Wireless Sensor Networks

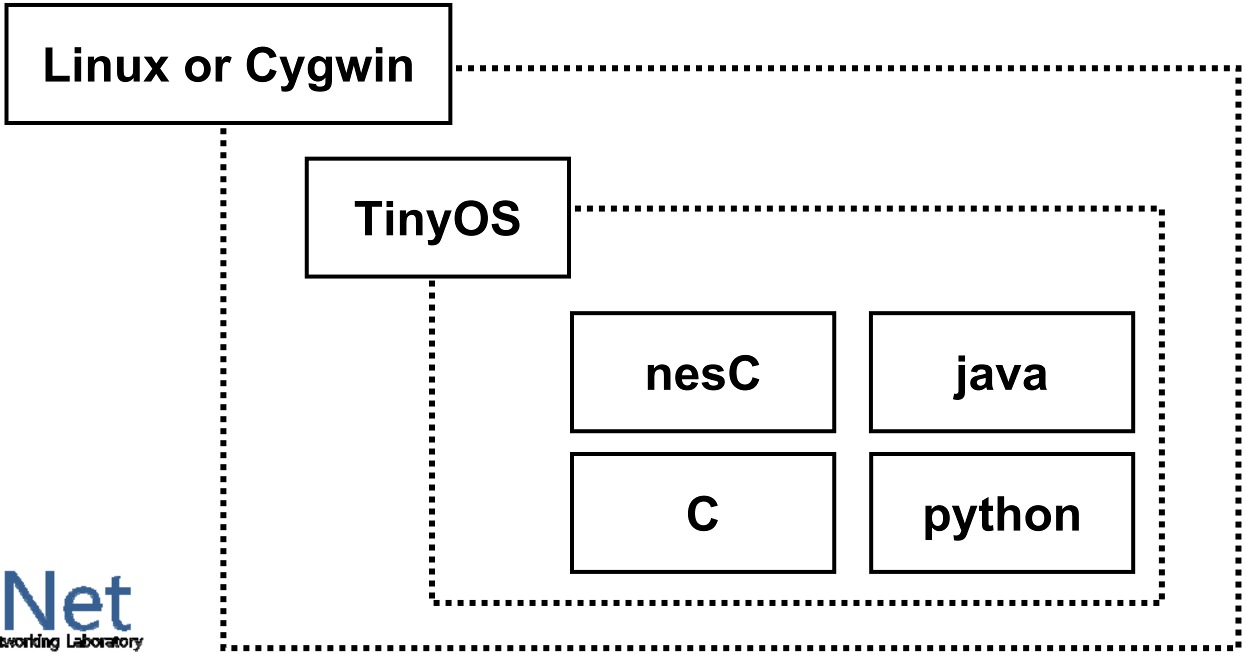
Blink Example

Lecturer: Junseok KIM
http://usn.konkuk.ac.kr/~jskim

WiMNet
Wireless and Mobile Networking Laboratory

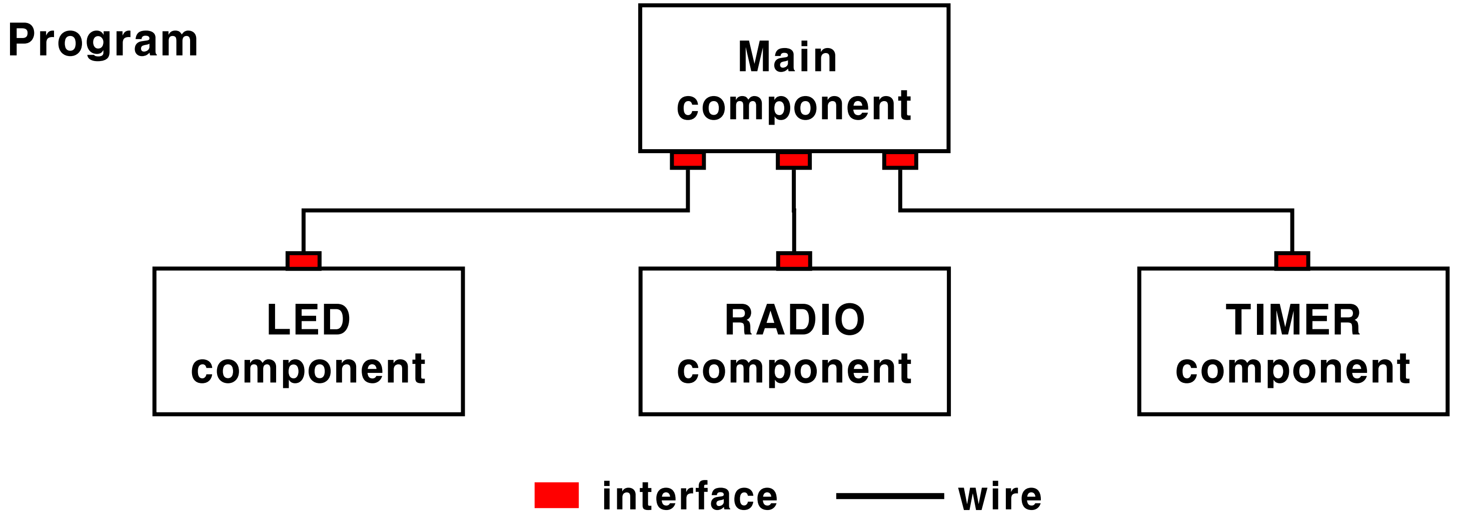
Introduction of TinyOS

- TinyOS is an unique Operating System (OS) for small wireless sensing devices
- Development environment of TinyOS is Linux
- Windows user uses Cygwin to make virtual linux environment
- C /nesC / phython are used for TinyOS programming
- NesC is used mainly



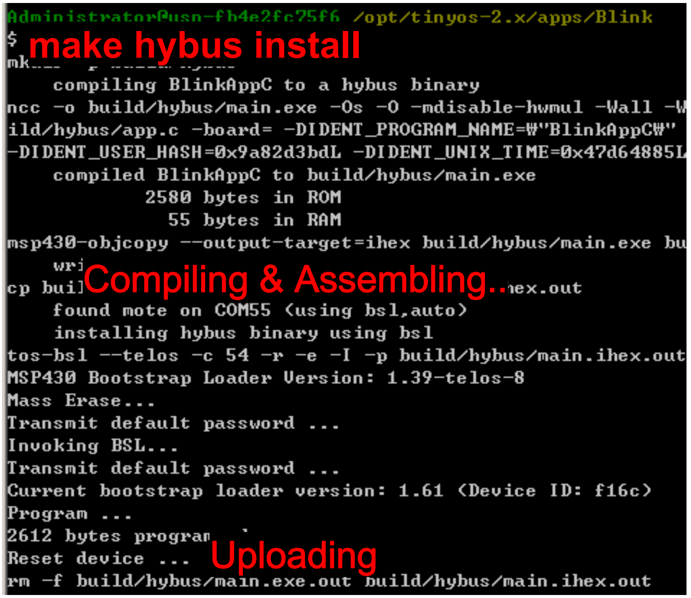
NesC

- NesC is extension of C language and component-based
- A TinyOS program consists of components and interfaces



Blink Example

- $ cd /opt/tinyos-2.x/apps/Blink
- $ make hybus install

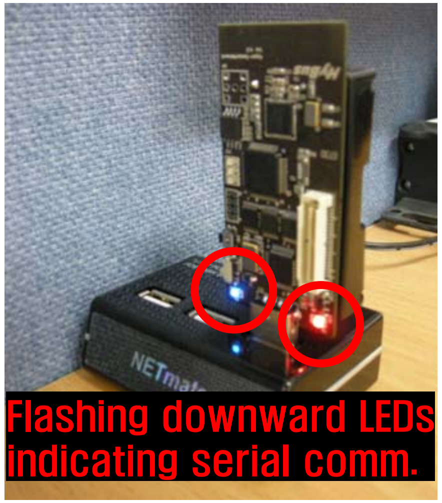


```
Administrator@usn-fb4e2fc75f6 /opt/tinyos-2.x/apps/Blink
$
mk
    compiling BlinkAppC to a hybus binary
ncc -o build/hybus/main.exe -Os -O -mdisable-hwmul -Wall -W
ild/hybus/app.c -board= -DIDENT_PROGRAM_NAME=₩"BlinkAppC₩"
-DIDENT_USER_HASH=0x9a82d3bdL -DIDENT_UNIX_TIME=0x47d64885L
    compiled BlinkAppC to build/hybus/main.exe
            2580 bytes in ROM
             55 bytes in RAM
msp430-objcopy --output-target=ihex build/hybus/main.exe bu
    wri
cp bui                                        ıex.out
    found mote on COM55 (using bsl,auto)
    installing hybus binary using bsl
tos-bsl --telos -c 54 -r -e -I -p build/hybus/main.ihex.out
MSP430 Bootstrap Loader Version: 1.39-telos-8
Mass Erase...
Transmit default password ...
Invoking BSL...
Transmit default password ...
Current bootstrap loader version: 1.61 (Device ID: f16c)
Program ...
2612 bytes program
Reset device ...
rm -f build/hybus/main.exe.out build/hybus/main.ihex.out
```

