RTU32 Series

RTU, PLC and Utility Controllers
RTU32, RTU32R and RTU32E

User Manual

Version 1.55-1 / May 2014 / Doc 40209

BRODERSEN
simplifying systems

Table of Content

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
Fax: +45 45 35 26 29
sales@brodersen.com
www.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 Powerful RTU, PLC and Industrial Controller with stand-alone functionality or integrated configurations

The Brodersen RTU32 series based on a 32-bit platform is a powerful RTU and PLC with leading edge functionality. As well as being a powerful 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.

2.2 Open Platform with Windows Embedded CE Operating System

The powerful platform is based on a fan less industrial PC platform with Windows Embedded CE Operating System. Anyone familiar with the Windows environment will find it easy to set up the RTU. Windows Embedded CE provides an open and adjustable platform with both the power and functionality required to control advanced industrial applications.

3 Getting Started

3.1 Configuration and default settings

The LAN interfaces on the RTU32 are default at delivery set as follows:

LAN1: 192.168.0.1, Subnet: 255.255.255.0
LAN2: DHCP

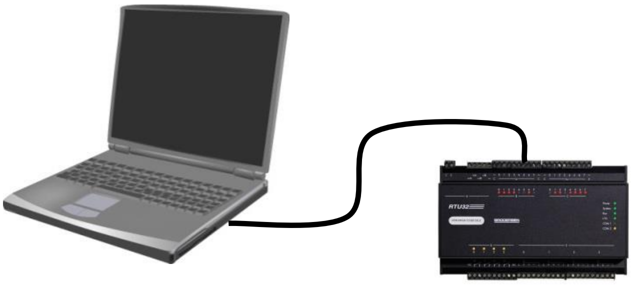
You can use LAN1 interface to configure the basic settings in the RTU32. The basic setting includes setup of network parameters as IP addresses, Subnet masks, default gateway, time settings, general username and password for accessing the web server pages and file system with FTP etc.

The configuration is done via web pages in the RTU32 Web Server. You must use your normal web browser to setup the basic settings for the RTU32. The web pages are optimized for MS Internet Explorer, Firefox and Chrome. If you are experiencing problems with using the latest MS Internet Explorer version, please enable Compatibility settings in the browser.

4 How to connect and prepare your PC to configure the RTU32

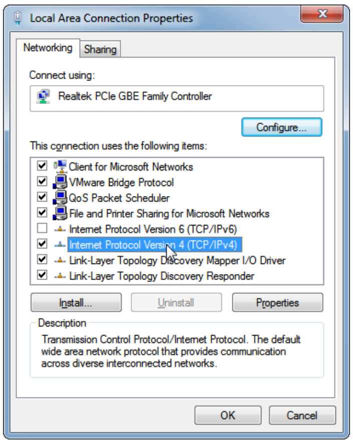
4.1 Wiring

Use the cross-wired or standard patch cable to connect your PC Ethernet interface to LAN1 Ethernet interface of the RTU32. Apply power to the RTU32 – make sure that you connect the correct power supply voltage to the RTU32. If you use have a 24-48VDC version, your power supply must be able to deliver at least minimum 2A if you have a standalone RTU32 configuration.



4.2 Setting up your PC Ethernet driver (Windows 7)

Your PC Ethernet driver must be setup to work in the same network segment as the LAN interface on the RTU32 you use. You must setup your PC Ethernet TCP/IP driver to work in the segment IP 192.168.0.x.
To setup the Ethernet driver select *start – Settings – Network Connections – Local Area Connections* and the following window will appear:



Local Area Connection Properties

Networking | Sharing

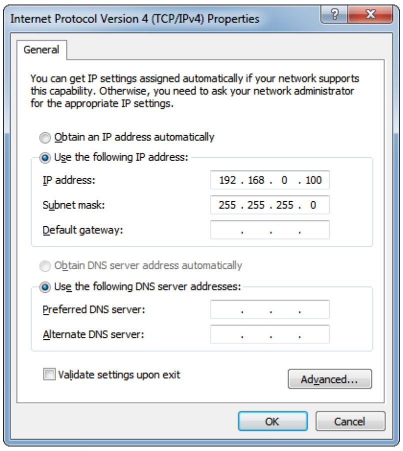
Connect using:
Realtek PCIe GBE Family Controller

This connection uses the following items:
- ☑ Client for Microsoft Networks
- ☑ VMware Bridge Protocol
- ☑ QoS Packet Scheduler
- ☑ File and Printer Sharing for Microsoft Networks
- ☐ Internet Protocol Version 6 (TCP/IPv6)
- ☑ Internet Protocol Version 4 (TCP/IPv4)
- ☑ Link-Layer Topology Discovery Mapper I/O Driver
- ☑ Link-Layer Topology Discovery Responder

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

Click on Internet Protocol 4 (TCP/IPv4), and click *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 192.168.0.100 as shown here below.

NOTE: If you want to re-configure your Ethernet TCP/IP driver back to initial settings, it is recommended that you note the settings before you change them for setting up the RTU32.



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 . 100
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

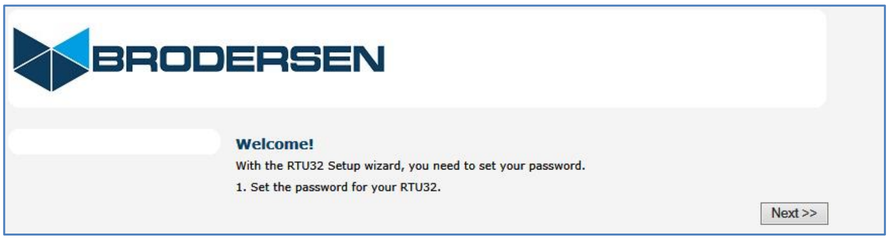
Now click *OK* and *OK* again at the Local Area Connection Properties window. Now your PC Ethernet driver is setup for communicating with the RTU32.
Now apply power to the RTU32. Wait for the RTU32 to boot up – takes approx. 30s. As a minimum the green Power and I/O LED on the front must be lit.

Alternative you can connect the RTU32 via LAN2 to your office network. LAN2 is default set to DHCP and will automatically obtain an IP address when powered up. You can find the RTU32 by scanning the network with the RTU32 Search tool which is installed with the Brodersen WorkSuite package. After you find the RTU32 you can connect via your browser.

4.3 First time configuration

Start your Internet browser and enter the IP address of the RTU32 in the address field:

Now the first web page will appear in your browser and you will be asked to enter a password.
Please store the entered password in a safe place for future use.



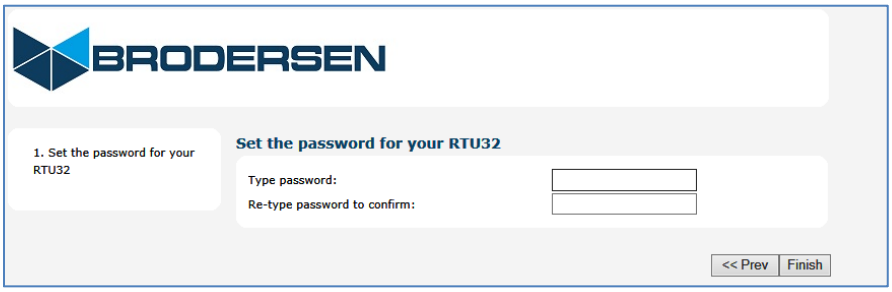
Welcome!

With the RTU32 Setup wizard, you need to set your password.

1. Set the password for your RTU32.

Click NEXT to continue.

Now you are asked to enter your administrator password. Note that you can always change it later. Username is fixed = admin.

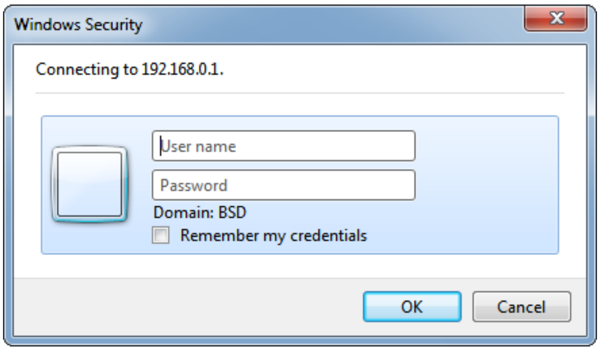


1. Set the password for your RTU32

Set the password for your RTU32

Type password:
Re-type password to confirm:

After typing in your password twice then select Finish to continue. Click Finish to save the settings and you will get asked to login to the RTU32 web server as shown below.



Connecting to 192.168.0.1.

Domain: BSD

□ Remember my credentials

User name is fixed = admin. Enter your password.

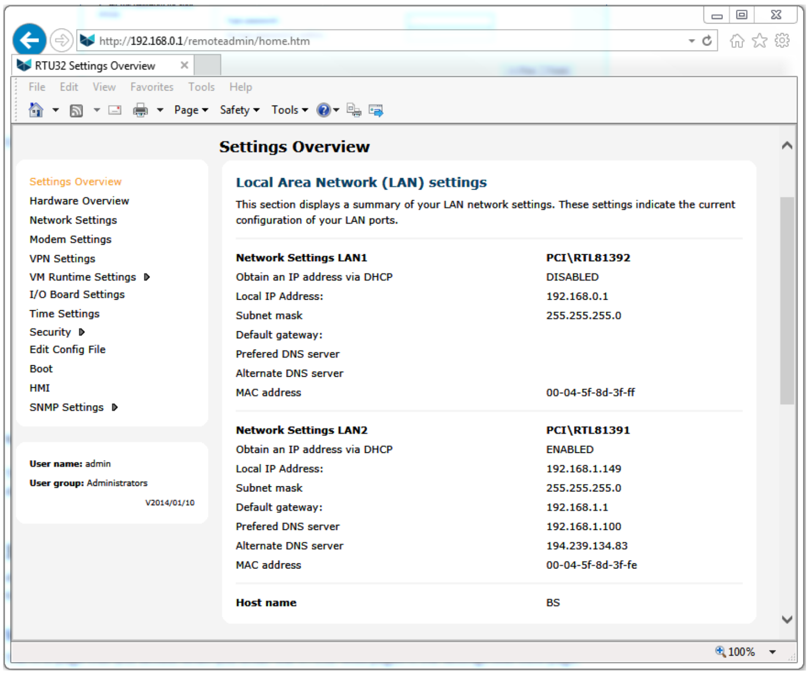
This is the login you will meet every time you try to login in the future.
After login in you will get to the first configuration web page.

5 Configuration parameters in the web pages

After the first time setting up the password, the main configuration page of the RTU32 will appear in your browser. Next time you connect to the web pages you will be asked to enter credentials to get access.

5.1 Settings overview

The first page that you access when you enter the RTU32 Web pages is the Settings overview page.



Settings Overview

User name: admin

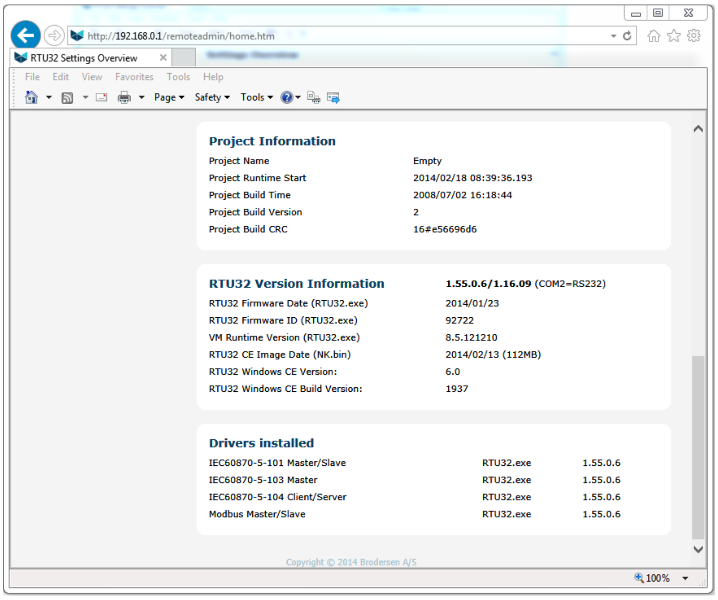
User group: Administrators

V2014/01/10

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	**PCI\RTL81392**
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	00-04-5f-8d-3f-ff
Network Settings LAN2	**PCI\RTL81391**
Obtain an IP address via DHCP	ENABLED
Local IP Address:	192.168.1.149
Subnet mask	255.255.255.0
Default gateway:	192.168.1.1
Prefered DNS server	192.168.1.100
Alternate DNS server	194.239.134.83
MAC address	00-04-5f-8d-3f-fe
Host name	BS



http://192.168.0.1/remoteadmin/home.htm
RTU32 Settings Overview
File Edit View Favorites Tools Help
Page Safety Tools
100%

Project Information

Project Name	Empty
Project Runtime Start	2014/02/18 08:39:36.193
Project Build Time	2008/07/02 16:18:44
Project Build Version	2
Project Build CRC	16#e56696d6

RTU32 Version Information	**1.55.0.6/1.16.09** (COM2=RS232)
RTU32 Firmware Date (RTU32.exe)	2014/01/23
RTU32 Firmware ID (RTU32.exe)	92722
VM Runtime Version (RTU32.exe)	8.5.121210
RTU32 CE Image Date (NK.bin)	2014/02/13 (112MB)
RTU32 Windows CE Version:	6.0
RTU32 Windows CE Build Version:	1937

Drivers installed

IEC60870-5-101 Master/Slave	RTU32.exe	1.55.0.6
IEC60870-5-103 Master	RTU32.exe	1.55.0.6
IEC60870-5-104 Client/Server	RTU32.exe	1.55.0.6
Modbus Master/Slave	RTU32.exe	1.55.0.6

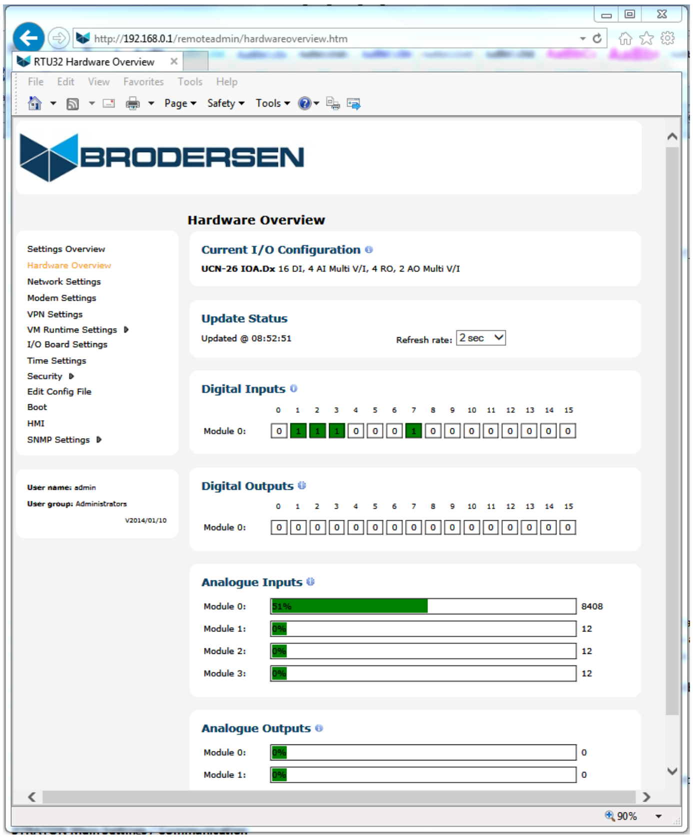
Copyright © 2014 Brodersen A/S

On this page you will get an overview of the network settings and version of the primary system files and drivers. If you are contacting your support office or distributor, you will always be asked for the software versions numbers.

IMPORTANT NOTE: The local IP will report 0.0.0.0 if the LAN port is not active (not connected/no connection). Check the Network Settings page for getting the last saved network settings.

5.2 Hardware Overview

The hardware overview page shows you online the current I/O configuration. It lists the physical connected I/O Expansion modules and you are able to read the status of each input or output. In the example below the RTU32 is equipped with internal I/O and input 1, 2, 3 and 7 are activated.



Hardware Overview

User name: admin
User group: Administrators
V2014/01/10

Current I/O Configuration

UCN-26 IOA.Dx 16 DI, 4 AI Multi V/I, 4 RO, 2 AO Multi V/I

Update Status

Updated @ 08:52:51 Refresh rate: 2 sec

Digital Inputs

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Module 0:	0	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0

Digital Outputs

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Module 0:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

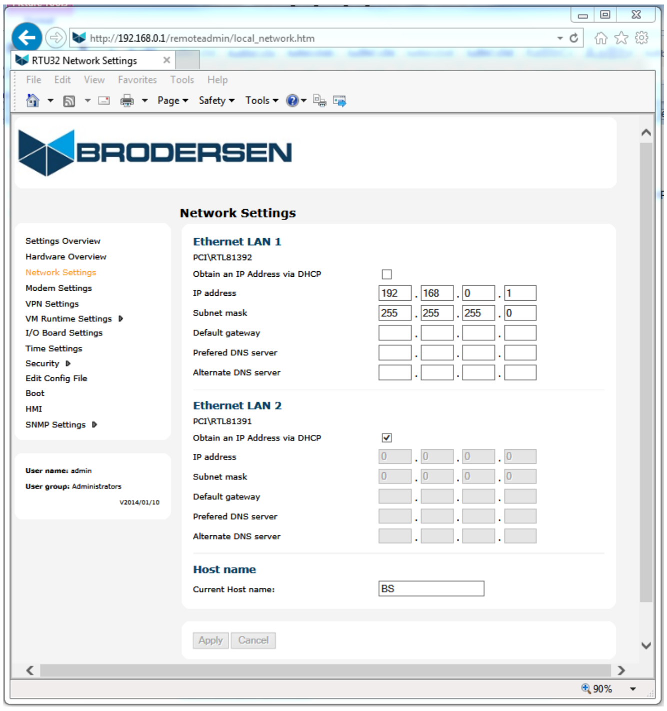
Analogue Inputs

Module 0:	51%	8408
Module 1:	0%	12
Module 2:	0%	12
Module 3:	0%	12

Analogue Outputs

Module 0:	0%	0
Module 1:	0%	0

5.3 Network settings



Network Settings

Ethernet LAN 1
PCI\RTL81392
Obtain an IP Address via DHCP □
IP address 192 . 168 . 0 . 1
Subnet mask 255 . 255 . 255 . 0
Default gateway . . .
Prefered DNS server . . .
Alternate DNS server . . .

Ethernet LAN 2
PCI\RTL81391
Obtain an IP Address via DHCP ☑
IP address 0 . 0 . 0 . 0
Subnet mask 0 . 0 . 0 . 0
Default gateway . . .
Prefered DNS server . . .
Alternate DNS server . . .

Host name
Current Host name: BS

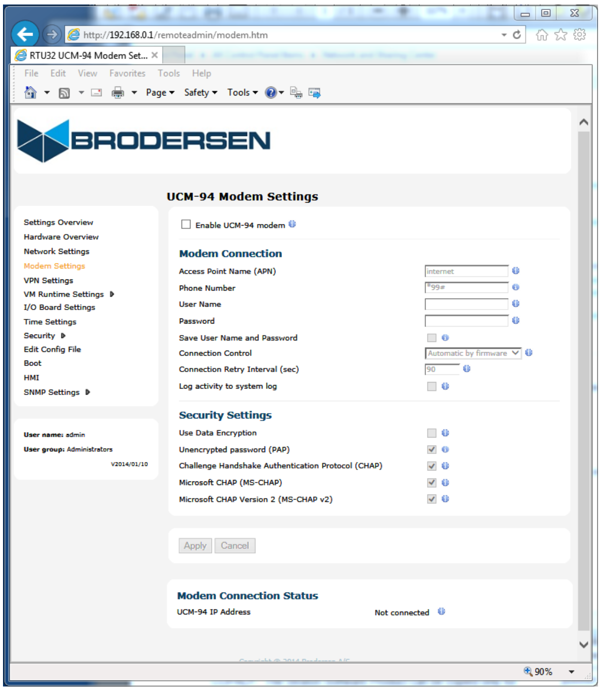
User name: admin
User group: Administrators
V2014/01/10

On the network setting page you can change the LAN1 and LAN2 settings to fit your local network. You must assign fixed IP addresses to gain access to the RTU32 with your browser, FTP client or Brodersen WorkSuite in your LAN network. If you do not know about the IP addresses and Subnet masks in your local network, please contact your local network manager or IT department for help.
You can also change the RTU Host Name. Default the host name of the RTU is BS.

After entering new settings, click Apply to save. Note that the new settings will NOT be activated before you reboot the RTU32. Use the Boot function on the menu at the left side of the page.

5.4 Modem Settings

If you use Brodersen UCM-94 3G/GPRS modem you must configure the RTU32 to work with the modem.



UCM-94 Modem Settings

□ Enable UCM-94 modem

Modem Connection
Access Point Name (APN) internet
Phone Number *99#
User Name
Password
Save User Name and Password □
Connection Control Automatic by firmware
Connection Retry Interval (sec) 90
Log activity to system log □

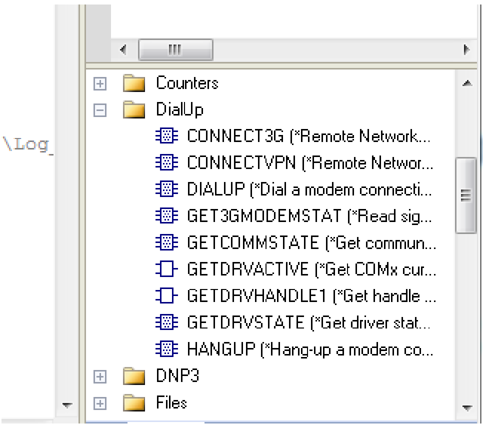
Security Settings
Use Data Encryption □
Unencrypted password (PAP) ☑
Challenge Handshake Authentication Protocol (CHAP) ☑
Microsoft CHAP (MS-CHAP) ☑
Microsoft CHAP Version 2 (MS-CHAP v2) ☑

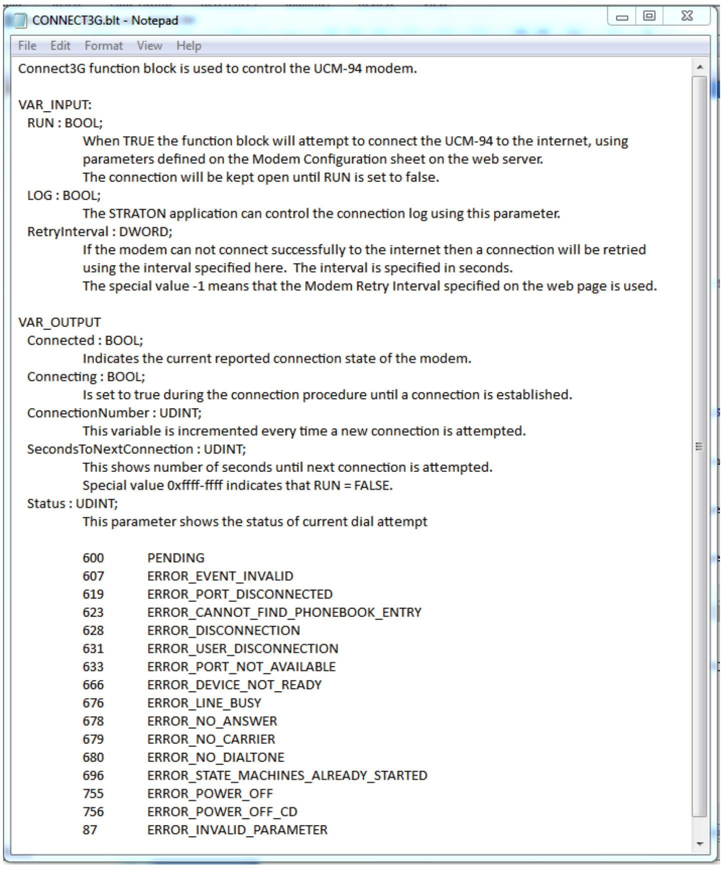
Modem Connection Status
UCM-94 IP Address Not connected

User name: admin
User group: Administrators
V2014/01/10

In order to ensure the modem is working you will need to enter all credentials correctly. The webpage include small help icons. Just put the cursor on them and you will get details for each point.

If you want to control the connection from the PLC application, you must select Connection Control from application. And then in your PLC application use the Function Block Connect3G placed in the Dial-up function group:


\Log_
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

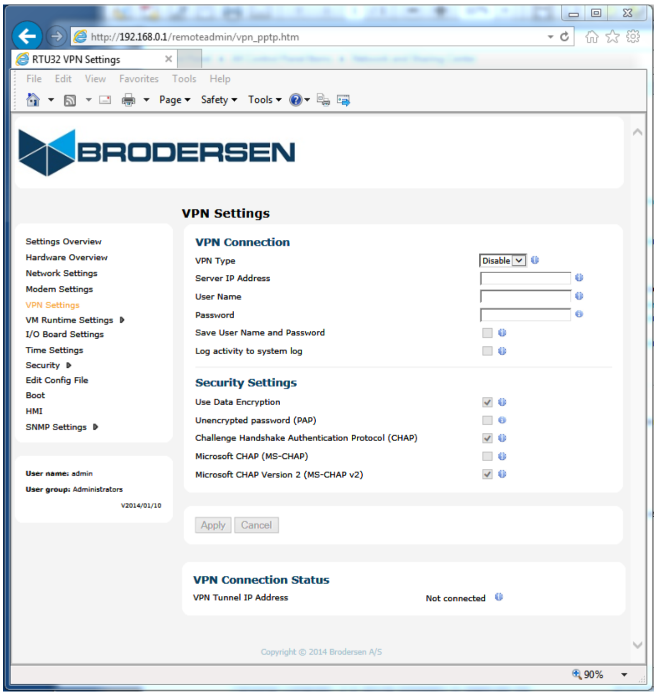


```
Connect3G function block is used to control the UCM-94 modem.

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 STRATON 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
```

You can at any time see the modem connection status of the modem at the bottom of the page. If it is connected it will show you the IP address assigned on the 3G/GPRS network.

5.5 VPN Settings

RTU32 include a VPN Client to support PPTP and L2TP (IPsec). And the VPN can be enabled and configured on the VPN page. Setting must be adjusted to meet the VPN Server settings.



User name: admin
User group: Administrators
V2014/01/10

VPN Settings

VPN Connection

VPN Type	Disable
Server IP Address	
User Name	
Password	
Save User Name and Password	□
Log activity to system log	□

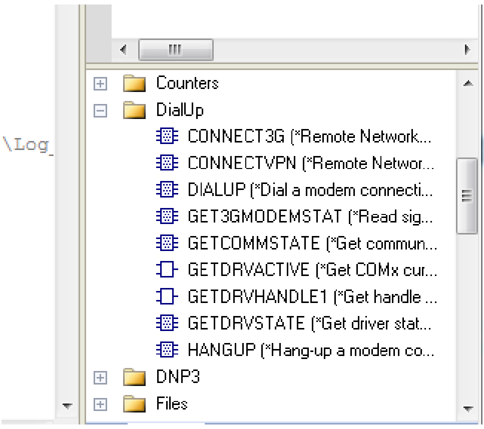
Security Settings

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

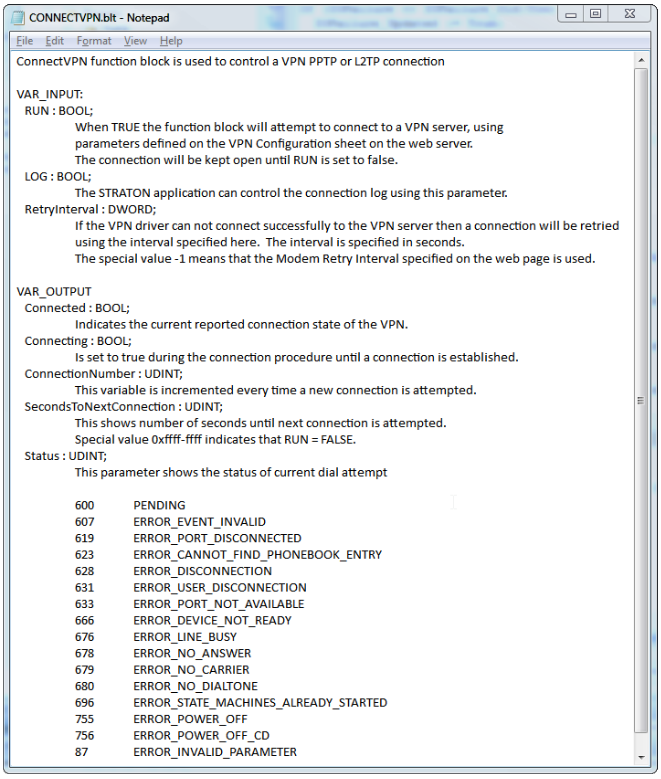
VPN Connection Status

VPN Tunnel IP Address: Not connected

In order to ensure the modem is working you will need to enter all credentials correctly. The webpage include small help icons. Just put the cursor on them and you will get details for each point.
Controlling connect of the VPN Client shall always be done from the PLC application. Use the Function Block ConnectVPN found in the Dial-up function group to manage the VPN connection.



