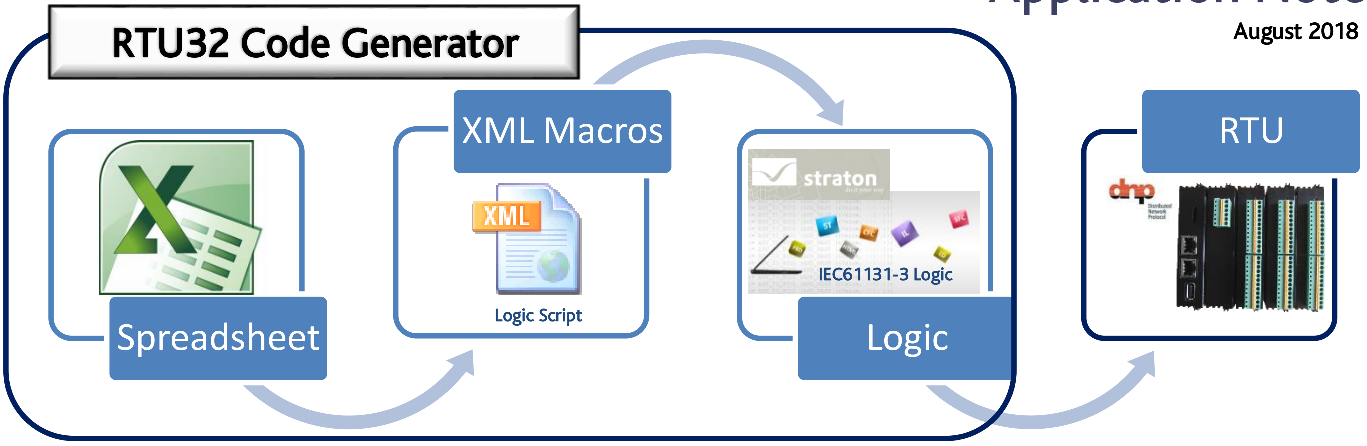
DNP3 Tools for Brodersen RTUs

Auto Generation of RTU32 Series Logic Applications

Application Note

August 2018

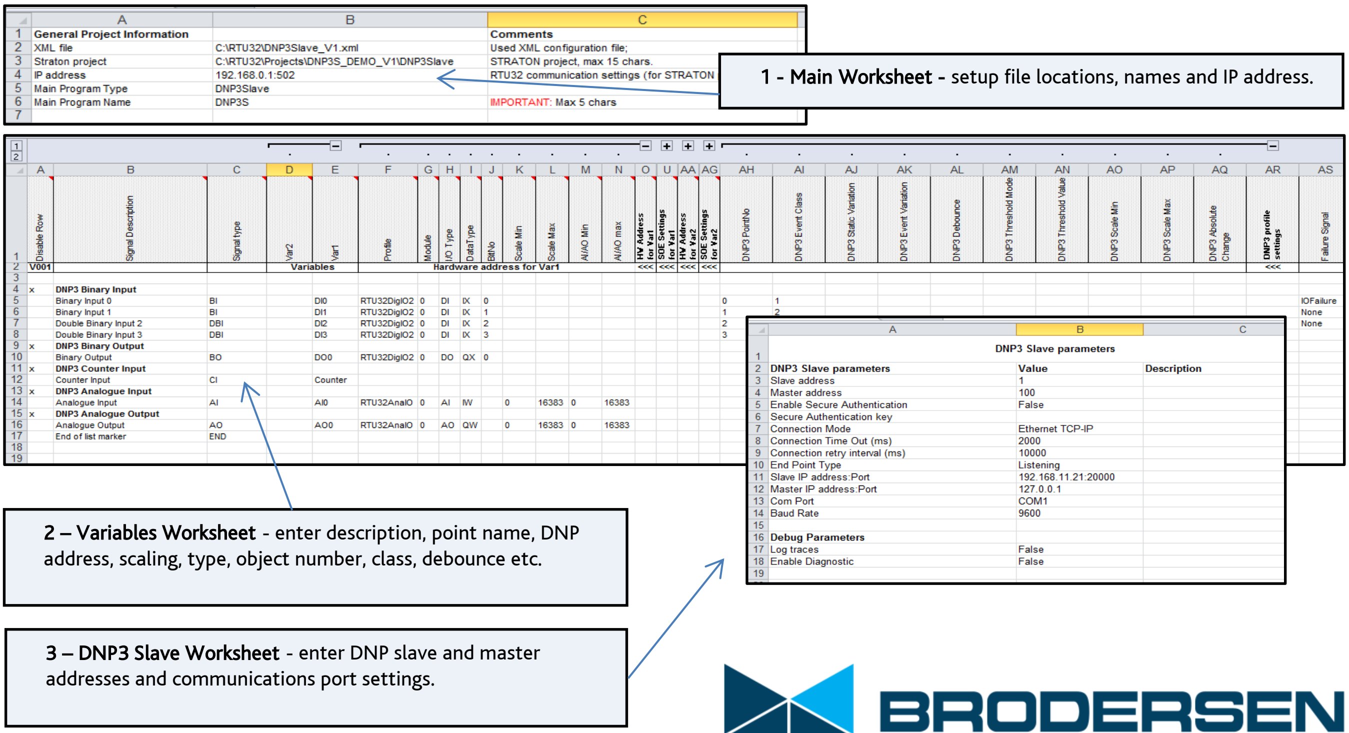


Overview

Brodersen RTU hardware is used in a wide range of applications for the remote management of assets that form the critical infrastructure that keeps our cities and towns operational and safe. These applications include the monitoring and control of wind farms, electrical substations, airports, railways, tunnels, broadcast equipment, IT facilities, building automation, water distribution systems and oil/gas pipelines.

The RTU32 DNP3 Code Generator allows integrators and end users to rapidly deploy Brodersen RTUs in new and existing SCADA systems where DNP3 protocol is being used. An important feature is that all logic code and configuration parameters used in the setup of the RTU are visible and 'exposed', ensuring flexible applications can be created. Major components include an EXCEL Workbook, XML Macro File and a Code Generator.

1) The EXCEL Workbook has three Worksheets that are used to setup the RTU configuration.



	A	B	C
1	**General Project Information**		**Comments**
2	XML file	C:\RTU32\DNP3Slave_V1.xml	Used XML configuration file;
3	Straton project	C:\RTU32\Projects\DNP3S_DEMO_V1\DNP3Slave	STRATON project, max 15 chars.
4	IP address	192.168.0.1:502	RTU32 communication settings (for STRATON
5	Main Program Type	DNP3Slave	
6	Main Program Name	DNP3S	IMPORTANT: Max 5 chars
7			

1 - Main Worksheet - setup file locations, names and IP address.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Disable Row	Signal Description	Signal type	Var2	Var1	Profile	Module	I/O Type	DataType	BitNo	Scale Min	Scale Max	AI/AO Min	AI/AO max
2	V001			Variables		Hardware address for Var1								
3														
4	x	**DNP3 Binary Input**												
5		Binary Input 0	BI		DI0	RTU32DigIO2	0	DI	IX	0				
