

TCP Control Using Service Interface

Currently our network enabled products support some limited control by way of HTTP GET requests which can be performed using similar techniques to a telnet connection but using formatted commands defined by HTTP protocol.

The below example assumes the device has been assigned the IP address 192.168.0.116 and we are acquiring the devices current status which is stored in a file labeled status.xml in a JSON format (please ignore the xml extension).

```
telnet 192.168.0.116 80
GET /status.xml HTTP/1.1
```

Please note in the example above the following

1. The first line is the telnet command to establish the connection.
2. The second line is the command being sent to the Bryston device.
 - a. GET represents the type of command being sent and in the case of Bryston equipment using Pic32 controller (which as of April 2023 represents all product with a network service interface) will only accept GET requests.
 - b. “/status.xml” is the path in which we want to execute and in this case the unit is going to return the status of the unit (seen below)
 - c. The last bit “HTTP/1.1” represents the version of HTTP commands being used and really this and the first part (“GET”) should never change.
 - d. Finally, we’ve ended the command with **two** line feed characters, this value is represented by the hex value 0xA or decimal 10.

The following is what a BR20 will reply with. The first 4 lines are generated by the web server to provide feedback to the client. 200 OK means the command executed correctly, anything other than 200 means there was a problem, and the error values are easy enough to google to obtain there meaning. In fact, you may have seen 404 not found in the past your self. The next three lines are headers generated by the server to inform the client of the content type and configuration of the server. Finally on line 5 we have the beginning of the data you’ll want to process, and, in this case, it follows the JSON format.

```
HTTP/1.1 200 OK
Connection: close
Content-Type: text/xml
Cache-Control: no-cache

{"response":
  {
    "sample_rate":
      {
        "441":
          {
            "status": "0"
          },
        "48":
          {
            "status": "0"
          }
        }
    }
  }
```

```
    },
    "882": {
      "status": "0"
    },
    "96": {
      "status": "0"
    },
    "1764": {
      "status": "0"
    },
    "192": {
      "status": "0"
    },
    "3528": {
      "status": "0"
    },
    "384": {
      "status": "0"
    },
    "64": {
      "status": "0"
    },
    "128": {
      "status": "0"
    },
    "256": {
      "status": "0"
    },
    "lock": "2"
  },
  "inputs": {
    "0": {
      "button": 0,
      "status": "2"
    },
    "1": {
      "button": 1,
      "status": "0"
    },
    "2": {
      "button": 2,
      "status": "0"
    },
    "3": {
      "button": 3,
      "status": "0"
    },
    "4": {
      "button": 4,
      "status": "0"
    },
    "5": {
      "button": 5,
```

```
        "status": "0"
    },
    "6":
    {
        "button": 6,
        "status": "0"
    },
    "7":
    {
        "button": 7,
        "status": "0"
    },
    "8":
    {
        "button": 8,
        "status": "0"
    },
    "9":
    {
        "button": 9,
        "status": "0"
    },
    "10":
    {
        "button": 10,
        "status": "0"
    },
    "11":
    {
        "button": 11,
        "status": "0"
    },
    "12":
    {
        "button": 12,
        "status": "0"
    },
    "13":
    {
        "button": 13,
        "status": "0"
    }
},
"power": "2",
"master_pwr": "2",
"power_led": "15",
"channelCount": "2",
"catnum": "10",
"volume": "180",
"upsample": "0",
"balance": "050% 12 0.0dB",
"assignment": "C231458976",
"master_mute": "0",
"options": "",
"src": "12",
"tsens": " 334"
}
}
```

Older Bryston products may use an xml formatted status rather than JSON.

Executing Commands

To tell a product to do something uses a common command that includes a unique value, and this unique value is the command being issued to the device.

```
telnet 192.168.0.116 80
GET /cmd.cgi?cmd=%2310MMUT01. HTTP/1.1
```

Let's break down the above command:

1. "GET /cmd.cgi?cmd="this is the HTTP command needed to send a command.
2. "%2310MMUT01." Is the command being sent to the unit and has the following breakdown:
 - a. "%23" is the beginning of a new command.
 - b. "10" is the device identification, this is unique to a Bryston model meaning all BR20's will be "10" and all BDA-3 and 3.14's is "30".
 - c. "MMUT01" is the command, in this case to mute the preamp's audio output.
 - d. "." The end of the command
3. "HTTP/1.1" the end of the HTTP command telling the server which version of HTTP is being used.

You will not get any meaningful feedback beyond what HTTP server responds with meaning the HTTP codes. The message received will always be "Success!" so long as the HTTP server responds 200 OK.

Examples

BR20	BDA-3/3.14	Description
%2310MMUT00.	%2330MMUT00.	Mute output
%2310MMUT01.	%2330MMUT01.	Unmute output
%2310MPWR01.	%2330MPWR01.	Power On
%2310MSRC01.	%2330MSRC01.	Input 01, AES2 on BR20 and HDMI1 on BDA3/3.14

You can find additional details for commands for the BR20 here:

<http://support.bryston.com/downloads/BR-20/Serial,%20IR,%20Hex%20Commands/BR-20%20Serial%20Commands.txt>

Additional details about commands for the BDA-3/3.14 in the BDA-3 manual on page 10

http://support.bryston.com/downloads/BDA3/documents/BDA3_MANUAL.pdf

SP3 commands can be found here:

http://support.bryston.com/downloads/sp3/documents/SP3_serial_protocols+commands.doc

