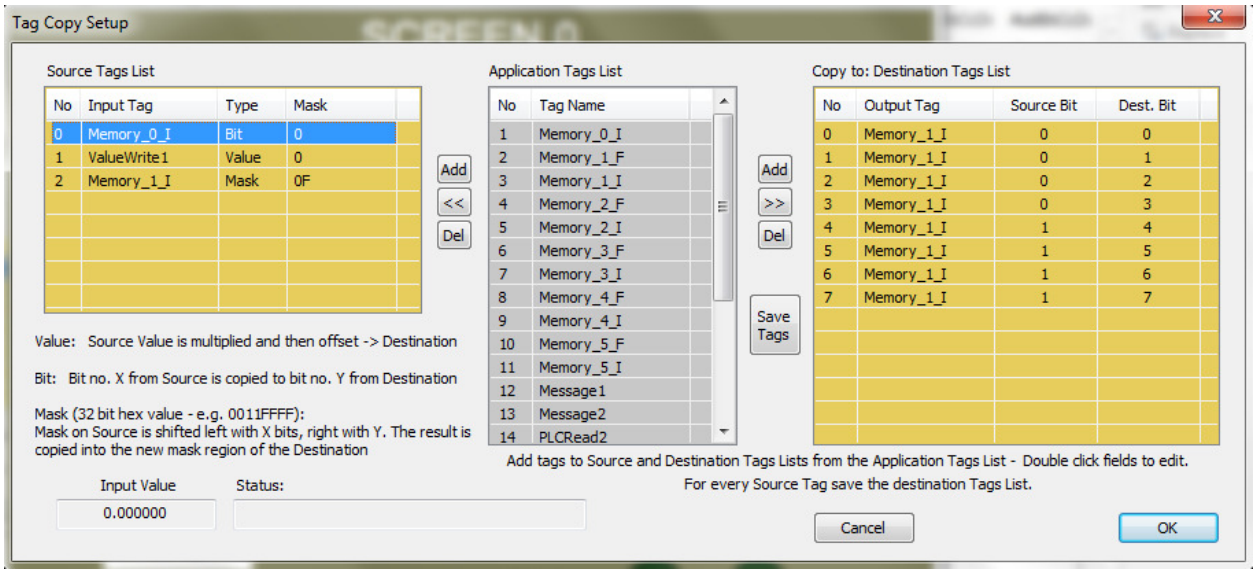
If the destination tag is a string type, the resulting value will be converted to a string. The only difference is that the offset number is used just to indicate with how many digits you want the string to be printed.



3.5) Bit copy

To copy the value of one bit from the source tag to another bit from destination tag select the *Bit* copy type. The last two columns from the destination list will change their names to **Source Bit** and **Dest. Bit**. That means that the source bit number from the source tag will be copied in the destination bit number of the destination tag. Double click the cells to edit the values. Press Enter to keep the changes.

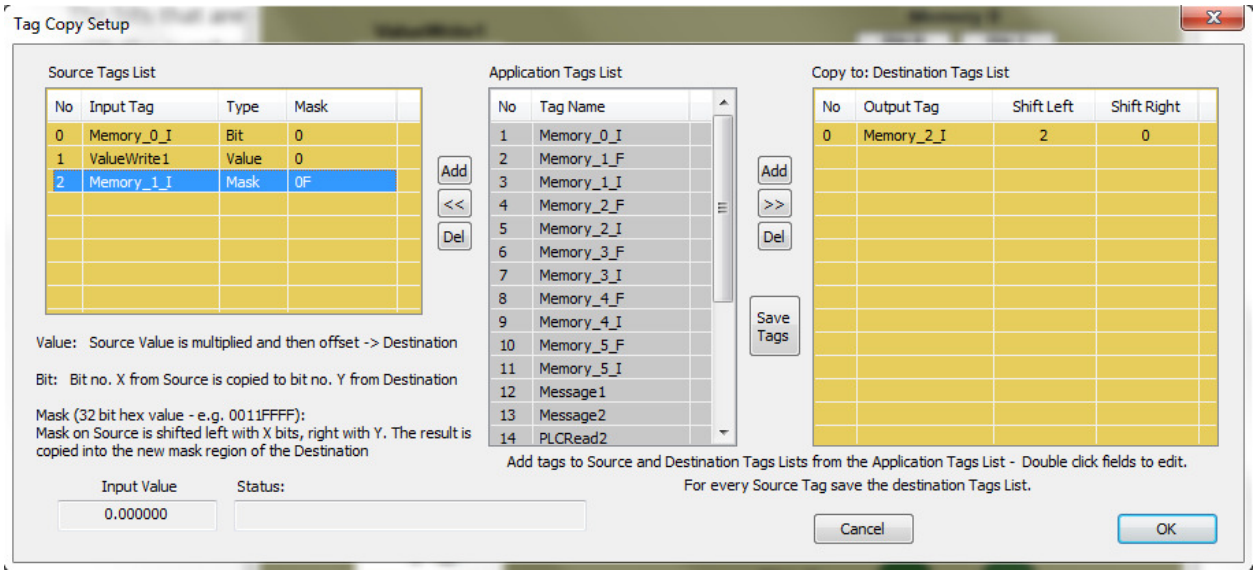
You can put multiple tags in the destination list. You can put multiple times the same destination tag, if you want to copy one or more bits from the source tag in multiple locations of the same destination tag.



3.6) Copy with bit mask

To copy with bit mask, select the *Mask* copy type for the source tag and edit the mask format in hexadecimal numbers. The last 2 columns from the destination list will change their names to **Shift Left** and **Shift Right**.

Only the bits that are not zero in the mask (that can be 0 or 1 in the source tag) will be copied and shifted left with the number from **Shift Left** column. After that they will be shifted right with the number from **Shift Right** column and will be inserted in the destination tag, at the new position.



3.7) Conclusions

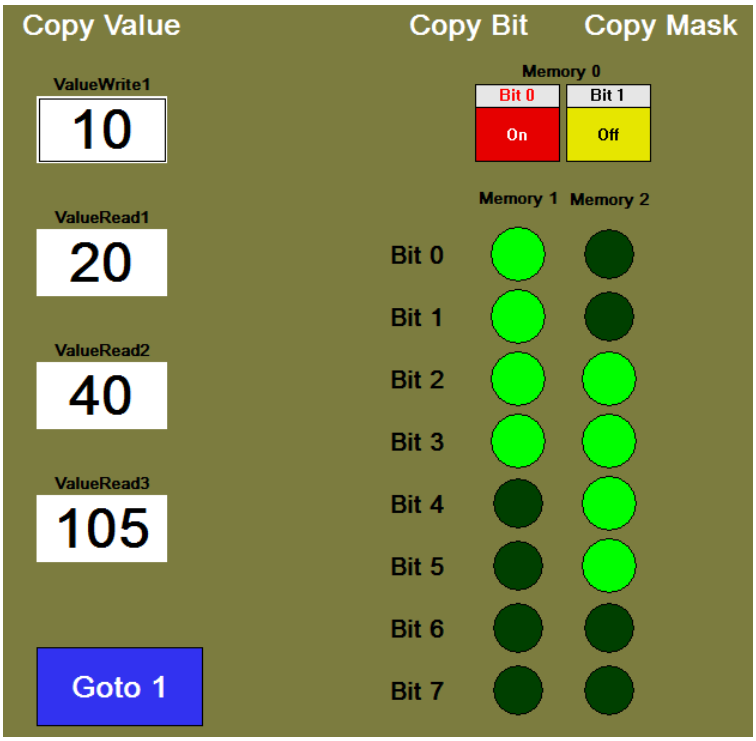
The utility can copy a value or bits from a source tag to one or many destination tags. It is useful when you want to send a command, or a value, to different machines, or to different actuators from one machine.

The settings presented in the pictures above can be visualized in the picture on the right.

For value copy, the input value in the integer *ValueWrite1* was set to 10.

Integer *Memory_0* bit 0 is copied into *Memory_1* bits 0, 1, 2 and 3.

Memory_1 integer is copied with mask 0x0F and shifted 2 bits left, into *Memory_2* integer.



4. Timers

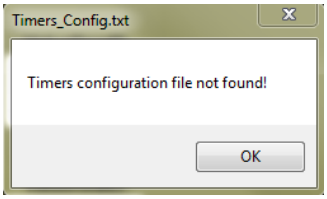
Introduction

The utility allows the user to setup different timers with different configuration options. Timers can be used to trigger events, modify tags values, record data or send emails. Topics:

4.1. Description – useful situations
4.2. Types of timers: non-cyclic, cyclic, time/date
4.3. Setup of a timer
4.4. Application examples
4.5. Conclusions

4.1) Description – useful situations

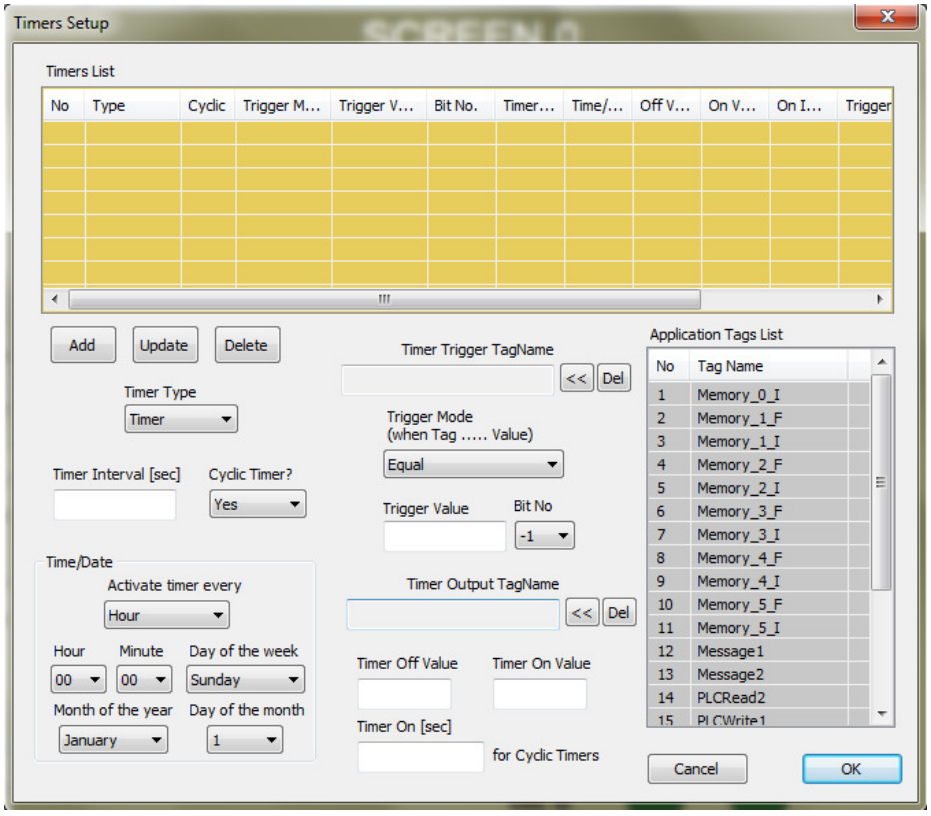
Timers are very useful in automation. Every PLC, doesn't matter how simple it is, has timers and counters integrated from start in its basic programming language. A timer is started/activated by a trigger. After the programmed time has passed an action is occuring. If the trigger is deactivated before the time has passed, no action will occur. The trigger can be the value of a tag or a certain time/date checked agains the computer's real time clock. A special type of timer is a cyclic timer that will restart timing as soon as the programmed duration has expired.



To start the setup dialog window, in the main window go to **Utils->Timers**. If the configuration file does not exist an information message will show up. Click OK and the **Timers Setup** window will open.

On the top of the window there is the list of timers. In the right bottom corner there is the list of application tags.

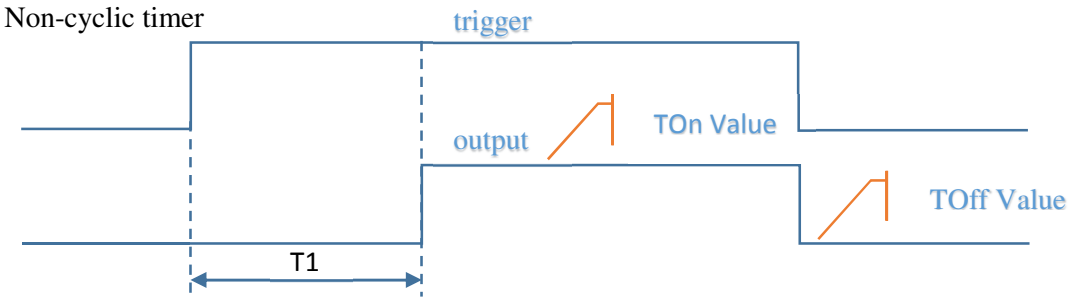
In the center and left side there are the settings that can be chosen to define a timer.



4.2) Types of timers: non-cyclic, cyclic, time/date

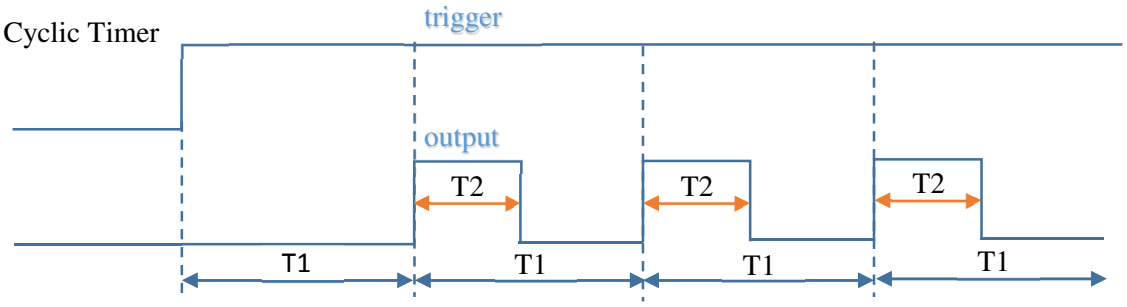
A timer has an input trigger condition and an output value. The input condition can be the value of a tag in relation with a constant value (<, >, =, <>) or the time/date of the computer's real time clock (e.g. every Monday at 8:00 AM). The output is a tag to which the timer can change its value. A timer is called *Off* when the time did not expire yet and is called *On* when the time has expired.

A non-cyclic timer is waiting for the trigger condition to become true and starts timing. After the time has passed, it is changing the value of the timer's output to a TOn value. When the trigger condition becomes false, the output value is changed to a TOff value. To restart the timer, the trigger condition has to become true again.

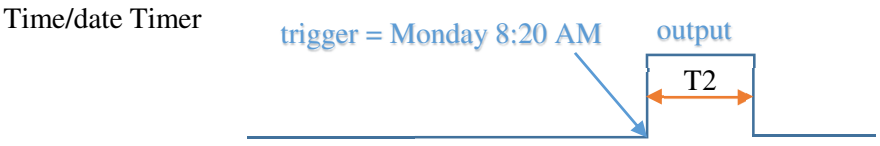


`

A cyclic timer is a special case where as long as the trigger condition is true, it will restart timing as soon as it ends (becomes On). In the On state it will start a second timer, at the end of which it will turn itself Off. In these conditions, for the cyclic timer it is relevant to talk about 2 simple timers with 2 output values (Ton and TOff).



A time/date timer is a combination between checking the real time clock of the computer together with an On timer. The timer will keep checking the time/date set by the user against the computer's time/date. When they are identical (to the minute) the timer will turn On, with the output having the value set to TOn value. Once in the On state, the timer is starting a duration timer set by the user. When the time expires the output is set to the TOff value.



4.3) Setup of a timer

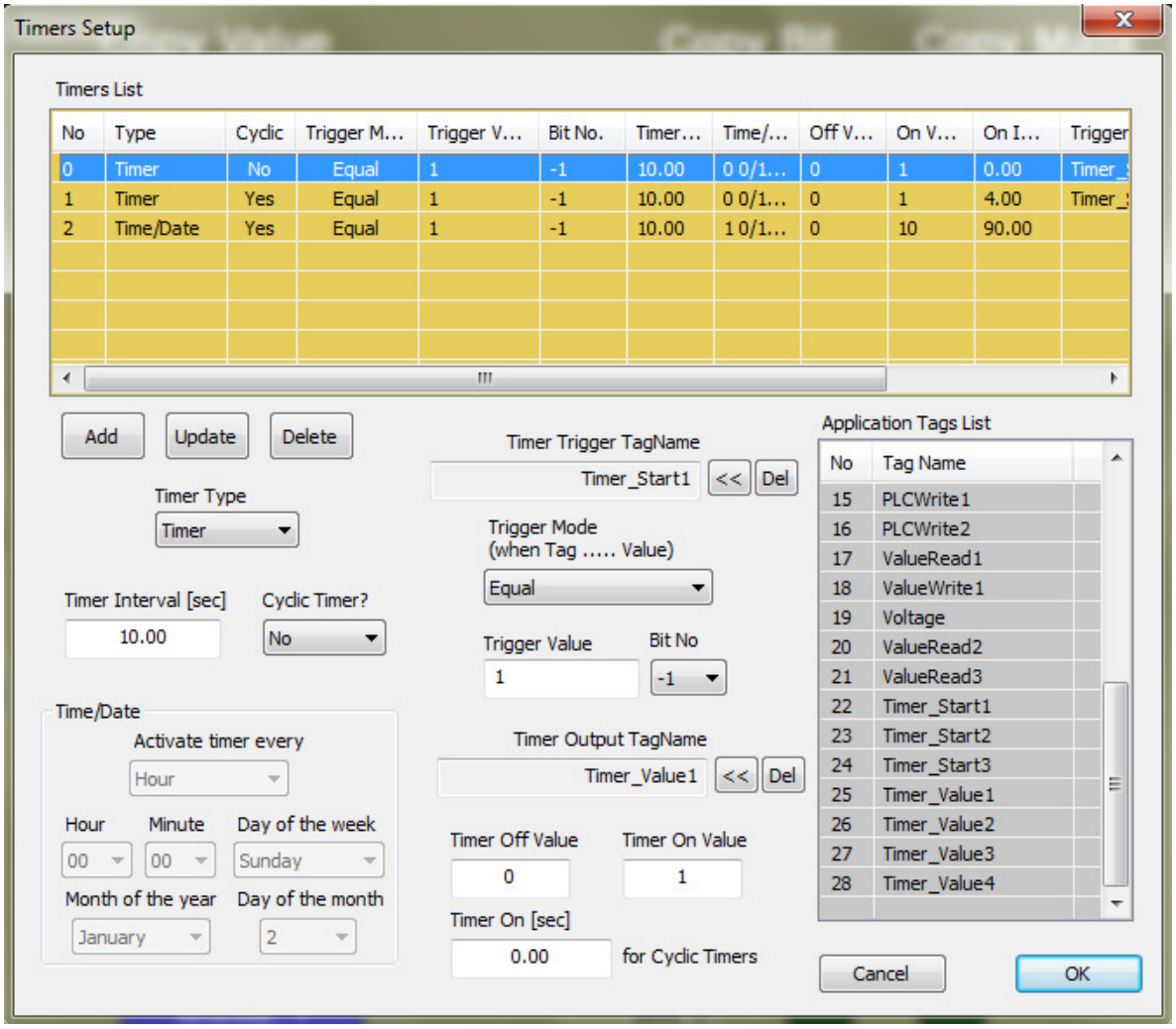
To setup a non-cyclic timer, select **Timer Type** as *Timer*, select *No* for **Cyclic Timer?**, and set a value in **Timer Interval**. The value is in seconds and it can be less than one (e.g. 0.25).

To set the trigger tag name select a tag from the tags application list and press **<<** button on the right of the **Timer Trigger TagName** text box. Use **Del** button the delete the selection. To change the tag name select another tag from the application tag list and press **<<** button.

To set the trigger condition select an option from the **Trigger Mode** combo. The options are: *Equal*, *Different*, *Less*, or *Greater*. Type the reference (constant) value in **Trigger Value** text box.

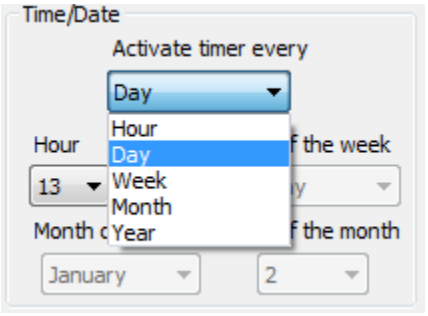
If the trigger tag name is of type integer and you want to check it at the bit level, select the bit number (0…31) in **Bit No** combo box. In this case the **Trigger Value** edit box will be forced to 1 and disabled. To compare the whole number as integer or float, select -1 in **Bit No** combo.

In the final, type the values for the output tag, when the timer is switching to TOff and TOn states, in **Timer On Value** and **Timer Off Value** text boxes. The **Timer On [sec]** is used only for cyclic timers.



Press **Add** button to add a new timer, or press **Update** to modify an existing one, after you have selected it first. Press **OK** to save in file and exit, or **Cancel** to exit without saving.

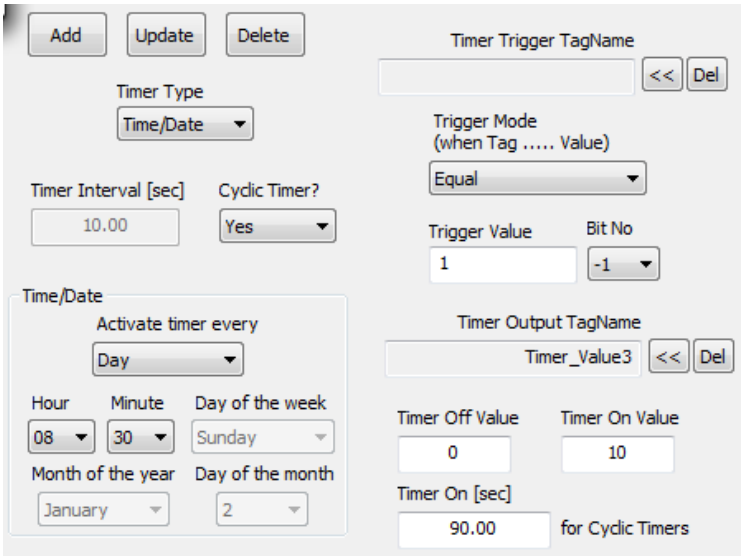
To add a cyclic timer, the settings are identical with two exceptions. First is that the selection **Cyclic Timer?** is *Yes* and the second is that a value is required in **Timer On [sec]** text box.



For the time/date timer, in the **Timer Type** combo select *Time/Date*. This will enable the Time/Date sub-panel in the bottom-left corner of the window. In the **Activate timer every** combo box there are the following choices: *Hour*, *Day*, *Week*, *Month*, and *Year*. As you proceed from *Hour* to *Year*, more combo selections are switching from being disabled to being enabled.

Select a **Timer Output TagName** from the application tags list. Type values for **Timer Off Value** and **Timer On Value**. And last, type a value in seconds in **Timer On [sec]** edit box, for how long the timer will stay On.

In the example below, the time/date timer will turn on every day at 8:30 AM. The output *Timer_Value3* will have the value 10. After 90 seconds the output *Timer_Value3* will change the value to 0 (zero).



If the output tag is of type string then string values can be setup in the **Timer Off Value** and **Timer On Value** text boxes.

4.4) Application examples

For non-cyclic timers the applications can be simple, with one timer, or more complicated, using more timers in a cascade combination. The simple applications refer to inputs and outputs. Example: when an input turns on, wait 20 second before turning on an output.

The cascade applications use the output from the previous timer to enable the next timer. This can be done with a single memory tag.
For example:
Timer T1 is set with 5 seconds, trigger tag = *Tag1*, trigger condition = *equal with 0*, tag output = *Tag1* and TOn = TOff = 1 (one).
Timer T2 is set with 10 seconds, trigger tag = *Tag1*, trigger condition = *equal with* 1, tag output = *Tag1* and TOn = TOff = 2 (two).
Timer T3 is set with 20 seconds, trigger tag = *Tag1*, trigger condition = *equal with* 2, tag output = *Tag1* and TOn = TOff = 0 (zero).
Tag1 starts with a value of 0 (zero) that will be the trigger for the timer T1. After 5 seconds *Tag1* will be 1 (one) and will be trigger for the timer T2. After another 10 seconds *Tag1* will be 2 (two), trigger for T3. After another 20 seconds *Tag1* will be 0 (zero) again.

