RTU32N Series

Compact RTU, PLC and Utility Controller

User Manual

Version 1363 / July 2020
UCN-28IO/1A005.81/91
UCN-42IO/1A005.81/91

C 0 1 2 3 4 5 6 7 C C 8 9 10 11 12 13 14 15 C 1 2 3 4 5 6 7 8 9 10
Digital Inputs / Section A
Digital Inputs / Section B
Expanded IOs / Section E
AI2 56%
BRODERSEN
RTU32
Digital Outputs / Section C
Analog Inputs / Section D
Expanded IOs / Section F
0 1 2 3
+ 0 1 2 3 4 5 6 7 − C + − + − + − + − C 1 2 3 4 5 6 7 8 9 10

BRODERSEN
simplifying systems

Table of Contents

1. Customer Information

1.1 Copyright Notice

1.2 Trademark Acknowledgement

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

1.3 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.

1.4 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.

1.5 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.

1.6 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, and 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
sales@brodersen.com
www.brodersen.com

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

1.7 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

2.1 RTU32N – The compact Brodersen RTU, PLC and Industrial Controller

The Brodersen RTU32N series is based on a 32-bit i.Mx6 ARM platform, providing a small footprint RTU and PLC with leading edge functionality. As well as being a versatile RTU and PLC with flexible I/O designed to perform embedded data processing, control, data logging and monitoring, it is also a networking communicator for collecting, managing and communicating data via protocols on different physical interfaces upwards and downwards in an industrial environment.

Typical applications for the RTU32N are:

- Control and monitoring of water and waste water pump stations
- Substation automation applications
- Railway level crossing monitoring
- Communication Gateway applications between SCADA protocols – including Modbus, DNP3 and IEC60870.

The RTU32N hardware platform includes a 200-900MHz CPU with 128-256MB RAM and integrated 128MB NAND Flash. In addition, an optional removable microSD Flash card (default is 4GB, supports up to 64GB), provides an easy to use portable access/upgrade option. The integrated Flash stores the main RTU32 firmware including embedded Linux OS and your PLC runtime application. Standard drivers like the Modbus Protocol Suite, IEC60870 or DNP3 are stored in the integrated NAND Flash. The removable microSD card can be used for storing backup files and data logging files, or as the 'boot source'. When the SD card is used as the boot source, all firmware and applications are then stored on the SD card, allowing for easy 'swap and go' (simply move the SD card to another CPU module if required).

2.2 WebServer configuration and PLC programming in all IEC61131 languages

The RTU32N, like the other RTU32 Series of RTUs is configured via the integrated WebServer. Settings like LAN port addressing, time functions and 3g/4g modem credentials are handled in webpages. Online monitoring of connected physical I/O and display of graphical overviews is possible via webpages.

Programming of logic functions and configuration of communications drivers is managed using the Brodersen WorkSuite software. WorkSuite is a powerful tool that is used for configuration and programming of all RTU32 Series products. Applications developed and running in the RTU32N will easily run on other RTU32 Series products. The WorkSuite software package will run in demo mode without any license (with some runtime restrictions).

3 Getting Started

3.1 Configuration and default settings

The LAN1 interface on the RTU32N is set to its factory default settings at delivery:

LAN1: 192.168.0.1, Subnet: 255.255.255.0

You can use the LAN1 interface to configure the basic settings in the RTU32N. The basic settings include setup of network parameters such as IP addresses, Subnet masks, default gateways, time settings, usernames and passwords for accessing the web server pages and file system with SFTP, SSH etc.

The configuration is done via web pages in the RTU32N Web Server. You must use your normal web browser to setup the basic settings for the RTU32N. The web pages are optimized for MS Internet Explorer/Edge, Firefox and Chrome. If you are experiencing problems with using the latest MS Internet Explorer/Edge version, please enable Compatibility settings in the browser.

4 How to connect and prepare your PC to configure the RTU32N

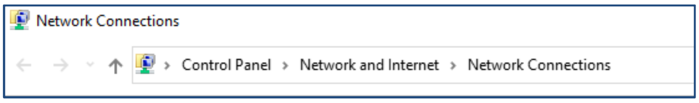
4.1 LAN Port Connection

Use a cross-wired or standard patch cable to connect your PC Ethernet interface to the LAN1 Ethernet interface of the RTU.

4.2 Setting up your PC Ethernet driver (Windows 10 example shown)

Your PC Ethernet driver must be setup to work in the same network segment as the LAN interface on the RTU32 if you want to connect with LAN1 which has a default IP address of 192.168.0.1. You must setup your PC Ethernet TCP/IP driver to work in the segment IP 192.168.0.x. Alternatively you can connect with LAN 2 which is set to DHCP by default. To connect via LAN2 you must have an appropriate router/DHCP server (you can use the RTU32 Search Tool that installs with WorkSuite to discover the RTU, or connect to your router diagnostic interface to find the IP address allocated to LAN2).

To setup your PC Ethernet driver to connect via Lan1, select *start – Settings – Network & Internet* and select *Change adapter options*.



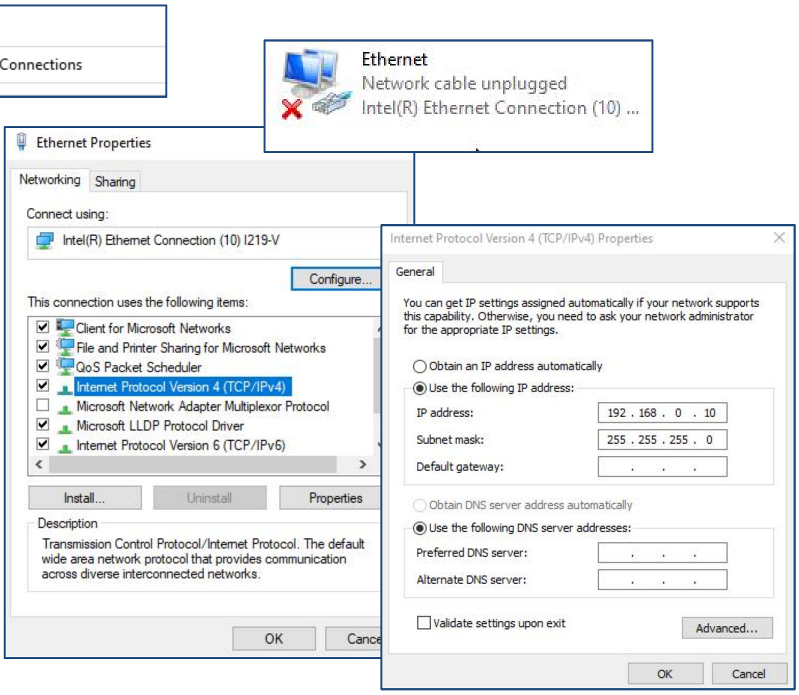
Network Connections

Control Panel › Network and Internet › Network Connections

Select *Properties* of the appropriate adapter and then Click on *Internet Protocol Version 4 (TCP/IPv4)* and select *Properties*. A new window for setting the PC IP address will appear. Set an IP address in the same segment as the selected LAN IP address, for example 192.168.0.10 as shown here below.

NOTE: If you need to re-configure your Ethernet TCP/IP driver back to initial settings, it is recommended that you note the settings before you change them.

Click *OK* and *OK* again - now your PC Ethernet driver is setup for communicating with the RTU32N via LAN1.



Ethernet
Network cable unplugged
Intel(R) Ethernet Connection (10) ...

Ethernet Properties

Networking | Sharing

Connect using:

Intel(R) Ethernet Connection (10) I219-V

Configure...

This connection uses the following items:

- ☑ Client for Microsoft Networks
- ☑ File and Printer Sharing for Microsoft Networks
- ☑ QoS Packet Scheduler
- ☑ Internet Protocol Version 4 (TCP/IPv4)
- □ Microsoft Network Adapter Multiplexor Protocol
- ☑ Microsoft LLDP Protocol Driver
- ☑ Internet Protocol Version 6 (TCP/IPv6)

Install... | Uninstall | Properties

Description

Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

OK | Cance

Internet Protocol Version 4 (TCP/IPv4) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 192 . 168 . 0 . 10

Subnet mask: 255 . 255 . 255 . 0

Default gateway: . . .

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server: . . .

Alternate DNS server: . . .

□ Validate settings upon exit

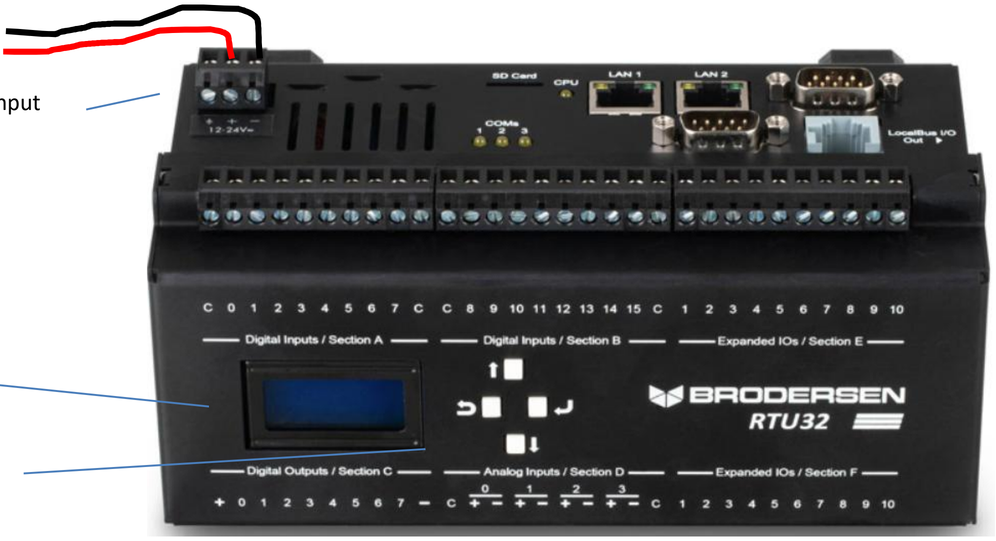
Advanced...

OK | Cancel

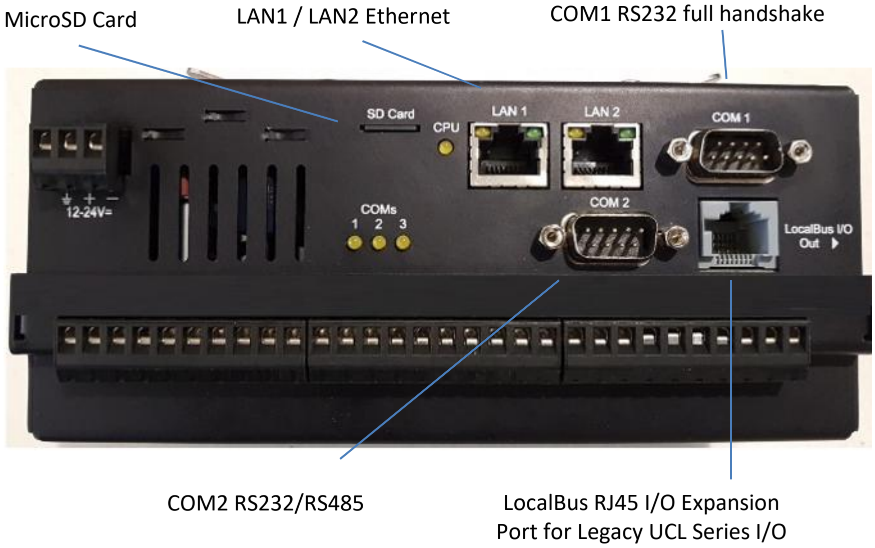
4.3 Applying power to the RTU32N

Power is applied to the RTU32N via the 3 way power connector located in the top left corner of the RTU. Wait for the RTU32N to boot up – it takes approx. 20s. The LCD front panel will light up and the amber Power LED at the bottom of the RTU must be lit.

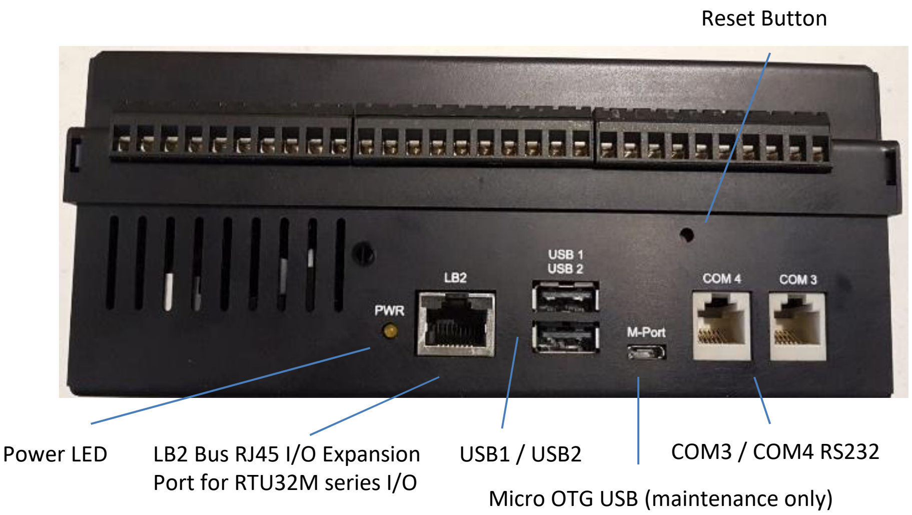
RTU32N – front view



RTU32N – top view

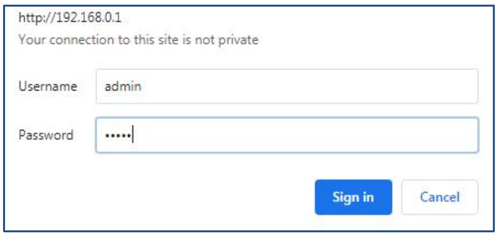

MicroSD Card
LAN1 / LAN2 Ethernet
COM1 RS232 full handshake
COM2 RS232/RS485
LocalBus RJ45 I/O Expansion
Port for Legacy UCL Series I/O

RTU32N – bottom view


Reset Button
Power LED
LB2 Bus RJ45 I/O Expansion
Port for RTU32M series I/O
USB1 / USB2
COM3 / COM4 RS232
Micro OTG USB (maintenance only)

4.4 First time configuration – example connecting via LAN1

Start your Internet browser and enter the IP address of the RTU32N in the address field eg. 192.168.0.1



http://192.168.0.1

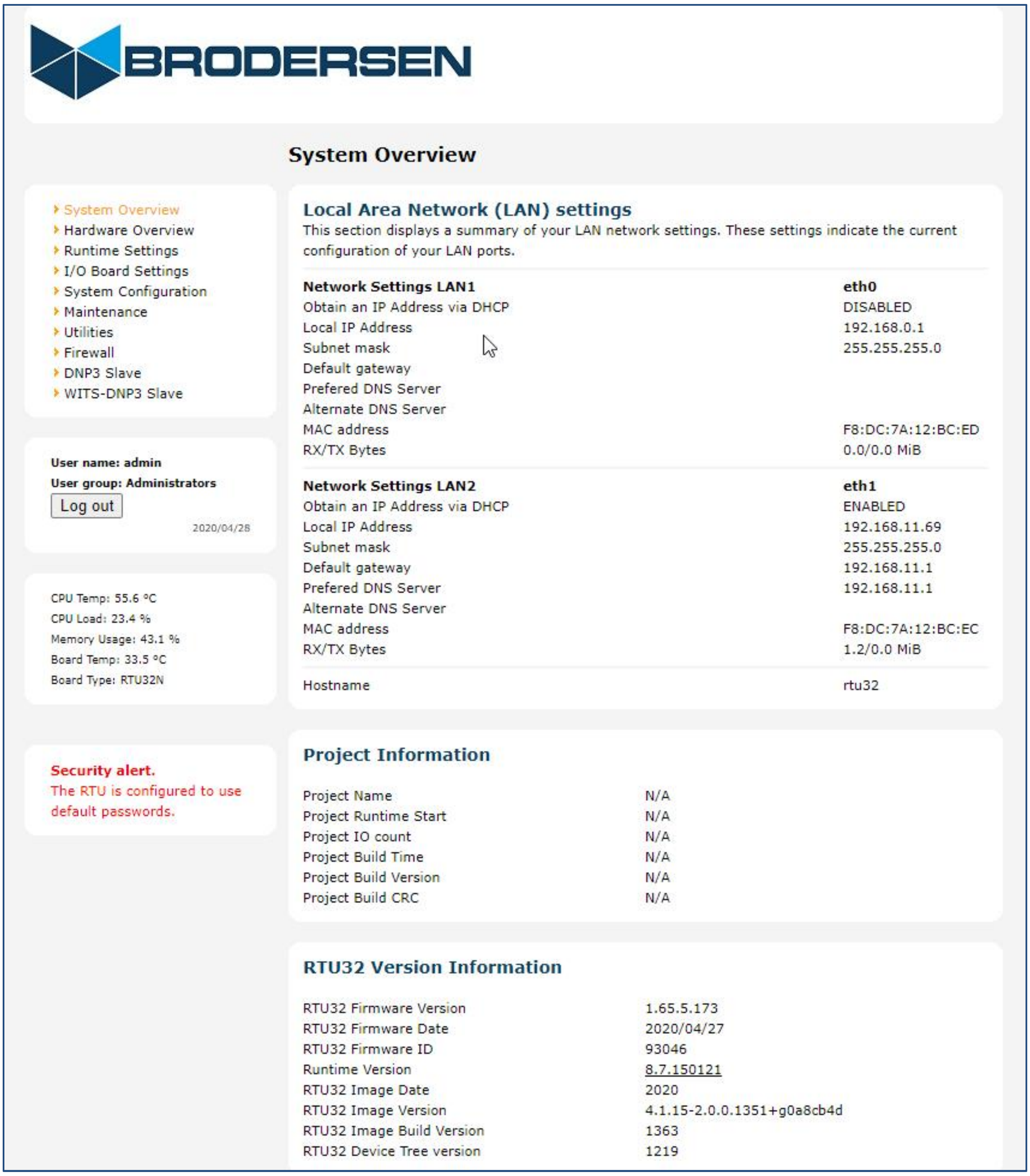
Your connection to this site is not private

Username	admin
Password	•••••

Before the first RTU web page will appear in your browser you will be asked to enter a username and password. The username is admin. The RTU32N is shipped with the password set to admin.

Note: Additional users/access levels can be created.

The RTU will then display the System Overview page. Information shown includes RTU status, network settings, project and version information. Menus allow access to additional RTU information including configuration parameters and status.



User name: admin
User group: Administrators

2020/04/28

CPU Temp: 55.6 °C
CPU Load: 23.4 %
Memory Usage: 43.1 %
Board Temp: 33.5 °C
Board Type: RTU32N

Security alert.
The RTU is configured to use default passwords.

Local Area Network (LAN) settings

This section displays a summary of your LAN network settings. These settings indicate the current configuration of your LAN ports.

Network Settings LAN1	**eth0**
Obtain an IP Address via DHCP	DISABLED
Local IP Address	192.168.0.1
Subnet mask	255.255.255.0
Default gateway	
Prefered DNS Server	
Alternate DNS Server	
MAC address	F8:DC:7A:12:BC:ED
RX/TX Bytes	0.0/0.0 MiB
Network Settings LAN2	**eth1**
Obtain an IP Address via DHCP	ENABLED
Local IP Address	192.168.11.69
Subnet mask	255.255.255.0
Default gateway	192.168.11.1
Prefered DNS Server	192.168.11.1
Alternate DNS Server	
MAC address	F8:DC:7A:12:BC:EC
RX/TX Bytes	1.2/0.0 MiB
Hostname	rtu32

Project Information

Project Name	N/A
Project Runtime Start	N/A
Project IO count	N/A
Project Build Time	N/A
Project Build Version	N/A
Project Build CRC	N/A

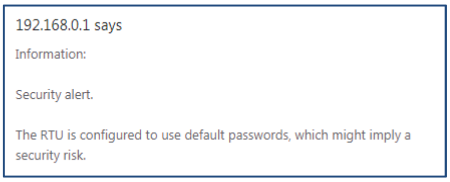
RTU32 Version Information

RTU32 Firmware Version	1.65.5.173
RTU32 Firmware Date	2020/04/27
RTU32 Firmware ID	93046
Runtime Version	8.7.150121
RTU32 Image Date	2020
RTU32 Image Version	4.1.15-2.0.0.1351+g0a8cb4d
RTU32 Image Build Version	1363
RTU32 Device Tree version	1219

5 Managing Secure Access to the RTU32N

5.1 Changing the default admin user password

After logging in for the first time you are warned to change the default password. This is done from the Maintenance >Password menu item (the RTU will continue to prompt with a security alert if the password is not changed from the default, as shown here).



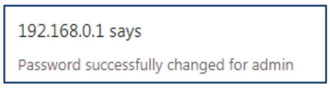
192.168.0.1 says

Information:

Security alert.

The RTU is configured to use default passwords, which might imply a security risk.

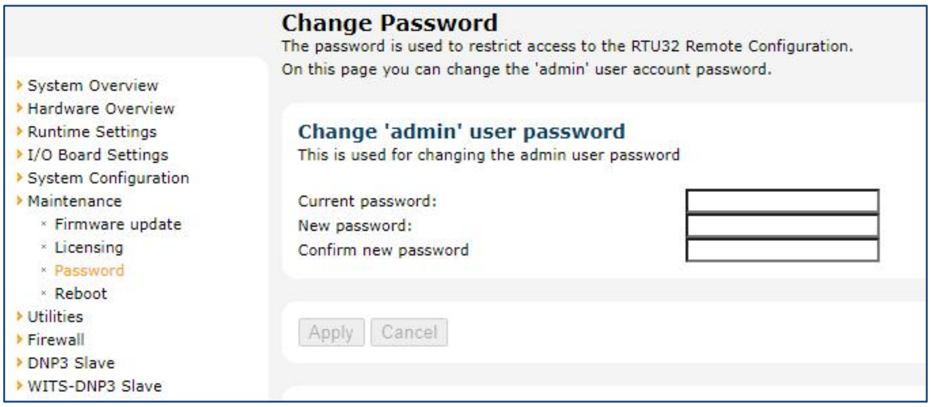
To change the admin user password from the default setting, you must first enter the existing password then enter and confirm the new password.



192.168.0.1 says

Password successfully changed for admin

You will then be prompted to log in to the RTU again using the new password.



Change Password

The password is used to restrict access to the RTU32 Remote Configuration.
On this page you can change the 'admin' user account password.

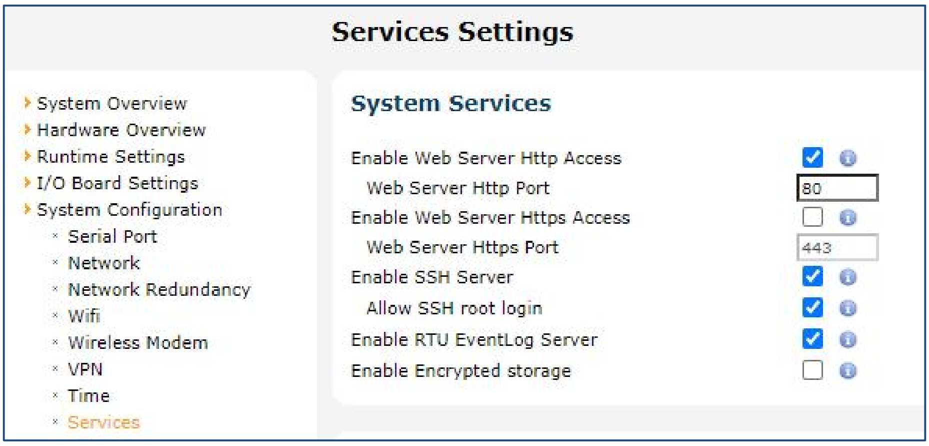
Change 'admin' user password

This is used for changing the admin user password

Current password:
New password:
Confirm new password

5.2 RTU System Services

The RTU System Services are managed from the System Configuration>Services menu.



Services Settings

System Services

Enable Web Server Http Access	☑
Web Server Http Port	80
Enable Web Server Https Access	☐
Web Server Https Port	443
Enable SSH Server	☑
Allow SSH root login	☑
Enable RTU EventLog Server	☑
Enable Encrypted storage	☐

5.2.1 Web Server

Enable/disable access and setting of port addresses for the web server (HTTP and HTTPS).

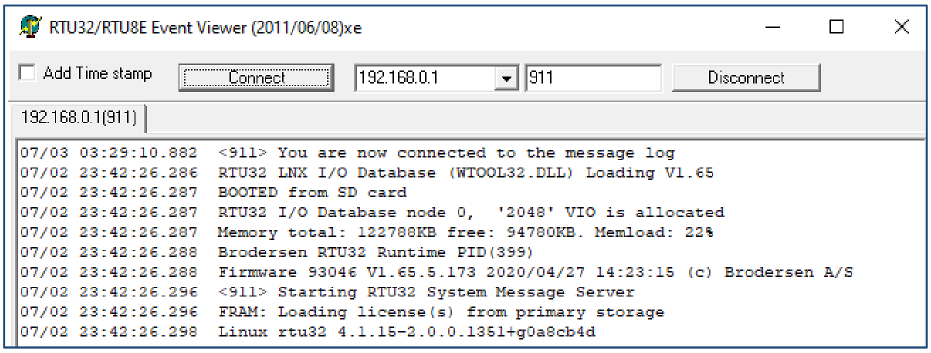
5.2.2 SSH Server

Enable the SSH Server to allow Secure Shell remote access to command line scripts from a SSH client application. Enable SSH root login if required.

5.2.3 RTU Event Log Server

If the RTU EventLog Server is enabled, then RTU events can be viewed from the RTU32 Event Viewer program (installs with WorkSuite).

Note: Port 911 will need to be open in the RTU Firewall Settings for the Event Viewer to be able to connect.



RTU32/RTU8E Event Viewer (2011/06/08)xe

☐ Add Time stamp　192.168.0.1　911

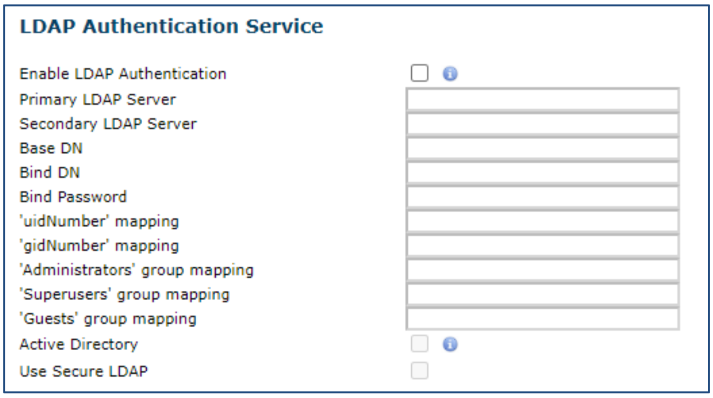
```
07/03 03:29:10.882  <911> You are now connected to the message log
07/02 23:42:26.286  RTU32 LNX I/O Database (WTOOL32.DLL) Loading V1.65
07/02 23:42:26.287  BOOTED from SD card
07/02 23:42:26.287  RTU32 I/O Database node 0,  '2048' VIO is allocated
07/02 23:42:26.287  Memory total: 122788KB free: 94780KB. Memload: 22%
07/02 23:42:26.288  Brodersen RTU32 Runtime PID(399)
07/02 23:42:26.288  Firmware 93046 V1.65.5.173 2020/04/27 14:23:15 (c) Brodersen A/S
07/02 23:42:26.296  <911> Starting RTU32 System Message Server
07/02 23:42:26.296  FRAM: Loading license(s) from primary storage
07/02 23:42:26.298  Linux rtu32 4.1.15-2.0.0.1351+g0a8cb4d
```

5.2.4 Encrypted Storage

The RTU32N can be set to boot/load firmware and store logic applications on an optional SD card. If encrypted storage mode is enabled, the RTU will keep configuration information in an encrypted ‘container’ area of the SD card. Note: once enabled, it is only disabled by a factory reset. The encrypted container can only be read by the RTU that created the container (this protects against the card being removed and sensitive configuration data being read by a third party).

5.2.5 LDAP – Lightweight Directory Access Protocol

When LDAP Authentication is enabled;



LDAP Authentication Service

Enable LDAP Authentication □
Primary LDAP Server
Secondary LDAP Server
Base DN
Bind DN
Bind Password
'uidNumber' mapping
'gidNumber' mapping
'Administrators' group mapping
'Superusers' group mapping
'Guests' group mapping
Active Directory □
Use Secure LDAP □

- Root and Admin users are still able to login using SSH, but only authenticated LDAP users can login to the RTU webserver.
- If the Primary LDAP Server can't be contacted then the Secondary LDAP Server is used instead (if no LDAP server is available, web access will be denied).
- Base DN specifies the base distinguished name (DN) to use as search base.
- Bind DN is a user in the Active directory, which is allowed to query the LDAP Server.
- Bind Password is the password for the user in 'Bind DN'.
- 'uidNumber' mapping is used for mapping the user number from the AD. This information can be found on the Domain controller as described below.
- 'gidNumber' mapping is used for mapping the group number from the AD. This information can be found on the Domain controller as described below.
- Administrators mapping is used for mapping a group in the AD to the Webserver's Administrators group eg. users in the AD 'Administrators' group mapping will have Administrator access to the RTU.
- Superusers mapping is used for mapping a group in the AD to the Webserver's Superusers group eg. users in the AD 'Superusers' group mapping will have Superuser access to the RTU.
- Guests mapping is used for mapping a group in the AD to the Webserver's Guests group eg. users in the AD 'Guests' group mapping will have Guest access to the RTU.

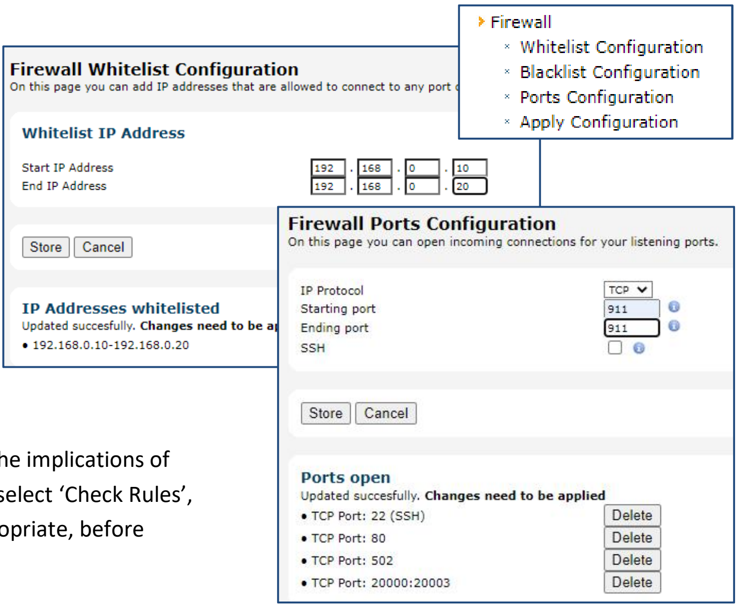
5.3 RTU Firewall Setup

The RTU firewall is a set of rules that determine if access is granted to incoming IP message packets.

The **Firewall>Whitelist Configuration** page allows specific IP addresses to access all/any ports on the RTU.

The **Firewall>Blacklist Configuration** page allows specific IP addresses to be blocked from access to all RTU ports.

The **Firewall>Ports Configuration** page allows management of the range of IP ports to be opened.



Firewall Whitelist Configuration
On this page you can add IP addresses that are allowed to connect to any port o

Whitelist IP Address

Start IP Address 192 . 168 . 0 . 10
End IP Address 192 . 168 . 0 . 20

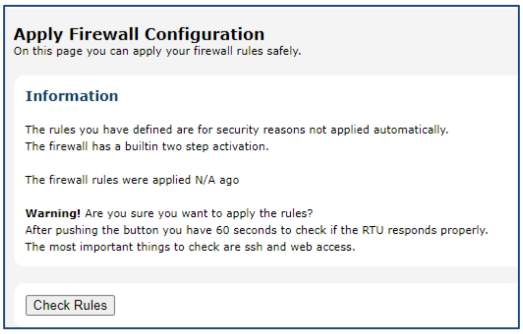
IP Addresses whitelisted
Updated succesfully. **Changes need to be ap**
- 192.168.0.10-192.168.0.20

Firewall Ports Configuration
On this page you can open incoming connections for your listening ports.

IP Protocol TCP
Starting port 911
Ending port 911
SSH □

Ports open
Updated succesfully. **Changes need to be applied**
- TCP Port: 22 (SSH)
- TCP Port: 80
- TCP Port: 502
- TCP Port: 20000:20003

The **Firewall>Apply Configuration** page allows time to test the implications of changing the firewall rules. This is a two step process - first select 'Check Rules', you then have 60 seconds to check if the new rules are appropriate, before applying the new rules.



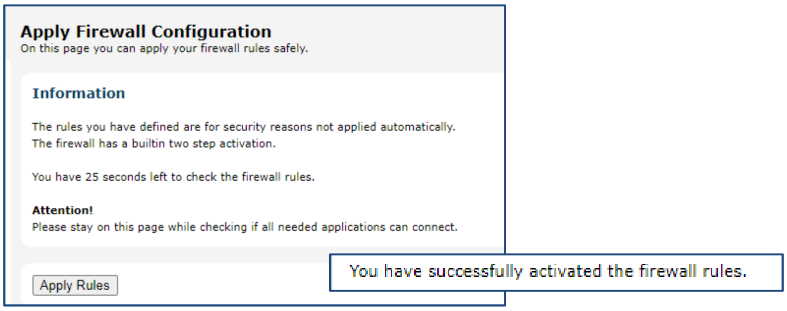
Apply Firewall Configuration
On this page you can apply your firewall rules safely.

Information

The rules you have defined are for security reasons not applied automatically.
The firewall has a builtin two step activation.

The firewall rules were applied N/A ago

Warning! Are you sure you want to apply the rules?
After pushing the button you have 60 seconds to check if the RTU responds properly.
The most important things to check are ssh and web access.



Apply Firewall Configuration
On this page you can apply your firewall rules safely.

Information

The rules you have defined are for security reasons not applied automatically.
The firewall has a builtin two step activation.

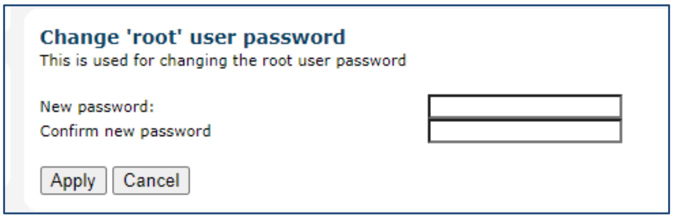
You have 25 seconds left to check the firewall rules.

Attention!
Please stay on this page while checking if all needed applications can connect.

You have successfully activated the firewall rules.

5.4 Root user access

The RTU32N allows access to the RTU operating system via a SSH client using the standard RTU passwords used for web access. Root level access is enabled in the System Configuration> Services menu and should only be enabled for advanced users that need to perform diagnostics. If root access is required a password must be created from the Maintenance >Password menu.



Change 'root' user password

This is used for changing the root user password

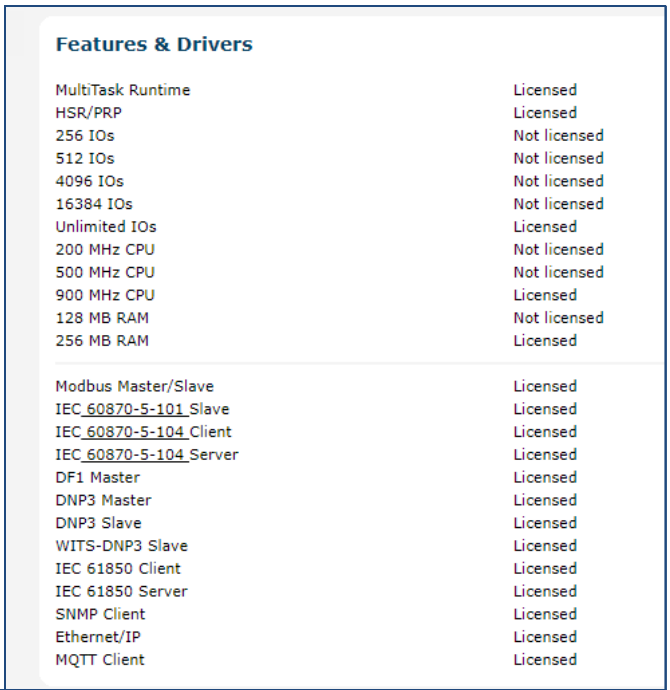
New password:

Confirm new password

6 RTU32N – Licensing

The RTU32N includes license options that manage the RTU functionality and enable use of advanced drivers and options, hence some restrictions may be imposed, including limiting the number of project I/O variables, drivers and driver tags/data points. The System Overview page includes information about the licensing, as shown in the example below.

The license limitations are managed by the firmware of the RTU32N. If you try to start a logic application in the RTU32N that exceeds these limits, the project will NOT start and an info message in the Runtime info window in WorkSuite will indicate the status of the limits in your project.



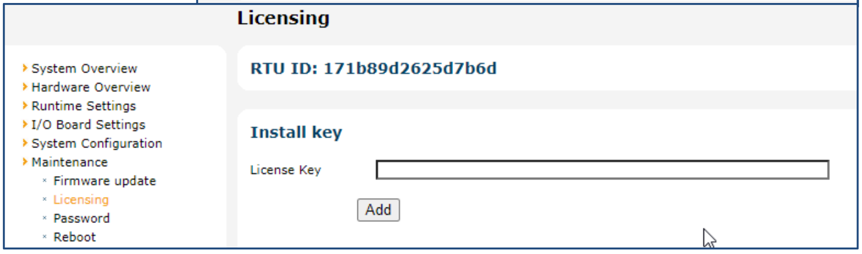
Features & Drivers

MultiTask Runtime	Licensed
HSR/PRP	Licensed
256 IOs	Not licensed
512 IOs	Not licensed
4096 IOs	Not licensed
16384 IOs	Not licensed
Unlimited IOs	Licensed
200 MHz CPU	Not licensed
500 MHz CPU	Not licensed
900 MHz CPU	Licensed
128 MB RAM	Not licensed
256 MB RAM	Licensed
Modbus Master/Slave	Licensed
IEC 60870-5-101 Slave	Licensed
IEC 60870-5-104 Client	Licensed
IEC 60870-5-104 Server	Licensed
DF1 Master	Licensed
DNP3 Master	Licensed
DNP3 Slave	Licensed
WITS-DNP3 Slave	Licensed
IEC 61850 Client	Licensed
IEC 61850 Server	Licensed
SNMP Client	Licensed
Ethernet/IP	Licensed
MQTT Client	Licensed

6.1 Adding Drivers and Runtime Options

The RTU32N Licensing interface allows loading and activation of runtime license options. The RTU ID is used by the Brodersen factory to generate a unique license code that enables the required functionality.

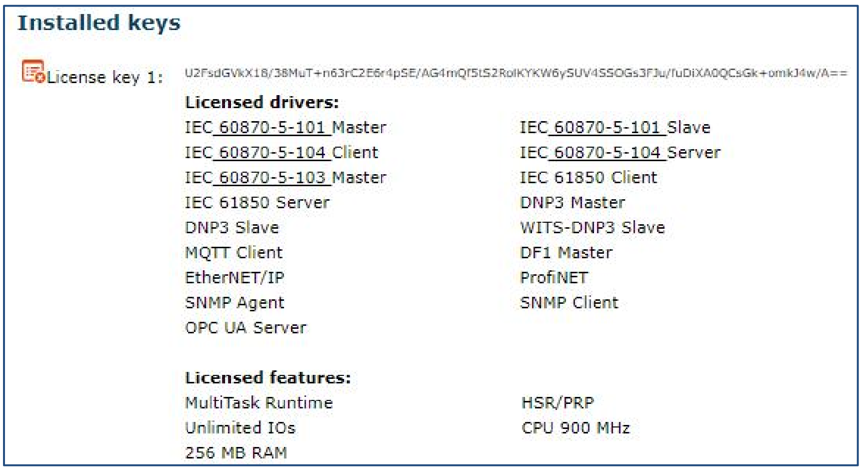
If you need additional drivers or other increased functionality that requires a license upgrade, please contact Brodersen or your authorized distributor and provide them with the RTU ID and details of the license requirements. You will then be sent a new license key that can be pasted in to the License Key field and then select 'Apply' to activate the new license options.



RTU ID: 171b89d2625d7b6d

License Key

The example here shows a RTU licensed with many driver and functionality options.