6		Binary Input 1	BI		DI1	RTU32DigIO2	0	DI	IX	1				
7		Double Binary Input 2	DBI		DI2	RTU32DigIO2	0	DI	IX	2				
8		Double Binary Input 3	DBI		DI3	RTU32DigIO2	0	DI	IX	3				
9	x	**DNP3 Binary Output**												
10		Binary Output	BO		DO0	RTU32DigIO2	0	DO	QX	0				
11	x	**DNP3 Counter Input**												
12		Counter Input	CI		Counter									
13	x	**DNP3 Analogue Input**												
14		Analogue Input	AI		AI0	RTU32AnaIO	0	AI	IW		0	16383	0	16383
15	x	**DNP3 Analogue Output**												
16		Analogue Output	AO		AO0	RTU32AnaIO	0	AO	QW		0	16383	0	16383
17		End of list marker	END											
18														
19														

	O	U	AA	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS
1	**HW Address for Var1**	**SOE Settings for Var1**	**HW Address for Var2**	**SOE Settings for Var2**	DNP3 PointNo	DNP3 Event Class	DNP3 Static Variation	DNP3 Event Variation	DNP3 Debounce	DNP3 Threshold Mode	DNP3 Threshold Value	DNP3 Scale Min	DNP3 Scale Max	DNP3 Absolute Change	**DNP3 profile settings**	Failure Signal
2	<<<	<<<	<<<	<<<											<<<	
3																
4																
5					0	1										IOFailure
6					1	2										None
7					2											None
8					3											

	A	B	C
1		**DNP3 Slave parameters**	
2	**DNP3 Slave parameters**	**Value**	**Description**
3	Slave address	1	
4	Master address	100	
5	Enable Secure Authentication	False	
6	Secure Authentication key		
7	Connection Mode	Ethernet TCP-IP	
8	Connection Time Out (ms)	2000	
9	Connection retry interval (ms)	10000	
10	End Point Type	Listening	
11	Slave IP address:Port	192.168.11.21:20000	
12	Master IP address:Port	127.0.0.1	
13	Com Port	COM1	
14	Baud Rate	9600	
15			
16	**Debug Parameters**		
17	Log traces	False	
18	Enable Diagnostic	False	
19			

2 – Variables Worksheet - enter description, point name, DNP address, scaling, type, object number, class, debounce etc.

3 – DNP3 Slave Worksheet - enter DNP slave and master addresses and communications port settings.

