Installing and Running Contiki

1st Workshop on Appl. of WSN for Env. Monitoring in Developing Countries

Trieste (Italy), 02.03.2011

Installing and Running Contiki

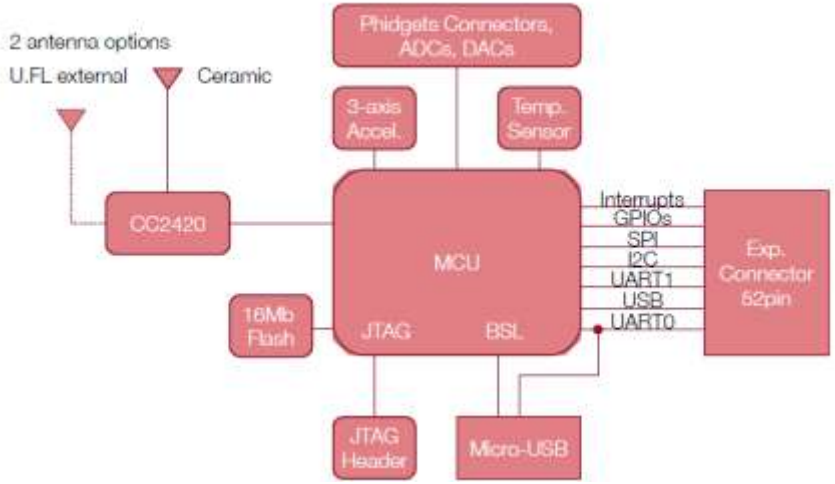
- **Outline**
  - Distributing sensor nodes
  - Installing Contiki
  - Contiki structure
  - Application files and makefile
  - Compiling an application
  - Running an application

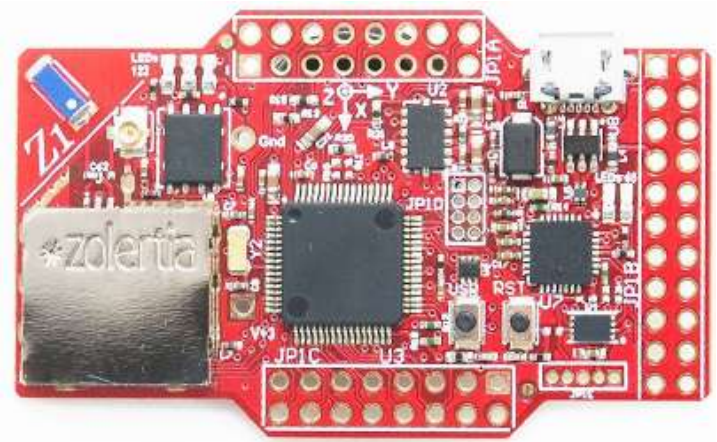
Installing and Running Contiki

- **Outline**
  - Distributing sensor nodes
  - Installing Contiki
  - Contiki structure
  - Application files and makefile
  - Compiling an application
  - Running an application

The Zolertia Z1 Platform

- **2nd Generation MSP430**
  - Low power 16-bit MCU 16 MHz
- **CC2420 Radio Transceiver**
  - 2.4 GHz IEEE 802.15.4 compliant
- **ADXL345 Digital Accelerometer**
  - 3-axis, ±2/4/8/16g
- **Low-power Temperature Sensor**
  - ±0.5°C accuracy, [-25,85]°C range





Installing and Running Contiki

Outline

- Distributing sensor nodes
- Installing Contiki
- Contiki structure
- Application files and makefile
- Compiling an application
- Running an application

Installing Contiki

- **Many Options to Install Contiki**
  - http://www.sics.se/contiki/download.html
  - Download stable version (contiki-2.4)
  - Download current development progress (contiki-2.x)
  - Download a virtual machine (Instant Contiki 2.4)

Installing Contiki

- **We will Install Contiki manually (no virtual machine)**
  - Quite simple!
    1. Download and install the compiler for the MSP430
    2. Download the Contiki tree
    3. Compile & Run an application!

