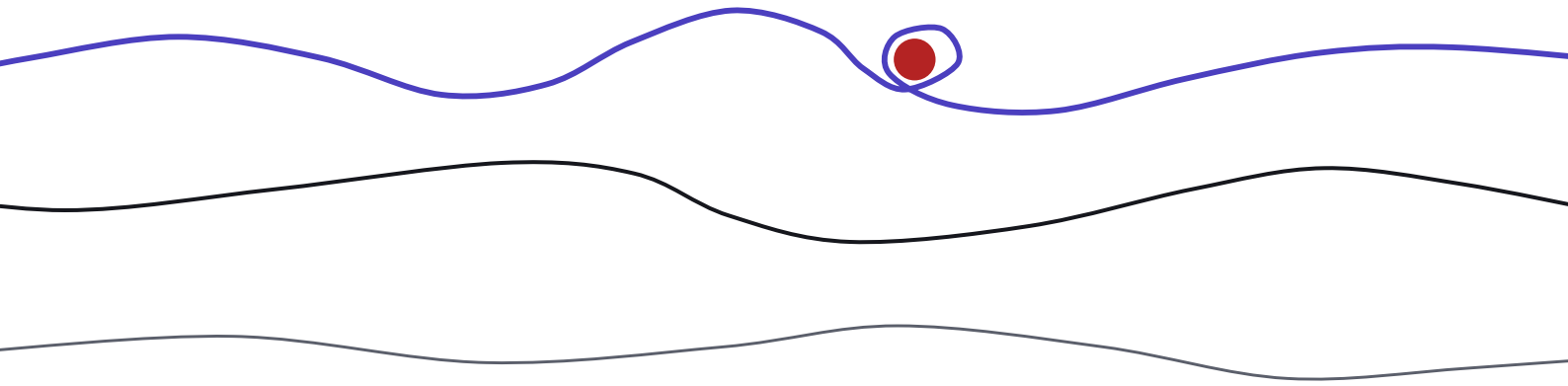
jupiter.au

HVAC Horizon

Runs the heating and cooling of one room in Home Assistant and forecasts the time, energy and cost to reach its target.



VERSION

v0.8.0

DATE

25 November 2026

AUTHOR

belikh

SOURCE

github.com/belikh/hvac-horizon

Contents

00.1 What it does 2
00.2 Who it is for 2
00.3 Where to start 2
00.4 Requirements 3

Start

01 **Install** 4
01.1 Install with HACS 4
01.2 Add a room 4
01.3 First run 5
01.4 Turn on control 5

Guides

02 **Use the room thermostat** 6
02.1 The room thermostat 6
02.2 Modes 6
02.3 Restart delay 7
02.4 Outdoor limits 8
02.5 Target temperature 8
02.6 Low and high target in auto 9
02.7 Fan mode, swing mode and humidity 9
02.8 Humidity target 9
02.9 Presets 9
02.10 Action 10
02.11 Errors 10

03 **Set up the integration** 11
03.1 Open the settings 11
03.2 Areas 11
03.3 Sections 12
03.4 Add a heater 12
03.5 Add an electricity price 13
03.6 Add presence and motion 13
03.7 Add windows, covers and free cooling 14
03.8 Add a second AC or other room devices 14
03.9 Errors on the form 14

04 **Read the forecast** 15

04.1 Forecast sensors 15

04.2 Status values 16

04.3 How long learning takes 17

04.4 How good the model is 17

04.5 Heating and drying 17

04.6 What the weather entity adds 18

04.7 Why the AC runs or not 18

05 **Actions and presets** 19

05.1 Which preset applies 19

05.2 Schedule blocks 19

05.3 Run a preset for a set time 20

05.4 Actions 20

05.5 Lock and PIN 21

05.6 Auto-tune the heater 22

05.7 House mode 22

06 **Troubleshoot** 22

06.1 Find the reason 22

06.2 Action texts 22

06.3 The AC does not start 24

06.4 The heater does not start 24

06.5 Sensor stalled 24

06.6 The AC is on and the compressor does not run 25

06.7 Manual changes 25

06.8 Off and restarts 26

06.9 Repair issues 26

06.10 Send a bug report 26

06.11 A device does not follow a command 26

06.12 The watchdog turned the outputs off 27

Concepts

07 **How it works** 28

07.1 One entry, one room 28

07.2 Room model 28

07.3 Models kept through an idle season 29

07.4 Corrections 29

07.5 Heating and drying 30

07.6 Forecast 30
07.7 Targets 31
07.8 Cooling decision 31
07.9 Heating 31
07.10 Several devices in one room 32
07.11 Safety rules 32
07.12 Storage 32
07.13 Terms 32

08 **Limits of the model** 33
08.1 What the model assumes 33
08.2 What the model does not include 33
08.3 What the forecast assumes 34
08.4 What this means for a reading 34
08.5 Known limits of control 34
08.6 What has run on real devices 35

Develop

09 **Architecture** 36
09.1 Layout 36
09.2 Pure modules 37
09.3 Coordinator 38
09.4 Home Assistant surface 39
09.5 One tick 40
09.6 Watchdog 41
09.7 Free cooling states 41
09.8 Add a setting 41
09.9 Add a published value 42

10 **Test framework** 42
10.1 Four levels 42
10.2 Fake devices 42
10.3 Simulated house 43
10.4 Closed-loop harness 44
10.5 What the closed loop asserts 44
10.6 Add a case 45

11 **Test and release** 45
11.1 Set up the test environment 46

11.2 Run the tests 46
11.3 Test layout 46
11.4 Guard tests 47
11.5 Build the documentation 48
11.6 Continuous integration 48
11.7 Release 48

Reference

12 **Settings** 50
12.1 How settings are stored 50
12.2 Room 50
12.3 Comfort and presets 51
12.4 Price 52
12.5 Heater 53
12.6 Occupancy and windows 54
12.7 Air quality and humidity 55
12.8 Devices and power limits 57
12.9 Advanced 58
12.10 Expert tuning 59
12.11 Room devices (`fleet`) 60

13 **Sensors** 61
13.1 Entities per entry 61
13.2 Primary sensors 61
13.3 Diagnostic sensors 63
13.4 Sensors disabled by default 64
13.5 Binary sensors 67
13.6 Attributes 69

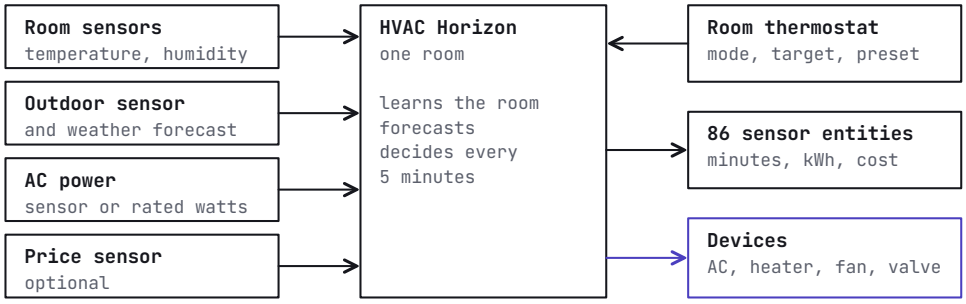
HVAC Horizon

00.1

What it does

HVAC Horizon controls the climate of one room in Home Assistant. You give it an air conditioner or a heater, room temperature sensors and an outdoor temperature sensor. It adds one thermostat for the room, 73 sensors and 13 binary sensors.

- It learns how fast the room gains heat and how fast the AC cools it, from readings it takes every 5 minutes.
- It forecasts the minutes, the energy and the cost for the room to reach its target.
- With control on, it drives the AC, a heater, and any second AC, fan, valve, cover or window actuator you add.



A new entry starts in observe-only mode. Architecture has a diagram of every part. It learns and forecasts, and it sends no commands until you turn control on.

00.2

Who it is for

HVAC Horizon fits one room with an AC, a heater on a switch, or both.

- **An AC with a power sensor** gives the full room model: the forecast of time, energy and cost comes from the measured power.
- **An AC with no power sensor** works with its rated power. The AC counts as drawing it while it reports that it runs, so the energy and the cost are estimates.
- **A room with heating only** has the modes `off` and `heat`. It learns the heater's rate and forecasts the heat-up time.

It does not fit a house you want on one thermostat. Each entry is one room.

AC cooling has run on one real installation. Every other device type has run in tests and in a simulated room only: see What has run on real devices.

00.3

Where to start

YOU WANT TO	READ
Install it and add a room	Install

YOU WANT TO	READ
Choose the mode, the target and the preset	Use the room thermostat
Add a heater, a price sensor or other devices	Set up the integration
Understand a forecast or a status	Read the forecast
Find out why a device is on or off	Troubleshoot
Look up one setting or sensor	Settings, Sensors

00.4

Requirements

- Home Assistant 2026.11.2 or later.
- A `climate` entity for the AC, with the modes `off` and `cool`, or a `switch` for a heater.
- For an AC: a power sensor in watts that reads the whole unit, or its rated power. A wall plug with power metering works.
- At least one room temperature sensor and one outdoor temperature sensor.

SECTION

Start

Install and first run.

01

Install

Install HVAC Horizon with HACS, add a room, and turn on control.

01.1

Install with HACS

[image: Add to Home Assistant]

1. Select the button above. Home Assistant opens the repository in HACS.
2. Select **Download**, then choose the newest version.
3. Restart Home Assistant.

Without the button, open HACS, select the three-dot menu and **Custom repositories**. Enter `https://github.com/belikh/hvac-horizon` with the category **Integration**.

HVAC Horizon needs Home Assistant 2026.11.2 or later.

01.2

Add a room

[image: Add integration]

1. Select the button above. Or open **Settings > Devices & services**, select **Add integration** and search for **HVAC Horizon**.
2. In the **Area** step, select the Home Assistant area of the room and select **Submit**. The next forms then suggest the AC and the sensors found in that area. Leave it empty to pick every entity yourself.
3. In the **Room** step, check the suggested devices and sensors, add the outdoor sensor, then select **Submit**.

FIELD	ENTITY
Area	The area of the room. The room's thermostat and sensors are placed in it
Air conditioner	The `climate` entity of the AC. Leave it empty for a room with heating only
AC power sensor (W)	A power sensor in watts that reads the whole AC unit
AC rated power (W)	The watts the AC draws while it runs, for an AC with no power sensor
Heater switch	The switch of the heater. A room needs an AC or a heater switch
Room temperature sensors	One or more temperature sensors in the room
Outdoor temperature sensor	A temperature sensor outside

4. In the **What next** menu, select **Finish now**.

Home Assistant creates one device named `HVAC Horizon` and the area's name. Without an area the name is the AC's, or the heater switch's in a room with no AC. The device holds the room thermostat and the sensors, and it is placed in the area. One AC or heater switch can have one entry. A second entry for the same device stops with `This device is already set up.`

01.3

First run

The entry starts in observe-only mode. It takes a reading every 5 minutes and sends no commands. After setup it also reads up to 10 days of recorder history in the background.

The Status sensor reads `learning` until the room model has enough data: 3 hours with the AC on, 3 hours with it off, and 3 separate AC runs. A room with heating only needs 12 intervals with the heater on. See Read the forecast.

01.4

Turn on control

With control on, HVAC Horizon sets the AC mode and setpoint and drives the other devices of the entry.

1. Open **Settings > Devices & services** and select **HVAC Horizon**.
2. On the entry, select **Configure**.
3. Select **Advanced** and turn on **Let HVAC Horizon control the room**. Select **Submit**.
4. Select **Save and close**. The room reloads and keeps its stored samples.

The room thermostat then shows its automatic mode: `heat_cool` in a room that can heat and cool, `auto` in a room that only cools. See Use the room thermostat.

> DEVICES START WHEN CONTROL IS ON
>
> With control on and the automatic mode, the AC starts when the room is above the cool threshold. A heater switch starts when the room is below the heat threshold. Set the thermostat to `off` first if you want nothing to start.

SEE ALSO

- Use the room thermostat: modes, targets and presets.
- Set up the integration: heaters, prices, presence and other devices.

SECTION

Guides

Use the thermostat, add devices, read the forecast and find faults.

02

Use the room thermostat

Choose the mode, the target temperature and the preset on the room climate entity, and read what the devices are doing.

02.1

The room thermostat

Each entry has one `climate` entity named `HVAC Horizon <AC name>`. Use it as the thermostat of the room. Leave the AC's own climate entity alone: HVAC Horizon sets its mode and setpoint.

The thermostat needs control on. With control off it reads `off` and its `control` attribute reads `observe only`. See Turn on control.

02.2

Modes

The mode decides which devices may run. A device starts only when the room needs it.

MODE	COOLING DEVICES	HEATING DEVICES
`heat_cool`	Run above the high target's threshold	Run below the low target's threshold
`auto`	Run above the cool threshold	None in the room
`cool`	Run above the cool threshold	Stay off
`heat`	Stay off	Run below the heat threshold
`dry`	The AC runs in its dry mode and does not cool	Run below the heat threshold
`fan_only`	The AC runs its fan and does not cool	Stay off
`off`	Stay off	Stay off

The default is the automatic mode: the room runs whatever it needs. Its name follows Home Assistant's meaning of the two modes.

- **`heat_cool`** in a room that can heat and cool. The card shows a low and a high target.
- **`auto`** in a room that only cools. The card shows one target.

Both are one behaviour, and the pages of this manual call it the automatic mode. A room with heating only has `off` and `heat`. Before 0.7.0 the automatic mode was named `auto` in every room. An automation that sets `auto` on a room that can heat must set `heat_cool`.

The thermostat lists `heat` when the entry can heat: a heater switch, `ac_heat`, `mpc_heat_plan` or a heating fleet device. It lists `dry` when the AC has a dry mode or the entry has a dehumidifier switch. It lists `fan_only` when the first AC has that mode and the room has no fleet devices.

Cooling devices are the AC, the second AC, cooling fleet devices, the whole-house fan and the window actuator. Heating devices are the heater switch, the AC in heat mode, the boiler, the valve and heating fleet devices.

The mode is stored. It is the same after a restart.

Off

`off` sends off to every device of the entry in the same call. No minimum run time delays it. Nothing starts while the room is off, including after a restart of Home Assistant.

To leave `off`, choose a mode. The `climate.turn_on` action returns to the mode you had before `off`.

Dry

In `dry` the AC runs in its own dry mode.

- With a room humidity sensor, the AC dries while the room is above the humidity target (60 % by default) and goes off 3 points below it.
- With no room humidity sensor, the AC dries until you change the mode.
- An entry with a dehumidifier switch runs that switch and leaves the AC off.

Drying cools the room. The heater may run at the same time to hold the heat target. In `dry` the heater obeys the outdoor heating limit, so it stays off when outdoors is above 22 °C. When the room falls 1 K below the heat threshold, the AC pauses. It dries again once the room is back at the threshold.

Fan

In `fan_only` the first AC runs in its own fan mode. Nothing cools and nothing heats, at any room temperature. The fan starts with no restart delay, because no compressor starts. A stalled room sensor turns the AC off.

02.3

Restart delay

A start after a stop waits for the restart delay, 3 minutes by default. The thermostat shows the wait in its `pending` attribute:

ATTRIBUTE PENDING

```
start in 3 min: restart delay
```

The device starts by itself when the delay ends. Set the delay with `min_off_min` in the **Expert tuning** section. 0 turns it off.

