mguyard / **appdaemon-coversmanager** Public

Code | Issues | Pull requests | Discussions | Actions | Projects | Security | Insights

main | **3** Branches | **25** Tags | Go to file | Go to file | Code

semantic-release-bot chore(release): 1.4.0 [skip ci] c6e8176 · 2 weeks ago 124 Commits

.github	ci: Update for repo rename	3 months ago
apps/CoversManager	feat: Manage locker (global and close)…	2 weeks ago
.gitignore	ci: Update .gitignore	3 months ago
.releaserc	ci: Adding Github workflows	3 months ago
CHANGELOG.md	chore(release): 1.4.0 [skip ci]	2 weeks ago
CoversManager-Logo.png	doc: Update logo and doc	3 months ago
LICENSE	Initial commit	4 months ago
README.md	feat: Manage locker (global and close)…	2 weeks ago
hacs.json	ci: Update HACS name	3 months ago
pyproject.toml	ci: Adding Github workflows	3 months ago

About

AppDaemon App to manage your covers with Home Assistant

#python #home-automation #automation #home-assistant #appdaemon-apps

Readme
GPL-3.0 license
Activity
1 star
1 watching
1 fork
Report repository

Releases 25

v1.4.0 Latest
2 weeks ago

+ 24 releases

No packages published

Contributors 2

mguyard Marc GUYARD
semantic-release-bot Semantic Rele…

Languages

● **Python** 100.0%

README | GPL-3.0 license

AppDaemon - Covers Manager

`All you need to manage your covers !`

license GPL-3.0 | last commit august | python 100.0% | languages 1

release v1.4.0 | release date august | Python Lint passing | HACS Validate passing

🔗 Quick Links

> - 📍 Overview
> - 📦 Features
> - ❔ How this works
> - 🚀 Getting Started
>   - 🔝 Requirements
>   - ⚙️ Installation
> - 🏷️ Configuration
>   - 🤖 AppDaemon
>   - 🪟 Covers Manager
>   - 🧩 Parameters
>   - 📄 Full Configuration Example
> - 🪲 Debug
> - 🤝 Contributing

📍 Overview

Objective

The objective of the Covers Manager project is to provide a comprehensive solution for managing covers. The project aims to simplify and streamline the process of managing covers by automating various tasks and providing an efficient tool for users. The Covers Manager project is designed to address the specific needs of managing covers, such as those used in home automation systems or other applications. It offers a range of features and functionalities to facilitate the management of covers, including opening, closing or manage covers position depending of sun position and temperature (indoor and outdoor). Overall, the objective of the Covers Manager project is to simplify and optimize the management of covers, providing users with a powerful and efficient tool for controlling their covers.

> ⓘ Note
>
> It's important to note that the Covers Manager project is under development. Project is open-source, and users are encouraged to adapt it to their own needs if necessary and purpose all evolutions by submitting a PR.

Motivation

This AppDaemon application was born out of the need to manage roller shutters as I did on Jeedom before migrating to Home Assistant. The aim was simple: recover the automatic opening and closing functions, as well as the ability to close the shutters proportionally according to the position of the sun.

Inspiration

I was greatly inspired by the Volets plugin on Jeedom by mika-nt28 as well as the work of BasBrus and Langestefan (https://community.home-assistant.io/t/custom-component-adaptive-cover/712626).

📦 Features

CoversManager is developped to help you with these features :

- Opening covers (based on time, lux, sunrise hour)
- Closing covers (based on time, lux, sunset hour)
- Adaptive covers management based on sun position, indoor and outdoor temperature (optional)
- Block adaptive changes when manual position change is detected

❔ How this works

CoversManager works in 3 modes :

- Opening (who manage covers opening - most of the time the morning)
- Closing (who manage covers closing - most of the time the evening)
- Adaptive (open or close covers fully or partially depending of sun position and indoor/outdoor temperature)

Opening

In your configuration, you can define one of the multiple type of opening supported.

