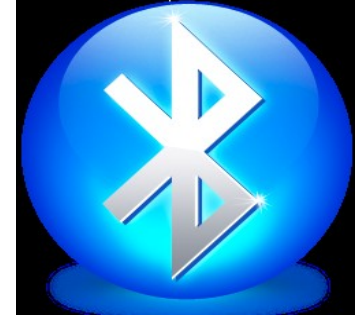


SHORT RANGE WIRELESS COMMUNICATION AND TINY OS PROGRAMMING



SHORT RANGE WIRELESS COMMUNICATIONS

- Wireless communication is major feature for just about every new electronic product
- Flexible, convenient, and enables remote monitoring and control without expensive wiring and cabling
- Day to day applications like Bluetooth, Wi-fi, setting up ad-hoc network, etc.

SHORT RANGE WIRELESS COMMUNICATION

- IEEE 802.11 standard which specifies the physical layer and media access control for implementing wireless local area network (WLAN/Wi-Fi) computer communication
- Operates on the 2.4 GHz and 5 GHz Industrial, Science and Medical (ISM) frequency bands
- IEEE 802.15.1 is basis for Bluetooth technology, and operates with 3 classes of devices based on range

SHORT RANGE WIRELESS COMMUNICATION

- IEEE 802.15.4 is a standard for low-rate wireless personal area networks
- Emphasizes low cost communication of nearby devices with little to no underlying infrastructure, intending to exploit this to lower power consumption even more
- Devices include power management functions such as link quality and energy detection
- It is the basis for the ZigBee, ISA100.11a, WirelessHART, and MiWi specifications

ZIGBEE

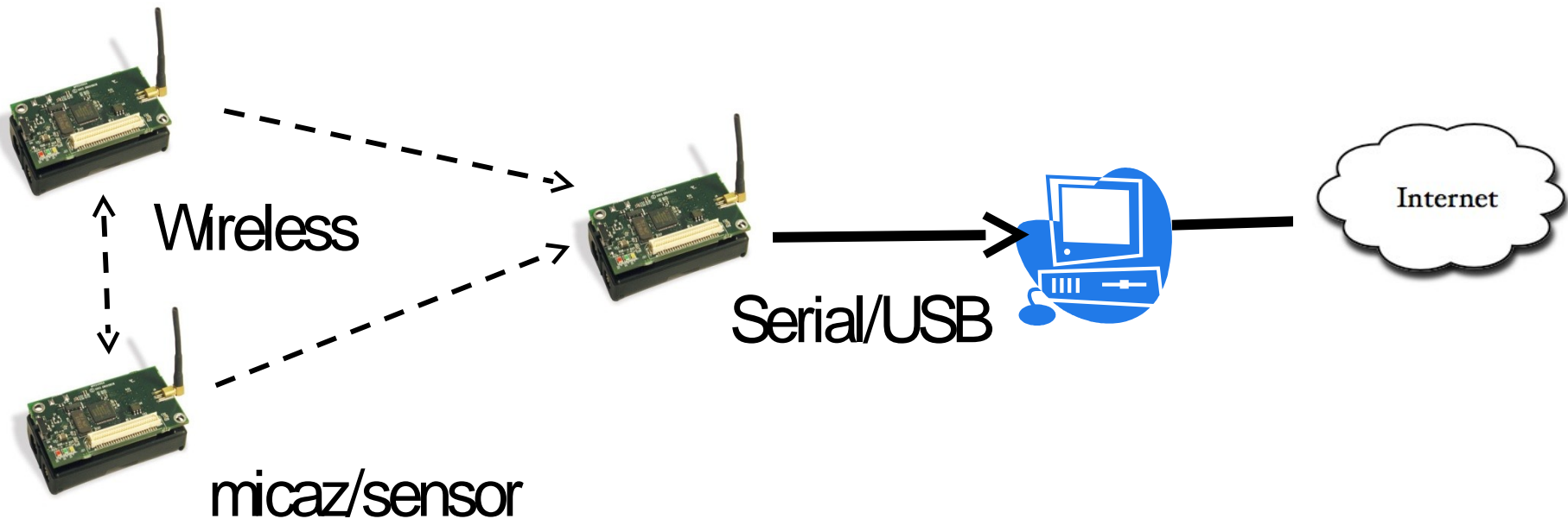
- Technology to create personal area networks built from small, low-power digital radios
- Designed for low-power consumption allowing batteries to last for a very long time due to sleep mode specifically designed to accommodate battery powered devices
- ZigBee features include:
 - Support for multiple network topologies such as point-to-point, point-to-multipoint and mesh networks
 - Low duty cycle and low average power consumption – provides long battery life
 - Low latency

