Mind-controlled car demo

Mind-controlled car demo

- **Move your mind-controlled car from the starting line to a given parking region**
- The shorter time you use, the higher points you get

Mind-controlled car

- Use power of alpha band in EEG to control cars
    - Alpha waves increase when we close our eyes and relax
- Use the ratio of alpha power to determine whether the small car move or not
    - If the ratio exceeds a threshold, the car move forward

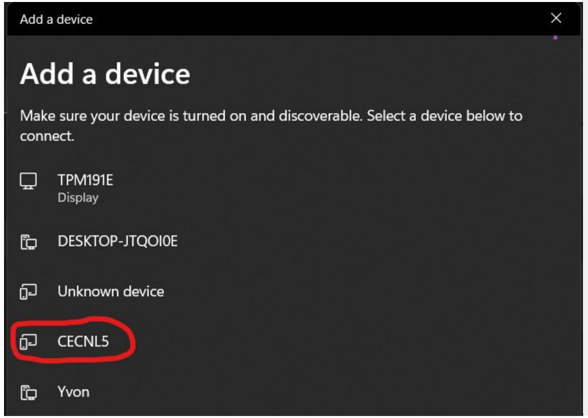
Requirements

Only 1 laptop in each group needs to do the following things

1. Clone all source codes (From Github)
2. Install softwares (Follow the instruction in readme)
   - Cygnus
   - Openvibe
   - Python (including the environment)
3. Check the functionality of cars
   - Bluetooth Module
   - Motors

Requirements - Check the functionality of small car

- Turn on the small car
- Connect the small car to your laptop via bluetooth
  - Find the device called **"CECNL YourGroupNumber"** in the bluetooth setting
  - The password is **1234** for all devices
  - Check the port number

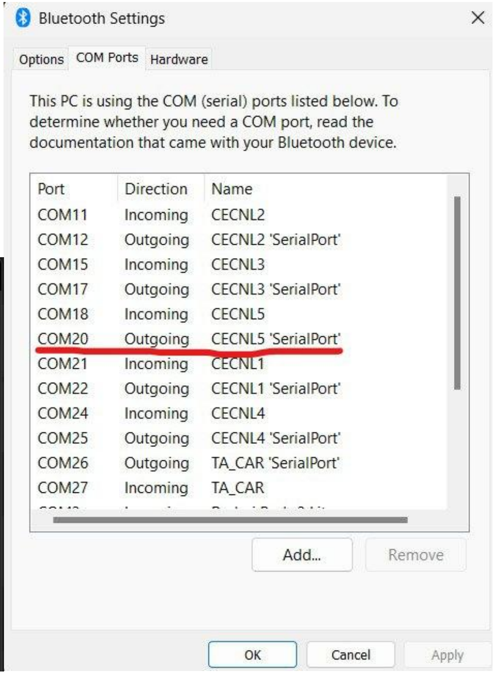


Add a device

Add a device

Make sure your device is turned on and discoverable. Select a device below to connect.

- TPM191E
  Display
- DESKTOP-JTQOI0E
- Unknown device
- CECNL5
- Yvon

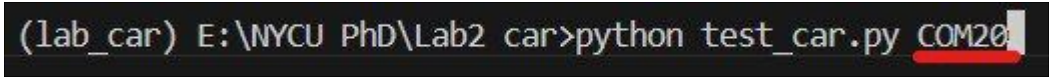


This PC is using the COM (serial) ports listed below. To determine whether you need a COM port, read the documentation that came with your Bluetooth device.

Port	Direction	Name
COM11	Incoming	CECNL2
COM12	Outgoing	CECNL2 'SerialPort'
COM15	Incoming	CECNL3
COM17	Outgoing	CECNL3 'SerialPort'
COM18	Incoming	CECNL5
COM20	Outgoing	CECNL5 'SerialPort'
COM21	Incoming	CECNL1
COM22	Outgoing	CECNL1 'SerialPort'
COM24	Incoming	CECNL4
COM25	Outgoing	CECNL4 'SerialPort'
COM26	Outgoing	TA_CAR 'SerialPort'
COM27	Incoming	TA_CAR

Requirements - Check the functionality of small car

- Activate the python environment
- Execute a python code “test_car.py” and enter the port number that the laptop used to connect with the car



```
(lab_car) E:\NYCU PhD\Lab2 car>python test_car.py COM20
```

