



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 of add Users and enable/disable them, 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.
- *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.
- *Theme*: You can choose the UI theme.

- **Admin Pincode:** The code of the Admin, this “user” can’t be disable.
- **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.
- **Select User:** In the case you added more users within the Action button, you can select them here and change their configuration.
- **User Code:** Here you can change the user code of the selected user. The code must be unique.
- **User Enable/Disable:** Here you can change the status of the user.
- **Delete User:** In the case you select “Yes” the selected user will be deleted.
- **History:** This option enables the function of save on history when a valid pincode is entered, when an invalid code is entered (if the user is disable it will be saved too), when a valid user pressed the buttons and when a user is enabled/disabled. 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.
- **Device/Proxy ID:** This property was added to improve finding a variable pattern. Insert the number of the device or proxy ID to be able to see all the variables associated to it.
- **Select Variable:** Here you can select one of all the variables of the previous ID. To be more helpful, the instant value of the variable was added to variable line.
- **Variable Pattern:** Once you selected the variable of the desired Device/Proxy ID, this property appears with the complete pattern. This pattern should be inserted on the right

property/properties so the driver replaces this patten with the variable value in real time.

3. Actions

- *1 Add User:* Use this button to add new users. This user can be enabled/disabled in different ways to handle restrictions. Each code and name must be unique.
- *2 Print Users:* It will print the users list on Lua Output.

4. Commands

- *Enable/Disable User:* This command allows you to trigger an enable/disable action to a user or a group of users. You must insert the name of the user respecting the uppercase and lowercase, and separating with commas if more than one user.
- *Enable/Disable ALL:* Use this command to mass update the status of all the users.

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 (wrong code or disabled user).
- *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.
- *Last Unlocked By*: Last user who had access.
- *Last Disabled User Code Entered*: Last disabled user who tried to unlock.

6. Using Real-Time variables

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

With an update, the finding of the variable was improved. Here you have detailed the new method, and the legacy method (still works of course).

NEW METHOD

With a recent update, some properties were added to find a desire variable way more faster than before.

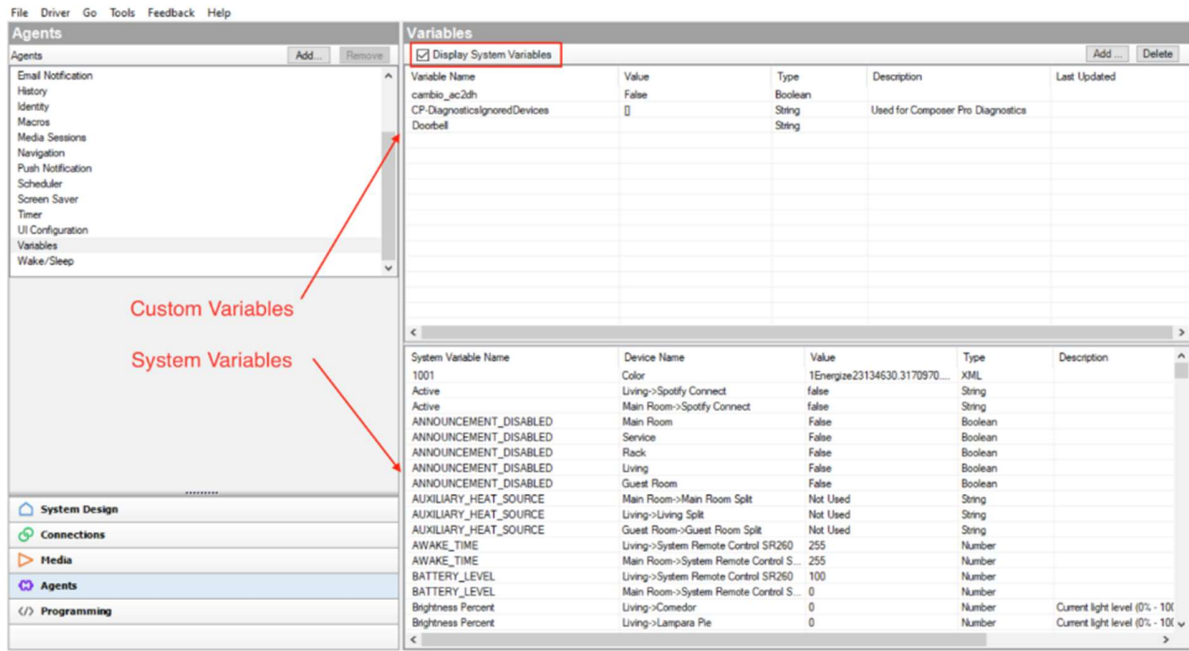
- Variable Helper -

Device/Proxy ID	60
	For Composer Pro Created Variables use 100001
Device Name	Living Split
Select Variable	[1104] HVAC_MODE (value='Cool')
Variable Pattern	{60.1104}
	Insert this pattern on the property to utilize the 'selected' variable. For more information, refer to the documentation.

You only need to insert the Device/Proxy ID. Afterward, a 'Select Variable' property will appear, displaying a list of all the variables associated with that ID. To enhance the search process, an instant value for each variable has been added on each line for reference purposes only. When you select a variable, the corresponding pattern will be presented. This pattern should be inserted into properties that support real-time variables, and the driver will replace it with the correct value.

LEGACY METHOD

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”.



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	STATE Device ID: 215 Variable ID: 1001	Boolean	
ContactState	Living->Doorbell	False	Boolean	
ContactState	Living->Doorbell 2	False	Boolean	

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

To use it on the properties, we must use this format: **{xxx,yyyy}** where “xxx” is the deviceID of the variable, and “yyyy” is the variableID.