02.4

Outdoor limits

An entry with a window actuator or a whole-house fan cools with outdoor air first. In the automatic mode its AC stays off while outdoors is below 16 °C. The limit ends when the open window or the fan has left the room 2 K above the cool threshold for 30 minutes, or cooled it by less than 0.3 K in 30 minutes. An entry with neither has no outdoor cooling limit. `cool` has no outdoor limit: the AC cools whenever the room is above the cool threshold.

In the automatic mode the AC does not cool while the heater runs or within 60 minutes of its last run. A room 2 K above the cool threshold cools at once. This stops the heater and the AC from taking turns.

In the automatic mode and `dry` the heater stays off while outdoors is above 22 °C. `heat` has no outdoor limit: the heater runs whenever the room is below the heat threshold.

When the room is below the heat threshold and a gate holds the heater off, `pending` names the gate:

PENDING	CAUSE
`heat held: outdoors above 22 °C`	The mode is automatic or `dry` and outdoors is above 22 °C.
`heat held: window open`	The window sensor reads open.
`heat held: house mode is away`	The house mode is `away`.
`heat held: another room is cooling`	Another entry of the same house cools.
`heat held: locked`	The lock is on.
`heat held: a sensor has stalled`	A room or outdoor sensor has stopped reporting.

02.5

Target temperature

The target is the comfort temperature, 16 to 32 °C in steps of 0.5. Setting it stores your value, and HVAC Horizon keeps it across restarts.

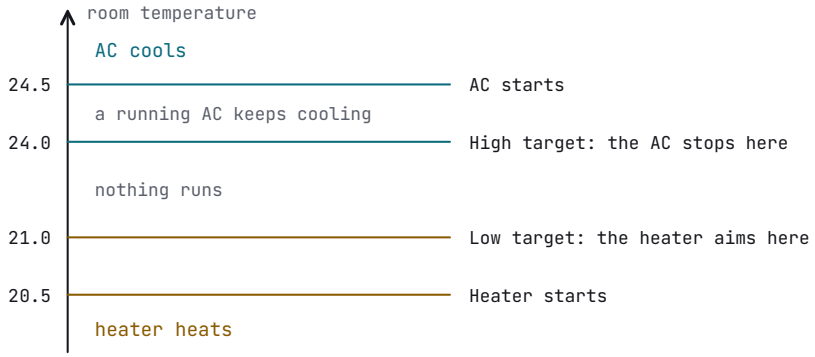
A target you set applies at once. The room returns to the `comfort` preset: a timed preset ends, and a schedule block with another preset is set aside until the schedule moves to its next block. Two states still override it: the vacation switch, and an away house or an empty room.

The thermostat shows the target you set, with these exceptions:

- A preset other than `comfort` shows its own setpoint.
- While the vacation switch is on, it shows `vacation_c`.
- In `heat` mode it shows the heat target. In `cool`, `dry`, `fan_only` and `off` it shows the cool target.
- In the automatic mode, a room that can heat shows two targets. See below.

The threshold the AC cools above can sit above the cool target. A high price raises it by up to 1 K, and the comfort-versus-energy weight and the mould bump raise it too. A low price does not lower it. The Cool above sensor shows the threshold. See Actions and presets.

The AC cools when the room is above the cool threshold. The heater heats when the room is 0.5 K or more below the heat target. The sensors Cool above and Heat below show both values.



02.6

Low and high target in auto

In the automatic mode, a room that can heat shows a low and a high target. The room heats to the low target and cools above the high one. Set both on the card, or with `climate.set_temperature` and `target_temp_low` and `target_temp_high`. The low target must not be above the high target.

Setting one temperature, in any mode, sets both targets to it. `cool` mode shows the high target and `heat` mode the low one.

02.7

Fan mode, swing mode and humidity

The card lists the fan modes and swing modes of the first AC, when the AC has them. A change goes to that AC at once. With `auto_fan` on, the integration sets the fan mode again on its next tick.

With a room humidity sensor set, the card shows the room humidity.

02.8

Humidity target

The card has a humidity target when the entry has a room humidity sensor and a device that dries: a dehumidifier switch or an AC with a dry mode. Set it on the card or with `climate.set_humidity`. The range is 35 to 75 % and the default is 60 %. The target is stored.

DEVICE	WHEN IT RUNS
Dehumidifier switch	Above the target, in every mode. Off 3 points below it.
AC dry mode	Above the target, in `dry` mode or with `dry_for_humidity` on. Off 3 points below it.
Humidifier switch	Below 40 %, or below the target minus 20 when the target is under 60 %.

02.9

Presets

PRESET	COOLS ABOVE	HEATS TO
`comfort`	Your target	Your target
`eco`	`eco_c`, or your target when unset	`heat_eco_c`, or your target minus 1
`boost`	`boost_c` (23 °C)	`heat_boost_c`, or your target plus 1
`away`	`away_c` (27 °C)	16 °C

A preset you choose holds until the schedule moves to another block. It is not kept across a restart. See Actions and presets for schedule blocks and timed presets.

02.10

Action

The `hvac_action` attribute shows what the devices do now.

ACTION	MEANING
`cooling`	The AC is in cool mode and draws 0.2 kW or more, or a cooling fleet device runs
`heating`	The heater switch is on, the AC heats, or a heating fleet device runs
`drying`	The AC is in dry mode and draws 0.2 kW or more, or the dehumidifier runs in `dry` mode
`fan`	The AC is in its fan mode and nothing heats or cools
`idle`	Nothing runs, or the AC is in cool, heat or dry mode and draws under 0.2 kW
`off`	The room is held off

The action follows the devices as they change. The mode is your choice and does not change when a device starts or stops.

02.11

Errors

MESSAGE	CAUSE	FIX
`HVAC Horizon entry is locked`	The lock is on, so the target, the preset and the mode cannot change.	Unlock with `hvac_horizon.set_lock`. See [[guides/services-and-presets#lock-and-pin

SEE ALSO

- Actions and presets: presets by schedule, timed presets and the actions.
- Troubleshoot: why a device is on or off.

03

Set up the integration

Add a heater, a price sensor, presence, windows and other devices to an HVAC Horizon entry, one form section at a time.

03.1

Open the settings

An entry has two places for its settings, as Home Assistant integrations do.

BUTTON	WHAT IT CHANGES	STORED IN
Configure	Every setting you tune: presets, price, heater, occupancy, devices, control	The entry's options
Reconfigure, in the three-dot menu	The room itself: its area, AC, power sensor, heater switch and sensors	The entry's data

To change a setting:

1. Open **Settings > Devices & services** and select **HVAC Horizon**.
2. On the entry, select **Configure**. The menu is titled **Configure HVAC Horizon**.
3. Select a section, change its fields and select **Submit**. The menu returns.
4. Select **Save and close**. The room reloads with the new settings and keeps its stored samples.

To change the room's devices or sensors, select the three-dot menu, then **Reconfigure**. It opens the **Room** form. On submit it reports `Room updated. Stored samples are kept.`

In **Configure**, nothing is saved until you select **Save and close**. A section you do not open keeps its values. A field you clear in a section is removed. The Lock PIN is the exception: an empty PIN box keeps the stored PIN.

The same sections appear when you add a new entry, after the **Room** step. Submit a section empty to keep its defaults.

Each field has a help text under it. It says what the field needs, what it changes, its range and its default, and when to leave it alone. A field that depends on another one says so, for example `Needs a heater switch.`

03.2

Areas

A room can belong to a Home Assistant area. Pick it in the **Area** step of a new entry, or in the **Room** form of **Reconfigure**.

- **Suggestions**. A new entry's forms suggest the entities of the area. The AC and its power sensor are suggested when the area has exactly one of each. Every temperature sensor is suggested as a room sensor, with the first humidity, motion and window sensor. A suggestion is stored only when you submit its form, so check each one.
- **Never suggested**. The heater switch, because a switch can be anything, and the outdoor sensor, because it is not in the room.
- **Name and place**. The entry is named after the area, and its thermostat and sensors are placed in it. Home Assistant places the device when it creates it, from the device's suggested area. Changing the area in **Room** moves it. A device you moved or removed from its area by hand

stays as you left it.

An entry reads the entities you stored. A sensor you add to the area later is suggested in the matching form of **Configure** or **Reconfigure**, and used after you submit that form.

03.3 Sections

SECTION	WHAT YOU SET THERE
Room (Reconfigure)	The area, the AC and its power sensor or rated power, the heater switch, the room sensors and the outdoor sensor
Comfort and presets	Setpoints for each preset, the vacation switch
Schedule blocks	Presets by time of day, one block per submit
Price	The price sensor and the cheap and expensive thresholds
Heater	The heater's power and algorithm, a valve, a boiler, AC heat. The heater switch itself is in Room
Occupancy and windows	Presence, motion, window sensors, a cover, a window actuator, the weather entity
Air quality and humidity	Humidity, CO2 and PM2.5 sensors, fans, a dehumidifier, a humidifier
Devices and power limits	A second AC, the dispatch mode, load shedding
Room devices	Extra coolers and heaters the integration chooses between, one device per submit
Advanced	Control, the lock PIN, peer rooms, the investigator
Expert tuning	PID and TPI gains, output limits, minimum on and off times, the stall window. The defaults suit most rooms

Settings lists every field of each section with its default and its effect.

03.4 Add a heater

1. Open **Room** and select the heater's switch in **Heater switch**.
2. Select **Submit**, then **Save and close**.

The room thermostat now lists `heat`. The options in the table below are in **Heater**. The heater runs when the room is 0.5 K or more below the heat target. In the automatic mode it also needs outdoors at or below 22 °C. `heat` has no outdoor limit. The heater output is a share of a 10 minute window, so the switch turns on and off inside each window.

TO	SET
Use a switch where on means off	`heater_inverted`
Heat with the AC and no heater switch	`ac_heat`
Account for radiators that stay warm after they stop	`radiators`
Use the TPI algorithm in place of PID	`heater_algorithm` to `tpi`

TO	SET
Limit the heater output	`output_min`, `output_max` in **Expert tuning**

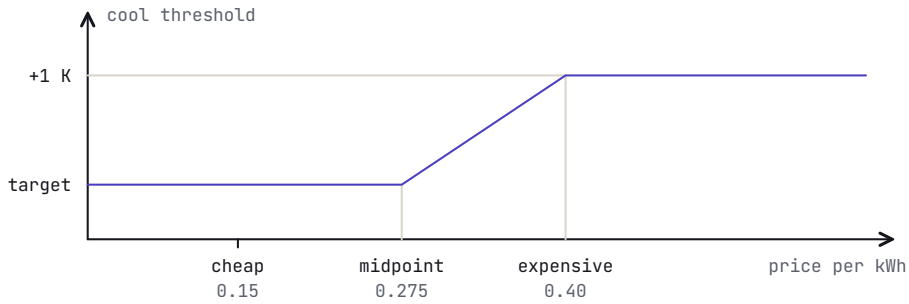
The PID gains come from a relay test when you run one. See `start_autotune` in Actions and presets.

03.5

Add an electricity price

1. Open **Price** and select the price sensor. Its unit is per kWh.
2. Set **Cheap price** and **Expensive price** in the same unit. The defaults are 0.15 and 0.40.

The price only raises the cool threshold. Up to the midpoint of the cheap and the expensive price the threshold is your target. From there it rises in a line, and at or above the expensive price it is 1 K higher.



The Cost to setpoint sensor needs a `forecast` attribute on the price sensor: a list of `time` and `value` pairs.

03.6

Add presence and motion

1. Open **Occupancy and windows**.
2. Select one or more entities in **Presence sources**. A source in state `home` or `on` means home.
3. Select a motion sensor in **Motion sensor in the room** if the room has one.

SITUATION	COOLS ABOVE	HEATS TO
Someone is home, or motion in the room	The preset's setpoint	The preset's heat setpoint
Motion idle for `motion_away_min` (30 minutes), someone home	`eco_c`, or `away_c` when unset	`heat_eco_c`, or comfort minus 1
Every presence source away for `away_delay_min` (30 minutes)	`eco_c`, or `away_c` when unset	16 °C

Motion in the room counts as occupied whatever the presence sources say.

03.7

Add windows, covers and free cooling

DEVICE	FIELD	WHAT HVAC HORIZON DOES
Window or door sensor	`window`	Pauses cooling and heating when the window is open longer than `window_delay_min` (10 minutes)
Cover or blind	`cover`	Closes the cover above 0.5 kW of solar gain. Needs `window_area_m2`.
Window actuator	`window_actuator`	Opens the window when outdoors is at least 2 K cooler than the room and at or below 24 °C
Whole-house fan	`whole_house_fan`	Runs when the room is above the cool threshold and outdoors is at least 2 K cooler

The AC does not cool while the entry holds the window open or runs the whole-house fan. When the room is still 2 K above the cool threshold after 30 minutes, the window closes, the fan stops and the AC cools. The same happens when the room is above the cool threshold and fell by less than 0.3 K in 30 minutes.

With an outdoor humidity sensor in `humidity`, the window and the economizer stay shut when the outdoor dew point is 1 K or more above the room's.

03.8

Add a second AC or other room devices

For two ACs that always run together, set `climate_2` and `power_2` in **Devices and power limits**.

For devices that HVAC Horizon should choose between, open **Room devices**. Give the new device a name, its kind (`cool` or `heat`), the climate entity or heater switch that drives it and its rated power, then select **Submit**. Turn on **Add another device after this one** to stay on the form. To remove devices, tick them under **Remove these devices**. See Room devices for each field.

`dispatch_mode` sets the choice. `fast` runs the device with the largest effect first. `economy` runs the device with the lowest cost per kelvin first. A device with no power sensor counts its `rated_w` while it runs.

03.9

Errors on the form

MESSAGE	CAUSE	FIX
`This device is already set up.`	Another entry uses the same AC or heater switch.	Reconfigure that entry.
`Pick an air conditioner or a heater switch. A room needs one of them.`	The Room step has neither.	Select one of them.
`Pick the AC's power sensor, or enter its rated power in watts.`	The AC has neither.	Select the sensor or enter the watts.

MESSAGE	CAUSE	FIX
`This is a room thermostat of HVAC Horizon. Pick the climate entity of the AC itself.`	The AC field holds the thermostat of an HVAC Horizon room.	Select the AC's own climate entity.
`The cheap price threshold must be below the expensive price threshold.`	The two prices are equal or swapped.	Lower the cheap price or raise the expensive one.
`The new block overlaps the block 22:00 to 06:30: eco.`	A new schedule block shares a time of day with the block named. The schedule uses one block at a time.	Remove that block first, or change the times.
`The minimum heater output must not be above the maximum.`	`output_min` is above `output_max` in Expert tuning.	Swap or correct the two values.
`Set both TPI thresholds, or leave both empty.`	One TPI threshold is set alone, which has no effect.	Set the other one, or clear both.
`Set both the start and the end of the new block.`	A schedule block has one time only.	Set both times, or clear both to add no block.
`The start and the end of a block must differ.`	Both times of a schedule block are the same.	Change one of them.
`Room devices are invalid` followed by a reason	The new device breaks a rule, such as a repeated name or a cooler on a switch.	Correct the field named in the reason.

A section with an error stores nothing from that section.

SEE ALSO

- Settings: every field with its default and effect.
- Use the room thermostat: modes, targets and presets.

