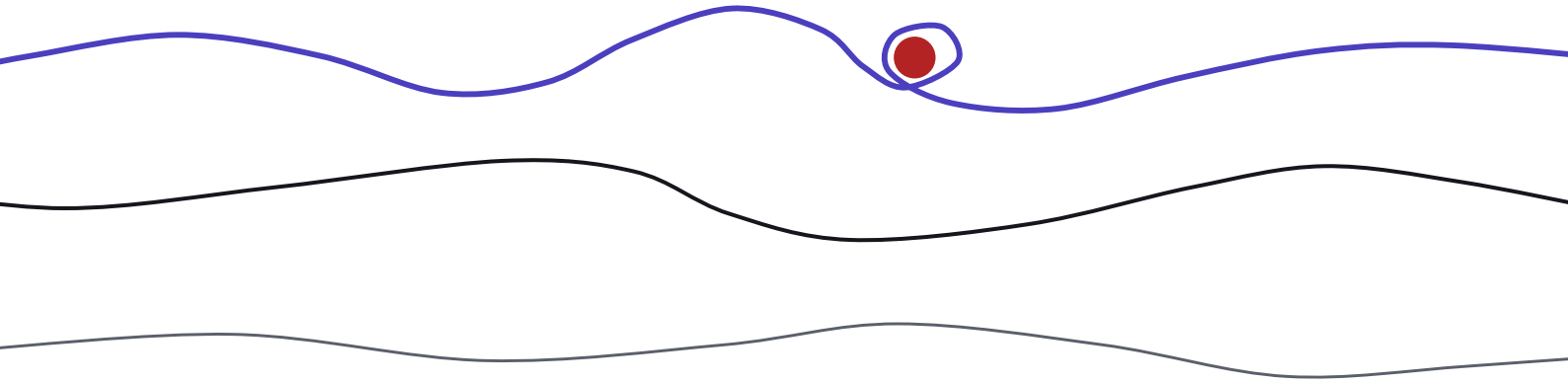
jupiter.au

ANKO Body Scale

Home Assistant integration for the ANKO iF1012D Bluetooth body scale: weight and body composition per person, read from passive BLE adverts.



VERSION

v2.0.0

DATE

25 November 2026

AUTHOR

belikh

SOURCE

github.com/belikh/ha-anko-body-scale

Contents

00.1 What you get 2
00.2 Where to start 2

Start

01 **Install** 3
01.1 Install with HACS 3
01.2 Install by hand 3

02 **Set up a scale** 4
02.1 Add the scale 4
02.2 Take the first weigh-in 4
02.3 Setup messages 5

Guides

03 **Manage people** 6
03.1 Add a person 6
03.2 Send the next weigh-in to a person 7
03.3 Skip a person who is away 7
03.4 Delete a person 7

04 **Add a second scale** 7
04.1 Add the scale 7
04.2 Change who uses a scale 8
04.3 Actions with several scales 8

05 **Resolve pending measurements** 8
05.1 Find the pending ID 8
05.2 Assign it to a person 9
05.3 Discard a guest's weigh-in 9
05.4 Answer from your phone 9
05.5 Add a person from an unknown weight 9
05.6 Move the last weigh-in to another person 10
05.7 Errors 10

06 **Automate weigh-ins** 10
06.1 Notify on a weigh-in 10
06.2 Show the sensors on a dashboard 11

06.3 Weigh something without a person 11
06.4 Example files 11

07 Upgrade to 2.0 12
07.1 What the migration does 12
07.2 Two scales with the same person 12
07.3 Changed actions 12
07.4 Removed sensors 12

08 Troubleshoot 12
08.1 The setup finds no scale 13
08.2 Person sensors show unknown 13
08.3 Every weigh-in becomes pending 13
08.4 The scale shows a weight and the person has no body fat 13
08.5 The method attribute is always bmi 13
08.6 Body composition looks wrong 14
08.7 Values are old after a restart 14
08.8 A scale entry shows an error 14

09 Report a problem 14
09.1 Collect the debug log 14
09.2 Download diagnostics 15

10 Remove the integration 15

Reference

11 Entities 16
11.1 Scale device 16
11.2 Person device 17
11.3 The method attribute 18
11.4 Restored state 18

12 Actions 18
12.1 Fields 19
12.2 Capture response 20
12.3 Reassign and reset 20
12.4 Errors 20

13 Events 21
13.1 anko_scale_weigh_in 21

13.2 anko_scale_pending_measurement 22

14 Settings 23

14.1 Attribution settings 23

14.2 Fixed values 24

15 Glossary 24

Concepts

16 How readings are assigned 26

16.1 From adverts to a weigh-in 26

16.2 Choosing the person 26

16.3 Worked example 27

16.4 Expected weight 27

16.5 Why weight 27

17 How body composition is calculated 27

17.1 The impedance route 28

17.2 The BMI route 29

17.3 Accuracy 29

18 Advert format 29

19 Limitations 30

Develop

20 Build and test 31

20.1 Layout 31

20.2 Documentation 32

20.3 Workflows 32

21 Release 32

ANKO Body Scale

The integration reads the ANKO iF1012D body scale, sold by Kmart Australia and advertised over Bluetooth as `IF_B7`. Home Assistant listens to the scale's *Advert* broadcasts. It opens no connection to the scale and sends nothing to the internet.

Each *Weigh-in* gives a weight and a raw impedance code. The integration assigns the weigh-in to a *Person* by weight and computes 12 body-composition values from that person's height, age and sex.

[image: Add to Home Assistant]

00.1

What you get

- Setup from the UI with Bluetooth discovery.
- One device per person with 14 sensors. They cover weight, BMI, body fat, fat mass, lean mass, muscle mass, body water, bone mass, basal metabolic rate, metabolic age and visceral fat index.
- People shared by every scale. Each scale can be limited to some of the people.
- A *Pending measurement* for a weigh-in that fits two people or nobody, with actions, events and phone notifications to resolve it.
- Six actions, two events and a repair that adds a person from an unknown weight.
- Weight history and pending measurements that survive a restart.
- A diagnostics download with names, birth dates and the Bluetooth address redacted.

