
Basics of Contiki-OS and using it for Wireless Sensor Network Applications

Shantanoo Desai
prepared for:
Prof. Dr. Anna Förster

Sustainable Communication Networks
University of Bremen

20th November 2015

Outline

CONTIKI-OS

- Contiki in a nutshell

- Requirements

Getting Started

- Initial Steps

- File Structure in Contiki

- Terminal Basics

First Program in Contiki: Hello-World

- Programming using Terminal

- Hello-World files

- Getting Output

- Understanding codes in Contiki

Programming a Sensor Node

- TelosB Sensor Node

- Connecting the TelosB to Contiki-OS

- Hello-World program on TelosB

Cooja Simulator in Contiki-OS

- Hello-World Simulation with Cooja

- Adding Sensor Nodes to Cooja

- Mote Output in Cooja

References

Contents of this section

CONTIKI-OS

Contiki in a nutshell

Requirements

What is Contiki?

CONTIKI-OS in a nutshell:

- Complete environment for programming Sensor Nodes
- Has everything for getting started in making Applications
- In-built simulator called COOJA
- Large pool of sensor compatibility e.g. TelosB, Zolertia Z1

Requirements

Requirements before we begin:

- VMware Virtual Player
 - VMware Workstation 12 Player (recent)
- Instant Contiki
 - Instant Contiki version 2.7/3.0

LOST??? - refer to this :

www.contiki-os.org/start.html and follow the steps.

Contents of this section

Getting Started

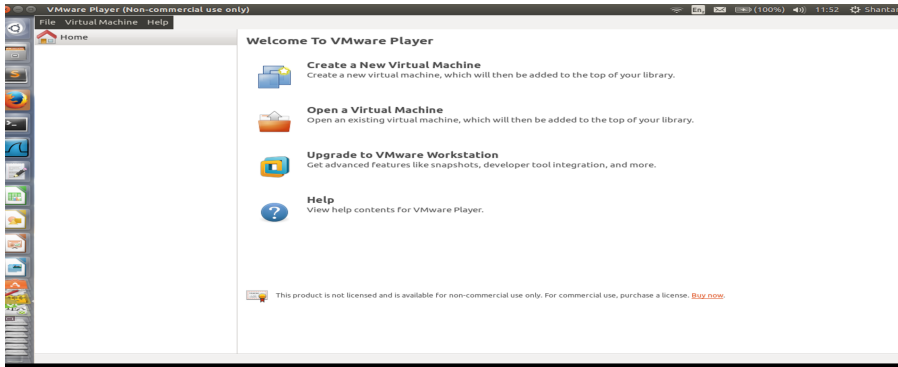
Initial Steps

File Structure in Contiki

Terminal Basics

Getting Started

- Open VMWare player and click on 'Open a Virtual Machine'



(Don't Worry ! if it looks different for Windows or MAC-OS! this is for Ubuntu.)

Getting Started (contd.)

- Navigate to your Instant Contiki 2.7 folder and select the “.vmx” file
- After booting of the virtual machine, Login with password: **user**

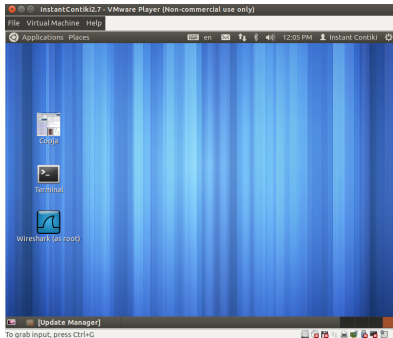


Figure: First look of Contiki-OS

File Structure in Contiki

- Click on 'Places' and then 'Home' folder (top left corner)
- target folders: **contiki** & **contiki-2.7** (choose any one and see the folders)

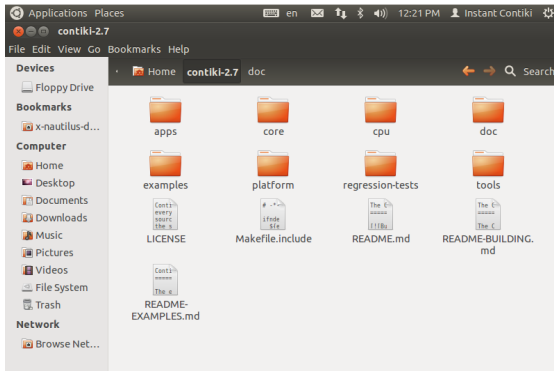


Figure: File Structure in Contiki

Folders in Contiki and their Usage

- **apps**: applications like webbrowser, telnet etc.
- **core**: source codes for main core of Contiki
 - **core/dev**: source codes for devices such as LED, battery sensor, button etc.
 - **core/net**: folders for MAC, and RPL routing protocol, IPv6 and IPv4, queuing packets and buffers etc.
- **cpu**: source files for all computational units for sensor nodes
- **examples**: Implementation of applications
- **tools**: general tools for testing applications

