TIA Portal
vs CODESYS
vs SIMATIC AX
vs ?

Does it matter anymore?

TIA Portal **CODESYS** **SIMATIC AX**

The same control-program model, compiled by three connectors into native projects. Swipe to compare.

The model is the master

Tags, hardware, data types, blocks and tests are records in the model, each with its source. The connectors only translate them.

Oracle model

```
// one PLC tag
{
  "id": "tag-di0",
  "symbol": "i_xLevelHighHighOk",
  "data_type": "BOOL",
  "io_class": "di",
  "meaning":
    "TRUE = level below high-high",
  "safety_relevance":
    { "class": "process-protection" },
  "electrical_function_id": "fn-di0",
  "true_is_healthy": true,
  "sources": [
    "circuit-design-r5.json",
    "docs/166 s4" ],
  "open_point_ids": [ "u-lvl100" ]
}

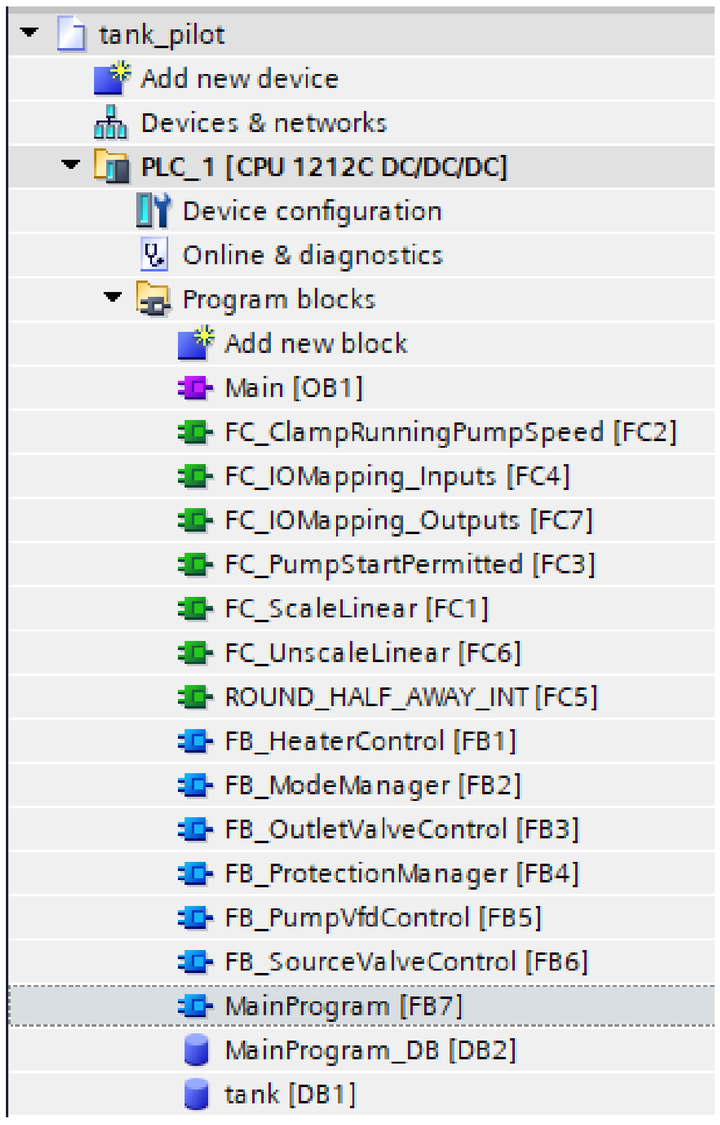
// the same function links it to the
// switch, terminals and wires
// on the schematic
```

```
// one function block
{
  "id": "pu-mode",
  "name": "FB_ModeManager",
  "kind": "function_block",
  "interface": {
    "in_out":  [ tank : dt-tank ],
    "statics": [ _xAutomaticFillRequest
                 : BOOL ],
    "temps":   [ rLevelSetpoint : REAL
                 (unit "%"), ... ],
    "constant_ids": [ "k-level-band",
                      "k-sp-min", ... ] },
  "body": { "mode": "authored",
    "text": "rLevelSetpoint :=
      tank.commands.rLevelSetpointPercent;
      IF rLevelSetpoint <
         LEVEL_SETPOINT_MIN_PERCENT THEN
      ..." }
}
```

