



Interpreter BIOzap służy do edycji i obsługi terapii tworzonych przez użytkowników. BIOzap został zaimplementowany w otwartych, bezpłatnie dostępnych urządzeniach zbudowanych na platformie ARDUINO, a zaprojektowanych do prowadzenia terapii pulsującym prądem elektrycznym, polem magnetycznym i elektromagnetycznym oraz innych projektach służących min. do biofeedbacku, eksterioryzacji i biorezonansu.

Użytkownicy mogą wymieniać się tworzonymi przez siebie terapiami poprzez strony internetowe, samodzielnie je modyfikować i załadowywać do urządzeń.

Baza terapii i opisu budowy urządzeń obsługujących interpreter BIOzap znajduje się pod adresem: <http://biotronika.pl>

Otwarta licencja: <http://biotronika.pl/biozap-licencja>



Zielona nazwa urządzenia oznacza, że obsługuje ono opisane polecenie. **Popielata** jest użyta, kiedy funkcja jest wspierana w danym urządzeniu w sposób ograniczony. W przypadku planowanej implementacji w przyszłości danej komendy użyto koloru **niebieskiego**.

>

| [free-PEMF](#) | [zapper++](#) |

Znak zachęty w trybie bezpośrednim. Interpreter jest gotowy do wprowadzenia komendy. Komendę należy zatwierdzić znakiem końca linii #10 (na klawiaturze wcisnąć Enter). Linie puste są ignorowane. Komenda zostanie wykonana natychmiast. W przypadku wykonania programu terapii zapisanego w pamięci EEPROM i na karcie SD w przytoczonych przykładach nie występuje znak zachęty. Przykład wprowadzenia pustej linii:

```
>
>
```

.

| [free-PEMF](#) | [zapper++](#) |

W przypadku dłuższego czasu wykonania polecenia co około 1 sekundę interpreter wyprowadza znak kropki. Przykład generowania sygnału o częstotliwości 7,83Hz przez 10 sekund:

```
>rec 783 10
.....OK
>
```

[komentarz]

| [free-PEMF](#) | [zapper++](#) |

Komentarz do programu terapii. Linia jest ignorowana przez interpreter. Należy korzystać wyłącznie ze znaków ASCII z pominięciem znaku dwukropka. Komentarze zapisują się do pamięci EEPROM.

```
# Terapia jednodominutowa
rec 783 60
```

print [tekst]

| [free-PEMF](#) | [zapper++](#) |

Wyprowadza tekst na ekranie i konsoli portu szeregowego. Ekran każdego urządzenia powinien udostępniać jedną linię na komunikaty programu terapii. Jeśli długość tekstu jest większa niż wielkość ekranu to tekst zostanie obcięty. Free-PEMF nie posiada ekranu i obsługuje wyłącznie konsolę. Przykład wyświetlenia tekstu na konsoli portu szeregowego:

```
>print Zaczynamy zapowanie
Zaczynamy zapowanie
>
```



mem [adres]

| [free-PEMF](#) | [zapper++](#) |

Przekierowuje ciąg komend do pamięci (memory) EEPROM urządzenia. Maksymalna długość programu terapii to 1000 znaków. W jednym wierszu może się znajdować jedna instrukcja. Linia powinna się kończyć CR-LF lub LF (#13#10 lub #10). Wyjście z trybu zapisu do pamięci następuje przez wysłanie znaku @, przekroczenie pojemności bufora (512 znaków) lub przekroczenie długości programu (1000 znaków). Przed załadowaniem nowego programu należy poprzedni usunąć poleceniem **rm**. Domyślnie program zajmuje pamięć od adresu 0. Podając [adres] od tego miejsca następuje nadpisanie pamięci. Wstawienie znaku @ w miejsce adresu powoduje dopisanie nowych komend na końcu istniejącego w pamięci skryptu.

```
>mem
rec 783 120
rec 1179 120
@
>
```

ls [|-n]

