CANopen Router User Manual

A-CANOR

Document No. D124-007 01/2020 Revision 1.3



CONTENTS

1. Pr	eface	5
1.1.	Intr	roduction to the CANopen Router5
1.2.	Fea	itures6
1.3.	Arc	hitecture6
1.4.	Ado	ditional Information10
1.5.	Sup)port10
2. In	stalla	tion11
2.1.	Мо	dule Layout11
2.2.	Мо	dule Mounting13
2.3.	CAI	Nopen and Power14
2.4.	Eth	ernet Port14
3. Se	etup	
3.1.	Inst	tall Configuration Software15
3.2.	Net	twork Parameters15
3.3.	Cre	ating a New Project19
3.4.	CAI	Nopen Router parameters22
3.5.	CAI	Nopen Master Mode26
3.	5.1.	CAN EDS File Management27
3.	5.2.	Adding CANopen Slave Devices29
3.	5.3.	General Configuration29
3.	5.4.	Mapping
	3.5.4	.1. EtherNet/IP Interface
	3.5.4	.2. Modbus TCP Interface
3.	5.5.	Parameterization
3.	5.6.	Device Discovery
3.6.	CAI	Nopen Slave Mode43
3.	6.1.	Virtual Device Map43
	3.6.1	.1. EtherNet/IP Interface45
	3.6.1	
3.7.	Мо	dule Download49
3.8.	Log	ix 5000 Configuration

	3.8	.1.	Add Module to I/O Configuration	52
	3.8	.2.	Importing UDTs and Mapping Routines	54
4.	Op	erati	ion	57
4	.1.	Logi	gix Message Routing	57
4	.2.	Logi	gix Assemblies	57
	4.2	.1.	Input Assembly	57
	4.2	.2.	Output Assembly	59
4	.3.	CIP	PMessaging	60
	4.3	.1.	SDO Passthrough	60
	Z	1.3.1.	.1. CIP Message:	60
	Z	1.3.1.	.2. Request Data:	61
	Z	1.3.1.	.3. Response Data:	61
	4.3	.2.	Slave Information	61
	Z	1.3.2.	2.1. CIP Message:	61
	Z	1.3.2.	2.2. Request Data:	62
	Z	1.3.2.	2.3. Response Data (when Command 0 was requested):	62
4	.4.	Mo	odbus Mapping	63
5.	Dia	gnos	stics	68
5	.1.	LED	Ds	68
5	.2.	Mo	odule Status Monitoring in Slate	69
5	.3.	Slav	ve Device Status Monitoring In Slate	77
5	.4.	CAN	Nopen Packet Capture	80
5	.5.	Mo	odule Event Log	82
5	.6.	We	eb Server	84
6.	Тес	hnic	cal Specifications	85
6	.1.	Dim	nensions	85
6	.2.	Elec	ctrical	86
6	.3.	Ethe	nernet	86
6	.4.	CAN	Nopen Network	87
6	.5.	Cert	rtifications	87
7.	Ind	ex		89

Revision History

Revision	Date	Comment
1.0	7 December 2018	Initial document
1.1	25 January 2019	Added Modbus Online in General Status Added Inhibit functionality
1.2	6 February 2019	Added parameter for inhibit when Communication is lost
1.3	8 January 2020	Added option to disable reading of Error register of Slave on connection establishment.
		Added option to force CANopen Router in Slave mode to startup in operational mode.

1. PREFACE

1.1. INTRODUCTION TO THE CANOPEN ROUTER

This manual describes the installation, operation, and diagnostics of the Aparian CANopen Router module. The CANopen Router, (hereafter referred to as the **module**,) provides intelligent data routing between either EtherNet/IP or Modbus TCP and the CANopen bus network. This allows the user to integrate CANopen devices into a Rockwell Logix platform (e.g. ControlLogix or CompactLogix) or any Modbus Master device with minimal effort.

The module can be configured to be either a CANopen Master or CANopen Slave allowing the user to not only integrate CANopen devices into a Logix or Modbus system, but to also allow the user to use Logix or Modbus devices in an existing CANopen network (by using the CANopen Router in Slave mode).

In a Logix system the module uses Direct-To-Tag technology allowing CANopen devices to exchange data with a Logix controller without the need to write any ladder or application code in Studio 5000.

The CANopen Router is configured using the Aparian Slate application. This program can be downloaded from <u>www.aparian.com</u> free of charge.



Figure 1.1. – Typical architecture using the CANopen Router

Slate allows the user to map up to 16 PDOs per CANopen Slave to Logix tags which will automatically be updated using Direct-To-Tag. When operating as a Modbus TCP Slave the module will provide various Modbus Holding registers to allow data exchange with a CANopen Slave.

Slate will also provide the user with the ability to change all parameters (using SDOs) of the slave based on the EDS file.

The module also provides a range of statistics to simplify the diagnostic process as well as a CANopen packet capture for remote diagnosis.

A built-in webserver provides detailed diagnostics of system configuration and operation, including the display of CANopen operation and communication statistics, without the need for any additional software.

1.2. FEATURES

- Module can operate as a CANopen Master or Slave.
- Module can interface to EtherNet/IP as well as Modbus TCP.
- Supports up to 64 CANopen Slaves (when in Master mode).
- Support for up to 16 PDOs (receive and transmit) per CANopen Slave.
- When using a Logix controller the module supports Direct-To-Tag so no Logix coding is required.
- Slate software provides a CANopen packet capture for better diagnosis of issues.
- Supports CiA 443 Bootloader Auto-enable.
- In Master Mode supports NMT message to initialize network.
- Time Synchronization of the CANopen network.
- Master supports SYNC for PDO communication.
- Supports all error and emergency (EMCY) messages and handling.
- Small form factor DIN rail mounted.

1.3. ARCHITECTURE

The figure below provides an example of the typical network setup for connecting various CANopen Slaves to a Logix controller via the CANopen Router.



Figure 1.2. – Typical network setup for connecting CANopen Slaves to a Logix Controller

The same applies for interfacing CANopen Slaves to a Controller using Modbus TCP (as shown below).



Figure 1.3. - Typical network setup for connecting CANopen Slaves to a Modbus Master

The next examples illustrate how the CANopen Router can be used as a CANopen Slave to allow Modbus devices and Logix controllers to integrate into an existing CANopen network.

Below is a typical network when the user is planning to use a Modbus device on an existing CANopen network using the CANopen Router.



Figure 1.4. – Modbus Device acting as a CANopen Slave via the CANopen Router

Below is a typical network when the user is planning to use a Logix controller on an existing CANopen network using the CANopen Router.



Figure 1.5. – Logix Controller acting as a CANopen Slave via the CANopen Router

1.4. ADDITIONAL INFORMATION

The following documents contain additional information that can assist the user with the module installation and operation.

Resource	Link
Slate Installation	http://www.aparian.com/software/slate
CANopen Router User Manual CANopen Router Datasheet Example Code & UDTs	http://www.aparian.com/products/CANopenRouter
Ethernet wiring standard	www.cisco.com/c/en/us/td/docs/video/cds/cde/cde205_220_420/installa tion/guide/cde205_220_420_hig/Connectors.html
CANopen Standards	https://www.can-cia.org/canopen/

Table 1.1. - Additional Information

1.5. SUPPORT

Technical support is provided via the Web (in the form of user manuals, FAQ, datasheets etc.) to assist with installation, operation, and diagnostics.

For additional support the user can use either of the following:

Resource	Link
Contact Us web link	www.aparian.com/contact-us
Support email	support@aparian.com

Table 1.2. – Support Details

2. INSTALLATION

2.1. MODULE LAYOUT

The module has two ports at the bottom of the enclosure as shown in the figure below. The ports are used for Ethernet and CANopen. The 5-way connector also provides power to the module. The Ethernet cable must be wired according to industry standards which can be found in the additional information section of this document.



Figure 2.1. – CANopen Router side and bottom view

The module provides three diagnostic LEDs as shown in the front view figure below. These LEDs are used to provide information regarding the module system operation, the Ethernet interface, and the CANopen interface.



Figure 2.2. – CANopen Router front and top view

The module provides four DIP switches at the top of the enclosure as shown in the top view figure above.

DIP Switch	Description
DIP Switch 1	Used to force the module into "Safe Mode". When in "Safe Mode" the module will not load the application firmware and will wait for new firmware to be downloaded. This should only be used in the rare occasion when a firmware update was interrupted at a critical stage.
DIP Switch 2	This will force the module into DHCP mode which is useful when the user has forgotten the IP address of the module.
DIP Switch 3	Reserved
DIP Switch 4	Applies the 120Ω terminating resistor across the CAN network (switched between Can-H and Can-L).
	NOTE: When the module is at the start or the end of the CANopen network the terminator must be switched on.

Table 2.1. - DIP Switch Settings

2.2. MODULE MOUNTING

The module provides a DIN rail clip to mount onto a 35mm DIN rail.



Figure 2.3 - DIN rail specification

The DIN rail clip is mounted on the bottom of the module at the back as shown in the figure below. Use a flat screw driver to pull the clip downward. This will enable the user to mount the module onto the DIN rail. Once the module is mounted onto the DIN rail the clip must be pushed upwards to lock the module onto the DIN rail.



Figure 2.4 - DIN rail mouting

2.3. CANOPEN AND POWER

A five-way CANopen connector is used to connect the CANopen CAN bus network as well as the Power+, Power- (GND), and earth. The module requires an input voltage of 10 - 28Vdc. **Refer** to the technical specifications section in this document.



Figure 2.5 – CANopen and Power connector



NOTE: Although the CANopen Router supports the CiA443 objects, the CANopen interface is not fault-tolerant.

2.4. ETHERNET PORT

The Ethernet connector should be wired according to industry standards. **Refer** to the additional information section in this document for further details.

3. SETUP

3.1. INSTALL CONFIGURATION SOFTWARE

All the network setup and configuration of the module is achieved by means of the Aparian Slate device configuration environment. This software can be downloaded from http://www.aparian.com/software/slate.



Figure 3.1. - Aparian Slate Environment

3.2. NETWORK PARAMETERS

The module will have DHCP (Dynamic Host Configuration Protocol) enabled as factory default. Thus, a DHCP server must be used to provide the module with the required network parameters (IP address, subnet mask, etc.). There are a number of DHCP utilities available, however it is recommended that the DHCP server in Slate be used.

Within the Slate environment, the DHCP server can be found under the Tools menu.



Figure 3.2. - Selecting DHCP Server

Once opened, the DHCP server will listen on all available network adapters for DHCP requests and display their corresponding MAC addresses.

hac address	Vendor	Requests	Elapsed	Assigned IP	Assign	Status	Identity
0:35:21:AB:2C	Aparian	2	1		Assign	Discover	

Figure 3.3. - DHCP Server

i

NOTE: If the DHCP requests are not displayed in the DHCP Server it may be due to the local PC's firewall. During installation, the necessary firewall rules are automatically created for the Windows firewall. Another possibility is that another DHCP Server is operational on the network and it has assigned the IP address.

To assign an IP address, click on the corresponding "Assign" button. The IP Address Assignment window will open.

MAC Address	Vendor	Requests	Elapsed	Assigned IP	Assign	Status	Identity
00:60:35:21:AB:2C	Aparian	16	2		Assign	Discover	
		IP Address	168 . able Static (1 41	Recent	223	
				Ok	Cancel	1	

Figure 3.4. - Assigning IP Address

The required IP address can then be either entered, or a recently used IP address can be selected by clicking on an item in the Recent List.

If the "Enable Static" checkbox is checked, then the IP address will be set to static after the IP assignment, thereby disabling future DHCP requests.

Once the IP address window has been accepted, the DHCP server will automatically assign the IP address to the module and then read the Identity object Product name from the device.

The successful assignment of the IP address by the device is indicated by the green background of the associated row.

	venuor	Requests	Elapsed	Assigned IP	Assign	Status	Identity
0:60:35:21:AB:2C	Aparian	262	6	192.168.1.41	Assign	Set Static	CANopen Router

Figure 3.5. - Successful IP address assignment

It is possible to force the module back into DHCP mode by powering up the device with DIP switch 2 set to the On position.

A new IP address can then be assigned by repeating the previous steps.



NOTE: It is important to return DIP switch 2 back to Off position, to avoid the module returning to a DHCP mode after the power is cycled again.

If the module's DIP switch 2 is in the On position during the address assignment, the user will be warned by the following message.



Figure 3.6. - Force DHCP warning

In addition to the setting the IP address, a number of other network parameters can be set during the DHCP process. These settings can be viewed and edited in Slate's Application Settings, in the DHCP Server tab.

Once the DHCP process has been completed, the network settings can be set using the Ethernet Port Configuration via the Target Browser.

The Target Browser can be accessed under the Tools menu.



Figure 3.7. - Selecting the Target Browser

The Target Browser automatically scans the Ethernet network for EtherNet/IP devices.



