UNIVERSITÉ BISHOP'S UNIVERSITY

Bishop's University

CS 501 – Internet of Things

Final project: Study of an IoT simulator

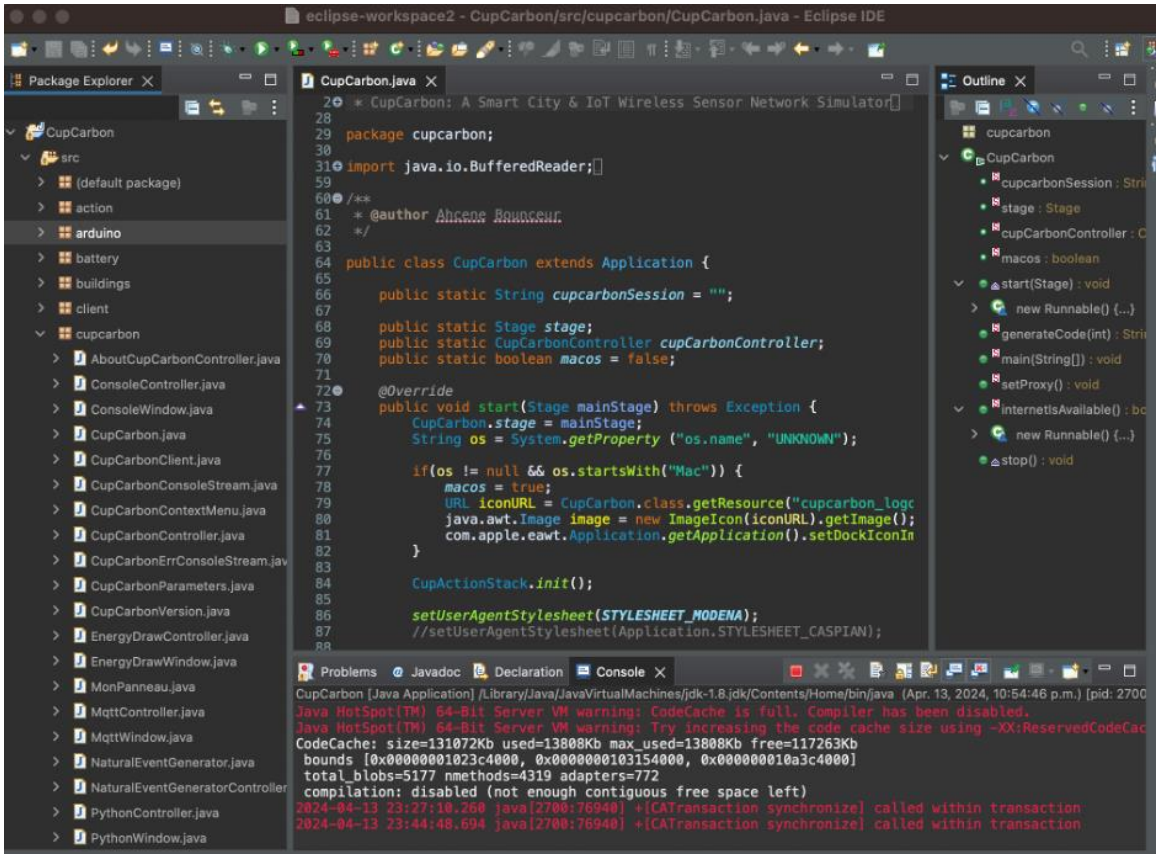
Group : FPP

Professor: **Dr. Mohammed Ayoub Alaoui Mhamdi**

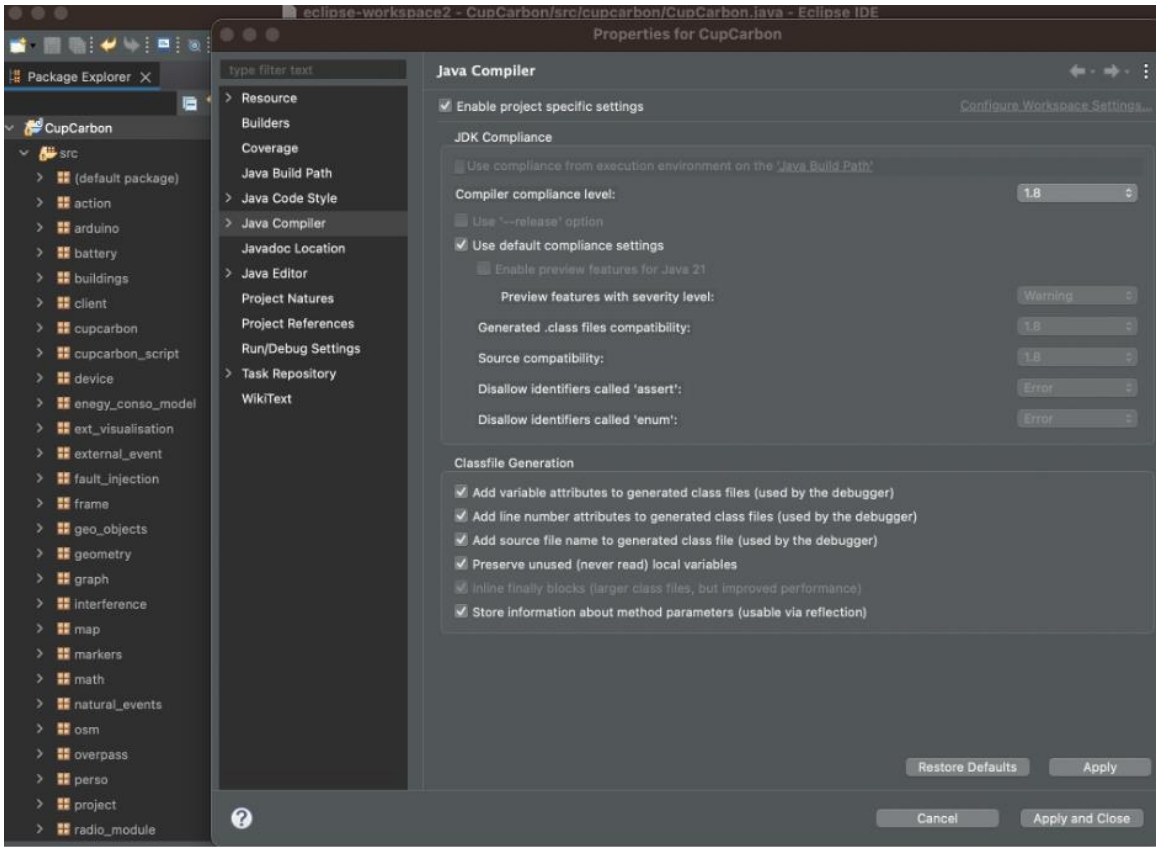
Student name	Student ID
Cicéron Ovide Aissoun	002239748
Ahmad Issa	002230777

