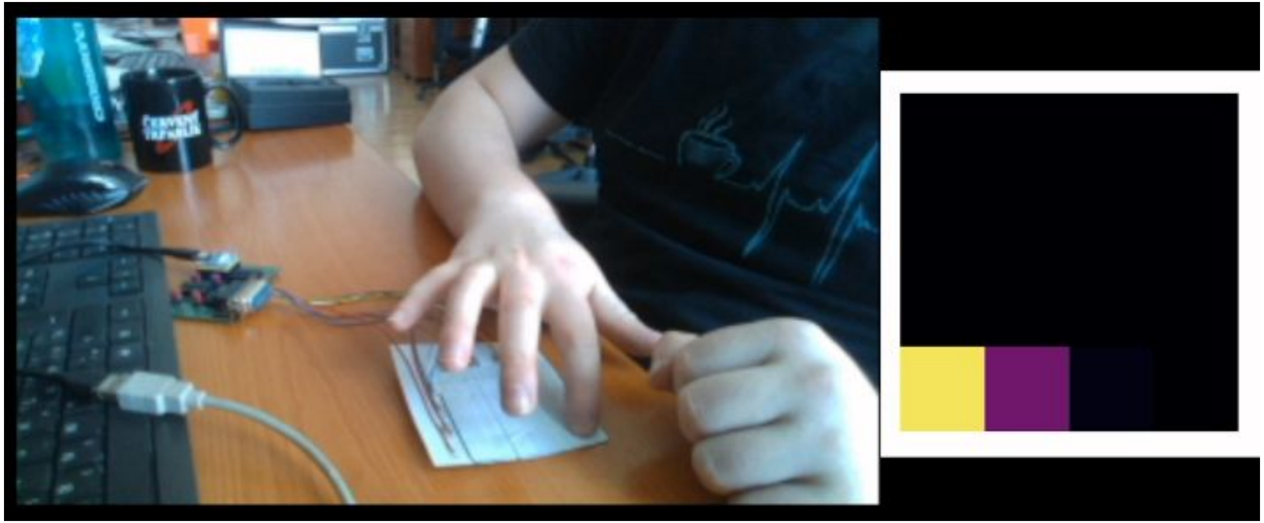
e-skin lab @ Humanoids

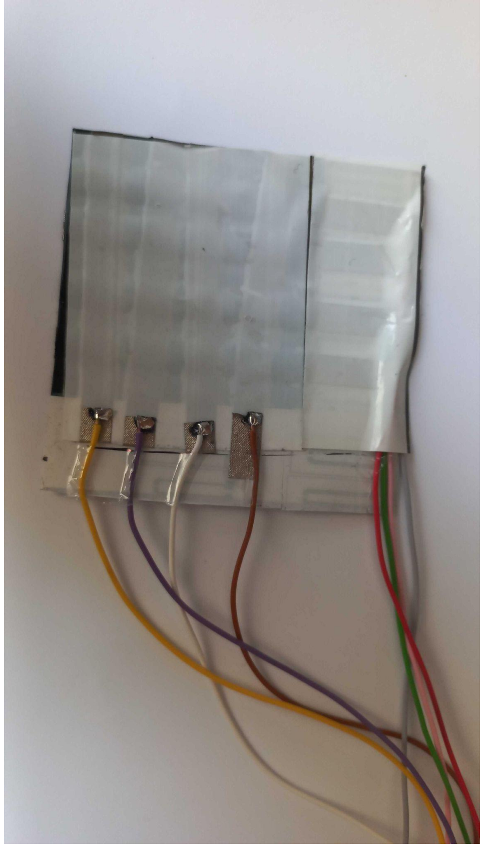
How to read these slides

- 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



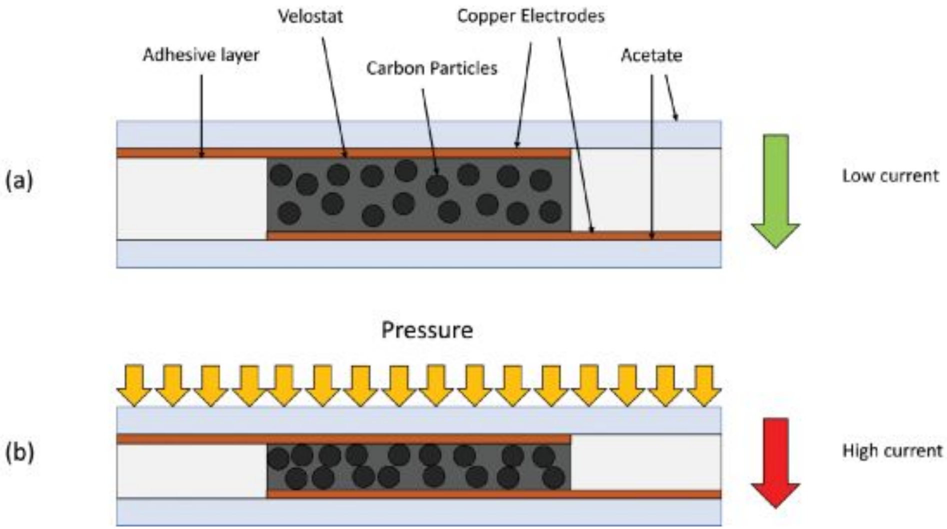


Velostat

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

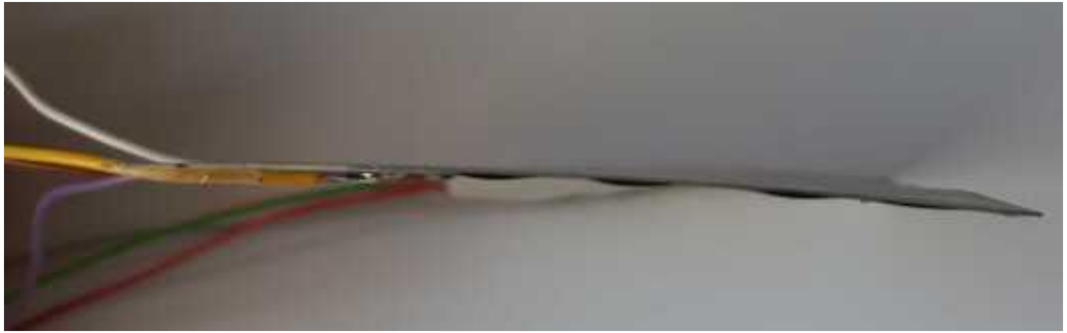
Velostat as a sensor

- 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**



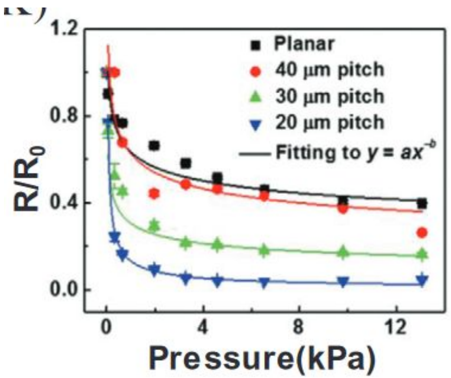
Additional sources of piezoresistivity - Surface effects

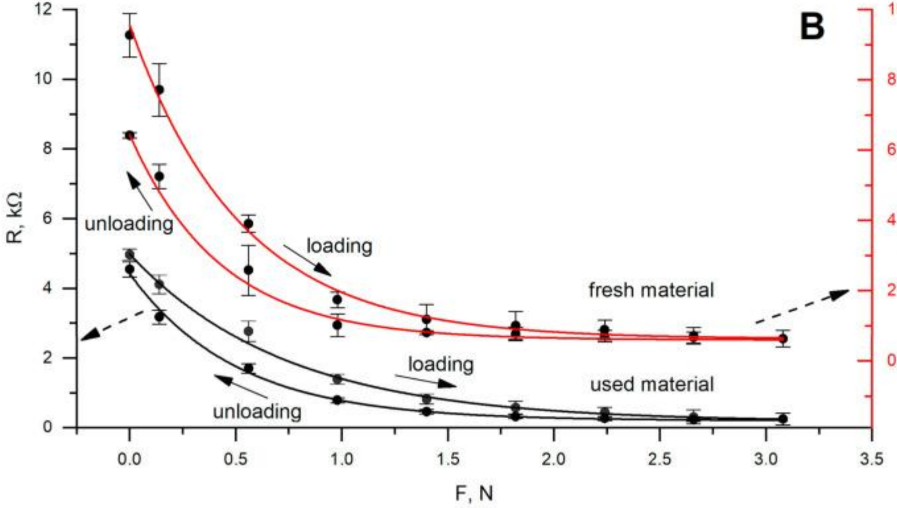
- Skin patch is warped
  - with pressure contact area between velostat and electrodes increases
- Roughness of surfaces
  - with pressure microscopic structures are deformed increasing contact area



