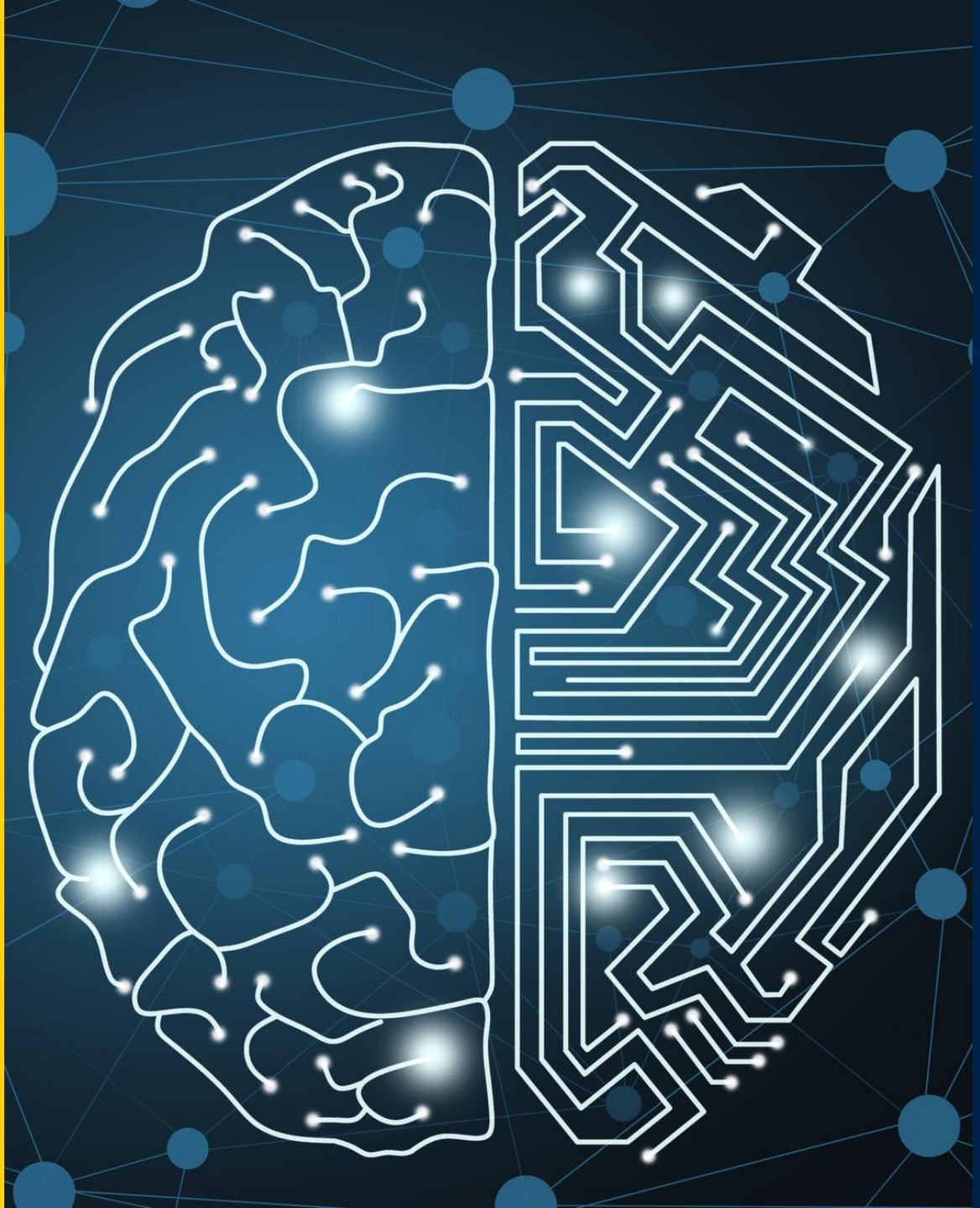
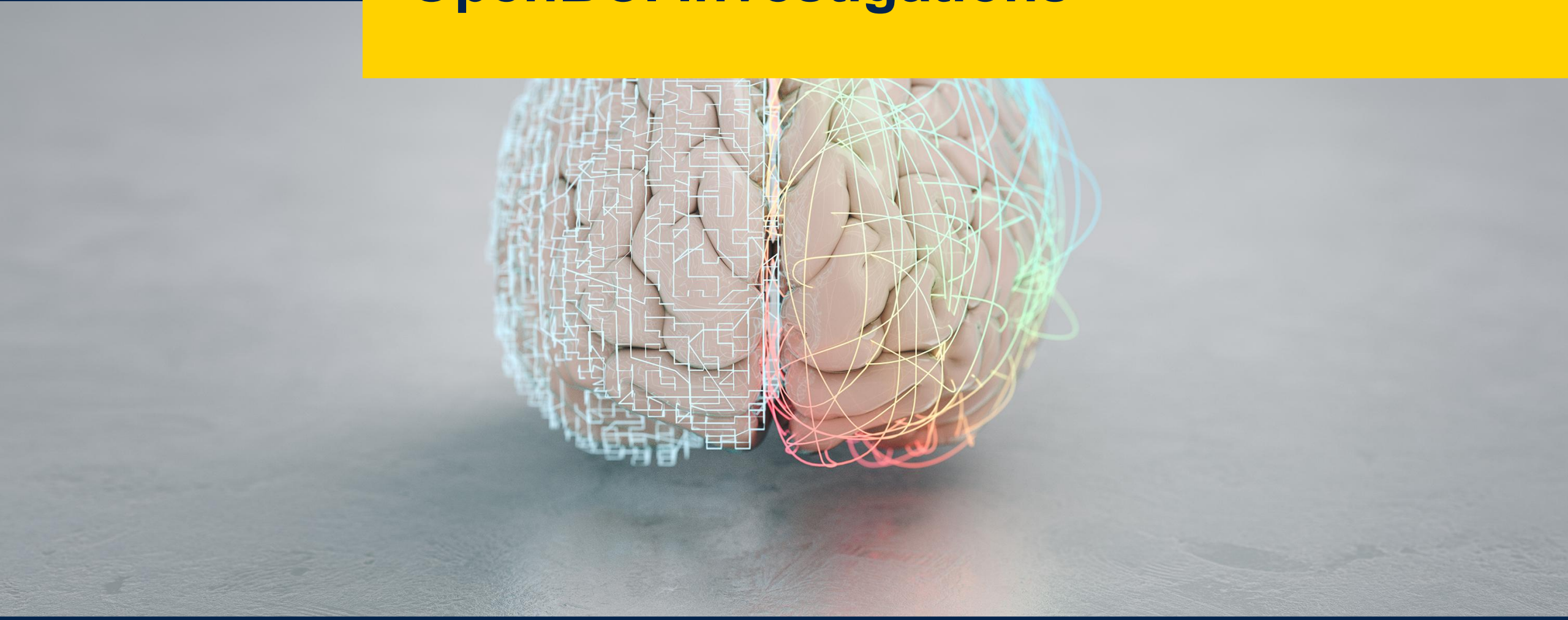