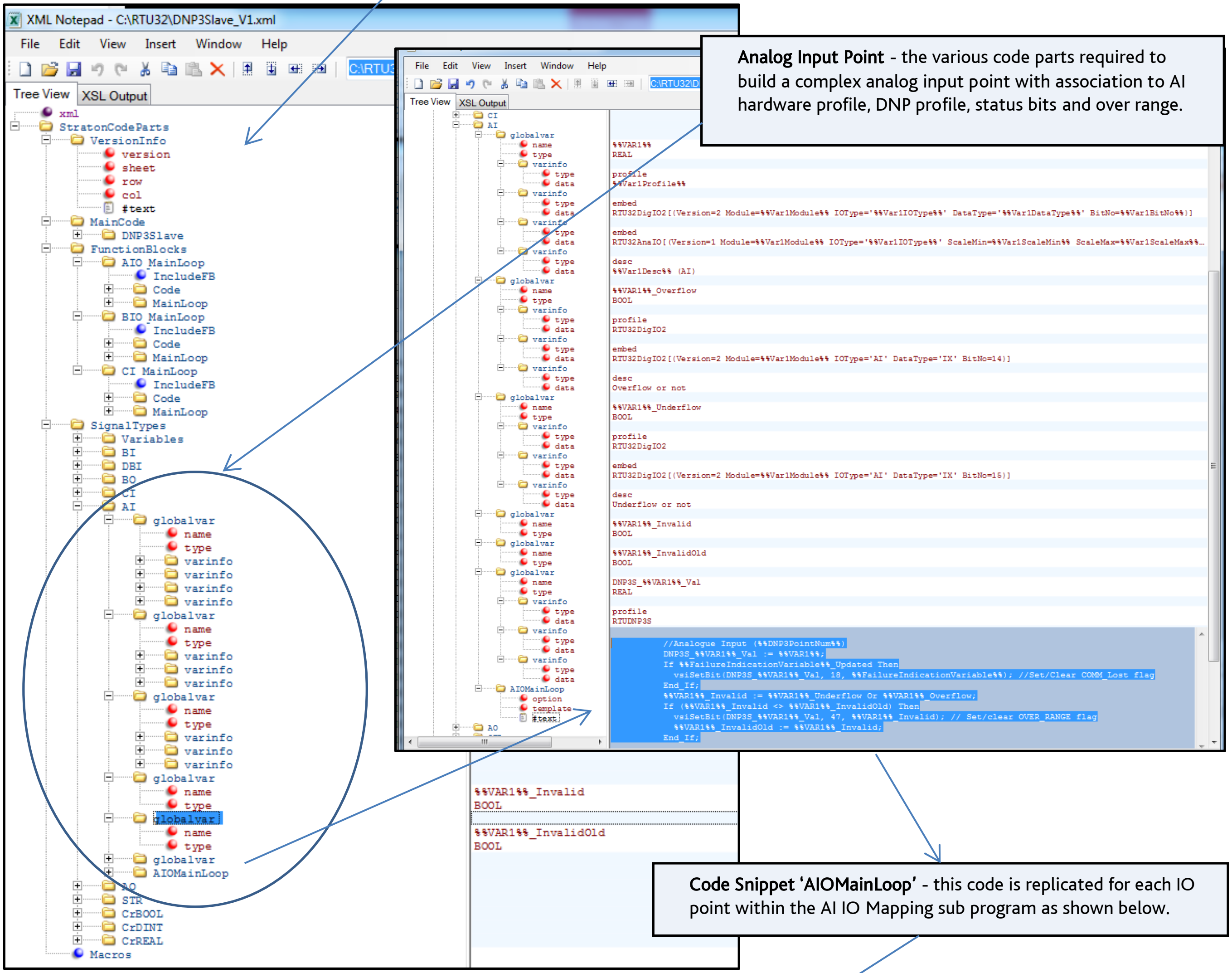
BRODERSEN
simplifying systems

DNP3 Auto Generation Tools – XML Macro File

2) The XML File contains the various code templates and source code ‘snippets’ used in the auto generation of the WorkSuite project variables, DNP3 slave communications setup and code logic.

XML File Overview - the various code parts required to build a WorkSuite project.

.

