Programma

- legOS
  - controllo motori
  - sensori touch
  - sensori light
  - display
  - suono
  - task
  - timing
- gruppi
- emulatore

Motori in legOS

#include <dmotor.h>

- motor_*X*_dir(enum *MotorDirection*)
  - es: motor_a_dir(off)
- motor_*X*_speed(int *Speed*)
  - es: motor_a_speed(50)

MotorDirection fwd, rev, off, brake

Speed MIN_SPEED .. MAX_SPEED (0..255)

x a, b, c

Sensori Touch in legOS

#include <dsensor.h>

- TOUCH_*X*
  - es: TOUCH_1

valori 1 (pressed), 0 (not pressed)

x 1, 2, 3

Sensori Light in legOS

#include <dsensor.h>

- due modalità:

  passive mode ds_passive(&SENSOR_*X*)

  active mode ds_active(&SENSOR_*X*)

- LIGHT_*X*
  - es: LIGHT_3

valori 0 (scuro), 100 (chiaro)

x 1, 2, 3

Uso LCD in legOS

#include <conio.h>

- lcd_int(int *x*)
- lcd_clear()
- cputs(char *s)

  - Massimo di cinque caratteri
  - Refresh ogni 100 ms

Suono in legOS

#include <dsound.h>

- note_t struct {
  unsigned char *frequenza*;
  unsigned char *durata*
  }
- dsound_play(note_t **nota*)

  - *frequenza* = 0 $\rightarrow$ 55 Hz
  - *durata* in 1/16 s

Task in legOS

#include <unistd.h>

- int execi(&*funzione*, int *argc*, char ***argv*, int *priorità*, int *stack_size*)
  - es: *pid* = execi(&*trova_bordo*, 0, NULL, 10, DEFAULT_STACK_SIZE);
- kill(pid_t *pid*)

- *funzione* deve essere una funzione di tipo int che accetti in ingresso un int ed un **char
- *priorità* non deve eccedere PRIO_HIGHEST (20)

Timing in legOS

- sleep(unsigned int *secondi*)
  - es: sleep(10)
- msleep(unsigned int *millisecondi*)
  - es: msleep(100)
- wait_event(wakeup_t **funzione*, wakeup_t *dati*)
  - waitevent(&touch_pressed, 0)

wakeup_t : torna 1 se l'evento si è verificato, 0 altrimenti