CODESYS FOR BASIC FESTO MOTION TERMINAL (VTEM) APPLICATION

Jana Koubová[1], Marie Martinásková[2]

[1]CTU Prague, Jana.Koubova@fs.cvut.cz
[2]CTU Prague, Marie.Martinaskova@fs.cvut.cz

Abstract:
This paper describes how to use CODESYS software environment for programming of a simple application using Festo Motion Terminal (FMT) - VTEM. This application consists of controlling the movement of one pneumatic actuator with proximity sensors using visualization in the CODESYS software environment. Firstly the main aspects of Festo Motion Terminal - VTEM are described. Secondly the prerequisites for use are described such as installation process of the software, packages and libraries needed for creating an application compatible with VTEM. At last the whole applications composition is described.

Keywords:
valve terminal, pneumatic actuator, CODESYS, Festo Motion Terminal, FMT, VTEM, structured text, , programmable logic controller, proximity switch, electropneumatics, proportional pneumatics, digital pneumatics

1. Introduction

Valve terminals are widely used as a mean to use pneumatics efficiently in automation. So far the valve terminals consisted of specific valves configured together in a cluster for one specific application. Festo Motion Terminal (FMT) introduces a universal valve VEVM, which can switch its inside configuration and thus perform function of different valves and components. [1] Festo Motion Terminal is a valve terminal designed to uphold the standards of Industry 4.0 containing 4 or 8 VEVM valves with a controller module CTMM, to which other controller modules can be connected. In our application the higher control module is CPX-CEC-V1-S3. The application for CPX module is created using CODESYS. CODESYS is development environment for programming of control applications in which all 5 programming languages recommended in IEC/EN 61131-3 for programmable logic controllers are available: Ladder Diagram (LD), Function Block Diagram (FBD), Structured Text (ST), Instruction List (IL), Sequential Function Chart (SFC). Additionally it includes Continuous Function Chart (CFC) which is evolution of FBD, which allows multiple function blocks to be connected. In this application ST and CFC were used.

Application shown in this paper will allow for manual and automatic (sequence) control of one pneumatic actuator with two end position proximity sensors. This document can serve as a guideline for the necessary software installation and creation of code for basic FMT application.

2. Specification of Festo Motion Terminal (FMT)

Structure

FMT design

Conceptually we can divide the FMT into 3 parts: control, pneumatic and input modules. Higher control is provided by CPX module (PLC), which is connected to CTMM controller. The CTMM provides communication directly for FMT terminal (the valves) and also CTMM is where the main air supply with fitting for tubing and main exhaust port with silencer are located. Single universal valves and CTMM are connected pneumatically and electrically via the manifold rail VABM. The manifold also connects CTMM and input modules electrically. On top of the manifold there is certain number of connection ports. Into these ports the valves and input modules are fitted. Leftover ports are covered with cover plates. Unused ports are covered with blanking plugs. [1]

1	CPX modules	CPX
2	Controller	CTMM
3	Valve body	VEVM
4	Identification holder	ASCF
5	Cover plate	VABB
6	Input module	CTMM
7	Cover cap	ISK
8	Connecting cable	NEBU
9	Position sensor	SDAP
10	Blanking plug	B
11	Fittings	QS
12	Silencers	U
13	Manifold rail	VABM
14	H-rail mounting	VAME

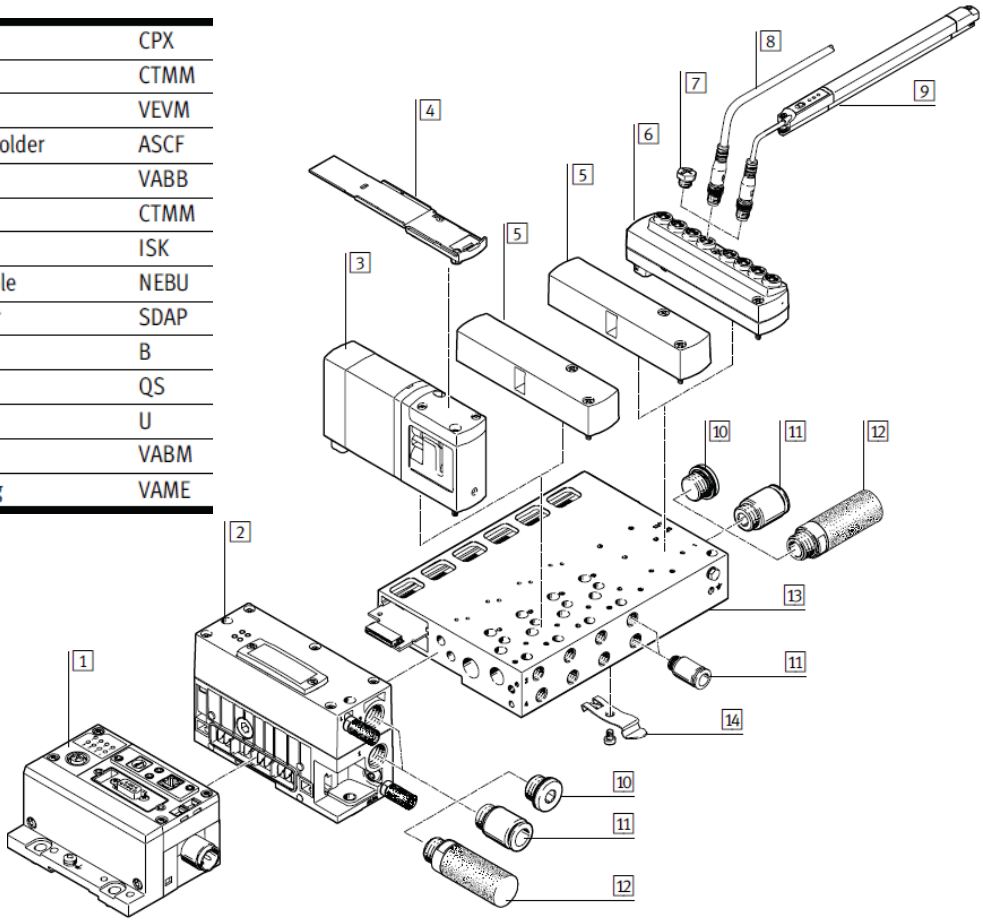


Fig. 1 Festo Motion Terminal – Peripherals overview [1]

VEVM universal valve

The main concept difference between a standard valve terminal and VTEM terminal are the valves. All the universal VEVM valves are the same in build, but they can substitute different components based on their inner parts alignment changing according to the running program. Thus instead of needing to change components for different application, it is only necessary to change the program. It is also not necessary to store multiple spare parts for different types of valves.