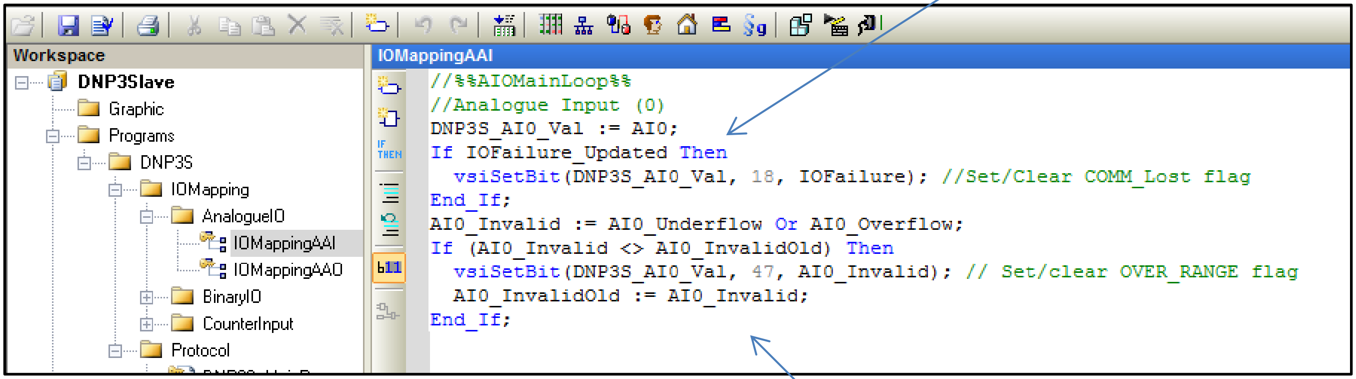


```
%%VAR1%%
REAL

profile
%%Var1Profile%%

embed
RTU32DigIO2[(Version=2 Module=%%Var1Module%% IOType='%%Var1IOType%%' DataType='%%Var1DataType%%' BitNo=%%Var1BitNo%%)]

embed
RTU32AnaIO[(Version=1 Module=%%Var1Module%% IOType='%%Var1IOType%%' ScaleMin=%%Var1ScaleMin%% ScaleMax=%%Var1ScaleMax%%…

desc
%%Var1Desc%% (AI)

%%VAR1%%_Overflow
BOOL

profile
RTU32DigIO2

embed
RTU32DigIO2[(Version=2 Module=%%Var1Module%% IOType='AI' DataType='IX' BitNo=14)]

desc
Overflow or not

%%VAR1%%_Underflow
BOOL

profile
RTU32DigIO2

embed
RTU32DigIO2[(Version=2 Module=%%Var1Module%% IOType='AI' DataType='IX' BitNo=15)]

desc
Underflow or not

%%VAR1%%_Invalid
BOOL

%%VAR1%%_InvalidOld
BOOL

DNP3S_%%VAR1%%_Val
REAL

profile
RTUDNP3S

          //Analogue Input (%%DNP3PointNum%%)
          DNP3S_%%VAR1%%_Val := %%VAR1%%;
          If %%FailureIndicationVariable%%_Updated Then
            vsiSetBit(DNP3S_%%VAR1%%_Val, 18, %%FailureIndicationVariable%%); //Set/Clear COMM_Lost flag
          End_If;
          %%VAR1%%_Invalid := %%VAR1%%_Underflow Or %%VAR1%%_Overflow;
          If (%%VAR1%%_Invalid <> %%VAR1%%_InvalidOld) Then
            vsiSetBit(DNP3S_%%VAR1%%_Val, 47, %%VAR1%%_Invalid); // Set/clear OVER_RANGE flag
            %%VAR1%%_InvalidOld := %%VAR1%%_Invalid;
          End_If;
```

```
%%VAR1%%_Invalid
BOOL

%%VAR1%%_InvalidOld
BOOL
```

Analog Input Point - the various code parts required to build a complex analog input point with association to AI hardware profile, DNP profile, status bits and over range.

Code Snippet ‘AIOMainLoop’ - this code is replicated for each IO point within the AI IO Mapping sub program as shown below.



```
//%%AIOMainLoop%%
//Analogue Input (0)
DNP3S_AI0_Val := AI0;
If IOFailure_Updated Then
  vsiSetBit(DNP3S_AI0_Val, 18, IOFailure); //Set/Clear COMM_Lost flag
End_If;
AI0_Invalid := AI0_Underflow Or AI0_Overflow;
If (AI0_Invalid <> AI0_InvalidOld) Then
  vsiSetBit(DNP3S_AI0_Val, 47, AI0_Invalid); // Set/clear OVER_RANGE flag
  AI0_InvalidOld := AI0_Invalid;
End_If;
```

