

RTU32S Series

Small RTU, PLC and Utility Controller

User Manual

Version 1.55-2 / July 2014 / Doc 40349





Table of Contents

1.	Customer Information	5
1.1	Copyright Notice	5
1.2	Trademark Acknowledgement	5
1.3	Disclaimer	5
1.4	Life Support Policy	5
1.5	Brodersen Customer Services	5
1.6	Technical Support	5
1.7	Product Warranty	6
2	Introduction	7
2.1	RTU32S – The small Brodersen RTU, PLC and Industrial Controller	7
2.2	WebServer configuration and PLC programming in all IEC61131 languages	7
2.3	RTU32S – Capabilities	7
3	Getting Started	8
3.1	Configuration and default settings	8
4	How to connect and prepare your PC to configure the RTU32S	8
4.1	Wiring	8
4.2	Setting up your PC Ethernet driver (Windows 7)	8
4.3	First time configuration	9
5	Configuration parameters in the web pages	11
5.1	Settings overview	11
5.2	Hardware Overview	11
5.3	Network settings	12
5.4	Modem Settings	13
5.5	VM Runtime Settings / Main Settings	14
5.5.1	Communication	14
5.5.2	Start-up	15
5.5.3	Options	15
5.6	I/O Board Settings	15
5.6.1	Current I/O Configuration	16
5.6.2	Additional I/O board configurations	16
5.7	Time Settings for RTC – Real Time Clock	16
5.7.1	Time Settings	16
5.7.2	Base Station time zone	17
5.8	Change password	17
5.9	Edit Config File	18
5.10	Boot	18
5.11	HMI	18
6	Getting Started with Brodersen WorkSuite tools	19
6.1	Introduction	19
6.2	Installing Brodersen WorkSuite Package	19
6.3	WorkSuite License	19
6.4	Compatibility with STRATON WorkBench	19
6.5	User documentation and training/guideline videos	19
7	Relevant RTU32 Series specific functions in WorkSuite	21
7.1	General	21
7.2	WorkSuite I/O Drivers for RTU32S	21
7.3	I/O Database Access	21
7.4	I/O database layout	21
7.5	Addressing example	22
7.6	Layout for the status register (ST)	22
7.7	WorkSuite I/O Board Driver	23
7.8	WorkSuite I/O Profiles Driver	24
7.8.1	Option 1 - Declaring I/O Profiles in variable list	24
7.8.2	Option 2 - Declaring I/O Profiles using the Profile Editor	25
7.9	Counter inputs	26
7.10	COM Port Settings	26
7.11	Modbus Drivers	27
7.11.1	ModbusRTU Master	27
7.11.2	ModbusRTU Slave	27



- 7.12 IEC60870-5-101/103/104 Drivers.....27
- 7.13 Watchdog.....27
- 7.14 Real time / Real time Clock28
- 7.15 Modem Functions for UCM-94 3G/GPRS Modem Series28
 - 7.15.1 Dial functions for GSM and PSTN Modems28
 - 7.15.2 Functions for control and monitoring the UCM-94 3G/GPRS Modem32
- 7.16 Data Logging.....34
- 7.17 Status information functions.....35
 - 7.17.1 Interface Board Status Info (STATBOARDINFO).....36
 - 7.17.2 RTU Internal Temperature (STATINTTEMP).....36
 - 7.17.3 LocalBus Status (STATLOCALBUS).....36
 - 7.17.4 RTU memory Status (STATMEMORY)37
- 7.18 Reading Text Files from WorkSuite38
- 8 Not supported features and PLC Functions in the RTU32S.....39
 - 8.1 List of WorkSuite Functions not presently supported by the RTU32S39
- 9 Distributed Event Binding Protocol / Redundancy Binding Communication40
 - 9.1 General.....40
 - 9.2 Network settings in RTU32 (redundancy binding)40
 - 9.3 Procedure for setting up binding/redundancy binding in WorkSuite40
- 10 Optional SNMP Extension Agent Driver41
- 11 RTU32S cycle / scan time41
 - 11.1 General.....41
 - 11.2 RTU32S Scan cycle mechanisms42
 - 11.3 RTU scan time42
 - 11.4 LocalBus scan time43
 - 11.5 Internal I/O Bus scan time43
- 12 RTU HMI.....44
 - 12.1 General.....44
- 13 RTU32 Utilities and System Event Log.....44
 - 13.1 General.....44
 - 13.2 RTU32 Software Bulk Update Tool45
 - 13.3 RTU32 Search Tool45
 - 13.4 System Log/Event Viewer.....45
 - 13.5 Remote Display / Remote Desktop Tool46
- 14 RTU32S Technical Description.....48
 - 14.1 General.....48
- 15 Hardware block diagrams48
- 10.3 Software block diagrams49
 - 15.1 VM PLC Runtime.....49
 - 15.2 RTU32 LocalBus for I/O Expansion49
 - 15.2.1 General.....49
 - 15.2.2 LocalBus (LB) driver50
 - 15.2.3 LB driver event information50
 - 15.3 RTU32S File System51
 - 15.3.1 General.....51
 - 15.3.2 Relevant folders to review and use51
 - 15.4 Software Versions52
- 16 History.....53
 - 16.1 New features since last update53
 - 16.2 Important notes about the latest versions.....53
- Appendix 1.....54
 - RTU32S runtime application measurement / benchmark.....54
 - Example 1: RTU32S runtime application program without any communication54
 - Example 2: RTU32S application program with ModbusTCP communication54
 - Comments to the bench mark tests and recommendations55
- Appendix 2.....56
 - RTU32 I/O addressing example 156
- Appendix 3.....57
 - RTU32 I/O addressing example 257
- Appendix 4.....58
 - RTU32S PLC Runtime application RESET58
- Appendix 5.....59



RTU32S complete Reset	59
Appendix 6.....	60
RTU32S firmware update	60



1. Customer Information

1.1 Copyright Notice

Copyright 2014, Brodersen A/S, ALL RIGHTS RESERVED.

No part of this document may be reproduced, copied, translated, or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the prior written permission of the original manufacturer.

1.2 Trademark Acknowledgement

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

1.3 Disclaimer

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

Brodersen A/S makes no representation or warranty that such application will be suitable for the specified use without further testing or modification.

1.4 Life Support Policy

BRODERSEN A/S's PRODUCTS ARE NOT FOR USE AS CRITICAL COMPONENTS, IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE PRIOR WRITTEN APPROVAL OF BRODERSEN A/S.

As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into body, or (b) support or sustain life and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in significant injury to the user.

2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

1.5 Brodersen Customer Services

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

1.6 Technical Support

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

To receive the latest version of the user manual, please visit our Web site at:

<http://www.brodersen.com>,

Choose the product in question under product search and under each product you will find accompanying data sheets, manuals, and user guides etc.

If you still cannot find the answer, gather all the information or questions that apply to your problem, and with the product close at hand, call your dealer. Our distributors are well trained and ready to give you the support you need to get the most from your Brodersen products. In fact, most problems reported are minor and are able to be easily solved over the phone.

In addition, technical support is available from Brodersen engineers every business day. We are always ready to give advice on application requirements or specific information on the installation and operation of any of our products. Please do not hesitate to call or e-mail us on support@brodersen.com.

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

Australia:
Brodersen Systems Asia Pacific P/L
832 High Street, Kew East
VIC 3012, Australia
Tel: +61 3 9249 9505
sales-ap@brodersen.com



1.7 Product Warranty

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

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

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



2 Introduction

2.1 RTU32S – The small Brodersen RTU, PLC and Industrial Controller

The Brodersen RTU32S series is based on a 32-bit ARM platform, providing a small footprint RTU and PLC with leading edge functionality. As well as being a versatile RTU and PLC with flexible I/O designed to perform embedded data processing, control, data logging and monitoring, it is also a networking communicator for collecting, managing and communicating data via protocols on different physical interfaces upwards and downwards in an industrial environment.

Typical applications for RTU32S are:

- Control and monitoring of small water pump stations
- 10kV substation automation applications
- Railway level crossing monitoring
- Communication Gateway applications between Modbus drivers, DNP3 and IEC60870.

The RTU32S hardware platform includes a 200MHz CPU with 64MB RAM and integrated 128MB NAND Flash. In addition, a removable SD Flash card (default size of 128MB), provides an easy to use portable access/upgrade option. The integrated Flash stores the main RTU32 firmware including OS and virtual machine for executing your PLC runtime application. Standard drivers like the Modbus Protocol Suite, IEC60870 and the RTU-to-RTU Distributed driver are stored in the integrated Flash. The removable SD Flash card includes your compiled runtime application as flat files and any optional drivers. The SD Flash card is also used for storing retain variables and data logging files.

2.2 WebServer configuration and PLC programming in all IEC61131 languages

The RTU32S, like the other RTU32 Series of RTUs is configured via the integrated WebServer. Settings like LAN port addressing, time functions and 3g modem credentials are handled in webpages. Online monitoring of connected physical I/O and display of graphical overviews is possible via webpages.

Programming of logic functions and configuration of communications drivers is managed using the Brodersen WorkSuite package software. WorkSuite is a powerful tool that is used for configuration and programming of all RTU32 Series products. Applications developed and running in the RTU32S will easily run directly on the larger RTU32 Series products. The full suite WorkSuite software package is included in the product package, and it will run in demo mode without any license (with some runtime restrictions).

2.3 RTU32S – Capabilities

The RTU32S can handle most of the functionality provided in the Brodersen WorkSuite Package, but to ensure reliable operation of the RTU32S, some restrictions are imposed, including limiting the number of project variables, drivers and driver tags/data points. These limits include;

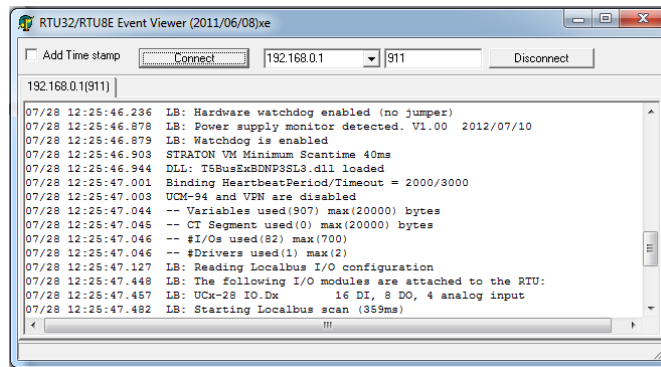
1. Additional drivers in the RTU32 are limited to two drivers (physical I/O, Modbus and IEC60870 drivers suite are not counted).
2. The number of I/O points associated to drivers is limited to 700 (IEC60870 driver data points excluded). All data types are counted as 1 point (eg. Boolean, Real, DWord).
3. Memory for plain/internal variables and complex variables is also limited. There is a limit of 20kbytes of memory for all plain/internal variable types in a RTU32S PLC project (e.g. Boolean, Bytes, Words that are not associated to a driver interface). These variables are counted based on their expected size, except for Strings that are always counted as 25bytes (regardless of String size defined/used). Complex variables like Arrays and Structures also have a limit of 20kbytes of memory.

The limitations are defined in the Virtual Machine of the RTU32S. If you try to start a runtime project in the RTU32S that exceeds these limits, the project will NOT start and an info message in the Runtime info window in WorkSuite will indicate the status of the limits in your project. Every time you start a runtime project the current status will be shown in the Runtime message window

#	Time	Event description
1	2014/07/28 12:22:17.351	-- Variables used(907) max(20000) bytes
2	2014/07/28 12:22:17.353	-- CT Segment used(0) max(20000) bytes
3	2014/07/28 12:22:17.356	-- #I/Os used(82) max(700)
4	2014/07/28 12:22:17.356	-- #Drivers used(1) max(2)



and in the SystemLog of the RTU32S:



If you reach the I/O or memory limits of the RTU32S, you can take advantage of the capability of the other RTU32 Series RTUs that will execute far larger projects and support more advanced functions and drivers than supported by RTU32S. Any runtime project/program running in the RTU32S – can directly and without ANY changes be downloaded and executed on larger RTU32 Series products. Contact us or our local partner for more details.

3 Getting Started

3.1 Configuration and default settings

The LAN interface on the RTU32S is set to its factory default settings at delivery:

LAN1: 192.168.0.1, Subnet: 255.255.255.0

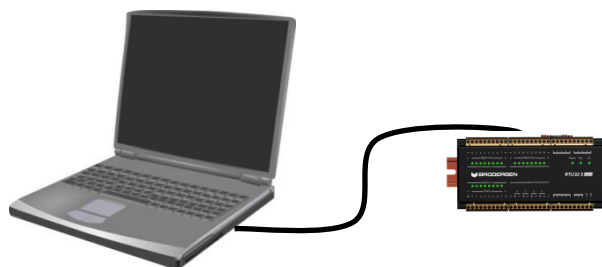
You can use the LAN1 interface to configure the basic settings in the RTU32S. The basic settings include setup of network parameters such as IP address, Subnet mask, default gateway, time settings, general username and password for accessing the web server pages and file system with FTP etc.

The configuration is done via web pages in the RTU32S Web Server. You must use your normal web browser to setup the basic settings for the RTU32S. 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 RTU32S

4.1 Wiring

Use a cross-wired or standard patch cable to connect your PC Ethernet interface to the LAN1 Ethernet interface of the RTU. Apply power to the RTU – make sure that you connect the correct power supply voltage.

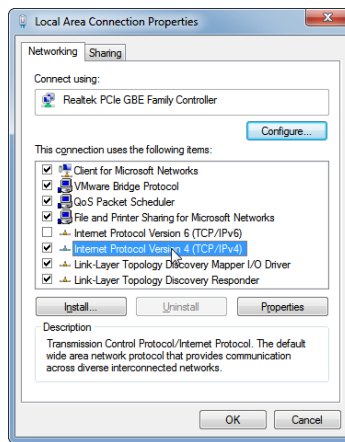


4.2 Setting up your PC Ethernet driver (Windows 7)

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.

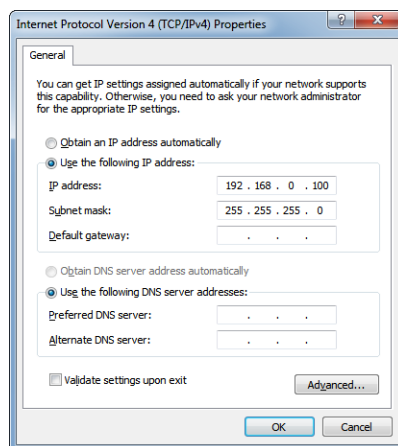


To setup the Ethernet driver select *start – Settings – Network Connections – Local Area Connections* and the following window will appear:



Click on Internet Protocol 4 (TCP/IP4), 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 RTU32S.

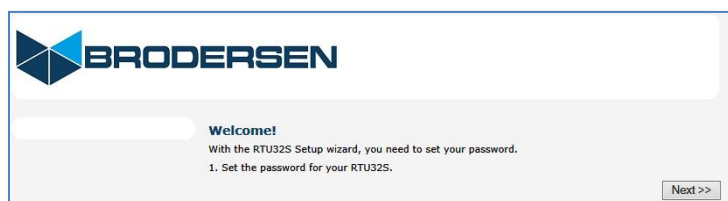


Click *OK* and *OK* again at the Local Area Connection Properties window. Now your PC Ethernet driver is setup for communicating with the RTU32S. Now apply power to the RTU32S. Wait for the RTU32S to boot up – takes approx. 20s. As a minimum the green Power LED on the front PANEL must be lit.

4.3 First time configuration

Start your Internet browser and enter the IP address of the RTU32S 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.



Click *NEXT* to continue.



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

1. Set the password for your RTU32S

Set the password for your RTU32S

Type password:

Re-type password to confirm:

<< Prev Finish

After typing in your password twice, select Finish to continue and save the settings.

The RTU will then restart and after ~30s you will get asked to login to the RTU32S web server as shown below.

Windows Security

Connecting to 192.168.0.1.

