M1-EECS: Initiation to research methodologies
Tutorial 1: Raspberry Pico, Micropython and Data Analysis for Sailing Control.

Emmanuel WITRANT

December 31, 2025

1 Introduction

The aim of this tutorial is to guide you on using inputs and outputs on a Raspberry Pico board and program this board using *MicroPython*. The interface between the PC and the electronic board is done using the Python IDE (integrated development environment) named *Thonny* (`https://thonny.org/`).

As a lab preparation, briefly summarize the interests for using the Raspberry Pico board, and programming in *MicroPython* and *Thonny*.

Download the Latex source of this document on `https://plmlatex.math.cnrs.fr/read/kzcxdxxpssyf` and upload the files on your own PLM Latex or Overleaf space to edit directly your report.

Provide the detailed I/O schematics of the Pico board.

2 Thonny and Pico

2.1 Basic setup

Open Thonny and follow the guidelines to connect and program the Pico board. Make the led blink with different levels of brightness. Report your code, with comments, in the following box.

2.2 Multiple sensors

You want to use Pico to get the temperature from three sensors DS18B20. In order to do so:

1. get the technical information on theses sensors and draw the electrical circuit;

2. build the circuit and ask the lab instructor for validation;
3. write the Python code to acquire the measurements from the sensors and print them in the IDE[1], with detailled comments for each command. What is the minimum sampling time that we can expect?

   Report your code and comments in the following box:

4. modify this code as follows:
   - the sampling can be defined by the user and doesn't impact the code execution (e.g. read the sensors values at the right times rather than freezing the whole algorithm with the `sleep()` function;
   - store the measurements in a vector, with the data organized from left to right in the same order as your sensors;
   - add a function to store the time stamp and temperatures in an array and log it in a file when its size is too large for Pico's memory;
   - play with the sampling time to make it as small as possible.

   Report your code in the following box:

[1]Some hints can be found at `https://how2electronics.com/interfacing-ds18b20-sensor-with-raspberry-pi-pico/`

2.3 Actuation capabilities: A382 servo control

The goal is now to combine sensing and actuation capabilities. As an exemple, you have to set the angle of an A382 servomotor according to one of the sensors' temperature.

1. Draw the schematics of the electrical scheme that allows you to perform this task.

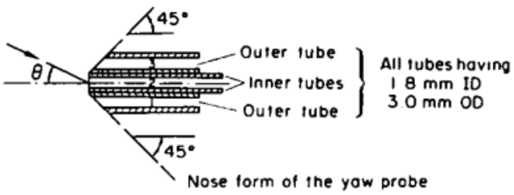
2. Update the code written in the previous section to include the servomotor actuation, ensuring that the motor rotates in a smooth way.

3 Cobra probes for flow measurement

3.1 Conrad-cobra probes

Conrad probes are considered the bests Pitot-like tubes for combined flow velocity and direction measurements. Consisting of 2 or 3 tubes soldered together, they provide an inexpensive solution. The 3-tubes version is sometimes referred to as "cobra" probe while "Conrad" denotes the 2-tubes version, but the litterature is not homogeneous in the name's choice. Experiments were carried out using 5 double-entries ABP2DRRN002ND2B3XX pressure transducers. The velocities and pressure distributions for such design are obtained using CFD simulations and are presented in Table 1.

We evaluate the design with three metalic tubes inserted in a 3D-printed structure depicted in Fig. 1b, following the design guidelines provided by [1]. The design with metal tubes proves more robust and reliable. The probes' outlets are connected to pressure transducers.



(a) Cobra probe's schematic [1]

(b) Metal tubes + 3D printed plastic version

Figure 1: Cobra probes

3.2 Electronics and flow connections

These initial tests are carried with 5 differential pressure sensors ABP2DRRN002ND2B3XX connected to a multiplexer TCA9548APWR and a Raspberry PI Pico, depicted on Figure 2 and with the connections indicated in Table 2. The tubes lengths are 20 cm from the probe to the T-tube connector and 12 cm (including extra Ts if necessary) from the T to the transducer.