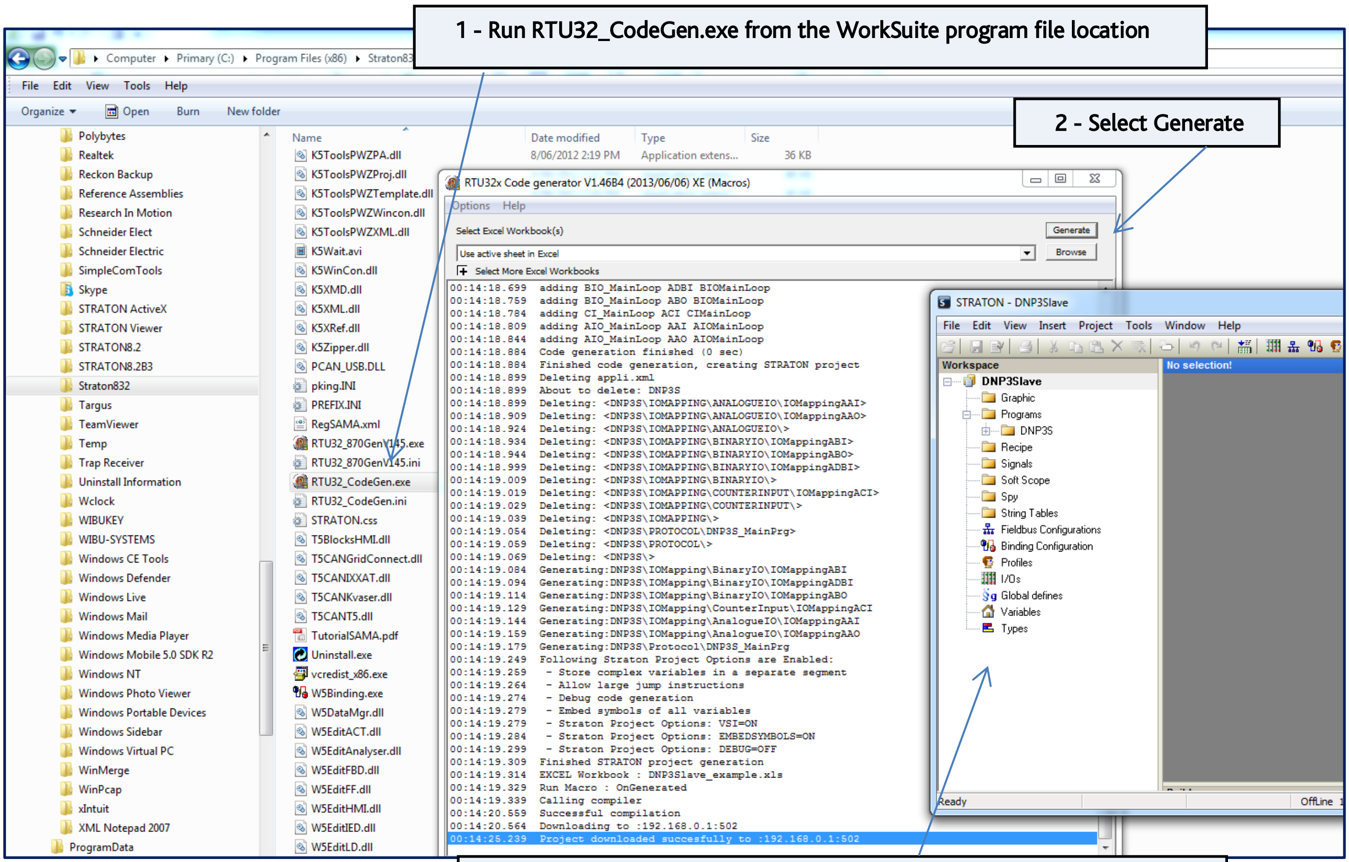
‘AIOMainLoop’ - this code ensures that for each analog input read from IO hardware (AI0), the related DNP3 point (DNP3S_AI0_Val) is updated and status bit 18 set/cleared if IO Bus failure is detected and status bit 47 set/cleared if the point is outside of range limits.

BRODERSEN
simplifying systems

Using the Tools – can it really be this easy?

3) **Copy the files to the following locations, and/or edit the file paths in the Main Worksheet as required.**
C:\RTU32 - the XML file (DNP3Slave_V1.xlm) and EXCEL Workbook (DNP3Slave_example.xls)
C:\RTU32\Projects\DNP3S_DEMO_V1 the location for creation of the WorkSuite project (DNP3Slave)
C:\Program Files(x86)\Brodersen\WorkSuite - the location of the code generator (RTU32_CodeGen.exe)

It is recommended to first open the EXCEL Workbook, then run the code generator from the WorkSuite program files location. Select Generate and the active worksheet in EXCEL will be used to create the code. When code generation is completed, if a valid IP address was entered, the code will be downloaded to the RTU. WorkSuite is then launched and the project is ready to use/review!



