Hall Technologies • 1234 Lakeshore Dr Suite #150 Coppell, TX 75019 • halltechav.com

HIVE NODE KITS

Relay, RS-232 and IR Control over IP

API Command List

June 1, 2023

COMMAND LIST TABLE OF CONTENTS

Table of Contents

1. INTRODUCTION

The Hall Technologies TCP API provides a simple yet powerful interface to the entire family of Hall Technologies products which are based on a concept of network-connected devices having addressable modules and ports organized according by their functional class. Each functional class has a common set of API commands for implementing its functionality, and these API commands are compatible with any device in that class from any product family.

Our modular design approach provides significant benefits in flexibility and portability for API users. A utility is available to discover Hive Nodes on a network and determine each device's available modules and ports, as well as their functional classes. Because class-specific commands are compatible across all products, a single driver can be designed to work across all products.

2. NETWORK CONNECTION

Hive Nodes are network-connected products that currently support network connectivity through Ethernet. By default, all devices are configured to automatically acquire their IP configuration via DHCP. However, if a DHCP server is not available, devices will assume a default static IP address as specified in each device's documentation.

The configuration of a device's network and I/O settings can be managed through the product's configuration web pages.

For detailed information about configuration and operation of the various products from a user perspective, please refer to each product's Quick Start or User Guide.

