User Guide

BCI spelling keyboard – team2021mindaffect

2020/2021
Modern Software Development Techniques

Thomas de Lange,
Thomas Jurriaans,
Damy Hillen,
Joost Vossers,
Jort Gutter,
Florian Handke,
Stijn Boosman

Radboud Universiteit

Table of Contents

Introduction

The module implements a keyboard for the BCI system developed by MindAffect[1].
This keyboard was made as part of the course Modern Software Development Techniques (MSDT) via the Radboud University Nijmegen.

The BCI-driven keyboard has the following functionality:

- Multiple keyboards (sub-keyboards) can be used.
- It supports adding your own configurable keyboard from a JSON file.
- The keyboard has word prediction buttons that predict the most logical next word to be typed.
- It has word correction which automatically corrects wrongly typed word.
- There is an option to let the keyboard say out loud what has been written (text-to-speech).

Figure 1: The BCI can be used in all sorts of contexts.

Setup

This section explains how to set up and use the BCI-driven keyboard.

Requirements

The recommended Python version to use this module is some version of Python version 3.8.

Since this keyboard implementation is based on the mindaffectbci module[2] developed by MindAffect, it is necessary to install this module. This can be done using pip with the following command:

```
pip install mindaffectBCI
```

All the other requirements can be found in `requirements.txt` and can be installed using pip as follows:

```
pip install -r requirements.txt
```

The required modules should now be installed as part of your Python environment.
When this is finished, and no errors have occurred, all the requirements will have been installed.

As of now, the required modules are:

- pyttsx3 (version 2.5)[3]
- gTTS[4]
- playsound (version 1.2.2) [5]
- google-cloud-texttospeech[6]
- symspellpy (version 6.7.0) [7]
- psychopy (version 2020.2.6) [8]
- opencv-python (version 4.4.0.46) [9]

Installation

Finishing up the installation of the module can be done by running the following command in the folder where `setup.py` is located:

```
python setup.py develop
```

Hardware

To be able to use the BCI-driven spelling keyboard, it is necessary to use a specific headset which will pick up the EEG signals in your brain.

All links below redirect to MindAffect's documentation pages:

- More information about the headset can be found here[10].
- Information on how to print your own headset with a 3D-printer can be found here[11].
- It is also possible to use the keyboard (and the entire BCI module) with a fake data stream that is streaming real brain data, which might be useful for testing. More information on this can be found here[12].

Usage

Before opening the application, make sure that the headset is properly connected to the device the keyboard will be used on. Information on the system and headset setup can be found on this documentation page[13] from MindAffect.

If flickering is activated (`use_flickering` in the `user_config`), the first thing that should appear when opening the application is a connection screen which gives feedback about what the status is of the connection to the Utopia Hub (Figure 2). If a connection is established, flickering will be used to control the keyboard. If the connection failed, the keyboard starts without flickering and will have to be used with the mouse.

Figure 2: The Connection screen.

Settings

After the connection screen, the settings menu will appear. The settings menu allows the user to change the layout and functionality of the keyboard. The user can move here from the keyboard itself by clicking on the cogwheel icon in the bottom left (Figure 3).

