

Legos Implementation of the Emachine

BAY Team

Yanmei Li, Alessandro Pinto, Bruno Sinopoli

Description of the tasks

- 2 Tasks t_1 , t_2
- t_1 :
 - 2 Hz
 - Counter which is displayed on the screen
- t_2 :
 - 1 Hz
 - Reads the light sensor and emits:
 - Two beeps if light level $<$ threshold
 - One beep otherwise

Core Code

```
int main(){
  Init();
  while (1){
    wait_event(&trigger_function,0);
    enable_trigger = 0;
    update_enabled_trigger();
    while
    (!trigger_queue_is_empty()){
      einterpreter();
      dispatch_tasks();
    }
    enable_trigger = 1;
  }
}
```

```
void einterpreter(){
  trigger_address curr_trigger;
  Instr instruction;
  int PC=0;
  curr_trigger = get_enabled_trigger();
  PC=curr_trigger.address;
  while (PC != -1){
    instruction = program[PC];
    switch (instruction.op_code){
    case call:
      exec(driver_function)
      PC= program[PC].next;
    case schedule:
      task_set <- task_function;
      task_set.number_of_tasks ++;
      PC= program[PC].next;
    case future:

    enqueue_trigger(program[PC].trigger,program[PC].address);
      PC=program[PC].next;
    }
  }
}
```

Execution Principle

```
{call,(int>(&beep_driver),4,1,1,1),  
{call,(int>(&lcd_driver),4,1,1,2),  
{call,(int>(&light_driver),4,1,1,3),  
{schedule,(int>(&t1),1,1,1,4),  
{schedule,(int>(&t2),1,1,1,5),  
{future,0,1,timer,6,-1},
```

```
wakeup_t trigger_function(wakeup_t data){  
  if (enable_trigger == 1){  
    if (sys_time > next_time){  
      active_trigger[timer]=1;  
      next_time = sys_time + tick;  
      return 1;  
    };  
    return 0;  
  };  
};
```

Emachine

(execution end)

Task_set={t1,t2..}

Future trigger={{timer,address}...}

```
void light_driver(){  
  light_port=LIGHT_1;  
  -----  
}
```



```
/*-----Tasks-----*/
```

```
void t2(){  
    static int count=0;  
    lcd_port=count++;  
}
```

```
void t1(){  
    if (light_port < 100)  
        beep_port=sys_beep;  
    else  
        beep_port=double_beep;  
}
```

```
/*-----Drivers-----*/
```

```
/*Attention!!! in this implementation they are not  
really atomic*/
```

```
void light_driver(){  
    light_port=LIGHT_1;  
}
```

```
void beep_driver(){  
    dsound_play(beep_port);  
}
```

```
void lcd_driver(){  
    lcd_int(lcd_port);  
}
```

```
/*----Ports !!!!----*/
```

```
static int light_port;  
static int lcd_port;  
static note_t *beep_port;
```

```
/*-----Drivers-----*/
```

```
extern void light_driver();  
extern void beep_driver();  
extern void lcd_driver();
```

```
/*-----Tasks-----*/
```

```
extern void t1();  
extern void t2();
```