00.2

Where to start

SEE ALSO

- Install
- Set up a scale
- Manage people
- Entities
- Troubleshoot

SECTION

Start

Install the integration and add your first scale.

01 Install

Add the integration to Home Assistant with HACS or by copying one folder.

Install with HACS, then restart Home Assistant. The integration needs Home Assistant 2026.9.0 or newer and a Bluetooth adapter or Bluetooth proxy in range of the scale.

01.1 Install with HACS

[image: Add to Home Assistant]

1. Select the badge above and choose **Download**. You can also search for **ANKO Body Scale** in HACS.
2. Restart Home Assistant.

If the search finds nothing, add the repository by hand:

1. In HACS, open the three-dot menu at the top right and choose **Custom repositories**.
2. Enter github.com/belikh/ha-anko-body-scale with category **Integration**.
3. Install **ANKO Body Scale** and restart Home Assistant.

01.2 Install by hand

1. Copy the `custom_components/anko_scale` folder from the repository into the `custom_components` directory of your Home Assistant configuration.
2. Restart Home Assistant.

RESULTING LAYOUT

```
<config>/custom_components/anko_scale/manifest.json
```

SEE ALSO

- Set up a scale

02

Set up a scale

Add the scale to Home Assistant, enter the first person and take the first weigh-in.

Add the scale from **Settings > Devices & Services**. The scale must be awake and in Bluetooth range: step on it for a moment before you start.

02.1

Add the scale

[image: Add integration]

1. Select the badge above, or go to **Settings > Devices & Services > Add Integration** and search for **ANKO Body Scale**. If Home Assistant shows a discovery card for the scale, select **Add** on the card and confirm that the scale is yours.
2. Choose the scale, advertised as `IF_B7`, from the **Device** list.
3. Enter the first person and select **Submit**.

FIELD	REQUIRED	RANGE	USE
Name	yes	text	Names the person's device. Its slug is the *Profile ID*.
Height (cm)	yes	100 to 250, default 170	Used in every body-composition value.
Sex	yes	Male or Female, default Male	Used in every body-composition value.
Birth date	one of the two	`YYYY-MM-DD`, in the past	The age follows the calendar.
Age in years	one of the two	5 to 120	Used only when no birth date is set.
Person entity	no	a `person` entity	The person is skipped while the entity is `not_home`.
Starting weight in kg	no	10 to 300	Seeds the *Expected weight*.

Home Assistant creates two entries. The scale entry has the scale device. The *People entry*, named **ANKO Body Scale people**, has one device per person.

When two or more people exist already, the setup of a new scale asks **Who uses this scale?** in place of the person form. See Add a second scale.

02.2

Take the first weigh-in

1. Step on the scale with bare feet. Each foot covers both metal pads on its side.
2. Stand still until the display stops changing.
3. Open the person's device. **Weight** and the body-composition sensors show the values of this

weigh-in.

02.3

Setup messages

MESSAGE	CAUSE	FIX
`No ANKO body scales were found on the network.`	Home Assistant has not seen an `IF_B7` advert.	Step on the scale to wake it, then start the setup again.
`This scale is already configured.`	The scale has an entry already.	Open the existing entry.
`Enter a name.`	The name is empty.	Type a name.
`Another person already has this name.`	The name gives a profile ID that is taken.	Choose another name.
`Enter a past date as YYYY-MM-DD, for example 1985-03-27.`	The birth date is malformed or in the future.	Type the date as year, month, day.
`Enter a birth date or an age.`	Both fields are empty.	Fill in one of them.

SEE ALSO

- Manage people
- Resolve pending measurements
- Entities

SECTION

Guides

Common tasks, as numbered steps.

03

Manage people

Add, edit and delete people, and send a weigh-in to a person by hand.

People live in the *People entry*. Open **Settings > Devices & Services > ANKO Body Scale**, find **ANKO Body Scale people** and select **Configure**.

MENU ITEM	RESULT
Add a person	Opens the person form from Set up a scale.
Add from a Home Assistant person	Asks for a `person` entity, then opens the form with the name filled in and the entity linked.
Edit a person	Changes the same fields. The *Profile ID* and the entity IDs stay the same.
Delete a person	Removes the person's device, entities and weight history.
Attribution settings	Opens the settings in Settings.

Saving reloads the entry.

03.1

Add a person

1. Select **Configure** on **ANKO Body Scale people**, then **Add a person**.
2. Fill in the name, height, sex and a birth date or an age.
3. Type a **Starting weight** if you know it. A person with no starting weight and no history cannot be matched by weight.
4. Select **Submit**. The new device appears with 14 sensors and a **Weighing in** button.

A scale set to everyone includes the new person. A scale limited to some people does not: add the person under **Configure** on that scale.

03.2

Send the next weigh-in to a person

Use this for a first weigh-in with no starting weight, or when two people have similar weights.

1. Press **Weighing in** on the person's device, or choose the person in **Next person** on the scale device.
2. Step on the scale within 60 seconds.

The next *Weigh-in* goes to that person and joins their weight history. The override ends after one weigh-in or 60 seconds. **Weighing in** arms every scale that allows the person. **Next person** arms one scale.

03.3

Skip a person who is away

1. Select **Configure** on **ANKO Body Scale people**, then **Edit a person**.
2. Choose the person's `person` entity in **Person entity** and select **Submit**.

While that entity is `not_home`, the integration never assigns a weigh-in to the person automatically.

03.4

Delete a person

1. Select **Configure** on **ANKO Body Scale people**, then **Delete a person**.
2. Choose the person and select **Submit**.

> HISTORY IS DELETED
>
> Deleting a person removes their device, entities and stored weights. The weight history cannot be recovered.

The deleted person leaves every scale's list. A scale left with no people holds each weigh-in as a *Pending measurement* until you pick people under **Configure** on that scale.

SEE ALSO

- Add a second scale
- Resolve pending measurements
- How readings are assigned

04

Add a second scale

Add another scale and choose which people it weighs.

Every scale shares the people in the *People entry*. A person weighed on any scale updates the same sensors and the same weight history.

04.1

Add the scale

1. Wake the new scale and go to **Settings > Devices & Services > Add Integration > ANKO Body Scale**.
2. Choose the scale from the **Device** list.
3. In **Who uses this scale?**, select the people and select **Submit**. With one person configured, this

step is skipped.