- 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



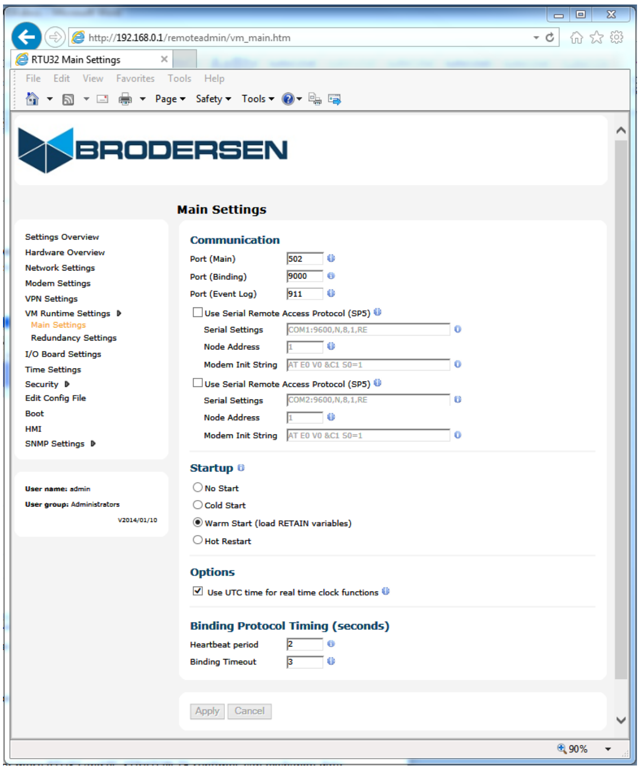
```
ConnectVPN function block is used to control a VPN PPTP or L2TP connection

VAR_INPUT:
  RUN : BOOL;
              When TRUE the function block will attempt to connect to a VPN server, using
              parameters defined on the VPN Configuration sheet on the web server.
              The connection will be kept open until RUN is set to false.
  LOG : BOOL;
              The STRATON application can control the connection log using this parameter.
  RetryInterval : DWORD;
              If the VPN driver can not connect successfully to the VPN server 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 VPN.
  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: It is not recommended to use IPsec level VPN (like L2TP) over 3G/GPRS network. It requires a very good and reliable 3G connection to work with IPsec type VPN. The reason is that if the Server feels any kind of latencies in the communication during pre-shared key updates, the Server will consider this as "not allowed interference" and the key renewal procedure will fail.

5.6 VM Runtime Settings / Main Settings

The VM (Virtual Machine) is the name for the PLC application runtime executer. In other words the VM runtime is to be considered as the PLC runtime.

RTU VM (Virtual Machine) settings are divided into 2 main sections/web pages. The basic setting page covers all general settings which are used for configuration during start-up. The second webpage covers RTU/PLC Redundancy settings.



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 COM2:9600,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

Options

☑ Use UTC time for real time clock functions

Binding Protocol Timing (seconds)

Heartbeat period 2
Binding Timeout 3

User name: admin
User group: Administrators
V2014/01/10

5.6.1 Communication

Configuration of basic start-up settings, communication ports and enabling of Remote Access Secure Protocols. Hold the cursor over the small icon ⓘ and you will get online help.

Port (Main)
The Main port is used for Brodersen WorkSuite communication for configuration and debugging. Default setting is 502.
This port is also used for the ModbusTCP Server.

Port (Binding)
Binding Port is used by the event based binding protocol, where two or more RTU32 and PC STRATON T5 runtimes can exchange data. Default setting is 9000.

Port (Event log)
This is the port where the RTU32 System Log can be viewed - either by connecting using telnet or the Brodersen Event Viewer tool. Default setting is 911.

Enable Serial Remote Access Protocol (SP5)
This protocol is to be used for remote serial communication from the Brodersen WorkSuite and is particularly well adapted to communication systems with severe constraints such as radios and modems. If you e.g. have the RTU32 placed on a remote site connected via serial communication via multi-drop radio or line connection – or even dial connection, you are able to remotely connect

to the RTU32 with the Brodersen WorkSuite and work online as 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 bauds, 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 RTU32 on a peer to peer network is 1. The address must be a value from 1 to 32767.

Modem Init String
Here you enter initialization string (AT commands) when using a modem.
RTU32 default is: AT E0 V0 &C1 S0=1

5.6.2 Start-up

The start-up option specifies how the PLC application is started when the RTU32 is powered up. The possible options are:
No Start – PLC application is NOT started when booting the RTU32.
Cold Start – PLC application is started without loading retained variables.
Warm Start – PLC application is started and retained variables are loaded (default).
Hot Restart – Only supported with battery backed RAM option and in Redundancy configuration!

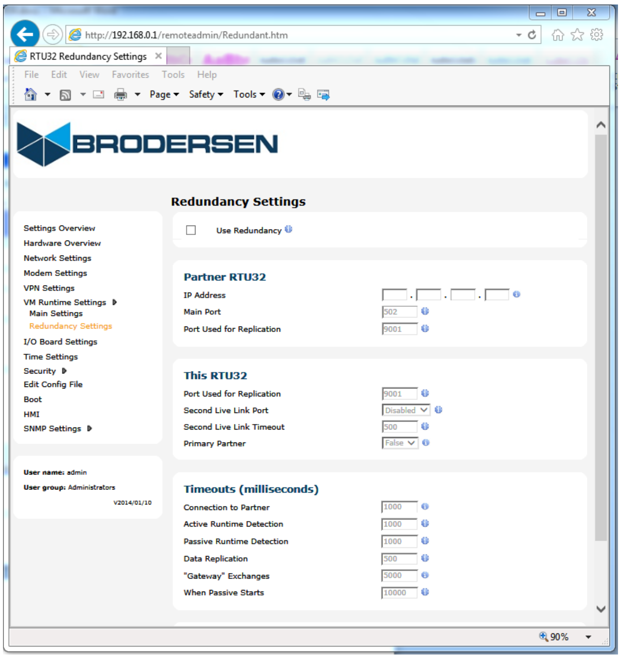
5.6.3 Options

Use UTC time for PLC real time functions
If enabled it instructs the all PLC 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 which is adjusted to daylight savings options.

NOTE: Drivers that is implemented via PLC Functions – like IEC60870 Driver – will use PLC RTU to time stamp events! Other drivers may take standard windows time or whatever time is defined in the applicable standards.

PLC Main Settings / Binding Protocol Timing (seconds)
When using binding on slower network connection like some optical fibre modems or GPRS/3G wireless connections, you will need to adjust the timing for the Binding driver.
The possible timing adjustments are:
Heartbeat period – how often a heartbeat is sending to check the binding link connection.
Binding Time-out – time-out on link check and general binding event.

5.7 VM Runtime Settings/ Redundancy Settings



User name: admin
User group: Administrators
V2014/01/10

Redundancy Settings

☐ Use Redundancy

Partner RTU32

IP Address	. . .
Main Port	502
Port Used for Replication	9001

This RTU32

Port Used for Replication	9001
Second Live Link Port	Disabled
Second Live Link Timeout	500
Primary Partner	False

Timeouts (milliseconds)

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

On the Redundancy Settings page you can enable and configure the RTU32 PLC Redundancy function.
The RTU PLC runtime provides you with a redundancy runtime features that enables configuration and control of redundant configured pairs of RTU32s in an application.

The redundant function is much more detailed described in the RTU32 PLC Redundancy User's Manual that you can find on the WorkSuite disc and available on our web. In this section you will find limited details of the specific settings on the webpage.

5.7.1 Redundancy configuration

In redundant configuration 2 x RTU32 works as sets of RTU/PLC's. You assign on each of the 2 modules the IP address on the partner RTU32, and adjust the timeout settings for the configuration. The RTUs are connected together with an Ethernet and a serial live-link channel using one of the serial ports on each RTU. The Ethernet link is used for online replication of the application and current state of the runtime application. The serial link is used for monitoring that the partner is available. Both connections are mandatory in a redundancy setup and it also ensures that one-fault will not make the redundancy runtime setup to fail.

When you program the RTU32 you will make one PLC runtime application, and it will automatically be copied to the partner RTU32. You will in Brodersen WorkSuite find special redundancy Functions for monitoring and controlling the redundant setup.

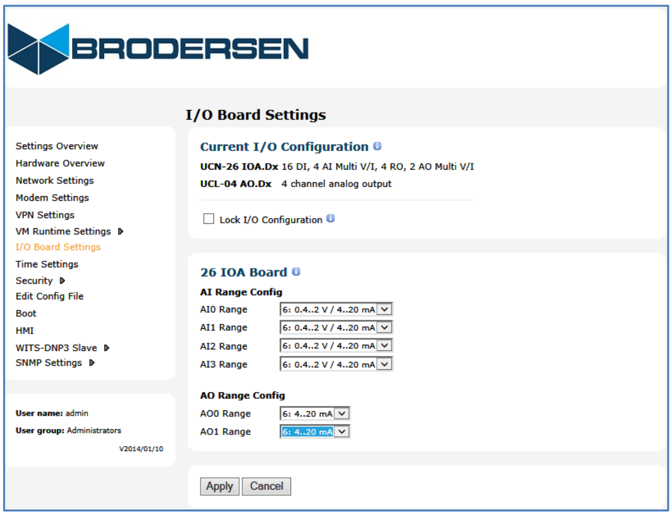
Redundancy function supported can be scribed as an active – passive runtime setup. The RTU that is power up first will be running as active runtime and the RTU booted last will automatically go in passive wait mode. And it will monitor the active and in case it detects that the active is not present or answering, it will start as active runtime in a hot-standby switch-over. The minimum switch time is around 200msec, but a typical redundancy application that also includes monitoring of communication channels will be around 1-2 sec. The delay is with monitoring of communication channels mainly dependent on the driver and channel timeouts – as a switch because of lost communication channel may not be detected before the actual driver times out.

5.8 I/O Board Settings

On this page you can:
- Read the actual I/O configuration on the RTU32.
- Lock an I/O configuration
- Configure the analogue inputs and outputs integrated in the RTU32.

5.8.1 Current I/O Configuration

Here the actual I/O configurations are listed. On the example below you will see that the configuration of I/Os includes RTU32 integrated I/O board (26IO) and a connected I/O Expansion module with 4 analogue outputs.



I/O Board Settings

Current I/O Configuration

UCN-26 IOA.Dx 16 DI, 4 AI Multi V/I, 4 RO, 2 AO Multi V/I
UCL-04 AO.Dx 4 channel analog output

☐ Lock I/O Configuration

26 IOA Board

AI Range Config

AI0 Range	6: 0.4..2 V / 4..20 mA
AI1 Range	6: 0.4..2 V / 4..20 mA
AI2 Range	6: 0.4..2 V / 4..20 mA
AI3 Range	6: 0.4..2 V / 4..20 mA

AO Range Config

AO0 Range	6: 4..20 mA
AO1 Range	6: 4..20 mA

User name: admin
User group: Administrators
V2014/01/10

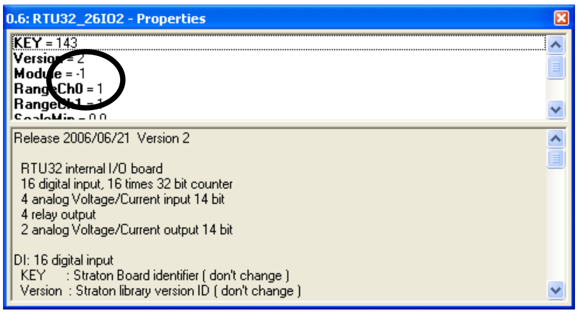
Default the I/O configuration is unlocked – it means that the actual I/O configuration is read every time the RTU32 is booted and when the PLC Runtime application is started.
By enabling lock I/O configuration, the actual I/O configuration is locked and stored, and NOT read at boot and/or application start-up. This is an advantage to use this function as you will then be able to trace a disconnected or defective I/O module in the RTU32 System Log.

5.8.2 26 IOA Board (only relevant if RTU32 include integrated I/Os)

The integrated I/O board type 26IOA includes both analogue inputs and outputs. The type and range of these I/O can be configured in 2 ways:

- On the webpage I/O Board Configuration
- Using the I/O Board Configurator in Brodersen WorkSuite. If you use the I/O Board Configurator you must add the 26IOA Board into the list of I/O Boards used – see details in the I/O Driver section.

NOTE that the I/O Board Configurator has the **highest priority.** That means that if you define the analogue ranges in PLC using the I/O Board configurator, the settings on the webpage will be completely ignored. The reasons for this is that it provides backwards compatibility with older PLC applications and you have the possibility to store these settings in your application program instead of settings them up on the webpage.
If you for some reason want to change this and only use the webpage configuration, you have to remove the Board I/O for 26IOA in PLC and reboot the RTU32. The webpage I/O configuration for AI/AO is only read at boot up.
NOTE: If you in general use Profile I/O defined variables for the physical I/O, you can use I/O Board Configuration to define the setting for the physical analogue I/O. In this case you MUST disable the digital and analogue output handling in the I/O Board configuration. This is done by setting Module = -1 for all outputs.



```
KEY = 143
Version = 2
Module = -1
RangeCh0 = 1
RangeCh1 = 1
ScaleMin = 0.0
```

```
Release 2006/06/21  Version 2

  RTU32 internal I/O board
  16 digital input, 16 times 32 bit counter
  4 analog Voltage/Current input 14 bit
  4 relay output
  2 analog Voltage/Current output 14 bit

DI: 16 digital input
  KEY      : Straton Board identifier ( don't change )
  Version  : Straton library version ID ( don't change )
```

And all variables used to write to the relevant outputs MUST be I/O Profile type. I/O Board input variables that is NOT disabled (Module >= 0) can be used as normal.

If you use the IEC60870 Configurator to setup IEC60870 Drivers it is recommended to configure the analogue ranges on the webpage – and not to use the Board I/O configuration! It can easily be confusing if you forget to provide the correct settings.

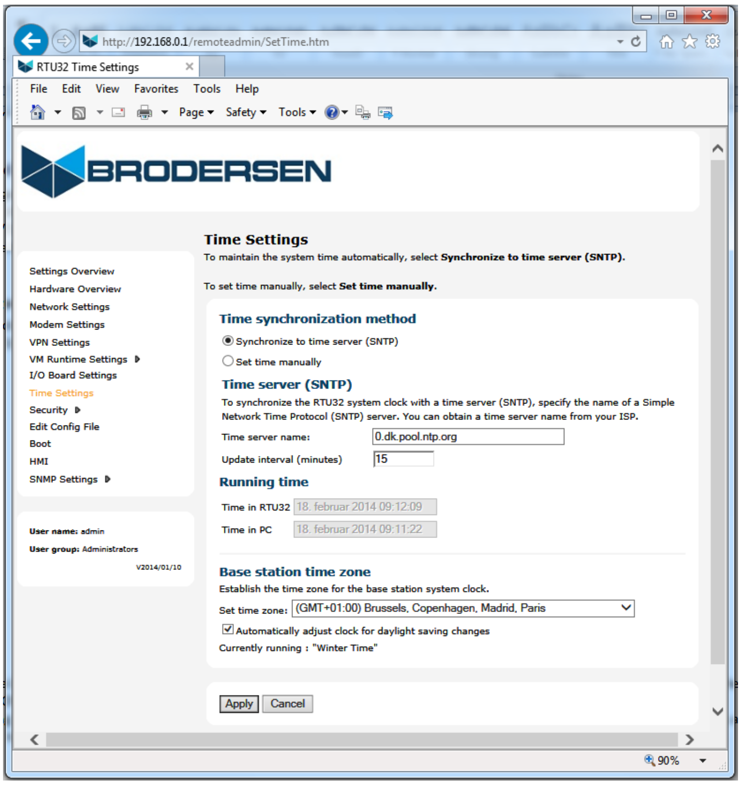
5.9 Time Settings for RTC – Real Time Clock

The time settings web pages is used for configure and adjusting the RTU32 real-time clock.

5.9.1 Time Settings

You can choose to:

- Enter then time manually
- Copy the time from your PC to the RTU32
- Set the time from a network time server using SNTP. If you use the SNTP you have to enter the domain name (like www.example.com) or IP address of the time server. If you enable SNTP clock synchronisation, the RTU32 will also work as a SNTP Time Server. That means that you can let one RTU32 in your application be the local time server being synchronised from a remote SNTP Server – and let the other RTU32 synchronise from the local RTU32 Time Server.



User name: admin

User group: Administrators

V2014/01/10

Time Settings

To maintain the system time automatically, select **Synchronize to time server (SNTP).**

To set time manually, select **Set time manually.**

Time synchronization method

● Synchronize to time server (SNTP)

○ Set time manually

Time server (SNTP)

To synchronize the RTU32 system clock with a time server (SNTP), specify the name of a Simple Network Time Protocol (SNTP) server. You can obtain a time server name from your ISP.

Time server name: 0.dk.pool.ntp.org

Update interval (minutes) 15

Running time

Time in RTU32 18. februar 2014 09:12:09

Time in PC 18. februar 2014 09:11:22

Base station time zone

Establish the time zone for the base station system clock.

Set time zone: (GMT+01:00) Brussels, Copenhagen, Madrid, Paris

☑ Automatically adjust clock for daylight saving changes

Currently running : "Winter Time"

NOTE: If you use time synchronisation via utility drivers like DNP3, IEC60870 or IEC61850, you shall configure the time settings to Set Time Manual.

5.9.2 Base Station time zone

If you use local time you have to set the time zone correctly in order to get the correct time stamps. Default it is setup for Central European Time (CET). You can also select to support automatic daylight saving adjustment. In this case it is using the Windows CE clock handling and is following the standard Microsoft adjustment procedures. Be careful to take this in consideration if you are working with IEC60870 drivers and uses Clock Synchronisation.
Note that it is required to add all necessary network settings. E.g. it requires a DNS address if you configure domain address as Time Server (SNTP Server).

5.10 Security

The RTU32 supports setup of User Authentication for the access to the available Servers. And the settings apply for all Servers that include Webserver, FTP Server and Telnet Server.

The main Username for the Administrator user is fixed to “admin”. And when you connect to the Webserver the first time you are asked to define the main Administrator password. You must store these credentials as you will need them to access the RTU.

Three levels of users with different configuration and access rights are configurable:

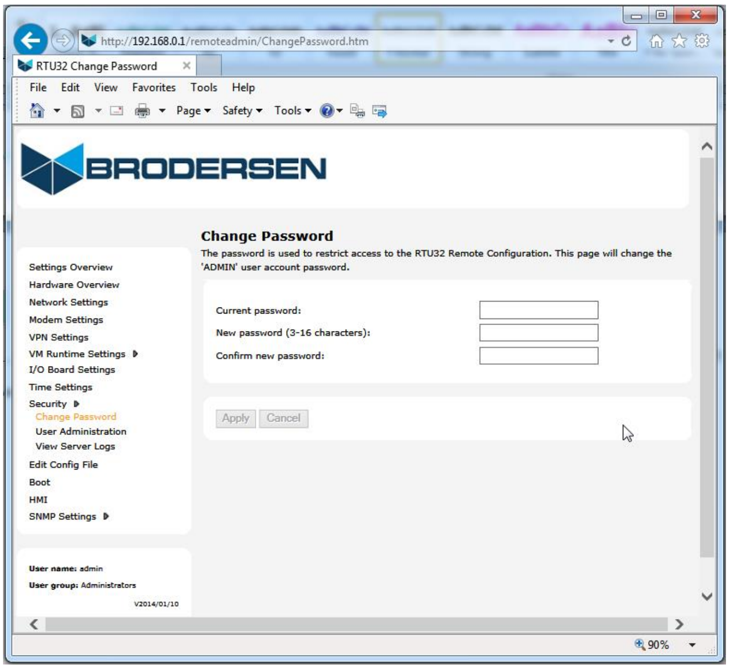
User type	**Access rights**
Administrator	Full access and configuration
SuperUser	Full Access and configuration except for the Security pages (read-only)
Guest	Read only access only

You can configure up to 10 user of each level.

Only Administrator user can configure the security authentication settings as described here.

5.10.1 Security / Change Password

You can here change the password for the main Administrator user only.



Change Password

The password is used to restrict access to the RTU32 Remote Configuration. This page will change the 'ADMIN' user account password.

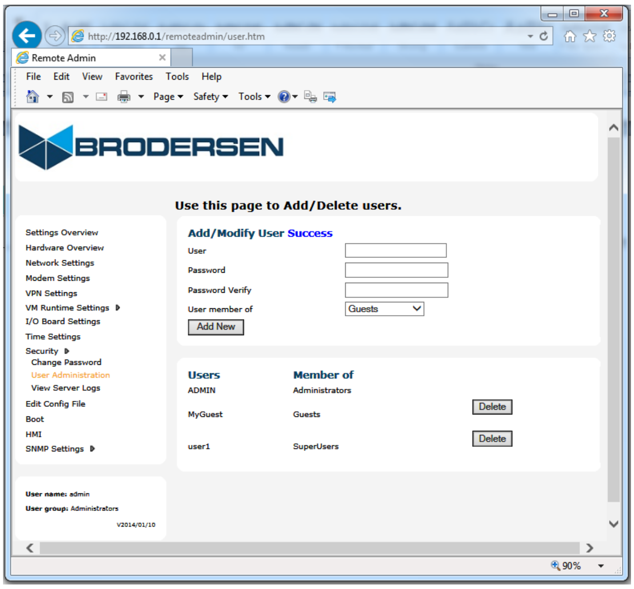
Current password:	
New password (3-16 characters):	
Confirm new password:	

User name: admin
User group: Administrators
V2014/01/10

5.10.2 User Administration

The User Administration page is used by the main Administrator user to manage users. You can add, change or remove users. All created users except for the main user will be listed on this page.

There are no specific rules for the username and passwords except that they are case sensitive.



Use this page to Add/Delete users.

Add/Modify User Success

User	
Password	
Password Verify	
User member of	Guests

Users	Member of	
ADMIN	Administrators	
MyGuest	Guests	Delete
user1	SuperUsers	Delete

User name: admin
User group: Administrators
V2014/01/10

5.10.3 View Server Logs

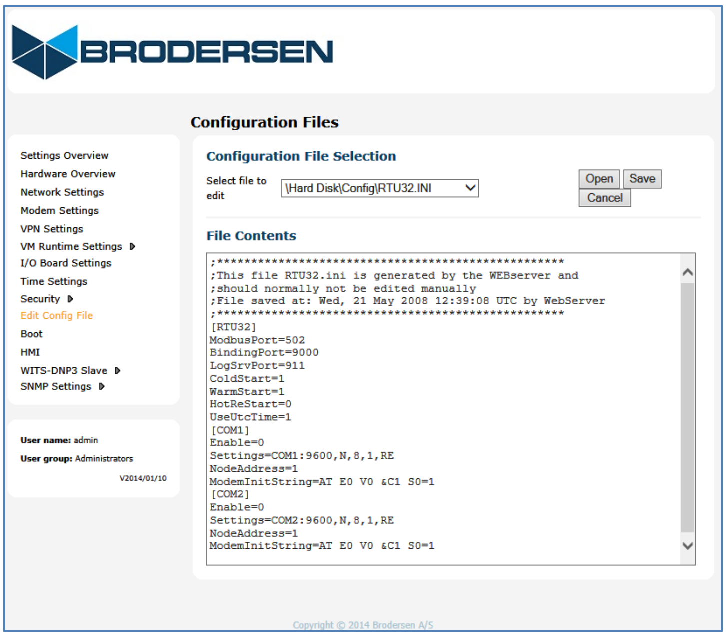
There are some logs available for monitoring which is based provided by the basic OS. They provide some limited information of access to the Servers.

5.11 Edit Config File

The Edit Config File page is a simple text file editor used for making changes and adjustments in text based files in the RTU32 file system. The editor will automatically list all files with the extension ".txt", ".dat" and ".ini" that is placed in the RTU ../Hard Disk/Config/.. folder.

Default a number of files will be listed if you view "Select file to edit". They are all files related to configuration of the RTU and reflect the actual settings on the webpages. These files should NOT be changed!

After opening a file, you can edit it like in any other text editor. Select save to save the configuration file on the RTU32.



Configuration Files

Configuration File Selection

Select file to edit: \Hard Disk\Config\RTU32.INI

File Contents