Figure 3.8. - Target Browser

Right-clicking on a device, reveals the context menu, including the Port Configuration option.



Figure 3.9. - Selecting Port Configuration

All the relevant Ethernet port configuration parameters can be modified using the Port Configuration window.

pe							Speed / Duplex Conf	figuration		
	M	Netho	d	DHC	D	~	Auto-negotiate			
							O Manual			
							Manual Configur	ration		
192	. 1945	168	12	1	0	247	Port Speed	100	~	
255	5 4 5	255	94	255	42	0	Duplex	Full Dup!	ex 🗸	
0	.	0		0		0				
0	1 26	0	12	0	(0	0	General			
0		0		0	40	0	MAC Addrose	00.60.35.1E	-A·F0	
							MAC Address	00.00.00.11	1.20	
							TCP Inactivity Time	out 120	(s)	
	192 255 0 0 0	192 . 255 . 0 . 0 .	Methor 192 . 168 255 . 255 0 . 0 0 . 0 0 . 0	Method 192 . 168 . 255 . 255 . 0 . 0 . 0 . 0 . 0 . 0 . 0 . 0 .	Method DHC 192 168 1 255 255 255 0 0 0 0 0 0 0 0 0 0 0	Method DHCP 192 . 168 . 1 . 255 . 255 . 255 . 0 . 0 . 0 . 0 . 0 . 0 .	Method DHCP 192 168 1 247 255 255 255 0 0 0 0 0 0 0 0 0 0 0 0 0	Method DHCP Auto-negotiate 192 168 1 247 255 255 255 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Method DHCP Auto-negotiate Manual 192 168 1 247 255 255 255 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 </td <td>Method DHCP Auto-negotiate 192 168 1 247 255 255 255 0 0 0 0 0 <t< td=""></t<></td>	Method DHCP Auto-negotiate 192 168 1 247 255 255 255 0 0 0 0 0 <t< td=""></t<>

Figure 3.10. - Port Configuration

Alternatively, these parameters can be modified using Rockwell Automation's RSLinx software.

3.3. CREATING A NEW PROJECT

Before the user can configure the module, a new Slate project must be created. Under the File menu, select New.

File	Device	To	ols	Windo	w	Help
	New		2	+ 🗗	4	₽.¢
-	Open					
22	Close					
	Save					
	Save As					
	Recent	۲				
	Exit					

Figure 3.11. - Creating a new project

A Slate project will be created, showing the Project Explorer tree view. To save the project use the Save option under the File menu.

A new device can now be added by selecting Add under the Device menu.



Figure 3.12. - Adding a new device

In the Add New Device window select the CANopen Router and click the Ok button.

mage	Device Name	 Description 	
	CANOpen Router	CANOpen Communication Module	
	Cell Connect	Cellular Text and Data Transfer Module	
	ControlNet Router	ControlNet to EtherNet/IP Module	
	DeviceNet Router	DeviceNet to AB-Ethernet PCCC Module	
	DF1 Router	DF1 to Logix Communication Module	
	DH485 Router	DH485 to Logix Communication Module	
1	DHCP Manager	Managed DHCP Module	

Figure 3.13 – Selecting a new CANopen Router

The device will appear in the Project Explorer tree as shown below, and its configuration window opened. The device configuration window can be reopened by either double clicking the module in the Project Explorer tree or right-clicking the module and selecting *Configuration*.

CANOpenR - Config	uration	
General CAN Bus L	gix Modbus Virtual Device Map (Disabled)	
Instance Name	CANOpenR	
Description		
IP Address	0 . 0 . 0 . 0 . Major Revision 1 v	
Router Mode	Master 🗸	
Primary Interface	EtherNet/IP ~	
	Ok Apply Cancel	

Figure 3.14. – CANopen Router configuration

Refer to the additional information section in this document for Slate's installation and operation documentation.

3.4. CANOPEN ROUTER PARAMETERS

The CANopen Router parameters are configured using Slate. **Refer** to the additional information section for documentation and installation links for Aparian Slate. The CANopen Router parameter configuration consists of a general configuration, CAN Bus configuration, Logix, Modbus, and Virtual Device Map (for operating as a CANopen slave). When downloading this configuration into the module it will be saved in non-volatile memory that persists when the module is powered down.



NOTE: When a firmware upgrade is performed, the module will clear all CANopen Router configuration and routing maps.

The general configuration is shown in the figure below. The CANopen Router general configuration window is opened by either double clicking on the module in the tree or right-clicking the module and selecting *Configuration*.

neral CAN Bus Lo	gix Modbus Virtual Device Map (Disabled)	
Instance Name	CANOR01	
Description	Demo CANopen Router project	
IP Address	192 168 1 . 247 Major Revision 1 ~	
Router Mode	Master V	
Primary Interface	EtherNet/IP V	

Figure 3.15. - General Configuration

The general configuration consists of the following parameters:

Parameter	Description
Instance Name	This parameter is a user defined name to identify between various CANopen Router modules.

Description	This parameter is used to provide a more detail description of the application for the module.
IP Address	The IP address of the target module. The user can use the target browse button to launch the target browser to the select the CANopen Router on the network.
Major Revision	The major revision of the module
Router Mode	Master
	The CANopen Router will operate as a CANopen Master on the CANopen network.
	Slave
	The CANopen Router will operate as a CANopen Slave on the CANopen network.
Primary Interface	EtherNet/IP
	The CANopen Router will be an EtherNet/IP target and exchange data with a Logix controller using Class 1 and Class 3 communication.
	Modbus TCP Slave
	The CANopen Router will be a Modbus TCP Slave and can exchange data with a Modbus TCP Master.

Table 3.1 - General configuration parameters

The CAN Bus configuration is shown in the figure below. The CAN Bus configuration window is opened by either double clicking on the module in the tree or right-clicking the module and selecting *Configuration*.

BAUD Rate	125k	~	SYNC
Node Address	1	~	COB ID 0x80 Interval 0 (ms) Counter Enable
Heartbeat Interval	0	(ms)	ТІМЕ
Slave Inactive Timeou	t 10000 Fail	(ms)	COB ID 0x100 Interval 0 (ms)
Slave Mode Opera	tional on Start-I	Jp	Emergency COB ID 0x0

Figure 3.16 – CAN Bus Configuration

The CANopen Communication configuration consists of the following parameters:

Parameter	Description
BAUD Rate	The CANopen bus BAUD rate. The following options are available:

	• 50k
	• 125k
	• 250k
	• 500k
	• 800k
	• 1M
Node Address	The module's node address on the CANopen bus network.
Heartbeat Interval	This is the rate (in milliseconds) at which the CANopen Router will send out heartbeat messages on the CANopen bus.
	To disable sending of CANopen heartbeat messages the user can set this value to zero.
Slave Inactive Timeout	The amount of time (in milliseconds) elapsed since the last communication from a specific CANopen slave before the CANopen Router will indicate that the device is offline.
	(Master Mode Only)
Inhibit On Comms Fail	The parameter will force the CANopen Router to inhibit the CANopen communication when either EtherNet/IP communication is lost, or Modbus TCP communication is lost.
Slave Mode Operational on Start-Up	When this is set the module (in slave mode) will start up in operational mode vs the normal pre-operational.
	(Slave Mode Only)
SYNC	COB ID
	The base address to use when sending and receiving SYNC messages on the CANopen network.
	Interval
	This is the rate (in milliseconds) at which the CANopen Router will send out SYNC messages on the CANopen bus. To disable sending of CANopen SYNC messages the user can set this value to zero.
	Counter Enable
	This will enable the optional parameter used to define and explicit relationship between the current SYNC cycle and PDO transmission.
TIME	COB ID
	The base address to use when sending and receiving TIME messages on the CANopen network.
	Interval
	This is the rate (in milliseconds) at which the CANopen Router will send out TIME messages on the CANopen bus. To disable sending of CANopen TIME messages the user can set this value to zero.
Emergency	COB ID
	The base address to use when sending and receiving EMCY messages on the CANopen network.

Table 3.2 – CANopen Communication parameters

The Logix configuration is shown in the figure below. The Logix configuration window is opened by either double clicking on the module in the tree or right-clicking the module and selecting *Configuration*.

CANOpenR - Configurat	on				
General CAN Bus Logix	Modbus Virtual Device Map	Disabled)			
Logix Configuration					
Logix Path	192.168.1.129.1.0				
ENIP Retry Limit	3 [0-5]				
ENIP TimeOut	1000 ms				
		Ok	Apply	Cancel	

Figure 3.17 Logix Configuration

The Logix configuration (used for Class 3 Direct-To-Tag communication) consists of the following parameters:

Parameter	Description
Logix Path	The CIP path to the Logix controller which will be used to exchange data with the various CANopen devices. The user can use the target browse button to launch the target browser to the select the Logix controller on the network.
ENIP Retry Limit	The amount of EtherNet/IP retries the module will make once no response was received from the Logix Controller.
ENIP Timeout	The time in milliseconds after which a retry is sent. Once the first retry is sent the next retry will be sent after the same amount of time. This will repeat until the ENIP Retry Limit is reached.

Table 3.3 – Logix parameters

The Modbus configuration is shown in the figure below. The Modbus configuration window is opened by either double clicking on the module in the tree or right-clicking the module and selecting *Configuration*.

eneral CAN Bus Logix M	dbus Virtual Device Man (Disabled)			
nicial of the additional cogin	That borree hap (blobbid)			
Modbus				
Local Node	3			
REAL / DINT Format	AA BB CC DD v			
Inactivity Timeout	2000 (ms)			
	Ok	Apply Cance	1	

Figure 3.18 – Modbus Configuration

The Modbus Communication configuration consists of the following parameters:

Parameter	Description			
Local Node	he Modbus Node address assigned to the CANopen Router.			
REAL / DINT Format	For a Real (single floating point) number this setting shows the format of t data will be presented when using a Modbus Primary Interface.			
	The format (byte re-ordering) options are as follows:			
	AA BB CC DD			
	• BB AA DD CC			
	• DD CC BB AA			
	CC DD AA BB			
Inactivity Timeout	The amount of time that no Modbus requests have been received before the CANopen Router indicates that the connection to the Modbus Master is no longer intact.			

Table 3.4 – Modbus parameters

3.5. CANOPEN MASTER MODE

The module can be configured to operate as a CANopen Master on the CANopen network (see the *General Configuration*).

3.5.1. CAN EDS FILE MANAGEMENT

Each CANopen slave device has an EDS file that is required to provide information needed to configure the device for data exchange. Slate manages the CANopen EDS library which is used for adding devices to the CANopen Router in Master mode.

The EDS File Management Tool is opened by selecting *CAN EDS File Management* under the Tool menu in the configuration utility.



Figure 3.19 – Launching the CAN EDS File Management Tool

Once the tool has been opened a list of slave devices already registered using their CAN EDS files.

er Vendor	Product	Cod	8		Filename		
(All)	×	*	*		*	Res	iet
Vendor	Product Name	Vendor Id	Code	Revision	EDS File	Rx PDOs	Tx PDOs
parian Inc.	CANopen Router	0x0	0x7D	1.0001	A-CANOR.eds	4	4
parian CiA443.eds	Aparian Universal CiA443	0x0	0x0	0.0001	Aparian CiA443.eds	10	10
CPDAS CO. LTd.	ICPDAS CANopen Slave Devic	e 0x13C	0x2053	0.0002	CAN_2053C.eds	10	10

Figure 3.20 – CAN EDS File Management Tool

To add an EDS file the user will need to select the *Add* option under the EDS File menu.

Catalog	EDS	File
	I	View
Filter	+	Add
Ve	X	Delete

Figure 3.21 – CAN EDS File Adding

The required CAN EDS file will need to be selected as shown below:

5 Select a CAN EDS File						×
← → ∨ ↑ 📕 « Docu	ments > Aparian	~ (Search Aparian			٩
Organize 🔹 New folder			1 1 1 1 1 1 1 1	E 🔹		?
📒 Email attachmer ^	Name	Date modified	Туре	Size		
Pictures Public	ZAN_2053C.eds	3/18/2013 3:06 AM	EDS File		24 KB	
 This PC 3D Objects Desktop Documents Downloads Music Pictures Videos 						
🍋 OS (C:) 🗸 🗸						
File name:	CAN_2053C.eds		CAN EDS (*.eds)		Cancel	~

Figure 3.22 – CAN EDS File Adding

Once the file has been selected the CAN EDS File Management tool will add the slave device to the device list and recompile the CAN EDS catalog.

A CAN EDS catalog can be exported from another Slate by exporting the CAN EDS catalog on one Slate and importing it in another. This is done by selecting either *Import* or *Export* under the Catalog menu as shown below:

Cat	alog	EDS	File	
4	Rebu	ild		
G,	Impo	rt		
¢	Expo	rt		-
	Close		Ļ	

Figure 3.23 – CAN EDS Catalog importing

3.5.2. Adding CANOPEN SLAVE DEVICES

The user will need to add each CANopen slave device to the CANopen Router which can then be configured. This is done by right-clicking on the *Slave Devices* item in the tree and selecting *Add CANopen Device*.