User name

Password

Domain: BSD

Remember my credentials

OK Cancel

The 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 RTU32S 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 RTU32S Web pages is the Settings Overview page:

Settings Overview

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	NUC900
Obtain an IP address via DHCP	DISABLED
Local IP Address:	192.168.1.221
Subnet mask	255.255.255.0
Default gateway:	192.168.1.1
Preferred DNS server	83.136.89.6
Alternate DNS server	192.168.1.100
MAC address	00-04-5F-93-49-b4

Host name RTU32S

Project Information

Project Name	RTU32SIOTest
Project Runtime Start	2014/07/08 10:32:51.912
Project Build Time	2014/07/08 08:33:53
Project Build Version	7
Project Build CRC	16#67023e9a

RTU32S Version Information 1.55.1.0/1.18.10B1

RTU32S Firmware Date (RTU32S.exe)	2014/07/04
RTU32S Firmware ID (RTU32S.exe)	92746 W910
VM Runtime Version (RTU32.exe)	8.5.121210
RTU32S CE Image Date (NK.bin)	2014/07/03
RTU32S Windows CE Version:	6.0
RTU32S Windows CE Build Version:	1937

Drivers installed

IEC60870-5-101 Master/Slave	RTU32.exe	1.55.1.0
IEC60870-5-103 Master	RTU32.exe	1.55.1.0
IEC60870-5-104 Client/Server	RTU32.exe	1.55.1.0
Modbus Master/Slave	RTU32.exe	1.55.1.0
DNP3 Slave	TSBusExBDNP3SL3.dll	3.1.52.482

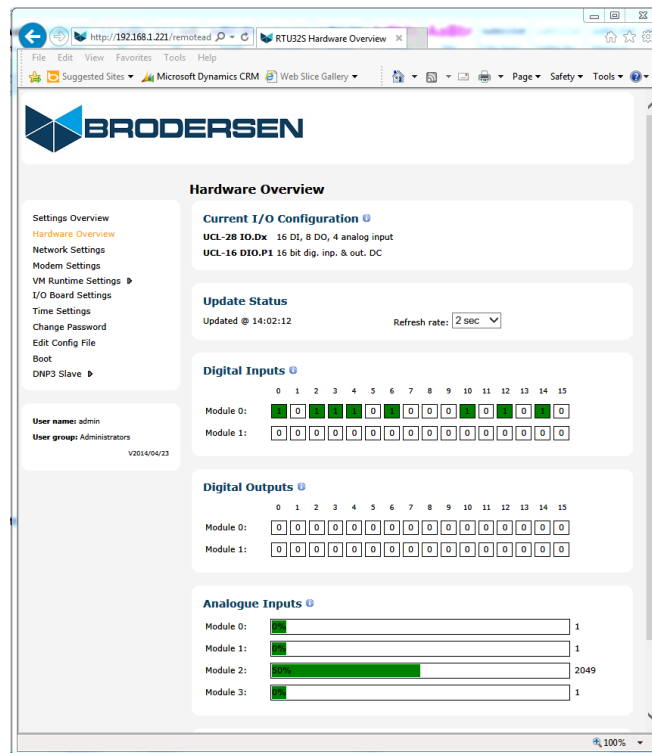
Copyright © 2014 Brodersen A/S

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

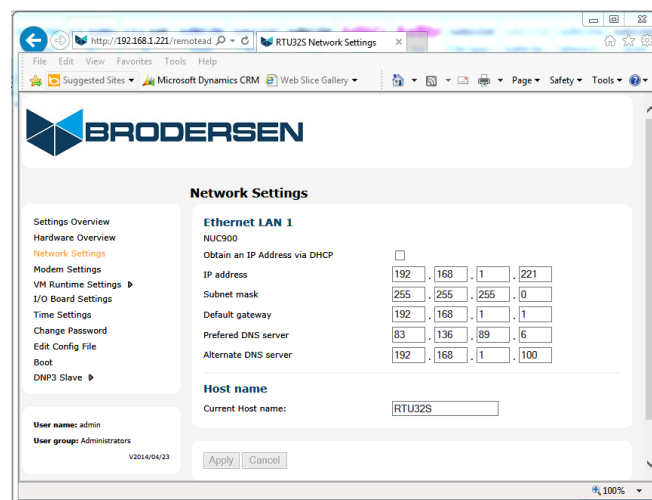
The Drivers installed section lists the installed drivers in your RTU32S. The full Modbus Suite and IEC60870 Drivers are installed in all RTU32S. Additional drivers like DNP3 Slave are installed if the Driver License is ordered with the RTU32S.

5.2 Hardware Overview

The hardware overview page shows you online the current I/O configuration status. 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 RTU32S is equipped with internal I/O and input 0, 2, 3, 4 and 6 are activated.



5.3 Network settings

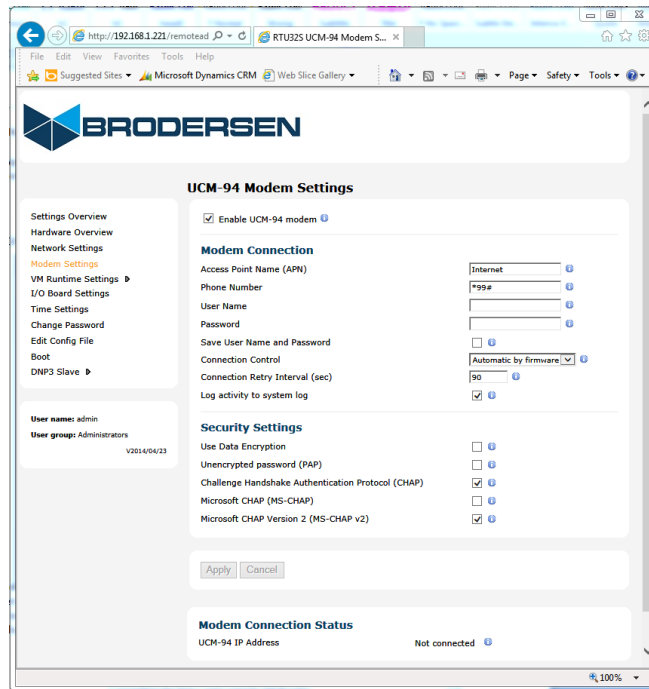


On the network setting page you can change the LAN1 settings to fit your local network. You must assign fixed IP addresses to gain access to the RTU32S with your browser, FTP client or Brodersen WorkSuite in your LAN network. If you do not know about the IP addresses and Subnet masks in your local network, 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 RTU32S.

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.

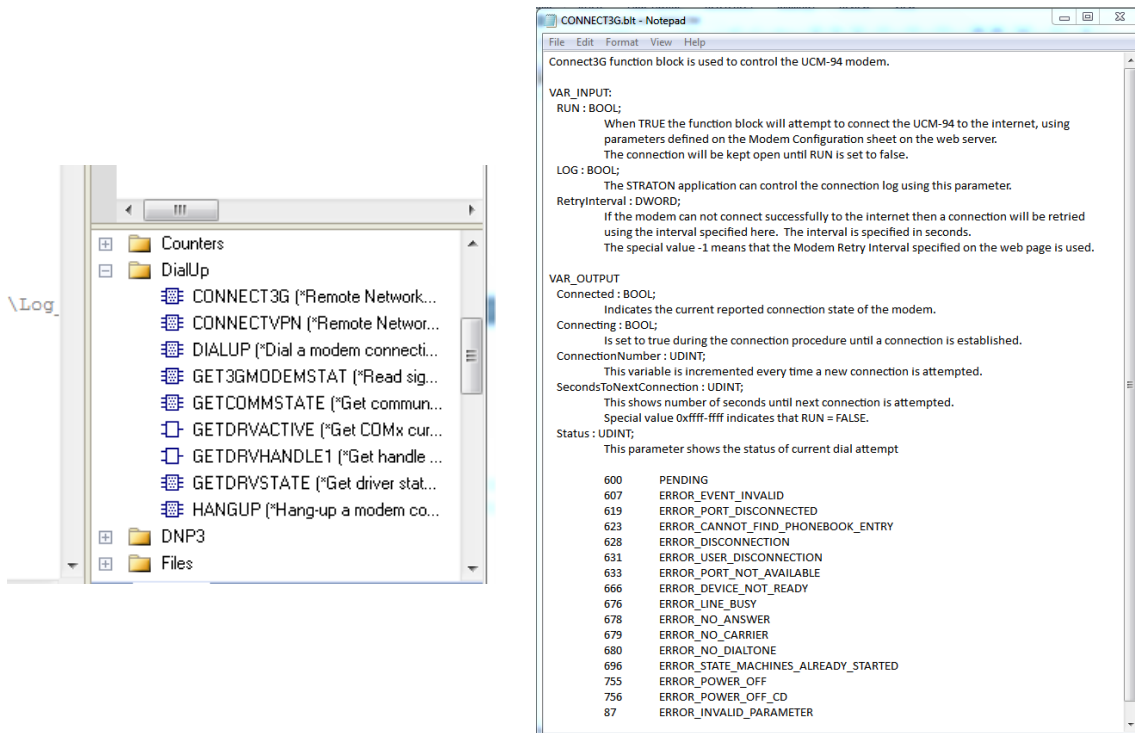
5.4 Modem Settings

If you use the Brodersen UCM-94 3G/GPRS modem you must configure the RTU32S to work with the modem.



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).

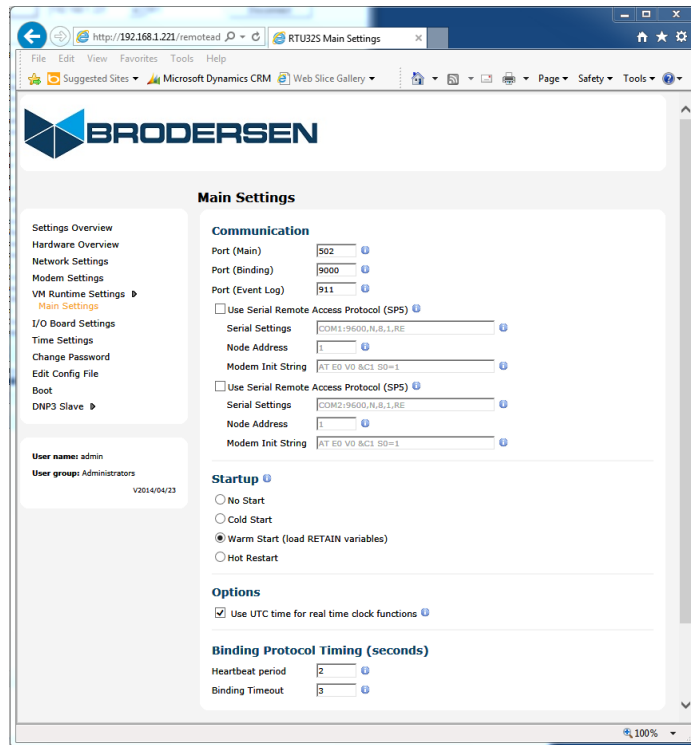


Note: The Settings Overview web page will include the modem connection status at the bottom of the web page if the modem is connected. It will also show you the IP address assigned on the 3G/GPRS network.



5.5 VM Runtime Settings / Main Settings

The VM (Virtual Machine) is the name for the PLC application runtime executer.



5.5.1 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)

The 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)

The Event Log port is where the RTU32S 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. e.g. if you have the RTU32S placed on a remote site connected via serial communication using multi-drop radio or line connection – or even dial connection, you are able to remotely connect to the RTU32S with the Brodersen WorkSuite and work online as if it was on your workshop table. The speed will however be slower depending on your actual serial communication speed.

It is possible to attach this protocol to two communication ports.

Serial settings

The syntax is: [M]COM{n}:{b},{p},{d},{s}[h][,LL] and the parameters in brackets [] are optional:
[M] Use modem connection



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

The following example shows setting for communicating on COM1 port, at 19200 baud, with Even parity, 8 data bits and 1 stop bit:
COM1:19200,E,8,1

Here a modem usage is enabled: MCOM2:9600,N,8,1

Here logging is enabled: COM1:9600,N,8,1,LL

Node address

Node Address (or Unit number) identifies the RTU32 on a multi drop network. The default number for RTU32S on a peer to peer network is 1. The address must be a value from 1 to 32767.

Modem Init String

Here you enter the initialization string (AT commands) when using a modem.

RTU32S default is: AT E0 V0 &C1 S0=1

5.5.2 Start-up

The start-up option specifies how the PLC application is started when the RTU32S 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!

5.5.3 Options

Use UTC time for PLC real time functions

If enabled, it instructs 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 saving if this option is enabled.

NOTE: Drivers that are implemented via PLC Functions – like the IEC60870 Driver – will use the PLC time stamp for 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.

5.6 I/O Board Settings

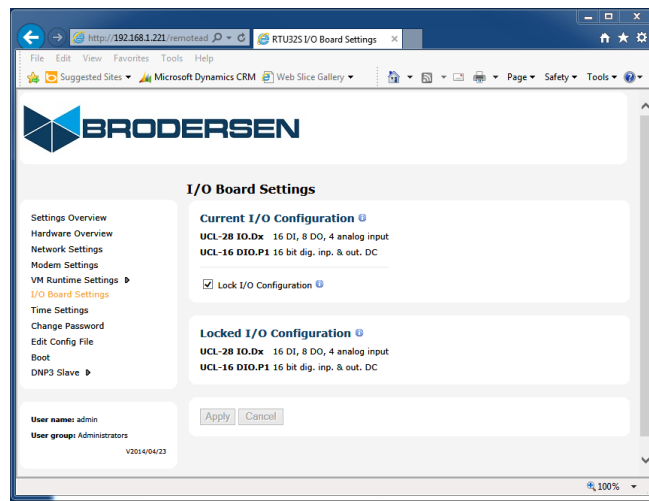
On this page you can:

- Read the actual I/O configuration on the RTU32.
- Lock an I/O configuration
- Adjust the settings for configurable inputs and outputs – if any in current configuration.



5.6.1 Current I/O Configuration

Here the actual I/O configurations are listed. In the example below you will see that the configuration of I/O boards include 28IO and a connected I/O Expansion module with 16DO/16DO.



By default the I/O configuration is unlocked. When unlocked the actual I/O configuration is read every time the RTU32S 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. Use the Lock function if you want to be able to trace a disconnected or defective I/O module in the RTU32 System Log.

5.6.2 Additional I/O board configurations

In future, some I/O Expansion modules are planned to have user configurable parameters. They will most likely be added here to adjust the setup for each configurable module added to the RTU32S configuration.

5.7 Time Settings for RTC – Real Time Clock

The time settings web page is used for configuring and adjusting the RTU32S real-time clock.

5.7.1 Time Settings

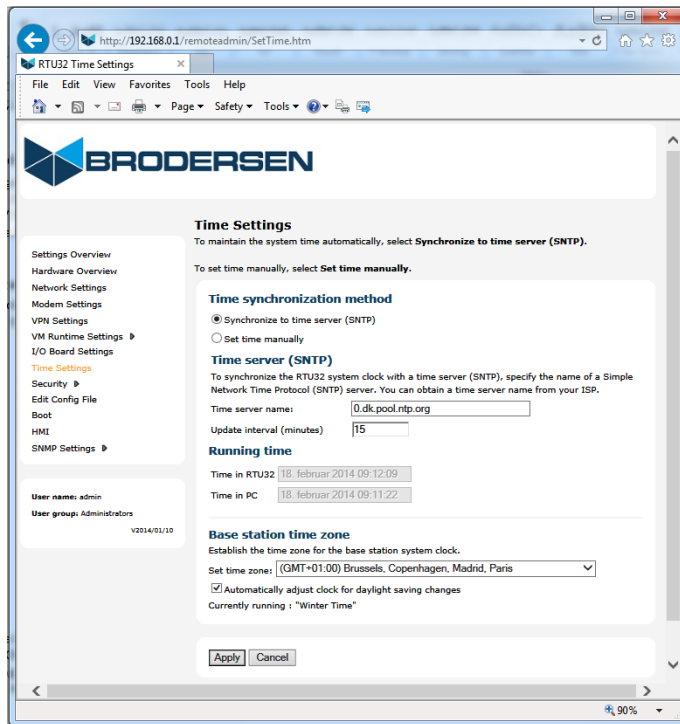
The RTU32S realtime clock can be configured from the Time Setting page.

You can define if the RTU should be synchronized via a SNTP Server, or you can use other synchronization options like manual time setting or synchronization via utility drivers that include this functionality, such as IEC60870 or DNP3.

NOTE: The time shown on the webpage is running in software and is only updated when the page is refreshed.

When set for synchronizing to a SNTP Server you must define the SNTP Server IP address or domain name. If you use a domain name, you must ensure that you have defined a DNS Server address in the LAN1 configuration.

Update interval time can be set from 2 to 20160 minutes.