SELECTION	RESULT
Everyone (default)	Any person can be weighed on the scale, including people you add later.
Some people	The scale matches only those people. People you add later are left out until you select them.
One person	Every weigh-in on the scale goes to that person, with no matching.

04.2

Change who uses a scale

1. Select **Configure** on the scale entry.
2. Change the selection and select **Submit**. `Select at least one person.` appears when the list is empty.

04.3

Actions with several scales

`assign_measurement`, `discard_measurement`, `set_next_person` and `capture_measurement` act on one scale. With more than one scale they need `config_entry_id`, and fail without it:

ERROR

> `More than one scale is configured. Set config_entry_id.`

`reassign_measurement` and `reset_person_history` act on a person and take no `config_entry_id`. A *Pending measurement* stays on the scale that measured it.

SEE ALSO

- Actions
- Manage people

05

Resolve pending measurements

Assign or discard a weigh-in that the integration could not match to one person.

A *Pending measurement* is a *Weigh-in* that fits two people, fits nobody, or arrived when nobody was eligible. The integration publishes no per-person values for it. Each one creates a persistent notification titled `ANKO scale: who was that?` that contains its `pending_id`.

A scale keeps 10 pending measurements at most and drops the oldest first. Each one expires after the pending timeout, 60 minutes by default. The integration checks for expired ones every 5 minutes and at each weigh-in.

05.1

Find the pending ID

Open **Pending measurements** on the scale device. The `pending` attribute lists each one:

ATTRIBUTES OF SENSOR.ANKO_BODY_SCALE_PENDING_MEASUREMENTS

```
pending:
  - pending_id: 3fa91c0b
    weight_kg: 74.3
    timestamp: "2026-11-07T02:40:11+00:00"
    candidates:
      - person_id: jupiter
        distance_kg: 1.2
      - person_id: sarah
        distance_kg: 2.1
```

The `anko_scale_pending_measurement` event carries the same ID.

05.2 Assign it to a person

Run the action from **Developer tools > Actions**:

ASSIGN A PENDING MEASUREMENT

```
action: anko_scale.assign_measurement
data:
  pending_id: 3fa91c0b
  person_id: jupiter
```

The person's sensors update and the weight joins their history. The notification closes.

05.3 Discard a guest's weigh-in

DISCARD A PENDING MEASUREMENT

```
action: anko_scale.discard_measurement
data:
  pending_id: 3fa91c0b
```

Only an administrator can run `discard_measurement`. A discarded weigh-in creates no person.

05.4 Answer from your phone

1. Select **Configure** on **ANKO Body Scale people**, then **Attribution settings**.
2. Type the notify action of your phone in **Notify action**, for example `notify.mobile_app_phone`, and select **Submit**.

Each pending measurement then sends a notification with up to three buttons. `I'm <name>` appears for the two closest people and assigns the weigh-in. `Not me` discards it.

05.5 Add a person from an unknown weight

When two or more people exist and a weigh-in is outside everyone's tolerance, Home Assistant shows the repair **A weigh-in matched nobody**.

1. Open **Settings > System > Repairs** and select the repair.
2. Fill in the person form. **Starting weight** holds the observed weight.
3. Select **Submit**. The integration adds the person and drops the pending weigh-in.

The next weigh-in is the first one published for the new person. A weight that is close to two people creates a pending measurement and no repair.

05.6

Move the last weigh-in to another person

REASSIGN THE LAST WEIGH-IN

```
action: anko_scale.reassign_measurement
data:
  person_id: sarah
```

The previous owner's sensors and history return to their earlier values. The action works on the most recent assigned weigh-in since the last restart.

05.7

Errors

MESSAGE	CAUSE	FIX
`No pending measurement with ID {pending_id}. It may have expired.`	The ID is wrong, or the pending timeout passed.	Read the ID from the sensor again. Raise the pending timeout in Settings.
`No person with ID {person}.`	`person_id` is not a *Profile ID*.	Use the slug of the person's name, for example `jupiter`.
`There is no recent measurement to reassign.`	No weigh-in has been assigned since the restart.	Weigh in again.

SEE ALSO

- How readings are assigned
- Actions
- Events

06

Automate weigh-ins

Trigger an automation on a weigh-in, show the sensors on a dashboard and use the example files.

The `anko_scale_weigh_in` event fires once for each assigned *Weigh-in*. Use it as an automation trigger.

06.1

Notify on a weigh-in

AUTOMATION

```
triggers:
  - trigger: event
    event_type: anko_scale_weigh_in
    event_data:
      person_id: jupiter
actions:
  - action: notify.mobile_app_phone
    data:
      message: "{{ trigger.event.data.person_name }}: {{
      trigger.event.data.weight_kg }} kg"
```

Replace `jupiter` with the person's *Profile ID* and `notify.mobile_app_phone` with your notify action.

06.2 Show the sensors on a dashboard

ENTITIES CARD

```
type: entities
title: Weigh-ins
entities:
  - sensor.jupiter_weight
  - sensor.jupiter_body_fat
  - sensor.sarah_weight
  - sensor.sarah_body_fat
  - sensor.anko_body_scale_pending_measurements
  - select.anko_body_scale_next_person
```

Entity IDs follow the names you chose for the people and the scale.

06.3 Weigh something without a person

`anko_scale.capture_measurement` waits for the next weigh-in and returns it. The weigh-in is assigned to nobody and joins no history.

SCRIPT STEP

```
- action: anko_scale.capture_measurement
  data:
    timeout: 60
  response_variable: reading
- action: notify.mobile_app_phone
  data:
    message: "{{ reading.weight_kg }} kg"
```

The action fails with `No settled weigh-in arrived before the timeout.` when nobody steps on the scale in time.

06.4 Example files

The `examples` folder of the repository holds three files to copy and edit. They use people named Jupiter and Sarah.

FILE	CONTENT
`dashboard.yaml`	A dashboard with one view per person and a scale view. Paste it into the raw configuration editor of a new dashboard.
`weight_loss_leaderboard.yaml`	Statistics sensors for the 30-day weight change and a template sensor that ranks people by kilograms lost and by share of their goal.
`pet_weighing.yaml`	A script that weighs you with the pet, then alone, and stores the difference in a helper. It uses `capture_measurement`.

