

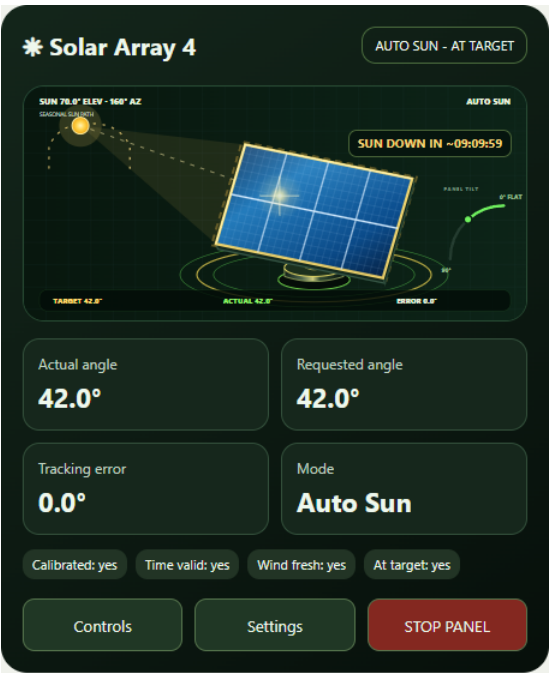


HACS Setup and Solar Tracker Cards

Tryon Electronics | Guide v1.0 | October 12, 2026

Install the Tryon Solar Tracker card once, then choose an Actual Solar Angle entity for each controller. The card shows live Home Assistant readings with the lightweight ESP32-style panel graphic. MQTT is not required when the trackers already use the ESPHome integration.

Step	Result
1. Prepare Home Assistant	Tracker entities are online and HACS is configured.
2. Install the custom repository	One JavaScript resource serves every tracker card.
3. Select a tracker entity	Add Array 4 or another array using its real Actual Solar Angle sensor.
4. Customize the visual options	Choose title, controls, overrides and an optional status template.
5. Update through HACS	Reload each phone/browser after downloading an update.



Sample daytime card from the project test screenshots. Opening the card does not command movement. Control actions operate the real tracker.

Contents: prerequisites; HACS installation; repository/resource setup; adding one or multiple arrays; graphics and controls; entity overrides; status templates; updates and troubleshooting; completion checklist and sources.



Before installing: Home Assistant and trackers

Requirement	Check
Home Assistant	This card requires 2026.6.0 or newer, as declared by its HACS manifest.
Tracker integration	Open Settings - Devices & services - ESPHome. Confirm the intended tracker is connected and its Actual Solar Angle sensor has a valid reading.
HACS	HACS must be downloaded and configured. If it already appears and works in the sidebar, skip the next page.
Account permissions	Use an account able to edit the dashboard and install/manage resources. Home Assistant service permissions still apply to controls.
Extra cards	No button-card, Mushroom or card-mod dependency is required by this standalone card.
MQTT	Optional. Prefer the ESPHome entity set when it already provides the tracker readings and controls.
Firmware	Installing the dashboard card does not compile or flash ESPHome firmware. Firmware updates remain a separate ESPHome operation.

Choose the right entity first

Find the sensor by its friendly name, for example Solar Array 4 Actual Solar Angle. Confirm which physical panel it belongs to. Renamed devices can keep old entity IDs, so the visible array name and sensor ID may differ.

Do not select Requested Solar Angle, a sun elevation sensor, a PV-power sensor or an ESP-NOW peer angle as the main tracker entity. Use the local tracker's calibrated Actual Solar Angle. PV power can be a separate Tracking Status override later.

Three separate settings areas

Area	Purpose
Card editor	Dashboard appearance and which entities the card uses.
Tracker local page / ESPHome entities	Calibration, travel limits, site/time, MQTT and safety tuning.
ESPHome YAML	Hardware package, compiled identity, credentials and firmware defaults.



Install HACS if it is not already configured

Use the official installation instructions for your Home Assistant installation type. For OS/Supervised, the official Get HACS downloader is available; Container/Core use different instructions. After downloading, restart Home Assistant and refresh the browser.

