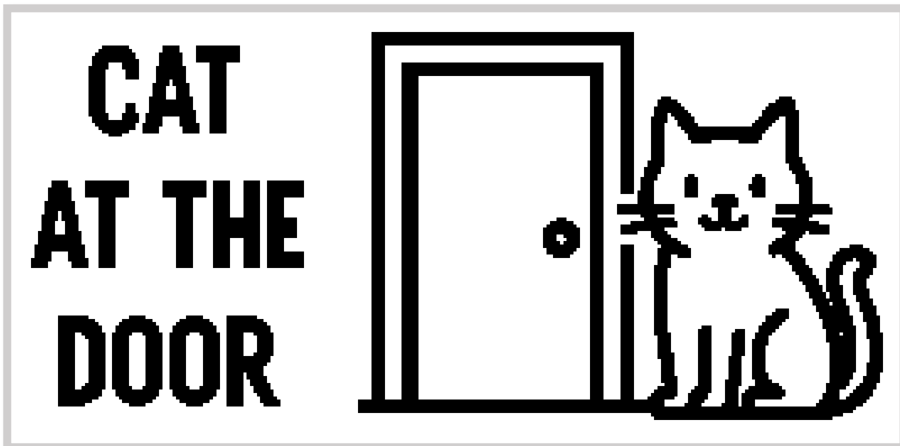
Andrea Favero, Groningen (NL)
09/06/2025 Rev. 0.1

A wireless system, detecting and notifying the presence of your cats at the door



An idea of this system, via a YouTube video: https://youtu.be/0kiuHv76AjQ

This manual is organized in several chapters, divided into 8 main sections:

1) Introduction → Project presentation
2) Supplies → List of necessary parts
3) Make → 3D prints, wiring, assembling, etc
4) Program → Programming the boards
5) Tuning → Adjusting the settings to your case
6) Troubleshooting → An initial guide on what could go wrong, and how to move on
7) How to use it → Explain how to use this system
8) Info → Side information

Summary

1. Project scope

Objective of this system is to notify you when your cat (or cats) is waiting for you in front of the door.

This system is useful in the following situations:

1. You cannot install an automatic cat door (such as those that recognize the cat's microchip).
2. You enjoy letting your cats go outside but you don't like waiting at the door for their return.
3. You'd like to focus on other tasks, and getting notified when it's time to let your cats back inside.

All the above situations represent my case.
Specifically, to the first point, we aren't allowed to modify the door, due to the building's regulations; if an automatic door was allowed, I'm quite sure it would have been our first choice.

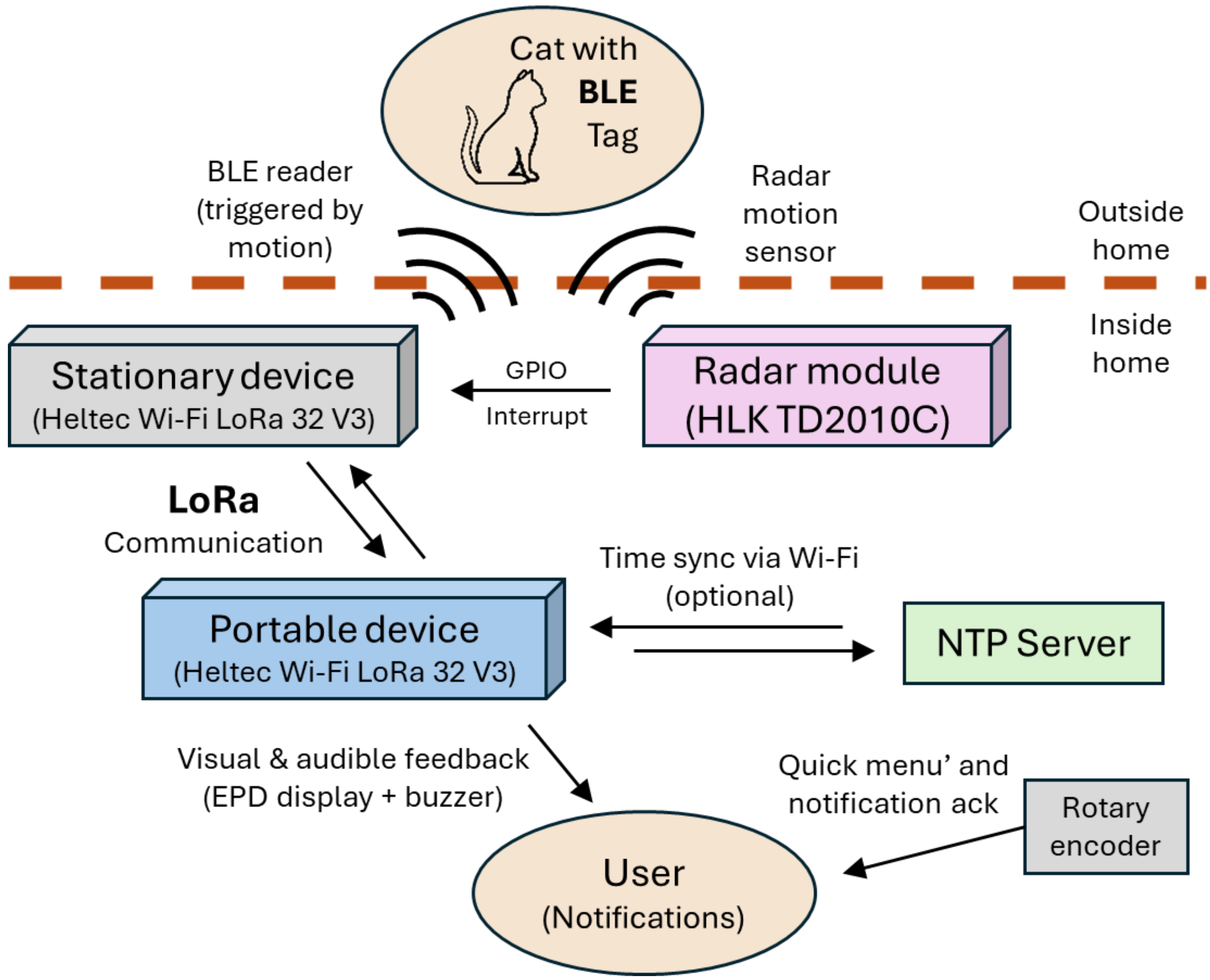
2. Project motivation

Despite the useful reasons of having a device like this, the motivation for this project came the moment my wife asked me "wouldn't be nice to build something more useful than Rubik's cube solver robots, something for our beloved cats?".

I've spent the last few years perfecting the art of building gloriously useless machines—pendulum-balancing robots, Rubik's Cube solvers, a timelapse camera, even a Pi calculator using the Monte Carlo method. So, when my wife suggested I build something useful, it was... a bit of a system shock.

Of course, she was again right …. It was time to build something useful (at least for once 😉).

3. System overview



1. The system notifies you when your cat is waiting in front of the door.
2. It can handle one or more cats.
3. The system is based on two devices:
   - Stationary device: acts as a sensor.
   - Portable device: serves as the user interface.
4. The two devices communicate using LoRa, a **Lo**ng-**Ra**nge wireless technology, ensuring reliable connection even across different concrete floors the house might have.
5. Cats are identified via a BLE (Bluetooth Low Energy) tag, so you are only notified when your cats are waiting.
6. A small BLE tag (e.g. the "Sticker" made by Tile) must be attached to the cat's collar.
7. The stationary device, powered by a phone charger, takes about 800mWh.
8. The portable device, battery operated, uses about 25mAh (by connecting to the stationary every 15 secs).
9. The battery lasts at least 50 hours, likely until 80 hours: This is several days usage for a few hours a day.
10. Wi-Fi is optionally. It "only" updates the date and time.
11. To be noted, the radar is sensitive to rain (false reading). We keep the cats inside when raining 😊

Note:
Apple AirTags don't work for this purpose, due to their anti-tracking feature (MAC address keeps chancing).

4. System workflow

1. **Quick settings (when quick changes needed):**
   - Switch on the portable device while pressing the knob.
   - Enable/disable the buzzer, set the volume, etc.
   - Select Exit and device works with new settings.

2. **Start**:
   - Switch on both the **stationary** and **portable** devices.
   - Stationary device enters idle mode briefly (default: 1 minute) to let cats' exit.
   - Portable device queries the stationary via LoRa (default every 15 seconds.

3. **Detection:**
   - Radar detects motion near the door (e.g. cat returning).
   - If motion is detected, BLE scanning is triggered.

4. **Identification:**
   - BLE tag signal is read.
   - If MAC address matches a registered cat, the system logs presence.

5. **Notification:**
   - If a cat is detected, the portable device displays the name and optionally buzzes.

6. **Sleep Mode:**
   - If left on, portable device enters Deep Sleep after a timeout (default: 150 minutes).

5. Devices' role

Stationary device:

1. Acts as a sensor.
2. Must be powered using a USB-C phone charger (500mA is sufficient).
3. Can be turned on or off using the built-in switch.
4. Should be fixed to the inside of your door, or wherever your cat usually waits.
5. It should be placed no more than 20 cm above the floor.
6. Some testing is required to find the right location: It must be checked whether the radar signal can pass through your door and detect the cat. In my case I placed on the fix frame just at the door's side.
7. The radar module must be configured using the HLKRadarTool app (available for Android and iOS); The app connects to the radar module via Bluetooth.

8. The stationary device functions as a responder—it is periodically queried by the portable device.
9. Settings are stored in a text file. For example, the association between cats' names and MAC addresses is one of the necessary configurations.
10. It's necessary to apply layers of aluminium tape in front of the device, preventing the radar from sensing the cat's motion inside the room.

Portable device:

1. Serves as user interface.
2. Can be turned on or off using the built-in switch.
3. It's battery-operated for easy portability.
4. The battery is recharged through the device, when connected to a USB-C phone charger.
5. It functions as the controller—it periodically queries the stationary device.
6. Settings are stored in a text file, like the stationary device.
7. Some of the settings can be adjusted using a rotary encoder with a push-button.

6. Safety aspects

A few safety-related aspects that must be considered:

1. Power the devices only with a power supply that provides Class 2 insulation. Phone chargers typically meet this requirement—but check to be sure, for your own safety.
2. The portable device includes a LiPo battery. Be careful not to damage the battery during assembly and keep the device at a safe distance whenever the battery is charging.
3. Adding a collar to cats implies some risk for them to get stuck while exercising their acrobat's skills.
4. The radar has a declared transmission power of 12dB (as reference, mobile phones >30dB). At this power, no thermal or non-thermal biological effects are expected, even at very close distance.
5. The LoRa transmission has a max declared power of 22dB; The low power, combined with the very low duty-cycle (< 1%), is far below any threshold of concern.
6. When cutting the battery's wires, cut them one at the time to prevent short circuiting the battery.

Said that, if you build and use the CAT AT THE DOOR system based on this guide, you do so at your own risk.

7. Legal aspects

This system is based on LoRa wireless communication protocol.

The legal license-free frequencies, for the LoRa transmissions, are Country regulated: The correct frequency must be set (see System's settings chapter).

In addition, the hourly average ToA (time-on-Air) might be limited where you live; For instance, I live in The Netherlands and the duty cycle must be below 1% (duty cycle = ToA (s) / 3600). This means the ToA must be up 36 seconds maximum per each device.
For 2 cats, with the default settings, the duty cycle is about 0.5%; Adding more cats is possible, and up to 4 cats the duty-cycle remains within the 1% with the default settings.
The code compares the resulting duty-cycle with the one set in the settings file (config.json) based on the number of cats and LoRa settings. In case the duty-cycle threshold is crossed, the period in between devices' connection is increased to respect the duty-cycle.

Said that, if you build and use the CAT AT THE DOOR system based on this guide, you do so at your own risk.

8. Checks before spending…

1) If you think the radar positioning might be challenging for your specific case, you might consider buying only the radar and run some tests.
   The radar response can be verified via a freeware smartphone application (HLKRadarTool).
   For the tests, you can refer to the chapters *Stationary device installation* and *Radar's tuning* later in this manual.

2) The chosen BLE tag is the smallest I could find, yet it might be quite bulky on small-size cats.
   The tag is 2.8cm wide and 0.8cm of thickness; The tag sleeve for the collar adds to it.
   As reference, some pictures of our cat Freya (ca 5Kg) with the collar on, at the end of this manual.
   Please note, most of the pictures published by the shops, for cats with these collars, are photo-collage.

9. Project Quick Start Guide

1. **Purchase the material**
   - Start with the radar in case of doubts.
2. **Attach BLE Tag**
   - Use a Tile Sticker BLE tag on your cat's collar.
   - Activate it via the Tile app (no subscription needed).
   - Retrieve its MAC address with the free "nRF Connect" app.
3. **Flash the Firmware**
   - Download MicroPython firmware for ESP32-S3 from https://micropython.org.
   - Flash it by dragging the .bin file onto the board (enter boot mode by holding BOOT while connecting).
4. **Download & Upload Code**
   - Get the files from GitHub: GitHub/cat_at_the_door
   - Use Thonny to upload the correct files to each device.
5. **Edit Configs**
   - ble_tags.json: Add your cat's name and BLE tag MAC address.
   - config.json: Set Wi-Fi, timezone, buzzer, radar timings, etc.
6. **Assemble Devices**
   - Print 3D cases and wire components as per manual.
   - Install the stationary device low and near the door. Shield with aluminium tape as instructed.
7. **Tune Radar**
   - Use HLKRadarTool App to configure radar thresholds and range.
8. **Power On & Test**
   - Turn on both devices after cats' leave.
   - Wait for radar + BLE detection.
   - Watch the display and buzzer confirm your cat's return.

10. Difficulty level

The difficulty of this project is overall Intermediate, with some Difficult aspects related to the LoRa settings, the radar placement and its tuning.

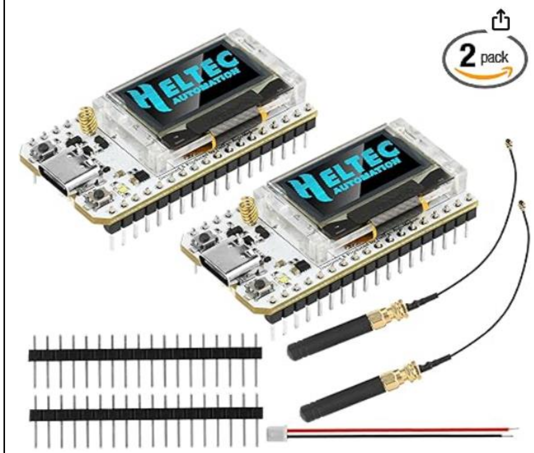
Be aware that the system may not work perfectly right after assembly:

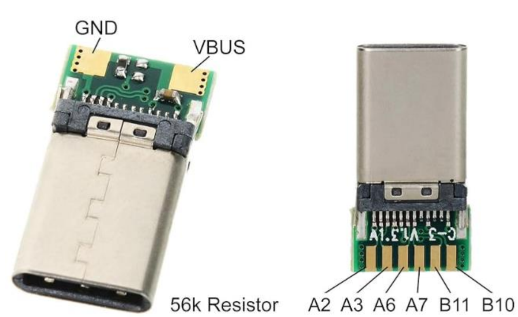
1. **Radar tuning is necessary**: each installation is unique.
2. **System configuration is needed**: each BLE tag has a different MAC address, so your cats.

I suggest to the interested makers, before anything else, to read thoroughly the present "How to" manual, then, after a cool-down time, to read it a second time…

… and now, let's start!

11. Supplies

Q. ty	Part	link to known shops	Cost	Notes
2	Heltec Wi-Fi LoRa 32 V3 boards. This must include the antenna and the cable with JST connector for the battery (plastic case not needed)	https://heltec.org/project/Wi-Fi-lora-32-v3/ or https://tinyurl.com/ydy8avte	~48 euro ~40 $ (2 sets)	
1x cat	BLE tag “Tile Sticker” (by Life360)	https://tinyurl.com/4t33j4kt	~ 30 euro ~ 25 $ (each)	
1x cat	Collar with Tag pocket Personalization (name, phone) from some shops	Amazon.com: https://tinyurl.com/3kru48r6 Temu: https://tinyurl.com/478d9dpn	~ 8 $ (each) ~ 4euro	
1	2in9V2 EPD Waveshare display (Black and White)	https://www.waveshare.com/product/2.9inch-e-paper-module.htm	~ 19 $	
1	LiPo battery 3.7V 2200mAh LP773575LC-PCM (35.5 x 77 x 8.4mm)	https://tinyurl.com/bdhr5ktu	~ 12 euro	

Q. ty	Part	link to known shops	Cost	Notes
1	HLK TD2010C radar	https://tinyurl.com/437httdj	~ 9 $ (each)	
1	Encoder with push button	https://tinyurl.com/36vhmfsk	~ 8 $ (2 pcs)	
1	Buzzer (DC 3.3-5V with transistor)	https://tinyurl.com/phcc4y4a	~ 8 $ (5 pcs)	
2	USB-C female, panel mounting	https://tinyurl.com/29fzs287	~ 11 $ (5 pcs)	
2	USB-C connector (With: D+ D- or with: A6 A7)	https://tinyurl.com/yzrfzus6	~ 6 $ (5 pcs)	
2	Mini Rocker Switch (slot 13.2 x 9 mm)	https://tinyurl.com/3saxe7m4	~ 3 $ (each)	

Additional material, you might already have available

Electronic and electrical small parts:

Q. ty	Part	Notes
1	90 deg male header 	For the header J3 of the Heltec board of portable device
Some	Dupont wires with female connector	For the Encoder and the Buzzer
1	Long USB-C cable	• The length depends on the distance of the stationary device from a power supply socket. • Alternatively, a power bank could be used; Be noted the stationary device consumes ~160mAh

Screws:

Quantity	Dimension	Head type	Note
~ 10	M3x10	Conical	
~ 10	M3x12	Conical	
1	M4x12	Conical	
4	M2.5 x 10		Bolt and nuts should be included with the EPD

Of course, some other common materials are needed (wires, solder, solder device, tire wraps, self-adhesive rubber feet, masking tape, etc).

Filament: Requested about 220g of filament for the 3D printed parts.

Power supply:

One phone charger to energize the stationary device.
The stationary device uses up to 250mA, suggesting a phone charger capable of 500mA or more.
The related cable must have a USB-C connector (data lines are not needed, only power lines).
In case routing the cable isn't an easy task, a power bank could be used instead; Be noted the stationary device consumes ~160mAh

To recharge the battery of the portable device, a phone charger with at least 500mA is requested.
Also in this case, the related cable must end with a USB-C connector (data lines are not needed, only power lines).

Aluminium tape:

The aluminium tape is used to block the radar from sensing into the room.
This prevents it from detecting motion inside—such as you walking past the device.

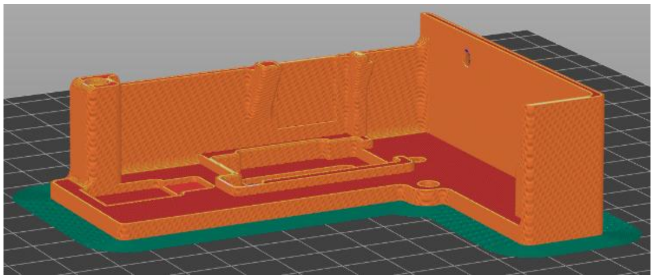
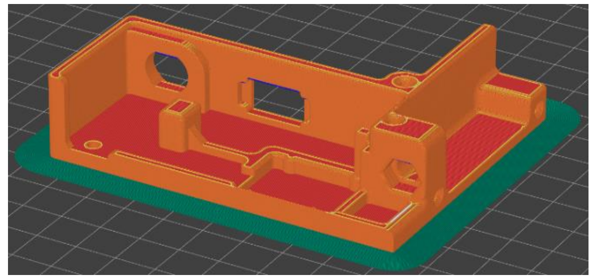
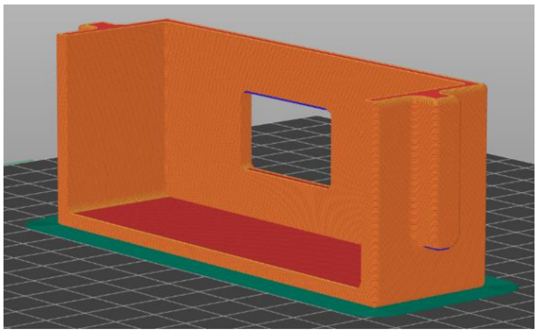
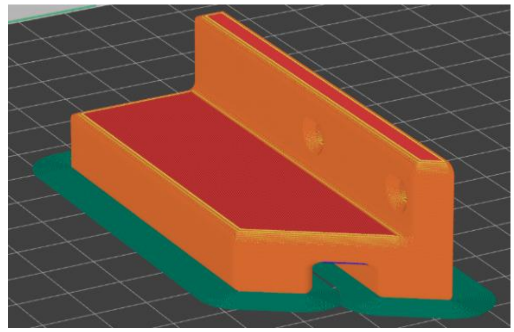
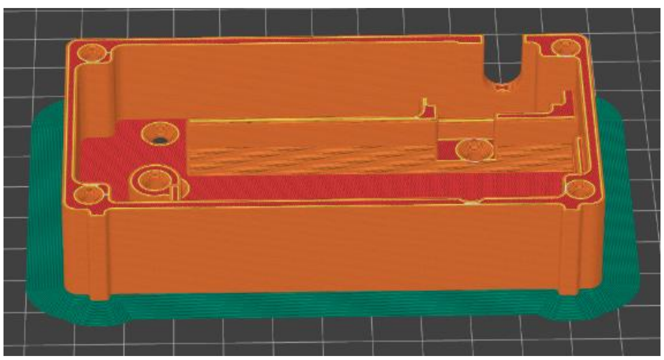
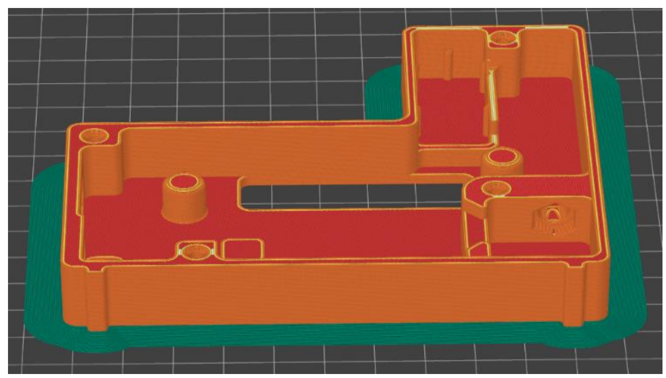
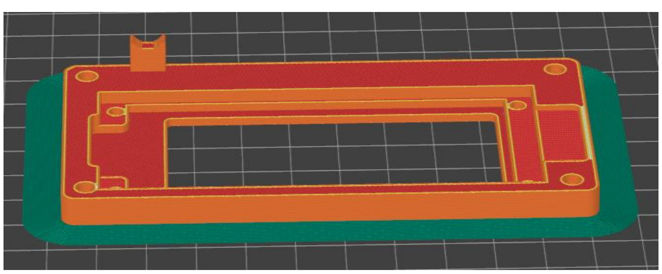
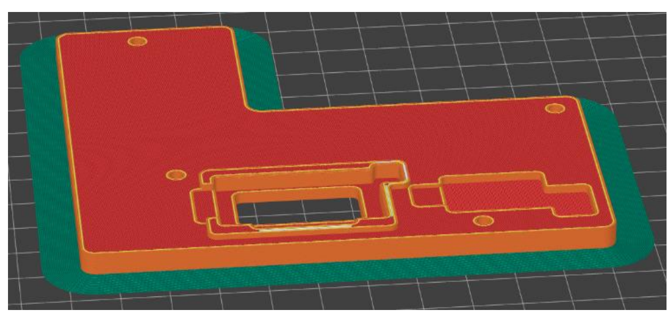
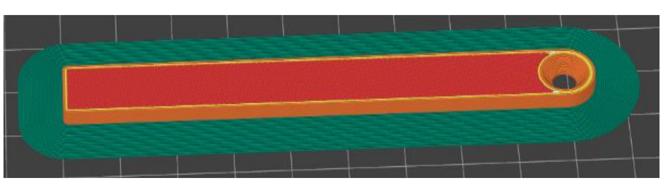
12. 3D printed parts

Device	Ref	Part	Filament Grams	Notes
Stationary	1	Case_stationary	23	
	2	Cover_stationary	21	
	3	Over_cover	30	
	4	Support_L	20	
	5	Support_R	20	
Portable	6	Battery_holder	3	
	7	Case_Board_portable	39	
	8	Case_EPD_portable	36	
	9	Cover_ Board_portable	9	
	10	Cover_EPD_portable	14	

Notes:

1. Filament weight is based on PETG density (1.23g/mm^3).
2. Printing settings I've used on my Ender 3 printer, for accurate result:
    1. 0.2mm layers
    2. Low speed (between 25 to 40mm/s for the external parts and 1st layer)
    3. 2 layers on vertical walls
    4. 2 layers on horizontal walls
    5. 30% filling
    6. 8mm brim
3. No support needed: All parts have been designed to be printed without supporting the overhangs.
4. The Case for the portable device has been split in two parts, for easier and better 3D printing.
5. The suggested part orientation for the 3D print is showed on below table.
6. The **stl** files are available at: https://github.com/AndreaFavero71/cat_at_the_door/tree/main/stl
7. The **step** files are only available at: https://github.com/AndreaFavero71/cat_at_the_door/tree/main/stp

Parts orientation during 3D printing:

Case_stationary	Cover_stationary
	
Over_cover	Support_L and Support_R
	
Case_EPD_portable	Case_Board_portable
	
Cover_EPD_portable	Cover_Board_portable
	
Battery_stopper	
	

13. Wiring

GPIO / Pins to connect

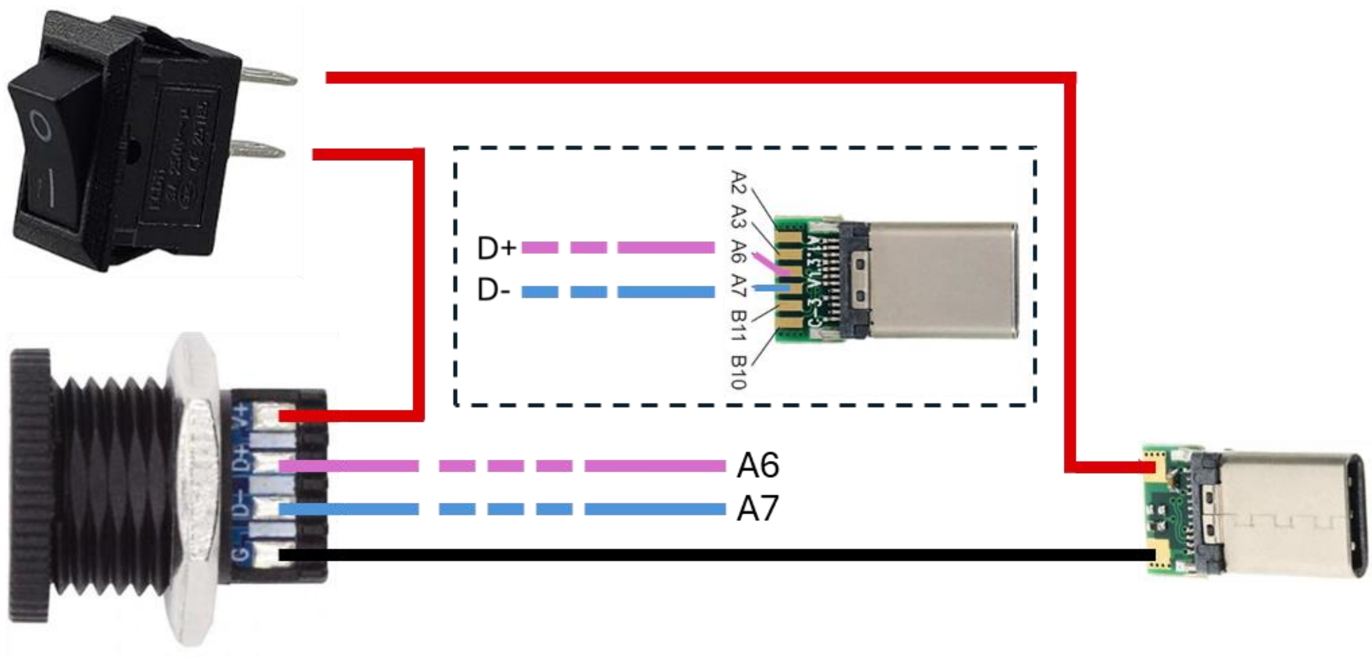
Stationary device:

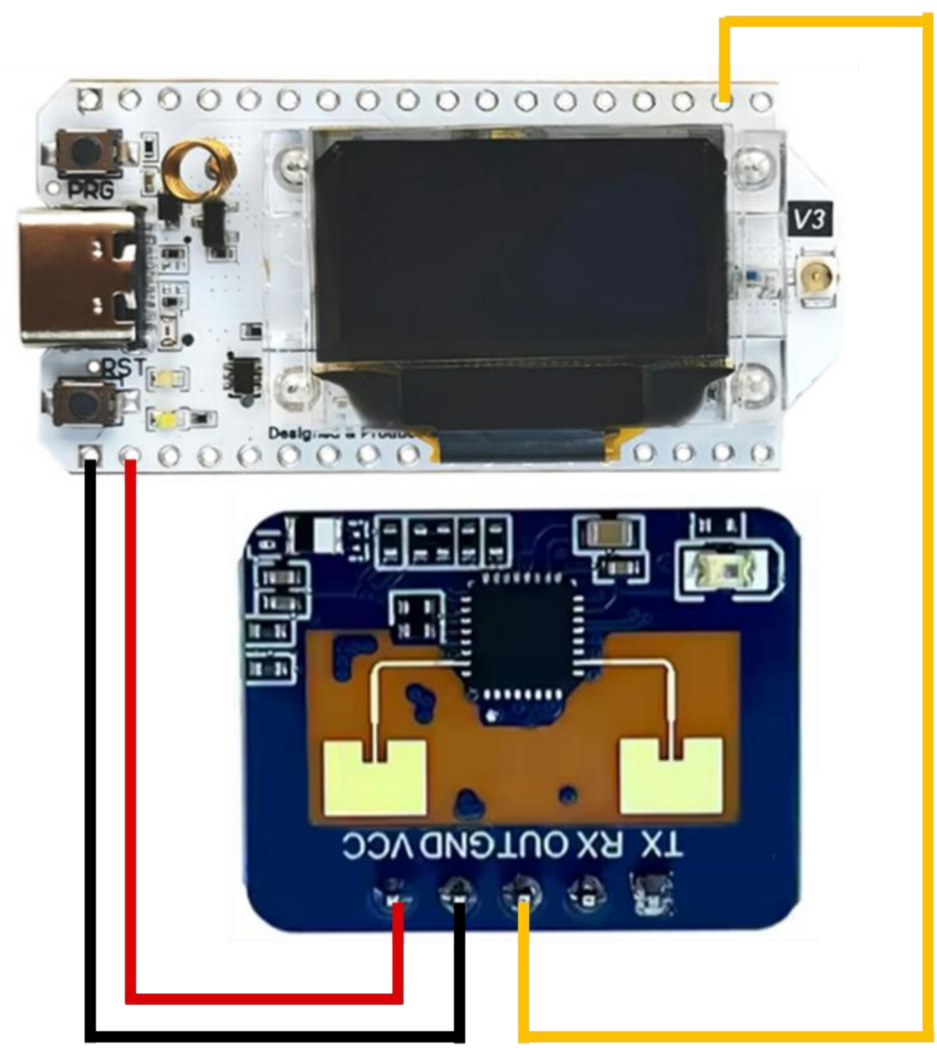
Header	Pin	Label/GPIO	Purpose
J2	1	GND	Ground connection for the radar module
J2	2	5V	+5V dc connection for the radar module
J3	17	GPIO6	Radar signal

Portable device:

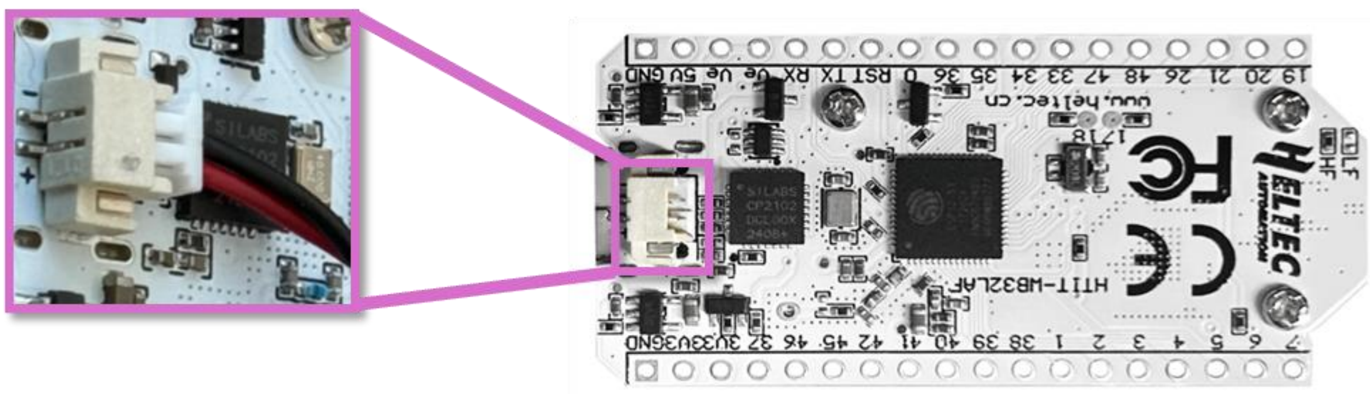
Header	Pin	Label/GPIO	Purpose
J3	1	GND	Ground connection for the EPD display, Encoder, Buzzer
J3	2	3V3	+3.3V dc connection for the EPD display
J3	3	3V3	+3.3V dc connection for the Encoder and buzzer
J3	5	GPIO46	EPD display Mosi pin
J3	6	GPIO45	EPD display Clock pin
J3	7	GPIO42	EPD display CS pin
J3	8	GPIO41	EPD display DC pin
J3	9	GPIO40	EPD display Reset pin
J3	10	GPIO39	EPD display Busy pin
J3	13	GPIO2	Encoder CLK pin
J3	14	GPIO3	Encoder DT pin
J3	15	GPIO4	Encoder push-button pin
J3	16	GPIO5	Buzzer control

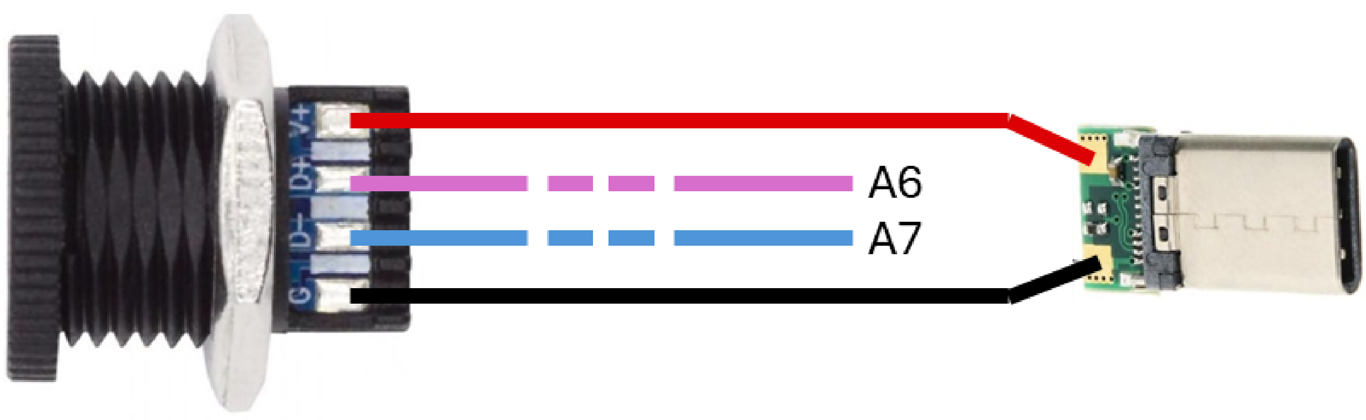
Stationary device, connections sketches:

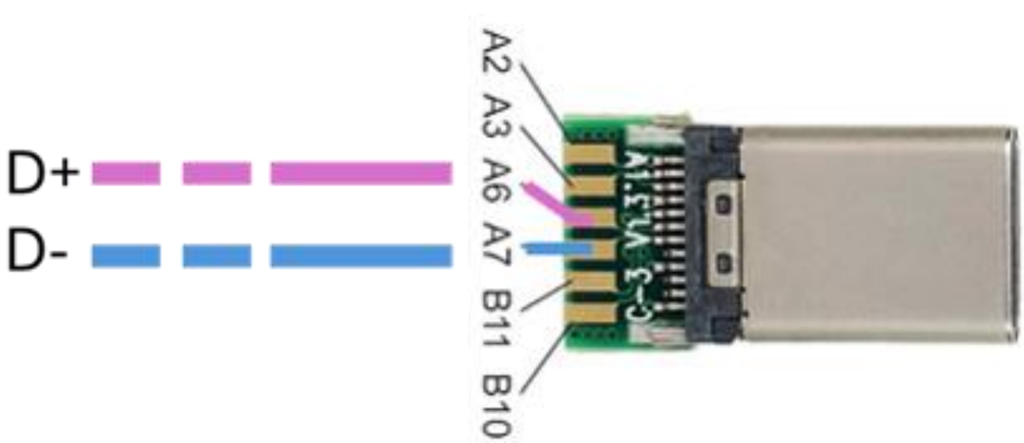




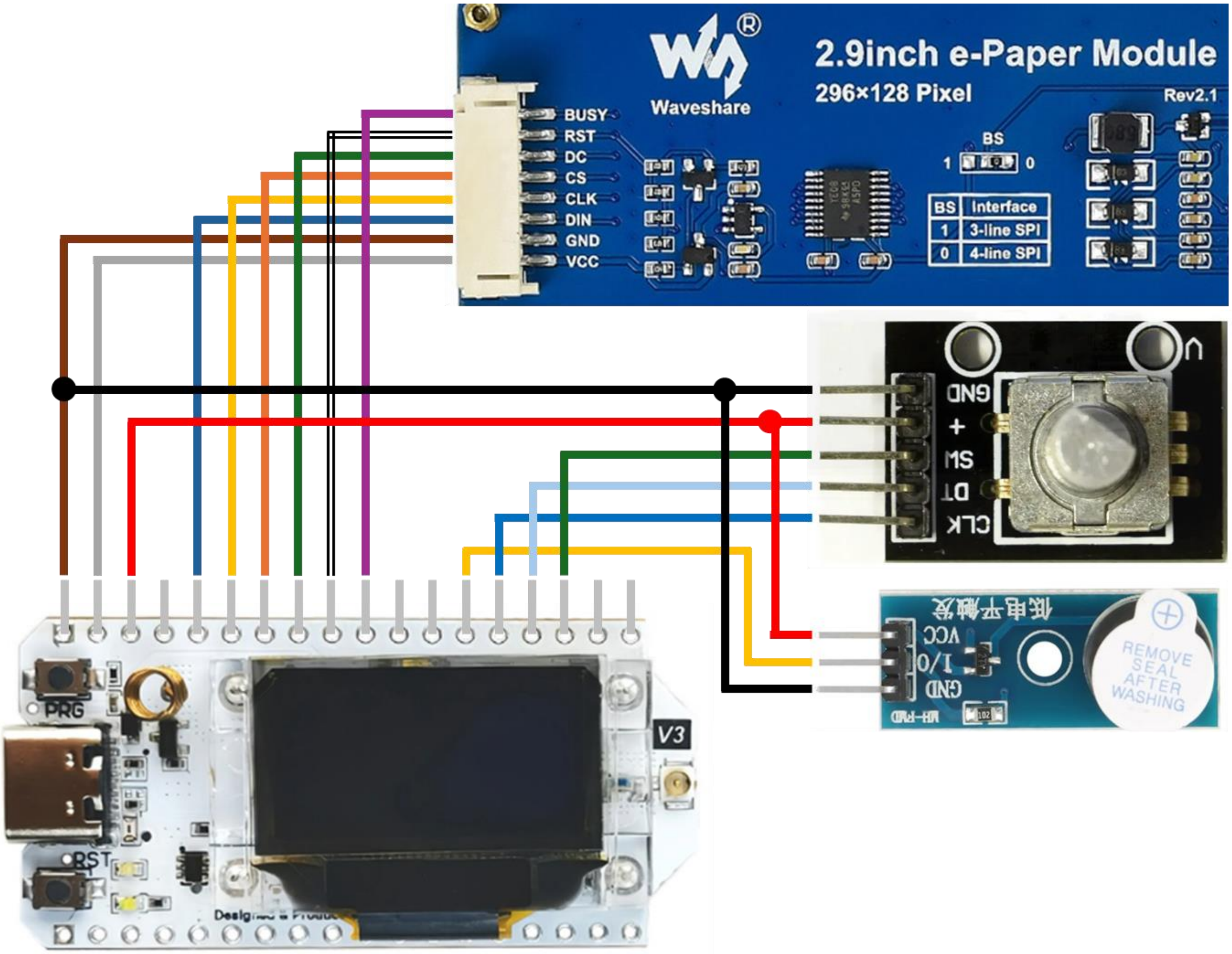
Portable device, connections sketches (1/2):







Portable device, connections sketches (2/2):



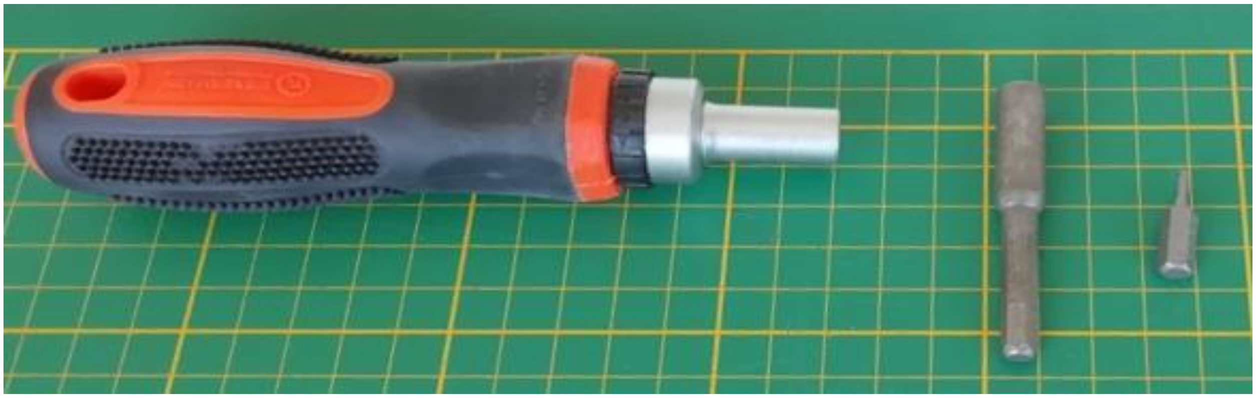
Notes:

1. **SAFETY:** When preparing the battery wiring, **cut one cable at the time** to prevent battery's short circuit.
2. For both the stationary and the portable device, it's necessary to connect the Heltec's board USB-C to the USB-C female panel socket. This serves as power supply for the stationary device, and as battery charger for the portable device.
3. Soldering the USB-C data lines (D+ <- -> A6, D- <- -> A7), between the USB-C socket and USB-C connector, is strongly suggested; It makes possible to reprogram the board, or just change some settings, without opening the devices 😊

14. Assembly order

Necessary tools:

- Allen key 2mm and 2.5mm
- A small screwdriver
- Plier with long and narrow tips



Stationary device assembly steps:

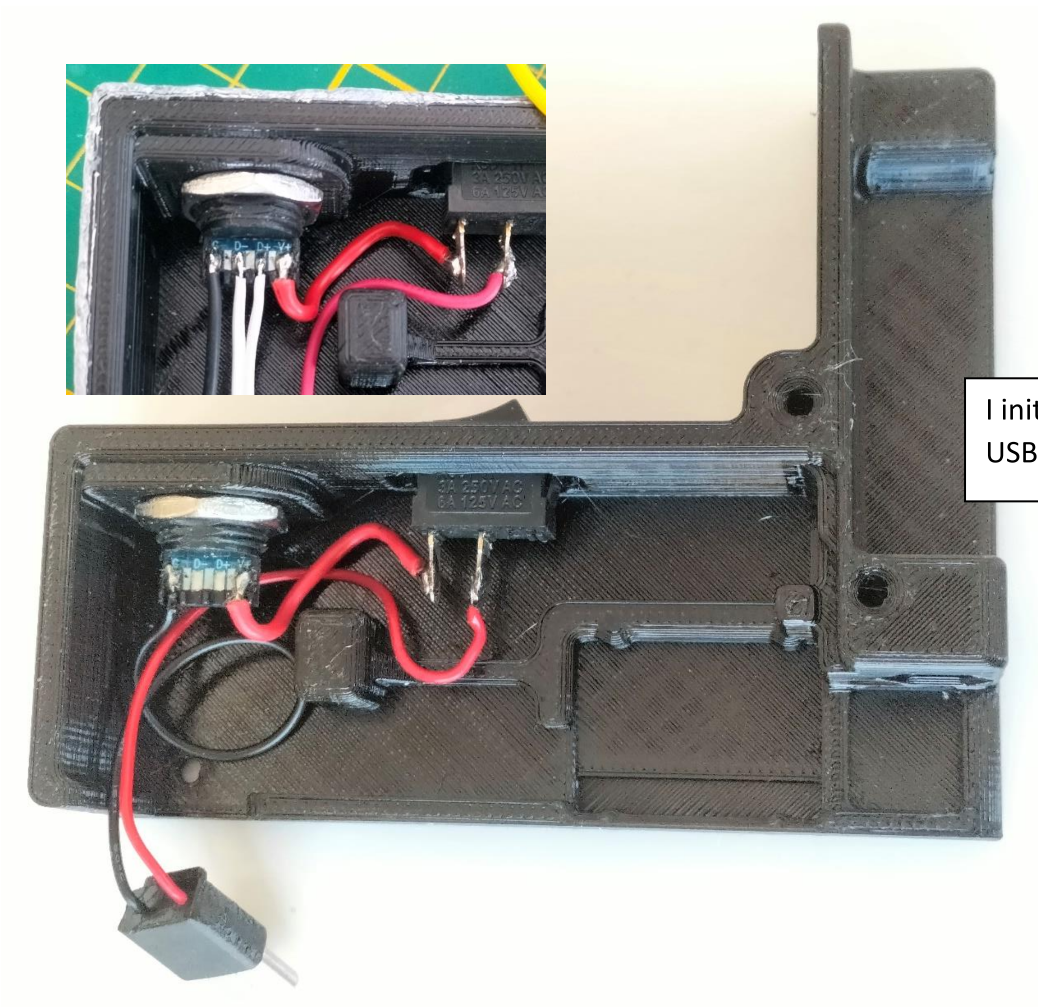
1. A couple of checks before the real assembly:
    a. Ensure the USC-C female port enters the slot; If not, clean the slot with a little file.
    b. Ensure the antenna enters the slot; If not, clean the slot with a little file.
    c. Ensure the display's plastic frame enters in the cover slot; If not, clean the slot.
2. Insert the USB-C female socket through the case and close the nut with a narrow plier.
3. Unscrews the antenna from the base and remove the washer and the nut (not used).
4. Insert the antenna through the hole, align the hexagon shape to the slot, and push it in.
5. Screw the antenna to its base (do not add the washer and the nut!).
6. Insert the switch.
7. Solder a red wire from the USB-C female **V+** pad to the switch.
8. Solder the red wire of the USB-C male connector to the other switch terminal.
9. Solder the black wire of the USB-C male connector to the **G** pad of USB-C female.
10. Position the Heltec board into the case.
11. Connect the USB-C connector.
12. Connect the antenna.
13. Ensures the antenna's cable being away from the screws' holes.
14. Position the cover, very gently.
15. Screw the cover to the case (# 4 screws M3x10 with conical head).
16. Add several layers of aluminium tape (ca 1mm thickness, about 12~16 layers) all around, **apart** the rear part and the top 30mm of the antenna's cover; Remove the aluminium tape from the Cover's hole (white led).

Portable device assembly steps:

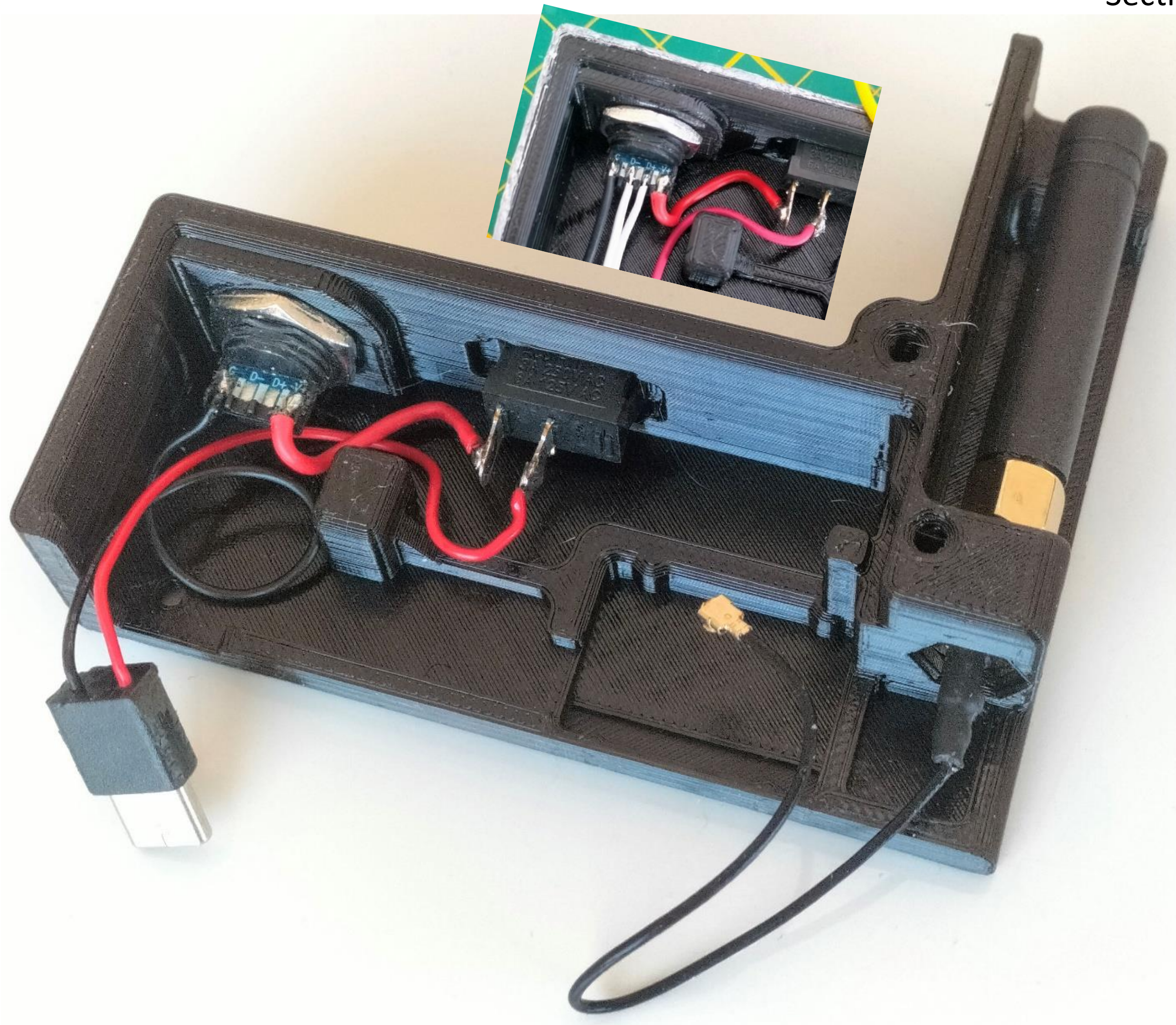
1. A couple of checks before the real assembly:
    a. Ensure the USC-C female port enters the slot; If not, clean the slot with a little file.
    b. Ensure the antenna enters the slot; If not, clean the slot with a little file.
    c. Ensure the Heltec display's plastic frame fits into the cover slot; If not, clean the slot.
    d. Ensure the little holes for the e-ink display screws are opened.
    e. Ensure the encoder board slides into the intended slot.
2. Solder the 90˚header pins to the Heltec J3 header.
3. Couple the two cases together (4 x M3x12mm screws with conical head)
4. Insert the USB-C female port to the case and close the nut with a narrow plier.
5. Insert the switch though the case cut-out.
6. Unscrews the antenna from its base, remove the washer and the nut (both not used).
7. Insert the antenna through the hole, align the hexagon shape to the slot, and push it in.
8. Screw the antenna to its base (do not add the washer and the nut!).
9. Position the battery into its slot, by passing the wires through the cases' hole.
10. Fix the battery via the Battery_stopper.
11. Solder a 5cm long red wire to the red cable from SH 1.25x2P male connector and to the switch terminal.
12. Cut the battery red wire close to its connector, and solder it to the other switch terminal. Safety: Do not cut the battery red wire together with the battery black wire, as it will short circuit the battery.
13. Cut the battery black wire close to its connector. Safety: Do not cut the battery red wire together with the battery black wire, as it will short circuit the battery.
14. Solder the black wire of the battery to a 5cm black wire, and to the black cable from SH 1.25x2P male.
15. Connect the encoder with the 5 Dupont wires. Suggested to wrap the 5 connectors with a couple of tape's loops, to prevent the detachment of a single connector.
16. Slide the encoder into the intended slot; Add the washer and close the nut. Pass the encoder wires through the cases' hole.
17. Place the EPD display into the Cover_EPD_portable recess. Secured it with the 4 provided M2.5 bolts.
18. Pass the EPD cables through the cases' hole
19. Fix the Cover_EPD_portable to the case (4 screws M3x10 with conical head).
20. Insert the Buzzer into the case; There are a few ribs with some little interference to keep the board in place.
21. Connect the SH 1.25x2P male connector to the Lithium battery connector of the Heltec's board.
22. Position the Heltec board into the case.
23. Connect the Encoder, EPD and Buzzer cables to the Heltec board.
24. Connect the USB-C connector to the one of Heltec's board.
25. Connect the antenna's cable to the LoRa IPEX connector of the Heltec's board.
26. Ensures the antenna's cable non passing over the holes for the screws.
27. Position the Cover_Board_portable, very gently.
28. Screw the Cover_Board_portable to the case (4 screws M3x10 with conical head).

