# RTU32M Series – Example 1

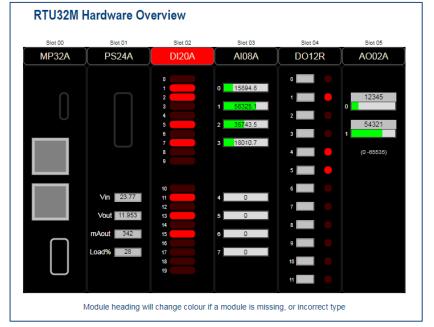
### RTU32M Application - Creating a Smart 'I/O Box'

### **Application Note**

March 2019



RTU32M – modular hardware



The 'No Cost HMI' – HTML page published by WorkSuite for viewing in a web browser (Smart Phone, Tablet, PC etc)

#### Introduction

Brodersen have been manufacturing products for use in remote monitoring and control solutions for almost 50 years. Our customer base is global and our products are used in a diverse range of Applications that include energy management systems, water and waste water SCADA, infrastructure monitoring, building automation and airport management systems.

This Application note provides an overview of how I/O points, logic, communications parameters and graphics are setup in the RTU32M. Local/HMI access can also be provided to user data via a web browser using HTML pages published from WorkSuite. Follow the steps described to create your own application, or use the link at the end of the document to download the project for review/modification.

#### **RTU32M Overview**

The RTU32M modular hardware provides a flexible architecture for creation of RTU solutions. Start with a CPU and power supply, then add I/O modules or system modules to create you desired solution. The I/O can be together in a single block, or distributed in a 'segmented' arrangement of I/O blocks. All RTU32M modules provide access to their status, firmware and hardware revision information and serial number. I/O modules include data values, data quality and timestamps of when the last change occurred (1ms resolution).

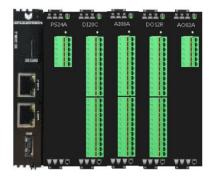


### **RTU32M Setup and Creation of an Application**

The hardware components

In this example, the selection of RTU32M modules used includes;

Slot	Module	Description
0	MP32A	CPU (2x LAN, 1x USB, 200-900MHz)
1	PS24A	Power Supply (10-30V DC)
2	DI20C	20ch Digital In (incl. 2x 5kHz counters)
3	Al08A	8ch Analog In
4	DO12R	12ch Relay Out
5	AO02A	2ch Analog Out

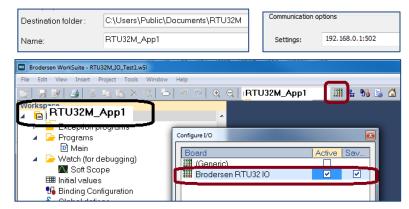


#### Creating an Application and module list in WorkSuite

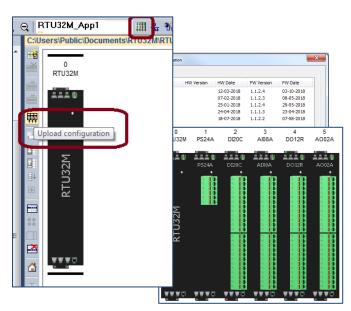
Brodersen RTU32M Applications require WorkSuite version  $\geq$  3.9.2. The link below downloads v3.9.2.1218. Brodersen WorkSuite 3.9.2.1218 (~450MB).

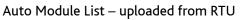
In this example an Application named 'RTU32M\_App1' was created using 'File>Add New Project', using the RTU32M default LAN 1 IP setting of 192.168.0.1:502.

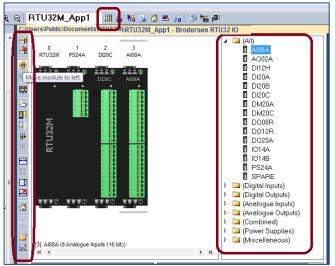
To access the RTU32M I/O Setup, the I/O Configuration Editor must be set to use Brodersen RTU32I/O. Use the I/O icon to access the I/O Editor.



WorkSuite allows creation of the RTU I/O module list by querying the RTU for its list of modules, or if the RTU is not present, the list can be created by adding/moving/deleting modules using 'drag and drop' from the menu.







Manual Module List – using 'drag and drop'

### Setup I/O, Create Variables, App. Download

Click on a module to change its setup

Each I/O module has its own configuration page with various setup parameters.

