

Part III

Training 3

6 HMI

Visualization is a graphical representation of the project variables which allows influence the PLC program in online mode via mouse and keypad. The CoDeSys visualization editor, which is part of the programming system provides graphic elements which can be arranged as desired and can be connected with project variables.

Visualization which is created in the programming system in many cases will be used as the only user interface available for controlling and watching the associated PLC program in online mode. For this purpose it must be possible to give inputs to the program solely by activating visualization elements.

Any visualization created in CoDeSys can later be used in different ways:

- It can be made available on CoDeSys HMI, a special runtime system for operating the visualization in full screen mode on a PLC computer.
- It can be made available as a Web-Visualization, which allows to call and operate it via the Internet (useful for remote maintenance purposes).
- It can be made available as a Target-Visualization, which can be started directly on the PLC.

When defining the name of the visualization object, please regard the following:

- A visualization named "PLC_VISU" per default automatically will be used as start visualization in a Target- or Web-Visualization resp. In CoDeSys HMI, if there not explicitly another visualization is configured for this.
- A visualization may not get the same name as another object within the project because this would result in problems when changing between visualizations.

For your first visualization select **Visualizations** tab - Right button click on Visualizations folder - **Add Object**. In New Visualization window type "PLC_VISU" (see Fig. 1).

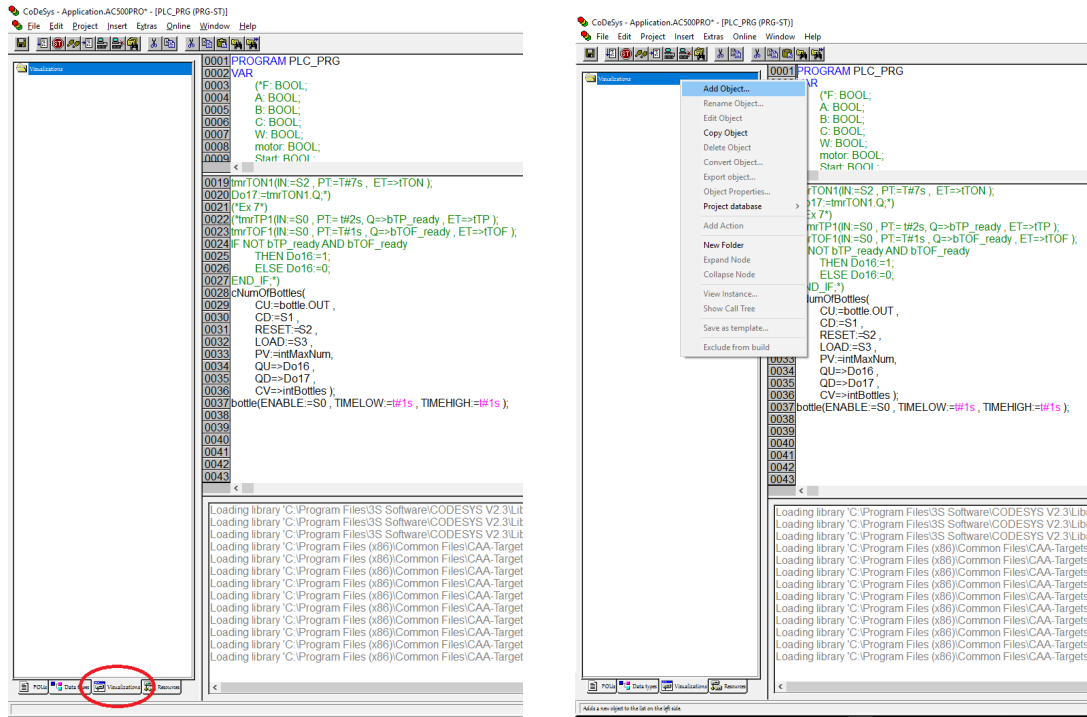


Figure 1: New Visualization creation

6.1 Visualization Elements

A visualization element is a graphical element, which is used to fill a visualization object. The available elements are offered in the CoDeSys menu bar. Each element gets a separate configuration.