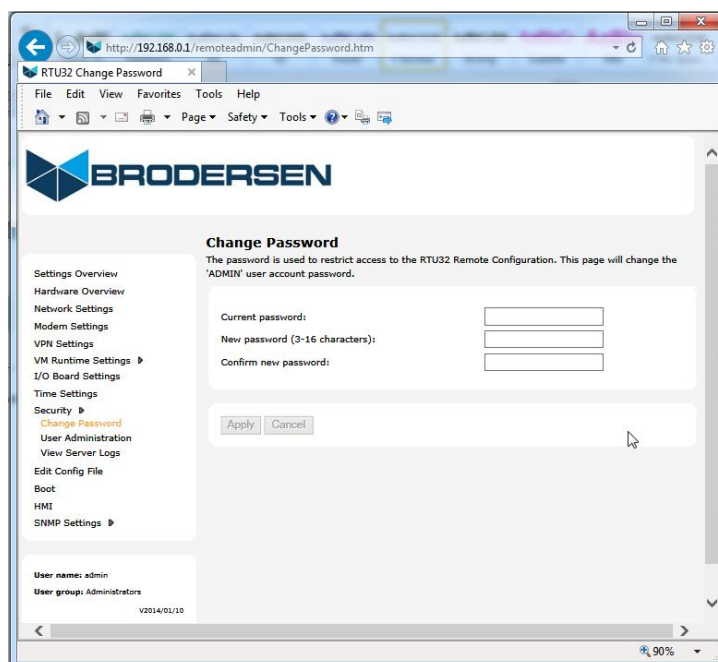


5.7.2 Base Station time zone

If you use local time you have to set the time zone correctly in order to get the correct time stamps. By 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 to consideration if you are working with IEC60870 drivers and use Clock Synchronisation.

5.8 Change password

Here you can change the password for your main login.



Note: the username and password covers access to all Servers in the RTU32S – including HTTP Server, Telnet Server and FTP Server

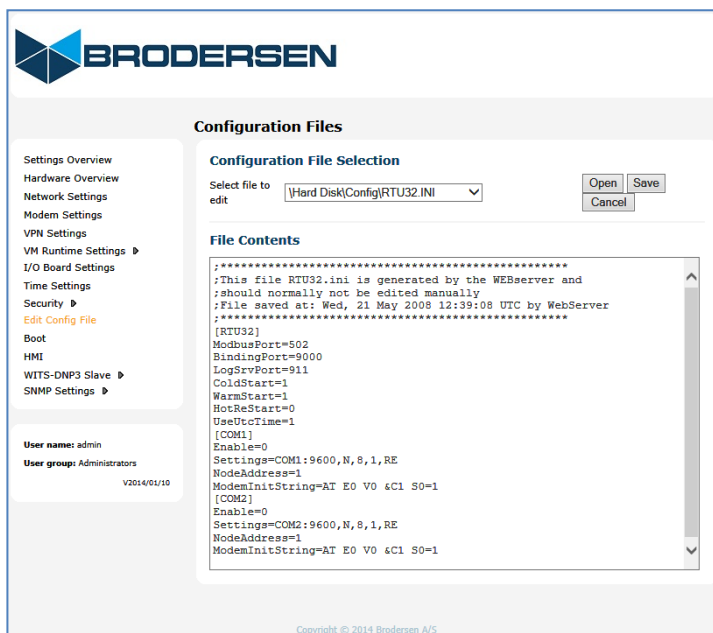


5.9 Edit Config File

The Edit Config File page is a simple text file editor used for making changes and adjustments in text based files in the RTU32S file system. The editor will automatically list all files with the extension “.txt”, “.dat” and “.ini” that are placed in the RTU ../Storage Card/Config/.. folder.

By 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 RTU32S.



You may add your own files to the ../Storage Card/Config/.. folder. You can create and place your own files using e.g. FTP to a RTU32S. The WorkSuite supports functions for reading textual files and can for example 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 RTU32S that include parameters used in a running PLC runtime application. Both the Brodersen WorkSuite and other developer tools provide options for read and write to text files.

You can also 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.

NOTE: For the normal user, the RTU32S system files need no changes – keep them unchanged. And do not edit the text files default placed in this folder, unless you are absolutely sure what impact it has.

5.10 Boot

After changing parameters on the configuration pages, you must unless otherwise noted, run the reboot function to activate the changes. After booting the RTU32S you can start working with Brodersen WorkSuite to create you application programs.

5.11 HMI

If you are an experienced user of other products in the Brodersen RTU32 Series you may miss the HMI link and HMI feature. This is presently not available in the first versions of RTU32S software, but it is planned to be added in a later firmware version of RTU32S software.



6 Getting Started with Brodersen WorkSuite tools

6.1 Introduction

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

The software components included in the package include:

- Brodersen WorkSuite PLC configuration and programming tool.
- WIBUKey driver for the USB license key dongle for WorkSuite.
- RTU32 Series tool package including:
 - o Event Viewer – the telnet tool used for monitoring the System Log(s) in RTU32 Series products.
 - o RTU32 Search tool for scanning and finding RTU32 products running on networks.
 - o RTU32 Series Remote Desktop tool
 - o 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 you to begin working with configuration and programming of the Brodersen RTU32 Series products.

6.2 Installing Brodersen WorkSuite Package

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

6.3 WorkSuite License

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

The license is locked to a USB hardware dongle 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 an unlimited number of I/O points in the RTU application program. All variables used for drivers are counted as I/Os.

6.4 Compatibility with STRATON WorkBench

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

The Brodersen WorkSuite is in fact a Brodersen OEM version of STRATON, but includes a range of new facilities:

1. One installation process for WorkSuite - installs all software including license key drivers and all drivers, functions and Brodersen Tools supported by the RTU32 Series products.
2. Eliminates the need to install special hardware libraries, tools etc. After the installation of the WorkSuite Package you are ready to go.
3. All drivers supported by the RTU32 Series of products are available in 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, they are 100% compatible with the WorkSuite. You can in other words use your license for the new Brodersen WorkSuite. That goes for ALL tag count licenses.
2. If you have an older STRATON license, please contact Brodersen at sales@brodersen.com or your local representative for update prices. Please remember to include a copy of your current License Certificate.

6.5 User documentation and training/guideline videos

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



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

We recommend that you use the Brodersen Training/Guideline Videos. They are found by visiting:

www.youtube.com/brodersensystems

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



7 Relevant RTU32 Series specific functions in WorkSuite

7.1 General

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

Besides the basic WorkSuite standard PLC programming environment and EN/IEC61131-3 programming languages, there are some functions and features which are specific for the RTU32S Product Series.

7.2 WorkSuite I/O Drivers for RTU32S

The WorkSuite I/O Drivers for the RTU32S are used to access the physical inputs and outputs on the RTU32S and the connected I/O expansion modules. All RTU inputs are read online when the PLC runtime cycle is started and all outputs are written online at the end of the runtime cycle.

NOTE: The drivers described in this section are directly online I/O.

All I/Os (including status flags/words) are stored in an independent memory mapped database in the RTU32S. The database is a mirror of the physical I/O. The Internal I/O Bus and the LocalBus for expansion modules are responsible for keeping this updated at any time. The WorkSuite I/O drivers handle the access to this database. You can program your logic application without any connected hardware. 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.

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 the RTU32S is implemented to support two methods of accessing the I/O database in the RTU32S. The two methods are called I/O Board and I/O Profiles.

I/O Board is used for declaring I/O variables based on the physical I/O layout. You are able to add for example 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 are more flexible and easy to use and allow declaring of a single input or output and include additional scaling possibilities for analogue I/O. Profiles are also used when you want to read status information from the RTU32S.

NOTE: Inputs can be declared by both I/O Board and I/O Profiles, but outputs 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 I/O addressing is structured. The database has basically 10 data types:

ST:	Status word for I/O
DI:	Digital inputs
AI:	Analogue inputs
VI:	Virtual inputs
ZI:	Auxiliary inputs (includes CI – 32bit counter inputs)
DO:	Digital outputs
AO:	Analogue outputs
YO:	Virtual outputs



ZO: Auxiliary outputs (includes 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_08DI	1						UCL-08DI.Ax
RTU32_08DO			1				UCL-08DO.Rx
RTU32_8DIO	1		1				UCL-8DIO.Px
RTU32_16DI	1						UCL-16DI.Dx
RTU32_16DIO	1		1				UCL-16DIO.Px
RTU32_32DI	2						UCL-32DI.Dx
RTU32_32DO			2				UCL-32DO.xx
RTU32_36IO	2			1			UCL-36IO.Px
RTU32_28IO	1	4	1		2X16	1 (RC)	UCx-28IO

7.5 Addressing example

To give a practical example, 2 different RTU32S 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 PLC runtime accessed database module number is not present.
 - Bit 3 IO configuration mismatch
 - Bit 4...15 Reserved
- Module 1: Reserved.

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

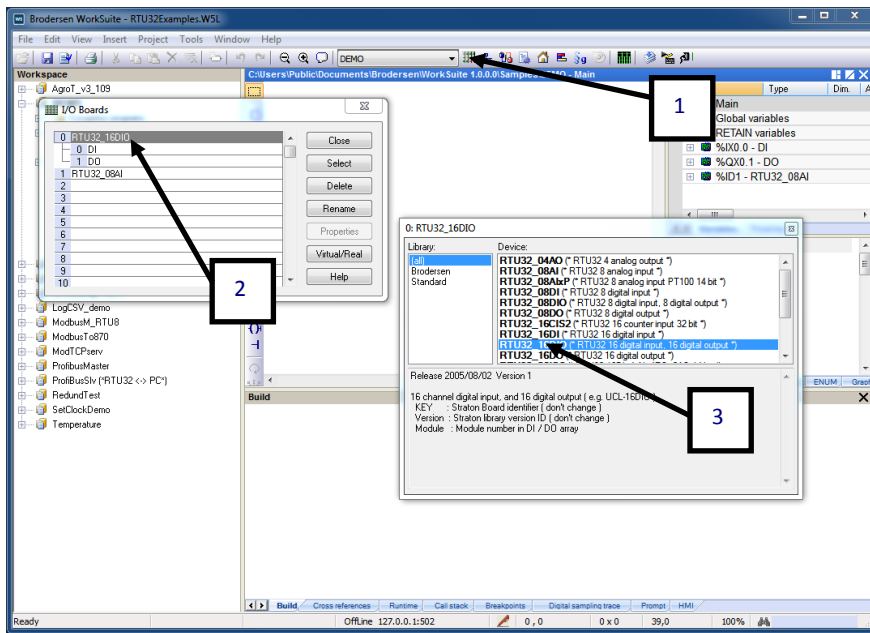


7.7 WorkSuite I/O Board Driver

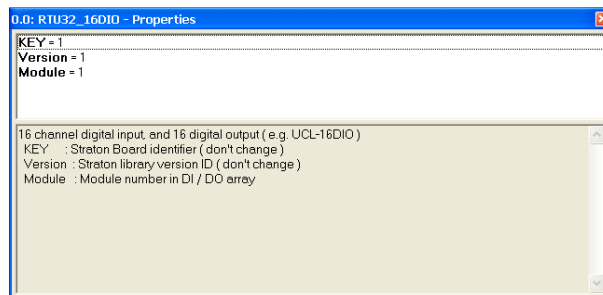
The I/O Board driver makes it possible to declare I/O in sections as per the physical board layouts. I.e. it is possible to add all 32 digital inputs in one process using the I/O Boards driver. At declaration it automatically declares the variables and puts them in the variable list.

Procedure for declaring board I/O:

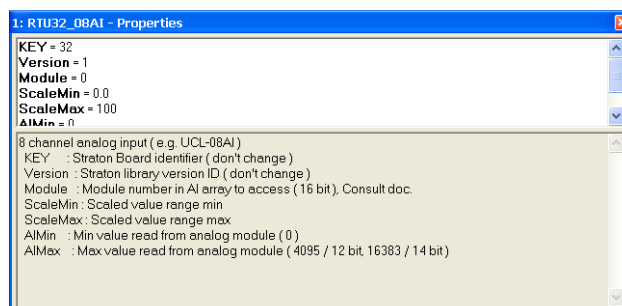
- Select Open I/O Board Configurator (1)
- Double click on the section 0 to add your first board (2)
- From the list of I/O board select the one you want (3)



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 has different types of I/O they will be shown individually in their own line. Every data type requires configuration.



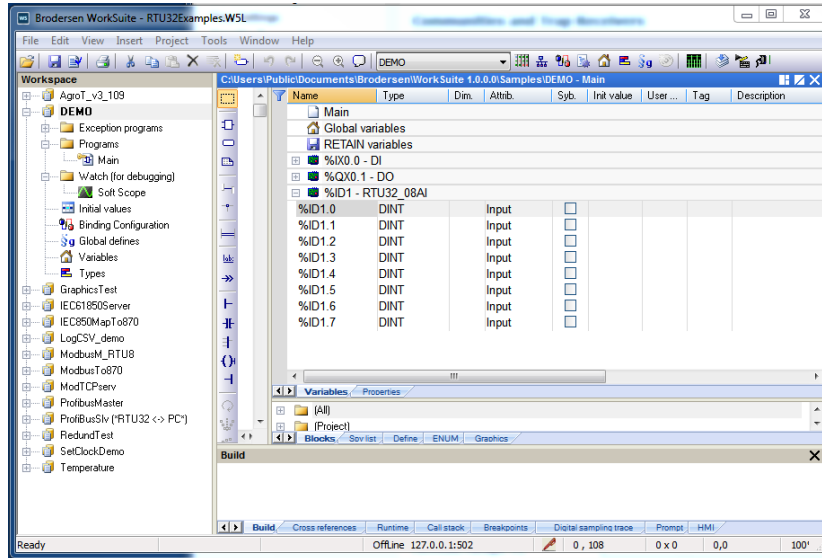
The properties for digital I/O are simple. Only the Module address needs to be setup. Use the addressing example to determine the Module address in your application.





The properties for analogue I/O require configuration. Besides the Module address of the first analogue channel, you can configure the scaling of the analogue I/O. Please note that the scaling will cover all I/O in the actual board section. On some board I/O it is possible to set the range of every input and output individually (for Internal I/O and newer analogue expansion modules).

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



The I/O is now ready to be used in your runtime application program.

For some of the analogue I/O you will find some qualifier inputs which are used for status information. They report underflow and overflow of the analogue value.

