

PRACTICAL-2

AIM

Introduction and implementation of different types of motes and deploy them using IoT architecture in Cooja.

THEORY

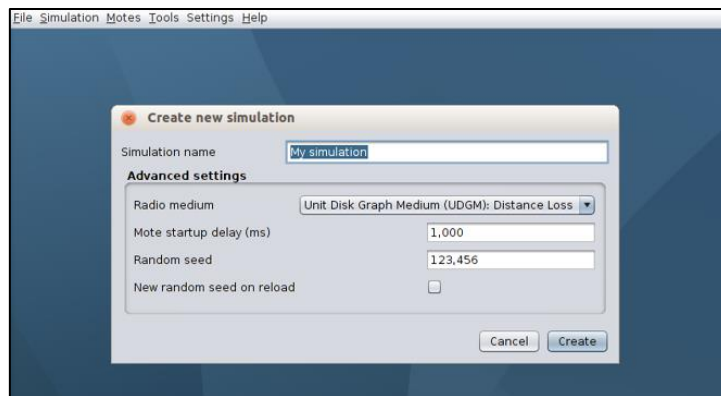
Cooja provides variety of motes to work. Some motes are described below.

- **micaZ mote**
 - The MICAz is a 2.4 GHz Mote module used for enabling low-power, wireless sensor networks.
 - Supported by MoteWorks™ wireless sensor network platform for reliable, ad-hoc mesh networking
 - MoteWork enables the development of custom sensor applications and is specifically optimized for low-power, battery-operated networks.
 - MoteWorks is based on the open-source TinyOS operating system and provides reliable, ad-hoc mesh networking, over-the-air-programming capabilities, cross development tools, server middleware for enterprise network integration and client user interface for analysis and a configuration.
- **sky mote**
 - Tmote Sky is an ultra low power wireless module for use in sensor networks, monitoring applications, and rapid application prototyping.
 - Tmote Sky leverages industry standards like USB and IEEE 802.15.4 to interoperate seamlessly with other devices.
 - By using industry standards, integrating humidity, temperature, and light sensors, and providing flexible interconnection with peripherals, Tmote Sky enables a wide range of mesh network applications.
 - Tmote Sky is a drop-in replacement for Moteiv's successful Telos design.
 - Tmote Sky includes increased performance, functionality, and expansion.
 - With TinyOS support out-of-the-box, Tmote Sky leverages emerging wireless protocols and the open source software movement.
 - Tmote Sky is part of a line of modules featuring on-board sensors to increase robustness while decreasing cost and package size.
- **ESB**
 - The ESB (Embedded Sensor Board) is a prototype wireless sensor network device developed at FU Berlin.
 - The ESB consists of a Texas Instruments MSP430 low-power microcontroller with 2k RAM and 60k flash ROM, a TR1001 radio transceiver, a 32k serial EEPROM, an RS232 port, a JTAG port, a beeper, and a number of sensors (passive IR, active IR sender/receiver, vibration/tilt, microphone, temperature).

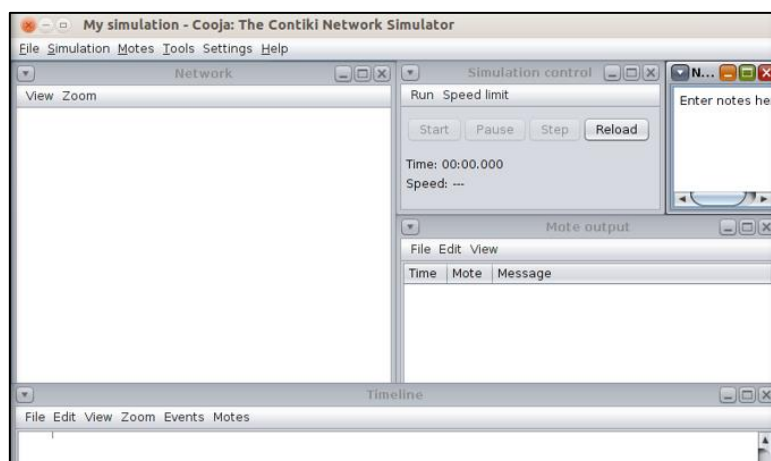
- The Contiki/ESB port contains drivers for most of the sensors. The drivers were mostly adapted from sources from FU Berlin.
- **CC430 mote**
 - The TI CC430 family of ultra-low-power system-on-chip (SoC) microcontrollers with integrated RF transceiver cores consists of several devices that feature different sets of peripherals targeted for a wide range of applications.
 - The architecture, combined with five low-power modes, is optimized to achieve extended battery life in portable measurement applications.
 - The devices feature the powerful MSP430 16-bit RISC CPU, 16-bit registers, and constant generators that contribute to maximum code efficiency.
 - The CC430 family provides a tight integration between the microcontroller core, its peripherals, software, and the RF transceiver, making these true SoC solutions easy to use as well as improving performance.
- **Z1 mote:**
 - The Z1 module is a general purpose development platform for wireless sensor networks (WSN) designed for researchers, developers, enthusiasts and hobbyists.
 - It is a platform compatible with the successful Tmote-family motes with several enhancements that offers roughly a 2x performance in several aspects.
- **Wismote mote:**
 - WiSMote is a sensor/actuator module well adapted to Wireless Sensor Network (WSN) applications.
 - The wireless link operates over the 2.4GHz ISM, a duty free frequency band.
 - With its wide range of embedded sensors and its variety of extension connectors, WiSMote is able to monitor any kind of physical measurements in fields like environment, healthcare, domotics, smart building, logistics or industrial applications.
 - WiSMote embed an small footprint operating system (Contiki) plus an IEEE 802.15.4 protocol stack compatible with Zigbee and 6LoWPAN (IPv6).