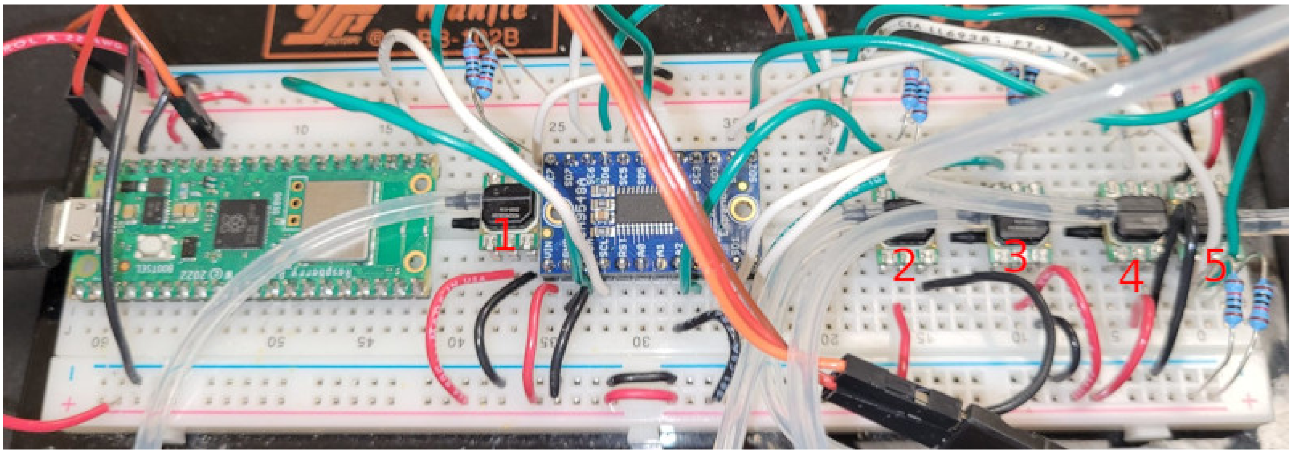


Figure 2: Prototype electronics with 5 sensors connected to a Rapberry PI and silicon tubes

3.3 Data analysis

Denoting the airflow angle as θ and p_l, p_c and p_r the pressures measured by the left, central and right probe, the total pressure p_0 and the dynamic pressure Δp are obtained from [1]:

$$p_l = p_0 + K_l(\theta)\Delta p \tag{1}$$
$$p_c = p_0 + K_c(\theta)\Delta p \tag{2}$$
$$p_r = p_0 + K_r(\theta)\Delta p \tag{3}$$

where $K_{l,c,r}$ are calibration functions that depend on θ only.

1. From the equations (1)-(3) and the reference litterature, design a method to calculate the calibration functions using the probe's measurements.

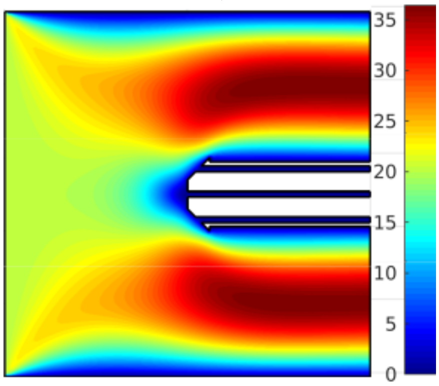
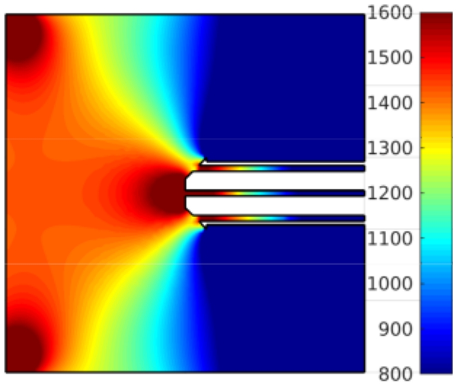
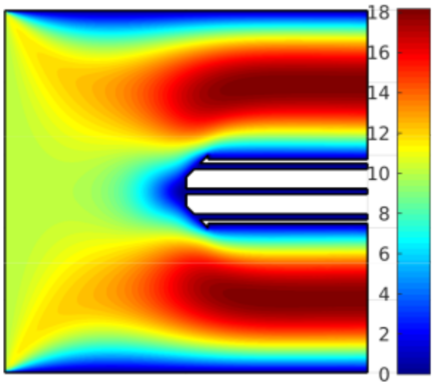
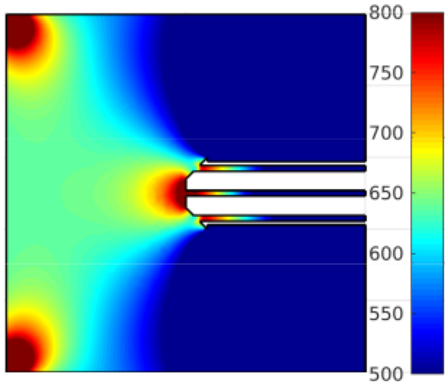
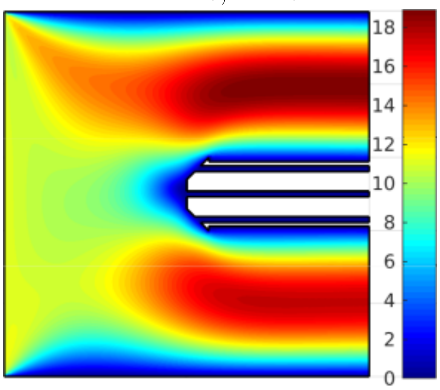
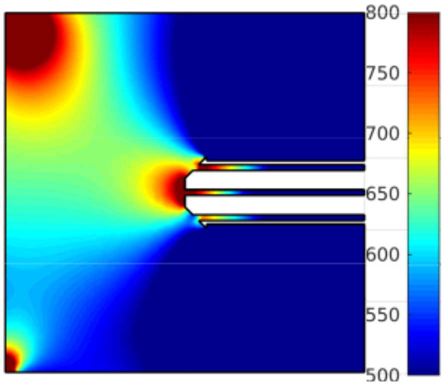
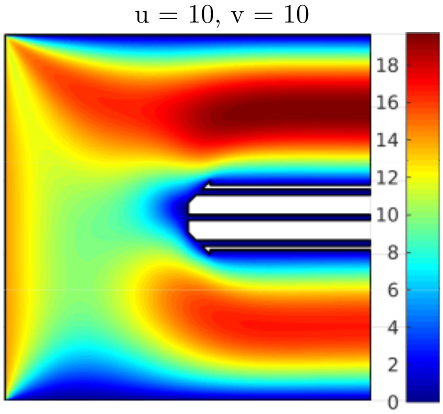
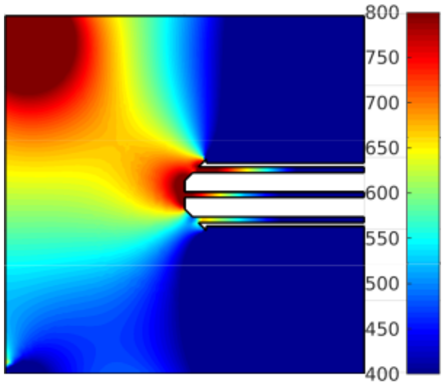
Velocity ($\sqrt{u^2 + v^2}$)	Pressure
u = 20, v = 0	
	