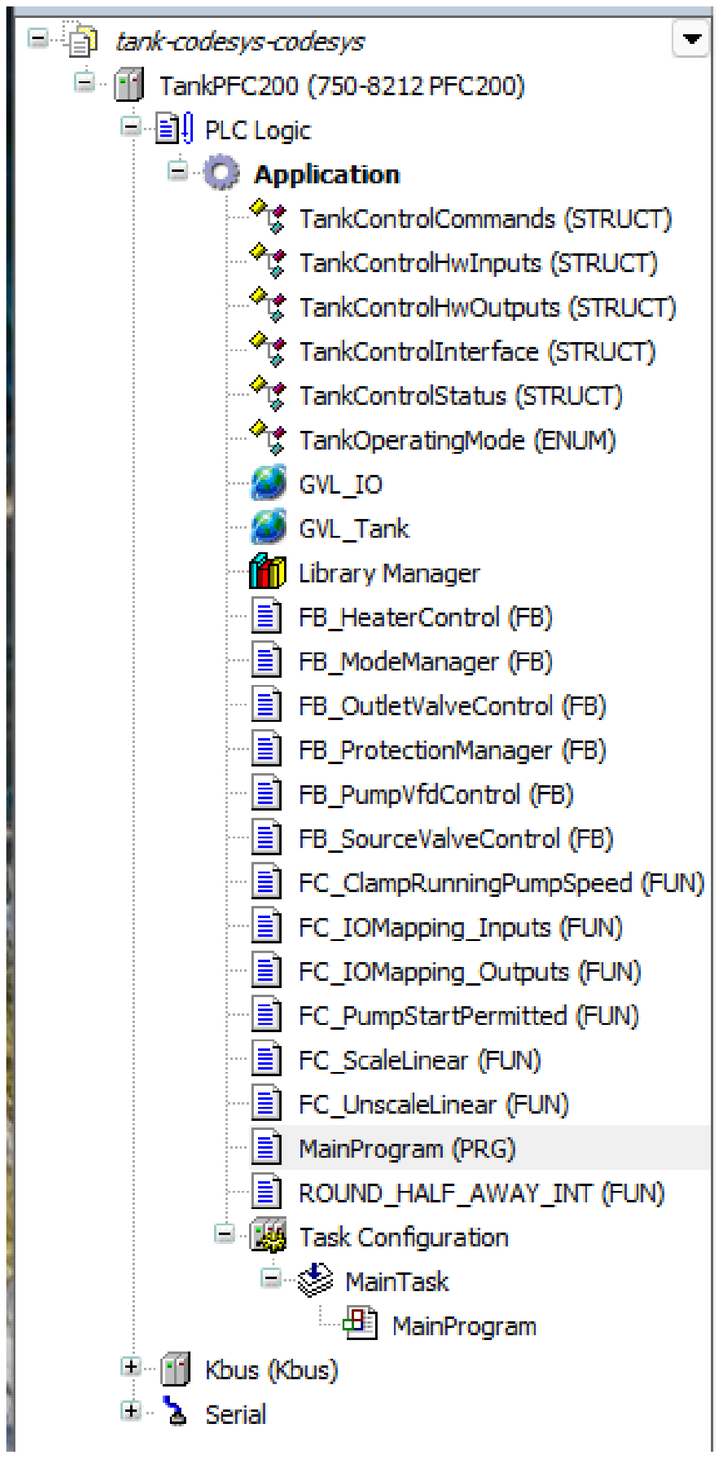
Project tree

The same blocks, types and tags from the model.
Each project is organised the way its own tool expects.