Go to the 'Insert' menu item and select freely from the following commands (see Fig. 2): 'Rectangle', 'Polygon', 'Polyline', 'Curve', 'Pie', 'Bitmap', 'Visualization', 'Button', 'Table', 'Meter', 'Bar Display', 'Histogram', 'Alarm table', 'Trend', 'WMF file', etc.

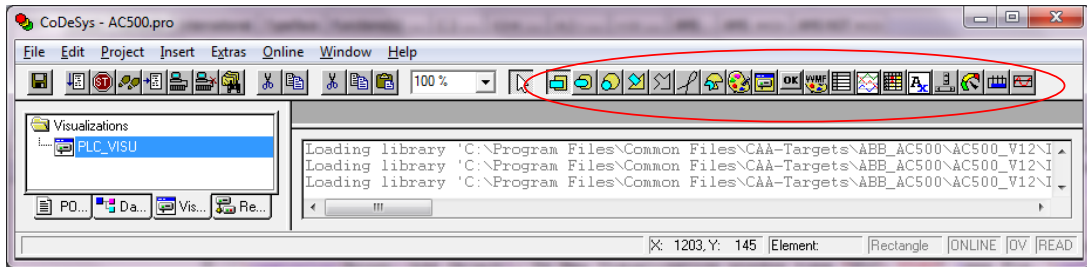


Figure 2: Elements

Select 'Rectangle' from the menu and then using arrow set the area where text should be placed. Double clicking on the object opens the window Regular Element Configuration (#N). In this window you can set different parameters of the object.

- In the **Text** Category type the Application Name: "Bottles packing line". Select suitable font and its size (see Fig. 3).
- In **Colors** Category choose color of the Rectangle and its border.

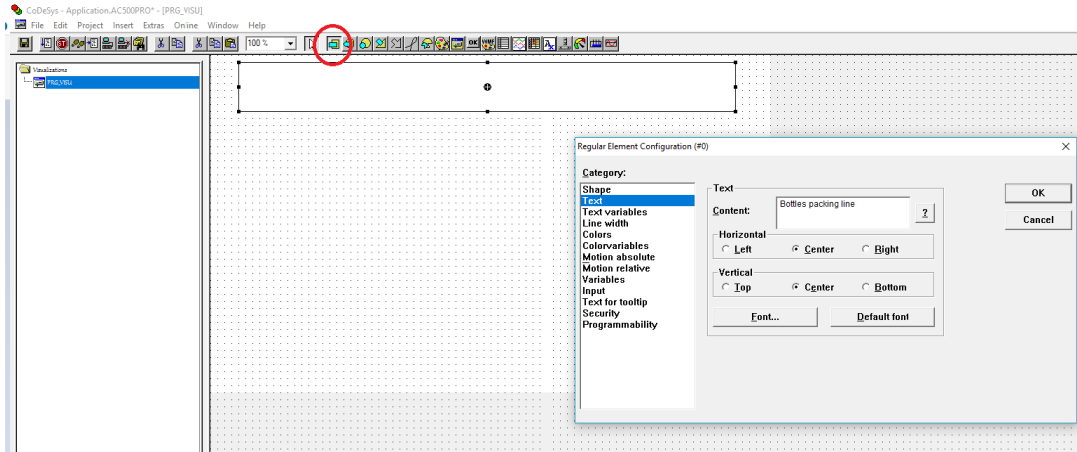


Figure 3: Adding Text

6.1.1 Buttons

Select 'Button' from the menu then using arrow set the area where button should be placed. Add 2 buttons which STARTs and STOPs the line.

For example for START button set the next parameters:

- Type "START" in the **Text** Category.
- In **Input** Category select Tap Variable (see Fig.4).
- To link main program variable with a button type *PLC_PRG.bStart*, where *PLC_PRG* - name of the program, *bStart* - name of the declared variable, that starts the line.