04

Read the forecast

Read the minutes, energy and cost to the target, the forecast status, and how long the room model takes to learn.

04.1

Forecast sensors

The forecast answers one question: with the AC running at its usual power, when does the room reach the cool threshold? While the room heats or dries, the same sensors show the heating or the drying forecast. See Heating and drying.

SENSOR	UNIT	MEANING
Minutes to setpoint	min	Time for the room to reach the threshold
Minutes to setpoint, low and high	min	The range around that time, from the uncertainty of the cooling rate
Energy to setpoint	kWh	AC energy for that time. An estimate when the AC has `ac_rated_w` and no power sensor.
Cost to setpoint	price unit	That energy priced slot by slot. Needs a price sensor with a `forecast` attribute. Empty when the price forecast does not cover the whole time: a cost over part of the time is not shown. The `reason` attribute of the Status sensor then reads `reachable; cost unknown: the price sensor has no price for part of this time`.
Status	text	The state of the forecast. See the next table.
Forecast error	min	Median error of the forecasts replayed against past cool-downs

With control off, the forecast aims at the AC's own setpoint. With control on, it aims at the cool threshold of the preset in use.

04.2

Status values

STATUS	MEANING	WHAT TO DO
`learning`	No room model yet. The `reason` attribute names the rule that is not met.	Let the AC run through on and off periods.
`no input`	A room sensor, the outdoor sensor or the AC entity has no reading.	Check the entity in **Developer tools > States**.
`no running power yet`	The AC draws under 0.2 kW and no stored reading holds a running power.	Run the AC once.
`at setpoint`	The room is at or below the threshold. Minutes, energy and cost are 0.	Nothing.
`reachable`	The forecast has minutes, energy and cost.	Nothing.
`unreachable`	The room does not reach the threshold within 12 hours at the current power.	The AC is too small for the conditions, or the model is still coarse.

04.3

How long learning takes

HVAC Horizon takes a reading every 5 minutes. The first room model needs all of these:

1. 36 readings with the AC on and 36 with it off: 3 hours of each.
2. 3 separate AC runs.
3. An outdoor minus room difference that varies by at least 1 K.
4. AC power that varies by at least 0.05 kW.

The model refits every 12 readings, about once an hour. The store keeps 8640 readings, 30 days. After setup the integration also reads up to 10 days of recorder history in the background, so a room with history can have a model on the first day. When a device stays off for more than 30 days its last model is kept and the Kept models sensor gives its age. See Models kept through an idle season.

Readings taken while the AC heats, dries or runs its fan only stay out of the cooling model. Readings taken while the outdoor sensor is stalled are not stored.

04.4

How good the model is

The Status sensor carries the fit quality in its attributes.

ATTRIBUTE	MEANING
`fit_confidence`	`low`, `medium` or `high`, from the standard errors of the fit
`k_per_h`	Heat gain toward outdoors, per hour
`c_k_per_kw`	Cooling per kW of AC power, in K per hour
`g_k_per_h`	Constant gain from people and appliances, in K per hour
`c_stderr`	Standard error of the cooling rate. It sets the low and high minutes.
`fit_residual_k_per_h`	Typical error of the model on its own readings, in K per hour

A new room reads `low` because it has few readings with the AC on. The confidence rises as the AC runs.

The Forecast error sensor comes from a backtest. For each past cool-down, the backtest fits a model on the readings before it and compares the predicted minutes with the measured minutes. The `accuracy_pairs` attribute lists the last 20 runs. The `range_coverage_pct` attribute is the percentage of `range_runs` runs whose measured minutes fell inside the forecast range of that run. A range of one standard error holds about 68 % of runs when the model fits the room. A lower value means the range is too narrow for this room. A run needs 60 readings, 5 hours, before its cool-down.

04.5

Heating and drying

The Forecast mode sensor names what the forecast sensors show.

FORECAST_MODE	WHEN	MINUTES TO SETPOINT COUNTS DOWN TO
`cool`	Every other time, including `fan_only`	The cool threshold
`heat`	The mode is `heat`, or `heat_cool` with the room below the heat target	The heat target
`dry`	The mode is `dry`	3 points below the humidity target (57 % by default)

In observe-only the mode is the AC's own mode. The `forecast_target` sensor holds the target of the forecast shown.

- **Heater switch.** The minutes come from the learned heat rate. Set `heater_power_w` in **Heater** to get the energy and the cost. Without it both are empty.
- **AC heat mode.** The minutes, the energy and the cost come from heat-mode intervals of the AC. The low and high minutes come from the standard error of that rate.
- **Dry mode.** It needs a room humidity sensor. The energy uses the mean AC power of the dry intervals.

The status reads `learning` until 12 intervals of that kind exist. The reason attribute names what is missing.

04.6

What the weather entity adds

Set `weather` in the **Occupancy and windows** section to give the forecast an outdoor temperature that changes over time.

- The integration calls `weather.get_forecasts` for the daily and hourly forecast, at most every 30 minutes. It keeps the last good result when a call fails.
- The hourly forecast becomes a temperature series that starts at the outdoor sensor's reading. The offset between the sensor and the forecast fades out over 3 hours.
- The integration learns the difference between your outdoor sensor and the forecast for each hour of the day, and applies it to the series.
- When the outdoor sensor stalls and the hourly forecast is under 2 hours old, the forecast temperature stands in. The Outdoor source sensor then reads `forecast`.

Without a weather entity, the forecast holds the outdoor temperature at its current value.

04.7

Why the AC runs or not

The Daily briefing sensor shows the next action in one line:

DAILY BRIEFING

```
Day: warm. Comfort 24.0 C. Next: cool: above comfort target.
```

The Status sensor's `investigation` attribute explains the current action and lists the recent ones. Action texts lists every text.

SEE ALSO

- Troubleshoot: what to check when the status does not change.
- Limits of the model: what the model leaves out.

- Sensors: every sensor and attribute.

05

Actions and presets

Set presets by schedule or timer, and call the HVAC Horizon actions for the lock, house mode, auto-tune, export and what-if.

05.1

Which preset applies

A preset sets the cool threshold and the heat target. HVAC Horizon uses the first of these that applies:

1. A timed preset, while its minutes run. See `start_timed_preset` below.
2. The preset you chose on the room thermostat, until the schedule moves to another block.
3. The schedule block that covers the current local hour.
4. `comfort`.

PRESET	COOLS ABOVE	HEATS TO
`comfort`	`comfort_c`	`comfort_c`
`eco`	`eco_c`, or `comfort_c` when unset	`heat_eco_c`, or comfort minus 1
`boost`	`boost_c`, or `comfort_c` when unset	`heat_boost_c`, or comfort plus 1
`away`	`away_c`	16 °C

Three things move these values:

- **Price.** With a price sensor, the cool threshold rises above the midpoint of the cheap and the expensive price, up to 1 K at the expensive price. A low price leaves it at your target.
- **Boost with a heater.** When heating is allowed, boost lifts its cool threshold to the heat target plus 0.5 K. The AC then does not cool while the heater heats.
- **Vacation.** While the vacation switch is on, `vacation_c` replaces the comfort target and heating holds 16 °C.

05.2

Schedule blocks

Open **Configure** and select **Schedule blocks**. A block applies a preset between two times of day.

1. Set **New block starts at** and **New block ends at**.
2. Choose the preset: `comfort`, `eco`, `boost` or `away`.
3. Select **Submit**. Turn on **Add another block after this one** first to stay on the form.

To remove blocks, tick them under **Remove these blocks** and select **Submit**.

A block whose end is before its start runs past midnight. Times outside every block use `comfort`. Two blocks cannot share a time of day: the form refuses a new block that overlaps one you have. The entry stores the blocks as `schedule_blocks`, a list of `start_h`, `end_h` and `preset`, which the diagnostics download shows.

MESSAGE	CAUSE
`Set both the start and the end of the new block.`	One of the two times is empty.
`The start and the end of a block must differ.`	Both times are the same.
`The new block overlaps the block 22:00 to 06:30: eco.`	The new block shares a time of day with the block named.

Before a comfort block, the heater starts early when the learned heat-up time needs it. The Heat-up time sensor shows the estimate in minutes.

05.3

Run a preset for a set time

BOOST FOR ONE HOUR

```
action: hvac_horizon.start_timed_preset
data:
  entry_id: 01M7W2QT14QZYW5QG2879X0FHA
  preset: boost
  minutes: 60
```

`minutes` is 1 to 720. When the time ends, the schedule preset applies again. Find the `entry_id` in the URL of the entry's page, or in its diagnostics download.

05.4

Actions

The actions exist as soon as the integration is installed. An action on an entry that is not loaded fails with `the HVAC Horizon entry <title> is not loaded`. A value an action refuses, such as a setpoint out of range, shows its reason in Home Assistant.

Every action except `set_house_mode` takes `entry_id`. The actions marked response only return data and change nothing. The actions marked admin need an administrator: a call from another user fails with `Unauthorized`.

ACTION	FIELDS	WHAT IT DOES
`hvac_horizon.start_timed_preset`	`preset`, `minutes`	Runs the preset for `minutes`.
`hvac_horizon.set_house_mode`	`mode`: `home` or `away`	Sets the house mode for every entry. `away` makes each room use its away setpoint and heat to 16 °C.
`hvac_horizon.set_lock`	`locked`, `pin`	Locks or unlocks control. Admin.
`hvac_horizon.start_autotune`	`setpoint`	Starts a relay test of the heater and stores PID gains from it. Admin.
`hvac_horizon.cancel_autotune`	none	Ends a running relay test and turns the heater off.

ACTION	FIELDS	WHAT IT DOES
`hvac_horizon.respond_suggestion`	`accept`	`true` stores the current heat-gain rate as the baseline. `false` hides the suggestion until the rate changes. Admin.
`hvac_horizon.generate_investigation_report`	none	Builds a 24-hour report on the Status sensor. Admin.
`hvac_horizon.reset_learning`	none	Clears the stored readings, the room model, the kept models, the backtest and the outdoor bias. Settings stay. Admin.
`hvac_horizon.export_samples`	none	Response only. Returns `samples`, the stored readings. Admin.
`hvac_horizon.export_csv`	none	Response only. Returns `csv`, the stored readings as CSV text. Admin.
`hvac_horizon.what_if`	`room_c`, `target_c`, `outdoor_c`, `power_kw`	Response only. Returns `status`, `minutes`, `energy_kwh`, `cost` and `reason` for the values you give.

`what_if` accepts 0 to 50 °C for `room_c` and `target_c`, -20 to 60 °C for `outdoor_c`, and 0 to 10 kW for `power_kw`.

05.5

Lock and PIN

The lock stops normal commands, so nobody changes the room by accident.

1. Open **Configure**, select **Advanced** and enter a value in **Lock PIN**. Select **Submit**, then **Save and close**.
2. Call `hvac_horizon.set_lock` with `locked: true`.
3. To unlock, call it with `locked: false` and `pin` set to the PIN.

`set_lock` needs an administrator, to lock and to unlock.

While locked, HVAC Horizon sends no AC setpoint, valve, cover or window command, and holds the heater output at 0. The room thermostat refuses a new target, preset or mode. A stalled sensor, a window open longer than `window_delay_min` or an away house still turns the AC and the room devices off. The lock is stored and stays on after a restart.

MESSAGE	CAUSE	FIX
`lock not changed: wrong PIN, or no PIN configured`	The entry has no PIN, or the `pin` in the call is wrong.	Set a PIN in **Advanced**, or send the right one.
`HVAC Horizon entry is locked`	You changed the thermostat while the lock is on.	Unlock first.

05.6

Auto-tune the heater

`start_autotune` switches the heater on and off around a setpoint and derives PID gains from the room's swing.

START A RELAY TEST

```
action: hvac_horizon.start_autotune
data:
  entry_id: 01M7W2QT14QZYW5QG2879X0FHA
  setpoint: 22
```

The setpoint is 15 to 26 °C and at least 2 K above the room. The test ends after 120 minutes, when the AC cools, or when the room reaches the heat threshold. The Status sensor's `autotune_status` attribute reads `running`, `done`, `timed_out`, `cancelled` or `reached_cutoff`. The gains apply to any of `heater_kp`, `heater_ki` and `heater_kd` that you left at its default.

> THE HEATER RUNS DURING THE TEST
>
> The relay test turns the heater on at full output in cycles. Do not start it while you test the AC.

05.7

House mode

`set_house_mode` stores the mode with each entry. It survives a restart. The default is `home`.

SEE ALSO

- Use the room thermostat: modes, targets and presets on the thermostat.
- Settings: the fields for presets and setpoints.

06

Troubleshoot

Find out why the AC or the heater is on or off, what each action text means, and how to clear each repair issue.

06.1

Find the reason

HVAC Horizon writes one action text on each tick. Read it in two places:

- The Daily briefing sensor, after `Next:`.
- The `investigation` attribute of the Status sensor, with the recent actions.

Then find the text in the table below.

06.2

Action texts

TEXT	CAUSE	WHAT TO DO
`cool: above comfort target`	The room is above the cool threshold. The preset name replaces `comfort` under another preset.	Nothing.
`off: at or below comfort target`	The room needs no cooling.	Nothing.
`off: user turned the room off`	The room thermostat is `off`.	Choose a mode.
`off: mode is heat: no cooling`	The room is above the cool threshold and the thermostat is in `heat` or `dry`. The text names the mode.	Choose `cool` or the automatic mode.
`fan: the AC runs its fan only`	The thermostat is in `fan_only`.	Choose another mode to cool or heat.
`cool: waiting for restart delay`	A start waits for `min_off_min` after the last stop. The same text exists for `heat` and `dry`.	Wait. The `pending` attribute of the thermostat shows the minutes.
`cool: waiting for compressor group`	Another entry in the same `compressor_group` started its AC less than 5 minutes ago.	Wait.
`off: window open N min`	The window sensor has been open for `window_delay_min` or longer.	Close the window.
`off: free cooling: window open or whole-house fan`	The entry holds the window actuator open, or the whole-house fan runs.	Nothing. The AC cools when the room is still 2 K above the cool threshold after 30 minutes, or fell by less than 0.3 K in 30 minutes.
`off: auto: the heater ran in the last 60 min`	The mode is automatic and the heater runs, or stopped less than 60 minutes ago.	Nothing. A room 2 K above the cool threshold cools at once. Choose `cool` to cool now.
`off: outdoor N C below cooling limit`	The mode is automatic, the entry has a window actuator or a whole-house fan, and outdoors is below 16 °C.	Nothing. The window or the fan cools the room. Choose `cool` to run the AC.
`off: away, room within setback`	The room is unoccupied and below `eco_c` or `away_c`.	Nothing.
`cool: pre-cool starts now`	A schedule event is near and the room needs the time to cool.	Nothing.
`off: dry paused, room below heat threshold`	Dry mode cooled the room to 1 K below the heat threshold.	Nothing. Drying resumes at the threshold.
`off: night hold: AC does not start`	`night_plan_act` is on and the night plan is `hold`.	Turn `night_plan_act` off to cool.
`off: sensor stalled`	A sensor stall blocks the AC.	See Sensor stalled.
`off: no room reading`	No room sensor has had a valid reading for the stall window.	Check the room sensors.
`inhibited after a manual change`	Someone changed the AC setpoint by hand, or turned the AC off at the unit, in the last 30 minutes.	See Manual changes.
`locked`	The lock is on.	Unlock with `hvac_horizon.set_lock`.

