Intelligent Irrigation System for Low-cost Autonomous Water Control in Small-scale Agriculture

Deliverable D2.2b

Starter-kit for smart irrigation system – v2

Responsible Editor:	WAZIUP e.V.
Contributors:	IRD, UPPA
Document Reference:	INTEL-IRRIS D2.2b
Distribution:	Public
Version:	1.1
Date:	June 2023

Authors table

DOCUMENT SECTION	AUTHOR(S)
SECTION 1	C. Pham (UPPA)
SECTION 2	Felix Markwordt (WAZIUP) & C. Pham & G. Gaillard (UPPA)
SECTION 3	F. Markwordt (WAZIUP) & C. Pham (UPPA)

Document revision history

Version	Date	Changes
V1.1	June 15th, 2023	Public release
V1.0	June 12th, 2023	First draft version for internal approval
V0.1	June 5th, 2023	First release for review

Executive Summary

Deliverable D2.2b describes the INTEL-IRRIS starter-kit for smart irrigation systems – v2. The starter-kit consists of the low-cost soil sensor device and the versatile edge-enabled IoT gateway with all packaged configuration and add-on software for out-of-the-box deployment.

Table of Contents

1. Sensor device part **5**
1.1. 2-Watermark version 6
1.2. New PCB with solar panel support 7
1.3. WaziSense v2 7
2. Gateway part **8**
2.1. Generic WaziGate framework 8
2.1.1. New features 8
2.1.2. Preparing AI framework 8
2.2. INTEL-IRRIS specific WaziGate distribution 8
2.2.1. Real-Time Clock support 8
2.2.2. Auto-configuration features 9
2.2.3. Integration of Home Assistant 10
2.2.4. Backup/Recovery procedures 12
2.2.5. Advanced scripts for dataset support 12
3. IIWA **14**
3.1. New User Interface 14
3.1.1. Dashboard 14
3.1.2. Sensor configuration 17
3.2. IIWA parameters menu 18
3.2.1. Basic vs Advanced view 18
3.3. IIWA REST API 21

List of Figures

Figure 1 – Intelligent irrigation in the box concept 9
Figure 2 – Starter-kit with Version 1 of soil sensor device 9
Figure 3 – All the parts of the soil sensor device 10
Figure 4 – The Raspberry Pi with the LoRa WaziHat 11
Figure 5 – The OLED and web-based user interface of the WaziGate 11
Figure 6 – Preparing the components of the starter-kit 12
Figure 7 – Preparing the starter kit for shipping 13
Figure 8 – The INTEL-IRRIS WaziGate dashboard 14
Figure 9 – The OLED interface of the INTEL-IRRIS WaziGate 15
Figure 10 – Default configuration of the starter-kit 15
Figure 11 – Automatic joining of WiFi network with QR code 16
Figure 12 – QR code for fast connection to the WaziGate's dashboard 17
Figure 13 – Devices with stickers 19
Figure 14 – The box for the starter-kit 19
Figure 15 – The content of the starter-kit 20
Figure 16 – INTEL-IRRIS Irrigation WaziApp architecture 22
Figure 17 – Integration of embedded irrigation app into dashboard 22
Figure 18 – INTEL-IRRIS Irrigation WaziApp user interface 23

1. Introduction

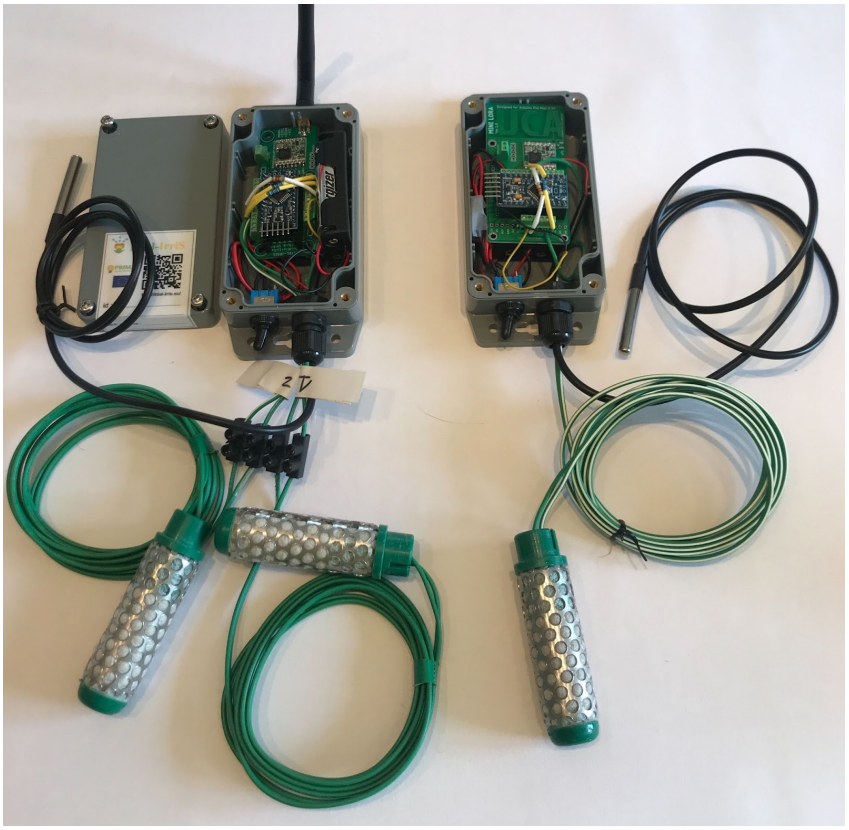
The "intelligent irrigation in-the-box" concept will be demonstrated by a starter-kit that consists of 1 **soil moisture sensor device** (either capacitive or tensiometer) developed by UPPA and **1 versatile IoT gateway** based on the WaziGate distribution developed by WAZIUP and further customized for INTEL-IRRIS by UPPA. On the gateway, a dedicated application to enhanced irrigation notification and parameters is part of the starter-kit

Refer to D2.2a “Starter-kit for smart irrigation system – v1” for a more detailed description of the starter-kit. This deliverable D2.2b “Starter-kit for smart irrigation system – v2” only addresses the new development to the starter-kit.

2. Sensor device part

2.1. 2-Watermark version

The 2-watermark device has been integrated in the starter-kit as one variant that could be deployed at some farms. It is only a summary as the details are presented in D1.2c Low-cost sensor generic platforms for connected irrigation system – v3.