| [free-PEMF](#) | [zapper++](#) |

Wyświetla listę (list) programu terapii zapisanego w pamięci EEPROM. Parametr -n służy do wyświetlenia adresów linii programu. Ostatnia zajęta komórka pamięci jest oznaczona znakiem @. Patrz: **mem [adres]**.

```
>ls
rec 783 120
rec 1178 120
>
```

ls [nazwprog.roz]

| [zapper++](#) |

Wyświetla listę programu terapii zapisanego na karcie SD. Brak programu o danej nazwie lub niedostępność karty SD zwraca błąd WE/WY.

```
>ls prog1.txt
Error sd input!
>
```

exe

| [free-PEMF](#) | [zapper++](#) |

Wykonaj (execute) program zapisany w pamięci EEPROM. Polecenie działa tylko z konsoli w trybie bezpośrednim.

```
>exe
.....OK
Teraz 2 fragment
.....OK
>
```

Program terapii zapisany w pamięci EEPROM:

```
# Terapia w EEPROM
rec 783 30
print Teraz 2 fragment
rec 1179 30
```



exe [nazwprog.roz]

| [free-PEMF](#) | [zapper++](#) |

Wykonaj program zapisany na karcie SD. W katalogu głównym karty SD powinien znajdować się wskazany plik programu. Jego nazwa składa się maksymalnie z 8 znaków oraz trzy-znakowego rozszerzenia (konwencja nazywania plików w systemie FAT). Długość programu terapii nie jest ograniczona.

```
>exe prog1.txt
```

rm

| [free-PEMF](#) | [zapper++](#) |

Usuń (remove) program z pamięci EEPROM. Wyczyść pamięć.

```
>rm
.....OK
>
```

sin [częstotliwość] [czas]

| [free-PEMF](#) | [zapper++](#) |

Generuj sygnał sinusoidalny o podanej w setnych częściach Hertza częstotliwości. Okres generowania sygnału podany jest w sekundach. Przykład wygenerowania sygnału wyjściowego o częstotliwości 11,79Hz przez 2 minuty.

```
sin 1179 120
```

rec [częstotliwość] [czas]

| [free-PEMF](#) | [zapper++](#) |

Generuje sygnał prostokątny (rectangle) o podanej częstotliwości w setnych częściach Hz. Czas podany jest w sekundach. Przykład wygenerowania sygnału wyjściowego o częstotliwości 7,83Hz przez 1 minutę.

```
rec 783 60
```

wait [czas]

| [free-PEMF](#) | [zapper++](#) |

Zatrzymanie programu terapii przez okres podany w milisekundach np. na dwie sekundy:

```
wait 2000
```

chp [0|1]

| [free-PEMF](#) | [zapper++](#) |

Zmień polaryzację (change polarity) sygnału wyjściowego. Domyślne po włączeniu jest ustawiona wartość 0 co oznacza w urządzeniach z oznaczonymi zaciskami, że biegun dodatni ma kolor czerwony, a ujemny czarny. Dla free-PEMF odpowiednio: zacisk (2) dodatni, a (4) ujemny. Dla wartości 1 polecenia polaryzacja zacisków zostaje zamieniona.

```
rec 783 120
chp 1
rec 783 120
chp 0
rec 1178 120
```

vout [0-150]

| [free-PEMF](#) | [zapper++](#) |

Ustaw napięcie na wyjściu (voltage out). Instrukcja bez parametru zwraca poziom napięcia wyjściowego. Wartości podawane są w dziesiątych częściach volta. Przykład ustawienia napięcia 12,0V .

```
>vout 120
OK
>vout
120
```



vmin [0-150]

| [zapper++](#) |

Ustawia dolną wartość napięcia sygnału (voltage minimum). Instrukcja bez parametru zwraca poziom napięcia wyjściowego. Wartości podawane są w dziesiątych częściach volta. Przykład ustawienia napięcia minimalnego na poziomie 0,2V .

```
>vmin 2
OK
>vmin
2
```

scan [częstotliwość do] [czas]