TEXT	CAUSE	WHAT TO DO
`locked: off, sensor stalled`	The lock is on and a safety stop turned the AC off. The reason can also be `window open` or `house away`.	Clear the cause, or unlock.

06.3

The AC does not start

Check these in order.

1. **Control.** The thermostat reads `off` and its `control` attribute reads `observe only` while control is off. See Turn on control.
2. **Mode.** The AC cools in the automatic mode and `cool` only.
3. **Threshold.** The AC cools when the room is above the Cool above sensor. A room 0.5 K or less above it does not start the AC.
4. **Restart delay.** The `pending` attribute reads `start in N min: restart delay`.
5. **The AC's own sensor.** An AC whose built-in sensor reads cooler than the room may not run its compressor at the room's target. `setpoint_regulation` sends a lower setpoint to make up the gap. The sensors Setpoint sent and Sensor offset show the values.

06.4

The heater does not start

1. **Mode.** The heater runs in the automatic mode, `heat` and `dry` only. It stays off in `cool` and `fan_only`.
2. **Outdoor limit.** In the automatic mode and `dry` the heater stays off while outdoors is above 22 °C, and the Heating allowed binary sensor is off. Choose `heat` to heat on a warm day.
3. **Threshold.** The heater runs when the room is below the Heat below sensor.
4. **AC cooling.** The heater stays off on a tick where the AC cools, and while another entry of the same house cools.
5. **Window and house.** An open window, an away house, vacation and the lock each hold the heater at 0.

When the room is below the heat threshold and one of these holds the heater, the `pending` attribute of the thermostat names it, for example `heat held: outdoors above 22 °C`.

06.5

Sensor stalled

A sensor is stalled when its last report is older than the stall window, 900 seconds by default. Set the window with `stall_window_s` in **Expert tuning**, 300 to 3600 seconds. A sensor that repeats the same value still reports, so a steady reading is not a stall. `unavailable` and `unknown` count as no reading.

CONDITION	AC AND ROOM DEVICES	HEATER
The outdoor sensor is stalled and no hourly forecast under 2 hours old exists	Off	0 %

CONDITION	AC AND ROOM DEVICES	HEATER
The outdoor sensor is stalled and a fresh hourly forecast exists	Run on the forecast temperature	Runs on the forecast temperature
Every room sensor is stalled	Off	0 %
One room sensor is stalled and another is fresh	Run on the fresh sensors	Unchanged

A stall also keeps the window actuator and the whole-house fan from starting. HVAC Horizon raises the repair issue `HVAC Horizon sensors not updating` with the list of sensors. The issue clears when they report again.

06.6

The AC is on and the compressor does not run

The AC failure binary sensor turns on, and a repair issue and a notification appear, when all of these hold:

- The entry has a power sensor.
- The first AC has been in `cool` or `dry` for 20 minutes or longer.
- The room still needs it for that whole time. For `cool` the room is more than 0.5 K above the Cool above sensor. For `dry` the room humidity is at or above the humidity target.
- The power was under 200 W for more than half of the last 20 minutes.

Check the unit for an error code. On a portable unit, check whether the water tank is full: such a unit keeps reporting its mode while the compressor rests.

The sensor turns off when the compressor has run for 15 of the last 20 minutes, when the AC leaves the mode, or when the room no longer needs it.

Two cases get no check: `dry` mode in a room with no humidity sensor, and the 20 minute coil dry after cooling.

A unit that rests its compressor for more than half the time in `dry` mode by design raises the issue while the humidity is above the target. Ignore the repair issue and disable the binary sensor for such a unit.

To get the alert on a phone, add an automation on the binary sensor:

NOTIFY WHEN THE AC STOPS DRYING OR COOLING

```yaml
triggers:
  - trigger: state
    entity_id: binary_sensor.hvac_horizon_bedroom_ac_failure
    to: "on"
actions:
  - action: notify.mobile_app_phone
    data:
      message: The AC is on and its compressor is not running. Check the water tank.
```

06.7

Manual changes

When the AC setpoint differs from the value HVAC Horizon last wrote by more than 0.05 K, HVAC Horizon treats it as your change. It sends no commands for 30 minutes. Each new change starts a new 30 minutes.

Turning the AC off at the unit, with its remote or on its own entity starts the same 30 minutes.

This applies to the first AC, once it has reported the mode HVAC Horizon set. After the 30 minutes the AC starts again when the room still needs it. An AC that you turn on again in the same mode within the 30 minutes keeps running, and HVAC Horizon takes that mode as the one it set. After any other mode change at the unit, HVAC Horizon sets its own mode again and raises `HVAC Horizon mode does not match`.

Four commands still go out in that time: a stall stop, every heater turn-off, a window close, and off from the room thermostat.

To change the room without the 30 minute pause, use the room thermostat.

06.8

Off and restarts

- Off from the room thermostat goes to every device in the same call.
- A room that is `off` when Home Assistant starts sends no start command on any tick.
- No device starts before the first valid room and outdoor readings arrive.
- When an entry is removed or control is turned off, HVAC Horizon turns off the outputs it commanded.

06.9

Repair issues

Each issue clears when its cause ends. None has a fix flow.

ISSUE	RAISED WHEN	CHECK
`HVAC Horizon sensors not updating`	A room or outdoor sensor is stalled.	The sensors listed in the issue.
`HVAC Horizon heater is not warming the room`	The heater ran 60 minutes and the room rose less than 0.5 K.	The heater, its plug and its switch.
`HVAC Horizon AC is not running its compressor`	The AC has been in `cool` or `dry` for 20 minutes with demand and its power stayed under 200 W for more than half of them.	The unit's error code, and the water tank of a portable unit.
`HVAC Horizon mode does not match`	The AC mode differs from the last mode written and the setpoint did not change.	The AC unit, and any other automation that sets its mode.

06.10

Send a bug report

Open the entry, select the three-dot menu and **Download diagnostics**. The file holds the settings, the room model and the last result, with entity IDs redacted. Attach it to an issue at github.com/belikh/hvac-horizon.

06.11

A device does not follow a command

The Unconfirmed commands sensor lists each device whose state differs from the last command on two ticks in a row, as `switch.heater: expected on, is off`. A command that the device refused is listed on the next tick, as `climate.ac: climate.set_hvac_mode failed:` and the reason. The log has one warning for each device.

A device that fails does not hold up the others. The tick goes on to the next device, and the safety

stop turns every other output off.

CAUSE	FIX
The device is offline or its integration lost the connection.	Restore the device. Each tick sends the command again.
Someone changed the device by hand.	Nothing. The entry is listed until the states agree.
A relay is stuck.	Replace it. HVAC Horizon reports it and cannot open it.

06.12

The watchdog turned the outputs off

A timer separate from the control tick checks every minute when the last tick finished. After 16 minutes with no finished tick it runs one tick itself. When that tick fails too, the next check turns off the heater, the AC, the second AC, the fleet devices and every switch the entry turned on. The log has `No control tick has finished for N s`. The Watchdog trips sensor counts these. Control resumes at the next tick that finishes.

A trip means every tick raised an error for 16 minutes, or one tick never returned. Download the diagnostics and the log and open an issue.

SEE ALSO

- Read the forecast: the forecast status and the room model.
- Use the room thermostat: modes and the restart delay.

SECTION

Concepts

How it works and why it is built this way.

07

How it works

The room model, the forecast, the control decision of each tick and the safety rules that sit above it.

07.1

One entry, one room

An entry models one room. Every 5 minutes it runs one tick: read the sensors, store a reading, refit the model when due, forecast, plan the targets and drive the devices. The tick is `HorizonCoordinator._async_update_data()`. See Architecture.



One tick runs every 5 minutes. Step 7 sends commands only with control on.

07.2

Room model

One equation describes the room:

ROOM MODEL

```
dT/dt = k (T_out - T) - c P + g
```

SYMBOL	UNIT	MEANING
`T`, `T_out`	°C	Room temperature (median of the room sensors) and outdoor temperature
`P`	kW	Electrical power of the AC. Below 0.2 kW counts as off.
`k`	1/h	Heat gain toward outdoors
`c`	K/h per kW	Cooling effect of the AC
`g`	K/h	Constant gain from people and appliances

`fit()` in `model.py` solves `k`, `c` and `g` by least squares over consecutive readings.

- It drops outliers: residuals beyond 3.5 median absolute deviations, at most 10 % of the readings.
- It rejects a fit whose normal matrix has a condition number above 1000.
- It returns a standard error for each value. The error of `c` gives the low and high minutes and the confidence label.
- `k` and `c` stay at or above zero. When the plain fit gives a negative `k` or `c`, the fit tries `k` at zero, `c` at zero and both at zero, and uses the one with the smallest squared error.

Readings where the AC heats, dries or runs its fan only are left out. Intervals with the heater on are left out of the passive drift. A reading is not stored while the outdoor sensor is stalled.

A room with no AC has no `c`. `learn_passive_rates()` in `tpi.py` fits `k` and `g` from the intervals with the heater off, and the heating model below adds the heater's rate.

The fit uses the readings of the last 14 days when they alone give a fit, and all 30 days otherwise. The `fit_window_days` sensor shows which.

07.3 Models kept through an idle season

The store holds 30 days of readings. An AC or heater that stays off for longer leaves no reading to fit its model from. Each refit therefore stores the last good value of six models: the room fit, the room rates, the heater rate, the AC heating rate, the drying rate and the corrections. Each is stored with the time of the newest reading behind it. When a refit finds too few readings for a model, the stored value serves. The forecast, the price plan and preheat keep running on it.

The `kept_models` sensor lists each model in use that comes from the store, with its age in days. A new fit replaces the kept value as soon as the readings give one: 3 hours of AC on and off time over 3 runs for cooling, 12 intervals for heating or drying. A kept model describes the room of the season it was learned in, so expect a larger forecast error until the new fit arrives. `reset_learning` clears the kept models.

07.4 Corrections

`corrections.py` fits three terms on what the room model leaves unexplained.

TEXT

```
g_eff = g + s S
c_eff = c + c_out (T_out - 30) + c_rh (RH - 50)
```

TERM	NEEDS	MEANING
`s`	`window_area_m2`	Extra gain per kW of solar gain S on the window
`c_out`	nothing extra	Change of `c` per K of outdoor temperature above 30 °C
`c_rh`	`room_humidity`	Change of `c` per percent of room humidity above 50 %

A term is kept when it has the sign physics gives it (`s` above zero, `c_out` and `c_rh` below zero) and lowers the error on the newest 30 % of the readings by 3 %. A term needs 144 readings that carry it. The cooling forecast applies the kept terms at each minute: the solar gain follows the clear-sky model scaled to the current reading. The `fit_corrections` sensor lists the kept terms.

07.5

Heating and drying

`mode_models.py` holds one model for each.

TEXT

```
heating:  dT/dt = k (T_out - T) + g + h
drying:   dRH/dt = -r
```

VALUE	SOURCE
`h` for a heater switch	The learned rise per hour at full output, from intervals with the heater on and the AC off
`h` for the AC in heat mode	`c_heat` `P`: the rise per kW from heat-mode intervals, times the mean power of those intervals
`r`	The mean fall of room humidity per hour over dry-mode intervals

Each needs 12 intervals. The forecast sensors show the heating forecast while the mode heats and the room is below the heat target, and the drying forecast in dry mode. See Heating and drying.

07.6

Forecast

`forecast()` in `forecast.py` steps the model forward one minute at a time with the exact exponential solution of the equation. It returns minutes, energy and cost, or `unreachable` when the room does not reach the target in 12 hours.

- The AC runs at its learned running power for the whole forecast.
- The outdoor temperature follows the hourly weather forecast when a weather entity is set. `outdoor.py` builds the series: it starts at the sensor reading, and the offset to the forecast fades over 3 hours. A learned bias for each hour of the day is added once 6 sensor and forecast pairs exist.
- The cost uses the price slots of the price sensor. A minute with no slot has no cost.

07.7

Targets

`_plan_control()` sets three values on each tick.

VALUE	MEANING
Cool threshold	The AC cools when the room is above it.
Heat target	The heater aims at it.
Heat threshold	The heat target minus 0.5 K. The PID heater and AC heat start below it.

The preset, your target, the price shift, occupancy, vacation and the mould bump set the cool threshold. The heat target follows the preset and never exceeds the cool threshold.

07.8

Cooling decision

`decide()` in `decide.py` returns one action for the AC. Its order is:

1. A window open longer than its delay: off. In the automatic mode, with a window actuator or a whole-house fan, outdoors below 16 °C: off, until free cooling has failed for 30 minutes. `cool` mode has no outdoor limit. In the automatic mode the AC also waits 60 minutes after the heater ran, unless the room is 2 K above the cool threshold.
2. A night pre-cool, when `night_plan_act` is on.
3. An unoccupied room: pre-cool when a schedule event is near, else cool only above the setback.
4. An occupied room: cool above the threshold.

A running AC keeps cooling until the room is at the threshold. An AC that is off starts once the room is 0.5 K above it.

The coordinator then applies the rules that sit above `decide()`: the room thermostat's mode (in `fan_only` the AC runs its fan and never cools), off from the thermostat, free cooling, the night hold and the cooling plan. Free cooling comes first: while the entry holds the window open or runs the whole-house fan, the AC stays off.

Setpoint regulation

Many ACs regulate on their own return-air sensor, which can read cooler than the room. HVAC Horizon lowers the setpoint it sends by the gap between the room median and the AC's `current_temperature`. An AC that reports no temperature is regulated from power: after 10 minutes in cool mode under 0.2 kW with the room above the threshold, the setpoint moves down one step per tick.

07.9

Heating

A heater switch runs as a share of a 10 minute window. `pid.py` or `tpi.py` gives the share.

- PID stops 0.5 K below the heat target.
- TPI heats up to the heat target, and up to its thresholds above the target when both are set.

The heater stays off in each of these cases:

- The thermostat's mode does not heat.
- Outdoors is above 22 °C and the mode is automatic or `dry`. `heat` mode has no outdoor limit.
- A window is open, the house is away or the lock is on.

- A sensor is stalled.
- The AC cools.

07.10

Several devices in one room

With `fleet` set, `choose_devices()` in `fleet.py` picks the devices for the gap to the target. `fast` orders them by effect, `economy` by cost per kelvin. Each device has its own rate in K per hour per kW. The rate you enter is replaced by a fitted rate once the device has run alone for 6 readings. A device with no power sensor counts its rated watts while it runs.

07.11

Safety rules

These hold on every tick, above every other decision.

- Off from the room thermostat goes to every device at once. A stop is never delayed.
- No start command goes out before the first valid room and outdoor readings.
- A room held off sends no start, including after a restart.
- A start after a stop waits for the restart delay. The delay counts from the device's own last change.
- A stalled sensor turns the AC off and sets the heater to 0.
- Unloading the entry turns off every output the entry commanded.

07.12

Storage

The entry stores its readings, the room model, the kept models, the thermostat's mode, targets and humidity target, the lock, the house mode and the commanded device states with Home Assistant's `Store`, version 4. `_HorizonStore` in `coordinator_common.py` migrates older versions.