Each valve comprises of four 2/2-way proportional valves positioned in bridge circuit (Fig. 2) with integrated sensors. Each 2/2-way valve has 2 piezo pilot valves which are connected to diaphragm poppet valve. The bridge configuration enables for each port to be set in one of following states: open, closed, pressurized, exhausted. By setting the states on port 2 and 4 the valve can simulate function of different components. [2][3]

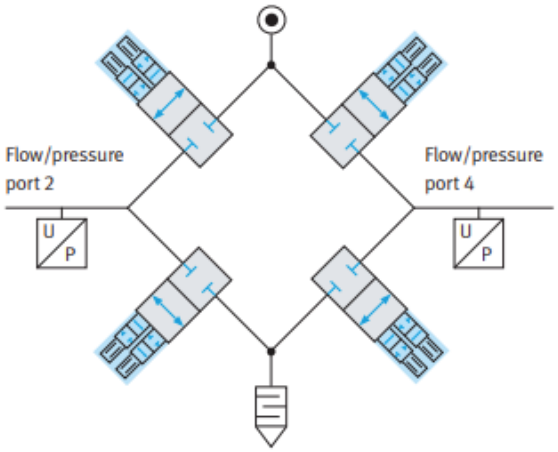


Fig. 2 Bridge circuit in the VEVM valve [3]

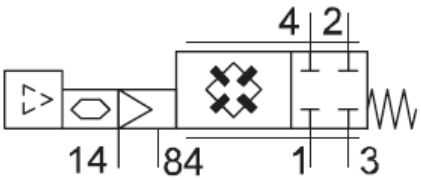


Fig. 3 Circuit symbol for VEVM valve [3]

2.1 Functionality

Motion Apps

The user can assign one Festo Motion App to each valve through the program at one time moment. Through the app, parameters for the function of the valve can be set. Festo Motion Apps are preprogrammed within the VTEM terminal. Through PLC program or in the Web Interface Festo Motion Apps and their parameters are assigned to valves. [2]

Web Interface

There is an option to connect the CTMM module directly to a computer via Ethernet connection. This lets the user interact with CTMM via Web Interface. This option is especially useful for understanding all the Festo Motion Applications and their parameters. In user-friendly environment, apps can be assigned to the valves and their parameters set. For the connection to be successful a specific IP address must be set on the computer to be able to see the terminal. The terminal controller CTMM IP address is then inserted into web browser (Mozilla Firefox is the recommended browser) address line and the interface is loaded – in factory setting the IP address is 192.168.4.2. [4]. Before the terminal can be accessed, user needs to sign in with a password (factory setting password is “vtem”). Working in this environment is very intuitive. For the connection to work only the first two numbers of IP address of the computer are given by the controller address, that means that PC address can be any variation of 192.168.x.x.

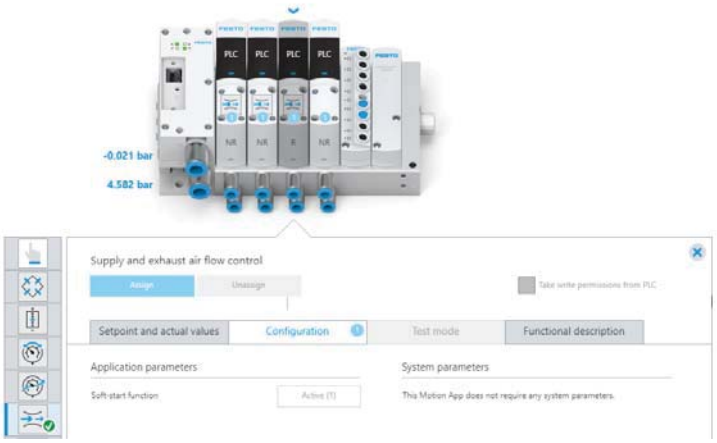


Fig. 4 Web interface for Festo Motion Terminal

PLC module control

Festo Motion Terminal can be controlled by local controller module of the CPX terminal which then communicates with the CTMM controller. In this case it is CPX-CEC-S1-V3. Into CPX the program is downloaded from computer. Programming can be done in software environment CODESYS. There are many CPX communication modules with many possible communication protocols (e.g. CAN, CANopen, Modbus). The technology as a whole has been designed to be in compliance with standards of Industry 4.0. For this project we used Ethernet connection between the CPX module and computer.

2.2 Necessary Software Installation

When installing, it is imperative that the version of the particular software downloaded is compatible with used devices and modules.
For best results installation should be completed in the following order:

a) CODESYS

1) Version of CODESYS software environment needs to be downloaded that corresponds to the higher control block device. For this project it is CODESYS V3.5 SP12 Patch6 pbF which supports CPX controller CPX-CEC-S1-V3. This download can be done from relevant webpage, where the used CODESYS is seen as in Fig. 5.

Relevant webpage for download:
https://www.festo.com/net/cs_cz/SupportPortal/default.aspx?q=codesys&tab=4&type=75#result

Fig. 5 Description of the used version of CODESYS from the Festo web

2) After download the installation is simple using the Setup_CODESYSV35SP12Patch6.exe file located within the downloaded .zip folder, which shall be extracted first prior installation. This installation can be run in default mode.

b) Festo Configuration Tool

Next step is to install the FCT – Festo Configuration Tool – PlugIn.

1) Download the FCT – PlugIn from relevant webpage where it is seen as in Fig.6.

Relevant webpage for download:
https://www.festo.com/net/cs_cz/SupportPortal/InternetSearch.aspx?q=codesys&tab=16&type=70#result

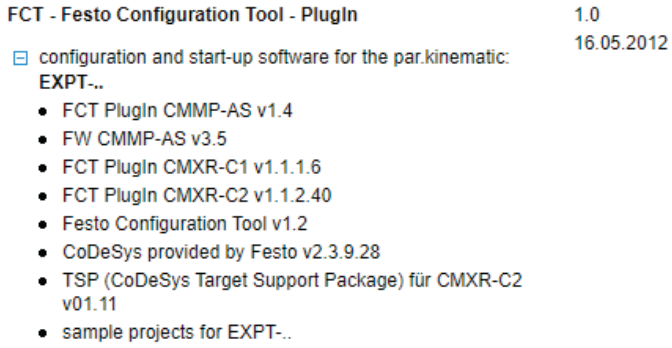


Fig. 6 Description of the used Configuration Tool from the Festo Web

2) After downloading of the .zip file, it is necessary to extract it and then to continue with the installation as described in Readme.txt file. No changes need to be executed in the installation process. When choice is offered chose "Install PlugIns". Last part of instructions in Readme.txt starting with "To work with CoDeSys…" concerning CoDeSys target needn't be finished.

c) Target Support Package

1) First Target Support Package CODESYS needs to be downloaded that corresponds to the version of CODESYS present on the PC and to the used controller block – for this project CPX-CEC-S1-V3 (3472425) without CPX-CTEL-2-M12-5POL-LK. This shall be downloaded from the relevant webpage. At the webpage this package will be seen as in Fig. 7.

Relevant webpage for download:
https://www.festo.com/net/cs_cz/SupportPortal/InternetSearch.aspx?q=codesys&tab=16&type=73#result

Fig. 7 Description of the used Target Support Package

2) This package needs to be then installed in CODESYS in Package Manager located in Tools tab. In Package Manager it is needed to click on "Install" and select the downloaded package from folder. After the installation the new package should be visible in your Package Manager window.

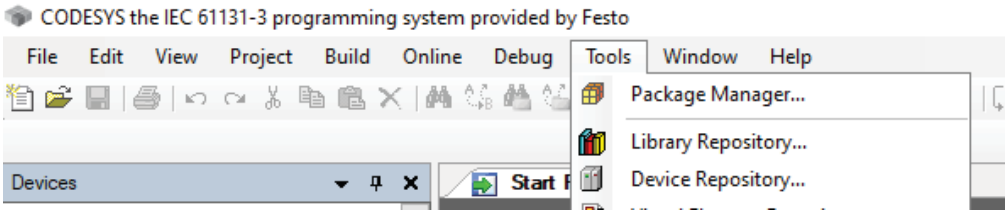


Fig. 8 Location of Package Manager and Library Repository tools in CODESYS software

d) Function Blocks CODESYS

To enable use of VTEM in CODESYS projects, Function blocks CODESYS library needs to be installed. This library has to be installed in CODESYS software in Library Repository.

1) The library needs to be downloaded from relevant webpage where the library is seen as in Fig. 9.

