

e-skin lab @ Humanoids

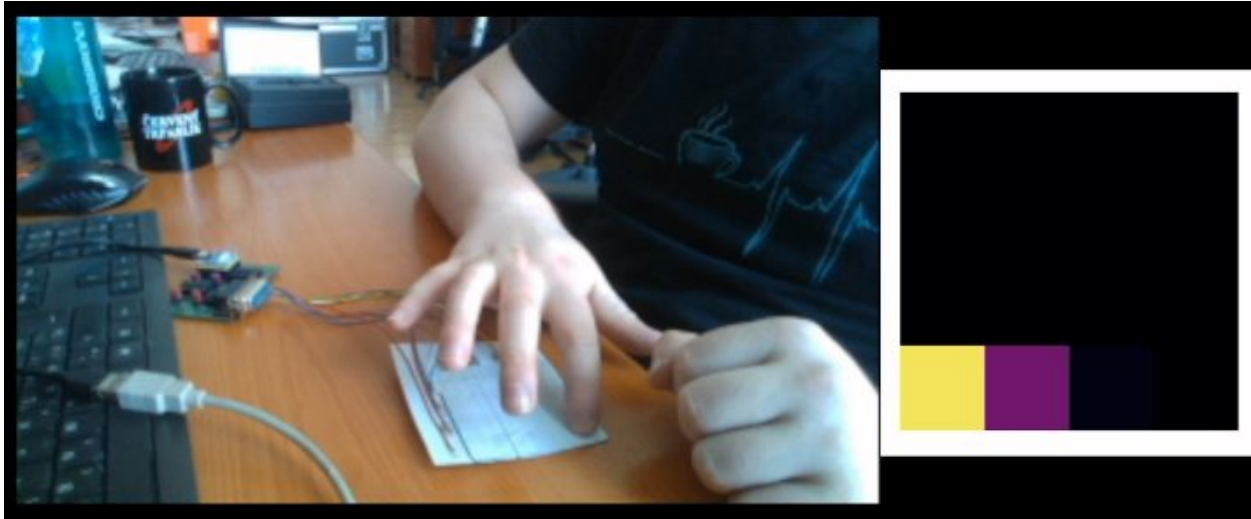
# How to read these slides

- some contain advanced information
  - these slides are there for students with higher knowledge of hardware
  - such slides will be marked with “evil iCub icon”
- Highlighted information is important for homework



# What we will make and use

- e-skin based on Velostat foil
- electronics based on RP2040 and MicroPython

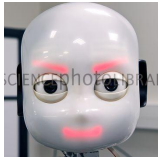


# Velostat

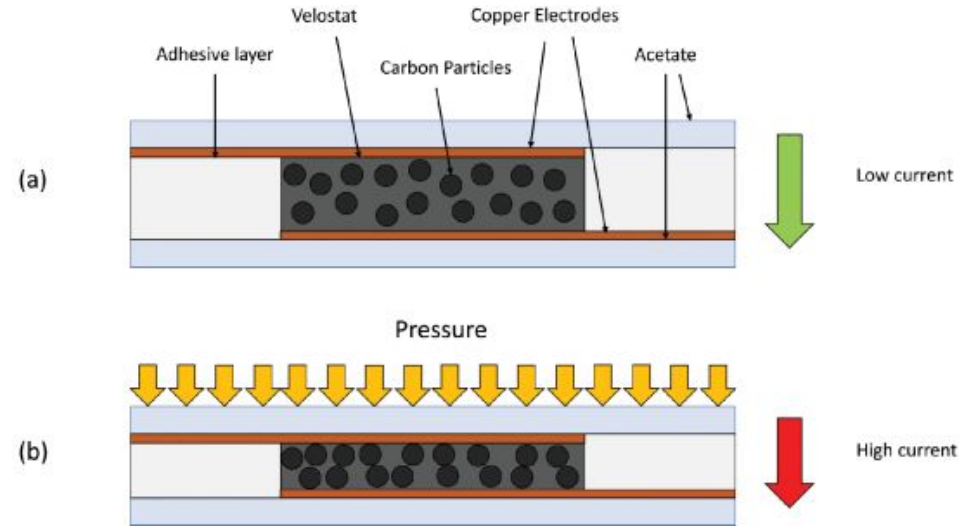
- Velostat = Polymeric foil filled with conductive carbon particles
- Primary developed for packaging
- Piezoresistive = Resistance dependent on deformation

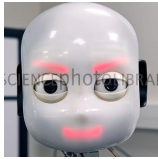


# Velostat as a sensor



- Electric conductivity thanks to
  - Percolation traces (randomly formed conductive paths)
  - Tunneling of electrons (even though particles are insulated by the polymer, they can still tunnel through with certain probability)
- **When pressed:**
  - particles moves closer together → more percolation traces + smaller distance between insulated particles → more current flowing → **lower resistance**





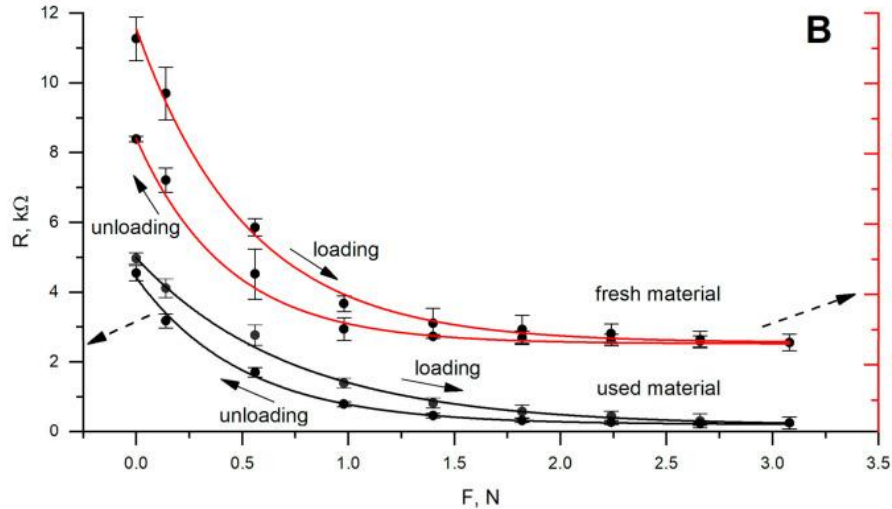
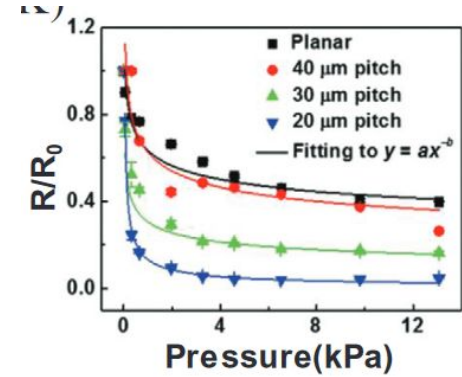
# Additional sources of piezoresistivity - Surface effects

- Skin patch is warped
  - with pressure contact area between velostat and electrodes increases
- Roughness of surfaces
  - with pressure microscopic structures are deformed increasing contact area