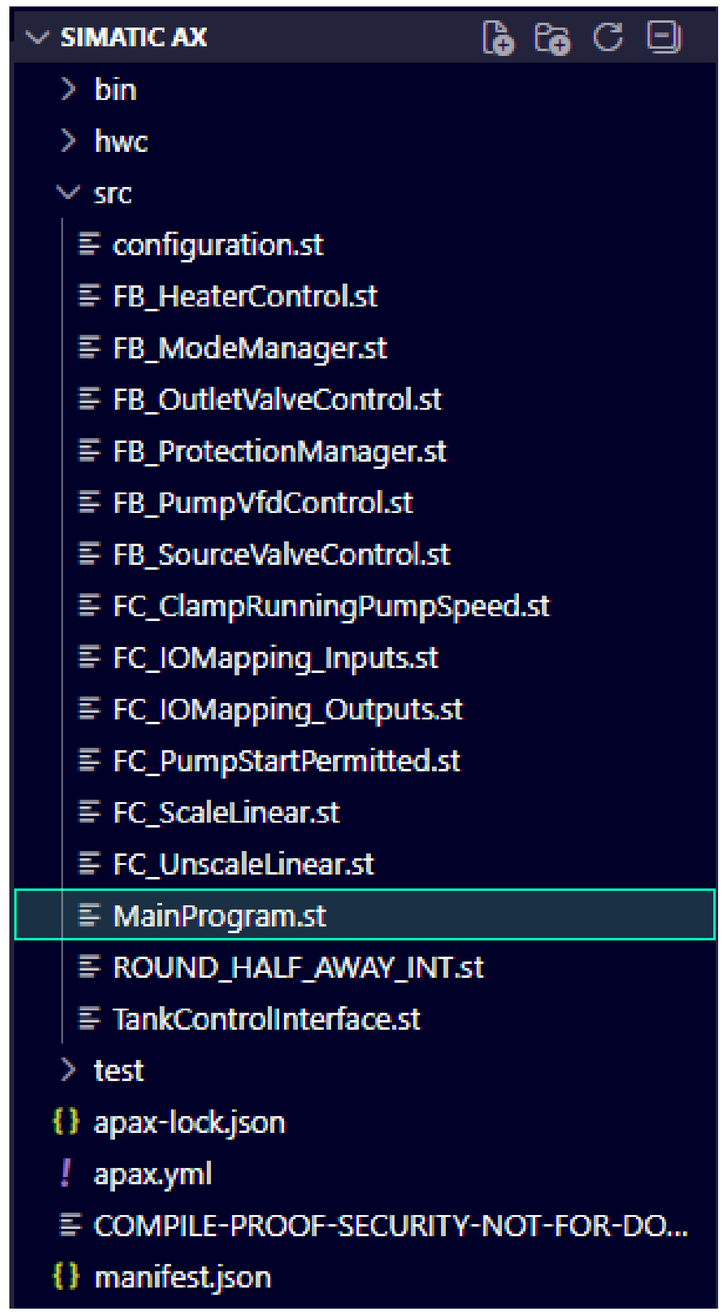
TIA Portal



CODESYS



SIMATIC AX



Main program

Same call order in all three. TIA writes instances with # and quotes globals;
AX works in a namespace; CODESYS takes plain IEC Structured Text.

TIA Portal

```
1
2       // STEP 1: read physical inputs and scale analogue inputs
3       "FC_IOMapping_Inputs"(tank := "tank");
4
5       // STEP 2: control logic
6       #_fbProtectionManager(tank := "tank");
7       #_fbModeManager(tank := "tank");
8       #_fbSourceValveControl(tank := "tank");
9       #_fbPumpVfdControl(tank := "tank");
10      #_fbOutletValveControl(tank := "tank");
11      #_fbHeaterControl(tank := "tank");
12
13      // STEP 3: scale analogue outputs and write physical outputs
14      "FC_IOMapping_Outputs"(tank := "tank");
```

CODESYS

```
1   // STEP 1: read physical inputs and scale analogue inputs
2   FC_IOMapping_Inputs(tank := tank);
3
4   // STEP 2: control logic
5   _fbProtectionManager(tank := tank);
6   _fbModeManager(tank := tank);
7   _fbSourceValveControl(tank := tank);
8   _fbPumpVfdControl(tank := tank);
9   _fbOutletValveControl(tank := tank);
10  _fbHeaterControl(tank := tank);
11
12  // STEP 3: scale analogue outputs and write physical outputs
13  FC_IOMapping_Outputs(tank := tank);
```