BCI e robotica: come la mente controlla la macchina

Vincenzo Mazzullo (AI Tools and Frameworks Architecture Specialist)

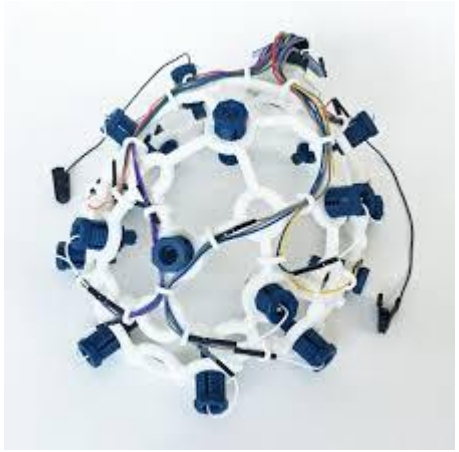
SRA AI SW & Tools
STMicroelectronics

OpenBCI Investigations



Starting point: OpenBCI project

3D printable Headset (up to 16 channels)



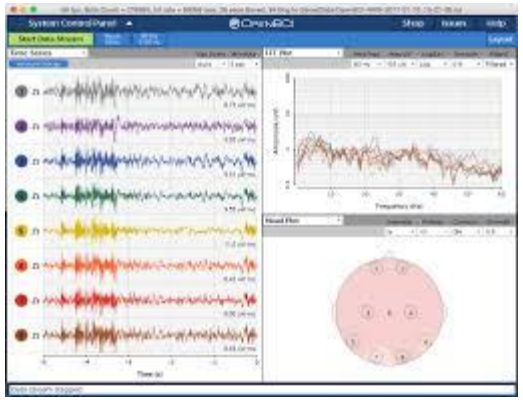
+

Open platform Arduino based

Custom board with STM32

+

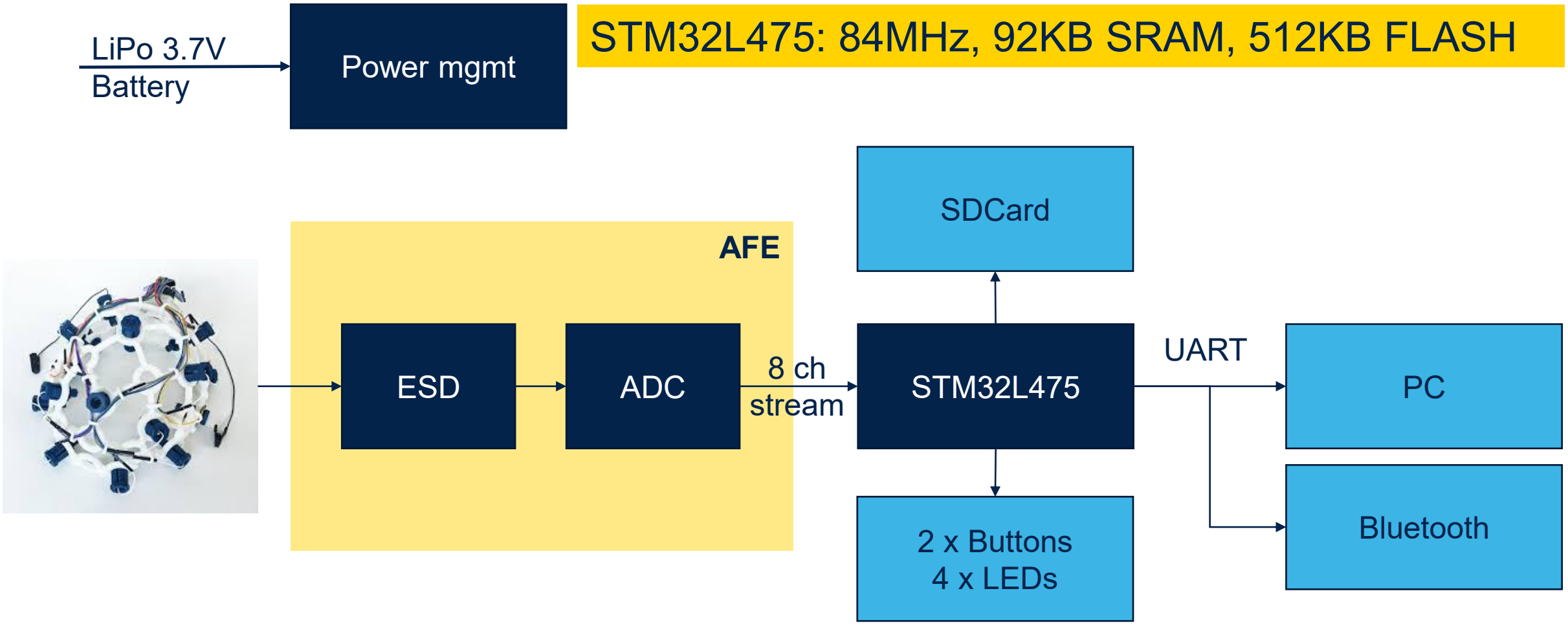