In this case would be **{215,1001}**.

7. Examples

7.1 Example 1

We want to block the opening of a door with the Pincode Driver with the Admin 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:

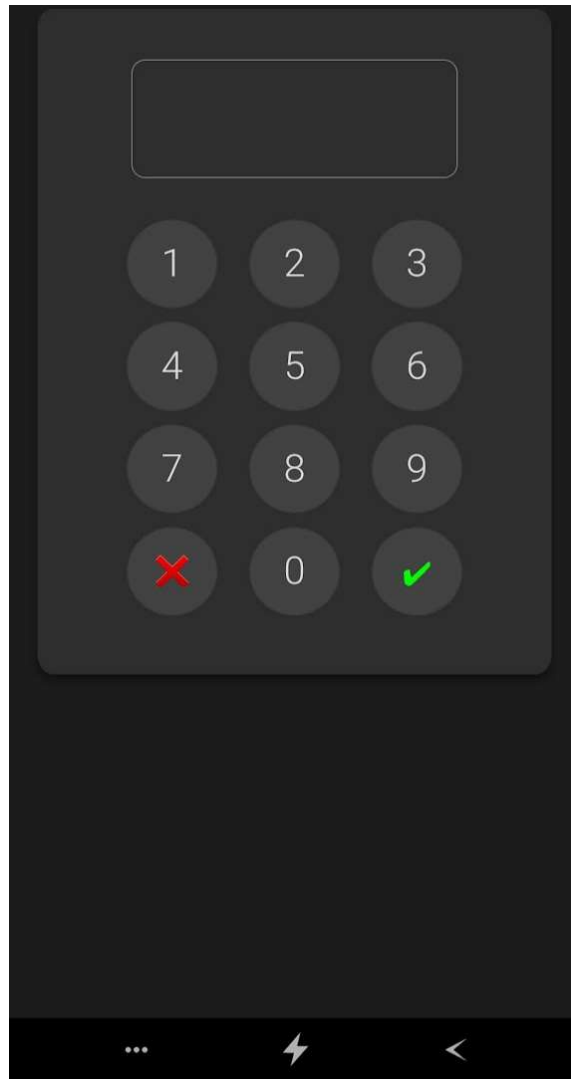
-Driver Configuration-	
Debug Mode	On
UI Icon	House Lock
Admin Pincode	123
Valid Pincode Text	Please, tap the "Open" button and the front door will be opened for 5 seconds! Status: {215,1001}
Valid Pincode Button Quantity	1
Button Center	— BUTTON CENTER —
Button Center Text or Icon	Text
Button Center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

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

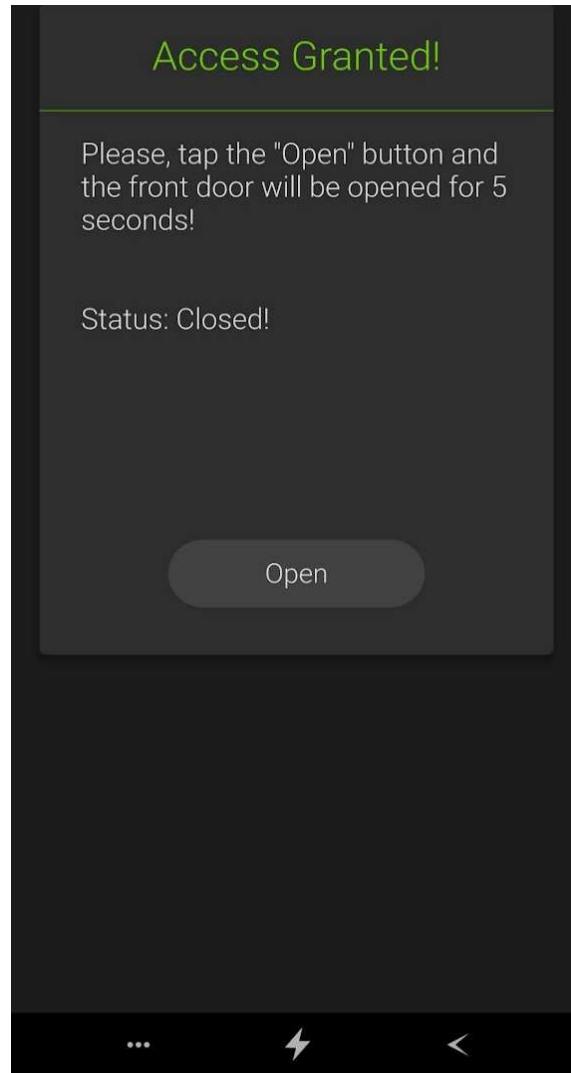
Then we need to make the "Center Button Pressed" event to open the door:

The screenshot shows a software interface with a menu bar (File, Driver, Go, Tools, Feedback, Help) and two main panels. The left panel, titled 'Programming', shows a tree view of 'Device Events' with 'Front Door' expanded to show '3 Center Button Pressed' selected. The right panel, titled 'Script', shows a script titled 'When the center button is pressed'. Below the title are 'Programming Controls' including buttons for 'Else', 'And', 'Or', 'Break', 'Stop', and 'Delay' (set to 5 seconds). A 'Comment' field is also present. Under 'Script Actions', there is a single action: 'Select Hall->Front Door'.

