LIVE-GRAPHING with LORAWAN and PYTHON

Rositsa Maksimova

14th Annual Meeting of the Bulgarian Section of SIAM
December 17 - 19, 2019, Sofia, Bulgaria (BGSIAM'19)

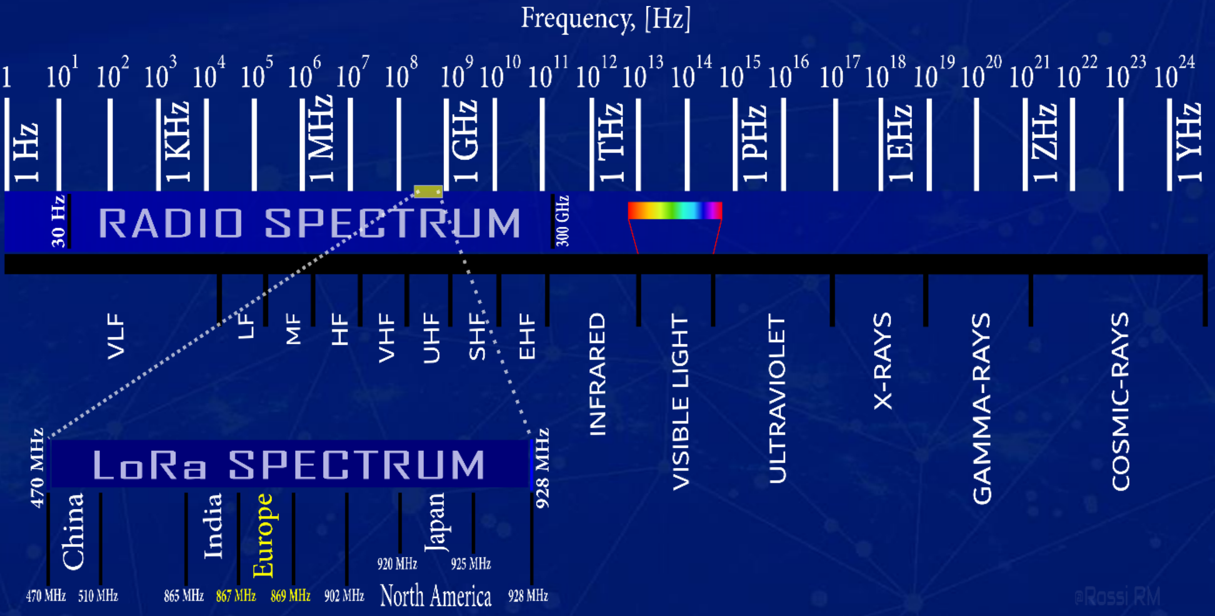
ABSTRACT

- The world of Internet of Things with LoRaWAN and Python
- The Things Network and a running gateway
- Dash Web Application for real-time monitoring of sensor data collected through LoRaWAN
- Deployment of a Dash Application

I. INTRODUCTION

- The Internet of Things
- Long Range Wide Area Network – LoRaWAN
- The Things Network - TTN
- Message Queuing Telemetry Transport – MQTT
- Python’s Dash library by Plotly

