



Pincode Protect

1. Intro

This webview driver provides the ability to block actions with pincode. When the right pincode is entered, the pinpad is removed, and text and buttons are available.

The driver provides some interesting features, like the ability to change from 1 to 3 buttons actions when a pincode is entered, an editable text field when the valid pincode is entered, the use of real-time variable (for example a gate status) in the main text field, the ability to change the UI icon, possibly of saving History events when a valid code is entered and a button is pressed.

Compatibility:

- T3/T4: Require OS 3.1.3+
- iOS and Android: Require OS 3.2+
- **OS 3.3.0+ is recommended**

2. Properties

- *Cloud Status*: Current status of the driver license.
- *Automatic Updates*: Enable driver automatics updates (Recommended!)
- *MAC*: MAC address of the director.
- *Driver*: Driver name.
- *Driver Version*: Current driver version.
- *Driver Information*: Last debug information.
- *Debug Mode*: If On, driver will print debug information on Lua output.
- *UI Icon*: With this property you can choose the icon suits you better.
- *Pincode*: The code that will trigger the action.
- *Valid Pincode Text*: It is the text that appears when a valid code is entered. This text can contain real-time variables to shown (see “Using Real-Time variables” appendix for more info). TIP: You can use “
” (without quotes) to specify a new line (enter).
- *Valid Pincode Button Quantity*: You can choose from 1 to 3 buttons to display when

someone enter the valid code:

- **1:** Button Center will be displayed.
- **2:** Button Left and Right will be displayed.
- **3:** Button Left, Center and Right will be displayed.
- **Button Left/Center/Right Text or Icon:** For each button you can select if you want an icon or a text be display.
- **Button Left/Center/Right Text:** For each button, if you selected “Text”, you can set the text you want to display. **Variables cannot be used here.**
- **Button Left/Center/Right Icon:** For each button, if you selected “Icon”, you can set the icon you want to display.
- **History:** This option enables the function of save on history when a valid pincode is entered, so the previous events can be monitored. When enabled, other properties will be shown allowing the programmer to configure the different cells of the History event.

3. Actions

N.A.

4. Commands

N.A.

5. Events

- ❖ **Correct Pincode Entered:** It will be triggered when the code is entered correctly.
- ❖ **Invalid Pincode Entered:** It will be triggered when an invalid code is entered.
- ❖ **Left Button Pressed:** It will be triggered when the left button is pressed. Be aware that Left button is present if 2 or 3 buttons are selected on “Valid Pincode Button Quantity” property.
- ❖ **Center Button Pressed:** It will be triggered when the center button is pressed. Be aware that Center button is present if 1 or 3 buttons are selected on “Valid Pincode Button Quantity” property.
- ❖ **Right Button Pressed:** It will be triggered when the right button is pressed. Be aware that Right button is present if 2 or 3 buttons are selected on “Valid Pincode Button Quantity” property.

NOTE: Be aware that if you program a button and then that button is eliminated because you

changed the “Valid Pincode Button Quantity”, the programming will be lost despite of added it again. For example, if you had “Valid Pincode Button Quantity” configured with 3 buttons and all of them with programming, if you change from 3 to 1 button, the programming of the center button will be remained, but button left and right will be lost although you change it again to 3 buttons.

5. Driver Variables

- ❖ **LAST_MENU_SELECTED**: Last menu name that selected the driver.
- ❖ **LAST_ROOM_SELECTED**: Last ID of the room that selected the driver.

6. Using Real-Time variables

To use variables on the properties, we need the variables parameters. These are: deviceID and variableID.

To find them, we need to go to “Variables” Agent. If the variable is a custom variable (was created by the programmer), you will see it on the sheet. If the variable was created by the system (or another driver), you must check the box “Display System Variables”.

The screenshot shows the 'Variables' agent window. The top table lists custom variables, and the bottom table lists system variables. The 'Display System Variables' checkbox is checked and highlighted with a red box. A red arrow points from the 'Custom Variables' label to the top table, and another red arrow points from the 'System Variables' label to the bottom table.