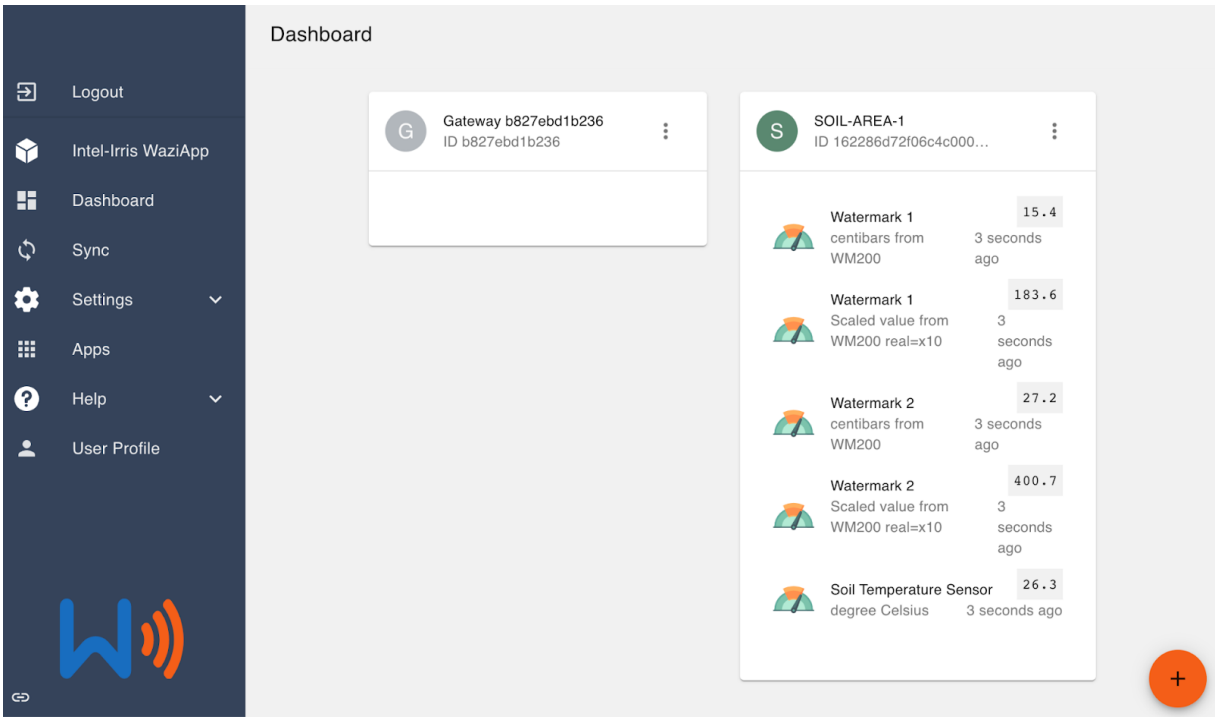
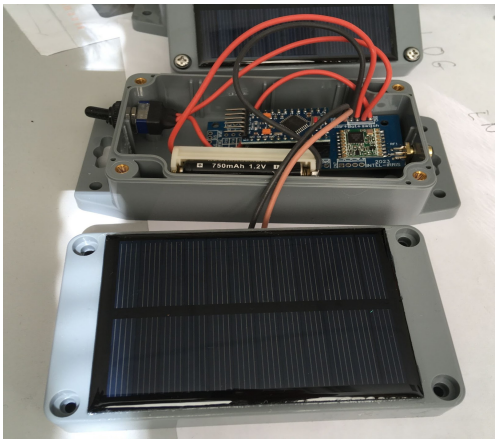


Figure 1 – 2-Watermark device (top) and view in WaziGate dashboard (bottom)

2.2. New DIY PCB with solar charging support

It is only a summary as the details will be presented in D1.2c Low-cost sensor generic platforms for connected irrigation system – v3.



2.3. WaziSense v2

The WaziSense v2 is the successor to the WaziSense v1. This new version is designed to facilitate the development of Minimum Viable Products, proof of concept prototypes and use for field deployment purposes. It is focused on use cases or conditions where resources like **power and internet access is limited.** With respect to the WaziSense v1, the new v2 board will have the following features and capabilities:

- approximately 10% smaller in size
- Solar maximum power point tracking and charging for Li-ion/Li-polymer batteries

- Programmable power rails for sensors and peripherals
- Ultra low power consumption in idle/sleep mode

It is only a summary as the details will be presented in D1.2c Low-cost sensor generic platforms for connected irrigation system – v3.

3. GATEWAY PART

3.1. Generic WaziGate framework

It is only a summary as the details are described in D2.1b "Final report on specifications & functionalities of the edge-enabled sensor-gateway framework for smart irrigation system"

3.1.1. New features

With respect to the previous version of the WaziGate (described in deliverable D2.1a), a new major revision of the WaziGate has been published, called V2.3.2. This version brings several important features:

1. Integrated SSH terminal to UI for remote maintenance
2. Introduced function to export gateways data in different manners
3. Set clock and timezone automatically
4. Compression of resources to support slow networks

3.1.2. Preparing AI framework

For development purposes, we have implemented AI models directly on the gateway to facilitate seamless experimentation. To simplify this process, we have integrated an application that runs JupyterLab in a docker container, equipped with essential machine learning packages pre-installed. Additionally, we provide convenient scripts for effortless installation of additional packages. These scripts serve as valuable examples, demonstrating how to utilize various functionalities such as leveraging linear regression to solve problems or dynamically accessing sensor and actuator values through the gateway's API.

3.2. INTEL-IRRIS specific WaziGate distribution

3.2.1. Real-Time Clock support

The support of a Real-Time Clock (RTC) module has been added in order to deploy the INTEL-IRRIS WaziGate in NO INTERNET mode. The RTC module is configured prior to deployment and will keep the correct time and date even after reboot. The RTC module can be directly connected to the radio LoRa hat as illustrated below

Figure 3 – INTEL-IRRIS WaziGate with RTC module

3.2.2. Auto-configuration features

The INTEL-IRRIS WaziGate provides a simple auto-configuration mechanism to automatically update and/or configure the gateway on boot for specific deployment settings using the base INTEL-IRRIS WaziGate SD card image: set frequency band, create pre-configured devices with pre-configured sensors,...

After flashing the INTEL-IRRIS WaziGate SD card image, one can insert the SD card in any computer (Windows, Linux, MacOS) to copy some configuration files in the /boot partition of the SD card. The /boot partition is in FAT32 format and therefore can easily be accessed (including Copy/Paste operation) from most operating systems without any additional software driver. It will usually appear as an additional drive named boot on the host operating system. There are basically 3 configuration files that can be put in this /boot partition:

- gateway.zip: a .zip archive with the latest Gateway content of the INTEL-IRRIS GitHub
- intel-irris-band.txt: simply contains either eu868 or eu433 or au915
- intel-irris-auto-config.sh: a script that mainly configures the WaziGate using its embedded REST API to create devices and sensors

This feature is documented in detail on the INTREL-IRRIS's GitHub:

https://github.com/CongducPham/PRIMA-Intel-IrriS/blob/main/Gateway/boot/README.md

The default configuration is to automatically configure the INTEL-IRRIS WaziGate with the starter-kit configuration.

- LoRaWAN mode (single channel)
- Cayenne LPP data format
- EU433 band (for Algeria and Morocco)
- 2 pre-configured devices with address 26011DAA and 26011DB1
- 26011DAA is a soil humidity device with the capacitive SEN0308 sensor
    - Device name is SOIL-AREA-1
    - temperatureSensor_0 as the internal default logical sensor on the WaziGate for soil humidity data. Display will show Soil Humidity Sensor/Raw value from SEN0308
    - temperatureSensor_5 as the internal default logical sensor on the WaziGate for the soil temperature data if a DS18B20 is connected. Display will show Soil Temperature Sensor/degree Celcius
    - analogInput_6 as the internal default logical sensor for battery voltage. Display will show Battery voltage/volt, low battery whebn lower than 2.85V
- 26011DB1 is a soil humidity device with the Watermark WM200 tensiometer sensor
    - Device name is SOIL-AREA-2
    - temperatureSensor_0 as the internal default logical sensor on the WaziGate for soil humidity data. It provides the converted resistance value in centibar, Taking into account the soil temperature data. Display will show Soil Humidity Sensor/centibars from WM200
    - temperatureSensor_1 as the internal default logical sensor on the WaziGate for soil humidity data. It provides the raw resistance value measured from the Watermark sensor. The value is scaled down by 10, so to get the real

resistance value one must multiply by 10. Display will show Soil Humidity Sensor/scaled value from WM200 real=x10

- temperatureSensor_5 as the internal default logical sensor on the WaziGate for the soil temperature data if a DS18B20 is connected. Display will show Soil Temperature Sensor/degree Celcius
- analogInput_6 as the internal default logical sensor for battery voltage. Display will show Battery voltage/volt, low battery when lower than 2.85V

