StuderLib
StuderLibEx
StuderLibGlobal

TXV 00395.02
first edition
September 2016
All rights reserved

Modifications history

Table 1.

Date	Publication	Modification description
September 2016	1	First version

Content

1 INTRODUCTION

Library : STUDER Lib

StuderLib, StuderLibEx, StuderLiGlobal library is supplied as part of Mosaic programming. The library contains function blocks enable communication with STUDER equipment like (Xtender,Vario string,Vario track and etc) via communication module Xcom-232i supplied by Studer company.

1.1 Supported services, properties and objects

Library : STUDER Lib

To connect the STUDER equipment like (Xtender,Vario string,Vario track and etc) thorugh RS232 interface to any cental control systém, STUDER company using proprietary

communication module Xcom-232i.
The communication channel CH2 (CH3, CH4) of any FOXTROT basic module connects to STUDER network though this communication module only.
The communication module Xcom-232i, equipped with a serial port RS-232, enables to be informed of the state of a system consisting of one or several Xtenders, VarioTracks or VarioStrings.
Xcom-232i allows to read all basic data and also to modify the configuration parameters.

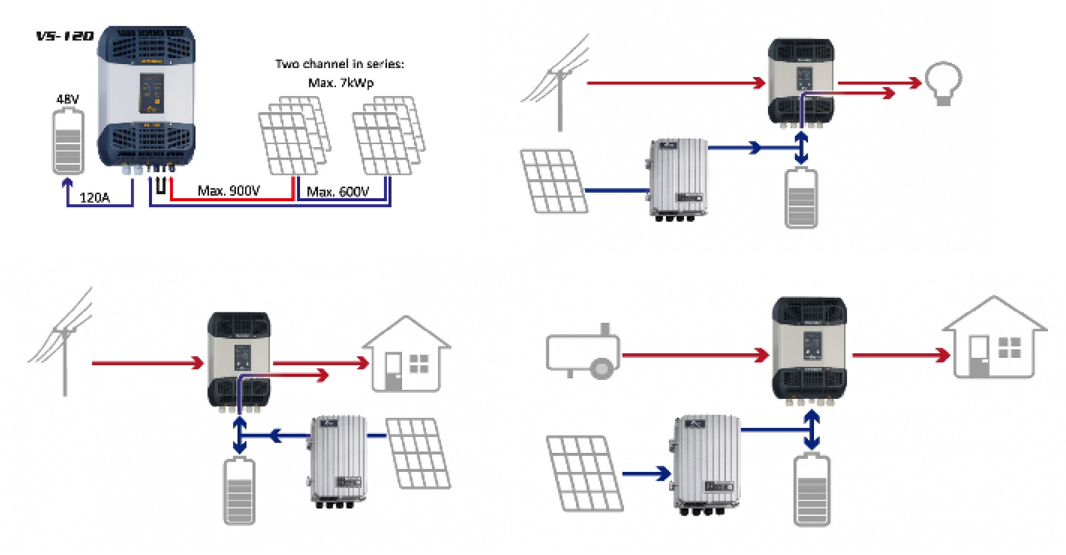


Figure 1. Examples of usage

Services:

- Read property
- Read parameters
- Write parameters

Object types:

- User info

  The available user information is the same as the values that can be chosen to be displayed on the RCC. The user information is related with the inverter parameters that can be configured with the RCC. The functionalities of each parameter are described in the RCC manual. You can easily find specific parameters by using the parameter index at the end of the manual.

- Parameter

  All parameters accessible from the remote control can also be modified with the protocol. The behaviour is the same as if a physical person changes the value with the remote control buttons. Currently, only changes at the level qsp are possible.

Property ID for User info

- Value

Property ID for Parameter

- Value qsp
- Min qsp
- Max qsp
- Level qsp

IMPORTANT*:The change of parameters when the inverters are in operation should be done carefully. The modification of parameters can restart the corresponding algorithm inside the inverter. For example, the change of a delay can restart the timer attached to it.*

1.2 Supported models and addressing

Library : STUDER Lib

Applied series and models:

- Xtender : address from 101 to 109
- BSP : address 601
- VarioTrack : address from 301 to 315
- VarioString : address from 701 to 715

IMPORTANT: STUDER Lib library supports protocol version: Date: 05.12.14, Version: V1.5.28 (R552)

Xtender Vario track Vario string RCC2 Xcom-232i

Figure 2. List of supported devices

1.3 Connecting the communication bus

Library : STUDER Lib

The devices of the Xtender range are equipped with an own communication bus for data exchange, configuration and updating of the system. Connection is being made by linking the devices with the communication cables. One gets then a bus online where a link end must be activated on the devices on each end, to obtain the configuration of the Figure 3.

Each device is equipped with a switch offering to choose between open "O" and terminated "T". The devices at the end of the line must be set on "T". The others, receiving two communication cables, must be set on "O".

NOTE:By default, the termination is not activated on each product of Studer Innotec SA .

IMPORTANT: A wrong setting of the link ends can lead to an erratic running of the installation or impede its updating.

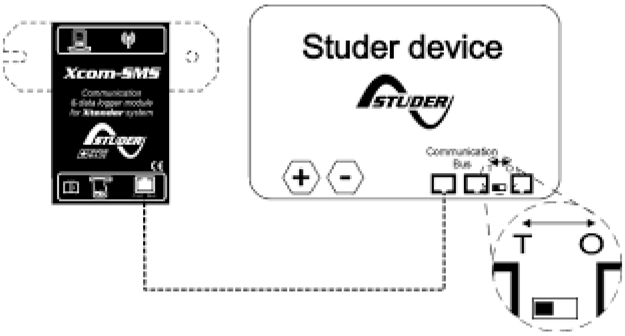
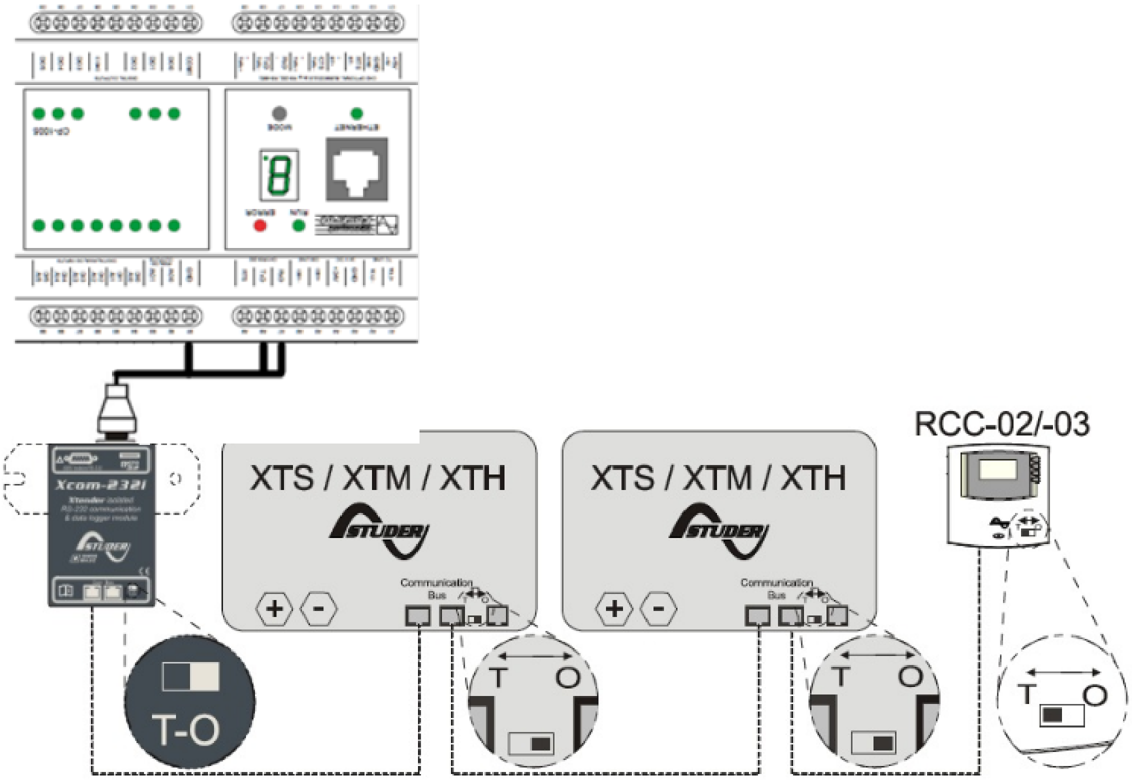


Figure 3. Connection between communication module Xcom-232

1.4 System connection

Library : STUDER Lib



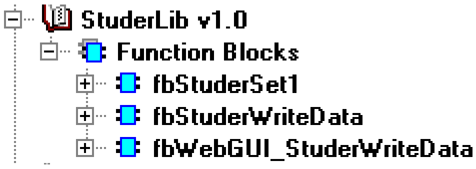
–

Figure 4. Hardware interconnection between Foxtrot and Studer network

1.5 Library overviewcautions and limits

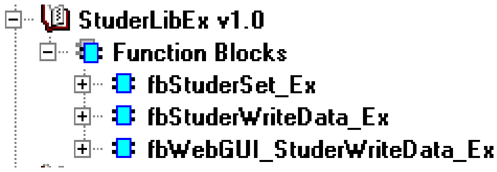
Library : STUDER Lib

Studer Library concist of two independent libraries like StuderLib and StuderLibEx and one depended StuderLibGlobal. StuderLibGlobal contains common functions and data structures for both libraries tuderLib and StuderLibEx. StuderLibGlobal library adds to project automatically when one of two libraries is added to project by Mosaic IDE user.
IMPORTANT: Please don't use StuderLibGlobal functions of functional block in your applications



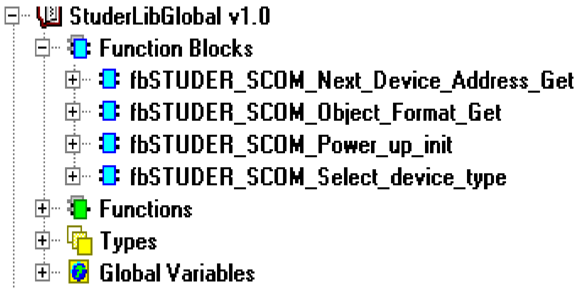
StuderLib v1.0
- Function Blocks
  - fbStuderSet1
  - fbStuderWriteData
  - fbWebGUI_StuderWriteData

Figure 5. Overview of StuderLib library in Mosaic IDE



StuderLibEx v1.0
- Function Blocks
  - fbStuderSet_Ex
  - fbStuderWriteData_Ex
  - fbWebGUI_StuderWriteData_Ex

Figure 6. Overview of StuderLibEx library in Mosaic IDE



StuderLibGlobal v1.0
- Function Blocks
  - fbSTUDER_SCOM_Next_Device_Address_Get
  - fbSTUDER_SCOM_Object_Format_Get
  - fbSTUDER_SCOM_Power_up_init
  - fbSTUDER_SCOM_Select_device_type
- Functions
- Types
- Global Variables

Figure 7. Overview of StuderLibGlobal library in Mosaic IDE

CAUTION!
Please don't use both libraries StuderLib and StuderLibEx at the same project, it can be cause of data structure collitions and compilation errors.