Do the same way for the STOP button. Set button color as "Red" in **Bitmap** Category.

6.1.2 Indicators

Operator must to know if line is working or not. We can provide such information using indicators. Choose Ellipse as an indicator.

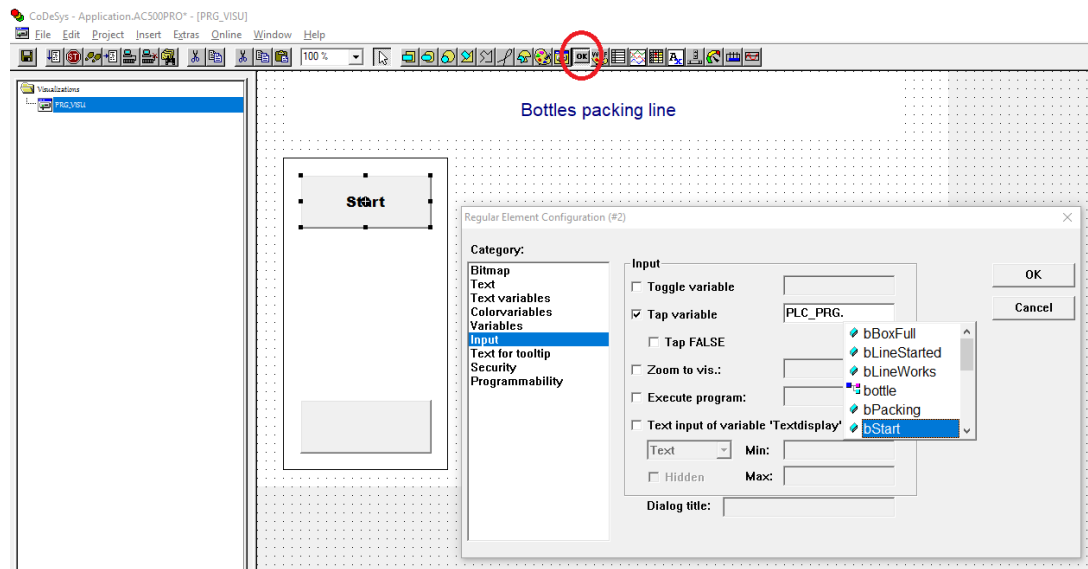


Figure 4: Category Input

- In **Text** Category type "Line started".
- To read variable value in **Input** Category check Toggle Variable and link it with program variable (for example: `PLC_PRG.bLineStarted`, where `PLC_PRG` - name of the program, `bLineStarted` - name of the declared variable, that indicates state of the line.)

A better way to draw operator's attention is to change color of the indicator while changing the state.

- In **Colors** Category choose **Alarm Color** as "Green".
- In **Variables** Category for **Change Color** as `PLC_PRG.bLineStarted`.

6.2 Compilation

In your program (`PLC_PRG`) change the code as you can start the line using one of the buttons (switch (S0)) or visual button (START).

```

IF bStart=1 OR S0 THEN
    bLineStarted :=1; (*line is started*)
END_IF;
IF bStop=1 OR S1 THEN
    bLineStarted :=0; (*line is stopped*)
END_IF;

```

Where `bStart`, `bStop`, `bLineStarted` are `BOOL` type variables.

Download program to the PLC. While running open visualization tab. Check how it works

1. Does click on the `START` button or turn on switch `S0` start the line?
2. Does indicator change its color while switching the line?

6.3 Other elements

Provide the following information to the operator:

- Using 'Meter' show how many bottles from 9 possible are in the box.
- Also use text to provide that information numerically (see help: Visualization, Configure Text).

In `Variables` Category set variable which shows bottle index in the box (for example: `PLC_PRG.intCurBottle`) to the `Textdisplay`.