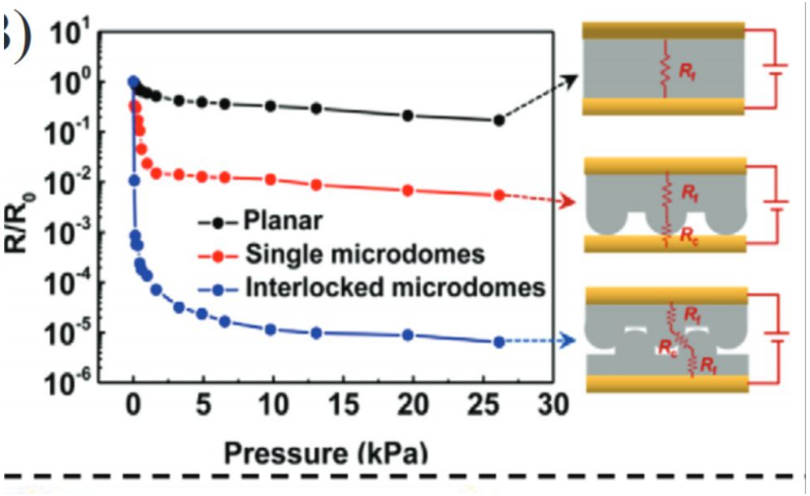
Typical response

- Generally shown as R to pressure or force
- Typically relative resistance R/R0
- 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

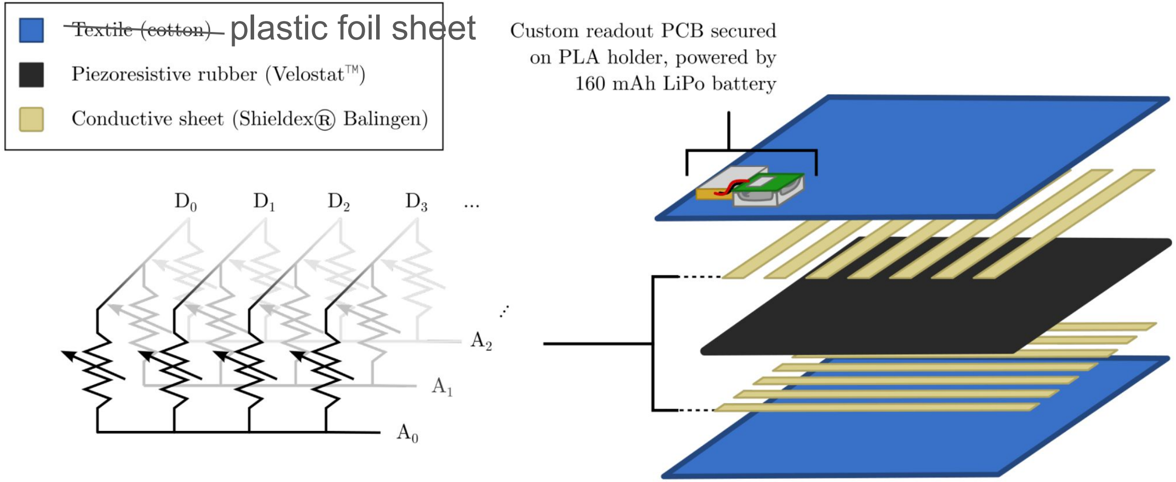
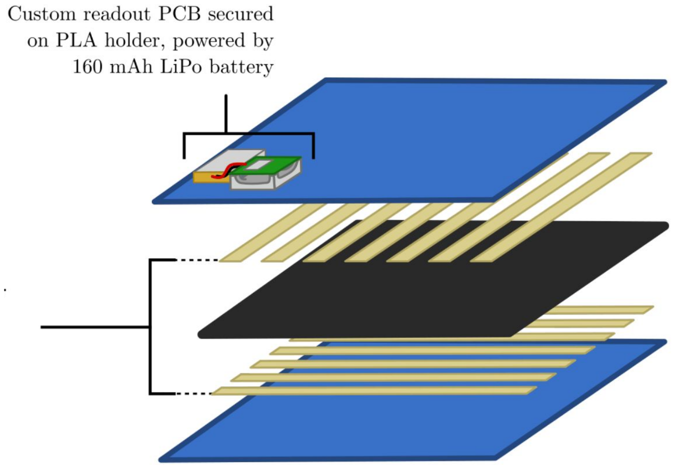

🟦 ~~Textile (cotton)~~ plastic foil sheet
⬛ Piezoresistive rubber (Velostat™)
🟨 Conductive sheet (Shieldex® Balingen)
Custom readout PCB secured on PLA holder, powered by 160 mAh LiPo battery
D_0
D_1
D_2
D_3
...
A_2
A_1
A_0

Image illustrative, taken from literature

Construction - Resistive array

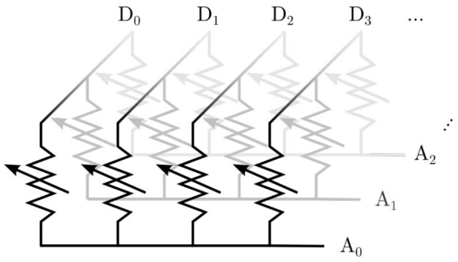
- With certain precision we can model e-skin as follows:
  - Where row and column electrodes overlap a resistor is formed
  - Each resistor is, thanks to velostat, piezoresistive and forms one tactile cell (taxel) as shown in schema



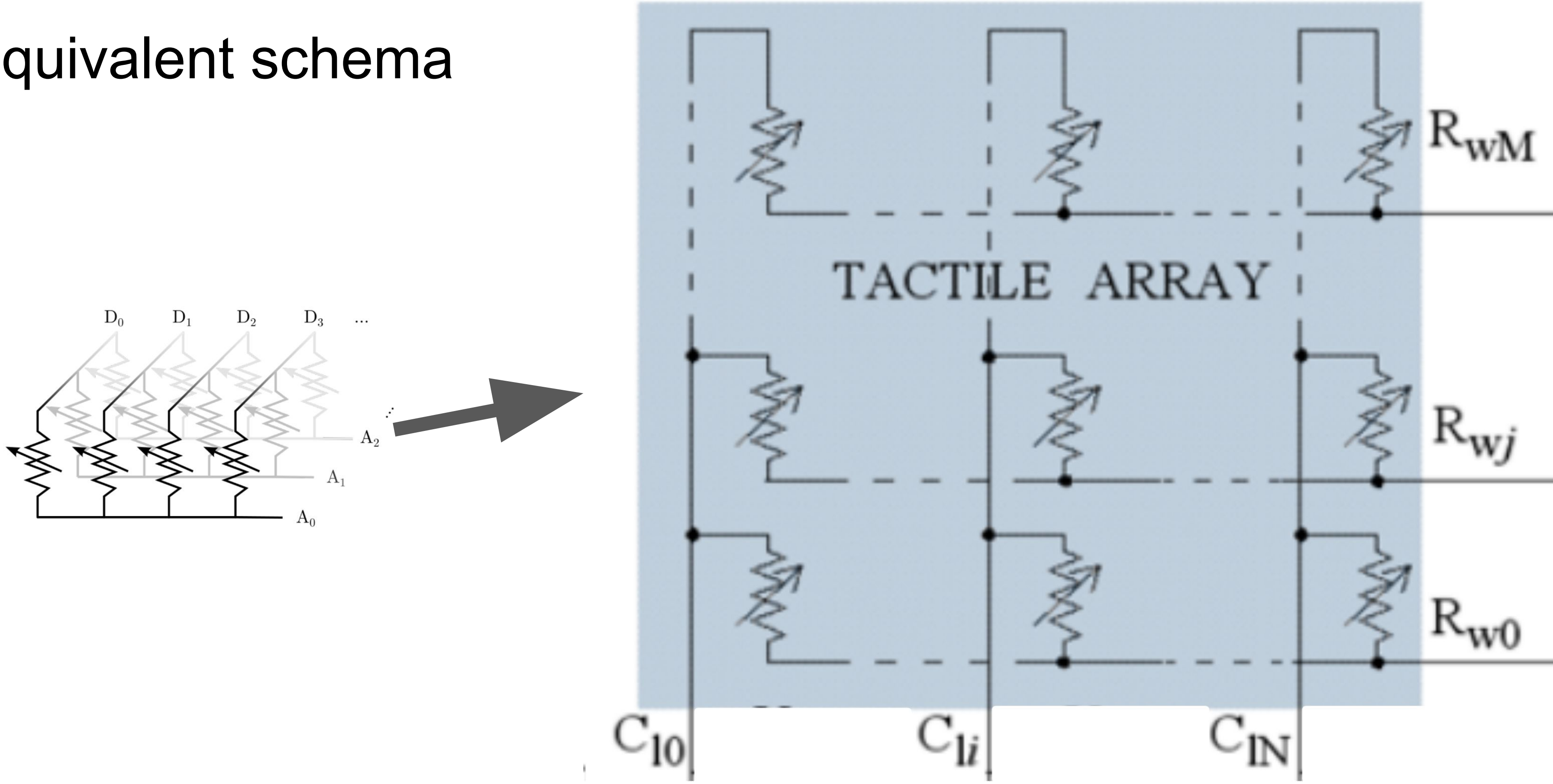
🟦 ~~Textile (cotton)~~ plastic foil sheet