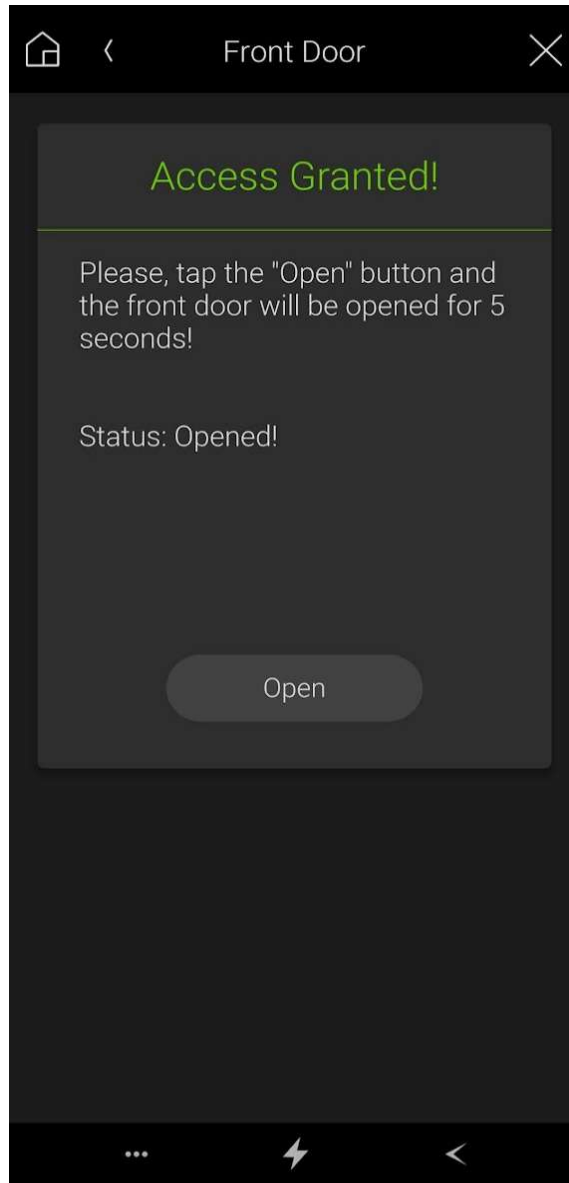
Here you can see on the UI before enter the code:



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	Name	Category	Subcategory	Type	Description
1	12/02/2023 12:17:41	Warning	Service/Pincode Protect/686	Pincode		Button Pressed! [Admin]	The Center button 'Open' was pressed by user 'Admin'!
2	12/02/2023 12:17:39	Warning	Service/Pincode Protect/686	Pincode		Access Granted! [Admin]	Correct Pincode Entered by user 'Admin'!

7.2 Example 2

We want to block the opening, closing and stopping of a garage door with the Pincode Driver with the Admin code “321”. So, we will configured that when someone enters the valid code, there will be 3 buttons with the three actions. Also, we want to have a Housekeeper user with the code “456” that only will have access between 9am to 6pm all days.

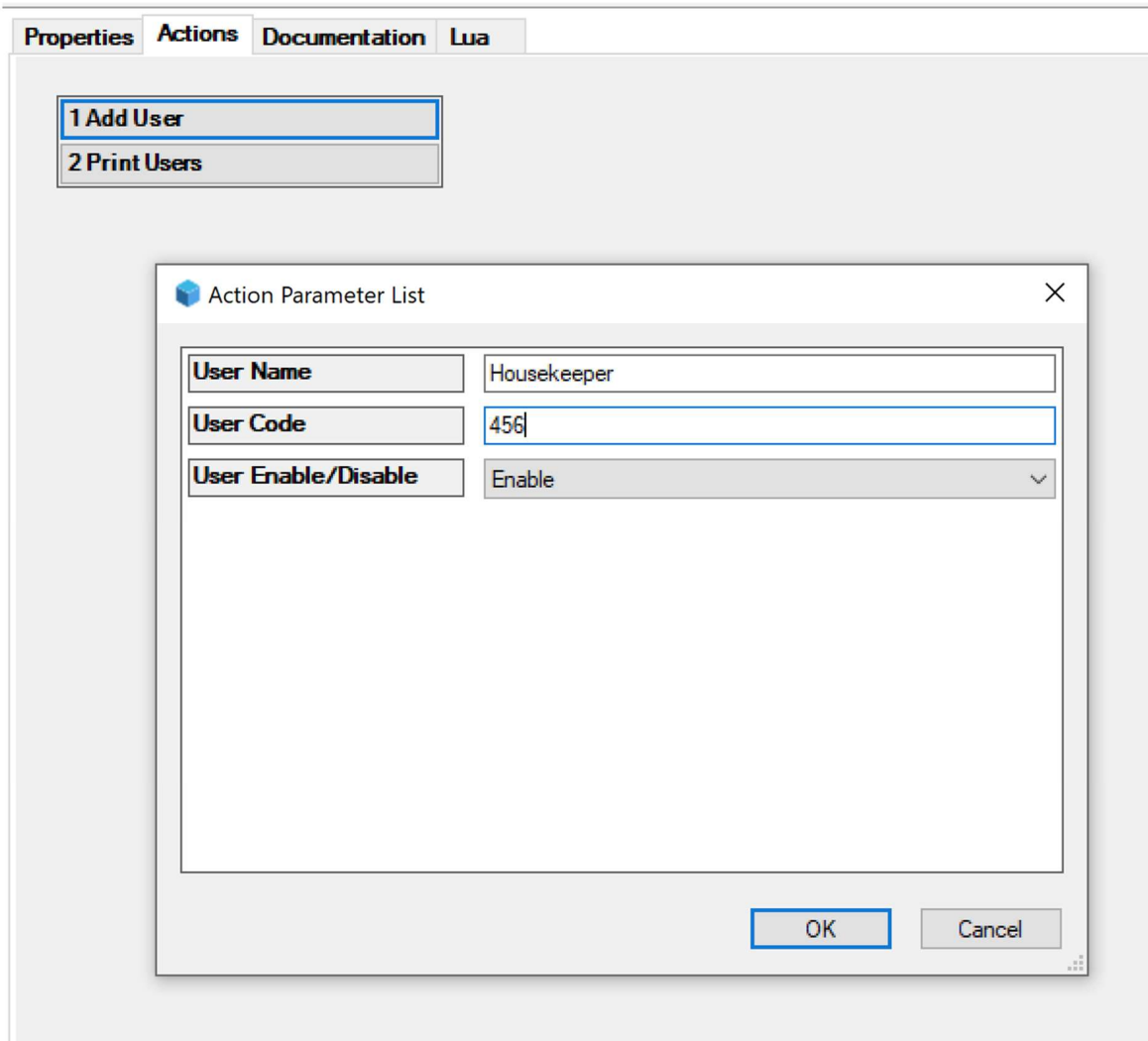
Note: To manage the garage door you will need the specific driver to handle it (for example the C4 “Relay Garage Door Controller”) and make the connections and configuration properly. Then you will need to hide the driver if you want to only control it via pincode driver.