Final project: Study of an IoT simulator

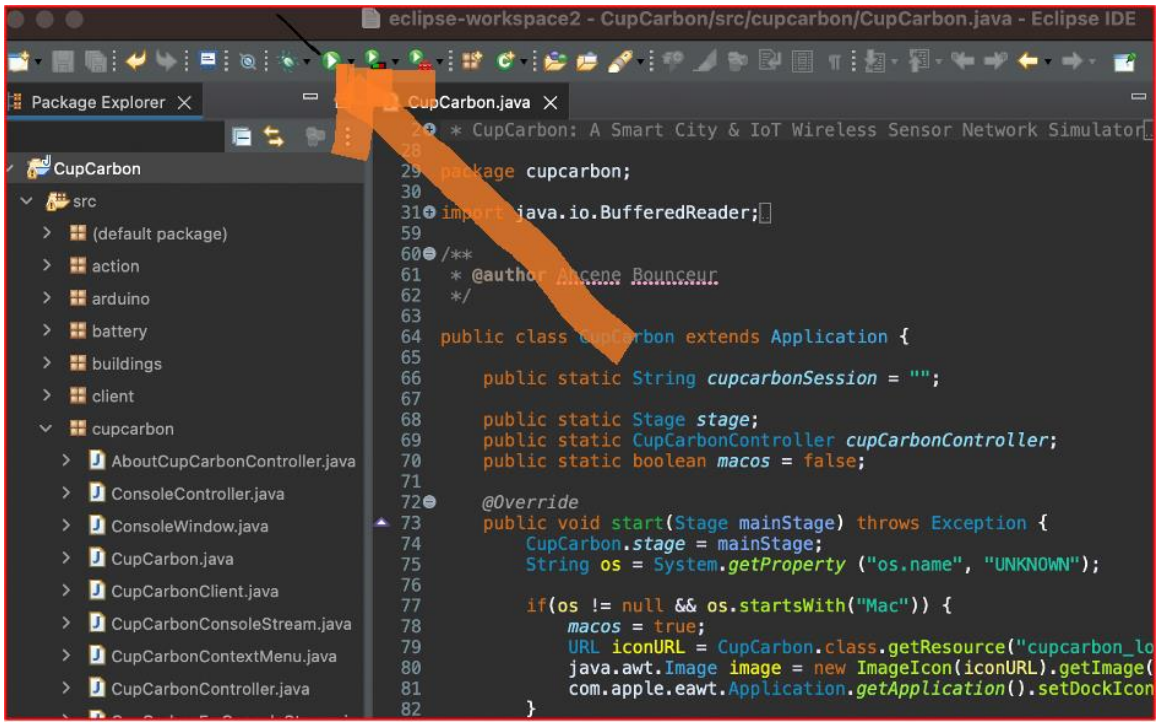
Main Eclipse IDE interface



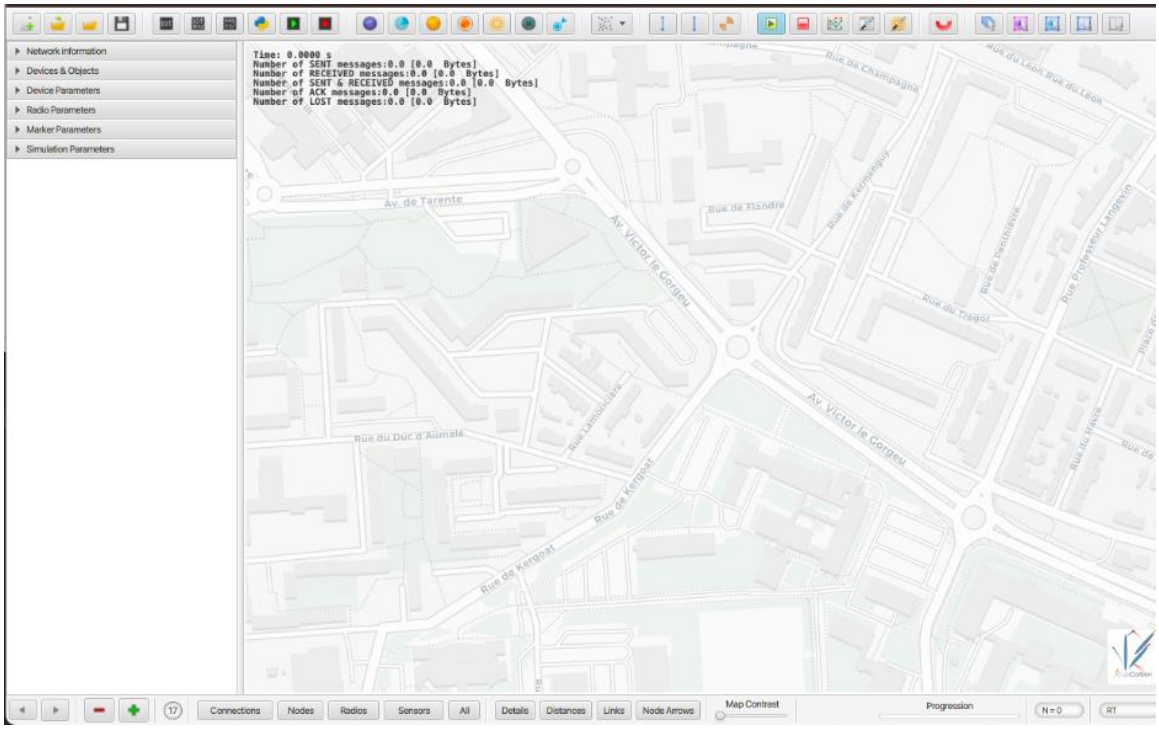