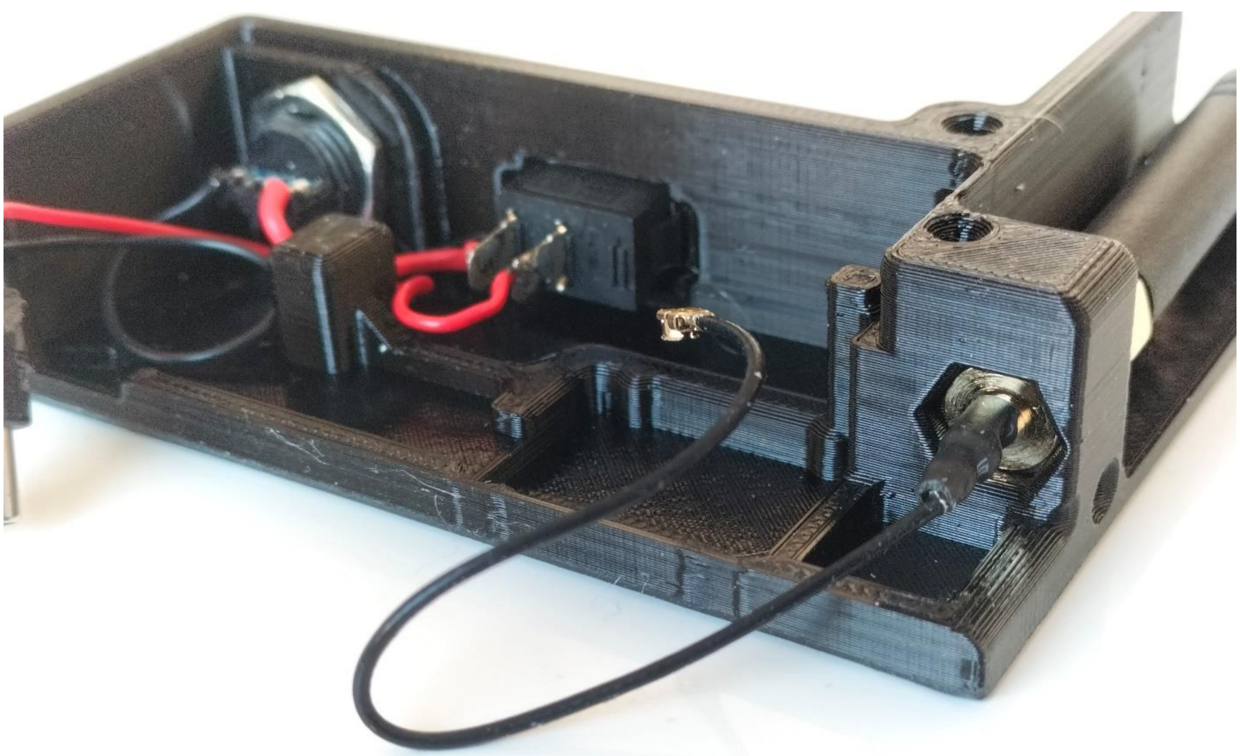
15. Assembly, step by step

Stationary device assembly details:



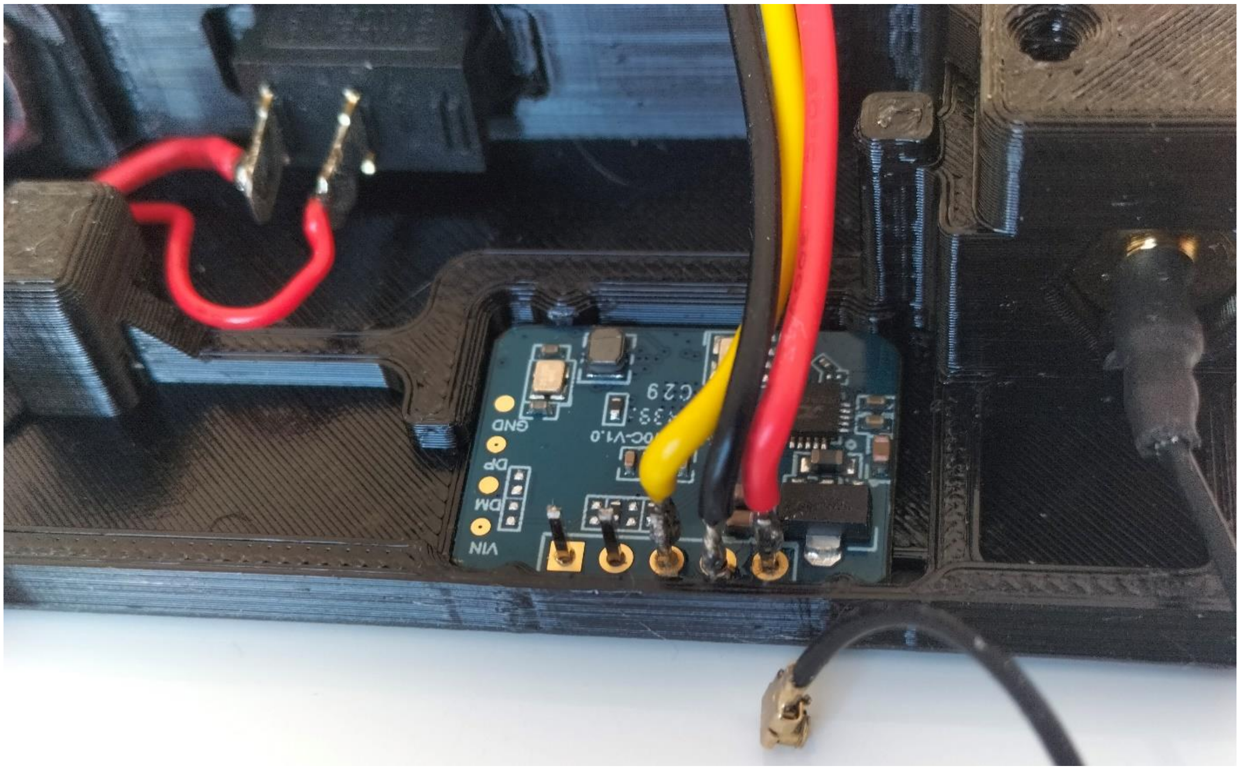
I initially forgot the USB-C data lines…



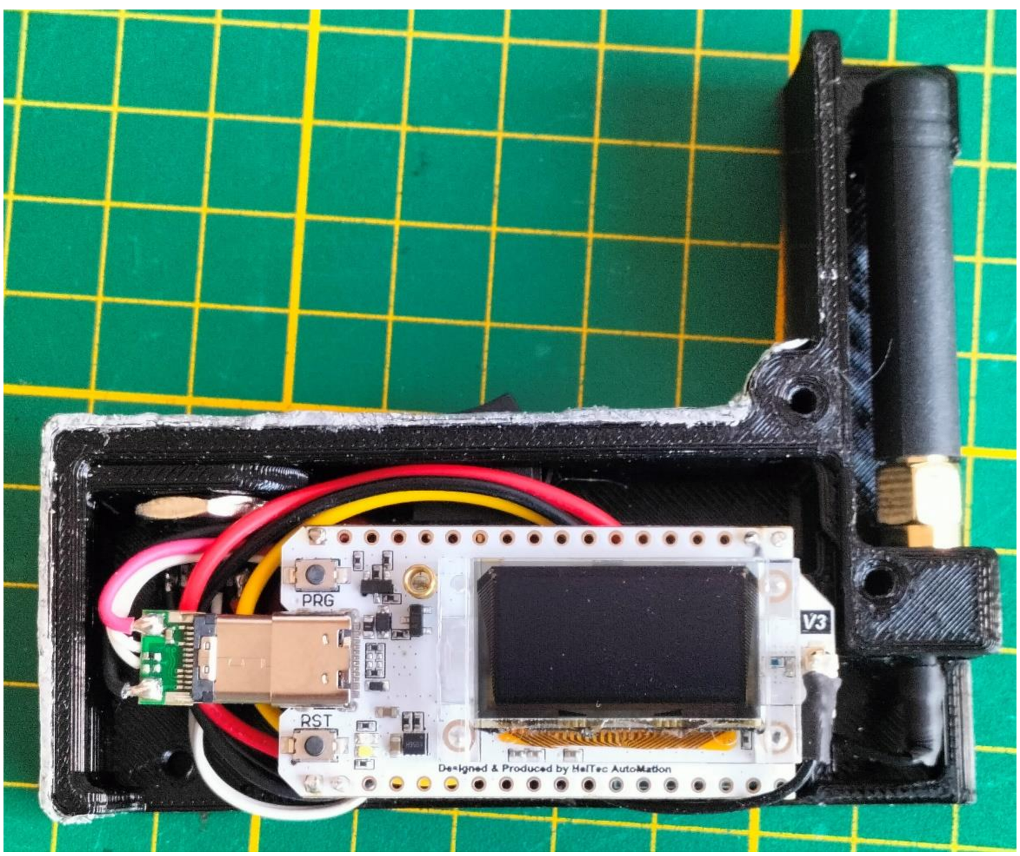


If you prefer soldering the wire directly to the header pins, those must be first shortened.

Gently press the board into the slot.

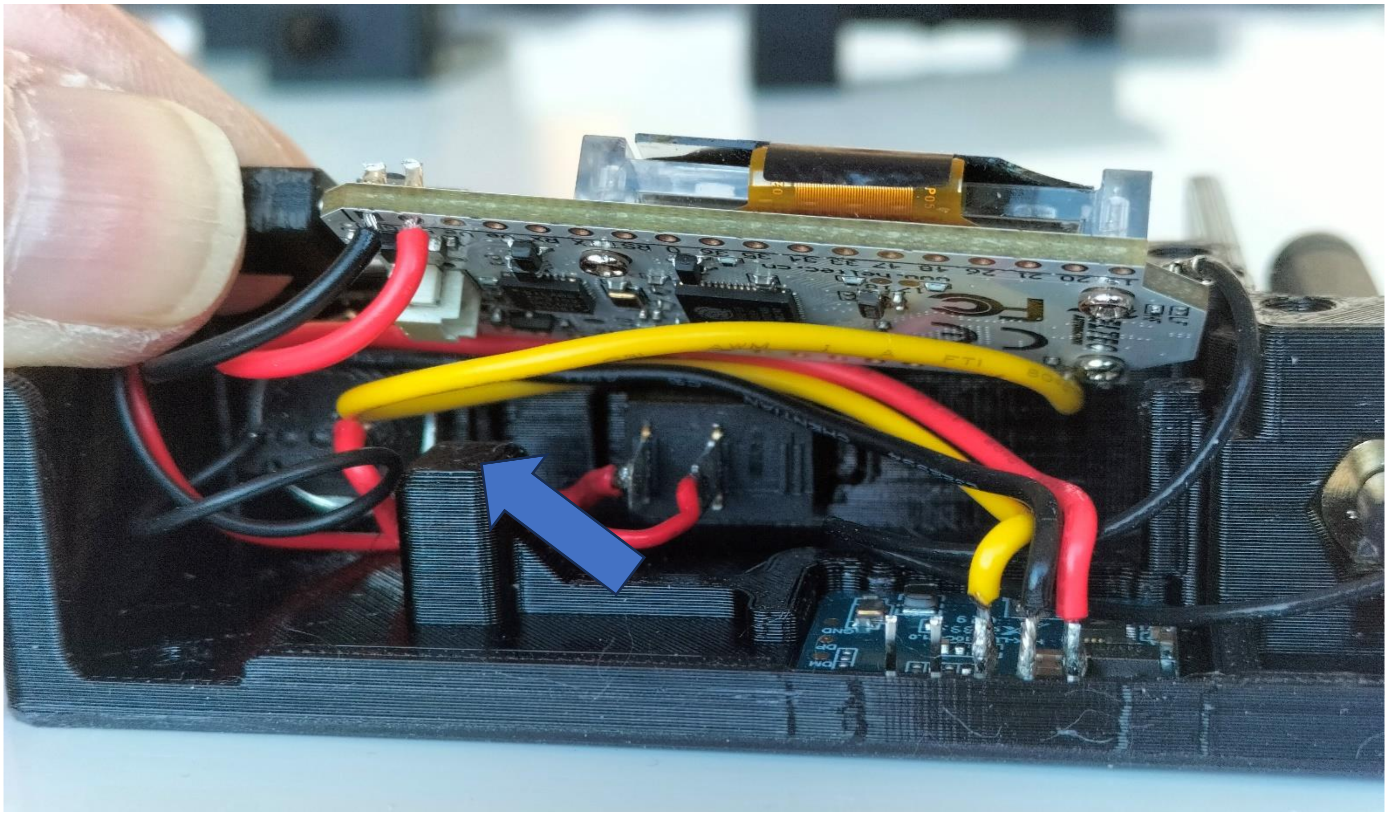


Connect the USB-C connector to the Heltec's board; Make sure the antenna's cable being out of the way from the screws' holes

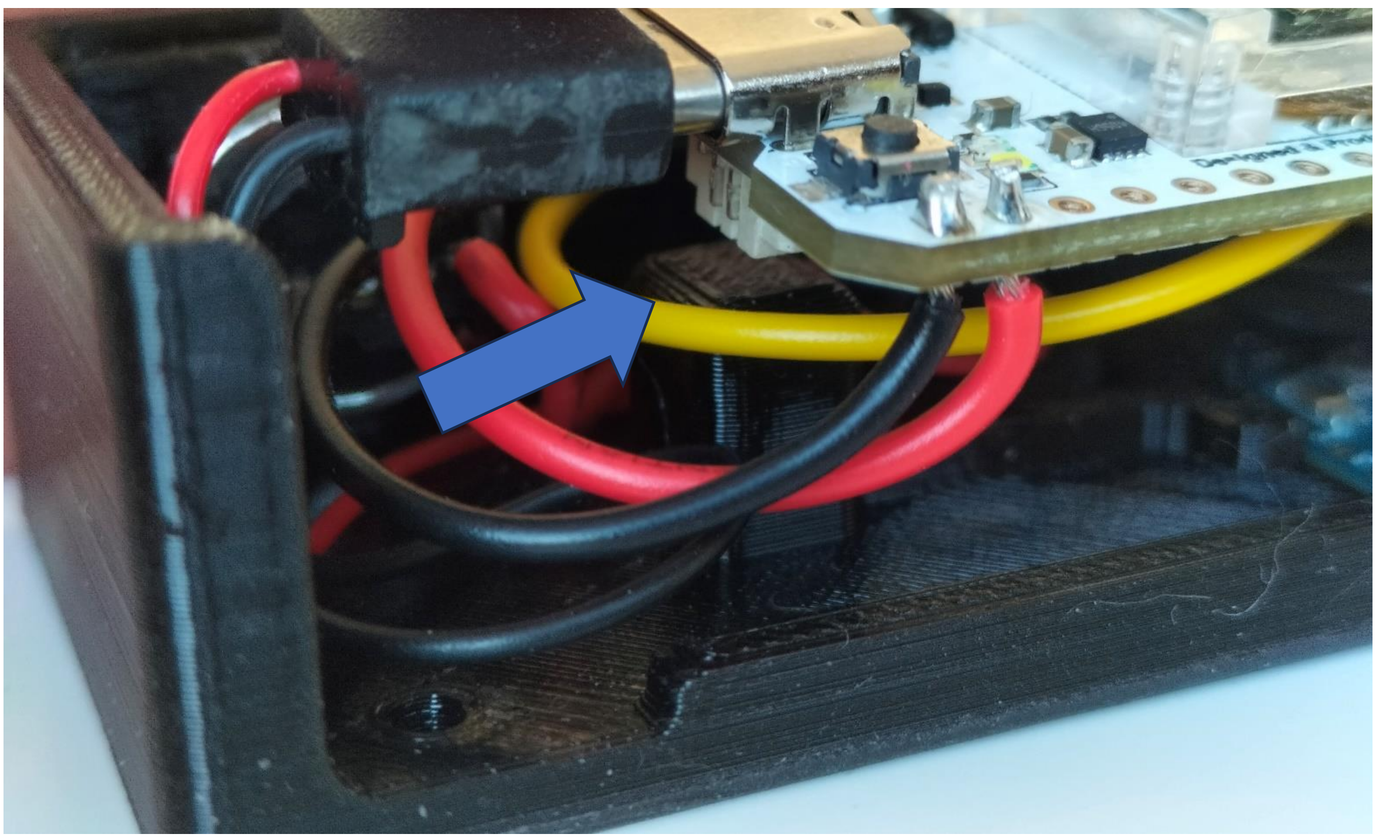


Make sure the antenna's cable being out of the way from the screws' holes.

Make sure to clear the space in between the battery connector and the Case pillar.



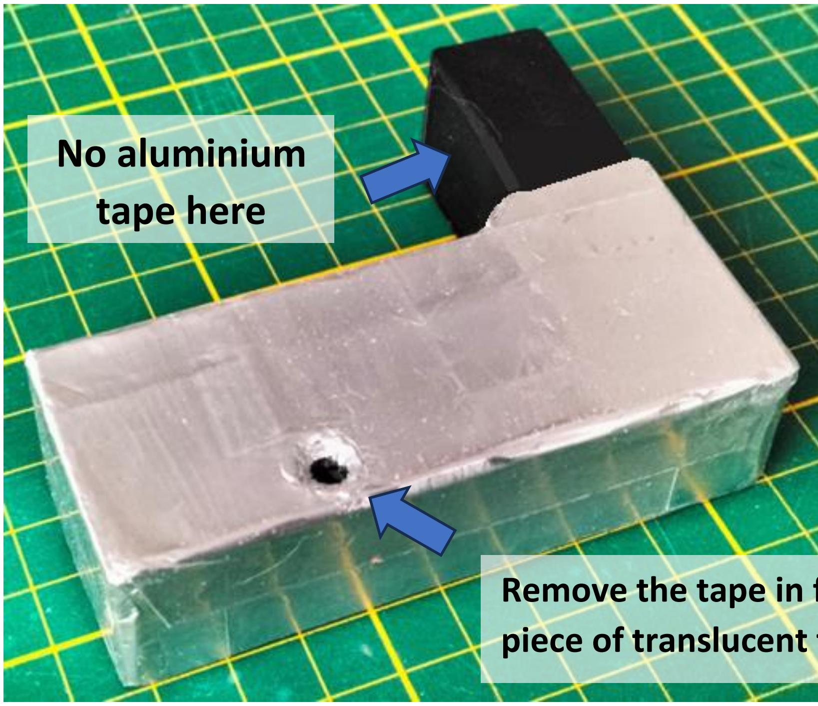
Make sure to clear the space in between the battery connector and the Case pillar.



Close the cover: 4 x M3x12mm

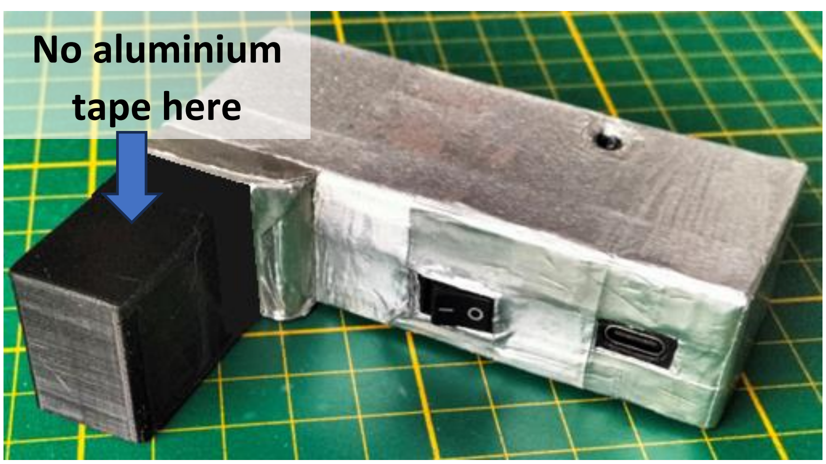
Wrap some the of stationary device sides with aluminium tape (ca 1mm thickness, about 12~16 layers).

The aluminium tape prevents the radar from sensing inside the house: The radar might sense you when passing by. The radar might also sense one of your cats that remained inside while others are outside; In this case, the user might want the “cat at the door” system monitoring the cats outside, without sensing those inside.



Covering the Oled display is not a problem, as not used.

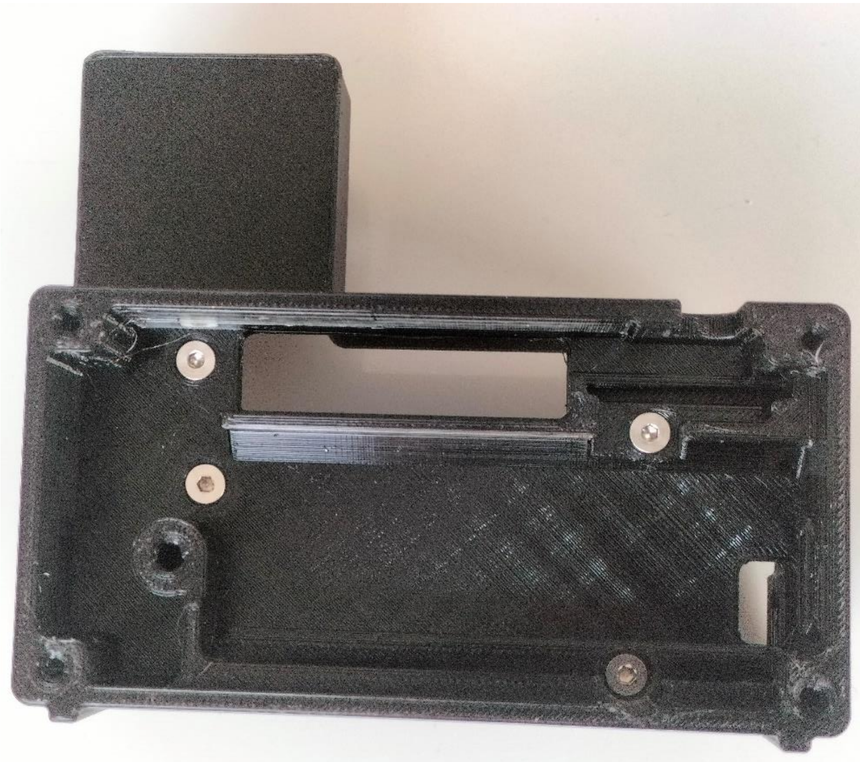
Differently, do not cover the last 30mm of the antenna’s cover part, otherwise the LoRa communication degrades.

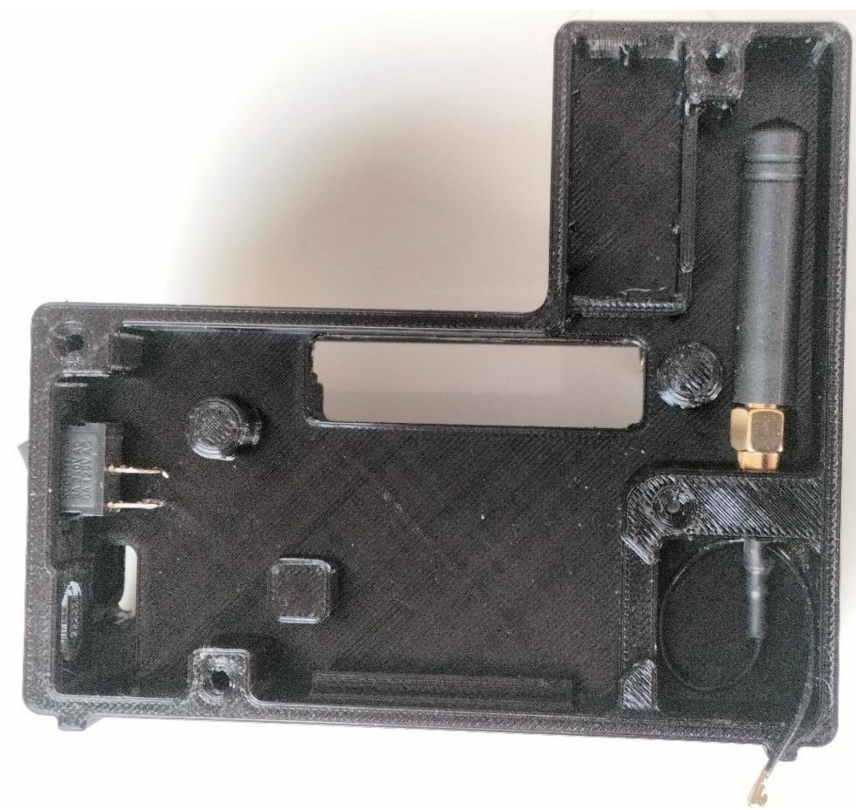


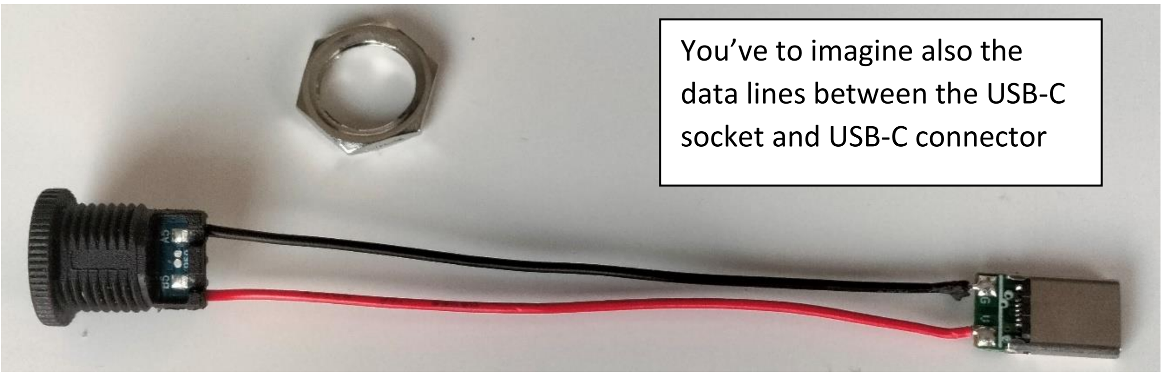
Do not cover the back side of the stationary device (radar)

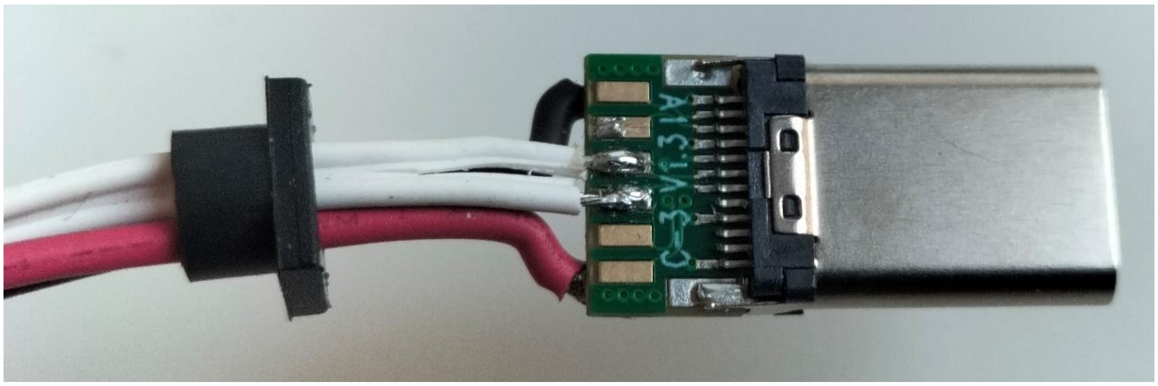
Portable device assembly details:

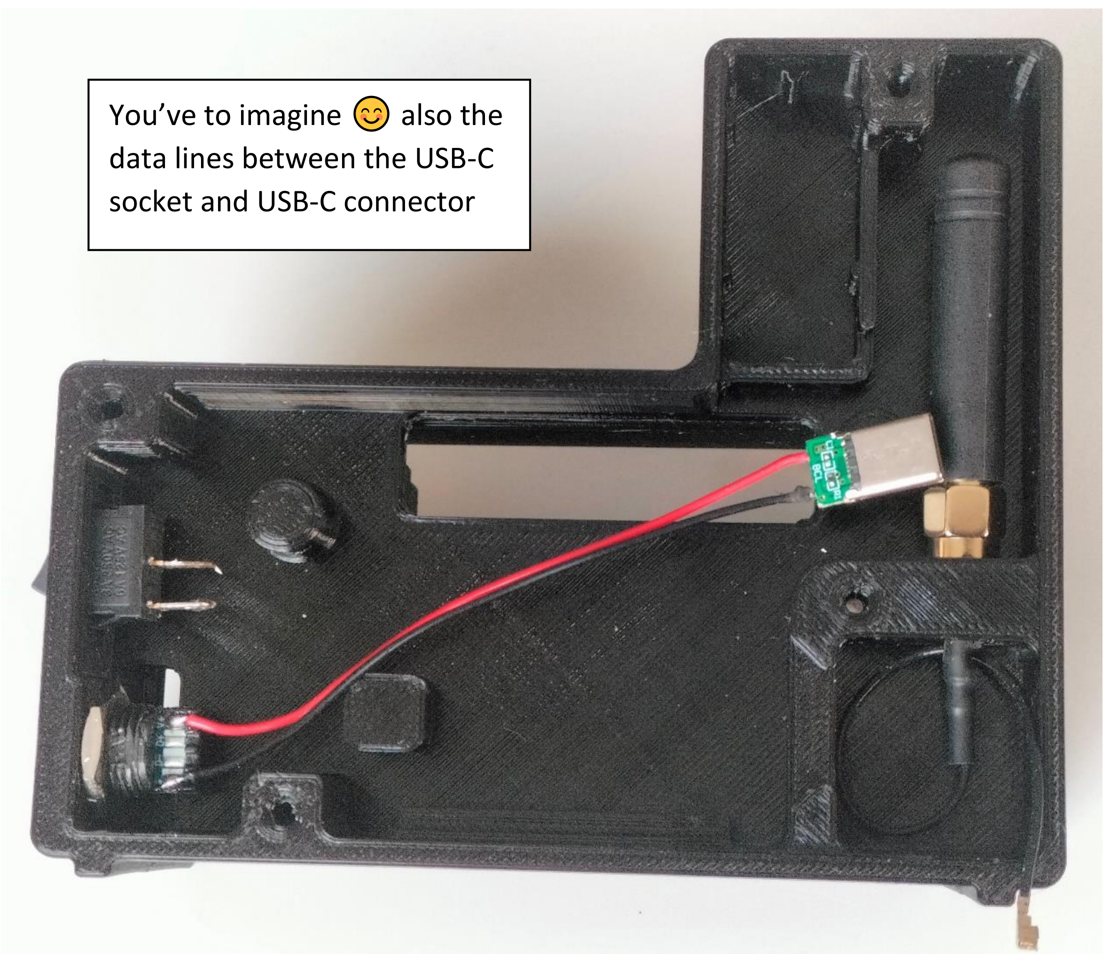
Connect the Case_Board_portable to the Case_EPD_portable, via 4 screws M3x10mm






You've to imagine also the
data lines between the USB-C
socket and USB-C connector




You've to imagine 😊 also the
data lines between the USB-C
socket and USB-C connector

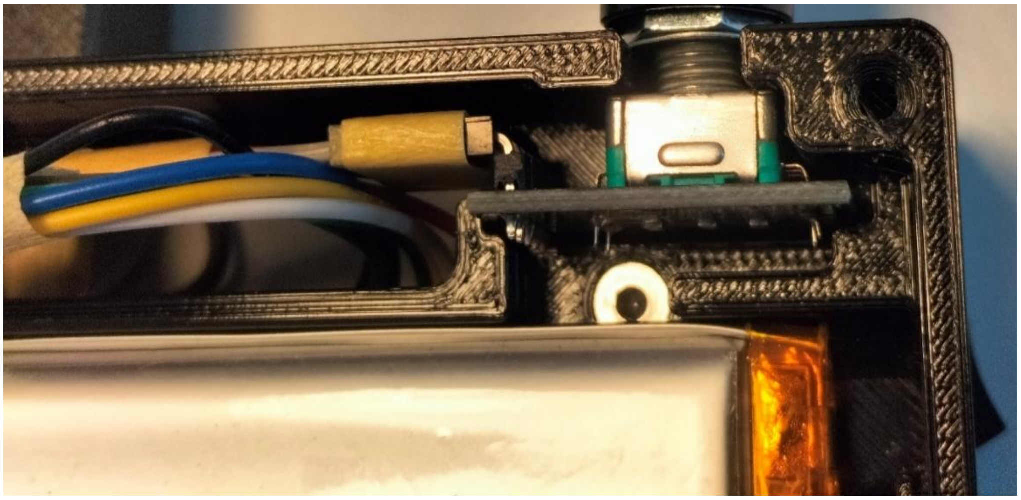
Position the battery with the cable on the right side, passed though the cases opening'.

Insert the Battery_stopper into the recess at the right and fix it with a M4x12mm screw.

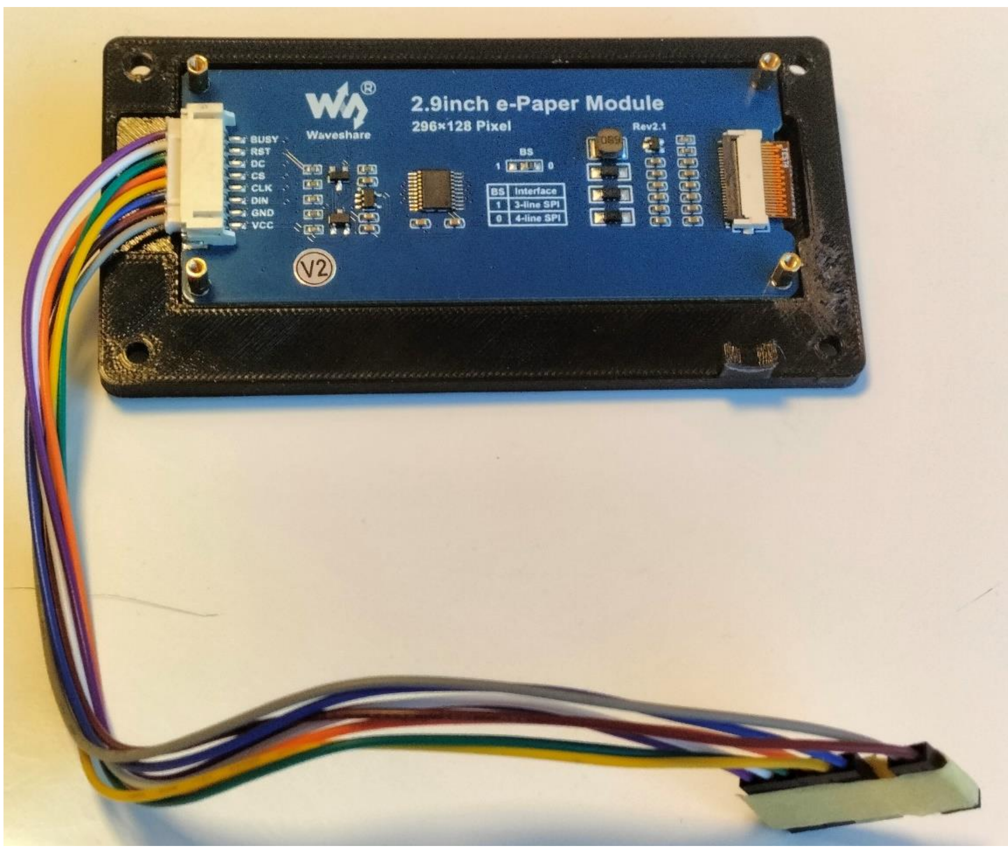
Connect the Encoder with the Dupont cables, preferably prepared upfront, by also adding a couple of turns with masking tape.

Slide the encoder board into the slot, fix it with the provided washer and nut; Pass the encoder cables through the cases' opening.

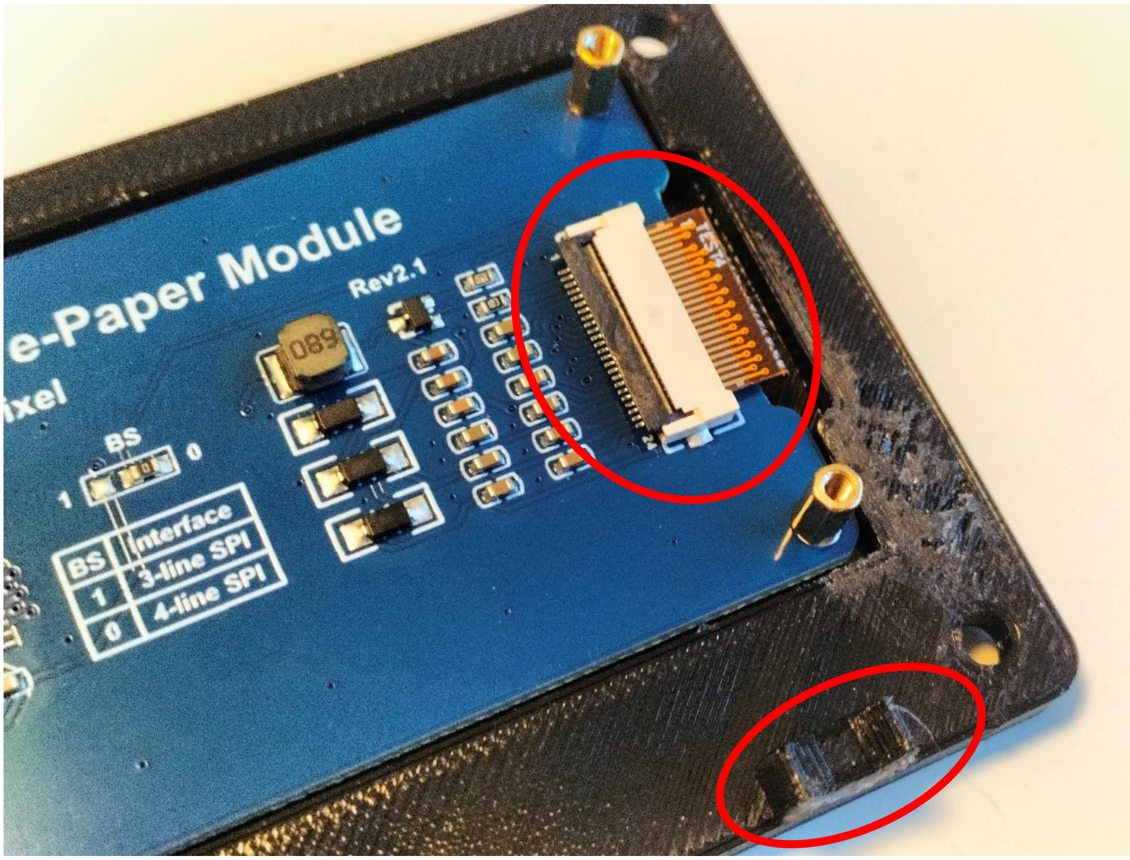
Insert the knob, by orienting the flat surface to the one on the rod



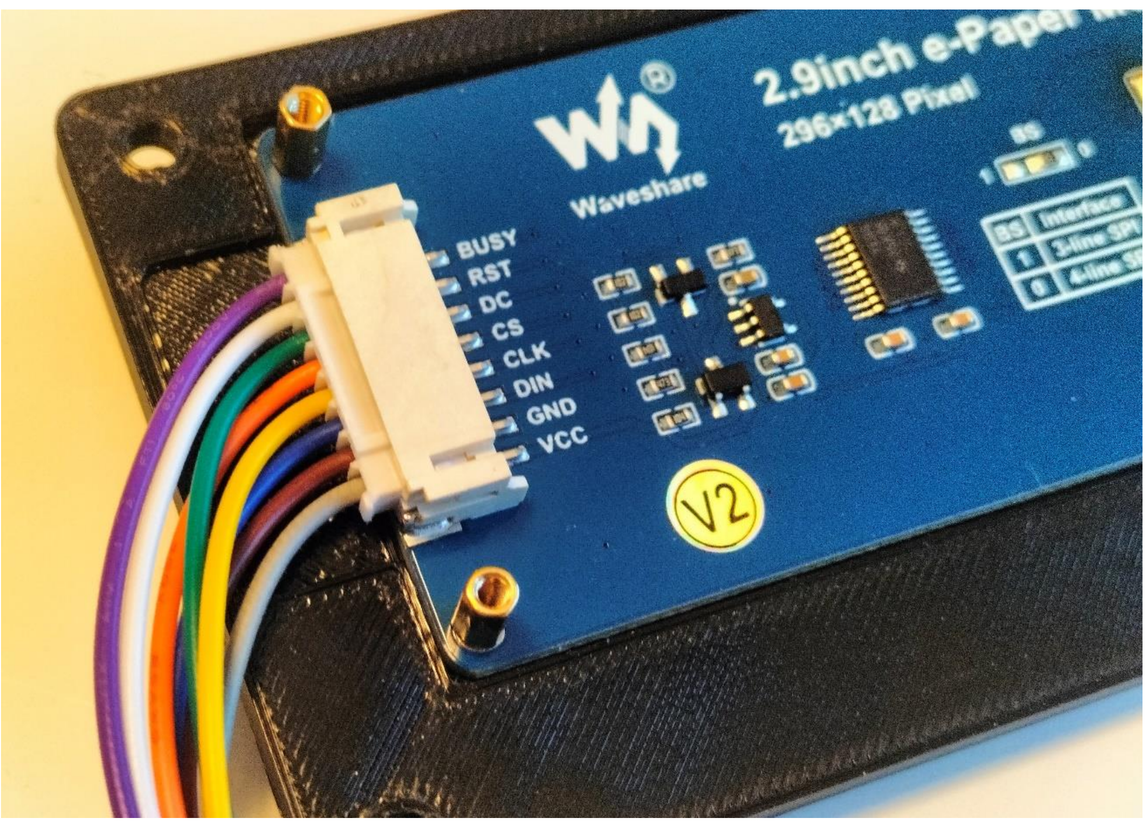
Gently insert the EPD display into the recess of Cover_EPD_portable: No force needed !



EPD side with the ribbon cable goes to the side where the Cover has the protrusion for the Encoder fixing



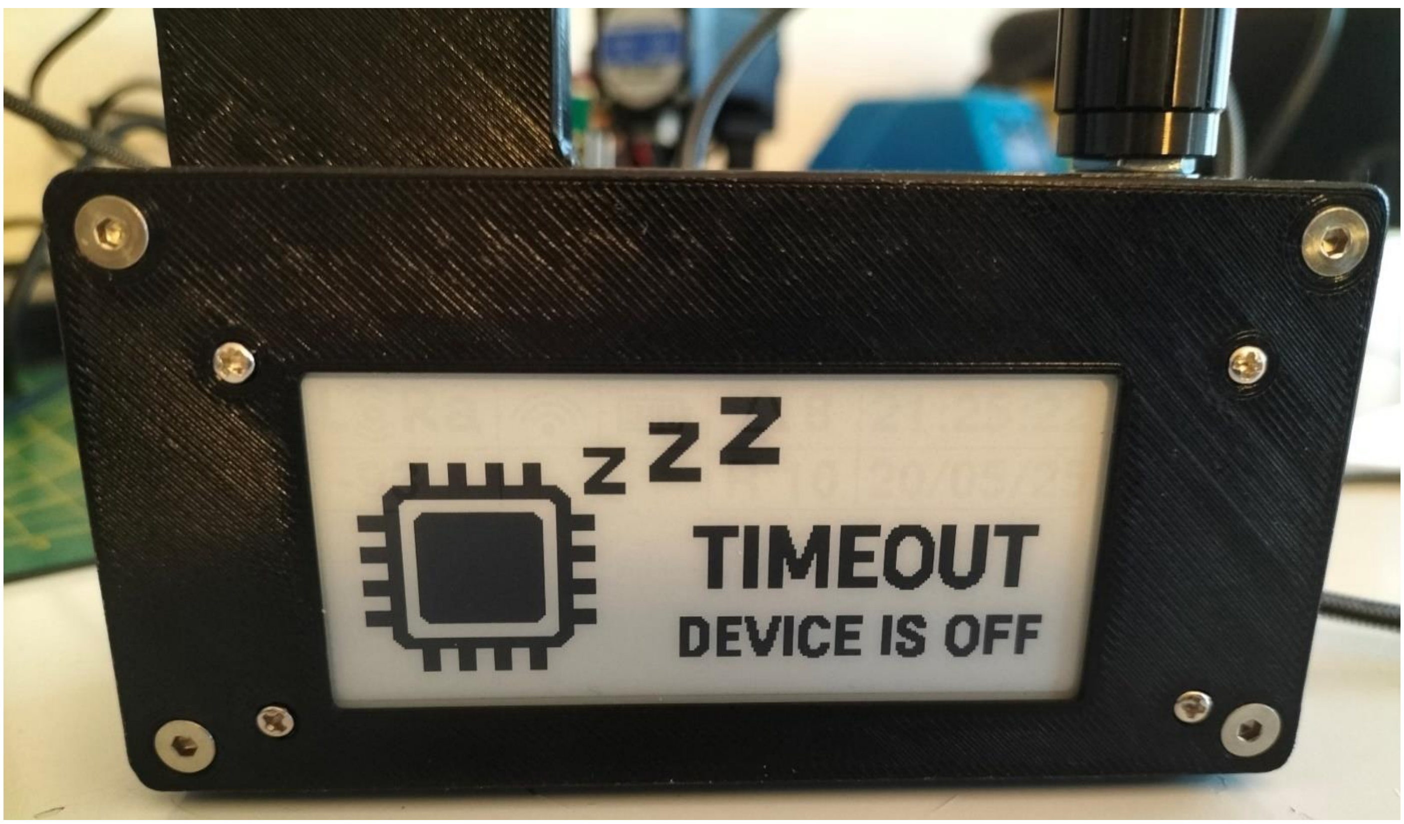
Insert the provided screws, and fix the display with it with the four hexagonal spacers



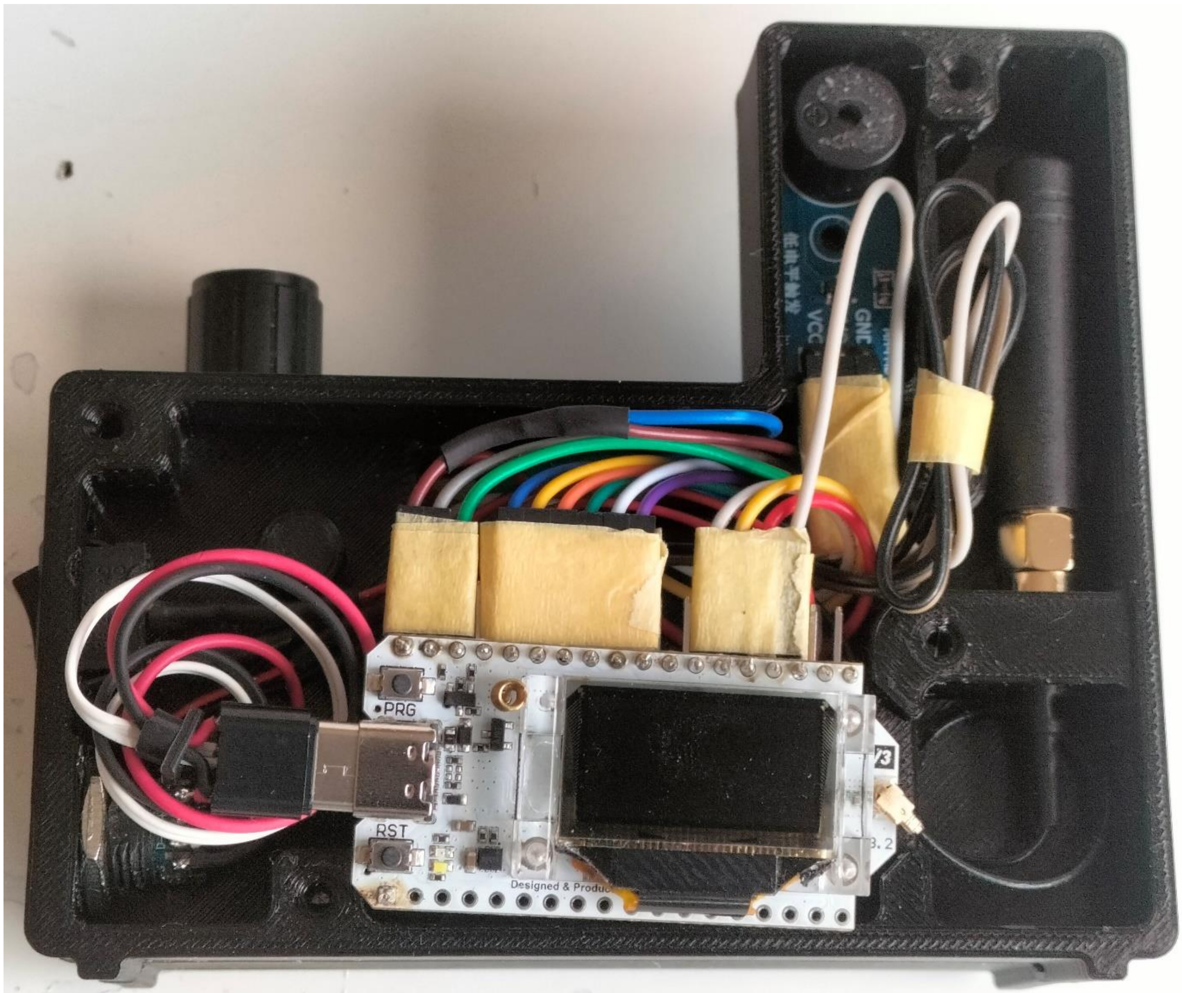
Pass the EPD wires though the upper opening of the cases.

Position the Cover_EPD_portable on the case, and the contact between them should happen without exerting noticeable force.

Close the Cover_EPD_portable with 4 screws M3x12mm.



The case has a few ribs generating a little interference with the Buzzer board: Gently press the buzzer into the slot.



Connects the Buzzer, EPD and Encoder to the Heltec board.

Organize the excess of wire.

Blocking upfront the Dupont connectors, with a couple of turns of masking tape, is of a great help; This also makes more unlikely the detaching of a single connector.

Group of connectors at the Heltec J3 header: 1, 2, 3;   5, 6, 7, 8, 9, 10;   13, 14, 15, 16

Group also the connectors at the Buzzer and Encoder connection.

Connect the LoRa antenna IPEX connector to the Heltec board. Ensure it being out of the way from the screw hole (see picture)

Connect the USB-C connector to the Heltec board; Excess of cable can be looped upfront.

Place the Cover_Board_portable, by gently searching the Oled display recess.

The Cover_Board_portable should come in touch with the Case without appreciable force.

Fix it with 4 screws M3x12mm

16. Programming the Heltec board

You might program the boards before assembling everything; Of course, without all the connections in place the code cannot be properly tested.

Step1: Download the OS

The OS must be downloaded from the MicroPython repository (https://MicroPython.org/download/ESP32_GENERIC_S3/), for a "generic" ESP32 S3 board.
The version I've used is: ESP32_GENERIC_S3-SPIRAM_OCT-20241129-v1.24.1.bin

Step2: Flash the OS

Connect the Heltec Wi-Fi Lora 32 V3 board to the PC via a USB-C cable having data transfer.
To flash the OS, the board must be set in the programming mode: Keep pressed the BOOT button while connecting it to the PC; Once connected, the BOOT button can be released.
The board will be recognized as a memory drive, and it will be sufficient dragging and release the BIN file into that memory drive.
The board will reboot right after

Step3: Check if the OS has been successfully loaded:

Open Thonny (download it from here: https://thonny.org/)
Click in the bottom right corner and verify is any ESP32 board is detected in one of the virtual COM ports.
Select that board.
Press the STOP button in Thonny interface.
Check if the MicroPython version, visible in the Thonny's shell windows, matches the BIN file.

Step4: Download the files from GitHub repository:

Via this link a helpful online tool opens: https://downgit.github.io/
Paste the address of the folder to be downloaded (https://github.com/AndreaFavero71/cat_at_the_door/tree/main/src).
Click Download, and a zip file of the folder will be land in your download folder.
In the zip file there are the files for the stationary device (src/stationary/) and for the portable device (src/portable/)

Step 5: Personalize the cat(s) name(s) and related Tag:

Both the devices (stationary and portable) use *ble_tags.json* text file to get to know the cat's names and the MAC address of the associated BLE tag.
The *ble_tags.json* file must have the same content at stationary and portable devices.
For the details check the *System's settings* chapter

Step 6: Personalize the settings:

Be noted the two devices (stationary and portable) use *config.json* text file for most of the settings. Many settings are not expected to be modify, for instance the GPIO pins used by the different functions (like LoRa, display, and others).

On the other hand, some settings should be customized:

- Time zone (refers to UTC, or formally to Greenwich)
- Dst (Daylight Saving Time): Set to 1 if your Country changes time in summer, 0 if don't.
- Wi-Fi name (Wi-Fi is only used for date and time synchronization).
- Wi-Fi password.
- BLE signal strength to filter out BLE tags with too week signal (i.e. the neighbours' tags).
- Buzzer (disable/enable, and number of beeping repeats).

For all the settings, their meaning and default values, check the specific *System's settings* chapter

Step 7: Upload the files to the boards:

Be noted the two devices (stationary and portable) use different files.

Connect the Heltec Wi-Fi Lora 32 V3 board to the PC via a USB-C cable having data transfer.
In Thonny, select View, then Files, to enable a window with the files available in the connected board and in the select PC's folder.
Via the files window of Thonny, navigate to the PC's folder where you downloaded the files.
Select all the files for the connected device (either stationary or portable); For the selection you can CTRL click each folder/file, or you can select (CTRL click) the first folder and the last file.
With the selected files in the PC's folder, right click over one of the selected files and select "Upload to /"
All the selected files will be copied to the Heltec board.

Repeat the process for the other device

Step 8: Backup the *config.json* and *ble_tags.json* in a safe place.

17. Project files

Stationary device:

Folder	File	Purpose	Notes
lib	ssd1306.py	Driver for the Oled display	Oled display only useful for debug
lib	sx1262.py	Driver for LoRa chip	Modified files from the original (https://github.com/ehong-tl/micropySX126X)
lib	sx126x.py	Driver for LoRa chip	
lib	_sx126x.py	Driver for LoRa chip	
\	boot.py	System file	Left empty, as no specific tasks necessary at the booting for the stationary device
\	config.json	Json file with most of the settings.	The same file is used for both the stationary and portable devices
\	main.py	Python file orchestrating the stationary device tasks	
\	ble_tags.json	Json file with cats' names and the BLE tags mac addresses	The same file is used for both the stationary and portable devices

Notes:

1. All the drivers are placed into the \lib library to keep the attention to the other files.
2. In MicroPython, files located in the \lib folder are imported without the need to specify their path.