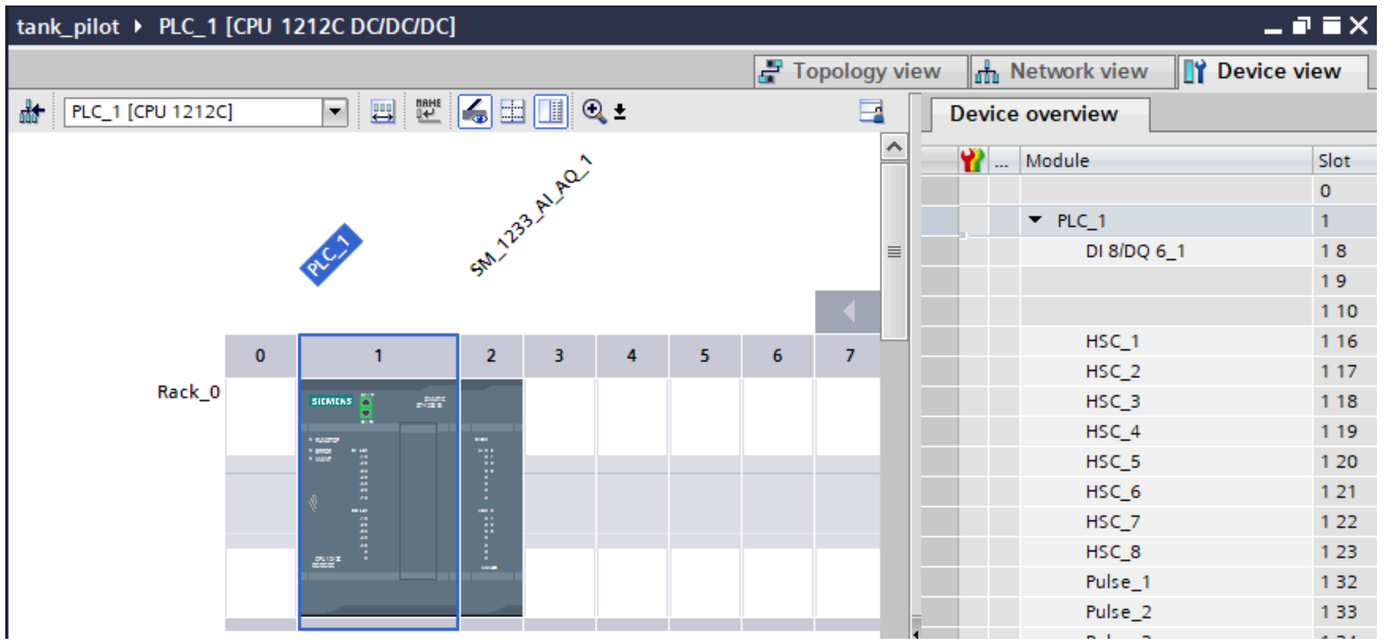
SIMATIC AX

```
16      // STEP 1: read physical inputs and scale analogue inputs
17      FC_IOMapping_Inputs(tank := tank);
18
19      // STEP 2: control logic
20      _fbProtectionManager(tank := tank);
21      _fbModeManager(tank := tank);
22      _fbSourceValveControl(tank := tank);
23      _fbPumpVfdControl(tank := tank);
24      _fbOutletValveControl(tank := tank);
25      _fbHeaterControl(tank := tank);
26
27      // STEP 3: scale analogue outputs and write physical outputs
28      FC_IOMapping_Outputs(tank := tank);
29  END_PROGRAM
```

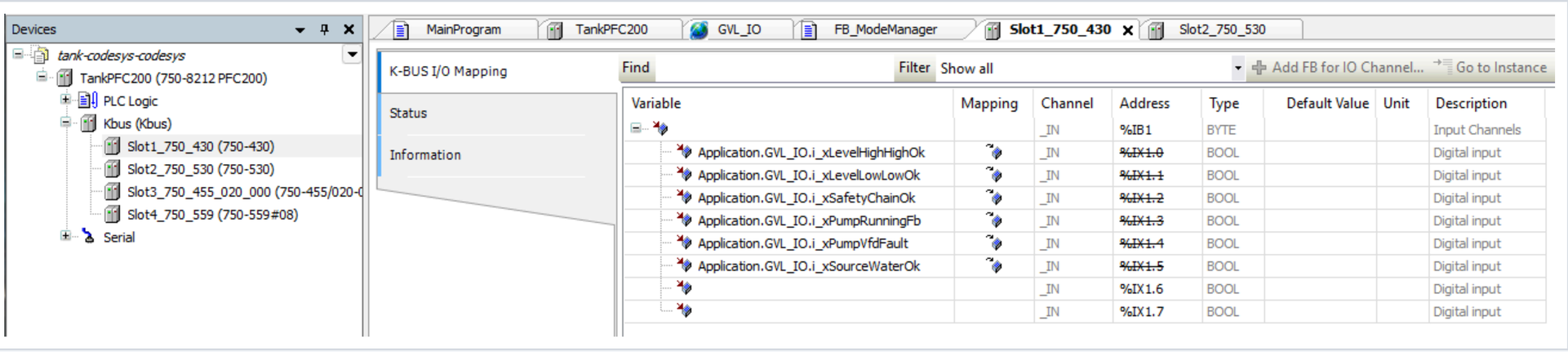
Hardware

TIA Portal and SIMATIC AX: S7-1200 G2 CPU 1212C with SM 1233 in slot 2.
CODESYS variant: WAGO PFC200 with its K-bus I/O in the model's slot order.