Installing Contiki

➢ Installing the latest MSP430 compiler

- Open your browser and type the following address
  http://ftp.openosian.net/mspgcc4
- Download the three .deb files

Index of ftp://ftp.openosian.net/mspgcc4/

Up to higher level directory

Name	Size	Last Modified
File: mspgcc4-binutils_2.20.1-20100815_i386.deb	6519 KB	08/15/2010 12:00:00 AM
File: mspgcc4-gcc_4.4.4-20100815_i386.deb	10737 KB	08/15/2010 12:00:00 AM
File: mspgcc4-libc_20100815-20100815_i386.deb	693 KB	08/15/2010 12:00:00 AM
OLD		08/15/2010 12:00:00 AM

Installing Contiki

➢ Installing the latest MSP430 compiler

- Install the mspgcc4-binutils (double-click on it)
- Install the mspgcc4-gcc by typing into the terminal

  sudo dpkg --force-overwrite -i mspgcc4-gcc_4.4.4-20100815_i386.deb

```
marconipc-2@marconipc-2:~/Desktop$ sudo dpkg --force-overwrite -i mspgcc4-gcc_4.
4.4-20100815_i386.deb
(Reading database ... 152037 files and directories currently installed.)
Unpacking mspgcc4-gcc (from mspgcc4-gcc_4.4.4-20100815_i386.deb) ...
dpkg: warning: overriding problem because --force enabled:
 trying to overwrite '/lib/libiberty.a', which is also in package mspgcc4-binuti
ls 0:2.20.1-20100815
Setting up mspgcc4-gcc (4.4.4-20100815) ...
Processing triggers for man-db ...
marconipc-2@marconipc-2:~/Desktop$
```

- Install the mspgcc4-libc (double-click on it)

Installing Contiki

- **Download the Latest Contiki Tree**
  - https://github.com/enricmcalvo/contiki-2.x
  - Unzip the file and put it in the home folder
  - Rename the unzipped folder as contiki-2.x
  - Open a terminal and go to the /home/contiki-2.x folder

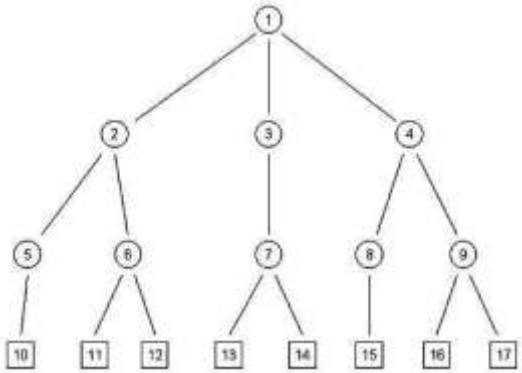
Installing and Running Contiki

Outline

- Distributing sensor nodes
- Installing Contiki
- Contiki structure
- Application files and makefile
- Compiling an application
- Running an application

Contiki Structure

- **Contiki folders organization**
  - **apps**
    - Architecture-independent applications
  - **core**
    - System source code
  - **cpu**
    - Microcontroller-specific code (one subdirectory for each CPU)
  - **examples**
  - **platform**
    - Specific board drivers (one subdirectory per platform)
  - **tools**
    - Bootstrap loaders, COOJA, MSPsim, serialdump, etc.



Contiki Structure

- **Contiki folders organization**
  - **core**
    - Contiki's file system
    - Basic source code for sensors, networking, timers, ….
  - **cpu**
    - Microcontroller-specific code (example: MSP430)
  - **platform**
    - Specific board drivers (example: Tmote Sky, Zolertia z1)
  - **tools**
    - Bootstrap loaders, COOJA, MSPsim, serialdump, etc.

Tip: Have a look at the folders, get familiar with their content!