DI Parameters	DO Parameters
Invert	Fault Mode
Count	
Debounce	

Al Parameters	AO Parameters
Range/Mode	Range/Mode
Filter	Scale Min/Max
Sig. Change	Fault Mode
Scale Min/Max	Fault Value

	0	1	2	3	4	
Inverted:						Digital Outputs configuration
Counter:						0 1 2 3
Debounce (mS):	0	0	0	0	0	Fault mode: Low V Low V Low
	10	- 11	12	13	14	
Inverted:						
Counter:						Module not present in configuration
Debounce (mS):	0	0	0	0		
Analogue Inputs Range: Filter: Delta Change:	s configuration 0 - 5 V Medium IIR 0.500 0.00 65535.0	<ul> <li>▼</li> <li>4-:</li> <li>Med</li> </ul>	1 20 mA V ium IIR V 0.500 0.00 55555.00	2 4 - 20 mA Medium IIR 0.500 0.00 65535.00	•	Analogue Outputs configuration 0 1 Range: Disabled Scale Min.: 0.00 65535.00 Fault mode: Value Fault value: 0.00 0.00 Specifies the value to set in case of hardware f
Scale Min.: Scale Max.:	00000.0					

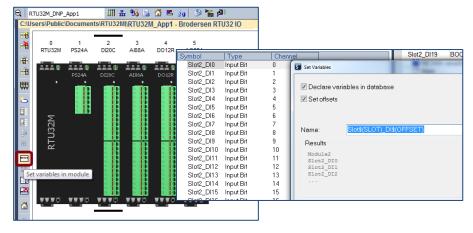
The 'Module not present' allows future module slots and reuse of similar setups (ie. not 'exactly' the same I/O).

#### Use the tag wizard to create variables

After creating a list of modules, variables are created for each I/O point using the 'Set Variables' icon.

Select/highlight each I/O module in turn, then select the Set Variables icon using syntax to identify each module and channel.

Starting with the DI20C in slot 2 Eg. Slot\$(SLOT)\_DI\$(OFFSET).



Then replace 'DI' with AI, DO and AO for modules 3-5, as shown below.

Symbol	Туре	Chan	nel	Slot2	Symbol Type			Slot:	Toymbol Type Toy	annel	S
Slot3_AI0	Analog Input	0	Set Variables		Slot4_D00 Output	utBit 0	Set Variables	_	Slot5_A00 Analog Output 0	Set Variables	
Slot3_Al1	Analog Input	1		(Base)	Slot4_D01 Output	utBit 1		The second second	Slot5_A01 Analog Output 1		
Slot3_Al2	Analog Input	2			Slot4_DO2 Output	utBit 2				Declare variables in c	Nataharo
Slot3_Al3	Analog Input	3	🗹 Declare variables in database		Slot4_DO3 Outpur	utBit 3	🗹 Declare variables in database				10100036
Slot3_Al4	Analog Input	4	Set offsets		Slot4_D04 Output	utBit 4	Set offsets			Set offsets	
Slot3_AI5	Analog Input	5			Slot4_D05 Output	utBit 5					
Slot3_Al6	Analog Input	6			Slot4_D06 Output	utBit 6					
Slot3_AI7	Analog Input	7			Slot4_D07 Output	utBit 7				Name Right?	LOT)_AO\$(OFFSE
			Name: Slot\$(SLOT)_AI\$(	DFFSET)	Slot4_D08_Output	utBit 8	Name: Slot\$(SLOT)_DO\$	(OFFSET)		Name: Slot\$(Sl	LUT)_AU\$(UFF3E

1 #

15

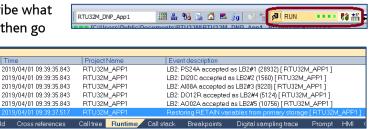
#### Download the initial 'I/O Box' Application

At this point the basic I/O box Application is ready to deploy/test. Select compile to first check the variables are all OK.



If any errors exist, the compiler warnings will describe what needs to be corrected. If the project compiles OK, then go online to download the RTU Application.

The Runtime TAB at the bottom of the WorkSuite display will show if the RTU32M modules have been correctly detected.



## Test the Application, add Module Status Info.

View I/O values from the module editor and/or variables list

The status/values of the I/O variables can be viewed online in both the I/O Editor window and the Variables List as shown here.