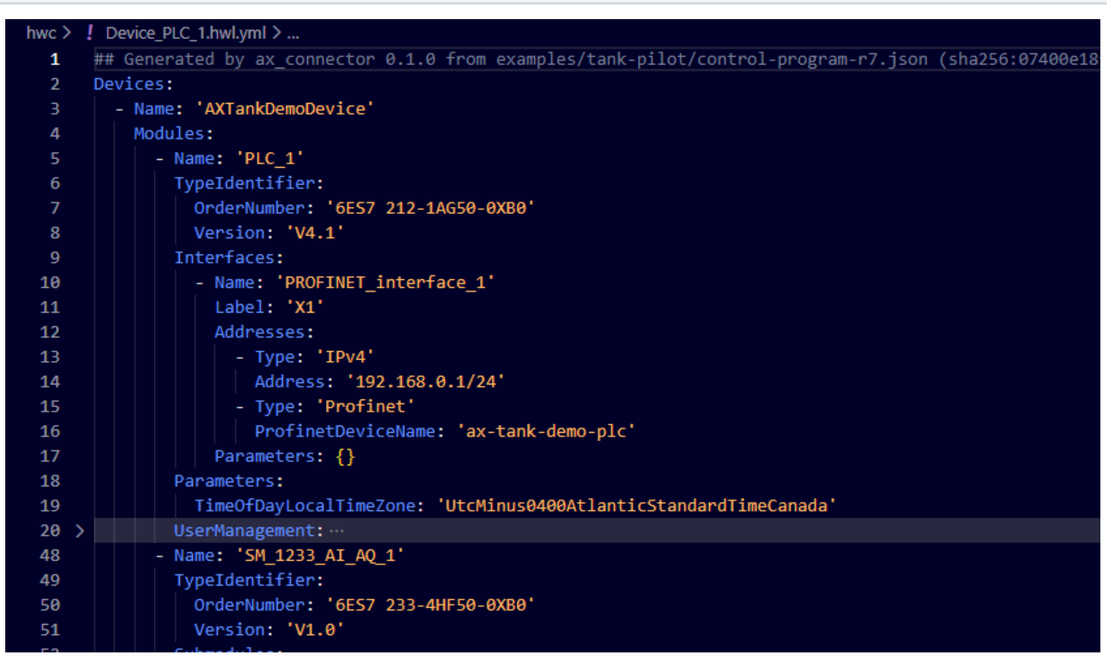
TIA Portal



CODESYS



SIMATIC AX



PLC tags

14 PLC points with the same symbols and comments, from one tag list in the model.
CODESYS maps them onto the WAGO K-bus channels (previous slide).

TIA Portal

	Name	Tag table	Data type	Address	Retain	Acces...	Writa...	Visibl...	Comment
1	i_iTankLevelRaw	Oracle IO	Int	%IW64		☑	☑	☑	tank level
2	i_iTankTemperatureRaw	Oracle IO	Int	%IW66		☑	☑	☑	tank temperature
3	q_iPumpSpeedRaw	Oracle IO	Int	%QW64		☑	☑	☑	VFD speed reference
4	i_xLevelHighHighOk	Oracle IO	Bool	%I0.0		☑	☑	☑	TRUE = level below high-high
5	i_xLevelLowLowOk	Oracle IO	Bool	%I0.1		☑	☑	☑	TRUE = level above low-low
6	i_xSafetyChainOk	Oracle IO	Bool	%I0.2		☑	☑	☑	TRUE = hardwired safety chain healthy
7	i_xPumpRunningFb	Oracle IO	Bool	%I0.3		☑	☑	☑	TRUE = VFD reports running
8	i_xPumpVfdFault	Oracle IO	Bool	%I0.4		☑	☑	☑	TRUE = VFD fault
9	i_xSourceWaterOk	Oracle IO	Bool	%I0.5		☑	☑	☑	TRUE = flow switch proves water at the pu...
10	q_xSourceValveOpen	Oracle IO	Bool	%Q0.0		☑	☑	☑	TRUE = open source valve
11	q_xPumpVfdEnable	Oracle IO	Bool	%Q0.1		☑	☑	☑	TRUE = enable VFD
12	q_xPumpVfdRun	Oracle IO	Bool	%Q0.2		☑	☑	☑	TRUE = run VFD
13	q_xHeaterEnable	Oracle IO	Bool	%Q0.3		☑	☑	☑	TRUE = heater on
14	q_xOutletValveOpen	Oracle IO	Bool	%Q0.4		☑	☑	☑	TRUE = open outlet valve

