

Model Based PLC Software Development and Tests

Model based PLC software development and test. Overview and classification of technologies illustrated for a MatLab/Simulink interface to Beckhoff TwinCAT.

- IEC 61131 (part 1-10)
- Beckhoff ADS (Automation Device Specification)
- MatLab PLC-Coder, TwinCAT TE1410

MatLab/Simulink

- Mathworks universe
(MatLab, Simulink, Stateflow, Toolboxes, Apps,..)
- Simulink is mainly used for signalflow based software
(Carnot THB)
- simplified view: define inputs, outputs, and an algorithm
inbetween

World of PLC Programming

- Hardware IO
- Realtime
- Safety
- IEC 61131 part 1-10 (since 1993, see next slide)
- Products (samples): - Siemens: Simatic Step7 - Beckhoff: TwinCAT 3/2 - CODESYS Group: CODESYS

Brief and Simplified Overview IEC 61131-3

deals with software architecture and
programming languages

- ladder diagram (relais, switches)
- function block diagram (OR, AND, ... blocks)
- ***structured text*** (similar to Pascal programming language)
- instruction list (close to assembler languages, deprecated)
- sequential function chart (similar to UML activity diagramm)
- data types (BOOL, BYTE, WORD, SINT, USINT, REAL, LREAL, DATE,)

unidirectional Bridge: Code-Generation

- MatLab/Simulink PLC Coder
- translating MatLab/Simulink to Structured Text FB's
- (who has experience with Simulink Coder (C) (embedded coder) ?
- some bugs, some tricks required to meet ST standards (like parameter passing in VAR_IN_OUT section) but overall good experience and readable code results

Data Interface: Connecting Signals

- MatLab Interface to TwinCAT
- Simulink/MatLab Simulations are enabled for calling PLC software
- PLC in the Loop Simulations
- currently used for PLC Software testing and verification, example Vitocal PLC tested in physical simulations environments over a one year cycle.
- estimate performance data (energy efficiency), parameter (PI controller) optimization, and so forth..

Technical Background (brief)

- Beckhoff TwinCAT vs. Codesys (both implementing IEC 61131-3)
- OPCUA as a common used interface (IoT, Industry 4.0, recommended)
- Beckhoff ADS (automation device specification) recommended when tied to Beckhoff
- TwinCAT ADS communication and TcXAEShell are limited to Windows OS (adstojava.dll, TcAdsDll.dll)
- ADS device concept: *Beckhoff Infosys*
- since 2021: TwinCAT/BSD, good experience running in a virtual machine on Windows (VirtualBox, WSL)

MatLab - TwinCAT

Connectivity: What is available?

- Beckhoff TE1410 (old version (buggy. slow, unusable))
- Beckhoff TE1410 (2.0 since 2021)
- easy to use and flexible MatLab-Tc Interface using TcAdsDll.dll, implemented in Java
- impression roundtrip follows

Beckhoff TE1410 class ADSPort

Help

TwinCAT.ADS.Port - MATLAB File Help

TwinCAT.ADS.Port - MATLAB File Help

TwinCAT.ADS.Port

TwinCAT.ADS.Port is a class.

Class Details

Superclasses	TwinCAT.ADS.Internal.Interop.ComObject , TwinCAT.ADS.SyncAsyncAdsPort
Sealed	false
Construct on load	false

Constructor Summary

[Port](#)

Property Summary

AutoStateUpdate	Determines if the state update is done through threads / Enable or disable state update
IsConnected	Get if the port has connected to the router Help for TwinCAT.ADS.Port/IsConnected is in
IsEnabled	Get if the port is enabled by the router Help for TwinCAT.ADS.Port/IsEnabled is inherite
IsServerEnabled	Determine if server functionality is enabled / Enable or disable server functionality Help
LocalAmsAddr	Get the local ams address Help for TwinCAT.ADS.Port/LocalAmsAddr is inherited from s
LocalPortName	Get the local port name Help for TwinCAT.ADS.Port/LocalPortName is inherited from su
RouterState	Get the router state Help for TwinCAT.ADS.Port/RouterState is inherited from superclass
Routes	Get an array of all known routes Help for TwinCAT.ADS.Port/Routes is inherited from su
RoutesD	Property Help for TwinCAT.ADS.Port/RoutesD is inherited from superclass TWINCAT.A
UnhandledIndications	Obtain array of unhandled indications received by this ADS port Help for TwinCAT.ADS.