Figure 3.24 – Adding a CANopen Slave Device

3.5.3. GENERAL CONFIGURATION

The General configuration is shown in the figure below. The Device General configuration window is opened by either double clicking on the slave device in the tree or right-clicking the slave device and selecting *Configuration*.

entity		CANopen Configuration
Node Address	2 ~	State Retrieval Heartbeat ~
Instance Name	ICPDASCANopenSla	State Retrieval Interval 5000 (ms)
EDS Filename	CAN_2053C.eds	Response Timeout 100 (ms)
		Receive SDO COB ID 0x600
		Transmit SDO COB ID 0x580
		Emergency COB ID 0x80
		Boot Loader Program 0 ~
		Disabled Error Register Read

Figure 3.25 – Device General configuration parameters

The General configuration consists of the following parameters:

Parameter	Description		
Node Address	The node addre network.	ess of the CANopen Slave device on the CANopen	
Instance Name	The device instar	nce name.	
EDS Filename	Filename of the I	EDS file used for this slave device.	
State Retrieval	This is the methor Disabled	od used to retrieve the operational state of the device.	
	The CANopen R device to send it	outer will not attempt or expect the CANopen slave s operational state.	
	Heartbeat		
	The CANopen slave device will automatically (without the need request from the Master) send its operational state.		
	í	NOTE: In many field devices the Heartbeat has been disabled by default. The user will need to go to the device parameters (see <i>Diagnostics</i> section) and change the <i>Producer Heartbeat Time</i> (parameter 1017) to a non-zero value.	
	Guarding		
	The CANopen Ro device at the <i>Sta</i>	outer will manually request the operational state of the termination to the termination of termination of the termination of terminat	

State Retrieval Interval	When using Stat interval at wh operational stat	When using State Retrieval method <i>Guarding</i> , this parameter will be the interval at which the CANopen Router will manually retrieve the operational state of the CANopen slave device.		
Response Timeout	The amount of t the CANopen CANopen Route	time the CANopen Router will wait before it will flag that slave device did not response to request from the er.		
Received SDO COB ID	This is the bas CANopen Slave	e COB ID used for receiving SDO messages from the device.		
	í	NOTE: Slate will automatically add the CANopen slave device node address to this base value (so the user must enter the base address e.g. 0x600). As an example, for node address 2 the configured Receive SDO SOB ID will be 0x602 by default.		
Transmit SDO COB ID	This is the base CANopen Slave	COB ID used for transmitting SDO messages from the device.		
	í	NOTE: Slate will automatically add the CANopen slave device node address to this base value (so the user must enter the base address e.g. 0x580). As an example, for node address 2 the configured Transmit SDO SOB ID will be 0x602 by default.		
Emergency COB ID	This is the bas messages from	se COB ID used for transmitting Emergency (EMCY) the CANopen Slave device.		
	í	NOTE: Slate will automatically add the CANopen slave device node address to this base value (so the user must enter the base address e.g. 0x80). As an example, for node address 2 the configured Emergency SOB ID will be 0x82 by default.		
Boot Loader Program	When using the Router to autor CANopen Slave	e sub-sea profile (CiA 443), this will allow the CANopen natically set the bootloader program to be used by the on start-up.		
Disabled Error Register Read	When this is e attempt to rea connection esta	nabled the CANopen Router in Master mode will not d the Error Register (0x1001) of the Slave device on blishment.		

Table 3.5 – Device General configuration parameters

3.5.4. MAPPING

The module can be configured to exchange data between either a Logix controller or Modbus Master and various CANopen Slave devices.

When the primary interface is EtherNet/IP, the CANopen Router will allow the user to read data from a CANopen Slave device into a Logix controller and/or write data to a CANopen Slave device from a Logix controller.

When the primary interface is Modbus TCP Slave, the CANopen Router will allow the user to read data from a CANopen Slave device into a Modbus Holding Register and/or write data to a CANopen Slave device from a Modbus Holding Register.

lannin	a (may of 16 itor	nc)									
	Function	Transmission Type	Update Interval	MPDO	MPDO Address	MPDO Index	MPDO Sub Index	Data Type	Element Count	Target Tag	Browse
**	~	~						~			



Parameter	Description
Function	There are two functions supported for mapping field device PDOs (process variables).
	TPDO x
	TPDOs are the PDOs received from the CANopen Slave device. A total of four TPDOs can be used if multiplexing is not used (see MPDO section). Each PDO received from the Slave device can be up to 8 bytes (e.g. two 32-bit Reals).
	RPDO x
	RPDOs are the PDOs sent to the CANopen Slave device. A total of four RPDOs can be used if multiplexing is not used (see MPDO section). Each PDO sent to the Slave device can be up to 8 bytes (e.g. two 32-bit Reals).
Transmission Type	Sync (RPDO only)
	The CANopen Router in Master mode will send out the PDO data to the CANopen slave once the SYNC packet has been sent.
	RemoteTxReq (TPDO only)
	The CANopen Router will request the PDO from the CANopen slave at the update interval. If the slave device has been configured to automatically send out the PDOs, it is recommended to set the update interval at 2 x the rate the PDOs will be sent.
	Evt-Timer (RPDO only)
	The CANopen Router in Master mode will send out the PDO data to the CANopen slave every Update Interval.
	Evt-Logix (RPDO only)

	The CANopen Router in Master mode will send out the PDO data to the CANopen slave every time the relevant PDO bit in the <i>SlaveOutputTriggers</i> of the Logix output assembly or Modbus Holding Register.				
Update Interval	The time (in milliseconds) at which the PDOs will be requested (when transmission type is <i>RemoteTxReq</i>) or at which the PDOs will be sent (when transmission type is <i>Evt-Timer</i>).				
MPDO	Each PDO can be multiplexed (if supported by the slave device) to have multiple process variables associated with it. With normal PDOs each PDO has a maximum of 8 bytes while with multiplexed PDOs each multiplexed process variable has maximum of 4 bytes. To enable Multiplexing the user must select the MPDO checkbox in the mapping of the PDO.				
	MPDO Address				
	The address of the process variable in the PDO.				
	MPDO Index				
	The index of the process variable in the PDO.				
	MPDO Sub Index				
	The sub index of the process variable in the PDO.				
Data Type	The data type to be used when copying to/from the Logix Tag or Modbus Holding Register.				
Element Count	The number of elements to be used for the specific PDO. For example, the user can have 2 x 32-bit real values or 8 x 8-bit integers.				
	(i) NOTE: The element count must be such that the element count multiplied by the data type size must not be greater than 8 bytes when not using multiplexing and 4 bytes when using multiplexing.				
Target Tag	When the Primary Interface is EtherNet/IP, this parameter will be the Logix Tag that will be used to exchange data with the specific CANopen slave device. The target tag can either be entered manually or if online with the controller the target tag can be updated using the target browser (see figure below).				

Table 3.6 – Device Mapping parameters

5 Logix Tag Browser	- D	×
⋧ 🗉 🗖		
Tagname	Data Type	^
H CANOR02_Real2	REAL[2]	
-CANOR02ConnSts	DINT	
E CANOR02ConnStsErrCnt	COUNTER	
E CANORM_Reals	REAL[2]	
CANReal2	REAL[2]	
CANReal2_0	REAL[2]	
CANReal2_1	REAL[2]	
-CANRealVal0	REAL	
-CANRealVal1	REAL	
CANSlaveDI16	INT	
H CANSlaveDINT	DINT[2]	
+ CANSlaveReal	REAL[2]	

Figure 3.27 – Target Tag selection

3.5.4.1. ETHERNET/IP INTERFACE

When using the EtherNet/IP interface, the TPDO data from the CANopen Slave device will be written into the Target Tag specified in the mapping, and the RPDO data sent to the CANopen Slave device will be read from the Target Tag specified in the mapping.



Figure 3.28 – Process variable (TPDO) from slave device to Target Tag



Figure 3.29 – Process variable (RPDO) from Target Tag to slave device



NOTE: The user must ensure that the selected Logix tag is sufficiently large to accommodate the specified PDO. For example, if the PDO returns two REAL values, the Logix Target Tag cannot be only one REAL.



NOTE: If there are duplicate mapping items in the mapping list then only the first mapped item (of all the duplicates) will be executed.

3.5.4.2. MODBUS TCP INTERFACE

When Modbus TCP has been selected as the primary interface, the process variables (TPDOs) from the CANopen Slave device will be stored in predefined Modbus Holding Register. The process variable (RPDOs) that will be sent to the CANopen Slave device will also be read from the predefined internal Modbus Registers.


Figure 3.30 – Process variable (TPDO) from slave device to Modbus Holding Register

In the above example the PDO value will be written to Modbus Holding Register 1408. This is calculated as follow:

Modbus Holding Register

Slave Device start- 1000 Slave Device 5 - 1000 + (Slave Mapping Index * 100) = 1400 Slave Device 5 PDO 3 - 1400 + ((PDO -1) * 4) = 1408



NOTE: There will be a 100-register gap between consecutive field devices. See the Modbus Mapping section for more details.



NOTE: Every PDO will consume four Modbus Holding Registers, because the max PDO size is 8 bytes which equals 4 Modbus words.

NOTE: The PDO offset in the Modbus Holding Register will depend on its location in the Mapping. For example, if a Slave Node with TPDO 3 at slave mapping index 4 (as shown below) will be at Modbus Holding Register 1416 (1400 + (5 - 1) * 4). The offset in the Modbus Holding registers is independent of the TPDO number (e.g. TPDO **2**).

			ems.)									
	Function		Transmission Type	n	Update Interval	MPDO	MPDO Address	MPDO Index	MPDO Sub Index	Data Ty	pe	Element Count
	TPDO 1	~	RemoteTxReq	~						Int-16	~	1
	RPDO 1	~	Evt - Logix	~						Int-16	~	1
	RPDO 2	~	Evt - Logix	~						Int-16	~	1
	RPDO 3	~	Evt - Logix	~						Int-16	~	1
	TPDO 2	~	RemoteTxReq	~						Int-16	~	1
*		~		~							~	

Figure 3.31 – Process variable (TPDO) at mapping index 5

NOTE: When sending process variables to the field device, the same example and calculation as above applies.

NOTE: To optimise the Modbus communication it is recommended to group all the TPDOs together and then all the RPDOs.



NOTE: The user will need to ensure that when writing to the CANopen Router Modbus Holding Registers that the registers holding data from the device are not inadvertently overwritten.

3.5.5. PARAMETERIZATION

Each field device provides a range of parameters that can be accessed using the SDO communication parameters of the CANopen Slave device. This will allow the user to view and (with certain parameters) change the settings in the slave device. To access the slave device parameters the user will need to open the *Status* window of the slave device and select the parameters tab (as shown below):

Downlo	ad to Router Write	Enabled Parameters						Auto Upd	ate
Index	Parameter	Live Value	Status	Store	Store V	Hex	Data Type	Access	^
1000	Device Type	0			0		UInt-32	ReadOnly	
-1001	Error Register	0			0		UInt-8	ReadOnly	
1018	Identity Object								
1003	pre-defined error field								
-1005	COB-ID SYNC	0			0		UInt-32	ReadWrite	-
-1008	Manufacturer device name	0			0		Vis-Str	Constant	
-1009	Manufacturer hardware ve	rsion 0			0		Vis-Str	Constant	
-100A	Manufacturer software vers	sion 0			0		Vis-Str	Constant	
-100C	Guard Time	0			0		UInt-16	ReadWrite	
-100D	life time factor	0			0		UInt-8	ReadWrite	
1010	store parameters								
1011	restore default parame	ters							
-1014	COB-ID Emergency Messa	ge O			0		UInt-32	ReadWrite	
-1017	Producer Heartbeat Time	1000	Ok.		0		UInt-16	ReadWrite	
1200	Server SDO Parameter								
1400	Receive PDO 0 Parame	eter							
1401	Receive PDO 1 Parame	ator							

Figure 3.32 – Slave device parameters

These parameters will be listed from the EDS file used to instantiate the CANopen Slave device. If the user wants to read a value from a specific parameter, right-click on the parameter and select *Refresh* (as shown below). This will read the parameter from the CANopen Slave and update it in the *Live Value* column.