Installed keys

License key 1: U2FsdGVkX18/38MuT+n63rC2E6r4pSE/AG4mQf5tS2RolKYKW6ySUV4SSOGs3FJu/fuDiXA0QCsGk+omkJ4w/A==

Licensed drivers:

IEC 60870-5-101 Master	IEC 60870-5-101 Slave
IEC 60870-5-104 Client	IEC 60870-5-104 Server
IEC 60870-5-103 Master	IEC 61850 Client
IEC 61850 Server	DNP3 Master
DNP3 Slave	WITS-DNP3 Slave
MQTT Client	DF1 Master
EtherNET/IP	ProfiNET
SNMP Agent	SNMP Client
OPC UA Server	

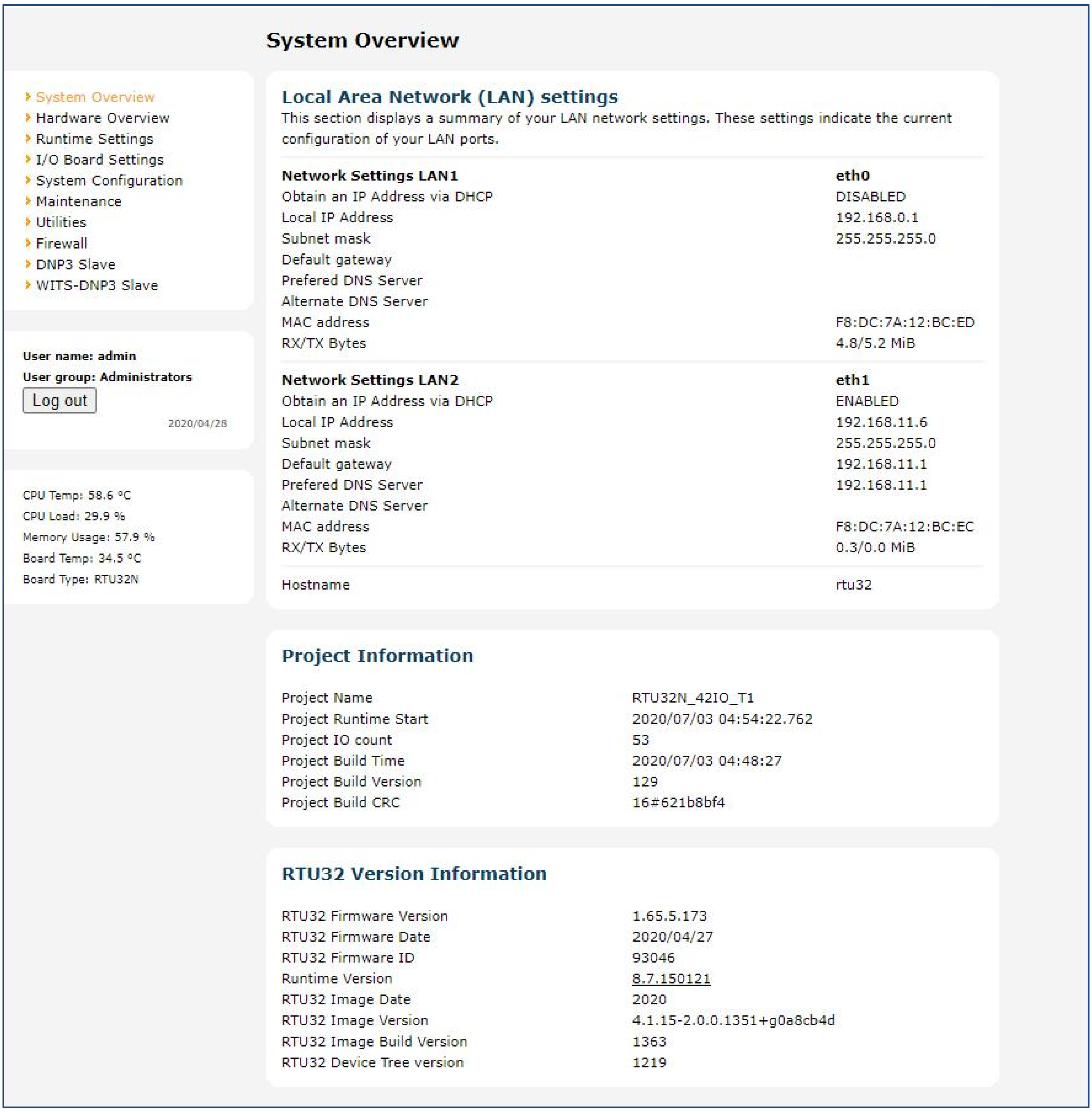
Licensed features:

MultiTask Runtime	HSR/PRP
Unlimited IOs	CPU 900 MHz
256 MB RAM	

7 Getting Online with your RTU32N

7.1 System Overview Web Page

The first page that you access when you enter the RTU32N Web pages is the System Overview page:



System Overview

User name: admin
User group: Administrators

2020/04/28

CPU Temp: 58.6 °C
CPU Load: 29.9 %
Memory Usage: 57.9 %
Board Temp: 34.5 °C
Board Type: RTU32N

Local Area Network (LAN) settings

This section displays a summary of your LAN network settings. These settings indicate the current configuration of your LAN ports.

Network Settings LAN1	**eth0**
Obtain an IP Address via DHCP	DISABLED
Local IP Address	192.168.0.1
Subnet mask	255.255.255.0
Default gateway	
Prefered DNS Server	
Alternate DNS Server	
MAC address	F8:DC:7A:12:BC:ED
RX/TX Bytes	4.8/5.2 MiB

Network Settings LAN2	**eth1**
Obtain an IP Address via DHCP	ENABLED
Local IP Address	192.168.11.6
Subnet mask	255.255.255.0
Default gateway	192.168.11.1
Prefered DNS Server	192.168.11.1
Alternate DNS Server	
MAC address	F8:DC:7A:12:BC:EC
RX/TX Bytes	0.3/0.0 MiB
Hostname	rtu32

Project Information

Project Name	RTU32N_42IO_T1
Project Runtime Start	2020/07/03 04:54:22.762
Project IO count	53
Project Build Time	2020/07/03 04:48:27
Project Build Version	129
Project Build CRC	16#621b8bf4

RTU32 Version Information

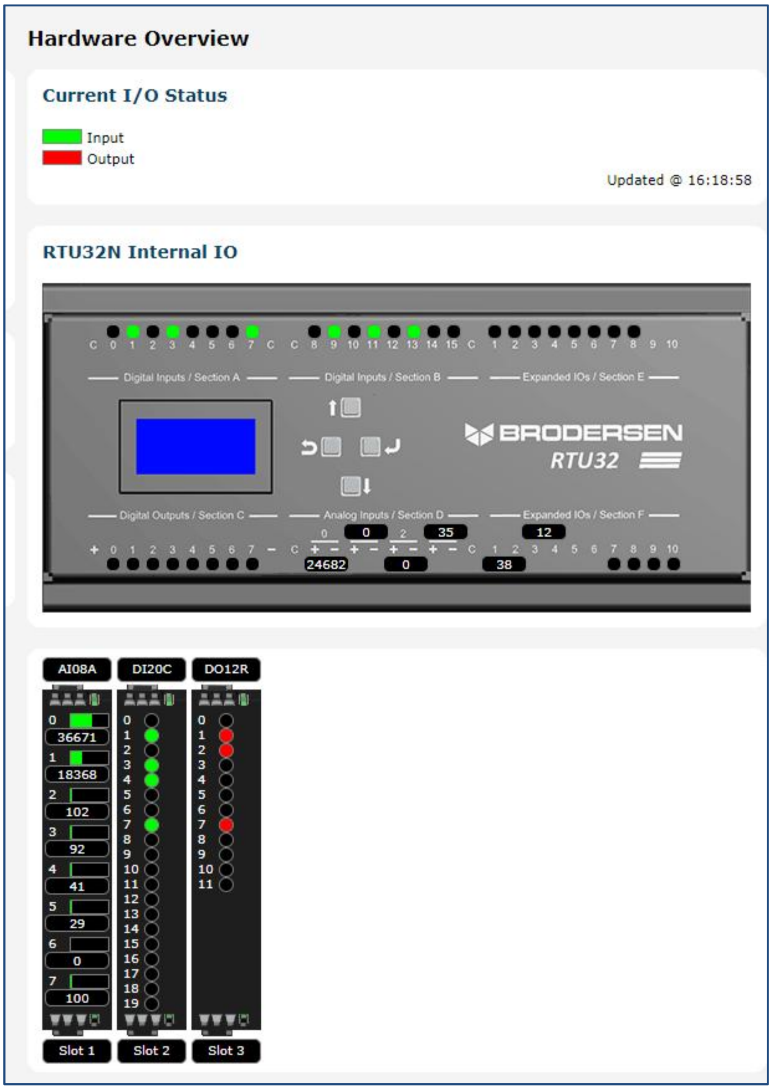
RTU32 Firmware Version	1.65.5.173
RTU32 Firmware Date	2020/04/27
RTU32 Firmware ID	93046
Runtime Version	8.7.150121
RTU32 Image Date	2020
RTU32 Image Version	4.1.15-2.0.0.1351+g0a8cb4d
RTU32 Image Build Version	1363
RTU32 Device Tree version	1219

On this page you will get an overview of the network settings, project application that is running in the RTU, I/O count, the RTU status and version information. If you are contacting your support office or distributor, you will always be asked for the RTU32N Firmware Version Information eg. 1.65.5.173 and the RTU Image Build Version eg. 1363.

7.2 Hardware Overview Web Page

The hardware overview page shows the current I/O configuration status. It lists the physical internal I/O and any RTU32M style expansion I/O connected via the LB2 bus. You are able to read the status of each input or output. In the example shown here, the RTU32N is connected with 3x LB2 expansion I/O modules. The status of the various analog and digital values can be seen.

Note: an appropriate logic application must be loaded/running for the hardware overview to show status values.



Hardware Overview

Current I/O Status

🟩 Input
🟥 Output

Updated @ 16:18:58

RTU32N Internal IO

7.3 Runtime Settings / Main Settings

The Runtime settings menu allows adjustment of the runtime (logic 'engine') setup.

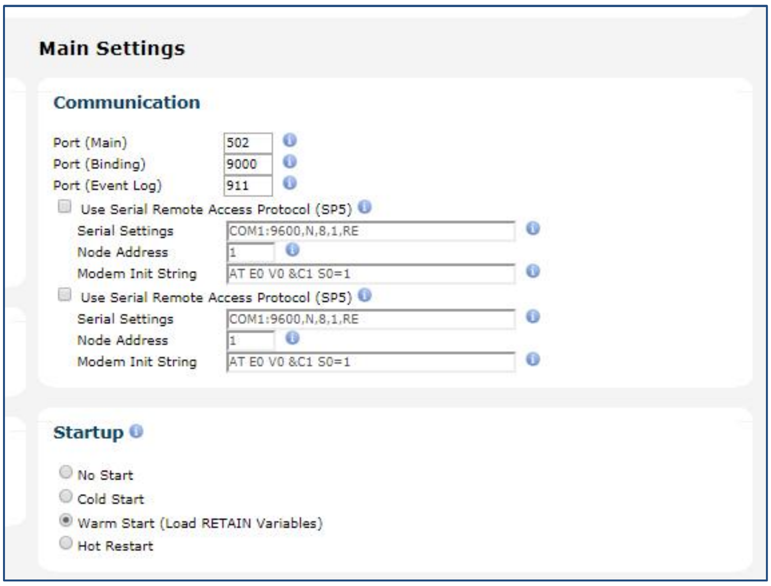
7.3.1 Main Settings / Communication

Configuration of start-up settings, communication ports and enabling of Remote Access Protocols.

Port (Main) - is used for Brodersen WorkSuite communications. Default is 502 - this port is also used for the ModbusTCP Server.

Port (Binding) - is used by the event based binding protocol, where two or more RTU32s can exchange data (redundancy mode). Default is 9000.

Port (Event log) - is for access to RTU System Logs, either by connecting using telnet or the Brodersen Event Viewer tool. Default is 911.



Main Settings

Communication

Port (Main) 502
Port (Binding) 9000
Port (Event Log) 911
□ Use Serial Remote Access Protocol (SP5)
Serial Settings COM1:9600,N,8,1,RE
Node Address 1
Modem Init String AT E0 V0 &C1 S0=1
□ Use Serial Remote Access Protocol (SP5)
Serial Settings COM1:9600,N,8,1,RE
Node Address 1
Modem Init String AT E0 V0 &C1 S0=1

Startup

○ No Start
○ Cold Start
● Warm Start (Load RETAIN Variables)
○ Hot Restart

Enable Serial Remote Access Protocol (SP5)
This protocol allows remote serial communication from the Brodersen WorkSuite and is well suited to communication using serial radios and modems. e.g. if you have the RTU32N placed on a remote site connected via serial communication using multi-drop radio or line connection – or even a dialup connection, you are able to remotely connect to the RTU32N with the Brodersen WorkSuite and work online as if it was on your workshop table. The speed will however be slower depending on your actual serial communication speed. It is possible to attach this protocol to two communication ports.

Serial settings
The syntax is: [M]COM{n}:{b},{p},{d},{s}[,h][,LL] and the parameters in brackets [ ] are optional:
[M] Use modem connection
{n} Number of the COM port to be used (1 … 8)
{b} Allowed speeds: 300,600,1200,2400,4800,9600,19200,38400,57600,115200
{p} Parity: N,E,O (None,Even,Odd)
{d} Data bits: 7 or 8
{s} Stop bits: 1 or 2
[h] RTS/CTS handshaking where h is either RD,RE,RT:l:t or RC:l:t. In the COM port section you will find details of how to use these parameters.
[LL] Log all bytes received and transmitted into the system log (port 911)

The following example shows setting for communicating on COM1 port, at 19200 baud, with Even parity, 8 data bits and 1 stop bit: COM1:19200,E,8,1
Here a modem usage is enabled: MCOM2:9600,N,8,1
Here logging is enabled: COM1:9600,N,8,1,LL

Node address
Node Address (or Unit number) identifies the RTU32 on a multi drop network. The default number for RTU32N on a peer to peer network is 1. The address must be a value from 1 to 32767.

Modem Init String
Here you enter the initialization string (AT commands) when using a modem.
RTU32N default is: AT E0 V0 &C1 S0=1

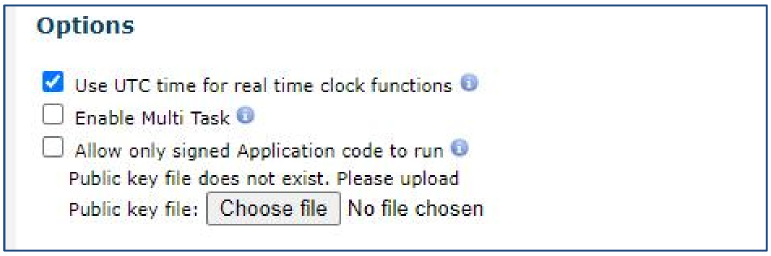
7.3.2 Main Settings / Start-up

The start-up option specifies how the logic application is started when the RTU32N is powered up. The possible options are:

No Start – logic application is NOT started when booting the RTU32.
Cold Start – logic application is started without loading retained variables.
Warm Start – logic application is started and retained variables are loaded (default) **.
Hot Restart – Only supported with battery backed RAM option and in Redundancy configuration!

** Note: RTU32N has 4kB NV FRAM for retained variables (eg. 1000x 32bit values). If a larger amount of variables need to be retained, they can be managed using a SD card with the F_SAVERETAIN and F_LOADRETAIN function blocks.
Eg. Boot from SD card: 'etc/rtu32/retain.bin'. Boot from Internal flash using SD card: '/run/media/mmcblk0p1/retain.bin'.

7.3.3 Main Settings / Options


Options

☑ Use UTC time for real time clock functions
☐ Enable Multi Task
☐ Allow only signed Application code to run
Public key file does not exist. Please upload
Public key file: Choose file No file chosen

Use UTC time for real time clock functions
If enabled, it instructs all logic real time functions (e.g. DTCurDate, DTCurTime and day_time) to return time in UTC. UTC time is not adjusted according to daylight savings.

If this parameter is disabled then time is returned in local time.

Note: Drivers that are implemented via PLC Functions – like the IEC60870 Driver – will use the logic time stamp for events. Other drivers may take standard windows time or whatever time is defined in the applicable standards eg. DNP3 always uses UTC time.

Enable Multi Task
This is a licensed option allowing the RTU to run up to 4x independent logic tasks (global variables can be accessed from any/all tasks).

7.3.4 Signed Application Code

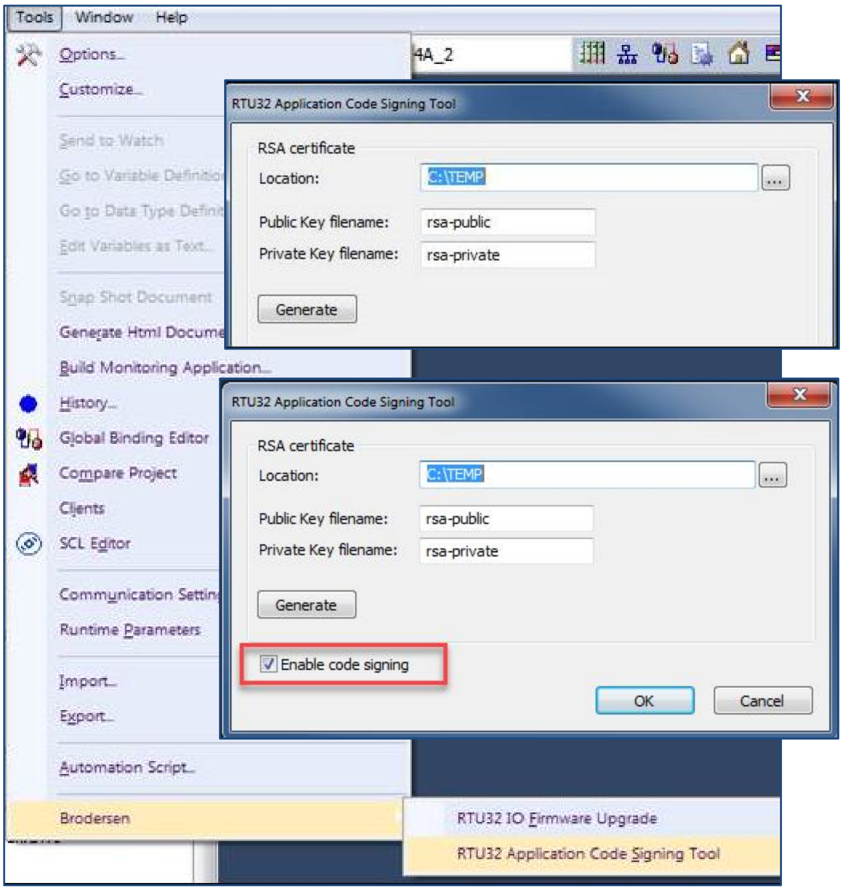
If 'Allow only signed Application code to run' is enabled then an additional layer of security is added to logic applications and RTU firmware updates (SWU files) using RSA public-key cryptosytem techniques. WorkSuite generates a public key (shared by WorkSuite to encrypt authorisation and the RTU to decrypt authorisation) and a private key (used by WorkSuite to encrypt authorisation).

Keys are generated from the WorkSuite Tools menu. Select Brodersen>RTU32 Application Code Signing Tool, set the location where keys are to be stored and select Generate.

eg. if using C: \Temp then

- A public key **rsa-public.pem** will be generated and stored in C:\Temp
- A private key **rsa-private.pem** will be generated and stored in C:\Temp

If the ' Enable code signing' check box is selected, then future compilations of RTU application logic will include an encrypted signature.



RTU32 Application Code Signing Tool

RSA certificate

Location: C:\TEMP

Public Key filename: rsa-public

Private Key filename: rsa-private

Generate

RTU32 Application Code Signing Tool

RSA certificate

Location: C:\TEMP

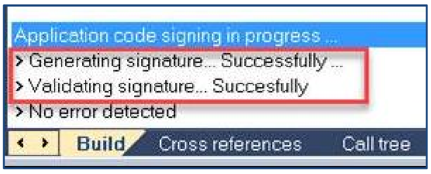
Public Key filename: rsa-public

Private Key filename: rsa-private

Generate

☑ Enable code signing

OK Cancel



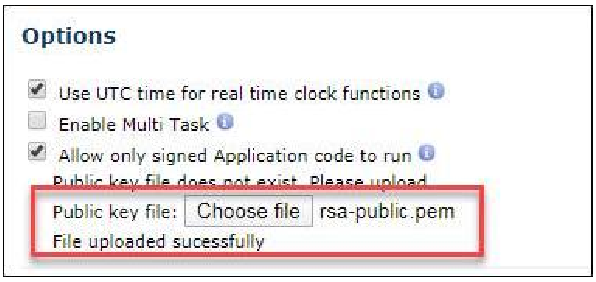
```
Application code signing in progress ...
> Generating signature... Successfully ...
> Validating signature... Succesfully
> No error detected
```

When compiling code in WorkSuite with the code signing feature enabled, the last part of the compilation process shows the status of the signature generation and validation.

To enable the use of signed application code in the RTU, select 'Allow only signed Application code to run', from the web server page Runtime Settings>Main Settings>Options.

You must also then also upload the Public key file generated by WorkSuite to the RTU, by selecting 'Choose file' and then browsing to the location where the key file is stored eg. C:\Temp as used earlier.

The public key file will then be sent to the RTU (as an encrypted packet sent securely using SSH) and stored as rtu32-app-public.pem in /etc/rtu32.



Options

☑ Use UTC time for real time clock functions

☐ Enable Multi Task

☑ Allow only signed Application code to run

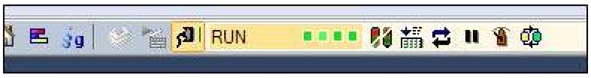
Public key file does not exist. Please upload

Public key file: Choose file rsa-public.pem

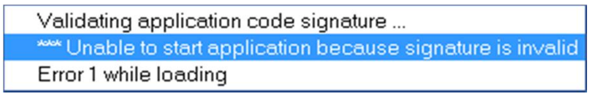
File uploaded sucessfully

Note: important that the 'File uploaded successfully' statement appears. You may need to ensure that your browser is not caching old JavaScript files (use CTRL-F5 to force reload).

Ensure changes are applied by selecting Apply and then reboot the RTU. Download the code to the RTU from WorkSuite and check the RTU is in the RUN state from the Debug status window.



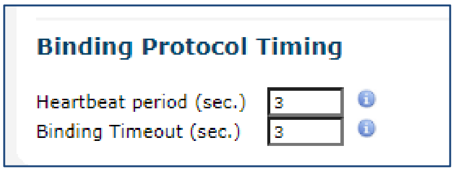
If the RTU will not run, check the Runtime window. If the signature file is missing or invalid, the following error message will be shown.



```
Validating application code signature ...
*** Unable to start application because signature is invalid
Error 1 while loading
```

7.3.5 Main Settings / Binding Protocol Timing

When using binding on slower network connections like some optical fibre modems or GPRS/3G wireless connections, you will need to adjust the timing for the Binding driver.



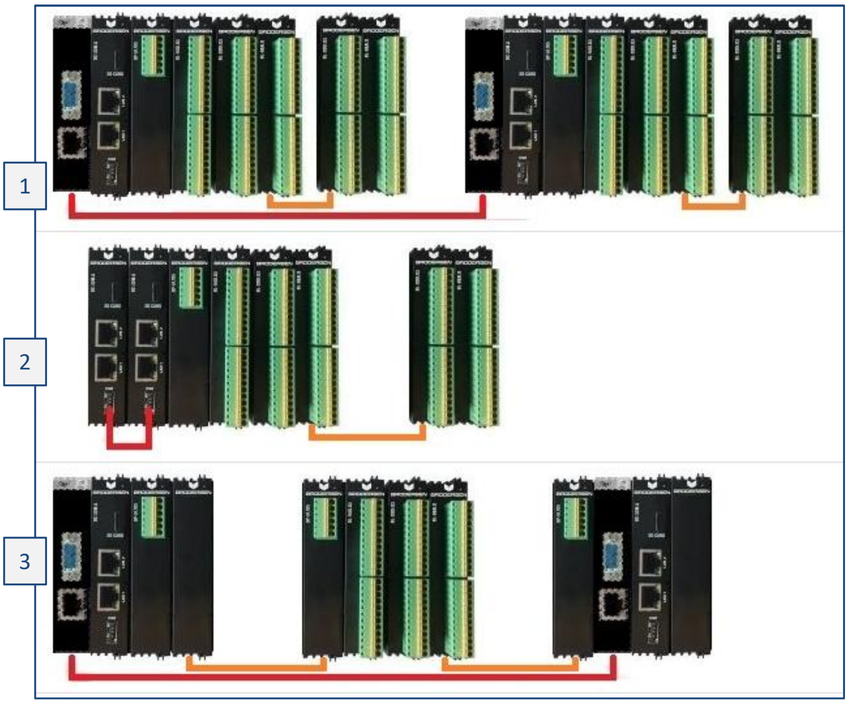
Binding Protocol Timing

Heartbeat period (sec.)	3
Binding Timeout (sec.)	3

Heartbeat period – how often the binding link connection is checked.
Binding Time-out – time-out on link check and general binding events.

7.4 Runtime / Redundancy Settings

The RTU32M/N supports three different redundancy modes. The modes are managed from the RTU web server and include;



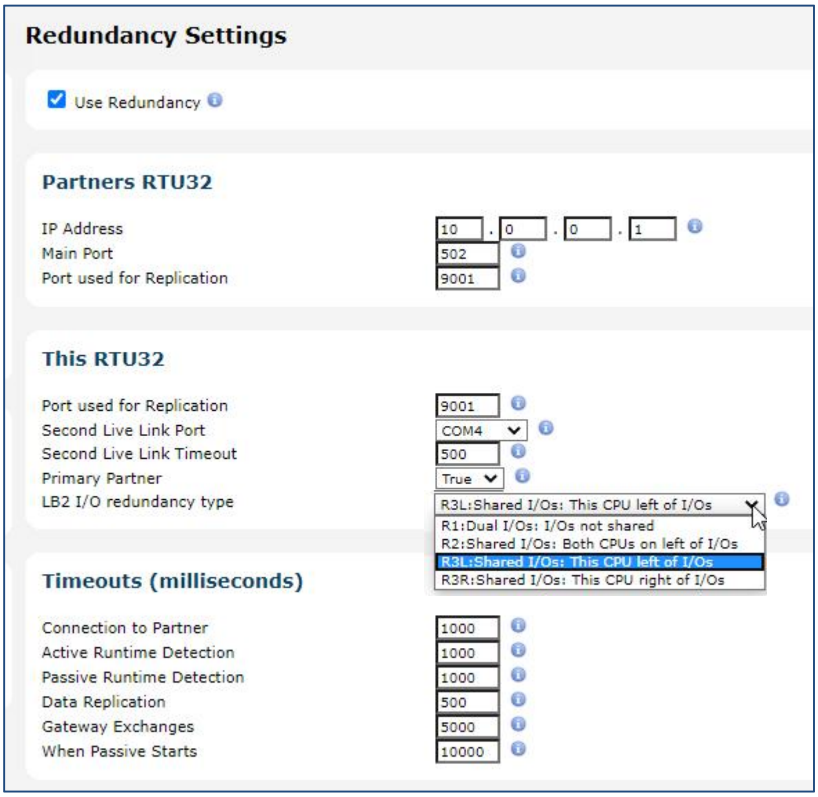
1. Dual I/Os - 2x RTUs with their own I/O. RTUs are connected together using CM02A modules (each module has 1x serial port and 1x Ethernet port).

2. Shared I/Os - 2x CPU on left side of I/O. CPUs are connected together using a REDLINK device cable (provides a dedicated serial and Ethernet connection using a special USB-USB cable). Note: this mode is not possible using RTU32N hardware.

3. Shared I/Os - A CPU on each side of the I/O. CPUs are connected together using CM02A modules (each module has 1x serial port and 1x Ethernet port).

Redundancy Operation Principles – the redundancy option allows two RTUs/CPUs to run as partners, where one is active and the other is passive. The active CPU keeps the passive updated (with both logic and real time data), so that if the active CPU fails, the passive CPU switches to active mode and 'hot restarts'.

Changeover typically occurs in less than 500 milli-seconds once failure is detected.



Redundancy Settings

☑ Use Redundancy

Partners RTU32

IP Address	10 . 0 . 0 . 1
Main Port	502
Port used for Replication	9001

This RTU32

Port used for Replication	9001
Second Live Link Port	COM4
Second Live Link Timeout	500
Primary Partner	True
LB2 I/O redundancy type	R3L:Shared I/Os: This CPU left of I/Os

R1:Dual I/Os: I/Os not shared
R2:Shared I/Os: Both CPUs on left of I/Os
R3L:Shared I/Os: This CPU left of I/Os
R3R:Shared I/Os: This CPU right of I/Os

Timeouts (milliseconds)

Connection to Partner	1000
Active Runtime Detection	1000
Passive Runtime Detection	1000
Data Replication	500
Gateway Exchanges	5000
When Passive Starts	10000

7.4.1 Redundancy / Partners RTU32

Each CPU needs to know the IP address and port settings of its 'partner' RTU32.

7.4.2 Redundancy / This RTU32

Each CPU needs to know if it is the Primary or Standby partner (left CPU is primary by default) and which redundancy mode it is using.

7.4.3 Redundancy Timeouts

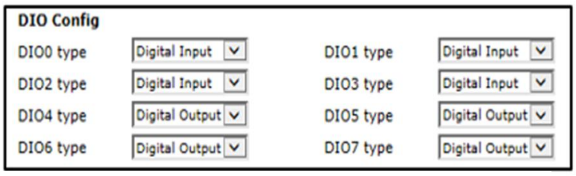
The timeout settings define various connection and heartbeat periods - default settings are recommended.

8 RTU32N Onboard I/O (28 and 42 I/O)

The RTU32N is available without onboard I/O and with two different types of onboard I/O. combinations:

28 IO 16 Digital Input, 8 PNP Output,
4AI (connectors E and F not used)

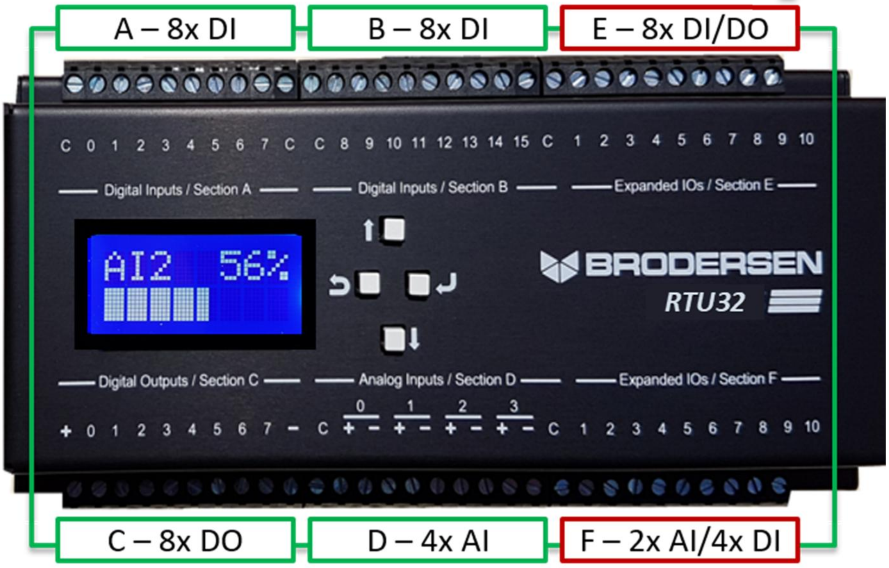
42 IO 16 Digital Input, 8 PNP Output, 4AI,
A 14ch daughter board is added that uses connectors E and F)
+ 8DIO (configurable input or output),
+ 4x Fast DI (use as DI or 2kHz counters).

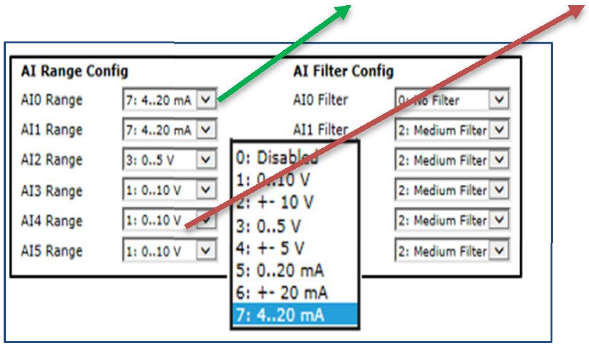


DIO Config

DIO0 type	Digital Input	DIO1 type	Digital Input
DIO2 type	Digital Input	DIO3 type	Digital Input
DIO4 type	Digital Output	DIO5 type	Digital Output
DIO6 type	Digital Output	DIO7 type	Digital Output

8x DI/DO – input or output mode set from web page


A – 8x DI
B – 8x DI
E – 8x DI/DO
C 0 1 2 3 4 5 6 7 C C 8 9 10 11 12 13 14 15 C 1 2 3 4 5 6 7 8 9 10
Digital Inputs / Section A
Digital Inputs / Section B
Expanded IOs / Section E
AI2 56%
BRODERSEN
RTU32
Digital Outputs / Section C
Analog Inputs / Section D
Expanded IOs / Section F
0 1 2 3
+ 0 1 2 3 4 5 6 7 – C + – + – + – + – C 1 2 3 4 5 6 7 8 9 10
C – 8x DO
D – 4x AI
F – 2x AI/4x DI



AI Range Config

AI0 Range	7: 4..20 mA
AI1 Range	7: 4..20 mA
AI2 Range	3: 0..5 V
AI3 Range	1: 0..10 V
AI4 Range	1: 0..10 V
AI5 Range	1: 0..10 V

AI Filter Config

AI0 Filter	0: No Filter
AI1 Filter	2: Medium Filter
	2: Medium Filter
	2: Medium Filter
	2: Medium Filter
	2: Medium Filter

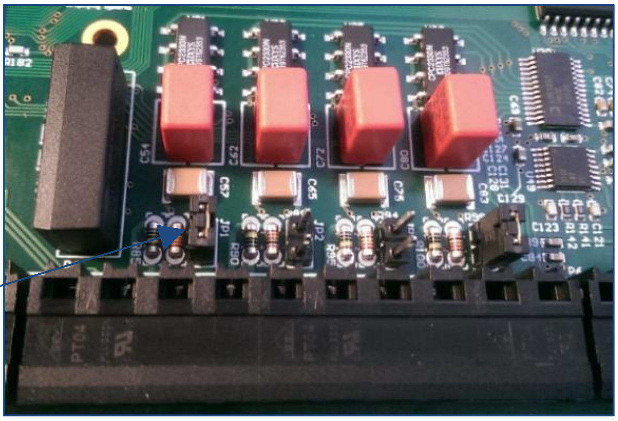
0: Disabled
1: 0..10 V
2: +- 10 V
3: 0..5 V
4: +- 5 V
5: 0..20 mA
6: +- 20 mA
7: 4..20 mA

Analog input ranges and filter values set from web page

- **Digital inputs, 10-30VDC (all DI channels count)**
- **Digital outputs, 10-30VDC (High Side Switched)**
- **Fast DI/counter inputs, 5-30VDC, up to 2 KHz**
- **Analog Inputs (16bit), multi-range:**
  0-5V, +-5V, 0-10V, +-10V
  0-20mA, +-20mA, 4-20mA

8.1 I/O board jumper configurations (.81 and .91 RTU32N variants)

Some I/O modules and expansion modules have user configurable parameters. They allow adjustment of the setup for each configurable module that is managed by the RTU32N configuration, as shown above. Note: the RTU32N analog input mode can be set to voltage or current for each channel via, on board jumpers (jumper must be ON for current). RTU32N controllers supplied with a part number ending with .81 have the analog input jumpers off (all set for voltage). The jumpers are accessible by removing the RTU32N top cover/lid.



Current/Voltage mode selection jumpers

8.2 RTU32N I/O Board Settings

On this web page page you can:
- Read the actual I/O configuration on the RTU32N.
- Lock an I/O configuration (allows detection of missing modules that are 'expected' on the legacy IO bus).
- Adjust the settings for configurable inputs and outputs – if any exist in the current RTU configuration.

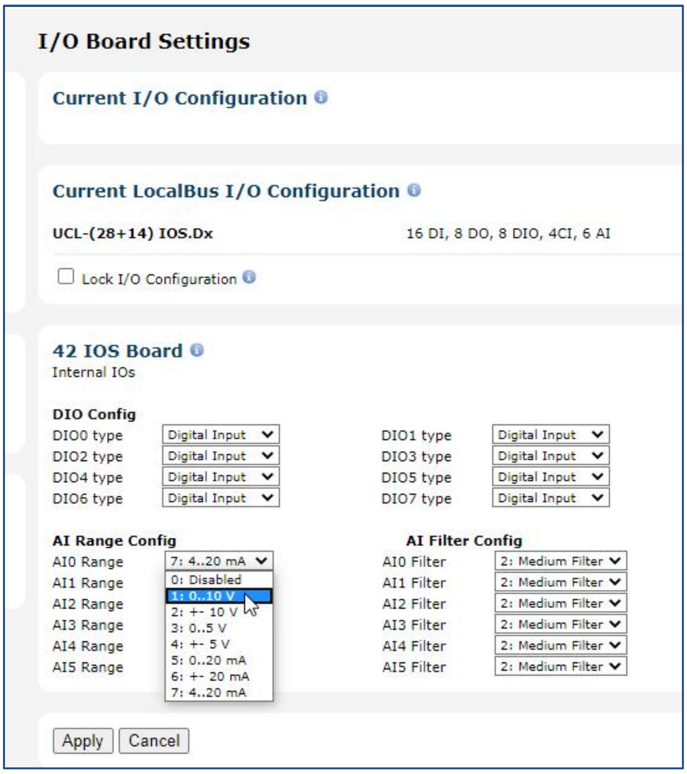
Here the actual I/O configurations are listed.

In the example below you can see that the RTU32N has the 14ch daughter board included, giving 42IO.

If additional expansion modules were connected, they would also be listed here.

By default the I/O configuration is unlocked. When unlocked the actual I/O configuration is read every time the RTU32N is booted and when the PLC Runtime application is started ie. I/O is then 'self discovering'.

By enabling lock I/O configuration, the actual I/O configuration is locked and stored, and NOT read at boot and/or application start-up. Use the Lock function if you want to be able to trace a disconnected or defective I/O module in the RTU32 System Log.



I/O Board Settings

Current I/O Configuration

Current LocalBus I/O Configuration

UCL-(28+14) IOS.Dx 16 DI, 8 DO, 8 DIO, 4CI, 6 AI

☐ Lock I/O Configuration

42 IOS Board
Internal IOs

DIO Config

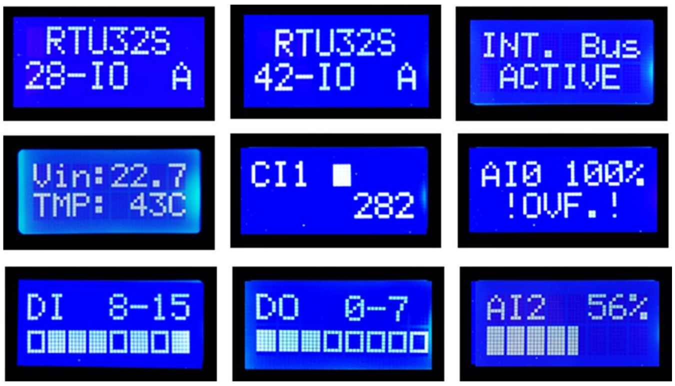
DIO0 type	Digital Input	DIO1 type	Digital Input
DIO2 type	Digital Input	DIO3 type	Digital Input
DIO4 type	Digital Input	DIO5 type	Digital Input
DIO6 type	Digital Input	DIO7 type	Digital Input

AI Range Config		AI Filter Config	
AI0 Range	7: 4..20 mA	AI0 Filter	2: Medium Filter
AI1 Range		AI1 Filter	2: Medium Filter
AI2 Range		AI2 Filter	2: Medium Filter
AI3 Range		AI3 Filter	2: Medium Filter
AI4 Range		AI4 Filter	2: Medium Filter
AI5 Range		AI5 Filter	2: Medium Filter

8.2.1 RTU32N Local Display

The local display is for showing the I/O and RTU status. The display turns off automatically after one minute, if idle. Press any button on the front of RTU and the LCD backlight turns on again.