CupCarbon Configuration: selection of Java 8



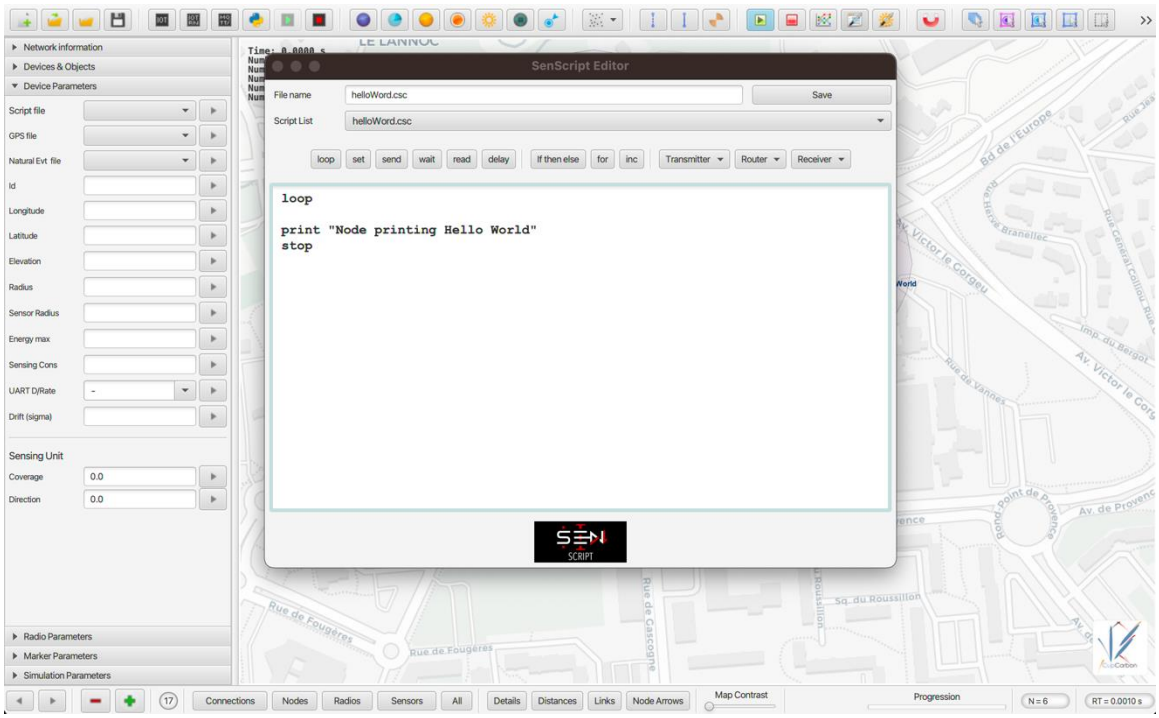
Running CupCarbon file to reach the home page