Relevant webpage for download:
https://www.festo.com/net/cs_cz/SupportPortal/InternetSearch.aspx?q=codesys&tab=16&type=74#result

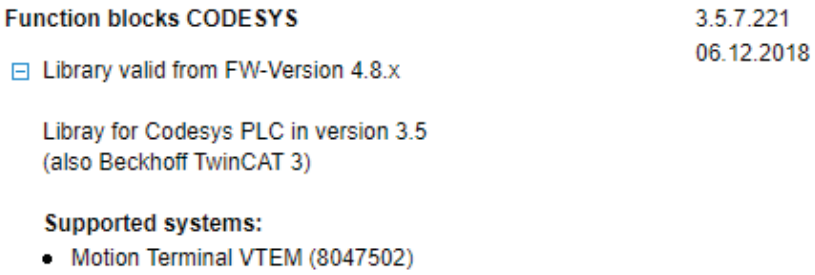


Fig. 9 Description of the used Function blocks CODESYS

2) Next step is to extract the .zip folder. In the folder there is Library subfolder and also Documentation subfolder, which contains additional description of the use of Function blocks. Library Repository is located in Tool same as Package Manager. In Library Repository by clicking "Install", finding and selecting the library file in the folder it will be installed. After installing the Support Package and VTEM Library restart of CODESYS software is recommended. After that the result should be as seen in Fig. 10.

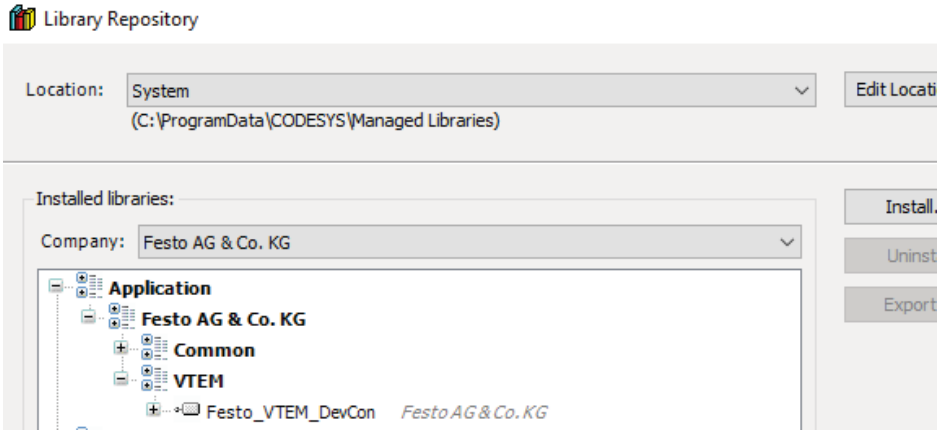


Fig. 10 Festo_VTEM_DevCon library installed in Library Repository

2.3 Connecting the FMT via CPX to computer using Ethernet

Connecting the CPX module Ethernet port with Ethernet port on your computer will allow for applications

to be downloaded to the CPX from CODESYS. Run/stop of the application is possible via position of rotary switch placed on the CPX module or the application can be started and stopped in CODESYS software environment. Proper connection is possible only if computers IP address is set so that it corresponds to the address of the CPX controller. Finding out the CPX IP address in CODESYS using "Scan Festo devices" as shown in Fig.11 is possible. This can be done prior to starting new project as long as the devices are connected in a network.

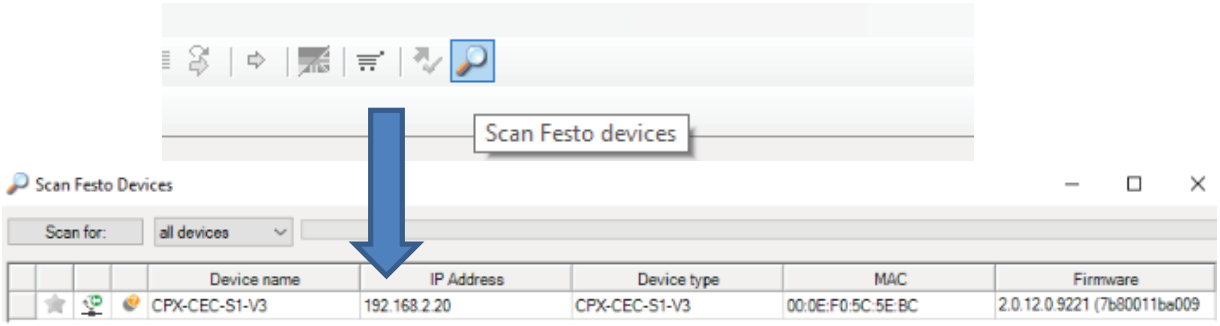


Fig. 11 Finding out the CPX controllers IP address

This means the IP address of the computer should be set to to 192.168.2.x and mask to 255.255.0.0. Last number of the IP address should be different than the one of the controller. One way that this can be done on PC run on Windows is to search in START for Network Connections and open Network Connections Control Panel. In this panel Properties of Ethernet should be opened. There in section Internet Protocol Version 4 (TCP/IPv4) is located – IP address and mask can be changed in its Properties.

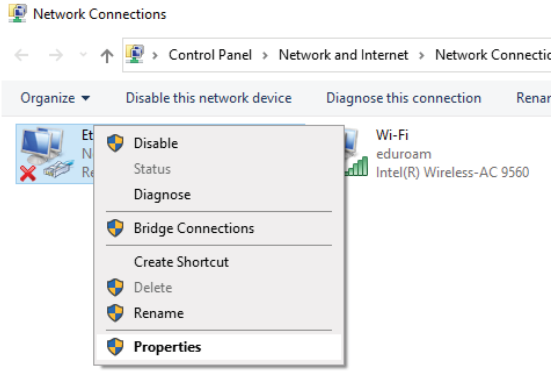


Fig. 12 Changing the IP address of Windows run PC

2.4 Overall project layout

The physical setup of this project is as follows. The FMT is connected to electricity (24V) by special connector cable and to compressed air supply (6 bar input to CTMM module). Components used are from Festo Didactic line. One double acting pneumatic actuator is connected to the first valve position on FMT. This connection is done so that port 2 under VEVM valve is connected to port on the actuator, which when pressurized causes retraction, and port 4 under VEVM valve is connected to port on the actuator, which when pressurized causes advancing as indicated on Fig. 13. This is to the contrary with usual convention, but is used here to be with compliance with logic used in the Web Interface and Apps provided with FMT. There are two 3-pin proximity sensors each at one end position of the piston. These are connected to the first two positions of the input module.

Project layout representation and photo

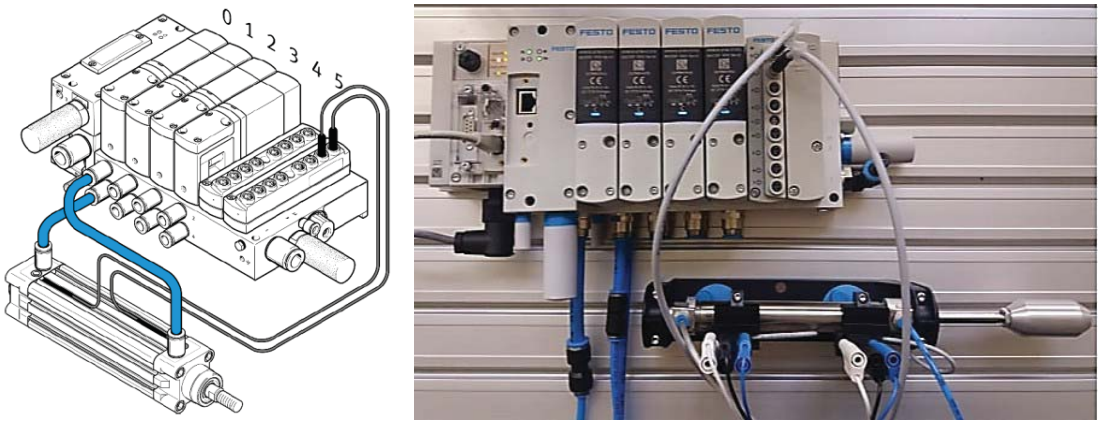


Fig. 13 Left: representation of actuator connection to FMT sourced from [5], right: photo of setup taken in the lab C1-109 CTU FS

3. CODESYS

3.1 Project set-up

New project can be open either from File tab or from the Start Page in CODESYS. After choosing project template (CPX-CEC project) for the type of controller (Fig. 14) it is necessary to make sure that the device set is the same as the one used in the project (Fig.15). In this case CPX-CEC-S1-V3. The language should be set as Structured Text. After that new project with structure as shown on Fig. 16 is formed. Next step is adding the Festo_VTEM_DevCon library to the project library using Library Manager of the project (Fig. 17, Fig. 18).