Terminal basics

[fragile] open terminal on the Desktop screen

- **pwd** shows the present working directory
- **ls** lists all the files/folders in the present directory
- **cd FOLDERNAME** to change directory from present to the desired one (example: from home directory to contiki-2.7)
- **cd ..** return to the previous directory
- **mkdir FOLDERNAME** makes a directory in the present working directory
- **gedit FILENAME** if you want to open a GUI based text editor

Contents of this section

- First Program in Contiki: Hello-World
 - Programming using Terminal
 - Hello-World files
 - Getting Output
 - Understanding codes in Contiki

Programming Basic: Hello-World

Programming using Terminal in Contiki-OS is preferred choice since it makes creation of files, compilation and outputs easier to view and manage.

1. Open terminal, change directory to either **contiki** or **contiki-2.7**¹
2. change to the *examples* folder
current directory *contiki-2.7/examples/*
3. change to the *hello-world* folder
current directory *contiki-2.7/examples/hello-world*

¹Hint: type `cd` and type half of the word and press Tab key for the completion

Inside the Hello-world folder

Files observed in the folder ²

- **hello-world.c**: simple program in Contiki-OS
- **Makefile**: file to control compilation through terminal
- **hello-world.csc**: simple simulation file using COOJA simulator
- **README.md**: a basic manual for the example

²HINT: use **ls** command

Getting Outputs in Contiki-OS

Assuming the present working directory is *contiki-2.7/examples/hello-world* do the following:

- Currently without any sensor node inserted into the USB slots, in Terminal type **make** and press enter
- after processing is done, type **./hello-world.native**³
- Observe the Output "Hello, world"
- to terminate program press **CTRL+C**

```
user@instant-contiki: ~/contiki-2.7/examples/hello-world
File Edit View Search Terminal Help
user@instant-contiki:~/contiki-2.7/examples/hello-world$ ./hello-world.native
Contiki 2.7 started
Rime started with address 2.1
MAC nullmac RDC nullrdc NETWORK NETWORK Rime
Hello, world
█
```

³use Tab key instead of typing everything

hello-world.c: How does it work?

```
#include "contiki.h" /* For contiki applications */
#include <stdio.h> /* printf() function usage */

/* Contiki application : Declare the PROCESS */
PROCESS(name_of_your_process, "Process Name");
/* Start your PROCESS */
AUTOSTART_PROCESSES (&name_of_your_process);

/* Declare what the PROCESS DOES*/
PROCESS_THREAD(name_of_your_process, ev, data)
{
PROCESS_BEGIN(); /* Begin the PROCESS*/
printf("Hello, world");
PROCESS_END(); /* end the PROCESS */
}
```


Makefile in Contiki

Makefile is used to compile the source code into appropriate object files for execution. In the main directory there exists a **Makefile.include** that is necessary for compilation with or without platforms.

```
# Comments using Pound Sign (#) in Makefile
CONTIKI_PROJECT = your-filename/s #Can be more file
all: $(CONTIKI_PROJECT) # take all the files
```

current folder is *contiki-2.7/examples/hello-world* and you want to find the **Makefile.include** in *contiki-2.7* folder and use it here

```
CONTIKI = ../..
include $(CONTIKI)/Makefile.include
```

Contents of this section

Programming a Sensor Node

- TelosB Sensor Node

- Connecting the TelosB to Contiki-OS

- Hello-World program on TelosB

TelosB Sensor Node Functionalities

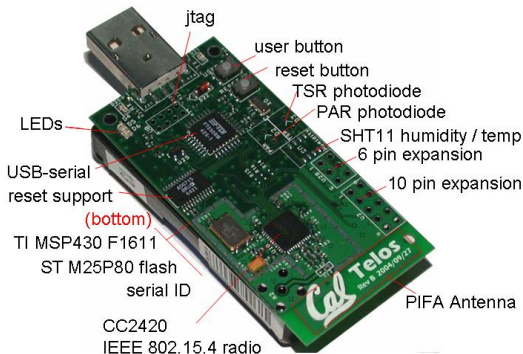


Figure: TelosB Sensor Node

Picture Courtesy: www.wsnblog.com

Initial Connectivity check for Sensor node

To check connectivity to the Virtual Machine do the following:

- Insert the TelosB in USB slot
- In the Virtual Machine Player click on the tab 'Virtual Machine'
- scroll to 'Removable Devices' and check for TelosB
- click on 'TelosB' and click on 'Connect(disconnect from host)'

It is always advisable to perform these steps when connecting/disconnecting sensor nodes to Contiki-OS

Hello-World on TelosB

In *contiki-2.7/examples/hello-world* do the following in the Terminal:

```
make TARGET=sky savetarget
make motelist
make hello-world.upload
make login
```

Understanding each line:

- the first input will make the directory target only for TelosB Sky node (good to do when using only one type of Sensor node)
- the second input will display on which USB port the TelosB is connected
- the third input will upload the code onto the TelosB
- the last line will create a login to the TelosB
- Press **RESET button** on the TelosB and observe output

Output on TelosB

```
connecting to /dev/ttyUSB0 (115200) [OK]
Rime started with address 50.0
MAC 32:00:00:00:00:00:00:00 Contiki 2.7 started.
Node id is set to 50.
CSMA ContikiMAC, channel check rate 8 Hz,
radio channel 26
Starting 'Hello world process'
Hello, world
```

Contents of this section

Cooja Simulator in Contiki-OS

 Hello-World Simulation with Cooja

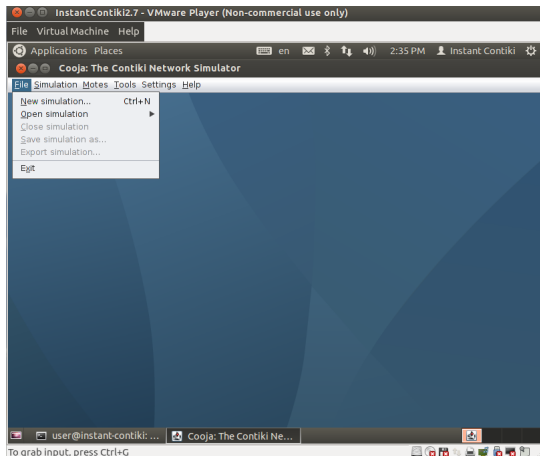
 Adding Sensor Nodes to Cooja

 Mote Output in Cooja

Running Cooja in Contiki

In Terminal:

```
$ cd contiki-2.7/tools/cooja  
$ ant run
```



Getting Started

- Click on **File – New Simulation**
- Give a name for the simulation project, click **Create**

Create new simulation

Simulation name

Advanced settings

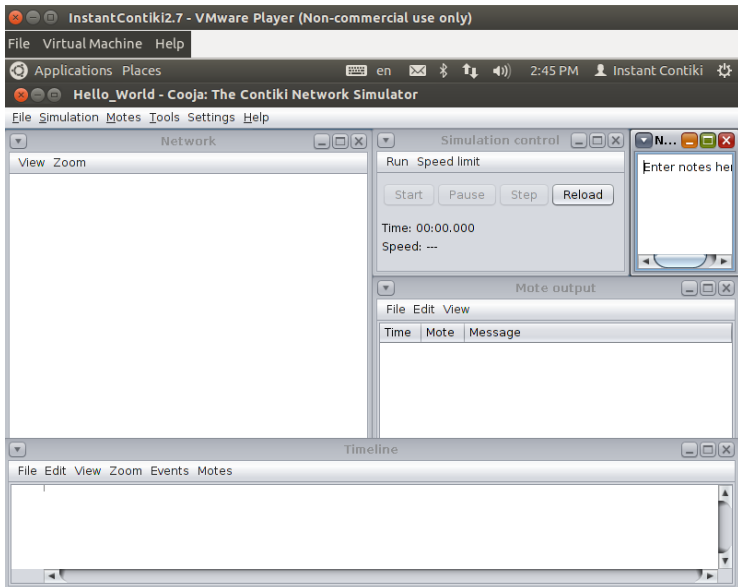
Radio medium

Mote startup delay (ms)

Random seed

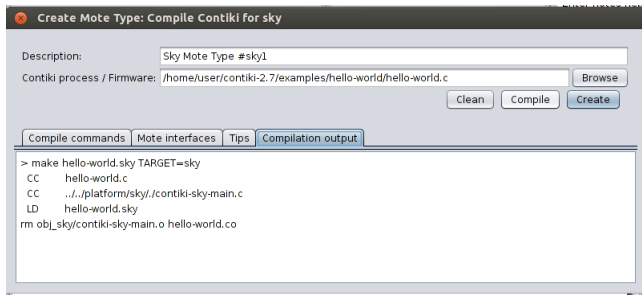
Simulation random seed. Controls the random behavior such as mote startup delays, node positions etc.

