UQ NeuroTech Mind Bot Race 2023 Info Sheet – Emotiv headset

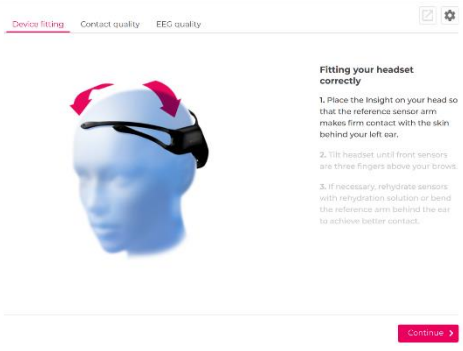
1. Installation of software
2. Setting up your EEG
3. Training your profile
4. Configuring your robot
5. Racing your robot

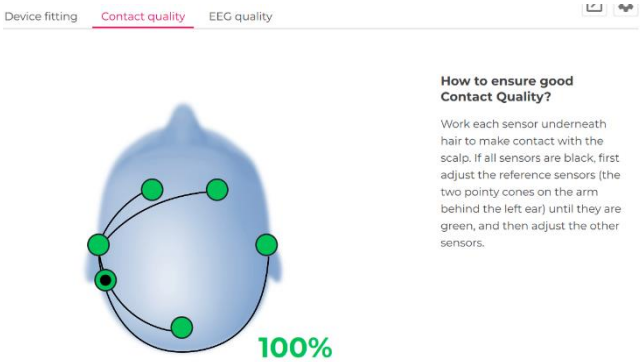
1. Installation of software

- Python: you can install the latest version from here. All of the code for the Emotiv will run on Python. If you have taken ENGG1001 or CSSE1001 or any other python programming course, you should already have this, as well as an IDE.
- Python code for the event: Please download from our github.
- Python IDE: a python IDE (Integrated Development Environment) is software dedicated to developing and running code. This is optional, as python comes with its basic code executer (IDLE), but if you have a preferred IDE please feel free to use that. VS code and Pycharm re famous examples of Python IDEs.
- Python libraries: once you have opened the provided program in your IDE, you will notice that we are "importing" some python libraries for our code. Some of these libraries are already installed onto your device, so all you need to do is to "import." For more rare libraries such as *websocket*, you need to open your terminal (command prompt/windows powershell) to use " pip install -module name-"
- Emotiv launcher: This is the base software that will allow connections between your laptop, EEG and API. Install from this link.
  Username: uqneurotech Password(first letter capital): Botrace2023
  We only have licenses for three devices total. Therefore, each team can only have one laptop that installs the application under our license, so please coordinate with your teammates accordingly. The regular license is free, so you can always make an account for yourself for future use.
  You will be asked which applications other than the Emotiv Launcher you would like to install. For this event, only the Emotiv BCI application is required.

2. Setting up your EEG

- Once you open the Emotiv launcher, you will be prompted to connect your Emotiv Insight