The range of LoRa in the Radio spectrum of the Electromagnetic Spectrum

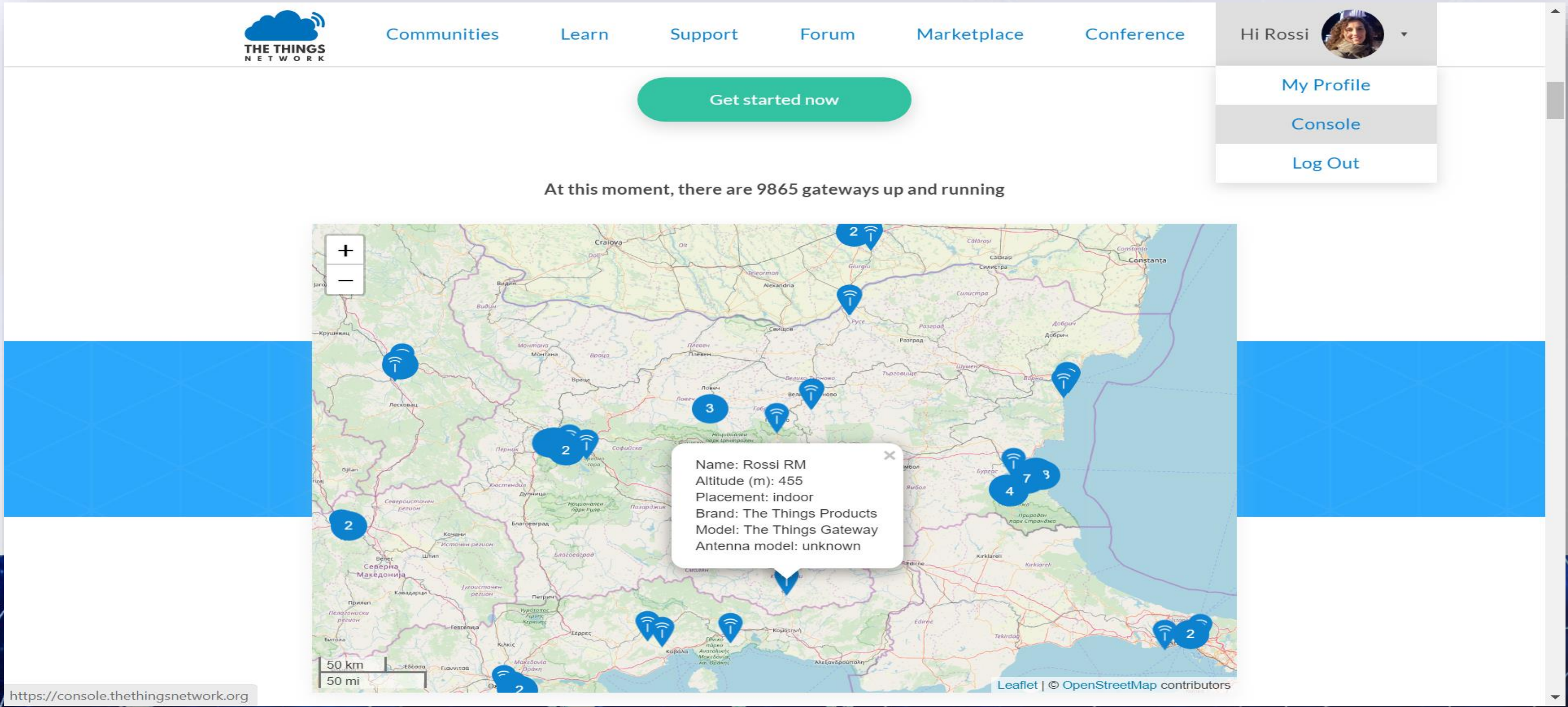


III. SOFTWARE OVERVIEW

3.1. APPLICATION in TTN

3.2. DASH WEB APPLICATION

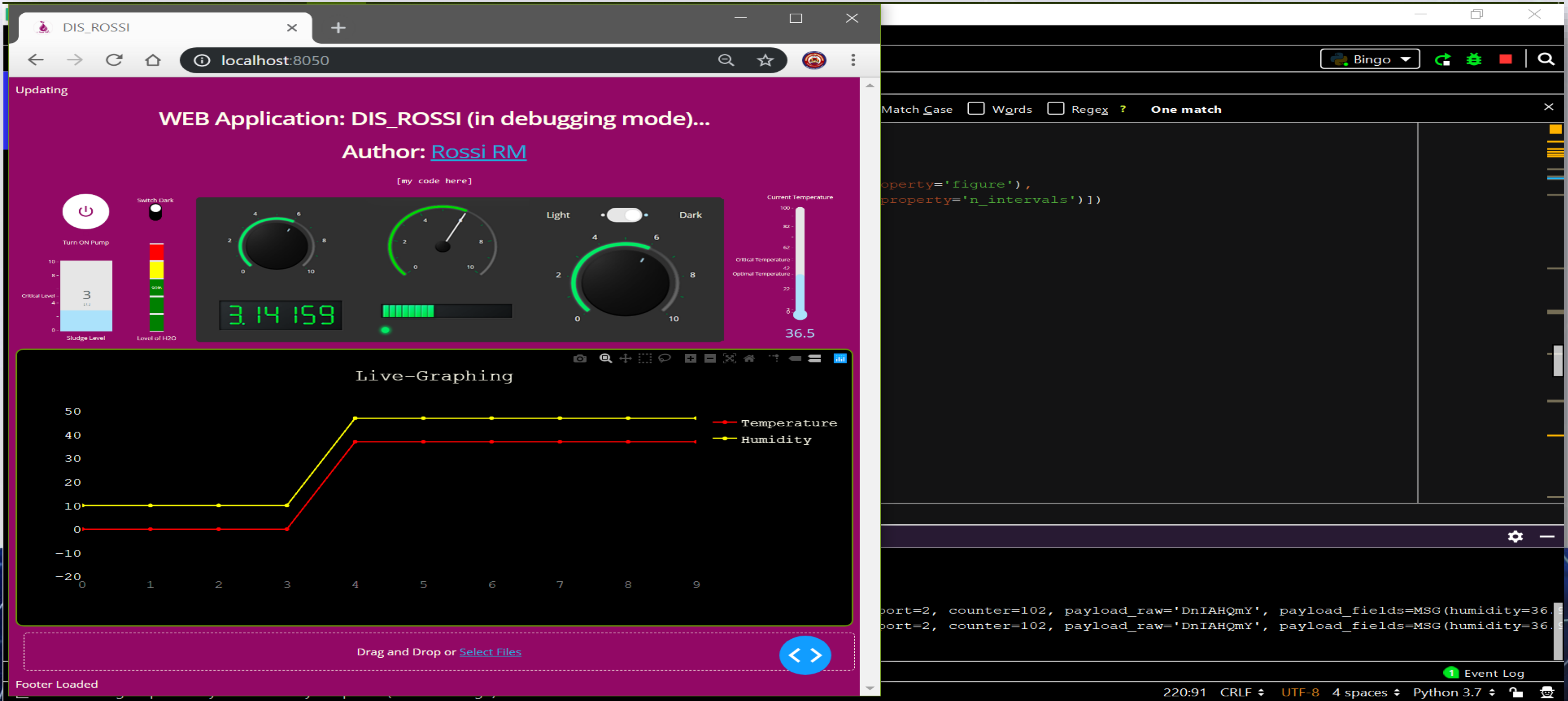
3.1. APPLICATION in TTN



At this moment, there are 9865 gateways up and running

Name: Rossi RM
Altitude (m): 455
Placement: indoor
Brand: The Things Products
Model: The Things Gateway
Antenna model: unknown