[Official HACS download instructions - select your installation type](#)

Add the HACS integration

1. Open Settings - Devices & services, then Add integration.
2. Search for HACS. If absent after installation/restart, hard-refresh or clear the browser cache.
3. Follow the setup prompts. Use the GitHub device code/link presented by Home Assistant to authorize HACS.
4. Return to Home Assistant, finish the integration setup and open HACS.

[Official HACS initial-configuration instructions](#)

Which GitHub credentials are used?

HACS authorization is separate from the GitHub username/read token in your ESPHome YAML. The card repository is public. You do not put a private firmware token, ESPHome API encryption key or OTA password into this card.

The Get HACS downloader is a way to obtain HACS; the configured HACS integration then manages custom repositories. The Tryon tracker card itself is a dashboard resource, not a firmware package or a Home Assistant integration.



Install the Tryon card through HACS

1. Open HACS and its three-dot menu, then Custom repositories.
2. Paste the repository URL below. Choose Dashboard as its type (older versions may use Lovelace or Plugin), then Add.
3. Find Tryon Solar Tracker Card in HACS and select Download. Follow the dialog; use main when you want the current development branch.
4. Reload the Home Assistant page. The card should become available under dashboard Edit - Add card.

<https://github.com/Tryon-Electronics/tryon-solar-tracker-card>

Open the Tryon card repository in your Home Assistant HACS

This is a custom repository; searching the default HACS catalog alone may not find it. Add its URL first. Menu wording can vary with HACS and Home Assistant versions.

Resource registration

For normal storage-mode dashboards, HACS registers the resource. If the picker reports the card missing, or you manage YAML resources, check that the following module is registered exactly once:

```
url: /hacsfiles/tryon-solar-tracker-card/tryon-solar-tracker-card.js
type: module
```

In a resource editor that calls the type JavaScript module, select that type. With YAML resources, put the entry in the dashboard's resources configuration. Do not put this resource entry inside an individual card.

When migrating from a manual `/local/tryon-solar-tracker-card.js` installation, remove or replace the old resource entry. Registering both old and HACS copies can load the wrong script. Reload afterward.



Add Solar Array 4 and other tracker cards

1. Open the intended dashboard, choose Edit, then Add card.
2. Choose Tryon Solar Tracker. If it is missing, use Manual and paste the example below after checking resource registration.
3. In the visual editor, select Array 4's Actual Solar Angle entity. Save.
4. Add another card for Array 2 or duplicate the first card and change its anchor entity. Verify the newly selected physical tracker before using controls.

```
type: custom:tryon-solar-tracker-card
entity: sensor.solar_array_4_actual_solar_angle
show_controls: true
controls_expanded: false
```

Entity IDs are examples. If Home Assistant kept an older ID after a rename, use the entity picker or copy the actual entity ID from the tracker's device page. Do not rename entities just to match this guide.