Installing and Running Contiki

Outline

- Distributing sensor nodes
- Installing Contiki
- Contiki structure
- Application files and makefile
- Compiling an application
- Running an application

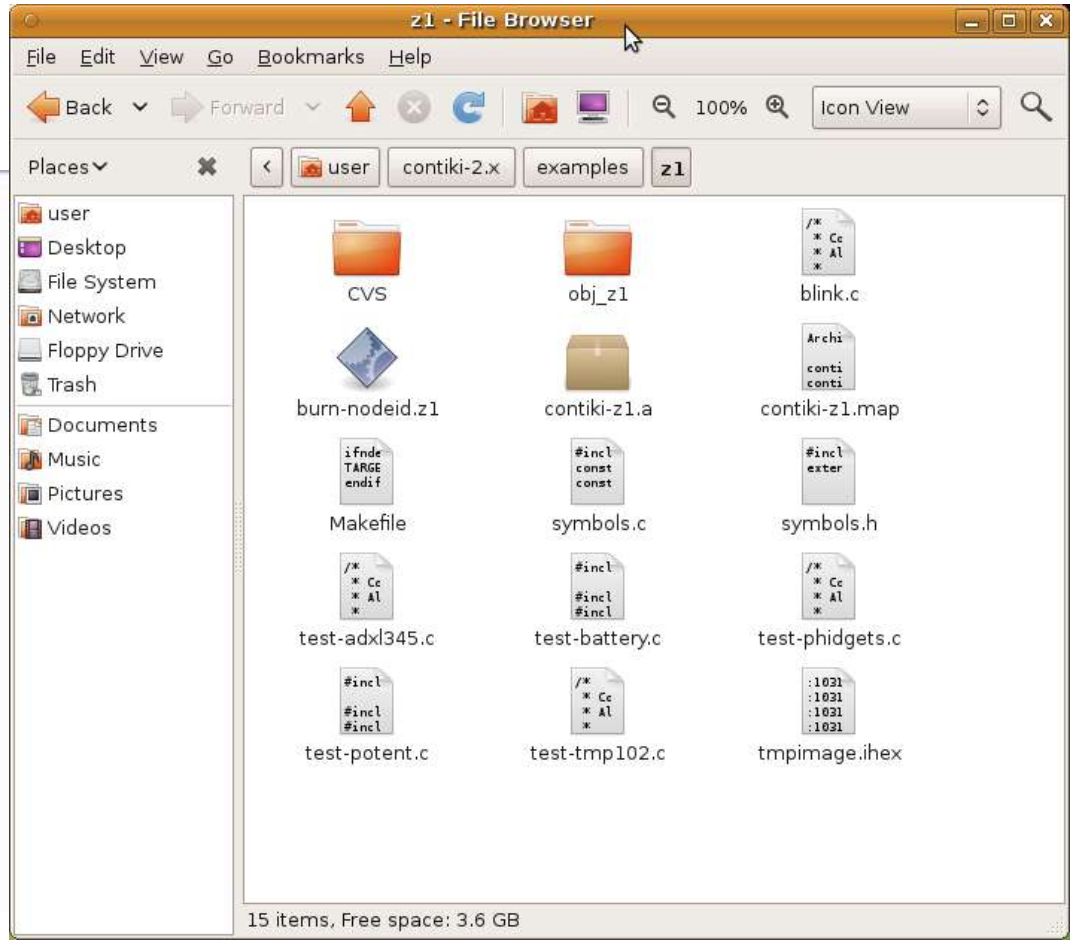
Application Files and Makefile

- **How Does a Typical Application Looks Like?**
  - Browse the example directory to get a flavour
  - Typically you have a folder for your application
    - Makefile
    - Main C file
    - Additional libraries
    - [compiled files]

Application Files and Makefile

- **Examples**

Name	Type
obj_jcreate	File folder
assignment1_boano.c	C File
assignment1_boano.jcreate	JCREATE File
contiki-jcreate.a	A File
contiki-jcreate.map	MAP File
Makefile	File
sentilla_leds.c	C File
sentilla_leds.h	H File
symbols.c	C File
symbols.h	H File
tmpimage.ihex	IHEX File



