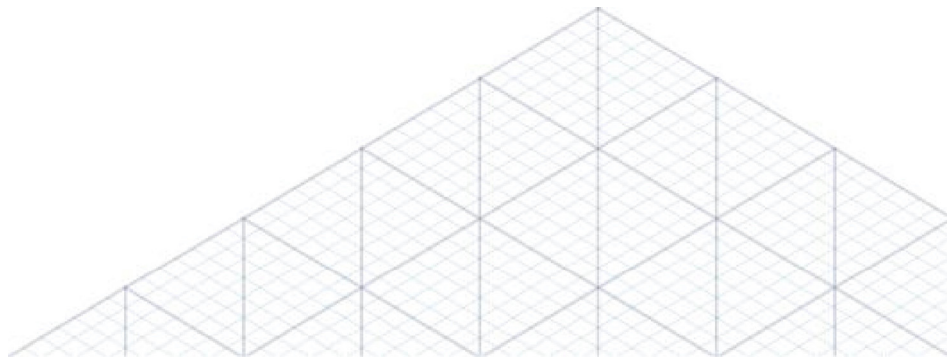




Brodersen WorkSuite Package Overview

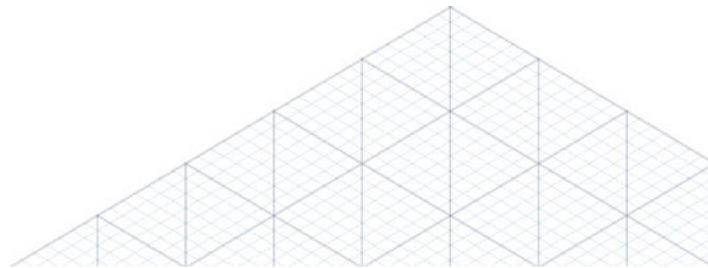
June 2014





IEC61131-3 Logic Editor

RTU32 Series Configuration & Programming



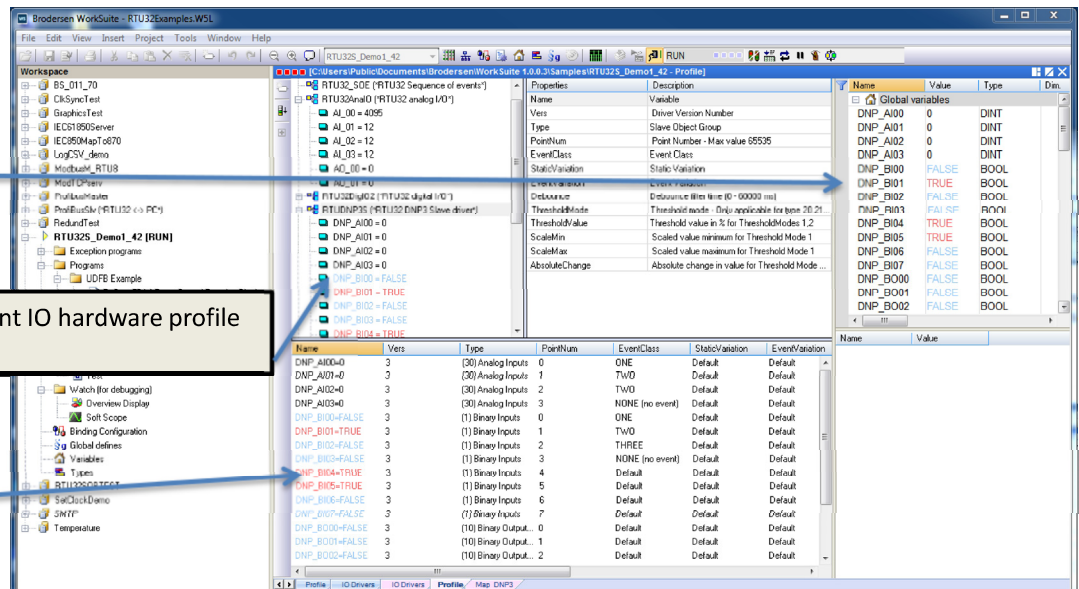
- Brodersen WorkSuite is a modern, flexible and open platform that integrates logic development with communications interfaces, has a wide range of functions and includes strong creation, commissioning and debugging tools.
- The configuration and programming facilities allow the Brodersen controllers to provide advanced functionality as a combined RTU, PLC and Communication Gateway.