07.13

Terms

action
What Home Assistant called a service before 2024.8, such as hvac_horizon.set_lock. Call one from Developer tools > Actions or from an automation.

area
A Home Assistant area, such as Bedroom. A room entry can belong to one: setup suggests the area's entities, and the room's thermostat and sensors are placed in it.

automatic mode
The thermostat mode in which the room runs whatever it needs. Home Assistant names it heat_cool in a room that can heat and cool, and auto in a room that only cools.

cool threshold
The room temperature above which the AC cools. It comes from the preset, your target, the price and occupancy.

heat target
The room temperature the heater aims at.

heat threshold
The heat target minus 0.5 K. A PID heater and AC heat start below it.

observe only
The state of an entry with control off. It learns and forecasts and sends no commands.

preset
One of comfort, eco, boost and away. Each has its own cool threshold and heat target.

restart delay
The time a device stays off before it may start again, 3 minutes by default.

room model
The three learned values k, c and g that describe how the room gains heat and how the AC cools it.

tick
One run of the control loop. It happens every 5 minutes and after each change on the room thermostat.

SEE ALSO

- Limits of the model: what the model assumes and leaves out.
- Architecture: the modules and the order of the tick.

08

Limits of the model

What the room model assumes, what it leaves out, and which parts of control are untested on hardware.

08.1

What the model assumes

The forecast and the fit use one equation for each room:

TEXT

```
dT/dt = k (T_out - T) - c P + g
```

- One thermal node per room. The air, walls, furniture and floor share one temperature. A room that cools fast at first and slowly later fits this model less well.
- Power is a proxy for cooling delivered. `P` is the electrical input in kW, and `c` is the effective room cooling per electrical kW, in K/h. The fit folds the unit's efficiency and the airflow into `c`. A correction moves `c` with the outdoor temperature and the room humidity when the readings support it.
- `k` and `g` are constants. `k` is the heat gain toward outdoors, and `g` is the steady internal gain from people and appliances.
- Rooms with several room sensors use their median. A sensor that stalls drops out of the median.

08.2

What the model does not include

- No humidity node. Room humidity enters only as a correction to `c`, and only with a room humidity sensor. AC dry intervals and AC heat intervals are kept out of the cooling fit and have their own models.
- No solar node. Solar gain enters as a correction to `g`, and only with a window area set. It has no lag: sun that heats a wall for hours is not modelled.
- A correction is kept only when it lowers the error on the newest 30 % of the readings by 3 %. A room with under 12 hours of readings that carry a term has no correction for it.
- The heating model has one rate per device, constant at every outdoor temperature. A heat

pump whose output falls on a cold day heats slower than the forecast.
- The drying model is one rate in percent per hour. It does not depend on the humidity or the temperature.
- No door or window opening, no occupancy change and no other heat source inside the fit. These show as noise, and the outlier filter drops the worst of them.
- No second room. Each entry models one room. Two rooms that exchange heat are two separate fits.

08.3 What the forecast assumes

`forecast()` steps the model forward in 1-minute steps.

- The AC runs at a constant power for the whole forecast. A unit that cycles uses less energy, so the energy figure is an upper bound.
- Without a weather entity, the outdoor temperature stays at its current value. A room that is cooling toward evening is forecast as if the outdoor temperature did not fall.
- With a weather entity, the outdoor temperature follows the hourly forecast, corrected by the learned difference between the outdoor sensor and the forecast. The forecast is then as good as the weather forecast for the site.
- The price slots set the cost. A minute with no slot has no cost.
- The forecast stops after 12 hours and reports `not reached within 12 hours` when the room settles above the target.

08.4 What this means for a reading

- The forecast is most reliable on a day like the days in the stored samples. A heatwave or a cold snap lies outside the fitted range.
- The range between `minutes_low` and `minutes_high` comes from the standard error of `c`. It does not include the errors listed above. The `range_coverage_pct` attribute of the Forecast error sensor shows how often past cool-downs fell inside it.
- The fit needs at least 3 hours of AC on and off time over 3 runs. Until then the status reads `learning`.

08.5 Known limits of control

- Setpoint regulation covers the first AC. A second AC and fleet climates get the plain target.
- A PID heater settles up to 0.5 K below the heat target, because it stops one deadband below it.
- In `dry` and `fan_only` the room thermostat shows one temperature, the cool target.
- The thermostat's fan mode, swing mode and `fan_only` mode are those of the first AC. A second AC and fleet devices have none.
- The humidity target sets the level the room dries above. The humidifier limit follows it and has no control of its own.
- Home Assistant's documentation says no temperature is set in `auto`. A room that only cools still takes one target in `auto`, because its automatic mode adds free cooling to what `cool` does. A room that heats and cools uses `heat_cool`, as documented.
- The Heatwave sensor is information only. It changes no device command.
- Two profiles of the simulation suite fail by design and are marked as expected failures: an 8 kW AC overshoots a small room, and the small unit's fitted rate collapses in `economy` mode with two ACs.
- The watchdog and the tick run in the same Home Assistant process. A process that hangs or loses power runs neither. A device that must fail off needs its own thermostat or timer.
- `commands_unconfirmed` reports a device that did not follow a command. It does not act on it, apart from the command that each tick sends again.

08.6

What has run on real devices

DEVICE OR FUNCTION	UNIT TESTS	SIMULATED ROOM	REAL DEVICE
AC cooling through a climate entity, with a power sensor	yes	yes	one installation
AC with rated power and no power sensor	yes	yes	no
Room with a heater and no AC	yes	no	no
Heater switch, PID and TPI	yes	yes	no
AC heat mode, dry mode and fan mode	yes	yes	no
A device that is offline, ignores commands or raises	yes	yes	no
Second AC and fleet devices	yes	yes	no
Valve and TRV offset	yes	yes	no
Boiler, fan, humidifier, dehumidifier, cover	yes	no	no
Window actuator and whole-house fan	yes	yes	no
Relay test	yes	yes	no
Load shedding	yes	no	no

A row with no under Real device is untested on hardware. Start it in observe-only and watch the first runs.

SEE ALSO

- How it works: the model, the forecast and the control decision.
- Sensors: the fit attributes and the forecast range sensors.

SECTION

Develop

Architecture, tests and release.

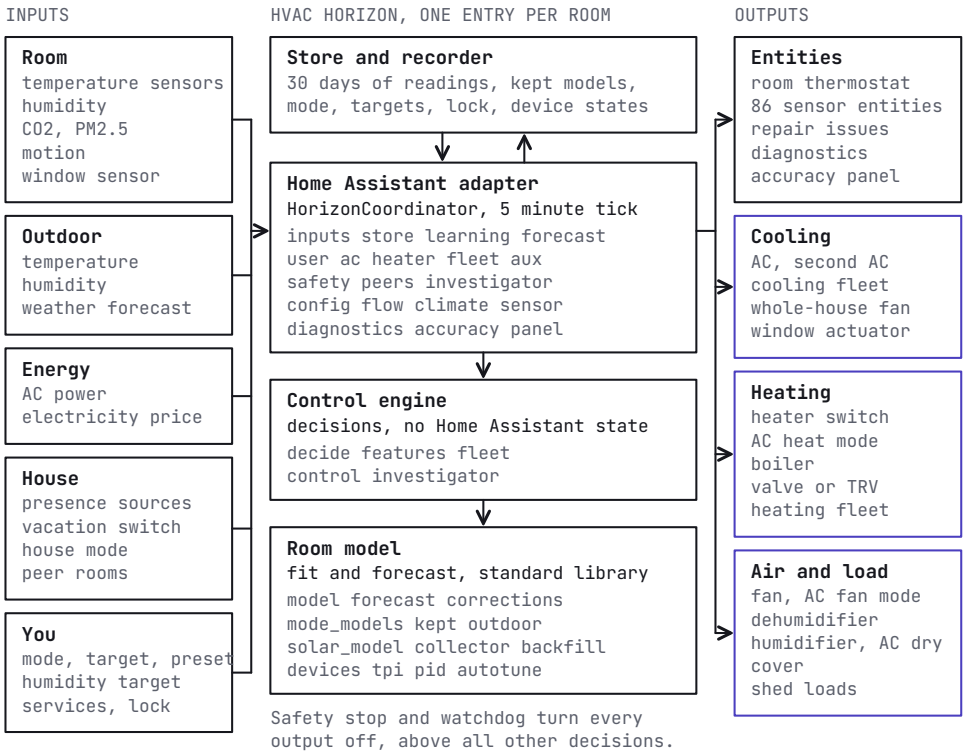
09

Architecture

The modules of HVAC Horizon, the order of one control tick, and where to add a setting or a published value.

09.1

Layout



The code is in `custom_components/hvac_horizon/` and has three layers.

LAYER	WHERE	MODULES	MAY IMPORT
Room model	`room_model/`	`model.py`, `forecast.py`, `corrections.py`, `mode_models.py`, `kept.py`, `outdoor.py`, `solar_model.py`, `collector.py`, `backfill.py`, `devices.py`, `tpi.py`, `pid.py`, `autotune.py`	The standard library and `room_model`
Control engine	`engine/`	`decide.py`, `features.py`, `fleet.py`, `control.py`, `investigator.py`	`room_model`, `engine` and `const.py`
Home Assistant adapter	the package root	`coordinator.py` and the `coordinator_*.py` mixins, `__init__.py`, `config_flow.py`, `climate.py`, `sensor.py`, `binary_sensor.py`, `diagnostics.py`, `accuracy_api.py`	Anything

The first two layers hold no Home Assistant state: functions take values and return values. `tests/test_layers.py` reads the imports of every module in `room_model/` and `engine/` and fails on one outside its column.

Both packages sit inside the integration folder, because HACS installs that folder only. Python runs the integration's `__init__.py` before a module below it, so importing `room_model` on its own needs the package copied out of the folder.

09.2 Pure modules

MODULE	JOB
`collector.py`	`Entry`, one reading. Sample history, running power, the cycle check and CSV export.
`model.py`	`fit()` learns `k`, `c` and `g` with standard errors. `fit_confidence()` labels the fit.
`forecast.py`	`forecast()` gives minutes, energy and cost to a target. `price_at()` reads the price slots.
`corrections.py`	`learn_corrections()` fits the sun, outdoor temperature and humidity terms. `apply()` gives the fit at given conditions.

MODULE	JOB
`mode_models.py`	The heating and drying models and their forecasts.
`kept.py`	The last good value of each learned model, to and from the store.
`outdoor.py`	The outdoor temperature series from the hourly forecast, the night low and the sensor-to-forecast bias.
`decide.py`	`decide()` returns one AC action. `cool_above_c()` gives the cool threshold.
`features.py`	Decision functions: presets, stall rule, inhibit, lock, restart delay, PWM, price shift, valve, shed, humidity, free cooling.
`fleet.py`	`choose_devices()` picks the heaters and coolers for a gap. `calibrate_rates()` fits each device's rate.
`tpi.py`	`learn_room_rates()`, the online gains and `tpi_on_percent()`.
`pid.py`	`Pid`, the heater PID controller.
`control.py`	`precool_start_in_min()` picks the cheapest start that meets a deadline.
`devices.py`	The room model with one coefficient per device, for two ACs.
`backfill.py`	`backtest()` replays past cool-downs.
`autotune.py`	`RelayTuner`, the relay test and its PID gains.
`solar_model.py`	Sun position and irradiance on a vertical window.
`investigator.py`	`investigate()` gives the rule-based answer. `CallBudget` limits model calls per day.

09.3

Coordinator

`HorizonCoordinator` in `coordinator.py` is a `DataUpdateCoordinator` with a 300 second interval. It holds `__init__`, the tick and `_plan_control`. It inherits one mixin per area.

MODULE	MIXIN	JOB
`coordinator_inputs.py`	`InputsMixin`	Reads the room and optional sensors. Stall detection, occupancy, window advice.
`coordinator_store.py`	`StoreMixin`	The stored payload, the commanded-state record, recorder backfill.
`coordinator_learning.py`	`LearningMixin`	Model and fleet refits, online gains, the gain suggestion.
`coordinator_forecast.py`	`ForecastMixin`	Weather, solar and price inputs, the forecast, pre-cool and preheat timing.

MODULE	MIXIN	JOB
`coordinator_user.py`	`UserMixin`	The thermostat's mode, targets, humidity target and presets, the lock, what-if and the relay test.
`coordinator_ac.py`	`AcMixin`	`_async_control`: the AC action, setpoint and mode writes, AC heat, dry mode, the second AC.
`coordinator_heater.py`	`HeaterMixin`	Heater gates, PWM windows, the heater switch and output limits.
`coordinator_fleet.py`	`FleetMixin`	Fleet choice and driving.
`coordinator_aux.py`	`AuxMixin`	Valve, load shedding, window actuator, free cooling, cover and fan mode.
`coordinator_safety.py`	`SafetyMixin`	The safety stop, the locked stop, the switch driver, `_send()` for every device command, the watchdog and the unconfirmed-command check.
`coordinator_peers.py`	`PeersMixin`	Requests between entries of one house.
`coordinator_investigator.py`	`InvestigatorMixin`	The mode-change explanation and the on-demand report.
`coordinator_common.py`	none	Helpers, constants, the logger, `_HorizonStore` and `room_device_info()`, the device of the entry with its suggested area.

Each mixin lists the instance attributes it reads as class-level annotations. A test that patches a module-level function patches the module that calls it, for example `coordinator_learning` for `fit`.

09.4

Home Assistant surface

MODULE	JOB
`__init__.py`	`async_setup` registers the actions, so they exist before an entry loads. `async_setup_entry` sets up the coordinator and the platforms and runs the safety stop on unload. A refused value in an action raises `ServiceValidationError`.

MODULE	JOB
`config_flow.py`	Three flows on shared section steps (`_SectionSteps`): setup, Reconfigure (the Room form, stored in the entry's data) and `HorizonOptionsFlow` behind Configure (every other section, stored in the options). `split_settings()` divides the two, and the coordinator reads both through `settings`. `_schema()` lists every key, selector and default. `SECTIONS` assigns each key to a form section. The two list settings, `schedule_blocks` and `fleet`, have their own steps with `_schedule_schema()` and `_fleet_schema()`.
`const.py`	Setting keys, defaults and limits.
`climate.py`	`HorizonClimate`, the room thermostat. It sets the mode, target and preset on the coordinator and never drives a device.
`sensor.py`	`STATUS_KEYS`, `MEASURES` (device class and unit) and the sensor entities.
`binary_sensor.py`	`BINARY_KEYS` and the binary sensor entities, for the values that are on or off.
`translations/en.json`	Form texts, entity names, action names and repair issues. `icons.json` holds the icons of entities and actions.
`diagnostics.py`	The diagnostics download, with entity IDs redacted.
`accuracy_api.py`	The JSON views and the sidebar panel. The panel is the web component in `frontend/accuracy-panel.js`, which calls the views through `hass.callApi`.

09.5

One tick

`_async_update_data()` runs these steps in order.