Slot5_AO0		<b>×</b>
12345		
	Force	
	Lock	
	Unlock	

Double click on one of the Output variables to change its state/value.

Slot1\_PS\_BusVoltageEnabled

Slot1\_PS\_OverloadWarning

Slot1 PS OverloadAlarm

Slot1 PS Voltsin

Slot1 PS Load

Slot1\_PS\_VoltsOut

Slot1 PS CurrentOut

Q	F	TU32M_DNP_App	p1 🏭	品 % %	🔂 🖻	🤞 🛛 🛸 🎽	RUN الأو		••• 🗱 👬 🖬	su 🐒 🥨 🛛				
	C:USers\Public\Documents\RTU32MU\RTU32M_App1 - Brodersen RTU32 IO													
	<u>اللہ</u>								🔺 🚞 (All)			Name	Value	Туре
	×	0	1 2	3	4	5			301A 🗍	RA		Slot2_DI13	TRUE	BOOL
		RTU32M P	S24A DI20C	A801A	DO12R	A002A			A00			Slot2_DI14	FALSE	BOOL
				_					DI1			Slot2_DI15	TRUE	BOOL
			S24A DI20C	AI08A	DO12R	AO02A			DI21			Slot2_DI16	FALSE	BOOL
		•	-524M D1200	- M100m	DO12R				DI21		=	Slot2_DI17	FALSE	BOOL
	<b></b>			R 🗐 🖪					DI21			Slot2_DI18	FALSE	BOOL
	5								DM	20A		Slot2_DI19	FALSE	BOOL
									DM			Slot3_Al0	6962.89	REAL
		Σ							DOI			Slot3_Al1	33266.5	REAL
		U32M	8 8						DO'			Slot3_Al2	15797.8	REAL
	∎+	, H							DO:			Slot3_Al3	65534	REAL
	+				1.11	2.10			101			Slot3_Al4	0	REAL
		[4] D012R (12	Relay Outputs)				_	н	101			Slot3_AI5	0	REAL
		PI 4						РИ	E PS2		-	Slot3_Al6	0	REAL
		Name	Value		Svm	bol		Tv	е	Channel		Slot3_AI7	0	REAL
	1	Art-No	DO12R		-	Slot2 DI1	8-FALSE	Inpu		18	<b>_</b>	Slot4_DO0	TRUE	BOOL
	×.	Description	12 Relay C	outputs						19		Slot4_DO1	FALSE	BOOL
	<u>6</u>											Slot4_DO2	FALSE	BOOL
	_					Slot3 Al0	=6962.89	Ana	log Input Val	0		Slot4_DO3	FALSE	BOOL
	35					Slot3 Al1	=33266.5		log Input Val			Slot4_DO4	FALSE	BOOL
						Slot3_Al2	=15797.8	Ana	log Input Val	2		Slot4_DO5	FALSE	BOOL
	2					Slot3 Al3	-65534	Ana	log Input Val	3		Slot4_DO6	TRUE	BOOL
	R					Slot3_Al4	=0		log Input Val		=	Slot4_DO7	FALSE	BOOL
						Slot3_Al5	=0	Ana	log Input Val	5	-	Slot4_DO8	FALSE	BOOL
						Slot3_Al6			log Input Val			Slot4_DO9	FALSE	BOOL
						Slot3 AI7			log Input Val			Slot4_DO10	FALSE	BOOL
					4				5.			Slot4_DO11	FALSE	BOOL
						Slot4_DO	0=TRUE	Out	out Bit	0		Slot5_AO0	12345	REAL
									out Bit	1	Ŧ	Slot5_AO1	0	REAL
					4			III			- F		ariablee	

¥ RTU32

PS24A

#### Additional variables for power supply status

Ō

Û.

0

Π

0

After proving the basic I/O functionality works as expected, take WorkSuite offline and add variables to show the status of the power supply module. Select the power supply module, then right click in the symbol area and select 'Insert/Set Variable'.

Double-click in the 'Type' field to select the various power supply status values and give each variable a name and select BOOL for status and WORD for analog values. The list below shows example power supply variables.