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

-Driver Configuration-	
Debug Mode	On
UI Icon	Garage
Admin 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
-Users Administration-	
Select User	
-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 an icon with arrow up.

Now we add the Housekeeper user:



Here you can see the events programming for the different buttons 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

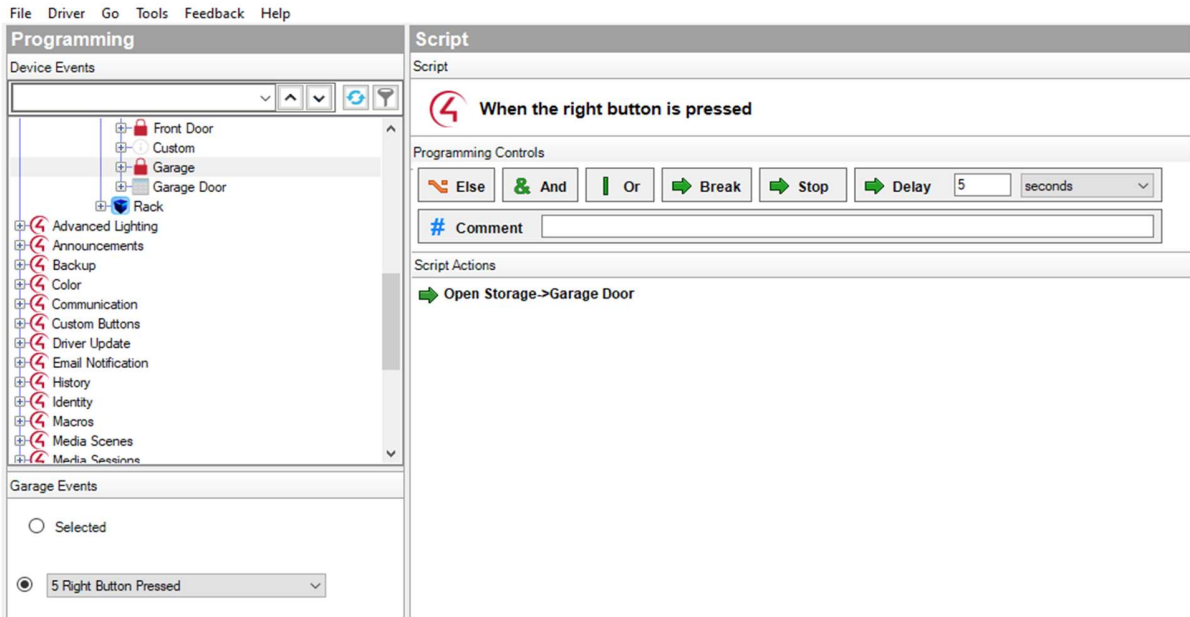
Programming Controls

Else And Or Break Stop Delay 5 seconds

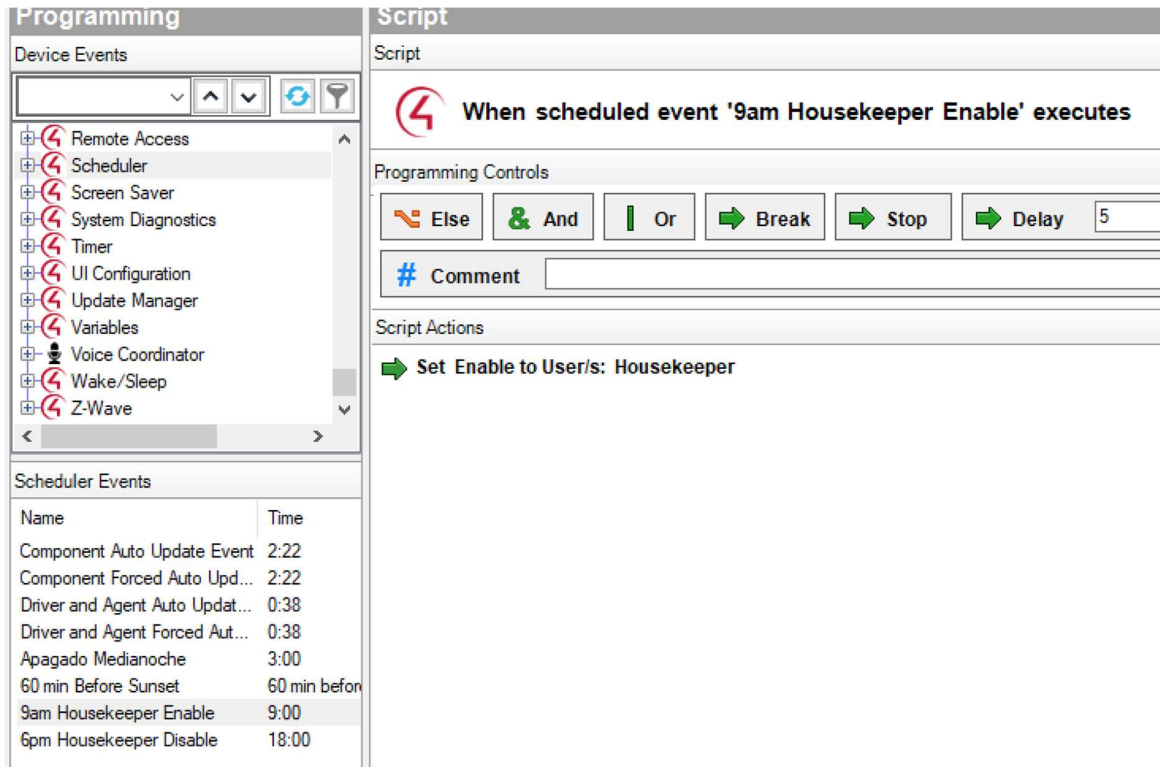
Comment

Script Actions

Stop Storage->Garage Door



And now we create two Schedules, one at 9am and the other at 6pm. Then we program the 9am schedule to enable the Housekeeper user, and the 6pm to disable it.