make hybus install

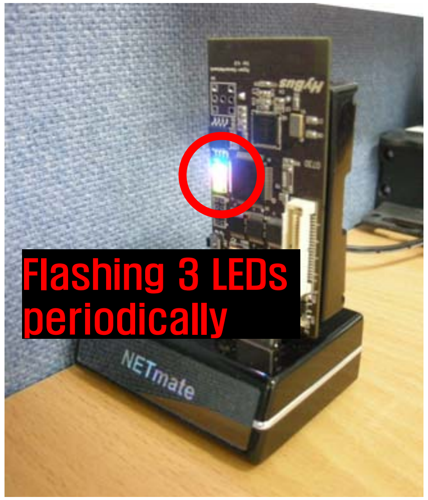
Compiling & Assembling..

Uploading



Flashing downward LEDs indicating serial comm.

during uploading

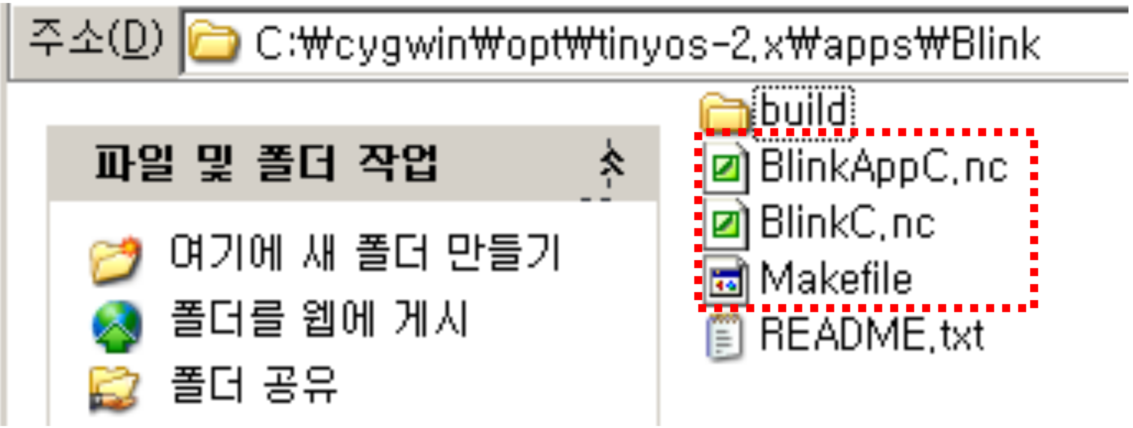


Flashing 3 LEDs periodically

after uploading

Blink Example (Cont.)

- There are three important files:
  - BlinkAppC.nc (configuration)
  - BlinkC.nc (module)
  - Makefile (for gcc complier)