OVERVIEW

Sensor code
(nesC/TinyOS)

Base station code
(nesC/TinyOS)

Gateway code
(Java, c, ...)



WHAT IS TINYOS?

An operating system for low power, embedded, wireless devices

- Wireless sensor networks (WSNs)
- Sensor-actuator networks
- Embedded robotics

Open source, open developer community

- <http://www.tinyos.net>
- *E-book: TinyOS Programming:*
<http://csl.stanford.edu/~pal/pubs/tinyos-programming.pdf>

NESC

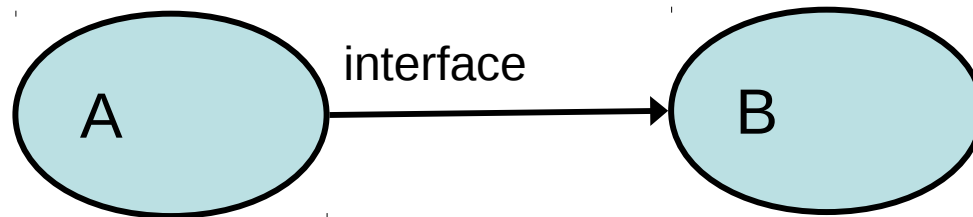
TinyOS and its applications are in nesC

- C dialect with extra features

Basic unit of nesC code is a component

Components connect via interfaces

- Connections called “wiring”



A SIMPLE TINYOS PROGRAM

In TinyOS, for every program, you need to create three files. Suppose the program we create is called 'Simple':

- Configuration File
- Component file
- Makefile

CONFIGURATION FILE

```
configuration SimpleAppC
{
}

Implementation
{
components SimpleC, MainC;
SimpleC.Boot -> MainC.Boot;
}
```

CONFIGURATION FILE

There are two components in this program: your component called SimpleC and the Main component MainC.

The MainC component provides the **Boot.booted** signal which essentially is the entry point of the application

COMPONENT FILE

```
module SimpleC
{
uses interface Boot;
}
implementation{
event void Boot.booted()
{
    //The entry point of the program
}}}
```

MAKEFILE

```
COMPONENT=SimpleAppC
```

```
include $(MAKERULES)
```

** the name of the top level configuration in the COMPONENT field.

AVRORA

Avrora is an open-source cycle-accurate **simulator** for embedded sensing programs.

It can emulate two typical platforms, Mica2 and MicaZ, and run AVR elf-binary or assembly codes for both platforms.

TinyOS + AVRORA Demo

INSTALLATION GUIDE

STEP 1 (VERY IMPORTANT): Check Java Version – if it is Open JDK, remove and install JDK 6 or 7 because Open JDK is not supported by Avrora.
DO NOT CHANGE JDK VERSION AFTER INSTALLING TINYOS.

STEP 2: Use this link to install Avrora :

<http://mythicalcomputer.blogspot.in/2008/09/running-tinyos-programs-using-avrora.html>

STEP 3: Install TinyOS using :

http://tinyos.stanford.edu/tinyos-wiki/index.php/Automatic_installation

Possible errors you may encounter:

Ncc command not found – In this case, install tiny-tools (one of the first 2 links, based on whether you have 32 or 64 bit systems, from this link :

<http://tinyos.stanford.edu/tinyos/dists/ubuntu/pool/main/t/tinyos-tools/>)

Python config error – Look in the file tinyos-main/support/make/sim.extra for the line `PYTHON_VERSION = ...` and enter your system's python version. (Eg: If your python version is 2.6.1, enter `PYTHON_VERSION=2.6`)

Main.class not found error – check if all the PATH variables have been set correctly, as well as `JAVA_HOME` (check using `echo $'PATH_NAME'`)