Physical specifications

- Compact size, close in size to RaspBerry Pi size and not bigger than a hand size.
- Uses STM32H5 microcontrollers or any MCU that has ARM Cortex-A76 or higher.
- Has an aluminum radiator (snappy or easy to remove).
- Has a capacitor that can hold enough power alert the server about this cutoff.
- Has 2 slots for SIM cards that support (click to snap in, click to snap out).
- LPDDR4X-4267 SDRAM (8GB SKUs available at launch).
- Has internal antenna with external antenna support.
- Has a microSD card slot, with support for high-speed SDR104 mode.
- Working temperature 0 - 80 Celsius degrees.
- Gets its power directly from power source (220V), meaning : one cable, no need for adapter.
- Has USB type-C connector for data & alternative power source.
- Has a v220 fuse breaker to keep the component safe.
- Has 4 CAT7 network connector, these 4 connectors will act like (36 GPU pins) to the outside world.
- Has 2 thermostat sensors, one inside the encloser, the other one is for the surroundings (outside encloser) but not with wires, extended it legs to be at the encloser level.
- Has a Bluetooth 5.4 module.
- has 2 tactile buttons, one to reset, the other for turning on Bluetooth.
- Switcher to switch On/Off power.
- Robust design that can stand pressured / jet air (for cleaning dust and debris).
- Has 15 SMD LED indicators that support RBG colors (white , red , green and blue). Like this one

Functional specs

- Accepted programing languages: C# , C++ , Python , PHP , JavaScript.
- We prefer to read the software/OS from an SD card for easy replacing & updating.
- We'll use only 4G module for the prototype, but in production will use 3G/4G/4G+/5G module.
- Loggin events, like temperature every X min, cellular connection …etc, log every 1 minute and bundle it to be pushed to the server ever 1 hour. Attached a JSON sample of expected data to be logged. Timing will be as defined in the confecrations.
- If powerline goes off, the board will keep running for ~1 minute allowing it to send alert to the server.
- Primary connection will be to the sim with the strongest signal.
- SIMs will rely on internal antenna, but if signal is weak, will connect external antenna.
- CAT7 cables will be used to send receive data from external world, it will be our GPUs pins. 4 pins for 12V
- 15 SMD LED indicators are explained below.
- Once the tactile button pushed, start Bluetooth pairing for 3 minutes then turn off of not paired.

Mobile application

To configure the PCB board for the first time, we will connect to this PCB via Bluetooth then open a very simple mobile App that has the following inputs as (initial configuration):

- Code
- Server URL

Once these configurations are transferred to the board, the PCB will receive the (extended configuration). More explanation below. We need your advice in this part : how can we communicate & display content?

Software flow

Once the PCB is switched on, from the confecrations check if the **CODE** value:

Case 1 : code == NULL

1. Turn on Bluetooth pairing for 3 minutes. Default device name : Edara Lite V1
2. Connect to the mobile App.
3. Get (initial configurations) mentioned in previse page.
4. Confirm receiving the (initial configurations).
5. User will click on **(get configuration)** button from the App.
6. The PCB will make a POST request to ${server_url}/api/get_config/${code}.
7. The PCB will receive a json file containing the extended configurations.
8. The PCB will store these comigrations locally to be accessible as GOLBAL parameters.

Case 2 : code NOT NULL

continue running the code normally:

1. Do the logging statuses mentioned above as timed in the (extended configuration).
2. Keep listening to Pusher's events, if receive a job then execute based on (extended configuration).

Extended configurations fields are:

- Device Code [ ]
- Server URL [ ]
- Token [ ]
- Bluetooth's Device Name [ ]
- Allow push notification? [ ]
- PUSHER_APP_ID [ ]
- PUSHER_APP_KEY [ ]
- PUSHER_APP_SECRET [ ]
- Send bulk report every [ ] min
- Gather status report every [ ] min
- Close time if no response [ ] sec

