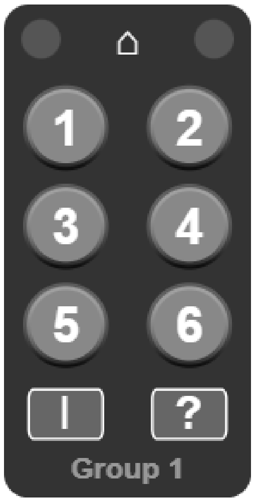
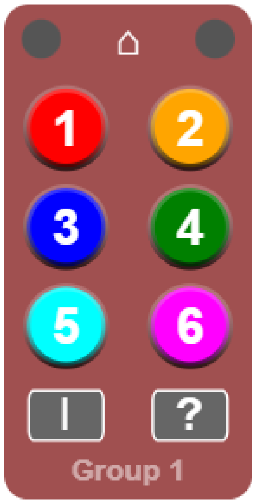
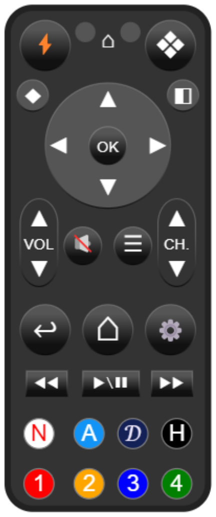
Remote Builder Help
Version 1.0
(8/14/24)

Table of Contents

Section 1 - Introduction and Installation

Introduction

Remote Builder allows you to create small but powerful "remote control" packages that can be placed on a dashboard or opened directly in the browser. In the initial release there are three modules as shown below. Each of these is covered in more detail later in the document.

Included with RB Standard	**Included with Remote Builder Advanced**	
Fixed 6 Button Remote	**Custom 6 Button Remote**	**TV Remote**
		

What are Remotes

These remotes are the digital equivalent of the ubiquitous remote controls that everyone has around their home for TV, audio, fans, lights, climate, garage doors etc. But unlike hardware based remote controls these digital equivalents can be customized in terms of their function and appearance.

Just like most physical remotes, these virtual remotes do not display status. They are intended for use within the home where the user will know if the requested action was performed or not without the need for displaying status.

Why Use Remotes

Remotes solve several problems experienced in the Hubitat environment.

1) Simple interface for discrete tasks. Hubitat dashboards can be a bit intimidating for non-technical people, but everyone knows how to use a remote control.
2) Delegation of authority. By programming a remote to perform a discrete set of tasks you can delegate authority over those devices, but only for the specified functions.
3) No Hubitat app or dashboard required. These remotes can load directly within the browser and will scale to fit the device.
4) A customizable interface. You can change the icons, icon colors and background colors used on the remote to your own liking.
5) Tooltips. Each button has a customizable tooltip which indicates the function of the button.

A Few Scenarios

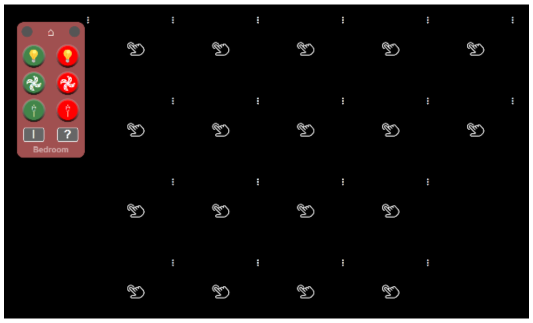
Your kids have some smart devices in their rooms, and you want an easy way to give them control over just those devices.

You have guests coming to stay and your guest bedroom has some automated elements. You can make a remote for changing the lights, fan, temperature, blinds, TV or whatever else you might have and share the cloud endpoint with your guests. They do not even need to access your WiFi.

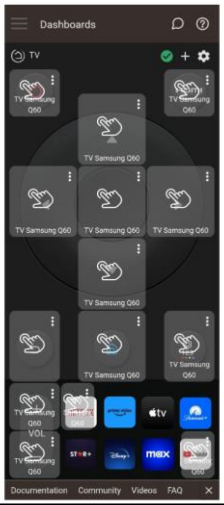
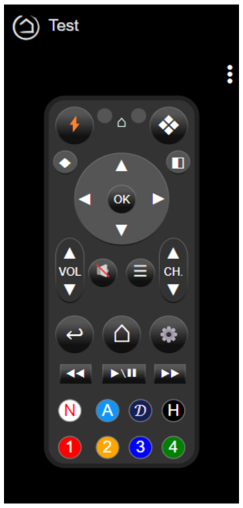
You have a cat\dog sitter that comes over to feed or walk your animals. You could make a remote that allows them to unlock the front door, disable the alarm and turn on the light when they arrive and then reverse it all when they leave.

You have some physical button remotes around the home. Now you can add a virtual remote that produces the exact same actions including the double click or long press

Your dashboard is cluttered with buttons. They take up too much space and they are hard to differentiate. The picture below shows 18 buttons (1x1) compared to an equivalent virtual remote (1x2) which has 6 buttons in three groups.



You would like to put your TV\Audio on a dashboard in a way that does not require a kludgy solution.

A background image plus 20 buttons.	One link added to dashboard.
 Dashboards TV TV Samsung Q60 VOL Documentation Community Videos FAQ	

How Remote Builder Works

In the initial release there are five components to Remote Builder of which two are required elements and three are optional elements.

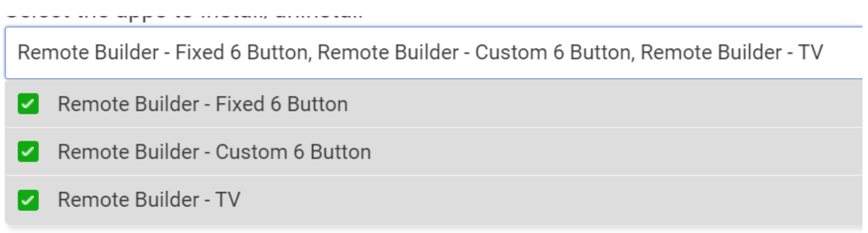
Required Elements

1) Remote Builder Parent App
2) Remote Builder Storage Driver – Device driver used for storing Remotes data.

Optional Elements

3) **Remote Builder Fixed 6 Button** (child app) – A 6 button whose actions can be customized but whose appearance cannot.
4) **Remote Builder Custom 6 Button** (child app) – A 6 button remote with the 3 button groups (18 buttons total) whose appearance can be customized.
5) **Remote Builder TV** (child app) - A 29 button TV remote including 10 customizable buttons that can point to unique functions or even to entirely different devices.

When you install Remote Builder from HPM be sure to select the modules you want installed!



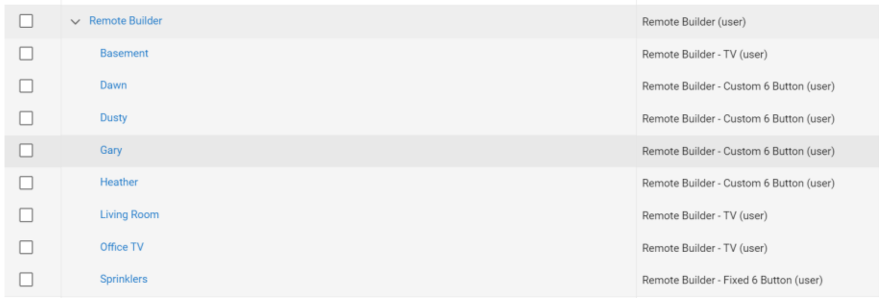
Remote Builder - Fixed 6 Button, Remote Builder - Custom 6 Button, Remote Builder - TV

