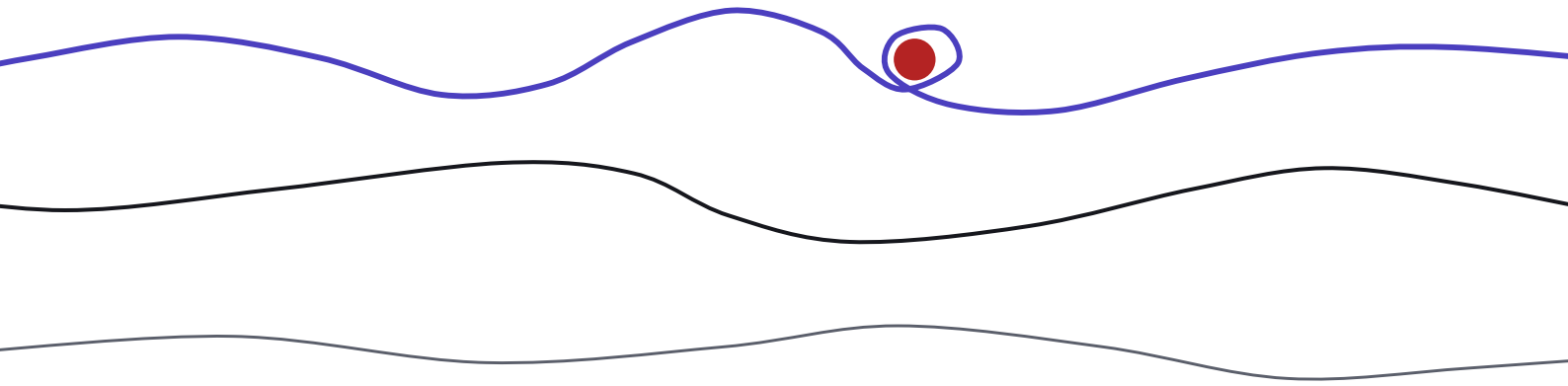
jupiter.au

Carpark Australia

Parking availability across Australia, in Home Assistant.



VERSION

v0.2.0

DATE

25 November 2026

AUTHOR

belikh

SOURCE

github.com/belikh/ha-australia-carpark

Contents

Use

01 **Install** 3

01.1 Install (HACS) 3

02 **Configure** 3

02.1 Area to track 3

02.2 Tracked car parks 4

Build

03 **Entities** 5

03.1 Scope-level sensors 5

03.2 Scope sensor attributes 5

03.3 Per-car-park entities (opt-in) 6

04 **Data sources** 6

04.1 Notes on the data 7

Write

05 **Develop** 8

05.1 Attribution 8

05.2 Licence 8

CARPARK AUSTRALIA

Carpark Australia

Carpark Australia is a Home Assistant integration that counts parking spaces across the area you choose and exposes a single set of roll-up sensors. Each sensor's attributes list the car parks in scope, how many spaces are free or occupied at each, and any pricing information the source publishes.

Install it from HACS, choose your data sources and your area, then ask "how many parking spaces are free near home?" without reading hundreds of sensors.

Install

Configure

Entities

Data sources

Develop



SEE ALSO

- Data sources
- Entities

SECTION

Use

Install and configure the integration.

01

Install

Add the integration from HACS and choose your data sources.

01.1

Install (HACS)

1. Open HACS and add a custom repository: `https://github.com/belikh/ha-australia-carpark` as type **Integration**.
2. Install **Carpark Australia** and restart Home Assistant.
3. Add the integration from **Settings → Devices & Services → Add integration**.
4. Choose the data sources, any car parks to track individually, and the area to track.

Transport for NSW needs an API key. Register at developer.transport.nsw.gov.au and paste it into the config flow. Every other source is token-free.

Configure

02

Configure

Choose your data sources, tracked car parks and area.

The config flow asks for:

1. **Data sources**: the providers to poll. Only selected providers are polled on each refresh.
2. **Tracked car parks**: optional. Search and select specific car parks to track with their own entities. Leave empty to only use the scope-level sensors.
3. **Area to track**: around the Home Assistant location, inside a zone, or all of Australia.

02.1

Area to track

Tracking every car park in Australia creates a lot of entities. Choose the area that covers where you actually park:

SCOPE	WHAT IT COUNTS
Around the Home Assistant location	Car parks within a radius (default 5,000 m) of your configured home position
Inside an existing zone	Car parks inside any Home Assistant zone you have defined
All of Australia	Everything the selected sources publish

Zones

Pick **Inside an existing zone** and choose one of your zones, such as `zone.home`, `zone.work` or a suburb boundary you drew. The integration reads the zone's latitude, longitude and radius, and counts the parking spaces inside it:

FIND PARKING BEFORE I LEAVE WORK

```
automation:
  - alias: "Find parking before I leave work"
    trigger:
      - platform: state
        entity_id: sensor.parking_spaces_free
        below: 50
    action:
      - service: notify.mobile_app
        data:
          message: "Only {{ states('sensor.parking_spaces_free') }} parking spaces
          free near work."
```

The `scope` attribute on each roll-up sensor tells you which area a figure covers: the zone name, `within 5000 m of home`, or `Australia`.

