

SMART COVER CONTROL MODULE FOR HOME ASSISTANT

Matúš Juštik*

Abstract

The aim of this work is the study and implementation of external blinds control for the Home Assistant home automation platform. The main goal is to design and implement a module capable of controlling blinds that do not have feedback, using only up, down, and stop commands. The designed module must at least comply with automation, such as shading and illumination of the interior according to the position of the sun with maximum use of the day, the adequate response of blinds to protect against possible weather, and other automation. The module will then be published to the appropriate channels to reach the Home Assistant user community. This work brings a better and more efficient solution for controlling blinds within the Home Assistant.

*xjusti00@stud.fit.vutbr.cz, Faculty of Information Technology, Brno University of Technology

1. Introduction

This solution has been developed to address the need for a more efficient and user-friendly way of controlling blinds in the Home Assistant platform. By providing enhanced control capabilities and simplified configuration processes, it aims to improve the overall experience for users who are looking to automate their home environments.

Our solution addresses the challenge of blinds such as **Model 1** that lack feedback mechanisms. It relies on a smart controller connected via Home Assistant for feedback. Our approach is based on predefined time settings that dictate the desired blind position. It's important to note that there is a noticeable gap in the current market, as existing solutions within the platform fail to provide a comprehensive fix to this issue. Each available solution has its limitations, which detract from the overall user experience.

There are currently two available solutions in the Home Assistant Community Store for controlling blinds [1]. However, one of them does not support tilting blinds [2], while the other [3] fails to synchronize if the user tries to control the blinds using a physical switch or any other method outside the integration. Moreover, the tilting function of these solutions does not perform as expected, resulting in incorrect calculations. Both solutions require users to manually

adjust settings in a configuration file. Although they work, they do not operate as expected.

Our solution offers not only straightforward control but also addresses the gap in the Home Assistant Community Store. The current solutions are plagued with usability problems and demand a significant amount of technical expertise to operate effectively. In contrast, our solution stands out for its ease of use and seamless integration. It comes with a user-friendly interface while in configuration settings. Our solution accurately calculates the blind's position to match its real-world counterpart, even during external control scenarios, ensuring a smooth experience for users. Additionally, it includes a feature to remember the blind's position between Home Assistant restarts or unpredictable events, providing added convenience.

2. Problems and Solutions

One of the first issues addressed was the need for easier control and smoother configuration for Home Assistant. The platform already offers pre-set UI elements for this purpose, which can be utilized. However, it is our task to ensure that the blind's position is correctly linked and calculated within the UI. The control elements are visible in Figure 1 and Figure 2. Figure 1 allows control through buttons that indicate the direction of the blind's movement or stop

it altogether when pressed. Figure 2 features sliding buttons that can be used to set the precise position to which the blind should move after adjustment.

As previously mentioned, our goal was to make the process of installing and configuring our system as simple as possible. We accomplished this by allowing users to add integration through the Home Assistant Community Store and then configure it using the user interface displayed in [Figure 4](#). This eliminates the need for users to manually modify Home Assistant's configuration files.

Currently, the addition of our solution through the Home Assistant Community Store is available as a custom integration, and we are awaiting approval for inclusion in HACS. We anticipate that by the time you read this text, the integration will be available in the Home Assistant Community Store, further enhancing accessibility for users and streamlining the steps required to add it to Home Assistant.

Another significant challenge arose during the calculation of blind positions, particularly when tilting blinds. While existing solutions often set the tilt immediately to 0 or 100 during movement, our solution addresses this limitation by simultaneously calculating both movement and tilting. This ensures that minor movements accurately reflect in the UI, preventing discrepancies between displayed and actual blind positions.

Furthermore, our solution effectively accommodates control outside the integration. By supporting functionalities such as remote or physical switch control, users can seamlessly manage blinds using external devices, as depicted in [Figure 6](#). This functionality is achieved through continuous monitoring of switch state changes, enabling our solution to differentiate between control inputs from within or outside the integration environment.

3. Automations

This solution provides a range of built-in automations that are designed to save time and improve the overall experience of using the blinds. For instance, the solution offers scheduled opening and closing of the blinds during the day, enabling users to set specific times for the blinds to open and close automatically. Moreover, the blinds can be set to open or close based on sunrise or sunset times, which helps to adjust the lighting in the room and create a comfortable and cozy atmosphere.

Another useful automation feature is the ability to automatically close the blinds in the evening if the

lights are turned on and it's already dark outside. This helps to maintain privacy and security while also reducing energy consumption.

Furthermore, the solution leverages Netatmo or Open Meteo API to detect wind speed, gusts, and other adverse weather conditions. This feature ensures that the blinds are automatically closed in the event of adverse weather conditions, such as strong winds or heavy rain, to prevent any damage to the blinds.

Overall, these automations make using the blinds a hassle-free experience while also ensuring that they are protected from damage caused by adverse weather conditions.

4. Use case

These are some of the use cases:

1. Blinds can be adjusted either by moving them up and down or by tilting the slats with the help of buttons.
2. Schedule a time for them to move, such as 3:00 PM, and blinds will move to specified spot.
3. Protect the blinds against bad weather. So, when there is a strong wind, gusty wind, or perhaps bad weather altogether, they will open.
4. Set the tilt open during the day so that when you use the blinds, the tilt will remain open.
5. During the night, when you choose a switch that turns the lights on, you can set that to close the blinds when the state of that switch turns on.

5. Conclusions

It is important to mention that while the current solution is functional, it is always possible to improve it further. One area that could be enhanced is the accuracy of the calculations, which could potentially be refined by incorporating more advanced algorithms. Another suggestion could be to add control elements to the system, which would provide greater flexibility and customization options for the user. Additionally, enhancing automation could help to streamline processes even further, allowing for more efficient use of resources. Despite these potential areas for improvement, it is worth noting that both the calculations and automation in the current solution are operating correctly and without error.

Acknowledgements

I would like to thank my supervisor Ing. JAN PLUSKAL, Ph.D. for his help.

References

- [1] Home assistant community store. online, 2023.
- [2] Cover time based integration. online, 2024.
- [3] Cover time based component. online, 2024.