Three time/date timers, T1, T2, and T3 can be set to switch On for 600 second (10 min) a tag *Tag2*, every Monday (T1), Wednesday (T2) and Friday (T3) at 6:30AM. The *Tag2* value can start a cyclic timer T4 which will turn some sprinklers on and off in a cycle of 30/30 seconds.

4.5) Conclusions

Timers are useful tools in many real life automation situations. Three type of timers can be used in *FeSCADA*: non-cyclic, cyclic and time/date. The trigger conditions are based on either arithmetic relations of a trigger tag with a constant number, defined by the user, or by checking a bit number for 0 or 1, in an integer tag. The output of the timers is an application tag selected by the user. Any value can be defined to be loaded in the output tag for the On and Off states. The values TOn and TOff are loaded in the output tag only in the moments of transition: Off -> On and On -> Off.

5. Recipes

Introduction

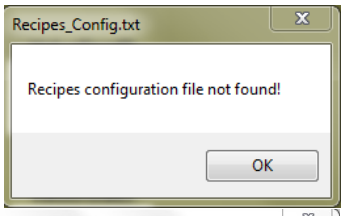
In the automation world the recipes are useful tools designed to memorize different machine settings. The change from one product to another is done just by pressing a button. Recipes are little databases in which the user can keep two, three sets of data settings, or hundreds. Topics:

5.1. Description – useful situations
5.2. Structure of a recipe: size, index, command and status
5.3. Setup of a recipe
5.4. Application examples
5.5. Conclusions

5.1) Description – useful situations

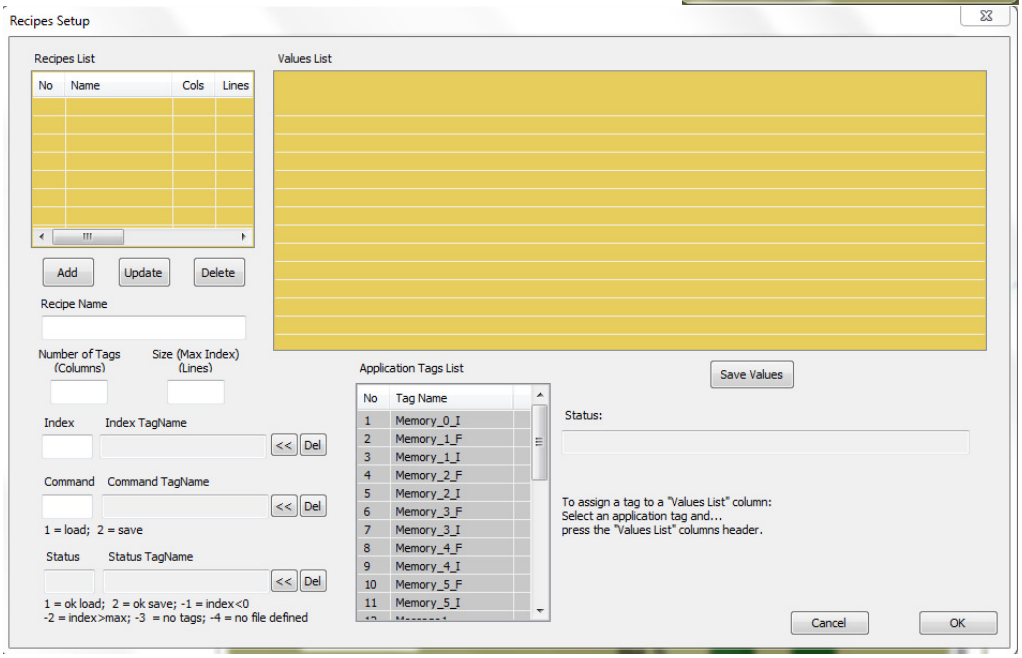
A production line in modern days can make different variations of the same kind of product. A dough mixer can make dough for different types of bread. A brewery can make different types of beer. The repetability and consistency of the properties and qualities of the product are important. Once the good settings for a product variation are found, they are saved and kept in a database.

The second purpose for using recipes is to be able to change the production settings as fast as possible, without involving too much the machine operator in the process.



To start the recipes setup window select **Utils->Recipes**. If the configuration file does not exist an information message will show up. Click OK and the **Recipes Setup** window will open (see figure below).

The **Recipes List** is in the left of the window. On the right there is the list with the data for each recipe, **Values List**. The list of **Application Tags** is at the bottom. Under the **Recipes List** there are the settings that can be chosen to define a recipe.



5.2) Structure of a recipe: size, index, command and status

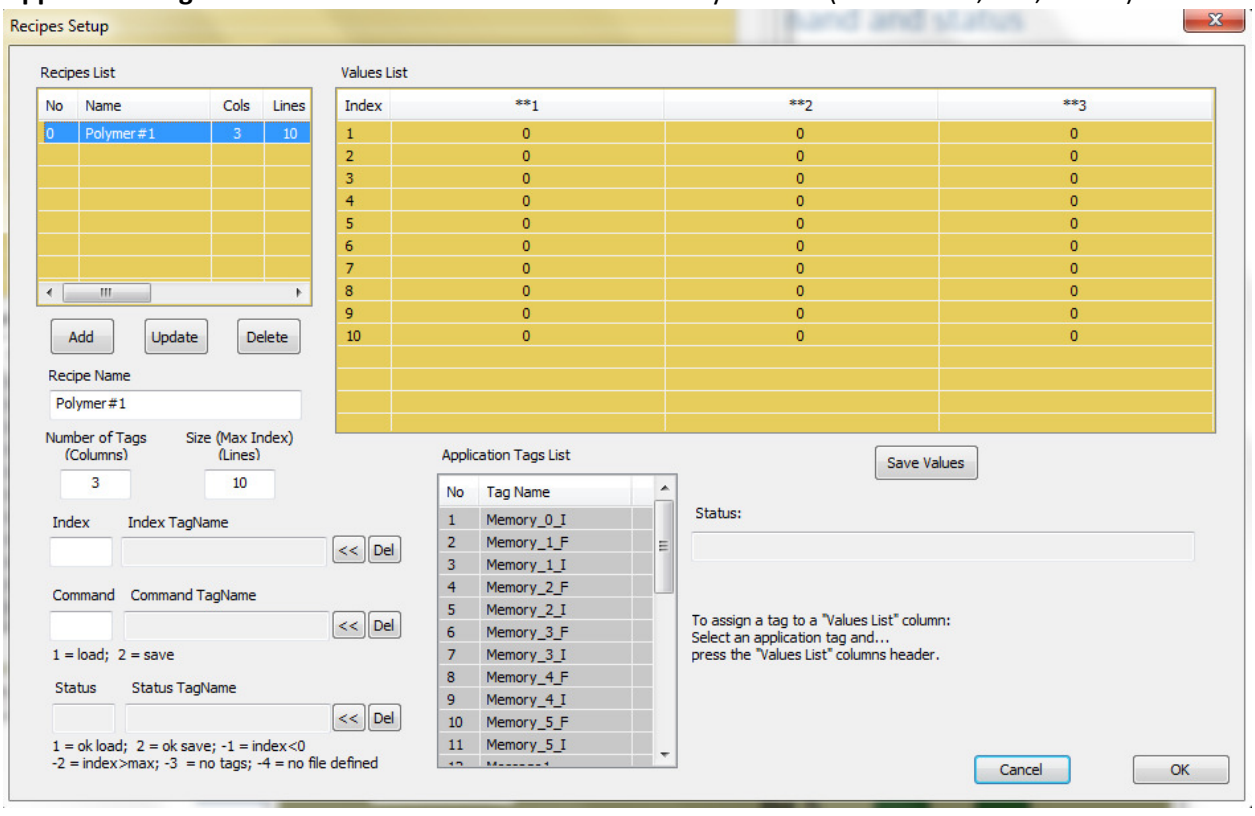
When a recipe is defined the first question is: how many parameters the recipe should contain? This is the size of the dataset. For example, we need 3 parameters to cure a kind of acrylic polymer in an oven: a pressure, a temperature and a time. The next question would be: how many recipes will be needed? This is more related with the storage in files, because a recipe can be resized anytime. These 2 numbers will define the *Size* of the recipe: *Number of Columns* and *Number of Lines*.

Once the size is defined the recipe is controlled with three parameters:

- *Index* – integer tag, the number of the dataset which has to be loaded as active
- *Command* – integer tag that will trigger the loading procedure
- *Status* – an integer tag that will acknowledge the command was executed

5.3) Setup of a recipe

To setup a recipe type a name in the **Recipe Name** text box, type the size of the recipe in **Number of Tags (Columns)** and **Size (Max Index) (Lines)** edit boxes and press **Add** button. To change a recipe select the recipe in **Recipes List**, edit the changes and press **Update** button. To delete a recipe press **Delete**. In the figure bellow a recipe called *Polymer#1* was defined with 3 columns and 10 lines. The **Values List** has 3 columns with the default names **1, **2, **3. To assign a tag to a column, select the tag from the **Application Tags** list and click on the header of the column you want (click on **1, **2, or **3).



Double click fields to edit values. After editing press Enter. Press **Save Values** button to save the last changes in files. If you select another recipe from the **Recipe List** without saving, the changes are lost. Press **OK** to save the recipes settings and close the **Recipe Setup** dialog window.

To assign the **Index TagName**, select the tag name from **Application Tags List** and press **<<** button. Press **Del** to delete. The same for **Command TagName** and **Status TagName**.

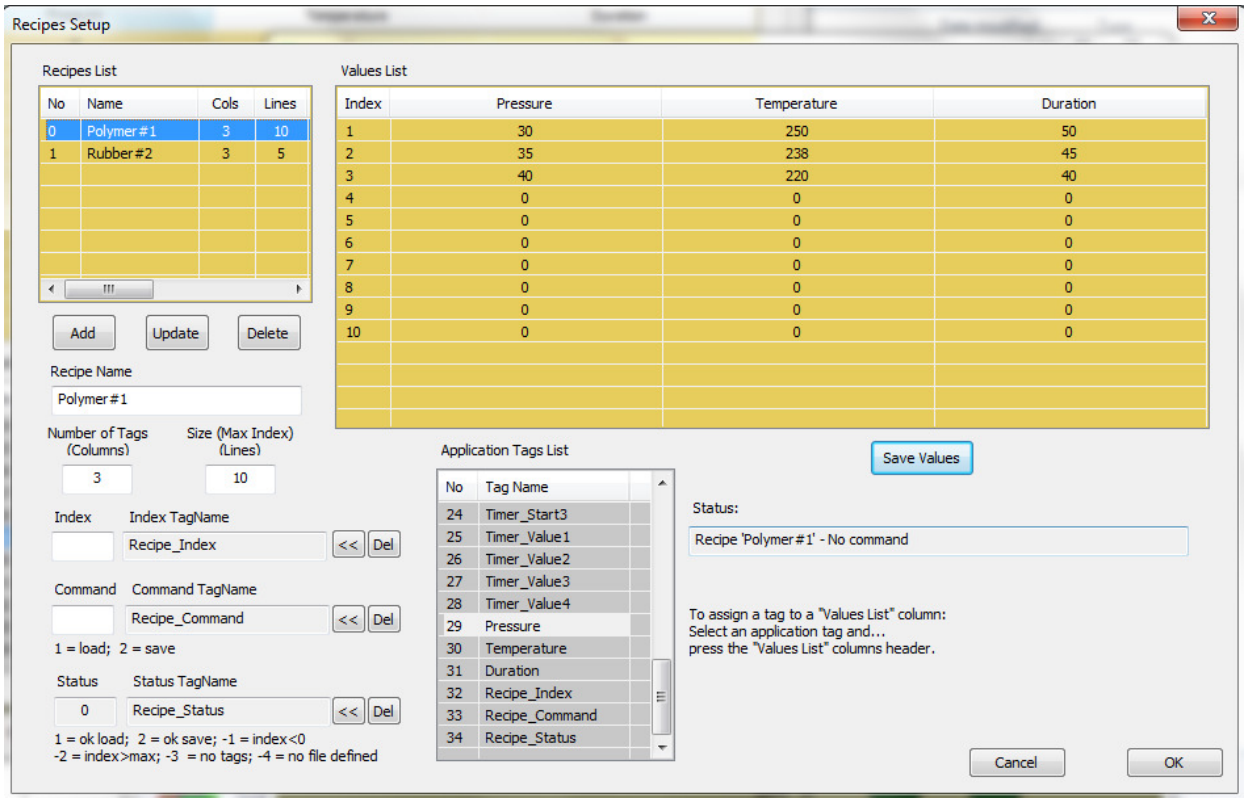
When the *Command* tag is equal with 1, if the *Index* is valid (> 0 and < max), the set of values are transferred to the assigned tags. The *Status* tag will become 1 to acknowledge the recipe was loaded. The *Command* tag has to be zero before another recipe *Index* can be loaded. This means that if the *Index* is changing while the *Command* tag is still 1, nothing will happen.

Command	Description
0	No command
1	Load recipe
2	Save recipe (in the memory)

Status	Description
0	No command, no errors
1	Recipe loaded
2	Recipe saved
-1	Index < 0
-2	Index > max
-3	No tags assigned in the recipe
-4	No data file found for the recipe

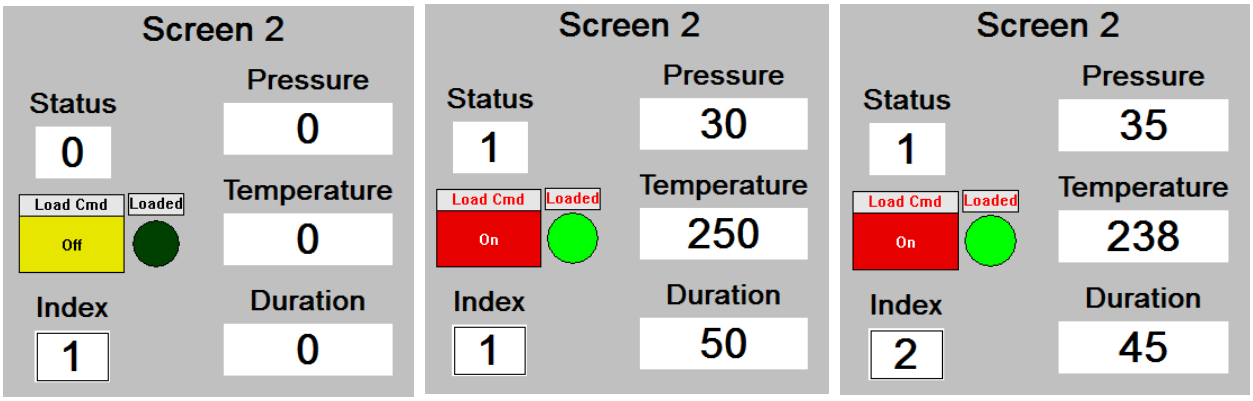
You can save a set of data from the tags to the recipe, but only in the memory of the computer, while the program is running. At the next start, the program will read the recipe data from the file.

In the figure below the tags *Pressure*, *Temperature* and *Duration* were assigned to the column headers. The tags *Recipe_Index*, *Recipe_Command* and *Recipe_Status* were assigned for recipe control.



5.4) Application examples

The simplest application example is to create a window with a button (command), an indicator (status) and a numeric display/entry for the index. The user can type an index number and press the button to load the recipe index into the assigned application tags. The figures bellow are showing the values for the tags *Pressure*, *Temperature* and *Duration* in the cases the index values 1 and 2 were selected.



A more complicated example can be realized in a combination of *FeSCADA* software with a PLC program. The PLC program will select the index and load the recipe based on the PLC logic.

5.5) Conclusions

Recipes are useful in automation. A recipe is a set of data in a matrix like table, with columns and lines. The data from any line can be loaded in selected application tags with the aid of an index tag and a command tag. After a recipe is loaded, an acknowledge value is displayed in a status tag.

6. Ramps

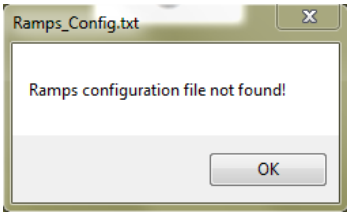
Introduction

The ramps are special tools for changing a value, usually a set-point value of a controller, in a continuous mode, up and down, in precise synchronization with a timer. They are called also: ramp and soak programs, time cams, or time functions. Topics:

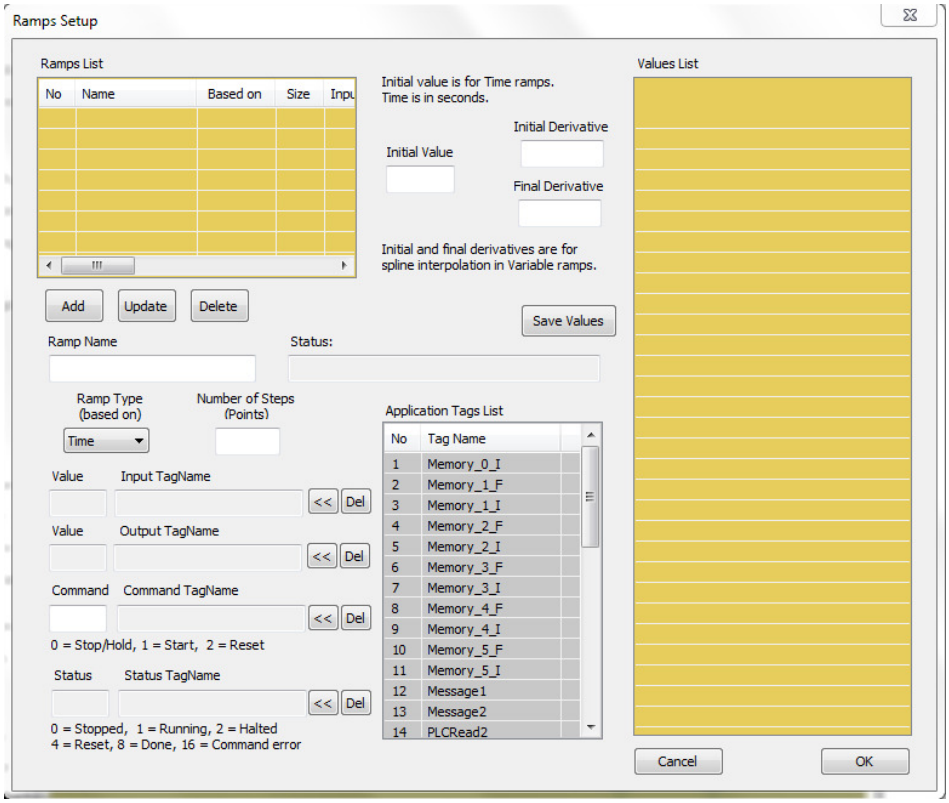
6.1. Description – useful situations
6.2. Types of ramps: time ramp, value based ramp
6.3. Structure of a ramp: size, input, output, command and status
6.4. Setup of a ramp
6.5. Application examples
6.6. Conclusions

6.1) Description – useful situations

Many industrial processes require that a process variable is to change its value slowly up or down in a certain time interval, after a precise ramp. Chemistry reactions for rubber curing or other polymers, food industry, beer brewing, distileries, etc.



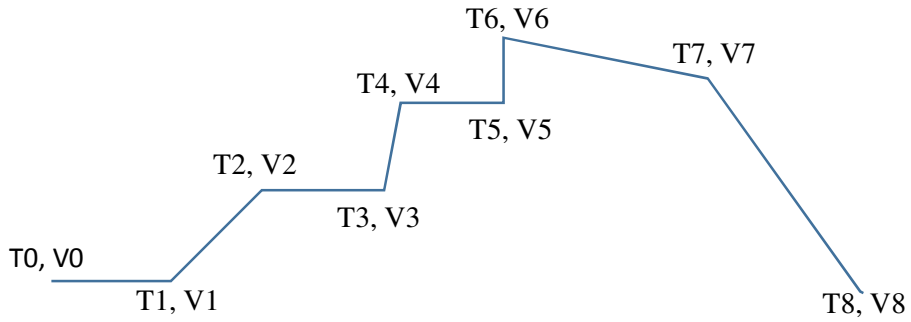
To start the ramps setup window select **Utils->Ramps**. If the configuration file does not exist an information message will show up. Click OK and the **Ramps Setup** window will open.



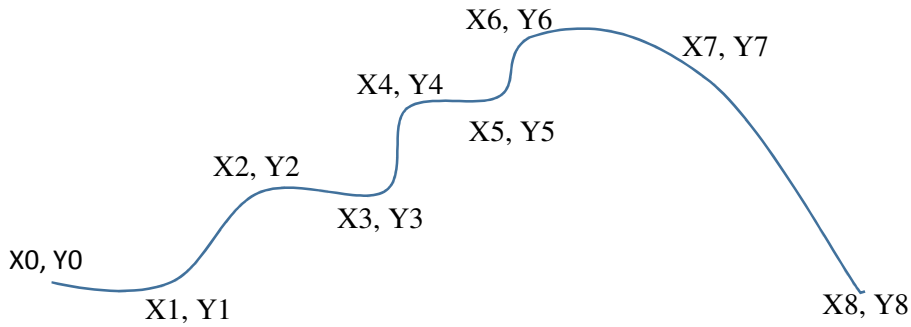
The list of ramps, **Ramps List,** is on the left of the window. On the right there is a list with the data for each ramp, **Values List**. The list of **Application Tags** is at the bottom. Under the **Ramps List** are the settings that can be chosen to define a ramp.

6.2) Types of ramps: time ramp, value based ramp

The classic way to define a ramp is based on time. Let N be the number of steps of the ramp. For every step a time and a value are associated, (T1, V1), (T2, V2), … (Tn, Vn). The ramp starts from the time T0 with an initial value V0. At the end of the time T1 the value will be V1, at the end of T2 the value will be V2, so on and so forth. The value V, which is the ***output*** variable, can be very precisely computed based on the time value which is the ***input*** variable. To run a time ramp one has to keep track of the time elapsed and to apply the changes accordingly, by interpolation between the 2 sets of values which are relevant for the actual time. The interpolation in linear. The example in the figure bellow is a time ramp with 8 steps. There are straight lines between the 9 points.



Another type of ramp can be defined, if instead of time we use just an input value. In this case the output value, ***y***, is a function of the input value, ***x***, ***y = f(x)*** , where the function is defined as a set of pairs (x1, y1), (x2, y2), … (xn, yn), with the condition that x1<x2<…<xn. The value of the output, ***y***, will be an interpolation between the 2 sets (***y***j, ***y***j+1) which are relevant for the input value, ***x ϵ (xj, xj+1)***. For this type of ramps, the user can choose between linear interpolation and spline interpolation. In the spline interpolation the function has the properties that the left and right, first and second, derivatives are equal in any point, resulting in a path that is smooth, with no corners.



That is why, the ramps can be defined in 2 ways: *time* based and *variable* based. In the case of a time based ramp, the program will keep track of the elapsed time. In the case of a variable based, the program will just compute the output based on the input.

6.3) Structure of a ramp: size, input, output, command and status

The structure of a ramp is defined, in the first place, by the number of steps required by the application. For each step a pair of numbers has to be setup: a time in seconds and the step's final value. In the case of a variable based ramp each step will be setup with 2 values and a choice: input variable, output value and the type of interpolation (linear or spline).

