LED Programming in Contiki OS

Cd contiki

Cd examples/

Mkdir ledmegha

Cd ledmegha

Gedit ledmotes.c

#include<stdio.h>

#include “contiki.h”

#include “dev/leds.h”

// blink the LEDs from 0 to 7 in binary and prints the status

// open leds.h

Static struct etimer blinktimer;

// etimer is event timer

Static uint8\_t blinks;

PROCESS(blink\_process, “LED BLINK PROCESS”);

AUTOSTART\_PROCESSES(&blink\_process);

PROCESS\_THREAD(blink\_process, ev, data)

{

PROCESS\_BEGIN();

Blinks=0;

While(1)

{

Etimer\_set(&blinktimer, CLOCK\_SECOND);

// CLOCK\_SECOND generates 1 to 10 ms timing signals according to motes. // open timer.h , contiki-> core-> sys-> timer.h

// use sky mote

PROCESS\_WAIT\_EVENT\_UNTIL(ev==PROCESS\_EVENT\_TIMER);

Leds\_off(LEDS\_ALL);

Leds\_on(blinks & LEDS\_ALL);

Blinks++;

//%0.2X shows last 3 binary digits

//leds\_get(), returns a character

Printf(“State of LED &0.2X \n”,leds\_get());

}

PROCESS\_END();

}

//go to examples folder, go to make file of hello-world

//create a new file named as , Makefile, save in the folder ledmegha

//copy the content of make file and change the project name to CONTIKI\_PROJECT=ledmotes

//name of the mote is “NATIVE”

1. Do terminal compilation, type make( in the path of the folder /ledmegha$ make TARGET=native)
2. Ls
3. ./ledmotes.native
4. State of leds are shown

// we will do for sky motes

STEP: To open cooja simulator

1. Go to contiki folder
2. Cd tools
3. Cd cooja
4. Ant run
5. Create a new simulation: LED App
6. Go to view, and enable LEDs
7. Go to motes, add new mote, sky motes
8. Browse to ledmotes.c file
9. Compile
10. Create 5 motes, linear positioning
11. X: **20 to 20**, y: **20 to 20**, z:  **0 to 100**
12. Start simulation control
13. Status of led is shown, right click , show led on sky 4
14. Go to tools, mote interface viewer, sky 1
15. Right click on mote, in network window, go to mote information, view mote(ports and its hardware)