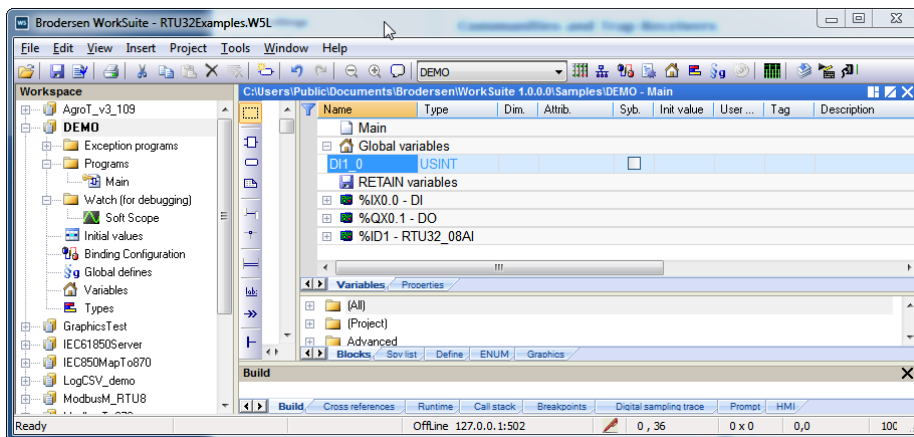
7.8 WorkSuite I/O Profiles Driver

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

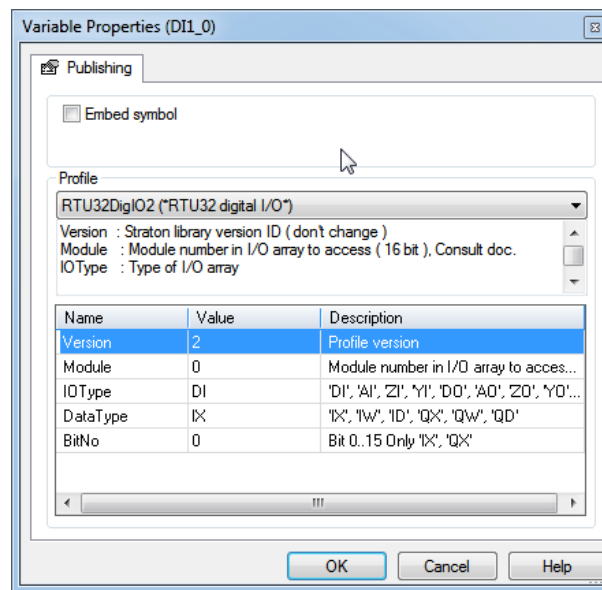
7.8.1 Option 1 - Declaring I/O Profiles in variable list

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

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



Right click on the DI1_0 variable and select Properties:



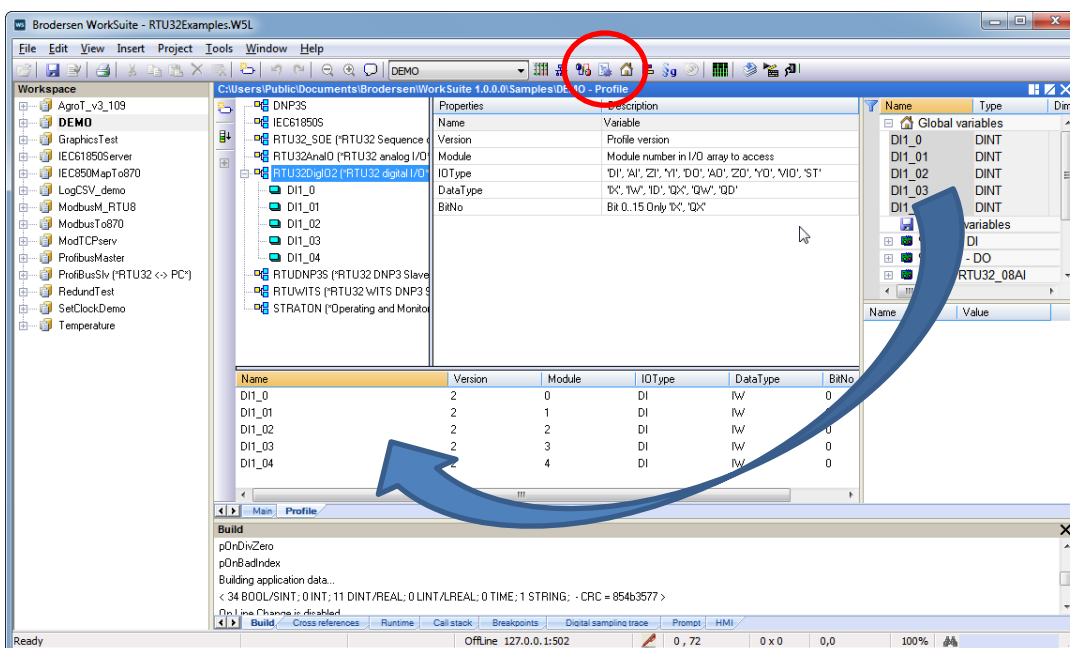
Select "RTU32DigIO" for digital I/O, set Module = 1 (second digital input module (read: Word)).
Select "DI" for Digital Input register.
Select "IW" for Input Word.
Say OK to store the settings.

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

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.





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

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 held ON 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 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 RTU32S.

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

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

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

- Setup the COM port for modem dial support – enable the Modem driver.
- Hardware handshake signal on null modem driver.
- Provide communication log in the general RTU32S System Log.

The extensions for these functions and features are:

- When using the modem driver a 'M' is added (e.g. MCOM1:9600,N,8,1)
- RTS / CTS hardware handshake control on the null modem serial driver is supported and enabled by the options as follows:
 - 'RD' RTS is kept inactive (low) at all time.
 - 'RE' RTS is kept active (high) at all time.
 - 'RT:Leading:Trailing' RTS is inactive when receiving data, and becomes active when transmitting data. The RTS Leading setting defines the delay from activating the RTS to when the first character is transmitted. The RTS Trailing setting defines the delay from when the last character is transmitted to RTS is deactivated. Leading and trailing time is in units of msec. - e.g. COM1:9600,N,8,1,RT:50:10
 - 'RC:Leading:Trailing' RTS is inactive when receiving data, and is activated when the RTU wants to transmit data. After activating RTS, the RTU will wait for CTS to become active, before start transmitting. The RTS Leading delay is still valid in this mode, and an adjustable delay from CTS activated to first character is then possible. However, by setting the Leading time to zero there is no unnecessary delay from CTS to first character (like a normal RTS / CTS function). After activating RTS the RTU waits up to 10 sec for the CTS signal. If timeout occurs, the transmission is discarded. The RTS Trailing setting defines the delay from when the last character is transmitted to RTS deactivated.
- Logging of serial data in the System Log.
 - 'LL' Log Link data. All bytes transmitted and received on the COM port, are logged in the internal log message system, and are available using telnet at port 911. This is primarily for debug purpose.
Examples: MCOM3:9600,N,8,1,LL
 COM2:57600,E,8,1,RT:30:10,LL

Note! If you just connect a serial device, e.g. ModbusRTU communication via a Null Modem cable to the RS232 COM port, it is NOT required to add any special extensions.

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. If the SP5 setting is enabled, the COM settings overrule any later settings in WorkSuite.

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



7.11 Modbus Drivers

The RTU32S PLC runtime supports function for several types of Modbus protocols:

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

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

7.11.1 ModbusRTU Master

The RTU32S supports several Master/Client drivers. You can in fact setup as many Master drivers as you have COM ports available. The Modbus Master/Client is setup using the Fieldbus Configurator. See the WorkSuite help for details.

7.11.2 ModbusRTU Slave

The RTU32S supports only one Slave/Server driver. You must add a MBSLAVERTU function block in the program logic for the Serial Modbus Slave driver (MBSLAVERTUEX function is NOT supported – supported in other RTU32 Series products). The MBSLAVERTU function defines the Modbus Slave address and port settings. You can find the function block in the “Advanced” function block folder. All variable assignments, addressing etc. are done in the Fieldbus Configurator. See WorkSuite for details.

7.12 IEC60870-5-101/103/104 Drivers

The Utility protocols IEC60870-5-101, 103 and -104 are implemented as a basic link driver in the RTU32S. The application layer which includes all the specific handling of request and events communication is managed in the RTU application program. A range of WorkSuite functions and example are provided for the different protocol functions and ASDU handling.

WorkSuite include drivers for:

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

The basic link drivers 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, provides the option to implement application layer support for this protocol as well.

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

7.13 Watchdog

The RTU32S has two built-in watchdog functions – all managed by a separate CPU on the interface board.

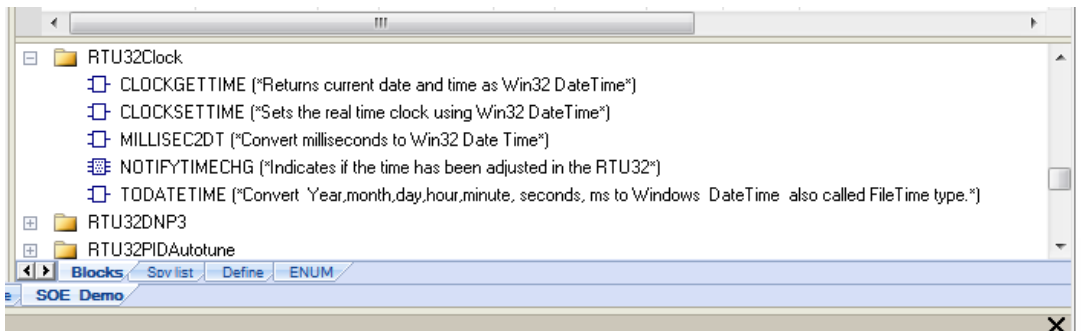
The system watchdog is monitoring all internal processes including the OS and communication processes between consoles and databases. In case of error the watchdog will wait 30sec for the error to be either self-healed or the process to be restored. If this is not happening the watchdog will boot the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

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



7.14 Real time / Real time Clock

The RTU32S has a hardware real time clock controlled by the operating system. It provides the RTU32S millisecond resolution in order to make accurate time stamps. From the RTU 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.



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

7.15 Modem Functions for UCM-94 3G/GPRS Modem Series

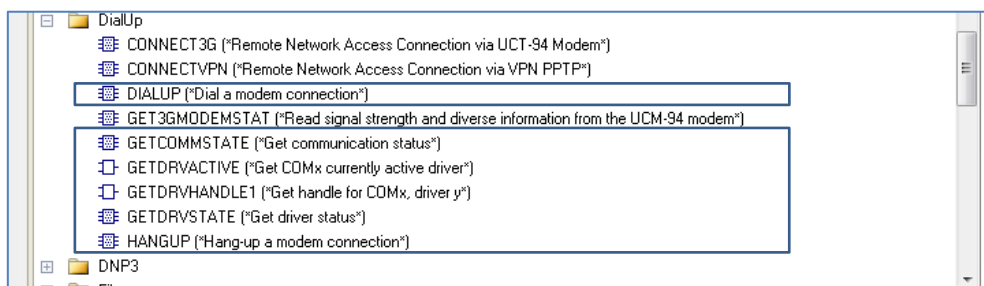
WorkSuite provides a range of modem control Functions. The DialUp library includes:

- Functions to control and monitor a UCM-94 3G/GPRS modem.
- Functions to control and monitor A PSTN/GSM modem dial-in/out

7.15.1 Dial functions for GSM and PSTN Modems

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

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

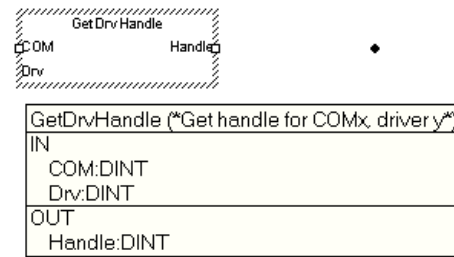
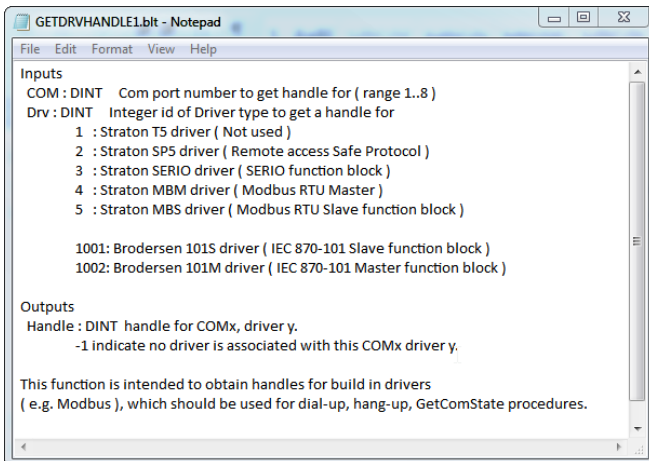


The modem function is enabled by adding M to the COM settings. See details in COM settings section.

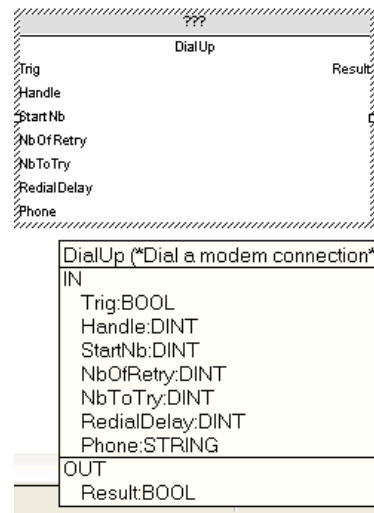
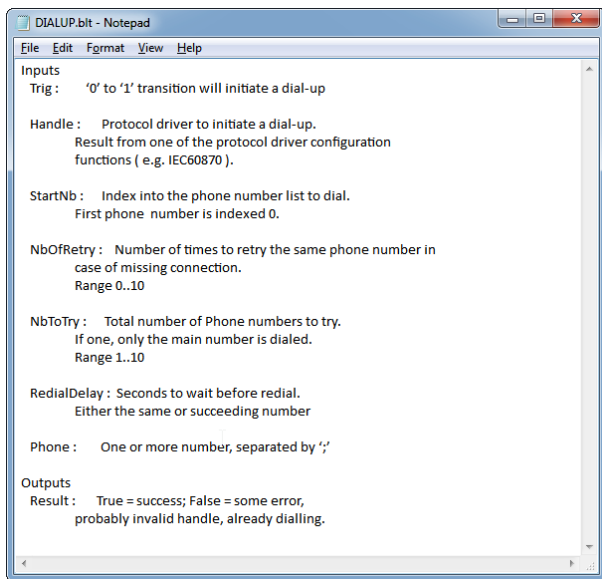
After enabling the modem dial driver, you must add a Modem driver handle in your application program (GetDrvHandle). Your dialup function must now refer to this handle. The DialUp function includes all the necessary parameters to establish and control a dial-up or dial-in session.

GetDrvHandle (Get driver handle).

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

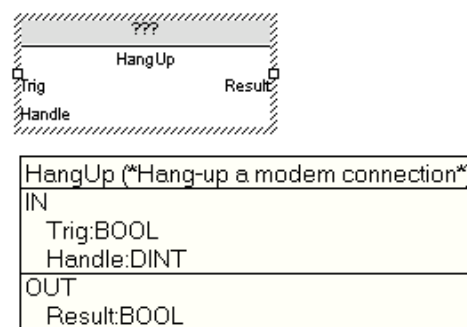
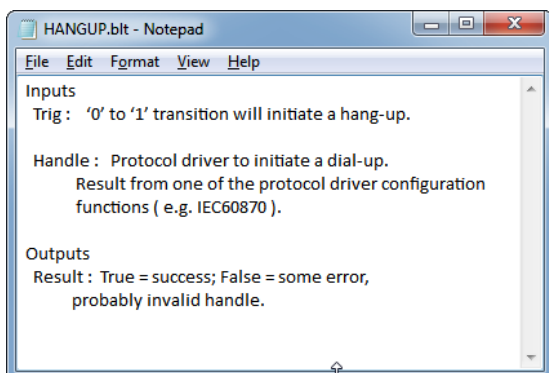


Dialup - Start a dial-up connection for one of the protocol drivers.



NOTE: One instance of this function block must be declared for each modem driver (physical modem) used.

HangUp - Hang-up current connection



NOTE: One instance of this function block must be declared for each modem driver (physical modem) used.

**GetCommState - Get communication status.**

```
GETCOMMSTATE.bit - Notepad
File Edit Format View Help
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

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

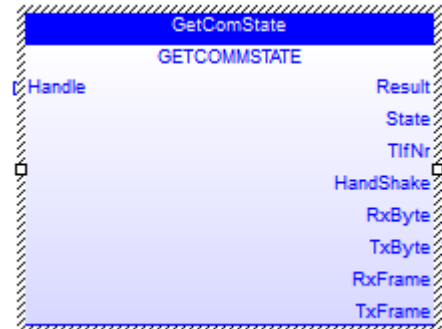
HandShake: bit 0 CTS ( in ) Clear To Send
bit 1 DSR ( in ) Data Set Ready
bit 2 RI ( in ) Ringing Indicator
bit 3 DCD ( in ) Data Carrier Detect
bit 4 DTR ( out ) Data Terminal Ready
bit 5 RTS ( out ) Request To Send
Bit 6..31 Not used

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 primarily intended for use with a modem driver, but could also be used to monitor data transmission of null modem drivers. Data returned when using multiple protocol drivers on the same COM port, is more port related than protocol driver related. Obtaining RxFrame values using handles from either passive or active drivers on the same COM port will return the same value.

Check your RTU32 CD or our homepage for useful examples of how to use the modem drivers.



GetDrvActive - Get active driver

This function will return the ID of the currently active driver (if any) on a COM port. It can be used by the RTU application to identify which protocol is presently active.