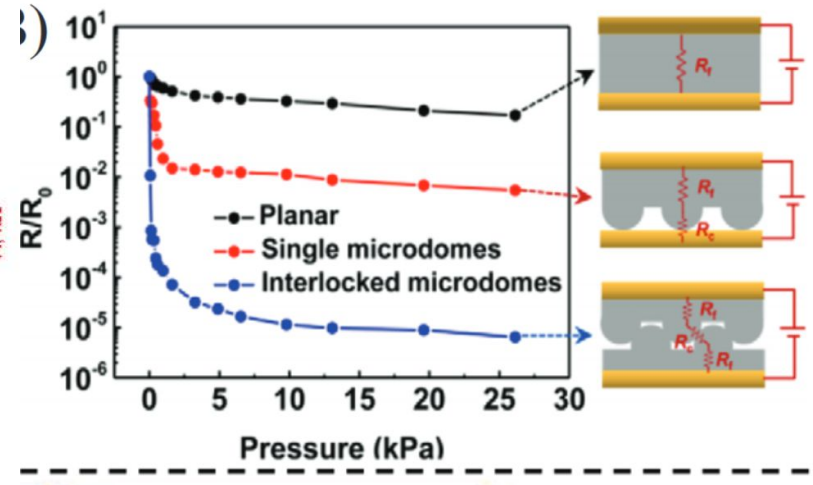


# Typical response

- Generally shown as R to pressure or force
- Typically relative resistance  $R/R_0$
- Resistance tactile sensors typically decreases with pressure



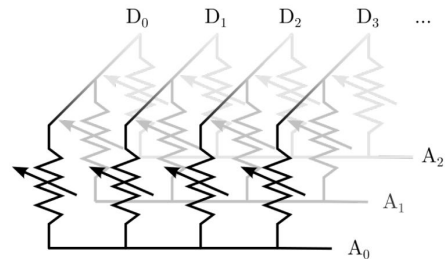
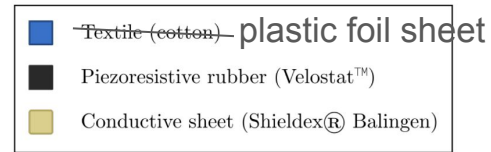
Velostat



SoTA microstructural sensors

# Construction - Resistive array

- Sandwich structure
- Row and column electrodes
- One sheet of velostat
- Insulation layers



Custom readout PCB secured on PLA holder, powered by 160 mAh LiPo battery

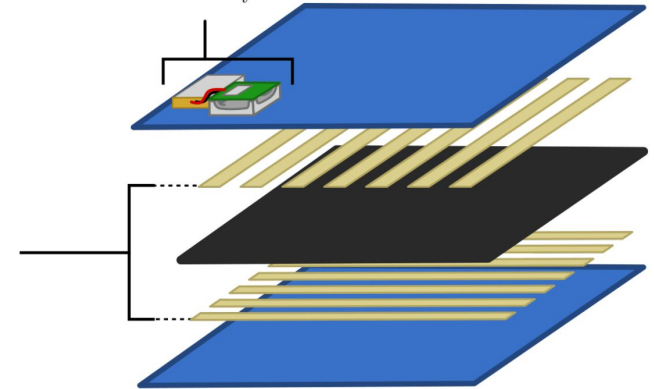
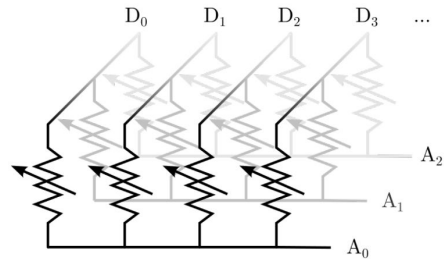


Image illustrative, taken from literature

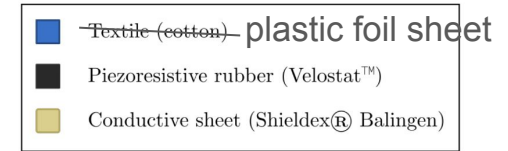
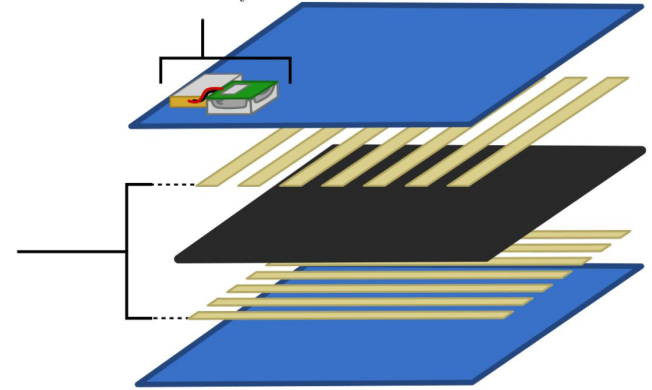


# Construction - Resistive array

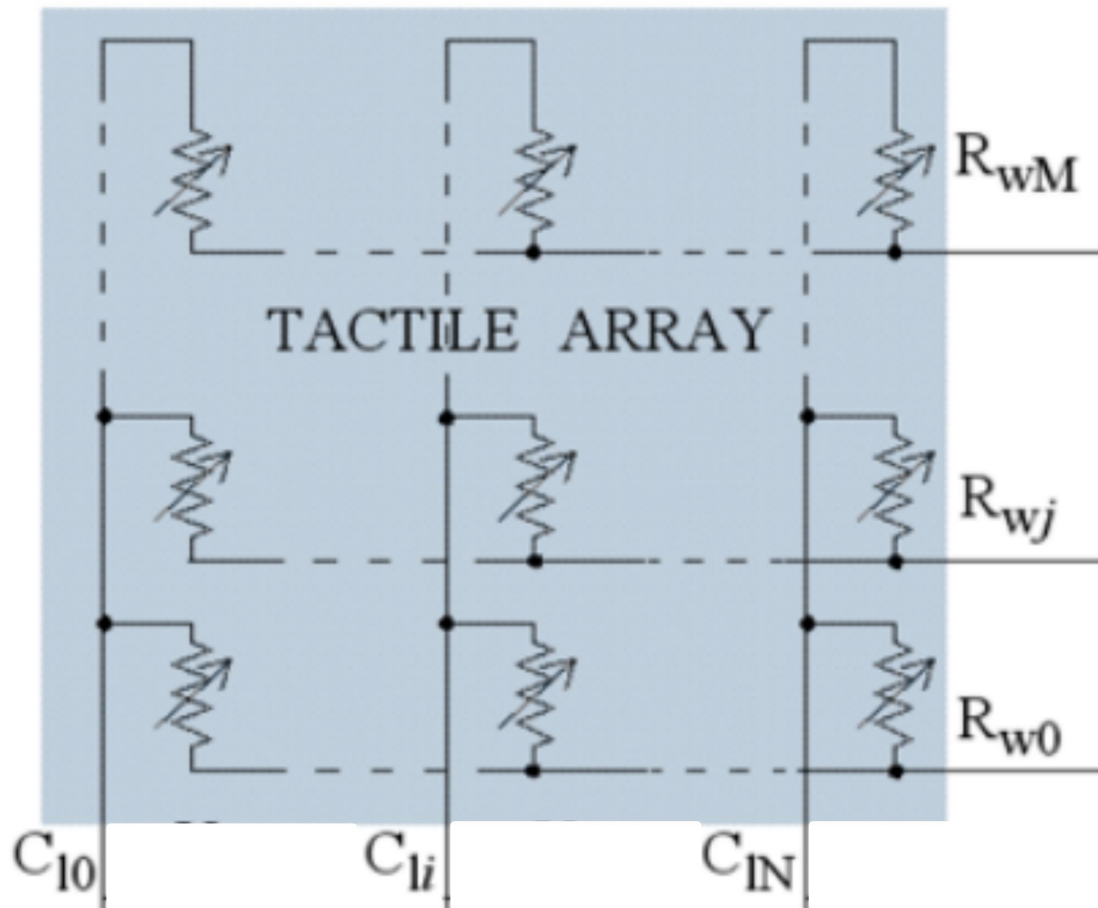
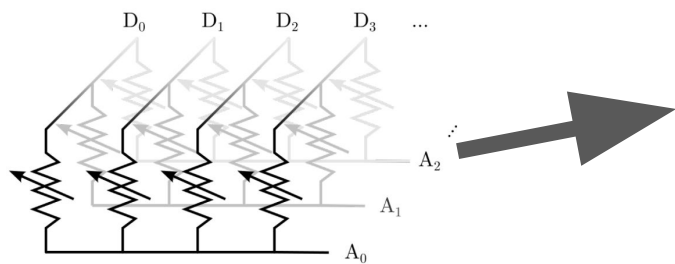
- With certain precision we can model e-skin as follows:
  - Where row and column electrodes overlap a resistor is formed
  - Each resistor is, thanks to velostat, piezoresistive and forms one tactile cell (taxel) as shown in schema



