

10. Anhang:

Entwicklung des BCIs im Überblick:

Selbstgebautes EEG	Ein EEG-Schaltkreis mit Instrumentierungsverstärker
Verbesserung des Codes	Der Code, der die Daten empfängt, wurde verbessert.
Konfiguration des Bluetooth	Das Bluetoothmodul wurde mit Commands zum Master gemacht.
Tastatur Emulation	Ab einem Grenzwert werden Tasten emuliert.
Ja-Nein-Programm	Mit einem Programm lässt sich „Ja“ oder „Nein“ sagen.
Spiel	Ein einfaches Spiel lässt sich mit dem BCI steuern.
Mausprogramm	Eine Maus lässt sich über den Bildschirm steuern.
Invertierte Bildschirmlupe	Der Bildschirm bewegt sich anstatt der Maus.
Schreibprogramm	Alle Buchstaben können ausgewählt werden.
Autovervollständiger	Um das Schreiben zu erleichtern, kann man Worte vervollständigen.
Sprachausgabe	Geschriebene Worte werden ausgesprochen.
Muskelstimulation	Mit dem BCI lassen sich Muskeln ansteuern.

Der Bogen für die Probanden:

Name: _____

Kontrollfragen

Schön, dass du an diesem Versuch teilnimmst! Du hast ja vorhin einige Fragen per Gedankensteuerung beantwortet. Nun bitte ich dich, die SELBEN Antworten auf die selben echten Fragen zu geben um dein gewolltest Ergebnis mit deinem erbrachten Ergebnis zu vergleichen. Vielen Dank!

Kreuze für JA an!

1. Heißt du „Echter Name“? ☐
2. Wenn der Mond zwischen Sonne und Erde ist, haben wir eine Sonnenfinsternis? ☐
3. Ist $(6+2) : 2 * 3 = 12$? ☐
4. Sie findet ihn nicht ganz so unwiderstehlich. Mag sie ihn? ☐
5. Ist die Erde flach? ☐
6. Liegt Hamburg in Deutschland? ☐
7. Steht in Australien die Sonne mittags im Süden? ☐
8. Ist $\sqrt{64} = 8$? ☐
9. Ist deine Tante die Schwester der Urgroßmutter deiner Tochter? ☐
10. Bist du 17 Jahre alt? ☐

Der Code, der die Daten ausliest und verwertet:

Da ich nicht wusste, wie die Daten ankommen, welche Synchronisationsbytes es gibt und in welcher Reihenfolge sie sind, habe ich mir ein Programm für Mindflex, ein anderes Spielzeug mit TGAT1 Chip aber ohne Bluetooth, von www.instructables.com heruntergeladen. Doch dies ist oft abgestürzt, und die Daten kamen unregelmäßig, weshalb ich es mehrere Monate verbessert und erweitert habe und mir die Rohdatenströme sowie in Neuroskys Algorithmen angeguckt habe, um es dann neu zu programmieren. Dabei half mir Sven Hünemörder, indem er mir viel erklärte und mir Mechaniken zeigte. Vielen Dank!

Code Änderungen:

In der folgenden Tabelle sieht man die wesentlichen Änderungen des Ursprungscode und was sie bewirken sollen:

Vorher	Nachher	Effekt
Rx und Tx wurden mit analogen Pins verbunden	Rx und Tx sowie Tx und Rx werden verbunden	besseres und einfacheres Auslesen der Daten
/	Einstelloption für HID oder Serial	/
Stringgröße von payloadData war 64	Stringgröße von payloadData ist 255	Die Angabe war falsch
/	Hinzufügen der Variablen für Alphawellen etc.	die Möglichkeit, alle Wellen anzuzeigen
/	Hinzufügen der Variable für Debugging	besseres Debugging
MySerial	Unterstütztes Serial1 benutzen	besserer Datenfluss
while (!mySerial.available());	while (Serial1.available() == 0) ohne !	Die Angabe war falsch
/	i in der Schleife vorher auf 0 setzen	bessere Funktionalität
/	Serial.flush(); hinzufügen	wartet, bis zum Ende der Uebertragung von Daten
return beim payloadLength Check	payloadLength einfach auf 0 setzen	bessere Funktionalität
/	die Daten für AW. etc. in Variablen speichern	die Möglichkeit, alle Wellen anzuzeigen
/	Tastatur oder Maus emulieren	/