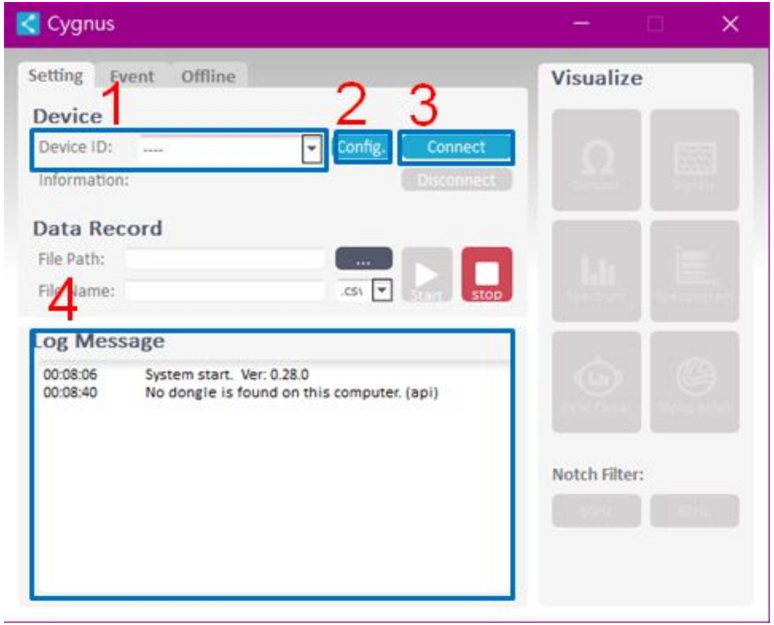
- Use keyboard (“WASD”) to test the functionality of your car

Pipeline

1. Put on EEG device and connect to dongle to you laptop
2. Cygnus - Connect to EEG device and show raw EEG signal
3. Openvibe acquisition server - Acquire EEG signal
4. Openvibe designer - Implement real-time EEG processing and calculate alpha power ratio
5. Python - Connect your laptop to small car via bluetooth and send fire command to the car based on current alpha power level

Cygnus - Connect to EEG device

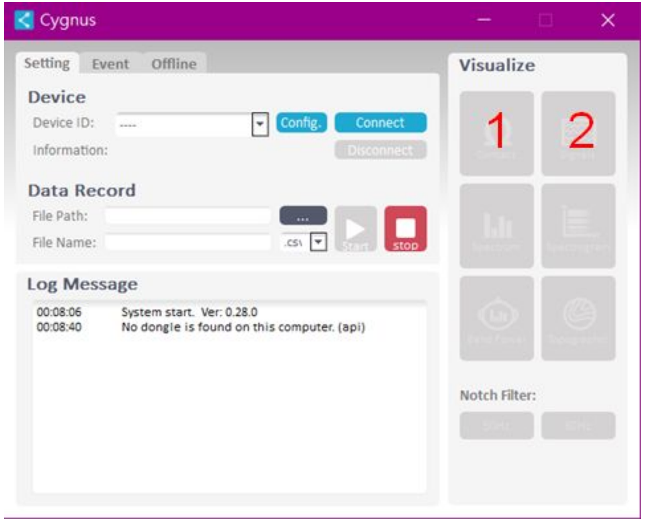
1. Select the correct device ID on your dongle
2. Connect your device to the laptop



```
00:08:06      System start.  Ver: 0.28.0
00:08:40      No dongle is found on this computer. (api)
```

Cygnus - Check EEG signals

1. Check the impedance of the channels you use
2. Adjust the position of each channel and try to lower the impedance
3. Check the raw EEG signal
   - No too many noise and artifacts



Device ID: ----

Information:

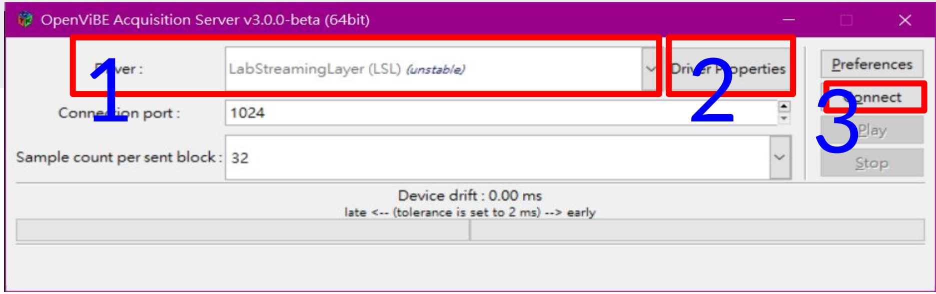
File Path:

File Name: .cs\

```
00:08:06        System start.  Ver: 0.28.0
00:08:40        No dongle is found on this computer. (api)
```

Openvibe acquisition server

1. Click driver button, and choose LabStreamingLayer
2. Click on driver properties button, and **make sure signal stream is from EEG signal and marker stream is None**
3. Click connect button, and check the terminal
4. Click play button
5. Check the delay time at the bottom of the window. If the delay time is not pretty high, it means that we have successfully received the EEG signals from LSL



OpenViBE Acquisition Server v3.0.0-beta (64bit)

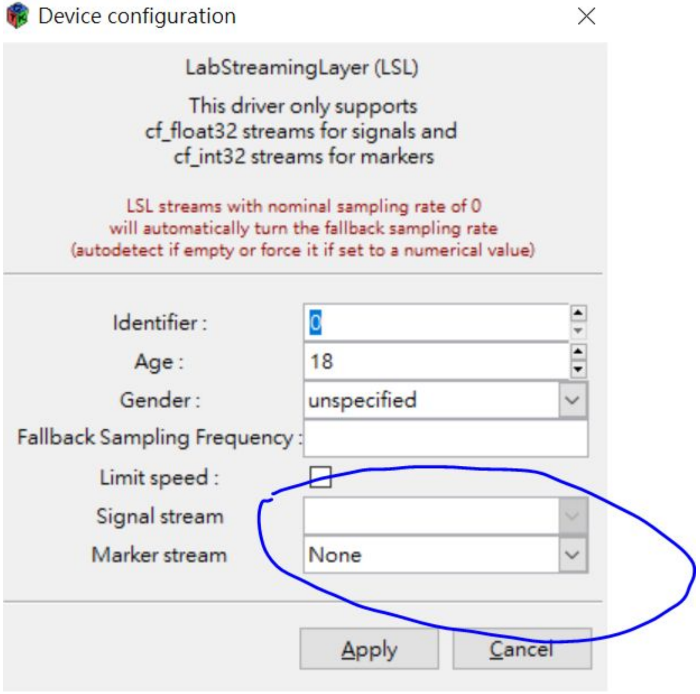
ver : LabStreamingLayer (LSL) *(unstable)*

Connection port : 1024

Sample count per sent block : 32

Device drift : 0.00 ms

late <-- (tolerance is set to 2 ms) --> early



Device configuration

LabStreamingLayer (LSL)

This driver only supports cf_float32 streams for signals and cf_int32 streams for markers

LSL streams with nominal sampling rate of 0 will automatically turn the fallback sampling rate (autodetect if empty or force it if set to a numerical value)

Identifier : 0

Age : 18

Gender : unspecified

Fallback Sampling Frequency :