u = 10, v = 0	
	
u = 10, v = 5	
	
u = 10, v = 10	
	

Table 1: CFD simulation of the Cobra probe

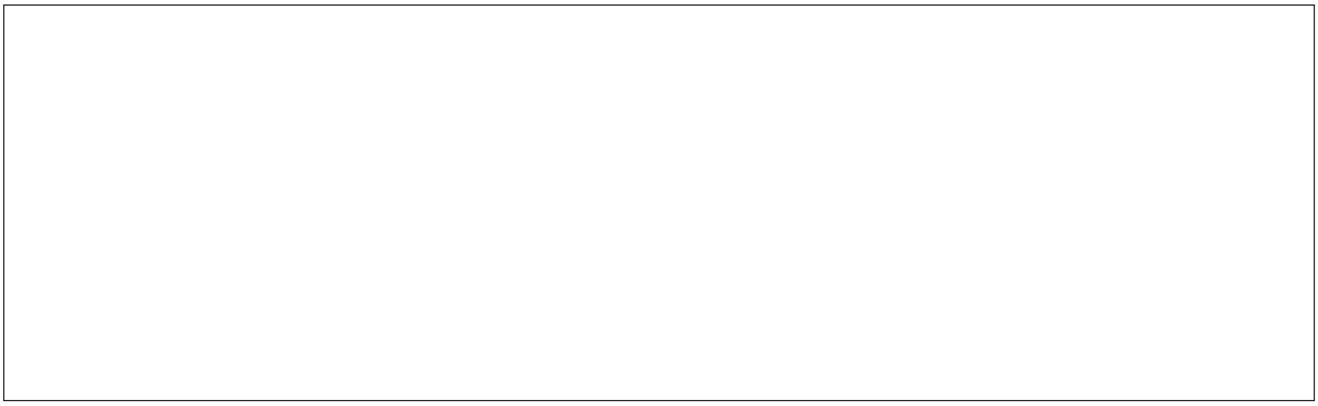
Sensor	Upper tube	Lower tube	Multiplexer channel (sc/sd)	Order in file
1	p_l	p_{atm}	5	5
2	p_l	p_c	0	1
3	p_c	p_{atm}	1	2
4	p_r	p_{atm}	6	4
5	p_r	p_c	2	3

Table 2: Summary of sensors'configuration for the electronic board used in Fig. 2

2. Download the data set `data_20250224_091018.csv`, corresponding to raw measurements taken in a basic wind tunnel that produced a turbulent flow. Design a CUMSUM Kalman filter that can smooth out the turbulences measured in the flow. Discuss the design and options

3. Write a Python code to implement your CUMSUM filter and calculate the calibration functions from the given data set.

4. Conclude on your method and the different design options.



References

[1] S.H. Chue. Pressure probes for fluid measurement. *Progress in Aerospace Sciences*, 16(2):147–223, 1975.