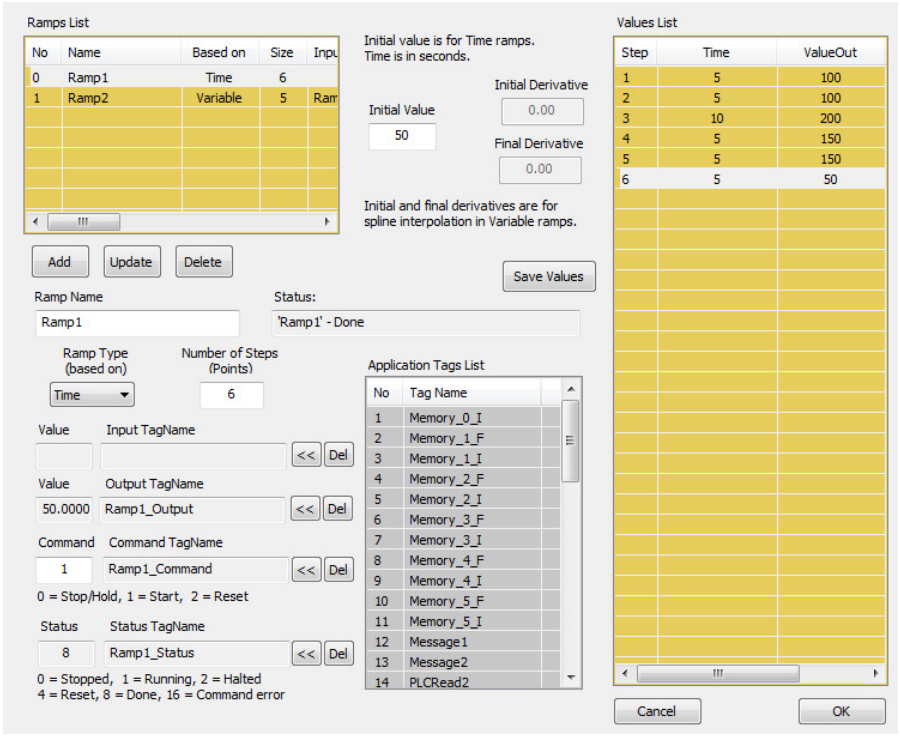
To control a ramp, a **command** integer and a **status** integer are defined. The command can *start*, *stop/hold* and *reset* the ramp. The status will show the result of the command.

Command	Description
0	**Stop/Halt** (if running, will halt at that time, the timing will stop)
1	**Start** (from the beginning or from the time where it halted)
2	**Reset** (reset the timer and start from the beginning)

Status	Description
0	Stopped
1	Running
2	Halted
4	Reset acknowledge
8	Done (finished running)
16	Command error (<> 0, 1 or 2)

6.4) Setup of a ramp

To setup a time ramp type a name in the **Ramp Name** text box, select *Time* in the **Ramp Type** combo box, type the number of steps in **Number of Steps** edit box and press **Add** button. To change the settings of a ramp select the ramp in **Ramps List**, edit the changes and press **Update** button. To delete a ramp press **Delete**. In the figure bellow a time ramp, **Ramp1**, was defined with 6 steps, a total time of 35 seconds, initial value 50, **Output TagName** *Ramp1_Output*, **Command TagName** *Ramp1_Command* and **Status TagName** *Ramp1_Status*.

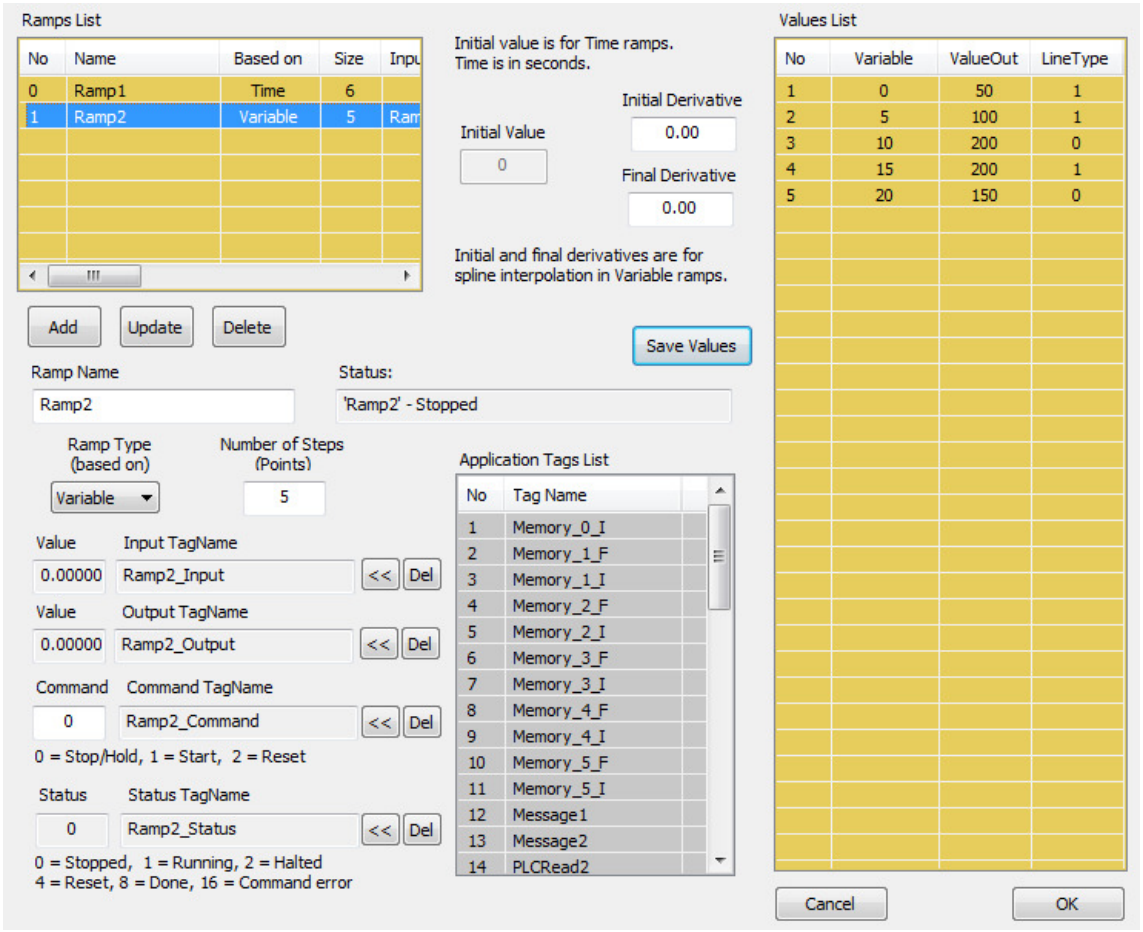


To edit the **Time** and **ValueOut** double click on the table cells. Press Enter to finish. Press **Save Values** button before selecting another ramp from **Ramps List**. To pick another tag for output, command or status, select the tag from the **Application Tags List** and press **<<** button.

The output tag can be integer or float. In our example the output tag is a float.

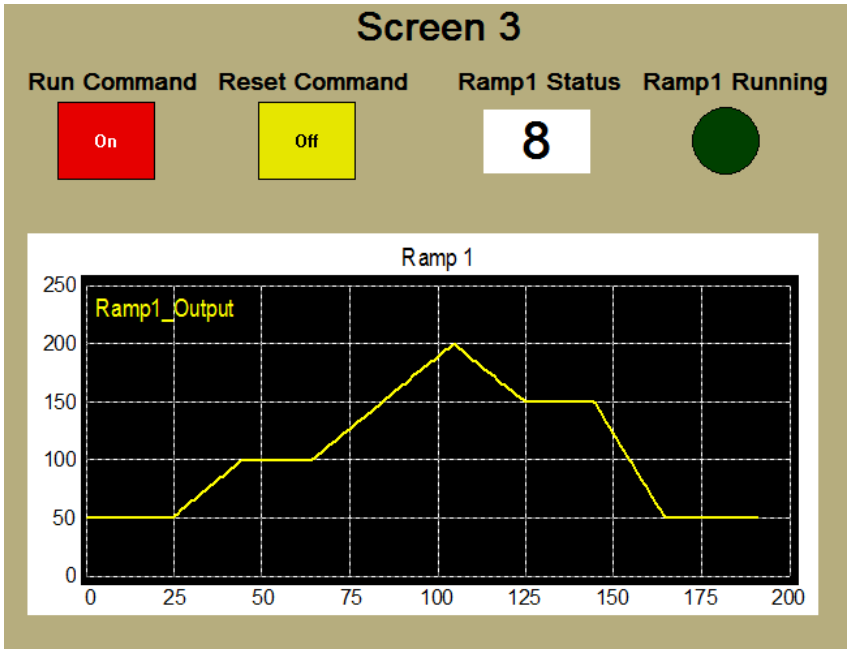
In the figure bellow a *Variable* **Ramp Type** was defined. It has 5 steps, the input **Variable** spans from 0 to 20, the **ValueOut** has values from 50 to 200. For the **LineType**, 1 means spline interpolation and 0 means linear interpolation. Initial and final derivatives are 0.0 so the function will start horizontally from the first point and will end horizontally to the last point.

For this type of ramp the **Input TagName** is required, and *Ramp2_Input* tag was selected.



6.5) Application examples

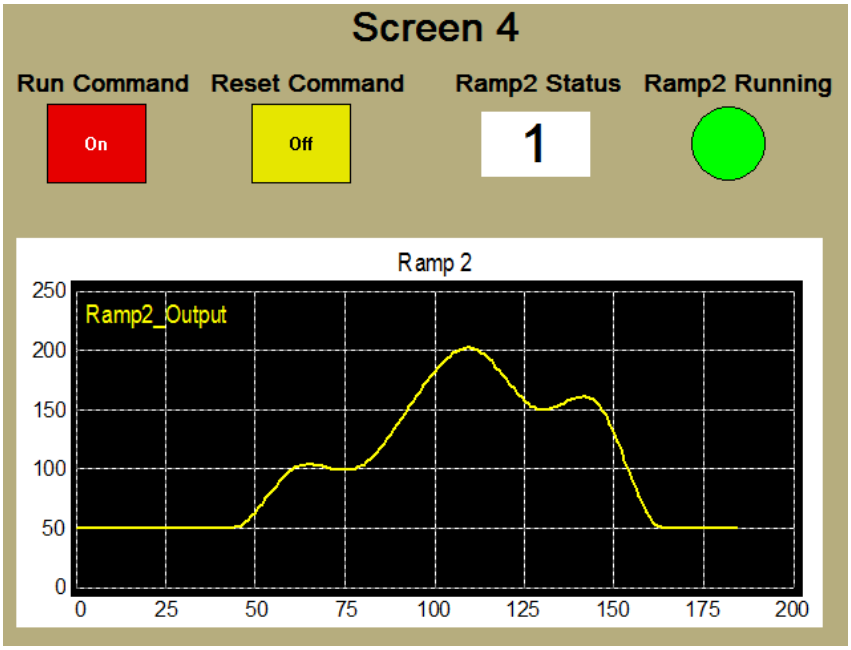
To create an example with a time ramp, a trend graph is used together with 2 buttons, Run and Reset, an indicator, to show when the ramp is running, and a numerical display to show the value of the status integer. The trend has a time base of 0.25 seconds as the unit for the X axis and a value span from 0 to 250 for Y axis.
Screen 3 shows the result of running *Ramp1*. Status 8 means the ramp has finished.



The user can define a *Time* ramp were the output is linear with the time in a ratio x to y. This output can be used as the input for a *Variable* type ramp. The difference now is that the spline interpolation can be used for a time-like ramp.

In Screen 4, a 0 to 30 seconds timer is used as 0 to 20 output, used as an input value for a *Variable* type ramp.

The ramp is similar with the one in the screen 3 but is using spline interpolation.



6.6) Conclusions

Ramps are of two types: time based and input variable based. To define a time based ramp select and edit the number of steps with times and values. To control the ramp select the command and status tags from the application's tags. The variable based ramps are defined as number of pairs of values (x,y). In this type of ramps, linear or spline interpolation can be used.

7. Data Logging

Introduction

The acronym SCADA stands for *Supervisory Control And Data Acquisition*. From here results that data acquisition is a really important part of a SCADA program. The possibility of recording data from an industrial process, using computers, has transformed the way of making and thinking industrial automation. The data recorded in one plant, somewhere in Europe or in China, can be accessed remotely and in real time from anywhere in the world.

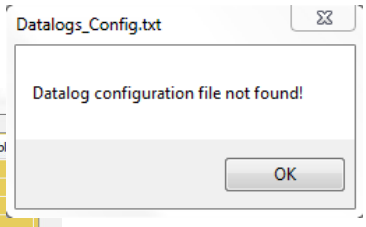
This chapter will cover the following topics:

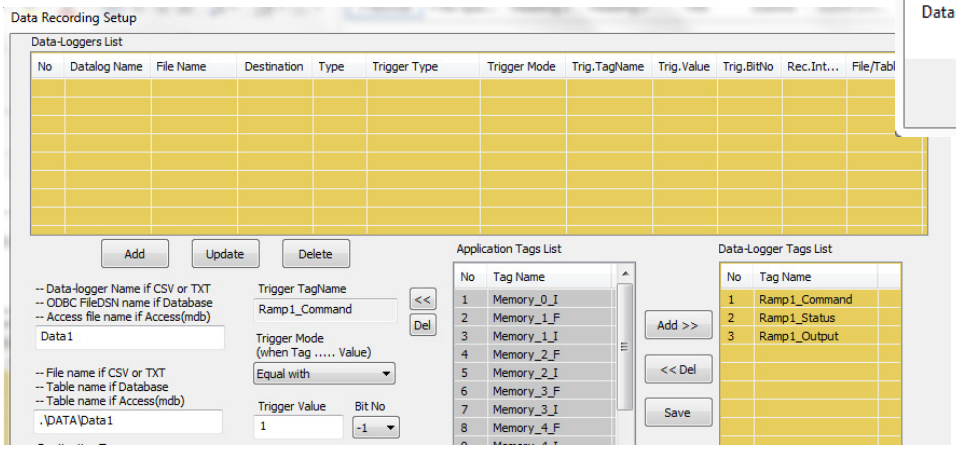
7.1. Description – useful situations
7.2. Types of data logging
7.3. Recording trigger types
7.4. Recording in files - Text files and CSV files
7.5. Recording in databases
7.6. Recording in Microsoft Access (mdb) databases
7.7. Other application examples
7.8. Conclusions

7.1) Description – useful situations

There are many reasons for recording data from an industrial process. Few of them are enumerated here: safety of the operations, maintenance supervision, batch data for quality insurance, production output and availability, OEE - Overall Equipment Effectiveness, etc. All these reasons demand different ways of recording data. Some require data at the end of the day, others require the recording of any change in the state of a machine. A good data logging program has to be flexible and to offer multiple choices for the setup of data recording.

To open the **Data Recording Setup** dialog window, select from the menu **Utils->Data Recording**. A notification message will inform if no data was previously configured. Press **OK**.





The list of the **Data-Loggers** is on the top of the window. The **Application Tags List** is under it.

7.2) Types of data logging

Any data logging will have to record *data*. The *data* is a set of tags: integers, reals or strings. When the *data* is recorded a date/time stamp is associated with it. The *data* can be saved in text files, in CSV files or in databases. In a database a data logging will be saved in a database table. *FeSCADA* is using the format from the table below to record a set of tags. The first line is the table header with the "Date1", "Time1", and all the *tag names* in the order they were selected from the **Application Tags List**.

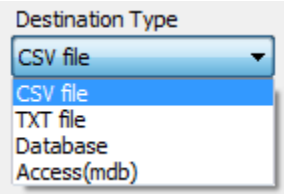
Observation! For text and CSV files the file header will be: "Date", "Time", plus the other tags names. Only for ODBC and MS Access databases the header will use "Date1" and "Time1". The reason is that "Date" and "Time" are reserved words for databases programming and SQL commands.

In the table below, there are 3 tags for the dataset. The recording took place every second when *Ramp1_Command* was true (=1).

Date1	*Time1*	*Ramp1_Command*	*Ramp1_Status*	*Ramp1_Output*
4/3/2019	14:06:09	1	1	51.09
4/3/2019	14:06:10	1	1	61.23
4/3/2019	14:06:11	1	1	72.15

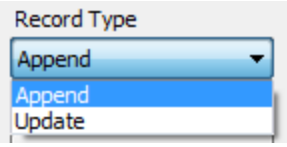
There are four formats for **Destination Type**:

- **CSV file** – this format is known as "Comma Separated Variable" and is recognized by Excel.
- **TXT file** – a normal text file which can be opened with any text file editor
- **Database** – ODBC database. The datalogger name has to be defined in "odbcad32.exe" as a File DSN. See *5) Recording in databases*, for details.
- **Access(mdb)** – An MS Access database file with the extention ".mdb"



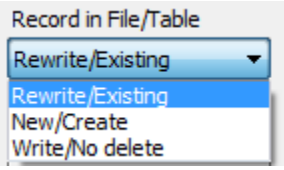
There are two modes for recording new data:

- **Append** – add data to the end of the file or database table
- **Update** – overwrite one set of data. This mode is more proned for databases, to update the status of a machine in a table from which other users can read at any time.

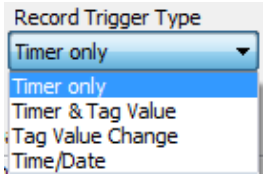


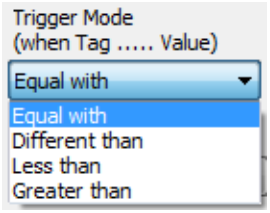
When the program is started there are three ways in which every file/table can be accessed:

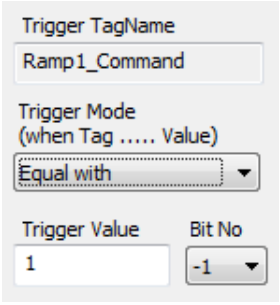
- **Rewrite/Existing** – delete and create a new one with the same name
- **New/Create** – create the file/table based on the name given by the user but attach to the basic name a suffix with the date and the time of creation, so that the final name becomes unique
- **Write/No delete** – it will open the file/table and will append at the end of it



7.3) Recording trigger types

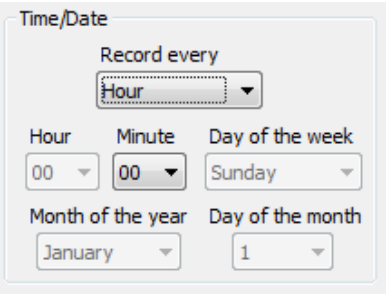






At the moment of recording, the values of the data tags in the dataset are captured and inserted in the recording command. There can be four different ways to trigger the recording command:

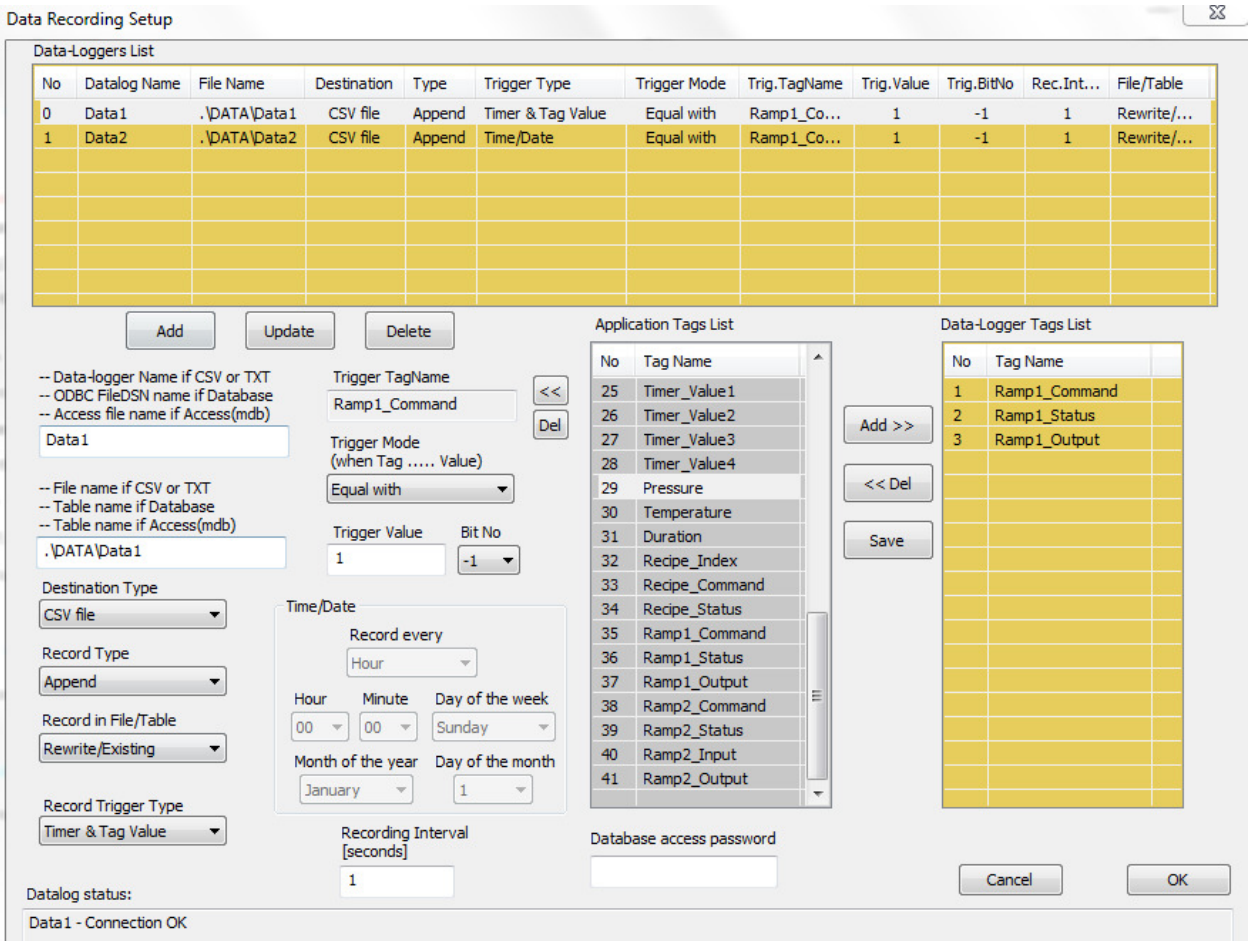
- **Timer only** – The simplest way of recording is based on a timer that will start once *FeSCADA* is starting. The timer will restart immediately after each recording. The timer value is set by the user.
- **Timer & Trigger Value** – This trigger is using a relational condition (=, <>, <, >) between a trigger tag and a constant number, or a Boolean condition, a bit number of the tag checked for true or false (1 or 0). When the condition is true and the timer has passed the recording is performed. The timer will restart automatically. The difference is that the recording will stop if the condition is false.
- **Tag Value Change** – This mode is monitoring a tag value for a change. If the timer has passed and the tag has a new value the recording will be made. The timer is set to limit the number of recordings if the tag value is changing fast. The fastest recording cycle is limited to 0.1 second, but the user can use other values for the timer.
- **Time/Date** – Using the computer's real time clock the user can set exact times and dates when the recording is to be made. The minimum interval is *Hour*, in the format MM (minutes). The next one is *Day* with HH:MM (hour, minutes). *Week* is the third one, in the format Day of the Week + HH:MM. *Month* is the 4th type, in the format Day of the month + HH:MM. The last one is *Year*, in the format Month of the year + Day of the month + HH:MM.



7.4) Recording in files - Text files and CSV files

To record in text or CSV files the user has to type in the **Datalog Name**, the **File Name**, select the **Destination Type** and press the **Add** button. With **Update** button the new changes are saved. With **Delete** a datalog is removed. For every datalog the user has to define the tags that are going to be saved. For this select from the **Application Tags List** and press **Add >>** button. The **<< Del** button will remove tags from the **Data-Logger Tags List**. Before you select another datalog press **Save** button to save the chosen tags in a configuration file. Press **OK** to exit and save the datalogs.

By default the file will be saved in the same folder where *FeSCADA* runs. To save in a different folder the path to the folder has to be included in the file name. For example, to record in a folder called "DATA" from the same folder with *FeSCADA*, the file name has to be: ".\DATA\Data2", where "Data2" is the name of the file and ".\DATA\" is the folder. You do not need to write the extensions ".txt" or ".csv" because the program will do it automatically.



Some special characters can be used in the name of the file: %Y, %M, %D, %H %m. These characters will be replaced with the strings: "_YYYY" the current year, "_MM" the current month, "_DD" the current day of the month, "_HH" the current hour, "_mm" the current minute.

The program will evaluate the "file name" using the real time clock of the computer. Whenever the "file name" is changing, a new file will be created and the old file will be closed.

For example, if the user wants to have a file for every hour of the day, one can setup a file name "Data%H" in the **File name/Table name** text box. That will create 24 files, in 24 hours, of the format: "Data_00.csv", "Data_01.csv", … "Data_23.csv". If the files exist, they are overwritten. In this way one can have only the data from the last 24 hours.

