

# HomeLab: Wireless Sensor Networks

*Università degli Studi dell'Aquila*  
*ICT Engineering*

**Embedded Systems (Dr. L. Pomante)**  
**A.A. 2017/18**

Ver. 4.1 22/11/2017

# HomeLab: Wireless Sensor Networks

## HomeLab WSN Kit:

- 3 sensor nodes Memsic/Crossbow IRIS
- 2 sensor boards Memsic/Crossbow MDA100CB (or BASICSB, or similar)
- 1 programming board Memsic/Crossbow MIB520, USB cable (optional).
- SW to be downloaded and installed: driver MIB520 Windows, WMware Player Windows, virtual machine [DebTos\\_1](#)
- Official site: [www.tinyos.net](http://www.tinyos.net)

**Constrained HomeLab: the kit has to be returned in 5-7 days**

# HomeLab WSN: Environment setup

## Installation:

- Follow your “holy book” at the link...  
[http://tinyos.stanford.edu/tinyos-wiki/index.php/Main\\_Page](http://tinyos.stanford.edu/tinyos-wiki/index.php/Main_Page)
- ...and use one of the methods (basing on your OS and skills) listed in:  
[http://tinyos.stanford.edu/tinyos-wiki/index.php/Installing\\_TinyOS](http://tinyos.stanford.edu/tinyos-wiki/index.php/Installing_TinyOS)

## Or

- Follow the method of previous version of HomeLab and download the VM from:  
<http://sing.stanford.edu/tinyos/dists/old/xubuntos-2.1-vm.tar.gz>

## Or

- Download the new VM (DebTos) from:  
[https://www.dropbox.com/s/ma67hoa8v4nk5fl/DebTos\\_1.7z?dl=0](https://www.dropbox.com/s/ma67hoa8v4nk5fl/DebTos_1.7z?dl=0)
- And use other facilities present in this guide
- **Note: VMWare Workstation Player**  
([https://my.vmware.com/en/web/vmware/free#desktop\\_end\\_user\\_computing/vmware\\_workstation\\_player/14\\_0](https://my.vmware.com/en/web/vmware/free#desktop_end_user_computing/vmware_workstation_player/14_0))  
**7-zip** (<http://www.7-zip.org>)

# HomeLab WSN: DebTos VM

## Why?

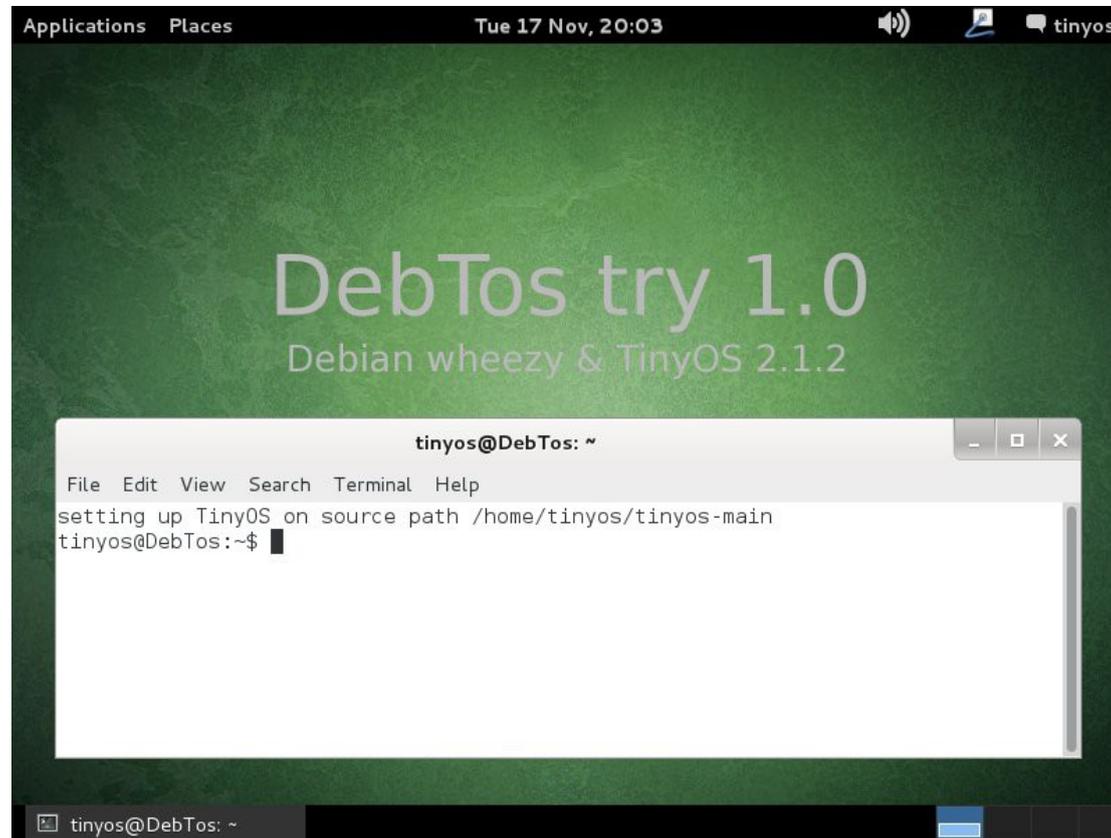
- Sometimes you can spend more time setting up the working environment than time to make a real experience with WSNs
- Everything is an exercise, but...
- ... we are trying to made the setup easier as possible, so we have prepared a **Debian 7** (Wheezy) with a working **TinyOS 2.1.2** (“still” latest version up to date), code highlight...