Custom readout PCB secured on PLA holder, powered by 160 mAh LiPo battery

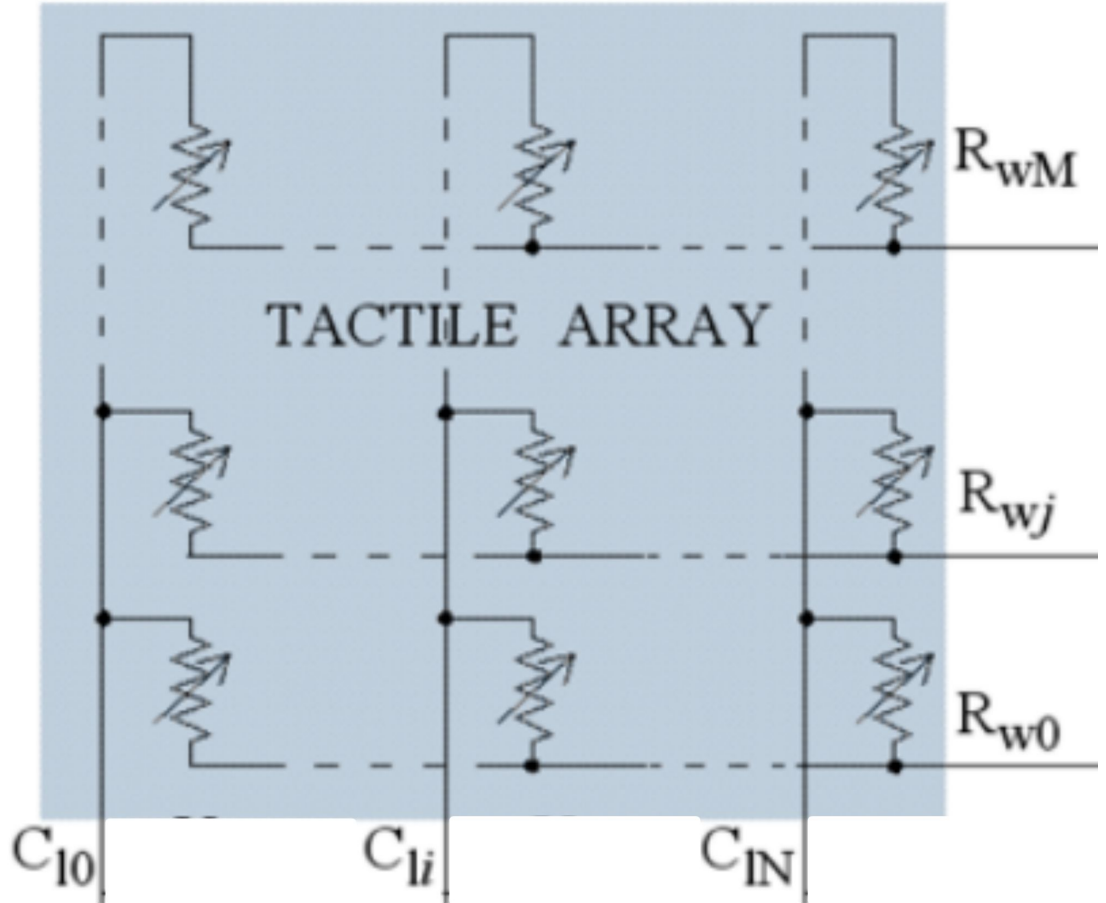


# Equivalent schema



# Equivalent schema

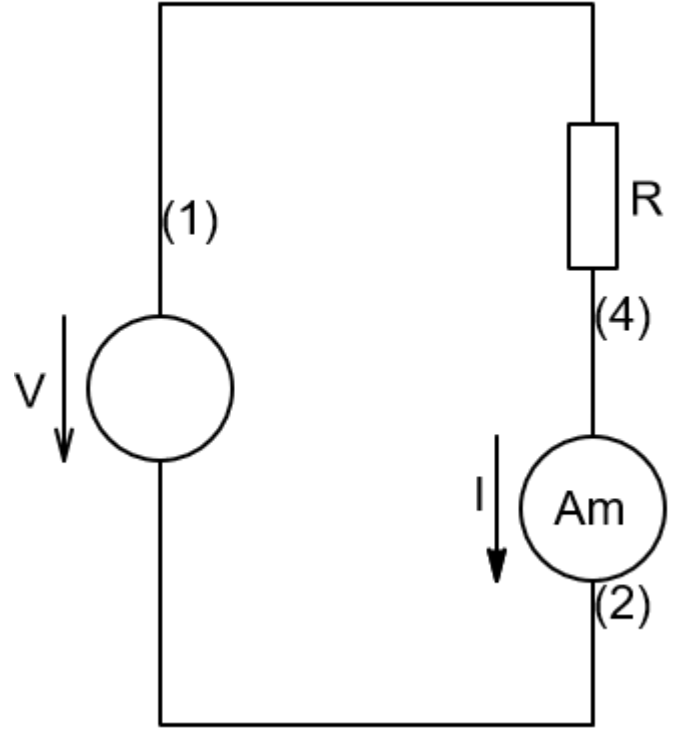
- We just re-drew the schema from 3D to 2D
- Similar to keyboard or memories



## now...how to measure resistance?

- we apply known constant voltage across the resistor and measure current

$$R = \frac{V}{I}$$



# How do we measure resistance without an Amp. meter

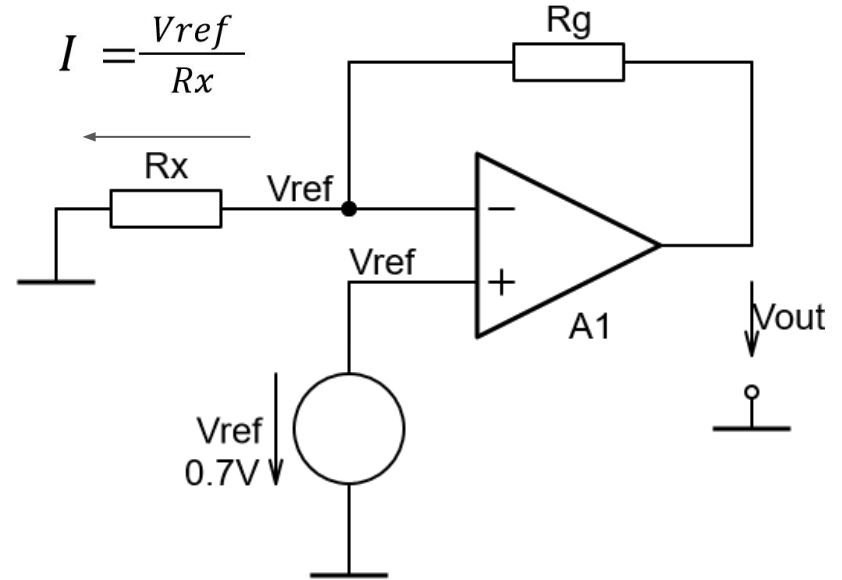
- we will use this circuit with operational amplifier
- Through  $R_x$  passes current  $I$
- We get this formula