Application Files and Makefile

➢ Makefile

- Allows us to compile our application from source code

```
# Give a name to your project
CONTIKI_PROJECT = project_name
all: $(CONTIKI_PROJECT)
# Define the target platform (Sentilla JCreate, Zolertia Z1, Tmote Sky)
ifndef TARGET
TARGET=jcreate   ---   TARGET=z1   ---   TARGET=sky
endif
# Define additional source files to be compiled (if any)
CONTIKI_TARGET_SOURCEFILES += additional_file.c
# Root of the Contiki tree
CONTIKI = ../../..
# Including the global Contiki makefile
include $(CONTIKI)/Makefile.include
```

Installing and Running Contiki

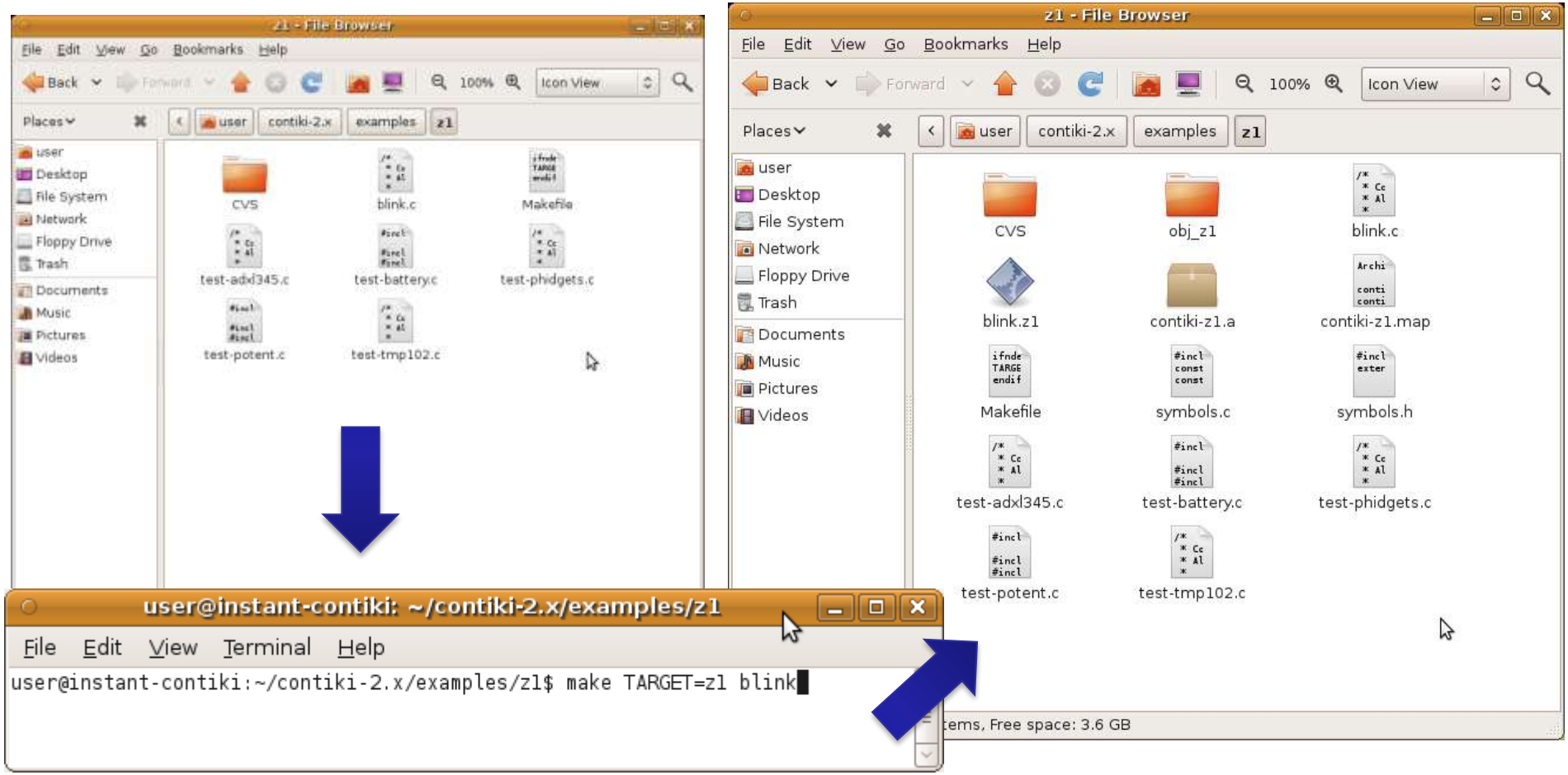
Outline

- Distributing sensor nodes
- Installing Contiki
- Contiki structure
- Application files and makefile
- Compiling an application
- Running an application

Compiling an Application

- **Three Things Needed**
  - An application to be compiled
  - All libraries included
  - A Makefile
- **make clean**
  - Cleans previously compiled code
- **make TARGET=platform application**
  - Or write those information in the makefile and just type make

Compiling an Application



Installing and Running Contiki

- **Outline**
  - Distributing sensor nodes
  - Installing Contiki
  - Contiki structure
  - Application files and makefile
  - Compiling an application
  - Running an application

Running an Application

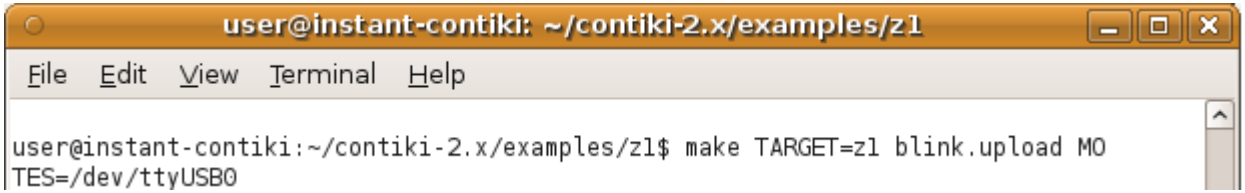
- **Checking connected devices**
  - make platform-motelist (Example: make z1-motelist)

```
user@instant-contiki: ~/contiki-2.x/examples/z1
File  Edit  View  Terminal  Help
user@instant-contiki:~/contiki-2.x/examples/z1$ make z1-motelist
../../tools/z1/motelist-z1
Reference  Device          Description