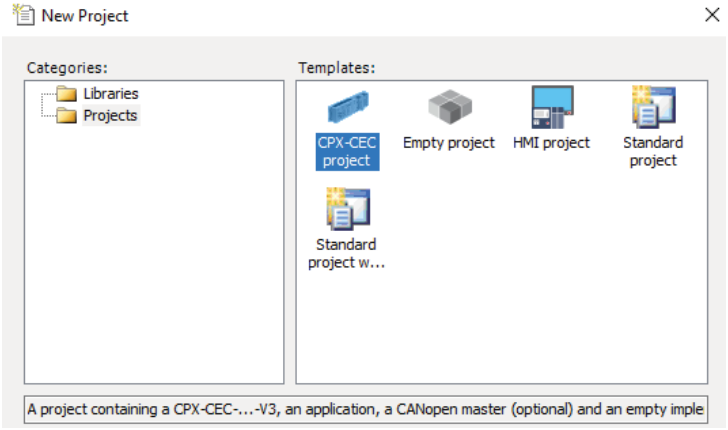


Fig. 14 Choosing the template of the new project

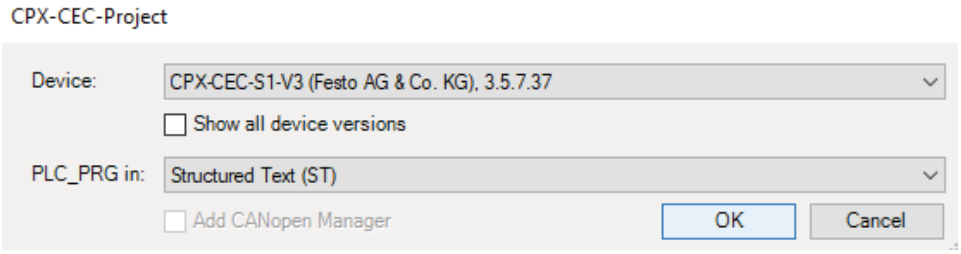


Fig. 15 Choosing device for the project from selection and programming language

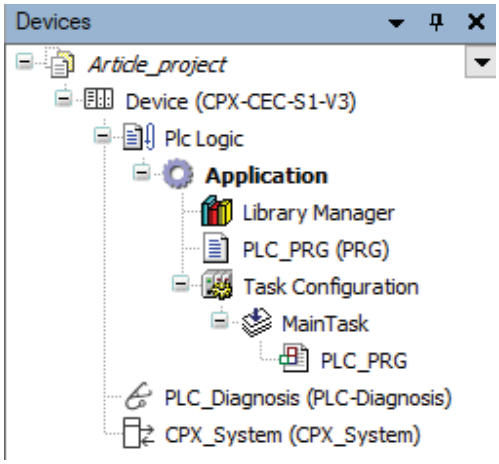


Fig. 16 New projects tree structure

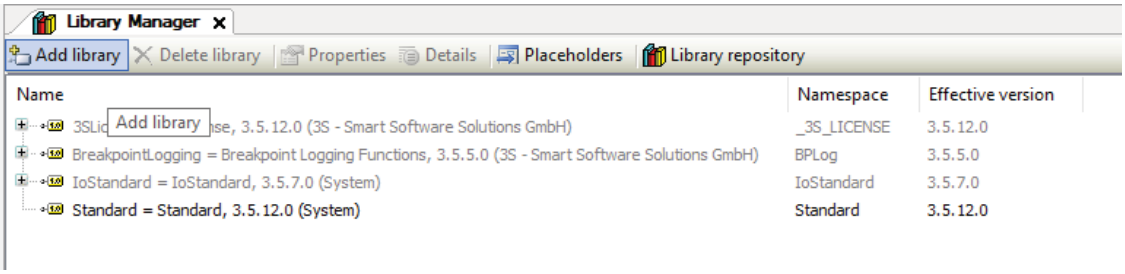


Fig. 17 Library Manager of the project

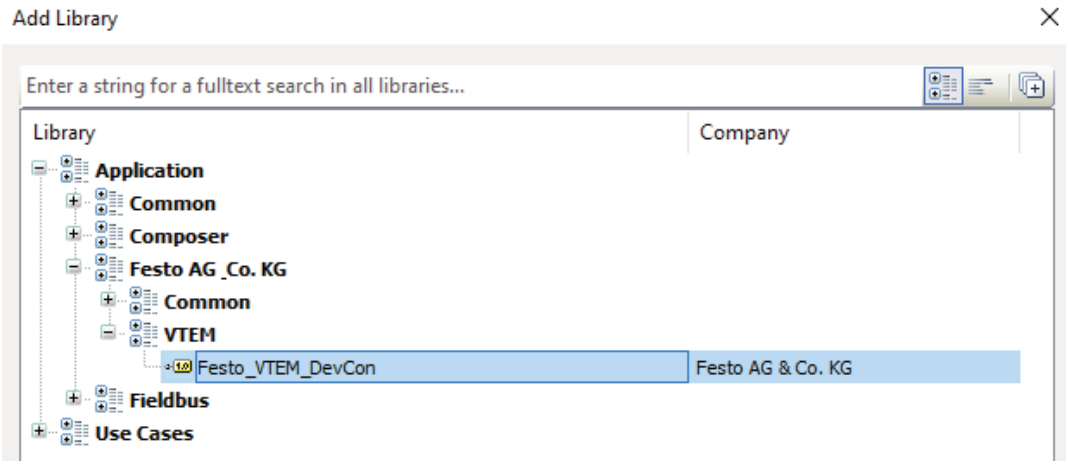


Fig. 18 Adding the Festo_VTEM_DevCon library to the project

3.2 Device set-up and gateway connection

Gateway connection

To establish gateway connection the Device in project tree structure needs to be open. The window that opens is Communication Settings. One way to easily create a gateway is to click on Login in the toolbar (Fig. 19). A following message will appear (Fig. 20). Alternatively the gateways can be added in the Communication Settings window by clicking at "Add gateway…" and then "Scan network" to find the controller block. Fig. 21 shows how active Gateway with device presents. If the PC IP address is set correctly then this automatic Gateway should work in default mode.