SEE ALSO

- Events
- Actions

07 Upgrade to 2.0

What the first start of version 2.0.0 changes in an existing installation.

Version 2.0.0 migrates an existing installation on its first start. Entity IDs, entity history and person devices stay the same.

07.1 What the migration does

1. It creates the *People entry*, **ANKO Body Scale people**.
2. It moves each scale's people, weight history and attribution settings into that entry.
3. It changes the unique IDs of person entities from `<address>_<person>_<key>` to `person_<person>_<key>`.

A scale's own entities, such as `sensor.anko_body_scale_weight`, keep their unique IDs. A profile from version 1.0 becomes **Person 1** with the *Profile ID* `legacy`.

A scale entry can show "retrying" for a few seconds during the first start, while the People entry loads.

07.2 Two scales with the same person

If two scales had a person with the same name, the two become one person. The migration keeps the weight history of the first scale, drops the other and logs a warning.

A scale that had every person follows the full list, so people you add later can use it. A scale that had some of the people keeps those people.

07.3 Changed actions

`reassign_measurement` and `reset_person_history` no longer accept `config_entry_id`. Remove the field from automations and scripts that set it.

07.4 Removed sensors

Version 1.3.0 removed the body-composition sensors of the scale device. Dashboards and automations that use `sensor.<scale name>_body_fat` and similar IDs must use the person's sensor, for example `sensor.jupiter_body_fat`.

SEE ALSO

- Add a second scale
- Entities

08 Troubleshoot

Causes and fixes for a scale that is not found, sensors with no value and wrong body-composition values.

08.1 The setup finds no scale

The setup stops with `No ANKO body scales were found on the network.`

The scale sends adverts only while it is awake. Step on it for a moment, then start the setup again. Check that `IF_B7` appears under **Settings > Devices & Services > Bluetooth**. If the Bluetooth adapter is far from the bathroom, put an ESPHome Bluetooth proxy near the scale.

08.2 Person sensors show unknown

No *Weigh-in* has been assigned to the person yet. Check **Pending measurements** on the scale device. If it is above 0, follow Resolve pending measurements. For a new person with no starting weight, press **Weighing in** before the first weigh-in.

08.3 Every weigh-in becomes pending

CAUSE	FIX
Two people are within the margin (2 kg) of each other.	Press **Weighing in** before stepping on, or link a `person` entity to each person.
The person's weight moved more than the tolerance (5 kg or 7 %) since the last weigh-in.	Assign the pending measurement once. The history then follows the new weight.
The scale's people list is empty.	Select people under **Configure** on the scale entry.
Everyone on the scale is `not_home`.	Check the `person` entities, or remove the link under **Edit a person**.

08.4 The scale shows a weight and the person has no body fat

The scale device's **Weight** follows every advert. The person's sensors update once per weigh-in, after the reading settles. Stand still until **Reading stable** turns on.

08.5 The method attribute is always bmi

The body-fat sensors carry a `method` attribute. `bmi` means the impedance route was rejected:

- The raw impedance code is 0: a foot missed a pad. Stand with bare feet on both pad pairs.
- The assumed impedance is outside 325 to 735 ohm. Enable **Impedance (raw)** on the person's device and check that the code is between 127 and 287.
- The person is under 16.

See How body composition is calculated.

08.6

Body composition looks wrong

Check the person's height, sex and birth date under **Configure > Edit a person**. Wet feet lower the impedance and socks block it.

08.7

Values are old after a restart

Sensors restore their last value after a restart and keep it until the next weigh-in. Step on the scale to refresh them.

08.8

A scale entry shows an error

MESSAGE	CAUSE	FIX
`The ANKO people entry is missing. Remove this scale and add it again to recreate it.`	The People entry was deleted.	Delete the scale entry and add the scale again.
`Waiting for the ANKO people entry to take over this scale's people.`	The 2.0 migration is running.	Wait a few seconds. Home Assistant retries.
`The ANKO people entry is not loaded.`	An action ran while the People entry was disabled or failed.	Enable or reload **ANKO Body Scale people**.

SEE ALSO

- Report a problem
- Limitations

09

Report a problem

Collect a debug log and a diagnostics file for a bug report.

Open an issue and fill in the bug report form. It asks for the scale model, the integration version and the debug log.

09.1

Collect the debug log

1. Open **Settings > Devices & Services > ANKO Body Scale**.
2. Choose **Enable debug logging** from the three-dot menu.
3. Repeat the problem.
4. Choose **Disable debug logging**. The browser downloads `home-assistant_anko_scale_<date>.log`.

The log covers only the time between the two clicks. It can include your Bluetooth address, so read the file before you post it.

To keep debug logging on across restarts, add this to `configuration.yaml`:

CONFIGURATION.YAML

```
logger:
  logs:
    custom_components.anko_scale: debug
```

09.2

Download diagnostics

1. Open **Settings > Devices & Services > ANKO Body Scale**.
2. Choose **Download diagnostics** from the three-dot menu of the scale entry.

CONTENT	DETAIL
`integration_version`	Version from `manifest.json`.
`entry`	Entry data and options. Names, birth dates, `person` entities, profile IDs, the notify action and the Bluetooth address are redacted.
`counts`	Adverts, stable adverts, weigh-ins, pending measurements and unknown weights since the entry loaded.
`rssi`	Signal strength of the last advert.
`last_raw_frame`, `recent_frames`	The last advert and the last 30 distinct adverts as hex, with the Bluetooth address redacted.

The diagnostics of the People entry list each person by index with the length of their history and the time of their last weigh-in. The file holds no weight history. `recent_frames` holds the parsed weight of each advert.

SEE ALSO

- Troubleshoot

10

Remove the integration

Delete the scale and people entries, then the integration files.

1. Open **Settings > Devices & Services > ANKO Body Scale**.
2. On each scale entry, open the three-dot menu and select **Delete**. Home Assistant removes the scale device and its entities.
3. Delete **ANKO Body Scale people** the same way. Home Assistant removes every person device and the stored weight history.
4. Remove the integration in HACS, or delete the `custom_components/anko_scale` folder.
5. Restart Home Assistant.