SW for processing the signals



SW for data logging

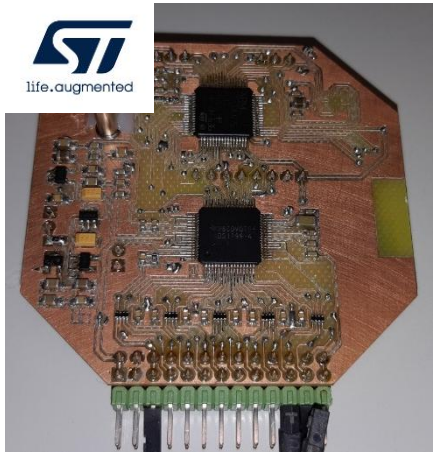
=

Custom board with STM32 microcontroller

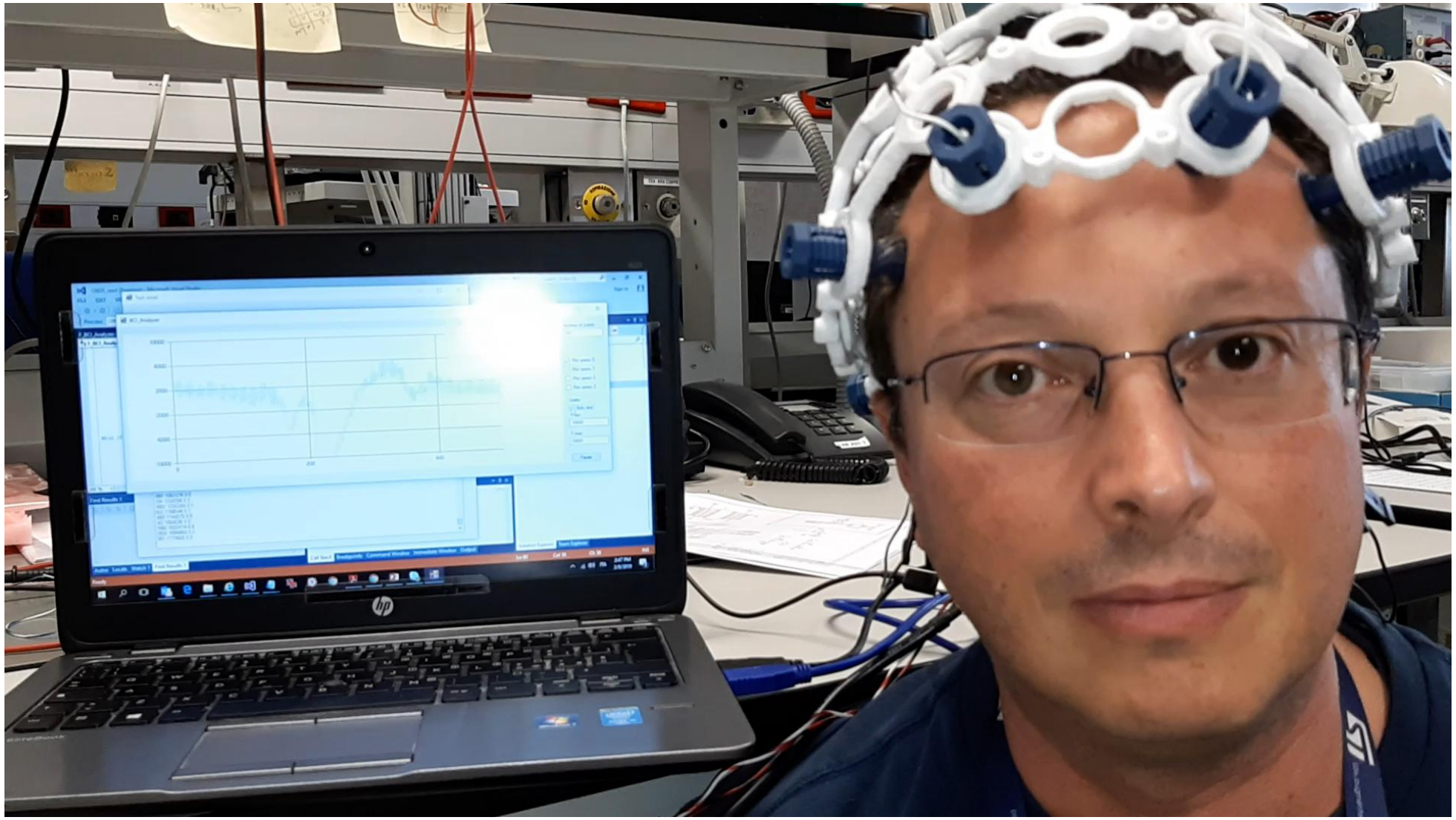


Software: Data logging

Headware



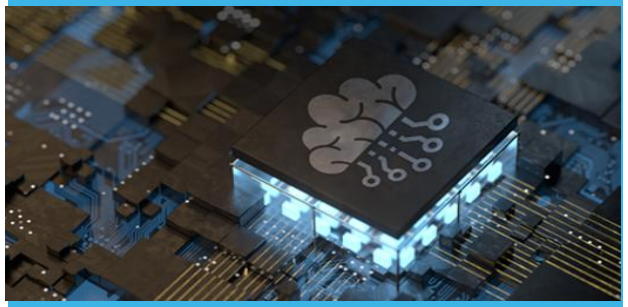
Hardware+Firmware



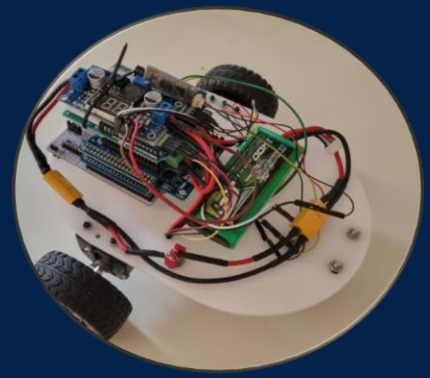
Data acquisition

Brain computer interface for robotic control

READ SIGNALS FROM THE BRAIN

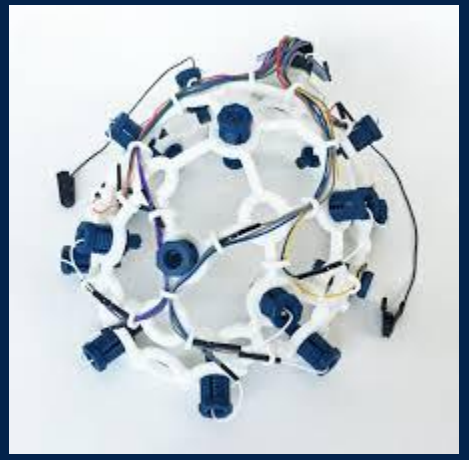