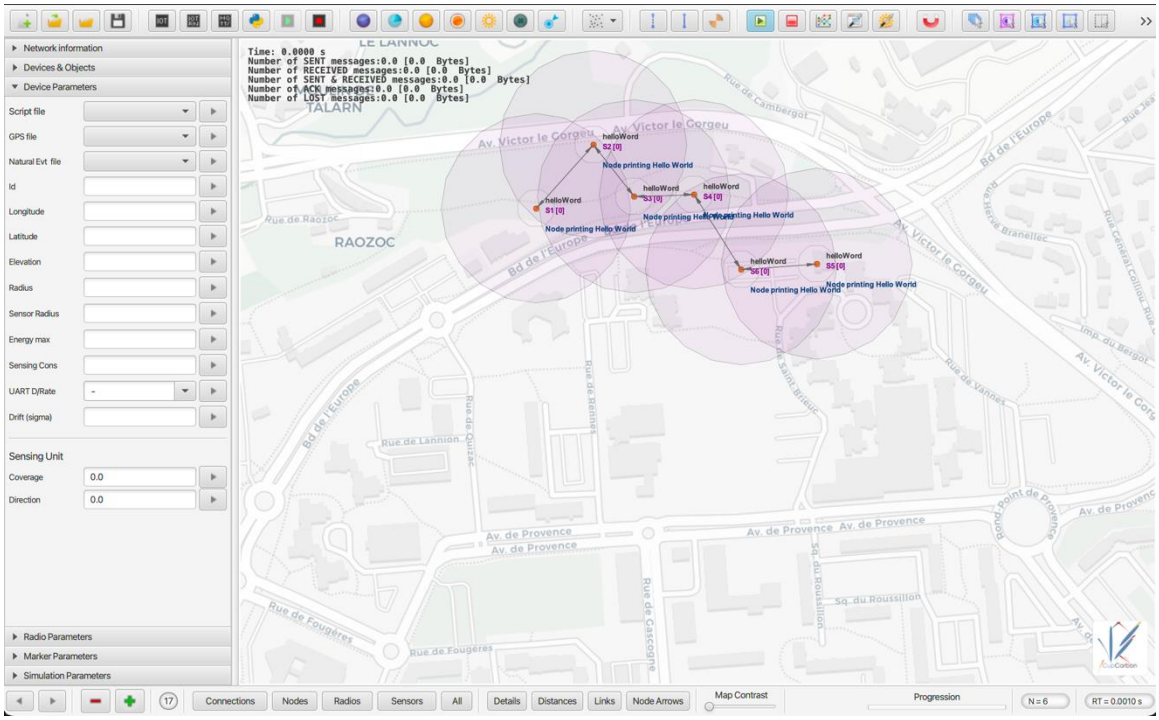
CupCarbon homepage



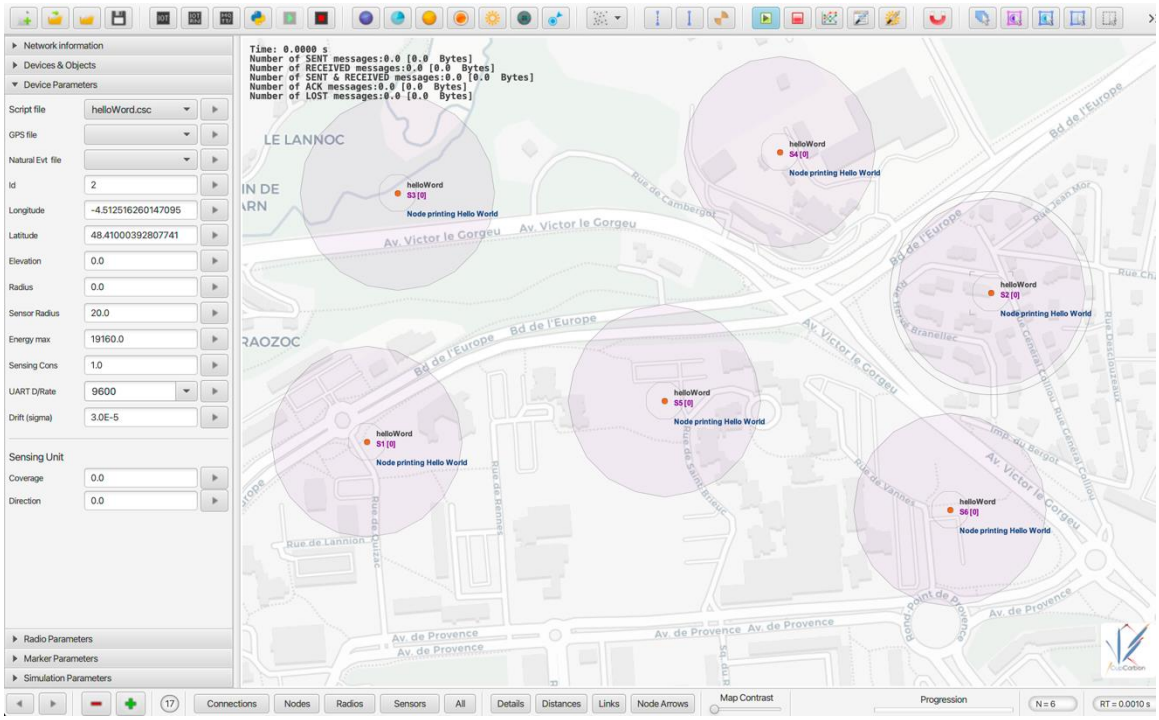
Testing IoT assignment#1 code hello world