Cooja Environment



Adding Sensor Nodes (Motes) in Simulation

- Click on **MOTES – Add MOTES**
- In the **Create New Mote Type**, select **Sky Mote**
- In the Dialog Box that appears, click on **Browse** and navigate to *contiki-2.7/examples/hello-world*
- select the *hello-world.c* file and click on **Compile** button, After that click on **Create**



Adding Sensor Nodes (Motes) in Simulation

- after pressing **Create**, window to add number of Motes in Simulation and their positions appears
- add 2 or any number of motes as per wish ...

Add motes (Sky Mote Type #sky1)

Number of new motes:

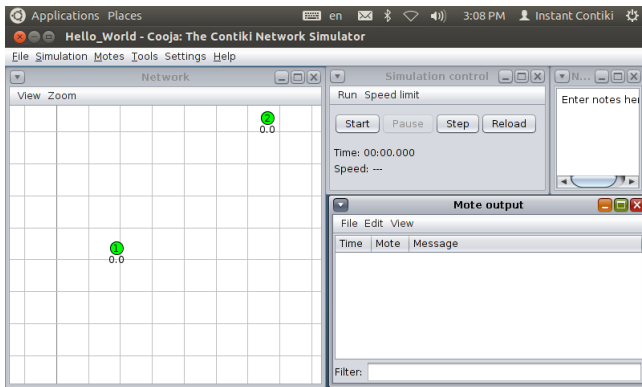
Positioning:

Position interval:

X	<input type="text" value="0"/>	<->	<input type="text" value="100"/>
Y	<input type="text" value="0"/>	<->	<input type="text" value="100"/>
Z	<input type="text" value="0"/>	<->	<input type="text" value="0"/>

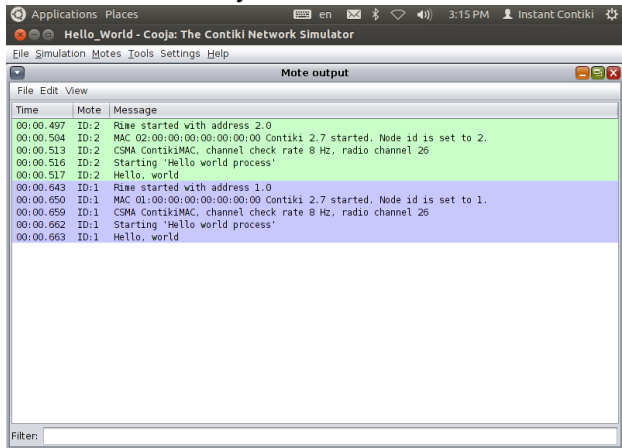
Adding Sensor Nodes (Motes) in Simulation

- The motes are placed randomly in the **Network** dialog box.
- use the **View** title-bar option in the **Network** for adding visual aide to the simulation
- Press **Start** to run simulation



Mote Output in Cooja

Observe the output in the **Mote Output** Dialog box . . .
Similar to the output in Terminal with actual Sky mote connected but only difference of MAC address



The screenshot shows the Cooja network simulator interface. The main window title is "Hello_world - Cooja: The Contiki Network Simulator". Below the menu bar, there is a "Mote output" dialog box. The dialog box contains a table with the following data:

Time	Mote	Message
00:00.497	ID:2	Rime started with address 2.0
00:00.504	ID:2	MAC 02:00:00:00:00:00 Contiki 2.7 started. Node id is set to 2.
00:00.513	ID:2	CSMA ContikiMAC, channel check rate 8 Hz, radio channel 26
00:00.516	ID:2	Starting 'Hello world process'
00:00.517	ID:2	Hello, world
00:00.643	ID:1	Rime started with address 1.0
00:00.650	ID:1	MAC 01:00:00:00:00:00 Contiki 2.7 started. Node id is set to 1.
00:00.659	ID:1	CSMA ContikiMAC, channel check rate 8 Hz, radio channel 26
00:00.662	ID:1	Starting 'Hello world process'
00:00.663	ID:1	Hello, world

At the bottom of the dialog box, there is a "Filter:" input field.

Exploring More in Cooja

- Right Click on any mote and go through many options available.

Make Projects run in COOJA from Terminal

- In the Terminal navigate to the desired project folder which has a '.csc' file in it
- in Terminal

```
make TARGET=cooja filename.csc
```

Last Line automatically starts the simulator without having to manually add motes.

REFERENCES

for better in-depth understanding for Programming in CONTIKI

- **Contiki-WIKI:**

<https://github.com/contiki-os/contiki/wiki>

- **PROCESSES:**

<https://github.com/contiki-os/contiki/wiki/Processes>