SIGNAL PROCESSING AND MACHINE LEARNING



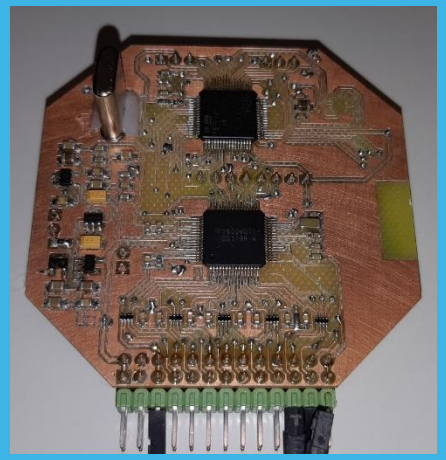
SEND COMMANDS TO A ROBOT

First Work: The hardware

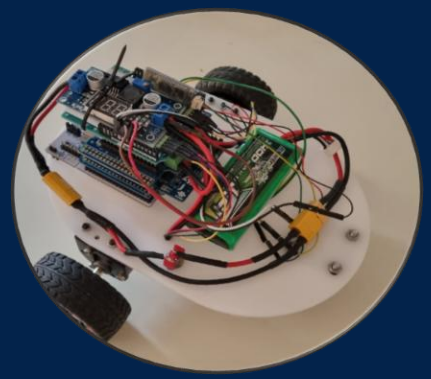


HELMET FROM THE OPEN BCI PROJECT

https://openbci.com/

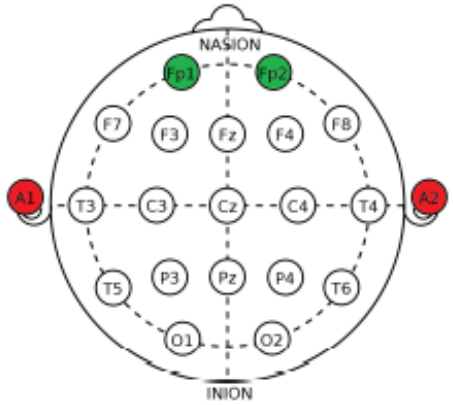


CUSTOM BOARD WITH AN STM32 MICROCONTROLLER



WILL-E: 3 WHEELED ROBOTIC PLATFORM

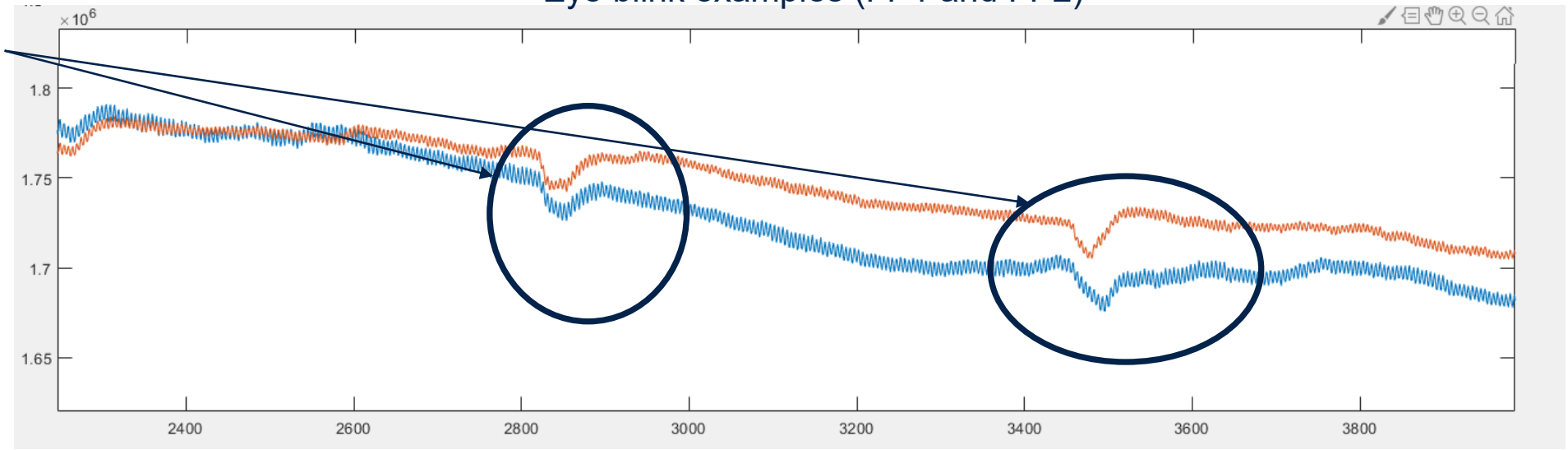
Machine learning on Electro-Oculography (EOG) signals