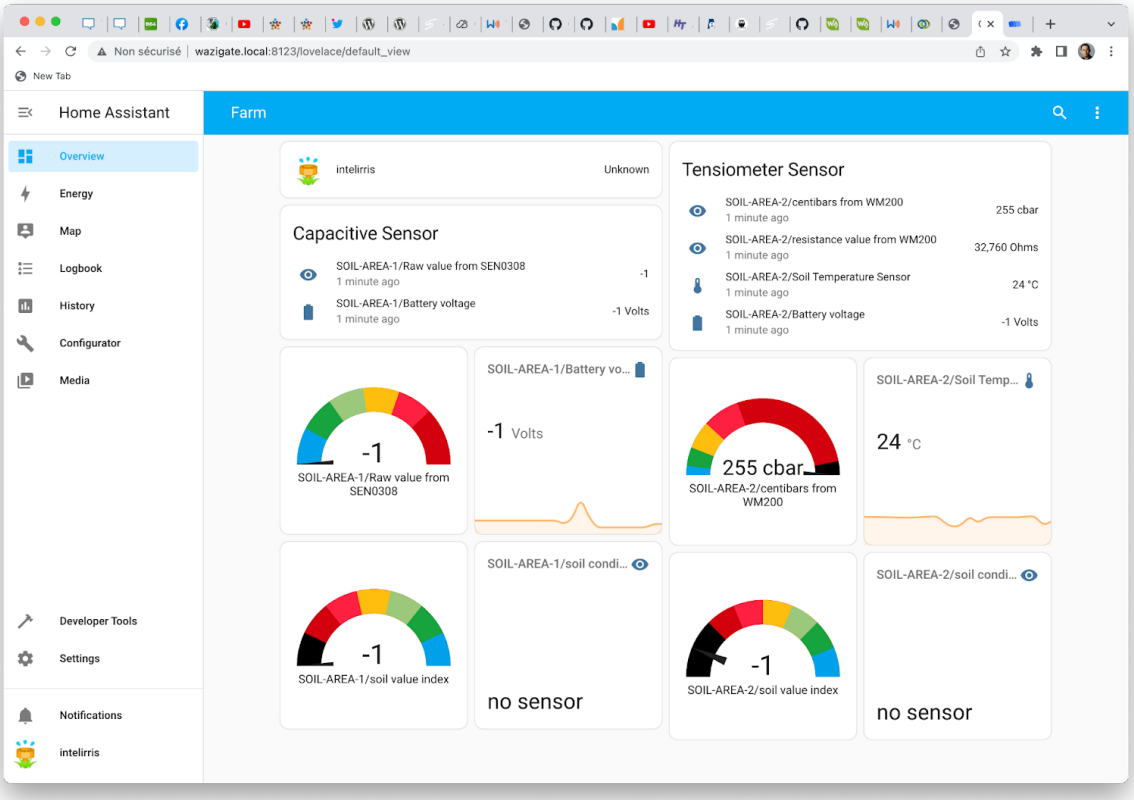
3.2.3. Integration of Home Assistant

The versatile Home Assistant (HA) framework has been integrated into the INTEL-IRRIS WaziGate and the INTEL-IRRIS starter-kit sensors are automatically added to the HA dashboard.

HA allows INTEL-IRRIS's users to integrate a much wider variety of sensors and also benefit from the HA mobile application to provide a unified view of all the sensors.

This feature is documented in detail on the INTEL-IRRIS's GitHub:

https://github.com/CongducPham/PRIMA-Intel-IrriS/blob/main/Gateway/homeassistant/README.md



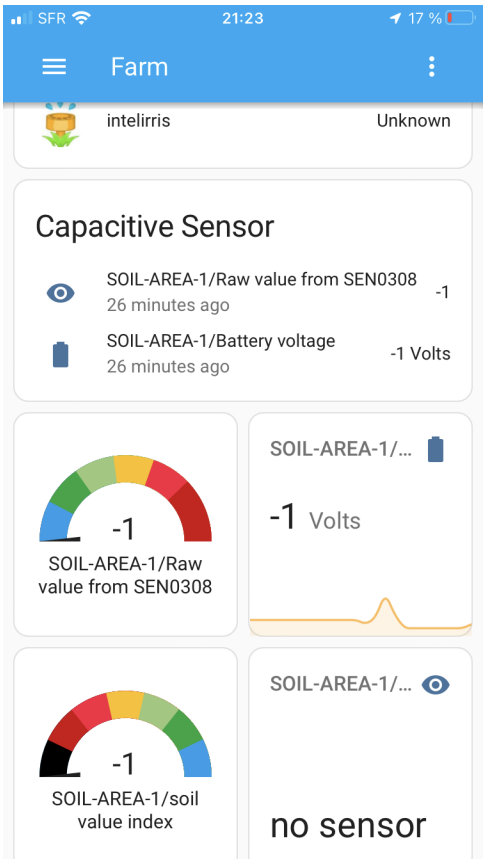


Figure 4 – HA dashboard (top) and HA mobile app (bottom)

3.2.4. Backup/Recovery procedures

A simple Backup/Recovery feature has been added in order to deploy the INTEL-IRRIS WaziGate in NO INTERNET mode and when frequent reboot can happen. 2 command scripts have been developed:

- backup_everything.sh
- restore_everything.sh

By default, all the sensor's data is backed up to a dedicated folder on the INTEL-IRRIS WaziGate every 6 hours (at 3am, 9am, 3pm and 9pm). If a USB dongle is connected to the INTEL-IRRIS WaziGate, then these sensor's data are also copied to the USB dongle.

A user can also simply manually run a backup through the new "Integrated SSH terminal to UI for remote maintenance" (see section 2.2.1).

A user can also take the sensor's data that has been backed up on a USB dongle to install a new INTEL-IRRIS WaziGate with all the previous sensor's data restored.

This feature is documented in detail on the INTEL-IRRIS's GitHub:

https://github.com/CongducPham/PRIMA-Intel-IrriS/blob/main/Gateway/sensor-backup/README.md

Note that the specific IIWA configuration of the sensors are also backed up. See section 3.1.2 for more information on the IIWA sensor configuration.

3.2.5. Advanced scripts for dataset support

In the previous version, the simplest (and maybe unique) procedure that existed to collect data was to use the Wazigate Dashboard and store the information in a sensor by sensor process. The extraction of data from various devices from various Wazigates has been improved by scripts solutions to be run both internally on the WaziGates or remotely.

We worked on Python/Bash global extraction scripts, in parallel with the development of data extraction features in WaziGate UI (see Sec.2.1.2 and D2.1b/Sec.2.2), and according to the requirements discussed in Bondy's technical meeting.

The users can now log into their Wazigates via a command terminal (SSH), and gather the information stored there, both from the Wazigate system and from WaziApp (see 3), by requesting data from the APIs. The extracted information can then be copied to the users' computers for further analysis, using e.g. SCP or SFTP.

Since only the rest APIs are involved, the same extraction features can also be applied remotely from the users' terminals, using Bash (curl) or Python. The same credentials are required for WaziGate system and IIWA REST scripts used remotely. The provided scripts have been validated for scenarios involving several gateways.

