NEUROPLATFORM FOR RESEARCH IN WETWARE COMPUTING

OBJECTIVE

Introducing the world's first Neuroplatform for 24/7, FULLY REMOTE wetware computing research, enabling unprecedented effectiveness and turnaround.

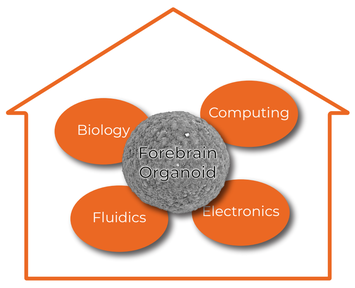
finalspark.com/neuroplatform

Researcher

EXPERIMENTS

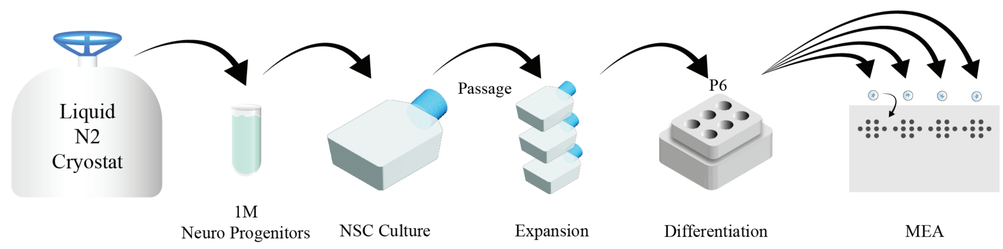
DATA

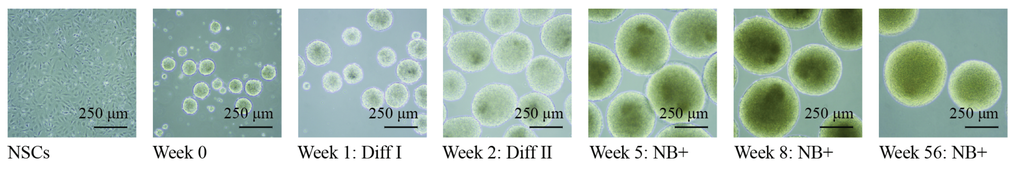
Fully Managed Remote Neuroplatform



24/7

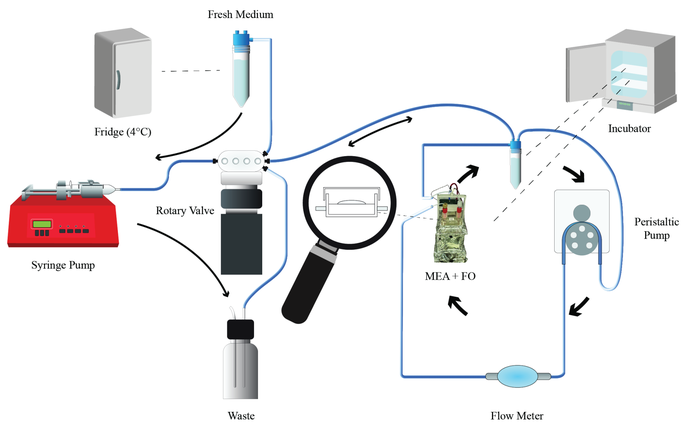
BIOLOGY



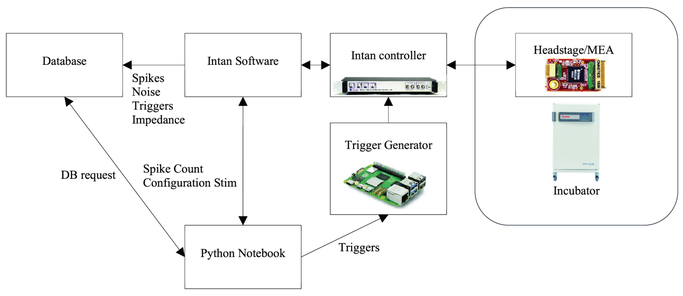


Protocol used for the generation of forebrain organoids (FO). Neural progenitors are first thawed, plated and expanded in T25 flasks. They are then differentiated in P6 dishes on orbital shakers, and finally manually placed on the MEA. (Below) Representative images showing various stages of FO formation and differentiation, taken at different time points. The scale bar represents 250 µm.

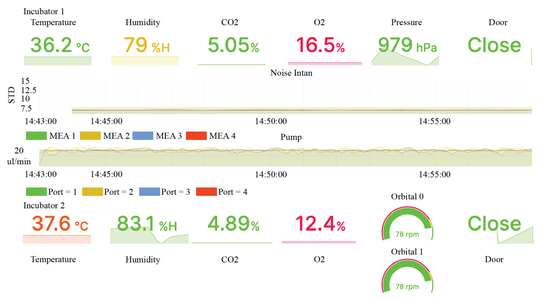
SYSTEM ARCHITECTURE



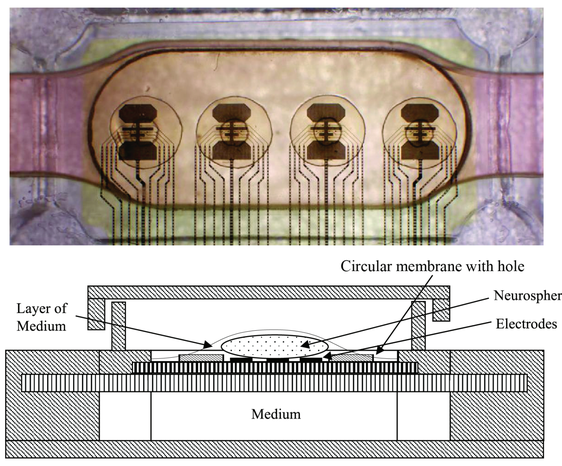
Microfluidic system illustrating the continuously operating primary system, which ensures constant flow in the medium chamber, and the secondary system responsible for medium replacing every 48 hours.