CAUTION!
If requeired to use StuderLib library functions in the PLC application program, you have to add this library to your project. Along with the library in the project StureLib automatically added following libraries ComLib, StdLib, SysLib because StuderLib uses some functions from these libraries. The library is supplied as part of the installation of the Mosaic version v2016.1.

CAUTION!
Library StuderLib requires preprogrammed PLC application profile
TXF 689 17 AP STUDERLIB_LICENCE . This profile is not a standard PLC and must be ordered separately to each PLC. Without this profile, the library operates only for a limited period of time (4 hours for testing and debugging purposes). Then block stops communication and returns back profile error indication. To renew library operation required to switch off PLC and restart (again, for a limited time) or after programming application profile TXF 689 17 (no time limit).

StuderLib library is not supported on systems with TC-650 for the system TC700 library is not compatible with the processor modules CP-7002, CP-7003 and CP-7005.
StuderLib library functions are supported in the central units of series K and L (TC700 CP-7000, CP-7004 and CP-7007, all variants of Foxtrot) from version v4.6.

2 FUNCTION BLOCKS

Library : STUDER Lib

STUDER Lib library is supplied as part of Mosaic programming environment. The library contains functions and function block provides the ability to control STUDER networks that are connected to the PLC serial channel through RS232 interface.

IMPORTANT! The major difference between two libraries is write data process to parameters of Studer devices.

- StuderLib operates with group parameters of same type devices.
- StuderLibEx operates with single parameter of selected device

For instance: In your project Studer network consist of 3 eXtenders with adresses 301,301,302.

a. To change value of parameter „Battery charge current“, through fbStuderWriteData from user program, required to define address of parameter '1138' and set value to real variable. Main functional block fbStuderSet1 will stop read process and will send new value to all devices (301,302,303) automatically. After write process will be finished, read endless cycle will be restarted.
NOTE: Write cycle takes at least 2 seconds per parameter. During write process new write command will be denied.

- **Advantages**-Do not need to care how many devices connected to network.
- **Disadvantages** – can't be control each device separately.

b. To change value of parameter „Battery charge current“, through fbStuderWriteData_Ex from user program, required to define address of parameter, adress of device connected to network, set value.
IMPORTANT: If selected parameter address is not supported by device, write command will be denied and deleted.

- **Advantages**-each device controls separately. Special feature for advanced Studer network driving.
- **Disadvantages** – More deeply knowledge of Studer protocol structure and broadcasting to each device separately.

Table 2. Functions of StuderLib library

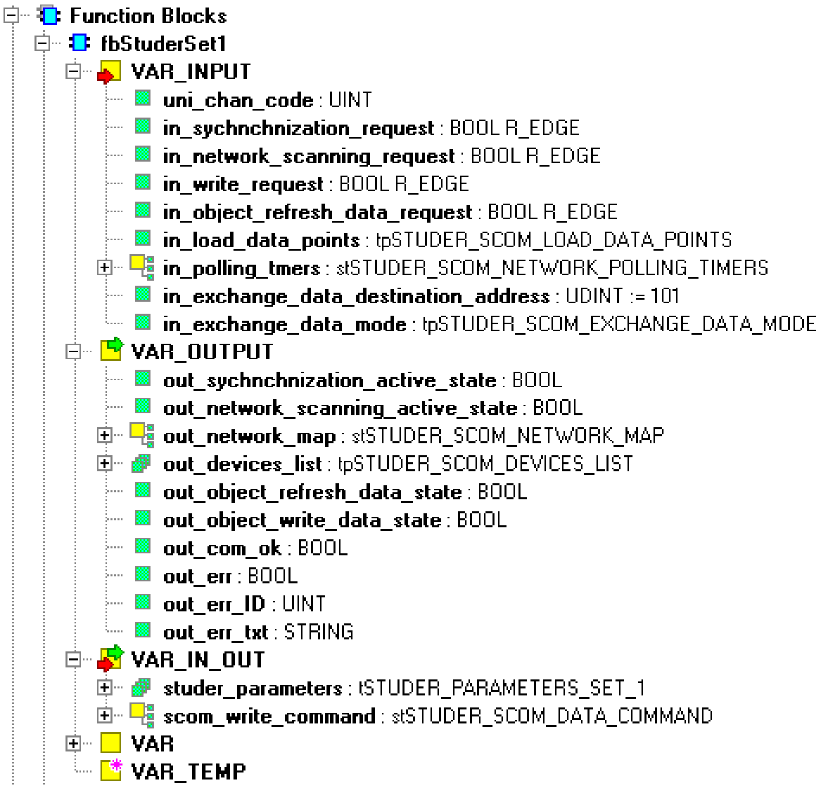
Function block	*Description*
fbStuderSet1	Communication with STUDER network
fbStuderWriteData	Write property values of object to all defined parameters of same type
fbWebGUI_StuderWriteData	Web UI for manual write property values.

Table 3. Functions of StuderLibEx library

Function block	*Description*
fbStuderSet_Ex	Communication with STUDER network
fbStuderWriteData_Ex	Write property values of object (type parameters)
fbWebGUI_StuderWriteData_Ex	Web UI for manual write property values.

2.1 Function block fbStuderSet1 (StuderLib)

Library : STUDER Lib



```
Function Blocks
  fbStuderSet1
    VAR_INPUT
      uni_chan_code : UINT
      in_sychnchnization_request : BOOL R_EDGE
      in_network_scanning_request : BOOL R_EDGE
      in_write_request : BOOL R_EDGE
      in_object_refresh_data_request : BOOL R_EDGE
      in_load_data_points : tpSTUDER_SCOM_LOAD_DATA_POINTS
      in_polling_tmers : stSTUDER_SCOM_NETWORK_POLLING_TIMERS
      in_exchange_data_destination_address : UDINT := 101
      in_exchange_data_mode : tpSTUDER_SCOM_EXCHANGE_DATA_MODE
    VAR_OUTPUT
      out_sychnchnization_active_state : BOOL
      out_network_scanning_active_state : BOOL
      out_network_map : stSTUDER_SCOM_NETWORK_MAP
      out_devices_list : tpSTUDER_SCOM_DEVICES_LIST
      out_object_refresh_data_state : BOOL
      out_object_write_data_state : BOOL
      out_com_ok : BOOL
      out_err : BOOL
      out_err_ID : UINT
      out_err_txt : STRING
    VAR_IN_OUT
      studer_parameters : tSTUDER_PARAMETERS_SET_1
      scom_write_command : stSTUDER_SCOM_DATA_COMMAND
    VAR
    VAR_TEMP
```

Figure 8. Structure of FB fbStuderSet1 (StuderLib)

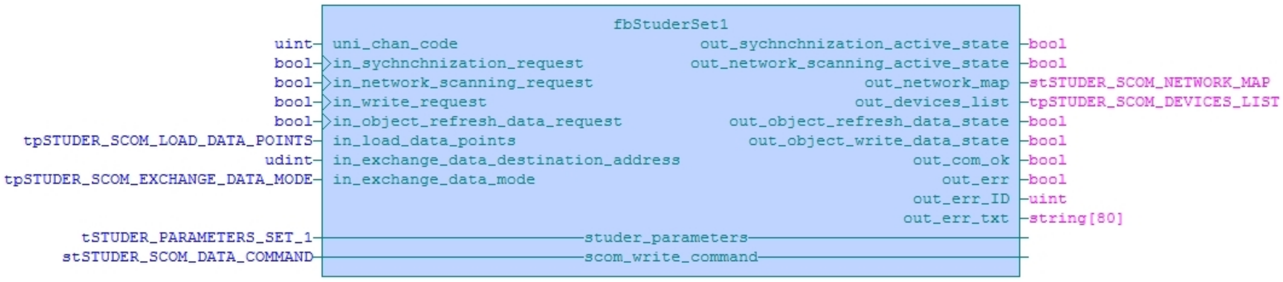

fbStuderSet1
uint
uni_chan_code
bool
in_sychnchnization_request
bool
in_network_scanning_request
bool
in_write_request
bool
in_object_refresh_data_request
tpSTUDER_SCOM_LOAD_DATA_POINTS
in_load_data_points
udint
in_exchange_data_destination_address
tpSTUDER_SCOM_EXCHANGE_DATA_MODE
in_exchange_data_mode
tSTUDER_PARAMETERS_SET_1
stSTUDER_SCOM_DATA_COMMAND
studer_parameters
scom_write_command
out_sychnchnization_active_state
bool
out_network_scanning_active_state
bool
out_network_map
stSTUDER_SCOM_NETWORK_MAP
out_devices_list
tpSTUDER_SCOM_DEVICES_LIST
out_object_refresh_data_state
bool
out_object_write_data_state
bool
out_com_ok
bool
out_err
bool
out_err_ID
uint
out_err_txt
string[80]

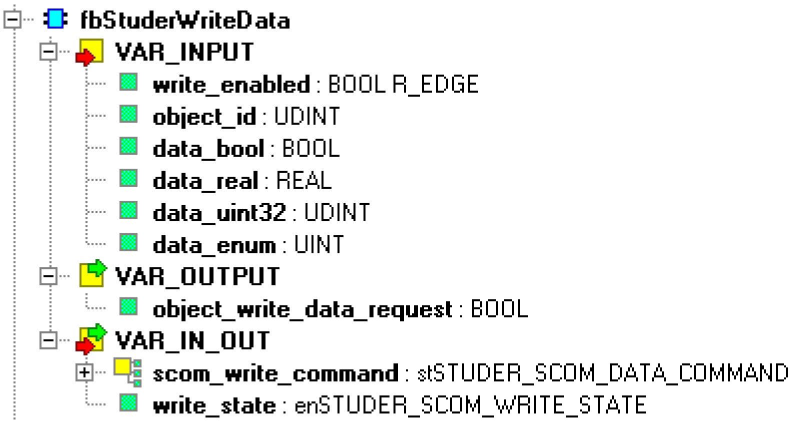
Figure 9. Representation of FB fbStuderSet1 (StuderLib) in CFC language

Table 4. fbStuderSet1 Variable descriptions:

	Variable	***Type***	***Description***
VAR_INPUT			
→	*uni_chan_code*	uint	Code of channel used for communication (see comLib)
→	*in_sychnchnization_request*	bool R_EDGE	Request to synchronize all parameters by user defined parameters in in_out struct „*studer_parameters*“
→	*in_network_scanning_request*	bool R_EDGE	Request to start Studer network scanning
→	*in_write_request*	bool R_EDGE	Request to write data to parameter according in_out „*scom_write_command*“ structure.
→	*in_object_refresh_data_request*	bool R_EDGE	Request to refresh object property values
→	*in_polling_tmers*	struct	Structure contains presets of systém timers and delays
→	*in_exchange_data_destination_address*	udint	Destination address of device to read object property values in manual mode (*in_exchange_data_mode* = STUDER_SCOM_EXCHANGE_DATA_MODE_MANUAL)
→	*in_exchange_data_mode*	enum	Mode of exchange data of object property values
VAR_OUTPUT			
↗	*out_sychnchnization_active_state*	bool	Synchronization process state of all defined parameters in in_out struct „*studer_parameters*“
↗	*out_network_scanning_active_state*	bool	Scanning process state of studer network, results saved in „*out_network_map*“ and „*out_devices_list*“ when process finished.
↗	*out_network_map*	struct	Structure contains count of all connected devices to Studer network
↗	*out_devices_list*	array struct	Structure contains connection state of all devices per address
↗	*out_object_refresh_data_state*	bool	State of refresh parameter values process
↗	*out_object_write_data_state*	enum	State of write data process
↗	*out_com_ok*	bool	Communication state {true-OK, false-faulty}
↗	*out_err*	bool	Error flag {true-error occured, false-no error}
↗	*out_err_ID*	uint	Number of error
↗	*out_err_txt*	string	Description of error
VAR_IN_OUT			
⇄	*studer_parameters*	array struct	Array contains instance of user defined parameters
⇄	*scom_write_command*	struct	Struct contains full information to write object property values

2.2 Function block fbStuderWriteData (StuderLib)

Library : STUDER Lib



```
fbStuderWriteData
  VAR_INPUT
    write_enabled : BOOL R_EDGE
    object_id : UDINT
    data_bool : BOOL
    data_real : REAL
    data_uint32 : UDINT
    data_enum : UINT
  VAR_OUTPUT
    object_write_data_request : BOOL
  VAR_IN_OUT
    scom_write_command : stSTUDER_SCOM_DATA_COMMAND
    write_state : enSTUDER_SCOM_WRITE_STATE
```

Figure 10. Structure of FB fbStuderWriteData (StuderLib)

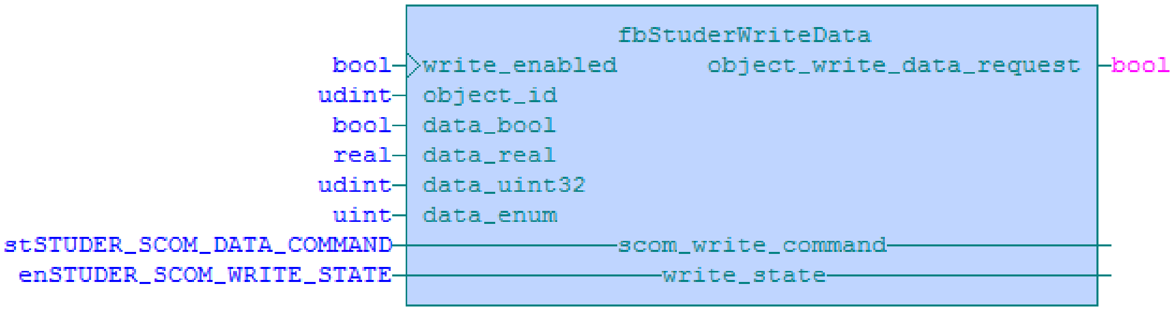


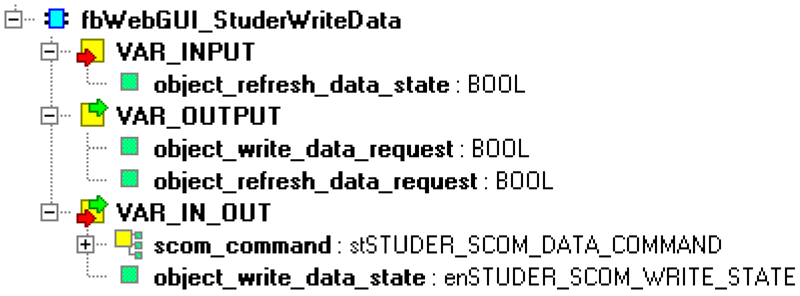
Figure 11.Representation of FB bStuderWriteData (StuderLib) in CFC language

Table 5. fbStuderWriteData Variable description:

	Variable	*Type*	*Description*
VAR_INPUT			
	write_enabled	bool R_EDGE	User application write data request
	object_id	udint	Object adress
	data_bool	bool	Object property type „bool“
	data_real	real	Object property type „real“
	data_uint32	udint	Object property type „uint32“
	data_enum	uint	Object property type „enum“
VAR_OUTPUT			
	object_write_data_request	bool	Variable indicates write object property values request.
VAR_IN_OUT			
	scom_write_command	struct	Struct contains full information to write object property values
	object_write_data_state	enum	State of write process.

2.3 Function block fbWebGUI_StuderWriteData (StuderLib)

Library : STUDER Lib



```
fbWebGUI_StuderWriteData
  VAR_INPUT
    object_refresh_data_state : BOOL
  VAR_OUTPUT
    object_write_data_request : BOOL
    object_refresh_data_request : BOOL
  VAR_IN_OUT
    scom_command : stSTUDER_SCOM_DATA_COMMAND
    object_write_data_state : enSTUDER_SCOM_WRITE_STATE
```

Figure 12. Structure of FB fbWebGUI_StuderWriteData (StuderLib)

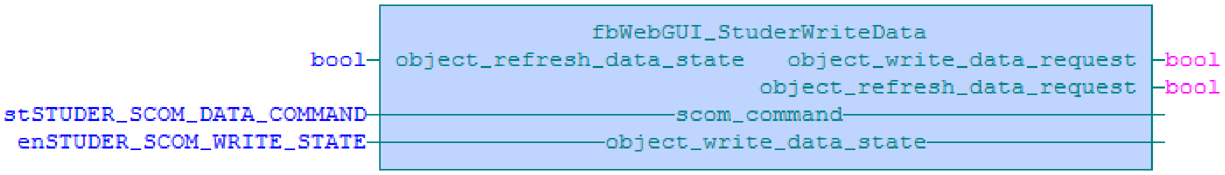

fbWebGUI_StuderWriteData
bool
object_refresh_data_state
object_write_data_request
bool
object_refresh_data_request
bool
stSTUDER_SCOM_DATA_COMMAND
scom_command
enSTUDER_SCOM_WRITE_STATE
object_write_data_state

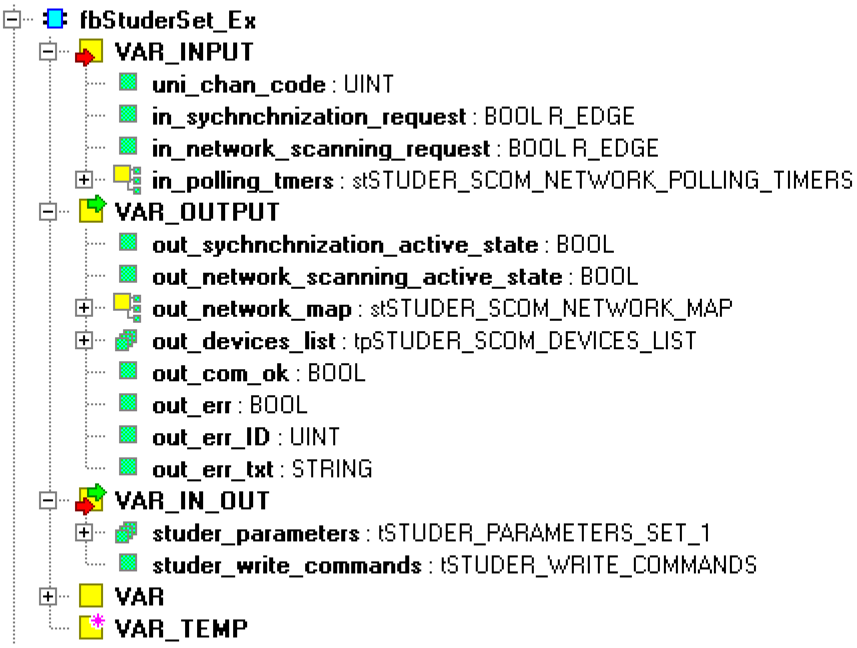
Figure 13. Representation of FB bStuderWriteData (StuderLib) in CFC language

Table 6. fbWebGUI_StuderWriteData Variable description:

	Variable	*Type*	*Description*
VAR_INPUT			
→	*object_refresh_data_state*	bool	State of last user manual commandto read property values of selected object (true- in process, false-finished)
VAR_OUTPUT			
←	*object_write_data_request*	bool	Variable indicates write object property values request.
←	*object_refresh_data_request*	bool	User manual command to read property values of selected object
VAR_IN_OUT			
↔	*scom_command*	struct	Struct contains full information to write object property values
↔	*object_write_data_state*	enum	State of write process.

