Before doing anything with Rpi3, I recommend installing the fan or cooler to the raspberry pi because it gets hotter while installing and doing make operation which leads to slower performance.

Hapticomm version

Log in the raspberry PI 3

```
$ mkdir ~/hapticomm_efficacy (or any other dedicated location and folder will be fine)
$ cd  ~/hapticomm_efficacy
$ git clone
$ https://github.com/bas1l/hapticomm-efficacy-psychophysics.git
$ cd hapticommlib/
$ sudo ./configure.sh
```

cmake

After this command: here is the problem with the old **CMake** version in Rpi3, the default version is 3.16.3, but we need 3.18 and higher. Use this tutorial to install the newest version OSDevLab: How to install latest Cmake for Raspberry Pi . The version I installed is Index of /files/v3.18 (cmake.org)

First uninstall the previous version of CMake:

```
$ sudo apt update
$ sudo apt install build-essential libtool autoconf unzip wget
$ sudo apt remove --purge --auto-remove cmake
```

Then Go to the official CMake webpage, then download and extract the latest version. Update the version and build variables in the following command to get the desired version:

```
$ version=3.23
$ build=2 ## don't modify from here
$ mkdir ~/temp
$ cd ~/temp
(wget
https://cmake.org/files/v$version/cmake-$version.$build.tar.gz
tar -xzvf cmake-$version.$build.tar.gz)
$ wget https://cmake.org/files/v3.18/cmake-3.18.4.tar.gz
(tar -xzvf cmake-$version.$build.tar.gz)
```

```
$ tar -xzvf cmake-3.18.4.tar.gz
cd cmake-$version.$build/
$ cd cmake-3.18.4/
$ ./bootstrap ## It will take some time :)
```

I had errors here with openSSl:

```
$ sudo apt-get install libssl-dev
$ ./bootstrap ## again
```

```
$ sudo make ## It will also take lot time ~ 1 - 1.5 hour
```

(there is a way (in theory) to make it faster by command $ make -j$(nproc) instead of `$ sudo make`)

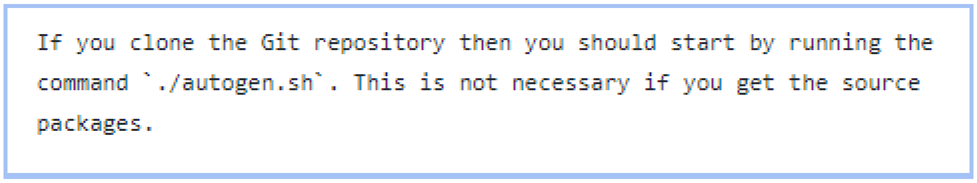
```
$ sudo make install
```

Cmake is installed successfully! Run : “`$ cmake --version`” to check

```
$ cd
$ pip install pyzmq
$ cd ~/hapticomm_efficacy/hapticommlib/
$ sudo ./configure.sh ## giving errors with cppzmq
```

libzmq

Installation via command `./autogen.sh`: from(libzmq/INSTALL at master · zeromq/libzmq · GitHub)



If you clone the Git repository then you should start by running the command `./autogen.sh`. This is not necessary if you get the source packages.

```
$ mkdir zeromq
$ cd zeromq
$ git clone https://github.com/zeromq/libzmq
$ cd libzmq
$ ./autogen.sh
$ mkdir build
$ cd build
$ cmake ..
$ sudo make -j4 install
```

cppmzq

Build cppzmq via cmake. This does an out of source build and installs the build files

- download and unzip the lib, cd to directory
- mkdir build
- cd build
- cd
- sudo make -j4 install

```
$ cd
$ mkdir zeromq_next
$ cd zeromq_next
$ git clone https://github.com/zeromq/cppzmq
$ cd cppzmq
$ mkdir build
$ cd build
$ cmake ..
$ sudo make -j4 install
```

```
$ cd
~/hapticomm_efficacy/hapticomm-efficacy-psychophysics/hapticommlib$
sudo ./configure.sh
$ cd ../build
$ sudo make
$ cd ..
```