- Off (you don't want CoversManager manage your opening)
- Time (you define at which time CoversManager will open your covers)
- Sunrise (your covers will open at sunrise time - calculated internaly by AppDaemon)
- Lux (you define a minimal lux to open covers)
- Prefer-Lux (it's a combinaison of lux and custom time - useful in case of issue with your lux sensor, custom time will be your backup)

> ⚠ Warning
>
> When using prefer-lux, you need to configure a time that will be later than the possible time where required lux will be triggered. Otherwise, opening will be used based on time as it will be the first triggered.

Closing

In your configuration, you can define one of the multiple type of opening supported.

- Off (you don't want CoversManager manage your opening)
- Time (you define at which time CoversManager will open your covers)
- Sunset (your covers will open at sunset time - calculated internaly by AppDaemon)
- Lux (you define a minimal lux to open covers)
- Prefer-Lux (it's a combinaison of lux and custom time or dusk - useful in case of issue with your lux sensor, custom time or dusk will be your backup)

> ⚠ Warning
>
> When using prefer-lux, you need to configure a time that will be later than the possible time where required lux will be triggered. Otherwise, closing will be used based on time as it will be the first triggered.

Adaptive

When adaptive mode is enable, each time the sun position change (based on sun.sun/azimuth), if sun is in window, CoversManager define the better cover position to :

- let in the sun (if indoor temperature is less than the setpoint defined or outdoor temperature is less than indoor temperature) - Open 100%
- keep the sun out but not the light (if indoor temperature is greater than the setpoint defined) - Partially open calculated with sun position and window parameters
- keep the sun out (if outdoor temperature is greater than outdoor high temperature defined) - Close 100%

> ⓘ Note
>
> To prevent covers from constantly moving as the sun does, there are two parameters (min_ratio_change and min_time_change) that define the minimum percent of position change to be executed, and the minimum time between two movements. Please look in parameters to know defaults values.
>
> If manual configuration is enabled, each time you move manually a cover (not by CoversManager), Adaptive mode is disable for the time configured.

🚀 Getting Started

🔝 Requirements

- A valid and functional deployment of `AppDaemon Addon` connected to Home Assistant
- A valid and functional `HACS` (Home Assistant Community Store) integration

⚙️ Installation

- Enable AppDaemon Apps in HACS
- Add Automation repository in HACS as AppDaemon repo : https://github.com/mguyard/appdaemon-coversmanager
- Install Covers Manager in HACS
- Install `Studio Code Server` addon to edit your AppDaemon & CoversManager configuration (optional if your prefer another method to modify your configuration)

🏷️ Configuration

🤖 AppDaemon

Firstly you need to configure your newly AppDaemon installation.

> ⓘ Note
>
> Please continue to next chapter if AppDaemon was already configured before this App

AppDaemon Main configuration is available in file `appdaemon.yaml` most of the time stored in `/add_config/<guid>_appdaemon/`

Please find below an example of basic configuration (It may need to be adapted to suit your configuration) :

```
---
secrets: /homeassistant/secrets.yaml
appdaemon:
  app_dir: /homeassistant/appdaemon/apps
  latitude: 48.80506979319244
  longitude: 2.12031248278925
  elevation: 130
  time_zone: Europe/Paris
  plugins:
    HASS:
      type: hass
http:
  url: http://127.0.0.1:5050
admin:
api:
hadashboard:

logs:
  main_log:
    filename: /config/logs/appdaemon.log
  error_log:
    filename: /config/logs/error.log
```

> ⚠ Warning
>
> Starting with the AppDaemon v0.15.0 addon, configuration of AppDaemon was moved to /addon_configs folder. But HACS still continue to download AppDaemon apps to /config (old folder). To resolve this, we add `app_dir` directive in `appdaemon:` section to use HACS supported folder. If you already have existing apps not coming from HACS, **I recommend to upload manually iopool Pump Manager in your actual app_dir or copying all your existing app before modifying app_dir directive**.