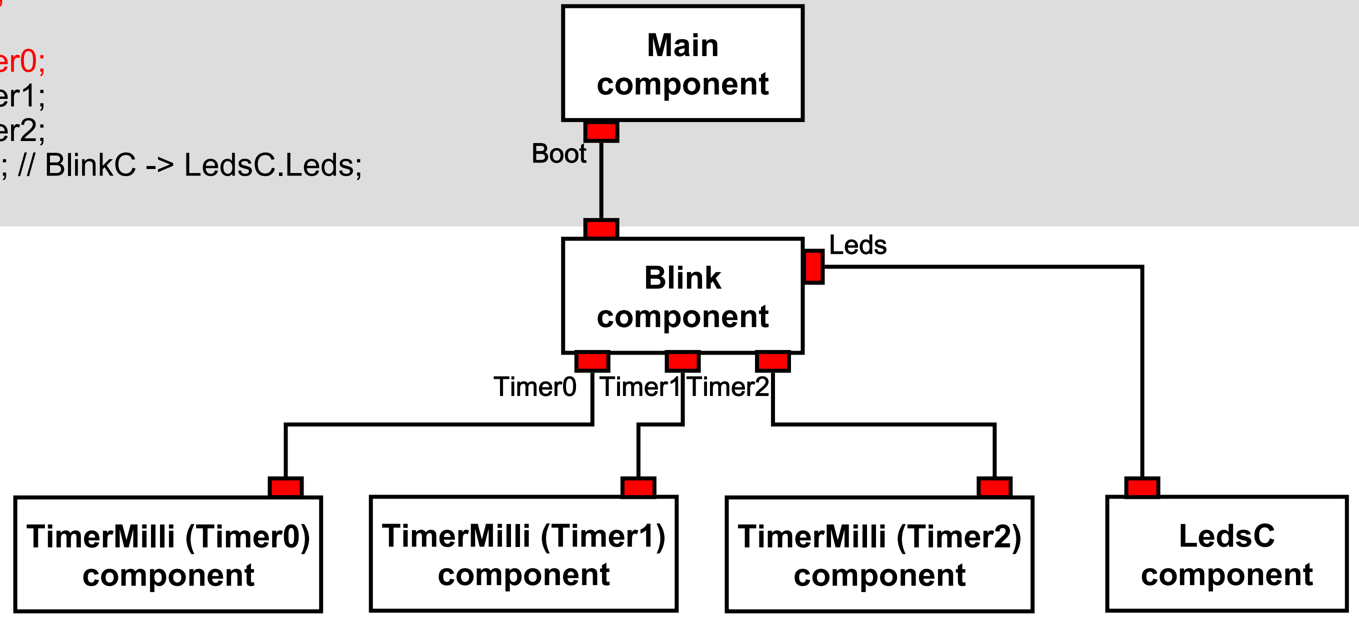
주소(D) C:₩cygwin₩opt₩tinyos-2.x₩apps₩Blink

build
BlinkAppC.nc
BlinkC.nc
Makefile
README.txt

Blink Example (Cont.)

BlinkAppC.nc

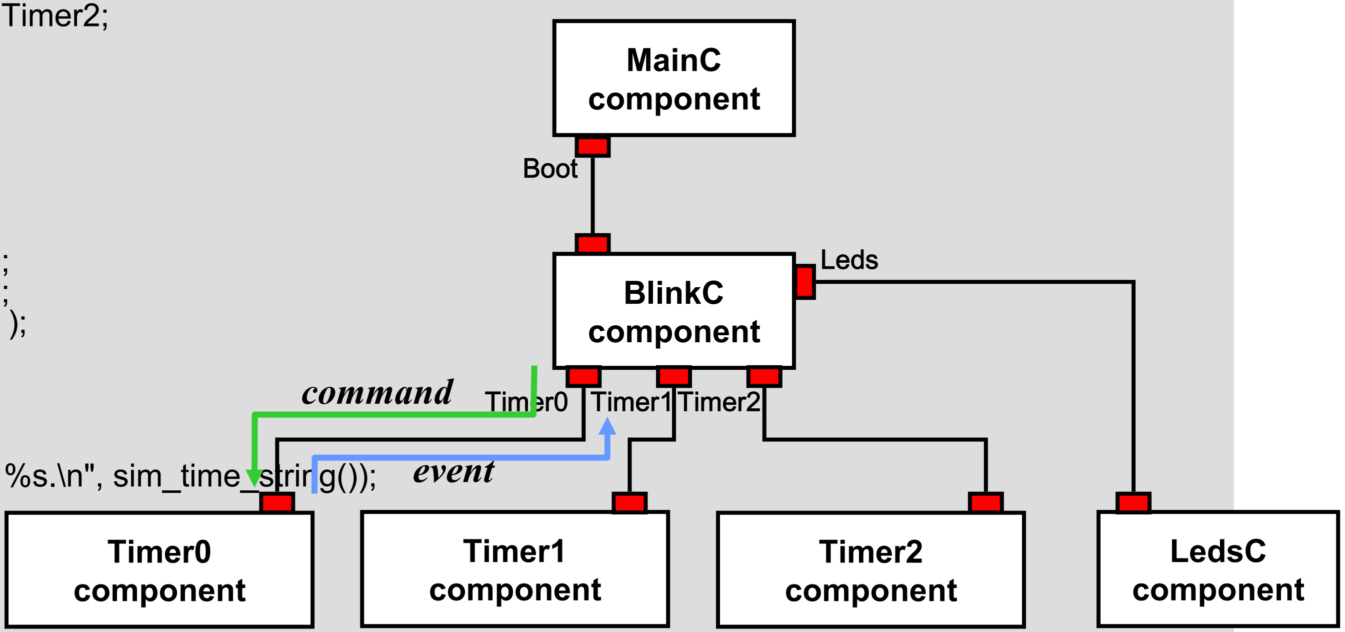
```
configuration BlinkAppC
{
}
implementation
{
  components MainC, BlinkC, LedsC;
  components new TimerMilliC() as Timer0;
  components new TimerMilliC() as Timer1;
  components new TimerMilliC() as Timer2;


  BlinkC -> MainC.Boot;

  BlinkC.Timer0 -> Timer0;
  BlinkC.Timer1 -> Timer1;
  BlinkC.Timer2 -> Timer2;
  BlinkC.Leds -> LedsC; // BlinkC -> LedsC.Leds;
}
```