General architecture of the Neuroplatform. The Jupyter Notebook serves as the main controller, enabling initiation and reading of spikes, configuration stimulation signals and access to the database via Python.

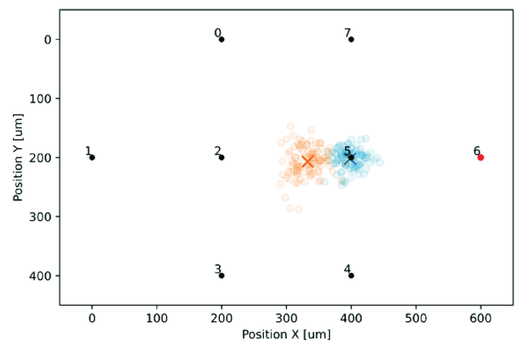


Graphic user interface to monitor critical parameters in the incubators.



Overview of the MEA, where the 32 electrodes are visible as 4 sets of 8 electrodes each. An FO is placed atop of each set of 8 electrodes, visible as a darker area. For each FO, the 2 circles correspond to a 2.5 mm circular membrane with a central hole. Cross-sectional view of the MEA setup, illustrating the air-liquid interface.

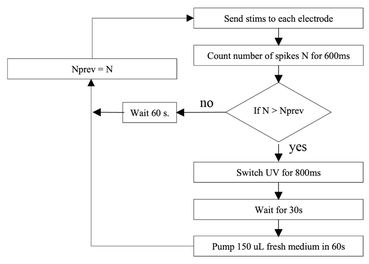
RESULTS



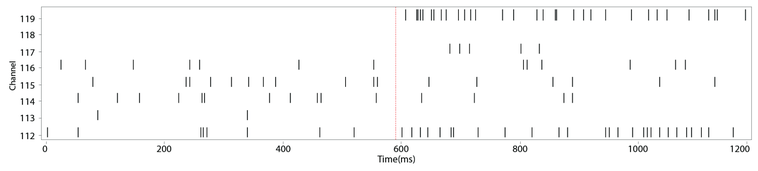
Center of activity modification

Graph showing the 2D layout of the 8 electrodes, the X and Y axis are normalized units showing the spatial coordinates of the electrodes. All electrodes can be used for both stimulation and reading. A 20 Hz stimulation signal is applied to electrode 6. The 100 blue circles represent the positions of the Center of Activity (CA) before 20 Hz stimulation, while the 100 red circles indicate the positions after the stimulation. The cross marks the average position.

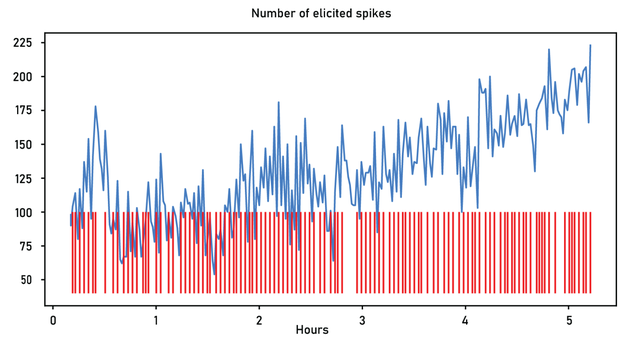
```python
for _ in range(4*60):
    for i in range(8):
        trigger.send(recordingTriggers[i])
        time.sleep(0.01) # 10 ms

        nb_spikes = read(trigger, intan, listeningTrigger) # Read number of spikes
        nb_spikes_ns = np.sum(nb_spikes[all_electrodes])

        diff_spikes = nb_spikes_ns-nb_spikes_ns_history[-1]
        nb_spikes_ns_history.append(nb_spikes_ns)

        if diff_spikes > 0: # Increase of spike activity
            triggerUV.send(800) # trigger UV for 100 ms

            time.sleep(30) # Wait 30s before removing dopa

            # Increase speed of the pump to 5 rpm (to remove dopa)
            peristaltic.rpm(5, PeristalticDirection.CounterClockWise)
        time.sleep(60) # wait 1 min
```



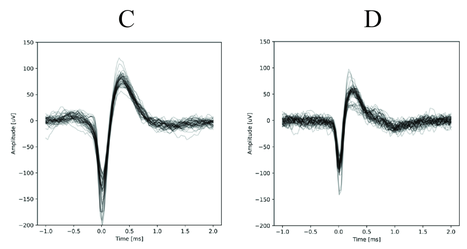
(Left) The Python source code implementing the closed-loop process.
(Right) Diagram illustrating the different steps involved in the closed-loop uncaging process of dopamine, which is repeated 240 times.



Timestamps of action potentials from the 8 electrodes before and after stimulation (shown as red line), showcasing the elicited spikes.



Graph displaying the number of elicited spikes over the 240 steps of the closed-loop (in blue) alongside the activation events of the UV light source (red).



C) Overlays of FO action potential recorded by the ALI system of the Neuroplatform.
(D) Overlays of FO action potentials recorded with an MCS system.

FinalSpark Sarl, rue du Clos 12, 1800 Vevey
Switzerland