```
;***************************************************
;This file RTU32.ini is generated by the WEBserver and
;should normally not be edited manually
;File saved at: Wed, 21 May 2008 12:39:08 UTC by WebServer
;***************************************************
[RTU32]
ModbusPort=502
BindingPort=9000
LogSrvPort=911
ColdStart=1
WarmStart=1
HotReStart=0
UseUtcTime=1
[COM1]
Enable=0
Settings=COM1:9600,N,8,1,RE
NodeAddress=1
ModemInitString=AT E0 V0 &C1 S0=1
[COM2]
Enable=0
Settings=COM2:9600,N,8,1,RE
NodeAddress=1
ModemInitString=AT E0 V0 &C1 S0=1
```

User name: admin
User group: Administrators
V2014/01/10

You may add your own files to the ../Hard Disk/Config/.. folder. It means also that you have to create and place your own file using e.g. FTP to a RTU32. The WorkSuite support functions for reading ".ini" files and can e.g. be used for setting up variable values – set points, alarm levels etc. from the web interface.

The idea is that you can add your own file to the RTU32 that include parameters used e.g. in a running PLC runtime application. Both the Brodersen WorkSuite and other developing tools provide options for read and write to text files.
Or you can save your application history or log book in a file. If you want to keep record of changes and updates of your application(s), you can enter it into to a text file and keep it on the RTU.

The file RTU32_MAN.ini file does include some advanced settings for handling QueueEvents for dedicated applications. And it also supports the possibility to decrease or increase the maximum retain buffer size. Default size is 10kB and should not be changed if you use any PLC runtime functions/features that use NVRAM. Contact Brodersen support for details.

NOTE: For the normal user the RTU32 system files needs no changes – keep them unchanged. And do not edit the text files default placed in this folder, unless you are absolute sure what impact it has.

5.12 Boot

After changing parameters on the configuration pages, you must unless otherwise is defined, run the reboot function to activate the changes.
After booting the RTU32 you can start working with Brodersen WorkSuite to create you application programs.

5.13 HMI

The RTU HMI is the link for monitoring Webserver HMI pages created in the Brodersen WorkSuite. The details of how to create and download HMI files to the RTU32 can be found in section 7.

5.14 SNMP settings

The SNMP setting page covers the basic settings for the SNMP Agent driver in the RTU32.

The RTU32 has support for:

- The basic Windows Embedded CE operating system SNMP Agent Driver which report information of network interfaces, network communication statistics etc. All implemented by Microsoft©.
- RTU32 SNMP Alarm Extension Agent. The RTU32 SNMP Alarm extension agent is used for application specific Trap alarming, Get/GetNext and Set functions and can be linked direct to physical I/Os and to STRATON via the basic I/O database in the RTU32. The SNMP function may not be available on all RTU32 platforms. So far RTU32, RTU32R and RTU32E support SNMP.

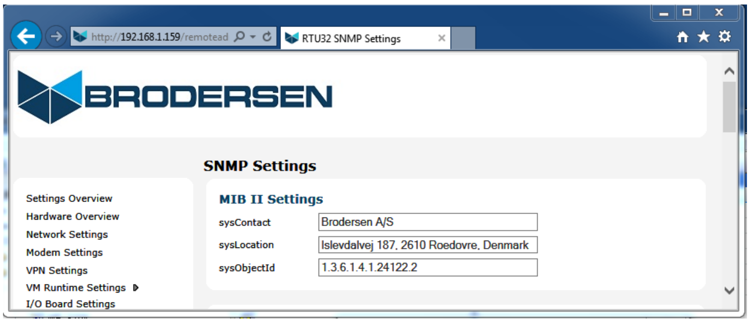
The general Windows CE SNMP Agent is always enabled in all RTU32 types. The enhanced SNMP Extension Agent used for alarming via Traps, monitoring and control is ONLY supported if you have bought a RTU32 with this option enabled.

Details about the Extended SNMP Agent function can be found in the RTU32 **SNMP User Manual.**

The configuration parameters for the general SNMP Agent in RTU32 are:

5.14.1 MIB II Settings

MIBII is the basic Windows CE SNMP Agent. You can define the some information in the basic agent. This information is Contact, Location and ObjectId. NOTE: Do not remove the object id unless you have full control over the SNMP Manager settings.



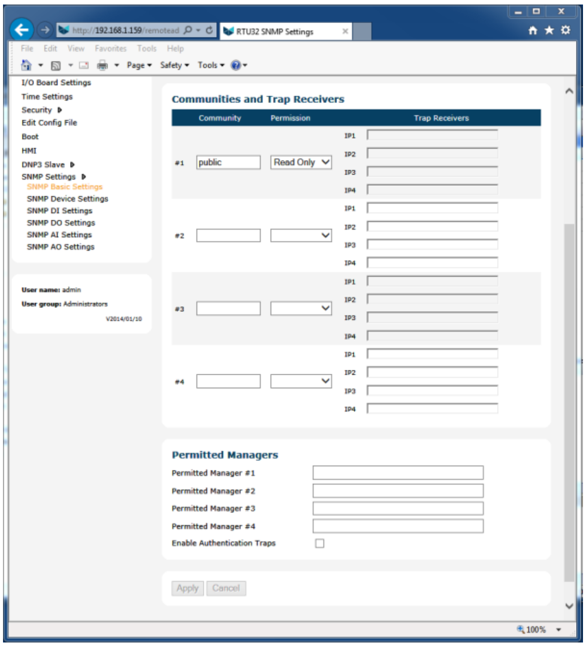
SNMP Settings

MIB II Settings

sysContact	Brodersen A/S
sysLocation	Islevdalvej 187, 2610 Roedovre, Denmark
sysObjectId	1.3.6.1.4.1.24122.2

5.14.2 Communication and Trap receiver

In this configuration area you are able to setup up to 4 communities with each 4 SNMP Managers IP addresses. Note that the community "public" is the default one and normally know by any SNMP Management software. In addition each community can be adjusted to different levels of access permissions.



User name: admin

User group: Administrators

V2014/01/10

Communities and Trap Receivers

	Community	Permission		Trap Receivers
#1	public	Read Only	IP1	
			IP2	
			IP3	
			IP4	
#2			IP1	
			IP2	
			IP3	
			IP4	
#3			IP1	
			IP2	
			IP3	
			IP4	
#4			IP1	
			IP2	
			IP3	
			IP4	

Permitted Managers

Permitted Manager #1

Permitted Manager #2

Permitted Manager #3

Permitted Manager #4

Enable Authentication Traps □

5.14.3 Permitted Managers

You can control access to Get/GetNext request and Set commands by adding Permitted Managers IP addresses. If no SNMP Managers are entered all Managers have access.
If you enable the Authentication Trap function, the permitted managers will get a Trap report if unauthorized has been attempted.

6 Getting Started with Brodersen WorkSuite tools

6.1 Introduction

The Brodersen WorkSuite Package includes all software required to configure and program Brodersen RTU32 Series of products currently covering RTU32, RTU32R, RTU32S and RTU32E.

The software components included in the package includes:

- Brodersen WorkSuite PLC 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.
  - RTU32 Series Remote Desktop tool
  - RTU32 Update tool for managing software updates for many RTUs over network connections.

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 your to begin working with configuration and programming of Brodersen RTU32 Series products.

6.2 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 with is included in the package. In the package you find also the WorkSuite License Certificate with License id number required to enter in the License utility after installing the software.
The standard license offers the possibility to create unlimited number of I/O in the RTU application program. All variables used for drivers are counted as I/Os.

6.3 Compatibility with STRATON WorkBench

The Brodersen WorkSuite Packages v1.0.x.x replaces all version of STRATON WorkBench. The WorkSuite is backwards compatible with all STRATON versions and support directly any older STRATON version 5.x, 6.x, 7.x and 8.x projects.

The Brodersen WorkSuite is in fact a Brodersen version of STRATON but provide a range of new facilities:
1. One installation process for WorkSuite install all software including license key drivers and all drivers, functions and Brodersen Tools supported by the RTU32 Series products.
2. It is no more needed 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 the WorkSuite and it makes the software version management must simpler for all parties.

Important Notes:
1. If you have existing licenses for STRATON WorkBench version 8.x it is 100% compatible with the WorkSuite version 1.0.0.x. You can in other words use your license for the new Brodersen WorkSuite. That goes for ALL tag count licenses.
2. If you have older license, please contact Brodersen at sales@brodersen.com or your local representative for updates prices. Please remember to include copy of your current License Certificate.

6.4 User documentation and training/guideline videos

When installing the Brodersen WorkSuite package a package of User Guide and Manual are installed as well. You get access to these documents from the Help menu within the WorkSuite tool.

WorkSuite has in addition 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 always recommend that you use the Brodersen Training/Guideline Videos. They are found by visiting:

www.youtube.com/brodersensystems

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

6.5 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 with is included in the package. In the package you find also the WorkSuite License Certificate with License id number required to enter in the License utility after installing the software.

The standard license offers the possibility to create unlimited number of I/O in the RTU application program. All variables used for drivers are counted as I/Os.

6.6 Installing Brodersen WorkSuite Package

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

7 Relevant RTU32 Series specific functions in WorkSuite

7.1 General

In general it is highly recommended to take a training course in RTU32 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 right away take advantage of all the RTU and WorkSuite powerful features. Contact your local distributor or us for RTU/WorkSuite training exercises.

Beside the basic WorkSuite standard PLC programming environment and EN/IEC61131-3 programming languages, there are some functions and features which are specific for the RTU32 Product Series.

7.2 WorkSuite I/O Drivers for RTU32

The WorkSuite I/O Drivers for RTU32 are used to access the physical inputs and outputs on the RTU32 and the connected I/O expansion modules. The procedure is in short that all inputs are read online when that 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 directly online I/Os.

All I/Os including status are stored in an independent memory mapped database in RTU32. The database is directly a mirror of the physical I/Os. And the Internal I/O Bus and the LocalBus for expansion modules is responsible for keeping this updated at any time.
The WorkSuite I/O drivers handle the access to this database. You can program your total application without any connected hardware. With this in mind you must be sure that you have the right I/O configuration on your RTU32 hardware when you try to run and debug your application program on a RTU32.

The database can optionally be access by other applications. A specific programmers API called WTOOL is then used. The RTU32 WTOOL SDK provides information about this. And all is found on in the WorkSuite installation package.

In order to control and differentiate new updates and versions of the I/O Profile driver in WorkSuite, the last digit in the names are used as version. You can keep older versions in STRATON by renaming the old hardware definitions before installing new ones (e.g. Rename “Brodersen” to “Brodersen_v0”).

7.3 I/O Database Access

The WorkSuite I/O driver for RTU32 is implemented to support two ways of accessing the I/O database in the RTU32. The two ways are called I/O Board and I/O Profiles.

I/O Board is used for declaring I/O variables after the physical I/O layout. You are able to add e.g. a 16DIO expansion module from a predefined board list. Not all I/O configurations are supported by the IO Board configuration – in this case use I/O Profile.

I/O Profiles gives the possibility to declare a single input or output and include additional scaling possibilities for analogue I/Os. Profiles are also used when you want to read status information from the RTU32 and read the DIL switches.

NOTE: Inputs can be declared by both Board and Profiles, but output must only be declared by one of them!

7.4 I/O database layout

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

ST:	Status word for I/O and read DIL switch settings
DI:	Digital inputs
AI:	Analogue inputs
YI	Virtual inputs
ZI	Auxiliary inputs (include also CI – 32bit counter inputs)
DO:	Digital outputs
AO:	Analogue outputs

YO: Virtual outputs

ZO: Auxiliary outputs (include also virtual outputs for resetting 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 (DINT)**. 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_08AIxP		8					UCL-08AIxP
RTU32_08DI	1						UCL-08DI.Ax
RTU32_08DO			1				UCL-08DO.Rx
RTU32_8DIO	1		1				UCL-8DIO.Px
RTU32_16CIS2	1				2X16	1 (RC)	UCL-16CIS.P
RTU32_16DI	1						UCL-16DI.Dx
RTU32_16DIO	1		1				UCL-16DIO.Px
RTU32_16DO			1				UCL-16DO.Rx
RTU32_26IO2	1	4	1	2	2X16	1 (RC)	UCN-26IOA
RTU32_32DI	2						UCL-32DI.Dx
RTU32_32DO			2				UCL-32DO.xx
RTU32_36IO	2		1				UCL-36IO.Px

7.5 Addressing example

To give a practical example, 2 different RTU32 I/O configurations with module addressing are shown in Appendix 2 and 3.

7.6 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.
Module 2: RTU32 DIL switch settings (switch 0 = bit 0, etc.).

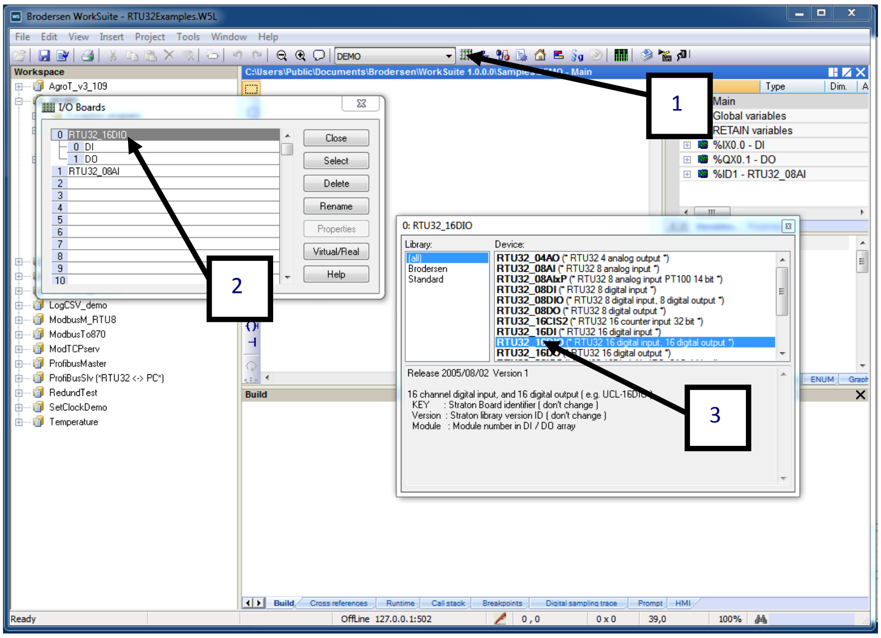
NOTE: The DIL switches located on the downwards side of the RTU32 are read in the status register (ST) module 2 (third word).

7.7 WorkSuite I/O Board Driver

The I/O Board driver makes it possible to declare I/Os in sections as the physical board layouts. I.e. it is possible to add all 32 digital inputs in one process using the I/O Boards driver. At declaration it automatically declares the variables and put them in the variable list.

Procedure for declaring board I/Os:

- Select Open I/O Board Configurator (1)
- Double click on the section 0 to add your first board (2)
- From the list of I/O board select the one you want (3)



0 RTU32_16DIO
0 DI
1 DO
1 RTU32_08AI

0: RTU32_16DIO

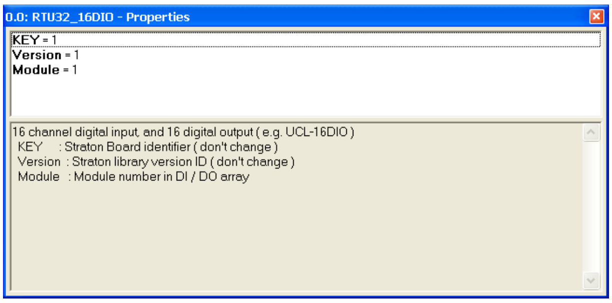
Library: (all), Brodersen, Standard

Device:
RTU32_04AO (* RTU32 4 analog output *)
RTU32_08AI (* RTU32 8 analog input *)
RTU32_08AIxP (* RTU32 8 analog input PT100 14 bit *)
RTU32_08DI (* RTU32 8 digital input *)
RTU32_08DIO (* RTU32 8 digital input, 8 digital output *)
RTU32_08DO (* RTU32 8 digital output *)
RTU32_16CIS2 (* RTU32 16 counter input 32 bit *)
RTU32_16DI (* RTU32 16 digital input *)
RTU32_16DIO (* RTU32 16 digital input, 16 digital output *)
RTU32_16DO (* RTU32 16 digital output *)

Release 2005/08/02 Version 1

16 channel digital input, and 16 digital output (e.g. UCL-16DIO)
KEY : Straton Board identifier (don't change)
Version : Straton library version ID (don't change)
Module : Module number in DI / DO array

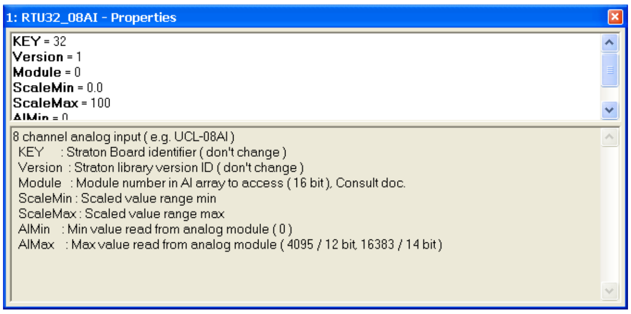
After the board is selected you will see it in your list. Now select properties to configure the board. If the board you have selected have different type of I/O they will be shown individually in its own line. Every data type require configuration.


0.0: RTU32_16DIO - Properties

KEY = 1
Version = 1
Module = 1

16 channel digital input, and 16 digital output (e.g. UCL-16DIO)
KEY : Straton Board identifier (don't change)
Version : Straton library version ID (don't change)
Module : Module number in DI / DO array

The properties for digital I/Os it quite simple. Only the Module address needs to be setup. Use the addressing example to determine the Module address in your application.

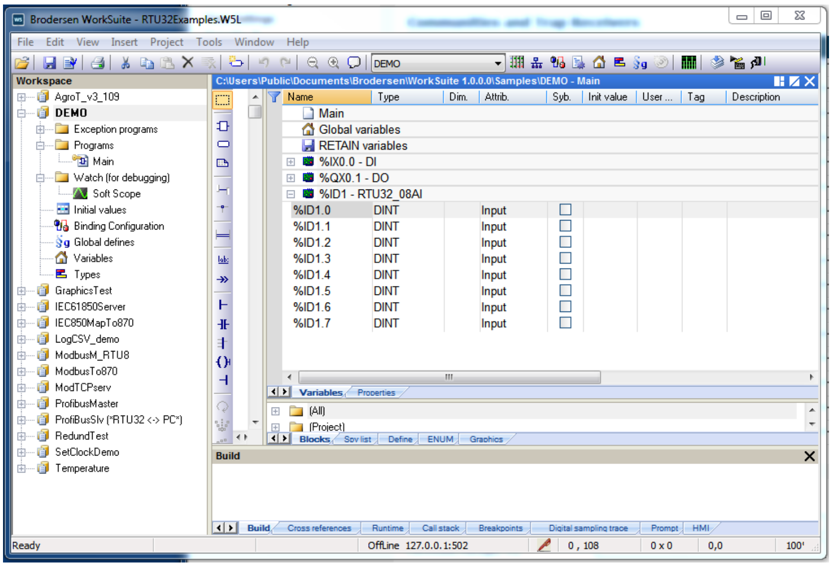

1: RTU32_08AI - Properties

KEY = 32
Version = 1
Module = 0
ScaleMin = 0.0
ScaleMax = 100
AIMin = 0

8 channel analog input (e.g. UCL-08AI)
KEY : Straton Board identifier (don't change)
Version : Straton library version ID (don't change)
Module : Module number in AI array to access (16 bit). Consult doc.
ScaleMin : Scaled value range min
ScaleMax : Scaled value range max
AIMin : Min value read from analog module (0)
AIMax : Max value read from analog module (4095 / 12 bit, 16383 / 14 bit)

The properties for analogue I/O include more configurations. Beside the Module address of the first analogue channel, you can configure the scaling of the analogue I/Os. Please note that the scaling will cover all I/O in the actual board section. On some board I/Os it is possible to set the range of every input and output individually. This is e.g. the case for Internal I/Os and newer analogue expansion modules.

After setting up the boards, you will find the all the board I/Os in the variable list.



Name	Type	Dim.	Attrib.	Syb.	Init value	User ...	Tag	Description
Main								
Global variables								
RETAIN variables								
%IX0.0 - DI								
%QX0.1 - DO								
%ID1 - RTU32_08AI								
%ID1.0	DINT		Input	□				
%ID1.1	DINT		Input	□				
%ID1.2	DINT		Input	□				
%ID1.3	DINT		Input	□				
%ID1.4	DINT		Input	□				
%ID1.5	DINT		Input	□				
%ID1.6	DINT		Input	□				
%ID1.7	DINT		Input	□				

The I/Os are now ready to be used in your runtime application program.

For some of the analogue I/Os you will find some qualifier input which is used for status information. They report e.g. underflow and overflow.

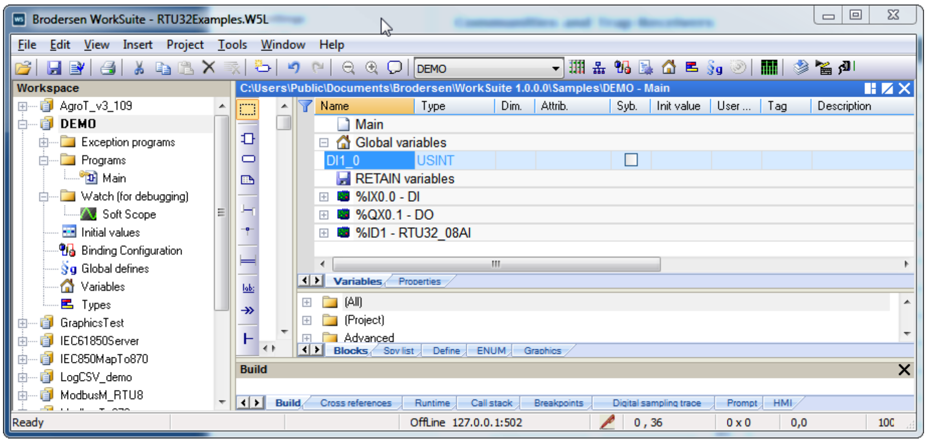
7.8 STRATON I/O Profiles Driver

The I/O Profile driver is used for declaring I/Os one by one directly in the variable list – or by using the Profile Editor. Any I/O digital or analogue can be declared this way. If you want status from RTU32 basic database or read the DIL switch settings, you must always use the I/O Profile driver.

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

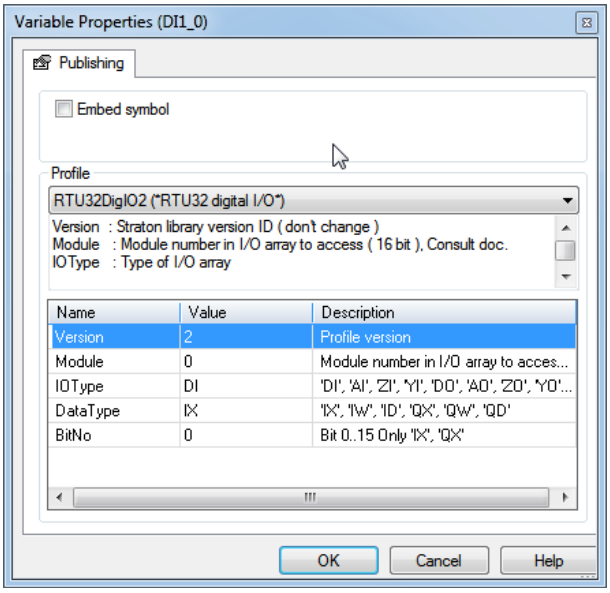
In the Variable list you add new variable under the Global variable structure. It is recommended to use Global Variable structure to have access to the I/O from any program in your STRATON project.

Double click on the variable name and enter a proper name for your I/O – e.g. DI1_0 as for the first digital input bit in the second DI module. Set the variable type to WORD.



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								

Right click on the DI1_0 variable and select Properties:



□ Embed symbol

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'

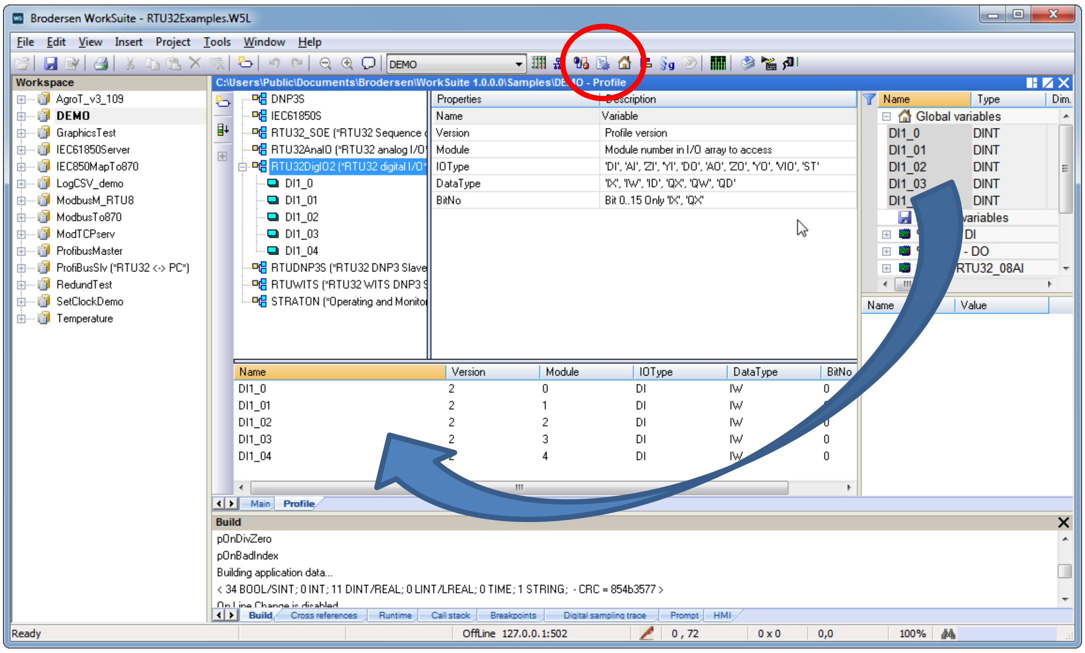
Select “RTU32DigIO” for digital I/Os, set Module = 1 (second digital input module (read: Word)).
Select “DI” for Digital Input register.
Select “IW” for Input Word.
Say OK to store the settings.

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 and DIL switches.

7.8.2 Option 2 - Declaring I/O Profiles using the Profile Editor

Another and often much faster way to declare an I/O Profile for RTU32 Series I/O is to use the Profile Editor in Brodersen WorkSuite.

The Profile Editor is a more user friendly way to setup several drivers including the Brodersen I/O LocalBus driver.



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	Dim.
DI1_0	DINT	
DI1_01	DINT	
DI1_02	DINT	
DI1_03	DINT	
DI1_	DINT	

Name	Version	Module	IOType	DataType	BitNo
DI1_0	2	0	DI	IW	0
DI1_01	2	1	DI	IW	
DI1_02	2	2	DI	IW	
DI1_03	2	3	DI	IW	0
DI1_04		4	DI	IW	0