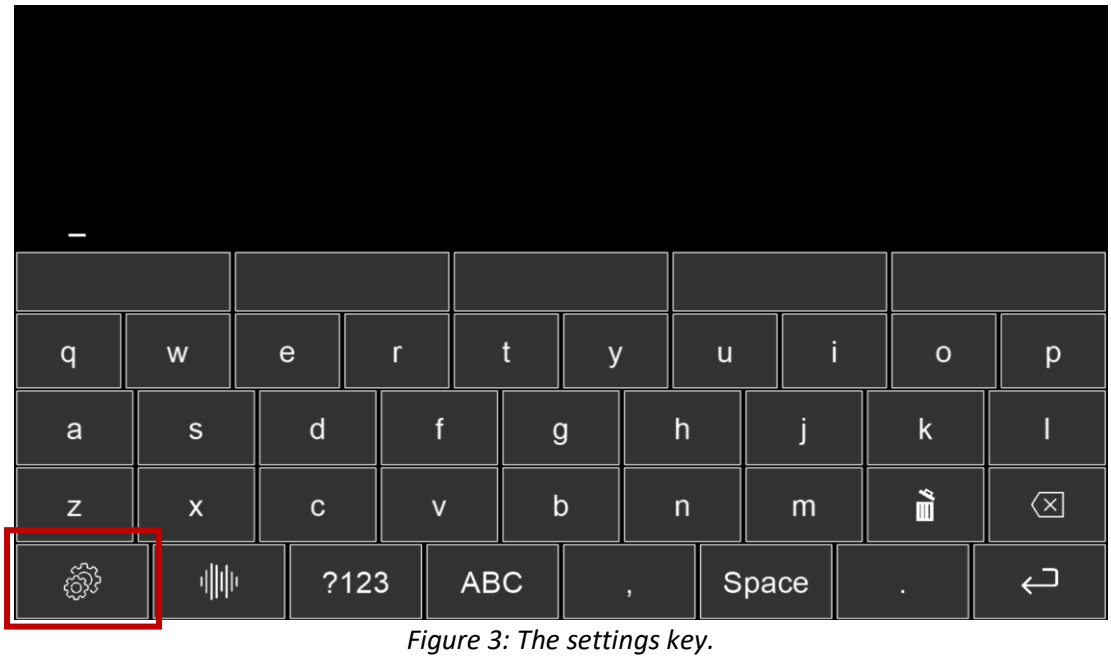

Figure 3: The settings key.

First of all, the word prediction and word correction can be toggled to be on or off (Figure 4).

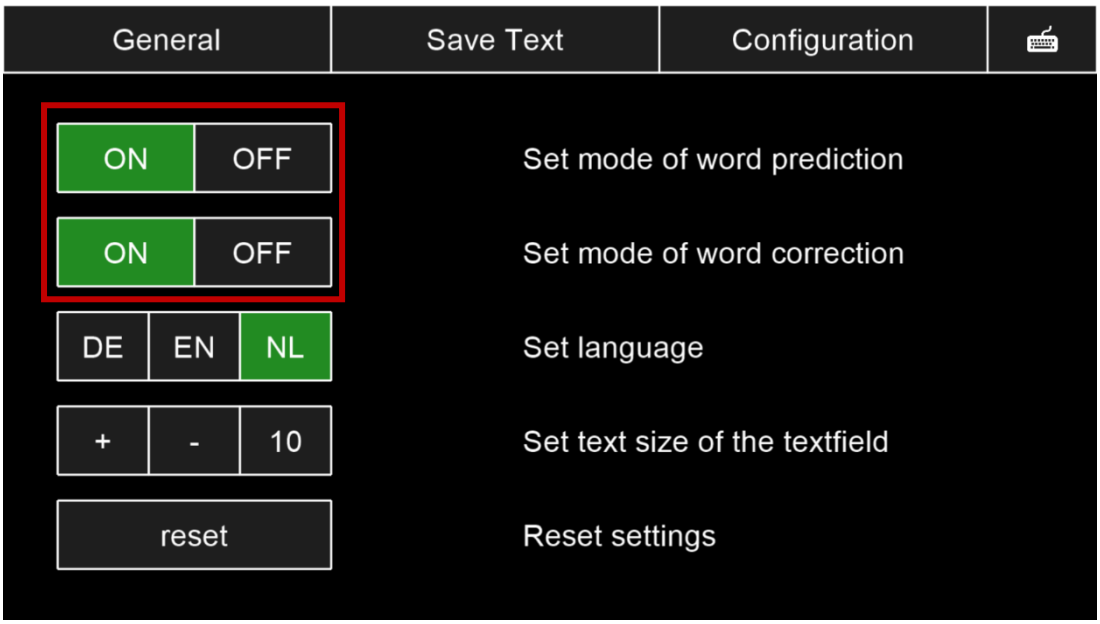

Figure 4: Toggle word prediction and correction.

