IEC60870 Configurator

for RTU32x Series

User Manual

for IEC60870 configurator V1.50

version 2.50 / March 2014 / Doc 40236

BRODERSEN
simplifying systems

Table of Content

1. Customer Information

Copyright Notice

Trademark Acknowledgement

Brand and product names are trademarks or registered trademarks of their respective owners.

Disclaimer

Brodersen A/S reserves the right to make changes, without notice, to any product, including circuits and/or software described or contained in this manual in order to improve design and/or performance. Brodersen A/S assumes no responsibility or liabilities for the use of the described product(s), conveys no license or title under any patent, copyright, or mask work rights to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified. Applications that are described in this manual are for illustration purposes only.
Brodersen A/S makes no representation or warranty that such application will be suitable for the specified use without further testing or modification.

Life Support Policy

BRODERSEN A/S's PRODUCTS ARE NOT FOR USE AS CRITICAL COMPONENTS, IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE PRIOR WRITTEN APPROVAL OF BRODERSEN A/S.
As used herein:
1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into body, or (b) support or sustain life and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

Brodersen Customer Services

Your satisfaction is our primary concern. Here is a guide to Brodersen customer services. To ensure you get the full benefit of our services, please follow the instructions below carefully.

Technical Support

We want you to get the maximum performance from your products. So if you run into technical difficulties, we are here to help. For the most frequently asked questions, you can easily find answers in the product documentation. These answers are normally a lot more detailed than the ones we can give over the phone. So please consult this manual first.
To receive the latest version of the user manual, please visit our Web site at:

http://www.brodersen.com,

Choose the product in question under product search and under each product you will find accompanying data sheets, manuals, user guides etc.
If you still cannot find the answer, gather all the information or questions that apply to your problem, and with the product close at hand, call your dealer. Our distributors are well trained and ready to give you the support you need to get the most from your Brodersen products. In fact, most problems reported are minor and are able to be easily solved over the phone.
In addition, technical support is available from Brodersen engineers every business day. We are always ready to give advice on application requirements or specific information on the installation and operation of any of our products. Please do not hesitate to call or e-mail us on support@brodersen.com .

Denmark:

Brodersen A/S
Islevdalvej 187
DK-2610 Roedovre
Tel.: +45 45 35 26 27
Fax: +45 45 35 26 29
sales@brodersen.com
www.brodersen.com

Product Warranty

Brodersen warrants to you, the original purchaser, that each of its products will be free from defects in materials and workmanship for two years from the date of purchase.

This warranty does not apply to any products which have been repaired or altered by persons other than repair personnel authorized by Brodersen, or which have been subject to misuse, abuse, accident or improper installation. Brodersen assumes no liability under the terms of this warranty as a consequence of such events. Because of Brodersen's high quality control standards and rigorous testing, most of our customers never need to use our repair service. If a Brodersen product is defective, it will be repaired or replaced at no charge during the warranty period. For out-of-warranty repairs, you will be billed according to the cost of replacement materials, service time, and freight. Please consult your distributor for more details. If you think you have a defective product, follow these steps:

1. Collect all the information about the problem encountered. (For example, Product type and s/n, hardware and software version etc.) Note anything abnormal and describe the error in a product failure report.
2. Call your distributor and describe the problem. Please have your manual, product, and any helpful information readily available.
3. If your product is diagnosed as defective, make arrangement with your distributor about this.
4. Carefully pack the defective product, a complete failure report and a photocopy of proof of purchase date (such as your sales receipt) in a shippable container. A product returned without proof of the purchase date is not eligible for warranty service.
5. Ship it to your distributor.

2. Introduction

Before you Begin Please Note That:

- The IEC60870 Drivers for RTU32 is made for utility applications with the RTU32 working as an Outstation, System Gateway or small data concentrator.
- The IEC60870 Drivers are a standard option in the RTU32 Series.
- The IEC60870 Configuration tool does in general take IEC60870 application layer data from a MS Excel Work Book file and generates a RTU32 application program. It supports the most commonly used driver features and does automatically generate the RTU32 application program code and download it into the RTU32. If you have special requirements you can edit and/or add any wanted IEC60870 function in Brodersen WorkSuite.
- The basic IEC60870 Drivers in RTU32 are implemented as link drivers. That means that you in the Brodersen WorkSuite application-programming environment do develop the application layer with a standard set of functions. Please note that this gives you the advantage to adjust the drivers to almost any requirements within the scope of the IEC60870 driver specifications. But it also requires knowledge of the protocol options.
- You must **NOT** have the RTU project you are generating open in the Brodersen WorkSuite. When you want to generate code, be sure that Brodersen WorkSuite is closed.
- It is required that the Brodersen WorkSuite version 1.0.0.0 or newer or STRATON WorkBench version 8.5 or newer is installed on your PC to run the Configurator tool.
- If you just want to create variables, the Code generator is also a useful tool.
- The IEC60870 Configurator has an upper limit in number of ADSUs to be created for a driver. Please see section about Work sheet – Variables for details.

RTU32 IEC60870 Configurator Features

The RTU32 IEC60870 Configurator is a tool that contains three basic elements;

Excel Work Book

The IEC60870 driver parameters, variables and ASDU types are entered by the user in an Excel sheet. Brodersen provide a standard Excel sheet for setting up all necessary parameters in an RTU32 driver. If required you can define or use your own Excel sheet if you get an introduction and training of the Configurator setup details.

IEC60870 Code Generator

The IEC60870 Code Generator is the Brodersen WorkSuite program generator which takes user configuration values defined in a MS Excel sheet and build up a Brodersen WorkSuite program, download it to the RTU32 and get it started. As the Code Generator uses the Brodersen WorkSuite compiler, it is required that the Brodersen WorkSuite is installed on your PC.

XML Database File – ONLY CHANGED OR EDITED WHEN USED IN SPECIAL APPLICATIONS

The IEC60870 Application layer is written as Brodersen WorkSuite structured text (ST) code. All this code, including basic driver settings and functions, and definitions of Excel sheet data readings are stored in a XML file.
The database XML file does include all settings used by the Configurator Tool – formatted and in clear text. An experienced user can enter own special configuration parameters and functions to this XML file.

IEC60870-5-104 Enhanced functions / NUC functions OPTIONAL

The RTU32 and the IEC60870-5-10x Code Generator supports the latest IEC60870-5-104 protocol specifications and NUC (Norwegian User Convention) requirements, which includes support for redundancy groups, several Client connections and multiple COA. The Code generator supports in details:

Up to 6 redundancy groups

- Each redundancy group has up to 8 prioritized transmission data queues.
- Each redundancy group is generated in a separate Excel sheet.

Each redundancy group support up to 6 Client connections

- The Clients are specified as allowed IP connections.
- Timing setting can be adjusted individually for each Client connection.
- The IEC60870-5-104 Server support up to 6 local COAs. This means that the RTU32 can be given up to 6 different COA addresses that are assigned in the IOA list in the Excel sheet.

Multiple COAs in an RTU32

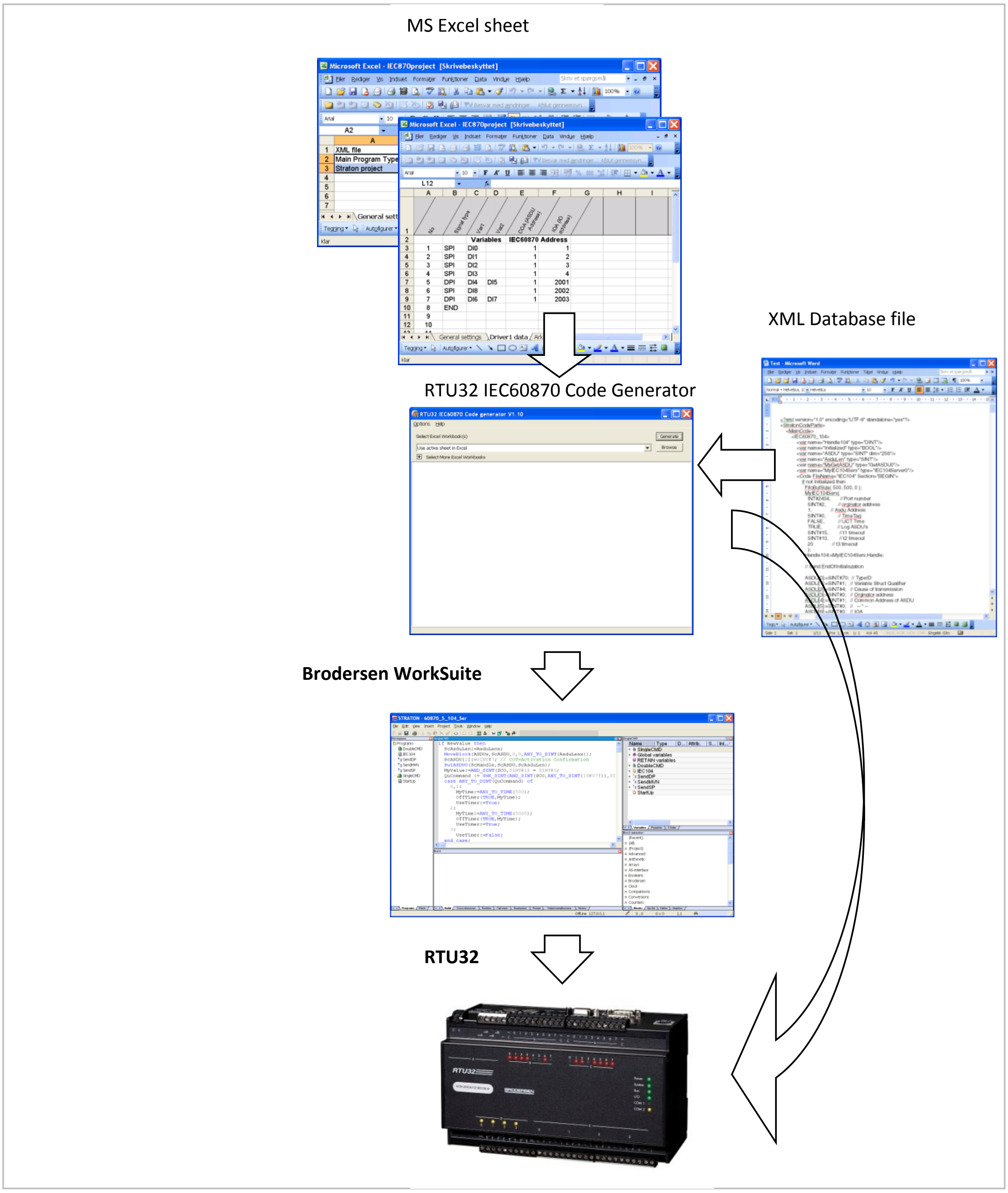
- Any ASDU type can be assigned its own COA address. It means that one physical RTU32 on the same link can represent several COAs. You just assign the COA in the ASDU list and the RTU32 will report as several single units with its own COA. It means that if you send a General Interrogation for a specific COA, it will send only the ASDUs defined for the specific COA. To read all COAs, the broadcast address 0xFFFF must be used.

Data transmission in Reverse Direction

- The actual Server link connection can receive and monitor data send in reverse direction – just like it was working as a Controlling Station (Client).

Block Diagram

The figure below shows Configurator tool components and their links.


MS Excel sheet
Microsoft Excel - IEC870project [Skrivebeskyttet]
XML file
Main Program Type
Straton project
General settings
Driver1 data
XML Database file
RTU32 IEC60870 Code Generator
RTU32 IEC60870 Code generator V1.10
Select Excel Workbook(s)
Use active sheet in Excel
Select More Excel Workbooks
Generate
Browse
Brodersen WorkSuite
RTU32

No	Signal type	Var1	Var2	COA (ASDU Address)	IOA (IO address)
		Variables		IEC60870 Address	
1	SPI	DI0		1	1
2	SPI	DI1		1	2
3	SPI	DI2		1	3
4	SPI	DI3		1	4
5	DPI	DI4	DI5	1	2001
6	SPI	DI8		1	2002
7	DPI	DI6	DI7	1	2003
8	END				
9					
10					

Supported IEC60870 Drivers in the Configurator

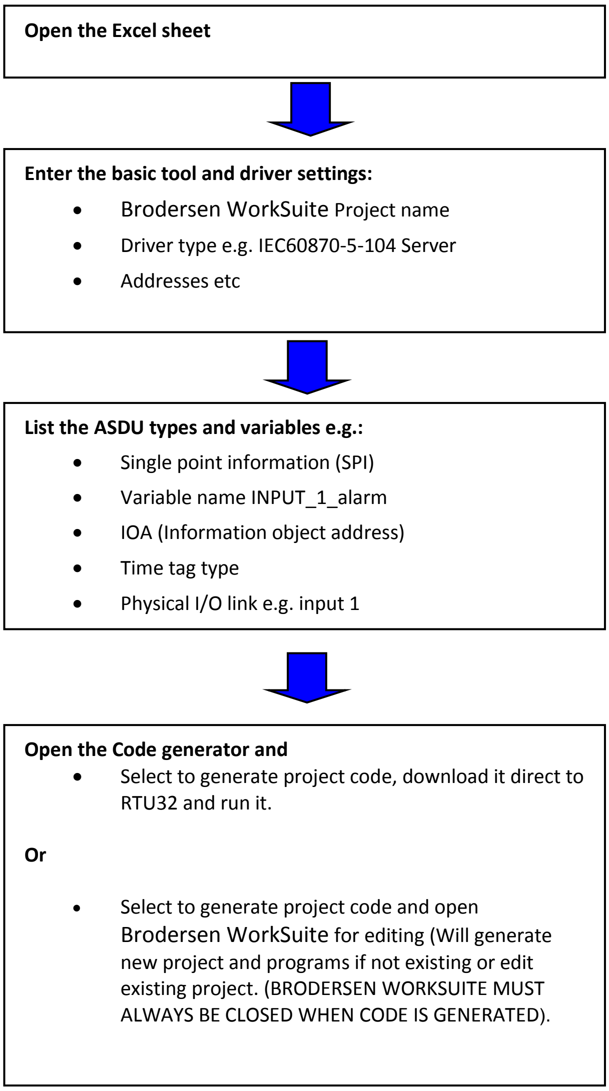
The RTU32 IEC870 Configurator supports the following IEC60870 drivers;

- **IEC60870-5-101 Slave – including dial connections**
- **IEC60870-5-101 Master – single connection only**
- **IEC60870-5-104 Server – communication in Normal and Reverse direction**.
- **IEC60870-5-104 Client – communication in Normal and Reverse direction**.
- **IEC60870-5-103 Master – with a link to IEC60870-5-104 Server**
- **IEC60870-5-103 Master**

The RTU32 IEC60870 Interoperability documents define all driver parameters supported by the IEC60870 Configurator program.
The Interoperability does NOT reflect the basic Brodersen WorkSuite link driver options as the application layer is built up in Brodersen WorkSuite and supports almost any possible function defined in the IEC60870-5-101 and -104 standards.

Work Flow

When developing an IEC60870 Driver in the RTU32, the work flow would be as shown below:



Open the Excel sheet

Enter the basic tool and driver settings:

- Brodersen WorkSuite Project name
- Driver type e.g. IEC60870-5-104 Server
- Addresses etc

List the ASDU types and variables e.g.:

- Single point information (SPI)
- Variable name INPUT_1_alarm
- IOA (Information object address)
- Time tag type
- Physical I/O link e.g. input 1

Open the Code generator and

- Select to generate project code, download it direct to RTU32 and run it.

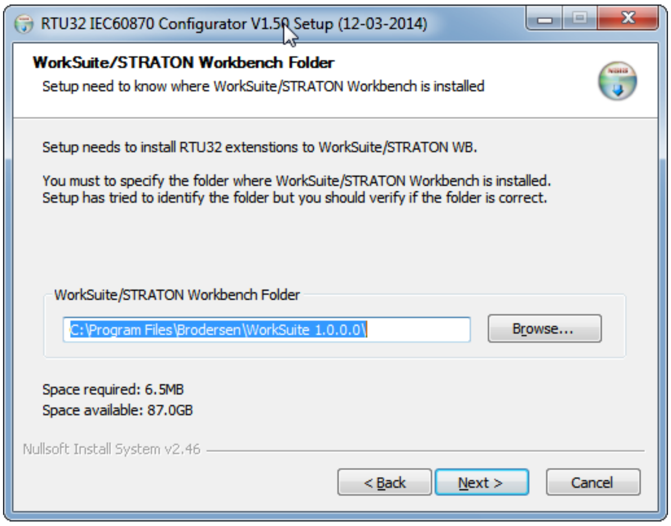
Or

- Select to generate project code and open Brodersen WorkSuite for editing (Will generate new project and programs if not existing or edit existing project. (BRODERSEN WORKSUITE MUST ALWAYS BE CLOSED WHEN CODE IS GENERATED).

3. Installation and Operation

Installation Details

Insert the Installation CD in your PC and start the installation application. Make sure that Brodersen WorkSuite is closed and follow the guidelines in the installation program.



RTU32 IEC60870 Configurator V1.50 Setup (12-03-2014)

WorkSuite/STRATON Workbench Folder

Setup need to know where WorkSuite/STRATON Workbench is installed

Setup needs to install RTU32 extenstions to WorkSuite/STRATON WB.

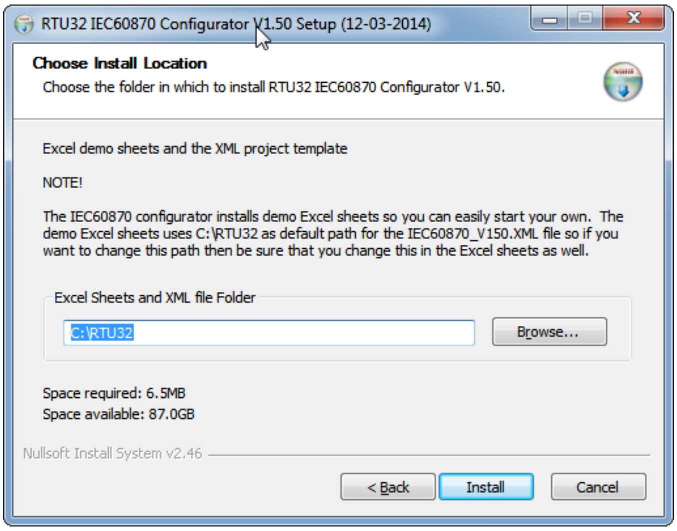
You must to specify the folder where WorkSuite/STRATON Workbench is installed.
Setup has tried to identify the folder but you should verify if the folder is correct.

WorkSuite/STRATON Workbench Folder: C:\Program Files\Brodersen\WorkSuite 1.0.0.0\

Space required: 6.5MB
Space available: 87.0GB

It is important that the IEC60870 Code generator is installed in the directory with WorkSuite or STRATON main program.
If you use Brodersen WorkSuite the path in Windows7 is normally: C:\Program Files\Brodersen\WorkSuite 1.x.x.x.

Next step is to define the location of where to install the Excel demo sheets and the XML project template file.



RTU32 IEC60870 Configurator V1.50 Setup (12-03-2014)

Choose Install Location

Choose the folder in which to install RTU32 IEC60870 Configurator V1.50.

Excel demo sheets and the XML project template

NOTE!

The IEC60870 configurator installs demo Excel sheets so you can easily start your own. The demo Excel sheets uses C:\RTU32 as default path for the IEC60870_V150.XML file so if you want to change this path then be sure that you change this in the Excel sheets as well.

Excel Sheets and XML file Folder: C:\RTU32

Space required: 6.5MB
Space available: 87.0GB

It is recommended to keep the default location: C:\RTU32

When the installation is finished, you will find a shortcut to the IEC60870 Code generator on the desktop and in the START-PROGRAMS-BRODERSEN-RTU32 IEC60870 CONFIGURATOR v1.50. Here you will also find a shortcut to the Excel sheet used for configuration and setup parameters.

The IEC60870 Configurator Setup will install the following on your PC:

Description	Files	Default folder
Excel sheet examples for the 6 driver types	IEC60870_101Slave_example_V150.xls IEC60870_104Server_example_V150.xls IEC60870_101Master_example_V150.xls IEC60870_104Client_example_V150.xls IEC60870_104Server_103Master_example_V150.xls IEC60870_103Master_example_V150.xls	C:\RTU32
XML database file	IEC60870_150.XML	C:\RTU32
IEC60870 Application Code Generator program	RTU32_870Gen.EXE	C:\Program Files\Brodersen\WorkSuite 1.0.0.0 *)
RTU32 IEC870 Code Generator help file (this doc.)	40236.PDF	C:\Program Files\Brodersen\IEC60870ConV1.50 *)

*) When using Brodersen WorkSuite. If using STRATON WorkBench the files shall be installed where you have your STRATON software installed – according to your selection during installation.

Brodersen WorkSuite should be version 1.0.0.0 or newer.
STRATON WorkBench should be version 8.5 – and you shall have the RTU32 CD v1.50 or newer installed.
RTU32 version is required to be version 1.52 – or newer.

Operation Procedure

- After installing the IEC60870 Configurator files, you are ready to go.
- To set-up the IEC60870 driver details, open a relevant IEC60870 example Excel file.
- Open the Main work sheet to setup the main settings for the WorkSuite Project name and IP address of RTU32. You may also define your own settings for used Excel sheet name.
- Open the Variables work sheet. Define all the variables, assign descriptions, ADSU type addresses and settings etc. After adding the data you must save the file.
- Now you are ready to generate the code for the RTU32. Open the Code Generator and activate Generate to create the RTU32 application program. You will have to select in the Option menu if you do not want to open the Brodersen WorkSuite with the generated program. If you select to open Brodersen WorkSuite (default), you must make sure that it is not already running – if it is running it must be closed.
- If you select to download the configuration to the RTU32, the application program will be compiled, downloaded and started in the RTU32 with the IP address defined in the Excel sheet Main work sheet.

4. MS Excel File – the general work sheet

Workbook Structure

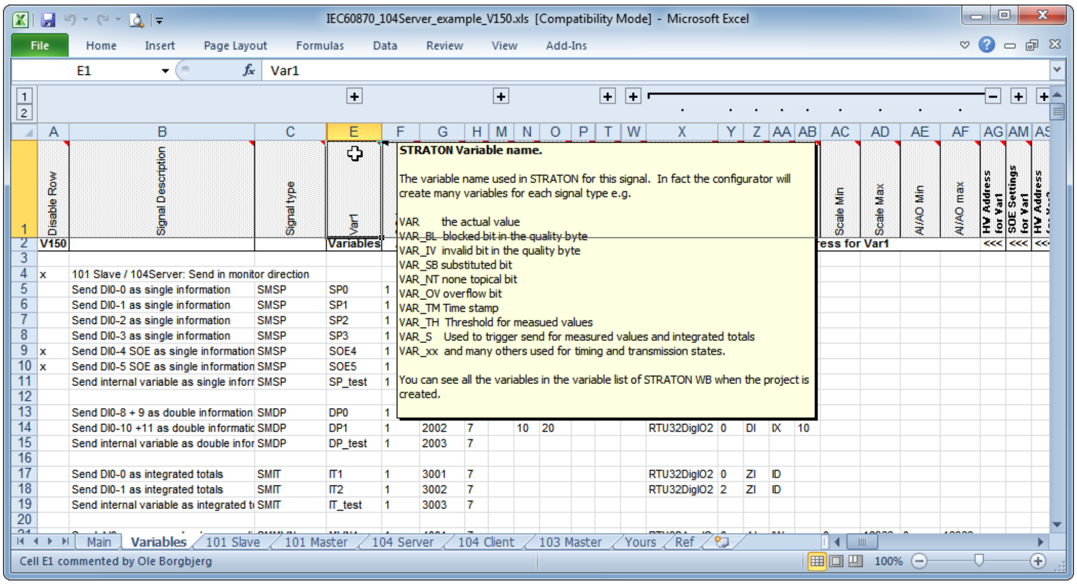
The Excel file is the basic user interface for the Configurator. All settings for setting up an IEC60870 driver are added here. The purpose for using the Excel sheets is to provide a well know working environment and give the user the opportunity to add own documentation etc.
The Excel workbook file includes several work sheets:

- Main
- Variables
- 101 Slave
- 101 Master
- 104 Server
- 104 Client
- 103 Master linked to 104 Server
- 103 Master

- Yours
- Refs (ASDU type reference list to ASDU TYPE IDENT numbers)

How to get Help

The Excel sheets contain a lot of pop up help windows. A help box will pop up when the mouse is moved over the cells that contain the little red triangle in the right upper corner. See the picture below:



E1: Var1

STRATON Variable name.