---------- --------------- ------------------------------------------
Z1RC0336   /dev/ttyUSB0    Silicon Labs Zolertia Z1
user@instant-contiki:~/contiki-2.x/examples/z1$
```

Running an Application

➢ Uploading the code

- make TARGET=platform application.upload [MOTES=XXX]
- MOTES=comX under Cygwin/Windows
- MOTES=/dev/ttyUSBX under Unix
- If MOTES is not specified all connected motes will be programmed



Running an Application

- **Uploading the code**
  - Reset device
  - Mass erase
  - Invoking BSL
  - Program
  - Reset

user@instant-contiki: ~/contiki-2.x/examples/z1

File Edit View Terminal Help

```
user@instant-contiki:~/contiki-2.x/examples/z1$ make TARGET=z1 blink.upload MO
TES=/dev/ttyUSB0
.
                                        ■■■
Reset device ...
Done
make[2]: Leaving directory `/home/user/contiki-2.x/examples/z1'
make -j 20 z1-upload-sequence
make[2]: Entering directory `/home/user/contiki-2.x/examples/z1'
+++++ Erasing /dev/ttyUSB0
MSP430 Bootstrap Loader Version: 1.39-goodfet-8
Use -h for help
Use --fromweb to upgrade a GoodFET.
Mass Erase...
Transmit default password ...
+++++ Programming /dev/ttyUSB0
MSP430 Bootstrap Loader Version: 1.39-goodfet-8
Invoking BSL...
Transmit default password ...
Current bootstrap loader version: 2.13 (Device ID: f26f)
Changing baudrate to 38400 ...
Program ...
22398 bytes programmed.
+++++ Resetting /dev/ttyUSB0
MSP430 Bootstrap Loader Version: 1.39-goodfet-8
Use -h for help
Use --fromweb to upgrade a GoodFET.
Reset device ...
Done
make[2]: Leaving directory `/home/user/contiki-2.x/examples/z1'
make[1]: Leaving directory `/home/user/contiki-2.x/examples/z1'
rm blink.co blink.ihex
user@instant-contiki:~/contiki-2.x/examples/z1$
```