Portable device:

Folder	File	Purpose	Notes
lib	config_utils.py	Manages the config.json file when settings are modified via the menu	
lib	datetime_converter.py	Converts datetime	
lib	deepsleep_icon.bin	Bin file with a “sleeping” icon	
lib	encoder_menu.py	Manages the menu via the encoder	
lib	epd2in9_V2.py	Driver for the EPD display	Modified version vs the original at https://github.com/wavesharetea m/
lib	helvetica22bold.py	Font 22 pixels for EPD display	
lib	helvetica26bold.py	Font 26 pixels for EPD display	
lib	icons.py	Collections of custom icons (LoRa, Wi-Fi, battery, etc)	
lib	rotary.py	Manage the encoder	
lib	rotary_irq_esp.py	Manage the encoder’s interrupts	
lib	ssd1306.py	Driver for the Oled display	Oled display only useful for debug
lib	sx1262.py	Driver for LoRa chip	
lib	sx126x.py	Driver for LoRa chip	
lib	_sx126x.py	Driver for LoRa chip	
lib	welcome_icon.py	Bin file with the “cat at the door” icon	
lib	writer.py	Manages text for EPD display	
\	boot.py	System file	Start the settings menu when the push button is pressed at the booting
\	config.json	Json file with most of the settings.	The same file is used for both the stationary and portable devices
\	main.py	Python file orchestrating the stationary device tasks	
\	ble_tags.json	Json file with cats’ names and the BLE tags mac addresses	The same file is used for both the stationary and portable devices

Notes:

3. All the drivers, and supporting files, are placed into the \lib library to keep the attention to the other files.
4. In MicroPython, files located in the \lib folder are imported without the need to specify their path.

18. Stationary device installation

The stationary device has been designed to be positioned inside, with the rear surface pointing toward outside. Tests are necessary to determine the best spots; This process involves tuning the radar as well.

Considerations:

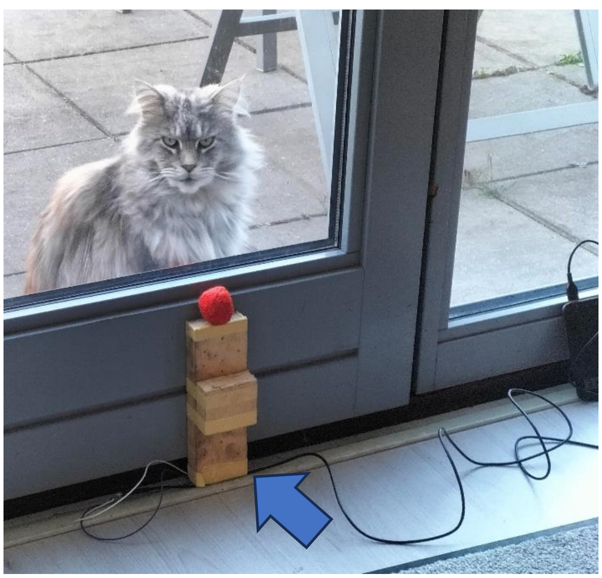
1. The radar senses moving objects; Be noted firm objects are detected as moving if the radar moves!
2. For the above reason, the stationary device could be mounted on the door only if it doesn't wobble when closed. This can be tested by shaking the door with one hand.
3. The radar couldn't easily sense through my door, while it could sense through a (marble?) composite material used as doorstep; This material is also used to support the large glass wall at the door's side.
4. Based on tests, I could place the stationary device on the door lower part, with the device protruding underneath to sense through the a (marble?) composite material.
5. Paint with metal is not favourable.

Just for reference, some pictures of my initial tests:

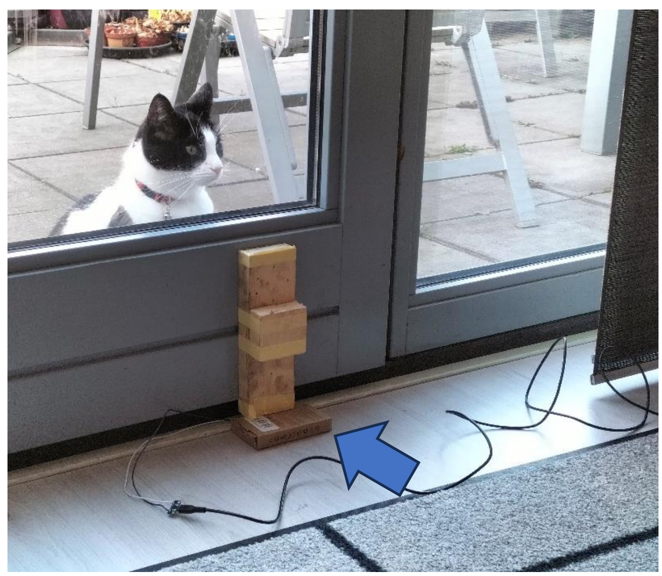
In this case, the radar was:

1. Fixed to a piece of wood, via masking tape.
2. It was powered by a 5 Vdc power bank; the output signal was left unconnected.
3. The radar response was checked directly via the HLKRadarTool application.
4. Tested in different locations.

In this test the radar was almost touching the floor (only raised by the black cable thickness)



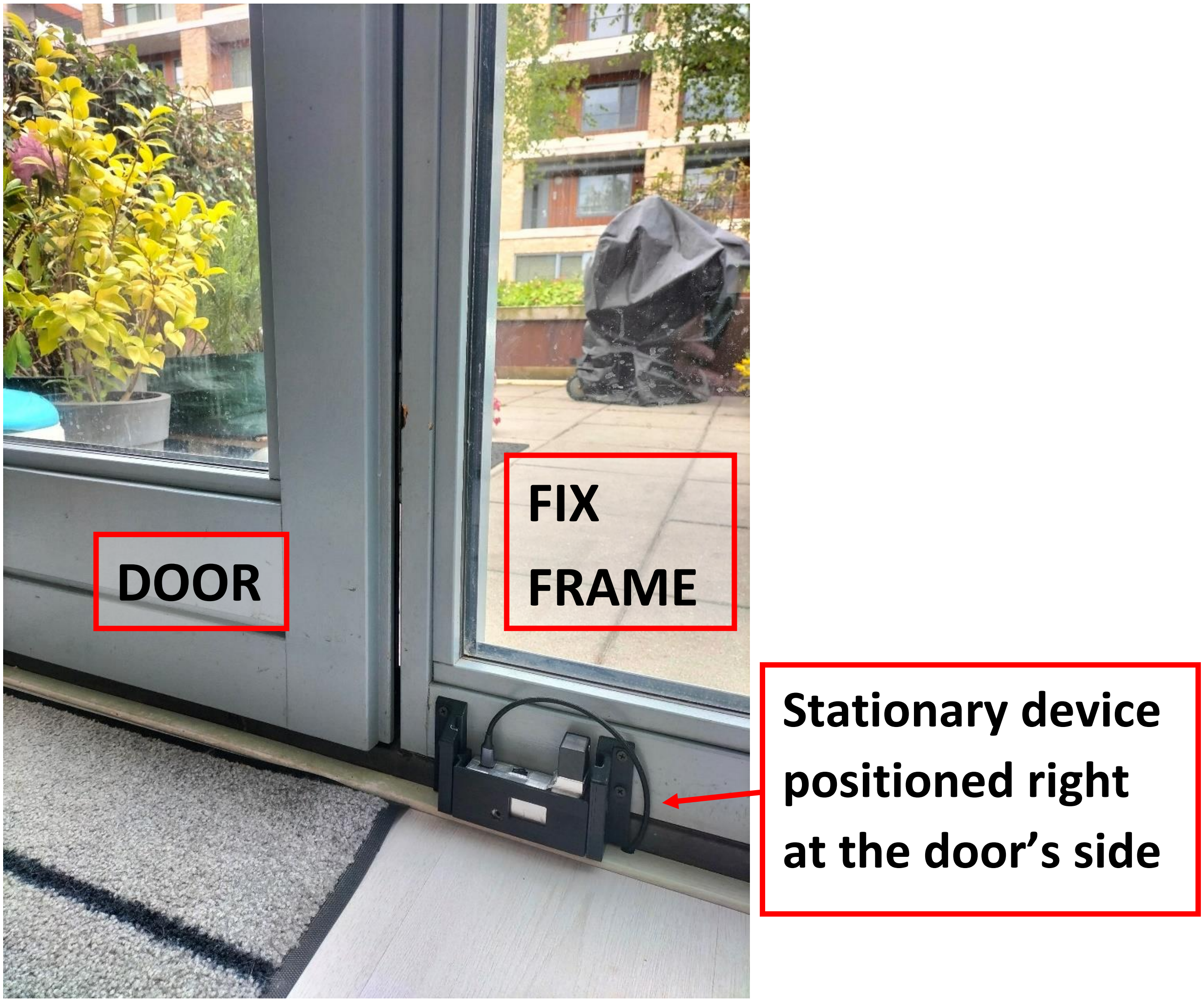
In this test the radar was raised a little bit more, via a cartoon box

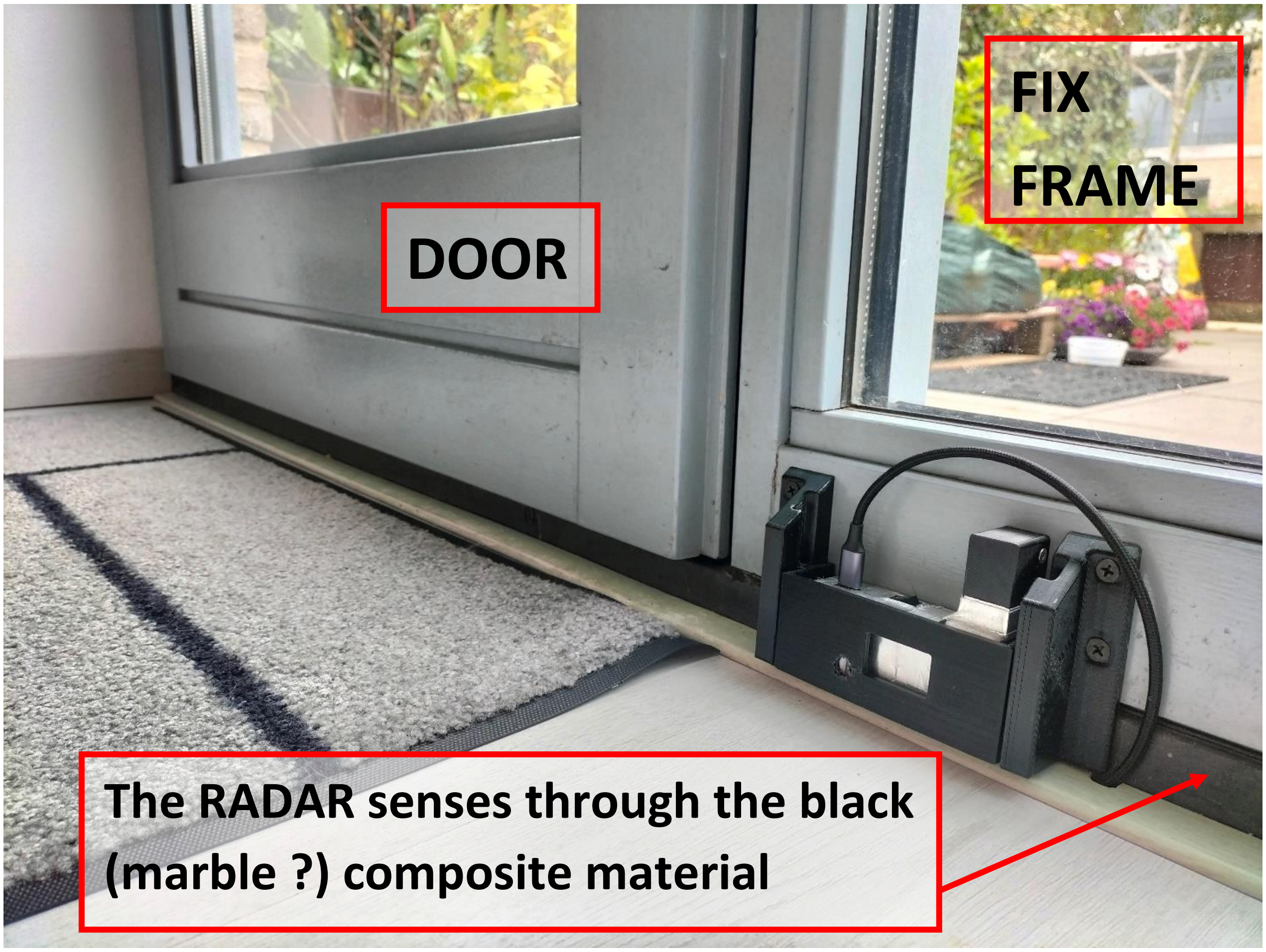


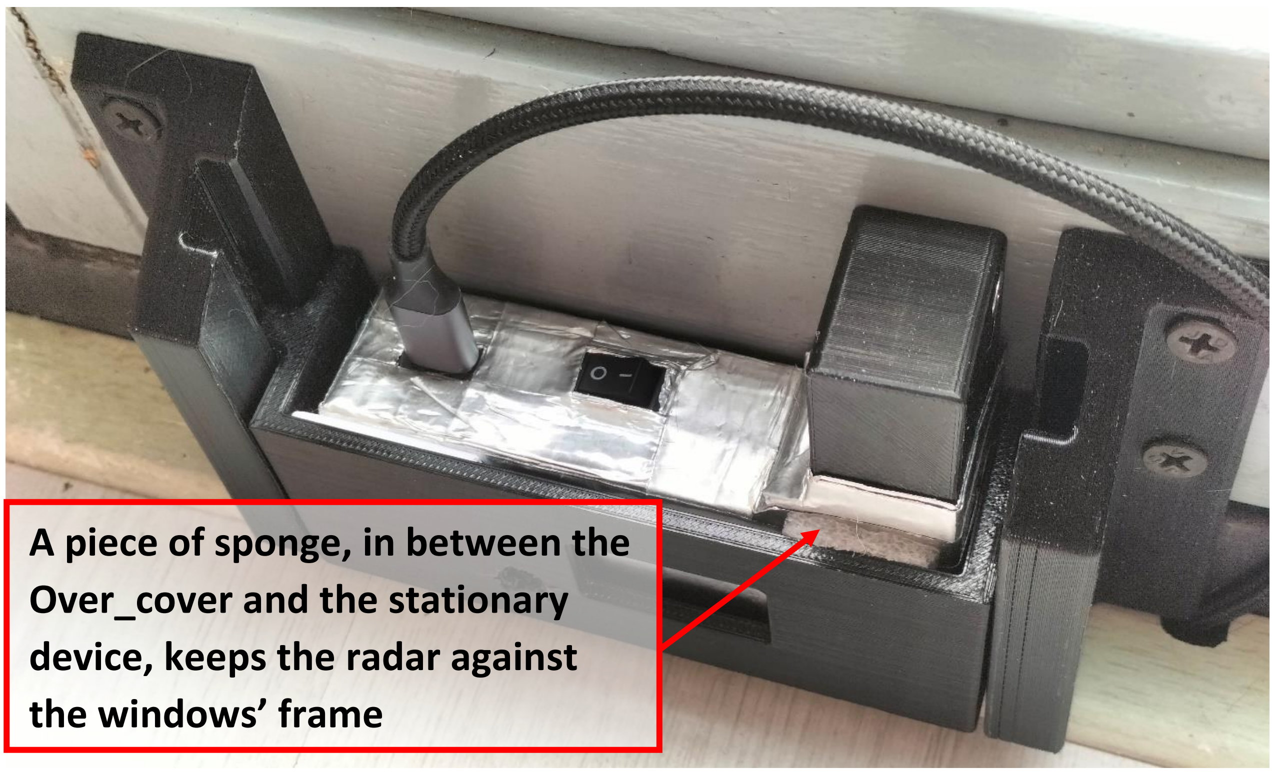
Notes:

1. Be noted, the radar senses also behind its own board: Keep some distance while testing.
2. I'm afraid, testing is simply necessary!

Just for reference, some pictures of **my final installation**:

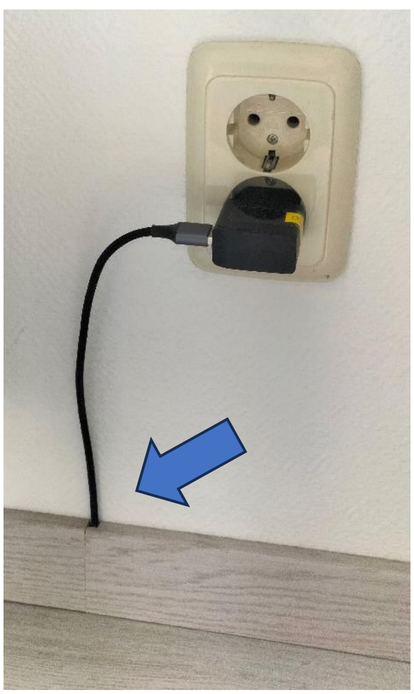


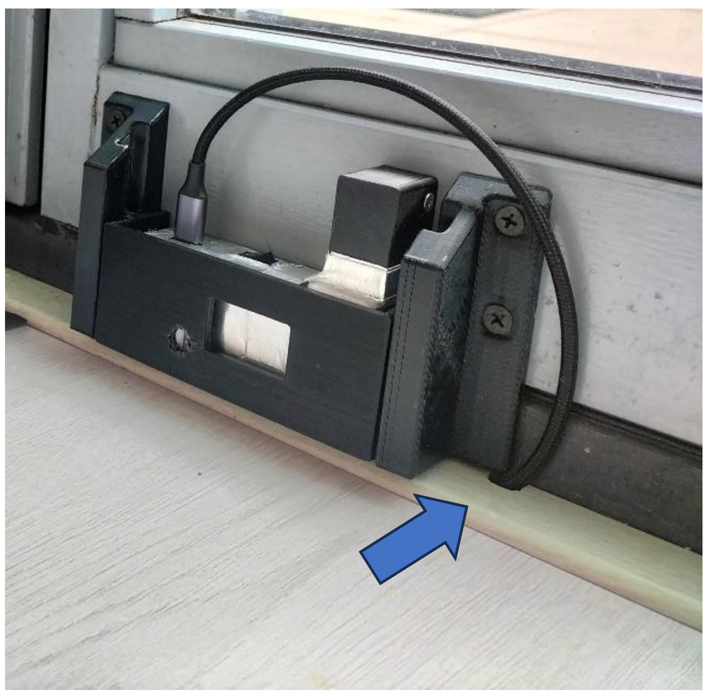

FIX
FRAME
DOOR
The RADAR senses through the black
(marble ?) composite material


A piece of sponge, in between the
Over_cover and the stationary
device, keeps the radar against
the windows' frame

USB-C cable routing:

I had the chance to route the cable behind the skirting board, and under the laminate's board; This allows for a cleaner, tidier, installation and avoids cables that you might trip over.





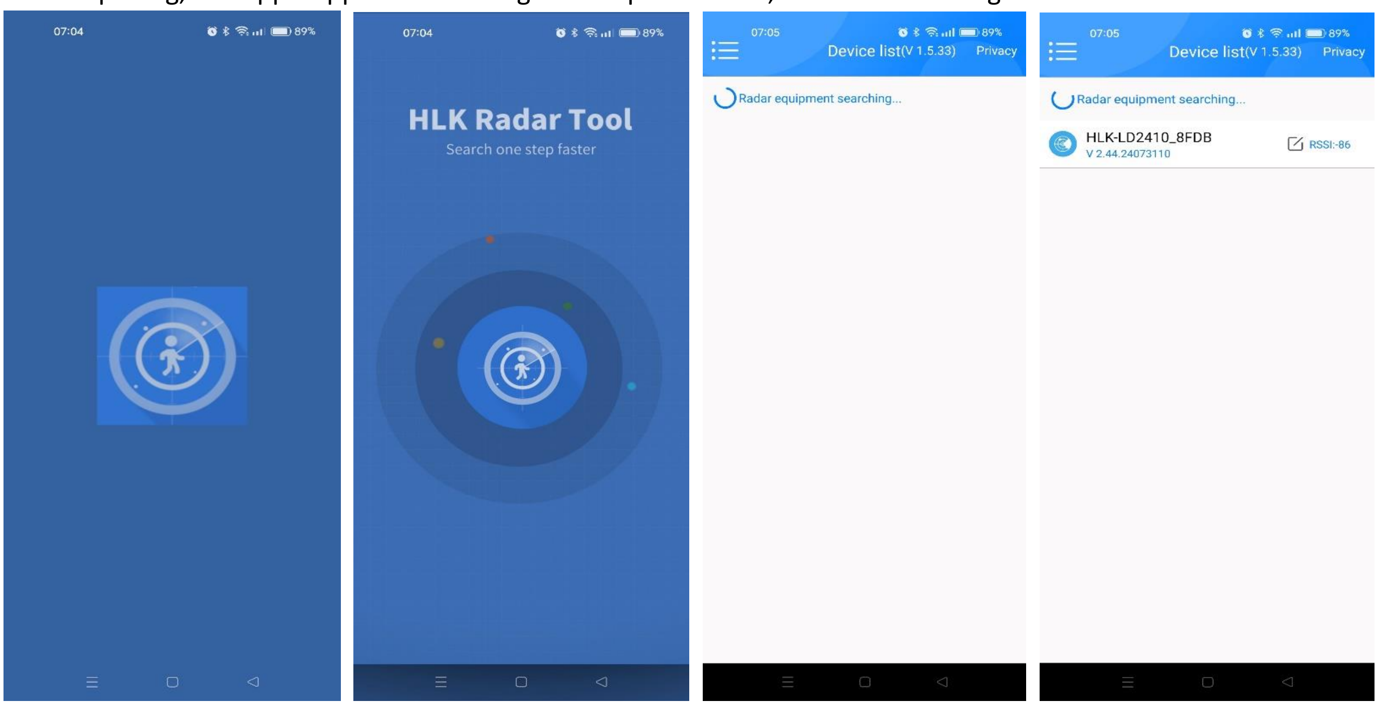
19. Radar’s tuning, via the App

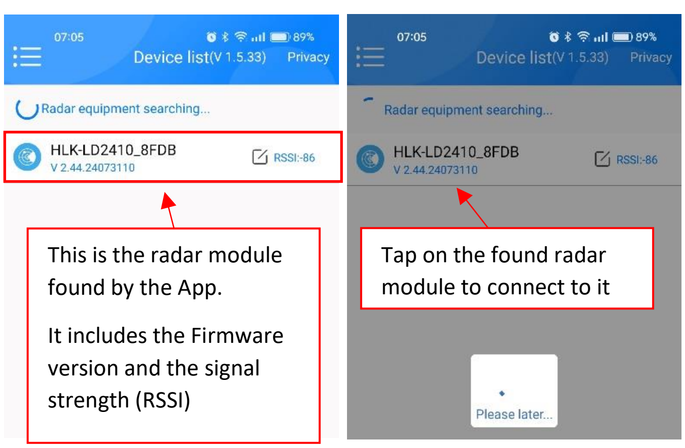
The radar board HLK TD2010C, made by Hi-Link, has been chosen because, among other features, it includes built-in Bluetooth capabilities, and it interacts with Hi-Link App: Settings can be made via your smartphone. Hi-Link radar’s docs: https://drive.google.com/drive/folders/1ypOlacBmmFXY6lDQ0f1hEJFmczNe-0WG

Download the free app **HLKRadarTool** from the Apple App Store or Google Play Store (search HLKRadarTool). In May 2025, the App’s icon on the phone looks like:

If requested, the default password is: **HiLink**

When opening, the App’s appearance changes a couple of times, as starts searching for radar modules:





Main page, holding most of the radar’s info:

Radar identification (+ mac address)

- Green (no detections)
- Red (Moving target detection)
- pink (Static target detection)

When Red or Pink the radar’s pin OUT changes level

Indicates which target (Moving or Static) triggers the pin OUT

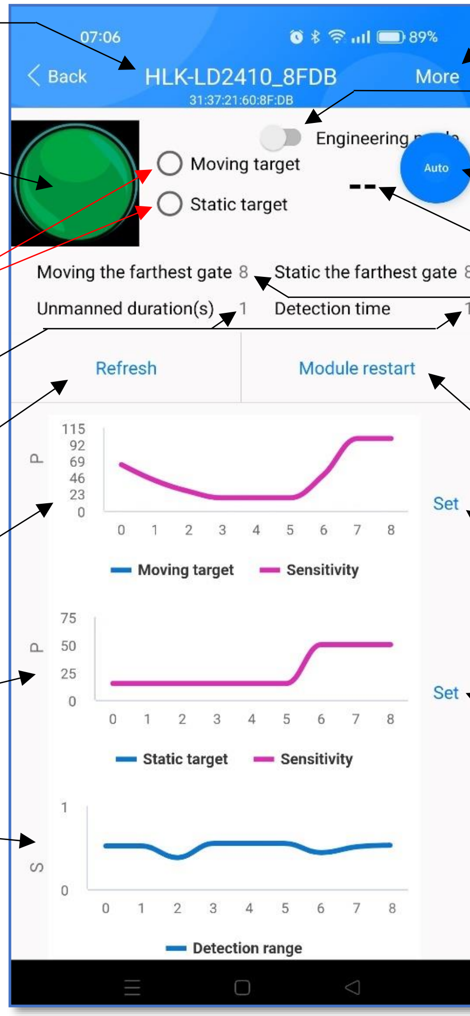
Time the radar holds the pin OUT triggered in case of detection

Force a screen refresh

Threshold (%) of the **Moving** target, per each distance gate (%)

Threshold (%) of the **Static** target, per each distance gate

Chart with real time distance, in meters, of the detected person



Enter the Parameters Settings menu

Slider enabling “Engineering mode”: It adds the real time radar’s reading to the Moving and Static charts

Automatic setting: Set thresholds after measuring background noise

Distance in meters of the detected person, or cat 😊

Number of enabled distance gates

Force a module’s restart

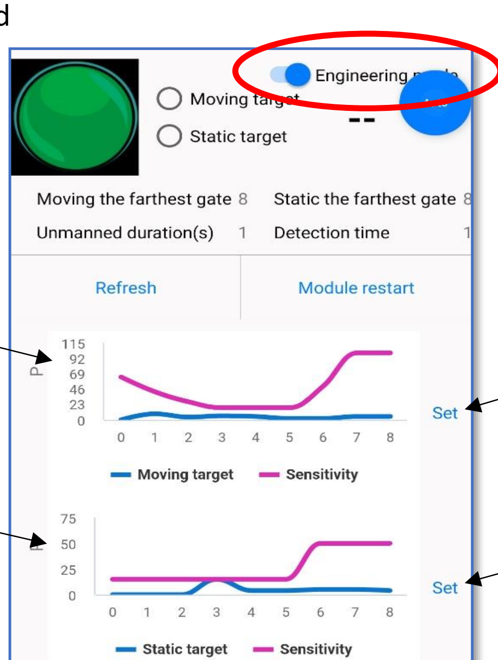
Enter the threshold setting for **Moving** target. It can be also reached from the setting menu

Enter the threshold setting for **Static** target. It can be also reached from the setting menu

When Engineering mode is selected

Real time detection, in blue, is added to the threshold (%), in purple, for the **Moving** target, per each distance gate

Real time detection, in blue, is added to the threshold (%), in purple, for the **Static** target, per each distance gate



Enter the threshold setting for **Moving** target. It can be also reached from the setting menu

Enter the threshold setting for **Static** target. It can be also reached from the setting menu

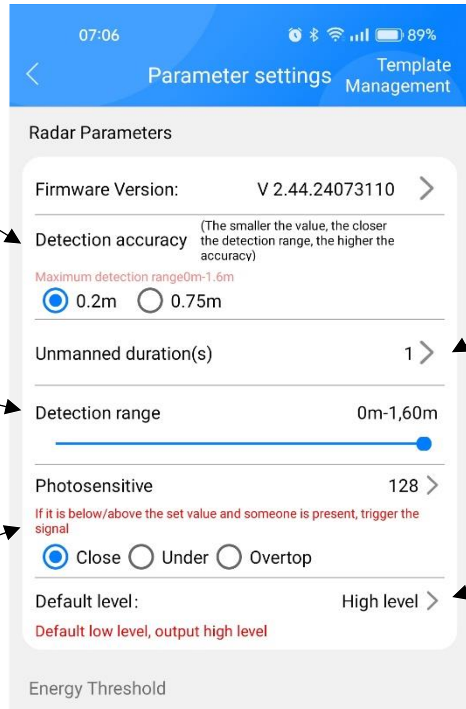
Parameters Settings page:

Detection accuracy: This affects the max radar range:

- Max range = 1.6 meters for the highest accuracy
- Max range = 6 meters with the lowest accuracy

Limits the radar range to a shorter distance. The available range depends on the set accuracy.