Create folder logs (if not already exist) in `/add_config/<guid>_appdaemon/` and add in `appdaemon.yaml` logs section, the log for Covers Manager :

```
logs:
    [...]
    CoversManager:
        name: CoversManager
        filename: /config/logs/CoversManager.log
```

Following the configuration change, you need to restart your AppDaemon addon.

AppDaemon dependancies

You need to add a python package to AppDaemon for this application to work. To do this, go to your AppDaemon add-on configuration and add `pydantic` in the `package python` field. `pydantic` should appear as a tag above the `package python` field.

SHOW ADD-ON ON MY

🪟 Covers Manager

To configure Covers Manager app you need to edit `apps.yaml` configuration most of the time stored in `/add_config/<guid>_appdaemon/apps/`

Please find below an simple example of configuration to add in file :

```
CoversManager:
    module: covers_manager
    class: CoversManager
    use_dictionary_unpacking: true
    log: CoversManager
    config:
        common:
            opening:
                type: "lux"
            closing:
                type: "lux"
                adaptive: True
            temperature:
                indoor:
                    sensor: "sensor.indoor_temperature"
                    setpoint: 23
                outdoor:
                    sensor: "sensor.outdoor_temperature"
                    high_temperature: 28
            lux:
                sensor: "sensor.outdoor_sensor_illuminance_lux"
                open_lux: 23
                close_lux: 5
        covers:
            cover.roller_shutter_1:
                window_heigh : 210
                window_azimuth: 180
            cover.roller_shutter_2:
                window_heigh : 210
                window_azimuth: 320
```

You have more configuration available. All is detailled in next chapter 🧩 Parameters

🧩 Parameters

Please find below all configuration parameters who don't apply to covers directly

Parent	Parameters	Description	Configuration Path	Def
-	dryrun	Enable a dryrun mode that don't execute open or close functions	config.dryrun	Fals
-	locker	A binary sensor who block opening for open and close when state is On (including all moves by adaptive mode)	config.locker	Nor
position	opened	Define the max position allowed (%) when cover is open	config.common.position.opened	100
position	closed	Define the min position allowed (%) when cover is closed	config.common.position.opened	0
position	min_ratio_change	Define minimum percent of move to allow action	config.common.position.min_ratio_change	5
position	min_time_change	Define minimum time in minutes allowed between move	config.common.position.min_time_change	10
opening	type	Define method to open covers the morning (Allowed value : off	time	sun
opening	time	Time to open covers - Only work with time or prefer-lux type	config.common.opening.time	Nor
opening	locker	A binary sensor who block opening when state is On	config.common.opening.locker	Nor

Parent	Parameters	Description	Configuration Path	Def
		(including opening by adaptive mode)		
closing	type	Define method to open covers the morning (Allowed value : off	time	sun
closing	time	Time to open covers - Only work with time or prefer-lux type	config.common.closing.time	Nor
closing	locker	A binary sensor who block closing when state is On (including all moves by adaptive mode)	config.common.closing.locker	Nor
closing	secure_dusk	Close at dusk in 2 layer if first closing method failed - Only work with time or prefer-lux type	config.common.closing.secure_dusk	Fals
closing	adaptive	Enable adaptive mode who close/open covers based on Sun position and indoor/outdoor temperature	config.common.closing.adaptive	Fals
manual	allow	Enable or Disable detection of manual position change of covers	config.common.manual.allow	Fals
manual	timer	Time to block movements when manual position change is detected. Required if config.common.manual.allow is True	config.common.manual.timer	Nor
temperature.indoor	sensor	Sensor who provide indoor temperature (Positive Integer - No Float)	config.common.temperature.indoor.sensor	Nor
temperature.indoor	setpoint	Indoor temperature setpoint. Below => We need to heat with sun / Above => We need to block sun	config.common.temperature.indoor.setpoint	Nor
temperature.outdoor	sensor	Sensor who provide outdoor temperature (Positive Integer - No Float)	config.common.temperature.outdoor.sensor	Nor
temperature.outdoor	low_temperature	Outdoor temperature to trigger to enable adaptive mode in addition to indoor_temperature	config.common.temperature.outdoor.low_temperature	Nor
temperature.outdoor	high_temperature	Outdoor temperature to trigger when we need to totally close cover to protect from heat. Required when Outdoor sensor is configured	config.common.temperature.outdoor.high_temperature	Nor
lux	sensor	Sensor who provide outside Lux	config.common.lux.sensor	Nor
lux	open_lux	Trigger in lux to open covers. Required if type of opening is lux or prefer-lux	config.common.lux.open_lux	Nor
lux	close_lux	Trigger in lux to close covers. Required if type of closing is lux or prefer-lux	config.common.lux.close_lux	Nor