The dataset identifies devices, sensors using Wazigate, INTEL-IRRIS, and user-defined information. Every piece of data is timestamped in an universal way. Several data extraction formats are available from UPPA's scripts:

- JSON: very wide-spread format, its advantage is its simple interaction with well-spread tools for further analysis such as Python. Data is stored as a list of sensors for each device comprising their identifiers and the list of time-value pairs that have been collected;


```
                    {
                        "value": 3.4,
                        "time": "2023-05-03T12:14:44+02:00"
                    },
                    {
                        "value": 2.46,
                        "time": "2023-05-03T12:17:53+02:00"
                    }
                ]
            }
        ]
    },
    "1": {
        "devName": "SOIL-AREA-2",
        "devAddr": "26011DAB",
        "devID": "6450e3fa68f3190a03d08f40",
        "sensors": [
            {
                "id": "temperatureSensor_0",
                "name": "Soil Humidity Sensor",
                "pretty_name": "Soil Humidity Sensor / Raw value from SEN0308",
                "values": [
                    {
                        "value": 800,
                        "time": "2023-05-02T08:00:00+02:00"
                    },
                    {
                        "value": -1,
                        "time": "2023-05-02T08:00:00+02:00"
                    },
                    {
                        "value": 97,
                        "time": "2023-05-02T12:24:25+02:00"
                    },
                    {
```

- CSV as discussed in Bondy: CSV is also wide-spread and researcher friendly. As discussed in Bondy, this extraction first gathers and sorts all the collected timestamps in order to display the values with a first shared column of time in the CSV table;

	A	B	C	D	E
1		SOIL-AREA-1			SOIL-AREA-2
2		6450e3fa68f3190a03d08f3c			6450e3fa68f3190a03d08f40
3		Soil Humidity Sensor / Raw value from SEN0308	Soil Temperature Sensor / degree Celsius	Battery voltage / volt	Soil Humidity Sensor / Raw value from SEN0308
4	2023-05-02T12:24:22+02:00	145	n/a	n/a	n/a
5	2023-05-02T12:24:23+02:00	n/a	n/a	3.21	n/a
6	2023-05-02T12:24:25+02:00	n/a	n/a	n/a	97

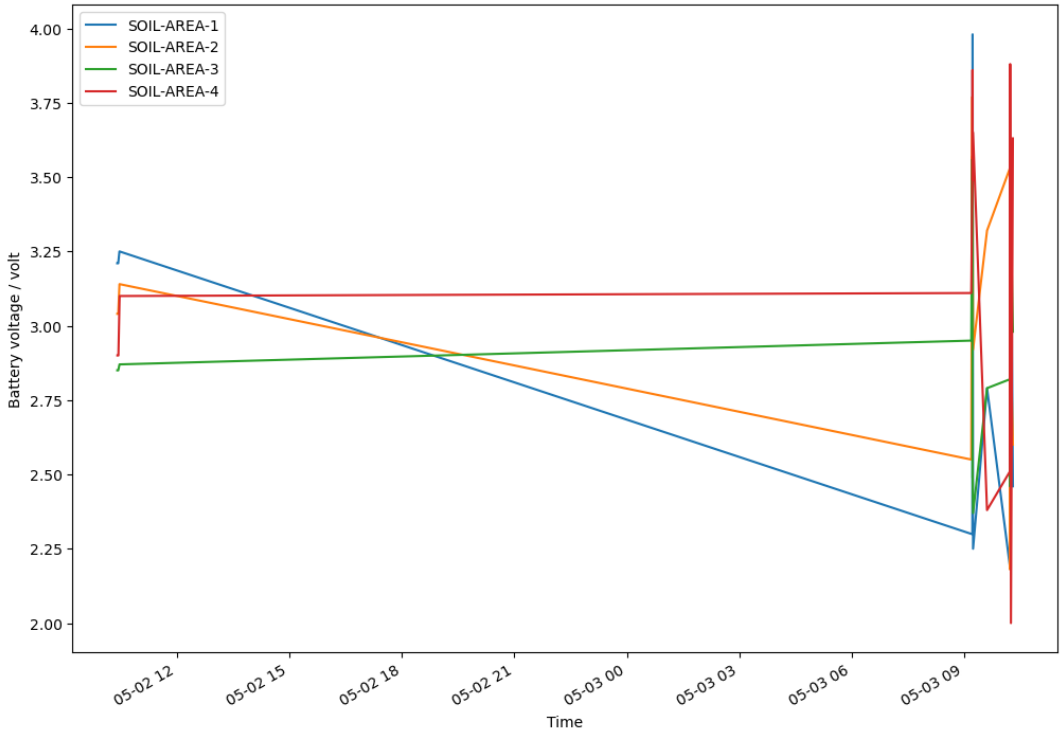
- CSV as proposed from the WaziGate UI: the more features the better, we converged with WAZIUP e.V. on a “all in one” CSV format, with a multiple line header and a sensor-based structure with independent (time-value) pairs of columns;

	S	T	U	V	W	X
	6450e3fa68f3190a03d08f45	analogInput_6	6450e3fa68f3190a03d08f4c	temperatureSensor_0	6450e3fa68f3190a03d08f4c	temperatureSensor_1
s	SOIL-AREA-3	Battery voltage / volt	SOIL-AREA-4	Watermark 1 / centibars from WM200	SOIL-AREA-4	Watermark 1 / scaled value from WM200 real=x10
9	2023-05-02T08:00:00+02:00	-1	2023-05-02T08:00:00+02:00	10	2023-05-02T08:00:00+02:00	550
4	2023-05-02T12:24:29+02:00	2.85	2023-05-02T08:00:00+02:00	255	2023-05-02T08:00:00+02:00	3276
4	2023-05-02T12:26:03+02:00	2.85	2023-05-02T12:24:40+02:00	172	2023-05-02T12:24:39+02:00	1720
8	2023-05-02T12:28:00+02:00	2.87	2023-05-02T12:26:14+02:00	172	2023-05-02T12:26:13+02:00	1720

The extraction scripts permit the users to select a subset of devices/sensors for extraction, using a template with each identifying information.

Note that INTEL-IRRIS auto configuration procedure sets default values for all its sensors. In Bondy’s CSV format, these initial values are filtered out; both the other portions of scripts (for JSON and “all in one”) are generic (no filtering).

Finally, we developed Python scripts to display as a plot the data, analyzed from the different formats as sources, and with the default INTEL-IRRIS values filtered out. These scripts exemplify the use of Pandas and Matplotlib libraries and validate the possibility to further exploit the datasets with AI/machine learning tools.