neral Map I	tems Parameters								
Downlo	ad to Router Write Enab	led Parameters						Auto Upd	late
Index	Parameter	Live Value	Status	Store	Store V	Hex	Data Type	Access	^
-1000	Device Type	0			0		UInt-32	ReadOnly	
-1001	Error Register	0			0		UInt-8	ReadOnly	
1018	Identity Object								
1003	pre-defined error field								
-1005	COB-ID SYNC	0			0		UInt-32	ReadWrite	
-1008	Manufacturer device name	0			0		Vis-Str	Constant	
-1009	Manufacturer hardware version	0			0		Vis-Str	Constant	
-100A	Manufacturer software version	0			0		Vis-Str	Constant	
-100C	Guard Time	0			0		UInt-16	ReadWrite	
-100D	life time factor	0			0		UInt-8	ReadWrite	
1010	store parameters								
1011	restore default parameters								
-1014	COB-ID Emergency Message	0			0		UInt-32	ReadWrite	
1017	Producer Heartbeat Time	1000	Ok.		0		UInt-16	ReadWrite	
1200	Server SDO Paramete	terresh							
1400	Receive PDO 0 Param	Aodify							
+ 1401	Receive PDO 1 Parameter								

Figure 3.33 – Slave device parameters - Read

When the Access to a device is *ReadWrite*, then the value can be modified by the user. This is done by right-clicking on the value and selecting *Modify*. The user will be able to enter the new value into the textbox which will then be downloaded into the slave device.

neral Map I	tems Parameters								
Downlo	ad to Router Write Enable	ed Parameters						Auto Upd	late
Index	Parameter	Live Value	Status	Store	Store V	Hex	Data Type	Access	^
- 1000	Device Type	0			0		UInt-32	ReadOnly	
-1001	Error Register	0			0		UInt-8	ReadOnly	
1018	Identity Object								
1003	pre-defined error field								
-1005	COB-ID SYNC	0			0		UInt-32	ReadWrite	
-1008	Manufacturer device name	0			0		Vis-Str	Constant	
-1009	Manufacturer hardware version	0			0		Vis-Str	Constant	
-100A	Manufacturer software version	0			0		Vis-Str	Constant	
-100C	Guard Time	0			0		UInt-16	ReadWrite	
-100D	life time factor	0			0		UInt-8	ReadWrite	
1010	store parameters								
1011	restore default parameters								
-1014	COB-ID Emergency Message	0			0		UInt-32	ReadWrite	
1017	Producer Heartbeat Time	Refresh	Ok.		0		Ulnt-16	ReadWrite	
1200	Server SDO Parameter	Modify							
1400	Receive PDO 0 Paramet	mouny							
1401	Receive PDO 1 Parameter								5

Figure 3.34 – Slave device parameters – Modify

Downlo	ad to Router Write Er	nabled Parameters						Auto Upo	late
Index	Parameter	Live Value	Status	Store	Store V	Hex	Data Type	Access	^
- 1000	Device Type	0			0		UInt-32	ReadOnly	
-1001	Error Register	0			0		UInt-8	ReadOnly	
1018	Identity Object								
1003	pre-defined error field								
-1005	COB-ID SYNC	0			0		UInt-32	ReadWrite	
-1008	Manufacturer device name	0			0		Vis-Str	Constant	
-1009	Manufacturer hardware versi	ion					s-Str	Constant	
-100A	Manufacturer software version	on S Set '	Value				× s-Str	Constant	
-100C	Guard Time	040-0					nt-16	ReadWrite	
-100D	life time factor	De		ale at Tim			Int-8	ReadWrite	
1010	store parameters	PR	Joucer nea	ribeat rim	le				
1011	restore default paramete	rs							
-1014	COB-ID Emergency Message	e Va	lue		1000		nt-32	ReadWrite	
-1017	Producer Heartbeat Time	1					nt-16	ReadWrite	
1200	Server SDO Parameter								
1400	Receive PDO 0 Paramete	эг	Ok		Canc	-ol			
1401	Receive PDO 1 Paramete	er			ound	CI			2

Figure 3.35 – Slave device parameters – Set Value



NOTE: The parameters in the slave device will not automatically be saved to non-volatile memory. The user will need to write the required codes to the

store parameters (parameter 1010) index to force the CANopen Slave device to store the updated parameters to NV memory. The user will require to write 0x65766173 to either 1010.1, 1010.2, or 1010.3 (as shown below). Alternatively, the user can use the *Send Store Parameters* in the Slave device status (see the *Diagnostics* section).

eneral Map Ite	ms Parameters								
Downloa	d to Router Write Enable	ed Parameters						Auto Upd	ate
Index	Parameter	Live Value	Status	Store	Store V	Hex	Data Type	Access	^
□ 1000	Device Type	0			0	1.1.1	UInt-32	ReadOnly	
-1001	Error Register	0			0		UInt-8	ReadOnly	
1018	Identity Object								
1003	pre-defined error field								
-1005	COB-ID SYNC	0			0		UInt-32	ReadWrite	
-1008	Manufacturer device name	0		501 - 002	0		Vis-Str	Constant	
- 1009	Manufacturer hardware version	0			0		Vis-Str	Constant	
-100A	Manufacturer software version	0			0		Vis-Str	Constant	
-100C	Guard Time	0			0		UInt-16	ReadWrite	
-100D	life time factor	0			0		UInt-8	ReadWrite	
1010	store parameters								
-1010.0	largest subindex supported	0			0		UInt-8	ReadOnly	
- 1010.1	save all parameters	0			0		UInt-32	ReadWrite	
-1010.2	save communication parameters	0			0		UInt-32	ReadWrite	
-1010.3	save application parameters	0			0		UInt-32	ReadWrite	
1011	restore default parameters								
-1014	COB-ID Emergency Message	0			0		UInt-32	ReadWrite	
7.017	D.I. II. II. IT.	0		E	0		10.1.10	D 04/3	×

Figure 3.36 – Slave device parameters – Save Parameters

The user can also select to Auto-Update the values in the parameter list by selecting the autoupdate checkbox (as shown below). Once set the values will automatically start updating.

neral Map	tems Parameters									
Downic	ad to Router	Write Enable	ed Parameters	5)					Auto Upd	late
Index	Parar	neter	Live Value	Status	Store	Store V	Hex	Data Type	Access	^
-1000	Device Type		0			0		UInt-32	ReadOnly	
-1001	Error Register		0	Ok.		0		UInt-8	ReadOnly	
1018	Identity Object									
1003	pre-defined en	ror field								
-1005	COB-ID SYNC		128	Ok.		0		UInt-32	ReadWrite	
-1008	Manufacturer der	vice name	CAN-205	Ok.		0		Vis-Str	Constant	
-1009	Manufacturer ha	dware version	1.3	Ok.		0		Vis-Str	Constant	
-100A	Manufacturer sof	tware version	1.30-201	Ok		0		Vis-Str	Constant	
-100C	Guard Time		0	Ok		0		UInt-16	ReadWrite	
-100D	life time factor		0			0		UInt-8	ReadWrite	
1010	store paramete	ers								
1011	restore default	parameters								
-1014	COB-ID Emergen	cy Message	0			0		UInt-32	ReadWrite	
-1017	Producer Heartb	eat Time	1000	Write Ok		0		UInt-16	ReadWrite	
1200	Server SDO Pa	rameter								
1400	Receive PDO () Parameter								
1401	Receive PDO	Parameter								

Figure 3.37 – Slave device parameters – Auto Update

3.5.6. DEVICE DISCOVERY

The device discovery function scans the CANopen network and displays all the devices found on the network. This is done by opening the module status form and selecting the *Discovery* tab. Slate will start scanning the CANopen network for slave devices once the Start Discovery button has been pressed (see below).

	CANOpen Statistics	CAN Statistics IV	loopus Stausucs	Live List Di	CIP Statistic	s Ethemet Citerits TC	F / ABF
							Start Discovery
Node	Vendor Id	Product Code	e Profile	Revision	Serial Number	EDS Filename	Status
2	0x13C	0x2053	401	4.002	0x6CD3683C	CAN_2053C.eds	Online
6	0x0	0x7B	0	1.002	0xEAFA1F35		Online - EDS Misma
				_	Canaci		
					ouncer		

Figure 3.38 – Device Discovery

Once all the devices have been found the user will be able to add any of the devices to the CANopen Router Slave devices tree. This is done by right-clicking on the device in the discovery list and selecting *Add Device* (as shown below).

	NOpen Stati	stics (JAN Statistics	Modbus Statistics	Live List Di	CIP Statistic	s Ethernet Clients I CH	P/ARP
								Start Discovery
Node	Vendor	ld	Product Cod	de Profile	Revision	Serial Number	EDS Filename	Status
2	0-13	ic .	0×2053	401	4.002	0x6CD3683C	CAN_2053C.eds	EDS: Product
6	+	Add	Device	0	1.002	0xEAFA1F35		Online - EDS Misma.
54	+.	Add	ALL Devices	443	2.432	0x13BDE	Aparian CiA443.eds	Online

Figure 3.39 – Device Discovery - Add

3.6. CANOPEN SLAVE MODE

The module can be configured to operate as a CANopen Slave on the CANopen network (see the *General Configuration*). The user will be able to map up to 16 PDOs per CANopen Router when operating as a Slave on the CANopen network.

3.6.1. VIRTUAL DEVICE MAP

The mapping for the CANopen Router when operating as a CANopen Slave will be done through the Virtual Device Map (as shown below).

14 0		Manual D	auton Man							
ral	CAN Bus Logix	Modbus Virtual D	evice Map							
ave	Tag Mapping (ma	x. of 16 items.)								
	PDO	Transmission Type	Update Interval	MPDO	MPDO Address	MPDO Index	MPDO Sub Index	Data Type	Element Count	
	~	~				101000000		~		

Figure 3.40 – CANopen Router as Slave – PDO Mapping

When the primary interface is EtherNet/IP, the CANopen Router will allow the user to receive data from a CANopen Master and write it into a Logix controller and/or send data to a CANopen Master from a Logix controller.

When the primary interface is Modbus TCP Slave, the CANopen Router will allow the user to received data from a CANopen Slave device and write it into a Modbus Holding Register and/or send data to a CANopen Master from a Modbus Holding Register.

Parameter	Description
Function	There are two functions supported for mapping PDOs (process variables) for the CANopen Router when operating as a CANopen slave.
	TPDO x
	TPDOs are the PDOs sent to the CANopen Master. A total of four TPDOs can be used if multiplexing is not used (see MPDO section). Each PDO received from the Slave device can be up to 8 bytes (e.g. two 32-bit Reals).
	RPDO x

	RPDOs are the PDOs received from the CANopen Master. A total of four RPDOs can be used if multiplexing is not used (see MPDO section). Each PDO sent to the Slave device can be up to 8 bytes (e.g. two 32-bit Reals).
	NOTE: The definitions for the TPDO and RPDO are swapped when the CANopen Router is operating as a CANopen slave. When operating as a CANopen Slave, the TPDO and RPDO are from a field device's perspective.
Transmission Type	Sync (TPDO only)
	The CANopen Router in Slave mode will send out the PDO data to the CANopen Master once a SYNC packet has been received.
	RemoteTxReq (RPDO only)
	The CANopen Router will receive the PDO from the CANopen Master.
	Evt-Timer (TPDO only)
	The CANopen Router in Slave mode will send out the PDO data to the CANopen Master every Update Interval.
	Evt-Logix (TPDO only)
	The CANopen Router in Slave mode will send out the PDO data to the CANopen Master every time the relevant PDO bit in the <i>SlaveModeOutputTriggers</i> of the Logix output assembly or Modbus Holding Register.
Update Interval	The time (in milliseconds) at which the PDOs will be sent (when transmission type is <i>Evt-Timer</i>).
MPDO	Each PDO can be multiplexed (if supported by the CANopen Master) to have multiple process variables associated with it. With normal PDOs each PDO has a maximum of 8 bytes while with multiplexed PDOs each multiplexed process variable has maximum of 4 bytes. To enable Multiplexing the user must select the MPDO checkbox in the mapping of the PDO.
	MPDO Address
	The address of the process variable in the PDO.
	MPDO Index
	The index of the process variable in the PDO.
	MPDO Sub Index
	The sub index of the process variable in the PDO.
Data Type	The data type to be used when copying to/from the Logix Tag or Modbus Holding Register.
Element Count	The number of elements to be used for the specific PDO. For example, the user can have 2 x 32-bit real values or 8 x 8-bit integers.
	NOTE: The element count must be such that the element count multiplied by the data type size must not be greater than 8 bytes when not using multiplexing and 4 bytes when using multiplexing.

Target Tag	When the Primary Interface is EtherNet/IP, this parameter will be the Logix Tag that will be used to exchange data with the CANopen Master. The target tag can either be entered manually or if online with the controller the target tag can be updated using the target browser (see
	figure below).

Table 3.7 – Device Mapping parameters

5 Logix Tag Browser		×
≈ 🗉 🗖		
Tagname	Data Type	^
CANOR02 Real2	REAL[2]	
CANOR02ConnSts	DINT	
GANOR02ConnStsErrCnt	COUNTER	
E CANORM_Reals	REAL[2]	
CANReal2	REAL[2]	
E CANReal2_0	REAL[2]	
E CANReal2_1	REAL[2]	
-CANRealVal0	REAL	
-CANRealVal1	REAL	
CANSlaveDI16	INT	
H CANSlaveDINT	DINT[2]	
CANSlaveReal	REAL[2]	

Figure 3.41 – Target Tag selection

3.6.1.1. ETHERNET/IP INTERFACE

When using the EtherNet/IP interface, the RPDO data from the CANopen Master will be written into the Target Tag specified in the mapping, and the TPDO data sent to the CANopen Master will be read from the Target Tag specified in the mapping.