Directly under that is the option to change the language (Figure 5). This setting will change the language for the word prediction and correction modules, as well as the text-to-speech option.

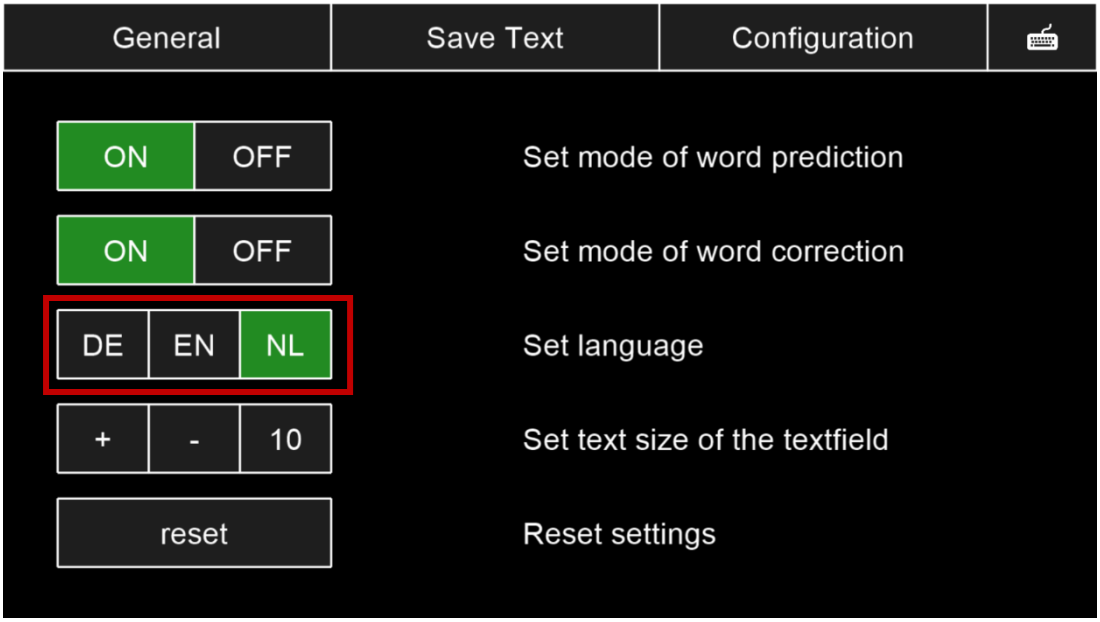

Figure 5: Change the language.

Underneath, the option is provided to change the text size of the displayed written text (Figure 6). This will not change the text size of the keyboard itself.