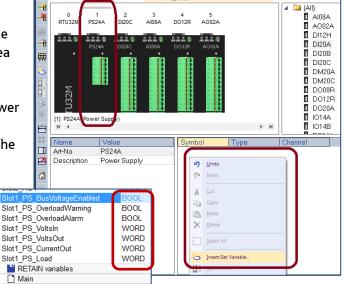
Current Overload Warning

Current Overload Alarm

Input Voltage

Load

Output Voltage Output Current



TU32M\\\RTU32M App1 - Brodersen RTU32 IC

A002A

DO12R

AI08A

#### Monitoring I/O Module Status and Version Information

The RTU32M I/O modules allow access to their status and version information. Data includes;

Module Type	Serial Number – High Bytes			
Operating Mode	Serial Number – Low Bytes			
Heartbeat Count	Internal Temperature			
Message Count	Power On Counter			
Hardware Revision	Watchdog Counter			
Firmware Revision	Up Time Since Restart			
Supply Voltage	Total Uptime Hours			

### Accessing and Storing I/O Module Status Info. Creating a Structured Tag Array

A 'structured tag' is the easiest way to view and store the I/O module status information. To create a structured tag, right click in the variables list and select 'Structures', or select 'Types>Structures' from the main program tree. The example structure here is named 'Module\_Info' and includes various unsigned integer and string variables.

Create a variable named 'RTU32M\_Module\_Status', with a type of Module\_Info. Create an array by giving the tag a dimension of 6.

To use structured tags and arrays the WorkSuite project settings must be set to include 'Complex Variables in a Separate Segment'. While making this change, also include 'Allocate status flags for variables with embedded symbols' and 'Embed symbols of all variables' (status flags are required to support manipulating of DNP point status bits and embedding of symbols allows referencing of variables by their tag name in HTML pages).

#### ST program to Access Module Info.

Right click the Programs folder and select 'Insert New Program' and create a ST program named 'RTU32M\_IO\_Status'. Open the program and right click in the variables area and create three local variables as shown here. Create code as shown below, compile and download the application. Check the logic reads the IO module information and updates the array.

RTU32M Module Status Module\_Info [0..5] Project settings C:\Users\Public\Documents\RTU32M\RTU32M General Runtime Compiler Debugging Advanced Communication parameters
Cycle time
Code Generation

📈 Version control	Version control						
Runtime system	Runtime system						
😫 Exchange IOs while	S Exchange IOs while stepping						
		1					
🔺 🔁 Programs	Programs						
🗗 Main							
🖻 RTU32M	IO Status						
Name	Type						
▲ C RTU32M IO Stat							
Inst LB2 INFO	LB2 INFO						
SerNo Str	STRING(255)						

DINT

On Line Change

Libraries
 Use external objects
 Code Checker rules

Version

IEC XML CS

Type

Dim.