These special characters can be used for databases too, to set the name of the table in which the tags will be recorded.

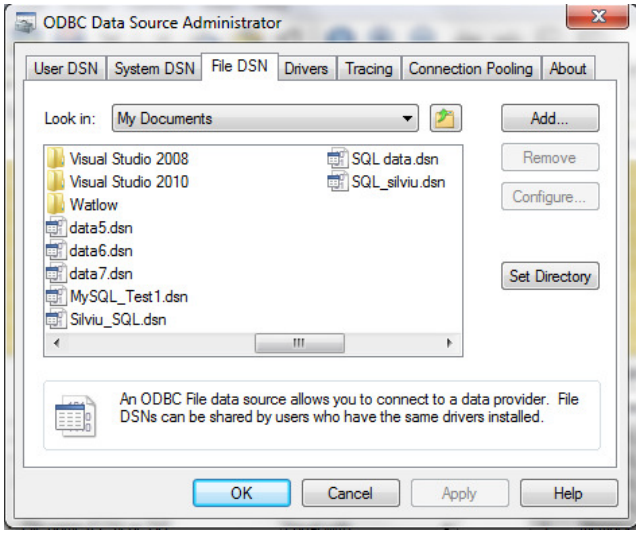
7.5) Recording in databases

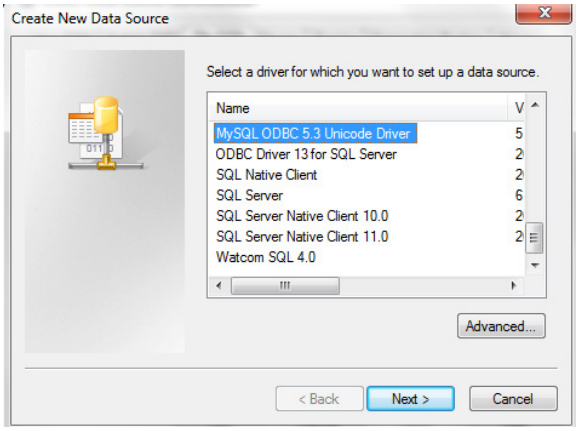
To record in a database, *FeSCADA* uses the ODBC drivers provided by Microsoft. ODBC stands for Open DataBase Connectivity. To start the **ODBC Data Source Administrator** on your computer go to **Start -> Control Panel** and double click the ODBC icon.

In the **File DSN** tab press **Add** to create a new .dsn file or select one to configure.

If the user presses **Add**, the **Create New Data Source** window will open. Select the driver you want and press **Next**>.

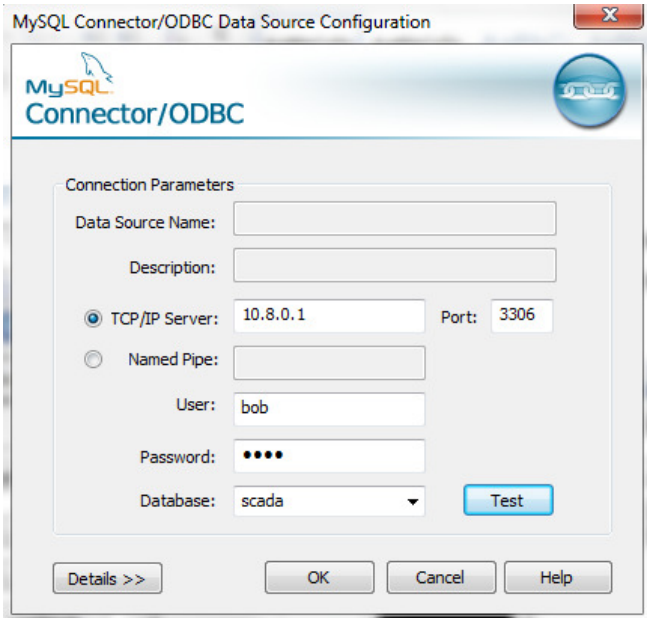
In our case we defined two DSN files: one for a MySQL database on a remote computer and another one for a Microsoft SQL Server database on the local computer.



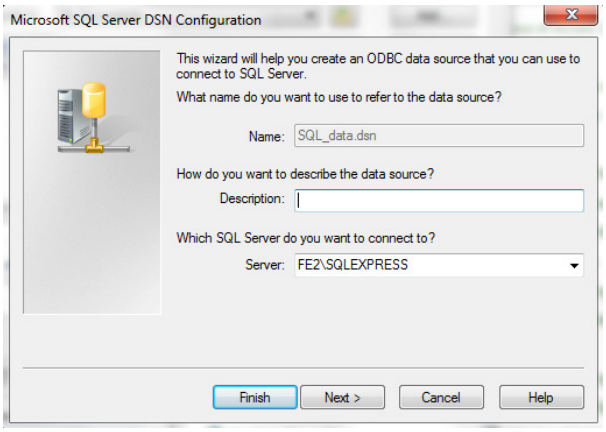


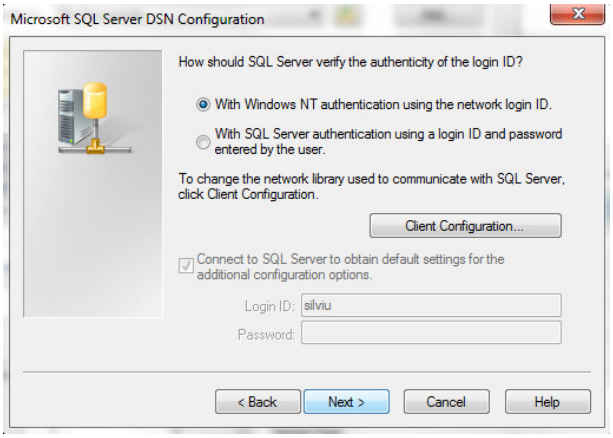
For MySQL we defined the file *SQL_silviu.dsn*. The file has all the details about how to connect to the database "scada" on a VPN network at the IP address 10.8.0.1, port 3306. The user "bob" has the rights to write in the database "scada".

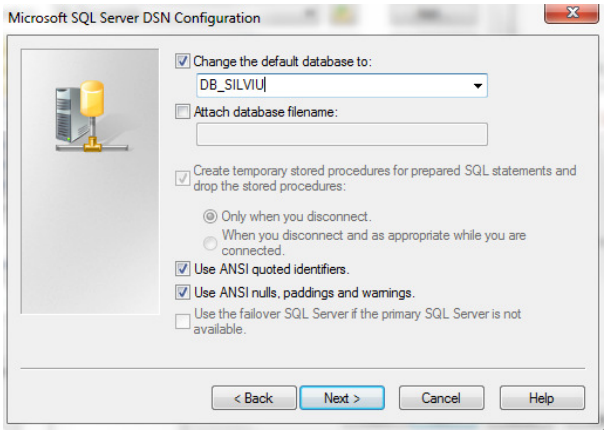
The access password has to be set for the datalog in the **Database Access Password** (under the **Application Tags List**).

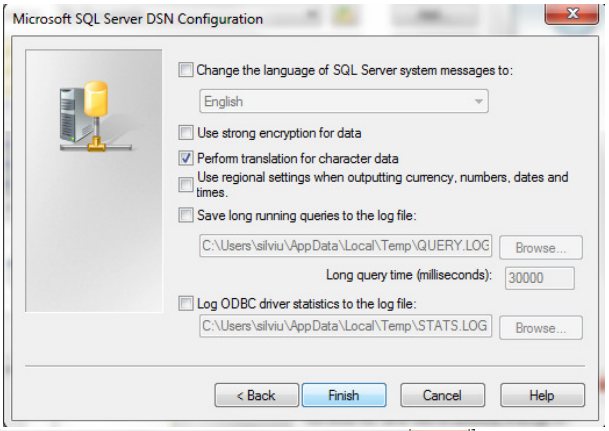


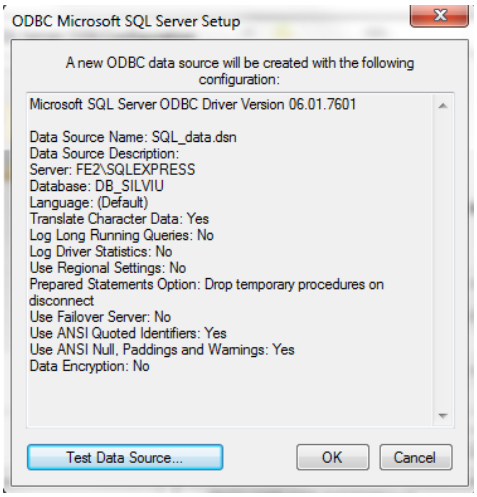
The following pictures are showing the steps required to setup the link with the SQL Server database, and save all the settings in the file *SQL_data.dsn*. The local server is FE2\SQLEXPRESS. It is connected with using Windows authentication loging ID. The database is "DB_SILVIU". Press **Next**. Press **Finish** button. Test Data Source. Test completed succesfully.

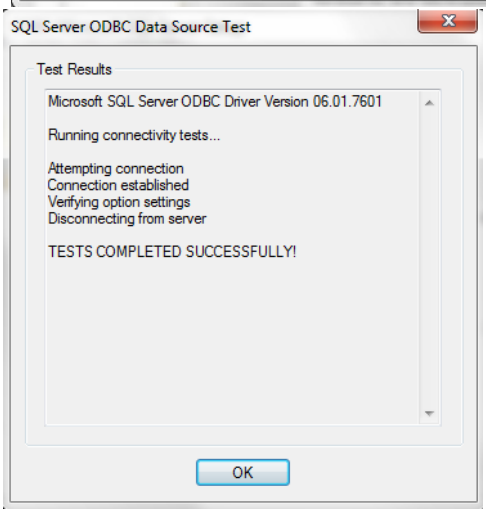








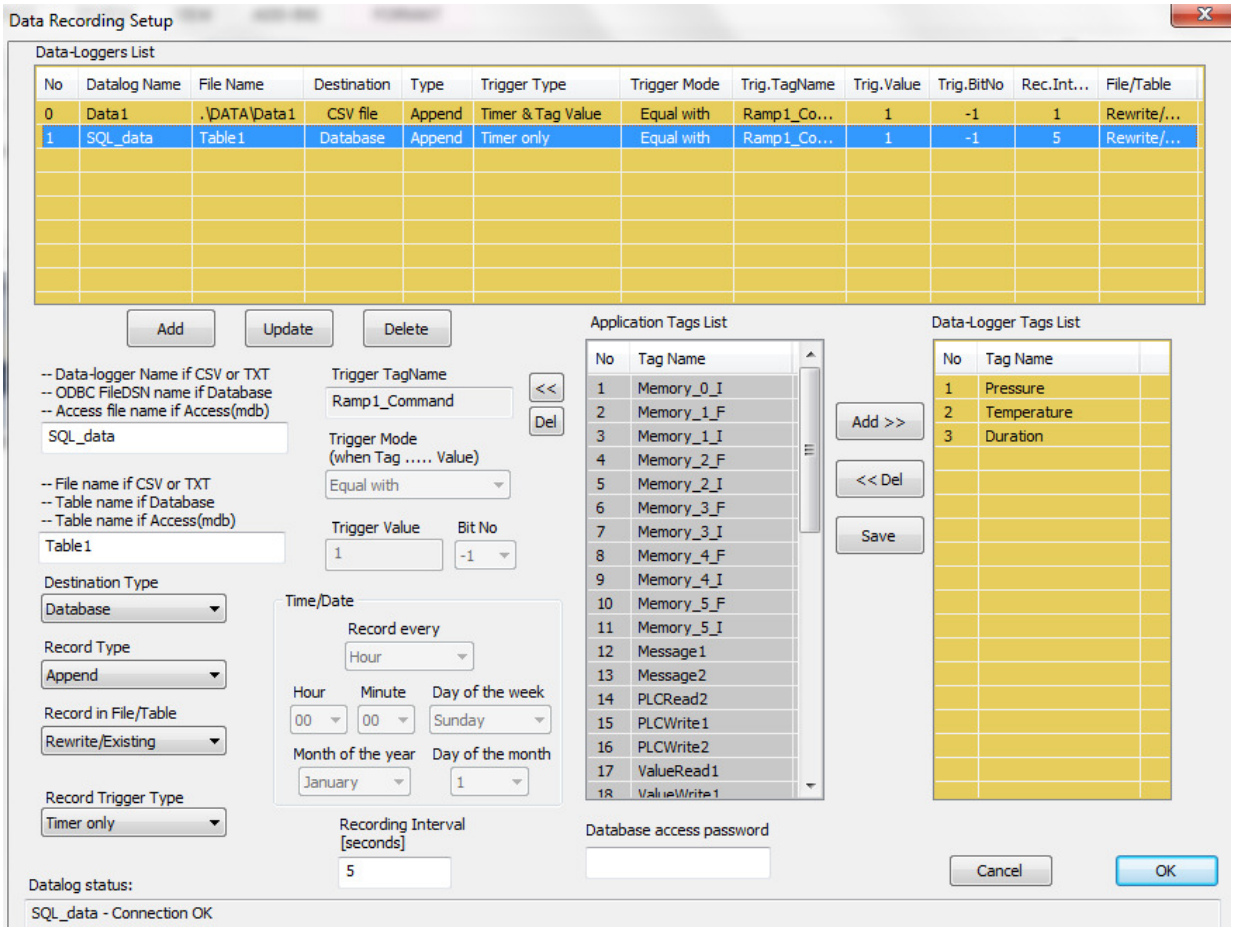




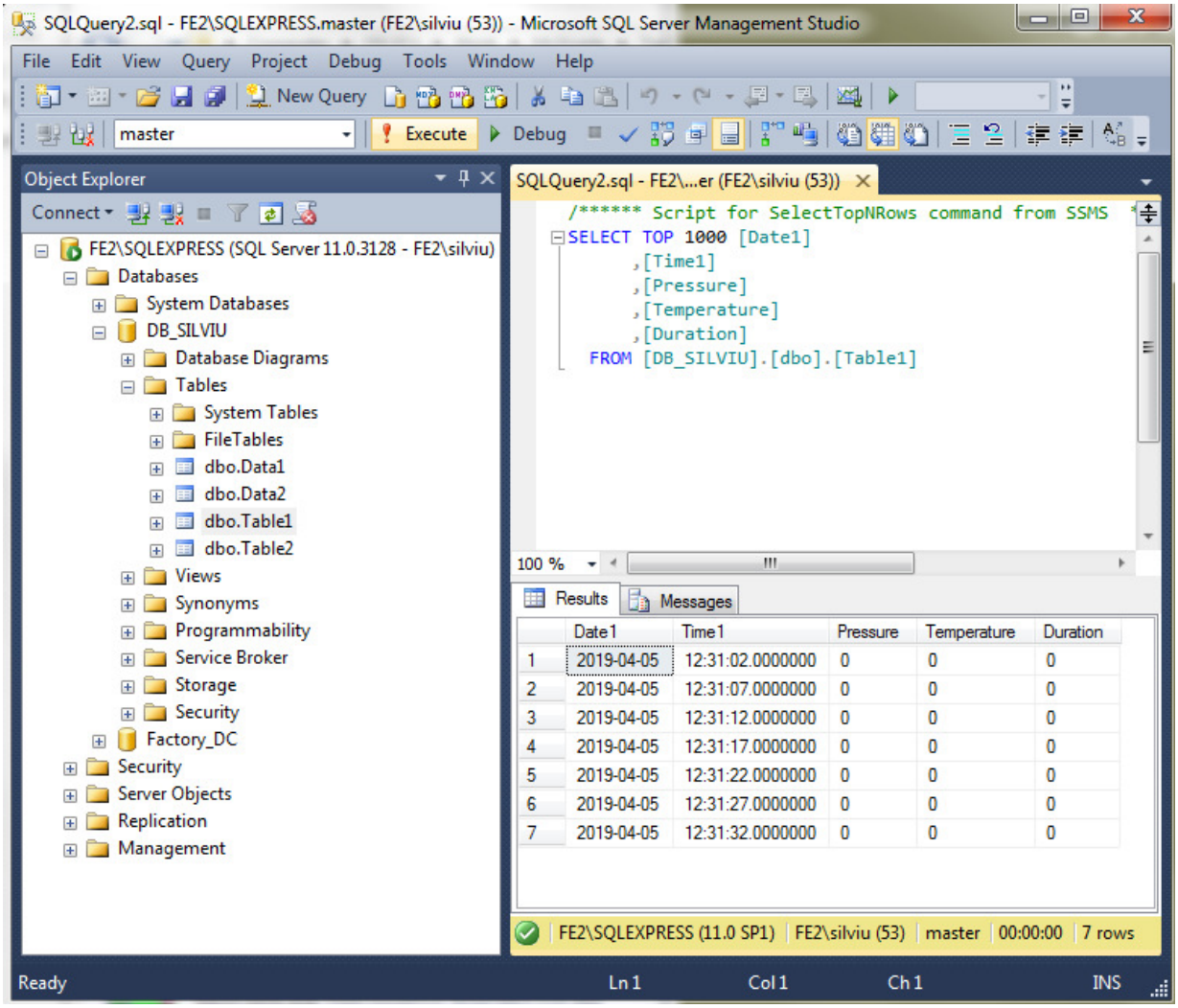
After the ODBC driver setup, the following settings are used in *FeSCADA* to record in a database:

- **ODBC FileDSN** = *SQL_data*
- **Table Name** = *Table1*
- **Destination Type** = *Database*
- **Record Type** = *Append*
- **Record in Table** = *Rewrite/Existing*
- **Record Trigger Type** = *Timer Only*
- **Recording Interval** = *5* seconds
- **Database Access Password** = none

When *FeSCADA* program starts it will try to connect to all the databases that are defined. For any database the **Datalog status:** text box, at the bottom of the **Data Recording Setup** window, will show the status of the connection. If an error occurs, the program will try to re-connect to the database every other 5 seconds.

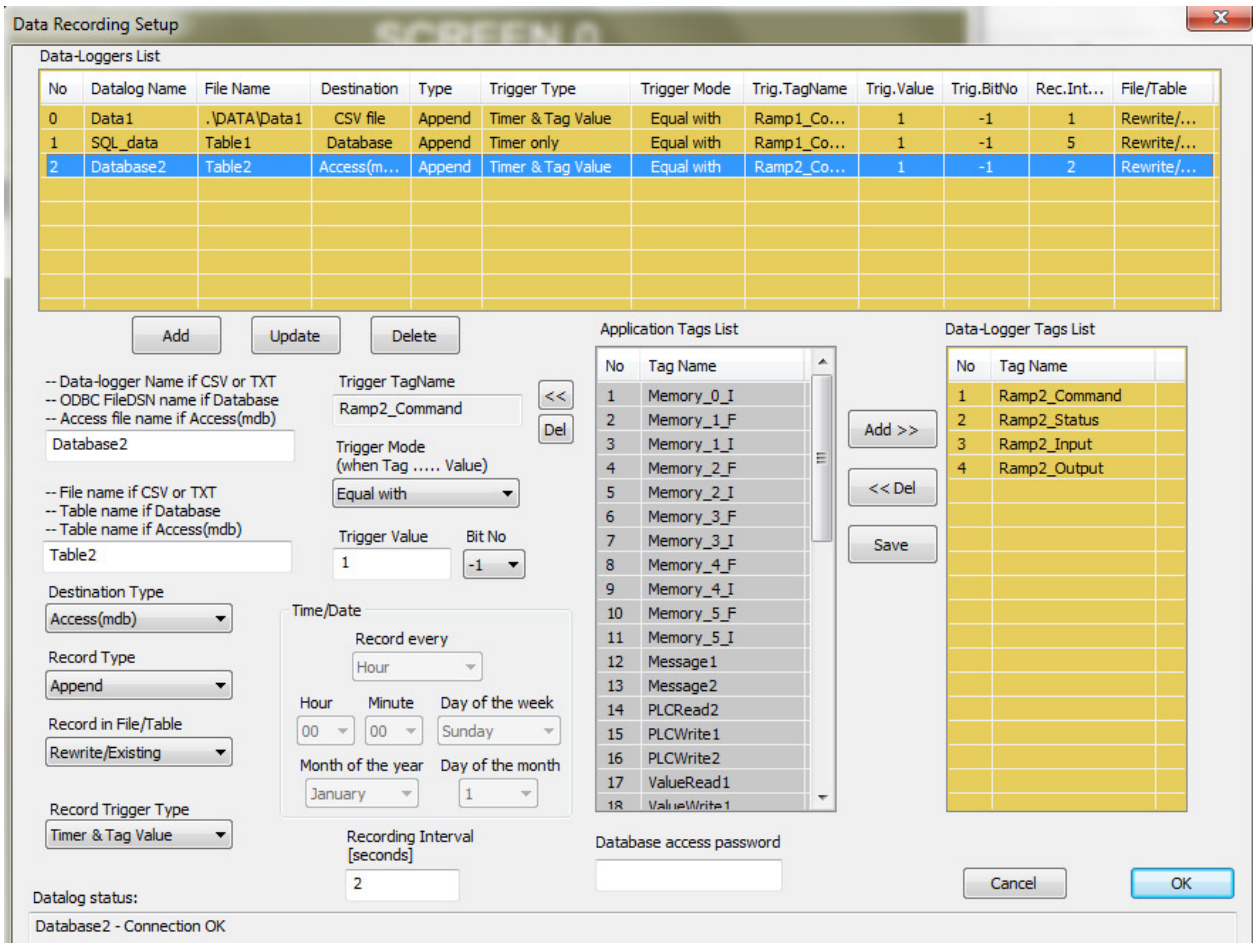


The resulting data in *Table1* of the SQL database FE2/SQLEXPRESS – DB_SILVIU can be seen in the picture below.

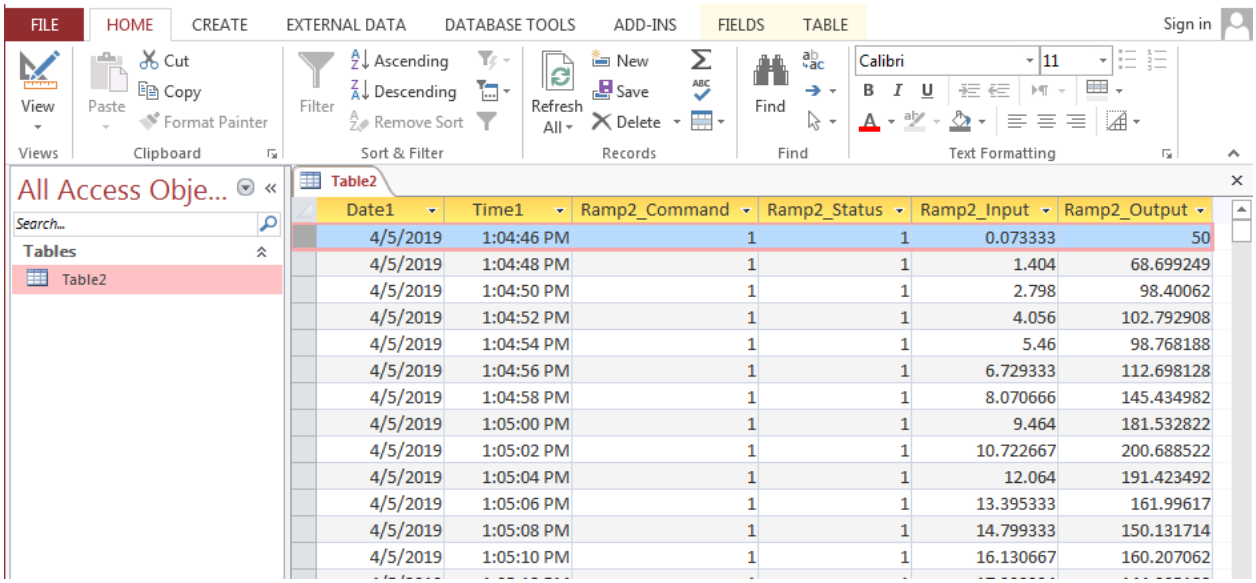


7.6) Recording in Microsoft Access (mdb) databases

To record in a Microsoft Access database, create an Access file "Database2.mdb" in the same folder where *FeSCADA* is running. Add a new datalog. Select **Access File Name** = *Database2* and select **Destination Type** = *Access(mdb)*. Setup the list of tags to be recorded based on **Timer & Tag Value**., every *2* seconds when *Ramp2_Command* is true. See the complete settings in the picture below.