Update CMakeLists.txt and add these lines:

```
$ nano CMakeLists.txt
#find cppzmq wrapper, installed by make of cppzmq
# B. External libraries
find_package(cppzmq)
target_link_libraries(libhapticomm cppzmq)
```

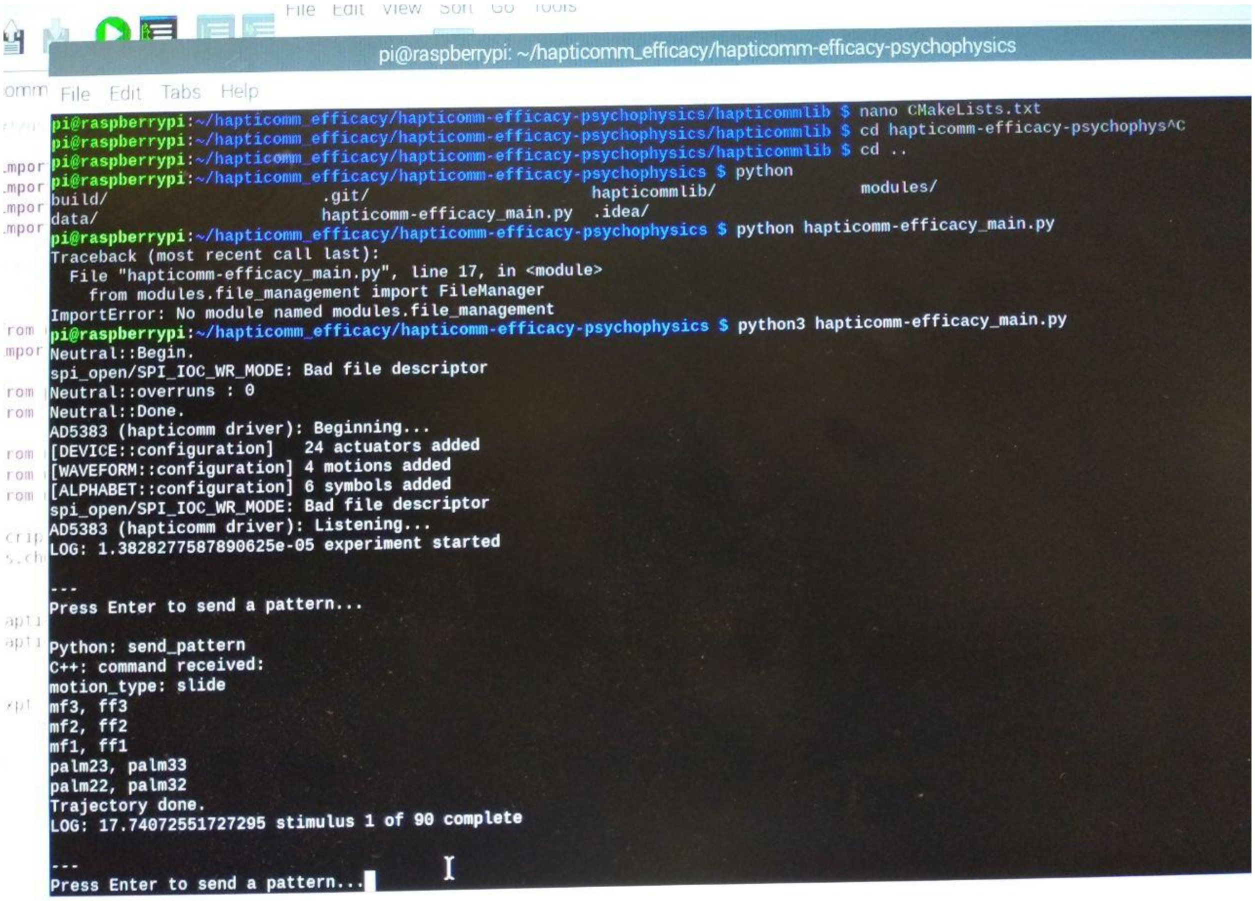
Examples to run the code:

```
$ python hapticomm-efficacy-main.py
$ python3 hapticomm-efficacy-main.py
```

