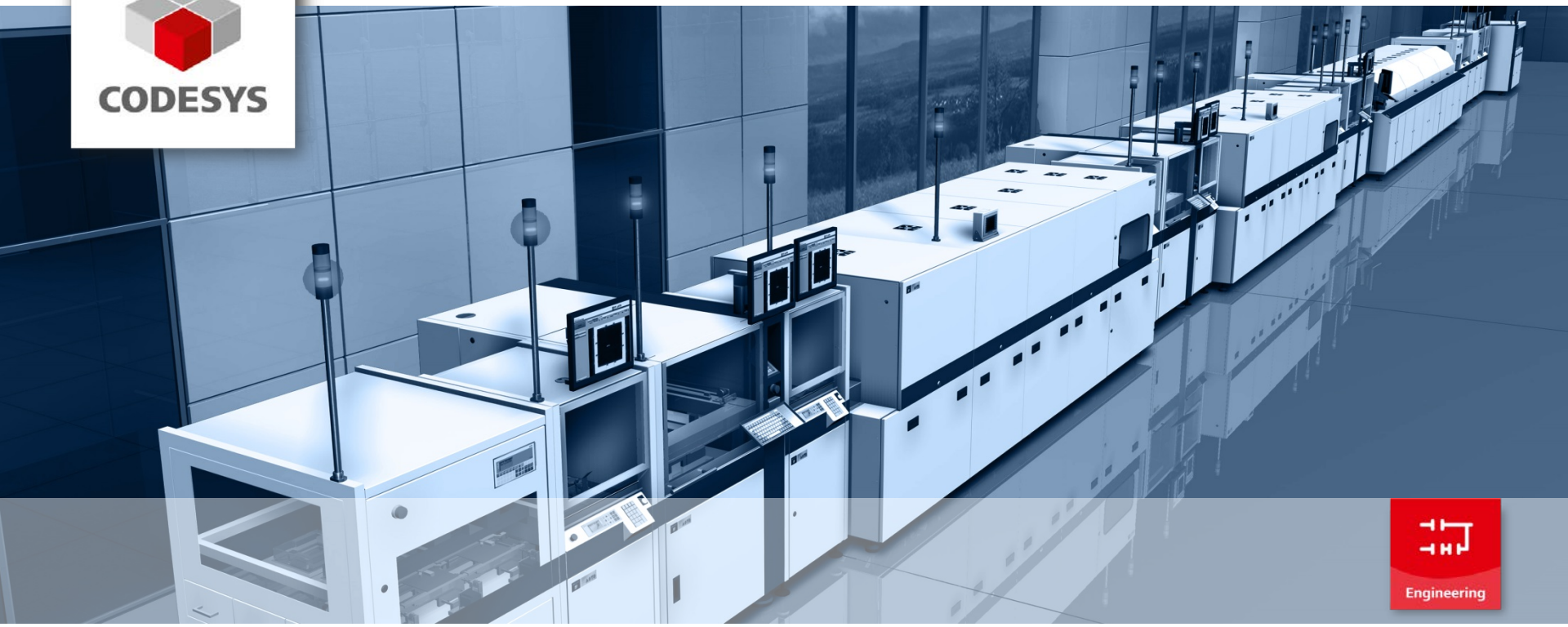


The CODESYS Application Composer: From module creation to plant engineering

CODESYS Users‘ Conference 2014, Dieter Hess

Agenda

1. What is the CODESYS Application Composer?
2. How to use the CODESYS Application Composer
3. Demo

What is the CODESYS Application Composer?

- The CODESYS Application Composer is a development tool for efficient creation of applications consisting of predefined function blocks.

- The CODESYS Application Composer allows to engineer and parameterize complete controller applications out of previously created software modules.
- It is possible to create a whole working application, without programming.

What is the CODESYS Application Composer?

In detail...

- Two aspects of application engineering:
  - Development of technology modules
    (e.g. control algorithms, motion control etc.)

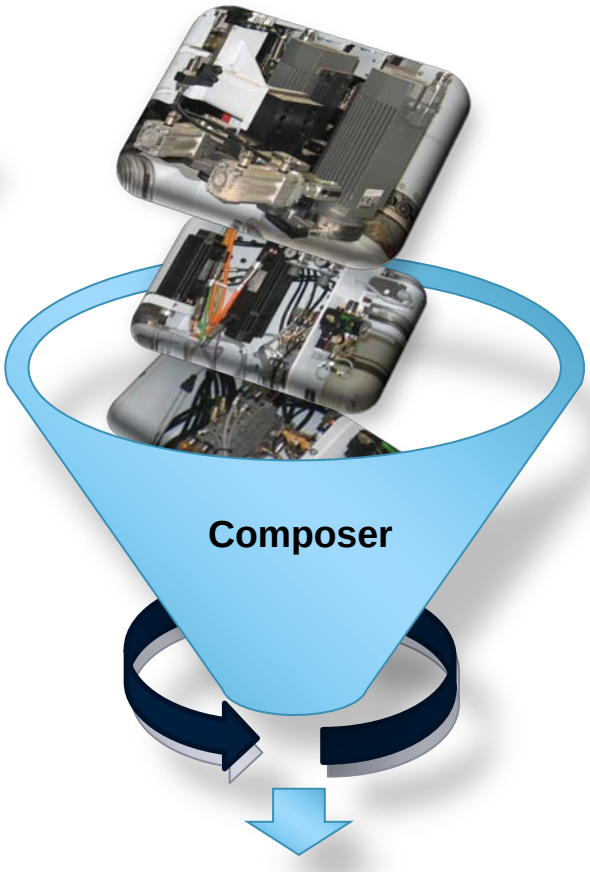
    => Demanding, innovative development work

  - Machine engineering (“Composing”)
    - Selection and parameterization of technology modules
    - I/O assignment
    - Distribution to controllers
    - Standard infrastructure (diagnosis, alarms etc.)

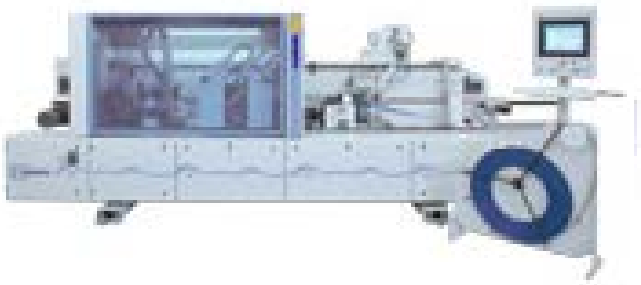
    => Routine tasks with high rationalization potential

What is the CODESYS Application Composer?

Technology modules



Machine variant X



What is the CODESYS Application Composer?

In which form is the CODESYS Application Composer supplied?

- In the CODESYS Store, you can purchase a CODESYS Package along with a ticket allowing for the installation of the license key on a Wibu dongle.

- With this workplace license, you can create and use modules.