Figure 3.42 – Process variable (RPDO) from CANopen Master to Target Tag





 \triangle

NOTE: The user must ensure that the selected Logix tag is sufficiently large to accommodate the specified PDO. For example, if the PDO returns two REAL values, the Logix Target Tag cannot be only one REAL.

3.6.1.2. MODBUS TCP INTERFACE

When Modbus TCP has been selected as the primary interface, the process variables (TPDOs) from the CANopen Router will be read from a predefined Modbus Holding Register. The process variables (RPDOs) that is received from the CANopen Master will be written to predefined internal Modbus Registers.



	PDO		Transmissi Type	on	Update Interval	MPDO	MPDO Address	MPDO Index	MPDO Sub Index	Data Ty	pe	Elemen Count
	TPDO 1	~	Evt - Timer	~	1000	10-10				Real-32	~	1
	TPDO 2	V	Evt - Timer	~	1000					Real-32	~	1
Г	TPDO 3	V	Evt - Timer	~	1000					Real-32	~	1

Figure 3.44 – Process variable (TPDO) from Modbus Holding Register to CANopen Master

In the above example the PDO value will be written to Modbus Holding Register 608. This is calculated as follow:

Modbus Holding Register Slave Mode start– 600 Slave Mode PDO 3 – 600 + ((PDO -1) * 4) = 608



NOTE: Every PDO will consume four Modbus Holding Registers, because the max PDO size is 8 bytes which equals 4 Modbus words.

i

NOTE: The PDO offset in the Modbus Holding Register will depend on its location in the Mapping. For example, if TPDO 3 at mapping index 5 (as shown

below) will be at Modbus Holding Register 616 (600 + (5 – 1) * 4). The offset in the Modbus Holding registers is independent of the TPDO number (e.g. TPDO **2**).

Sla	ve Tag Mappin	g (m	nax. of 16 items.)									
	PDO		Transmissior Type	ו	Update Interval	MPDO	MPDO Address	MPDO Index	MPDO Sub Index	Data Ty	pe	Elemen Count
	TPDO 1	~	Evt - Timer	\sim	1000					Real-32	~	1
	RPDO 1	~	RemoteTxReq	\sim						Real-32	~	1
	RPDO 2	~	RemoteTxReq	\sim						Real-32	~	1
-	RPDO 3	~	RemoteTxReq	\sim						Real-32	~	1
Г	TPDO 2	~	Evt - Timer	~	1000					Real-32	~	1
**		~		~							~	

Figure 3.45 – Process variable (TPDO) at mapping index 5

NOTE: When receiving process variables from a CANopen Master, the same example and calculation as above applies.

(i)

NOTE: To optimise the Modbus communication it is recommended to group all the TPDOs together and then all the RPDOs.

NOTE: The user will need to ensure that when writing to the CANopen Router Modbus Holding Registers that the registers holding data from the device are not inadvertently overwritten..

3.7. MODULE DOWNLOAD

Once the CANopen Router configuration has been completed, it must be downloaded to the module.

Before downloading the Connection Path of the module should be set. This path will automatically default to the IP address of the module, as set in the module configuration. It can however be modified, if the CANopen Router is not on a local network.

The Connection path can be set by right-clicking on the module and selecting the Connection Path option.



Figure 3.46 - Selecting Connection Path

The new connection path can then be either entered manually or selected by means of the Target Browser.

CANOpenR - Co	onnection Path		
Connection Path			44. 1 44
192.168.1.247			Browse
	Ok	Cancel	

Figure 3.47 - Connection Path

To initiate the download, right-click on the module and select the Download option.



Figure 3.48 - Selecting Download

Once complete, the user will be notified that the download was successful.



Figure 3.49 - Successful download

Within the Slate environment the module will be in the Online state, indicated by the green circle around the module. The module is now configured and will start operating immediately.



Figure 3.50 - Module online

3.8. LOGIX 5000 CONFIGURATION

3.8.1. ADD MODULE TO I/O CONFIGURATION

When the module operates in a Logix "owned" mode the CANopen Router will need to be added to the Logix 5000 I/O tree. The module will need to be added as a generic Ethernet module. This is done by right clicking on the Ethernet Bridge in the Logix 5000 and selecting *New Module* after which the *ETHERNET-MODULE* is selected to be added as shown in the figure below.



NOTE: See the next section for importing the configuration (L5X).



Figure 3.51 - Add a Generic Ethernet Module in Logix 5000

The user must enter the IP address of the CANopen Router that will be used. The assembly instance and size must also be added for the input, output, and configuration in the connection parameters section. Below are the required connection parameters.

Connection Parameter	Assembly Instance	Size
Input	125	91 (32-bit)
Output	126	68 (32-bit)
Configuration	102	0 (8-bit)

Table 2.9 Legivelace	connection	noromotors for	the CANenen	Doutor
Table 3.8 - Logix class 1	L connection	parameters for	the CANopen	Router

Conn	ection module into				
Type:	ETHERNET-MODULE Generic Ethernet M	lodule			
Vendor:	Rockwell Automation/Allen-Bradley				
Parent:	eth				
Name:	CANOB01	Connection Parar	neters		
Description:			Assembly Instance:	Size:	
		Input:	125	91	(32-bit)
	×	Output:	126	68	(32-bit)
Comm Format	: Data-DINT 🗸 🗸	Configuration:	102	0	(8-bit)
Address / Ho	ost Name				
IP Addre	ss: 192 . 168 . 1 . 247	Status Input:			
) Host Nar	ne.	Status Output;			

Figure 3.52 - Logix General module properties in Logix 5000



NOTE: The user will need to enter the exact connection parameters before the module will establish a class 1 connection with the Logix controller.

Next the user needs to add the connection requested packet interval (RPI). This is the rate at which the input and output assemblies are exchanged. The recommended value is 100ms. Refer to the technical specification section in this document for further details on the limits of the RPI.

Modul ieneral	Connection	Keth (ETHERNE	I-MODULE 1.1)		
Reque	sted Packet I	nterval (RPI):	100.0 🌲 ms	(1.0 - 3200.0 ms)	
🛄 Inhil	bit Module	-			
🗌 Maj	or Fault On Co	ontroller If Connec	tion Fails While in P	lun Mode	
🔽 Use	e Unicast Con	nection over Ethe	rNet/IP		
Modu	ile Fault				

Figure 3.53 - Connection module properties in Logix 5000

Once the module has been added to the Logix 5000 I/O tree the user must assign the User Defined Types (UDTs) to the input and output assemblies. The user can import the required UDTs by right-clicking on *User-Defined* sub-folder in the *Data Types* folder of the IO tree and selecting *Import Data Type*. The assemblies are then assigned to the UDTs with a ladder copy instruction (COP) as shown in the figure below.



Figure 3.54 – Logix 5000 I/O module tree

3.8.2. IMPORTING UDTS AND MAPPING ROUTINES

To simplify the mapping of the input image, a Logix 5000 Routine Partial Import (L5X) file is provided.

This file can be imported by right-clicking on the required Program and selecting the Import Routine option.

⊢ 🔂 Tasks │ 🕂 🛱 MainT │ 🕂 🛱 Ma	ask inPro	ogram	
		New Routine	
		Import Routine	
Unsch			
🖨 🔤 Motion G	X	Cut	Ctrl+X

Figure 3.55 – Logix 5000 Importing CANopen Router specific routine and UDTs

Lookin	CANopen Rou	iter	00	🥗 🛄 v	
LOOK III.	- crittoponnici	^			
	Name		Status	Date modified	Туре
Quick access	CANOpenRo	uterRoutineAndUDTs.L5X	\odot	12/10/2018 10:51	Logix Des
Desktop					
-					
Libraries					
This PC					
	1				
Network		(
	File name:	CANOpenRouterRoutineAndU	DTs.L5X	~	Import
	Files of type:	RSLogix 5000 XML Files (* L5X)	~	Cancel
	Files containing:	Routine		~	Help
		<u>~</u>			

Figure 3.56 - Selecting partial import file

The import will create the following:

- The required UDTs (user defined data types)
- Two controller tags representing the Input and Output assemblies.
- A routine mapping the CANopen Router module to the aforementioned tags.

The user may need to change the routine to map to the correct CANopen Router module instance name, and make sure that the mapping routine is called by the Program's Main Routine.



Figure 3.57 - Imported Logix 5000 objects

Refer to the additional information section of this document for an example Logix 5000 project as well as the required UDTs.

4. OPERATION

4.1. LOGIX MESSAGE ROUTING

Once the module has been configured correctly, the CANopen Router will be ready to send and receive the configured PDOs on the CANopen network and route the data to and from the selected Logix tags. Once the PDO is successfully sending or receiving data **PDOxOk** bit in the respective Slave input assembly will be set. Refer to the diagnostics section of this document for a more detailed explanation of the various indicators that can be used to diagnose the module.

4.2. LOGIX ASSEMBLIES

When the module operates in a Logix "owned" mode the Logix controller will establish a class 1 cyclic communication connection with the CANopen Router. An input and output assembly is exchanged at a fix interval (RPI). The UDTs provided will convert the input and output arrays into tag-based assemblies. Refer to the additional information section in this document for the input and output UDTs.



NOTE: If communication to the Logix controller is lost, then (when in Master mode) the CANopen Router will force the CANopen network to the pre-operational state.

4.2.1. INPUT ASSEMBLY

The following parameters are used in the input assembly of the module.

Parameter	Datatype	Description
InstanceNameLen	DINT	This parameter is the instance name length of the module that was configured under the general CANopen Router configuration in Slate.
InstanceName	SINT[16]	This parameter is the instance name of the module that was configured under the general CANopen Router configuration in Slate.
Status.ConfigValid	BOOL	Set if a valid configuration is executing in the module.
Status.DuplicateNode	BOOL	Set if a duplicate node is detected on the network.
Status.NetworkOperational	BOOL	The current state of the CANopen network is operational.

Status.NetworkPreOperational	BOOL	The current state of the CANopen network is pre- operational.
Status.NetworkStopped	BOOL	The current state of the CANopen network is stopped.
Status.MasterMode	BOOL	The CANopen Router is operating as a CANopen Master.
Status.SlaveMode	BOOL	The CANopen Router is operating as a CANopen Slave.
Status.MBOnline	BOOL	This is reserved for EtherNet/IP interface.
Status.Inhibited	BOOL	Module CANopen sending and receiving has been inhibited.
TransactionRate	DINT	The transaction rate is the number of CANopen messages per second that the module is currently routing.
DeviceTemperature	REAL	The internal temperature of the CANopen Router module.
UTCTime	DINT[2]	The UTC time on the CANopen network. This has already been formatted for Logix and can be viewed in LINT – Date/Time format.
RxCANCount	DINT	Received CAN message count.
TxCANCount	DINT	Transmitted CAN message count.
CrcErrCanCount	DINT	CAN CRC failed message count.
BitErrCanCount	DINT	CAN Bit error count.
StuffErrCanCount	DINT	CAN Stuff error count.
PdoTxCount	DINT	The number of PDO packets transmitted.
PdoRxCount	DINT	The number of PDO packets received.
SdoTxCount	DINT	The number of SDO packets transmitted.
SdoRxCount	DINT	The number of SDO packets received.
TimePcktCount	DINT	The number of TIME packets received or sent.
SyncPcktCount	DINT	The number of SYNC packets received or sent.
EmergencyPcktCount	DINT	The number of EMCY packets received or sent.
HeartbeatPcktCount	DINT	The number of Heartbeat packets received.
TagReads	DINT	The total number of Logix tag reads executed by the module.
TagWrites	DINT	The total number of Logix tag writes executed by the module.
ConnectionFailures	DINT	The number of failed class 3 connection attempts.

		Note: Logix tag reading and writing requires the module to first establish a class 3 connection with the Logix Controller.
TagErrors	DINT	The number of failed tag access (read/write) requests.
		These may include privileged violations, non- existing tags, etc.
Slave[x]	AparianCANopenRouterNode[64]	A total of 64 CANopen slaves can be mapped into the Logix input assembly. The below structure will be repeated for each mapped CANopen Slave.
		SlaveAddress
		The node address of the mapped slave on the CANopen network.
		Online
		When the last response received from the slave is less than the <i>Slave Inactive Timeout</i> parameter in the CAN Bus configuration, the slave is considered online, and this bit is set.
		ErrorReceived
		Set when the last EMCY message received from the slave has an error.
		PdoError
		Set if one of the PDOs are not operating correctly.
		Initializing
		Set when the slave is in the initialize state.
		Stopped
		Set when the slave is in the stopped state.
		Operational
		Set when the slave is in the operational state.
		PreOperational
		Set when the slave is in the pre- operational state.
		For PDO (may 16) has a bit to indicate that it is
		operating as expected.