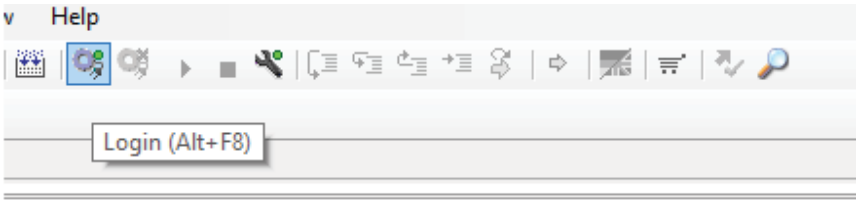


Fig. 19 Location of Login button on upper toolbar in CODESYS

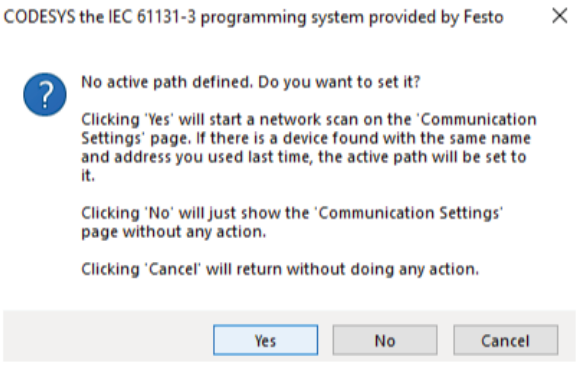


Fig. 20 Message about starting scan and setting pathway

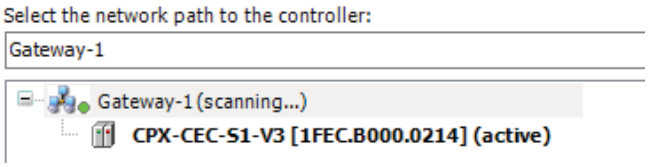


Fig. 21 Active pathway in the Communication Setting window

Configuration scan

In the project tree CPX_System should be opened. In the first folder Module Configuration by clicking on Actual Configuration and performing a Scan (Fig. 22) the actual configuration of modules (Fig. 23) and information about them should appear if your Gateway is working properly. After clicking on the Apply button the specification of the modules will be added to the tree structure of the project (Fig. 24).

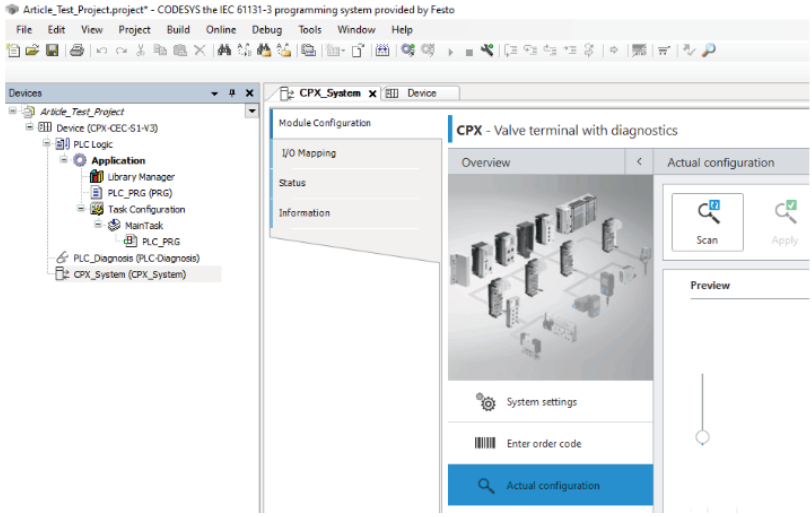


Fig. 22 Scanning for actual configuration of modules

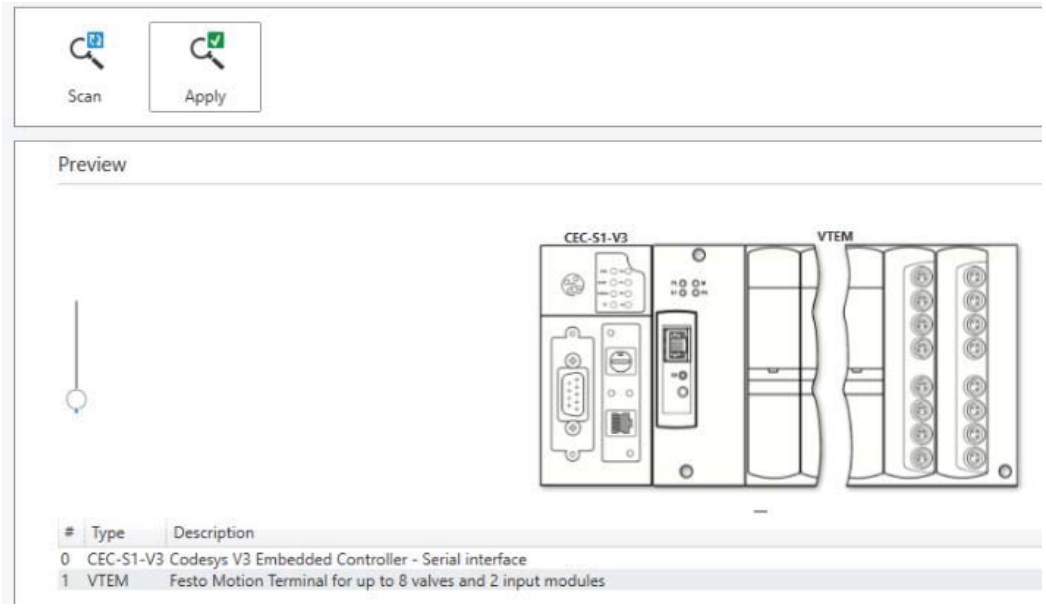


Fig. 23 Actual configuration of modules

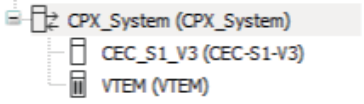
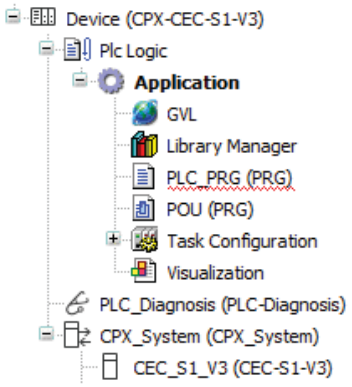


Fig. 24 Modules added to program tree

3.3 Parts of the application

To create a fairly basic application for FMT Application it should contain:

- Global Variable List (GVL)
- Library Manager
- Structured text (ST) POU (Program Organization Unit)



- Continuous Function Chart (CFC) POU
- Task Configuration
- Visualization

Some of them are automatically created when new project is started such as the Library Manager in which the VTEMDevCon library needs to be added as discussed in section Project Set-up.
More information about some of the sections is provided bellow.

Fig. 25 Finished projects Application tree

Global Variable List - GVL

To define variables that will be used throughout the Application, not only the ST program, Global Variable List has to be created. Variables here should be the valve and terminal variables and also the variables that will be used in Visualization as well as in POUs. To create GVL it is possible to right-click on Application and choose Global Variable List in the Add Object section. How to create GVL and which variables are used in this project is shown in Fig. 26.