The variable name used in STRATON for this signal. In fact the configurator will create many variables for each signal type e.g.

VAR the actual value
VAR_BL blocked bit in the quality byte
VAR_IV invalid bit in the quality byte
VAR_SB substituted bit
VAR_NT none topical bit
VAR_OV overflow bit
VAR_TM Time stamp
VAR_TH Threshold for measued values
VAR_S Used to trigger send for measured values and integrated totals
VAR_xx and many others used for timing and transmission states.

You can see all the variables in the variable list of STRATON WB when the project is created.

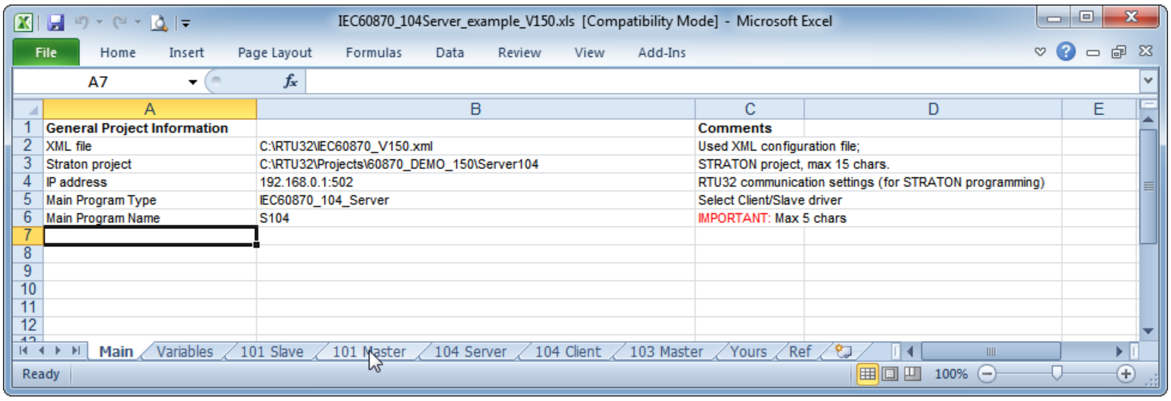
	A	B	C	E	F	G	H	N	O	X	Y	Z	AA	AB
1	Disable Row	Signal Description	Signal type	Var1										
2	V150			Variables										
3														
4	x	101 Slave / 104Server: Send in monitor direction												
5		Send DI0-0 as single information	SMSP	SP0	1									
6		Send DI0-1 as single information	SMSP	SP1	1									
7		Send DI0-2 as single information	SMSP	SP2	1									
8		Send DI0-3 as single information	SMSP	SP3	1									
9	x	Send DI0-4 SOE as single information	SMSP	SOE4	1									
10	x	Send DI0-5 SOE as single information	SMSP	SOE5	1									
11		Send internal variable as single inforr	SMSP	SP_test	1									
12														
13		Send DI0-8 + 9 as double information	SMDP	DP0	1									
14		Send DI0-10 +11 as double informatic	SMDP	DP1	1	2002	7	10	20	RTU32DigIO2	0	DI	IX	10
15		Send internal variable as double infor	SMDP	DP_test	1	2003	7							
16														
17		Send DI0-0 as integrated totals	SMIT	IT1	1	3001	7			RTU32DigIO2	0	ZI	ID	
18		Send DI0-1 as integrated totals	SMIT	IT2	1	3002	7			RTU32DigIO2	2	ZI	ID	
19		Send internal variable as integrated t	SMIT	IT_test	1	3003	7							
20														

AC	AD	AE	AF	AG	AM	AS
Scale Min	Scale Max	AI/AO Min	AI/AO max	HW Address for Var1	SOE Settings for Var1	HW Address
ess for Var1				<<<	<<<	<<<

Cell E1 commented by Ole Borgbjerg

Work Sheet – Main

In the Main work sheet all basic driver setting are entered – see figure below:



	A	B	C
1	**General Project Information**		**Comments**
2	XML file	C:\RTU32\IEC60870_V150.xml	Used XML configuration file;
3	Straton project	C:\RTU32\Projects\60870_DEMO_150\Server104	STRATON project, max 15 chars.
4	IP address	192.168.0.1:502	RTU32 communication settings (for STRATON programming)
5	Main Program Type	IEC60870_104_Server	Select Client/Slave driver
6	Main Program Name	S104	IMPORTANT: Max 5 chars

Main Configuration parameters

XML file

This field defines the name and location of the XML file used by the Configurator. The XML file contains source code, functions, macros etc. used by the Configurator. The inexperienced user should here keep the default settings.

Straton / Brodersen WorkSuite project

This field defines the name of the RTU32 application program. You can freely define the name. If you are updating existing project, you must remember to keep the same name.
Note: Project name maximum length is 15 characters.

IP address

This field defines the IP address of the RTU32. It is used by the Configurator when downloading the application. The IP address must indicate port 502 (:502). If you just generate the application program and open Brodersen WorkSuite, you need to setup the IP address of the RTU32 target in the WorkSuite Tool-Communication parameters menu.

Main Program Type

Defines what type of driver you want to configure. Only one driver can be created by the specific Excel sheet. Select if you want to create a 101Master, 101Slave etc. driver.
When you have selected the wanted driver – only the specific driver work sheet has to be configured. E.g. if you select the 104 Server you should only configure the 104 Server work sheet. The other driver specific work sheet will be ignored.

NOTE: You can create multiple drivers or Redundancy Groups in the same RTU32 project. You just have to define each driver individually in its own Excel Workbook and assign unique Main Program Names for each driver.

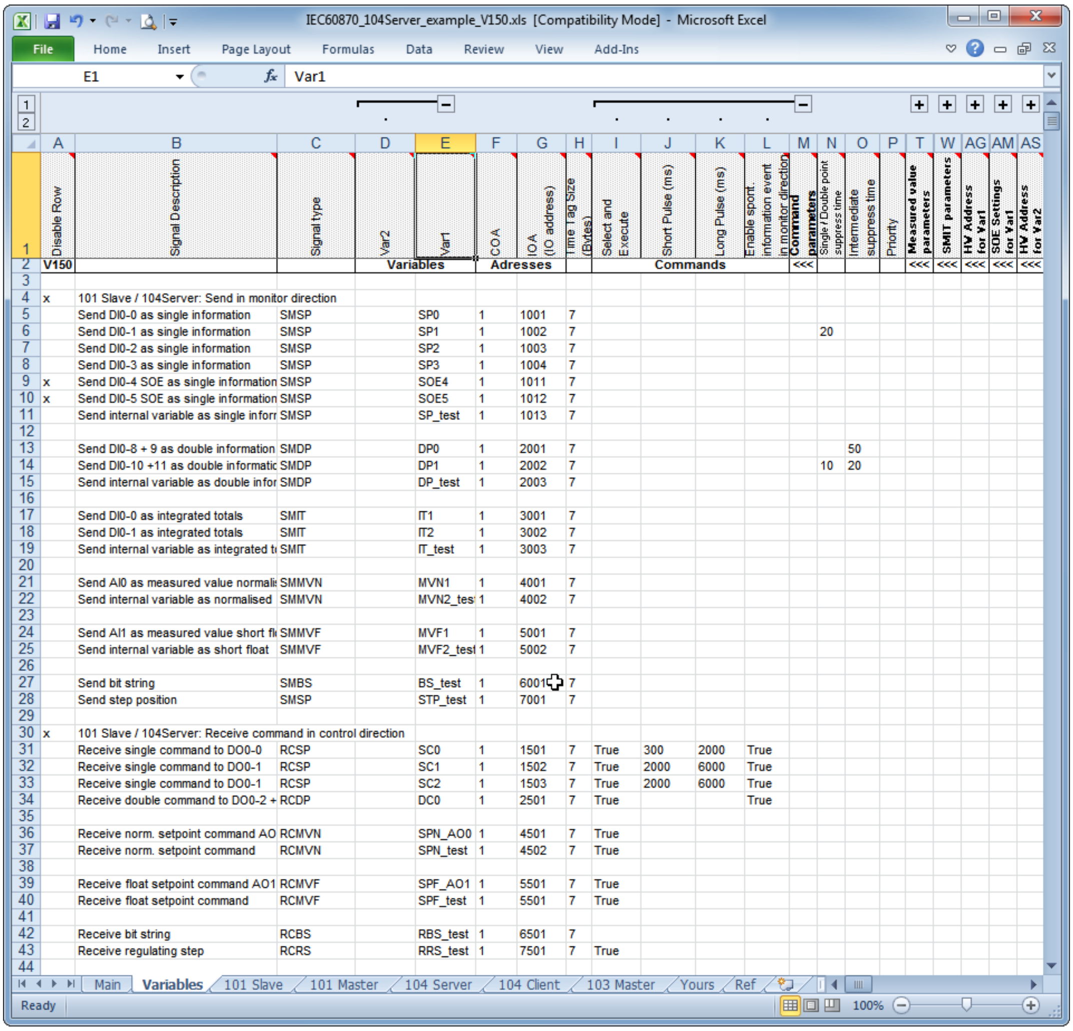
Main Program Name

Defines the name of the RTU32 driver program. Can freely be selected, but **ONLY** 5 characters

See comments in the Excel sheet for more details.

Work Sheet – Variables

The work sheets *Variables* after the *Main* sheet, define the specific driver data settings for variables, ASDU types, addresses etc:

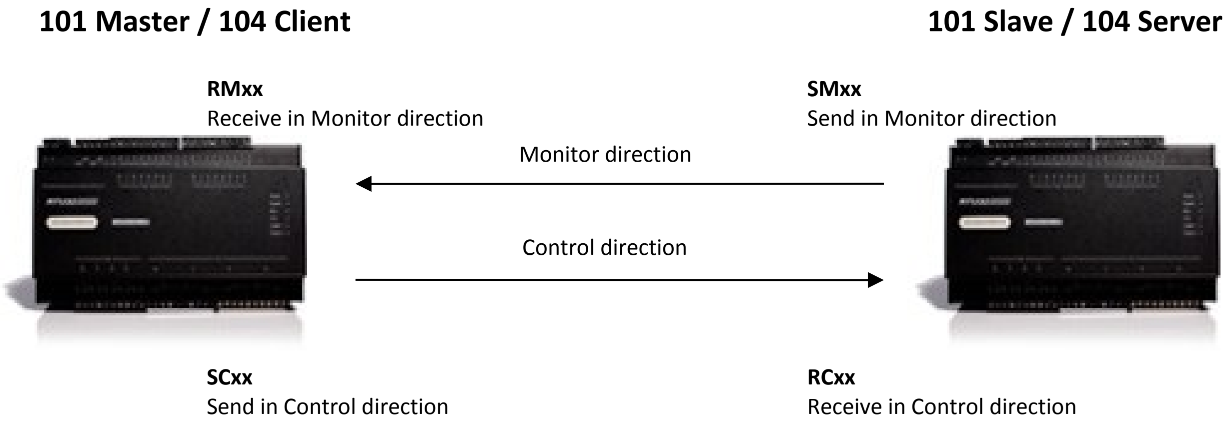


	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	T	W	AG	AM	AS
1	Disable Row	Signal Description	Signal type	Var2	Var1	COA	IOA (IO address)	Time Tag Size (Bytes)	Select and Execute	Short Pulse (ms)	Long Pulse (ms)	Enable spont. information event in monitor direction	Command parameters	Single / Double point suppress time	Intermediate suppress time	Priority	Measured value parameters	SMIT parameters	HV Address for Var1	SOE Settings for Var1	HV Address for Var2
2	V150			Variables		Adresses			Commands				<<<				<<<	<<<	<<<	<<<	<<<
3																					
4	x	101 Slave / 104Server: Send in monitor direction																			
5		Send DI0-0 as single information	SMSP		SP0	1	1001	7													
6		Send DI0-1 as single information	SMSP		SP1	1	1002	7						20							
7		Send DI0-2 as single information	SMSP		SP2	1	1003	7													
8		Send DI0-3 as single information	SMSP		SP3	1	1004	7													
9	x	Send DI0-4 SOE as single information	SMSP		SOE4	1	1011	7													
10	x	Send DI0-5 SOE as single information	SMSP		SOE5	1	1012	7													
11		Send internal variable as single inforr	SMSP		SP_test	1	1013	7													
12																					
13		Send DI0-8 + 9 as double information	SMDP		DP0	1	2001	7							50						
14		Send DI0-10 +11 as double informatic	SMDP		DP1	1	2002	7						10	20						
15		Send internal variable as double infor	SMDP		DP_test	1	2003	7													
16																					
17		Send DI0-0 as integrated totals	SMIT		IT1	1	3001	7													
18		Send DI0-1 as integrated totals	SMIT		IT2	1	3002	7													
19		Send internal variable as integrated t	SMIT		IT_test	1	3003	7													
20																					
21		Send AI0 as measured value normali	SMMVN		MVN1	1	4001	7													
22		Send internal variable as normalised	SMMVN		MVN2_tes	1	4002	7													
23																					
24		Send AI1 as measured value short fl	SMMVF		MVF1	1	5001	7													
25		Send internal variable as short float	SMMVF		MVF2_test	1	5002	7													
26																					
27		Send bit string	SMBS		BS_test	1	6001	7													
28		Send step position	SMSP		STP_test	1	7001	7													
29																					
30	x	101 Slave / 104Server: Receive command in control direction																			
31		Receive single command to DO0-0	RCSP		SC0	1	1501	7	True	300	2000	True									
32		Receive single command to DO0-1	RCSP		SC1	1	1502	7	True	2000	6000	True									
33		Receive single command to DO0-1	RCSP		SC2	1	1503	7	True	2000	6000	True									
34		Receive double command to DO0-2 +	RCDP		DC0	1	2501	7	True			True									
35																					
36		Receive norm. setpoint command AO	RCMVN		SPN_AO0	1	4501	7	True												
37		Receive norm. setpoint command	RCMVN		SPN_test	1	4502	7	True												
38																					
39		Receive float setpoint command AO1	RCMVF		SPF_AO1	1	5501	7	True												
40		Receive float setpoint command	RCMVF		SPF_test	1	5501	7	True												
41																					
42		Receive bit string	RCBS		RBS_test	1	6501	7													
43		Receive regulating step	RCRS		RRS_test	1	7501	7	True												
44																					

The variable work sheet defines the project variables and IEC60870 data types. You list your ASDU data types and link them together with variables and RTU32 physical I/O if required.

The Variable work sheet is divided into sections that you can open and close (open or hide columns). Each section has a specific configuration function e.g. configuration parameters for Measured Values or SOE settings.

The Signal Types or ASDU types are named according to their function. See the figure below that indicates how the ASDU type names and communication directions are supposed to be understood.


101 Master / 104 Client
101 Slave / 104 Server
RMxx
Receive in Monitor direction
SMxx
Send in Monitor direction
Monitor direction
Control direction
SCxx
Send in Control direction
RCxx
Receive in Control direction

In the work sheet named Ref is a reference from ASDU naming and IEC60870 Type Identities.
For each ASDU type you can define several options. E.g. for a Double Command you can define the variable name used in Brodersen WorkSuite, the IOA address, time tag size, select and execute options etc.

The Configurator support Measured Values type Normalized and Short Floating point. For these types please note that the following parameters have to be configured:

Normalized measured value:

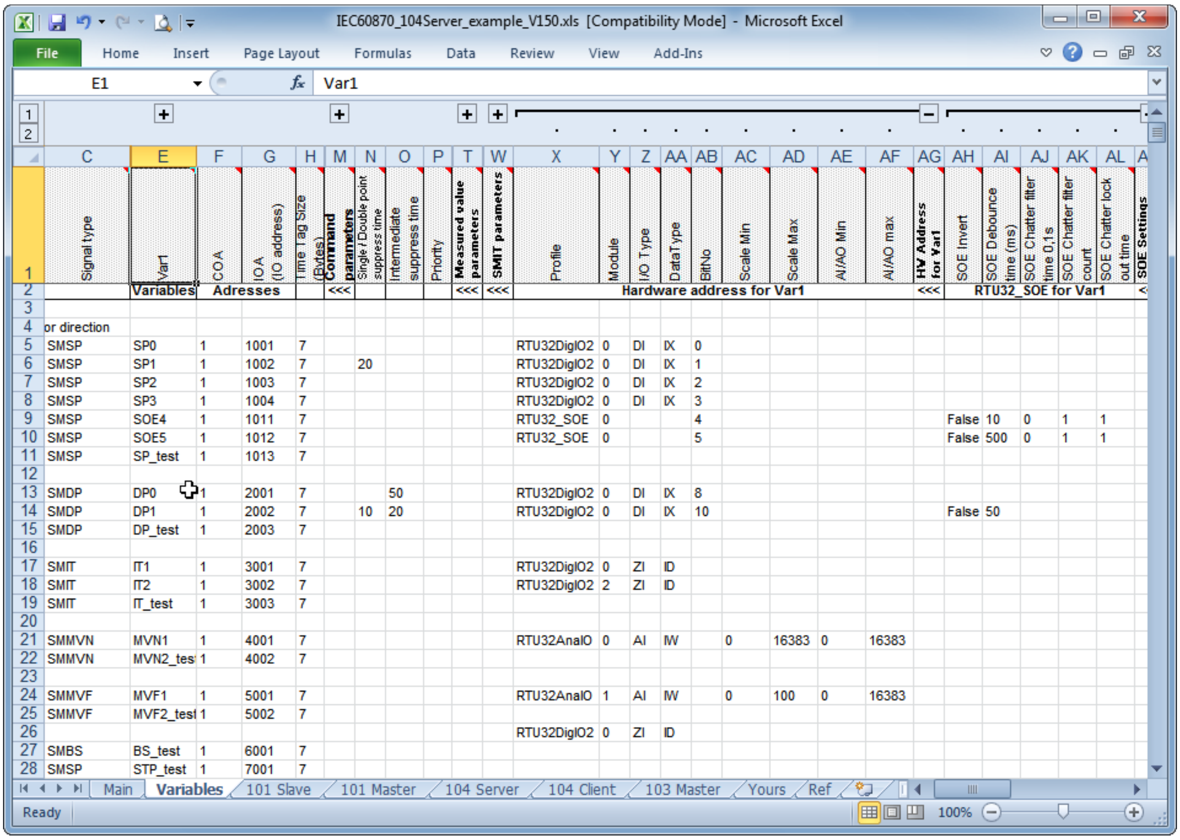
Analog resolution has to be setup (14 bit for RTU32 internal AI/AO and some I/O expansion. 12 bit for standard I/O Expansion).
For the physical I/O you have to define the full resolution in both ScaleMin / ScaleMax and AI/AOMin / AI/AOMax.
Threshold must be configured in %

Short Floating point measured value:

Analogue resolution shall NOT be configured.
For the physical I/O you have to define the wanted ScaleMin and ScaleMax. The full resolution in AI/AOMin / AI/AOMax has to be configured.
Threshold must be configured in actual value according to scaled range.

How to link Signal Types or ASDUs to physical I/O

The Configurator gives you the option to any link any ASDU to either direct online physical I/Os or for Single or Double Point information in Monitor Direction also to Sequence Of Event data I/O. The figure below shows you example of how this is configured.



	C	E	F	G	H	M	N	O	P	T	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	A
1	Signal type	Var1	COA	IOA (IO address)	Time Tag Size (Bytes)	Command parameters	Single / Double point suppress time	Intermediate suppress time	Priority	Measured value parameters	SMIT parameters	Profile	Module	I/O Type	DataType	BitNo	Scale Min	Scale Max	AI/AO Min	AI/AO max	HV Address for Var1	SOE Invert	SOE Debounce time (ms)	SOE Chatter filter time 0,1s	SOE Chatter filter count	SOE Chatter lock out time	SOE Settings
2		Variables	Adresses			<<<				<<<	<<<	Hardware address for Var1									<<<	RTU32_SOE for Var1					<
3																											
4	or direction																										
5	SMSP	SP0	1	1001	7							RTU32DigIO2	0	DI	IX	0											
6	SMSP	SP1	1	1002	7		20					RTU32DigIO2	0	DI	IX	1											
7	SMSP	SP2	1	1003	7							RTU32DigIO2	0	DI	IX	2											
8	SMSP	SP3	1	1004	7							RTU32DigIO2	0	DI	IX	3											
9	SMSP	SOE4	1	1011	7							RTU32_SOE	0			4						False	10	0	1	1	
10	SMSP	SOE5	1	1012	7							RTU32_SOE	0			5						False	500	0	1	1	
11	SMSP	SP_test	1	1013	7																						
12																											
13	SMDP	DP0	1	2001	7			50				RTU32DigIO2	0	DI	IX	8											
14	SMDP	DP1	1	2002	7		10	20				RTU32DigIO2	0	DI	IX	10						False	50				
15	SMDP	DP_test	1	2003	7																						
16																											
17	SMIT	IT1	1	3001	7							RTU32DigIO2	0	ZI	ID												
18	SMIT	IT2	1	3002	7							RTU32DigIO2	2	ZI	ID												
19	SMIT	IT_test	1	3003	7																						
20																											
21	SMMVN	MVN1	1	4001	7							RTU32AnalO	0	AI	IW		0	16383	0	16383							
22	SMMVN	MVN2_tes	1	4002	7																						
23																											
24	SMMVF	MVF1	1	5001	7							RTU32AnalO	1	AI	IW		0	100	0	16383							
25	SMMVF	MVF2_test	1	5002	7																						
26												RTU32DigIO2	0	ZI	ID												
27	SMBS	BS_test	1	6001	7																						
28	SMSP	STP_test	1	7001	7																						

All physical I/O and SOE data are linked using the Profile I/O Driver for RTU32 in Brodersen WorkSuite. The field that needs to be configured is defined in this example. For SOE (Sequence Of Event) and filtering options – see separate section in this manual.

Upper limit in number of ASDUs to use with the IEC60870 Configurator

The IEC60870 Configurator for RTU32 is not limited to any specific number of ASDUs, but we recommend that do not exceed the number that is listed in Appendix 3.

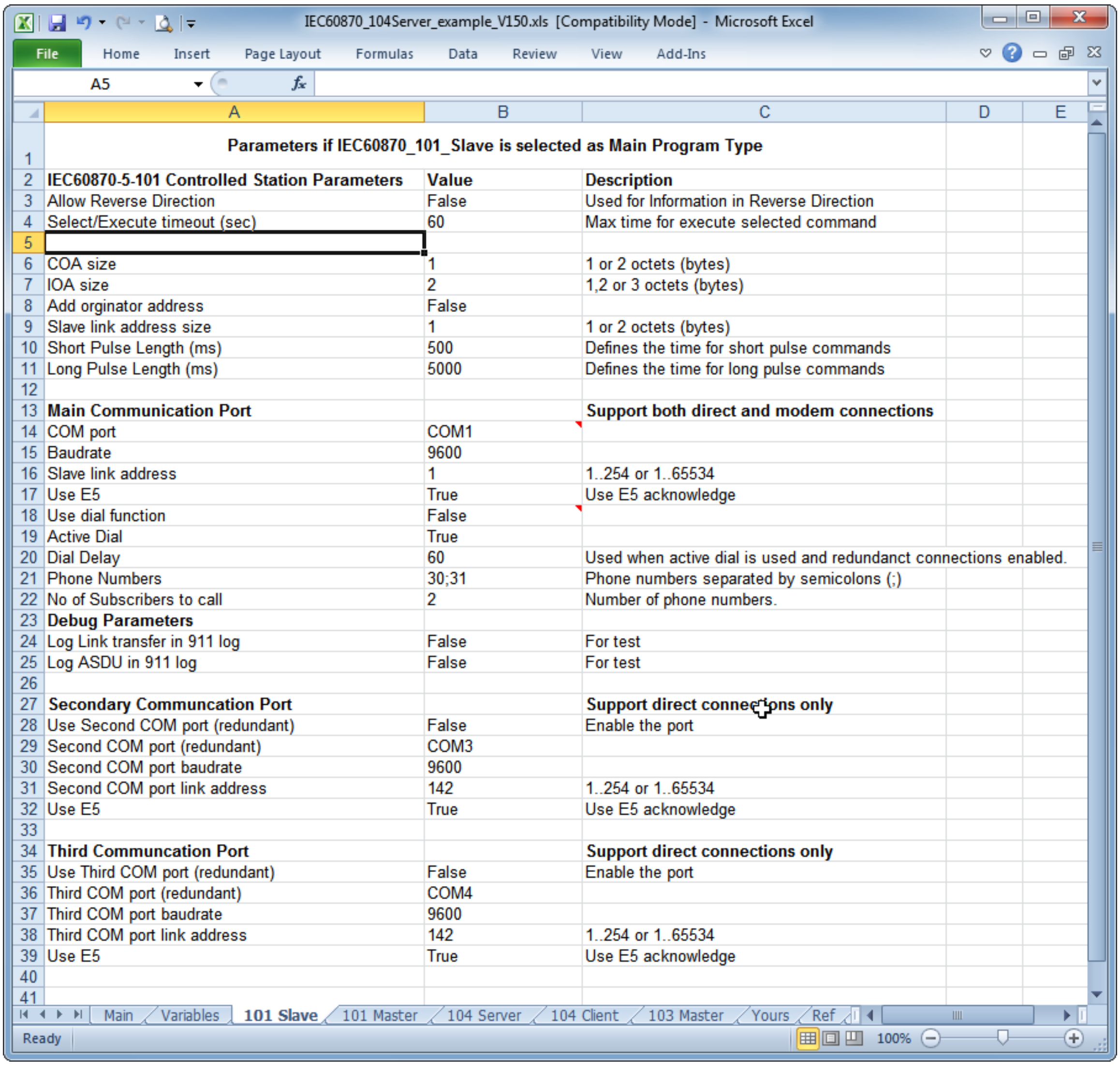
See comments in the Excel sheet for details

In the Excel sheet you will find comments and syntax descriptions for the different data – just put you cursor to the header to get details.