Variable Name	Value	Type	Description	Last Updated
cambio_ac2dh	False	Boolean		
CP-DiagnosticsIgnoredDevices	[]	String	Used for Composer Pro Diagnostics	
Doorbell		String		

System Variable Name	Device Name	Value	Type	Description
1001	Color	1Energize23134630.3170970...	XML	
Active	Living->Spotify Connect	false	String	
Active	Main Room->Spotify Connect	false	String	
ANNOUNCEMENT_DISABLED	Main Room	False	Boolean	
ANNOUNCEMENT_DISABLED	Service	False	Boolean	
ANNOUNCEMENT_DISABLED	Rack	False	Boolean	
ANNOUNCEMENT_DISABLED	Living	False	Boolean	
ANNOUNCEMENT_DISABLED	Guest Room	False	Boolean	
AUXILIARY_HEAT_SOURCE	Main Room->Main Room Split	Not Used	String	
AUXILIARY_HEAT_SOURCE	Living->Living Split	Not Used	String	
AUXILIARY_HEAT_SOURCE	Guest Room->Guest Room Split	Not Used	String	
AWAKE_TIME	Living->System Remote Control SR260	255	Number	
AWAKE_TIME	Main Room->System Remote Control S...	255	Number	
BATTERY_LEVEL	Living->System Remote Control SR260	100	Number	
BATTERY_LEVEL	Main Room->System Remote Control S...	0	Number	
Brightness Percent	Living->Comedor	0	Number	Current light level (0% - 100%)
Brightness Percent	Living->Lampara Pie	0	Number	Current light level (0% - 100%)

Then, you must find the variable of interest, and then place the cursor above the variable to get the parameters:

LAST_MENU_SELECTED	Living->Disney +		String	
LAST_ROOM_SELECTED	Living->Door	273	Number	
STATE	Living->Door	Open	String	
LAST_MENU_SELECTED	Living->Door	security	String	
StateVerified	Living->Doorbell	True	Boolean	
ContactState	Living->Doorbell	False	Boolean	
ContactState	Living->Doorbell 2	False	Boolean	

STATE
Device ID: 215
Variable ID: 1001

In this case, deviceID=215 and variableID=1001.

To use it on the properties, we must use this format: “{xxx,yyy}” (without quotes) where “xxx” is the deviceID of the variable, and “yyy” is the variableID.

In this case would be “{215,1001}” (without quotes).

7. Examples

7.1 Example 1

We want to block the opening of a door with the Pincode Driver with the code “123”, so when the code is entered we need a “Open” button to send the action. Additionally, we want to be able to see the status of the door so we can monitor when it is opened or closed. Finally, we want to have a record in History when someone entered a valid code and when the button is pressed.

To accomplish this, we could configure the driver as the next image:

-Driver Configuration-	
Debug Mode	On
UI Icon	Lock
Pincode	123
Valid Pincode Text	Please, tap the "Open" button and the front door will be opened for 5 seconds! Status: {215,1001}!
Valid Pincode Button Quantity	1
Button Center	— BUTTON CENTER —
Button Center Text or Icon	Text
Button Center Text	Open
-History-	
History	On
History Level	Warning
History Category	Pincode Protect
History Sub Category	Front Door

Note that we use “
”, this is a HTML element that means “new line” (enter).

Then we need to make the “Center Button Pressed” event to open the door:

The screenshot displays a software interface for programming a device. It is divided into two main sections: **Programming** and **Script**.