Goal: recognize voluntary/involuntary blinks and right/left winks from fronto polar sensors (Fp1 and Fp2)

Eye blink examples (FP1 and FP2)

Eye blinks



Data collection

8 healthy adults without neurological disorders and without previous experience with BCI

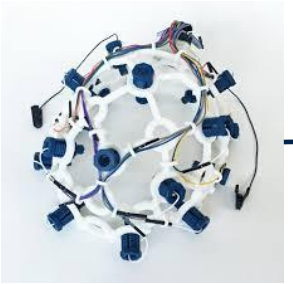
Visual cue was presented on the screen indicating:

- LW (left wink)
- RW (right wink)
- VB (voluntary blink)

Note: IB (Involuntary blink) was spontaneous

Balanced dataset with 2600 events has been collected and labelled

Firmware pipeline



EOG sensors

ADC

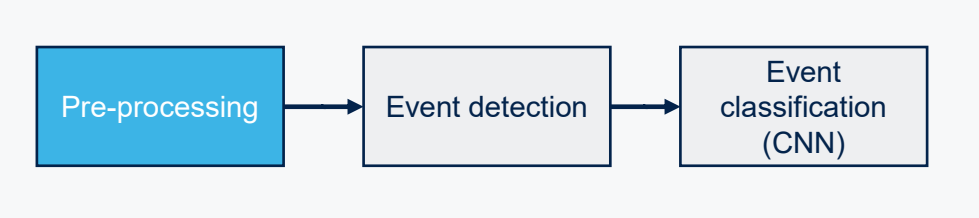
Pre-processing

Event detection

Event classification (CNN)

Blinks (Voluntary, Involuntary)

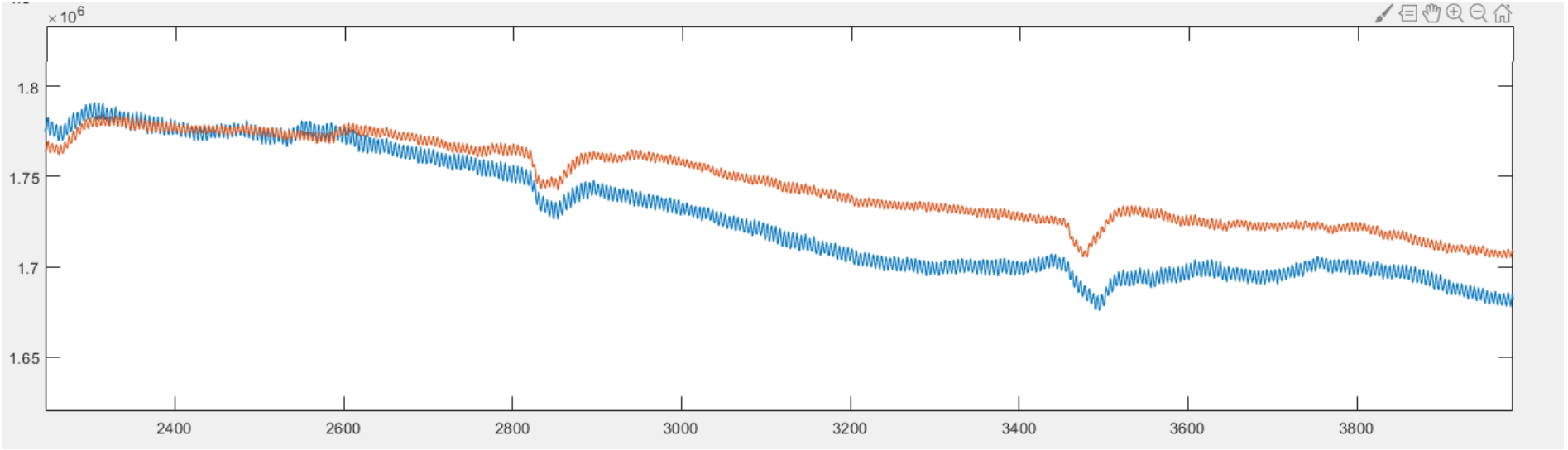
Winks (Left, Right)

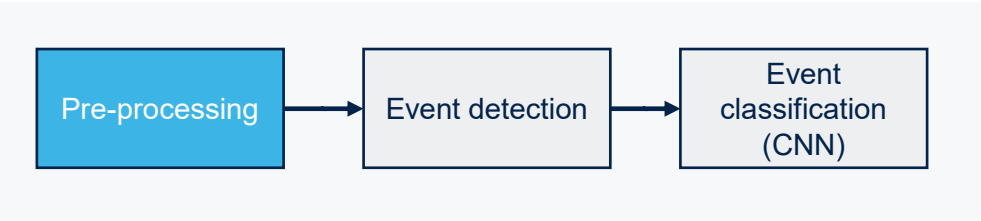


Pre-processing

Aims at:

- Eliminating the DC component
- Reducing noise (50Hz/60Hz)