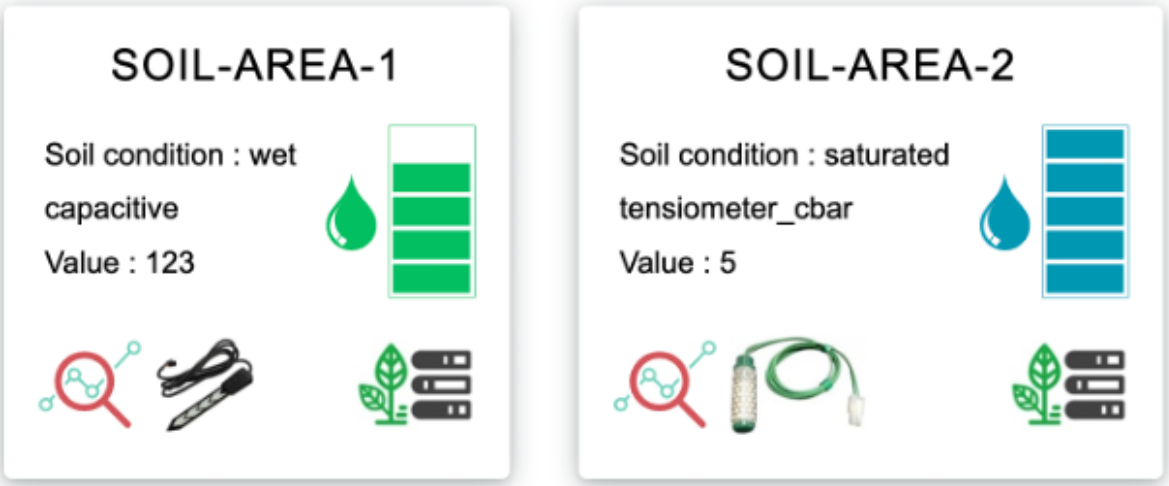


4. IIWA

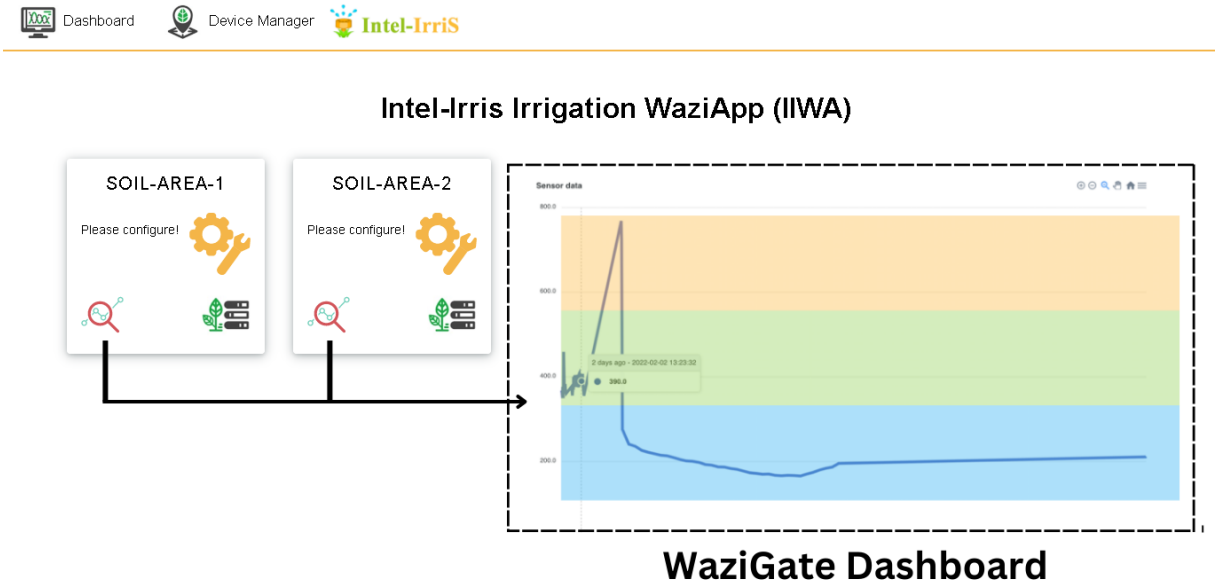
4.1. New User Interface

4.1.1. Dashboard

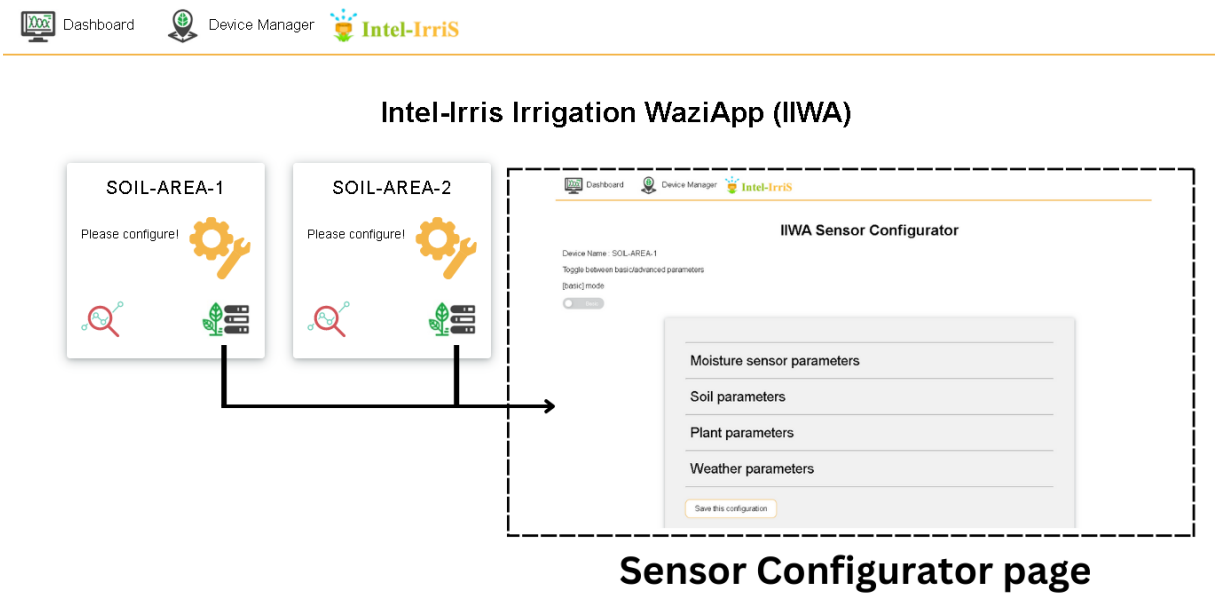
IIWA has a new Dashboard that shows all WaziGate LoRaWAN devices that have been added to the application. The devices are displayed on cards which show valuable information to a user. These information include: the name assigned to the LoRaWAN device, soil moisture condition, a graphical representation of the soil moisture condition, soil moisture index value, and an image of the soil moisture sensor type that a device has.



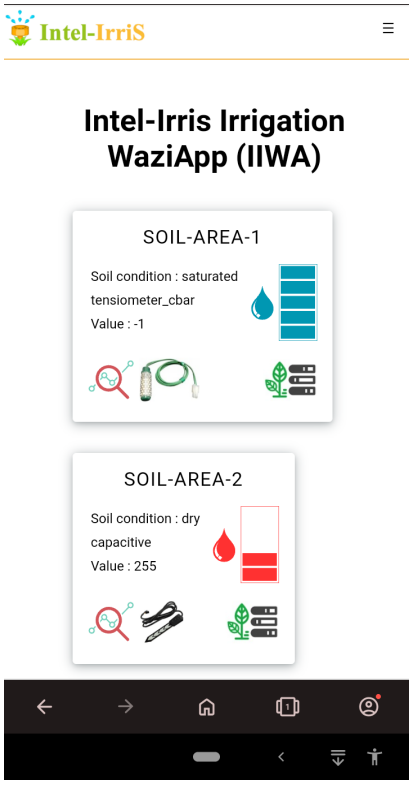
Also included on the card list are two clickable graphical icons. One icon enables users to visualize the historical sensor values of each device by opening the sensor graph page on the WaziGate Dashboard.

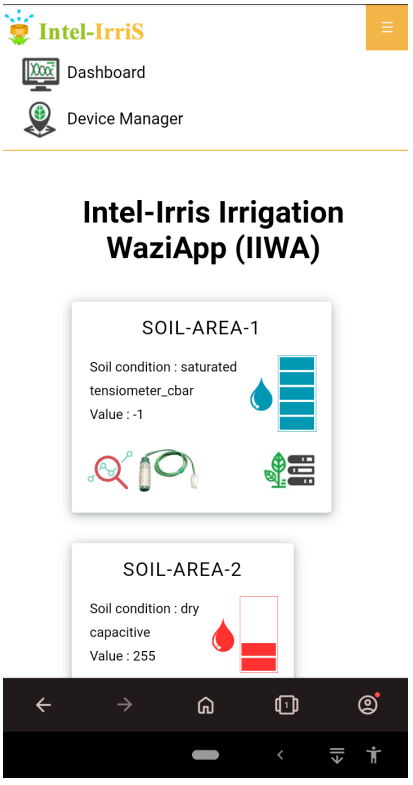


The other icon enables a user to configure a device's sensor parameters by using the Sensor Configurator page. Once a user clicks the icon, the Sensor Configurator page is opened and the configurations can be made on the UI. If a device's sensor has not been configured, the soil index value is not computed and the Dashboard will prompt the user to configure the device's sensor.