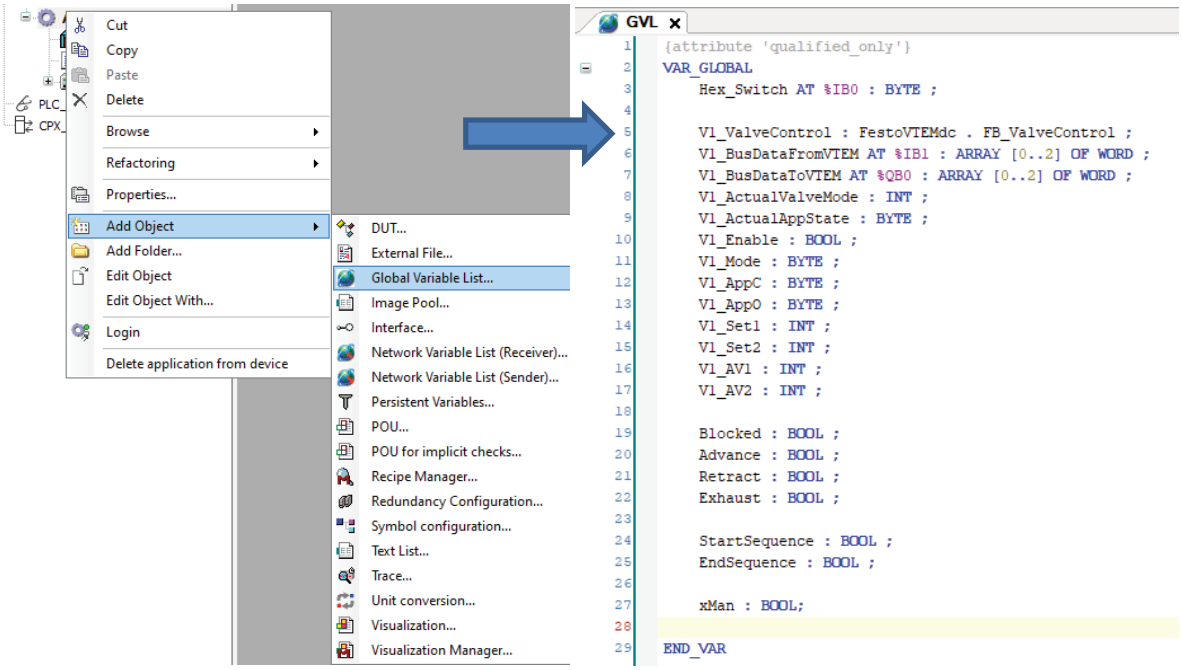


Fig. 26 On the left: creation of GVL, on the right: variables used in the project

Program organization unit in CFC (Continuous Function Chart) - POU

For proper communication between used variables in CPX block and FMT a function block in CFC needs to be used. To the formal parameters of the function block which are prepared in the function block the actual parameters from CPX variables are assigned. This block is prepared specifically for VTEM valve control. CFC POU can be added using same process as above with GVL but choosing POU. Type of POU should be kept as Program and Implementation Language can be chosen. In this case it is CFC which is default when creating new POU. After creating the CFC POU there will be ToolBox window on the right side of the screen. Box is to be chosen and dropped into the down section of the split screen on the left side. To insert the prefabricated setting right-click the box needs to be performed and Input Assistant chosen as shown in Fig. 27. There choice of FB_ValveControl needs to be done in Operations. Note that FestoVTEMdc is only visible when the special library is added previously in Library Manager. In the Input Assistant there is also a Documentation present (in Fig. 28), which explains the input and output variables and also defines their type.

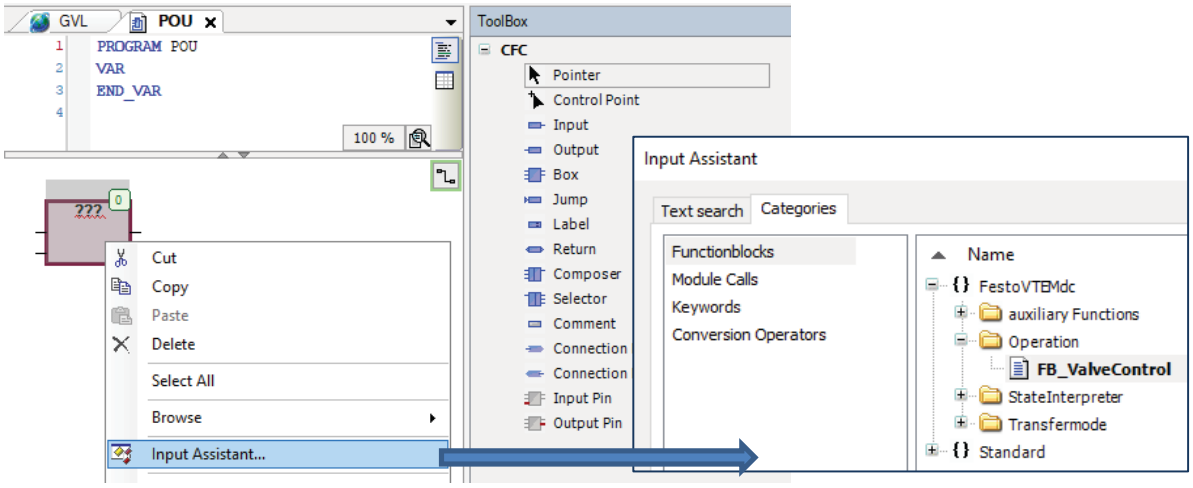


Fig. 27 Input of ValveControl function block

The prefabricated block will appear. To this block inputs and outputs should be connected as in Fig. 29. This function block encompasses only one of the VTEM valves, if more valves are to be used each should have their own function block and their own set of global variables.

Show documentation Insert with arguments Insert with namespace prefix

Documentation:

This is the central function block for the operation of Motion Apps on a valve. The FB serves to operate all Motion Apps with their specific setpoint values. It is able to execute and to stop Motion Apps and to deal with occuring errors. Parameterisation is not possible with this function block.

awBusDataFromVTEM	ARRAY [0..2] OF WORD	VAR_INPUT	6 bytes of process data that is received from the Motion Terminal
xEnable	BOOL	VAR_INPUT	has to be set to TRUE in order to operate a Motion App on the valve, otherwise any motion is stopped
bySetValveMode	BYTE	VAR_INPUT	ID of the Motion App to be executed, 60 for 'commissioning', '61 for 'stop', 62 for 'acknowledge'
bySetAppControl	BYTE	VAR_INPUT	SetAppControl of the chosen bySetMode, see documentation of the valve mode for its meaning
bySetAppOption	BYTE	VAR_INPUT	SetAppOption of the chosen bySetMode, see documentation of the valve mode for its meaning
iSetpointValue1	INT	VAR_INPUT	SetPoint1 of the chosen bySetMode, see documentation of the valve mode for its meaning
iSetpointValue2	INT	VAR_INPUT	SetPoint2 of the chosen bySetMode, see documentation of the valve mode for its meaning
xManualAcknowledge	BOOL	VAR_INPUT	errors can be acknowledged (after they got inactive) with a rising edge at this input
awBusDataToVTEM	ARRAY [0..2] OF WORD	VAR_OUTPUT	6 bytes of process data that is sent to the Motion Terminal
byActualValveMode	BYTE	VAR_OUTPUT	actual valve mode of the Motion Terminal
byActualValveState	BYTE	VAR_OUTPUT	actual valve state of the Motion Terminal
byActualAppState	BYTE	VAR_OUTPUT	Byte1 of the received process data: app state of the desired Motion App, see documentation of the valve mode for its meaning
iActualValue1	INT	VAR_OUTPUT	ActualValue1 of the desired Motion App, see documentation of the valve mode for its meaning
iActualValue2	INT	VAR_OUTPUT	ActualValue2 of the desired Motion App, see documentation of the valve mode for its meaning
iResponseToValveModeSet	eResponseToValveModeSet	VAR_OUTPUT	enumeration that indicates the status of execution of the desired valve mode
iErrorCode	INT	VAR_OUTPUT	contains the malfunction code of the latest error, 0 if no error occurred so far