Main component
Boot
Blink component
Leds
Timer0
Timer1
Timer2
TimerMilli (Timer0) component
TimerMilli (Timer1) component
TimerMilli (Timer2) component
LedsC component

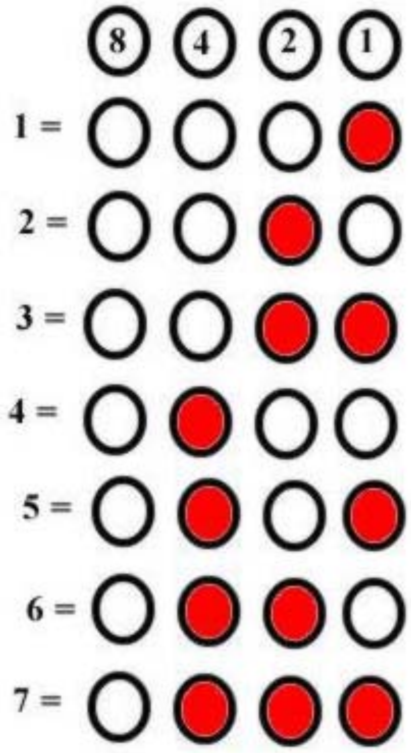
Blink Example (Cont.)

BlinkC.nc

```
module BlinkC @safe()
{
  uses interface Timer<TMilli> as Timer0;
  uses interface Timer<TMilli> as Timer1;
  uses interface Timer<TMilli> as Timer2;
  uses interface Leds;
  uses interface Boot;
}
implementation
{
  event void Boot.booted()
  {
    call Timer0.startPeriodic( 250 );
    call Timer1.startPeriodic( 500 );
    call Timer2.startPeriodic( 1000 );
  }

  event void Timer0.fired()
  {
    dbg("BlinkC", "Timer 0 fired @ %s.\n", sim_time_string());
    call Leds.led0Toggle();
  }

  event void Timer1.fired()
  {
    dbg("BlinkC", "Timer 1 fired @ %s \n", sim_time_string());
    call Leds.led1Toggle();
  }

  event void Timer2.fired()
  {
    dbg("BlinkC", "Timer 2 fired @ %s.\n", sim_time_string());
    call Leds.led2Toggle();
  }
}
```



Homework: With a Timer, make a binary counter



	8	4	2	1
1 =	○	○	○	●
2 =	○	○	●	○
3 =	○	○	●	●
4 =	○	●	○	○
5 =	○	●	○	●
6 =	○	●	●	○
7 =	○	●	●	●

Q and A