✓ Remote Builder - Fixed 6 Button

✓ Remote Builder - Custom 6 Button

✓ Remote Builder - TV

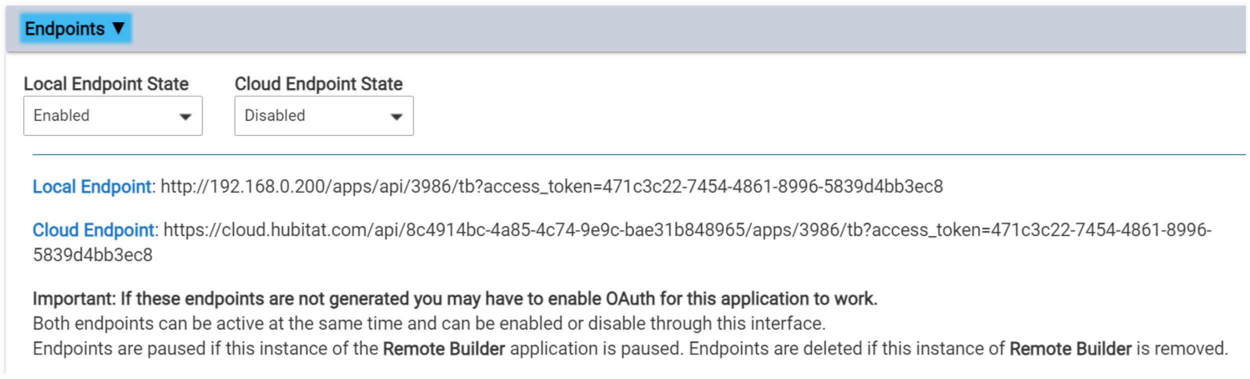
The Remote Builder parent app is the primary organizing app.



☐	Remote Builder	Remote Builder (user)
☐	Basement	Remote Builder - TV (user)
☐	Dawn	Remote Builder - Custom 6 Button (user)
☐	Dusty	Remote Builder - Custom 6 Button (user)
☐	Gary	Remote Builder - Custom 6 Button (user)
☐	Heather	Remote Builder - Custom 6 Button (user)
☐	Living Room	Remote Builder - TV (user)
☐	Office TV	Remote Builder - TV (user)
☐	Sprinklers	Remote Builder - Fixed 6 Button (user)

Endpoints

Hubitat uses endpoints to publish information both within the local network and to the cloud. Remote Builder uses these endpoints to publish these remotes, so they are accessible to anyone with the network address. Each Remote Builder module has an Endpoints section where the local and cloud endpoints are available. Endpoints can also be enabled or disabled here.



Local Endpoint State: Enabled

Cloud Endpoint State: Disabled

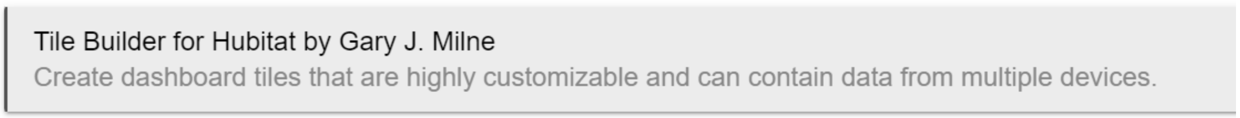
Local Endpoint: http://192.168.0.200/apps/api/3986/tb?access_token=471c3c22-7454-4861-8996-5839d4bb3ec8

Cloud Endpoint: https://cloud.hubitat.com/api/8c4914bc-4a85-4c74-9e9c-bae31b848965/apps/3986/tb?access_token=471c3c22-7454-4861-8996-5839d4bb3ec8

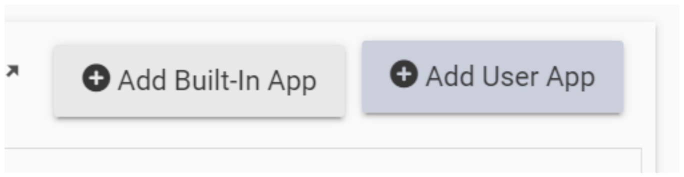
Important: If these endpoints are not generated you may have to enable OAuth for this application to work.
Both endpoints can be active at the same time and can be enabled or disable through this interface.
Endpoints are paused if this instance of the **Remote Builder** application is paused. Endpoints are deleted if this instance of **Remote Builder** is removed.

Remote Builder Installation

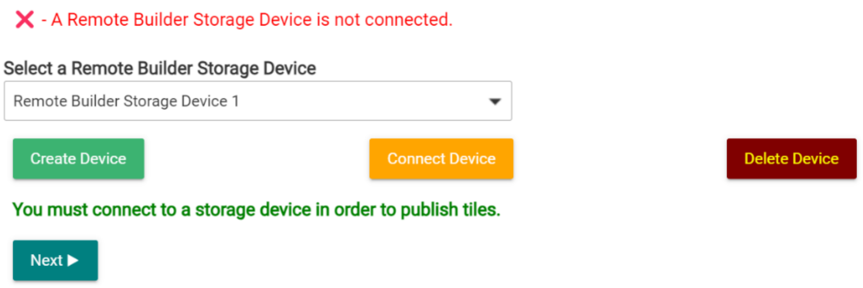
Remote Builder is listed in Hubitat Package manager. Choose to install by tags and select the **Dashboards** tag. Select **Remote Builder for Hubitat** and complete the installation process. This will place the code on your hub and then there are a few steps to complete the installation.


Tile Builder for Hubitat by Gary J. Milne
Create dashboard tiles that are highly customizable and can contain data from multiple devices.

1. Go to the Apps tab and click on **Add User App**



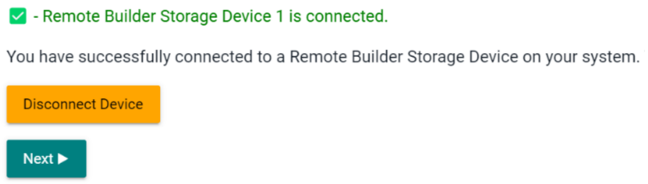
2. Select **Remote Builder** from the list of available apps.
3. **Remote Builder** will install and bring you to the parent screen.
4. **Select the appropriate License Type (see paragraph below on licensing).**
5. Under **Device Creation** you must first create the storage device and then connect the app to the device. Just use the default device for now.


❌ - A Remote Builder Storage Device is not connected.

Select a Remote Builder Storage Device: Remote Builder Storage Device 1

You must connect to a storage device in order to publish tiles.

6. Once the device is created and connected it will look like this.



✓ - Remote Builder Storage Device 1 is connected.

You have successfully connected to a Remote Builder Storage Device on your system.

7. We can now create our first remote.

Licensing

Remote Builder has Standard and Advanced versions just like **Tile Builder** and shares the same key. If you already have a Tile Builder key you can use it on your Remote Builder installation to unlock the Advanced version.

In the first release the **Standard** version only gives you access to the **Fixed 6 Button Remote** but you can create as many of these as you wish. The **Advanced** version adds the Custom 6 Button Remote and the TV Remote.

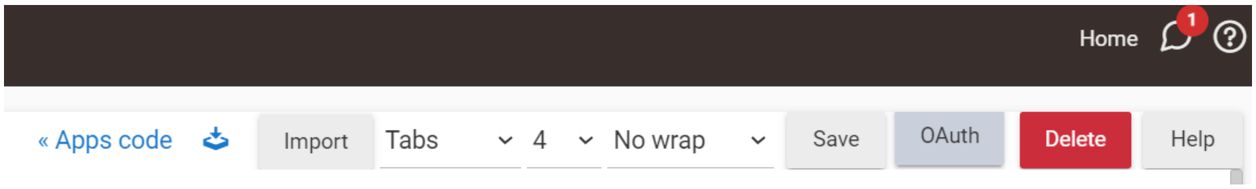
To obtain an Advanced license for Remote Builder and Tile Builder go to the licensing section in the parent app and follow the instructions. When you purchase a license, it allows me to work on developing new and improved modules for the betterment of Hubitat vs working a part time job to generate the same income.

