Niles GXR2 control information

As of – May 1, 2021

Please note that these notes are a work in progress and may contain incomplete or incorrect information. They are intended for reference only.

The information contained in this document was obtained by monitoring network traffic between a Niles GXR2 and one of its controllers using Wireshark. Buttons on the controller were pushed, and the results recorded and compared to see which bytes changed. These results could be duplicated by bridging a laptop between a Niles GXR-2 and a controller. The controller uses standard RJ-45 female connectors with the following pin outs:

1 TX+

2 TX-

3 RX+

4 IR Signal

5 IR Ground

6 RX-

7 UI Power (12VDC)

8 UI Ground

The transmit and receive pairs on the connectors comply with the IEEE 803.3 standard, so the transmit pair from the controller can be connected to the receive pair of the laptop using a network bridge. Do not plug the laptop directly into any of the controller ports on the GXR2, as damage to both the laptop and the GXR2 will, most likely, occur. In other words, if you don’t know how to properly do this, don’t try it.

Controllers are two-way devices. They send data to the GXR-2 using UDP protocol with assigned IP addresses in the range of 10.100.x.x. They receive data from the GXR-2 using UDP protocol with multicast addresses in the range of 232.0.0.x. All controllers receive the same multicast messages.

The Niles GXR-2 is a DHCP server and it issues a lease to each controller in the range of 10.100.0.x/16. This precludes connecting the GXR-2 to your local area network, unless you use a smart switch to filter out DHCP packets. If you make the mistake of directly connecting the GXR-2 to your network, its DHCP server will issue a lease to any device that asks for one. Since the DHCP server is not configurable and doesn’t issue a gateway, your network will start acting weird.

A simple way to control the GXR-2 is to configure a Raspberry Pi computer with an Apache web server and then connect the wired connection of the Raspberry Pi directly into the management port. The Niles GXR-2 will give the wired connection of the Raspberry Pi a DHCP lease for its 10.100.X.X network, and the wireless connection will allow other computers on the local network to access the Web server on the Raspberry Pi.

There are two kinds of traffic used for control and status in the Niles system. They are keypad commands and status update messages. Keypad commands are used to relay button presses from controllers to the GXR-2. They are sent from controllers using IP addresses that are assigned to them by the DHCP server on the GXR-2. Update messages are sent from the GXR-2 using multicast addresses. These messages are used to update display settings, such as turning on or off button illumination or updating LCD displays on System Control Keypads. Keypad commands start with 0x00 and Status update messages start with 0x03. Keyboard commands are sent Unicast using UDP from controllers to the Niles GXR-2. Status messages are sent Multicast from the Niles GXR-2 to controllers.

The basic format for a keypad command is as follows:

00

uu - Counter

00

vv - Zone ID/InputID

00

ww- unknown

00

xx-unknown

00

Yy - command

00

zz

zone IDs 21, 22, 23, 24, 25, 26

input IDs 81, 82, 83, 84, 85, 86

Commands:

01 Select first input

02 Select second input

03 Select third input

04 Select fourth input

05 Select fifth input

06 Select sixth input

0c Volume -

0d Volume +

2b Previous

2c Next

0a Off

0b Mute

There are two different types of commands sent by controllers; zone specific and device specific. Zone specific commands apply to functions like input selection and volume that are operative in a single zone. Device specific commands apply to functions, like Prev or Next, that are specific to one input across mutiple zones. When sending a zone specific command, a zone id is used to specify where the command applies. For device specific commands, a device ID is used to specify the applicable device.

The navigation controls (Prev and Next) are device specific. For them to properly work, the controller must know which device is selected. Niles handles this by updating the controller on the status of the device selected. It uses a multicast message that updates the device information in the controller. Then, the controller is able to form the control message. A typical navigation control message looks like this:

00 12 00 81 00 0b 61 06 2b 01 ff

The fourth byte is the input ID.

Input IDs for navigation controls

81 device 1

82 device 2

83 device 3

84 device 4

85 device 5

86 device 6

The ninth byte is the control code.

2b prev

2c next

Status messages

Status messages are sent using multicast addresses. Multicast messages are used because display controllers and iremote stations need status from more than one zone. Also the primary GXR2 needs to push status information to expansion devices. By making all controllers and expansion devices members of specific multicast groups, a single status messages can be targeted to all intended recipients at the same time. The Multicast address format is as follows:

232.0.0.1 - Status messages for Zone 1

232.0.0.2 - Status messages for Zone 2

232.0.0.3 - Status messages for Zone 3

232.0.0.4 - Status messages for Zone 4

232.0.0.5 - Status messages for Zone 5

232.0.0.6 - Status messages for Zone 6

The first byte of a status message is 03 Hexadecimal

The third byte is the Zone ID

21 Zone 1

22 Zone 2

23 Zone 3

24 Zone 4

25 Zone 5

26 Zone 6

Sixth byte – status message type

31 Zone on, no change

33 Zone on, input button push

15 Power button push

13 Zone off, no change

Twelth byte – device status

00 no input device selected

01 first input device selected

02 second input device selected

03 third input device selected

04 fourth input device selected

05 fifth input device selected

06 sixth input device selected