Agenda

1. What is the CODESYS Application Composer?
2. How to use the CODESYS Application Composer
3. Demo

How to use the CODESYS Application Composer

Example: Gas station

How to use the CODESYS Application Composer

Units of a gas station

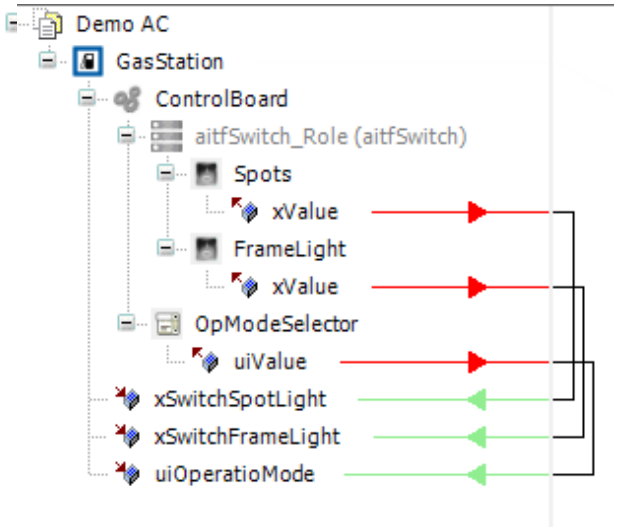
How to use the CODESYS Application Composer

Situation

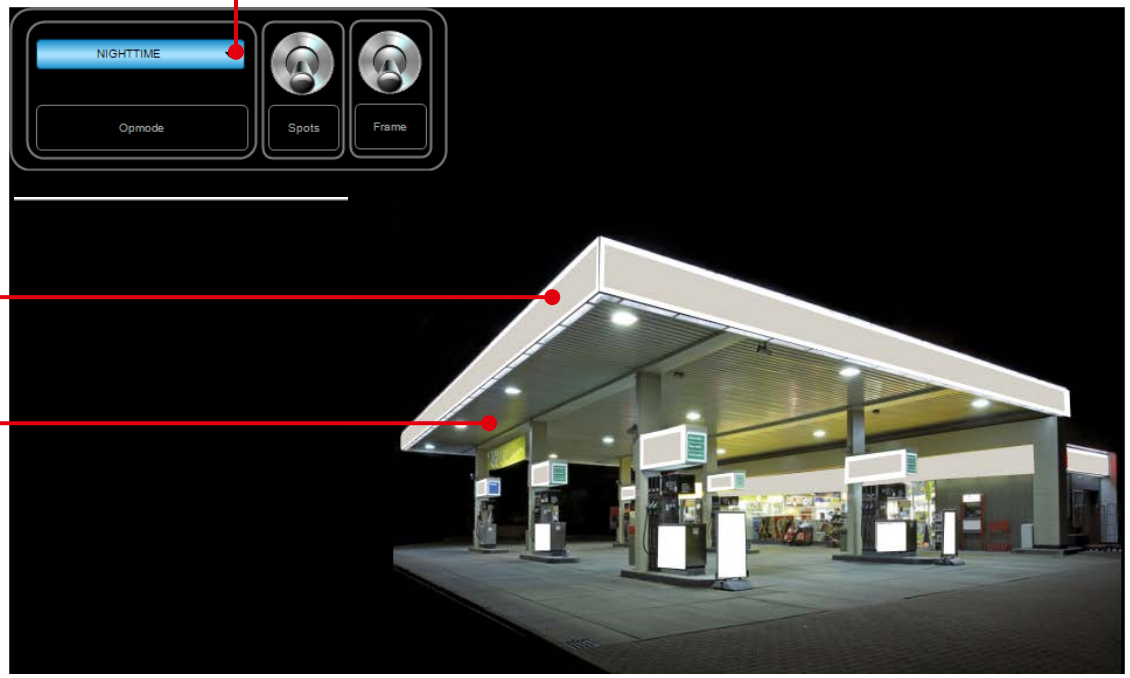
- As a system supplier, you have a wide product variety.
- You offer various configuration options and your customers' wishes can be realized at very short notice - even during commissioning.
- Even after many years, phased-out products should not cause any problems. You need to support new, compatible devices.
- Your service staff must engineer new plants or maintain existing plants with only little training.

How to use the CODESYS Application Composer

Development gas station - control center



Control center

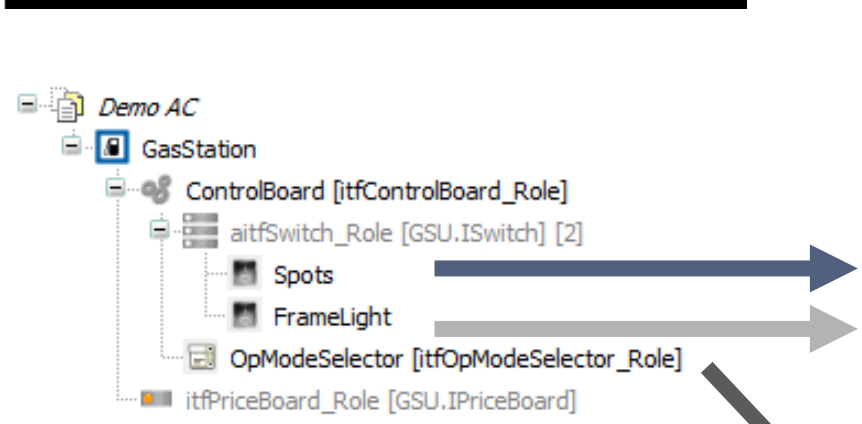


Front illumination

Spots

How to use the CODESYS Application Composer

Development gas station - control center



GasStation.ControlBoard.Spots

Parameter | E/A | HMI | Information

Parameter	Typ	Wert	Beschreibung	> Min	< Max
sID_Name	STRING(25)	**'Spots'**	sID_Desc		

GasStation.ControlBoard.FrameLight

Parameter | E/A | HMI | Information

Parameter	Typ	Wert	Beschreibung	> Min	< Max
sID_Name	STRING(25)	**'Frame'**	sID_Desc		

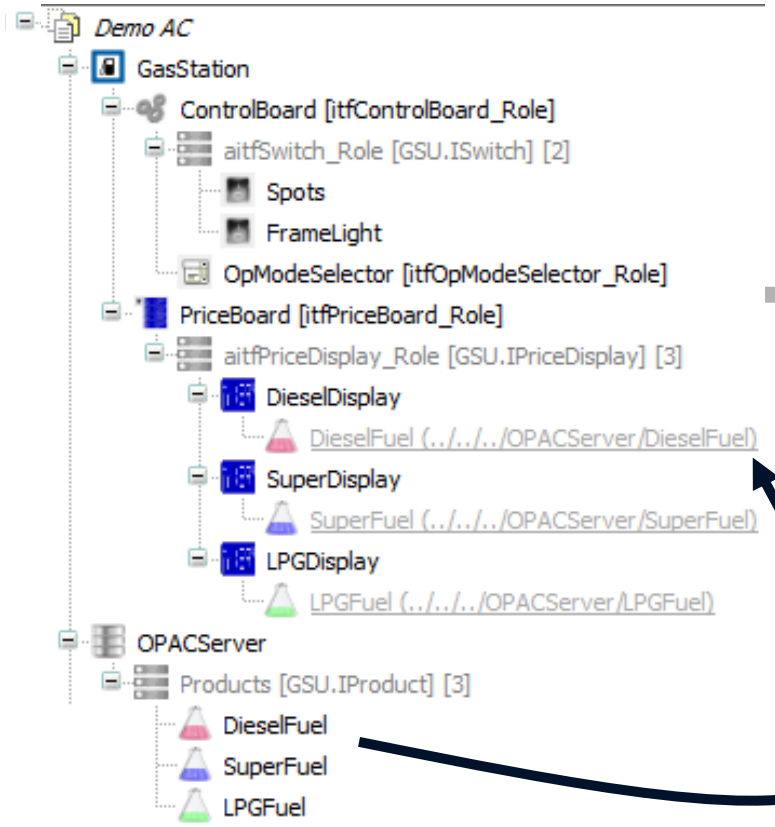
GasStation.ControlBoard.OpModeSelector

Parameter | E/A | HMI | Information

Parameter	Typ	Wert	Beschreibung	> Min	< Max
sID_Name	STRING(25)	**'Opmode'**	sID_Desc		