The Profile Editor Tool allows rapid IO setup

1) Define Variables

2) 'Drag and Drop' variables to relevant IO hardware profile eg. RTU32 Digital IO

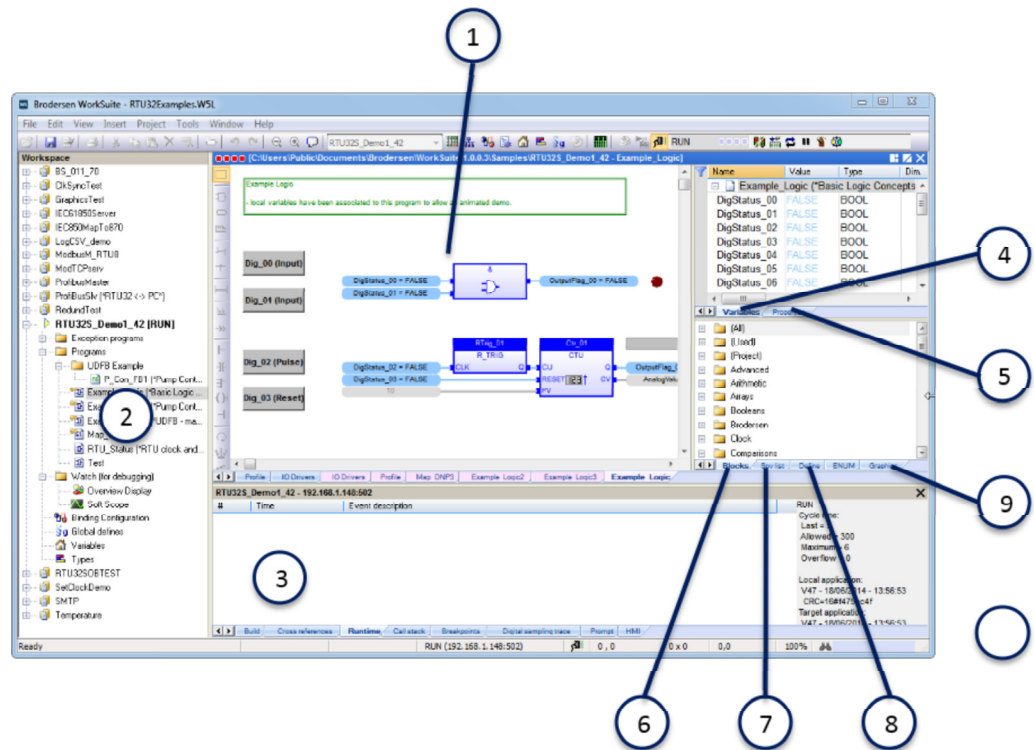
3) Define Hardware Address



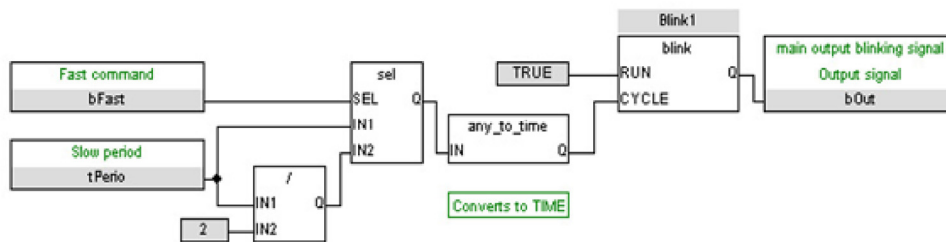
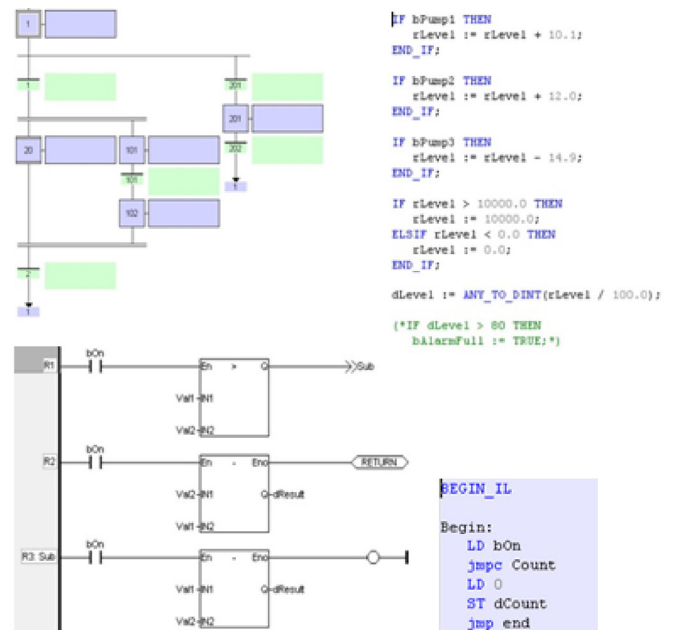
Name	Value	Type
DNP_AI00	0	DINT
DNP_AI01	0	DINT
DNP_AI02	0	DINT
DNP_AI03	0	DINT
DNP_BI00	FALSE	BOOL
DNP_BI01	TRUE	BOOL
DNP_BI02	FALSE	BOOL
DNP_BI03	FALSE	BOOL
DNP_BI04	TRUE	BOOL
DNP_BI05	FALSE	BOOL
DNP_BI06	FALSE	BOOL
DNP_BI07	FALSE	BOOL
DNP_BO00	FALSE	BOOL
DNP_BO01	FALSE	BOOL
DNP_BO02	FALSE	BOOL

Name	Vers	Type	PointNum	EventClass	StaticVariation	EventVariation
DNP_AI00=0	3	(30) Analog Inputs	0	ONE	Default	Default
DNP_AI01=0	3	(30) Analog Inputs	1	TWO	Default	Default
DNP_AI02=0	3	(30) Analog Inputs	2	TWO	Default	Default
DNP_AI03=0	3	(30) Analog Inputs	3	NONE (no event)	Default	Default
DNP_BI00=FALSE	3	(1) Binary Inputs	0	ONE	Default	Default
DNP_BI01=TRUE	3	(1) Binary Inputs	1	TWO	Default	Default
DNP_BI02=FALSE	3	(1) Binary Inputs	2	THREE	Default	Default
DNP_BI03=FALSE	3	(1) Binary Inputs	3	NONE (no event)	Default	Default
DNP_BI04=TRUE	3	(1) Binary Inputs	4	Default	Default	Default
DNP_BI05=TRUE	3	(1) Binary Inputs	5	Default	Default	Default
DNP_BI06=FALSE	3	(1) Binary Inputs	6	Default	Default	Default
DNP_BI07=FALSE	3	(7) Binary Inputs	7	Default	Default	Default
DNP_BO00=FALSE	3	(10) Binary Output...	0	Default	Default	Default
DNP_BO01=FALSE	3	(10) Binary Output...	1	Default	Default	Default
DNP_BO02=FALSE	3	(10) Binary Output...	2	Default	Default	Default

1. Main document/program/graphics view
2. The workspace list
3. The output window (messages/diags)
4. The variable editor is used for declaring variables and instances
5. The property grid and "z-order" for designing graphic documents
6. The "blocks" pane lists available functions and blocks
7. The "spy list" enables quick dynamic watch of variables during debug
8. The "define" pane shows various types of definitions
9. The "graphics" pane lists all available kinds of graphic objects



- Supports all five IEC 61131-3 languages
 - Sequential function chart
 - Structured text
 - Ladder programming
 - Instruction list
 - Function block programming
- Easy programming tool
- Fast compiler
- Powerful online-change functionality
- Allows online changes of logic
- German, English, French, Italian, Spanish, Korean