2.4 Function block fbStuderSet_Ex (StuderLibEx)

Library : STUDER Lib



```
fbStuderSet_Ex
  VAR_INPUT
    uni_chan_code : UINT
    in_sychnchnization_request : BOOL R_EDGE
    in_network_scanning_request : BOOL R_EDGE
    in_polling_tmers : stSTUDER_SCOM_NETWORK_POLLING_TIMERS
  VAR_OUTPUT
    out_sychnchnization_active_state : BOOL
    out_network_scanning_active_state : BOOL
    out_network_map : stSTUDER_SCOM_NETWORK_MAP
    out_devices_list : tpSTUDER_SCOM_DEVICES_LIST
    out_com_ok : BOOL
    out_err : BOOL
    out_err_ID : UINT
    out_err_txt : STRING
  VAR_IN_OUT
    studer_parameters : tSTUDER_PARAMETERS_SET_1
    studer_write_commands : tSTUDER_WRITE_COMMANDS
  VAR
  VAR_TEMP
```

Figure 14. Structure of FB fbWebGUI_StuderWriteData (StuderLib)

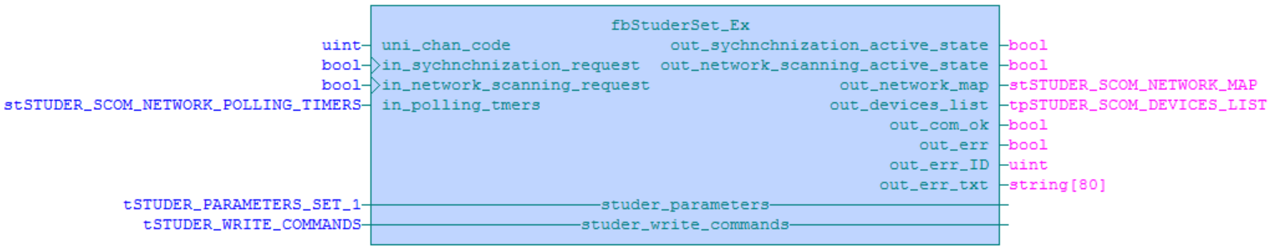


Figure 15. Representation of FB fbStuderSet_Ex (StuderLibEx) in CFC language

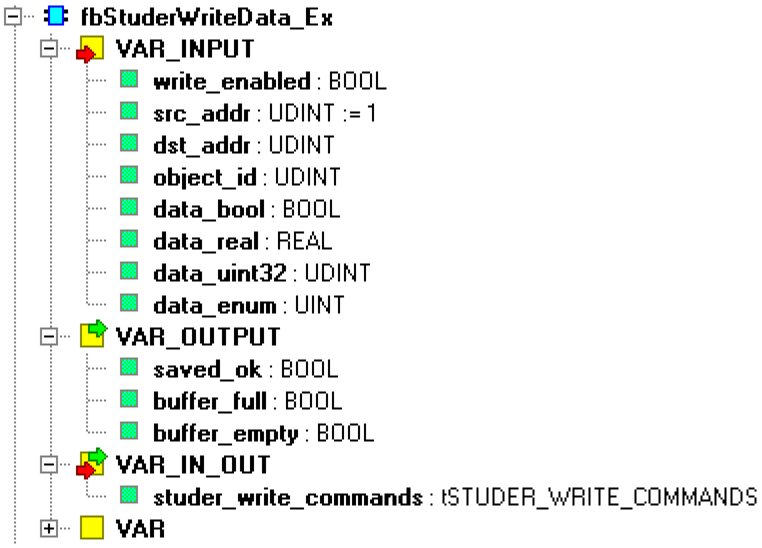
Table 7. fbStuderSet_Ex Variable descriptions:

	Variable	*Type*	*Description*
VAR_INPUT			
	uni_chan_code	uint	Code of channel used for communication (see comLib)
	in_sychnchnization_request	bool R_EDGE	Request to synchronize all parameters by user defined parameters in in_out struct „*studer_parameters*“
	in_network_scanning_request	bool R_EDGE	Request to start Studer network scanning
	in_polling_tmers	struct	Structure contains presets of systém timers and delays
VAR_OUTPUT			
	out_sychnchnization_active_state	bool	Synchronization process state of all defined parameters in

	Variable	*Type*	*Description*
			in_out struct „*studer_parameters*“
	out_network_scanning_active_state	bool	Scanning process state of studer network, results saved in „*out_network_map*“ and „*out_devices_list*“ when process finished.
	out_network_map	struct	Structure contains count of all connected devices to Studer network
	out_devices_list	array struct	Structure contains connection state of all devices per address
	out_com_ok	bool	Communication state {true-OK, false-faulty}
	out_err	bool	Error flag {true-error occured, false-no error}
	out_err_ID	uint	Number of error
	out_err_txt	string	Description of error
VAR_IN_OUT			
	studer_parameters	array struct	Array contains instance of user defined parameters
	scom_write_command	struct	Struct contains full information to write object property values

2.5 Function block fbStuderWriteData_Ex (StuderLibEx)

Library : STUDER Lib



```
fbStuderWriteData_Ex
  VAR_INPUT
    write_enabled : BOOL
    src_addr : UDINT := 1
    dst_addr : UDINT
    object_id : UDINT
    data_bool : BOOL
    data_real : REAL
    data_uint32 : UDINT
    data_enum : UINT
  VAR_OUTPUT
    saved_ok : BOOL
    buffer_full : BOOL
    buffer_empty : BOOL
  VAR_IN_OUT
    studer_write_commands : tSTUDER_WRITE_COMMANDS
  VAR
```

Figure 16. Structure of FB fbWebGUI_StuderWriteData_Ex (StuderLibEx)

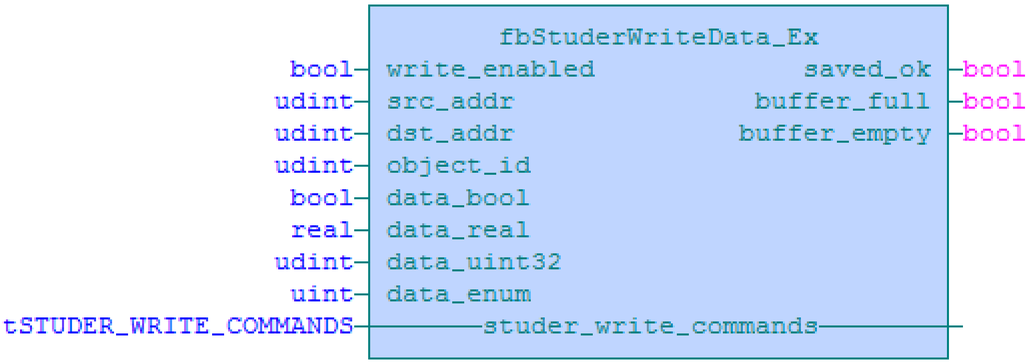


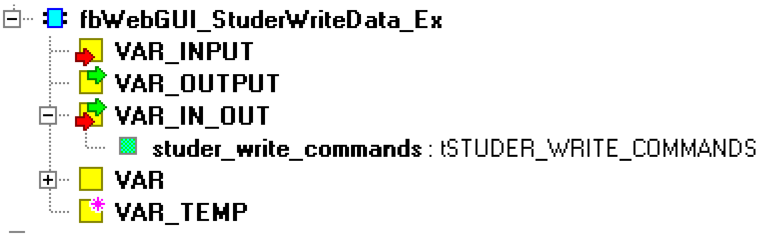
Figure 17. Representation of FB bStuderWriteData_Ex (StuderLibEx) in CFC language

Table 8. fbWebGUI_StuderWriteData_Ex Variable description:

	Variable	***Type***	***Description***
VAR_INPUT			
➡	*write _enabled*	bool	Request to save new write command record to FIFO buffer
➡	*src_addr*	udint	Source address, by default is 1 (Foxtrot address)
➡	*dst_addr*	udint	Destination address, address of Studer device
➡	*object_id*	udint	Object ID, address of Studer parameter
➡	*data_bool*	bool	Boolean value of Studer parameter
➡	*data_real*	real	Real value of Studer parameter
➡	*data_uint32*	udint	UINT32 value of Studer parameter
➡	*data_enum*	uint	Enumerated value of Studer parameter
VAR_OUTPUT			
➡	*saved_ok*	bool	New command saved in FIFO buffer
➡	*buffer_full*	bool	New command record rejected, FIFO buffer is full.
➡	*buffer_empty*	bool	Write command FIFO buffer is empty
VAR_IN_OUT			
⇄	*scom_commands*	struct	Struct contains full information to write object property values

2.6 Function block fbWebGUI_StuderWriteData_Ex (StuderLibEx)

Library : STUDER Lib



fbWebGUI_StuderWriteData_Ex
- VAR_INPUT
- VAR_OUTPUT
- VAR_IN_OUT
  - studer_write_commands : tSTUDER_WRITE_COMMANDS
- VAR
- VAR_TEMP

Figure 18. Structure of FB fbWebGUI_StuderWriteData_Ex (StuderLibEx)

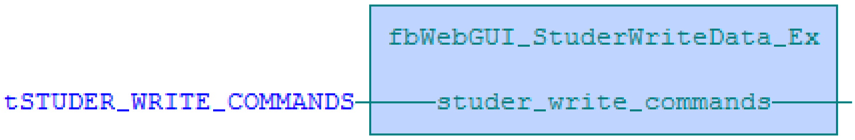


Figure 19. Representation of FB bStuderWriteData (StuderLib) in CFC language

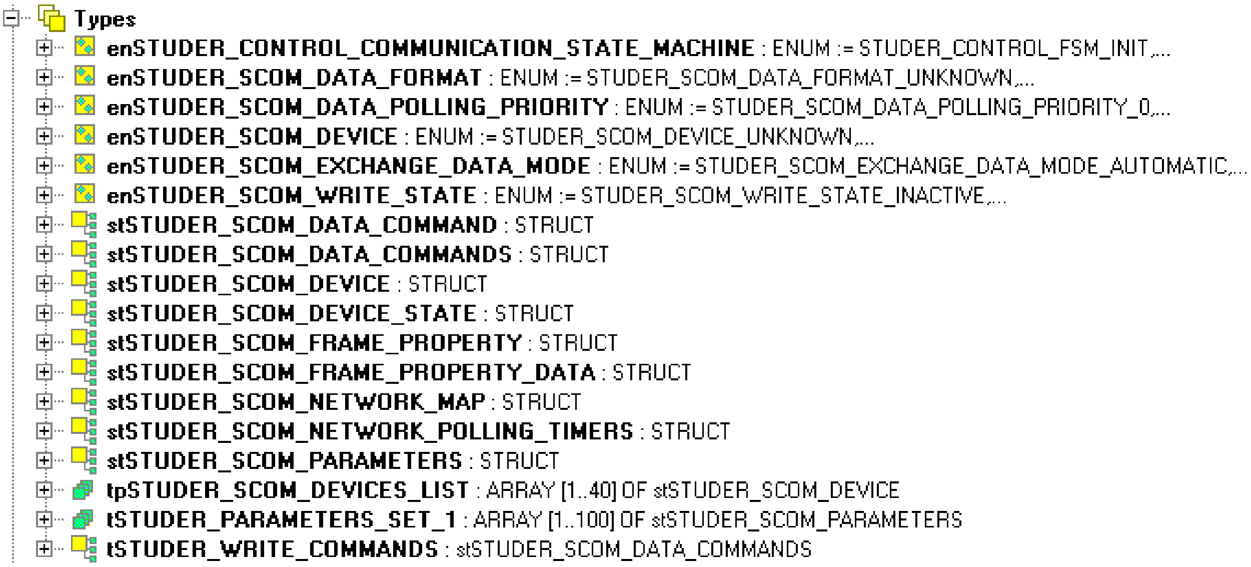
Table 9. fbWebGUI_StuderWriteData_Ex Variable description:

	Variable	*Type*	*Description*
VAR_INPUT			
VAR_OUTPUT			
VAR_IN_OUT			
⇄	*scom_commands*	struct	Struct contains full information to write object property values