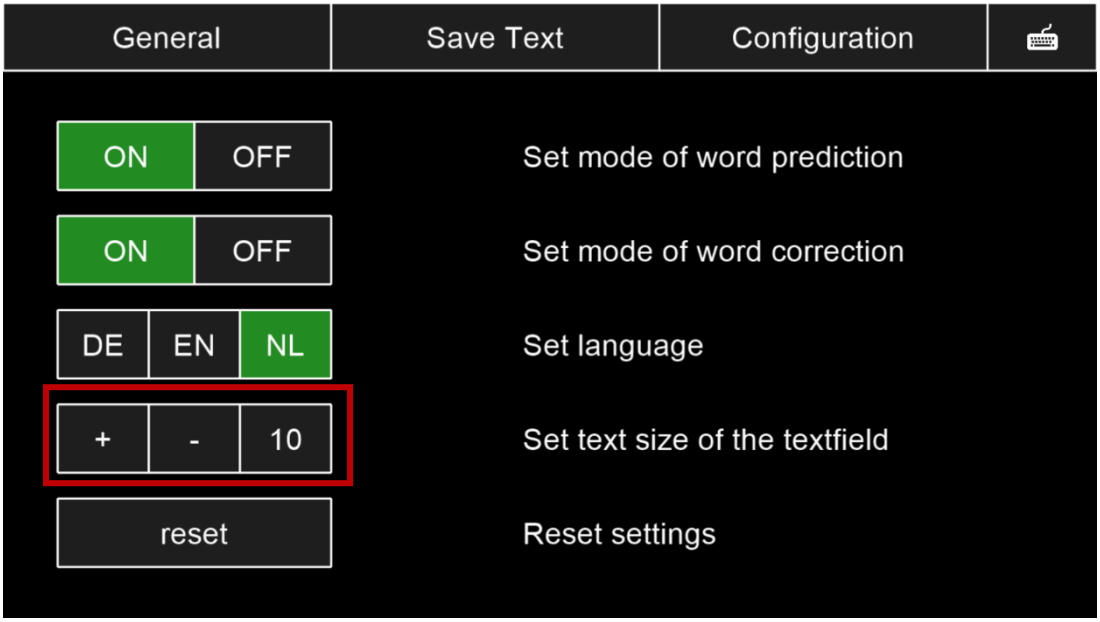

Figure 6: Change the text size.

The button at the bottom reset the settings to their default. These default settings are specified in the `user_config` file.

At the top of the screen, a navigation bar can be seen (Figure 7). This allows the user to move between different parts of the settings menu or go (back) to the keyboard.

- **Save Text** redirects to a menu where the user can save the current written text on the keyboard into a .txt file.
- **Configuration** redirects to a menu where the user can perform certain configuration tasks (Figure 8). The options there are:
  - Calibration: opens the calibration screen that also appears upon starting the application.
  - Cued prediction: starts 10 prediction trials to test if the keyboard works correctly.
  - Electrode Quality: shows whether the EEG electrodes are working properly.
- **The keyboard icon** on the right takes the user back to the keyboard where they can type.

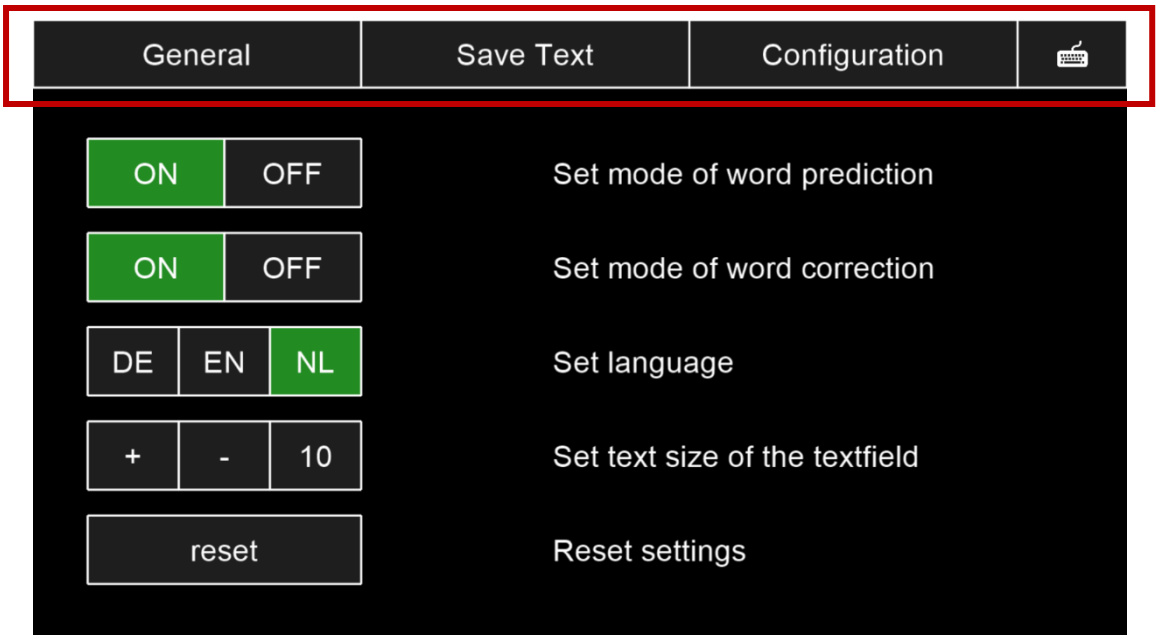

Figure 7: The navigation bar.