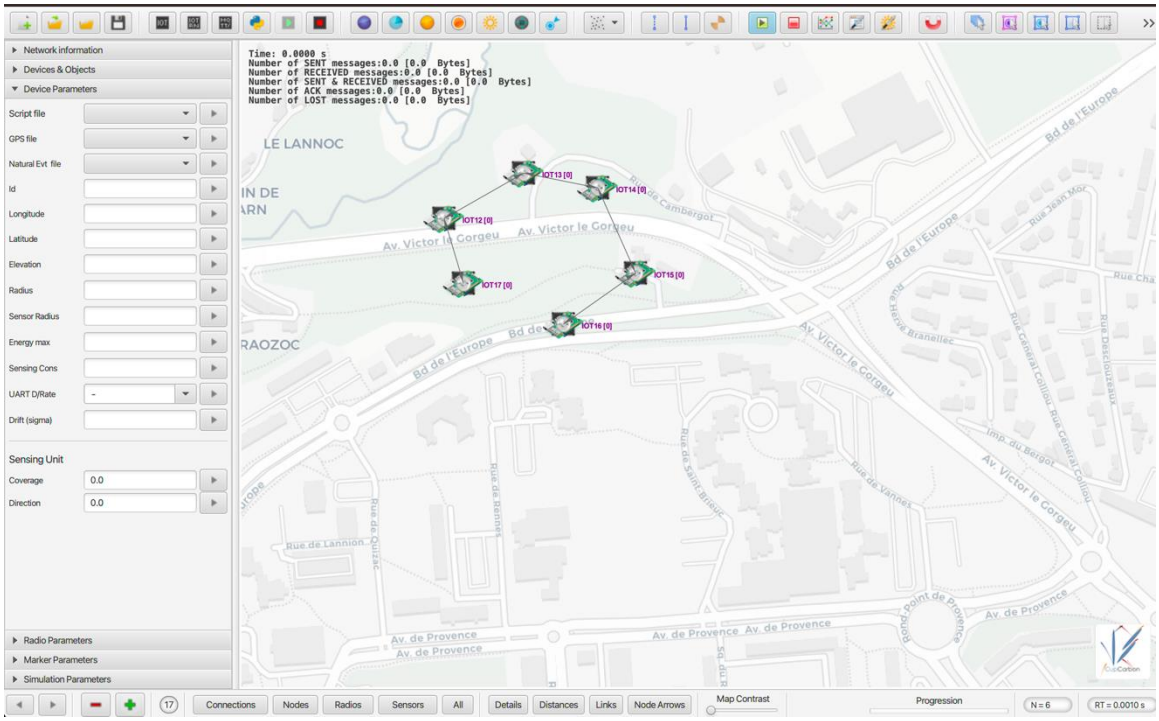
Printing hello world



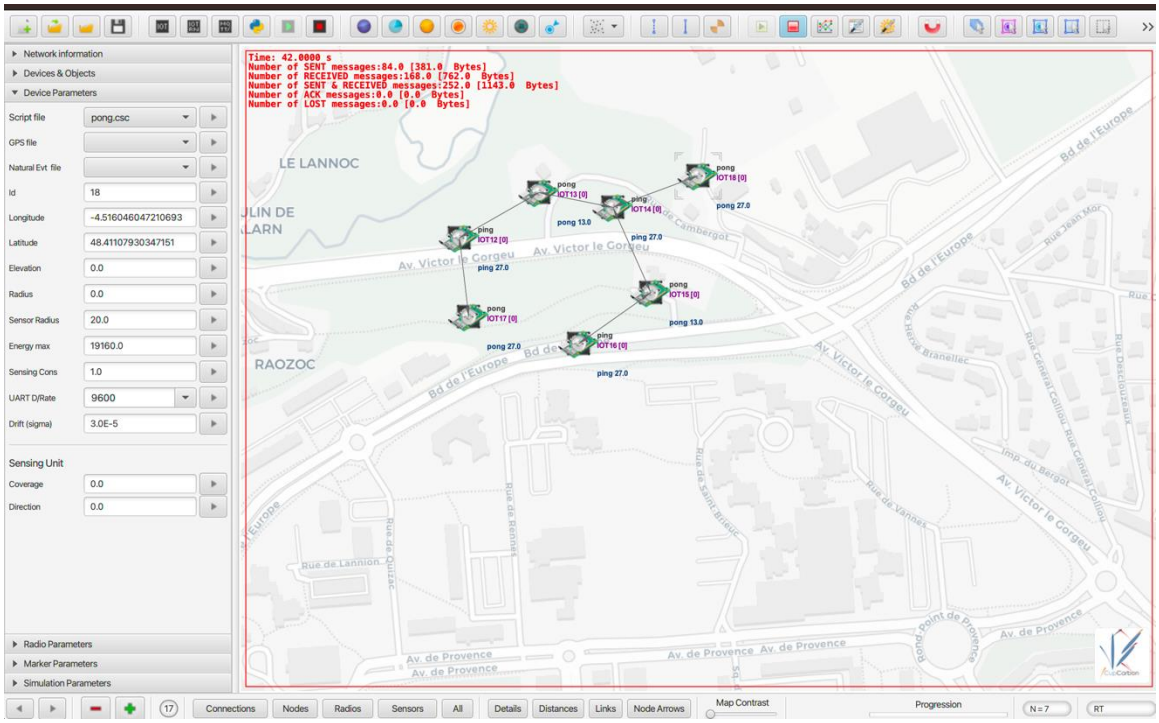
Looping hello world: no communication between the sensor nodes due to distance