Fig. 28 Documentation for block variables accessible in Input Assistant

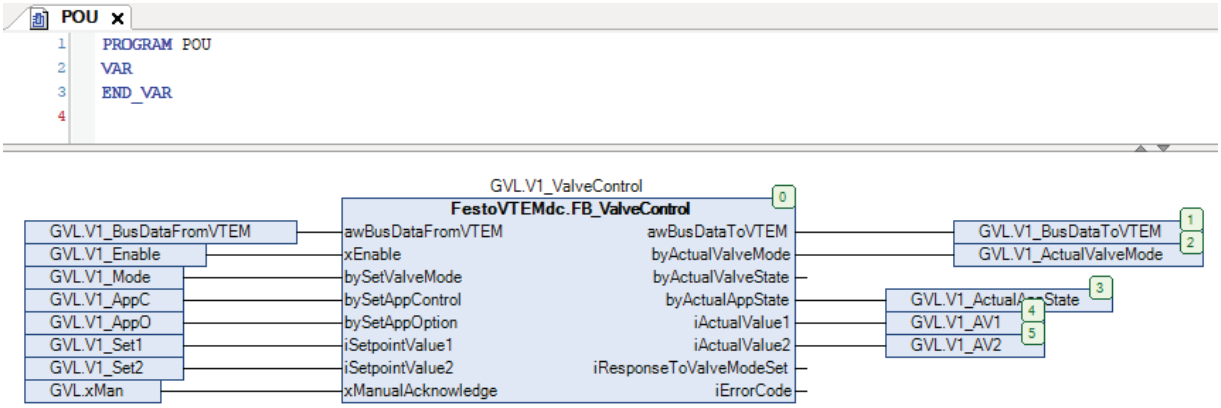


Fig. 29 Finished function block

Task Configuration

In Task Configuration Main Task is already present with the ST program in it. This tool allows for setting of different tasks and their type, when are they executed and so on. This can be left in default setting. The ST program will be executed cyclically every 20 ms. For more complex solutions Task Configuration tool can be used to create tasks that have different priority or are started "on event". Main Task configuration is shown in Fig. 30

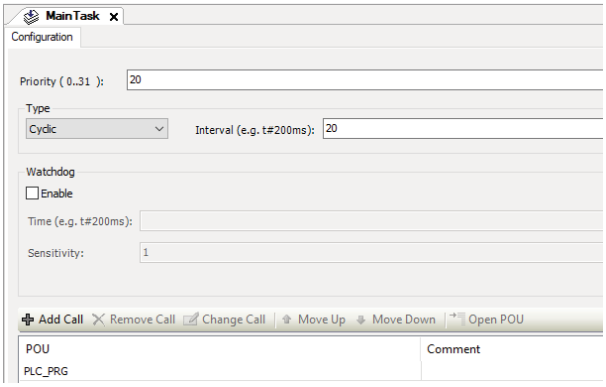


Fig. 30 Main Task Configuration in default mode

Visualization

There is no machine interface panel for the operator therefore the control of movement of the actuator is performed using the Visualization in CODESYS software environment.

Visualization is again created via right-click on Application and choice from the menu Add Object (Fig.31). After the visualization is created there is the visualization window on the left and ToolBox on the right. From the ToolBox in this project the main objects needed are buttons which are used to control BOOL variables in the ST program. These are located in the Common controls tab. Buttons can be labeled by clicking on them once when they are already selected. When button is selected the configurations of the button are displayed on the right. Most important is the Inputconfiguration, which allows the button to be linked with variables.

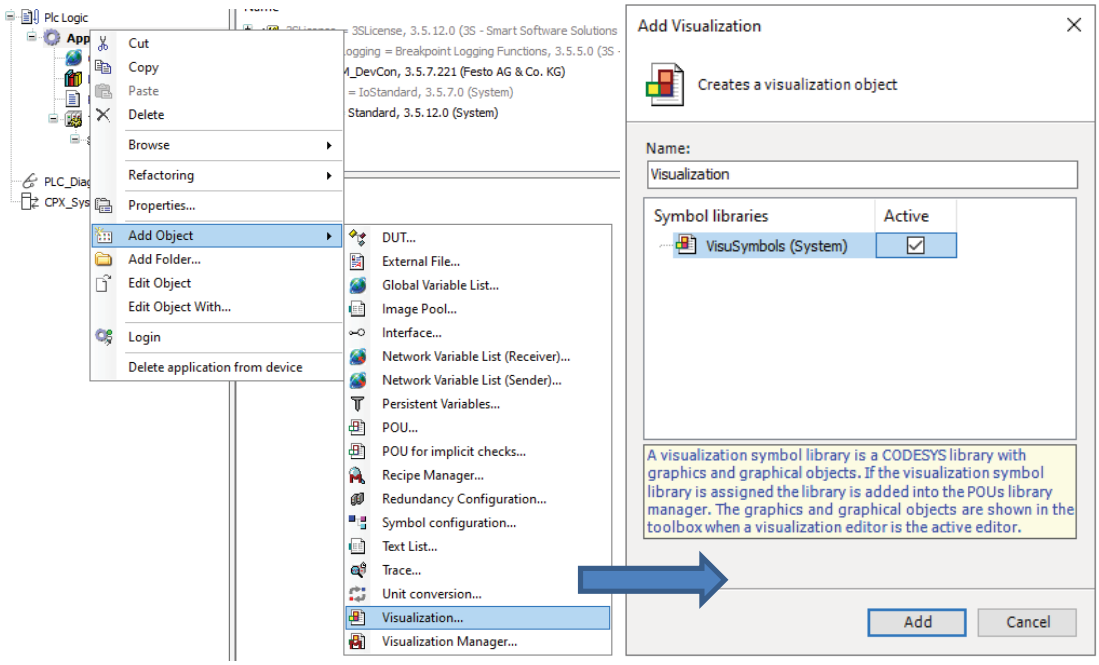


Fig. 31 Input of visualization in the Application

Fig. 32 shows button configuration. Variable that is flipped on mouse tap can be either written in or by clicking on symbol of 3 dots on the right side of the variable name tab you can open Input Assistant. Also it is possible to open Input Assistant by right-clicking the variable name tab, when writing is enabled. By inputting the "Advance" BOOL variable to the Inputconfiguration on Tap of the button labeled "ADVANCE" you link this variable with the use of this button. Same process is followed with other buttons as well. Of course other visualization elements are used such as labels, images and eventually sliders.

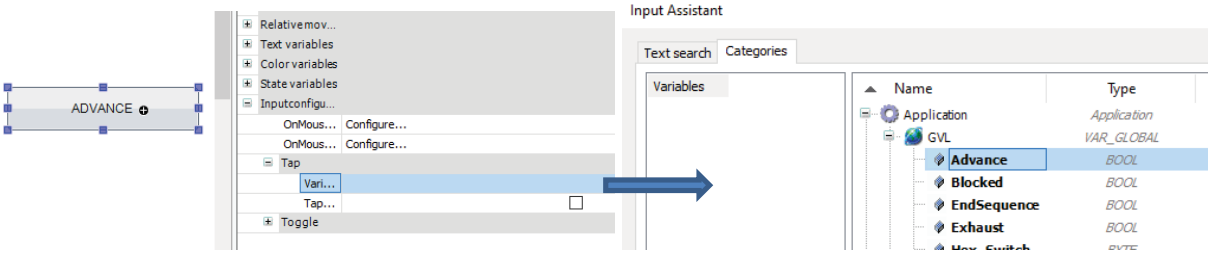


Fig. 32 Linking BOOL variable to the button in button Configuration

In the finished visualization all the buttons are linked to the BOOL variables of the corresponding names. The finished visualization used for this project is shown in Fig. 33. After sequence is started it must be ended before manual control can be resumed. Only button "BLOCKED" working as a safety measure can be used any time and after the piston is blocked the system returns to manual control. In this case sequence means that no matter what position the piston is in it will retract itself and then start cycle of full extension and full retraction.