```
pOnDivZero
pOnBadIndex
Building application data...
< 34 BOOL/SINT; 0 INT; 11 DINT/REAL; 0 LINT/LREAL; 0 TIME; 1 STRING;  - CRC = 854b3577 >
On Line Change is disabled
```

As shown in example above the Profile editor gives you a better overview, and it is easy to edit the declared variables. It does also support drag and drop from variable list. It 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.

Analogue inputs on the RTU32, RTU32R and RTU32E internal I/O board support both underflow and overflow detection. These are read via the 2 last bits of the word where the AI values are read (the AI value is a 14bit value read from bit 0-13). Bit 14 reports overflow and bit 15 reports underflow.

7.9 Counter inputs

Some I/O boards have support for counter inputs. Auxiliary inputs (ZI) are used for reading 32bit counter inputs. The counter values are stored in registers (modules) and can be reset by using a virtual output (ZO).

Reset of counters is done on an output raising edge. Make sure that the reset output is hold for minimum 2sec in order to ensure that that one I/O scan has been performed. This could be done by using a timer in the WorkSuite application program.

7.10 WorkSuite SOE Driver for RTU32

The SOE (Sequence Of Event) Driver is used for storing digital inputs events in a firmware FIFO with time stamp and status information.

The features of the SOE Driver are:

- Buffering of digital input events in the firmware independent of the PLC runtime application cycle.
- Available for the first 200 digital inputs on a RTU32 node (internal DI and Expansion module DI).
- Support debounce and chattering filter functions.
- Each SOE I/O are buffered with four type of information:
  - The DI value (BOOLEAN)
  - Full time stamp (LINT)
  - Blocked status if chattering filters are active (BOOLEAN)
  - Buffer overflow status (BOOLEAN)

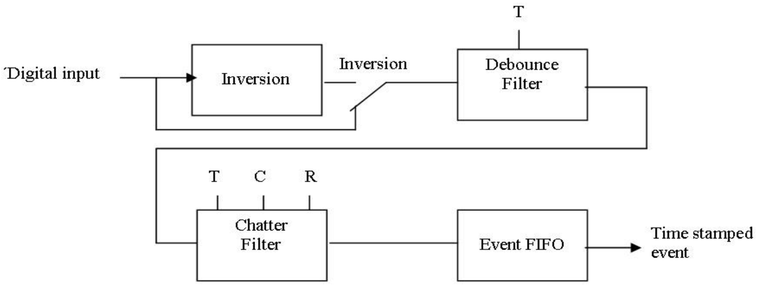
SOE buffered data sets in the FIFO buffer are read from the PLC application and each buffered data sets will only be available in one PLC cycle.

Note: As the values read from the FIFO is ONLY available in one PLC cycle, you have to create a routine in WorkSuite to move the relevant data sets values to another buffer – like an IEC60870 driver transmit buffer/queue.

7.10.1 Filter functions for SOE

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

Filter block diagram



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

- Inversion of input value. Used for inverting the input value transmitted to the next level.

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

7.10.2 Debounce filter function for SOE

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

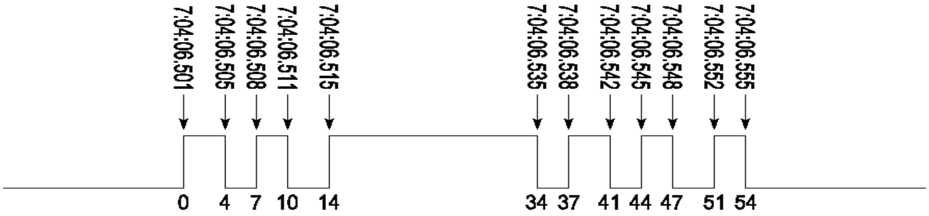


Figure show example of how SOE events are handled by the filter at T=10ms:

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

7.10.3 Chatter filter function for SOE

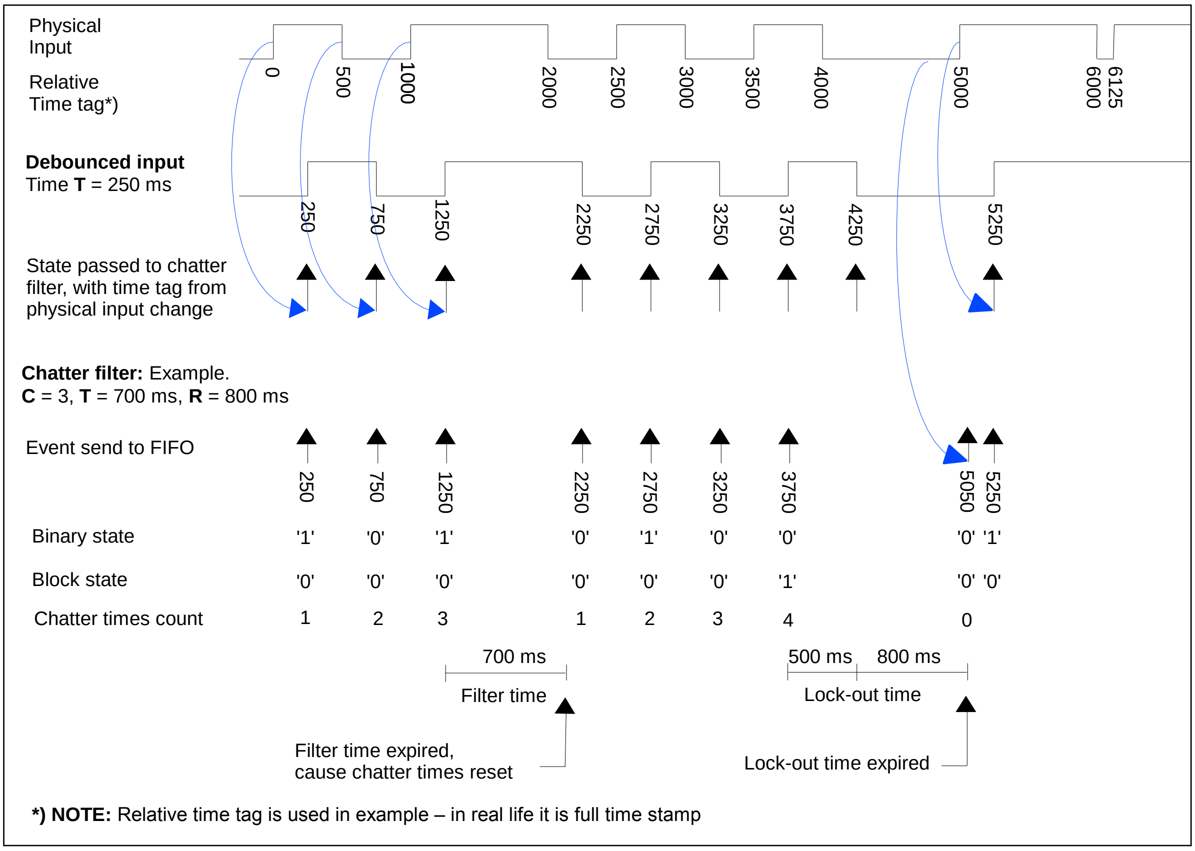
An adjustable chatter filter is provided for each digital input.
A digital input is disabled if the number of state changes, encountered during a defined time interval, is excessively high. While the chatter filter is ON, all state transitions are ignored. While it is OFF, state transitions are gated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

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

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

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

Chatter function diagram:



Detailed diagram with comments to be found in appendix 4. The example of the chatter filter functions includes also the debounce filter as they are interconnected and will often both be used on applications.

7.10.4 RTU SOE Profiles I/O Driver

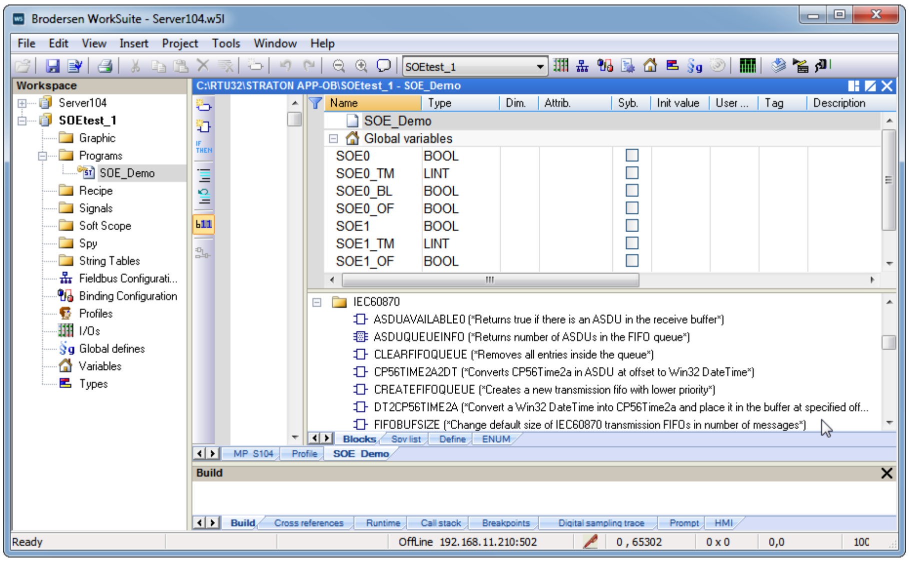
The SOE Profile I/O driver is used for declaring SOE variables directly in the variable list or by using the Profile Editor. As defined earlier a SOE will include a set of up to four variables:

- The actual DI event value – Boolean value
- Full time stamp – LINT value (Win32 DateTime format - use /IEC60870/DT2CP56Time2a to convert it)
- Blocked status – Boolean value (0= not blocked, 1=blocked – Chatter filter is in action)
- Buffer overflow – Boolean value indicating if event buffer is full.

7.10.5 Declaring SOE Profile I/Os in variable list (option 1)

In the Variable list you add four new variables under the Global variable structure. It is recommended to use Global Variable structure to have access to the SOE from any program in your STRATON project.

Create four new variables per data set – e.g. SOE0, SOE0_TM, SOE0_BL, SOE_ST.

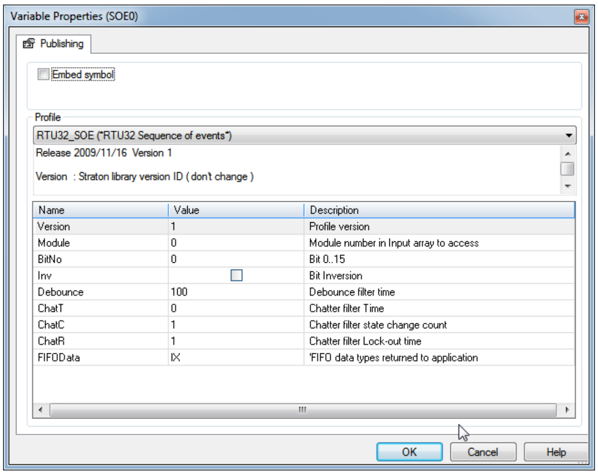

Brodersen WorkSuite - Server104.w5l
File Edit View Insert Project Tools Window Help
SOEtest_1
Workspace
Server104
SOEtest_1
Graphic
Programs
SOE_Demo
Recipe
Signals
Soft Scope
Spy
String Tables
Fieldbus Configurati...
Binding Configuration
Profiles
I/Os
Global defines
Variables
Types
C:\RTU32\STRATON APP-OB\SOEtest_1 - SOE_Demo
Blocks Spy list Define ENUM
MP_S104 Profile SOE_Demo
Build
Build Cross references Runtime Call stack Breakpoints Digital sampling trace Prompt HMI
Ready
OffLine 192.168.11.210:502
0 , 65302
0 x 0
0,0
100

Name	Type	Dim.	Attrib.	Syb.	Init value	User ...	Tag	Description
SOE_Demo								
Global variables								
SOE0	BOOL			□				
SOE0_TM	LINT			□				
SOE0_BL	BOOL			□				
SOE0_OF	BOOL			□				
SOE1	BOOL			□				
SOE1_TM	LINT			□				
SOE1_OF	BOOL			□				

IEC60870
ASDUAVAILABLE0 (*Returns true if there is an ASDU in the receive buffer*)
ASDUQUEUEINFO (*Returns number of ASDUs in the FIFO queue*)
CLEARFIFOQUEUE (*Removes all entries inside the queue*)
CP56TIME2A2DT (*Converts CP56Time2a in ASDU at offset to Win32 DateTime*)
CREATEFIFOQUEUE (*Creates a new transmission fifo with lower priority*)
DT2CP56TIME2A (*Convert a Win32 DateTime into CP56Time2a and place it in the buffer at specified off...
FIFOBUFSIZE (*Change default size of IEC60870 transmission FIFOs in number of messages*)

For each of the four variables in the data set you will need to configure the profile properties.

Right click on the SOE0 variable (this example variable for a data set value variable) and select Properties.



□ Embed symbol

Profile

RTU32_SOE (*RTU32 Sequence of events*)

Release 2009/11/16 Version 1

Version : Straton library version ID (don't change)

Name	Value	Description
Version	1	Profile version
Module	0	Module number in Input array to access
BitNo	0	Bit 0..15
Inv	□	Bit Inversion
Debounce	100	Debounce filter time
ChatT	0	Chatter filter Time
ChatC	1	Chatter filter state change count
ChatR	1	Chatter filter Lock-out time
FIFOData	IX	'FIFO data types returned to application

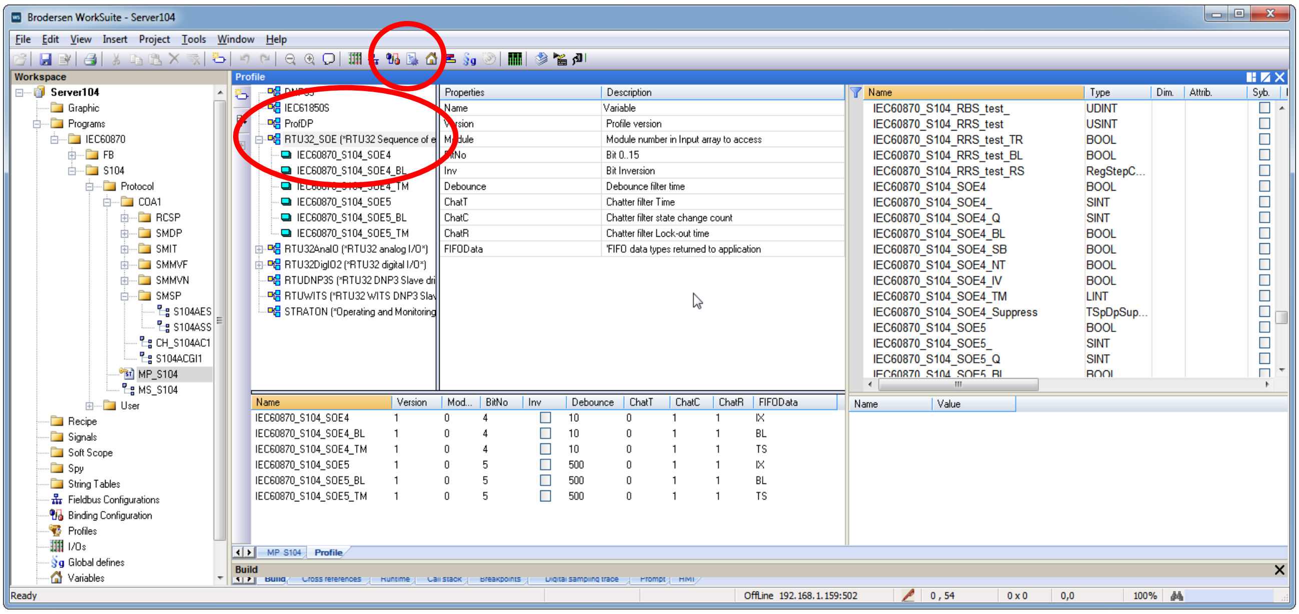
Select the “RTU32SOE” profile, set Module = 0 (first digital input module [word]).
Set BitNo = 0 (first digital input)
Check Inv if you want to invert the bit value

Set Debounce filter time in milliseconds
Set chatter filter time in units of 100millisec
Set chatter filter change counts (must be >=1)
Set chatter filter lock out time (must be >=1)
Select FIFOData type IX for the value of the data set (other settings are TS=timestamp, BL=blocked value, OR=status FIFO overrun)

Repeat this procedure for all the variables in the data set and you have configured one SOE data set of variables. If required, you may configure only some of the data from the data set. Note that you have to configure ALL filters setting equal for each variable in a data set.

7.10.6 Declaring profile I/Os using the Profile Editor (Option 2)

Another way to declare an I/O Profile for RTU32 SOE is to use the Profile Editor in Brodersen WorkSuite. The Profile Editor is a more user friendly way to setup several drivers including the Brodersen I/O LocalBus driver.



Properties	Description
Name	Variable
Version	Profile version
Module	Module number in Input array to access
BitNo	Bit 0..15
Inv	Bit Inversion
Debounce	Debounce filter time
ChatT	Chatter filter Time
ChatC	Chatter filter state change count
ChatR	Chatter filter Lock-out time
FIFOData	'FIFO data types returned to application

Name	Type	Dim.	Attrib.	Syb.
IEC60870_S104_RBS_test_	UDINT			□
IEC60870_S104_RRS_test	USINT			□
IEC60870_S104_RRS_test_TR	BOOL			□
IEC60870_S104_RRS_test_BL	BOOL			□
IEC60870_S104_RRS_test_RS	RegStepC...			□
IEC60870_S104_SOE4	BOOL			□
IEC60870_S104_SOE4_	SINT			□
IEC60870_S104_SOE4_Q	SINT			□
IEC60870_S104_SOE4_BL	BOOL			□
IEC60870_S104_SOE4_SB	BOOL			□
IEC60870_S104_SOE4_NT	BOOL			□
IEC60870_S104_SOE4_IV	BOOL			□
IEC60870_S104_SOE4_TM	LINT			□
IEC60870_S104_SOE4_Suppress	TSpDpSup...			□
IEC60870_S104_SOE5	BOOL			□
IEC60870_S104_SOE5_	SINT			□
IEC60870_S104_SOE5_Q	SINT			□
IEC60870_S104_SOE5_BL	BOOL			□

Name	Version	Mod...	BitNo	Inv	Debounce	ChatT	ChatC	ChatR	FIFOData
IEC60870_S104_SOE4	1	0	4	□	10	0	1	1	IX
IEC60870_S104_SOE4_BL	1	0	4	□	10	0	1	1	BL
IEC60870_S104_SOE4_TM	1	0	4	□	10	0	1	1	TS
IEC60870_S104_SOE5	1	0	5	□	500	0	1	1	IX
IEC60870_S104_SOE5_BL	1	0	5	□	500	0	1	1	BL
IEC60870_S104_SOE5_TM	1	0	5	□	500	0	1	1	TS

Name	Value

As shown in example above the Profile editor gives you a better overview, and it is easy to edit the declared variables. It does also support drag and drop from variable list. See STRATON Help for more details.

7.11 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 RTU32.
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...8
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 is 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 RTU32 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 become active when transmitting data. The RTS Leading setting defines the delay from activating the RTS to the first character is transmitted. The RTS Trailing setting defines the delay from the last character is transmitted to RTS is deactivated.

Important Note! If you are using RS485 on COM2 on RTU32 (DIN rail mounted RTU type), RTS must be added to your settings string. If not added nothing will control send and receive data on RS485.

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 is 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 normal RTS / CTS function). After activating RTS the RTU wait up to 10 sec for the CTS signal. If timeout occur, transmission is discarded. The RTS Trailing setting defines the delay from the last character is transmitted to RTS is 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 primary for debug purpose.
Examples: MCOM2:9600,N,8,1,LL
COM2:57600,E,8,1,RT:30:10,LL

Note! If you just connect a serial device for e.g. ModbusRTU communication via Null Modem cable to the RS232 COM port, you will normally require to add any special extensions.
Note! 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. And if the SP5 is enabled the COM settings overrules any later settings in WorkSuite.

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

7.12 Modbus Drivers

The RTU32 PLC runtime support 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. In addition you will in WorkSuite find support for ModbusUDP, and Function Blocked handled ModbusRTU Slave.

7.12.1 ModbusRTU Master

The RTU32 support 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 used the Fieldbus Configurator. See WorkSuite help for details.

7.12.2 ModbusRTU Slave

The RTU32 support several Slave/Server drivers. You have to add a function block for each Modbus Slave driver. You 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.

7.13 IEC60870-5-101/103/104 Drivers

Utility protocol according to IEC60870-5-101, 103 and -104 are implemented as a basic link driver in the RTU32. 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 includes 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 includes 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, provide 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.

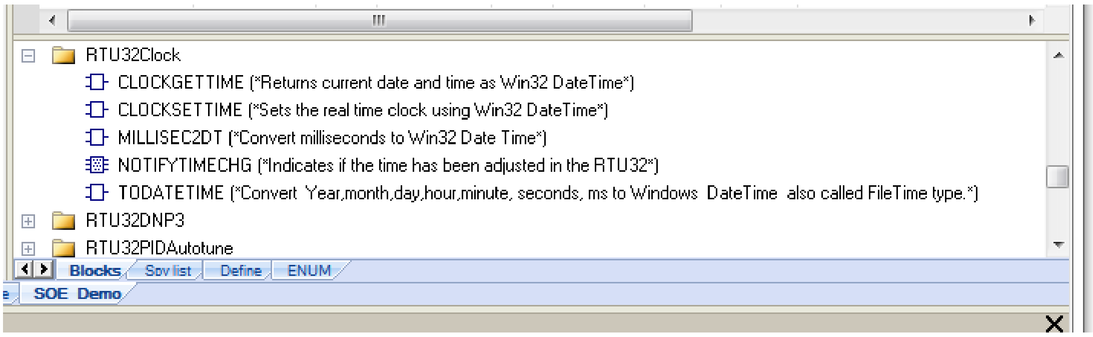
7.14 Watchdog

The RTU32 has a built-in watchdog function on several levels.

In newer hardware version you should note that the watchdog function will power cycle the RTU32 every 180 sec if no Flash card or a Flash card with non-compliant software is installed. Also RTUs with no PLC application running will be detected by the watchdog function.

7.15 Real time / Real time Clock

The RTU32 has a hardware real time clock and is controlled in the operating system. It provide in RTU32 support for millisecond resolution in order to make accurate time stamping. From the STRATON application program you have access to the real time clock via some special functions as shown below. You can both read and set the clock.



RTU32Clock
CLOCKGETTIME (*Returns current date and time as Win32 DateTime*)
CLOCKSETTIME (*Sets the real time clock using Win32 DateTime*)
MILLISEC2DT (*Convert milliseconds to Win32 Date Time*)
NOTIFYTIMECHG (*Indicates if the time has been adjusted in the RTU32*)
TODATETIME (*Convert Year,month,day,hour,minute, seconds, ms to Windows DateTime also called FileTime type.*)
RTU32DNP3
RTU32PIDAutotune

On the configuration webpages you can also define how to synchronise the clock. For now options for manual setup and SNTP Server synchronisation is possible.

7.16 Multiple serial drivers on same COM port

RTU32 provide extended functions for working with multiple serial protocol drivers on the same COM port. A switch function is implemented for selecting relevant protocols. This is the same switch function that is used when enabling the SP5 (Secure Remote Protocol).

The multiple driver function provides you with possibility to work with both direct serial connections and Modem dial connections. Even Master and Slave drivers are possible. The control and configuration when using multiple protocols has to be setup in the RTU application. Please contact Brodersen to get a detailed specification of the possibilities and limitations.

7.17 Modem and VPN Functions

WorkSuite provide a range of modem and VPN control Functions. The DialUp library includes:

- Functions to control and monitor UCM-94 3G/GPRS modem.

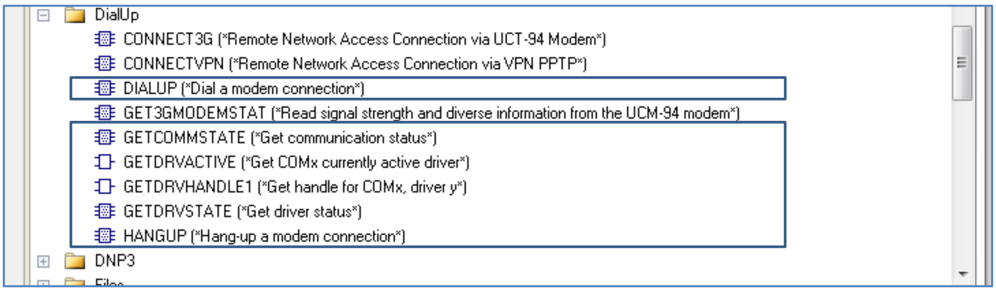
- Functions to control and monitor VPN Client

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

7.17.1 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 modem connection state. A number of WorkSuite functions are available for this:



DialUp
- CONNECT3G (*Remote Network Access Connection via UCT-94 Modem*)
- CONNECTVPN (*Remote Network Access Connection via VPN PPTP*)
- DIALUP (*Dial a modem connection*)
- GET3GMODEMSTAT (*Read signal strength and diverse information from the UCM-94 modem*)
- GETCOMMSTATE (*Get communication status*)
- GETDRVACTIVE (*Get COMx currently active driver*)
- GETDRVHANDLE1 (*Get handle for COMx, driver y*)
- GETDRVSTATE (*Get driver status*)
- HANGUP (*Hang-up a modem connection*)

DNP3

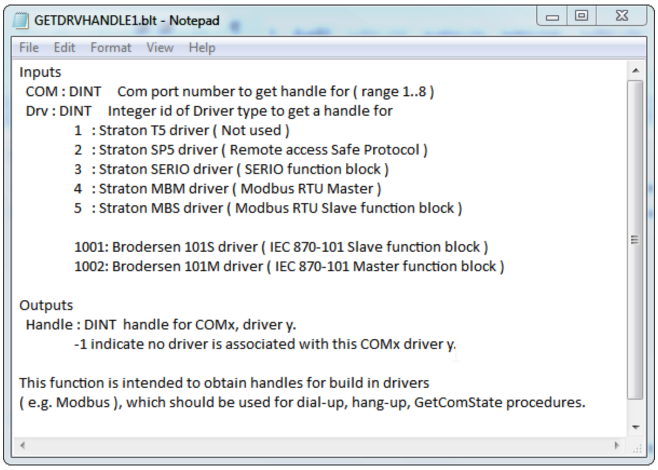
Files

The modem function is enabled by adding M to the COM settings. See details in COM settings section.

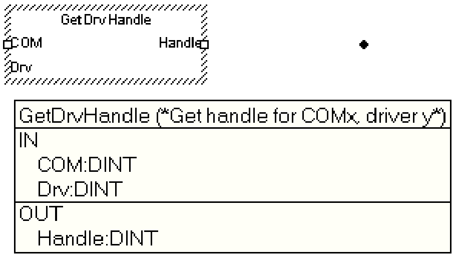
After enabling the modem dial driver, you must add a Modem driver handle in your application program (GetDrvHandle). Your dialup function must now refer to this handle. The DialUp function includes all the necessary parameters to establish and control a dial-up or dial-in session.

GetDrvHandle (Get driver handle).

This function is used to get handle of build-in protocol drivers, like ModbusRTU master or slave. This should be used when operating a build in protocol driver (e.g. dialing). To obtain the handle, COM port number and driver number must be provided.

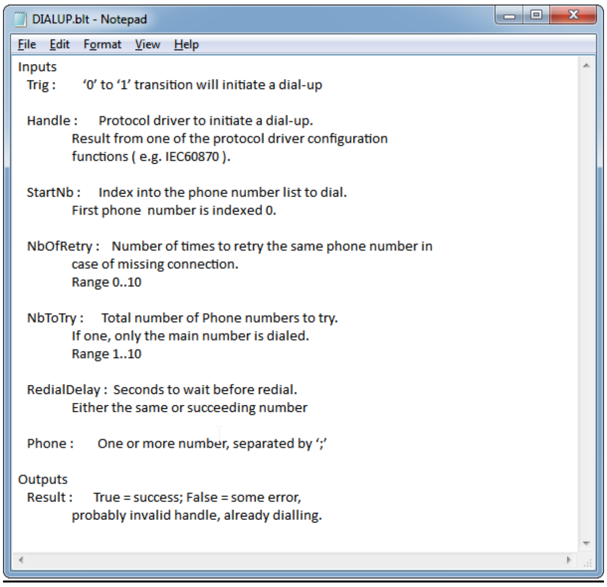


```
Inputs
 COM : DINT    Com port number to get handle for ( range 1..8 )
 Drv : DINT    Integer id of Driver type to get a handle for
         1  : Straton T5 driver ( Not used )
         2  : Straton SP5 driver ( Remote access Safe Protocol )
         3  : Straton SERIO driver ( SERIO function block )
         4  : Straton MBM driver ( Modbus RTU Master )
         5  : Straton MBS driver ( Modbus RTU Slave function block )

         1001: Brodersen 101S driver ( IEC 870-101 Slave function block )
         1002: Brodersen 101M driver ( IEC 870-101 Master function block )

Outputs
 Handle : DINT  handle for COMx, driver y.
         -1 indicate no driver is associated with this COMx driver y.

This function is intended to obtain handles for build in drivers
( e.g. Modbus ), which should be used for dial-up, hang-up, GetComState procedures.
```

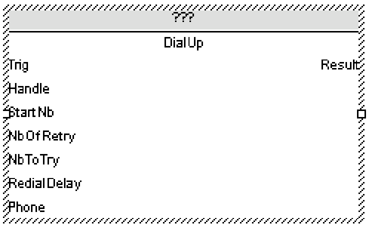


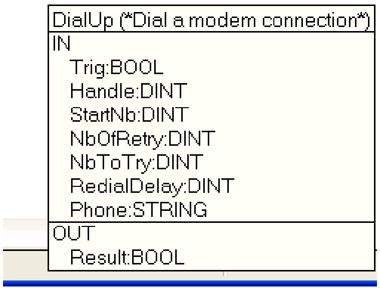
```
GetDrvHandle (*Get handle for COMx, driver y*)
IN
  COM:DINT
  Drv:DINT
OUT
  Handle:DINT
```

Dialup - Start a dial-up connection for one of the protocol drivers.



```
Inputs
  Trig :     '0' to '1' transition will initiate a dial-up

  Handle :    Protocol driver to initiate a dial-up.
          Result from one of the protocol driver configuration
          functions ( e.g. IEC60870 ).

  StartNb :   Index into the phone number list to dial.
          First phone  number is indexed 0.

  NbOfRetry :   Number of times to retry the same phone number in
          case of missing connection.
          Range 0..10

  NbToTry :    Total number of Phone numbers to try.
          If one, only the main number is dialed.
          Range 1..10

  RedialDelay : Seconds to wait before redial.
          Either the same or succeeding number

  Phone :     One or more number, separated by ';'

Outputs
  Result :     True = success; False = some error,
          probably invalid handle, already dialling.
```

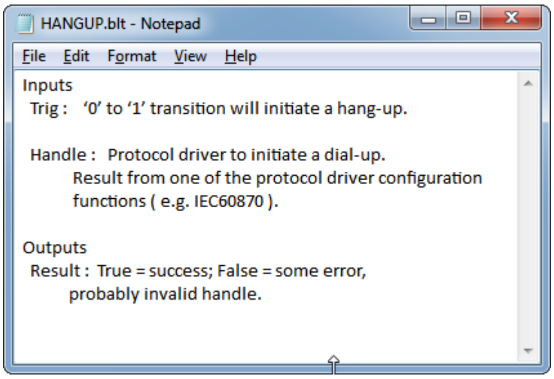




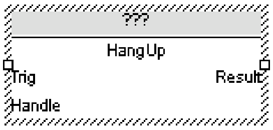
```
DialUp (*Dial a modem connection*)
IN
  Trig:BOOL
  Handle:DINT
  StartNb:DINT
  NbOfRetry:DINT
  NbToTry:DINT
  RedialDelay:DINT
  Phone:STRING
OUT
  Result:BOOL
```

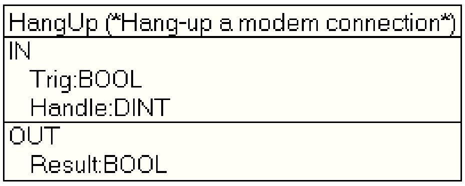
NOTE: There should be declared one instance of this FB for each modem driver (physical modem) used.

HangUp - Hang-up current connection



```
Inputs
  Trig :  '0' to '1' transition will initiate a hang-up.

  Handle :  Protocol driver to initiate a dial-up.
        Result from one of the protocol driver configuration
        functions ( e.g. IEC60870 ).

Outputs
  Result :  True = success; False = some error,
        probably invalid handle.
```



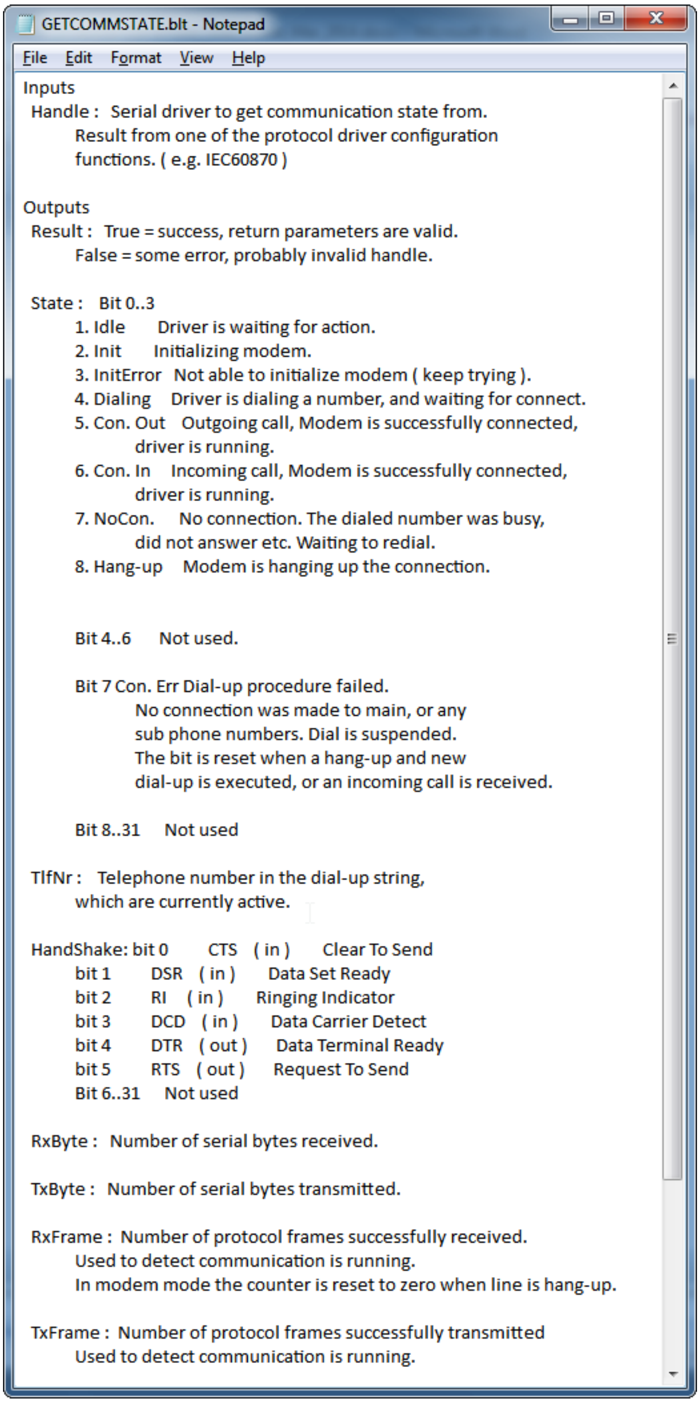


```
HangUp (*Hang-up a modem connection*)
IN
  Trig:BOOL
  Handle:DINT
OUT
  Result:BOOL
```

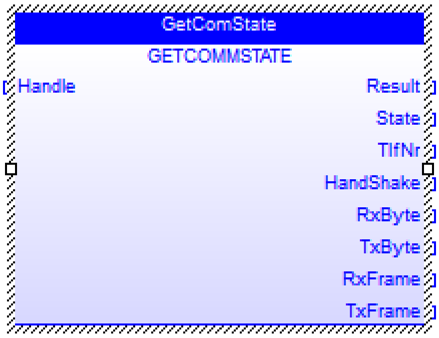
NOTE: There should be declared one instance of this FB for each modem driver (physical modem) used.

GetCommState - Get communication status.

Called regular by RTU application to get current status of the driver, and then determine which action should be taken. This function is primary intended for monitor data transmission on modem driver, but could be used by null modem drivers as well, to monitor data transmission.



```
Inputs
 Handle :  Serial driver to get communication state from.
        Result from one of the protocol driver configuration
        functions. ( e.g. IEC60870 )

Outputs
 Result :  True = success, return parameters are valid.
        False = some error, probably invalid handle.

 State :   Bit 0..3
        1. Idle       Driver is waiting for action.
        2. Init       Initializing modem.
        3. InitError  Not able to initialize modem ( keep trying ).
        4. Dialing    Driver is dialing a number, and waiting for connect.
        5. Con. Out   Outgoing call, Modem is successfully connected,
                driver is running.
        6. Con. In    Incoming call, Modem is successfully connected,
                driver is running.
        7. NoCon.     No connection. The dialed number was busy,
                did not answer etc. Waiting to redial.
        8. Hang-up    Modem is hanging up the connection.


        Bit 4..6      Not used.

        Bit 7 Con. Err Dial-up procedure failed.
                No connection was made to main, or any
                sub phone numbers. Dial is suspended.
                The bit is reset when a hang-up and new
                dial-up is executed, or an incoming call is received.

        Bit 8..31     Not used

 TlfNr :   Telephone number in the dial-up string,
        which are currently active.

 HandShake: bit 0      CTS   ( in )      Clear To Send
        bit 1       DSR   ( in )      Data Set Ready
        bit 2       RI    ( in )      Ringing Indicator
        bit 3       DCD   ( in )      Data Carrier Detect
        bit 4       DTR   ( out )     Data Terminal Ready
        bit 5       RTS   ( out )     Request To Send
        Bit 6..31     Not used

 RxByte :  Number of serial bytes received.

 TxByte :  Number of serial bytes transmitted.

 RxFrame :  Number of protocol frames successfully received.
        Used to detect communication is running.
        In modem mode the counter is reset to zero when line is hang-up.

 TxFrame :  Number of protocol frames successfully transmitted
        Used to detect communication is running.
```

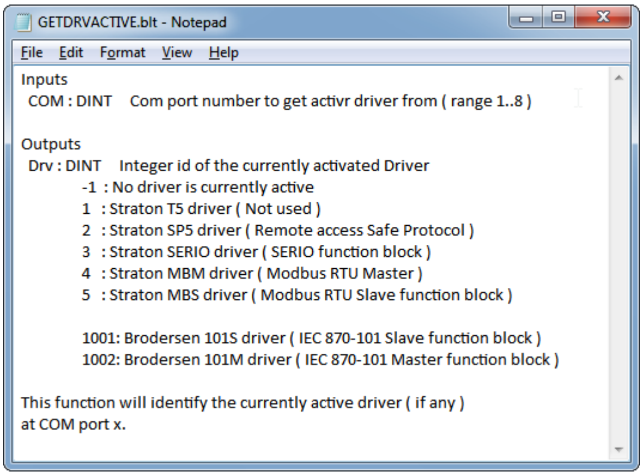


Note: This function is primary intended for modem driver, but could be used by null modem drivers as well, to monitor data transmission. Data returned when using multiple protocol drivers at the same COM port, is more port related than protocol driver related. Obtaining e.g. RxFrame value using handle from either passive or active driver at the same COM port will return the same value.

How to use the modem dial functions may be better understood if you get one of our available WorkSuite examples. Check your RTU32 CD or our homepage for useful examples.

GetDrvActive - Get active driver

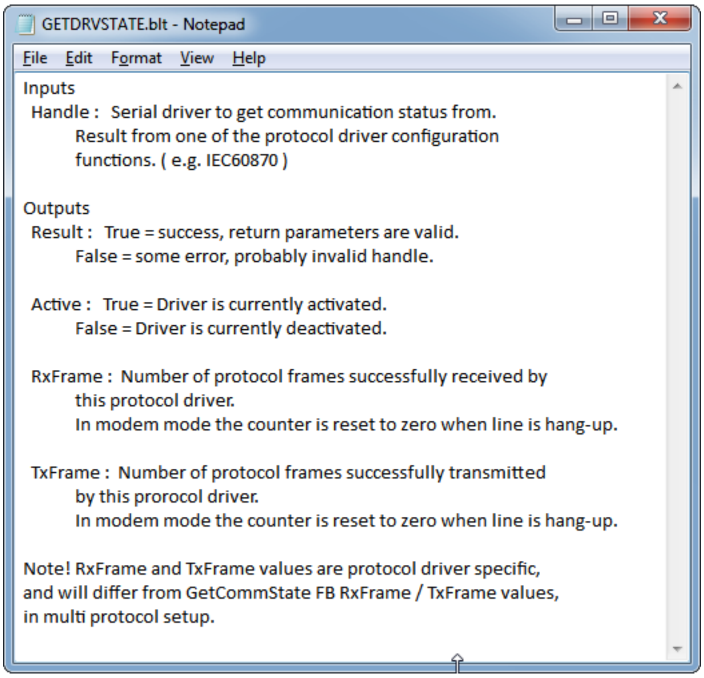
Will return ID of the currently active driver (if any) on a COM port. Used by RTU application to identify which protocol is actually active in a multiprotocol setup.