IVa

92.168.0.1

Debug

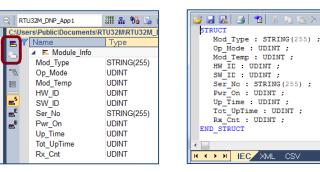
Edit Edit

Edit. endia

Disabled

V4-2019/0

I [C:		s\RTU32M\\\RTU32M_App1 - Brodersen RTU32 IO ]				
		I/O modules in slots 2-5	^ <b>Y</b>	Name	Type	C
		*************		Slot1 PS Load	WORD	
	3			RTU32M Module Status	Module Info	[[
	4 i := 2;		- L-	PETAIN variables		
		DC // loop through the IO modules by slot number		•	III	
	<pre>6 Inst_LB2_IN 7 SerNo Str :</pre>					_
		<pre>= HTOA(ANY_TO_DINT(Inst_LB2_INFO.SID2)) + ':'; // Extract the serial number = SerNo Str + HTOA(ANY TO DINT(Inst LB2 INFO.SID1));</pre>		ame	Value	
		<pre>le Status[i].0p Mode := ANY TO UDINT(Inst LB2 INF0.SIDI);</pre>		RTU32M_Module_Statute		
		<pre>le Status[i].0p_Hode .= ANY TO UDINT(Inst_LB2_INFO.OF);</pre>		.Mod_Type	'DI20C'	
		<pre>le Status[i].HW ID := ANY TO UDINT(Inst LB2 INFO.HWPID);</pre>		.Op_Mode	5	
		le Status[i].SW ID := ANY TO UDINT(Inst LB2 INFO.SWPID);		.Mod_Temp	41	
		le Status[i].Ser No := SerNo Str;		.HW ID	92980	
		le Status[i].Pwr On := ANY TO UDINT(Inst LB2 INFO.PwOnCnt);		.sw id	93010	
		le Status[i].Up Time := ANY TO UDINT(Inst LB2 INFO.UpTim);		.Ser No	'4629E8:E21;	21283'
1.1		le Status[i].Tot UpTime := ANY TO UDINT(Inst LB2 INFO.TotUpTim);		.Pwr On	393	
1.1	17 RTU32M Modu	le Status[i].Rx Cnt := ANY TO UDINT(Inst LB2 INFO.RXCNT);		.Up_Time	7	
1	18 CASE ANY T	O DINT(Inst LB2 INFO.MTYP) OF // Show the module type as a string value		.Tot_UpTime	2691	
1	19 0:	RTU32M_Module_Status[i].Mod_Type := 'N/A';			671	
1	20 1:	RTU32M_Module_Status[i].Mod_Type := 'MP32A';		.Rx_Cnt		
	21 28932:	RTU32M_Module_Status[i].Mod_Type := 'PS24A';		RTU32M_Module_Statu 		
	22 1560:	RTU32M_Module_Status[i].Mod_Type := 'DI20C';		.Mod_Type	'AI08A'	
	23 1580:	RTU32M_Module_Status[i].Mod_Type := 'DI20B';		.Op_Mode	5	
	24 1592:	RTU32M_Module_Status[i].Mod_Type := 'DI20A';		.Mod_Temp	39	
	25 1060:	RTU32M_Module_Status[i].Mod_Type := 'DI12H';		.HW_ID	92971	
	26 16408:	RTU32M_Module_Status[i].Mod_Type := 'DM20C';		.SW_ID	93014	
	27 16434:	RTU32M_Module_Status[i].Mod_Type := 'DM20A';		.Ser_No	'4629E8:E21;	22783'
	28 9220:	RTU32M_Module_Status[i].Mod_Type := 'AI08A';		.Pwr On	227	
	29 4612: 30 5124:	RTU32M_Module_Status[i].Mod_Type := 'D008R'; RTU32M_Module_Status[i].Mod_Type := 'D012R';		.Up_Time	8	
	31 5636:	RTU32M Module Status[i].Mod Type := 'D012R';		.Tot_UpTime	1456	
	32 10756:	RTU32M Module Status[i].Mod Type := 'A002A';		.Rx Cnt	796	
	33 16516:	RTU32M Module Status[i].Mod Type := 'IO14A';		In Content of Conte		
	34 16520:	RTU32M Module Status[i].Mod Type := 'IO14B';			'D012R'	
	35 16:	RTU32M Module Status[i].Mod Type := 'SPARE';		.Mod_Type		
	36 Else			.Op_Mode	5	
	37	RTU32M Module Status[i].Mod Type := 'UNKNOWN';		.Mod_Temp	35	
	38 END CASE;			.HW_ID	92984	
	39 END FOR;			.SW_ID	93012	
	-		-	.Ser_No	'01C82247:94	4192954



Name

5

### Create HMI Graphics to Monitor the RTU32M

#### Enhance the ST program logic for use in the graphics page

Create four new REAL variables to store scaled values for the power supply and four new BOOL variables to indicate module fault status. Create code as shown here to scale the power supply values and monitor the module operating mode.

42 // Map and scale Power Supply Info as REALS for HMI	Slot1_Supply_V	23.77
43 Slot1_Supply_V := ( ANY_TO_REAL ( Slot1_PS_VoltsIn ) / 100.0 );	Slot1_Output_V	11.959
44 Slot1_Output_V := ( ANY_TO_REAL ( Slot1_PS_VoltsOut ) / 1000.0 );	Slot1 Output mA	340
<pre>45 Slot1_Output_mA := ANY_TO_REAL ( Slot1_PS_CurrentOut );</pre>	Slot1 Load Percent	27
<pre>46 Slot1_Load_Percent := ANY_TO_REAL ( Slot1_PS_Load );</pre>	Slot2 Fail	FALSE
47	Slot3 Fail	FALSE
48	Slot4 Fail	FALSE
49 // Determine Module Operating Status for HMI	Slot5 Fail	FALSE
<pre>50 Slot2_Fail := ( RTU32M_Module_Status[2].Op_Mode &lt;&gt;5 ); 51 Slot3 Fail := ( RTU32M_Module_Status[3].Op_Mode &lt;&gt;5 );</pre>	RETAIN variables	TALOL
52 Slot4 Fail := ( RTU32M Module Status[4].Op Mode <>5 );	RETAIN Variables	
53 Slot5 Fail := ( RTU32M Module Status[5].0p Mode <>5 );	•	