Work Sheet – 101 Slave

The work sheet 101 Slave includes the 101 Slave configuration parameters. Relevant if you have selected 101 Slave in the Main sheet.

Main Configuration parameters



	A	B	C
1	Parameters if IEC60870_101_Slave is selected as Main Program Type		
2	**IEC60870-5-101 Controlled Station Parameters**	**Value**	**Description**
3	Allow Reverse Direction	False	Used for Information in Reverse Direction
4	Select/Execute timeout (sec)	60	Max time for execute selected command
5			
6	COA size	1	1 or 2 octets (bytes)
7	IOA size	2	1,2 or 3 octets (bytes)
8	Add orginator address	False	
9	Slave link address size	1	1 or 2 octets (bytes)
10	Short Pulse Length (ms)	500	Defines the time for short pulse commands
11	Long Pulse Length (ms)	5000	Defines the time for long pulse commands
12			
13	**Main Communication Port**		**Support both direct and modem connections**
14	COM port	COM1	
15	Baudrate	9600	
16	Slave link address	1	1..254 or 1..65534
17	Use E5	True	Use E5 acknowledge
18	Use dial function	False	
19	Active Dial	True	
20	Dial Delay	60	Used when active dial is used and redundanct connections enabled.
21	Phone Numbers	30;31	Phone numbers separated by semicolons (;)
22	No of Subscribers to call	2	Number of phone numbers.
23	**Debug Parameters**		
24	Log Link transfer in 911 log	False	For test
25	Log ASDU in 911 log	False	For test
26			
27	**Secondary Communcation Port**		**Support direct connections only**
28	Use Second COM port (redundant)	False	Enable the port
29	Second COM port (redundant)	COM3	
30	Second COM port baudrate	9600	
31	Second COM port link address	142	1..254 or 1..65534
32	Use E5	True	Use E5 acknowledge
33			
34	**Third Communcation Port**		**Support direct connections only**
35	Use Third COM port (redundant)	False	Enable the port
36	Third COM port (redundant)	COM4	
37	Third COM port baudrate	9600	
38	Third COM port link address	142	1..254 or 1..65534
39	Use E5	True	Use E5 acknowledge
40			
41			

IEC60870-5-101 Controlled Station Parameters

Allow Reverse Direction
Enable the function of reading Information’s in reverse directions.

Select/Execute timeout (sec)
Time from an output is selected till it must be executed. In case of timeout time is elapsed the output must be selected again before executed.

COA size
Common Address of ASDU (COA) address size in bytes (octets).

IOA size
Information Object Address (IOA) address size in bytes (octets).

Add originator field
Use originator byte (octet) in 101 frames. Normally NOT used.

Slave link address size
Define the Link address size in bytes (octets).

Short Pulse Length (ms)
Short pulse length for Single and Double Commands in ms.

Long Pulse Length (ms)
Long pulse length for Single and Double Commands in ms.

Main Communication Port

COM port
Primary COM port used for 101 Slave driver. Note that the port cannot be used for any other driver in RTU32. If only one port is required for your 101 Slave driver – this is the one. If you use dial modem on the 101Slave connection, it should be selected here; COMx means direct Null Modem connection and MCOMx means that you use dial modem on the actual COM port.

Baudrate
Baudrate used on selected COM port.

Slave link address
Define the Link address of 101 Slave. Note that the address range depend on the size selected above.

Use E5
Enable using E5 acknowledge.

Use dial function
Enable dial modem function. Used when the 101 Slave connection is via a dial modem like PSTN or GSM.
The dial function includes modem management and passive/active dial functions. NOTE: The modem SHALL be configured to 8 data bit, Even parity and 1 stop bit.

Active Dial
Enable active dial out function. Enable the function of the RTU32 to dial the 101 Master Station in case of data in the queue. If not enabled the 101 Master is ALWAYS responsible for establishing connections to the RTU32.

Dial Delay
Define the delay for dial out to 101 Master in case of data in the queue. Only relevant if active dial out is enabled.

Phone Numbers
Define the phone number(s) of the host to dial. Up to 10 numbers can be defined – should be separated by a semi colon (;).

No of subscribers to call
Define the total number of phone numbers to call if the connection with the first dialed number fails (if busy, not connected etc.). If set to 2 as default, the RTU32 will try call the first and second number defined in the Phone Numbers cell.

Debug Parameters
Log Link transfer in 911 log
Enable to log Link communication in the RTU32 system log that can be review with Telnet on port 911. Should be disabled under normal operation.

Log ASDU in 911 log
Enable to log application data communication in the RTU32 system log that can be review with Telnet on port 911. Should be disabled under normal operation.

Secondary Communication Port

Use Second COM port (redundant)
Enable to use a second 101 Slave Link connection to the main Link driver data queue. The second link establishes a mirror of the data queue available via the main port. It is import that the host is managing the connections as if the data queue is emptied the data is lost and cannot be re-established.

Second COM port (redundant)
Second COM port used for 101 Slave driver. Note that the port cannot be used for any other driver in RTU32.

Second COM port baudrate
Baudrate used on selected COM port.

Second COM port link address
Define the Link address of second 101 Slave. Note that the address range depend on the size selected above.

Use E5
Enable using E5 acknowledge.

Third Communication Port

Use Third COM port (redundant)
Enable to use a second 101 Slave Link connection to the main Link driver data queue. The second link establishes a mirror of the data queue available via the main port. It is import that the host is managing the connections as if the data queue is emptied the data is lost and cannot be re-established.

Third COM port (redundant)
Second COM port used for 101 Slave driver. Note that the port cannot be used for any other driver in RTU32.

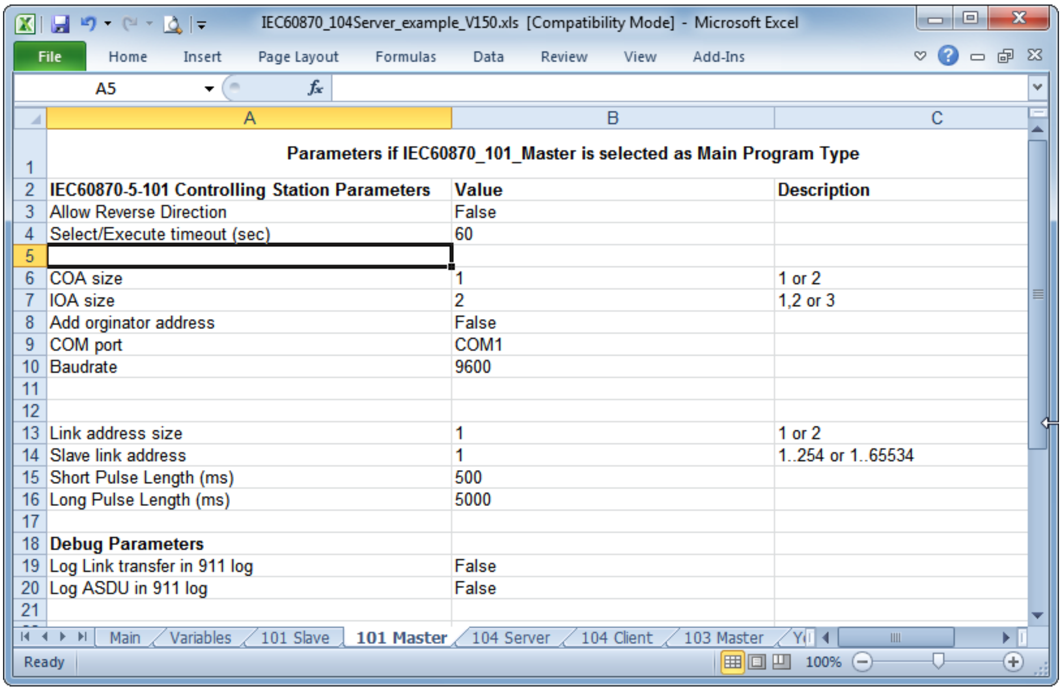
Third COM port baudrate
Baudrate used on selected COM port.

Third COM port link address
Define the Link address of second 101 Slave. Note that the address range depend on the size selected above.

Use E5
Enable using E5 acknowledge.

Work Sheet – 101 Master

The Configurator 101 Master is a limited version supporting only point-to-point – meaning only one Slave can be connected.
The work sheet:



	A	B	C
1	Parameters if IEC60870_101_Master is selected as Main Program Type		
2	**IEC60870-5-101 Controlling Station Parameters**	**Value**	**Description**
3	Allow Reverse Direction	False	
4	Select/Execute timeout (sec)	60	
5			
6	COA size	1	1 or 2
7	IOA size	2	1,2 or 3
8	Add orginator address	False	
9	COM port	COM1	
10	Baudrate	9600	
11			
12			
13	Link address size	1	1 or 2
14	Slave link address	1	1..254 or 1..65534
15	Short Pulse Length (ms)	500	
16	Long Pulse Length (ms)	5000	
17			
18	**Debug Parameters**		
19	Log Link transfer in 911 log	False	
20	Log ASDU in 911 log	False	
21			

Main Configuration parameter

IEC60870-5-101 Controlling Station Parameters

Allow Reverse Direction
When set to true sending data in Reverse Direction is enabled. NOTE: This is a special function.

Select/Execute timeout (sec)
Time from an output is selected till it must be executed. In case of timeout time is elapsed the output must be selected again before executed.

COA size
Common Address of ASDU (COA) address size in bytes (octets).

IOA size
Information Object Address (IOA) address size in bytes (octets).

Add originator field
Use originator byte (octet) in 101 frames. Normally NOT used.

COM port
COM port used for 101 Master. Note that the port cannot be used for any other driver in RTU32.

Baudrate
Baudrate used on selected COM port.

Master link address size
Define the Link address size in bytes used (octets).

Slave link address
Define the Link address of 101 Slave to connect. Note that the address range depend on the size selected above.

Short Pulse Length (ms)
Short pulse length for Single and Double Commands in ms.

Long Pulse Length (ms)
Long pulse length for Single and Double Commands in ms.

Debug Parameters

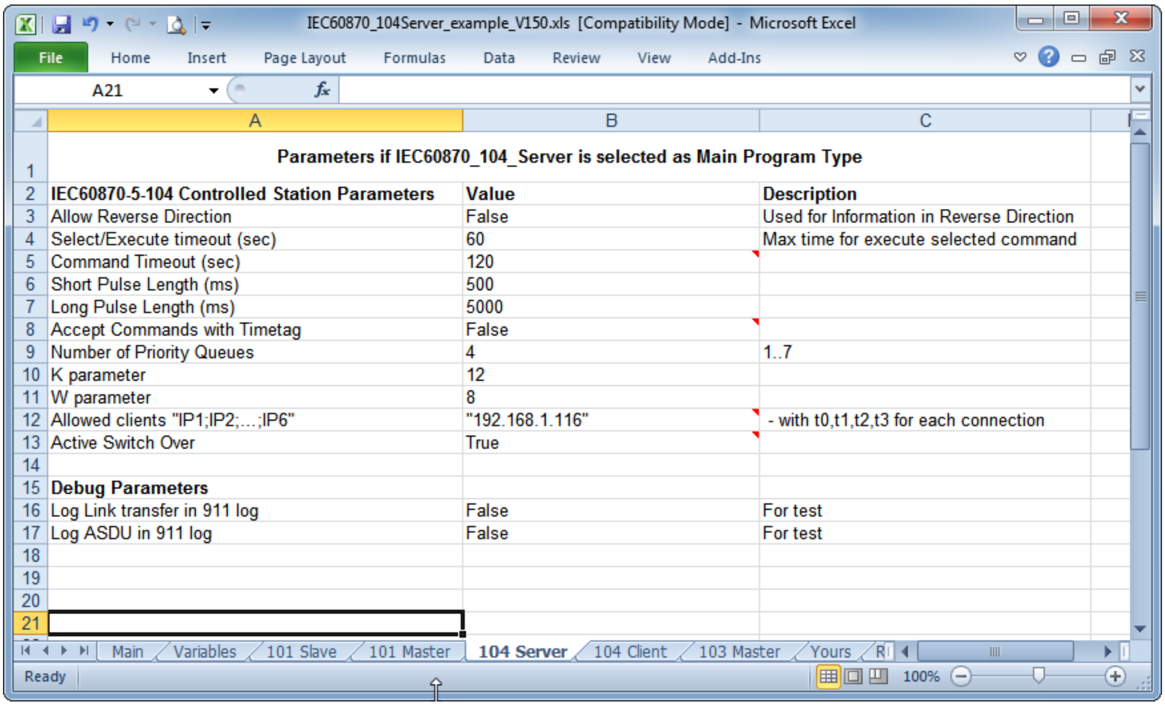
Log Link transfer in 911 log
Enable to log Link communication in the RTU32 system log that can be review with Telnet on port 911. Should be disabled under normal operation.

Log ASDU in 911 log
Enable to log application data communication in the RTU32 system log that can be review with Telnet on port 911. Should be disabled under normal operation.

Work Sheet – 104 Server

The 104 Server configurations support one single redundancy group. If you need more than one redundancy group, you must make an Excel sheet and make a code configuration for each redundancy group. Each redundancy group should be generated in the same WorkSuite program, but have different Main Program Name in the Main sheet.

The work sheet:



	A	B	C
1	Parameters if IEC60870_104_Server is selected as Main Program Type		
2	**IEC60870-5-104 Controlled Station Parameters**	**Value**	**Description**
3	Allow Reverse Direction	False	Used for Information in Reverse Direction
4	Select/Execute timeout (sec)	60	Max time for execute selected command
5	Command Timeout (sec)	120	
6	Short Pulse Length (ms)	500	
7	Long Pulse Length (ms)	5000	
8	Accept Commands with Timetag	False	
9	Number of Priority Queues	4	1..7
10	K parameter	12	
11	W parameter	8	
12	Allowed clients "IP1;IP2;…;IP6"	"192.168.1.116"	- with t0,t1,t2,t3 for each connection
13	Active Switch Over	True	
14			
15	**Debug Parameters**		
16	Log Link transfer in 911 log	False	For test
17	Log ASDU in 911 log	False	For test
18			
19			
20			
21			

Main Configuration parameters

IEC60870-5-104 Controlling Station Parameters

Allow Reverse Direction
When set to true sending data in Reverse Direction is enabled. NOTE: This is a special function.

Select/Execute timeout (sec)
Time from an output is selected till it must be executed. In case of timeout time is elapsed the output must be selected again before executed.

Command Timeout (sec)
Defines the time for a command to be invalid. If a command is newer or older than the defined Command timeout, the command is not executed.

Short Pulse Length (ms)
Short pulse length for single and double commands in ms.

Long Pulse Length (ms)
Long pulse length for single and double commands in ms.

Accept Commands with Timetag
Process information in control direction may be sent with or without a time tag, but must not be mixed.
Select between commands with time tags CON 58..64 or without time tag CON 45..51 Default is false.

Number of Priority Queues

Define the number of transmit queues for 104Server. Up to 7 can be defined. Data in a high priority queue will always be sent before data lower priority queues.

K parameter

The Maximum differences receive sequence number to send state variable APDUs. This according to the 104 protocol standard – default values is most likely to be used.

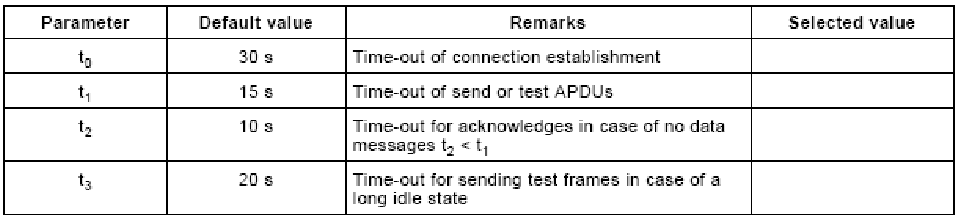
W parameter

Latest acknowledge after receiving W I format APDUs. This according to the 104 protocol standard – default values is most likely used.

Allowed clients "IP1;IP2;…;IP6"

Any 104 Client connecting to the RTU32 104 Server driver must be defined here. The IP address can be followed by specific timeout settings. The timeout are defined in table below. If no timeout is defined the default values will be used. Syntax of entering IP and timeouts – see the notes in the Excel sheet.

The t_0 to t_3 are defined in the 104 standard according to the description below:



Parameter	Default value	Remarks	Selected value
t_0	30 s	Time-out of connection establishment	
t_1	15 s	Time-out of send or test APDUs	
t_2	10 s	Time-out for acknowledges in case of no data messages $t_2 < t_1$	
t_3	20 s	Time-out for sending test frames in case of a long idle state	

Active Switch Over

Active switch over is used for enabling that the connection to a 104 Client automatically is switched over to a new client that is trying to get connections (using the STARTDT activate command). If Active switch is set to FALSE, no new Clients are allowed to get connection as long as one is already active.

Debug Parameters

Log Link transfer in 911 log

Enable to log Link communication in the RTU32 system log that can be review with Telnet on port 911. This parameter should be set to false during normal operation.

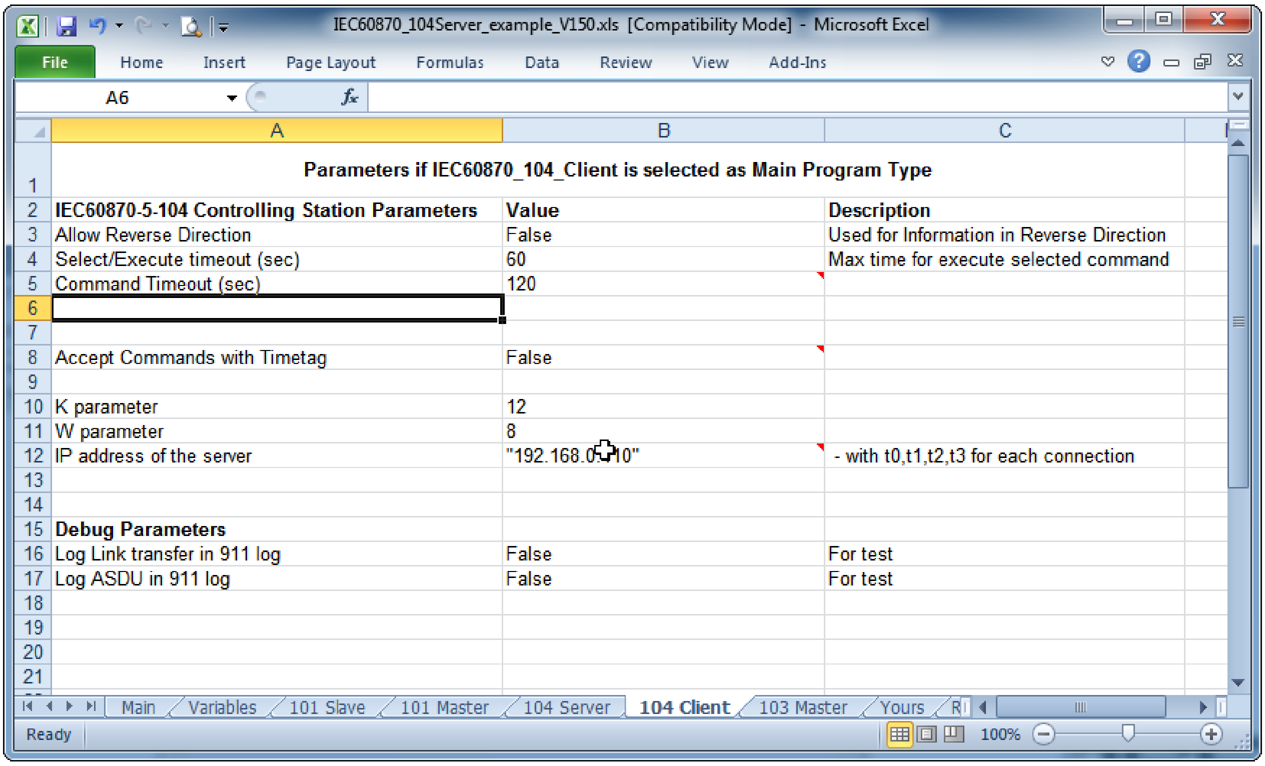
Log ASDU in 911 log

Enable to log application data communication in the RTU32 system log that can be review with Telnet on port 911. This parameter should be set to false during normal operation.

Work Sheet – 104 Client

The 104 Client driver is used for managing connections to 104 servers.

The work sheet:



	A	B	C
1	Parameters if IEC60870_104_Client is selected as Main Program Type		
2	**IEC60870-5-104 Controlling Station Parameters**	**Value**	**Description**
3	Allow Reverse Direction	False	Used for Information in Reverse Direction
4	Select/Execute timeout (sec)	60	Max time for execute selected command
5	Command Timeout (sec)	120	
6			
7			
8	Accept Commands with Timetag	False	
9			
10	K parameter	12	
11	W parameter	8	
12	IP address of the server	"192.168.0.10"	- with t0,t1,t2,t3 for each connection
13			
14			
15	**Debug Parameters**		
16	Log Link transfer in 911 log	False	For test
17	Log ASDU in 911 log	False	For test
18			
19			
20			
21			

Main Configuration parameters

IEC60870-5-104 Controlling Station Parameters

Allow Reverse Direction
When set to true sending data in Reverse Direction is enabled. NOTE: This is a special function.

Select/Execute Timeout (sec)
Time from an output is selected till it must be executed. In case of timeout time is elapsed the output must be selected again before executed.

Command Timeout (sec)
Defines the time for a command to be invalid. If a command is newer or older than the defined Command timeout, the command is not executed.

K parameter
The Maximum differences receive sequence number to send state variable APDUs. This according to the 104 protocol standard – default values is most likely used.

W parameter
Latest acknowledge after receiving W I format APDUs. This according to the 104 protocol standard – default values is most likely used.

IP address

IP address of the RTU32 – used by the Configurator to locale where to download the application. The IP address must indicate port 502 (:502). If you just generate the application program and open Brodersen WorkSuite, you need to setup the IP address of the RTU32 target in the WorkSuite Tool-Communication parameters menu.

Debug Parameter

Log Link transfer in 911 log

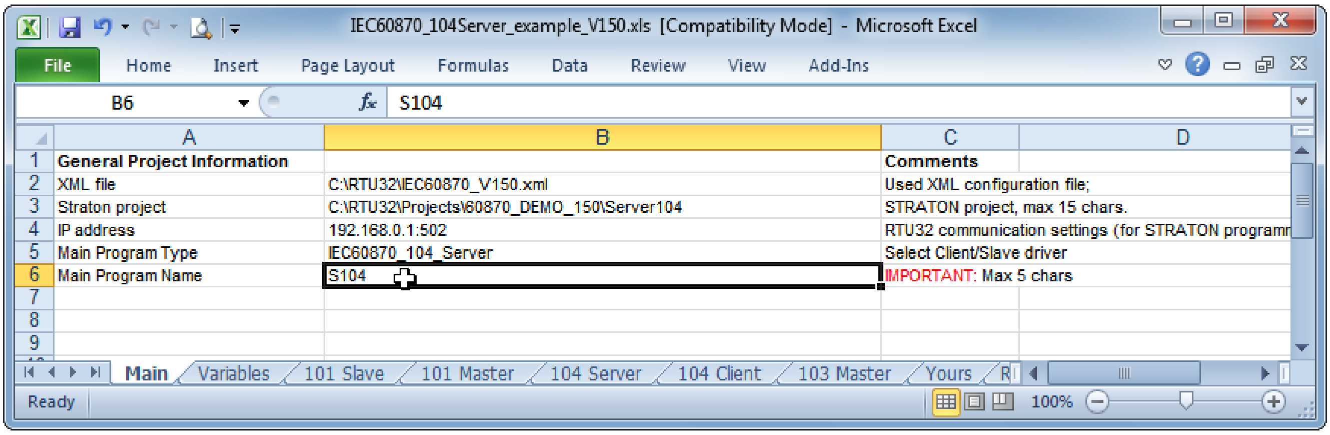
Enable to log Link communication in the RTU32 system log that can be review with Telnet on port 911. This parameter should be set to false during normal operation.

