Design of Embedded Systems (DES)

Jozef Hooman

http://www.cs.ru.nl/~hooman/DES/

Course 6
8 October 2014
Planning

• **Today**: industrial examples + discuss #10 + explain #11 (LED clock)
  13:30 – 15:00: Available in lab for questions

• **15 Oct**: discuss #11 (clock demo) + first assignment with small Lego Rover using **LeJOS & Eclipse**

• **22 Oct**: discuss Rover results + Bluetooth extension

• **29 Oct & 5 Nov**: Autumn break*

• **12 Nov**: discuss Rover results + DSL/Eclipse explanation
  NOTE: in **HG 01.028**

  *when needed: finalize Xenomai part
Exercise 11: Clock

- Fantazein (http://www.youtube.com/watch?v=p--CEJmZmgE)
Exercise 11: Clock

Full sweep (back and forth) with frequency of ≈ 8Hz

LEDs on/off to write text
D0: LED on top  D7: lowest LED  (LED is on = bit is 1)

interrupt when S6 goes from high to low and then to high again

→ Enable interrupt on parallel port
Clock interrupt

light sensor blocked

Interrupt S6

light sensor
Parallel port

Parallel port consists of 3 bytes:
- Data port: 8 bits D0 … D7
- Status port: 5 in (read only) (S3 … S7), 3 unused
- Control port: 4 out (r/w) (C0 … C3), 4 setup/unused

Note: not all bits corresponds to physical pins
E.g., C4 enables interrupts

Green pins (18 – 25) are grounded to 0 Volt
Generate interrupt

If interrupts enabled, i.e., bit C4 has been set and voltage on S6 (pin 10) is raised from 0 V to +5 V (e.g., by connecting it to ground [one of pins 19-25] and then disconnecting it, *as done by push button*) then parallel port generates interrupt on IRQ 7
Enable/disable interrupts

- Address of control port byte: 0x37A
- Set C4 to enable interrupts before starting ISR task

```c
ioperm(0x37A, 1, 1); /* set port input/output permissions to read 1 byte*/
byte = inb(0x37A);  /* get the byte */
byte = byte | 0x10;  /* hex 10 = binary 00010000 */
outb(byte, 0x37A);  /* output the byte */
```

- Reset C4 to disable interrupts at the end of main

```c
byte = inb(0x37A);
byte = byte & 0xEF; /* hex EF = binary 11101111 */
outb(byte, 0x37A);
```

NOTE: on lab PCs result may depend on setting of previous programs
Set/Reset Data Bits

To avoid segmentation fault: “ioperm(0x378, 1, 1);”
Use “char byte”, read data port with “byte = inb(0x378);”

• To set D3 (= bit 4) to 1:
  byte = byte | 0x08;  // 0x08 = 0000 1000
  outb(byte, 0x378)

• To set D3 to 0:
  byte = byte & 0xF7;  // 0xF7 = 1111 0111
  outb(byte, 0x378)
Exercise #11

a) Write X on LED clock - write in one direction only
   *PCs contain demo program* `draw_x_on_led_clock`
   *to test if hardware is OK (./draw_x_on_led_clock)*

b) Center X, i.e. put it exactly in the middle
   *(make corrections for the size of the light sensor)*

c) Write X in both directions

Bonus:
• Does your program work on both clocks?
• Will it continue correctly after disturbances?
• Write a message or number on the LED clock

Switch clocks off when not used
Assignment for 15 October 2014

Xenomai exercise #11
(programmable LED clock)

Optional:
• Does your program work on both clocks?
• Will it continue correctly after disturbances
• Write a message or number on the LED clock

Mail results
• source files + Makefile + explanation
before Tuesday 14 October 18:00