```
GETDRVACTIVE.bit - Notepad
File Edit Format View Help
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

This function can be called regularly by the RTU application to get the status of a driver.

```
GETDRVSTATE.bit - Notepad
File Edit Format View Help
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.

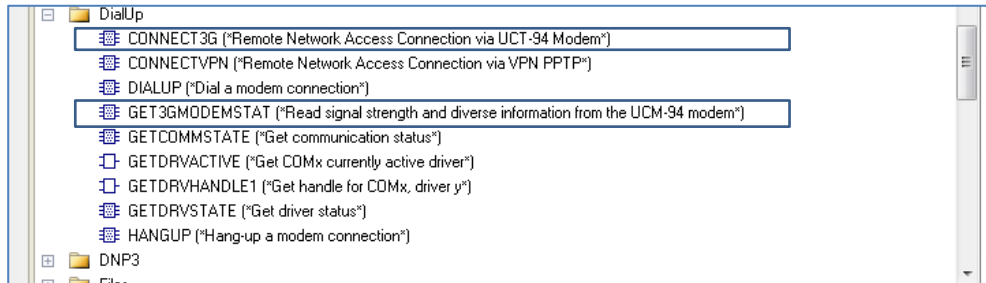
TxFrame : Number of protocol frames successfully transmitted
by this prorocol driver.
In modem mode the counter is reset to zero when line is hang-up.

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



7.15.2 Functions for control and monitoring the UCM-94 3G/GPRS Modem

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



The UCM-94 modem connect procedure can be done automatically at start-up or controlled in the PLC program using Functions listed shown below. Automatic or PLC controlled connection is defined on the modem configuration webpage. If automatic connection 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.

The available Functions are listed below.

Connect3G Function for PLC control

The Connect3G function is used when the PLC application controls the Modem connection. If automatic connection is selected on the RTUs web page this Function is not used.

```
CONNECT3G.blk - Notepad
File Edit Format View Help
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.

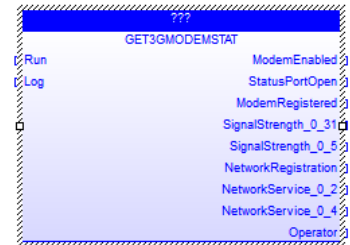
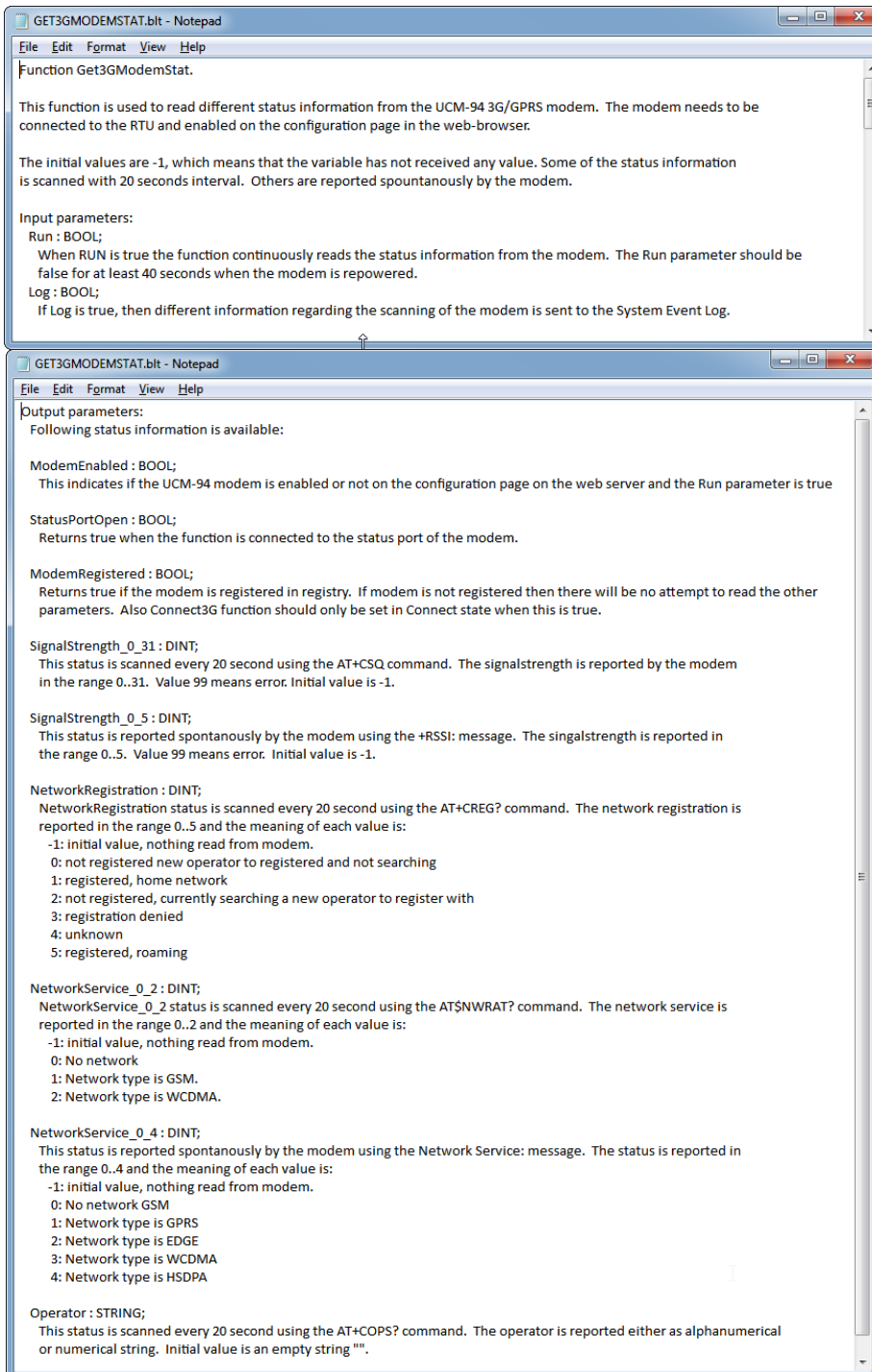
VAR_OUTPUT
  Connected : BOOL;
    Indicates the current reported connection state of the modem.
  Connecting : BOOL;
    Is set to true during the connection procedure until a connection is established.
  ConnectionNumber : UDINT;
    This variable is incremented every time a new connection is attempted.
  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 be used for monitoring the state of the UCM-94 modem and the 3G/GPRS connection status.



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 use 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. Note that the field strength does not equal QoS on the actual 3G/GPRS network connection. This is dependent on the network provider's equipment status and load on the provider network.

NOTE: It is recommended to use scanned values only as the spontaneously reported values are not always reported (due to the network provider account settings and type of SIM card).

7.16 Data Logging

WorkSuite provides functions for logging data into one or more file(s) in the RTU32S. In general 2 functions are available:

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

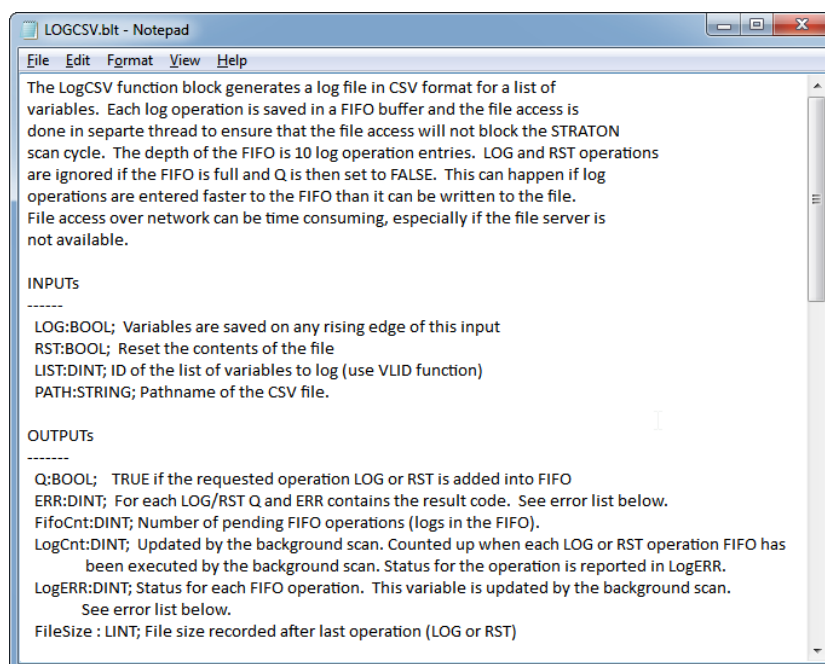
LOGCSV Function

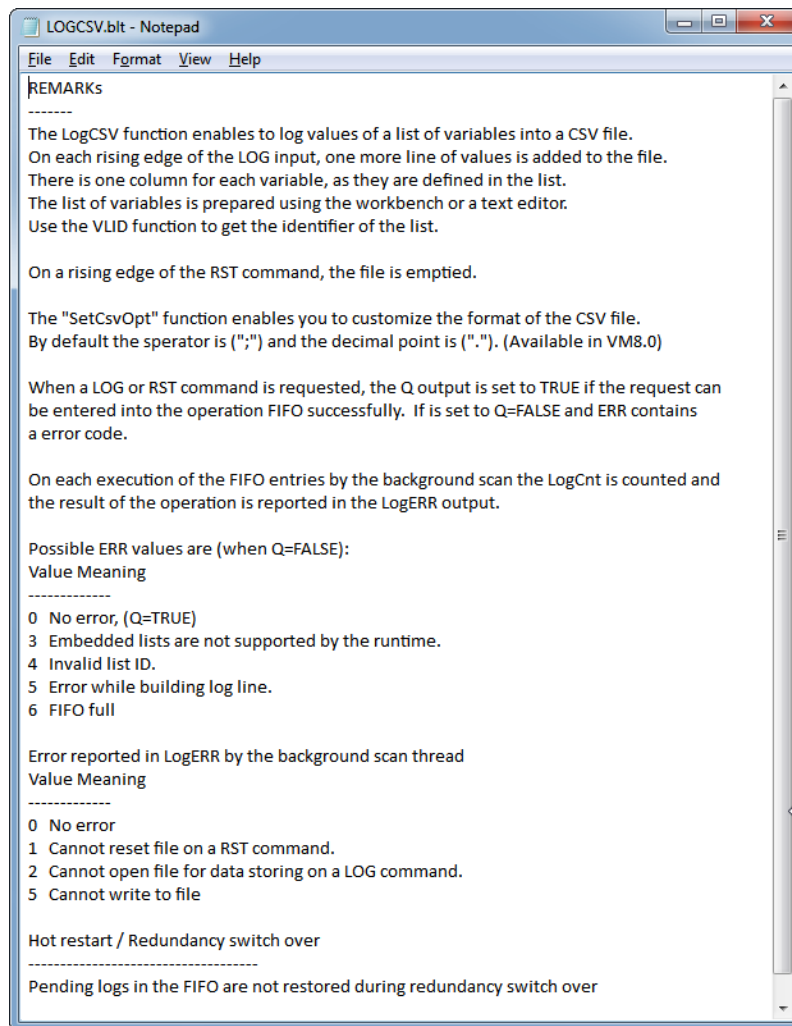
The LOGCSV Function is found in the Brodersen Function library folder.



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

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

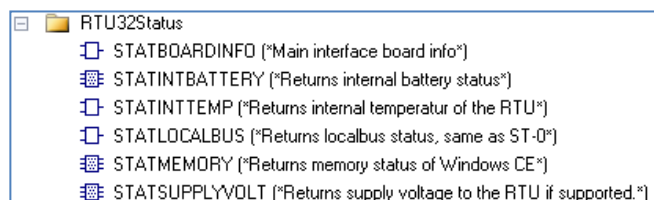




In the RTU32Example projects in WorkSuite you will find a WorkSuite Demo project LOGCSV_Demo with a log UDFB (User Defined Function Block) block named TMultiFileLog. With this UDFB you will be able to control the number of files to log and limit file size. This allows management of log files in a ring buffer, ensuring that your log files will not fill up the available storage memory.

7.17 Status information functions

RTU32S Series products with firmware version 1.55 and newer have hardware status information available.



Only some of the status functions are available in the RTU32S. The following sections list the functions supported by the RTU32S.



7.17.1 Interface Board Status Info (STATBOARDINFO)

The function StatBoardInfo returns information supplied by the main interface board in the RTU32S.

```
StatBoardInfo.blk - Notepad
File Edit Format View Help
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 avaiable; 0=Yes 1=No
```



7.17.2 RTU Internal Temperature (STATINTTEMP)

The function StatIntTemp returns internal temperature of the RTU32S.

```
StatIntTemp.blk - Notepad
File Edit Format View Help
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;
```



7.17.3 LocalBus Status (STATLOCALBUS)

The function StatLocalbus returns relevant I/O bus status values

```
StatLocalbus.blk - Notepad
File Edit Format View Help
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
```



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

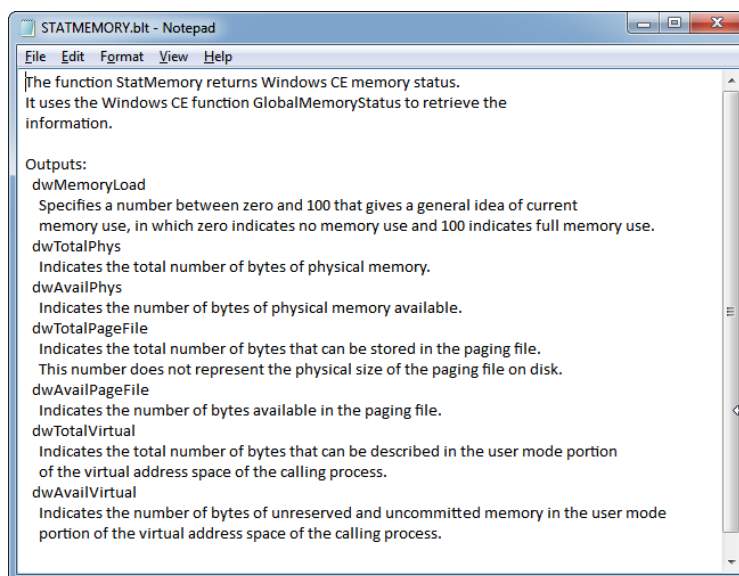
Here are some typical errors for the LocalBus;

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



7.17.4 RTU memory Status (STATMEMORY)

The function StatMemory returns Windows CE memory status.

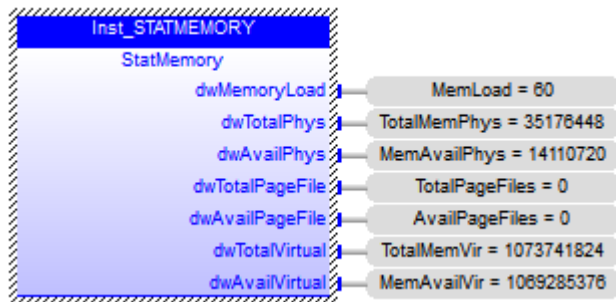


The commonly used monitoring functions are dwMemory Load, dwTotalPhys and dwAvailPhys.

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

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

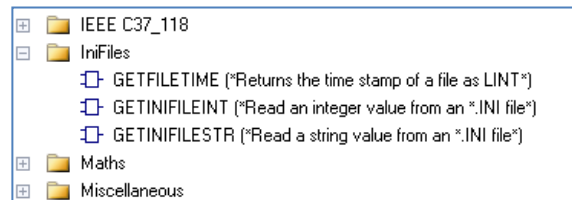
The dwMemoryLoad tells you in percent the actual RAM program memory load. With a medium PLC runtime application you will see a load of approx. 60-63%. Only applications with multiple driver connections and a lot of logic will require larger amount of memory. The maximum load recommended on a RTU is 90%.



Typical memory status in a running RTU32S

7.18 Reading Text Files from WorkSuite

To provide the functionality to read variables and strings from an external file in the RTU32S, a few simple read file functions are supported. The functions are found in the IniFiles folder of the WorkSuite logic blocks



Note: You have to place the text file in the RTU32S file system manually – use FTP to do this and use the Storage Card (SD Flash). MS Explorer can be used to explorer the RTU32 files – enter in the address field, like: <ftp://192.168.0.1/>. An empty folder will come up – right click in it and select login using your username and password (same as used for entering the web pages).

The file structure must be the same as other RTU .ini files (such as SNMP.txt and RTU32.ini). The structure is simple and looks like this:

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

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

The functions available are:

- GetFileTime

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

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

- GetIniFileInt

Read an Integer value from a section and item.

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



GetIniFileStr

Read a String variable from a section and item.

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

The idea with these functions is that you can place your 'dynamic' variables outside the RTU runtime application program. You can then use the same application program for several sites and manage the differences for each site in the file.