⬛ Piezoresistive rubber (Velostat™)

🟨 Conductive sheet (Shieldex® Balingen)

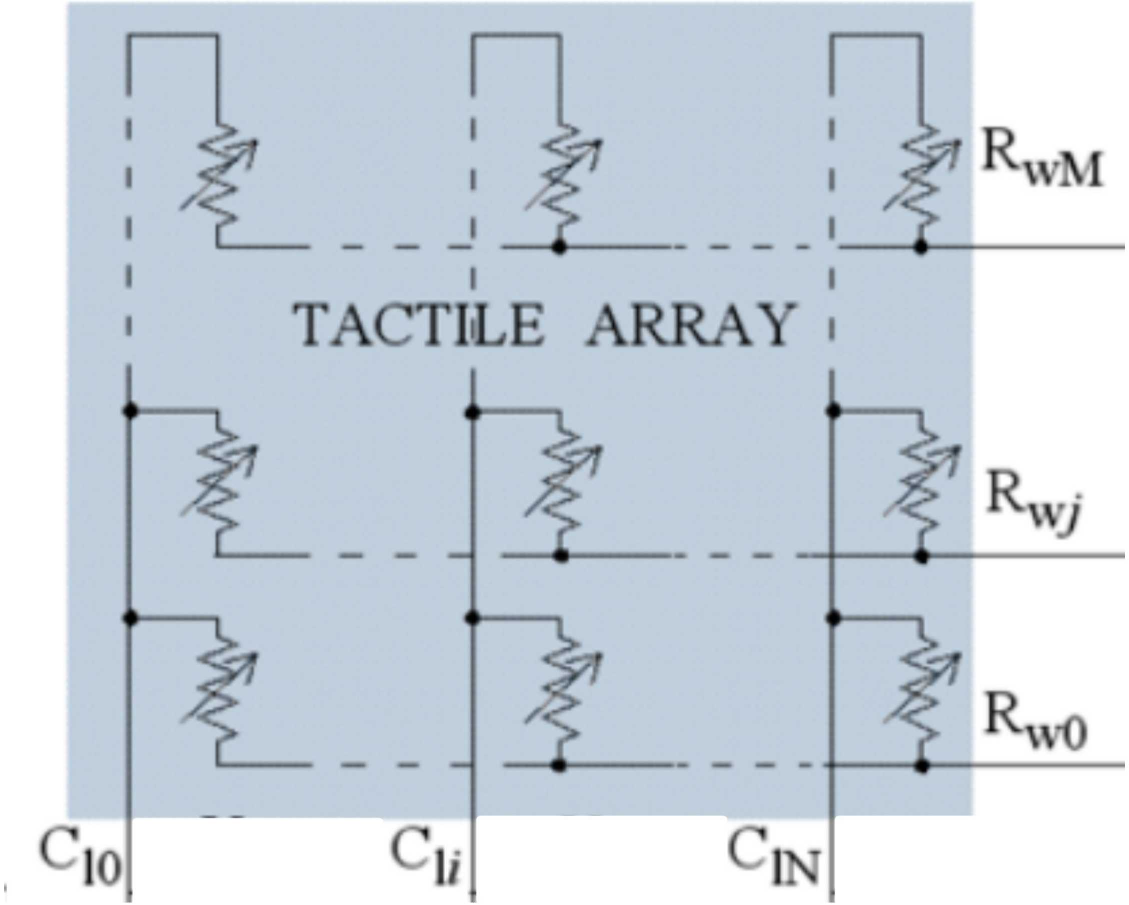


Equivalent schema



Equivalent schema

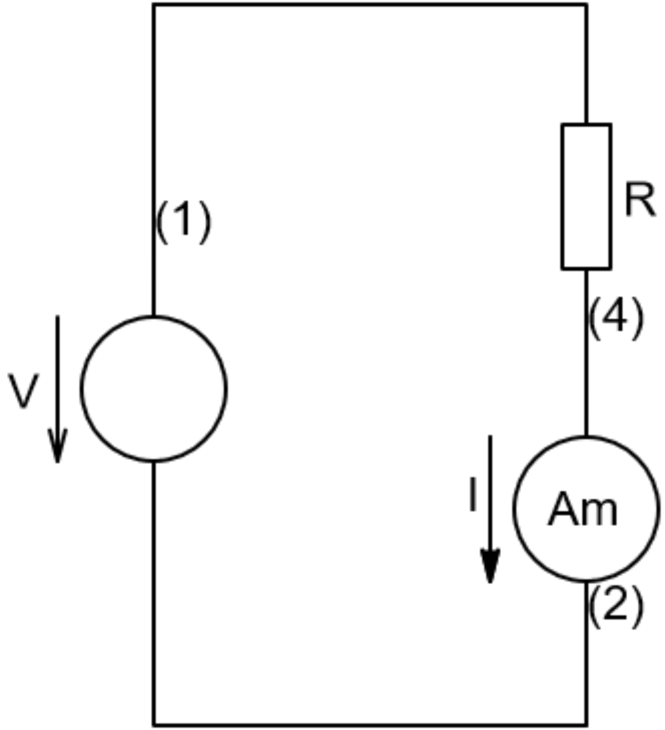
- We just re-drew the schema from 3D to 2D
- Similar to keyboard or memories



now…how to measure resistance?