If you don’t want to pay for an Advanced license you are still free to use the Standard version as much as you wish.

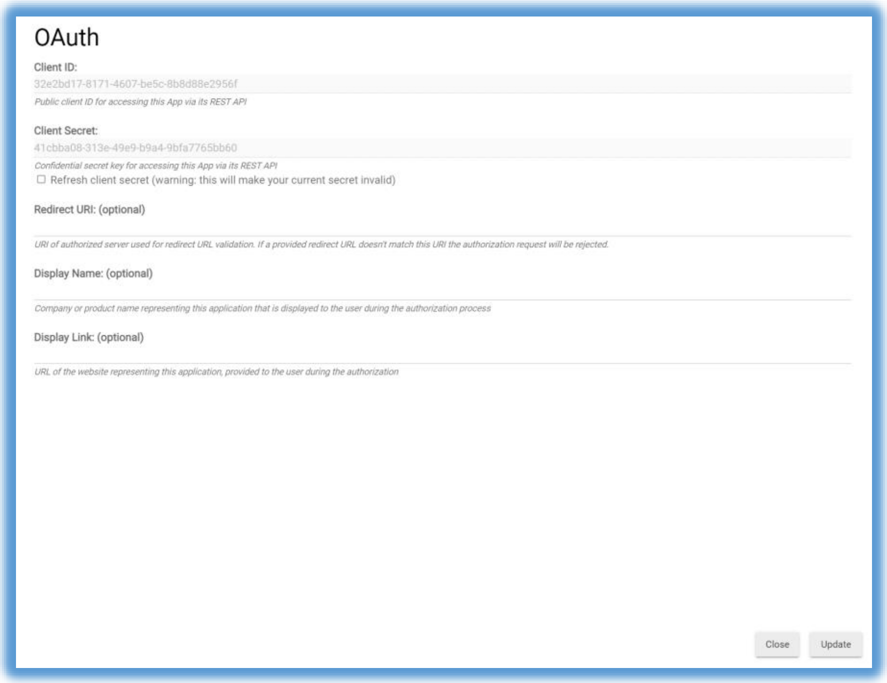
Section 2 - Remote Builder Standard

Enabling OAuth

Important! – Before you can create a Remote you must enable OAuth in the code for each module that you wish to use. To do this go to: Developer Tools \ Apps Code \ Remote Builder – Fixed 6 Button (for example) and in the upper right corner click on OAuth



You will get a screen that looks like this. Click on Update and you are all done.



OAuth

Client ID:
32e2bd17-8171-4607-be5c-8b8d88e2956f
Public client ID for accessing this App via its REST API

Client Secret:
41cbba08-313e-49e9-b9a4-9bfa7765bb60
Confidential secret key for accessing this App via its REST API
☐ Refresh client secret (warning: this will make your current secret invalid)

Redirect URI: (optional)

URI of authorized server used for redirect URL validation. If a provided redirect URL doesn't match this URI the authorization request will be rejected.

Display Name: (optional)

Company or product name representing this application that is displayed to the user during the authorization process

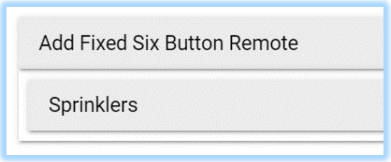
Display Link: (optional)

URL of the website representing this application, provided to the user during the authorization

You must repeat this for each Remote Builder child app code instance that you install.

Creating a Fixed 6 Button Remote

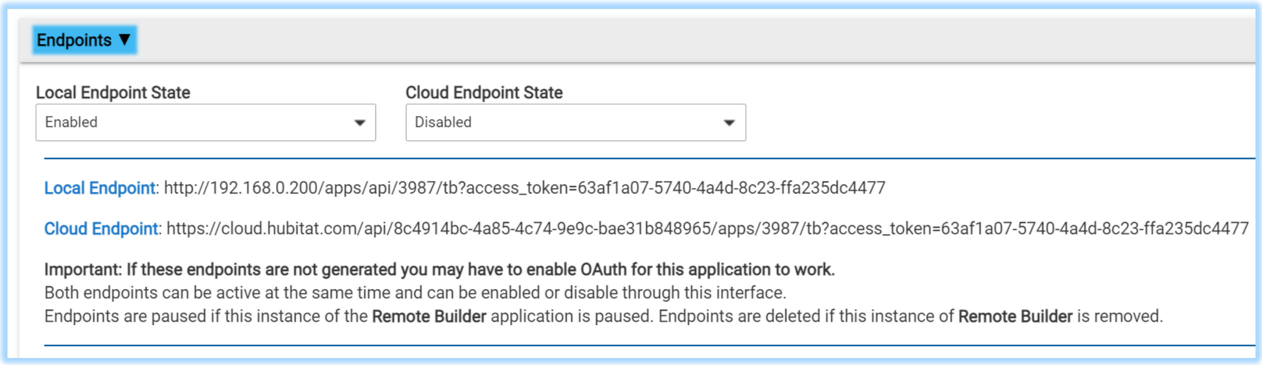
Within the **Remote Builder** parent app go to the section called **Create \Edit Remotes** and select Add **Fixed Six Button Remote**. The **Fixed 6 Button Remote** main screen will be displayed.



The app display is divided into 5 sections.

Introduction: A brief description of the program you are running and what it can do.

Endpoints: Where the generated remote can be accessed as well as being enabled\disabled.



Local Endpoint State: Enabled

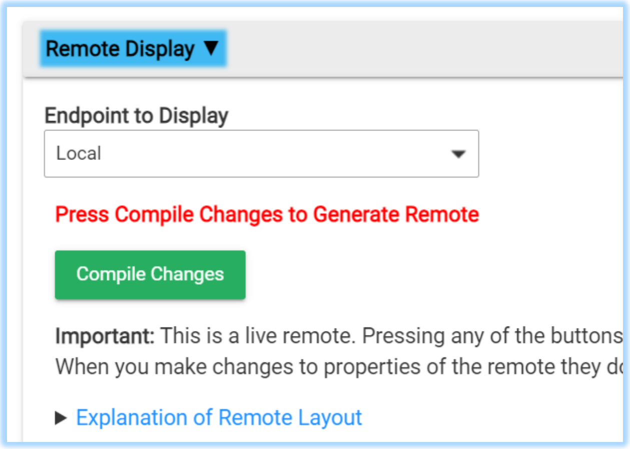
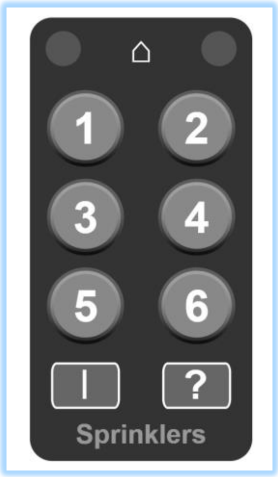
Cloud Endpoint State: Disabled

Local Endpoint: http://192.168.0.200/apps/api/3987/tb?access_token=63af1a07-5740-4a4d-8c23-ffa235dc4477

Cloud Endpoint: https://cloud.hubitat.com/api/8c4914bc-4a85-4c74-9e9c-bae31b848965/apps/3987/tb?access_token=63af1a07-5740-4a4d-8c23-ffa235dc4477

Important: If these endpoints are not generated you may have to enable OAuth for this application to work.
Both endpoints can be active at the same time and can be enabled or disable through this interface.
Endpoints are paused if this instance of the **Remote Builder** application is paused. Endpoints are deleted if this instance of **Remote Builder** is removed.

Remote Display: This is where the generated remote can be previewed and tested. The actual remote is stored in a compiled form for efficiency. When changes are made, they must be compiled to take effect.