The new Dashboard is also responsive on various device screens such as computers and smartphones.





4.1.2. Sensor configuration

The IIWA sensor configurations are categorized based on the sensor, soil, plant and the weather. For each category, there are multiple parameters that the user can provide.

1. Moisture sensor parameters : the sensor type is a mandatory parameter that a user needs to provide so that the soil index value can be computed. INTEL-IRRIS soil humidity sensors can either be: capacitive, tensiometer (cbar) or tensiometer (raw sensor value). Other parameters that a user can define are: sensor age in months, maximum sensor value and the minimum sensor value. The maximum and minimum sensor values are automatically defined by the application once the user selects the sensor type.
2. Soil parameters: the soil irrigation type is a mandatory parameter that a user needs to provide so that the soil index value can be computed. Other parameters that a user can define are:
    (a) soil type: clay, silty, loamy, sandy
    (b) soil temperature source: give a temperature value or provide details of a Waziup LoRaWAN temperature sensor.
    (c) soil salinity
    (d) soil bulk density
    (e) soil field capacity.

    The available choices for soil irrigation type are: submersion, furrow, sprinkler, drip, subirrigation.

3. Plant parameters: the available plant parameters are:
    (a) Plant type: tomatoes, potatoes, watermelon, artichoke, maize, wheat, citrus, apple
    (b) Planting date
    (c) Plant category: vegetable, cereal, fruit tree
    (d) Plant variety
4. Weather parameters:
    (a) Region: arid, semi-arid, dry
    (b) Weekly evaporation (in millimeters)
    (c) Weekly pluviometry (in millimeters)

All the sensor configuration data is stored in a JSON file.

```
{
    "globals": {
        "soil_salinity": "undefined",
        "soil_bulk_density": "undefined",
        "soil_field_capacity": "undefined",
        "weather_region": "undefined",
        "weather_weekly_evaporation": "undefined",
        "weather_weekly_pluviometry": "undefined"
    },
    "sensors": [
        {
            "value": {
                "sensor_type": "capacitive",
                "sensor_age": "0",
                "sensor_max_value": "800",
                "sensor_min_value": 0,
                "soil_type": "loamy",
                "soil_irrigation_type": "sprinkler",
                "soil_salinity": "undefined",
                "soil_bulk_density": "undefined",
                "soil_field_capacity": "undefined",
                "plant_category": "undefined",
                "plant_type": "undefined",
                "plant_variety": "undefined",
                "plant_planting_date": "undefined",
                "weather_region": "undefined",
                "weather_weekly_evaporation": "undefined",
                "weather_weekly_pluviometry": "undefined",
                "last_sensor_value": -1
            },
            "soil_temperature_source": {
                "soil_temperature_device_id": "undefined",
                "soil_temperature_sensor_id": "undefined",
                "soil_temperature_value": "undefined"
            },
            "device_id": "63d38d9d26c0473ae456730f",
            "sensor_id": "temperatureSensor_0"
        }
    ]
}
```

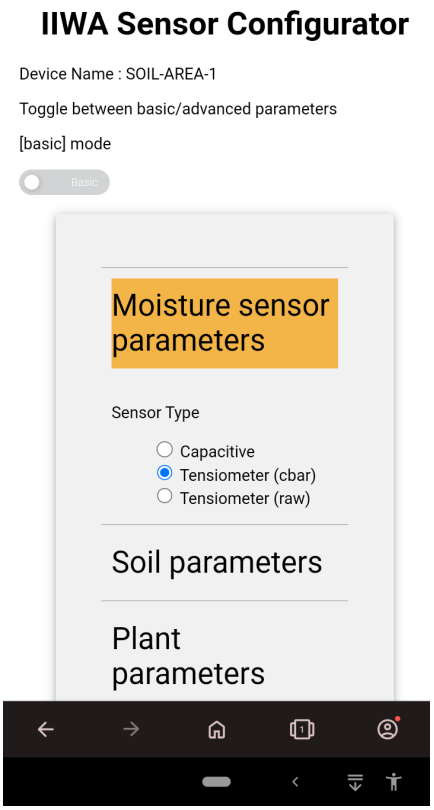

4.2. IIWA parameters menu

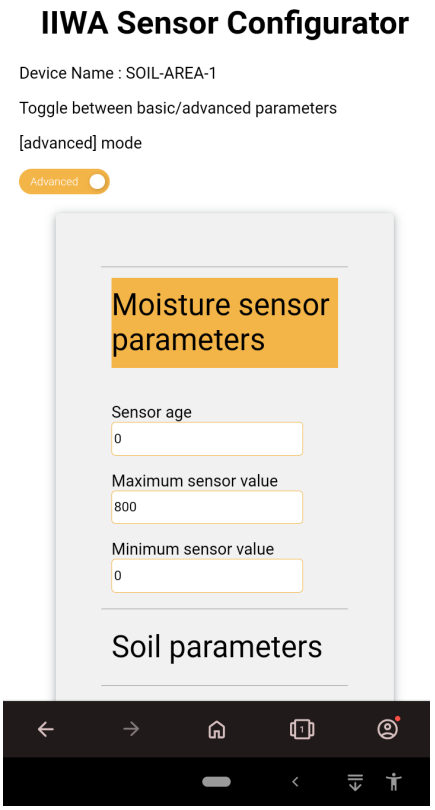
4.2.1. Basic vs Advanced view

The basic and advanced view mode enables a user to easily configure the parameters by filtering only the parameter “group” that they wish to see and configure. Each IIWA sensor configuration category (sensor, soil, plant and weather) has a subgroup of basic and advanced parameters.

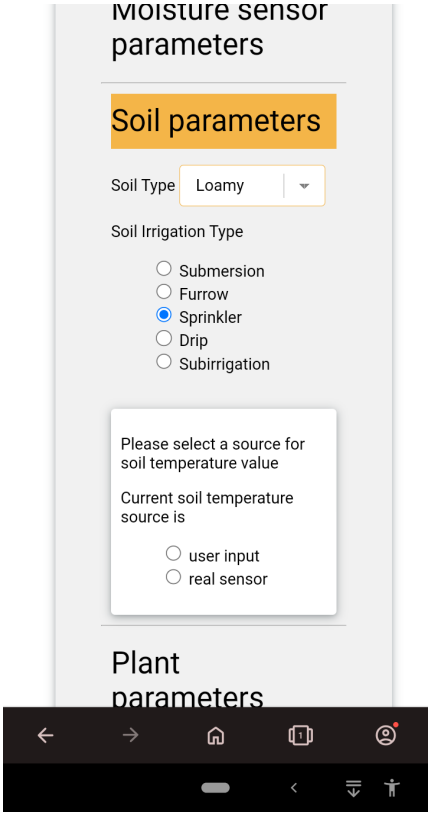
As the user defines various parameters on the basic and advanced views the application will save their selected parameters for each view mode.