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

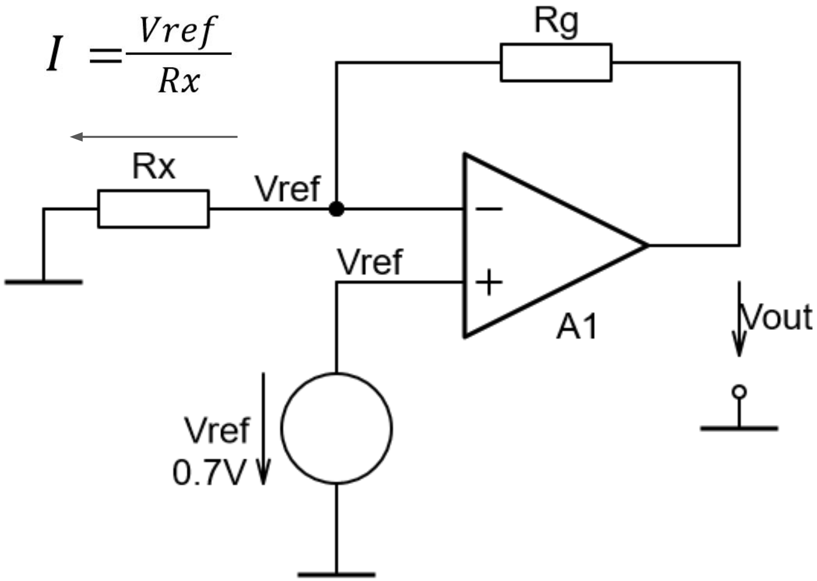
$$R = \frac{V}{I}$$



How do we measure resistance without an Amp. meter

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

$$I = \frac{Vref}{Rx}$$

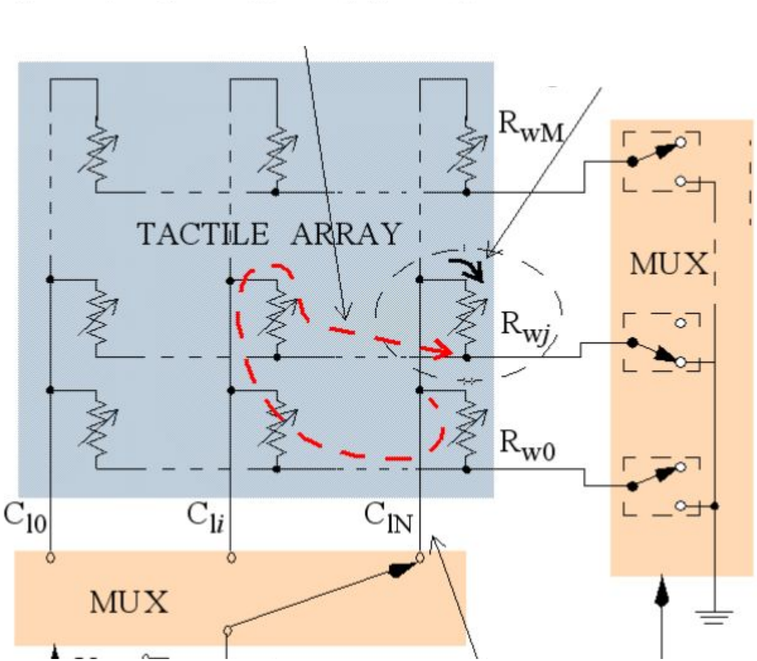


$$Vout = \left(\frac{Rg}{Rx} + 1\right)Vref$$

Rx → Measured resistor
Rg → Known (gain) resistor
Vref → Voltage reference **(0.7V)**

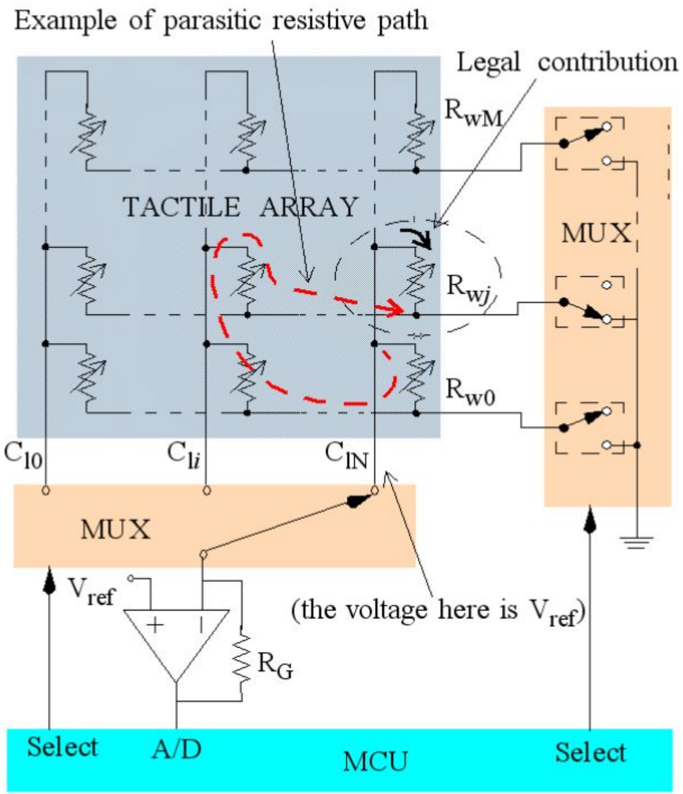
How to measure resistive array?

- Well, first we need to address individual resistor
- Let’s try apply voltage to column CIN and row Rwj.
- To do this, let’s use some electronic switches (multiplexers, MUX)



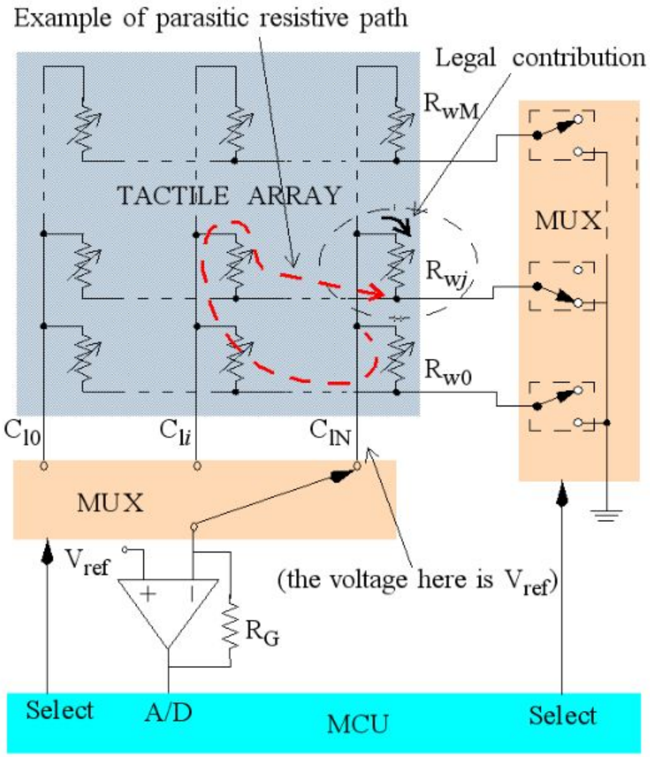
How to measure resistive array?

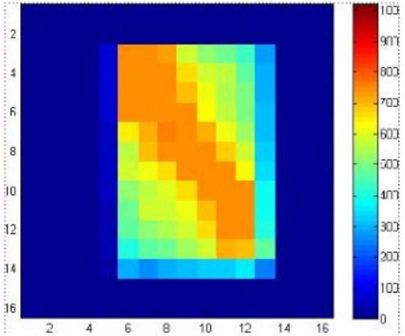
- Now we can add operational amplifier and use switches to switch between resistors
- But wait what?
  - when we pass current through resistor there is also a current marked with red dashed line!
  - WHY?! well simply because current passes wherever it can and nothing prevents it passing through all the resistors, even though we applied voltage only to CIN and Rwj.



Crosstalk currents

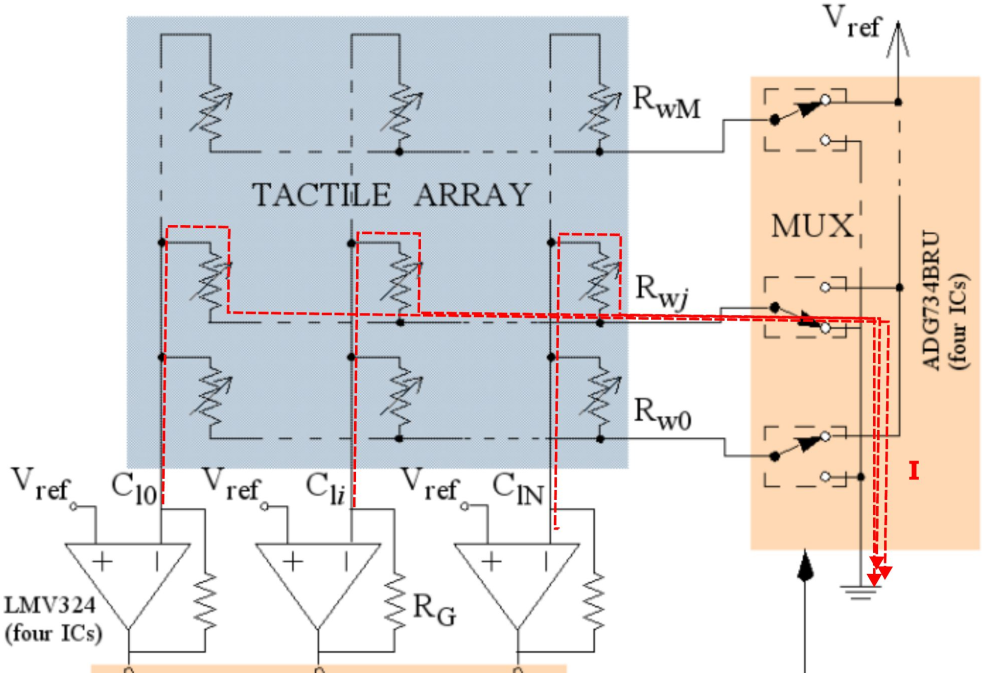
- For reason on previous slide when we try to measure only one resistor the measurement is influenced by all the resistors in the array
- These unwanted currents are called crosstalk currents.
- The result of them is very blurred tactile image
- so what can we do about it?