$$V_{out} = \left( \frac{R_g}{R_x} + 1 \right) V_{ref}$$

$R_x$  → Measured resistor

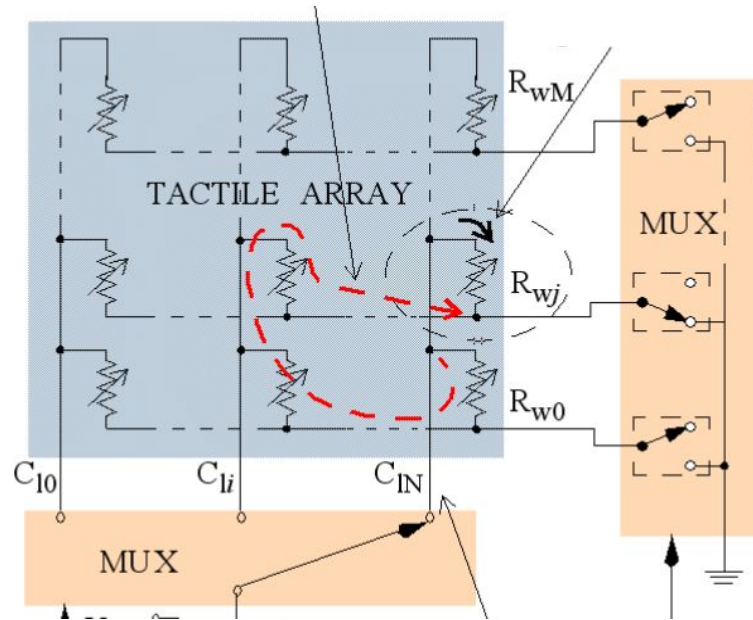
$R_g$  → Known (gain) resistor

$V_{ref}$  → Voltage reference (**0.7V**)



# How to measure resistive array?

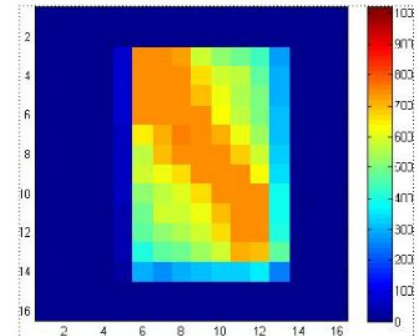
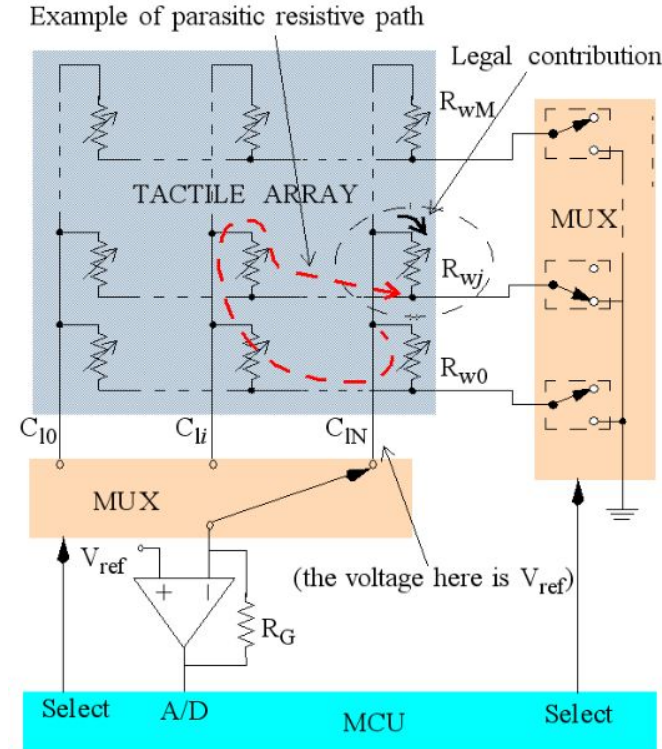
- Well, first we need to address individual resistor
- Let's try apply voltage to column  $C_{IN}$  and row  $R_{wj}$ .
- To do this, let's use some electronic switches (multiplexers, MUX)





# Crosstalk currents

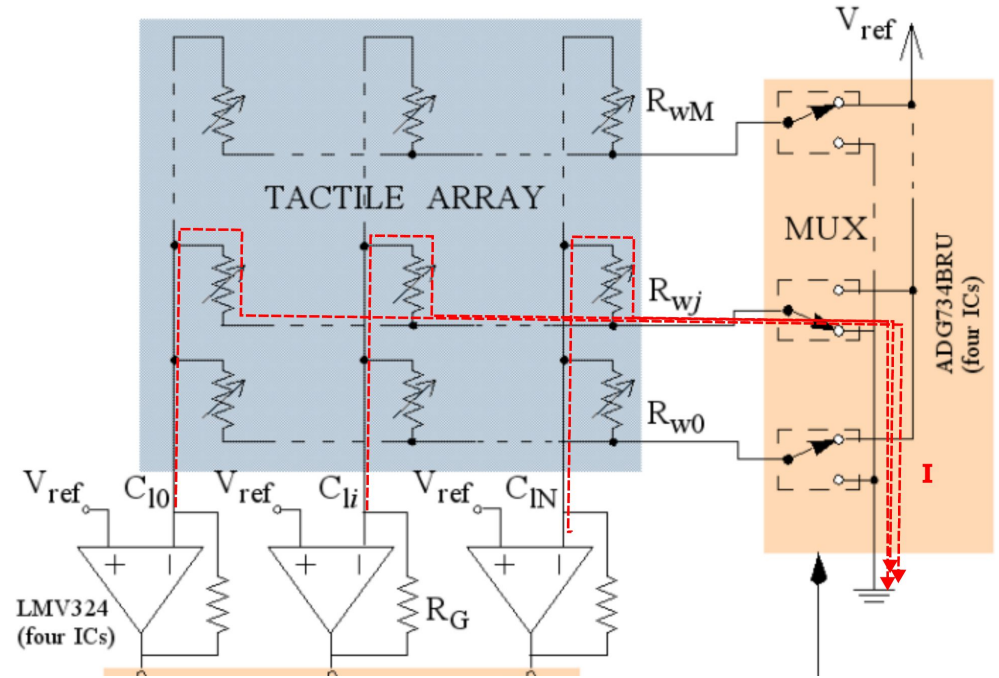
- For reason on previous slide when we try to measure only one resistor the measurement is influenced by all the resistors in the array
- These unwanted currents are called crosstalk currents.
- The result of them is very blurred tactile image
- so what can we do about it?





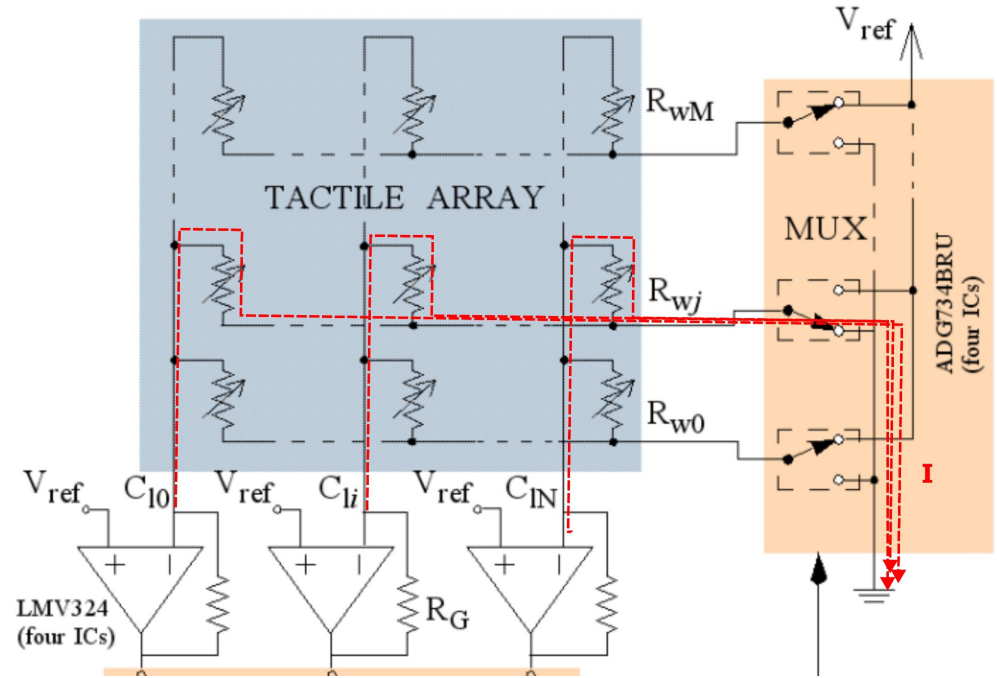
# Crosstalk current compensation

- We will try to suppress the crosstalk by adding more operational amplifiers
- under each column we add one
- Next, we will not use column mux and in row mux we no longer switch between individual rows but for each row we switch between  $V_{ref}$  and GND signals.



