

NR500 Series Industrial Cellular VPN Router

Application Note 067

Modbus Sparkplug B

Version:V1.0.0Date:Apr 2023Status:Confidential





Directory

1. Introduction	3
1.1 Overview	3
1.2 Compatibility	3
1.3 Version	3
1.4 Corrections	3
2. Topology	4
3. Configuration	5
3.1 Configuration on Modbus Slave	5
3.2 Configuration on Modbus Poll	6
3.3. Configuration on Modbus to Sparkplug B	8
4. Testing	10



1. Introduction

1.1 Overview

This document contains information regarding the configuration and use of Modbus Sparkplug B.

This guide has been written for use by technically competent personnel with a good understanding of the communications technologies used in the product, and of the requirements for their specific application.

1.2 Compatibility

This application note applies to: **Models Shown:** NR500 series. **Firmware Version:** V3.0.0 (1f18909-1) or newer **Other Compatible Models:** None

1.3 Version

Updates between document versions are cumulative. Therefore, the latest document will include all the content of previous versions.

Release Date	Doc. Version	Firmware Version	Change Description		
2023/04/12	V1.0.0	V3.0.0 (1f18909-1)	First released		

1.4 Corrections

Appreciate for corrections or rectifications to this application note, and if any request for new application notes please email to: **support@navigateworx.com**



2. Topology



- 1. NR500 Router runs as Modbus Master and send Modbus Sparkplug B data using MQTT, it acts as Modbus Master and MQTT EoN Node.
- 2. A serial device support Modbus protocol and acts as Modbus Slave, NR500 router poll the Modbus data from end device (Modbus Slave).
- 3. NR500 router run as MQTT EoN Node, it can convert Modbus data or the status of Local-IO to sparkplug B data to MQTT broker via MQTT protocol.
- 4. The SCADA/IIoT Host can subscribe the specified topic from the MQTT broker to read the register value/Local-IO of NR500 router or send NCMD/DCMD to write the register value/Local-IO of NR500 router.

Note: For this Application Note, we will set Modbus Master's Connection Type as "TCP" as an example, which means that NR500(Modbus Master) will connect to the Modbus Slave and read the value via Ethernet port. Of course, it also works with Serial Port (RS232/RS485).



3. Configuration

3.1 Configuration on Modbus Slave

 Here we use "Modbus Slave" software to simulate the end device (Modbus Slave device), and the TCP Port: 502, Slave ID: 1, Function Code: 01-Coil Status, like below settings:

	- 🗆 X
Window Help	
Connection Setup X	
Connection OK Modbus TCP/IP	
Serial Settings COM5 Mode RTU ASCI B Data bits Flow Control None Parity DSR CTS RTS Toggle DSR CTS RTS Toggle TCP/IP Server IP Address T22.16.1.163 VAny Address I go IPv4 I go Unit ID IPv6	
Window Help	- 🗆 X
Slave Declinition Slave ID: Slave ID: Function: DK Function: Cancel Address: Cancel Address: Pows New New New Find Cancel Insert CRC/LRC error Not when using TCP/IP Not when using TCP/IP	
	Window Help Connection Setup Image: Setial Setting: Connection OK Serial Setting: COMS Inscription Data bits Flow Control None Parity DSR CTS Port Stop Ba Image: Parity DSR CTCP/P Server IP Address IP Address Ipone Unit ID IP N6 Window Help Slave Definition Verw Image: One One Slave Definition Verw Image: One O Image: One Image: One Image: One Image: One



3.2 Configuration on Modbus Poll

 Go to Application>Modbus Master>Modbus Poll, add a "Connection List" and specify the "Connection Type" as "TCP", specify the "TCP Settings" to connect to Modbus Slave, like below:

Connecti	on Settin	gs				
Connect	ion List					
		Index	1]	
		Enable	<			
		Description	Sparkpl	ugB-slave]	
		Scan Rate	10000] ⑦	
		Reconnect Interval	60] ⑦	
		Response Timeout	1000]	
		Delay Between Polls	0] ⑦	
		Connection Type	ТСР	~]	
		Enable Show Status				
		Enable Verbose Log				
TCP Sett	ings				c.c.	
		Server Address	192.168	3.111.153	🕇 🔶 The IP	address of
		Server Port	502		Modb	us slave
		Connection Timeout	10] ⑦	
Channel	List					
Index	Enable	Description	Slave ID	Function Code	Register Address	÷
1	true		1	01-Coil-Status	0	$\mathbb{Z}\otimes$ -
					Save	Close

- 2. Click Save.
- Enable "Channel List", and specify the Slave ID as "1", Function Code as "01-Coil-status", Register Address to "0", then it will poll the value from register address 0 of Modbus Slave:



Channel Settings	
Channel List	
Index	1
Enable	
Description	
Slave ID	1
Function Code	01-Coil-Status
Register Address	0
Data type	Bool 🗸
Multiple Register	
	Save Close

4. Click Save>Save>Apply. (Note: This is a secondary list, it needs to double click save)

5. Go to **Application>Modbus Master>Status**, then we can check the router had read the value from Modbus Slave successfully.





3.3. Configuration on Modbus to Sparkplug B

1. Go to **Application>Modbus Sparkplug>Modbus To Sparkplug B**, click enable button and specify the SCADA/IIoT Host ID, Group ID and Node ID.

Status Modbus To Sparkplug B X.509	Certificate				
Sparkplug B Settings				a.	
	Enable				
SCADA/II	oT Host ID	scada_host_id			
	Group ID	Sparkplug B Devices			
	Node ID	NR500	0		
No	de Update	30	?		
	Trigger	Cyclic 🗸	0		
Dev	ice Update	30	?		
Upda	ate Method	Force Update 🗸	0		
Historical Dat	ta Location	FLASH ~	0		
Historical Upda	ite Interval	10	0		
Enable Ve	erbose Log				
MQTT Broker Settings					
Index Enable MQTT Broker Address Kee	palive Qo	DS			\oplus
Devices Settings					
Index Device Type Device ID Connec	tion Index	Slave ID			÷
				Sav	е Арріу

Note:

1. SCADA/IIOT Host ID: NR500 subscribes the specified topic to get the status of SCADA/IIOT Host.

The format of the specified topic: STATE/[SCADA/IIoT Host ID],

Qos: the Qos of the MQTT broker,

Content: ONLINE/OFFLINE.

- 2. Node updates operates on NDATA message type.
- 3. Device updates operates on DDATA message type.
- 4. Trigger only operates on Device update.

There are two update methods, [SCADA/IIoT Host Online] and [Force update]. Under SCADA/IIOT Host online Method, Modbus to Sparkplug B only will publish the data to MQTT broker after received the SCADA/IIoT Host ONLINE message. Under Force update method, Modbus to Sparkplug B will force publish the data to MQTT broker even SCADA/IIoT Host OFFLINE.

2. Click Save > Apply.



3. Enable "MQTT Broker settings", click enable button and specify the MQTT Broker Address, select MQTT Version and config Username and Password.

Status	Broker Settings						
Sparkplug	MQTT Broker Setti	ings					
		Index	1				
		Enable					
		MQTT Broker Address	192.168.111.153	0			
	2	MQTT Version	MQTTv3.1.1 ~				
		Keepalive	20	0			
		Enable SSL					
		Username	test				
		Password	•••••				
		Qos	1 ~				
				Save	Close		1
					_		
MQTT Bro	oker Settings						<u> </u>
Index	Enable MQTT Brok	ker Address Keepalive	e Qos				÷
Devices S	ettings						
Index	Device Type De	vice ID Connection	Index Slave ID				(i)
						Save	Apply

- 4. Click Save>Save>Apply.
- 5. Enable "Devices settings", click enable button and select the Device Type, enter Device ID and select Connection Index and Slave ID.