3 DATA TYPES

Library : STUDER Lib

Libraryies StuderLib and StuderLibEx, haven't data types.
Library StuderLibGlobal have data types defined shared following types of variables for both StuderLib and StuderLibEx libraries:



Types
enSTUDER_CONTROL_COMMUNICATION_STATE_MACHINE : ENUM := STUDER_CONTROL_FSM_INIT,...
enSTUDER_SCOM_DATA_FORMAT : ENUM := STUDER_SCOM_DATA_FORMAT_UNKNOWN,...
enSTUDER_SCOM_DATA_POLLING_PRIORITY : ENUM := STUDER_SCOM_DATA_POLLING_PRIORITY_0,...
enSTUDER_SCOM_DEVICE : ENUM := STUDER_SCOM_DEVICE_UNKNOWN,...
enSTUDER_SCOM_EXCHANGE_DATA_MODE : ENUM := STUDER_SCOM_EXCHANGE_DATA_MODE_AUTOMATIC,...
enSTUDER_SCOM_WRITE_STATE : ENUM := STUDER_SCOM_WRITE_STATE_INACTIVE,...
stSTUDER_SCOM_DATA_COMMAND : STRUCT
stSTUDER_SCOM_DATA_COMMANDS : STRUCT
stSTUDER_SCOM_DEVICE : STRUCT
stSTUDER_SCOM_DEVICE_STATE : STRUCT
stSTUDER_SCOM_FRAME_PROPERTY : STRUCT
stSTUDER_SCOM_FRAME_PROPERTY_DATA : STRUCT
stSTUDER_SCOM_NETWORK_MAP : STRUCT
stSTUDER_SCOM_NETWORK_POLLING_TIMERS : STRUCT
stSTUDER_SCOM_PARAMETERS : STRUCT
tpSTUDER_SCOM_DEVICES_LIST : ARRAY [1..40] OF stSTUDER_SCOM_DEVICE
tSTUDER_PARAMETERS_SET_1 : ARRAY [1..100] OF stSTUDER_SCOM_PARAMETERS
tSTUDER_WRITE_COMMANDS : stSTUDER_SCOM_DATA_COMMANDS

Figure 20. List of shared data types implemented in StuderLibGlobal

Table 10. Description of data types

Name	***Type***	***Description***
enSTUDER_CONTROL_COMMUNICATION_STATE_MACHINE	enum	*Not applicable for user program, used for internal system implementation*
enSTUDER_SCOM_DATA_FORMAT	enum	Type of parameter data format
enSTUDER_SCOM_DATA_POLLING_PRIORITY	enum	Polling to data priority during endless loop read process
enSTUDER_SCOM_DEVICE	enum	Type of Studer equipment
enSTUDER_SCOM_EXCHANGE_DATA_MODE	enum	Write command operation mode (Automatic or manual)
enSTUDER_SCOM_WRITE_STATE	enum	*Not applicable for user program, used for internal system implementation*
stSTUDER_SCOM_DATA_COMMAND	struct	Structure used to write data to Studer parameters.
stSTUDER_SCOM_DATA_COMMANDS	struct	*Not applicable for user program, used for internal system implementation*
stSTUDER_SCOM_DEVICE	struct	Structure contains instance of „*stSTUDER_SCOM_DEVICE_STATE*“ structure
stSTUDER_SCOM_DEVICE_STATE	struct	Structure contains indication of Studer device states
stSTUDER_SCOM_FRAME_PROPERTY	struct	Structure contains presets and data values of Studer parameter
stSTUDER_SCOM_FRAME_PROPERTY_DATA	struct	Structure contains data value per each type parameters

stSTUDER_SCOM_NETWORK_MAP	struct	Structure contains count of connected Studer devices to network.
stSTUDER_SCOM_NETWORK_POLLING_TIMERS	struct	Structure contains different type of timers for polling to data endless loop.
stSTUDER_SCOM_PARAMETERS	struct	Structure contains Studer parameters defined from user program.
tpSTUDER_SCOM_DEVICES_LIST	array of structs	Array of „stSTUDER_SCOM_DEVICE_STATE“ structures.
tSTUDER_PARAMETERS_SET_1	array of structs	Array of „*stSTUDER_SCOM_PARAMETERS*“ structures
tSTUDER_WRITE_COMMANDS	array of structs	*Not applicable for user program, used for internal system implementation*

3.1 Enumerated data types

Library : STUDER Lib

Following enumerated data types are defined in StuderLibGlobal and shared for both StuderLib and StuderLibEx libraries

Table 11. List of enumerated variables.

Variable	***Description***
enSTUDER_SCOM_DATA_FORMAT	STUDER_SCOM_DATA_FORMAT_UNKNOWN STUDER_SCOM_DATA_FORMAT_BOOL STUDER_SCOM_DATA_FORMAT_ENUM STUDER_SCOM_DATA_FORMAT_UINT16 STUDER_SCOM_DATA_FORMAT_UINT32 STUDER_SCOM_DATA_FORMAT_FLOAT
enSTUDER_SCOM_DATA_POLLING_PRIORITY	STUDER_SCOM_DATA_POLLING_PRIORITY_0, STUDER_SCOM_DATA_POLLING_PRIORITY_1, STUDER_SCOM_DATA_POLLING_PRIORITY_2, STUDER_SCOM_DATA_POLLING_PRIORITY_3, STUDER_SCOM_DATA_POLLING_PRIORITY_4, STUDER_SCOM_DATA_POLLING_PRIORITY_5
enSTUDER_SCOM_DEVICE	STUDER_SCOM_DEVICE_UNKNOWN STUDER_SCOM_DEVICE_XTENDER STUDER_SCOM_DEVICE_BSP STUDER_SCOM_DEVICE_VARIO_TRACK STUDER_SCOM_DEVICE_VARIO_STRING STUDER_SCOM_DEVICE_XCOM_232I
enSTUDER_SCOM_EXCHANGE_DATA_MODE	STUDER_SCOM_EXCHANGE_DATA_MODE_AUTOMATIC STUDER_SCOM_EXCHANGE_DATA_MODE_MANUAL

3.2 Structure data types

Library : STUDER Lib

Following „structure“ data types are defined in StuderLibGlobal and shared for both StuderLib and StuderLibEx libraries

Table 12. stSTUDER_SCOM_DATA_COMMAND structure

Variable	*Type*	*Description*
dst_addr	udint	Sestination address of this command (Studer device address)
frame_property	struct	Presets and settings of specific object, property of object and value

Table 13. stSTUDER_SCOM_DEVICE structure

Variable	*Type*	*Description*
state	struct	Structure contains indication of Studer device states

Table 14. stSTUDER_SCOM_DEVICE_STATE structure

Variable	*Type*	*Description*
connection	bool	State of Studer device connection.(true-connected,false-disconnected)
device_type	enum	Studer device type. See „enSTUDER_SCOM_DEVICE“.
dst_addr	udint	Network address of Studer device.

Table 15. stSTUDER_SCOM_FRAME_PROPERTY structure

Variable	*Type*	*Description*
object_id	udint	Identifier of the object in is type
value	struct	See „stSTUDER_SCOM_FRAME_PROPERTY_DATA“
value_min	struct	See „stSTUDER_SCOM_FRAME_PROPERTY_DATA“ (read only)
value_max	struct	See „stSTUDER_SCOM_FRAME_PROPERTY_DATA“ (read only)

Note: ***„***value_min***“*** and ***„***value_max***“*** are read only.

Table 16. stSTUDER_SCOM_FRAME_PROPERTY_DATA structure

Variable	*Type*	*Description*
data_bool	bool	Parameter data value type „bool“
data_float	real	Parameter data value type „float“
data_uint32	udint	Parameter data value type „uint32“
data_enum	uint	Parameter data value type „enum“

Table 17. stSTUDER_SCOM_NETWORK_MAP structure

Variable	*Type*	*Description*
xtender_count	usint	Count of connected devices type „Xtender“
bsp_count	usint	Count of connected devices type „BSP“
vario_track_count	usint	Count of connected devices type „Vario track“
vario_string	usint	Count of connected devices type „Vario string“

Table 18. stSTUDER_SCOM_NETWORK_POLLING_TIMERS structure

Variable	*Type*	*Description*	*Default value*
polling_time_synchronization	time	Polling time delay for data synchronization process	0.3s
polling_time_network_scanning	time	Polling time delay for data scanning process	0.3s
polling_time_object_write	time	Time delay to start write process	2.0s
polling_time_object_refresh	time	Time delay to start refresh process	0.9s
polling_time_priority_1	time	Polling time delay for read data process priority 1	0.3s
polling_time_priority_2	time	Polling time delay for read data process priority 2	5m00s
polling_time_priority_3	time	Polling time delay for read data process priority 3	15m
polling_time_priority_4	time	Polling time delay for read data process priority 4	1h
polling_time_priority_5	time	Polling time delay for read data process priority 5	23h59m
write_responce_timeout_time	time	Responce time out from Studer device on write command	0.5s
read_responce_timeout_time	time	Responce time out from Studer device on read command	0.07s

Table 19. stSTUDER_SCOM_PARAMETERS structure

Variable	*Type*	*Description*
enabled		Status of defined parameter.
dst_addr		Destination address of Studer device
object_type	uint	STUDER_SCOM_OBJECT_TYPE_USER_INFO STUDER_SCOM_OBJECT_TYPE_PARAMETER
object_id	udint	Object address
property_id	uint	Identifier of the property we want to access for this particular object Always must be „1“
value	struct	See „stSTUDER_SCOM_FRAME_PROPERTY_DATA“
value_format	enum	See „enSTUDER_SCOM_DATA_FORMAT“
last_error	uint	Last command error code
polling_priority	enum	See „enSTUDER_SCOM_DATA_POLLING_PRIORITY“
value_sychrinized	bool	Variable indicates synchronization staus of parameter
last_update	dt	Last update date and time

4 CONSTANTS

Library : STUDER Lib

Libraryies StuderLib and StuderLibEx, haven't constants.

Library StuderLibGlobal have defines shared following constants of variables for both StuderLib and StuderLibEx libraryies:

Table 20. List of constants

Symbolic name	***Type***	***Value***	***Description***
STUDER_SCOM_DEVICE_MAX_COUNTER	usint	25	Maximum number of possible connected Studer devices to network
STUDER_SCOM_ERROR_CODE_APPLICATION_PROFILE_IS_MISSING	uint	200	Error code of application profile is missing.
STUDER_SCOM_OBJECT_TYPE_USER_INFO	uint	1	Type of Studer protocol object
STUDER_SCOM_OBJECT_TYPE_PARAMETER	uint	2	Type of Studer protocol object
STUDER_SCOM_REGISTERS_MAP_MAX_SIZE	uint	676	Maximum number of supported Studer objects.
STUDER_SCOM_SET_1_PARAMETER_MAX_COUNTER	uint	100	The maximum number of parameters that the user can define

5 GLOBAL VARIABLES

Library : STUDER Lib

Libraryies StuderLib, StuderLibEx, StuderLibGlobal haven't global variables.