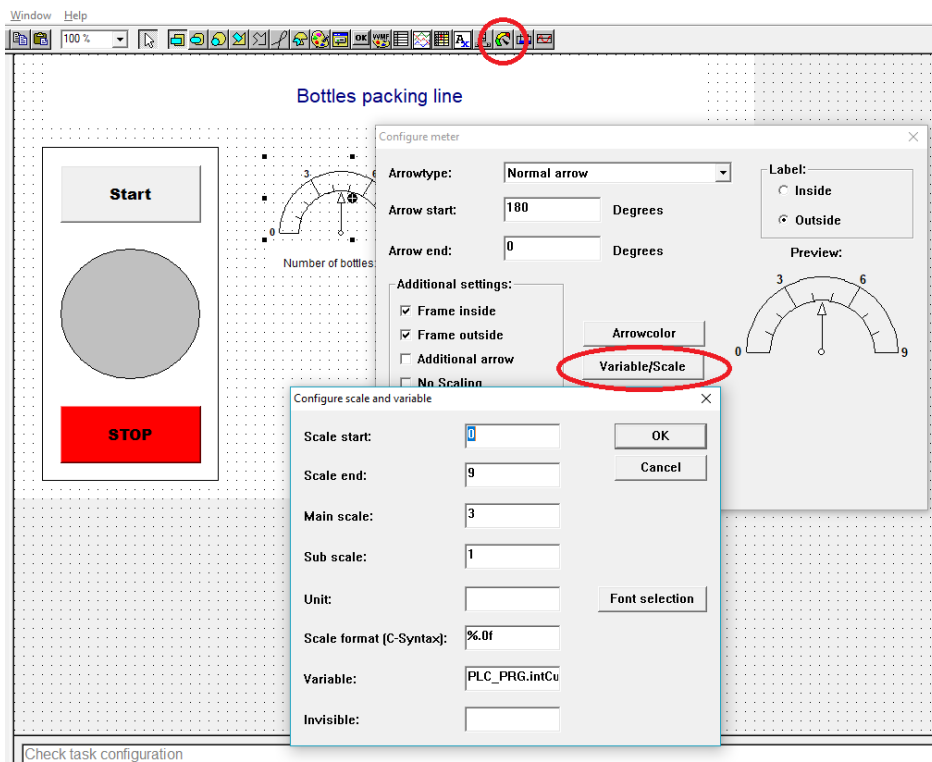


Figure 5: Meter Settings

6.3.1 Motion

Add one more 'Rectangle'. In order to show sent out packed box, the x-axis coordinates should be changed. In **Motions Absolute** enter `intX_offset` variable. This variable can shift the element in the X direction, depending on the respective variable value.

