



Wireless Sensor Networks

Exercise 1



Before starting

- <http://vecchio.iet.unipi.it/vecchio/didattica/maps/slides/>
 - Password: studente
- TinyOS environment
 - Install TinyOS on your notebook/PC
 - Xubuntos VM
- Installing TinyOS
 - Windows: it requires Cygwin
 - Linux: rpm or deb based
 - http://docs.tinyos.net/tinywiki/index.php/Getting_started
- Xubuntos
 - Xubuntu Linux with pre-installed TinyOS
 - C:\vecchio in Lab PCs
 - Username: xubuntos Password: tinyos



- /opt/tinyos-2.x
 - apps
 - tos
 - support
 - doc
- To compile:
 - cd into app (e.g. Blink) directory
 - *make <platform>*
 - Our platform is *telosb*
- To compile and (re)install: *make <platform> (re)install*
- To produce documentation: *make platform docs*
- Before installing: *motelist*
 - Use VM menu to attach to guest OS



Before starting

- Description of API (html documentation or source code in tos)

```
Terminal - xubuntos@xubuntos-tinyos: /opt/tinyos-2.1.1/tos/interfaces
File Edit View Terminal Go Help
/**
 * Commands for controlling three LEDs. A platform can provide this
 * interface if it has more than or fewer than three LEDs. In the
 * former case, these commands refer to the first three LEDs. In the
 * latter case, some of the commands are null operations, and the set
 * of non-null operations must be contiguous and start at Led0. That
 * is, on platforms with 2 LEDs, LED 3's commands are null operations,
 * while on platforms with 1 LED, LED 2 and LED 3's commands are null
 * operations.
 *
 * @author Joe Polastre
 * @author Philip Levis
 */

#include "Leds.h"

interface Leds {

/**
 * Turn on LED 0. The color of this LED
 */
  async command void led00n();

/**
```

The screenshot shows a Mozilla Firefox browser window displaying the HTML documentation for the 'Interface: tos.interfaces.Leds'. The browser's address bar shows the file path: file:///opt/tinyos-2.1.0/doc/nescdoc/telosb/index.html. The page content includes a navigation sidebar on the left with links to various components like 'Everything', 'Packages', 'Alarm', 'Boot', 'Counter', 'GeneralIO', 'HplMsp430GeneralIO', 'Init', 'Leds', 'LocalTime', 'McuPowerOverride', 'McuPowerState', 'McuSleep', 'Msp430Capture', 'Msp430ClockInit', 'Msp430Compare', 'Msp430Timer', 'Msp430TimerControl', 'Msp430TimerEvent', 'Scheduler', 'TaskBasic', and 'Timer'. The main content area is titled 'Interface: tos.interfaces.Leds' and contains the following information:

- interface Leds**
- Constants for manipulating LEDs.
- Author:** Philip Levis
- Date:** March 21, 2005
- Commands**
- command uint8_t [get\(\)](#)
Get the current LED settings as a bitmask.
- command void [led00f\(\)](#)
Turn off LED 0.
- command void [led00n\(\)](#)
Turn on LED 0.
- command void [led0Toggle\(\)](#)
Toggle LED 0; if it was off, turn it on, if was on, turn it off.



- Integer types (size of int is platform dependent)

	8 bits	16 bits	32 bits	64 bits
signed	int8_t	int16_t	int32_t	int64_t
unsigned	uint8_t	uint16_t	uint32_t	uint64_t

- error_t: possible values SUCCESS, FAIL
- Variables can be defined outside functions

```
...  
implementation  
{  
  uint8_t mycounter = 0;  
  
  event void AnInterface.anEvent()  
  {  
    mycounter++;  
  }  
...  
...
```



```
• interface Leds {  
    async command void led0On();  
    async command void led0Off();  
    async command void led0Toggle();  
    async command void led1On();  
    async command void led1Off();  
    async command void led1Toggle();  
    async command void led2On();  
    async command void led2Off();  
    async command void led2Toggle();  
    async command uint8_t get();  
    async command void set(uint8_t val);  
}
```



```
interface Timer<precision_tag>
{
    command void startPeriodic(uint32_t dt);
    command void startOneShot(uint32_t dt);
    command void stop();
    event void fired();
    ...
}
```

- Precision tag: specifies timer resolution
 - TMilli
 - TMicro
 - T32khz



- Each sensor node is equipped with three leds: 0, 1, and 2.
- Write an application that ciclically turns the three leds on/off according to the following scheme:

$\textcircled{0} 1 2 \rightarrow 0 \textcircled{1} 2 \rightarrow 0 1 \textcircled{2} \rightarrow 0 \textcircled{1} 2 \rightarrow \textcircled{0} 1 2 \rightarrow 0 \textcircled{1} 2 \rightarrow \dots$

where $\textcircled{0}$ denotes the led that is turned on (while the others are off).

- Wait 0.2 second between a configuration and the next one.
- Make a copy of the Blink directory and modify it.