| [free-PEMF](#) | [zapper++](#) |

Skanuje, czyli płynnie zmienia częstotliwość od ostatnio ustawionej do docelowej. Przeskanowanie pasma trwać będzie określony czas. Jeśli częstotliwość wyjściowa nie była ustawiona to przyjmuje się wartość 0,01Hz. Jeśli wartość startowa jest wyższa niż docelowa to urządzenie będzie zmniejszać stopniowo częstotliwość. W przykładzie skanowanie od 15Hz do 20Hz będzie trwało 2 minuty.

```
rec 1500 5
scan 2000 120
```

bat

| [free-PEMF](#) | [zapper++](#) |

Podaj poziom baterii. Zwrotnie otrzymuje się wartość napięcia. Wynik należy podzielić przez 10. W przykładzie zmierzone napięcie baterii wynosi 12,5V.

```
>bat
125
>
```

cbat [0-254]

| [free-PEMF](#) | [zapper++](#) |

Kalibracja pomiaru napięcia baterii (calibrate battery). Za parametr należy podać zmierzone napięcie na zaciskach zasilania. W urządzeniu free-PEMF należy zmierzyć napięcie pomiędzy (1) i (5) pinem gniazda. Napięcie podajemy w liczbach całkowitych po przemnożeniu przez 10. Np. kalibracja dla zmierzonego napięcia 12,3V.

```
>cbat 123
>OK
```

off

| [free-PEMF](#) | [zapper++](#) |

Wyłącz urządzenie. Całkowite wyłączenie nie jest możliwe jeśli port USB jest podłączony do włączonego komputera PC. W przykładzie przed wyłączeniem urządzenie wygeneruje krótki sygnał dźwiękowy.

```
beep 200
off
```

beep [czas]

| [free-PEMF](#) | [zapper++](#) |

Generowanie sygnału dźwiękowego o określonym czasie w milisekundach. Przykład generowania sygnału dźwiękowego przez 0,2s

```
beep 200
```

jump [etykieta] [ilość powtórzeń]

| [free-PEMF](#) | [zapper++](#) |

Skok do etykiety. Każda etykieta składa się z dwukropka oraz nazwy (bez spacji). Etykieta zajmuje osobną linię i może się składać maksymalnie z 14 znaków.



Wielkość znaków ma znaczenie, np. litery: A i a są traktowane jako różne. Licznik powtórzeń jest liczbą z zakresu od 1 do 255. Po wykonaniu odpowiedniej ilości skoków program przejdzie do komendy znajdującej się po **jump**. Jeśli jako ilość powtórzeń zostanie wprowadzone 0 to skok do etykiety zostanie zignorowany. Brak parametru ilości powtórzeń oznacza bezwarunkowy skok, który nie jest limitowany przez ilość powtórzeń. Przykład programu trwającego 10 minut i kończącego się sygnałem dźwiękowym:

```
:start
rec 783 30
chp 1
rec 783 30
chp 0
jump start 10
beep 500
```

hr

| [free-PEMF](#) |

Podaj aktualny puls użytkownika (heart rate). Wynik jest zwracany w ilości skurczów serca na minutę.

```
>hr
48
>
```

jump [etykieta] [min] [max]

| [free-PEMF](#) |

Skok do etykiety po osiągnięciu odpowiedniego poziomu pulsu mierzonego przez przystawkę pulsometr-biofeedback. Skok jest wykonywany gdy wartość pulsu użytkownika jest pomiędzy określonymi wartościami **min** \leq **hr** \leq **max** . Poziomy puls min i max mogą przyjmować wartości od 20 do 255. Liczby od 0 do 10

oznaczają wartości referencyjne pulsu zapisane w pamięci EEPROM podczas kalibracji urządzenia dla danego użytkownika.

```
rec 1179 120
chp 0
# jeśli puls poniżej 2 poziomu ref. skocz deep_delta
jump deep_delta 20 2
beep 200
off
:deep_delta
rec 100 120
chp 1
rec 100 120
beep 200
off
```

Przykładowy skrypt terapii trwającej 8 minut

```
# Terapia rytm Ziemi
# Czas 8 minut
beep 200
rec 783 120
chp 1
rec 783 120
chp 0
rec 1179 120
chp 1
rec 1179 120
beep 200
off
```