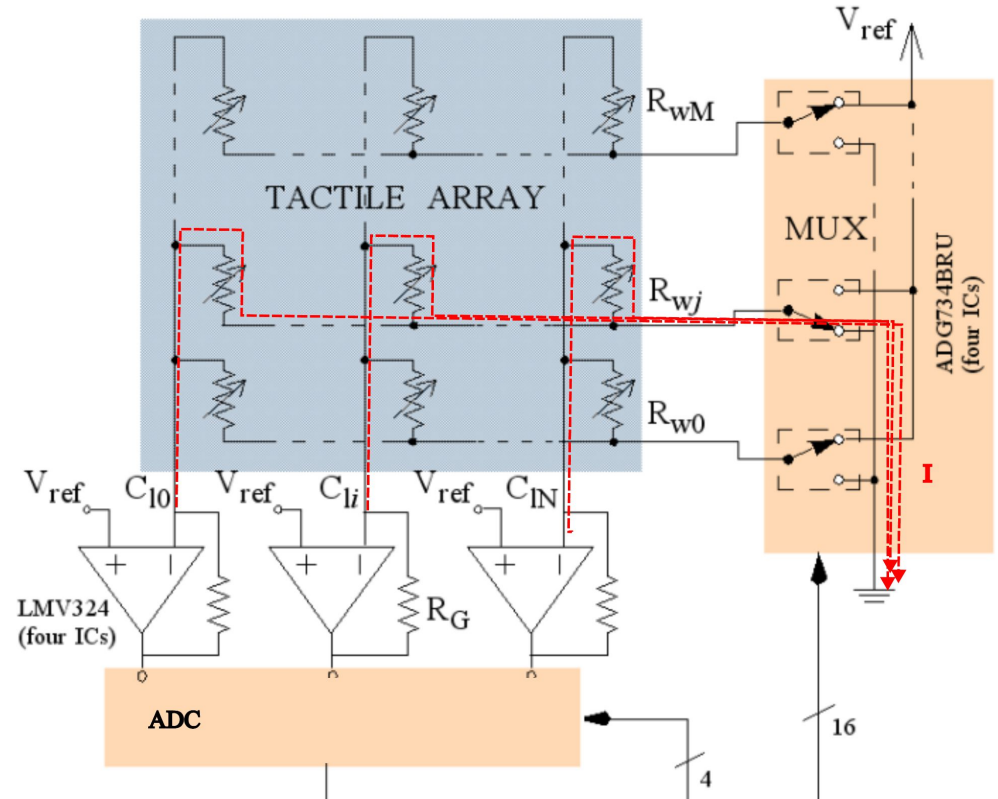
# Crosstalk current compensation

- As a result we do not try to pass current through individual resistors but through **one row at once**
- **We only select rows**
- Since all not-selected rows are on  $V_{ref}$  potential, which is same as potential on all the columns, there is **no voltage across resistors in not-selected rows and therefore no crosstalk current flows**



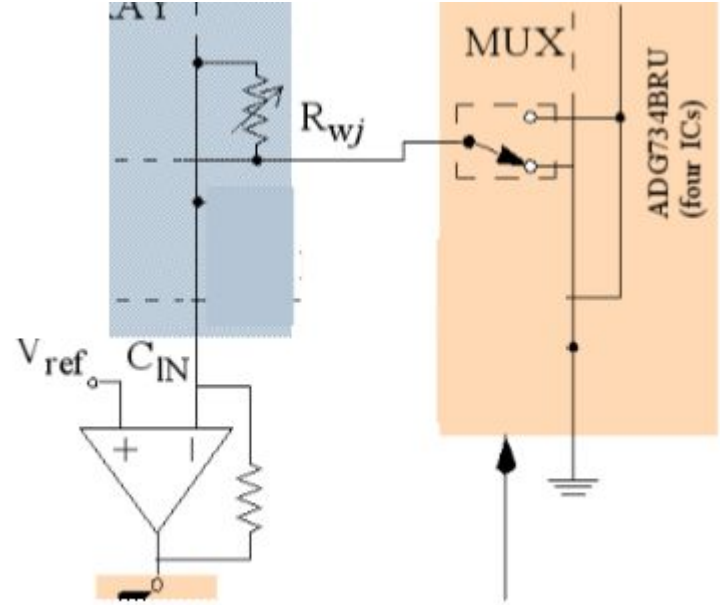
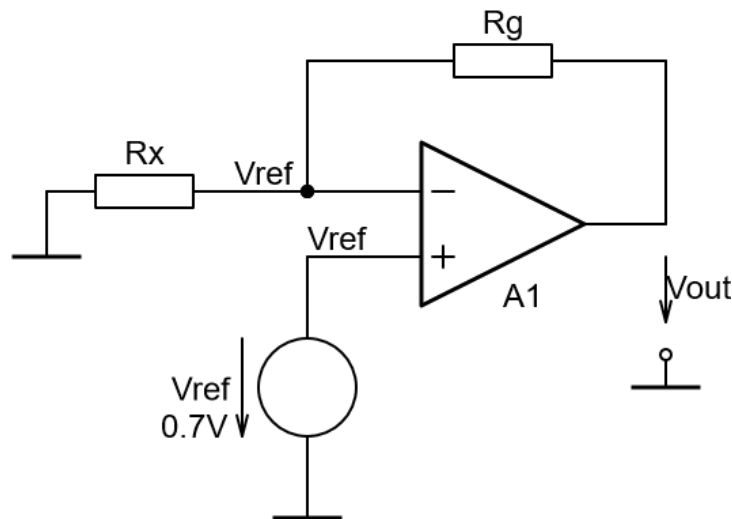
# Crosstalk current compensation

- To get resistance value for individual resistors in the array we will measure each operational amplifier output voltage using ADC.
- even though current passes through all the resistors in the row, these currents do not influence each other and therefore we can pretend we measure each other individually.