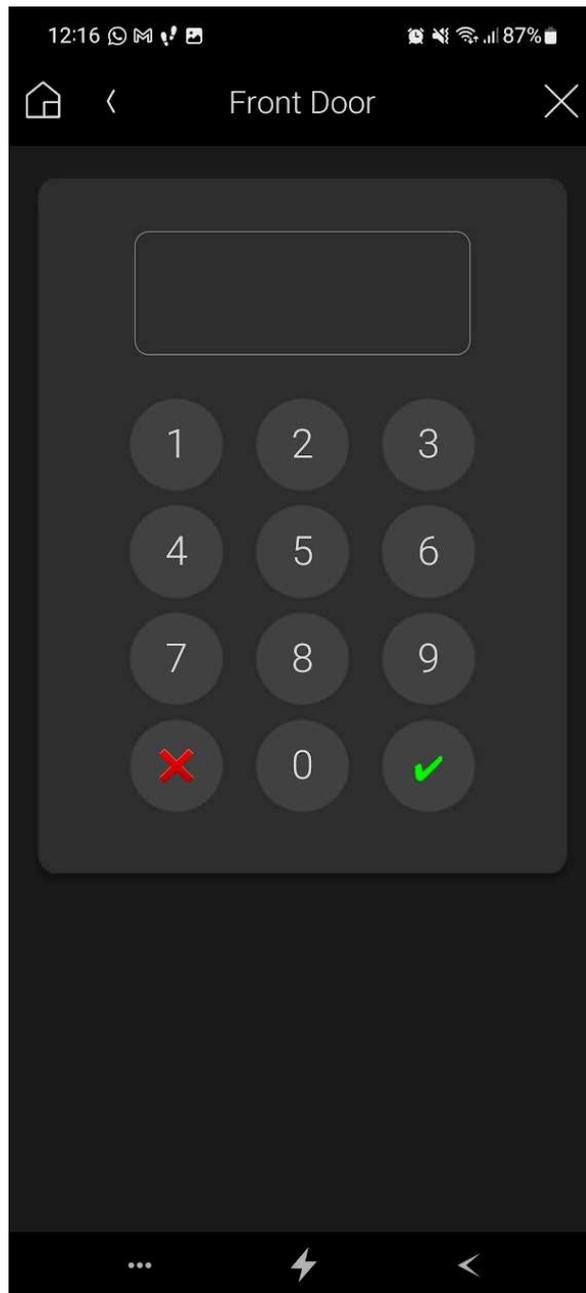
Programming Section:

- Device Events:** A tree view on the left shows a hierarchy starting with **Front Door**. Under it are **Custom**, **Garage**, and **Garage Door**. Below these are various event categories, each with a red circular icon: **Advanced Lighting**, **Announcements**, **Backup**, **Color**, **Communication**, **Custom Buttons**, **Driver Update**, **Email Notification**, **History**, **Identity**, **Macros**, **Media Scenes**, and **Media Sessions**.
- Front Door Events:** A section below the tree view shows a radio button for **Selected** and a dropdown menu with **3 Center Button Pressed** selected.

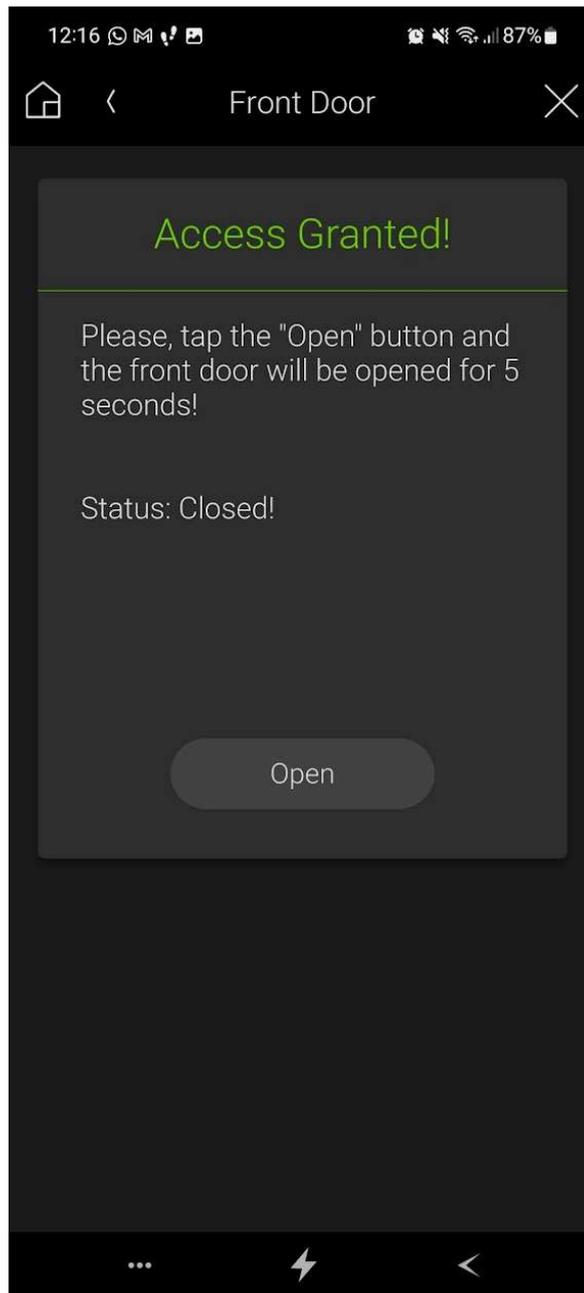
Script Section:

- Script:** The title of the script is **When the center button is pressed**, indicated by a red circular icon.
- Programming Controls:** A row of buttons for logic: **Else**, **& And**, **| Or**, **➡ Break**, **➡ Stop**, and **➡ Delay**. The **Delay** button has a text input field containing **5** and a unit dropdown menu set to **seconds**.
- Script Actions:** A section containing a single action: **➡ Select Hall->Front Door**.

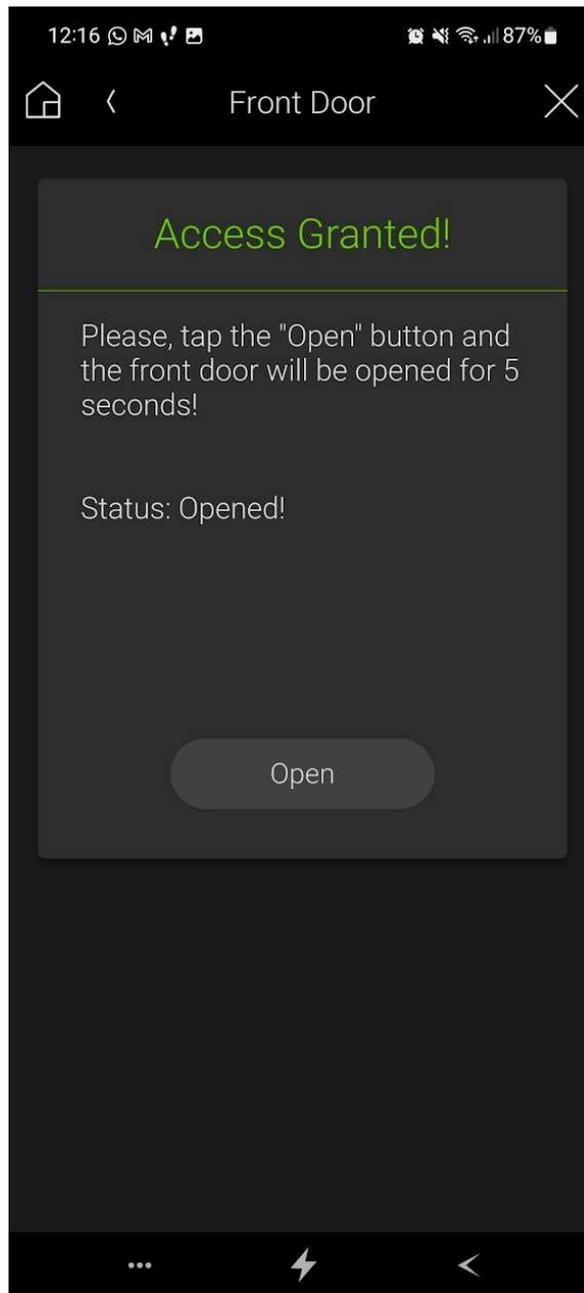
Here you can see on the UI before enter the code:



Right away after the code has been entered:



And finally, after the button is pressed the door will be opened for 5 seconds, so you will see the status "Opened" in that period of time:



And here we can see the history event recorded:

Date/Time		Severity		Query History		5 Items	
From:	<input checked="" type="checkbox"/> 01/09/2022	To:	<input type="checkbox"/> 08/09/2022	<input checked="" type="checkbox"/> Critical	<input checked="" type="checkbox"/> Warning	<input type="checkbox"/> Info	
Name: (All)		Category: Pincode Protect		Subcategory: Front Door		Type: (All)	
				Get Next		20 of 0	
#	Date/Time	Severity	Name	Category	Subcategory	Type	Description
1	08/09/2022 12:16:54	Warning	Storage/Pincode Protect/864	Pincode Protect	Front Door	Button Pressed!	The Center button was pressed!
2	08/09/2022 12:16:22	Warning	Storage/Pincode Protect/864	Pincode Protect	Front Door	Access Granted!	Correct Pincode Entered!