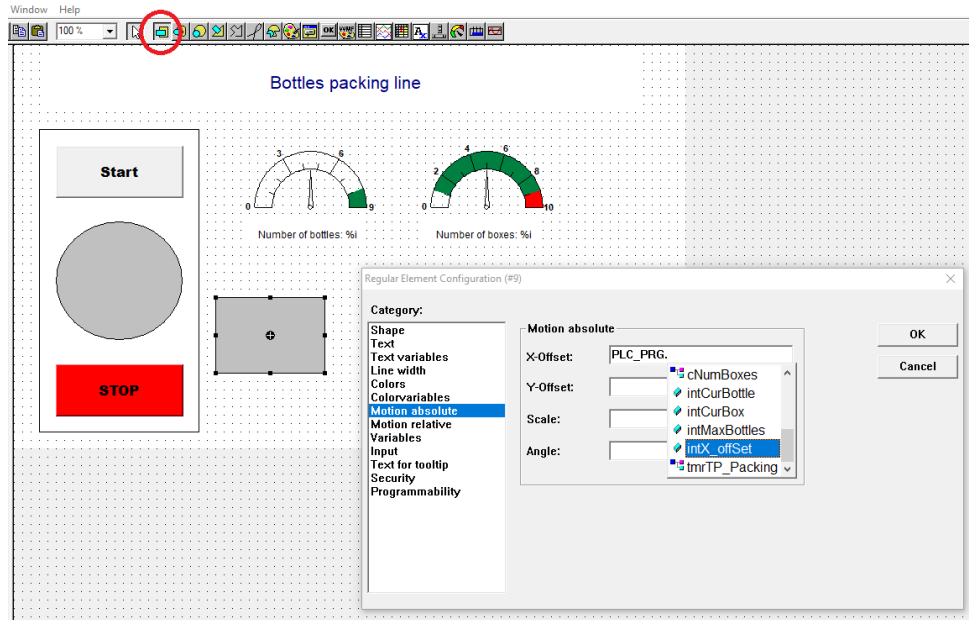


Figure 6: Adding Motion

NB! Do not forget to calculate new value of the `intX_offset` in your program.

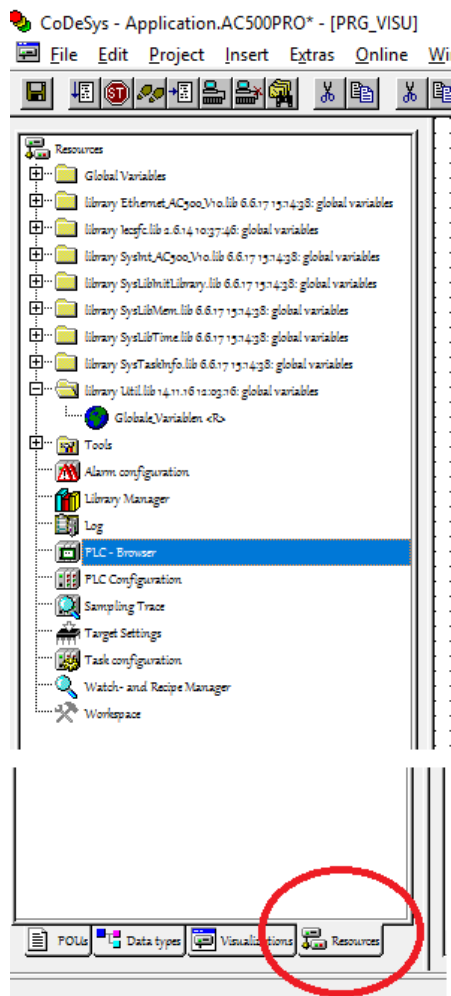
```

IF bPacking THEN
  intX_offset:= x_offSet + 1; (*new position of the box*)
  IF intX_offset = 250 THEN (*where 250 is the right edge of apps window*)
    intX_offset:= 0; (*initial position of the box if window edge is reached*)
  END_IF;
  ELSE intX_offset:= 0; (*initial position of the box if not packing*)
  END_IF;

```

Below the box provide information how many boxes have been packed.

6.3.2 Time



While in Login mode open **PLC-Browser** window. Type *time* command. You can see that time variable has been reset.

Set the right time to you PLC.

1. date 2017-mm-dd,
2. time HH:MM:SS.

Now, that information can be provided by your application.

On the visual part of you application add one more 'Rectangle'.

If you enter "%t", followed by a certain sequence of special placeholders, then this location will be replaced in Online mode by the system time. The placeholders define the display format, see the following table. See help file: Text.

Attention: Do not insert any other characters before "%t" in the 'Content' field.

Examples:

1. %t %a %b %d. %m. %y %H: %M: %S
→ Displays in online mode: Sun Aug 01.10.17 16:32:45
2. Between the placeholders you can insert any text strings:
 - %t Today is %A %d of %B %Y
 - → Displays in online mode: Today is Sunday 01 of October 2017.