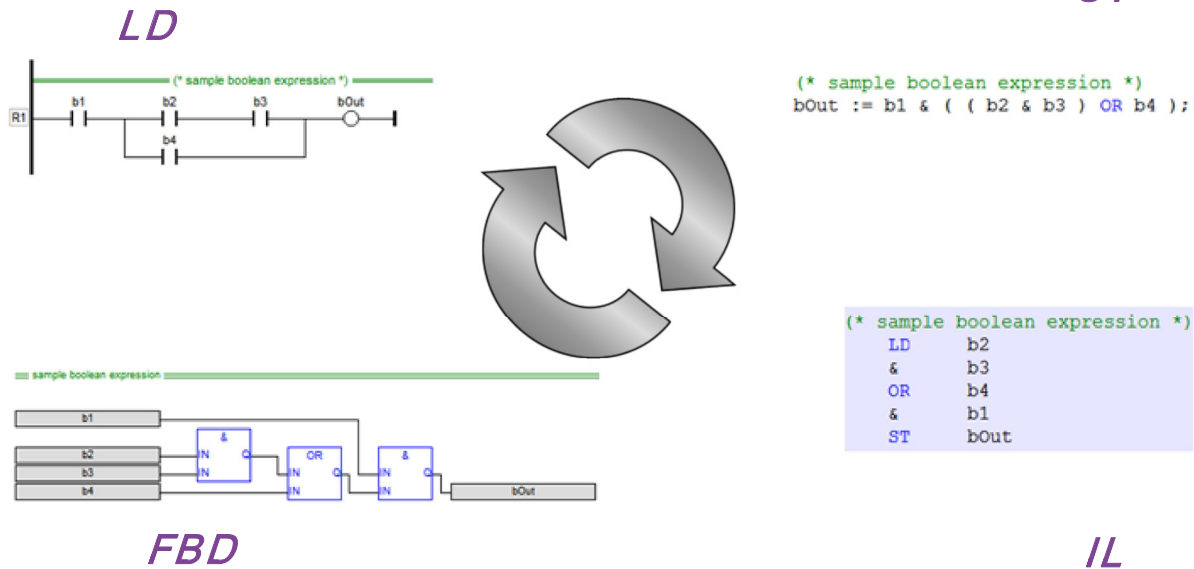
```

BEGIN_IL
Begin:
  LD bOn
  jmpc Count
  LD 0
  ST dCount
  jmp end

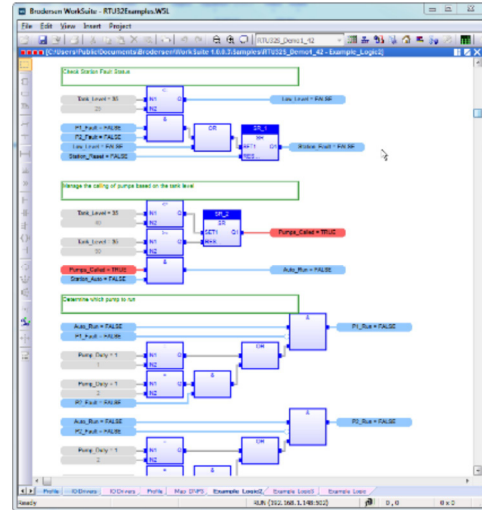