How to use the CODESYS Application Composer

Development gas station

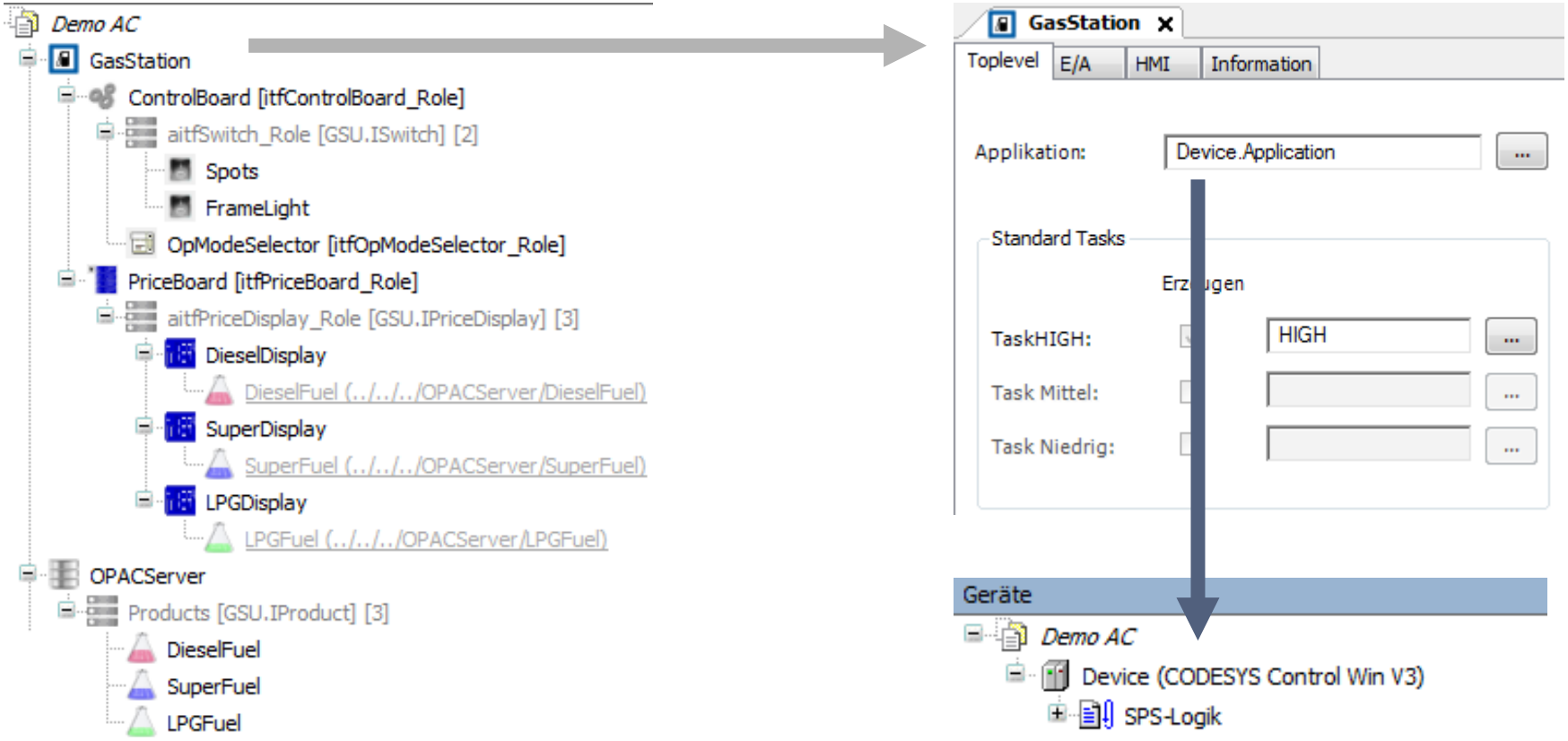


Drag and Drop

How to use the CODESYS Application Composer

Development gas station

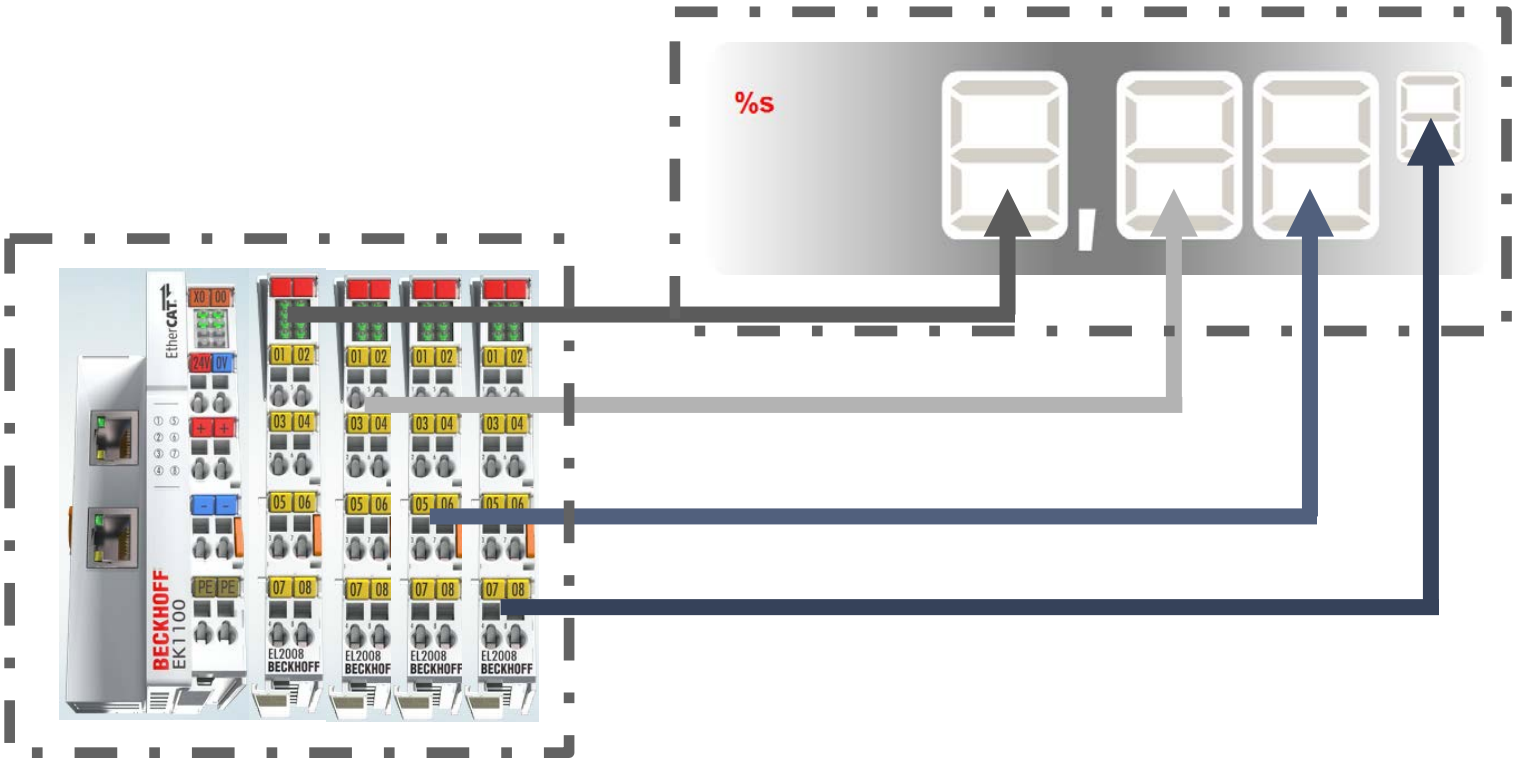
- In the top level object you decide on which controller the code will be processed.



How to use the CODESYS Application Composer

The device configuration including I/O mapping can be generated automatically.