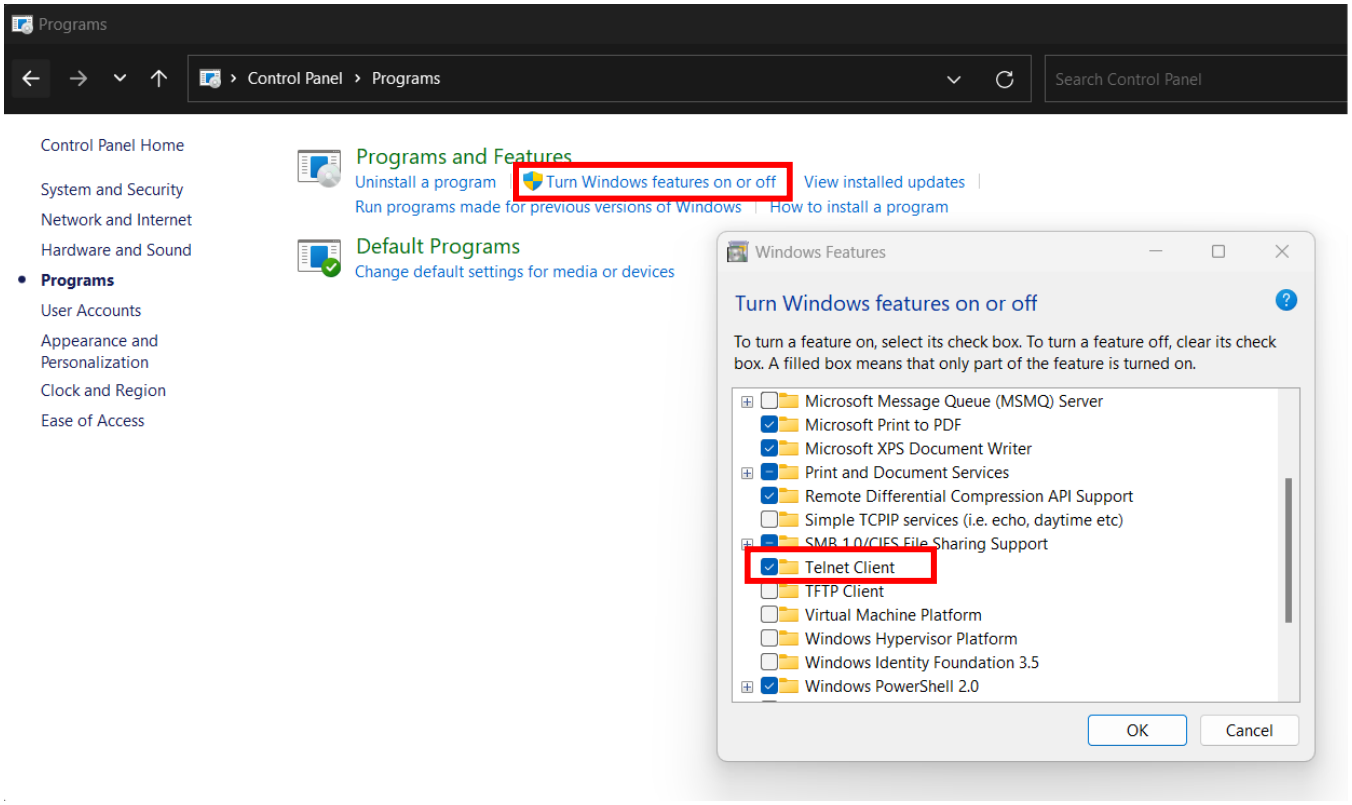
3. TCP API CONNECTION

TCP API requests and responses are transmitted over a raw TCP socket connection, specifically on port number 4998. These socket connections can be either momentary, where you send a request, receive a response, and then disconnect immediately, or persistent, where you keep a connection open for multiple requests and/or responses. The Hive-Node-Mini, for instance, can support up to eight simultaneous open connections.

ENABLING TELNET CLIENT

The Hive Node Mini can be accessed by most terminal programs such as PuTTY or Hercules, but if you are using Windows CMD then you will need to make sure it is enabled. Before logging in to IP controller via command-line interface, make sure that Telnet Client is enabled. By default, Telnet Client is disabled in Windows OS. To turn on Telnet Client, do as follows.

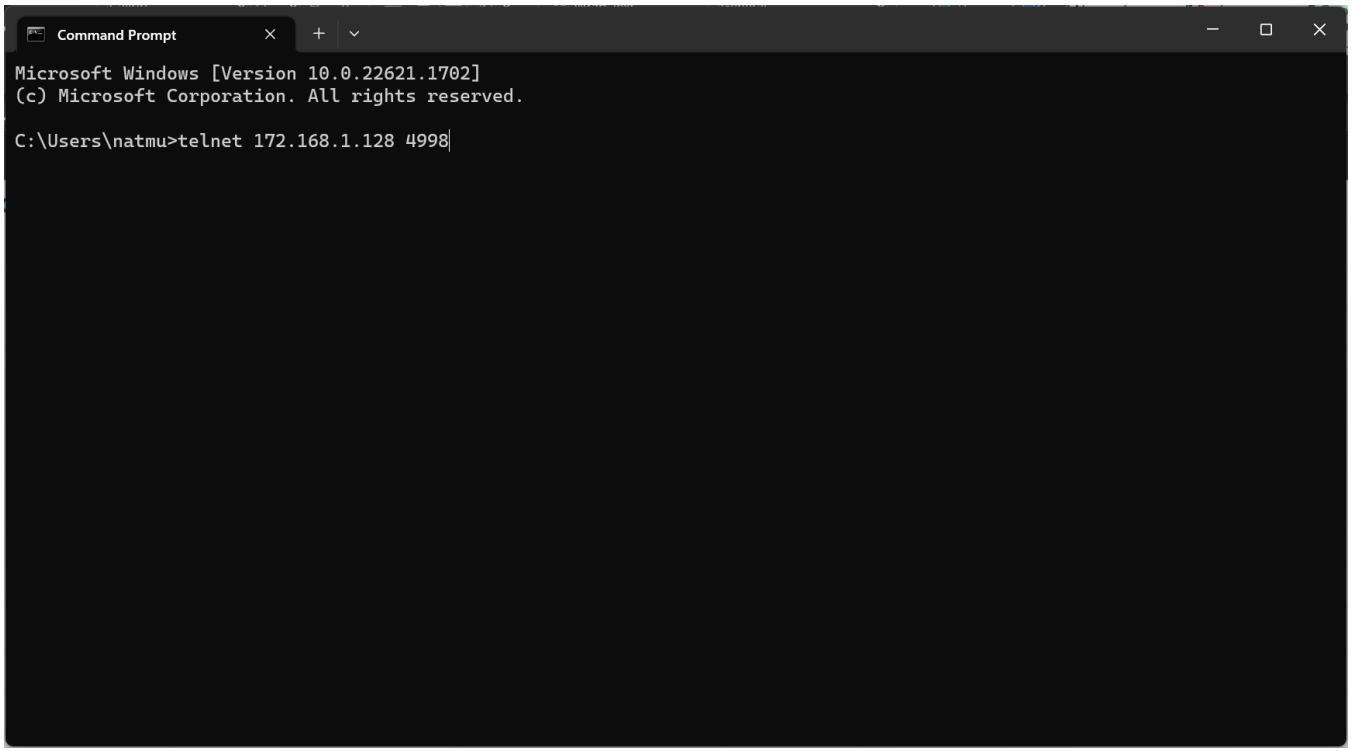
1. Choose **Start > Control Panel > Programs**
2. In the **Programs and Features** area, click **Turn Windows features on or off**.
3. In the **Windows Features** window, select **Telnet Client** check box.