Running an Application

- **Useful commands**
  - make z1-reset
  - make z1-motes
  - make z1-motelist
  - make login

Resetting the device, like the reset button

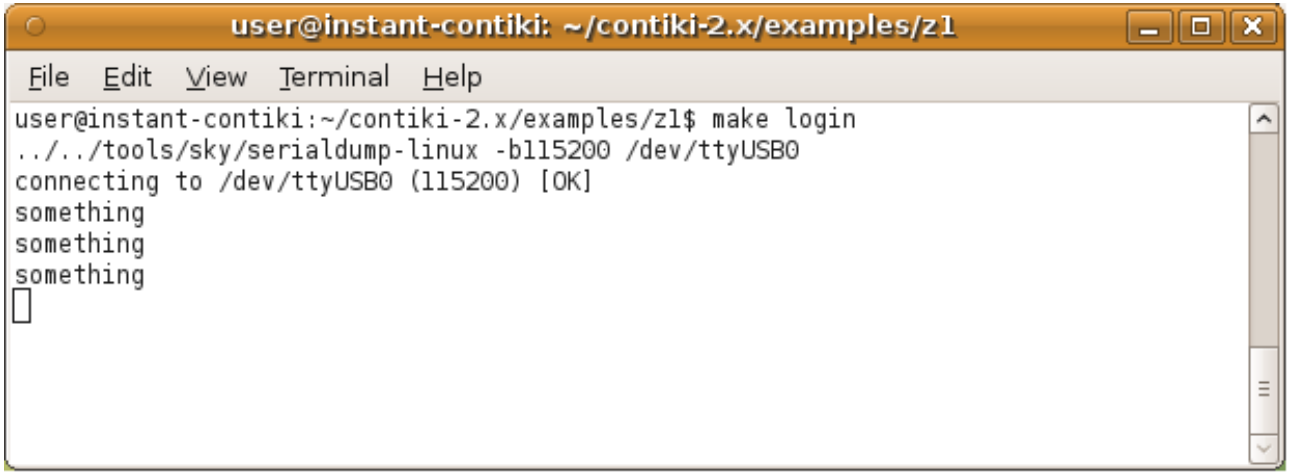
```
user@instant-contiki: ~/contiki-2.x/examples/z1
File  Edit  View  Terminal  Help

user@instant-contiki:~/contiki-2.x/examples/z1$ make z1-reset
make -k -j 20 z1-reset-sequence
make[1]: Entering directory `/home/user/contiki-2.x/examples/z1'
../../tools/z1/z1-bsl-nopic --z1 -c /dev/ttyUSB0 -r
MSP430 Bootstrap Loader Version: 1.39-goodfet-8
Use -h for help
Use --fromweb to upgrade a GoodFET.
Reset device ...
Done
make[1]: Leaving directory `/home/user/contiki-2.x/examples/z1'
user@instant-contiki:~/contiki-2.x/examples/z1$
```

Running an Application

- **Useful commands**
  - make z1-reset
  - make z1-motes
  - make z1-motelist
  - make login