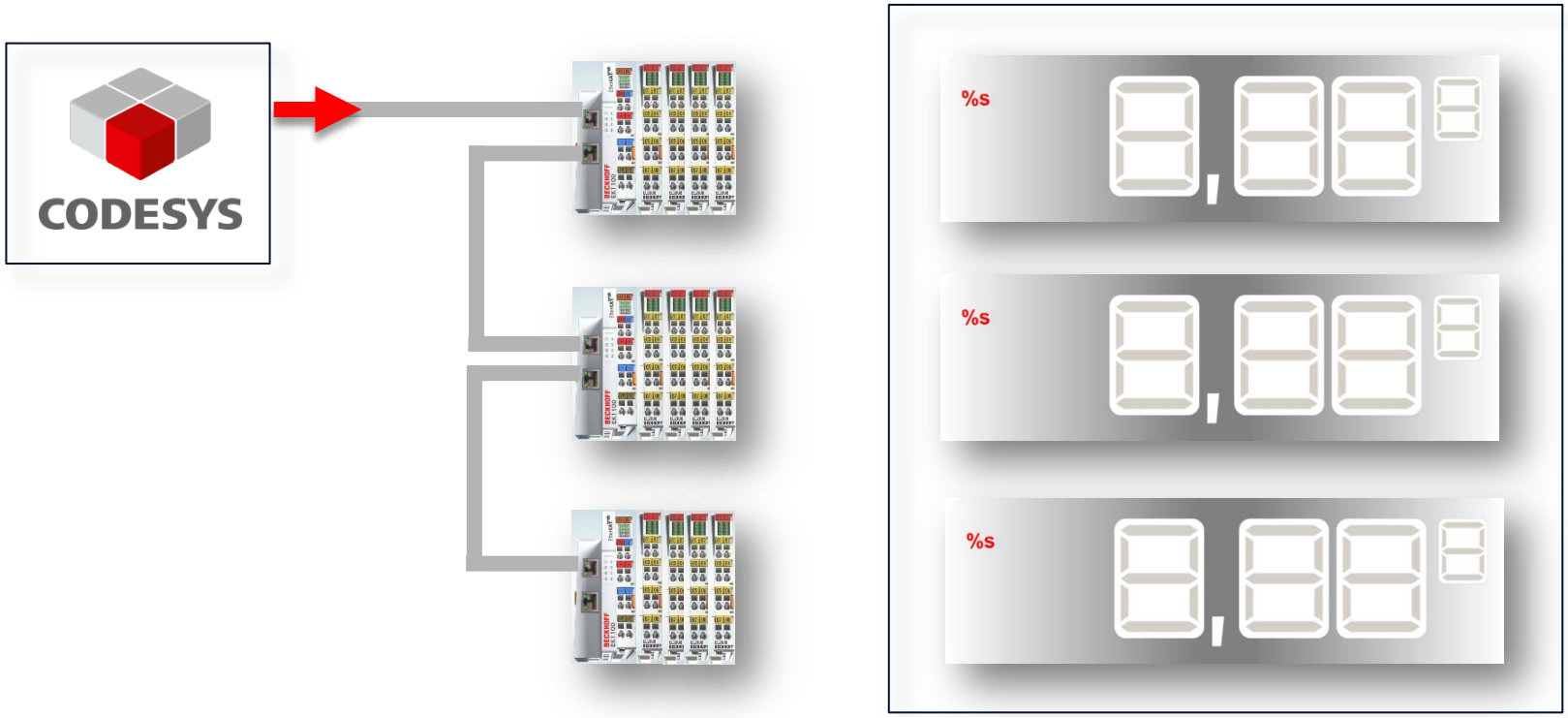
- Each display on the display element could be realized using an EtherCAT slave EK1100 + an EL2008 clamp per digit.



How to use the CODESYS Application Composer

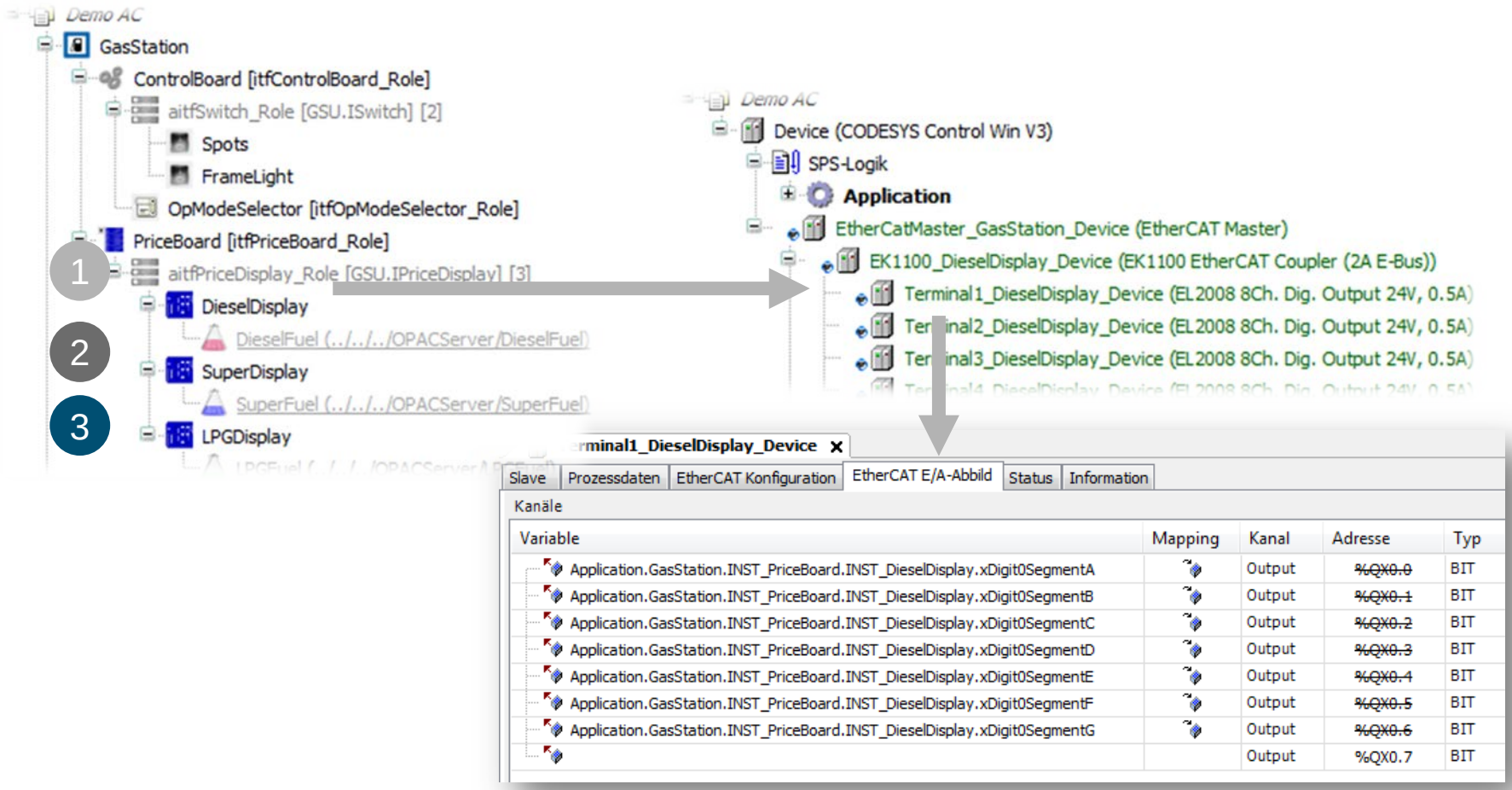
Three displays require a higher configuration effort.

- The Composer can take over this task for you.



