

Neurosky MindWave a zpracování dat pomocí mikrokontroléru

- Vedoucí semestrální práce: Ing. Pavel Mautner
- Vypracovali: Pour Vojtěch, Hájek Sebastián, Hrbek Tomáš

1. Cíl projektu

Cílem projektu je vytvoření programu pro komunikaci s EEG čelenkou Neurosky Mindwave pro mikrokontrolér M5Stack Core 2. Mikrokontrolér má umět přijímat data z čelenky za pomoci protokolu Bluetooth, kde následně tato data dekoduje a zobrazí je na GUI (Uživatelském rozhraní) a dále data zašle pomocí protokolu Uart do druhého mikrokontroléru, jenž bude dále řídit distribuci dat.

2. Řešení projektu

• Mikrokontrolér 1 (M1)

Tento mikrokontrolér zajišťuje Bluetooth spojení s EEG čelenkou a dekodování přijatých dat. Mikrokontrolér 1 komunikuje s Mikrokontrolérem 2 pomocí protokolu UART.

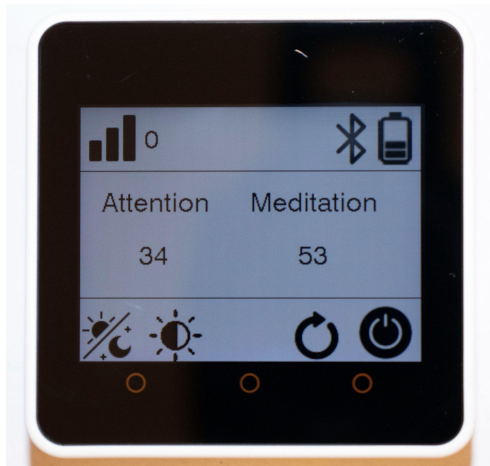
- `void connectBluetooth() { ... }`
Funkce, která vyhledá bluetooth zařízení, vypíše nalezená zařízení do Serialu a následně se snaží připojit k zařízení s odpovídající MAC adresou, která je deklarována v úvodní části programu.

```
8 String MAC_ADDRESS = "MAC_ADRESA_EEG_CHELENKY";
```
- `void drawGui() { ... }`
Funkce vykreslující GUI.
- `void updatePoorQuality() { ... }`
Funkce aktualizující hodnotu kvality signálu čelenky na obrazovce
- `void updateAttention() { ... }`
Funkce aktualizující hodnotu Attention (Pozornosti) na obrazovce
- `void updateMeditation() { ... }`
Funkce aktualizující hodnotu Meditation (Meditace) na obrazovku
- `void bluetoothBlinking() { ... }`
Funkce zajišťující blikání ikony při připojování k čelence
- `void drawBluetoothConnecting() { ... }`
Funkce vykreslující uživateli na obrazovku informaci o probíhajícím připojování bluetooth
- `void drawBattery() { ... }`
Funkce vykreslující stav baterie na obrazovku
- `byte ReadOneByte() { ... }`
Funkce která po kontrole připojení přečte jeden Byte
- `void loop() { ... }`
Hlavní tělo programu. Nejdříve zjistí, zda-li je M1 připojen k čelence a pokud ne, tak zavolá funkci pro vykreslení blikající ikony stavu Bluetooth a funkci pro připojení. Poté pomocí příslušných funkcí vykreslí GUI s posledními přijatými hodnotami. Program čeká než přijdou 2 byty s hodnotou 170 (sync) za kterými následuje délka payload (balíku dat). Pokud je payload delší než 169, tak se přeskakuje. Poté se čte payload do paměti indexovaný délkou payload a zároveň se sčítá každý přečtený byte pro získání checksum. Další přečtený byte je odeslaný checksum. Pokud jsou hodnoty checksumu získaného sčítáním hodnot a odeslaného checksum stejné, tak proběhl přenos v pořádku.

Dále dochází k samotnému parsování přijatých dat. Prochází se postupně celým payloadem. Na pozici 2 jsou hodnoty kvality signálu čelenky. Na pozici 4 jsou hodnoty attention. Na pozici 5 jsou hodnoty meditation. Na pozici 127(0x80) jsou hodnoty raw dat. Na pozici 131(0x83) jsou hodnoty frekvencí různých pásem. Poté se zavolají funkce pro aktualizaci displeje. Nakonec se posílají data po UART sběrnici jako 4 zprávy.

