

# Running DACA2 over Contiki And Matlab

---

This guide covers the installation of Contiki OS 3.0. and matlab software. You will need either a Linux-based operating system, or an operating system that supports VirtualBox or VMWare Player.

Contiki is an open source operating system for the Internet of Things, it connects tiny low-cost, low-power microcontrollers to the Internet.

With Contiki, development is easy and fast: Contiki applications are written in standard C, with the Cooja simulator Contiki networks can be emulated before burned into hardware, and Instant Contiki provides an entire development environment in a single download.

Visit the [Contiki OS site](#) for more information. The following references are the must-go places to search for detailed information:

- [The Contiki wiki page](#)
- [The Contiki mailing list](#)

## Install Contiki

There are several ways to install Contiki, from scratch by installing from sources or using virtual environments, depending on the flavour and time availability.

To work with Contiki you will need three items:

- The Contiki source code [for DACA2](#)
- A target platform (virtual platform or a real hardware one).
- A toolchain to compile the source code for such target platform.

This document assumes Contiki will run in an Unix environment, as the virtualized environments run on Ubuntu.

## Fresh Contiki Installation

The following instructions will guide you to install Contiki on your machine. These instructions were tested in Ubuntu-like devices (version 12.04 and onwards). If you are

looking for a ready to use setup, skip this section and download one of the Virtual Machines in the next section.

This will install support for the ARM Cortex-M3 and MSP430 platforms, as well as support for Cooja, the Contiki's emulator to be discussed in the next sections.

### **Install the toolchain and required libraries**

The following are the minimal recommended libraries to run Contiki.

#### **Ubuntu (Linux) installation**

To install the toolchain and required dependencies, run in a terminal the following:

```
sudo apt-get update
sudo apt-get install gcc-arm-none-eabi gdb-arm-none-eabi
sudo apt-get -y install build-essential automake gettext
sudo apt-get -y install gcc-arm-none-eabi curl graphviz unzip wget
sudo apt-get -y install gcc gcc-msp430
sudo apt-get -y install openjdk-7-jdk openjdk-7-jre ant
```

### **Get Contiki on your machine**

Contiki source code is actively supported by contributors from universities, research centers and developers from all over the world. The Contiki DACA2 source code is available at:

<https://github.com/davidsantosb/Contiki-DACA2>

The source code is hosted at [Contiki GitHub repository](#)

Nevertheless this repository does not include latest modifications for DACA including RPL stop and RPL leaf mode during run-time that are modified over the files of Contiki RPL in: contiki\core\net\rpl

## **Upload the DACA2 algorithm to the nodes**

### **Upload Border-Router on the platform**

At Contiki-Daca2 over the folder contiki\examples\ipv6\rpl-border-router are the Files to Compile the Border-Router.

First you need to choose the platform to be uploaded, for sky motes use:

```
Make TARGERT=sky savetarget
And for Re-motes use
```

```
Make TARGERT=zoul savetarget
```

After you choose the target, to compile and upload the border-router to a sky or a remote platform run in a terminal:

```
make border-router.upload
```

When the border-router example is uploaded to the sky platform run on the terminal:

```
Make connect-router
```

This will open the tunslip6 interface and open an interface tun0 to connect to the border router. By default it will configure prefix: "aaaa::0"

### **Upload udp-sender on the platform**

At Contiki-Daca2 over the folder `contiki\examples\ipv6\rpl-collect` are the files to Compile the udp-sender. To compile the code of `udp-sender.c` to a sky or a remote select the platform as it was done for the border router and run in a terminal:

```
Make TARGERT=sky savetarget % for Sky motes
Make TARGERT=zoul savetarget % for Re-motes
make udp-sender.upload
```

You can upload this code to several sky motes at once just by connecting them all through a USB port and it will upload the code to all the motes.

## **DACA2 on Matlab software**

After programming the motes and creating the tunslip interface open the matlab (preferable version 2015a) and upload the IP addresses and positions X and Y of each mote as it is done on the matlab workspace over the file `udp_comm_cooja.m`:

A brief example can be seen here:

```
IPs=cell(10,1);
```

```
IPs(1) = cellstr('aaaa::212:7401:1:101');
```

This will declare as string the IP direction of node 1.

The positions of the nodes must be delivered on an int array called `pos` of  $2 \times N$  where  $N$  is the number of nodes.

Position of node 1 would be for example `pos(1, :) = 0,0`

After all the IPs and positions are saved execute file WirelessSensorNetworkV4.m to execute the algorithm and then run section UDP Message sender on udp\_comm\_cooja.m to send all the CH0 and CH1 configuration messages to the nodes. If you have the values of the batteries and you wish to update them to matlab use the example code on section Table Search on the same file.

After this you have:

- Programed all the nodes and started border-router, hence RPL
- Selected the subset of nodes using K-means and PSO
- Configured operation modes on all the nodes.

Next you need to stop RPL so press the button once of the border-router. This would stop the RPL execution. Next execute the section of udp\_comm\_cooja.m called Check for the dead node with the number of times you want to check for a dead node. This can be any number. After the program finds a message with a node informing of a dead node re execute the file WirelessSensorNetworkV4.m this will automatically reselect the subset of nodes taking into account there is a dead node which wont be selected.

This is all you need to reproduce the DACA2 Algorithm over a real WSN.

For further questions or information contact me at:

[santosh@javeriana.edu.co](mailto:santosh@javeriana.edu.co)

Thanks for your attention.