

Detecting when a MultiHop Radio has Stopped Communicating with the Wireless Network

OVERVIEW

Detecting when a MultiHop radio has dropped out of a wireless network requires the Modbus master/host system to track communications errors. When a Modbus master detects communication errors, it can determine how many consecutive errors are acceptable before considering a radio in a failed condition. This technical note shows how the DX83/GatewayPro can be configured to detect communication errors and set default conditions. A Modbus master system should be similarly configured to detect bad radios or links.

How MULTI HOP NETWORKS ARE DIFFERENT FROM DX80 NETWORKS

The MultiHop wireless family and the DX80 wireless family use different base architectures that operate similarly but are functionally different to a Modbus master system.

The DX80 system uses a star-based architecture with the Gateway connected to the host system and wireless Nodes communicating with the Gateway. To the Modbus master system, the Gateway appears as a single Modbus slave device containing register data for all the Nodes in the wireless system. The host interrogates the Gateway for all register data, sends control messages, or reads status information. The Gateway maintains the network status information because the host only communicates with the Gateway.

The MultiHop wireless family is a hierarchical tree architecture. All radio repeaters and radio slaves connect to one MultiHop radio master, creating a multiple level network structure. Each radio is addressed individually; the MultiHop radio master is only an access point for the Modbus master system. The Modbus master must send messages to a particular device to access the data registers at that device and the host must handle status information (link quality).

This technical note shows how the DX83 detects radio link status. (A Modbus master PC or PLC system should have a similar setup.)

The DX83 or GatewayPro radio use Read/Write maps to communicate with remote devices like the MultiHop radios. Each Read map entry defines the remote Modbus slave, the register to access, and how often to read the register.

The user may also define how many read failures are acceptable in the system. For example, after five read failures, a default value will be placed in the local register. This default value is typically a “safe condition” value but is also something a host recognizes as an error condition. The radio’s status is inferred based on the lack of communication to the radio.

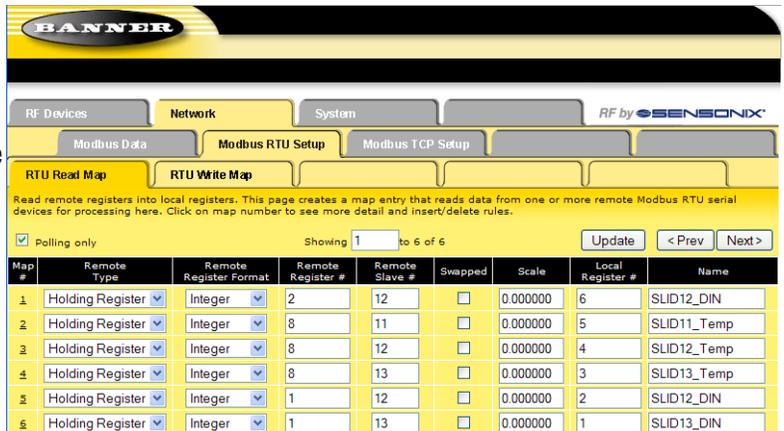
DX83/GATEWAYPRO EXAMPLE

The RTU Read Map screen shot shows the DX83 reading I/O registers on several MultiHop Radios. Click on the Map # to show the Expanded RTU Read Map screen to select specific conditions.

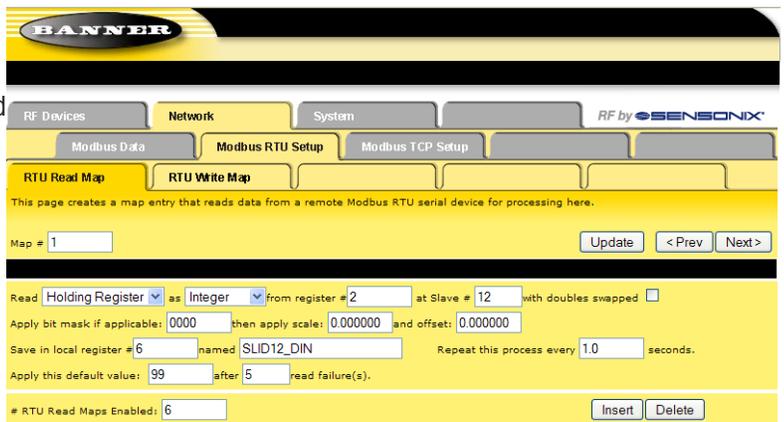
In the example shown, register 2 of MultiHop Radio ID 12 is read into the DX83’s register 6. After 5 read failures, the default value is written to the DX83’s register 6 to indicate MultiHop Radio ID 12 has stopped communicating with the wireless network.

Select the default value to indicate an error based on the I/O type. For example, writing a 99 to a register that normally receives a discrete value easily indicates an error.

Note: The MultiHop Radio ID is recognized by the host system as the Modbus Slave ID for that radio. For example, MultiHop Radio 12 is Modbus Slave 12.



Map #	Remote Type	Remote Register Format	Remote Register #	Remote Slave #	Swapped	Scale	Local Register #	Name
1	Holding Register	Integer	2	12	<input type="checkbox"/>	0.000000	6	SLID12_DIN
2	Holding Register	Integer	8	11	<input type="checkbox"/>	0.000000	5	SLID11_Temp
3	Holding Register	Integer	8	12	<input type="checkbox"/>	0.000000	4	SLID12_Temp
4	Holding Register	Integer	8	13	<input type="checkbox"/>	0.000000	3	SLID13_Temp
5	Holding Register	Integer	1	12	<input type="checkbox"/>	0.000000	2	SLID12_DIN
6	Holding Register	Integer	1	13	<input type="checkbox"/>	0.000000	1	SLID13_DIN



Map # 1

Read Holding Register as Integer from register # 2 at Slave # 12 with doubles swapped

Apply bit mask if applicable: 0000 then apply scale: 0.000000 and offset: 0.000000

Save in local register # 6 named SLID12_DIN Repeat this process every 1.0 seconds.

Apply this default value: 99 after 5 read failure(s).

RTU Read Maps Enabled: 6