GUI (Uživatelské rozhraní) Mikrokontroléru 1

Light mode:



Dark mode:



Stav baterie a indikace nabíjení:	Síla signálu senzoru EEG čelenky spolu s přesnou hodnotou:	Stav Bluetooth: Pokud probíhá vyhledání EEG čelenky, tak bliká. Pokud selže připojení tak je ikona přeškrtnutá
Tlačítko jasu obrazovky - přepíná mezi 4 hladinami jasu:	Tlačítko denního/nočního nasvícení - přepíná mezi tmavým a světlým režimem obrazovky	Tlačítko pro restartování mikrokontroléru:

Tlačítko pro vypnutí mikrokontroléru: 	Všechny ikony se barevně přizpůsobí dle nastavení módu obrazovky
--	---

Struktura paketu odesílaná členkou:

- Asynchronní sériový přenos dat
- Paket se skládá ze 3 částí:
 - Hlavička paketu
 - 2x Synchronizační Byte
 - Byte udávající délku dat
 - Samotná data (Maximální délka je 169 Bytů)
 - Checksum paketu
- Délka paketu - 4 až 173 Bytů

[SYNC] [SYNC] [DELKA] ... [DATA] ... [KONTROLA]

1 byte 1 byte 1 byte 0 - 169 bytes 1 byte

Hodnoty dekodované z paketu:

- POOR_SIGNAL Quality
 - 1 Byte
 - nabývá hodnot 0 až 255
 - Hodnota 200 - Speciální stav (Členka není nasazena)
 - hodnota vyjadřuje kvalitu naměřených dat
 - hodnoty Attention a Meditation nejsou aktualizované, pokud je nízká kvalita naměřených dat
- Attention eSense
 - 1 Byte
 - nabývá hodnot 0 až 100
 - Úroveň "pozornosti"
- Meditation eSense
 - 1 Byte
 - nabývá hodnot 0 až 100
 - Úroveň "relaxace"
- Raw data
 - 2 Byty
 - nabývá hodnot od -32768 do 32768
- Úroveň mozkové aktivity
 - 8*3 Byty
 - Mozkové vlny v 8 různých pásmech v tomto pořadí
 - delta (0.5 - 2.75Hz)
 - theta (3.5 - 6.75Hz)
 - low-alpha (7.5 - 9.25Hz)

- high-alpha (10 - 11.75Hz)
- low-beta (13 - 16.75Hz)
- high-beta (18 - 29.75Hz)
- low-gamma (31 - 39.75Hz)
- mid-gamma (41 - 49.75Hz)
- Hodnoty v jednotlivých pásmech jsou bezjednotkové a mají smysl pouze při porovnávání mezi jednotlivými pásmy

• Mikrokontrolér 2 (M2)

M2 přijímá dekodovaná data z M1. Po odeslání dat z M1 se na displeji M2 zobrazí pouze požadovaná data.

- `void drawGui(screenMode COLOR_MODE){...}`
Funkce vykreslující GUI
- `void writeScreenData(dataMode DATA_MODE, screenMode COLOR_MODE, String data){...}`
Funkce vykreslující data na obrazovku.
- `void attentionHandler(Event& e){...}`
Funkce nastaví do paměti M2 příslušný mód pro attention. Po opětovném zmáčknutí se zobrazování zastaví.
- `void meditationHandler(Event& e){...}`
Funkce nastaví do paměti M2 příslušný mód pro meditation. Po opětovném zmáčknutí se zobrazování zastaví.
- `void qualityHandler(Event& e){...}`
Funkce nastaví do paměti M2 příslušný mód pro quality. Po opětovném zmáčknutí se zobrazování zastaví.
- `void rawHandler(Event& e){...}`
Funkce nastaví do paměti M2 příslušný mód pro raw data. Po opětovném zmáčknutí se zobrazování zastaví.
- `void loop() {...}`
Nejdříve program zkontroluje připojení k MQTT klientovi a poté zda-li nejsou na UART sběrnici data. Data přijatá na UART sběrnici postupně načítá do paměti. Poté zkontroluje, zda mají přijatá data délku 17. Pokud ano, tak se poté pomocí funkce `.substring(...).toInt()` z oné proměnné načítají do paměti hodnoty odpovídající jednotlivým typům. Dále se podle vybraného módu zobrazí na obrazovku pouze požadované hodnoty. Tyto hodnoty jsou pomocí MQTT publish odeslány pro další zpracování
- `void reconnect(){...}`
Zkontroluje připojení k MQTT serveru a pokud je spojení přerušeno tak naváže nové spojení
- `void setupwifi(){...}`
Nastaví wifi mod a poté připojí k wifi za využití hodnot daných na začátku programu.