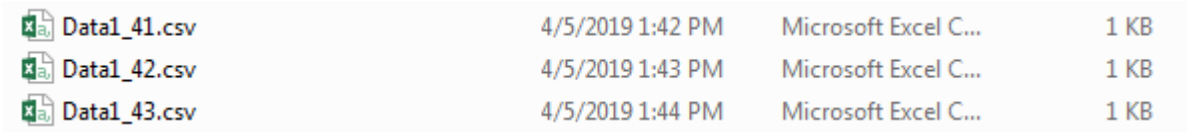
The results of *Table2* in the *Database2.mdb*, after *Ramp2_Command* was started.



7.7) Other application examples

Using the option **Record in File** = *New/Create* for the data-logger with the **File Name** = ".*DATA**Data1*" will create a new file every time the program starts. In this file the name "Data1" will be padded with the suffix that includes the year, month, day of the month, the hour, minute and the second. See the picture below.

Using the special characters "%m" in the title of **File Name** = ".*DATA**Data1%m*", and selecting **Record in Table** = *Rewrite/Existing*, a new file will be created every new minute, from 00 to 59 as a file name Data1_xx.csv. See figure below. If they were existing they will be rewritten.



Using the option **Record in Table** = *Write/No delete* in the example above, will not delete the old data in the files. The files will be open and the data will be appended at the end.

The same rules apply with table names in databases.

7.8) Conclusions

For the data recording of the tags values, *FeSCADA* offers four types of destinations: text file, CSV file, any database for which an ODBC driver is installed on your computer, and Microsoft Access database.

The data can be updated or appended. The files/tables can be rewritten, created as unique names, with new names which will include the date/time in the file/table name, or the data can be added at the end of the file/table without deleting the old one.

By using the special characters %Y, %M, %D, %H, %m in the name of the file/table, new files/tables will be created as soon as the date/time from the real time clock of the computer is changing: the year, the month, the day of the month, the hour, or the minute.

The recording trigger options are covering the following options: timer only, timer & tag value, tag value change, and date/time.

There is no program limit for how many data-loggers the application can run.

8. Emails

Introduction

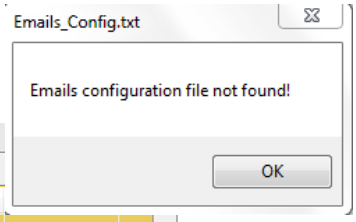
Smart mobile phones are changing the way the peoples communicate with each other. Today, an email can be viewed almost as fast as a text message. In this new communication era, having a machine sending you an email is not anymore a strange or luxuriant feature. And it can be very useful for maintenance and troubleshooting. *FeSCADA* was developed with the possibility to send emails that incorporate tags data in the email body text. This chapter will cover the following topics:

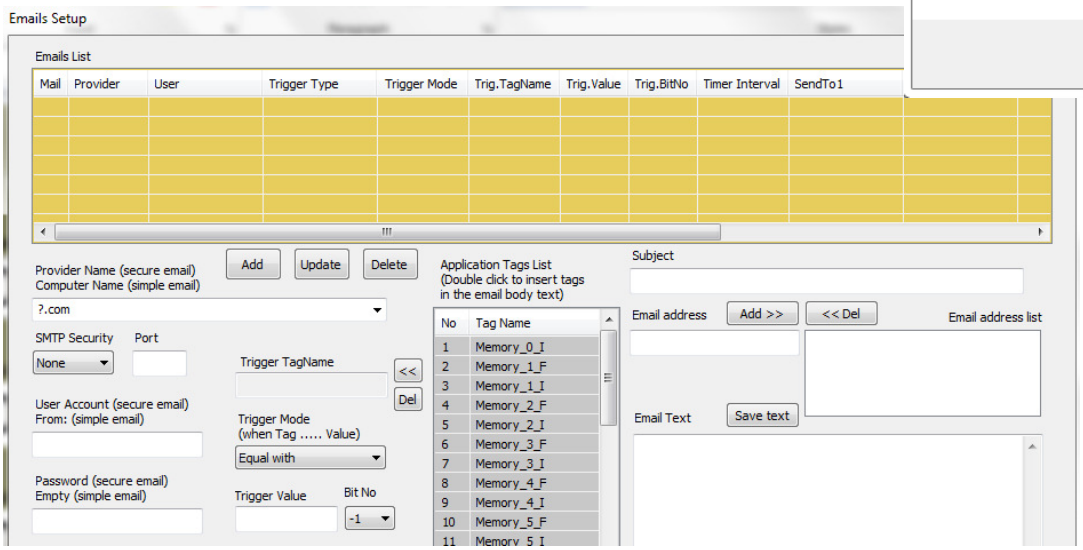
8.1. Description – useful situations
8.2. Types of emails
8.3. Emailing trigger types
8.4. Setup of an email
8.5. Sending text messages via email
8.6. Application examples
8.7. Conclusions

8.1) Description – useful situations

Like for data recording, there are many reasons for emailing data to targeted users from an industrial machine/process. If you are a maintenance technician or engineer, the sooner you can receive an alarm message, the sooner you can react. The fact that you do not have to check every other few minutes a display, to see if there are or not alarms present, is a big relief. Another advantage is that the right maintenance technician is alerted (electrical or mechanical), and is not like in the classic case where the machine operator will have to call in for anybody from the maintenance department.

To open the **Emails Setup** dialog window, select from the menu: **Utils->E-mails**. A notification message will inform if no emails were previously configured. Press **OK**.





The list of the **Emails** is on the top of the window. The **Application Tags List** is under it.

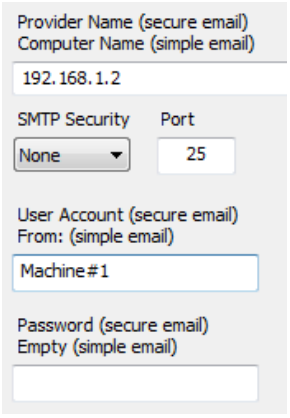
8.2) Types of emails

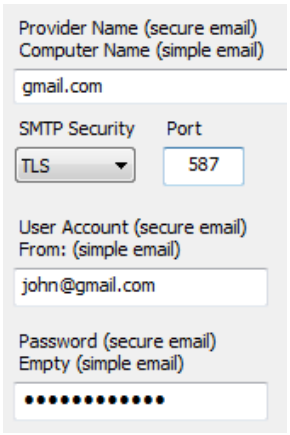
There are two types of emails that *FeSCADA* is able to send: *simple* or *secure*.

A *simple* email is using the old, 1982, SMTP (Simple Mail Transfer Protocol) to send an electronic mail. There is no encryption or protection for the transferred login data through the network. This type of email is still used, but only in local networks (homes, factories) that are protected from the Internet with firewalls. The protocol is using TCP/IP connection on the port number 25.

A *secure* email is using the *Extended* SMTP adopted as a new standard in 2008. It adds a method of encrypted authentication of the user, who needs to log in with a user ID and a password to the email server (gmail, yahoo, or other email services providers). If the authentication is successful the email is received to be delivered to any other destination on the Internet. The authentication can be SSL (Secure Socket Layer) on port number 465, or TLS (Transport Layer Security) on port number 587.

In the pictures on the right, settings for a *simple* and a *secure* emails are shown.

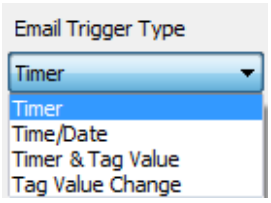




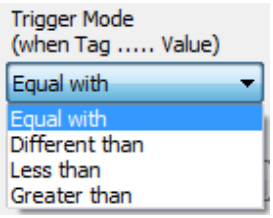
8.3) Emailing trigger types

There can be four different ways to trigger the transmission of an email:

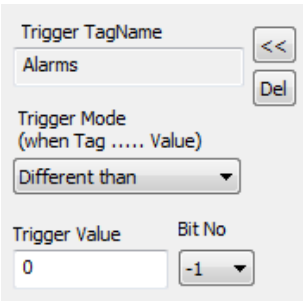
- **Timer only** – The simplest way of sending an email is based on a timer that will start once *FeSCADA* is starting. The timer will restart immediately after each email transmission. The timer value, set by the user, is in minutes.



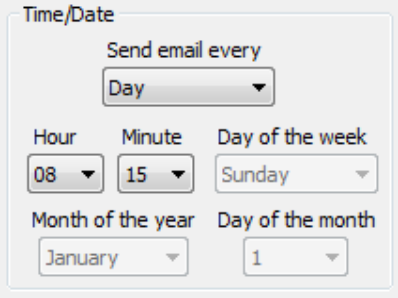
- **Timer & Trigger Value** – This trigger is using a relational condition (=, <>, <, >) between a trigger tag and a constant number, or a Boolean condition, a bit number of the tag checked for true or false (1 or 0). When the condition is true, and the timer expired, the email will be sent and the timer will restart, the same as for Timer Only. The difference is that the email will not be sent if the condition is false.



- **Tag Value Change** – This mode is monitoring a tag value for a change. If the timer has passed and the tag has a new value the email will be sent. The timer is set to limit the number of emails sent if the tag value is changing fast. The fastest cycle is limited to 1 minute, but the user can use other values for the timer.

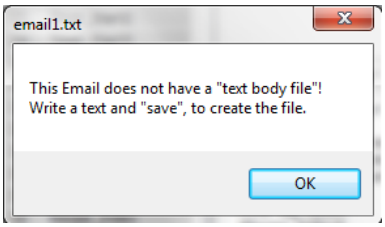


- **Time/Date** – Using the computer's real time clock the user can set exact times and dates when an email is to be sent. The minimum interval is *Hour*, in the format MM (minutes). The next one is *Day* with HH:MM (hour, minutes). *Week* is the third one, in the format Day of the Week + HH:MM. *Month* is the 4th type, in the format Day of the month + HH:MM. The last one is *Year*, in the format Month of the year + Day of the month + HH:MM.

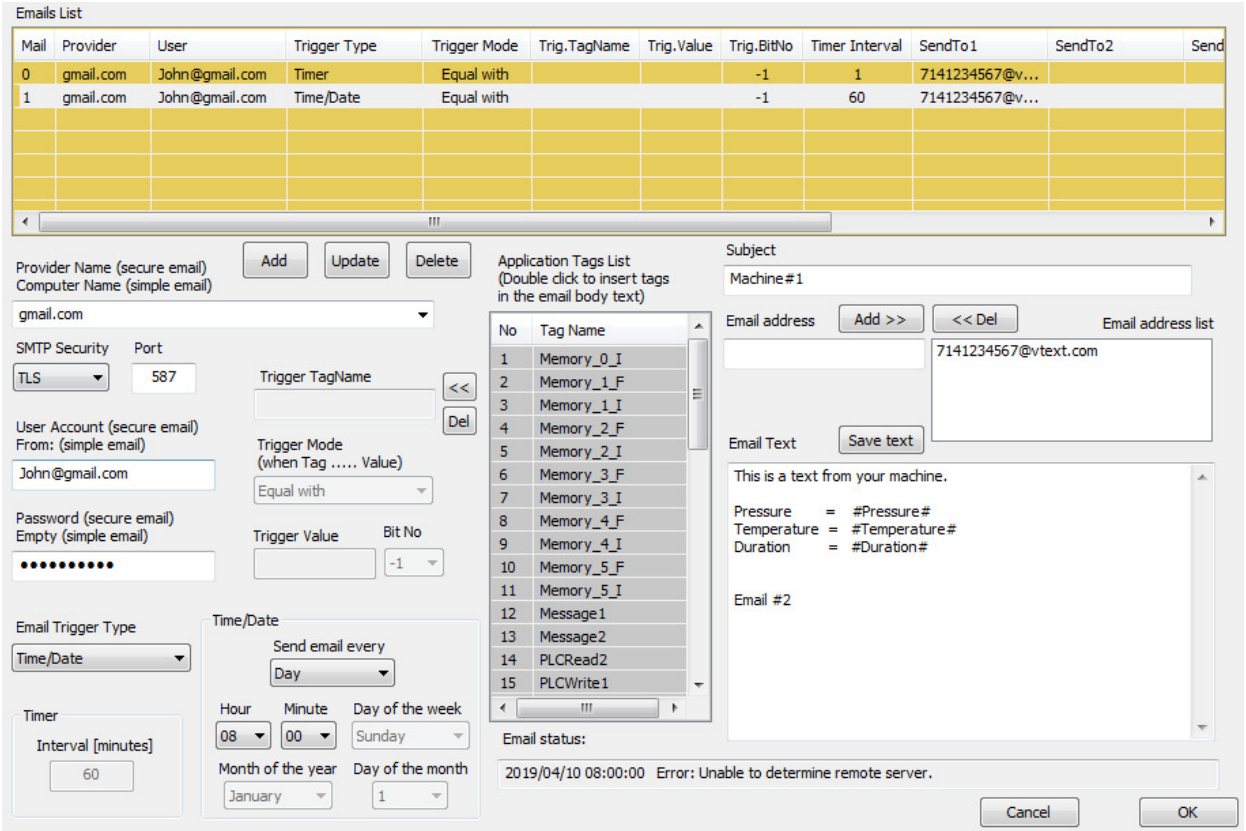


8.4) Setup of an email

To setup a secure email the user has to type the SMTP server address in the **Provider Name**, select the *TLS* option in **SMTP Security**, type *587* in **Port** edit box, type your email address in **User Account**, type your password in **Password**, select the **Email Trigger Type** (*Timer* in our case), and press **Add** to insert it into **Emails List**. You can change settings and press **Update** to save, or press **Delete** to remove it from the list.



After inserting a new line in the **Emails List**, select the line. You will see a message that will inform that there is no text message defined for the selected email. Type your text in **Email Text** edit box. Press **Save text** button to save.

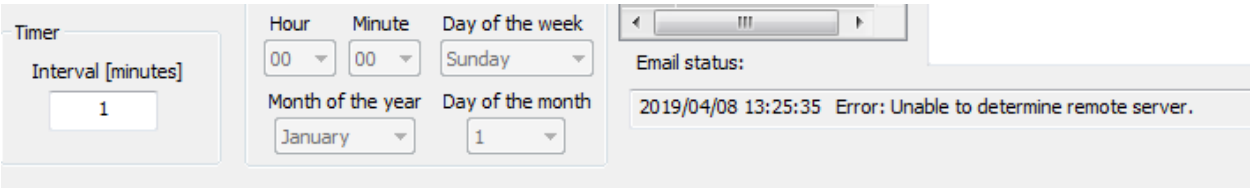


If you want to insert a tag value, from the moment of email transmission, in the text content, put the cursor where you want the data and double click in the **Application Tags List** the tag you want. The tag will be inserted in your text in the format *#TagName#*. At the moment of email transmission this text will be replaced by the tag's value: integer, float, or string.

You can send an email to maximum 5 recipient addresses. To setup the recipients list, type an email address in **Email address** text box and press **Add>>** button above it. The address will be transferred to **Email address list**. To remove a recipient, select it with the mouse and press **<<Del** button, above the **Email address list**. After inserting all the recipients' addresses, press **Update** button to record them in the **Emails List**.

Press **OK** to close the **Emails Setup** dialog window with saving the changes, or press **Cancel** to exit without saving.

The status of each email can be read in the **Email status** text box. The details include the date, time and a short description of the error. If the email was sent, the message will read: email *sent successfully*.



8.5) Sending text messages via email

Using your own email account you can send text messages to your mobile phone (or other phones) by sending an email to the email address specified by your mobile phone communication provider. For example, if your provider is Verizon Wireless, if you send an email to the address 7141234567@vtext.com, where we suppose the 7141234567 is your phone number, the email will be sent as a text message to your phone. The same procedure is used by other mobile phone communication providers. For AT&T the email has to be sent at 7141234567@txt.att.net.

8.6) Application examples

In many machines there is a series of bits that describe different alarms. These alarms can be spread on few different registers. Each register, if zero, means no alarm. Using the option **Email Trigger Type** = *Tag Value Change* for a tag, the user can receive an email every time when the alarm registers are changing their value.

8.7) Conclusions

FeSCADA program can send simple or secure emails. Every email can be sent to up to 5 recipients. The trigger types are: timer only, timer and tag value, tag value change and date/time. Tags can be inserted in the email's body text. They will be replaced with their values at the moment of email transmission.

9. Screens Editing

Introduction

The chapter will help the user to create screens and add different controls, in order to develop the visual interaction with a machine/process. Its content will cover the following topics:

9.1. Description – visualization of machine/process data
9.2. Modes of visualization for different data types
9.3. Graphic controls types: passive (display) and active (entry)
9.4. Screens windows and screen navigation
9.5. Controls creation and properties, tags assignment
    9.5.1. Navigation buttons
    9.5.2. Push buttons
    9.5.3. Numerical display/entry
    9.5.4. Digital indicator
    9.5.5. Analog indicator
    9.5.6. Value based message
    9.5.7. Sliders
    9.5.8. Trends
    9.5.9. Text
    9.5.10. Images
9.6. Single and multiple controls selection: alignment, size, arrange
9.7. Copy/Paste options
9.8. Transparency limitations: screen background and image controls
9.9. Application development – screens and controls number limits
9.10. Conclusions

9.1) Description – visualization of machine/process data

For a person it is much faster to interpret a change in a shape or in a color than to interpret a change of a number. In automation, the visualization of a machine/process is trying to accomplish exactly this: to reduce the effort (or increase the speed) of an operator in understanding the status of a machine/process.

A shape can change its length, width, or the geometry: circle, rectangle, triangle, etc. Also the shapes can change the position and orientation. The color is a natural signal for humans. Instinctual feelings tell us that red is danger and green is okay.

On the other hand, a computer is exchanging only numbers with a PLCs. So, what makes it possible to transform numbers in visual objects? And backwards, to transform operator interactions with shapes and colors in numbers sent to the machine/process?

That is the Graphical User Interface (GUI) of the computer. The GUI is including a display or a monitor that is capable to render graphical objects like: windows, icons, menus and pointers. To interact with a GUI the operator can use a keyboard, touchpad, mouse, joystick, pointing stick, etc.

From the point of view of *FeSCADA* program, any *control* is a *window* that has different visual properties. The link between the visual control window and the number from the machine/process is done when the user is assigning a *tag* to a *control*.

9.2) Modes of visualization for different data types

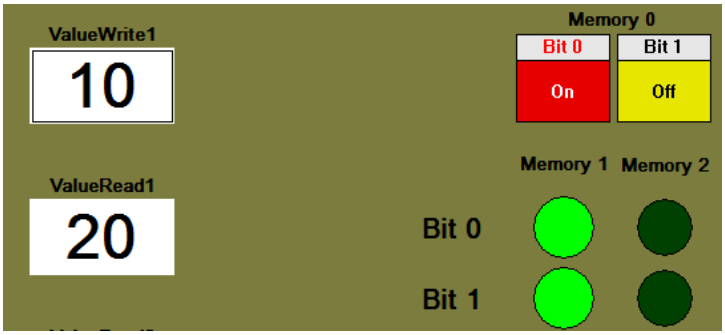
The types of data that a controller of a machine/process is using are: Boolean, integer, real, and strings. For an operator, the graphical images easy to understand are: on/off buttons and switches, on/off indicators for sensors, analog indicators for levels, pressures, and temperatures, sliders to change an analog value between minimum and maximum limits, colored text messages and images with symbols or pictures. The following table is showing the analog graphical control based on different data types.

Data type	Values (examples)	Visual control to display	Visual control to change
Boolean	0, 1 (on, off)	Digital indicator (color, shape) Image (show, hide)	Push button (color, shape)
integer	-100, -1, 0, 5, 750	Analog indicator (color, position, orientation)	Slider (color, position, orientation)
real	-10.1, 0.0, 1e-5, 5.25		
string	“Alarm present”	Text (size, color)	Text (size, color)

9.3) Graphic controls types: passive (display) and active (entry)

In automation there is a clear difference between inputs and outputs. The inputs can be read only. The outputs can be read and written (changed, set/reset). The same is the case with the visual controls in a SCADA program.

A *passive* graphic control is a digital or analog indicator. It is changing the color or the shape to reflect that the tag value has changed, but it will not respond to any input from the mouse or from the keyboard.

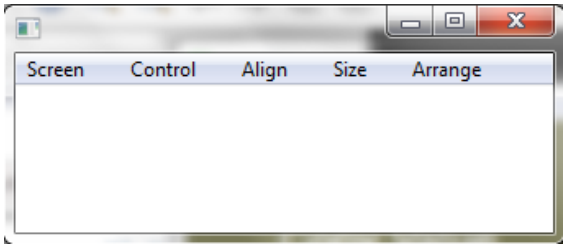


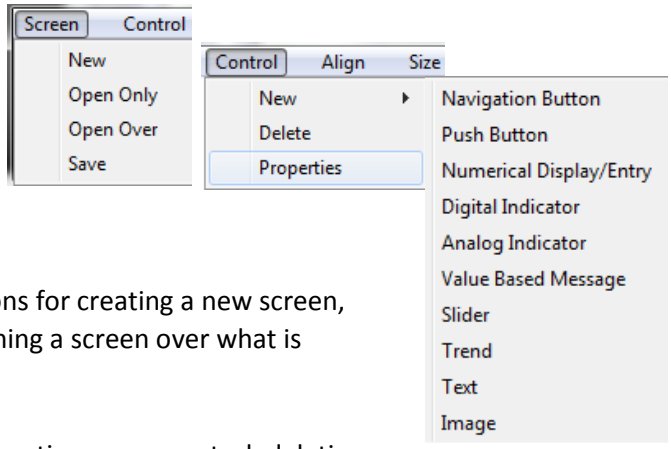
A push button is an *active* graphic control. It will respond to the action of the mouse. A click of the button will change the value of the tag assigned to the push button control. In the same time, if the tag was changed in other parts of the program, the button state can change to reflect the real value of the tag.

A numerical display is *passive*. It can show the value of a tag. A numerical entry is *active*. It can change the value of a tag by double clicking and using the keyboard to dial in a new number. The difference is that a numerical display/entry shows a number, not a graphical object.

9.4) Screens windows and screen navigation

In order to create or modify screens, *FeSCADA* is using a dedicated window called Edit window. To open the screen editor select from the main menu: **Utils -> Screens Edit**.

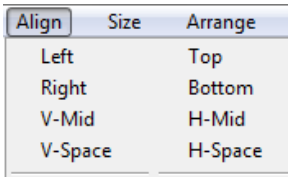




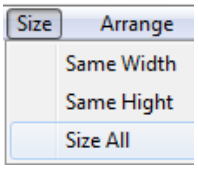
The Edit window has the following main menu:

- **Screen** – will open a submenu with options for creating a new screen, opening a screen in empty window, opening a screen over what is already in the window, saving a screen.

- **Control** – shows the submenu used for creating a new control, deleting a control and opening the dialog setup for the properties of a control. The *New* option will show the controls that can be created: Navigation Button, Push Button, etc. (see picture on the right).

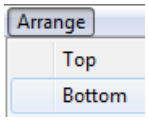


- **Align** – shows options for aligning and positioning controls based on a reference control or among a selected group of controls.

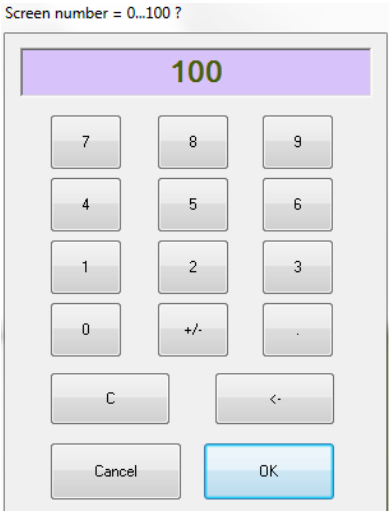


- **Size** – is a tool for making a control or a group of selected controls have the same size with a reference control.

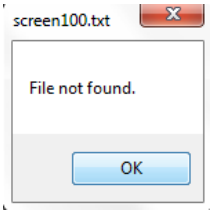
- **Arrange** – a tool for Z order, for positioning a control window in front or in the back of another control window.