# Crosstalk current compensation

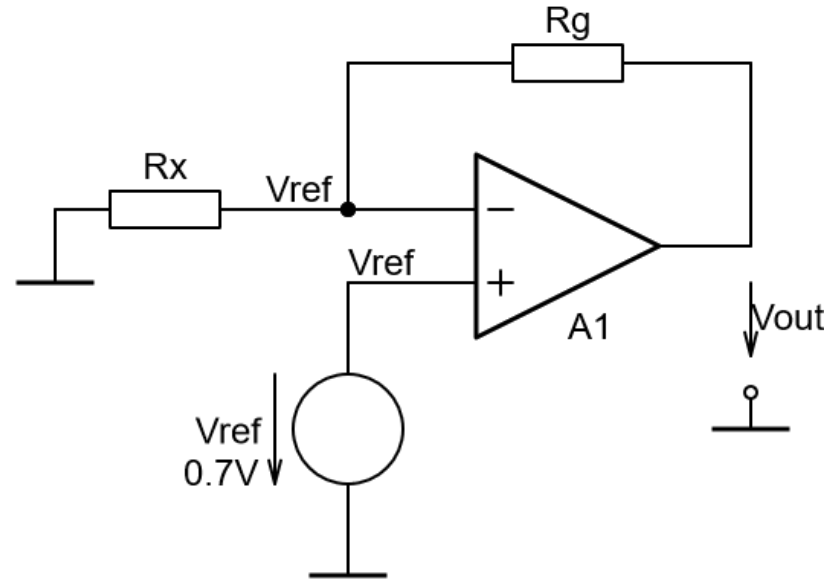
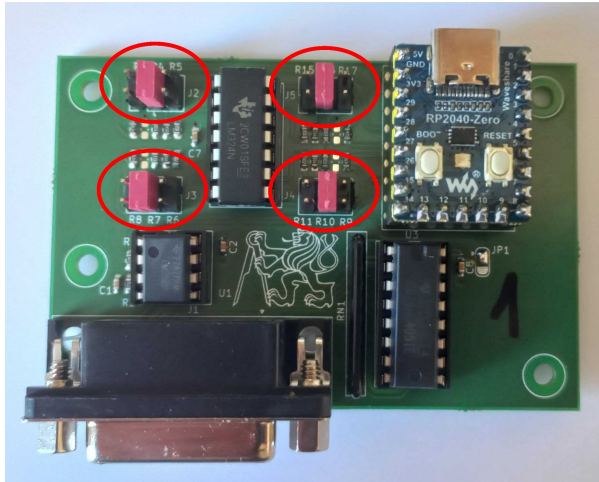
- Ok, so let's now focus only on one measured resistor in selected row
- wait, that's what we started with!



$$V_{out} = \left( \frac{R_g}{R_x} + 1 \right) V_{ref}$$

# How do we get Rg?

- Selection of Rg can be done using jumper on board
- Values (from center → out):
  - 1 kΩ
  - 4.7 kΩ
  - 10 kΩ



$$V_{out} = \left( \frac{R_g}{R_x} + 1 \right) V_{ref}$$

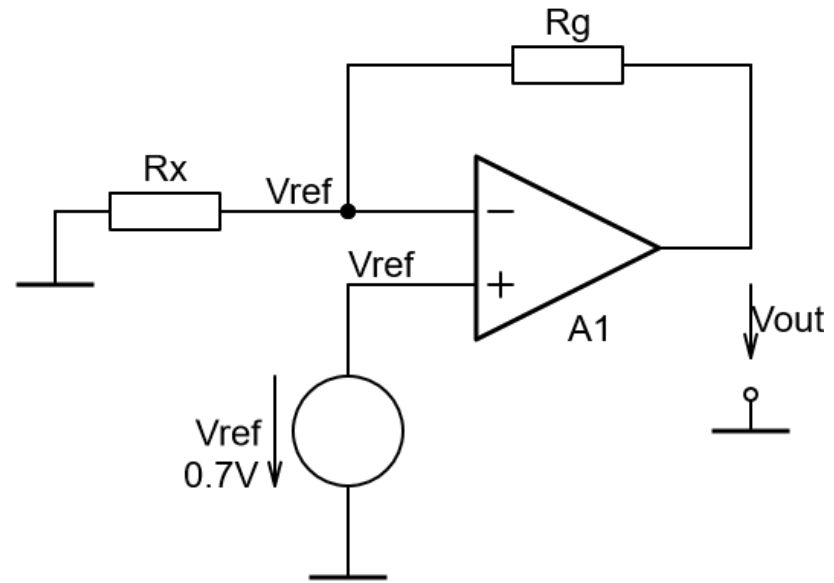
Rx → Measured resistor

Rg → Known (gain) resistor

Vref → Voltage reference (**0.7V**)

## In summary

- We switch between rows using multiplexer
- Columns are sampled by ADC
- Thanks to operational amplifiers (assuming ideal) we can simplify to single resistor measurement
- To select row in MicroPython use function `set_row`



$$V_{out} = \left( \frac{R_g}{R_x} + 1 \right) V_{ref}$$

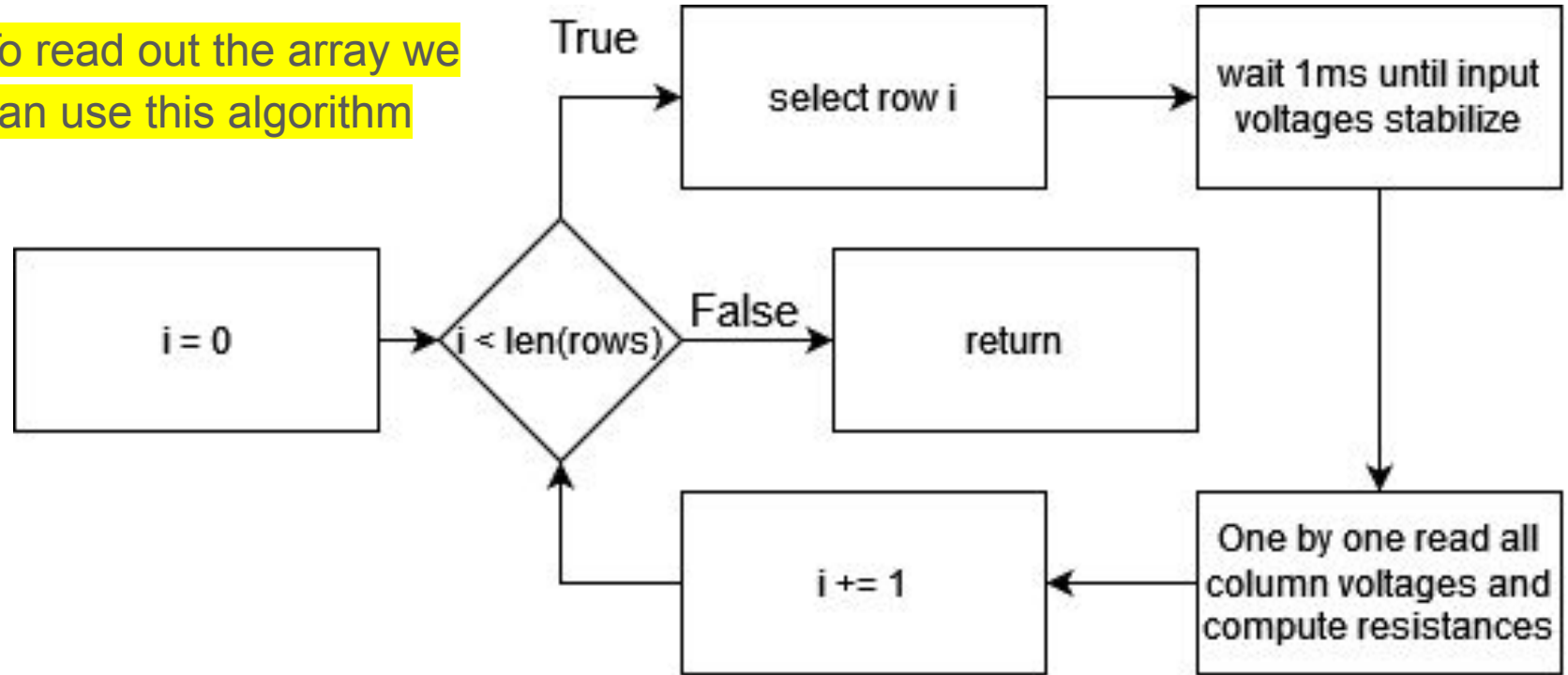
$R_x$  → Measured resistor

$R_g$  → Known (gain) resistor

$V_{ref}$  → Voltage reference (**0.7V**)

