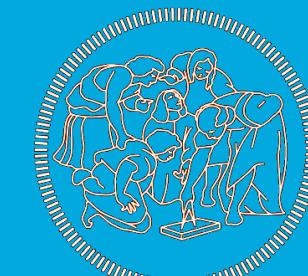


# MULTI - AGENT LEARNING

with

# LEGO EDUCATION SPIKE PRIME



POLITECNICO  
MILANO 1863

Alessandro D'Onofrio - 10864445



## Technical Specifications

- 6 LPF2 I/O ports for motors and sensors
- 115 kB port speed (ports E and F are prepared for “high-speed”)
- MicroPython integrated OS
- 100MHz M4 320 KB RAM 1M FLASH processor
- 2000 mHa lithium battery
- 6 axis Gyro Sensor integrated
- Bluetooth Classic 4.2 (BTC)
- Bluetooth Low Energy 4.2 (BLE)
- USB connection
- 32 MB of memory for programs, sound, and content
- 5×5 LED matrix white display
- Three-button navigation, including light interface
- Speaker

# COMPUTATIONAL PERFORMANCES

On the hub is possible to load and run pre-compiled micropython scripts that will make it execute specific actions moving its motors or using its sensors and their measurements.

**How much is powerful the hub?**

< 0.1 sec - the hub timer is not fully precise



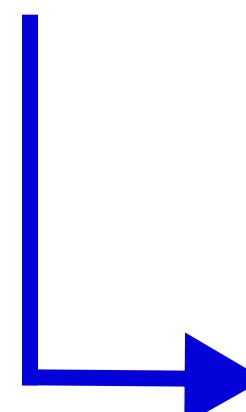
for a matrix multiplication with dimension  $10 \times 10$

**Memory problem:** not able to store a matrix  $100 \times 100$  (32768 bytes)

# BLUETOOTH - BLE

The hub uses a bluetooth low energy connection:

- Good stability, in a range of 10 meters from the other connected device
- Good latency, < 1 ms to transfer a program on the hub
- Upload speed on the hub ~4.84 kB/s

 The Lego Official Firmware is limited in terms of Bluetooth connection

The Pybricks firmware supports three BLE characteristics:

1. Pybricks Command/Event Characteristic
2. Pybricks Hub Capabilities Characteristic
3. Nordic UART Service

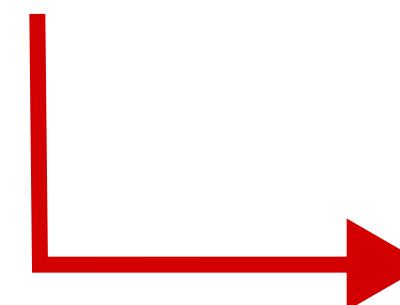
The first two are specific to Pybricks. They are used to download programs to the hub and run them.

The Nordic UART Service (NUS) is a standard characteristic that can be thought of as a BLE serial port. The Pybricks firmware uses it for things like `input()` and `print()`.

 a possible alternative:  
**pybricks and pybricksdev**

# WHAT CAN BE DONE?

Run the code directly on the micropython os installed on the hub.  
It is the easiest and possibly more effective way in order to avoid connection issues and concurrency issues.

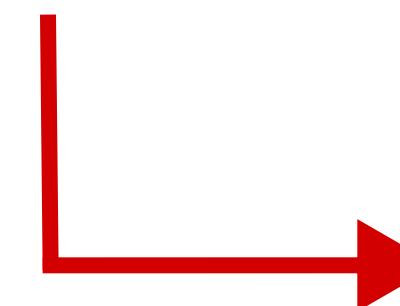


**Storage and memory problems, it is not able to handle an high amount of data**



we can try to implement Regret Matching on it and see how it works

Use Pybricks/Pybricksdev library, command line tools and firmware to perform computation on a computer and send instructions via Bluetooth on the hub, that is listening on the standar input.



**What do we need to understand?**



start a listening program on the hub directly through python code

# THANK YOU