Method Summary

AddAdsIndicationHandler	Execute this method with the specified arguments. Hel
AddDeviceNotification	Add device notification Help for TwinCAT.ADS.Port/Ad
AddDeviceNotificationEx	Add device notification extended Help for TwinCAT.AD
Connect	Connect to Router if not connected Help for TwinCAT.A
CreateAddDeviceNotificationRequest	Create add device notification request for synchronous
CreateAddDeviceNotificationRequestEx	Create add device notification request for synchronous
CreateDelDeviceNotificationRequest	Create delete device notification request for synchrono
CreateGroupRequest	Create group request for synchronous and asynchron
CreateInstance	
CreateNotificationBuffer	Create notification buffer Help for TwinCAT.ADS.Port/C
CreateNotificationBufferGroup	Create notification buffer group Help for TwinCAT.ADS.
CreateNotificationHandler	Create notification handler Help for TwinCAT.ADS.Port
CreateReadDeviceInfoRequest	Create read device information request for synchronou
CreateReadRequest	Create read request for synchronous and asynchronou
CreateReadStateRequest	Create read state request for synchronous and asynch
CreateReadWriteRequest	Create read write request for synchronous and asynch
CreateSumRequest	Create sum request for synchronous and asynchronou
CreateSumRequestL	Create sum request for synchronous and asynchronou
CreateValueContainer	Create type info object Help for TwinCAT.ADS.Port/Cre
CreateWriteControlRequest	Create write control request for synchronous and asyn
CreateWriteRequest	Create write request for synchronous and asynchronou
DelDeviceNotification	Delete device notification Help for TwinCAT.ADS.Port/D
Disconnect	Disconnect if connected Help for TwinCAT.ADS.Port/Di
GetAdsSymbolHandle	Get Ads symbol handle Help for TwinCAT.ADS.Port/Ge

TcXaeShell IDE, FB's and MAIN PRG

The screenshot displays the TcXaeShell IDE interface. The top menu bar includes options like Datei, Bearbeiten, Ansicht, Projekt, Erstellen, Debuggen, TwinCAT, TwinSAFE, PLC, Team, Scope, Extras, Fenster, and Hilfe. Below the menu is a toolbar with various icons for file operations and project management. The main workspace is divided into three panes:

- Projektmappen-Explorer:** Shows a hierarchical view of the project structure. The project is named "timestwo" (Projekt 1). It contains a "SYSTEM" folder with sub-items like Lizenz, Echtzeit, I/O Idle Task, Tasks, PicTask, Routing, Type System, and TcCOM Objekte. There is also a "MOTION" folder and an "SPS" folder. Under "SPS", there is a "timestwo" folder containing "timestwoProjekt" (with sub-items: Externe Datentypen, References, DUTs, GVLs, POU), "MAIN (PRG)", "timestwo (FB)", "VISUs", "PicTask (PicTask)", and "timestwo.tmc". At the bottom, there is a "timestwo Instance".
- Code Editor:** Displays the source code for the "MAIN (PRG)" file. The code is written in a structured text language (likely STL or similar) and includes comments. The code is as follows:

```
1 PROGRAM MAIN
2 VAR
3     count : ULINT;
4     strName : STRING := 'hello world';
5     fb : timestwo;
6     trigger : BOOL := FALSE;
7 END_VAR

1 count := count + 1;
2 IF trigger = TRUE THEN
3     fb();
4     trigger := FALSE;
5 END_IF
```
- Ausgabe:** Shows the output of the build process. The output text is:

```
Ausgabe anzeigen von: Build
Generate TMC information ...
Import symbol information ...
SPS.timestwo : message: generate boot information...
----- Alles neu erstellen: 1 erfolgreich, 0 fehlerhaft, 0
```

The bottom status bar indicates the current project is "timestwo" and the target is "PC-102B67".