Limit speed : □

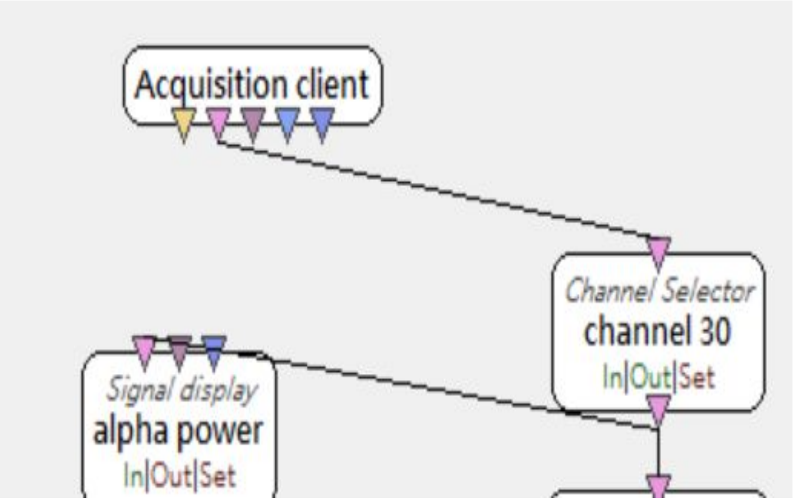
Signal stream

Marker stream None

Openvibe designer

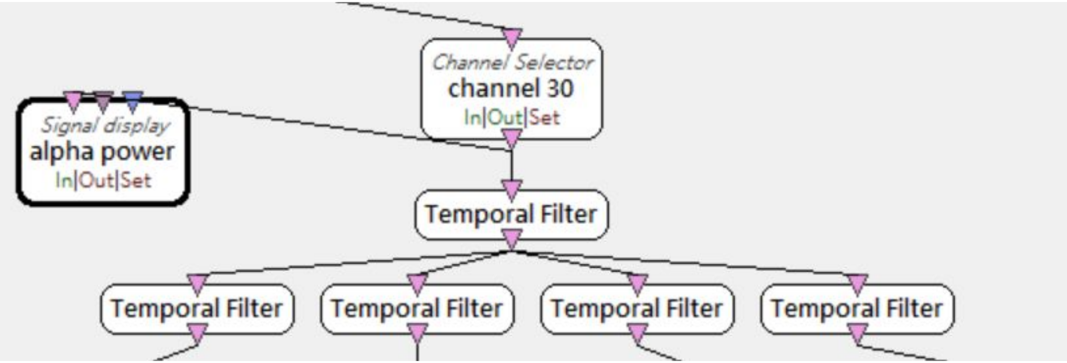
1. Open “alpha_wave_bandpower.xml”
    - A program for the real-time EEG processing
2. The procedure of EEG processing
    a. Select a certain EEG channel
    b. Bandpass filtering
    c. Extract 4 EEG sub-band power (delta, theta, alpha, beta)
    d. Calculate the alpha power ratio
3. Send the current alpha power ratio to python via LSL

Openvibe designer - part 1



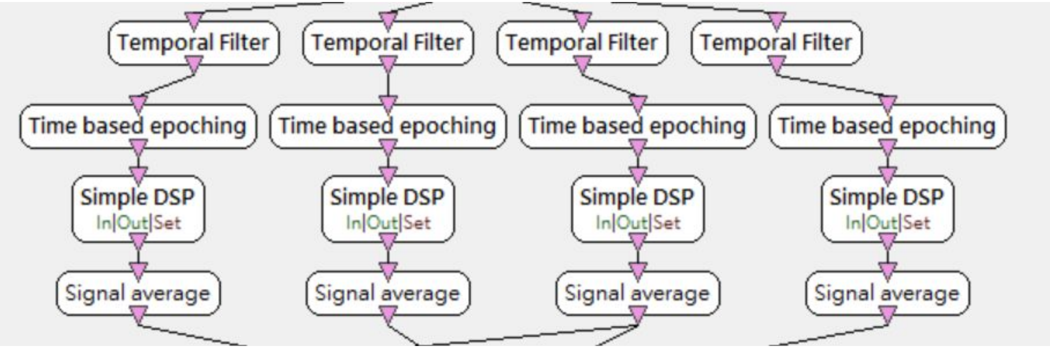
- Acquisition client: Receive brain wave data from openvibe acquisition server.
- Channel selector: Double-click on ”Channel Selector” and change the “channel list” options to select which channel’s signal you want to output .
- Display: Display EEG brain wave signal (double-click and select “rename” to change the name of signal display to “alpha power”)

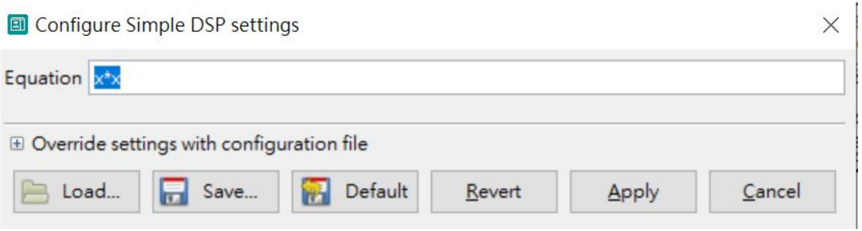
Openvibe designer - part 2



- Output of “channel selector” is the input of “temporal filter”. Double-click on “temporal filter” to set the filter from 1 to 40 HZ.
- Then the output signal are input to four temporal filter, which are set to 1~4hz, 5~8hz, 8~12hz, 12~40hz (which are delta wave, theta wave, alpha wave and beta wave)