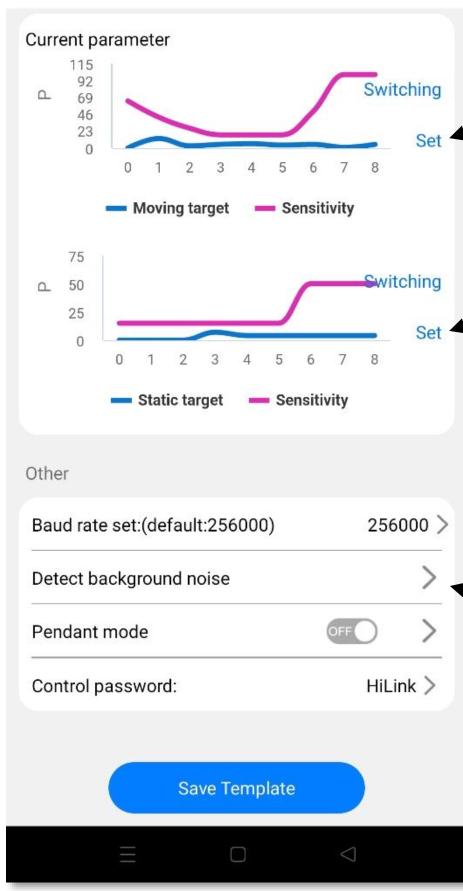
The radar has a photodiode, useful to switch a lamp on in case of low light and radar sensing something. “Cat at the door” project doesn’t use the Photosensitive feature.



Set the time the pin OUT maintains its status if a detection happens. Within this period, multiple detections aren’t discriminated. Note: The Stationary device uses an interrupt to capture the radar OUT signal, and it forces a variable for a longer time: Preferred having the radar updating its OUT pin as frequent as possible 😊

Set the pin OUT level (0V or 3.3V) when the radar detects something. **It must be changed to High Level.**

scroll down ...



Enter the threshold setting for **Moving** target. It can be also reached from the setting menu.

Enter the threshold setting for **Static** target. It can be also reached from the setting menu.

It allows the measurement of background noise, for both the Moving and Static targets, on each of the distance gates.

It reports the background noise, without altering the settings.

This feature helps to prevent false-positive detection by setting on setting the thresholds above the background noise levels.

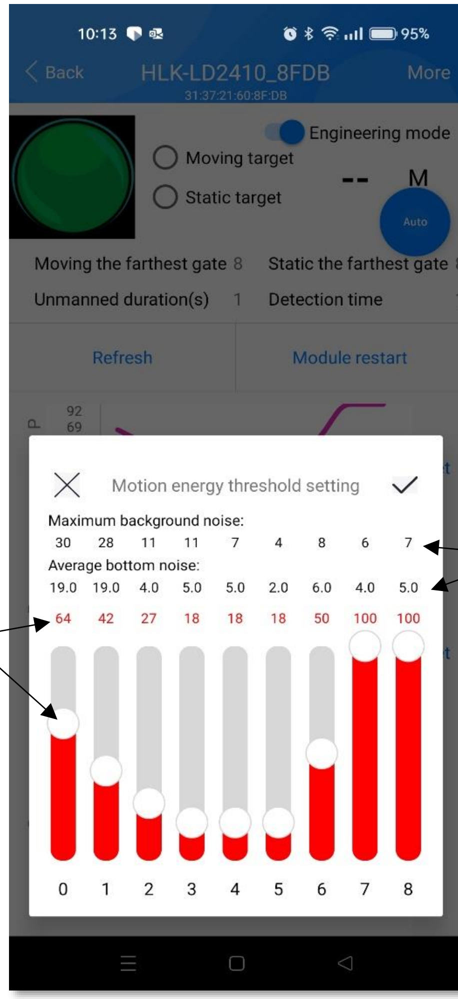
Setting the thresholds:

This menu can be reached from the Main page or from the Settings menu

When manually setting the energy thresholds, the Moving and Static thresholds are on independent menus, but the process is the same; Bars are in red for the Moving target, and in Blue for the Static target.

This first one is the *Motion energy threshold setting*

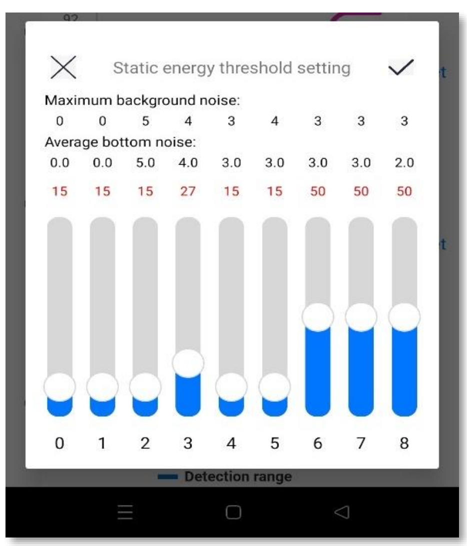
These are the energy threshold values; manually adjust the values via the slider. Set the threshold to 100 for the distance gates you'd like to disable the radar.



These values indicate the max and the average energy measured during the last "background noise" check.

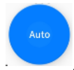
These values are also "captured" if the "Auto" setting was chosen.

When selecting the *Static energy threshold setting*, the process is just identical



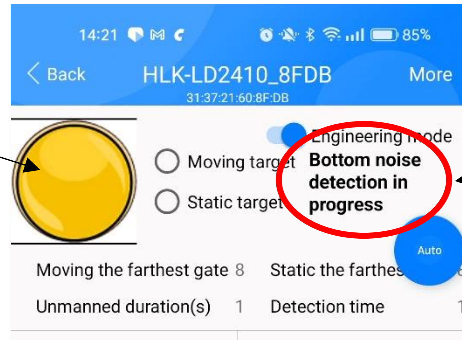
These are my settings 😊, combined with the highest accuracy (0.2m) and therefore the max range limitation to 1.6 meters.

Auto tuning:

The Auto tuning can be chosen by taping on 

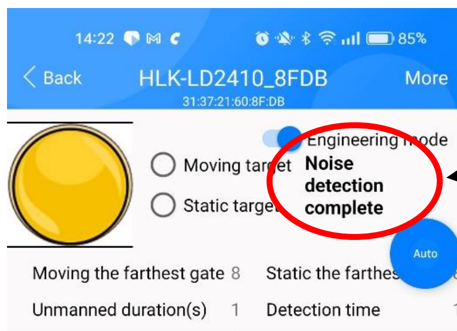
, and accepting the conditions.
This is not my preferred approach, but it might help as first setting.

The LED turns yellow during the Auto setting period.



The background noise starts

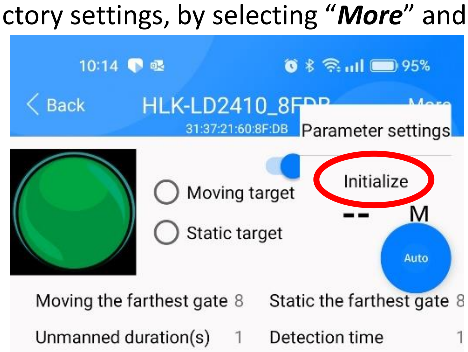
After about 2 minutes:



After about 2 minutes the background noise is measured.
The App decides the energy threshold for the Motion and Static target.
Note: Not explained how these values are calculated.
The App comes back to its operational mode, with updated thresholds.

Factory settings:

It's possible to reset the radar to the factory settings, by selecting "***More***" and then "***Initialize***":

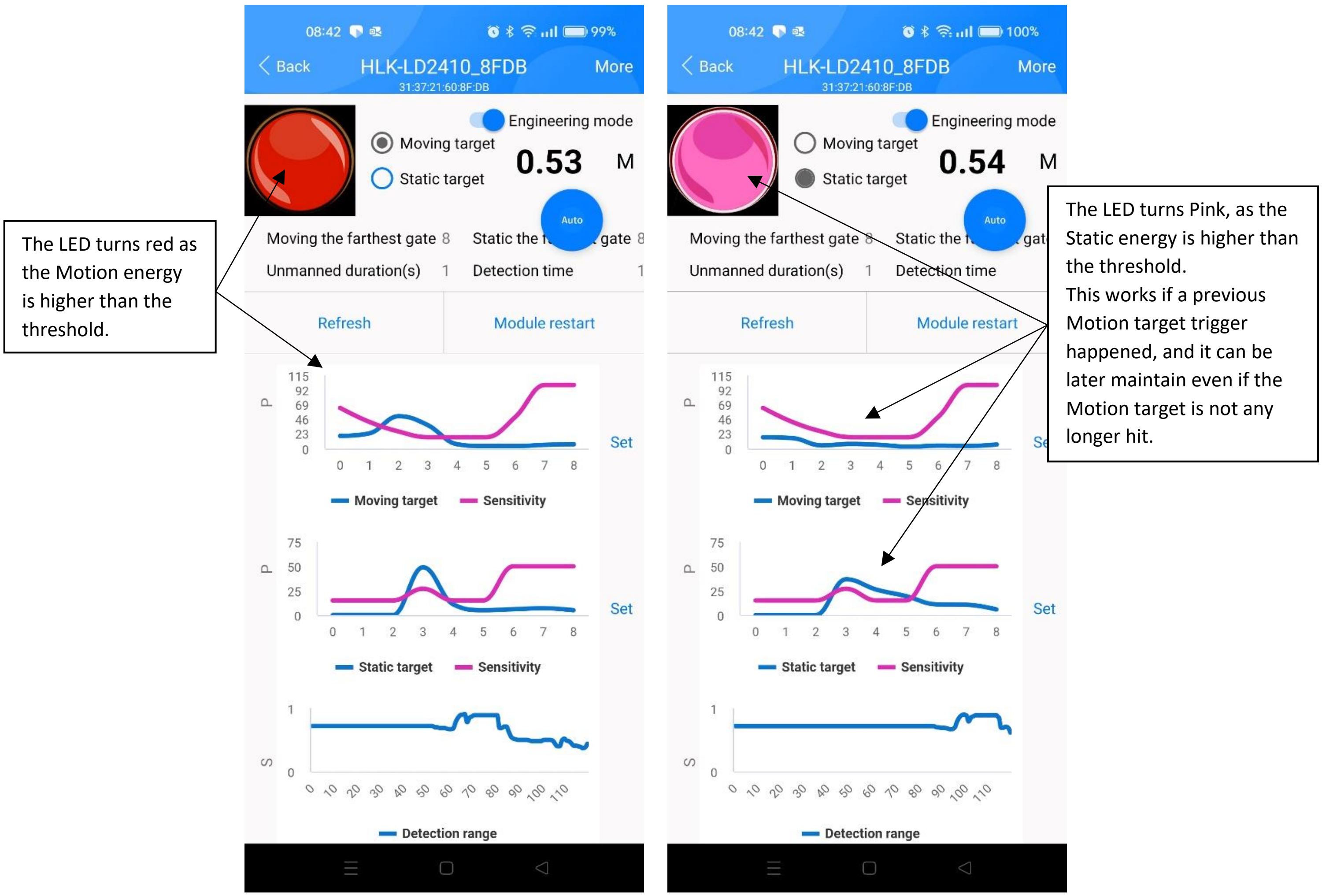


Interpreting the results, and (my initial) misconception:

To get the radar triggering the output, there must always be an initial crossing of the Motion target; afterward, the signal might be maintained by the static motion detection.

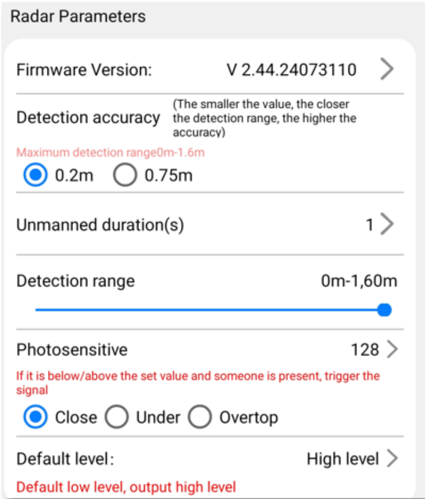
This can be easily checked by setting to 100 all the gates for the *Motion energy threshold setting*

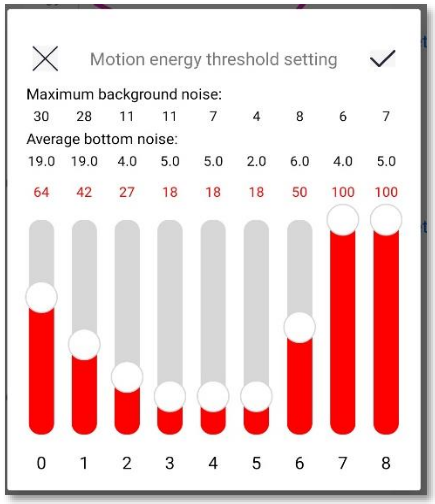
This means the led will initially turn red, and afterward it might turn to pink

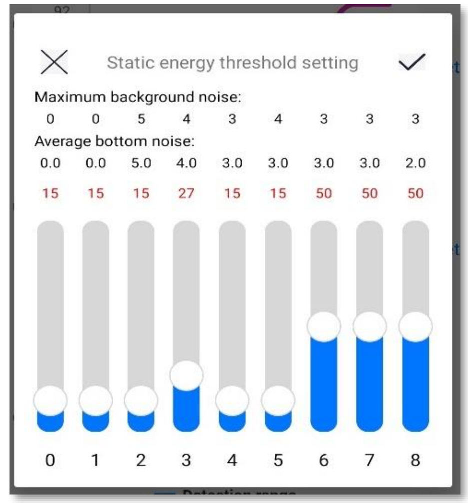


On the third chart, *Detection range*, plots the detection distance; When the radar detection cease, the chart keeps plotting the latest distance

Just as reference, here below are my radar settings:







20. BLE Tag activation

The BLE tags Tile, model Sticker, from Life360, must be activated, otherwise it doesn't "advertise". Advertising is the Tags' process of sending out data packages.

The activation process is not complex, but it requires following the below steps:

1. Download the free Tile app (Android or iOS).
2. It requires you to make an account. **Note:** No need for any subscription, for the scope of this project
3. Select "Add a Device".
4. Select "Tile Devices".
5. Press the little button on the side of the Sticker BLE Tag, and select "Next"
6. The App tries to connect with the Tag, keep it close to the phone.
7. The Sticker BLE Tag gets activated, select Next.
8. Select Cat, then Next and Next again.
9. The new Sticker BLE Tag is associated with the Tile App; An email will confirm the Tile has been added.
10. Select the new Sticker BLE Tag
11. Scroll up the list of options.
12. Select "More Options".
13. Edit the Name: Use the name of the cat that is going to use that tag.
14. Press the "back" button of your phone.
15. The edited name should now be associated with the new Sticker BLE Tag.

Notes:

1. Within the "More Options" menu, scroll up completely and the activation date / time is indicated, in addition to the model of the Tile and the conformity with the relevant Standards.
2. From now on, the BLE tag Sticker advertises every 2s.

I might add in future the pictures I took during this process.

21. BLE tag mac address

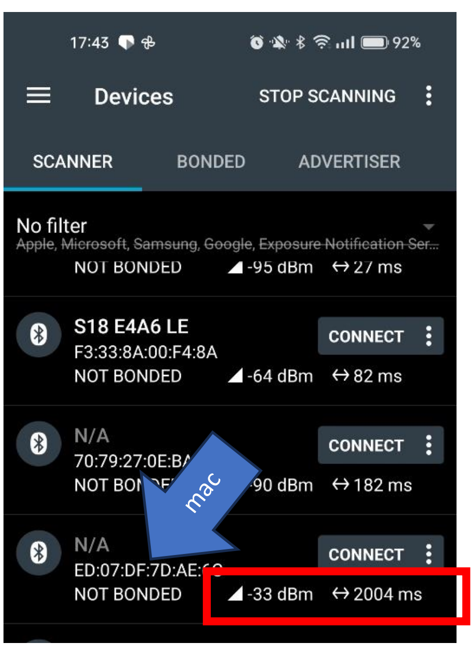
Once the BLE tags Tile Sticker is activates, it advertises every 2.0 sec.
The next step is to get to know the mac address of the newly activated tag, to be later used in the ble_tags.json.

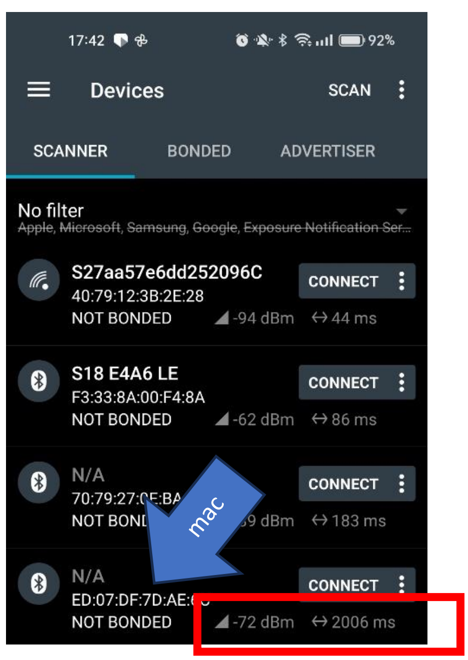
There are plenty of methods to retrieve the mac address, here it is proposed a simple one.

1. Grab a piece of aluminium foil, the one commonly used for cooking.
2. Download the free App “nRF Connect”, available for smartphones or other devices; It doesn’t require making an account.
3. You can watch the plenty of introductory pages or just skip (as I did).
4. Position the smartphone close to BLE tag.
5. Press the SCAN button on the top-right corner.
6. The phone screen will populate with one or plenty of BLE tags.
7. By watching the rightmost column search for a BLE tag having about 2000ms.
8. In the middle column, the signal strength (RSSI) should be one of the highest, i.e. > -40 dBm
9. To be sure it’s your new BLE Tag, wrap it with the aluminium foil, and check if the signal strength (RSSI) drops considerably (i.e. < -70).
10. Annotate the mac address.

Steps 7 and 8

Step 9





22. System's settings

As anticipated, the CAT AT THE DOOR system won't magically work right after assembling it: Tuning is needed!
This has to do with differences between each installation, cats' names, mac address of the BLE tags, etc.

Parameters that are more likely to differ on each system, are into two json files*: config.json* and *ble_tags.json*

On below tables are listed these parameters, with the proposed value and some basic information on the parameter meaning.

ble_tags.json :

Parameter (dict key)	**Sub-parameter**	**Default value**	**My value**	**Info**
"0"	"name"	""	"MACCHIA"	Name of your 1st cat
	"mac"	"00:00:00:00:00:00"	"D0:CD:D1:15:8E:6C"	BLE Tag mac address of your 1st cat
"1"	"name"	""	"FREYA"	Name of your second cat
	"mac"	"00:00:00:00:00:00"	"ED:07:DF:7D:AE:6C"	BLE Tag mac address of your 2nd cat

Notes:

1. Maintain the quotes, the curly parentheses and the commas.
2. The file ble_tags.json, with the same content, must be available at stationary and portable devices; If this is not the case, the system will certainly not work or misbehave.
3. The ble_tags.json file can be expanded by adding more cats, just keep the same structure. The code is already organized to handle 4 cats. Displaying more than 4 cats' names simultaneously isn't nice.
4. By adding more cats, the ToA (time-on-Air) must be calculated, to check compliancy with the eventual regulations of the Country.

This is how the ble_tags.json looks like:

```
{
    "0": {
        "name": "MACCHIA",
        "mac": "D0:CD:D1:15:8E:6C"
        },

    "1": {
        "name": "FREYA",
        "mac", "ED:07:DF:7D:AE:6C"
        }
 }
```

config.json (part 1 of 2):

Parameter (dict key)	Sub-parameter	Default value	unit	Range	Info
wifi	*enable*	1	Bool	0 / 1	1=enable, 0=disable. Wi-Fi is only used by the portable device to synchronize date and time
	ssid	your_ SSID	String		Your Wi-Fi ssid name
	password	your_ PSWD	String		Your Wi-Fi password
time	*timezone*	1	Int	-12 ~ +14	Your Country time zone vs UTC
	dst	1	Bool	0 / 1	1=enable, 0=disable. Indicates whether your Country applies Day Saving Time
buzzer	*enable*	1	Bool	0 / 1	1=enable, 0=disable. When enabled, it buzzes when your cat is waiting at the door
	volume	1	Int	0 ~ 9	Buzzer volume
	repeats	6	Int	0 ~ 50	Number of repeats. Each time the buzzer makes 4 very short beep, and each repeat happens every *lora_interval_s* seconds.
scanners	*ble_scan_time_secs*	6	Secs	≥ 4	Scanning time to detect BLE tags
	ble_rssi_threshold	-100	dB	< -50	Threshold filtering out BLE tags with signal strength weaker than the threshold
	radar_scan_time_interval_secs	10	Secs	≥ 10	Cadence for the stationary device on checking the radar status; The scanner OUT pin is linked to an input interrupt, that set a variable that remains clamped for the time of this parameter
	radar_ignor_period_s	120	Secs	> 0	Period the radar detection status is not considered, after the stationary device boots. It allows the cats to move away the radar range
	radar_keep_signal_period_s	60	Secs	> 4 * *radar_scan_time_interval_secs*	Period the radar detection status is maintained, after a detection (gives time the portable device to "see" it)
system	*lora_interval_s*	15	Secs	See note2	Cadence for the portable device to connect with the stationary device. This value should be slightly higher than *radar_scan_time_interval_secs*
	timeout_m	150	Mins		Period for portable device to enter deepsleep (extra low power consumption) in the case it gets forgotten switched-on.

config.json (part 2 of 2):

In this second part of the config.json file, there are several parameters with range = *Fix*. These parameters must not be changed; otherwise, the system will stop working; These are listed in the file to keep them grouped and insulated from the code.

Parameter (dict key)	Sub-parameter	Default value	unit	Range	Info
oled_ display	*oled_enable*	0	bool	0 / 1	1=enable, 0=disable. Only for development
	oled_backlight_pin	36	Int	Fix	MCU GPIO pin wired to the Oled backlight
	oled_rst_pin	21	Int	Fix	MCU GPIO pin wired to the Oled reset
	oled_sda_pin	17	Int	Fix	MCU GPIO pin wired to the Oled I2C SDA
	oled_scl_pin	18	Int	Fix	MCU GPIO pin wired to the Oled I2C SCL
	oled_brightess_level	3	Int	0 ~ 4	Oled brightness (PWM on *oled_backlight_pin*)
	oled_i2c_address	0x3C	Hex	Fix	MCU address for I2C with Oled display
lora	*spi_bus*	1	Int	Fix	MCU SPI bus (hardware SPI)
	clk_pin	9	Int	Fix	MCU GPIO pin wired to SX1262 SPI clock pin
	mosi_pin	10	Int	Fix	MCU GPIO pin wired to SX1262 SPI mosi pin
	miso_pin	11	Int	Fix	MCU GPIO pin wired to SX1262 SPI miso pin
	cs_pin	8	Int	Fix	MCU GPIO pin wired to SX1262 SPI cs pin
	irq_pin	14	Int	Fix	MCU GPIO pin wired to SX1262 SPI clock pin
	rst_pin	12	Int	Fix	MCU GPIO pin wired to SX1262 SPI reset pin
	dio1_pin	13	Int	Fix	GPIO pin used by LoRa signalling to the ESP32-S3 events or interrupts
	frequency_MHz	868	Int	See note1	LoRa frequency: Regulated by Countries Same value must be devices
	bw_KHz	250.0	Float	125.0, 250.0, 500.0	LoRa Bandwidth. Same value must be devices
	tx_power	5	Int	-9 ~ +22	Initial LoRa transmission power, in dBm
	spread_factor	9	Int	7 ~ 12	LoRa Spread Factor. Same value must be devices
	code_rate	7	int	5 ~ 8	LoRa Coding Rate. Same value must be devices
	duty_cyle_percent	1.0	float	See note2	LoRa Duty-Cycle: Regulated by Countries

Notes:

1. Parameter ***frequency_MHz***: The legal license-free frequencies for the LoRa transmissions are Country regulated:
    - Europe: LoRa can operate in the 433.05 to 434.79 MHz (EU433) and 863 to 870 MHz (EU863) bands.
    - North America: LoRa typically uses the 902-928 MHz (US915) band.
    - Asia: LoRa often uses the 433 MHz band (e.g., AS433) and sometimes the 915-928 MHz band (AS923).
    - Australia: The 915-928 MHz (AU915) band is common.
    - Other regions: LoRa can also operate in other bands like 865-867 MHz (IN865) in India.

    In The Netherland, where I live, the allowed license-free LoRa frequency is 868

2. Parameter ***duty_cyle_percent***: Most of the Countries limit the hourly time of each LoRa device is transmitting.
    For instance, in The Netherland, the allowed duty-cycle is 1%: This means the total Time-on-Air, per hour, must be no max than 36 seconds.
    Duty-cycle individually applies on each device.
    In the CAT AT THE DOOR system:
    a) Time-on-Air depends on the LoRa related settings, number of cats and the period between each connection (***lora_interval_s***). Some charts as guidance on next pages.
    The code automatically checks if the settings warrant the set *duty_cycle_percent*, that gets eventually corrected 😊.
    b) The stationary device is the one with higher ToA, as it transmits more data than the portable one; The code cross-checking, and below charts, are referring to the stationary device case.

3. Below listed LoRa parameters must be identical on the stationary and portable devices, differently the LoRa communication doesn't work:
    - frequency_MHz
    - spread_factor
    - bw_KHz
    - code_rate

4. Check following info to properly set the LoRa parameters.

Choosing the right LoRa settings

LoRa settings are important:
- To conform to your Country's regulation.
- To ensure reliable communication.

Check your Country regulations for:
- **frequency_MHz** (Note1 on previous page)
- maximum **duty-cycle.**

Duty-cycle is the ratio between the Lora Time-on-Air and the time of one hour.
It is regulated by most of the Countries, typically with value of 0.1%, 1% or 10%.
In the Country where I live, The Netherland, the maximum duty-cycle is 1%: This means the LoRa ToA (Time-on-Air) must be maximum 36 seconds per hour.

ToA depends on:
- LoRa communication parameters (most of them are available in config.json)
- Exchanged data quantity (i.e. number of cats)
- How often the communication is established (parameter *lora_interval_s* in config.json)

LoRa Communication Parameters

These parameters need to be adjusted to balance communication quality, regulation and power consumption. For LoRa communication,

frequency_MHz: The radio frequency used to transmit and receive signals, expressed in megahertz (MHz). It must match between communicating devices. Common frequencies include 433, 868, and 915 MHz, depending on your Country.

bw_KHz (Bandwidth): The width of the frequency range used for transmission, expressed in kilohertz (kHz). Typical values are 125, 250, or 500 kHz. Wider bandwidth reduces ToA (and duty-cycle) but also reduces the signal range and sensitivity.

