

N-Touch Panel Builder TUTORIAL

Using JavaScript and the Full Editor



The goal of this tutorial is to introduce the JavaScript® interface of the N-Touch controller. Throughout the tutorial, programming terminology is used, but little to no knowledge of JavaScript is assumed. Instead, you are encouraged to copy/paste the examples and modify them to suit your needs.

Resources for JavaScript

1. Mozilla® Developer Network (function reference):
<https://developer.mozilla.org/en-US/>
2. Stack Overflow (programmer Q&A):
<https://stackoverflow.com/>
3. W3Schools (tutorial-based help):
<http://www.w3schools.com/js/>
4. jQuery (N-Touch has this useful library available for general use):
<https://api.jquery.com/>

Resources for Module Help

1. Source code and function list are available on the N-Touch:
`http://<unit ip>/panels/modules.php`
2. Press **F12** to launch the built-in Google Chrome™ debugger on your web browser. The debugger console may show clues to the problem.
3. If using the Chrome debugger, the method `".print()"` will echo the entire object to the console.

SVSi devices list their most recent commands on the **Logs** page. Look at the **Logs** to see if the command you expected appears. You may wish to periodically reset the **Logs** to clear out stale commands.

N-Touch Controller | JasonNTouch

Settings Logs LLDP Panel Builder

Your IP address is 169.254.55.86 Refresh Logs Reset Logs

Command Log

Elapsed Time	IP	Port	Method	Command
4 hrs, 28 min, 20 sec	Web page	0	Boot	boot

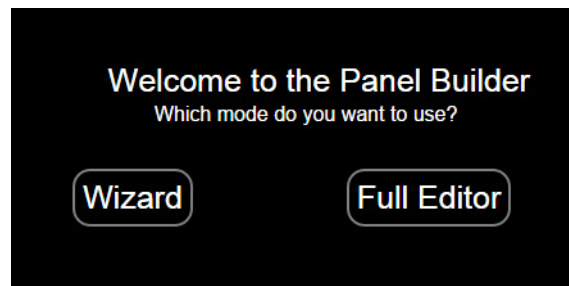
SVSi | 256.461.7143 | support@svsiav.com

For this tutorial, we are going to build a 2x2 matrix. We are also going to turn a TV off and on using serial and enabling and disabling the DVI port. We will then build a single switch button that can have a decoder toggle back and forth based on feedback. Our system will be built entirely with N-Series encoders and decoders.

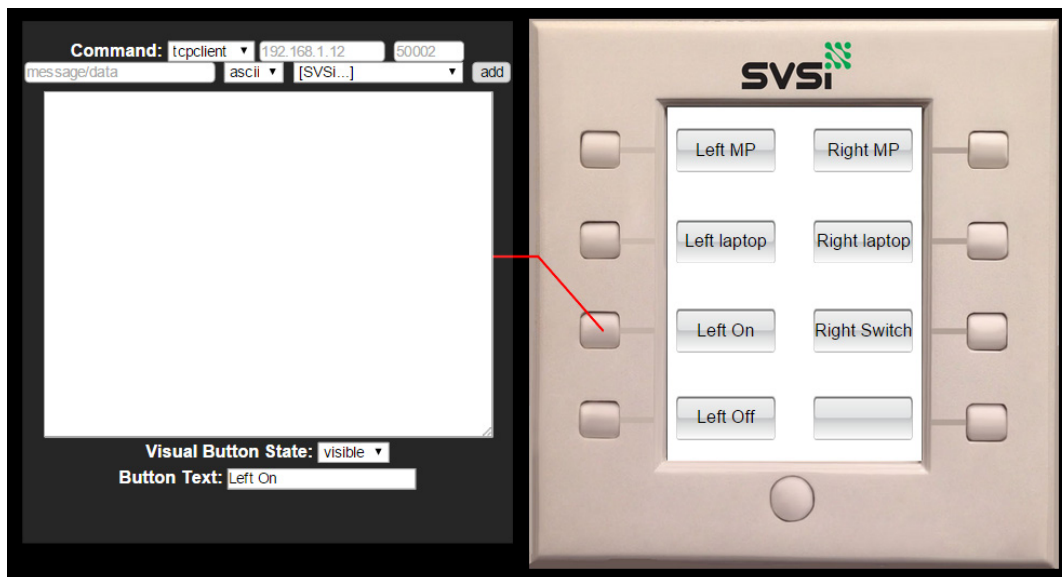


Note: This tutorial is identical whether the N-Touch is controlling an N1000, N2000, or N3000 based system.

Though most of our work will be in the **Full Editor**, the **Wizard** is a convenient way to quickly create a button grid. When you first open the **Panel Builder**, the welcome screen (shown below) appears. From this screen, you can choose to launch the **Panel Builder's** easy-to-use **Wizard** or use the **Full Editor** tool. For this part of the tutorial, choose **Wizard** to continue.