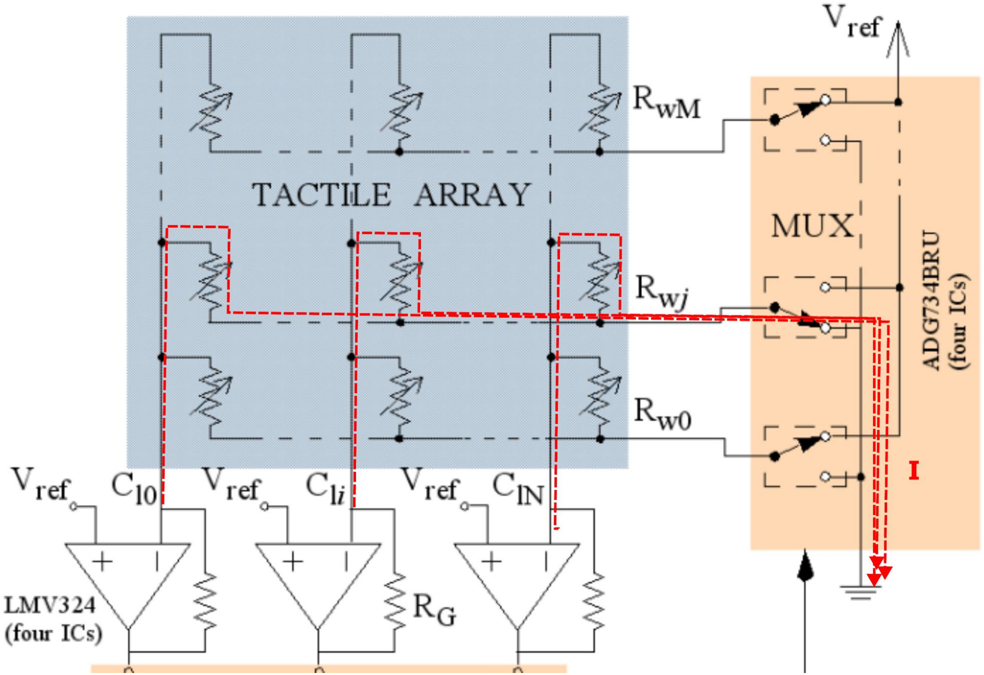
Crosstalk current compensation

- 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 Vref and GND signals.



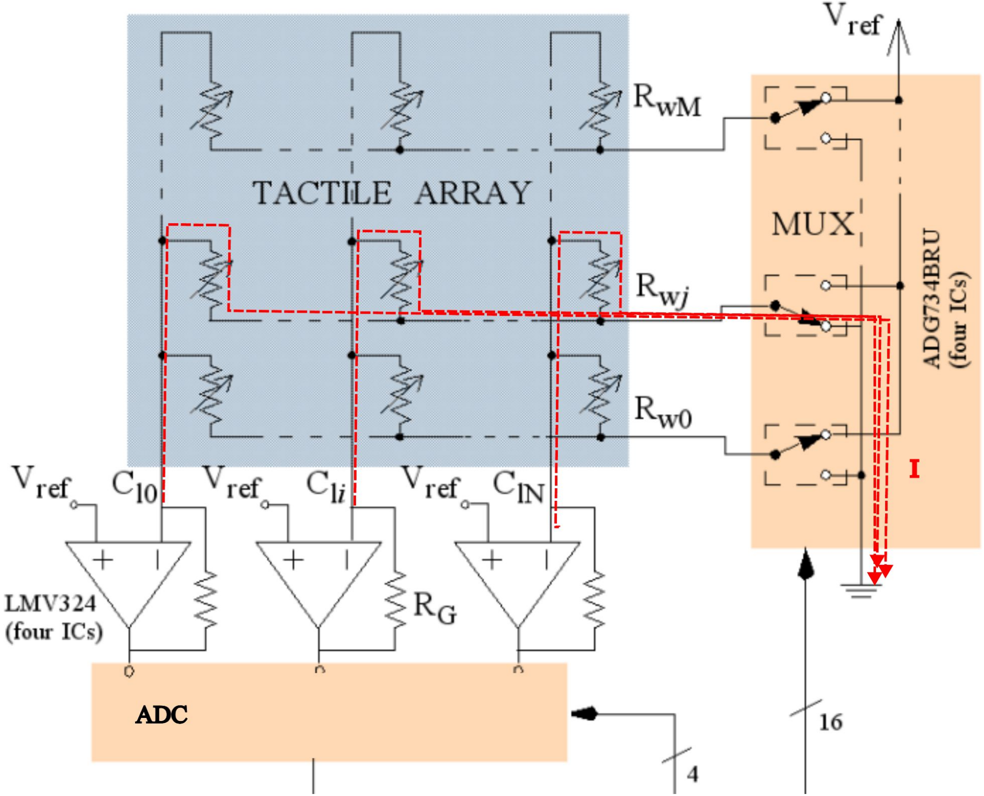
Crosstalk current compensation

- 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 Vref 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**



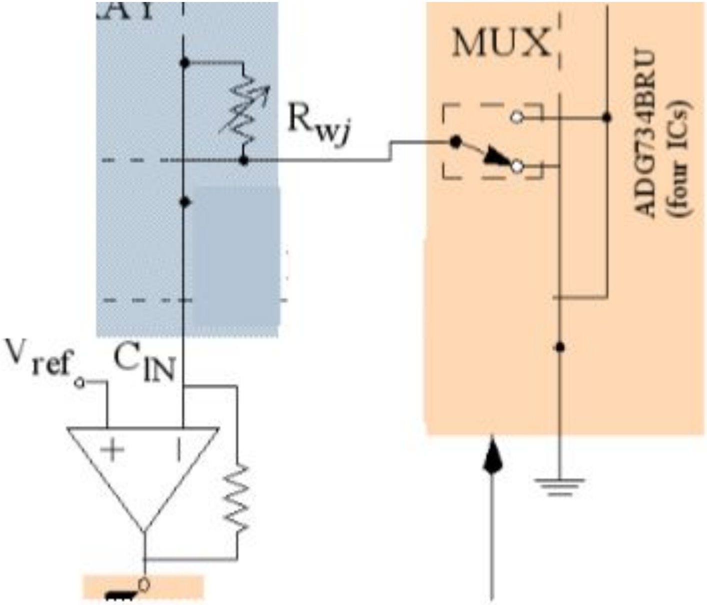
Crosstalk current compensation

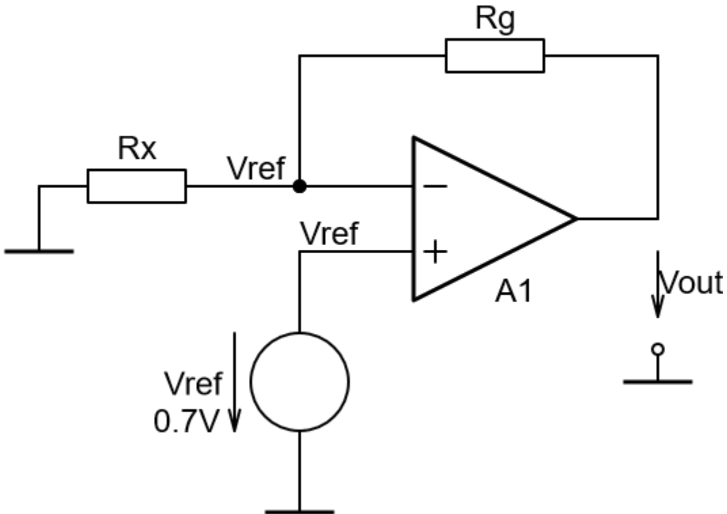
- To get resistance value for individual resistors in the array we will measure each operational amplifier output voltage using ADC.
- even though current passes through all the resistors in the row, these currents do not influence each other and therefore we can pretend we measure each resistor individually.