7.2 Example 2

We want to block the opening, closing and stopping of a garage door with the Pincode Driver with the code "321". So we will configured that when someone enters the valid code, there will be 3 buttons with the three actions.

Note: To manage the garage door you will need the specific driver to handle it (for example the C4 "Relay Garage Door Controller") and make the connections and configuration properly. Then you will need to hide the driver if you want to only control it via pincode driver.

For the pincode driver, we could configure the driver as the next image:

-Driver Configuration-	
Debug Mode	On
UI Icon	Garage
Pincode	321
Valid Pincode Text	Now you can control the garage door:
Valid Pincode Button Quantity	3
Button Left	— BUTTON LEFT —
Button Left Text or Icon	Icon
Button Left Icon	Arw Dn
Button Center	— BUTTON CENTER —
Button Center Text or Icon	Text
Button Center Text	Stop
Button Right	— BUTTON RIGHT —
Button Right Text or Icon	Icon
Button Right Icon	Arw Up
-History-	
History	On
History Level	Warning
History Category	Pincode Protect
History Sub Category	Garage Door

Noticed that we select the Garage icon to the UI display, and we selected three buttons: The left one is an icon arrow down, the center one is a text with "Stop", and the right one is a icon with arrow up.

Here you can see the events programming of the different actions:

Programming

Device Events

Front Door
Custom
Garage
Garage Door
Rack

- Advanced Lighting
- Announcements
- Backup
- Color
- Communication
- Custom Buttons
- Driver Update
- Email Notification
- History
- Identity
- Macros
- Media Scenes
- Media Sessions

Garage Events

Selected

3 Left Button Pressed

Script

Script

When the left button is pressed

Programming Controls

Else And Or Break Stop Delay 5 seconds

Comment

Script Actions

Close Storage->Garage Door

Programming

Device Events

Front Door
Custom
Garage
Garage Door
Rack

- Advanced Lighting
- Announcements
- Backup
- Color
- Communication
- Custom Buttons
- Driver Update
- Email Notification
- History
- Identity
- Macros
- Media Scenes
- Media Sessions