# Read out algorithm and timing

- To read out the array we can use this algorithm

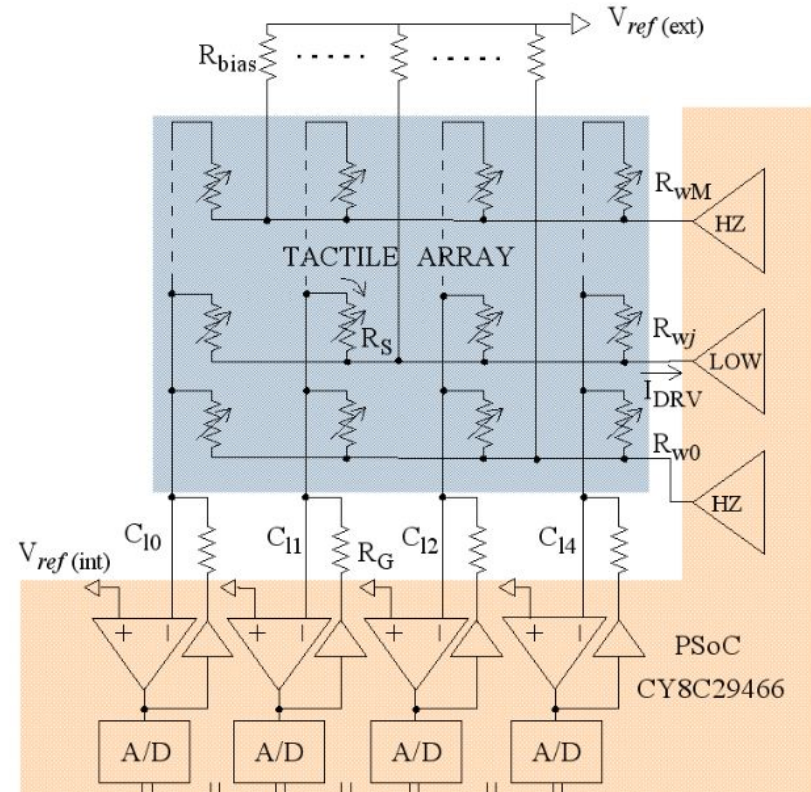






# Real Implementation

- Pull-up resistors to  $V_{ref}$
- Mux switches between GND and High Impedance
- Each op amp sampled by analog to digital converter





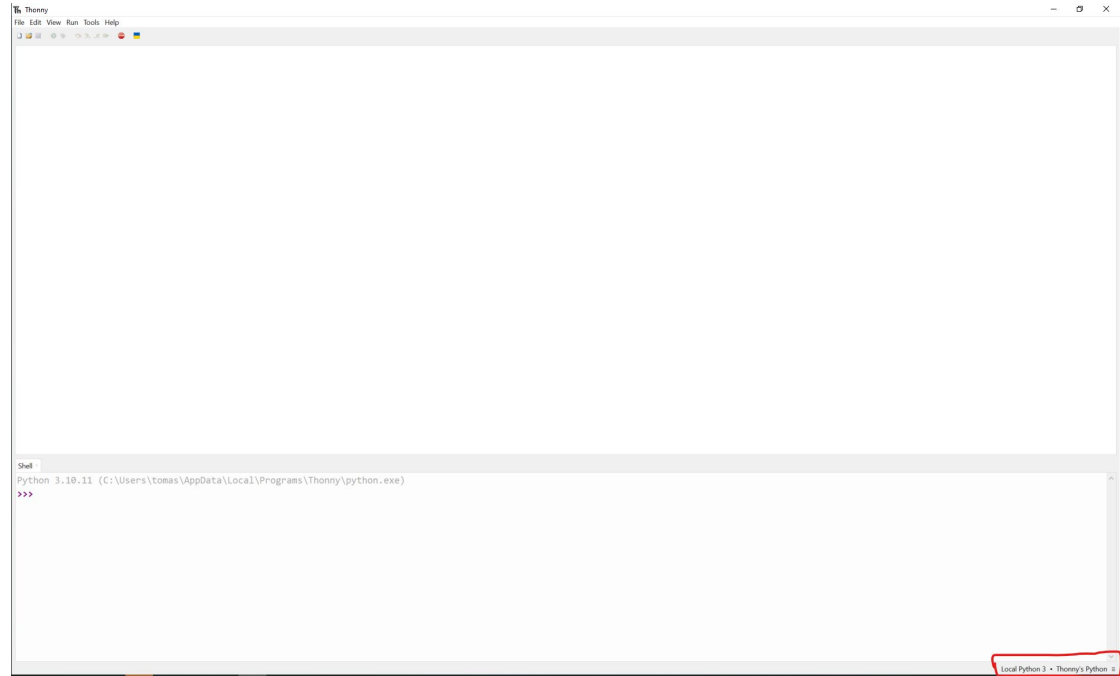
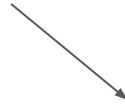
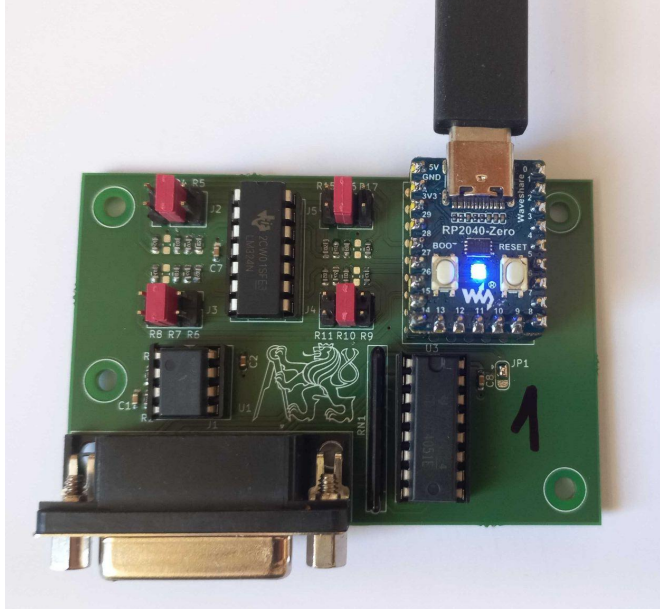
# Crash Course to MicroPython (MP)

- MP = simple Python compiled for embed HW
- All hardware peripherals are objects with certain methods from ``machine`` module
- **Sadly no standard objects (things differ between platforms)**
- For your script to run automatically, it must be named `main.py`
- To edit script loaded in chip's memory we use Thonny IDE