LOGGING IN VIA COMMAND-LINE INTERFACE

1. Choose **Start > Run**
2. In the Run dialog box, enter **cmd** and then click **OK**.
3. Enter **telnet 172.168.1.128** if the device's IP address is 172.168.1.128 and then press **Enter**.

   (The 4998 on the end changes the port communication to 4998).



4. The device will display a **#** as the command prompt. The device is now ready to execute the API commands.

4. API REQUESTS

4.1 COMMAND/RESPONSE FORMAT

TCP API requests adhere to a straightforward serialized command/response pattern. In this exchange, a network client sends a command to the Hive-Node-Mini, which in turn sends a response back to the client.

The format for these requests is consistent, typically consisting of a single line of printable text. This line begins with a command name, often succeeded by comma-separated parameters. Responses usually mirror this format, with some exceptions where multiple lines are included in the response. All requests should conclude with a carriage-return, with responses always ending in the same manner.

Within this document, the structure of a command along with its associated parameters is consistently outlined as follows:

command,<module>:<port>,<parameter1>,<parameter2>,…,<parameterN>,[parameter]

Notes:

- A complete TCP API request is comprised of a single line of printable text ending with a carriage-return (ASCII value 13).
- Both commands and parameters are case-sensitive.
- Every request begins with a unique command name, followed either by a comma (if parameters are present) or a carriage-return.
- Required parameters are symbolized by a unique string enclosed in angle brackets (<>).
- Optional parameters are symbolized by a unique string enclosed in square brackets ([]).
- Consecutive parameters are separated by a comma (,).
- Parameter values are either specified by the client (in the request) or returned by the Hive-Node-Mini (in the response). Regardless of the context, a parameter's valid values are always restricted. Within this document, these valid values are defined either by an explicit list or a range. In both scenarios, the notation incorporates a vertical bar as follows:
    - An explicit list uses the vertical bar to separate values (e.g., 1|3|7).
    - A range separates minimum and maximum values with a vertical bar, using an ellipsis to represent any intermediate integer values (e.g., 0|…|9).

- A range separates minimum and maximum values with a vertical bar, using an ellipsis to denote any intermediate integer values (for instance, 0|…|9).

Responses to requests indicate either success or error, and may also return information such as settings or status.

The format of a success response usually echoes the command and parameters, as demonstrated below. Note that the response name typically omits the command's verb prefix (for instance, the get_NET command which responds with NET).

response,<module>:<port>,<parameter1>,<parameter2>,…,<parameterN>,[parameter]

The format of an error response can vary between different product lines but generally follows the format shown below:

<error_prefix><error_code>

5. API Commands

All device commands use TCP Port 4998 and a Carriage Return (0D). The only exception to this is when using RS-232 which will send the commands IP to port 4999 and the on directly to the RS232 port. Below are a couple of common commands you can use to verify your connection.

Hive Node Common	TCP Port: 4998	Delimiter: Carriage Return {0D}	
Command Description	Command Syntax	Example	Response
Get Version Info	getversion	getversion{0D}	710-3000-25.rc1{0D}
Get Device Info	getdevices	getdevices{0D}	device,0,0 ETHERNET{0D}device,1,1 RELAYSENSOR{0D}endlistdevices{0D}

The Hive Node Relay has 4 relays and 4 sensors. Jumpers need to be installed for them and will change how they can be used. The relays can be used as standalone SPST (most common) or grouped SPDT or DPDT. Sensors can be used as Contact Closures or Voltage Sensors. Please refer to the Manual for more information about the jumpers and config. Setting the Sensor UDP ports can be done through the web page which can then be used by creating UDP listeners at those ports.