Programming

Device Events

- Remote Access
- Scheduler
- Screen Saver
- System Diagnostics
- Timer
- UI Configuration
- Update Manager
- Variables
- Voice Coordinator
- Wake/Sleep
- Z-Wave

Scheduler Events

Name	Time
Component Auto Update Event	2:22
Component Forced Auto Upd...	2:22
Driver and Agent Auto Updat...	0:38
Driver and Agent Forced Aut...	0:38
Apagado Medianoche	3:00
60 min Before Sunset	60 min before
9am Housekeeper Enable	9:00
6pm Housekeeper Disable	18:00

Script

Script

When scheduled event '6pm Housekeeper Disable' executes

Programming Controls

Else
 & And
 | Or
 ➔ Break
 ➔ Stop
 ➔ Delay

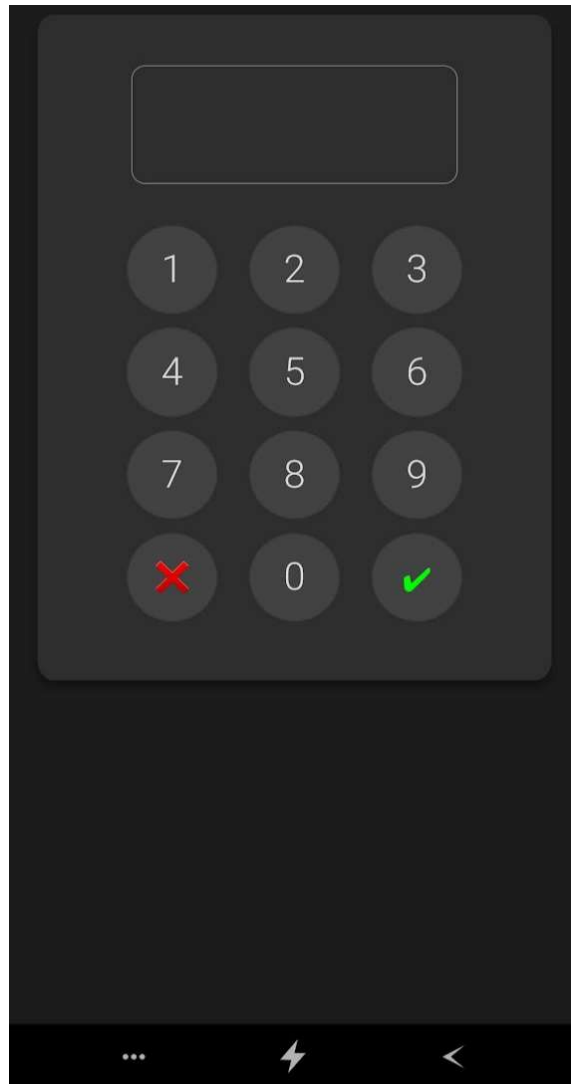
Comment

Script Actions

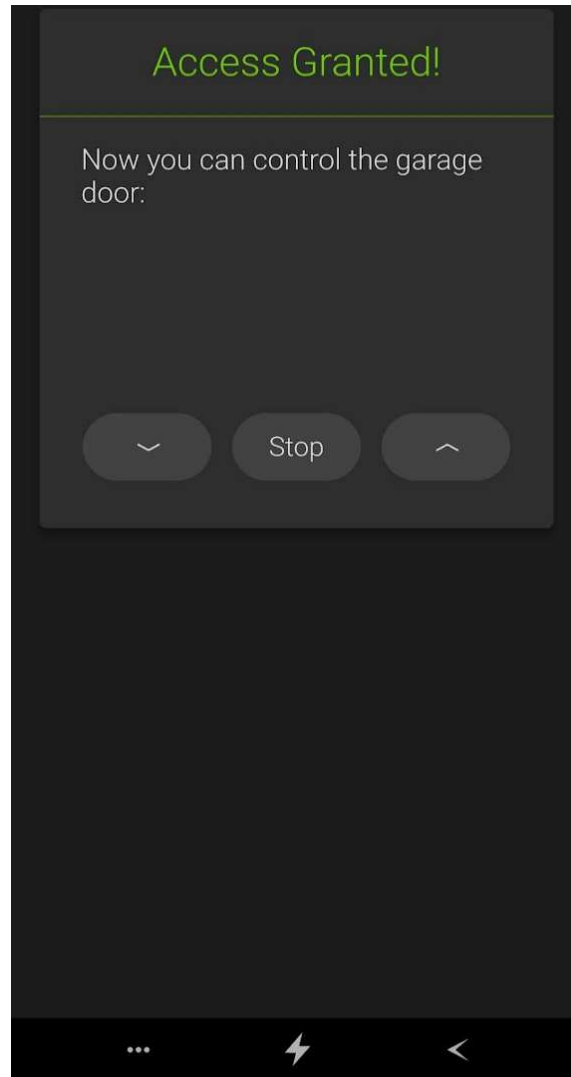
Set Disable to User/s: Housekeeper

You should test it enable/disable correctly.