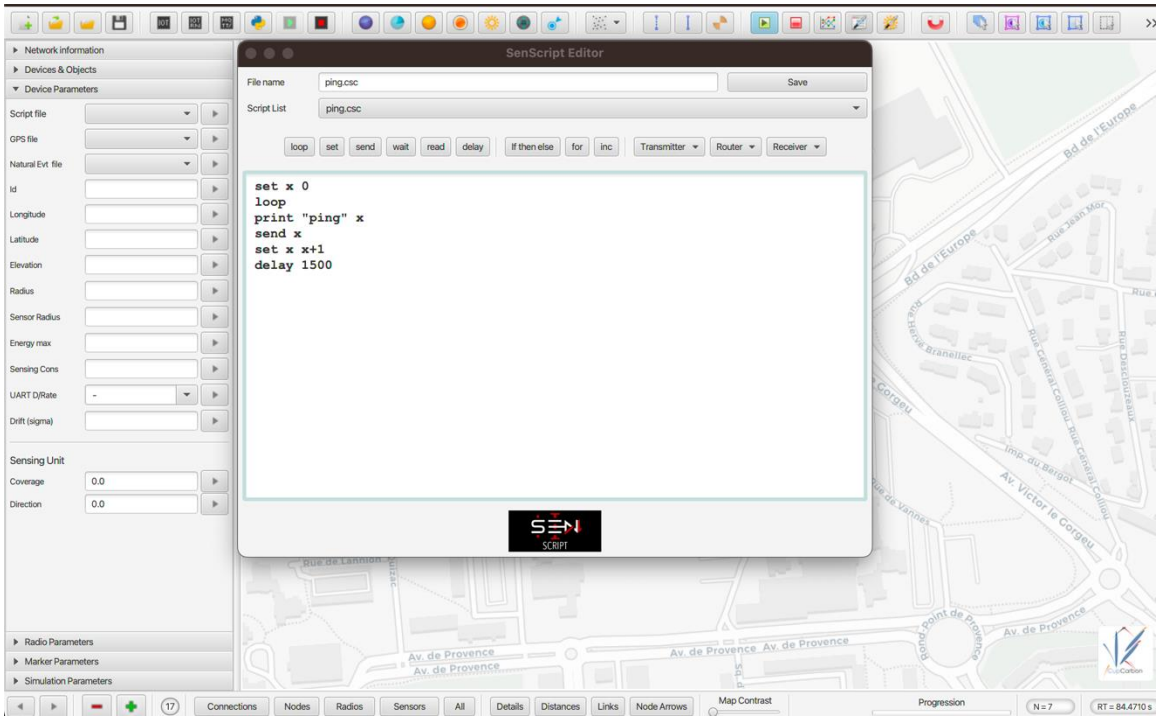
Communication between nodes



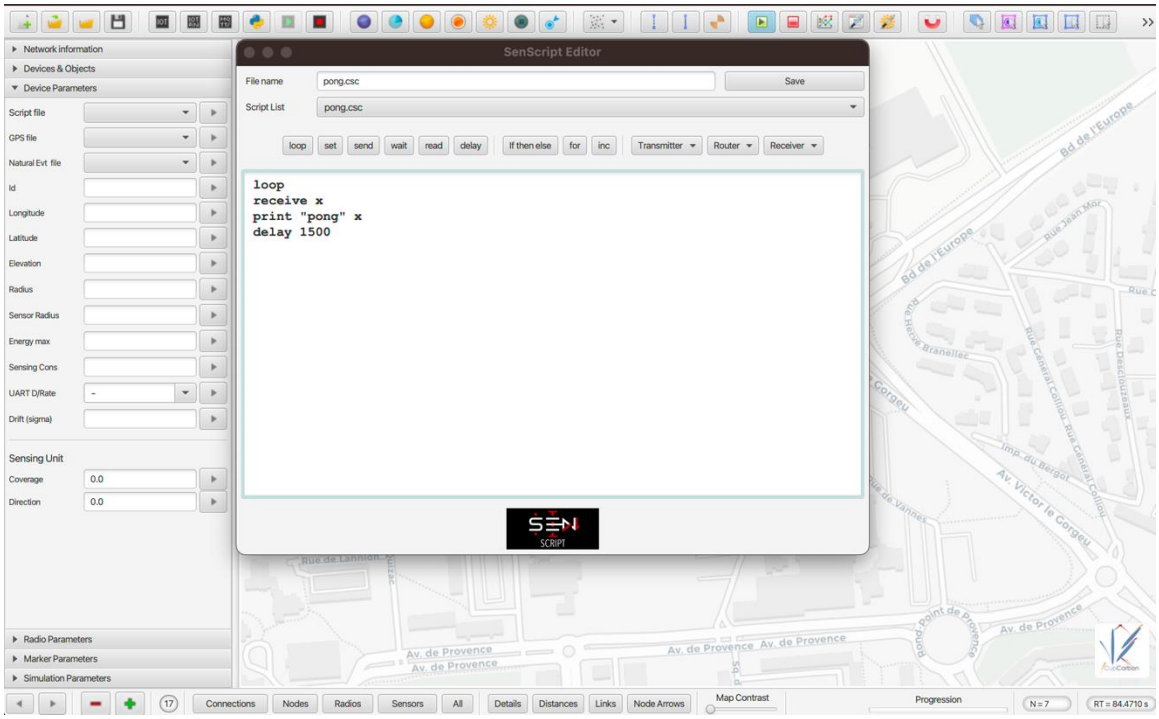
Sending and receiving ping pong



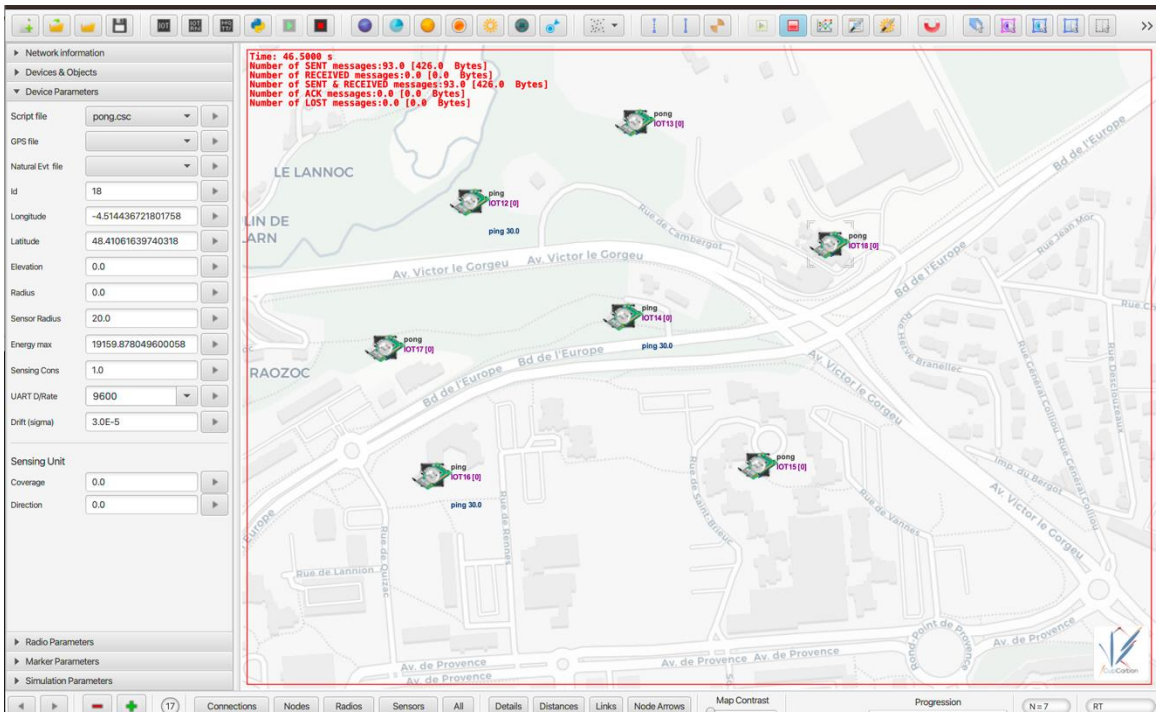
Sending Ping code



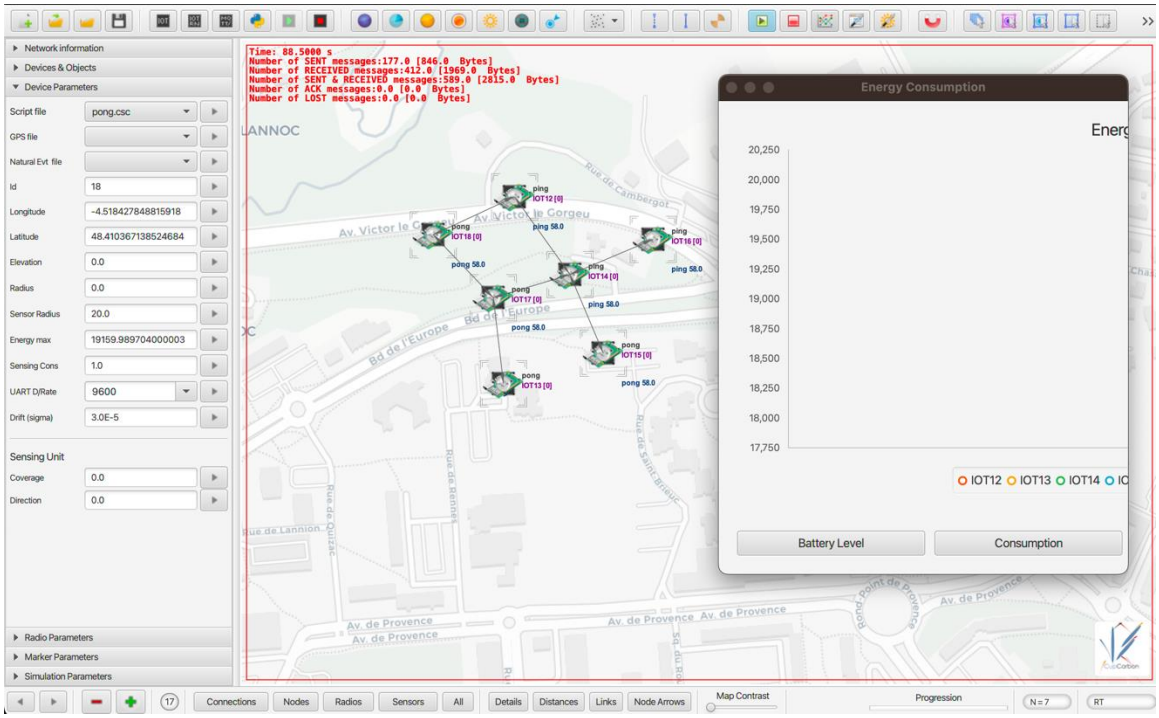
Receiving Pong code



Not Sending or receiving due to distance



Showing energy consumption window



Comparison between CupCarbon and Cooja Simulators for IoT and Wireless Sensor Networks

This report provides a comparative analysis of two prominent simulation tools used in the research and development of Internet of Things (IoT) and Wireless Sensor Networks (WSNs): CupCarbon and Cooja.