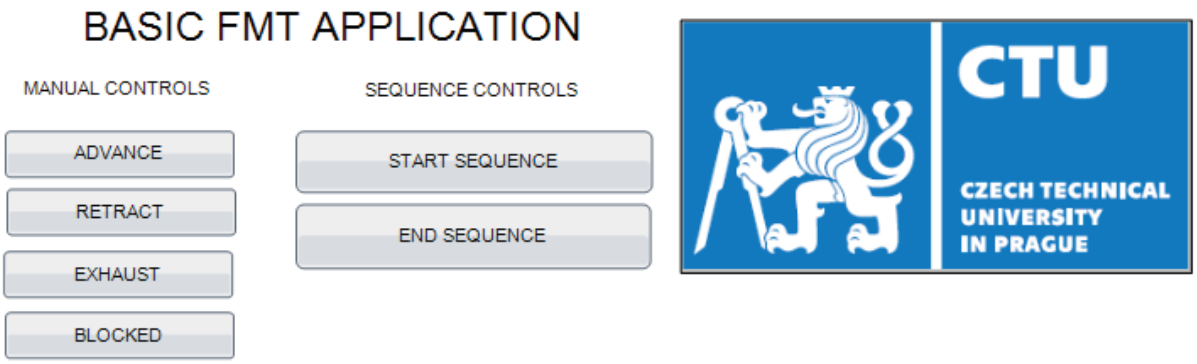


Fig. 33 Finished visualization of the project

Program in structured text (ST)

Finished program is shown here in Fig. 34 as an example of possible FMT application. This program enables manual and sequence control mode of one pneumatic actuator. Manual control allows for advancing, retraction, exhaust and blockage of the actuator. There is also a simple sequence, which uses the proximity switches. Sequence runs after "START SEQUENCE" is activated until either "END SEQUENCE" is activated or button "BLOCKED" is activated as a safety measure. After one of these events the manual control is resumed. The sequence consists of initial retraction of the actuator and then cycles of full extension and full retraction.

Below the logic of the code is described.

This section is already present in project tree. In the upper window local program variables are defined. These are variables that are confined only to this program. In this case it is only the State variable. In the lower window the program in language ST is written. The code used in the program is shown in Fig. 34.

In State 10 the dependence of the running of the program on the position of rotary switch is established. Also in this step the valve opening degree of supply air flow is set, valve is enabled for use and any lingering error statements are acknowledged by setting variable xMan to TRUE.

In state 20 mode of the valve is set to ECO drive and state of the app to Blocked as safe initial setting of the valve.

The program then cycles between states until one of the conditions is met. In this loop manual control is performed as well as the switch to sequence. If the sequence mode is activated via the visualization the program cycles in different states waiting again for one of the conditions that would warrant exiting this loop is met. In all the loops additional control of the stare of rotary switch (Hex Switch) is inserted. If the switch has been moved from "run" position the valve will not be enabled.

As was mentioned in visualization "blocked" state as safety element can be activated anytime and sends the program to the manual control loop, if all the conditions for the program to run are met.

This Application, the movements of the actuator, is controled by the buttons from the Visualisation window.

```
PROGRAM PLC_PRG
VAR
State : INT;
END_VAR
```

```
POU( ) ;

CASE State OF
0 :
    GVL . Vl_Enable := FALSE;
    State := 10;

10:
    IF (GVL . Hex_Switch = 2 ) THEN
    GVL.Vl_Set2:=10000;
        GVL . Vl_Enable := TRUE;
        GVL.xMan := TRUE;
        State := 20;
    ELSE
        State := 0;
    END_IF

20 :
    GVL . Vl_Mode := 6 ;
    GVL . Vl_AppC := 0 ;
    GVL.xMan := FALSE;
    State := 100;

100:
    IF GVL . Advance THEN
        GVL . Vl_AppC := 1 ;
        State := 120 ;
    ELSE
        State := 120;
    END_IF

120:
    IF GVL . Retract THEN
        GVL . Vl_AppC := 2 ;
        State := 140 ;
    ELSE
        State :=140;
    END_IF

140:
    IF GVL . Exhaust THEN
        GVL . Vl_AppC := 3 ;
        State := 160 ;
    ELSE
        State := 160;
    END IF

160:
    IF NOT ( GVL . Hex_Switch = 2 ) THEN
        GVL . Vl_Enable := FALSE ;
        State := 0;
    ELSE
        State := 180 ;
    END_IF
180:
    IF ( GVL.Vl_ActualValveMode = 61 ) THEN
        State := 10;
    ELSE
        State := 190;
    END_IF
190:
    IF ( GVL.StartSequence = TRUE) THEN
        State:= 195;
    ELSE
        State :=100;
    END_IF

195:
        GVL . Vl_AppC := 2 ;
        IF (GVL.Vl_ActualAppState = 1) THEN
        State:= 200;
        ELSE
        State:= 195;
        END IF
200:
    IF (GVL.Vl_ActualAppState = 4 ) THEN
        GVL . Vl_AppC := 2 ;
    END_IF
    IF (GVL.Vl_ActualAppState = 1) THEN
        GVL . Vl_AppC := 1 ;
    END_IF
    IF (GVL.EndSequence = TRUE) THEN
        State := 10;
    ELSE
        State := 210;
    END_IF
210:
    IF NOT ( GVL . Hex_Switch = 2 ) THEN
        State := 10;
    ELSE
        State := 200;
    END_IF
END_CASE
IF GVL . Blocked THEN
    GVL . Vl_AppC := 0 ;
    State := 0;
END_IF
```

Fig. 34 Application POU program in ST

4. Conclusion

In this paper, we have summarized the function of Festo Motion Terminal and its valves. We have also provided a thorough installation and connection guide. Further we demonstrated how to structure a program for FMT application. We have described the components of that program that are mostly given for any such application, thus leaving the reader with a blueprint for designing their own application and developed basis for more complex projects.

Acknowledgements

The authors acknowledge and thank for the initial training provided by Festo, s. r. o. in Prague.

References

[1] *Motion Terminal VTEM* [online]. Festo, 2018/03, , 33 [cit. 2020-02-18]. Available at: https://www.festo.com/cat/en-gb_gb/data/doc_ENGB/PDF/EN/VTEM_EN.PDF

[2] Digital pneumatics|Festo Motion Terminal. *Festo* [online]. [cit. 2020-03-30]. Available at: https://www.festo.com/vtem/en/cms/10169.htm

[3] Digital simplicity: World first Festo Motion Terminal VTEM. In: *Festo* [online]. 2017/11, p. 16 [cit. 2020-03-30]. Available at: https://www.festo.com/net/SupportPortal/Files/468005/PSI__Festo_Motion_Terminal_EN.pdf

[4] FESTO AG & CO. KG. *Motion Terminal VTEM: Instructions for use* [online]. In: . [cit. 2020-07-13]. Dostupné z: https://www.festo.com/net/SupportPortal/Files/469076/VTEM_2017-10_8071722g1.pdf

[5] FESTO AG & CO. KG. *Festo VTEM DevCon Version 3.5.7.233: Library for the control (Device Control) of the Motion Terminal VTEM from Festo and its elemts.* [online]. 2019 [cit. 2020-07-13].

Selected article from
Tento dokument byl publikován ve sborníku

Nové metody a postupy v oblasti přístrojové techniky, automatického řízení a informatiky 2020

New Methods and Practices in the Instrumentation, Automatic Control and Informatics 2020

14. 9. – 16. 9. 2020, Zámek Lobeč

ISBN 978-80-01-06776-5

Web page of the original document:
http://iat.fs.cvut.cz/nmp/2020.pdf

Obsah čísla/individual articles:
http://iat.fs.cvut.cz/nmp/2020/

Ústav přístrojové a řídicí techniky, FS ČVUT v Praze, Technická 4, Praha 6