1 - Run RTU32_CodeGen.exe from the WorkSuite program file location

2 - Select Generate

```
00:14:18.699  adding BIO_MainLoop ADBI BIOMainLoop
00:14:18.759  adding BIO_MainLoop ABO BIOMainLoop
00:14:18.784  adding CI_MainLoop ACI CIMainLoop
00:14:18.809  adding AIO_MainLoop AAI AIOMainLoop
00:14:18.844  adding AIO_MainLoop AAO AIOMainLoop
00:14:18.884  Code generation finished (0 sec)
00:14:18.884  Finished code generation, creating STRATON project
00:14:18.899  Deleting appli.xml
00:14:18.899  About to delete: DNP3S
00:14:18.899  Deleting: <DNP3S\IOMAPPING\ANALOGUEIO\IOMappingAAI>
00:14:18.909  Deleting: <DNP3S\IOMAPPING\ANALOGUEIO\IOMappingAAO>
00:14:18.924  Deleting: <DNP3S\IOMAPPING\ANALOGUEIO\>
00:14:18.934  Deleting: <DNP3S\IOMAPPING\BINARYIO\IOMappingABI>
00:14:18.944  Deleting: <DNP3S\IOMAPPING\BINARYIO\IOMappingABO>
00:14:18.999  Deleting: <DNP3S\IOMAPPING\BINARYIO\IOMappingADBI>
00:14:19.009  Deleting: <DNP3S\IOMAPPING\BINARYIO\>
00:14:19.019  Deleting: <DNP3S\IOMAPPING\COUNTERINPUT\IOMappingACI>
00:14:19.029  Deleting: <DNP3S\IOMAPPING\COUNTERINPUT\>
00:14:19.039  Deleting: <DNP3S\IOMAPPING\>
00:14:19.054  Deleting: <DNP3S\PROTOCOL\DNP3S_MainPrg>
00:14:19.059  Deleting: <DNP3S\PROTOCOL\>
00:14:19.069  Deleting: <DNP3S\>
00:14:19.084  Generating:DNP3S\IOMapping\BinaryIO\IOMappingABI
00:14:19.094  Generating:DNP3S\IOMapping\BinaryIO\IOMappingADBI
00:14:19.114  Generating:DNP3S\IOMapping\BinaryIO\IOMappingABO
00:14:19.129  Generating:DNP3S\IOMapping\CounterInput\IOMappingACI
00:14:19.144  Generating:DNP3S\IOMapping\AnalogueIO\IOMappingAAI
00:14:19.159  Generating:DNP3S\IOMapping\AnalogueIO\IOMappingAAO
00:14:19.179  Generating:DNP3S\Protocol\DNP3S_MainPrg
00:14:19.249  Following Straton Project Options are Enabled:
00:14:19.259   - Store complex variables in a separate segment
00:14:19.264   - Allow large jump instructions
00:14:19.274   - Debug code generation
00:14:19.279   - Embed symbols of all variables
00:14:19.279   - Straton Project Options: VSI=ON
00:14:19.284   - Straton Project Options: EMBEDSYMBOLS=ON
00:14:19.299   - Straton Project Options: DEBUG=OFF
00:14:19.309  Finished STRATON project generation
00:14:19.314  EXCEL Workbook : DNP3Slave_example.xls
00:14:19.329  Run Macro : OnGenerated
00:14:19.339  Calling compiler
00:14:20.559  Successful compilation
00:14:20.564  Downloading to :192.168.0.1:502
00:14:25.239  Project downloaded succesfully to :192.168.0.1:502
```

3 - If the code compiles OK it will be downloaded to the RTU32 and the WorkSuite application and new DNP3 slave project will be launched.

The code generator can be used a number of ways ie. to create a complete solution or to create a template that provides the base IO and comms portion of a project. Experienced integrators can easily edit the XML files to add or modify the code examples provided to create function blocks, sub-programs and main logic programs.