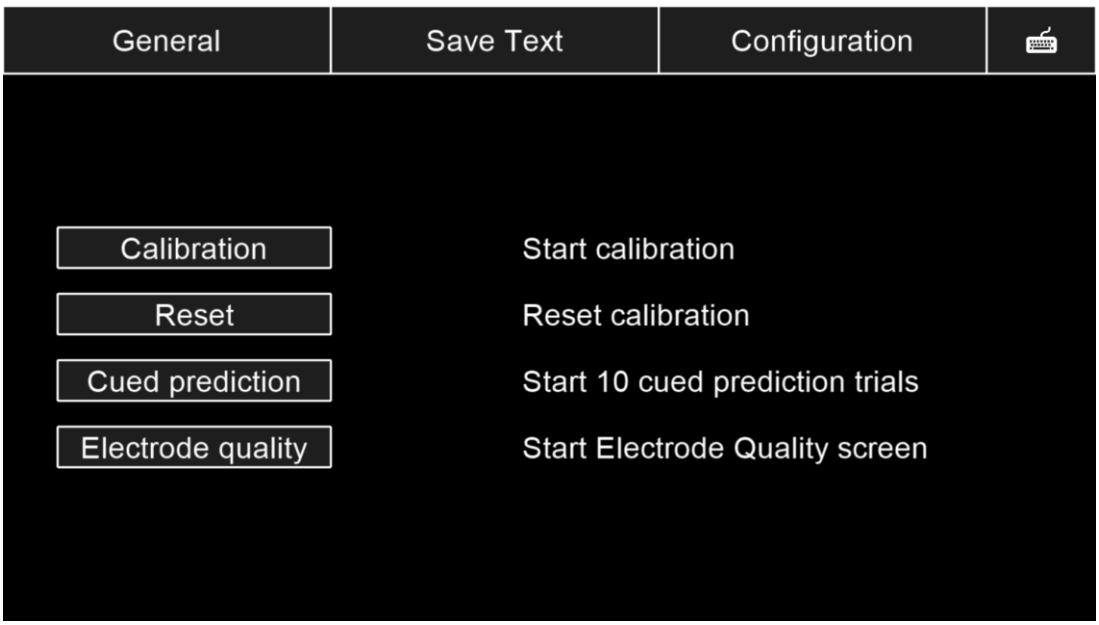

Figure 8: The Configuration menu.

Typing

To facilitate BCI-driven spelling, the keyboard assigns a specific flickering pattern to each key which the EEG headset can pick up. For a user to select a certain key or to type a letter, they have to be focussed and look at the desired key while it is flickering. The headset will pick up the flickering pattern associated with the key and will send this back to the keyboard which will then select the key and type the letter (or carry out the action tied to the key).

Word prediction, autocompletion and correction

The keyboard supports basic word prediction, autocompletion and word correction. Similar to a keyboard on a mobile phone, the keyboard provides a number of extra keys on top of the keyboard where the best suggestions will be displayed (Figure 9). These keys can be selected by the user like any other key and the predicted word will then be typed, enabling the user to type faster. A word typed with such a key can be removed in its entirety by using the backspace key.