1. **Weather**. `_refresh_weather()` fetches the daily and hourly forecast, at most every 30 minutes.
2. **Read and store**. `_read_inputs(now)` builds an `Entry`. The tick appends it and caps the history at `MAX_ENTRIES`.
3. **Refit**. Every 12 readings, `_refit()` runs `learn_room_rates()`, `fit()` and `backtest()`.
4. **Forecast**. `_forecast(now)` sets `minutes`, `energy_kwh`, `cost`, `status` and the fit attributes.
5. **Context**. The tick sets the day type, night plan, occupancy, solar gain, window advice, the economizer phase, the heat source and the motion preset.
6. **Window actuator**. `_drive_window_actuator(result)` opens or closes the actuator.
7. **Targets**. `_plan_control(now, result)` sets the preset, the cool threshold, the heat target and the heat threshold.
8. **Plans**. `mpc_plan()` and `mpc_heat_plan()` run when a fit and price slots exist.
9. **Heater output**. PID or TPI gives the output. `_heater_gate_closed(result)` sets the gate.
10. **Cover and load shedding**. `_drive_cover()` and `_drive_shed()`.
11. **AC control**. `_async_control(now, result)` runs with control on. It handles off from the thermostat, the lock, the manual-change pause, the stall, free cooling and the mode, then calls `decide()` and writes the AC. It also drives the fan, humidifier, dehumidifier, boiler and

whole-house fan.

12. **Heater switch.** `_async_heater_step(now, result)` runs in a `finally` block, so it runs when the AC step raises.
13. **Investigation and valve.** `_publish_investigation()` and `_drive_valve()`.
14. **Publish.** The result dict becomes `coordinator.data`. The sensors and the thermostat read it.

The room thermostat calls `async_refresh()` after each change, so a new mode or target takes effect at once.

09.6

Watchdog

`start_watchdog()` in `coordinator_safety.py` starts a 60-second timer at setup. It reads `_last_tick_done_s`, which the tick sets when it returns without an error. When no tick has finished for 16 minutes, the first check runs one tick itself, unless a tick is running. After a clock jump or a suspend that tick finishes and nothing trips. When it fails or a tick hangs, the next check turns every output off.

09.7

Free cooling states

`free_cooling_step()` in `features.py` is the only code that changes the free cooling phase.

FROM	TO	WHEN
`idle`	`running`	The window this entry opened is open, or the whole-house fan runs.
`running`	`idle`	Neither is the case.
`running`	`gave_up`	The room is 2 K above the cool threshold after 30 minutes, or above it and fell by less than 0.3 K in 30 minutes.
`gave_up`	`idle` or `running`	The room is at or below the cool threshold.

While the phase is `gave_up` the window closes, the fan stops and the AC cools.

09.8

Add a setting

1. Add the key and its default to `const.py`.
2. Add it to `_schema()` and to one section of `SECTIONS` in `config_flow.py`. Add its label and its help text to `translations/en.json`, under `config.step` and with the same text under `options.step`. A test fails when the two differ.
3. Read it in the coordinator through `self.settings`, which holds the entry's data and options, and make it change a device command or a published value.
4. Add a row to `SETTINGS` in `tests/ha/test_settings_effect.py` with the level you expect: `control` or `published`.
5. Add it to Settings.

`test_every_setting_in_const_has_an_audit_row` fails when step 4 is missing. The row's test fails when the setting changes nothing. See Guard tests.

09.9

Add a published value

Write the key to the result dict in the mixin that computes it. Add it to `STATUS_KEYS` in `sensor.py`, or to `BINARY_KEYS` in `binary_sensor.py` when it is on or off, with its name under `entity` in `translations/en.json`. Or return it from `extra_state_attributes`. `tests/test_published_keys.py` fails when a key has neither.

SEE ALSO

- Test and release: the tests and the release steps.
- How it works: the room model and the control decision.

10

Test framework

The fake devices, the simulated house and the closed-loop harness that the tests run the integration against, and how to add a case.

10.1

Four levels

The suite tests the integration at four levels. Each level catches what the one before it cannot.

LEVEL	WHERE	WHAT RUNS	WHAT IT PROVES
Pure	`tests/test_*.py`	One function of `room_model/` or `engine/`	The model and each decision, on exact numbers and on random inputs
Entry	`tests/ha/`	One entry in a test Home Assistant	Setup, the forms, the thermostat, actions, restarts and unload
Basics	`tests/ha/test_basics.py`	The thermostat against a room that warms and cools	The rules every thermostat user expects, read from the outside
Closed loop	`tests/sim/`	12 to 24 simulated hours of a house the integration does not know	Comfort, cycling, safety stops and device faults over a whole day

10.2

Fake devices

`tests/fakes.py` holds `FakeAc` and `FakeSwitch`. Each is a real `ClimateEntity` or `SwitchEntity`, added to Home Assistant's own `climate` and `switch` integrations with `add_devices()`. A command the integration sends goes through Home Assistant's actions, as it does to a real device.

A COOL-ONLY AC AND A HEATER PLUG

```
ac = FakeAc("climate.bedroom", modes=("off", "cool"), min_c=17, max_c=30,
fan_modes=("auto", "high"))
```

```
plug = FakeSwitch("switch.heater")
await add_devices(hass, ac, plug)
```

A REAL DEVICE DOES THIS	THE FAKE DOES IT TOO
Refuses a mode it does not list	Home Assistant raises `ServiceValidationError`
Refuses a setpoint outside `min_temp` to `max_temp`	The same error
Reports `hvac_modes`, `target_temp_step`, `current_temperature`, `hvac_action`, `fan_modes`	Every attribute comes from the entity
Changes `last_changed` on a new mode and keeps it on a new setpoint	Home Assistant's state machine sets it
Is changed with its remote control	`ac.press_remote(mode="cool")`, `plug.flip_by_hand(True)`

Each fake records the commands it received in `calls`. Three flags make it fail as real devices fail:

FLAG	THE DEVICE	STANDS FOR
`available_now = False`	Reads `unavailable`	A device that is offline
`ignore_commands = True`	Accepts a command and does not change	A lost radio packet, a stuck relay
`fail_commands = True`	Raises `HomeAssistantError` on a command	An integration that lost its connection

`tests/ha/test_fakes.py` holds the fakes to this table. A test built on a fake that drifts from Home Assistant would prove nothing.

10.3 Simulated house

`tests/sim/house.py` is a room in plain Python with no Home Assistant import. It is richer than the model the integration fits, so the integration never sees its own equation back.

PART	MODEL
Room	Two thermal nodes, air and mass. The integration fits one.
Weather	A daily sine with slow noise, solar gain from 06:00 to 18:00, an outdoor humidity
AC	The modes `off`, `cool`, `heat`, `dry` and `fan_only`. Capacity and COP fall on a hot day, heating capacity falls on a cold one.
AC thermostat	Starts 0.5 K past the setpoint and stops 1 K on the other side, on the unit's own sensor
AC sensor	Can read a fixed bias away from the room air, as a return-air sensor does

PART	MODEL
Compressor	A minimum run time, a restart lockout and a stop on a change of mode
Humidity	One node: it moves toward outdoors, rises with people and falls while a compressor cools or dries
Heater	A resistive load on a switch
Sensors	Noise, and two faults: `dropout` (unavailable) and `stuck` (no new report)

All randomness comes from one seeded generator. A run repeats exactly.

10.4 Closed-loop harness

`Harness` in `tests/sim/harness.py` wraps each simulated device in a fake device, installs a real entry and runs the clock. Each minute it steps the house and fires due timers. Every 5 minutes it writes the sensors and runs one tick. The result is a `Trace`:

FIELD	HOLDS
`rows`	One row per minute: air, mass and outdoor temperature, humidity, each AC's mode and compressor, the heater, the power
`calls`	Every command a device received, with its minute
`ticks`	Each tick's minute, whether it finished, and its action text
`command_failures`	Every command a device or Home Assistant refused
`unconfirmed`	The Unconfirmed commands list at each tick where it was not empty

A `Profile` describes one run: the envelope, the weather, the ACs and their modes, a heater, the thermostat mode and extra settings. `SensorFault` and `DeviceFault` schedule failures by minute.

A HEAT PUMP ON A COLD NIGHT

```
profile = Profile("heat_pump", TIMBER, COOL_NIGHT, ((AC, NORMAL_AC),),
start_hour=18.0, mode="heat", settings=(("ac_heat", True),))
trace = await Harness(hass, freezer, profile).run(12)
assert all(ok for _, ok, _ in trace.ticks) and trace.command_failures == []
```

10.5 What the closed loop asserts

`tests/sim/test_closed_loop.py` runs nine cooling profiles for 24 hours each, and each again with the room sensors failing for an hour.

- No tick fails and no device refuses a command.
- An AC starts at most 6 times in an hour and only after its restart delay.
- The heater and an AC never run in the same minute.

- After 3 hours the room is within 1.5 K of the target for 90 % of the minutes.
- With every room sensor dead, each device is off within 20 minutes and stays off.

`tests/sim/test_device_modes.py` runs the device types that have not run on real hardware:

RUN	ASSERTS
Heat pump, cold night	The AC heats, never cools, and holds the room within 1.5 K
Dry mode, humid room	The AC dries to 3 points below the humidity target and never cools
Fan mode, hot day	The AC stays in `fan_only` for 12 hours and no compressor starts
AC with rated power	The comfort band holds from `hvac_action` alone
AC whose sensor reads 2 K low	The room still reaches the target
Cool-only AC, cold humid night	With AC heat and AC dry enabled, the AC receives only `off` and `cool`
AC unavailable, ignoring or raising for an hour	Every tick finishes, the fault is reported, and the room recovers

Two of these runs found defects when they were written. A cool-only AC was sent `heat`. An AC that raised on a command ended every tick, and the watchdog's stop then raised at the same device before it reached the other outputs.

10.6 Add a case

1. A fault a user reports about basic behaviour: add a test to `tests/ha/test_basics.py` first, then fix it.
2. A device type or a mode: add a run to `tests/sim/test_device_modes.py` with a `Profile`.
3. A way a device fails: add a flag to the fake, a test of the flag to `tests/ha/test_fakes.py`, and a `DeviceFault` kind.
4. A new behaviour of a real device that the fake lacks: change the fake and its test in the same commit.

`tests/ha/test_setup.py` and `tests/ha/test_settings_effect.py` predate the fake devices. Most of their tests set a device's state directly and register their own action handlers. Use the fake devices for new tests in those files.

SEE ALSO

- Test and release: run the suite, the guard tests and the release steps.
- What has run on real devices: which rows the simulation covers.

11 Test and release

Set up the test environment, run the suite, read the guard tests, and cut a release that HACS offers to users.

11.1

Set up the test environment

The tests run on Python 3.14 against Home Assistant 2026.11.2.

CREATE THE ENVIRONMENT

```
python3 -m venv .venv
.venv/bin/pip install -r requirements-test.txt
```

`requirements-test.txt` pins `pytest-homeassistant` 2026.11.2, `pytest` 9.0.3 and `pytest-asyncio` 1.4.0. `pytest-homeassistant` is Home Assistant's own test package. Its release number is the Home Assistant version it tests.

11.2

Run the tests

RUN THE FULL SUITE

```
PYTHON=.venv/bin/python scripts/check.sh
```

The script runs `python -m pytest -q` and exits non-zero on any failure. The full suite takes about 11 minutes: the settings audit takes 6 of them.

RUN EVERYTHING EXCEPT THE SETTINGS AUDIT

```
.venv/bin/python -m pytest -q --ignore=tests/ha/test_settings_effect.py
```

RUN ONE FILE

```
.venv/bin/python -m pytest tests/test_model.py -q
```

A passing run ends with a line like this:

OUTPUT

```
1549 passed, 2 xfailed
```

The 2 expected failures are simulation profiles. See Known limits of control.

Read the warnings as well as the failures. Fix a deprecation warning in the same change that shows it.

11.3

Test layout

PATH	CHECKS
`tests/test_*.py`	The pure modules, such as `decide.py`, `features.py`, `model.py` and `fleet.py`. `tests/test_model.py` checks the solver on 500 random systems and the constrained fit on 300 random data sets against a search of every allowed fit.
`tests/ha/test_config_flow.py`	The setup, Reconfigure and Configure forms and their texts.

PATH	CHECKS
`tests/ha/test_setup.py`	Setup, control, the room thermostat, actions, restarts and unload, against fake devices.
`tests/ha/test_basics.py`	The basics of a thermostat, from the outside: mode, target, off, restart and dead sensors against a simulated room.
`tests/ha/test_settings_effect.py`	Every setting changes a device command or a published value.
`tests/fakes.py`	The fake AC and switch, real Home Assistant entities. `tests/ha/test_fakes.py` tests them. See Test framework.
`tests/ha/test_device_faults.py`	A device that raises does not end the tick or the safety stop.
`tests/sim/`	The integration in a closed loop with a simulated room: 24 simulated hours for each of 9 cooling profiles, and the device modes and device faults. Marker `sim`.
`tests/test_no_dead_code.py`	Every public function has a caller.
`tests/test_layers.py`	A module in `room_model/` or `engine/` imports only its layer.
`tests/test_published_keys.py`	Every key the coordinator publishes is on a sensor or read back.

`pytest.ini` sets `asyncio_mode = auto` and `addopts = -p no:respx`. Home Assistant installs its own httpx alias before `respx` may load.

11.4 Guard tests

Four groups of tests exist to stop work that looks done and does nothing.

The thermostat contract. `test_room_mode_decides_which_devices_may_run` in `tests/ha/test_setup.py` runs each mode against a hot room and a cold room. It asserts which devices receive a start command. It was added after the AC cooled a room in `heat` mode: the mode was stored and shown, and control never read it.

The basics. `tests/ha/test_basics.py` uses the room thermostat as a person does. Each test sets a mode and a target, lets a simulated room run minute by minute and reads the heater switch, the AC and the thermostat state. No test reads the coordinator. The rules it holds:

- `heat` warms a room below the target and `cool` cools a room above it, at outdoor temperatures from 5 to 35 °C.
- The room reaches the target within 4 hours and stays within 1 K of it.
- A new target starts or ends a run. The thermostat shows the target you set.
- `off` and a mode change end a run in the same call.
- The heater and the AC do not run against each other at any minute, and the automatic mode does not take turns between them over a day.
- With a window actuator, the automatic mode cools with the window on a cold day and runs the AC when the window does not help.
- The mode and the target survive a restart. Dead room sensors turn the devices off.
- Twelve seeded random days hold the same rules: a random mode, start and outdoor temperature, sensors with 0.15 K of noise, one restart and one target change at random minutes. The heater starts at most 36 times and the AC at most 18 times in 6 hours.

Add a case here for every fault a user reports about basic behaviour, before the fix.

The settings audit. `tests/ha/test_settings_effect.py` runs a scripted day twice for each setting: once with a base entry and once with the setting changed. It compares the device commands and the published data of the two runs.

LEVEL IN SETTINGS	THE TEST PASSES WHEN
`control`	The device commands differ.
`published`	The published data differs. The row gives the reason in `why`.
`exempt`	Not run. The row gives the reason in `why`.

`test_the_scripted_day_is_repeatable` runs the same entry twice and asserts equal output. Without it a difference would prove nothing. When a new setting does not show its level, extend `House.day()` with a phase that exercises it.

Dead code and unread keys. `test_no_dead_code.py` fails when a public function has no caller outside its own test file. `test_published_keys.py` fails when a published key is neither on a sensor nor read by the coordinator.

11.5 Build the documentation

The pages are in `docs-src/`. `docs/*.md` is generated from them.