PRACTICAL IMPLEMENTATION

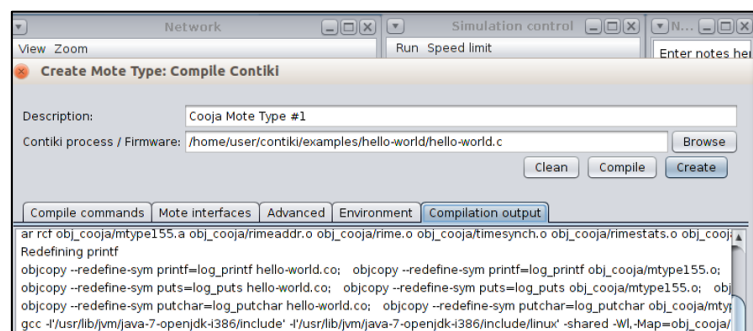
- Open Cooja simulator.
- Go to file > create new simulation



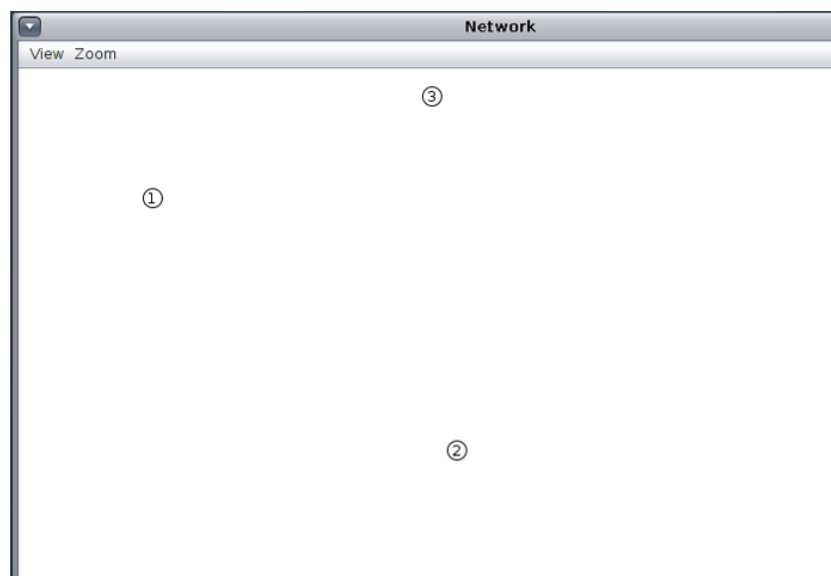
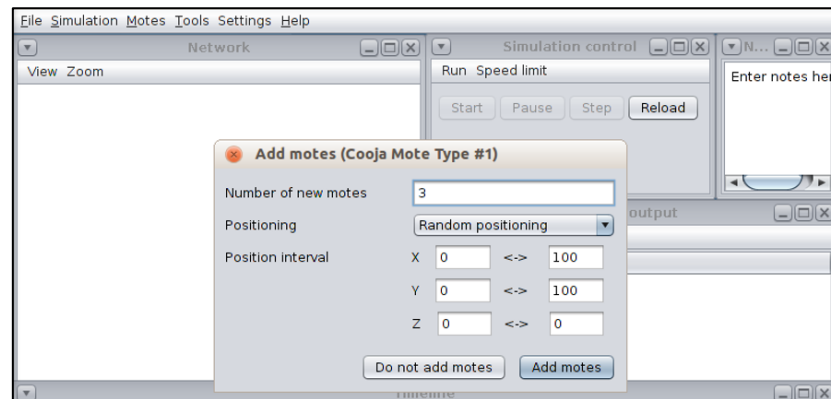
- Give it any name you like and keep the rest of settings as default.