In a **Text** Category set right placeholders in order to provide information

- Day of the week;

- Date;
- Time;

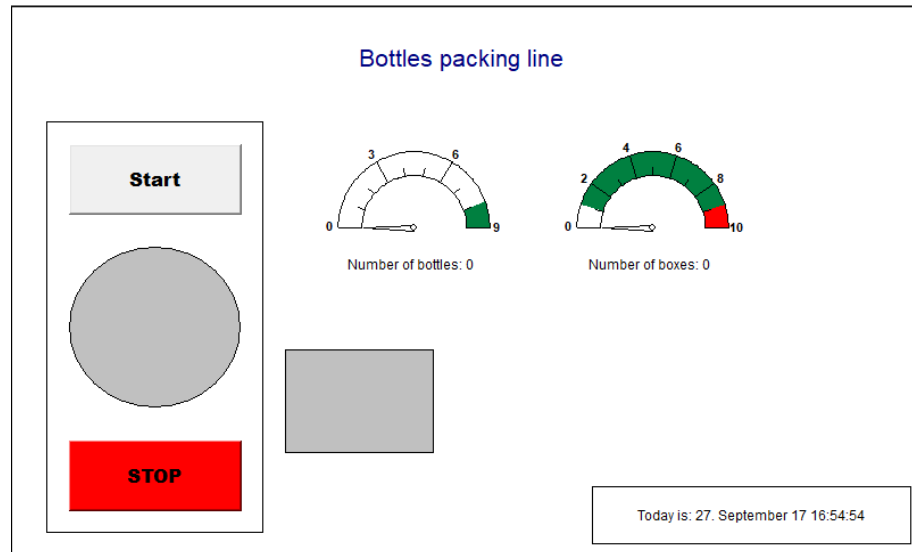


Figure 7: GUI

Additional example provided by ABB can be seen: [ABB AC500 PLC First Visualization](#)

7 Appendix: Wrapping machine visualization-listing

Declaration of variables:

```
PROGRAM PLC_PRG
VAR
  bLineStarted: BOOL;
  bStop: BOOL;
  bTrayFull: BOOL;
  bottle: BLINK;
  bStart: BOOL;
  tmrTP_Packing: TP;
  bBoxFull: BOOL;
  intCurBottle: INT;
  intX_offSet: INT;
  bPacking: BOOL;
  bLineWorks: BOOL;
  cNumBottles: CTU;
  intCurBox: INT;
  cNumBoxes: CTU;
  intTop: INT;
END_VAR
```

```
VAR CONSTANT
  intMaxBottles:INT := 9;
END_VAR
```

Programm listing

```
(*Visualization of the line turning on and off *)
IF bStart=1 OR S0 THEN
  bLineStarted:=1; (*line is started*)
END_IF;
IF bStop=1 OR S1 OR bTrayFull THEN
  bLineStarted:=0; (*line is stopped*)
END_IF;
(*Line working conditions*)
bLineWorks:=bLineStarted AND NOT bPacking;
(*Simulation of the sensor reaction on bottle *)
bottle(ENABLE:=bLineWorks, TIMELOW:= t#500ms , TIMEHIGH:= t#500ms); (*bottle: BLINK;*)
```

```
(*Physical output*)
IF NOT bBoxFull THEN
    D016:=bottle.OUT;
ELSE D016:=0;
END_IF;
(*Counting bottles*)
cNumBottles( (*cNumBottles: CTU;*)
CU:=bottle.OUT,
RESET:= bStart OR S0 OR (tmrTP_Packing.ET=tmrTP_Packing.PT),
PV:=intMaxBottles,
Q=>bBoxFull,
CV=>intCurBottle);
(*Packing bottles*)
tmrTP_Packing(IN:=bBoxFull, PT:= t#3s, Q=>bPacking);
(*Counting boxes*)
cNumBoxes(( *cNumBoxes: CTU;*)
CU:=bPacking,
RESET:=bStop OR S1 OR bTrayFull,
PV:=10,
Q=>bTrayFull,
CV=>intCurBox);
(*Box animation*)
IF bPacking THEN
    intX_offSet:=intX_offSet+2;
    IF intX_offSet=400 THEN
        intX_offSet:=0;
    END_IF;
ELSE intX_offSet:=0;
END_IF;
(*(*Filling the Box*)
IF bottle.OUT AND intCurBottle>0 THEN
    intTop:=intTop-1;
END_IF;
IF bPacking THEN
    intTop:=0;
END_IF;*)
```