Every screen is assigned with a number. Screen 0 (zero) is the first screen that the application will open. When the user is opening a screen the edit window will prompt him to type a screen number between 0 and 100. If the screen does not exist a message box will inform that the file "screen*X*.txt" was not found, where *X* is the screen number.

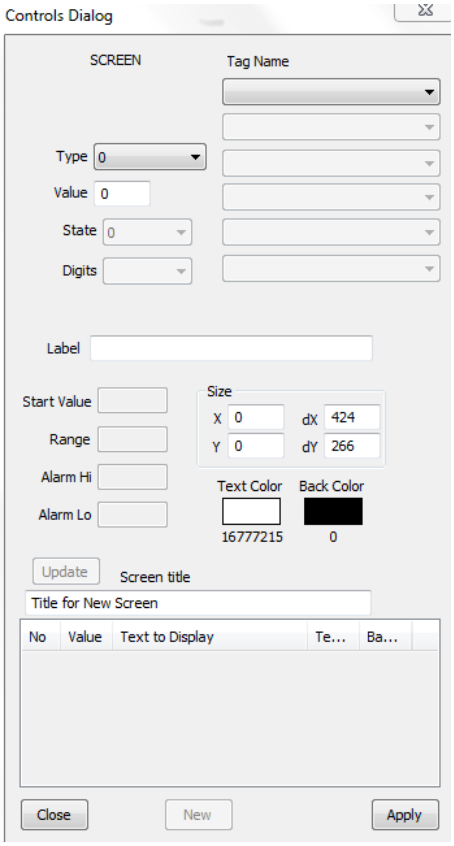


When the user is saving a screen the program will prompt him to type a number.

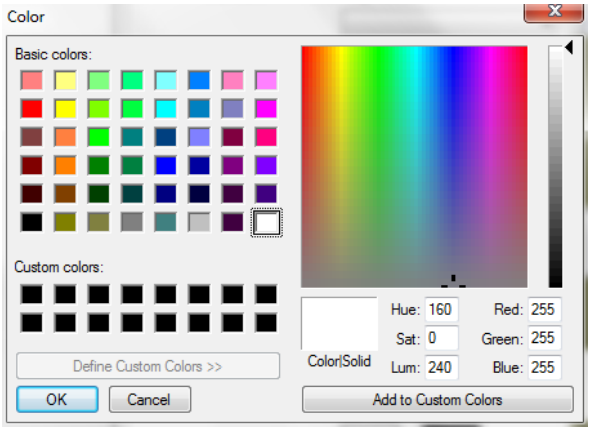


Selecting menu **Screen->New** will clean the edit window of any controls that were inserted before and will show a black window with a white title: "*Title for New Screen*".

The background color and the title's text and color can be changed. For this select **Control->Properties**. The window called **Controls Dialog** will open. This dialog window can stay open during the entire time of editing a screen. Whenever the user is clicking (selecting) a control (or the screen window) the properties of that control (or screen) are loaded into the settings of the control dialog. If the user is changing the settings and is pressing **Apply**, the new settings are sent to the control (or screen).



If the user clicks on any point of the screen the control dialog will show SCREEN on top, white for **Text Color**, black for the **Back Color** and "*Title for New Screen*" in the **Screen title** text box. By clicking the white rectangle in **Text Color** setting, the user will open the color palet window where another color can be selected.



9.5) Controls creation and properties, tags assignment

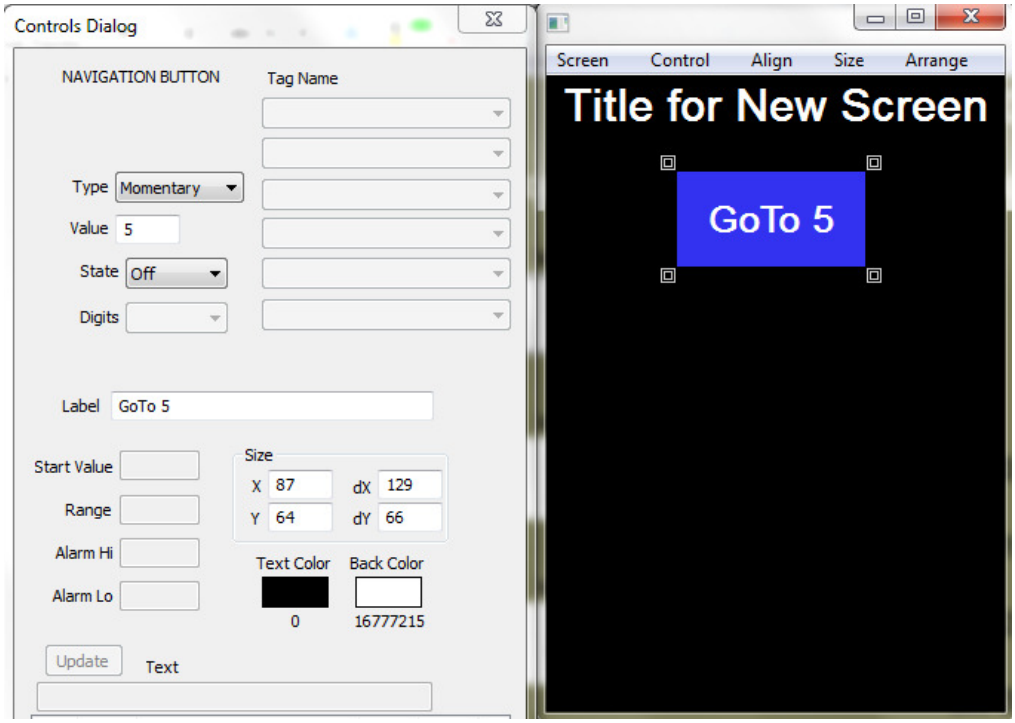
To add a control (child window) in an edited screen, the user has to select **Control->New->…** from the Edit window menu. There are 10 different controls. For any of them, the **New** button on the **Controls Dialog** window will become enabled. With the mouse, left click and drag to select an area on the screen. Pressing the **New** button in the **Controls Dialog** will create the new control in the area selected with the mouse. After this, the user has to select the control with the mouse (click on it) to retrieve its properties in the **Controls Dialog**.

This is the general method to insert a new control, other than the Copy/Paste, which will be discussed later in this chapter.

In the next part of this chapter, examples of inserting all of the 10 controls will be given and their properties will be discussed.

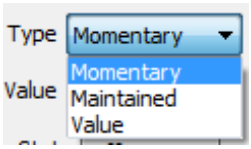
9.5.1) Navigation buttons

The **Navigation Button** is used for opening another screen, or changing the focus to that screen, if the screen is already opened. The only properties that are important for a **Navigation Button** are the **Label**, which is the text that the user wants to show on the button, and the **Value**. The **Value** will indicate which screen will be opened. After changing any settings press **Apply** to send them to the control.

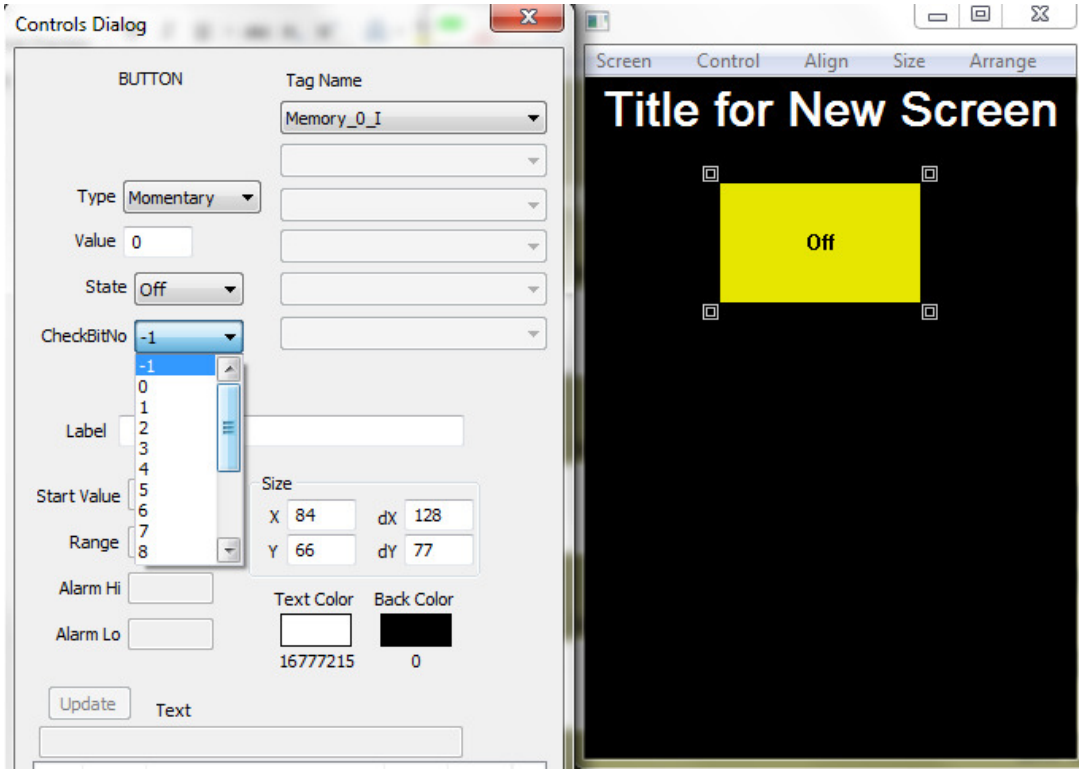


9.5.2) Push buttons

To insert a new Push Button select **Control->New->Pushbutton** on the menu, select an area with the mouse and press **New** button. After selecting the **Push Button**, the **Controls Dialog** will display its default settings. There are three types of **Push Buttons**: *Momentary*, *Maintained* and *Value*. The default is *Momentary*, which will turn the button on as long as the mouse is pressing it and will turn it off when the mouse click stops. For *Maintained*, the button will switch on and off at two consecutive clicks of the mouse. *Value* is used to have a sequence of states that are more than 2. In this case, at any mouse click the number will increase by one, up to the number set by the user, and after that will restart from zero. The number is set in the property field **Value** of the **Controls Dialog**.

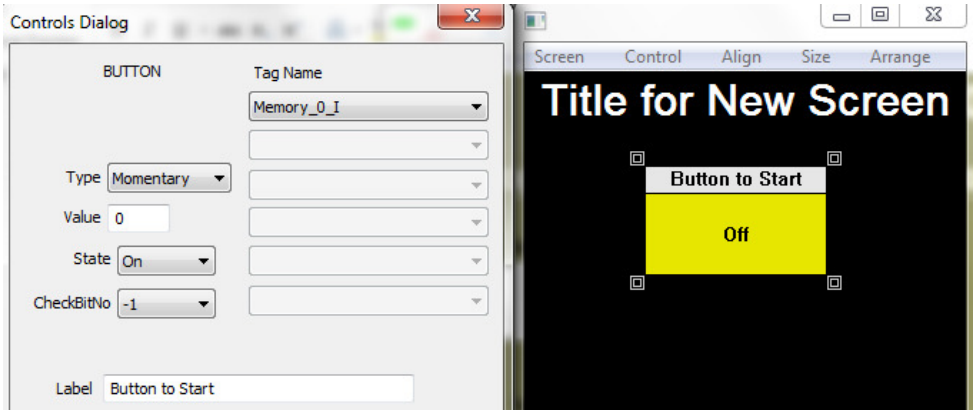


The next important property of a **Push Button** is the **Tag Name** assignment. By clicking the **Tag Name** combo box, the user will find the list with all the application's tags. In the example below, the tag *Memory_0_I* was selected.



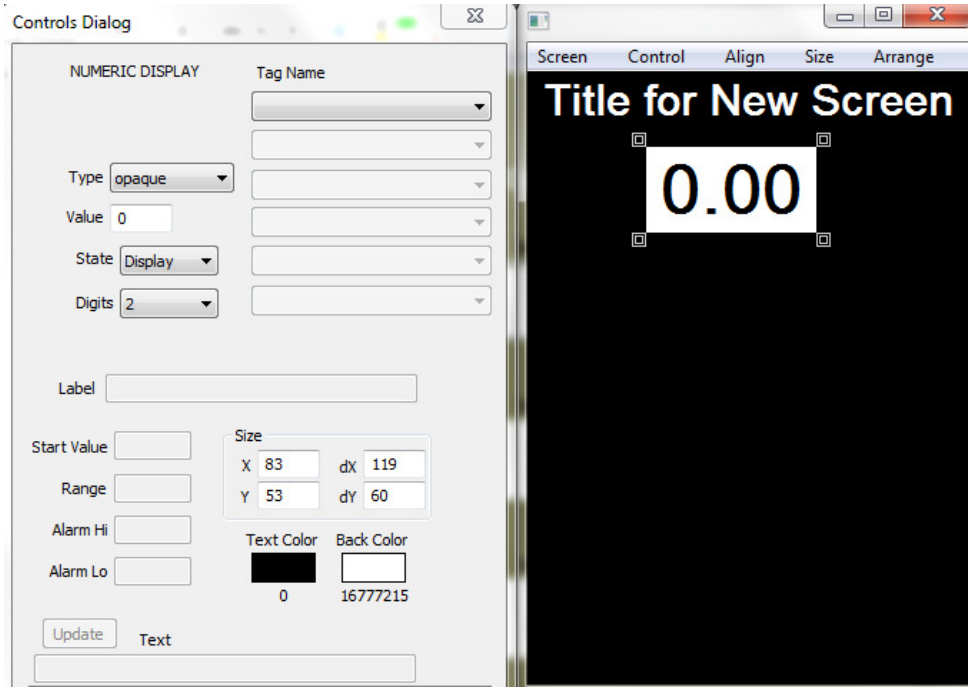
If the tag is an integer, the user has the option to change its value as a whole number or just for a bit number of the integer. The bit number can be selected from **CheckBitNo** combo box. If the selection is -1 then the tag will be changed as a whole integer. For changing a bit number, select a number from 0 to 15.

If the **Label** text box is empty the button appears as in the picture above. If a text is set in the **Label** edit box, it will appear on top of the button, as in the picture bellow. Press **Apply** after changing any setting.

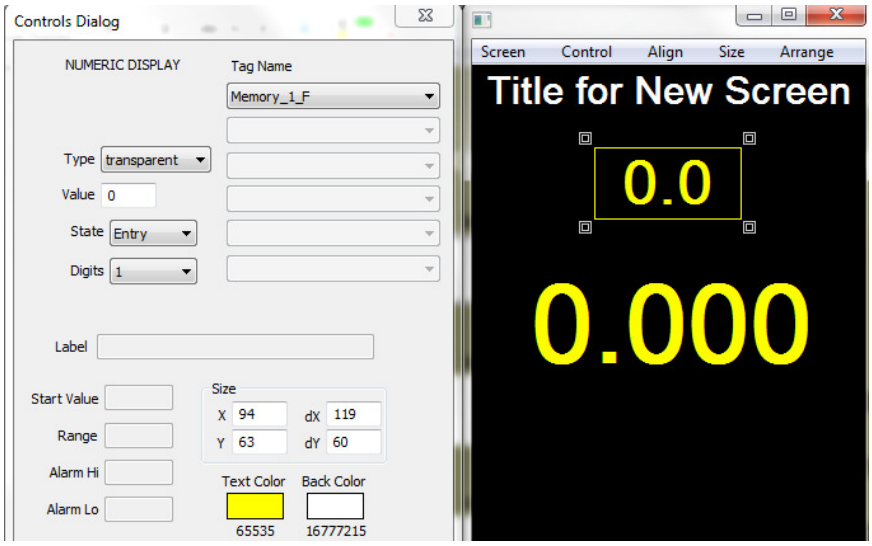


9.5.3) Numerical display/entry

To insert a new numerical display select **Control->New->Numerical Display/Entry**. Select an area with the mouse and press **New**. Click on the new control to retrieve its default properties. The new numerical display can be *opaque* or *transparent* (against the background of the screen or an image). It is represented as a real number with 2 digits, for arithmetic precision, and has no tag assigned in **Tag Name** combo box. From the **State** combo box the user can select *Display* or *Entry*. Entry means that by double clicking the control the user can write a new value. This is true only when the application is running the screen in normal mode (not edit mode). Press **Apply** after changing any setting.

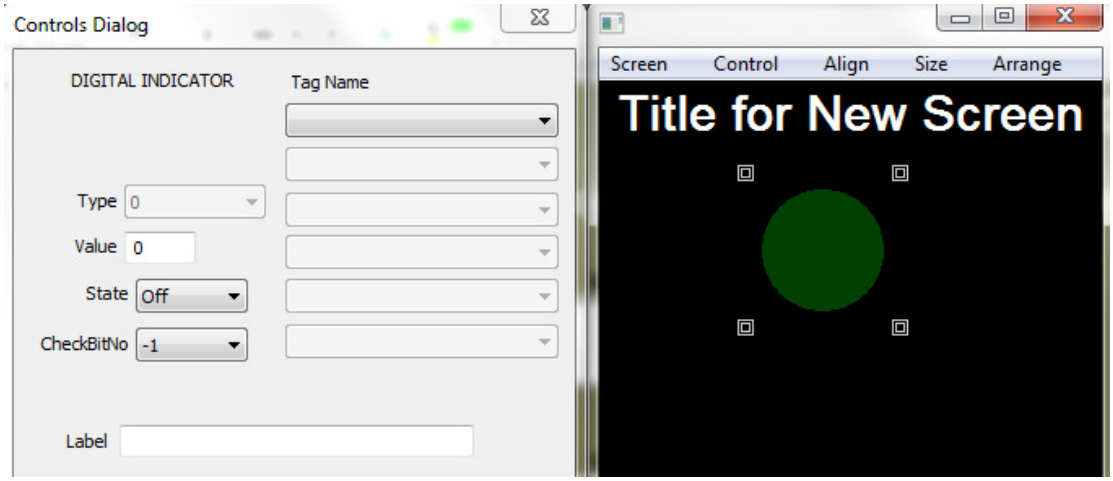


The size of the text for displaying the number is changing automatically, based on the height of the control window. It is 80% of the height. In this mode the only adjustment needed from the user is the width, to show all the digits that the number can have.

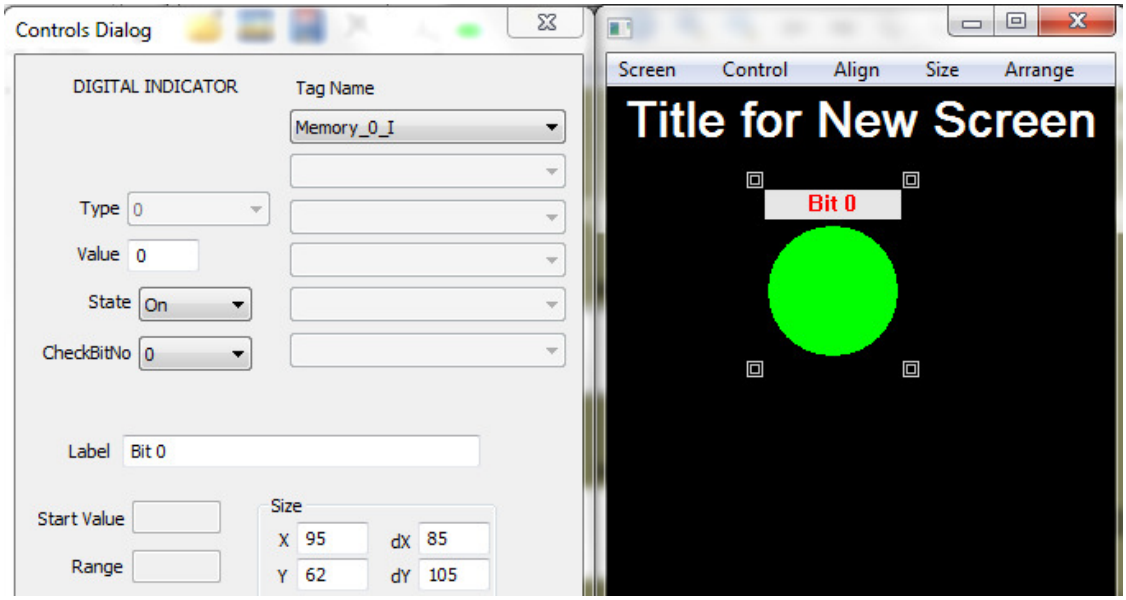


9.5.4) Digital indicator

A digital indicator can be inserted by selecting **Control->New->Digital Indicator**. Select an area with the mouse and press **New**. Click the new control to retrieve its properties in the **Controls Dialog**. Press **Apply** if a setting is changed.



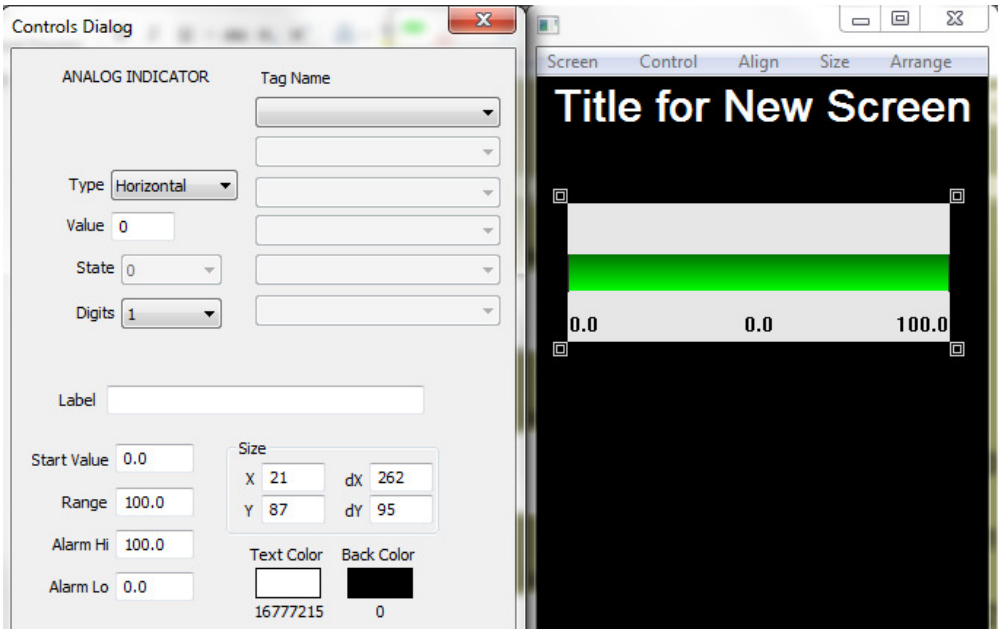
A Digital Indicator can be set for the whole tag's value of for a bit number from the tag's integer value. If **CheckBitNo** combo box is -1 then the tag is checked as a whole number. The indicator will show *Off* if the value of the tag is zero and *On* if different than zero. If the **CheckBitNo** is set to a value from 0 to 15, the indicator will show *On* if that bit number is true (=1) and *Off* if it is false (=0).



If the user is writing a text in the **Label** text box, after pressing **Apply** the text will appear above the indicator. The text is red when the indicator is *On* and black when is *Off*.

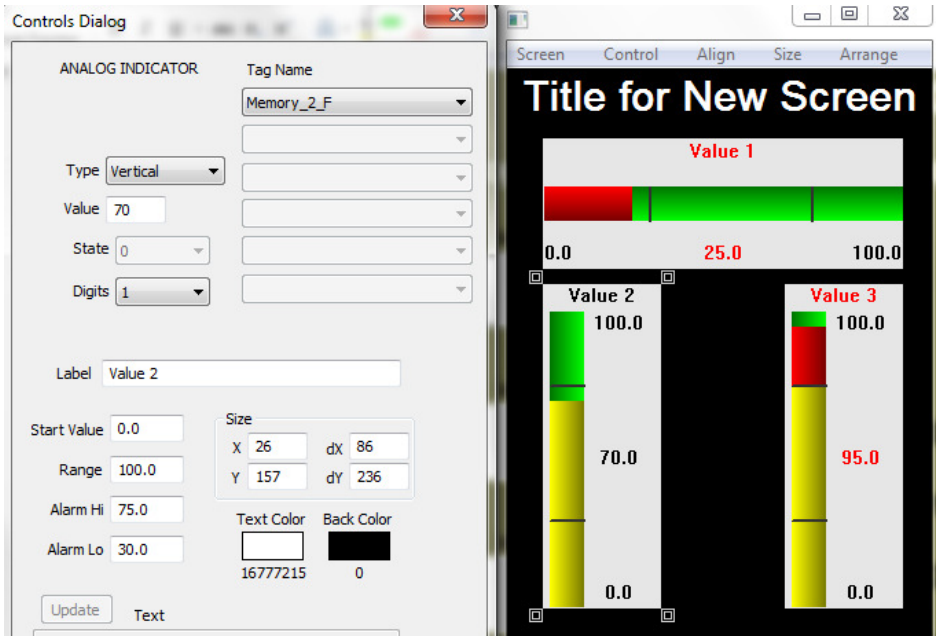
9.5.5) Analog indicator

To insert an analog indicator, select **Control->New->Analog Indicator**. Select an area with the mouse and press **New**. Click the new control to retrieve its properties in the **Controls Dialog**. Press **Apply** if a setting is changed.