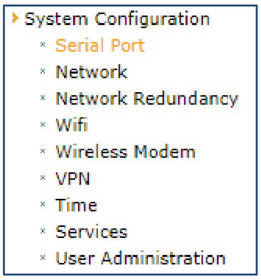
Vin: Voltage input to RTU
TMP: Is the temperature inside RTU
Type: 42-IO A indicates the type is a RTU with 42 built in I/O
INT Bus ACTIVE Indicates expansion modules on the bus are ACTIVE
INT Bus PASIVE Indicates exp. module is deactivated, or cable defective
AI Gives the value of the selected analogue channel
AI0 !OVF! Selected Analogue channel 0 has Overflow
AI0 !UDF! Selected Analogue channel 0 has Underflow
DI Gives status of selected inputs
DO Gives status of selected output
CT Gives value of selected counter.



The LCD auto scrolls through IO and RTU status displays. After 60s user inactivity, the display backlight turns off. Press Enter or Escape to halt/restart the auto scroll. When the scroll is halted, the up and down arrows can be used to navigate through the various displays.

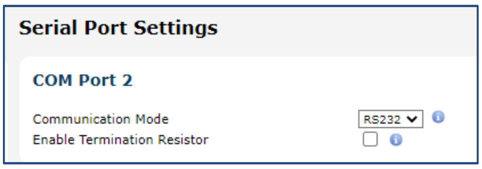
Note: if extra IO modules are added to the RTU, the additional values are then included in the display list.

9 System Configuration



9.1 COM 2 Serial Port

COM Port 2 of the RTU is configurable as RS232 or RS485.



Serial Port Settings

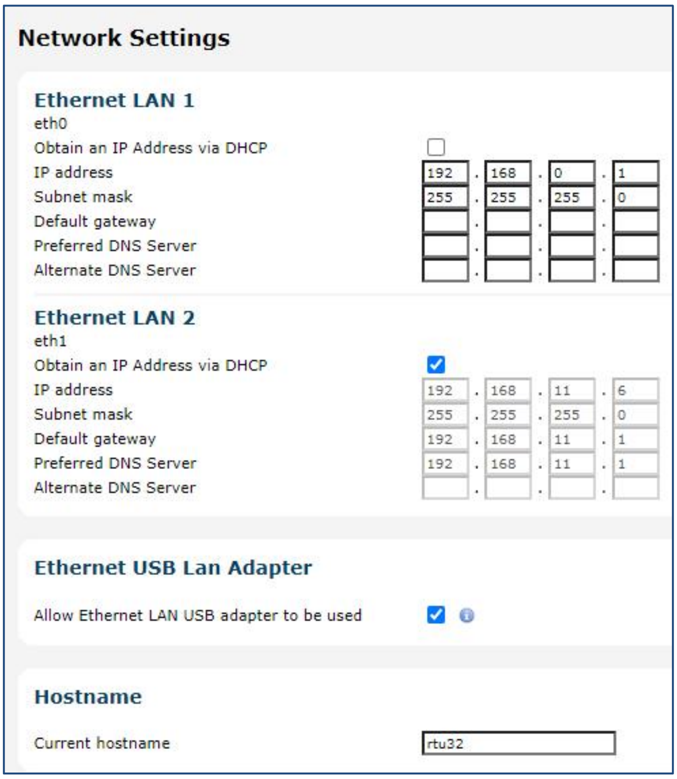
COM Port 2

Communication Mode	RS232
Enable Termination Resistor	☐

9.2 RTU Network Settings

On the System Configuration>Network Settings page you can change the LAN port settings to fit your local network. You can assign fixed IP addresses or select DHCP to have the address set via a router. You can then gain access to the RTU32N with your browser, SFTP client or Brodersen WorkSuite in your LAN network. You can also change the RTU Host Name. The default host name of the RTU32N is **rtu32**.

After entering new settings, click Apply to save. Note that the new settings are activated within a few seconds (you do not need to restart/reboot the RTU32N.



Network Settings

Ethernet LAN 1

eth0

Obtain an IP Address via DHCP	☐
IP address	192 . 168 . 0 . 1
Subnet mask	255 . 255 . 255 . 0
Default gateway	. . .
Preferred DNS Server	. . .
Alternate DNS Server	. . .

Ethernet LAN 2

eth1

Obtain an IP Address via DHCP	☑
IP address	192 . 168 . 11 . 6
Subnet mask	255 . 255 . 255 . 0
Default gateway	192 . 168 . 11 . 1
Preferred DNS Server	192 . 168 . 11 . 1
Alternate DNS Server	. . .

Ethernet USB Lan Adapter

Allow Ethernet LAN USB adapter to be used	☑

Hostname

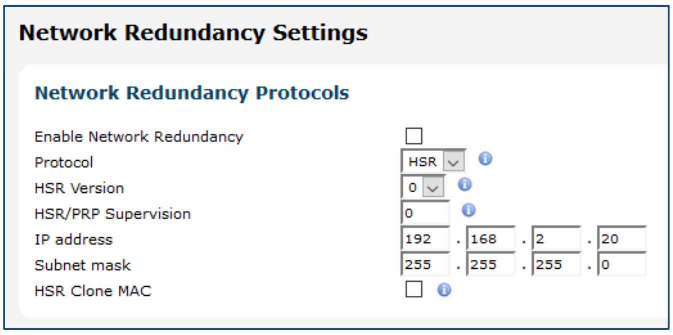
Current hostname	rtu32

9.3 Adding a USB-Ethernet adapter

The RTU32N supports addition of a USB Ethernet Adapter (Important, use only the Brodersen USB Ethernet adapter). After inserting the USB Ethernet adapter, ETHERNET LAN 3 will be included on the Network Settings web page and it must be configured in the same way as LAN 1 and LAN 2.

9.4 Network Redundancy Protocols

Modern substation automation systems use network redundancy to improve reliability and network changeover times. The RTU32N System Configuration>Network Redundancy page allows selection of network redundancy protocols HSR (High Availability Seamless Redundancy – where both LAN ports are used to ensure two network paths exist between adjacent nodes in a ring format), or PRP (Parallel Redundancy Protocol – where both LAN ports are used to communicate via independent/separated networks).



Network Redundancy Settings

Network Redundancy Protocols

Enable Network Redundancy	☐
Protocol	HSR
HSR Version	0
HSR/PRP Supervision	0
IP address	192 . 168 . 2 . 20
Subnet mask	255 . 255 . 255 . 0
HSR Clone MAC	☐

9.4.1 HSR – High Availability Seamless Redundancy

- Select HSR
- Select the correct HSR version
- HSR/PRP supervision "Specifies the last byte of the multicast address used for HSR/PRP supervision"
- Enter the IP Address that will be common for both LAN interfaces.
- Enter subnet mask or use default
- HSR clone MAC address specifies whether the eth0 adapters MAC is used for both LAN adapters (supported by HSR only).

9.4.2 PRP – Parallel Redundancy Protocol

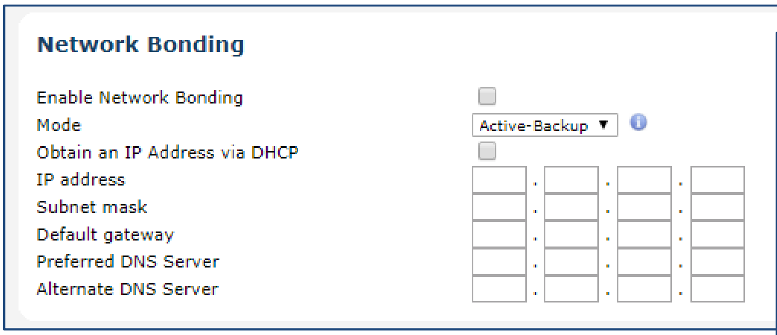
- Select PRP

- HSR/PRP supervision “Specifies the last byte of the multicast address used for HSR/PRP supervision”

- Enter the IP Address that will be common for both LAN interfaces.
- Enter subnet mask or use default
- HSR clone MAC is only supported by HSR

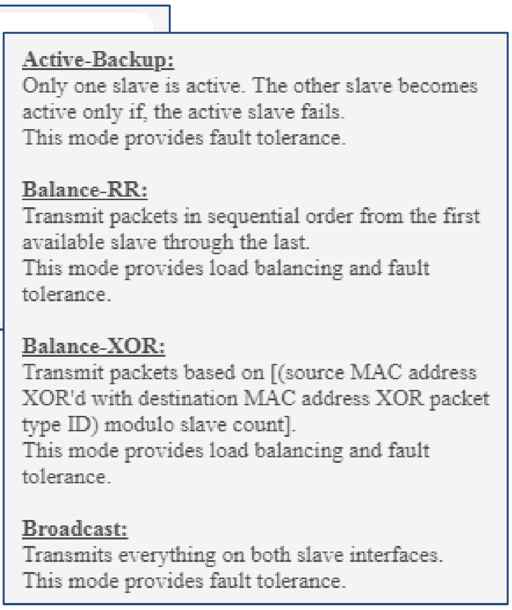
9.5 Network Redundancy - Bonding

Network Bonding allows both RTU LAN ports to act as a single interface to allow primary/standby modes or load balancing in network redundancy applications.



Network Bonding

Enable Network Bonding □
Mode: Active-Backup ▼
Obtain an IP Address via DHCP □
IP address: . . .
Subnet mask: . . .
Default gateway: . . .
Preferred DNS Server: . . .
Alternate DNS Server: . . .



Active-Backup:
Only one slave is active. The other slave becomes active only if, the active slave fails.
This mode provides fault tolerance.

Balance-RR:
Transmit packets in sequential order from the first available slave through the last.
This mode provides load balancing and fault tolerance.

Balance-XOR:
Transmit packets based on [(source MAC address XOR'd with destination MAC address XOR packet type ID) modulo slave count].
This mode provides load balancing and fault tolerance.

Broadcast:
Transmits everything on both slave interfaces.
This mode provides fault tolerance.

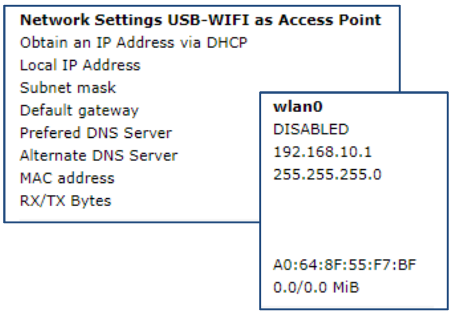
The four bonding modes include;

- Active- Backup
- Balance – RR
- Balance – XOR
- Broadcast

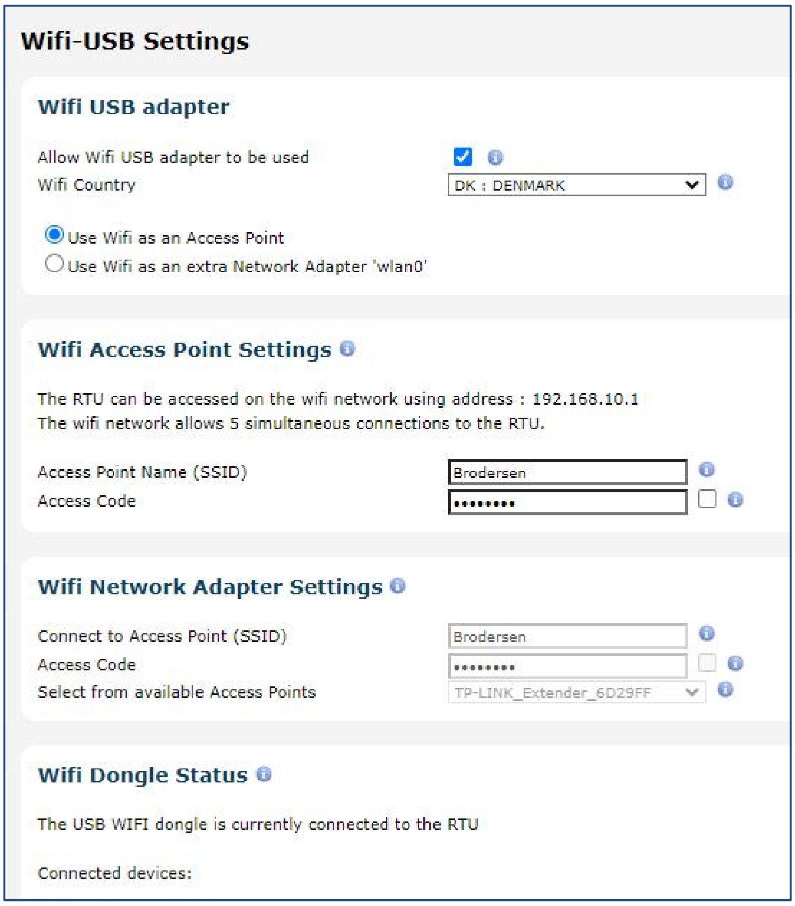
9.6 RTU Wifi Settings

The RTU32N supports addition of a USB WiFi Adapter to add an access point (default mode) or managed node (Important, use only Brodersen recommended USB adapters). The adapter can be disabled, configured and the mode changed from the Wifi-USB Settings page..

After inserting the USB Wifi adapter, an additional IP port will be added to the Network Settings web page.



Network Settings USB-WIFI as Access Point	wlan0
Obtain an IP Address via DHCP	DISABLED
Local IP Address	192.168.10.1
Subnet mask	255.255.255.0
Default gateway	
Prefered DNS Server	
Alternate DNS Server	
MAC address	A0:64:8F:55:F7:BF
RX/TX Bytes	0.0/0.0 MiB



Wifi-USB Settings

Wifi USB adapter

Allow Wifi USB adapter to be used ☑
Wifi Country: DK : DENMARK

● Use Wifi as an Access Point
○ Use Wifi as an extra Network Adapter 'wlan0'

Wifi Access Point Settings

The RTU can be accessed on the wifi network using address : 192.168.10.1
The wifi network allows 5 simultaneous connections to the RTU.

Access Point Name (SSID): Brodersen
Access Code: •••••••• □

Wifi Network Adapter Settings

Connect to Access Point (SSID): Brodersen
Access Code: •••••••• □
Select from available Access Points: TP-LINK_Extender_6D29FF

Wifi Dongle Status

The USB WIFI dongle is currently connected to the RTU

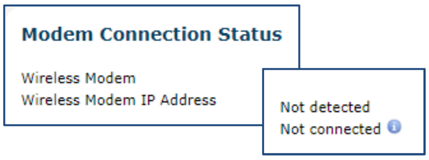
Connected devices:

9.7 Wireless Modem Settings

The RTU32N supports Brodersen IM51A and IM51B 3g/4g modems via a USB port.

In order to ensure the modem operates you will need to enter all credentials correctly. The webpage includes small help icons - put the cursor on them and you will get details for each item.

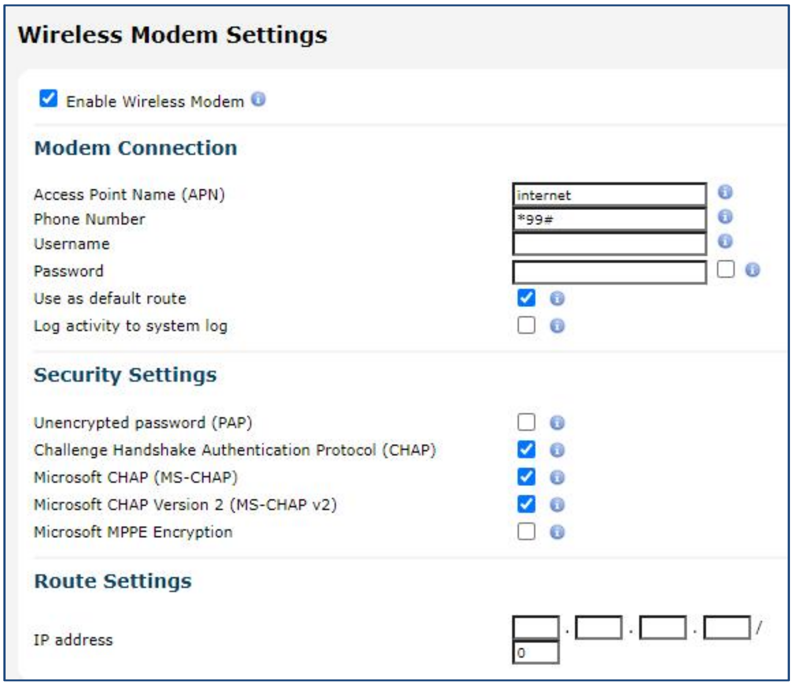
The modem connection status will be shown.



Modem Connection Status

Wireless Modem	Not detected
Wireless Modem IP Address	Not connected

If a connection is established, an additional IP port will also be added to the Network Settings web page.



Wireless Modem Settings

☑ Enable Wireless Modem

Modem Connection

Access Point Name (APN)	internet
Phone Number	*99#
Username	
Password	□
Use as default route	☑
Log activity to system log	□

Security Settings

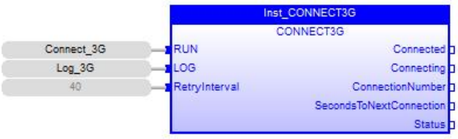
Unencrypted password (PAP)	□
Challenge Handshake Authentication Protocol (CHAP)	☑
Microsoft CHAP (MS-CHAP)	☑
Microsoft CHAP Version 2 (MS-CHAP v2)	☑
Microsoft MPPE Encryption	□

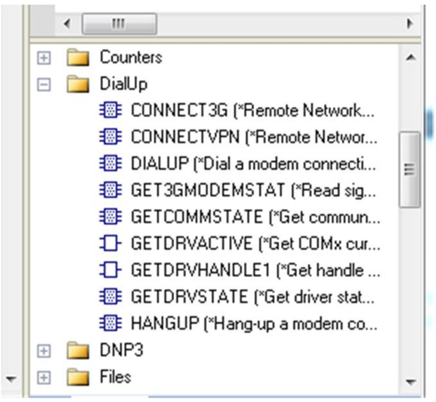
Route Settings

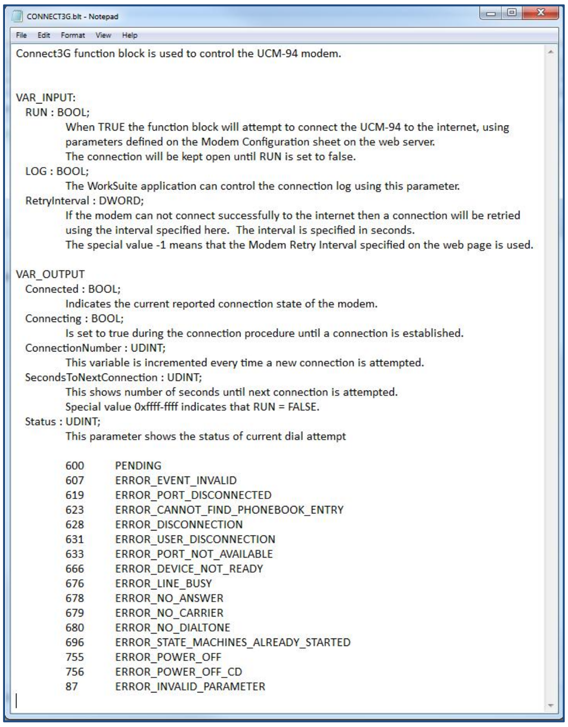
IP address	. . . / 0

9.7.1 Wireless Modem Control From Logic

If you want to control the connection from the RTU logic application, you must select Connection Control from application. And then in your logic application use the Function Block ‘Connect3G’ (found in the Dial-up function group).




Counters
DialUp
CONNECT3G (*Remote Network...
CONNECTVPN (*Remote Networ...
DIALUP (*Dial a modem connecti...
GET3GMODEMSTAT (*Read sig...
GETCOMMSTATE (*Get commun...
GETDRVACTIVE (*Get COMx cur...
GETDRVHANDLE1 (*Get handle ...
GETDRVSTATE (*Get driver stat...
HANGUP (*Hang-up a modem co...
DNP3
Files



```
Connect3G function block is used to control the UCM-94 modem.


VAR_INPUT:
  RUN : BOOL;
          When TRUE the function block will attempt to connect the UCM-94 to the internet, using
          parameters defined on the Modem Configuration sheet on the web server.
          The connection will be kept open until RUN is set to false.
  LOG : BOOL;
          The WorkSuite application can control the connection log using this parameter.
  RetryInterval : DWORD;
          If the modem can not connect successfully to the internet then a connection will be retried
          using the interval specified here.  The interval is specified in seconds.
          The special value -1 means that the Modem Retry Interval specified on the web page is used.

VAR_OUTPUT
  Connected : BOOL;
          Indicates the current reported connection state of the modem.
  Connecting : BOOL;
          Is set to true during the connection procedure until a connection is established.
  ConnectionNumber : UDINT;
          This variable is incremented every time a new connection is attempted.
  SecondsToNextConnection : UDINT;
          This shows number of seconds until next connection is attempted.
          Special value 0xffff-ffff indicates that RUN = FALSE.
  Status : UDINT;
          This parameter shows the status of current dial attempt

          600     PENDING
          607     ERROR_EVENT_INVALID
          619     ERROR_PORT_DISCONNECTED
          623     ERROR_CANNOT_FIND_PHONEBOOK_ENTRY
          628     ERROR_DISCONNECTION
          631     ERROR_USER_DISCONNECTION
          633     ERROR_PORT_NOT_AVAILABLE
          666     ERROR_DEVICE_NOT_READY
          676     ERROR_LINE_BUSY
          678     ERROR_NO_ANSWER
          679     ERROR_NO_CARRIER
          680     ERROR_NO_DIALTONE
          696     ERROR_STATE_MACHINES_ALREADY_STARTED
          755     ERROR_POWER_OFF
          756     ERROR_POWER_OFF_CD
          87      ERROR_INVALID_PARAMETER
```

Note: The Settings Overview web page will include the modem connection status at the bottom of the web page if the modem is connected. It will also show you the IP address assigned on the cellular network.

9.8 VPN Settings

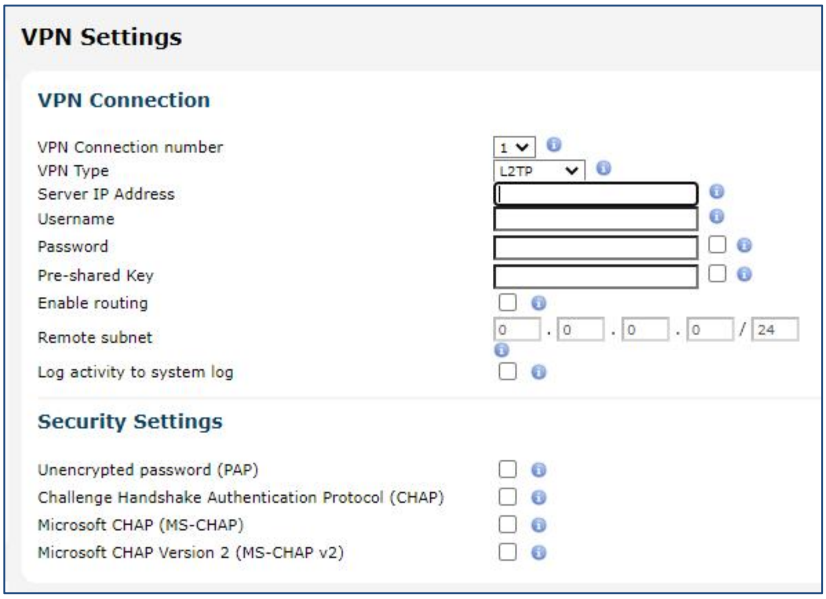
The RTU VPN Client supports PPTP and L2TP (IPsec) connections for up to two VPN Servers.

The settings for each VPN server interface can be adjusted from this webpage.

The VPN connection status will be shown and if a connection is established, an additional IP port will also be added to the Network Settings web page.

In order to ensure the VPN client connects you will need to enter all credentials correctly. The webpage includes small help icons - just put the cursor on them and you will get details for each item.

Controlling the connection of the VPN Client is managed from the RTU logic application. Use the Function Block ConnectVPN found in the Dial-up function block group to manage VPN connections.



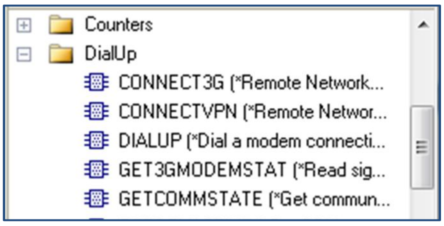
VPN Settings

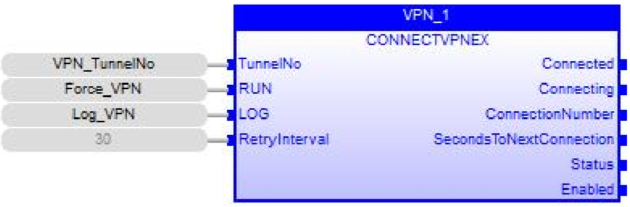
VPN Connection

VPN Connection number: 1
VPN Type: L2TP
Server IP Address:
Username:
Password: □
Pre-shared Key: □
Enable routing: □
Remote subnet: 0 . 0 . 0 . 0 / 24
Log activity to system log: □

Security Settings

Unencrypted password (PAP) □
Challenge Handshake Authentication Protocol (CHAP) □
Microsoft CHAP (MS-CHAP) □
Microsoft CHAP Version 2 (MS-CHAP v2) □


Counters
DialUp
CONNECT3G (*Remote Network...
CONNECTVPN (*Remote Networ...
DIALUP (*Dial a modem connecti...
GET3GMODEMSTAT (*Read sig...
GETCOMMSTATE (*Get commun...



9.9 RTU Time Settings

The time settings web page is used to configure and set the RTU32N real-time clock.

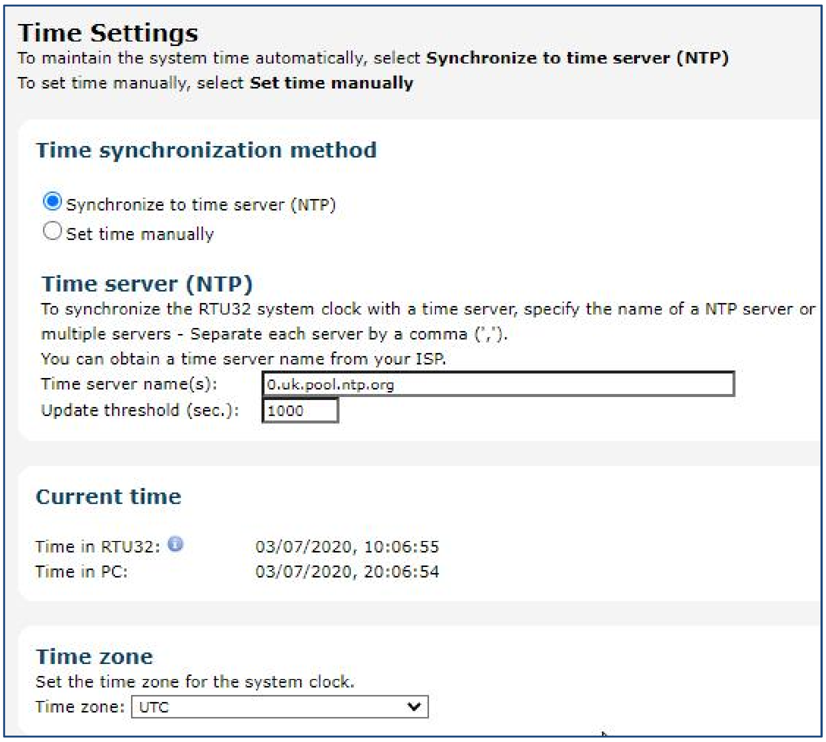
9.9.1 Time Settings

You can define if the RTU should be synchronized via a SNTP Server, or you can use other synchronization options like manual time setting or synchronization via utility drivers that include this functionality, such as IEC60870 or DNP3. NOTE: The time shown on the webpage is running in software and is only updated when the page is refreshed.

When set for synchronizing to a SNTP Server you must define the SNTP Server IP address or domain name. If you use a domain name, you must ensure that you have defined a DNS Server address in the LAN1 configuration. Update interval can be set from 2 to 20160 minutes.

9.9.2 Base Station time zone

If you use local time you have to set the time zone correctly. By default it is setup for Co-ordinated Universal Time (UTC).



Time Settings

To maintain the system time automatically, select **Synchronize to time server (NTP)**
To set time manually, select **Set time manually**

Time synchronization method

● Synchronize to time server (NTP)
○ Set time manually

Time server (NTP)

To synchronize the RTU32 system clock with a time server, specify the name of a NTP server or multiple servers - Separate each server by a comma (',').
You can obtain a time server name from your ISP.
Time server name(s): 0.uk.pool.ntp.org
Update threshold (sec.): 1000

Current time

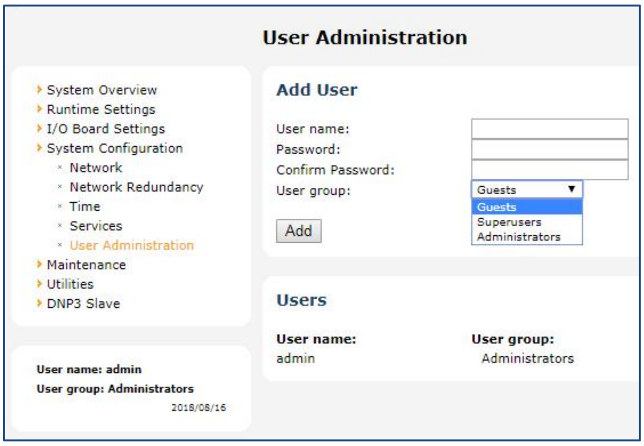
Time in RTU32: 03/07/2020, 10:06:55
Time in PC: 03/07/2020, 20:06:54

Time zone

Set the time zone for the system clock.
Time zone: UTC

9.10 User Administration

The User Administration page allows setup of user names and passwords. Each user is associated to a group of users – there can be up to 10 users in each group.



Add User

User name:
Password:
Confirm Password:
User group: Guests

Users

User name:	User group:
admin	Administrators

User name: admin
User group: Administrators
2018/08/16

The three user groups have increasing levels of privileges and include;

Guests	Read only access
Superusers	Full access and configuration, except for the Security pages (read only)
Administrators	Full access and configuration access

10 Maintenance

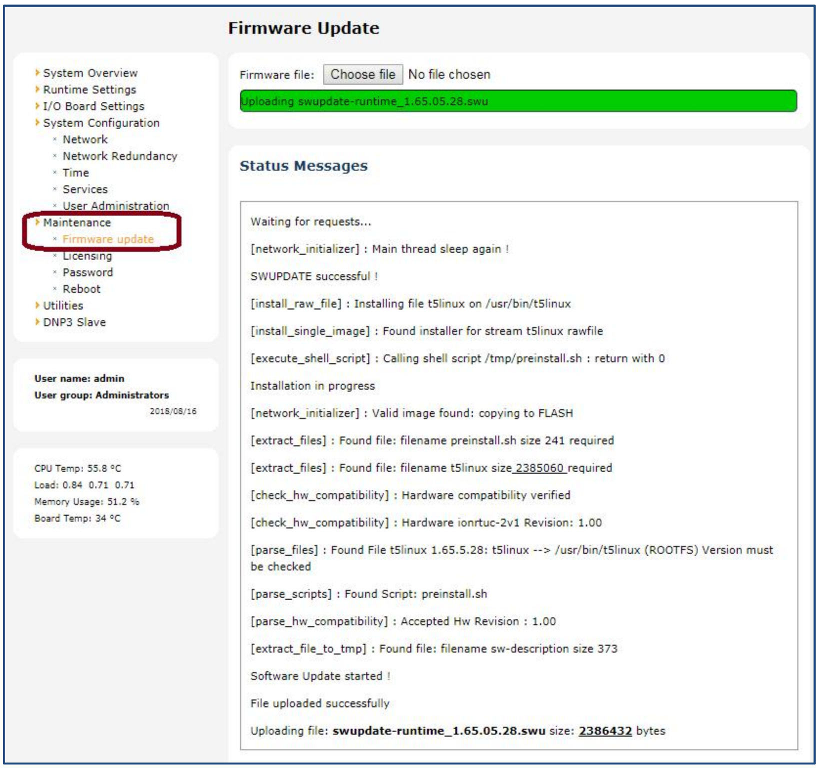
10.1 – Firmware Update

The Firmware Update interface allows loading of updates to the RTU.

Update files with a .swu extension are selected using the 'Choose File' button, then loaded to the RTU.

The update script provides feedback to the user as the update progresses, as shown in the example below.

Note: sometimes the firmware update may fail to report it has completed. If this occurs, reboot the RTU and check the System Overview page to see if the firmware version information has been updated.



Firmware Update

Firmware file: No file chosen

Uploading swupdate-runtime_1.65.05.28.swu

User name: admin
User group: Administrators
2018/08/16

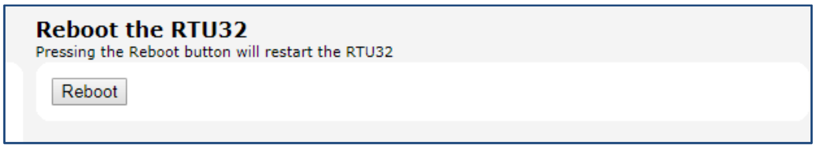
CPU Temp: 55.8 °C
Load: 0.84 0.71 0.71
Memory Usage: 51.2 %
Board Temp: 34 °C

Status Messages

```
Waiting for requests...

[network_initializer] : Main thread sleep again !

SWUPDATE successful !

[install_raw_file] : Installing file t5linux on /usr/bin/t5linux

[install_single_image] : Found installer for stream t5linux rawfile

[execute_shell_script] : Calling shell script /tmp/preinstall.sh : return with 0

Installation in progress

[network_initializer] : Valid image found: copying to FLASH

[extract_files] : Found file: filename preinstall.sh size 241 required

[extract_files] : Found file: filename t5linux size 2385060 required

[check_hw_compatibility] : Hardware compatibility verified

[check_hw_compatibility] : Hardware ionrtuc-2v1 Revision: 1.00

[parse_files] : Found File t5linux 1.65.5.28: t5linux --> /usr/bin/t5linux (ROOTFS) Version must
be checked

[parse_scripts] : Found Script: preinstall.sh

[parse_hw_compatibility] : Accepted Hw Revision : 1.00

[extract_file_to_tmp] : Found file: filename sw-description size 373

Software Update started !

File uploaded successfully

Uploading file: swupdate-runtime_1.65.05.28.swu size: 2386432 bytes
```

10.2 Reboot

After changing parameters on the configuration pages, you must unless otherwise noted, run the reboot function to activate the changes. After booting the RTU32N you can start working with Brodersen WorkSuite to create your application programs.

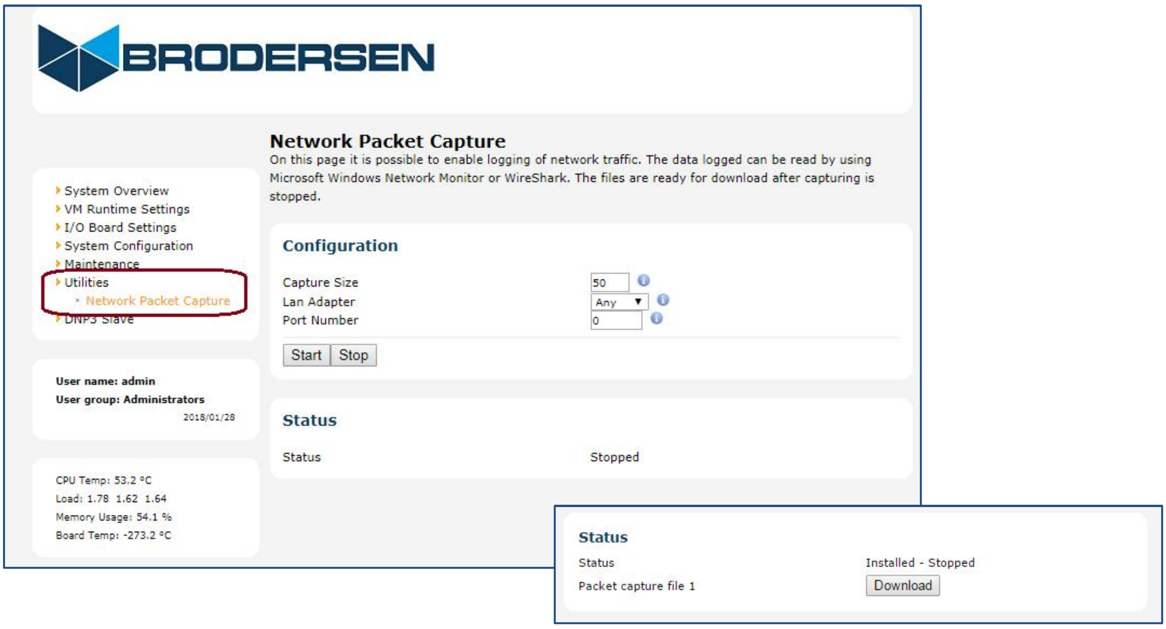


Reboot the RTU32

Pressing the Reboot button will restart the RTU32

11 Utilities - Network Packet Capture

The Network Packet Capture utility allows IP traffic to be monitored and saved to a .pcap file. The file can then be uploaded to a PC and viewed by applications including WireShark. Capture begins when the start button is selected. Files are ready for uploading when the capture is stopped.



Network Packet Capture

On this page it is possible to enable logging of network traffic. The data logged can be read by using Microsoft Windows Network Monitor or WireShark. The files are ready for download after capturing is stopped.

Configuration

Capture Size: 50
Lan Adapter: Any
Port Number: 0

Status

Status: Stopped

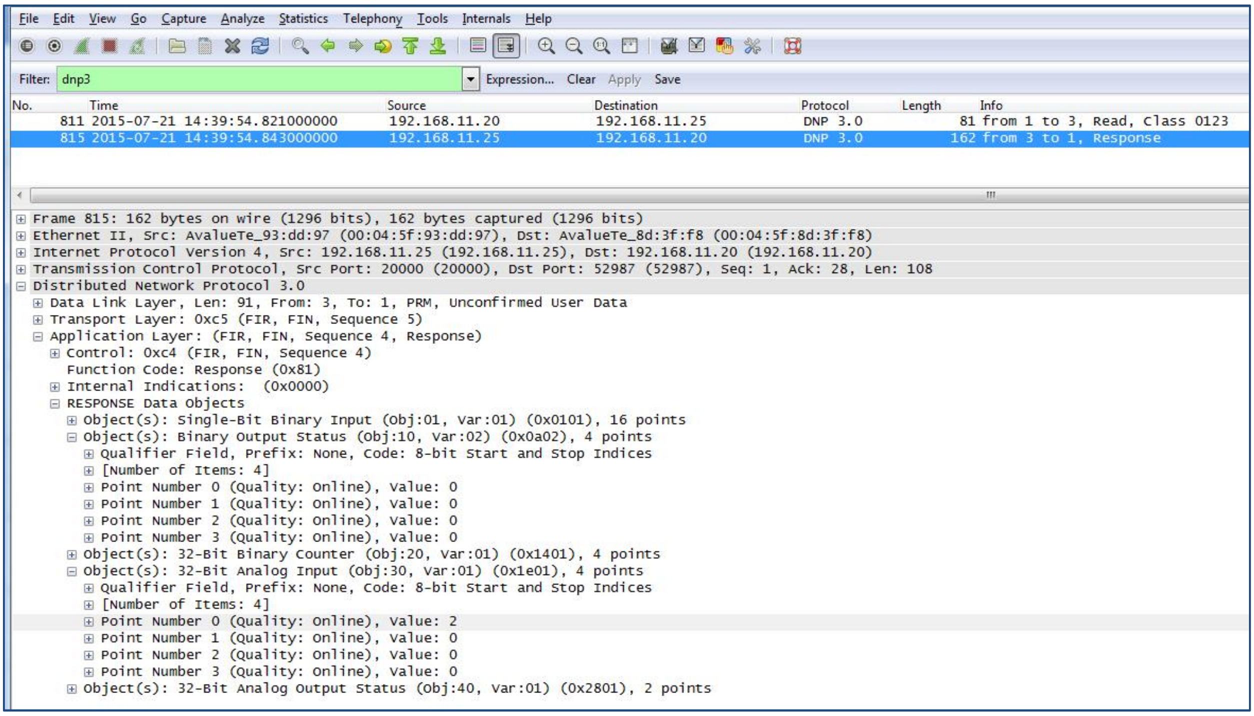
User name: admin
User group: Administrators
2018/01/28

CPU Temp: 53.2 °C
Load: 1.78 1.62 1.64
Memory Usage: 54.1 %
Board Temp: -273.2 °C

Status

Status: Installed - Stopped
Packet capture file 1: Download

Eg. example below shows a captured DNP3 integrity poll and the response from the RTU32N can be viewed in WireShark. A dnp3 filter has been applied to allow the message to be decoded.