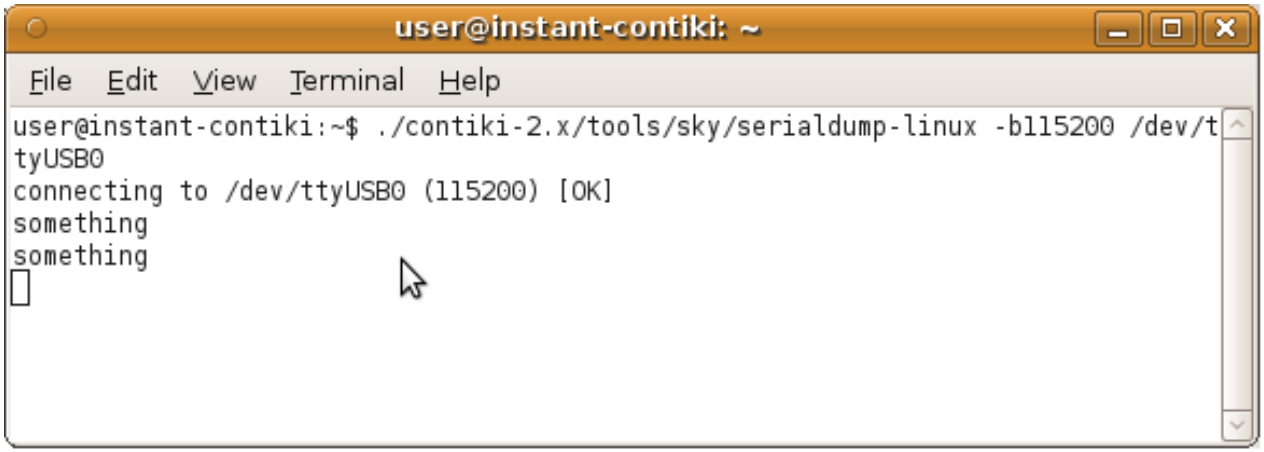
Connecting to the mote!
Through the serial interface



Running an Application

- **Useful commands**
  - make z1-reset
  - make z1-motes
  - make z1-motelist
  - make login

Connecting to the mote!
Exploiting the serialdump (tools/…)



Running an Application

➢ Burning the node-id

- make z1-motelist

```
user@instant-contiki: ~/contiki-2.x/examples/z1
File  Edit  View  Terminal  Help
user@instant-contiki:~/contiki-2.x/examples/z1$ make z1-motelist
../../tools/z1/motelist-z1
Reference  Device           Description
---------- ---------------- ---------------------------------------------
Z1RC0336   /dev/ttyUSB0     Silicon Labs Zolertia Z1
user@instant-contiki:~/contiki-2.x/examples/z1$
```

- Take the reference ID and type:

make clean

make burn-nodeid.upload nodeid=XXX nodemac=XXX && make login

```
user@instant-contiki: ~/contiki-2.x/examples/z1
File  Edit  View  Terminal  Help
user@instant-contiki:~/contiki-2.x/examples/z1$ make z1-motelist
../../tools/z1/motelist-z1
Reference  Device           Description
---------- ---------------- ---------------------------------------------
Z1RC0336   /dev/ttyUSB0     Silicon Labs Zolertia Z1
user@instant-contiki:~/contiki-2.x/examples/z1$ make burn-nodeid.upload nodeid=336 no
demac=336 && make login
```

Running an Application

- **Summary of steps**
  1. Create a folder

Running an Application

- **Summary of steps**
  1. Create a folder
  2. Create C file and makefile (or paste existing ones)

Running an Application

- **Summary of steps**
  1. Create a folder
  2. Create C file and makefile (or paste existing ones)
  3. Edit the makefile and set the names accordingly
     - TARGET plaftorm
     - Project name as the name of the C file
     - Upload

Running an Application

- **Summary of steps**
  1. Create a folder
  2. Create C file and makefile (or paste existing ones)
  3. Edit the makefile and set the names accordingly
     - TARGET plaftorm
     - Project name as the name of the C file
     - Upload
  4. Edit the C file

Running an Application

- **Summary of steps**
  1. Create a folder
  2. Create C file and makefile (or paste existing ones)
  3. Edit the makefile and set the names accordingly
     - TARGET plaftorm
     - Project name as the name of the C file
     - Upload
  4. Edit the C file
  5. Compile and upload!

