

## **Installation of Contaki OS:**

1. Visit the Website URL: <http://www.contiki-os.org/download.html>
2. Click on [Download Instant Contiki](#) Link and download Contaki 2.7 image
3. Download and install VMWare Player form the link:  
[https://my.vmware.com/en/web/vmware/free#desktop\\_end\\_user\\_computing/vmware\\_workstation\\_player/15\\_0](https://my.vmware.com/en/web/vmware/free#desktop_end_user_computing/vmware_workstation_player/15_0)
4. Mount the downloaded Contaki 2.7 image on to VM Player
5. Power on the system and Log into Instant Contiki. The password is **user**

## **Starting Cooja:**

1. open a terminal window and go to the Cooja directory: `cd contiki-2.7/tools/cooja/`
2. Start Cooja with the command: `ant run` and When Cooja first starts, it will first compile itself, which may take some time.
3. When Cooja is compiled, it will start with a blue empty window.

**Note:** Contiki is OS but Cooja is a framework.

## **Running Cooja Simulations:**

1. Click the **File** menu and click **New simulation....**
2. Cooja now opens up the **Create new simulation** dialog. In this dialog, we may choose to give our simulation a **new name**, but for this example, we'll shall give **sois1**. Click the **Createbutton**.
3. Cooja brings up the new simulation which have **five windows**:
  - a. **Network Window:** at the top left of the screen, shows all the notes in the simulated network - it is empty now, since we have no notes in our simulation.
  - b. **Simulation Control Window:** Here we start, pause, and reload our simulation.
  - c. **Notes Window:** on the top right is where we can put notes for our simulation.
  - d. **Note Output:** on the right side of the screen, shows all serial port printouts from all the notes.

- e. **Timeline Window:** at the bottom of the screen, shows all communication events in the simulation over time - very handy for understanding what goes on in the network.

#### 4. Add Motes:

- a. Before we can simulate our network, we must add one or more motes. We do this via the **Motes menu**, where we click on **Add motes....** Since this is the first mote we add, we must first create a mote type to add. Click **Create new mote type...** and select one of the available mote types. For this example, we click **Sky mote...** to create an emulated Sky mote type.
- b. Cooja opens the **Create Mote Type** dialog, in which we can choose a name for our mote type as well as the Contiki application that our mote type will run. For this example, we stick with the suggested name, and instead click on the **Browse...** button on the right hand side to choose our Contiki application.

#### 5. Contiki process/Firmware:

- a. Go to the directory `/home/user/contiki/examples/hello-world/`
- b. Choose the file **hello-world.c**. This file contains a simple Contiki application that randomly broadcasts a UDP packet to its neighbors. Click the **Open button** to choose the file.
- c. Now Cooja will verify that the selected Contiki application compiles for the platform that we have selected. Click the **Compile button**. This will take some time the first time around, expect it to take a minute at least. The compilation output will show up in the white panel at the bottom of the window.
- d. Click the **Create button** to create the mote type. The window will close.
- e. Cooja will now ask us if we want to add motes from the newly created mote type to the simulation. We change the number of motes to add in the **Number of motes** field to 10.
- f. We click the **Add motes** button to add the motes to the simulation.
- g. We can now see the 10 motes we added to the simulation in the **Network window**. Click the **Start button** to start the simulation.
- h. We see printouts from the simulated motes appearing in the **Mote output window**. The **Network** window shows communication going on in the network.

The **Timeline window** shows communication and radio events over time - the small gray lines are ContikiMAC periodically waking the radio up. We can click the **Pause button** to pause the simulation.

### **Executing Project From Terminal:**

1. From Home folder navigate to following location: ***cd contiki-2.7/examples/raghu-sois/***
2. Enter command ***make***
3. Enter command : ***make TARGET=native***
4. Enter command : ***make TARGET=sky***
5. Enter command : ***make TARGET=micaz***

Compilation for contiki environment  
No hardware

6. To run the file for native environment run the command: ***./myFirst.native***
7. To run the file for skymote run the command: ***./myFirst.sky***

***Note: here it shows cannot execute binary file as no hardware device(sky mote, micaz contiki doesn't support micaz as is has only 64kb size) is connected***

### **Loading an Application on to a Hardware:**

1. ***Make TARGET=sky savetarget***
2. ***make appname.upload***
3. ***make login or sudo make login***

***Note: Check the size of myFirst.native and myFirst.sky and interpret.***

### **LED BLINKING (ENABLE AND DISABLE LED ON MOTES) from Terminal:**

1. Create a folder by name raghu-led in path ***/home/contiki-2.7/examples*** using command ***mkdir raghu-led***
2. Create a C file by name ***ledmotes.c***
3. Create a proper make file for ***ledmotes***
4. Compile the project led using command ***make or make TARGET=native*** for contiki OS Execution.

5. Execute the LED project using the command: *./ledmotes.native*
6. See the output from 0 to 7 as motes have 3 led (binary)
7. Compile the project led using command *make or make TARGET=sky* for SKY Motes.
8. Execute the LED project using the command: *./ledmotes.sky*
9. The terminal throws an error (*cannot execute binary file*) as we don't have any physical sky motes.

### **LED BLINKING (ENABLE AND DISABLE LED ON MOTES) from Cooja:**

#### **LED Control Using Button Sensor:**

Scenario: When user press a button LED should ON (glow) and on next press LED should OFF  
Use SKY Mote.

You cant execute this in native mode (but you can compile) as we required a button we should do it in cooja simulator

1. Create a C File named *buttonled.c*
2. Create a a proper make file for *buttonled.c*
3. Open Cooja, use sky mote and compile for skymote.
4. Select 1 skymote, open LED panel by right click on mote and selset show LED's on sky1.
5. Right click on mote, select mote interface, from drop down menu select button.
6. Start the simulation, press button and view the output.