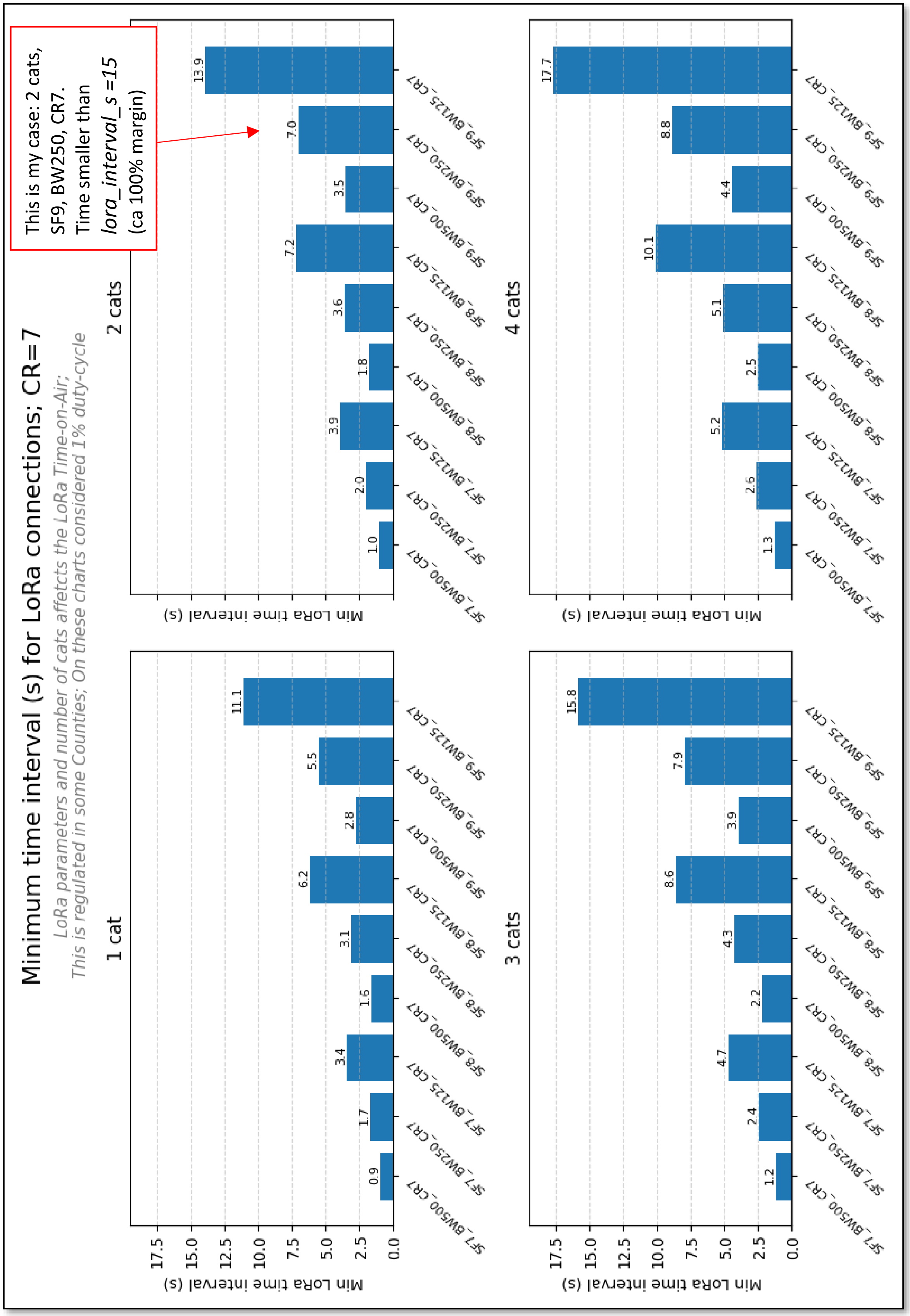
spread_factor (SF): Determines how long the signal is spread over time. Values range from 7 (fast, less range) to 12 (slow, long range). Higher SF increases reliability and range but increase ToA (and duty-cycle).

Code_rate (CR): Adds redundancy to help correct errors in transmission. Expressed as 4/x, where x = 5 to 8 (e.g., cr=5 means coding rate 4/5). Higher values improve error correction but increase ToA (and duty-cycle).

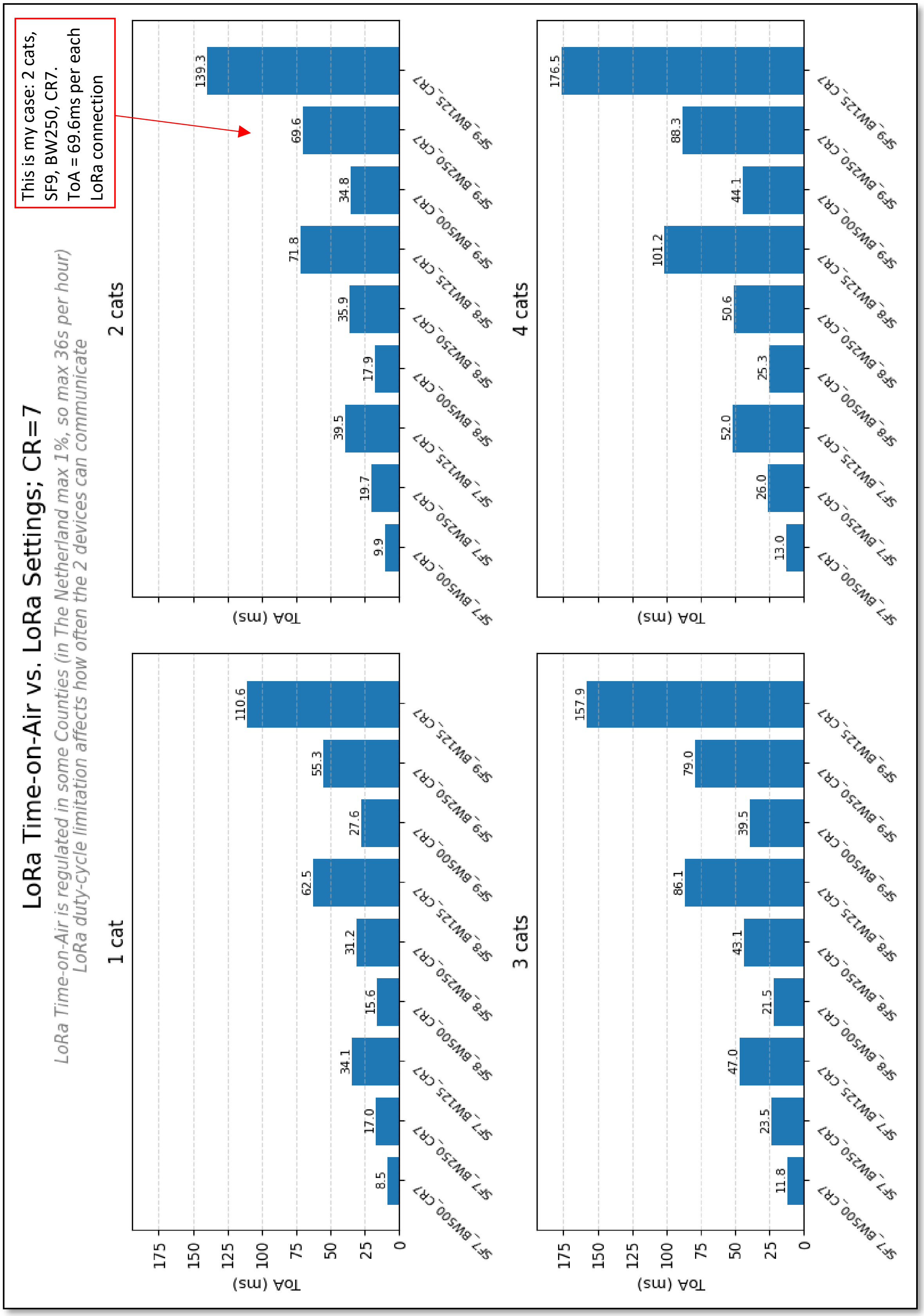
Directions:
For longer range or noisy environments, use: spread_factor = 10~12, bw_KHz = 125, cr = 5–7
For faster communication over shorter distances: spread_factor = 7~9, bw_KHz = 250~500, cr = 5
Higher SF and CR, and lower bw_KHz, increase airtime and power usage (lowers the battery autonomy).
Starts with the default values; In case the LoRa RSSI value, plotted on the EPD display, stays below -110 then you might need adjusting the LoRa parameters.
Consistency: Always configure both devices identically

Below chart shows the minimum *lora_interval_s* depending on how many cats you have and LoRa settings. Parameters are ordered by Spread factor (SF) then by Bandwidth (BW), and CR=7.

CR=7 (Slower, stronger forward error correction)



Previous chart is derived from the Time-on-Air, calculated for the different cases:



ToA is calculated by a realistic occurrence of the different strings sent by the stationary device.

23. Troubleshooting

Below issues were encountered during this project:

1. Radar doesn’t read through the door.
2. Radar senses the movements into the room.
3. Radar senses without clear reason.
4. LoRa symbol never visible on display.
5. LoRa symbol disappears from the display.

Troubleshooting solutions, or at least directions:

1. Finding the radar spot is one of the challenges of this project. For this reason, the explanations for the Stationary device placement and those to tune the radar are covered in detail.

   Anyhow, if you couldn’t find the sweet spot in your use case (the radar does not senses through the door or through the wall next to the door), here are suggested a few alternatives:

   - If the outside of the door is well protected from bad weather, the entire stationary device could be placed outdoors. This requires the possibility to power it via a USB-C cable.
   - If the door’s exterior is not weather-protected, you can place only the radar module outside. In this case, you’ll need to build a small enclosure and connect it to the rest of the stationary device using three wires.

   In both scenarios, placing the radar outside significantly increases its sensitivity.

2. To prevent the radar from sensing into the room, like you passing by the stationary device, it is sufficient to shiel the stationary device. This aspect is covered in the “Assembly steps-by-step”.

3. Prevent the radar from moving within the Over_cover (box containments): The stationary device movements/vibrations within the box are interpreted by the radar as object movements. Add a piece of sponge to keep the device forced against the wall.
   Radar senses reflection from metal and water: It provides false positive when raining of when the outer surface (wall/door/frame) has moving droplets.

4. The LoRa symbol appears on display if the LoRa connection is successful. This is verified at every inquiry sent to the stationary device.

   If the symbol is never visible, double check the config.json file having the same values on both devices. If that’s the case, check if the LoRa antenna cable is connected to the IPEX connector.

5. The LoRa symbol appears on display if the LoRa connection is successful. This is verified at every inquiry sent to the stationary device.

   Move the portable device closer to the (powered) stationary device: When the two devices are within 20 centimetres of distance, check if the symbol becomes visible and if the value underneath (LoRa RSSI).

   The expected RSSI should be > -55; If that's not the case free up the last 25~30mm of the antenna cover from the aluminium tape.

   If the symbol disappears while moving the portable device away from the stationary, you might increase the *lora spread_factor* (default value is 9, max value is 12). Each unit increased by ca 2.5dB, at the cost of increasing the LoRa connection ToA (time-on-Air).

   If the max power is still not enough you might consider decreasing the lora bandwidth *bw_KHz* (default is 250.0, next value is 125.0). This action results will increase the LoRa transmission time.

   If this is not enough, it might suggest an issue with the

   Just for reference, I live in a 2 floors house, and the portable device connects well to the stationary also by crossing one (concrete) floor, while using the default LoRa settings

   .

24. How to use it

Once the cats have left the house, switch both the devices on.

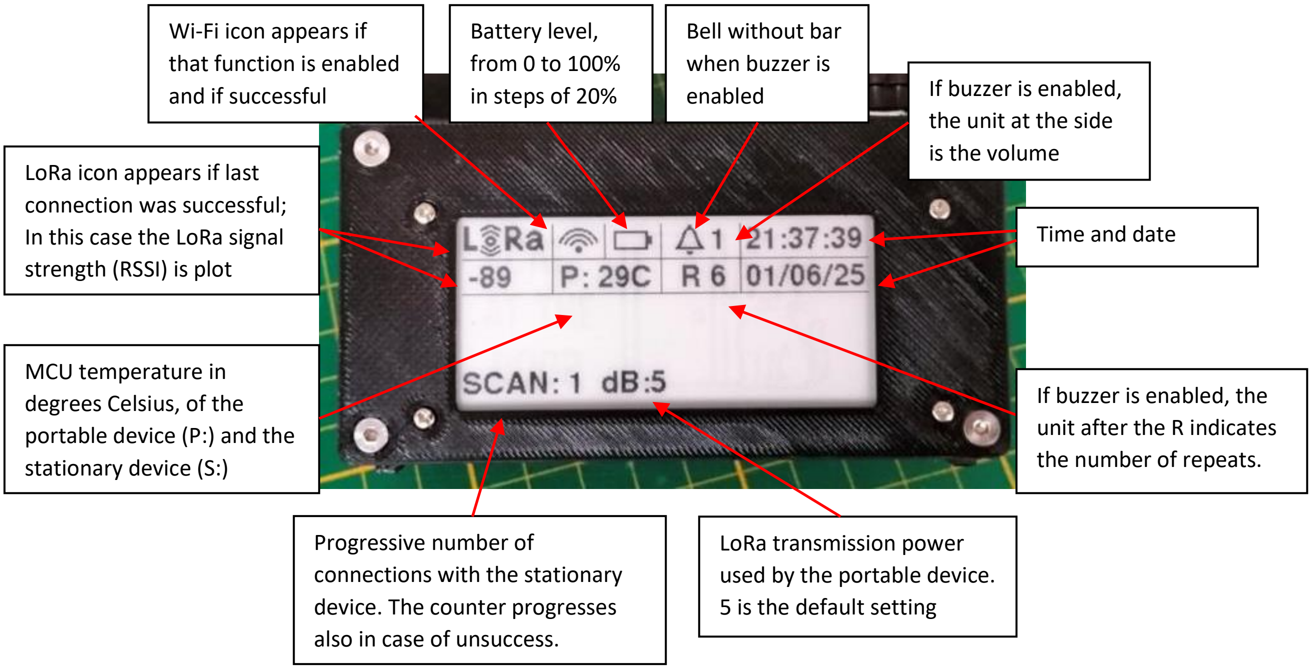
The stationary device:

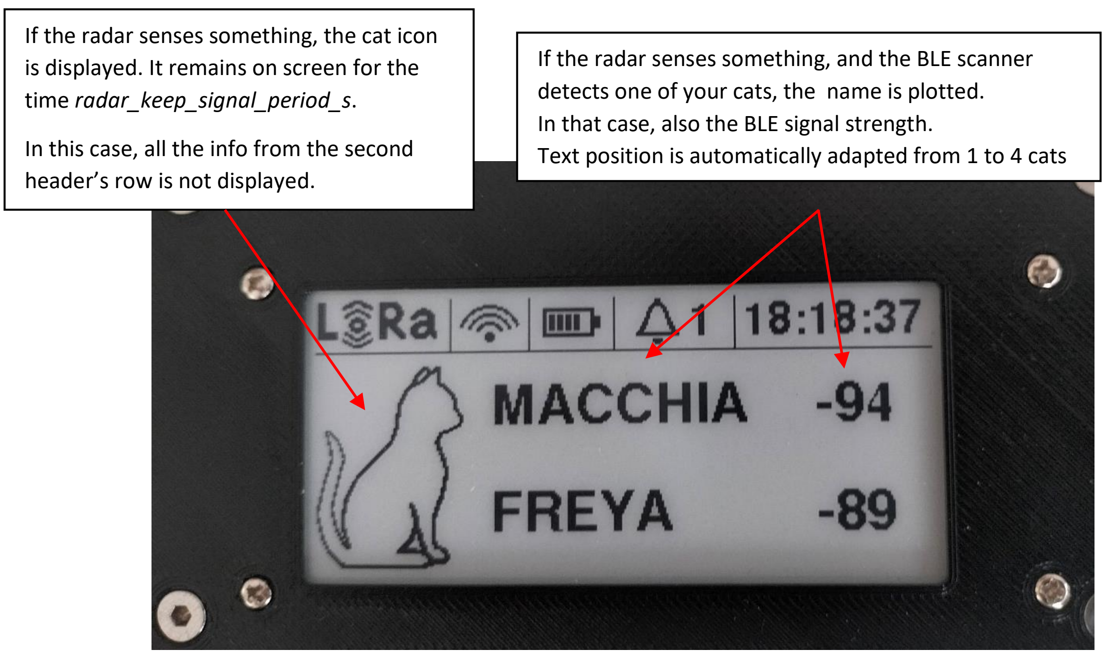
1. starts in an idle state for a given time (parameter radar_ignor_period_s , default is 120 seconds), allowing the cats to move out from the radar's sensing range.
2. periodically checks the radar signal status (default interval is 10 seconds).
   a. If a cat comes back and waits in front of the door (roughly in front of the stationary device), it is detected by the radar
   b. Each time the radar senses motion, that information is retained for a certain duration (default is 60 seconds).
   c. when the radar is triggered, the device scans for BLE tags; The scanning duration and the minimum required signal strength are configurable.
   d. If a detected MAC address matches one of your cats' tags, this information is stored.
3. replies to each received query, proving below info:
   a. If idling.
   b. MCU temperature.
   c. Whether the radar has detected any motion / presence.
   d. Which of your cats are waiting at the door.

The portable device:

4. Periodically queries the stationary device (default interval is 15 seconds).
5. Uses lightsleep mode (low power consumption) in between the connections to the stationary device.
6. It displays system information, such as:
   a. If LoRa connection is successful, and the signal strength.
   b. If Wi-Fi connection is successful.
   c. battery level.
   d. If the buzzer is active, its volume, and the number of buzzing repeats.
   e. date and time.
   f. MCU temperature, in degrees Celsius, of both devices (S=stationary, P=portable).
   g. Number of scans.
7. If radar detection, a cat icon is displayed.
8. If a cat's tag is detected, the related cat's name is displayed and the tag's signal strength. This provides rough estimate of the cat's distance from the door (helps realizing when the tag's battery needs replacing).
9. Activates the buzzer (if enabled in the settings) for a few beeps (default is 6).
   a. To stop the buzzer, press the knob.
   b. After stopping, a reminder will appear after 2 minutes.
   c. The reminder remains visible until cleared (long pressing the knob, until the display refreshes).
10. If the time-out is reached (default 150 minutes), the device plots a sleeping image to the screen and enters in Deep sleep mode (extra low-power consumption). To exit the deep sleep, switch it off and then on.

Display elements





Quick settings (portable device)

If the knob is kept pressed when the device is power on, a quick setting menu is activated.

Parameters available in the quick settings menu:

MENU ITEM	unit	Range	Info
INFO	Bool		Plots a short summary of the settings for the below parameters
ENABLE	Bool	0 / 1	Refers to the buzzer enable. 1=enable, 0=disable. When enabled, it buzzes when your cat is waiting at the door
VOLUME	Int	0 ~ 9	Refers to the buzzer volume
REPEATS	Int	0 ~ 50	Refers to the buzzer repeats. Number of repeats. Each time the buzzer makes 4 very short beep, and each repeat happens every *lora_interval_s* seconds.
EXIT….			Saves the settings Quits the menu' Starts the normal mode

How to use the quick menu':

1. Due to the slow reactivity of this display technology, after each action give some time to the display to update.
2. Rotate the knob to move to the next MENU ITEM or to change the parameter value.
3. Press the knob to confirm the selection.
4. Press the knob to access the MENU ITEM.
5. Press the MENU ITEM to close the INFO mode.

25. Q&A

1. Is Wi-Fi necessary?
   No, Wi-Fi is only used to retrieve the correct date and time from the internet.

2. How many cats can this system manage?
   The system can manage several cats; however, the display is optimized for up to 4 cats (more has not been tested).
   If more than two cats, it's likely the LoRa polling interval will need to be increased.

3. Why the cat detection has delay?
   There always is some delay from the cat approaching the radar and the notification.
   When using the default settings:
   a) scanner is activated every 10 seconds.
   b) BLE scanning lasts 6 seconds.
   c) BLE communication every 15 seconds.
   The smallest delay will be if c starts 1 sec right after a, resulting in 16 seconds' delay (a+b=16).
   The longest delay will be if c starts right before b ends, resulting in 31 seconds' delay (a+b +c=31).
   The expected average delay is about 23 seconds.

4. Is it possible to reduce the detection delay?
   A shorter delay might be possible to achieve; An average delay of 23 seconds is anyhow so much shorter compared when we forget the cats are still outside 😊

5. How long does the Tile Sticker battery last?
   According to the manufacturer, the battery lasts up to 3 years.
   https://www.life360.com/en-nl/tile-trackers/product/black-sticker

6. How do I replace the Tile Sticker battery?
   The battery of this small device is not replaceable.
   When the time comes... I'm quite sure we'll find a hacking method on the internet. 😊

7. Is the LoRa frequency country-dependent?
   Yes, the legal license-free frequencies for LoRa transmissions are country-regulated.
   (See the "System's settings" chapter.)

8. Is the LoRa ToA (Time-on-Air) regulated?
   Very likely—please double-check if your country imposes such restrictions.

9. How much battery life does the portable device have?
   Battery autonomy depends on various factors, but a full charge typically lasts between 40 to 90 hours.

10. How do I charge the battery?
    The portable device integrates a charging circuit. To charge, connect a phone charger to the USB-C port and switch the device on.
    Charging only starts when the battery level is very low.

11. Is it possible to disable the buzzer function?
    Yes. Press the knob while powering on the device.
    When a menu item appears, turn the knob one step at a time and wait for ENABLE to appear. Press the knob to enter the parameter, turn to 0, press to confirm, then turn to EXIT… and press again to quit.
    The device remembers the setting after reboot. Use the same method to re-enable the buzzer.

12. Can I adjust the buzzer volume?
    Yes, through the same startup menu (press the knob while powering on).
    Select VOLUME, turn to the desired value (0–9), confirm, and EXIT.

13. Can I change the number of buzzer repeats?
    Yes, also via the startup menu.
    Select REPEATS, set the value (0–50), confirm, and EXIT.

14. How do I stop the buzzer when it is buzzing?
    When the buzzer starts, simply press the knob. An "ACKNOWLEDGED" message will appear on the display.

15. How do I clear the REMINDER message?
    When the REMINDER is displayed, press and hold the knob for a long time—until the next display refresh.

16. The portable device reboots by itself. Why?
    Both devices use a Watchdog Timer (WDT) that forces a reboot if the device stalls due to voltage issues, temperature, software bugs, etc.
    Auto-rebooting is a safety patch. These stalls were present since the project start, and the root cause is still unknown.

17. Does the buzzer activate when only the radar senses the cat?
    No. With the current code, the buzzer only activates if the radar senses motion and a BLE tag from one of your cats is detected.
    Radar alone will only show a large cat icon on the display.
    In our case, the BLE tag is necessary, because other cats often pass by.
    A future code update might include buzzer activation based on radar-only detection (this is on the To-Do list).

26. To do list

As per today, 09/06/2025, the system is up and running since 01/06/2025, and it's working as desired.

Said that, I still have in mind some additions that aren't implemented yet:

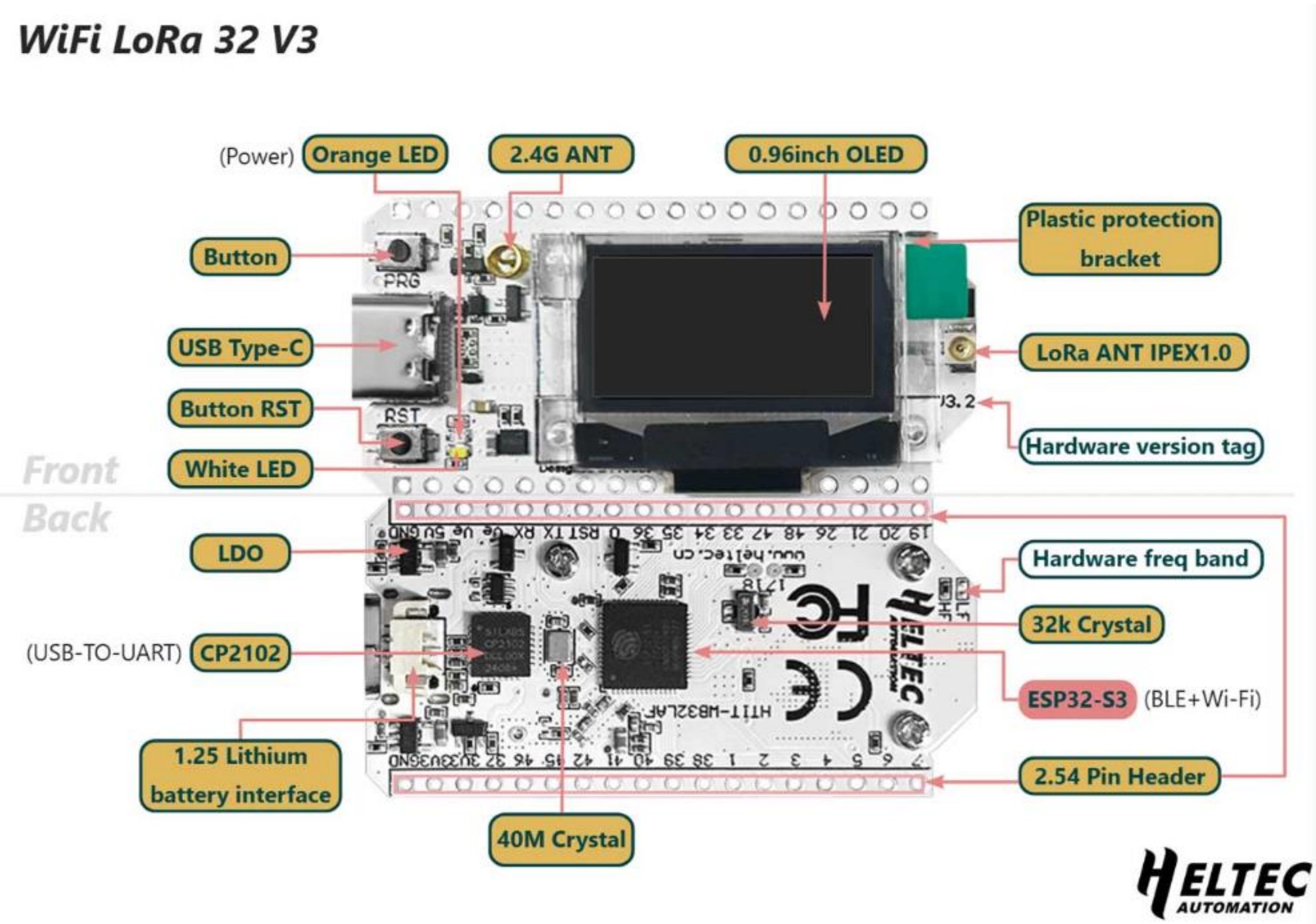
1. Add more parameters to the setting menu' accessible at the portable device booting.
2. In the portable device, add a voltage divider to sense the 5Vdc (J2, 2); The device can detect when energized via the USB-C, allowing a "charging" mode in which the other functions (i.e. LoRa transmissions) could be kept disabled.
3. Add a setting for a simplified version of this project (no BLE Tag), and the buzzer links to the radar sensing something.
4. Add a web server as additional UI interface.