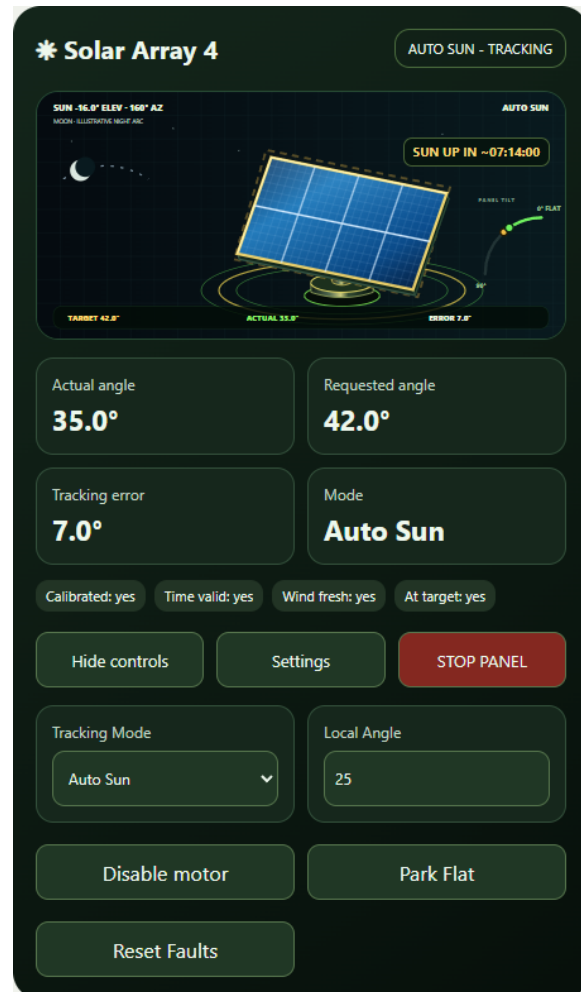
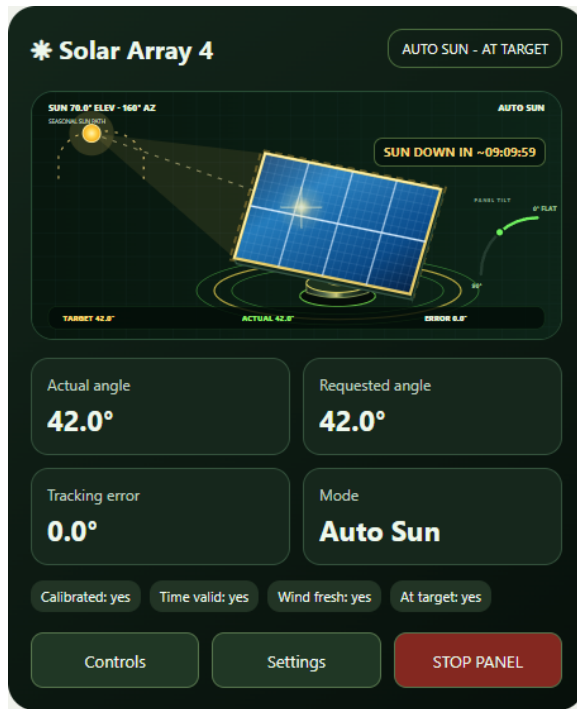
Two-array YAML example

```
type: grid
columns: 2
square: false
cards:
  - type: custom:tryon-solar-tracker-card
    entity: sensor.solar_array_2_actual_solar_angle
    title: Solar Array 2
    show_controls: true
  - type: custom:tryon-solar-tracker-card
    entity: sensor.solar_array_4_actual_solar_angle
    title: Solar Array 4
    show_controls: true
```

For a narrow phone screen, use one column or separate dashboard sections. Each card uses its own anchor; changing a title alone does not change which device it controls.



Basic view, expanded controls and graphics



Left: basic view. Right: expanded controls and night graphic. Both screenshots use sample states for illustration; they are not a claim about your controller's current position or operating mode.

The blue panel reflects the reported actual angle; requested/actual/error readouts show tracking progress. The gauge uses 0 degrees flat and 90 degrees upright. Daytime shows SUN DOWN IN; nighttime shows SUN UP IN. Seasonal paths use site/clock information when available; the moon is illustrative.

Graphics are local SVG with a lightweight dark grid. There is no landscape image or continuous animation. If actual feedback is unavailable, the card does not invent a panel position. A wrong sensor selection cannot be corrected by changing the picture or title.



Card options and real tracker controls

Option	Meaning
entity	Actual Solar Angle anchor. Related entities are discovered on the same device.
title	Optional dashboard heading. Does not rename the ESP32 or change hardware identity.
show_controls	True enables the card controls; false gives a display-only card. It is not a substitute for Home Assistant account permissions.
controls_expanded	True opens the control panel initially; false keeps the compact/basic view.
Entity overrides	Select explicit readings or controls if names changed or automatic discovery is ambiguous.
status / status_template	Sensor or Jinja text for the top-right label. See the template page.