Crosstalk current compensation

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

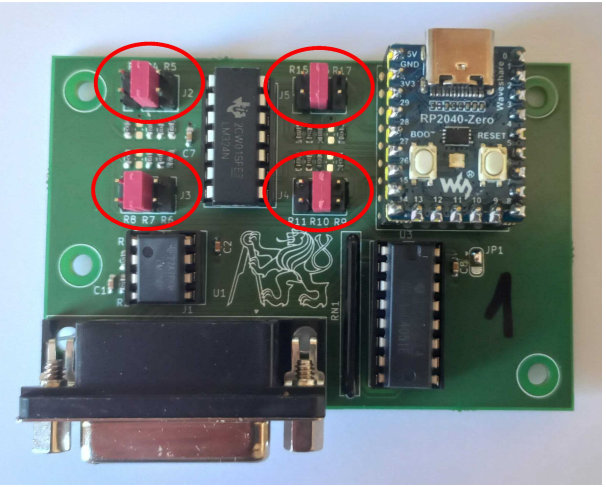


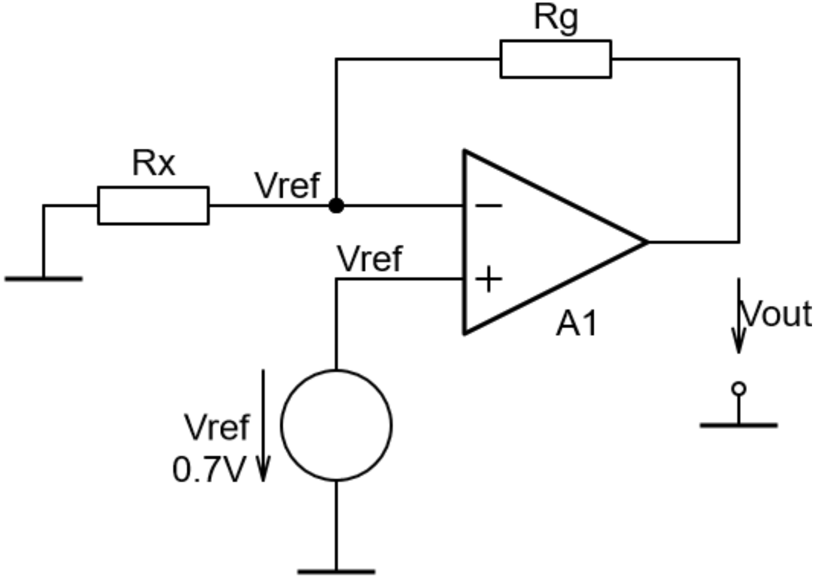


$$Vout = \left(\frac{Rg}{Rx} + 1\right)Vref$$

How do we get Rg?

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



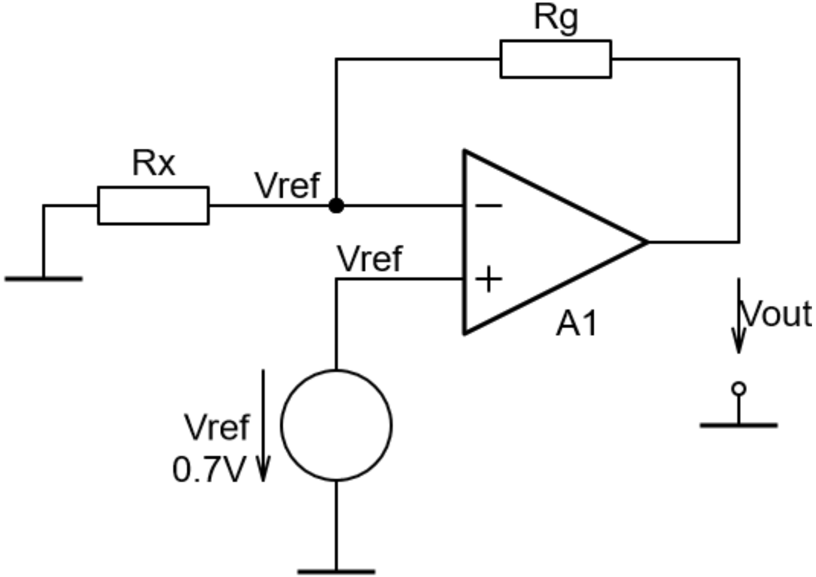


$$Vout = \left(\frac{Rg}{Rx} + 1\right)Vref$$

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`

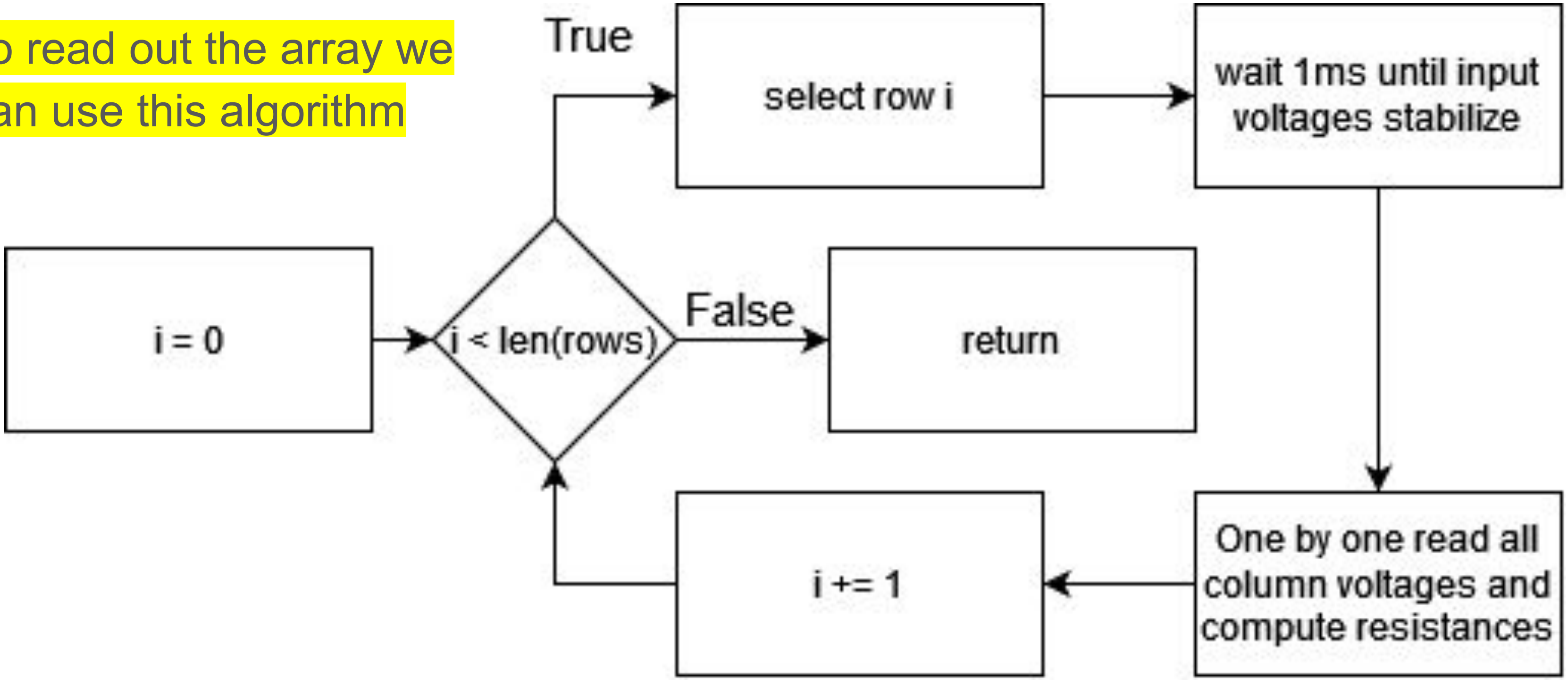


$$Vout = \left(\frac{Rg}{Rx} + 1\right)Vref$$

Rx → Measured resistor
Rg → Known (gain) resistor
Vref → Voltage reference **(0.7V)**