Log ASDU in 911 log

Enable to log application data communication in the RTU32 system log that can be review with Telnet on port 911. This parameter should be set to false during normal operation.

Multiple 104 Client Connections

Each Excel sheet configures a single connection to single 104 server. If you need to connect to more than one server then you will need to configure an Excel Worksheet for each connection. Each client work sheet must define same “Straton/WorkSuite project”. The “Main Program Name” must be different for each connection as this identifies the specific driver connection inside the generated program.



B6 S104

	A	B	C	D
1	General Project Information		Comments	
2	XML file	C:\RTU32\IEC60870_V150.xml	Used XML configuration file;	
3	Straton project	C:\RTU32\Projects\60870_DEMO_150\Server104	STRATON project, max 15 chars.	
4	IP address	192.168.0.1:502	RTU32 communication settings (for STRATON programm	
5	Main Program Type	IEC60870_104_Server	Select Client/Slave driver	
6	Main Program Name	S104	IMPORTANT: Max 5 chars	
7				
8				
9				

Work Sheet – 103 Master

The 103 Master driver comes in two flavors in this version:

1. IEC60870_103_Master

   This configuration is used to setup the 103 master driver to receive data from 103 slaves and enter them into Brodersen WorkSuite variables.

   Following signal types are supported RM3_T1, RM3_T2, RM3_T9 and SC3_GLC.
2. IEC60870_104_Server_103_Master

   As the first configuration this is used to setup the 103 master driver to receive data from 103 slaves and enter them into Brodersen WorkSuite variables.

   Furthermore this configuration enables a 104 server which can take the values received from the 103 slaves and forward them automatically to the 104 protocol. You will need to adjust both the 104 Server and 103 Master sheets.

   Following signal types are supported SMMVF, SMIT, RM3_T1, RM3_T2, RM3_T9 and SC3_GLC.

Signal types supported:

SMMVF This can take measurands II, TypID 9 data on the 103 protocol and forward them as floating point to the 104 connection.

SMIT This can take counters TypID 205 on the 103 protocol and forward the as integrated total to the 104 connection. TypID 205 is Siemens P50/55 private definition data.

RM3_T1 Is used to receive TypID 1 data (time-tagged messages) from 103 slaves and put it into Brodersen WorkSuite variables.

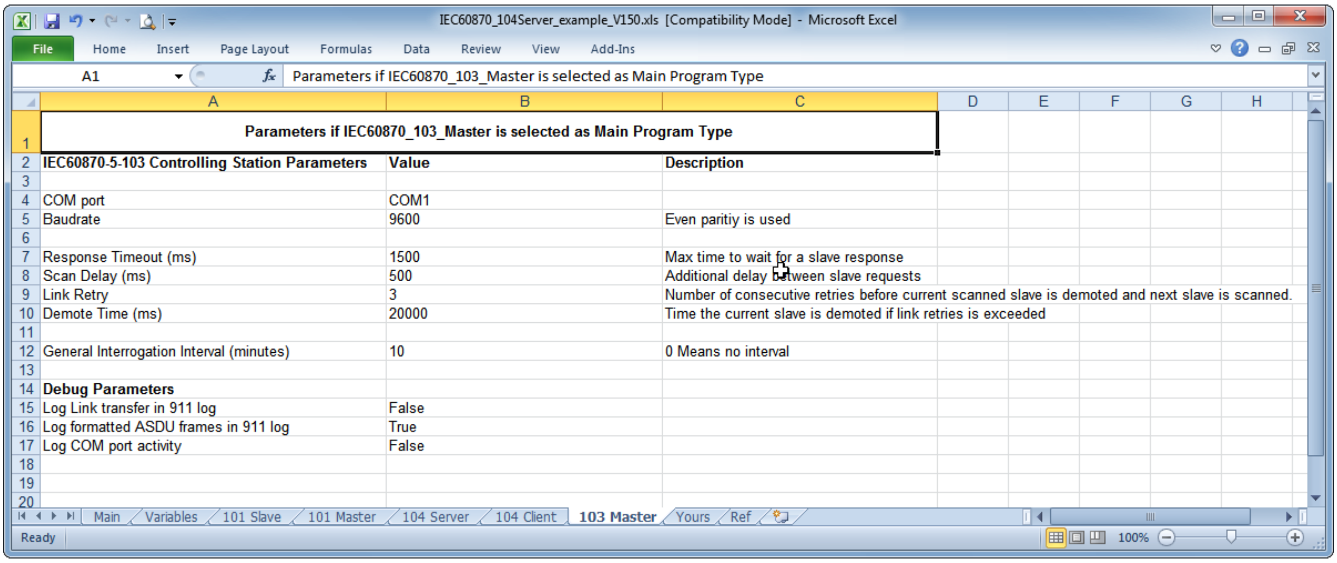
RM3_T2 Is used to receive TypID 2 data (time-tagged messages with relative time) from 103 slaves and put into Brodersen WorkSuite variables.

RM3_T9 Is used to receive TypID 9 data (measurands II) from 103 slaves and put it into Brodersen WorkSuite variables.

SC3_GLC Is used to send TypID 20 command (general command) to 103 slaves.

The 103 Master supports multi drop connections. When using multi drop connections tests has shown that external converters RS232/RS485 used on RTU32 RS232 COM ports gives the best performance.

The work sheet:



Parameters if IEC60870_103_Master is selected as Main Program Type

IEC60870-5-103 Controlling Station Parameters	Value	Description
COM port	COM1	
Baudrate	9600	Even paritiy is used
Response Timeout (ms)	1500	Max time to wait for a slave response
Scan Delay (ms)	500	Additional delay between slave requests
Link Retry	3	Number of consecutive retries before current scanned slave is demoted and next slave is scanned.
Demote Time (ms)	20000	Time the current slave is demoted if link retries is exceeded
General Interrogation Interval (minutes)	10	0 Means no interval
Debug Parameters		
Log Link transfer in 911 log	False	
Log formatted ASDU frames in 911 log	True	
Log COM port activity	False	

Main Configuration parameters

IEC60870-5-103 Controlling Station Parameters

COM port
The COM port used for 103 Master. Note that the port cannot be used for any other driver in Brodersen WorkSuite.

Baudrate
Baudrate used on selected COM port.

Response Timeout (ms)
Define the maximum time the master should wait for the Slave to answer before it sends a retry.

Scan Delay (ms)
As the RTU32 is a powerful communicator the Master will send a new Slave request a few msec after getting a reply. Some 103 Slaves do not manage to handle this speed, so it is recommended to define a scan delay to prevent communication failures. Depending on the actual configuration it is recommended that you set the Scan delay to e.g. 50msec. You can then try to decrease the delay until you get communication failures but remember to leave a good margin to prevent occasional retries.

Link Retries
Number of retires before the Slave is demoted. See under Demote time.

Demote time
After a Slave has been requested for data the number of retires defined above, the Slave will be demoted for a period of time (Demote Time). It means that the Slave is not requested for data for this period. The Demote function is used for keeping the

data flow running in a multi drop application even if one or more Slaves are not replying. If the demote function is not used, a single Slave that is not responding will slow down the complete communication as the driver would constantly be waiting for answers for the missing Slave

Debug Parameters

Log Link transfer in 911 log

Enable to log Link communication in the RTU32 system log that can be review with the Event Viewer or Telnet on port 911. The log is made by the 103 link layer function. This parameter should be set to false during normal operation.

Log formatted ASDU frames in 911 log

Enable to log application data communication in the RTU32 system log that can be review with the Event Viewer or Telnet on port 911. This parameter should be set to false during normal operation. Below is shown a log where the master is sending command to slave address 101:

```
05/16 14:40:44.709  COM1: >>68,0A,0A,68,73,65,14,81,14,00,FF,20,02,01,A3,16
05/16 14:40:44.709  COM1:   >>SLV(101) Typ=20 General command
05/16 14:40:44.709  COM1:     COT=20  general command
05/16 14:40:44.709  COM1:     COA=0 FUN=255 INF=32
05/16 14:40:44.709  COM1:     Val=2 = ON
05/16 14:40:44.709  COM1:     RII=1
05/16 14:40:44.766  COM1: <<=R101=ACK: User data accepted
05/16 14:40:45.362  COM1: <<68,0E,0E,68,28,65,01,81,14,00,FF,20,02,23,A1,36,11,01,50,16
05/16 14:40:45.362  COM1:   <<SLV(101) Typ=1 Time-tagged message
05/16 14:40:45.362  COM1:     COT=20  positive acknowledgement of command
05/16 14:40:45.362  COM1:     COA=0 FUN=255 INF=32
05/16 14:40:45.362  COM1:     Val=2 = ON
05/16 14:40:45.362  COM1:     SIN=1
05/16 14:40:45.962  COM1: <<68,0E,0E,68,08,65,01,81,0C,00,FF,20,02,23,A1,36,11,01,28,16
05/16 14:40:45.962  COM1:   <<SLV(101) Typ=1 Time-tagged message
05/16 14:40:45.962  COM1:     COT=12  remote operation
05/16 14:40:45.962  COM1:     COA=0 FUN=255 INF=32
05/16 14:40:45.962  COM1:     Val=2 = ON
05/16 14:40:45.962  COM1:     SIN=1
```

Log COM port activity

Enable to log comport activity into the RTU32 system log that can be review with the Event Viewer or Telnet on port 911. Here the log is made by the low level serial driver. This parameter should be set to false during normal operation.

5. The IEC60870 Code Generator

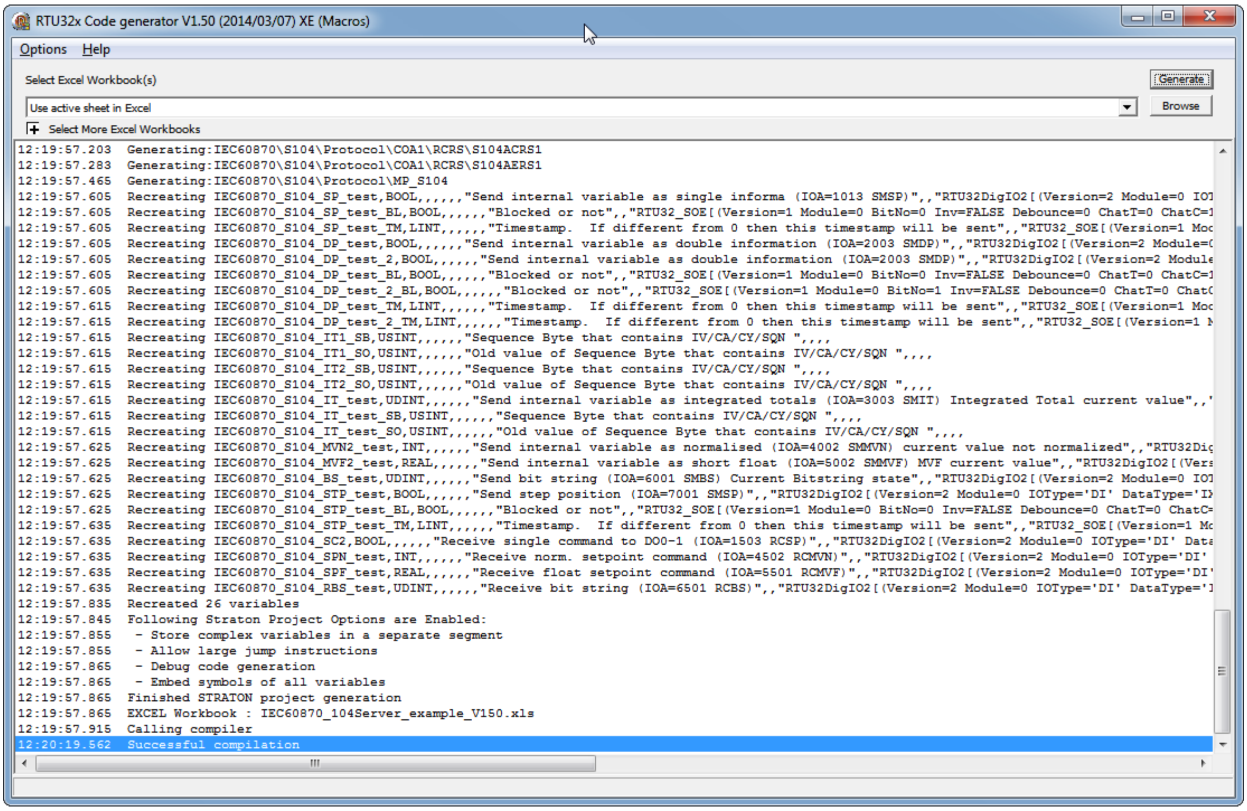
General

The IEC60870 Code Generator is the program that takes all the information entered into the Excel sheet and generates the Brodersen WorkSuite application project/program code. It will also compile the project and if selected, download it into the RTU32 and start it. If the Brodersen WorkSuite project does already exist, it will just update the existing one. And it will not compromise any other programs already implemented in the Brodersen WorkSuite project. If the Brodersen WorkSuite project does not exists, the code generator will create a new project with the name defined in the Excel sheet.

User Interface

A normal user should only use the Generate bottom to activate the code generation.

The system event window gives the user details of the code generating process. Simple errors like wrong file names etc. will be reported here.


RTU32x Code generator V1.50 (2014/03/07) XE (Macros)
Options
Help
Generate
Browse
Select More Excel Workbooks

Select Excel Workbook(s)
Use active sheet in Excel

```
12:19:57.203  Generating:IEC60870\S104\Protocol\COA1\RCRS\S104ACRS1
12:19:57.283  Generating:IEC60870\S104\Protocol\COA1\RCRS\S104AERS1
12:19:57.465  Generating:IEC60870\S104\Protocol\MP_S104
12:19:57.605  Recreating IEC60870_S104_SP_test,BOOL,,,,,,"Send internal variable as single informa (IOA=1013 SMSP)",,"RTU32DigIO2[(Version=2 Module=0 IOT
12:19:57.605  Recreating IEC60870_S104_SP_test_BL,BOOL,,,,,,"Blocked or not",,"RTU32_SOE[(Version=1 Module=0 BitNo=0 Inv=FALSE Debounce=0 ChatT=0 ChatC=1
12:19:57.605  Recreating IEC60870_S104_SP_test_TM,LINT,,,,,,"Timestamp.  If different from 0 then this timestamp will be sent",,"RTU32_SOE[(Version=1 Moc
12:19:57.605  Recreating IEC60870_S104_DP_test,BOOL,,,,,,"Send internal variable as double information (IOA=2003 SMDP)",,"RTU32DigIO2[(Version=2 Module=(
12:19:57.605  Recreating IEC60870_S104_DP_test_2,BOOL,,,,,,"Send internal variable as double information (IOA=2003 SMDP)",,"RTU32DigIO2[(Version=2 Module
12:19:57.605  Recreating IEC60870_S104_DP_test_BL,BOOL,,,,,,"Blocked or not",,"RTU32_SOE[(Version=1 Module=0 BitNo=0 Inv=FALSE Debounce=0 ChatT=0 ChatC=1
12:19:57.605  Recreating IEC60870_S104_DP_test_2_BL,BOOL,,,,,,"Blocked or not",,"RTU32_SOE[(Version=1 Module=0 BitNo=1 Inv=FALSE Debounce=0 ChatT=0 Chat(
12:19:57.615  Recreating IEC60870_S104_DP_test_TM,LINT,,,,,,"Timestamp.  If different from 0 then this timestamp will be sent",,"RTU32_SOE[(Version=1 Moc
12:19:57.615  Recreating IEC60870_S104_DP_test_2_TM,LINT,,,,,,"Timestamp.  If different from 0 then this timestamp will be sent",,"RTU32_SOE[(Version=1 M
12:19:57.615  Recreating IEC60870_S104_IT1_SB,USINT,,,,,,"Sequence Byte that contains IV/CA/CY/SQN ",,,,
12:19:57.615  Recreating IEC60870_S104_IT1_SO,USINT,,,,,,"Old value of Sequence Byte that contains IV/CA/CY/SQN ",,,,
12:19:57.615  Recreating IEC60870_S104_IT2_SB,USINT,,,,,,"Sequence Byte that contains IV/CA/CY/SQN ",,,,
12:19:57.615  Recreating IEC60870_S104_IT2_SO,USINT,,,,,,"Old value of Sequence Byte that contains IV/CA/CY/SQN ",,,,
12:19:57.615  Recreating IEC60870_S104_IT_test,UDINT,,,,,,"Send internal variable as integrated totals (IOA=3003 SMIT) Integrated Total current value",,'
12:19:57.615  Recreating IEC60870_S104_IT_test_SB,USINT,,,,,,"Sequence Byte that contains IV/CA/CY/SQN ",,,,
12:19:57.615  Recreating IEC60870_S104_IT_test_SO,USINT,,,,,,"Old value of Sequence Byte that contains IV/CA/CY/SQN ",,,,
12:19:57.625  Recreating IEC60870_S104_MVN2_test,INT,,,,,,"Send internal variable as normalised (IOA=4002 SMMVN) current value not normalized",,"RTU32Dig
12:19:57.625  Recreating IEC60870_S104_MVF2_test,REAL,,,,,,"Send internal variable as short float (IOA=5002 SMMVF) MVF current value",,"RTU32DigIO2[(Vers
12:19:57.625  Recreating IEC60870_S104_BS_test,UDINT,,,,,,"Send bit string (IOA=6001 SMBS) Current Bitstring state",,"RTU32DigIO2[(Version=2 Module=0 IOT
12:19:57.625  Recreating IEC60870_S104_STP_test,BOOL,,,,,,"Send step position (IOA=7001 SMSP)",,"RTU32DigIO2[(Version=2 Module=0 IOType='DI' DataType='I
12:19:57.625  Recreating IEC60870_S104_STP_test_BL,BOOL,,,,,,"Blocked or not",,"RTU32_SOE[(Version=1 Module=0 BitNo=0 Inv=FALSE Debounce=0 ChatT=0 ChatC=
12:19:57.635  Recreating IEC60870_S104_STP_test_TM,LINT,,,,,,"Timestamp.  If different from 0 then this timestamp will be sent",,"RTU32_SOE[(Version=1 Mc
12:19:57.635  Recreating IEC60870_S104_SC2,BOOL,,,,,,"Receive single command to DO0-1 (IOA=1503 RCSP)",,"RTU32DigIO2[(Version=2 Module=0 IOType='DI' Data
12:19:57.635  Recreating IEC60870_S104_SPN_test,INT,,,,,,"Receive norm. setpoint command (IOA=4502 RCMVN)",,"RTU32DigIO2[(Version=2 Module=0 IOType='DI'
12:19:57.635  Recreating IEC60870_S104_SPF_test,REAL,,,,,,"Receive float setpoint command (IOA=5501 RCMVF)",,"RTU32DigIO2[(Version=2 Module=0 IOType='DI'
12:19:57.635  Recreating IEC60870_S104_RBS_test,UDINT,,,,,,"Receive bit string (IOA=6501 RCBS)",,"RTU32DigIO2[(Version=2 Module=0 IOType='DI' DataType='I
12:19:57.835  Recreated 26 variables
12:19:57.845  Following Straton Project Options are Enabled:
12:19:57.855   - Store complex variables in a separate segment
12:19:57.855   - Allow large jump instructions
12:19:57.865   - Debug code generation
12:19:57.865   - Embed symbols of all variables
12:19:57.865  Finished STRATON project generation
12:19:57.865  EXCEL Workbook : IEC60870_104Server_example_V150.xls
12:19:57.915  Calling compiler
12:20:19.562  Successful compilation
```

Code Generation Details

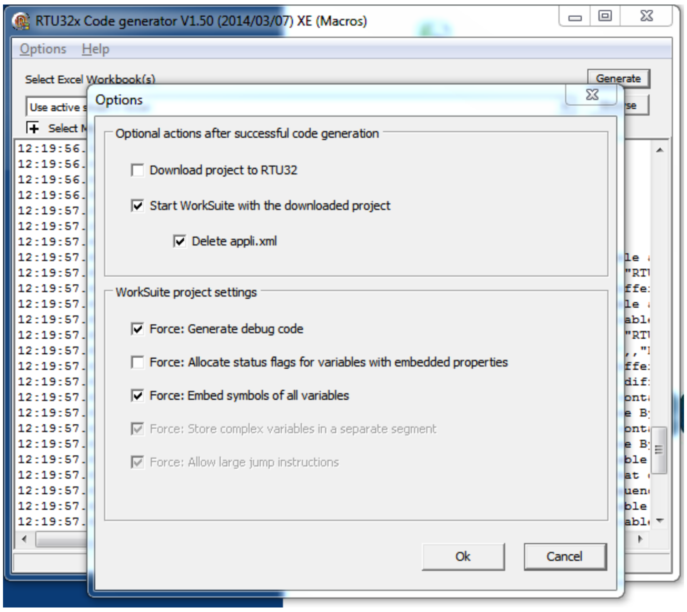
The code generator is using all the data and settings defined in the Excel sheet to generate a Brodersen WorkSuite application program. All settings and references for the code generation that is not defined in the Excel sheet are defined in the XML database file. See XML database file section for details.

When you activate the code generation it will search for the Brodersen WorkSuite project defined in the Excel sheet. If there is a Brodersen WorkSuite project, it will open it and **ADD** the IEC60870 driver code to this project in Structured Text (ST).

If there is no existing Brodersen WorkSuite project, the generator will create a new project according to the name defined in the Excel sheet.

Configuration Options Window

The Code generator has option menu that is used to configure what should be done during and after protocol code generation.



Options

Optional actions after successful code generation

☐ Download project to RTU32

☑ Start WorkSuite with the downloaded project

☑ Delete appli.xml

WorkSuite project settings

☑ Force: Generate debug code

☐ Force: Allocate status flags for variables with embedded properties

☑ Force: Embed symbols of all variables

☑ Force: Store complex variables in a separate segment

☑ Force: Allow large jump instructions

```
12:19:56
12:19:56
12:19:56
12:19:56
12:19:57
12:19:57
12:19:57
12:19:57
12:19:57
12:19:57
12:19:57
12:19:57
12:19:57
12:19:57
12:19:57
12:19:57
12:19:57
12:19:57
12:19:57
12:19:57
12:19:57
12:19:57
12:19:57
12:19:57
12:19:57
```

Optional actions after successful code generation

- Download project to RTU32
  - When this option is enable, the generated Brodersen WorkSuite project will be downloaded to the RTU using the IP address that is specified on the main sheet of the Excel workbook.
- Start Brodersen WorkSuite with the downloaded project
  - Here you select if the Brodersen WorkSuite should be started.
  - WorkSuite uses the file ‘appli.xml’ to remember which programs and windows were open when the project was last edited.   WorkSuite may in some situations not show the protocol files created by the generator in correct program folders. If you experience this then deleting the appli.xml will ensure that the all folders and program files are displayed correctly.

Brodersen WorkSuite project settings

- Force: Generate debug code

  This project settings option embeds some debugging code to allow online break points.
- Force: Embed symbols of all variables

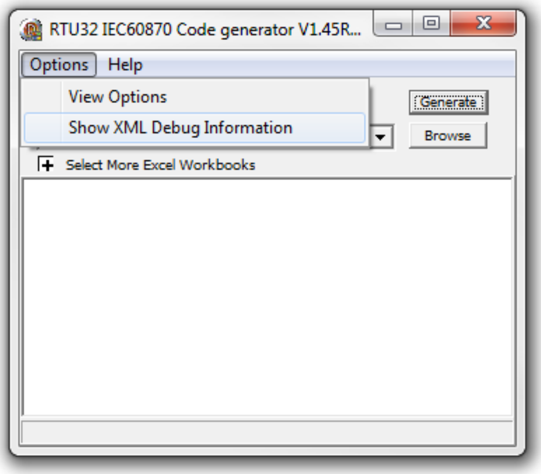
  This project settings option instructs Brodersen WorkSuite to embed and download the names of all variable to the RTU.
- Force: Store complex variables in a separate segment

  This project setting option is a mandatory option and cannot be disabled.
- Force: Allow large jump instructions.