Control	Action
Controls / Hide controls	Expands/collapses the controls; this alone sends no motor command.
Tracking Mode	Selects Auto Sun, Local Angle or Park Flat on the real device.
Local Angle	Sets the controller's requested local target; Local Angle mode uses it within normal limits.
Enable / Disable motor	Toggles normal motor enable. Fault/safety/calibration conditions can still block movement.
Park Flat	Selects Park Flat mode; requests the saved minimum angle, not necessarily exactly 0 degrees.
Reset Faults	Clears latched motor faults. Correct the physical cause before resuming.
STOP PANEL	Calls the tracker Stop Panel button to stop/disable motor operation. It remains visible in the basic view when controls are enabled.
Settings	Opens the selected Home Assistant device page when device discovery is available. Use the tracker's device/local interface for calibration and advanced settings.

Loading the card never commands movement. Unavailable controls are disabled; a button that reports unknown before its first press can still be a valid actionable button.



Entity overrides: readings and configuration

Expand Entity overrides (optional) in the visual editor. Automatic discovery prefers the selected device and original entity names, excluding ESP-NOW peer telemetry. Ambiguous matches require a selection instead of guessing another array.

YAML option / visual label	Domain	What to select
requested / Requested Solar Angle	sensor	Requested target from the selected tracker.
error / Tracking Error	sensor	Target-minus-actual error.
mode / Tracking Mode	select	Auto Sun / Local Angle / Park Flat selector.
status / Tracking Status	sensor	Tracker status, or deliberate alternate reading such as PV watts.
sun_elevation / Sun Elevation	sensor	This tracker's calculated solar elevation.
sun_azimuth / Sun Azimuth	sensor	This tracker's calculated solar azimuth.
latitude / Site Latitude	number	Saved installation latitude.
longitude / Site Longitude	number	Saved installation longitude.
facing / Panel Facing Azimuth	number	Saved orientation of the tilt plane.
minimum / Panel Minimum Angle	number	Saved lower travel limit.
maximum / Panel Maximum Angle	number	Saved upper travel limit.
epoch / Tracker Epoch	sensor	Controller absolute clock for solar timing.

If clock/location entities are unavailable, the graphic can fall back to Home Assistant location/sun information and browser time, marked in the scene. A missing countdown often means location/time or sun.sun information is missing; it is not proof of a motor fault.



Entity overrides: indicators and action buttons

YAML option / visual label	Domain	What to select
safety_park / Safety Bus Park Active	binary_sensor	Combined safety park/hold state.
motor_enable / Motor Enable	switch	Selected tracker's normal motor-enable switch.
local_angle / Local Requested Angle	number	Selected tracker's local target number.
calibrated / Panel Calibrated	binary_sensor	Calibration-valid indicator.
time_valid / Time Valid	binary_sensor	Controller time-valid indicator.
wind_fresh / Wind Data Fresh	binary_sensor	Network wind freshness.
at_target / Panel At Target	binary_sensor	Whether measured angle is within the target band.
motor_fault / Motor Fault	binary_sensor	Aggregate latched motor fault indicator.
angle_fault / Angle Sensor Fault	binary_sensor	Invalid/unavailable angle feedback indicator.
wrong_fault / Wrong Direction Fault	binary_sensor	Optional detailed wrong-direction indicator.
no_move_fault / No Movement Fault	binary_sensor	Optional detailed no-movement indicator.
reset_faults / Reset Motor Fault	button	Fault reset action for this tracker.
stop / Stop Panel	button	Stop/disable action for this tracker.

Wrong Direction and No Movement entities are optional and disabled by default in firmware/Home Assistant. Motor Fault already aggregates those normal faults. Enable the detailed entities on the ESPHome device page if you want their separate indicators.

Missing required data is reported as missing/unknown rather than clear. If Settings is absent, the card may not have resolved the device registry; verify the anchor and integration. Do not select a nearby peer's STOP button to make your own button active.



PV power and your own status template

To display watts in the top-right badge, edit the card visually, expand Entity overrides, and choose your PV-power sensor under Tracking Status. In normal sensor mode the card adds its `unit_of_measurement`, for example 6535 W. No template is needed for that basic result.