You can edit the file from the web editor with your browser or directly send a new file to the RTU32S with FTP.

More help is found in WorkSuite by pressing F1.

8 Not supported features and PLC Functions in the RTU32S

8.1 List of WorkSuite Functions not presently supported by the RTU32S

Some Functions and Functions Blocks in Brodersen WorkSuite are not supported by the RTU32S in the current version. The functions that are not supported are listed below:

Folder	Function / Function Block
Brodersen	SUPPLYMON
DialUp	CONNECTVPN
DNP3	All (for Master only!)
Files	All
IEC61850	All
IEEE C37_118	All
Miscellaneous	ACTIVESWITCH
Miscellaneous	GETACTIVESTATE
Redundancy	All
RTU32PIDAutotune	All
RTU32Status	STATINBATTERY
RTU32Status	STATSUPPLYVOLT
XML	All



9 Distributed Event Binding Protocol / Redundancy Binding Communication

9.1 General

The Distributed Binding protocol is a proprietary protocol for fast time stamped event based data transfer directly from one Brodersen RTU runtime to another. The Binding protocol is communicating directly from one RTU32 to another RTU32 in a LAN/WAN network. It can run regardless of any other application. In a network with several Brodersen RTUs you can use the Global Binding Editor to bind any global declared variables together. Binding can be used between any type of RTU32 Series product and even with STRATON PC runtimes (soft PLCs).

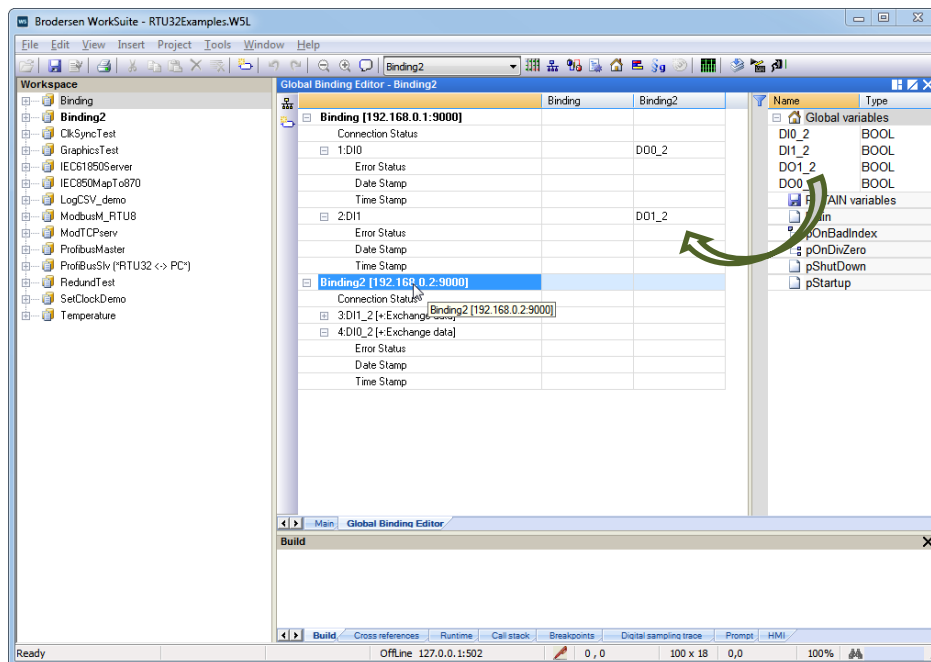
Redundancy Binding is used in a redundancy network connection using 2-6 separate network segments. Both the unit publishing variables (producer) and the unit subscribing variables (consumer) has to be equipped with 2-6 Ethernet network interfaces. Each network connection must be setup on the units as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

9.2 Network settings in RTU32 (redundancy binding)

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

9.3 Procedure for setting up binding/redundancy binding in WorkSuite

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



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



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

After configuration you download ALL projects to the relevant RTUs. Use the “Download All Projects...” in the Project menu,

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

10 Optional SNMP Extension Agent Driver

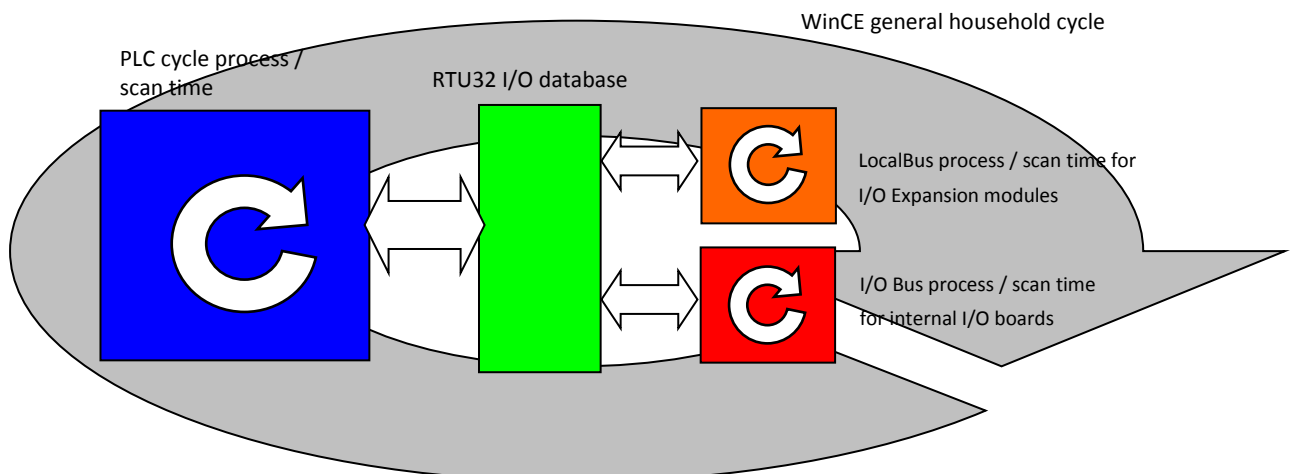
RTU32S does not yet support the SNMP Agent driver.

11 RTU32S cycle / scan time

11.1 General

The RTU32S includes 4 processes which are relevant when determining the timing performance for a RTU32S:

- PLC runtime program cycle/scan time
- LocalBus cycle for expansion I/O modules
- Internal I/O Bus cycle/scan time
- Windows CE general household cycle



Each cycle is running fully independent (asynchronous) from each other and must be considered without internal synchronisation. The cycles do 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 RTU 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 driver is reading/writing I/Os and errors on the I/O modules. The LocalBus itself is a simple but robust shift register bus.

The Internal I/O Bus is managing the scan for the built-in I/Os in the RTU32S. It is providing fast I/O updates because it uses an internal parallel I/O bus communication.

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



11.2 RTU32S Scan cycle mechanisms

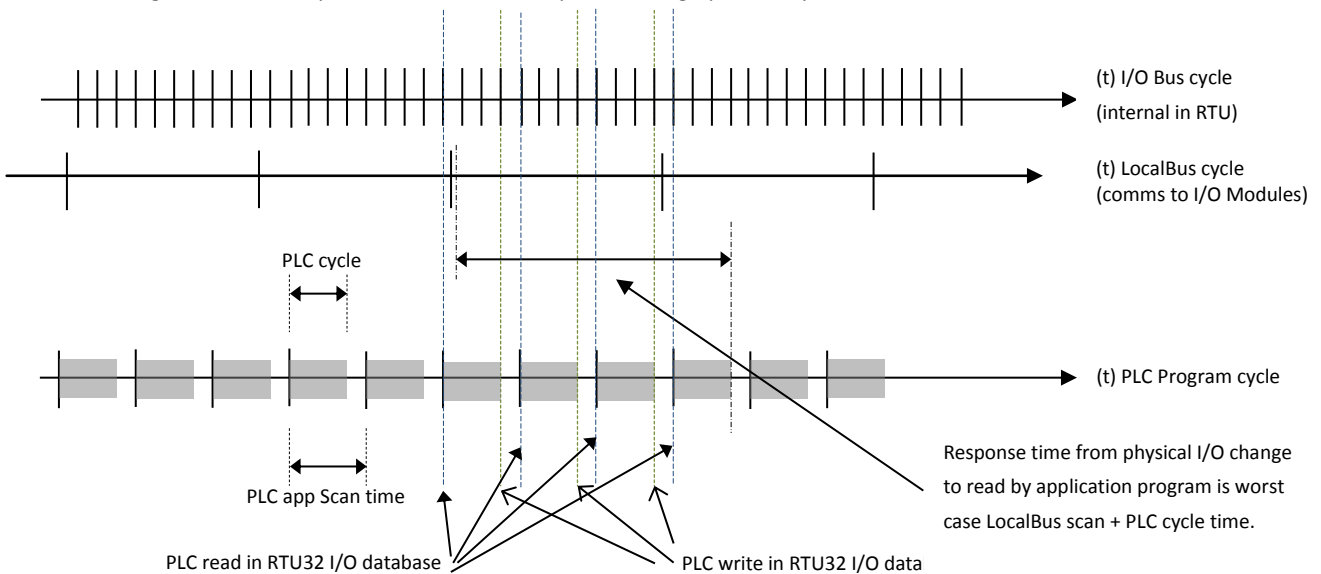
All tasks in general run independent of each other.

The two I/O cycles are running as fast as possible to ensure that the basic RTU32S I/O database at any time reflects the physical I/Os – both internal I/O and I/O on the expansion modules connected. The PLC runtime program cycle time is depending on the size of the application program, and can vary from less than 1 msec to >100 msec.

At runtime application start, all inputs are read from the I/O database and at the end of each cycle the outputs are written to the I/O database.

The runtime application scan time (time between each cycle start) can be adjusted – see section *RTU scan time* section for details.

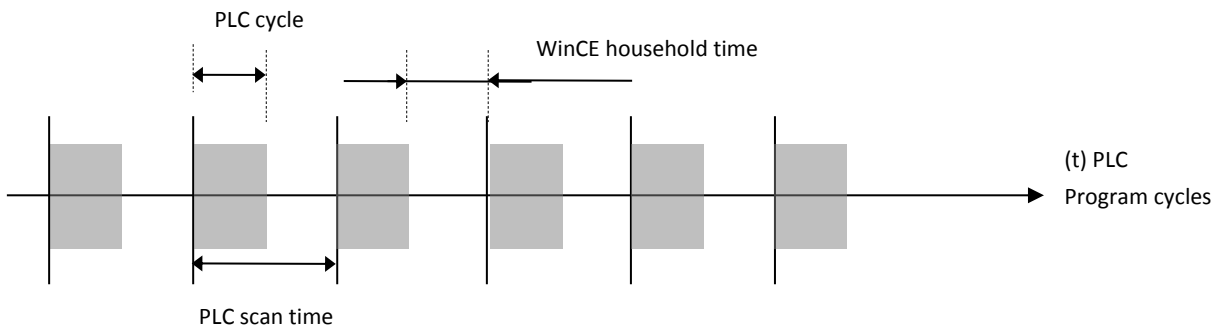
The link timing between these processes can best be explained in a graphical way:



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 RTU application program size and number of connected I/O expansion modules.

11.3 RTU scan time

The RTU scan time can be set to 0 (zero) which automatically assigns some time for operating system household activities between each runtime program cycle (minimum 40msec in RTU32S). In this case the program runs as “fast” as possible. The automatic assignment of household execution time ensures all necessary operating system processes can be maintained with a good margin.



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

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



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

11.4 LocalBus scan time

The LocalBus driver is used for internal I/O boards and all connected I/O Expansion modules.

The scan time of the local bus to Expansion I/O is running at a fixed initial scan time and is also dependant on the number of I/O Expansion modules connected. 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 is configured and the I/Os are scanned. Cyclically the I/O configuration is checked for errors.

LocalBus scan time on the RTU32S is from 80-100msec depending on I/O configuration.

11.5 Internal I/O Bus scan time

The scan time of the RTU32S internal I/O is 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 RTU32S.

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

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

Configuration

I/O board configuration 28IO with 16 digital inputs (also working as counter inputs), 8 digital PNP outputs, 4 analogue inputs.

I/O type	Scan time
Digital inputs/outputs	10 ms
Analogue inputs	100 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.



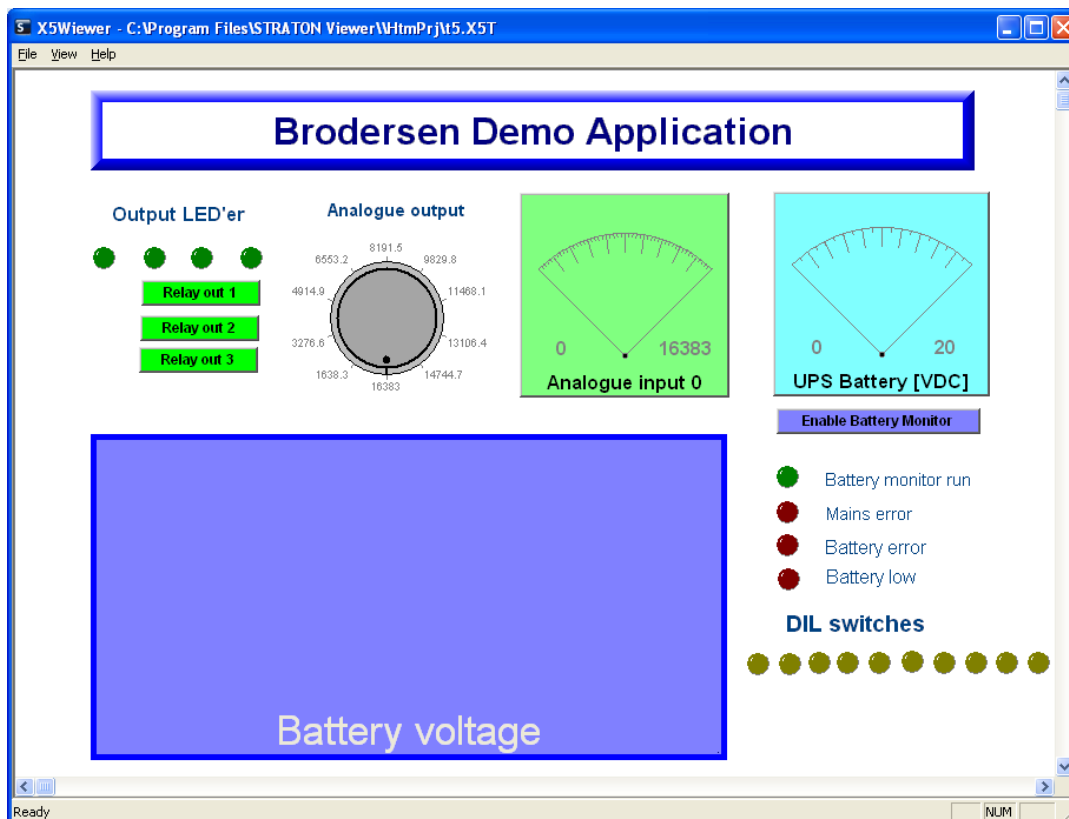
12 RTU HMI

12.1 General

The RTU32S Series support simple HMI functions developed in the WorkSuite tool. By adding a graphic program to your PLC application and downloading it to the RTU you can monitor data online from your PC using an IE browser or the special WorkSuite HMI Viewer. The HMI graphics require downloading separately (they are not downloaded automatically with the PLC runtime application).

Note: at this time the HMI pages cannot be accessed from the RTUs standard web interface.

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.

13 RTU32 Utilities and System Event Log

13.1 General

Brodersen provide some utility programs for the RTU32 Series 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



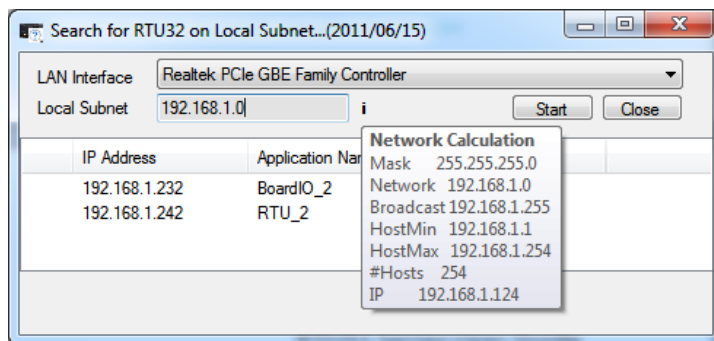
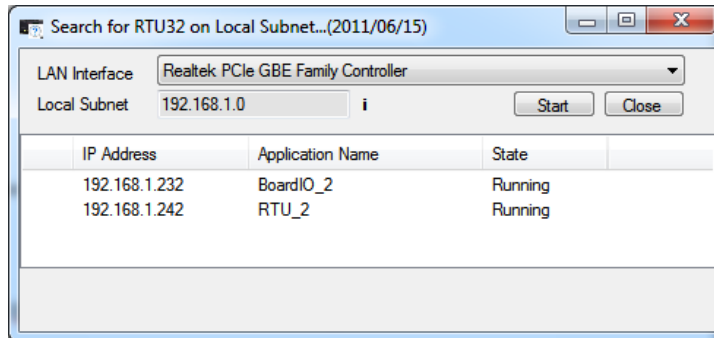
13.2 RTU32 Software Bulk Update Tool

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 the LAN Interface. Be careful that you select the right one before you start the search procedure.