Table 4.1 - Logix 5000 input assembly parameters

4.2.2. OUTPUT ASSEMBLY

The following parameters are used in the output assembly of the module.

Parameter	Datatype	Description
GenOperation.NetworkPreOperational	BOOL	When the CANopen Router is the CANopen Master, this bit will force the CANopen network to be PreOperational.

		NOTE: When other NetworkPreOperational and NetworkStop bits are not set then the CANopen Router will set the network state to Operational.
GenOperation.NetworkStop	BOOL	When the CANopen Router is the CANopen Master, this bit will force the CANopen network to be Stopped.
		NOTE: When other NetworkPreOperational and NetworkStop bits are not set then the CANopen Router will set the network state to Operational.
GenOperation.Inhibit	BOOL	Inhibit the CANopen communication.
UTC	DINT[2]	When the CANopen Router is a CANopen Master, the user can write the Logix WallClock time to the UTC tag which will be converted into the CANopen time format for when sending TIME messages.
SlaveModeOutputTriggers.TxPDO x Trigger	BOOL[16]	When the CANopen Router is operating as a CANopen Slave, these bits are used to trigger sending of TPDOs when the Transmission Type is Evt – Logix . Each time the bit is toggle (either from 1 to 0 or from 0 to 1) the respective PDO will send the data to the CANopen Master.
SlaveOutputTriggers[x].TxPDO y Trigger	BOOL[16]	When the CANopen Router is operating as a CANopen Master, these bits are used to trigger sending of PDOs to the CANopen Slave device when the Transmission Type is Evt – Logix . Each time the bit is toggle (either from 1 to 0 or from 0 to 1) the respective PDO will send the data to the CANopen Slave device.

Table 4.2 - Logix 5000 output assembly parameters

4.3. CIP MESSAGING

The CANopen Router will allow the user to extract certain information from CANopen Slave devices using CIP messages. Below are the required parameters for SDO parameter extraction from the slave device as well as operational data from the slave device.

4.3.1. SDO PASSTHROUGH

4.3.1.1. CIP MESSAGE:

Parameter	Description
Service Code	0x65 (Hex)
Class	0x417 (Hex)
Instance	1
Attribute	N/A

Request Data Length

9 - 489

Table 4.3 – SDO Passthrough Message

4.3.1.2. REQUEST DATA:

Parameter	Data Type	Description
Node	SINT	The Node Address of the CANopen Slave
Function	SINT	0 – Upload from Slave 1 – Download to Slave
Index	INT	SDO Parameter Index
Sub-index	SINT	SDO Parameter Sub-index
Timeout	INT	The time in milliseconds if not response was received before the request will timeout.
Data Length	INT	The length of the data to follow below (when doing a upload the data length will be zero).
Data	SINT[0-480]	The data to be sent when doing a download.

Table 4.4 – SDO Passthrough Request

4.3.1.3. RESPONSE DATA:

Parameter	Data Type	Description
Status	SINT	This is the status of the request.
		0 – Success
		1 – Failed
		2 – Timeout
Data Length	SINT	The length of the data returned.
Data	SINT[]	The data from the SDO request. The number of bytes will be equal to the Data Length in the response.

Table 4.5 – SDO Passthrough Response

4.3.2. SLAVE INFORMATION

4.3.2.1. CIP MESSAGE:

|--|

Service Code	0x66 (Hex)
Class	0x417 (Hex)
Instance	1
Attribute	N/A
Request Data Length	2

Table 4.6 – Slave Information Message

4.3.2.2. REQUEST DATA:

Parameter	Data Type	Description
Node	SINT	The Node Address of the CANopen Slave
Command	SINT	The command to be sent to the CANopen Slave device.
		0 – Return the status of the Slave.
		1 – Reset Node
		2 – Set Mode to Operational
		3 – Set Mode to Pre-Operational (PDOs will stop communicating)
	4 – Set Mode to Operational (PDOs and SDOs will stop communicating)	

Table 4.7 – Slave Information Request

4.3.2.3. RESPONSE DATA (WHEN COMMAND 0 WAS REQUESTED):

Parameter	Data Type	Description
Status	SINT	Bit 0 – Online Bit 1 – Error Received Bit 2 to 7 - Reserved
State	SINT	0 – Initializing 1 – Disconnected 2 – Connecting 3 – Preparing 4 – Stopped 5 – Operational 126 – Unknown 127 – Pre-operational
Inactive Time	INT	Time since the last communication received from the slave.
Last Error Code	INT	The code of the last error received (refer to the Slave device user manual for a description of the error codes).
Last Error Type	SINT	The type of the last error received. Bit 0 - Generic Bit 1 - Current

		Bit 2 - Voltage Bit 3 - Temperature Bit 4 - Communication Bit 5 - Device Profile Specific Bit 6 - Reserved Bit 7 - Manufacturer Specific
PDO		Repeat the below for each PDO (max 16)
Flags	SINT	Flags Bit 0 - Configured Flags Bit 1 - TransactionOk Flags Bit 2 - Transmit PDO Flags Bit 3 - Receive PDO Flags Bit 4 to 7 - Reserved
Transaction Count	DINT	The number of transactions completed.
Timeout	INT	The number of times the PDO receiving has timed out.

Table 4.8 –	Slave	Information	Response
-------------	-------	-------------	----------

4.4. MODBUS MAPPING

When the primary interface of the CANopen Router is set to Modbus TCP, the CANopen Router will operate as a Modbus TCP Slave supporting the following Modbus registers.



NOTE: If communication to the Modbus Master is lost, then (when in Master mode) the CANopen Router will force the CANopen network to the pre-operational state.

Register Type:	Holding Registers			
Parameter Byte Date Register		Description		
Mastar Status	Length	Туре		
	1	1		This parameter is the instance name
				Inis parameter is the instance name
			_	length of the module that was
Instance Name Length	4	DINT	0	configured under the general
				CANopen Router configuration in
				Slate.
	16	SINT[16]	2	This parameter is the instance name
Instance Name				of the module that was configured
				under the general CANopen Router
				configuration in Slate.
Status				
Bit 0 – Configuration Valid				
Bit 1 – Duplicate Node		DINT		
Bit 2 – Network Operational	4		10	
Bit 3 – Network Pre-Operational				
Bit 4 – Network Stopped				

Bit 5 – Master Mode				
Bit 6 – Slave Mode				
Bit 7 – Modbus Online				
Bit 8 - Inhibited				
Bit 0 to 21 - Pasarvad				
Bit 9 to 31 - Reserved				
Transaction Pata	4		12	The transaction rate is the number of
	4	DINT	12	the module is currently routing
				The internal temperature of the
Device Temperature	4	REAL	14	CANopen Router module.
	8		16	The UTC time on the CANopen
	0		10	network.
RxCANCount	4	DINT	20	Received CAN message count.
TxCANCount	4	DINT	22	Transmitted CAN message count.
CrcErrCanCount	4	DINT	24	CAN CRC failed message count.
BitErrCanCount	4	DINT	26	CAN Bit error count.
StuffErrCanCount	4	DINT	28	CAN Stuff error count.
PdoTxCount	4	DINT	30	The number of PDO packets
DdoDyCount	4		22	transmitted.
Puorxcount	4	DINI	52	The number of SDO packets
SdoTxCount	4	DINT	34	transmitted.
SdoRxCount	4	DINT	36	The number of SDO packets received.
				The number of TIME packets received
TimePcktCount	nepcktCount 4 L	DINI	38	or sent.
SyncPektCount	Δ		40	The number of SYNC packets received
Synci excedunt	-	DINI		or sent.
EmergencyPcktCount	4	DINT	42	The number of EMCY packets
				received or sent.
HeartbeatPcktCount	4	DINT	44	received
				The total number of Logix tag reads
TagReads	4	DINT	46	executed by the module.
	4	DINT	40	The total number of Logix tag writes
Tagwrites	4	DINT	40	executed by the module.
				The number of failed class 3
				connection attempts.
ConnectionFailures	4		50	Note: Logix tag reading and writing
				requires the module to first establish
				a class 3 connection with the Logix
				Controller.
				The number of failed tag access
TagErrors	4		52	(read/write) requests.
				These may include privileged
				violations, non-existing tags, etc.
Slave Device Status x 64	1			
Slave Address	1	SINT	100 + (2 x	The node address of the mapped slave
		5111	Slave Idx)	on the CANopen network.

Status Bit 0 – Online Bit 1 – Error Received Bit 2 – PDO Error Bit 3 – Initializing Bit 4 – Stopped Bit 5 – Operational Bit 6 – Pre-Operational Bit 7 – Reserved	1	SINT	100 + (2 x Slave Idx)	 Online When the last response received from the slave is less than the <i>Slave Inactive Timeout</i> parameter in the CAN Bus configuration, the slave is considered online, and this bit is set. ErrorReceived Set when the last EMCY message received from the slave has an error. PdoError Set if one of the PDOs are not operating correctly. Initializing Set when the slave is in the initialize state. Stopped Set when the slave is in the stopped state. Operational Set when the slave is in the stopped state. PreOperational Set when the slave is in the pre-operational state.
PDO Status	2	DINT	101 + (2 x Slave Idx)	PDOxOk Each PDO (max 16) has a bit to indicate that it is operating as expected.
Master Output				
General Operation Bit 0 – Pre-Operational Bit 1 – Stopped Bit 2 – Inhibit Bit 3 to 31 – Reserved	4	DINT	300	 When the CANopen Router is the CANopen Master, the pre-operational bit will force the CANopen network to be PreOperational. When the CANopen Router is the CANopen Master, this stopped bit will force the CANopen network to be Stopped. When Inhibit has been set it will stop all CANopen communication. NOTE: When other NetworkPreOperational and NetworkStop bits are not set then the CANopen Router will set the network state to Operational.
UTC	8	DINT[2]	302	When the CANopen Router is a CANopen Master, the user can write the UTC Time (Unix Time format in microseconds) which will be converted into the CANopen time

				format for when sending TIME messages.
SlaveModeOutputTriggers Bit 0 – TxPDO0Trigger Bit 1 – TxPDO1Trigger Bit 15 – TxPDO15Trigger	4	DINT	306	When the CANopen Router is operating as a CANopen Slave, these bits are used to trigger sending of TPDOs when the Transmission Type is Evt – Logix . Each time the bit is toggle (either from 1 to 0 or from 0 to 1) the respective PDO will send the data to the CANopen Master.
Slave Device TPDO Trigger	1			
SlaveOutputTriggers (x 64 slaves) Bit 0 – TxPDO0Trigger Bit 1 – TxPDO1Trigger Bit 15 – TxPDO15Trigger	4 x 64	DINT	400 + (2 x PDO ldx)	TxPDOyTrigger When the CANopen Router is operating as a CANopen Master, these bits are used to trigger sending of PDOs to the CANopen Slave device when the Transmission Type is Evt – Logix . Each time the bit is toggle (either from 1 to 0 or from 0 to 1) the respective PDO will send the data to the CANopen Slave device.
Slave Mode PDOs x 16 PDOs				
PDO x	8 x 16	SINT	600	When the CANopen Router is operating as a CANopen Slave, each of the PDOs configured in the Virtual Device Map will be accessible from these Modbus Holding Registers (HR). For example, the first PDO in the Slave Tag Mapping will be at HR 600, PDO number two will be at HR 604, and so on.
Slave Device x 16 PDOs				
Slave Map Index 0 PDO x	8 x 16	SINT	1000	When the CANopen Router is operating as a CANopen Master, each of the PDOs configured in the Slave Device Map will be accessible from these Modbus Holding Registers (HR). For example, the first PDO in the Tag Mapping will be at HR 1000, PDO number two will be at HR 1004, and so on.
Slave Map Index 1 PDO x	8 x 16	SINT	1100	When the CANopen Router is operating as a CANopen Master, each of the PDOs configured in the Slave Device Map will be accessible from these Modbus Holding Registers (HR).

				For example, the first PDO in the Tag Mapping will be at HR 1100, PDO number two will be at HR 1104, and so on.
Slave Map Index 2 PDO x	8 x 16	SINT	1200	When the CANopen Router is operating as a CANopen Master, each of the PDOs configured in the Slave Device Map will be accessible from these Modbus Holding Registers (HR). For example, the first PDO in the Tag Mapping will be at HR 1200, PDO number two will be at HR 1204, and
				····
Slave Map Index 63 PDO x	8 x 16	SINT	1000	When the CANopen Router is operating as a CANopen Master, each of the PDOs configured in the Slave Device Map will be accessible from these Modbus Holding Registers (HR). For example, the first PDO in the Tag Mapping will be at HR 7300, PDO number two will be at HR 7304, and so on.

Table 4.9 – Modbus Mapping

5. DIAGNOSTICS

5.1. LEDS

The module provides three LEDs for diagnostics purposes as shown in the front view figure below. A description of each LED is given in the table below.