```
Inputs
 COM : DINT    Com port number to get activr driver from ( range 1..8 )

Outputs
 Drv : DINT    Integer id of the currently activated Driver
        -1  : No driver is currently active
        1   : Straton T5 driver ( Not used )
        2   : Straton SP5 driver ( Remote access Safe Protocol )
        3   : Straton SERIO driver ( SERIO function block )
        4   : Straton MBM driver ( Modbus RTU Master )
        5   : Straton MBS driver ( Modbus RTU Slave function block )

        1001: Brodersen 101S driver ( IEC 870-101 Slave function block )
        1002: Brodersen 101M driver ( IEC 870-101 Master function block )

This function will identify the currently active driver ( if any )
at COM port x.
```

GetDrvState - Get driver status

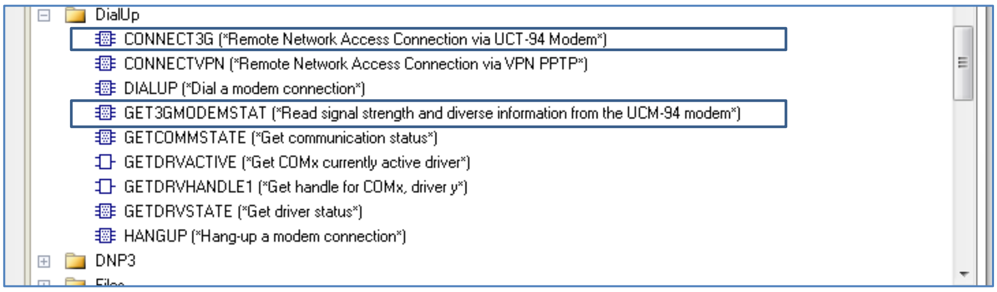
Called regular by RTU application to get status of a driver.



```
Inputs
 Handle :  Serial driver to get communication status from.
        Result from one of the protocol driver configuration
        functions. ( e.g. IEC60870 )

Outputs
 Result :  True = success, return parameters are valid.
        False = some error, probably invalid handle.

 Active :  True = Driver is currently activated.
        False = Driver is currently deactivated.

 RxFrame :  Number of protocol frames successfully received by
        this protocol driver.
        In modem mode the counter is reset to zero when line is hang-up.

 TxFrame :  Number of protocol frames successfully transmitted
        by this prorocol driver.
        In modem mode the counter is reset to zero when line is hang-up.

Note! RxFrame and TxFrame values are protocol driver specific,
and will differ from GetCommState FB RxFrame / TxFrame values,
in multi protocol setup.
```

7.17.2 Functions for control and monitor UCM-94 3G/GPRS Modem

Control and monitoring functions are available for controlling and monitoring the UCM-94 family modems used for 3G/GPRS communication.



```
DialUp
    CONNECT3G (*Remote Network Access Connection via UCT-94 Modem*)
    CONNECTVPN (*Remote Network Access Connection via VPN PPTP*)
    DIALUP (*Dial a modem connection*)
    GET3GMODEMSTAT (*Read signal strength and diverse information from the UCM-94 modem*)
    GETCOMMSTATE (*Get communication status*)
    GETDRVACTIVE (*Get COMx currently active driver*)
    GETDRVHANDLE1 (*Get handle for COMx, driver y*)
    GETDRVSTATE (*Get driver status*)
    HANGUP (*Hang-up a modem connection*)
DNP3
```

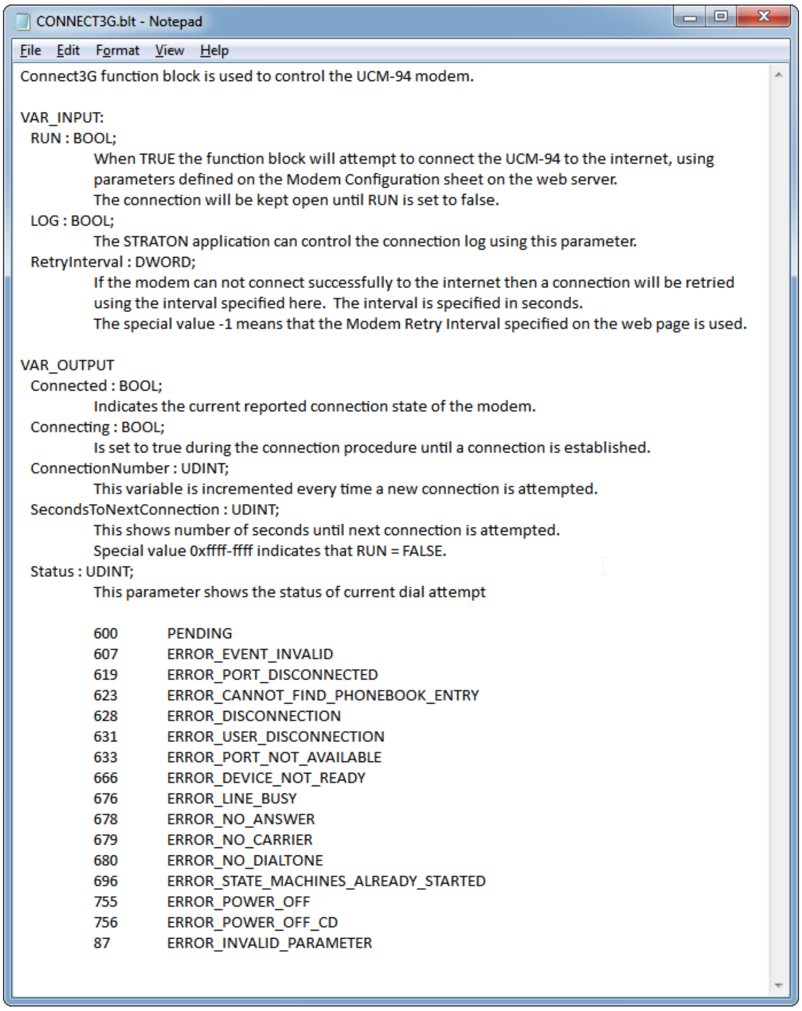
The UCM-94 modem connect procedure can be done automatically at start-up or controlled in the PLC program using Function listed shown below. Automatic or PLC controlled connect is defined on the modem configuration webpage. If automatic connect is used, the monitoring Function is still applicable to use for monitoring the connection status.

The low level modem handling (send AT commands, connect detection etc.) is done by firmware.

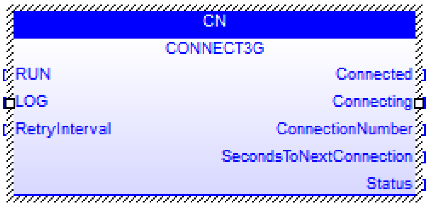
Please find here listed the available Functions.

Connect3G Function for PLC control

The Connect3G function is used when the PLC application control the Modem connection. If automatic connection is selected this Function is not used.

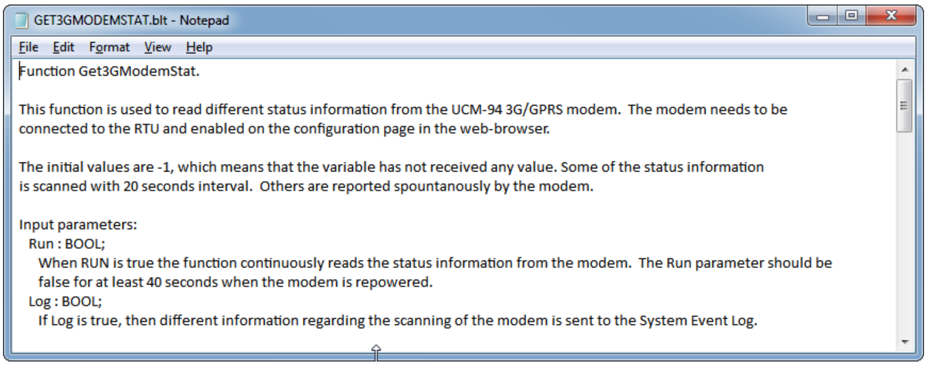


```
Connect3G function block is used to control the UCM-94 modem.

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 STRATON 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
```

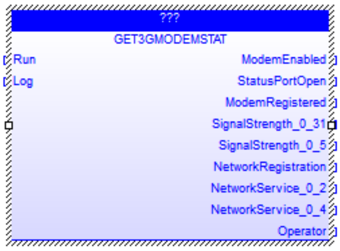


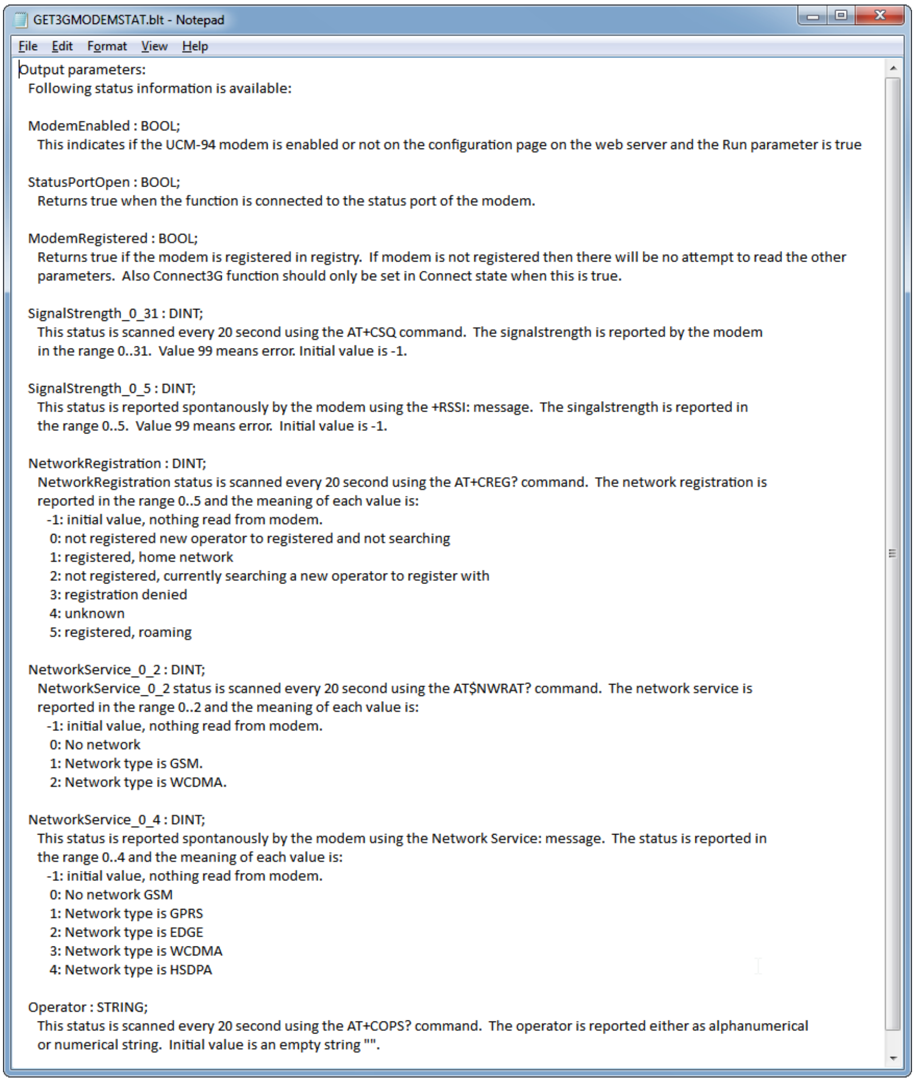
Get3GModemState Function

The Get3GModemState function can always be used for monitoring the actual state of the UCM-94 modem and the 3G/GPRS connection status.



```
Function Get3GModemStat.

This function is used to read different status information from the UCM-94 3G/GPRS modem.  The modem needs to be
connected to the RTU and enabled on the configuration page in the web-browser.

The initial values are -1, which means that the variable has not received any value. Some of the status information
is scanned with 20 seconds interval.  Others are reported spountanously by the modem.

Input parameters:
  Run : BOOL;
    When RUN is true the function continuously reads the status information from the modem.  The Run parameter should be
    false for at least 40 seconds when the modem is repowered.
  Log : BOOL;
    If Log is true, then different information regarding the scanning of the modem is sent to the System Event Log.
```





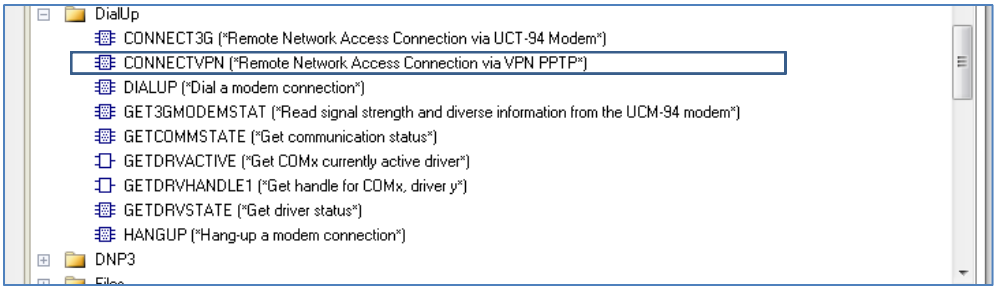
```
Output parameters:
  Following status information is available:

  ModemEnabled : BOOL;
    This indicates if the UCM-94 modem is enabled or not on the configuration page on the web server and the Run parameter is true

  StatusPortOpen : BOOL;
    Returns true when the function is connected to the status port of the modem.

  ModemRegistered : BOOL;
    Returns true if the modem is registered in registry.  If modem is not registered then there will be no attempt to read the other
    parameters.  Also Connect3G function should only be set in Connect state when this is true.

  SignalStrength_0_31 : DINT;
    This status is scanned every 20 second using the AT+CSQ command.  The signalstrength is reported by the modem
    in the range 0..31.  Value 99 means error. Initial value is -1.

  SignalStrength_0_5 : DINT;
    This status is reported spontanously by the modem using the +RSSI: message.  The singalstrength is reported in
    the range 0..5.  Value 99 means error.  Initial value is -1.

  NetworkRegistration : DINT;
    NetworkRegistration status is scanned every 20 second using the AT+CREG? command.  The network registration is
    reported in the range 0..5 and the meaning of each value is:
      -1: initial value, nothing read from modem.
       0: not registered new operator to registered and not searching
       1: registered, home network
       2: not registered, currently searching a new operator to register with
       3: registration denied
       4: unknown
       5: registered, roaming

  NetworkService_0_2 : DINT;
    NetworkService_0_2 status is scanned every 20 second using the AT$NWRAT? command.  The network service is
    reported in the range 0..2 and the meaning of each value is:
      -1: initial value, nothing read from modem.
       0: No network
       1: Network type is GSM.
       2: Network type is WCDMA.

  NetworkService_0_4 : DINT;
    This status is reported spontanously by the modem using the Network Service: message.  The status is reported in
    the range 0..4 and the meaning of each value is:
      -1: initial value, nothing read from modem.
       0: No network GSM
       1: Network type is GPRS
       2: Network type is EDGE
       3: Network type is WCDMA
       4: Network type is HSDPA

  Operator : STRING;
    This status is scanned every 20 second using the AT+COPS? command.  The operator is reported either as alphanumerical
    or numerical string.  Initial value is an empty string "".
```

The 3Gmodem status function provides status information as listed above. The field strength that indicates radio link level is submitted in two ranges. It is recommended to the SignalStrength_0_31 output for monitoring the field strength. Compared to rssi dBm levels, the reported 0-31 range can be explained like this:

SignalStrength_0_31 value 0 = -113 dBm or less
SignalStrength_0_31 value 1 = -111 dBm
SignalStrength_0_31 value 2...30 = -109... -53 dBm
SignalStrength_0_31 value 31 = -51 dBm or greater
SignalStrength_0_31 value 99 = not known or not detectable

To have a proper radio link connection, the field strength on this scale should not be less than 20. And note that the field strength does not equal QoS on the actual 3G/GPRS network connection. This is dependent on the provider equipment status and load on the provider network.

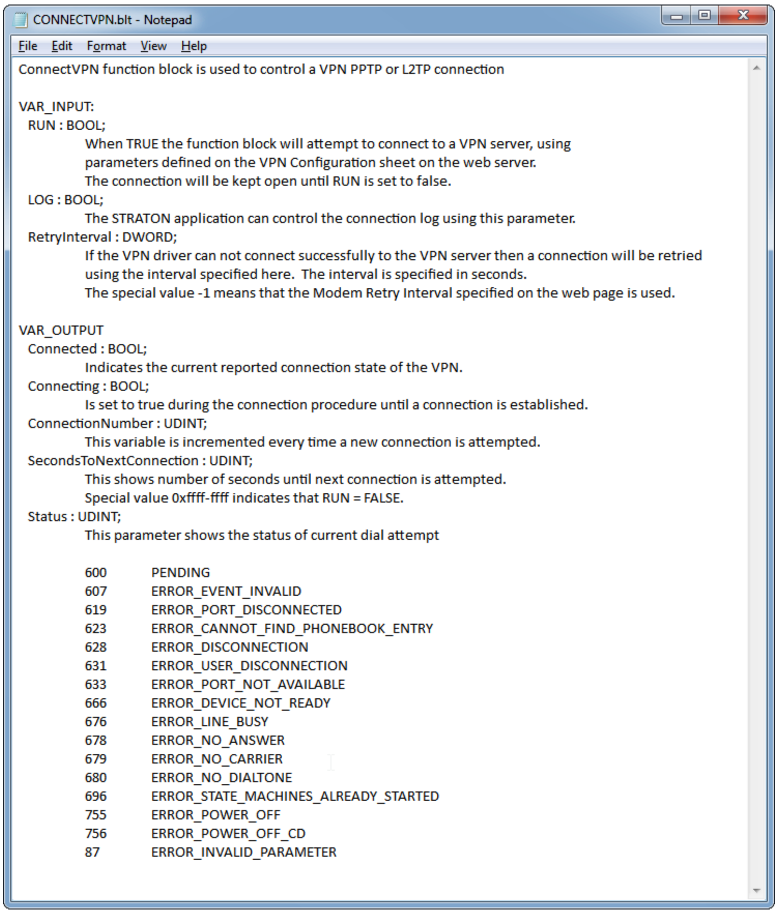
7.17.3 Function for control and monitor VPN Client



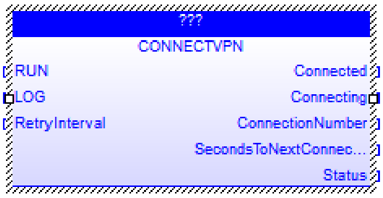
DialUp
CONNECT3G (*Remote Network Access Connection via UCT-94 Modem*)
CONNECTVPN (*Remote Network Access Connection via VPN PPTP*)
DIALUP (*Dial a modem connection*)
GET3GMODEMSTAT (*Read signal strength and diverse information from the UCM-94 modem*)
GETCOMMSTATE (*Get communication status*)
GETDRVACTIVE (*Get COMx currently active driver*)
GETDRVHANDLE1 (*Get handle for COMx, driver y*)
GETDRVSTATE (*Get driver status*)
HANGUP (*Hang-up a modem connection*)
DNP3
Files

The VPN Client is always controlled by the PLC application. The main setting is required on the VPN webpage and the connection control and monitoring is handled by using the ConnectVPN Function. The VPN Client support basic VPN functions and all possible parameters are according to the webpage settings. Note that L2TP VPN required pre-shared key configuration and continuous key updated when connected. It is only recommended to use L2TP VPN Client over stabile network connections as the key exchange processes may be very sensitive to any latency in the communication.

ConnectVPN
Below the features of the ConnectVPN Function is shown,



```
ConnectVPN function block is used to control a VPN PPTP or L2TP connection

VAR_INPUT:
  RUN : BOOL;
        When TRUE the function block will attempt to connect to a VPN server, using
        parameters defined on the VPN Configuration sheet on the web server.
        The connection will be kept open until RUN is set to false.
  LOG : BOOL;
        The STRATON application can control the connection log using this parameter.
  RetryInterval : DWORD;
        If the VPN driver can not connect successfully to the VPN server 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 VPN.
  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
```



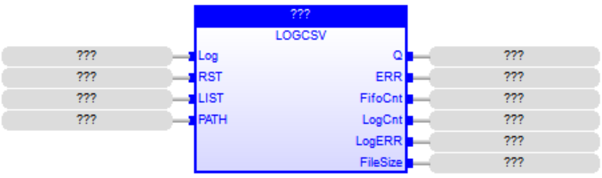
7.18 Data Logging

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

1. LOGCSV Function is a dedicated RTU32 Series logging function. And you can find a special Function in the WorkSuite Demo examples installed together with your WorkSuite Package. The main difference is that this data log function handles the read/write functions in a background scan and do 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 of CSV-file. It is a standard file write function and it is not recommended to use for data logging as standalone Function.

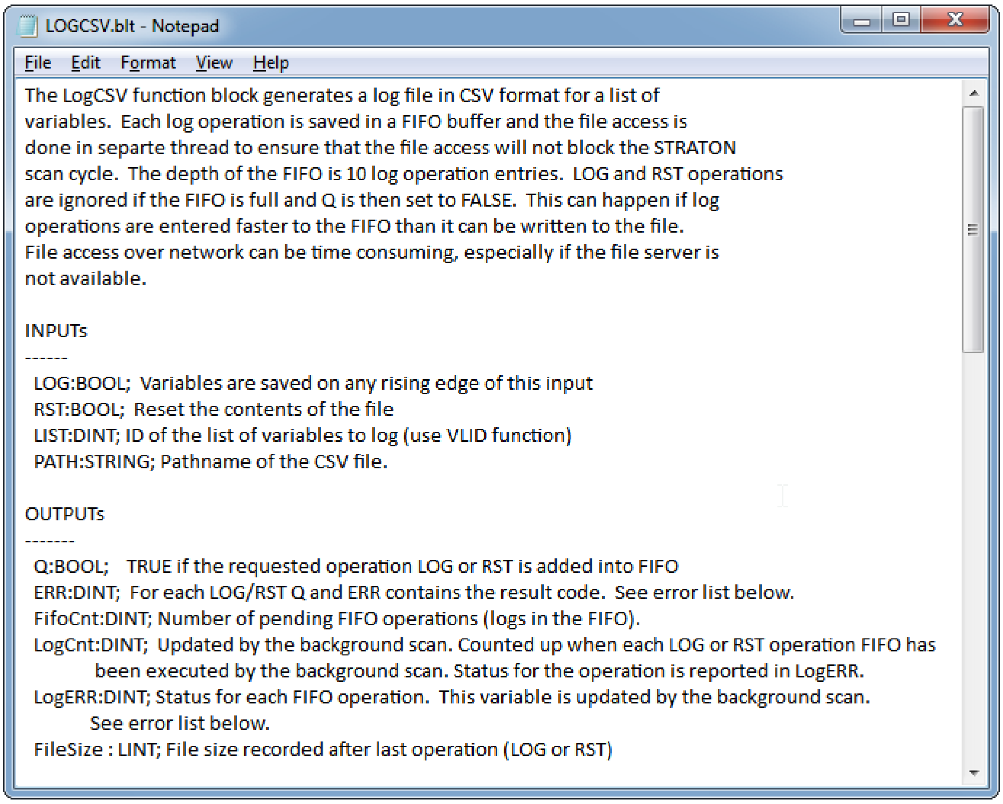
LOGCSV Function

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

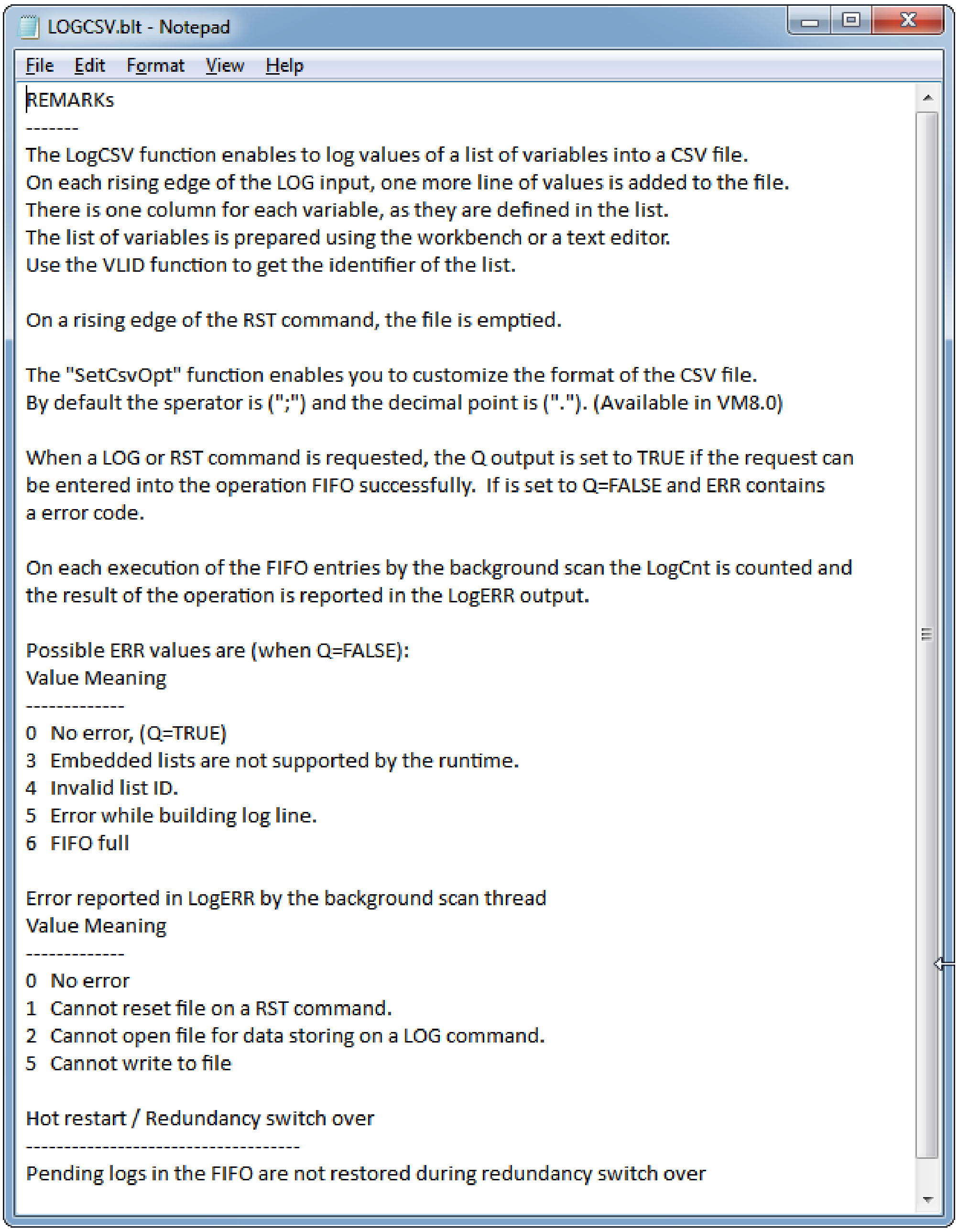


The LogCSV function block generates a log file in CSV format for a list of variables. Each log operation is saved in a FIFO buffer and the file access is done in separate thread to ensure that the file access will not block the PLC runtime cycle. The depth of the FIFO is 10 log operation entries. LOG 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
```

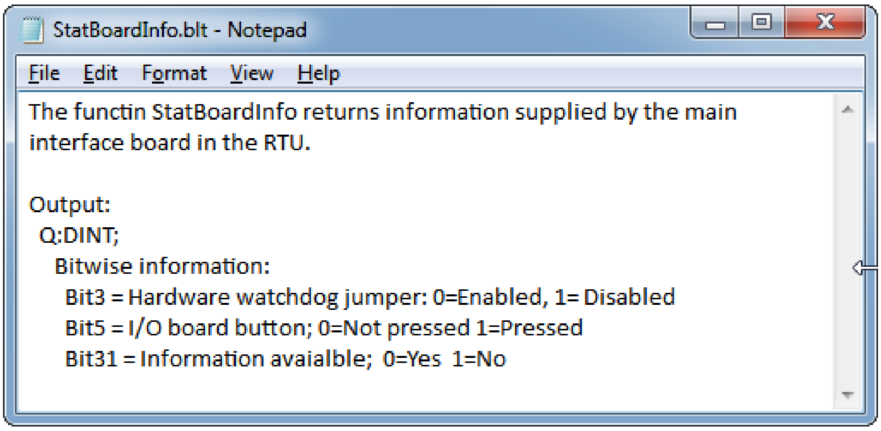
In the RTU32Example projects in WorkSuite you will find a WorkSuite Demo project LOGCSV_Demo with a log UDFB block named TMultiFileLog. With this Function using the LOGCSV function you will be able to control number of files to log and limit file size. And in fact use this as a ring buffer, to ensure that your log file will not fill up the Flash disc.

7.19 Status information functions

RTU32 Series products with firmware version 1.55 and newer have hardware status information available. Some models in the RTU32 Series have full support for the status functions and some modules have only limited support. Under each function is noted any limitation in support of function.

7.19.1 Interface Board Status Info (STATBOARDINFO)

The function StatBoardInfo returns information supplied by the main interface board in the RTU.

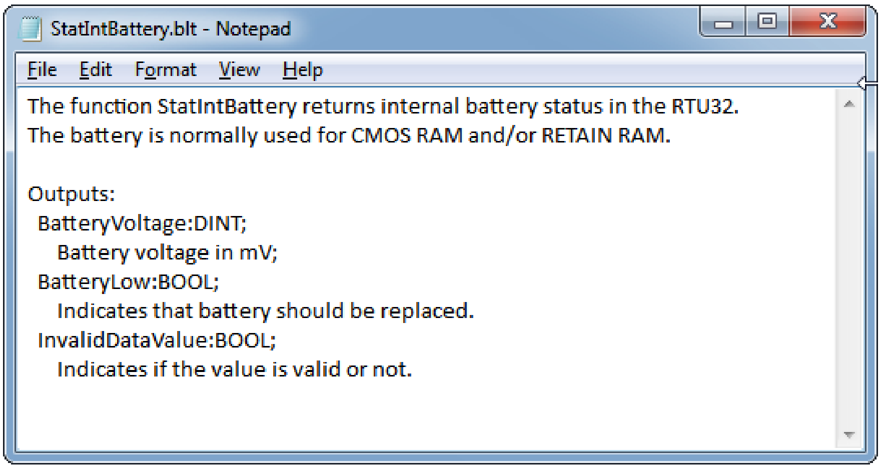


```
The functin StatBoardInfo returns information supplied by the main
interface board in the RTU.

Output:
 Q:DINT;
   Bitwise information:
     Bit3 = Hardware watchdog jumper: 0=Enabled, 1= Disabled
     Bit5 = I/O board button; 0=Not pressed 1=Pressed
     Bit31 = Information avaialble;  0=Yes  1=No
```

7.19.2 Battery Status (STATINTBATTERY)

The function StatIntBattery returns internal battery status in the RTU32.