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



Right away after the code has been entered:



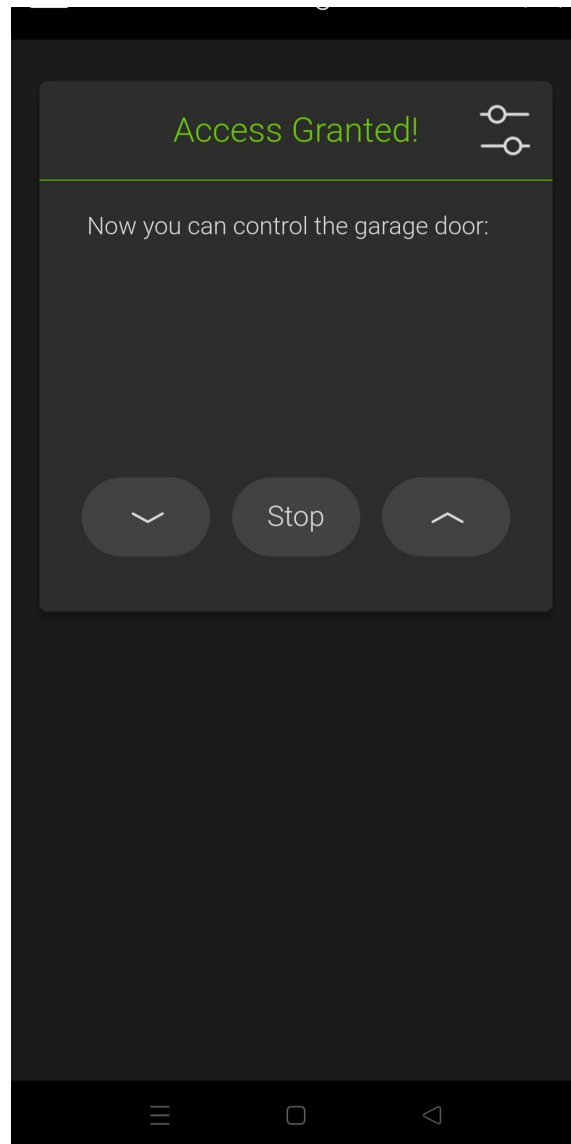
And here we can see the history events recorded:

#	Date/Time	Severity	Name	Category	Subcategory	Type	Description
1	12/02/2023 12:43:09	Warning	Service/Pincode Protect/688	Pincode Protect	Garage Door	Access Denied! [Housekeeper]	Disable User Pincode Entered by user 'Housekeeper' !
2	12/02/2023 12:43:02	Warning	Service/Pincode Protect/688	Pincode Protect	Garage Door	Button Pressed! [Housekeeper]	The Left button 'Arw Dn' was pressed by user 'Housekeeper' !
3	12/02/2023 12:43:00	Warning	Service/Pincode Protect/688	Pincode Protect	Garage Door	Button Pressed! [Housekeeper]	The Center button 'Stop' was pressed by user 'Housekeeper' !
4	12/02/2023 12:42:59	Warning	Service/Pincode Protect/688	Pincode Protect	Garage Door	Button Pressed! [Housekeeper]	The Right button 'Arw Up' was pressed by user 'Housekeeper' !
5	12/02/2023 12:42:56	Warning	Service/Pincode Protect/688	Pincode Protect	Garage Door	Access Granted! [Housekeeper]	Correct Pincode Entered by user 'Housekeeper' !
6	12/02/2023 12:42:46	Warning	Service/Pincode Protect/688	Pincode Protect	Garage Door	Button Pressed! [Admin]	The Right button 'Arw Up' was pressed by user 'Admin' !
7	12/02/2023 12:42:44	Warning	Service/Pincode Protect/688	Pincode Protect	Garage Door	Button Pressed! [Admin]	The Center button 'Stop' was pressed by user 'Admin' !
8	12/02/2023 12:42:42	Warning	Service/Pincode Protect/688	Pincode Protect	Garage Door	Button Pressed! [Admin]	The Left button 'Arw Dn' was pressed by user 'Admin' !
9	12/02/2023 12:42:39	Warning	Service/Pincode Protect/688	Pincode Protect	Garage Door	Access Granted! [Admin]	Correct Pincode Entered by user 'Admin' !