Hive Node Relay	TCP Port: 4998	Delimiter: Carriage Return {0D}	
Command Description	Command Syntax	Example	Response
Set the Node to Relay Sensor	set_<Type>,1:1	set_RELAYSENSOR,1:1{0D}	RELAYSENSOR,1:1{0D}
Get Relay State 1-4	getstate,1:<relay 1-4 >	getstate,1:1{0D}	state,1:1,0{0D}
Get Sensor State 1-4	getstate,2:<sensor 1-4 >	getstate,2:1{0D}	state,2:1,1{0D}
Sets Relay 1-4	setstate,1:<relay 1-4 >,<state 1 or 0>	setstate,1:1,1{0D}	state,1:1,1{0D}

The Hive Node RS232 uses TCP 4998 to configure and TCP port 4999 to send strings IP to the RS232 port. It also receives RS232 strings from the connected device at the same 4999 port.

Hive Node RS232	TCP Port: 4998	Delimiter: Carriage Return {0D}	
Command Description	Command Syntax	Example	Response
Set the Node to Serial cable	set_IR,1:1,<Cable Type>	set_IR,1:1,SERIAL{0D}	IR,1:1,SERIAL{0D}
Get Serial port configuration	get_SERIAL,1:1	get_SERIAL,1:1{0D}	SERIAL,1:1,19200,FLOW_NONE,PARITY_NO,STOPBITS_1{0D}
Get Serial port configuration	get_SERIAL,1:1	get_SERIAL,1:1{0D}	SERIAL,1:1,19200,FLOW_NONE,PARITY_NO,STOPBITS_1{0D}
Set Serial port configuration	set_SERIAL,1:1,<Baud rate>,FLOW_<Type>,PARITY_<Type>,STOPBITS_<Number>	set_SERIAL,1:1,9600,FLOW_HARDWARE,PARITY_EVEN,STOPBITS_2{0D}	SERIAL,1:1,9600,FLOW_HARDWARE,PARITY_ODD,STOPBITS_2{0D}
IP to RS232 TCP Port: 4999			

The Hive Node IR uses TCP 4998 to both and to send IR commands out of the emitter or emitters. The kit comes with a tri-port connector and 3 emitters for each port. Commands can be sent discreetly to each port, or it can be configured for just one emitter connected directly to the port without the tri-port adapter. In this case the Node needs to be configured for just one port from the web page or through the API below. IR commands can be learned using a utility from the product page and the node itself which has an IR sensor.

Hive Node IR	TCP Port: 4998	Delimiter: Carriage Return {0D}	
Command Description	Command Syntax	Example	Response
Set the Node to IR type Tri-Port	set_IR,1:1,<Cable Type>	set_IR,1:1,IRTRIPORT{0D}	IR,1:1,IRTRIPORT{0D}
Set the Node to IR type Single Emmiter	set_IR,1:1,<Cable Type>	set_IR,1:1,IR{0D}	IR,1:1,IR{0D}
Send IR Command to Tri-Port Emmiter or Single Emmiter. Example is Power command for a Samsung display going out of port 1.	sendir,1:<IR Port 1,2 or 3>,<ID>,<freq>,<repeat>,<offset>,<Commands>	sendir,1:1,1,38000,1,1,172,172,22,64,22,64,22,64,22,21,22,21,22,21,22,21,22,21,22,64,22,64,22,64,22,21,22,21,22,21,22,21,22,21,22,64,22,21,22,21,22,64,22,21,22,64,22,21,22,64,22,64,22,21,22,21,22,64,22,64,22,21,22,1820{0D}	completeir,1:1,1{0D}
Send IR Command to Tri-Port Emmiter. Example is Power command for a Samsung display going out of port 3.	sendir,1:<IR Port 1,2 or 3>,<ID>,<freq>,<repeat>,<offset>,<Commands>	sendir,1:3,1,38000,1,1,172,172,22,64,22,64,22,64,22,21,22,21,22,21,22,21,22,21,22,64,22,64,22,64,22,21,22,21,22,21,22,21,22,21,22,64,22,21,22,21,22,64,22,21,22,64,22,21,22,64,22,64,22,21,22,21,22,64,22,64,22,21,22,1820{0D}	completeir,1:3,1{0D}

1234 Lakeshore Drive, Coppell, TX 75019

halltechav.com / support@halltechav.com

(714)641-6607