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

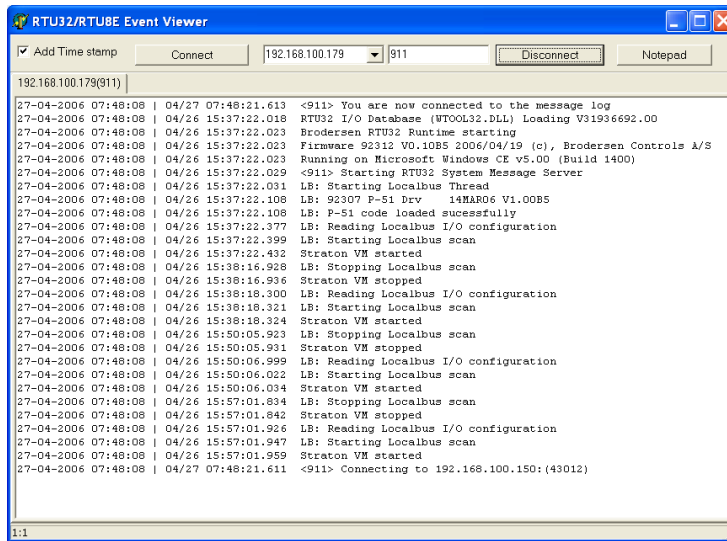


When you start the search process, the network will be scanned for Brodersen RTUs. Only RTUs with their RTU PLC runtime running will answer. If you select an RTU on the list of RTUs found on the network and right-click, you will be able to open either the Webpage or the Event Viewer (status Log info).

13.4 System Log/Event Viewer

The system log provides you with information about started processes in the RTU32S, errors and status information's. You can also view debug or information text from your WorkSuite application program which will be printed in the system log. COM port communications can be logged if you add the right parameters to the COM port setup – 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:

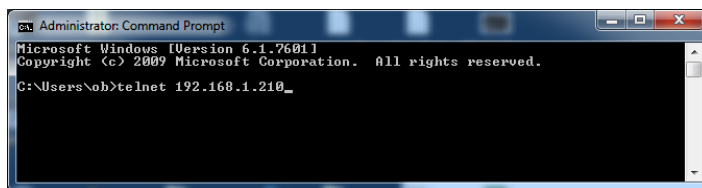


The RTU32 allows max 20 continuous connections and the log FIFO include max 100 messages.

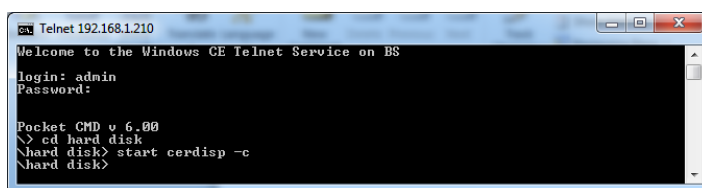
13.5 Remote Display / Remote Desktop Tool

The RTU32S series includes a remote desktop function, where the Windows CE Cerdisp function can be used for showing your RTU32S Desktop on your local PC.

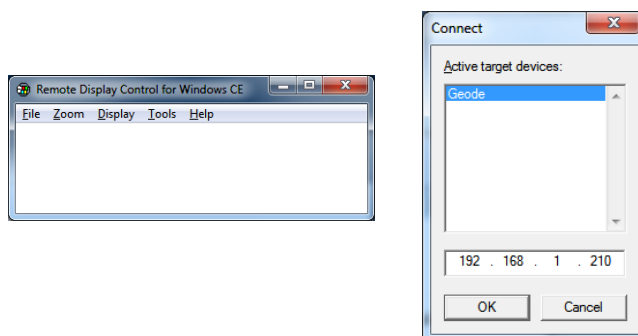
If you want to use the Remote Desktop function, you must start the program on your RTU32. Use telnet to connect to your RTU like this:



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

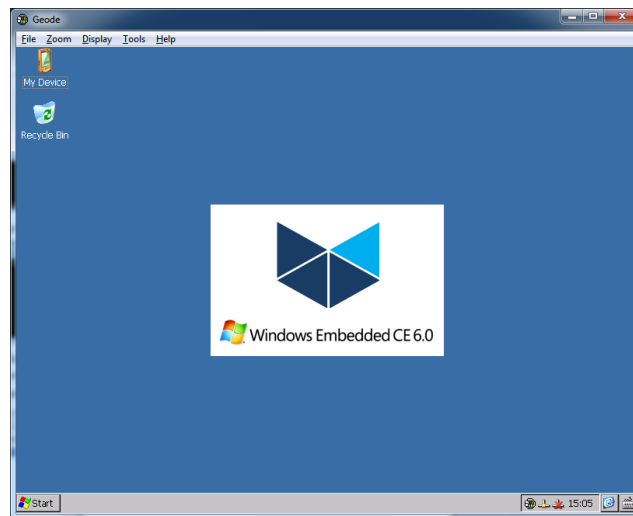


Now you are ready to start the Remote Display Control for Windows CE to get access to the remote desktop. It is found under the /Brodersen/WorkSuite/RTU32 Tools folder in your Start menu. After starting the RTU32 Remote Desktop Host you must use the File menu and select Connect. Now the Connect window will appear and you enter the IP address of your RTU32S and enter OK.





- The remote desktop will then appear on your computer.



NOTE: This feature is only used for special adjustments as the remote desktop task has low priority.



14 RTU32S Technical Description

14.1 General

The RTU32S Small RTU/PLC and Utility Controller is based on an ARM RISC platform running Windows Embedded CE operating system, taking advantage of the well-known embedded Microsoft environment facilities. An industrial power supply is integrated to support industrial supply levels. The basic RTU32S includes a range of communication and other interfaces. The RTU32S provides interfaces like LocalBus for expansion I/O and integrated I/O.

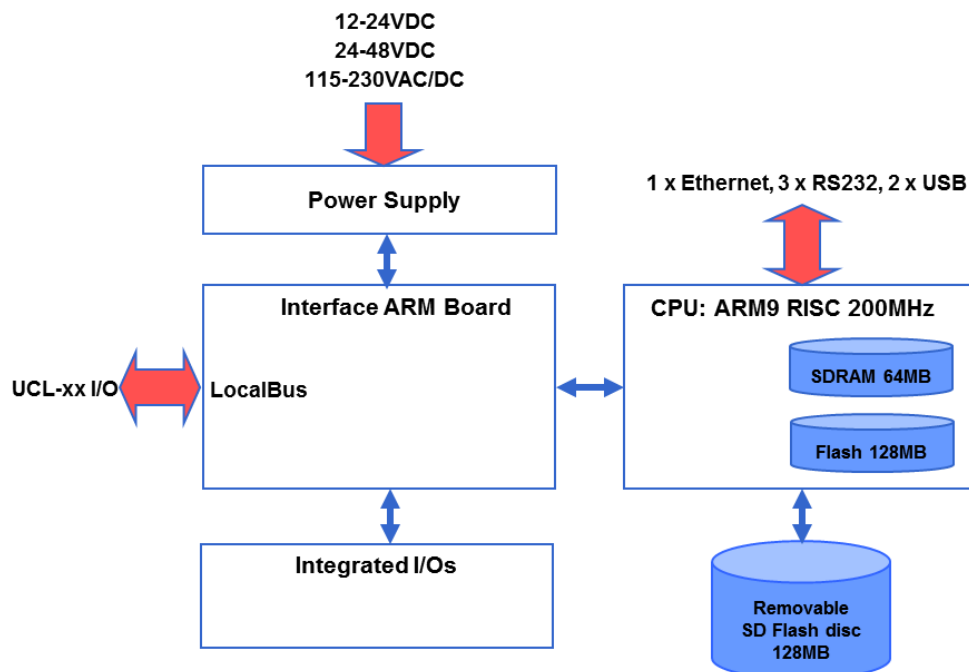
The RTU32S software is stored on an internal Flash. During start-up, the operating system and applications are moved to RAM where it is executed. Retained variables are stored executing a special command in the application program.

Windows Embedded CE is the RTU32S real time operating system. Integrated I/O and LocalBus for external I/O connectivity are controlled in an implemented I/O database. A PLC VM (Virtual Machine) is ported to the Windows Embedded CE real-time task. This enables the PLC runtime application program to be executed in the RTU32S.

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 RTU32S hardware is provided in the RTU32S Data Sheet and in the RTU32S Wiring and Mounting Instruction manual.

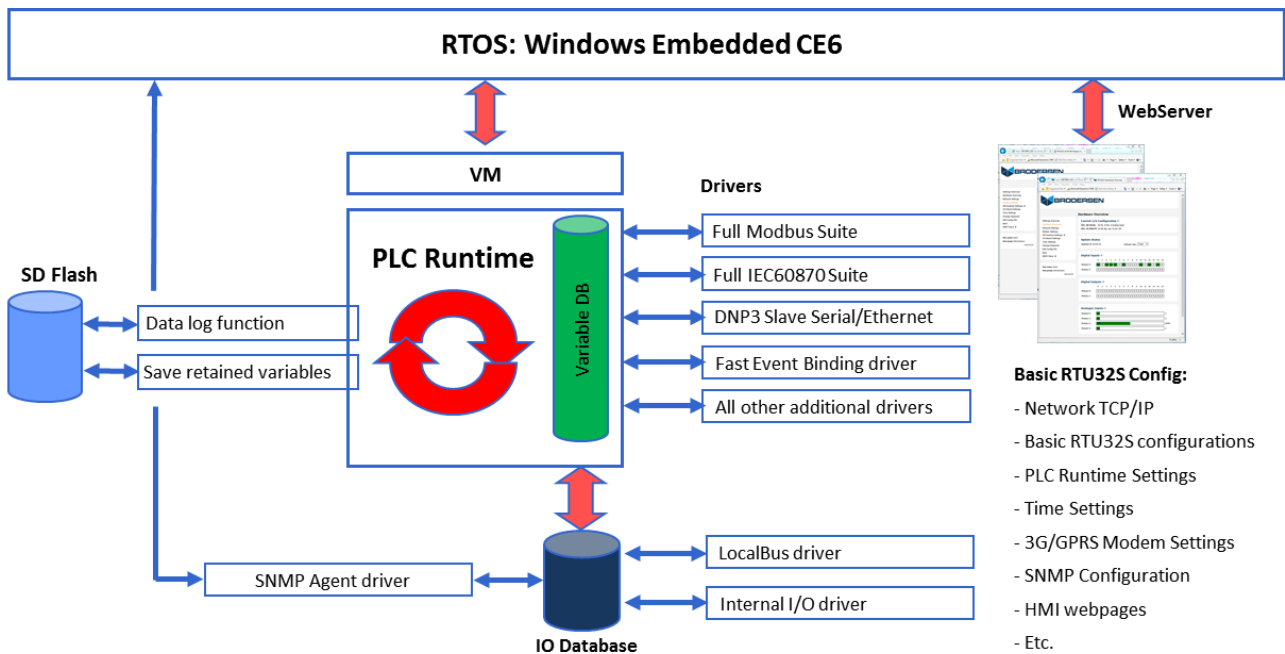
15 Hardware block diagrams



Note: the block diagram includes 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 information about the RTU32S and its functions or possibilities, please contact Brodersen support.

15.1 VM PLC Runtime

The WorkSuite programming tool fully supports EN/IEC61131 and is used for developing PLC programs in the RTU32S. The application program kernel is implemented and runs in the Windows Embedded CE real-time task. WorkSuite offers complete PLC functionality as found in both normal PLC runtimes and in SoftPLCs, and supports all features needed in today's industrial environment. WorkSuite supports programming languages including Structured Text, Function Block, Ladder, Instruction List and Sequential Function Chart. The VM PLC runtime supports cold restart, hot restart and on-line changes.

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

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

The RTU32S application kernel is used in many products and applications worldwide. More details about WorkSuite including training and help can be gained by contacting your local Brodersen distributor or Brodersen support.

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 a data type sorted IO-data structure. The IO-data structures operate with the data types as defined in the WorkSuite I/O Driver.



15.2.2 LocalBus (LB) driver

The LocalBus driver is running cyclic in a high priority thread, and always started at RTU32S boot time. However start and stop of LB scan is controlled by PLC runtime application.

When started, LB configuration is read, and a 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 cannot be built, the 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. This allows support of a lock function which freezes the LocalBus configuration. If the lock is enabled the configuration read at PLC runtime application start is 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 RTU32S LocalBus driver has a maximum limit of 12 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 from port 911. The information about the LocalBus driver is identified by "LB:" and includes information and status like:

```
07/16 11:18:13.109 LB: Starting Localbus Thread
07/16 11:18:13.110 WD: Attach (1) Localbus :Timeout: 10000ms, Total Threads: 1
07/16 11:18:13.112 WD: Attach (2) TimerThread :Timeout: 5000ms, Total Threads: 2
07/16 11:18:13.118 LB: I/O Controller device driver v1.00.02 loaded
07/16 11:18:13.133 LB: IOM:IOSRTUI S/N:5CED505E3435 Firmware 92749 V0.10B7 2013/02/28
07/16 11:18:13.171 LB: Internal I/O board type 28 IO is detected 10/27 10:26:02.079 LB: P-51 Firmware 92307 V1.11 2007/04/23
07/16 11:18:13.174 LB: Hardware watchdog enabled (no jumper)
07/16 11:18:13.826 LB: Power supply monitor detected. V1.00 2012/07/10
07/16 11:18:13.826 LB: Watchdog is enabled
07/16 11:18:13.852 STRATON VM Minimum Scantime 40ms
07/16 11:18:13.891 DLL: T5BusExBDNP3SL3.dll loaded
07/16 11:18:13.951 Binding HeartbeatPeriod/Timeout = 2000/3000
07/16 11:18:13.952 UCM-94 and VPN are disabled
07/16 11:18:13.985 -- Variables used(264) max(20000) bytes
07/16 11:18:13.986 -- CT Segment used(68) max(20000) bytes
07/16 11:18:13.986 -- #I/Os used(27) max(700)
07/16 11:18:13.987 -- #Drivers used(0) max(2)
07/16 11:18:14.062 LB: Reading Localbus I/O configuration
07/16 11:18:14.117 LB: The following I/O modules are attached to the RTU:
07/16 11:18:14.126 LB: UCx-28 IO.Dx 16 DI, 8 DO, 4 analog input
07/16 11:18:14.132 LB: UCx-16 DIO.P1 16 bit dig. inp. & out. DC
07/16 11:18:14.157 LB: Starting Localbus scan (99ms)
07/16 11:18:14.181 RetainVar: 8+16+32+64+TM+ST+CT(0+0+0+0+0+0)=0 (Max=20000)
07/16 11:18:14.196 WD: Attach (3) Main Comm Thread :Timeout: 120000ms, Total Threads: 3
07/16 11:18:14.216 WD: Attach (4) VM Thread :Timeout: 10000ms, Total Threads: 4
07/16 11:18:14.219 Straton Project : 'RTU32S_demo'
07/16 11:18:14.219 Build time : 2014/07/14 13:15:38
07/16 11:18:14.222 Build version : 24
07/16 11:18:14.223 Build CRC : 16#6261ef6d
07/16 11:18:14.412 Straton VM started (446ms)
```

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

```
07/17 12:39:58.119 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 when the PLC application starts. It will tell you what configuration was expected and what is actually detected.