CHECK AND BUILD

```
npx jupiter-docs check
npx jupiter-docs build --md
```

Commit `docs-src/` and the regenerated `docs/*.md` together. `docs/feature-matrix.md` is generated by `tests/ha/test_settings_effect.py` from its audit table. Regenerate it with `HVAC_WRITE_MATRIX=1 python -m pytest tests/ha/test_settings_effect.py -k feature_matrix`. The suite fails when the page is stale.

11.6 Continuous integration

`.github/workflows/ci.yml` runs on each push: the test suite on Python 3.14, `hassfest` and the HACS validation.

11.7 Release

1. Set `version` in `custom_components/hvac_horizon/manifest.json` and in `docs.config.json` to the same value.
2. Rename the `Unreleased` heading in `CHANGELOG.md` to the version and the date.
3. Check `homeassistant` in `hacs.json`. It is the minimum Home Assistant version, and it equals the `pytest-homeassistant` pin.
4. Run `scripts/check.sh`, commit and push to `main`.
5. Wait for CI to pass on the commit.
6. Tag the commit `v<version>`, push the tag and publish a GitHub release with the changelog section as its notes.

TAG AND PUBLISH

```
git tag v0.6.2
git push origin v0.6.2
gh release create v0.6.2 --title "0.6.2" --notes-file notes.md
```

HACS offers the newest release to users. A user installs it from HACS and restarts Home Assistant.

SEE ALSO

- Test framework: the fake devices, the simulated house and the harness.
- Architecture: the modules and the tick that the tests cover.

SECTION

Reference

Every setting, sensor and attribute, in tables with defaults.

12

Settings

Every setting of an HVAC Horizon entry, by form section, with its default and what it changes.

12.1

How settings are stored

Each entry has 89 settings. The seven of the Room step are the entry's data: change them with **Reconfigure**. The other 82 are its options: change them with **Configure**. Two are required: `room_sensors` and `outdoor`. A room also needs `climate` or `heater_switch`, and an AC needs `power` or `ac_rated_w`. See Set up the integration.

The Key column is the name in the config entry and in the diagnostics download. The label is the text on the form.

`tests/ha/test_settings_effect.py` runs each setting at two values over a scripted day. It fails when a setting stops changing the device commands or the published data.

12.2

Room

KEY	FORM LABEL	DEFAULT	EFFECT
`area`	Area	unset	The Home Assistant area of the room. Names a new entry and holds its device. Changes no device command.

KEY	FORM LABEL	DEFAULT	EFFECT
`climate`	Air conditioner	unset	The AC that HVAC Horizon reads and, with control on, drives. Leave it empty for a room with heating only. A room needs this or `heater_switch`.
`power`	AC power sensor (W)	unset	Power of the whole AC unit in watts. Feeds the room model, the energy forecast and the running check: below 0.2 kW the AC counts as off. An AC needs this or `ac_rated_w`.
`ac_rated_w`	AC rated power (W)	unset	For an AC with no power sensor. The AC counts as drawing this while its `hvac_action` is `cooling`, `heating` or `drying`. An AC that reports no action counts by its mode.
`heater_switch`	Heater switch	unset	Switch for the heater plug. The heater output runs in 10 minute windows.
`room_sensors`	Room temperature sensors	required	One or more temperature sensors in the room. The room temperature is their median. A stalled sensor drops out of the median.
`outdoor`	Outdoor temperature sensor	required	Outdoor temperature. Feeds the room model, the forecast, the 16 °C cooling limit of the automatic mode with a window actuator or whole-house fan, and the 22 °C heating limit of the automatic mode and `dry`.

12.3 Comfort and presets

KEY	FORM LABEL	DEFAULT	EFFECT
`schedule`	Calendar or schedule for pre-cooling	unset	Its `next_event` attribute, an ISO time, sets the deadline for pre-cooling. When the room cannot reach the setpoint by the event and the event is within 120 minutes, cooling starts at once.
`vacation`	Vacation switch	unset	While on, cooling holds `vacation_c` and heating holds 16 °C.
`vacation_c`	Vacation temperature (°C)	`27.0`	Cooling target while vacation is on.
`away_delay_min`	Minutes before an empty house counts as away	`30.0`	Delay before an empty house counts as away.
`comfort_c`	Comfort temperature (°C)	`24.0`	The normal target. Range 16 to 30.
`eco_c`	Eco temperature (°C)	unset	Target for the eco preset. An unoccupied house cools to this value when set.
`comfort_energy_weight`	Comfort or savings (0 to 1)	unset	0 keeps the comfort target. 1 uses the eco setpoint. Needs `eco_c`.
`heat_eco_c`	Eco heating temperature (°C)	unset	Heat target for eco.
`heat_boost_c`	Boost heating temperature (°C)	unset	Heat target for boost.
`boost_c`	Boost temperature (°C)	`23.0`	Target for the boost preset.
`away_c`	Away temperature (°C)	`27.0`	Cooling target for an unoccupied house when `eco_c` is not set. Also the away preset target.
`schedule_blocks`	Schedule blocks (its own step)	`[]`	Presets by time of day. See [[guides/services-and-presets#schedule-blocks

12.4 Price

KEY	FORM LABEL	DEFAULT	EFFECT
`price`	Electricity price sensor	unset	Sets the price used for cost. Without it the target does not move with price.
`energy`	AC energy counter (kWh)	unset	Cumulative energy counter. The room model uses it for interval power when the power sensor has no interval mean.
`price_cheap`	Cheap price	`0.15`	Sets where the price shift starts: the midpoint of this price and `price_expensive`. Up to the midpoint the cool threshold is your target.
`price_expensive`	Expensive price	`0.4`	At or above this price the cool threshold is 1 K above your target.

12.5 Heater

KEY	FORM LABEL	DEFAULT	EFFECT
`boiler`	Boiler switch	unset	On when the heat source is the boiler, at 5 °C outdoors or below, and the room is below comfort minus 1 K.
`valve`	Radiator valve opening	unset	Opening in percent. Fully open 1 K below the target, closed at the target.
`trv_temperature`	Radiator valve's own temperature sensor	unset	The TRV's own reading, compared with the room. Each 1 K of offset moves the opening by 100 points, clamped to 0 to 100.
`radiators`	Heater stays warm after it stops	`False`	Counts the residual heat after the heater stops: a 1 K peak that decays over 20 minutes.
`heater_power_w`	Heater power (W)	unset	Rated power of the heater on the heater switch. Gives the heating forecast its energy and cost.

KEY	FORM LABEL	DEFAULT	EFFECT
`heater_algorithm`	Heater control method	`pid`	`pid` or `tpi`. PID stops 0.5 K below the heat target. TPI heats up to the heat target.
`ac_heat`	Heat with the AC	`False`	Lets the AC heat when the heat source is the AC and no heater switch is set.
`mpc_heat_plan`	Plan AC heating by price	`False`	With the AC as heat source, no heater switch and someone home, the heat plan starts AC heat when its first step says heat.
`heater_inverted`	Heater switch is inverted	`False`	Swaps the on and off commands for the heater switch.

12.6 Occupancy and windows

KEY	FORM LABEL	DEFAULT	EFFECT
`weather`	Weather entity	unset	Supplies the daily and hourly forecast. The day type, the night plan, the outdoor temperature series and the outdoor fallback use it. See Read the forecast.
`heatwave_warning`	Heatwave warning sensor	unset	A binary sensor. While it is on, the Heatwave sensor is on. It changes no device command.
`cover`	Blind or cover	unset	Closes the cover above 0.5 kW of solar gain. Opens it again below that.
`window`	Window or door sensor	unset	A window open longer than `window_delay_min` pauses cooling. The heater is also gated.
`motion`	Motion sensor in the room	unset	Motion in the room means occupied. No motion for `motion_away_min` means unoccupied: the room cools to `eco_c` or `away_c` and heats to the eco heat setpoint.

KEY	FORM LABEL	DEFAULT	EFFECT
`presence`	Presence sources	unset	A source in state `on` or `home` means home. All sources away for `away_delay_min` means away. With a motion sensor, motion in the room counts as occupied whatever presence says. Once motion is idle, an empty house heats to the 16 °C frost floor.
`window_actuator`	Window opener	unset	Opens on natural ventilation or economizer advice. Closes when the advice ends. The AC does not cool while the entry holds the window open. When the room is still 2 K above the cool threshold after 30 minutes, the window closes and the AC cools. The same happens when the room is above the cool threshold and fell by less than 0.3 K in 30 minutes.
`motion_away_min`	Minutes without motion before the room is unused	`30.0`	Used only with `motion`. Minimum 1.
`window_delay_min`	Minutes a window may stay open	`10.0`	Minutes a window may stay open before cooling pauses. 0 to 120.
`window_azimuth`	Window direction (degrees from north)	`180.0`	Direction of the window for solar gain.
`window_area_m2`	Window area (m²)	unset	Needed for solar gain.
`irradiance`	Sun sensor on the window (W/m²)	unset	Measured sun on the window for solar gain.

12.7 Air quality and humidity

KEY	FORM LABEL	DEFAULT	EFFECT
`humidity`	Outdoor humidity sensor	unset	Keeps windows shut to humid air. Window advice, the economizer and the window actuator stay closed when the outdoor dew point is 1 K or more above the room's. With no `room_humidity`, the limit is an outdoor dew point of 20 °C.
`room_humidity`	Room humidity sensor	unset	Keeps room humidity between 40 % and the thermostat's humidity target (60 % by default). Above the target the dehumidifier runs. Below 40 % the humidifier runs. Sets the drying status.
`co2`	CO2 sensor	unset	Above 1000 ppm the room fan runs.
`fan`	Room fan switch	unset	Runs for ventilation.
`dehumidifier`	Dehumidifier switch	unset	On while room humidity is above the humidity target (60 % by default), off 3 points below it. In `dry` mode with no humidity sensor it stays on. Off while the room is held off.
`humidifier`	Humidifier switch	unset	On while room humidity is below 40 %, off at 43 %. Off while the room is held off.
`pm25`	PM2.5 sensor	unset	Above 25 µg/m³ the room fan runs.
`whole_house_fan`	Whole-house fan switch	unset	On when the room is above the cool threshold, outdoors is at least 2 K cooler and the house is not sealed. The AC stays off while the fan runs. When the room is still 2 K above the cool threshold after 30 minutes, the fan stops and the AC cools. The same happens when the room is above the cool threshold and fell by less than 0.3 K in 30 minutes.

KEY	FORM LABEL	DEFAULT	EFFECT
`coil_dry`	Dry the AC coil after cooling	`False`	Runs the AC dry mode for 20 minutes after cooling.
`dry_for_humidity`	Dehumidify with the AC	`False`	Above the humidity target (60 % by default), with no dehumidifier set, the AC runs dry mode. Dry mode uses a lot of water.
`economizer_fan`	Help free cooling with the room fan	`False`	Runs the fan switch in the economizer fan assist phase. Needs home, unlocked and no stalled sensor.
`auto_fan`	Set the AC fan speed from the room	`False`	Sets the AC fan mode from the gap to the target while cooling.
`house_sealed`	Never run the whole-house fan	`False`	Stops the whole-house fan.

12.8

Devices and power limits

KEY	FORM LABEL	DEFAULT	EFFECT
`climate_2`	Second AC in the room	unset	Adds a second cooling device to the room model.
`power_2`	Second AC power sensor (W)	unset	Power of the second AC. With `climate_2` and this field, the two-device fit runs.
`whole_house_power`	Whole-house power sensor (W)	unset	Total house power, W. Used with `shed_max_w`.
`shed_max_w`	House power limit (W)	unset	House limit, W. Above it, the loads in `shed_loads` switch off.
`shed_loads`	Loads to switch off	unset	One `entity_id|watts` entry per load. The first entry sheds first.
`compressor_group`	Compressor group	unset	Entries with the same name keep 5 minutes between their AC starts.

KEY	FORM LABEL	DEFAULT	EFFECT
`fleet`	Room devices	`[]`	The heaters and coolers in the room. HVAC Horizon runs the ones that cover the gap to the target. See Settings for the device keys.
`dispatch_mode`	Which room device runs first	`fast`	`fast` runs the device that delivers the most first. `economy` runs the device with the lowest cost per kelvin first.

12.9

Advanced

KEY	FORM LABEL	DEFAULT	EFFECT
`lock_pin`	Lock PIN	unset	Needed to lock control. See Troubleshoot.
`night_plan_act`	Let the night plan act	`False`	Pre-cool cools to comfort. Hold stops a cool start.
`accuracy_panel`	Show the accuracy panel	`False`	Adds this entry to one shared sidebar panel.
`peer_entry`	Peer room	unset	When this room is 2 K warmer, the peer entry's valve opens.
`peer_setpoint_mode`	Peer room lowers its setpoint	`False`	With a peer entry, a room 2 K warmer lowers the peer's cool setpoint by 1 K instead of opening its valve.
`control`	Let HVAC Horizon control the room	`False`	Off: observe only. HVAC Horizon learns and forecasts and sends no commands. On: it drives the AC, the heater and the other devices.
`ai_conversation`	Conversation agent for explanations	unset	Writes the investigator answer when the AC mode changes, within the daily budget.
`ai_budget_per_day`	Explanations per day	unset	Daily limit on model calls.

12.10

Expert tuning

Controller gains, output limits and timing. The defaults suit most rooms.

KEY	FORM LABEL	DEFAULT	EFFECT
`heater_kp`	PID proportional gain	`20.0`	Used at the default. A value you set wins. At the default, the gain from a relay test applies.
`heater_ki`	PID integral gain (per hour)	`60.0`	Same rule as `heater_kp`.
`heater_kd`	PID derivative time (hours)	`0.5`	Same rule as `heater_kp`.
`tpi_coef_int`	TPI room coefficient	`0.6`	TPI gain on the room gap.
`tpi_coef_ext`	TPI outdoor coefficient	`0.01`	TPI gain on the outdoor gap.
`tpi_threshold_high`	TPI stop above target while rising (°C)	unset	With the low threshold, forces the TPI output to 0 while the room rises and is this far above the target. Both must be above 0 to apply. TPI then heats up to the larger threshold above the target.
`tpi_threshold_low`	TPI stop above target while falling (°C)	unset	With the high threshold, forces the TPI output to 0 while the room falls and is this far above the target. Both must be above 0 to apply.
`max_on_percent`	TPI maximum output (%)	unset	Caps the TPI output, 0 to 100 %.
`output_min`	Minimum heater output (%)	`0.0`	Lower limit for the heater output.
`output_max`	Maximum heater output (%)	`100.0`	Upper limit for the heater output.
`output_min_entity`	Minimum heater output from a number entity	unset	When set, its value overrides `output_min`.
`output_max_entity`	Maximum heater output from a number entity	unset	When set, its value overrides `output_max`.
`min_on_min`	Minimum AC on time (not used)	unset	Deprecated and ignored. A stop is never delayed.
`min_off_min`	Restart delay (minutes)	`3.0`	Restart delay. A start waits this long after the device stopped. 0 disables it. 0 to 120.

KEY	FORM LABEL	DEFAULT	EFFECT
`stall_window_s`	Seconds before a silent sensor counts as failed	`900.0`	Range 300 to 3600 seconds. See Troubleshoot.
`setpoint_regulation`	Correct the AC setpoint for its own sensor	`True`	Lowers the setpoint sent to the AC by the gap between the room median and the AC's own `current_temperature`, so an AC whose sensor reads cool still runs. An AC with no such attribute is regulated from power.