Read out algorithm and timing

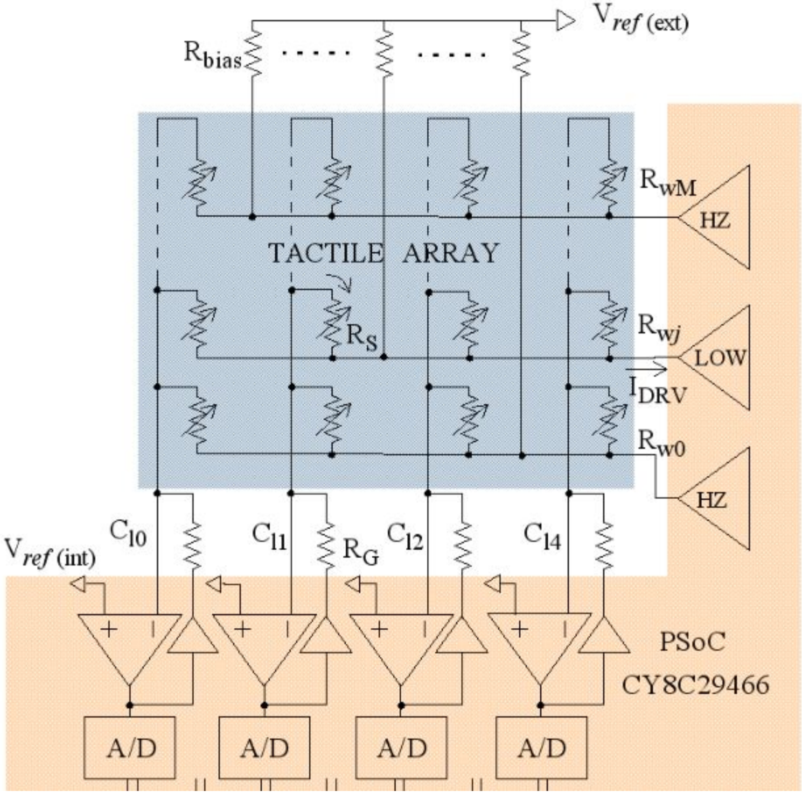
- To read out the array we can use this algorithm



i = 0
i < len(rows)
True
select row i
wait 1ms until input voltages stabilize
One by one read all column voltages and compute resistances
i += 1
False
return

Real Implementation

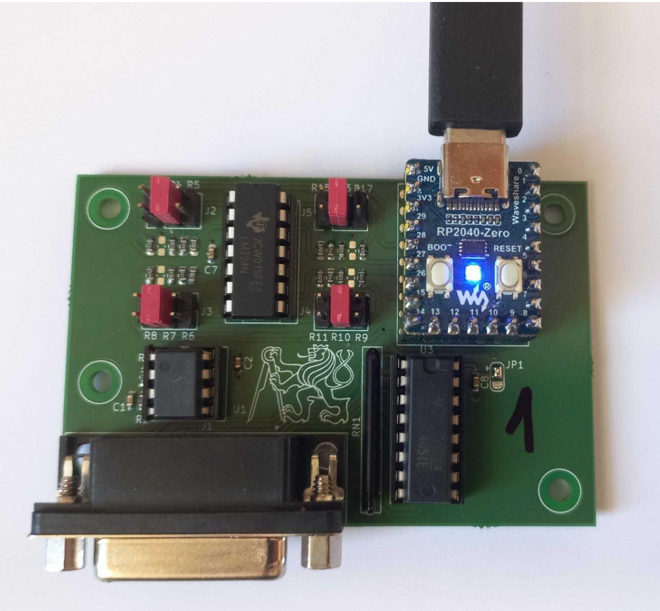
- Pull-up resistors to Vref
- Mux switches between GND and High Impedance
- Each op amp sampled by analog to digital converter

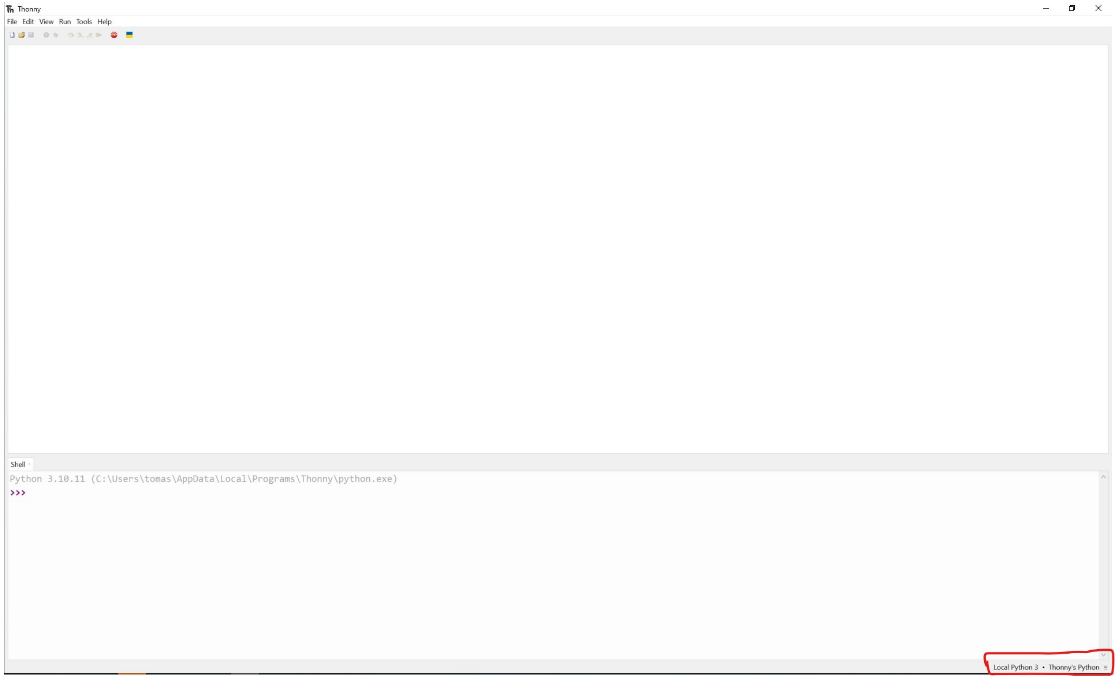


Crash Course to MicroPython (MP)

- 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

How to edit MP script on HW using Thonny



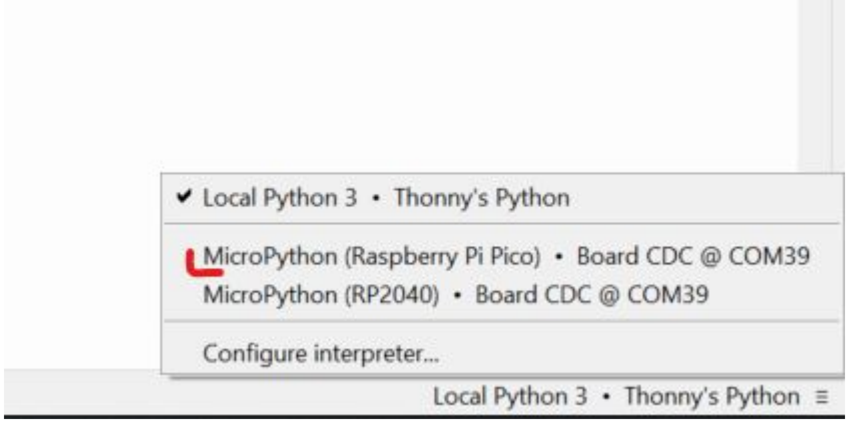


```
Python 3.10.11 (C:\Users\tomas\AppData\Local\Programs\Thonny\python.exe)
>>>
```

Local Python 3 • Thonny's Python

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



✔ Local Python 3 • Thonny's Python

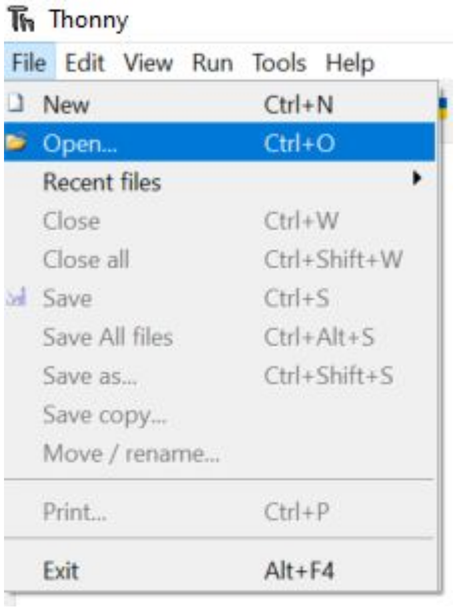
MicroPython (Raspberry Pi Pico) • Board CDC @ COM39

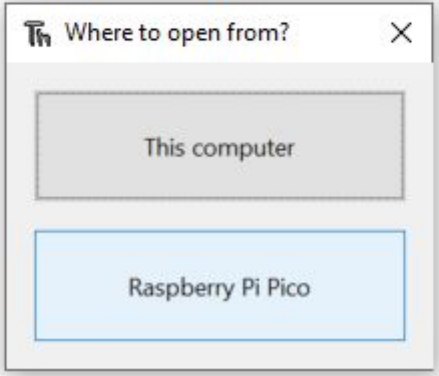
MicroPython (RP2040) • Board CDC @ COM39

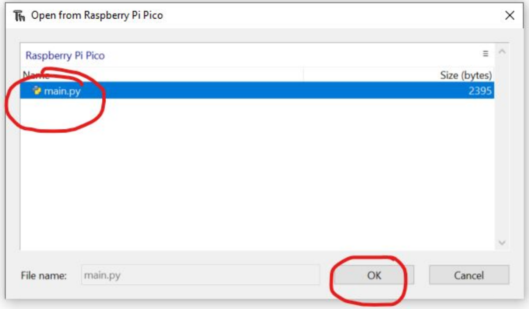
Configure interpreter...