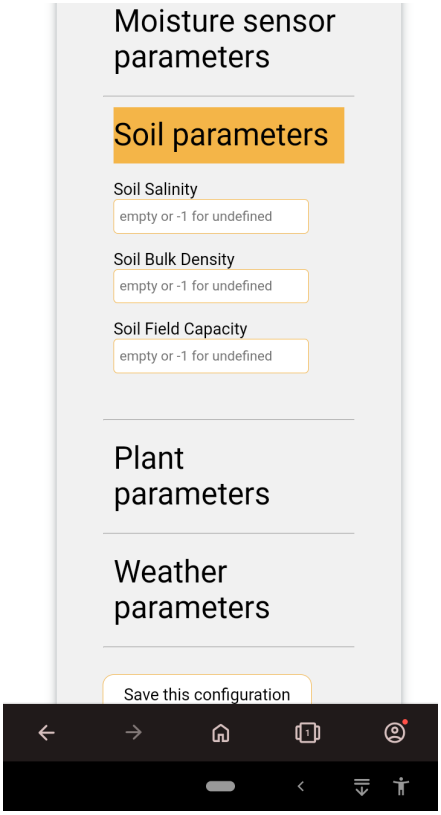
1. Moisture sensor parameters: basic view on left and advanced view on right



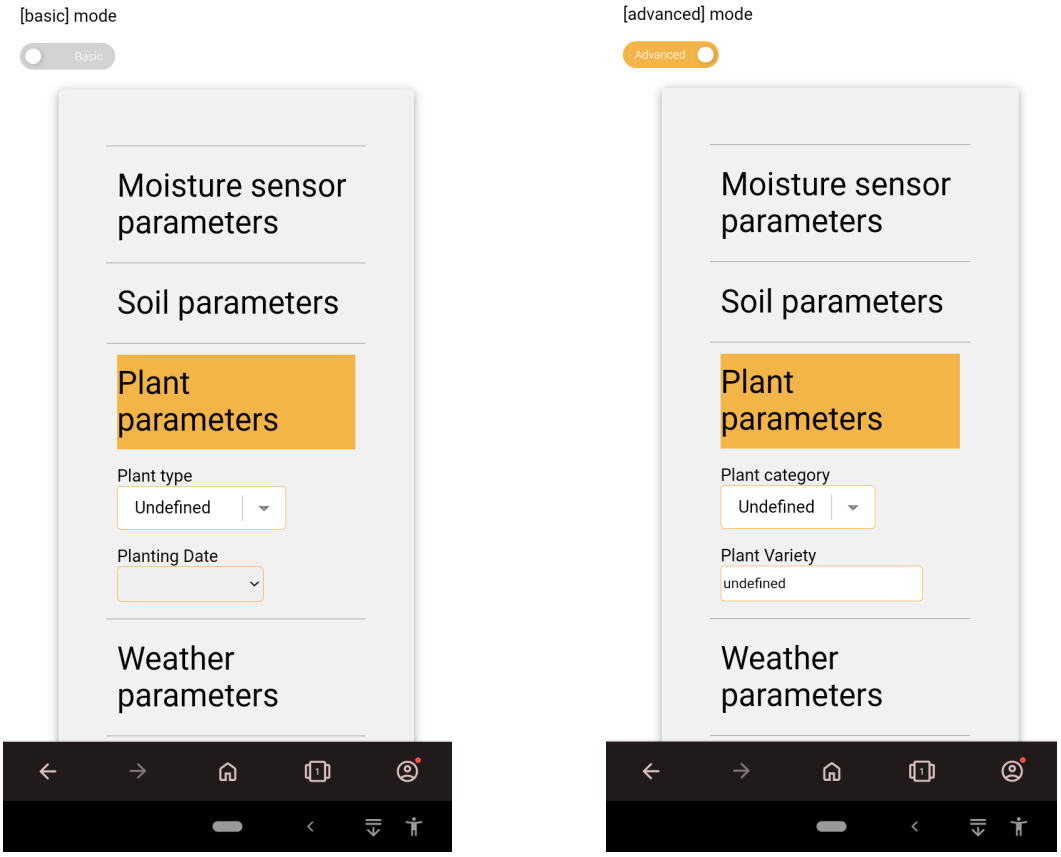


2. Soil parameters: basic view on left and advanced view on right

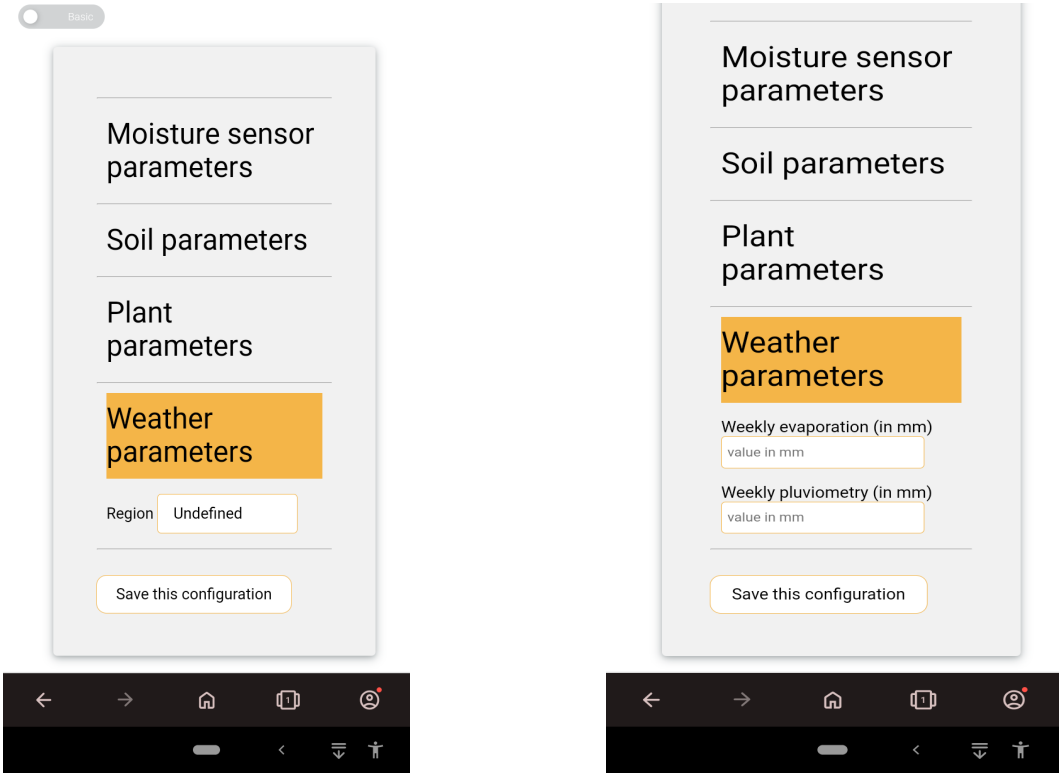




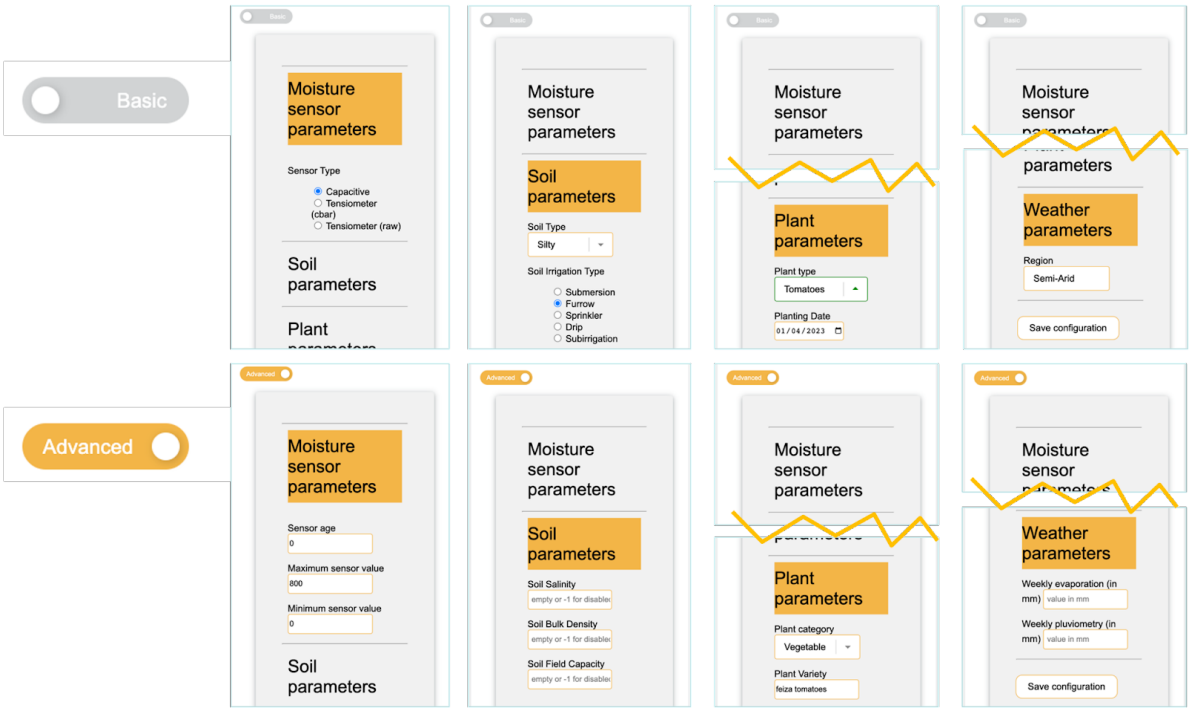
3. Plant parameters: basic view on left and advanced view on right