### Create a RTU32M Hardware Overview graphic page

Right click on the Watch folder and select 'Insert New Item', then select Watch and Graphics. Name the new page 'Overview1'.

Use the graphics editor tools to add shapes and links to variables to create a Hardware Overview as shown below.

🔺 🗎 RTU32M_App1		53 Slot5 Fail
Exception programs	S Insert New Item	
🔺 🔁 Programs		
🗈 Main	Categories:	Available Items:
🔁 RTU32M_IO_Status	(All)	6a Spy list
🔺 🔁 Watch (for debugging)	Programs Watch	🖽 Recipe
SOVERVIEW1	Resources	Graphics
📉 Soft Scope	Embedded HMI	Soft Scope
Initial values	Shortcuts Others	E Test Sequence
🛂 Binding Configuration	Outers	

REAL

REAL REAL

REAL BOOL BOOL BOOL

BOOL

							· · ·	Name	Value
								Identifier	Edit box
	RTU32M Ha	rdware Ove	rview					Object type	Text input
								Variable symbol	Slot1_Supply_V
<b>•</b> •••••••••••••••••••••••••••••••••••								Spying delay	
	Slot 00	Slot 01	Slot 02	Slot 03	Slot 04	Slot 05	<b>.</b> .	Border size	2
	MP32A	PS24A	DI20A	AI08A	DO12R	AO02A		Border color	
	. ,			<u> </u>	↓			Border style	3DDOWN
+∞i			0		0			Data format	%s
-→⊚  ·			1	0 0				Background color	
<u> 2 </u>			2		1			Background mode	OPAQUE
<u>↔</u>			3	1 0		0		Text	23.77
			4		2			Text color	
↓ <u>11</u>			5	2 0				Text mode	NORMAL
Tai			6	3 0	3	1		Fontname	ARIAL
<b>P</b>			7	3 0		(0 -65535)		Font size	70
<b>P</b>			9			(0-00000)	· ·	Action	STATIC
					5		· ·	Minimum value	
<b>1</b>								Maximum value	
<b>E</b> 1		F1000-000-7	10		6			Direction	CENTER
2		Vii 23.77	11	4 0				Setgain	1.000000
		Vout 11.959	12	<b>c</b>	7			Set offset	0.000000
		11.959	13	5 0					
		mAout 358	15	6 0				Properties	Z Order Variables
			16	-	9			Bar Graph	
		Load% 29	17	7 0				▶ Binary	
			18		10			▶ Bitmap	
			19					▶ Charts	
					11			▶ Combo	
				1			L.	Embedded HMI (S	Simulation only)
		dule heading will	hange colour if	a module is miss	ing, or incorrect ty			▶ LED Displays	
			mange eviour ir	a module is illiss	ing, or incorrect ty	·····	· · ·	▶ Links	
							• •	Meters	

Go online with the RTU to check the control actions operate correctly and that the values displayed are correct.

## Published Graphics - Remote Monitoring

#### Export the graphics file for use with X5 Viewer or as HTML

Graphics pages created in WorkSuite can be exported for use in the X5 Viewer Tool, or as HTML files for use by web browsers.

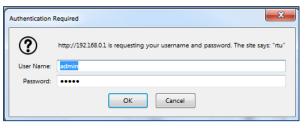


Generate HTML5 WEB page							
Page Generation Refresh rate:	1000 <u>A</u> ms						
Download Page							
Download Page on Target after Generation							
Settings:	192.168.0.1:502						
Driver:	K5NET5.DLL						
Folder:	www/pages/						
☑ Display Page after Download							
Import	OK Cancel						

#### Connect to the RTU – enter the User Name and Password

You are now able to connect to the RTU using you Smart Phone, Tablet or PC.