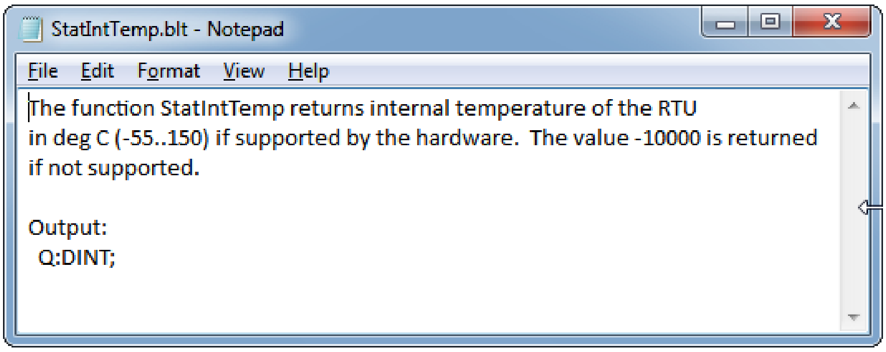
The battery is normally used for CMOS RAM and/or RETAIN RAM.



```
The function StatIntBattery returns internal battery status in the RTU32.
The battery is normally used for CMOS RAM and/or RETAIN RAM.

Outputs:
 BatteryVoltage:DINT;
   Battery voltage in mV;
 BatteryLow:BOOL;
   Indicates that battery should be replaced.
 InvalidDataValue:BOOL;
   Indicates if the value is valid or not.
```

7.19.3 RTU Internal Temperature (STATINTTEMP)

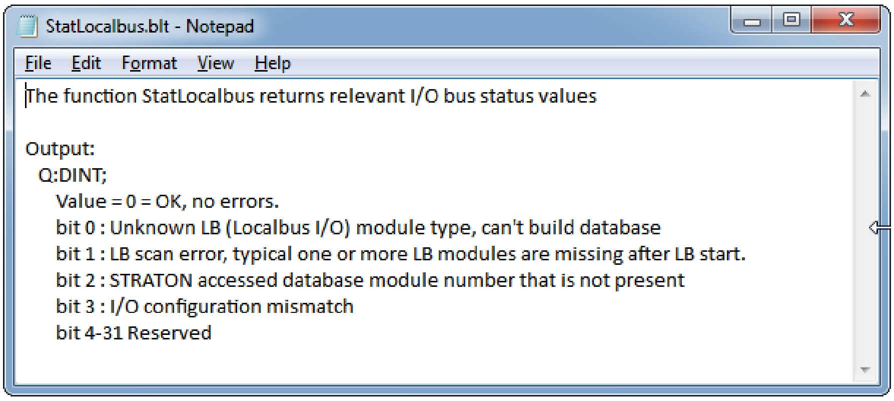
The function StatIntTemp returns internal temperature of the RTU.



```
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.

Output:
 Q:DINT;
```

7.19.4 LocalBus Status (STATLOCALBUS)

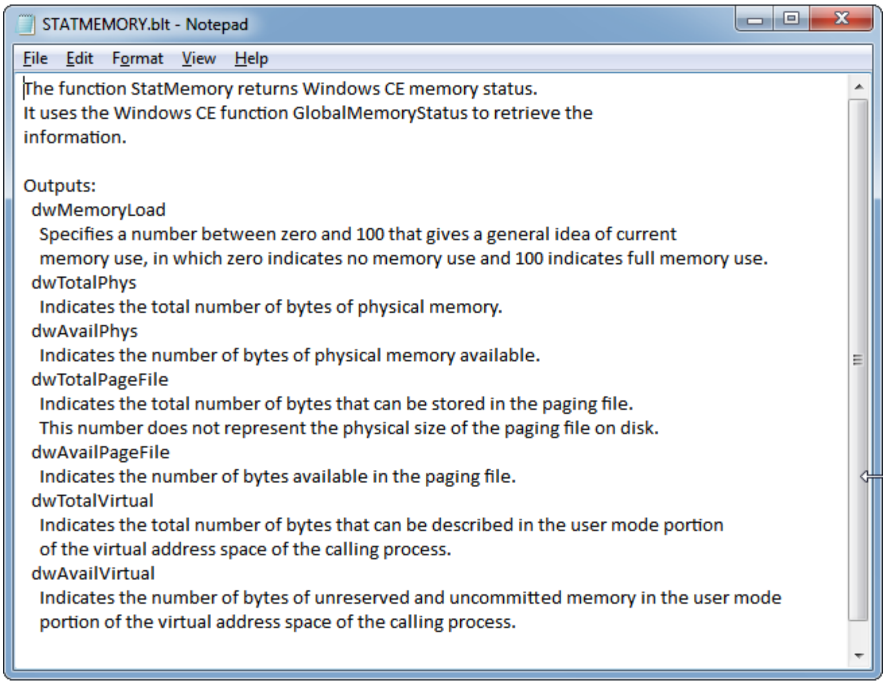
The function StatLocalbus returns relevant I/O bus status values



```
The function StatLocalbus returns relevant I/O bus status values

Output:
  Q:DINT;
    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 that is not present
    bit 3 : I/O configuration mismatch
    bit 4-31 Reserved
```

7.19.5 RTU memory Status (STATMEMORY)

The function StatMemory returns Windows CE memory status.



```
The function StatMemory returns Windows CE memory status.
It uses the Windows CE function GlobalMemoryStatus to retrieve the
information.

Outputs:
 dwMemoryLoad
  Specifies a number between zero and 100 that gives a general idea of current
  memory use, in which zero indicates no memory use and 100 indicates full memory use.
 dwTotalPhys
  Indicates the total number of bytes of physical memory.
 dwAvailPhys
  Indicates the number of bytes of physical memory available.
 dwTotalPageFile
  Indicates the total number of bytes that can be stored in the paging file.
  This number does not represent the physical size of the paging file on disk.
 dwAvailPageFile
  Indicates the number of bytes available in the paging file.
 dwTotalVirtual
  Indicates the total number of bytes that can be described in the user mode portion
  of the virtual address space of the calling process.
 dwAvailVirtual
  Indicates the number of bytes of unreserved and uncommitted memory in the user mode
  portion of the virtual address space of the calling process.
```

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

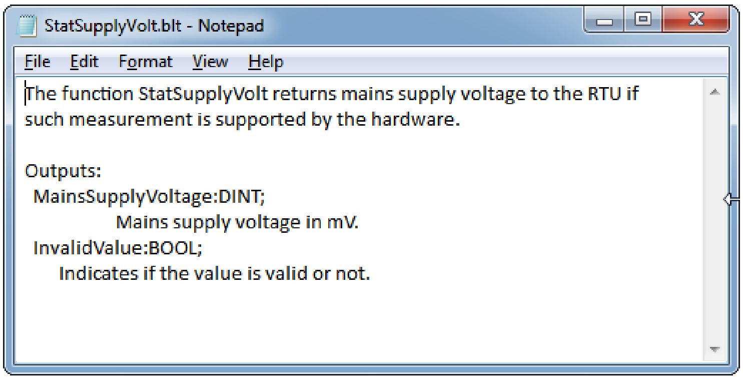
The dwTotalPhys tell you how much RAM program memory that is available in the RTU. The rest of the RAM memory up to around 128MB is used for storage (file system etc.).

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

And finally the dwMemoryLoad tell you in percent the actual RAM program memory load. With a medium PLC runtime application you will have a load on approx. 60-63%. Only applications with multiple IEC61850 connections using large communication models will require larger amount of memory. The maximum load recommended on a RTU is 90%.

7.19.6 Supply Voltage Status (STATSUPPLYVOLT)

The function StatSupplyVolt returns mains supply voltage.

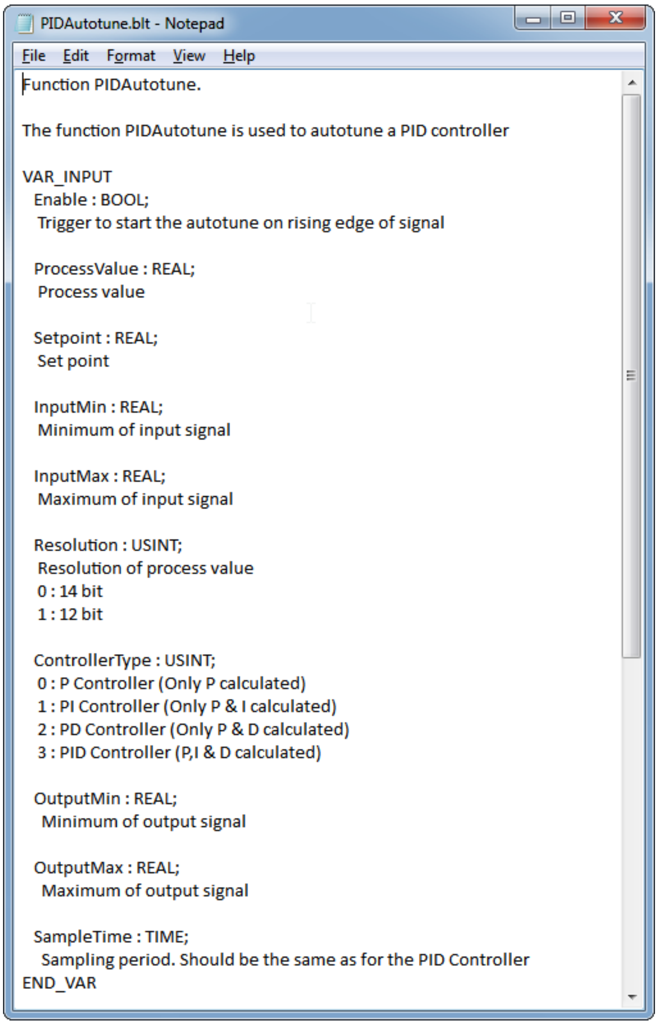


```
The function StatSupplyVolt returns mains supply voltage to the RTU if
such measurement is supported by the hardware.

Outputs:
 MainsSupplyVoltage:DINT;
            Mains supply voltage in mV.
 InvalidValue:BOOL;
     Indicates if the value is valid or not.
```

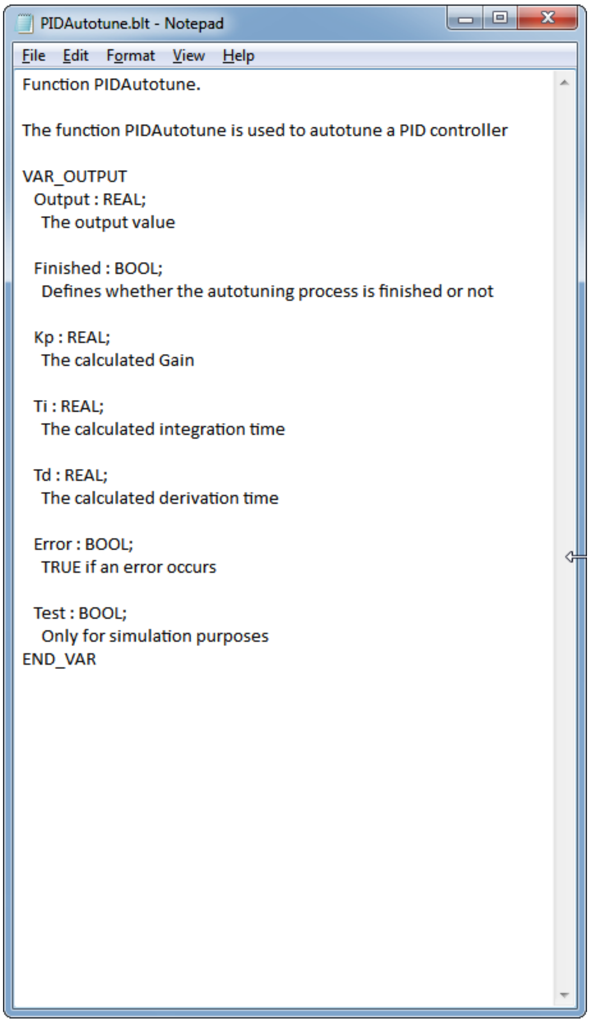
Only supported by RTU32E.

7.20 Function PIDAutotune.

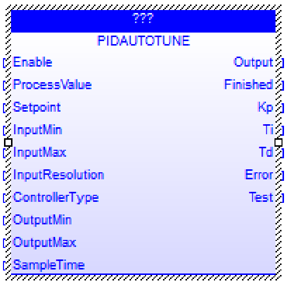
The function PIDAutotune is used to autotune a PID controller:



```
Function PIDAutotune.

The function PIDAutotune is used to autotune a PID controller

VAR_INPUT
  Enable : BOOL;
   Trigger to start the autotune on rising edge of signal

  ProcessValue : REAL;
   Process value

  Setpoint : REAL;
   Set point

  InputMin : REAL;
   Minimum of input signal

  InputMax : REAL;
   Maximum of input signal

  Resolution : USINT;
   Resolution of process value
   0 : 14 bit
   1 : 12 bit

  ControllerType : USINT;
   0 : P Controller (Only P calculated)
   1 : PI Controller (Only P & I calculated)
   2 : PD Controller (Only P & D calculated)
   3 : PID Controller (P,I & D calculated)

  OutputMin : REAL;
    Minimum of output signal

  OutputMax : REAL;
    Maximum of output signal

  SampleTime : TIME;
    Sampling period. Should be the same as for the PID Controller
END_VAR
```

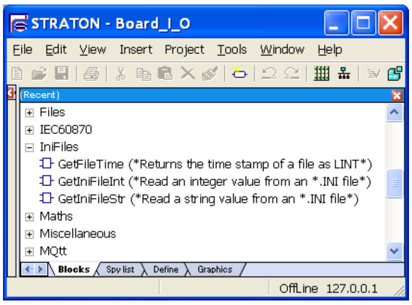


```
Function PIDAutotune.

The function PIDAutotune is used to autotune a PID controller

VAR_OUTPUT
  Output : REAL;
   The output value

  Finished : BOOL;
   Defines whether the autotuning process is finished or not

  Kp : REAL;
   The calculated Gain

  Ti : REAL;
   The calculated integration time

  Td : REAL;
   The calculated derivation time

  Error : BOOL;
   TRUE if an error occurs

  Test : BOOL;
   Only for simulation purposes
END_VAR
```



7.21 Reading Text Files from STRATON

To provide possibility to read variables and string from an external file in the RTU32, a few simple read file function is supported. The functions are found in the IniFiles folder.



- Files
- IEC60870
- IniFiles
  - 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*)
- Maths
- Miscellaneous
- MQtt

You have to place the text file in the RTU32 file system manually – use FTP to do this. MS Internet explorer can be used to explorer the RTU32 files. Use FTP instead of HTTP in the address field, like: ftp://192.168.100.243/. An empty folder will come up – right click in it and login using your username and password (same as used for entering the web pages).

The file structure must be as the SNMP.txt and RTU32.ini. The structure is simple and looks like this:

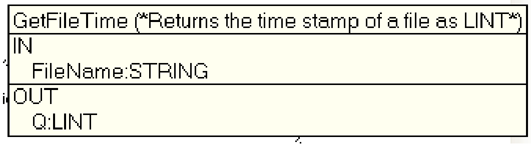
[ABC]
def=123
ghi=456

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

The functions available are:

- GetFileTime

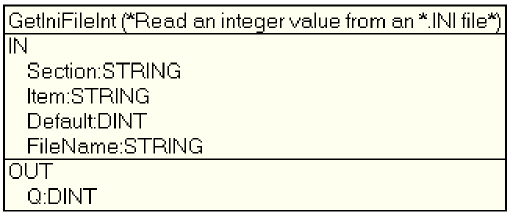
  With this function you can check the time of the file so you only read the variables in case of changes.