Filter: dnp3

```
No.   Time                           Source          Destination     Protocol  Length  Info
  811 2015-07-21 14:39:54.821000000  192.168.11.20   192.168.11.25   DNP 3.0        81 from 1 to 3, Read, Class 0123
  815 2015-07-21 14:39:54.843000000  192.168.11.25   192.168.11.20   DNP 3.0       162 from 3 to 1, Response
```

```
⊞ Frame 815: 162 bytes on wire (1296 bits), 162 bytes captured (1296 bits)
⊞ Ethernet II, Src: AvalueTe_93:dd:97 (00:04:5f:93:dd:97), Dst: AvalueTe_8d:3f:f8 (00:04:5f:8d:3f:f8)
⊞ Internet Protocol Version 4, Src: 192.168.11.25 (192.168.11.25), Dst: 192.168.11.20 (192.168.11.20)
⊞ Transmission Control Protocol, Src Port: 20000 (20000), Dst Port: 52987 (52987), Seq: 1, Ack: 28, Len: 108
⊟ Distributed Network Protocol 3.0
  ⊞ Data Link Layer, Len: 91, From: 3, To: 1, PRM, Unconfirmed User Data
  ⊞ Transport Layer: 0xc5 (FIR, FIN, Sequence 5)
  ⊟ Application Layer: (FIR, FIN, Sequence 4, Response)
    ⊞ Control: 0xc4 (FIR, FIN, Sequence 4)
      Function Code: Response (0x81)
    ⊞ Internal Indications:  (0x0000)
    ⊟ RESPONSE Data Objects
      ⊞ Object(s): Single-Bit Binary Input (Obj:01, Var:01) (0x0101), 16 points
      ⊟ Object(s): Binary Output Status (Obj:10, Var:02) (0x0a02), 4 points
        ⊞ Qualifier Field, Prefix: None, Code: 8-bit Start and Stop Indices
        ⊞ [Number of Items: 4]
        ⊞ Point Number 0 (Quality: Online), Value: 0
        ⊞ Point Number 1 (Quality: Online), Value: 0
        ⊞ Point Number 2 (Quality: Online), Value: 0
        ⊞ Point Number 3 (Quality: Online), Value: 0
      ⊞ Object(s): 32-Bit Binary Counter (Obj:20, Var:01) (0x1401), 4 points
      ⊟ Object(s): 32-Bit Analog Input (Obj:30, Var:01) (0x1e01), 4 points
        ⊞ Qualifier Field, Prefix: None, Code: 8-bit Start and Stop Indices
        ⊞ [Number of Items: 4]
        ⊞ Point Number 0 (Quality: Online), Value: 2
        ⊞ Point Number 1 (Quality: Online), Value: 0
        ⊞ Point Number 2 (Quality: Online), Value: 0
        ⊞ Point Number 3 (Quality: Online), Value: 0
      ⊞ Object(s): 32-Bit Analog Output Status (Obj:40, Var:01) (0x2801), 2 points
```

12 DNP3 Slave – Configuration

The DNP3 Slave Configuration page allows slave setup parameters to be edited ‘outside’ of the RTU logic application.

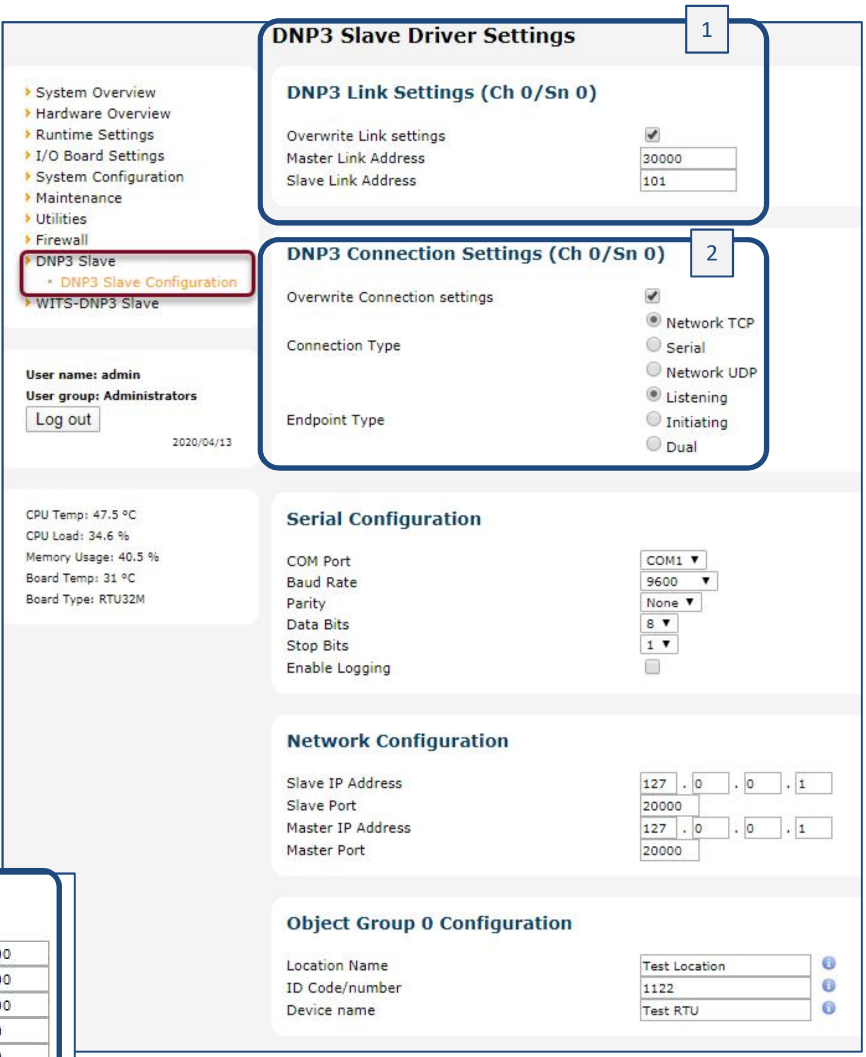
Link Settings [1]
By default the ‘Overwrite Link Settings’ parameter is disabled. If enabled, the RTU will read the DNP3 settings defined here, rather than the settings defined in the logic application.

Connection Settings [2]
By default the ‘Overwrite Connection Settings’ parameter is disabled. If enabled, the RTU will read the DNP3 settings defined here, rather than the settings defined in the logic application.

These parameters are stored in a file (etc/rtu32/config/dnp3s.ini) that can be manipulated by the logic application and/or overwritten by the SCADA host.

Note 1: the dnp3s.ini file is created by the RTU if settings are changed/applied.

Note 2: the parameters/values are only read after a RTU restart/boot.



User name: admin
User group: Administrators
2020/04/13

CPU Temp: 47.5 °C
CPU Load: 34.6 %
Memory Usage: 40.5 %
Board Temp: 31 °C
Board Type: RTU32M

DNP3 Slave Driver Settings

DNP3 Link Settings (Ch 0/Sn 0)

Overwrite Link settings	☑
Master Link Address	30000
Slave Link Address	101

DNP3 Connection Settings (Ch 0/Sn 0)

Overwrite Connection settings	☑
Connection Type	● Network TCP ○ Serial ○ Network UDP
Endpoint Type	● Listening ○ Initiating ○ Dual

Serial Configuration

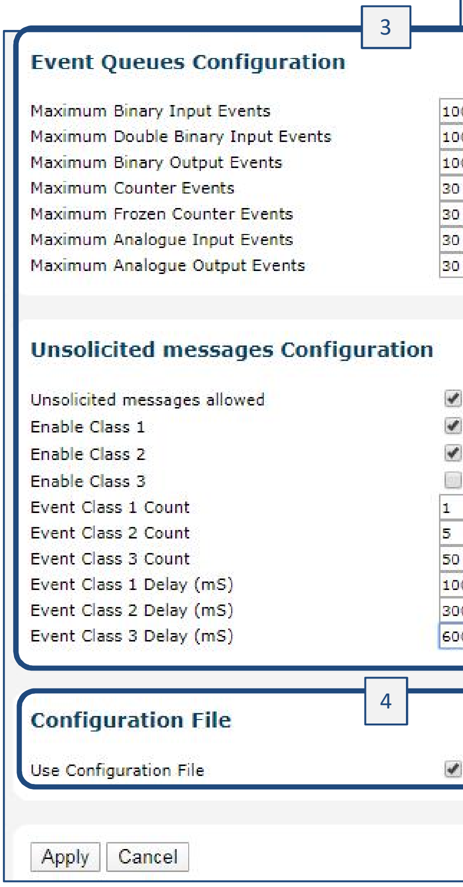
COM Port	COM1 ▼
Baud Rate	9600 ▼
Parity	None ▼
Data Bits	8 ▼
Stop Bits	1 ▼
Enable Logging	□

Network Configuration

Slave IP Address	127 . 0 . 0 . 1
Slave Port	20000
Master IP Address	127 . 0 . 0 . 1
Master Port	20000

Object Group 0 Configuration

Location Name	Test Location
ID Code/number	1122
Device name	Test RTU



Event Queues Configuration

Maximum Binary Input Events	100
Maximum Double Binary Input Events	100
Maximum Binary Output Events	100
Maximum Counter Events	30
Maximum Frozen Counter Events	30
Maximum Analogue Input Events	30
Maximum Analogue Output Events	30

Unsolicited messages Configuration

Unsolicited messages allowed	☑
Enable Class 1	☑
Enable Class 2	☑
Enable Class 3	□
Event Class 1 Count	1
Event Class 2 Count	5
Event Class 3 Count	50
Event Class 1 Delay (mS)	1000
Event Class 2 Delay (mS)	30000
Event Class 3 Delay (mS)	60000

Configuration File

Use Configuration File	☑

DNP3 Application Layer Settings [3]:
The application layer settings allow setup of the size of DNP3 event queues and event reporting rules outside of the logic application. The application layer settings from the dnp3s.ini file are used instead of the values defined in the Fieldbus Editor application layer settings associated to the logic application, if the dnp3s.ini file exists.

DNP3 Points Configuration File [4]:
The parameter ‘Use Configuration File’ determines if the RTU reads the DNP point parameters (eg. Event class, threshold, debounce etc) from an external file (etc/rtu32/dnp3cfg.act), or from the DNP3 slave profile associated to the logic application.

Further details regarding ‘Host Managed DNP3 Slave Setup’ where the RTU32 DNP3 slave setup is managed outside of the logic application is available in a separate document.

13 .Getting Started with the Brodersen WorkSuite tools (> v3.9.2.1356)

13.1 Introduction

The Brodersen WorkSuite Package includes all software required to configure and program the Brodersen RTU32 Series.

The software components included in the package include:

- Brodersen WorkSuite logic configuration and programming tool.
- WIBUKey driver for the USB license key dongle for WorkSuite.
- RTU32 Series tool package including:
    - Event Viewer – the telnet tool used for monitoring the System Log(s) in RTU32 Series products.
    - RTU32 Search tool for scanning and finding RTU32 products running on networks.

All software is installed in one installation process. The installation process will ask you what software to install and after making your selection, the installation process will run automatically and install and configure the all software, making it easy for you to begin working with configuration and programming of the Brodersen RTU32 Series products.

13.2 Installing Brodersen WorkSuite Package

Go to our website www.brodersen.com and find the WorkSuite Installation Guide. It will guide you through all the installation details.

13.3 WorkSuite License

The WorkSuite tool requires a license to configure and program RTUs. Without a license the software package works in DEMO mode.

The license is locked to a USB hardware dongle that is included in the package. In the package you find also the WorkSuite License Certificate with a License id number required to enter in the License utility after installing the software.
The standard license offers the possibility to create an unlimited number of I/O points in the RTU application program. All variables associated to communications drivers (except for Modbus Slave) are also counted as I/Os.

13.4 Compatibility with STRATON WorkBench

The Brodersen WorkSuite Package replaces all versions of STRATON WorkBench. The WorkSuite is backwards compatible with all STRATON versions and supports directly any older STRATON version 5.x, 6.x, 7.x and 8.x projects.

WorkSuite is a Brodersen OEM version of STRATON, but includes a range of additional facilities:
1. One installation process for WorkSuite - installs all software including license key drivers and all drivers, functions and Brodersen Tools supported by the RTU32 Series products.
2. Eliminates the need to install special hardware libraries, tools etc. After the installation of the WorkSuite Package you are ready to go.
3. All drivers supported by the RTU32 Series of products are available in WorkSuite, making for easier software version management.

Important Notes:
1. If you have existing licenses for STRATON WorkBench version 8.x, they are 100% compatible with the WorkSuite.
2. If you have an older Straton license, please contact Brodersen at sales@brodersen.com or your local representative for update prices. Please remember to include a copy of your current License Certificate.

13.5 User documentation and training/guideline videos

When installing the Brodersen WorkSuite package, User Guides and Manuals are installed as well. You get access to these documents from the Help menu within WorkSuite.

WorkSuite has also implemented a help function that means that you can get help by pressing F1. If no valid help opens, please check the PDF manuals available under the general WorkSuite Help menu or check our website www.brodersen.com for documentation.

We recommend that you use the Brodersen Training/Guideline Videos.

www.youtube.com/brodersensystems

Here you will find step by step guidelines with details of how to get started and how to setup drivers etc.

14 RTU32N I/O Setup in WorkSuite

14.1 General

It is highly recommended to take a training course on the RTU32 product series and the WorkSuite package. It will save you a lot of time and is a good investment because you will get familiar with the software and hardware and be able to quickly take advantage of all the RTU and WorkSuite powerful features. Contact your local distributor, or Brodersen for RTU/WorkSuite training exercises.

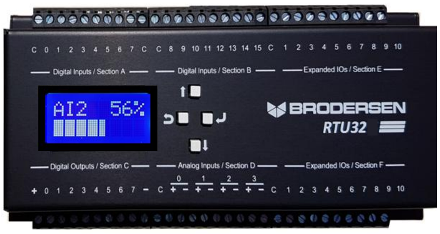
Besides the basic WorkSuite standard PLC programming environment and EN/IEC61131-3 programming languages, there are some functions and features which are specific for the RTU32N Product Series.

14.2 WorkSuite I/O Drivers for RTU32N

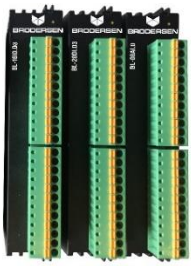
The WorkSuite I/O Drivers for the RTU32N are used to access the physical inputs and outputs on the RTU32N and the connected I/O expansion modules. All RTU inputs are read online when the PLC runtime cycle is started and all outputs are written online at the end of the runtime cycle.

NOTE: The drivers described in this section are for online I/O (both onboard the RTU32N and via expansion I/O modules if fitted).

The RTU32N supports newer LB2/RTU32M Series expansion I/O modules via the LB2 port (located at the bottom of the RTU) and 'legacy' UCL Series expansion I/O modules via the LocalBus port (located at the top of the RTU).


AI2 56%
BRODERSEN
RTU32
Digital Inputs / Section A
Digital Inputs / Section B
Expanded IOs / Section E
Digital Outputs / Section C
Analog Inputs / Section D
Expanded IOs / Section F

RTU32N



New LB2 Series I/O

Older UCL Series I/O

All I/Os (including status flags/words) are stored in an independent memory mapped database in the RTU32N. The database is a mirror of the physical I/O. The Internal I/O Bus and the LocalBus for expansion modules are responsible for keeping this updated at any time.

The WorkSuite I/O drivers handle the access to this database. You can program your logic application without any connected hardware.

You must be sure that you have the correct I/O configuration and modules when you try to run and debug your application program on the RTU32N (LB2 I/O modules can be flagged as 'not present').

14.3 INTRODUCTION to LB2/RTU32M I/O series

The Brodersen LB2 Series I/O modules can be used with RTU32N & RTU32M Series products. The modules are in two parts - the bottom part contains the backplane bus and top part contains the I/O board and logic. All LB2 modules can be hot swapped.

The modules are equipped with a 200 MHz processor to process I/O, perform filtering, SOE, debounce, module clock and general module logic. Diagnostic variables are available for all modules - see the module datasheets for full info. Firmware updates are handled from Brodersen Worksuite.

Use only genuine Brodersen bus cables for connection to Brodersen RTUs and for extension of I/O module blocks. The LB2 bus connection cables are a special build, to handle the power requirements and shielding to run bus communications. The maximum overall length of a complete system is 30m. Each I/O module & Power supply module is calculated as 2cm. The cables are available in various lengths from 25cm to 200cm.

Each LB2 Bus supports up to 60 I/O modules.

LB2 Cables

Description	**Part Number**
100cm LB2 Cable	UCC-610/1
200cm LB2 Cable	UCC-610/2
25cm LB2 Cable	UCC-610/25
50cm LB2 Cable	UCC-610/50

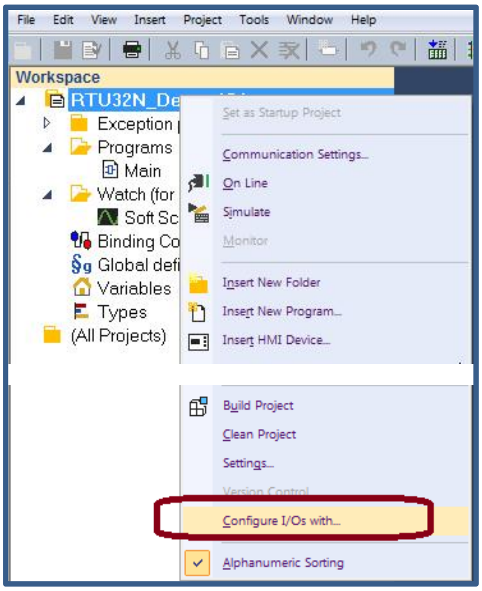
MODULE BACKPLANE PARTS

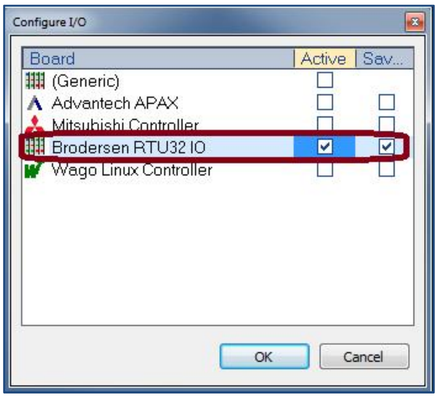
Description	**Part Number**
Bus I/O module Start	BB21A
BUS I/O module, Middle	BB21B
BUS I/O module, Expansion (End)	BB21C
BUS Power supply module, Start	BB41A
BUS Power supply module, Middle	BB41B
BUS P/S module, Extend (power to right side of P/S only)	BB41E
BUS CPU module, Start	BB61A
BUS CPU module, Middle	BB61B
BUS CPU module, Redundant (stby)	BB61R
BUS System module, Start	BB81A
BUS System module, Middle	BB81B

14.4 LB2 Series Module Configurator

All LB2 Series modules are configured in Worksuite.

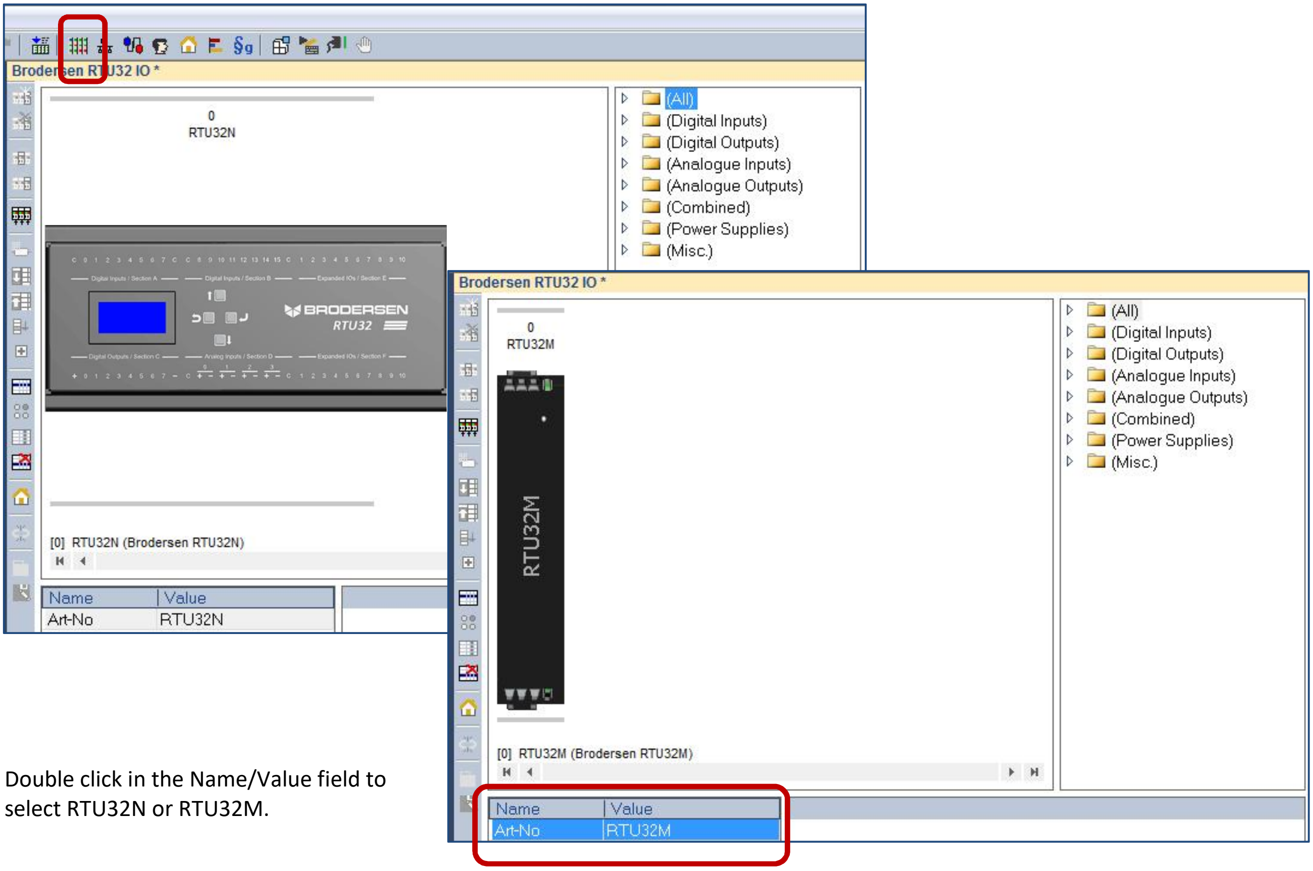
After creating a new project, right click the project heading and select 'Configure I/Os with' and select the Brodersen RTU32IO options.





Board	Active	Sav...
(Generic)	□	
Advantech APAX	□	□
Mitsubishi Controller	□	□
Brodersen RTU32 IO	☑	☑
Wago Linux Controller	□	□

Then select the 'Open I/Os' icon in the top toolbar to access the Module Configurator.



Name	Value
Art-No	RTU32N

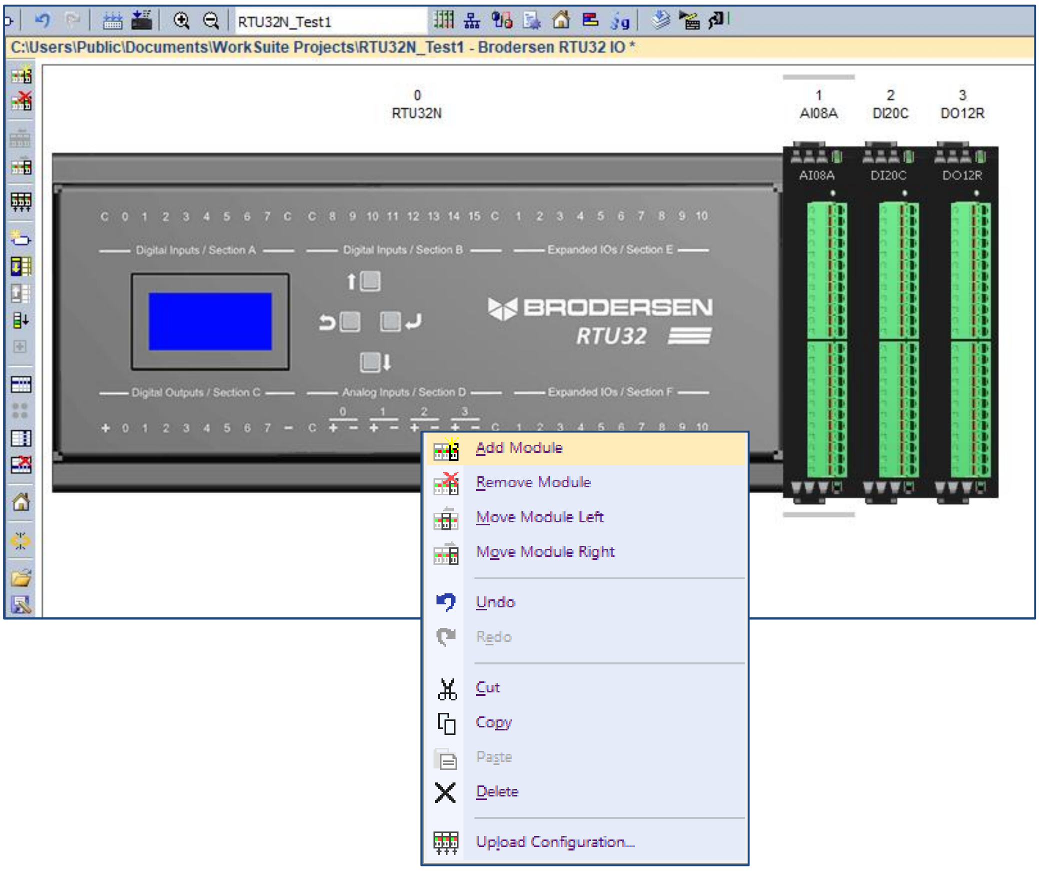
Name	Value
Art-No	RTU32M

Double click in the Name/Value field to select RTU32N or RTU32M.

14.4.1 Managing LB2 Series I/O Modules

The Module Configurator allows creation and editing of the I/O module setup using a mix of toolbar icons and drag and drop menu items.

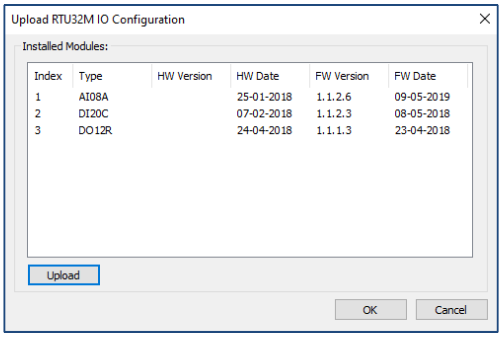
Insert new module
Remove module
Move module left
Move module right
Upload configuration
Insert variable
Move down
Move up
Sort symbols
Iterate property
Set variables in module
Set status variables
Set all offsets
Remove module variables
Declare variables in database
Disable Configuration
Import variables
Export variables



14.4.2 Inserting a module

Modules can be inserted in several ways.

- Drag and drop a module from the module list.
- Right click in the I/O module window and select ADD Module
- Copy existing I/O using Windows copy/paste commands.
- Right click in the I/O module window and select 'Upload Configuration' (uploads modules from the online RTU).


Upload RTU32M IO Configuration

Installed Modules:

Index	Type	HW Version	HW Date	FW Version	FW Date
1	AI08A		25-01-2018	1.1.2.6	09-05-2019
2	DI20C		07-02-2018	1.1.2.3	08-05-2018
3	DO12R		24-04-2018	1.1.1.3	23-04-2018

14.4.3 Moving a module

Modules can be moved left or right by placing the mouse cursor on the module – then press and hold the left mouse button to drag the module to the desired position, or press the right mouse button and select 'Move Module Left' or 'Move Module Right'.

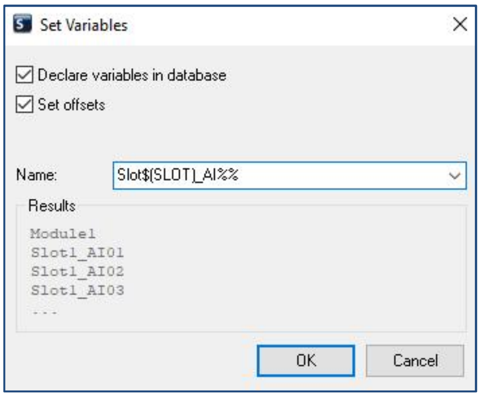
14.4.4 Remove a module in Worksuite

Modules can Removed/Deleted by placing the mouse cursor on the module - then press Delete on your keyboard, or press the right mouse button and select 'Remove Module'.

14.4.5 Add variables to a module

In the left pane of the I/O module window, the Icon ▦ 'Set Variables In Module' opens the variables editor. Remember to check 'Declare variables in database' if the variables are to be used as Global Variables.

Use of syntaxes allows flexibility to engineer your own variable names.



Set Variables

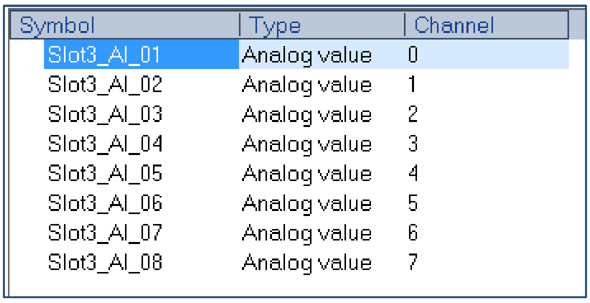
☑ Declare variables in database
☑ Set offsets

Name: Slot$(SLOT)_AI%%

Results

```
Module1
Slot1_AI01
Slot1_AI02
Slot1_AI03
...
```

Use the Drop Down list to select a suitable syntax, or make your own. For example, if the syntax $(SLOT)_$(OFFSET)_VAR is selected, the variables will be as shown below for an 8 channel analogue input module.



Symbol	Type	Channel
Slot3_AI_01	Analog value	0
Slot3_AI_02	Analog value	1
Slot3_AI_03	Analog value	2
Slot3_AI_04	Analog value	3
Slot3_AI_05	Analog value	4
Slot3_AI_06	Analog value	5
Slot3_AI_07	Analog value	6
Slot3_AI_08	Analog value	7

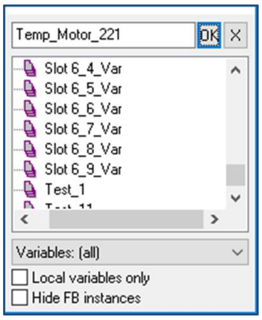
As another example, the syntax could be:
Module_Nr_$(SLOT)_Analogue_In_$(OFFSET)

This would create eight variables as shown below;
'Module_Nr_2_Analogue_In_1'
'Module_Nr_2_Analogue_In_2'
'Module_Nr_2_Analogue_In_3'
…….
'Module_Nr_2_Analogue_In_7'
'Module_Nr_2_Analogue_In_8'

14.4.6 Rename auto generated variables

In cases where variable names can't be easily generated through a syntax, they can be created/edited manually by placing the mouse cursor on the variable and using DoubleClick to access the variables editor.

The user can then rename of the variable or create a new variable.

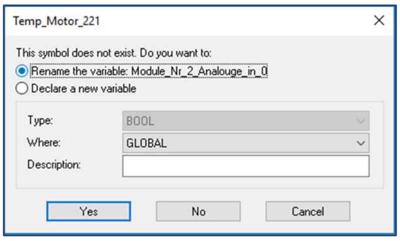


Temp_Motor_221

Slot 6_4_Var
Slot 6_5_Var
Slot 6_6_Var
Slot 6_7_Var
Slot 6_8_Var
Slot 6_9_Var
Test_1

Variables: (all)

□ Local variables only
□ Hide FB instances



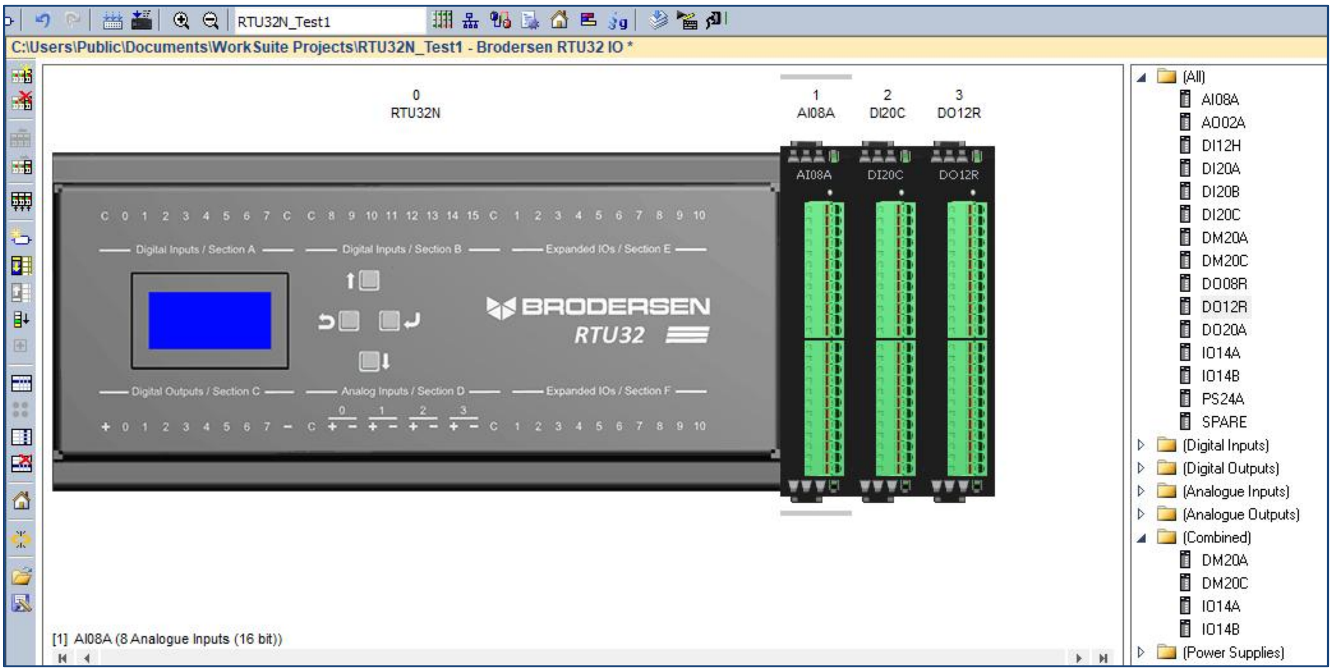
Temp_Motor_221

This symbol does not exist. Do you want to:
● Rename the variable: Module_Nr_2_Analouge_in_0
○ Declare a new variable

Type: BOOL
Where: GLOBAL
Description:

14.4.7 Configure Module Defaults

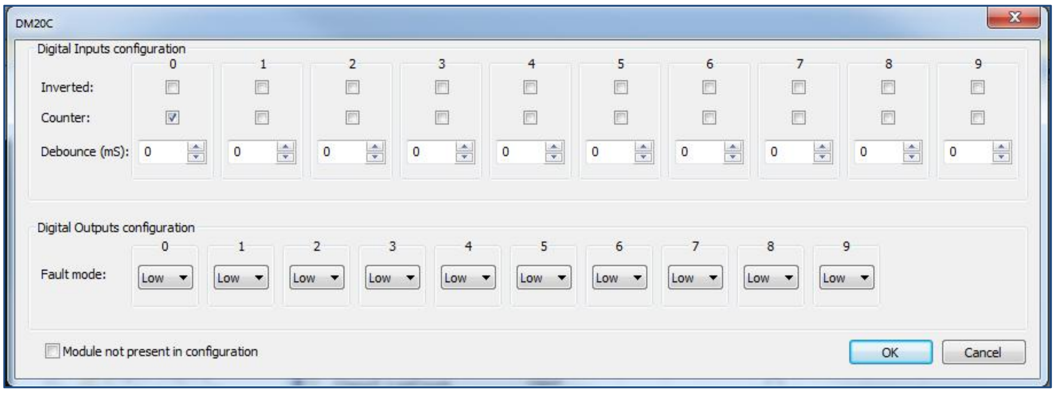
After creating or uploading a module array, module configurations can be modified from their default settings if required. Features and settings for each module can be found on their respective datasheets.



DoubleClick on a module in the I/O configurator to access the module I/O editor.

14.4.8 Configure Combination DI/DO module example

In the example below a combination DI/DO module with 10 DI and 10 DO has been selected.



Digital Inputs configuration

	0	1	2	3	4	5	6	7	8	9
Inverted:	□	□	□	□	□	□	□	□	□	□
Counter:	☑	□	□	□	□	□	□	□	□	□
Debounce (mS):	0	0	0	0	0	0	0	0	0	0

Digital Outputs configuration

	0	1	2	3	4	5	6	7	8	9
Fault mode:	Low	Low	Low	Low	Low	Low	Low	Low	Low	Low

□ Module not present in configuration

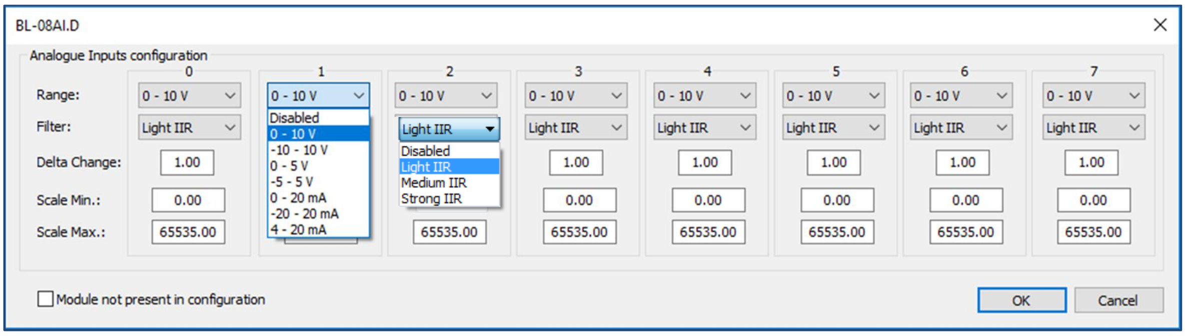
Configuration Fields - Inputs and Outputs:

- **Inverted:** Select if input is to be inverted.
- **Counter:** Select if input is a counter. Note: If an input is receiving a 5 Khz pulse/count, but not defined as a counter, the I/O bus will be spammed with 5000 pulses/sec!
- **Debounce:** Debounce in milliseconds, determines how long time the I/O must be high or low before it is detected as state change.
- **Fault Mode:** Fault mode is when the module loses contact with the CPU. It is necessary to select for all channels.
    - Keep (I/O module will keep last state)

- o Low (Output will be forced Low)
- o High (Output will be forced High)

14.4.9 Configure AI module example

In the example below an eight channel AI module has been selected.



BL-08AI.D

Analogue Inputs configuration

	0	1	2	3	4	5	6	7
Range:	0 - 10 V	0 - 10 V	0 - 10 V	0 - 10 V	0 - 10 V	0 - 10 V	0 - 10 V	0 - 10 V
Filter:	Light IIR		Light IIR	Light IIR	Light IIR	Light IIR	Light IIR	Light IIR
Delta Change:	1.00			1.00	1.00	1.00	1.00	1.00
Scale Min.:	0.00			0.00	0.00	0.00	0.00	0.00
Scale Max.:	65535.00		65535.00	65535.00	65535.00	65535.00	65535.00	65535.00

☐ Module not present in configuration

Configuration Fields - Inputs:

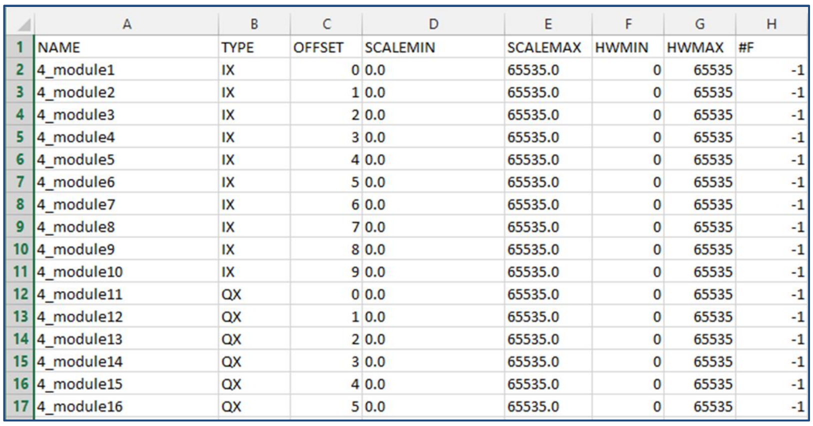
- **Range:** Select input type for each channel in dropdown menu.
- **Filter:** select filter type for each channel in dropdown menu.
- **Delta change:** See AI module datasheet
- **Scale Min:** See AI module datasheet
- **Scale Max:** See AI module datasheet

14.4.10 Export/Import Module variables.

WorkSuite allows exporting and importing of variables for alternative methods of editing using document and spreadsheet software.