  This project setting option is a mandatory option and cannot be disabled.

The Brodersen WorkSuite project settings are the same as those that also can be altered from within the Brodersen WorkSuite.

XML Debug Information


RTU32 IEC60870 Code generator V1.45R...
Options
Help
View Options
Show XML Debug Information
Generate
Browse
Select More Excel Workbooks

The “Show XML Debug Information” options is used to show the Code Generator extended XML debug facility, which is default disabled. The debug facility provides useful information for the experienced user when the XML file is edited.

6. Filter functions on SOE and Double Point Information

General

RTU32 provide functions to handle firmware buffed events with accurate time stamps and filtering. The Sequence Of Event (SOE) I/O Driver in RTU32 is handling this. All settings for enabling and configure the filters and events are done in the Excel Variable sheet as described under the section Work Sheet – Variables.

A Double Point intermediate suppression filter in the IEC60870 Driver is available when using the IEC60870 Configurator. The filters are used for suppressing both Intermediate and indeterminate (faulty) state reporting from slow double point data processes.
NOTE: At the time of this v1.50 release only RTU32/RTU32R supports SOE. RTU32E and RTU32S do not support SOE.

Filter functions for SOE

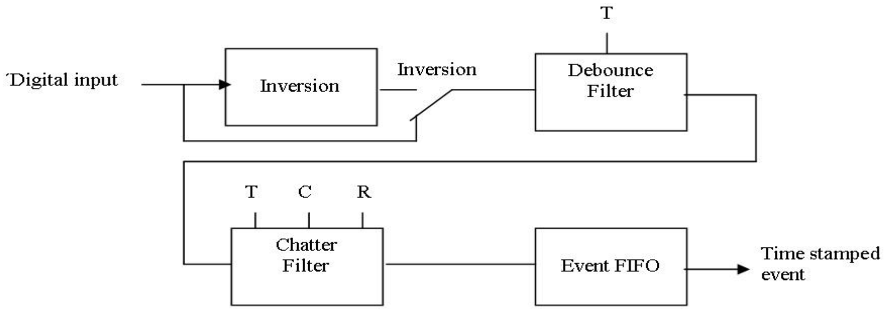
In general the filter functions for SOE data include/provide;

- Buffering of digital input events in the firmware independent of the PLC runtime application cycle – and that means time stamping accuracy of 2-3ms for internal DI and better than 10ms on Expansion I/O DI.
- Available for the first 200 digital inputs on a RTU32 node which include internal DIs and Expansion module DI (that mean e.g. 200 Single Point Information or 100 double point information)..
- Support debounce and chattering filter functions with properties adjustable in the Variable Sheet.
- Blocking of DPI data when chatter filter is active.

The firmware buffered events with time stamp are moved to the IEC60870 Driver transmission queue according to the selected driver and the other configuration settings.

The SOE filter is a multi-stage processing filter that handles the physical value before presenting it as a SOE value. The processing of the value is individually adjustable for each SOE.

Filter block diagram:



The multistage include the following functions/level before the data sets are sent to the event FIFO:

- Inversion of input value. Used for inverting the input value transmitted to the next level.
- Debounce filter. Debounce filter can be used on the first 200 inputs and prevents the processing of fast state changes of the inputs, like for example, those caused by contact bouncing. Signal changes are ignored depending on the preset time.
- Chatter filter. Chatter filter can be used for the first 200 inputs. It limits the number of events to a configurable value during a configurable time period. This should prevent multiple event registrations for the same input, e.g. disturbance influences due to slowly changing inputs (because the hysteresis is possibly set to small).

Debounce filter function for SOE

An adjustable debounce filter is provided for each digital input. On detection of an input change, a timer with the filter time ‘T’ is started for the changed input, and forwarding of the information is suppressed. If the input changes back to the original state, before the timer has expired, the timer is cancelled. If the input doesn’t changes back, and the timer expire, the new input state is forwarded. The time stamp forwarded is the time when the actual input change was detected.
The value range for the filter time is 0 to 32767 millsec, the value 0 deactivates the debounce filter. The figure shows an example of a debounce filter in action:

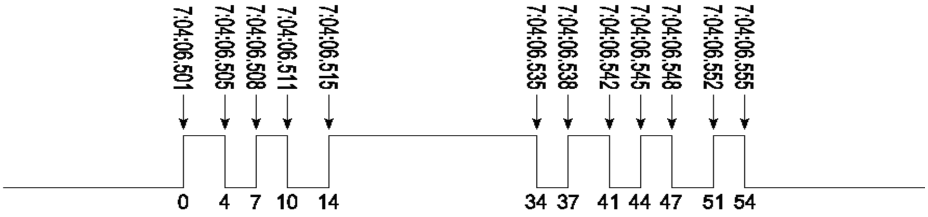


Figure shows an example of how SOE events are handled by the filter at T=10ms:

- Stable ON Event at 7:04:06.515 is forwarded at 7:04:06.525, with time stamp 7:04:06.515
- Stable OFF Event at 7:04:06.555 is forwarded at 7:04:06.565, with time stamp 7:04:06.555
- If filter time T=25 milliseconds, no event is recorded with these settings.

Chatter filter function for SOE

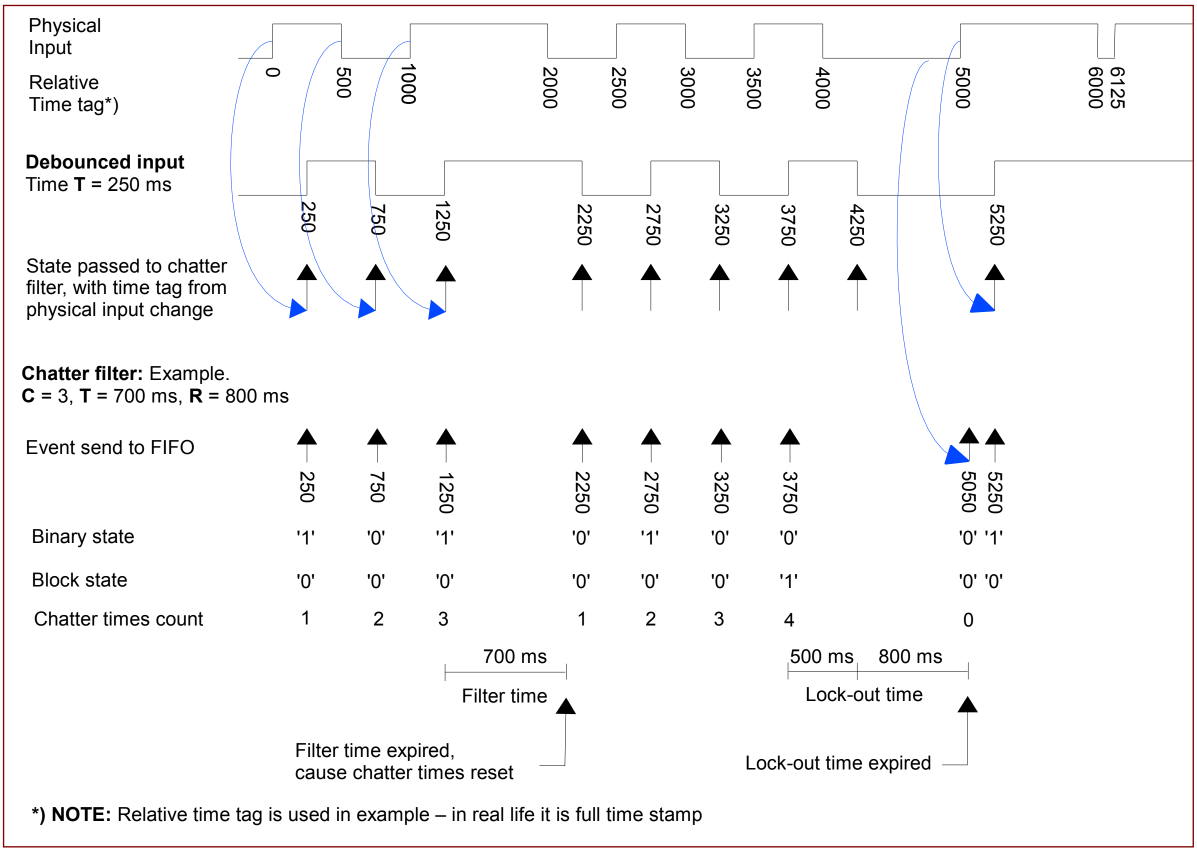
An adjustable chatter filter is provided for each digital input.
A digital input is disabled if the number of state changes, encountered during a defined time interval, is excessively high. While the chatter filter is ON, all state transitions are ignored. While it is OFF, state transitions are gated through without further delay. Events are reported whenever the chatter filter state changes from OFF to ON, or from ON to OFF. Three parameters are used to adjust the filter, and is individual adjustable for each input:

Filter time [T]:
The filter time ‘T’ define the minimum time a state must be stable in order not to increment the Chatter times counter, when state change occur. However every time a state is stable longer than the filter time, the Chatter times counter is reset.
The filter time is configurable from 0.1 to 6553.5 sec, in units of 100 ms. If set to 0 the chatter filter is disabled.

Chatter times (changes) [C]:
The maximum allowable numbers of consecutive state transitions ‘C’, that can occur with state width less than the filter time, without turning the chatter filter ON. If the number is exceeded, the chatter filter will turn ON, and any further transitions will be ignored for the duration of the “lock-out” period.
The maximum number of state transitions is configurable from 1 to 255 changes.

Lock-out time [R]:
The Lock-out time ‘R’ is defined as the time during which the chatter filter is ON (if triggered), before turned OFF again. It is programmable from 0.1 to 6553.5 sec, in units of 100 ms.

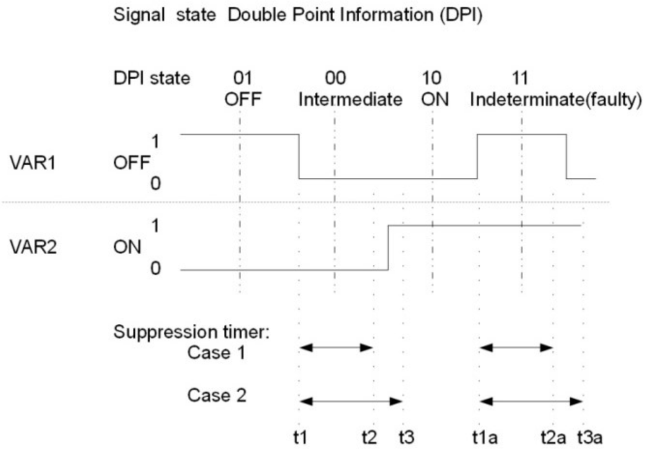
Chatter function diagram:



The example of the chatter filter functions includes also the debounce filter as they are interconnected and will often both be used on applications. Detailed diagram with comments to be found in Appendix 1

Double point intermediate suppression filter

The transmission of double point intermediate (DPI=0) and indeterminate state (DPI=3) (see IEC60870-5-101 / 104) can be prevented by specifying “Intermediate suppress time” in the RTU32 IEC60870 Configurator for each SMDP (Send in Monitor direction Double Point information) signal type. The time is entered in number of 100 ms units from 1 to 65000 (100 ms to 6500 seconds). A value of 0 disables the filter.



The figure shows the signal definition for the DPI. Double indication is represented by two DI. There are no requirements to the position of these two DI on the I/O board. They can in fact be on placed on two different I/O boards.
The normal state of a DPI is either 01 or 10 (OFF or ON).
When the DPI enters either 00 or 11 state a suppression timer is started. If the timer runs out and the state is still either 00 or 11 (Case 1- t2 or 1-t2a) then a DP information message is sent to the controlling station with DPI 0 or 3. The message is sent at time t2.
The reported time tag depends on the hardware profile that is used. RTU32_SOE profile reports time tag t1 as reported by the debounce and chattering filters. RTU32DigIO2 reports time tag t2.

In Case 2 the suppression timer does not run out (t3 or t3a) before signal enters normal state and thus no DPI 0 or 3 message is generated.

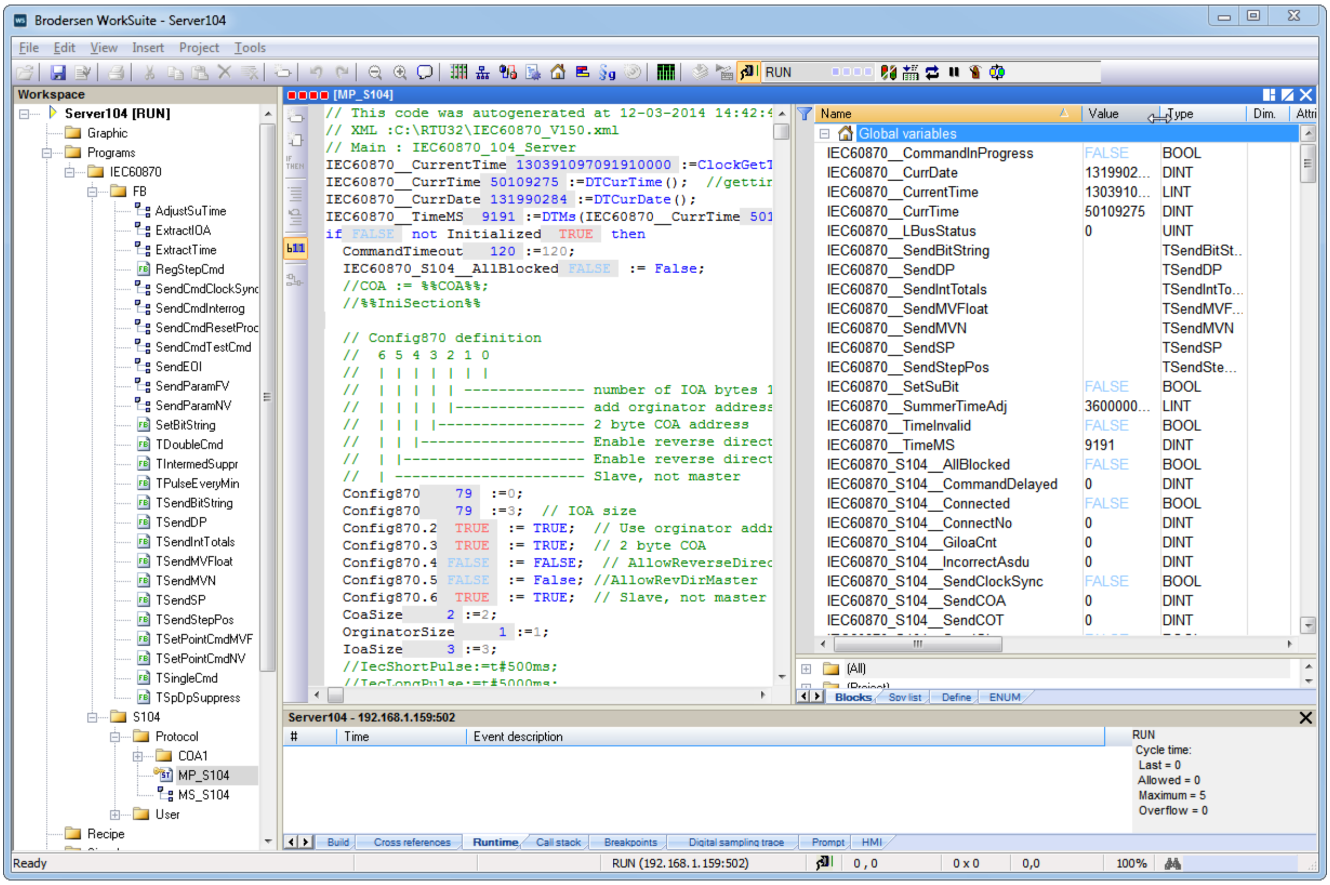
7. The project in Brodersen WorkSuite

General

The Brodersen WorkSuite project is generated after a fixed template with one or more IEC10x program and a number of Function Blocks. The IEC101 or IEC104 are programs which include all the specific code for that actual project. The Functions Blocks are fixed and are just added in as standard functions depending of what ASDU types that are required in the project.

The generated IEC10x program is generated in Structured Text code and structured in order to maintain the same template for all possible driver options. We have included comments in the Structured Text code for helping experienced users to edit the project. If you are not experience Brodersen WorkSuite programmer and familiar with the basic IEC60870 protocol specifications, we must highly recommend that you ask your local distributor for training.

The Brodersen WorkSuite project code that is generated by the generator is placed in a separate project folder named IEC60870.



```
// This code was autogenerated at 12-03-2014 14:42:4
// XML :C:\RTU32\IEC60870_V150.xml
// Main : IEC60870_104_Server
IEC60870__CurrentTime 130391097091910000 :=ClockGetT
IEC60870__CurrTime  50109275 :=DTCurTime();  //gettir
IEC60870__CurrDate  131990284 :=DTCurDate();
IEC60870__TimeMS  9191 :=DTMs(IEC60870__CurrTime 501
if FALSE  not Initialized  TRUE  then
  CommandTimeout   120 :=120;
  IEC60870_S104__AllBlocked FALSE  := False;
  //COA := %%COA%%;
  //%%IniSection%%

  // Config870 definition
  //  6 5 4 3 2 1 0
  //  | | | | | | |
  //  | | | | | --------------- number of IOA bytes 1
  //  | | | | |---------------- add orginator address
  //  | | | |------------------ 2 byte COA address
  //  | | |-------------------- Enable reverse direct
  //  | |---------------------- Enable reverse direct
  //  | ----------------------- Slave, not master
  Config870     79  :=0;
  Config870     79  :=3;  // IOA size
  Config870.2  TRUE  := TRUE;  // Use orginator addr
  Config870.3  TRUE  := TRUE;  // 2 byte COA
  Config870.4 FALSE  := FALSE;  // AllowReverseDirec
  Config870.5 FALSE  := False; //AllowRevDirMaster
  Config870.6  TRUE  := TRUE;  // Slave, not master
  CoaSize      2 :=2;
  OrginatorSize      1 :=1;
  IoaSize      3 :=3;
  //IecShortPulse:=t#500ms;
  //IecLongPulse:=t#5000ms;
```

Name	Value	Type	Dim.	Attri
Global variables				
IEC60870__CommandInProgress	FALSE	BOOL		
IEC60870__CurrDate	1319902...	DINT		
IEC60870__CurrentTime	1303910...	LINT		
IEC60870__CurrTime	50109275	DINT		
IEC60870__LBusStatus	0	UINT		
IEC60870__SendBitString		TSendBitSt..		
IEC60870__SendDP		TSendDP		
IEC60870__SendIntTotals		TSendIntTo...		
IEC60870__SendMVFloat		TSendMVF...		
IEC60870__SendMVN		TSendMVN		
IEC60870__SendSP		TSendSP		
IEC60870__SendStepPos		TSendSte...		
IEC60870__SetSuBit	FALSE	BOOL		
IEC60870__SummerTimeAdj	3600000...	LINT		
IEC60870__TimeInvalid	FALSE	BOOL		
IEC60870__TimeMS	9191	DINT		
IEC60870_S104__AllBlocked	FALSE	BOOL		
IEC60870_S104__CommandDelayed	0	DINT		
IEC60870_S104__Connected	FALSE	BOOL		
IEC60870_S104__ConnectNo	0	DINT		
IEC60870_S104__GiloaCnt	0	DINT		
IEC60870_S104__IncorrectAsdu	0	DINT		
IEC60870_S104__SendClockSync	FALSE	BOOL		
IEC60870_S104__SendCOA	0	DINT		
IEC60870_S104__SendCOT	0	DINT		

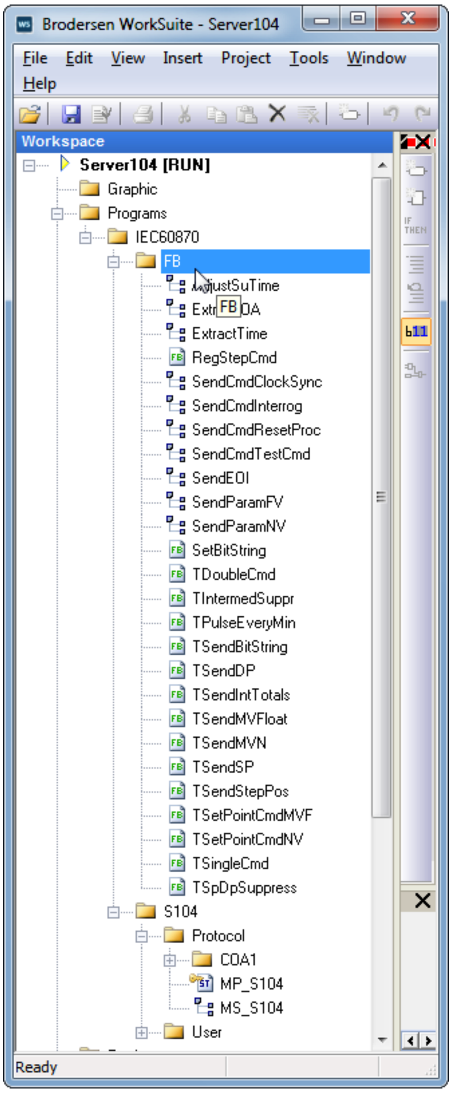
#	Time	Event description

```
RUN
 Cycle time:
  Last = 0
  Allowed = 0
  Maximum = 5
  Overflow = 0
```

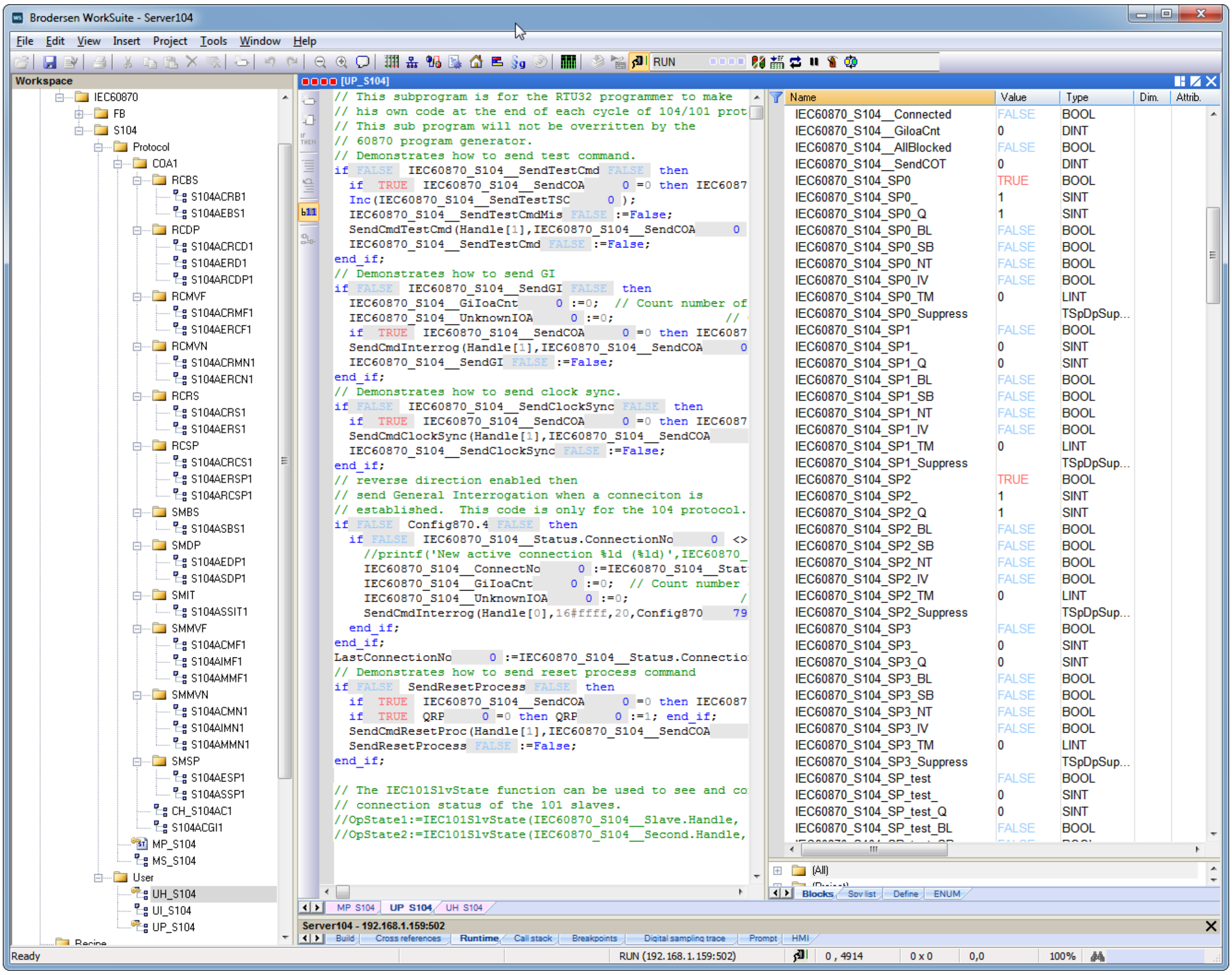
Programs and functions

Under the IEC60870 folder a number of sub-folders are generated.
The "FB" folder includes all the Function blocks used by the driver. The Function blocks are handling IEC870 driver functions and are fixed code blocks that cannot be edited.
The folder called "S104" as above is the name you give it in the Excel Main sheet as the Main Program Name. It includes again 2 sub-folders; "Protocol" which have the driver handling programs and "User" which have some examples of user programs.