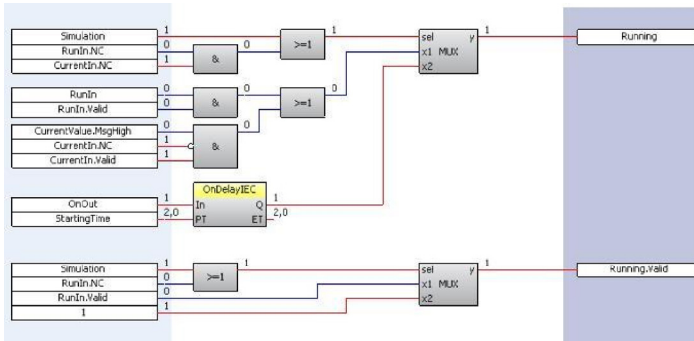
Count:
  LD dCount
  add 1
  ST dCount

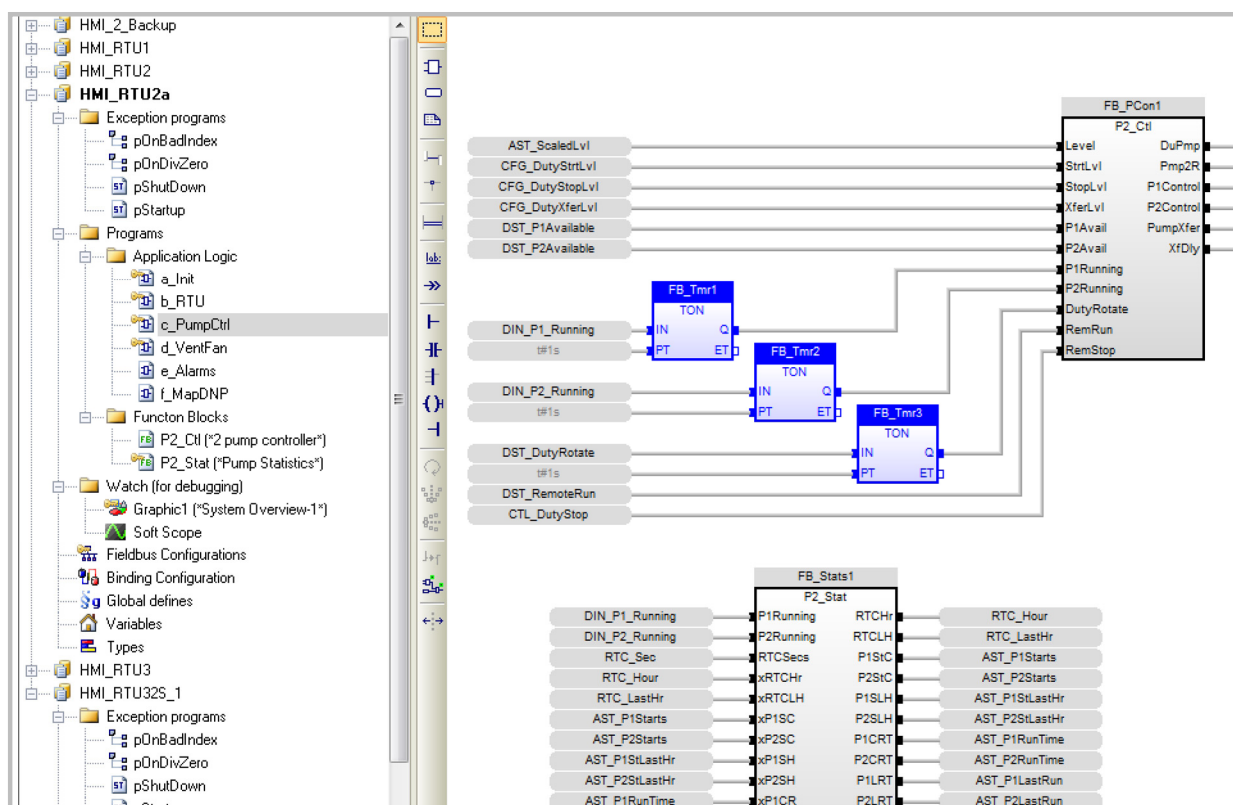
End:
END_IL
            
```