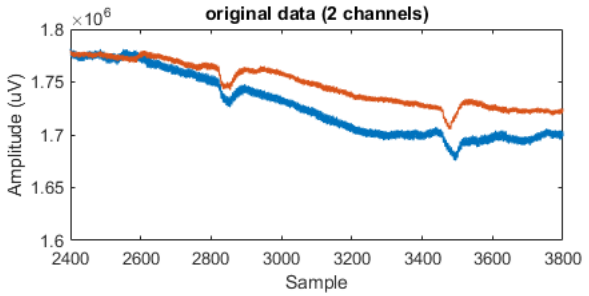


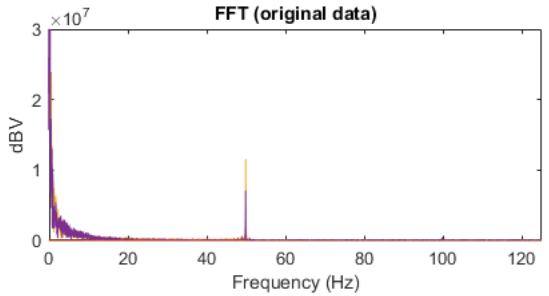


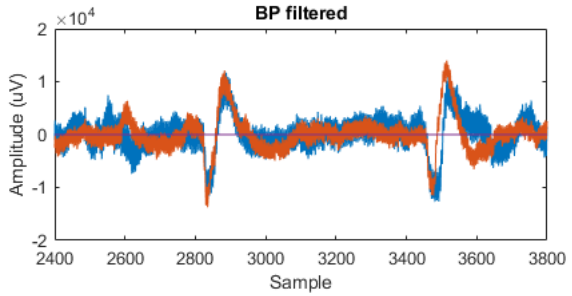
Pre-processing example

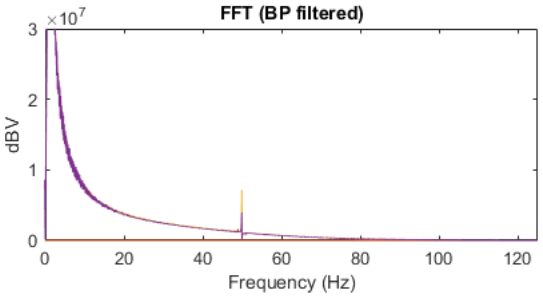
Real-time IIR band-pass filter
Cut-off frequencies:
f1= 0.1Hz
f2=50 Hz