SEE ALSO

- Install

SECTION

Reference

Entities, attributes and configuration options.

11

Entities

Every entity of the scale device and the person devices, with its category, default state and unit.

Each scale is one device. Each *Person* is one device that all scales share.

11.1

Scale device

PLATFORM	ENTITY	CATEGORY	DEFAULT	DESCRIPTION
`sensor`	Weight	none	enabled	Weight of the latest advert, in kg. It changes while the reading settles.
`sensor`	Impedance (raw)	diagnostic	disabled	Raw impedance code of the latest advert, with no unit. Empty when a foot missed a pad.
`sensor`	Last person	none	enabled	Name of the person who got the last weigh-in on this scale.
`sensor`	Pending measurements	diagnostic	enabled	Count of pending measurements. The `pending` attribute lists them.
`binary_sensor`	Reading stable	diagnostic	enabled	On when the scale has finished measuring (status byte `0xA1`).

PLATFORM	ENTITY	CATEGORY	DEFAULT	DESCRIPTION
`select`	Next person	none	enabled	Sends the next weigh-in on this scale to the chosen person. `Automatic` clears it.

The `pending` attribute of **Pending measurements** is a list. Each item has `pending_id`, `weight_kg`, `timestamp` and `candidates`.

11.2 Person device

Each sensor shows the value of the person's last assigned *Weigh-in*. Entity IDs follow the person's name, for example `sensor.jupiter_body_fat`.

PLATFORM	ENTITY	UNIT	CATEGORY	DEFAULT	DESCRIPTION
`sensor`	Weight	kg	none	enabled	Weight of the weigh-in.
`sensor`	Impedance (raw)	none	diagnostic	disabled	Raw impedance code of the weigh-in.
`sensor`	BMI	none	none	enabled	Body mass index.
`sensor`	Body fat	%	none	enabled	Fat mass as a share of weight.
`sensor`	Fat mass	kg	none	enabled	Weight minus lean mass.
`sensor`	Lean mass	kg	none	enabled	Fat-free mass.
`sensor`	Muscle mass	kg	none	enabled	Lean mass minus bone mass.
`sensor`	Body water	%	none	enabled	73.2 % of lean mass, as a share of weight.
`sensor`	Bone mass	kg	none	enabled	5.21 % of lean mass for men, 4.90 % for women.
`sensor`	Basal metabolic rate	kcal/d	none	enabled	Mifflin-St Jeor equation.
`sensor`	Metabolic age	yr	none	enabled	Age shifted by body fat, 15 at least.
`sensor`	Visceral fat index	none	none	enabled	1 to 30, from BMI, age and sex.

PLATFORM	ENTITY	UNIT	CATEGORY	DEFAULT	DESCRIPTION
`sensor`	Ideal weight (min)	kg	diagnostic	enabled	Weight at BMI 18.5 for the person's height.
`sensor`	Ideal weight (max)	kg	diagnostic	enabled	Weight at BMI 24.9 for the person's height.
`button`	Weighing in	none	none	enabled	Sends the next weigh-in to this person, on every scale that allows them, for 60 seconds.

11.3 The method attribute

Every person sensor except **Weight** and **Impedance (raw)** has a `method` attribute.

VALUE	MEANING
`bia_assumed`	Lean mass comes from the impedance equation, with the assumed conversion of the raw code to ohms.
`bmi`	Body fat comes from BMI, age and sex.

How body composition is calculated explains both routes.

11.4 Restored state

Sensors and **Reading stable** restore their last value after a restart. They stay available with that value until the next weigh-in.

SEE ALSO

- Actions
- How body composition is calculated

12 Actions

The six actions of the anko_scale domain, with their fields, defaults and errors.

ACTION	ACTS ON	ADMIN ONLY	EFFECT
`anko_scale.assign_measurement`	scale	no	Assigns a *Pending measurement* to a person and fires `anko_scale_weigh_in` with source `assigned`.
`anko_scale.discard_measurement`	scale	yes	Drops a pending measurement.
`anko_scale.set_next_person`	scale	no	Sends the next weigh-in on the scale to a person, for 60 seconds or one weigh-in.
`anko_scale.capture_measurement`	scale	no	Waits for the next weigh-in and returns it without assigning it.
`anko_scale.reassign_measurement`	person	no	Moves the last assigned weigh-in to another person.
`anko_scale.reset_person_history`	person	yes	Clears a person's stored weights.

12.1

Fields

ACTION	FIELD	REQUIRED	DEFAULT	VALUE
`assign_measurement`	`pending_id`	yes	none	ID from the **Pending measurements** sensor or the pending event.
`assign_measurement`	`person_id`	yes	none	The person's *Profile ID*.
`assign_measurement`	`config_entry_id`	with several scales	the only scale	The scale entry.
`discard_measurement`	`pending_id`	yes	none	ID of the pending measurement.
`discard_measurement`	`config_entry_id`	with several scales	the only scale	The scale entry.
`set_next_person`	`person_id`	no	empty	Profile ID. Empty clears the override.
`set_next_person`	`config_entry_id`	with several scales	the only scale	The scale entry.
`capture_measurement`	`timeout`	no	60	Seconds to wait, 5 to 300.

ACTION	FIELD	REQUIRED	DEFAULT	VALUE
`capture_measurement`	`config_entry_id`	with several scales	the only scale	The scale entry.
`reassign_measurement`	`person_id`	yes	none	Profile ID of the new owner.
`reset_person_history`	`person_id`	yes	none	Profile ID.

12.2 Capture response

`capture_measurement` returns a response and must be called with `response_variable`.

RESPONSE

```
weight_kg: 81.2
timestamp: "2026-11-07T02:40:11+00:00"
```

One capture can wait on a scale at a time.

12.3 Reassign and reset

`reassign_measurement` restores the previous owner's sensor values and history, then publishes the weigh-in for the new person with source `reassigned`. The last weigh-in is held in memory, so the action has nothing to move after a restart.

`reset_person_history` leaves the person's sensors as they are. If the person has a starting weight, it becomes the *Expected weight* again.