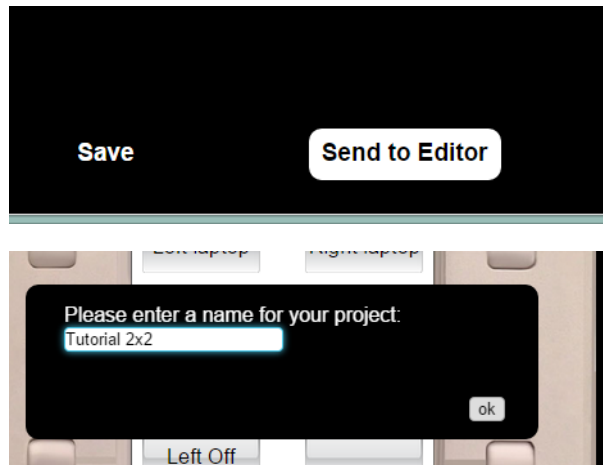


Using the **Wizard**, create the button text as follows. The scripts inside the buttons are blank for now.

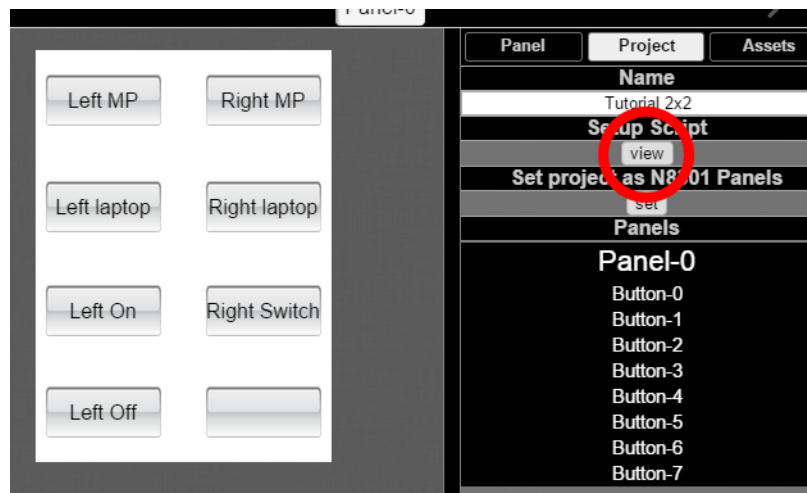


The bottom-right button is blank and reserved for your own experimentation. A common function would be to use that button to go to another panel.

Next, click **Send to Editor** in the bottom-right. The system warns you that the project will no longer be editable in the **Wizard**, and then prompts for a project name:



The project is now open in the **Full Editor**. The *Panel Builder Wizard Tutorial* only used the `tcpclient` command. A far more flexible method of control is to set up JavaScript objects of all our units. To do this, we need to create a Startup Script where we define these. See the screenshot of where to click to edit this script. The button is called **View** and it is in the far-right under the **Project** tab.



We are now going to define units in our system. Use the N-Able software to find your units. N-Able is SVSi's free device management software, available at the following link: For this tutorial, our encoders and decoders and IP addresses are as follows:

Device	Type	IP
Left TV	N2-DEC	169.254.10.25
Right TV	N2-DEC	169.254.10.26
Media Player	N2-ENC	169.254.10.27
Laptop	N2-ENC	169.254.10.28

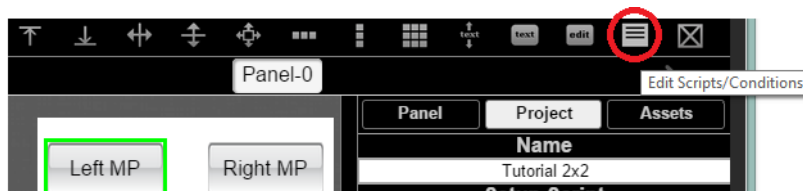
We can then create our JavaScript objects as follows. For your setup, you can copy and paste this code and change the IP addresses to match your units. Then click **Save**.

```
//Create N_Series objects of all our units
LeftTV = N_Series("169.254.10.25");
RightTV = N_Series("169.254.10.26");
MediaPlayer = N_Series("169.254.10.27");
Laptop = N_Series("169.254.10.28");
```

LeftTV, RightTV, MediaPlayer, and Laptop are now N_Series objects that can be used throughout our scripts. Defining an N_Series also calls `.getStatus()` to do a one-time update of the current status of that unit.

This is far more convenient than using raw IP addresses with the `tcpclient` function. Also, the objects have feedback and methods to make our code easier which will be shown next.

Back in the **Full Editor**, click the first button (**Left MP**) to select it, and then click the **Edit Scripts/Conditions** button. See screenshot for the button's location.



Enter (copy/paste) these scripts into the script editor and click **Save** to accept changes.