TcXaeShell IDE, FB's and MAIN PRG

timstwo MAIN Target Browser ✱ X timstwo

ADS Enter Filter...

N4271W05
PC-102B67
350: PlcTask
851: Port851
350: PlcTask
851: Port851

Name	Type	Size	Category	Full-Name
Constants		0	Struct	Constants
Global_Version		0	Struct	Global_Version
MAIN		0	Struct	MAIN
count	ULINT	8	Primitive	MAIN.count
fb	timstwo	32	Struct	MAIN.fb
in	LREAL	8	Primitive	MAIN.fb.in
out	LREAL	8	Primitive	MAIN.fb.out
factor	LREAL	8	Primitive	MAIN.fb.factor
strName	STRING(80)	81	String	MAIN.strName
trigger	BOOL	1	Primitive	MAIN.trigger
TwinCAT_SystemInfoVar		0	Struct	TwinCAT_SystemInfoVarList

Simple Interface Test Script

```
%% interfacing TwinCAT project: timestwo.prj

% get an instance of the ADS Port Object
Port = TwinCAT.ADS.Port();

% get target port object
% ads = Port.GetPortConnection(strcat('Local',':851'))
% rhrw, TwinCAT/BSD
ads = Port.GetPortConnection(strcat('39.206.37.142.1.1',':851'))

% set parameter
ads.Write('MAIN.fb.factor', 2.22);

% read string
s = ads.Read('MAIN.strName', []);
fprintf('Name: %s\n', s);

% attention ! datatype
c = ads.Read('MAIN.count', [])

% set input
ads.Write('MAIN.fb.in', 123.456);

% trigger PLC cycle
ads.Write('MAIN.trigger', true);

% get result
r = ads.Read('MAIN.fb.out', [])
```

Simulink Block's are available too

The image displays three screenshots related to the Beckhoff TwinCAT ADS interface in Simulink.

Top Left: Simulink Library Browser
The 'Beckhoff TwinCAT ADS Interface/Synchronous' library is selected. The left pane shows the library structure, and the right pane displays the 'TC ADS Symbol Interface' block with its input and output ports.

Top Right: Simulation Controls
A partial view of the Simulink simulation controls, showing 'Run', 'Step Forward', 'Stop', and 'Data Inspector' buttons.

Bottom: TwinCAT Simulink Interface
This window provides a detailed view of the ADS symbols and block settings.

ADS symbols table:

BlockPort	Name	Type	NetId
[Empty table body]			

General Simulink block settings:

- Stop simulation: [Dropdown menu]
- Sample time: 0.000000 s
- Synchronisation:
 - ☒ No synchronisation
 - ☐ Synchronised data exchange
 - Buffer length: 10.0 s
 - Delay time: -1 ms
 - ☐ Time stamp output port

Block diagram and data table:

The block diagram shows the 'TC ADS Symbol Interface' block with its input and output ports. Below it, a table lists the data types and sizes for the interface:

Name	Type	Indx	Indx	Size	Full-Name
fb	timesTwo	0x40	0x5E	32	MAIN fb
count	ULINT	0x40	0x5E	8	MAIN count
stName	STRING	0x40	0x5E	81	MAIN stName
trigger	BOOL	0x40	0x5E	1	MAIN trigger

Block diagram and data table:

The block diagram shows the 'TC ADS Symbol Interface' block with its input and output ports. Below it, a table lists the data types and sizes for the interface:

Name	Type	Indx	Indx	Size	Full-Name
fb	timesTwo	0x40	0x5E	32	MAIN fb
count	ULINT	0x40	0x5E	8	MAIN count
stName	STRING	0x40	0x5E	81	MAIN stName
trigger	BOOL	0x40	0x5E	1	MAIN trigger

MatLab-Tc Interface Simulink Doc

Help

viessmann.MatLabIO

Documentation

Search Documentation

CONTENTS

Documentation Home

Java Documentation (Supplemental Software)

TcADS_Interface (Supplemental Software)

All classes

TcADS_Interface

TcADS_Interface.RangeMode

Test_TcADS_Interface

PACKAGE

CLASSES

TOO

DEPRECATED

INDEX

HELP

PREV CLASS

NEXT CLASS

FRAMES

NO FRAMES

SUMMARY

NESTED

FIELD

CONSTR

METHOD

DETAIL

FIELD

CONSTR

METHOD

viessmann.MatLabIO

Class TcADS_Interface

java.lang.Object

viessmann.MatLabIO.TcADS_Interface

public class **TcADS_Interface**

extends java.lang.Object

MatLab/Simulink to Beckhoff TwinCAT Interface

Overview

On Windows PC's Beckhoff provides DLL's and a Java library for TwinCAT communication by ADS. With TwinCAT default installation settings these can be found here:

- c:/TwinCAT/AdsApi/TcAdcDll/TcAdcDll.dll
- c:/TwinCAT/AdsApi/AdcToJava/shq/adsobjecs.dll
- c:/TwinCAT/AdsApi/AdcToJava/ToJavaToAds.jar

Because MatLab is implemented very tight with Java, it is obvious to provide a class "TcADS_Interface" which provides a wrapper with convenience functions to connect MatLab/Simulink to Beckhoff ADS. For Simulink connectivity a library Library/libtrads.dll calling MatLab \$-functions is provided.

Alternatives

Since October 2021 Beckhoff distributes a new version 5.0.1.0 of TC420. This is basically a MatLab Class providing similar methods (but less new and convenient than the TcADS_Interface) for ADS communication. It is very likely that, under the hood the same Windows DLL TcAdcDll.dll is used. For Simulink there are library blocks for synchronous and asynchronous communication, as well as a Tc Symbol Interface for click GUI configuration. For completeness the .NET ADS communication library is mentioned. Details and how to use it from MatLab could be found in the <https://info.boschbeckhoff.com/portal>.

Bugs

MatLab will not find adsobjecs.dll in TwinCAT location c:/TwinCAT/Common64/, even with proper PATH settings. As workaround copy adsobjecs.dll to MatLab installation path, e.g. c:/Apps/MATLAB/R2020b/bin/win64/

Concepts

TwinCAT Variables (datapoints) are described by "ADS Symbol Entries". These hold meta data i.e. address and type informations. In a first step you have to determine the Symbol Entry, e.g. by the path and name of the variable. Then read/write operations can be executed, variables are addressed by symbol entries. There are two flavours of read/write functions. Single ADS service calls are executed by passing a single scalar ADS Symbol entry. If a vector of Symbol Entries is passed to the function, all read/write operations are executed within one ADS service call. This is implemented by TwinCAT sum-commands. Latter could be used to improve performance, one service call will take approx. 15 ms on a common average Windows PC.

MatLab usage example

matlab>

% set path, MatLab can instantiate classes defined in jar files
javaddpath('c:/TwinCAT/AdcApi/AdcToJava/TcJavaToAds.jar');
javaddpath([pwd, '/jars/MatLab-Simulink_TcADS_Interface-v0.2.jar']);

% instantiate and connect an interface object
ads = viessmann.MatLabIO.TcADS_Interface;

TwinCAT TcXAEShell

The screenshot displays the TwinCAT TcXAEShell environment. The top menu bar includes options like Datei, Bearbeiten, Ansicht, Projekt, Erstellen, Debuggen, TwinCAT, TwinSAFE, PLC, Team, Scope, Extras, Fenster, and Hilfe. Below the menu is a toolbar with various icons for file operations and development tools.