Export: Select the I/O module of interest and then select the Export Variables icon in the toolbar on the left to create a CSV export file.

The file includes Scale min, Scale Max, Variable name etc.

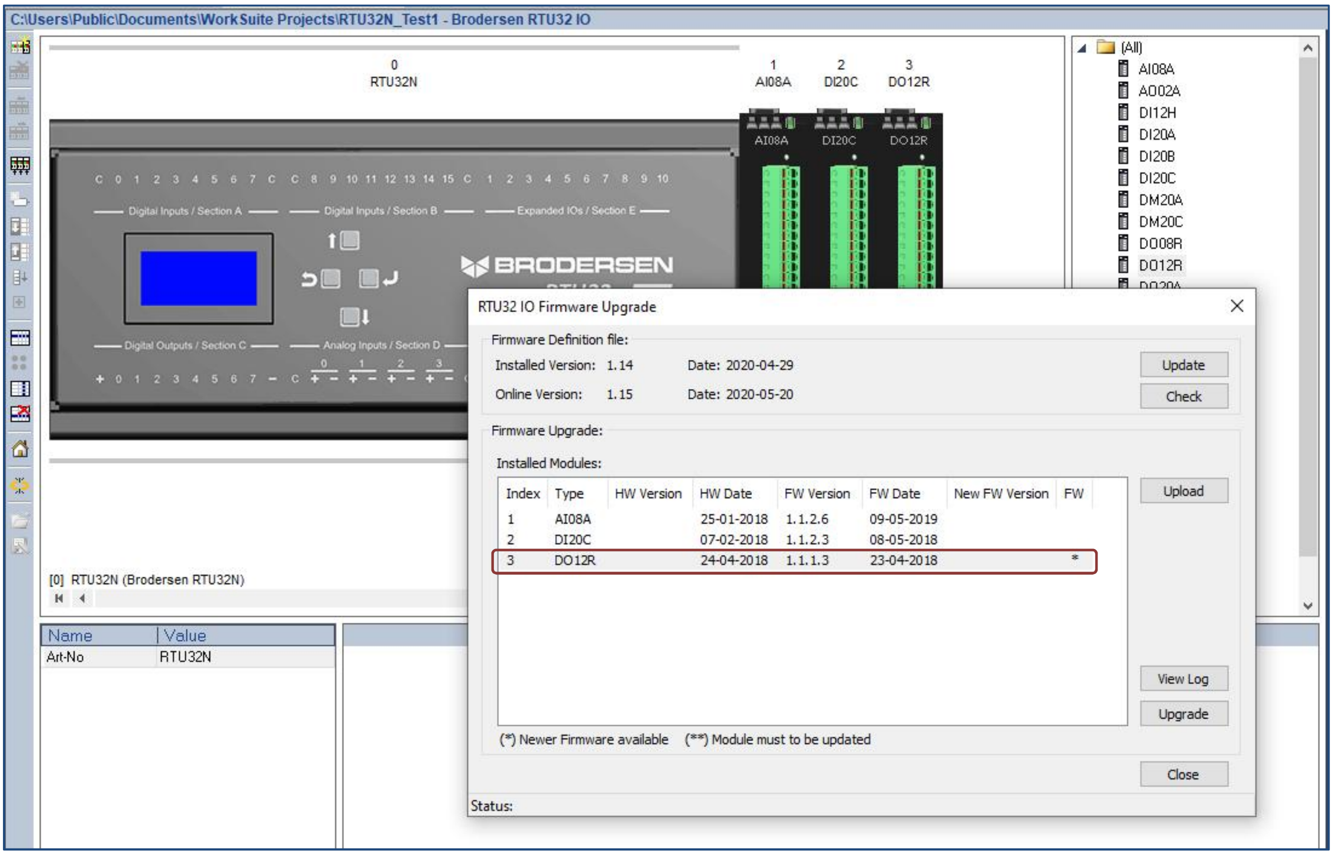


	A	B	C	D	E	F	G	H
1	NAME	TYPE	OFFSET	SCALEMIN	SCALEMAX	HWMIN	HWMAX	#F
2	4_module1	IX	0	0.0	65535.0	0	65535	-1
3	4_module2	IX	1	0.0	65535.0	0	65535	-1
4	4_module3	IX	2	0.0	65535.0	0	65535	-1
5	4_module4	IX	3	0.0	65535.0	0	65535	-1
6	4_module5	IX	4	0.0	65535.0	0	65535	-1
7	4_module6	IX	5	0.0	65535.0	0	65535	-1
8	4_module7	IX	6	0.0	65535.0	0	65535	-1
9	4_module8	IX	7	0.0	65535.0	0	65535	-1
10	4_module9	IX	8	0.0	65535.0	0	65535	-1
11	4_module10	IX	9	0.0	65535.0	0	65535	-1
12	4_module11	QX	0	0.0	65535.0	0	65535	-1
13	4_module12	QX	1	0.0	65535.0	0	65535	-1
14	4_module13	QX	2	0.0	65535.0	0	65535	-1
15	4_module14	QX	3	0.0	65535.0	0	65535	-1
16	4_module15	QX	4	0.0	65535.0	0	65535	-1
17	4_module16	QX	5	0.0	65535.0	0	65535	-1

Import: Select the Import Variables icon in the toolbar on the left to read in updates from a CSV file.

14.4.11 Firmware update I/O modules

Firmware update of modules on the LB2 bus is done through the Firmware update utility in the Brodersen WorkSuite. Select Tools>Brodersen>RTU32 IO Firmware Upgrade to access the upgrade tools. As shown below, the user interface allows uploading of the existing module firmware info and updating of individual modules. The Update and Check buttons allow updating of WorkSuite with the latest IO firmware from the Brodersen FTP server (you will require an active Internet connection to perform an update of your WorkSuite files). Any modules detected with older firmware will be marked with an asterisk to indicate there is newer firmware available (** indicating that the update is required).



C:\Users\Public\Documents\WorkSuite Projects\RTU32N_Test1 - Brodersen RTU32 IO

[0] RTU32N (Brodersen RTU32N)

Name	Value
Art-No	RTU32N

RTU32 IO Firmware Upgrade

Firmware Definition file:

Installed Version: 1.14 Date: 2020-04-29

Online Version: 1.15 Date: 2020-05-20

Firmware Upgrade:

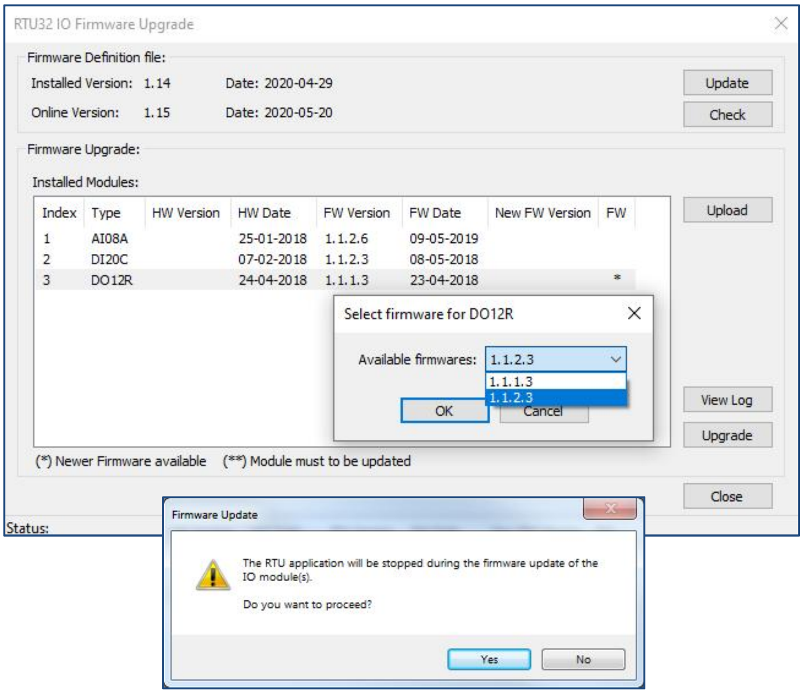
Installed Modules:

Index	Type	HW Version	HW Date	FW Version	FW Date	New FW Version	FW
1	AI08A		25-01-2018	1.1.2.6	09-05-2019		
2	DI20C		07-02-2018	1.1.2.3	08-05-2018		
3	DO12R		24-04-2018	1.1.1.3	23-04-2018		*

(*) Newer Firmware available (**) Module must to be updated

Status:

To update a module, double click on it then select the firmware and you will be prompted to continue.



RTU32 IO Firmware Upgrade

Firmware Definition file:

Installed Version: 1.14 Date: 2020-04-29

Online Version: 1.15 Date: 2020-05-20

Firmware Upgrade:

Installed Modules:

Index	Type	HW Version	HW Date	FW Version	FW Date	New FW Version	FW
1	AI08A		25-01-2018	1.1.2.6	09-05-2019		
2	DI20C		07-02-2018	1.1.2.3	08-05-2018		
3	DO12R		24-04-2018	1.1.1.3	23-04-2018		*

(*) Newer Firmware available (**) Module must to be updated

Status:

Select firmware for DO12R

Available firmwares: 1.1.2.3

1.1.1.3
1.1.2.3

Firmware Update

The RTU application will be stopped during the firmware update of the IO module(s).

Do you want to proceed?

14.5 RTU32N Onboard I/O and UCL I/O Modules

The RTU32N on board I/O and UCL style expansion I/O uses the WorkSuite I/O Profile Editor to manage the association of variables to the physical channels

I/O Profiles are flexible and easy to use and allow declaring of a single input or output and include additional scaling possibilities for analogue I/O. Profiles are also used when you want to read status information from the RTU32N and LB2 Series I/O modules.

14.5.1 I/O database layout

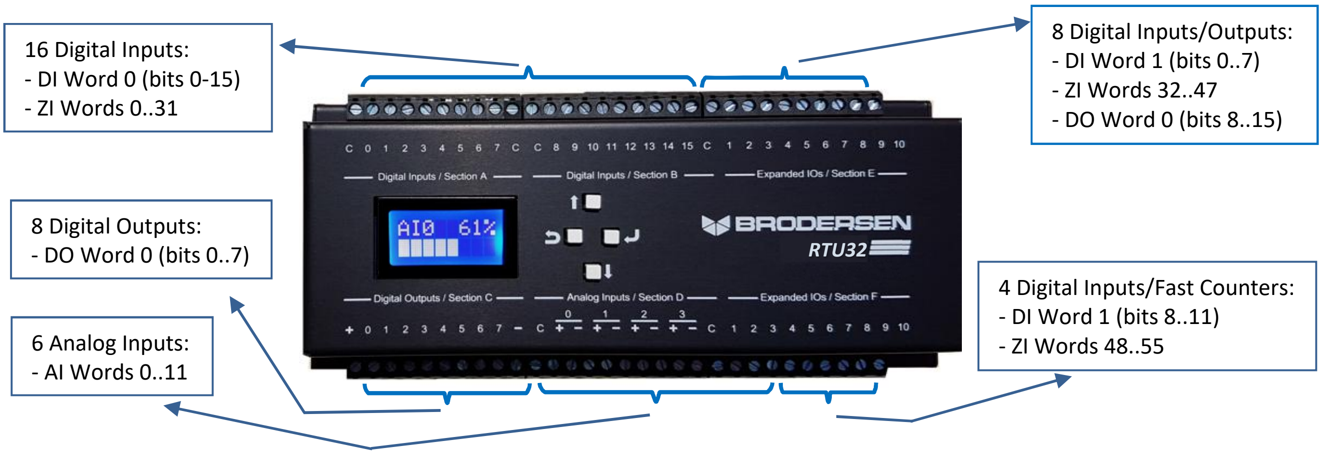
Before you start programming it is important to understand the I/O database layout and how the I/O addressing is structured. The database has basically 10 data types:

ST:	Status word for I/O	DO:	Digital outputs
DI:	Digital inputs	AO:	Analogue outputs
AI:	Analogue inputs	YO:	Virtual outputs
YI:	Virtual inputs	ZO:	Aux. outputs (incl. virtual outputs for resetting counters)
ZI:	Auxiliary inputs (includes 32bit counters)	VIO:	Virtual I/O used for database access

The virtual input and output register YI, YO and VIO are normally not used (reserved for data access to special applications accessing the basic database API). Auxiliary registers are used for special data types. Counters are read and reset through ZI and ZO registers. All data types are called **Modules** and are defined in **Word values (16bit)**. Each data type is addressed individually starting with 0 (zero). Each I/O Board has its own module layout which is defined in the table below.

I/O Boards	DI	AI	DO	AO	ZI	ZO	UCL-type
RTU32_04AO				4			UCL-04AO
RTU32_08AI			8				UCL-08AI
RTU32_08DI	1						UCL-08DI.Ax
RTU32_08DO			1				UCL-08DO.Rx
RTU32_8DIO	1		1				UCL-8DIO.Px
RTU32_16DI	1						UCL-16DI.Dx
RTU32_16DIO	1		1				UCL-16DIO.Px
RTU32_32DI	2						UCL-32DI.Dx
RTU32_32DO			2				UCL-32DO.xx
RTU32_36IO	2			1			UCL-36IO.Px
RTU32_28IO	1	4	1		2X16	1 (RC)	UCx-28IO

14.5.2 RTU32N Addressing example


C 0 1 2 3 4 5 6 7 C C 8 9 10 11 12 13 14 15 C 1 2 3 4 5 6 7 8 9 10
Digital Inputs / Section A
Digital Inputs / Section B
Expanded IOs / Section E
AI0 61%
BRODERSEN
RTU32
Digital Outputs / Section C
Analog Inputs / Section D
Expanded IOs / Section F
+ 0 1 2 3 4 5 6 7 − C + − + − + − + − C 1 2 3 4 5 6 7 8 9 10

16 Digital Inputs:
- DI Word 0 (bits 0-15)
- ZI Words 0..31

8 Digital Inputs/Outputs:
- DI Word 1 (bits 0..7)
- ZI Words 32..47
- DO Word 0 (bits 8..15)

8 Digital Outputs:
- DO Word 0 (bits 0..7)

4 Digital Inputs/Fast Counters:
- DI Word 1 (bits 8..11)
- ZI Words 48..55

6 Analog Inputs:
- AI Words 0..11

Further examples of RTU32N I/O configurations with module addressing are shown in Appendix 2 and 3.

14.5.3 Layout for the status register (ST)

In the I/O database the status word (ST) reports relevant status values:

Module 0:	I/O bus status:	
	Word value = 0 OK, no errors	
	Bit 0	Unknown LB (LocalBus I/O) module type, can't build database.
	Bit 1	LB scan error, typical one or more LB modules are missing after LB start.
	Bit 2	STRATON accessed database module number is not present.
	Bit 3	IO configuration mismatch
	Bit 4...15	Reserved
Module 1:	Reserved.	

NOTE: The I/O Status can also be read using the RTUStat Function in WorkSuite.

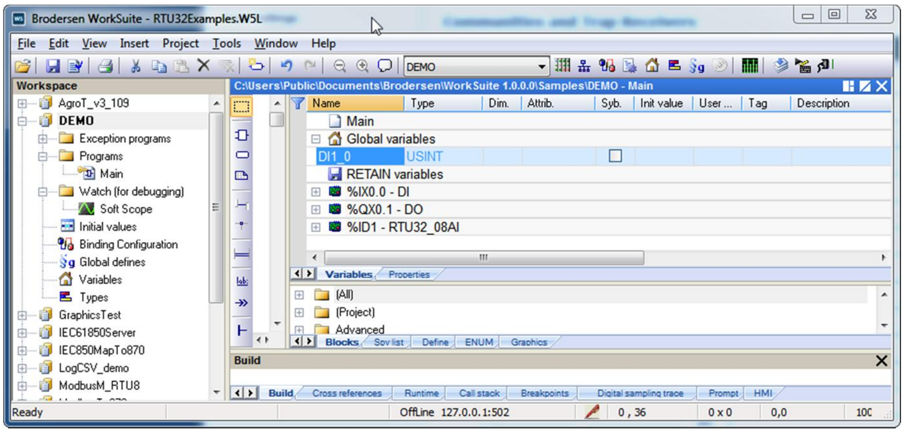
14.5.4 I/O Profiles Driver

The I/O Profile driver is used for declaring I/O one by one directly in the variable list – or by using the Profile Editor. Any digital or analogue I/O can be declared this way. If you want I/O status values (in addition to I/O data values) from the RTU32N I/O database, you must use the I/O Profile driver to access I/O.

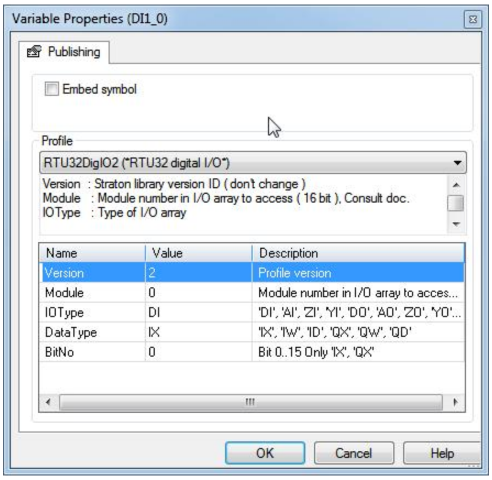
14.5.5 Option 1 - Declaring I/O Profiles in variable list

In the Variable list you can add new variables under the Global variable structure (right click and select 'Add Variable'). It is recommended to use the Global Variable structure to allow access to the I/O from any program in your WorkSuite project.

After creating a new variable, double click on the variable name and enter an appropriate name for your I/O – e.g. DI1_0 for the first digital input bit in the second DI module. Set the variable type to WORD, USINT or UDINT.


Brodersen WorkSuite - RTU32Examples.W5L
File Edit View Insert Project Tools Window Help
DEMO
Workspace
AgroT_v3_109
DEMO
Exception programs
Programs
Main
Watch (for debugging)
Soft Scope
Initial values
Binding Configuration
Global defines
Variables
Types
GraphicsTest
IEC61850Server
IEC850MapTo870
LogCSV_demo
ModbusM_RTU8
C:\Users\Public\Documents\Brodersen\WorkSuite 1.0.0.0\Samples\DEMO - Main
Name Type Dim. Attrib. Syb. Init value User... Tag Description
Main
Global variables
DI1_0 USINT
RETAIN variables
%IX0.0 - DI
%QX0.1 - DO
%ID1 - RTU32_08AI
Variables Properties
(All)
(Project)
Advanced
Blocks Sov list Define ENUM Graphics
Build
Build Cross references Runtime Call stack Breakpoints Digital sampling trace Prompt HMI
Ready OffLine 127.0.0.1:502 0 , 36 0 x 0 0,0 100

Right click on the DI1_0 variable and select Properties:


Variable Properties (DI1_0)

Profile: RTU32DigIO2 ("RTU32 digital I/O")

Version : Straton library version ID (don't change)
Module : Module number in I/O array to access (16 bit), Consult doc.
IOType : Type of I/O array

Name	Value	Description
Version	2	Profile version
Module	0	Module number in I/O array to acces...
IOType	DI	'DI', 'AI', 'ZI', 'YI', 'DO', 'AO', 'ZO', 'YO'...
DataType	IX	'IX', 'IW', 'ID', 'QX', 'QW', 'QD'
BitNo	0	Bit 0..15 Only 'IX', 'QX'

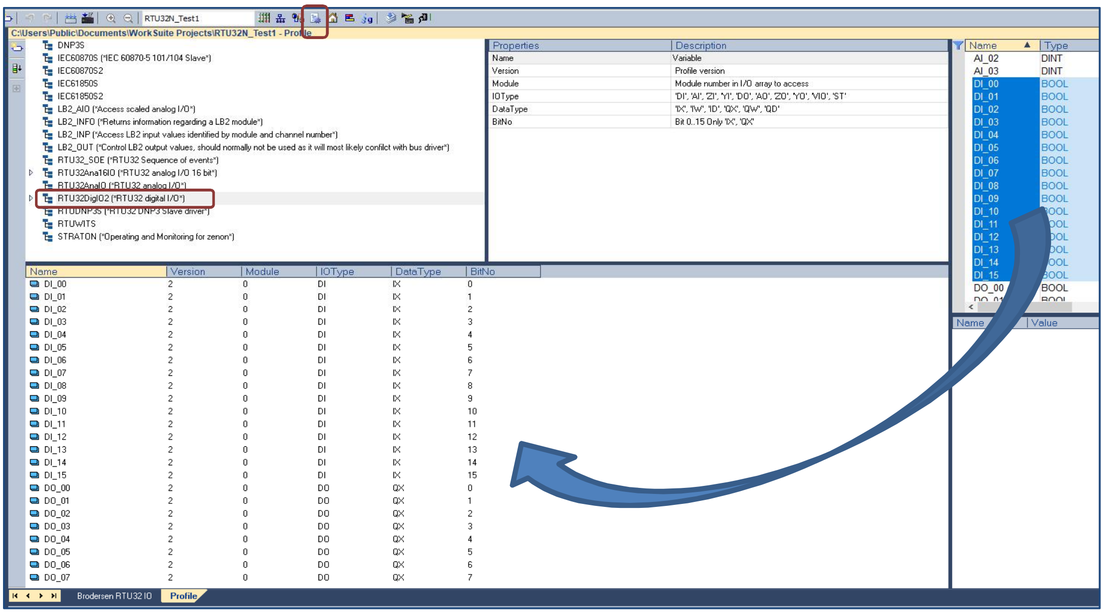
Select "RTU32DigIO" for digital I/O, set Module = 1 (second digital input module (read: Word)).
Select "DI" for Digital Input register.
Select "IW" for Input Word.
Say OK to store the settings.

Now the physical I/O is defined as a variable and can be used in your application program. If you want a single bit value you can either declare the variable as a bit (IX) or you can just write "DI1_0.2" to read the 3rd bit value.

The same procedure is used for reading the status words.

14.5.6 Option 2 - Declaring I/O Profiles using the Profile Editor (Recommended Method)

The easier/faster way to declare an I/O variable for RTU32 Series I/O is to use the Profile Editor in Brodersen WorkSuite.


RTU32N_Test1

C:\Users\Public\Documents\WorkSuite Projects\RTU32N_Test1 - Profile

- DNP3S
- IEC60870S (*IEC 60870-5 101/104 Slave*)
- IEC60870S2
- IEC61850S
- IEC61850S2
- LB2_AIO (*Access scaled analog I/O*)
- LB2_INFO (*Returns information regarding a LB2 module*)
- LB2_INP (*Access LB2 input values identified by module and channel number*)
- LB2_OUT (*Control LB2 output values, should normally not be used as it will most likely confilct with bus driver*)
- RTU32_SOE (*RTU32 Sequence of events*)
- RTU32Ana16IO (*RTU32 analog I/O 16 bit*)
- RTU32AnaIO (*RTU32 analog I/O*)
- RTU32DigIO2 (*RTU32 digital I/O*)
- RTUDNP3S (*RTU32 DNP3 Slave driver*)
- RTUWITS
- STRATON (*Operating and Monitoring for zenon*)

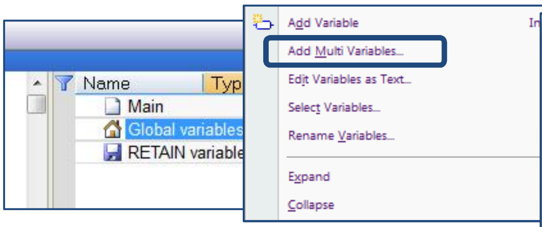
Properties	Description
Name	Variable
Version	Profile version
Module	Module number in I/O array to access
IOType	'DI', 'AI', 'ZI', 'YI', 'DO', 'AO', 'ZO', 'YO', 'VIO', 'ST'
DataType	'IX', 'IW', 'ID', 'QX', 'QW', 'QD'
BitNo	Bit 0..15 Only 'IX', 'QX'

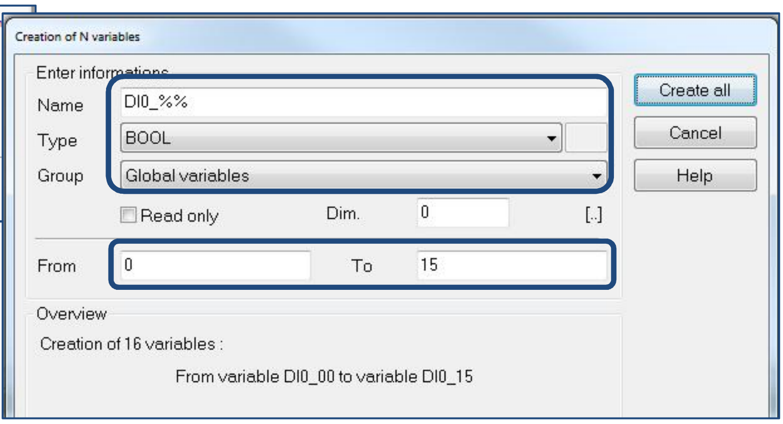
Name	Type
AI_02	DINT
AI_03	DINT
DI_00	BOOL
DI_01	BOOL
DI_02	BOOL
DI_03	BOOL
DI_04	BOOL
DI_05	BOOL
DI_06	BOOL
DI_07	BOOL
DI_08	BOOL
DI_09	BOOL
DI_10	BOOL
DI_11	BOOL
DI_12	BOOL
DI_13	BOOL
DI_14	BOOL
DI_15	BOOL
DO_00	BOOL

Name	Version	Module	IOType	DataType	BitNo
DI_00	2	0	DI	IX	0
DI_01	2	0	DI	IX	1
DI_02	2	0	DI	IX	2
DI_03	2	0	DI	IX	3
DI_04	2	0	DI	IX	4
DI_05	2	0	DI	IX	5
DI_06	2	0	DI	IX	6
DI_07	2	0	DI	IX	7
DI_08	2	0	DI	IX	8
DI_09	2	0	DI	IX	9
DI_10	2	0	DI	IX	10
DI_11	2	0	DI	IX	11
DI_12	2	0	DI	IX	12
DI_13	2	0	DI	IX	13
DI_14	2	0	DI	IX	14
DI_15	2	0	DI	IX	15
DO_00	2	0	DO	QX	0
DO_01	2	0	DO	QX	1
DO_02	2	0	DO	QX	2
DO_03	2	0	DO	QX	3
DO_04	2	0	DO	QX	4
DO_05	2	0	DO	QX	5
DO_06	2	0	DO	QX	6
DO_07	2	0	DO	QX	7

As shown in the example above, the Profile editor gives you a better overview, and it is easy to edit the declared variables. It also supports drag and drop from the variable list. This means that you can create the variables first and then drag and drop complete lists of variables to the Profile I/O as shown above. See WorkSuite Help for more details.

Example 1 – create 16x Boolean variables that will be associated to the onboard Digital Inputs of the RTU32N:




Creation of N variables

Enter informations

Name: DI0_%%

Type: BOOL

Group: Global variables

□ Read only Dim. 0 [..]

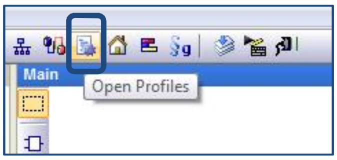
From 0 To 15

Overview

Creation of 16 variables :

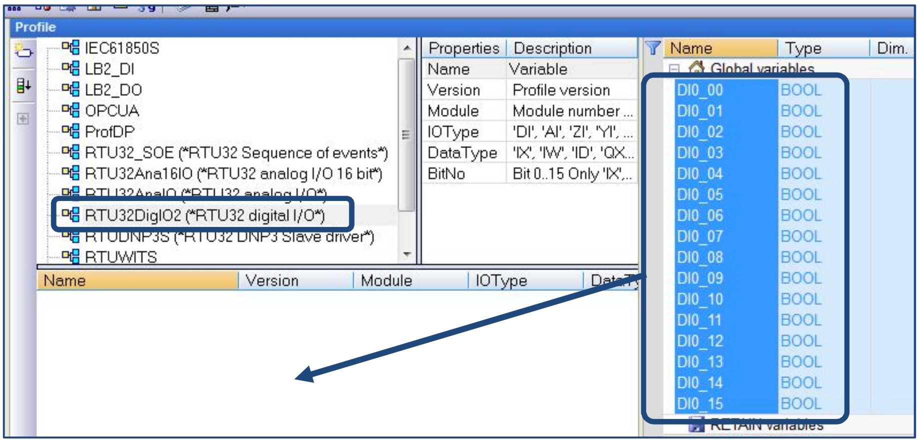
From variable DI0_00 to variable DI0_15

Right click on the Global Variables header and select 'Add Multi name and enter 0 to 15, then select 'Create All' and 16x Boolean points will be added to the variables database named DI0_00 through to DI0_15. At this point the 16x variables exist, but they have not been associated to the RTUs physical I/O.



Open the Profile editor and select the 'RTU32DigIO2' profile, then drag the 16x variables in to the blank area below the list of profiles.

By default, all points are then defined as DI points, associated to module 0, bit 0 and IEC type IX (input bit).


Profile

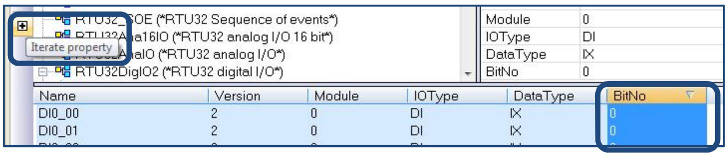
- IEC61850S
- LB2_DI
- LB2_DO
- OPCUA
- ProfDP
- RTU32_SOE (*RTU32 Sequence of events*)
- RTU32Ana16IO (*RTU32 analog I/O 16 bit*)
- RTU32AnaIO (*RTU32 analog I/O*)
- RTU32DigIO2 (*RTU32 digital I/O*)
- RTUDNP3S (*RTU32 DNP3 Slave driver*)
- RTUWITS

Properties	Description
Name	Variable
Version	Profile version
Module	Module number ...
IOType	'DI', 'AI', 'ZI', 'YI', ...
DataType	'IX', 'IW', 'ID', 'QX'...
BitNo	Bit 0..15 Only 'IX',...

Name	Version	Module	IOType	DataT

Name	Type	Dim.
Global variables		
DI0_00	BOOL	
DI0_01	BOOL	
DI0_02	BOOL	
DI0_03	BOOL	
DI0_04	BOOL	
DI0_05	BOOL	
DI0_06	BOOL	
DI0_07	BOOL	
DI0_08	BOOL	
DI0_09	BOOL	
DI0_10	BOOL	
DI0_11	BOOL	
DI0_12	BOOL	
DI0_13	BOOL	
DI0_14	BOOL	
DI0_15	BOOL	
RETAIN variables		

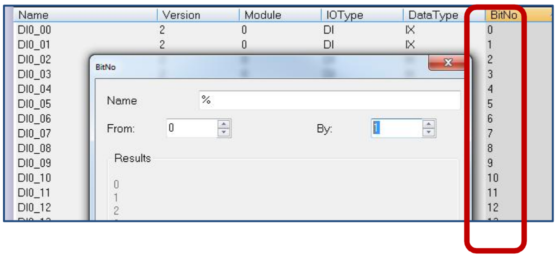
Highlight the ‘BitNo’ column, then select the ‘+’ icon to quickly iterate the bits from 0 by 1 to create bits 0-15.



Iterate property
RTU32_SOE (*RTU32 Sequence of events*)
RTU32Ana16IO (*RTU32 analog I/O 16 bit*)
RTU32AnaIO (*RTU32 analog I/O*)
RTU32DigIO2 (*RTU32 digital I/O*)

Module	0
IOType	DI
DataType	IX
BitNo	0

Name	Version	Module	IOType	DataType	BitNo
DI0_00	2	0	DI	IX	0
DI0_01	2	0	DI	IX	0



Name	Version	Module	IOType	DataType	BitNo
DI0_00	2	0	DI	IX	0
DI0_01	2	0	DI	IX	1
DI0_02					2
DI0_03					3
DI0_04					4
DI0_05					5
DI0_06					6
DI0_07					7
DI0_08					8
DI0_09					9
DI0_10					10
DI0_11					11
DI0_12					12

BitNo

Name: %

From: 0 By: 1

Results

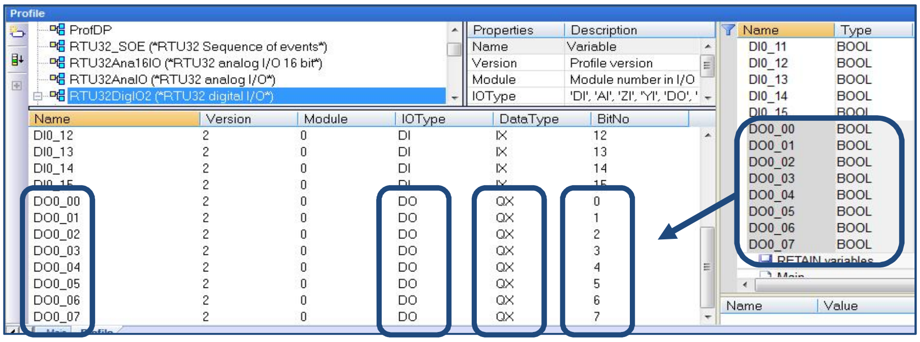
0
1
2

Example 2 – create 8x Boolean variables that will be associated to the onboard Digital Outputs of the RTU32N:

Repeat the process above, but name the variables DO0_00 through to DO0_07.

Ensure the new variables have their IO type set to ‘DO’, Data Type set to ‘QX’ and their bit numbers set to 0-7 in the IO Profile as shown here.

Note: you can highlight and modify portions of a column using ‘click’ then ‘shift-double click’ to select an alternate IO or Data type.



Profile
ProfDP
RTU32_SOE (*RTU32 Sequence of events*)
RTU32Ana16IO (*RTU32 analog I/O 16 bit*)
RTU32AnaIO (*RTU32 analog I/O*)
RTU32DigIO2 (*RTU32 digital I/O*)

Properties	Description
Name	Variable
Version	Profile version
Module	Module number in I/O
IOType	'DI', 'AI', 'ZI', 'YI', 'DO', '

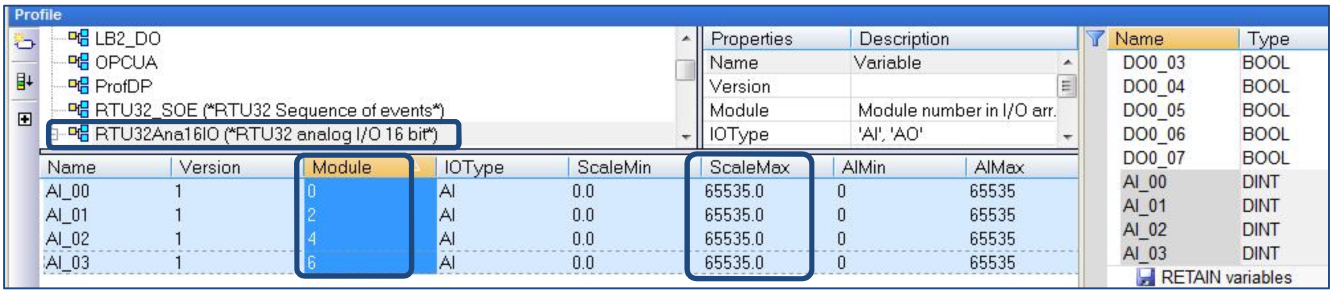
Name	Version	Module	IOType	DataType	BitNo
DI0_12	2	0	DI	IX	12
DI0_13	2	0	DI	IX	13
DI0_14	2	0	DI	IX	14
DI0_15	2	0	DI	IX	15
DO0_00	2	0	DO	QX	0
DO0_01	2	0	DO	QX	1
DO0_02	2	0	DO	QX	2
DO0_03	2	0	DO	QX	3
DO0_04	2	0	DO	QX	4
DO0_05	2	0	DO	QX	5
DO0_06	2	0	DO	QX	6
DO0_07	2	0	DO	QX	7

Name	Type
DI0_11	BOOL
DI0_12	BOOL
DI0_13	BOOL
DI0_14	BOOL
DI0_15	BOOL
DO0_00	BOOL
DO0_01	BOOL
DO0_02	BOOL
DO0_03	BOOL
DO0_04	BOOL
DO0_05	BOOL
DO0_06	BOOL
DO0_07	BOOL

Example 3 – create 4x Analog variables that will be associated to the onboard Analog Inputs of the RTU32N:

Repeat the point creation process used previously but choose DINT rather than Boolean and name the points AI_00 through to AI_03.

Ensure that the ‘RTU32Ana16IO’ profile is used and change the module address used to be offset by 2 using the iteration tool, as shown below.



Profile
LB2_DO
OPCUA
ProfDP
RTU32_SOE (*RTU32 Sequence of events*)
RTU32Ana16IO (*RTU32 analog I/O 16 bit*)

Properties	Description
Name	Variable
Version	
Module	Module number in I/O arr.
IOType	'AI', 'AO'

Name	Version	Module	IOType	ScaleMin	ScaleMax	AIMin	AIMax
AI_00	1	0	AI	0.0	65535.0	0	65535
AI_01	1	2	AI	0.0	65535.0	0	65535
AI_02	1	4	AI	0.0	65535.0	0	65535
AI_03	1	6	AI	0.0	65535.0	0	65535

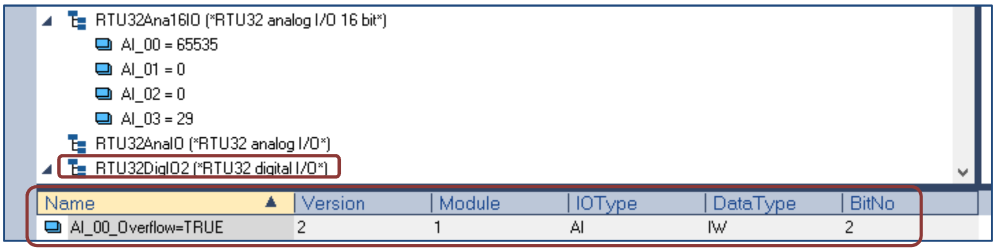
Name	Type
DO0_03	BOOL
DO0_04	BOOL
DO0_05	BOOL
DO0_06	BOOL
DO0_07	BOOL
AI_00	DINT
AI_01	DINT
AI_02	DINT
AI_03	DINT

The RTU32N models BRS-xxIO/1A005.81 and .91 have 16bit analog values, that are stored as a 32bit value (2x module WORDS). The first 16bit Module WORD eg. Module WORD 0 contains the measured value for the first AI channel. The second 16bit Module WORD eg. Module WORD 1 contains the status flags for the first AI channel. The measured value can be scaled 0-65535 (default), or to alternative values such as 0-16383 (compatible with RTU32 controllers), or 0-10000 (preferred by some end users as simple percentage representation) by changing the Scale Min and ScaleMax values.

14.5.7 RTU32N Analog Input Status (16bits for each AI channel)

The example below shows the Overflow status of AI ch 0 as a Boolean variable that is included in the RTU32DigIO2 profile. The variable ‘AI_00_Overflow’ is referencing Module Word 1, IOType=AI, DataType=IW (Input Word) and Bit 2 (Overflow status bit).

Analog Input Status Word:
Bit 0: OK
Bit 1: Underflow
Bit 2: Overflow
Bit 3: Invalid data
Bit 4: Disabled
Bits 5..65535: Reserved



RTU32Ana16IO (*RTU32 analog I/O 16 bit*)
AI_00 = 65535
AI_01 = 0
AI_02 = 0
AI_03 = 29
RTU32AnaIO (*RTU32 analog I/O*)
RTU32DigIO2 (*RTU32 digital I/O*)

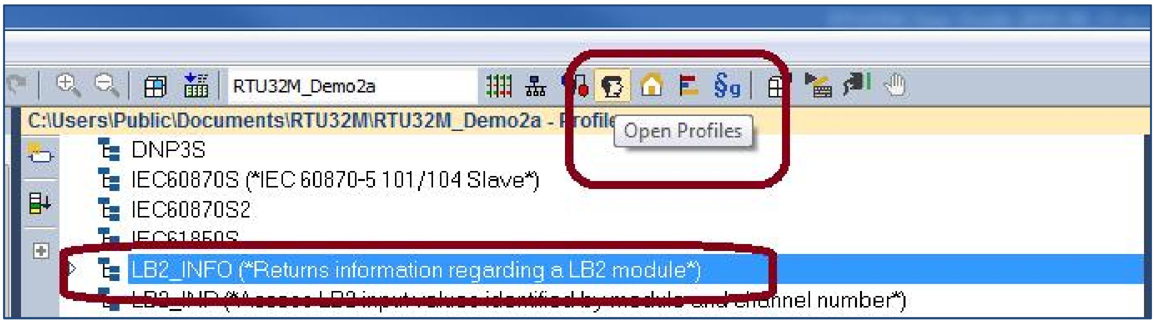
Name	Version	Module	IOType	DataType	BitNo
AI_00_Overflow=TRUE	2	1	AI	IW	2

14.5.8 Counter inputs

The RTU32N I/O board digital inputs have support for counter inputs. Auxiliary inputs (ZI) are used for reading 32bit counter inputs. The counter values are stored in Module WORDS (2 per counter) and can be reset by using a virtual output (ZO). Reset of counters is done on an output rising edge. Make sure that the reset output is held ON to ensure that that one I/O scan has been performed. This can be done by using a timer in the WorkSuite application program.