The integration does not create zones. It only reads the ones you already have.

02.2

Tracked car parks

During setup you can search and select specific car parks to track individually. Each selected car park becomes a device with its own entities:

ENTITY	TYPE	MEANING
`Spaces free`	sensor	Free spaces right now
`Occupancy`	sensor	Percentage of the car park that is occupied
`Total spaces`	sensor	Capacity, for car parks that publish no live count
`Spaces available`	binary sensor	On when at least one space is free

Leave the selection empty to only use the scope-level sensors. You can change the selection later under **Configure** on the integration card.

Entities

SECTION

Build

Entities, data sources and attributes.

03

Entities

Scope-level sensors and per-car-park entities.

03.1

Scope-level sensors

The scope sensor is the main roll-up entity. It counts parking spaces across the whole area in scope, so you get one number instead of one sensor per car park:

ENTITY	TYPE	MEANING
`Parking spaces free`	sensor	Free spaces counted across every car park in scope
`Parking spaces total`	sensor	Capacity counted across every car park in scope
`Parking spaces occupied`	sensor	Occupied spaces counted across every car park in scope
`Parking spaces free percentage`	sensor	Free share of the counted capacity
`Car parks tracked`	sensor	How many car parks are in scope

The roll-up sensors carry a `scope` attribute naming the area they cover, plus `carparks_counted_free`, `carparks_counted_total` and `carparks_counted_both` so you can see how many car parks contributed to each figure.

03.2

Scope sensor attributes

The `carparks` attribute on each roll-up sensor lists every car park in scope:

- name, provider, operator
- free, total and occupied spaces

- occupancy percentage
- pricing (when the source publishes it)
- address, coordinates, URL

The `free_carparks` attribute lists only the car parks that currently have at least one free space, with the same detail. Long lists are truncated to keep the attribute readable.

03.3

Per-car-park entities (opt-in)

During setup you can search and select specific car parks to track individually. Each selected car park becomes a device with its own entities:

ENTITY	TYPE	MEANING
`Spaces free`	sensor	Free spaces right now
`Occupancy`	sensor	Percentage of the car park that is occupied
`Total spaces`	sensor	Capacity, for car parks that publish no live count
`Spaces available`	binary sensor	On when at least one space is free

Leave the selection empty to only use the scope-level sensors.

Configure

04

Data sources

The councils and operators that publish parking data.

SOURCE	DATA	CAR PARKS	AVAILABILITY
City of Adelaide (UPark)	Free spaces per car park	10	Live
Transport for NSW	Occupancy per facility	Not listed	Live, needs an API key
Brisbane City Council	Occupancy forecast per paid parking zone	617	Forecast only
Wilson Parking	Locations, services, EV and accessible parking	568	Locations only
City of Sydney	Council car park locations	2	Locations only
City of Perth	Car park locations and capacity	32	Capacity only
ACT Government	Car park asset locations	1,368	Locations only
City of Hobart	Car park and parking structure locations	12	Locations only

Counts are as published by each source and change over time.

04.1

Notes on the data

- **Brisbane publishes a forecast.** It has no live count. The council's occupancy forecasting dataset predicts how many bays will be occupied for each hour of the day, using parking meter transactions and occupancy surveys. Treat the Brisbane figures as an estimate.
- **Adelaide publishes free spaces but no capacity.** Its free-space counts are included in `Parking spaces free`, but they are not part of the percentage calculation, which needs both figures.
- **Several sources publish locations only.** Wilson Parking, the City of Sydney, the ACT Government and the City of Hobart do not publish live availability. Their car parks appear in the `Car parks (<provider>)` catalogue sensor and count towards `Car parks tracked`. They produce no per-park sensors.
- **Private operators are included.** Wilson Parking operates car parks in every state and territory and is listed as a source in its own right, with the services each site offers (monthly parking, book-a-bay, EV charging, accessible parking).
- **Car parks without coordinates are skipped when you scope by distance or zone.** Every source listed above publishes coordinates except the ACT Government polygon layer, whose centroids are used instead.
- **Pricing information.** None of the current data sources publish a price per hour or per stay. When a source does publish pricing, it will appear in the `pricing` attribute of each car park listed in `carparks` and `free_carparks`.

Configure

SECTION

Write

Develop the integration.

05

Develop

Run tests, lint and type checks.

BASH

```
uv sync
uv run pytest
uv run ruff check .
uv run mypy
```

Hassfest can be run with `python -m homeassistant.hassfest` via the script in `.github/workflows/`.

05.1

Attribution

- City of Adelaide UPark availability reports: City of Adelaide open data.
- Brisbane City Council parking meter and occupancy forecasting data: Brisbane City Council open data, CC BY 4.0.
- City of Sydney, City of Perth, ACT Government and City of Hobart car park data: the relevant council's ArcGIS open data service.
- Transport for NSW car park data: Transport for NSW Car Park API.
- Wilson Parking car park details: wilsonparking.com.au.
- Adelaide car park coordinates: © OpenStreetMap contributors, licensed under the Open Database License (ODbL).

05.2

Licence

MIT. Unofficial; not endorsed by any council or operator.

Install