The left sidebar shows the 'Projektmappen-Explorer' (Project Explorer) with a tree structure of the project 'Heizkreis_v1 - TcXaeShell'. The tree includes folders for 'SYSTEM', 'MOTION', 'SPS', and 'fbHc_testbench'. Under 'fbHc_testbench', there is a 'MAIN (PRG)' folder containing 'MAIN (PRG)', 'VISUs', 'fbHc_testbench.tmc', 'PicTask (PicTask1)', and 'fbHc_testbench Instance'.

The main workspace is divided into two panes. The top pane, titled 'MAIN [Online] * typHc_Para * Target Browser Bibliotheksverwalter', displays a table of variables:

Ausdruck	Datentyp	Wert	Vorbereiteter ...	Adresse	Kommentar
* fbhc	VI_DefaultLibrary.FbHc				
* trigger	BOOL	FALSE			
* lochmi	typHc_LochMI				Loc HMI structure
* para	typHc_Para				Parameter
* kpis	typHc_KPis				KPis
* typScpKPis	typScpKPis				
* uiActualStep	INT	0			Actual step screed program

The bottom pane shows a ladder logic program (LAD) with the following code:

```
1
2 IF trigger FALSE = TRUE THEN
3
4   fbhc(typLochMI:=lochmi, typPara:=para, typKPis:=kpis );
5   trigger FALSE := FALSE;
6
7 END_IF
8 RETURN
```

At the bottom, the 'Fehlerliste' (Error List) pane shows a single error:

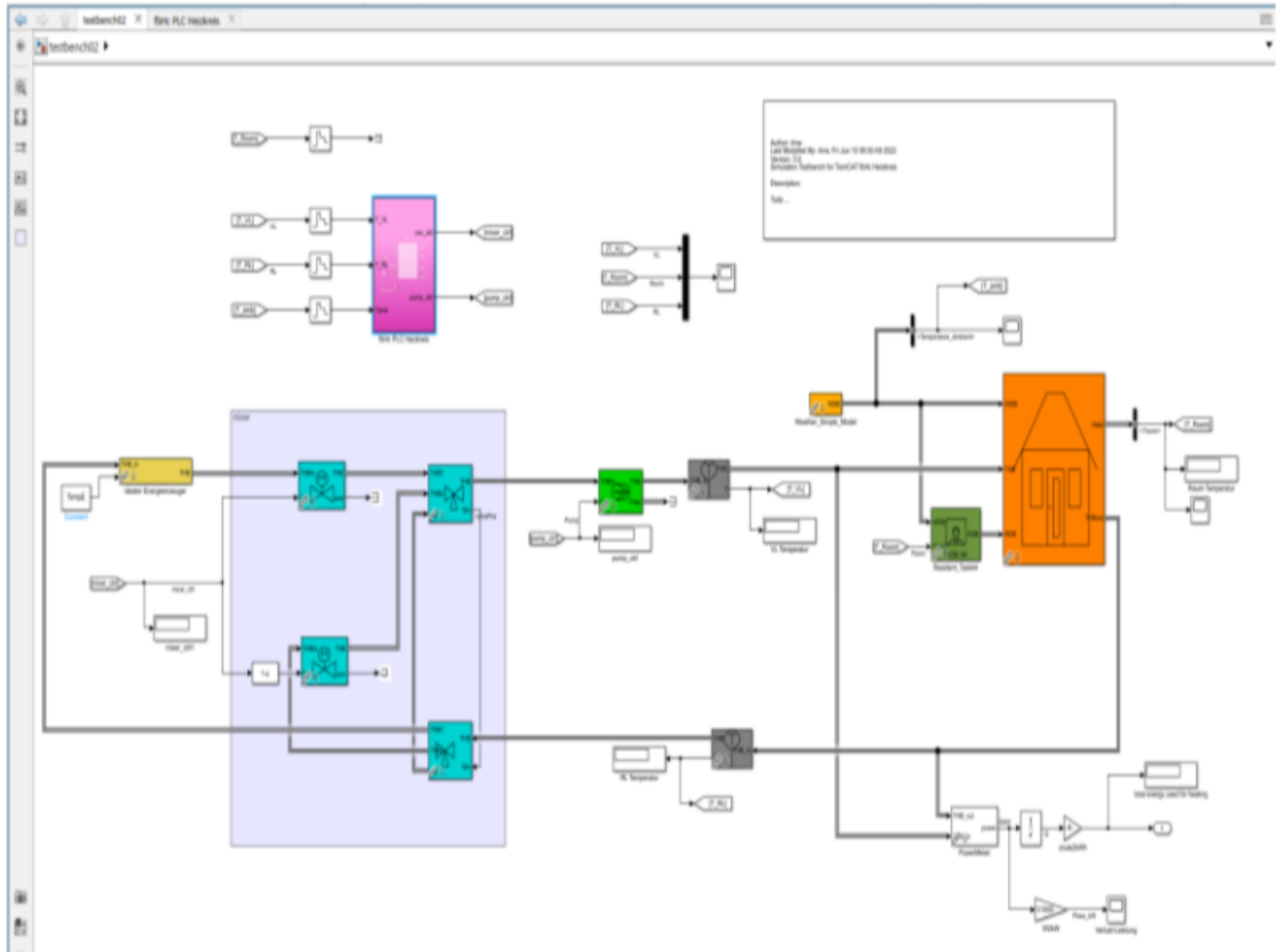
1 Fehler 0 Warnungen 0 von 8 Mitteilungen Löschen Erstellen + IntelliSense