12.4 Errors

MESSAGE	CAUSE
`No person with ID {person}.`	`person_id` matches no profile ID.
`No pending measurement with ID {pending_id}. It may have expired.`	The ID is unknown on that scale.
`There is no recent measurement to reassign.`	No weigh-in has been assigned since the restart.
`More than one scale is configured. Set config_entry_id.`	A scale action ran without `config_entry_id`.
`No loaded ANKO scale entry with ID {entry_id}.`	`config_entry_id` names an entry that is not a loaded scale.
`No settled weigh-in arrived before the timeout.`	`capture_measurement` timed out.
`Another capture is already waiting for a weigh-in.`	A second `capture_measurement` started on the same scale.
`The ANKO people entry is not loaded.`	The People entry is disabled or failed to load.

MESSAGE	CAUSE
`No loaded scale allows this person.`	**Weighing in** was pressed for a person who is on no loaded scale.

SEE ALSO

- Events
- Resolve pending measurements

13

Events

The two events the integration fires, with every field of their data.

13.1

anko_scale_weigh_in

Fires once for each *Weigh-in* that is assigned to a person.

EVENT DATA

```
entry_id: 01KV25DSCCM1DXPM40CFMM4H0W
person_id: jupiter
person_name: Jupiter
weight_kg: 81.2
impedance_raw: 165
confidence: 1.0
source: weight_match
timestamp: "2026-11-07T02:40:11+00:00"
```

FIELD	VALUE
`entry_id`	Config entry ID of the scale that measured.
`person_id`	The person's *Profile ID*.
`person_name`	The person's name.
`weight_kg`	Weight in kg.
`impedance_raw`	Raw impedance code, or null when a foot missed a pad.
`confidence`	0 to 1. See below.
`source`	How the person was chosen. See below.
`timestamp`	Time of the weigh-in, ISO 8601.

SOURCE	MEANING
`weight_match`	The weight matched the person's *Expected weight*.
`single_person`	The scale has one person.

SOURCE	MEANING
`forced`	**Weighing in**, **Next person** or `set_next_person` chose the person.
`assigned`	`assign_measurement` or a phone notification assigned a pending measurement.
`reassigned`	`reassign_measurement` moved the weigh-in.

`confidence` is 1.0 for every source except `weight_match` with two or more candidates. There it is the gap between the two closest people divided by twice the margin, capped at 1.0. With the default margin of 2 kg, a gap of 3 kg gives 0.75.

13.2

anko_scale_pending_measurement

Fires when a weigh-in becomes a *Pending measurement*.

EVENT DATA

```
entry_id: 01KV25DSCCM1DXPM40CFMM4H0W
pending_id: 3fa91c0b
weight_kg: 74.3
impedance_raw: 150
candidates:
  - person_id: jupiter
    distance_kg: 1.2
  - person_id: sarah
    distance_kg: 2.1
reason: ambiguous
timestamp: "2026-11-07T02:40:11+00:00"
```

FIELD	VALUE
`entry_id`	Config entry ID of the scale that measured.
`pending_id`	Eight hex characters. Pass it to `assign_measurement` or `discard_measurement`.
`weight_kg`	Weight in kg.
`impedance_raw`	Raw impedance code, or null.
`candidates`	People with an expected weight, closest first, each with `person_id` and `distance_kg`.
`reason`	Why the weigh-in is pending. See below.
`timestamp`	Time of the weigh-in, ISO 8601.

REASON	MEANING
`ambiguous`	The closest person is within tolerance, and the next person is less than the margin further away.
`no_match`	Nobody is within tolerance. With two or more people this also creates a repair.

REASON	MEANING
`no_eligible_person`	The scale's people list is empty, or everyone on it is `not_home`.

SEE ALSO

- Automate weigh-ins
- How readings are assigned

14

Settings

The attribution settings and the fixed timing values, with ranges and defaults.

14.1

Attribution settings

Open **Configure** on **ANKO Body Scale people** and choose **Attribution settings**. The settings apply to every scale.

SETTING	DEFAULT	RANGE	EFFECT
Tolerance (kg)	5	1 to 50, step 0.5	A weigh-in matches a person within this distance of their *Expected weight*.
Tolerance (% of expected weight)	7	1 to 50, step 0.5	The larger of the two tolerances applies.
Margin to runner-up (kg)	2	0 to 20, step 0.5	The closest person must be this much closer than the next person.
Readings kept per person	50	5 to 200	Size of the weight history behind the expected weight.
Pending timeout (minutes)	60	1 to 1440	Life of a *Pending measurement*.
Notify action	empty	text	A notify action such as `notify.mobile_app_phone`. It receives each pending measurement with answer buttons.

For a person with an expected weight of 80 kg, the default tolerance is `max(5, 80 × 7 %)` = 5.6 kg.

14.2

Fixed values

These values are constants in `const.py` and `coordinator.py`.

VALUE	CONSTANT	MEANING
3 adverts	`SETTLE_FRAMES`	Stable adverts that make a weigh-in.
0.5 kg	`SETTLE_SPREAD_KG`	Largest spread between those adverts.
5 s	`SETTLE_WINDOW_S`	Time window for those adverts.
5 s	`SETTLE_TIMEOUT_S`	Silence after which the last stable adverts become the weigh-in.
0.3	`EWMA_ALPHA`	Weight of the newest reading in the expected weight.
60 s	`FORCE_ASSIGN_TIMEOUT_S`	Life of a **Weighing in** or **Next person** override.
10	`MAX_PENDING`	Pending measurements kept per scale.
30	`MAX_RECENT_FRAMES`	Distinct adverts kept for diagnostics.

The person fields and their ranges are in Set up a scale.

SEE ALSO

- How readings are assigned
- Manage people

15

Glossary

Terms used in these pages.

Advert
A Bluetooth Low Energy advertisement. The scale broadcasts adverts while someone stands on it, and Home Assistant reads them without connecting.

BIA
Bioelectrical impedance analysis: an estimate of fat-free mass from the body's resistance to a small current between the feet.

Expected weight
The exponentially weighted average (alpha 0.3) of a person's stored weights. The integration compares each weigh-in with it.

Pending measurement
A weigh-in the integration could not assign to one person. It waits on the scale until you assign it, discard it or it times out.

People entry
The config entry named ANKO Body Scale people. It holds every person, their weight history and the attribution settings, and all scales share it.