Libraries
Add-Ins

CODESYS

```
VAR_GLOBAL
    i_iTankLevelRaw       : INT;  (* tank level *)
    i_iTankTemperatureRaw : INT;  (* tank temperature *)
    q_iPumpSpeedRaw       : INT;  (* VFD speed reference *)
    i_xLevelHighHighOk    : BOOL; (* TRUE = level below high-high *)
    i_xLevelLowLowOk      : BOOL; (* TRUE = level above low-low *)
    i_xSafetyChainOk      : BOOL; (* TRUE = hardwired safety chain healthy *)
    i_xPumpRunningFb      : BOOL; (* TRUE = VFD reports running *)
    i_xPumpVfdFault       : BOOL; (* TRUE = VFD fault *)
    i_xSourceWaterOk      : BOOL; (* TRUE = flow switch proves water at the pump *)
    q_xSourceValveOpen    : BOOL; (* TRUE = open source valve *)
    q_xPumpVfdEnable      : BOOL; (* TRUE = enable VFD *)
    q_xPumpVfdRun         : BOOL; (* TRUE = run VFD *)
    q_xHeaterEnable       : BOOL; (* TRUE = heater on *)
    q_xOutletValveOpen    : BOOL; (* TRUE = open outlet valve *)
```

SIMATIC AX

```
CONFIGURATION MyConfiguration
    TASK Main(Priority := 1);
    PROGRAM P1 WITH Main: MainProgram;
    VAR_GLOBAL
        tank: TankControlInterface;
        i_xLevelHighHighOk AT %I0.0 : BOOL;
        i_xLevelLowLowOk AT %I0.1 : BOOL;
        i_xSafetyChainOk AT %I0.2 : BOOL;
        i_xPumpRunningFb AT %I0.3 : BOOL;
        i_xPumpVfdFault AT %I0.4 : BOOL;
        i_xSourceWaterOk AT %I0.5 : BOOL;
        i_iTankLevelRaw AT %IW64 : INT;
        i_iTankTemperatureRaw AT %IW66 : INT;
        q_xSourceValveOpen AT %Q0.0 : BOOL;
        q_xPumpVfdEnable AT %Q0.1 : BOOL;
        q_xPumpVfdRun AT %Q0.2 : BOOL;
        q_xHeaterEnable AT %Q0.3 : BOOL;
        q_xOutletValveOpen AT %Q0.4 : BOOL;
        q_iPumpSpeedRaw AT %QW64 : INT;
```

Same function block

FB_ModeManager, written by each connector from the one model body.
TIA marks locals with #. AX and CODESYS write them plain.

TIA Portal

```
    #rLevelSetpoint := #tank.commands.rLevelSetpointPercent;
    IF #rLevelSetpoint < #LEVEL_SETPOINT_MIN_PERCENT THEN
        #rLevelSetpoint := #LEVEL_SETPOINT_MIN_PERCENT;
    ELSIF #rLevelSetpoint > #LEVEL_SETPOINT_MAX_PERCENT THEN
        #rLevelSetpoint := #LEVEL_SETPOINT_MAX_PERCENT;
    END_IF;

    #xLowLowOnlyRecovery := #tank.status.wAlarmWord0 = #ALARM_LEVEL_LOW_LOW;
    #tank.status.wWarningWord0 := WORD#16#0000;
    IF #tank.commands.rLevelSetpointPercent <> #rLevelSetpoint THEN
        #tank.status.wWarningWord0 := #tank.status.wWarningWord0 OR #WARNING_SETPOINT_CLAMPED;
    END_IF;
    IF #tank.commands.xSystemEnable AND
       (#tank.commands.xManualModeRequest = #tank.commands.xAutomaticModeRequest) THEN
        #tank.status.wWarningWord0 := #tank.status.wWarningWord0 OR #WARNING_MODE_REQUEST_CONFLICT;
    END_IF;
```