```
07/17 12:41:16.032 LB: Reading Localbus I/O configuration
07/17 12:41:16.087 LB: Warning, I/O config is locked by user, and mismatch is detected
07/17 12:41:16.087 LB: The following I/O modules was expected attached to the RTU:
07/17 12:41:16.096 LB: UCx-28 IO.Dx 16 DI, 8 DO, 4 analog input
07/17 12:41:16.103 LB: UCx-16 DIO.P1 16 bit dig. inp. & out. DC
07/17 12:41:16.104 LB: The following I/O modules are attached to the RTU:
07/17 12:41:16.113 LB: UCx-28 IO.Dx 16 DI, 8 DO, 4 analog input 10/27 10:31:41.974 LB: Reading Localbus I/O configuration
```



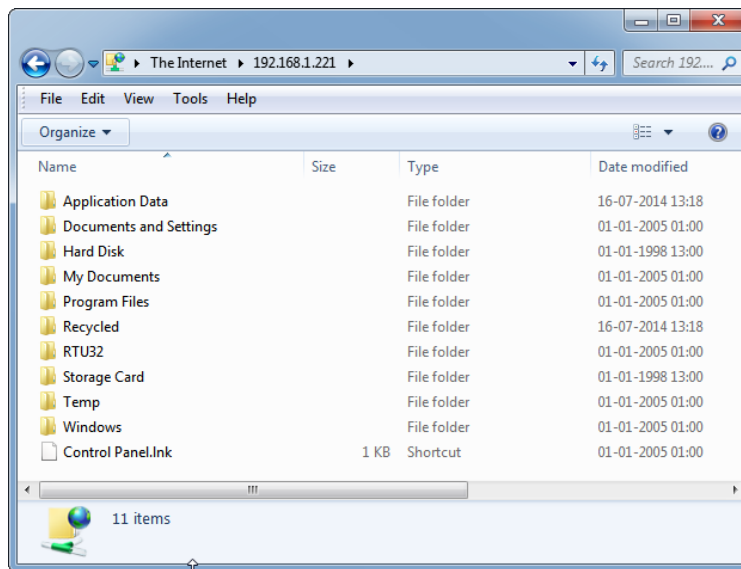
The Event log messages are useful for service engineers to detect errors in configuration.

15.3 RTU32S File System

15.3.1 General

When you apply power to the RTU32S, it will boot from the internal Flash disk. All necessary files for OS, PLC kernel VM and standard drivers are located on the internal Flash. The PLC application program and optional drivers are located on the SD Flash disc (Storage Card). During boot the OS is moved to run on the internal RAM. You will not have access to any files on the internal Flash disc (/Hard Disk) except for a Documents and Settings folder. And these folder and files must NOT be deleted or changed. They are managed and used by the OS only.

If you access a running RTU32S with FTP you will see the root directory like:

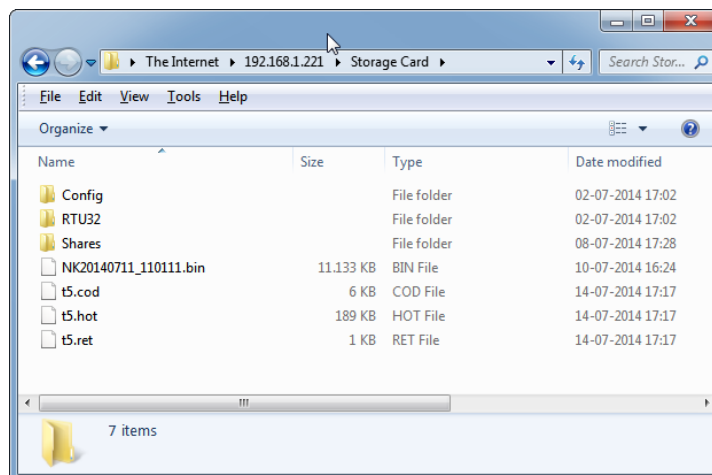


The root directory contains a number of subfolders and system files.

15.3.2 Relevant folders to review and use

../Storage Card/ folder

The Storage Card folder and files include the PLC runtime project files (t5.*) and folders for configuration and optional drivers files.





If you want to manually to delete a PLC runtime project in the RTU32S, you must delete the t5.cod file. See more details in the appendix about resetting RTU32S. If you configure data logging you should ALWAYS log on the Storage Card – never on the internal flash disc. It is recommended that you create a LOG folder and store all log files there.

NOTE: In the figure above there is a NKxxxxx.bin file. You will only find this file on the Storage Card if you have updated the RTU32S firmware.

The RTU32 sub-folder includes the driver files and other system files required for the PLC runtime.

The Config sub-folder contains the configuration files. The RTU32.ini file defines the basic RTU32 configuration. The SNMP*.txt files are used by the optional SNMP Agent driver. If you are going to use the read text file function in PLC runtime, your file/s should be placed in this folder.

15.4 Software Versions

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

RTU32S Version Information	1.55.1.0/1.18.10
RTU32S Firmware Date (RTU32S.exe)	2014/07/10
RTU32S Firmware ID (RTU32S.exe)	92746 W910
VM Runtime Version (RTU32.exe)	8.5.121210
RTU32S CE Image Date (NK.bin)	2014/07/10
RTU32S Windows CE Version:	6.0
RTU32S Windows CE Build Version:	1937

The software in the RTU32S includes 3 main software parts:

- Basic RTU32S Version – like above 1.55.1.0/1.18.10
- VM Runtime version
- Windows CE image

The VM runtime project details include the name and build version of the PLC runtime in the RTU32S.

RTU32S Firmware includes the specific RTU32S functions and includes all files in the internal Flash. The RTU32S firmware is assigned an id (e.g. 92312), a date and a version. The RTU32S CE Image in the single NK.BIN file is assigned a date and version – it is stored on the internal flash and is not visible via ftp.

VM version is the version of the Virtual Machine SDK supported by the RTU32S (Virtual Machine = PLC runtime handler). It provides general information of what version of Brodersen WorkSuite is supported by the RTU32S. The RTU32S is developed and maintained to support full compatibility, but the VM version tells you if the latest WorkSuite functions are supported in the RTU32S.

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 **RTU32S 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
1.55.1.0	1.18.10	1.55-1	<ul style="list-style-type: none">• Released as version 1.55.1.0 / 1.18.10.• Runs on RTU32S based on ARM platform.• Supports 3G Modem.• WWW new version 1.55.• Alignment problem fixed.• New function block and Function library added:<ul style="list-style-type: none">○ RTU32 Status functions.○ Hardware overview webpage added to monitor online status on physical I/O.○ Driver version view added to webpage Settings overview.
1.54.0.1	1.18.02	None	<ul style="list-style-type: none">• First released as version 1.54.0.1 / 1.18.02.

16.2 Important notes about the latest versions

None.



Appendix 1

RTU32S runtime application measurement / benchmark

The RTU32S runtime cycle time is measured in a number of different applications below to provide examples of expected runtime periods.

The RTU32S scan time is set in WorkSuite to 0, which means that WorkSuite automatically assigns 40msec of time for household activities. It also ensures that your logic application, even with extensive use of WinCE functions like WinSock for communication drivers will run smoothly and avoid any overrun situations.

Example 1: RTU32S runtime application program without any communication

Test configuration:

- 1 x RTU32S 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, some simple calculations are executed, data is stored in internal registers and then copied to an analogue output. (~15 PLC instructions per analogue channel).

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

6 digital inputs are controlling 6 counters connected to a digital output.

Result:

The typical PLC scan time is 40msec and cycle time is 2-3mS (42-43mS total).

Example 2: RTU32S application program with ModbusTCP communication

Test configuration:

- 1 x RTU32S 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, some simple calculations are executed, data is stored in internal registers and then copied to an analogue output. (~15 PLC instructions per analogue channel).

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

6 digital inputs are controlling 6 counters connected to a digital output.

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

Result:

The typical PLC scan time is 40msec and cycle time is 2-3mS (42-43mS total).



Comments to the bench mark tests and recommendations

Using basic PLC logic functions does not have a large impact on the cycle time and load on the RTU32S. Using any kind of Windows function within the application does have impact on the cycle time. Functions such as Ethernet/Winsock, file write functions (data logging), read/write time and get status from OS controlled consoles like LAN adapter status will have impact on the scan time. Large applications with +20 connections to TCP Servers (e.g. ModbusTCP), data logging to 4 files and constantly reading status of a network adapter will mean a scan time of about 30-50msec.

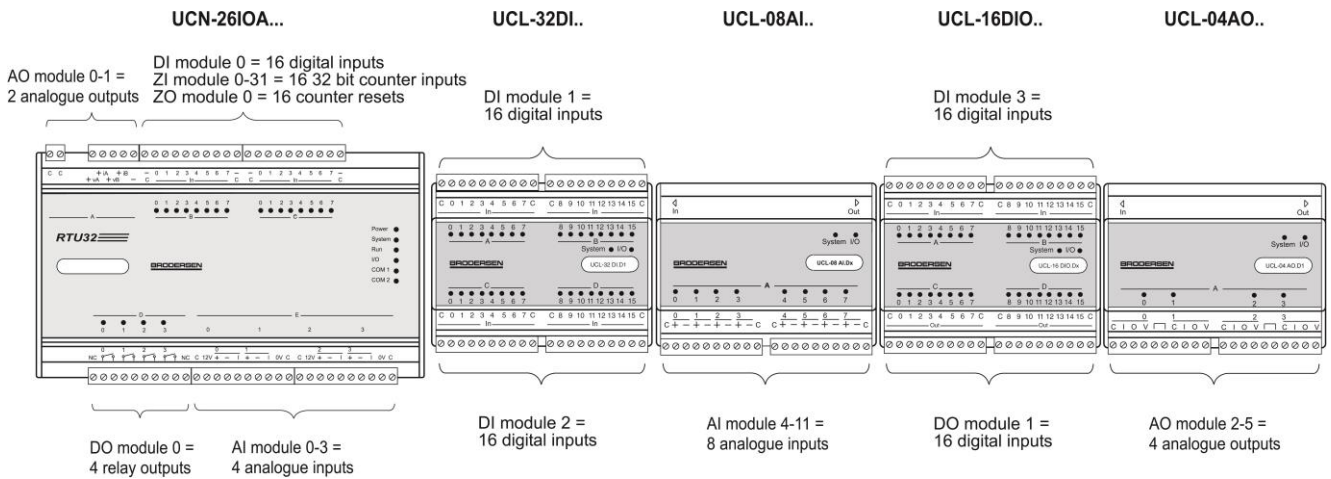
It is recommended to always set the PLC scan time to the highest acceptable scan time. However, the RTU32S will always keep a minimum of 40msec for household activities no matter how you set the PLC scan time (periodic cycle time) in your PLC project.



Appendix 2

RTU32 I/O addressing example 1

The addressing example shows the RTU32 - the same structure is used by ALL RTUs in the Brodersen RTU32 Series.



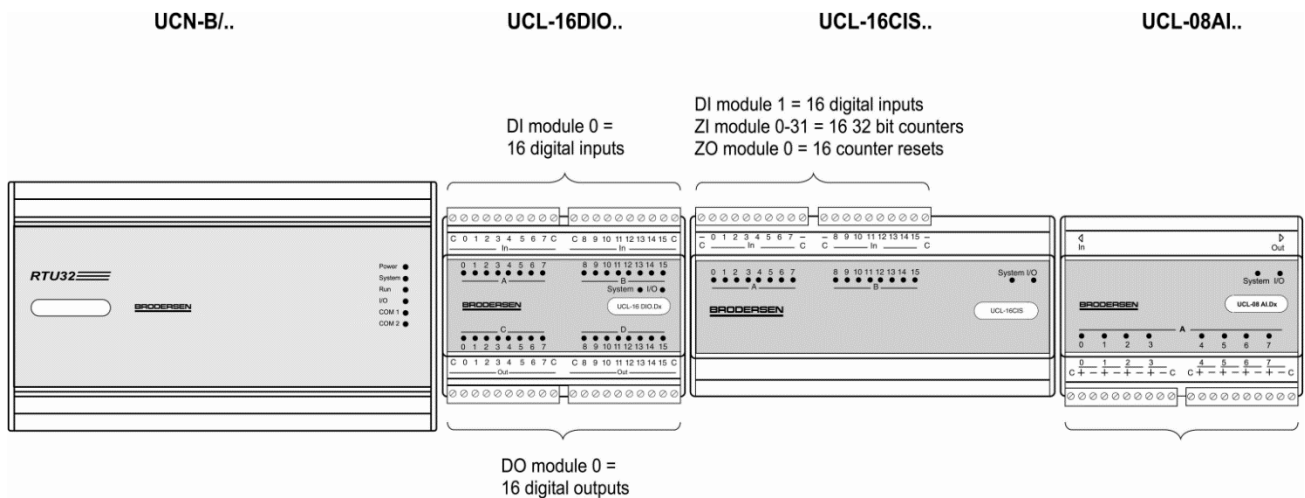
NOTE: Analogue input module with 16bit resolution and single isolated channels type UCL-08AIC does covers two modules (two words) for each input. First Module (Word) is the 16bit analogue values and the second Module (Word) does report channel status information like overflow, underflow, disabled, error etc.



Appendix 3

RTU32 I/O addressing example 2

The addressing example shows the RTU32 - the same structure is used by ALL RTUs in the Brodersen RTU32 Series.



NOTE: Analogue input module with 16bit resolution and single isolated channels type UCL-08AIC does covers two modules (two words) for each input. First Module (Word) is the 16bit analogue values and the second Module (Word) does report channel status information like overflow, underflow, disabled, error etc.



Appendix 4

RTU32S PLC Runtime application RESET

The RTU runtime application uploaded and executed on the RTU32S is stored in a number of files on the root of the removable Flash disc (SD Flash card). If your runtime application is not running correctly, or is taking up all CPU resources making it impossible to establish contact via the WorkSuite tool or the webserver pages, you can delete the complete RTU runtime application.

To manually delete the RTU32S runtime application you will need your laptop/PC and an internal or external SD Flash card reader.

Procedure:

Power down the RTU32S and remove the SD Flash card.
The SD card is removed by pushing it to release.

Delete the T5.* files from the root of the Flash card.

Place the SD card back in the RTU32S and apply power. When the RTU restarts the same firmware is installed, but the RTU runtime application is reset.

You can now download and execute a new RTU application via WorkSuite.



Appendix 5

RTU32S complete Reset

You can make a complete reset of the RTU32S that will reset it back to factory default settings. Note that during reset process all settings in the RTU32S are lost.

To make a complete reset you will need a small screwdriver, pointed pen or paperclip for activating the reset button. On the underside of the RTU32S (to the right of the SD card location) you will find a small 3mm hole. Behind this is a small push button that is used in the reset process.

Procedure:

Power down the RTU32S.

Use a small pointed device to push/activate the reset button while you apply power to the RTU32S. Keep the reset button pressed for a minimum of 9 seconds before it is released.

The internal Flash memory of the RTU32S is now cleaned and a new installation of the firmware is installed during boot. The RTU32S will then start-up with factory default settings. LAN IP address = 192.168.0.1 and password erased.



Appendix 6

RTU32S firmware update

You can update the firmware in the RTU32S. The firmware is contained in one single file (NK.BIN).

NOTE: During a firmware update your PLC runtime application will be lost. If you have additional drivers installed on the Flash card, it is recommended you make a copy of the driver file/s (/RTU32/*.*.dll) before you start the update

To update the firmware in the RTU32S you will need your laptop/PC with an internal or external SD Flash card reader and the new NK.BIN file.

Procedure:

Power down the RTU32S and remove the SD Flash card.
The Flash card is removed by pushing it to release.

Delete all files from the root of the Flash card.
Copy the new NK.bin firmware file to the root of the Flash card.

Place the SD card in the RTU32S and apply power. When it boots the new firmware is installed and the RTU32S will reset to factory default settings.
During boot up the COM LEDs sequence will be:
COM1: ON for 15-20s (while new firmware is copied to on-board Flash)
COM2: ON for 1-2s (new firmware detected)
COM1: ON for ~10s (boot loader startup)
COM2: ON for ~10s (OS booting up)

The firmware file on the SD card will then be renamed to NKyyyyymmdd_hhmmss.bin

After updating the RTU32S firmware, it will start with the factory default settings - LAN1 IP address is set to 192.168.0.1, subnet 255.255.255.0, password is reset and an empty PLC application running in the RTU32S.

If you have made a backup of a driver file, you must make sure that it is compatible with the new firmware and you must place it in the SD Flash card folder \RTU32\ to make it work.