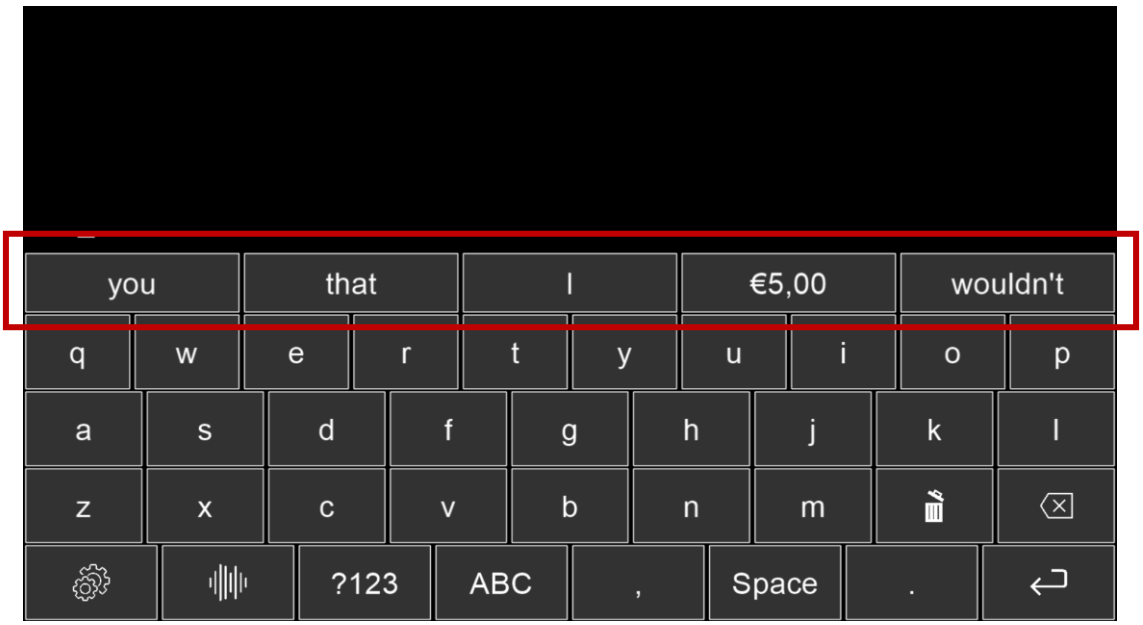

Figure 9: The prediction keys.

The word correction of the keyboard is based on frequency lists. Word prediction is done via N-gram files that have been trained on free to use e-books (for example from Project Gutenberg[14]).
The standard (pretrained) languages that are provided in this module are English, Dutch, and German. It is possible to add custom languages via creating new N-gram files by training them on texts in the desired language.
While typing, the N-gram files are updated based on what the user types. This means that frequently used words, names, or sentences will appear more often in the word prediction keys.

Text-to-speech

The keyboard supports text-to-speech. On the keyboard, a key can be found which reads the current line of written text out loud (Figure 10).

It depends on the internet connection, which text-to-speech engine is used:

- Offline: uses the pytssx3[3] module.
- Free online: uses the gTTS[4] module which uses Google Translate's TTS api.
- Paid online: uses the Google WaveNet TTS.[15] To use this feature, credentials need to be provided.