Enter a template through the options

Use Tracking Status template (optional), directly below that sensor selector. This template follows whichever sensor you select; it does not hardcode a PV entity name:

```
{{ states(entity) ~ ' ' ~  
  (state_attr(entity, 'unit_of_measurement') or '') }}
```

`entity` means the selected Tracking Status sensor. `tracker_entity` means the Actual Solar Angle anchor. Use `~` to join text; or `" "` handles a missing unit. Leave the template empty to return to normal sensor-plus-unit display.

Equivalent YAML example

```
type: custom:tryon-solar-tracker-card  
entity: sensor.solar_array_4_actual_solar_angle  
status: sensor.totals_pv_power  
status_template: >~  
  {{ states(entity) ~ ' ' ~  
    (state_attr(entity, 'unit_of_measurement') or '') }}
```

`show_controls: true`

Home Assistant evaluates Jinja and pushes changes to the card. Output is escaped text, not HTML. The card does not append a second unit to template output. Invalid expressions show **TEMPLATE ERROR** and a warning until rendering succeeds.

Actual-angle-unavailable, motor-fault, safety-park and motor-disabled status messages take priority over a custom PV label. A template cannot hide those controller conditions.

A sensor state can be unknown/unavailable. For custom templates, decide how to display that case rather than treating it as measured zero. The selected sensor is still needed if your template uses `entity`.



Updates, resources and troubleshooting

Open the repository in HACS. Use its update action when offered, or three-dot menu - Redownload - main for the latest development branch. Reload every phone/browser afterward. All cards share the updated resource; you do not need to edit each card's YAML.

The card update does not update ESPHome firmware, change GPIO wiring or migrate saved tracker settings. Keep the card repository and firmware repository distinct.

Symptom	What to check
Card not found / Custom element does not exist	Confirm HACS download completed, the JavaScript module resource is registered once and the browser reloaded. Use the exact custom type tryon-solar-tracker-card.
Old appearance or missing template box	Redownload main in HACS, refresh/reopen the editor, and check for a duplicate old /local resource. The current guide describes card v0.5.1.
Missing / unknown entities	Confirm the main sensor belongs to the intended device. Pick renamed entities in overrides. Required missing data is not a clear fault state.
Motor Fault / Angle Sensor Fault	Check the physical controller and its service page; updating the card does not fix a hardware fault or calibration.
STOP PANEL disabled	Check that the intended Stop Panel entity exists and is available. An unknown pre-press button state can be normal; unavailable is different.
Duplicate MQTT and ESPHome entities	Select one device/integration consistently. Use the native ESPHome entities when available.
No sunrise/sunset countdown	Check tracker time/location entities and Home Assistant sun.sun/location fallback. Verify the updated card resource loaded.
Several cards slow on a phone	Use the lightweight current card, one-column layout and collapsed controls. Remove duplicate resources. Card rendering delays and ESP32/MQTT controller delays need separate diagnosis.



Completion checklist and references

- [] HACS is configured and the custom repository downloaded.
- [] The HACS JavaScript module is registered once; no stale manual duplicate.
- [] Each card has the correct local Actual Solar Angle anchor.
- [] Actual/requested/error/mode match the physical controller readings.
- [] Required indicators and STOP action belong to the intended device.
- [] Optional overrides/templates render correctly after browser refresh.
- [] Each phone/browser loads the current resource.
- [] Calibration, travel limits and firmware setup remain documented separately.

Verified sources

Card baseline: v0.5.1, commit bc3d239, README, hacs.json, src/card.js and existing test screenshots. Screenshot values are sample states. Guide version v1.0 is independent of firmware and card versions.

[Tryon Solar Tracker card repository and current README](#)

[Official HACS download instructions](#)

[Official HACS initial configuration](#)

[Official HACS custom repositories](#)

[Official HACS dashboard resources](#)

[Tryon wiring guide and settings/YAML manual](#)

Menu labels can change between releases; the linked official documentation and repository README provide current instructions. For support, include Home Assistant/HACS/card versions and unresolved entity names. Do not include passwords, API keys or tokens.