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

- GetIniFileInt

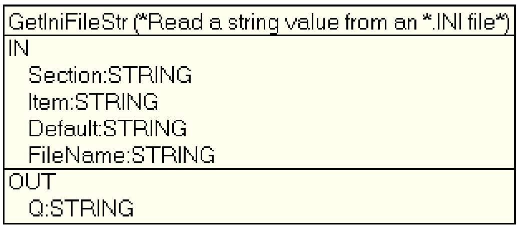
  Read an Integer value from a section and item.



```
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 idea with these functions is that you can place all your variables outside STRATON application program. You can e.g. use same application program for several sites and only have the differences in the file.

You can edit the file from the web editor with your browser, use the RTU32TOOL editor or directly send a new file to the RTU32 with FTP.

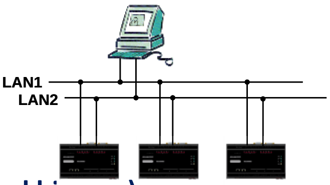
More help is found in STRATON by pressing F1.

8 Event Binding Protocol / Redundancy Dual Binding Communication

8.1 General

The STRATON Binding protocol is a proprietary protocol for fast time stamped event based data transfer directly from one STRATON runtime to another. The binding protocol is communicating directly from one RTU32 to another RTU32 in a LAN/WAN network. It can run disregard less of any other application. In a network with several RTU32s you can with a Global Binding Editor bind any global declared variables together. The same can be used between a STRATON PC runtime and an RTU32.

Dual binding is used in redundancy network connection using 2 separate network segments. Both the unit publishing variables (producer) and the unit subscribing variables (consumer) has to be equipped with 2 Ethernet network interfaces. Each network connection must be setup on the units as 2 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 will get the correct value. The first received event will be reported at the subscriber side and the same event with the same time stamp received secondly will be ignored.



8.2 Network settings in RTU32 (dual binding)

If you want to run dual binding using the RTU32 you must configure both Ethernet interfaces. No additional settings are necessary as the networks as seen as direct accessible parallel networks. Routing to the 2 networks is done as default. If you ping an address on any of the 2 selected network segments, the ping request will find its way to the correct segment automatically.

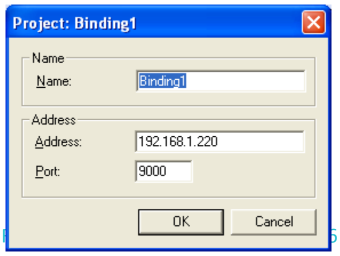
8.3 Procedure for setting up binding/dual binding in STRATON

You need to program all the RTU32 (and PC with STRATON runtime) before starting to bind (link) I/Os together. I/Os and variables to bind must be declared as Global variables. After making the program you must start the Global Binding Editor (find it under Start/Programs/STRATON/.)

The dual binding option is configured in the Global Binding Editor. See Global Binding Editor Help for general details of using the Global Binding Editor.

To add a redundancy communication network channel in the editor, you must enter 2 IP addresses of each variable you publish and subscribe.

Below figure shows the configuration window which will be added with additional configuration fields for an IP and port address.



Project: Binding1

Name
Name: Binding1

Address
Address: 192.168.1.220
Port: 9000

If you define 2 IP addresses (IP1/IP2), the compiler will automatically setup to use the two communication channels for parallel event reporting of events.

Get our How-to documents for details about setting up a binding application.

9 Optional SNMP Extension Agent Driver

The RTU32 can be delivered with support for SNMP Extension Agent Driver used for Network Alarm and Management applications. The SNMP Agent Driver provides SNMP protocol functions for Trap Alarming, Get/GetNext and Set Command. All features are fully configurable in webpages.

Note that if this option is not ordered, you will still have access to the Web configuration pages.

See the SNMP driver User's Manual for details.

10 RTU32 cycle / scan time

10.1 General

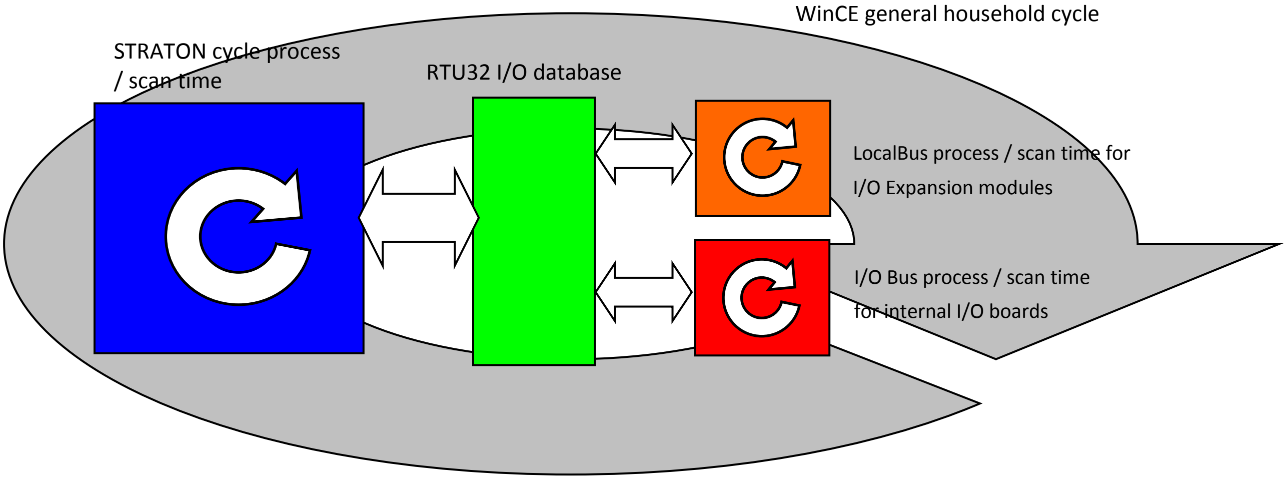
The RTU32 does in basic these include 4 processes which are relevant when determine timing performance for a RTU32:

STRATON EN/IEC61131 application program cycle/scan time

LocalBus cycle for expansion I/O modules

Internal I/O Bus cycle/scan time

Windows CE general household cycle


WinCE general household cycle
STRATON cycle process / scan time
RTU32 I/O database
LocalBus process / scan time for I/O Expansion modules
I/O Bus process / scan time for internal I/O boards

Each cycle are running fully independent (asynchronous) from each other and must be considered without internally synchronisation. The cycles does however have influence on each other as they all do load the same CPU and they are also assigned different priority in order to obtain the best possible overall performance. In general the processes for STRATON application program and I/O handling are running in high priority threads in Windows CE.

The LocalBus driver is internally communicating with a sub-processor, which is managing the low level LocalBus driver. It is running as fast as possible and the scan time is directly dependant on the number of connected LocalBus I/O Expansion modules. The LocalBus are reading/writing I/Os and errors on the I/O modules. The LocalBus itself is a simple but robust shift register bus.

Internal I/O Bus is managing the scan for the built-in I/Os in the RTU32. It is providing fast I/O updates because it uses an internal parallel I/O bus communication.

The STRATON program cycle is executing the actual application program developed in the Brodersen WorkSuite. It is also running in a high priority tread. The STRATON developing environment provides several functions to control and optimise the application program processes. Refer to STRATON Training course for details.

10.2 RTU32 Scan cycle mechanisms

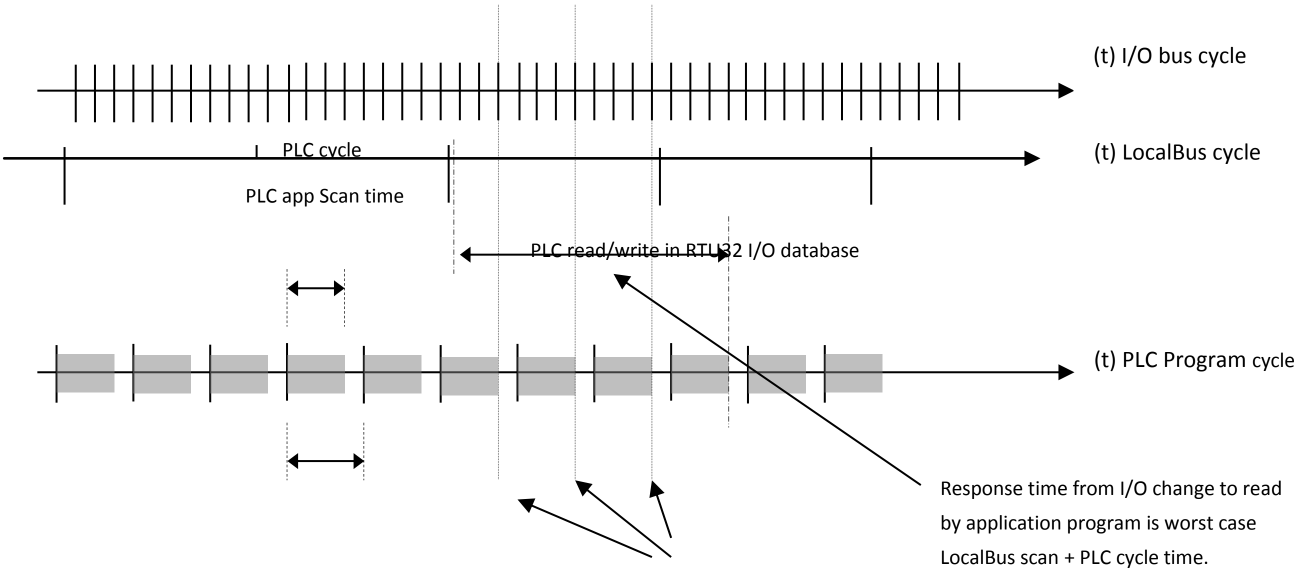
The 3 first tasks described above do run independent of each other.

The 2 I/O cycles are running as fast as possible to ensure that the basic RTU32 I/O database at any time reflect the physical I/Os – both internal I/O and I/O on the expansion modules connected.

The STRATON total program cycle time is depending of the size of the application program.

At the end of every STRATON cycle the I/Os defined in the application program is read/written to the basic RTU32 database. The STRATON scan time (time between each cycle) can be adjusted – see section STRATON scan time section for details.

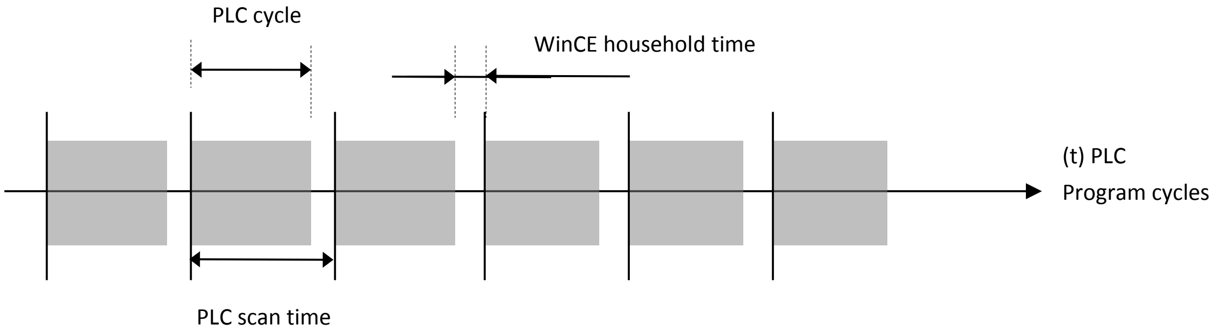
The link timing between these processes can best be explained in a graphical way.


(t) I/O bus cycle
PLC cycle
(t) LocalBus cycle
PLC app Scan time
PLC read/write in RTU32 I/O database
(t) PLC Program cycle
Response time from I/O change to read
by application program is worst case
LocalBus scan + PLC cycle time.

Note: The timing in the figure is shown to explain the link between the cycles. The timing of the cycles in a real application will be according to the actual STRATON application program size and number of connected I/O expansion modules.

10.3 STRATON scan time

The STRATON scan time can be set to 0 (zero) which automatically assign some time for operating system household between each STRATON program cycle. In this case the program runs as "fast" as possible. This is however not all correct. The automatic assignment of household execution time leaves relative much time for household to ensure all necessary operating system processes can be maintained with a good margin.



If you set the STRATON 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 to short time for household operations, the firmware will automatically ensure that your STRATON program do NOT compromise necessary household time. And it means that you will get a slower scan cycle time than adjusted.

To give a picture of the general STRATON application scan time performance, please see appendix 1 with some example benchmark measurements.

10.4 LocalBus scan time

The scan time of the local bus to Expansion I/O are running as fast as possible and is directly depending of the number of connection I/O Expansion modules. Data communication is a simple and reliable shift register bus.

The procedure for the communication is that the I/O modules are initialised at boot up. After that the internal I/O database in configured and the I/Os are started scanned. Cyclically the I/O configuration is checked for errors.

As a rule of thumb the following calculations and scan time examples can be used:

The maximum number of either input, or output connected determinate the scan time.

I.e. 16 x UCL-16DI configuration will have the same scan time as 16 x UCL-16DI, 16 x UCL-16DO.

As a rule of thumb the scan time could be calculated as 6 ms + number of modules (16 DI, DO) * 0,25ms

Configuration	Scan time *)	Calculation
1 x UCL-16DI	5..7 ms	(6 + 1 * 0.25)
16 x UCL-16DI	10..12 ms	(6 + 16 * 0.25)
32 x UCL-16DI	14..17 ms	(6 + 32 * 0.25)
32 x UCL-32DI	20..24 ms	(6 + 32 * 0.50)
32 x UCL-16DO	14..17 ms	(6 + 32 * 0.25)
32 x UCL-08AI	14..17 ms	(6 + 32 * 0.25)
16 x UCL-16DI, 16 x UCL-16DO	**)10..12 ms	(6 + 32 * 0.25)

*) As it is indicated the scan time varies during scan. That is caused by the cyclic configuration control of the I/O modules which is done occasionally.

**) The scan time reflect scanning of one analogue input or output on an analogue expansion module. As the analogue input expansion modules do include several inputs or outputs which are multiplexed, you must multiply the scan time with numbers of input/output to get the real scan time.

Example: If you use one UCL-08AI.D1 Expansion module with 8 AI on a RTU32, you must expect the scan time for each of the inputs to be 8 x (6 + 1 x 0,25) = 50ms. In other words it needs 8 total scans before all 8 AI on the modules has been read.

10.5 Internal I/O Bus scan time

The scan time of the RTU32 internal I/O are running as fast as possible. The communication is handled via mapped I/O in a sub CPU. The scan time is fixed and independent of any other tasks running in the RTU32, i.e. it has no influence how many I/O expansion modules that are connected to the RTU32.

The procedure for the communication is that the I/O board is initialised at boot up. After that the internal I/O database in configured and the I/Os are started scanned.

As a rule of thumb the following calculations and scan time can be used:

Configuration

I/O board configuration 26IO with 16 digital inputs (also working as counter inputs), 4 relay outputs, 4 configurable analogue inputs and 2 configurable outputs.

I/O type	Scan time

Digital inputs/outputs	5 ms
Analogue inputs	150 ms
Analogue outputs	50 ms

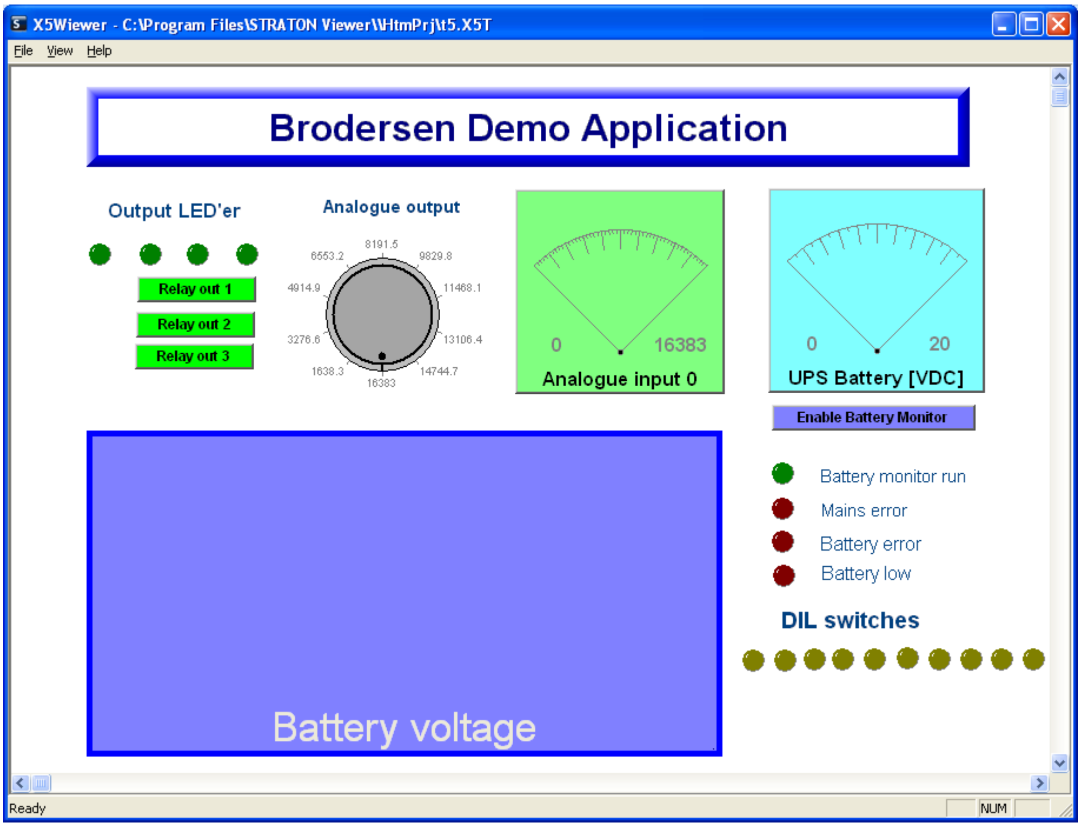
The differences between digital and analogue update time is caused by sampling time in analogue converters and the multiplexer. Also signal stabilizing in the electronic analogue circuits are contributing time to the scan time.

11 RTU HMI

11.1 General

The HMI function included standard in the RTU32 Series support simple HMI function developed in the WorkSuite tool. By adding a graphic program to your PLC application and download it to the RTU, makes it possible to monitor it online in the RTU with your PC using an IE browser or the special WorkSuite Viewer. The HMI graphics requires to be downloaded separately and is not downloaded automatically with the PLC runtime application.

Example:



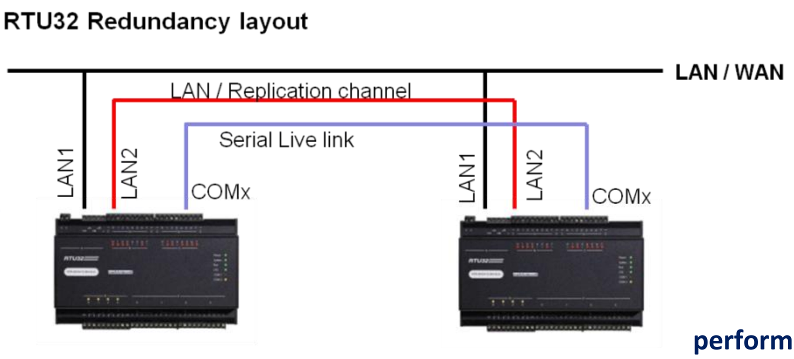
Please look in the HMI User Guide for details: *40277 200 RTU32 HMI User Guide.pdf*. The Guide is found in your WorkSuite-Help-Tutorial-Brodersen list of product documents.

12 RTU32 Redundancy

12.1 Redundancy introduction

RTU32 support a simple configurable STRATON runtime redundancy function that makes the RTU32 running in a redundancy setup. 2 RTU32 units can be configured to run redundancy – one active and one passive. When configured for redundancy on the configuration webpage, the 2 RTU32 in a configuration setup automatically establish redundancy mode. As a user you see the application as one RTU32 – you program and debug on one RTU32 and the redundancy function take care of establishing an equal mirror of the application and variable database to the other RTU32 in the redundancy setup (the passive partner).

During operation the active RTU32 handle application execution and the passive RTU32 is holding the application runtime as passive and on-the-fly are all runtime data updated between the RTU32 units in the setup. A live link channel must be used to optimize switch time and ensure that only one RTU32 is active in the setup.



12.2 Redundancy performance

The redundancy function provided in the RTU32 support redundancy application where switch time between active-passive of 200ms and upwards can be accepted. The switch-over time is much dependent on the communication link(s) between the redundancy partners. And the fact that it will take time to determine if a failure in reality has occurred. Typical switch time after tuning the time settings for an RTU32 redundant application will be in the area of 0,5 – 3sec.

12.3 Use and configure RTU32 in redundancy applications

If you are considering or planning a redundant configuration, please see the dedicated STRATON/RTU32 Redundant User Manual. And see the help in STRATON that hold information in the standard STRATON Redundancy the RTU32 support.

You will also find details of the special Redundancy Function Blocks in STRATON that is implemented to add functions to manage your redundant application. And you will find a detailed STRATON redundancy example on the RTU32 CD.

13 RTU32 Utilities and System Event Log

13.1 General

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

- Software Bulk Update Tool
- RTU32 Search Tool
- Event Viewer for RTU32 maintenance and configuration and Event Viewer for RTU32 system log monitoring
- Remote Display Tool

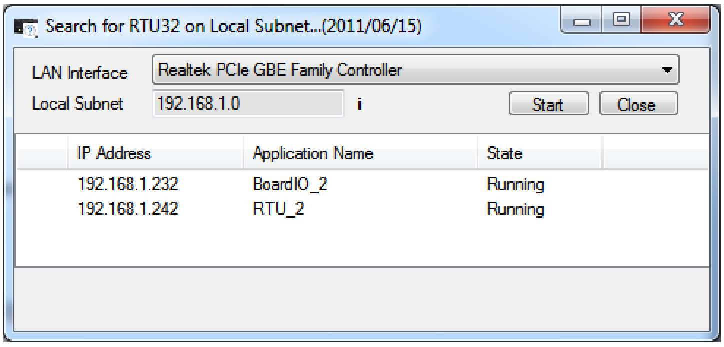
13.2 RTU32 Software Bulk Update Tool

See the Bulk Update Tool User Guide on CD for details.

13.3 RTU32 Search Tool

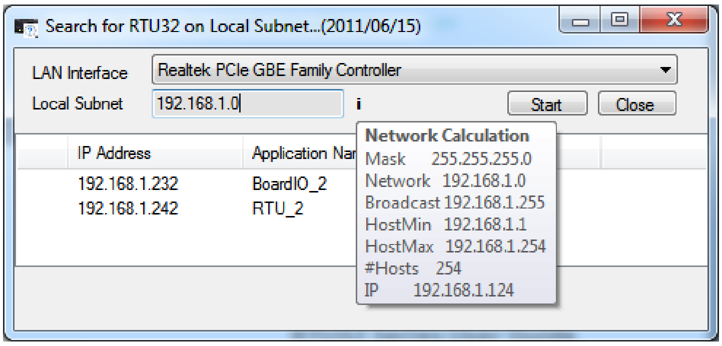
The RTU Search Tool is used to find RTUs on an Ethernet network. It will search using the network card/interface you define under LAN Interface. Be careful that you select the right one before you start the search procedure.

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



LAN Interface: Realtek PCIe GBE Family Controller
Local Subnet: 192.168.1.0

IP Address	Application Name	State
192.168.1.232	BoardIO_2	Running
192.168.1.242	RTU_2	Running



LAN Interface: Realtek PCIe GBE Family Controller
Local Subnet: 192.168.1.0

IP Address	Application Nar
192.168.1.232	BoardIO_2
192.168.1.242	RTU_2

Network Calculation

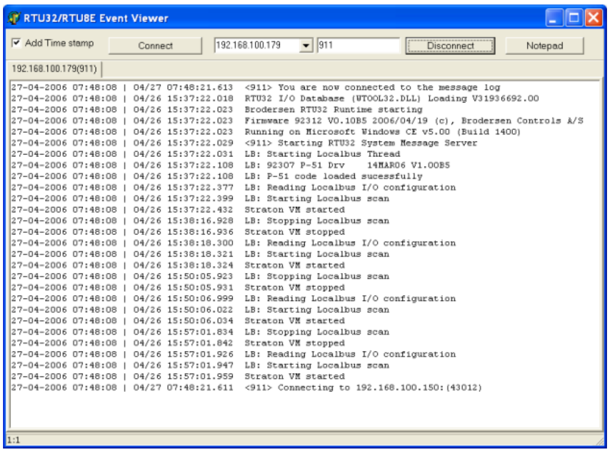
Mask	255.255.255.0
Network	192.168.1.0
Broadcast	192.168.1.255
HostMin	192.168.1.1
HostMax	192.168.1.254
#Hosts	254
IP	192.168.1.124

When you start the search process, the network will be scanned for Brodersen RTUs. Only RTUs with STRATON 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).

13.4 System Log/Event Viewer

The system log provides you with information about started processes in the RTU32, errors and status information's. You can furthermore add debug or information texts in your STRATON application program which will be printed in the system log. COM port communication can be logged if you add the right parameters to the COM port setup – 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 Event Viewer is a simple Telnet utility. The System Log is available on port 911 and look like this in the Event Viewer:



☑ Add Time stamp
192.168.100.179
911

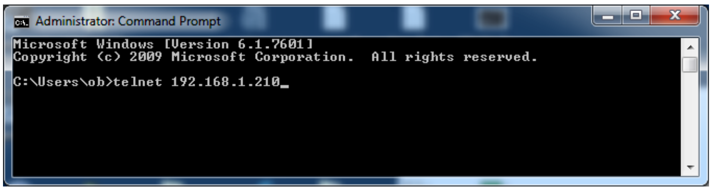
```
27-04-2006 07:48:08 | 04/27 07:48:21.613  <911> You are now connected to the message log
27-04-2006 07:48:08 | 04/26 15:37:22.018  RTU32 I/O Database (UTOOL32.DLL) Loading V31936692.00
27-04-2006 07:48:08 | 04/26 15:37:22.023  Brodersen RTU32 Runtime starting
27-04-2006 07:48:08 | 04/26 15:37:22.023  Firmware 92312 V0.10B5 2006/04/19 (c), Brodersen Controls A/S
27-04-2006 07:48:08 | 04/26 15:37:22.023  Running on Microsoft Windows CE v5.00 (Build 1400)
27-04-2006 07:48:08 | 04/26 15:37:22.029  <911> Starting RTU32 System Message Server
27-04-2006 07:48:08 | 04/26 15:37:22.031  LB: Starting Localbus Thread
27-04-2006 07:48:08 | 04/26 15:37:22.108  LB: 92307 P-51 Drv    14MAR06 V1.00B5
27-04-2006 07:48:08 | 04/26 15:37:22.108  LB: P-51 code loaded sucessfully
27-04-2006 07:48:08 | 04/26 15:37:22.377  LB: Reading Localbus I/O configuration
27-04-2006 07:48:08 | 04/26 15:37:22.399  LB: Starting Localbus scan
27-04-2006 07:48:08 | 04/26 15:37:22.432  Straton VM started
27-04-2006 07:48:08 | 04/26 15:38:16.928  LB: Stopping Localbus scan
27-04-2006 07:48:08 | 04/26 15:38:16.936  Straton VM stopped
27-04-2006 07:48:08 | 04/26 15:38:18.300  LB: Reading Localbus I/O configuration
27-04-2006 07:48:08 | 04/26 15:38:18.321  LB: Starting Localbus scan
27-04-2006 07:48:08 | 04/26 15:38:18.324  Straton VM started
27-04-2006 07:48:08 | 04/26 15:50:05.923  LB: Stopping Localbus scan
27-04-2006 07:48:08 | 04/26 15:50:05.931  Straton VM stopped
27-04-2006 07:48:08 | 04/26 15:50:06.999  LB: Reading Localbus I/O configuration
27-04-2006 07:48:08 | 04/26 15:50:06.022  LB: Starting Localbus scan
27-04-2006 07:48:08 | 04/26 15:50:06.034  Straton VM started
27-04-2006 07:48:08 | 04/26 15:57:01.834  LB: Stopping Localbus scan
27-04-2006 07:48:08 | 04/26 15:57:01.842  Straton VM stopped
27-04-2006 07:48:08 | 04/26 15:57:01.926  LB: Reading Localbus I/O configuration
27-04-2006 07:48:08 | 04/26 15:57:01.947  LB: Starting Localbus scan
27-04-2006 07:48:08 | 04/26 15:57:01.959  Straton VM started
27-04-2006 07:48:08 | 04/27 07:48:21.611  <911> Connecting to 192.168.100.150:(43012)
```

The RTU32 allows max 20 continuous connections and the log FIFO include max 100 messages.

13.5 Remote Display / Remote Desktop Tool

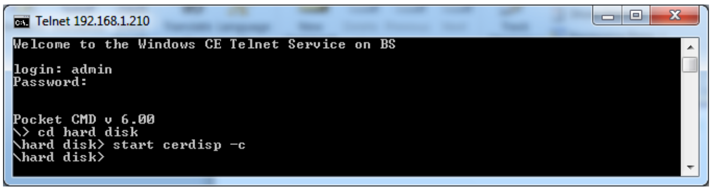
The RTU32 series includes a remote desktop function, where the Windows CE Cerdisp function can be used for showing your RTU32 Desktop on your local PC.

If you want to use the Remote Desktop function, you must start the program on your RTU32. Use telnet to connect to your RTU like this:



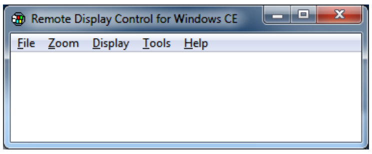
```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\ob>telnet 192.168.1.210_
```

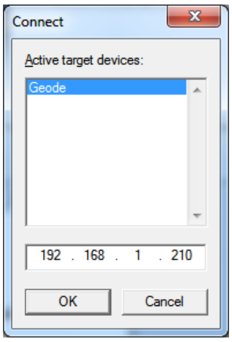
Enter your login and password when requested. After that change folder to Hard Disk and start the Cerdisp function as shown below:



```
Welcome to the Windows CE Telnet Service on BS

login: admin
Password:

Pocket CMD v 6.00
\> cd hard disk
\hard disk> start cerdisp -c
\hard disk>
```

Now you are ready to start the Remote Display Control for Windows CE window for getting access to the remote desktop. It is found under the /Brodersen RTU32 in your Start menu. After starting the RTU32 Remote Desktop Host you must use the File menu and select Connect. Now the Connect window will appear and you select the “Geode”, check the IP address and enter OK.



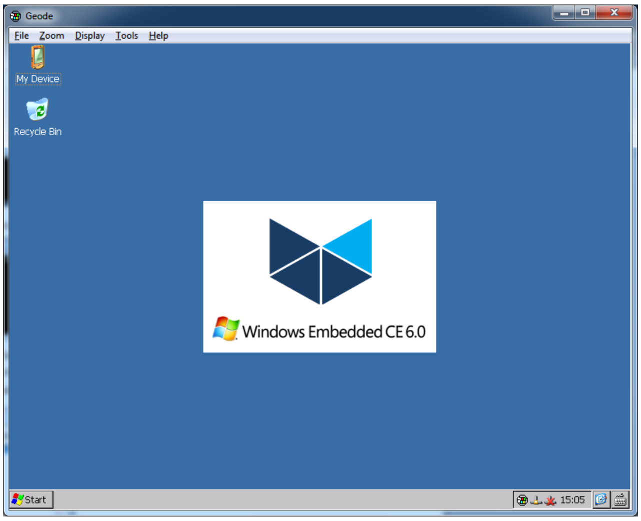


Active target devices:

Geode

192 . 168 . 1 . 210

- And now the remote desktop will appear on your computer



14 RTU32 Technical Description

14.1 General

The RTU32 RTU/PLC and Utility Controller are based on an Industrial PC platform running Windows CE operating system with all the well-known embedded Microsoft environment facilities. An industrial power supply is integrated to support industrial supply levels. The basic IPC includes a range of communication and other interfaces. The RTU32 provides additional interfaces like LocalBus for expansion I/O and integrated I/O.

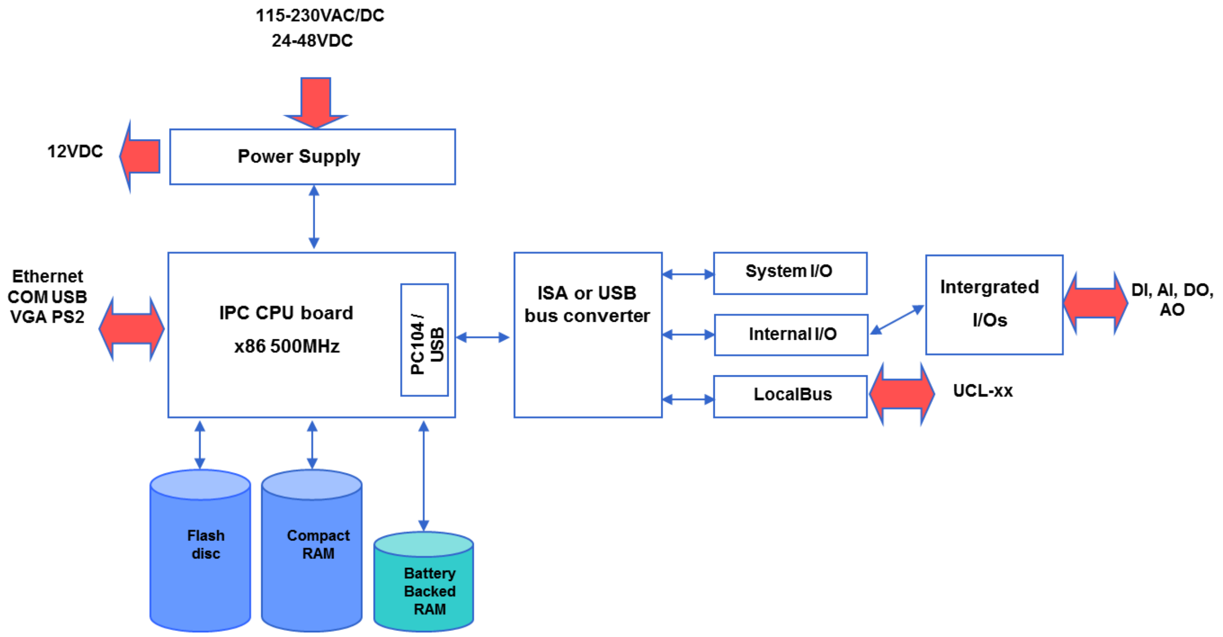
The RTU32 software is stored on a removable Flash. During start-up, the operating system and applications are moved to RAM where it is executed. Retained variables are stored executing a special command in the application program.

Windows CE is the RTU32 real time operating system. Integrated I/O and LocalBus for external I/O connectivity are controlled in an implemented I/O database. A STRATON PLC VM (Virtual Machine) is ported to the Windows CE real-time task. This enables the STRATON PLC runtime application program to be executed in the RTU32.

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.

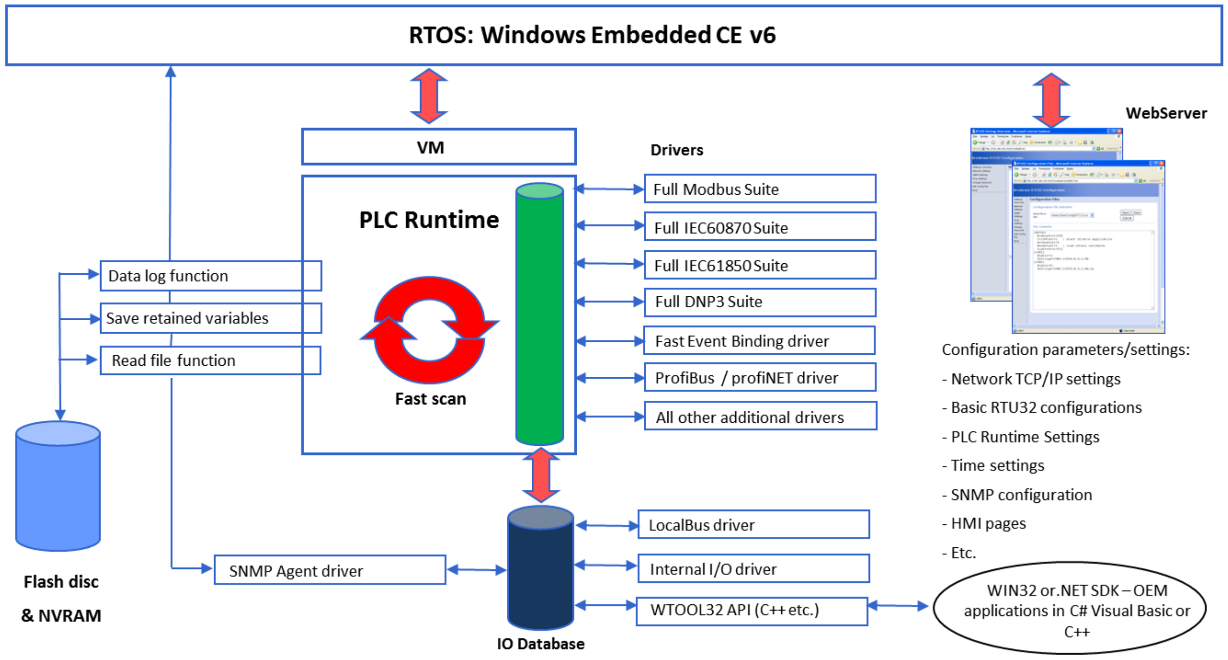
Detailed technical information about the RTU32 hardware is provided in the RTU32 Data Sheet and in the RTU32 Wiring and Mounting Instruction manual.

15 Hardware block diagrams



Be aware of that the block diagrams include both standard configuration and optional hardware components. See Data sheet for details.

10.3 Software block diagrams



The software block diagram shows the general software overview. If you require more info to explorer the RTU32 and it functions or possibilities, please contact Brodersen support.

15.1 VM PLC Runtime

The WorkSuite programming tool fully supports EN/IEC61131 and is used for making PLC programs in the RTU32. The application program kernel is implemented and runs in Windows CE real-time task. WorkSuite offers complete PLC functionality like in both normal PLC runtimes and in SoftPLCs, and supports all features needed in today's industrial environment. WorkSuite supports programming languages such as 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.

WorkSuite is an outstanding PLC product which 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.

15.2 RTU32 LocalBus for I/O Expansion

15.2.1 General

The LocalBus is physically a simple string of shift registers, where data is clocked serial, and latched parallel to I/O. The raw data is shifted to / from two data buffers, and processed by type ID dependent functions into data type sorted IO-data structure. The IO-data structures operate with the data types as defined in the STRATON I/O Driver.

15.2.2 LocalBus (LB) driver

The LocalBus driver is running cyclic in a high priority thread, and always started at RTU32 boot time. However start and stop of LB scan is controlled by STRATON application.

When started, LB configuration is read, and database is built accordingly. The driver has knowledge of every module type layout, and builds an IO-data structure which matches the physical connected modules. If a module of unknown type is connected, the IO-data structure could not be build, LB scan is not started, and an error is reported in the system event log.

After reading the LocalBus configuration, a copy of the IO data structure is also saved on the flash. The reason for that is to support a lock function which freezes the LocalBus configuration. If the lock is enabled the configuration read at STRATON application start are compared to the saved configuration on the Flash. In case the configurations do not match, the local bus scan is not started and an error is reported. The lock is implemented as a configuration option on the I/O Config webpage. In case the lock function is disabled (default), the configuration is just copied to the Flash.

NOTE: The LocalBus driver has the limitation of max 32 external I/O Expansion modules.

15.2.3 LB driver event information

Information and status of the I/O drivers are reported in the system event log read at port 911. The information about the LocalBus driver **is** identified by "LB:" and include information and status like:

```
10/27 10:26:01.994  LB: Starting Localbus Thread
10/27 10:26:02.079  LB: P-51 Firmware 92307 V1.11     2007/04/23
10/27 10:26:02.079  LB: P-51 code loaded sucessfully
10/27 10:26:02.201  LB: Internal I/O board type 26IO is detected
10/27 10:26:03.213  LB: AVR Firmware 92391 V1.05  2006/07/17
10/27 10:26:03.797  LB: Reading Localbus I/O configuration
10/27 10:26:03.881  LB: The following I/O modules are attached to the RTU:
10/27 10:26:03.887  LB: UCx-26 IOA.Dx      16 DI, 4 AI Multi V/I, 4 DO, 2 AO Multi V/I
10/27 10:26:03.889  LB: UCx-16 DIO.P1      16 bit dig. inp. & out. DC
10/27 10:26:03.900  LB: Starting Localbus scan
10/27 10:26:03.930  Straton VM started
```

If the I/O configuration changed while the RTU32 is running or changed while locked, it will be reported in the log like:

```
10/27 10:29:22.395  LB: Error, Localbus I/O configuration mismatch detected
```

If you have locked the I/O configuration, and the physical I/O configuration is not correct you will get a report in the Event log. It will tell you what configuration was expected and what is actually detected.