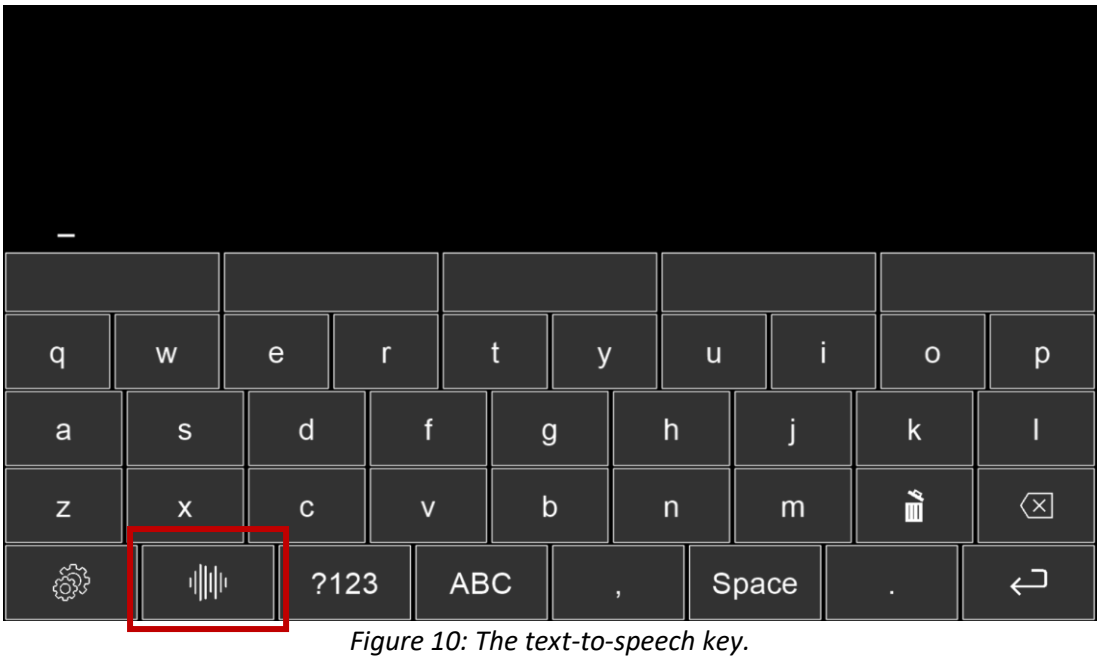

Figure 10: The text-to-speech key.

Adding custom keyboards

It is possible to add keyboards with a custom layout to the module. For this, a JSON file needs to be constructed which holds the structure of the keyboard. One way to do this, is by using this keyboard editor[16] that generates JSON files.

Unlike regular keyboards, only one output (character, word or action) is allowed per key, much like for mobile keyboards. This means no usage of shift to select a different thing on the same key. It is however possible to construct multiple keyboards (for example lower- and uppercase) and switch between them using a key on the keyboard.

The specific details on the structure of the JSON files and the possible options for keys with specific functionality can be found on our documentation page under the tab 'Guide to creating custom JSON keyboards'.

Support

For Frequently Asked Questions about the BCI implementation, go to MindAffect's FAQ page[17].

For further support, please contact MindAffect directly[1] or raise an issue on their projects github page[18].

Copyright license

All files in this project fall under the following license:

References

[1] MindAffect's official website:
https://www.mindaffect.nl/

[2] The pypi page for the mindaffectBCI module:
https://pypi.org/project/mindaffectBCI/

[3] The pypi page for the ttsx3 module:
https://pypi.org/project/pyttsx3/

[4] The pypi page for the gTTS module:
https://pypi.org/project/gTTS/

[5] The pypi page for the playsound module:
https://pypi.org/project/playsound/

[6] The pypi page for the google-cloud-texttospeech module:
https://pypi.org/project/google-cloud-texttospeech/

[7] The pypi page for the symspell module:
https://pypi.org/project/symspellpy/

[8] The pypi page for the psychopy module:
https://pypi.org/project/PsychoPy/

[9] The pypi page for the opencv module:
https://pypi.org/project/opencv-python/

[10] MindAffect's documentation page on the layout of the EEG headset:
https://mindaffect-bci.readthedocs.io/en/latest/headset_layout.html

[11] MindAffect's documentation page on how to print your own EEG headset:
https://mindaffect-bci.readthedocs.io/en/latest/printing_guide.html

[12] MindAffect's documentation page on how to use the BCI with a fake data stream:
https://mindaffect-bci.readthedocs.io/en/latest/FAQ.html?highlight=fake#can-i-use-the-mindaffectbci-without-an-eeg-acquisition-device

[13] MindAffec't documentation on how to get started with the headset and the BCI:
https://mindaffect-bci.readthedocs.io/en/latest/quickstart.html

[14] The Gutenberg Project, a library of over 60.000 free eBooks:
https://www.gutenberg.org/

[15] Google WaveNet TTS:
https://cloud.google.com/text-to-speech

[16] A keyboard editor to generate JSON files:
http://www.keyboard-layout-editor.com/#/

[17] MindAffect's FAQ:
https://mindaffect-bci.readthedocs.io/en/latest/FAQ.html

[18] MindAffect's github page:
https://github.com/mindaffect/pymindaffectBCI