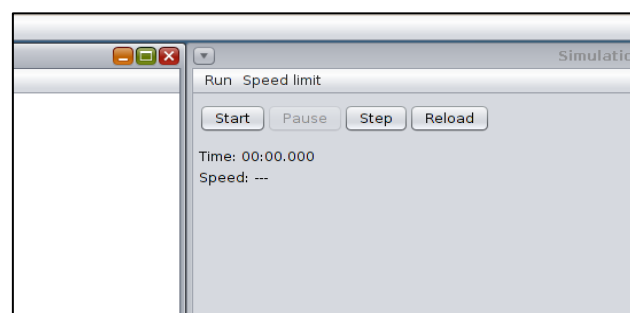
- We will add 2 cooja motes.
- For that, go to Motes > Add motes > Create new mote type > Cooja mote.



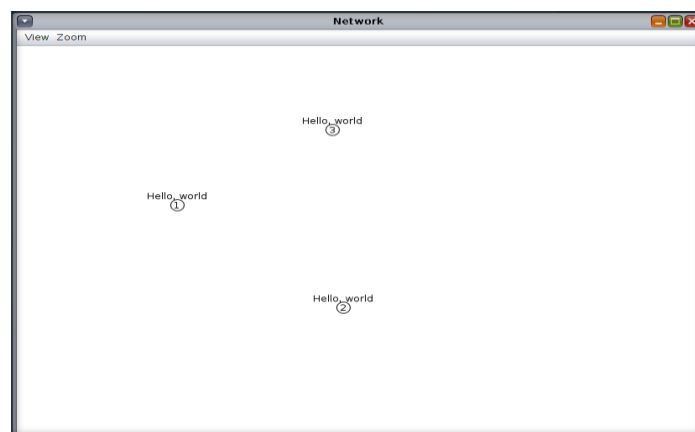
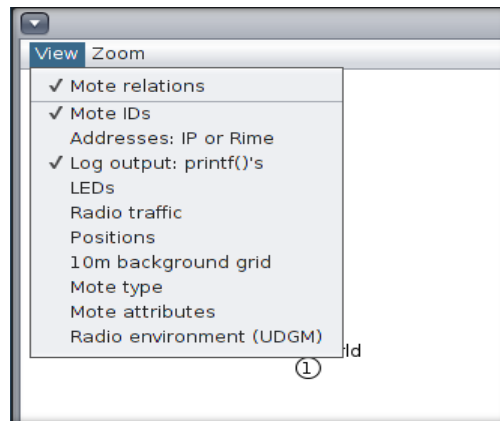
- It will compile the program and show compilation output.
- Click on “create”



- 3 notes will be added at random positions now.
- To run the simulation, click on start.



- Go to view > Log output: printf()'s



Mote output		
Time	Mote	Message
00:00.236	ID:2	MAC 02:00:00:00:00:00 Rime/CSMA/nullrdc, channel check ra...
00:00.236	ID:2	Starting 'Hello world process'
00:00.236	ID:2	Hello, world
00:00.382	ID:1	Contiki-2.6-900-ga6227e1 started. Node id is set to 1.
00:00.382	ID:1	Rime started with address 1.0
00:00.382	ID:1	MAC 01:00:00:00:00:00 Rime/CSMA/nullrdc, channel check ra...
00:00.382	ID:1	Starting 'Hello world process'
00:00.382	ID:1	Hello, world
00:00.899	ID:3	Contiki-2.6-900-ga6227e1 started. Node id is set to 3.
00:00.899	ID:3	Rime started with address 3.0
00:00.899	ID:3	MAC 03:00:00:00:00:00 Rime/CSMA/nullrdc, channel check ra...
00:00.899	ID:3	Starting 'Hello world process'
00:00.899	ID:3	Hello, world

CONCLUSION

In this practical, I learnt about different types of motes.