	Compiled Fixed Remote
	

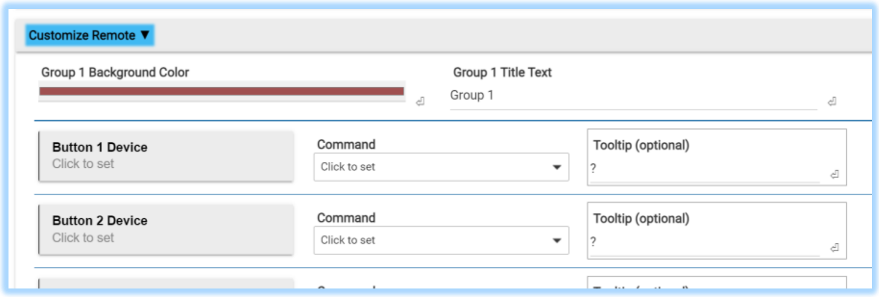
Endpoint to Display: Local

Press Compile Changes to Generate Remote

Important: This is a live remote. Pressing any of the buttons
When you make changes to properties of the remote they d

▶ Explanation of Remote Layout

Customize Display: This is where you select the device and actions that will be generated by a button press as well as the Tooltip that is displayed.



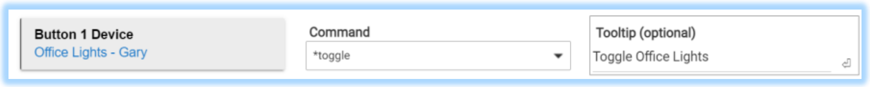
Customize Remote ▼

Group 1 Background Color

Group 1 Title Text: Group 1

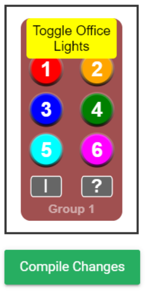
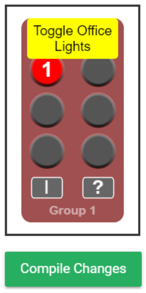
Button 1 Device: Click to set	Command: Click to set	Tooltip (optional): ?
Button 2 Device: Click to set	Command: Click to set	Tooltip (optional): ?

After a device\action is selected it looks like this.



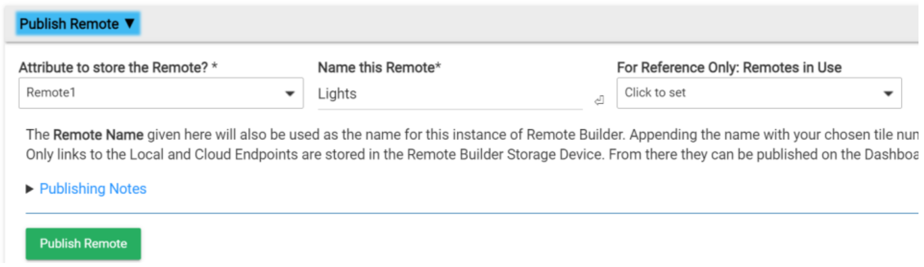
Button 1 Device: Office Lights - Gary	Command: *toggle	Tooltip (optional): Toggle Office Lights

After you Compile the remote it looks like this when you mouse over button 1.

Disabled Unassigned Buttons: **False**	Disabled Unassigned Buttons: **True**
	

Publish Remote: If you are familiar with Tile Builder you will know that publishing the output to a tile is a necessary step to viewing the output. With **Remote Builder** publishing a remote to an attribute is optional, this is because Remotes work perfectly well with just the Endpoint address.

To publish a remote, select the attribute prefix to publish it under. Attributes begin with the prefix Remote1 – Remote25. You can then click on the Publish Remote button.



Publish Remote ▼

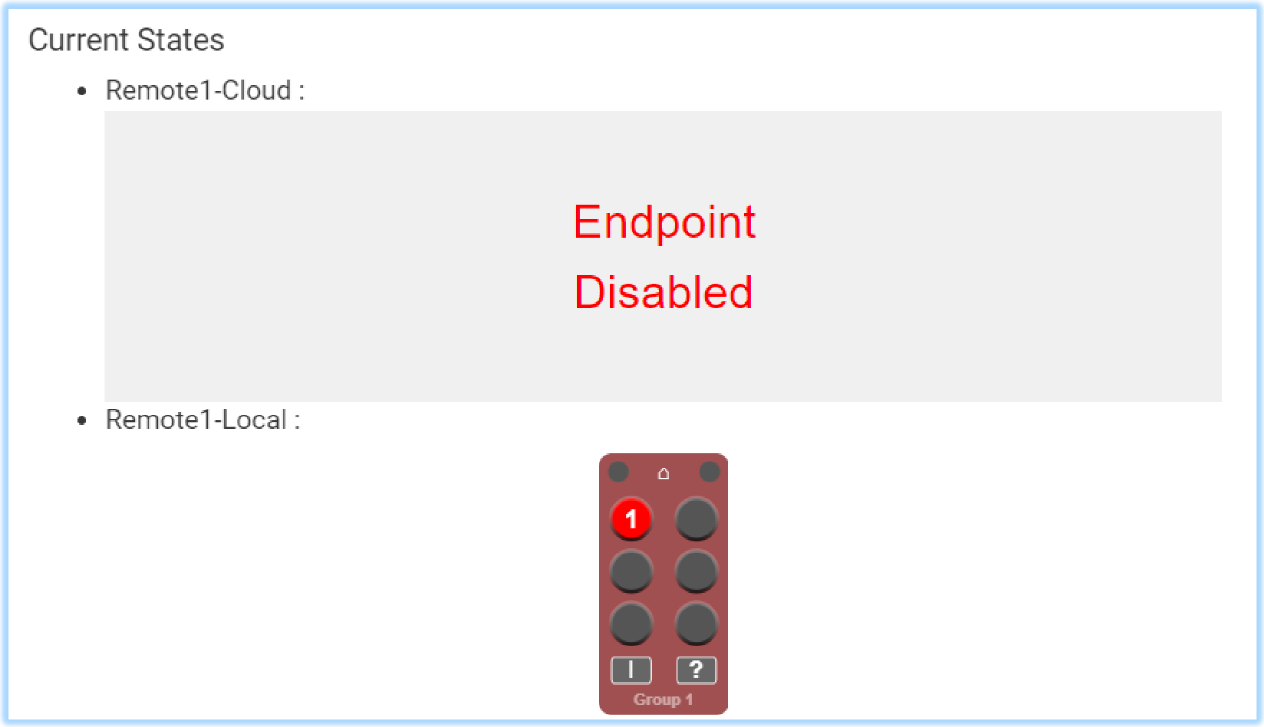
Attribute to store the Remote? *	Name this Remote*	For Reference Only: Remotes in Use
Remote1	Lights	Click to set

The **Remote Name** given here will also be used as the name for this instance of Remote Builder. Appending the name with your chosen tile nun
Only links to the Local and Cloud Endpoints are stored in the Remote Builder Storage Device. From there they can be published on the Dashboa

▶ Publishing Notes

Publish Remote

If we look within the **Remote Builder Storage Device** we will find that the attributes **Remote1-Local** and **Remote1-Cloud** are populated with links to the remote that we generated.


Current States

- Remote1-Cloud :

Endpoint Disabled

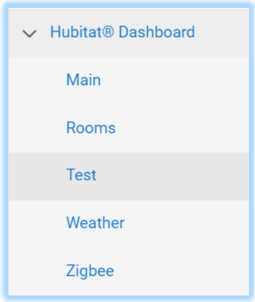
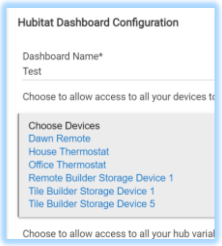
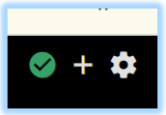
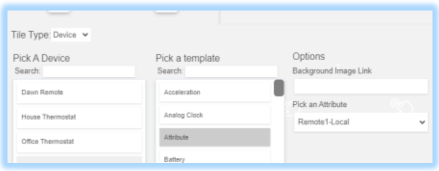
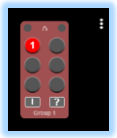
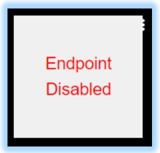
- Remote1-Local :

Note: This is a live preview of the Endpoint. This remote is active and can be clicked on to execute commands so be careful where you click.