An analog indicator can be *Horizontal* or *Vertical*. It has a **Start Value**, a **Range** value, an **Alarm Hi** value, and an **Alarm Lo** value. The user can change the precision of the numerical values from the **Digits** combo box. The **Label** text will appear on top of the control.

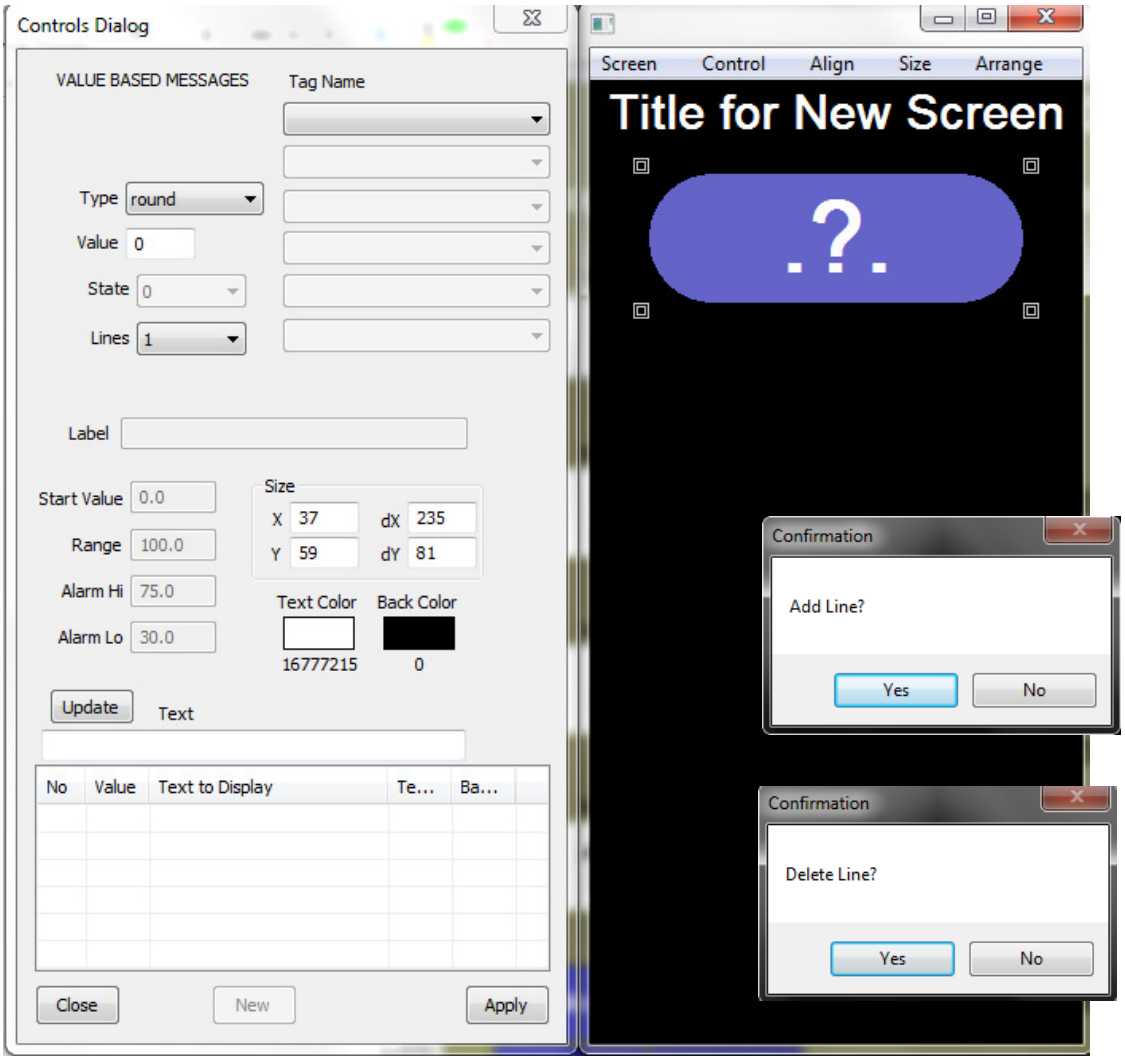
If the value of the tag is lower than the **Alarm Lo** value, the indicator shows red, if it is in between **Alarm Lo** and **Alarm Hi** values, the indicator shows yellow. If the tag's value is above the **Alarm Hi** value, what is under the **Alarm Hi** is yellow and what is above **Alarm Hi** is red.



9.5.6) Value based messages

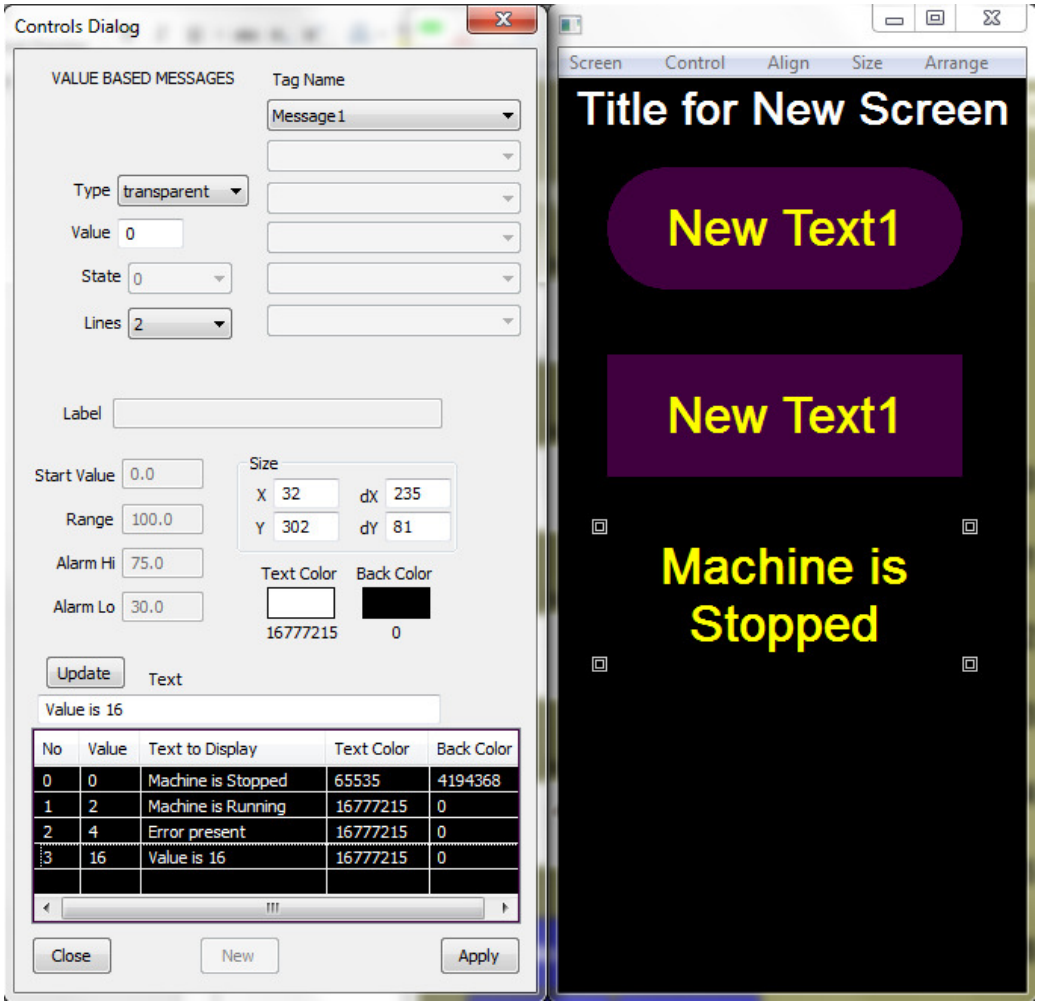
To insert a value based message, select **Control->New->Value Based Message**. Select an area with the mouse and press **New**. Click the new control to retrieve its properties in the **Controls Dialog**. Press **Apply** if a setting is changed.

A value based message is a text message that is changing based on the tag's value. The texts and the values have to be defined by the user. Whenever the tag's value is equal with one of the values from the list, the text related with that number is selected and displayed in the control window. If the number is not defined the ".?." text is displayed. By default, the control type is *round* (to the left and right sides) and with the text size of *1* line per control height (dY). The type can be *square*, *round* or *transparent* (against the background and other images). The user can select *1*, *2*, *3*, or *4* lines per control height.



To insert a pair (**Value**, **Text to Display**) right click an empty line on the list at the bottom of **Controls Dialog** and select **Yes**. To delete an existing line, right click on the line and select **Yes**. The user can change the colors of the text and the color of the background of the text. Double click the cells in the list to edit the value and the text. Press Enter to finish and **Apply** to save.

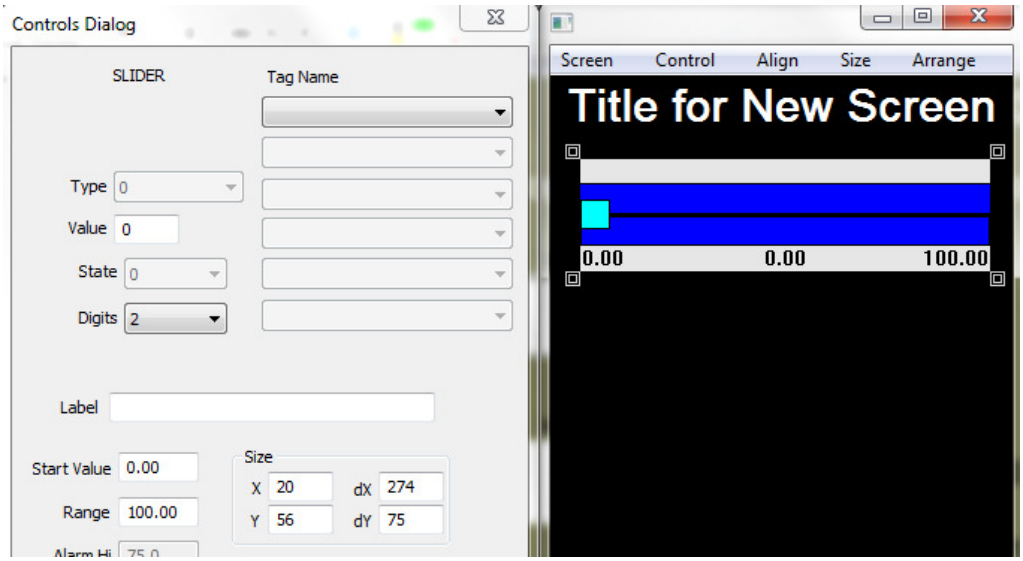
In the picture bellow, the three **Type** options are showing: *round*, *square* and *transparent*. The selected control is the transparent one, at the bottom. The text size is *2* lines per control height. Tag *Message1* was selected in **Tag Name** combo box.



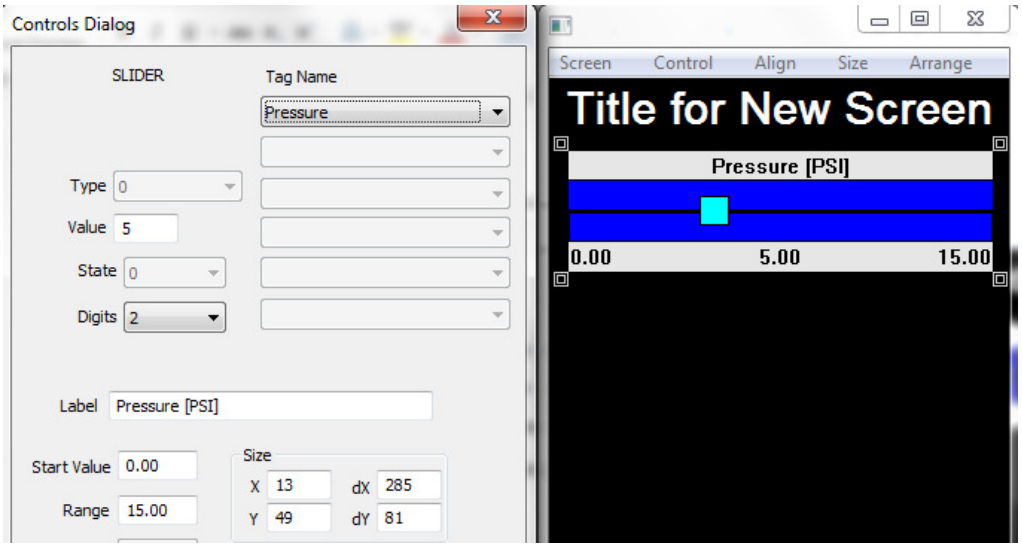
9.5.7) Sliders

A slider is an entry control used to change the value of a tag with the aid of the mouse. To insert a slider, select **Control->New->Slider**. Select an area with the mouse and press **New**. Click the new control to retrieve its properties in the **Controls Dialog**. Press **Apply** if a setting is changed.

A slider has a **Start Value** and a **Range** value that have to be defined by the user. The change of the value can be done in two ways: left click the slider in any part of its window and drag the slider's cursor left or right. Release the mouse when the value is what you want. The second way is accomplished by right clicking the slider. In this case a numerical input pad will appear and the user can dial in a very precise number. Both ways will restrict the user to select a number that is less than the **Start Value** or bigger than the **Start Value** + **Range**. Both ways are operational in the normal application run (not edit mode).



As for the other controls, the user can select the number of **Digits** for the arithmetic precision and a **Label** text to be displayed on the top of the control window.



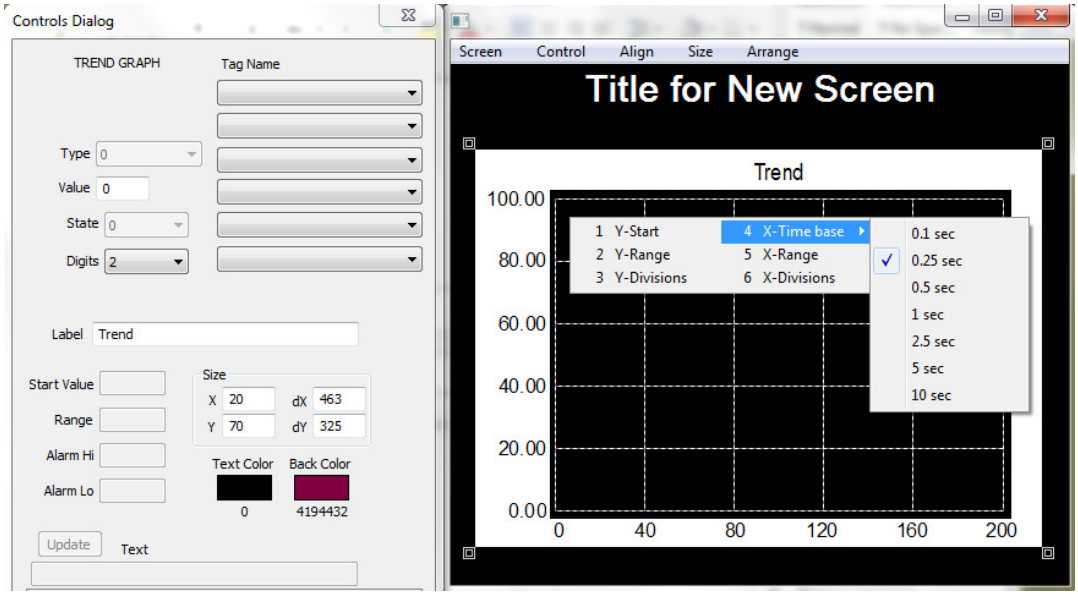
9.5.8) Trends

A trend is a time chart where the values of up to 6 tags can be visualized about how they evolve in time. To insert a trend, select **Control->New->Trend**. Select an area with the mouse and press **New**. Click the new control to retrieve its properties in the **Controls Dialog**. Press **Apply** if a setting is changed.

A trend has an X axis and a Y axis. The Y axis is representing the values of the tags. The X axis is representing the time. To set the settings for the X and Y axes right click on the trend window. A floating menu appears with options for selection of the following settings:

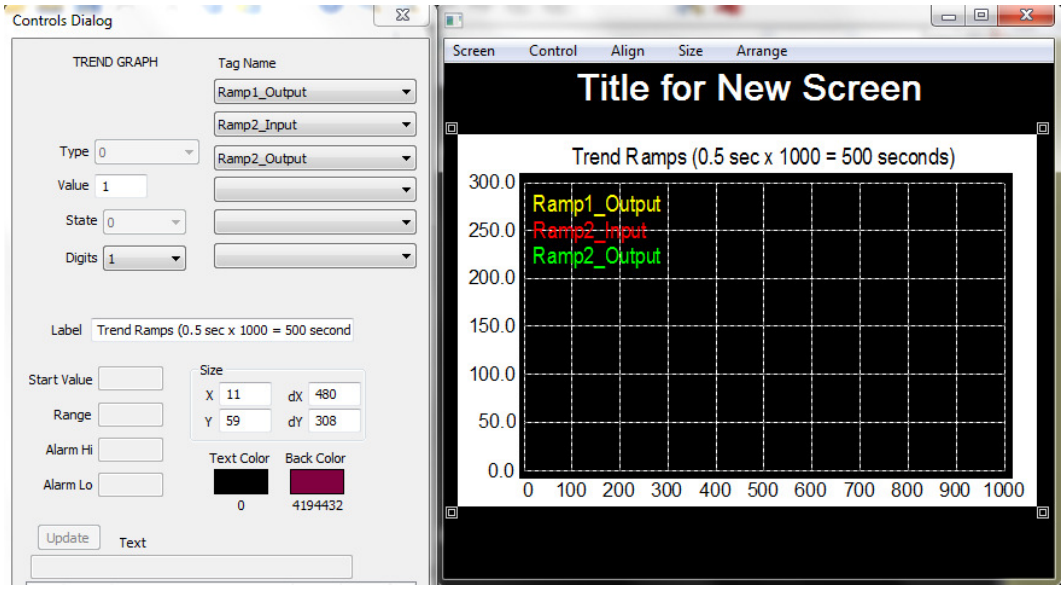
- **Y–Start** value, **Y–Range** value and **Y–Divisions** number of horizontal intervals for the graph. In the example below the Y-Start value is 0 (zero), Y-Range value is 100 and Y-Divisions value is 5.

- **X–Time base** value, **X–Range** value and **X–Divisions** number for the graph. The **X–Time base** has different time cycle values: 0.1, 0.25, 0.5, 1, 2.5, 5, 10 seconds. The tag values will be read only at the selected time interval. The **X–Range** is setting the number of time cycles that the trend can display. The **X–Divisions** is for the number of vertical intervals inside the graph. In the example below the cycle time is 0.25 seconds, the range is 200 (200 x 0.25 = 50 seconds), and there are 5 divisions.



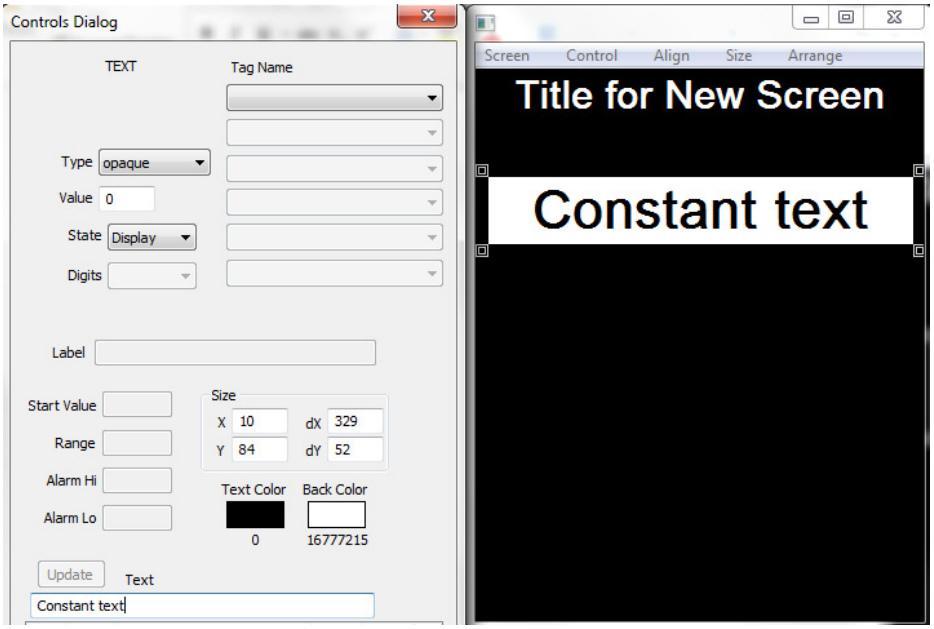
Label is the title that will appear on the top side of the trend window. **Digits** is the precision of the numbers representation on the Y axis. All the six **Tag Name** combo boxes are enabled and the user can select tags from the application's tags in any one or in all of them. Every combo box has a predefined color. The name of the tags will appear in the assigned color in the top-left aria of the graph.

The trend below is set to represent three tags with a Y-Start to Y-Range of 0 to 300, with an X-Range of 1000 and an X-Time base of 0.5 seconds for the time cycle (1000 x 0.5 = 500 seconds).



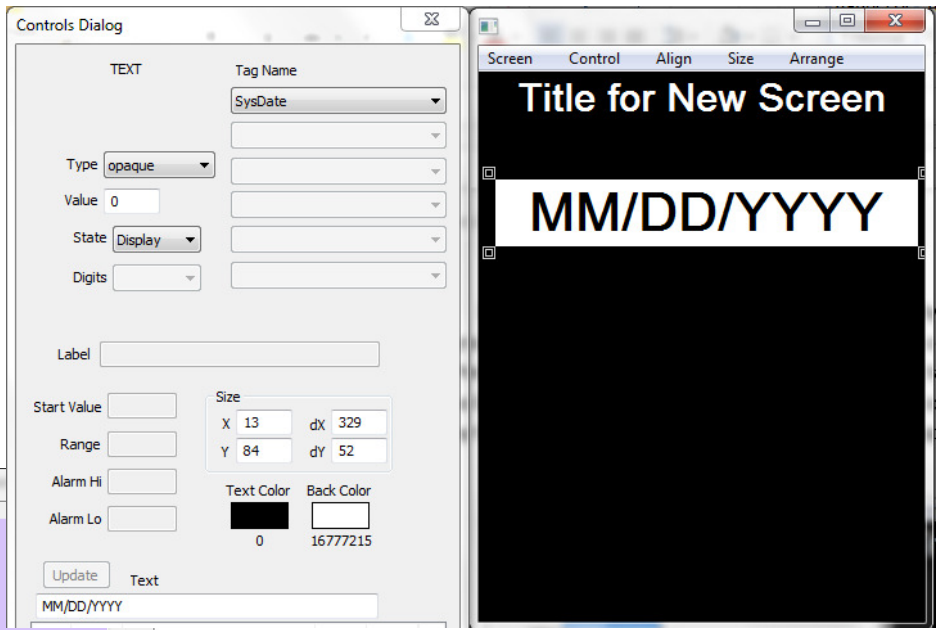
9.5.9) Text

A text is a constant string or a tag string that can be displayed in a control window with a predefined size and color. To insert a text, select **Control->New->Text**. Select an area with the mouse and press **New**. Click the new control to retrieve its properties in the **Controls Dialog**. Press **Apply** if a setting is changed.