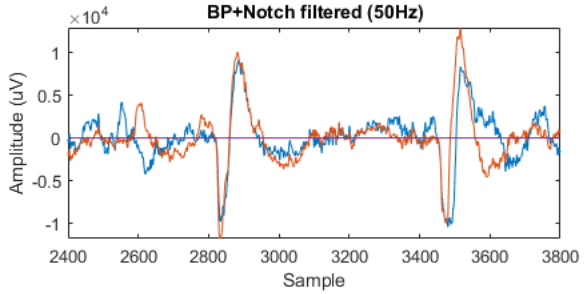
Real time IIR notch filter
f=50Hz

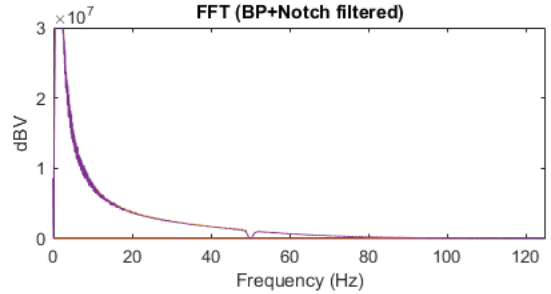


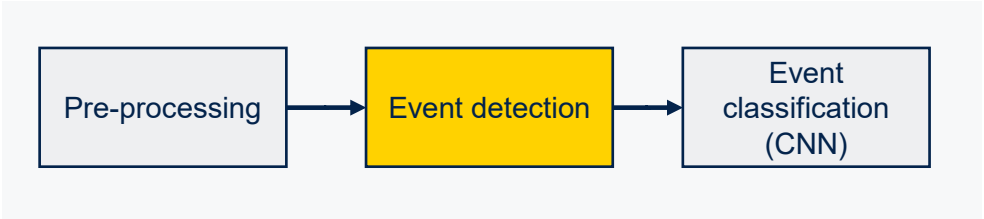






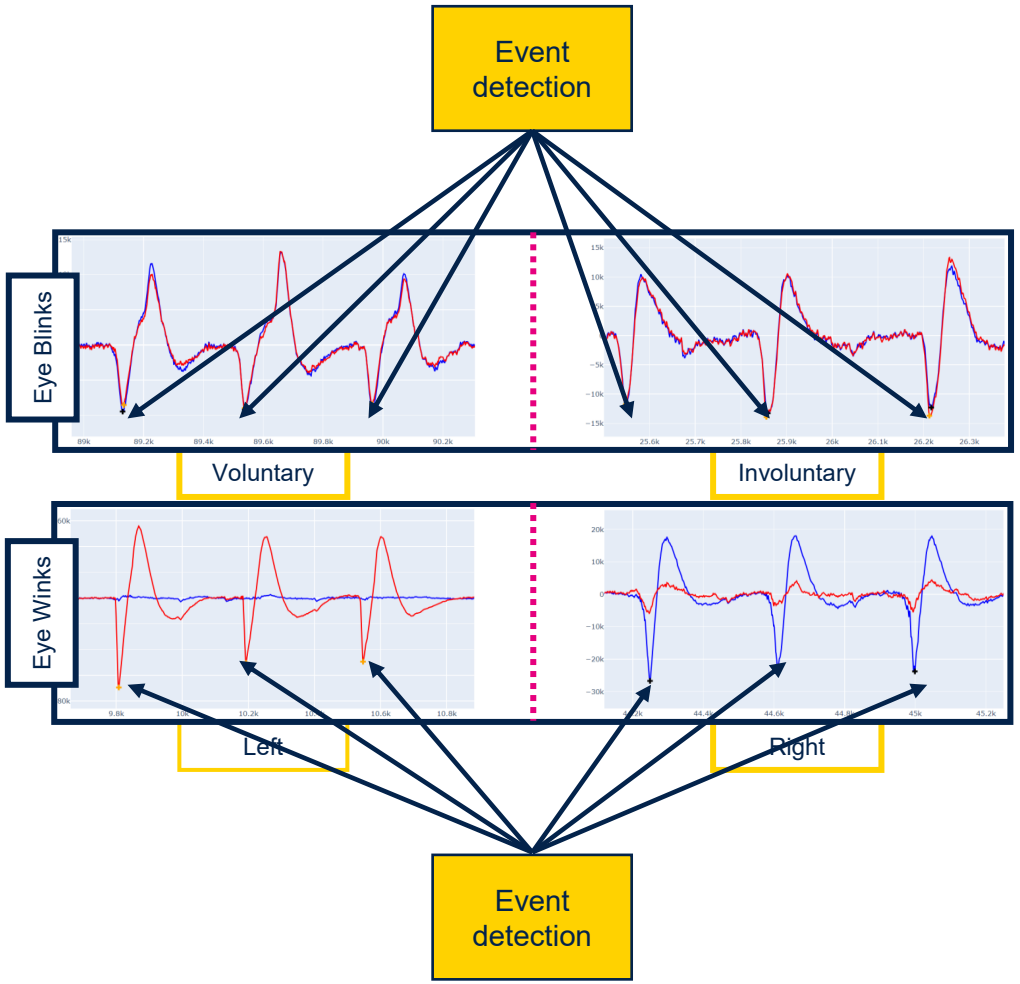




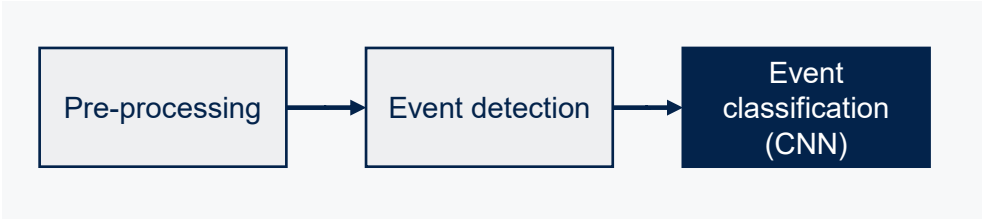


Event detection

- Aims to detect negative peaks
- Algorithm is based on z-score
- It triggers the collection of 200 samples



Ref.: JPG van Brakel. Robust peak detection algorithm (using z-scores). Stack Overflow: New York, NY, USA, 2014



Event Classification Results

Accuracy: 99%

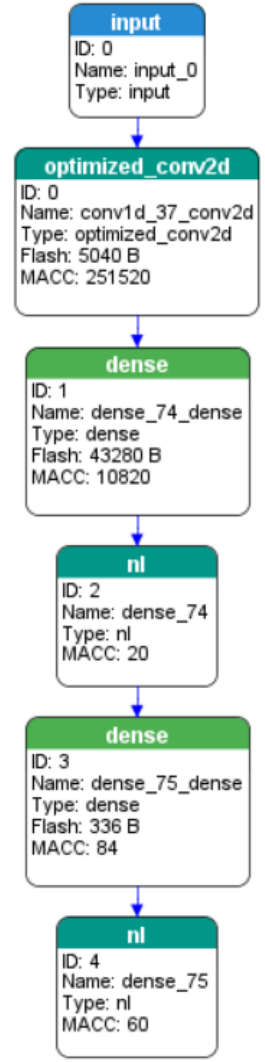
Footprint on STM32L475 MCU:
- Flash: 66 KB
- RAM: 11 KB

Complexity
- 270K MACC

Confusion matrix:

	VB	IB	LW	RW
VB	98.6%	1%	0.3%	0.1%
IB	0.7%	99,3%	0	0
LW	0	0.2%	99.8%	0
RW	0.3%	0	0	99.7%

VB: Voluntary (or forced) Blinks
IB: Involuntary Blinks
LW: Left Winks
RW: Right Winks