Accessing the Remote

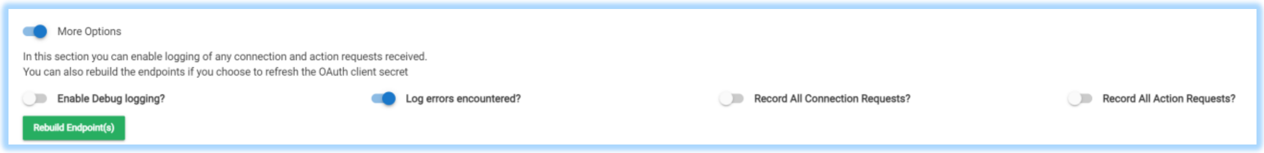
There are 2 typical ways to access the remote. The first is by using the links provided in the Endpoints section. These can be stored as favorites or shared via email or text.

The other method is to add the remote device to a Hubitat Dashboard. This is how you do that.

Go the desired Hubitat Dashboard entry listed under Apps. It will look something like this.	 Hubitat® Dashboard Main Rooms Test Weather Zigbee
Now add the **Remote Builder Storage Device** to your list of allowed devices so it looks like this.	 Hubitat Dashboard Configuration Dashboard Name* Test Choose to allow access to all your devices to Choose Devices Dawn Remote House Thermostat Office Thermostat Remote Builder Storage Device 1 Tile Builder Storage Device 1 Tile Builder Storage Device 5 Choose to allow access to all your hub varial
Now go to the Dashboard you selected and click the + sign to add a device.	
Pick the Remote Builder Storage Device. Select attribute and then the Remote you wish to publish. In this case it is **Remote1-Local.**	 Tile Type: Device Pick A Device Search: Dawn Remote House Thermostat Office Thermostat Pick a template Search: Acceleration Analog Clock Attribute Battery Options Background Image Link Pick an Attribute Remote1-Local
Once you have done that it will appear on the Dashboard in the position and size you have selected. Remotes are built with Vector Graphics so they can be scaled to whatever size you wish.	
If the Endpoint is disabled it will look like this. You can enable or disable the Endpoint using the Remote Builder App instance for this remote.	

Logging

Under the section **More Options** you can enable\disable the logging of connections to the Endpoint as well as actions initiated by the remotes.

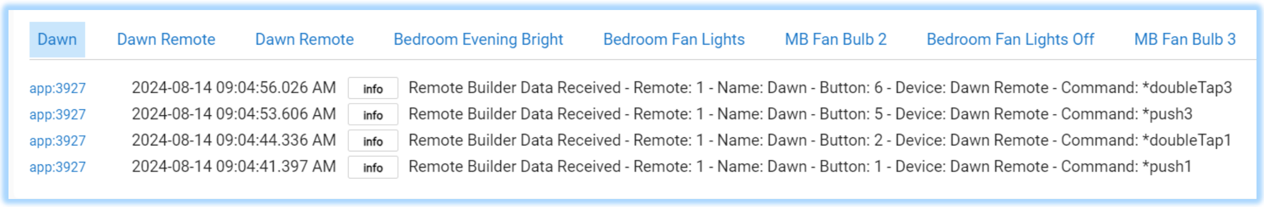


More Options

In this section you can enable logging of any connection and action requests received.
You can also rebuild the endpoints if you choose to refresh the OAuth client secret

Enable Debug logging? — off
Log errors encountered? — on
Record All Connection Requests? — off
Record All Action Requests? — off

With “**Record All Action Requests**” enabled the log will look something like this.



```
app:3927   2024-08-14 09:04:56.026 AM  info  Remote Builder Data Received - Remote: 1 - Name: Dawn - Button: 6 - Device: Dawn Remote - Command: *doubleTap3
app:3927   2024-08-14 09:04:53.606 AM  info  Remote Builder Data Received - Remote: 1 - Name: Dawn - Button: 5 - Device: Dawn Remote - Command: *push3
app:3927   2024-08-14 09:04:44.336 AM  info  Remote Builder Data Received - Remote: 1 - Name: Dawn - Button: 2 - Device: Dawn Remote - Command: *doubleTap1
app:3927   2024-08-14 09:04:41.397 AM  info  Remote Builder Data Received - Remote: 1 - Name: Dawn - Button: 1 - Device: Dawn Remote - Command: *push1
```

Debug logging is really intended for use by the developer or at the request of the developer in a troubleshooting scenario.

Section 3 - Remote Builder Advanced

Introduction

The Advanced version of Remote Builder contains two modules at the initial launch of the product. The first one is the **Custom 6 Button Remote** and the second is the **TV Remote**.

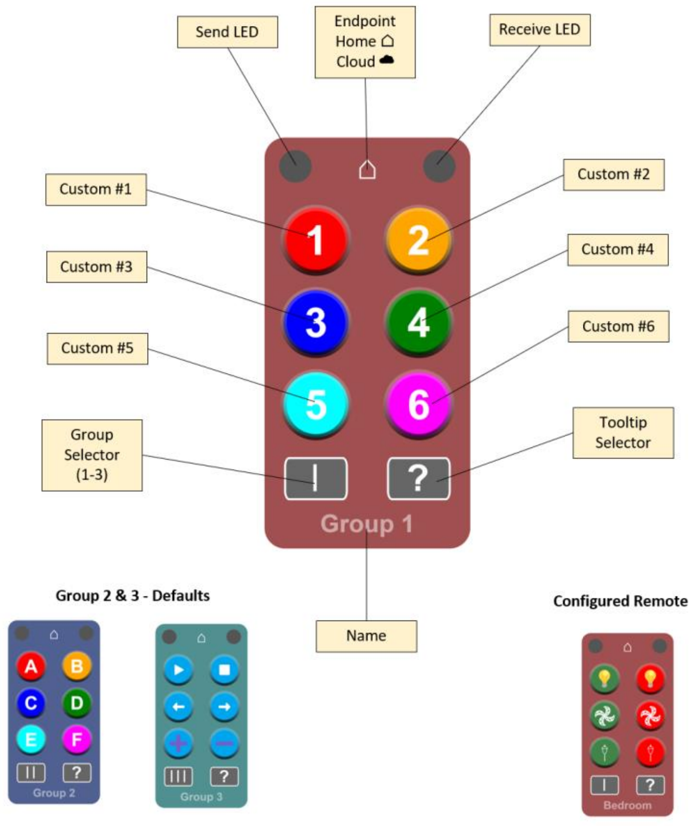
This section covers the use of these two modules but it does not repeat some of the basic elements such as publishing and logging that have already been discussed in the Remote Builder Standard section.

Creating a Custom 6 Button Remote

This section assumes you have already read the **Creating a Fixed Six Button Remote** section and only discusses those parts of the process which are different.