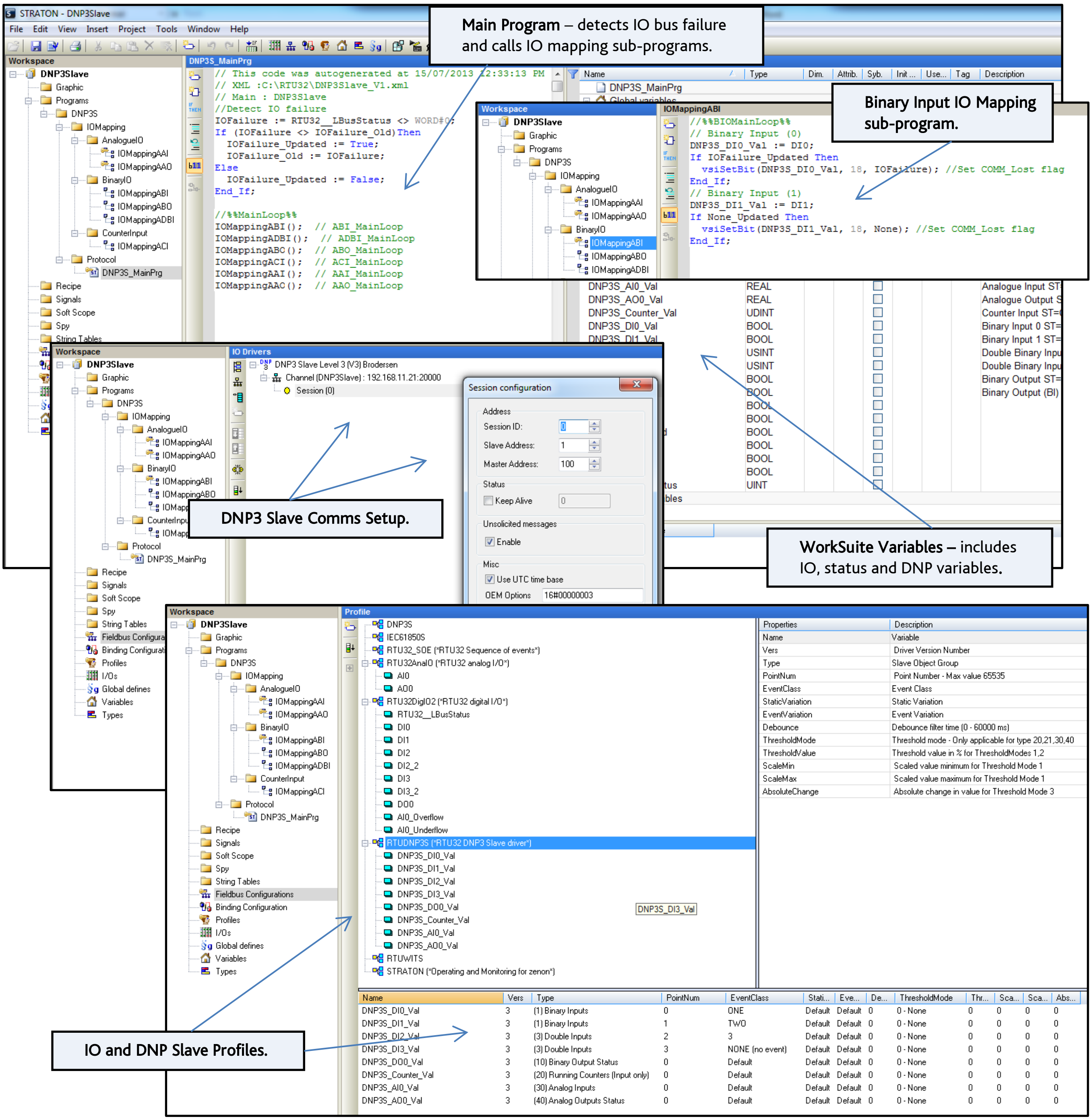
Application Examples for using the RTU32 DNP3 Code Generator Tools:

- A DNP3 'IO Box' – easy. The EXCEL Worksheet contains relevant example IO points – make a few more by copying/editing these points in an easy to use spreadsheet and the project is ready to be generated.
- A complex RTU controller – easy if you know what you are doing! Add your code within the XML file or add it later after the project has been created ie. sometimes it is easier to build a small 'prototype' application in WorkSuite first, then add the code from your proven project to the tools. The code generator will then use/reuse your code as required.

BRODERSEN
simplifying systems

The end result – a ready to use WorkSuite application!

4) A WorkSuite application containing the variables and setup as defined in the EXCEL Workbook.



Main Program – detects IO bus failure and calls IO mapping sub-programs.

DNP3S_MainPrg

```
// This code was autogenerated at 15/07/2013 12:33:13 PM
// XML :C:\RTU32\DNP3Slave_V1.xml
// Main : DNP3Slave
//Detect IO failure
IOFailure := RTU32__LBusStatus <> WORD#0;
If (IOFailure <> IOFailure_Old)Then
  IOFailure_Updated := True;
  IOFailure_Old := IOFailure;
Else
  IOFailure_Updated := False;
End_If;

//%%MainLoop%%
IOMappingABI();  // ABI_MainLoop
IOMappingADBI();  // ADBI_MainLoop
IOMappingABO();  // ABO_MainLoop
IOMappingACI();  // ACI_MainLoop
IOMappingAAI();  // AAI_MainLoop
IOMappingAAO();  // AAO_MainLoop
```

Binary Input IO Mapping sub-program.

IOMappingABI

```
//%%BIOMainLoop%%
// Binary Input (0)
DNP3S_DI0_Val := DI0;
If IOFailure_Updated Then
  vsiSetBit(DNP3S_DI0_Val, 18, IOFailure); //Set COMM_Lost flag
End_If;
// Binary Input (1)
DNP3S_DI1_Val := DI1;
If None_Updated Then
  vsiSetBit(DNP3S_DI1_Val, 18, None); //Set COMM_Lost flag
End_If;
```

WorkSuite Variables – includes IO, status and DNP variables.