6 FUNCTIONS

Library : STUDER Lib

Libraryies StuderLib, StuderLibEx, StuderLibGlobal haven't accessable functions.

7 COMMUNICATION CHANNEL SETTINGS

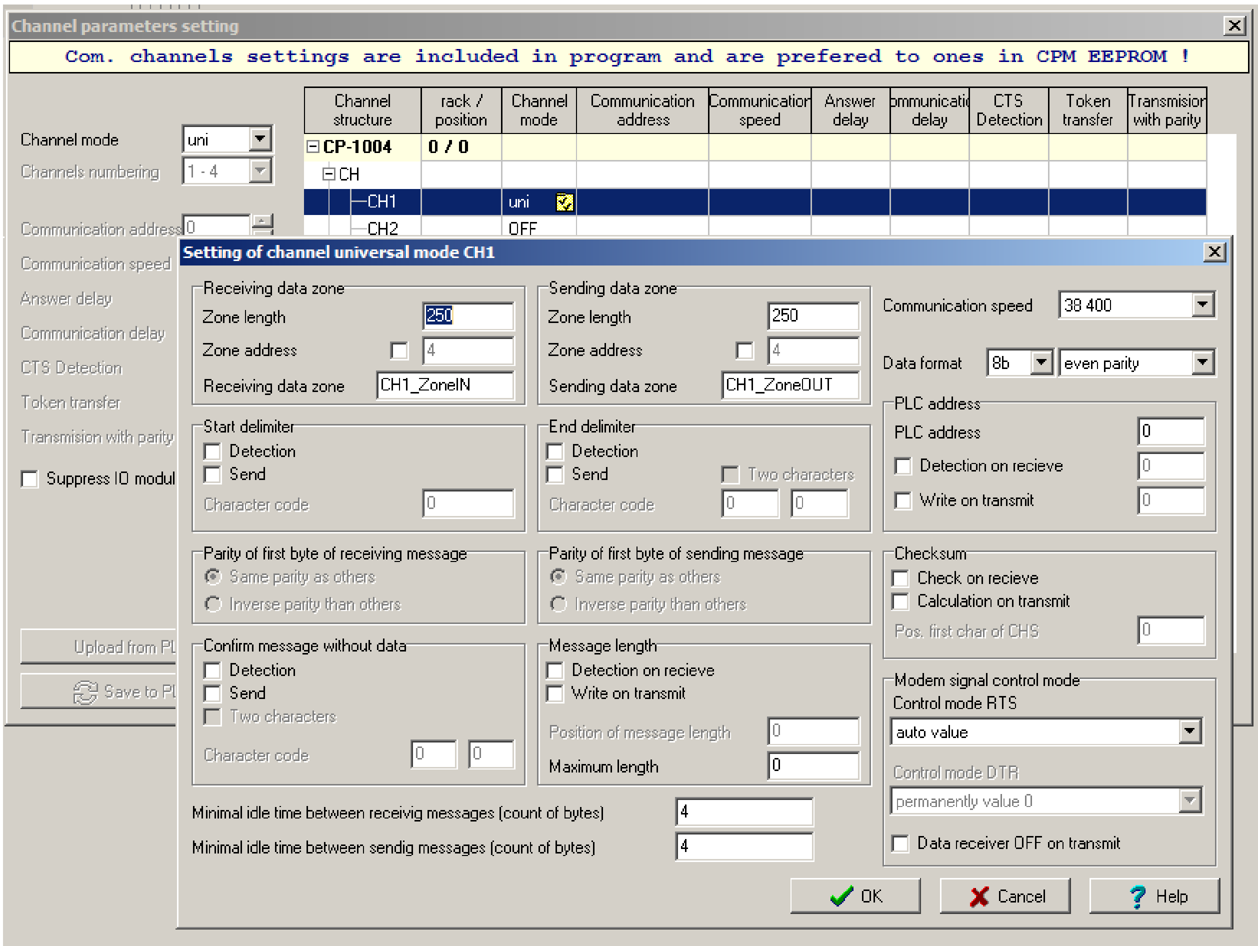
Library : STUDER Lib

STUDER Library require channels set to UNI mode. This setting can be made in the project manager.

7.1 Serial channel setting

Library : STUDER Lib

NOTE: Following global declaration applied for both StuderLib, StuderLibEx libraries.



Channel parameters setting

Com. channels settings are included in program and are prefered to ones in CPM EEPROM !

Channel mode: uni
Channels numbering: 1 - 4
Communication address: 0
Communication speed
Answer delay
Communication delay
CTS Detection
Token transfer
Transmision with parity
□ Suppress IO modul

Channel structure	rack / position	Channel mode	Communication address	Communication speed	Answer delay	ommunicati delay	CTS Detection	Token transfer	Transmision with parity
CP-1004	0 / 0								
CH									
CH1		uni							
CH2		OFF							

Setting of channel universal mode CH1

Receiving data zone
Zone length: 250
□ Zone address: 4
Receiving data zone: CH1_ZoneIN

Sending data zone
Zone length: 250
□ Zone address: 4
Sending data zone: CH1_ZoneOUT

Communication speed: 38 400
Data format: 8b, even parity

PLC address
PLC address: 0
□ Detection on recieve: 0
□ Write on transmit: 0

Start delimiter
□ Detection
□ Send
Character code: 0

End delimiter
□ Detection
□ Send
□ Two characters
Character code: 0, 0

Parity of first byte of receiving message
● Same parity as others
○ Inverse parity than others

Parity of first byte of sending message
● Same parity as others
○ Inverse parity than others

Checksum
□ Check on recieve
□ Calculation on transmit
Pos. first char of CHS: 0

Confirm message without data
□ Detection
□ Send
□ Two characters
Character code: 0, 0

Message length
□ Detection on recieve
□ Write on transmit
Position of message length: 0
Maximum length: 0

Modem signal control mode
Control mode RTS: auto value
Control mode DTR: permanently value 0
□ Data receiver OFF on transmit

Minimal idle time between receivig messages (count of bytes): 4
Minimal idle time between sendig messages (count of bytes): 4

Figure 21. Serial communication setup

IMPORTANT:

Pay attantion please to seriál communication settings: ***38400*,8bit, even parity**

Open channel parameters setting in Project manager | HW configuration (yellow icon near CPU). Set desired channel (for example CH1) to UNI mode. Open setting of channel universal mode (yellow icon at desired channel). Sending and receiving zone length should be 250 byte and minimal idle times greater than zero and control mode RTS set to auto. Communication speed, data format, parity should be set: 38400,8bit, even parity (see previous picture).

8 EXAMPLES

8.1 Parameter definition

Library : STUDER Lib

NOTE: Following global declaration applied for both StuderLib, StuderLibEx libraries.

```
VAR_GLOBAL RETAIN
   studer_polling_tmers : stSTUDER_SCOM_NETWORK_POLLING_TIMERS;
   studer_parameters    : tSTUDER_PARAMETERS_SET_1;
END_VAR
```

8.2 Studer parameters setup

Library : STUDER Lib

NOTE: Following parameter setup function applied for both StuderLib, StuderLibEx libraries.