Local Python 3 • Thonny's Python ≡

How to edit MP script on HW using Thonny

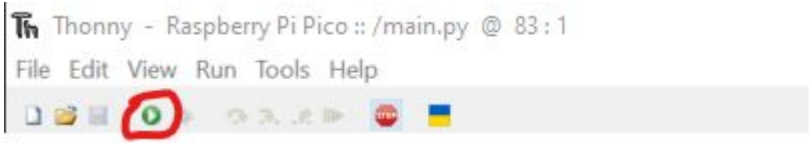






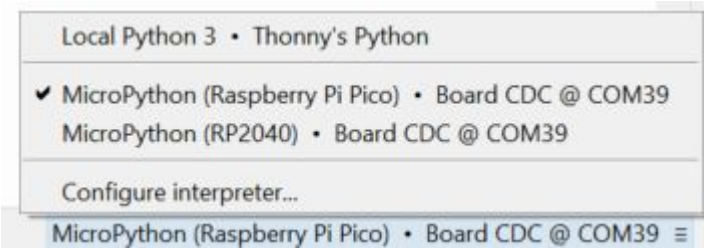
To store your progress, just hit Ctrl+s, it will store to RP2040 memory

How to run MicroPython script

- 

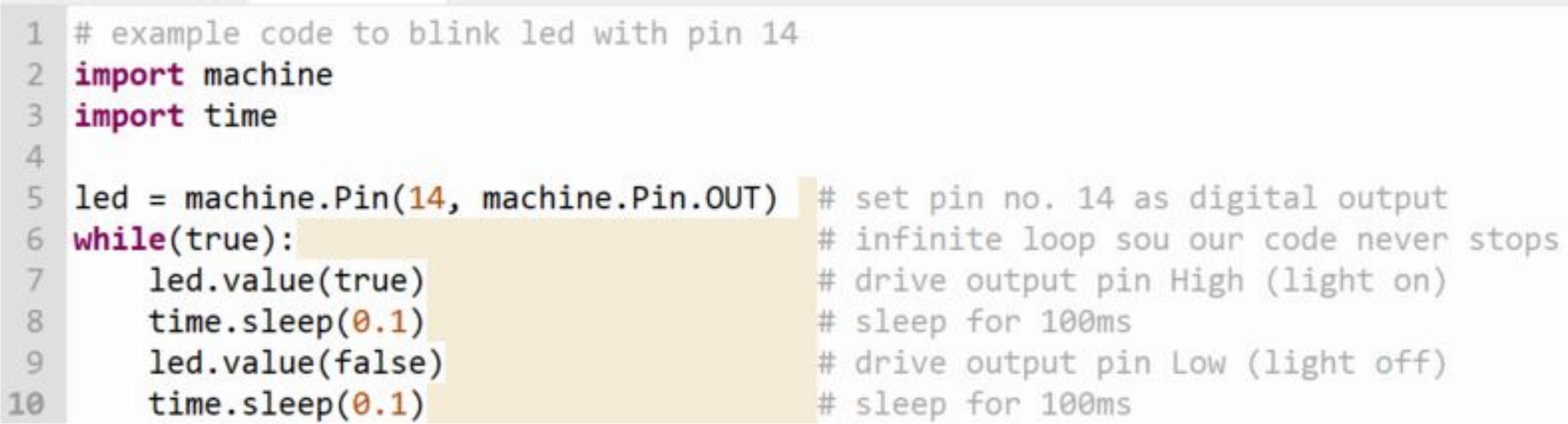
- Press reset button on board (this restarts board and main.py is automatically executed)
    - note resetting board causes it to disconnect from PC and must be manually reconnect even though Thonny shows board being connected

- While running Python script reading data from board (f.e.: example01_visualizaton.py) either close Thonny or set it to local interpreter to free serial port


Local Python 3 • Thonny's Python
✔ MicroPython (Raspberry Pi Pico) • Board CDC @ COM39
MicroPython (RP2040) • Board CDC @ COM39
Configure interpreter...
MicroPython (Raspberry Pi Pico) • Board CDC @ COM39 ≡

General Purpose Input Output (GPIO) in MP

- used for: buttons, digital communication, LEDs



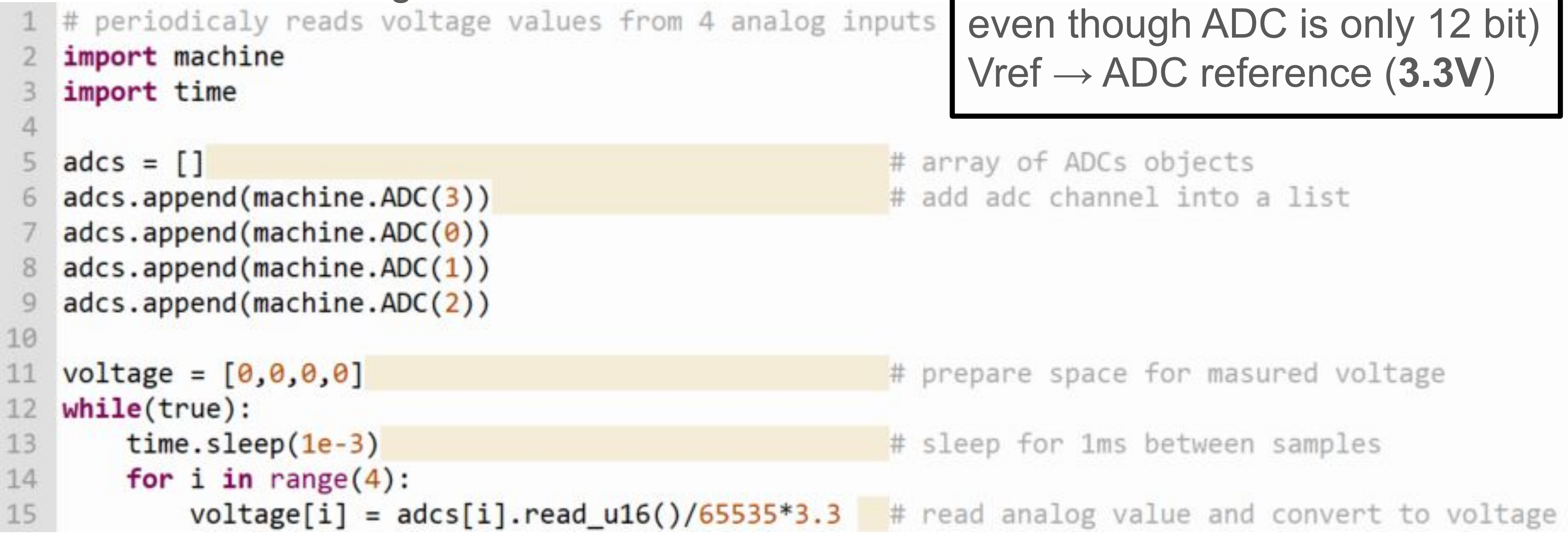
```
# example code to blink led with pin 14
import machine
import time

led = machine.Pin(14, machine.Pin.OUT)  # set pin no. 14 as digital output
while(true):                            # infinite loop sou our code never stops
    led.value(true)                     # drive output pin High (light on)
    time.sleep(0.1)                     # sleep for 100ms
    led.value(false)                    # drive output pin Low (light off)
    time.sleep(0.1)                     # sleep for 100ms
```

Read analog values in MP

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

- 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)
Vref → ADC reference (**3.3V**)



```
# periodicaly reads voltage values from 4 analog inputs
import machine
import time

adcs = []                                       # array of ADCs objects
adcs.append(machine.ADC(3))                     # add adc channel into a list
adcs.append(machine.ADC(0))
adcs.append(machine.ADC(1))
adcs.append(machine.ADC(2))

voltage = [0,0,0,0]                             # prepare space for masured voltage
while(true):
    time.sleep(1e-3)                            # sleep for 1ms between samples
    for i in range(4):
        voltage[i] = adcs[i].read_u16()/65535*3.3   # read analog value and convert to voltage
```

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 July1, 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, Sutinys 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.