Under Protocol you will find folders with sub program generated to handle all the ASDU types for each COA defined



Variables for driver and ASDU types (101 Slave and 104 Server)



```
// This subprogram is for the RTU32 programmer to make
// his own code at the end of each cycle of 104/101 prot
// This sub program will not be overritten by the
// 60870 program generator.
// Demonstrates how to send test command.
if FALSE  IEC60870_S104__SendTestCmd FALSE  then
  if  TRUE  IEC60870_S104__SendCOA      0 =0 then IEC6087
  Inc(IEC60870_S104__SendTestTSC      0 );
  IEC60870_S104__SendTestCmdMis FALSE :=False;
  SendCmdTestCmd(Handle[1],IEC60870_S104__SendCOA      0
  IEC60870_S104__SendTestCmd FALSE :=False;
end_if;
// Demonstrates how to send GI
if FALSE  IEC60870_S104__SendGI FALSE  then
  IEC60870_S104__GiIoaCnt      0 :=0;  // Count number of
  IEC60870_S104__UnknownIOA      0 :=0;              //
  if  TRUE  IEC60870_S104__SendCOA      0 =0 then IEC6087
  SendCmdInterrog(Handle[1],IEC60870_S104__SendCOA      0
  IEC60870_S104__SendGI FALSE :=False;
end_if;
// Demonstrates how to send clock sync.
if FALSE  IEC60870_S104__SendClockSync FALSE  then
  if  TRUE  IEC60870_S104__SendCOA      0 =0 then IEC6087
  SendCmdClockSync(Handle[1],IEC60870_S104__SendCOA
  IEC60870_S104__SendClockSync FALSE :=False;
end_if;
// reverse direction enabled then
// send General Interrogation when a conneciton is
// established.  This code is only for the 104 protocol.
if FALSE  Config870.4 FALSE  then
  if FALSE  IEC60870_S104__Status.ConnectionNo      0  <>
    //printf('New active connection %ld (%ld)',IEC60870_
    IEC60870_S104__ConnectNo      0 :=IEC60870_S104__Stat
    IEC60870_S104__GiIoaCnt      0 :=0;  // Count number
    IEC60870_S104__UnknownIOA      0 :=0;              /
    SendCmdInterrog(Handle[0],16#ffff,20,Config870     79
  end_if;
end_if;
LastConnectionNo      0 :=IEC60870_S104__Status.Connectio
// Demonstrates how to send reset process command
if FALSE  SendResetProcess FALSE  then
  if  TRUE  IEC60870_S104__SendCOA      0 =0 then IEC6087
  if  TRUE  QRP      0 =0 then QRP      0 :=1; end_if;
  SendCmdResetProc(Handle[1],IEC60870_S104__SendCOA
  SendResetProcess FALSE :=False;
end_if;

// The IEC101SlvState function can be used to see and co
// connection status of the 101 slaves.
//OpState1:=IEC101SlvState(IEC60870_S104__Slave.Handle,
//OpState2:=IEC101SlvState(IEC60870_S104__Second.Handle,
```

Name	Value	Type	Dim.	Attrib.
IEC60870_S104__Connected	FALSE	BOOL		
IEC60870_S104__GiloaCnt	0	DINT		
IEC60870_S104__AllBlocked	FALSE	BOOL		
IEC60870_S104__SendCOT	0	DINT		
IEC60870_S104_SP0	TRUE	BOOL		
IEC60870_S104_SP0_	1	SINT		
IEC60870_S104_SP0_Q	1	SINT		
IEC60870_S104_SP0_BL	FALSE	BOOL		
IEC60870_S104_SP0_SB	FALSE	BOOL		
IEC60870_S104_SP0_NT	FALSE	BOOL		
IEC60870_S104_SP0_IV	FALSE	BOOL		
IEC60870_S104_SP0_TM	0	LINT		
IEC60870_S104_SP0_Suppress		TSpDpSup...		
IEC60870_S104_SP1	FALSE	BOOL		
IEC60870_S104_SP1_	0	SINT		
IEC60870_S104_SP1_Q	0	SINT		
IEC60870_S104_SP1_BL	FALSE	BOOL		
IEC60870_S104_SP1_SB	FALSE	BOOL		
IEC60870_S104_SP1_NT	FALSE	BOOL		
IEC60870_S104_SP1_IV	FALSE	BOOL		
IEC60870_S104_SP1_TM	0	LINT		
IEC60870_S104_SP1_Suppress		TSpDpSup...		
IEC60870_S104_SP2	TRUE	BOOL		
IEC60870_S104_SP2_	1	SINT		
IEC60870_S104_SP2_Q	1	SINT		
IEC60870_S104_SP2_BL	FALSE	BOOL		
IEC60870_S104_SP2_SB	FALSE	BOOL		
IEC60870_S104_SP2_NT	FALSE	BOOL		
IEC60870_S104_SP2_IV	FALSE	BOOL		
IEC60870_S104_SP2_TM	0	LINT		
IEC60870_S104_SP2_Suppress		TSpDpSup...		
IEC60870_S104_SP3	FALSE	BOOL		
IEC60870_S104_SP3_	0	SINT		
IEC60870_S104_SP3_Q	0	SINT		
IEC60870_S104_SP3_BL	FALSE	BOOL		
IEC60870_S104_SP3_SB	FALSE	BOOL		
IEC60870_S104_SP3_NT	FALSE	BOOL		
IEC60870_S104_SP3_IV	FALSE	BOOL		
IEC60870_S104_SP3_TM	0	LINT		
IEC60870_S104_SP3_Suppress		TSpDpSup...		
IEC60870_S104_SP_test	FALSE	BOOL		
IEC60870_S104_SP_test_	0	SINT		
IEC60870_S104_SP_test_Q	0	SINT		
IEC60870_S104_SP_test_BL	FALSE	BOOL		

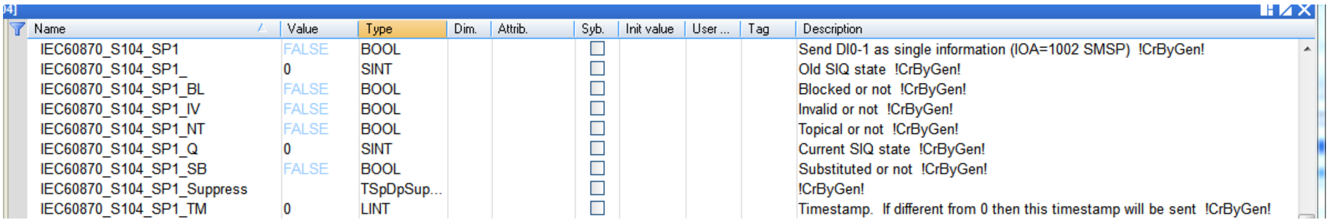
In Brodersen WorkSuite you will find both some general drivers variables and a number of variables for each declared ASDU data type. The general variables are placed in the main program variable folder (e.g. MP_IEC104) and the ASDU specific variables are placed in the Global variables folder. Each variable has its own specific function in the driver and changing anything may lean to errors and a not functional driver.

A few guidelines for the most often used variables is however necessary for the user. Each variable include a description that explain what the variable is used for. For the most commonly used variables, we have included some sections with explanations – see the next sections. In addition this manual will be updated with an Appendix with details of all the possible system and ASDU related variables in a Brodersen WorkSuite application project, Look at our homepage to get the latest version of this document.

Global driver variables (101 Slave and 104 Server)

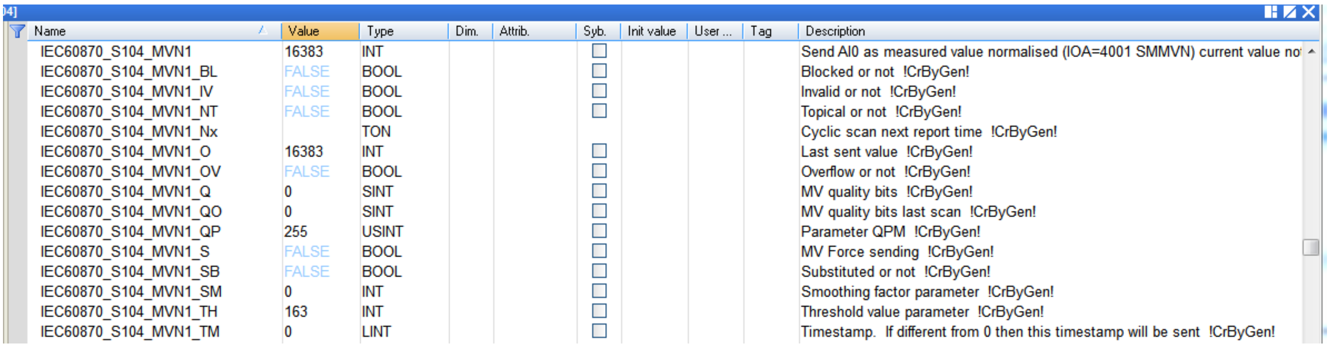
The Global variable folder contains all the ASDU specific variables used by the driver. If you sort the variables after names, they will be listed according to the variables names you have defined in the Excel sheet.
For each declared ASDU type variable a number of Brodersen WorkSuite variable is created. Each of them can be described as driver protocol control parameters. The Single Point Information with the Brodersen WorkSuite variable name SP1 includes the following variables:

SP1 [SINT] = Actual input value
SP1_ [BOOL] = Old input value
SP1_BL [BOOL] = When set to true, the information is blocked
SP1_IV [BOOL] = Invalid bit (part of quality octet)
SP1_NT [BOOL] = Non Topical (part of quality octet)
SP1_Q [SINT] = Quality octet (is the SINT value which include then single bits)
SP1_SB [BOOL] = Substitute bit (part of quality octet)
SP1_TM [LINT] = Time stamp in Win32 format



Name	Value	Type	Dim.	Attrib.	Syb.	Init value	User ...	Tag	Description
IEC60870_S104_SP1	FALSE	BOOL			□				Send DI0-1 as single information (IOA=1002 SMSP) !CrByGen!
IEC60870_S104_SP1_	0	SINT			□				Old SIQ state !CrByGen!
IEC60870_S104_SP1_BL	FALSE	BOOL			□				Blocked or not !CrByGen!
IEC60870_S104_SP1_IV	FALSE	BOOL			□				Invalid or not !CrByGen!
IEC60870_S104_SP1_NT	FALSE	BOOL			□				Topical or not !CrByGen!
IEC60870_S104_SP1_Q	0	SINT			□				Current SIQ state !CrByGen!
IEC60870_S104_SP1_SB	FALSE	BOOL			□				Substituted or not !CrByGen!
IEC60870_S104_SP1_Suppress		TSpDpSup...			□				!CrByGen!
IEC60870_S104_SP1_TM	0	LINT			□				Timestamp. If different from 0 then this timestamp will be sent !CrByGen!

The analogue variables include more variables as for threshold values etc. One of the interesting variables is e.g. xxxx_S which is used for forcing transmission of a Measured value. Description of variables is in the Brodersen WorkSuite application.



Name	Value	Type	Dim.	Attrib.	Syb.	Init value	User ...	Tag	Description
IEC60870_S104_MVN1	16383	INT			□				Send AI0 as measured value normalised (IOA=4001 SMMVN) current value no
IEC60870_S104_MVN1_BL	FALSE	BOOL			□				Blocked or not !CrByGen!
IEC60870_S104_MVN1_IV	FALSE	BOOL			□				Invalid or not !CrByGen!
IEC60870_S104_MVN1_NT	FALSE	BOOL			□				Topical or not !CrByGen!
IEC60870_S104_MVN1_Nx		TON							Cyclic scan next report time !CrByGen!
IEC60870_S104_MVN1_O	16383	INT			□				Last sent value !CrByGen!
IEC60870_S104_MVN1_OV	FALSE	BOOL			□				Overflow or not !CrByGen!
IEC60870_S104_MVN1_Q	0	SINT			□				MV quality bits !CrByGen!
IEC60870_S104_MVN1_QO	0	SINT			□				MV quality bits last scan !CrByGen!
IEC60870_S104_MVN1_QP	255	USINT			□				Parameter QPM !CrByGen!
IEC60870_S104_MVN1_S	FALSE	BOOL			□				MV Force sending !CrByGen!
IEC60870_S104_MVN1_SB	FALSE	BOOL			□				Substituted or not !CrByGen!
IEC60870_S104_MVN1_SM	0	INT			□				Smoothing factor parameter !CrByGen!
IEC60870_S104_MVN1_TH	163	INT			□				Threshold value parameter !CrByGen!
IEC60870_S104_MVN1_TM	0	LINT			□				Timestamp. If different from 0 then this timestamp will be sent !CrByGen!

In general you will find description of the variables in the description row in the Brodersen WorkSuite variable list.

Variables used for managing and controlling the IEC60870-5-104 Client driver.

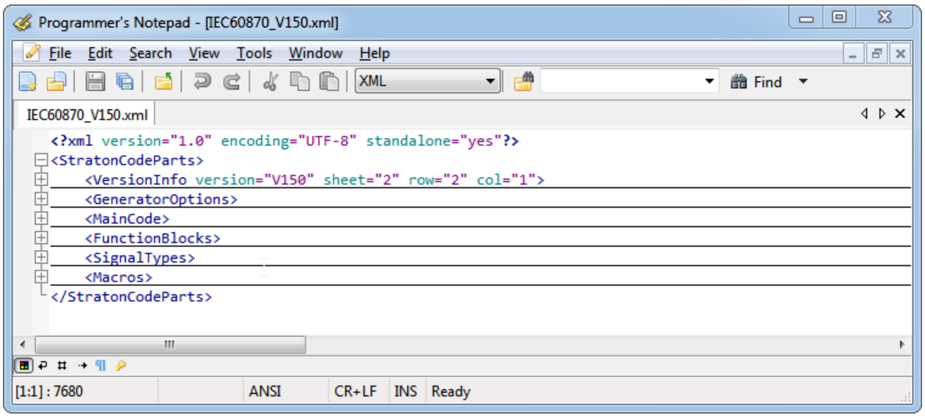
The general management and controlling functions are handled in the Brodersen WorkSuite application by using the variables created by the IEC60870 Configurator.
The Configurator generates the driver and all information's and commands are read/controlled via the generated global variables. The variables used in monitor direction and in control direction are listed in the Appendix 2.

8. The XML Database File

General

The XML file is a database file for the Code Generator and is normally not subject for change.
But experienced users may see it as a way to add application specific parts or changes. This way the code generator can be used in special application.
It includes more or less all settings for the Code Generator which are not entered in the Excel sheet.

The Sections of the XML file


Programmer's Notepad - [IEC60870_V150.xml]
File Edit Search View Tools Window Help
XML
Find
IEC60870_V150.xml
[1:1] : 7680
ANSI
CR+LF
INS
Ready

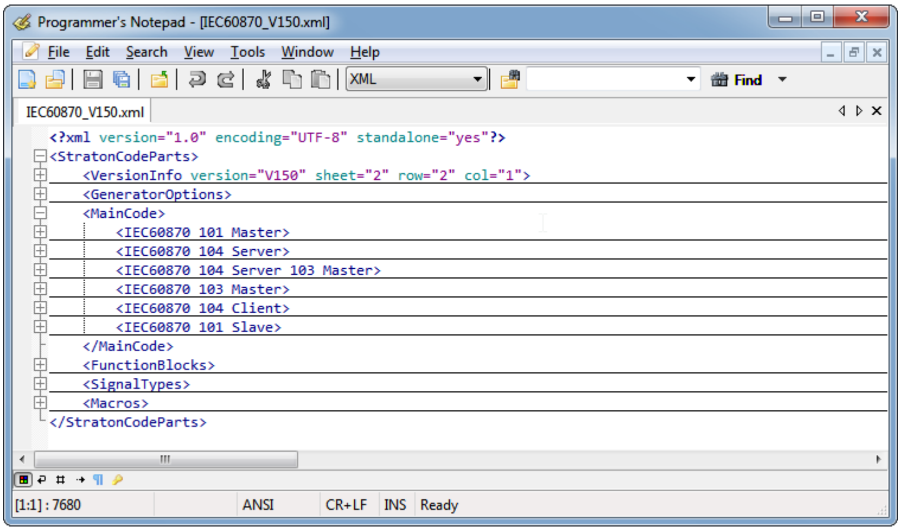
```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<StratonCodeParts>
    <VersionInfo version="V150" sheet="2" row="2" col="1">
    <GeneratorOptions>
    <MainCode>
    <FunctionBlocks>
    <SignalTypes>
    <Macros>
</StratonCodeParts>
```

The picture shows the main sections of the XML file. They are:

- VersionInfo which is used to keep track of which changes are made between versions.
- MainCode section contains the driver types that defined. E.g. 101 Master, 101 Slave, 104 Server etc.
- FunctionBlocks, this section together with the MainCode Section contains definitions of all function blocks that the Configurator can add to a project. It is possible to add your own function blocks here.
- SignalTypes section contains definition of what code and variables are created for each row in the Excel sheet.
- Macros section contains different code that function blocks and signal types repeatedly uses.

The MainCode Section

The MainCode section contains code for all the drivers defined in the XML file.   Here it is possible for experienced user to add new drivers if needed which then could be selected from the Excel sheet “Main”.

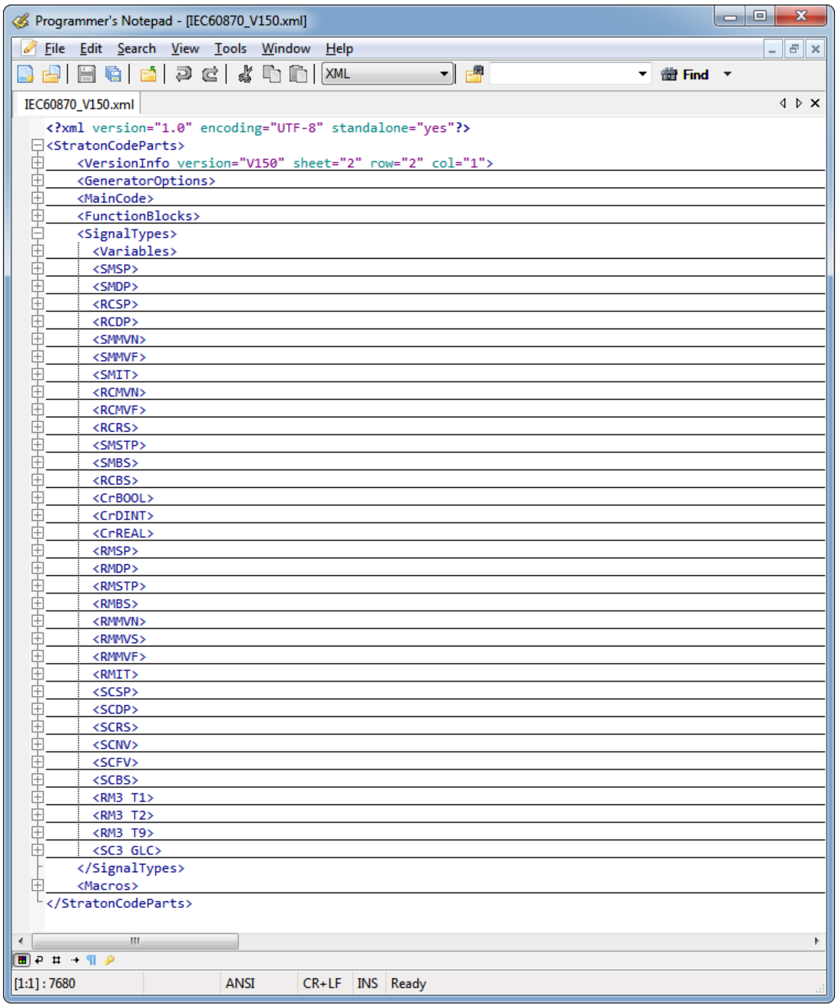

Programmer's Notepad - [IEC60870_V150.xml]
File Edit Search View Tools Window Help
XML
Find
IEC60870_V150.xml
[1:1] : 7680
ANSI
CR+LF
INS
Ready

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<StratonCodeParts>
    <VersionInfo version="V150" sheet="2" row="2" col="1">
    <GeneratorOptions>
    <MainCode>
        <IEC60870 101 Master>
        <IEC60870 104 Server>
        <IEC60870 104 Server 103 Master>
        <IEC60870 103 Master>
        <IEC60870 104 Client>
        <IEC60870 101 Slave>
    </MainCode>
    <FunctionBlocks>
    <SignalTypes>
    <Macros>
</StratonCodeParts>
```

Each driver section contains a list of <UserInput> sections that declares which input is needed and where to read the input from the Excel sheet, with indication of a sheet, row and colon number. It also contains list of <globalvar> sections that declares which global variables are to be created for each driver. Then there is a <Code> section where the structure text code for the driver is entered. The code section also specifies the filename and the folder name of the structure text program.

The SignalType Section

The SignalType section contains all signal types that can be selected in the Signal Type colon of the Excel Sheet. There is a separate section for each Signal Type. This section also contains the <Variables> section which declares how the "Variables" sheet should be read by the configurator.

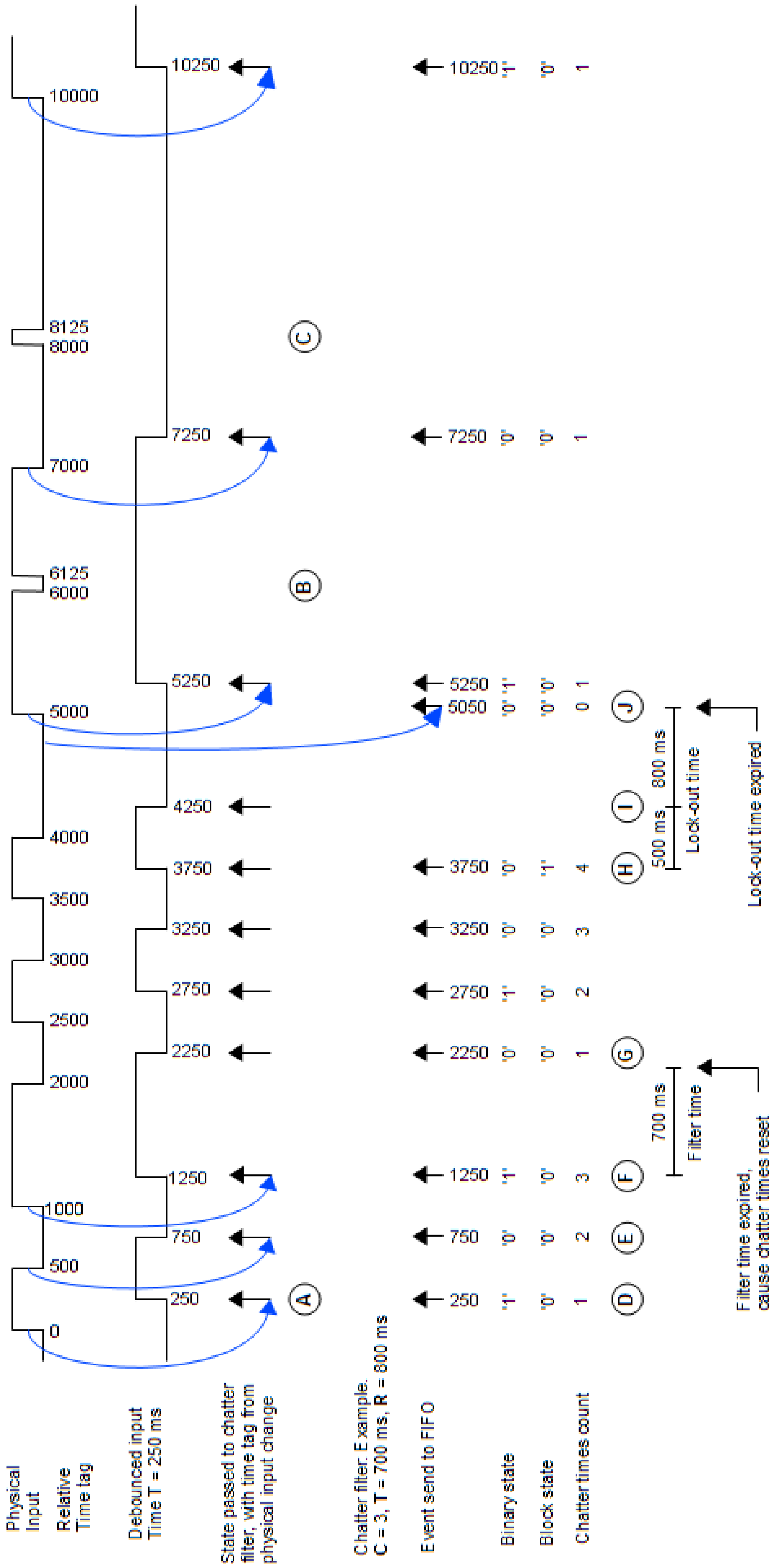

Programmer's Notepad - [IEC60870_V150.xml]
File Edit Search View Tools Window Help
XML
Find
IEC60870_V150.xml
[1:1] : 7680
ANSI
CR+LF
INS
Ready

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<StratonCodeParts>
    <VersionInfo version="V150" sheet="2" row="2" col="1">
    <GeneratorOptions>
    <MainCode>
    <FunctionBlocks>
    <SignalTypes>
      <Variables>
      <SMSP>
      <SMDP>
      <RCSP>
      <RCDP>
      <SMMVN>
      <SMMVF>
      <SMIT>
      <RCMVN>
      <RCMVF>
      <RCRS>
      <SMSTP>
      <SMBS>
      <RCBS>
      <CrBOOL>
      <CrDINT>
      <CrREAL>
      <RMSP>
      <RMDP>
      <RMSTP>
      <RMBS>
      <RMMVN>
      <RMMVS>
      <RMMVF>
      <RMIT>
      <SCSP>
      <SCDP>
      <SCRS>
      <SCNV>
      <SCFV>
      <SCBS>
      <RM3 T1>
      <RM3 T2>
      <RM3 T9>
      <SC3 GLC>
    </SignalTypes>
    <Macros>
</StratonCodeParts>
```

More details of the XML file are provided in the RTU32 IEC60870 Training – contact us for more details about training.