## Note

If you use a Windows OS as host machine, you need however to install MIB520 drivers:  
<http://www.ftdichip.com/Drivers/VCP.htm>

and install it while connecting the MIB520 to an USB port. At the end of the procedure they will be created two virtual COM ports.

# HomeLab WSN: DebTos VM



*Note:*

*Username: tinyos, Password: tinyos*

# HomeLab WSN: Lessons' summary

The following material is a free adaptation of the “Working Group Tutorial” (note: some figures are currently missing in the web pages):

[http://tinyos.stanford.edu/tinyos-wiki/index.php/TinyOS\\_Tutorials](http://tinyos.stanford.edu/tinyos-wiki/index.php/TinyOS_Tutorials)

- Lesson 1.1: Getting started with TinyOS
- Lesson 1.2: Modules and the TinyOS Execution Model
- Lesson 1.3: Mote-mote radio communication
- Lesson 1.4: Mote-PC serial communication and SerialForwarder
- Lesson 1.5: Sensing
- Lesson 1.5.1: ADC
- Lesson 1.6 Boot Sequence

- Lesson 1.7 Storage
- Lesson 1.8 Resource Arbitration and Power Management
- Lesson 1.9 Concurrency
- Lesson 1.11 TOSSIM

- Lesson 1.10 Platforms
- Lesson 1.12 Network Protocols
- Lesson 1.13 TinyOS Toolchain
- Lesson 1.14 Building a simple but full-featured application
- Lesson 1.15 The TinyOS printf Library
- Lesson 1.16 Writing Low-Power Applications
- Lesson 1.17 TOSThreads Tutorial
- Lesson 3.2 Rssi Demo

**Mandatory**

**Strongly recommended**

**Some lessons at your choice**

# Lesson 1.1: Getting started with TinyOS

[\(web link\)](#)

Since we are using a VM with a pre-installed TinyOS environment much of the lesson could be avoided. So:

- physically connect the MIB520 to an USB port (some problems could occur with USB 3.0)
- logically connect the MIB520 to the VM
- by means of a shell go to the folder `/home/tinyos/tinyos-main/apps/Blink`

Since we are using Iris nodes we have:

- `make iris` to compile
- `make iris install mib510,/dev/ttyUSB0` to compile and program the node
- `make iris reinstall mib510,/dev/ttyUSB0` to program the node without compiling
- using the notation `install.ID` or `reinstall.ID` (ID is a 16 bit number without sign so from 0 to 65535) it is possible to assign an ID to each node when programming it

(Please, let you note that we need to use mib510 also if we are actually using a mib520 programming board.)

# Lesson 1.1: Getting started with TinyOS

(web link)

Then:

- *make iris*
- *make iris reinstall mib510,/dev/ttyUSB0* (ID is not needed for Blink)

and the node leds should blink like described in the tutorial.