Name	Type	Dim.	Attrib.	Syb.	Init ...	Use...	Tag	Description
DNP3S_MainPrg								
Global variables								
DNP3S_AI0_Val	REAL			□				Analogue Input ST
DNP3S_AO0_Val	REAL			□				Analogue Output S
DNP3S_Counter_Val	UDINT			□				Counter Input ST=
DNP3S_DI0_Val	BOOL			□				Binary Input 0 ST=
DNP3S_DI1_Val	BOOL			□				Binary Input 1 ST=
	USINT			□				Double Binary Inpu
	USINT			□				Double Binary Inpu
	BOOL			□				Binary Output ST=
	BOOL			□				Binary Output (BI)
	BOOL			□				
	BOOL			□				
d	BOOL			□				
	BOOL			□				
	BOOL			□				
	BOOL			□				
tus	UINT			□				
bles								

DNP3 Slave Comms Setup.

IO Drivers

- DNP3 Slave Level 3 (V3) Brodersen
  - Channel (DNP3Slave) : 192.168.11.21:20000
    - Session (0)

Session configuration

Address
Session ID: 0
Slave Address: 1
Master Address: 100

Status
☐ Keep Alive 0

Unsolicited messages
☑ Enable

Misc
☑ Use UTC time base
OEM Options 16#00000003

IO and DNP Slave Profiles.

Profile

- DNP3S
- IEC61850S
- RTU32_SOE (*RTU32 Sequence of events*)
- RTU32AnalO (*RTU32 analog I/O*)
  - AI0
  - AO0
- RTU32DigIO2 (*RTU32 digital I/O*)
  - RTU32__LBusStatus
  - DI0
  - DI1
  - DI2
  - DI2_2
  - DI3
  - DI3_2
  - DO0
  - AI0_Overflow
  - AI0_Underflow
- RTUDNP3S (*RTU32 DNP3 Slave driver*)
  - DNP3S_DI0_Val
  - DNP3S_DI1_Val
  - DNP3S_DI2_Val
  - DNP3S_DI3_Val
  - DNP3S_DO0_Val
  - DNP3S_Counter_Val
  - DNP3S_AI0_Val
  - DNP3S_AO0_Val
- RTUWITS
- STRATON (*Operating and Monitoring for zenon*)

DNP3S_DI3_Val

Properties	Description
Name	Variable
Vers	Driver Version Number
Type	Slave Object Group
PointNum	Point Number - Max value 65535
EventClass	Event Class
StaticVariation	Static Variation
EventVariation	Event Variation
Debounce	Debounce filter time (0 - 60000 ms)
ThresholdMode	Threshold mode - Only applicable for type 20,21,30,40
ThresholdValue	Threshold value in % for ThresholdModes 1,2
ScaleMin	Scaled value minimum for Threshold Mode 1
ScaleMax	Scaled value maximum for Threshold Mode 1
AbsoluteChange	Absolute change in value for Threshold Mode 3

Name	Vers	Type	PointNum	EventClass	Stati...	Eve...	De...	ThresholdMode	Thr...	Sca...	Sca...	Abs...
DNP3S_DI0_Val	3	(1) Binary Inputs	0	ONE	Default	Default	0	0 - None	0	0	0	0
DNP3S_DI1_Val	3	(1) Binary Inputs	1	TWO	Default	Default	0	0 - None	0	0	0	0
DNP3S_DI2_Val	3	(3) Double Inputs	2	3	Default	Default	0	0 - None	0	0	0	0
DNP3S_DI3_Val	3	(3) Double Inputs	3	NONE (no event)	Default	Default	0	0 - None	0	0	0	0
DNP3S_DO0_Val	3	(10) Binary Output Status	0	Default	Default	Default	0	0 - None	0	0	0	0
DNP3S_Counter_Val	3	(20) Running Counters (Input only)	0	Default	Default	Default	0	0 - None	0	0	0	0
DNP3S_AI0_Val	3	(30) Analog Inputs	0	Default	Default	Default	0	0 - None	0	0	0	0
DNP3S_AO0_Val	3	(40) Analog Outputs Status	0	Default	Default	Default	0	0 - None	0	0	0	0

The completed RTU application can be downloaded from WorkSuite or transferred using various file transfer methods eg. using FTP, using file transfer from a modern SCADA host (eg. ClearSCADA) or by updating the files on the RTUs flash memory storage card.

RTU32 Series memory cards can be easily updated or 'swapped'.

Additional product information is available from our website, or from the authorised distributor in your region.

BRODERSEN
simplifying systems

www.brodersen.com
Brodersen Systems Asia Pacific P/L
832 High Street, Kew East, VIC 3102, Australia
Tel: +61 3 9249 9505, Email: sales-ap@brodersen.com