Tab. 2: Darstellung der Veränderungen des Ursprungscode

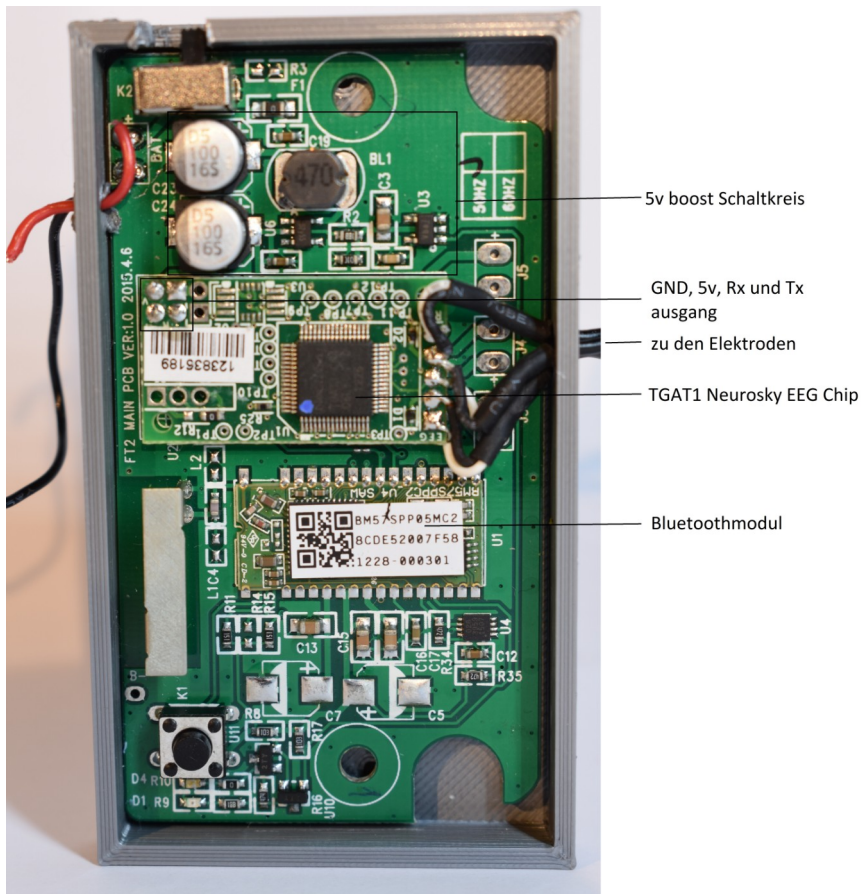
Alle Arduino Codes für dieses Projekt funktionieren im Grundprinzip gleich und sind wie folgt:

1. Variablen definieren `//int attention = 0;`
2. Byte lesen definieren `//byte ReadOneByte()`
3. nach Synchronisationsbytes suchen `//if (ReadOneByte() == 170){}`
4. Daten speichern `//payloadData[i] = ReadOneByte();`
5. Checksumme überprüfen `//if (checksum == generatedChecksum){}`
6. nach Indikator Bytes suchen für
Attention, Meditation und Qualität `//case 4:`
7. Attention, Meditation und Qualität in
einer Variable speichern `//attention = payloadData[i]`
8. die Nutzung dieser Daten, z.B ein
Tastendruck `//Keyboard.write('Taste');`

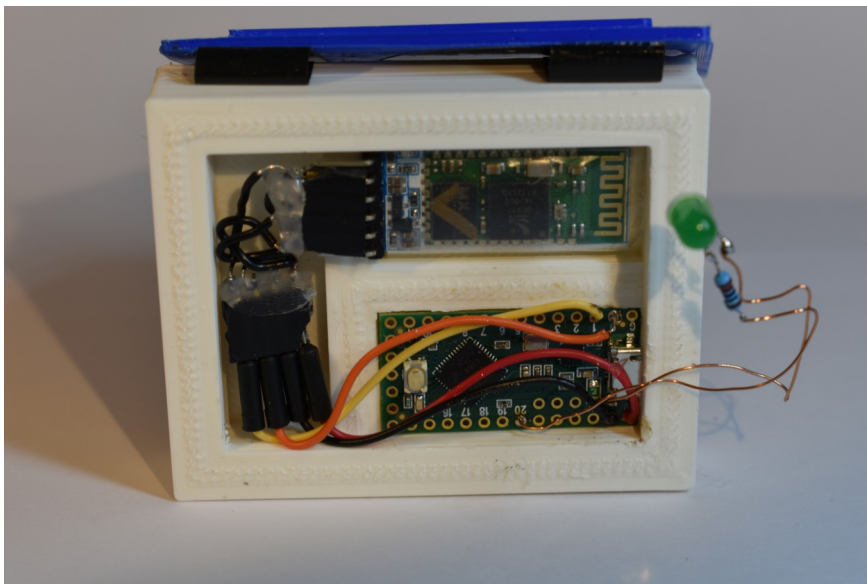
Selbstversuch zur Überprüfung der Validität aus der letzten Arbeit:

Erwartete Antwort	Antwort	Im folgenden Experiment wollte ich die Zuverlässigkeit eines solchen Sprachprogramms testen, indem ich meine Mutter bat, mir zufällig 20 „Jas“ und 20 „Neins“ zu geben, die ich mental reproduzieren musste. Um zu testen, ob das Nachdenken über eine Frage auf das Ergebnis Auswirkungen hat, stellte mir meine Mutter noch zehn echte Fragen, z.B: „Ist dein Alter 14?“. Fünf davon sollten mit „Ja“ beantwortet werden. Außerdem wurden mir noch zehn Kopfrechenaufgaben gestellt.
1=Ja 0=Nein	1=Ja 0=Nein	
1	1	Ergebnisse: Erstaunlicherweise habe ich, wie man in der linken Tabelle sieht, in 50 von 50 Antworten die richtige gegeben. Das lässt vermuten, dass dieses Verfahren sehr valide ist. Bei „Konzentration“ kann man ja denken, dass beim Lösen von Matheaufgaben oder beim Nachdenken aus Versehen „Jas“ bzw. „Neins“ ermittelt werden. Doch auch das konnte ich widerlegen, denn beim normalen Unterhalten mit Headset sowie bei komplexen Fragen oder Matheaufgaben kam es zu keinen Ausschlägen (also weder „Ja“ noch „Nein“) während des Denkprozesses. Das liegt daran, dass es bei dieser Konzentration vor allem darauf ankommt, dass man einen Tunnelblick herstellt, alles ausblendet so wie an nichts denkt. Beim Entspannen oder bei der Meditation sollte man für das bestmögliche Ergebnis keine Gedanken haben. Bei Vielen (Probanden im FSZ) und mir half es, Hintergrundgeräuschen zu lauschen.
1	1	
1	1	
1	1	
0	0	
1	1	
0	0	
0	0	
0	0	
1	1	
0	0	
1	1	
0	0	
1	1	
0	0	
0	0	
0	0	
1	1	
0	0	
1	1	
1	1	
0	0	Die Antwort, die riskanter ist, muss die schwierigere sein. Hier zeigte sich, dass ich den Entspannungszustand leichter herstellen konnte als den fokussierten. Deswegen sollte man je nach Situation die Antwort, bei der man die höhere Sicherheit benötigt, dass sie zutreffend beantwortet wird, der Herstellung des schwieriger zu erzeugenden Geisteszustand zuordnen, um einer unkontrollierten Antwort vorzubeugen. Außerdem sollte man bei wichtigen Entscheidungen natürlich darauf achten, diese mehrfach zu fragen. Bei mir ist es manchmal vorgekommen, dass ich zu viele Gedanken nach einer Frage hatte, so dass ich es nicht geschafft habe, mich zu konzentrieren oder zu entspannen und eine Antwort zunächst ausblieb. In solchen Fällen sollte man auch die Validität der Antwort infrage stellen. Als Intervallzeit zwischen Frage und Antwort dachte ich zunächst, einen Grenzwert von 15 Sekunden anzusetzen. Dieser Effekt sollte dafür sorgen, dass es bei unsicheren Antworten durch den Denkprozess zu keinem Ergebnis kommt und nicht zu einem falschen. Meine Vermutung war, dass es zu keiner Antwort kommt, da man die beiden Geisteszustände dann nicht generieren kann. Jedoch zeigte sich in Versuchen mit anderen Probanden, dass dies nicht der Fall war und die Festlegung der Intervallzeit keine Lösung für jeden ist.
0	0	
0	0	
1	1	
1	1	
0	0	
1	1	
1	1	
1	1	
0	0	
0	0	
1	1	
0	0	
1	1	
1	1	
0	0	
1	1	
0	0	
1	1	
0	0	
0	0	

Tab. 3: Erwartete und Gegebene Antworten auf 50 (Fragen)



Das EEG Headset im Inneren



Der Teensy mit Bluetoothmodul

Codebeispiel Spektrogramm:

```
import serial
import numpy as np
from numpy.fft import fft, fftshift
import pygame
import sys

pygame.init()
WIDTH = 800
HEIGHT = 512
WHITE = (255,255,255)
screen = pygame.display.set_mode((WIDTH, HEIGHT))
screen.fill(WHITE)
clock = pygame.time.Clock()

B = [0]*1024

for t in range(360):
    s=serial.Serial('COM6',38400,timeout = 1)
    S=[]
    for i in range(256):
        S.append(int(s.readline()))
    s.close()

    B = B + S
    B = B[256: 1280]
    h = np.hamming(1024)
    w = []
    for i in range(1024):
        w.append(B[i] * h[i])
    A = fft(w) / 25.5
    mag = np.abs(fftshift(A))
    freq = np.linspace(-0.5, 0.5, len(A))
    response = 20 * np.log10(mag)
    response = np.clip(response*5, -127, 127)

    for i in range(128):
        pygame.draw.rect(screen, (response[i]+128, 0, 0), (t*4, i*4, 4, 4))
    for e in pygame.event.get():
        if e.type == pygame.QUIT:
            pygame.quit()
            sys.exit()
    pygame.display.update()
```



Das BCI im Betrieb

Vielen Dank!