```
10/27 10:31:41.974  LB: Reading Localbus I/O configuration
10/27 10:31:42.006  LB: Warning, I/O config is locked by user, and mismatch is detected
10/27 10:31:42.007  LB: The following I/O modules was expected attached to the RTU:
10/27 10:31:42.013  LB: UCx-26 IOA.Dx      16 DI, 4 AI Multi V/I, 4 DO, 2 AO Multi V/I
10/27 10:31:42.015  LB: UCx-16 DIO.P1      16 bit dig. inp. & out. DC
10/27 10:31:42.015  LB: The following I/O modules are attached to the RTU:
10/27 10:31:42.019  LB: UCx-26 IOA.Dx      16 DI, 4 AI Multi V/I, 4 DO, 2 AO Multi V/I
```

The Event log messages are useful for service people to detect errors in configuration.

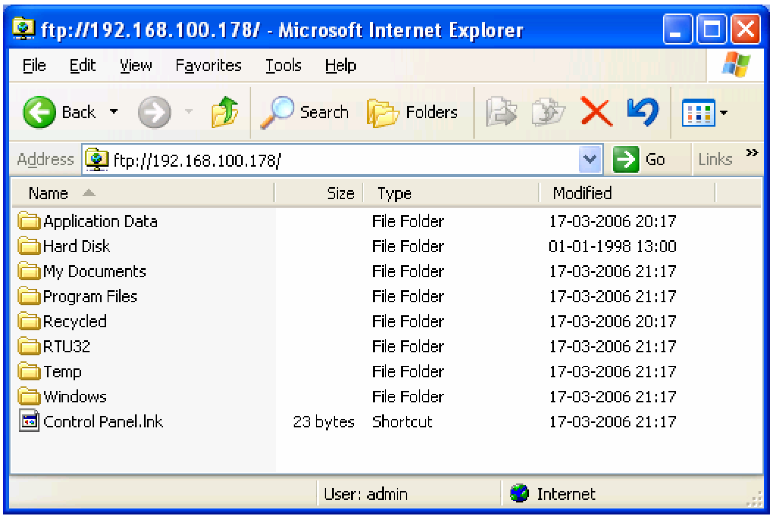
15.3 RTU32 File System

15.3.1 General

When you apply power to the RTU32, it will boot from the flash disk. All necessary files for OS, STRATON kernel VM, drivers, application program etc. are located on the flash disc.

During boot the OS is moved to run on the internal RAM. The files located on the flash disc will be moved to the /hard disk/folder.

If you access the running RTU32 with FTP you will see the root directory like:


ftp://192.168.100.178/ - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Folders
Address ftp://192.168.100.178/ Go Links
User: admin
Internet

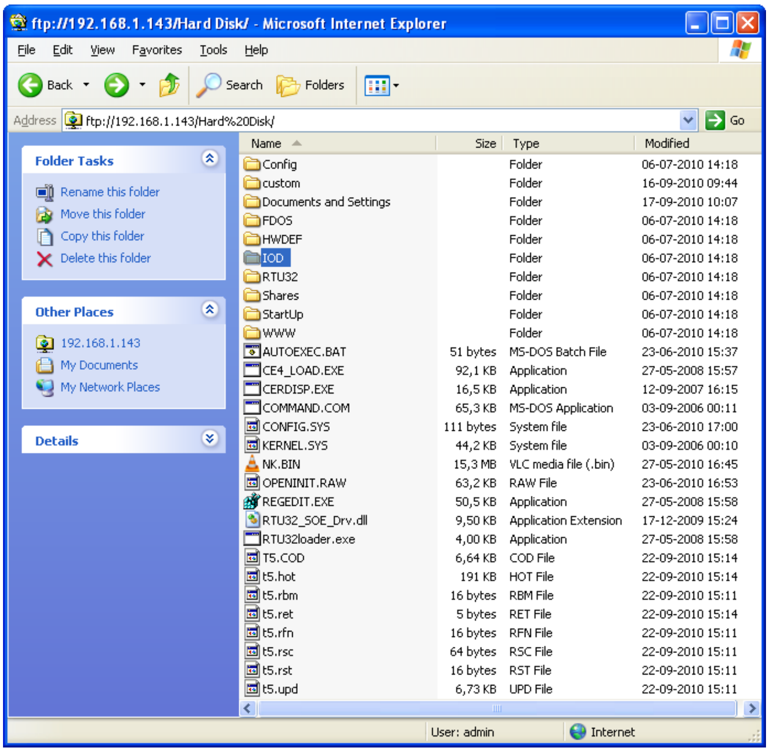
Name	Size	Type	Modified
Application Data		File Folder	17-03-2006 20:17
Hard Disk		File Folder	01-01-1998 13:00
My Documents		File Folder	17-03-2006 21:17
Program Files		File Folder	17-03-2006 21:17
Recycled		File Folder	17-03-2006 20:17
RTU32		File Folder	17-03-2006 21:17
Temp		File Folder	17-03-2006 21:17
Windows		File Folder	17-03-2006 21:17
Control Panel.lnk	23 bytes	Shortcut	17-03-2006 21:17

The root directory contains a number of subfolder and system files.

What do the folders contain?

../Hard disk/ folder

Contain all the files which are actually on the flash disk.

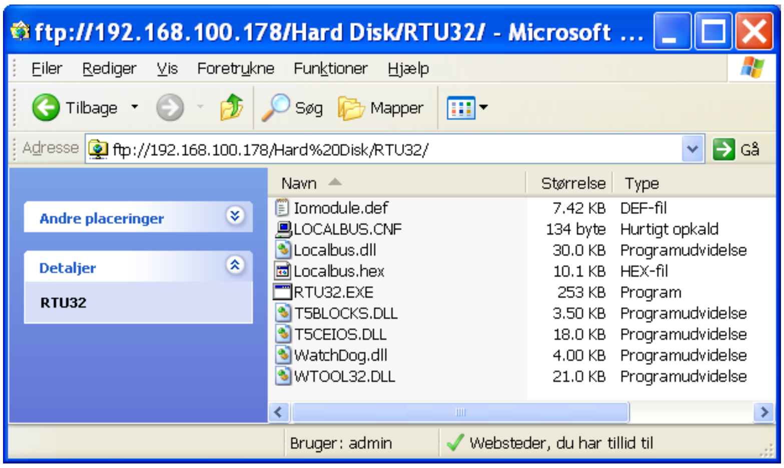


Name	Size	Type	Modified
Config		Folder	06-07-2010 14:18
custom		Folder	16-09-2010 09:44
Documents and Settings		Folder	17-09-2010 10:07
FDOS		Folder	06-07-2010 14:18
HWDEF		Folder	06-07-2010 14:18
IOD		Folder	06-07-2010 14:18
RTU32		Folder	06-07-2010 14:18
Shares		Folder	06-07-2010 14:18
StartUp		Folder	06-07-2010 14:18
WWW		Folder	06-07-2010 14:18
AUTOEXEC.BAT	51 bytes	MS-DOS Batch File	23-06-2010 15:37
CE4_LOAD.EXE	92,1 KB	Application	27-05-2008 15:57
CERDISP.EXE	16,5 KB	Application	12-09-2007 16:15
COMMAND.COM	65,3 KB	MS-DOS Application	03-09-2006 00:11
CONFIG.SYS	111 bytes	System file	23-06-2010 17:00
KERNEL.SYS	44,2 KB	System file	03-09-2006 00:10
NK.BIN	15,3 MB	VLC media file (.bin)	27-05-2010 16:45
OPENINIT.RAW	63,2 KB	RAW File	23-06-2010 16:53
REGEDIT.EXE	50,5 KB	Application	27-05-2008 15:58
RTU32_SOE_Drv.dll	9,50 KB	Application Extension	17-12-2009 15:24
RTU32loader.exe	4,00 KB	Application	27-05-2008 15:58
T5.COD	6,64 KB	COD File	22-09-2010 15:14
t5.hot	191 KB	HOT File	22-09-2010 15:14
t5.rbm	16 bytes	RBM File	22-09-2010 15:11
t5.ret	5 bytes	RET File	22-09-2010 15:14
t5.rfn	16 bytes	RFN File	22-09-2010 15:11
t5.rsc	64 bytes	RSC File	22-09-2010 15:11
t5.rst	16 bytes	RST File	22-09-2010 15:11
t5.upd	6,73 KB	UPD File	22-09-2010 15:11

If you want to save files to the Flash in the RTU32, they should always be saved in the ../Hard Disk/ folder.

./RTU32/ folder

Contain the RTU32 specific DLL driver files.



ftp://192.168.100.178/Hard Disk/RTU32/ - Microsoft ...

Adresse ftp://192.168.100.178/Hard%20Disk/RTU32/

RTU32

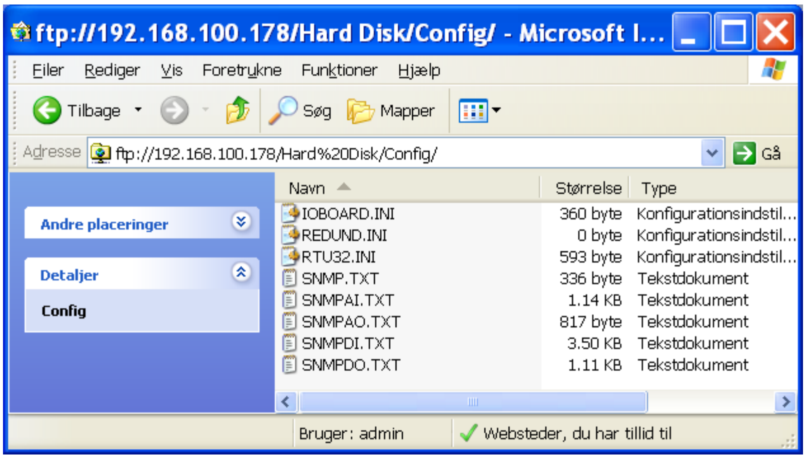
Navn	Størrelse	Type
Iomodule.def	7.42 KB	DEF-fil
LOCALBUS.CNF	134 byte	Hurtigt opkald
Localbus.dll	30.0 KB	Programudvidelse
Localbus.hex	10.1 KB	HEX-fil
RTU32.EXE	253 KB	Program
T5BLOCKS.DLL	3.50 KB	Programudvidelse
T5CEIOS.DLL	18.0 KB	Programudvidelse
WatchDog.dll	4.00 KB	Programudvidelse
WTOOL32.DLL	21.0 KB	Programudvidelse

Bruger: admin ✓ Websteder, du har tillid til

The RTU32.exe file is the actual virtual machine file for executing the STRATON application program.

../Config/ folder

The folder contains the configuration files. The RTU32.ini file defined the basic RTU32 configurations. The SNMP.txt file is used by the optional SNMP Agent driver. If you are going to use the read text file function in STRATON, the file should be placed in this folder



ftp://192.168.100.178/Hard Disk/Config/ - Microsoft I...

Adresse ftp://192.168.100.178/Hard%20Disk/Config/

Config

Navn	Størrelse	Type
IOBOARD.INI	360 byte	Konfigurationsindstil...
REDUND.INI	0 byte	Konfigurationsindstil...
RTU32.INI	593 byte	Konfigurationsindstil...
SNMP.TXT	336 byte	Tekstdokument
SNMPAI.TXT	1.14 KB	Tekstdokument
SNMPAO.TXT	817 byte	Tekstdokument
SNMPDI.TXT	3.50 KB	Tekstdokument
SNMPDO.TXT	1.11 KB	Tekstdokument

Bruger: admin ✓ Websteder, du har tillid til

Flash Disc Files

The files on the Flash disc contain all the files necessary to run the RTU32. As mentioned before the files equal the ../Hard Disk/ folder you will find on a running RTU32 using FTP.

The NK.BIN is the Windows CE image file.

The t5.* is the STRATON runtime files. If you by any reason have a problem with an application program freezing the RTU32, you can delete these files and the runtime application deleted. You can now download a new application from the Brodersen WorkSuite.

The /Documents and Settings/ folder contain all you special settings for Windows CE – including network setup etc. If you delete this folder and all its content, the RTU32 will start-up as a virgin RTU32 (as delivered from the factory).

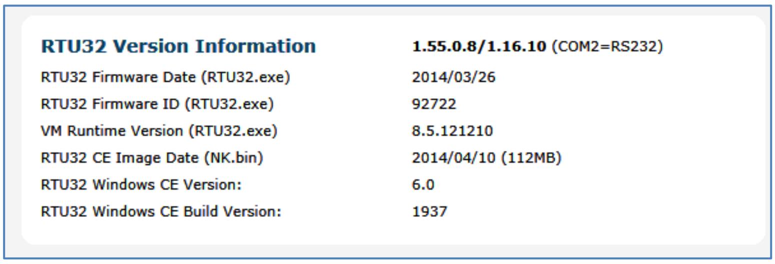
The /StartUp folder is used if you want some special application to be started when the RTU32 boot up. All executable files and dll placed in this folder is started.

The HWDEF folder contains the Brodersen WorkSuite hardware library for RTU32. These files are not used by the RTU32.

The RTU32loader.exe does load all the dedicated files in the RTU32 folder during boot.

15.4 Software Versions

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



RTU32 Version Information	**1.55.0.8/1.16.10** (COM2=RS232)
RTU32 Firmware Date (RTU32.exe)	2014/03/26
RTU32 Firmware ID (RTU32.exe)	92722
VM Runtime Version (RTU32.exe)	8.5.121210
RTU32 CE Image Date (NK.bin)	2014/04/10 (112MB)
RTU32 Windows CE Version:	6.0
RTU32 Windows CE Build Version:	1937

The software in the RTU32 includes 3 main software parts:

- Basic RTU32 Version – like above *1.55.0.8/1.16.10*
- VM Runtime version
- Windows CE image

The VM runtime project details include name and built version of the PLC runtime running in the RTU32.

RTU32 Firmware includes the specific RTU32 functions and do include all files in the ../RTU32/ folder on the Flash. The RTU32 firmware is assigned an id (e.g. 92312), a date and a version

RTU32 CE Image is the single NK.BIN file is assigned a date and version.

VM version is the version of the Virtual Machine SDK supported by the RTU32 (Virtual Machine = PLC runtime handler). It provides in general information of what versions of Brodersen WorkSuite is supported by the RTU32. The RTU32 is developed and maintained to support full compatibility, but the VM version tells you if the latest STRATON functions is supported in the RTU32.

Windows CE version is the MS Platform builder and is defined by a general version and a build version.

If you contact Brodersen or your local distributor / system integrator for support, it is important that you have the **RTU32 Version Information** ready together with the serial number of your device(s).

16 History

16.1 New features since last update

The table below highlights changes and enhancements made between the versions.

RTU32 VM	CE image	User Manual	Added functions and features
1.55	1.16.10 CE 6.3	1.55	• Released as version 1.55.0.9 / 1.16.10. • Run on RTU32, RTU32R and RTU32E. RTU32S is using same base version which is adapted for ARM platform in the RTU32S. • The firmware supports the new RTU32E series with 64 IO board • Support 3G Modem and VPN. • New function block and Function library added: ○ RTU32 Status functions. ○ PID Autotune function. • Hardware overview webpage added to monitor online status on physical I/O. • Three levels Security login to Web-, FTP- and Telnet Server added. • Driver version view added to webpage Settings overview. • SNMP driver updated to support 512 DI • Software programming tool STRATON replaced by OEM version Brodersen WorkSuite where all hardware library is included to support all new features added in v 1.55.0.9.
1.52	1.16.03 CE 6.3	1.52	• Released as version 1.51 / 1.16.03 • The firmware supports the new RTU32R series (19 inch rack) with 60 IO board with 0-20 V inputs • Updates to the 103 Master driver to support command direction. • New function block added IECStatusEx to read status of 103 slaves. • RTU32 TOOL CD V1.52 is able to install on STRATON V8.3 Build 2 installation.
1.51	1.16.03 CE 6.3	1.51	• Released as version 1.51 / 1.16.03 • IEC61850 Client Driver updated with simplified functions for handling both local and remote SCL files. • IEC60870 handle storage increased from 60 to 120 handles. • IEC60870-5-103 master protocol enabled and includes some basic function in standard version. • Image adjusted to enable zenon SCADA start and run. • USB_RTUIO driver added in BETA version. • Remote Desktop updated for Windows CE 6.0 • Watchdog timer for timer task changed from 1 to 5 seconds. • RTU32 CD v151 updated with BulkUpdateTool (replace old firmware update tool) • Brodersen DNP3 driver installed into STRATON Workbench
1.50	1.16.02 CE 6.3	1.50	• Microsoft CE version 6 SR3 is used. • STRATON VM 8.1 is implemented. • The firmware supports the new RTU32R series (19 inch rack) with 60 IO board. • The firmware supports the new RTU32 with 8 RS485 Optical isolated COM ports as COM1...COM8. • The RTU32 Tool CD contains two new applications: *RTU32Search* and *RTU32 Remote Desktop Host* which can be accessed from the Start menu when installed. • The old RTU32Update program is now dismissed and no longer supported. A new RTU32 Bulk Update program has now available on request. • A better support for the 61850 SCL files that are downloaded by STRATON WB. • RTU32 Tool CD V1.50 is delivered together with the RTU32. • Enhanced UserReg.reg file to adjust Windows CE registry. Read UserReg.info found on the RTU32 flash for more details. • New and adjusted function blocks: - ClearFifoQueue for clearing 60870 transmissions buffers.

			- SetCsvOption to set separations chars for the LogCSV and LogFileCSV functions. - IEC101SlvState contains new options to read toggle bit counter and comm counter and to enable listen mode on the port.
1.49	1.10.12 CE 5.0	1.49	• System log is kept when non-volatile RAM option is included in the RTU32. • New clock parameter (3) added to DAY_TIME function. Deliver full complete time stamp incl. millisec in a string variable. • New function NetworkAdapter added. Returns information of the specified network adapter. • New function LOGCSV added. See section 7 for details. • Redundant adjustments. Now replication channel also acts as live link together with the serial link. • FreeDOS is now used for booting and loading Windows CE 5.0. • Registry changed to support new browser versions in Windows 7. • Full support for DNP3 Server/Slave. • RTU32 Tool CD V1.49 is delivered together with the RTU32.
1.45	1.10.07 CE 5.0	1.45	•

16.2 Important notes about the latest versions

In updates of functions and features, we sometime see it necessary to do implementations there can give compatibility problems when using STRATON or WorkSuite application program developed for earlier versions. In this update please note as follows:

Version 1.46: FreeDOS system added means that the CompactFlash used for firmware has to be formatted for FreeDOS. That means in practical life that you cannot use Flash discs prior to version 1.46 as the formatting is NOT compliant with the new versions (V1.46+) of RTU32 software/firmware. We can however provide you with a tool to format the flash for FreeDOS.

Appendix 1

RTU32 STRATON applications measurement / benchmark

The RTU32 STRATON runtime cycle time is measured on a number of different applications. Small or large application with or without serial communication. The RTU32 scan time is in STRATON set to 0, which means that STRATON automatically assigns time for household. This setting is recommended as it ensures that your application will avoid any overrun situations.

Example 1: STRATON application program without any additional serial communication

Small test configuration:

1 x RTU32 without internal I/O

1 x UCL-16DIO Expansion module

1 x UCL-08AI Expansion module

2 x UCL-04AO Expansion modules

All 8 analogue inputs are scaled, and some simple math executed, before stored in an internal register, and copied to an analogue output. (Approx. 15 STRATON instructions pr. channel).

10 digital inputs are controlling one timer each, which output is connected to a digital output.

6 digital inputs are controlling 6 counters, which output is connected to a digital output.

Result:

The typical STRATON scan time is 2ms.

Large test configuration:

1 x RTU32 without internal I/O

6 x UCL-16DIO Expansion modules

6 x UCL-08AI Expansion modules

12 x UCL-04AO Expansion modules

All 48 analogue inputs are scaled, and some simple math executed, before stored in an internal register, and copied to an analogue output (approx. 15 STRATON instructions pr. channel).
40 digital inputs are controlling one timer each, which output is connected to a digital output.

30 digital inputs are controlling 30 counters, which output is connected to a digital output.

Result:

The typical STRATON scan time 4ms.

Example 2: STRATON application program with Modbus communication

A number of tests are executed on the STRATON build in Modbus slave / master driver, using the Fieldbus configurator.

Small test configuration:

1 x RTU32 without internal I/O

1 x UCL-16DIO Expansion module

1 x UCL-08AI Expansion module

2 x UCL-04AO Expansion modules

All 8 analogue inputs are scaled, and some simple math executed, before stored in an internal register, and copied to an analogue output. (Approx. 15 STRATON instructions pr. channel).

10 digital inputs are controlling one timer each, which output is connected to a digital output.

6 digital inputs are controlling 6 counters, which output is connected to a digital output.

ModbusRTU Slave

Modbus RTU slave on COM1 9600 baud. 50 words are read/write in each frame.

Modbus TCP slave. 50 words are read/write in each frame.

Result:

There is no noticeable change in STRATON scan time, whenever communication was used or not. STRATON scan time is approx. 2ms.

ModbusTCP Client

Modbus TCP master. 50 words are read/write in each frame.

Result:

There is no noticeable change in STRATON scan time whenever communication was used or not. STRATON scan time is approx. 2ms.

Example 3: STRATON application program with additional STRATON binding protocol

RTU32 is running the standard application, and using binding protocol to transfer events to another RTU32.

STRATON scan time is measured on the following applications.

Small test configuration:

1 x RTU32 without internal I/O

1 x UCL-16DIO Expansion module

1 x UCL-08AI Expansion module

2 x UCL-04AO Expansion modules

All 8 analogue inputs are scaled, and some simple math executed, before stored in an internal register, and copied to an analogue output (approx. 15 STRATON instructions pr. channel).

10 digital inputs are controlling one timer each, which output is connected to a digital output.

6 digital inputs are controlling 6 counters, which output is connected to a digital output.

Binding setup

16 digital points (BOOL) and 8 analogue points (DINT) are transferred to another RTU32 every 1sec. The points are internal memory points, and are all changed in the same STRATON cycle.

Result:

The typical STRATON scan time is 3ms.

If data transfer are stopped (change of binding variables are stopped), STRATON scan time is still typical 3ms. I.e. there is no noticeable (less than 1ms.) change in performance if events are generated or not.

Large test configuration:

1 x RTU32 without internal I/O

6 x UCL-16DIO Expansion modules

6 x UCL-08AI Expansion modules

12 x UCL-04AO Expansion modules

All 48 analogue inputs are scaled, and some simple math executed, before stored in an internal register, and copied to an analogue output (approx. 15 STRATON instructions pr. channel).

40 digital inputs are controlling one timer each, which output is connected to a digital output.

30 digital inputs are controlling 30 counters, which output is connected to a digital output.

Binding setup

96 digital points (BOOL) and 48 analog points (DINT) are transferred to another RTU32 every 1sec. The points are internal memory points, and are all changed in the same STRATON cycle.

Result:

The typical STRATON scan time is 4 to 7ms.

If data transfer are stopped (change of binding variables are stopped), STRATON scan time is typical 4ms. I.e. when variables are changed additional time (3ms) is added to generate the events and communication.

Example 4: STRATON application program with IEC-60870-5-101 communication

Small test configuration:
One EN/IEC60870-5-101 Slave unbalanced mode on COM2 9600 baud.
STRATON scan time is measured on the following application.
8 single point inputs (single point information)
8 pulse outputs (single command)

Result:
The typical STRATON scan time is 2 to 3ms.

Large test configuration:
One EN/IEC60870-5-101 Slave unbalanced mode on COM2 9600 baud.
STRATON scan time is measured on the following applications.
96 single point inputs (single point information)
96 pulse outputs (single command)

Result:
The typical STRATON scan time is 4 to 5ms.

Small test configuration:
One IEC60870-5-104 Server. STRATON scan time is measured on the following application.
8 single point inputs (single point information)
8 pulse outputs (single command)

Result:
The typical STRATON scan time is 2 to 3ms.

Large test configuration:
One IEC60870-104 Server. STRATON scan time is measured on the following applications.
96 single point inputs (single point information)
96 pulse outputs (single command)

Result:
The typical STRATON scan time is 4 to 5ms.

Comments to the example and measurements

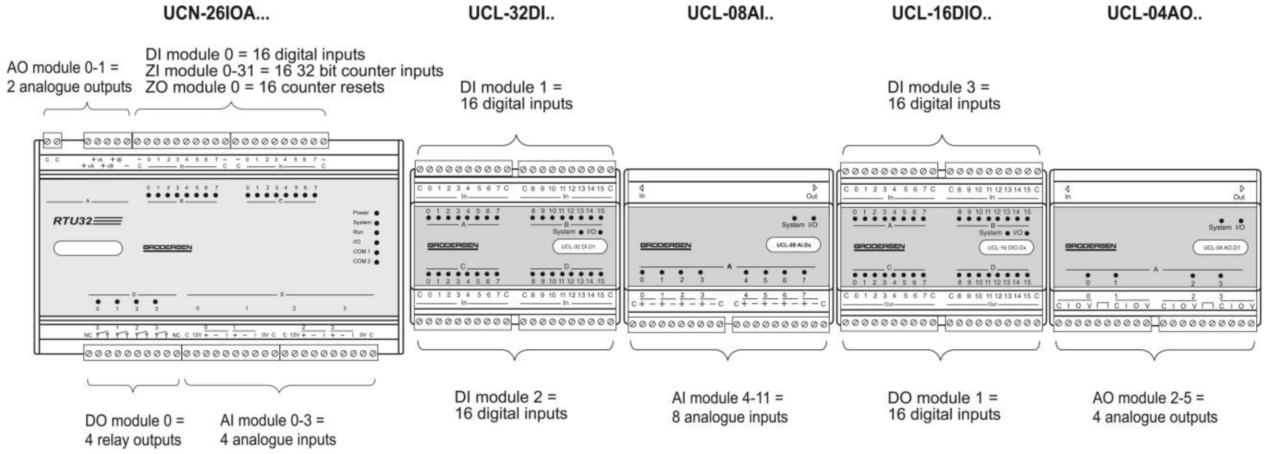
There are small differences in STRATON scan time whenever IEC60870-5-101 or IEC60870-5-104 communication is used. The ModbusRTU slave test showed also that no noticeable time was added to the STRATON cycle, due serial communication. It could be expected to be the same case for IEC60870-5-101 and IEC60870-5-104 communication.

Windows CE remote performance monitor show the different communication types tested only contribute with a few percent (less than 5) of the CPU load. If many COM ports are added to the system, the load will increase, but should not be of general concern.

High load ping test at the Ethernet port show impact on the STRATON cycle time, and the large STRATON binding application test, show an increase in cycle time when all data are exchanged. I.e. the use of binding protocol to many RTUs, where lots of data are transferred at high rate, should be considered carefully if fast timing is an issue of concern.

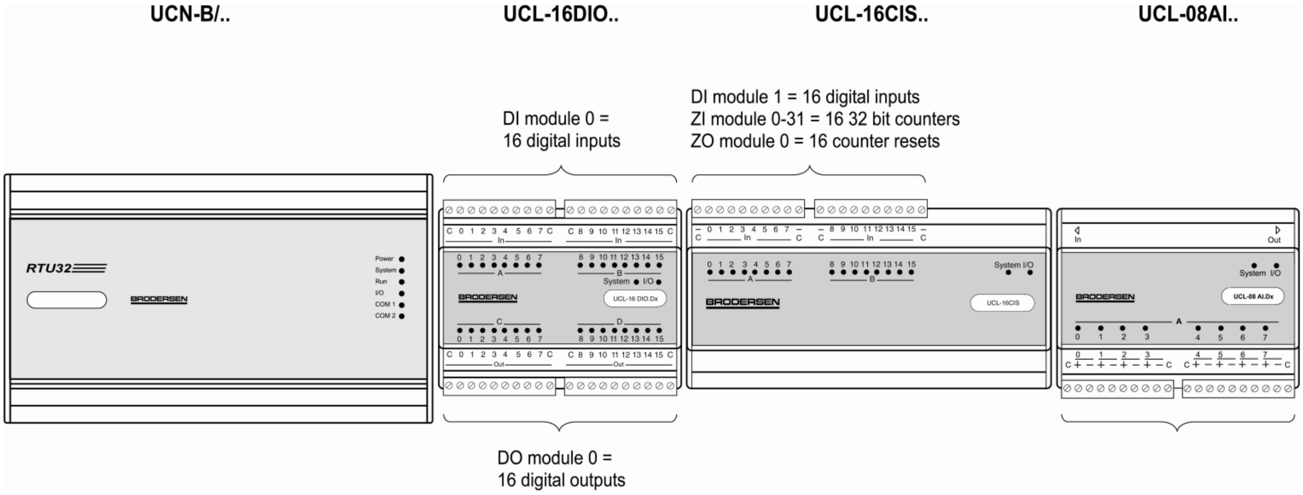
Appendix 2

RTU32 I/O addressing example 1


UCN-26IOA...
UCL-32DI..
UCL-08AI..
UCL-16DIO..
UCL-04AO..
AO module 0-1 = 2 analogue outputs
DI module 0 = 16 digital inputs
ZI module 0-31 = 16 32 bit counter inputs
ZO module 0 = 16 counter resets
DI module 1 = 16 digital inputs
DI module 3 = 16 digital inputs
RTU32
BRODERSEN
DO module 0 = 4 relay outputs
AI module 0-3 = 4 analogue inputs
DI module 2 = 16 digital inputs
AI module 4-11 = 8 analogue inputs
DO module 1 = 16 digital inputs
AO module 2-5 = 4 analogue outputs

Appendix 3

RTU32 I/O addressing example 2



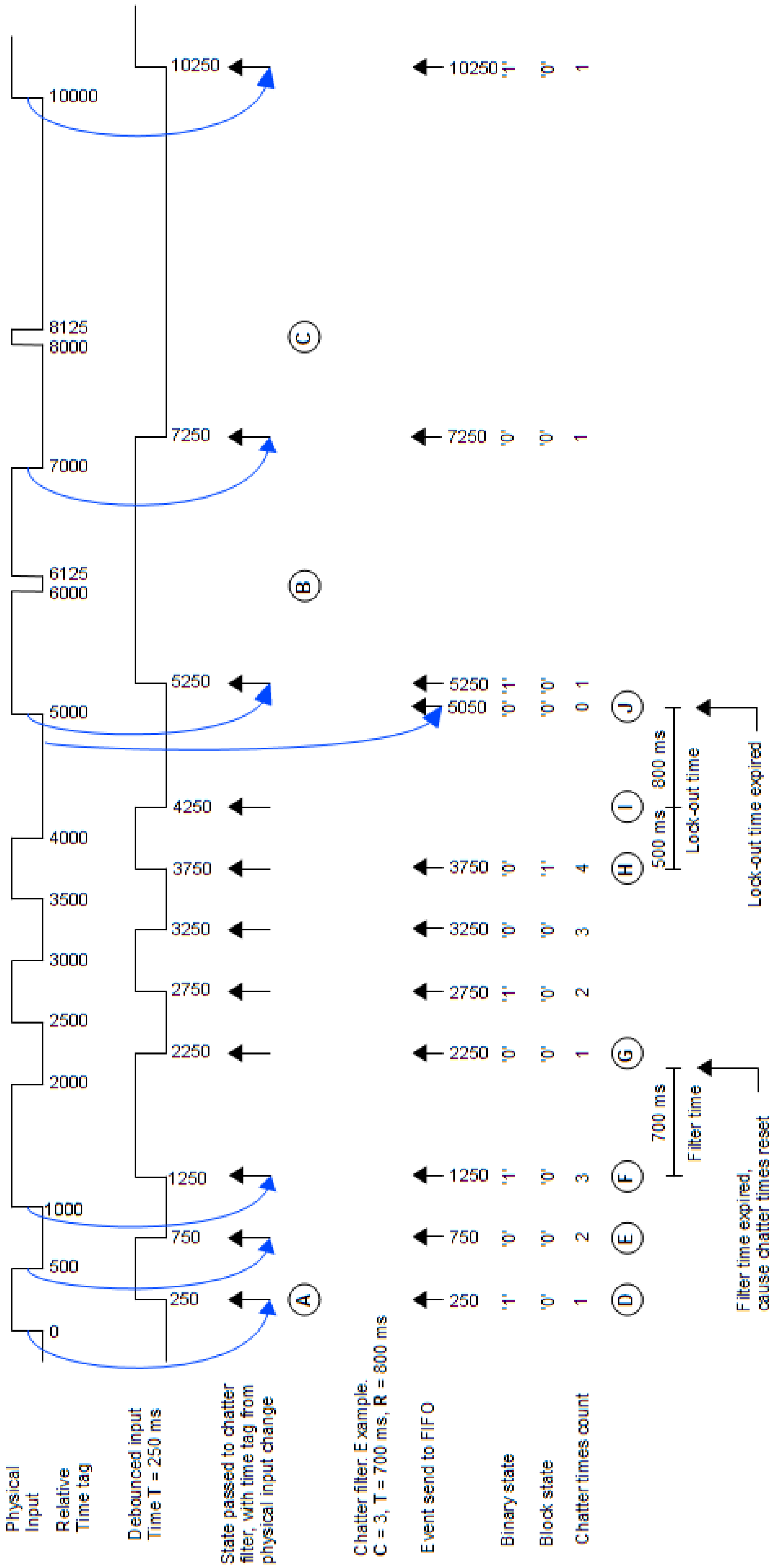
DI module 0 =
16 digital inputs

DI module 1 = 16 digital inputs
ZI module 0-31 = 16 32 bit counters
ZO module 0 = 16 counter resets

DO module 0 =
16 digital outputs

Appendix 4

Debounce and chatter filter figure



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

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

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

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

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

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

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

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

Point J **Input state is stable longer than 800 ms, which cause the lock-out timer to expire, and the chatter filter to turn OFF. A chatter filter generated event, with block state OFF and unchanged binary state, will enter the FIFO. Time stamp will be current time minus debounce time 'T', to obtain consistent time line. Further input state transitions will now pass the filter again.**