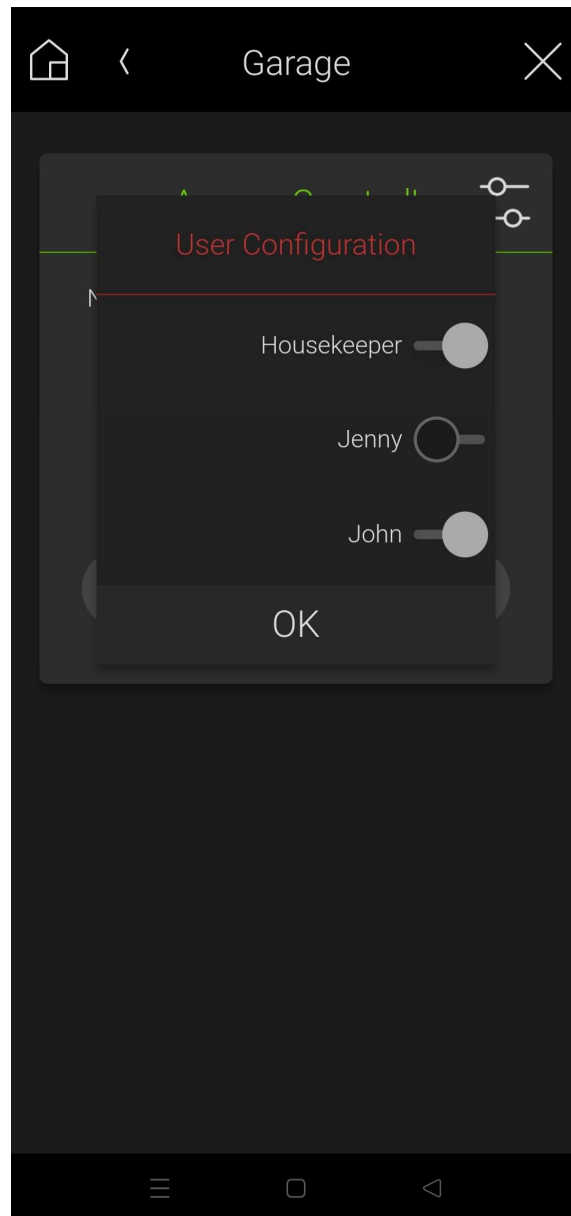
(Note the Housekeeper user was disabled to show how it is recorded in History.)

UPDATE APRIL 2023:

Now the users can be enabled/disabled on the UI if the admin pincode was entered. When this happens, a configuration button appears:



If you press that button, the user's menu is shown:

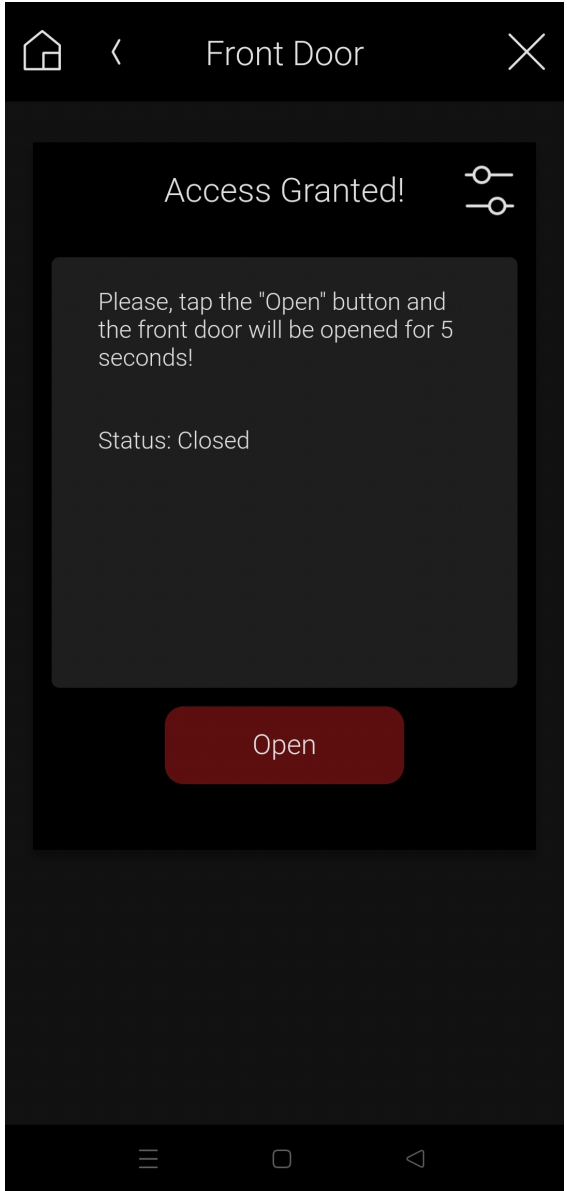


You can toggle the selector to enable or disable each user. The user status will be remaining that way until the admin changes it on the UI or a driver action that was previously configured is executed (for example a schedule).

When a user status is changed and the History record is enabled, the action will be recorded.

Also a Modern theme was added. Here you can see some examples:

1	2	3
4	5	6
7	8	9
X	0	✓



Front Door



Access Granted!