Language conversion i.e. convert Ladder Logic (LD) code to/from Structured Text (ST)



Translation from other IEC61131-3 editors to WorkSuite – it's easy!

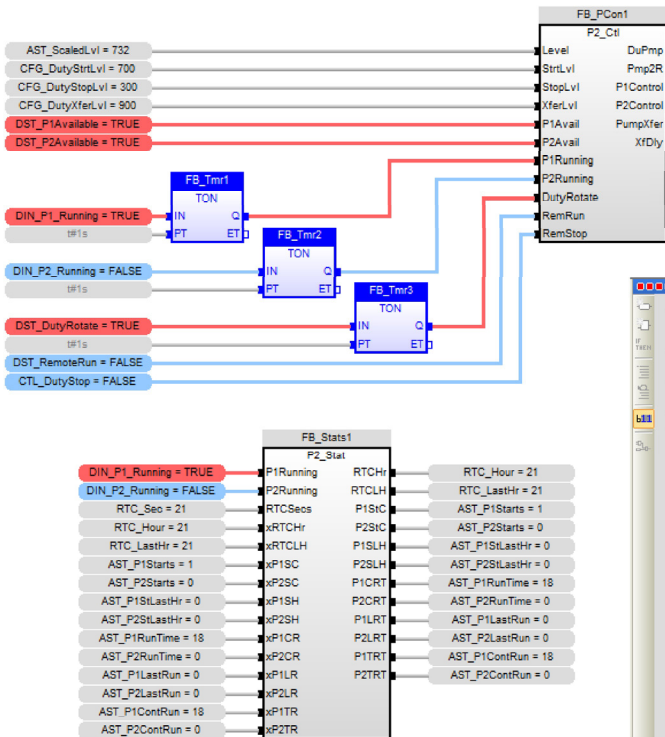




Example Pump Control Logic – using Function Block Diagram

9

1) Debug a User Program



2) Debug a User Defined Function Block

```

(* Duty Rotation Logic *)
(* Pump Stop Event*)
PumpStopEvent FALSE := (P1Running_prev TRUE OR P2Running_prev FALSE) AND NOT P1Running AND NOT P2Running;

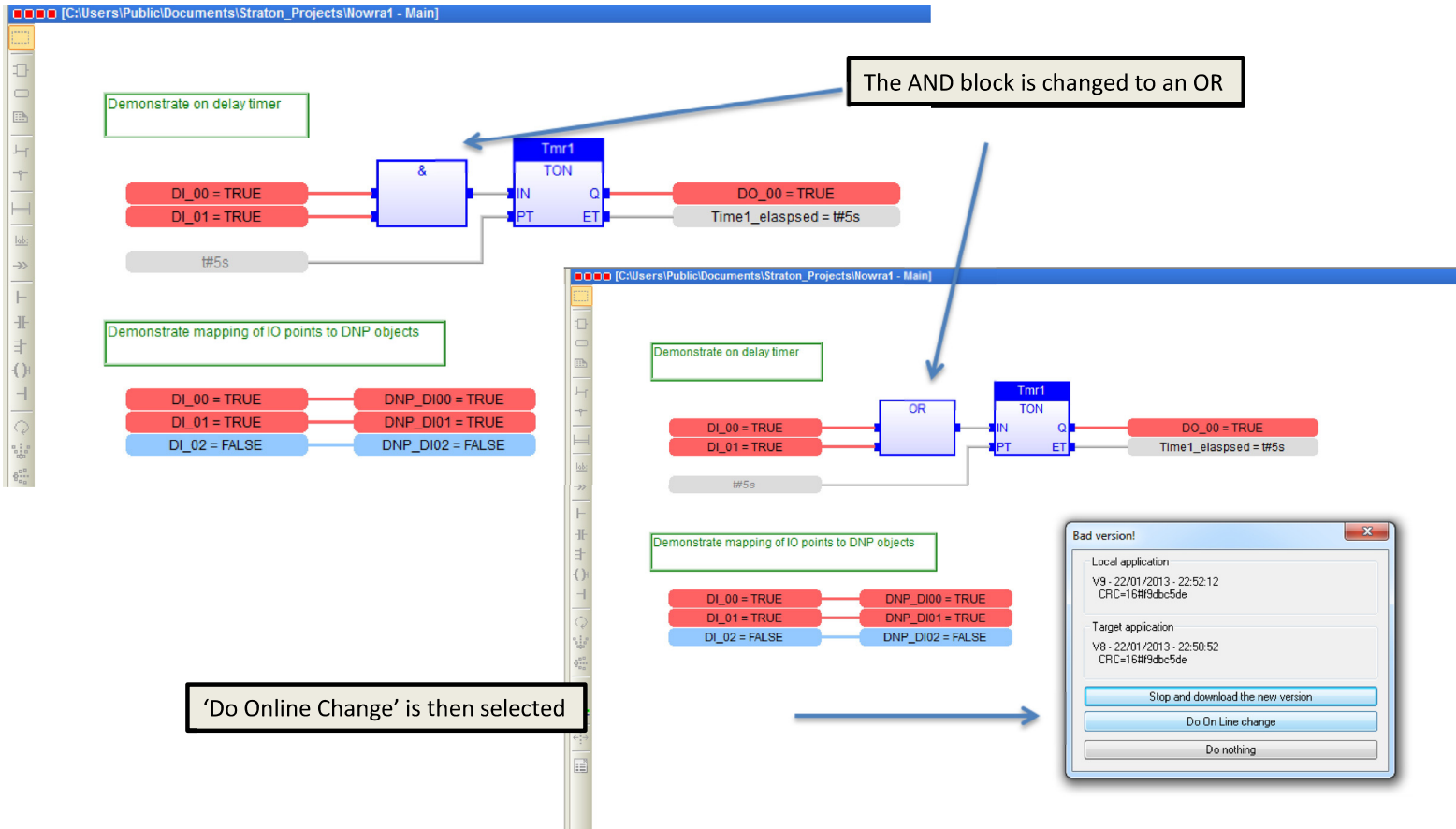
(* On a pump stop event, rotate the duty pump *)
IF (PumpStopEvent FALSE AND DutyRotate AND NOT (Level > XferLvl) AND NOT XferActive FALSE) THEN
  DutyPump 1 := DutyPump 1 + 1;
  IF (DutyPump 1 > 2) THEN
    DutyPump 1 := 1;
  END_IF;
END_IF;

(* Ensure DutyPump has a sensible value (when the application is initialised it may have an unknown state) *)
IF (DutyPump 1 < 1 OR DutyPump 1 > 2) THEN