9. Appendix 1

Debounce and chatter filter figure



The time line is relative in ms, just to simplify the diagram. In reality this is a full time stamp.

Point A State change is delayed 250 ms due the 250 ms debounce filter setting. However time stamp used is the time where the physical input, actually changed state.

Point B, C Pulse width is less than 250 ms in both cases, and hence filtered out.

Point D First event pass through the chatter filter without further delay, and is placed in the SOE FIFO queue. Chatter times counter 'C' is incremented to 1

Point E,F Second and third events pass through the chatter filter without further delay, and is placed in the SOE FIFO queue. As events at point E and F, both occur so the pulse width is less than the chatter filter time 'T', 'C' is now incremented to 3.

Point G Time between point F and D events is 1000 ms, which is longer than chatter filter time 'T', so chatter times counter 'C' was reset, and incremented to 1 at point G.

Point H At this point 'C' reach 4, and the filter turns ON. All further input state transition will be blocked, until the input state is stable for a period longer than lock-out time 'R'.

Point I A input state transition occur 500 ms after previous transition, which is within lock-out time 'R'. The time is then reset, and a new full lock-out period 'R', with stable input state, is required before the chatter filter is turned OFF.

Point J Input state is stable longer than 800 ms, which cause the lock-out timer to expire, and the chatter filter to turn OFF. A chatter filter generated event, with block state OFF and

10. Appendix 2 – Brodersen WorkSuite Variable list

List of variables generated in Brodersen WorkSuite application program. Each ASDU type will beside the value it selves include some additional variable for qualifier, status etc. The list below is of possible ASDU types and their global variables in Brodersen WorkSuite.

NOTE: It is only a guide list – see the Brodersen WorkSuite application for details. Each variable created by the generator has a description.

The “<MPN>” in the variable name will be replaced by the “Main Program Name” entered in the “Main” sheet.
The “<Var1>” and “<Var1>” will be replaced with that is entered into the colon Var1 and Var2 in the variable sheet.

Variables Global to All Drivers/Redundancy Groups

Name	**Type**	**Description in Brodersen WorkSuite Variable list**
IEC60870__SendBitString	TSendBitString	Used to send Bitstrings
IEC60870__SendDP	TSendDP	Used to send Double Points
IEC60870__SendIntTotals	TSendIntTotals	Used to send Integrated Totals
IEC60870__SendMVFloat	TSendMVFloat	Used to send Single Points
IEC60870__SendMVN	TSendMVN	Used to send Single Points
IEC60870__SendSP	TSendSP	Used to send Single Points
IEC60870__SendStepPos	TSendStepPos	Used to send StepPosition
IEC60870__LBusStatus	UINT	
IEC60870__CommandInProgress	BOOL	Used to test if a command (SC/DP) select/execute is in progress
IEC60870__CurrentTime	LINT	
IEC60870__CommandDelayed	DINT	
IEC60870__TimeMS	DINT	
IEC60870__CurrDate	DINT	
IEC60870__CurrTime	DINT	
IEC60870__SummerTimeAdj	LINT	
IEC60870__SetSuBit	BOOL	Set SU bit in time field. Mainly used when sending commands. Set to true before sending if SU should be set

Variables Global to Single Driver/Redundancy Group

Name	**Type**	**Description in Brodersen WorkSuite Variable list**
IEC60870_<MPN>__IncorrectAsdu	DINT	
IEC60870_<MPN>__AllBlocked	DINT	All Measured values are blocked
IEC60870_<MPN>__UnknownIOA	DINT	
IEC60870_<MPN>__SendCOA	DINT	
IEC60870_<MPN>__SendTestCmd	BOOL	
IEC60870_<MPN>__SendClockSync	BOOL	
IEC60870_<MPN>__SendGI	BOOL	
IEC60870_<MPN>__SendTestTSC	DINT	
IEC60870_<MPN>__SendTestTSCRec	DINT	
IEC60870_<MPN>__SendTestCmdMis	BOOL	
IEC60870_<MPN>__ConnectNo	DINT	
IEC60870_<MPN>__Connected	BOOL	
IEC60870_<MPN>__GiIoaCnt	DINT	Counts number if IOA received during General Interrogation
IEC60870_<MPN>__SendCOT	DINT	

Master/Client: Receive in Monitor Direction

RMSP – Receive Monitor Direction Single Point

IEC60870_<MPN>_<Var1>	BOOL	Single point (IOA=1 RMSP)
IEC60870_<MPN>_<Var1>_SIQ	SINT	Current SIQ state
IEC60870_<MPN>_<Var1>_NT	BOOL	Topical or not
IEC60870_<MPN>_<Var1>_COT	DINT	Last COT received
IEC60870_<MPN>_<Var1>_TM	LINT	Last receive time
IEC60870_<MPN>_<Var1>_UPD	BOOL	The point is now updated

RMDP – Receive Monitor Direction Double Point

IEC60870_<MPN>_<Var1>	BOOL	Double point (IOA=101 RMDP)
IEC60870_<MPN>_<Var2>	BOOL	Double point (IOA=101 RMDP)
IEC60870_<MPN>_<Var1>_DIQ	SINT	Current DIQ state
IEC60870_<MPN>_<Var1>_NT	BOOL	Topical or not
IEC60870_<MPN>_<Var1>_COT	DINT	Last COT received
IEC60870_<MPN>_<Var1>_TM	LINT	Last receive time
IEC60870_<MPN>_<Var1>_UPD	BOOL	The point is now updated

RMMVN – Receive Monitor Direction Measured Value, Normalized

IEC60870_<MPN>_<Var1>	INT	Norm value (IOA=401 RMMVN)
IEC60870_<MPN>_<Var1>_QDS	USINT	Current QDS state
IEC60870_<MPN>_<Var1>_NT	BOOL	Topical or not
IEC60870_<MPN>_<Var1>_COT	DINT	Last COT received
IEC60870_<MPN>_<Var1>_TM	LINT	Parameter Time to be used when sending command
IEC60870_<MPN>_<Var1>_UPD	BOOL	The point is now updated
IEC60870_<MPN>_<Var1>_PAR	INT	Parameter new value
IEC60870_<MPN>_<Var1>_QPM	USINT	Parameter QPM value
IEC60870_<MPN>_<Var1>_CT	DINT	Parameter COT, Last received Cause of Transmission
IEC60870_<MPN>_<Var1>_NK	BOOL	Parameter NAK. Negative acknowledge
IEC60870_<MPN>_<Var1>_SE	BOOL	Parameter Trigger Send cmd
IEC60870_<MPN>_<Var1>_AV	BOOL	Parameter Activation confirmation, set true if COT=7 is received
IEC60870_<MPN>_<Var1>_TR	TON	Parameter Running timer to measure timeout on the command
IEC60870_<MPN>_<Var1>_TT	BOOL	Parameter Timer trigger. To control if the timer should be running
IEC60870_<MPN>_<Var1>_TE	BOOL	Parameter Timeout error flag, typically activated if time windows are not synchronized

RMMVS – Receive Monitor Direction Measured Value, Scaled

IEC60870_<MPN>_<Var1>	INT	Scaled value (IOA=402 RMMVS)
IEC60870_<MPN>_<Var1>_QDS	USINT	Current QDS state
IEC60870_<MPN>_<Var1>_NT	BOOL	Topical or not
IEC60870_<MPN>_<Var1>_COT	DINT	Last COT received
IEC60870_<MPN>_<Var1>_TM	LINT	Last receive time
IEC60870_<MPN>_<Var1>_UPD	BOOL	The point is now updated

RMMVF – Receive Monitor Direction Measured Value, Floating point

IEC60870_<MPN>_<Var1>	REAL	Float value (IOA=501 RMMVF)
IEC60870_<MPN>_<Var1>_QDS	USINT	Current QDS state
IEC60870_<MPN>_<Var1>_NT	BOOL	Topical or not
IEC60870_<MPN>_<Var1>_COT	DINT	Last COT received
IEC60870_<MPN>_<Var1>_TM	LINT	Parameter Time to be used when sending command
IEC60870_<MPN>_<Var1>_UPD	BOOL	The point is now updated
IEC60870_<MPN>_<Var1>_PAR	REAL	Parameter new value
IEC60870_<MPN>_<Var1>_QPM	USINT	Parameter QPM value
IEC60870_<MPN>_<Var1>_CT	DINT	Parameter COT, Last received Cause of Transmission
IEC60870_<MPN>_<Var1>_NK	BOOL	Parameter NAK. Negative acknowledge
IEC60870_<MPN>_<Var1>_SE	BOOL	Parameter Trigger Send cmd
IEC60870_<MPN>_<Var1>_AV	BOOL	Parameter Activation confirmation, set true if COT=7 is received
IEC60870_<MPN>_<Var1>_TR	TON	Parameter Running timer to measure timeout on the command
IEC60870_<MPN>_<Var1>_TT	BOOL	Parameter Timer trigger. To control if the timer should be running
IEC60870_<MPN>_<Var1>_TE	BOOL	Parameter Timeout error flag, typically activated if time windows are not synchronized

RMSTP – Receive Monitor Direction Step Position

IEC60870_<MPN>_<Var1>	SINT	Step position (IOA=102 RMSTP)
IEC60870_<MPN>_<Var1>_QDS	USINT	Current QDS state
IEC60870_<MPN>_<Var1>_NT	BOOL	Topical or not
IEC60870_<MPN>_<Var1>_COT	DINT	Last COT received
IEC60870_<MPN>_<Var1>_T	BOOL	Transient state
IEC60870_<MPN>_<Var1>_TM	LINT	Last receive time
IEC60870_<MPN>_<Var1>_UPD	BOOL	The point is now updated

RMBS – Receive Monitor Direction Bitstring Value

IEC60870_<MPN>_<Var1>	UDINT	Bitstring (IOA=301 RMBS)
IEC60870_<MPN>_<Var1>_QDS	USINT	Current QDS state
IEC60870_<MPN>_<Var1>_NT	BOOL	Topical or not
IEC60870_<MPN>_<Var1>_COT	DINT	Last COT received
IEC60870_<MPN>_<Var1>_TM	LINT	Last receive time
IEC60870_<MPN>_<Var1>_UPD	BOOL	The point is now updated

RMIT - Receive Monitor Direction Integrated Totals

IEC60870_<MPN>_<Var1>	DINT	Integrated totals (IOA=601 RMIT)
IEC60870_<MPN>_<Var1>_SEQ	USINT	Current QDS state
IEC60870_<MPN>_<Var1>_NT	BOOL	Topical or not
IEC60870_<MPN>_<Var1>_COT	DINT	Last COT received
IEC60870_<MPN>_<Var1>_TM	LINT	Last receive time
IEC60870_<MPN>_<Var1>_UPD	BOOL	The point is now updated

Master/Client: Send in Control Direction

SCSP – Send in Control Direction Single Command

IEC60870_<MPN>_<Var1>	BOOL	Single command (IOA=2001 SCSP)
IEC60870_<MPN>_<Var1>_QU	SINT	QU part of SCO byte
IEC60870_<MPN>_<Var1>_SC	SINT	Current SCO byte
IEC60870_<MPN>_<Var1>_CT	DINT	COT, Last received Cause of Transmission
IEC60870_<MPN>_<Var1>_NK	BOOL	NAK. Negative acknowledge
IEC60870_<MPN>_<Var1>_SS	BOOL	Trigger Send Select cmd
IEC60870_<MPN>_<Var1>_SE	BOOL	Trigger Send Execute cmd
IEC60870_<MPN>_<Var1>_DE	BOOL	Trigger Send deactivation cmd
IEC60870_<MPN>_<Var1>_RS	BOOL	Reset all send states
IEC60870_<MPN>_<Var1>_AV	BOOL	Activation confirmation, set true if COT=7 is received
IEC60870_<MPN>_<Var1>_AT	BOOL	Activation termination. Set to true if COT=10 is received
IEC60870_<MPN>_<Var1>_TM	LINT	Time to be used when sending command
IEC60870_<MPN>_<Var1>_TR	TON	Running timer to measure timeout on the command
IEC60870_<MPN>_<Var1>_TT	BOOL	Timer trigger. To control if the timer should be running
IEC60870_<MPN>_<Var1>_TE	BOOL	Timeout error flag, typically activated if time windows are not synchronized

SCDP – Send in Control Direction Double Command

IEC60870_<MPN>_<Var1>	BOOL	Double command (IOA=2101 SCDP)
IEC60870_<MPN>_<Var1>_QU	SINT	QU part of DCO byte
IEC60870_<MPN>_<Var1>_DC	SINT	Current DCO byte
IEC60870_<MPN>_<Var1>_CT	DINT	COT, Last received Cause of Transmission
IEC60870_<MPN>_<Var1>_NK	BOOL	NAK. Negative acknowledge
IEC60870_<MPN>_<Var1>_SS	BOOL	Trigger Send Select cmd
IEC60870_<MPN>_<Var1>_SE	BOOL	Trigger Send Execute cmd
IEC60870_<MPN>_<Var1>_DE	BOOL	Trigger Send deactivation
IEC60870_<MPN>_<Var1>_RS	BOOL	Reset all send states
IEC60870_<MPN>_<Var1>_AV	BOOL	Activation confirmation, set true if COT=7 is received
IEC60870_<MPN>_<Var1>_AT	BOOL	Activation termination. Set to true if COT=10 is received
IEC60870_<MPN>_<Var1>_TM	LINT	Time to be used when sending command
IEC60870_<MPN>_<Var1>_TR	TON	Running timer to measure timeout on the command
IEC60870_<MPN>_<Var1>_TT	BOOL	Timer trigger. To control if the timer should be running
IEC60870_<MPN>_<Var1>_TE	BOOL	Timeout error flag, typically activated if time windows are not synchronized

SCNV – Send in Control Direction Setpoint Command, Normalized

IEC60870_<MPN>_<Var1>	INT	Normalised set-point (IOA=2401 SCNV) normalized value
IEC60870_<MPN>_<Var1>_QD	SINT	Current QDS byte
IEC60870_<MPN>_<Var1>_CT	DINT	COT, Last received Cause of Transmission
IEC60870_<MPN>_<Var1>_NK	BOOL	NAK. Negative acknowledge
IEC60870_<MPN>_<Var1>_SS	BOOL	Trigger Send Select cmd
IEC60870_<MPN>_<Var1>_SE	BOOL	Trigger Send Execute cmd
IEC60870_<MPN>_<Var1>_DE	BOOL	Trigger Send deactivation
IEC60870_<MPN>_<Var1>_RS	BOOL	Reset all send states
IEC60870_<MPN>_<Var1>_AV	BOOL	Activation confirmation, set true if COT=7 is received
IEC60870_<MPN>_<Var1>_AT	BOOL	Activation termination. Set to true if COT=10 is received

IEC60870_<MPN>_<Var1>_TM	LINT	Time to be used when sending command
IEC60870_<MPN>_<Var1>_TR	TON	Running timer to measure timeout on the command
IEC60870_<MPN>_<Var1>_TT	BOOL	Timer trigger. To control if the timer should be running
IEC60870_<MPN>_<Var1>_TE	BOOL	Timeout error flag, typically activated if time windows are not synchronized

SCFV – Send in Control Direction Setpoint Command, Floating

IEC60870_<MPN>_<Var1>	REAL	Floating set-point (IOA=2501 SCFV) short float value
IEC60870_<MPN>_<Var1>_QD	SINT	Current QDS byte
IEC60870_<MPN>_<Var1>_CT	DINT	COT, Last received Cause of Transmission
IEC60870_<MPN>_<Var1>_NK	BOOL	NAK. Negative acknowledge
IEC60870_<MPN>_<Var1>_SS	BOOL	Trigger Send Select cmd
IEC60870_<MPN>_<Var1>_SE	BOOL	Trigger Send Execute cmd
IEC60870_<MPN>_<Var1>_DE	BOOL	Trigger Send deactivation
IEC60870_<MPN>_<Var1>_RS	BOOL	Reset all send states
IEC60870_<MPN>_<Var1>_AV	BOOL	Activation confirmation, set true if COT=7 is received
IEC60870_<MPN>_<Var1>_AT	BOOL	Activation termination. Set to true if COT=10 is received
IEC60870_<MPN>_<Var1>_TM	LINT	Time to be used when sending command
IEC60870_<MPN>_<Var1>_TR	TON	Running timer to measure timeout on the command
IEC60870_<MPN>_<Var1>_TT	BOOL	Timer trigger. To control if the timer should be running
IEC60870_<MPN>_<Var1>_TE	BOOL	Timeout error flag, typically activated if time windows are not synchronized

SCRS – Send in Control Direction Regulating Step Command

IEC60870_<MPN>_<Var1>	BOOL	Regulating step (IOA=2201 SCRS) FALSE=next step LOWER, TRUE=next step HIGHER
IEC60870_<MPN>_<Var1>_QU	SINT	QU part of RCO byte
IEC60870_<MPN>_<Var1>_RC	SINT	Current RCO byte
IEC60870_<MPN>_<Var1>_CT	DINT	COT, Last received Cause of Transmission
IEC60870_<MPN>_<Var1>_NK	BOOL	NAK. Negative acknowledge
IEC60870_<MPN>_<Var1>_SS	BOOL	Trigger Send Select cmd
IEC60870_<MPN>_<Var1>_SE	BOOL	Trigger Send Execute cmd
IEC60870_<MPN>_<Var1>_RS	BOOL	Reset all send states
IEC60870_<MPN>_<Var1>_AV	BOOL	Activation confirmation, set true if COT=7 is received
IEC60870_<MPN>_<Var1>_AT	BOOL	Activation termination. Set to true if COT=10 is received
IEC60870_<MPN>_<Var1>_TM	LINT	Time to be used when sending command
IEC60870_<MPN>_<Var1>_TR	TON	Running timer to measure timeout on the command
IEC60870_<MPN>_<Var1>_TT	BOOL	Timer trigger. To control if the timer should be running
IEC60870_<MPN>_<Var1>_TE	BOOL	Timeout error flag, typically activated if time windows are not synchronized

SCBS – Send in Control Direction Bitstring Command

IEC60870_<MPN>_<Var1>	UDINT	Bitstring command (IOA=2301 SCBS) bit string
IEC60870_<MPN>_<Var1>_CT	DINT	COT, Last received Cause of Transmission
IEC60870_<MPN>_<Var1>_NK	BOOL	NAK. Negative acknowledge
IEC60870_<MPN>_<Var1>_SE	BOOL	Trigger Send cmd
IEC60870_<MPN>_<Var1>_RS	BOOL	Reset all send states
IEC60870_<MPN>_<Var1>_AV	BOOL	Activation confirmation, set true if COT=7 is received
IEC60870_<MPN>_<Var1>_AT	BOOL	Activation termination. Set to true if COT=10 is received
IEC60870_<MPN>_<Var1>_TM	LINT	Time to be used when sending command
IEC60870_<MPN>_<Var1>_TR	TON	Running timer to measure timeout on the command
IEC60870_<MPN>_<Var1>_TT	BOOL	Timer trigger. To control if the timer should be running

IEC60870_<MPN>_<Var1>_TE	BOOL	Timeout error flag, typically activated if time windows are not synchronized

Slave/Server: Receive in Control Direction

RCSP – Receive in Control Direction Single Command

IEC60870_<MPN>_<Var1>	BOOL	Single command (IOA=2001 RCSP)	
IEC60870_<MPN>_<Var1>_BL	BOOL	Blocked or not	IF = true then command is blocked
IEC60870_<MPN>_<Var1>_Q	SINT	Current SQ state	
IEC60870_<MPN>_<Var1>SC	SingleCmd		
IEC60870_<MPN>_<Var1>_	SINT	Old SQ state	

RCDP – Receive in Control Direction Double Command

IEC60870_<MPN>_<Var1>	BOOL	Double command (IOA=2101 RCDP)	
IEC60870_<MPN>_<Var2>	BOOL	Double command (IOA=2101 RCDP)	
IEC60870_<MPN>_<Var1>_BL	BOOL	Blocked or not	IF = true then command is blocked
IEC60870_<MPN>_<Var1>_	SINT		
IEC60870_<MPN>_<Var1>_Q	SINT		
IEC60870_<MPN>_<Var1>DC	DoubleCmd		

RCMVN – Receive in Control Direction Setpoint Command, Normalized

IEC60870_<MPN>_<Var1>	INT	Normalised set-point (IOA=2401 RCMVN)	
IEC60870_<MPN>_<Var1>_BL	BOOL	Blocked or not	IF = true then command is blocked
IEC60870_<MPN>_<Var1>_TR	BOOL	New command received	
IEC60870_<MPN>_<Var1>_NSP	SetPointCmdNV		
IEC60870_<MPN>_<Var1>_	INT		

RCMVF – Receive in Control Direction Setpoint Command, Floating

IEC60870_<MPN>_<Var1>	REAL	Floating set-point (IOA=2501 RCMVF)	
IEC60870_<MPN>_<Var1>_BL	BOOL	Blocked or not	IF = true then command is blocked
IEC60870_<MPN>_<Var1>_TR	BOOL	New command received	
IEC60870_<MPN>_<Var1>_NSP	SetPointCmdMVF		
IEC60870_<MPN>_<Var1>_	REAL		

RCRS – Receive in Control Direction Regulating Step Command

IEC60870_<MPN>_<Var1>	USINT	Regulating step (IOA=2201 RCRS)	
IEC60870_<MPN>_<Var1>_TR	BOOL	New command received	
IEC60870_<MPN>_<Var1>_BL	BOOL	Blocked or not	IF = true then command is blocked
IEC60870_<MPN>_<Var1>_RS	RegStepCmd		

RCBS – Receive in Control Direction Bitstring Command

IEC60870_<MPN>_<Var1>	UDINT	Bitstring command (IOA=2301 RCBS)	
IEC60870_<MPN>_<Var1>_BL	BOOL	Blocked or not	IF = true then command is blocked
IEC60870_<MPN>_<Var1>SB	SetBitString		
IEC60870_<MPN>_<Var1>_	UDINT		