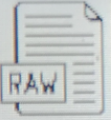
GUI (Uživatelské rozhraní) Mikrokontroléru 2

Light mode:



Dark mode:



Tlačítko jasu obrazovky - přepíná mezi 4 hladinami jasu: 	Tlačítko denního/nočního nasvícení - přepíná mezi dark mode a light mode 	Tlačítko pro restartování mikrokontroléru: 
Tlačítko pro vypnutí mikrokontroléru: 	Tlačítko pro přepnutí do režimu Attention: 	Tlačítko pro přepnutí do režimu Meditation: 
Tlačítko pro přepnutí do režimu Signal Quality: 	Tlačítko pro přepnutí do režimu Raw dat: 	Všechny ikony se barevně přizpůsobí dle nastavení módu obrazovky

3. Popis komunikačního rozhraní

M1 vysílá data jako 4 zprávy ve formátu String. Přenos dat probíhá z pinu 27 M1 do pinu 19 M2. M2 přečte přijaté data která agreguje do paměti. Po kontrole příjmu všech dat se následně rozdělí na substrings a převedou se na celočíselnou hodnotu aby se program zbavil prázdných znaků a mohl s daty dále pracovat. Poté je vykreslí na obrazovku s příslušným popiskem módu.

Struktura posílaných dat z M1 do M2

- paket všech dat Attention, Meditation, Quality a Raw data
 - data se odesílají jako **čtyři po sobě jdoucí řetězce**
 - v každém řetězci je odeslána jedna hodnota
 - **3x3 znaky a 1x8 znaků (celkem 17 znaků)**
 - neposílá se jeden celý řetězec, který by obsahoval všechny hodnoty, kvůli výraznému šumu
 - M2 poté načte všech 17 znaků např. 020020000+0000512 a znovu je rozdělí a uloží do jednotlivých hodnot
- hodnoty **Attention, Meditation, Quality** (první 3x3 znaky)
 - získají se z přijatého paketu => prvních 9 znaků (každá hodnota 3 znaky), které se rozdělí po třech a převedou se na číslo
 - např. odeslaná číslice 5 => se odešle jako 2 *prázdné znaky* a 5 " 5" => a M2 ji převede zpátky na číslo metodou toInt() => 5
- hodnota **raw data** (posledních 8 znaků)
 - získají se z přijatého paketu => posledních 8 znaků, které se převedou na číslo
 - např. 512 => " +512" => 512, -12512 => " -12512" => -12512

4. Schéma zapojení



5. Vlastnosti řešení(features)

- Po ztrátě signálu čelenky se po 5s pokusí M1 o opětovné navázání spojení
- M1 se připojí pouze k určenému zařízení.
- Uložení hodnoty jasů a módu obrazovky do paměti
- M1 má debug výstup přes USB

6. Použitá zařízení

2x M5Stack Core2

2x Vodiče dostatečné délky pro spojení komunikačních pinů

1x Neurosky Mindwave EEG čelenka

7. Rozdělení práce

- Mikrokontrolér 1 (M1) - Vojtěch Pour
- Mikrokontrolér 2 (M2) - Sebastián Hájek
- Dokumentace - Tomáš Hrbek
- Prezentace - práce všech zúčastněných

8. Kontaktní informace

- Vojtěch Pour - pourv@gapps.zcu.cz
- Sebastián Hájek - sebhajek@gapps.zcu.cz
- Tomáš Hrbek - tomhrbek@gapps.zcu.cz