Beschreibung

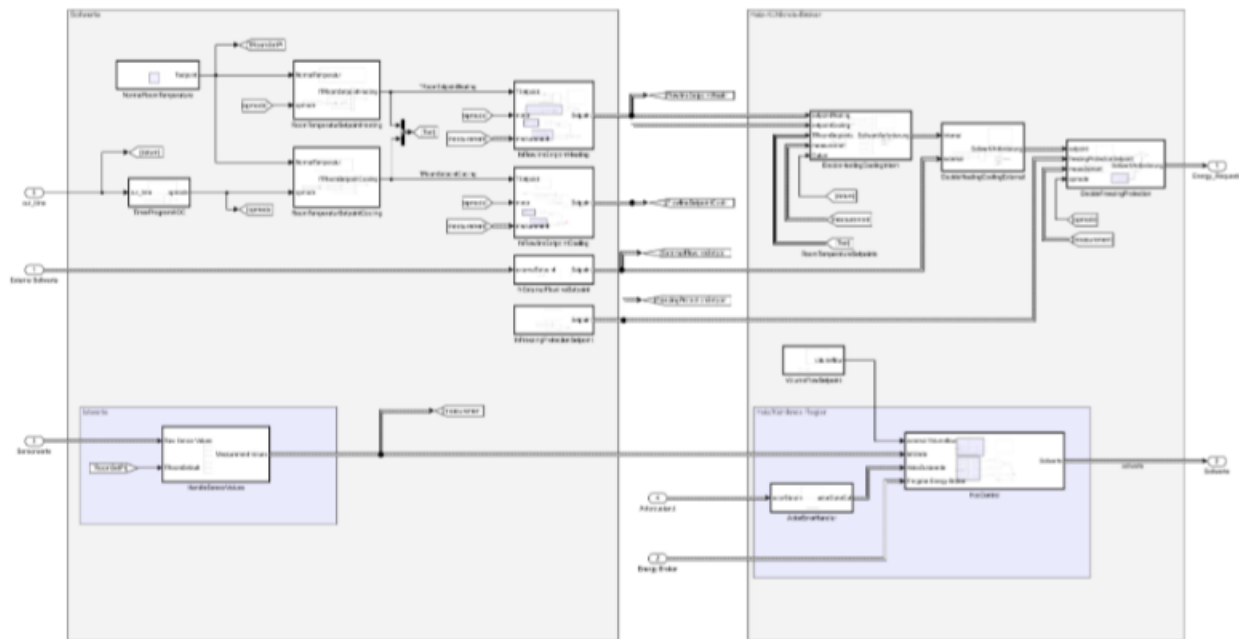
Die Bibliothek 'TCIMXS3, 1.0.0.0 (Beckhoff)' konnte nicht geöffnet werden. (Grund: Die Bibliothek 'TCIMXS3, 1.0.0.0 (Beckhoff)' ist nicht installiert.)