How to use the CODESYS Application Composer

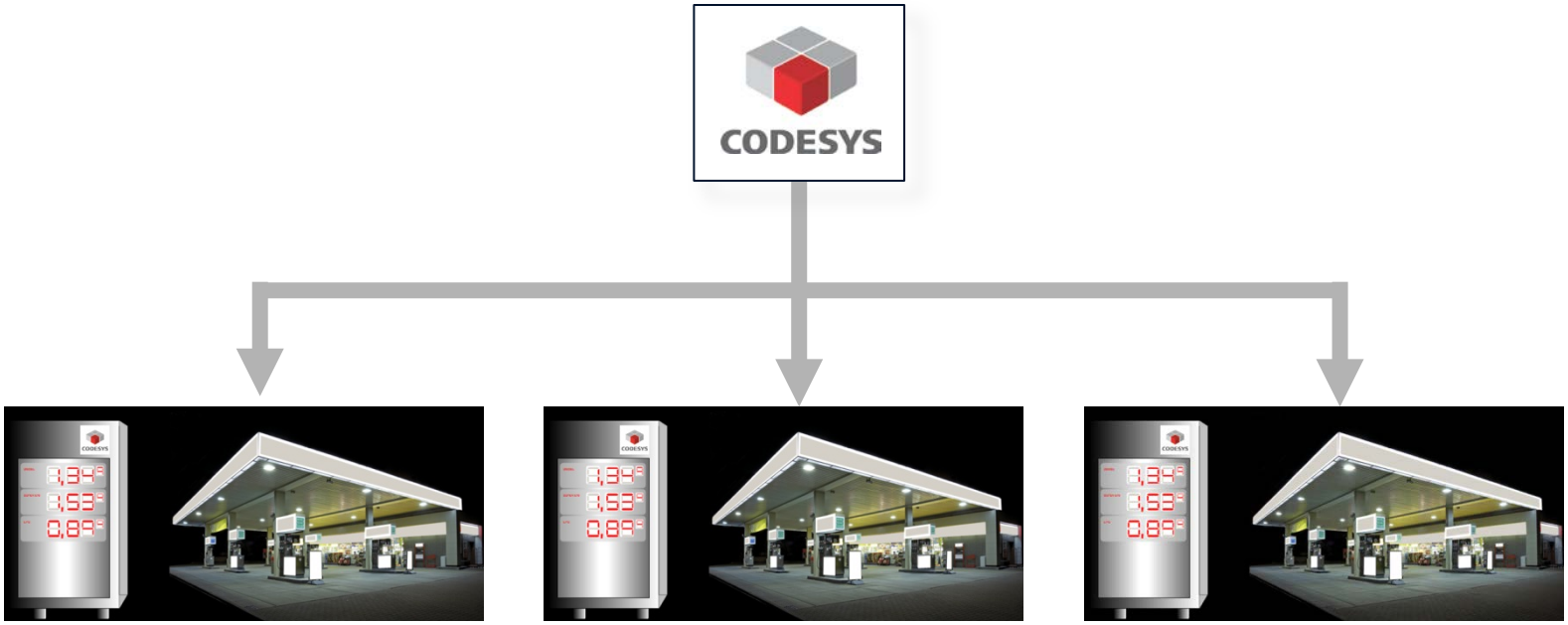
The Composer is ideal for recurring configuration tasks.



How to use the CODESYS Application Composer

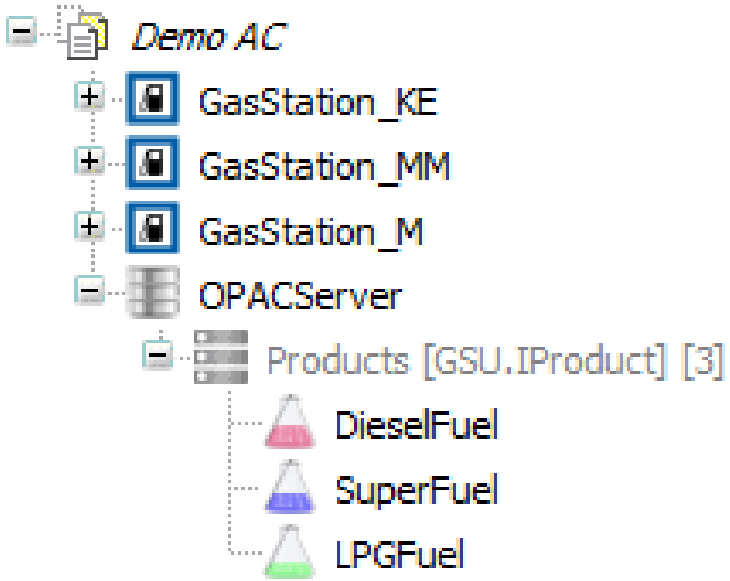
You receive a new order.

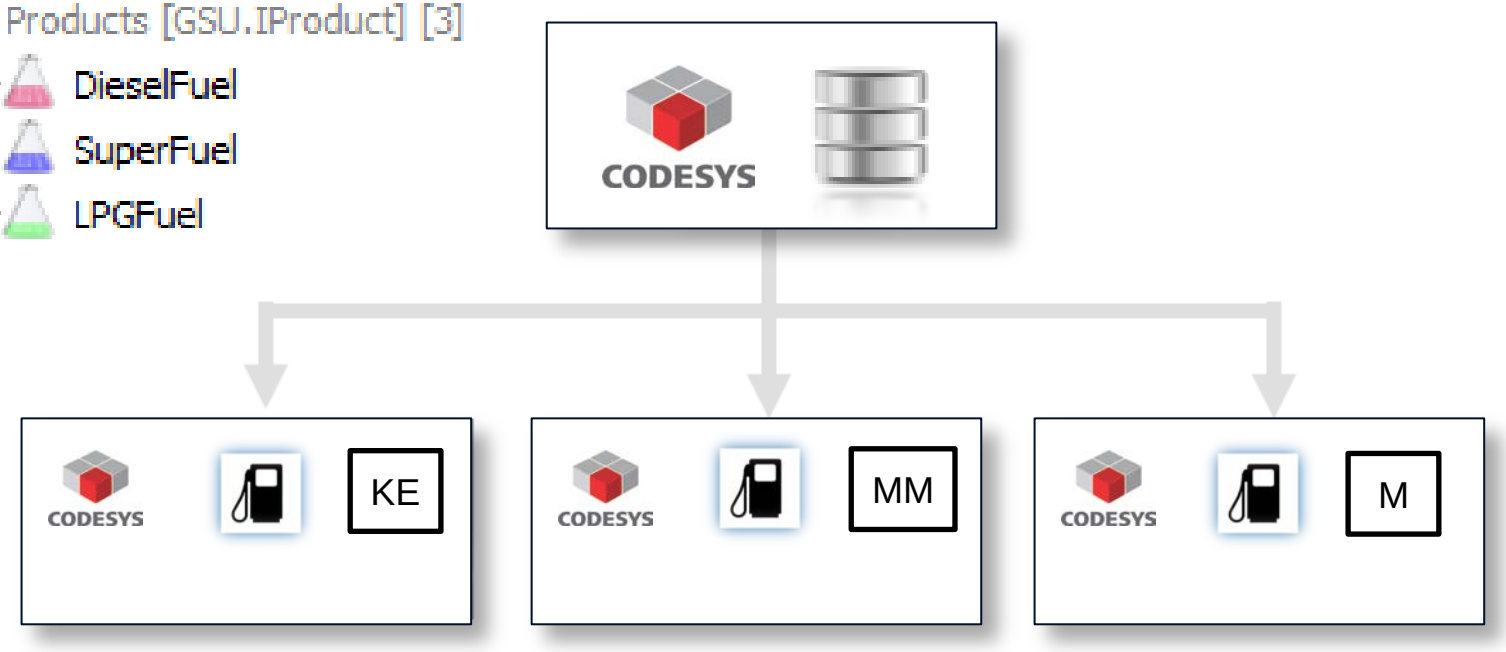
- The service station operator has meanwhile expanded his network.
- You delivered gas stations in different configurations.
- Price displays are to be controlled from the headquarters.