Status	Device Set	tings						
Sparkplug	Devices Se	ettings						
			Index	1				
			Device Type	Modbus-Slave	~			
		2	Device ID	slave1				
			Connection Index	1	~ ⑦			
			Slave ID	1	0			
	Metrics Se	ttings						
	Index	Name	Modbus Channel	Index Data Type	Mode	\oplus		
					Save	Close		
			mistorical opuate in					
			Enable Verbos	se Log 🗹				
MQTT Bro	ker Settings	3						
Index	Enable	MQTT Broker A	ddress Keepalive	e Qos				(\pm)
1	true	192.168.111.	153 20	1				\boxtimes
Devices S	ettings							
Index	Device Type	Device 1	D Connection	Index Slave ID				()
							Save	Apply

Note: When Device Type is Modbus-Slave, NR500 will convert modbus data to sparkplug B data and publish it to remote MQTT broker. If Device Type is Local-IO, NR500 will convert the status of local IO to the sparkplug B data and publish it to remote MQTT broker.



6. Enable "Metrics Settings", enter Name and Modbus Channel Index, select Data Type and Mode.

Metrics Settings					
Metrics Settings					
Index	1				
Name	Boolean		?		
Modbus Channel Index	1.0		0		
Data Type	BOOLEAN	~			
Mode	Read-Only	~	0		
			Save	С	lose

Note: The Format of Modbus Channel Index: X.Y; [X]:Modbus poll channel index, [Y]:Modbus register address.

- 7. Click Save>Save>Apply.
- 8. Go to **Application>Modbus Sparkplug B>Status**, NR500(Modbus to Sparkplug B) had connected to the remote server successfully via MQTT protocol.

<u>Status</u> Modl	bus To Sparkplug B X.509 Certificate	2
Modbus To Sparkpl	lug B Status	
	Enable	Enabled
	MQTT Connection Status	192.168.111.153 Connected
	SCADA/IIoT Host Status	OFFLINE
	Historical Data Record	EoN(0), Devices(0)

4. Testing

1. Run the MQTT Client (MQTT.fx), to subscribe the topic, then we can get the sparkplug B data successfully.

🛞 MQTT.fx - 5.2.0 -	Standard Edition							- 🗆 ×
MQTT.fx Extras	Help							
local broker		× 0 (Connect D	isconnect		MQTT 3 🛔	-	MQTT.b
Publish Subs	cribe Sparkplug Explore	r Scripts	Broker Status	Log				
spBv1.0/Sparkplug	B Devices/#	•	Subscribe			Qo50	QoS1 QoS	2 Autoscroll
spBv1.0/#		723	Topic Filter					List Tab
	Dump Messages (JSON) Mute	Unsubscribe	1429	14:37:24.52644762	spBv1.0/Sparkplug B Devices/DDATA/M	NR500/slave1		
spBv1.0/Sparkplug B D	evices/#	110	1430	14:37:33.52653874	spBv1.0/Sparkplug B Devices/NDATA/N	NR500		
	Dump Messages (JSON) Mute	Unsubscribe	1431	14:37:33.52653874	spBv1.0/Sparkplug B Devices/NDATA/N	NR500		
			1432	14:37:54.52674774	spBv1.0/Sparkplug B Devices/DDATA/N	NR500/slave1		
			1433	14:37:54.52674774	spBv1.0/Sparkplug B Devices/DDATA/N	NR500/slave1		
			1434	14:38:05.52685444	spBv1.0/Sparkplug B Devices/NDATA/N	NR500		
			1435	14:38:05.52685444	spBv1.0/Sparkplug B Devices/NDATA/N	NR500		
			1436	14:38:24.52704795	spBv1.0/Sparkplug B Devices/DDATA/№	NR500/slave1		
			1437	14:38:24.52704796	spBv1.0/Sparkplug B Devices/DDATA/N	NR500/slave1		
Topics Collector (0)	Scan	Stop 🔍	spBv1.0/Spar	kplug B Devices/DDAT	A/NR500/slave1			
			{ "timestar "metrics" "name"	mp" : 1681367904824, " : [{ : "Boolean",		1437 QoS 0 13-04-2023 14:38:24	52704796	
			"timest "dataTy "value" }], "seq" : 6 } Content Type	tamp" : 168136790482 ype" : "Boolean", " : true 52 application	vsparkplug	Name	Value No content	in table
			Payload decoded	with Sparkpli	g Pretty Format Decoder			