For monitoring purpose user should select and define Studer parameters according to „Technical specification - Xtender serial protocol". Function block „fbStuderSet1" reads defined parameters value one by one in sequence determined according to priority of the parameter.
Following function „fSystemSetup" demonstrates how to define following Studer system:
1. Xtender infos - 2 pcs, address: 101,102
2. BFS infos - 1 pcs
3. VarioTrack infos - 3 pcs, address: 301,302,303

```
FUNCTION fSystemSetup : bool
  VAR_INPUT
  END_VAR

   (*********************  Xtender infos 101 ***********************)
   studer_parameters[1].enabled          := true;
   studer_parameters[1].dst_addr         := 101;
   studer_parameters[1].object_type      := STUDER_SCOM_OBJECT_TYPE_USER_INFO;
   studer_parameters[1].object_id        := 3000; (* Battery voltage *)
   studer_parameters[1].property_id      := 1;
   studer_parameters[1].value_format     := STUDER_SCOM_DATA_FORMAT_FLOAT;
   studer_parameters[1].polling_priority := STUDER_SCOM_DATA_POLLING_PRIORITY_1;

   studer_parameters[8].enabled          := true;
   studer_parameters[8].dst_addr         := 101;
   studer_parameters[8].object_type      := STUDER_SCOM_OBJECT_TYPE_USER_INFO;
   studer_parameters[8].object_id        := 3075; (*State of the neutral transfer relay*)
   studer_parameters[8].property_id      := 1;
   studer_parameters[8].value_format     := STUDER_SCOM_DATA_FORMAT_ENUM;
   studer_parameters[8].polling_priority := STUDER_SCOM_DATA_POLLING_PRIORITY_1;

   studer_parameters[10].enabled          := true;
   studer_parameters[10].dst_addr         := 101;
   studer_parameters[10].object_type      := STUDER_SCOM_OBJECT_TYPE_USER_INFO;
   studer_parameters[10].object_id        := 3081; (*Energy AC-In from of the current day *)
   studer_parameters[10].property_id      := 1;
   studer_parameters[10].value_format     := STUDER_SCOM_DATA_FORMAT_FLOAT;
   studer_parameters[10].polling_priority := STUDER_SCOM_DATA_POLLING_PRIORITY_5;

   studer_parameters[11].enabled          := true;
   studer_parameters[11].dst_addr         := 101;
   studer_parameters[11].object_type      := STUDER_SCOM_OBJECT_TYPE_USER_INFO;
   studer_parameters[11].object_id        := 3083; (* Consumers energy of the current day *)
   studer_parameters[11].property_id      := 1;
   studer_parameters[11].value_format     := STUDER_SCOM_DATA_FORMAT_FLOAT;
   studer_parameters[11].polling_priority := STUDER_SCOM_DATA_POLLING_PRIORITY_5;

   studer_parameters[13].enabled          := true;
   studer_parameters[13].dst_addr         := 101;
   studer_parameters[13].object_type      := STUDER_SCOM_OBJECT_TYPE_USER_INFO;
   studer_parameters[13].object_id        := 3136; (* Output active power *)
   studer_parameters[13].property_id      := 1;
   studer_parameters[13].value_format     := STUDER_SCOM_DATA_FORMAT_FLOAT;
   studer_parameters[13].polling_priority := STUDER_SCOM_DATA_POLLING_PRIORITY_1;

   studer_parameters[14].enabled          := true;
   studer_parameters[14].dst_addr         := 101;
   studer_parameters[14].object_type      := STUDER_SCOM_OBJECT_TYPE_USER_INFO;
   studer_parameters[14].object_id        := 3137; (* Input active power *)
   studer_parameters[14].property_id      := 1;
   studer_parameters[14].value_format     := STUDER_SCOM_DATA_FORMAT_FLOAT;
   studer_parameters[14].polling_priority := STUDER_SCOM_DATA_POLLING_PRIORITY_1;

   (**********************  Xtender infos 102 ************************)
   studer_parameters[15].enabled          := true;
   studer_parameters[15].dst_addr         := 102;
   studer_parameters[15].object_type      := STUDER_SCOM_OBJECT_TYPE_USER_INFO;
   studer_parameters[15].object_id        := 3000; (* Battery voltage *)
   studer_parameters[15].property_id      := 1;
   studer_parameters[15].value_format     := STUDER_SCOM_DATA_FORMAT_FLOAT;
   studer_parameters[15].polling_priority := STUDER_SCOM_DATA_POLLING_PRIORITY_1;

   studer_parameters[22].enabled          := true;
   studer_parameters[22].dst_addr         := 102;
   studer_parameters[22].object_type      := STUDER_SCOM_OBJECT_TYPE_USER_INFO;
   studer_parameters[22].object_id        := 3075; (* State of the neutral transfer relay *)
   studer_parameters[22].property_id      := 1;
   studer_parameters[22].value_format     := STUDER_SCOM_DATA_FORMAT_ENUM;
   studer_parameters[22].polling_priority := STUDER_SCOM_DATA_POLLING_PRIORITY_1;

   studer_parameters[24].enabled          := true;
   studer_parameters[24].dst_addr         := 102;
   studer_parameters[24].object_type      := STUDER_SCOM_OBJECT_TYPE_USER_INFO;
   studer_parameters[24].object_id        := 3081; (*Energy AC-In from of the current day *)
   studer_parameters[24].property_id      := 1;
   studer_parameters[24].value_format     := STUDER_SCOM_DATA_FORMAT_FLOAT;
   studer_parameters[24].polling_priority := STUDER_SCOM_DATA_POLLING_PRIORITY_1;

   studer_parameters[25].enabled          := true;
   studer_parameters[25].dst_addr         := 102;
   studer_parameters[25].object_type      := STUDER_SCOM_OBJECT_TYPE_USER_INFO;
   studer_parameters[25].object_id        := 3083; (*Consumers energy of the current day *)
   studer_parameters[25].property_id      := 1;
   studer_parameters[25].value_format     := STUDER_SCOM_DATA_FORMAT_FLOAT;
   studer_parameters[25].polling_priority := STUDER_SCOM_DATA_POLLING_PRIORITY_1;

   studer_parameters[27].enabled          := true;
   studer_parameters[27].dst_addr         := 102;
   studer_parameters[27].object_type      := STUDER_SCOM_OBJECT_TYPE_USER_INFO;
   studer_parameters[27].object_id        := 3136; (* Output active power *)
   studer_parameters[27].property_id      := 1;
   studer_parameters[27].value_format     := STUDER_SCOM_DATA_FORMAT_FLOAT;
   studer_parameters[27].polling_priority := STUDER_SCOM_DATA_POLLING_PRIORITY_1;

   studer_parameters[28].enabled          := true;
   studer_parameters[28].dst_addr         := 102;
   studer_parameters[28].object_type      := STUDER_SCOM_OBJECT_TYPE_USER_INFO;
   studer_parameters[28].object_id        := 3137; (* Input active power *)
   studer_parameters[28].property_id      := 1;
   studer_parameters[28].value_format     := STUDER_SCOM_DATA_FORMAT_FLOAT;
   studer_parameters[28].polling_priority := STUDER_SCOM_DATA_POLLING_PRIORITY_1;

   (********************* BFS infos ************************)
   studer_parameters[31].enabled          := true;
   studer_parameters[31].dst_addr         := 601;
   studer_parameters[31].object_type      := STUDER_SCOM_OBJECT_TYPE_USER_INFO;
   studer_parameters[31].object_id        := 7002;  (* State of Charge *)
   studer_parameters[31].property_id      := 1;
   studer_parameters[31].value_format     := STUDER_SCOM_DATA_FORMAT_FLOAT;
   studer_parameters[31].polling_priority := STUDER_SCOM_DATA_POLLING_PRIORITY_1;


   (********************* VarioTrack infos 301 ************************)
   studer_parameters[38].enabled          := true;
   studer_parameters[38].dst_addr         := 301;
   studer_parameters[38].object_type      := STUDER_SCOM_OBJECT_TYPE_USER_INFO;
   studer_parameters[38].object_id        := 11004;  (* Power of the PV generator *)
   studer_parameters[38].property_id      := 1;
   studer_parameters[38].value_format     := STUDER_SCOM_DATA_FORMAT_FLOAT;
   studer_parameters[38].polling_priority := STUDER_SCOM_DATA_POLLING_PRIORITY_1;


   (********************* VarioTrack infos 302 ************************)
   studer_parameters[45].enabled          := true;
   studer_parameters[45].dst_addr         := 302;
   studer_parameters[45].object_type      := STUDER_SCOM_OBJECT_TYPE_USER_INFO;
   studer_parameters[45].object_id        := 11004;  (* Power of the PV generator *)
   studer_parameters[45].property_id      := 1;
   studer_parameters[45].value_format     := STUDER_SCOM_DATA_FORMAT_FLOAT;
   studer_parameters[45].polling_priority := STUDER_SCOM_DATA_POLLING_PRIORITY_1;


   (********************* VarioTrack infos 303 ************************)
   studer_parameters[52].enabled          := true;
   studer_parameters[52].dst_addr         := 303;
   studer_parameters[52].object_type      := STUDER_SCOM_OBJECT_TYPE_USER_INFO;
   studer_parameters[52].object_id        := 11004;  (* Power of the PV generator *)
   studer_parameters[52].property_id      := 1;
   studer_parameters[52].value_format     := STUDER_SCOM_DATA_FORMAT_FLOAT;
   studer_parameters[52].polling_priority := STUDER_SCOM_DATA_POLLING_PRIORITY_1;

END_FUNCTION
```

IMPORTANT: If needed to read writable parameters, required to change some parameters in structure.

Example:

Declaration of read only parameter type „USER INFO“ for VarioTrack wth address 303

```
   studer_parameters[52].enabled          := true;
   studer_parameters[52].dst_addr         := 303;
   studer_parameters[52].object_type      := STUDER_SCOM_OBJECT_TYPE_USER_INFO;
   studer_parameters[52].object_id        := 11004;  (* Power of the PV generator *)
   studer_parameters[52].property_id      := 1;
   studer_parameters[52].value_format     := STUDER_SCOM_DATA_FORMAT_FLOAT;
   studer_parameters[52].polling_priority := STUDER_SCOM_DATA_POLLING_PRIORITY_1;
```

Declaration of read only parameter type „TYPE_PARAMETER“ for Xtender wth address 303

```
   studer_parameters[52].enabled          := true;
   studer_parameters[52].dst_addr         := 303;
   studer_parameters[52].object_type      := STUDER_SCOM_OBJECT_TYPE_PARAMETER;
   studer_parameters[52].object_id        := 1138;  (* Battery charge current *)
   studer_parameters[52].property_id      := STUDER_SCOM_OBJECT_TYPE_PARAMETER_ID_VALUE_QSP;
   studer_parameters[52].value_format     := STUDER_SCOM_DATA_FORMAT_FLOAT;
   studer_parameters[52].polling_priority := STUDER_SCOM_DATA_POLLING_PRIORITY_1;
```

8.3 Main program

Library : STUDER Lib

In „Main“ program defined three function blocks:
fb_StuderSet1 - function block from StuderLib library.
fb_WebGUI_StuderWriteData – local function block for STUDER Lib library web user interface control,see below.