Now, jump to:

[http://tinyos.stanford.edu/tinyos-wiki/index.php/Getting\\_Started\\_with\\_TinyOS](http://tinyos.stanford.edu/tinyos-wiki/index.php/Getting_Started_with_TinyOS)

and start directly from “4 - Components and Interfaces” to the end of the lesson.

## Note

*/dev/ttyUSB0* and */dev/ttyUSB1* (used later) could appear with a different numbering on your system depending on what USB peripherals you have connected to the VM. In such a case, you can use the *dmesg* command after connected the programming board.

Q: what happened in your Host OS when you plug in the MIB520?

# Lesson 1.2: Modules and the TinyOS Exec. Model

(web link)

Follow the whole lesson:

[http://tinyos.stanford.edu/tinyos-wiki/index.php/Modules\\_and\\_the\\_TinyOS\\_Execution\\_Model](http://tinyos.stanford.edu/tinyos-wiki/index.php/Modules_and_the_TinyOS_Execution_Model)

Let's go...

# Lesson 1.3: Mote-mote radio communication

(web link)

Follow the whole lesson:

[http://tinyos.stanford.edu/tinyos-wiki/index.php/Mote-mote\\_radio\\_communication](http://tinyos.stanford.edu/tinyos-wiki/index.php/Mote-mote_radio_communication)

Let's go...

# Lesson 1.4: Mote-PC serial communication and SerialForwarder

([web link](#))

Follow the lesson:

[http://tinyos.stanford.edu/tinyos-wiki/index.php/Mote-PC\\_serial\\_communication\\_and\\_SerialForwarder](http://tinyos.stanford.edu/tinyos-wiki/index.php/Mote-PC_serial_communication_and_SerialForwarder)

from the beginning to "Sending a packet to the serial port in TinyOS"; have a look at the remaining part of the lesson if you have troubles with Java and see the note in the following slide

## Note

The right parameter to be used in this HomeLab is:

- `comm serial@/dev/ttyUSB1:iris` (or 57600 instead of Iris).

## Note

Using "&" after shell commands, these will be executed in background avoiding to block the shell.

Q: what happens on /dev/ttyUSB1 or COMx?? Have a look... ([Tep 113](#))

# Lesson 1.4: Mote-PC serial communication and SerialForwarder

([web link](#))

Note 11/12/2015: SOLVED in ver.4.1

If you get some errors compiling apps that involve java, try the following:

- Open a terminal and type ***echo \$CLASSPATH***
  - If you see only:  
*:/home/tinyos/tinyos-main/support/sdk/java*
- You have to replace the following line in */home/tinyos/tinyos-main/tinyos.env* (if you don't see the file, within document explorer, set View->Show Hidden Files)

```
export CLASSPATH=$CLASSPATH:$TOSROOT/support/sdk/java
```

with the following:

```
export CLASSPATH=$CLASSPATH:$TOSROOT/support/sdk/java:$TOSROOT/support/sdk/java/tinyos.jar:
```

- Close all terminals and open again

# Lesson 1.5: Sensing

([web link](#))

Follow the whole lesson:

<http://tinyos.stanford.edu/tinyos-wiki/index.php/Sensing>

Then, to finish this lesson, let you try to modify the proper *DemoSensor* component to change the sensor type (from voltage to light and then to temp).

## Note **VERIFY THE PATHS...**

If we compile with *make iris*, we are not specifying any sensor board and so the compiler will use, if needed, the *DemoSensor* component in *opt/tinyos-2.1.0/tos/platforms/micaz* (this a common component for *micaz* and *iris*) that will use the batteries voltage as sensor (the only one available without a sensor board!!!).

In order to use a real sensor board it is needed to specify this, for example, by means of the *SENSORBOARD* environment variable and by compiling in this way (we could also modify the Makefile in the app folder to make such an option the default one):

- If you have a MDA100CB sensor board
  - *SENSORBOARD=mda100 make iris*
- If you have a BASICSB sensor board
  - *SENSORBOARD=basicsb make iris*

## Note **VERIFY THE PATHS...**

If you are using a MDA100CB, it is needed another modification to the VM. You have to copy the file *TempImpIP.nc* from *opt/tinyos-2.1.0/tos/sensorboards/mda100/cb* to *opt/tinyos-2.1.0/tos/sensorboards/mda100*.