Main TinyOS Directory Structure

• Files should have been installed under:
  – /opt/tinyos-2.x/
  – C:\cygwin\opt\tinyos-2.x\n
• Contains:
  – apps/
  – support/
  – tos/
“tos” Directory

- chips/
  - nc code for various functional blocks of different ICs: microcontrollers, radios, flash memory, etc.

- interfaces/
  - interfaces for core TOS: Active Message, Boot, Init, Random, BitVector, StdControl

- lib/
  - more complicated devices, protocol-level support: deluge, timer, serial

- platforms/
  - sensor-platform specific code

- sensorboards/

- system/
  - implementation of remaining interfaces

- types/
  - typedefs in .h files
“chips” Directory

- at45db: Atmel 512K Data Flash Memory
- atm128: ATmega128 microcontroller, used in mica family, Atlas
- cc1000: Chipcon radio
- cc2420: Chipcon 802.15.4 radio
- pxa27x: Intel 32-bit XScale processor
- msp430: TI microcontroller in Telos family
“atm128” Directory

- .h for general hardware
- adc
- i2c
- pins
- spi
- timer
- uart
- sim
“tos” Directory

• chips/
  - nc code for various functional blocks of different ICs: microcontrollers, radios, flash memory, etc.

• interfaces/
  - interfaces for core TOS: Active Message, Boot, Init, Random, BitVector, StdControl

• lib/
  - more complicated devices, protocol-level support: deluge, timer, serial

• platforms/
  - sensor-platform specific code

• sensorboards/

• system/
  - implementation of remaining interfaces

• types/
  - typedefs in .h files
“platforms” Directory

• Several predefined platforms part of TOS:
  – mica/mica2/mica2dot/micaz
  – telosa/telosb

• We will be adding another:
  – atlas

• Contains platform-specific configurations/implementations

• hardware.h
configuration PlatformSerialC {
    provides interface Init;
    provides interface StdControl;
    provides interface SerialByteComm;
}
implementation {
    components HplAtm128UartC as UART, PlatformC;

    SerialByteComm = UART.Uart0;
    Init = UART.Uart0Init;
    StdControl = UART.Uart0RxControl;
    StdControl = UART.Uart0TxControl;
}
Main TinyOS Directory Structure

- apps/
- support/
- tos/
“support” Directory

• make/
  - Makefile scripts for build system

• sdk/
  - c and python for TOSSIM
  - java for supporting java apps/utils
Support Apps/Utils?

- apps/tests/TestSerial

- TestSerialC.nc, TestSerialAppC.nc compile into hex, gets loaded on a platform

- TestSerial.java, TestSerialMsg.java compiled into class file, run on a PC under JRE

- PC reads/writes COM1; Platform read/writes UART0
“make” Directory

- Microcontroller directories: avr, msp, pxa27x
- targets: mica2, micaz, telosb, etc.
- Makerules
“sdk” Directory

• Only need to worry about: python/tinyos/tossim/*

• Support libraries for simulator

• Need to set:

    PYTHONPATH=/opt/tinyos-2.x/support/sdk/python
Main TinyOS Directory Structure

- apps/
- support/
- tos/
“apps” Directory

• Sample applications
• Good ones to test with:
  – Blink
  – tests/TestSerial
TOSSIM

- Can compile projects to TOSSIM, creates XML and python code
- Define network of devices
- Use python environment to set through platform or network operation
Issues

• TOSSIM for TinyOS-2.0 has some problems...

• First: Only micaz supported right now
Compiling for TOSSIM

• Example: apps/Blink

• Linux:
  make micaz sim

• Cygwin:
  make micaz sim-cygwin
This Creates...

- in apps/Blink/:
  - TOSSIM.py
  - TOSSIM.pyc
  - _TOSSIM.dll
  - app.xml
implementation {

    event void Boot.booted() {
      call Timer0.startPeriodic( 250 );
      call Timer1.startPeriodic( 500 );
      call Timer2.startPeriodic( 1000 );
    }

    ...
}
Running Simulation

- Make sure PYTHONPATH is set!
- `> python`

```python
>>> from TOSSIM import *
>>> t = Tossim([])
>>> t.runNextEvent()
0
>>> m = t.getNode(0);
>>> m.bootAtTime(243);
>>> t.runNextEvent()
1
```
BlinkC

implementation
{

    uint8_t templInt = 0;

    event void Boot.booted()
    {
        templInt = 2;
        call Timer0.startPeriodic( 250 );
        call Timer1.startPeriodic( 500 );
        call Timer2.startPeriodic( 1000 );
    }

...
Looking Deeper

- Want to see internal state
- >>> from TOSSIM import *
- >>> from tinyos.tossim.TossimApp import *
- >>> n = NescApp("The App", "app.xml")
- >>> vars = n.variables.variables()
- >>> t = Tossim(vars)
- >>> m = t.getNode(0);
- >>> v = m.getVariable("BlinkC.tempInt")
- >>> v.getData()
- 0
Continue Running

- >>> m.bootAtTime(234);
- >>> t.runNextEvent()
- 1
- >>> v.getData()
- 2
Debug Statements

• Second TOSSIM 2.x Issue: Debug Statements aren't working

```java
implementation
{
    event void Boot.booted()
    {
        dbg("Boot", "Booted", 6);
        ...
    }
    ...
...
```
Smart House Tour
Smart House Tour

- After Oak Hammock Gate:
  - First right
  - Go to end of road
  - Left
  - First right