12.11

Room devices (`fleet`)

`fleet` is a list of the heaters and coolers in the room. When it is set, HVAC Horizon picks which devices to run for the room's gap to the target. `dispatch_mode` sets the order: `fast` runs the device that delivers the most first, `economy` runs the device with the lowest cost per kelvin first. The `dispatch` sensor shows the choice. See Sensors.

The **Room devices** step adds one device per submit. Each device is stored with these keys.

KEY	REQUIRED	DEFAULT	MEANING
`device_id`	required	none	Name of the device. Must be unique in the list.
`kind`	required	none	`cool` or `heat`.
`rated_w`	required	none	Watts the device draws while it runs. Used when there is no power sensor.
`entity_id`	optional	none	The `climate` entity that drives the device. A `heat` device can use a `switch` instead. Without it the device is chosen but not driven.
`power_sensor`	optional	none	Power sensor in watts. Its reading replaces `rated_w` while the device runs.

KEY	REQUIRED	DEFAULT	MEANING
`rate_k_per_h_per_kw`	optional	none	Room change in kelvin per hour for each kW the device draws. A device without a rate is not chosen. Once the device has run alone for 6 intervals of 5 minutes, the fitted rate replaces it.

FLEET

```
[
  {"device_id": "split", "kind": "cool", "rated_w": 900, "rate_k_per_h_per_kw":
  4.0, "entity_id": "climate.lounge_ac"},
  {"device_id": "panel", "kind": "heat", "rated_w": 1500, "rate_k_per_h_per_kw":
  2.0, "entity_id": "switch.panel_heater"}
]
```

Fleet devices use the same restart delay `min_off_min` as the AC. The form shows `Room devices are invalid` with the reason when the list breaks a rule.

SEE ALSO

- Set up the integration: how to open each form section.
- Sensors: what the entry publishes.

13

Sensors

Every sensor and attribute an HVAC Horizon entry publishes, with its key, unit and meaning.

13.1

Entities per entry

Each entry creates one device named `HVAC Horizon` and the entry's title: the area's name, or the AC's when no area is set. The device holds one `climate` entity for the room, 73 sensors and 13 binary sensors. See Use the room thermostat for the climate entity.

A sensor's entity ID is `sensor.hvac_horizon_<title>_<name>`, and a binary sensor's starts with `binary_sensor`. The names are entity translations, so Home Assistant can show them in the user's language once a translation exists.

Numeric sensors have a device class: temperatures, temperature differences, durations, energy and power. Home Assistant shows them in your unit system. The Key column is the suffix of the unique ID and the name of the value in the diagnostics download.

13.2

Primary sensors

These 12 sensors have no entity category. Home Assistant shows them on the device page and in dashboards.

KEY	NAME	UNIT	MEANING
`minutes_to_setpoint`	Minutes to setpoint	min	Forecast minutes for the room to reach the cool threshold at the current running power. In a heating or drying forecast, the minutes to that target.
`energy_kwh`	Energy to setpoint	kWh	AC energy for that time. An upper bound for a unit that cycles.
`cost`	Cost to setpoint	none	That energy priced slot by slot from the price forecast. Empty with no price sensor.
`status`	Status	none	State of the forecast: `learning`, `no input`, `no running power yet`, `at setpoint`, `reachable` or `unreachable`. Carries the attributes below.
`forecast_error_min`	Forecast error	min	Median absolute error of the backtest runs, in minutes.
`target_c`	Target temperature	°C	With control on, the temperature the AC cools above this tick. With control off, the comfort setpoint.
`preset`	Preset	none	The preset in use: `comfort`, `eco`, `boost` or `away`.
`cool_on_above_c`	Cool above	°C	The AC cools when the room is above this value.
`heat_on_below_c`	Heat below	°C	The heater heats when the room is below this value: the heat target minus 0.5 K.
`minutes_low`	Minutes to setpoint, low	min	Low end of the forecast range, from the standard error of the cooling rate.
`minutes_high`	Minutes to setpoint, high	min	High end of the forecast range.
`forecast_mode`	Forecast mode	none	What the forecast sensors show: `cool`, `heat` or `dry`.

13.3

Diagnostic sensors

These 21 sensors have the entity category `diagnostic` and are enabled. They show under **Diagnostic** on the device page.

KEY	NAME	UNIT	MEANING
`samples`	Samples collected	none	Stored readings. The store keeps 8640, 30 days at 5 minutes.
`day_type`	Day type	none	Day class from the forecast high, or the outdoor reading with no weather entity.
`night_plan`	Night plan	none	`pre-cool`, `hold` or none, from the room and the coming night's low.
`precool_status`	Pre-cool status	none	Why a pre-cool start exists or not, for example `start in 25 min`, `late: starts now, 30 min to the event` or `no schedule entity`.
`daily_briefing`	Daily briefing	none	One line: the day type, the comfort target and the next action.
`window_advice`	Window advice	none	`open`, `close` or `hold` for windows, from the room, the outdoor temperature and the outdoor humidity.
`ventilation`	Ventilation	none	Why the room fan runs or not, from CO2 and PM2.5.
`humidity`	Humidity action	none	`dehumidify`, `humidify` or `within band`, from room humidity.
`compliance_pct`	Comfort compliance	%	Share of stored readings inside the comfort band.
`occupancy`	Occupancy	none	`home` or `away`, from the presence sources and `away_delay_min`.
`learning_suggestion`	Learning suggestion	none	Set when the fitted heat-gain rate drifts from the stored baseline. Answer with `respond_suggestion`.
`sensors_stalled`	Stalled sensors	none	The sensors with no report inside the stall window.

KEY	NAME	UNIT	MEANING
`mpc_status`	MPC plan effect	none	Whether the cooling plan changed the decision this tick.
`heat_source`	Heat source	none	`boiler` at 5 °C outdoors or below, otherwise `ac`.
`dispatch`	Room dispatch	none	The fleet devices chosen for the current gap, with each device's rate and power.
`pending`	Pending action	none	A start held by the restart delay, for example `start in 2 min: restart delay`, or the gate that holds a wanted heater off, for example `heat held: window open`.
`ac_setpoint_sent_c`	Setpoint sent	°C	The setpoint last sent to the AC, after setpoint regulation.
`ac_sensor_offset_k`	Sensor offset	K	Room median minus the AC's own temperature reading.
`outdoor_source`	Outdoor source	none	`sensor`, or `forecast` while the outdoor sensor is stale and the hourly forecast is under 2 hours old.

13.4

Sensors disabled by default

These 40 sensors hold internal detail. To use one, open it in **Settings > Devices & services > Entities** and turn on **Enabled**. The coordinator publishes every key whether or not its sensor is enabled.

KEY	NAME	UNIT	MEANING
`ventilation_rate`	Ventilation rate	none	Fan duty from CO2, 0 to 1.
`fan_speed`	Fan speed	none	AC fan mode chosen by `auto_fan`.
`price_used_per_kwh`	Price used per kWh	none	The price used this tick. The last good price stands in for 60 minutes when the sensor drops.
`solar_gain_kw`	Solar gain	kW	Solar gain through the window, measured or modelled.

KEY	NAME	UNIT	MEANING
band_action	Band action	none	heat, cool or idle against the comfort band.
samples_heat	AC heat intervals	none	Stored intervals with the AC in heat mode. The cooling fit leaves them out.
samples_dry	AC dry intervals	none	Stored intervals with the AC in dry mode. The cooling fit leaves them out.
heater_feedback	Heater feedback	none	Whether the heater switch state matches the command.
valve_opening_pct	Valve opening	%	Valve opening sent to the TRV.
cover_action	Cover action	none	close, open or none for the cover, from solar gain.
second_ac	Second AC	none	State of the second AC drive.
rooms_max_age_s	Room reading age	s	Age of the oldest room reading in use.
safety_output_pct	Safety output	%	Heater output forced by a stalled sensor.
economizer	Economizer	none	off, open_windows or fan_assist.
motion_preset	Motion preset	none	home or away from the motion sensor.
shed	Loads shed	none	Loads switched off by the power limit this tick.
restored	Load switched back on	none	Load switched back on this tick.
mpc_plan	MPC plan steps	none	Cooling plan steps, 15 minutes each for 4 hours.
mpc_heat_steps	MPC heat plan steps	none	Heat plan steps.
solar_scale	Solar scale	none	Learned ratio of measured to modelled solar gain.
heater_pid_output	Heater PID output	%	PID output before limits.
heater_output_pct	Heater output	%	Heater output after limits and gates.
tpi_output_pct	TPI output	%	TPI output.
preheat	Preheat	none	off, waiting, active or cannot reach comfort before a comfort block.
heat_up_min	Heat-up time	min	Learned minutes to heat to comfort.

KEY	NAME	UNIT	MEANING
`backtest_runs`	Backtest runs	none	Count of backtest runs.
`trv_offset_k`	TRV offset	K	TRV reading minus room reading.
`room_fit`	Two-device room fit	none	The two-device room fit.
`forecast_days`	Forecast day counts	none	Count of forecast days by day type.
`outdoor_forecast_next_c`	Outdoor forecast ahead	none	Outdoor series 1, 3 and 6 hours ahead in °C, with the learned bias.
`solar_factor`	Solar cloud factor	none	Share of clear-sky solar gain from the forecast condition: sunny 1.0, partlycloudy 0.7, cloudy 0.4, rainy 0.2.
`outdoor_bias_now_k`	Outdoor forecast bias	K	Learned outdoor sensor minus forecast for this hour of day. 0 until 6 pairs exist.
`outdoor_bias_pairs`	Outdoor bias pairs	none	Pairs the bias has learned from. `reset_learning` clears them.
`forecast_target`	Forecast target	none	Target of a heating forecast in °C, or of a drying forecast in percent. Unknown for a cooling forecast.
`fit_corrections`	Fit corrections	none	The kept corrections for sun, outdoor temperature and humidity, with `improvement_pct`. Unknown with none.
`fit_window_days`	Fit window	d	Days of readings the fit used: 14 or 30.
`kept_models`	Kept models	none	Each model in use that comes from an earlier fit, with its age in days: `fit`, `tpi`, `heat_rate`, `ac_heat`, `dry`, `corrections`. Unknown when every model is from the stored readings.
`heat_model`	Heat model	none	The heating model: `source` (`heater` or `ac`), `k_per_h` and `power_kw`.
`dry_model`	Dry model	none	The drying model: `pct_per_h` and `samples`.
`free_cooling`	Free cooling	none	`idle`, `running` or `gave_up`.

KEY	NAME	UNIT	MEANING
`commands_unconfirmed`	Unconfirmed commands	none	Devices whose state differs from the last command on two ticks in a row, and commands a device refused since the last tick. Enabled by default.
`watchdog_trips`	Watchdog trips	none	Times the watchdog turned the outputs off since the entry loaded. Enabled by default.

A list or dict value shows as JSON text cut to 255 characters in the state. The full value is in the `value` attribute. A key the coordinator did not publish this tick reads `unknown`.

13.5

Binary sensors

These 13 values are on or off. Each is a `binary_sensor` entity with the entity category `diagnostic`. Before 0.7.0 they were sensors that read `True` or `False`.

KEY	NAME	DEVICE CLASS	ON MEANS	ENABLED
`drying_needed`	Drying needed	none	Room humidity is above the humidity target (60 % by default) and the room needs no cooling. In `dry` mode the cooling condition does not apply.	yes
`mould_risk`	Mould risk	`problem`	The estimated surface humidity is 80 % or more. Every cool threshold is then 1 K higher.	yes
`heatwave`	Heatwave	`heat`	`heatwave_warning` is on, or the daily highs reach 30 °C three days running. It changes no device command.	yes

KEY	NAME	DEVICE CLASS	ON MEANS	ENABLED
`heating_allowed`	Heating allowed	none	Heating may run. Off while outdoors is above 22 °C in the automatic mode or `dry`: the heater, the boiler and AC heat then stay off. On in `heat` mode.	yes
`external_change`	External change	none	The AC setpoint differs from the one HVAC Horizon last wrote: someone changed it by hand.	yes
`vacation`	Vacation mode	none	The vacation switch is on.	yes
`whole_house_fan`	Whole-house fan	`running`	The whole-house fan may run.	yes
`heating_failure`	Heating failure	`problem`	The heater ran 60 minutes and the room rose less than 0.5 K.	yes
`ac_failure`	AC failure	`problem`	The AC has been in `cool` or `dry` for 20 minutes while the room still needs it, and its power stayed under 200 W for more than half of them. Needs the power sensor. See [[guides/troubleshoot#the-ac-is-on-and-the-compressor-does-not-run	The AC is on and the compressor does not run]].
`outdoor_forecast_used`	Outdoor forecast used	none	A fresh hourly forecast feeds the forecast, the plans, pre-cool and preheat.	yes
`heater_pwm_on`	Heater PWM on	`running`	The heater is on inside the current 10 minute window.	no
`boiler_demand`	Boiler demand	none	The boiler should run.	no
`house_cooling`	House cooling	none	Another entry of the same house cools.	no

A binary sensor reads `unknown` until its value is computed.

13.6

Attributes

These attributes sit on the entity named in the Entity column. They are read from the coordinator's last result.

ATTRIBUTE	ENTITY	CONTENT
`reason`	Status	Why the forecast or the fit is in its current state: the fit reason (`fitted`, `not enough data`, `need at least 3 hours of AC on and off time over 3 runs`) or the forecast reason (`reachable`, `not reached within 12 hours`). It does not hold the control decision. The action text is in `investigation` and the Daily briefing.
`k_per_h`	Status	Heat-gain rate toward outdoors of the fitted model, in 1/h.
`c_k_per_kw`	Status	Cooling effect of the AC, in K per hour per kW.
`g_k_per_h`	Status	Constant gain of the fitted model (internal heat), in K per hour.
`autotune_status`	Status	State of the last relay test: `running`, `done`, `timed_out`, `cancelled` or `reached_cutoff`. None before any test.
`autotune`	Status	Gains kp, ki and kd after a relay test. None before.
`shed_skipped`	Status	Set when a shed load entry is malformed.
`investigation`	Status	Rule-based answer to why the AC runs or does not, with recent actions.
`investigation_ai`	Status	Conversation agent answer, when `ai_conversation` is set.
`investigation_ai_mode`	Status	The AC mode that the agent answer explains. None with no answer.
`ai_calls_today`	Status	Model calls made today, counted against `ai_budget_per_day`.
`investigation_report`	Status	The 24-hour report from `generate_investigation_report`. None until it runs.
`investigation_report_source`	Status	`model` when the agent answered, otherwise `rule`.
`investigation_report_at`	Status	ISO time the report was stored.

ATTRIBUTE	ENTITY	CONTENT
`room_rate_coefficients`	Status	k, g from AC-off intervals and c from AC-on intervals. None until fitted.
`auto_tpi_gains`	Status	Online cool gain on c, clamped from 0.5 to 1.5. 1.0 until it moves.
`accuracy_pairs`	Forecast error	The last 20 backtest runs, each with predicted, measured and error minutes.
`range_coverage_pct`	Forecast error	Percentage of the backtest runs whose measured minutes fell between the low and high forecast of that run.
`range_runs`	Forecast error	Count of backtest runs behind `range_coverage_pct`.

SEE ALSO

- Read the forecast: how to read the forecast sensors and the status.
- Settings: the settings that feed each sensor.