In the example below the module in Slot 2 was removed to show that the monitoring program created earlier is correctly indicating a module status error.



Slot 00       Slot 01       Slot 02       Slot 03       Slot 04       Slot 05         MP32A       PS24A       DI20A       Al08A       DO12R       AO02A         0       1       Slot 03       Slot 04       Slot 05         MP32A       PS24A       DI20A       Al08A       DO12R       AO02A         0       0       0       56313,1       2       35739.5       3       10       54321       3         Vin 23.77       1       0	۵	i 19	2.168.0.1/OVERVIEW1.	ntm	***	· 🖂 🕁					
MP32A       PS24A       DI20A       AI08A       DO12R       AO02A         0       15891.8       0       15891.8       1       0       12345         1       56313.1       2       35739.5       3       0       12345         2       35739.5       3       18011.7       4       0       5       5         Vin       23.77       11       4       0       7       0       8       0         Vin       23.77       11       5       0       8       0       9       0       1         Vin       23.77       11       5       0       8       0       9       0       1       1         Mout       364       15       6       0       9       0       1<	RTU32M Hardware Overview										
Vin       23.77         Vin       23.77         Vin       23.77         Vin       23.77         Vin       10         11       5         0       12         0       12345         0       1         0       1         0       1         0       1         0       1         0       1         0       1         10       1         11       1         11       1         11       1         11       1         11       1         12       1         13       1         14       1         15       0         16       1		Slot 00	Slot 01	Slot 02	Slot 03	Slot 04	Slot 05				
Vin 23.77       11       4       0       15891.8       1       1       1       1       12345         Vin 23.77       11       4       0       7       5       0       1         MAout 364       15       6       0       9       0       15891.8       1       1       1       1       12345       0       1       12345       0       1 <th></th> <th>MP32A</th> <th>PS24A</th> <th>DI20A</th> <th>AI08A</th> <th>DO12R</th> <th>AO02A</th>		MP32A	PS24A	DI20A	AI08A	DO12R	AO02A				
Module heading will change colour if a module is missing, or incorrect type			Vout 11.958 mAout 364 Load% 30	1         2         3         4         5         6         7         8         9         10         11         12         13         14         15         16         17         18         19	1       56313.1         2       35739.5         3       18011.7         4       0         5       0         6       0         7       0	1	0 54321 1 (0 -65535)				



#### Download the example RTU32M project

Download the example project file from the Brodersen FTP server using the link below.

http://download.brodersen.com/PETER\_KING/RTU32M\_App1.zip (~67kB)

To load the demo project in to WorkSuite, first ensure you have a suitable WorkSuite version (> 3.9.2.1160)

The link below downloads WorkSuite version 3.9.2.1218. Brodersen WorkSuite 3.9.2.1218 (~450MB).

The demo project can be loaded in to an existing Project List, or a new Project List can be created as shown below. Select File>New Project List. Select an appropriate destination folder and give the project list a name eg. 'Demo\_Apps'.

Then select File>Add Existing Project>From zip. Browse to the downloaded zip file and select it. You may be prompted to add items from the projects User Library – ignore this.

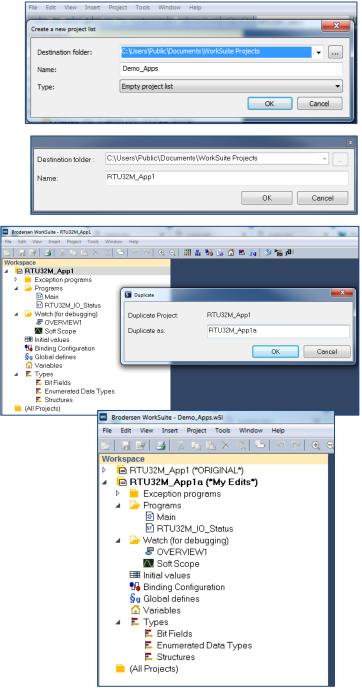
Once the file is loaded you can 'charge ahead' and start playing with the code and try it in your RTU (assuming you have the same I/O module layout).

If you have a different module layout you will need to edit the application.

It is recommended that you first duplicate the project, so that you have an original copy to refer back to, if you have difficulty making edits.

You can also edit the Project descriptions to make referencing similar projects easy...

Note: The demo project assumes the default RTU IP address of 192.168.0.1 is used.



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