3.2. DASH WEB APPLICATION



Updating

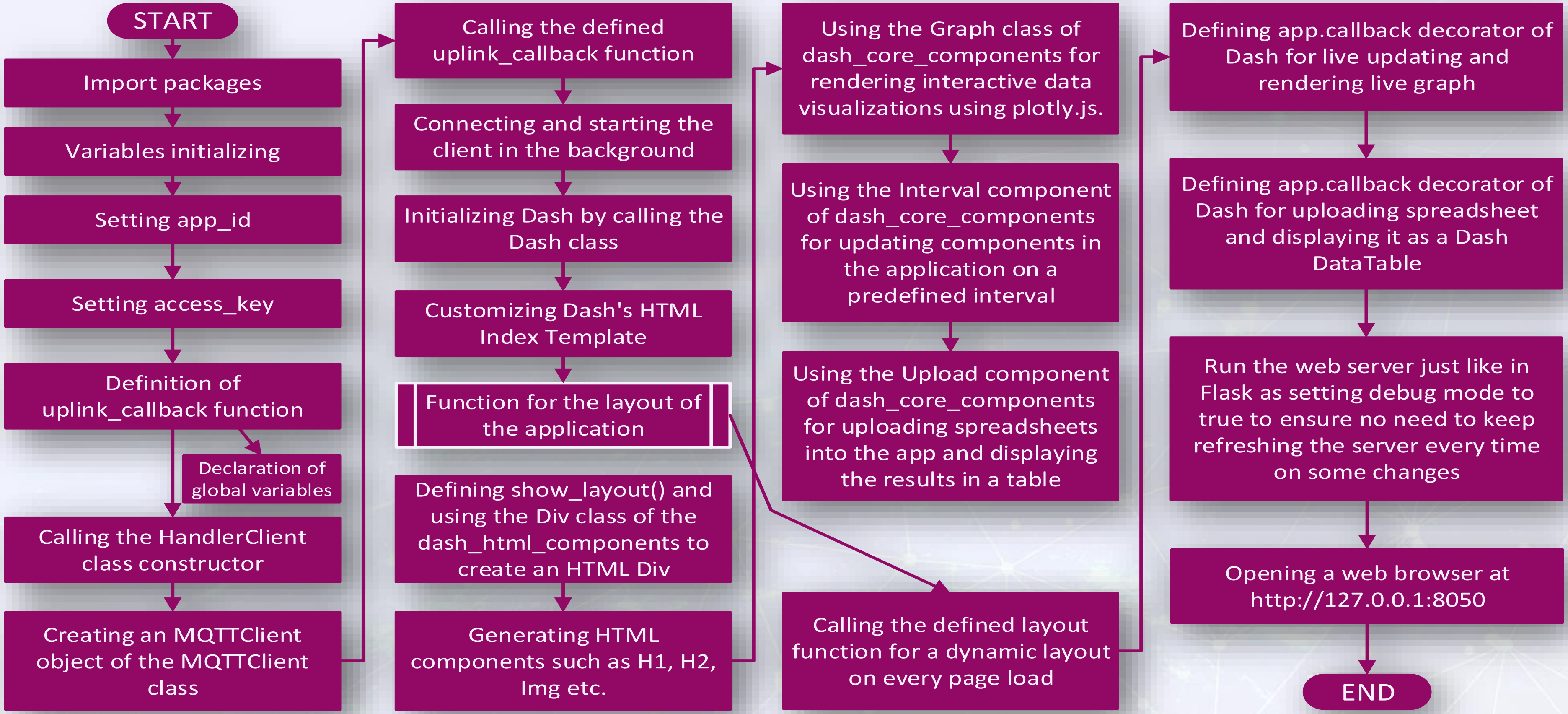
WEB Application: DIS_ROSSI (in debugging mode)...

Author: Rossi RM

[my code here]

```
operty='figure'),
property='n_intervals')])
```

```
port=2, counter=102, payload_raw='DnIAHQmY', payload_fields=MSG(humidity=36.
port=2, counter=102, payload_raw='DnIAHQmY', payload_fields=MSG(humidity=36.
```


START
Import packages
Variables initializing
Setting app_id
Setting access_key
Definition of uplink_callback function
Declaration of global variables
Calling the HandlerClient class constructor
Creating an MQTTClient object of the MQTTClient class
Calling the defined uplink_callback function
Connecting and starting the client in the background
Initializing Dash by calling the Dash class
Customizing Dash's HTML Index Template
Function for the layout of the application
Defining show_layout() and using the Div class of the dash_html_components to create an HTML Div
Generating HTML components such as H1, H2, Img etc.
Using the Graph class of dash_core_components for rendering interactive data visualizations using plotly.js.
Using the Interval component of dash_core_components for updating components in the application on a predefined interval
Using the Upload component of dash_core_components for uploading spreadsheets into the app and displaying the results in a table
Calling the defined layout function for a dynamic layout on every page load
Defining app.callback decorator of Dash for live updating and rendering live graph
Defining app.callback decorator of Dash for uploading spreadsheet and displaying it as a Dash DataTable
Run the web server just like in Flask as setting debug mode to true to ensure no need to keep refreshing the server every time on some changes
Opening a web browser at http://127.0.0.1:8050
END

CONCLUSION

- All ideas in the present paper can be used to conduct personal and scientific experiments in various scientific fields
- After a deployment the ability for remote access from each Internet connected node of the world would provide the freedom in sense of IoT
- The advantage of such a custom dashboard over ready dashboards is the ability and flexibility of its source code.
- This paper tends to be the basis of an intended development of a dissertation of its first author with a potential scientific application

LIVE-GRAPHING with LORAWAN and PYTHON

THANK YOU!