Slave/Server: Send in Monitor/Reverse Direction

SMSP – Send in Monitor Direction Single Point Information

IEC60870_<MPN>_<Var1>	BOOL	Single point (IOA=1001 SMSP)	
IEC60870_<MPN>_<Var1>_	SINT	Old SIQ state	
IEC60870_<MPN>_<Var1>_Q	SINT	Current SIQ state	
IEC60870_<MPN>_<Var1>_BL	BOOL	Blocked or not	IF = true then information is blocked
IEC60870_<MPN>_<Var1>_SB	BOOL	Substituted or not	
IEC60870_<MPN>_<Var1>_NT	BOOL	Topical or not	
IEC60870_<MPN>_<Var1>_IV	BOOL	Invalid or not	
IEC60870_<MPN>_<Var1>_TM	LINT	Timestamp. If different from 0 then this timestamp will be sent	

SMDP – Send in Monitor Direction Double Point Information

IEC60870_<MPN>_<Var1>	BOOL	Double point (IOA=1101 SMDP)	
IEC60870_<MPN>_<Var2>	BOOL	Double point (IOA=1101 SMDP)	
IEC60870_<MPN>_<Var1>_	SINT	Old DIQ state	
IEC60870_<MPN>_<Var1>_Q	SINT	Current DIQ state	
IEC60870_<MPN>_<Var1>_BL	BOOL	Blocked or not	IF = true then information is blocked
IEC60870_<MPN>_<Var1>_SB	BOOL	Substituted or not	
IEC60870_<MPN>_<Var1>_NT	BOOL	Topical or not	
IEC60870_<MPN>_<Var1>_IV	BOOL	Invalid or not	
IEC60870_<MPN>_<Var1>_TM	LINT	Timestamp. If different from 0 then this timestamp will be sent	

SMMVN – Send in Monitor Direction Measured Value, Normalized

IEC60870_<MPN>_<Var1>	INT	Norm value (IOA=1401 SMMVN) current value not normalized	
IEC60870_<MPN>_<Var1>_TH	INT	Threshold value parameter	
IEC60870_<MPN>_<Var1>_SM	INT	Smoothing factor parameter	
IEC60870_<MPN>_<Var1>_BL	BOOL	Blocked or not	IF = true then information is blocked
IEC60870_<MPN>_<Var1>_SB	BOOL	Substituted or not	
IEC60870_<MPN>_<Var1>_NT	BOOL	Topical or not	
IEC60870_<MPN>_<Var1>_IV	BOOL	Invalid or not	
IEC60870_<MPN>_<Var1>_OV	BOOL	Overflow or not	
IEC60870_<MPN>_<Var1>_O	INT	Last sent value	
IEC60870_<MPN>_<Var1>_Q	SINT	MV quality bits	
IEC60870_<MPN>_<Var1>_QO	SINT	MV quality bits last scan	
IEC60870_<MPN>_<Var1>_Nx	LINT	Cyclic scan next report time	
IEC60870_<MPN>_<Var1>_S	BOOL	MV Force sending	
IEC60870_<MPN>_<Var1>_QP	USINT	Parameter QPM	
IEC60870_<MPN>_<Var1>_TM	LINT	Timestamp. If different from 0 then this timestamp will be sent	

SMMVF – Send in Monitor Direction Measured Value, Floating Point

IEC60870_<MPN>_<Var1>	REAL	Float value (IOA=1501 SMMVF) MVF current value	
IEC60870_<MPN>_<Var1>_TH	REAL	Threshold value parameter	
IEC60870_<MPN>_<Var1>_SM	REAL	Smoothing factor parameter	
IEC60870_<MPN>_<Var1>_BL	BOOL	Blocked or not	IF = true then information is blocked
IEC60870_<MPN>_<Var1>_Q	SINT	Quality bits	
IEC60870_<MPN>_<Var1>_SB	BOOL	Substituted or not	

IEC60870_<MPN>_<Var1>_NT	BOOL	Topical or not
IEC60870_<MPN>_<Var1>_IV	BOOL	Invalid or not
IEC60870_<MPN>_<Var1>_OV	BOOL	Overflow or not
IEC60870_<MPN>_<Var1>_VO	REAL	MV last reported
IEC60870_<MPN>_<Var1>_QO	SINT	MV quality bits last scan
IEC60870_<MPN>_<Var1>_Nx	LINT	Cyclic scan next report time
IEC60870_<MPN>_<Var1>_S	BOOL	MV Force sending
IEC60870_<MPN>_<Var1>_QP	USINT	Parameter QPM
IEC60870_<MPN>_<Var1>_TM	LINT	Timestamp. If different from 0 then this timestamp will be sent

SMSTP – Send in Monitor Direction Step position

IEC60870_<MPN>_<Var1>	SINT	Step position (IOA=1201 SMSTP) Current StepPos state	
IEC60870_<MPN>_<Var1>_VO	SINT	VTI Old state	
IEC60870_<MPN>_<Var1>_VC	SINT	VTI current state	
IEC60870_<MPN>_<Var1>_Q	SINT	Quality bits	
IEC60870_<MPN>_<Var1>_QO	SINT	Old Quality bits	
IEC60870_<MPN>_<Var1>_BL	BOOL	Blocked or not	IF = true then information is blocked
IEC60870_<MPN>_<Var1>_SB	BOOL	Substituted or not	
IEC60870_<MPN>_<Var1>_NT	BOOL	Topical or not	
IEC60870_<MPN>_<Var1>_IV	BOOL	Invalid or not	
IEC60870_<MPN>_<Var1>_OV	BOOL	Overflow or not	
IEC60870_<MPN>_<Var1>_T	BOOL	Current StepPos transient state	
IEC60870_<MPN>_<Var1>_TM	LINT	Timestamp. If different from 0 then this timestamp will be sent	

SMBS – Send in Monitor Direction Bitstring Value

IEC60870_<MPN>_<Var1>	UDINT	Bitstring (IOA=1301 SMBS) Current Bitstring state	
IEC60870_<MPN>_<Var1>_	UDINT	Last scan BitString state	
IEC60870_<MPN>_<Var1>_Q	SINT	Quality bits	
IEC60870_<MPN>_<Var1>_QO	SINT	Last scan quality bits	
IEC60870_<MPN>_<Var1>_BL	BOOL	Blocked or not	IF = true then information is blocked
IEC60870_<MPN>_<Var1>_SB	BOOL	Substituted or not	
IEC60870_<MPN>_<Var1>_NT	BOOL	Topical or not	
IEC60870_<MPN>_<Var1>_IV	BOOL	Invalid or not	
IEC60870_<MPN>_<Var1>_OV	BOOL	Overflow or not	
IEC60870_<MPN>_<Var1>_TM	LINT	Timestamp. If different from 0 then this timestamp will be sent	

SMIT – Send in Monitor Direction Integrated Totals

IEC60870_<MPN>_<Var1>	DINT	Integrated totals (IOA=1601 SMIT) Integrated Total current value
IEC60870_<MPN>_<Var1>_SB	USINT	Sequence Byte that contains IV/CA/CY/SQN
IEC60870_<MPN>_<Var1>_SO	USINT	Old value of Sequence Byte that contains IV/CA/CY/SQN
IEC60870_<MPN>_<Var1>_SQ	USINT	Sequence number
IEC60870_<MPN>_<Var1>_CY	BOOL	Carry
IEC60870_<MPN>_<Var1>_CA	BOOL	Counter adjusted
IEC60870_<MPN>_<Var1>_IV	BOOL	Invalid or not
IEC60870_<MPN>_<Var1>_Nx	LINT	Cyclic scan next report time
IEC60870_<MPN>_<Var1>_S	BOOL	Integrated totals. Force sending
IEC60870_<MPN>_<Var1>_TM	LINT	Timestamp. If different from 0 then this timestamp will be sent
IEC60870_<MPN>_<Var1>_P	TPulseEveryMin	Generates a pulse to control the desired send interval

103 Master Driver

RM3_T1 – Receive in Monitor Direction TypID=1 Time-tagged message

IEC60870_<MPN>_<Var1>	USINT	Received value
IEC60870_<MPN>_<Var1>_COT	USINT	Received Cause of Transmission
IEC60870_<MPN>_<VAR1>_SIN	USINT	Received SUPPLEMENTARY INFORMATION
IEC60870_<MPN>_<VAR1>_Updated	BOOL	Set to TRUE when New value is received
IEC60870_<MPN>_<VAR1>_TM	LINT	Time received from 103 device

RM3_T2 – Receive in Monitor Direction TypID=2 Time-tagged message with relative time

IEC60870_<MPN>_<VAR1>	USINT	Received value
IEC60870_<MPN>_<VAR1>_SIN	USINT	SUPPLEMENTARY INFORMATION
IEC60870_<MPN>_<VAR1>_UD	BOOL	Set to TRUE when new value is received
IEC60870_<MPN>_<VAR1>_TM	LINT	Time received from 103 device

RM3_T9 – Receive in Monitor Direction TypID=9 Measurands II

IEC60870_<MPN>_<VAR1>	REAL	Received value
IEC60870_<MPN>_<VAR1>_IV	BOOL	Received Invalid flag
IEC60870_<MPN>_<VAR1>_OV	BOOL	Received overflow flag
IEC60870_<MPN>_<VAR1>_UD	BOOL	Set to TRUE when new value is received

SC3_GLC – Send in Control Direction TypID=20 General command

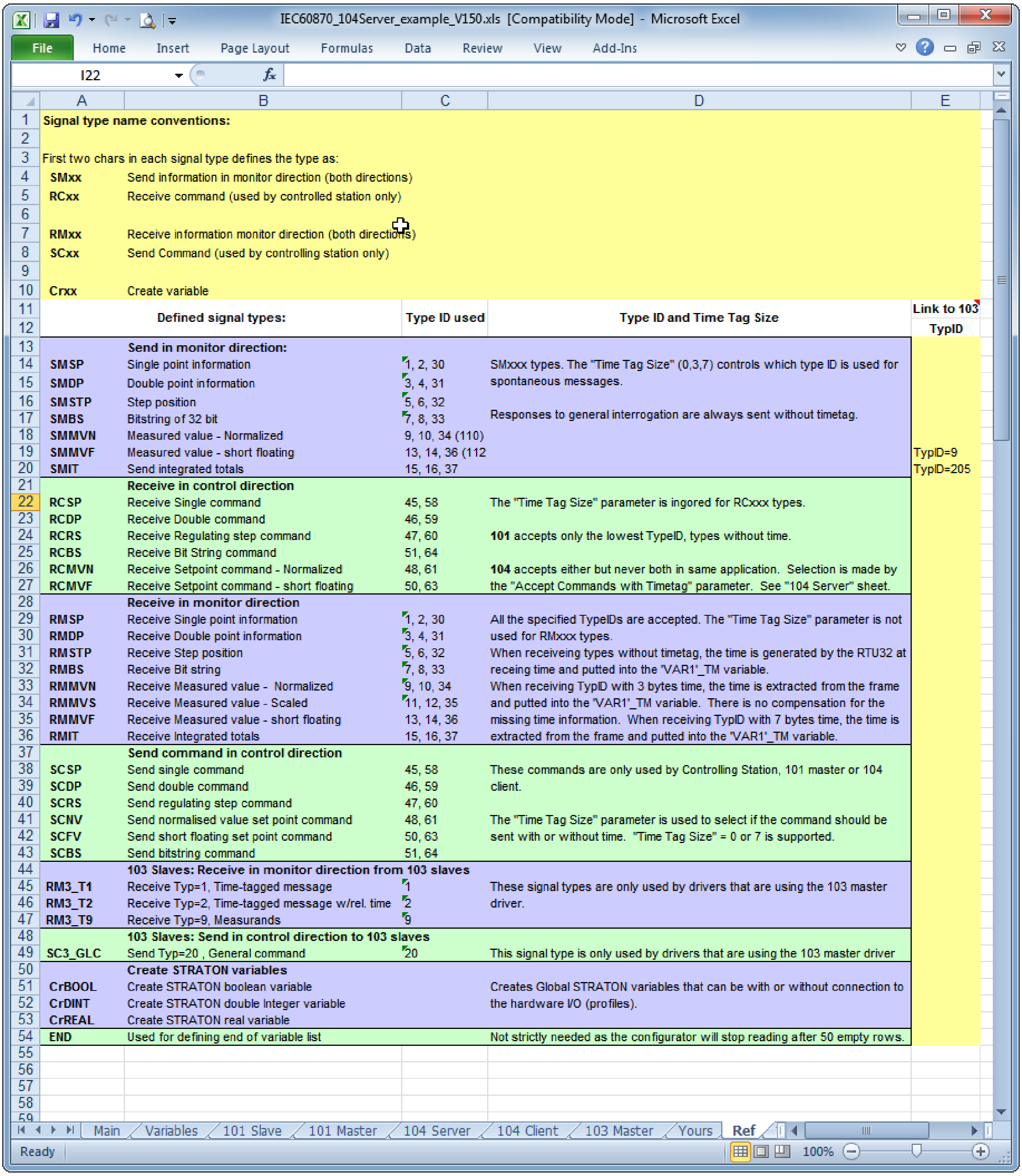
IEC60870_<MPN>_<VAR1>	BOOL	Value of the command
IEC60870_<MPN>_<VAR1>_RII	SINT	RII Return Information Identifier: Incremented and sent in the command
IEC60870_<MPN>_<VAR1>_SIN	USINT	SUPPLEMENTARY INFORMATION returned in the response
IEC60870_<MPN>_<VAR1>_COT	USINT	Cause of Transmission of received response
IEC60870_<MPN>_<VAR1>_Send	BOOL	User must set this to TRUE to trigger Send command
IEC60870_<MPN>_<VAR1>_Timer	TON	Running timer to measure timeout on the command
IEC60870_<MPN>_<VAR1>_TimerTrigger	BOOL	Timer trigger. To control if the timer should be running
IEC60870_<MPN>_<VAR1>_Timeout	BOOL	Command timeout flag, activated if slave does not respond to the command

11. Appendix 3 – Maximum ASDUs to be handled by the IEC60870 Configurator

General requirement / limits	Max ASDUs	
RMSP: Receive Single point information	200	IEC60870 Master/Client functions
RMDP: Receive Double point information	200	
RMSTP: Receive Step position	100	
RMBS: Receive Bit string	50	
RMMVN: Receive Measured value - Normalized	200	
RMMVS: Receive Measured value - Scaled	200	
RMMVF: Receive Measured value - short floating	200	
RMIT: Receive Integrated totals	200	
SCSP: Send single command	200	
SCDP: Send double command	200	
SCRS: Send regulating step command	100	
SCNV: Send normalized value set point command	200	
SCFV: Send short floating set point command	200	
SCBS: Send bitstring command	50	
RCSP: Single command	200	IEC60870 Server/Slave functions
RCDP: Double command	200	
RCRS: Regulating step command	100	
RCBS: Bit String command	50	
RCMVN: Setpoint command - Normalized	200	
RCMVF: Setpoint command - short floating	200	
SMSP: Single point information	400	
SMDP: Double point information	200	
SMSTP: Step position	100	
SMBS: Send Bit string	50	
SMMVN: Measured value - Normalized	200	
SMMVF: Measured value - short floating	200	
SMIT: Send integrated totals	200	
RM3_T1: Receive time-tagged message	400	IEC60870-5-103 Master functions
RM3_T2: Receive time-tagged message with relative time	200	
RM3_T9: Receive measurands II	200	
SC3_GLC: General command	200	

12. Appendix 4 – Excel Ref Sheet

The Excel workbook contains a Ref sheet that contains a short description for each signal type available. It also describes which ASDU type IDs are generated for each signal type.



	A	B	C	D	E
1	**Signal type name conventions:**				
2					
3	First two chars in each signal type defines the type as:				
4	**SMxx**	Send information in monitor direction (both directions)			
5	**RCxx**	Receive command (used by controlled station only)			
6					
7	**RMxx**	Receive information monitor direction (both directions)			
8	**SCxx**	Send Command (used by controlling station only)			
9					
10	**Crxx**	Create variable			
11		**Defined signal types:**	**Type ID used**	**Type ID and Time Tag Size**	**Link to 103**
12					**TypID**
13		**Send in monitor direction:**			
14	**SMSP**	Single point information	1, 2, 30	SMxxx types. The "Time Tag Size" (0,3,7) controls which type ID is used for	
15	**SMDP**	Double point information	3, 4, 31	spontaneous messages.	
16	**SMSTP**	Step position	5, 6, 32		
17	**SMBS**	Bitstring of 32 bit	7, 8, 33	Responses to general interrogation are always sent without timetag.	
18	**SMMVN**	Measured value - Normalized	9, 10, 34 (110)		
19	**SMMVF**	Measured value - short floating	13, 14, 36 (112		TypID=9
20	**SMIT**	Send integrated totals	15, 16, 37		TypID=205
21		**Receive in control direction**			
22	**RCSP**	Receive Single command	45, 58	The "Time Tag Size" parameter is ingored for RCxxx types.	
23	**RCDP**	Receive Double command	46, 59		
24	**RCRS**	Receive Regulating step command	47, 60	**101** accepts only the lowest TypeID, types without time.	
25	**RCBS**	Receive Bit String command	51, 64		
26	**RCMVN**	Receive Setpoint command - Normalized	48, 61	**104** accepts either but never both in same application. Selection is made by	
27	**RCMVF**	Receive Setpoint command - short floating	50, 63	the "Accept Commands with Timetag" parameter. See "104 Server" sheet.	
28		**Receive in monitor direction**			
29	**RMSP**	Receive Single point information	1, 2, 30	All the specified TypeIDs are accepted. The "Time Tag Size" parameter is not	
30	**RMDP**	Receive Double point information	3, 4, 31	used for RMxxx types.	
31	**RMSTP**	Receive Step position	5, 6, 32	When receiveing types without timetag, the time is generated by the RTU32 at	
32	**RMBS**	Receive Bit string	7, 8, 33	receing time and putted into the 'VAR1'_TM variable.	
33	**RMMVN**	Receive Measured value - Normalized	9, 10, 34	When receiving TypID with 3 bytes time, the time is extracted from the frame	
34	**RMMVS**	Receive Measured value - Scaled	11, 12, 35	and putted into the 'VAR1'_TM variable. There is no compensation for the	
35	**RMMVF**	Receive Measured value - short floating	13, 14, 36	missing time information. When receiving TypID with 7 bytes time, the time is	
36	**RMIT**	Receive Integrated totals	15, 16, 37	extracted from the frame and putted into the 'VAR1'_TM variable.	
37		**Send command in control direction**			
38	**SCSP**	Send single command	45, 58	These commands are only used by Controlling Station, 101 master or 104	
39	**SCDP**	Send double command	46, 59	client.	
40	**SCRS**	Send regulating step command	47, 60		
41	**SCNV**	Send normalised value set point command	48, 61	The "Time Tag Size" parameter is used to select if the command should be	
42	**SCFV**	Send short floating set point command	50, 63	sent with or without time. "Time Tag Size" = 0 or 7 is supported.	
43	**SCBS**	Send bitstring command	51, 64		
44		**103 Slaves: Receive in monitor direction from 103 slaves**			
45	**RM3_T1**	Receive Typ=1, Time-tagged message	1	These signal types are only used by drivers that are using the 103 master	
46	**RM3_T2**	Receive Typ=2, Time-tagged message w/rel. time	2	driver.	
47	**RM3_T9**	Receive Typ=9, Measurands	9		
48		**103 Slaves: Send in control direction to 103 slaves**			
49	**SC3_GLC**	Send Typ=20 , General command	20	This signal type is only used by drivers that are using the 103 master driver	
50		**Create STRATON variables**			
51	**CrBOOL**	Create STRATON boolean variable		Creates Global STRATON variables that can be with or without connection to	
52	**CrDINT**	Create STRATON double Integer variable		the hardware I/O (profiles).	
53	**CrREAL**	Create STRATON real variable			
54	**END**	Used for defining end of variable list		Not strictly needed as the configurator will stop reading after 50 empty rows.	
55					
56					
57					
58					

13. Appendix 5 - Update history

Version 1.50 updates

Following updates have been made since version 1.45:

General

IEC60870 Configurator version 1.50 does work with new Brodersen WorkSuite v1.0.0.x and STRATON WorkBench 8.5. Older versions of STRATON WorkBench are not supported. If you use STRATON WorkBench v8.5, you must ensure that the RTU32 CD v1.50 or newer has been installed. It is required to have the correct hardware library in STRATON WorkBench.

RTU32 firmware version 1.52 or newer is required in the RTU32 Series.

IEC60870 Generator updates

The PLC runtime is updated to use cyclic time calculation instead of RTC for SMMVF and SMMVN.
Fixed that the PLC code line "Inc(IEC60870_%%MainPrgName%%__SendTestTSC);" in some OS configuration needs the semi-colon.
Mask bit 3 as it is not an LB error. Just indicates that a variable is accessing invalid module.

Installed files

The program is now in general installed in a "/BRODERSEN" folder instead of the "/BRODERSEN RTU32" folder. All Excel examples are updated with "v150".

IEC60870 Configurator V1.50, V1.45, V1.40 and V1.30 can co-exists on same computer

An effort has been made to make it possible to have the older versions of the Configurator on the same computer. This ensures that projects made with older versions of the Configurator do not need to be upgraded to V1.50 unless you plan to do it.

Version 1.45 updates

Following updates have been made since version 1.40:

103 Master driver

New driver or Main program type added: IEC60870_103_Master. Now there are two 103 drivers available in the XML file: IEC60870_103_Master and IEC60870_104_Server_103_Master. Command direction implemented in both 103 drivers and new signal types added (RM3_T1, RM3_T2, RM3_T9 and SC3_GLC). This update requires RTU32 firmware 1.52+.

New Options in the Generator

The generator contains new Options window and is able to start STRATON Workbench (available since V8) W6Main.

STRATON Workbench V8.3 Installer supported

STRATON WorkBench V8.3 is now installed using the InstallShield installer that installs STRATON WorkBench in \Program Files and \ProgramData folders. This is now detected and support by the Configurator Installer.

Version 1.40 updates

Following updates have been made since version 1.30:

1 of N : SELECT / BREAK OFF /EXECUTE confirmation

While a Single or Double Command is selected and the 101 Slave or 104 Server is waiting for Execute no other select or execute is accepted. This rule is applicable per controlled station.

When a select and execute command (single command (TypID 45,58) or double command (TypID 46,59)) has been given to the controlled station, no new select and execute or direct command (except for TypID 47/48,50,51 or 60,61/63 64) is accepted by the 101 Slave or the 104 Server until the ongoing select and execute command is either completed, postponed or reach timeout.

The Configurator creates the variable *IEC60870__CommandInProgress* that is set to true while a command execution is being processed.

Individual Short / Long Pulse Duration for each Command

The Variable sheet now contains two new colons (Colon J and K) to configure the pulse lengths for Single and Double Commands, used by the Signal Types: RMSP and RMDP.   Here it is possible to configure both short and long pulse length for each command. If the cell is kept empty then the default pulse lengths from the driver main sheet is used.

Enable Spontaneous Information Event in Monitor Direction

This parameter (Colon L) is used by RCSP and RCDP commands. If enabled the change of output caused by the Single or Double commands will cause a spontaneous message in monitor direction. Default settings is now False. (In prior versions this could not be configured and the parameter was always True).

Single/Double point information suppress time filter

This parameter (Colon N) defines how long the single and double point states must be stable before it is sent. This filter is used by the signal types SMSP and SMDP. The filter time is entered in number of 100 ms units.

Longer variable names

Brodersen WorkSuite has since version 7.4 been able to manage long variable names. The IEC60870 Configurator uses this now to make more structured naming of GLOBAL variables created by the Configurator. All variables names are now prefixed as follows:

IEC60870__<xxxxx>
These variables are global variables that are shared by on ore more Excel Workbook (can be used by many drivers/redundancy groups). All these variables are created by the XML file.

IEC60870_<main program name>__<xxxxx>
These variables are global variables that belong to a specific Excel Workbook (a single driver/redundancy group). These variables are created by the XML file.

IEC60870_<main program name>_<Varx from Excel sheet >
Variables that belong to a specific Excel Workbook (single driver/redundancy group). These variables are created by the user, using Var1 and Var2 colons in the Variable sheet.

Extension to the XML language

The XML language tag <var> can now have a new attribute “init=value”. E.g.:

```
<var name="Initialized" type="BOOL" init="False"/>
```

This is used to enter initial values to the variables defined in the XML file. The format of the init value must apply to how Brodersen WorkSuite formats init values.