Figure 5.1 - CANopen Router front view

LED	Description
Run	The module Run LED will provide information regarding the operational state of the CANopen network.
	Solid Green – CANopen network is operational
	Flashing Green – CANopen network is pre-operational
	Blink Green – CANopen network is stopped
Eth	The Ethernet LED will light up when an Ethernet link has been detected (by plugging in a connected Ethernet cable). The LED will flash every time traffic was detected.
Err	The Err LED will provide information regarding the operational condition of the CANopen devices.
	CANopen Master
	Solid Red – No configuration has been loaded on the CANopen Router.

Flashing Red – The primary interface (EtherNet/IP or Modbus TCP) to the CANopen Router has been lost.
Blink Red – There is an issue with at least one CANopen Slave device.
Off – There are no issues.
CANopen Slave
Solid Red – No configuration has been loaded on the CANopen Router.
Flashing Red – The primary interface (EtherNet/IP or Modbus TCP) to the CANopen Router has been lost.
Blink Red – There is an issue with at least one PDO in the CANopen Router when operating as a CANopen Slave device.
Off – There are no issues.

Table 5.1 - Module LED operation

5.2. MODULE STATUS MONITORING IN SLATE

The CANopen Router can provide a range of statistics which can assist with module operation, maintenance, and fault finding. The statistics can be accessed in full by Slate or using the web server in the module.

To view the module's status in the Aparian-Slate environment, the module must be online. If the module is not already Online (following a recent configuration download), then right-click on the module and select the *Go Online* option.

S Aparian-Slate - CANopen	Rout	er5Dec2018
File Device Tools Win	ndow	Help
: [•] • • • • • • • • • • • • • • • • • •		E 2 余
Project Explorer		- ₽ ×
CANOpenRouter5Dec201	B And D	outor
Configuration	1	Configuration
Slave Devices	8	Connection Path
002] ICPDASC/	11	Go Online
- 🔲 [006] CANopenF	ł	Download
[054] AparianUn	Ť	Upload
	$\overline{\mathbf{A}}$	Verify Configuration
	Ъ	Сору
	¢	Export
	×	Delete

Figure 5.2. - Selecting to Go Online

The Online mode is indicated by the green circle behind the module in the Project Explorer tree.

File Device Tools Wind	low	Help
roject Explorer ■ & CANopenRouter5Dec2018		+ # X
CANOpenR (CANope Configuration Configuration Configuration Canopen Packet Ca Canopen Packet Ca Canopen Packet Ca Canopen Packet Ca Configuration C		Configuration Connection Path Ethernet Port Configuration Verify Configuration Identity Status Go Offline CANOpen Packet Capture Generate Status Report

Figure 5.3. - Selecting online Status

The Status monitoring window can be opened by either double-clicking on the *Status* item in the Project Explorer tree, or by right-clicking on the module and selecting *Status*.

The status window contains multiple tabs to display the status of the module. Most of these parameters in the status windows are self-explanatory or have been discussed in previous sections.

eneral CANOpen Statisti	cs CAN Statistics Modbus Sta	tistics Live List Discovery CIP Sta	tistics Ethernet Clients	TCP/ARP
Primary Interface	Modbus Online	MAC Address	00:60:35:1F:FA:E0	
Transaction Rate	2	Temperature	40.2 °C	
Up Time	0d - 00:40:35	Processor Scan	8.1 us	
Configured Node	1	Ethernet Cable Length	≈5m	
Mode	Master	DIP Switches SW1 - S	Safe Mode Off	
State	Operational	SW2 - I	Force DHCP Off	
Duplicate Detect	Ok	SW3 - 1	Reserved Off	
CAN Day	4	(Up	dated only on boot up)
CAN Time	28416			

Figure 5.4. - Status monitoring - General

The General tab displays the following general parameters and can also be used to set the module time to the PC time:

Parameter	Description
Primary Interface	The primary interface that was selected (EtherNet/IP or Modbus) and if the interface is online.
Transaction Rate	The transaction rate is the number of CANopen PDOs per second that the module is currently routing.
Up Time	Indicates the elapsed time since the module was powered-up.
Configured Node	The user required node address as specified in the module configuration.
Mode	The CANopen Router can either be configured as a CANopen Master or CANopen Slave.
State	The operational state of the CANopen network.
	Pre-operational
	Stopped
Duplicate Detect	Indicates if there is a duplicate node (same as the local node) on the CANopen network.
CAN Day	Current CAN Day (based on the Time on the CANopen network)
CAN Time	Current CAN Time (based on the Time on the CANopen network)
MAC Address	Displays the module's unique Ethernet MAC address.
Temperature	The internal temperature of the module.
Processor Scan	The amount of time (microseconds) taken by the module's processor in the last scan.
Ethernet Cable Length	Approximate length of the Ethernet cable (accurate to 5m).
DIP Switch Position	The status of the DIP switches when the module booted.
	Note that this status will not change if the DIP switches are altered when the module is running.

Table 5.2 - Parameters displayed in the Status Monitoring – General Tab

Counter	Value			
PDO 1x Count	468			
PDO Rx Count	1247			
SDO Tx Count	218			
SDO Rx Count	72			
TIME Packet Count	385			
SYNC Packet Count	0			
EMCY Packet Count	0			
HEARTBEAT / State Packet Count	655			

Figure 5.5. - Status monitoring – CANopen Statistics

The CANopen Statistics tab displays the following general parameters:

Parameter	Description		
PDO Tx Count	The number of PDO packets transmitted.		
PDO Rx Count	The number of PDO packets received.		
SDO Tx Count	The number of SDO packets transmitted.		
SDO Rx Count	The number of SDO packets received.		
TIME Packet Count	The number of TIME packets received or sent.		
SYNC Packet Count	The number of SYNC packets received or sent.		
EMCY Packet Count	The number of EMCY packets received or sent.		
Heartbeat Packet Count	The number of Heartbeat packets received.		

Table 5.3 - Parameters displayed in the Status Monitoring – CANopen Statistics Tab
ieral CANOpen Statistics C	AN Statistics	Modbus Statisti	s Live List	Discovery	CIP Statistics	Ethernet Clients	TCP / ARP	
CAN Statistics	Clear	Counters						
Counter		Value						
RxCanPacketCount		11043						
TxCanPacketCount		20954						
CAN CRC Errors		0						
CAN Bit Errors		0						
Can Stuff Errors		0						

Figure 5.6. - Status monitoring – CAN Statistics

The CAN Statistics tab displays the following general parameters:

Parameter	Description
RxCANPacketCount	Received CAN message count.
TxCANPacketCount	Transmitted CAN message count.
CAN CRC Errors	CAN CRC failed message count.
CAN Bit Errors	CAN Bit error count.
CAN Stuff Errors	CAN Stuff error count.

Table 5.4 - Parameters displayed in the Status Monitoring – CAN Statistics Tab

eneral CANOpen Statistics C	AN Statistics	LOGIX STRUSTICS	LIVE LIST	Discovery	CIP Statistics	Ethernet Clients	TCP/ARP	
Logix Statistics	Clear	Counters						
Counter		Value						
Current Connections		1						
Connection Failures		0						
Tag Not Exist Errors		0						
Privilege Violations		0						
Tag Reads		0						
Tag Writes		9						
ENIP Retries		0						
ENIP Failures		0						
General Access Errors	1	0						

Figure 5.7. - Status monitoring – Logix Statistics

The Logix Statistics tab displays the following general parameters:

Parameter	Description
Current Connections	The number of current open class 3 connections.
Connection Failures	The number of failed attempts at establishing a class 3 connection with a Logix controller.
Tag Not Exist Errors	The number of tag read and tag write transactions that failed due to the destination tag not existing.
Privilege Violations	The number of tag read and tag write transactions that failed due to a privilege violation error.
	This may be caused by the External Access property of the Logix tag being set to either None or Read Only.
Tag Reads	The number of tag read transactions executed by the CANopen Router module.
Tag Writes	The number of tag write transactions executed by the CANopen Router module.
ENIP Retries	This count increases when no response was received from the Logix Controller by the time the ENIP timeout is reached.
ENIP Failures	This count increases when the ENIP Retry Limit is reached and no response has been received from the Logix Controller.
Tag Access General Error	This count increases when a tag cannot be accessed for any other reason not reported above.

Table 5.5 - Parameters displayed in the Status Monitoring – Logix Statistics Tab

		4	·		
Modbus Statistics			Clea	r Counters	
Counter	Value	Counte	r	Value	
Tx Packet Count	17430 II	egal Function		0	
Rx Packet Count	17430 II	egal Data Address		0	
Checksum Errors	0 11	egal Data Value		0	
Parity Errors	0 5	lave Device Failure		0	
Timeout Errors	0 A	cknowledge - Repo	nse Delay	0	
Data Too Large	0 5	lave Device Busy		0	
Map Item Not Found	0	legative Acknowled	je	0	
Node Mismatch	0 N	lemory Parity Error		0	
Data Alignment Errors	0				

Figure 5.5. - Status monitoring – Modbus Statistics

Statistic	Description
Tx Packet Count	The number of Modbus packets sent by the module.
Rx Packet Count	The number of Modbus packets received by the module.
Checksum errors	The number of corrupted Modbus packets received by the module.
Parity errors	The number of bytes with parity errors received by the module.
Timeout Errors	The number of message response timeouts the module has encountered.
Data Too Large	The number of Modbus requests or responses where the data was too large to process.
Map Item Not Found	The number of Modbus requests did not match any mapped items.
Node Mismatch	The received Modbus request did not match the module's Modbus node address.
Data Alignment Errors	The Modbus request and associated mapped item is not byte aligned with the destination Logix tag.
Illegal Function	The number of times the Modbus device responded with an Illegal Function exception.
Illegal Data Address	The number of times the Modbus device responded with an Illegal Data Address exception.
Illegal Data Value	The number of times the Modbus device responded with an Illegal Data Value exception.
Slave Device Failure	The number of times the Modbus device responded with a Device Failure exception.

Acknowledge –Response Delay	The number of times the Modbus device responded with an Acknowledge exception.
Slave Device Busy	The number of times the Modbus device responded with a Slave Busy exception.
Negative Acknowledge	The number of times the Modbus device responded with a Negative Acknowledge exception.
Memory Parity Error	The number of times the Modbus device responded with a Memory Parity exception.

Table 5.6 - Parameters displayed in the Status Monitoring – Modbus Statistics Tab

10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119	0	1-M	2	3	4	5	6	7	8	9		
20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119	10	11	12	13	14	15	16	17	18	19		
30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 14	20	21	22	23	24	25	26	27	28	29		
40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127	30	31	32	33	34	35	36	37	38	39		
50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 140 140	40	41	42	43	44	45	46	47	48	49		
60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 140	50	51	52	53	54	55	56	57	58	59		
70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 127	60	61	62	63	64	65	66	67	68	69		
80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127	70	71	72	73	74	75	76	77	78	79		
90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127	80	81	82	83	84	85	86	87	88	89		
100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127	90	91	92	93	94	95	96	97	98	99		
110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127	100	101	102	1 <mark>0</mark> 3	104	105	106	107	108	109		
120 121 122 123 124 125 126 127	110	111	112	113	114	115	116	117	118	119		
	120	121	122	123	124	125	126	127				

Figure 5.6. - Status monitoring – Live List

The Live List provides the online status of each CANopen Slave device on the CANopen network. When the address in the live list is green when the last response received from the slave is less than the *Slave Inactive Timeout* parameter in the CAN Bus configuration (i.e. the device is considered to be online). When the address in the live list is turquoise with "-M" next to it then it is the node address of the CANopen Master.

5							
ter	Value						
ount	0						
pen Count	0						
ose Count	0						
n Count	0						
Class 3 Timeout Count							
Class 3 Forward Open Count							
ose Count	0						
1 Count	0						
	punt pen Count ose Count o Count punt pen Count ose Count o Count	punt0pen Count0ose Count0o Count0o punt0pen Count0ose Count0ose Count0o count0	punt0pen Count0ose Count0o Count0pen Count0pen Count0ose Count0ose Count0ose Count0	punt0pen Count0ose Count0ocount0pen Count0pen Count0ose Count0ose Count0ocont0	punt0pen Count0ose Count0a Count0pen Count0ose Count0a count0a count0a count0	punt0pen Count0ose Count0a Count0pen Count0ose Count0ose Count0a Count0a Count0	Dunt0pen Count0ose Count0o Count0pen Count0ose Count0ose Count0o Count0

Figure 5.7. - Status monitoring – CIP statistics

The CIP statistics tab displays the following general parameters:

Statistic	Description
Class 1 Timeout Count	Number of times a Class 1 connection has timed out
Class 1 Forward Open Count	Number of Class 1 Connection establish attempts
Class 1 Forward Close Count	Number of Class 1 Connection close attempts
Class 1 Connection Count	Number of Class 1 Connections currently active
Class 3 Timeout Count	Number of times a Class 3 connection has timed out
Class 3 Forward Open Count	Number of Class 3 Connection establish attempts
Class 3 Forward Close Count	Number of Class 3 Connection close attempts
Class 3 Connection Count	Number of Class 3 Connections currently active

Table 5.7. - CIP Statistics

5.3. SLAVE DEVICE STATUS MONITORING IN SLATE

To view the CANopen Slave device's status in the Aparian-Slate environment, the module must be online. If the module is not already Online (following a recent configuration download), then right-click on the module and select the *Go Online* option.