This is the script to switch LeftTV to the MediaPlayer:

```
LeftTV.switch(MediaPlayer);
```

This is the script to switch LeftTV to the Laptop:

```
LeftTV.switch(Laptop);
```

Using the `switch` script switches the stream of a decoder. It can accept a raw stream number (such as 200) or it can accept another `N_Series` object. The `.getStatus()` used earlier found the current stream number of the MediaPlayer. It then used that number for the switch command. Click **File > Save Project**, and then click **Here's a link to your project** to display the panel in your browser:



The **Left MP** and **Left laptop** buttons should now be active. Click the buttons to switch LeftTV between the two sources.

Next, we need to do the same for the RightTV. Return to the **Full Editor** and click the **Right MP** button to select. As before, choose the **Edit Scripts/Conditions** icon (in the top-right of the screen).

Enter (copy/paste) these scripts into the script editor and click **Save** to accept changes.

This is the script to switch RightTV to the MediaPlayer:

```
RightTV.switch(MediaPlayer);
```

This is the script to switch RightTV to the Laptop:

```
RightTV.switch(Laptop);
```

The code is exactly the same except the variable was swapped for RightTV. Click **File > Save Project**, and those buttons should now be active and allowing switching. The N-Touch now has a functional 2x2 matrix.

Next, we are going to use serial commands to turn the LeftTV display off and on. Go back to the Startup Script you created earlier (by clicking on the **view** button under **Project**) to define your serial on and serial off string. These are the hex commands to turn an LG® TV off and on. Your TV may be different.

```
//Turn LG TV on
LG_TV_ON = "6b 61 20 30 20 30 31 0d";
//Turn LG TV off
LG_TV_OFF = "6b 61 20 30 20 30 30 0d";
```

The Startup Script now looks like this:

```
//Create N_Series objects of all our units
LeftTV = N_Series("169.254.10.25");
RightTV = N_Series("169.254.10.26");
MediaPlayer = N_Series("169.254.10.27");
Laptop = N_Series("169.254.10.28");

//Turn LG TV on
LG_TV_ON = "6b 61 20 30 20 30 31 0d";

//Turn LG TV off
LG_TV_OFF = "6b 61 20 30 20 30 30 0d";
```

Click **Save** to accept changes.

Back in the **Full Editor**, continue (with the same process used for the other buttons) to edit the scripts for the **Left On** and **Left Off** buttons. These are the commands to send the serial hex string and turn the DVI on and off.

Left On script:

```
LeftTV.serialHex(LG_TV_ON);
LeftTV.dviOn();
```

Left Off script:

```
LeftTV.serialHex(LG_TV_OFF);
LeftTV.dviOff();
```

The `.serialHex()` is a function built in to `N_Series` objects to send a space-delimited string of hex characters from the serial port. Because `LG_TV_ON` was defined in the Startup Script, we can use a variable name instead of retyping a long string of characters. Assuming an LG TV is connected properly to the LeftTV's serial port, those buttons should now turn the TV off and on. The `dviOff()` and `dviOn()` are also built-in `N_Series` commands. Some TVs will automatically turn off and on when disconnected. If yours supports this, using these commands may be the preferred method.

With our final button (**Right Switch**) we will demonstrate feedback. **Right Switch** will toggle between our two encoders depending on the stream it is currently on. This will be the most complicated script. See below:

```
RightTV.getStatus(function() {
    if(RightTV.status.stream == MediaPlayer.status.stream) {
        RightTV.switch(Laptop);
    } else {
        RightTV.switch(MediaPlayer);
    }
});
```

The commands are defined individually below:

```
RightTV.getStatus(
This updates the status of RightTV. The getStatus() function accepts another function as a parameter. This is called a "callback function". This function gets executed when the get status is finished.

if(RightTV.status.stream == MediaPlayer.status.stream) {
```

All the statuses are available in a `.status` object of your `N_Series` object. To look at the current stream number of `RightTV`, we look at the `RightTV.status.stream`. The if statement is asking, "Does this stream number equal the stream number of the `MediaPlayer`?"

```
RightTV.switch(Laptop);
```

If the stream equals the `MediaPlayer` stream, then switch to the Laptop.

```
} else {  
Otherwise....
```

```
RightTV.switch(MediaPlayer);  
...switch to the MediaPlayer.
```

The panel is now complete. It supports a 2x2 matrix, serial commands, turning the DVI off and on, and a toggle button that looks at the feedback of a decoder.

IT SIMPLY WORKS

Distribution | Switching | Recording | Windowing | Wall Processing | Control

© Southern Vision Systems, Inc. • 256. 461.7143 • www.svsia.com

SVSi is a trademark of Southern Vision Systems, Inc. in the United States and other countries.

All other brand names, product names and trademarks are the property of their respective owners. Certain trademarks, registered trademarks, and trade names may be used to refer to either the entities claiming the marks and names or their products. SVSi disclaims any interest in the marks and names of others.



Made in the USA.