Person
A profile with a name, height, sex and a birth date or age. Each person has one device with 14 sensors and a button.

Profile ID
The lowercase slug of a person's name, for example 'jupiter'. Actions and events use it as 'person_id'.

Weigh-in
One settled reading: 3 stable adverts within 0.5 kg and 5 seconds, or the last stable adverts after 5 seconds of silence.

SECTION

Concepts

How the integration works and why.

16

How readings are assigned

The steps that turn the scale's adverts into one weigh-in and choose the person it belongs to.

The scale reports a weight and an impedance code. It does not say who is standing on it, so the integration chooses the person by weight.

16.1

From adverts to a weigh-in

The scale sends adverts while someone stands on it. The settling buffer turns them into one *Weigh-in*:

1. It collects adverts that carry the stable status.
2. Three stable adverts within 0.5 kg and 5 seconds make a weigh-in.
3. If the scale goes quiet for 5 seconds first, the last stable adverts make the weigh-in.

One step on the scale gives one set of values.

16.2

Choosing the person

1. If **Weighing in**, **Next person** or `set_next_person` named a person in the last 60 seconds, that person gets the weigh-in.
2. The integration sets aside people whose `person` entity is `not_home`.
3. If the scale has one person, that person gets the weigh-in.
4. Each remaining person with an *Expected weight* becomes a candidate. The distance is the difference between the weigh-in and the expected weight.
5. The closest candidate wins when two conditions hold. The distance is within the tolerance, `max(5 kg, 7 % of the expected weight)`. The next candidate is at least the margin, 2 kg, further away.
6. Any other weigh-in becomes a *Pending measurement*.

16.3

Worked example

Jupiter has an expected weight of 81.0 kg and Sarah 64.0 kg.

WEIGH-IN	DISTANCE TO JUPITER	DISTANCE TO SARAH	RESULT
80.2 kg	0.8 kg	16.2 kg	Jupiter, source `weight_match`.
72.5 kg	8.5 kg	8.5 kg	Pending, reason `no_match`: Jupiter's tolerance is 5.67 kg and Sarah's is 5 kg.
95.0 kg	14.0 kg	31.0 kg	Pending, reason `no_match`, and a repair to add a person.

With Jupiter at 70.0 kg and Sarah at 68.0 kg, a weigh-in of 69.2 kg is 0.8 kg from Jupiter and 1.2 kg from Sarah. Both are within tolerance and the gap is 0.4 kg, so the weigh-in is pending with reason `ambiguous`.

16.4

Expected weight

The expected weight is an exponentially weighted average of the person's stored weights, oldest first, with alpha 0.3:

UPDATE FOR EACH STORED WEIGHT

```
expected = 0.3 × weight + 0.7 × expected
```

The history holds the last 50 weights. A starting weight is the first item of the history. Every assigned weigh-in joins the history, including forced and assigned ones, so one press of **Weighing in** seeds a new person.

A person with an empty history has no expected weight and is never a candidate.

16.5

Why weight

The advert carries no user slot, and the integration never connects to the scale to read one. Weight is the only signal in the advert that differs between people. The `person` entity filter and the **Weighing in** button cover the cases where weight is not enough.

SEE ALSO

- Settings
- Resolve pending measurements
- Events

17

How body composition is calculated

The equation behind each body-composition sensor, its source and its error.

Weight and the raw impedance code come from the scale. Every other value is computed from them and the person's height, age and sex.

VALUE	METHOD	SOURCE
Lean mass	`13.055 + 0.204·weight + 0.394·height²/impedance - 0.136·age + 8.125·male`	Wu et al., Nutr J 2015;14:52. Foot-to-foot *BIA*, 554 Taiwanese adults aged 16 to 75, DXA reference, standard error 3.17 kg.
Body fat, fat mass	Weight minus lean mass	Derived.
Body fat fallback	`1.20·BMI + 0.23·age - 10.8·male - 5.4`	Deurenberg et al., Br J Nutr 1991;65:105. Standard error 4.1 %.
Body fat fallback, under 16	`1.51·BMI - 0.70·age - 3.6·male + 1.4`	Deurenberg et al. Standard error 4.4 %.
Body water	73.2 % of lean mass	Pace and Rathbun, J Biol Chem 1945.
Basal metabolic rate	`10·weight + 6.25·height - 5·age + 5` for men, `- 161` for women	Mifflin et al., Am J Clin Nutr 1990.
Bone mass	5.21 % of lean mass for men, 4.90 % for women	Heuristic.
Muscle mass	Lean mass minus bone mass	Heuristic.
Metabolic age	Age plus one year per point of body fat above 19 % (men) or 28 % (women), 15 at least	Heuristic. The midpoints follow the American Council on Exercise ranges.
Visceral fat index	`1.5·(BMI - 22) + 0.1·age + 2·male`, limited to 1 to 30	Heuristic.
Ideal weight (min), (max)	BMI 18.5 and 24.9 at the person's height	Derived.

Weight is in kg, height in cm, impedance in ohm, and `male` is 1 for men and 0 for women. No published equation exists for the four heuristic values on a consumer scale. Read them as trends.

17.1

The impedance route

The scale does not report ohms. Byte 13 of its advert is a raw code. The code is 0 without foot contact. It was 117 to 128 for a 61.5 kg man and 165 for a 71.65 kg man in the frames captured so far.

The conversion to ohms is unknown. The integration multiplies the code by 2.56, the scaling that version 1 used. Results from this route carry `method: bia_assumed`. Treat them as provisional until the conversion is calibrated against known resistors.

The integration uses the impedance equation when all three conditions hold:

- The assumed impedance is between 325 and 735 ohm, the range Wu et al. observed. That is a raw code of 127 to 287.
- The person is 16 or older.
- The lean mass is between 50 % and 95 % of the weight.

17.2

The BMI route

When a condition fails, body fat comes from the Deurenberg equation, limited to 3 % to 60 %. The sensors carry `method: bmi`. A weigh-in with no foot contact still records the weight and uses this route.

17.3

Accuracy

Foot-to-foot impedance at one frequency gives an individual body-fat error of about ±5 percentage points against a DXA scan. The BMI route does not see muscle, so it reads high for muscular people. Compare values from the same route over weeks, at the same time of day, with dry bare feet.

