

Facultat d'Informàtica de Barcelona
Universitat Politècnica de Catalunya

Real-Time Systems

4-LAB segway

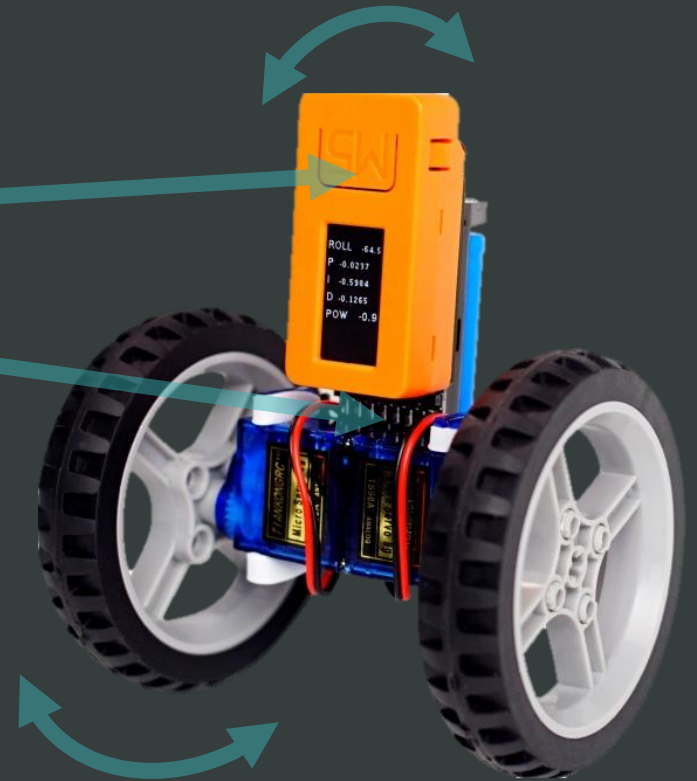
Antonio Camacho Santiago
`antonio.camacho.santiago@upc.edu`

The segway is called [BalaC M5](#)

The main objective is to keep a self-balancing robot (segway) upwards
This is accomplished by pushing back and forth the system to keep it stable
using again a PID controller

The systems has:

- M5StickC ESP32-PICO-D4
- Two motor drivers connected to the main board via I2C
- IMU
- WiFi
- LCD screen
- Lipo 16340 battery
- Many more features...



Tools:

A template for Arduino+PlatformIO+ESP32+M5StickC is provided with this document `segway.zip`

The template doesn't use tasks! There is only code in `setup()` and `loop()`

However, by default ESP32 uses FreeRTOS, so task creation is straight-forward

Note also that ESP32 is a dual-core RISC-V micro and implements a SMP (symmetric multi processing) version of FreeRTOS capable to attach tasks to cores if required.

For doing so use `xTaskCreatePinnedToCore()` instead of `xTaskCreate()`

TODO:

Define task and assign cores if required

Add as much features as possible (LCD, WiFi, joystick, telemetry, debugging, control tuning, ...)

...and keep the segway upwards

Zip the project and send it to the raco

Tips and tricks:

The board is powered on when connected to usb

To switch off the board, the power button should be pressed during at least 6 seconds

There is a switch in the lowest part of the motor driver to activate the servos (remember to switch it off when finished)

The PID controller is critical, it can not loose any deadline!

Also IMU filtering is mandatory to smoothly control de self-balancing robot