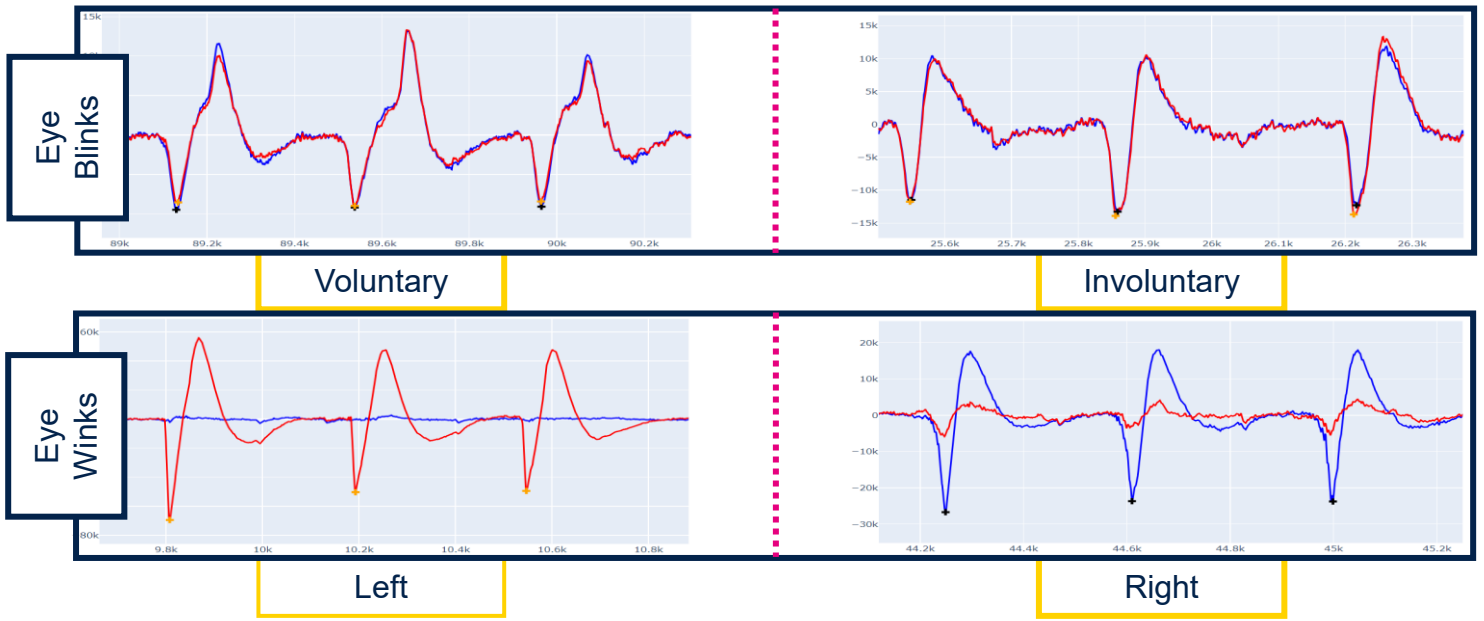
Execution on STM32L475 MCU

Inference time: 0.02 s
Acquisition time: 200/250 = 0.8 s (fs = 250 hz)
Total time: 0.82 s

Power consumption: 0.67 mW

GO AHEAD 20cm

NOTHING TO DO

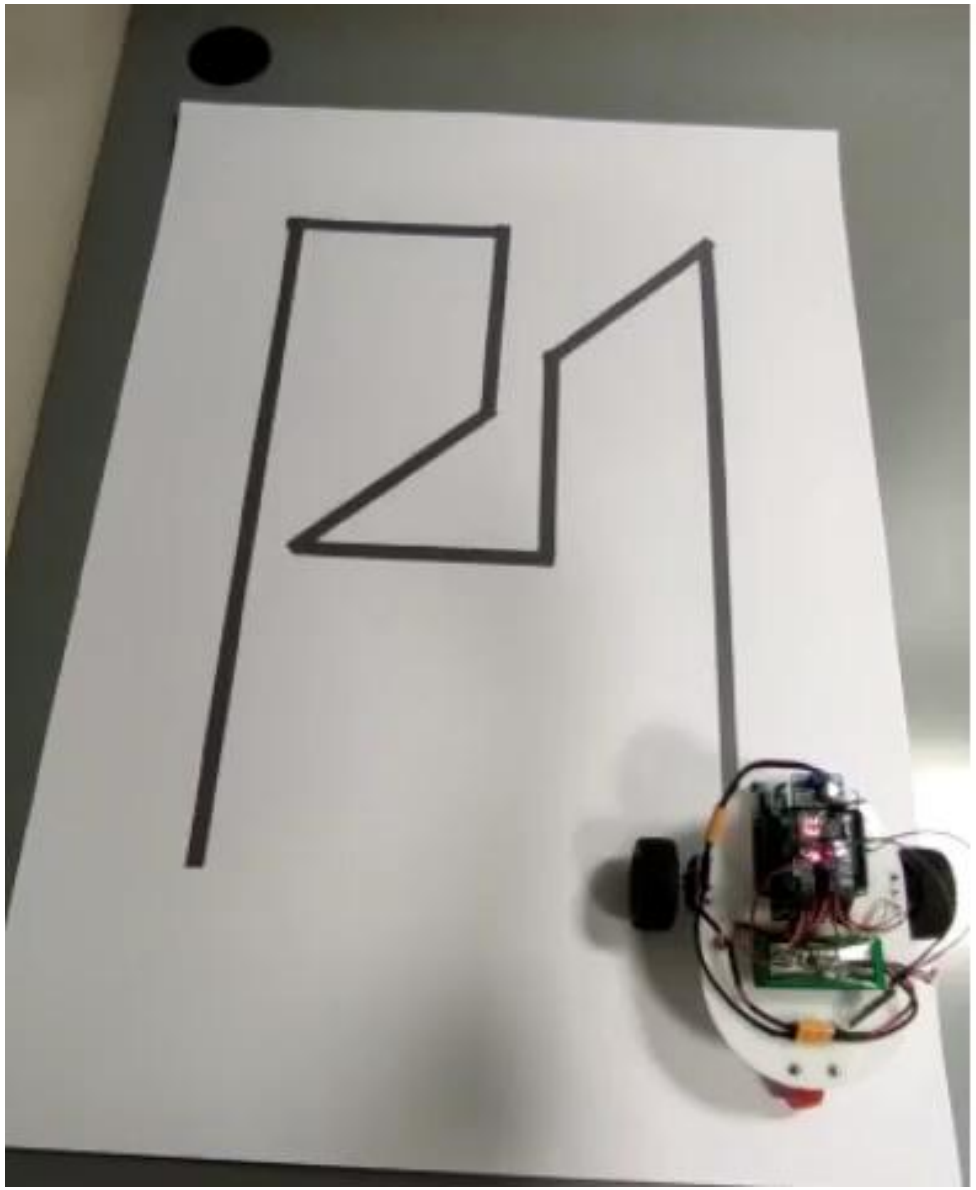


TURN LEFT 45°

TURN RIGHT 45°

Commands sent to the Robot
VB: Go ahead 20 cm
IB: Nothing to do
LW: Turn 45° left
RW: Turn 45° right

Complete algorithm running on the microcontroller



Our technology starts with You