The status bar at the bottom indicates 'Projektmappen-Exp...' and 'Team Explorer: Verb...'.

Simple House controlled by PLC heating circuit program



Code Generation Example (HCC Development)



ST Code Generation Example

Projektmappen-Explorer

Projektmappen-Explorer durchsuchen (Strg+i)

- Lizenz
- Echtzeit
- Tasks
- Routing
- Type System
- TcCOM Objekte
- MOTION
- SPS
 - HeizKuehlKreis
 - HeizKuehlKreisProjekt
 - Externe Datentypen
 - References
 - DUTs
 - GVLs
 - POUs
 - ActorErrorHandler (FB)
 - c_DecideHeatingCoolingExter (FB)
 - FehlerEnum (ENUM)
 - GVL_1538516398
 - HandleSensor/Values (FB)
 - Heiz_Kuehl_Kreis (FB)
 - MAIN (PRG)
 - MATLABFunction (FUN)
 - rt_celid (FUN)
 - rt_flood (FUN)
 - rt_fmdd (FUN)

TRoom

TimerProgramHCC * X MAIN

TRoomSensors_t Solwert_t Target Browser RawSensors_t ExtSolwert_t

```
1 FUNCTION_BLOCK TimerProgramHCC
2 VAR_INPUT
3   cur_time: LREAL;
4   rtp_idx: LREAL;
5 END_VAR
6 VAR_OUTPUT
7   opmode: SINT;
8 END_VAR
9 VAR
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91 (* t is the number of week-minutes 0..7*1440, *)
92 (* starting every Sunday morning 0:00 *)
93
94 (* local helper function *)
95
96 (* 'TimerProgramHCC:69' n=1; *)
97 b_n := 0;
98 (* 'TimerProgramHCC:70' for i = 1:7 *)
99
100 FOR b_i := 0 TO 6 DO
101   (* 'TimerProgramHCC:71' if (t > (i-1) * 24*60) && (t <= i*24*60) *)
102
103   IF (u_0 > (((DINT_TO_LREAL(b_i) + 1.0) - 1.0) * 24.0) * 60.0) AND (TRUNC(u_0) <= (((b_i + 1) * 24) * 60)) THEN
104     (* 'TimerProgramHCC:72' n = i; *)
105     b_n := b_i;
106   END_IF;
107
108 END_FOR;
109
110 (* 'TimerProgramHCC:35' td = t - (n-1) * 24*60; *)
111 td := u_0 - (((DINT_TO_LREAL(b_n) + 1.0) - 1.0) * 24.0) * 60.0;
112 (* minutes of day *)
```

Fehlerliste

Summary

- MatLab/Simulink models can be interfaced with PLC's
- OPCUA, or a vendor specific protocol like Beckhoff ADS
- MatLab PLC-Coder can be a huge benefit fir time and code quality (if properly used)

That's all for now

thank you for your patience!

- <https://www.plcopen.org/iec-61131-1|2|3|4|5|7|8>
- <https://infosys.beckhoff.com>
- <https://en.wikipedia.org>
- <https://www.viessmann.com>
- <https://www.mathworks.com>