Garage Events

Selected

4 Center Button Pressed

Script

Script

When the center button is pressed

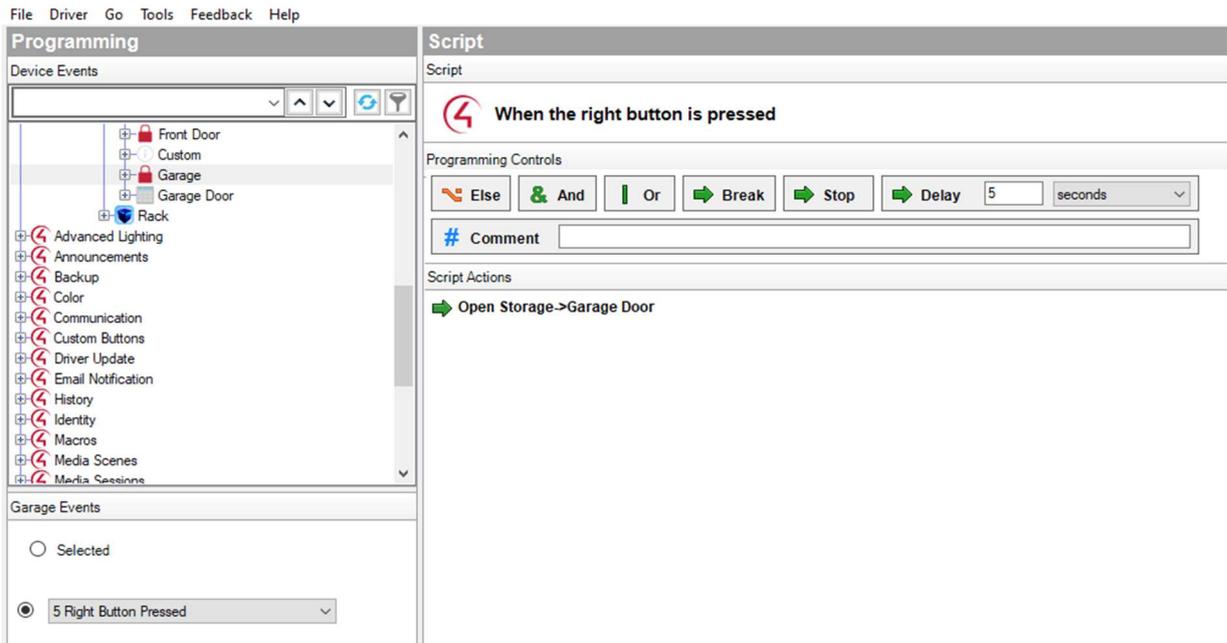
Programming Controls

Else And Or Break Stop Delay 5 seconds