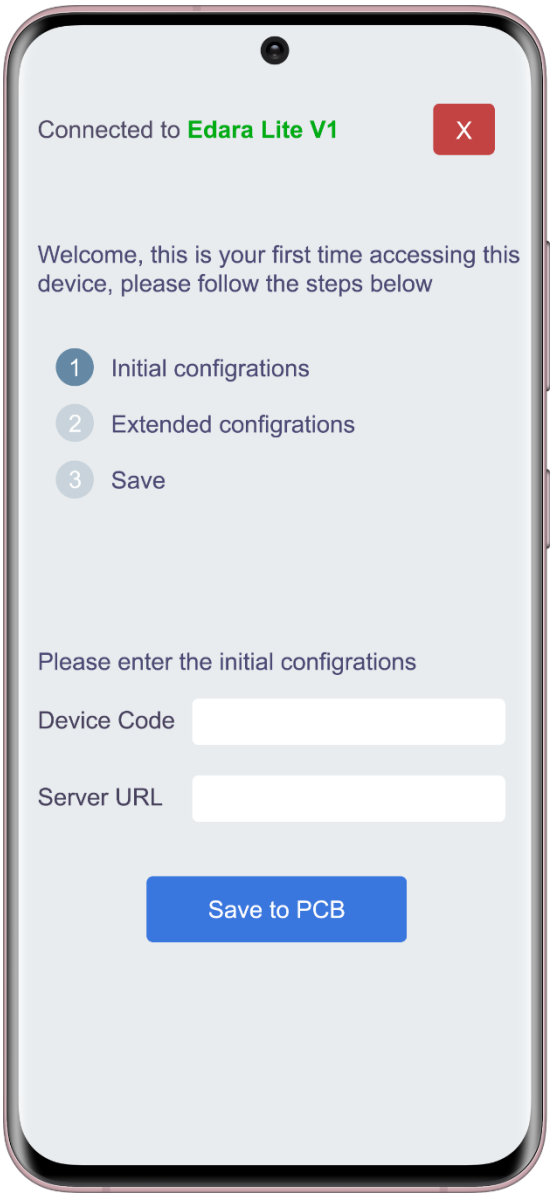
Question

Can we use (**node.js**) OR (**Node Red**) OR (**nGinx server + PHP**) on the STM32H5 MCUs directly WITHOUT installing an operating system? Answering this will decide which MCU we going to use.

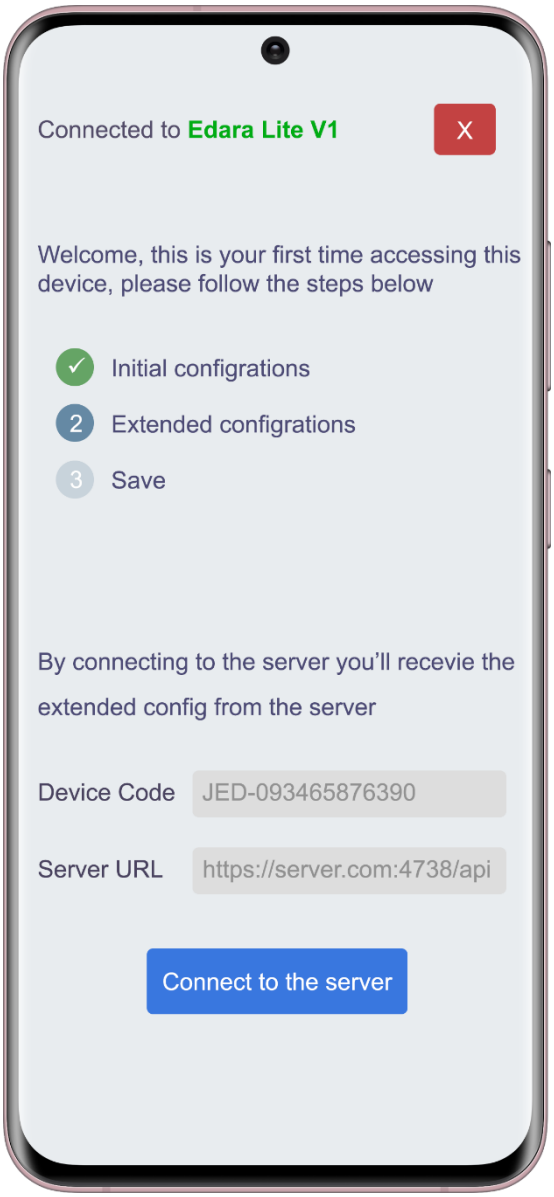
We need to use device for "general-purpose" so performance is important, yet we are not sure what path to take, making it like a server (**Node.js , Node Red , nGinx + PHP**) OR using an OS like (**Ubuntu or Android**)