Running an Application

- **Summary of steps**

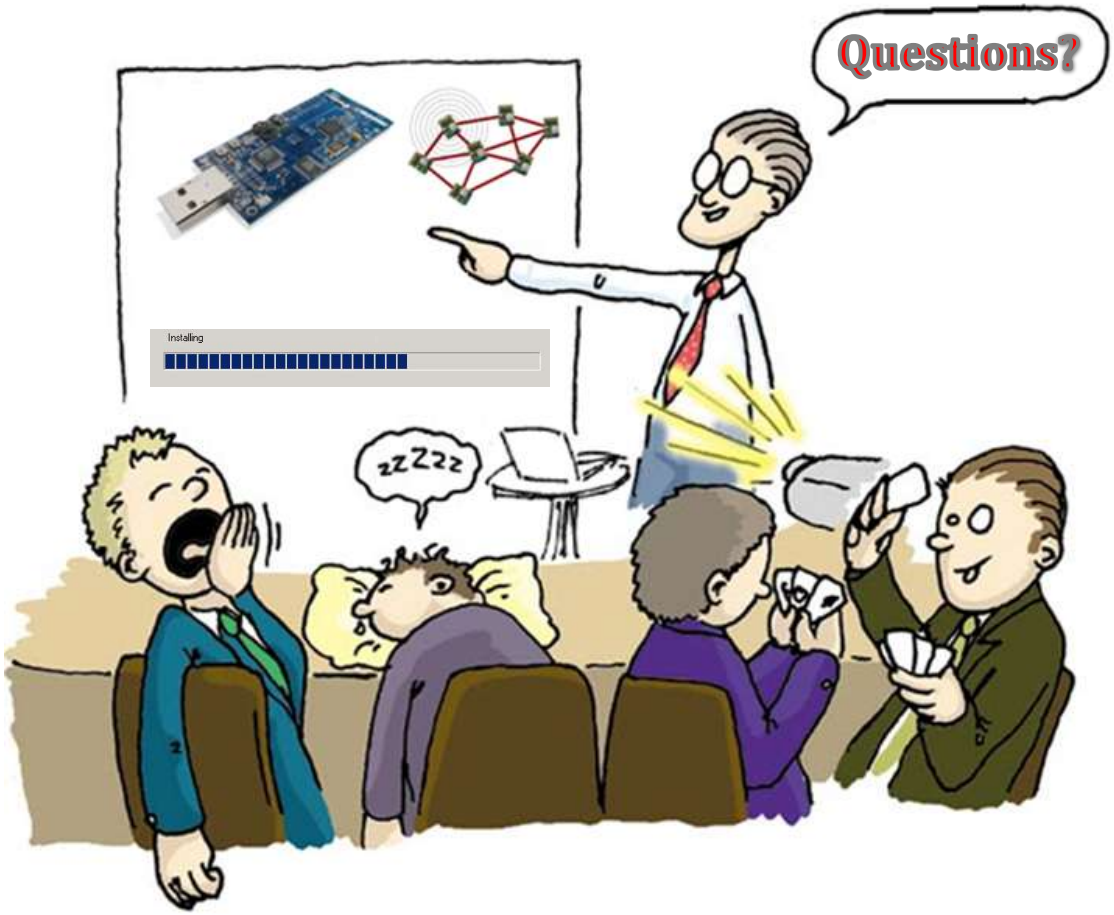
1. Create a folder
2. Create C file and makefile (or paste existing ones)
3. Edit the makefile and set the names accordingly
   - TARGET plaftorm
   - Project name as the name of the C file
   - Upload
4. Edit the C file
5. Compile and upload!
6. Connect to the serial port with make z1-reset && make login

Running an Application

- **What's next?**
  - Run the hello-world program
  - Write and run the hello-world-reloaded program
  - Experimenting with LEDs
  - Experimenting with timers
  - Experimenting with sensors
  - Have fun!

References

- **Z1 Wiki**
  - http://zolertia.sourceforge.net/wiki/index.php/Main_Page
  - http://zolertia.sourceforge.net/wiki/index.php/Mainpage:Contiki_Lesson_0

- **Contiki Tutorials**
  - http://www.sics.se/contiki/tutorials/

Questions & Answers