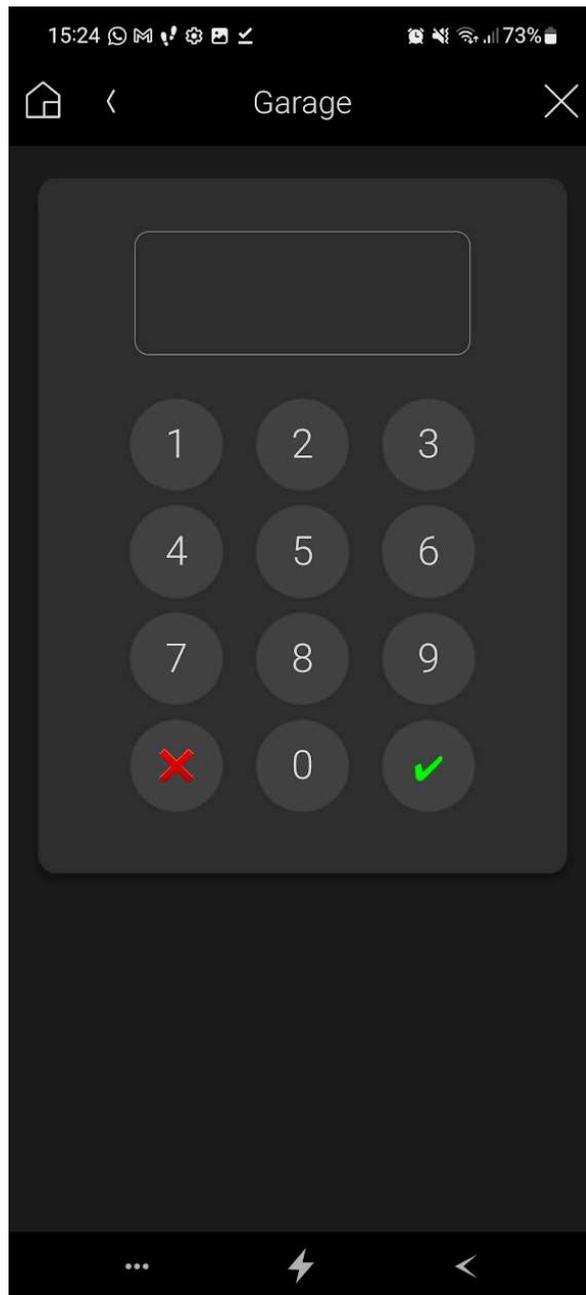
Comment

Script Actions

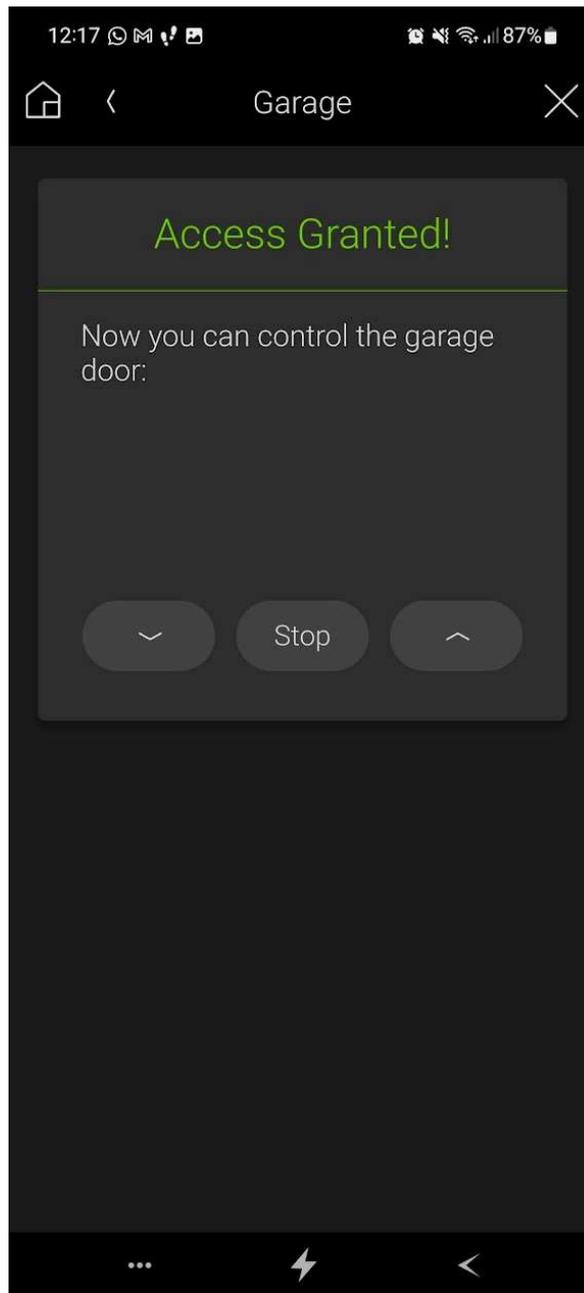
Stop Storage->Garage Door



Here you can see on the UI before enter the code:



Right away after the code has been entered:

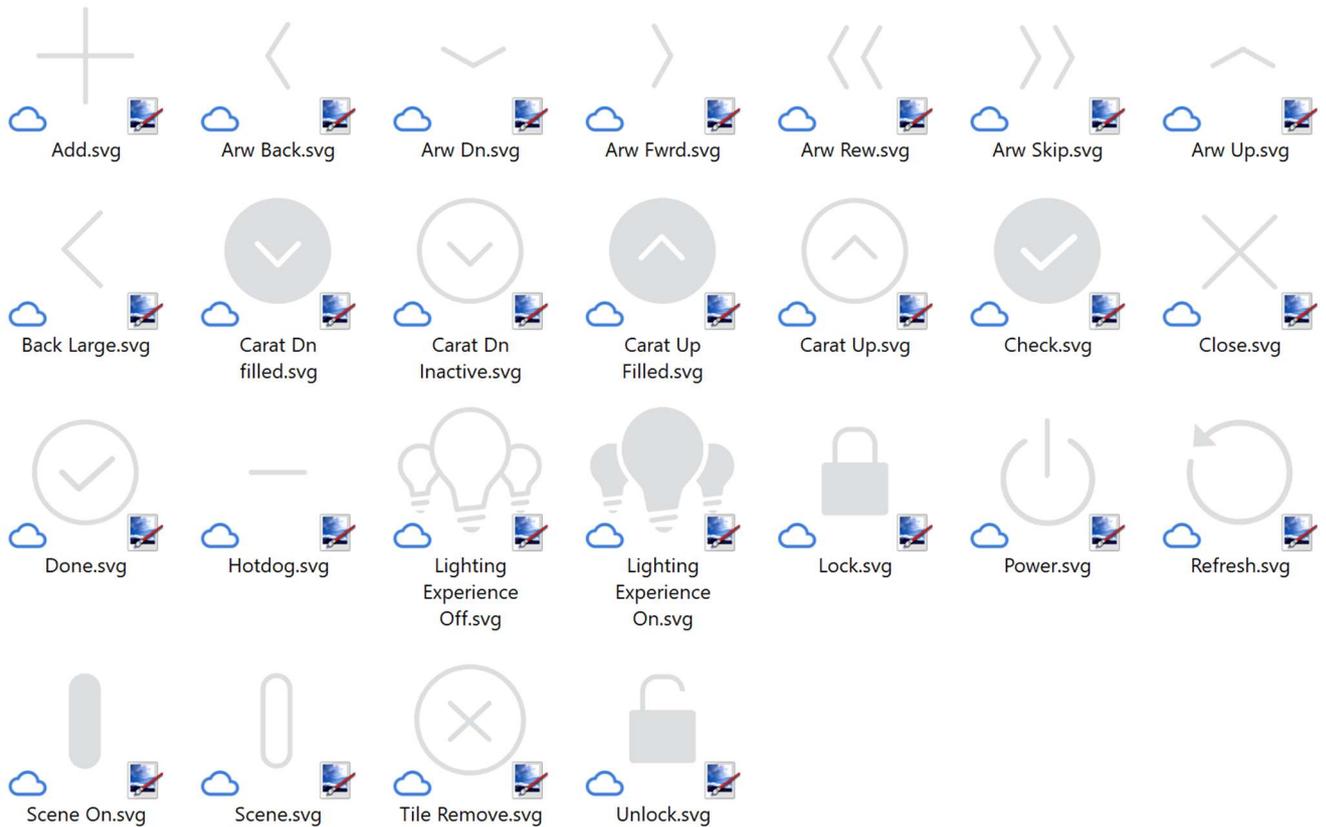


And here we can see the history events recorded:

Date/Time		Severity		Query History		4 Items	
From:	<input checked="" type="checkbox"/> 01/09/2022	To:	<input type="checkbox"/> 08/09/2022	<input checked="" type="checkbox"/> Critical	<input checked="" type="checkbox"/> Warning	<input type="checkbox"/> Info	Get Next: 20 of 0
Name	Category	Subcategory	Type				
(All)	Pincode Protect	Garage Door	(All)				
#	Date/Time	Severity	Name	Category	Subcategory	Type	Description
1	08/09/2022 12:48:17	Warning	Storage/Pincode Protect/890	Pincode Protect	Garage Door	Button Pressed!	The Right button was pressed!
2	08/09/2022 12:48:16	Warning	Storage/Pincode Protect/890	Pincode Protect	Garage Door	Button Pressed!	The Center button was pressed!
3	08/09/2022 12:48:15	Warning	Storage/Pincode Protect/890	Pincode Protect	Garage Door	Button Pressed!	The Left button was pressed!
4	08/09/2022 12:48:12	Warning	Storage/Pincode Protect/890	Pincode Protect	Garage Door	Access Granted!	Correct Pincode Entered!

8. Button icons

Here you can see the list of icons and their names, available to use in each button if the “icon” option was selected:



*More icons could be added in a future without updating this documentation

9. Limitations

These are the current driver limitations:

- Sometimes, when you select the driver in the UI, a black background and a spinning wheel can appear for a couple of seconds before the pinpad shows up. Apparently, in that cases the Control4 app take more time to load the web-page and that is why you must wait a little bit.

10. Best practices

- The main purpose of this driver is to block actions, not to replace other driver's functionality. For example, if you want to block a garage door actions, you must first configure a garage door driver properly (for example control4 Relay Garage Door Controller), hide it from navigation and then program the buttons of the Pincode driver to send the

actions to the garage door driver as you wish.

- This driver relay on webview support of Control4 systems. This means, like any webview driver on the market, an interaction of several programming codes (Lua, JS, CSS, HTML, etc), so any change that Control4 makes could lead on the driver not working properly. Despite of the purpose of this driver is to add one more step of access control, this is not a security driver per se. Be conscious of this and test the driver as much as possible to make sure it fulfils your needs and performs accordly to your interest.
- Try to use always at least one variable, so the customer can see the change of state when a button is pressed.

11. Warranty and disclaimer

Barbini.dev provide drivers tested as much as possible. However, due to the highly variants of systems setups and the constant changing of the electronics systems, updates and/or modifications may be required to fixing bugs or improve the driver functionality.

Many of our drivers interact with systems APIs (like Control4 and/or any other 3rd party system). Therefore, if any API changes, the driver may stop functioning correctly. **Barbini.dev** cannot guarantee long-term functionality of any driver developed that uses APIs. **Barbini.dev** has the right to repair, provide updates, or discontinue a driver at any time. These repairs or updates could be free or required additional expenses. Despite the above, our efforts will always be to maintain the drivers as long as possible and free of charge of upgrades as much as possible.

13. Trial and Showroom Licenses

All the drivers have 7 days of trial, allowing dealers to test the features of any driver in a customer project before purchasing it. The trial automatically begins when a driver is installed in a project.

We enable all the Showroom licenses to have unlimited access to the drivers.

Developed By: Barbini.dev

Contact: barbini.dev@gmail.com



VERSION 22092801: Official release