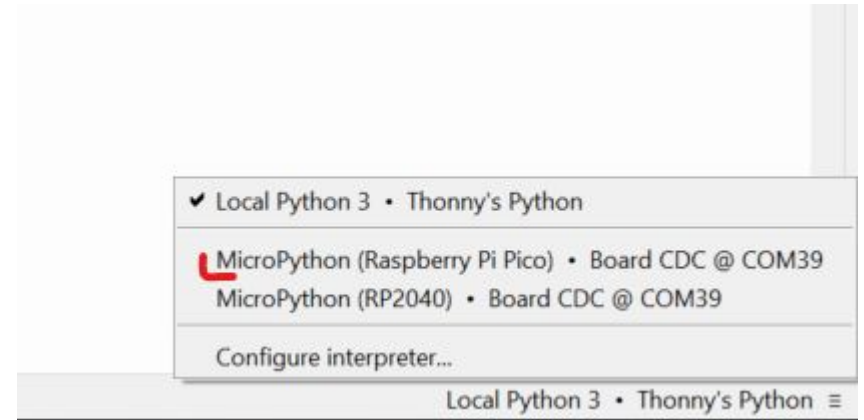
**MicroPython**

# How to edit MP script on HW using Thonny

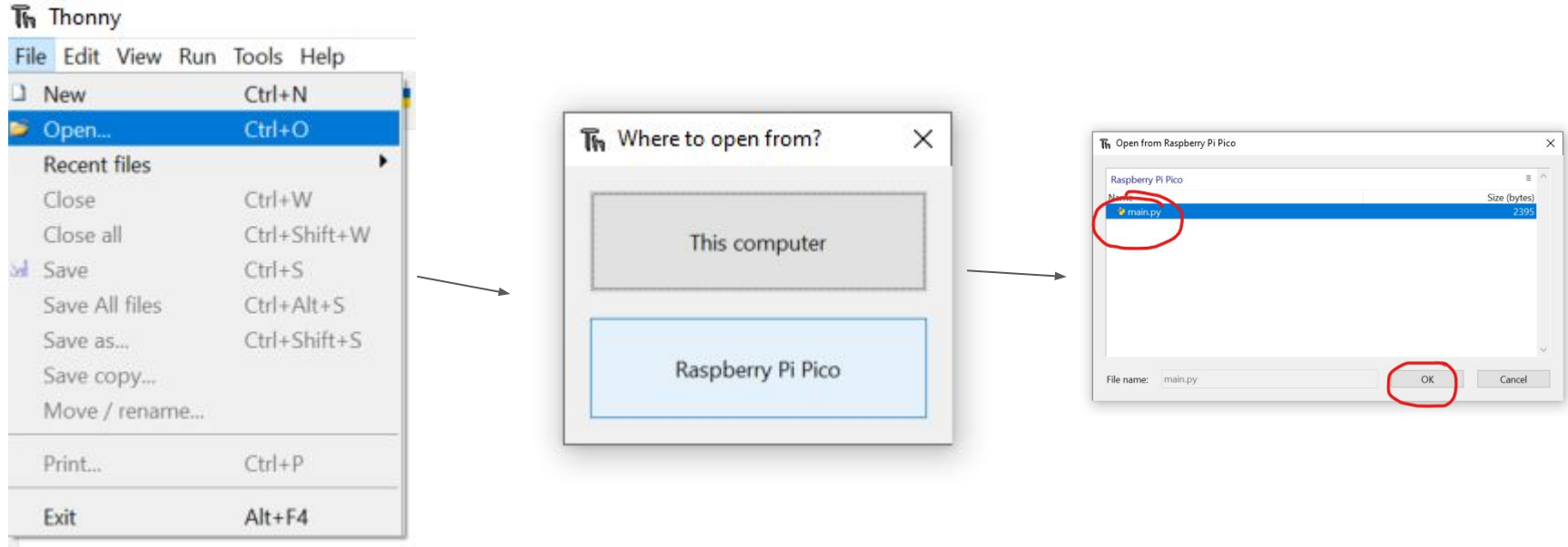


# How to edit MP script on HW using Thonny

Select marked interpreter and note what is RP2040 listed as (COM39 in example), you will need this for Python scripting in your homework

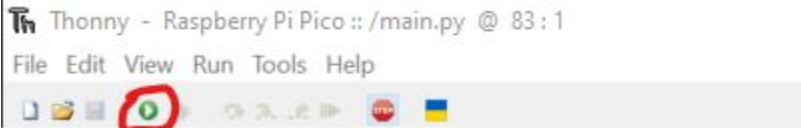


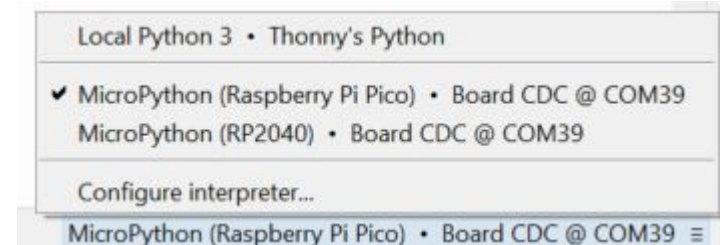
# How to edit MP script on HW using Thonny



To store your progress, just hit Ctrl+s, it will store to RP2040 memory

# How to run MicroPython script

-  Thonny - Raspberry Pi Pico :: /main.py @ 83:1  
File Edit View Run Tools Help  
The screenshot shows the Thonny IDE interface. The title bar reads "Thonny - Raspberry Pi Pico :: /main.py @ 83:1". Below the title bar is a menu bar with "File", "Edit", "View", "Run", "Tools", and "Help". Underneath the menu bar is a toolbar with several icons. The "Run" icon, which is a green play button, is circled in red.
- Press reset button on board (this restarts board and main.py is automatically executed)
  - note resetting board causes it to disconnect from PC and must be manually reconnect even though Thonny shows board being connected
- While running Python script reading data from board (f.e.: example01\_visualization.py) either close Thonny or set it to local interpreter to free serial port



# General Purpose Input Output (GPIO) in MP

- used for: buttons, digital communication, LEDs

```
1 # example code to blink led with pin 14
2 import machine
3 import time
4
5 led = machine.Pin(14, machine.Pin.OUT) # set pin no. 14 as digital output
6 while(true): # infinite loop sou our code never stops
7     led.value(true) # drive output pin High (light on)
8     time.sleep(0.1) # sleep for 100ms
9     led.value(false) # drive output pin Low (light off)
10    time.sleep(0.1) # sleep for 100ms
```

## Read analog values in MP

$$V = \frac{n}{2^N - 1} V_{ref}$$

- for some weird reason ADCs in MP return 16 bit numbers with maximum of 65535 even if given ADC is 12 bit

$n$  → ADC output number  
 $N$  → ADC bit resolution (**16 bit** even though ADC is only 12 bit)  
 $V_{ref}$  → ADC reference (**3.3V**)

```
1 # periodically reads voltage values from 4 analog inputs
2 import machine
3 import time
4
5 adcs = [] # array of ADCs objects
6 adcs.append(machine.ADC(3)) # add adc channel into a list
7 adcs.append(machine.ADC(0))
8 adcs.append(machine.ADC(1))
9 adcs.append(machine.ADC(2))
10
11 voltage = [0,0,0,0] # prepare space for masured voltage
12 while(true):
13     for i in range(4):
14         voltage[i] = adcs[i].read_u16()/65535*3.3 # read analog value and convert to voltage
15         time.sleep(10e-3) # sleep for 10ms between samples
```

# Bibliography

- M. Hopkins, R. Vaidyanathan and A. H. McGregor, "Examination of the Performance Characteristics of Velostat as an In-Socket Pressure Sensor," in *IEEE Sensors Journal*, vol. 20, no. 13, pp. 6992-7000, 1 July 1, 2020, doi: 10.1109/JSEN.2020.2978431. keywords: {Loading;Sensor phenomena and characterization;Sockets;Prosthetics;Temperature sensors;Mechanical sensors;Piezoresistive measurement;pressure sensing;prosthetics;prosthetic fitting;Velostat;wearable sensors}
- Dzedzickis A, Sutinyas E, Bucinskas V, Samukaite-Bubniene U, Jakstys B, Ramanavicius A, Morkvenaite-Vilkonciene I. Polyethylene-Carbon Composite (Velostat®) Based Tactile Sensor. *Polymers (Basel)*. 2020 Dec 3;12(12):2905. doi: 10.3390/polym12122905. PMID: 33287414; PMCID: PMC7761878.
- Tang, R., Lu, F., Liu, L., Yan, Y., Du, Q., Zhang, B., Zhou, T., & Fu, H. (2021). Flexible pressure sensors with microstructures. In *Nano Select* (Vol. 2, Issue 10, pp. 1874–1901). Wiley. <https://doi.org/10.1002/nano.202100003>
- Proesmans, R.; Verleysen, A.; Vleugels, R.; Veske, P.; De Gusseme, V.-L.; Wyffels, F. Modular Piezoresistive Smart Textile for State Estimation of Cloths. *Sensors* **2022**, 22, 222. <https://doi.org/10.3390/s22010222>
- VIDAL-VERDÚ, Fernando, et al. Three realizations and comparison of hardware for piezoresistive tactile sensors. *Sensors*, 2011, 11.3: 3249-3266.