14.5.9 LB2_INFO Profile – monitoring module status

LB2 Module status information is available via the Profile Editor, using the LB2_INFO Profile.



C:\Users\Public\Documents\RTU32M\RTU32M_Demo2a - Profile

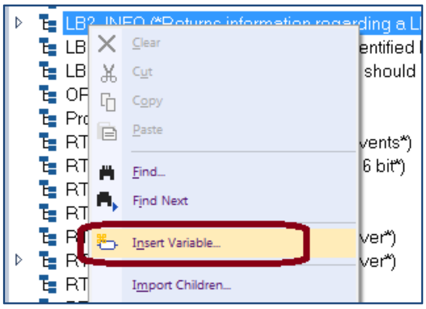
DNP3S
IEC60870S (*IEC 60870-5 101/104 Slave*)
IEC60870S2
IEC61850S
LB2_INFO (*Returns information regarding a LB2 module*)
LB2_INP (*Access LB2 input values identified by module and channel number*)

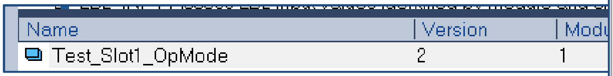
The LB2_INFO profile allows access to module parameters and status information including:

Parameter Name	Variable Type	Module Type
OP:	BYTE	Current Operation Mode
		0 = No Module
		4 = Stopped
		5 = Operational
		6 = Operational with heartbeat timeout
		127 = Pre-operational
		150 = Missing
HBCNT:	DWORD	Number of heratbeat frames received
RXCNT:	DWORD	Number of frames received
EMCY:	LINT	Read last EMCY codes
HWPID:	DWORD	Module hardware ID
SWPID:	DWORD	Module firmware ID
SID1:	DWORD	Module serial ID1
SID2:	DWORD	Module serial ID2
SID3:	DWORD	Module serial ID3
SID4:	DWORD	Module serial ID4
MainV:	WORD	Main supply voltage (mV)
SuppV:	WORD	Internal supply voltage (mV)
IntC:	WORD	Internal temperature (C)
PwOn#:	WORD	Power on counter
WD#:	WORD	Watchdog restart counter
UpTim:	DWORD	Uptime in hours
TotUpTim: DWORD	Total uptime	

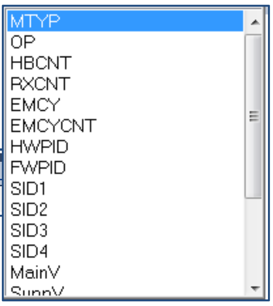
LB2_INFO Profile – Allows access to RTU32M Module Information

To use the Profile Editor – first create a variable by right clicking the LB2_INFO heading and selecting 'Insert Variable'. Eg. create a variable named 'Test_Slot1_OpMode', and give it a BYTE format, then select the 'OP' data type.





Name	Version	Modu
Test_Slot1_OpMode	2	1

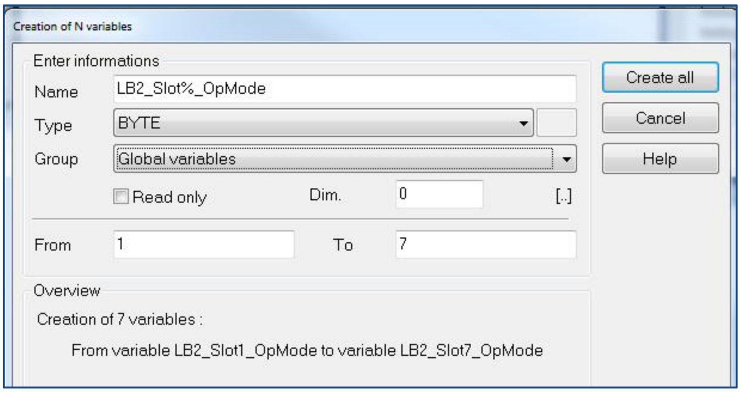


Load your application and check that the module returns the expected operating code (5 = operational).

Note: you can create more local or global variables using the Wizard tool in the main variables editor eg. right click in the variables list and select 'Add Multi Variables'.

The example here creates 7x variables named LB2_Slot1_OpMode thru to LB2_Slot7_OpMode…

After creating each group of new variables they can then be dragged in to the Profile editor and assigned their parameters and setting using column change and iterate functions…



Creation of N variables

Enter informations

Name: LB2_Slot%_OpMode

Type: BYTE

Group: Global variables

☐ Read only   Dim. 0 [..]

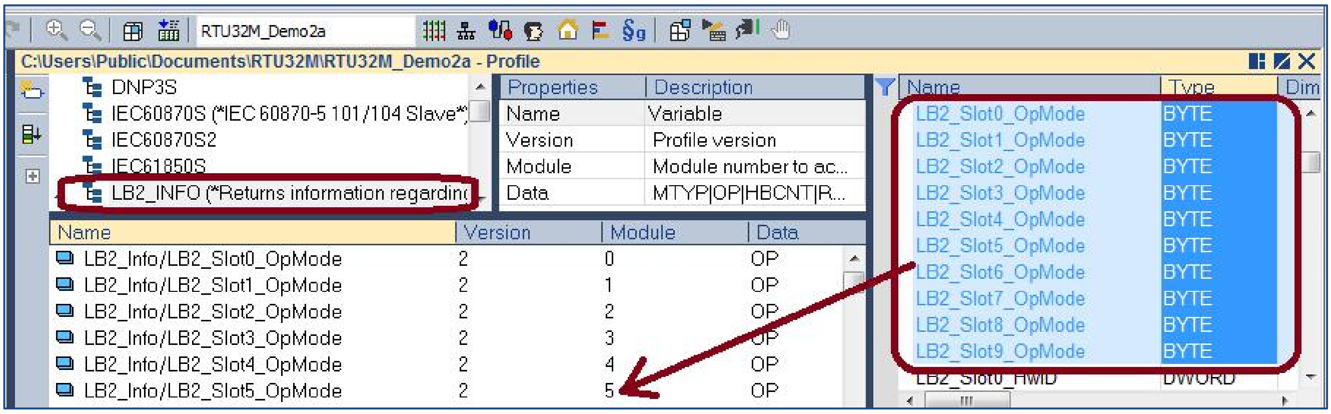
From 1   To 7

Overview

Creation of 7 variables :

From variable LB2_Slot1_OpMode to variable LB2_Slot7_OpMode

Then first ensure that the correct profile is highlighted, by selecting the header (LB2_INFO), then drag the new variables into the lower pane of the profile editor.



C:\Users\Public\Documents\RTU32M\RTU32M_Demo2a - Profile

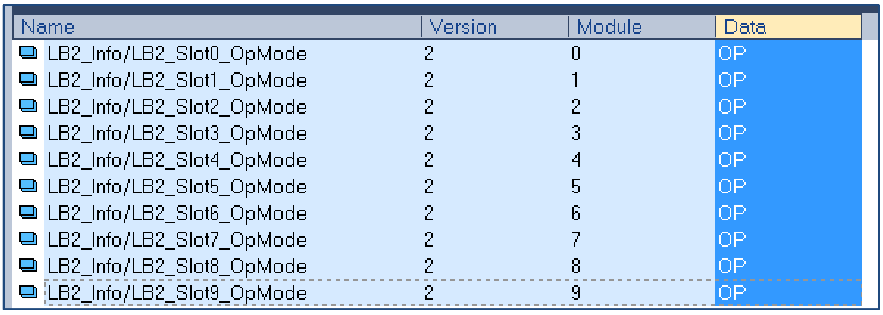
RTU32M_Demo2a
DNP3S
IEC60870S (*IEC 60870-5 101/104 Slave*)
IEC60870S2
IEC61850S
LB2_INFO (*Returns information regarding

Properties	Description
Name	Variable
Version	Profile version
Module	Module number to ac...
Data	MTYP|OP|HBCNT|R...

Name	Version	Module	Data
LB2_Info/LB2_Slot0_OpMode	2	0	OP
LB2_Info/LB2_Slot1_OpMode	2	1	OP
LB2_Info/LB2_Slot2_OpMode	2	2	OP
LB2_Info/LB2_Slot3_OpMode	2	3	OP
LB2_Info/LB2_Slot4_OpMode	2	4	OP
LB2_Info/LB2_Slot5_OpMode	2	5	OP

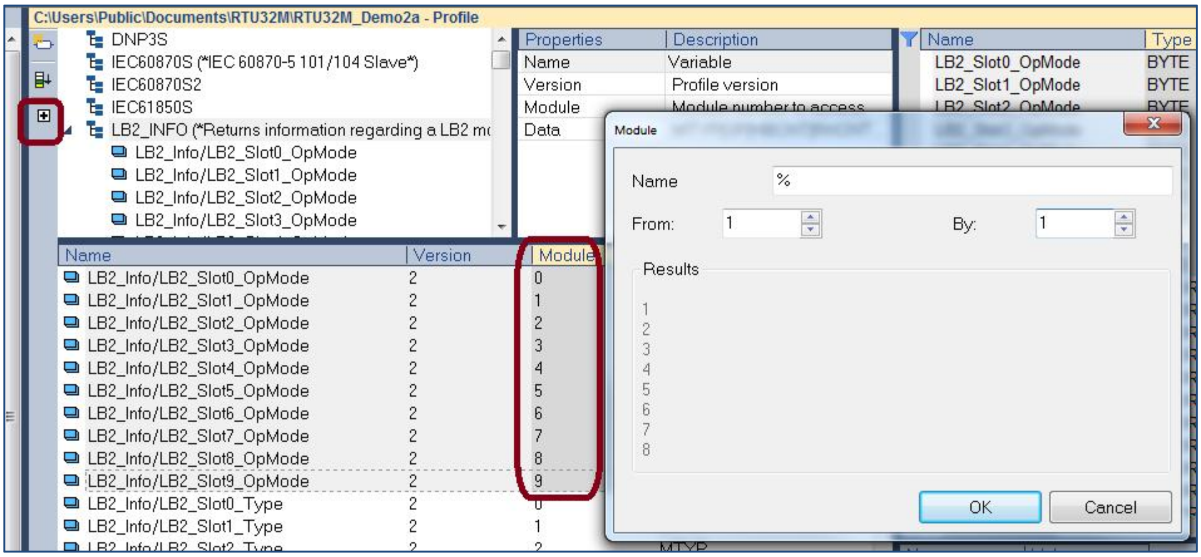
Name	Type	Dim
LB2_Slot0_OpMode	BYTE	
LB2_Slot1_OpMode	BYTE	
LB2_Slot2_OpMode	BYTE	
LB2_Slot3_OpMode	BYTE	
LB2_Slot4_OpMode	BYTE	
LB2_Slot5_OpMode	BYTE	
LB2_Slot6_OpMode	BYTE	
LB2_Slot7_OpMode	BYTE	
LB2_Slot8_OpMode	BYTE	
LB2_Slot9_OpMode	BYTE	
LB2_Slot0_HWID	DWORD	

To make a column contents change of the Data column, select the first value, then use SHIFT-Double Left Click to bring up the selection window. Select OP for Operating mode to change all of the selected values at once.



Name	Version	Module	Data
LB2_Info/LB2_Slot0_OpMode	2	0	OP
LB2_Info/LB2_Slot1_OpMode	2	1	OP
LB2_Info/LB2_Slot2_OpMode	2	2	OP
LB2_Info/LB2_Slot3_OpMode	2	3	OP
LB2_Info/LB2_Slot4_OpMode	2	4	OP
LB2_Info/LB2_Slot5_OpMode	2	5	OP
LB2_Info/LB2_Slot6_OpMode	2	6	OP
LB2_Info/LB2_Slot7_OpMode	2	7	OP
LB2_Info/LB2_Slot8_OpMode	2	8	OP
LB2_Info/LB2_Slot9_OpMode	2	9	OP

The iterate property button will allow you to rapidly increment the module address for each tag. First select the relevant tags in the module column by selecting the first value, then SHIFT-Left click the last value. Then select the 'Iterate Property' button and set the appropriate start/from value and increment/by value.



C:\Users\Public\Documents\RTU32M\RTU32M_Demo2a - Profile

- DNP3S
- IEC60870S (*IEC 60870-5 101/104 Slave*)
- IEC60870S2
- IEC61850S
- LB2_INFO (*Returns information regarding a LB2 m
  - LB2_Info/LB2_Slot0_OpMode
  - LB2_Info/LB2_Slot1_OpMode
  - LB2_Info/LB2_Slot2_OpMode
  - LB2_Info/LB2_Slot3_OpMode

Properties	Description
Name	Variable
Version	Profile version
Module	Module number to access
Data	

Name	Type
LB2_Slot0_OpMode	BYTE
LB2_Slot1_OpMode	BYTE
LB2_Slot2_OpMode	BYTE

Name	Version	Module
LB2_Info/LB2_Slot0_OpMode	2	0
LB2_Info/LB2_Slot1_OpMode	2	1
LB2_Info/LB2_Slot2_OpMode	2	2
LB2_Info/LB2_Slot3_OpMode	2	3
LB2_Info/LB2_Slot4_OpMode	2	4
LB2_Info/LB2_Slot5_OpMode	2	5
LB2_Info/LB2_Slot6_OpMode	2	6
LB2_Info/LB2_Slot7_OpMode	2	7
LB2_Info/LB2_Slot8_OpMode	2	8
LB2_Info/LB2_Slot9_OpMode	2	9
LB2_Info/LB2_Slot0_Type	2	0
LB2_Info/LB2_Slot1_Type	2	1
LB2_Info/LB2_Slot2_Type	2	2

Module

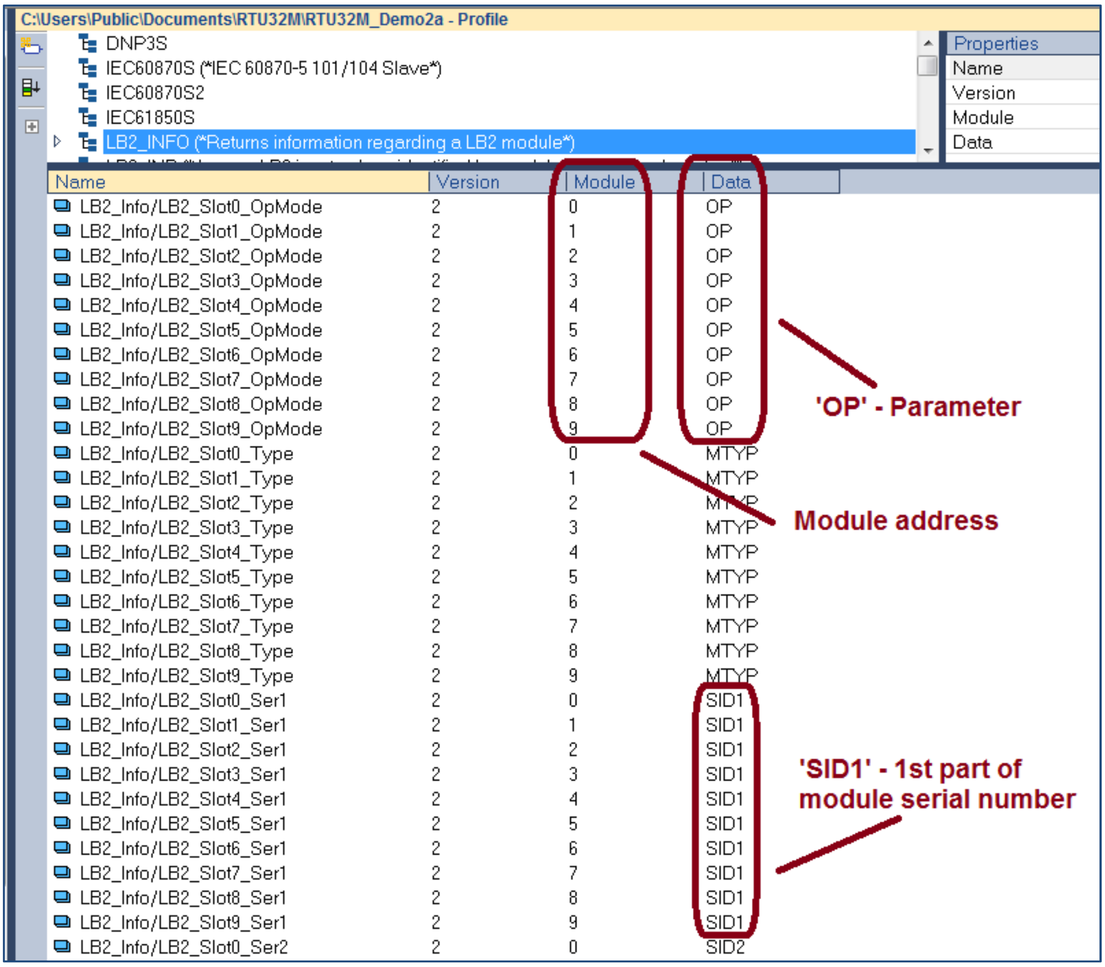
Name: %

From: 1   By: 1

Results

1
2
3
4
5
6
7
8

If you want to monitor additional module parameters and status information, repeat the above steps and create further variables, as per the example shown below.



C:\Users\Public\Documents\RTU32M\RTU32M_Demo2a - Profile

- DNP3S
- IEC60870S (*IEC 60870-5 101/104 Slave*)
- IEC60870S2
- IEC61850S
- LB2_INFO (*Returns information regarding a LB2 module*)

Properties
Name
Version
Module
Data

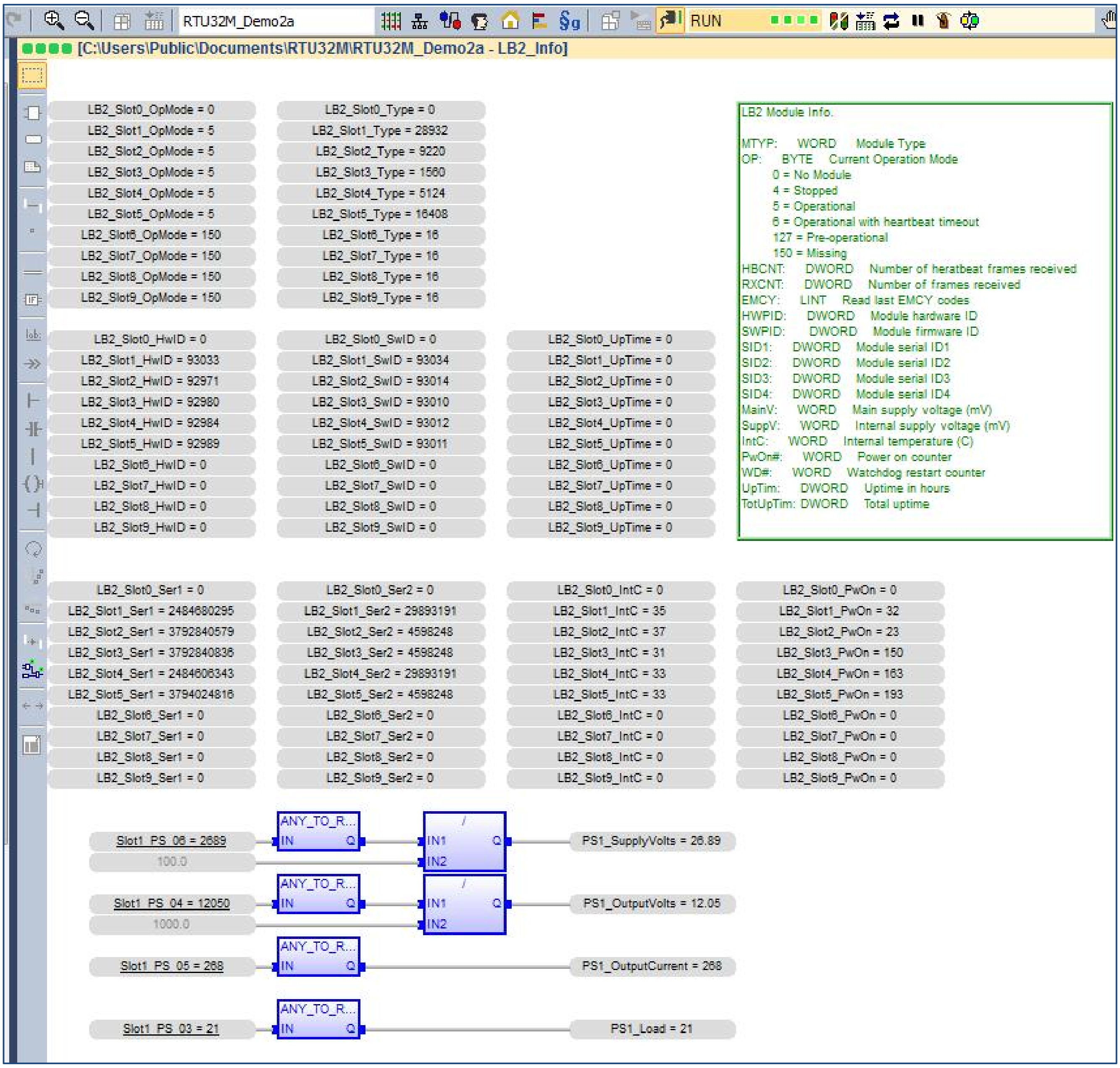
Name	Version	Module	Data
LB2_Info/LB2_Slot0_OpMode	2	0	OP
LB2_Info/LB2_Slot1_OpMode	2	1	OP
LB2_Info/LB2_Slot2_OpMode	2	2	OP
LB2_Info/LB2_Slot3_OpMode	2	3	OP
LB2_Info/LB2_Slot4_OpMode	2	4	OP
LB2_Info/LB2_Slot5_OpMode	2	5	OP
LB2_Info/LB2_Slot6_OpMode	2	6	OP
LB2_Info/LB2_Slot7_OpMode	2	7	OP
LB2_Info/LB2_Slot8_OpMode	2	8	OP
LB2_Info/LB2_Slot9_OpMode	2	9	OP
LB2_Info/LB2_Slot0_Type	2	0	MTYP
LB2_Info/LB2_Slot1_Type	2	1	MTYP
LB2_Info/LB2_Slot2_Type	2	2	MTYP
LB2_Info/LB2_Slot3_Type	2	3	MTYP
LB2_Info/LB2_Slot4_Type	2	4	MTYP
LB2_Info/LB2_Slot5_Type	2	5	MTYP
LB2_Info/LB2_Slot6_Type	2	6	MTYP
LB2_Info/LB2_Slot7_Type	2	7	MTYP
LB2_Info/LB2_Slot8_Type	2	8	MTYP
LB2_Info/LB2_Slot9_Type	2	9	MTYP
LB2_Info/LB2_Slot0_Ser1	2	0	SID1
LB2_Info/LB2_Slot1_Ser1	2	1	SID1
LB2_Info/LB2_Slot2_Ser1	2	2	SID1
LB2_Info/LB2_Slot3_Ser1	2	3	SID1
LB2_Info/LB2_Slot4_Ser1	2	4	SID1
LB2_Info/LB2_Slot5_Ser1	2	5	SID1
LB2_Info/LB2_Slot6_Ser1	2	6	SID1
LB2_Info/LB2_Slot7_Ser1	2	7	SID1
LB2_Info/LB2_Slot8_Ser1	2	8	SID1
LB2_Info/LB2_Slot9_Ser1	2	9	SID1
LB2_Info/LB2_Slot0_Ser2	2	0	SID2

'OP' - Parameter

Module address

'SID1' - 1st part of module serial number

Example Logic Application Page – showing various LB2_INFO parameters and status values.



```
LB2_Slot0_OpMode = 0        LB2_Slot0_Type = 0
LB2_Slot1_OpMode = 5        LB2_Slot1_Type = 28932
LB2_Slot2_OpMode = 5        LB2_Slot2_Type = 9220
LB2_Slot3_OpMode = 5        LB2_Slot3_Type = 1560
LB2_Slot4_OpMode = 5        LB2_Slot4_Type = 5124
LB2_Slot5_OpMode = 5        LB2_Slot5_Type = 16408
LB2_Slot6_OpMode = 150      LB2_Slot6_Type = 16
LB2_Slot7_OpMode = 150      LB2_Slot7_Type = 16
LB2_Slot8_OpMode = 150      LB2_Slot8_Type = 16
LB2_Slot9_OpMode = 150      LB2_Slot9_Type = 16

LB2_Slot0_HwID = 0          LB2_Slot0_SwID = 0          LB2_Slot0_UpTime = 0
LB2_Slot1_HwID = 93033      LB2_Slot1_SwID = 93034      LB2_Slot1_UpTime = 0
LB2_Slot2_HwID = 92971      LB2_Slot2_SwID = 93014      LB2_Slot2_UpTime = 0
LB2_Slot3_HwID = 92980      LB2_Slot3_SwID = 93010      LB2_Slot3_UpTime = 0
LB2_Slot4_HwID = 92984      LB2_Slot4_SwID = 93012      LB2_Slot4_UpTime = 0
LB2_Slot5_HwID = 92989      LB2_Slot5_SwID = 93011      LB2_Slot5_UpTime = 0
LB2_Slot6_HwID = 0          LB2_Slot6_SwID = 0          LB2_Slot6_UpTime = 0
LB2_Slot7_HwID = 0          LB2_Slot7_SwID = 0          LB2_Slot7_UpTime = 0
LB2_Slot8_HwID = 0          LB2_Slot8_SwID = 0          LB2_Slot8_UpTime = 0
LB2_Slot9_HwID = 0          LB2_Slot9_SwID = 0          LB2_Slot9_UpTime = 0
```

```
LB2 Module Info.

MTYP:     WORD     Module Type
OP:     BYTE    Current Operation Mode
        0 = No Module
        4 = Stopped
        5 = Operational
        6 = Operational with heartbeat timeout
        127 = Pre-operational
        150 = Missing
HBCNT:     DWORD    Number of heratbeat frames received
RXCNT:     DWORD    Number of frames received
EMCY:     LINT    Read last EMCY codes
HWPID:     DWORD    Module hardware ID
SWPID:     DWORD    Module firmware ID
SID1:     DWORD    Module serial ID1
SID2:     DWORD    Module serial ID2
SID3:     DWORD    Module serial ID3
SID4:     DWORD    Module serial ID4
MainV:     WORD    Main supply voltage (mV)
SuppV:     WORD    Internal supply voltage (mV)
IntC:     WORD    Internal temperature (C)
PwOn#:     WORD    Power on counter
WD#:     WORD    Watchdog restart counter
UpTim:     DWORD    Uptime in hours
TotUpTim: DWORD    Total uptime
```

```
LB2_Slot0_Ser1 = 0             LB2_Slot0_Ser2 = 0          LB2_Slot0_IntC = 0     LB2_Slot0_PwOn = 0
LB2_Slot1_Ser1 = 2484680295    LB2_Slot1_Ser2 = 29893191   LB2_Slot1_IntC = 35    LB2_Slot1_PwOn = 32
LB2_Slot2_Ser1 = 3792840579    LB2_Slot2_Ser2 = 4598248    LB2_Slot2_IntC = 37    LB2_Slot2_PwOn = 23
LB2_Slot3_Ser1 = 3792840836    LB2_Slot3_Ser2 = 4598248    LB2_Slot3_IntC = 31    LB2_Slot3_PwOn = 150
LB2_Slot4_Ser1 = 2484606343    LB2_Slot4_Ser2 = 29893191   LB2_Slot4_IntC = 33    LB2_Slot4_PwOn = 163
LB2_Slot5_Ser1 = 3794024816    LB2_Slot5_Ser2 = 4598248    LB2_Slot5_IntC = 33    LB2_Slot5_PwOn = 193
LB2_Slot6_Ser1 = 0             LB2_Slot6_Ser2 = 0          LB2_Slot6_IntC = 0     LB2_Slot6_PwOn = 0
LB2_Slot7_Ser1 = 0             LB2_Slot7_Ser2 = 0          LB2_Slot7_IntC = 0     LB2_Slot7_PwOn = 0
LB2_Slot8_Ser1 = 0             LB2_Slot8_Ser2 = 0          LB2_Slot8_IntC = 0     LB2_Slot8_PwOn = 0
LB2_Slot9_Ser1 = 0             LB2_Slot9_Ser2 = 0          LB2_Slot9_IntC = 0     LB2_Slot9_PwOn = 0
```

```
Slot1_PS_06 = 2689     PS1_SupplyVolts = 26.89
100.0
Slot1_PS_04 = 12050    PS1_OutputVolts = 12.05
1000.0
Slot1_PS_05 = 268      PS1_OutputCurrent = 268
Slot1_PS_03 = 21       PS1_Load = 21
```

14.6 COM Port Settings

When using serial drivers the COM port parameters are defined as “COM1:9600,N,8,1”. In addition, some specific control and debug parameters can be added for the RTU32N.

The COM port setting strings follow the normal WorkSuite conventions. COMa:b,c,d,e (e.g. COM1:9600,N,8,1).

a) COM port number:	1...3
b) Baud rate:	300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200
c) Byte size:	7, 8 (bit)
d) Parity	N, E, O (none, even, odd)
e) Stop bits	1, 2

A range of additional parameters on the COM port settings are supported to control:

- Setup the COM port for modem dial support – enable the Modem driver.
- Hardware handshake signal on null modem driver.
- Provide communication log in the general RTU32N System Log.

The extensions for these functions and features are:

- When using the modem driver a ‘M’ is added (e.g. MCOM1:9600,N,8,1)
- RTS / CTS hardware handshake control on the null modem serial driver is supported and enabled by the options as follows:
  ‘RD’ RTS is kept inactive (low) at all time.
  ‘RE’ RTS is kept active (high) at all time.
  ‘RT:Leading:Trailing’ RTS is inactive when receiving data, and becomes active when transmitting data. The RTS Leading setting defines the delay from activating the RTS to when the first character is transmitted. The RTS Trailing setting defines the delay from when the last character is transmitted to RTS is deactivated.
  Leading and trailing time is in units of msec. - e.g. COM1:9600,N,8,1,RT:50:10

  ‘RC:Leading:Trailing’ RTS is inactive when receiving data, and is activated when the RTU wants to transmit data. After activating RTS, the RTU will wait for CTS to become active, before start transmitting. The RTS Leading delay is still valid in this mode, and an adjustable delay from CTS activated to first character is then possible. However, by setting the Leading time to zero there is no unnecessary delay from CTS to first character (like a normal RTS / CTS function). After activating RTS the RTU waits up to 10 sec for the CTS signal. If timeout occurs, the transmission is discarded. The RTS Trailing setting defines the delay from when the last character is transmitted to RTS deactivated.
- Logging of serial data in the System Log.

  ‘LL’ Log Link data. All bytes transmitted and received on the COM port, are logged in the internal log message system, and are available using telnet at port 911. This is primarily for debug purpose.
  Examples: MCOM3:9600,N,8,1,LL
  COM2:57600,E,8,1,RT:30:10,LL

Notes:

If you just connect a serial device, e.g. ModbusRTU communication via a Null Modem cable to the RS232 COM port, it is NOT required to add any special extensions.
Brodersen WorkSuite has a limitation of max 31 chars in the settings string.

If you enable the SP5 (Secure Protocol for remote communication) on the RTU32 Webpage, you must set the COM parameters settings for the specific port on the webpage. If the SP5 setting is enabled, the COM settings overrule any later settings in WorkSuite.

The serial port settings use the same syntax no matter where you configure it.

14.7 Modbus Drivers

The RTU32N PLC runtime supports function for several types of Modbus protocols:

- ModbusRTU Master and Slave
- ModbusASCII Slave
- ModbusTCP Client and Server

The drivers are setup using the Fieldbus Configurator in WorkSuite. Refer to WorkSuite training and programming examples. You will find support for ModbusUDP, and Function Block managed Modbus drivers in the WorkSuite help.

14.7.1 ModbusRTU Master

The RTU32N supports several Master/Client drivers. You can in fact setup as many Master drivers as you have COM ports available. The Modbus Master/Client is setup using the Fieldbus Configurator. See the WorkSuite help for details.

14.7.2 ModbusRTU Slave

The RTU32N supports one or more Slave/Server drivers. You must add a MBSLAVERTU function block in the program logic to define the Modbus Slave address and port settings. You can find the function block in the “Advanced” function block folder. All variable assignments, addressing etc. are done in the Fieldbus Configurator. See WorkSuite for details. When more than one Slave/Server is required – use the MBSLAVERTUEX function block to include the relevant server ID (defined in the Fieldbus port settings).

14.8 IEC60870-5-101/103/104 Drivers

The Utility protocols IEC60870-5-101, 103 and -104 are implemented as a basic link driver in the RTU32N. The application layer which includes all the specific handling of request and events communication is managed in the RTU application program. A range of WorkSuite functions and example are provided for the different protocol functions and ASDU handling.

WorkSuite include drivers for:

- IEC60870-5-101 Master
- IEC60870-5-101 Slave
- IEC60870-5-103 Master
- IEC60870-5-104 Server
- IEC60870-5-104 Client

The basic link drivers include handles for the link driver where you can define buffers, link address etc. The fact that the serial link is similar to the IEC60870-5-103, provides the option to implement application layer support for this protocol as well.

To use the IEC60870 driver, you must use the RTU32 Series Code Generator tool. This is used for simple configuration of any IEC60870 Driver.

14.9 Watchdog

The RTU32N has two built-in watchdog functions – all managed by a separate CPU on the interface board.

The system watchdog is monitoring all internal processes including the OS and communication processes between consoles and databases. In case of error, the watchdog will wait 30sec for the error to be either self-healed or the process to be restored. If this is not happening the watchdog will boot the RTU.

The application watchdog is monitoring the RTU/PLC runtime applications including all related tasks such as drivers. If any of the related tasks fail, stop or freeze for more than 10sec, the application watchdog will boot the RTU.

14.10 Real time Clock

The RTU32N has a hardware real time clock controlled by the operating system. It provides the RTU32N millisecond resolution in order to make accurate time stamps. From the RTU application program you have access to the real time clock via logic function blocks. You can both read and set the clock.

On the configuration webpages you can also define how to synchronise the clock. Options for manual setup and SNTP Server synchronisation are possible. If you use utility drivers (eg. DNP3 and IEC61850) to synchronise the RTC you must disable SNTP on webpage by setting it to manual sync.

14.11 Modem Functions

WorkSuite provides a range of modem control Functions. The DialUp library includes:

- Functions to control and monitor PSTN/GSM modem dial-in/out

14.11.1 Dial functions for GSM and PSTN Modems

Dial functions for communication with serial drivers over PSTN and GSM modems are provided.

The low level modem handling (send AT commands, connect detection etc.) is done by firmware. However, it is up to the application programmer to decide which protocol driver should be activated, when and where to dial, and respond to the modem connection state. A number of WorkSuite functions are available for this:

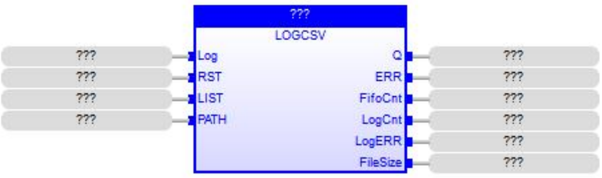
14.12 Data Logging

WorkSuite provides functions for logging data into one or more file(s) in the RTU32. In general 2 functions are available:

1. LOGCSV Function - a dedicated RTU32 Series logging function. You can find an example of this being used in the WorkSuite Demo examples installed together with your WorkSuite Package. This data log function handles the read/write functions in a background scan and does not interfere with the general PLC cycle. You will find details of the LOGCSV file function below.
2. LogFileCSV – simple function for logging data into a CSV-file. It uses a standard file write function and it is not recommended to use for data logging as a standalone Function.

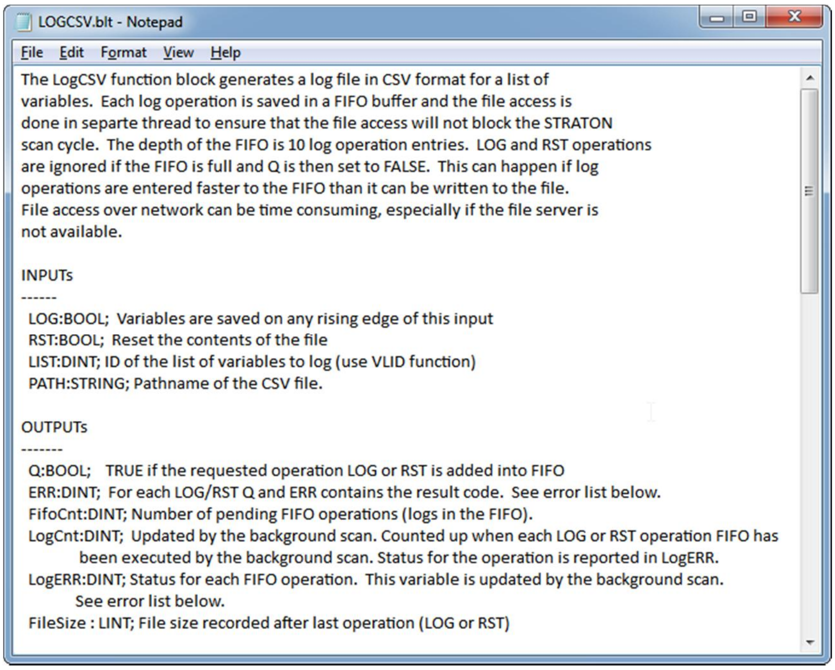
LOGCSV Function

The LOGCSV Function is found in the Brodersen Function library folder.

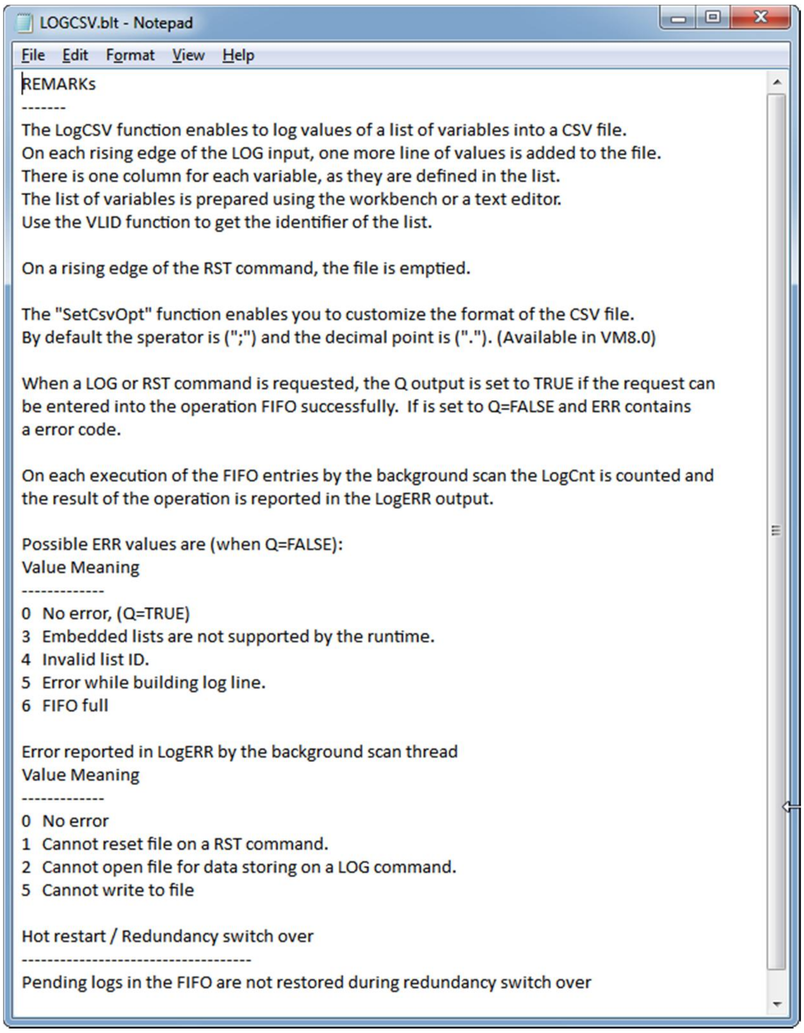


The LogCSV function block generates a log file in CSV format for a list of variables. Each log operation is saved in a FIFO buffer and the file access is done in a separate thread to ensure that the file access will not block the PLC runtime cycle. The depth of the FIFO is 10 log operation entries. LOG and RST operations are ignored if the FIFO is full and Q is then set to FALSE. This can happen if log operations are entered faster to the FIFO than it can be written to the file.