Figure 5.8. - Selecting to Go Online

The Online mode is indicated by the green circle behind the module in the Project Explorer tree.



Figure 5.9. - Selecting Slave Device Online Status

The Status monitoring window can be opened by right-clicking on the CANopen Slave device and selecting *Status*.

The status window contains multiple tabs to display the status of the CANopen Slave device. Most of these parameters in the status windows are self-explanatory or have been discussed in previous sections.

lode Status		Information and Control
Status	Online	Instance Name ICPDASCANopenSla
State	Operational	EDS Filename CAN_2053C.eds
Inactive Time	200	Operational
Last Error Code	0x0	Pre-Operational
Last Error Type		Stop
		Reset
		Sand Steep Decementary

Figure 5.10. – CANopen Slave Device Online Status

The CANopen Slave Device Status tab displays the following general parameters:

Statistic	Description
Status	The current online/offline status of the slave device.
State	The current state of the slave device.
	Operational
	Pre-operational
	Stopped
Inactive Time	The amount of time (in milliseconds) that have elapsed since the last response from the slave device.
Last Error Code	The last error code received from the slave device.
Last Error Type	The last error type received from the slave device.
Control	The status form allows the user to send certain control messages to the device.
	Operational
	Send command to the specific field device to go into operational mode.
	Pre-Operational
	Send command to the specific field device to go into pre-operational mode.
	Stop
	Send command to the specific field device to go into stop mode.
	Reset
	Send command to the specific field device to reset.

Send Store Parameters
 Send the write command to parameter index 1010 sub-index 2, to save all parameters in the CANopen Slave device to non-volatile memory.

Table 5.8. - CANopen Slave Device Online Status

5.4. CANOPEN PACKET CAPTURE

The module provides the capability to capture the CANopen traffic for analysis. The will allow the user and a remote support team to resolve any possible issues on site. To invoke the capture of the module, double-click on the CANopen Packet Capture item in the Project Explorer tree.



Figure 5.11 - Selecting CANopen Packet Capture

The CANopen Packet Capture window will open and automatically start capturing all CANopen packets.

CANOpe	nR2 - CANOpe	en Packet Ca	pture						
	0	1.22	12-21	111 State 1420			2 8 2 .	2.0	
Index	▲ Time	Status	Dirn	NodelD	Function	COB-ID	Description	Data	
	Press STOP	to view resul	lts.						
Capturing	Packets : 3	39							

Figure 5.12 – CANopen packet capture

To display the captured CANopen packets, the capture process must first be stopped, by pressing the Stop button.

Index	Time	Status	Dirn	NodelD	Function	COB-ID	Description	Data	
40048	0d - 00:52:38.980	Ok	Tx	6	RPDO 1	0x0206	Receive PDO 1	06 02 00 C0 79 44	
40049	0d - 00:52:39.020	Ok	Tx	54	TPDO 3	0x03B6	Transmit PDO 3	B6 03	
40050	0d - 00:52:39.230	Ok	Tx	2	RSDO	0x0602	Receive SDO	02 06 40 17 10 00 00 00 00 00	
40051	0d - 00:52:39.230	Ok	Rx	54	TPDO 1	0x01B6	Transmit PDO 1	B6 01 95 75 FE 41 FE 25 36 41	
40052	0d - 00:52:39.230	Ok	Rx	2	TSDO	0x0582	Transmit SDO	82 05 4B 17 10 00 E8 03 00 00	
40053	0d - 00:52:39.230	Ok	Rx	54	NMT Err Ctrl	0x0736	Operational	36 07 05	
40054	0d - 00:52:39.320	Ok	Rx	2	TPDO 1	0x0182	Transmit PDO 1	82 01 01 04	
40055	0d - 00:52:39.380	Ok	Tx	6	RPDO 1	0x0206	Receive PDO 1	06 02 00 C0 79 44	
40056	0d - 00:52:39.650	Ok	Rx	2	NMT Err Ctrl	0x0702	Operational	02 07 05	
40057	0d - 00:52:39.700	Ok	Тх	4	RSDO	0x0604	Receive SDO	04 06 40 01 10 00 00 00 00 00	
40058	0d - 00:52:39.750	Ok	Tx	2	RSDO	0x0602	Receive SDO	02 06 40 00 10 00 00 00 00 00	
40059	0d - 00:52:39.750	Ok	Rx	2	TSDO	0x0582	Transmit SDO	82 05 43 00 10 00 91 01 01 00	
40060	0d - 00:52:39.780	Ok	Tx	6	RPDO 1	0x0206	Receive PDO 1	06 02 00 C0 79 44	

Figure 5.13 – CANopen Packet Capture complete

The captured CANopen packets are tabulated as follows:

Statistic	Description
Index	The packet index, incremented for each packet sent or received.
Time	The elapsed time since the module powered up.
Status	The status of the packet. Received packets are checked for valid CANopen constructs and valid checksums.
Dirn	The direction of the packet, either transmitted (Tx) or received (Rx).
NodelD	The Source Node address for the packet
Function	The CANopen function.
COB-ID	The COB-ID for the specific packet.

Description	Description of the packet that was received.
Data	The raw packet data.

Table 5.9 – CANopen Packet Capture fields

The packet capture can be saved to a file for further analysis, by selecting the *Save* button on the toolbar. Previously saved CANopen Packet Capture files can be viewed by selecting the *CANopen Packet Capture Viewer* option in the tools menu.

le Device	Too	ls Window Help		
	Q. 参回 乡	Target Browser DHCP Server Event Viewer DeviceFlash		
	w.	Packet Capture Viewers	يت (DF1 Packet Capture Viewer
	G C B A	GSD File Management	ű	DH485 Packet Capture Viewer
		CAN EDS File Management Build DeviceNet EDS Catalog	ŭ ŭ	Modbus Packet Capture Viewer DNP3 Packet Capture Viewer
		Application Settings	ů.	HDLC Packet Capture Viewer
			ŭ ŭ	AT Packet Capture Viewer
			J.	CANopen Packet Capture Viewer
			J.	Profibus Packet Capture Viewer

Figure 5.14 - Selecting the CANopen Packet Capture Viewer

5.5. MODULE EVENT LOG

The CANopen Router module logs various diagnostic records to an internal event log. These logs are stored in non-volatile memory and can be displayed using Slate or via the web interface. To view them in Slate, select the Event Viewer option in the Project Explorer tree.



Figure 5.15. - Selecting the module Event Log

The Event Log window will open and automatically read all the events from the module. The log entries are sorted so as to have the latest record at the top. Custom sorting is achieved by double-clicking on the column headings.

2X			
Upload	ed 488 records.	Filter (All)	~
Index	 Up Time 	Event	
487	0d - 02:25:43	Modbus Comms Ok	
486	0d - 02:25:43	Config valid	
485	0d - 00:58:28	Modbus Comms Ok	
484	0d - 00:58:28	Config valid	
483	0d - 00:47:37	Modbus Comms Ok	
482	0d - 00:47:37	Config valid	
481	0d - 00:46:12	Modbus Comms Ok	
480	0d - 00:46:12	Config valid	
<mark>47</mark> 9	0d - 00:45:40	Config valid	
478	0d - 00:45:07	Modbus Comms Ok	
<mark>477</mark>	0d - 00:45:07	Config valid	
476	0d - 00:43:51	Modbus Comms Ok	
475	0d - 00:43:51	Config valid	
474	0d - 00:34:06	EMCY from Node 6 - Code 0x0000 Type 0x00	
473	0d - 00:33:41	Modbus Comms Ok	
472	0d - 00:33:41	Config valid	
471	0d - 00:04:20	Modbus Comms Ok	
470	0d - 00:04:20	Config valid	



The log can also be stored to a file for future analysis, by selecting the Save button in the tool menu. To view previously saved files, use the Event Log Viewer option under the tools menu.

5.6. WEB SERVER

The CANopen Router provides a web server allowing a user without Slate or Logix 5000 to view various diagnostics of the module. This includes Ethernet parameters, system event log, advanced diagnostics, and application diagnostics (e.g. CANopen statistics).



NOTE: The web server is view **only** and thus no parameters or configuration can be altered from the web interface.

🖌 💕 Aparian	× +		- 🗆 X	
← → C ③ Not secure	192.168.1.247		☆ G :	
Module: CANopen	Router Serial: 351FFAE0 Firmw	are Rev: 1.2	aparian	
Overview	Device Name	CANopen Router		
Ethernet	Serial number	351FFAE0		
EventLogs	Firmware Revision	1.2		
Diagnostics	Module Status	Configured		
Diagnosucs	Vendor Id	1370		
Application	Product Type	12		
www.aparian.com	Product Code	123		
	Uptime	52m 58s		
	Switches	0:0:0:0		
	Temperature	39.7660°C		
	Copyright 2015 Aparian Inc	. All rights reserved		

Figure 5.17. - Web interface

6. TECHNICAL SPECIFICATIONS

6.1. DIMENSIONS

Below are the enclosure dimensions as well as the required DIN rail dimensions. All dimensions are in millimetres.



Figure 6.1 – CANopen Router enclosure dimensions



Figure 6.2 - Required DIN dimensions

6.2. ELECTRICAL

Specification	Rating
Power requirements	Input: 10 – 28V DC, (70 mA @ 24 VDC)
Power consumption	1.7 W
Connector	5-way terminal, 5.08mm pitch.
Conductors	24 – 18 AWG
Enclosure rating	IP20, NEMA/UL Open Type
Temperature	-20 – 70 °C
Earth connection	Yes, terminal based
Emissions	IEC61000-6-4
ESD Immunity	EN 61000-4-2
Radiated RF Immunity	IEC 61000-4-3
EFT/B Immunity	EFT: IEC 61000-4-4
Surge Immunity	Surge: IEC 61000-4-5
Conducted RF Immunity	IEC 61000-4-6

Table 6.1 - Electrical specification

6.3. ETHERNET

Specification	Rating
Connector	RJ45
Conductors	CAT5 STP/UTP
ARP connections	Max 20
TCP connections	Max 20
CIP connections	Max 10
Communication rate	10/100Mbps
Duplex mode	Full/Half
Auto-MDIX support	Yes

Table 6.2 - Ethernet specification

6.4. CANOPEN	NETWORK
--------------	---------

Specification	Rating
Connector	5-way terminal, 5.08mm pitch.
Modes	CANopen Master
	CANopen Slave
CANopen Slave Count	64
PDO Count per Device	16
Supported Baud Rates	50k
	125k
	250k
	500k
	800k
	1M
CiA 443 Support	Yes
NMT messages	Operational Control (e.g. Stopped, Pre-operational, Operational)
	SYNC
	ТІМЕ
	EMCY

Table 6.3 – CANopen specification



NOTE: Although the CANopen Router supports the CiA443 objects, the CANopen interface is not fault-tolerant.

6.5. CERTIFICATIONS

Certification	Mark
CE Mark	CE
RoHS2 Compliant	RoHS ₂
UL Mark	
File: E494895	

CLASS 1, DIV 2, GROUPS A, B, C, D

Table 6.4 – Certifications

7. INDEX

Α

Activity, 68 assembly instance, 53

С

CANopen Router, 5, 21, 53, 57, 74 CANopen Router, 5 CANopen Router, 74 CANopen Router general configuration, 22, 23, 25 CANopen ROUTER parameters, 22 Contact Us, 10

D

DHCP, 12, 15, 16, 17 dimensions, 85 DIN rail, 13, 85 DIP, 12

Ε

Ethernet Bridge, 52 Ethernet connector, 14 EtherNet/IP, 5 *Export*, 28

F

Field device general configuration, 29 firmware upgrade, 22

I

Import, 28 input assembly, 57, 71, 72, 73 input voltage, 14 Instance Name, 30 IP Address, 16, 23 LED, 68, 69 Logix tag, 59, 64

Μ

L

MODBUS, 75

0

output assembly, 57, 59, 60

Ρ

partial import, 55

R

requested packet interval (RPI), 53 Rockwell Automation, 19 RSLinx, 19 RSLogix 5000, 52, 53, 54, 55, 56, 59, 60, 84

S

Safe Mode, 12 Slate, 21, 22, 57, 63, 69, 84 statistics, 69, 82 Support email, 10

Т

Target Browser, 17, 18, 50

U

User Defined Types (UDTs), 54

W

web server, 69, 84