Some Dependencies to run the code, if errors happen:

```
$ pip install pynput
$ pip3 install pyzmq
$ pip3 install pynput
```

Output:


pi@raspberrypi: ~/hapticomm_efficacy/hapticomm-efficacy-psychophysics
File Edit Tabs Help

```
pi@raspberrypi:~/hapticomm_efficacy/hapticomm-efficacy-psychophysics/hapticommlib $ nano CMakeLists.txt
pi@raspberrypi:~/hapticomm_efficacy/hapticomm-efficacy-psychophysics/hapticommlib $ cd hapticomm-efficacy-psychophys^C
pi@raspberrypi:~/hapticomm_efficacy/hapticomm-efficacy-psychophysics/hapticommlib $ cd ..
pi@raspberrypi:~/hapticomm_efficacy/hapticomm-efficacy-psychophysics $ python
build/                 .git/                    hapticommlib/          modules/
data/                  hapticomm-efficacy_main.py  .idea/
pi@raspberrypi:~/hapticomm_efficacy/hapticomm-efficacy-psychophysics $ python hapticomm-efficacy_main.py
Traceback (most recent call last):
  File "hapticomm-efficacy_main.py", line 17, in <module>
    from modules.file_management import FileManager
ImportError: No module named modules.file_management
pi@raspberrypi:~/hapticomm_efficacy/hapticomm-efficacy-psychophysics $ python3 hapticomm-efficacy_main.py
Neutral::Begin.
spi_open/SPI_IOC_WR_MODE: Bad file descriptor
Neutral::overruns : 0
Neutral::Done.
AD5383 (hapticomm driver): Beginning...
[DEVICE::configuration]   24 actuators added
[WAVEFORM::configuration] 4 motions added
[ALPHABET::configuration] 6 symbols added
spi_open/SPI_IOC_WR_MODE: Bad file descriptor
AD5383 (hapticomm driver): Listening...
LOG: 1.3828277587890625e-05 experiment started

---
Press Enter to send a pattern...

Python: send_pattern
C++: command received:
motion_type: slide
mf3, ff3
mf2, ff2
mf1, ff1
palm23, palm33
palm22, palm32
Trajectory done.
LOG: 17.74072551727295 stimulus 1 of 90 complete

---
Press Enter to send a pattern...
```

Figure 1. Output of *haptiomm-efficacy_main.py*

Install Python 3.6 and higher

Please, follow the steps:
https://installvirtual.com/install-python-3-on-raspberry-pi-raspbian/amp/

Change default python

```
$ "sudo update-alternatives --config python"
```

And then enter "1" or "2" depending which python2 or python3 is numerated there.

Then check by entering "python" command, it will automatically open the default python version:

```
$ python
```

If SPI does not work

1) Method:

```
sudo raspi-config
```

https://www.mathworks.com/help/supportpkg/raspberrypiio/ref/enablespi.html

Motor winding by default is different

```
git clone https://github.com/bas1l/hapticomm-efficacy-psychophysics.git
```

```
git checkout dev_forZhanat
```

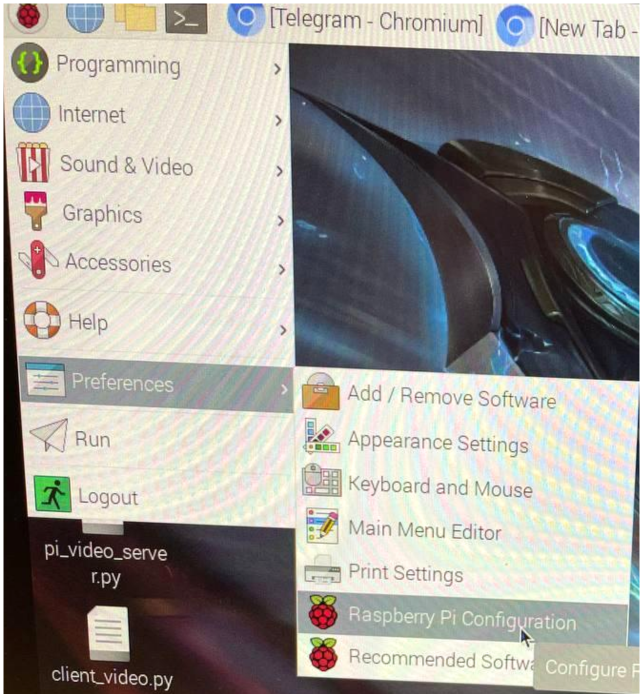
```
sudo pip3 install keyboard
```

```
elif c == "i":
            stim = {
                'type': "tap",
                'nb_actuators': 1,
                'width': 1,
                'length': 1,
                'actuators': ["ff1"]
            }
```