CODESYS

```
rLevelSetpoint := tank.commands.rLevelSetpointPercent;
IF rLevelSetpoint < LEVEL_SETPOINT_MIN_PERCENT THEN
    rLevelSetpoint := LEVEL_SETPOINT_MIN_PERCENT;
ELSIF rLevelSetpoint > LEVEL_SETPOINT_MAX_PERCENT THEN
    rLevelSetpoint := LEVEL_SETPOINT_MAX_PERCENT;
END_IF;

xLowLowOnlyRecovery := tank.status.wAlarmWord0 = ALARM_LEVEL_LOW_LOW;
tank.status.wWarningWord0 := WORD#16#0000;
IF tank.commands.rLevelSetpointPercent <> rLevelSetpoint THEN
    tank.status.wWarningWord0 := tank.status.wWarningWord0 OR WARNING_SETPOINT_CLAMPED;
END_IF;
IF tank.commands.xSystemEnable AND
   (tank.commands.xManualModeRequest = tank.commands.xAutomaticModeRequest) THEN
    tank.status.wWarningWord0 := tank.status.wWarningWord0 OR WARNING_MODE_REQUEST_CONFLICT;
END_IF;
```

SIMATIC AX

```
    rLevelSetpoint := tank.commands.rLevelSetpointPercent;
    IF rLevelSetpoint < LEVEL_SETPOINT_MIN_PERCENT THEN
        rLevelSetpoint := LEVEL_SETPOINT_MIN_PERCENT;
    ELSIF rLevelSetpoint > LEVEL_SETPOINT_MAX_PERCENT THEN
        rLevelSetpoint := LEVEL_SETPOINT_MAX_PERCENT;
    END_IF;

    xLowLowOnlyRecovery := tank.status.wAlarmWord0 = ALARM_LEVEL_LOW_LOW;
    tank.status.wWarningWord0 := WORD#16#0000;
    IF tank.commands.rLevelSetpointPercent <> rLevelSetpoint THEN
        tank.status.wWarningWord0 := tank.status.wWarningWord0 OR WARNING_SETPOINT_CLAMPED;
    END_IF;
    IF tank.commands.xSystemEnable AND
       (tank.commands.xManualModeRequest = tank.commands.xAutomaticModeRequest) THEN
        tank.status.wWarningWord0 := tank.status.wWarningWord0 OR WARNING_MODE_REQUEST_CONFLICT;
```

Every tool has its quirks

Convert 2.5 from REAL to INT. Same code, different answers.
So the model uses one explicit rounding function in every tool.

	TIA Portal	CODESYS	SIMATIC AX
Implicit conversion 2.5 -> INT	2	3	2
Implicit conversion -2.5 -> INT	-2	-3	-2
Model function ROUND_HALF_AWAY_INT(2.5)	3	3	3

Measured, not assumed: PLCSIM Advanced for TIA Portal, CODESYS simulation,
a SIMATIC AX probe. Each connector writes the model function, so all three agree.

How do I know they match the model?

Every connector run ends with checks, not with a screenshot.

	TIA Portal	CODESYS	SIMATIC AX
Compile	0 errors 0 warnings	0 errors 0 warnings	0 errors 0 warnings
Compared with the model	block interfaces read back	POU interfaces and 14 tags read back	build and unit tests
Same 14 test cases	14 / 14 passed PLCSIM Advanced	14 / 14 passed CODESYS simulation	14 / 14 passed AxUnit

And the supply-chain question?

CODESYS can't run a Siemens CPU. So the CODESYS project comes from a variant of the tank model: a declared change, not a redesign.

Changed in the variant

PLC: S7-1200 G2 CPU 1212C + SM 1233
-> WAGO PFC200 750-8212
+ 750-430 / 530 / 455 / 559 I/O

HMI: Siemens panel
-> Weintek cMT2078X

PLC I/O pages of the drawings, regenerated for the new modules

Identical to the base

Control program: blocks, logic, tag names and the 14 tests

Safety circuit

Drive (VFD) wiring

Field terminals and fused feeds

Device tags of everything that did not change

Changing the PLC vendor becomes a change to the model. Every connector regenerates.

The model is the master.
The PLC tool is the delivery format.

Customer standard is Siemens? **TIA Portal** **SIMATIC AX**

Open platform or another vendor? **CODESYS**

Same tags. Same logic. Same tests. Any platform.

Still ahead: HMI screens from the model, and running on real hardware.