File access over network can be time consuming, especially if the file server is not available.



```
The LogCSV function block generates a log file in CSV format for a list of
variables.  Each log operation is saved in a FIFO buffer and the file access is
done in separte thread to ensure that the file access will not block the STRATON
scan cycle.  The depth of the FIFO is 10 log operation entries.  LOG and RST operations
are ignored if the FIFO is full and Q is then set to FALSE.  This can happen if log
operations are entered faster to the FIFO than it can be written to the file.
File access over network can be time consuming, especially if the file server is
not available.

INPUTs
------
 LOG:BOOL;  Variables are saved on any rising edge of this input
 RST:BOOL;  Reset the contents of the file
 LIST:DINT; ID of the list of variables to log (use VLID function)
 PATH:STRING; Pathname of the CSV file.

OUTPUTs
-------
 Q:BOOL;   TRUE if the requested operation LOG or RST is added into FIFO
 ERR:DINT;  For each LOG/RST Q and ERR contains the result code.  See error list below.
 FifoCnt:DINT; Number of pending FIFO operations (logs in the FIFO).
 LogCnt:DINT;  Updated by the background scan. Counted up when each LOG or RST operation FIFO has
        been executed by the background scan. Status for the operation is reported in LogERR.
 LogERR:DINT; Status for each FIFO operation.  This variable is updated by the background scan.
        See error list below.
 FileSize : LINT; File size recorded after last operation (LOG or RST)
```

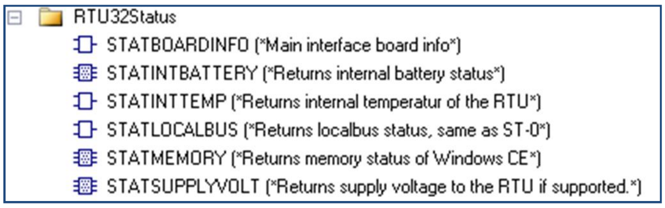


```
REMARKs
-------
The LogCSV function enables to log values of a list of variables into a CSV file.
On each rising edge of the LOG input, one more line of values is added to the file.
There is one column for each variable, as they are defined in the list.
The list of variables is prepared using the workbench or a text editor.
Use the VLID function to get the identifier of the list.

On a rising edge of the RST command, the file is emptied.

The "SetCsvOpt" function enables you to customize the format of the CSV file.
By default the sperator is (";") and the decimal point is ("."). (Available in VM8.0)

When a LOG or RST command is requested, the Q output is set to TRUE if the request can
be entered into the operation FIFO successfully.  If is set to Q=FALSE and ERR contains
a error code.

On each execution of the FIFO entries by the background scan the LogCnt is counted and
the result of the operation is reported in the LogERR output.

Possible ERR values are (when Q=FALSE):
Value Meaning
-------------
0  No error, (Q=TRUE)
3  Embedded lists are not supported by the runtime.
4  Invalid list ID.
5  Error while building log line.
6  FIFO full

Error reported in LogERR by the background scan thread
Value Meaning
-------------
0  No error
1  Cannot reset file on a RST command.
2  Cannot open file for data storing on a LOG command.
5  Cannot write to file

Hot restart / Redundancy switch over
------------------------------------
Pending logs in the FIFO are not restored during redundancy switch over
```

In the RTU32 Example projects in WorkSuite you will find a WorkSuite Demo project LOGCSV_Demo with a log UDFB (User Defined Function Block) block named TMultiFileLog. With this UDFB you will be able to control the number of files to log and limit file size. This allows management of log files in a ring buffer, ensuring that your log files will not fill up the available storage memory.

14.13 Status information functions

RTU32 Series products have hardware status information available.

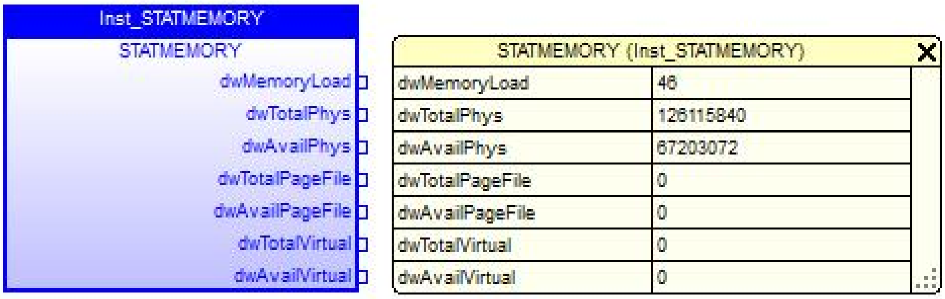


RTU32Status
- STATBOARDINFO (*Main interface board info*)
- STATINTBATTERY (*Returns internal battery status*)
- STATINTTEMP (*Returns internal temperatur of the RTU*)
- STATLOCALBUS (*Returns localbus status, same as ST-0*)
- STATMEMORY (*Returns memory status of Windows CE*)
- STATSUPPLYVOLT (*Returns supply voltage to the RTU if supported.*)

Although most of these status functions are supported in the RTU32M/N firmware, only some of the status functions are available/relevant in the RTU32M/N (ie. many are available directly from the RTU32M modules). The following sections list the functions supported by the RTU32M/N.

14.13.1 RTU Memory Status (STATMEMORY)

The function StatMemory returns the RTU32M/N memory status.



STATMEMORY (Inst_STATMEMORY)

dwMemoryLoad	46
dwTotalPhys	126115840
dwAvailPhys	67203072
dwTotalPageFile	0
dwAvailPageFile	0
dwTotalVirtual	0
dwAvailVirtual	0

Typical memory status in a running RTU32M/N

The commonly used monitoring functions are dwMemory Load, dwTotalPhys and dwAvailPhys.

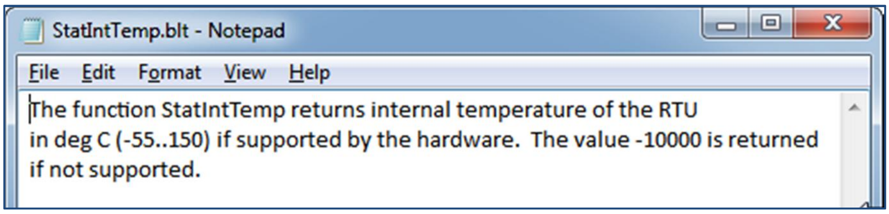
The dwMemoryLoad tells you in percent the actual RAM program memory load. With a medium PLC runtime application you will see a load of approx. 50%. Only applications with multiple driver connections and a lot of logic will require larger amount of memory. The maximum load recommended on a RTU is 90%.

The dwTotalPhys tells you how much RAM program memory is available in the RTU. The rest of the RAM memory up to around 64MB is used for storage (file system etc.).

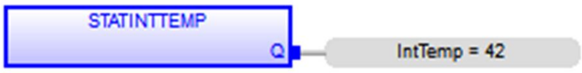
The dwAvailPhys tells you how much available program memory you have in your RTU. The rest is used by the Operating System and your PLC application.

14.13.2 RTU Internal Temperature (STATINTTEMP)

The function StatIntTemp returns internal temperature of the RTU32N.

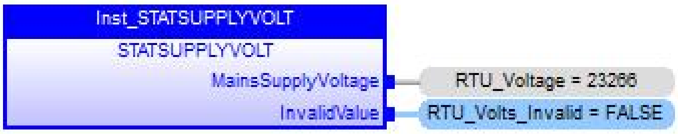


The function StatIntTemp returns internal temperature of the RTU in deg C (-55..150) if supported by the hardware. The value -10000 is returned if not supported.



14.13.3 RTU Supply Voltage (STATSUPPLYVOLT)

The function StatSupplyVolt returns the RTU32N supply voltage (mV).



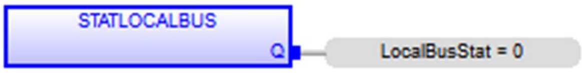
14.13.4 LocalBus Status (STATLOCALBUS)

The function StatLocalbus returns relevant I/O bus status values

If the value of the Word register = 0, then all is fine.

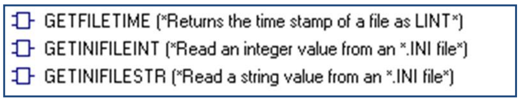
Here are some typical errors for the LocalBus;

Bit 0	Bit 0: if true then it is typically when an unsupported I/O modules is connected to the RTU.
Bit 1	Bit 1: Indicate LB scanning error – this is the typical error indication when an I/O Module fails or are disconnected in a running application.
Bit 2	Bit 2: Will be true if in your runtime project addresses an I/O point that is not available in the database. Be careful if you use same project for several I/O configurations.
Bit 3	Bit 3: The I/O configuration mismatch will be true if the I/O configuration expected by the runtime is not present at start-up. If you have locked the I/O configuration this bit will go true at runtime start-up – and so will bit 1.

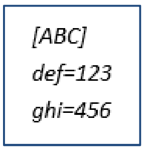


14.14 Reading Text Files from WorkSuite

To provide the functionality to read variables and strings from an external file in the RTU32M/N, a few simple read file functions are supported.


GETFILETIME (*Returns the time stamp of a file as LINT*)
GETINIFILEINT (*Read an integer value from an *.INI file*)
GETINIFILESTR (*Read a string value from an *.INI file*)

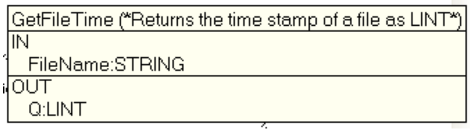
Note: You have to place the text file in the RTU32M/N file system manually – use sFTP to do this and use the Nand Flash location (or SD Flash). A SSH client SFTP interface (such as BitVise) can be used to explorer the RTU32M/N files. The file structure must be the same as other RTU .ini files (such as SNMP.txt and RTU32.ini). The structure is simple and looks like this:


```
[ABC]
def=123
ghi=456
```

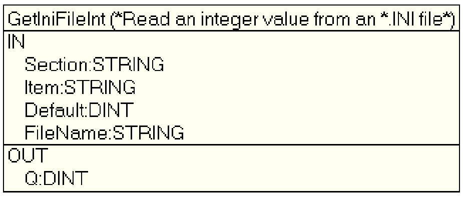
ABC is called the section and def the item. NOTE: all are case sensitive.

The functions available are:

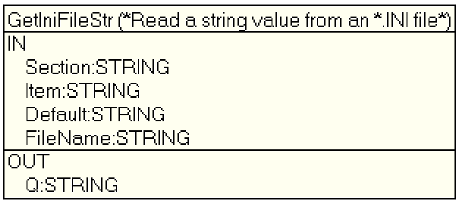
- **GetFileTime** - With this function you can check the time of the file so you only read the variables in case of changes.


GetFileTime (*Returns the time stamp of a file as LINT*)
IN
FileName:STRING
OUT
Q:LINT

- **GetIniFileInt** - Read an Integer value from a section and item.


GetIniFileInt (*Read an integer value from an *.INI file*)
IN
Section:STRING
Item:STRING
Default:DINT
FileName:STRING
OUT
Q:DINT

GetIniFileStr - Read a String variable from a section and item.


GetIniFileStr (*Read a string value from an *.INI file*)
IN
Section:STRING
Item:STRING
Default:STRING
FileName:STRING
OUT
Q:STRING

The primary use of these functions is to support ‘dynamic’ variables outside the RTU runtime application program. You can then use the same application program for several sites and manage the differences for each site in the file. You can edit the file from the web editor with your browser or directly send a new file to the RTU32M/N from your SCADA host using FTP.

15 Distributed Event Binding Protocol / Redundancy Binding Communication

15.1 General

The Distributed Event Binding protocol is a proprietary protocol for fast time stamped event based data transfer directly from one Brodersen RTU runtime to another. The Binding protocol is communicating directly from one RTU32 to another RTU32 in a LAN/WAN network. It can run regardless of any other application. In a network with several Brodersen RTUs you can use the Global Binding Editor to bind any global declared variables together.

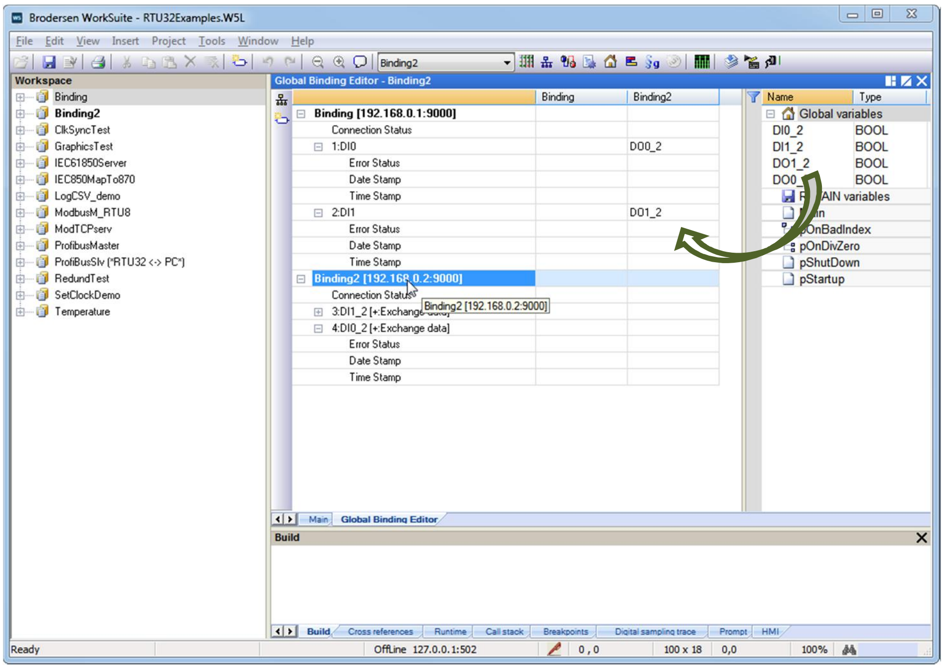
Redundancy Binding is used in a redundancy network connection using 2-6 separate network segments. Both the unit publishing variables (producer) and the unit subscribing variables (consumer) has to be equipped with 2-6 Ethernet network interfaces. Each network connection must be setup on the units as separate network segments using different TCP/IP subnet masks. The compiler also takes care of handling the sorting out of timing issues of any changes in values. This means that you on the subscriber side always willget the correct value. The first received event will be reported at the subscriber side and the same event with the same time stamp received on other network interfaces will be ignored.

15.2 Network settings in RTU32 (redundancy binding)

If you want to run redundancy Binding using the RTU32 you must configure all Ethernet interfaces. No additional settings are necessary as the networks is seen as direct accessible parallel networks. Routing to the networks is done as default. If you ping an address on any of the one of selected network segments, the ping request will find its way to the correct segment automatically.

15.3 Procedure for setting up binding/redundancy binding in WorkSuite

You need to configure all the RTU32 runtimes before starting to bind (link) I/Os together. I/Os and variables to bind must be declared as Global variables. Use variable naming that makes sense for you. You could for example call all input variables published for binding *BindInput_DI0* and *BindInput_DI1*, and you could call outputs *BindOutput_DO0.* After defining the variables for use in both comms direction for all relevant projects (RTUs), you must start the Global Binding Editor (find it in WorkSuite under the Tools menu). Note that in WorkSuite you will need all Projects used for binding in the same project list.



	Binding	Binding2
Binding [192.168.0.1:9000]		
Connection Status		
1:DI0		DO0_2
Error Status		
Date Stamp		
Time Stamp		
2:DI1		DO1_2
Error Status		
Date Stamp		
Time Stamp		
Binding2 [192.168.0.2:9000]		
Connection Status		
3:DI1_2 [+:Exchange data]		
4:DI0_2 [+:Exchange data]		
Error Status		
Date Stamp		
Time Stamp		

Name	Type
Global variables	
DI0_2	BOOL
DI1_2	BOOL
DO1_2	BOOL
DO0_	BOOL
R AIN variables	
in	
pOnBadIndex	
pOnDivZero	
pShutDown	
pStartup	

You assign the IP addresses to each project. If using redundancy Binding, you add up to maximum 6 x "IP:TCP port" to the address of the RTU. If two network interfaces are used for binding (dual redundant binding), the two IP addresses are defined with a separator "|" – like 192.168.1.222:9000|192.168.2.21:9000"

As shown above, the OUTPUTS from Publishing RTU project are linked to the subscribing RTUs INPUTS.

After configuration you download ALL projects to the relevant RTUs. Use the "Download All Projects…" in the Project menu.

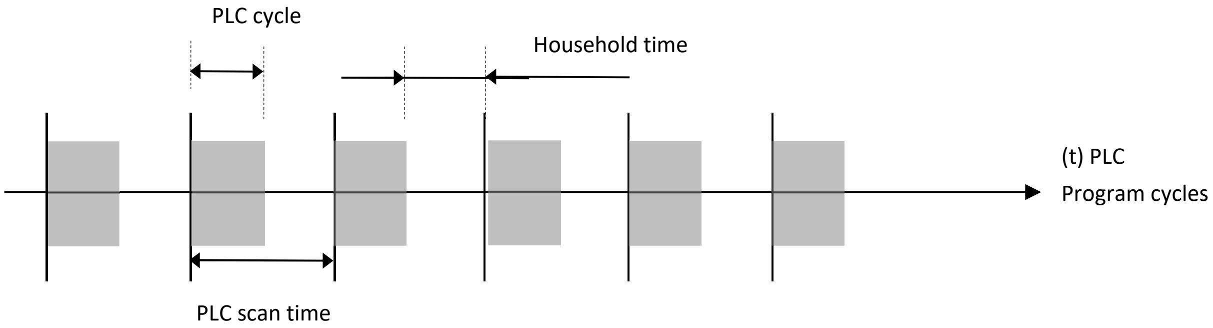
16 RTU32 cycle / scan time

All tasks run independent of each other. The RTU32 I/O database reflects the physical I/O. The PLC runtime program cycle time depends on the size of the application program, and can vary from less than 1 msec to >100 msec. At runtime application start, all inputs are read from the I/O database and at the end of each cycle the outputs are written to the I/O database.

16.1 RTU scan time

The RTU scan time can be set to 0 (zero) which automatically assigns some time for operating system household activities between each runtime program cycle. In this case the program runs as fast as possible. The automatic assignment of household execution time ensures all necessary operating system processes can be maintained with a good margin.

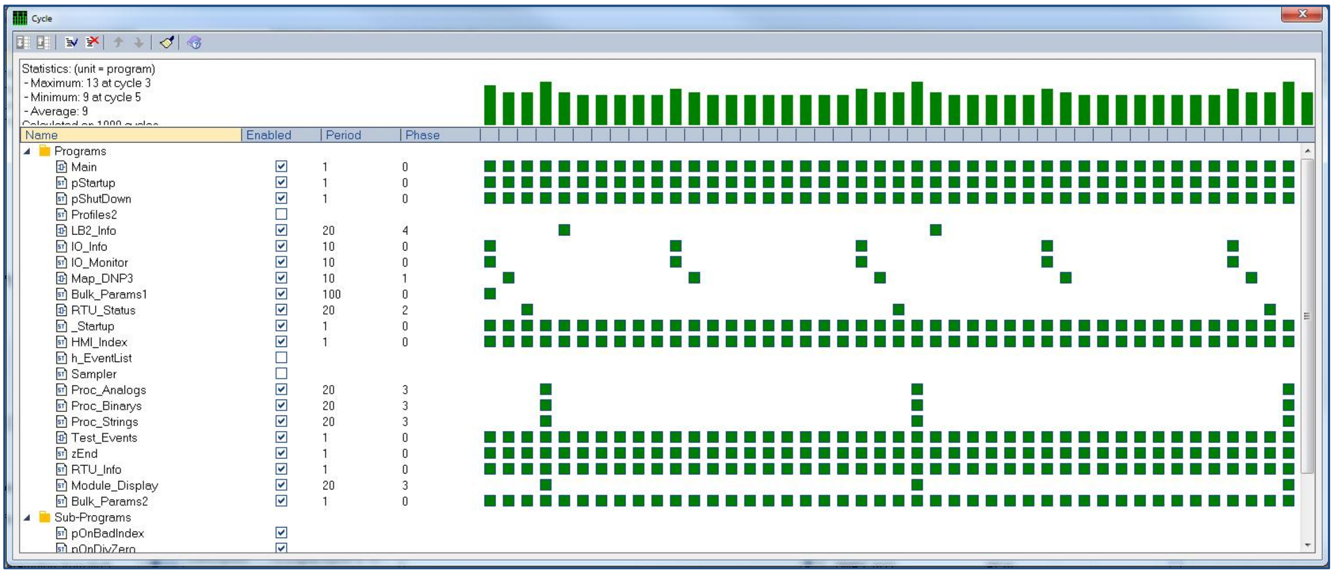
Note: RTU32M/N CPU with clock speeds of 200MHz have a forced delay of 40ms between cycles.



If you set the PLC scan time to a fixed value, you can in some cases obtain a better general scan time for your application program. But it is IMPORTANT to realise that you are then responsible to leave time for household operations.

IMPORTANT NOTICE: If you leave too short time for household operations, the firmware will automatically ensure that your PLC program does NOT compromise necessary household time. It also means that you will get a slower scan cycle time than adjusted. If your RTU/PLC application includes many TCTP/IP based connections over LAN connections, it is recommended to fix the scan time to a value leaving more time to the OS household than automatically assigned.

WorkSuite 3.9 includes a scheduler that allows a visual setup for the period and phase of programs (right click the project heading).



Statistics: (unit = program)
- Maximum: 13 at cycle 3
- Minimum: 9 at cycle 5
- Average: 9
Calculated on 1000 cycles

Name	Enabled	Period	Phase
Programs			
Main	☑	1	0
pStartup	☑	1	0
pShutDown	☑	1	0
Profiles2	□		
LB2_Info	☑	20	4
IO_Info	☑	10	0
IO_Monitor	☑	10	0
Map_DNP3	☑	10	1
Bulk_Params1	☑	100	0
RTU_Status	☑	20	2
_Startup	☑	1	0
HMI_Index	☑	1	0
h_EventList	□		
Sampler	□		
Proc_Analogs	☑	20	3
Proc_Binarys	☑	20	3
Proc_Strings	☑	20	3
Test_Events	☑	1	0
zEnd	☑	1	0
RTU_Info	☑	1	0
Module_Display	☑	20	3
Bulk_Params2	☑	1	0
Sub-Programs			
pOnBadIndex	☑		
pOnDivZero	☑		

16.2 LB2 Bus and Internal I/O Bus scan time

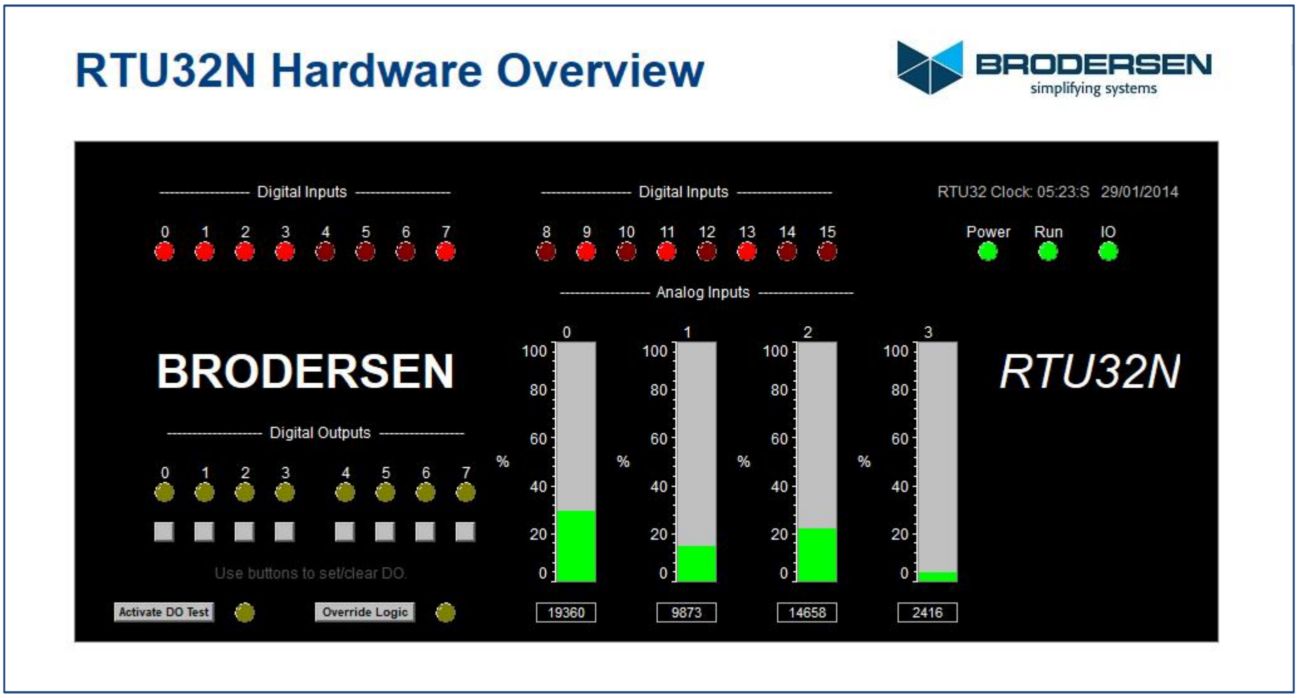
The scan time of the RTU32M internal I/O is running as fast as possible (in micro seconds).

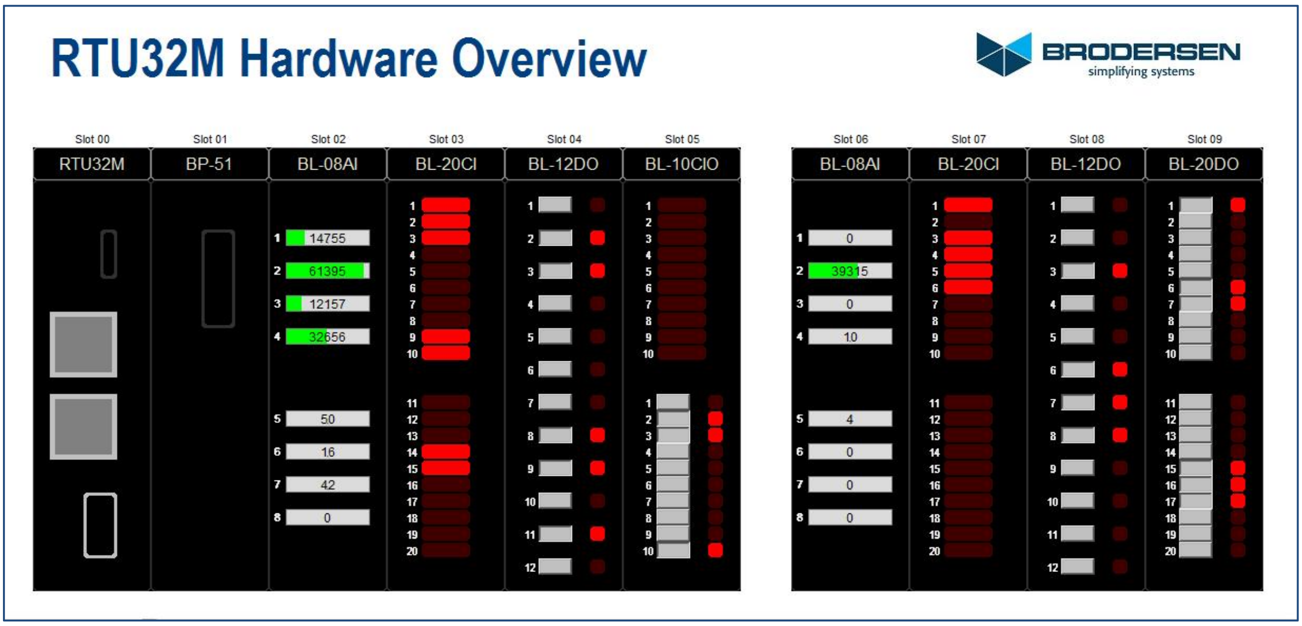
17 RTU HMI

17.1 Examples

The RTU32 Series support simple HMI functions developed in WorkSuite. You can easily add graphic displays to your application and monitor data online from your PC using the WorkSuite HMI Viewer.

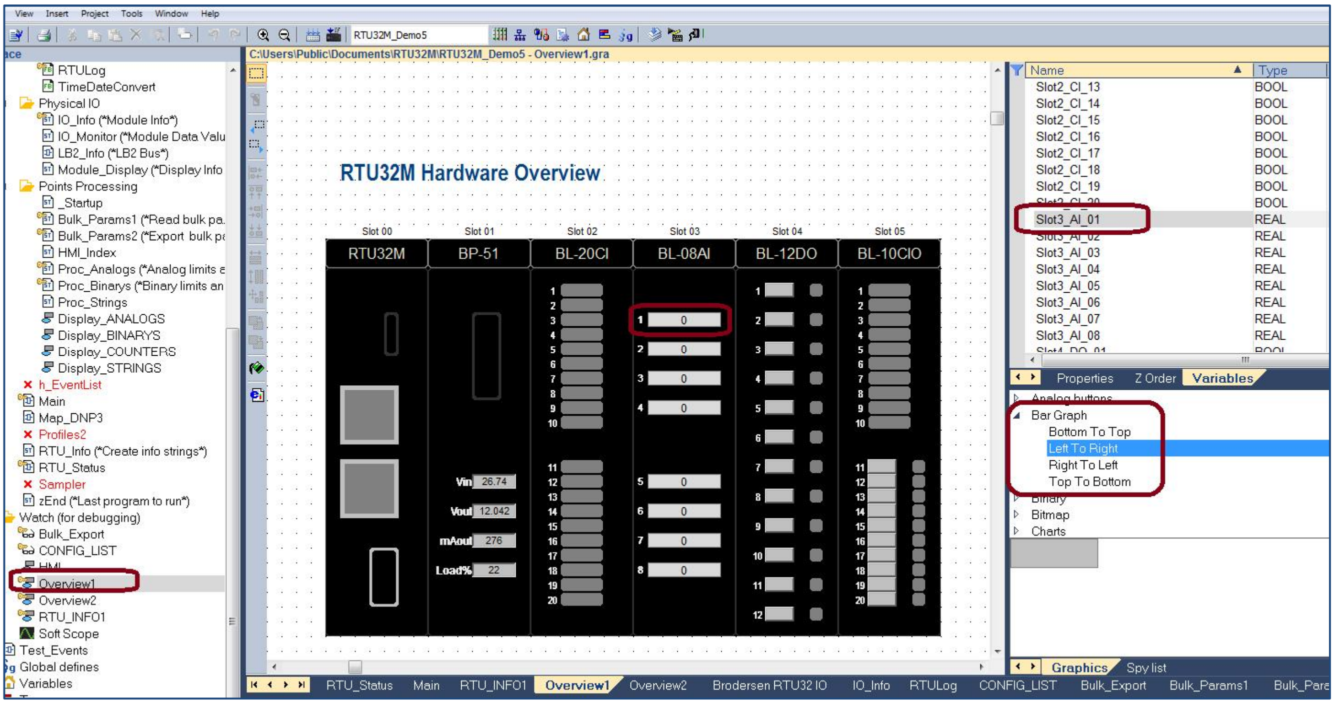
Examples:





17.2 Creating and Viewing RTU Graphics

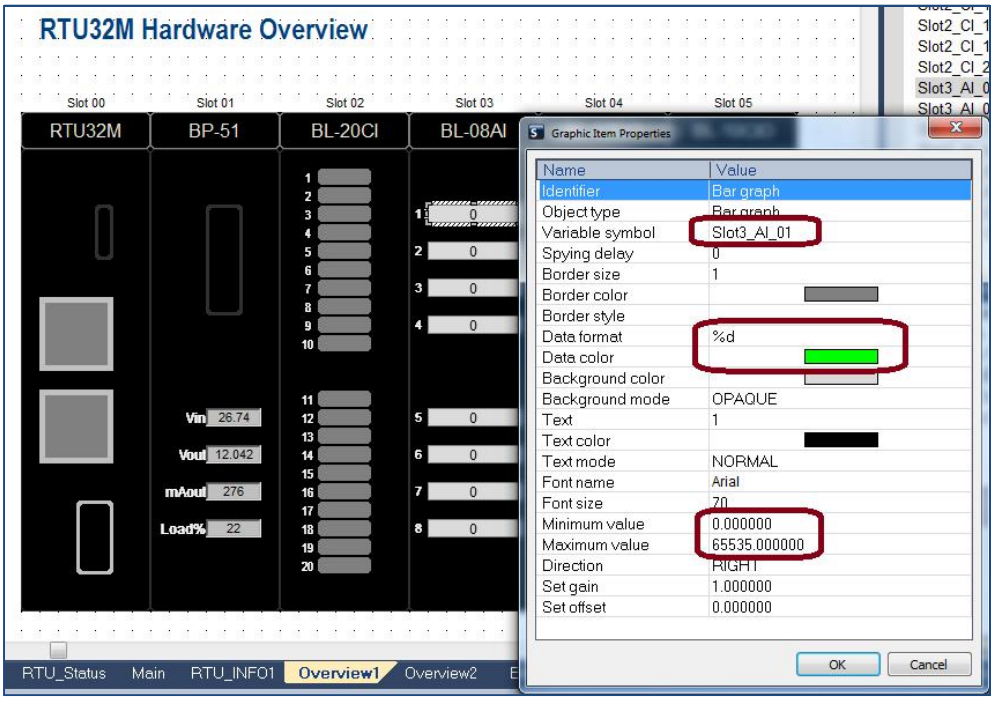
The example page below can be built using the standard WorkSuite graphic tools.



Name	Type
Slot2_CI_13	BOOL
Slot2_CI_14	BOOL
Slot2_CI_15	BOOL
Slot2_CI_16	BOOL
Slot2_CI_17	BOOL
Slot2_CI_18	BOOL
Slot2_CI_19	BOOL
Slot2_CI_20	BOOL
Slot3_AI_01	REAL
Slot3_AI_02	REAL
Slot3_AI_03	REAL
Slot3_AI_04	REAL
Slot3_AI_05	REAL
Slot3_AI_06	REAL
Slot3_AI_07	REAL
Slot3_AI_08	REAL
Slot4_DO_01	BOOL

The page can include static text, shapes and a range of items that provide the animation.

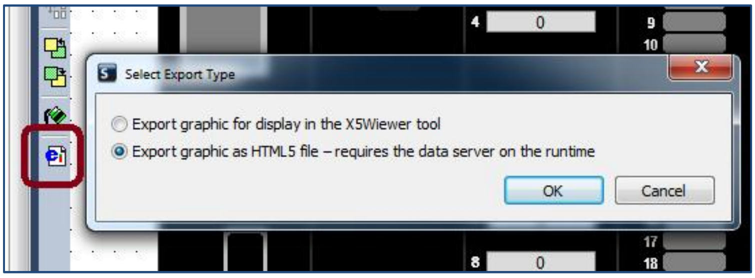
The example here shows the setup of a horizontal bar graph that moves from left to right and uses the value from the variable named ‘Slot3_AI_01’ to show the value as both a text value and a scaled bar.



Graphic Item Properties

Name	Value
Identifier	Bar graph
Object type	Bar graph
Variable symbol	Slot3_AI_01
Spying delay	0
Border size	1
Border color	▇
Border style	
Data format	%d
Data color	🟩
Background color	▇
Background mode	OPAQUE
Text	1
Text color	■
Text mode	NORMAL
Font name	Arial
Font size	70
Minimum value	0.000000
Maximum value	65535.000000
Direction	RIGHT
Set gain	1.000000
Set offset	0.000000

The pages can be viewed when simulating or debugging a logic application and can also be published for use by the X5 HMI Graphics Viewer (a free client application shipped with WorkSuite), or published as a .htm file for use within web browser applications such as Internet Explorer and Chrome.



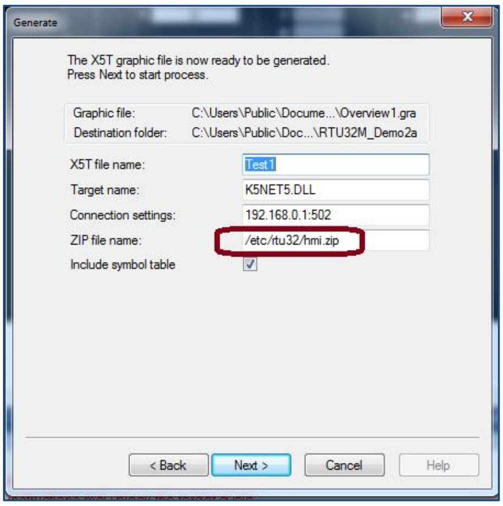
Select Export Type

○ Export graphic for display in the X5Wiewer tool

● Export graphic as HTML5 file – requires the data server on the runtime

The X5 Viewer installs in the same location on your PC as WorkSuite eg. Programs>Brodersen>WorkSuite.

The WorkSuite generate tools create a .zip file for use by the X5 Viewer and a .htm file, based on your Export selection .

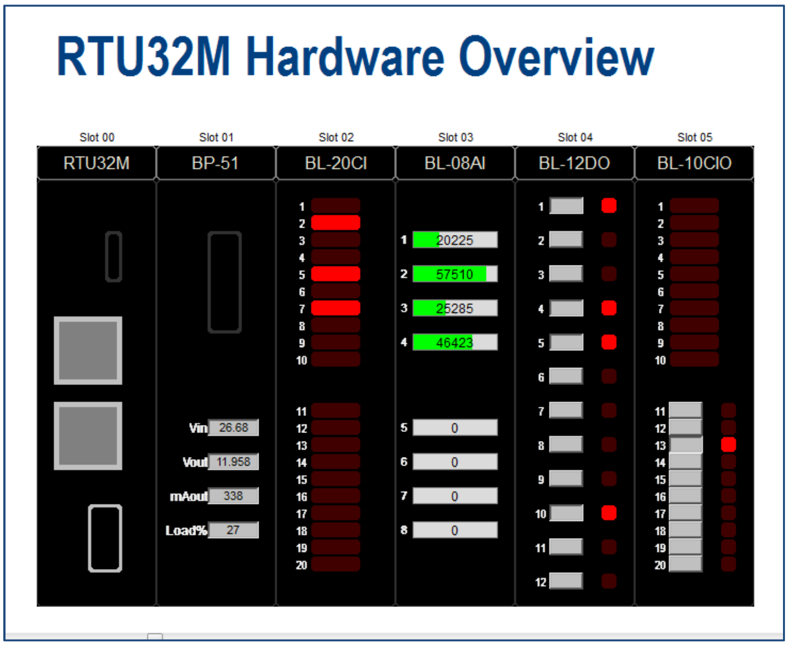


Generate

The X5T graphic file is now ready to be generated.
Press Next to start process.

Graphic file: C:\Users\Public\Docume...\Overview1.gra
Destination folder: C:\Users\Public\Doc...\RTU32M_Demo2a

X5T file name: Test1
Target name: K5NET5.DLL
Connection settings: 192.168.0.1:502
ZIP file name: /etc/rtu32/hmi.zip
Include symbol table ☑



RTU32M Hardware Overview

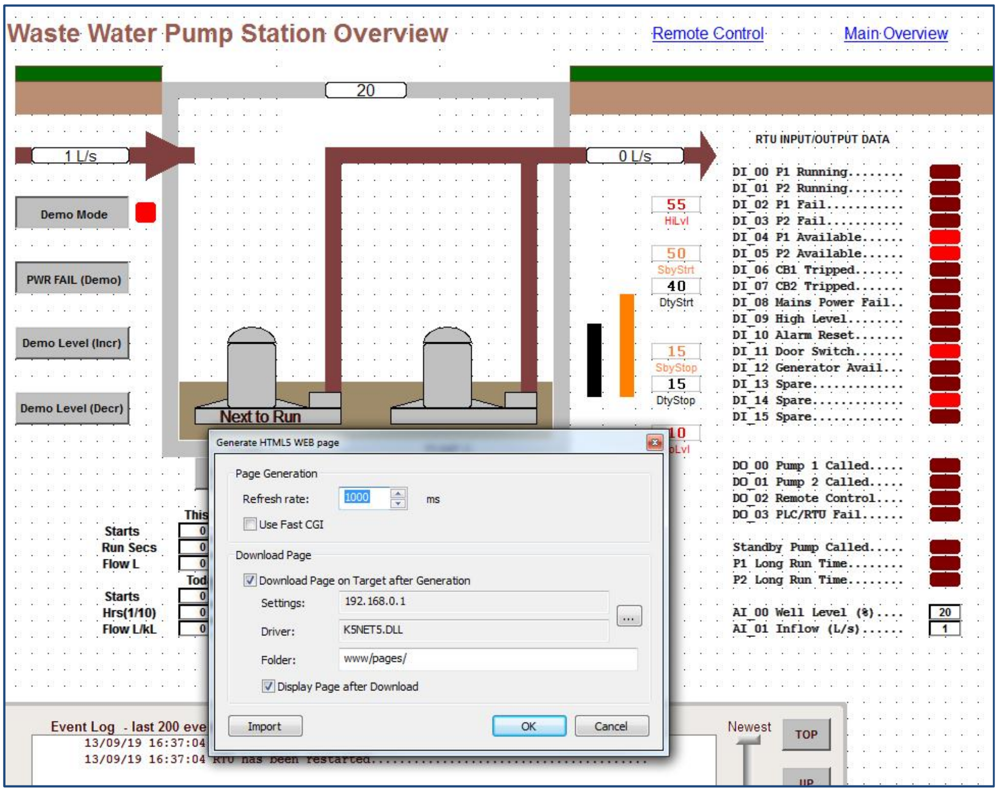
The published files look the same when viewed in the X5 Viewer or a browser, as shown above.