Openvibe designer - part 3

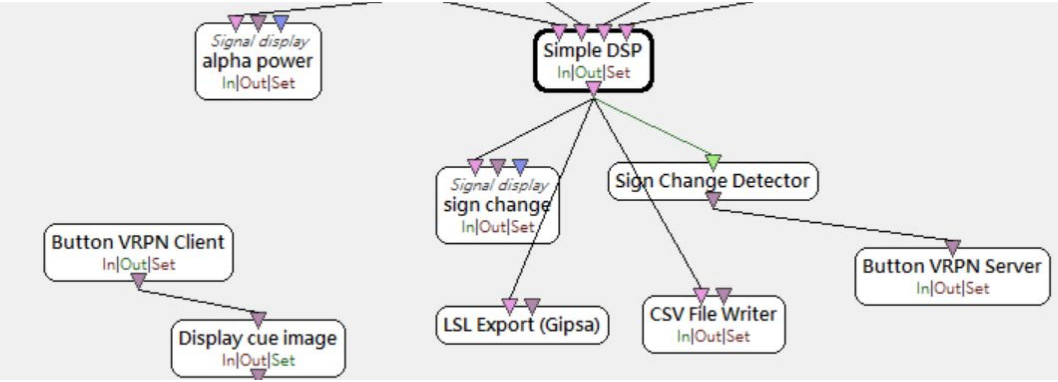



Configure Simple DSP settings

Equation x*x

- Every output of temporal filters are input to “time based epoching” in order to add more sample data. Double click on it, set ”duration” to 1 and set “epoch internal” to 0.0625
- Then connect output to “simple DSP”, double click on it and set equation to X*X , in order to obtain the power of the signal
- Next, output of the “simple DSP” is connected to “signal average“ to average the signal

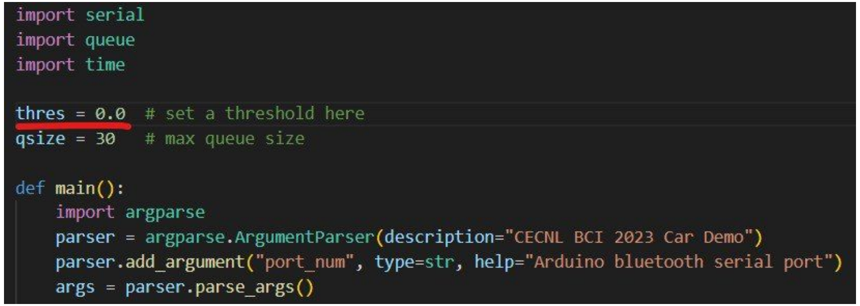
Openvibe designer - part 5



1. Connect alpha power ratio to “signal display”(named as sign change) to inspect real-time alpha power
2. LSL export (Gipsa): output the alpha power ratio to LSL (we will receive it from python later)
3. Sign change detector: it is used to detect the sign of input data.
4. “Button VRPN Server” and “Button VRPN Client” are use to transfer data
5. Display an image: If the alpha wave exceeds the threshold, the fish picture will be displayed, otherwise it will be empty

Python part

1. Connect the small car to your laptop via bluetooth
2. Pull the current alpha power ratio from LSL
3. Obtain the average alpha power ratio
4. Set a threshold for average alpha power ratio
    - If it exceed the threshold, send a signal to small car and the car will move forward



```python
import serial
import queue
import time

thres = 0.0  # set a threshold here
qsize = 30   # max queue size

def main():
    import argparse
    parser = argparse.ArgumentParser(description="CECNL BCI 2023 Car Demo")
    parser.add_argument("port_num", type=str, help="Arduino bluetooth serial port")
    args = parser.parse_args()
```

Python part

Assume your computer connects out to car through port “COM(your port number)”

> **test_car.py**: check car functionalities, you can control the car via keyboard (WASD). To execute, put “COM(your port number)” at the end of argument.

> **mind_controlled_car.py**: control car using alpha power ratio of received EEG stream. To execute, put “COM(your port number)” at the end of argument.