  DutyPump 1 := 1;
END_IF;

(* Duty Transfer Logic *)
(* If the wet well level is greater than the transfer level, then run the alternate pump
 * otherwise run the normal duty pump
 *)
Transfer_Tmr (XferActive FALSE, t#100s);

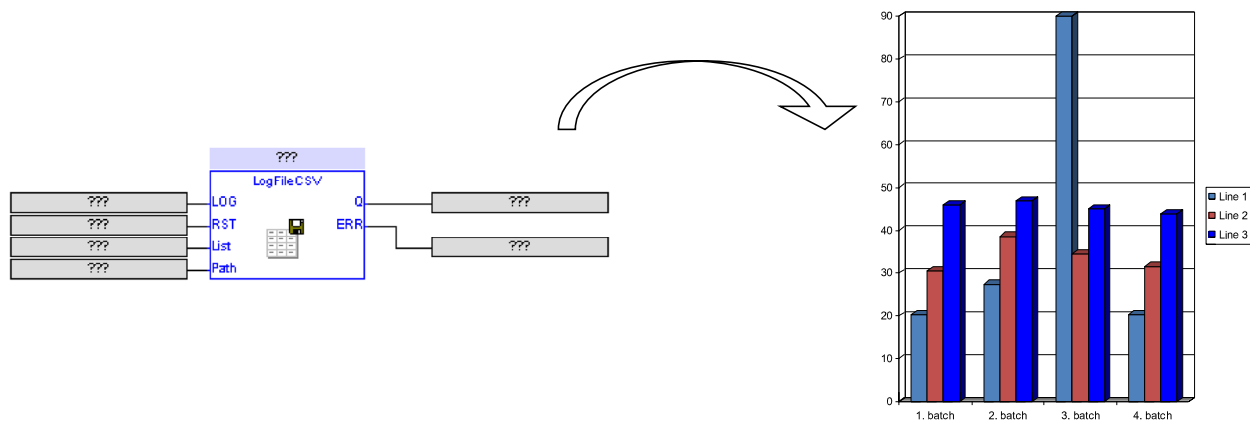
IF (Level > XferLvl) AND NOT XferActive FALSE THEN
  XferActive FALSE := TRUE;
END_IF;
    
```

3) Debug Structured Text Source Code



RTU32 supports

- Flexible design of your own data logging processes in WorkSuite
- Event or time based logging of all kind of PLC variables
- Time stamp with millisec resolution
- Log to CSV text files locally on the Flash disc, connected Flash memory stick or network folder
- Dedicated Log Functions
- Manage your log files with WorkSuite file functions like Copy, Rename and Delete files



1) Create (and monitor) in WorkSuite

2) Publish as HTML file

3) Download to RTU

4) View remotely via RTU Web Server