How to use the CODESYS Application Composer

Which changes have to be made in the project?





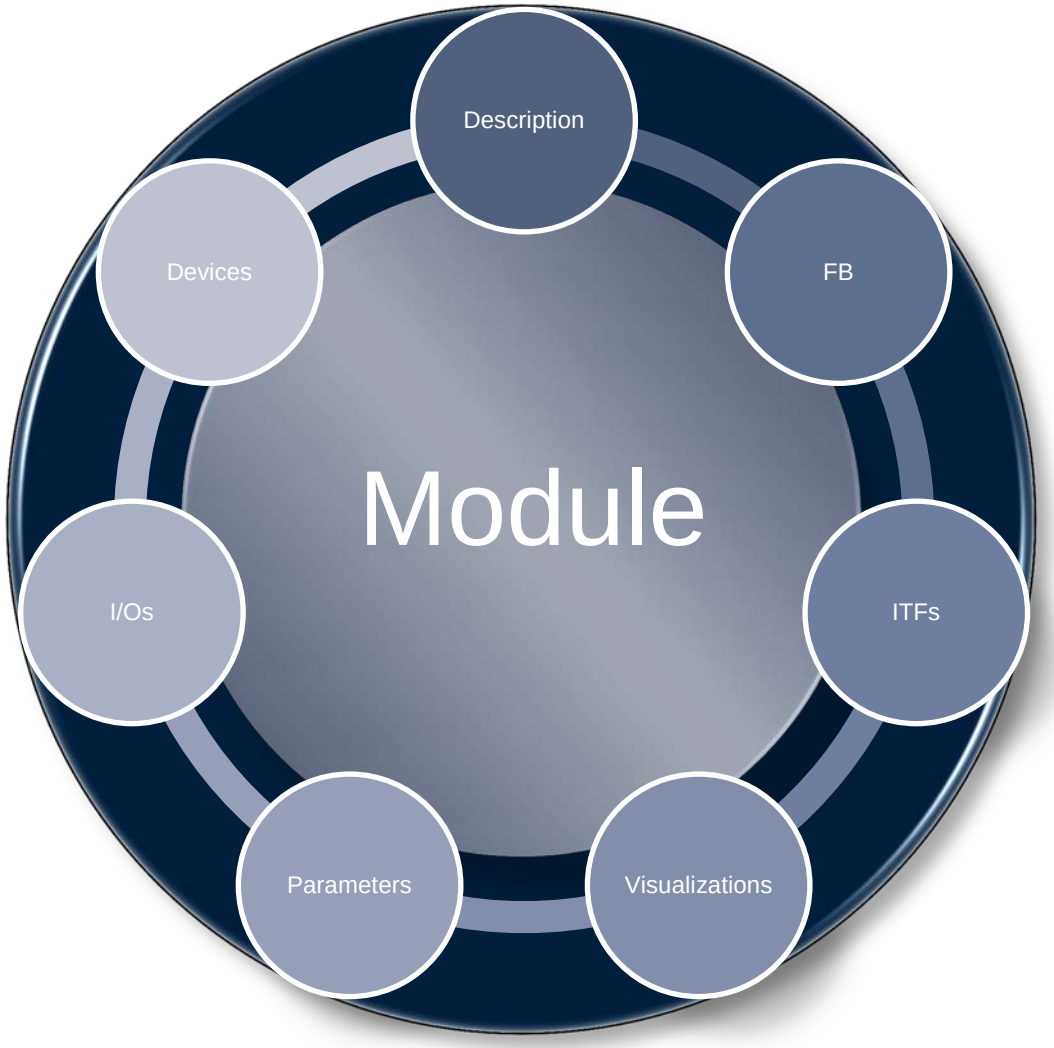
How to use the CODESYS Application Composer

Distribution of tasks to controllers

- With CODESYS, you can use controllers of different manufacturers in one project.
- Over 350 device manufacturers on the market use the CODESYS Runtime System.
- These devices can easily be connected to each other, and there is a communication interface even to external products.

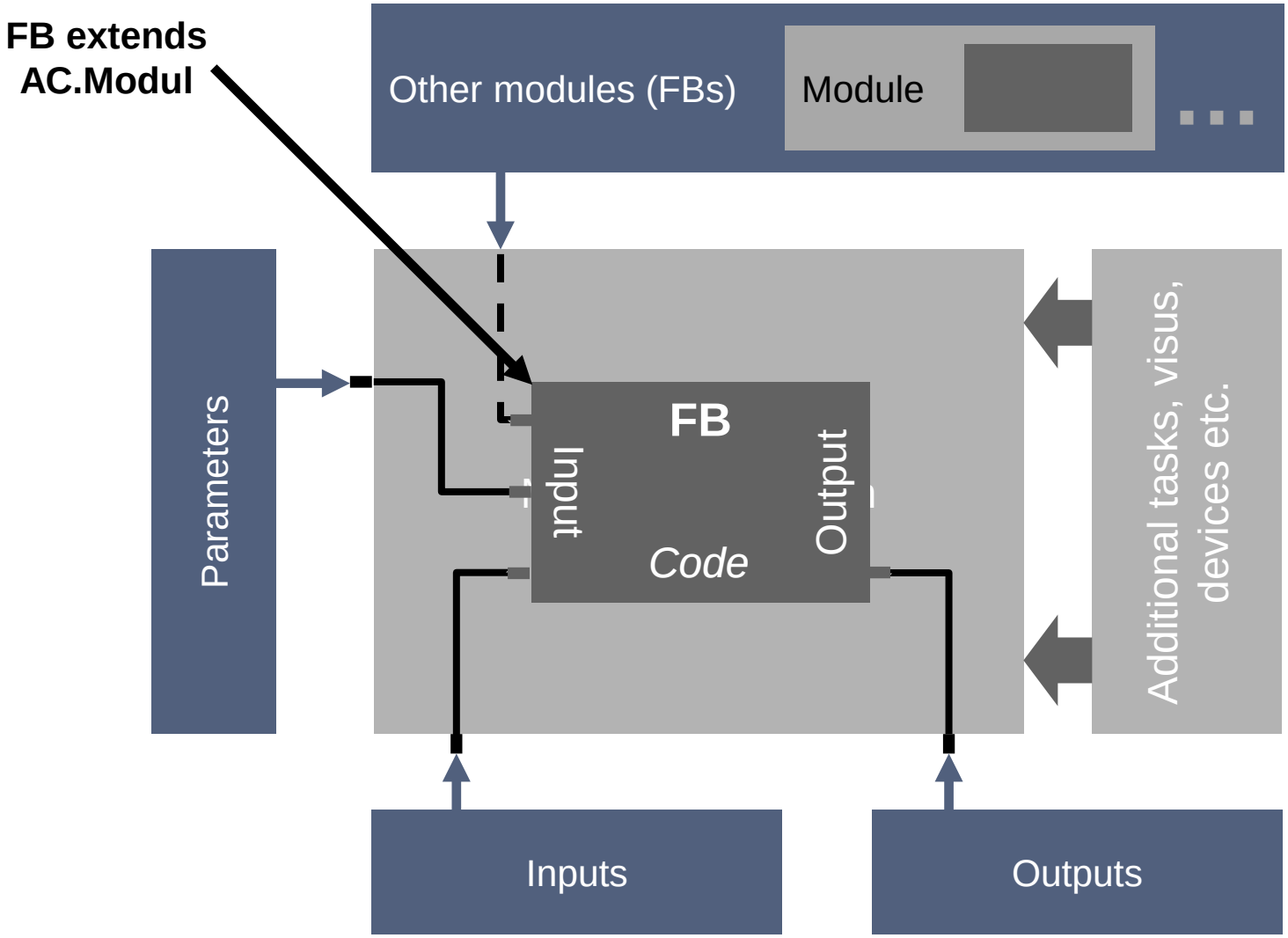
How to use the CODESYS Application Composer

How are modules created?