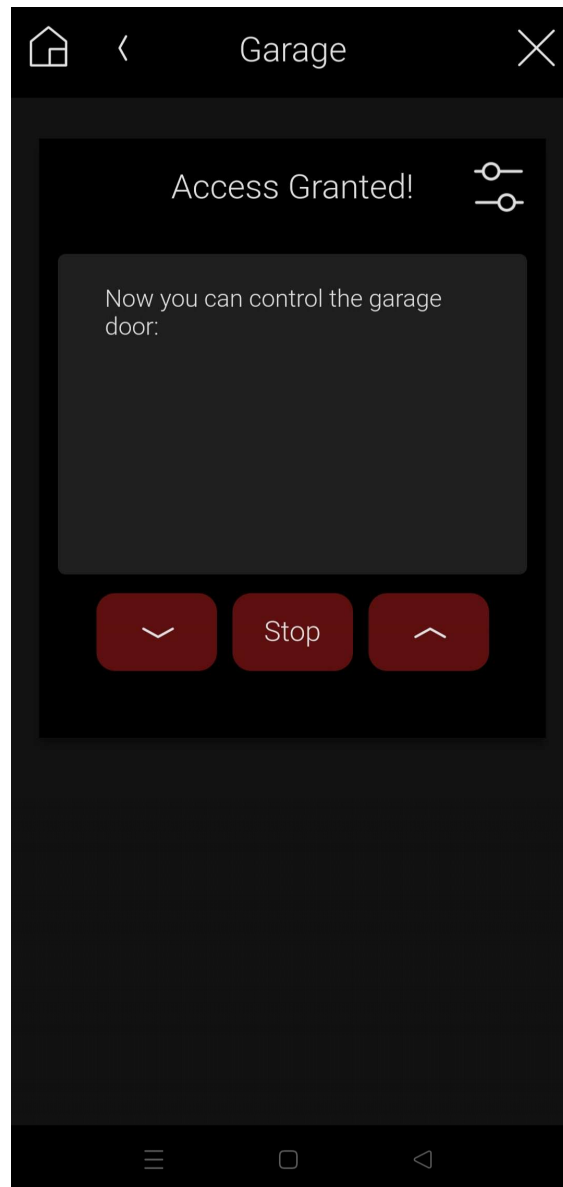


Please, tap the "Open" button and the front door will be opened for 5 seconds!

Status: Closed

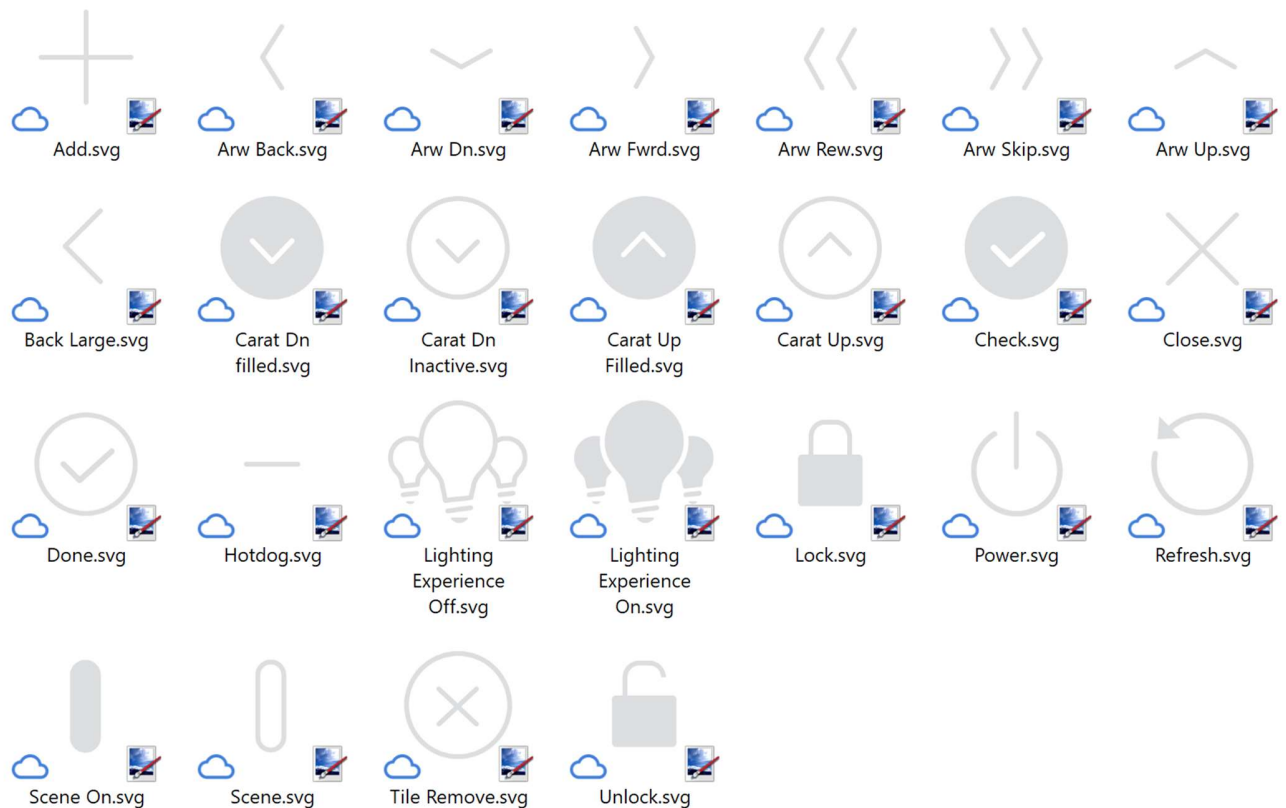
Open





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 repairments 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 10 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@highqualityautomation.com



22092701: Initial release.

22092702: Correct and Invalid pincode events bug fixed.

22093001: Added UI icons: Projector, TV, Blinds, Projector Screen, Media Sources (Apple TV, Console, blu-ray, etc).

22102003: Fixed iOS button icons not loading.

23021503: Added User features (Add, disable, enable and remove users) and added new variables.

230410xx: Added User administration menu (enable/disable) from UI when the Admin code is entered.

230429xx: Added Modern theme.

231204xx: Added an improvement for boolean variables. Now they are displayed as "True" or "False" rather than "1" or "0". A new efficient method for quickly locating variables in the driver's properties was introduced, eliminating the need to leave the driver page.

240514xx: Fixed bug in iOS devices not working properly with the driver if there was an apostrophe symbol inside a string.

240516xx: Fixed a bug where some buttons were not executing the event correctly if they were pressed on the borders.