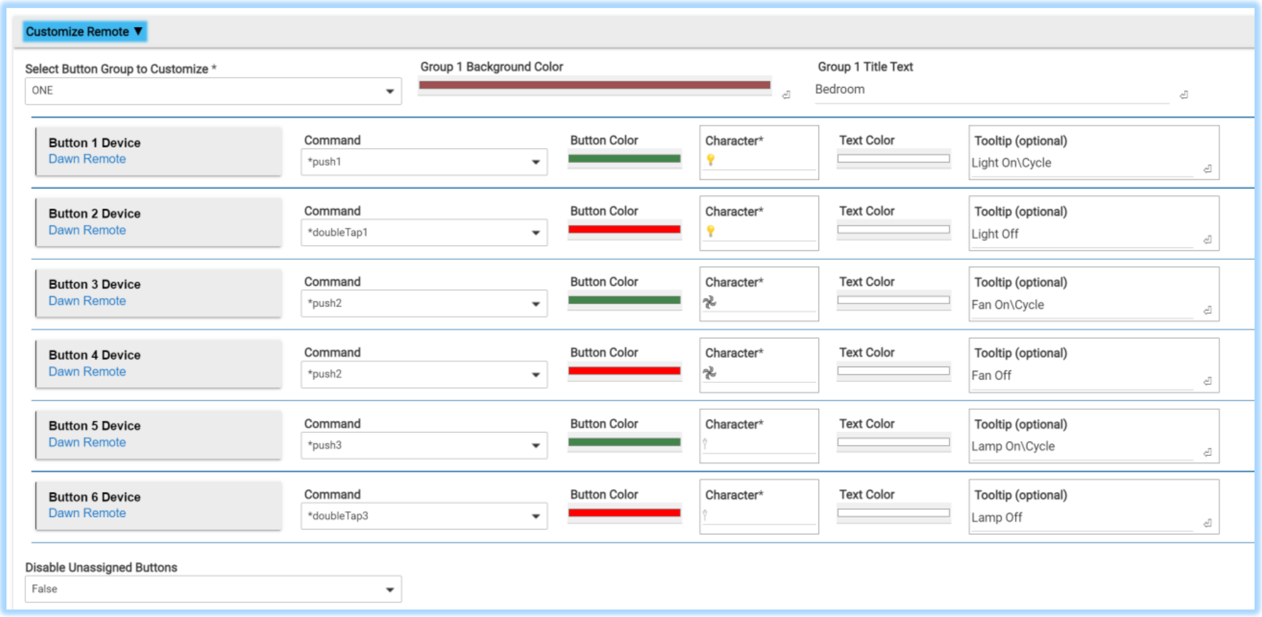
As the title suggests, this type of remote provides a lot more customization options than its **Fixed** sibling. The custom remote also offers one other big advantage, which is that it supports 3 button groups effectively making it an 18-button remote.

Overview of 6 Button Custom Remote



Remote Customization Options

The diagram below illustrates the various customization options available for each button group.



Customize Remote ▼

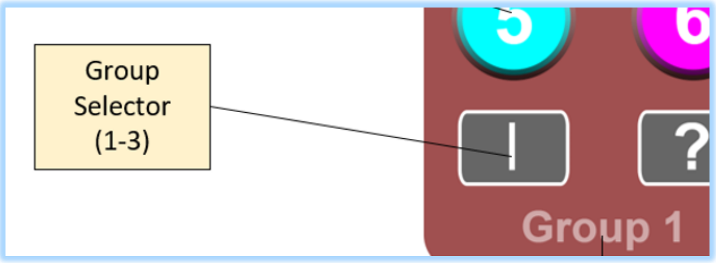
Select Button Group to Customize *	Group 1 Background Color	Group 1 Title Text
ONE	🟫	Bedroom

	Command	Button Color	Character*	Text Color	Tooltip (optional)
Button 1 Device Dawn Remote	*push1	🟩	💡	⬜	Light On\Cycle
Button 2 Device Dawn Remote	*doubleTap1	🟥	💡	⬜	Light Off
Button 3 Device Dawn Remote	*push2	🟩		⬜	Fan On\Cycle
Button 4 Device Dawn Remote	*push2	🟥		⬜	Fan Off
Button 5 Device Dawn Remote	*push3	🟩		⬜	Lamp On\Cycle
Button 6 Device Dawn Remote	*doubleTap3	🟥		⬜	Lamp Off

Disable Unassigned Buttons: False

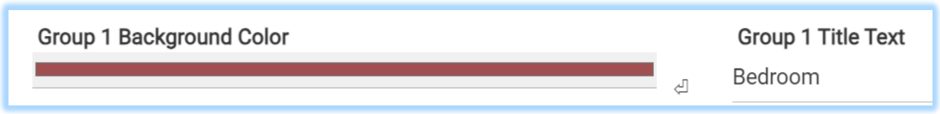
Button Groups

A Custom 6 Button Remote has three button groups for a total of 18 buttons. You can toggle through these three button groups using the Group Selector button as shown below.



Other Settings

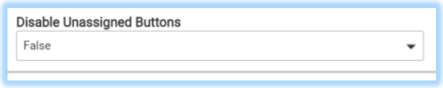
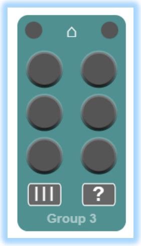
You can change the background color for each button group and give it a name.



Group 1 Background Color

Group 1 Title Text

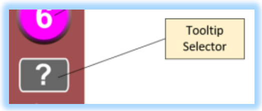
Bedroom

Disable Unassigned Buttons You can choose to disable those buttons that do not have a valid device\command assigned.	 Disable Unassigned Buttons: False
In this case none of the buttons on Group 3 are assigned so they all appear greyed out.	

ToolTip Selector

When using a device that has a pointer device such as a mouse, placing the cursor over one of the action buttons will cause the assigned tooltip to be revealed. This acts as a reminder of what action will be initiated by pressing the button.

But most phones and tables do not have a pointing device that can be used to reveal the Tooltip. For touch devices the Tooltip can still be revealed using the Tooltip Selector button shown below.



The Tooltip Selector acts as a toggle. The picture above shows it in it’s normal state. When it is active the question mark is displayed in green. Once the Tooltip Selector is activated, clicking on any of the buttons will display the tooltip, but does not invoke the assigned action. The Tooltip selector remains in this state until it is toggled off again.

Synthetic Commands

If you have tried out Remote Builder you may have noticed that some of the commands begin with an * and may look like *toggle as in the example above.

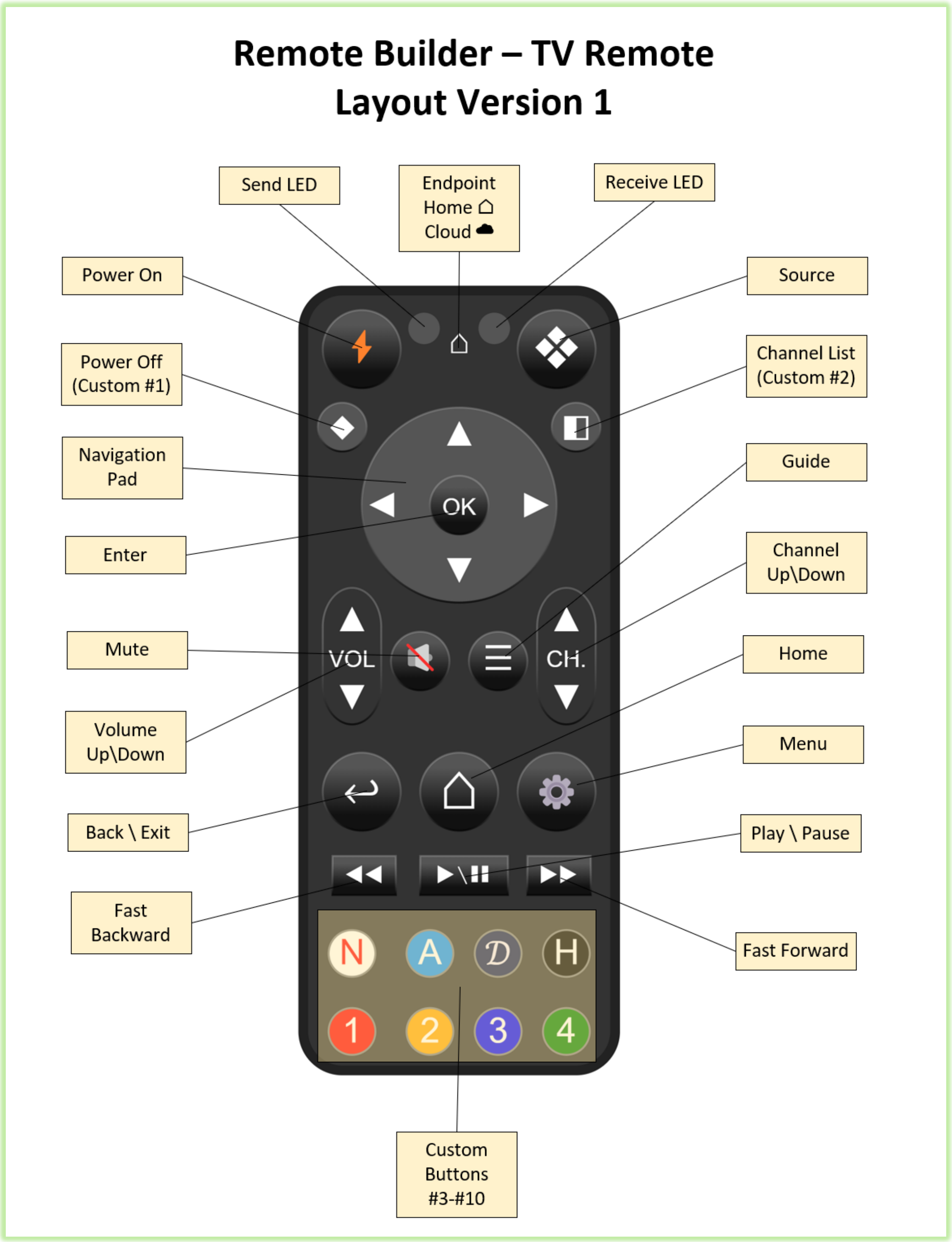
If remote builder sees a device with On\Off commands but it does not have a toggle command then it inserts a synthetic toggle into the available command list and prefixes it with an * so it appears at the top.