4. Weather parameters: basic view on left and advanced view on right



Summary of all parameters



4.3. IIWA REST API

In the INTEL-IRRIS WaziGate distribution (see 2.2), two scripts (Python & Bash) make interface with this REST API. From the command line, a user can add, delete a device, and collect its sensor's configurations and parameters. These APIs are also used when a user updates data on the UI.

1. POST request for adding a Waziup LoRaWAN device to IIWA:

   http://{WaziGate_IP_Address}:5000/devices/{LoRaWAN_deviceID}

2. DELETE request for removing a Waziup LoRaWAN device from IIWA:

   http://{WaziGate_IP_Address}:5000/devices/{LoRaWAN_deviceID}

3. GET request for obtaining list of Waziup LoRaWAN devices added to IIWA:

   http://{WaziGate_IP_Address}:5000/devices

4. POST request for adding a sensor configuration to IIWA configuration file:

   http://{WaziGate_IP_Address}:5000/devices/{LoRaWAN_deviceID}/sensors/{LoRaWAN_sensorID}

5. GET request for obtaining sensor configuration file data:

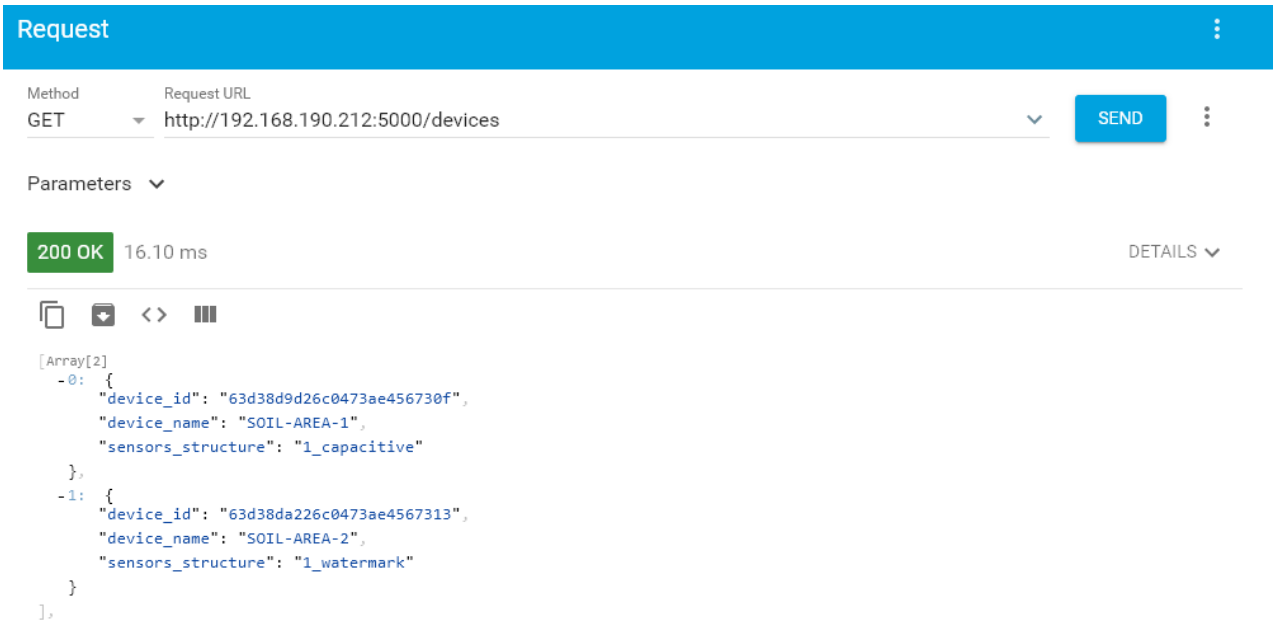
   http://{WaziGate_IP_Address}:5000/sensors_configurations

6. GET request for obtaining configurations for a particular sensor:

   http://{WaziGate_IP_Address}:5000/devices/{LoRaWAN_deviceID}/sensors/{LoRaWAN_sensorID}

Where: {WaziGate_IP_Address} is the IP address of the WaziGate running IIWA

Below is a snapshot of a GET request to obtain a list of Waziup LoRaWAN devices added to IIWA:



5. CONCLUSIONS

D2.2b presented the INTEL-IRRIS starter-kit v2. The main improvements consist in the 2-watermark version of the soil sensor device, the support for future AI processing and dataset features and the development of the new version of IIWA embedded application for out-of-the-box deployment.

This starter-kit version will been extensively tested by INTEL-IRRIS partners to ensure that all desirable features are in place.

REFERENCES

[1] D1.2a “Low-cost sensor generic platforms for connected irrigation system” http://intel-irris.eu/wp-content/uploads/2022/01/D1.2a.pdf

[2] D2.1a “First report on specifications & functionalities of the edge-enabled sensor-gateway framework for smart irrigation system” http://intel-irris.eu/wp-content/uploads/2022/03/D2.1a.pdf

[3] List of hardware parts. Technical Annex for D1.2a. https://github.com/CongducPham/PRIMA-Intel-IrriS/blob/main/Tutorials/Intel-IrriS-low-cost-sensor-hardware-parts.pdf

[4] WaziGate framework from WAZIUP https://www.waziup.io/documentation/wazigate/

[5] INTEL-IRRIS GitHub https://github.com/CongducPham/PRIMA-Intel-IrriS

[6] INTEL-IRRIS WaziGate SD card image on INTEL-IRRIS web site http://intel-irris.eu/results

Acronyms List

Acronym	Explanation
AI	Artificial Intelligence
API	Application Programing Interface
CSV	Comma-Separated Values
FTP	File Transfer Protocol
IIWA	INTEL-IRRIS Irrigation WaziApp
JSON	JavaScript Object Notation
HA	Home Assistant
PCB	Printed Cirvuit Board
REST API	REpresentational State Transfer API
RTC	Real Time Clock
SCP	Secure Copy
SFTP	Secure FTP
SSH	Secure Shell
UI	User Interface
USB	Universal Serial Bus

Project Co-ordinator Contact

Pr. Congduc Pham

University of Pau

Avenue de l'Université

64000 PAU

FRANCE

Email: Congduc.Pham@univ-pau.fr

ACKNOWLEDGEMENT

This document has been produced in the context of the PRIMA INTEL-IRRIS project. The INTEL-IRRIS project consortium would like to acknowledge that the research leading to these results has received funding from the European Union through the PRIMA program.