Note: the destinations for X5 and html5 files are different.

X5 use: /etc/rtu32/hmi.zip

Html use: www/pages

Note: FastCGI is presently not supported.



Waste Water Pump Station Overview

```
13/09/19 16:37:04
13/09/19 16:37:04 RTU has been restarted.
```

Generate HTML5 WEB page

Page Generation
Refresh rate: 1000 ms
□ Use Fast CGI

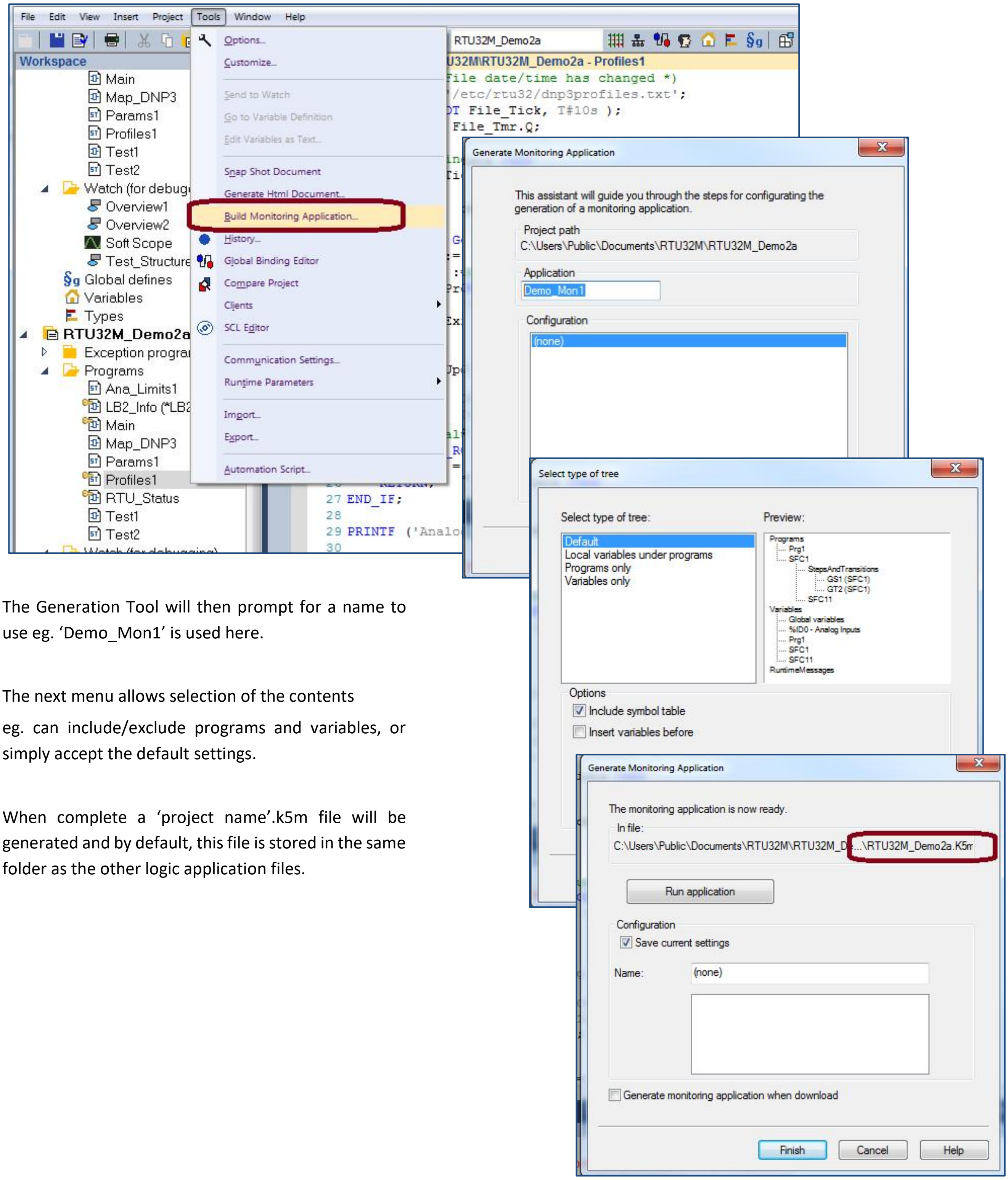
Download Page
☑ Download Page on Target after Generation
Settings: 192.168.0.1
Driver: K5NET5.DLL
Folder: www/pages/
☑ Display Page after Download

18 RTU Monitoring Applications

18.1 Creating and Viewing RTU Monitor Applications

WorkSuite allows publishing of the RTU32 Series logic applications as pages that can be viewed online using the Monitoring tool.

From the Tools menu select ‘Build Monitoring Application’.



```
file date/time has changed *)
'/etc/rtu32/dnp3profiles.txt';
DT File_Tick, T#10s );
File_Tmr.Q;
27 END_IF;
28
29 PRINTF ('Analo
```

Generate Monitoring Application

This assistant will guide you through the steps for configurating the generation of a monitoring application.

Project path: C:\Users\Public\Documents\RTU32M\RTU32M_Demo2a

Application: Demo_Mon1

Configuration: (none)

Select type of tree:

- Default
- Local variables under programs
- Programs only
- Variables only

Options

☑ Include symbol table

☐ Insert variables before

Generate Monitoring Application

The monitoring application is now ready.

In file: C:\Users\Public\Documents\RTU32M\RTU32M_D...\RTU32M_Demo2a.K5m

Configuration

☑ Save current settings

Name: (none)

☐ Generate monitoring application when download

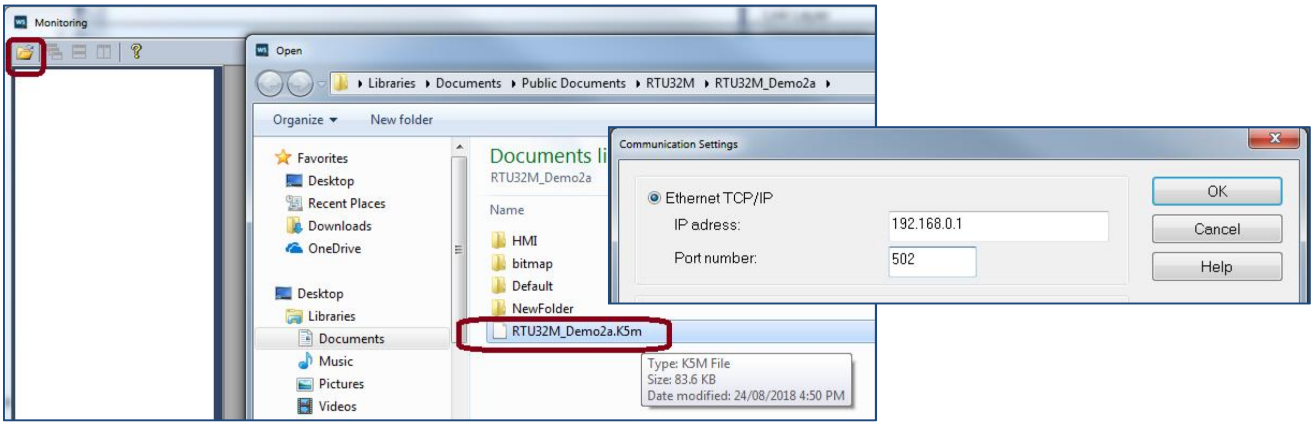
The Generation Tool will then prompt for a name to use eg. ‘Demo_Mon1’ is used here.

The next menu allows selection of the contents

eg. can include/exclude programs and variables, or simply accept the default settings.

When complete a ‘project name’.k5m file will be generated and by default, this file is stored in the same folder as the other logic application files.

The monitoring application can be viewed using the 'W5Monitoring.exe' application that is located in the same folder as the WorkSuite application eg. Program Files(x86)\Brodersen\WorkSuite3.9.2. Use the folder icon to select the monitoring application.



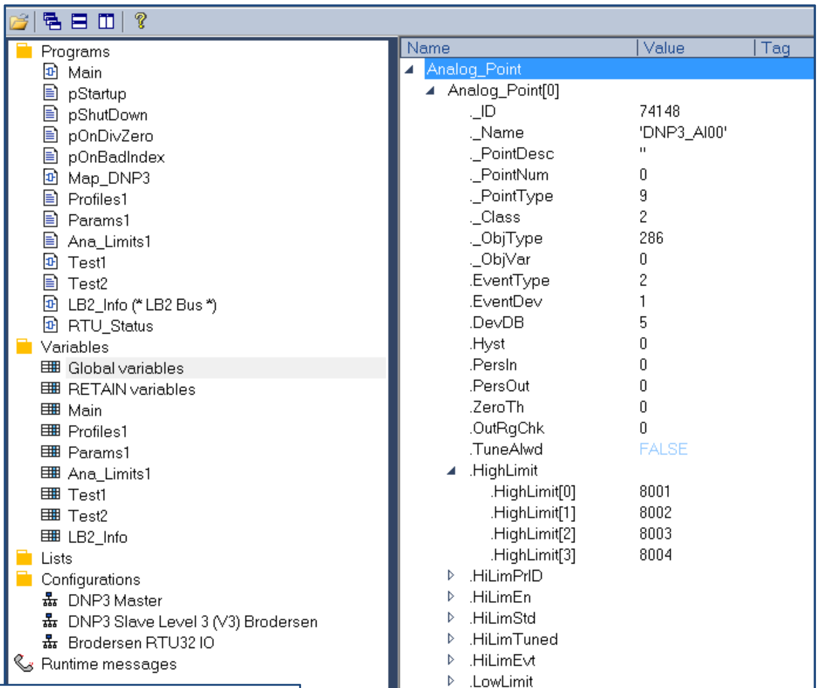
● Ethernet TCP/IP

IP adress: 192.168.0.1

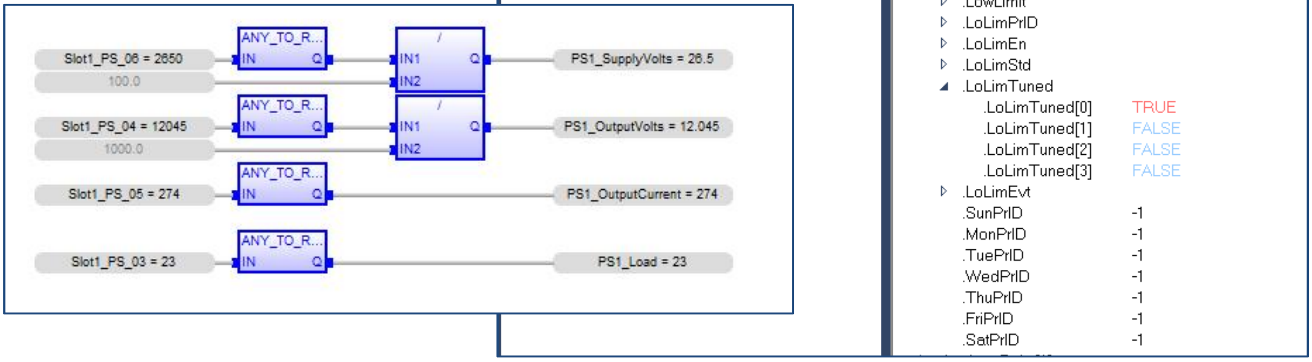
Port number: 502

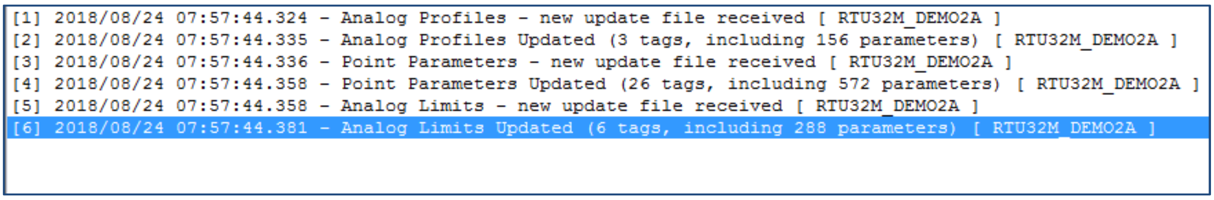
You will then be prompted for the RTUs communications settings. The example above shows setup of the IP address and port number.

Once connected, you can view variables (local/global and complex/structured tags), programs and runtime information as shown below.


- Programs
  - Main
  - pStartup
  - pShutDown
  - pOnDivZero
  - pOnBadIndex
  - Map_DNP3
  - Profiles1
  - Params1
  - Ana_Limits1
  - Test1
  - Test2
  - LB2_Info (* LB2 Bus *)
  - RTU_Status
- Variables
  - Global variables
  - RETAIN variables
  - Main
  - Profiles1
  - Params1
  - Ana_Limits1
  - Test1
  - Test2
  - LB2_Info
- Lists
- Configurations
  - DNP3 Master
  - DNP3 Slave Level 3 (V3) Brodersen
  - Brodersen RTU32 IO
- Runtime messages

Name	Value	Tag
Analog_Point		
Analog_Point[0]		
_ID	74148	
_Name	'DNP3_AI00'	
_PointDesc	''	
_PointNum	0	
_PointType	9	
_Class	2	
_ObjType	286	
_ObjVar	0	
EventType	2	
EventDev	1	
DevDB	5	
Hyst	0	
PersIn	0	
PersOut	0	
ZeroTh	0	
OutRgChk	0	
TuneAlwd	FALSE	
HighLimit		
HighLimit[0]	8001	
HighLimit[1]	8002	
HighLimit[2]	8003	
HighLimit[3]	8004	
HiLimPrID		
HiLimEn		
HiLimStd		
HiLimTuned		
HiLimEvt		
LowLimit		
LoLimPrID		
LoLimEn		
LoLimStd		
LoLimTuned		
LoLimTuned[0]	TRUE	
LoLimTuned[1]	FALSE	
LoLimTuned[2]	FALSE	
LoLimTuned[3]	FALSE	
LoLimEvt		
SunPrID	-1	
MonPrID	-1	
TuePrID	-1	
WedPrID	-1	
ThuPrID	-1	
FriPrID	-1	
SatPrID	-1	




```
[1] 2018/08/24 07:57:44.324 - Analog Profiles - new update file received [ RTU32M_DEMO2A ]
[2] 2018/08/24 07:57:44.335 - Analog Profiles Updated (3 tags, including 156 parameters) [ RTU32M_DEMO2A ]
[3] 2018/08/24 07:57:44.336 - Point Parameters - new update file received [ RTU32M_DEMO2A ]
[4] 2018/08/24 07:57:44.358 - Point Parameters Updated (26 tags, including 572 parameters) [ RTU32M_DEMO2A ]
[5] 2018/08/24 07:57:44.358 - Analog Limits - new update file received [ RTU32M_DEMO2A ]
[6] 2018/08/24 07:57:44.381 - Analog Limits Updated (6 tags, including 288 parameters) [ RTU32M_DEMO2A ]
```

19 RTU32 Utilities and System Event Log

19.1 General

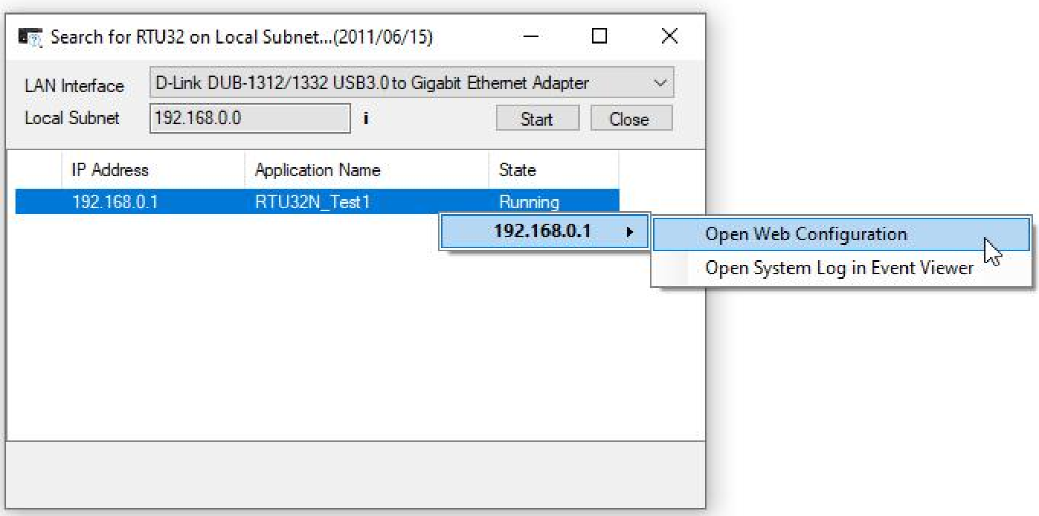
Brodersen provide some utility programs for the RTU32 Series on the RTU32 CD included in each package. The tools include:

- RTU32 Search Tool
- Event Viewer for RTU32 system log monitoring

19.2 RTU32 Search Tool

The RTU Search Tool is used to find RTUs on an Ethernet network. It will search using the network card/interface you define under the LAN Interface. Be careful that you select the right one before you start the search procedure.

The Local Subnet will automatically detect the network details and if you place your cursor on the “i” you will get network details.



Search for RTU32 on Local Subnet...(2011/06/15)

LAN Interface: D-Link DUB-1312/1332 USB3.0 to Gigabit Ethernet Adapter

Local Subnet: 192.168.0.0 i

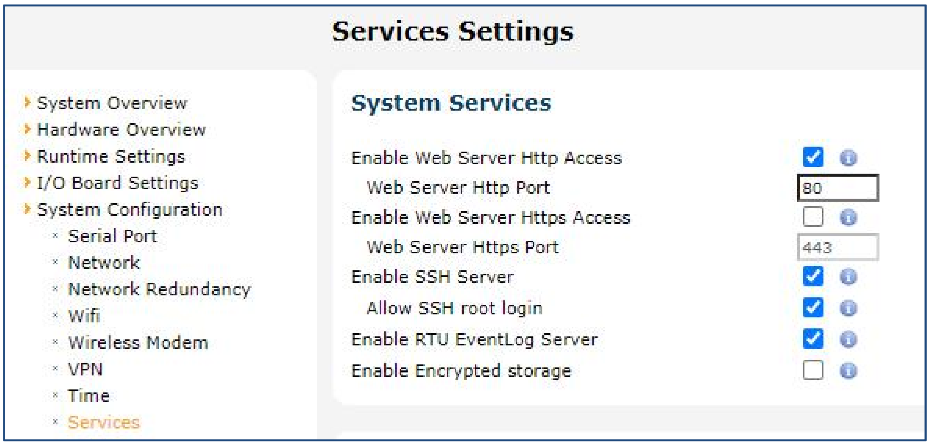
IP Address	Application Name	State
192.168.0.1	RTU32N_Test1	Running

When you start the search process, the network will be scanned for Brodersen RTUs. Only RTUs with their RTU PLC runtime running will answer. If you select an RTU on the list of RTUs found on the network and right-click, you will be able to open either the Webpage or the Event Viewer (status Log info).

19.3 System Log/Event Viewer

The system log provides you with information about active processes in the RTU32N, errors and status information. You can also view debug or information text from your WorkSuite application program which will be printed in the system log. COM port communications can be logged if you add the right parameters to the COM port setup (,LL extension) – see COM port settings for details.

The System Log can be viewed by using the Brodersen Event Viewer utility or a standard Telnet window. The RTU EventLog Server is enabled by default in the System Configuration>Services menu.

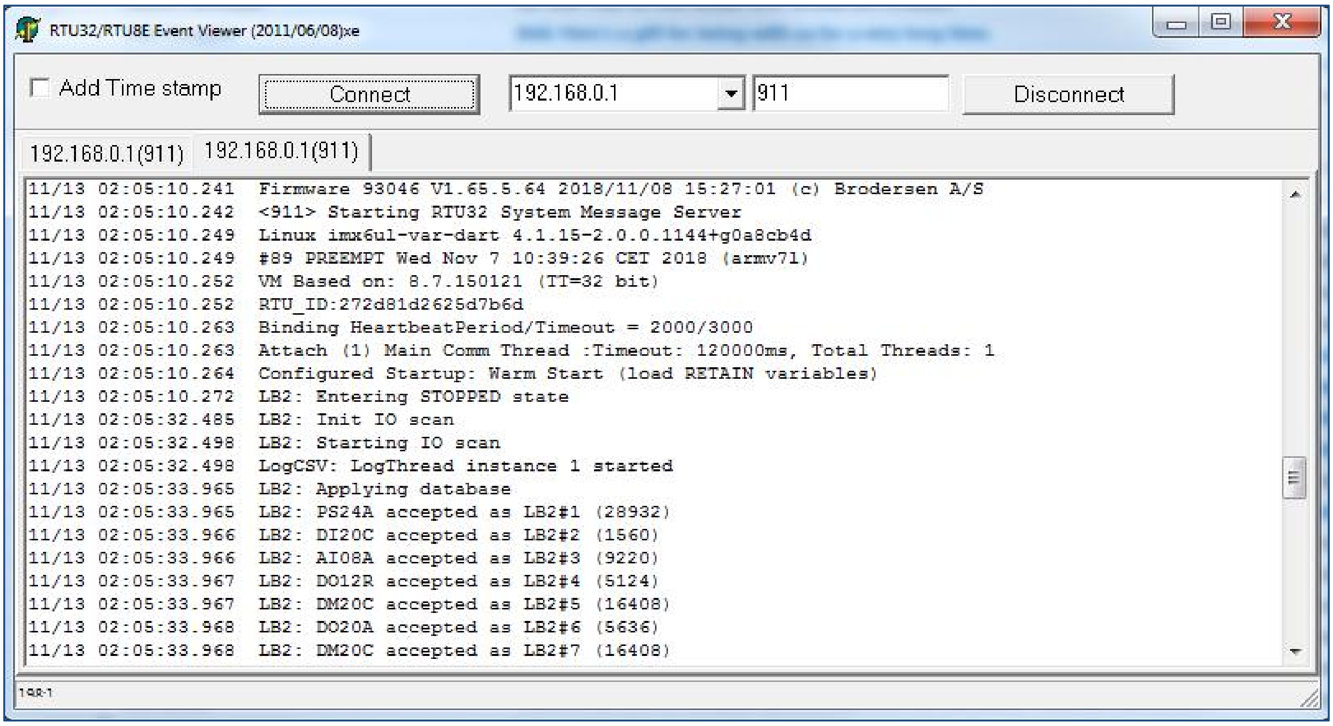


Services Settings

System Services

Enable Web Server Http Access	☑
Web Server Http Port	80
Enable Web Server Https Access	☐
Web Server Https Port	443
Enable SSH Server	☑
Allow SSH root login	☑
Enable RTU EventLog Server	☑
Enable Encrypted storage	☐

The event logs look like this in the Event Viewer:



```
11/13 02:05:10.241  Firmware 93046 V1.65.5.64 2018/11/08 15:27:01 (c) Brodersen A/S
11/13 02:05:10.242  <911> Starting RTU32 System Message Server
11/13 02:05:10.249  Linux imx6ul-var-dart 4.1.15-2.0.0.1144+g0a8cb4d
11/13 02:05:10.249  #89 PREEMPT Wed Nov 7 10:39:26 CET 2018 (armv7l)
11/13 02:05:10.252  VM Based on: 8.7.150121 (TT=32 bit)
11/13 02:05:10.252  RTU_ID:272d81d2625d7b6d
11/13 02:05:10.263  Binding HeartbeatPeriod/Timeout = 2000/3000
11/13 02:05:10.263  Attach (1) Main Comm Thread :Timeout: 120000ms, Total Threads: 1
11/13 02:05:10.264  Configured Startup: Warm Start (load RETAIN variables)
11/13 02:05:10.272  LB2: Entering STOPPED state
11/13 02:05:32.485  LB2: Init IO scan
11/13 02:05:32.498  LB2: Starting IO scan
11/13 02:05:32.498  LogCSV: LogThread instance 1 started
11/13 02:05:33.965  LB2: Applying database
11/13 02:05:33.965  LB2: PS24A accepted as LB2#1 (28932)
11/13 02:05:33.966  LB2: DI20C accepted as LB2#2 (1560)
11/13 02:05:33.966  LB2: AI08A accepted as LB2#3 (9220)
11/13 02:05:33.967  LB2: DO12R accepted as LB2#4 (5124)
11/13 02:05:33.967  LB2: DM20C accepted as LB2#5 (16408)
11/13 02:05:33.968  LB2: DO20A accepted as LB2#6 (5636)
11/13 02:05:33.968  LB2: DM20C accepted as LB2#7 (16408)
```

The RTU32 allows a maximum of 20 continuous connections and the log FIFO includes a maximum of 100 messages.

20 RTU32N Technical Description

20.1 General

The RTU32N Compact RTU/PLC is based on a 32bit i.Mx6 CPU architecture running an Embedded Linux operating system. An industrial power supply is integrated to support industrial supply levels. The basic RTU32N includes a range of communication and other interfaces. The RTU32N provides interfaces like LocalBus and LB2 Bus for expansion I/O and integrated I/O.

The RTU32N software is stored on an internal Flash or micro SD card. During start-up, the operating system and applications are moved to RAM where it is executed. Retained variables are stored in NV RAM.

Using the Ethernet network for primary communication provides all the advantages of existing TCP/IP networking communication facilities. Fast, reliable and secure communication is the main features and all networking components (software, routers, switches, etc.) are available. In addition, serial ports for interfacing to application specific protocols (e.g. Modbus, Fieldbus, utility and traffic proprietary protocols etc.) are available.

Detailed technical information about the RTU32N hardware is provided in the RTU32N Data Sheet and in the RTU32N Wiring and Mounting Instruction manual.

20.2 PLC Runtime

The WorkSuite programming tool fully supports EN/IEC61131 and is used for developing PLC programs in the RTU32N. The application program kernel is implemented and runs in the Windows Embedded CE real-time task. WorkSuite offers complete PLC functionality as found in both normal PLC runtimes and in SoftPLCs, and supports all features needed in today's industrial environment. WorkSuite supports programming languages including Structured Text, Function Block, Ladder, Instruction List and Sequential Function Chart. The VM PLC runtime supports cold restart, hot restart and on-line changes.

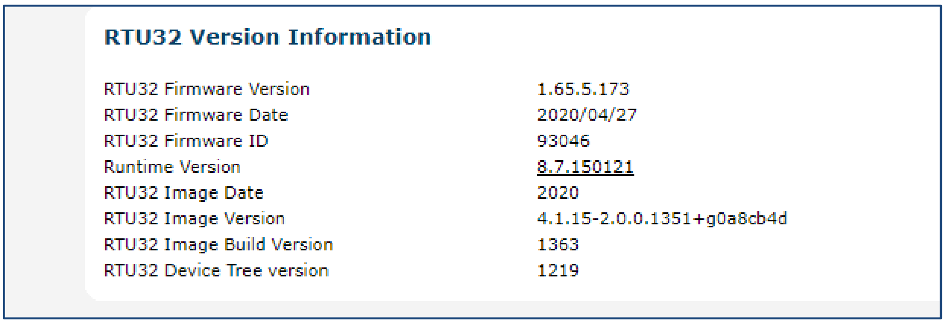
The Brodersen WorkSuite is used for configuration protocols, programming and debugging. It supports several tools for multi-program handling and documentation. It is also a powerful tool for complete system design and programming, providing unique functions for event based communication. The Global Binding Editor makes it possible to publish and subscribe variables in a large network with minimum communication load. The events are time stamped and can also be used directly in ZenOn SCADA HMI applications.

Programming, debugging and upload and download of application programs can be done remotely via Ethernet or RS232.

The RTU32N application kernel is used in many products and applications worldwide. More details about WorkSuite including training and help can be gained by contacting your local Brodersen distributor or Brodersen support.

20.3 Software Versions

The Settings overview page in the RTU32 includes the software version for the actual build.



RTU32 Version Information

RTU32 Firmware Version	1.65.5.173
RTU32 Firmware Date	2020/04/27
RTU32 Firmware ID	93046
Runtime Version	8.7.150121
RTU32 Image Date	2020
RTU32 Image Version	4.1.15-2.0.0.1351+g0a8cb4d
RTU32 Image Build Version	1363
RTU32 Device Tree version	1219

If you contact Brodersen or your local distributor / system integrator for support, it is important that you have the **RTU32N Version Information** ready together with the serial number of your device(s).

Appendix 1

RTU32N runtime application measurement / benchmark

The RTU32N runtime cycle time is measured in a number of different applications below to provide examples of expected runtime periods.

The RTU32N cycle time is set to 0 by default in WorkSuite. The cycle time is the period between the start of each logic cycle. When the cycle time is set to 0 there is no delay between successive logic cycles. Eg. If the logic cycle is set to 100ms and the time taken to complete a logic cycle is 3ms, then the logic process will be idle for 97ms. If the time taken to complete a logic cycle is >100mS, then the logic cycle will be completed, however the 'scan overrun' flag will bet set, indicating the logic cycle time was exceeded.

Note: CPUs with a clock speed of 200MHz are assigned a fixed delay between logic cycles of 40msec for 'household' activities to ensure sufficient time for communication drivers to run smoothly and avoid any overrun situations.

Example 1: RTU32N runtime application program without any communications

Test configuration:

1 x RTU32N with 42 internal I/O (6x AI, 16x DI, 8x DO, 8x DI/DO, 4x Fast DI)

1 x DI20C Expansion module (20x DI channels)

1 x AI08A Expansion module (8x AI channels)

1 x DO12R Expansion module (12x DO channels)

Logic:

- 14 analogue inputs are scaled, some simple calculations are executed, data is stored in internal registers.
- 20 digital inputs are controlling one timer each, connected to a digital output.
- 4 fast digital inputs are controlling 4 counters.

Number of scans completed in one second and average scan period.

CPU Clock Speed	Scans Per Second	Average Scan Period
200MHz	23	43ms
500MHz	473	<3ms

Example 2: RTU32N application program with ModbusTCP communications

Test configuration:

Same configuration as above, but Modbus TCP master protocol is used to read and write 50 holding registers from/to a Modbus slave (~15 read and write messages per second).

Number of scans completed in one second and average scan period.

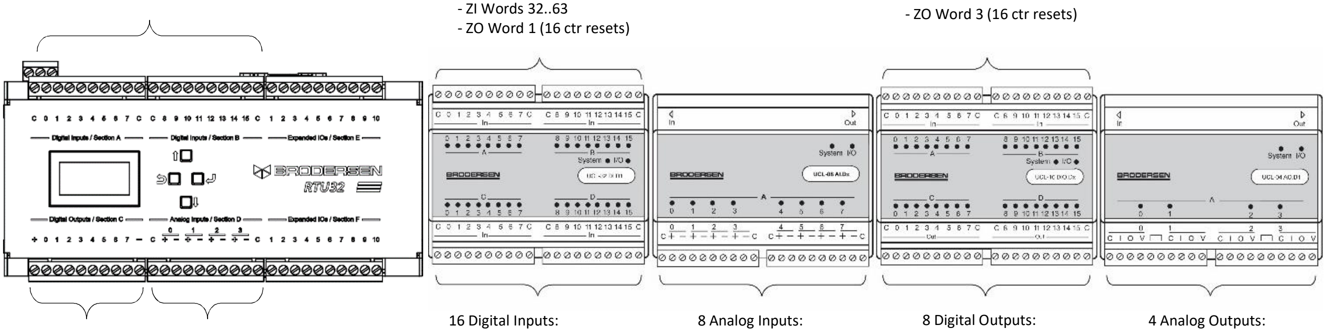
CPU Clock Speed	Scans Per Second	Average Scan Period
200MHz	21	47ms
500MHz	436	<3ms

Appendix 2

RTU32N-28 I/O addressing example 1 – using UCL legacy I/O modules

The addressing example shows the RTU32N-28IO.

UCN-28IOA… **UCL-32DI.Dx** **UCL-08AIC.D** **UCL-16DIO.Px** **UCL-04AO.Dx**


Digital Inputs / Section A
Digital Inputs / Section B
Expanded IOs / Section E
Digital Outputs / Section C
Analog Inputs / Section D
Expanded IOs / Section F
BRODERSEN
RTU32
In
Out
System I/O

16 Digital Inputs:
- DI Word 0 (bits 0..15)
- ZI Words 0..31
- ZO Word 0 (16 ctr resets)

16 Digital Inputs:
- DI Word 1 (bits 0..15)
- ZI Words 32..63
- ZO Word 1 (16 ctr resets)

16 Digital Inputs:
- DI Word 3 (bits 0..15)
- ZI Words 96..127
- ZO Word 3 (16 ctr resets)

8 Digital Outputs:
- DO Word 0 (bits 0..7)

4 Analog Inputs:
- AI Words 0..7

16 Digital Inputs:
- DI Word 2 (bits 0..15)
- ZI Words 64..95
- ZO Word 2 (16 ctr resets)

8 Analog Inputs:
- AI Words 8..23

8 Digital Outputs:
- DO Word 1 (bits 0..15)

4 Analog Outputs:
- AO Words 0..3

Appendix 3

RTU32N-42 I/O addressing example 2 – using UCL legacy I/O modules

The addressing example shows the RTU32N-42IO.

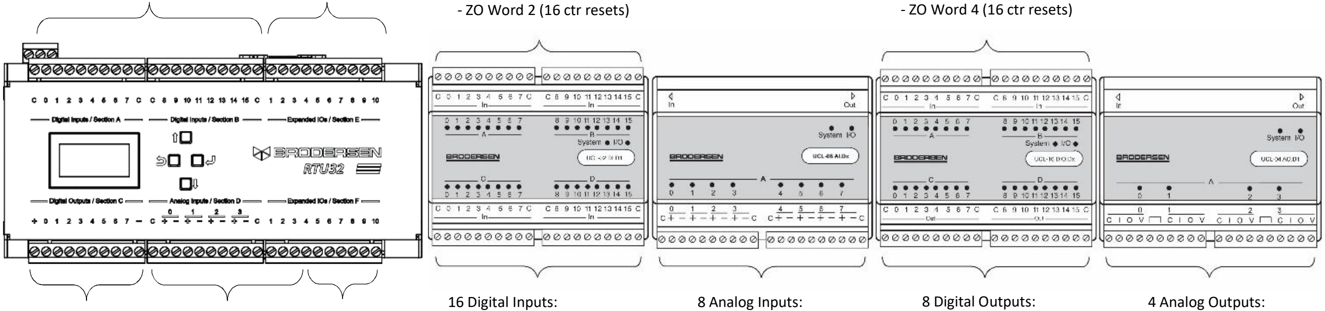
UCN-42IOA… **UCL-32DI.Dx** **UCL-08AIC.D** **UCL-16DIO.Px** **UCL-04AO.Dx**

16 Digital Inputs:
- DI Word 0 (bits 0..15)
- ZI Words 0..31
- ZO Word 0 (16 ctr resets)

8 Digital Inputs/Outputs:
- DI Word 1 (bits 0..7)
- ZI Words 32..47
- ZO Word 1 (8 ctr resets
- DO Word 0 (bits 8..15)

16 Digital Inputs:
- DI Word 2 (bits 0..15)
- ZI Words 64..95
- ZO Word 2 (16 ctr resets)

16 Digital Inputs:
- DI Word 4 (bits 0..15)
- ZI Words 127..159
- ZO Word 4 (16 ctr resets)


BRODERSEN
RTU32
Digital Inputs / Section A
Digital Inputs / Section B
Expanded IOs / Section E
Digital Outputs / Section C
Analog Inputs / Section D
Expanded IOs / Section F

8 Digital Outputs:
- DO Word 0 (bits 0..7)

6 Analog Inputs:
- AI Words 0..11

4 DI/Fast Counters:
- DI Word 1 (bits 8..11)
- ZI Words 48..55
- ZO Word 1 (4 ctr resets)

16 Digital Inputs:
- DI Word 3 (bits 0..15)
- ZI Words 96..127
- ZO Word 3 (16 ctr resets)

8 Analog Inputs:
- AI Words 12..27

8 Digital Outputs:
- DO Word 1 (bits 0..15)

4 Analog Outputs:
- AO Words 0..3

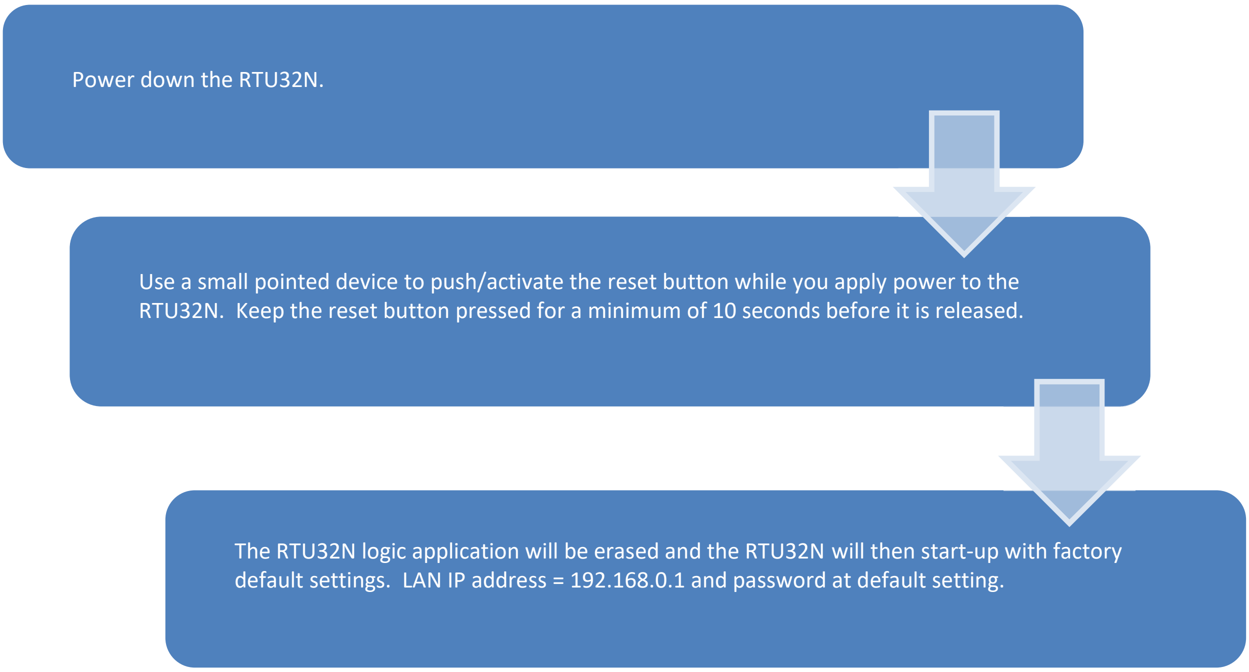
Appendix 4 – RTU32N factory Reset

RTU32N complete Reset

You can make a complete reset of the RTU32N that will reset it back to factory default settings. Note that during reset process all settings in the RTU32N are lost.

To make a complete reset you will need a small screwdriver, pointed pen or paperclip for activating the reset button.
On the underside of the RTU32N (located near COM4) you will find a small 3mm hole. Behind this is a small push button that is used in the reset process.

Procedure:



Power down the RTU32N.

Use a small pointed device to push/activate the reset button while you apply power to the RTU32N. Keep the reset button pressed for a minimum of 10 seconds before it is released.

The RTU32N logic application will be erased and the RTU32N will then start-up with factory default settings. LAN IP address = 192.168.0.1 and password at default setting.