Remote Builder also adds synthetic commands for button controllers by adding *push1 - *push4 and *doubleTap1 - *doubleTap4. This allows these commands to be implemented via a single button push even though the commands push and doubleTap require a parameter.

The result is that a single button can be used to turn any device on and off vs requiring two buttons, one to turn on and the other to turn off.

Creating a TV Remote

The TV Remote is just a virtual rendition of a physical TV remote that we are all used to. The default layout is shown in the diagram below.

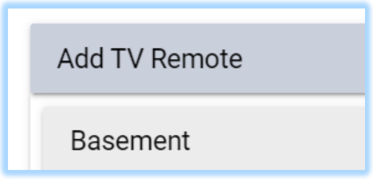


Requirements

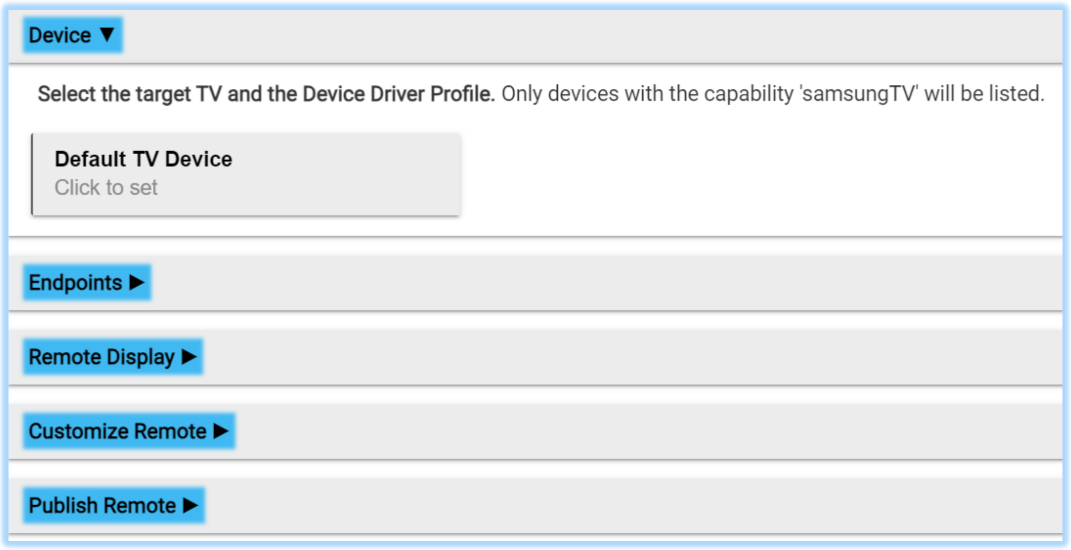
This remote is designed specifically to control a TV and can only operate in conjunction with a device driver that has implemented the SamsungTV capability. Before you can use this remote you must be sure that your Samsung TV is already part of your Hubitat device list and is responding to commands.

Create a New TV Remote

Go into the Remote Builder Parent App and click on **Add TV Remote** to create a new TV remote.



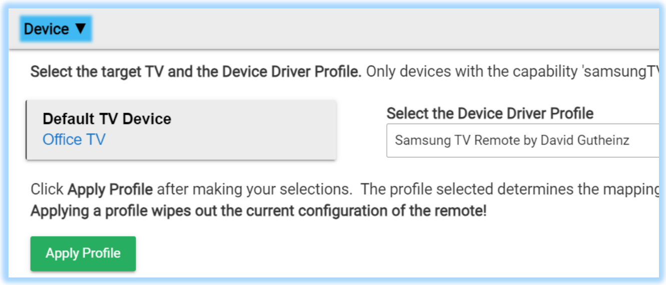
You will now see the app to configure a new TV remote which looks like this.



Select the target TV and the Device Driver Profile. Only devices with the capability 'samsungTV' will be listed.

Select TV Device

Select a TV device. If your list is empty, you don't have one configured within your environment and this TV Remote is of no use to you.



Device ▼

Select the target TV and the Device Driver Profile. Only devices with the capability 'samsungT

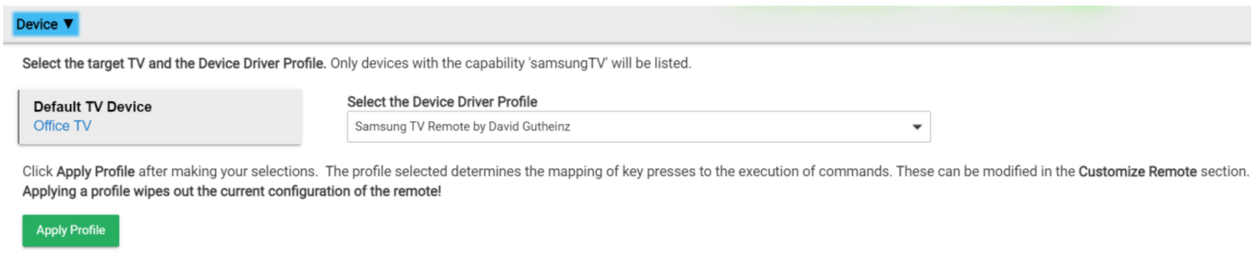
Default TV Device
Office TV

Select the Device Driver Profile
Samsung TV Remote by David Gutheinz

Click **Apply Profile** after making your selections. The profile selected determines the mapping
Applying a profile wipes out the current configuration of the remote!

Apply Profile

The only device driver supported on release is the Samsung TV Remote driver by David GutHeinz and it is automatically selected. Others will be added in future versions if they are adequately in demand and supported.



Device ▼

Select the target TV and the Device Driver Profile. Only devices with the capability 'samsungTV' will be listed.