Parameters for covers are :

Parent	Parameters	Description	Configuration Path	Default	Type	Status
config.covers.	window_heigh	Window Heigh in centimeters	config.covers..window_heigh		PositiveInt	Required
config.covers.	window_azimuth	Window Azimuth in the middle of window	config.covers..window_azimuth		Int between 0-360	Required
config.covers..positional	action	True if cover is positional	config.covers..positional.action	True	Boolean	Optional

Parent	Parameters	Description	Configuration Path	Default	Type	Status
config.covers..positional	status	True if cover provide is position	config.covers..positional.status	True	Boolean	Optional
config.covers..fov	left	What is the left FOV angle between window_azimuth and the sun azimuth entering in window	config.covers..fov.left	90	Int between 0-180	Optional
config.covers..fov	right	What is the right FOV angle between window_azimuth and the sun azimuth leaving in window	config.covers..fov.right	90	Int between 0-180	Optional

Tip

You can declare multiple covers in the same configuration. If you need a specific global configuration for one or more covers, you can also create a new application configuration for these covers.

📄 Full Configuration Example

```yaml
CoversManager:
    module: covers_manager
    class: CoversManager
    use_dictionary_unpacking: true
    log: CoversManager
    config:
        common:
            locker: "binary_sensor.alarm_status"
            position:
                opened: 100
                closed: 0
                min_ratio_change: 5
                min_time_change: 10
            opening:
                type: "prefer-lux"
                time: "10:00:00"
                locker: "binary_sensor.locker_opening" # If at least one of opening and global locker are True, lock
            closing:
                type: "prefer-lux"
                secure_dusk: True
                adaptive: True
                locker: "binary_sensor.locker_closing" # If at least one of closing and global locker are True, lock
            manual:
                allow: true
                timer: 01:00:00
            temperature:
                indoor:
                    sensor: "sensor.indoor_temperature"
                    setpoint: 23
                outdoor:
                    sensor: "sensor.outdoor_sensor_temperature"
                    low_temperature: 25
                    high_temperature: 28
            lux:
                sensor: "sensor.outdoor_sensor_illuminance_lux"
                open_lux: 23
                close_lux: 5
        covers:
            cover.roller_1:
                window_heigh : 210
                window_azimuth: 180
                positional:
                    action : True
                    status: True
                fov:
                    left: 60
                    right: 70
            cover.roller_shutter_2:
                window_heigh : 210
                window_azimuth: 320
                positional:
                    action : True
                    status: True
                fov:
                    left: 90
                    right: 90
```

🪲 Debug

To help to debug and understand an issue, you can enable the debug mode in app. For this, edit the app configuration and set `log_level` to DEBUG

```
CoversManager:
    [...]
    log: CoversManager
    log_level: DEBUG    <---- HERE
    config:
        [...]
```

If you need some assistance, you can open a topic in Discussions or by opening an Issue

🤝 Contributing