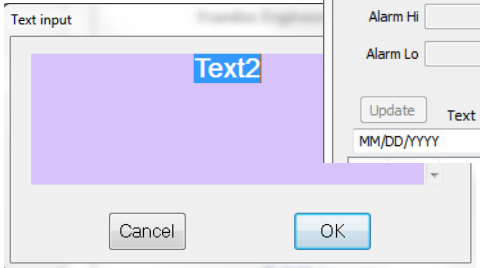


By default the text control will be inserted empty (with no text). It is of type *opaque*, with a white background color and black text color. The text, if it is a constant string, has to be typed in the **Text** edit box. Press **Apply** to see the results.

If the text is a string tag, the tag has to be selected in **Tag Name** combo box. Even if it is a string tag assigned to the text control, the user can still check the size of the text by typing something in **Text** edit box and clicking **Apply**. In the picture below, the tag *SysDate* is of the form "MM/DD/YYYY". This text was written in **Text** edit box to check if the whole string is going to appear in the text control window.

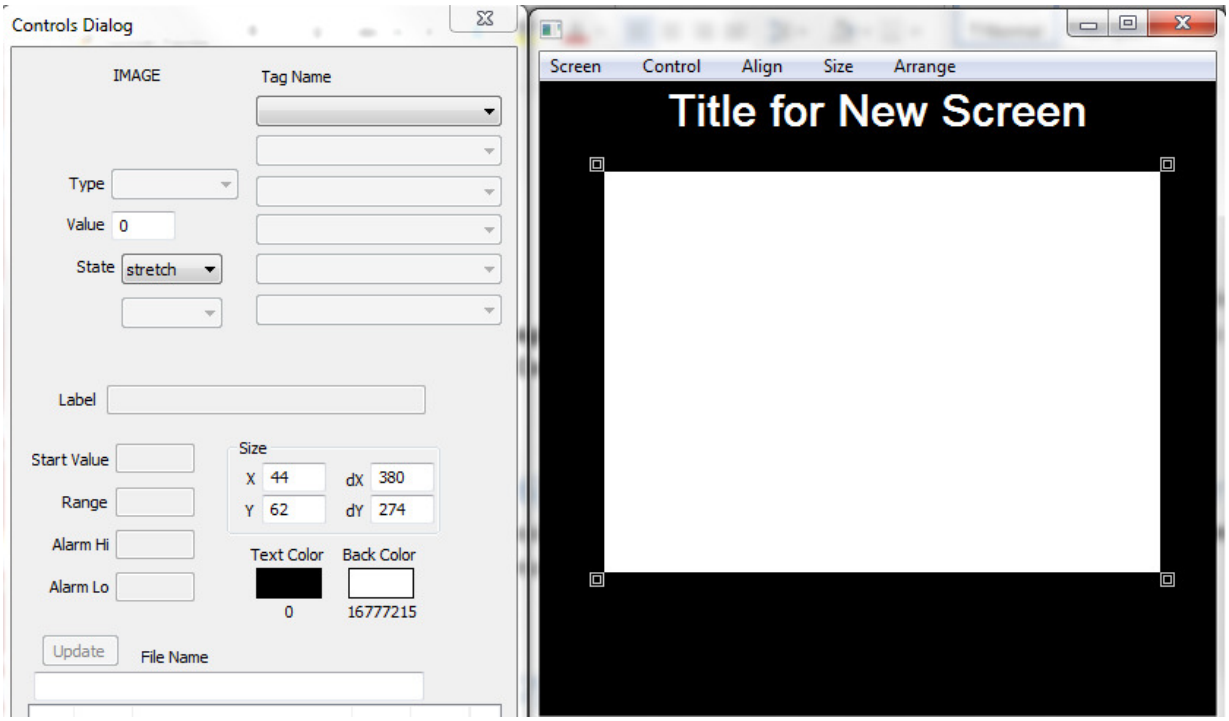


For a string tag, the **State** combo box can be selected to *Entry*. If the text control is an *Entry*, the user can double click the control and open a text input window. Type the new text and press **OK**. This is operational in normal application run (not in edit mode).



9.5.10) Images

A bitmap image can be inserted in a screen window by choosing **Control->New->Image**. Select an area with the mouse and press **New**. Click the new control to retrieve its properties in the **Controls Dialog**. Press **Apply** if a setting is changed. By default a white square is showing in the image area. The color can be changed with the **Back Color** setting.



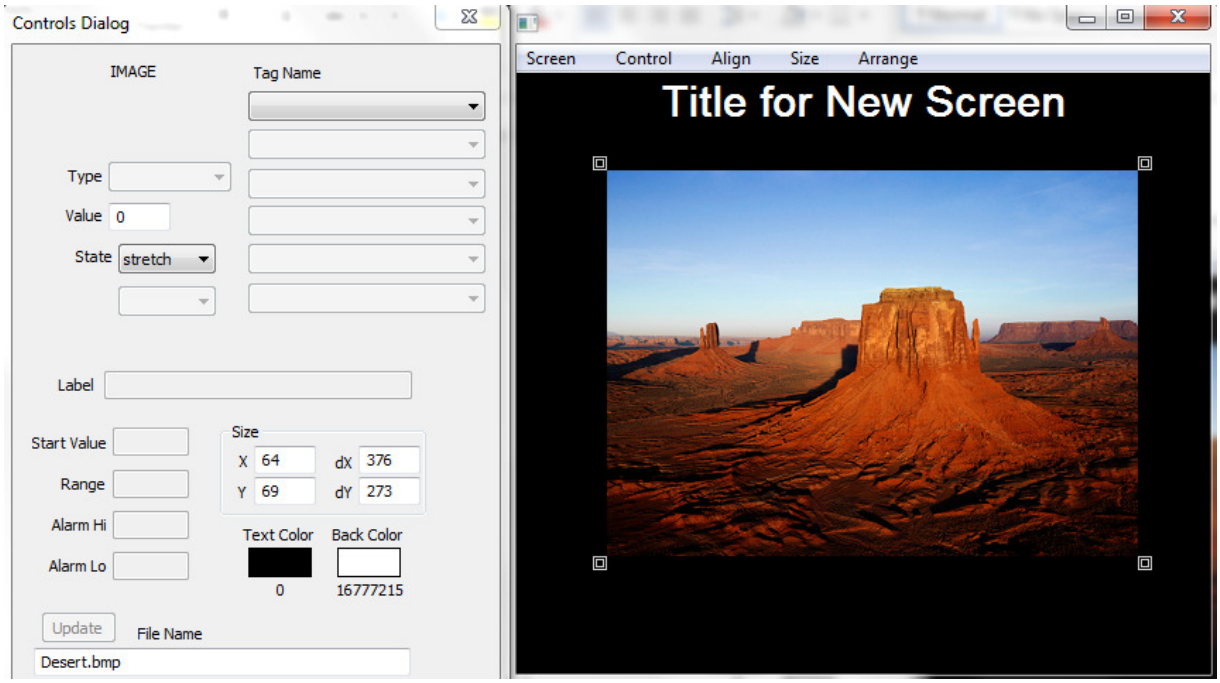
To select a bitmap image, the user has to do three things:

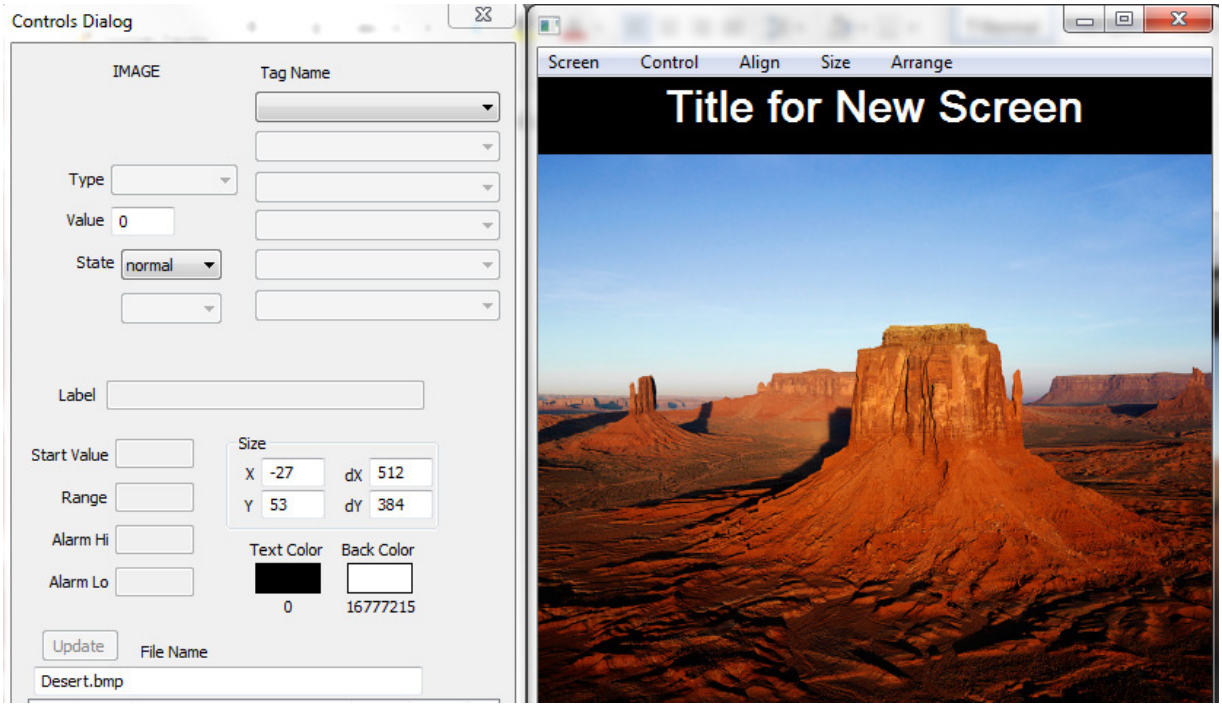
a) Create a folder "IMG" in the same folder in which *FeSCADA* program is running;
b) Copy the bitmap picture in the "IMG" folder (all images should reside here);
c) Type the name of the bitmap file in **File Name** text box, in the form "*FileName.bmp*" and press **Apply**.

The image display can be *stretched* or *normal*. The option is selectable in **State** combo box. In *stretch* mode the user can resize the image by clicking and dragging the little corner rectangles. The *normal* option will resize the image to its original size. The user cannot modify the size of an image that is displayed as *normal*.

A tag can be assigned to an image. If an image does not have a tag assigned, the image will always be shown. An assigned tag has the hide/show effect. Whenever the tag is true the image is shown. When the tag is false (zero) the image is hidden. In this way the user can make animation like visualizations.

In the pictures below a *stretched* and a *normal* size view are selected for the same bitmap image called “Desert.bmp”, in the folder “IMG”.

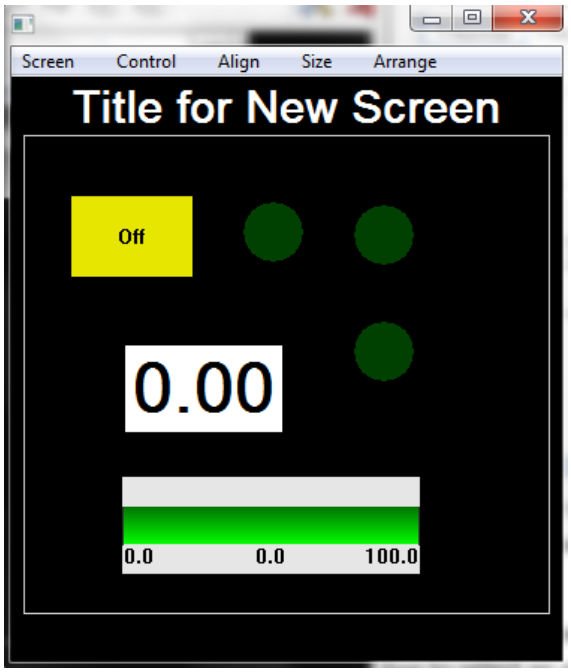




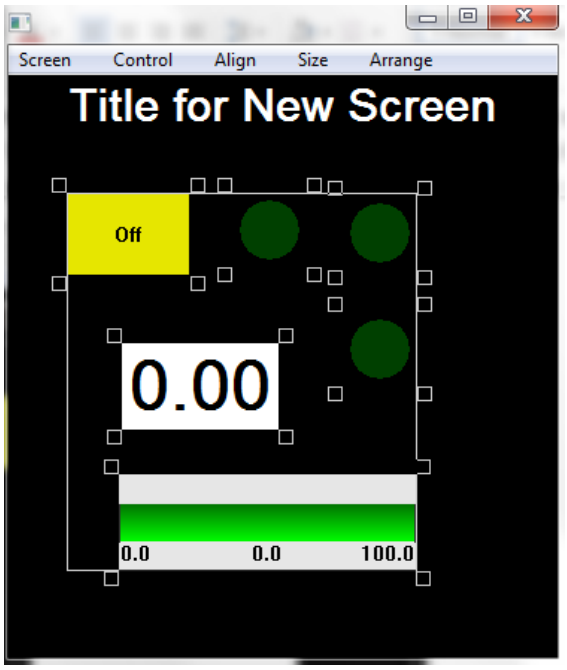
9.6) Single and multiple controls selection: alignment, size, arrange

When a control is clicked with the left mouse button and the 4 little double lined rectangles appear at the control's corners, the control is said to be selected as *single selection*. If the **Controls Dialog** window is open, the control's properties are retrieved and shown.

A *multiple selection* is happening if the user is clicking somewhere on the screen and while is keeping the left button pressed is dragging the mouse to select an area. When the left button is released, the program will check if there are any window controls inside the selection box. If there are, they will be put in a list of selected controls. This selection can be moved by clicking on one of the selected controls and dragging it to a new position. The whole grup of selected controls will move in the same time.

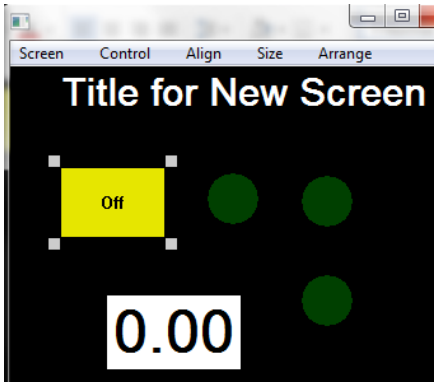


The Selection Box



Multiple Selection Result

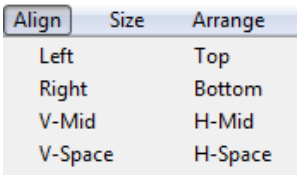
Another way to realize a *multiple selection* is to keep pressed the **Ctrl** key in the moment the user is clicking on a control. If the control was not part of the multiple selection, it will be added to the multiple selection list. If the control was already in the multiple selection list and the user is clicking again on it while the **Ctrl** key is pressed, the control will be removed from the multiple selection list.



In a multiple selection, the little rectangles at the corners of the controls have a single line. In a single selection, the little rectangles are drawn with a double line.

A *reference* control is a control on which the user is clicking with the right mouse button. The little 4 rectangles at the control's corners have a filled appeareance. A *reference* control is needed whenever the user is editing with **Align** or **Size** menu options.

In the purpose of speeding up the process of screens editing, the alignment and sizing tools are helping the user to position the controls in an ordered and elegant fashion. Most of the alignment and sizing options are done in regard to the *reference* control. The reference control will not move and will not change the size, but the *single selection* or *multiple selection* will move, or change size, to match the reference control.



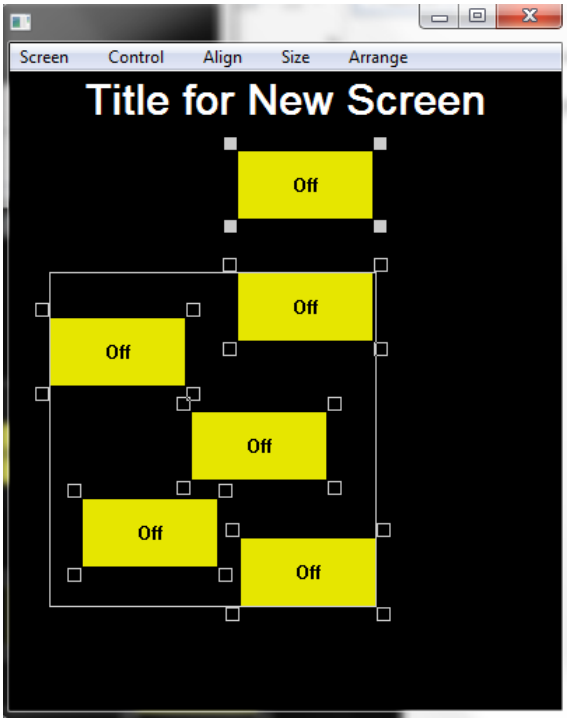
The **Align** sub-menu has the following options:

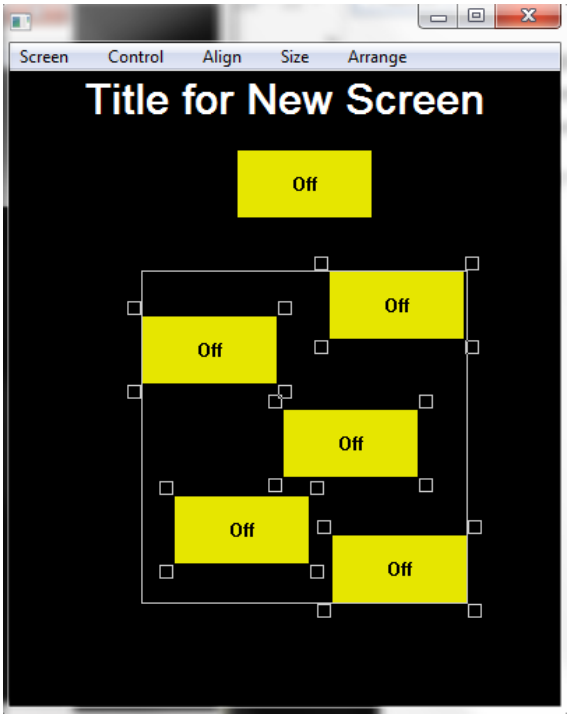
- *Left*, *Right*, *Top*, *Bottom* (of the reference control),
- *V-Mid* (vertical middle of the reference control),
- *H-Mid* (horizontal middle of the reference control),
- *V-Space* (make equal vertical spaces between controls - in a multiple selection only),
- *H-Space* (make equal horizontal spaces between controls - in a multiple selection only).

There are three aligning situations to a reference control.

1) If *single selection* the alignment is applied only to the selected control
2) If *multiple selection* and the reference control ***is not*** part of the multiple selection then the whole selected group will move to align with the reference control. In this case the controls within the group will maintain their relative position in regard to eachother.
3) If *multiple selection* and the reference control ***is*** part of the multiple selection then every control from the selection list will move independently of the others to align to the reference control.

In the pictures below there is an example of a multiple selection alignment to an external reference control to the *V-Mid* (vertical middle).

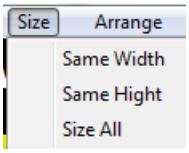




The whole group of multiple selection controls moved to align with the external reference control. The relative positions between the controls of the multiple selection did not change.

In the pictures below there is an example of a multiple selection alignment to a reference control that is part of the multiple selection list. The alignment option is the same as above, *V-Mid* (vertical middle). The last command was *V-Space*, to set equal vertical spaces between controls.

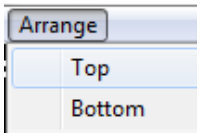
Sizing to a reference control is used to make the selected controls to have the same height, width, or both, equal with the reference control's size. In this case there is no difference between a single selection and a multiple selection.



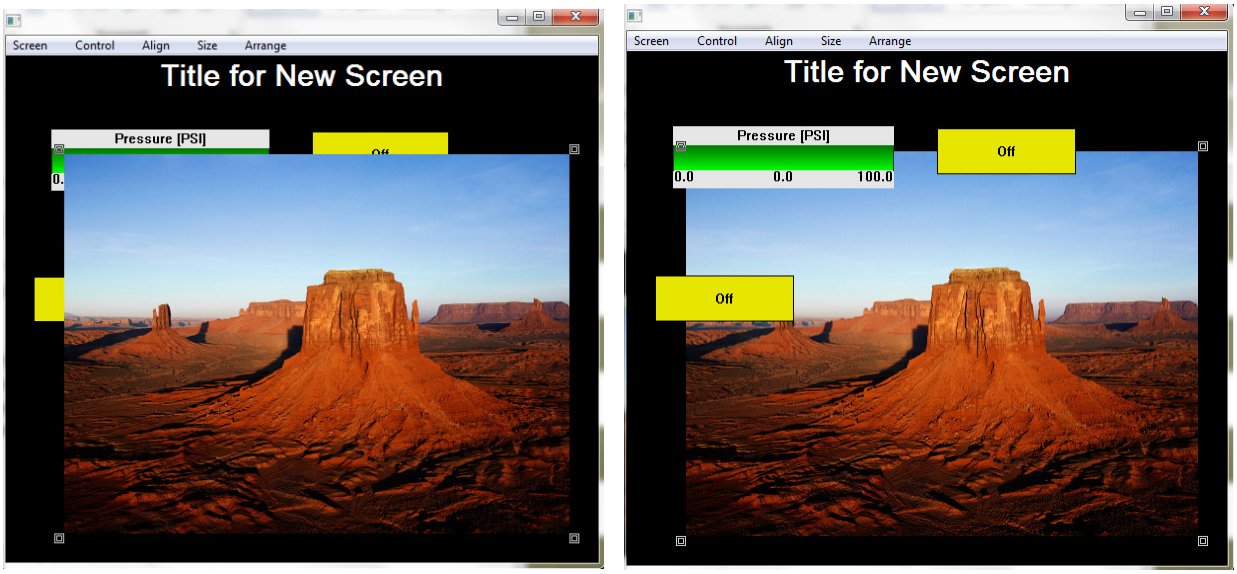
Either internal reference or external reference control the same rezult, for *Size All*.

To be able to control the Z order of multiple windows that are overlapping the user can use the **Arrange** menu options.

- *Top* – Brings the control's window in front of all other windows,
- *Bottom* – Sends the control's window behind all other windows.



Here is an example on how to put the image control in the back of all the other controls (*Bottom*).



9.7) Copy/Paste options

For single selection, press ***Ctrl+C***, put the mouse cursor where you want the new control, and press ***Ctrl+V***. The top left corner of the new control will be located at the position where the mouse cursor was when ***Ctrl+V*** was pressed. The copy will have the same sizes and the same properties as the original control. You can press ***Ctrl+V*** more than one time to paste other copies.

For multiple selection, after the selection is active, put the mouse cursor where you want the new controls and directly press ***Ctrl+V***. You do not need to press ***Ctrl+C*** when a multiple selection is active. Apart from the new position, all the new controls will have the properties of their counterparts from the original multiple selection.

To delete controls, press the ***Delete*** key after either single or multiple selection. For a single selection, the second option is to select menu **Controls->Delete**.

9.8) Transparency limitations: screen background and image controls

Some controls have the property of transparency. This is true only if the control is above the screen's background or above an image.

If two images are overlapping, the one that is on top will be completely visible, the one at the bottom will be visible only partially.

If a transparent control is in front of any other control but image, the transparency is not true, because the other control, beneath the transparent control, will not be visible.

9.9) Application development – screens and control number limits

There is no program limit for the number of the controls that can be put on one screen. Only the computer and the Windows OS limitations. For example, the GDI objects that Windows allows for an application is 10,000. If more that 10K are needed, the program stops.

The limit to 100 screens is made just for the reasonable thought that there are no such small and medium applications that will need more than maximum a dozen of windows.

9.10) Conclusions

This chapter has introduced all the controls and all the edit tools that can be used to develop new screens for visualization of a machine/process with *FeSCADA*.

Only one instance of *FeSCADA* software can run at a given time, from the same folder, but more projects can be developed and run on one computer, in the same time, in different folders. If more projects are needed, create a new folder for each application and copy the *FeSCADA.exe* and *FeSCADA.lic* files in it.