Default TV Device
Office TV

Select the Device Driver Profile
Samsung TV Remote by David Gutheinz

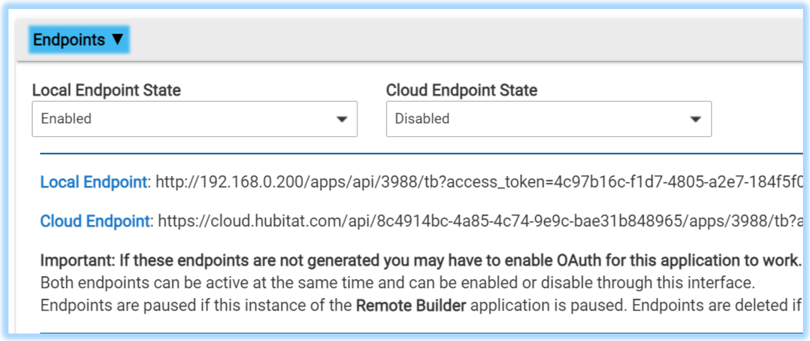
Click **Apply Profile** after making your selections. The profile selected determines the mapping of key presses to the execution of commands. These can be modified in the **Customize Remote** section.
Applying a profile wipes out the current configuration of the remote!

Apply Profile

Click on **Apply Profile** to generate the initial remote. Your remote is now active using default settings.

Endpoints

Your endpoints are automatically created and are accessible under the Endpoints section. By default only the local endpoint is enabled which means it is only accessible via your local network.



Endpoints ▼

Local Endpoint State: Enabled

Cloud Endpoint State: Disabled

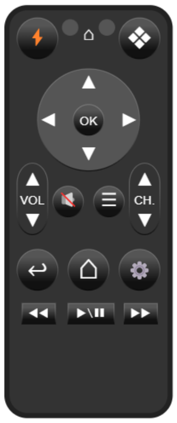
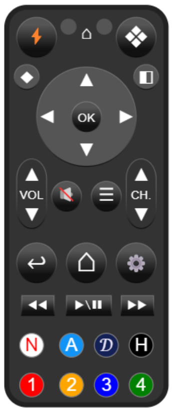
Local Endpoint: http://192.168.0.200/apps/api/3988/tb?access_token=4c97b16c-f1d7-4805-a2e7-184f5f0

Cloud Endpoint: https://cloud.hubitat.com/api/8c4914bc-4a85-4c74-9e9c-bae31b848965/apps/3988/tb?a

Important: If these endpoints are not generated you may have to enable OAuth for this application to work.
Both endpoints can be active at the same time and can be enabled or disable through this interface.
Endpoints are paused if this instance of the **Remote Builder** application is paused. Endpoints are deleted if

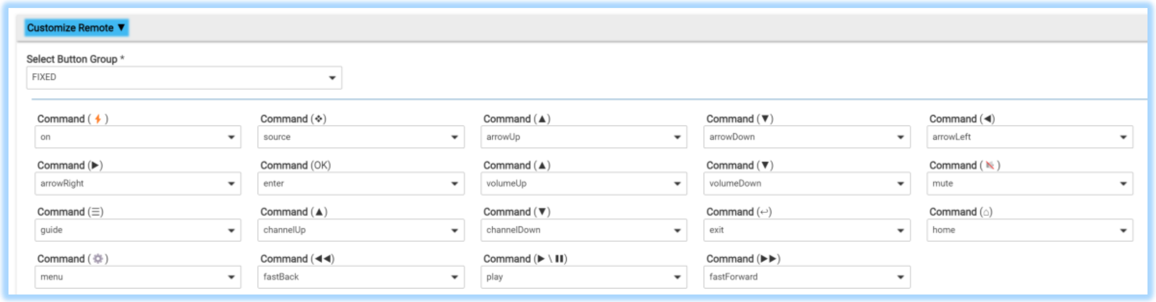
Customize TV Remote

The TV remote has two types of buttons, Fixed and Custom. With Fixed buttons the only configurable option is to change the command being executed by that button press.

TV Remote – Fixed Buttons Only	TV Remote – Fixed and Custom Buttons
	
There are a total of 19 fixed buttons.	There are a total of 10 custom buttons.

TV Remote Fixed Buttons

With Fixed buttons only the command to be executed can be changed.



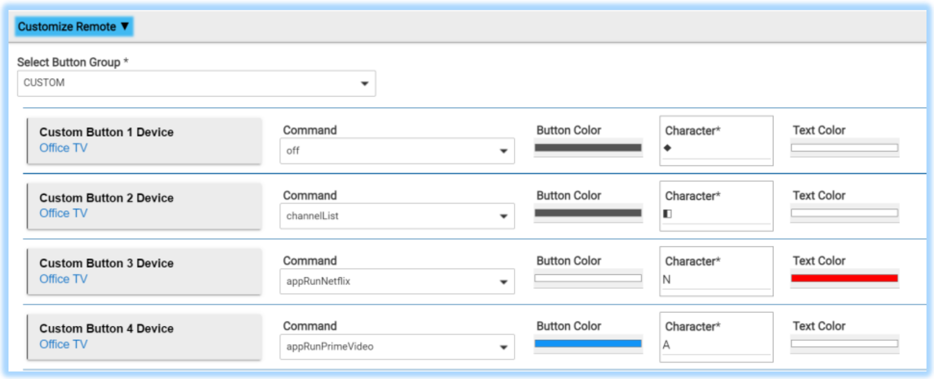
Customize Remote ▼

Select Button Group *
FIXED

Command (⚡) on	Command (✥) source	Command (▲) arrowUp	Command (▼) arrowDown	Command (◀) arrowLeft
Command (▶) arrowRight	Command (OK) enter	Command (▲) volumeUp	Command (▼) volumeDown	Command (🔇) mute
Command (☰) guide	Command (▲) channelUp	Command (▼) channelDown	Command (↩) exit	Command (⌂) home
Command (⚙) menu	Command (◀◀) fastBack	Command (▶ \ ❚❚) play	Command (▶▶) fastForward	

TV Remote Custom Buttons

With Custom buttons multiple properties can be changed.



Customize Remote ▼

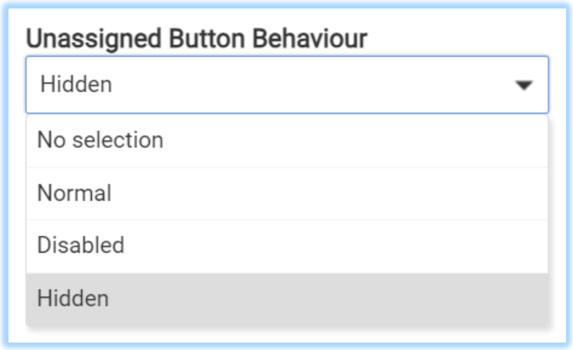
Select Button Group *
CUSTOM

	Command	Button Color	Character*	Text Color
Custom Button 1 Device Office TV	off	■	◆	□
Custom Button 2 Device Office TV	channelList	■	◧	□
Custom Button 3 Device Office TV	appRunNetflix	□	N	🟥
Custom Button 4 Device Office TV	appRunPrimeVideo	🟦	A	□

Note: You can even select to execute different commands on devices other that the TV itself. For example you could have a custom button that toggled the lights or closed the blinds.

Other Commands

The unassigned button behavior has one additional property in the TV Remote which is to allow for unassigned custom buttons to be hidden vs just disabled. This gives the remote a cleaner look.



Publish TV Remote

Publishing the TV remote is the same as other Remote Builder modules. See the description under the Fixed 6 Button Remote for more information.

It’s a Wrap

Well, if you made it this far you are ready to build your own remotes and get the most out of them. I look forward to seeing some of the designs that people come up with and share on the community forums.

I have ideas for four additional **Remote Builder** modules of the same quality as the **TV Remote**. Whether I will see them through depends largely on the **Hubitat** communities’ willingness to acknowledge value in quality software and donate towards the ongoing development of the project.

This is the day the Lord has made, let us rejoice and be glad.