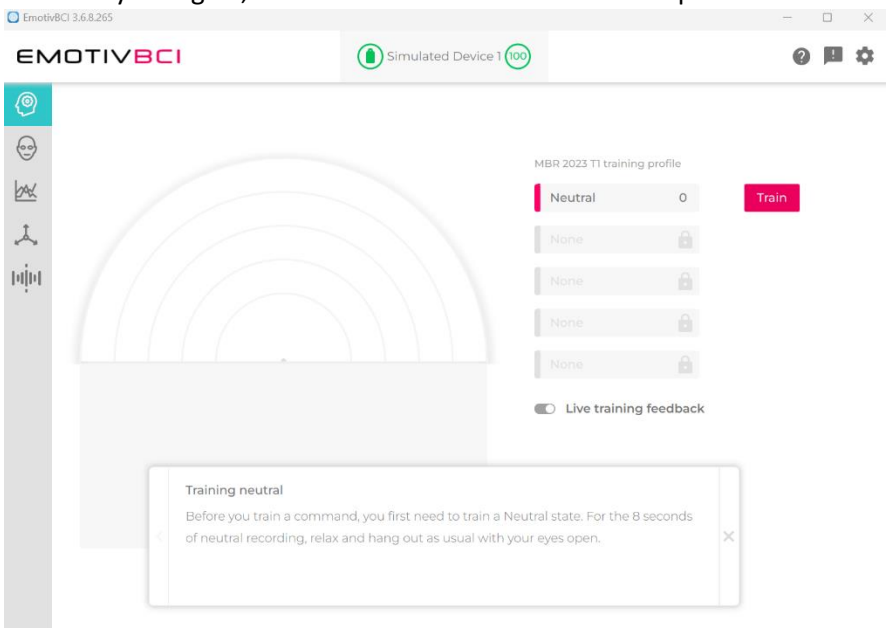


headset. Press the button on the top side and check if the white light on the side turns on, and place on your head as instructed. Then, check that the contact quality and EEG quality is

high enough. The values will fluctuate every second, but you would want to aim for at least 70% consistently; contact quality will be higher than EEG quality. Try adjusting the positions, get the electrodes as close as possible to your scalp and away from hair in order to maximise contact. You can use the electrolyte gel provided in the box to increase conductivity. If there are any electrodes in the red zone, there will be a significant reduction in conductivity. After you change any position of the headset, ensure you wait a few seconds for the change in quality to be reflected onto the app.

3. Training your profile

- Once you have your headset connected, now is the time to train your profiles for your robot to operate based on. In simple terms, you will think of different movements, and the application will analyse the brainwave trends after repeated input of these "thoughts." After a sufficient number of inputs, the algorithm will be able to detect what you are thinking of – and you will be able to connect those neural outputs to movement of our robots. You will start with "neutral" – and then on to push, pull, among other options that you can add. The more training you do the better: we recommend at least 20 for each profile. They don't have to be exclusively thoughts, motor commands are also neural inputs too.



4. Configuring your robot

- For this event, we will be using the Micromelon Rovers kindly provides to us by UQ MARS. Please download the code provided in our github. Once you've done so, there are some parts where you need to change: these are marked by +----------+ marks. Most of the code is for the API, so you don't need to understand any of it(not many of us understand either). Just follow the directions in the relevant parts of the code and ask any questions to our team.

```
933
934         self.c.bind(inform_error=self.on_inform_error)
935         try:
936             self.rc = RoverController()
937             self.rc.connectBLE(83) #change the number to the designator of your team's rover.
938             self.rc.startRover()
939         except:
```

```
1116            if command == 'open':
1117                if self.c.session_id == '':
1118                    self.profile_name = "MBR 2023 T1"  #change this accordingly to your team
1119                    self.start(self.profile_name)
1120                    self.c.bind_async(self.loop, create_session_done = self.receive_message)
1121                    self.loop.run_until_complete(self.wait_for_event())
```

```
1180        elif command == 'connect robot': #only required if you hadn't connected the rover when you started the code.
1181            self.rc = RoverController()
1182            self.rc.connectBLE(83) #make sure the designator for your robot is correct.
1183            self.rc.startRover()
1184        else:
1185            print("Invalid command")
```

```
1299    def move_robot(self, *args, **kwargs):
1300
1301        """
1302        this is the main function you will deal with when controlling your robot.
1303        """
1304        #the two lines below allow the function to receive data coming from the EEG in real-time
1305        data = kwargs.get('data')
1306        print('mc data: {}'.format(data))
```

- Utilise different commands for the micromelon rover to create a function that moves your robot in the desired direction, according to your trained profiles.

5. Racing your robot

- To run the code, simply run the base program from your chosen software. **Make sure that the Emotiv BCI app is closed.** The Emotiv launcher can stay open- it may be useful when checking connectivity during the runs. There are three steps to running your code.
- Press run on your IDE. This is usually a play button. Running the code should automatically connect you to the micromelon rover. If not, you can always connect later. When you run the program for the first time, there will be a popup window on your emotiv launcher – press accept.
- Type “open” as a command in your Python terminal. This will establish a connection between your device and headset.
- Type “train” as a command in your Python terminal. This will load the training profiles from the Emotiv BCI.
- Type “move robot” as a command in your Python terminal. This will call the function you set up in the previous step.

  These are some of the key steps to getting your robot up and running. Of course, you are more than welcome to make any modifications to get you better chance for winning!