SEE ALSO

- Entities
- Advert format
- Limitations

18

Advert format

The byte layout of the scale's manufacturer data and how the parser reads it.

The scale broadcasts its reading in the manufacturer data of each *Advert*, under the local name `IF_B7`. The layout was reverse engineered from the MovingLife app and confirmed against nRF Connect captures. It belongs to the Senssun protocol family.

`parser.py` reads the payload after the 2-byte company ID, which Bleak and ESPHome move into the dictionary key. A payload shorter than 17 bytes is ignored.

BYTES	CONTENT	DECODING
10 to 11	Weight	Big-endian, hundredths of a kg. `0x1FB8` is 81.20 kg.
13	Raw impedance code	0 means no foot contact.
14	Status	`0x01` while measuring, `0xA1` once finished.
15 to 16	Frame counter and checksum	Not read.

An advert with status `0x01` gives a weight, no impedance and `stable: false`. Any other status gives `stable: true` and the impedance code, or no impedance when the code is 0.

The integration is a passive listener (`iot_class: local_push`). It registers a Bluetooth callback for the scale's address and never connects.

SEE ALSO

- How body composition is calculated
- How readings are assigned

19

Limitations

What the integration cannot do, and the reason for each limit.

LIMIT	REASON	WHAT TO DO
People with similar weights produce pending measurements.	Weight is the only signal that tells people apart.	Press **Weighing in** first, or link a `person` entity to each person.
Body fat from impedance is provisional.	The conversion of the raw impedance code to ohms is assumed (2.56 ohm per count).	Follow the trend of one person over weeks.
Socks, shoes or one foot off the pads give `method: bmi`.	The scale reports code 0 without contact on both pad pairs.	Weigh in with bare feet.
One impedance value per weigh-in.	The scale measures foot to foot at one frequency. It cannot measure arms or the trunk separately.	None.
Home Assistant cannot start a weigh-in or read the scale's memory.	The integration listens to adverts and never connects.	Step on the scale.
A weigh-in out of Bluetooth range is lost.	The scale sends adverts only while someone stands on it.	Put a Bluetooth proxy near the scale.
`reassign_measurement` fails after a restart.	The last weigh-in is held in memory.	Reassign before you restart.
One model is supported.	The parser was checked against the ANKO iF1012D (`IF_B7`) only.	Open an issue with a diagnostics file for another Senssun scale.

SEE ALSO

- How body composition is calculated
- Troubleshoot

SECTION

Develop

Build, test and release the integration.

20

Build and test

Run the tests, lint, type checks and the documentation check that CI runs.

Tests run against a pinned Home Assistant version with uv. Python 3.14.2 or newer is required.

CHECK THE CODE

```
uv sync --locked                                # create .venv from uv.lock
uv run pytest
uv run ruff check custom_components tests
uv run ruff format --check custom_components tests
uv run mypy
```

`pyproject.toml` holds the dev dependency group: pytest-homeassistant-custom-component 0.13.367, ruff 0.16.9 and mypy 2.3.1. `uv.lock` pins the full tree.

20.1

Layout

FILE	PURPOSE
`__init__.py`	Entry setup, unload, entry migration and action registration.
`parser.py`	Decodes one advert. No Home Assistant imports.
`attribution.py`	Chooses a person by weight. No Home Assistant imports.
`body_metrics.py`	Body-composition equations. No Home Assistant imports.
`coordinator.py`	One per scale: Bluetooth callback, settling buffer, pending measurements, notifications.
`people.py`	The people hub: profiles, weight history and per-person state.
`migration.py`	Moves people from scale entries to the People entry.

FILE	PURPOSE
`store.py`	Persists history and pending measurements.
`config_flow.py`	Setup, the People menu and the scale's people list.
`repairs.py`	The fix flow of the unknown-weight repair.
`sensor.py`, `binary_sensor.py`, `button.py`, `select.py`	Platforms.
`diagnostics.py`	Diagnostics with redaction.
`strings.json`, `translations/en.json`	All user-facing strings. `tests/test_translations.py` checks that the two match.

20.2 Documentation

The pages are Markdown in `docs-src`. `jupiter-docs` from belikh/docs generates `docs/*.md` and the site from them. Edit `docs-src` only.

CHECK AND BUILD THE PAGES

```
git clone https://github.com/belikh/docs .jupiter-docs
(cd .jupiter-docs && npm ci && npm run update:design)
export JUPITER_DESIGN_DIR="$PWD/.jupiter-docs/vendor/design"
node .jupiter-docs/packages/docs/bin/jupiter-docs.mjs check
node .jupiter-docs/packages/docs/bin/jupiter-docs.mjs build
```

Commit `docs-src` and the regenerated `docs` together. `.jupiter-docs` and `dist` are in `.gitignore`.

20.3 Workflows

WORKFLOW	RUNS ON	CHECKS
`tests.yaml`	push to `main`, pull request	pytest, ruff check, ruff format, mypy.
`docs.yaml`	push to `main` and pull request that touch the pages	`jupiter-docs check`, `build --check` and the site build.
`hassfest.yaml`	push to `main`, pull request, daily	Home Assistant manifest and translation rules.
`hacs.yaml`	push to `main`, pull request, daily	HACS repository rules.
`release.yaml`	tag push	Creates the GitHub release.

SEE ALSO

- Release

21 Release

Tag a version and let the release workflow publish the notes from CHANGELOG.md.

1. Add a `## <version>` section to `CHANGELOG.md`.
2. Set `version` in `custom_components/anko_scale/manifest.json` to the same value.
3. Push a tag named `<version>` or `v<version>`.

TAG 2.0.1

```
git tag v2.0.1
git push origin v2.0.1
```

The workflow `.github/workflows/release.yaml` copies the matching `CHANGELOG.md` section into the release body and attaches `manifest.json`. Without a matching section it generates the notes from the commits.

A tag with a hyphen, such as `2.1.0-rc.1`, publishes a pre-release. Its notes come from the section of the base version, `## 2.1.0`.

docs.jupiter.au reads the pages of the newest release and rebuilds once a day.

SEE ALSO

- Build and test