1. CupCarbon

1.1. Overview

CupCarbon is an open-source simulation tool designed for IoT and WSN systems. It offers a user-friendly graphical interface, enabling developers and researchers to design, simulate, and test complex scenarios for IoT and sensor networks. CupCarbon is notable for its integration with the Eclipse Integrated Development Environment (IDE), providing a convenient and powerful platform for development.

1.2. Key Features

- Graphical Interface: CupCarbon's graphical interface allows users to visually design their network by placing sensors, devices, and connections on a map, facilitating the conceptualization and modification of network designs.
- Simulation Capabilities: It simulates the behavior of sensors and devices in an IoT network, including energy consumption, communication protocols, and sensor coverage, helping to understand the performance of the network in real-world conditions.
- Scripting Support: Users can write scripts to control the behavior of devices and sensors in their simulation, enabling the testing of complex scenarios and behaviors in IoT systems.
- Real-world Maps: CupCarbon uses real-world geographic information system (GIS) data, allowing the simulation of networks in actual locations, which is crucial for understanding the performance of the network in specific geographical areas.
- Research and Education: CupCarbon is widely used in academic and research settings for teaching IoT and WSN concepts, as well as for conducting research in these fields.
- Integration with Eclipse: The integration with Eclipse provides users with access to the robust features of this IDE, including code editing, management, and debugging tools, facilitating the development of IoT applications and the simulation of sensor networks.

1.3. Supported Languages

CupCarbon primarily supports the following languages:
- Java: CupCarbon is developed in Java, leveraging the cross-platform capabilities and extensive libraries available in Java.
- Scripting Language: CupCarbon uses its own scripting language for defining the behavior of sensors and devices within the simulation environment, enabling users to control the actions and interactions of network elements.

- Uppaal Stratego: For specific tasks related to verifying and optimizing strategies within the networks, CupCarbon can integrate with Uppaal Stratego, a tool for model checking of real-time systems.

1.4. Mote Types

CupCarbon offers a variety of motes (or sensor nodes) to simulate different scenarios in IoT and WSN environments, including:
- Generic Motes: Basic sensor nodes that can be customized to simulate various sensor and actuator functionalities.
- Specific Protocol Motes: Motes designed to simulate specific communication protocols used in IoT and WSNs, such as ZigBee, LoRa, and Bluetooth Low Energy (BLE).
- Mobile Motes: Motes that can move along predefined paths or in response to the simulation dynamics, allowing for the study of dynamic networks.
- Energy-aware Motes: Motes that simulate the energy consumption of sensor nodes, enabling the study of network longevity and sustainability.
- Actuators: Simulated devices that can respond to sensor data or external inputs, allowing for the simulation of complete IoT scenarios.
- Gateway Motes: Motes that simulate the gateways connecting sensor networks to the internet or other networks, enabling the study of data aggregation, transmission, and integration with cloud services.

2. Cooja Simulator

2.1. Overview

Cooja is another prominent simulation tool used in the research and development of IoT and WSNs. It is specifically designed for simulating networks using the Contiki OS, making it highly specialized for Contiki-based IoT devices and applications.

2.2. Key Features

- Integration with Contiki OS: Cooja is tightly integrated with the Contiki OS ecosystem, offering detailed support for Contiki's networking stacks and protocols.
- Network Layer Emulation: Cooja can effectively emulate the network layer, allowing for detailed analysis of communication protocols and interactions within the network.
- Plugin System: Cooja supports plugins, enabling users to extend its capabilities and integrate additional features as needed.
- Community and Resources: Cooja benefits from a dedicated community and a wealth of documentation and resources for Contiki OS users.

2.3. Comparison with CupCarbon

Platform and Protocol Support:
- CupCarbon is more versatile and suitable for a broader range of IoT simulations, while Cooja provides in-depth support for Contiki OS-based simulations.

User Interface and Usability:
- CupCarbon's graphical interface and integration with real-world GIS data make it user-friendly for designing and testing in specific locations, whereas Cooja focuses on network layer emulation and protocol analysis.

Community and Documentation:
- Both tools have dedicated communities, but Cooja might have more concentrated resources for Contiki OS users.

Conclusion

The choice between CupCarbon and Cooja depends on the specific requirements of the project, including the operating system of the devices being simulated, the need for real-world geographical context, and the specific protocols and technologies in use. CupCarbon's versatility and GIS integration make it suitable for a broader range of IoT and WSN simulations, while Cooja's specialization in Contiki OS-based systems can be beneficial for projects within that ecosystem.

References

https://youtu.be/OHHDE76F7Qc?si=m0-J7m8nu6tzk_An

https://github.com/bounceur/CupCarbon

http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

https://www.eclipse.org/

https://o7planning.org/fr/10619/installation-de-e-fx-clipse-sur-eclipse

http://gluonhq.com/products/scene-builder/

CS501 Slides by Dr. Mohammed Ayoub Alaoui Mhamdi https://moodle.ubishops.ca/course/view.php?id=4106