NOTE: Following example of „Main“ program shows integration StuderLib library to user program.

```
 PROGRAM prgMain
  VAR_INPUT
  END_VAR
  VAR_OUTPUT
  END_VAR
  VAR
    (* Structurs initialization *)
    scom_write_command         : stSTUDER_SCOM_DATA_COMMAND;

    (* Function blocks *)
   fb_StuderSet1              : fbStuderSet1;
   fb_WebGUI_StuderWriteData  : fbWebGUI_StuderWriteData;
   fb_StuderWriteData         : fbStuderWriteData;


    (* local variables *)
    power_up_init             : bool;
    sychnchnization_request   : bool; //request triggered by power up procedure or by user
command
    network_scanning_request  : bool; //request triggered by power up procedure or by user
command

   power_up_init             : bool; // One time power up trigger

   (* Variables for monitoring studer parameters *)
   xtender_battery_voltage      : real;
   xtender_output_active_power  : real;
   xtender_input_active_power   : real;
   xtender_input_daily          : real;
   xtender_output_daily         : real;
   vario_track_power_of_pv      : real;


   exchange_data_destination_address : udint := 101; (* (101 to 109 a single XTH, XTM or XTS
inverter) *)
   exchange_data_mode                : enSTUDER_SCOM_EXCHANGE_DATA_MODE;
   exchange_data_mode_bool           : bool;

   object_write_data_request         : bool;

   write_enabled       : bool;
   poz_TReleON_prev    : bool;
   poz_TRele_1_ON_prev : bool;
   object_id           : udint;
   rele_state          : bool;

   poz_TReleON         : bool; // Web UI button command for relay 1
   poz_TRele_1_ON      : bool; // Web UI button command for relay 2
  END_VAR

(* One time initialization on power up *)
  if power_up_init = false then
     power_up_init := true;
      power_up_init := true;
      fSystemSetup();
      sychnchization_request  := true;
      network_scanning_request := true;
      exchange_data_mode := STUDER_SCOM_EXCHANGE_DATA_MODE_AUTOMATIC;
      studer_polling_tmers.polling_time_synchronization   := T#0.3s;
      studer_polling_tmers.polling_time_network_scanning  := T#0.3s;
      studer_polling_tmers.polling_time_object_write      := T#2.0s;
      studer_polling_tmers.polling_time_object_refresh    := T#0.9s;
      studer_polling_tmers.polling_time_priority_1        := T#00m00.1s;
      studer_polling_tmers.polling_time_priority_2        := T#1m00s;
      studer_polling_tmers.polling_time_priority_3        := T#10m;
      studer_polling_tmers.polling_time_priority_4        := T#1h;
      studer_polling_tmers.polling_time_priority_5        := T#23h59m;
  end_if;

 (* Following code shows 2 methodes to write value to studer parameters *)

(* 1. Write command from Web IU *)
   fb_WebGUI_StuderWriteData(
   scom_command              := scom_write_command,
   object_refresh_data_state := fb_StuderSet1.out_object_refresh_data_state,
   object_write_data_state   := fb_StuderSet1.out_object_write_data_state
                             );
(* 2. Write command from program *)
(* Following block shows how switch charging ralay manually from program or Web UI *)
  write_enabled := false; // reset every PLC cycle
// New write command starts when last one finished
if fb_StuderSet1.out_object_write_data_state = STUDER_SCOM_WRITE_STATE_ACTIVE then
        if poz_TReleON_prev <> poz_TReleON then
            poz_TReleON_prev := poz_TReleON;
            write_enabled    := true;
            object_id        := 3075;
            rele_state       := poz_TReleON;
        elsif poz_TRele_1_ON_prev <> poz_TRele_1_ON then
               poz_TRele_1_ON_prev := poz_TRele_1_ON;
               write_enabled    := true;
               object_id        := 3076;
               rele_state       := poz_TRele_1_ON;
        end_if;
end_if;

    (* Following FB save write command to command structure „scom_write_command“ *)
fb_StuderWriteData(write_enabled      := write_enabled,
                   write_state        := fb_StuderSet1.out_object_write_data_state,
                   object_id          := object_id,
                   data_bool          := rele_state,
                   scom_write_command := scom_write_command
                   );
// Save request from both sources
object_write_data_request := fb_WebGUI_StuderWriteData.object_write_data_request OR
                             fb_StuderWriteData.object_write_data_request;


   fb_StuderSet1(uni_chan_code                        := CH1_uni,
                 in_sychnchization_request            := sychnchization_request,
                 in_network_scanning_request          := network_scanning_request,
                 in_object_refresh_data_request       := fb_WebGUI_StuderWriteData.object_refresh_data_request,
                 in_write_request                     := object_write_data_request,
                 studer_parameters                    := studer_parameters,
                 scom_write_command                   := scom_write_command,
                 in_polling_tmers                     := studer_polling_tmers,
                 in_exchange_data_destination_address := exchange_data_destination_address,
                 in_exchange_data_mode                := exchange_data_mode
                );

sychnchization_request     := false;
network_scanning_request   := false;
```

```
 xtender_battery_voltage       := studer_parameters[1].value.data_float;
   xtender_output_active_power := studer_parameters[13].value.data_float +
                                  studer_parameters[27].value.data_float;
   xtender_input_active_power  := studer_parameters[14].value.data_float +
                                  studer_parameters[28].value.data_float;
   xtender_output_daily        := studer_parameters[11].value.data_float +
                                  studer_parameters[25].value.data_float;
   xtender_input_daily         := studer_parameters[10].value.data_float +
                                  studer_parameters[24].value.data_float;
   vario_track_power_of_pv     := studer_parameters[38].value.data_float +
                                  studer_parameters[45].value.data_float +
                                  studer_parameters[52].value.data_float;

END_PROGRAM
```

NOTE: Following example of „Main“ program shows integration StuderLibEx library to user program.

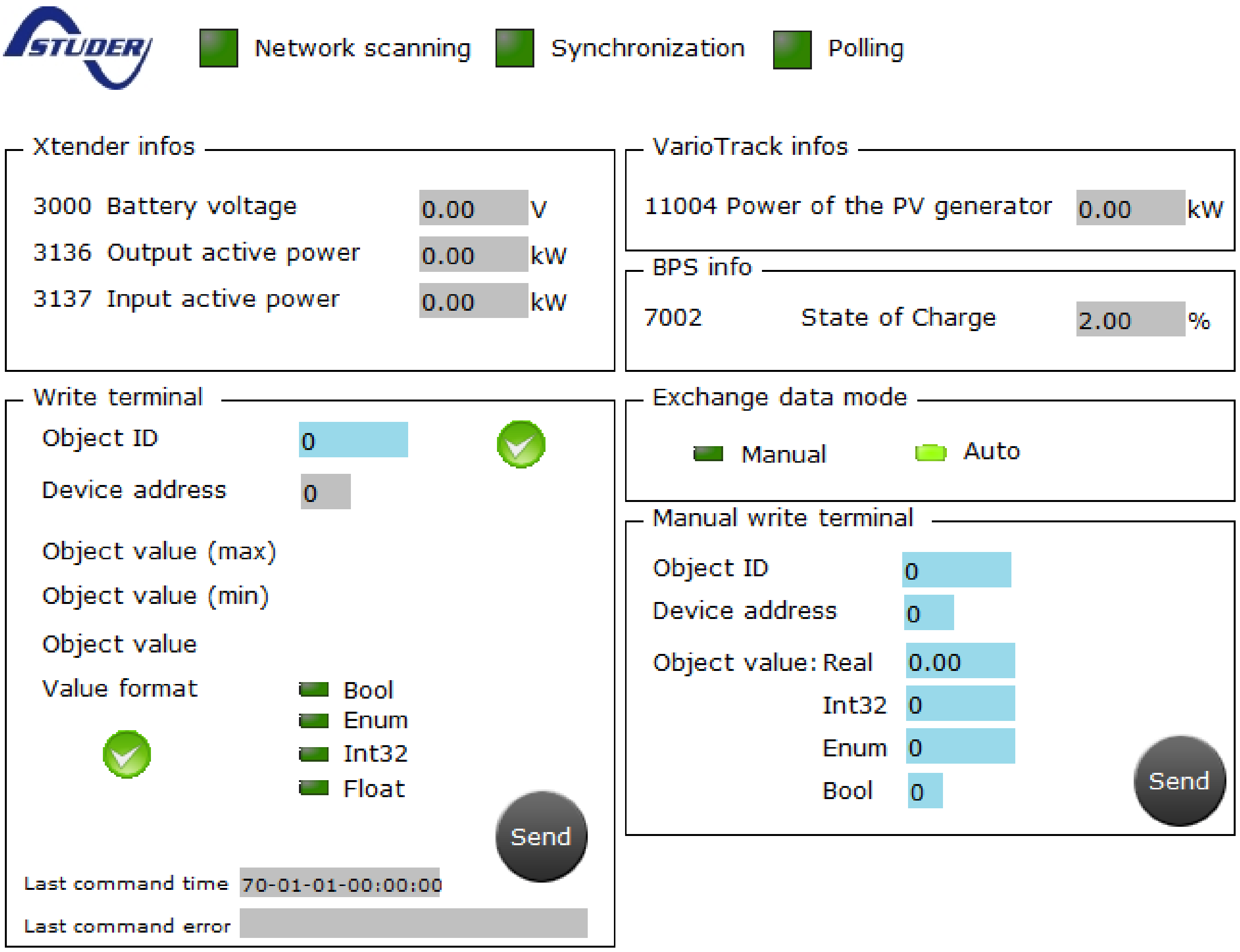
```
 PROGRAM prgMain
  VAR_INPUT
  END_VAR
  VAR_OUTPUT
  END_VAR
  VAR
    (* Structurs initialization *)
    scom_write_command       : stSTUDER_SCOM_DATA_COMMAND;

    (* Function blocks *)
   fb_StuderSet_Ex           : fbStuderSet_Ex;
   fb_WebGUI_StuderWriteData : fbWebGUI_StuderWriteData_Ex;
   fb_StuderWriteData        : fbStuderWriteData_Ex;


    (* local variables *)
    power_up_init            : bool;
    sychnchnization_request  : bool; //request triggered by power up procedure or by user
command
    network_scanning_request : bool; //request triggered by power up procedure or by user
command

   write_48_2          : bool;
   write_12_8          : bool;
  END_VAR

(* One time initialization on power up *)
  if power_up_init = false then
     power_up_init := true;
      power_up_init := true;
      fSystemSetup();
      sychnchnization_request  := true;
      network_scanning_request := true;
      exchange_data_mode := STUDER_SCOM_EXCHANGE_DATA_MODE_AUTOMATIC;
      studer_polling_tmers.polling_time_synchronization   := T#0.3s;
      studer_polling_tmers.polling_time_network_scanning  := T#0.3s;
      studer_polling_tmers.polling_time_object_write      := T#2.0s;
      studer_polling_tmers.polling_time_object_refresh    := T#0.9s;
      studer_polling_tmers.polling_time_priority_1        := T#00m00.1s;
      studer_polling_tmers.polling_time_priority_2        := T#1m00s;
      studer_polling_tmers.polling_time_priority_3        := T#10m;
      studer_polling_tmers.polling_time_priority_4        := T#1h;
      studer_polling_tmers.polling_time_priority_5        := T#23h59m;
  end_if;
```

```
   fb_WebGUI_StuderWriteData(studer_write_commands := studer_write_commands );
  (* Zapisovani dat z aplikace *)

   if write_48_2 = true then
      write_48_2 := false;
       fb_StuderWriteData(write_enabled         := true,
                          src_addr              := 1,
                          dst_addr              := 101,
                          object_id             := 1138,
                          data_real             := 48.2,
                          studer_write_commands := studer_write_commands
                         );

   elsif write_12_8 = true then
         write_12_8 := false;
         fb_StuderWriteData(write_enabled         := true,
                            src_addr              := 1,
                            dst_addr              := 101,
                            object_id             := 1138,
                            data_real             := 12.8,
                            studer_write_commands := studer_write_commands
                       );
   end_if;

fb_StuderSet_Ex(uni_chan_code                        := CH1_uni,
                in_sychnchnization_request           := sychnchnization_request,
                in_network_scanning_request          := network_scanning_request,
                studer_parameters                    := studer_parameters,
                in_polling_tmers                     := studer_polling_tmers,
                studer_write_commands                := studer_write_commands
               );

sychnchnization_request     := false;
network_scanning_request    := false;


END_PROGRAM
```

8.4 Example of simple web user interface contol.

Library : STUDER Lib

Please visit our web site to download following test project



🟩 Network scanning 🟩 Synchronization 🟩 Polling

Xtender infos

3000	Battery voltage	0.00	V
3136	Output active power	0.00	kW
3137	Input active power	0.00	kW

VarioTrack infos

11004	Power of the PV generator	0.00	kW

BPS info

7002	State of Charge	2.00	%

Write terminal

Object ID 0 ✓

Device address 0

Object value (max)

Object value (min)

Object value

Value format ● Bool ● Enum ● Int32 ● Float

✓

Send

Last command time 70-01-01-00:00:00

Last command error

Exchange data mode

● Manual ● Auto

Manual write terminal

Object ID 0

Device address 0

Object value: Real 0.00

Int32 0

Enum 0

Bool 0

Send

Figure 22. Example of web GUI