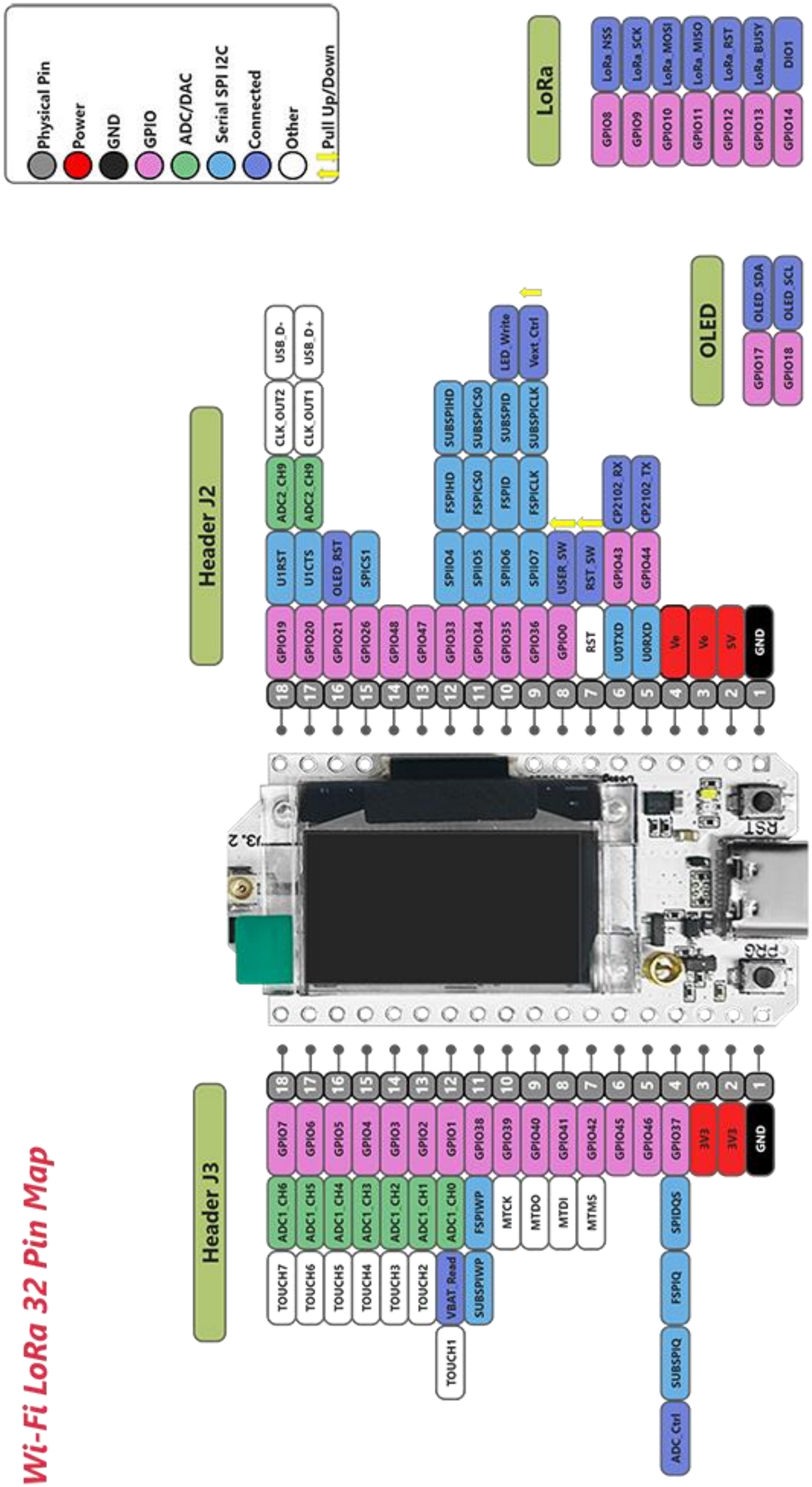
27. Heltec Wi-Fi LoRa 32 V3 board

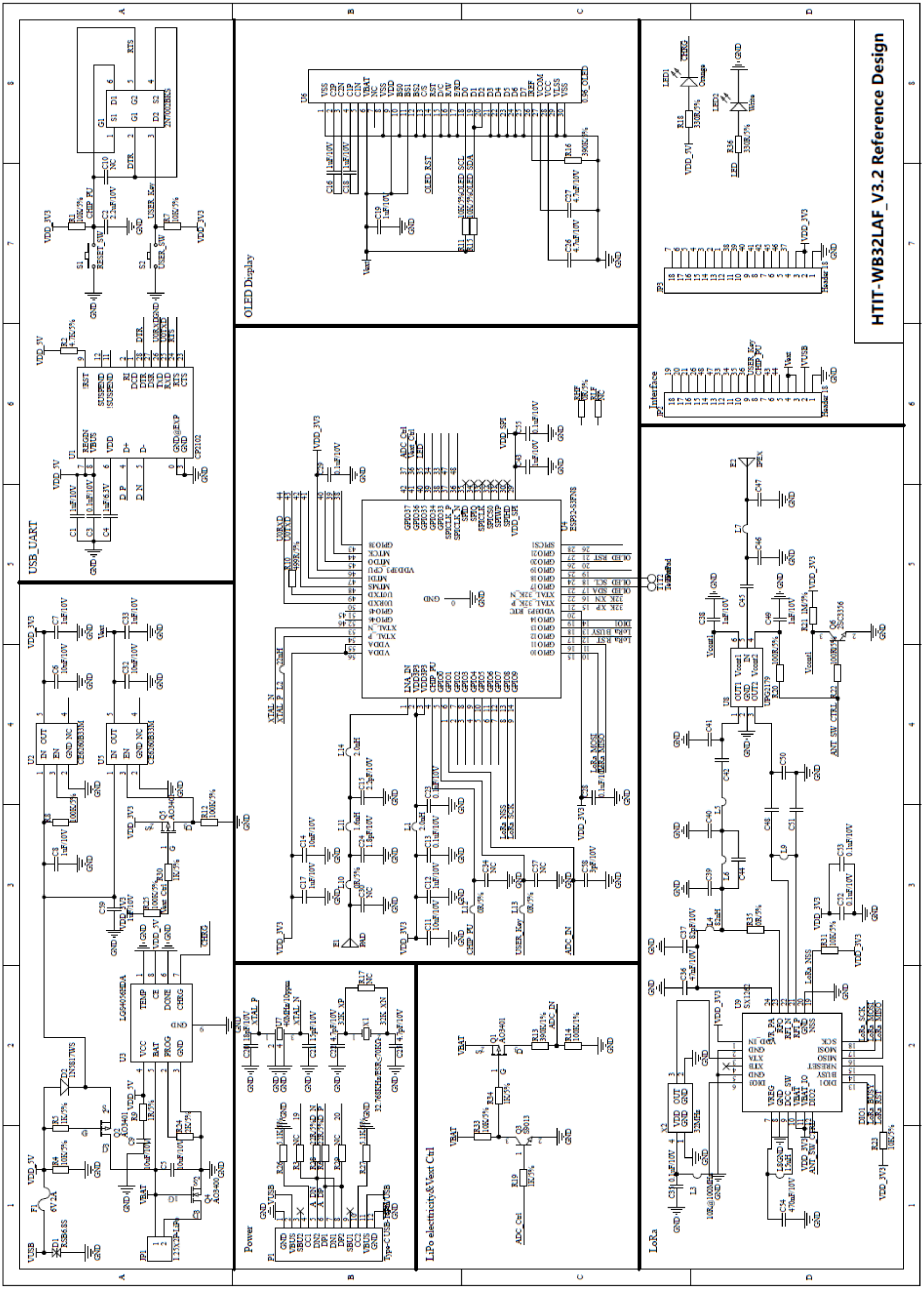
The Heltec Wi-Fi LoRa 32 V3 board has been chosen for this project due to its useful features:

- **LoRa** (Long Range) communication protocol, with its dedicated antenna. This enables reliable communication between the two devices, even across different (concrete) floors.
- **BLE** (Bluetooth Low Energy) protocol is the core of this system. It is used to scan for the BLE tags.
- **Battery management**, which is useful for the portable device. It supports battery operation, charging, and charge level monitoring.
- **ESP32-S3** microcontroller, which allows for low power consumption between periodic communications with the stationary device (using the light sleep function).
- **Wi-Fi**, which is optional, but allows the system to synchronize the date and time via an internet NTP service.
- A small **OLED display**, which is especially useful during code development and debugging.

Documentation at: https://heltec.org/project/wifi-lora-32-v3/




Wi-Fi LoRa 32 Pin Map
Physical Pin
Power
GND
GPIO
ADC/DAC
Serial SPI I2C
Connected
Other
Pull Up/Down
LoRa
GPIO8 LoRa_NSS
GPIO9 LoRa_SCK
GPIO10 LoRa_MOSI
GPIO11 LoRa_MISO
GPIO12 LoRa_RST
GPIO13 LoRa_BUSY
GPIO14 DIO1
OLED
GPIO17 OLED_SDA
GPIO18 OLED_SCL
Header J2
18 GPIO19 U1RST ADC2_CH9 CLK_OUT2 USB_D-
17 GPIO20 U1CTS ADC2_CH9 CLK_OUT1 USB_D+
16 GPIO21 OLED_RST
15 GPIO26 SPICS1
14 GPIO48
13 GPIO47
12 GPIO33 SPIIO4 FSPIHD SUBSPIHD
11 GPIO34 SPIIO5 FSPICS0 SUBSPICS0
10 GPIO35 SPIIO6 FSPID SUBSPID LED_Write
9 GPIO36 SPIIO7 FSPICLK SUBSPICLK Vext_Ctrl
8 GPIO0 USER_SW
7 RST RST_SW
6 U0TXD GPIO43 CP2102_RX
5 U0RXD GPIO44 CP2102_TX
4 Ve
3 Ve
2 5V
1 GND
Header J3
18 GPIO7 ADC1_CH6 TOUCH7
17 GPIO6 ADC1_CH5 TOUCH6
16 GPIO5 ADC1_CH4 TOUCH5
15 GPIO4 ADC1_CH3 TOUCH4
14 GPIO3 ADC1_CH2 TOUCH3
13 GPIO2 ADC1_CH1 TOUCH2
12 GPIO1 ADC1_CH0 VBAT_Read TOUCH1
11 GPIO38 FSPIWP SUBSPIWP
10 GPIO39 MTCK
9 GPIO40 MTDO
8 GPIO41 MTDI
7 GPIO42 MTMS
6 GPIO45
5 GPIO46
4 GPIO37 SPIDQS FSPIQ SUBSPIQ ADC_Ctrl
3 3V3
2 3V3
1 GND
RST
PRG


HTIT-WB32LAF_V3.2 Reference Design
USB_UART
OLED Display
Interface
LoRa
Power
LiPo electricity&Vext Ctrl

List of GPIO used by the code, and already connected at the Heltec's board

Stationary and portable device , GPIO used by the code already connected:

Header	Pin	GPIO / Label	Purpose
		GPIO8 / LoRa_NSS	(LoRa SX1262 SPI) CS pin
		GPIO9 / LoRa_SCK	(LoRa SX1262 SPI) CLK pin
		GPIO10 / LoRa_MOSI	(LoRa SX1262 SPI) MOSI pin
		GPIO11 / LoRa_MISO	(LoRa SX1262 SPI) MISO pin
		GPIO12 / LoRa_RST	(LoRa SX1262 SPI) Reset pin
		GPIO13 / LoRa_BUSY	(LoRa SX1262 SPI) Busy pin
		GPIO14 / LoRa_DIO1	(LoRa SX1262 SPI) IRQ pin
		GPIO17 / OLED_SDA	Oled display, I2C SDA line
		GPIO18 / OLED_SCL	Oled display, I2C SCL line
J2	16	GPIO21 / OLED_RST	Oled display reset
J2	6	GPIO36 / Vext_Ctrl	Oled display backlight

Stationary device, GPIO used by the code already connected:

Header	Pin	GPIO / Label	Purpose
J2	10	GPIO35 / LED White	Onboard white led

Portable device, GPIO used by the code yet already connected:

Header		Pin	GPIO / Label	Purpose
J3		12	GPIO1 / VBAT_Read	Reads the battery voltage
J3		37	GPIO37 / ADC_Ctrl	Connects battery to reading circuit

28. Sensors' choice

Objective of this system is detecting our cats when they're out of the door and waiting for us to let them inside. On below table are listed the sensor's types considered, and the reasons leading to the final choice.

Sensor type	**Pros, cons. considerations**
Cat's microchip	Cat's microchips are passive RFID: These components can be read at no more than 10~15 centimetres of distance, not a reliable option for this application. These sensors can be reliably used in small and obliged passages, like the cat flap doors. We cannot modify the door, due to building's policy, so these cat's flap doors weren't an option for us.
PIR	(Passive Infra-Red) sensors are largely influenced by surrounding temperature. I haven't tried, but I think it will be difficult to find a setting working both in winter and in summer; Furthermore, our door points to west, so often the sun hits it theoretically reducing the PIR sensors' reliability.
Camera	I like computer vision, but I haven't even consider it as a viable option. This system must especially work in winter, where in The Netherlands there is little light; A camera based, will also require massive training to distinguish our cats from the many others in this large garden, under very different light conditions. I feared this way would turn into and endless project.
BLE tag	(Bluetooth Low Energy) are small, battery-powered devices using Bluetooth to transmit data wirelessly. This is perfect to distinguish our cats from the others, but determining the distance from the sensor is not reliable.
24GHz FMCW radar	FMCW (Frequency Modulated Continuous Wave) can determine the distance of reflective objects (metal and water); It isn't influenced much by light or temperature. Another positive aspect is that it can be place inside, as the 24GHz can still pass through (non-metal) materials. The radar is sensitive to water, making it not a good choice when raining; In our case, we prefer keeping the cats inside when raining, so still not a blocking aspect. This technology has become very popular to sense human presence, leading to a good option of choices and rather low cost. The radar module used in this project has been specifically developed to sense human's presence, yet it works well with cats as well. This seemed a good way to only sense the cats once waiting in front of the door.
Others ?	Of course, there might be the technologies out there that didn't cross my mind 😊
BLE tag + 24GHz FMCW radar	By combining the features of the BLE tag and the 24GHz FMCW radar it seemed the way to go for our specific case. The BLE scanner gets activated only in case the radar senses a movement. The system is now working since some days, and so far, it is working as desired.

29. LoRa communication

For the Lora communication parameters, and their explanation, check System settings chapters.
In this chapter, it is differently disclosed the strings exchanged by the two devices.

Portable device queries the portable device via a very short string

ck	>

Field	Presence	Description
ck		Means “check”. When this string is decoded, the stationary device replies with the “cats_status”
>	optional	This optional character is always placed at the end > LoRa RSSI signal above the max. Portable can lower the transmission power < LoRa RSSI signal below the min. Portable should increase the transmission power

Stationary device replies with two main types of strings, both having optional fields:
When no cats are detected at the door:

txx,	d	00	>

txx,	r	00	>

When cats are detected at the door:

txx,	r	01	-95	>

Field	Presence	Description
txx,	optional	When used, the “t” is always placed at the start. Stationary device MCU temperature, wherein xx is the value in degrees Celsius Comma separates from the following field, allowing temperature value identification. Info shared on 5% of the replies (4 consecutive times out of 80).
d	optional	When used, the “d” is always placed at the start, unless the temperature is also shared. Radar is deactivated, as the device is just started
r	optional	When used, the “r” is always placed at the start, unless the temperature is also shared. Radar is sensing, or has sensed, something
>	optional	This optional character is always placed at the end > LoRa RSSI signal above the max. Portable can lower the transmission power < LoRa RSSI signal below the min. Portable should increase the transmission power
00		First digit identifies the cat (cat 0 in this example). Second digit identifies the BLE not detected. In case of more cats, a comma is added: 00,10,20 (cats 0, 1 and 2 are not at the door).
01		First digit identifies the cat (cat 0 in this example). Second digit identifies the BLE being detected In case of more cats, a comma is added: 01,21 (cat 0 and cat 2 are at the door, not cat 1)
-95	Only when BLE detection	When the cat’s BLE is detected, the BLE RSSI value is sent to the portable device. This provides an indication of how close/far the cat is from the stationary device, and ultimately if the BLE Tag battery is getting too low. In case of more cats at the door, the BLE RSSI value position is right after the cat identification: As example, r01-95, 21-105 means cat 0 is at the door (01) and its BLE RSSI is -95 while the cat 2 is also at the door (21) and its BLE RSSI is -105.

Notes:

1. The number of characters has been minimized, to limit as much as possible the Time-on-Air (regulated by Country)
2. In the LoRa communication, the cats are identified by a number; Via the ble_tags.json the cat’s name and its BLE tag gets associated.

30. LoRa dynamic power adj

The parameter *tx_power*, in the lora settings, determines the initial power used by the SX1262 LoRa chip; This power level is initially used by the stationary and by the portable devices.
Once the devices are powered, such parameter will be dynamically adjusted to try to get the LoRa signal strength within and acceptable bandwidth.

With this function, the LoRa power gets reduced in case the signal is strong (>-80dBm), resulting in lower power consumption; Differently, the power gets increase if the signal is weak (<-100 dBm), to ensure proper communications.
The dynamic adjustment of the transmission power allows variation between -9 and 22dBm (full range of the SX1262 LoRa chip).

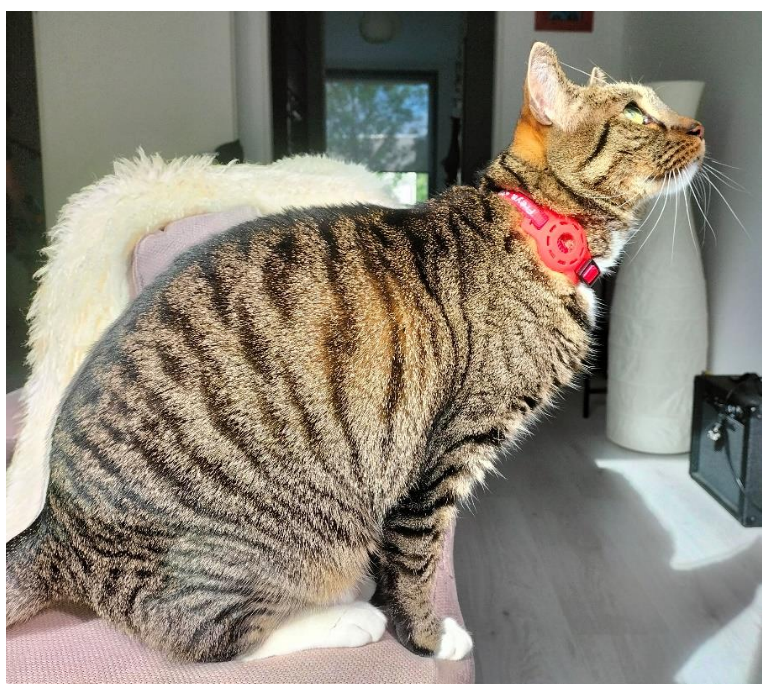
There are four main cases

1. Weak signal:
    a. The sender appends the ”<” character to the communication string.
    b. The receiver increases the transmission power by 1 dBm.
    c. If the max transmission power is reached, no further adjustments will be made.
2. Strong signal
    a. The sender appends the ”>” character to the communication string.
    b. The receiver decreases the transmission power by 1 dBm.
    c. If the min transmission power is reached, no further adjustments will be made.
3. No feedback
    a. This happens when the other device doesn’t require transmission power adjustment.
    b. The sender appends no “<” nor ”>” character to the communication string.
    c. The receiver maintains the same transmission power.
4. No RSSI signal:
    a. First it checks if no signal happens for at least 4 consecutive times.
    b. Then it increases the power to max (22 dBm), in the attempt to re-establish the communication with the other device; The other device does the same…
    c. In case of no signal for consecutive 15 times, the power is brought back to the initial value (value from config.json). This considers one of the two devices has been forgotten switched on.

This control is rather basic, to limit the transmission characters to the base minimum (limiting ToA to the max). Beside being simple, it allows the two devices to set the transmission power so that the other device receives properly.

31. Pictures collections

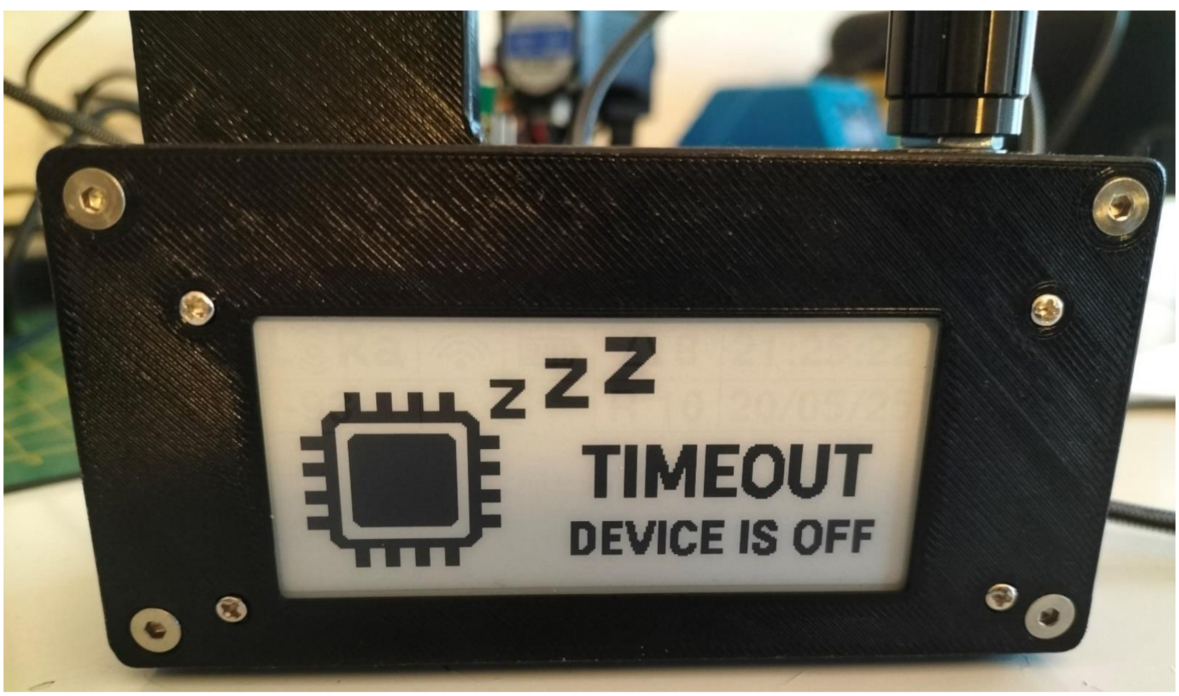
Some pictures as reference for the collar tag's sleeve size, on our cat Freya (ca 5 Kg); Freya doesn't seem annoyed by the new collar, having the pocket for the BLE tag.
At the date of these pictures, the BLE Tag for FREYA was not yet ordered, as we were testing with MACCHIA 😊



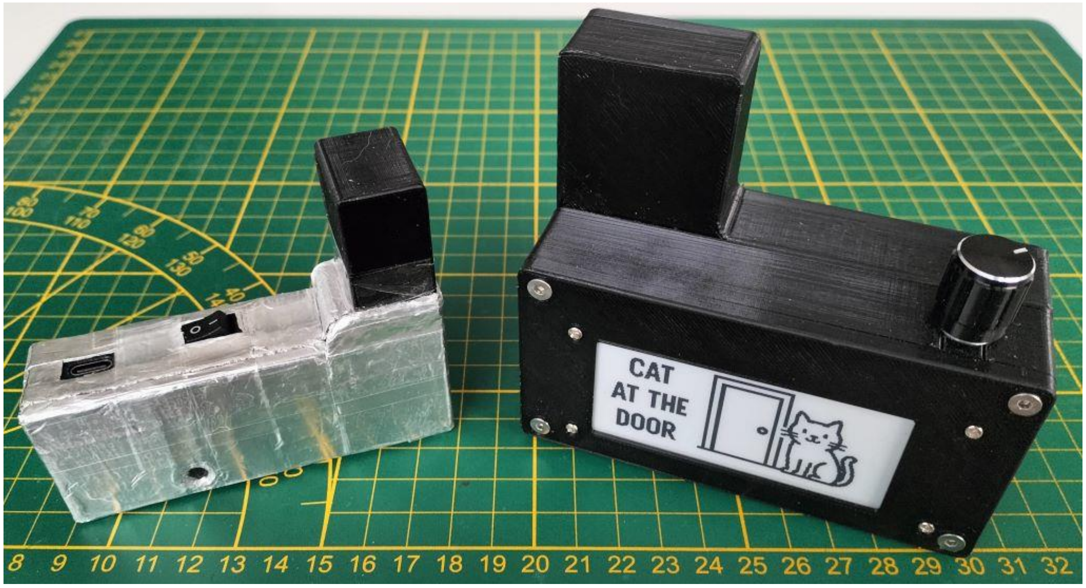
Opening page at the switch on, masking the time for the libraries loading:

Portable device: Both our cats are at the door:

Portable device in deepsleep:



The stationary device is partially taped with aluminium foil



Th small Oled display is not used

32. Credits

- Andreas Spiess, and his very informative videos about radars; I'd suggest starting with the video "24GHz Radar Presence Detector that Works (LD2410)", as of a great inspiration for this project: https://www.youtube.com/watch?v=dAzHXpP3FcI&t

- Heltec forum, where the initial precious info was found (http://community.heltec.cn/)
- ehong-tl for the LoRa communication driver (https://github.com/ehong-tl/micropySX126X)
- peterhinch for the Writer and font_to_py drivers (https://github.com/peterhinch/micropython-font-to-py/tree/master)
- MacEachran for the rotary encoder driver (https://github.com/MikeTeachman/micropython-rotary)
- Sgall17a for the encoder menu (https://github.com/sgall17a/encodermenu/tree/main)

OF course, I should also mention ChatGPT and Deepseek, always available to help 😊

33. About the author

My name is Andrea Favero.

I was born in Italy in 1971. I'm married to Raffaella, and we have two children: Luca and Alice.

Since 1994, I have worked as an engineer—primarily in R&D for companies producing small kitchen appliances, starting in 1997.

In 2015, we moved to Groningen, in the Netherlands.

In 2019, I had the opportunity to attend a Python course and quickly fell in love with coding.

In 2021, I decided to explore computer vision and Raspberry Pi by setting myself the challenge of building a Rubik's Cube-solving robot. The result is the CUBOTino robot series.

Several other projects have followed since then.

Contacts:

- I am not into social media.
- I can be reached via email: andrea.favero71@gmail.com

Published projects:

- https://www.youtube.com/@andreafavero71
- https://www.instructables.com/member/AndreaFavero/instructables/

34. Revisions

<table>
<tr><th>Rev</th><th>Date</th><th>Notes</th></tr>
<tr><td>0</td><td>03/06/2025</td><td>First release</td></tr>
<tr><td>1</td><td>09/06/2025</td><td>Code:
Added duty_cycle_percent parameter in the config.json
Added code_rate parameter in the config.json
Added validation of the lora_interval_s settings, via ToA calculation.
Improved the LoRa power adjustment method.
Instructions:
More info and guidance on LoRa parameter settings.
Charts of lora_interval_s , and Time-on-Air, as function of the number of cats and LoRa parameters settings; This provides reference for guidance.
Explained the LoRa strings exchanged by the 2 devices.
Explained the new parameters, added to the config.json.</td></tr>
<tr><td>2</td><td></td><td></td></tr>
<tr><td>3</td><td></td><td></td></tr>
</table>