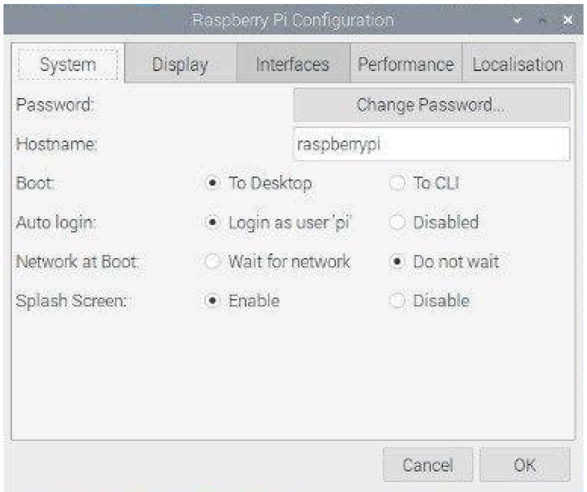
tap, tap-and-hold, slide

2) Method:

On the left up corner click on the Raspberry pi icon to open the menu:



Choose "Preferences" -> "Raspberry Pi Configuration". You should see this new window:



Password: Change Password...

Hostname: raspberrypi

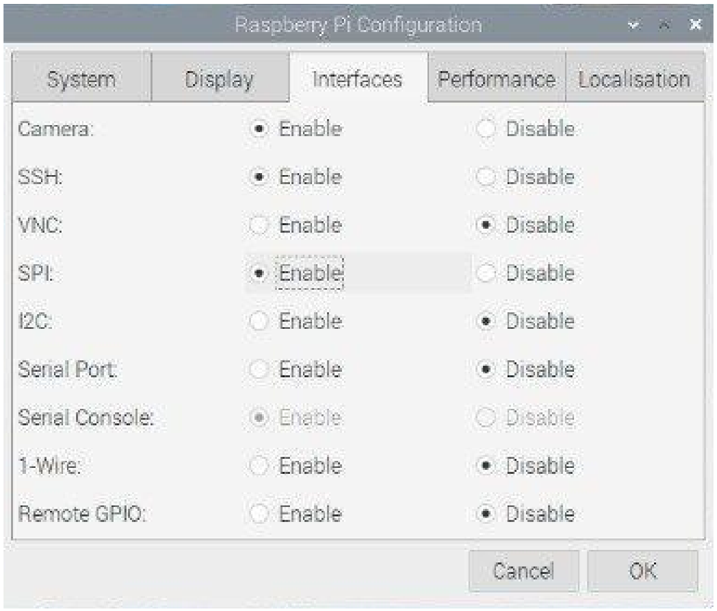
Boot: ● To Desktop ○ To CLI

Auto login: ● Login as user 'pi' ○ Disabled

Network at Boot: ○ Wait for network ● Do not wait

Splash Screen: ● Enable ○ Disable

Please, choose "Interfaces".



Camera:	● Enable	○ Disable
SSH:	● Enable	○ Disable
VNC:	○ Enable	● Disable
SPI:	● Enable	○ Disable
I2C:	○ Enable	● Disable
Serial Port:	○ Enable	● Disable
Serial Console:	● Enable	○ Disable
1-Wire:	○ Enable	● Disable
Remote GPIO:	○ Enable	● Disable

In SPI press "Enable".