Course “Algorithmic Foundations of Sensor Networks”
Lecture 6: ContikiOS Introduction and Tutorial

Sotiris Nikoletseas
Associate Professor

Department of Computer Engineering and Informatics
University of Patras, Greece

Spring Semester 2016-2017
The hardware

TelosB mote

- Texas Instruments MSP430 Microcontroller 8MHz
- 10kB RAM
- IEEE 802.15.4
- Integrated Onboard Antenna
- Integrated Temperature, Light, Humidity sensor
- Low power
- USB programmable
Contiki

- OS for the IoT and constrained devices networking
- Open source
- Written in C (BSD license)
- Support for IPv4, IPv6
Event driven

- Flow controlled by events (timers, hw interrupts, radio etc)
- Kind on resources
- Harder to program (state variables, callbacks)
Multithreading

- Each thread is given a timeslot
- Scheduled by OS
- Costly in terms of memory (stack space), CPU
Protothreads

- Event-driven
- Threaded programming style
- Cooperative scheduling
Contiki processes

Contiki processes are event-driven

```c
#include "contiki.h"

/*------------------*/
PROCESS(hello_world_process, "Hello world process");
AUTOSTART_PROCESSES(&hello_world_process);
/*------------------*/
PROCESS_THREAD(hello_world_process, ev, data)
{
    PROCESS_BEGIN();
    /*...Doing stuff...*/
    PROCESS_END();
}
/*------------------*/
```
Contiki timers

- etimer sends an event on expiration
- ctimer calls function on expiration
- rtimer calls function on real-time
- Functions:

```c
void timer_set(struct timer *t, clock_time_t interval);
void timer_reset(struct timer *t);
void timer_restart(struct timer *t);
int timer_expired(struct timer *t);
```
Contiki leds

- 3 LEDS on board
- Pretty easy to play with!
- Definitions @ contiki/core/dev/leds.h

```c
unsigned char leds_get(void);
void leds_set(unsigned char leds);
void leds_on(unsigned char leds);
void leds_off(unsigned char leds);
void leds_toggle(unsigned char leds);
```
Contiki Events

- Loop structure
- Wait for an event

```c
PROCESS_BEGIN();
active = 0;
SENSORS_ACTIVATE(button_sensor);

while(1) {
    PROCESS_WAIT_EVENT_UNTIL(ev == sensors_event &&
                              data == &button_sensor);
    leds_toggle(LEDS_ALL);
    if(!active) {
        /* activate light sensor */
        SENSORS_ACTIVATE(light_sensor);
        printf("Light: %d\n", light_sensor.value(0));
    } else {
        /* deactivate light sensor */
        printf("Light: %d\n", light_sensor.value(0));
        SENSORS_DEACTIVATE(light_sensor);
    }
    active ^= 1;
    leds_toggle(LEDS_ALL);
}
PROCESS_END();
```
Install instant Contiki
- Instant Contiki is an entire Contiki development environment in a single image
- Install instant Contiki in VMware player
- Full installation guide: http://www.contiki-os.org/start.html

Check motes
- Connect TelosB
- Give permissions to port
  - `sudo chmod 777 /dev/ttyUSB0`
- Go to a contiki example folder
  - `make TARGET=sky motelist`

Make project
- Go to: `/contiki/examples/hello-world`
- `make TARGET=sky hello-world`
Upload code

- inside project file
  - make TARGET=sky hello-world.upload
- you can choose motes using index from motelist
  - make TARGET=sky hello-world.upload MOTE=1
- if using the same TARGET you can save it and omit it
  - make TARGET=sky savetarget
  - make hello-world.upload
Serial Debug

- You can print debug messages through serial port using `printf()`

```c
#include "contiki.h"

#include <stdio.h> /* For printf() */

PROCESS(hello_world_process, "Hello world process");

AUTOSTART_PROCESSES(&hello_world_process);

PROCESS_THREAD(hello_world_process, ev, data)
{
    PROCESS_BEGIN();

    printf("Hello, world\n");

    PROCESS_END();
}
```
Where to begin IV

- Read with any serial terminal
  - Built-in: make TARGET=sky login
  - CuteCom

- Baud rate: 115200
- Stop bits: 1
- No parity
- DONâ€™T forget giving PERMISSIONS
References

TelosB

Contiki
- http://www.contiki-os.org/
- http://www.slideshare.net/ADunkels/building-day-2-upload-1?next_slideshow=1

Cutecom
- http://cutecom.sourceforge.net/