Screen of the app

Step 1

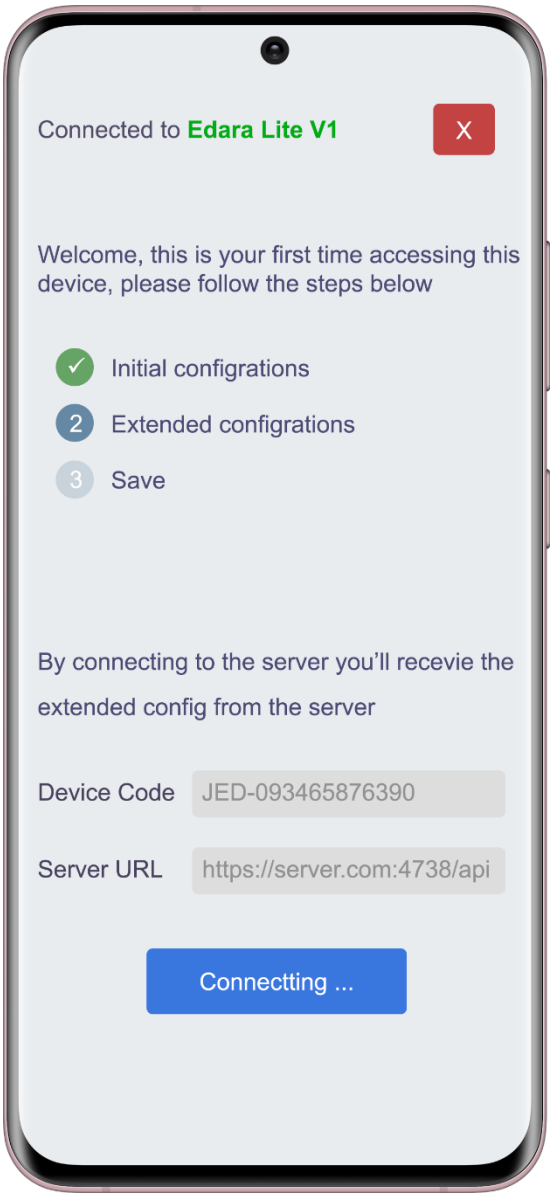


Step 2

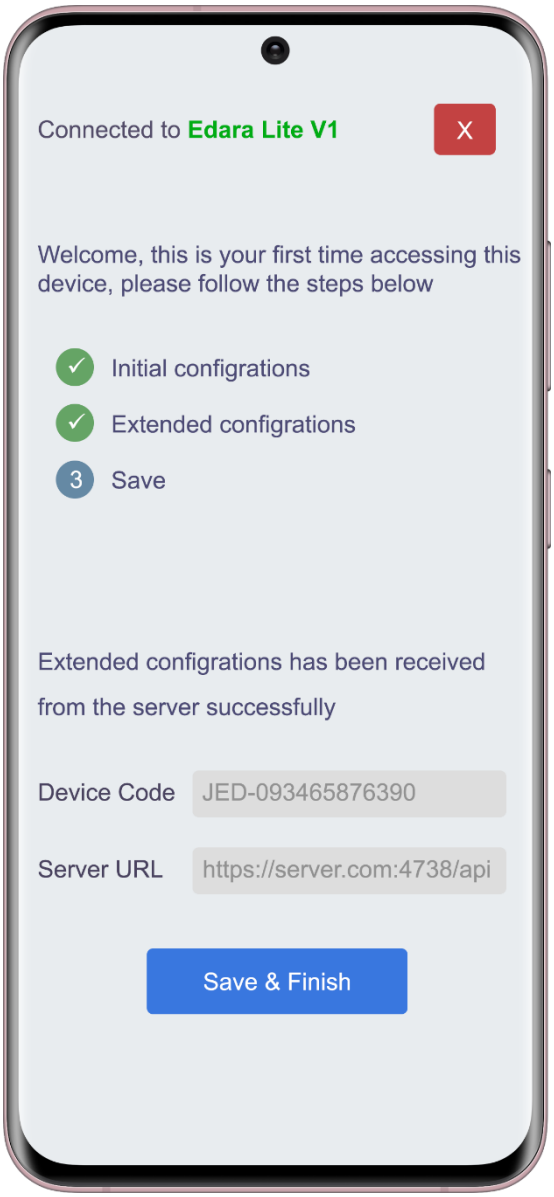


Screen of the app

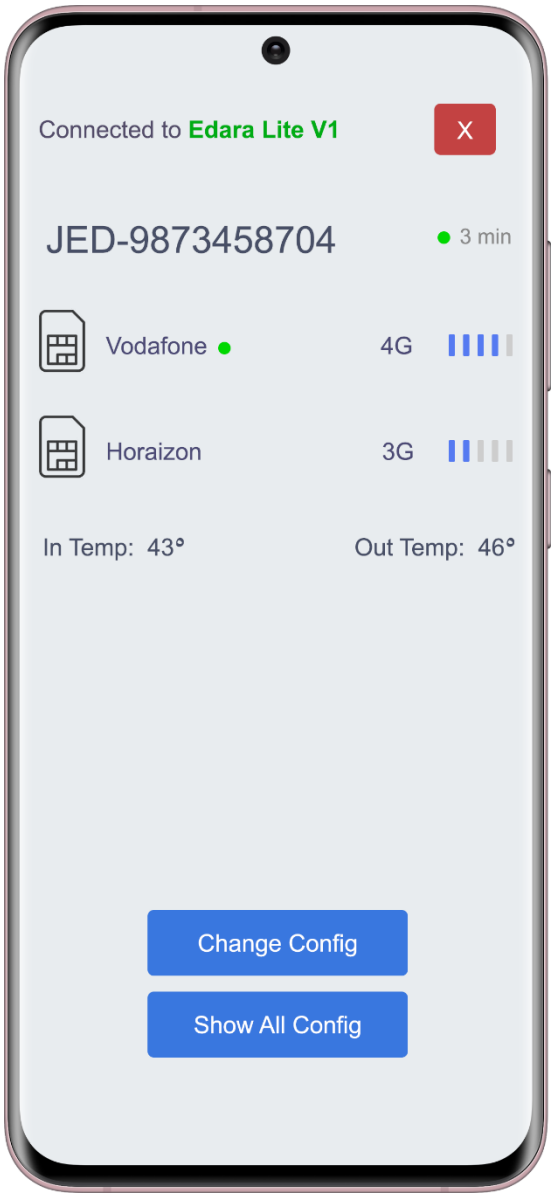
Step 3



Step 4

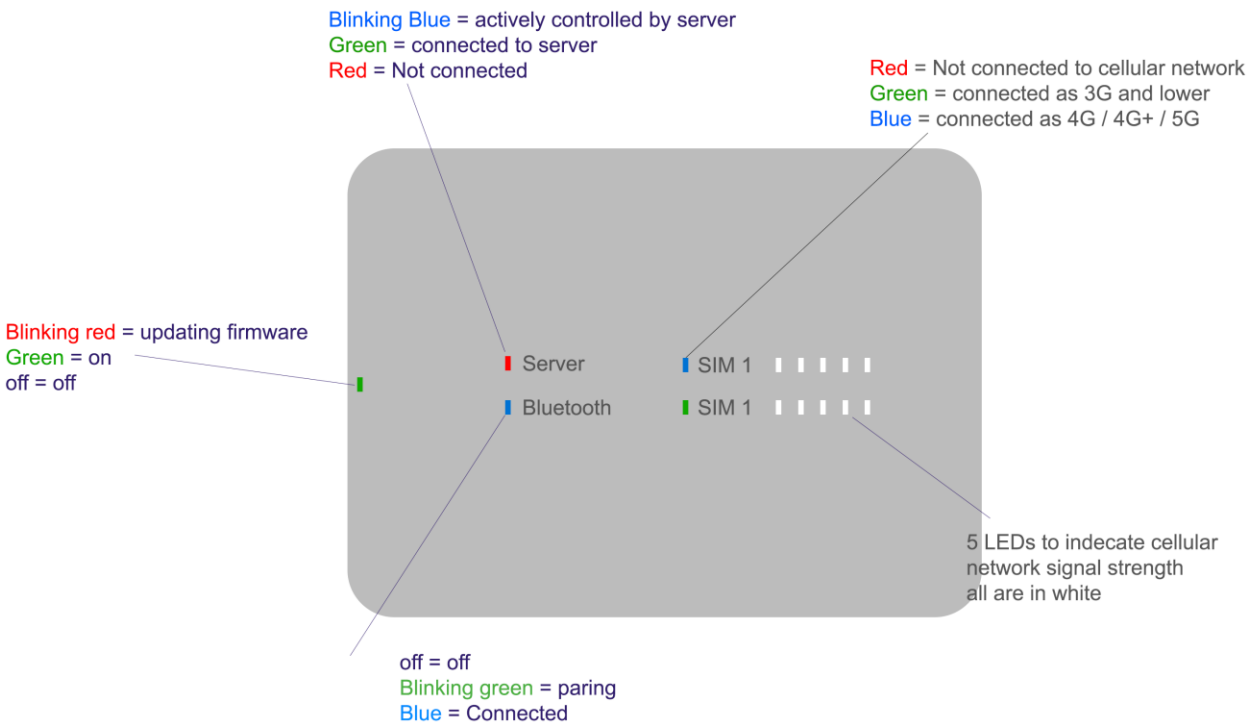


This screen is after setting up configurations



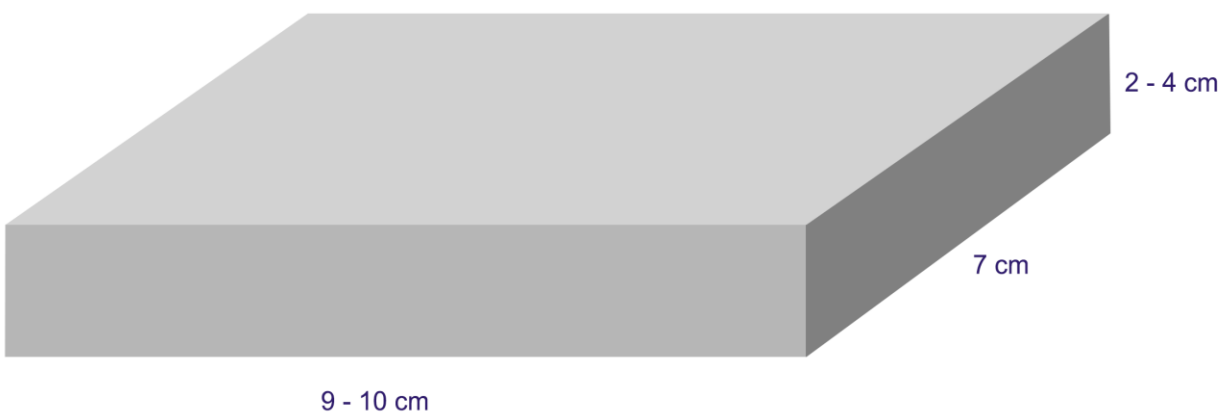
Please keep in mind that these statistics will be shown in real-time from the PCB and maximum update time will be every 1 second.

A summary visualization of the LEDs functionality



Please note the placement shown here is not mandatory. You can place and order the LEDs any where you like, if you can place it as show it will be a plus 😊

Here are the expected dimensions



Expected Deliverable

1. Consultation and finalizing this document (1 - 2 online meeting) within maximum 2 days.
2. Schematic diagram + design file
3. List of components' model name as word or excel file
4. PCB board design + design file + Gerber file
5. Mobile App (android) as APK + open source
6. Code open source
7. A working PCB

PS : I'm committed with a timeline, if we don't agree on the first one, I'll move to another freelancer.

Milestones

Milestone	Time (days)	Price ($)
Schematics	5	125
PCB	3	125
Open-source files	1	100
Android mobile App	5	100
Components' list	1	50
After I receive & test the PCB & make sure is 100% working (device & app)	30	100

You can change the marked numbers above 👆 with you actual time.

The prototype of this PCB will be manufactured by PCBWay and will be paid separately.

..... The end .....

To make sure that you really read my file, I'll contact you if you post in UpWork the following without mentioning your previse works or portfolio : I read your file and I have the answer in to your question regarding the MCU & OS and I can finish this project (as design & files) is XX days and manufacturing it using PCBWay in XX days.. then ad here your input.....