How to use the CODESYS Application Composer

Module implementation



How to use the CODESYS Application Composer

Let's look at the module "GasStation".

GasStation

```
MODULE GasStation IMPLEMENTED_BY GasStation
SEC std.MetaData
    Desc        := TL.GasStation_Desc ;
    Category    := 'Misc' ;
    Icon_16     := ImagePool.GasStation_16 ;
    Icon_32     := ImagePool.GasStation_32 ;
END_SEC
SEC Toplevel
    SEC STANDARD_TASK : HIGH [3 lines]
    END_SEC
    GVL_NAME := 'GVL_%InstanceName%' ;
END_SEC
SEC Io
    SEC Input : xSwitchSpotLight [3 lines]
    END_SEC
    SEC Input : xSwitchFrameLight [3 lines]
    END_SEC
    SEC Input : uiOperatioMode [3 lines]
    END_SEC
END_SEC
SEC Slots
    SEC Slot : itfControlBoard [5 lines]
    END_SEC
    SEC Slot : itfPriceBoard [5 lines]
    END_SEC
END_SEC
SEC std.Visu [2 lines]
END_SEC
```

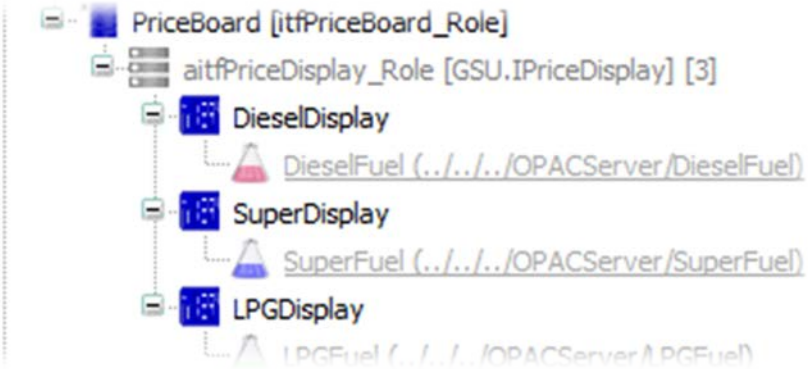
GasStation

```
FUNCTION_BLOCK GasStation EXTENDS AC.Module
VAR_INPUT
    xSwitchSpotLight : BOOL;
    xSwitchFrameLight : BOOL;
    uiOperatioMode : UINT;
    itfControlBoard : IControlBoard;
    itfPriceBoard : IPriceBoard;
END_VAR
VAR_OUTPUT
    xFrameLight : BOOL;
    xSpotlight : BOOL;
END_VAR
```

How to use the CODESYS Application Composer

Now let's take a look at the "PriceBoard".

- In the configuration phase, you define the number of displays.

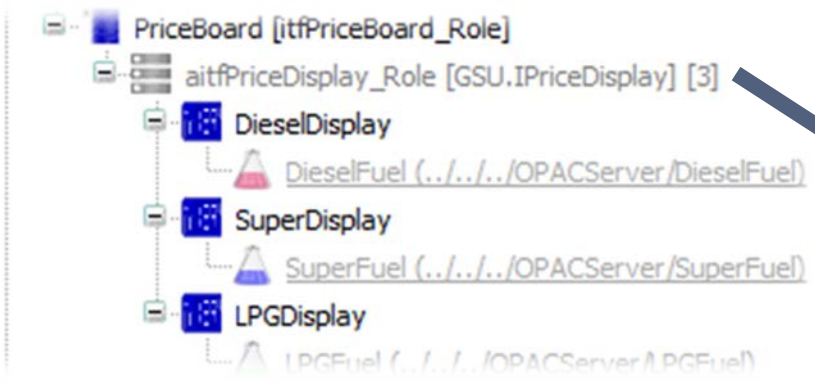


- In the previous presentation, the number of displays was programmed in CFC.
  - A complex management mechanism had to be implemented.

=> The CODESYS Application Composer already covers these management mechanisms.

How to use the CODESYS Application Composer

The necessary code is generated automatically.



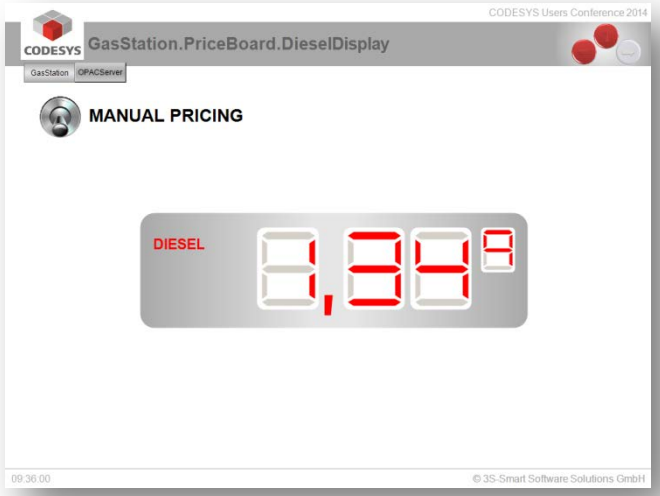

```
MODULE PriceBoard IMPLEMENTED_BY PriceBoard
SEC std.MetaData
    Desc        := TL.PriceBoard_Desc ;
    Category    := 'Misc' ;
    Icon_16     := ImagePool.PriceBoard_16 ;
    Icon_32     := ImagePool.PriceBoard_32 ;
END_SEC
SEC Slots
    SEC Slot_Multi : aitfPriceDisplay
        Variable := aitfPriceDisplay;
        Var_Count := uiNumberOfPriceDisplay;
        Role := TL.aitfPriceDisplay_Role;
        Cardinality := [0 .. 5];
        Type := SUBMODULE;
        Name := TL.aitfPriceDisplay_Name;
        Desc := TL.aitfPriceDisplay_Desc;
    END_SEC
END_SEC
```

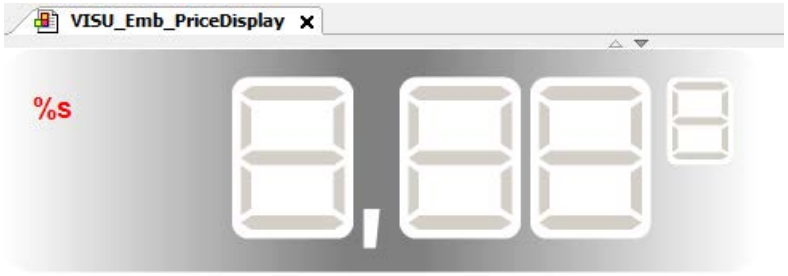
```
FUNCTION_BLOCK PriceBoard EXTENDS AC.Module IMPLEMENTS IPriceBoard, IOperation
VAR_INPUT
    itfOPACServer : IOPACServer;
    aitfPriceDisplay : POINTER TO IPriceDisplay;
    uiNumberOfPriceDisplay : UINT := 0;
END_VAR
```

How to use the CODESYS Application Composer

We distinguish between two forms of visualization:

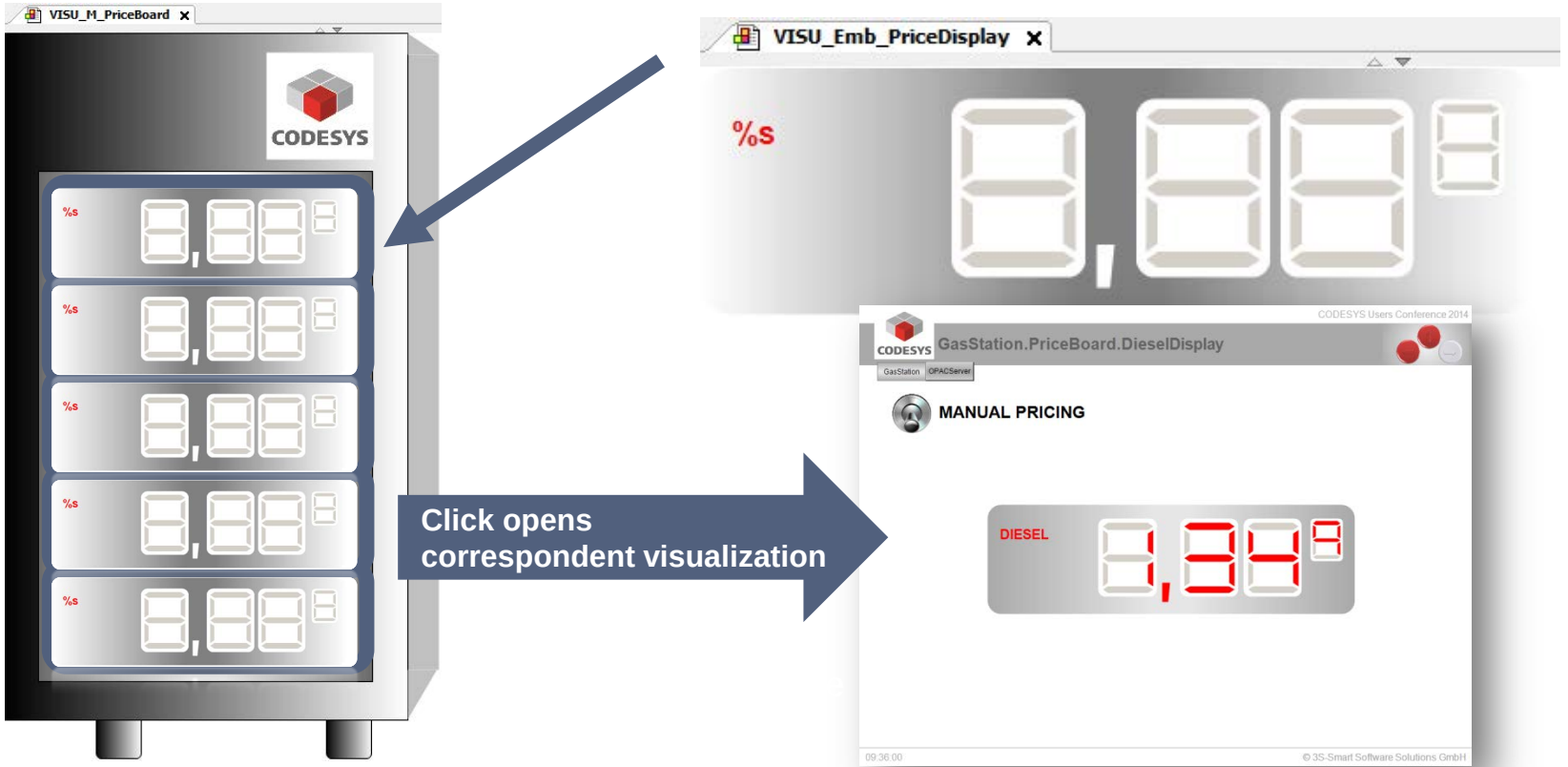
```
SEC std.Visu
    Page := [ VISU_M_PriceDisplaySetValue ] ;
    Embedded := [ VISU_Emb_PriceDisplay ] ;
END_SEC
```





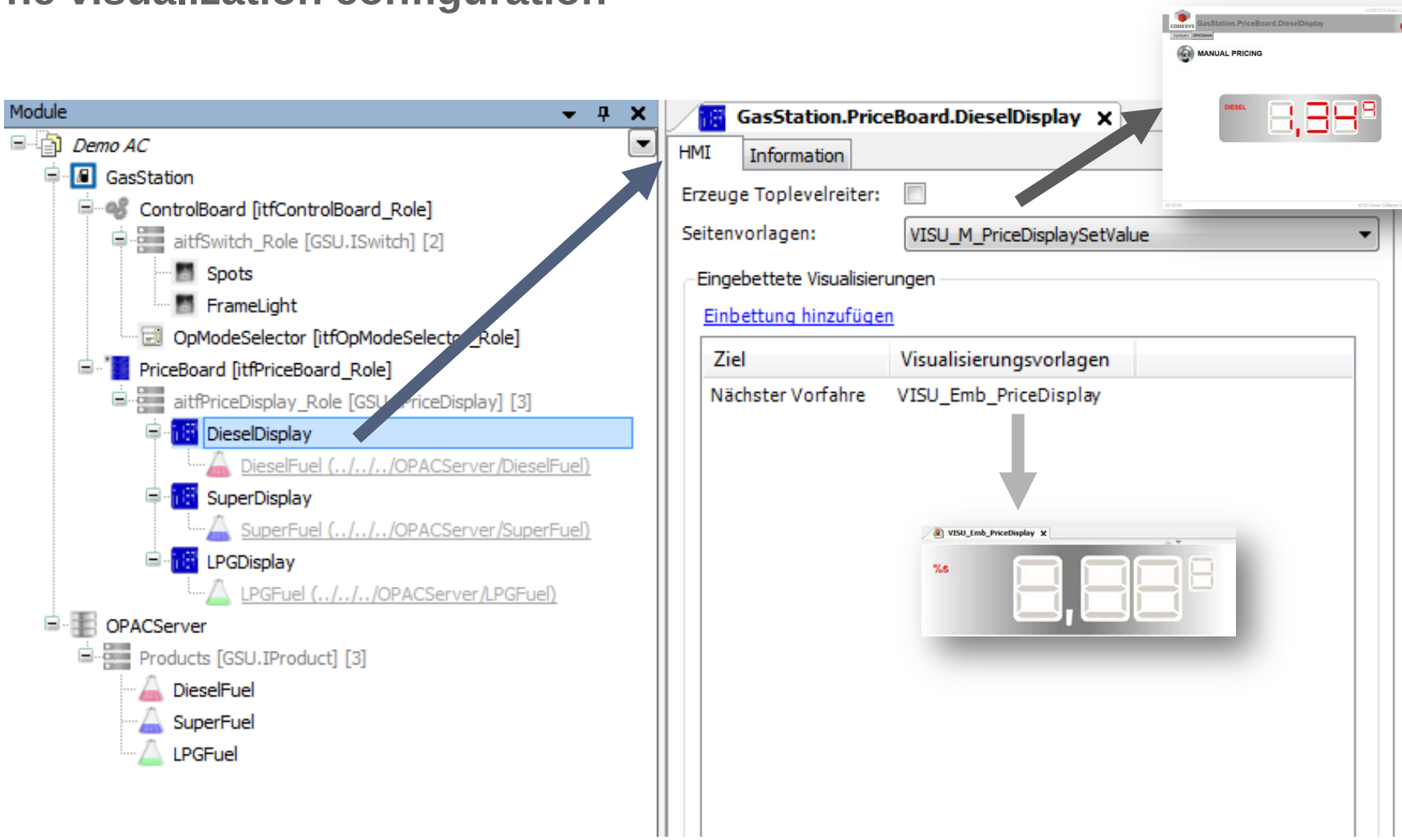
How to use the CODESYS Application Composer

Combination of visualizations



How to use the CODESYS Application Composer

The visualization configuration



Agenda

1. What is the CODESYS Application Composer?
2. How to use the CODESYS Application Composer
3. Demo

Thank you for your attention.